Long Range Navigator (LRN): Extending robot planning horizons beyond metric maps

Matt Schmittle*
University of Washington
schmttle@uw.edu

Rohan Baijal*
University of Washington
rbaijal@uw.edu

Nathan Hatch
Overland AI
nathan@overland.ai

Rosario Scalise
University of Washington

Mateo Guaman Castro
University of Washington

Sidharth Talia
University of Washington

Khimya Khetarpal
Google DeepMind, Mila

Byron Boots
University of Washington

Siddhartha Srinivasa
University of Washington

Abstract: A robot navigating an outdoor environment with no prior knowledge of the space must rely on its local sensing, in the form of a local metric map or local policy with some fixed horizon. A limited planning horizon can often result in myopic decisions leading the robot off course or worse, into very difficult terrain. In this work, we make a key observation that effective long range navigation only necessitates identifying good frontier directions for planning instead of full map knowledge. To address this, we introduce Long Range Navigator (LRN), which learns to predict 'affordable' frontier directions from camera images. LRN is trained entirely on unlabeled egocentric videos, making it scalable and adaptable. In off-road tests on Spot and a large vehicle, LRN reduces human interventions and improves decision speed when integrated into existing navigation stacks. personalrobotics.github.io/lrn

Keywords: Field Robotics, Navigation, Perception, Planning

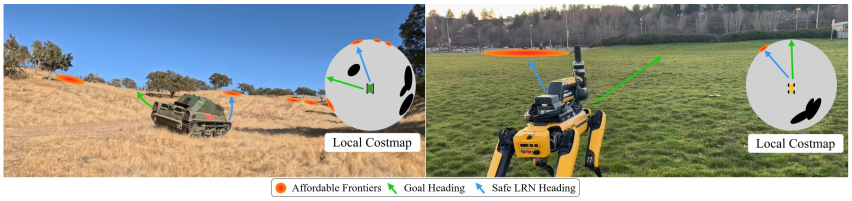

Figure 1: LRN **Overview.** Our approach LRN finds affordable frontiers as intermediate representation for the robot to head towards and selects one near the goal heading. On the right is the local perception (LEFT 50m, RIGHT 8m) where LRN changes the default navigation direction (green) to an affordable one (blue).

1 Introduction

Autonomous off-road mobile robots require long-range waypoint navigation, often in environments where prior information (e.g. satellite imagery) is inaccurate or unavailable. Our goal is efficient navigation in these large-scale scenarios where our domain is significantly (10X or more) larger than the robot's sensor footprint. The central research question is,

How can we enable robots to make less myopic decisions facilitating long-range navigation *with only* incomplete *knowledge of the environment?*

We assume that the robot has GPS localization together with target waypoints. Notably, the robot is provided no other information about the environment and must find its own path to the goal with minimal human intervention and as fast as possible. Mobile robots with no prior information traditionally create a metric cost map based on onboard sensory information (e.g. cameras, lidar, and odometry) [1, 2]. This suffices for short horizon targets, but with long-range goals, creating large

[*] Equal contribution

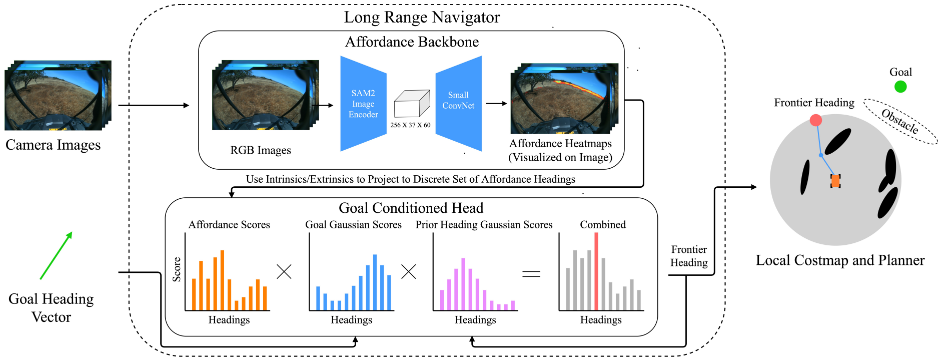

Figure 2: **Overview of our approach**. LRN is fed with egocentric camera images and a goal heading vector. LRN is composed of the following components, 1) **Affordance Backbone**: computes *affordable* frontiers in the image space as goal agnostic heatmaps. These affordance hotspots are then projected into a discrete set of affordable headings for the robot to follow, 2) **Goal Conditioned Head**: multiplies affordance scores with a discrete gaussian score around the goal and a separate gaussian around the previous prediction (to maintain consistency). The maximum combined score heading (red) is selected. The local system can then use that frontier as a goal for local planning instead of the true goal. This process is repeated with new sensor information

cost maps is intractable due to limited sensor range, compute and memory. In particular, additional depth information required to project features onto the map is often sparse and noisy, resulting in a limited horizon of the cost map and the area outside this horizon being a fog of unknown space.

A go-to approach to deal with unknown space is to heuristically assign it a fixed cost effectively planning straight to the goal once outside the map [2]. However, this leads to highly inefficient and potentially unsafe paths in many scenarios. A **key observation** is that field operators can determine the robot's long range strategy by simply analyzing the robot's image feed, without the requirement of a complete terrain map. For example, it is possible to spot the opening in a wall of trees from images without mapping every tree. We refer to the boundaries between known/unknown regions as *frontiers*. Frontiers which visually appear open i.e. possible to navigate to and continue beyond are referred to as *affordable* frontiers. With this, we offer the following **key insight**:

A robot can reason further by learning to identify distant affordable *frontiers as intermediate goals.*

We propose improvements to current heuristic-based approaches by learning affordable frontiers in the image space. Specifically, we leverage the SAM2 foundation model [3], pre-trained on a large corpus of image data, to generate meaningful embeddings from camera sensor images. These embeddings capture notions of scene segmentation, invariant to lighting, camera angle, and texture. We train a small LRN specific decoder to predict long-range affordance heatmaps in a *goal-agnostic* manner. We then project the heatmaps into heat scores for different headings the robot could follow and re-weight each heading based on the goal context. The top scoring direction is passed to the local system as a heading to follow to reach the goal. We refer to our method as **Long Range Navigator** (LRN) depicted in Fig. 2. **Key contributions** of our work are:

- We introduce LRN, an approach to extend the planning horizon of robot navigation systems using rich camera data. Our key insight is leveraging an intermediate affordance representation, amenable to image foundation models that suffices to guide local planning.
- To reduce the dependency on human-expert annotations and enable fast training for target environments, we leverage video point tracking (CoTracker [4]) to automatically label human walking videos. With only one hour of video, we train LRN for deployment on Spot.
- We demonstrate the efficacy of LRN in real world outdoor navigation tasks, on a quadruped Spot robot and a Big Vehicle traveling distances of over a kilometer.

2 Problem Setup

We consider the long-range navigation problem setting where the robot perceives its environment from its local sensors (e.g. camera, lidar), and is tasked with navigating to a distant goal $g \in \mathcal{G}$ in a static, unknown environment. The robot is equipped with a local policy $\pi_\lambda : \mathcal{O} \to \mathcal{A}$, that maps the current observation o to primitive actions $a \in \mathcal{A}$ and plans under a cost function $C : (s, a, s') \to \mathbb{R}$.

Planning is limited to some horizon H due to sensor or compute limitations. We assume the policy receives an observation o, plans, executes an action, and replans at some frequency in a model predictive control fashion. As it executes actions, the robot creates a path ξ^π. The **objective** is to minimize the expected cost of navigating to the goal in an unknown environment ϕ.

$$J(\xi^\pi) := \operatorname*{argmin}_{\xi^\pi} \mathbb{E}_{\phi\sim\oplus}\Big[C(\xi^\pi)\Big] \tag{1}$$

3 Long Range Navigator (LRN)

Core to our approach is the idea that while low-level controller policy π_λ is limited to a local horizon, it does not mean that useful sensor data does not exist beyond that horizon for navigation. Consider a set of frontier nodes $f \in \mathcal{F}$ at the periphery of horizon H which borders known/unknown space. Under optimal substructure property [5], if $f^* \in \mathcal{F}$ on the optimal path from start to goal and π_λ is optimal up to H then π_λ planning to f^* is acting globally optimally. In practice, this is hard because π_λ is usually sub-optimal and f^* depends on unobserved information (e.g. backside of a hill). That being said, there may exist information within the robot's sensor range that can help estimate affordable frontiers which seem possible to navigate to and continue beyond.

To estimate affordable frontiers from state s, we first define the value of a frontier f given a goal g as $V(s,g,f)$. Note, this is an approximation of the true value function. $V(s,g,f)$ can be decomposed into two parts, namely $A(s,f)$ and $D(f,g)$. Formally,

$$V(s,g,f) = A(s,f)\; D(f,g) \tag{2}$$

$$A(s,f) = P(\exists\xi_{sf})\; P(\exists\xi_{fg}, \exists g \mid d(s,g) > H), \tag{3}$$

where $A(s,f)$ measures the *affordability score*. This score is computed by multiplying the probability there exists a path from s to f and the probability there exists some path from f to some distant goal g beyond the local horizon H. $D(f,g)$ measures the cost estimate of navigation from the frontier f conditioned on the goal. Given, $V(s,g,f)$ and π_λ we can define the policy we seek π.

$$f_\pi = \operatorname*{argmax}_{f\in F} V(s,g,f) \tag{4}$$

$$\pi(s,g) = \pi_\lambda(s, f_\pi) \tag{5}$$

LRN is a bi-level system, with two components: one to estimate $A(s) = [A(s,f)]_{f\in F}$ and the other to estimate $D(f,g)$. We implement $A(s)$ to estimate *all affordance scores* via a learned mapping from images to affordances (selective attention) in the image space. The second component scores frontier states given a goal context. The overall algorithm is depicted in Alg. 1. In practice, we discretize the space around the robot into angular bins which constitutes the space of frontiers $\mathcal{F}$.

Evaluating frontiers using image data can be challenging, due to potential projection errors or occlusions when mapping frontiers to image space. Besides, frontiers may not clearly associate with important features in the image. For example, if there is a distant opening in the trees but the local costmap might not yet reach the treeline. Then the costmap frontier will be far from the treeline, so evaluating its affordability will require relating it to information in the image far from the frontier. Further, we would like to leverage pre-trained foundation models to maximize performance and many like SAM2 and DINO [6] operate in image space. For these reasons, we propose instead to learn an intermediate representation of affordable image frontiers and project them to local frontiers.

From the navigation stack's perspective, the frontiers need to be converted from image to metric space , yet projecting them into 3D demands precise long-range depth. Instead, image frontiers are projected to rays using camera intrinsics and selecting the point at a distance H along that ray. The projected image frontier is not where the image point truly is in 3D space. So to make this projection reasonable for navigation the affordable image frontier must have a clear line of sight path from the projected point at distance H to the true 3D point associated with the image frontier. This property is impossible to enforce perfectly without a prior map. Instead, we approximate it by learning a mapping from images to affordance heatmaps via automatic data labeling and human labels.

3.1 Learning Affordances from Unlabeled Videos

While we show results from learning image affordances from hand-labeled data (Appendix C), this approach is tedious and scales poorly with more data. This raises the question - *can we learn such*

Algorithm 1 LRN: Long Range Navigator

```
Require: $k$ frontier bins, start state $s_{start}$, goal $g$, goal stdev $\sigma_g$, prev stdev $\sigma_p$, EMA $\alpha$
  I - Supervised Train Affordance Backbone
  Track videos $\mathcal{D}_v$ into trajectories $\mathcal{D}_\xi$
  Convert $\mathcal{D}_\xi$ into $\mathcal{D}$ of (image, heatmap) pairs
  Train $A(s)$ on $\mathcal{D}$ via supervised learning
  II - Online Control w/ Dynamic Planning
  $s \leftarrow s_{start}$
  $\mathbf{p} \leftarrow [1]_{i=1}^{k}$
  while $s \neq s_{goal}$ do
      $\mathbf{b}_{filtered} \leftarrow$ Affordance_Backbone$(s)$
      $\mathbf{g} \leftarrow [\mathcal{N}(x_i; g, \sigma_g)]_{i=1}^{k}$
      $\mathbf{v} \leftarrow \mathbf{b}_{filtered} * \mathbf{g} * \mathbf{p}$
      $f_\pi \leftarrow \text{argmax}(\mathbf{v})$
      $\hat{a} \sim \pi_\lambda(s, f_\pi)$
      $s \leftarrow \text{Execute}(s, a)$
      $\mathbf{p} \leftarrow [\mathcal{N}(x_i; f_\pi, \sigma_p)]_{i=1}^{k}$
  end while
```

Algorithm 2 Affordance_Backbone $A(s)$

```
Require: EMA $\alpha$
  heatmap $\leftarrow$ PredictHeatmap$(s)$
  $\mathbf{b} \leftarrow$ Project(heatmap)
  $\mathbf{b}_{norm} \leftarrow \mathbf{b} / \sum_{i=1}^{k} b_i$
  $\mathbf{b}_{filtered} = \alpha * \mathbf{b}_{norm} + (1 - \alpha) * \mathbf{b}_{filtered}$
  Return $\mathbf{b}_{filtered}$
```

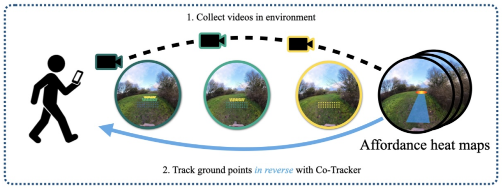
1. Collect videos in environment
Affordance heat maps
2. Track ground points in reverse with Co-Tracker

Figure 3: **Auto-Labeled affordance heatmaps.** We collect walking videos in representative environments and track ground points in reverse to extract affordance heatmaps for training LRN.

affordances from unlabeled videos? Concretely, we utilize unlabeled ego-centric videos to generate affordable image frontiers. The **key insight** is that a trajectory's endpoint should represent a good frontier from the perspective of the trajectory start. Given 3D poses, we can project a trajectory's endpoint into image space using camera parameters. But over long trajectories, small pose errors can misplace points in the sky or on obstacles. Furthermore, this restricts us to use only robot data and ignore abundant and easy-to-collect ego-centric videos - a person only needs to walk!

Since LRN only needs points in image space, we decide to forgo precise localization and instead use the video tracker model CoTracker [4]. CoTracker tracks a grid of points in image space. To get a trajectory, we run the video in reverse and select a subset of the grid right in front of the camera. Once the point becomes occluded (CoTracker provides this) we mark it as an affordable frontier and the rest of the trajectory as 0 (not affordable). Fig. 3 shows this visually. Other parts of the image are left unlabeled and do not incur a loss. We found marking the vertical column around affordable hotspots as 0 reduced false positives, particularly in the sky. We train a small de-convolution decoder to predict these heatmaps. We use an MSE loss with L2 regularization. Note, the Big Vehicle results were obtained from an earlier version of LRN trained solely from human labeled images.

3.2 Goal Conditioning

This section explains how $D(f, g)$ is calculated, and for simplicity we denote it by $\mathbf{g}$, the vector of goal-conditioned scores over all f. Given the affordance heatmaps, the scores are projected to a discrete set of angular bins. For each bin, LRN takes the sum of scores falling in that bin. Bins with total scores less than a threshold h_{thresh} are set to 0. The max is taken for overlapping bins between cameras, and scores are normalized. An exponential moving average (EMA) filter is applied with weight α to reduce fluctuations over time. The vector of k discretized bin scores is denoted as $\mathbf{b}$.

$$\mathbf{b} = [b_1, b_2, \ldots, b_k] \qquad \mathbf{b}_{\text{norm}} = \frac{\mathbf{b}}{\sum_{i=1}^{k} b_i}$$

$$\mathbf{b}_{\text{filtered}} = \alpha \cdot \mathbf{b}_{\text{norm}} + (1 - \alpha) \cdot \mathbf{b}_{\text{filtered}}$$

The goal conditioned cost function (Fig. 2) takes in $\mathbf{b}_{filtered}$ the goal angle μ_g and the previous selected heading μ_p. The goal heading and the previously selected heading each define a Gaussian centered on μ_g and μ_p. Similar to how the EMA filter reduces fluctuations in the individual scores, the previous selected heading is used to reduce fluctuations in the final selected heading. We apply these functions to the k discrete bins to obtain a goal cost vector and a consistency cost vector

$$\mathbf{g} = [\mathcal{N}(x_i; g, \sigma_g)]_{i=1}^{k} \qquad \mathbf{p} = [\mathcal{N}(x_i; f_\pi, \sigma_p)]_{i=1}^{k}$$

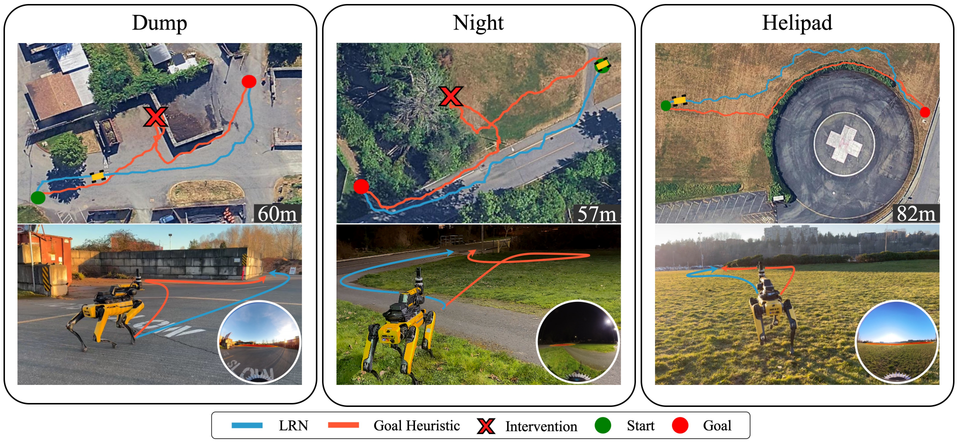

Figure 4: **GPS plots of `LRN` and Goal Heuristic Baseline on each course.** Where the Goal Heuristic blindly charges towards the goal, the `LRN` makes earlier decisions to avoid difficult terrain. The round image at the bottom right of each Spot image shows the image the robot observes and the corresponding heatmaps. Red denotes high scores. In the dump and night scenario, there is high score to the side of the wall and on the sidewalk to the bridge. For the helipad course, the `LRN` incorrectly puts some heat on the bushes also, highlighting some sub-optimal `LRN` predictions. For qualitative results of Trav. Depth and NoMaD see Appendix A.

Here, σ_g and σ_p are both fixed parameters. All the vectors are multiplied to obtain final scores. The maximum scoring angle is then selected.

$$f_\pi = \arg\max(\mathbf{b}_{\text{filtered}} \cdot \mathbf{g} \cdot \mathbf{p})$$

4 Experimental Design

We instantiate `LRN` in two setups, namely, Spot, and a Big Vehicle platform both operating outdoors. Our experiment design is motivated by empirically studying the following research questions:

[Q1.] Can the intermediate affordance representation proposed in `LRN` exhibit more-efficient, safe-navigation capability compared to other approaches? (See Sec. 5.1)

[Q2.] Considering the connection between affordance quality and overall system performance, do better affordances lead to more-efficient paths? (See Sec. 5.2)

[Q3.] How does auto-labeling versus human labeling affect affordance quality? (See Sec. 5.3) For both robot's local policies, we follow a traditional perception, planning, and control pipeline where perception creates a metric costmap, planning finds a path through it, and control tracks that path re-planning with new-information. Details can be found in Appendix E. For each robot, we collected training data in the representative biome excluding the test sites. Spot's data was auto-labeled while the Big Vehicle experiment used an early version of `LRN` trained on human labeled data. We use MobileSam [7] image encoder for Spot and SAM2 [3] for Big Vehicle. LRN performs inference on raw front/rear fisheye lens from a Insta360 camera on Spot and four time-synced cameras for Big Vehicle. On both platforms (Big Vehicle with more compute) `LRN` achieves a ~4hz inference with autonomy also running. For more details and parameters see Appendix F.

4.1 Baselines

We consider the following baselines: **1. Goal Heuristic (Big Vehicle, Spot)** plans the shortest path to a set of points at the edge of the costmap nearest the goal similar to a high uniform unknown cost. Goal Heuristic is implemented using the Goal Conditioned Head of `LRN` but given a uniform distribution for $b_{filtered}$ to focus our comparison on the image affordances from `LRN`. **2. NoMaD [8] (Spot)** is a state-of-the-art diffusion policy that predicts a trajectory from a ego-camera view and a goal image. We fine-tuned it on data from the same training site as LRN and left Spot's internal obstacle avoidance active while forgoing the local stack, as NoMaD's designed to handle perception-to-control end-to-end. NoMaD has been shown to excel when a dense topological map is available, but our off-road experiments provide only a single goal image with no prior map. This mismatch exposes the challenges of long-horizon navigation in unmapped outdoor terrain. **3. Traversability + Depth Anything V2 (Spot)** combines a visual traversability estimator trained on our datasets and Depth Anything V2 [9] monocular depth model. The intuition is that distant traversable points

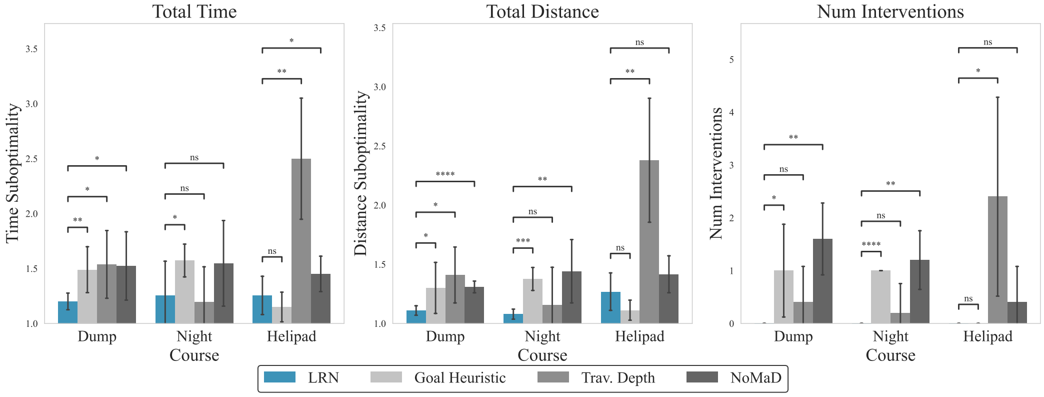

Figure 5: **Comparisons on Spot tests.** We report average and 95 % confidence intervals for 3 real world experiments with 5 runs each. Time and Distance suboptimality are with respect to human baseline runs. * denote statistical significance : $^*p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$, and $^{****}p < 0.0001$

should be hotspots. We combine the two outputs by muliplying their normalized scores to produce a heatmap which is used instead of the LRN hotspots. See Appendix D for more details.

4.2 Real World Robot Evaluation

We explain the experimental procedure for each robot. For interventions, a teleoperator intervenes when the robot is stuck in a local minimum or risks danger. The operator then guides the robot out of the local minimum and towards the goal until the visualizations indicate it will not return to the same local minimum. Interventions due to local-perception failures are excluded.

Spot We perform 5 tests per approach on Spot (60 total) and another 30 tests total for the ablation. To illustrate the challenges of a limited range metric map, we found specific scenarios in semi-urban environments that showcase the failures practitioners would encounter. We tested on three courses: dump, night, and helipad as seen in Fig. 4. **Dump:** A bug-trap wall hides the goal 60 m from the start position. The wall is outside the range of the local map until the robot gets close. **Night:** Partially lit by streetlights, the robot must travel 57 m across a river; bushes and the river block the direct route, but a wide bridge to the left offers a way over. **Helipad:** The robot must get to the other side of the helipad 82m away. The environment is an open field, but a hedge blocks the direct line to the goal. Each approach was tested for five trials per course (given weather and foot-traffic limitations). We measure Total Distance, Human Interventions, and Total Time.

Big Vehicle This is a more holistic test of LRN on a full system. Thus, we provide each approach with a single waypoint 660m away which crosses three hills two of which have dense clusters of trees that should be avoided. We use similar metrics as Spot. Given time constraints, we were only able to run each method for one trial demonstrating the system but not fully testing it.

4.3 Offline Evaluation

We test the offline performance on human-labeled test sets (330 Spot and 315 Big Vehicle images) unseen during training but from the same environments. Since no other affordance heatmap predictor exists to our knowledge we compare against Traversability + Depth Anything V2 [9]. Separate LRN models were trained for auto-labeled and hand-labeled data for each environment. To evaluate these methods we binarize the target heatmaps with a threshold of 0.15. We use Area Under the Reciever Operator Curve (AUROC), F1 score, Precision, Recall, False Positive Rate (FPR), and False Negative Rate (FNR). All metrics are [0, 1].

5 Results

5.1 [Q1.] Can the intermediate affordance representation proposed in LRN exhibit more efficient safe navigation capability as compared to other approaches?

For Spot, we report in Fig. 5 that LRN outperformed Goal Heuristic on all metrics on Dump and Night courses and was comparable on Helipad course. Compared to Trav. Depth and NoMaD, LRN outperformed on Dump and Heli mostly and saw competitive performance on Night.

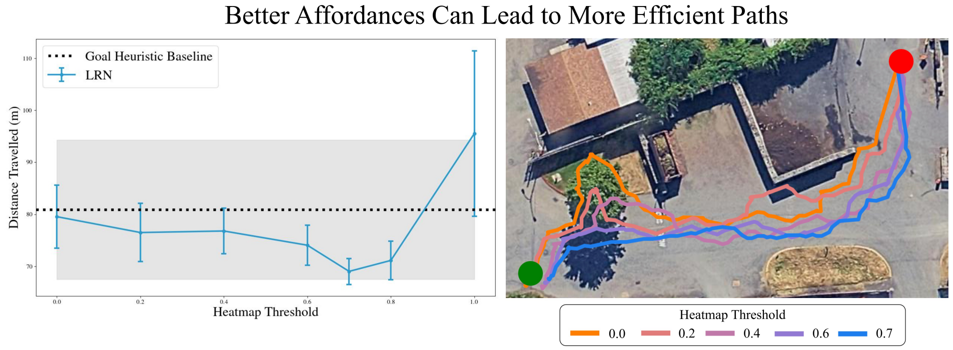

Figure 6: **Better affordances can lead to more efficient paths.** We modify the heatmap threshold affecting the affordance set size. Low threshold means LRN considers more potentially poor options i.e. everything is affordable. High threshold means very few or no options i.e. nothing is affordable. $h_{thresh} = 0.7$ gives the optimal affordance heatmaps resulting in shorter travel distances for LRN. We fit a line (not shown) from threshold 0.0 to 0.7 an find there is a correlation ($p < 0.05$) between affordance quality and traversal distance.

LRN, Trav. Depth and NoMaD see a higher total distance in Helipad compared to the Goal Heuristic due to these predictive models switching directions more frequently causing wandering. Finally, unlike all other methods, LRN needed no interventions. Qualitative analysis shows in Fig. 4 that LRN **can make earlier decisions to avoid large obstacles** compared to Goal Heuristic highlighting its longer range reasoning (See Appendix A for more qualitative results). The GPS paths for LRN visibly are more jagged, particularly in helipad due to some switching of LRN directions which slows the robot down. For the Big Vehicle, Table. 1 shows that LRN **achieves a higher average and max speed with zero interventions**. This is likely due to LRN taking a longer, more open route around dense/difficult terrain. Despite only being a single run, the result shows promise LRN can handle real-world navigation on a full system. In Fig. 7, the Goal Heuristic (orange) twice hits treelines and needs a 60 m intervention, whereas LRN (blue) skirts the trees, briefly detouring South at the end of the course (likely from a fixed σ_g), reaching the goal unaided.

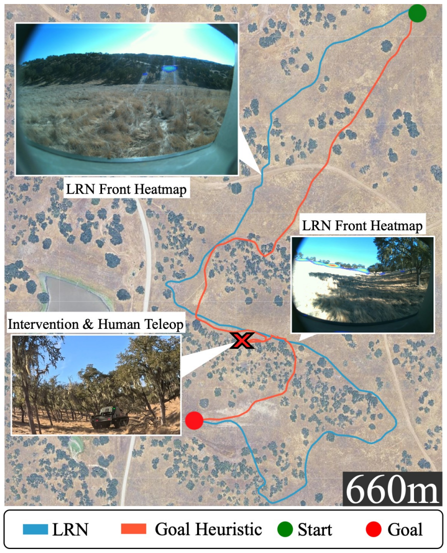

Figure 7: **Full scale demonstration**. The Goal Heuristic needed teleop after wedging into dense trees by a ditch, whereas LRN skirted the forests, missed one late gap, and reached the goal via a slightly longer route.

Algorithm	Interventions	Avg./Max Speed (m/s)	Distance (m)	Time (s)
Goal Heuristic	1	4.02 / 8.09	**941.79**	**247.00**
LRN	**0**	**4.98 / 8.66**	1435.09	289.00

Table 1: **Metrics for the Big Vehicle demonstration**.

Remark I: Through these experiments, we see LRN drives down interventions and shortens routes by spotting hazards early, improving overall navigation performance.

5.2 [Q2.] Considering the connection between the quality of the affordance model against system performance, do better affordances lead to more efficient paths?

To evaluate the connection between affordance heatmap quality and navigation performance, we run an ablation on the Dump course (Fig. 6). To vary heatmap quality, we adjust the heatmap threshold h_{thresh} from 0 to 1.0. At $h_{\text{thresh}} = 0$, most frontiers are predicted as affordable, so the robot heads directly to the goal and gets stuck in local minima like the Goal Heuristic. With $h_{\text{thresh}} = 1$, almost nothing appears affordable, so the robot oscillates between the goal and rare hotspots. Between extremes, we see how a h_{thresh} best tuned for the system translates to better navigation performance.

Remark II: By adjusting the heatmap quality via h_{thresh} we see a correlation between improved affordances and improved performance, indicating affordance qualitiy's impact on system performance. But we note, better affordances may not always lead to shorter paths as LRN is a heuristic that cannot predict what the environment will be like beyond view.

5.3 [Q3.] How does auto-labeling versus human labeling affect affordance quality?

We report offline test results shown in Table 2. We note that the performance for both Big Vehicle and Spot is better in the human labeled data. This is not surprising as any error in approximating the correct affordances in auto-labeling induces a bias thereby impacting performance. When trained with human-expert labeled data, LRN can be viewed as equipped with oracle affordance labels. That being said, we observe that both LRN hand labels and auto labels outperform Trav. Depth for all metrics suggesting auto labels can still provide a useful learning signal. This is supported by Spot test where LRN, trained only with auto-labels, outperformed other methods.

System	**Metric**	LRN Auto	LRN Hand	**Trav. Depth**
Big Vehicle	AUROC ↑	0.63	**0.84**	0.56
	F1 ↑	0.11	**0.52**	0.09
	Prec. ↑	0.08	**0.51**	0.07
	Rec. ↑	0.17	**0.52**	0.13
	FPR ↓	0.03	**0.01**	0.03
	FNR ↓	0.83	**0.48**	0.87
Spot	AUROC ↑	**0.93**	0.77	0.61
	F1 ↑	0.10	**0.32**	0.14
	Prec. ↑	0.06	**0.30**	0.14
	Rec. ↑	**0.35**	**0.35**	0.13
	FPR ↓	0.01	**0.0**	**0.0**
	FNR ↓	**0.65**	**0.65**	0.87

Table 2: **Classification metrics for heatmap backbone on test set.** Prec. and Rec. stand for Precision and Recall. We use the highest scoring heatmap thresholds for each method.

Remark III: We note that LRN's performance is dependent on the accuracy of the affordance model. Having access to perfect affordances e.g. in human-expert labels used for training in Big Vehicle induces less bias and therefore better performance. In contrast, automatic labeling of affordance labels while less burdensome can result in less accurate affordances, and therefore induce higher bias shown in offline evaluation and some wandering behavior on Spot. That being said, empirical Spot results suggest auto-labels still provide sufficient signal to improve navigation performance.

6 Related Work

Subgoal Planning ([10, 11, 12, 13, 14]): These works focus on planning to subgoals or frontiers similar to LRN. Most similar is Stein et al. [10], which predicts goal reachability and cost-to-go for subgoals. They focus on indoor environments and uses lidar input whereas LRN targets outdoor environments with camera input. Qi et al. [13] use cameras, learning navigable areas by back-projecting robot paths into image space. Unlike our work, they rely on perfect depth in simulated game environments. **Near to Far learning** ([15, 16, 17, 18, 19, 20]): These works try to extend the metric map at a lower resolution. While these methods have shown to extend the map some, they still require depth for mapping (or make a flatground assumption) which has limited range and can be occluded. **Visual navigation** ([21, 8, 22]): These works are similar in practice but target local navigation. They require topological maps for longer range navigation. We compare against NoMaD [8] on the shorter Spot tests. Further, LRN could be combined with these local methods to enable longer range reasoning when a map is not available. **Traversability Prediction** ([23, 24, 25, 26]): These works predict traversability in image space. While this task is related, traversability alone is not sufficient to find affordable frontiers. We show this via comparing with the Traversability + Depth Anything V2 baseline. **Cost Inpainting** ([27, 28, 1]): These methods attempt to in-paint cost in unknown space. Notably [28] uses a diffusion model to predict a larger map given the local costmap, but does not use sensor information besides the costmap for prediction.

7 Conclusion

In summary, we presented LRN, a novel method for thinking beyond metric costmaps to make less myopic navigation decisions. LRN achieves this by leveraging vision foundation models and an intermediate affordance representation trained solely from video data. Through extensive experiments, we demonstrated that better affordances can improve performance by making less myopic decisions, that LRN auto-labeling enables rapid and effective data collection, and reported overall improved long-range navigation as compared to multiple baselines both qualitatively and quantitatively, across two distinct platforms. **Future work** should consider how additional inputs such as the local costmap and sparse depth can improve performance when available. Further, a key question is how to use memory of past frontiers and observations to avoid dead-ends and wandering.

8 Limitations

While LRN has shown better overall long range navigation, our method is not without its limitations. We here discuss key failure modes.

First, LRN does not reason about depth explicitly. Without depth, we are implicitly assuming that the angular distance to goal from an LRN hotspot is a sufficient proxy for distance to goal. This assumption can break when two or more hotspots are equal angular distance from the goal heading but in reality, one is much closer to the goal. This appears as occasional wandering on the Helipad and Big Vehicle courses due to open environments with many hotspots in LRN predictions. That said, LRN recovered by eventually finding more direct hotspots and reaching the goal without intervention. Beyond incorporating depth, this issue can be addressed by adding a detector for wandering behavior and reducing σ_p encouraging sticking to one decision or reverting to Goal Heuristic until fluctuations in LRN stabilize. A good signal for this is non-monotonic erratic fluctuations is distance to goal, which may not always be decreasing (e.g. going further around an obstacle), but should change smoothly.

Second, LRN can exhibit switching behavior due to fluctuating heat scores. Small fluctuations in score between two very different directions causes the robot to switch back and forth. This problem motivated the EMA filter and previous heading gaussian score but we found it does not completely alleviate the issue and more exploration is needed. We would like to explore learning the goal conditioned head with history to see if it can learn to maintain consistent headings.

Third, while some heatmaps seemed reasonable in online tests, we noticed more optimism from the Spot model where it puts heat on obstacles near an opening. We attribute this to the automated labels which some have small tracking errors putting heat on the edges of openings. Future work on reducing tracking error or filtering bad labels could improve performance on this front.

Finally, LRN is a heuristic for exploring unknown space. While we show it can be an improved heuristic over other methods, all heuristic frontier approaches suffer from not truly knowing the whole environment. Thus LRN cannot guarantee improved performance because a frontier that looks good from one perspective may actually lead down an unseen bad path. Incorporating history into the LRN could help so if the robot reached a dead end it could remember a previous hotspot and backtrack to that position.

References

[1] 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 *Robotics: Science and Systems*, 2023.

[2] M. Patel, J. Frey, D. Atha, P. Spieler, M. Hutter, and S. Khattak. Roadrunner m&m – learning multi-range multi-resolution traversability maps for autonomous off-road navigation, 2024.

[3] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*, 2024.

[4] N. Karaev, I. Rocco, B. Graham, N. Neverova, A. Vedaldi, and C. Rupprecht. CoTracker: It is better to track together. *ECCV*, 2024.

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844.

[6] M. Oquab, T. Darcet, T. Moutakanni, H. V. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, R. Howes, P.-Y. Huang, H. Xu, V. Sharma, S.-W. Li, W. Galuba, M. Rabbat, M. Assran, N. Ballas, G. Synnaeve, I. Misra, H. Jegou, J. Mairal,

P. Labatut, A. Joulin, and P. Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[7] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023.

[8] A. Sridhar, D. Shah, C. Glossop, and S. Levine. NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration. *OOD Workshop Conference on Robot Learning*, 2023.

[9] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *arXiv:2406.09414*, 2024.

[10] G. J. Stein, C. Bradley, and N. Roy. Learning over subgoals for efficient navigation of structured, unknown environments. *Conference on Robot Learning (CORL)*, 2018.

[11] Y. Gao, J. Wu, X. Yang, and Z. Ji. Efficient hierarchical reinforcement learning for mapless navigation with predictive neighbouring space scoring. *IEEE Transactions on Automation Science and Engineering*, 2023. doi:10.1109/TASE.2023.3312237.

[12] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, pages 146–151, 1997. doi:10.1109/CIRA.1997.613851.

[13] W. Qi, R. T. Mullapudi, S. Gupta, and D. Ramanan. Learning to move with affordance maps. In *iclr*, volume abs/2001.02364, 2020.

[14] J. Liang, Y. Cao, Y. Ma, H. Zhao, and G. Sartoretti. Hdplanner: Advancing autonomous deployments in unknown environments through hierarchical decision networks. *IEEE Robotics and Automation Letters*, 2025.

[15] R. Hadsell, P. Sermanet, J. Ben, A. Erkan, M. Scoffier, K. Kavukcuoglu, U. Muller, and Y. LeCun. Learning long-range vision for autonomous off-road driving. *Journal of Field Robotics*, 26(2):120–144, 2009. doi:https://doi.org/10.1002/rob.20276.

[16] M. Bajracharya, B. Tang, A. Howard, M. Turmon, and L. Matthies. Learning long-range terrain classification for autonomous navigation. In *IEEE International Conference on Robotics and Automation*, pages 4018–4024, 2008. doi:10.1109/ROBOT.2008.4543828.

[17] M. Wang, J. Zhou, J. Tu, and C. Liu. Learning long-range terrain perception for autonomous mobile robots. *International Journal of Advanced Robotic Systems*, 7, 2010. doi:10.5772/7245.

[18] P. Moghadam, W. S. Wijesoma, and M. D. P. Moratuwage. Towards a fully-autonomous vision-based vehicle navigation system in outdoor environments. In *International Conference on Control Automation Robotics & Vision*, pages 597–602, 2010. doi:10.1109/ICARCV.2010.5707247.

[19] M. Nava, J. Guzzi, R. O. Chavez-Garcia, L. M. Gambardella, and A. Giusti. Learning long-range perception using self-supervision from short-range sensors and odometry. *IEEE Robotics and Automation Letters*, 4(2):1279–1286, 2019. doi:10.1109/LRA.2019.2894849.

[20] W. Zhang, Q. Chen, W. Zhang, and X. He. Long-range terrain perception using convolutional neural networks. *Neurocomputing*, 2018.

[21] D. Shah, A. Sridhar, N. Dashora, K. Stachowicz, K. Black, N. Hirose, and S. Levine. ViNT: A foundation model for visual navigation. In *Conference on Robot Learning (CORL)*, 2023.

[22] X. Meng, N. Ratliff, Y. Xiang, and D. Fox. Neural autonomous navigation with riemannian motion policy. *CoRR*, 2019.

[23] J. Frey, M. Mattamala, L. Piotr, N. Chebrolu, C. Cadena, G. Martius, M. Hutter, and M. Fallon. Wild visual navigation: Fast traversability learning via pre-trained models and online self-supervision. In *Robotics: Science and Systems*, 2024.

[24] R. Schmid, D. Atha, F. Scholler, S. Dey, S. A. Fakoorian, K. Otsu, B. Ridge, M. Bjelonic, L. Wellhausen, M. Hutter, and A.-a. Agha-mohammadi. Self-supervised traversability prediction by learning to reconstruct safe terrain. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2022.

[25] S. Jung, J. Lee, X. Meng, B. Boots, and A. Lambert. V-strong: Visual self-supervised traversability learning for off-road navigation. *icra*, 2024.

[26] L. Wellhausen, R. Ranftl, and M. Hutter. Safe robot navigation via multi-modal anomaly detection. *IEEE Robotics and Automation Letters*, 5:1326–1333, 2020.

[27] A. Shaban, X. Meng, J. Lee, B. Boots, and D. Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning (CORL)*, volume 164, pages 619–629, 2022.

[28] E. Fahnestock, E. Fuentes, P. R. Osteen, S. Ancha, and N. Roy. Learning semantic traversability priors using diffusion models for uncertainty-aware global path planning. In *IEEE International Conference on Robotics and Automation*, 2024.

[29] T. Miki, L. Wellhausen, R. Grandia, F. Jenelten, T. Homberger, and M. Hutter. Elevation mapping for locomotion and navigation using gpu. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2273–2280. IEEE, 2022.

[30] M. Likhachev, G. J. Gordon, and S. Thrun. ara^* : Anytime a^* with provable bounds on sub-optimality. In *Advances in Neural Information Processing Systems*, 2003.

A Qualitative Spot Results

Fig. 8 shows sample paths each approach took on all courses. As shown there were multiple interventions for the baselines because the robot got off course and needed to be corrected. We also see variations in performance of Traversability + Depth Anything V2 and NoMaD across courses. For example, Traversability + Depth Anything V2 does quite well in the Night course but on Helipad incurs a lot of wandering due to the more open environment.

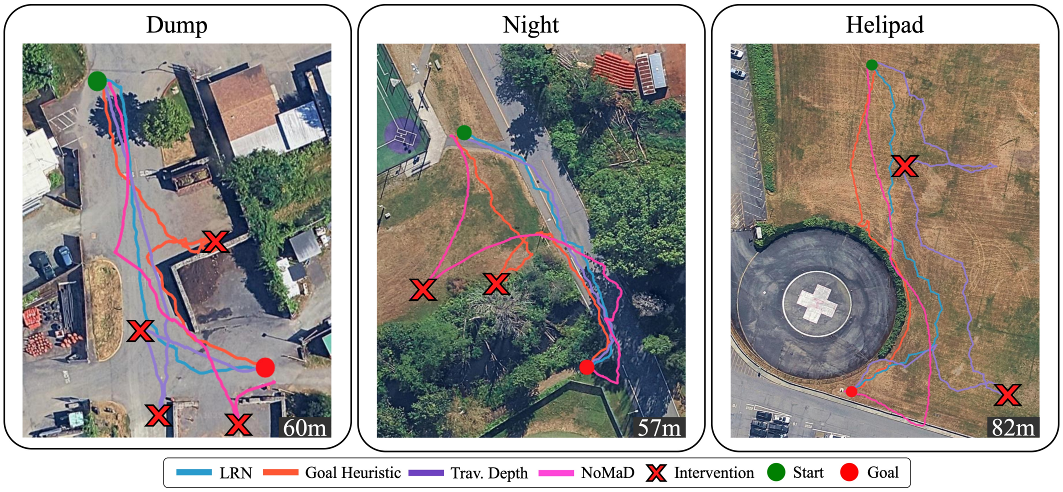

Figure 8: **GPS plots of all approaches on each course.** Many of the baselines incurred interventions for going off course and exhibit various degrees of wandering

B OOD Qualitative Results

While our method allows for fast re-training in a target environment we expect the model, leveraging vision foundation models, should generalize some to OOD conditions. Fig. 9 shows a few examples in diverse environments with various biomes (urban/forest), lighting conditions (night/day), and slopes (hilly/flat). The model was trained on data from walking around a park which had some overcast/sunny skies, flat ground, and brown winter vegetation. While the predictions in OOD settings are lower confidence as expected, they still predict promising visual frontiers highlighting the power of visual foundation models as a backbone of our method.

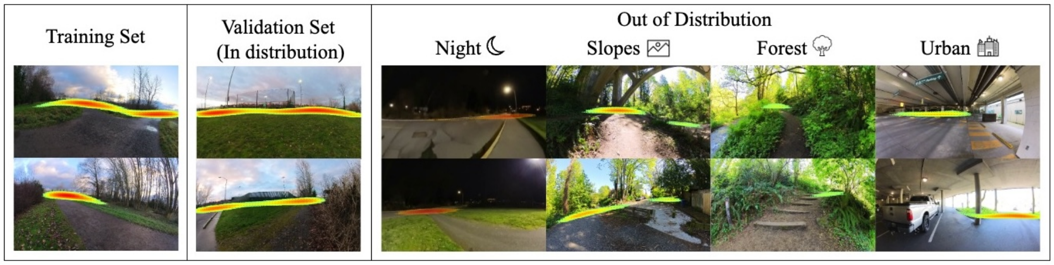

Figure 9: **OOD predicted heatmap examples.** Shown are images with a 0.5 threshold on heatmaps (nominal 0.7) to allow for lower-confident OOD predictions.

C Additional Heatmap Predictions

C.1 Big Vehicle

A sample of qualitative heatmap results can be found in Fig. 10. Traversability + Depth Anything V2 has varying levels of performance. In the leftmost image, it finds distant hills traversable and thus predicts them as a high score, not considering the uncertainty of getting to the hills. Right of that, it predicts sky as traversable and distant. This happens on and off and is due to fluctuations in depth predictions from Depth Anything V2. In the next two right images, it gets close to the correct hotspots but has no reasoning for whether the robot can continue from that point, thus marking paths leading into dense trees as traversable.

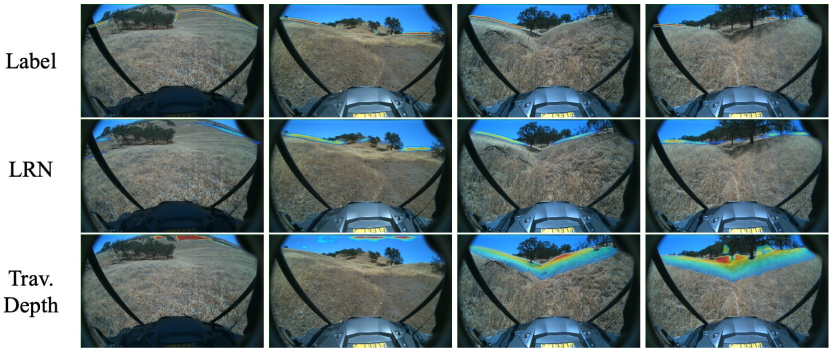

Figure 10: **Big Vehicle heatmap predictions** compared to human-labeled heatmaps on the test set.

LRN, on the other hand, gets much closer to the human labels, identifying key openings between trees. While it mostly gets the correct hotspots, it tends to smooth the heat between them more than the true labels, resulting in some heat on undesirable areas.

C.2 Spot

Qualitative heatmap results are presented in Fig. 11. As shown, Traversability + Depth Anything V2 is very sensitive to fluctuations in depth prediction, sometimes giving no hot spots in the heatmap, whereas LRN tended to be more stable. LRN also seemed overly optimistic compared to human labels, which we think contributes to some of the switching behavior in real-world tests.

D Traversability + Depth Anything V2

In this section we explain more about the Traversability + Depth Anything V2 baseline. The heatmaps are created by combining the traversability and monocular depth. We first normalize their individual scores. Depth is only normalized in regions that have a non-zero traversability. We then multiply the scores to produce a heatmap similar to LRN and threshold the values, which are then used instead of LRN hotspots. The monocular depth model we used was Depth Anything V2 base model which was the largest model we could run at a reasonable rate on the Orin AGX.

For Big Vehicle, we choose the V-Strong [25] traversability model which had been trained for the California hills environment. For Spot, a V-Strong model was not available so we trained a traversability model using the same model and training as LRN, but instead of considering only the hotspot to be 1 in the loss, we mark the whole trajectory as traversable. To improve traversability further we take a trick from V-Strong and expand the traversable region by making a SAM mask seeded from the robot's path.

Fig. 12 shows the intermediate outputs that lead to the final heatmap scores on Spot. As shown, traversability reasonably covers the space of traversable terrain but emphasizes regions directly in front of the robot, likely due to training trajectories heading straight out. Monocular depth gives

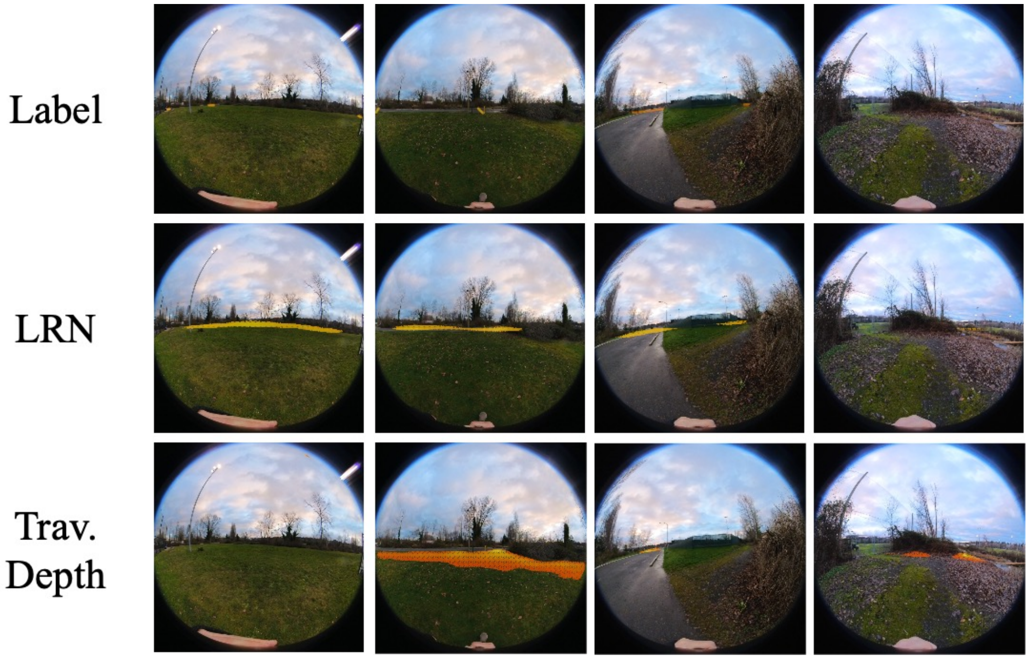

Figure 11: **Spot heatmap predictions** compared to human-labeled heatmaps on the test set.

reasonable values but becomes foggier further from the robot. Combining and thresholding the two tends to produce heatmaps that resemble LRN (e.g. column 3), but can also be overly optimistic or pessimistic. In practice, the biggest issue was fluctuations in depth prediction, leading to instability in hotspot locations.

E Local Policies

Spot We leverage Elevaton Mapping CuPy [29] a fast elevation mapping software. The local elevation map is a square with 16m width/height and created from a combination of depth cameras and an ouster OS1 lidar . The elevation map is converted to a costmap for planning by mapping slopes to cost via simple rules. We then use an ARA^* planner [30] over a lattice to plan a path through that map. A carrot point 3m ahead along the planned path is used to compute a body frame velocity which is passed to Spot's internal navigation system. The internal navigation system handles locomotion and performs some obstacle avoidance. The entire stack and LRN is run on a Jetson Orin AGX in realtime.

Big Vehicle We had an opportunity to deploy on a Big Vehicle platform. It is a 12 ton tracked vehicle equipped with three front facing cameras and one rear camera amongst other lidar and odometry sensors. The local stack in this demonstration was a heavily optimized perception planning and control stack with a circular costmap of radius 50m. The planner is a search-based planner that will plan to the goal but stop once it reaches a frontier node at the edge of the costmap within a short tolerance of the goal heading. This allows the planner some flexibility in case an obstacle is blocking the exact goal.

F LRN Setup

Spot We collected datasets in two semi-urban environments by walking around with an Insta360 camera collecting videos totaling 54 minutes. Since the points of interest are the ground, the tracked points can drift over long times. In order to have clean data, we chopped the videos into 2 minute segments. The videos were then processed by our automatic labeling pipeline to produce a dataset

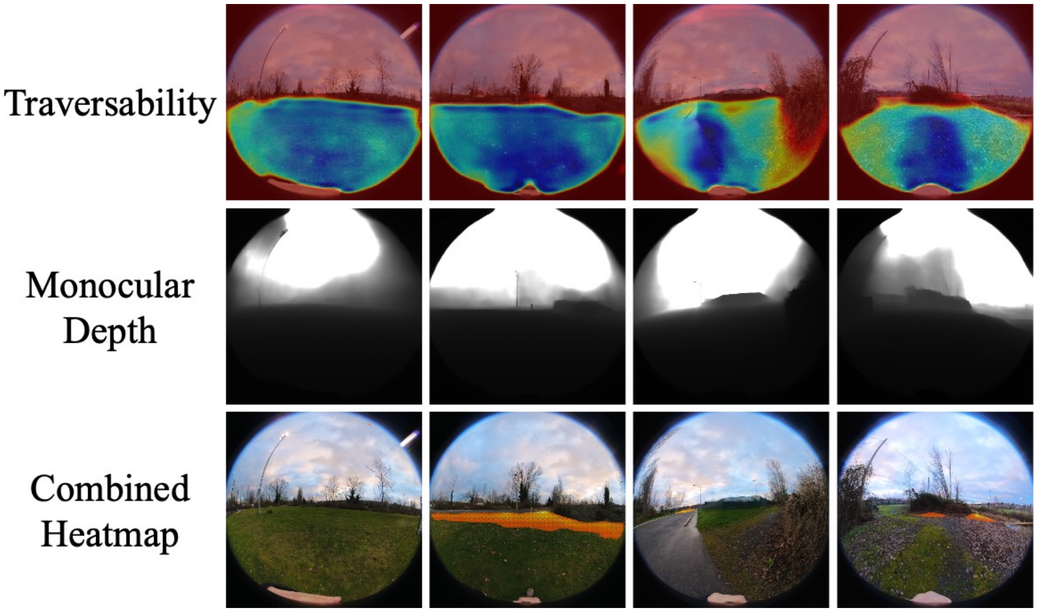

Figure 12: **Traversability + Depth Anything V2.** Blue indicates more traversable regions, while Red indicates less traversable areas. Similarly for depth, darker is closer and lighter is further. The resulting heatmap is thresholded to focus on hotspots, but this threshold can be overly conservative, sometimes leading to no hotspots (e.g., column 1).

of 92,711 heatmap labels. The training used an MSE loss with L2 weight regularization to train a 5 layer decoder network.

Big Vehicle The Big Vehicle experiment used an early version of LRN trained on human labeled data. The dataset was 1,901 human labeled heatmap images from a California oak savanna. Notably the labeled images were only front-facing camera images from a different Big Vehicle than the one we deployed on. We asked humans to label all affordable frontiers in each image. They additionally had access to future/past observations and side-cameras for context (some example images are shown in the supplementary materials). Human selected regions were positive labels and the rest of the image is assumed negative. We additionally add some Gaussian blur around positive labels as we found the smooth transitions helped with training. To improve robustness to visual variations, we further augmented the dataset by applying random color jitter, sharpness adjustments, rotations, and blur increasing the dataset size to 11,406 images. Our loss was a pixel-wise MSE loss with additional L2 regularization on the weights.

F.1 Goal Conditioned Head

Spot For the goal-conditioned level we use an angle discretization of 5 degree width bins, $h_{thresh} = 0.7$, $\alpha = 0.1$, $\sigma_g = 90$, and $\sigma_p = 110$. Additionally, to avoid walking past the goal we implement two additions. First, when the robot is within 30m of the goal it linearly reduces the σ_g based on its distance to goal. Second, when the robot is within 12m of the goal it switches to heading straight to the goal.

Big Vehicle For Big Vehicle we used a threshold of $h_{thresh} = 0.15$. The EMA filter used to reduce noise in affordance scores was set to $\alpha = 0.1$. For goal costs we used $\sigma_g = 70$ and $\sigma_p = 100$ weighting previous predictions less than the goal direction. When the robot got within 75m of the goal it would return to heading straight for the goal. There was no linear decrease in σ_g like in the Spot experiments.