Cross-Modal Contrastive Learning of Representations for Navigation using Lightweight, Low-Cost Millimeter Wave Radar for Adverse Environmental Conditions

Jui-Te Huang, Chen-Lung Lu, Po-Kai Chang, Ching-I Huang, Chao-Chun Hsu,
Zu Lin Ewe, Po-Jui Huang, and Hsueh-Cheng Wang*

Abstract—Deep reinforcement learning (RL), where the agent learns from mistakes, has been successfully applied to a variety of tasks. With the aim of learning collision-free policies for unmanned vehicles, deep RL has been used for training with various types of data, such as colored images, depth images, and LiDAR point clouds, without the use of classic map–localize–plan approaches. However, existing methods are limited by their reliance on cameras and LiDAR devices, which have degraded sensing under adverse environmental conditions (e.g., smoky environments). In response, we propose the use of single-chip millimeter-wave (mmWave) radar, which is lightweight and inexpensive, for learning-based autonomous navigation. However, because mmWave radar signals are often noisy and sparse, we propose a cross-modal contrastive learning of representations (CM-CLR) method that maximizes the agreement between mmWave radar data and LiDAR data in the training stage. We evaluated our method in real-world robot compared with 1) a method with two separate networks using cross-modal generative reconstruction and an RL policy and 2) a baseline RL policy without cross-modal representations. Our proposed end-to-end deep RL policy with contrastive learning successfully navigated the robot through smoke-filled maze environments and achieved better performance compared with generative reconstruction methods, in which noisy artifact walls or obstacles were produced. All pretrained models and hardware settings are open access for reproducing this study and can be obtained at `https://arg-nctu.github.io/projects/deeprl-mmWave.html`.

I. Introduction

Navigation and collision avoidance are fundamental capabilities for mobile robots, and state estimation and path planning, which are integral to such capabilities, have undergone much development. Deep reinforcement learning (RL) is another method for realizing these capabilities, where robots learn to solve problems through trial and error. Extensive efforts have been made to formulate training methods for deep RL models using various types of data, such as those from RGB cameras, depth images [1], [2], and laser range finder (LiDAR) point clouds [3], [4]. However, optical sensors are often useless in adverse weather and environmental conditions: LiDAR inputs are degraded by rain, fog, and dust, and cameras are sensitive to variable lighting in dark and sunny weather. In search
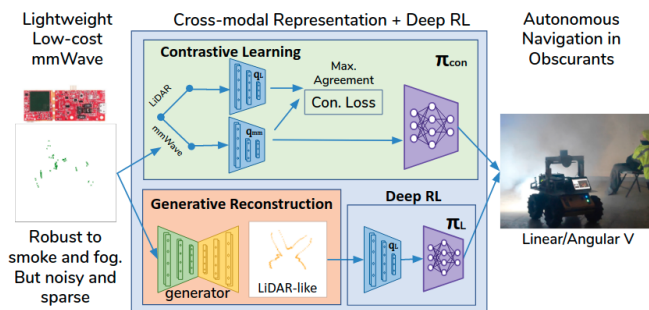
Fig. 1: Cross-modal contrastive learning of representations is proposed to generate representations of range data from mmWave radars, which can operate even in smoke-filled environments. The proposed method is compared with a generative reconstruction model.

and rescue missions, for example, adverse environmental conditions are commonly encountered and result in degraded sensing. In the DARPA Subterranean Tunnel Circuit, there are challenging environments in unknown and long coal mine tunnels with smoke filled adverse conditions that impair commonly used camera and LiDAR sensors.

Millimeter wave (mmWave) radar is an alternative sensor that has promise for overcoming these challenges that can operate despite variable weather and lighting conditions [5], [6]. Recently, low-cost, low-power, and lightweight single-chip mmWave radar has emerged as an innovative sensor modality. Methods are proposed to provide state estimation [7] through neural networks or reconstruct dense grid maps [8] with conditional GAN. Although the results showed promise, navigation using maps reconstructed from mmWave data faces problems from sensor noise and multipath reflections, which may be mistakenly rendered as obstacles on the reconstructed map. Another thought is to have the RL agent learn the noise model of mmWave radars and correspond accordingly. However, directly training an end-to-end deep RL policy network in a simulation using mmWave data requires some channel models [9] that are not yet available in commonly used simulation frameworks in the robotics literature. Therefore, a large amount of mmWave training data from the real world for offline RL training is needed which makes the method inefficient to almost infeasible.

To this extent, we explored contrastive learning to learn cross-modal representations between mmWave radar and LiDAR. Cross-modal representations were analyzed in

[10], where the authors proposed a variational autoencoder (VAE)–based method, which was to be applied in real-world aerial navigation, for learning robust visuomotor policies from simulated data alone. In our study, we leveraged the contrastive learning framework by regarding mmWave radar data as an augmentation of LiDAR data. Specifically, an encoder was trained to learn a sufficient number of patterns in mmWave data for the RL agent, which then navigated an unmanned ground vehicle (UGV) using only noisy raw radar inputs with no reconstruction.

We used lightweight and low-cost single-chip mmWave sensors, which can operate under challenging foggy or smoke-filled environments. To enable navigation despite noisy and low-resolution mmWave radar inputs, we proposed approaches for learning how to control UGVs and evaluated these approaches in a smoke-filled environment. We summarize our contributions as follows:

1) **We showed how cross-modal contrastive learning of representations (CM-CLR) can be used to maximize the agreement between mmWave and LiDAR inputs**: We introduced contrastive loss into our network and formulated a training method for achieving consistency between mmWave and LiDAR latent representations during forward propagation. We collected a dataset comprising both mmWave and LiDAR inputs in various indoor environments, such as corridors, open areas, and parking lots. All LiDAR data were only used for training.

2) **We demonstrated that deep RL with cross-modal representations can be used for navigation**: The representation obtained from contrastive learning was integrated into an end-to-end deep RL network π_{con} (Fig. 1). We demonstrated that inputs from only mmWave radar signals enabled navigation, unlike LiDAR and camera signals, in visually challenging smoke-filled maze environments.

3) **We evaluated our approaches in real-world smoke-filled environments**: We comprehensively evaluated a) the proposed end-to-end contrastive learning method and b) deep generative models that generate LiDAR-like range data as inputs for a deep RL–based control policy π_L network. We used a conditional generative adversarial network (cGAN) and VAEs, where raw mmWave inputs are used to reconstruct dense LiDAR-like range data. We showed that the navigation achieved by our proposed method can better avoid obstacles and respond to situations in which the vehicle is trapped.

II. RELATED WORK

A. Contrastive Learning

Contrastive learning is a type of learning framework where the model learns patterns in a dataset, which are organized in terms of similar versus dissimilar pairs, while subject to similarity constraints. In [11], a simple contrastive learning framework (SimCLR) was proposed for learning visual representations, with the application of simple data augmentation for contrastive prediction tasks, and the authors incorporated cosine similarity temperature-scaled cross-entropy loss into

their method. The performance of the proposed framework matched that of supervised learning methods. That study further analyzed the effects of different data augmentation operators and concluded that a combination of cropping and color distortions maximizes learning efficiency. Another study [12] proposed the CURL framework for learning with high-dimensional inputs (i.e., images), in which contrastive learning, for representation learning, is combined with RL, for control policy learning. The framework outperformed state-of-the-art image-based RL in several DeepMind control suite tasks and Atari game tasks and even nearly matched the sample efficiency of using state-based features for RL. In this work, we considered mmWave data to be an augmentation of LiDAR data applied a contrastive learning framework to learn cross-modal representations through maximization of the agreement between encoded mmWave radar data and LiDAR data.

B. Deep Reinforcement Learning for Navigation

Recent studies have trained deep networks either through a pre-generated occupancy map for motion commands or a mapless end-to-end approach that relates range inputs to actions. Kahn *et al*. [13] demonstrated affordance prediction for future paths with a network trained through self-supervised learning, which allowed for navigation in paths that were previously regarded as untraversable in the classic SLAM and planning approach. Niroui *et al*. [14] presented a map-based frontier prediction network that was trained using the RL framework in urban search and rescue applications. Both of these works combined learning modules with classic control modules, in which deep networks are not responsible for predicting motion commands. On the other hand, [15] proposed a method for multi-robot navigation with a high-level voroni-based planar and a low-level deep RL-based controller, where the RL agent is only responsible for the controller. Mapless end-to-end approaches (e.g., in [3], [4], [16]) use range data as inputs to predict suitable motion commands through a trained network. Pfeiffer *et al*. [3] developed a network that learns through feature extraction in a convolutional neural network (CNN) layer and through decision making in fully connected (FC) layers using data-driven behavioral cloning. In another study [16], an agent network was trained through RL algorithms [17], [18] in virtual environments. Another study [4] pretrained the agent network using behavioral cloning prior to training with the RL framework. Finally, in two studies [4], [16], further downsampling was performed or fewer range data points were used to avoid overfitting or obtain better sim-to-real performance.

However, all these studies used LiDAR devices or cameras as input sensors, which are vulnerable in obscurant-filled environments. There has been earlier work using an array of sonars [19], [20], which used the propagation of acoustic energy and could penetrate smoke to sense obstacles from the environment [21]. However, low angular resolution and short range limited the applications. Thus, we used mmWave radar instead of LiDAR for range sensing because mmWave signals can penetrate smoke and fog. Our network architecture
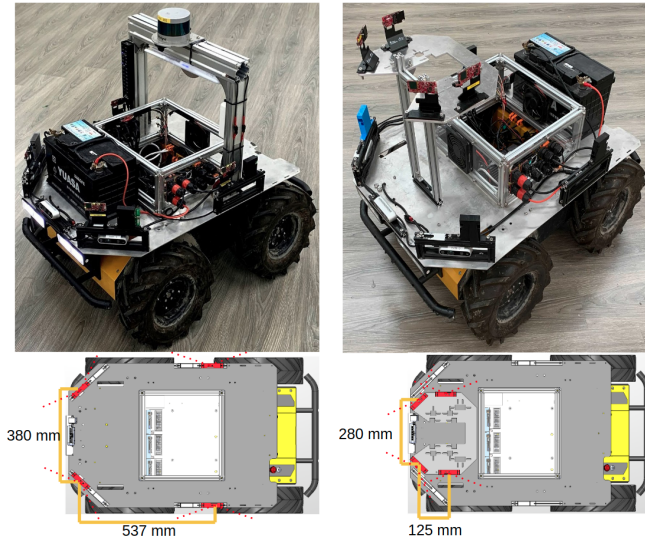
(a) Sensor configuration 1. (b) Sensor configuration 2.

Fig. 2: Two configurations of mmWave radar modules placements mounted on a Clearpath Husky. The field of views of both configurations were constrained to 240°. In (a) the Velodyne 3D LiDAR was only for data collection. All experiments and data collection in the paper were conducted using configuration 1. (b) Configuration 2 was designed for the potential use of smaller robots, and the discussions are described in Sec.V-E

is similar to that proposed in [3], which included feature extraction CNN layers and decision-making FC layers.

III. DEEP REINFORCEMENT LEARNING WITH CROSS-MODAL CONTRASTIVE LEARNING

A. Pre-processing

We projected all range data to a two-dimensional (2D) plane with a 240° front-facing field of view and 1° resolution. Our range data were obtained from LiDAR as $x_l \in \mathbb{R}^{241}$ and mmWave as $x_m \in \mathbb{R}^{241}$ at each time step. Each of the four mmWave radars were running at 15 Hz, and synchronized into 10 Hz. We then applied an outlier removal filter by clustering the point clouds before projection. To do so, we first accumulated 5 consecutive frames to increase features. Then each point would be kept if there exist 3 neighbor points within 1.2 meter radius. There may exist missing value at a certain angles after applying the filtering, and we applied a padding of a maximum range value of 5 m to ensure sufficient input dimensions.

B. Contrastive Learning

To overcome the problem of noisy and sparse mmWave radar signals in navigation, we proposed contrastive learning, the agreement between mmWave and LiDAR inputs is maximized through the tuning of a trainable encoder with contrastive loss.

Intuitively, learning a control policy can be done directly from mmWave-obtained range data, without the use of LiDAR data. However, there exist two issues of noise and lack of simulator access. Training using real-world data, which lack false cases (e.g., running into walls), results in agents
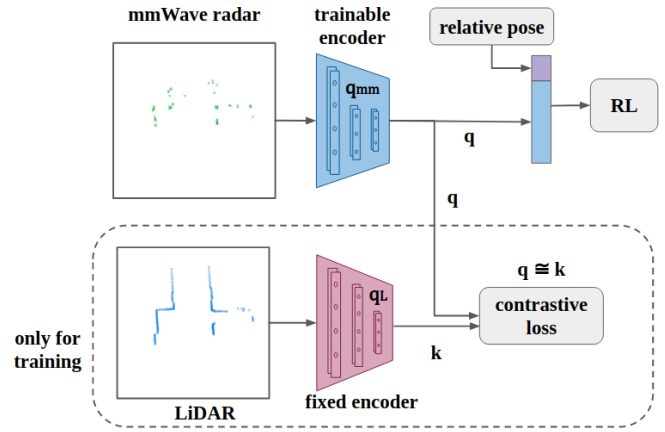

Fig. 3: Query encoder for mmWave radar range data q_{mm} is used to generate a representation q that is similar to the representation k generated by the key encoder q_L, which are trained from LiDAR range data in simulation. The weights of q_L are set unchanged during contrastive learning.

that have a high frequency of collisions with obstacles. In contrastive learning, the agent learns a good representation of some high-dimensional inputs through different forms of data augmentation and contrastive loss. We regarded mmWave range data as a noisy and randomly dropped augmentation of LiDAR range data, and we trained an encoder using mmWave data to generate a latent representation that is identical to that from the LiDAR ground truth. Because the agent in the simulation has learnt a good navigation strategy with such a representation, our new agent should, in theory, be capable of navigating highly similar representations that are generated from the trained encoder and mmWave input.

C. Learning from LiDAR Data for Navigation in a Virtual Subterranean Environment

First, we aimed to train the agent to navigate in long corridors, tunnels, or caves in subterranean environments using RL algorithms. The reward settings were expected to encourage the agent to move forward and avoid collisions at the same time to obtain high rewards. The agent should also receive some rewards to keep moving and exploring. High angular velocity was expected to decrease the rewards so that the agent will not make sharp turns unless there was no opening sensed at front. The model was used as a low-level collision avoidance controller instead of a high-level planar.

To do so, we used a variety of virtual subterranean environments [22] in Gazebo simulation software, comprising an artificial tunnel with obstacles and cliffs, a subway station, and a natural cave. We used the cave environment because it has the most diverse corridor geometries and terrains. We followed the OpenAI Gym interfaces to integrate our gym environment with Gazebo. We analyze data on the agent's observations, calculate rewards, assign actions, and reset the agent. The settings and justifications for them are as follows:

- Observations: The three-dimensional simulated LiDAR inputs constituted an observation space. As stated in Section III, we sampled the range data for −120° to 120°, with a 1° resolution from the point cloud of a

height reaching the UGV's height, resulting in a total of 241 range values. The agent is also given the information of motion direction, the relative 2D position of the agent is defined with respect to the previous time point by $x_p \in \mathbb{R}^2$. The total observation O of our RL agent is the combination of x_l and x_p, which has the dimension of $O \in \mathbb{R}^{243}$.

- Actions: A continuous space of linear and angular velocities was used to represent actions. The linear velocity space was normalized to $[0, 1]$ to prevent the agent from moving backward, whereas the angular velocity space was normalized to $[-1, 1]$. Angular velocity was represented as a continuous rather than discrete space to allow for smooth motion (which is desirable).

- Reward: We designed the rewards according to three preferences. The first, $r\_forward$, incentivizes the agent to move straight and perform as few turns as possible, where ω is the angular velocity. The second, r_move, encourages the agent to keep moving. Finally, $r\_collision$ punishes the agent from colliding into obstacles, which results in the end of an episode.

$$r\_forward = 2^{(1-|\omega|) \times 5}$$

$$r\_move = \begin{cases} 10 & if \quad \text{the distance} \\ & \quad \text{between frames} > 0.04m \\ -10 & else \end{cases}$$

$$r\_collision = \begin{cases} -50 & if \quad \text{collision} \\ 0 & else \end{cases}$$

Finally, these three rewards were combined with a discount factor λ on $r\_forward$ into r, which was then sent to the agent as r_agent using a log scale.

$$r = \lambda \times r\_forward + r\_move + r\_collision$$

$$r\_agent = sign(r) \times log(1 + |r|)$$

We used the deep deterministic policy gradient (DDPG) [17] and the recurrent deterministic policy gradient (RDPG) [23] algorithms to train collision avoidance policies with data collected in the Gazebo virtual subterranean environment. DDPG is an off-policy RL algorithm that trains a deterministic policy to be the agent's control policy. DDPG uses an actor-critic method to learn a deterministic policy π^θ with parameter θ and Q-value estimator Q^ϕ with parameter ϕ. In contrast to DDPG, where the network is updated using transitions $t = (o_t, a, r, o_{t+1})$ sampled from the experience replay buffer, RDPG uses the recurrent network and trains the network with an entire trajectory history $h_t = (o_1, a_1, o_2, a_2, \ldots, a_{t-1}, o_t)$ and back propagation through time. The critic network updates ϕ to minimize the Bellman error, and the actor network's objective is to maximize $Q^\phi(h_t, a_t)$.

Our network architecture was similar to those of [3] and [24], which used three consecutive frames, whereas our network includes LSTM cells. Studies have reported the importance of including a recurrent network when learning sequential actions. Our neural network architecture comprised minimum pooling for downsampling the inputs, two convolution layers for feature extraction, two FC layers, one recurrent layer with an LSTM cell, and one FC layer to generate action outputs. Training process took 55 hours for 4 million steps on an Intel i7-8700 CPU with Nvidia RTX2070 GPU. We denoted our control policy trained in the simulation using LiDAR data as π_L.

D. Navigation with Policy From Contrastive Learning

To calculate the contrastive loss, we separated the original RL network into a feature extraction network of convolution layers, and a decision-making network, which comprised the FC layers and the recurrent layer. The goal of the contrastive learning was to maximize the similarity of the encoded representations between the mmWave input and the matching ground-truth LiDAR input; this was done to allow the mmWave representation to be used by the decision-making network, which was trained with RL in the simulation. Thus, the key encoder $q_L(k|x_l)$ and the query encoder $q_{mm}(q|x_m)$ of contrastive learning were designed as the architecture of the feature extraction layers of the RL policy network. During contrastive learning, real-world LiDAR range data were used as the input for the key encoder. The mmWave range data, which can be construed as an augmentation of ground truth LiDAR data, were taken as the input of the query encoder. In CURL [12], the bilinear inner product with a momentum key encoder was used for training with contrastive loss. However, we froze the parameters of the key encoder, which was already trained with RL during simulation. We used a fixed key encoder with euclidean distances to measure the similarity between k and q.

$$\mathcal{L}_{contrastive} = \mathbb{E}_{x_l, x_m}[(q_{mm}(q|x_m) - q_L(k|x_l))^2]$$

We trained the network with the collected dataset to converge (20 epochs), and stopped the training to prevent overfitting. The performance of the trained mmWave control policy π_{con} was evaluated in a real-world maze and an indoor corridor.

IV. GENERATIVE RECONSTRUCTION BASELINES

We considered generative reconstructions of mmWave range data into LiDAR-like data to serve as a key baseline because prior work [8] showed promising results. We implemented two state-of-the-art supervised deep generative models, which were used as inputs of the baseline control policies for navigation tasks.

A. Conditional Generative Adversarial Network for Range Data

A cGAN comprises a generator G and a discriminator D. Given an mmWave range data point x_m and a random vector z as the input, the generator attempts to generate a less noisy and denser range data point y that is similar to the ground truth LiDAR range data point x_l. By contrast, the discriminator attempts to distinguish the real LiDAR range data point x_l generated from y. The objective of a cGAN can be expressed as follows.

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x_m, y}[logD(x_m, y)] + \mathbb{E}_{x_m, z}[log(1 - D(x_m, G(x_m, z)))]$$

To make the generated range data even more similar to the LiDAR ground truth, we adapted L1 loss in the objective function following the pix2pix approach of a previous work [25].

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x_m, y, z}[|y - G(x_m, z)|]$$

The final objective function for the generator is as follows.

$$\mathcal{L}_{total} = \mathcal{L}_{cGAN} + \lambda_1 \mathcal{L}_{L1}$$

We designed the generator to have an encoder–decoder structure. The encoder has the same convolution feature extraction architecture as that of our deep RL networks. For the decoder model, we implemented three deconvolution layers and an FC layer to adjust the output dimension. For our discriminator, we used the Markovian discriminator technique [26], also called patchGAN in pix2pix [25]. The patched discriminator has four layers of one-dimensional convolution for feature extraction, and the output is a 1×14 patch. We also tested a nonpatched discriminator that outputs a single value to discriminate real objects from counterfeit ones. We found that the nonpatched discriminator had slightly better L1 performance. We trained the model using our collected dataset. The data collection process is described in Section V-A.

B. Variational Autoencoder

In addition to a cGAN, we used a similar network architecture to train a VAE [27], which comprised a probabilistic encoder $q(z|m)$ and a generative model $p(l|z)$, where z is the latent variable, sampled from the output of the encoder. The objective function of the VAE minimizes the variational lower bound of the log likelihood. In [28], the objective function was interpreted as accounting for probabilities across data modalities. The objective function in our case is as follows.

$$\mathbb{E}_{z \sim q(z|x_m)} \left[log p(x_l|z) - D_{KL}(q(z|x_m)||p(z)) \right]$$

V. EVALUATION

A. Data Collection

Range data from four 60-GHz TI 6843 mmWave radar modules and a Velodyne VLP-16 LiDAR and motion commands were recorded at a frame rate of 5 Hz as training data using a Clearpath Husky. A total of 110,381 frames (approximately 6 robot hours) were recorded. The experimental environments included a corridor, a parking area, and outdoor open areas. The indoor corridors had widths from 2 to 4 m; the parking area had a wide passage with a maximum width of 6 m, and the outdoor open area had a maximum width of 10 m. All data in this study, which comprised the aforementioned types of data as well as images and wheel odometry, are open source and in ROS bag and hdf5 file formats.

B. Reconstruction Results

In addition to the previously mentioned reconstruction methods, the polynomial curve fitting method in the polar coordinate was used as a baseline. Among the 1st-order to 15th-order curves, the 8th-order curve had the lowest L1 distance value. To evaluate the performance of different
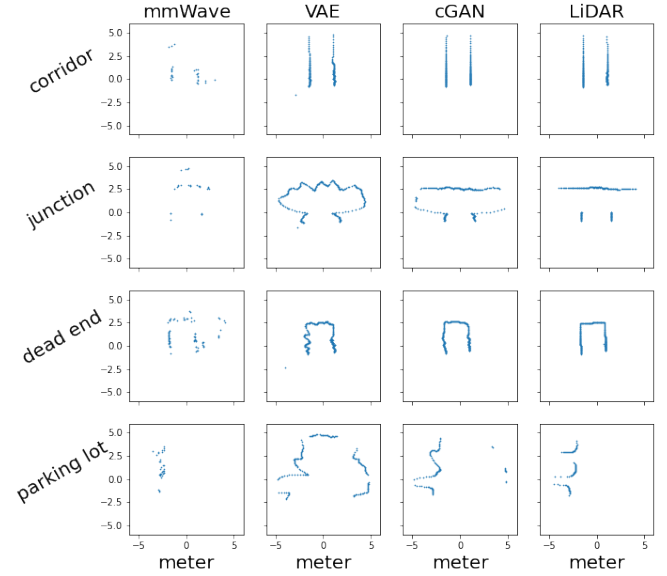

Fig. 4: Raw mmWave range data and ground truth LiDAR range data collected in different environments with different reconstruction methods.

reconstruction methods, we compared the L1 distance between the reconstructed mmWave data and the ground truth LiDAR data. The results are displayed in Table I. The qualitative results of different reconstruction methods are illustrated in Fig. 4. Data collected in scenes that were from a corridor or parking lot and were not a maze used in the training steps were all considered in the evaluation.

TABLE I: L1 loss of various reconstruction methods in different environments.

Method	Corridor	Parking Lot	Maze
raw mmWave	1.99	0.39	2.53
curve fitting	0.75	1.52	1.84
VAE	0.23	0.27	1.65
cGAN	**0.09**	**0.11**	**1.53**

As indicated in Table I, the raw mmWave data had a lower L1 distance in the parking lot than in the other environments. This is because the parking lot had wider passages, which caused most range data points to exceed the stipulated maximum distance of 5 m. Thus, most mmWave data and LiDAR ground truth data points were forced to be at 5 m, which resulted in a low L1 distance, regardless of whether the value was absent or higher. Among all the reconstruction methods, the cGAN had the lowest L1 distance, which implies that the reconstruction data of the cGAN were most similar to the ground truth LiDAR data. Both the cGAN and VAE could generate geometries that were similar to the ground truth, but the cGAN outperformed the VAE by generating straighter, smoother, and more consistent walls and fewer "phantom walls." We found that both methods had difficulty reconstructing the exact shapes of cars in the parking lot scene using sparse mmWave data. By contrast, reconstructing the geometry of corridors was easy, as indicated by the lower L1 loss in the corridor scene. We noted that the average performance decreased in our test scene in the maze of the

(a) A maze environment built by cardboard.

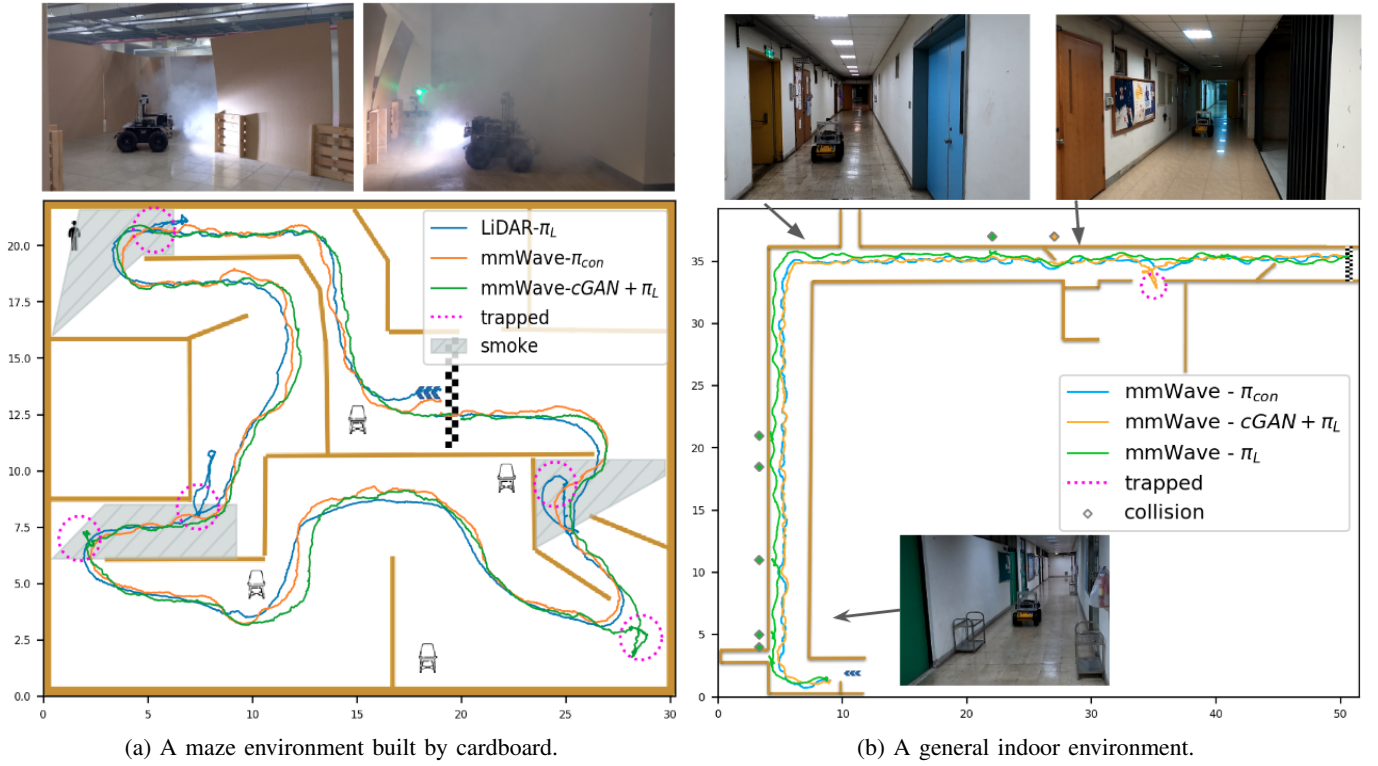(b) A general indoor environment.

Fig. 5: Navigation experiments and visualized trajectories of various control policies. (a) A maze environment built by cardboard. The robot was trapped due to the smoke (LiDAR + π_L) or phantom walls (mmWave-$cGAN + \pi_L$). (b) A general indoor environment, where mmWave-$cGAN + \pi_L$ has some cases of collisions and instances of being trapped caused by reconstruction errors.

underground basement. This might be because the mmWave signals could penetrate the cardboard, resulting in inaccurate reconstructions.

C. Navigation in a Smoke-Filled Maze

1) Experiment Environment: The testing environment was a cardboard maze in a basement (23 m × 28 m) with widths ranging from 2.25 m to 6 m, and the walls were 1.78 m high. One lap was approximately 100 m long and contained 20 turns, which were mostly $90°$ turns, except for some sharp turns. We used four chairs and one mannequin as obstacles. The robot navigated in the maze, similar to that in the tunnel without intersections. The LeGO-LOAM method [29] was applied with the LiDAR data to obtain robot trajectories; the trajectories of some methods are plotted in Fig. 5 for qualitative evaluation.

2) Metrics: Robot navigation failures were quantified according to the number of **collisions** and the number of instances of being **trapped**, which were a result of noisiness or sparseness in the mmWave radar data. Specifically, the robot collided into an obstacle when the obstacle was undetected and became trapped when the range estimates were inaccurate, which could be due, for example, to phantom walls. We also recorded odometry results from wheel and LiDAR, but the distance traveled did not significantly differ between the methods evaluated.

3) Ground truth: We used navigation with LiDAR data, using both the RL control policy π_L and classic A* planning

π_{cl} in a smoke-free environment, as the ground truth. The goal points of classic planning were set to follow the walls, and a pure pursuit algorithm and PID controller were used. Both methods resulted in the robot safely traversing the maze without colliding or being trapped.

4) Smoke trials: A smoke machine emitted dense smoke at three locations (Fig. 5) for 15s before the UGV arrived. To adhere to safety regulations, we did not fill the entire basement with smoke. We conducted a few smoke trials for π_L, $cGAN + \pi_L$, and π_{con} to observe how the smoke affected the control policies. The smoke prevented the LiDAR signals from penetrating the passage, which caused the UGV to be trapped. We found that navigation methods using mmWave radar were unaffected by smoke.

TABLE II: Navigation performance in maze with respect to number of collisions and instances of being trapped.

Inputs	Methods	Avg. Coll.	Avg. Trapped	Env. Note
LiDAR	π_L	0	0	No Smoke, GT
	π_{cl}	0	0	No Smoke, GT
	π_L	0	3	Trapped in Smoke
mmWave	π_{cl}	-	-	Unable to navigate
	$cGAN + \pi_{cl}$	1.60	1.60	Collision and trapped
	π_L	1.20	0.60	Baseline - w/o CM
	$cGAN + \pi_L$	0	1.17	Baseline - generative
	π_{con}	**0**	**0**	**Ours - contrastive**

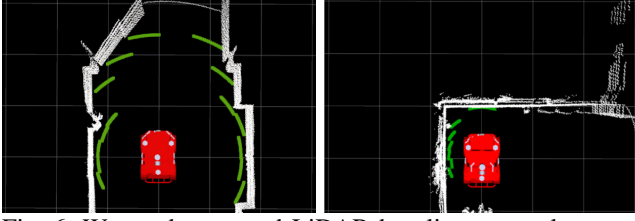*5) Generative reconstruction baselines:* For each method, the test was run for five laps, and we recorded the number of

Fig. 6: We used truncated LiDAR baseline to emulate sonar ring inputs (green line). The emulated sonar ring has 15° resolution and 3 m of maximum range. In the experiments we found that low angular resolution made it difficult to find an opening to navigate through narrow passage (left); short range is also a disadvantage of making turns (right).

collisions and instances of being trapped. We evaluated the classic π_{cl} and RL π_L approaches, comparing them with π_{cl} and π_L enhanced by a reconstruction method (i.e., cGAN) using direct raw mmWave inputs. We found that the local map constructed using raw mmWave inputs lacked occupancy, resulting in a failure to navigate from the use of π_{cl}. Conversely, through the use of the cGAN to reconstruct the range data, the robot could navigate using the classic planning method, albeit with considerable collisions and instances of being trapped due to reconstruction errors. The control policy π_L was more robust to noisy raw mmWave inputs without any tuning of the neural network, and π_L managed to navigate through the maze even without reconstruction. When $cGAN + \pi_L$ was used, the robot collided less frequently but got trapped more frequently.

6) Cross-modal contrastive learning of representations: π_{con} performed the best among all of the evaluated methods, where the deep RL control policy was used after training with contrastive loss. π_{con}, as an end-to-end network, allowed the robot to safely traverse the entire maze in all five trials without colliding or being trapped.

D. Navigation in a General Indoor Corridor

To further evaluate the navigation performance in a general indoor environment, we conducted the second experiment in a corridor located at another building which was not used for training data collection. The environment has two major corridors with 30 m and 50 m in length, respectively, two 90° turns and two narrow gates. For safety considerations, we did not emit smoke in this environment. The details of this environment are shown in Fig. 5.

1) Truncated LiDAR (Emulated Sonar Ring) baseline : In addition to previously mentioned baselines for navigation, we introduced a truncated LiDAR baseline to emulate a sonar ring to be compared with. In the review [21] a sonar ring was in the resolution of 15° resolution and a maximum range of 3 m. Following such setting, we carried out an experiment with decimated and truncated LiDAR measurements of 24 beams up to 3 m. For the size of the Husky robot (0.67 m in width) low angular resolution makes it difficult for the robot to find an opening in a relatively narrow passage (1.45 m in width in our experiment). Short range measurement is also a disadvantage while making turns, shown in Fig. 6. The quantitative results in Table III showed that the robot was trapped at both of the narrow passages.

2) Generative reconstruction baselines: With the help of the reconstruction model, both π_{cl} and π_L are able to navigate through the corridor with some collisions and instances of being trapped, but they happened more frequently than in the maze environment. We observed that the reconstruction model sometime generated corridor geometry with errors in angle compared to the ground truth when the vehicle was $10° \sim 45°$ relative to the wall. The errors in angle caused minor reconstruction loss but misled the vehicle to finally collide into the wall. This implies our dataset should be more diverse with more data of the robot not parallel to the walls to improve the generative reconstruction performance.

3) Cross-modal contrastive learning: π_{con} also performed the best in this environment. Only one instance of being trapped happened among all five trials when the robot navigated to the narrow gate in the environment.

TABLE III: Navigation performance in the indoor corridor.

Inputs	Methods	Avg. Coll.	Avg. Trapped	Env. Note
LiDAR	π_L	0	0	No Smoke, GT
	π_{cl}	0	0	No Smoke, GT
Truncated	π_{cl}	0	2	Emulate Sonar
mmWave	π_{cl}	-	-	Unable to navigate
	$cGAN + \pi_{cl}$	3.2	1	Collision and trapped
	π_L	7	0	Baseline - w/o CM
	$cGAN + \pi_L$	0.6	0.6	Baseline - generative
	π_{con}	**0**	**0.2**	**Ours - contrastive**

E. General Discussions

We found that the results of Table II and III are consistent that the proposed contrastive learning method obtained better performance than baseline methods. The results may indicate that learning the representation may be more robust across different environments. We conjectured that for $cGAN + \pi_L$ to achieve the same performance as π_{con}, it is better for the reconstructed mmWave data to be similar to detailed LiDAR data. Without such similarity, the feature extraction network in π_L may generate misleading representations for the decision-making network. The results of generative baseline in different environments were found to decrease the number of trapped cases, and a slightly increased number of collisions in dense wall environments. The results may indicate that the generative reconstruction in different environments may be sensitive to the training dataset, which is consistent with the results of Table I. Future work could collect more training data in diverse environments to further verify the influences of training data on learning reconstruction.

We had tested two different mmWave sensor configurations on the same robot (Husky), shown in Fig. 2. Configuration 1 was used to collect training data and conduct the experiments, and configuration 2 was designed for potential use of smaller robots. We found that the different mmWave sensor configurations did not affect navigation performance on the same Husky robot. However, We suggested that additional transfer learning efforts considering robot size, speed, and reward settings may be needed to apply the method to a different robot.

An alternative to contrastive learning is hallucination, which has been used in other cross-modal tasks [30], [31]. [30] training a modality hallucination architecture to learn a multimodal convolutional network for object detection. The hallucination was trained to take an RGB input image and mimic the depth mid-level activations. Cross-modal studies on high-bandwidth inputs like images and low-bandwidth inputs (LiDAR/mmWave) could be further studied.

VI. CONCLUSIONS

Learning how to navigate is difficult under adverse environmental conditions, where sensing data are degraded. As a remedy, we leveraged mmWave radars and proposed a method that uses cross-modal contrastive learning of representations with both simulated and real-world data. Our method allows for automated navigation in obscurant-filled environments using data from only single-chip mmWave radars. In a comprehensive evaluation, we incorporated state-of-the-art generative reconstruction with a control policy for navigation tasks. Our method performed best in the smoke-filled environment against its counterparts, with results that were comparable to those from groundtruth LiDAR approaches in a smoke-free environment. Future studies can apply our method to learning across modes and platforms.

REFERENCES

[1] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A Platform for Embodied AI Research," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[2] J. Zhang, J. T. Springenberg, J. Boedecker, and W. Burgard, "Deep reinforcement learning with successor features for navigation across similar environments," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 2371–2378.

[3] M. Pfeiffer, M. Schaeuble, J. Nieto, R. Siegwart, and C. Cadena, "From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots," in *2017 ieee international conference on robotics and automation (icra)*. IEEE, 2017, pp. 1527–1533.

[4] M. Pfeiffer, S. Shukla, M. Turchetta, C. Cadena, A. Krause, R. Siegwart, and J. Nieto, "Reinforced imitation: Sample efficient deep reinforcement learning for mapless navigation by leveraging prior demonstrations," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 4423–4430, 2018.

[5] S. H. Cen and P. Newman, "Radar-only ego-motion estimation in difficult settings via graph matching," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 298–304.

[6] ——, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[7] Y. Almalioglu, M. Turan, C. X. Lu, N. Trigoni, and A. Markham, "Milli-rio: Ego-motion estimation with low-cost millimetre-wave radar," *IEEE Sensors Journal*, 2020.

[8] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: Robust indoor mapping with low-cost mmwave radar," in *ACM International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2020.

[9] I. A. Hemadeh, K. Satyanarayana, M. El-Hajjar, and L. Hanzo, "Millimeter-wave communications: Physical channel models, design considerations, antenna constructions, and link-budget," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 870–913, 2017.

[10] R. Bonatti, R. Madaan, V. Vineet, S. Scherer, and A. Kapoor, "Learning controls using cross-modal representations: Bridging simulation and reality for drone racing," *arXiv preprint arXiv:1909.06993*, 2019.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *International conference on machine learning*. PMLR, 2020, pp. 1597–1607.

[12] M. Laskin, A. Srinivas, and P. Abbeel, "Curl: Contrastive unsupervised representations for reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 5639–5650.

[13] G. Kahn, P. Abbeel, and S. Levine, "Badgr: An autonomous self-supervised learning-based navigation system," 2020.

[14] F. Niroui, K. Zhang, Z. Kashino, and G. Nejat, "Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 610–617, 2019.

[15] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Transactions on Vehicular Technology*, pp. 1–1, 2020.

[16] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 31–36.

[17] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning." in *ICLR*, Y. Bengio and Y. LeCun, Eds., 2016.

[18] S. Gu, E. Holly, T. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3389–3396.

[19] S. Walter, "The sonar ring: Obstacle detection for a mobile robot," in *Proceedings. 1987 IEEE International Conference on Robotics and Automation*, vol. 4. IEEE, 1987, pp. 1574–1579.

[20] S. Fazli and L. Kleeman, "Wall following and obstacle avoidance results from a multi-dsp sonar ring on a mobile robot," in *IEEE International Conference Mechatronics and Automation, 2005*, vol. 1. IEEE, 2005, pp. 432–437.

[21] L. Kleeman and R. Kuc, *Sonar Sensing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 491–519. [Online]. Available: https://doi.org/10.1007/978-3-540-30301-5_22

[22] O. Robotics. (2016) Darpa subterranean challenge virtual competition. [Online]. Available: https://bitbucket.org/osrf/subt/wiki/Home

[23] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *CoRR*, vol. abs/1512.04455, 2015. [Online]. Available: http://arxiv.org/abs/1512.04455

[24] T. Fan, P. Long, W. Liu, and J. Pan, "Fully distributed multi-robot collision avoidance via deep reinforcement learning for safe and efficient navigation in complex scenarios," *CoRR*, vol. abs/1808.03841, 2018. [Online]. Available: http://arxiv.org/abs/1808.03841

[25] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[26] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European conference on computer vision*. Springer, 2016, pp. 702–716.

[27] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2014.

[28] A. Spurr, J. Song, S. Park, and O. Hilliges, "Cross-modal deep variational hand pose estimation," in *CVPR*, 2018.

[29] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[30] J. Hoffman, S. Gupta, and T. Darrell, "Learning with side information through modality hallucination," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 826–834.

[31] M. R. U. Saputra, P. P. B. de Gusmao, C. X. Lu, Y. Almalioglu, S. Rosa, C. Chen, J. Wahlström, W. Wang, A. Markham, and N. Trigoni, "Deeptio: A deep thermal-inertial odometry with visual hallucination," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1672–1679, 2020.