Title

SACSoN: Scalable Autonomous Control for Social Navigation

Journal

IEEE Robotics and Automation Letters, 9(1)

ISSN

2377-3766

Authors

Hirose, Noriaki

Shah, Dhruv

Sridhar, Ajay

et al.

Publication Date

2024

DOI

10.1109/lra.2023.3329626

SACSoN: Scalable Autonomous Control for Social Navigation

Noriaki Hirose[1,2], Dhruv Shah[1], Ajay Sridhar[1] and Sergey Levine[1]

***Abstract*— Machine learning provides a powerful tool for building socially compliant robotic systems that go beyond simple predictive models of human behavior. By observing and understanding human interactions from past experiences, learning can enable effective social navigation behaviors directly from data. In this paper, our goal is to develop methods for training policies for socially unobtrusive behavior, such that robots can navigate among humans in ways that don't disturb human behavior in visual navigation using only onboard RGB observations. We introduce a definition for such behavior based on the *counterfactual* perturbation of the human: if the robot had not intruded into the space, would the human have acted in the same way? By minimizing this counterfactual perturbation, we can induce robots to behave in ways that do not alter the natural behavior of humans in the shared space. Instantiating this principle requires training policies to minimize their effect on human behavior, and this in turn requires data that allows us to model the behavior of humans in the presence of robots. Therefore, our approach is based on two key contributions. First, we collect a large dataset where an indoor mobile robot interacts with human bystanders. Second, we utilize this dataset to train policies that minimize counterfactual perturbation. We provide supplementary videos and make publicly available the visual navigation dataset on our project page[1].**

I. Introduction

Even the simplest forms of interaction between humans, such as how to pass someone in a hallway, are governed by complex non-verbal cues, and may be challenging to script. In order for robots to inhabit the same environments as people, they must also be cognizant of basic social cues and etiquette, even for seemingly simple navigational tasks. While a range of prior works have proposed approaches for modeling human behavior [1, 2], the complexity of such interactions often defies analytic modeling techniques.

We approach this challenge from a data-driven perspective: acquiring policies for navigation around humans by leveraging data of human-robot interactions to *learn* how to navigate in socially unobtrusive ways. We propose a definition for such behavior, which is based on the *counterfactual* perturbation of humans. Specifically, we consider whether humans would have acted in the same way if the robot had not intruded into their space. By minimizing this counterfactual perturbation, we can guide robots to behave in a manner that does not alter the natural behavior of humans in the shared space. To instantiate this principle, we train the SACSoN (**S**calable **A**utonomous **C**ontrol for **So**cial **N**avigation) policy to minimize the impact on human behavior for vision-based navigation using a single camera. This requires us to both formalize the notion of counterfactual perturbation into an objective, and to collect a dataset that has the kinds of human-robot interactions that can allow our model to learn to predict human behavior in the presence of robots. Thus, our work focuses on two complementary technical components: the design of a policy learning method that can utilize predictive models of humans for unobtrusive navigation, and the collection of a large dataset of human-robot interactions to train these predictive models.

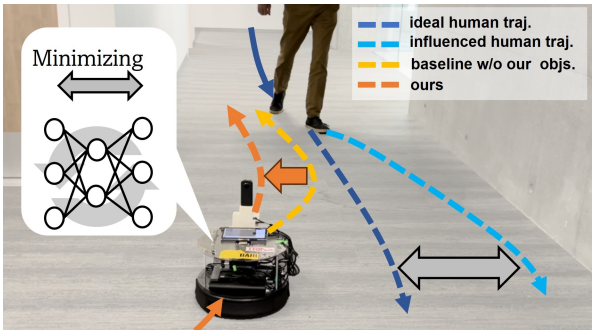


Fig. 1: **SACSoN is a socially unobtrusive vision-based navigation policy in the human-occupied spaces.** We penalize counterfactual perturbations (gray) from the intended human trajectory (navy) and generate the compliant commands (orange).

To collect such a dataset, we propose a data collection system, which we call HuRoN (**Hu**man-**Ro**bot interaction data collection for vision-based **N**avigation) system. In contrast to previous social navigation datasets that involve expensive manual tele-operation [3, 4], or simple scripted policies that fail to capture data diversity [5]. Instead, we devise an intelligent system that can *autonomously* collect rich interaction data with little-to-no human intervention, and can improve its data collection policy over time as the ever-growing dataset is reused to further train our policy.

We deploy our data collection system to collect the HuRoN dataset, which comprises over 75 hours of robotic navigation in 5 different office environments populated by people. To the best of our knowledge, this represents the largest such dataset of an autonomous mobile robot interacting with humans, with over 4000 individual human-robot interactions. In the process of collecting the HuRoN dataset, our robot traveled for a combined total of about 58.7 km over four months. Since our dataset includes time sequences of camera images, 2D LiDAR, and wheel odometry, our dataset can be useful for visual SLAM tasks including visual odometry estimation and depth estimation.

Our work makes the following contributions: (i) a model-based method for learning a socially compliant SACSoN policy for visual navigation around humans, (ii) an autonomous data collection system, HuRoN, that encourages rich in-

[1]UC Berkeley, [2]Toyota Motor North America

[1] `sites.google.com/view/SACSoN-review`

TABLE I: **Survey of public datasets for learning vision-based navigation policies in real-world environments around humans.** HuRoN is the largest available visual navigation dataset of an autonomous policy interacting with humans in real-world environments.

Dataset	Human	Policy	Duration [hour]	Distance [km]	Sensors
KITTI odometry [6]	✗	teleop	0.7	22.2	stereo RGB, 3D LiDAR, GPS
NCLT [7]	✗	teleop	34.9	147.4	RGB, 3D LiDAR, odom, GPS, IMU
GO Stanford [8]	✗	teleop	10.3	16.7	spherical RGB, odom
FLOBOT [9]	✗	autonomous	0.46	0.2	RGBD, stereo RGB, 3D and 2D LiDAR, odom, IMU.
RECON [10]	✗	autonomous	25.0	81.0	stereo fisheye RGBD, thermal, 2D LiDAR, GPS, IMU
JRDB [3]	✓	teleop	1.1	2.3	stereo RGBD, 3D and 2D LiDAR, IMU
SCAND [4]	✓	teleop	8.7	40.0	RGBD, 3D LiDAR, odom
THÖR [5]	✓	scripted	1.0	1.0	3D LiDAR, motion capture, eye-tracking glasses
MuSoHu [11]	✓	no robots	20.0	100.0	spherical RGB, stereo RGBD, 3D LiDAR, IMU
HuRoN (Ours)	✓	autonomous	75.0	58.7	spherical RGBD, fisheye RGB, 2D LiDAR, odom, bumper

teractions with human pedestrians using a novel training objective, and (iii) the HuRoN dataset, a large and diverse dataset comprising over 4000 human-robot interactions of an autonomous robot operating in a densely populated office-space environment. Please see the project page for the dataset and videos.

II. Related Work

Social navigation has been widely studied in the literature [12–14]. Model-based approaches based on the dynamic pedestrian model have clasically been applied for behavior modeling [1, 2, 15]. These methods determine the robot's actions in a virtual space with the predicted pedestrians' behavior [16–27], considering social momentum [20], a maximum entropy model [19], a model predictive controller [21], or a classical planner [22, 23]. Social navigation has also been viewed through the lens of model-free data-driven learning such as reinforcement learning [28–32].

Our method using the pedestrians' predictive model belongs to the former. However, different from prior works, including model-based reinforcement learning, we apply the predictive model to estimate the counterfactual perturbation from the pedestrians' intended trajectory and train the control policy offline by penalizing the perturbations. Hence our control policy enables the robot to navigate to the target position while allowing the pedestrian to walk as intended. Moreover, since our approach is end-to-end learning, the robot actions can be derived from raw images without detecting and predicting pedestrians in inference.

Similar to the data-driven approaches, our training method needs a large dataset. For vision-based navigation, prior works in collecting real-world data tend to use manual teleoperation, which is expensive and scales poorly [3, 4, 6–8]. Instead, in addition to the training method, our work focuses on autonomous data collection of rich human-robot interactions, aiming to train our control policy.

While there has been prior work on autonomously collecting robot navigation data [5, 9, 10, 33], our task is particularly challenging due to the dynamic agents (i.e., humans) present in the environment. To autonomously learn an accurate predictive model and socially-compliant behavior around humans, the training data must contain rich human-robot interactions, with humans walking close to the robot, and it must include a wide perceptual and behavioral diversity. The closest prior works are SCAND [4], which is teleoperated in indoor and outdoor environments, and CoBoT, THÖR [5, 33], which are autonomous but contain no visual observations; therefore, they have limited utility for learning visual navigation policies. MuSoHu [11] collects a dataset without using real robots. Instead, they have human participants walk in human-occupied spaces. Hence, they do not include any interactions between real robots and humans. Table I summarizes the existing robot navigation datasets, highlighting scene, method, size, and contained sensor signals. In addition to the training method, we propose the HuRoN system that can autonomously collect a large and diverse dataset of rich human interactions, and can be scaled with minimal human effort to multiple environments.

III. Preliminaries

We propose a method and dataset for social compliant robotic navigation with a learning-based approach. The design of our method extends ExAug [34], a control policy for vision-based navigation that optimizes a goal-directed cost function (but does not by itself consider interaction with humans). This system can navigate to user-specified goal images using a combination of a topological graph and a learned low-level control policy, and its design is related to a number of recent works on vision-based navigation with learned policies and topological maps [8, 34–38]. We build our data collection system, HuRoN, on top of the same visual navigation system.

The control policy in ExAug predicts control velocities $\{v_i, \omega_i\}_{i=1\dots N_s} = \pi^c_\phi(I_t, I_g)$ from the current image I_t and subgoal image I_g, and commands the linear velocity v_1 and the angular velocity ω_1 to the robot to reach the position of I_g, similar to receding horizon control. Here, N_s is the control horizon and t is the current step number. We commonly show the learnable parameters (e.g., ϕ) as a subscript on the model function (e.g., π^c_ϕ). The control policy is paired with a topological memory that contains images as nodes and temporal distance between them as the edges. The ExAug control policy π^c_ϕ is trained to minimize the objective

$$J_{\text{nav}}(\phi) := J_{\text{pose}}(\phi) + w_c J_{\text{col}}(\phi) + w_r J_{\text{reg}}(\phi), \qquad (1)$$

where J_{pose} corresponds to the prediction error in the relative pose estimates, J_{col} penalizes collisions, and J_{reg} is a regular-

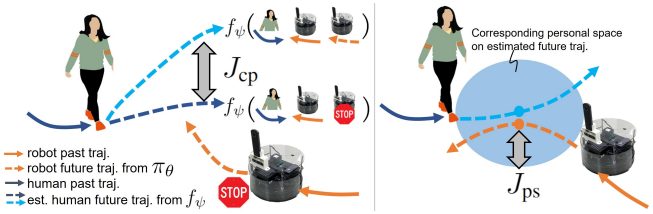


Fig. 2: **Our proposed objectives J_{cp} and J_{ps} for training SACSoN policy.** J_{cp} penalizes the counterfactual perturbation from the estimated intented pedestrian's trajectory (left). J_{ps} penalizes the personal space violation in the future space (right).

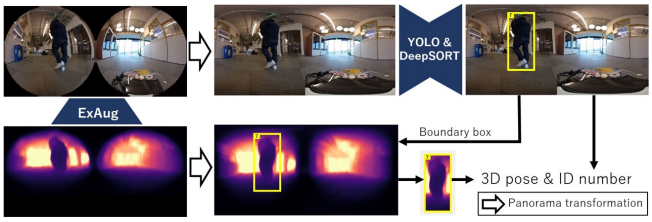


Fig. 3: **Pedestrian detection and tracking.** We use a combination of YOLO and DeepSORT to detect and track pedestrians from visual observations, and estimate their relative position using the scaled depth estimates from ExAug's perception module.

ization term for predicted velocities. ExAug uses a geometric and kinematics model to estimate the relevant states of the robot in a virtual space and calculate these objectives, akin to the model predictive control. These objectives enable us to train the policy by minimizing the differentiable cost J_{nav} without imitating the ground truth values. Please refer to the original paper for implementation details of this system [34].

Overview: Section IV introduces our method to train the SACSoN policy, which aims to enable robotic navigation among humans with minimal disruption. In addition to J_{nav}, we introduce two new objectives using the counterfactual human trajectories. We pre-train the predictive model of the pedestrians' future trajectory to estimate the counterfactual human trajectories in training. Section V describes the HuRoN system for autonomously collecting a dataset with human-robot interactions that allows us to effectively train the SACSoN policy. Our data collection system includes two key components. First, we use a policy that is similar to SACSoN, but optimized to *encourage* rather than *avoid* interactions with humans, so as to gather the maximum number of human-robot interactions. Second, the HuRoN system is designed for scalable, autonomous data collection, and includes a number of design choices to enable autonomy and continual improvement that we detail in Section V.

IV. Learning a Socially Compliant Policy

We posit that a possible way to achieve "social compliance" is for robots to avoid disrupting the *intended behavior* of pedestrians, i.e., allow humans to carry on with their activities without disruption. In our proposed method, we penalize the counterfactual perturbation of the intended trajectories of the pedestrians. We define the intended trajectory of a pedestrian as the predicted trajectory of the pedestrian from our predictive model conditioned on the robot being stationary and non-intrusive. Our method aims to control the robot so that the humans in the environment do not act differently than they would have if the robot had been stationary. This principle could be further generalized to minimize the difference to other counterfactual situations, such as ones where the robot is absent all together, but we focus on the stationary robot counterfactual as a simple instantiation of the principle. For safety, the complete design of our full objective function also includes a term to penalize the predicted distance between the human and the robot, to encourage the robot to maintain clearance, as well as the standard navigation terms described in the preceding section. Thus, we add two terms to J_{nav}, forming our full objective:

$$\min_\theta J(\theta) := J_{\text{nav}}(\theta) + w_{cp} J_{\text{cp}}(\theta) + w_{ps} J_{\text{ps}}(\theta), \quad (2)$$

where J_{cp} is an objective to suppress the counterfactual perturbation (Fig. 2 left) and J_{ps} is an objective to penalize the penetration of the personal space of the pedestrians (Fig. 2 right), where w_{cp} and w_{ps} are weights for each objective. Here, our control policy π_θ predicts velocity commands $\{v_i, \omega_i\}$ from $I_{t:t-N_p}$ and I_g, defined as follows:

$$\{v_i, \omega_i\}_{i=1\ldots N_s} = \pi_\theta(I_{t:t-N_p}, I_g) \quad (3)$$

Concatenating the past image frames gives the robot additional context that can be useful to avoid obstacles, detect pedestrians in the environment, and reduce partial observability [39].

J_{cp}: To train the policies without distracting pedestrians, we design J_{cp} using counterfactual pedestrians trajectories,

$$J_{\text{cp}}(\theta) = \frac{1}{N_s} \sum_{i=1}^{N_s} (\hat{h}^{gw}_{t+i} - \hat{h}_{t+i})^2, \quad (4)$$

where $\hat{h}^{gw}_{t+i}$ is the estimated pedestrian's 2D trajectory conditioned on the robot virtually stopping at the current position to give way and $\hat{h}_{t+i}$ is the estimated pedestrian's 2D trajectory conditioned on the robot future action. By minimizing J_{cp} with the other objectives to train our control policy, the pedestrian can walk a path similar to what they would have taken when the robot stopped and gave way, while allowing the robot itself to move toward the goal position. Here, we estimate $\hat{h}_{t+i}$ as

$$\hat{h}_{t+1:t+\beta} = f_\psi(h_{t-\alpha:t}, r_{t-\alpha:t}, r_{t+1:t+\beta}) \quad (5)$$

where f_ψ is a trained predictive model of a pedestrian's future trajectory, conditioned on their past trajectory $h_{t-\alpha:t}$, as well as the robot's past trajectory $r_{t-\alpha:t}$ and future trajectory $r_{t+1:t+\beta}$. All trajectories in Eqn. 5 are on the current robot coordinate. The values for $r_{t-\alpha:t}$ are obtained from past wheel odometry, and $r_{t+1:t+\beta}$ is derived by integrating the velocity commands $\{v_i, \omega_i\}_{i=1\ldots N_s}$ from our control policies. To obtain $h_{t-\alpha:t}$, we use YOLO [40, 41] and DeepSORT [42] to detect and track pedestrians in the images (processed into a panorama) from the recorded observations of the robot [43], and project these detections in 3D using

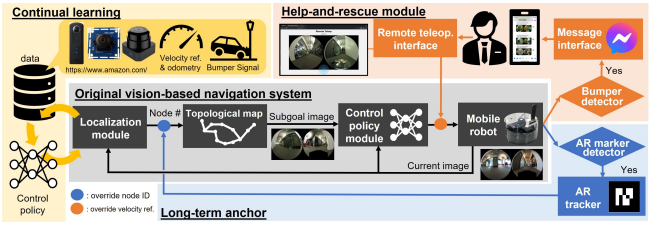


Fig. 4: **HuRoN System overview.** We design our autonomous data collection platform around a vision-based navigation system (gray) that uses a topological graph and a learned control policy. Our proposed system has three key components: a help-and-rescue module for collision recovery (orange), long-term anchors for localization (blue), and continual learning (yellow).

the depth and scale estimates [44] obtained from the ExAug perception module, as shown in Fig. 3.

For the other counterfactual trajectory, we input a zero vector instead of $r_{t+1:t+\beta}$ to estimate $\hat{h}^{gw}_{t+1:t+\beta}$ as $f_\psi(h_{t-\alpha:t}, r_{t-\alpha:t}, \mathbf{0})$. Giving the zeros vector as the robot future trajectory corresponds to stopping at the current pose. Note that we only consider scenes involving a single pedestrian for simplicity; for scenes with multiple pedestrians, we consider the nearest non-stationary pedestrians for training, since they are most likely to interact with the robot. To obtain an accurate predictive model f_ψ, we collect an interaction-enriched dataset using the HuRoN system (Section V), and train f_ψ before training π_θ.

J_{ps}: We design J_{ps} to encourage the robot to avoid the personal space of the pedestrians.

$$J_{\text{ps}}(\theta) = \min_i\{|\mathcal{R}_h + \mathcal{R}_r - c(d_i)|\}, \tag{6}$$

where $\mathcal{R}_h$ is the personal space, $\mathcal{R}_r$ is the robot radius, d_i is the distance on 2D plane between the future pedestrians' position $\hat{h}_{t+i}$ and the future robot position r_{t+i}, and c is the function to limit d_i between 0 and $\mathcal{R}_h + \mathcal{R}_r$ to penalize the robot trajectories only penetrating the personal space. J_{ps} may be alternatively defined as the mean of the set $\{|\mathcal{R}_h + \mathcal{R}_r - c(d_i)|\}$, but empirically, we found the min formulation of Eqn. 6 to better capture the desired behavior.

V. Autonomous Data Collection System

For our counterfactual objective to effectively supervise the robot's policy, we rely on the predictive model f_ψ to make accurate predictions about hypothetical human-robot interactions. This requires training f_ψ on a diverse dataset that contains many interactions between pedestrians and our robot. Therefore, the second major contribution of our work is an autonomous data collection system that can collect such a dataset. During collection, we wish to *maximize* interactions between the robot and pedestrians, while also maintaining autonomy, to collect high-quality data.

A. System design

We design a scalable data collection system with the data collection policy π_ρ that is largely autonomous and can operate in large, indoor environments without any high-fidelity indoor positioning system. Our proposed system (see Fig. 4) builds on top of the existing ExAug navigation system using three key components: (a) help-and-rescue module for collision recovery, (b) long-term anchors for coarse localization in the environment, and (c) continual learning for improving performance over the course of deployment.

Encouraging interactions: In contrast to the SACSoN policy, the data collection policy π_ρ is trained to collect a dataset with enriched human-robot interactions. We introduce an additional objective J_{int} to encourage the robot to approach pedestrians while collecting data towards the desired goal.

$$\min_\rho J(\rho) := J_{\text{nav}}(\rho) + w_i J_{\text{int}}(\rho), \tag{7}$$

where w_i is a scaling factor. Here, we employs the same network structure π_ρ as π_θ of Eqn. 3 for the data collection control policy. J_{int} is designed to minimize the distance between human and robot trajectories as $J_{\text{int}}(\rho) = \min_i\{|r_{t+i} - h_{t+i}|\}$ where $\mathcal{R} = \{r_{t+1}, r_{t+2}, \dots r_{t+N_s}\}$ and $\mathcal{H} = \{h_{t+1}, h_{t+2}, \dots h_{t+N_s}\}$ are the robot and human trajectories estimated by same approach in Section IV. Similar to J_{ps}, we only penalize the smallest $|r_i - h_i|$ by giving min formulation to better capture the desired interaction behavior.

Help-and-rescue module: To make the data collection process as seamless and autonomous as possible, we designed a pipeline for autonomous recovery from collisions and remote help in case of irrecoverable collisions for the challenging obstacles. We built a messaging and remote teleoperation interface, where the robot sends a signal to a remote operator when in need of remote teleoperation.

Long-term anchors: To overcome the limitation of localization in the repetitive environments, we place AR tags [45] throughout the environment at approximately 10 meters apart. Since these tags are located at fixed anchor locations, we can use their coarse positions to find the corresponding nodes of our topological graph. Please see the supplemental material on our project page for more information about Help-and-rescue module and Long-term anchors.

Continual learning: As the data collection system is deployed, it may encounter novel challenges—such as varying environmental lighting throughout the day, new obstacles in the environment etc.—and it must adapt the learned data collection behavior to these changes. To achieve this, our system adopts a continual learning approach, where training data at the end of each day of deployment is used to *fine-tune* the data collection policy to incorporate new experience. We also use data collected across multiple days, and times of day, to augment the fine-tuning data by chaining diverse trajectory segments between the same subgoals [46]. This allows us to effectively incorporate experience over multiple days of deployment, while also improving robustness to variations across different days and times of day.

B. Data collection

We use the above HuRoN system with the data collection policy π_ρ to autonomously collect over 75 hours of robot navigation data in 5 diverse human-occupied environments,

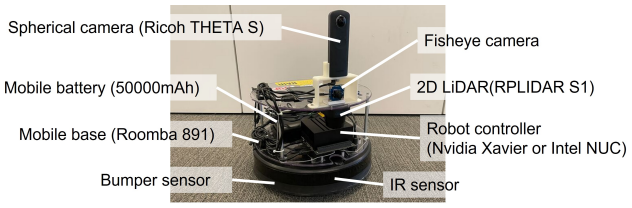


Fig. 5: **Data collection platform.** We collect spherical and fisheye RGB images, 2D LiDAR, global odometry (using long-term visual anchors), and bumper signals using our robotic system.

capturing over 4000 rich interactions with humans. We describes the robotic system used for data collection environment setup, as well as key characteristics of the dataset.

Robot and Environment Setup: Figure 5 shows an overview of our platform, built on top of an iRobot Roomba base [44, 47]. The robot is equipped with two visual sensors (a spherical camera, and a 170° wide-angle RGB camera), and a 2D LiDAR. We use two identical data collection robots with identical sensors, equipped with different onboard computers: an NVIDIA Jetson Xavier AGX, and an Intel i5 NUC, with all computation run onboard without a dedicated GPU. Our system commands angular and linear velocity commands to the base, and has access to the bumper collision sensor for triggering our help-and-rescue module.

During deployment, we instrument the environment with N_{AR} AR tags to coarsely define the robot's route for data collection (approximately 10 m apart), and collect an example trajectory by teleoperation. This example trajectory is subsampled at a fixed frame rate of 0.5 fps to generate a topological graph of the environment. Additionally, we associate each AR tag with neighboring image nodes by collecting their ID and relative pose estimates in the robot's local frame. Starting with a *base* control policy that does not encourage human interactions, HuRoN system autonomously collects data that is used to train a new control policy that can interact with humans (Section V-A) and can improve with increasing environmental experience (Section V). Please see our supplemental materials for further details.

Implementation details: We use the same hyperparameters and architecture for training π_θ and π_ρ. Following ExAug [34], we set the control horizon $N_s = 8$ and the past observations $N_p = 5$ (see Eqn. 3). For pedestrian detection and tracking, we use the spherical camera on the robot to allow detection and interactions with pedestrians behind it. For the trajectory chaining procedure described in Section V, we merge multiple trajectories across several days from an environment to enhance robustness to visual distractors. We use a batch size of 80, with the training pair (past observations and subgoal images) sampled from the same trajectory for one half of the batch, and the pair coming from different trajectories in the other half of the batch. We empirically set the weights $w_i = 1.5$, $w_{cp} = 10.0$, and $w_{ps} = 100.0$ for each objective, after analyzing closed-loop navigation performance.

For training π_θ, we pre-train f_ψ with $\alpha = N_s - 1$ and $\beta = N_s$ by minimizing the MSE loss using supervised learning and frozen f_ψ while training π_θ. We calculate the gradient of π_θ by back-propagation via f_ψ for updating π_θ. To train a more accurate predictive model, we generate the human and robot trajectories by social force model [1] and mix them with our real data in the batch. One half of the batch is from our real dataset and the other half of the batch is from the social force model. Please see our supplemental materials for more information on the simulation data. Following [48], we set the personal space $\mathcal{R}_h$ as 0.45 and the robot radius $\mathcal{R}_r$ as 0.25 including a small margin. All other hyperparameters are replicated from ExAug [34].

Fig. 6: **Example scenes from the HuRoN Dataset.** We collected our dataset in 5 different environments, spanning over 75 hours of data collection and 4000 rich human interactions, containing raw visual observations (cropped spherical images shown here).

Dataset Characteristics: We collected the HuRoN dataset over the course of 24 days in 5 diverse environments, spread across 3 university buildings. The dataset spans 75 hours and 58 kilometers of autonomous robot navigation trajectories, containing over 4000 interactions with humans. The dataset includes visual observations (spherical and fisheye), 2D LiDAR scans, velocity information, and collision signals from the bumper. Figure 6 shows example images of rich human-robot interactions captured in our dataset.

To evaluate the efficacy of the proposed interaction objective J_{int} (Section V-A), our dataset contains two equal subsets: the *interaction-enriched dataset* corresponding to data collected by the collection policy with interaction objective ($w_i = 1.5$), and the *naïve dataset* collected without ($w_i = 0$). We have released this dataset publicly on our project page.

VI. Evaluation

We design our experiments to evaluate the socially compliant control policy π_θ with our proposed objectives J_{cp} and J_{ps}, as well as the proposed interaction-enriched dataset collected by our autonomous data collection system. Specifically, we study the following questions:

- **Q1.** Does our proposed objective lead to better socially unobtrusive behavior?
- **Q2.** Does our proposed data collection system lead to more interactions, and does this in turn lead to better predictive models of pedestrians?
- **Q3.** How does the navigation capabilities of our policy improve over the course of collecting our dataset?

A. Socially Compliant Navigation

Towards answering **Q1**, we train two different policies with and without our proposed objectives J_{cp} and J_{ps}. Here, the control policy without J_{cp} and J_{ps} corresponds to the most relevant baseline method, ExAug [34]. In addition, we train different social navigation policy on the

TABLE II: **Closed-loop Evaluation of trained control policies.** We evaluate the performance on different training datasets with and without J_{int}. We also evaluate the performance with and without objectives J_{cp} and J_{ps} during training. The policy without J_{cp} and J_{ps} is ExAug.

Method	Training dataset	GR ↑	SPL ↑	STL ↑	CP ↓ [#]	CO ↓ [#]	PSV [s] ↓
ExAug [34]	with J_{int} (ours)	0.800	0.692	0.595	20	6	85.248
Ours	no J_{int} (baseline)	0.667	0.517	0.365	8	11	84.915
Ours	with J_{int} (ours)	**1.000**	**0.888**	**0.692**	**1**	**2**	**57.609**

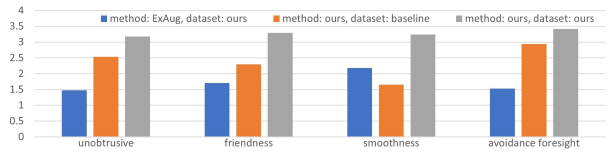


Fig. 7: **Evaluation of socialness by human rating.** The values are average of scores from 17 subjects. Larger is better in all ratings.

naïve dataset without the proposed interaction objective. We conduct fifteen experiments using the real robot across a few days (five experiments in each three difference real environments). In these experiments, we use goal images which were collected over two months ago to evaluate the robustness of the policies to environmental changes. The distance between the start and goal positions ranges from 13.0 to 37.8 meters, which is considered relatively long for vision-based navigation in indoor settings. In order to ensure equivalent experimental conditions, we request during the evaluation that the pedestrians navigate around the robot, creating similar interaction scenarios for each control policy. If the robot collides with a pedestrian or obstacle, we request the pedestrian to distance themselves from the robot's perimeter, and we allow the robot to continue navigation.

Table II presents the comparison of our method to the above baselines along several metrics: Goal arrival Rate (GR), Success weighted by Path Length (SPL) [49], Success weighted by Time Length (STL) [50], Collision count for Pedestrians (CP), Collision count for static Objects (CO), and Personal Space Violation duration (PSV). Our control policy trained on our proposed dataset with J_{cp} and J_{ps} shows a clear improvement over ExAug and the control policy trained on the naïve dataset without J_{int} in all metrics. In particular, our method decreases the collision counts for pedestrians by more than 80%, reduces PSV by over 30%, and successfully leads the robot to the goal position.

Furthermore, we conduct a user study to evaluate the robot's behavior in real-world environments. We recruited 17 subjects from a university campus, encompassing diverse genders, races, and backgrounds; however, there was a bias with 80% being students. We conduct 3 navigation experiments by three different control policies in Table II for each subject (51 experiments in total). We ask them to walk around the robot without explaining which control policy we are running, and we have them evaluate the social compliance and smoothness of our policy between 4 and 0 (larger is better) for four questions after each experiment. Fig. 7 demonstrates the advantage of our method in human ratings across all questions. The comparison suggests that our proposed objective improves the robot's ability to navigate unobtrusively in the presence of humans, and our proposed dataset collected via an interaction-seeking policy leads to better performance for our method.

TABLE III: **Training a Pedestrian Dynamics Model.** We evaluate performance on different training datasets with and without J_{int}, along with the effect of data presence from the social forces model.

Real dataset	Social force model [1]	MSE ↓	Cosine ↑
–	✓	0.0147	0.792
no J_{int} (baseline)		0.0099	0.852
no J_{int} (baseline)	✓	0.0095	0.856
with J_{int} (ours)		0.0084	0.872
with J_{int} (ours)	✓	**0.0083**	**0.876**

In Fig. 8, we qualitatively observe the robot's behavior to be significantly more "compliant" when trained with the interaction-enriched dataset (left). Even in the narrow corridors, our control policy makes space for the pedestrians while still maintaining clearance from the walls. The control policy trained on the naïve dataset does not take avoidance action when a pedestrian approaches the robot, so the robot often violates personal space, collides with the pedestrian (top right), or fails to reach the goal (bottom right).

B. The Value of Interaction-Rich Data

Modeling Pedestrian Dynamics: While the previous evaluation studies the end-to-end performance of our system, in the next experiment we specifically examine the pedestrian prediction model at the core of our method, and how its predictive accuracy changes based on the composition of the training dataset. Our aim is to understand whether our proposed interaction-seeking data collection scheme actually leads to more accurate pedestrian prediction models. For **Q2**, we train the predictive model f_ψ on a combination of three datasets: the interaction-enriched dataset, the naïve dataset, and the simulation dataset from social force model [1]. We report the mean squared error, to capture how close each predicted point is to the true future positions, and the cosine similarity score, that measures the alignment between the vectors corresponding to the predicted and true positions (a scale-invariant metric proposed in GNM [39]).

Table III shows the evaluation results of the predictive model. We find that a predictive model trained with the interaction-enriched dataset leads to better predictions, both in terms of the direction and scale, suggesting that the proposed objective indeed allows better prediction of future human behavior. In addition, its performance is much better than the trained model solely from the simulator. Since the simulation dataset from social force model can help to improve the predictive performance by mixing with our real dataset in training, we use these models for training the socially compliant control policy in Table II. Fig. 9 illustrates the predictive model in action for two example interactions. Estimated trajectories of our predictive model trained on our dataset with J_{int} (magenta) coincide well with the ground truth pedestrians trajectories (red), different from the estimated trajectories trained on the naïve dataset (cyan).

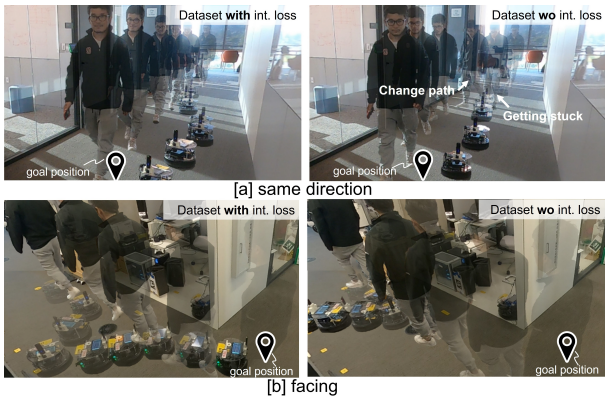


Fig. 8: **Qualitative Examples of Learned Behavior.** A social navigation policy trained on the interaction-enriched subset of HuRoN (left) leads to better handling of human pedestrians while successfully reaching the goal, without intruding in their personal space. Training on the naïve dataset results in a conservative policy (right) that gets stuck and collides with pedestrian.

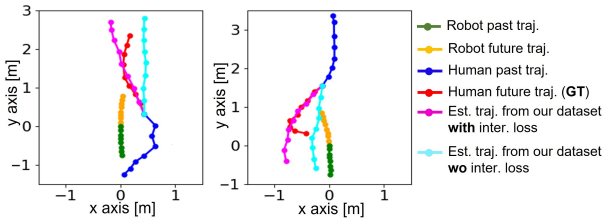


Fig. 9: **Examples of using the learned human dynamics model to predict future human positions**, conditioned on past human positions (blue), past robot positions (green), and future robot plans (yellow). Training with the interaction-enriched dataset (magenta) leads to better predictions than the naïve dataset (cyan).

Moreover, to investigate the effect of the proposed interaction loss on the quality of the data collected, we conduct controlled experiments with 5 human participants tasked with interacting with the data collection system running two different collection policies: one that encourages interactions and another that does not. While quantifying the amount of human interaction is a challenging problem by itself [51], we propose three metrics that coarsely capture these interactions: (i) the mean distance of the robot to an observed pedestrian, (ii) bounding box area (in sq. pixels) of the observed pedestrian, as detected by an object detector [40], and (iii) the offset (in pixels) of the observed pedestrian from the center of the robot's frame (e.g., this would correspond to the visual servoing error for a follower robot [52]). Table IV shows the results of this evaluation on the two subsets of our dataset. We observe the explicit difference, which results in better predictive model and socially compliant control policies.

Continual Learning with the HuRoN System: Lastly, we evaluate how the navigation capabilities of the robotic policy improve over the course of collecting our dataset. While this experiment does not directly evaluate the robot's ability to interact with humans, it does show how our data collection system can enable autonomous improvement, validating the scalability of our data gathering approach for **Q3**.

We deploy HuRoN system to operate autonomously, with occasional remote assistance, to collect data through the

TABLE IV: **Evaluation of the interaction objective.** A policy trained on the interaction-enriched dataset (with J_{int}) drives closer to the pedestrians, and captures more prominent interactions.

Dataset	Distance [m] ↓	Area [px^2] ↑	Offset [px] ↓
no Jint (baseline)	2.67	0.99×10^4	128.51
with J_{int} (ours)	**2.43**	$\mathbf{1.40 \times 10^4}$	**98.55**

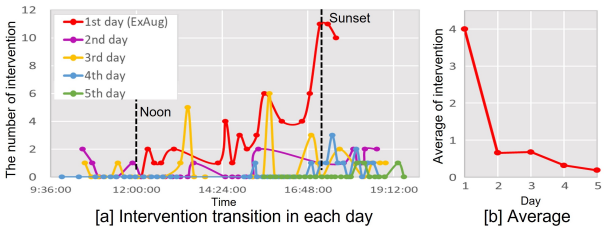


Fig. 10: **Self-Improvement with Continual Learning.** HuRoN can improve with increasing experience, a 95.5% reduction in interventions over the course of 5 days (right). While the performance varies across times of day due to variability in appearances (left, red), continual learning enables our system to eventually learn consistent collision-free behavior across the day (green).

environment. At the end of a collection day, this data is used to *fine-tune* the policy to incorporate the new experience (as described in Sec. V) . Figure 10(a) shows the average number of remote interventions requested by the data collection system during different times of the day. We notice that at the start, the variability in environmental lighting is significant and the initialized model (red) performs significantly worse as the day progresses. However, HuRoN system is able to quickly incorporate this new experience and improve it's performance in subsequent data collection days, requiring fewer interventions each time. Over the course of multiple days (b), our system learns near-perfect autonomous navigation in the challenging indoor environment with dynamic obstacles, requesting an average of 0.18 interventions per a 10 minute trajectory, representing a 95.5% improvement over the day 1 baseline.

VII. Discussion

In this paper, we proposed a method for training the SACSoN policy for vision-based navigation to build the socially unobstrusive navigation system. In training SACSoN policy, we introduced novel objectives using the predictive model of the pedestrians' future trajectories to suppress the counterfactual perturbation from the intended human trajctories. To obtain an accurate predictive model for a better SACSoN policy, we proposed the HuRoN system, a scalable data collection system, to autonomously collect a dataset with enriched human-robot interactions. HuRoN system has the data collection control policy to interact with the pedestrians while collecting the dataset. We used this data collection system to collect the HuRoN dataset: the publicly available dataset of visual navigation around humans, spanning over 75 hours of data collected in 5 different environments and comprising over 4000 rich human-robot interactions. Our experiments show that policies trained on the collected dataset enables the real robot to navigate with the socially unobtrusive behavior.

Our SACSoN policy, when trained on a dataset with enriched human-robot interactions, still has some limitations. Our current system only learns simple social interactions such as avoiding a pedestrian's personal space and giving way to the pedestrians by considering the closest pedestrians' behavior. To understand more complex scenes, we will need to incorporate better objectives accounting for multiple pedestrians and their grouping in the data collection and deployment policies.

We believe that HuRoN opens up many exciting avenues for socially compliant navigation systems in human inhabited spaces. The possibility of scaling such as system to new environments and platforms and objectives is promising. The limitation of the HuRoN dataset we present is that it lacks complex scenes that include groups of multiple pedestrians. Also, the environments in the dataset are limited to office buildings.

ACKNOWLEDGMENTS

This research was supported by Berkeley DeepDrive at the University of California, Berkeley, and Toyota Motor North America. Additionally, partial support for this research was provided by ARL DCIST CRA W911NF-17-2-0181. The authors would like to express their gratitude to Marwa Abdulhai, Qiyang Li, Manan Tomar, Mitsuhiko Nakamoto, Roxana Infante, Ami Katagiri, Katie Kang, Zheyuan Hu, Oier Mees, Jakub Grudzien Kuba, Pranav Atreya, Isadora White, Zhiyuan Zhou, Anjali Thakrar, Niclas Joswig, Kyle Stachowicz, and Catherine Glossop for their valuable assistance in evaluating the SACSoN.

We consulted the Committee for Protection of Human Subjects at our home university (UC Berkeley) and it was determined that the study does not meet the definition of research with human subjects set forth in Federal Regulations at 45 CFR 46.102.

REFERENCES

[1] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995. 1, 2, 5, 6, 11

[2] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1688–1694. 1, 2

[3] R. Martin-Martin, M. Patel, H. Rezatofighi, A. Shenoi, J. Gwak, E. Frankel, A. Sadeghian, and S. Savarese, "Jrdb: A dataset and benchmark of egocentric robot visual perception of humans in built environments," *IEEE transactions on pattern analysis and machine intelligence*, 2021. 1, 2

[4] H. Karnan, A. Nair, X. Xiao, G. Warnell, S. Pirk, A. Toshev, J. Hart, J. Biswas, and P. Stone, "Socially compliant navigation dataset (scand): A large-scale dataset of demonstrations for social navigation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 807–11 814, 2022. 1, 2

[5] A. Rudenko, T. P. Kucner, C. S. Swaminathan, R. T. Chadalavada, K. O. Arras, and A. J. Lilienthal, "Thör: Human-robot navigation data collection and accurate motion trajectories dataset," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 676–682, 2020. 1, 2

[6] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361. 2

[7] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "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. 2

[8] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep visual mpc-policy learning for navigation," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3184–3191, 2019. 2

[9] Z. Yan, S. Schreiberhuber, G. Halmetschlager, T. Duckett, M. Vincze, and N. Bellotto, "Robot perception of static and dynamic objects with an autonomous floor scrubber," *Intelligent Service Robotics*, vol. 13, no. 3, pp. 403–417, 2020. 2

[10] D. Shah, B. Eysenbach, N. Rhinehart, and S. Levine, "Rapid exploration for open-world navigation with latent goal models," in *Conference on Robot Learning*. PMLR, 2022, pp. 674–684. 2

[11] D. M. Nguyen, M. Nazeri, A. Payandeh, A. Datar, and X. Xiao, "Toward human-like social robot navigation: A large-scale, multi-modal, social human navigation dataset," *arXiv preprint arXiv:2303.14880*, 2023. 2

[12] C. Mavrogiannis, F. Baldini, A. Wang, D. Zhao, P. Trautman, A. Steinfeld, and J. Oh, "Core challenges of social robot navigation: A survey," *ACM Transactions on Human-Robot Interaction*, vol. 12, no. 3, pp. 1–39, 2023. 2

[13] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 874–883, 2007.

[14] J. Mumm and B. Mutlu, "Human-robot proxemics: physical and psychological distancing in human-robot interaction," in *Proceedings of the 6th international conference on Human-robot interaction*, 2011, pp. 331–338. 2

[15] D. Mehta, G. Ferrer, and E. Olson, "Autonomous navigation in dynamic social environments using multi-policy decision making," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1190–1197. 2

[16] M. Luber, L. Spinello, J. Silva, and K. O. Arras, "Socially-aware robot navigation: A learning approach," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 902–907. 2

[17] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3931–3936.

[18] A. Bajcsy, S. L. Herbert, D. Fridovich-Keil, J. F. Fisac, S. Deglurkar, A. D. Dragan, and C. J. Tomlin, "A scalable framework for real-time multi-robot, multi-human collision avoidance," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 936–943.

[19] M. Pfeiffer, U. Schwesinger, H. Sommer, E. Galceran, and R. Siegwart, "Predicting actions to act predictably: Cooperative partial motion planning with maximum entropy models," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2096–2101. 2

[20] C. I. Mavrogiannis, W. B. Thomason, and R. A. Knepper, "Social momentum: A framework for legible navigation in dynamic multi-agent environments," in *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 2018, pp. 361–369. 2

[21] X. Xiao, T. Zhang, K. Choromanski, E. Lee, A. Francis, J. Varley, S. Tu, S. Singh, P. Xu, F. Xia, *et al.*, "Learning model predictive controllers with real-time attention for real-world navigation," *arXiv preprint arXiv:2209.10780*, 2022. 2

[22] X.-T. Truong and T.-D. Ngo, ""to approach humans?": A unified framework for approaching pose prediction and socially aware robot navigation," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 10, no. 3, pp. 557–572, 2017. 2

[23] H.-Y. Chen, P.-H. Huang, and L.-C. Fu, "Social crowd navigation of a mobile robot based on human trajectory prediction and hybrid sensing," *Autonomous Robots*, pp. 1–13, 2023. 2

[24] R. Bhaskara, M. Chiu, and A. Bera, "Sg-lstm: Social group lstm for robot navigation through dense crowds," *arXiv preprint arXiv:2303.04320*, 2023.

[25] V. Narayanan, B. M. Manoghar, R. P. RV, and A. Bera, "Ewarenet: Emotion-aware pedestrian intent prediction and adaptive spatial profile fusion for social robot navigation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7569–7575.

[26] C. Rösmann, M. Oeljeklaus, F. Hoffmann, and T. Bertram, "Online trajectory prediction and planning for social robot navigation," in *2017*

IEEE International Conference on Advanced Intelligent Mechatronics (AIM). IEEE, 2017, pp. 1255–1260.

[27] Z. Chen, C. Song, Y. Yang, B. Zhao, Y. Hu, S. Liu, and J. Zhang, "Robot navigation based on human trajectory prediction and multiple travel modes," *Applied Sciences*, vol. 8, no. 11, p. 2205, 2018. 2

[28] Y. F. Chen, M. Everett, M. Liu, and J. P. How, "Socially aware motion planning with deep reinforcement learning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 1343–1350. 2

[29] Y. F. Chen, M. Liu, M. Everett, and J. P. How, "Decentralized non-communicating multiagent collision avoidance with deep reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 285–292.

[30] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3052–3059.

[31] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *2019 international conference on robotics and automation (ICRA)*. IEEE, 2019, pp. 6015–6022.

[32] Y.-J. Mun, M. Itkina, S. Liu, and K. Driggs-Campbell, "Occlusion-aware crowd navigation using people as sensors," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12 031–12 037. 2

[33] J. Biswas and M. M. Veloso, "Localization and navigation of the cobots over long-term deployments," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1679–1694, 2013. 2

[34] N. Hirose, D. Shah, A. Sridhar, and S. Levine, "Exaug: Robot-conditioned navigation policies via geometric experience augmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 4077–4084. 2, 3, 5, 6

[35] N. Savinov, A. Dosovitskiy, and V. Koltun, "Semi-parametric topological memory for navigation," in *International Conference on Learning Representations*, 2018.

[36] X. Meng, N. Ratliff, Y. Xiang, and D. Fox, "Scaling local control to large-scale topological navigation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 672–678.

[37] D. Shah and S. Levine, "Viking: Vision-based kilometer-scale navigation with geographic hints," *Robotics: Science and Systems*, 2022.

[38] N. Kim, O. Kwon, H. Yoo, Y. Choi, J. Park, and S. Oh, "Topological semantic graph memory for image-goal navigation," in *Conference on Robot Learning*. PMLR, 2023, pp. 393–402. 2

[39] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "Gnm: A general navigation model to drive any robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7226–7233. 3, 6

[40] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788. 3, 7

[41] "Object detection by yolov5," https://github.com/ultralytics/yolov5. 3

[42] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649. 3

[43] "Pedestrian detection and tracking by yolov5 and deepsort," https://github.com/HowieMa/DeepSORT_YOLOv5_Pytorch. 3

[44] T. Niwa, S. Taguchi, and N. Hirose, "Spatio-temporal graph localization networks for image-based navigation," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 3279–3286. 4, 5

[45] "Ros wrapper for alvar, an open source ar tag tracking library," http://wiki.ros.org/ar_track_alvar. 4, 10

[46] Y. Chebotar, K. Hausman, Y. Lu, T. Xiao, D. Kalashnikov, J. Varley, A. Irpan, B. Eysenbach, R. Julian, C. Finn, *et al.*, "Actionable models: Unsupervised offline reinforcement learning of robotic skills," *arXiv preprint arXiv:2104.07749*, 2021. 4

[47] "Roomba Drivers," https://github.com/AutonomyLab/create_robot. 5

[48] N. Tsoi, M. Hussein, O. Fugikawa, J. Zhao, and M. Vázquez, "An approach to deploy interactive robotic simulators on the web for hri experiments: Results in social robot navigation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 7528–7535. 5

[49] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018. 6

[50] A. Francis, C. Pérez-d'Arpino, C. Li, F. Xia, A. Alahi, R. Alami, A. Bera, A. Biswas, J. Biswas, R. Chandra, *et al.*, "Principles and guidelines for evaluating social robot navigation algorithms," *arXiv preprint arXiv:2306.16740*, 2023. 6

[51] J. Wang, W. P. Chan, P. Carreno-Medrano, A. Cosgun, and E. Croft, "Metrics for evaluating social conformity of crowd navigation algorithms," in *2022 IEEE International Conference on Advanced Robotics and Its Social Impacts (ARSO)*. IEEE, 2022, pp. 1–6. 7

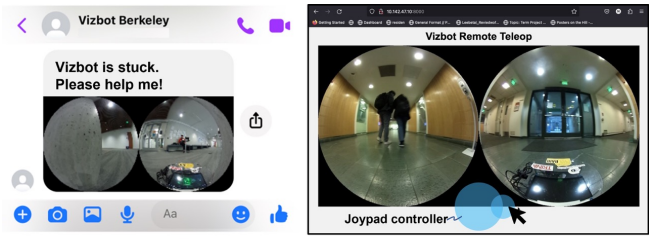
[52] J. L. Giesbrecht, H. K. Goi, T. D. Barfoot, and B. A. Francis, "A vision-based robotic follower vehicle," in *Unmanned Systems Technology XI*, vol. 7332. SPIE, 2009, pp. 451–462. 7

APPENDIX

A. Localization with Long-term Anchors

In order to avoid navigation failure by the localization errors, we placed some AR tags along the topological graph to assist localization. Our idea is simply overriding the estimated node number by the node number associated with the AR tags. When collecting the topological graph, we also save the list of $\{n_i^{ar}, p_i^{ar}, n_i^{node}\}_{i=1...N_{ar}}$. Here n_i^{ar} and p_i^{ar} are the detected AR tag number and its pose on the robot coordinate, respectively [45]. n_i^{node} is the node number on the topological graph, which detects the AR tag of n_i^{ar}.

We basically override the estimated node number by n_j^{node} when detecting AR tag of n_i^{ar} in the data collection. If the multiple node images detect the same AR tag in the topological graph, we use the closest one to assist moving forward. However, the mobile robot may pass over the node location linked to the AR tag and still detect its AR tag. Such a case causes unnatural movement like stopping abruptly because the subgoal image will be behind the current robot pose. To avoid unnatural behavior, the robot compares the estimated pose of AR tag with p_i^{ar} to detect whether the its is passing by the tag. If the robot is passing by, it is overwritten with the next node number n_i^{ar} + 1.



[a] screen shot of facebook messenger [b] Application for remote teleoperation

Fig. 11: **Help-and-rescue module.** The robot messages the operator for assistance when stuck (left), and can be rescued remotely over the internet using our web interface (right).

B. Help-and-rescue module

In the help-and-rescue module, we implement a pipeline for autonomous recovery from collisions and seeking remote help in case of irrecoverable collisions for the challenging obstacles (e.g., that may be shorter than the camera height, made of glass etc.). When a collision is detected by the robot's collision detector sensor (e.g., a mechanical contact sensor), an automatic backup maneuver is executed. This maneuver drives the robot away from the obstacle for a short distance along the normal vector corresponding to the point of contact. Specifically, the robot moves back about 0.5 meter and rotates about 45 degrees at a point. The direction of rotation is determined by the detection of two bumper sensors in right and left. If the left sensor detects a collision, the robot rotates to the right; if the right sensor detects a collision, the robot rotates to the left.

This allows the robot to automatically recover from 70% of the simple collisions where the robot accidentally runs into challenging obstacles (e.g., that may be shorter than the camera height, made of glass etc.). Complete autonomy, however, may not be possible to achieve. The robot may drive itself into a convex hull of multiple obstacles, leading to repeated collisions, or get it's wheels stuck (e.g., on an air vent) and be unable to rescue itself. Only for these accidental cases, we use a messaging and remote teleoperation interface in Fig. 11 to recover and continues the data collection without any physical interventions.

C. Trajectory Chaining for Continual Learning

To chain the different sequences in training, we need to take T_{gt} between current and subgoal image from different sequences. Fig. 12 visualizes how to obtain T_{gt} from different sequences. Since we place AR tags along the topological graph to assist the localization module, some frames in our dataset detect AR tag and estimate the relative pose for each AR tag. In Fig. 12, T_{mc} and T_{mg} indicate the estimated relative pose against same AR tag from different sequence s_c and s_g. Here, n_c and n_g are corresponding node number on s_c and s_g.

To take various pairs of current and subgoal images, we randomly select two step numbers within $N_m = 18$ as n_{cr} and n_{gr} and decide the node number of the current image as n_c - n_{cr} on s_c and the node number of the subgoal image as n_g + n_{gr} on s_g, respectively. The sign of n_{cr} and n_{gr} are decided so that the subgoal image position is forward with respect to the current image position. Note that we assume that the dataset can be collected with a positive linear velocity. Since N_m is not large number, we can have accurate relative pose T_{oc} between $n_c - n_{cr}$ and n_c, and an accurate relative pose T_{og} between n_g and $n_g + n_{gr}$ from the odometry. As the result, we calculate T_{gt} between $n_c - n_{cr}$ and $n_g + n_{gr}$ as $T_{gt} = T_{oc} \cdot T_{mc} \cdot T_{mg}^{-1} \cdot T_{og}$.

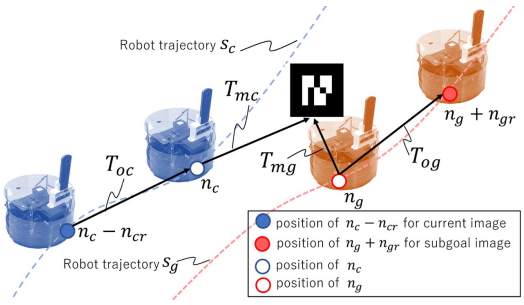


Fig. 12: **Relative positions from different sequences using AR tags.** We obtain the relative pose using the odometry of the robot and the detected pose of the AR tag for continuous learning.

D. Network structures

Figure 13 describes the neural network architecture of π_θ. An 8-layer CNN is used to extract the image features z from the image history $I_{t:t-N_p}$ and the subgoal image I_g, with each layer using BatchNorm and ReLU activations. Following our previous work, ExAug, the predicted velocity commands $\{v_i, \omega_i\}_{i=1...N_s}$ from 3 fully-connected layers "FCv" are conditioned on the robot size $\{r_s, v_l\}$ and z. A scaled tanh activation is given to limit the output velocities

as per the specified constraints. We can control the robot by giving v_1 and ω_1 as the actual robot velocity command.

In addition to the core part of our control policy, we can implement "FCt" to estimate traversability $\{t_i\}_{i=1...N_s}$, following ExAug. We integrate the velocities to obtain waypoints predictions and feed them to a set of fully-connected layers "FCt" along with the observation embedding z and target robot size r'_s, followed by a sigmoid function to limit $t_i \in (0, 1)$. Although $r_s = r'_s$ in training, we found the flexibility of an independent $r'_s \neq r_s$ crucial to the collision-avoidance performance of our system in inference. Note that we can remove the gray color part to construct π_θ for the minimum implementation.

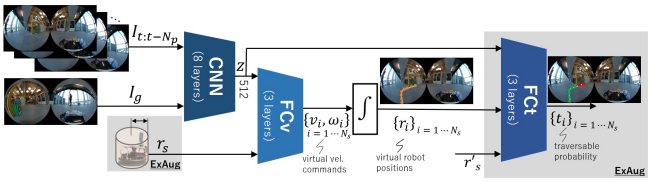


Fig. 13: **Network structure of our control policy.**

In our evaluation section, we train the predictive model f_{θ_p} for the pedestrians dynamics. Fig. 14 is the network structure of f_{θ_p}. At first, we feed the concatenated past human trajectory $h_{t-\alpha:t-1}$ and the past robot trajectory $r_{t-\alpha:t-1}$ into "FC1" with the three fully connected layers using BatchNorm and ReLU activations to extract the features z_p. Then, we predict the human future trajectory condition on the robot actions (=future trajectories) by giving z_p with $r_{t-1:t+\beta}$. Here, the last layer of "FC2" with three fully connected layers has the tanh activation to limit the human velocity within $\pm$ 1.5 m/s.

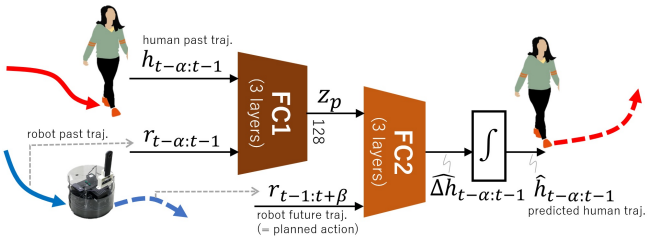


Fig. 14: **Network structure of our model to predict pedestrians dynamics.**

E. Simulation dataset from social force model

To evaluate the effectiveness of SACSoN dataset, we generate the pedestrian trajectories and the robot trajectories from the social force model [1]. In addition, we mix this simulation dataset with the real data from the SACSoN dataset in training to improve the accuracy of the predictive model for the pedestrians' future trajectories. In this appendix, we show the implementation details to generate the simulation dataset.

We set two agents (the robot and the pedestrian) with different initial velocity: 0.8 m/s for the pedestrian and 0.3 m/s for the robot toward the goal position, because the pedestrian is much faster than our robot in our case. Note that the initial velocity decides a nominal velocity for each agent, not a maximum velocity. To simulate these agents in each scenario, we randomly place these agents on their own circle with varying radii, such that the two circles centered at the origin. We decide the robot's and pedestrian's goal position as the opposite side of their respective circles. However, we randomly shorten the goal position for the robot to stop before arriving at the original goal to emulate giving way to the pedestrian. We decide the radius for the robot's circle as 2.0 m and the radius for the pedestrian's circle as 5.3 m. Since the center of these circles is the origin, the robot and the pedestrian often has the interaction around the origin, because the radius for the pedestrian 5.3 m is calculated as $\frac{0.8}{0.3}\times$ 2.0 m. We run the social force model with these hyperparameters for 80 steps and collect 10000 scenarios.

In training, we randomly choose the scenario to make the batch. Since our predictive model estimates the pedestrians trajectory on the robot local coordinate, we transform the sampled trajectories before making batch. In Fig. 15, we show the examples of the simulation dataset from the social force model.

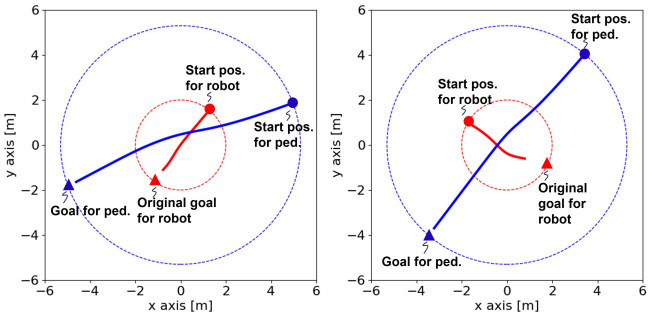


Fig. 15: **Examples of simulation dataset from sorcial force model.**