

TADPO: Reinforcement Learning Goes Off-road

Zhouchonghao Wu^{*1}, Raymond Song^{*1}, Vedant Mundheda^{*1}, Luis E. Navarro-Serment^{1,2},
Christof Schoenborn^{1,2}, and Jeff Schneider¹

Abstract—Off-road autonomous driving poses significant challenges such as navigating unmapped, variable terrain with uncertain and diverse dynamics. Addressing these challenges requires effective long-horizon planning and adaptable control. Reinforcement Learning (RL) offers a promising solution by learning control policies directly from interaction. However, because off-road driving is a long-horizon task with low-signal rewards, standard RL methods are challenging to apply in this setting. We introduce TADPO, a novel policy gradient formulation that extends Proximal Policy Optimization (PPO), leveraging off-policy trajectories for teacher guidance and on-policy trajectories for student exploration. Building on this, we develop a vision-based, end-to-end RL system for high-speed off-road driving, capable of navigating extreme slopes and obstacle-rich terrain. We demonstrate our performance in simulation and, importantly, zero-shot sim-to-real transfer on a full-scale off-road vehicle. To our knowledge, this work represents the first deployment of RL-based policies on a full-scale off-road platform. Source code is available at [this link](#) and video at [this link](#).

I. INTRODUCTION

Autonomous ground vehicles have achieved remarkable progress in structured environments such as highways and urban roads, with detailed maps, high-quality annotations, and where the vehicle-terrain dynamics are easy to model. In contrast, off-road autonomy remains an open challenge. Vehicles must navigate unstructured environments such as sand, gravel, vegetation, and steep slopes, where terrain-vehicle interactions are complex, uncertain, and difficult to model. These conditions require both adaptive control strategies and long-horizon planning without relying on the dense mapping and annotation pipelines available in urban driving. Safe navigation in these settings depends on the vehicle’s ability to perceive and reason about traversable regions in real time, while avoiding obstacles at high speeds.

These challenges motivate the use of Reinforcement Learning (RL), which can directly learn control policies from interaction, bypassing the need for explicit dynamics models, dense maps, or costly labeling. RL offers the potential to leverage large-scale simulation data for training while generalizing to real-world conditions at deployment.

At the same time, applying RL to off-road autonomy is challenging due to low-signal rewards, long-horizon decision-making, complex terrain dynamics, and decision making

*: Equal contributions, order randomized.

¹Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213. {zhouchow, rysong, vmundhed, jeff4}@andrew.cmu.edu

²National Robotics Engineering Center, Robotics Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15201. {lenscmu, cschoenb}@nrec.ri.cmu.edu



Fig. 1: Autonomous vehicle avoiding obstacles (top) and taking corners at speed (bottom) controlled using TADPO-trained end-to-end policies.

in unstructured environments. Such conditions exacerbate difficulties in exploration, and standard RL methods often fail to acquire robust policies without additional guidance.

A promising strategy is teacher-guided RL, where demonstrations or expert actions are distilled into a student policy while continuing to explore beyond the teacher’s demonstrations. This combination enables RL policies to benefit from expert guidance during training while operating without privileged information. Crucially, we find that such a framework also enables strong sim-to-real transfer, allowing policies trained entirely in simulation to be deployed on real full-scale off-road vehicles without fine-tuning. Fig 1 demonstrates our vehicle avoiding obstacles and doing long-distance high-speed control using a policy trained by our method TADPO.

Following this approach, we present three major contributions in this paper:

- Teacher Action Distillation with Policy Optimization (TADPO), a novel extension of Proximal Policy Optimization (PPO) that enables concurrent learning from fixed demonstrations and on-policy interactions to tackle long-horizon planning and hard exploration problems.
- A vision-based, end-to-end RL system for high-speed off-road driving. We demonstrate high performance navigation through extreme slopes and obstacle-rich terrains in simulation.
- The first successful deployment, to the author’s knowledge, on a full-scale off-road vehicle of RL-based policies, demonstrating end-to-end and zero-shot sim-to-real capabilities.

II. RELATED WORK

Off-Road Driving: Multiple works have explored end-to-end RL methods for off-road [1], [2], [3] and on-road

[4], [5], [6] driving. Off-road works are limited and often focus on immediate obstacle avoidance, lacking long-term planning, or are tested in unrealistic simulations. Meanwhile, on-road driving research focuses on the unpredictable behavior of other road users, rather than the variability of terrain encountered in off-road environments. An issue among RL methods presented in these works is their inability to explore efficiently, rendering them ineffective in obstacle-rich environments where simulation is computationally expensive and dynamics are highly complex. As a result, exploration in these scenarios is challenging without external guidance. Another issue common among RL methods is that autonomous driving is often goal-conditioned, requiring longer-horizon planning and reasoning for successful off-road navigation. Some existing works [7], [8] seek to tackle this issue, but in end-to-end navigation tasks that involve image inputs, such methods require sampling image inputs becoming computationally intractable.

Sampling methods like Model Predictive Path Integral (MPPI) [9] and Cross-Entropy Method (CEM) [10] have been applied in the off-road autonomy domain in previous works. MPPI, in particular, has been applied in various domains like UAVs [11], [12] and the off-road autonomy domain [13], [14]. Long Range Navigator (LRN) [15] learns affordance-based intermediate representations from unlabeled ego-centric videos to guide long-horizon off-road planning beyond local metric maps, reducing myopic decisions and improving navigation efficiency. Although the sampling methods vary between these techniques, they all require sampling a large number of trajectories to select a feasible action sequence that minimizes a cost function. While effective for generating control actions in complex, nonlinear systems, the dense sampling they require makes high-quality, real-time operation for long-horizon planning computationally impractical. Some attempts like RL+MPPI [16] and TD-MPC [17] have been made to improve sampling efficiency by learning a state-dependent control action distribution and learning a terminal value function, thereby reducing the required number of samples and planning horizon.

There are also recent works that learn costmaps instead of policies directly from experience. Self-supervised methods predict terrain costs from exteroceptive and proprioceptive signals [18], while SALON adapts online to new environments using speedmaps and uncertainty avoidance [19]. Risk-aware IRL further refines cost functions by accounting for uncertainty and safety [20]. RoadRunner [21] predicts terrain traversability and elevation maps from camera and LiDAR inputs in a self-supervised, low-latency pipeline. In parallel, Zhu et al. [22] use Deep Inverse Reinforcement Learning to learn traversability cost functions from expert demonstrations, incorporating vehicle kinematics for trajectory planning.

Reinforcement Learning: Proximal Policy Optimization (PPO) and Soft Actor-Critic (SAC) are common modern RL techniques for solving complex robotics tasks. PPO [23] is an on-policy RL framework that performs stable policy learning by limiting policy updates through a clipped surrogate objective function. SAC [24] is an off-policy

RL algorithm that optimizes a stochastic policy and value function, enabling efficient and stable learning for continuous control tasks. Both methods have been successfully applied to robotics tasks involving visual inputs and continuous action spaces, including dexterous manipulators, bipedal robots, and unmanned aerial and ground vehicles [25], [26], [27], [28].

Several works augment RL with external guidance through demonstrations or teacher-student frameworks [29], [30]. Approaches such as DAGger [31], offline RL methods like IQL [32], and PPO variants with demonstrations [33] highlight the benefits of combining imitation and reinforcement learning. Despite their successes, these algorithms encounter unique challenges in the proposed off-road driving problem, including navigating diverse terrains and long-horizon planning [34], [35], [36], [37], [38].

Several works use Visual Foundation models such as DinoV2 [39], SAM [40], and SAM2 [41] to extract image features for vision-based RL. These foundational models help bridge the domain gap between simulation and real-world.

III. BACKGROUND

We model the control problem of an off-road autonomous vehicle as a Markov Decision Process (MDP), represented by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where \mathcal{S} is the state space, \mathcal{A} is the action space, $P(s'|s, a)$ is the transition dynamics function, $r : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ is the reward function, and $\gamma \in [0, 1)$ is the discount factor.

Proximal Policy Optimization (PPO) [23] is an on-policy algorithm in the Policy Gradient family. Given a θ -parameterized policy π_θ and a set of trajectories collected by it, PPO employs an actor-critic architecture where the actor learns the policy and the critic estimates the value function to guide policy improvement, maximizing a clipped surrogate objective in Equation (4) to update the policy.

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

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

$$L^{\text{entropy}}(\theta) = \mathbb{E}_t \left[-H[\pi_\theta(\cdot|s_t)] \right] \quad (3)$$

$$L^{\text{PPO}}(\theta) = L^{\text{CLIP}}(\theta) - c_1 L^{\text{VF}}(\theta) + c_2 L^{\text{entropy}}(\theta) \quad (4)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the ratio of the probability of the action a_t under the current policy (π_θ) to the policy used to collect the rollout ($\pi_{\theta_{\text{old}}}$), $\hat{A}_t = \sum_{i=t}^{t+T} (\gamma\lambda)^{i-t} \delta_i$ is the estimate of the advantage, with $\delta_t = R_t + \gamma V_{\pi_{\theta_{\text{old}}}}(s_{t+1}) - V_{\pi_{\theta_{\text{old}}}}(s_t)$, $R_t = \sum_{i=t}^{t+T} \gamma^{i-t} r(s_i, a_i) + \gamma^{T-t+1} V_{\pi_{\theta_{\text{old}}}}(s_{T+1})$ is the discounted return, and T is the number of transitions, $H[\pi_\theta(\cdot|s_t)]$ is the entropy of the action distribution induced by the policy given the state s_t , $V_{\pi_{\theta_{\text{old}}}}(s_t)$ is the expected return of the policy at state s_t , and c_1, c_2 are constants.

The advantage estimator \hat{A}_t quantifies the benefit of executing action a_t over the expected return $V_{\pi_{\theta_{\text{old}}}}(s_t)$ of $\pi_{\theta_{\text{old}}}$ at s_t . This implies that the policy gradient update derived from Equation (4) remains sound only when $V_{\pi_{\theta_{\text{old}}}}$ maintains a faithful approximation of the policy's expected returns, a key motivation behind our formulation in Equation (7).

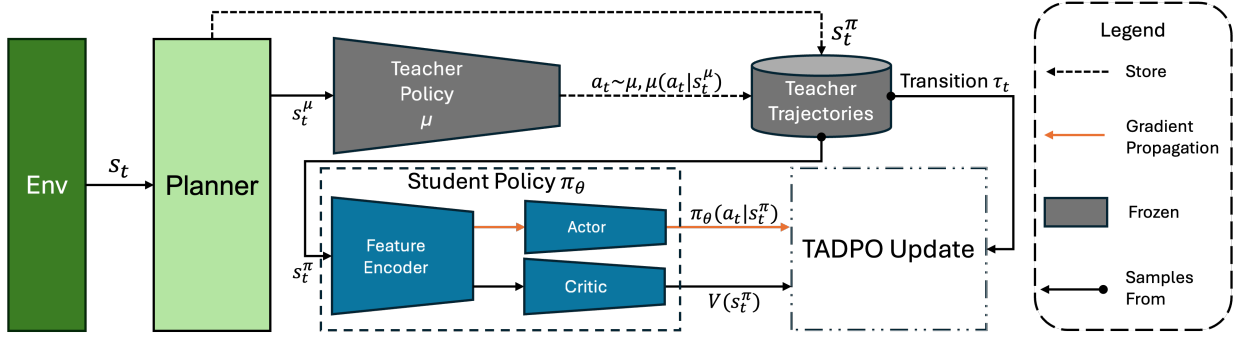


Fig. 2: **Teacher Action Distillation Rollout and Update Process.** The teacher demonstration buffer is frozen while training the student policy. The student policy performs a TADPO update with a probability p solely on the actor and the feature encoder of the policy, using the critic to estimate the advantage of the teacher rollout over the student for any environment state.

PPO struggles with exploration in complex long-horizon tasks, often failing to learn effective policies [33]. Its undirected randomness leads to inefficient sampling, with actions rarely aligned to task objectives. While expert knowledge could help, PPO’s on-policy nature prevents leveraging off-policy data like demonstrations, limiting its use in domains where effective exploration is challenging.

IV. TADPO: TEACHER ACTION DISTILLATION WITH POLICY OPTIMIZATION

We propose a novel method to train a θ -parameterized policy π_θ by extending PPO to incorporate demonstrations from a teacher policy μ . This extension allows π_θ to be trained concurrently using its own on-policy rollouts and demonstrations by the teacher policy.

A. Teacher Action Distillation Policy Gradient

Given a pre-trained teacher policy μ , we define a loss function L^{TAD} in Equation (5) to train a student policy π_θ . This loss is computed solely on teacher rollouts, where actions at each time step t are sampled from the teacher policy, i.e., $a_t \sim \mu$.

$$L^{\text{TAD}}(\theta) = L^\mu(\theta) + c_2 L^{\text{entropy}}(\theta) \quad (5)$$

$$\rho_t(\theta) = \frac{\pi_\theta(a_t | s_t^\pi)}{\mu(a_t | s_t^\mu)} \quad (6)$$

$$\hat{\Delta}_t = R(a_t, s_t) - V_{\pi_{\theta, \text{old}}}(s_t^\pi) \quad (7)$$

$$L^\mu(\theta) = \mathbb{E}_{a_t \sim \mu} \left[\max \left(0, \min(\rho_t(\theta), 1 + \epsilon_\mu) \hat{\Delta}_t \right) \right] \quad (8)$$

where L^{entropy} is defined in Equation (3) and ϵ_μ is a hyperparameter. It is important to note that μ and π_θ can be defined to operate on distinct observation spaces, denoted by s_t^μ and s_t^π respectively, despite being derived from the same underlying environment state s_t . The relaxed assumptions on the relative structure of the teacher and student models allow the teacher to utilize privileged observations or higher-capacity architectures.

In Equation (6), $\rho_t(\theta)$ is defined as the ratio of probability of a_t under π_θ to the probability under μ . This concept is analogous to the probability ratio r_t in PPO. In Equation

(7), $\hat{\Delta}$ estimates the advantage of the achieved return of the teacher rollout at state s_t over the expected return of the student. We observe that applying the policy gradient update from L_μ to the student policy exclusively when $\hat{\Delta} > 0$ (i.e., when the teacher outperforms the student’s expectations) results in stable updates and higher reward attainment by the student.

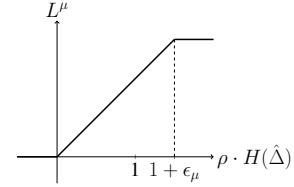


Fig. 3: A single timestep of the teacher distillation loss function L^μ as a function of $\rho \cdot H(\hat{\Delta})$, where $H(\cdot)$ is the Heaviside step function.

In Equation (8), L_μ is clipped as illustrated in Figure 3. This clipping mechanism prevents policy gradient updates when ρ_t exceeds $1 + \epsilon_\mu$, effectively halting updates when the probability ratio of action a_t under π_θ relative to μ surpasses this threshold, indicating that the student policy has already captured the desired behavior.

Thus, L^{TAD} in Equation (5) ensures that the policy gradient propagates only when two conditions are met: (i) the teacher rollout’s return exceeds the student policy’s expected return, and (ii) the student policy’s probability of executing action a_t is not substantially higher than that of the teacher policy. Analogous to PPO, L^{entropy} in Equation (5) modulates the student policy’s exploration.

B. Training Procedure

TADPO involves learning a student policy π_θ through concurrent utilization of teacher demonstrations and student rollouts. Transitions are collected from both the pre-trained teacher policy μ and the student policy π_θ being trained, and stored in separate teacher and student buffers, respectively. With probability p , the algorithm samples transitions from the teacher’s buffer and performs a TADPO update, which likely involves distilling knowledge from the teacher. Otherwise, it samples from the student’s buffer and performs a standard

Algorithm 1 TADPO

Input: Teacher μ , Student π , Sampling prob. p
Return: Student policy params θ
 $\mathcal{B}_\mu \leftarrow N_\mu$ transitions: $\{\tau_{a_t \sim \mu} = (s_t^\mu, a_t, R_t, \mu(a_t | s_t^\mu))\}$
for iter = 1 to I **do**
 $\mathcal{B}_\pi \leftarrow N_\pi$ transitions:
 $\{\tau_{a_t \sim \pi_{\theta_{old}}} = (s_t^\pi, a_t, R_t, \pi_{\theta_{old}}(a_t | s_t^\pi))\}$
for epoch = 1 to K **do**
while $\mathcal{B}_\pi \neq \emptyset$ **do**
 Sample $r \sim \mathcal{U}(0, 1)$
if $r > p$ **then**
 Sample n transitions $\tau \sim \mathcal{B}_\pi$ w/o replacement
 $\theta \leftarrow \text{PPOUpdate}(\tau)$
else
 Sample n transitions $\tau \sim \mathcal{B}_\mu$ w/o replacement
 $\theta \leftarrow \text{TADPOUpdate}(\tau)$
 Reset $\mathcal{B}_\mu, \mathcal{B}_\pi$

PPO update. As shown in Algorithm 1, this alternating process continues for multiple iterations and epochs, allowing the student to learn from both its own experiences and the teacher’s expertise. In our implementation, $\hat{\Delta}_t$ is normalized to have unit standard deviation within each mini-batch. As shown in Figure 2, during the TADPO update, gradient propagation occurs exclusively through the actor and feature encoder components of the student policy π_θ , while the critic remains frozen, ensuring the value function maintains independent state-value estimates based solely on the student’s experiences.

V. END-TO-END OFF-ROAD AUTONOMY

We employ a hierarchical architecture to achieve end-to-end off-road autonomy. Given a final goal \mathbf{p}_g , a global planner 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. Training

An MPPI controller interpolates sparse waypoints using the cost function from [13] to generate dense waypoints for training the teacher policy. 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 Section IV-B while operating solely with sparse waypoints provided by the global planner. Hence, training with sparse waypoints, requires π_θ to learn sophisticated planning capabilities to maneuver through ditches and avoid obstacles. Figure 4 offers an illustration of how the TADPO training pipeline works in the off-road driving task.

B. Reward Function, Observation and Action Spaces, and Model Architecture

The reward function includes progress toward waypoints, penalties for collisions, damage, and jerk, and a success bonus. Observations combine proprioceptive and visual inputs. Proprioception includes normalized speed, roll, pitch, and

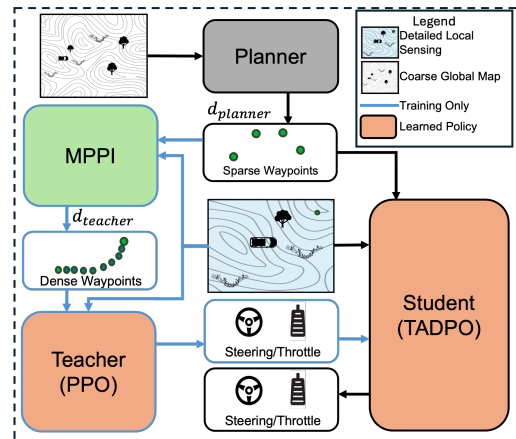


Fig. 4: Hierarchical Autonomy Pipeline. During training, MPPI generates dense waypoints for a teacher policy to follow, providing demonstrations for TADPO, which tracks sparse waypoints. During deployment, TADPO tracks sparse waypoints directly without MPPI. In simulation, $d_{planner} = 80$ and $d_{teacher} = 6$. In real-world deployment, $d_{planner} = 20$ and $d_{teacher} = 4$



Fig. 5: A comparison of the training vehicle in simulation environment and the deployment vehicle in deployed environment. A large embodiment gap can be observed both vehicle dynamics and the terrains.

waypoint encodings, with teachers using dense and students using sparse waypoint plans. Visual inputs include a stack of three frames from top-down and forward views: teachers get high-resolution local maps; students use wider, lower-resolution ones. We use NatureCNN [42] as our feature encoder in simulation. The controller outputs throttle and steering commands. The specific hyperparameters are described in Table I.

VI. TRAINING AND EVALUATION IN SIMULATION

A. Simulator

We use BeamNG.tech [43] as the simulator for training and evaluating our algorithms. BeamNG.tech offers a highly realistic simulation environment, featuring advanced vehicle dynamics, sensor simulation, and damage modeling, allowing us to train and test our algorithms in a realistic environment that closely mirrors real-world conditions. We use etk800 as our vehicle in Simulation. A comparison of the simulation vehicle and the deployment vehicle is shown in Figure 5.

Hyperparameters	Value
Update ratio (ϵ_μ)	0.5
Teacher policy ratio (p)	0.5
Learning Rate	3e-4
Discount Factor (γ)	0.99
Clip Range (ϵ)	0.2
Number of Epochs	20
Mini-batch Size	256
Number of Steps per Update	2048
Value Function Coefficient (λ_v)	0.5
Entropy Coefficient (λ_e)	0.001
Teacher Demonstration Buffer Size	1e5
CNN Vision Backbone	NatureCNN with 256 dim latent space [42]
MLP Head Architecture	[128,64,64]
Vision-backbone	DinoV2-ViT-S/14 (frozen)
Decoder MLP	[768, 4]

TABLE I: Hyperparameters for Teacher and Student Training. Common hyperparameters between simulation training and real-world training are in section 1. Network architecture for simulation and real-world policies are listed in section 2 and 3 respectively.

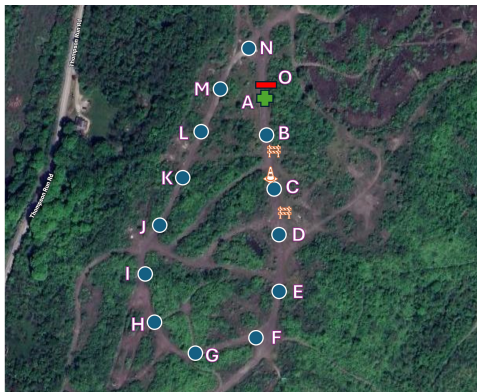


Fig. 6: Waypoints supplied to the planning algorithm to generate the training trajectory. Groups of obstacles are placed along the trajectory, and the terrain is generally made up of uneven tracks. The test track is around 800m in length overall (A-O). We test the vehicle’s dynamic handling over the entire course, but the obstacles are only placed randomly between B-D (around 120m).

B. Training, Demonstration, and Testing Datasets

We train teacher and student policies in a simulated desert environment. Sparse waypoints (80 m apart) are generated using an A* planner over a coarse global map. A fixed set of start-goal pairs is used for teacher training and demonstration trajectories. Dense waypoints (6 m apart) between sparse ones are generated via an MPPI controller using semantic segmentation and depth. A separate trajectory set is used for evaluation. Expert demonstrations use these MPPI-generated dense paths.

Trajectories span varied off-road terrains: (i) obstacle-rich (natural and artificial), (ii) extreme slopes (ditches, cliffs), and (iii) hybrid. We collect 15 teacher demos for (i) and (ii), and 20 for (iii). The evaluation set includes 8 trajectories for (i) and (ii), and 15 for (iii).

C. Simulation Evaluation Metrics

We evaluate policy performance across test trajectories, with the mode of the distribution as the selected action. All baselines utilize an A* planner to generate sparse waypoints. For each episode, we compute the following metrics:

- Success Rate (s_r): 1 if the vehicle reaches the goal within radius r , 0 otherwise.
- Completion Percentage (c_p): Maximum progress toward the goal, normalized by the initial distance.
- Mean Speed (m_s): Average vehicle speed during the episode.

VII. REAL-WORLD VEHICLE EVALUATION

A. Platform

We deploy our policy on a Sabercat (pictured on the right in Figure 5), a 2-ton full-scale off-road vehicle designed to handle challenging terrain and equipped with skid steer control, forward-facing RGB camera, additional stereo cameras, and odometry sensors. Its large size, high cost, and real-world dynamics make it a challenging platform for autonomous navigation, where any collision would result in significant damage. Deploying RL policies on Sabercat allows us to evaluate vision-based obstacle avoidance and traversability reasoning under realistic conditions, including uneven terrain, variable surface types, and dynamic environmental changes, providing a compelling demonstration of zero-shot sim-to-real performance.

B. Adaptations in Training Procedure and Observations Space

In an effort to minimize the Sim2Real gap, we attempt to mimic a similar sensor setup and observation space from our simulator on the real vehicle with a few modifications. Due to the restricted sensor configuration on the real vehicle, a detailed BEV local map is costly to construct with stereo depth. Accordingly, we remove the top down camera from the observation space and decrease the waypoint spacing to 20m and 4m apart for the teacher and student respectively.

As a result, we train our TADPO Policy to output forward velocity and yaw rate given waypoint information and forward-facing image with the same rewards as above. This controller is then deployed on the real vehicle with zero finetuning on real world images or scenarios.

More specifically, for vision features, we follow [19], [44] and use a frozen DinoV2 ViT-S/14[39] as our vision backbone to feed image features to our policy. Additionally, we employ a Probabilistic Road Map (PRM) global planner from Open Motion Planning Library[45] to generate sparse waypoints.

Further, we train our TADPO controller using BeamNG[43] in an off-road forest environment with similar traffic barrels, and then deploy the trained policy zero-shot in a similar off-road forest environment in Pittsburgh, PA.

C. Evaluation Trajectory

To demonstrate the performance of our system in the real world, we set up a test track with two different configurations near Pittsburgh, PA (as shown in Figure 6).

Controller		Extreme Slopes			Obstacles			Hybrid			
		sr	cp	ms	sr	cp	ms	sr	cp	ms	ti
MPPI + Teacher		0.88	0.96	5.83	1.00	1.00	5.91	0.94	0.96	5.69	2.02
MPC (Nonreal-time)	CEM[10] + PID	0.88	0.96	5.51	1.00	1.00	5.16	0.87	0.94	5.13	3.47
	MPPI[9] + PID	0.88	0.96	5.39	1.00	1.00	5.87	0.87	0.94	5.43	2.02
	RL+MPPI[16] + 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[10] + PID	0.38	0.49	5.52	0.25	0.38	5.16	0.27	0.43	5.13	0.13
	MPPI[9] + PID	0.38	0.57	5.43	0.25	0.48	5.48	0.27	0.46	5.54	0.12
	RL+MPPI[16] + 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[31]	0.00	0.58	1.96	0.00	0.83	1.62	0.00	0.79	1.68	0.002
	PPO[23]	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[24]	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[32]	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 II: Our method ([†]) compared with baselines, where *sr* denotes Success Rate, *cp* denotes Completion Percentage, *ms* denotes Mean Speed, and *ti* is the Time of Inference for one control step. (Real-time) denotes allotting a limited compute budget for the main control loop necessary for real-time deployment. MPPI+Teacher is used to provide supervision or demonstrations for RL and IL methods that require it. For RL and IL methods, the model architecture is kept constant to ensure fair comparison.

One configuration is designed to test long-range, high-speed control and evaluates the vehicle’s ability to handle complex terrain dynamics at speed. This requires the vehicle to traverse steep terrain at the traction limit and maintain control in tight turns.

The other configuration is designed to test obstacle avoidance and assesses the control policy’s ability to plan over a long horizon and avoid unmapped obstacles. For obstacles, we selected groups of traffic barrels and placed them between sparse waypoints along the vehicle’s path.

The first configuration runs from A to O in the figure and covers a distance of approximately 800m. The second configuration runs from B to D in the figure and covers a distance of approximately 120m. We tested the second configuration over 12 runs with randomized barrel placements.

D. Real World Evaluation Metrics

We use similar metrics to our simulation metrics except replacing success rate with cross track error.

- Cross Track Error (*cte*): Cross Track Error is the

Configuration	<i>cte</i> (m)	<i>cp</i>	<i>ms</i> (m/s)
Long distance High-speed Control	0.45	1.00	3.41
Obstacle Avoidance	1.50	0.71	2.29

TABLE III: Our method performance using three metrics: mean cross-track error (*cte*), completion percentage (*cp*), and mean speed (*ms*). The test track for the long-distance control configuration has a total length of 800m. Obstacle-avoidance performance is evaluated over 12 runs on a 120m track, with randomized barrel placements.

perpendicular distance from the linear interpolated path between dense waypoints to the vehicle center

- Completion Percentage (*cp*): Same as in Section VI-C.
- Mean Speed (m/s) (*ms*): Same as in Section VI-C.

VIII. RESULTS AND DISCUSSION

A. Simulation: MPC Baselines

In Table II, MPC (Nonreal-time) baselines demonstrate the performance of CEM, MPPI and RL+MPPI controllers with a long planning horizon h and high number of samples N . These controllers calculate the next waypoints while the simulation is paused, enabling them to determine the next action before resuming the simulation. The results show that, given a sufficient number of samples and an extended planning horizon, these controllers achieve similar levels of performance. The cost function and dynamics model implementation are similar to that of [13].

MPC (Real-time) baselines demonstrate the performance of these controllers when operating under real-time constraints with a limited computational budget. In contrast to MPPI, CEM employs a more iterative approach to sampling and evaluating action sequences, which results in a higher computational cost. RL+MPPI improves upon MPPI by incorporating a learned terminal value function and a state-dependent action distribution, reducing the N and h requirement. However, all of these methods experience a substantial decline in performance when executed with a reduced computational budget for real-time inference.

B. Simulation: RL and IL Baselines

We evaluate TADPO against established RL and imitation learning techniques in Table II, adapted as necessary for our

domain. When applicable, all policies are supplied with the same teacher policy μ trained in V-A. The model architecture is held constant across all the policies to ensure a fair comparison. The result is shown in Table II.

Dagger is used to distill the behavior of the teacher through supervised learning. In long-horizon planning tasks, DAgger suffers from compounding errors. As the policy accumulates errors and deviates from expert trajectories, it encounters previously unseen states, leading to significant performance degradation compared to the teacher policy.

The **PPO** policy is trained analogously to the teacher policy μ described in Section V-A, but using only sparse waypoints. As explained in Section III, PPO faces challenges in effective exploration and struggles to differentiate between various terrain types and obstacles, resulting in a suboptimal, overly cautious navigation strategy.

The **PPO + BC** policy aims to distill teacher behavior by adding a KL divergence loss to the PPO loss function as $L^{\text{KL}} = L^{\text{PPO}} - \beta \text{KL}[\pi(a_t|s_t^\pi), \mu(a_t|s_t^\mu)]$, introducing a term that aligns the policy π with the teacher policy μ across all encountered states. While this method provides strong supervision, it encounters challenges similar to DAgger when the student queries the expert from out-of-distribution states. Moreover, the unconstrained updates from the KL divergence term lead to training instability, resulting in convergence to a suboptimal policy.

The **SAC** policy struggles due to entropy maximization, which leads to excessive exploration of irrelevant states and reduced focus on task-specific goals, making it less effective in environments requiring targeted exploration across multiple distinct tasks.

The **SAC + Teacher** policy incorporates teacher demonstrations into the SAC framework by pre-populating a portion of the replay buffer. We maintain consistency with TADPO by using an equivalent buffer size and setting the teacher trajectory ratio to $p = 0.5$. However, SAC’s performance degrades in multi-task problems [46].

The **IQL** policy is trained using teacher demonstrations to reinforce behavior by learning the Q-values associated with those actions. Although IQL demonstrates some success in navigating steep slopes, its overall performance lags behind TADPO, as it is known to encounter difficulties in multi-task problems, such as off-road autonomy [47].

TADPO: The TADPO policy’s success rate (sr) and completion percentage (cp) notably exceed those of other real-time baseline methods. Furthermore, the policy achieves a comparably high mean speed (ms) across all test trajectory sets. Ablation studies show that $\epsilon_\mu = 0.5$ and constant $p = 0.5$ yield the best performance, used in baseline comparisons.

C. Real-World Evaluation

We demonstrate and evaluate the trained student policy on the full-scale Sabercat vehicle for both long-distance high-speed control and obstacle avoidance as shown in Figure 1. As seen in Table III, a policy trained using TADPO achieves a high percentage of obstacle avoidance and minimum cross-track error for long-distance high speed control demonstrating

robust navigation and waypoint tracking without any fine-tuning on the real vehicle. A higher cross-track error when encountering obstacles occurs because the vehicle deviates from the desired path to avoid them and then returns to the path afterward. The policy modulates its speed to safely navigate around obstacles.

IX. CONCLUSIONS

We introduce TADPO, an extension of PPO that enables simultaneous learning from expert demonstrations and on-policy environment interactions to tackle long-horizon planning and hard exploration challenges. By training a TADPO-based policy, we develop an end-to-end off-road autonomy pipeline capable of real-time, long-range navigation in complex, obstacle-rich, and diverse terrains. Our experiments show that TADPO outperforms RL and IL baselines, validating its effectiveness. Our experiments demonstrate strong performance in simulation and, importantly, zero-shot sim-to-real transfer on a full-scale off-road vehicle. This work represents, to our knowledge, the first deployment of end-to-end RL-based policies on a full-scale off-road platform. Our future work includes extending this framework to more diverse terrains.

X. ACKNOWLEDGMENT

This work was supported in part by the U.S. Army Research Office and the U.S. Army Futures Command under Contract No. W519TC-23-C-0030. The authors would like to acknowledge the contributions of Prasanna Kannappan, Anoushka Alavilli, Sam Zieger, Jessica Kasemer, and Daniela Resasco. The authors would like to thank BeamNG GmbH for the academic license for BeamNG.tech.

REFERENCES

- [1] D. Kalaria, S. Sharma, S. Bhagat, H. Xue, and J. M. Dolan, “Wroom: An autonomous driving approach for off-road navigation,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.08855>
- [2] C. Hensley and M. Marshall, “Off-road navigation with end-to-end imitation learning for continuously parameterized control,” in *SoutheastCon 2022*, 2022, pp. 591–597.
- [3] Y. Wang, J. Wang, Y. Yang, Z. Li, and X. Zhao, “An end-to-end deep reinforcement learning model based on proximal policy optimization algorithm for autonomous driving of off-road vehicle,” in *Proceedings of 2022 International Conference on Autonomous Unmanned Systems (ICAUS 2022)*, W. Fu, M. Gu, and Y. Niu, Eds. Singapore: Springer Nature Singapore, 2023, pp. 2692–2704.
- [4] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, “Learning to drive in a day,” 2018. [Online]. Available: <https://arxiv.org/abs/1807.00412>
- [5] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, “Navigating occluded intersections with autonomous vehicles using deep reinforcement learning,” 2018. [Online]. Available: <https://arxiv.org/abs/1705.01196>
- [6] A. R. Fayjie, S. Hossain, D. Oualid, and D.-J. Lee, “Driverless car: Autonomous driving using deep reinforcement learning in urban environment,” in *2018 15th International Conference on Ubiquitous Robots (UR)*, 2018, pp. 896–901.
- [7] S. Nasiriany, V. H. Pong, S. Lin, and S. Levine, “Planning with goal-conditioned policies,” 2019. [Online]. Available: <https://arxiv.org/abs/1911.08453>
- [8] E. Chane-Sane, C. Schmid, and I. Laptev, “Goal-conditioned reinforcement learning with imagined subgoals,” 2021. [Online]. Available: <https://arxiv.org/abs/2107.00541>

- [9] G. Williams, A. Aldrich, and E. Theodorou, "Model predictive path integral control using covariance variable importance sampling," 2015. [Online]. Available: <https://arxiv.org/abs/1509.01149>
- [10] M. Kobilarov, "Cross-entropy motion planning," *The International Journal of Robotics Research*, vol. 31, no. 7, pp. 855–871, 2012. [Online]. Available: <https://doi.org/10.1177/0278364912444543>
- [11] V. Mundheda, D. D. K. and H. Kandath, "Control barrier function-based predictive control for close proximity operation of uavs inside a tunnel," 2023. [Online]. Available: <https://arxiv.org/abs/2303.16177>
- [12] V. Mundheda, K. Mirakhor, R. K. S, H. Kandath, and N. Govindan, "Predictive barrier lyapunov function based control for safe trajectory tracking of an aerial manipulator," 2022. [Online]. Available: <https://arxiv.org/abs/2212.04625>
- [13] T. Han, A. Liu, A. Li, A. Spitzer, G. Shi, and B. Boots, "Model predictive control for aggressive driving over uneven terrain," 2024. [Online]. Available: <https://arxiv.org/abs/2311.12284>
- [14] H. Lee, T. Kim, J. Mun, and W. Lee, "Learning terrain-aware kinodynamic model for autonomous off-road rally driving with model predictive path integral control," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, p. 7663–7670, Nov. 2023. [Online]. Available: <http://dx.doi.org/10.1109/LRA.2023.3318190>
- [15] M. Schmittle, R. Bajjal, N. Hatch, R. Scalise, M. G. Castro, S. Talia, K. Khetarpal, B. Boots, and S. Srinivasa, "Long range navigator (lrn): Extending robot planning horizons beyond metric maps," 2025. [Online]. Available: <https://arxiv.org/abs/2504.13149>
- [16] Y. Qu, H. Chu, S. Gao, J. Guan, H. Yan, L. Xiao, S. E. Li, and J. Duan, "Rl-driven mppi: Accelerating online control laws calculation with offline policy," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 2, pp. 3605–3616, 2024.
- [17] N. Hansen, X. Wang, and H. Su, "Temporal difference learning for model predictive control," 2022. [Online]. Available: <https://arxiv.org/abs/2203.04955>
- [18] M. G. Castro, S. Triest, W. Wang, J. M. Gregory, F. Sanchez, J. G. R. III, and S. Scherer, "How does it feel? self-supervised costmap learning for off-road vehicle traversability," 2023. [Online]. Available: <https://arxiv.org/abs/2209.10788>
- [19] M. Sivaprakasam, S. Triest, C. Ho, S. Aich, J. Lew, I. Adu, W. Wang, and S. Scherer, "Salon: Self-supervised adaptive learning for off-road navigation," 2024. [Online]. Available: <https://arxiv.org/abs/2412.07826>
- [20] S. Triest, M. G. Castro, P. Maheshwari, M. Sivaprakasam, W. Wang, and S. Scherer, "Learning risk-aware costmaps via inverse reinforcement learning for off-road navigation," 2023. [Online]. Available: <https://arxiv.org/abs/2302.00134>
- [21] J. Frey, M. Patel, D. Atha, J. Nubert, D. Fan, A. Agha, C. Padgett, P. Spieler, M. Hutter, and S. Khattak, "Roadrunner – learning traversability estimation for autonomous off-road driving," 2024. [Online]. Available: <https://arxiv.org/abs/2402.19341>
- [22] Z. Zhu, N. Li, R. Sun, D. Xu, and H. Zhao, "Off-road autonomous vehicles traversability analysis and trajectory planning based on deep inverse reinforcement learning," in *2020 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Oct. 2020, p. 971–977. [Online]. Available: <http://dx.doi.org/10.1109/IV47402.2020.9304721>
- [23] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: <https://arxiv.org/abs/1707.06347>
- [24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," 2018. [Online]. Available: <https://arxiv.org/abs/1801.01290>
- [25] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba, "Learning dexterous in-hand manipulation," 2019. [Online]. Available: <https://arxiv.org/abs/1808.00177>
- [26] M. Abreu, N. Lau, A. Sousa, and L. P. 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)*, 2019, pp. 1–8.
- [27] C. Qi, C. Wu, L. Lei, X. Li, and P. Cong, "Uav path planning based on the improved ppo algorithm," in *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE)*, 2022, pp. 193–199.
- [28] H. Taheri, S. R. Hosseini, and M. A. Nekoui, "Deep reinforcement learning with enhanced ppo for safe mobile robot navigation," 2024. [Online]. Available: <https://arxiv.org/abs/2405.16266>
- [29] H. Wang, H. Luo, W. Zhang, and H. Chen, "Cts: Concurrent teacher-student reinforcement learning for legged locomotion," 2024. [Online]. Available: <https://arxiv.org/abs/2405.10830>
- [30] G. Spigler, "Proximal policy distillation," 2025. [Online]. Available: <https://arxiv.org/abs/2407.15134>
- [31] S. Ross, G. J. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," 2011. [Online]. Available: <https://arxiv.org/abs/1011.0686>
- [32] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," 2021. [Online]. Available: <https://arxiv.org/abs/2110.06169>
- [33] G. Libardi and G. D. Fabritiis, "Guided exploration with proximal policy optimization using a single demonstration," 2021. [Online]. Available: <https://arxiv.org/abs/2007.03328>
- [34] Z. Peng, Q. Li, C. Liu, and B. Zhou, "Safe driving via expert guided policy optimization," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1554–1563. [Online]. Available: <https://proceedings.mlr.press/v164/peng22a.html>
- [35] T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, G. Dulac-Arnold, I. Osband, J. Agapiou, J. Z. Leibo, and A. Gruslys, "Deep q-learning from demonstrations," 2017. [Online]. Available: <https://arxiv.org/abs/1704.03732>
- [36] B. Kang, Z. Jie, and J. Feng, "Policy optimization with demonstrations," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2469–2478. [Online]. Available: <https://proceedings.mlr.press/v80/kang18a.html>
- [37] J. B. Martin, R. Chekroun, and F. Moutarde, "Learning from demonstrations with sacr2: Soft actor-critic with reward relabeling," 2021. [Online]. Available: <https://arxiv.org/abs/2110.14464>
- [38] S. S. M. James W. Mock, "A comparison of ppo, td3 and sac reinforcement algorithms for quadruped walking gait generation," 2023. [Online]. Available: <https://arxiv.org/abs/2304.02643>
- [39] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2023.
- [40] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
- [41] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryalı, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: <https://arxiv.org/abs/2408.00714>
- [42] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," 2013. [Online]. Available: <https://arxiv.org/abs/1312.5602>
- [43] BeamNG GmbH, "BeamNG.tech." [Online]. Available: <https://www.beamng.tech/>
- [44] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert, "V-strong: Visual self-supervised traversability learning for off-road navigation," 2024. [Online]. Available: <https://arxiv.org/abs/2312.16016>
- [45] I. A. Şucan, M. Moll, and L. E. Kavraki, "The Open Motion Planning Library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, December 2012, <https://ompl.kavrakilab.org>.
- [46] T. Yu, S. Kumar, A. Gupta, S. Levine, K. Hausman, C. Finn, S. University, U. Berkeley, and R. at Google, "Multi-task reinforcement learning without interference," 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:233305688>
- [47] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," 2022. [Online]. Available: <https://arxiv.org/abs/2205.09991>