Off-road Autonomous Driving via Guided Reinforcement Learning

Vedant Mundheda

CMU-RI-TR-25-78

Jul 24, 2025

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Dr. Jeff Schneider *(chair)*
Dr. Wennie Tabib
Brian Yang

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my sister and parents

Abstract

Off-road autonomous driving presents a complex set of challenges, including navigation through unmapped environments, variable terrain geometries, and uncertain, non-stationary dynamics. These conditions demand planning and control strategies that are both long-horizon and adaptable. Traditional Model Predictive Control (MPC) methods rely on dense sampling and precise dynamics modeling, which limits their feasibility for real-time planning in unstructured terrains. In contrast, Reinforcement Learning (RL) approaches offer fast execution but suffer from poor exploration efficiency, particularly in obstacle-dense and dynamically diverse settings.

This thesis proposes a hierarchical autonomy framework that integrates a low-frequency, long-horizon planner with a high-frequency, reactive RL-based controller. To overcome the exploration limitations of RL, the thesis introduces a novel teacher-student training paradigm. A teacher policy, trained off-policy using expert trajectories or heuristics, guides the learning process of a student policy trained on-policy. The thesis further extends the Proximal Policy Optimization (PPO) algorithm with a new hybrid policy gradient formulation that effectively leverages off-policy guidance alongside stable on-policy updates.

The proposed approach is validated in a realistic off-road simulation environment and benchmarked against standard RL and imitation learning baselines, showing improved terrain traversal and obstacle avoidance. Additionally, the trained policy is deployed on Sabrecat, a full-scale autonomous off-road ground vehicle. Experimental results demonstrate successful real-time execution, robust obstacle avoidance, and generalization to novel, complex terrains. This thesis contributes a practical and scalable solution to long-horizon off-road autonomy by combining hierarchical planning and guided reinforcement learning.

Acknowledgments

My journey at Carnegie Mellon University has been nothing short of transformative. It has been a whirlwind of challenges, deep learning (pun intended), late-night breakthroughs, and growth — none of which would have been possible without the support, guidance, and encouragement of some truly remarkable individuals. I am deeply grateful to all those who stood by me and made this journey so meaningful.

First and foremost, I would like to express my deepest gratitude to my advisor, Prof. Jeff Schneider. His insightful guidance, constant encouragement, and belief in my ideas have been instrumental in shaping this thesis. His mentorship taught me how to think critically, communicate clearly, and stay focused on the big picture - lessons that go far beyond research. I am truly fortunate to have had the opportunity to learn from him.

I would also like to thank my thesis committee members, Prof. Wennie Tabib and Brian Yang, for their time and valuable feedback.

I owe everything to my parents and my sister, who have been my strongest source of strength and unconditional support. Their endless patience, understanding, and belief in my potential have kept me going through every obstacle. Whether it was late-night phone calls or just knowing they were there for me, their presence has been my emotional anchor.

This thesis would not have been possible without the collaboration of my amazing project partners — Zhouchonghao Wu, Raymond Song, and Aman Mehra. Working with them was not just intellectually rewarding, but also a lot of fun. I am thankful for the countless hours we spent brainstorming, debugging, and building together.

I also want to thank my closest friends — Rohit (Pool champion), Alex (Kidding Champion), Karan (Crying champion), Parth (Driving champion), Raymond (Lifting Champion), Nomaan (Chess champion) and Aniket (Reginald Champion)— for being an unwavering source of support throughout this journey. From technical discussions and mock interviews to sports and comic relief, they have been by my side in ways big and small. I genuinely cherish the memories we made during this chapter of our lives.

A big shoutout to all my other friends — Michaela, Wentse, Xintong,

Mayank, Krish, Anisha, Arsh, Vihaan, Aryan, Srujan, and Yash — for all the good times, random conversations, late-night hangouts, parties and their constant positivity. Your friendship made CMU feel like home.

Finally, I'd like to thank everyone else — professors, labmates, classmates, and staff — who contributed to my experience at CMU, directly or indirectly. This thesis may carry my name, but it reflects the collective impact of a community that supported and inspired me every step of the way.

Thank you all for making this journey so special.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Autonomous ground vehicles (AGVs) have emerged as one of the most promising technologies in robotics and intelligent transportation systems. Their ability to perform navigation tasks without human intervention has opened up a wide range of applications, including urban delivery, search and rescue, planetary exploration, agricultural monitoring, and self-driving taxis. In structured environments such as cities and highways, the success of AGVs has been driven by high-definition maps, reliable sensor fusion, and robust control strategies. However, achieving autonomy in unstructured off-road environments remains a largely unsolved problem.

Off-road autonomy refers to the navigation of AGVs in natural terrains without predefined roads, markings, or structured environments. These terrains can include forests, deserts, rocky paths, agricultural fields, snow, or muddy regions. The unpredictability and diversity of these settings introduce unique challenges that are not present in urban driving. Such environments are highly dynamic, often partially observable, and subject to physical uncertainties that require real-time adaptation.

There is significant real-world demand for off-road autonomy. Applications include defense logistics in battlefield terrains, autonomous exploration in extraterrestrial landscapes, remote delivery in inaccessible regions, and automated farming. Human intervention in these environments can be dangerous, slow, or expensive. Hence, robust autonomous systems that can perceive, plan, and act in real time over long

Figure 1.1: Real-world deployment platform used for reinforcement learning experiments. The off-road autonomous vehicle, equipped with onboard sensors and compute, is capable of traversing complex terrains such as gravel, vegetation, and slopes—providing a challenging and realistic testbed for learning-based control algorithms.

horizons are necessary.

1.2 Challenges in Off-Road Autonomy

Navigating off-road environments presents a set of interdependent challenges that compound the difficulty of achieving reliable autonomy:

1. **Complex Vehicle-Terrain Dynamics**: The interaction between a vehicle and natural terrain involves nonlinear dynamics due to wheel-soil interactions, varying friction coefficients, deformable terrain, and sudden changes in elevation. Modeling these interactions accurately is nearly impossible due to the stochastic and heterogeneous nature of terrain surfaces.
2. **Terrain Diversity**: Off-road environments can consist of sand, mud, gravel, rocks, slopes, and vegetation—often within the same operational zone. Each type of terrain requires a different control response for effective traversal.
3. **Unstructured and Dynamic Obstacles**: Unlike road scenarios, off-road terrains lack clear semantic boundaries, and obstacles can include bushes, logs, animals, or erosion features that dynamically alter the navigable path.
4. **Limited Observability**: Sensors such as LiDAR and stereo cameras may fail in dense vegetation, dust, or water-logged areas, leading to noisy or incomplete observations.
5. **Long-Horizon Planning with Local Reactivity**: Traversal decisions must balance local control with strategic foresight. For example, choosing a path through soft sand might appear optimal locally but can lead to entrapment if a slope lies ahead.

Figure 1.2 provides visual examples of such terrain complexities, ranging from cliff faces and loose sand to dense vegetation and extreme elevation changes.

Figure 1.2: Autonomous vehicle navigating diverse off-road terrains: (i) steep cliff face, (ii) loose sand and gravel, (iii) dense vegetation, and (iv) extreme slopes scattered with boulders. These scenarios highlight the challenges of perception, traction, and control in real-world environments.

1.3 Limitations of Existing Approaches

1.3.1 Model Predictive Control (MPC)

Model Predictive Control is a popular method in autonomous driving that uses a dynamic model of the vehicle to predict future states over a receding horizon and optimize control actions. In off-road applications, MPC has been used for aggressive maneuvering and constrained obstacle avoidance [5, 27]. However, these methods require:

- **Accurate Dynamics Models**: Modeling tire-terrain interaction, suspension deformation, and slip dynamics is highly complex and often terrain-dependent.

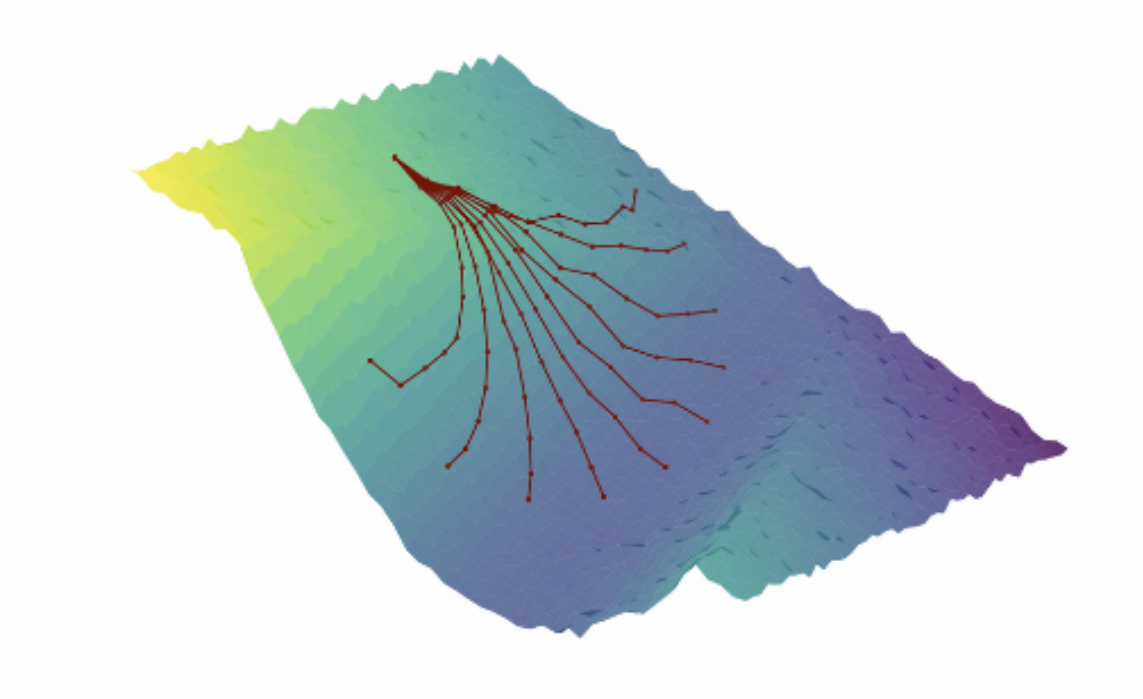

Figure 1.3: MPC incurs high computational cost, as sampling and rolling out complex interactions in real time is expensive.

- **High Computational Cost**: MPC performs iterative optimization, which becomes intractable in real time when dealing with nonlinear dynamics and high-dimensional environments.
- **Limited Planning Horizon**: The computation budget restricts planning depth, making MPC insufficient in scenarios requiring long-horizon decision making.

1.3.2 Reinforcement Learning (RL)

RL has shown promise in learning control policies for high-dimensional systems with complex dynamics . In off-road driving, RL-based controllers have been used to handle terrain-induced uncertainties [11]. However, existing methods face several issues:

- **Poor Exploration**: Without proper guidance or reward shaping, RL agents struggle with sparse reward signals in large state spaces.

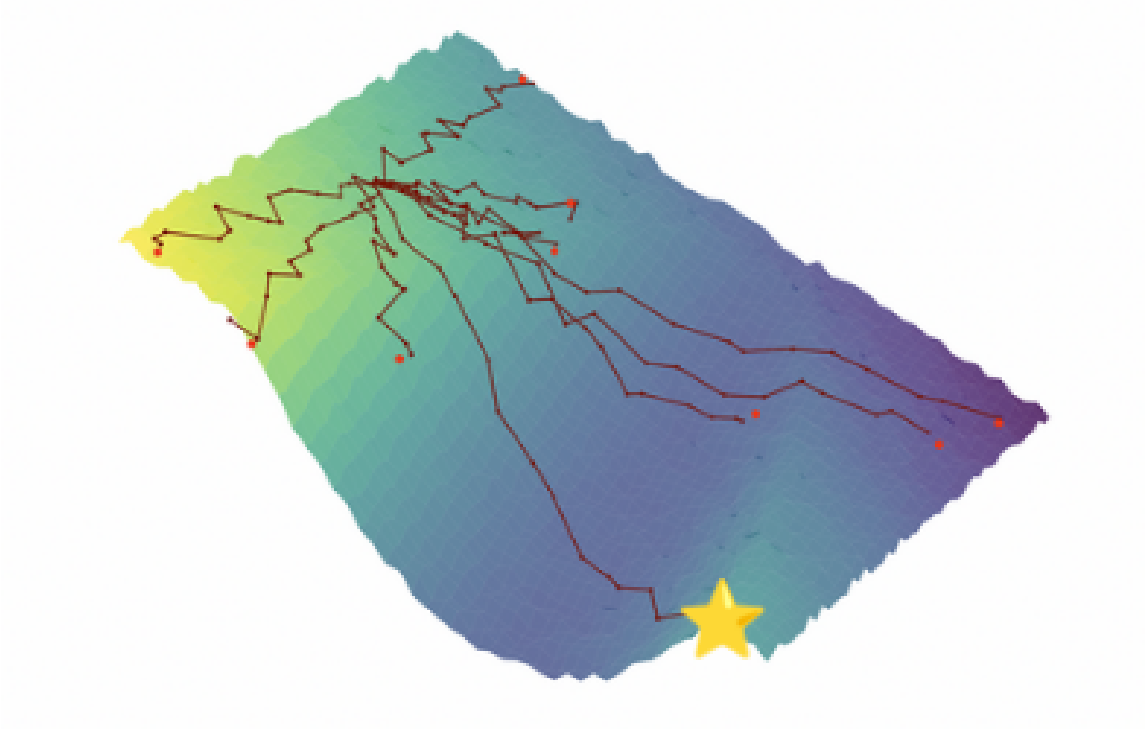

Figure 1.4: Due to sparse rewards and high-dimensional state spaces, RL methods often exhibit inefficient exploration, which can lead to degenerate or unstable policy behavior.

- **Lack of Planning**: RL policies are typically reactive and short-sighted unless trained with auxiliary mechanisms.
- **Domain Gap**: Policies trained in simulation often fail to generalize to real-world due to differences in terrain appearance and dynamics [7].

1.4 Our Approach

In this thesis, we present a novel method that addresses the shortcomings of both MPC and RL approaches by introducing a hybrid technique that combines long-horizon planning capabilities with real-time reactive control. Our contributions are summarized below:

- **Teacher Action Distillation with Policy Optimization (TADPO)**: We propose TADPO, a novel extension of Proximal Policy Optimization (PPO)

[26], which enables the policy to learn from both high-quality, offline expert demonstrations and on-policy rollouts. This framework accelerates learning and improves policy robustness.

- **Hierarchical Autonomy Stack**: We design an end-to-end navigation system that integrates a TADPO-based reactive controller with a global planner. The global planner provides coarse waypoints, while the local policy executes fine-grained maneuvers in real time.
- **Real-World Deployment**: We implement and deploy our system on a full-scale autonomous off-road vehicle platform and evaluate its performance on complex terrain types including sand, gravel, slopes, and vegetation.
- **Comprehensive Evaluation**: We benchmark TADPO against standard MPC and RL baselines under equal computation constraints and demonstrate improved performance in trajectory efficiency, terrain adaptability, and inference speed.

1.5 Thesis Overview

The rest of this thesis is organized as follows:

- **Chapter 2**: Reviews the existing literature on off-road autonomy, control methods, reinforcement learning, and hybrid approaches.
- **Chapter 3**: Presents the TADPO algorithm and discusses the theoretical formulation, learning objectives, and training pipeline.
- **Chapter 4**: Describes the full autonomy system including the perception module, planner, and deployment setup.
- **Chapter 5**: Describes the experimental setup and Simulation details.
- **Chapter 6**: Provides results in simulation and compares then with existing baselines.
- **Chapter 7**: Provides the deployment procedure on Sabrecat.
- **Chapter 8**: Summarizes the contributions and outlines future research directions.

Chapter 2

Background and Related Works

This chapter outlines the theoretical foundations and situates our work within the broader landscape of autonomous off-road navigation, reinforcement learning, model-based control, and learning from demonstration. We begin by formalizing the decision-making problem faced by an autonomous ground vehicle operating in unstructured environments. Subsequently, we review the key control paradigms, including Proximal Policy Optimization (PPO), Model Predictive Control (MPC), and hybrid learning approaches. Finally, we discuss prior work in imitation learning and demonstration-guided reinforcement learning that inspire our proposed method.

2.1 Background

The decision-making and control task for autonomous off-road navigation is modeled as a Markov Decision Process (MDP), defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$:

- $\mathcal{S}$: a continuous or high-dimensional state space capturing observations such as terrain geometry, vehicle velocity, and sensor readings.
- $\mathcal{A}$: the action space, including continuous control commands such as throttle, brake, and steering.
- $P(s'|s, a)$: the (often unknown) transition probability capturing the dynamics of vehicle-terrain interactions.
- $r(s, a)$: a reward function quantifying task performance, such as forward progress

or terrain traversability.

- $\gamma \in [0, 1)$: the discount factor encoding the trade-off between immediate and future rewards.

The goal of the agent is to learn an optimal policy $\pi^*(a|s)$ that maximizes expected cumulative reward:

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right].$$

Off-road driving presents unique challenges due to uncertain, discontinuous dynamics, non-flat terrains, and perceptual aliasing from visual observations. These properties demand both high sample efficiency and real-time inference capabilities.

2.2 Proximal Policy Optimization

Proximal Policy Optimization (PPO) [25] is a first-order, on-policy reinforcement learning algorithm designed for stability and sample efficiency. PPO builds upon policy gradient methods, where the policy π_θ is optimized directly via gradient ascent on expected return. However, naive updates can result in large policy shifts, destabilizing learning. PPO mitigates this by clipping the objective function, constraining updates within a trust region.

The PPO loss consists of three components:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t \right) \right], \tag{2.1}$$

$$L^{\text{VF}}(\theta) = \mathbb{E}_t \left[\left(V_{\pi_{\theta_{\text{old}}}}(s_t) - R_t \right)^2 \right], \tag{2.2}$$

$$L^{\text{entropy}}(\theta) = \mathbb{E}_t \left[-H[\pi_\theta(\cdot|s_t)] \right], \tag{2.3}$$

$$L^{\text{PPO}}(\theta) = L^{\text{CLIP}} - c_1 L^{\text{VF}} + c_2 L^{\text{entropy}}. \tag{2.4}$$

Here, $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio of the new and old policy, $\hat{A}_t$ is the advantage estimate, and $H[\pi_\theta]$ promotes exploration by maximizing policy entropy.

PPO has achieved state-of-the-art performance in tasks such as legged locomotion [1], robotic manipulation [20], and aerial control [22]. However, in sparse-reward,

long-horizon domains like off-road driving, PPO suffers from poor exploration and slow convergence. Moreover, its on-policy nature prevents leveraging expert demonstrations or off-policy trajectories, limiting its scalability in real-world training.

2.3 Model Predictive Control

Model Predictive Control (MPC) is a classical paradigm in robotics and control that plans over a finite horizon by solving an optimization problem at every timestep. Given a known or learned dynamics model, MPC simulates future trajectories and selects actions that minimize a cumulative cost. Sampling-based variants are particularly suited to high-dimensional and nonlinear dynamics.

Two widely used MPC algorithms are:

- **Model Predictive Path Integral (MPPI)** [29]: Samples trajectories using stochastic control noise and reweights them by exponentiated cumulative cost. MPPI is fully differentiable and parallelizable.
- **Cross-Entropy Method (CEM)** [14]: Iteratively samples and refines a Gaussian distribution over action sequences by minimizing trajectory cost.

MPPI has been applied in high-speed off-road racing [5] and traversability-aware control [16], but its dense sampling requirement makes long-horizon planning intractable in real time. Hybrid approaches like TD-MPC [6] reduce this burden using terminal value estimates, while RL+MPPI [23] learns proposal distributions to guide sampling, but both still depend on expensive dynamics rollouts.

2.4 Reinforcement Learning for Driving

RL algorithms are increasingly used for autonomous driving due to their capacity to learn policies from interaction without needing explicit supervision. Soft Actor-Critic (SAC) [4], an off-policy method that maximizes entropy, is particularly effective in continuous control domains. However, RL in off-road contexts remains underexplored.

Off-road RL methods such as WROOM [11] demonstrate reactive obstacle avoidance on rough terrain, but lack long-term planning. Other works use imitation learning from human demonstrations [7], or rely on simplified simulators [28]. On-road RL

research focuses on decision-making under traffic interactions [9, 13], which differs from off-road navigation dominated by terrain geometry and traction variability.

Exploration remains a key bottleneck for RL in off-road driving, especially under sparse reward signals. Goal-conditioned RL [3, 19] attempts to address this by conditioning the policy on a desired goal state, but such methods are computationally demanding when using pixel-level observations.

2.5 Imitation Learning and Demonstration-Based Methods

Imitation Learning (IL) aims to learn a policy that mimics expert behavior. One of the foundational IL algorithms is DAgger [24], which aggregates states visited by a learner and queries the expert to improve robustness. Offline RL methods like Implicit Q-Learning (IQL) [15] learn value functions from fixed datasets without explicit policy optimization.

Hybrid methods combining RL and IL offer a promising direction. PPO+D [17] integrates a single expert demonstration with PPO training but suffers from scalability issues in visual domains with large replay buffers. Other teacher-student architectures, such as EGPO [21] and SAC+LfD [18], alternate control between student and teacher. However, these rely on SAC, which exhibits high variance and unstable convergence in long-horizon visual control [10].

Some methods like DQfD [8] and policy distillation [12] operate in discrete or low-dimensional action spaces, limiting applicability to complex driving tasks. Our work introduces TADPO, a new approach that directly distills long-horizon expert behavior into PPO, allowing stable and efficient training on off-road tasks while retaining the benefits of real-time policy inference.

Chapter 3

TADPO: Teacher Action Distillation with Policy Optimization

In this chapter, we present TADPO (Teacher Action Distillation with Policy Optimization), a novel algorithm designed to integrate the strengths of reinforcement learning and imitation learning by directly distilling actions from a pre-trained expert policy into a student PPO agent. This method enables learning in complex, long-horizon tasks such as off-road navigation by leveraging both teacher guidance and on-policy experience.

3.1 Motivation

Traditional PPO struggles with exploration in off-road environments where terrain interactions are complex and rewards are sparse. While imitation learning from expert demonstrations can improve sample efficiency and guide the agent, it often requires full offline training or pretraining stages. In contrast, TADPO enables seamless integration of demonstration-based updates within the PPO learning loop. Our goal is to augment PPO with guided supervision from a teacher policy while retaining its on-policy training advantages.

3.2 Teacher Action Distillation Policy Gradient

Given a pre-trained teacher policy μ and a student policy π_θ, we define the TADPO loss:

$$L^{\text{TAD}}(\theta) = L^{\mu}(\theta) + c_2 L^{\text{entropy}}(\theta), \tag{3.1}$$

$$\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t^\pi)}{\mu(a_t|s_t^\mu)}, \tag{3.2}$$

$$\hat{\Delta}_t = R(a_t, s_t) - V_{\pi_{\theta_{\text{old}}}}(s_t^\pi), \tag{3.3}$$

$$L^{\mu}(\theta) = \mathbb{E}_{a_t\sim\mu}\left[\max\left(0, \min(\rho_t(\theta), 1+\epsilon_\mu)\hat{\Delta}_t\right)\right]. \tag{3.4}$$

This formulation clips the advantage-weighted log-likelihood ratio, analogous to PPO, but applied to off-policy teacher rollouts. The update is applied only if the teacher outperforms the student's value prediction, avoiding updates from poor demonstrations.

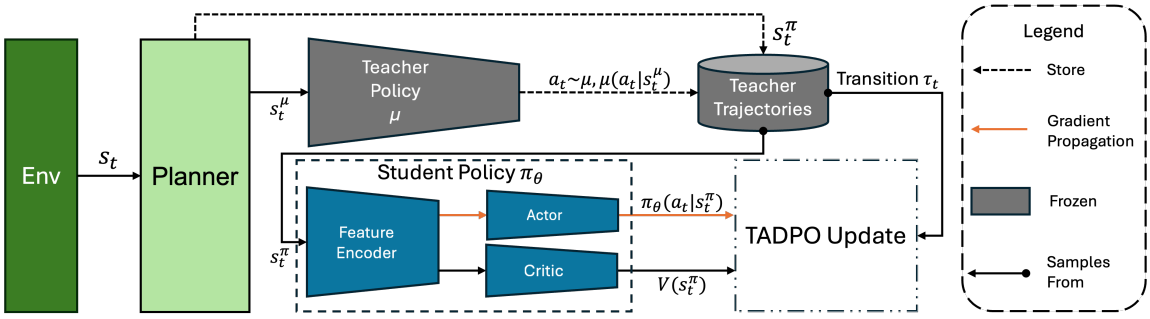

Figure 3.1: TADPO training process: At each iteration, the agent performs a TADPO update with probability p, wherein it samples transitions from the teacher policy's buffer. These transitions are used to optimize the student's policy via a KL-regularized PPO objective. During this update, only the actor and encoder parameters of the student policy are updated, while the critic remains fixed. This selective update strategy stabilizes training and encourages the student to align its behavior with the teacher without overfitting to the critic's value estimates.

3.3 Loss Clipping Mechanism

To ensure training stability, we adopt a clipped surrogate loss similar to PPO. As illustrated in Figure 3.2, updates are suppressed when the student already assigns high probability to the teacher action, ensuring that learning saturates naturally as the student mimics the teacher.

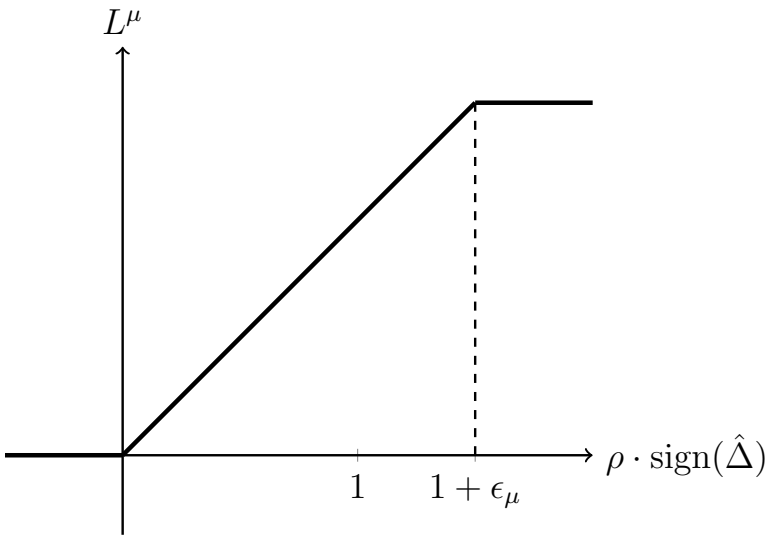

Figure 3.2: A single timestep visualization of the teacher distillation loss L^μ as a function of $\rho \cdot \text{sign}(\hat{\Delta})$, where ρ denotes the importance weight and $\hat{\Delta}$ represents the estimated advantage. Analogous to PPO's clipped objective, this modified clipping mechanism ensures stable optimization while promoting convergence toward higher reward policies.

3.4 Training Algorithm

TADPO alternates between PPO and TADPO updates. During training, trajectories are collected from both the teacher policy μ and the current student policy π_θ. With probability p, the algorithm performs a TADPO update on a batch of teacher transitions; otherwise, a standard PPO update is applied to student-generated data.

Algorithm 1 An algorithm with caption

Input: Teacher policy μ, Student policy π, Teacher sample probability p
Return: Parameters of student policy θ
Collect N_μ teacher transitions $\mathcal{B}_\mu \leftarrow \{\tau_{t_{a_t \sim \mu}} = (s_t^\mu, a_t, R_t, \mu(a_t|s_t^\mu))\}$
for epoch $= 1$ to K **do**
  while $\mathcal{B}_\pi \neq \emptyset$ **do**
    $r \sim \mathcal{U}(0, 1)$
    if $r > p$ **then**
      Sample n transitions $\tau \leftarrow \tau_t \sim \mathcal{B}_\pi$ without replacement
      $\theta \leftarrow \text{PPOUpdate}(\tau)$
    else
      Sample n transitions $\tau \leftarrow \tau_t \sim \mathcal{B}_\mu$ without replacement
      $\theta \leftarrow \text{TADPOUpdate}(\tau)$
    end if
  end while
  Reinitialize B_μ and B_π
end for

3.4.1 Partial Parameter Updates

Importantly, TADPO updates are applied only to the actor and encoder modules of the student. The critic, trained only on on-policy rollouts, provides value estimates consistent with the student's state visitation. This separation ensures that $\hat{\Delta}_t$ reflects the true improvement from student policy perspective.

3.5 Discussion

By selectively distilling advantageous teacher actions, TADPO combines the exploration stability of PPO with the sample efficiency of demonstration-guided learning. It avoids the brittleness of fully offline imitation learning, and mitigates the instability of off-policy RL algorithms like SAC in visual control. Empirical results (Chapter 6) demonstrate that TADPO enables superior long-horizon performance and learning efficiency on challenging off-road driving benchmarks.

Chapter 4

End-to-End Off-Road Autonomy

This chapter presents our hierarchical architecture for achieving robust and scalable end-to-end off-road autonomous navigation. We describe the integration of global planning with learned policies, outlining both the training and deployment phases of our system, and detailing the reward structure and observation modalities employed throughout.

4.1 Hierarchical Autonomy Architecture

We employ a hierarchical architecture to achieve end-to-end off-road autonomy. Given a final goal $\mathbf{p}_g$, a global planner (implemented using A*) generates sparse waypoints utilizing a coarse global map. These waypoints are tracked by an RL controller trained using TADPO. As the globally planned sparse waypoints may be suboptimal and fail to account for all obstacles, the RL controller must incorporate long-horizon planning capabilities to effectively track these waypoints. A detailed exposition of the implementation and code specifics is provided in Appendix **??**.

4.2 Training and Deployment

Training An MPPI controller interpolates sparse waypoints using the cost function from [5] to generate dense waypoints for training the teacher policy. It achieves high

10.45

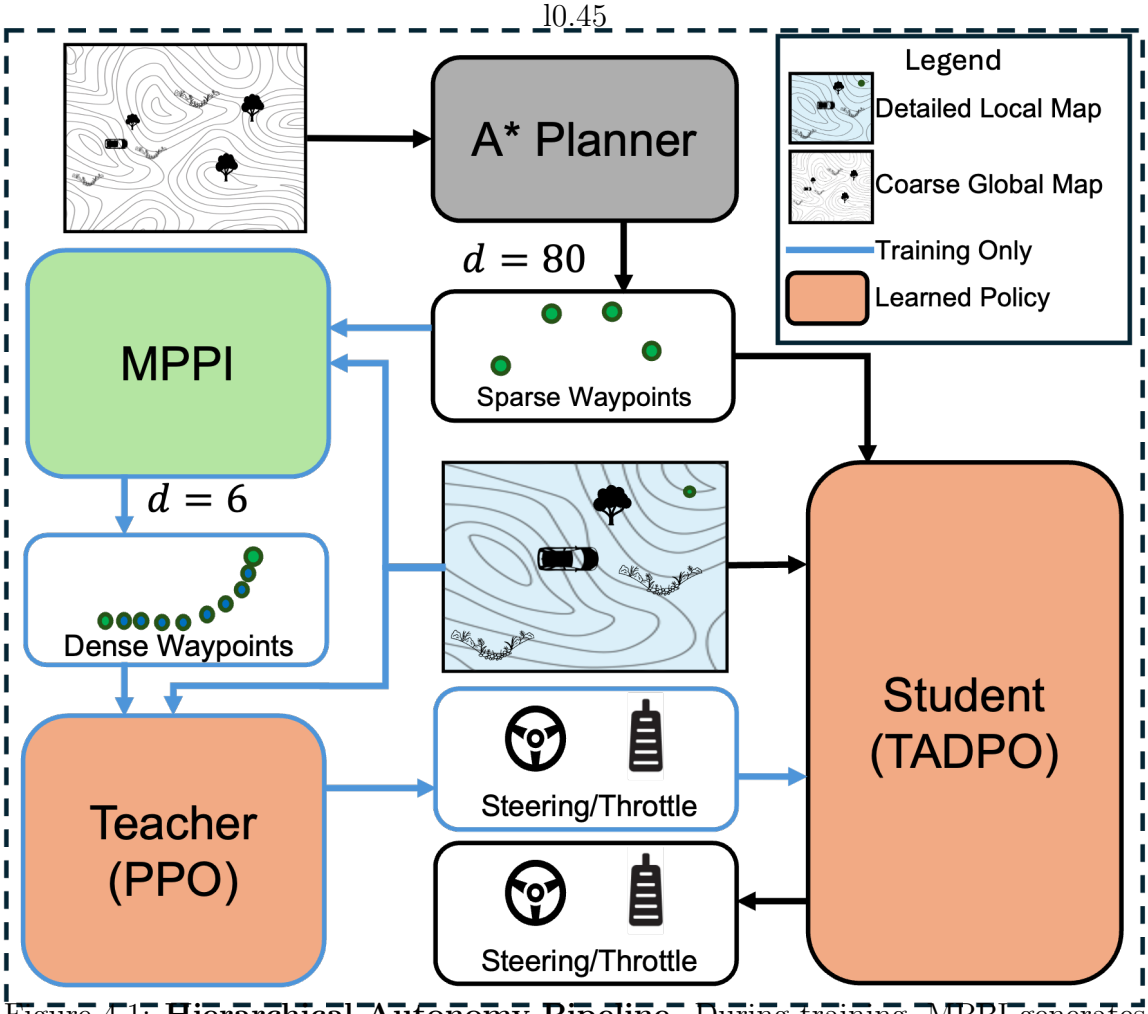

Figure 4.1: **Hierarchical Autonomy Pipeline.** During training, MPPI generates dense waypoints for a teacher policy to follow, providing demonstrations for TADPO, which tracks sparse waypoints. During deployment, TADPO tracks sparse waypoints directly without MPPI.

success rates with sufficient compute but degrades with real-time constraint, as shown in Table 6.1. Appendix **??** offers more details about the MPPI controller.

The PPO teacher policy μ is trained using dense MPPI waypoints. The student policy π_θ is then trained to distill the teacher behavior via the TADPO training procedure in **??** while operating solely with sparse waypoints provided by the global planner. This necessitates π_θ to acquire sophisticated planning abilities to traverse complex terrains, such as ditches and obstacles, despite being trained with reduced waypoint granularity.

Deployment During deployment, we employ an A* global planner that generates sparse waypoints for navigation. These waypoints are subsequently tracked by the policy π_θ, establishing an end-to-end framework for off-road autonomous navigation. This hierarchical approach enables efficient long-range navigation while allowing the policy to handle local terrain traversal and obstacle avoidance. More details about the global planner can be found in Appendix **??**.

4.3 Reward Function, Observation and Action Spaces

The reward function used to train both the teacher and student policies consists of the following components:

- **Progress reward** incentivizing goal-directed movement by measuring reduction in distance to the target waypoint.
- **Penalty terms** for collisions and vehicle damage to encourage safe navigation.
- **Jerk penalty** to reduce abrupt accelerations and promote smooth driving.
- **Success reward** upon reaching the designated waypoint.

Observations are a combination of proprioceptive and visual inputs. Proprioceptive state features include normalized velocity, roll, pitch, and encodings of the current and upcoming waypoints—dense for teachers and sparse for students. Visual inputs comprise top-down and forward camera views over a stack of three historical frames. Teacher policies receive high-resolution local top-down maps, while student policies

are provided wider-area but lower-resolution observations to enhance long-range awareness.

The policies directly output continuous control actions for throttle and steering.

4.4 Rewards

The reward function is designed to encourage progress towards the desired waypoint at t, while penalizing collisions, excessive jerk, and vehicle damage. Additionally, a success reward is granted upon reaching the final waypoint.

1. Progress: $c_1 * (||\vec{p_{t-1}} - \vec{w_{i,t}}|| - ||\vec{p_t} - \vec{w_{i,t}}||)$
2. Collision: $\begin{cases} c_2 & \text{if } > \\ 0 & \text{otherwise} \end{cases}$
3. Damage: $c_3 * \texttt{dam}$
4. Jerk: $c_4 * (||a_t - a_{t-1}||/dt)$
5. Success: $\begin{cases} c_5 & \text{if } ||\vec{p_{t-1}} - \vec{w_{i,t}}|| < w \\ 0 & \text{otherwise} \end{cases}$

where c_i for $i \in \{1, \ldots, 5\}$ are scaling factors for the rewards, with values of 1, -2, -1, -0.003, and 1, respectively. The progress reward reflects the distance the vehicle travels toward the goal, with the maximum reward between two waypoints being equal to the distance between them.

These rewards are significantly sparse for exploration in the off-road navigation problem that involve navigating diverse terrains and obstacles.

Chapter 5

Experimental Setup and Simulation

This chapter details the simulation platform, training and evaluation environments, dataset generation methodology, and performance metrics used to benchmark the proposed learning-based policies. Our goal is to build an evaluation framework that realistically reflects the challenges of off-road autonomous navigation while enabling scalable learning through simulation.

5.1 Simulation Platform

We use `BeamNG.tech` [2] as our high-fidelity simulation environment. BeamNG offers advanced physics-based vehicle dynamics, soft-body damage modeling, and detailed terrain rendering, making it well-suited for testing and training autonomous driving algorithms in off-road scenarios.

We deploy a customized version of the `etk800` vehicle model for all experiments. The simulated vehicle is a mid-size car with the following physical specifications:

- Length: 4.7 meters
- Width: 2.0 meters
- Height: 1.4 meters

It features a simulated internal combustion engine, automatic transmission, and an enforced speed limit of $30,\text{m/s}$ for safety and realism in navigation tasks. Figure **??** illustrates the vehicle operating in various terrains within the simulator.

5.2 Terrain and Task Setup

All tasks are situated in a large, desert-like off-road map with natural elevation variations, soft sand patches, rocky outcrops, and synthetic obstacles. We define multiple start-goal configurations and construct navigation tasks using waypoint-based guidance.

5.2.1 Sparse and Dense Waypoints

We define navigation paths using two types of waypoints:

- **Sparse Waypoints** (80m80m apart): Computed using A* planning over a coarse elevation and cost map.
- **Dense Waypoints** (6m6m apart): Generated via an MPPI controller that incorporates semantic segmentation and depth information, filling in traversable paths between sparse points.

These dense waypoint sequences serve as high-quality reference trajectories for both learning and demonstration purposes.

5.3 Datasets

5.3.1 Training and Demonstration Data

Expert demonstration data is generated by executing the MPPI planner to track dense waypoint paths. These demonstrations are used to train the teacher policy, which in turn generates trajectories for student policy training under the TADPO framework.

We construct three terrain categories with distinct characteristics:

- **(i) Obstacle-rich**: Scenarios with dense clutter, including natural objects (rocks, vegetation) and synthetic barriers (trailers, fences).
- **(ii) Extreme slopes**: Sloped and rugged areas featuring sandy cliffs and ditches, testing the controller's stability and maneuvering skill.
- **(iii) Hybrid terrains**: Tasks combining elements of obstacles, slopes, and uneven surfaces to create highly challenging environments.

We collect the following number of expert demonstrations:

- 15 trajectories each for categories (i) and (ii)
- 20 trajectories for category (iii)

5.3.2 Evaluation Dataset

The test dataset is constructed independently, ensuring no overlap with the training scenarios. It includes:

- 8 test trajectories each for (i) and (ii)
- 15 trajectories for (iii)

Each trajectory poses a unique challenge, as illustrated in Figure **??**, and is used to benchmark the generalization performance of learned policies.

5.4 Policy Evaluation Metrics

To evaluate the effectiveness of the learned policies across different terrain conditions, we define three quantitative metrics. These metrics assess success, progress, and efficiency during navigation episodes. For each policy rollout, we assume a vehicle position ptpt and speed vtvt at time step t1,...,Tt1,...,T, goal position pgpg, an acceptance radius racceptraccept, and control period .

`Success Rate (sr):` A binary indicator of whether the vehicle successfully reaches the vicinity of the goal. An episode is considered successful if the final position lies within the specified radius of the goal:

$$\texttt{sr} = \begin{cases} 1 & d_2(\mathbf{p}_T, \mathbf{p}g) < r\text{accept } 0 \\ \text{otherwise} & \end{cases} \tag{5.1}$$

`Completion Percentage (cp):` Measures the closest distance the agent reaches to the goal throughout the trajectory, normalized as a progress metric:

$$\texttt{cp} = 1 - \min_{t \in \{1,\ldots,T\}} d_2(\mathbf{p}_t, \mathbf{p}_g) \tag{5.2}$$

`Mean Speed (ms):` Captures the average speed of the vehicle during the episode, computed using consecutive position differences:

$$\texttt{ms} = \frac{\sum_{t=1}^{T-1} d_2(\mathbf{p}_t, \mathbf{p}_{t+1})}{\tau \cdot T} \tag{5.3}$$

Here, d2(,)d2(,) denotes the Euclidean distance in the 2D plane. For policies that produce a distribution over actions, we use the mode of the distribution as the selected control command at each timestep.

All methods, including baselines, are evaluated under the same conditions: each is supplied with sparse waypoints from a global A* planner and operates within the same test trajectories. This ensures a fair and consistent comparison across all algorithms.

Chapter 6

Simulation Results

This chapter presents the quantitative evaluation of our proposed method (TADPO) against a comprehensive set of baselines in a high-fidelity off-road simulation environment. We compare TADPO with traditional model predictive control (MPC) methods, reinforcement learning (RL), and imitation learning (IL) algorithms across various terrain challenges. Evaluation is conducted on a held-out test set of trajectories in three categories: *Extreme Slopes*, *Obstacle-rich*, and *Hybrid* terrains.

6.1 Model Predictive Control Baselines

We evaluate several variants of MPC controllers, divided into two groups based on their computational feasibility for real-time deployment.

6.1.1 MPC (Non-Real-time)

In Table 6.1, MPC (Non-real-time) controllers, including CEM, MPPI, and RL+MPPI, are run with long planning horizons (hh) and large sample sizes (NN). These settings assume the simulation can be paused between steps, enabling the controller to compute optimal actions without real-time constraints. All controllers use a simple PID controller to track the planned waypoints.

The results show that with a sufficient planning budget, these methods achieve high success rates and trajectory completion percentages across all terrain types.

6.1.2 MPC (Real-time)

We also evaluate the same MPC controllers under real-time constraints, with significantly reduced hh and NN to meet strict inference time limits. Under these conditions,

performance deteriorates sharply—particularly for CEM due to its iterative optimization strategy.

MPPI performs relatively better, and RL+MPPI provides modest improvements by leveraging a learned terminal value function and state-dependent action distribution to reduce the required planning complexity. Nevertheless, all MPC-based baselines experience a substantial drop in success rates when constrained to real-time operation.

6.2 Reinforcement and Imitation Learning Baselines

We compare TADPO with several standard RL and IL baselines using the same model architecture for fair comparison. All policies are evaluated under real-time inference constraints.

6.2.1 Imitation Learning

DAgger trains a student policy to imitate the teacher via supervised learning over aggregated trajectories. However, it struggles in long-horizon tasks, where early deviations from the expert trajectory lead the policy into unseen, compounding error states—resulting in zero success across all terrains.

6.2.2 On-Policy Reinforcement Learning

PPO is trained using sparse waypoints, with no dense supervision. It performs poorly due to ineffective exploration and an inability to distinguish between terrain features.

PPO + BC augments PPO with behavior cloning via a KL divergence penalty:

$$\mathcal{L}_{\text{KL}} = \mathcal{L}_{\text{PPO}} - \beta \, \text{KL}\left[\pi(a_t \mid s_t^{\pi}), \mu(a_t \mid s_t^{\mu})\right] \tag{6.1}$$

While this provides stronger supervision, the method suffers from instability due to unconstrained updates from the KL term and fails to generalize in long-horizon tasks.

6.2.3 Off-Policy Reinforcement Learning

SAC struggles in our setup as its entropy-regularized policy encourages unnecessary exploration, distracting it from task-relevant behaviors. **SAC + Teacher** introduces teacher demonstrations to pre-fill the replay buffer (with a fixed ratio p=0.5p=0.5), but still underperforms due to SAC's difficulty in handling multi-task dynamics.

IQL is more stable than SAC and demonstrates some promise on challenging terrain like steep slopes. However, it still falls short compared to TADPO, particularly in scenarios that require dynamic obstacle avoidance and generalization across terrains.

6.3 TADPO Performance

Our proposed method TADPO outperforms all other baselines across all terrain categories under real-time constraints. It achieves the highest success rates and completion percentages, while maintaining competitive mean speeds. Notably, TADPO's inference time remains extremely low (ti=0.002ti=0.002s), making it highly suitable for real-world deployment.

Through ablation studies, we determined that a teacher-query probability of $p = 0.5$ and a fixed teacher dropout $\epsilon_\mu = 0.5$ yield the best performance. Detailed ablation results and sensitivity analyses are provided in Appendix **??**.

graphicx

	Controller	Extreme Slopes			Obstacles			Hybrid			ti
		sr	cp	ms	sr	cp	ms	sr	cp	ms	(s)
	MPPI + Teacher	0.88	0.96	5.83	1.00	1.00	5.91	0.94	0.96	5.69	2.02
MPC (Non-real-time)	CEM + PID	0.88	0.96	5.51	1.00	1.00	5.16	0.87	0.94	5.13	3.47
	MPPI + PID	0.88	0.96	5.39	1.00	1.00	5.87	0.87	0.94	5.43	2.02
	RL+MPPI + PID	0.88	0.96	5.26	1.00	1.00	5.88	0.87	0.94	5.40	1.77
MPC (Real-time)	CEM + PID	0.38	0.49	**5.52**	0.25	0.38	5.16	0.27	0.43	5.13	0.13
	MPPI + PID	0.38	0.57	5.43	0.25	0.48	**5.48**	0.27	0.46	5.54	0.12
	RL+MPPI + PID	0.38	0.61	5.32	0.25	0.50	5.46	0.27	0.52	**5.63**	0.12
	TADPO†	**0.75**	**0.87**	4.99	**0.85**	**0.96**	5.26	**0.67**	**0.88**	5.30	**0.002**
RL/IL (Real-time)	DAgger	0.00	0.58	1.96	0.00	0.83	1.62	0.00	0.79	1.68	0.002
	PPO	0.00	0.14	0.38	0.00	0.25	0.49	0.00	0.37	0.40	0.002
	PPO+BC	0.00	0.25	0.94	0.00	0.40	0.78	0.00	0.32	0.84	0.002
	SAC	0.00	0.01	1.71	0.00	0.16	1.64	0.00	0.24	1.61	0.002
	SAC+Teacher	0.00	0.50	1.21	0.00	0.29	1.24	0.00	0.58	1.24	0.002
	IQL	0.25	0.49	4.85	0.13	0.71	5.01	0.07	0.76	5.03	0.002
	TADPO†	**0.75**	**0.87**	**4.99**	**0.85**	**0.96**	**5.26**	**0.67**	**0.88**	**5.30**	**0.002**

Table 6.1: Quantitative comparison of TADPO (†) against baselines. Metrics: `sr` = success rate, `cp` = completion percentage, `ms` = mean speed, `ti` = inference time per control step. Real-time methods operate under strict compute constraints, simulating deployment feasibility.

Chapter 7

Real-World Deployment on Sabrecat

7.1 Introduction

To validate the real-world applicability of our algorithm, we deploy the trained reinforcement learning (RL) policy onto a full-scale off-road robotic platform — **Sabrecat** — located at the National Robotics Engineering Center (NREC). This chapter details the integration of our perception and control pipeline with Sabrecat's onboard systems and demonstrates that policies trained in simulation using our **TADPO (Teacher Action Distillation with Policy Optimization)** framework can be transferred effectively to a real-world off-road driving scenario.

7.2 Hardware and Sensor Setup

Sabrecat is a rugged, off-road autonomous ground vehicle equipped with multiple sensing and actuation systems. For this deployment, we constrain our setup to mimic the minimal sensing configuration used during training. Specifically:

- **RGB Camera:** A forward-facing monocular camera is used as the sole sensor input.
- **Onboard GPU:** Real-time inference is performed on an NVIDIA RTX-class GPU onboard the vehicle.
- **ROS Integration:** All perception and control modules are integrated into the ROS (Robot Operating System) middleware for modular communication.

While the robot is equipped with GPS, LiDAR, and IMUs, these are unused during autonomous control to maintain sensor fidelity with the training setup.

7.3 Perception: Segment Anything Feature Encoder

We leverage the Segment Anything Model (SAM) as a general-purpose visual encoder. The key steps are:

- Extract 256-dimensional visual embeddings from intermediate layers of SAM's ViT-B encoder.
- Use these embeddings as the state representation input to the downstream RL policy.
- **Fine-tune SAM** using a supervised segmentation objective on ground-truth masks obtained from simulation, improving domain alignment.

The frozen encoder ensures consistency between training and deployment, while fine-tuning helps adapt to task-specific semantics such as terrain, drivable regions, and obstacles.

7.4 Policy Training with TADPO

The control policy is trained in simulation using the TADPO algorithm. Briefly, this involves:

- A student policy π_θ trained using Proximal Policy Optimization (PPO).
- Periodic supervised updates from a teacher policy μ, using a buffer of expert demonstrations.
- Only the 256D SAM features from the front-facing camera are used as input; no ground-truth state is exposed to the policy.

Training is performed across procedurally generated off-road environments with varying terrain conditions and obstacle layouts.

7.5 System Deployment Using ROS

The trained policy is deployed on Sabrecat using a modular ROS-based software stack:

- **Camera Node:** Publishes raw RGB images from the front-facing camera.
- **Encoder Node:** Applies SAM to generate 256D features from each image.
- **Policy Node:** Uses the frozen RL policy to compute control commands from embeddings.

Figure 7.1: Sabrecat navigating around an orange barrel from the left.

- **Actuation Node:** Converts high-level actions into steering and throttle commands.

The model is exported using TorchScript for efficient inference. The full pipeline runs at approximately 7 Hz, sufficient for real-time navigation on rough terrain.

7.6 Field Evaluation: Barrel Obstacle Avoidance Test

To assess the effectiveness of the deployed policy, we conduct a series of trials in a controlled off-road environment at NREC.

Test Setup

We design a **Barrel Obstacle Avoidance Test** where large plastic barrels are arranged in a randomized pattern along a 100-meter gravel track. These barrels act as discrete obstacles requiring evasive maneuvers by the robot. The course is

Figure 7.2: Sabrecat performing an evasive maneuver to avoid multiple barrels.

unstructured and includes real-world disturbances such as gravel, vegetation, shadows, and minor elevation changes.

Evaluation Procedure

- The robot starts from a fixed position and is tasked with navigating the course autonomously.
- Only the front-facing RGB camera is used for perception.
- The SAM encoder and policy network run onboard with no external computation.

7.7 Discussion

Our deployment of the TADPO-trained policy on Sabrecat highlights several strengths:

- **Simplicity of Input:** Using only a single RGB camera stream, the agent is able to navigate challenging off-road terrain.
- **Generalization:** Fine-tuned SAM features enable generalization to real-world scenes without requiring explicit re-training.

- **Modular Deployment:** ROS-based system allows seamless integration and real-time operation with onboard computation.

This validates our training paradigm and visual representation strategy, demonstrating the viability of deploying vision-based RL policies on real-world autonomous systems.

We successfully deployed a visually-guided reinforcement learning policy, trained entirely in simulation using the TADPO algorithm, on a real off-road robotic platform — Sabrecat. The use of SAM for visual perception, combined with teacher-guided policy optimization, enabled robust obstacle avoidance in a real-world barrel navigation task. These results mark a significant step toward sim-to-real transfer for autonomous off-road navigation using learned policies.

Chapter 8

Conclusion and Future Work

In this work, we presented TADPO, a novel extension of Proximal Policy Optimization (PPO) that enables hybrid learning from expert demonstrations and on-policy environment interactions. TADPO addresses two core challenges in long-horizon robotic planning: (i) sample inefficiency due to hard exploration, and (ii) multi-skill generalization required for off-road navigation. By incorporating expert rollouts into the training loop and using a teacher-query mechanism with scheduled dropout, TADPO facilitates efficient imitation while preserving the ability to adapt through reinforcement.

We integrate TADPO into a complete off-road autonomy stack by combining it with a sparse global A* planner and semantic MPPI-based local waypoint generation. This results in an end-to-end framework capable of high-speed, real-time navigation in highly unstructured and obstacle-dense terrains. Notably, the final TADPO-based policy is able to mimic the behavior of a computationally intensive MPPI controller, while maintaining comparable success rates and trajectory quality, but with a drastically lower inference time—suitable for deployment in real-world systems.

Extensive evaluations in the BeamNG.tech simulator demonstrate that TADPO outperforms a wide range of baselines, including:

Non-real-time MPC methods such as CEM and MPPI,

Real-time MPC variants with limited computational budgets,

Standard RL methods (e.g., PPO, SAC, IQL), and

Imitation learning techniques (e.g., DAgger, PPO+BC).

While MPC methods exhibit strong performance with sufficient compute, they fail to meet real-time constraints. Meanwhile, traditional RL and IL methods struggle to generalize across diverse terrain features due to inadequate exploration and overfitting to narrow skill distributions. TADPO addresses both issues through its hybrid learning formulation and efficient control architecture.

8.1 Future Work

This work opens up several promising directions for future research:

- **Multimodal Sensor Integration:** Currently, TADPO primarily relies on RGB images and depth data. Future iterations could leverage multimodal sensing, including LiDAR and inertial measurements, to enhance robustness and adaptability in low-visibility or high-speed scenarios.
- **Adaptive Teacher Querying:** We plan to explore dynamic teacher-query scheduling strategies that adaptively decide when to query based on model uncertainty or terrain complexity.
- **Hierarchical Skill Composition:** Future extensions of TADPO could involve hierarchical reinforcement learning to explicitly model and learn reusable off-road driving skills (e.g., obstacle avoidance, slope climbing, narrow path traversal).
- **Robustness under Distribution Shift:** Finally, we aim to evaluate and improve TADPO under out-of-distribution terrain conditions and adversarial perturbations, further increasing its suitability for real-world deployment.

We believe TADPO provides a scalable and practical approach for off-road autonomous driving, and serves as a foundation for future learning-based systems that require real-time, multi-skill control in complex environments.

Code and Resources: The source code, pre-trained models, and demonstration videos for this project are available at: https://github.com/tadpo-algorithm/tadpo.

Bibliography

[1] Miguel Abreu, Nuno Lau, Armando Sousa, and Luis Paulo Reis. Learning low level skills from scratch for humanoid robot soccer using deep reinforcement learning. In *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 1–8, 2019. doi: 10.1109/ICARSC.2019.8733632. 2.2

[2] BeamNG GmbH. BeamNG.tech. URL https://www.beamng.tech/. 5.1

[3] Elliot Chane-Sane, Cordelia Schmid, and Ivan Laptev. Goal-conditioned reinforcement learning with imagined subgoals, 2021. URL https://arxiv.org/abs/2107.00541. 2.4

[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018. URL https://arxiv.org/abs/1801.01290. 2.4

[5] Tyler Han, Alex Liu, Anqi Li, Alex Spitzer, Guanya Shi, and Byron Boots. Model predictive control for aggressive driving over uneven terrain, 2024. URL https://arxiv.org/abs/2311.12284. 1.3.1, 2.3, 4.2

[6] Nicklas Hansen, Xiaolong Wang, and Hao Su. Temporal difference learning for model predictive control, 2022. URL https://arxiv.org/abs/2203.04955. 2.3

[7] Crockett Hensley and Matthew Marshall. Off-road navigation with end-to-end imitation learning for continuously parameterized control. In *SoutheastCon 2022*, pages 591–597, 2022. doi: 10.1109/SoutheastCon48659.2022.9763997. 1.3.2, 2.4

[8] Todd Hester, Matej Vecerik, Olivier Pietquin, Marc Lanctot, Tom Schaul, Bilal Piot, Dan Horgan, John Quan, Andrew Sendonaris, Gabriel Dulac-Arnold, Ian Osband, John Agapiou, Joel Z. Leibo, and Audrunas Gruslys. Deep q-learning from demonstrations, 2017. URL https://arxiv.org/abs/1704.03732. 2.5

[9] David Isele, Reza Rahimi, Akansel Cosgun, Kaushik Subramanian, and Kikuo Fujimura. Navigating occluded intersections with autonomous vehicles using deep reinforcement learning, 2018. URL https://arxiv.org/abs/1705.01196. 2.4

[10] Suresh S. Muknahallipatna James W. Mock. A comparison of ppo, td3 and sac reinforcement algorithms for quadruped walking gait generation, 2023. URL 10.4236/jilsa.2023.151003. 2.5

[11] Dvij Kalaria, Shreya Sharma, Sarthak Bhagat, Haoru Xue, and John M. Dolan. Wroom: An autonomous driving approach for off-road navigation, 2024. URL https://arxiv.org/abs/2404.08855. 1.3.2, 2.4

[12] Bingyi Kang, Zequn Jie, and Jiashi Feng. Policy optimization with demonstrations. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2469–2478. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/kang18a.html. 2.5

[13] Alex Kendall, Jeffrey Hawke, David Janz, Przemyslaw Mazur, Daniele Reda, John-Mark Allen, Vinh-Dieu Lam, Alex Bewley, and Amar Shah. Learning to drive in a day, 2018. URL https://arxiv.org/abs/1807.00412. 2.4

[14] Marin Kobilarov. Cross-entropy motion planning. *The International Journal of Robotics Research*, 31(7):855–871, 2012. doi: 10.1177/0278364912444543. URL https://doi.org/10.1177/0278364912444543. 2.3

[15] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021. URL https://arxiv.org/abs/2110.06169. 2.5

[16] Hojin Lee, Taekyung Kim, Jungwi Mun, and Wonsuk Lee. Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control. *IEEE Robotics and Automation Letters*, 8(11): 7663–7670, November 2023. ISSN 2377-3774. doi: 10.1109/lra.2023.3318190. URL http://dx.doi.org/10.1109/LRA.2023.3318190. 2.3

[17] Gabriele Libardi and Gianni De Fabritiis. Guided exploration with proximal policy optimization using a single demonstration, 2021. URL https://arxiv.org/abs/2007.03328. 2.5

[18] Jesus Bujalance Martin, Raphael Chekroun, and Fabien Moutarde. Learning from demonstrations with sacr2: Soft actor-critic with reward relabeling, 2021. URL https://arxiv.org/abs/2110.14464. 2.5

[19] Soroush Nasiriany, Vitchyr H. Pong, Steven Lin, and Sergey Levine. Planning with goal-conditioned policies, 2019. URL https://arxiv.org/abs/1911.08453. 2.4

[20] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation, 2019. URL https://arxiv.org/abs/1808.00177. 2.2

[21] Zhenghao Peng, Quanyi Li, Chunxiao Liu, and Bolei Zhou. Safe driving via expert guided policy optimization. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 1554–1563. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/peng22a.html. 2.5

[22] Chenyang Qi, Chengfu Wu, Lei Lei, Xiaolu Li, and Peiyan Cong. Uav path planning based on the improved ppo algorithm. In *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE)*, pages 193–199, 2022. doi: 10.1109/ARACE56528.2022.00040. 2.2

[23] Yue Qu, Hongqing Chu, Shuhua Gao, Jun Guan, Haoqi Yan, Liming Xiao, Shengbo Eben Li, and Jingliang Duan. Rl-driven mppi: Accelerating online control laws calculation with offline policy. *IEEE Transactions on Intelligent Vehicles*, 9(2):3605–3616, 2024. doi: 10.1109/TIV.2023.3348134. 2.3

[24] Stephane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning, 2011. URL https://arxiv.org/abs/1011.0686. 2.5

[25] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347. 2.2

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347. 1.4

[27] Qifan Tan, Cheng Qiu, Jing Huang, Yue Yin, Xinyu Zhang, and Huaping Liu. Path tracking control strategy for off-road 4ws4wd vehicle based on robust model predictive control. *Robotics and Autonomous Systems*, 158:104267, 2022. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2022.104267. URL https://www.sciencedirect.com/science/article/pii/S0921889022001567. 1.3.1

[28] Yiquan Wang, Jingguo Wang, Yu Yang, Zhaodong Li, and Xijun Zhao. An end-to-end deep reinforcement learning model based on proximal policy optimization algorithm for autonomous driving of off-road vehicle. In Wenxing Fu, Mancang Gu, and Yifeng Niu, editors, *Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022)*, pages 2692–2704, Singapore, 2023. Springer Nature Singapore. ISBN 978-981-99-0479-2. 2.4

[29] Grady Williams, Andrew Aldrich, and Evangelos Theodorou. Model predictive path integral control using covariance variable importance sampling, 2015. URL https://arxiv.org/abs/1509.01149. 2.3