Imitative Models for Passenger-Scale Autonomous Off-Road Driving

Nitish Dashora[1*], Sunggoo Jung[2*], Dhruv Shah[1], Valentin Ibars[2], Osher Lerner[1], Chanyoung Jung[2]
Rohan Thakker[2], Nicholas Rhinehart[1], Ali-akbar Agha-mohammadi[2]
[*]Equal contribution [1]UC Berkeley, [2]NASA JPL

***Abstract*— Vision-based control of autonomous vehicles presents major challenges, particularly outside of well structured environments with clear road boundaries and lane markings. Learning control policies from human driving data offers an appealing alternative to classical navigation pipelines: by learning to directly associate observations with actions that avoid obstacles and achieve navigational goals, it is possible to circumvent many of the challenges associated with manually engineering a driving system for unstructured or off-road settings. However, integrating learning-based approaches into robust high-performance control systems presents a major challenge. In this paper, we describe a system for passenger-scale autonomous navigation in off-road environments that combines imitative models with low-level model-predictive control. Although the system learns to control the vehicle directly through perception, it is designed to integrate together learning-based components with constraints and trajectory optimization so as to provide a complete navigational system. Our experiments demonstrate the performance of the system in real-world scenarios over complex off-road terrains, and characterize its potential for improvement with the scaling of data collection and interventions. For a video description, see our link here**

Fig. 1: Our autonomous off-road driving system can traverse challenging trails using imitation learning. Our approach uses a conditional density estimator to predict the most likely expert path given the vehicle's visual observations, and then uses a planner to produce a control sequence to track this path. The vehicle can avoid untraversable bushes, rocks, and other obstacles, and uses an entirely data-driven planning criterion.

I. Introduction

Autonomous navigation with full-scale vehicles in off-road environments like deserts and swamps presents a number of challenges that make conventional geometry-based techniques difficult to apply. On-road navigation can be formulated as a problem of finding collision-free paths that respect known traffic rules and lane constraints, but off-road navigation presents complex physical situations that defy these clean abstractions due to the unstructured and unpredictable nature of the environment. For example, such settings might contain passable obstacles, such as soft vegetation [1], and negative obstacles, such as ravines [2]. In the absence of cues such as roads marked with lanes and traffic signals, off-road vehicles need to reason about the traversability and semantics of their environment in real-time, while also planning a safe and smooth trajectory through it. Inferring traversability in such environments would require algorithms that can reason beyond geometric notions of traversability. For example, a tall bush is traversable for a large vehicle, but a similar-sized tree or rock is not. A successful imitation learning approach should be able to directly extract useful driving patterns from prior experience and infer relevant information like traversability in a *data-driven* manner.

A large amount of research and development in building large-scale off-road autonomy stacks has enabled implementations of off-road driving using a combination of environment mapping, path planning and model-based control [3], [4], [5]. Such a "geometric" stack tends to be very reliable by being conservative about its traversability estimates – through acting overly reliant on modeling the 3D geometry of the scene for path planning – and generally involves lots of hand-engineering (fine-tuning) and heuristics. Imitation learning (IL) [6], [7] is a powerful paradigm that can enable the system to learn relevant cues *directly* from prior experience and improve its performance as we gather more data. However, learning-based systems can be unreliable in the presence of out-of-distribution objects in the scene and difficult to integrate with existing autonomy stacks. An ideal autonomy solution would be able to leverage the ability of IL to learn directly from data, while not giving up the reliable conservativeness of a geometric stack.

In this work, we present **RACER-L4P**, the learning for planning (L4P) method for autonomous off-road driving that is enabled by data-driven learning from Deep Imitative Models (DIM) [8]. We train an IL-based planner to infer a 2D navigation path in top-down coordinates (rather than the conventionally used action predictions) and combine it with a model-based controller. While the particular components that we use to build **RACER-L4P** draw on prior work [8], [9], [10], our system serves as a proof-of-concept to enable passenger-sized autonomous driving over off-road terrain without any manual perceptual feature or cost-function engineering required. The key contribution of this work is a solution to autonomous off-road driving in challenging off-road environments with unmapped terrain. We implement **RACER-L4P** on a passenger-sized off-road driving platform and deploy it for autonomous operation. Our experiments demonstrate that **RACER-L4P** can successfully infer and follow acceptable paths in off-road unstructured environments. Furthermore, our system continually learns from incoming navigational data via collision intervention collection, creating a stack that is constantly improving itself in an iteration-based manner as shown in our performance analysis experiments.

II. Related Work

A. Current Off-Road Autonomy Stacks

Driving in off-road environments requires the vehicle to assess the traversability of the terrain. This can be achieved using geometry-based or appearance-based methods as summarized in Papadakis [11]. Geometry-based methods involve constructing a terrain map [12], [13] from depth measurements obtained via sensors such as LiDAR and stereo cameras. This terrain map is used to generate a traversability cost by performing stability analysis, using features like surface normals and the maximum or minimum height of the terrain, which can be used by motion planning and control algorithms to plan vehicle's actions [14], [15], [16]. Appearance-based methods incorporate higher-level costs through concepts such as semantic segmentation, object detection, and instance segmentation [17], [18], [19], where machine learning techniques are widely applied today. In contrast to these approaches, we describe a system that utilizes imitation learning to directly learn perception from raw observations, bypassing an explicit representation of traversability across a map. We integrate this imitation learning process into a more conventional planning and control pipeline that still allows us to impose constraints and employ state-of-the-art tracking methods. This hybrid method is also robust as it scales with additional data collection whereas purely geometry-based methods cannot easily learn from incoming data automatically.

B. Imitation Learning for Navigation and Driving

Imitation Learning (IL) methods provide a simple way to implicitly model traversability by learning from prior experience. Expert demonstrations can provide a simple way to learn complex relationships between scene features and traversability. For navigating between points, goal-conditioned IL [20] has been used for control in many settings but often does not generalize well causing poor performance when out of distribution. Using IL with external goal direction has been shown to work well in simulation [8] and can be directly integrated with geometric information to improve out-of-distribution performance for off-road navigation [9]. But these algorithms have not been demonstrated on large vehicles in the real world. In [21], authors utilized an inverse reinforcement learning (IRL) framework for predicting traversability, which learns from human demonstrations. The agent was modeled within a Markov decision process (MDP) and a 2-D reward map was learned via maximum-entropy IRL incorporating geometry-based and appearance-based cues. The method was evaluated with a passenger-size off-road vehicle, but IRL was still required to repeatedly solve the MDP during inference which is not suitable for longer range of navigation and high-speed of driving.

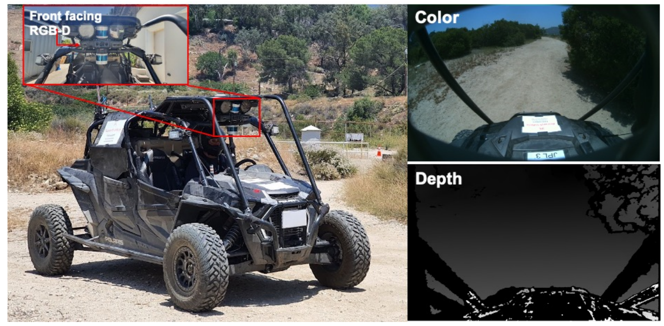

Fig. 2: Left: full vehicle with sensor suite, which consists of a high-resolution stereo camera and depth sensor. Right: RGB and depth images as viewed from the onboard camera during data collection and real-time inference.

Many similar on-road end-to-end methods for navigation have been explored for mapping raw input directly to steering angles [22], [23], [24], but these inherently differ by their problem formulation as they are on-road. Navigation on-road affords the model a largely unimodal distribution of potential paths and furthermore is heavily guided by road segmentation and line markers. Some attempts at using learning for end-to-end driving have been made [25], [26] but these methods do not use any explicit goal-direction, Instead, they do supervised behavior cloning (BC) to predict a most-likely target steer which may not extend to visually complex environments that have multiple valid expert paths. More importantly, these methods only learn to implicitly drive fast while dodging obstacles, not navigate from point A to B. Our proposed method, on the other hand, works at the trajectory-level which enables heuristic-based scoring to follow high-level goals. Additionally, our system is not fully end-to-end, as a low-level MPPI controller is used for operating on a planned short-range trajectory.

More explicit off-road traversability estimation methods that rely on learning exist and can classify full scenes [27] or semantically segment traversable areas [28], [3], but both require extensive annotation which can be hard to get for diverse, unstructured scenes in off-road environments. We propose a method for driving by implicitly learning traversability using unlabeled data.

III. Summary of the **RACER-L4P** System

Autonomous driving in off-road settings requires a complete navigational system that can process visual observations, plan paths that satisfy user constraints, and execute those paths with high-performance control methods. In this paper, we will describe the design of our learning-enabled system for addressing this challenge in the context of controlling passenger-sized vehicles. We begin by describing the hardware and software system architecture.

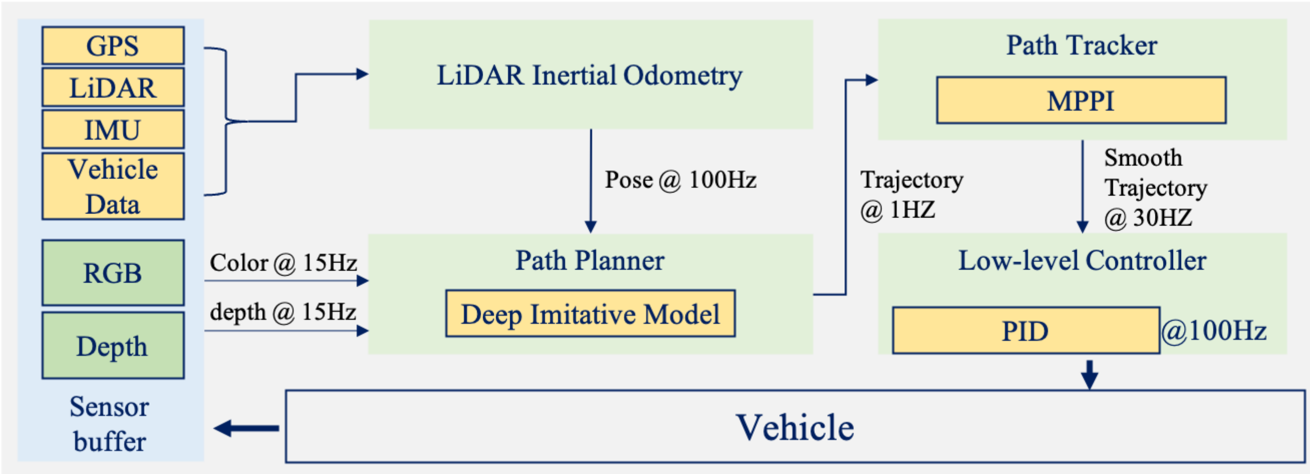

Fig. 3: Framework overview of our system. The system takes LiDAR-inertial odometry for past-trajectory input and RGB & depth images as visual inputs. It outputs the planned short-range path ($\leq$15m) using the DIM policy. The MPPI then smooths the path and generates position, velocity, throttle, and steer control commands to be applied to the low-level PID controller.

A. Hardware Setup and Sensor Suite

We instantiate our **RACER-L4P** system on a Polaris MRZR-X platform (see Fig. 2), which offers the impressive capability to drive on a variety of challenging off-road terrains. The vehicle is equipped with a baseline sensor package, computing resources, and autonomy stack. The sensor package includes an inertial measurement unit (IMU), radar arrays, stereo camera pairs front and backward, and four lidars. The computing resources are comprised of networked GPUs populated with a basic "drive by wire" vehicle management system to execute driving commands (e.g., accelerate, break, turn). Our method uses the color and depth images from the RGB-D sensor package at the front of the vehicle.

B. Software Stack

The **RACER-L4P** software is built based on the NeBula system [10]. In this study, we are focusing on short range planning to achieve resilient navigation at high speeds in off-road settings. A traditional planner requires pre-built/fine-tuned prior information, such as occupancy information, traversability analysis, etc. As opposed to a traditional planner, our architecture requires only a short pose history of the vehicle, as well as an RGB-D image. Fig. 3 illustrates our software architecture. The odometry and RGB-D images are sent to the deep imitative model, which outputs a cost function. The system then uses this cost function to produce a path plan attached to the vehicle frame. For compatibility purposes, we use homogeneous transformation matrix to transform the trajectory to environment's frame. This generated path is provided to a model predictive path integral (MPPI) module to generate smooth control sequences.

IV. Learning Off-Road Navigation with Imitative Models

Our off-road driving software stack consists of two subsystems: imitation-learning based spatial planning and control tracking. The purpose of the former is a data-driven method to infer the *highest-quality path given the perceptual data*, and the latter to infer the *controls that track this path*. For spatial planning, we adopt deep imitative models [8], which learn conditional density estimators of expert-like spatial trajectories, and use them to predict time-profiled trajectories at 10Hz for 4 seconds (i.e., each trajectory is 40 2D points). The planned trajectories are transmitted to the low-level controller to infer control actions. Specifically, we employ a Model Predictive Path Integral Controller (MPPI) [29] to generate command signals which can follow the trajectory precisely for our control sub-system.

A. Tracking with Model-Predictive Path Integral Control

We follow the MPPI framework to determine the controls to send to the robotic system when given a path to follow. MPPI works by sampling trajectories and utilizing a predictive model of the dynamics. After sampling a trajectory from the current distribution, it rolls out the dynamics with the predictive model. The control distribution desired is one which minimizes costs of control, state-dependent custom costs, and any constraints that the user may specify. To get the best distribution, we can treat the optimization problem as one of a KL-divergence minimization between a desired optimal distribution and the current trajectory distribution. This is done via iteratively updating the means of the current distribution with the rolled out dynamics.

B. Spatial Planning with Deep Imitation Learning

Geometric-based planning is a well-studied area in autonomous robotics. In general, geometric pipelines use costmaps built through traversability assessment using features based on the conventional algorithms such as surface fitting [30] and height based cost map generation [4]. However, rule-based algorithms like these are significantly affected by the quality of the parameters and often require significant computational resources for high-frequency collision checking and feature processing. More importantly, in

Fig. 4: Example intervention scenes used for training. The deployed imitative policy often makes wrong predictions (plans shown in red), requiring a human expert to intervene and provide corrective behavior (shown in cyan). We use this corrective behavior as training data to continually improve our system.

rapidly changing environments, such as in off-road terrain, it is hard to perform robustly with static (no human-in-the-loop) geometric pipelines.

For these reasons, we adopt an approach that derives the planning criterion from expert driving data, rather than human engineering. Our proposed method leverages contextual cues from visual sensor information as well as odometry as context inputs (the latter for improving the smoothness of the trajectories). Using learned planning provides a way to be robust to small changes in terrain structure as so often occurs in off-road driving. Furthermore, it removes the need for careful heuristic construction while learning to use computationally efficient features during deployment. Lastly, by incorporating a prior of expert driving experience, the model can correlate visual cues with odometric labels to infer object traversability relatively inexpensively.

We now describe our proposed model's formulation and setup: let $o_t = (x_{\leq t}, i_t)$ denote the sensor information available to the vehicle, where $x_t \in \mathbb{R}^3$ is the vehicle's odometry position, and $i_t \in [0,1]^{H\times W\times 4}$ is an RGB-D image. Let $\tau \in \mathbb{R}^H$ denote a trajectory of potential future positions: $\tau \doteq x_{t+1:t+H}$, and τ_i denote x_{t+i}.

Our goal is to learn a cost-function for planning, C_{learned}. We use this function to select the best τ, τ^*. This trajectory is then given to the MPPI controller for tracking. Our receding-horizon path planner uses Eq. (1):

$$\tau^* = \text{argmin}_\tau C_{\text{learned}}(\tau, o_{\leq t}) \tag{1}$$

C. Learned Imitative Cost

We design this learned cost to be a conditional probability density function of possible future trajectories approximated by a neural architecture. We learn this function, $q(\tau|o_{\leq t})$, by maximum likelihood estimation, following [8]. Once trained, we use the negative log-likelihood of a trajectory in the data-distribution as the learned cost function: $C_{\text{learned}}(\tau, o_{\leq} t) = -\log q(\tau|o_{\leq t})$. While [8] uses gradient-based planning to approximately identify τ^*, we found using a pre-generated library of paths to be more computationally efficient. We generate this library from the centroids of k-means on the training trajectories (we used $K = 200$).

D. Model Architecture

We design $q(\tau|o_{\leq t})$ to be a conditional autoregressive normalizing flow, which is a universal approximator (it can theoretically model any density function) [31]. A normalizing flow enables exact inference of an arbitrary point in event space, which enables us to evaluate the planning criterion exactly for any candidate trajectory. The model uses a learned function, f_{enc} to encode i_t and combines it with x_t via another learned function, g_{enc}. This creates a contextual vector, $z \in \mathbb{R}^d$ where d represents the contextual vector dimensionality. More precisely, we have $z = g_{enc}(f_{enc}(i_t), x_t)$ where f_{enc} is MobileNet-v2 [32] and g_{enc} is a Multilayer Perceptron. Given z and a base distribution sample $x \in \mathbb{R}^H$, we can generate a sample in the data-distribution, $\tau = F(x; o_{\leq t})$, $\tau \in \mathbb{R}^H$. Sampling from the model enables one to inspect the model's predictions for expert-like trajectories, although it is not used to plan (the model's probability density function is used to plan).

E. Distribution Shift

A well-known issue with offline imitation learning is the compounding errors problem [33]. More specifically, when an imitative model (or policy) that scores well on offline metrics is deployed, the visitation distribution can generally diverge from the training distribution – the deployment of the model violates a key supervised learning assumption of IID data. As this occurs, errors may begin building up and cause catastrophic failure due to poor performance of the function approximator when out of distribution. We address this problem by employing DAgger to improve deployment performance [33]: after initial training, we deployed the model and collected corrective maneuvers after interventions, then we combined the original training dataset with the interventions captured during deployment and retrained the model for redeployment. This DAgger intervention loop was then repeated multiple times. Example visualizations of this process are shown in Figure 4.

F. Implementation Details

The data pre-processing pipeline involved subsampling the expert driver data stream to create discrete input data examples. Every training sample contained a certain length of past trajectory, $x_{t-p:t}$, an expert demonstration at time t in a trajectory level, $x_{t:t+h}$, and the current RGB-D information from the front facing camera, $i_t \in [0,1]^{H\times W\times 4}$. For the real-time inference, we used a pre-generated trajectory library with 200 elements. During every planning step, we found the minimum cost trajectory from the library by minimizing the

Fig. 5: Satellite view of the evaluation environment, located at the North Arroyo Seco Trail in Pasadena, California, USA.

planning criterion based on Eq. (1). Our model was trained with the Pytorch framework on a training station with two GTX 1080 GPUs and deployed with four onboard GTX 3080 GPUs.

The output of the short range planner is fed into the high-level controller to generate the control commands. To this end, we use the Model Predictive Path Integral (MPPI) Control framework [29]. Note that we used an augmented bicycle model with extra terms that include brake, throttle, and steering delays. Parameters are carefully determined via vehicle geometric modeling and the individual field testing.

In all experiments, the software stack was executed on our **RACER-L4P** system in real-time. In order to run the software stack and monitor the vehicle status, the operator boarded the passenger seat. An external laptop connected by an Ethernet cable was used to communicate remotely with an onboard computer.

V. Evaluation on a Passenger-Sized Off-Road Vehicle

Fig. 2 shows our experimental platform with sensor suites used for data collection and actual hardware testing. Real-time LiDAR-inertial odometry (LIO) [34] is configured as a state estimator using the built-in 3D-LiDAR. This dataset contains trajectories from an expert human driver in a desert-like environment in southern California. An example snapshot of the collection site is shown in Fig. 1 and Fig. 5. Approximately 4 hours of the dataset was collected from the test environment, with 1 hour serving as validation data. The observation data is itemized below.

- Vehicle Pose: Vehicle pose is estimated from LIO[34] which is part of our NeBula[10] system. The estimated current pose measurements update at 10 Hz and contain 3D pose (position vector and quaternion orientation).
- Color image: High-resolution color images were recorded from the onboard forward-facing stereo camera at 15 Hz.
- Depth image: Gray-scale depth images were also recorded from the forward-facing depth camera at 15 Hz

The raw color and depth images are time-synced, reshaped, and concatenated into a $[200 \times 320 \times 4]$ array and fed into the learning pipeline with the estimated pose.

A. Main Metrics

Our main metrics are based on the interventions by a human safety driver. When the deployed system starts behaving unsafely (e.g. may soon collide with an obstacle), the human safety driver intervenes to correct the vehicle. We recorded these interventions both as a measure of system performance and to further improve the system. We compute metrics based on the interventions per unit distance and interventions per unit time.

B. Behavior Cloning Baseline

Furthermore, instead of modeling $q(\tau|o_{\leq t})$ as a conditional autoregressive normalizing flow, we also compare to modeling it as a conditional Gaussian distribution in trajectory space, which we refer to as the Behavior Cloning ("BC") model, because a typical BC approach minimizes L2 error with a deterministic prediction, which is equivalent to fitting a Gaussian distribution and always outputting its mean. We expect a Gaussian distribution not to be able to realize the true multi-mode expert trajectory distribution, due to the high degrees of uncertainty generally present in driving, let alone off-road driving.

C. Experimental Results

All experiments were conducted using the **RACER-L4P** vehicle described in Section III on the Arroyo Seco Trail (see Fig. 5). The test site is a trail surrounded by bushes, whose structure is dynamic due to seasonal changes in the desert, and due to maintenance and weeding. The width of the trail is between 2–5m, and the total length of the path used in the experiment is about 600m. For this metric evaluation, the vehicle traversed a total of 1684 m. Driver intervention was calculated during on-policy driving.

a) Online metrics: We first study the performance of on-policy driving across model architecture with one iteration of intervention data. Specifically, we aim to analyze how much of an empirical performance boost we can get by using a conditional flow-based architecture instead of a BC model. Furthermore, we'd like to see how this scales with the inclusion of intervention data. Our results are shown in Table I and show how the flow model performs better after an iteration. As hypothesized, the obstacle rich nature of the trail motivates the usage of a multiple-mode distribution representation, which only the flow model can approximate.

Model	$\frac{\text{Interventions}}{\text{Minute}} \downarrow$	$\frac{\text{Interventions}}{\text{100m}} \downarrow$
BC, post-intervention data inclusion	2.479	2.833
Flow, post-intervention data inclusion	**2.004**	**2.197**

TABLE I: Online metrics of our models. Compared to the BC model, it can be seen that the flow model shows about 1.2 times higher performance. This motivates the usage of the Flow for its better performance across 1 data aggregation iteration.

We also sweep across data modality by running an ablation study across image data type. Our goal is to determine if, with the Flow-based architecture, the depth map is a significant component of the system setup. We do this by

Fig. 6: Example observations from our evaluations, with the vehicle's future path visualized in green. Our system can learn from the training data and expert corrective behavior to learn autonomous trail-following behavior directly from egocentric visual observations.

running identical experiments with RGB data and RGB-D data and show our results in Table II. Evidently, the inclusion of depth before any iterations seems to help performance. As seen in Figure 2, the depth map provides a reasonable obstacle map which provides a robust policy for navigation.

Model	$\frac{\text{Interventions}}{\text{Minute}}\downarrow$	$\frac{\text{Interventions}}{\text{100m}}\downarrow$
RGB-D Flow, pre-intervention data inclusion	**5.179**	**5.561**
RGB Flow, pre-intervention data inclusion	6.5	6.97

TABLE II: Online metrics with and without a concatenated depth map. In the flow model, utilizing a depth map is crucial, and informs how we should run our later dagger iterations.

For our final experiment, we use an RGB-D policy using a conditional-flow estimator as an imitative path-planner integrated with MPPI. We collect 3 DAgger iterations and show our results in Table III. As expected, the final iteration model showed improved performance by about 3 times.

Model Iteration	$\frac{\text{Interventions}}{\text{Minute}}\downarrow$	$\frac{\text{Interventions}}{\text{100m}}\downarrow$
Iteration 0	5.179	5.561
Iteration 1	2.004	2.197
Iteration 2	1.740	1.895
Iteration 3	**1.63**	**1.5**

TABLE III: Final DAgger iteration results are shown above for every iteration. After an iteration, the model was retrained with the interventions and re-evaluated. Inclusion of intervention data improves the model performance significantly and is scalable.

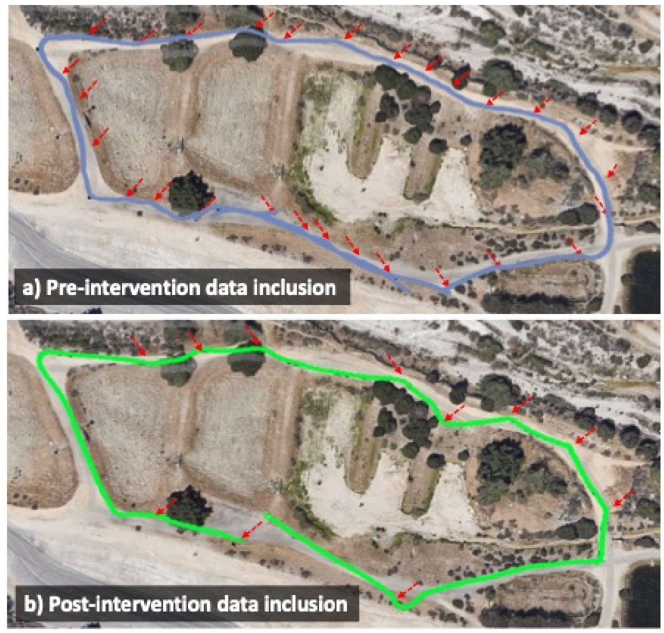

Fig. 7: Satellite view showing the human interventions made to our autonomous imitation-based system driving over the course of a dirt loop: before (top) and after (bottom) performing a round of training with the included interventions. Each arrow indicates a location at which the human intervened.

VI. Discussion

In this paper, we presented a system for learning-based visual navigation in off-road environments for passenger-sized vehicles. Our system combines an imitative model with trajectory optimization and model-predictive control to drive a vehicle through off-road, desert-like environments, and the design enables potential incorporation of other path-based costs, whether learning-based or hand-designed, as might come from a standard geometric pipeline.

The performance of the imitative system and integration with MPPI also displays how robotic systems can take advantage of learning-based methods while also retaining robust control stacks. Our work shows that DIM and MPPI integrate well with each other, eliminating the need to choose either classical or learned methods. By utilizing MPPI for command generation, we remove the need for the imitative model to learn the vehicle's dynamics and control system. Keeping the task for the model simpler is also a straightforward way to reduce the model's sample complexity.

ACKNOWLEDGMENT

The work is partially supported by the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration (80NM0018D0004), and Defense Advanced Research Projects Agency (DARPA).

References

[1] H. Schafer, A. Hach, M. Proetzsch, and K. Berns, "3d obstacle detection and avoidance in vegetated off-road terrain," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 923–928.

[2] Y. Alon, A. Ferencz, and A. Shashua, "Off-road path following using region classification and geometric projection constraints," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, vol. 1. IEEE, 2006, pp. 689–696.

[3] D. Maturana, P.-W. Chou, M. Uenoyama, and S. Scherer, "Real-time semantic mapping for autonomous off-road navigation," in *Proceedings of 11th International Conference on Field and Service Robotics (FSR '17)*, September 2017, pp. 335 – 350.

[4] J. Choi, J. Lee, D. Kim, G. Soprani, P. Cerri, A. Broggi, and K. Yi, "Environment-detection-and-mapping algorithm for autonomous driving in rural or off-road environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 974–982, 2012.

[5] K. Chu, M. Lee, and M. Sunwoo, "Local path planning for off-road autonomous driving with avoidance of static obstacles," *IEEE transactions on intelligent transportation systems*, vol. 13, no. 4, pp. 1599–1616, 2012.

[6] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Advances in neural information processing systems*, vol. 29, 2016.

[7] X. B. Peng, A. Kanazawa, S. Toyer, P. Abbeel, and S. Levine, "Variational discriminator bottleneck: Improving imitation learning, inverse rl, and gans by constraining information flow," *arXiv preprint arXiv:1810.00821*, 2018.

[8] N. Rhinehart, R. McAllister, and S. Levine, "Deep imitative models for flexible inference, planning, and control," in *International Conference on Learning Representations*, 2020. [Online]. Available: https://openreview.net/forum?id=Skl4mRNYDr

[9] N. Dashora, D. Shin, D. Shah, H. Leopold, D. Fan, A. Agha-Mohammadi, N. Rhinehart, and S. Levine, "Hybrid imitative planning with geometric and predictive costs in off-road environments," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4452–4458.

[10] A. Agha, K. Otsu, B. Morrell, D. D. Fan, R. Thakker, A. Santamaria-Navarro, S.-K. Kim, A. Bouman, X. Lei, J. Edlund *et al.*, "Nebula: Quest for robotic autonomy in challenging environments; team costar at the darpa subterranean challenge," *arXiv preprint arXiv:2103.11470*, 2021.

[11] P. Papadakis, "Terrain traversability analysis methods for unmanned ground vehicles: A survey," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 4, pp. 1373–1385, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2013.01.006

[12] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[13] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[14] K. Otsu, G. Matheron, S. Ghosh, O. Toupet, and M. Ono, "Fast approximate clearance evaluation for rovers with articulated suspension systems," *Journal of Field Robotics*, vol. 37, no. 5, pp. 768–785, 2019.

[15] R. Thakker, N. Alatur, D. D. Fan, J. Tordesillas, M. Paton, K. Otsu, O. Toupet, and A.-a. Agha-mohammadi, "Autonomous off-road navigation over extreme terrains with perceptually-challenging conditions," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 161–173.

[16] P. Krüsi, P. Furgale, M. Bosse, and R. Siegwart, "Driving on point clouds: Motion planning, trajectory optimization, and terrain assessment in generic nonplanar environments," *Journal of Field Robotics*, vol. 34, no. 5, pp. 940–984, 2017.

[17] M. J. Procopio, J. Mulligan, and G. Grudic, "Learning terrain segmentation with classifier ensembles for autonomous robot navigation in unstructured environments," *Journal of Field Robotics*, vol. 26, no. 2, pp. 145–175, 2009.

[18] H. Lu, L. Jiang, and A. Zell, "Long range traversable region detection based on superpixels clustering for mobile robots," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2015, pp. 546–552.

[19] P. Ewen, A. Li, Y. Chen, S. Hong, and R. Vasudevan, "These maps are made for walking: Real-time terrain property estimation for mobile robots," *IEEE Robotics and Automation Letters*, 2022.

[20] Y. Ding, C. Florensa, M. Phielipp, and P. Abbeel, "Goal-conditioned imitation learning," *Advances in Neural Information Processing Systems*, 2019. [Online]. Available: http://arxiv.org/abs/1906.05838

[21] Y. Zhang, W. Wang, R. Bonatti, D. Maturana, and S. Scherer, "Integrating kinematics and environment context into deep inverse reinforcement learning for predicting off-road vehicle trajectories," *arXiv preprint arXiv:1810.07225*, 2018.

[22] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *NIPS*, 1988.

[23] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," 2016. [Online]. Available: https://arxiv.org/abs/1604.07316

[24] M. Bojarski, P. Yeres, A. Choromanska, K. Choromanski, B. Firner, L. Jackel, and U. Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," 2017. [Online]. Available: https://arxiv.org/abs/1704.07911

[25] Y. Pan, C.-A. Cheng, K. Saigol, K. Lee, X. Yan, E. A. Theodorou, and B. Boots, "Agile autonomous driving using end-to-end deep imitation learning," *Robotics: Science and Systems XIV*, 2018.

[26] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, "Off-road obstacle avoidance through end-to-end learning," in *NIPS*, 2005.

[27] I. Kostavelis, L. Nalpantidis, and A. Gasteratos, "Supervised traversability learning for robot navigation," in *Towards Autonomous Robotic Systems*, R. Groß, L. Alboul, C. Melhuish, M. Witkowski, T. J. Prescott, and J. Penders, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 289–298.

[28] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox, "Semantic terrain classification for off-road autonomous driving," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=AL4FPs84YdQ

[29] A. A. Grady Williams and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of guidance, control and dynamics (JGCD)*, 2017.

[30] A. Broggi, E. Cardarelli, S. Cattani, and M. Sabbatelli, "Terrain mapping for off-road autonomous ground vehicles using rational b-spline surfaces and stereo vision," in *2013 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2013, pp. 648–653.

[31] G. Papamakarios, E. T. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference." *J. Mach. Learn. Res.*, vol. 22, no. 57, pp. 1–64, 2021.

[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[33] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[34] S. Fakoorian, K. Otsu, S. Khattak, M. Palieri, and A.-a. Agha-mohammadi, "Rose: Robust state estimation via online covariance adaption," *submitted to International Symposium of Robotics Research (ISRR)*, 2022.