IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

Learning to Drive Anywhere with Model-Based Reannotation

Noriaki Hirose[1,2], Lydia Ignatova[1], Kyle Stachowicz[1], Catherine Glossop[1], Sergey Levine[1] and Dhruv Shah[1,3]

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 **Re**A**nnotation**, enables imitation learning from noisy, passive data, such as low-quality crowd-sourced demonstrations or even videos from the web.

***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.**

I. Introduction

Machine learning has demonstrated remarkable success across a range of tasks, including natural language processing [1], [2] and computer vision [3], [4], [5]. A key factor driving these advancements is the availability of large and diverse training datasets. In robotics, lack of data is a major bottleneck: intentional, centralized data-collection efforts are costly, requiring real-world robots and human operators, while Internet-scraped data is rarely directly applicable to the robotics domain [6], [7].

In this paper, we study the problem of developing an end-to-end robot navigation policy capable of generalizing to a wide range of outdoor and indoor environments and navigating to distant goals hundreds of meters away. Training such an end-to-end policy requires large amounts of diverse data to grant broad coverage over possible environments. Previous navigation works [8] 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 [9], limiting the breadth of generalization that can be achieved from this high-quality data alone.

Facing this data limitation, we turn our attention to making use of more abundant sources of *passive data* – data that lacks actions or only provides low-quality action labels. 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 the actions in such 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 at all.

To enable the use of these cheap, scalable data sources, we propose robust model-based learning to train a short-horizon expert *relabeling model* for generating high-quality actions connecting two nearby states. We use this short-horizon relabeling model 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 model 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 evaluations 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 six different cities across three continents.

Our primary contributions are 1) a framework to learn a

[1]UC Berkeley, [2]Toyota Motor North America, [3]Princeton University

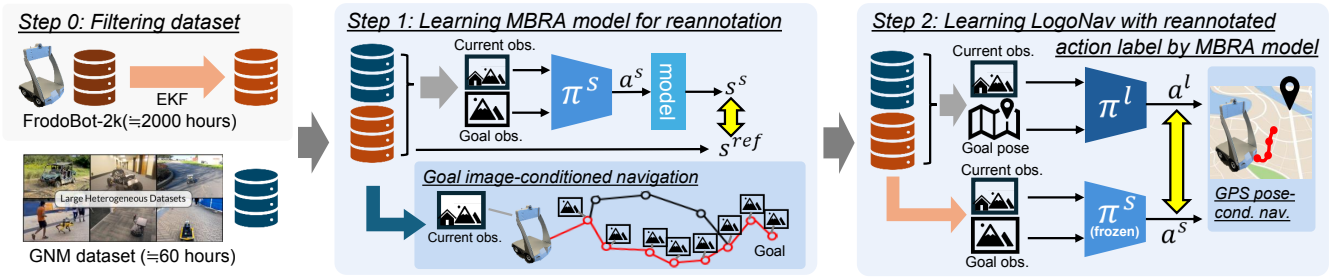


Fig. 2: Overview of MBRA. We propose a two-step process: In the first stage, we train a short-horizon reannotation policy with a robust MBL approach on the noisy dataset, which can be used for short-horizon image-conditioned navigation and which we leverage to relabel the noisy dataset with improved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

well-generalized long-horizon policy by applying a short-horizon relabeling MBRA (**M**odel-**B**ased **Re**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 that we evaluate in 6 countries, 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.

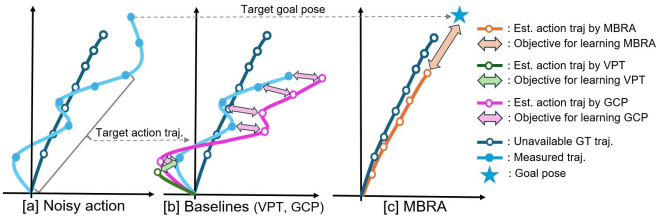


Fig. 3: The aim of MBRA is to relabel low-quality data with actions that are *better* than the actions in the dataset, in the sense that they more effectively link states over short-horizon trajectory snippets. Compared to methods that use one- or multi- step inverse models (e.g., VPT, GCP) or the original noisy actions, training on actions from MBRA leads to significantly more effective policies.

II. Related Work

Vision-based robot navigation has been widely explored to navigate toward goal positions given visual observations from a monocular camera. [10], [11], [12] 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 [13]. Some works[14] use exploration with a topological memory to seek out a distant image goal, while others[15], [16] 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) [10], [8], [9], reinforcement learning (RL) [16], [17], [18], and model-based learning (MBL) [12], [19] 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 [12], [20], GPS [14], 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 [21] and inverse dynamics models (VPT) [22] 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 [23], [24], [25], [26] offer good local trajectory estimation, its accuracy relies on having consistent, good visual features in the image view.

Prior methods have also sought to leverage suboptimal data, including data without actions, by training a separate model to infer the action given the current and next state [22]. We find that this approach performs worse than MBRA on the highly suboptimal datasets we consider, both because the action prediction must leverage other datasets that contain distributional shift. Perhaps even more importantly, MBRA generates synthetic actions that are optimized to reach future states in a trajectory while satisfying basic navigational objectives via a forward model, rather than greedily maximizing the probability of the action given a pair of adjacent states, leading to smoother and more reasonable behavior.

Related to inverse model methods such as VPT, multi-step goal-conditioned policies [9], [27] (GCPs) train a model to predict the action given the current state and a (more distant) future state. While these approaches can also take non-greedy actions, unlike MBRA, they do not optimize for actions that satisfy navigational objectives. They also still suffer from distributional shift when the action labels in the target domain are unavailable, or else must use noisy action labels if they are present, both of which degrade performance (see Fig. 3[b]). In our experiments, we find that MBRA significantly outperforms such methods when using large datasets with low-quality actions.

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

In this paper, we focus on learning a long-horizon navigation policy from a highly diverse but suboptimal dataset

$\mathcal{D}_n$. In particular, we wish to learn high-quality navigation from a crowd-sourced dataset; this requires us to train a relabeler 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, such that $|\mathcal{D}^*| \ll |\mathcal{D}_n|$.

While observations in $\mathcal{D}_n$ might represent high state coverage, the actions are low-quality: both because of inaccuracy due to state estimation errors and heterogeneity of uncurated human operators with varying skill levels. Our key insight is that a short-horizon model-based *expert* trained from these noisy datasets can be used to relabel entire trajectories. This leads to a high-quality training dataset for an end-to-end navigation policy that can imitate these clean actions (Fig. 2).

A. Learning a Short-Horizon Relabeling Model, MBRA

To train an accurate MBRA model $\{a_i^s\}_{i=0\ldots N-1} = \pi^s(O_c; O_g)$ to infer the optimal actions occurring between the current observation O_c and the goal observation O_g for re-annotation, we propose the robust MBL to leverage the entire suboptimal dataset $\mathcal{D}_n$ as well as $\mathcal{D}^*$ in training.

Overview of learning MBRA model. As illustrated in Fig. 3[b], single-step VPT and multi-step GCPs struggle to learn from noisy action labels during training. To address this, we use Model-based Learning (MBL). MBL prioritizes the final goal state, but rather than mimicking the potentially noisy actions in the dataset, it utilizes a forward model to generate synthetic actions. Crucially, these actions are optimized both to reach future states in the trajectory and to satisfy basic navigational constraints (as shown in Fig. 3[c]). This approach leads to smoother and more reasonable behavior, allowing MBL to leverage the noisy FrodoBots-2k more effectively and resulting in better labels for the entire FrodoBots-2k dataset. This allows us to improve the action label quality, while also preserving teleoperators' intent, such as avoiding collisions with pedestrians, staying on paths, etc.

Learning architecture based on MBL. We design the following model-based objective J_{mbl} for learning π^s to reach target state s^{ref} with keeping basic navigation constraints such as avoiding collision. Note that we only give the further target state s^{ref} instead of giving the individual adjacent target states such as s_i^{ref} in each i-th step not to be sensitive for low-quality states in the dataset as shown in Fig. 3[b].

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

where $\{s_i^s\}_{i=0\ldots N-1}$ are the estimated states at each step. The states $\{s_i^s\}_{i=0\ldots N-1}$ are calculated by computing roll-outs through a differentiable forward dynamic model f. The forward model considers the current observation O_c and generated actions $\{a_i^s\}_{i=0\ldots N-1}$ from the short-horizon MBRA model, π^s [28]:

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

While the states $\{s_i^s\}_{i=0\ldots N-1}$ are conditioned on actions $\{a_i^s\}_{i=0\ldots N-1}$ 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. We do not modify the forward dynamics model f during this training π_s. Detail implementation is shown in the Sec.V-E.

Note that while we are still training π^s on the suboptimal dataset in addition to $\mathcal{D}^*$, we are not directly imitating actions. By relying on the forward model f and a reasonable distant target state s^{ref}, we can mitigate the effects of both the suboptimal action labels as well as noisy tracking information (Fig. 3[c]). Therefore, the MBRA model can leverage the visually and behaviorally diverse dataset, despite the low-quality actions.

B. Learning a Long-Horizon Navigation Policy, LogoNav

To train our long-horizon navigation policy, we first re-annotate the crowd-sourced dataset $\mathcal{D}_n$ with our learned π^s. This gives us a clean set of action labels that can be distilled into an end-to-end navigation policy π^l. We want a navigation policy π^l to predict actions as $\{a_i^l\}_{i=0\ldots N-1} = \pi^l(O_c, p_g)$ 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 relabeling model, on the order of 50 meters, compared to the previous 3 meters. We train this policy using imitation learning on the re-annotated action commands $\{a_i^s\}_{i=0\ldots N-1}$ from the short-horizon relabeling model such as $\min J_{il} := \sum_{i=0}^{N-1}(a_i^s - a_i^l)^2$. By imitating the cleaned action commands linking O_c and O_g, our long-horizon policy, LogoNav , can learn navigational affordances, such as staying on paths, avoiding collisions, and not disturbing pedestrians, which is representative of the "good" navigation behavior modeled by the MBRA model. Note that we co-train on the relabeled $\mathcal{D}_n$ as well as the high-quality dataset $\mathcal{D}^*$. We freeze π^s while training π^l.

IV. Implementation

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

A. Passive Dataset

We evaluate our approach with two different passive datasets, a crowd-sourced robotic dataset, FrodoBots-2k, and an in-the-wild YouTube video dataset described in [29]. 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 [37] 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 is more than 25 times larger than other datasets and includes a diverse set of real robot trajectories teleoperated by humans.

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

Dataset	Policy	hour	Sensors
KITTI odom [30]	teleop	0.7	RGB, 3D LiDAR, GPS
NCLT [31]	teleop	34.9	RGB, 3D LiDAR, odom, GPS, IMU
GO Stanford [20], [12]	teleop	10.3	RGBs, odom
FLOBOT [32]	auto	0.46	RGBD, 3D and 2D LiDAR, odom, IMU.
RECON [14]	auto	25.0	stereo RGBD, 2D LiDAR, GPS, IMU
JRDB [33]	teleop	1.1	stereo RGBD, 3D and 2D LiDAR, IMU
SCAND [34]	teleop	8.7	RGBD, 3D LiDAR, odom
TartanDrive [35]	teleop	5.0	RGBD, GPS, IMU
HuRoN [36]	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

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 sensor measurements cannot be reliably used to estimate robot poses, policies trained on raw actions have poor performance. The main factors of 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 delay. Details of the robot system are shown in [38] and Sec. V-A.

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 [29], 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, and environment types (urban, rural, etc.).

In addition to the passive data, we use the public expert datasets RECON [14], GO Stanford [20], [12], CoryHall [39], TartanDrive [35], HuRoN [36], Seattle [40], and SCAND [34] 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) [41] 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 [42], the signal remains too noisy for direct training.

C. Training Details

We describe the training settings for both our short-horizon relabeling model, and long-horizon navigation policy.

Short-horizon relabeling model: Following [19], to encourage the MBRA model to smoothly connect between O_c and O_g without collision, we design s_i^s by three components, $[\hat{p}_i, \hat{c}_i, \Delta a_i^s]$, 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], where p_g is further goal pose.

Since we design $\{a_i^s\}_{i=0\ldots N-1}$ as the linear and angular velocities, we calculate the unicycle model to integrate the velocity commands for N = 8 steps at 3 Hz and generate the virtual robot poses $\{\hat{p}_i\}_{i=0\ldots N-1}$ in the dynamic forward model f. In addition, we estimate 3D point clouds from the current image O_c via the depth estimation model [28] and count collided 3D points at $\{\hat{p}_i\}_{i=0\ldots N-1}$ as $\{\hat{c}_i\}_{i=0\ldots N-1}$ in f. By penalizing $\hat{c}_i$ to be smaller values, the MBRA model learns to generate collision-free actions. While our objectives are not explicitly designed to learn semantic behaviors such as staying on paths or avoiding pedestrians, the generated trajectories effectively connect O_c and O_g, implicitly capturing such behaviors. Although the action labels are unusable due to heavy noise, the image sequences from teleoperation still reflect the teleoperator's semantic intent.

Furthermore, since the robot system has L steps system delay [43], [44] 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 [45], [46], 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. More details are shown in [19] and our supplemental code base.

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 J_{il} using the same parameters and settings as the short-horizon relabeling model otherwise. Since the action space for long-horizon navigation is the 2D pose, following [9], we use the integrated pose commands from the MBRA model as the supervision. In inference, we apply the PD controller to calculate the velocity commands from the generated target pose, similar to [9], [27]. The observation space of O_c and O_g is the image space for both policies.

Network design: Figure 4 shows the network architecture of both our MBRA model, π^s and LogoNav policy, π^l. For 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 $\{O_i\}_{i=-M\ldots0}$ and generate a history embedding with EfficientNet-B0. We pass in these visual fea-

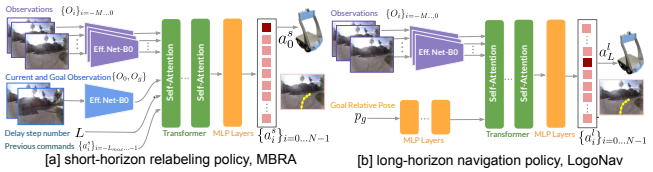


Fig. 4: Network architecture. In addition to the visual observations, We feed the delay step and the previous actions to consider the system delay in the MBL objective. For the long-horizon navigation policy, we replace the visual encoder for the current and the goal observation with the MLP layers for the goal pose.

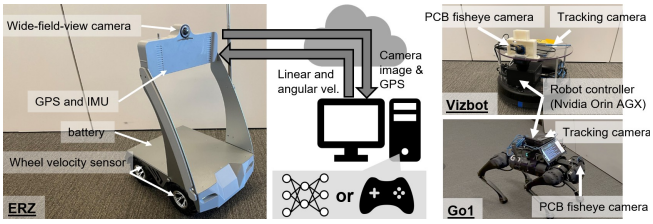


Fig. 5: Overview of the robot hardwares and systems. ERZ can be controlled over a internet connection for data collection and for deploying our navigation policy. Vizbot and Go1 with different cameras are controlled from the onboard robot controller with ROS.

tures, 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 action commands $\{a_i^s\}_{i=0\ldots N-1}$.

For π^l, we replace the visual encoder for O_c and O_g with MLP layers for the local goal pose p_g on the current robot coordinate in our implementation. In addition, we 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 [47] and [9]. More details are shown in the supplemental code base.

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 MBRA more effective at learning relabeler from low-quality datasets than IL?

A. Evaluation Setup

We describe both the short-horizon and long-horizon navigation tasks, as well as the robot hardware, on which we evaluate our method, along with their associated baselines.

Short-horizon navigation policy: Short-horizon navigation policy (MBRA model) 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 [10], [12], [9]. 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 as the current node at each time step, following [9], [27]. We 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 LogoNav policy can navigate to goals between 25-100 meters from the initial robot pose. We rely on GPS(outdoor) and tracking camera [48](inside) to get robot positions and specify goals. We evaluate longer trajectories by setting multiple subgoals at intervals of approximately 80 meters apart. At every step, we calculate the 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.

Robot hardware: The FrodoBot "Earth Rover Zero" (ERZ), shown in Fig. 5, is a low-cost RC car used both for the FrodoBots-2k dataset collection and our main evaluation. The ERZ 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. 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. The rover can turn in place and lasts for five hours on a fully charged battery.

In addition, we conduct additional evaluations with different robot hardware and systems with the wheeled mobile robot, VizBot [49] and the quadruped robot, Unitree Go1 to analyze the cross-embodiment performance of our policy.

B. Baseline Method

In our evaluation, we use the following two baselines, NoMaD and behavior clolning (BC). For BC, we evaluate various annotation methods as the ground truth action labels to compare with our MBRA model.

NoMaD [27]: We deploy the NoMaD policy [27] 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.

Behavior Cloning [9]: We train a long-horizon navigation policy on reannotated action labels by following several baseline methods instead of using our MBRA. Similar to our methods, we sample 8 steps robot trajectory at 3.0 Hz in the current robot coordinates in all baseline methods. All learning setups except annotation are same as our method.

Raw action label: As the simplest action commands, we annotate the robot trajectory by integrating the teleoperator's velocity command.

Filtered action label: We give the mentioned EKF for entire FrodoBots-2k dataset to estimate the less-noisy robot pose. We transform them into the local robot coordinate.

Visual SLAM [21]: Following [21], we estimate the global trajectories with one of the state-of-the-art viusal SLAM, DPVO [26]. To have better pose estimation, we rectify all images in FrodoBots-2k dataset for DPVO.

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

	FrodoBots-2K Data		Score	
Policy	Usage	Relabeler	GS	COV
NoMaD [27]	GNM only	-	0.333	0.471
Behavior Cloning	✓	filtered action [42]	0.286	0.429
	✓	raw action	0.286	0.567
	✓	filtered action [42]	0.286	0.624
	✓	visual SLAM [26]	0.286	0.486
	✓	VPT [22]	0.095	0.314
	✓	GCP [9]	0.619	0.757
LogoNav	✓	MBRA	**0.857**	**0.924**

VPT [22]: We train the inverse dynamics model (IDM) to estimate the relative pose between two consecutive observations such as $p^i_{i+1} = f_{idm}(O_i, O_{i+1})$ by imitating the ground-truth relative pose in the dataset. In training, we sample 9 flames from the current frame to the 8-step future frame and estimate the relative poses between each frame by IDM. Then we integrate the estimated relative poses to have the robot trajectories on the current coordinate.

GCP [9]: Following [9], we train the policy as GCP to estimate the robot trajectory to link 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 others are same as the original paper [9].

For VPT and GCP, 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 data ablation study in the evaluation section V-E. By mixing small FrodoBots-2k dataset with the clean GNM dataset, VPT and GCP can suppress the negative effect of the noisy FrodoBots-2k dataset and can learn the target robot characteristics. Besides, our MBRA model can use full FrodoBots-2k dataset in training due to the robust learning architecture of the model-based learning.

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

To answer **Q1**, we evaluate the long-horizon navigation policies trained with MBRA and several baselines. 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 pose before it fails. Our policy with MBRA shows stronger performance than all baselines for both goal success rate and coverage rate. In Sec. V-E, we conduct an investigation to analyze the advantageous gap of MBRA to answer **Q3**.

Figure 6 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 GCP could not execute this action, failing by colliding with bushes or requiring interventions to avoid falling down stairs. To show the capability of MBRA, 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 scenes. As shown in Fig. 7, 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 deployed LogoNav on two more robotic embodiments, including VizBot [49] in an indoor setting, and the Unitree Go1 quadruped robot in an outdoor setting. We conducted 10 trials from up to 100 meters away in different challenging environments with some obstacles for each embodiment and method. We show the quantitative results in Table III and show the robotic behaviors in Fig. 8 and the supplemental videos. We achieve strong goal-reaching behavior with collision avoidance compared to the strongest baseline in Table II, highlighting the policy's generalization ability. Note that we apply the same policy in Table II and feed the generated actions without any adaptation. Action conversion is internally applied in each robot setup.

D. Training policies on in-the-wild video with MBRA

For **Q2**, we evaluate the capability of MBRA on different passive data sources, action-free in-the-wild video. We use the MBRA model to generate the action labels for the in-the-wild videos and train the short-horizon visual navigation policy conditioned on goal images, $\{a^v_i\}_{i=0\ldots N-1} = \pi^v(O_c, O_g)$. During training, we use the same objective J_{il} to imitate the action labels generated by MBRA. We train three goal image-conditioned policies, one with the GNM dataset alone and another two with GNM + in-the-wild videos with different annotations, visual SLAM [26] and our MBRA model to evaluate how well 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 IV, 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, we achieve a high success rate by training with diverse video data. Besides, visual SLAM often fails on videos with fisheye lenses, dense crowds, or few features, leading to worse performance than the others.

E. Evaluating MBRA for effective crowd-sourced data use

To answer **Q3**, we compare MBRA and GCP that demonstrated the strongest performance in Table II. We train several relabelers with different data setups for each method and deploy them 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 relabelers.

Table VI shows the goal success rate and the subgoal coverage rate for each policy. We find that GCP completely deteriorates the performance by imitating the noisy raw

TABLE III: Quantitative analysis with quadruped robot, Go1 and wheeled robot, VizBot for cross-embodiment analysis.

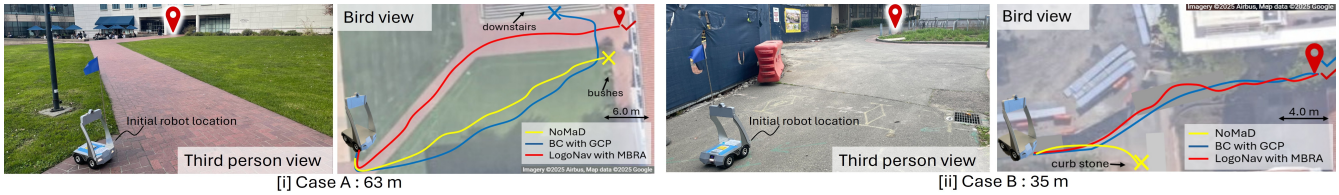
Method		Go1 (outside)		VizBot (inside)	
Policy	Relabeler	GS	COV	GS	COV
Behavior Cloning	GCP	0.300	0.680	0.200	0.630
LogoNav	MBRA	**0.800**	**0.850**	**0.600**	**0.820**

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

GNM	YouTube video (LeLaN)	GS	SC
✓	✗	0.500	0.680
✓	✓(Visual SLAM [26])	0.125	0.313
✓	✓(MBRA)	0.875	0.909

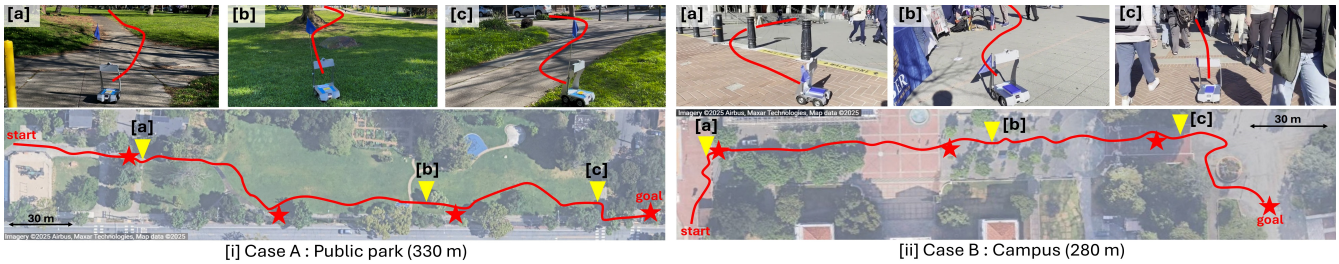
TABLE V: Evaluation of the goal image-conditioned navigation at six countries.

Policy	Dataset	GS	SC
GCP [9]	GNM	0.500	0.736
GCP [9]	GNM + FrodoBots-2k (1%)	0.792	0.906
MBRA	GNM	0.833	0.899
MBRA	GNM + FrodoBots-2k (full)	**0.958**	**0.983**



[i] Case A : 63 m [ii] Case B : 35 m

Fig. 6: Policy rollouts for goal pose-conditioned navigation with long-horizon policies. Our policy, LogoNav trained with MBRA can keep traveling on the road and arrive at the goal pose.



[i] Case A : Public park (330 m) [ii] Case B : Campus (280 m)

Fig. 7: 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.

Fig. 8: Visualization of cross-embodiment analysis.

TABLE VI: Comparison of MBRA and GCP on short-horizon navigation.

Dataset		GCP		MBRA	
GNM	FrodoBots-2k	GS	SC	GS	SC
✓	✗	0.500	0.680	0.875	0.960
✓	raw action	0.000	0.308	0.500	0.777
✗	filtered action	0.125	0.377	0.875	0.940
✓	filtered action(1%)	0.750	0.887	0.875	0.889
✓	filtered action(10%)	0.375	0.638	0.875	0.970
✓	filtered action(40%)	0.500	0.641	**1.000**	**1.000**
✓	filtered action(70%)	0.375	0.748	**1.000**	**1.000**
✓	filtered action	0.375	0.576	**1.000**	**1.000**

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, GCP cannot effectively leverage the entire FrodoBots-2k dataset. Besides, MBL enables us to scalably learn our MBRA model from the noisy data. MBRA model 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 model. 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 trajectory. To the best of our knowledge, we are the first to conduct a global evaluation for visual navigation. We evaluate GCP and MBRA model trained with and without the FrodoBots-2k dataset. Findings are summarized in Table V. MBRA model as short-horizon goal image-conditioned navigation policy had better performance for both goal reaching and subgoal coverage than GCP.

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 MBRA approach to reannotating noisy crowd-sourced data and action-free in-the-wild videos 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 (GCP), 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 [36]. 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 of human operators in the data, such preferences are not explicitly enforced.

ACKNOWLEDGMENTS

This work 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 robot hardware and computational resources for our evaluations.

REFERENCES

[1] A. Vaswani *et al.*, "Attention is all you need," *NeurIPS*, vol. 30, 2017.
[2] T. Brown *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.
[3] A. Radford *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*. PMLR, 2021, pp. 8748–8763.
[4] A. Kirillov *et al.*, "Segment anything," in *Proceedings of ICCV*, 2023, pp. 4015–4026.
[5] M. Oquab *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023.
[6] A. O'Neill *et al.*, "Open x-embodiment: Robotic learning datasets and rt-x models," *arXiv preprint arXiv:2310.08864*, 2023.
[7] A. Khazatsky *et al.*, "Droid: A large-scale in-the-wild robot manipulation dataset," *arXiv preprint arXiv:2403.12945*, 2024.
[8] D. Shah *et al.*, "Gnm: A general navigation model to drive any robot," in *ICRA*. IEEE, 2023, pp. 7226–7233.
[9] ——, "Vint: A foundation model for visual navigation," *arXiv preprint arXiv:2306.14846*, 2023.
[10] N. Savinov *et al.*, "Semi-parametric topological memory for navigation," in *International Conference on Learning Representations*, 2018.
[11] D. Pathak *et al.*, "Zero-shot visual imitation," in *Proceedings of CVPR workshops*, 2018, pp. 2050–2053.
[12] N. Hirose *et al.*, "Deep visual mpc-policy learning for navigation," *IEEE RA-Letters*, vol. 4, no. 4, pp. 3184–3191, 2019.
[13] X. Meng *et al.*, "Scaling local control to large-scale topological navigation," in *ICRA*. IEEE, 2020, pp. 672–678.
[14] D. Shah *et al.*, "Rapid exploration for open-world navigation with latent goal models," in *CoRL*. PMLR, 2022, pp. 674–684.
[15] D. Shah and S. Levine, "Viking: Vision-based kilometer-scale navigation with geographic hints," *arXiv preprint arXiv:2202.11271*, 2022.
[16] K. Stachowicz *et al.*, "Fastrlap: A system for learning high-speed driving via deep rl and autonomous practicing," in *Conference on Robot Learning*. PMLR, 2023, pp. 3100–3111.
[17] N. Hirose *et al.*, "Selfi: Autonomous self-improvement with rl for vision-based navigation around people," in *CoRL*, 2024.
[18] K. Stachowicz *et al.*, "Lifelong autonomous fine-tuning for navigation foundation models," in *CoRL*, 2024.
[19] N. Hirose *et al.*, "Exaug: Robot-conditioned navigation policies via geometric experience augmentation," in *ICRA*, 2023, pp. 4077–4084.
[20] ——, "Gonet: A semi-supervised deep learning approach for traversability estimation," in *IROS*, 2018, pp. 3044–3051.
[21] X. Liu *et al.*, "Citywalker: Learning embodied urban navigation from web-scale videos," *arXiv preprint arXiv:2411.17820*, 2024.
[22] B. Baker *et al.*, "Video pretraining (vpt): Learning to act by watching unlabeled online videos," *NeurIPS*, vol. 35, pp. 24 639–24 654, 2022.
[23] C. Campos *et al.*, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE T-RO*, vol. 37, no. 6, p. 1874–1890, Dec. 2021.
[24] R. Murai *et al.*, "Mast3r-slam: Real-time dense slam with 3d reconstruction priors," *arXiv preprint arXiv:2412.12392*, 2024.
[25] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *arXiv preprint arXiv:2108.10869*, 2022.
[26] Z. Teed *et al.*, "Deep patch visual odometry," *arXiv preprint arXiv:2208.04726*, 2023.
[27] A. Sridhar *et al.*, "Nomad: Goal masked diffusion policies for navigation and exploration," in *ICRA*. IEEE, 2024, pp. 63–70.
[28] N. Hirose and K. Tahara, "Depth360: Self-supervised learning for monocular depth estimation using learnable camera distortion model," in *IROS*. IEEE, 2022, pp. 317–324.
[29] N. Hirose *et al.*, "Lelan: Learning a language-conditioned navigation policy from in-the-wild video," in *CoRL*, 2024.
[30] A. Geiger *et al.*, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*. IEEE, 2012, pp. 3354–3361.
[31] N. Carlevaris-Bianco *et al.*, "University of michigan north campus long-term vision and lidar dataset," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1023–1035, 2016.
[32] Z. Yan *et al.*, "Robot perception of static and dynamic objects with an autonomous floor scrubber," *Intelligent Service Robotics*, vol. 13, no. 3, pp. 403–417, 2020.
[33] R. Martin-Martin *et al.*, "Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *TPAMI*, vol. 45, no. 6, pp. 6748–6765, 2021.
[34] H. Karnan *et al.*, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE RA-Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022.
[35] S. Triest *et al.*, "Tartandrive: A large-scale dataset for learning off-road dynamics models," in *ICRA*. IEEE, 2022, pp. 2546–2552.
[36] N. Hirose *et al.*, "Sacson: Scalable autonomous control for social navigation," *IEEE Robotics and Automation Letters*, 2023.
[37] "FrodoBots-2k," https://huggingface.co/datasets/frodobots/FrodoBots-2K, Accessed: 2025-08-04.
[38] "Earth Rovers SDK," https://github.com/frodobots-org/earth-rovers-sdk, Accessed: 2025-08-04.
[39] G. Kahn *et al.*, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *ICRA*. IEEE, 2018, pp. 5129–5136.
[40] A. Shaban *et al.*, "Semantic terrain classification for off-road autonomous driving," in *Conference on Robot Learning*. PMLR, 2022, pp. 619–629.
[41] R. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.
[42] S. Y. Chen, "Kalman filter for robot vision: A survey," *IEEE Transactions on Industrial Electronics*, vol. 59, no. 11, pp. 4409–4420, 2012.
[43] L. Cui *et al.*, "Learning-based adaptive optimal control of linear time-delay systems: A policy iteration approach," *IEEE Transactions on Automatic Control*, vol. 69, no. 1, pp. 629–636, 2023.
[44] A. Alnajdi *et al.*, "Machine learning-based predictive control of nonlinear time-delay systems: Closed-loop stability and input delay compensation," *Digital Chemical Engineering*, vol. 7, p. 100084, 2023.
[45] W. H. Kwon *et al.*, "General receding horizon control for linear time-delay systems," *Automatica*, vol. 40, no. 9, pp. 1603–1611, 2004.
[46] ——, "A simple receding horizon control for state delayed systems and its stability criterion," *Journal of Process Control*, vol. 13, no. 6, pp. 539–551, 2003.
[47] C. Chi *et al.*, "Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots," *arXiv preprint arXiv:2402.10329*, 2024.
[48] "Intel RealSense T265," https://www.intel.com/content/dam/support/us/en/documents/emerging-technologies/intel-realsense-technology/IntelRealSenseTrackingT265Datasheet.pdf, Accessed: 2025-08-04.
[49] T. Niwa *et al.*, "Spatio-temporal graph localization networks for image-based navigation," in *IROS*. IEEE, 2022, pp. 3279–3286.