Learning to Drive Anywhere with Model-Based Reannotation

Noriaki Hirose[1,2], Lydia Ignatova[1], Kyle Stachowicz[1], Catherine Glossop[1], Sergey Levine[1], Dhruv Shah[1,3]
[1] University of California Berkeley [2] Toyota Motor North America [3] Princeton University
noriaki.hirose@berkeley.edu

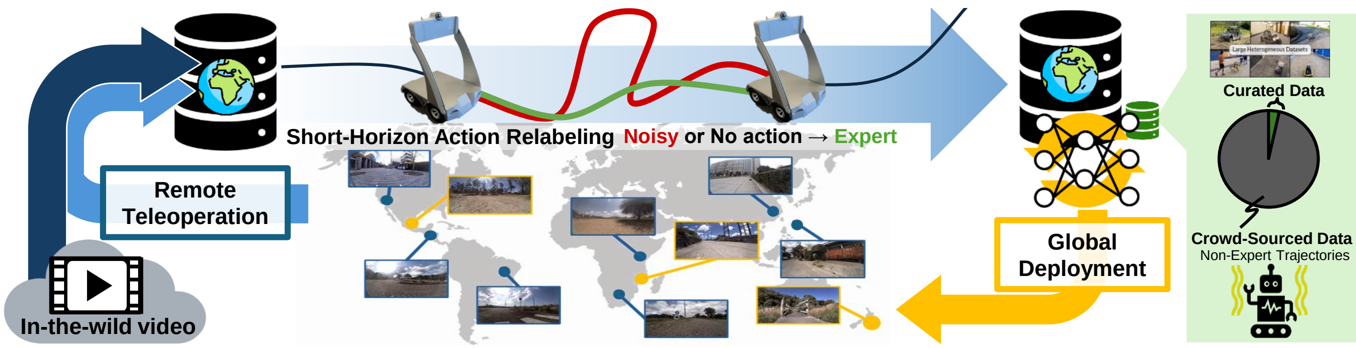


Fig. 1: We train a highly generalizable navigation policy that can control robots in a variety of conditions and be deployed zero-shot in new environments across the world. Our proposed method, **M**odel-**B**ased **R**e**A**nnotation, enables imitation learning from noisy, crowd-sourced robot data.

Abstract—**Developing broadly generalizable visual navigation policies for robots is a significant challenge, primarily constrained by the availability of large-scale, diverse training data. While curated datasets collected by researchers offer high quality, their limited size restricts policy generalization. To overcome this, we explore leveraging abundant, passively collected data sources, including large volumes of crowd-sourced teleoperation data and unlabeled YouTube videos, despite their potential for lower quality or missing action labels. We propose Model-Based ReAnnotation (MBRA), a framework that utilizes a learned short-horizon, model-based expert model to relabel or generate high-quality actions for these passive datasets. This relabeled data is then distilled into LogoNav, a long-horizon navigation policy conditioned on visual goals or GPS waypoints. We demonstrate that LogoNav, trained using MBRA-processed data, achieves state-of-the-art performance, enabling robust navigation over distances exceeding 300 meters in previously unseen indoor and outdoor environments. Our extensive real-world evaluations, conducted across a fleet of robots (including quadrupeds) in six cities on three continents, validate the policy's ability to generalize and navigate effectively even amidst pedestrians in crowded settings. We open-source our models and codes and provide supplementary videos on our project page** [1]

I. Introduction

Machine learning has demonstrated remarkable success across a range of tasks, including natural language processing [46, 4] and computer vision [33, 22, 31]. A key factor driving these advancements is the availability of large and diverse training datasets. In the field of robotics, lack of data is a major bottleneck: intentional, centralized data-collection efforts are extremely costly, requiring real-world robots and human operators, while Internet-scraped data is rarely directly applicable to the robotic domain [30, 21].

In this paper, we study the problem of developing an end-to-end navigation policy capable of generalizing to deployment in highly diverse outdoor and indoor environments. Our goal is to train a single end-to-end policy capable of navigation over hundreds of meters, while generalizing to a broad distribution of unstructured environments. Training such an end-to-end policy requires large amounts of diverse data to grant broad coverage of the set of all possible environments. Previous navigation works [38] have relied on centrally collected datasets generated by robotics researchers. While these datasets tend to be high-quality, the sum total of these datasets is on the order of dozens of hours [39] — sufficient to learn simple abilities but not scalable to learning more capable policies.

Facing this data limitation, we turn our attention to making use of more abundant sources of *passive data*. For example, crowd-sourced data, collected in a decentralized fashion by a large user base, has high state coverage and a diverse set of environments compared to what can be collected in a centralized fashion. However, the challenging nature of remote data collection with non-expert demonstrators makes it difficult to train good policies directly on these datasets. In-the-wild video is another passive data source that contains diverse environments and can enable more generalized performance. However, in-the-wild video does not have associated actions readily available for training robot policies.

To enable the use of these large amounts of cheap, scalable passive data, we propose robust model-based learning to train a short-horizon expert *relabeling policy* for generating high-quality actions connecting two nearby states. We use this short-horizon policy to annotate actions in the passive dataset, which then gives us much cleaner and higher-quality actions than in the original dataset. The outputs of this relabeling policy are then distilled into the long-horizon policy that can be conditioned on visual goals or on a future GPS waypoint for navigating over long distances.

We deploy our system in a comprehensive set of evalu-

[1] `https://model-base-reannotation.github.io/`

ations on both image-conditioned and waypoint-conditioned autonomous navigation tasks across a fleet of low-cost robots deployed globally as well as various embodiments including the quadruped robot and find that it is able to deliver strong generalized performance in a suite of tasks in six different cities across three continents.

Our primary contributions are 1) a framework to learn a well-generalized long-horizon policy by applying a short-horizon relabeling MBRA (**M**odel-**B**ased **R**e**A**nnotation) model to the passive data, 2) an instantiation of the MBRA relabeler on the FrodoBots-2k dataset and YouTube videos, yielding a strong short-horizon policy, and 3) LogoNav (**Lo**ng-range **Go**al Pose-conditioned **Nav**igation policy), a policy trained with MBRA that achieves robust goal-reaching capabilities at 300+ meter scales, even while navigating around pedestrians in crowded environments. Please see our supplemental materials for videos of LogoNav exhibiting robust driving behavior in complex long-horizon navigation settings.

II. Related work

Vision-based robot navigation has been widely explored to navigate toward goal positions given visual observations from a monocular camera. [34, 32, 13] train short-horizon policies to generate actions with access to a single goal observation. These short-horizon policies often utilize topological memory to extend the range of navigation [27]. Some works[37] use exploration with a topological memory to seek out a distant image goal, while others[36, 41] use a GPS signal for localization and navigate toward a goal provided as a 2D position in cartesian coordinates. Goal images and poses require prior access to the target environment and knowledge of the environment's geometry. Various learning methodologies such as imitation learning (IL) [34, 38, 39], reinforcement learning (RL) [41, 17, 42], and model-based learning (MBL) [13, 14] have been explored for training goal-conditioned vision-based policies on publicly available robot datasets.

These methods require a sequence of image observations and corresponding actions parsed from accurate wheel odometry [13, 12], GPS [37], and other reliable sensors. These datasets are collected via intentional, centralized teleoperation efforts with the downstream goal of training a navigation policy and, therefore, contain goal-directed trajectories. Collecting data of this sort at a global scale would require a massive unified effort that would be costly and time-consuming.

Robot learning with passive data. Visual SLAM [25] and inverse dynamics models [3] can be used to estimate trajectories for first-person videos, allowing us to train policies that use these trajectories as approximations of robot actions from action-free and non-robot data. While visual SLAM and its successors [5, 28, 43, 44] offer good local trajectory estimation, its accuracy relies on having consistent, good visual features in the image view.

Robotic foundation models (RFMs) [39, 40] trained with IL on curated data can address an embodiment gap issue and augment passive data sources with consistent robotic actions [16]. However, current RFMs still lack coverage of diverse environments and cannot leverage passive data with noisy action labels during training. To address these issues, we learn an expert relabeling MBRA policy with MBL to better approximate reasonable robot actions. Since MBL is robust to noisy action labels during training, we can train the MBRA policy with passive data sources and use it to reannotate large amounts of passive data with a smaller embodiment gap. We can then train a long-horizon goal pose-conditioned navigation policy, LogoNav. LogoNav can successfully perform a diverse set of long-distance navigation tasks and demonstrates an explicit advantage against the baseline policies.

III. Learning Short-Horizon Relabeling Policies with Model-based Learning

In this paper, we focus on learning long-horizon navigation policy from a highly diverse but suboptimal passive dataset $\mathcal{D}_n$. This requires us to train policies that can predict actions that are better than those found in the original dataset. We assume access to a smaller *clean* dataset $\mathcal{D}^*$ that contains high-quality behavior but with $|\mathcal{D}^*| \ll |\mathcal{D}_n|$.

While observations in $\mathcal{D}_n$ might represent high state coverage, the short-term behavior present in the data is highly suboptimal, both because the actions themselves are inaccurate due to state estimation errors and because the heterogeneous population of human operators has widely varying levels of skill when driving. To address this, we learn the MBRA policy, using any method capable of learning goal-conditioned policies, as the step 1 in Fig 2, and we choose to use MBL due to its robustness to noisy and suboptimal data. This model should provide a series of better approximate actions $\{a_i^s\}_{i=0...N-1}$ linking two observations O_c and O_g. Then, the long-horizon navigation policy is trained to imitate the relabeled actions from MBRA, as shown in step 2 of Fig. 2.

A. Learning a Short-Horizon Relabeling Policy, MBRA

We train a policy $\{a_i^s\}_{i=0...N-1} = \pi^s(O_c; O_g)$, which we call the MBRA policy, to infer the optimal actions occurring between the current observation O_c and the goal observation O_g via the model predictive control(MPC)-inspired learning approach. Our approach directly optimizes an objective function in the counterfactual space instead of imitating the original action labels to train MBRA on the joint dataset $\mathcal{D}_n \cup \mathcal{D}^*$.

We use the following model-based objective for learning a relabeler π^s, following Hirose et al. [14]:

$$\min J_{mbl} := \sum_{i=0}^{N-1} (s^{ref} - s_i^s)^2. \quad (1)$$

In Equation 1, s^{ref} is the target state and $\{s_i^s\}_{i=0...N-1}$ are the estimated states at each step. s_i^s is defined by three components, $[\hat{p}_i, \hat{c}_i, \Delta a_i^s]$, to encourage the policy to smoothly move toward the target goal pose p^g without collision. Here $\hat{p}_i$ is the i-th virtual robot pose, $\hat{c}_i$ is the estimated collision state at i-th virtual robot pose $\hat{p}_i$ (where zero indicates no collision), and Δa_i^s indicates the action difference, $a_{i+1}^s - a_i^s$. Accordingly, we define s^{ref} as [p_g, 0.0, 0.0].

The states $\{s_i^s\}_{i=0...N-1}$ are calculated by computing rollouts through a differentiable dynamic forward model f (in

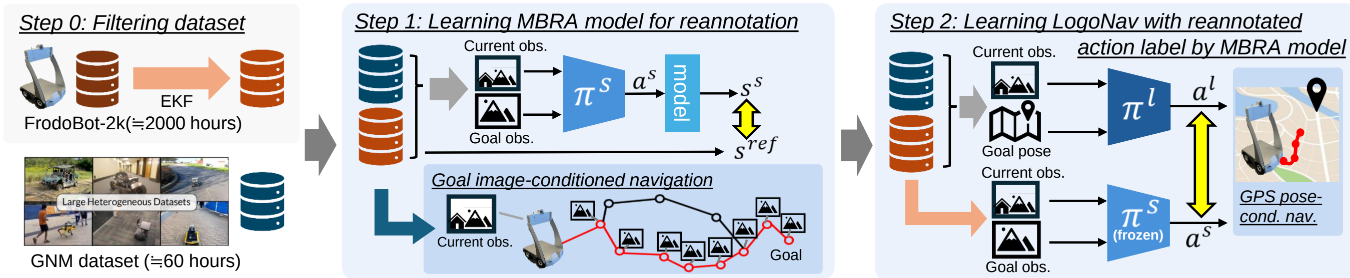


Fig. 2: **Overview of MBRA.** We propose a two-step process: In the first stage, we train a short-horizon reannotation policy with a robust MBL approach on the noisy dataset, which can be used for short-horizon image-conditioned navigation and which we leverage to relabel the noisy dataset with improved action labels. In step 2, we train a long-horizon navigation policy with the generated action labels.

this case, the unicycle model). The forward model [11] considers the current observation O_c and generated actions $\{a_i^s\}_{i=0 \dots N-1}$ from the short-horizon MBRA policy, π^s:

$$\{s_i^s\}_{i=0 \dots N-1} = f(O_c, \{a_i^s\}_{i=0 \dots N-1}), \qquad (2)$$

where O_c is the current observation. While the states $\{s_i^s\}_{i=0 \dots N-1}$ are conditioned on actions $\{a_i^s\}_{i=0 \dots N-1}$ and f is differentiable, we can calculate the gradient of π_s to minimize J_{mbl} in each training step and learn π_s by repetitively update the parameters of π_s similar to other machine learning approaches. Note that we freeze f while training π_s.

Our approach differs from pure IL in that it does not imitate the noisy action commands at each step. While the target pose p_g is relatively far from the robot current pose, the negative impact of suboptimal or noisy actions is mitigated. Therefore, MBRA is robust to data with these properties. Implementation details and discussion are included in the later sections.

B. Learning a Long-Horizon Navigation Policy, LogoNav

After reannotating the passive dataset with the short-horizon relabeling expert, MBRA, we have a clean set of action labels that can be distilled into a more complex long-horizon navigation policy. We want a navigation policy π^l to predict actions as follows:

$$\{a_i^l\}_{i=0 \dots N-1} = \pi^l(O_c, p_g), \qquad (3)$$

where O_c is the current observation and p_g is the 2D relative goal pose from the robot coordinate. Notably, p_g is at least 10 times further than the usual goal pose for the short-horizon navigation policy, on the order of 50 meters, compared to the previous 3 meters. We train this IL policy on the reannotated action commands $\{a_i^s\}_{i=0 \dots N-1}$ from the short-horizon navigation policy,

$$\min J_{il} := \sum_{i=0}^{N-1} (a_i^s - a_i^l)^2. \qquad (4)$$

By imitating the cleaned action commands linking O_c and O_g, our long-horizon policy, LogoNav , can learn reasonable conventions such as staying on paths, avoiding collisions, and not disturbing pedestrians, which is representative of the "good" navigation behavior modeled by the MBRA policy. Note that we only reannotate the noisy passive data, using accurate raw action labels from $\mathcal{D}^*$. We freeze π^s while training π^l.

TABLE I: Survey of public datasets for learning vision-based navigation policies in real-world.

Dataset	Policy	hour	Sensors
KITTI odom [10]	teleop	0.7	RGB, 3D LiDAR, GPS
NCLT [6]	teleop	34.9	RGB, 3D LiDAR, odom, GPS, IMU
GO Stanford [12, 13]	teleop	10.3	RGBs, odom
FLOBOT [47]	auto	0.46	RGBD, 3D and 2D LiDAR, odom, IMU.
RECON [37]	auto	25.0	stereo RGBD, 2D LiDAR, GPS, IMU
JRDB [26]	teleop	1.1	stereo RGBD, 3D and 2D LiDAR, IMU
SCAND [20]	teleop	8.7	RGBD, 3D LiDAR, odom
TartanDrive [45]	teleop	5.0	RGBD, GPS, IMU
HuRoN [15]	teleop	75.0	RGBs, 2D LiDAR, odom, bumper
FrodoBots-2k	teleop	2000	RGBs, GPS, IMU, odom,
FrodoBots-2k-filtered	teleop	700	RGBs, filtered 2D localization

IV. Implementation

We provide the implementation details of our navigation system, covering the dataset used, network and objective design, and hyperparameter settings.

A. Passive Dataset

We evaluate our approach with two different datasets, a crowd-sourced robotic dataset, FrodoBots-2k, and an in-the-wild YouTube video dataset described in [16]. We focus on results using FrodoBots-2k to demonstrate the effectiveness of our proposed approach and additionally evaluate its capabilities on the YouTube video dataset.

Crowd-sourced robotic dataset: The FrodoBots-2k dataset [1] includes 2000 hours of data from over 10 cities and was collected as part of FrodoBots AI, where users explore locations worldwide by teleoperating robots to reach target positions. The FrodoBots-2k dataset is significantly larger than other publicly available datasets for vision-based navigation tasks. As shown in Table I, the full version of the FrodoBots-2k dataset is more than 25 times larger than other datasets and includes a diverse set of real robot trajectories teleoperated by humans.

While the scale and diversity of this dataset are enticing, the inexpensive hardware setup of the robots and crowd-sourcing approach result in significant noise. Since the sensor measurements cannot be reliably used to estimate robot poses, policies trained on the raw actions have poor performance. The main factors of the noisy action labels are 1) robot inconsistencies and corresponding user adjustments, 2) low-cost GPS and IMU, 3) inevitable wheel slips during turning,

4) robot vibration during turning, and 5) system time delay. The details of the robot system and the noisy action labeling are shown in the appendix.

In-the-wild YouTube videos: We also evaluate the ability of MBRA to enable the use of non-robot data. We reannotate 100 hours of action-free in-the-wild YouTube videos, listed in [16], and train a version of LogoNav with the generated actions. These videos include inside and outside walking tours from 32 different countries across varying weather conditions, time of day, and environment types (urban, rural, etc.).

In addition to the passive data, we use the public expert datasets RECON [37], GO Stanford [12, 13], CoryHall [18], TartanDrive [45], HuRoN [15], Seattle [35], and SCAND [20] with accurate action labels. The weighting of each dataset is the same as the original GNM dataset mixture.

B. Pre-Processing and Filtering

As shown on the leftmost side of Fig. 2, we use a classical state estimation pipeline to get better coarse robot pose estimates for FrodoBots-2k. We use a smoothing system based on a bidirectional Extended Kalman Filter (EKF) [19] to fuse raw actions with wheel speed measurements, GPS location, and compass heading (all of which are noisy) to get a smoothed estimate of the robot's position. We also filter out data where the robot is paused for a long time to prioritize learning desirable behaviors. The cleaned and filtered data consists of approximately 700 hours of real-world navigation trajectories collected worldwide, which is still an order of magnitude larger than any currently available visual navigation dataset as shown in Table I. While the EKF-based state estimation helps produce a less noisy action estimate [7], the signal remains too noisy for direct training.

C. Training Details

We describe the training settings for both our short- and long-horizon policies.

Short-horizon relabeling policy: Since the robot system has L steps system delay [9, 2] when operating remote robot via internet, we design our objective and network architecture to account for system delay to prevent overshooting or oscillating around target trajectories. Inspiring the previous works of model predictive control [24, 23], we consider the robotic states with the previous action commands $\{a_i\}_{i=-L\ldots-1}$ to genrate the actions $\{a_i\}_{i=0\ldots N-1}$.

In training, we set the observation and action rate for trajectory sampling at 3 Hz for consistency with the GNM dataset. During training, we randomly select an image frame from the entire dataset as the current observation, and then randomly select a goal frame from up to $N_g = 20$ steps (about 7 seconds) in the future. This short distance to the goal lets us learn precise labels to reannotate the action between O_c and O_g. A more detailed description of our MBL configuration is available in the appendix.

Long-horizon navigation policy: For long-horizon navigation, we use a larger $N_g = 100$ to sample a goal position up to 33 seconds into the future. We reannotate actions with the short-horizon MBRA model to get high-quality action labels for the FrodoBots-2k dataset. This process yields action labels with a chunk size of $N = 8$ steps. We train on the IL objective outlined in Eq. 4 using the same parameters and settings as the short-horizon navigation policy otherwise.

Following [39], we design the network structure for each policy. The details are shown in the appendix.

V. Evaluation

To evaluate LogoNav and the impact of MBRA relabeling in the real world, we focus our experiments on answering the following questions:

Q1 Can we apply MBRA to learn an effective long-horizon navigation policy?
Q2 Can we use MBRA for action-free in-the-wild data?
Q3 Is MBL be better than IL for learning relabelers?

A. Evaluation Setup

We concretely describe both short-horizon and long-horizon navigation tasks we evaluate our method on along with their associated baselines.

Short-horizon navigation policy: Our short-horizon navigation policy can navigate the robot toward a goal up to 3 meters away, so we use a topological memory to enable the robot to navigate to further goal positions, similar to other vision-based navigation approaches [34, 13, 39]. To collect this goal loop, we teleoperate the robot and record image observations at a fixed frame rate of 1 Hz. To deploy the policy, we start from the initial observation and continuously estimate the closest node from the topological memory. At each time step, we estimate the current node following [39, 40] and feed the image from the next node as the goal image O_g to our policy to compute the next action.

Long-horizon navigation policy: Our long-horizon navigation policy can navigate to goals between 25-100 meters from the initial robot pose. We rely on GPS to get robot positions and specify goals. We evaluate longer trajectories by setting multiple subgoals at intervals of approximately 80 meters apart. At every time step, we calculate the current relative goal pose p_g on the way to the next goal pose. When $|p_g| < 5.0$ m, we consider the goal reached and update to the next subgoal for a longer trajectory.

We use the same robotic platform, Earth Rover Zero (ERZ), that was used to collect the FrodoBots-2k dataset for our main evaluation results and different robot platforms for cross embodiment analysis. See the A appendix for more details

B. Long-horizon Navigation Policy (LogoNav): GPS Goals

To answer **Q1**, we evaluate the long-horizon navigation policies trained with MBRA and five baselines: a NoMaD-like policy and IL policies trained on GNM + FrodoBots-2k with three different annotation approaches, 1) filtered action by EKF, 2) VPT, and 3) Multi-step VPT. Details of the baselines are shown in the appendix. We select 7 outdoor locations and evaluate each policy 3 times for each goal. In Table II, we show the goal success rate and the coverage rate for each method. The coverage rate is the ratio of the distance reached by the robot to the distance of the target goal

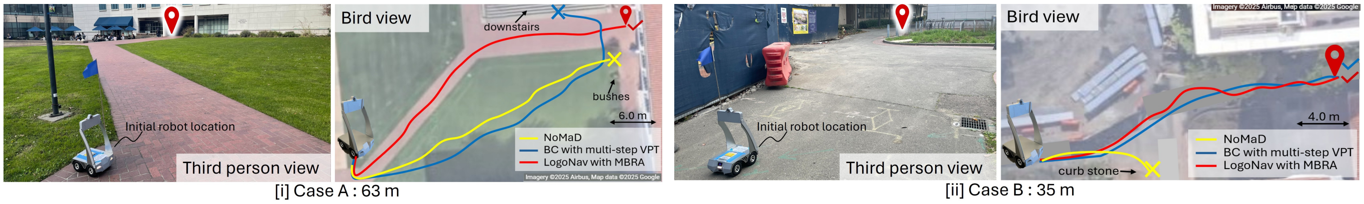


Fig. 3: **Policy rollouts for goal pose-conditioned navigation with long-horizon policies.** Our policy trained with MBRA can keep traveling on the road and arrive at the goal pose.

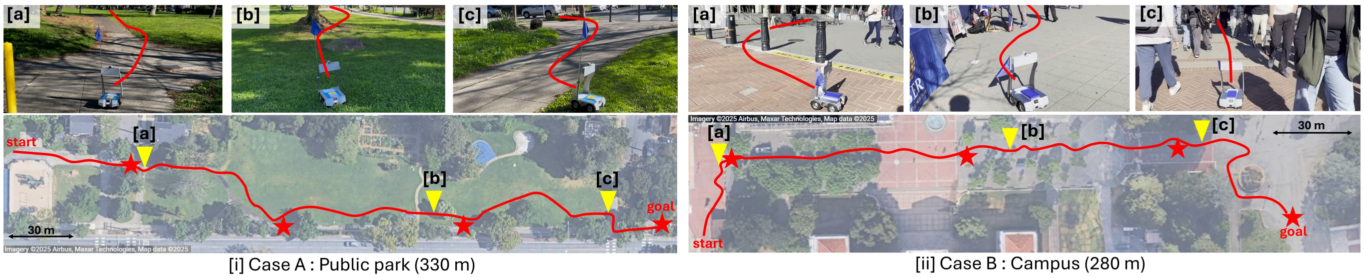


Fig. 4: **Long-horizon navigation with multiple subgoals** The ERZ can travel for about 20 minutes without collision and arrive at the goal about 300 m away. The red stars indicate the subgoal locations.

TABLE II: Evaluation of LogoNav on long-horizon pose-conditioned navigation tasks. "GS" and "COV" indicate the goal success rate and the coverage rate, respectively.

	FrodoBots-2K Data		Score	
Policy	usage	Relabeler	GS	COV
NoMaD [40]	GNM only	-	0.333	0.471
Behavior Cloning [39]	✓	EKF [7]	0.286	0.624
	✓	VPT [3]	0.000	0.071
	✓	multi-step VPT [3, 39]	0.619	0.752
LogoNav	✓	MBRA	**0.857**	**0.924**

TABLE III: Evaluation of MBRA on action-free in-the-wild YouTube videos. "GS" and "SC" indicate the goal success rate and the subgoal coverage rate.

Dataset		MBRA	
GNM	YouTube video (LeLaN)	GS	SC
✓	✗	0.500	0.680
✓	✓	0.875	0.909

pose before it fails. Our policy with MBRA shows stronger performance than the five baselines for both goal success rate and coverage rate. Since our MBRA can be trained on the FrodoBots-2k dataset, MBRA can give more reasonable annotation for the FrodoBots-2k dataset and enables us to have better LogoNav. Later, we conduct more detail investigation about the relabelers for **Q3**.

Figure 3 shows the third-person view at the start position and the robot trajectories on a bird-eye-view map in two scenes. Our policy distilled from MBRA actions was the only one to successfully navigate to the distant goal pose in both scenes, making a sharp left turn at the start to stay on path in case A. In contrast, both NoMaD and multi-step VPT could not execute this action, failing by colliding with bushes or requiring interventions to avoid falling down stairs. To show the capability of MBRA in long-horizon navigation, we provide several subgoals, specified by latitude, longitude, and azimuth angle values, at intervals of approximately 80 meters, and evaluate LogoNav with MBRA on traversing these subgoals in two different scenes. As shown in Fig. 4, our navigation system with our policy enables us to navigate the robot toward a goal 300 meters away without collision, even in human-occupied spaces.

Moreover, we deploy LogoNav on two more robotic embodiments, including VizBot [29], a small Roomba-like robot, in an indoor setting, and the Unitree Go1 quadruped robot in an outdoor setting. We achieve strong goal-reaching behavior from up to 100 meters away, highlighting the policy's generalization ability. The behaviors of these embodiments are shown in the supplemental materials.

C. *Training navigation policies on in-the-wild video*

For **Q2**, we evaluate the capability of MBRA with action-free in-the-wild video. We use MBRA policy to generate the action labels for the in-the-wild videos and train the short-horizon visual navigation policy conditioned on goal images, $\{a_i^s\}_{i=0 \dots N-1} = \pi^s(O_c, O_g)$. During training, we use the same objective as Eq. 4 to imitate the action labels generated by MBRA. We train two goal image-conditioned policies with the GNM dataset alone and GNM + in-the-wild videos to evaluate how well our MBRA enables us to close the embodiment gap between robot and in-the-wild data.

To evaluate the performance in a variety of situations, we collect the topological memories on four indoor trajectories and four outdoor trajectories and deploy the policies with the ERZ. The distance from the initial node to the goal node is between 10.0 m and 31.0 m. As shown in Table III, the policy trained with the MBRA-annotated in-the-wild video data has an explicit advantage compared to the policy trained only on the GNM dataset. Although the training dataset does not contain the data from the target robot, ERZ, in our evaluation,

TABLE IV: Comparison of MBRA and multi-step VPT.

Dataset		Multi-step VPT		MBRA	
GNM	FrodoBots-2k	GS	SC	GS	SC
✓	✗	0.500	0.680	0.875	0.960
✓	raw label	0.000	0.308	0.500	0.777
✗	filtered label	0.125	0.377	0.875	0.940
✓	filtered label(1%)	0.750	0.887	0.875	0.889
✓	filtered label	0.375	0.576	**1.000**	**1.000**

TABLE V: Evaluation of the goal image-conditioned navigation at 6 countries.

Policy	Dataset	GS	SC
Multi-step VPT [39]	GNM	0.500	0.736
Multi-step VPT [39]	GNM + FrodoBots-2k (1%)	0.792	0.906
MBRA	GNM	0.833	0.899
MBRA	GNM + FrodoBots-2k (full)	**0.958**	**0.983**

we achieve a high success rate by training the policy with diverse in-the-wild video data.

D. Evaluating MBRA on effectively using crowd-sourced data

To answer **Q3**, we compare MBL and IL, which correspond to the relabellers MBRA and multi-step VPT that demonstrated the strongest performance in Table II. We train several relabelers with different data setups for each learning method and deploy these relabelers as the short-horizon navigation policy in the same eight environments and topological memories as in the previous section to more thoroughly explore the capabilities of each of these methods.

Table IV shows the goal success rate and the subgoal coverage rate for each policy. We find that IL completely deteriorates the performance by imitating the noisy raw action of FrodoBots-2k dataset. The EKF filtering helps a bit, and incorporating the GNM data improves performance as well. In our data ablation study, we find that GNM + only 1% FrodoBots-2k dataset can help to improve the performance. However, IL cannot effectively leverage the entire FrodoBots-2k dataset. Besides, MBL enables us to scalably learn our MBRA from the noisy data. MBRA trained on GNM + filtered 100% FrodoBots-2k dataset successfully arrived at the goal position in all cases.

In the final experiment, we aim to assess the generalization capabilities of MBRA policies. To this end, we deploy the short-horizon navigation policy on robots in diverse environments across 6 countries: USA, Mexico, China, Mauritius, Costa Rica, and Brazil. In total, we collect 24 topological graphs and evaluate each target trajectory. To the best of our knowledge, we are the first to conduct a global evaluation for visual navigation. We evaluate multi-step VPT and MBRA policies trained with and without the FrodoBots-2k dataset. Findings are summarized in Table V. MBRA had better performance for both goal reaching and subgoal coverage than multi-step VPT.

Comparing multi-step VPT and MBRA The key differences between multi-step VPT and MBRA are highlighted in Fig. 5.

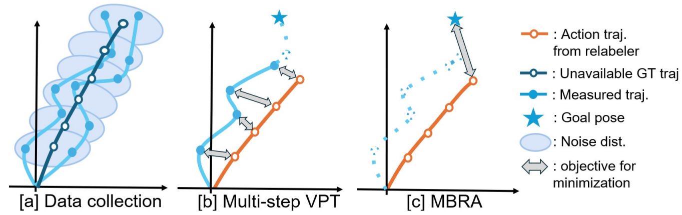


Fig. 5: **IL and MBL on the noisy dataset.**

If the noise in the dataset comes from a Gaussian distribution, as shown in Fig. 5[a], the action labels in the data are expected to be inconsistent for observations along the GT trajectories. And it is expected that the GT trajectories itself are noisy due to data collection by non-expert teleoperators. Imitating such a noisy trajectories as shown in Fig. 5[b] is impractical because it will be heavily skewed by intermediate inconsistent noisy actions, leading to incorrect reannotations. In contrast, MBRA is more robust to noisy data because it prioritizes the final goal pose, which is typically further from the individual positions and therefore can leverage all FrodoBots-2k dataset to train MBRA, leading better reannotation for FrodoBots-2k dataset.

VI. Conclusion

MBRA allows us to leverage large amounts of low-quality passive data for learning long-horizon navigation policies, making affordable passive data useful for training broadly generalizable and capable visual navigation policies. MBRA trains a short-horizon image-conditioned navigation policy to reannotate imprecise trajectory action labels. Then, the reannotated labels are used as ground truth to train a goal-pose conditioned long-horizon policy, which learns reasonable conventions such as staying on paths and avoiding collisions. We evaluate our method on robots in 6 countries across multiple continents and observe significant improvements over baselines. These results indicate that our model provides a broadly applicable, capable, and generalizable solution for visual navigation.

Limitations: Our approach to reannotating noisy crowd-sourced data in the long-horizon navigation setting works well but leaves room for improvement. In the model-based approach, we may sometimes generate unreasonable actions because of inaccuracies in the robot model. While we find the model-based approach to generally outperform the imitation-based relabeler, it does require some strong conditions on the model itself that could prove difficult to translate to more complex tasks like manipulation. One axis of future improvement is developing a more accurate differentiable model by incorporating more accurate 3D geometry, environment semantics, and dynamic object behaviors, such as pedestrian behavior [15]. It would also be helpful to consider not only goal reaching but also to incorporate humans' preferences into the objective design, particularly when navigating in crowds or in settings where semantic conventions are important (e.g., not driving on grass when it is inappropriate). While our model inherits some semantic behaviors (like staying on paths) from the tendencies exhibited by the human operators in the data, such preferences are not enforced explicitly.

Acknowledgment

This research was supported by Berkeley AI Research at the University of California, Berkeley and Toyota Motor North America. And, this work was partially supported by DARPA TIAMAT, ARL DCIST CRA W911NF-17-2-0181, NSF IIS-2246811, and NSF IIS-2150826.We thank Frodobots AI for providing the robot hardware and computational resources for our evaluations.

Appendix

Model-based learning considering time delay: In training MBRA policy, we introduce a novel objective design considering the variable system delay to consistently learn the robotic behavior during the delay period. In the system with L steps delay, the robot has to act by the previous action commands $\{a_i^s\}_{i=-L\ldots 1}$ for L steps. In other words, after L steps running $\{a_i^s\}_{i=-L\ldots 1}$, the robot can act according to the action commands a_0. Hence, without taking into account the robotic behavior of $\{a_i^s\}_{i=-L\ldots 1}$ in training, the generated action command a_0 causes overshooting and/or oscillation against the target trajectories. We train MBRA by considering the past action commands $\{a_i^s\}_{i=-L\ldots 1}$ to generate more consistent action commands.

We feed the previous action commands $\{a_i\}_{i=-L_{max}\ldots -1}$ and a randomly selected virtual delay step L $(\leq L_{max})$ into our network in addition to the visual observations. $L_{max}(=6)$ is the maximum step number of the assumed time delay. Then we calculate the following objective considering the robotic state $\{s_i^s\}_{i=-L\ldots N-1}$ for $L+N$ steps

$$\min_{\theta_s} J_{mbl}(\theta_s) := \sum_{i=-L}^{N-1} (s^{ref} - s_i^s)^2, \quad (5)$$

where we estimate the states by using the previous action commands $\{a_i^s\}_{i=-L\ldots -1}$ and the generated action commands $\{a_i^s\}_{i=0\ldots N-1}$ from our policy as $\{s_i^s\}_{i=-L\ldots N-1} = f(O_c, \{a_i^s\}_{i=-L\ldots N-1})$. In our implementation, s_i^s includes the robot virtual poses p_i and the number of collision points c_i in the estimated virtual 3D environment. The distance between the goal pose p_g and each p_i is penalized, and $c_i = 0$ or collision-free behavior is encouraged by setting $c^{ref} = 0$. Details are shown in the original paper of ExAug [14]. By minimizing J_{mbl}, we can train π^s while considering the behavior during the uncontrollable time delay by $\{a_i^s\}_{i=0\ldots N-1}$.

In inference, we decide L depending on the system architecture and feed the generated a_0 to control the robot. In our implementation, we set $L=2$ for the ERZ and $L=0$ for other robots . A more detailed description of our MBL configuration is available in the original paper of ExAug [14].

Network Architecture: Figure 6 shows the network architecture of both our image-goal conditioned and goal-pose conditioned models. For short-horizon navigation policy π^s, we concatenate the current observation O_c and the goal observation O_g and generate a goal-conditioned embedding with EfficientNet-B0. In addition, we concatenate the image observation history $\{O_i\}_{i=-M\ldots 0}$ and generate a history embedding with EfficientNet-B0. We pass in these visual features, the system delay L and the previous action commands $\{a_i^s\}_{i=-L\ldots -1}$ to a set of Transformer and fully connected MLP layers to produce a sequence of 2D poses $\{a_i\}_{i=0\ldots N-1}$.

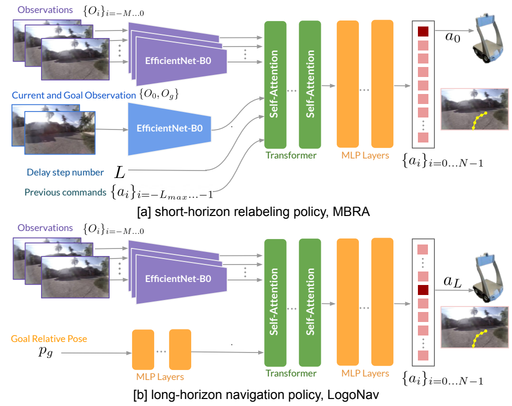


Fig. 6: **Network architecture.** In addition to the observations, We feed the delay step and the previous actions to consider the delay in the MBL objective. For the LogoNav policy, we replace the visual encoder for O_c and O_g with the MLP layers for p_g.

For the long-horizon navigation policy π^l, we replace the visual encoder for O_c and O_g with MLP layers for the 2D goal pose p_g and no longer include system delay length L and previous actions $\{a_i^s\}_{i=-L\ldots -1}$. Instead of considering the delay during training, we use the L^{th} step of the output during inference, similar to [8] and [39].

Evaluation platform: We use three robot platforms, FrodoBot "Earth Rover Zero" (ERZ), VizBot, and the quadruped robot Go1 for our cross-embodiment analysis.

FrodoBot "Earth Rover Zero" (ERZ): The FrodoBot "Earth Rover Zero" (ERZ), shown in Fig. 7, is the platform used both for dataset collection and our navigation policy deployment. The ERZ is available at \$349 and includes a host of sensors such as front and back side cameras, GPS, an IMU unit including gyroscope, accelerometer and compass sensors, and wheel velocity sensors in all four wheels.

However, the inexpensive hardware setup of the robots and crowd-sourcing approach result in significant noise. The main factors of the noisy action labels are 1) robot inconsistencies and corresponding user adjustments, 2) low-cost GPS and IMU, 3) inevitable wheel slips during turning, 4) robot vibration during turning, and 5) system time delay. The details of the robot hardware and the reason of the noisy action labeling are shown in the appendix.

The operators can introduce action label noise by intuitively making adjustments for windy conditions, controller drift, and imbalanced wheels, which are not reflected by the visual observations of the robot. On top of that, the robot turns by using friction to slip in place, making the turning behavior highly dependent on the road condition and robot pose estimation from wheel speed sensors inaccurate. This slippage also causes significant vibration of the robot chassis that inhibits the ability to use IMU measurements for action estimation. In addition, all signals are sent between the robot and the workstation via

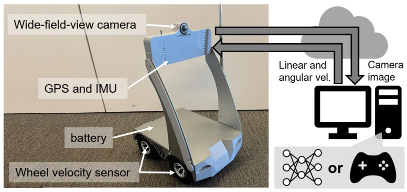
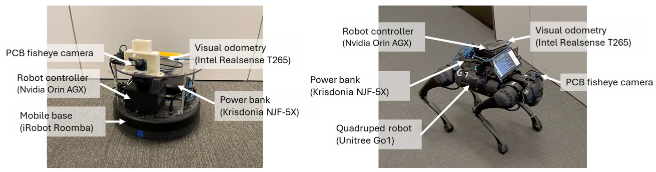


Fig. 7: **Overview of Earth Rover Zero (ERZ) and its system.** The ERZ can be controlled over a 4G internet connection for gaming and data collection and for deploying our navigation policy.



[i] Rooma-based mobile robot: VizBot [ii] Quadruped robot: Unitree Go1

Fig. 8: **Overview of robotic platforms, VizBot and Go1.** These robots mounts different camera from ERZ and can be controlled with an onboard robot controller on an Nvidia Orin AGX with ROS.

the internet, resulting in about a 0.7-second time delay between the observations and the velocity commands.

All measurements from the sensors can be accessed through the platform's API. Linear and angular velocity commands can also be sent to the robot from user teleoperation for data collection (gaming) or from our trained policies for navigation.

Other robotic platforms: We conduct additional evaluations with different robot hardware and systems to analyze the cross-embodiment performance of our policy. We show the overview of the VizBot [29] and the Unitree Go1 quadruped robot in Fig. 8. Different from the ERZ, we deploy our trained policy on an Nvidia Orin AGX mounted on the robot and evaluate navigation performance. Instead of using GPS, we mount a tracking camera on top of our robot for localization indoors. In addition, we use a different camera, a PCB-mounted fisheye camera, and use its image observations for inference.

Baseline methods: In our evaluation of long-horizon navigation, we use the following two baselines, NoMaD and IL. For IL, we evaluate various annotation methods as the ground truth action labels to compare with our MBRA relabeler.

NoMaD [40]: We deploy the original NoMaD policy [40] for exploration and generate 30 possible trajectories. Out of these options, we select the best trajectory by measuring the distance between the last predicted position and the goal pose and selecting the minimum one to control the robot.

Imitation learning [39]: We train a long-horizon navigation policy on reannotated action labels by following several baseline methods instead of using our MBRA. All learning setups except annotation are same as our method.

Raw action label: As the simplest action commands, we annotate the robot trajectory with the recorded GPS and the compass readings at 1.0 Hz. To match the image frame rate, we linearly interpolate between adjacent timesteps. For training, we sample the raw robot poses for 8 steps at 3.0 Hz and transform them into the local robot coordinate frame to be used as ground truth.

Filtered action label: We give the above mentioned Extended Kalman filter (EKF) for entire FrodoBots-2k dataset to estimate the less-noisy robot pose. Similar to the raw action label, we sample the filtered pose for 8 steps at 3.0 Hz in training and transform them into the local robot coordinate.

VPT [3]: We train the inverse dynamics models to estimate the relative pose between two consecutive observations such as $p^i_{i+1} = f_{idm}(O_i, O_{i+1})$. In training, we sample 9 image flames from the current frame to the 8-step future frame at 3.0 Hz as $\{O_i\}_{i=0...8}$ and estimate the relative poses $\{p^i_{i+1}\}_{i=0...7}$ between each frame. Then we integrate the estimated relative poses $\{p^i_{i+1}\}_{i=0...7}$ to have the trajectories in the current local coordinate and use the local trajectories as the ground truth.

Multi-step VPT [3, 39, 16]: Following [39], we train the robotic foundation model to estimate the robotic action to move between two frames, O_c and O_g. Since we want to annotate the actions for 8 steps, we select O_g as the 8 step future frame from O_c in training. The other training setups are same as the original paper [39]. In training, we sample O_c and O_g (8 step future frame for O_c) and estimate the robotic action to move toward the location of O_g and supervise its estimated action commands.

For training relabelers such as VPT and multi-step VPT, we use both the curated GNM dataset and 1 $\%$ FrodoBots-2k dataset to be accurate models. We decide the ratio of the FrodoBots-2k dataset as 1 $\%$ according to the ablation study in the appendix. By mixing small FrodoBots-2k dataset with the clean GNM dataset, our model can suppress the negative effect of the noisy FrodoBots-2k dataset and can learn the target robot characteristics. Note that we use all FrodoBots-2k dataset to train the long-horizon navigation policy.

Action and observation space: The action space of our policy and relabeler is defined as the pose—comprising the (x, y) position and yaw angle—on a 2D plane. Following the original implementation of [14], the MBRA model internally generates a sequence of linear and angular velocities, which are then integrated to produce action commands in the position space for training LogoNav.

During inference with LogoNav, we employ a PD controller to generate linear and angular velocity commands that track the generated pose commands, in accordance with the original implementations of [38, 39, 40]. Since our robotic hardware platforms including ERZ, Vizbot, and Go1 are controlled via linear and angular velocity commands (which are subsequently translated into low-level control actions such as wheel angular velocities or leg joint angles), we directly apply these commands to guide the robot toward the target goal pose. Furthermore, during inference with the MBRA model used as a goal image-conditioned navigation policy (see Sec.V-D), we directly use the linear and angular velocity commands internally generated by the model, without employing a PD controller. As visually illustrated in Figs.2 and 6, our observations consist of raw camera images. Prior to being input to the network, these images are processed using standard normalization.

References

[1] FrodoBots-2k. https://huggingface.co/datasets/frodobots/FrodoBots-2K, Accessed: 2025-01-29.

[2] Aisha Alnajdi, Atharva Suryavanshi, Mohammed S Alhajeri, Fahim Abdullah, and Panagiotis D Christofides. Machine learning-based predictive control of nonlinear time-delay systems: Closed-loop stability and input delay compensation. *Digital Chemical Engineering*, 7:100084, 2023.

[3] Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video pretraining (vpt): Learning to act by watching unlabeled online videos, 2022. URL https://arxiv.org/abs/2206.11795.

[4] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[5] Carlos Campos, Richard Elvira, Juan J. Gomez Rodriguez, Jose M. M. Montiel, and Juan D. Tardos. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, December 2021. ISSN 1941-0468. doi: 10.1109/tro.2021.3075644. URL http://dx.doi.org/10.1109/TRO.2021.3075644.

[6] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

[7] S. Y. Chen. Kalman filter for robot vision: A survey. *IEEE Transactions on Industrial Electronics*, 59(11): 4409–4420, 2012. doi: 10.1109/TIE.2011.2162714.

[8] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[9] Leilei Cui, Bo Pang, and Zhong-Ping Jiang. Learning-based adaptive optimal control of linear time-delay systems: A policy iteration approach. *IEEE Transactions on Automatic Control*, 69(1):629–636, 2023.

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[11] Noriaki Hirose and Kosuke Tahara. Depth360: Self-supervised learning for monocular depth estimation using learnable camera distortion model. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 317–324. IEEE, 2022.

[12] Noriaki Hirose, Amir Sadeghian, Marynel Vázquez, Patrick Goebel, and Silvio Savarese. Gonet: A semi-supervised deep learning approach for traversability estimation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3044–3051. IEEE, 2018.

[13] Noriaki Hirose, Fei Xia, Roberto Martín-Martín, Amir Sadeghian, and Silvio Savarese. Deep visual mpc-policy learning for navigation. *IEEE Robotics and Automation Letters*, 4(4):3184–3191, 2019.

[14] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Exaug: Robot-conditioned navigation policies via geometric experience augmentation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4077–4084. IEEE, 2023.

[15] Noriaki Hirose, Dhruv Shah, Ajay Sridhar, and Sergey Levine. Sacson: Scalable autonomous control for social navigation. *IEEE Robotics and Automation Letters*, 2023.

[16] Noriaki Hirose, Catherine Glossop, Ajay Sridhar, Oier Mees, and Sergey Levine. Lelan: Learning a language-conditioned navigation policy from in-the-wild video. In *8th Annual Conference on Robot Learning*, 2024.

[17] Noriaki Hirose, Dhruv Shah, Kyle Stachowicz, Ajay Sridhar, and Sergey Levine. Selfi: Autonomous self-improvement with rl for vision-based navigation around people. In *8th Annual Conference on Robot Learning*, 2024.

[18] Gregory Kahn, Adam Villaflor, Bosen Ding, Pieter Abbeel, and Sergey Levine. Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation. In *2018 IEEE international conference on robotics and automation (ICRA)*, pages 5129–5136. IEEE, 2018.

[19] R.E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82 (1):35–45, 1960.

[20] Haresh Karnan, Anirudh Nair, Xuesu Xiao, Garrett Warnell, Sören Pirk, Alexander Toshev, Justin Hart, Joydeep Biswas, and Peter Stone. Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation. *IEEE Robotics and Automation Letters*, 7(4):11807–11814, 2022.

[21] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024.

[22] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[23] Wook Hyun Kwon, Jin Won Kang, Young Sam Lee, and Young Soo Moon. A simple receding horizon control for state delayed systems and its stability criterion. *Journal of Process Control*, 13(6):539–551, 2003.

[24] Wook Hyun Kwon, YS Lee, and Soo Hee Han. General receding horizon control for linear time-delay systems.

Automatica, 40(9):1603–1611, 2004.

[25] Xinhao Liu, Jintong Li, Yichen Jiang, Niranjan Sujay, Zhicheng Yang, Juexiao Zhang, John Abanes, Jing Zhang, and Chen Feng. Citywalker: Learning embodied urban navigation from web-scale videos. *arXiv preprint arXiv:2411.17820*, 2024.

[26] Roberto Martin-Martin, Mihir Patel, Hamid Rezatofighi, Abhijeet Shenoi, JunYoung Gwak, Eric Frankel, Amir Sadeghian, and Silvio Savarese. Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments. *IEEE transactions on pattern analysis and machine intelligence*, 45(6):6748–6765, 2021.

[27] Xiangyun Meng, Nathan Ratliff, Yu Xiang, and Dieter Fox. Scaling local control to large-scale topological navigation, 2020. URL https://arxiv.org/abs/1909.12329.

[28] Riku Murai, Eric Dexheimer, and Andrew J. Davison. Mast3r-slam: Real-time dense slam with 3d reconstruction priors, 2024. URL https://arxiv.org/abs/2412.12392.

[29] Takahiro Niwa, Shun Taguchi, and Noriaki Hirose. Spatio-temporal graph localization networks for image-based navigation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3279–3286. IEEE, 2022.

[30] Abby O'Neill, Abdul Rehman, Abhinav Gupta, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[31] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[32] Deepak Pathak, Parsa Mahmoudieh, Guanghao Luo, Pulkit Agrawal, Dian Chen, Yide Shentu, Evan Shelhamer, Jitendra Malik, Alexei A Efros, and Trevor Darrell. Zero-shot visual imitation. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 2050–2053, 2018.

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[34] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *International Conference on Learning Representations*, 2018.

[35] Amirreza Shaban, Xiangyun Meng, JoonHo Lee, Byron Boots, and Dieter Fox. Semantic terrain classification for off-road autonomous driving. In *Conference on Robot Learning*, pages 619–629. PMLR, 2022.

[36] Dhruv Shah and Sergey Levine. Viking: Vision-based kilometer-scale navigation with geographic hints. *arXiv preprint arXiv:2202.11271*, 2022.

[37] Dhruv Shah, Benjamin Eysenbach, Nicholas Rhinehart, and Sergey Levine. Rapid exploration for open-world navigation with latent goal models. In *Conference on Robot Learning*, pages 674–684. PMLR, 2022.

[38] Dhruv Shah, Ajay Sridhar, Arjun Bhorkar, Noriaki Hirose, and Sergey Levine. Gnm: A general navigation model to drive any robot. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7226–7233. IEEE, 2023.

[39] Dhruv Shah, Ajay Sridhar, Nitish Dashora, Kyle Stachowicz, Kevin Black, Noriaki Hirose, and Sergey Levine. Vint: A foundation model for visual navigation. *arXiv preprint arXiv:2306.14846*, 2023.

[40] Ajay Sridhar, Dhruv Shah, Catherine Glossop, and Sergey Levine. Nomad: Goal masked diffusion policies for navigation and exploration. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 63–70. IEEE, 2024.

[41] Kyle Stachowicz, Dhruv Shah, Arjun Bhorkar, Ilya Kostrikov, and Sergey Levine. Fastrlap: A system for learning high-speed driving via deep rl and autonomous practicing. In *Conference on Robot Learning*, pages 3100–3111. PMLR, 2023.

[42] Kyle Stachowicz, Lydia Ignatova, and Sergey Levine. Lifelong autonomous fine-tuning for navigation foundation models. *arXiv preprint arXiv:TODO*, 2024.

[43] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras, 2022. URL https://arxiv.org/abs/2108.10869.

[44] Zachary Teed, Lahav Lipson, and Jia Deng. Deep patch visual odometry, 2023. URL https://arxiv.org/abs/2208.04726.

[45] Samuel Triest, Matthew Sivaprakasam, Sean J Wang, Wenshan Wang, Aaron M Johnson, and Sebastian Scherer. Tartandrive: A large-scale dataset for learning off-road dynamics models. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2546–2552. IEEE, 2022.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[47] Zhi Yan, Simon Schreiberhuber, Georg Halmetschlager, Tom Duckett, Markus Vincze, and Nicola Bellotto. Robot perception of static and dynamic objects with an autonomous floor scrubber. *Intelligent Service Robotics*, 13(3):403–417, 2020.