Visual Representation Learning for Preference-Aware Path Planning

Kavan Singh Sikand[1], Sadegh Rabiee[1], Adam Uccello[1,2], Xuesu Xiao[1], Garrett Warnell[1,2], Joydeep Biswas[1]

***Abstract*— Autonomous mobile robots deployed in outdoor environments must reason about different types of terrain for both safety (*e.g.,* prefer dirt over mud) and deployer preferences (*e.g.,* prefer dirt path over flower beds). Most existing solutions to this preference-aware path planning problem use semantic segmentation to classify terrain types from camera images, and then ascribe costs to each type. Unfortunately, there are three key limitations of such approaches – they 1) require pre-enumeration of the discrete terrain types, 2) are unable to handle hybrid terrain types (*e.g.,* grassy dirt), and 3) require expensive labelled data to train visual semantic segmentation. We introduce Visual Representation Learning for Preference-Aware Path Planning (VRL-PAP), an alternative approach that overcomes all three limitations: VRL-PAP leverages unlabelled human demonstrations of navigation to autonomously generate triplets for learning visual representations of terrain that are viewpoint invariant and encode terrain types in a continuous representation space. The learned representations are then used along with the same unlabelled human navigation demonstrations to learn a mapping from the representation space to terrain costs. At run time, VRL-PAP maps from images to representations and then representations to costs to perform preference-aware path planning. We present empirical results from challenging outdoor settings that demonstrate VRL-PAP 1) is successfully able to pick paths that reflect demonstrated preferences, 2) is comparable in execution to geometric navigation with a highly detailed manually annotated map (without requiring such annotations), 3) is able to generalize to novel terrain types with minimal additional unlabeled demonstrations.**

I. Introduction and Related Work

Autonomous navigation through unstructured human environments is a well-studied problem in robotics, and has seen a number of different approaches. A common class of autonomous navigation is *geometric navigation*, which plans obstacle-free paths purely via geometric collision-checking. Geometric navigation has been shown to be successful over long-term deployments in indoor settings [1], [2], [3].

However, geometric navigation is unable to reason about paths over different terrain that appears equally valid geometrically (*e.g.,* sidewalk *vs.* dirt *vs.* gravel), but have different costs due to reliability of navigation or social norms; or terrain that appears geometrically impassable but is actually navigable (*e.g.,* tall grass). This shortcoming of geometric navigation has motivated a field of research in the space of *visual navigation*, which uses image data from the mobile robot to reason about the environment while navigating.

[1] The authors are with the Department of Computer Science, The University of Texas at Austin, Austin, TX. Email: {kvsikand@cs, srabiee@cs, adam.uccello@austin, xiao@cs, warnellg@cs, joydeepb@cs }.utexas.edu

[2] Adam Uccello and Garrett Warnell are also with the United States Army Research Laboratory (ARL)

End-to-end learning solutions to the visual navigation problem, which involve using deep neural networks to learn a policy which predicts control commands given raw sensory inputs, have recently become a field of great interest. The supervised approach to this learning problem is to use a reference policy (usually provided by a human) as the training signal indicating the desired behaviour for a given sensory input [4]. To avoid the need to provide training labels for every input, Reinforcement Learning (RL) has gained popularity as a method for learning end-to-end policies in a variety of simulation domains [5] and more recently on real robots [6]. BADGR [7] leverages the available sensing redundancy on a mobile robot to learn behaviour on different types of terrain in a self-supervised manner. By exploring off-policy paths, they are able to learn a planner that ignores geometric obstacles that the robot can safely traverse (*e.g.,* tall grass). In this work, we handle a different case: when geometric information tells us there are *no* obstacles in a given region, but visual information tells us it would be preferable to avoid that region anyway.

While end-to-end approaches are attractive due to their ability to be learned from high-level navigation demonstrations, they have been shown to have significant difficulty generalizing to new environments [8]. To resolve this generalizability issue, a number of approaches start by processing the input to produce some intermediate representation of the environment, such as cost maps, segmentation maps [9], [10], or traversability estimates [11], and then perform planning using that data as an input. For example, GoNet [11] uses Generative Adversarial Networks (GANs) to predict the traversability of an environment given nominal examples of navigation for a mobile robot. Because there are a variety of ways of pre-processing visual information which can be useful for different specific downstream navigation tasks, there has also been work focused on choosing between various intermediate representations, and fusing these outputs together before selecting an appropriate action [12].

Although intermediate data representations such as semantic segmentation and traversability estimation provide helpful generalizability properties, they often require dense manual labelling of training data, a time-intensive process which is required to handle any new terrain type. To ameliorate this shortcoming, inverse reinforcement learning (IRL) with visual semantic segmentation [9] learns the navigation cost associated with each semantic class autonomously from human demonstrations. A similar approach to learning visual navigation is to frame it as a reinforcement learning problem given semantic segmentation of input images [13]. However, these approaches still rely on the outputs of a pre-trained se-

mantic segmentation network, and require manual annotation to extend the semantic segmentation to novel terrain types. Recently, a class of self-supervised "near-to-far" learning techniques have been gaining popularity in the space of traversibility estimation [14] [15]. In these approaches, robots fuse exteroceptive sensor data with proprioceptive sensor data to help classify terrain based on the experiences of the robot while traversing it [15]. These approaches solve the data-efficiency problems of semantic segmentation through self-supervision, but cannot learn to distinguish between terrain classes with similar proprioceptive responses or incorporate preferences based solely on social norms.

Our approach retains the generalizability benefits of using an intermediate representation while removing the dependence on explicit labelling of visual information. In our approach, both the visual representations and the navigation planner can be adapted to a new environment using only unlabeled human-provided demonstrations.

II. Preference-Aware Path Planning

We consider the path planning problem in the context of a state space $\mathcal{S}$, action space $\mathcal{A}$, and deterministic transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Our state space is comprised of states $s = [x, y, \theta, \phi]$, where $[x, y, \theta] \in SE(2)$ denote the robot's position, and $\phi \in \Phi$ denotes the visual appearance of the ground at this location. We define Φ as the space of visual appearances relevant for preference-aware planning. The action space and transition function are defined by the kinodynamic constraints of the robot. Given a start state s_0 and goal G, the local path planning problem is the search for a finite receding horizon sequence of actions, $(a_0, \ldots, a_{N-1}) \in \mathcal{A}^N$ such that the resulting trajectory $\Gamma : i \in \{1, \ldots, N\} \to \mathcal{S}$, which is defined by $\Gamma(i) = \mathcal{T}(\Gamma(i-1), a_{i-1})$, exhibits minimal cost $J(\Gamma)$, $J : \mathcal{S}^N \to \mathbb{R}^+$. Since this is a receding horizon local planning problem, the final state of the optimal solution $\Gamma^*(N)$ may not reach G, but the trajectory must be optimal with respect to the cost such that $\Gamma^* = \arg_\Gamma \min J(\Gamma)$. This paper is primarily concerned with defining J, and uses a previously-established sampling-based local planner to recover Γ. In purely geometric approaches to local planning (i.e., those that consider only geometric obstacles and treat all free space as equal), a common choice for J is

$$J(\Gamma) = J_f(\Gamma(N), G) + J_l(\Gamma) + J_{cl}(\Gamma)\ , \qquad (1)$$

where J_f is the cost based on progress towards G (e.g., $J_f(s, G) = ||G - s||$), J_l is the cost based on the free path length of the trajectory, and J_{cl} the cost based on obstacle clearance [16] along Γ. J_l and J_{cl} using geometric obstacles detected by an on-board LiDAR sensor.

Unlike purely geometric approaches, the path planning method we propose seeks to make preference-aware planning decisions also based on the appearance $\phi \in \Phi$ of the terrain underlying each of the states in the robot's trajectory. For a preference-aware planner that reasons only about distinct semantic classes, Φ would be the set of discrete known semantic classes. In contrast, in our approach $\Phi \subset \mathbb{R}^k$ is a *continuous space* of k-dimensional learned visual representations relevant for preference-aware planning. To incorporate this visual information into the path planning problem, we add an additional term to Eq. 1, redefining J as

$$J(\Gamma) = J_f(\Gamma(N), G) + J_l(\Gamma) + J_{cl}(\Gamma) + J_p(\Gamma)\ , \qquad (2)$$

where $J_p(\Gamma)$ computes a cost based on the appearance of the terrain over which the trajectory Γ traverses. Intuitively, this cost should be large for trajectories that cause the robot to traverse undesirable terrain, and small otherwise.

Instead of specifying J_p manually, we learn it from human demonstrations that implicitly provide information about terrain desirability using a representation learning approach. In the next section, we will discuss this learning problem.

III. Visual Representation-Based Preference Learning

While each robot state $s \in S$ has some true visual appearance $\phi \in \Phi$, the robot does not have a-priori information about it. Instead, the robot observes image patches of the ground $I \in \mathbb{I}$, which are then used to infer ϕ as follows. First, we use an image projection operator $P : S \times S \to \mathbb{I}$, to identify image patches of the ground in one state as seen by another robot state: $P(s_1, s_2)$ returns the image patch I_1 corresponding to the state s_1 while observing it from the state s_2. Note that $P(s_1, s_2)$ needs access to the full image observation of the robot while at pose s_2 – we assume this image to be available, and omit it from the notation for simplicity. This projection operator can be derived from the camera's extrinsic and intrinsic calibration, and the relative positions of the states s_1 and s_2. We then apply a visual representation function f_{vis} to infer the visual appearance information from this image patch. Thus, the appearance ϕ_1 of state s_1 is inferred via the visual observations from a different state s_2 as $\phi_1 = f_{\text{vis}}(P(s_1, s_2))$.

Given an initial state s_0 from which the robot can observe future states along trajectory Γ, we formulate the preference-aware cost $J_p(\Gamma)$ as

$$J_p(\Gamma) = \sum_{t=0,\ldots,N} \gamma^t J_c(\phi_t), \quad \phi_t = f_{\text{vis}}(P(\Gamma(t), s_0)), \quad (3)$$

where $f_{\text{vis}} : \mathbb{I} \to \Phi$ is a visual representation function that maps image patch observations $I \in \mathbb{I}$ to the visual appearance of the ground $\phi \in \Phi$, $J_c : \Phi \to \mathbb{R}^+$ is a cost function that uses these embeddings to produce a real-valued cost, and γ^t is a discount factor to ensure states in the distant future don't have an overbearing impact on the current cost calculation. We propose to learn f_{vis} via *representation learning*, which has recently shown great success at closing the gap between supervised and unsupervised learning for visual tasks such as image recognition [17] and video representation learning [18]. We leverage unlabeled human demonstrations to learn the functions f_{vis} and J_p, as described in Section III-A.

The training data for this learning problem consists of a set of human-provided demonstrations $D = \left\{\Gamma^D_{i=1:M}\right\}$, where each demonstration Γ^D_i consists of a sequence of robot locations and image observations collected by manually

driving a robot from an arbitrary start to a goal location, following a trajectory that encodes human preferences.

A. Visual Representation Learning

The goal of the visual representation function f_{vis} is to map an image patch $I \in \mathbb{I}$ to a low-dimensional representation vector $\phi \in \Phi$ that captures only the salient visual information from the patch relevant to preference-aware planning. Ideally, we would like this representation to exhibit two properties: 1) separability of patches of terrain with different preference values, and 2) invariance to viewpoint changes (that is, the visual appearance ϕ of a given patch of terrain is the same, no matter where it is observed from). We learn this function using a triplet loss function, a form of contrastive loss [19] which requires identification of training triplets that encourage the desired separability and invariance properties. We next discuss the method for triplet identification as well as the loss function used to learn f_{vis}.

Loss Function. We define a loss function for f_{vis} such that the learned result exhibits the above properties when trained over training triplets collected in a self-supervised procedure. In this loss function, we require a triplet of image patches $\langle I^a, I^s, I^d \rangle$, which are referred to as the anchor, similar, and dissimilar patches respectively. Our loss function

$$L_{\text{vis}}(f_{\text{vis}}, \langle I^a, I^s, I^d \rangle) = \max(||f_{\text{vis}}(I^a) - f_{\text{vis}}(I^s)|| - ||f_{\text{vis}}(I^a) - f_{\text{vis}}(I^d)|| + \delta, 0), \quad (4)$$

enforces that in the embedding space $f_{\text{vis}}(I^a)$ is closer to $f_{\text{vis}}(I^s)$ than it is to $f_{\text{vis}}(I^d)$ by at least a fixed margin δ. Given a training dataset of triplets $R_D = \{\langle I_i^a, I_i^s, I_i^d \rangle_{i=1:N}\}$, the visual representation learning problem finds f_{vis}^* which satisfies

$$f_{\text{vis}}^* = \arg_{f_{\text{vis}}} \min \sum_{\langle I_i^a, I_i^s, I_i^d \rangle \in R_D} L_{\text{vis}}(f_{\text{vis}}, \langle I^a, I^s, I^d \rangle). \quad (5)$$

Next, we explain the process of obtaining our training dataset R_D from the demonstrated trajectories D such that this loss function will encourage learning representations which satisfy the desired properties given above.

Similar Patch Extraction. In order to enforce viewpoint invariance, we choose triplets such that I^a and I^s are different views of the *same* location in the real world. Viewpoint invariance requires that for all arbitrary states s, s', s'', the visual representation of the image patch of the observation of state s should be the same as seen from s' and s'':

$$\forall s, s', s'' \in S, \quad f_{\text{vis}}(P(s, s')) = f_{\text{vis}}(P(s, s'')). \quad (6)$$

To extract pairs I^a, I^s which correctly enforce this property from the human demonstrations, we use the following procedure. For three arbitrary time-steps $t_1 < t_2 < t_3$ in a human demonstration trajectory Γ_i^D, the anchor image patch I^a and similar image patch I^s are selected as the image projections of $\Gamma_i^D(t_3)$ from time-steps t_1, t_2 respectively:

$$I^a = P(\Gamma_i^D(t_3), \Gamma_i^D(t_1)), \quad I^s = P(\Gamma_i^D(t_3), \Gamma_i^D(t_2)). \quad (7)$$

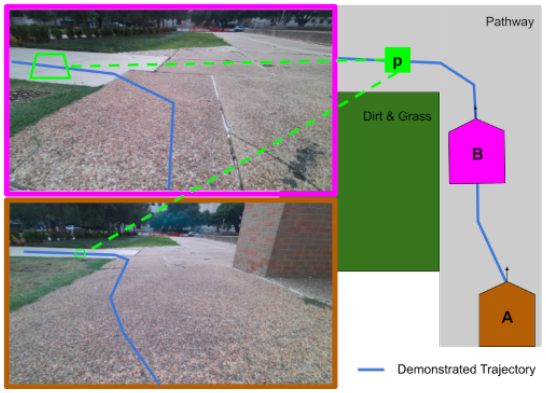
Fig. 1a illustrates this procedure for similar patch extraction.

Dissimilar Patch Extraction. When selecting patches to identify as dissimilar (I^d), we seek to ensure that regions the human demonstrator chose to avoid are distant in the embedding space from regions the demonstration traversed. Rather than ask for human annotation, our approach infers this preference based on the sequence of future demonstration states in each Γ_i^D. For each demonstration trajectory, the robot generates a hypothesized trajectory $\hat{\Gamma}_i^D$ such that they both start and end at the same points ($\hat{\Gamma}_i^D(0) = \Gamma_i^D(0), \hat{\Gamma}_i^D(N) = \Gamma_i^D(N)$) and the length of the hypothesized path[1] is shorter than that of the demonstration: $||\hat{\Gamma}_i^D|| < \Gamma_i^D$. The dissimilar patch I^d is then selected as the image projection of a randomly chosen state on the hypothesized patch that is distant from the demonstration trajectory:

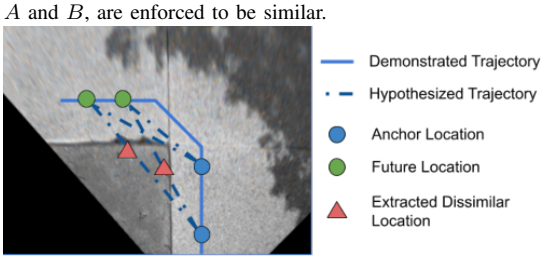
$$I^d = P(\hat{s}, \Gamma_i^D(t_1)) \quad (8)$$
$$\hat{s} \in \hat{\Gamma}_i^D : \min_t ||\hat{s} - \Gamma_i^D(t)|| > T,$$

where T is a tunable threshold. By selecting patches along this hypothesized path which is also far from the demonstrated trajectory, we can infer that this region was explicitly avoided by the human demonstrator. Fig. 1b provides a visualization of this patch selection procedure.

Triplet Selection. The complete training dataset $R_D = \{\langle I_i^a, I_i^s, I_i^d \rangle_{i=1:N}\}$ is generated in a self-supervised manner by repeating the above procedure over exhaustively chosen time-steps t_1, t_2, t_3 for all demonstration trajectories Γ_i^D – since there exists a large number of such time-steps in each demonstration trajectory, we are able to construct a sizable training set with a small number of human demonstrations. Note that this patch extraction strategy depends upon human demonstrations that encode clear avoidance of a particular



(a) Similar Patch Extraction. The visual representations of patches at location p, as observed from the robot at states A and B, are enforced to be similar.



(b) Dissimilar Patch Extraction

Fig. 1: Patch Extraction Procedure

[1]We assume the demonstrations are non-trivial, such that $\hat{\Gamma}_i^D$ always exists and has no geometric obstacles, else the corresponding demonstration trajectory Γ_i^D is discarded.

region of terrain: that is, straight-line demonstrations provide no new data to the system, and sub-optimal navigation through a homogeneous environment can lead to mislabelled training data. Imposing this restriction on the demonstration set is a reasonable trade-off given that the number of demonstrations needed to train the system is quite small.

B. Visual Preference Cost Function

The cost function $J_c : \Phi \to \mathbb{R}^+$ is responsible for taking the visual appearance of a single patch of terrain ϕ, as obtained from f_{vis}, and outputting a real-valued traversal cost, reflecting the cost incurred by travelling over this terrain. These individual patch costs are then combined together in Eq. 3 to contribute to the overall cost of a trajectory Γ.

Loss Function. To train our cost function, we use the same training set that was extracted in Section III-A. Our loss function has a margin δ_c, and can be defined as:

$$L_c(J_c, \phi^p, \phi^n) = \max(J_c(\phi^p) - J_c(\phi^n) + \delta_c, 0). \quad (9)$$

This loss function enforces that $J_c(\phi^n)$ is at least δ_c greater than $J_c(\phi^p)$. We therefore choose ϕ^n such that it is a patch of terrain that should have a high cost relative to ϕ^p. To do this, we find J_c^* which is the cost function J_c such that:

$$J_c^* = \arg_{J_c} \min \sum_{\langle I_i^a, I_i^s, I_i^d \rangle \in R_D} L_c(J_c, f_{\text{vis}}(I^a), f_{\text{vis}}(I^d)) + L_c(J_c, f_{\text{vis}}(I^s), f_{\text{vis}}(I^d)). \quad (10)$$

Here, we use $I^a \cup I^s$ (image patches over which the robot traversed during demonstration) as the patches which generate ϕ^p, and we use I^d (image patches that were explicitly avoided during demonstration) as the patches which generate ϕ^n. By comparing the produced costs in a pairwise fashion, we can enforce a strict ordering among the terrain types that are present. From the demonstrations we are unable to determine the absolute cost of a region, but *are* given relative preference information, and therefore we do not use a regression-based cost function.

IV. Implementation Details

In this section we discuss details of our implementation that allowed the method described above to be deployed in real-time during our experimental evaluation.

Ground-Plane Homography. When working with the robot's visual data, we first apply a homography to transform the images to overhead views, determined by the intrinsics and extrinsics of the robot's camera. Fig. 1b demonstrates the result of this transformation. After this transformation, rectangular image patches of constant size correspond to constant size rectangular regions on the ground plane. In our implementation, we chose a patch size of 40×40 pixels, representing approximately $0.3m^2$ in the real world, comparable to the size of our robot. Patches extracted using this ground-plane homography are the input to f_{vis}.

Local Cost-Map. We retain the costs for each observed patch (the output of J_c) in a local costmap centered on the robot's current position, using the robot's odometry to transform the existing costmap between time-steps. This affords the robot a short-term memory of visual information it has observed, but which is no longer in its view, which helps our implementation handle sharp turns and narrow field-of-view cameras. We recompute J_c for any patches which can be observed by the robot, and we recompute J_p from this costmap for each trajectory at every time step.

Network Structure. In our implementation, the visual representation function f_{vis} takes the form of a neural network with 2 convolutional layers followed by 3 densely-connected layers with nonlinear activation functions, and our representation ϕ is a 6-dimensional vector. The cost function J_c is a small 3-layer Multi-Layer Perceptron (MLP) with a ReLU activation function to prevent negative outputs. These network sizes were experimentally chosen to maximize accuracy while retaining real-time performance on the mobile robot.

Batched Cost Computation. Because our formulation computes costs for each image patch independently, we are able to parallelize the computation of patch costs for each image. The small patch size combined with the compact network structure allows our algorithm to process hundreds of patches per time-step on our robot's GPU (Nvidia GeForce GTX 1050TI), which is enough to process an entire image observation. Our processing of visual information occurs at 20Hz during the planning process; significantly faster than FCHarDNet [20], a segmentation network designed for efficiency in compute-constrained environments, which was only able achieve $\sim 6\text{Hz}$ when running on the same GPU.

V. Experimental Results

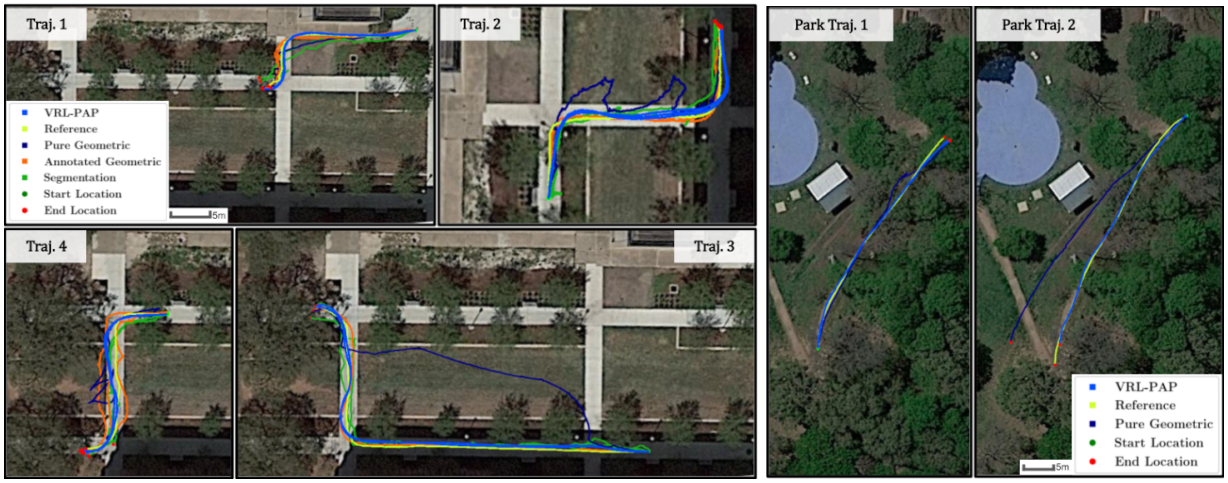
We evaluate **VRL-PAP** in a variety of real-world environments by measuring its 1) *accuracy* at following desired paths compared to other visual and geometric navigation planners; 2) *adaptability* to novel terrain types from limited unlabeled demonstration; and 3) *scalability* to long trajectories in the real world.

A. Experimental Setup

All experiments were performed on a Clearpath Jackal Unmanned Ground Vehicle equipped with a VLP-16 LiDAR, a Microsoft Azure Kinect RGB-D camera, and an Nvidia GeForce GTX 1050TI. The LiDAR is used to perceive geometric obstacles, and the RGB channels of the Kinect camera is used for obtaining visual information.

We compare **VRL-PAP** to four baselines:

- **Reference:** A reference trajectory of the correct preference-aware path provided via joystick by a human operator. These trajectories were not used as part of the training process and are not considered demonstrations.
- **Annotated Geometric:** A geometric planner using a detailed hand-annotated navigation graph of the evaluation environment including desirable paths. This is the primary navigation planner for the Autonomous Mobile Robotics Laboratory at UT Austin, and builds upon the work in [1]. This consists of a global planner, which performs A-star search over the topological map of the environment represented as a graph, and a local planner, which performs sampling-based trajectory roll-out to determine the performed action based on a cost function similar to Eq. 1.



(a) Primary Evaluation Environment Trajectories **(b)** Secondary Evaluation Environment Trajectories

Fig. 2

Planner	Trajectory 1			Trajectory 2			Trajectory 3			Trajectory 4		
	Mean Hausdorff Distance (m)	Off-Path Duration (s)	Intervention Count	Hausdorff Distance (m)	Off-Path Duration (s)	Intervention Count	Hausdorff Distance (m)	Off-Path Duration (s)	Intervention Count	Hausdorff Distance (m)	Off-Path Duration (s)	Intervention Count
Preference Learning	**0.95**	**0**	**0**	**0.93**	**0**	**0**	1.37	**0**	**0**	**1.49**	**0**	**0**
Segmentation	2.40	13	3.5	1.56	2.5	2	2.23	8.5	4	2.66	5.5	3
Annotated Geometric	1.05	1.25	1	1.15	0.25	0.5	**1.32**	**0**	1	2.17	4	**0**
Pure Geometric	> 1.83[2]	24	4	> 4.56[2]	18	5	> 10.02[2]	62	7	> 4.01[2]	15	5

TABLE I: Mean Metrics in Primary Evaluation Environment.

- **Pure Geometric:** The geometric planner described above with a much more coarse global navigation graph.
- **Segmentation:** A state of the art preference-aware planner using semantic segmentation to build a local cost map for planning: The Army Research Laboratory's Autonomy Stack, which uses FCHarDNet [20] for semantic segmentation, trained on the RUGD Vision dataset [21].

In these experiments **VRL-PAP** was trained using 17 human demonstrations in the primary evaluation environment, most of which consisted of navigating a single turn. The robot covered approximately 80m of terrain over the course of about 2 minutes of demonstration data. These demonstrations covered all visually distinct regions of terrain in the evaluation environment, but none of them matched any of the evaluation trajectories. For Section V-C and Section V-E, an additional 4 demonstrations were given in the complex environment, totalling in 40s of data covering 40m.

During deployment, the robot uses Episodic Non-Markov Localization [22], which fuses LiDAR and odometry data to provide robust global localization, allowing for us to evaluate the navigation performance of these systems.

B. Accuracy in Following Desired Paths

We ran repeated trials with each of these navigation methods on four evaluation trajectories, ranging from 10 to 40m in length. Fig. 2a shows these trajectories, which traverse a real-world environment that includes multiple types of valid sidewalk, shadows cast by trees and buildings, and multiple types of undesired terrain including dirt, grass, and shrubs.

We use an undirected Hausdorff distance, which measures the distance from each point in the trajectory to the closest point in the reference trajectory, to quantitatively evaluate the accuracy of each autonomously executed trajectory:

$$H(\Gamma_a, \Gamma_b) = \max(h(\Gamma_a, \Gamma_b), h(\Gamma_b, \Gamma_a)), \qquad (11)$$
$$h = \sum_{a \in \Gamma_a} \min_{b \in \Gamma_b} ||a - b||.$$

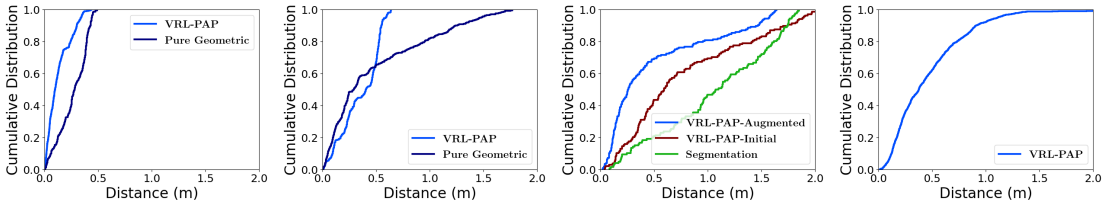
Additionally, we evaluate the duration of time for which the robot was on undesirable terrain type, and the number of operator interventions necessary to prevent the robot from taking unsafe actions (*e.g.,* driving into dense grass or off the side of a concrete pathway).

The results of these experiments are presented in Table I. From these results, we see that **VRL-PAP** performs comparably to **Annotated Geometric** baseline, without access to the hand-made navigation graph for this environment. Further, **VRL-PAP** never needed human intervention, while all of the baseline approaches did. The pre-trained segmentation-based approach struggled to handle terrain that was not in its training dataset (short shrubs and smooth dirt), which motivates the adaptability experiment in Section V-D.

C. Accuracy in Secondary Environment

To investigate the accuracy of **VRL-PAP** in a more complex scenario, we performed evaluation in an unstructured park environment, which included paths that were less clearly delineated than those in the primary evaluation environment. Fig. 2b shows the two trajectories over which we evaluated **VRL-PAP** and the **Pure Geometric** baseline, performing two trials of each. Fig. 3a and Fig. 3b show the cumulative distribution of the distance (CDF) from the reference trajectory when executing **VRL-PAP** and the **Pure Geometric** baseline – in both cases **VRL-PAP** more closely follows the

[2]Error would have been higher; included extensive manual intervention



(a) Park Traj. 1 **(b)** Park Traj. 2 **(c)** Adaptability Experiment **(d)** Extended Deployment

Fig. 3: Cumulative Distribution Functions measuring distance to Reference trajectory

reference trajectory, staying within $0.75m$ at all times.

D. Adaptability to Novel Terrain Types

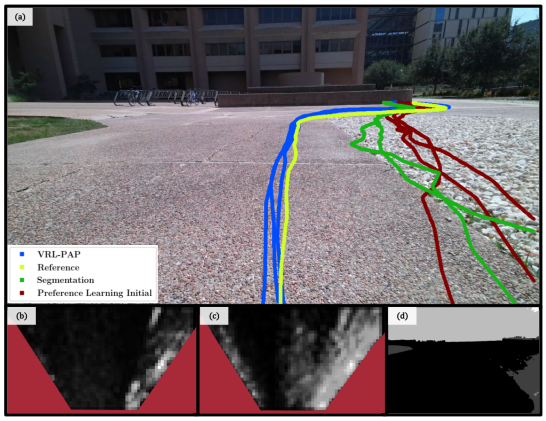


Fig. 4: Comparison of the behavior of path planners in the presence of a novel undesirable type of terrain (gravel). (a) Visualization of the trajectories traversed by each planner. (b) Top-down view of the predicted terrain cost by **VRL-PAP-initial** before being provided any training data including gravel, and (c) by **VRL-PAP-augmented** after re-training given a few short human demonstrations. Darker regions indicate lower navigation cost. Red regions are out of the robot's field of view. (d) Camera-frame view of the predicted cost map by the semantic segmentation approach.

In principle, both semantic segmentation and **VRL-PAP** should be customizable to novel terrain types with sufficient training data. However, a key feature of **VRL-PAP** is that it can adapt to novel terrain types given only unlabeled human demonstrations. In contrast, for a visual semantic segmentation-based approach, adapting to new types of terrain requires collecting labelled segmentation images, which is significantly more onerous.

To evaluate the adaptability of **VRL-PAP**, we first deployed it in an environment with a new class of undesirable terrain. The initial model **VRL-PAP-initial** failed to avoid the terrain type, as did the segmentation-based approach. However, after providing 3 unlabeled human demonstrations of a different trajectory in the new environment, totalling just 27 seconds of driving and covering approximately 25m of terrain, our updated model **VRL-PAP-augmented** was able to successfully avoid the undesirable terrain. Fig. 3c and Fig. 4 show the results of this experiment.

E. Scalability to Extended Deployments

Finally, we provide an example of **VRL-PAP** navigating a long trajectory, demonstrating its ability to stay on desirable paths over a long period of time without human intervention. Fig. 5 shows the evaluation trajectory – it circumnavigates the park environment used in Section V-C. This trajectory covers $440m$ of autonomous navigation, during which the robot was provided 4 sequential navigation goals, and required 0 manual interventions. Fig. 3d shows a CDF of the distance between **VRL-PAP** and the human-provided reference trajectory, showing that it stayed less than $1m$ away from the reference for over 90% of the trajectory. The model used for this environment was trained using the same small set of demonstration trajectories from Section V-C, and was able to generalize well enough to traverse the entire park environment.

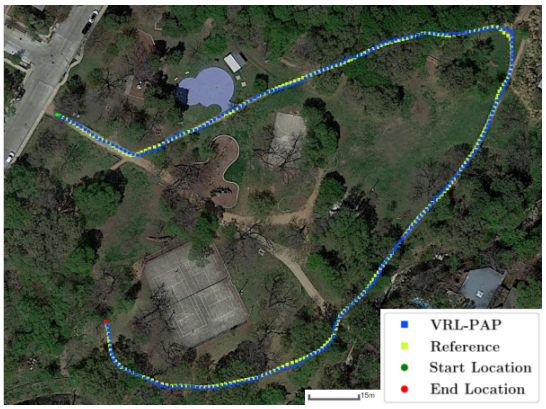


Fig. 5: Extended Evaluation Trajectory

VI. Conclusion

In this work we presented **VRL-PAP**, a method for preference-aware path planning based on visual representations, which is learned from unlabelled human demonstrations. We provided a formulation for this approach which enforces desired properties of viewpoint invariance and separability on the learned visual representations. Finally, we demonstrated this approach's capacity to successfully navigate in a variety of environments and transfer to novel terrain types with no manual annotation of training data.

VII. Acknowledgements

This work has taken place in the Autonomous Mobile Robotics Laboratory (AMRL) at UT Austin. AMRL research is supported in part by NSF (CAREER-2046955, IIS-1954778, SHF-2006404), ARO (W911NF-19-2-0333), DARPA (HR001120C0031), Amazon, JP Morgan, and Northrop Grumman Mission Systems. The views and conclusions contained in this document are those of the authors alone.

REFERENCES

[1] J. Biswas and M. Veloso, "The 1,000-km challenge: Insights and quantitative and qualitative results," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 86–96, 2016.

[2] N. Hawes, C. Burbridge, F. Jovan, L. Kunze, B. Lacerda, L. Mudrova, J. Young, J. Wyatt, D. Hebesberger, T. Kortner *et al.*, "The strands project: Long-term autonomy in everyday environments," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 146–156, 2017.

[3] P. Khandelwal, S. Zhang, J. Sinapov, M. Leonetti, J. Thomason, F. Yang, I. Gori, M. Svetlik, P. Khante, V. Lifschitz *et al.*, "Bwibots: A platform for bridging the gap between ai and human–robot interaction research," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 5-7, pp. 635–659, 2017.

[4] A. Giusti, J. Guzzi, D. C. Cireşan, F.-L. He, J. P. Rodríguez, F. Fontana, M. Faessler, C. Forster, J. Schmidhuber, G. Di Caro *et al.*, "A machine learning approach to visual perception of forest trails for mobile robots," *IEEE Robotics and Automation Letters (RA-L)*, vol. 1, no. 2, pp. 661–667, 2015.

[5] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS)*, 2013.

[6] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with autorl," *IEEE Robotics and Automation Letters (RA-L)*, vol. PP, pp. 1–1, 02 2019.

[7] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 2, pp. 1312–1319, 2021.

[8] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion control for mobile robot navigation using machine learning: a survey," *The International Journal of Robotics Research (IJRR)*, 2020.

[9] M. Wigness, J. G. Rogers, and L. E. Navarro-Serment, "Robot navigation from human demonstration: Learning control behaviors," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1150–1157.

[10] A. Mousavian, A. Toshev, M. Fišer, J. Košecká, A. Wahid, and J. Davidson, "Visual representations for semantic target driven navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8846–8852.

[11] N. Hirose, A. Sadeghian, M. Vázquez, P. Goebel, and S. Savarese, "Gonet: A semi-supervised deep learning approach for traversability estimation," in *International Conference on Intelligent Robots and Systems(IROS)*. IEEE, 2018, pp. 3044–3051. [Online]. Available: https://doi.org/10.1109/IROS.2018.8594031

[12] W. Shen, D. Xu, Y. Zhu, L. Fei-Fei, L. Guibas, and S. Savarese, "Situational fusion of visual representation for visual navigation," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 2881–2890.

[13] J. Bruce, N. Sunderhauf, P. Mirowski, R. Hadsell, and M. Milford, "Learning deployable navigation policies at kilometer scale from a single traversal," in *Proceedings of The 2nd Conference on Robot Learning (CORL)*, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87, 2018, pp. 346–361.

[14] O. Mayuku, B. W. Surgenor, and J. A. Marshall, "A self-supervised near-to-far approach for terrain-adaptive off-road autonomous driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 14 054–14 060.

[15] L. Wellhausen, A. Dosovitskiy, R. Ranftl, K. Walas, C. Cadena, and M. Hutter, "Where should i walk? predicting terrain properties from images via self-supervised learning," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1509–1516, 2019.

[16] A. Richardson and E. Olson, "Iterative path optimization for practical robot planning," 09 2011, pp. 3881–3886.

[17] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML)*. PMLR, 2020, pp. 1597–1607.

[18] R. Qian, T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui, "Spatiotemporal contrastive video representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 6964–6974.

[19] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[20] P. Chao, C.-Y. Kao, Y. Ruan, C.-H. Huang, and Y. Lin, "Hardnet: A low memory traffic network," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3551–3560, 2019.

[21] M. Wigness, S. Eum, J. G. Rogers, D. Han, and H. Kwon, "A rugd dataset for autonomous navigation and visual perception in unstructured outdoor environments," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[22] J. Biswas and M. Veloso, "Episodic non-markov localization: Reasoning about short-term and long-term features," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014, pp. 3969–3974.