PACER: Preference-conditioned All-terrain Costmap Generation

Luisa Mao[1], Garrett Warnell[1,2], Peter Stone[1,3], Joydeep Biswas[1,4]

Abstract—In autonomous robot navigation, terrain cost assignment is typically performed using a semantics-based paradigm in which terrain is first labeled using a pre-trained semantic classifier and costs are then assigned according to a user-defined mapping between label and cost. While this approach is rapidly adaptable to changing user preferences, only preferences over the types of terrain that are already known by the semantic classifier can be expressed. In this paper, we hypothesize that a machine-learning-based alternative to the semantics-based paradigm above will allow for rapid cost assignment adaptation to preferences expressed over *new* terrains at deployment time without the need for additional training. To investigate this hypothesis, we introduce and study PACER, a novel approach to costmap generation that accepts as input a single birds-eye view (BEV) image of the surrounding area along with a user-specified *preference context* and generates a corresponding BEV costmap that aligns with the preference context. Using a staged training procedure leveraging real and synthetic data, we find that PACER is able to adapt to new user preferences at deployment time while also exhibiting better generalization to novel terrains compared to both semantics-based and representation-learning approaches. We release our code and dataset at https://github.com/ut-amrl/PACER_RAL_2025.git

Index Terms—Vision-based navigation, Deep Learning for Visual Perception

I. Introduction and Related Work

ROBUST autonomous navigation in a wide variety of environments is a long-standing goal in robotics. While there has been significant progress in collision-free navigation [1], [2], successful navigation in human environments additionally requires alignment with human preferences, e.g., preferring to cross a busy street at a crosswalk even if doing so results in a longer path [3], [4].

In this paper, we examine how robots can assign terrain costs that align with human preferences for terrain-aware navigation. This specific focus represents a special case of the broader challenge of human preference-aligned navigation [5]. Such alignment is crucial not only for terrain-aware navigation, but also for adherence to constraints like social norms [6], [7]. An alternative to learning to predict navigation *costs* is to directly learn navigation *policies* aligned with human preferences [8]–[10]. We choose to focus on predicting navigation costs due to the ease of integration with existing cost-based navigation planners [11]–[13]. While the focus of
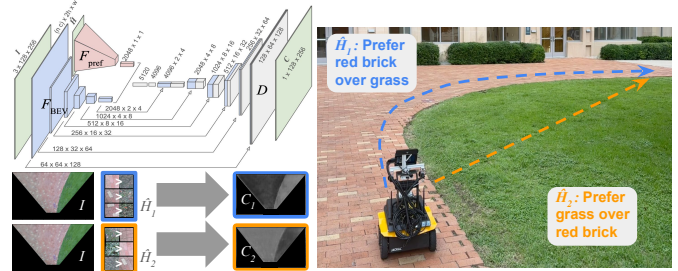
Fig. 1: Given an input image I and a preference context $\hat{H}$ of n ordered pairs of terrain patches where the left terrain is more preferred than the right, PACER generates a costmap consistent with this preference. Changing the preference context leads to changed terrain costs, which results in a different plan aligned to the new operator preference. The paths planned according to the different preferences are shown above. In the costmap, black represents low cost and white represents high cost.

this work is on terrain-based cost evaluation, we recognize that a full navigation system would incorporate additional costs using established methods such as layered costmaps [14].

We are particularly interested in terrain cost assignment approaches that can rapidly adapt to newly-expressed terrain preferences. Prevalent approaches to incorporate human preference into navigation such as inverse reinforcement learning (IRL) and preference-based IRL (PbIRL) based on terrain patch clusters do not admit this type of rapid adaptation due to the amount of additional data required to express new preferences [15]–[17]. Instead, to the best of our knowledge, existing solutions for rapid adaptation to preferences rely on first segmenting terrains into a prescribed set of classes, and assigning each class a manually-specified cost [4], [18], [19]. While such approaches allows rapid adaptation to new preferences, they are restricted to expressing preferences over the pre-defined list of terrain classes known to the segmentation algorithm.

Representation learning is a more recent approach to terrain-aware navigation that allows preferences to no longer be limited to terrains with predefined labels [15], [16], [20], [21]. Patch-based representation learning methods for terrain understanding typically involve mapping small square patches from the bird's-eye view (BEV) to a representation vector that is further converted into a scalar cost value. Although continuous representation spaces are theoretically generalizable to new terrains, training such a space effectively is challenging in practice, as even humans may struggle to identify terrain types from small patches in the presence of homography artifacts or difficult lighting conditions. A further limitation is that each new terrain preference ordering necessitates retraining the utility function in the representation space, making these approaches less adaptable to changing preferences.

Towards overcoming the limitations of the semantics-based and representation-learning paradigms to terrain cost assignment, we propose and study PACER, a novel approach to costmap generation that accepts as input a single birds-eye
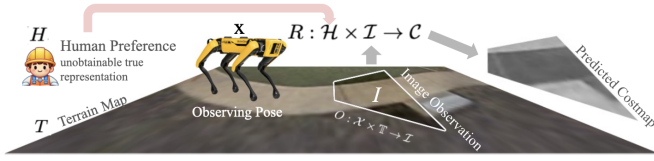
Fig. 2: Relationships between spaces of Terrains, Image Observations, and Costmaps. There exists a hidden "true" costing function based on human preferences directly on terrains. PACER approximates this function from visual observations of terrains.

view (BEV) image of the surrounding terrain along with a user-specified preference context and generates a corresponding BEV cost map that aligns with that preference context (see Fig. 1). By *preference context*, we mean a small set of terrain patches and pairwise preferences over those patches that are supplied at deployment time. We design PACER to exhibit three design desiderata: *(1)* it is capable of representing a prior over terrain preferences; *(2)* it is capable of adapting to a wide variety of preference contexts; and *(3)* it is able to assign aligned costs to terrains that appear in both the preference context and the BEV image, even for novel terrain types.

Using real and synthetic terrain data, we implement a training pipeline to realize these three properties and evaluate the resulting preference-conditioned costmap functions over a wide variety of BEV images. Additionally, we study the impact of the resulting costmaps on cost-optimal navigation behavior with respect to adherence to human preferences. We find that our method overcomes limitations in prior works by being easily adaptable to new operator preferences and producing fine-grained costmaps that illicit desirable navigation behaviors even in previously unseen environments.

II. THE TERRAIN-AWARE PREFERENCE-ALIGNED PLANNING PROBLEM

We now develop the terrain-aware preference-aligned planning problem. We will first formulate the path planning problem, and then we will discuss the problem of learning preference-aligned terrain costs.

A. Path Planning

In this paper, we are concerned with the general problem of planning a path in a robot state space $\mathcal{X}$ (SE(2) for ground vehicles) from a start and goal pose $x_1, G \in \mathcal{X}$ as the problem of finding the finite trajectory $\Gamma_S = [x_1, ..., x_S]$ consisting of S states $x \in \mathcal{X}$ which minimizes a total objective function

$$\Gamma_S = \arg_\Gamma \min ||x_S - G|| + \lambda \mathcal{J}(\Gamma), \qquad (1)$$

where $||x_S - G||$ is the distance between the final state x_S and G, and $\mathcal{J}(\Gamma)$ is the cost function scaled by the relative weight λ.

A cost function $\mathcal{J}(\Gamma)$ may include various terms such as the geometric cost of obstacles, social navigation cost, or terrain cost,

$$\mathcal{J}(\Gamma) = \mathcal{J}_{\text{geometric}}(\Gamma) + \alpha \mathcal{J}_{\text{social}}(\Gamma) + \beta \mathcal{J}_{\text{terrain}}(\Gamma) \qquad (2)$$

where α, β are relative weights. This paper is concerned with the terrain cost term $\mathcal{J}_{\text{terrain}}(\Gamma)$ of the general function.

B. Preference-Aligned Terrain Costs

To better understand preference-aligned terrain costs, we first introduce a fixed terrain function to represent the spatial distribution of the terrains in the world. Let a terrain map $T : \mathcal{X} \to \mathcal{T}$ be a function that maps a robot pose $x \in \mathcal{X}$ to the terrain $\tau \in \mathcal{T}$ that the robot interacts with when in pose x, where a terrain τ captures all the properties of the ground relevant to robot navigation.

Additionally, we assume the human has an unknown true cost function $H : \mathcal{T} \to \mathbb{R}^{0+}$ mapping terrains to scalar real-valued costs based on their preferences. This cost function is influenced by various factors, including the personal preferences of human operator, the environment, and the task at hand. Let $\mathcal{H}$ denote the continuous space of such cost functions, such that $H \in \mathcal{H}$.

For terrain-aware navigation, the robot relies on its visual observations to infer terrain-specific costs. We assume that these observations arrive in the form of images generated according to a black-box observation function $O : \mathcal{X} \times \mathbb{T} \to \mathcal{I}$, i.e., $I = O(x, T)$, where x is the observing pose of the robot, T is a terrain map, $\mathbb{T}$ is the space of terrain maps, and $\mathcal{I}$ is the space of images. In practice, most methods operate on synthetic birds-eye-views generated from the original camera images. BEV images can be generated via static ground-plane homography [15], [16], or a BEV accumulation algorithm [22]. Henceforth, we define input images to be BEV images. We assume that the the visual appearance of the terrain provides sufficient information for the robot to perform terrain-aware navigation. The observation function is thus fixed, but unknown to the robot.

During planning, the terrain cost of a pose is found using a costmap $C : \mathcal{X} \to \mathcal{R}^{0+}$ that maps from robot poses to costs. We introduce a costmap generation function $R : \mathcal{I} \times \mathcal{H} \to \mathcal{C}$ as the function mapping from the space of images $\mathcal{I}$ to the space of costmaps $\mathcal{C}$, conditioned on an unknown human cost function that belongs to $\mathcal{H}$.

Since the robot has no direct access to the terrain map T and there is no clear representation of H, the terrain-aware preference-aligned planning problem is thus to learn the function R such that, given an image observation of terrain, the optimal trajectory planned with respect to R is also optimal with respect H. The conditions in the next section will be introduced as our analyses of how we address this problem.

III. NECESSARY CONDITIONS FOR PREFERENCE-ALIGNED NAVIGATION

Seeking training tasks to help us compute valid preference-conditioned costmap functions $R(\cdot|H)$, we now state a set of necessary conditions for these tasks to produce costmaps that are consistent with human preferences for terrain.

In particular, we will state conditions for *equivalence* and *partial ordering*, and we will show that Rs that produce costmaps that yield optimal trajectories consistent with a human preference must obey these conditions.

Let $R(\cdot|H)$ denote a costmap generated according to an $H \in \mathcal{H}$, and $C|_x$ denote that costmap $C \in \mathcal{C}$ is evaluated at pose x. For a generated costmap $R(\cdot|H)$ to be consistent with

H, we specify it must exhibit both *equivalence* and *partial ordering*. By equivalence, we mean that the terrains at two poses are given the same cost by H if and only if the costmap generated by R from an image observation and evaluated at those two poses have equal cost, i.e.,

$$H_i(T(x_1)) = H_i(T(x_2)) \iff R(O(\cdot, T) \mid H_i)\Big|_{x_1}$$
$$= R(O(\cdot, T) \mid H_i)\Big|_{x_2} \quad \forall x_1, x_2, H_i . \quad \text{(NC1)}$$

By partial ordering, we mean that H assigns a preference order over the terrains at two poses if and only if the costmap generated by R from an image observation assigns those two poses the same preference order, i.e.,

$$H_i(T(x_1)) < H_i(T(x_2)) \iff R(O(\cdot, T) \mid H_i)\Big|_{x_1}$$
$$< R(O(\cdot, T) \mid H_i)\Big|_{x_2} \quad \forall x_1, x_2, H_i . \quad \text{(NC2)}$$

We let $O(\cdot, T)$ denote an image observation captured from any observing pose from which x_1, x_2 are visible.

We now provide a brief proof that (NC1), (NC2) are necessary for aligning the preferences of H_i with R. Specifically, if the most optimal path with respect to $R(\cdot \mid H_i)$ has the same optimal cost when evaluated with H_i, then the conditions (NC1),(NC2) must hold. For a trajectory Γ composed of discrete poses, let the cumulative cost function for the human's evaluation be denoted by $H|_\Gamma \equiv \sum_{x_i \in \Gamma} H(T(x_i))$. Similarly, let the cumulative cost function for the generated costmap be denoted as $R|_\Gamma \equiv \sum_{x_i \in \Gamma} R(O(\cdot, T) \mid H)|_{x_i}$. Given this setup, the following theorem establishes the necessity of the conditions (NC1), (NC2) such that $H|_{\Gamma^*} = R|_{\overline{\Gamma}}$.

Theorem 1. *Let* $\Gamma^* = \arg_\Gamma \min H|_\Gamma$, $\overline{\Gamma} = \arg_\Gamma \min R|_\Gamma$ *denote the optimal trajectories with respect to H and R respectively. If the optimal trajectory with respect to R has equal cost to the optimal path with respect to H when both are evaluated on H such that $H|_{\Gamma^*} = R|_{\overline{\Gamma}}$, then conditions* (NC1) *and* (NC2) *hold.*

Proof. Since $H|_{\Gamma^*} = R|_{\overline{\Gamma}}$, we must have that:

(a) $H|_{\Gamma_1} < H|_{\Gamma_2} \Rightarrow R|_{\Gamma_1} < R|_{\Gamma_2}$ for all paths Γ_1, Γ_2. Otherwise, there exist paths Γ_1, Γ_2 such that $H|_{\Gamma_1} < H|_{\Gamma_2}$ and $R|_{\Gamma_1} \geq R|_{\Gamma_2}$. Then, Γ_2 may be selected as $\overline{\Gamma}$, but has greater cost than $H|_{\Gamma^*}$ when evaluated on H, which is a contradiction.

(b) $R|_{\Gamma_1} < R|_{\Gamma_2} \Rightarrow H|_{\Gamma_1} < H|_{\Gamma_2}$ for all paths Γ_1, Γ_2. Otherwise, there exist paths Γ_1, Γ_2 such that $R|_{\Gamma_1} < R|_{\Gamma_2}$ and $H|_{\Gamma_1} > H|_{\Gamma_2}$ (by contraposition on (a), we eliminate the case where $R|_{\Gamma_1} < R|_{\Gamma_2}$ and $H|_{\Gamma_1} = H|_{\Gamma_2}$). Then, Γ_1 may be selected as $\overline{\Gamma}$, but may have greater cost than $H|_{\Gamma^*}$ when evaluated on H, which is a contradiction.

By (a) and (b), we have that $H|_{\Gamma_1} < H|_{\Gamma_2} \iff R|_{\Gamma_1} < R|_{\Gamma_2}$. By contraposition, we also have $H|_{\Gamma_1} = H|_{\Gamma_2} \iff R|_{\Gamma_1} = R|_{\Gamma_2}$. Finally, since a path Γ can consist of a single state, conditions (NC1) and (NC2) must also then hold. $\square$

In the next section, we use conditions (NC1) and (NC2) to define training tasks for learning the optimal R from data, which drives our proposed approach to the online generation of costmaps which result in preference-aligned navigation.

IV. PREFERENCE-ALIGNED ALL-TERRAIN COSTMAP GENERATION

We now present our proposed approach for computing aligned terrain costmaps, which we refer to as Preference-aligned, All-terrain Costmap genERation (PACER). PACER introduces the notion of a preference context and comprises several components, including a neural network architecture, and a data curation and training methodology based on the three design desiderata.

A. Preference Context

The preference-aligned terrain costs discussed in Section II depend on a human's cost function $H : \mathcal{T} \to \mathbb{R}^{0+}$. Unfortunately, we do not have access to H directly since it is known only to the human operator. Therefore, we propose to obtain and utilize an approximate representation of H that we call a *preference context*.

We define a preference context $\hat{H}$ as a set of n image patch pairs $\widetilde{I} \succ \widetilde{I}^{'}$ constructed from human input such that the human prefers the terrain observed in image $\widetilde{I}$ over the terrain observed in $\widetilde{I}^{'}$, where an $\widetilde{I} \in \mathcal{I}$ is an observation of terrain as a small image patch. The small patch may be a part of a larger bird's-eye-view image of the ground. More specifically, $\hat{H}$ consists of n preferences derived from H and is defined as $\hat{H} \equiv \{(\widetilde{I}_1 \succ \widetilde{I}^{'}_1), ...(\widetilde{I}_n \succ \widetilde{I}^{'}_n)\}$. Fig 3 shows some example preference contexts with $n = 3$ patch pairs and their corresponding costmaps.

In our implementation, the n pairwise preferences are expressed using image patches of size $h \times w$. $\hat{H}$ is then represented by vertically concatenating the patches within a pair with the more-preferred terrain patch on top and forming a single $(n \cdot c) \times 2h \times w$ tensor, where c is the number of color channels. Given a finite preference context, it is impossible to specify all pairwise preferences over the terrains, especially since the terrain set is continuous - hence it is not possible to specify a stronger sufficient condition that would guarantee that the algorithm generates human-aligned costs.

B. Model Architecture

To generate costmaps, we propose to approximate functions $R : \mathcal{I} \times \mathcal{H} \to \mathcal{C}$, which require H as input, with functions $\hat{R} : \mathcal{I} \times \hat{\mathcal{H}} \to \mathcal{C}$, where $\hat{\mathcal{H}}$ is the space of all preference contexts as defined above. We model $\hat{R}$ as a neural network with a two encoders and a single decoder. The input image is passed through a BEV image encoder F_{BEV} to form an image embedding, and, similarly, the input preference context is passed through a preference context encoder F_{pref} to form a preference embedding. The output costmap is then generated by concatenating these embeddings and then passing them through a decoder D. A visual depiction of this architecture is provided in Fig. 1.

C. Loss Function

PACER is trained using supervised machine learning, i.e., given a dataset $\mathcal{D} = \{(\hat{H}, I, C_T)_i\}_{i=1}^N$ of preference context,
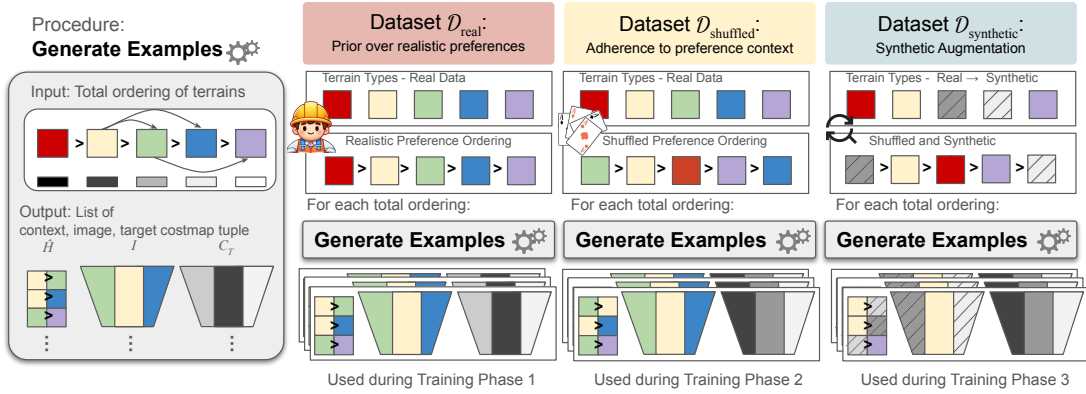
Fig. 3: Overview of the dataset structure. Each training example contains a preference context, image, and target costmap. We vary the preferences and images, resulting in a large combinatorial dataset despite the relatively small amount of real recorded data. In a later training phase, we also augment with synthetic data by artificially finding and replacing certain terrain types with synthetic terrain textures. The real-valued costs assigned to terrain types based on an input total ordering are shown in the Generate Examples procedure, where black represents low cost and white is high cost.

image, and target costmap tuples, we seek the parameters ϕ of $\hat{R}$ that minimize a loss between the real and predicted costmaps. More specifically, we seek ϕ^* such that

$$\phi^* = \arg\min_\phi \mathbb{E}_{(I,\hat{H},C_T)\sim\mathcal{D}} \left[\ell\left(\hat{R}_\phi(I, \hat{H}), C_T \right) \right] , \quad (3)$$

where we use the binary cross entropy loss averaged over each pixel as the loss function ℓ.

V. DATASET CURATION AND TRAINING PACER

We now describe the dataset curation and training process for PACER. PACER is trained using three distinct phases of supervised machine learning, each corresponding to a unique training dataset that corresponds to one of the desiderata described in Section I. In what follows, we will first describe how we generate training examples, then describe each of the three training phases and the training procedure.

A. Training Example Generation

The datasets $\mathcal{D}$ we use to train the PACER model consist of tuples of preference contexts, images, and target costmaps $(\hat{H}, I, C_T)$. To construct these datasets, we bootstrap off of semantic terrain classification and use a pretrained terrain patch classifier that assigns one of L predefined semantic labels to a given terrain image patch.

The inputs to the training example generation process are a single image I along with a total ordering over terrain types $\tau_1 \succ \tau_2 \succ \ldots \succ \tau_L$, where each terrain type τ_l corresponds to a bank of image patches. PACER assumes that the cost value associated with τ_l is given by $H(\tau_l) = \frac{l-1}{L-1}$. The bank corresponding to τ_l consists of patches of that type extracted from images collected during robot deployment.

We use these inputs to generate $\hat{H}$ and C_T. To generate $\hat{H}$, we first choose n ordered pairs from the total ordering over the L terrain types without replacement. For each of the resulting ordered pairs, we sample uniformly at random patches from the corresponding patch banks, and use these $2n$ patches to construct $\hat{H}$ according to the process detailed in Section IV-A above. To generate C_T, we perform semantic segmentation on

I and transform the segmented image into C_T by setting the cost for a pixel labelled l to be $H(\tau_l)$.

Constructing training examples in this way encourages $\hat{R}_{\phi*}$ to follow our necessary conditions. First, because C_T assigns the same cost value to image locations that received the same semantic label, $\hat{R}_{\phi*}$ is encouraged to identify regions of visually-similar terrain and assign *equivalent* costs within the region, as per condition (NC1). Second, because both $\hat{H}$ and C_T are, by construction, consistent with H, $\hat{R}_{\phi*}$ is encouraged to predict costmaps given $\hat{H}$ which preserve the partial ordering of H, as per condition (NC2). Interestingly, assuming sequential segmented images are temporally consistent, we observe that $\hat{R}_{\phi*}$ is encouraged to be viewpoint-invariant.

During inference time, there are no semantic labels and only the visual appearances of terrains are considered.

B. Dataset Size

The size of the space from which we sample data is very large. From a total ordering of L discrete terrain types, there are $m = \binom{L}{2}$ different ordered pairs of terrains and $\binom{m}{n}$ different sets of n pairs. For each set of n pairs, there $n!$ ways to shuffle the pairs to construct the preference context, yielding $\binom{m}{n} \cdot n!$ possible preference contexts. Moreover, for each terrain type in the preference context, we sample a patch from the bank. Our dataset contains a bank of around 800 patches for each terrain label, and about 950 full images. Therefore, for each total ordering, we have $\binom{\binom{L}{2}}{n} n!$ arrangements of labels into preference contexts, where we sample a patch from a bank of 800 patches for each terrain type. For L terrains, $n = L \log L$ pairs are needed to describe a total ordering, though we evaluate on a smaller $n = 3$ pairs due to size considerations for the model and dataset.

C. Training Phases

Each of the three system desiderata stated in Section I is manifested in a distinct training phase, each of which utilizes a unique training dataset generated using the procedure described above. More specifically, these phases generate

datasets $\mathcal{D}_{\text{real}}$, $\mathcal{D}_{\text{shuffled}}$, and $\mathcal{D}_{\text{synthetic}}$, which promote adherence to prior preferences in seen terrains, robustness to new preferences, and robustness to new terrains, respectively.

A visualization of each of these phases is given in Figure 3, and we describe each phase in more detail below.

Training Phase 1: Pretraining with Real Data and Realistic Preferences. To promote a prior towards an overall "realistic" ordering (as per our first desired property), PACER's first training phase constructs and utilizes a dataset $\mathcal{D}_{\text{real}}$ generated using real-world data collected from robot deployments around our campus at The University of Texas at Austin and realistic preferences over terrain classes. An example of a realistic preference ordering is as follows: `concrete` $\succ$ `pebble` $\succ$ `grass` $\succ$ `marble` $\succ$ `bush`. The "realistic preferences" were defined by the first author according to considerations for robot safety (e.g. preferring grass over loose marble for a wheeled robot) and societal norms (e.g. preferring concrete over grass to avoid trampling lawns, even though both terrains are relatively safe).

Training Phase 2: Augmentation with Changed Preferences. During deployment in terrains not seen during training, the robot should adhere to preferences given by the operator (as per our second desired property). Even when operator preferences contradict "realistic preferences", the robot should follow operator preferences over learned priors. To encourage this adherence to the ordering in the preference context, we train using the same real data but with changed preferences on a smaller corpus of data by using a randomly-permuted total ordering over terrain labels.

Training Phase 3: Augmentation with Synthetic Terrains. To promote the model's ability to generalize to terrains unseen during training (as per our third desired property), we further train with synthetically augmented data.

We pick a random subset of terrains to replace and randomly permute the preference order. An image containing at least one such terrain is selected, and those terrains are artificially replaced with terrain textures from an open-source database [23] using dense segmentation. We used 14 synthetic textures. The training example is formed with a preference context (where terrains have been replaced), the image, and a costmap with costs reassigned according to the new preference order.

In the first phase, only $\mathcal{D}_{\text{real}}$ is used for training examples. In the second, both $\mathcal{D}_{\text{real}}$ and $\mathcal{D}_{\text{shuffled}}$ are used. In the third, all three datasets are used. Within a phase, training examples are drawn uniformly among the datasets used. Before training on $\mathcal{D}_{\text{real}}$, weights are initialized randomly. After completing a training phase, we switch to the next phase starting from the previous trained weights. We trained for 100 epochs in phase 1, 5 in phase 2, and 100 in phase 3.

VI. Experiments

To evaluate PACER, we seek to answer the following questions empirically:

1 How effectively is the robot able to navigate in terrains *seen during training* when the preference context contains **(a)** only seen terrains or **(b)** only previously unseen terrains?

2 How effectively is the robot able to navigate in *unseen* terrains when the preference context contains **(a)** only those unseen terrains or **(b)** only seen terrains?

By dividing deployment scenarios into the four situations above, we will be able to understand the performance of PACER under the four ways to combine *seen* and *unseen* terrains in the preference context and environment. In **1b** and **2b**, the preference context does not provide information about the terrains appearing in the environment, so the robot must rely on learned priors about realistic cost assignment. We term this scenario as having an *uninformative context*.

Evaluations are performed using simulated experiments on an aerial map. In a later section, we also provide results from real robot deployments. We compare against STERLING [15] (a representation learning approach) and a classifier (a semantics-based approach) as baselines. The classifier is the same as used generate the training data for PACER. The same model for PACER is utilized across all environments. No retraining or fine-tuning is done. Additionally, a single context is used for the duration of a simulated deployment in an environment (i.e. the context is not switched out midway through the path-planning).

To quantify the navigation performance in our experiments, we posit that factors such as the distance traversed or closeness to a human-defined trajectory do not matter as much as traveling on only the preferred terrains. We therefore assign terrain types in each experiment a low, medium, and high cost, according to the human deployer's preference and report the proportion of the planned path which belonging to each tier. The assignment of semantic terrain types to these three tiers is based on a hidden total ordering of all the terrains which appear in the environment. Each of the methods and baselines tested are deployed with a preference (provided according to their respective representations) consistent with the hidden total ordering. Note that we have purposely chosen this metric to be different from the commonly-used Hausdorff distance between the planned trajectory and one defined by a human operator, which can vary greatly when there are multiple valid paths to the goal.

A. Aerial Map Experiments

In our simulated experiments, we build aerial maps from drone footage of three locations around our campus, which we consider seen environments. We also use open-source aerial maps [24] from around the world, covering a wide variety of both urban and natural terrain types and which we consider unseen. For each of the seen and unseen environments, we provide a start and goal location and test varying operator preferences. We test realistic preferences, and "inverted" preferences, in which each of the pairwise orderings in the realistic preference are reversed. Given the robot's pose on the aerial map, the robot's projected bird's eye view can be found, and used as input to generate the local costmap. Planning is done using the A* algorithm [25] on the costmap. In Fig. 5, we show examples of paths planned using PACER in our aerial map simulator. These experiments are an evaluation purely over planning based on terrain preference, with no
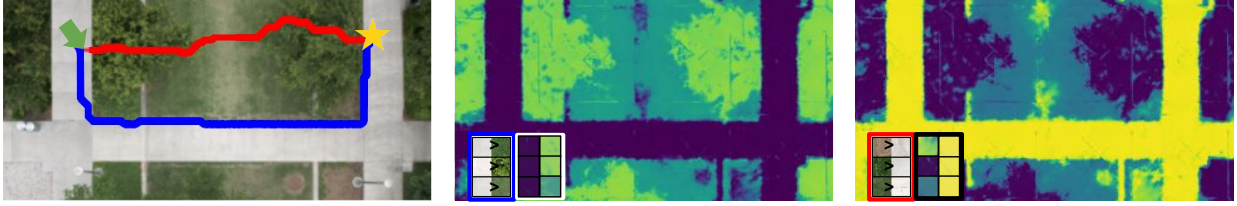
6

Fig. 4: The effect of changing preference on path planning is shown on the left image of an aerial map. The blue path corresponds to the middle costmap and the red path corresponds to the right costmap. The generated costmaps reflect the preferences provided in the context. Dark purple corresponds to low cost and yellow is high cost.

Fig. 5: Examples of several paths planned using our method in an urban unseen environment and their corresponding preference contexts. Here, we visualize the large scale of these simulated deployments, and the diversity of visual terrain appearances.

kinodynamic constraints, no errors due to localization, and no costs associated with elevation.

Tab. I displays the results for *seen* environments. Towards answering question **1a**, PACER has similar results to the STERLING baseline, as both approaches were trained on the same in-distribution data. When no useful context has been provided for PACER (i.e. the context contains only unseen terrains which do not match the environment as per **1b**), the results are on par with the classifier baseline. PACER's success despite the lack of an informative preference context shows that the model has captured a prior over realistic cost-assignment for in-distribution terrains.

Tab. II displays the results for *unseen* environments. Towards answering question **2a**, when given an informative context, PACER outperforms the STERLING baseline. Though representation-learning approaches like STERLING should theoretically generalize due to their continuous representation space, this is contingent on similar terrains forming clusters in this space, which may not be the case for unseen terrains. Fapt install r-base-coreor unseen environments, the classifier baseline has been omitted, as it allows no way for a user to provide terrain preferences for classes that are not predefined. PACER overcomes the limitations of previous paradigms, as it both allows preferences in unseen environments to be expressed and generalizes well to these unseen environments. Additionally, when no useful context has been provided for the unseen terrains (per **2b**), PACER is not able to adapt.

B. Ablation Study

To understand the effects of each phase in the training process, we perform an ablation study using the same environments as in the aerial simulator experiments. In the $\mathcal{D}_{\text{real}}$ phase, the model is trained only on real data and realistic preferences. In the $\mathcal{D}_{\text{shuffled}}$ phase, the model is pretrained on real data with realistic preferences, and then trained on a

Method	Low (%)	Medium (%)	High (%)
PACER	73.83%	19.70%	6.47%
PACER (uninformative context)	67.97%	15.89%	16.15%
STERLING	74.72%	18.65%	6.63%
Classifier	65.86%	21.46%	12.69%
Ground Truth	82.82%	13.69%	3.49%

Tab. I: Proportion of planned paths that traverse low, medium, and high-cost terrains in seen environments, relating to **1a,1b**.

Method	Low (%)	Medium (%)	High (%)
PACER	81.85%	8.00%	10.15%
PACER (uninformative context)	53.83%	2.20%	43.97%
STERLING	61.39%	10.12%	26.86%
Ground Truth	91.85%	6.02%	2.13%

Tab. II: Proportion of planned paths that traverse low, medium, and high-cost terrains in unseen environments, relating to **2a,2b**.

In-Dist. Scenario	Low (%)	Medium (%)	High (%)
Realistic Preference			
$\mathcal{D}_{\text{real}}$	95.97%	1.70%	2.32%
$\mathcal{D}_{\text{shuffled}}$	70.50%	18.94%	10.55%
$\mathcal{D}_{\text{synthetic}}$	76.66%	18.43%	4.91%
Inverted Preference			
$\mathcal{D}_{\text{real}}$	18.55%	20.97%	60.48%
$\mathcal{D}_{\text{shuffled}}$	69.21%	22.56%	8.23%
$\mathcal{D}_{\text{synthetic}}$	70.22%	21.32%	8.46%

Tab. III: Proportion of planned paths that traverse low, medium, and high-cost terrains in seen environments for each training phase. The model after each phase of training is tested using realistic and inverted contexts.

Out-of-Dist. Scenario	Low (%)	Medium (%)	High (%)
Realistic Preference			
$\mathcal{D}_{\text{real}}$	60.78%	7.05%	32.18%
$\mathcal{D}_{\text{shuffled}}$	65.16%	6.44%	28.40%
$\mathcal{D}_{\text{synthetic}}$	88.09%	2.57%	9.34%
Inverted Preference			
$\mathcal{D}_{\text{real}}$	22.59%	15.31%	62.10%
$\mathcal{D}_{\text{shuffled}}$	39.86%	13.96%	46.17%
$\mathcal{D}_{\text{synthetic}}$	65.65%	22.10%	12.25%

Tab. IV: Proportion of planned paths that traverse low, medium, and high-cost terrains in unseen environments for each training phase. The model after each phase of training is tested using realistic and inverted contexts.

smaller amount of changed preferences. In $\mathcal{D}_{\text{synthetic}}$ phase, the model is trained according to all three phases.

Results are shown in Tab. III, IV. In seen environments, the model trained on $\mathcal{D}_{\text{real}}$ performs the best of the three with realistic preferences, but is unable to adapt when the preferences are inverted. Results for realistic preferences (black) are significantly better than those for inverted preference (blue) for this method in Tab. III. In unseen environments (Tab. IV), this method performs the worst regardless of preference context. The model trained on $\mathcal{D}_{\text{shuffled}}$ is able to adapt to changing preference orderings in seen environments, seen by comparing results for realistic preferences (black) to inverted preferences (blue) in Tab. III. However, it is unable to recognize and match new terrains in the preference context to the new environment as evidenced by the drop in performance from Tab. III to Tab. IV. The model trained on $\mathcal{D}_{\text{synthetic}}$ is

Test Dataset	$\mathcal{D}_{\text{real}}$	$\mathcal{D}_{\text{shuffled}}$	$\mathcal{D}_{\text{synthetic}}$
Model			
$\mathcal{D}_{\text{real}}$	0.017	0.099	0.072
$\mathcal{D}_{\text{shuffled}}$	0.086	0.019	0.064
$\mathcal{D}_{\text{synthetic}}$	0.053	0.069	0.013

Tab. V: Margin ranking error for each of the three models on the test set of each of the three datasets. Lower error is better.

shown to both respect changing preference order and recognize new terrains. Our findings indicate that training the model to respond to preferences reduces performance with realistic preferences in seen environments, as D_{shuffled} and $D_{\text{synthetic}}$ have slightly poorer performance than D_{real} with realistic but far better performance with inverted preferences. Furthermore, performance across all three methods decreases from seen to unseen since the scenario is harder, but $D_{\text{synthetic}}$ has the least drop in performance. Variable preferences and unseen environments introduce greater complexity, requiring models to adapt and generalize. We prioritize these harder, practical scenarios, accepting a trade-off with peak accuracy in simpler settings.

C. Performance on Test Set

There are infinite costmaps which may satisfy a preference over terrains (given as a partial order of terrains), since the scale of the numerical costs do not matter as long as the ordering is followed. Therefore, rather that direct comparison with a "ground-truth" costmap, we report the *margin ranking error* of predicted costmaps using holdout set from each dataset in Tab. V. We sample the 500 points across the costmap, then for each pair of points, calculate the error as 0 if the relative magnitude of the two costs is correct. Otherwise the error is the difference between the two costs. We report the mean error for each category.

For each dataset, the model which was trained on that dataset had the lowest error. Across all datasets, $\mathcal{D}_{\text{synthetic}}$ had the lowest or second-lowest error, and had therefore the overall best performance.

D. Discussion

The results presented in this section demonstrate that PACER fulfills our three design desiderata. The adherence of PACER to realistic preferences when not given an informative preference context fulfills the first key property of being able to make inferences when there is no context by capturing a prior over realistic preferences (**1b**). As per the second key property, PACER has been shown to align costs to preferred terrains even as preferences are varied (**1a**). An example of paths planned according to different preferences is shown in Fig. 4. The performance of PACER in both seen and unseen environments when given an informative preference context shows that PACER exhibits the third key property (**1a, 2a**).

VII. REAL ROBOT EXPERIMENTS

We now seek to demonstrate that PACER performs well during execution in the real world. We deploy our method, STERLING, and the classifier approach on a mobile robot at

Fig. 6: The four environments where real-robot trials were performed. Blue arrows and yellow stars show start and goal locations respectively. The red dashed line marks the intended path based on operator preference.

Environment	Env 1	Env 2	Env 3	Env 4
Classifier	**5/5**	4/5	1/5	2/5
STERLING	0/5	2/5	0/5	**4/5**
Ours	**5/5**	**5/5**	**5/5**	**4/5**

Tab. VI: Number of successes per 5 trials of different approaches across various environments. A trial is a success if the robot reaches the goal without traversing across undesirable terrain and without operator intervention.

four different locations on the UT campus which are not included in the simulated environments. These four locations cover `red brick`, `concrete sidewalk`, `grass`, `mulch`, and `pebble pavement`.

Since all methods are trained to be view-point invariant and platform-agnostic, we trained them all with the same data collected from a Boston Dynamics Spot, and deployed zero-shot on a Clearpath Jackal which has significant differences in viewpoint and mobility than the Spot. We evaluate the performance of each method with a realistic preference on a variety of terrain types. In these experiments, we measure the robot's ability to execute the plan, which includes robustness to a different viewpoint and platform. We integrate each method with a sampling-based local planner [26] and maintain the same planner parameters to ensure fairness.

Tab. VI shows results from real robot experiments. Our approach had the most successful trials across all environments. While the classifier performed well in environments 1 and 2, we hypothesize that difficult lightning conditions and variations in terrain appearance caused the failures in environments 3 and 4. Though STERLING performed as the best baseline in the simulated experiments (which involved only planning), it seemed to be unable to execute these plans in real-robot experiments. Many of the failure cases involved the robot driving slightly off-path and just grazing the undesirable terrains. In patch-based representations, a single patch may contain multiple different terrains (e.g. half sidewalk and half grass), so the cost assigned to the patch would be some combination of the different terrain costs, resulting in a coarser degree of control on the physical robot. Our approach overcomes this limitation since it directly outputs a fine-grained costmap in a single forward pass.

VIII. LIMITATIONS AND FUTURE WORK

The experiments reported in this paper were conducted on a single campus with one robot. Future work should extend the study to more varied terrains and deploy multiple robots with diverse sensors. Additionally, we aim to evaluate the ease with which human users can express preferences using our method, as well as explore alternative mechanisms for preference expression in costmap generation. Incorporating a variable-length preference context would also be an interesting direction for future work, enabling greater flexibility and

adaptability in representing user preferences. Exploring other model architectures could further enhance the system's robustness and performance in diverse scenarios. Finally, incorporating depth data into the costmap generation is another key area for improvement, as our current reliance on a homography transformation assumes flat ground, leading to limitations like the inability to avoid obstacles such as concrete curbs, despite their preferable terrain.

IX. Conclusion

In this paper we presented PACER, a novel architecture and training approach to quickly produce costmaps according to arbitrary user preferences and new terrains with no fine-tuning. Our approach was evaluated against semantics-based and representation-learning baselines in both simulated and real robot experiments. We have shown this approach to be highly adaptable to new preferences and terrains, as well as able to infer the traversability of some terrains according to realistic preferences.

X. Acknowledgements

References

[1] X. Xiao, Z. Xu, Z. Wang, Y. Song, G. Warnell, P. Stone, T. Zhang, S. Ravi, G. Wang, H. Karnan *et al.*, "Autonomous Ground Navigation in Highly Constrained Spaces: Lessons Learned From the Benchmark Autonomous Robot Navigation Challenge at ICRA 2022 [competitions]," *IEEE Robotics & Automation Magazine*, pp. 148–156, 2022.

[2] X. Xiao, Z. Xu, G. Warnell, P. Stone, F. G. Guinjoan, R. T. Rodrigues, H. Bruyninckx, H. Mandala, G. Christmann, J. L. Blanco-Claraco *et al.*, "Autonomous Ground Navigation in Highly Constrained Spaces: Lessons Learned From the 2nd BARN Challenge at ICRA 2023," *arXiv preprint arXiv:2308.03205*, 2023.

[3] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially Compliant Navigation Dataset (SCAND): A Large-scale Dataset of Demonstrations for Social Navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.

[4] X. Meng, N. Hatch, A. Lambert, A. Li, N. Wagener, M. Schmittle, J. Lee, W. Yuan, Z. Chen, S. Deng, G. Okopal, D. Fox, B. Boots, and A. Shaban, "TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation," in *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023.

[5] J. Choi, C. Dance, J.-e. Kim, K.-s. Park, J. Han, J. Seo, and M. Kim, "Fast adaptation of deep reinforcement learning-based navigation skills to human preference," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3363–3370.

[6] 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.

[7] M. Wulfmeier, D. Rao, and I. Posner, "Incorporating human domain knowledge into large scale cost function learning," in *Neural Information Processing Systems, Deep Reinforcement Learning Workshop*, 2016.

[8] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An Autonomous Self-supervised Learning-based Navigation System," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1312–1319, 2021.

[9] ——, "Land: Learning to Navigate from Disengagements," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1872–1879, 2021.

[10] A. H. Raj, Z. Hu, H. Karnan, R. Chandra, A. Payandeh, L. Mao, P. Stone, J. Biswas, and X. Xiao, "Rethinking Social Robot Navigation: Leveraging the Best of Two Worlds," in *2024 IEEE International Conference on Robotics and Automation (ICRA). IEEE*, 2024.

[11] D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window Approach to Collision Avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[12] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Aggressive Driving with Model Predictive Path Integral Control," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 1433–1440.

[13] S. Macenski, T. Moore, D. V. Lu, A. Merzlyakov, and M. Ferguson, "From the desks of ROS maintainers: A survey of modern &; capable mobile robotics algorithms in the robot operating system 2," *Robotics and Autonomous Systems*, vol. 168, p. 104493, Oct. 2023. [Online]. Available: http://dx.doi.org/10.1016/j.robot.2023.104493

[14] D. V. Lu, D. Hershberger, and W. D. Smart, "Layered costmaps for context-sensitive navigation," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems.* IEEE, 2014, pp. 709–715.

[15] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone, "STERLING: Self-Supervised Terrain Representation Learning from Unconstrained Robot Experience," in *Proceedings of The 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 229. PMLR, 2023, pp. 2393–2413.

[16] K. S. Sikand, S. Rabiee, A. Uccello, X. Xiao, G. Warnell, and J. Biswas, "Visual representation learning for preference-aware path planning," in *2022 International Conference on Robotics and Automation (ICRA).* IEEE, 2022, pp. 11 303–11 309.

[17] H. Karnan, E. Yang, G. Warnell, J. Biswas, and P. Stone, "Wait, That Feels Familiar: Learning to Extrapolate Human Preferences for Preference-aligned Path Planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 008–13 014.

[18] P. Jiang, P. Osteen, M. Wigness, and S. Saripalli, "Rellis-3d dataset: Data, benchmarks and analysis," in *2021 IEEE international conference on robotics and automation (ICRA).* IEEE, 2021, pp. 1110–1116.

[19] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "Ga-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8138–8145, 2022.

[20] X. Yao, J. Zhang, and J. Oh, "RCS: Ride Comfort-aware Visual Navigation via Self-supervised Learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2022, pp. 7847–7852.

[21] J. Zürn, W. Burgard, and A. Valada, "Self-supervised Visual Terrain Classification from Unsupervised Acoustic Feature Learning," *IEEE Transactions on Robotics*, vol. 37, no. 2, pp. 466–481, 2020.

[22] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter, "Elevation Mapping for Locomotion and Navigation Using GPU," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* IEEE, 2022, pp. 2273–2280.

[23] P. Haven, "Poly haven textures," 2024, accessed: 2024-06-05. [Online]. Available: https://polyhaven.com/textures

[24] "OpenAerialMap: An Open Source Platform for Aerial Imagery," 2024, accessed: 2024-06-05. [Online]. Available: https://www.openaerialmap.org/

[25] P. Hart, N. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968. [Online]. Available: https://doi.org/10.1109/tssc.1968.300136

[26] J. Biswas, "Amrl autonomy stack," 2013, accessed: 2024-06-05. [Online]. Available: https://github.com/ut-amrl/graph_navigation