Help Me Through: Imitation Learning Based Active View Planning to Avoid SLAM Tracking Failures

Kanwal Naveed, Wajahat Hussain, Irfan Hussain, Donghwan Lee, *Member, IEEE*, and Muhammad Latif Anjum

***Abstract*—Large-scale evaluation of state-of-the-art visual simultaneous localization and mapping (SLAM) has shown that its tracking performance degrades considerably if the camera view is not adjusted to avoid the low-texture areas. Deep reinforcement learning (RL)-based approaches have been proposed to improve the robustness of visual tracking in such unsupervised settings. Our extensive analysis reveals the fundamental limitations of RL-based active view planning, especially in transition scenarios (entering/exiting the room, texture-less walls, and lobbies). In challenging transition scenarios, the agent generally remains unable to cross the transition during training, limiting its ability to learn the maneuver. We propose human-supervised RL training (imitation learning) and achieve significantly improved performance after ∼50 h of supervised training. To reduce longer human supervision requirements, we also explore fine-tuning our network with an online learning policy. Here, we use limited human-supervised training (∼20 h), and fine-tune the network with unsupervised training (∼45 h), obtaining encouraging results. We also release our multimodel, human supervised training dataset. The dataset contains challenging and diverse transition scenarios and can aid the development of imitation learning policies for consistent visual tracking. We also release our implementation.**

***Index Terms*—Active view planning, imitation learning, reinforcement learning (RL), simultaneous localization and mapping (SLAM), tracking failure, visual navigation.**

I. Introduction

IF the camera is not being controlled by a human, the state-of-the-art monocular visual simultaneous localization and mapping (SLAM) suffers track loss even at short ranges [2]. Perhaps the reason behind this behavior is the fact that well-known visual SLAM datasets have limitations [3]. Naveed et al. (DI-SLAM [2]) evaluated visual SLAM in situations where the camera movement was not under the control of the human operator. ORB-SLAM [4] was unable to reach the goal, which was a few meters away, in any of the 50 episodes. They proposed a deep reinforcement learning (RL) approach to avoid moves that might result in tracking failure. In this work, we evaluate this deep RL-based approach across transitions (doors, corners, texture-less lobbies) and report its fundamental limitations.

Our experiments suggest that while this deep RL approach [2] works fine if the start and the goal are inside a single room or a large vicinity of textured walls, it fails during transitions. We trained this approach, especially for such transitions. Even after training for ∼12 h (710 episodes) on a single route, it was unable to complete the route it was trained for. The inspection of the training episodes revealed it could not cross the door even once during training either (710 episodes) [see Fig. 1(a)]. The lack of success (sparse rewards) during training resulted in poor learning experiences. Here, the sparse reward means that the RL agent is unable to successfully complete the transition without tracking loss. Hence, there is very limited feedback to learn a transition policy consisting of actions, which lead to success. Sparse rewards are usually expected in tasks with multiple actions and long horizons [5]. Our analysis reveals that in the case of visual tracking, the rewards are sparse, even in the case of short-horizon tasks such as crossing the door. In this work, we propose imitation learning-based active view planning to overcome such limitations [see Fig. 1(b) and (c)].

Visual SLAM's classic pipelines (tracking, mapping, loop closing) rely heavily on frame-to-frame feature matching. Loop closing, which helps recover the lost track, might not work across transitions, as one has to go beyond a previously visited area. In this case, learning-based navigation agents, which do not rely on explicit feature matching or mapping, offer attractive alternatives [6], [7]. Such learning-based agents have been shown to reach a goal even if it is several rooms apart [8], [9] or located in large cities [10]. Instead of being short-sighted by frame-to-frame matching, these agents manage to learn the big picture (map) using the structured nature of the man-made world, e.g., McDonald's is most likely to be found downtown compared to fields [8], coffee can be found in the kitchen, and one needs to leave the bedroom for that [9].

However, learning-based approaches have underperformed compared to classic approaches for tracking, mapping, and localization tasks. Sattler et al. [11] showed that deep

Received 24 November 2024; revised 12 May 2025; accepted 7 June 2025. Date of publication 24 June 2025; date of current version 14 July 2025. This work was supported by the Khalifa University of Science and Technology under Award CIRA-2021-085, Award 8434000534, and Award RC1-2018-KUCARS. This article was recommended for publication by Associate Editor F. Fraundorfer and Editor S. Behnke upon evaluation of the reviewers' comments. *(Corresponding author: Muhammad Latif Anjum.)*

Kanwal Naveed, Wajahat Hussain, and Muhammad Latif Anjum are with the Robotics and Machine Intelligence Lab, School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad 44000, Pakistan (e-mail: knaveed.phdee17seecs@seecs.edu.pk; wajahat.hussain@seecs.edu.pk; latif.anjum@seecs.edu.pk).

Irfan Hussain is with the Khalifa University Center for Autonomous Robotic Systems, Khalifa University, Abu Dhabi 127788, UAE (e-mail: irfan.hussain@ku.ac.ae).

Donghwan Lee is with the Reinforcement Learning Research Lab, School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea (e-mail: donghwan@kaist.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/TRO.2025.3582817, provided by the authors.

Digital Object Identifier 10.1109/TRO.2025.3582817

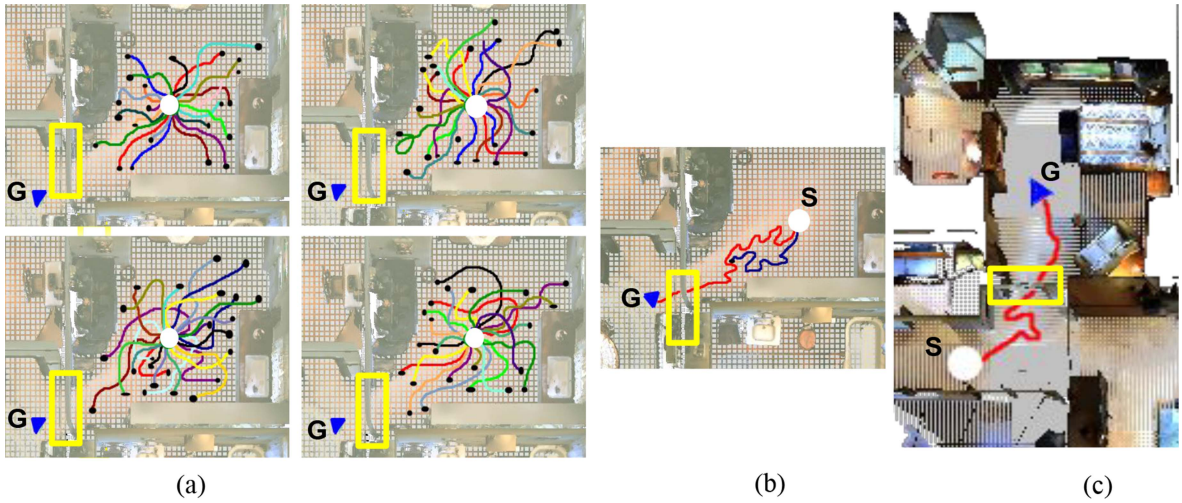

Fig. 1. *Mission Impossible to Cross the Door!* (a) We evaluate ORB-SLAM3's [1] tracking performance on a short route, with a start point (white circle) inside the room and goal (blue triangle) just across the door (yellow box). We retrained the deep RL approach (DI-SLAM [2]), designed to avoid visual tracking failures, on this specific route for ∼12 h (710 episodes). Each subfigure shows 25 paths (representative of a total of ∼175 paths) adopted by this DI-SLAM approach after training intervals of 3 h. These trajectories represent the paths taken by DI-SLAM during the training phase, and the black dot at the end of each trajectory indicates the point of tracking failure. This deep approach fused with ORB-SLAM3 could not cross the door even a single time. (b) DI-SLAM is trained under human supervision. The human operator is able to cross the door in the second attempt (red line). (c) Our deep imitation learning-based approach manages to complete a longer and novel route that includes a door after manual training of ∼20 h only (with fine-tuning of ∼45 h) on completely nonoverlapping scenes. Figure best viewed with digital zoom (**S**: start, **G**: goal). (a) Retraining of DI-SLAM on a specific route. (b) Training of DI-SLAM under human supervision. (c) Imitation-SLAM (ours) in action.

relocalization approaches have fundamental limitations and, therefore, are not ready to replace classic relocalization pipelines especially in large-scale scenarios [11]. Finally, the performance of these learning-based approaches degrades if the domain/scenario changes, e.g., navigation policy for urban interior might be different from commercial shops or factories. This performance depreciation is well understood in the case of medical images [12]. Classic navigation pipelines do not rely on learning and, therefore, are not affected by the domain change. In our opinion, efforts should be made to improve the robustness of classic pipelines along with developing newer learning-based approaches.

Self-driving vehicles (SDVs) have been the focus of research for the automatic navigation of mobile agents in dynamic scenes. The interest of technology giants, such as NVIDIA [13], WAYMO [14], UBER [15], and TESLA amongst others, have accelerated the progress in evaluating RL for visual navigation. After the seminal success of deep mind's Atari playing agent [16], which learned how to play games just by observing the video feed (raw pixels) and game scores, similar success has not been matched for SDV for deep RL approaches despite considerable progress.

The way forward to deploy deep RL approaches to visual tracking can be summarized from the literature. First, it is more tractable to learn from expert human behavior (imitation learning [17]) than directly practising inside a simulator [13], [14], [15], [18], [19], [20]. Second, it is a better strategy to start from easier challenges (learning to maintain the lane on a highway as compared to urban scene traffic) [13]. Lastly, understanding the scene using object detection (cars, signals) [21] or intermediate representations (simplifying the scene using a top view) helps learn better policies [14], [15], [19].

We take inspiration from these findings and try to overcome the fundamental limitations of the deep RL approach to tracking failures in visual SLAM across transitions. Instead of relying on learning from practising within a simulator [2], [16], we propose to imitate the behavior of human experts. We incorporate human expertise to actively plan the viewpoint of the robot by choosing the most appropriate action in any scenario, which can avoid dangerous situations that lead to SLAM tracking failures. We focus our efforts on entering/exiting doors, featureless lobbies, and texture-less walls. We also include challenging scenarios where the robot is unable to locate any features while exiting or entering a room and when both walls of the lobby are featureless (see Fig. 2). These scenarios are extremely challenging for a deep RL-based SLAM approach [2].

There are a few challenges before we apply imitation learning to avoid tracking failures in visual SLAM. In the case of SDV, humans drive the car naturally, and deep networks are expected to clone this behavior. There are racing-based entertainment games (e.g., need for speed) that can be used to generate data for SDV training. Leveraging this ease, months and year-long manual driving data has been generated which has propelled SDV research (see Table I). In the case of visual SLAM, creating successful attempts of crossing the transitions is nontrivial since the human has to cross the transition such that the tracking does not fail. This means that human has to decide what action to take next so that the robot's view can maximize the tracked features (active view planning). As the experiments suggest, even human-controlled SLAM fails to track. Therefore, creating successful journeys via active view planning across transitions is considerably more *time-consuming* and laborious as compared to driving. To this end, we provide a novel manual training dataset of ∼50 h containing challenging tracking scenarios.

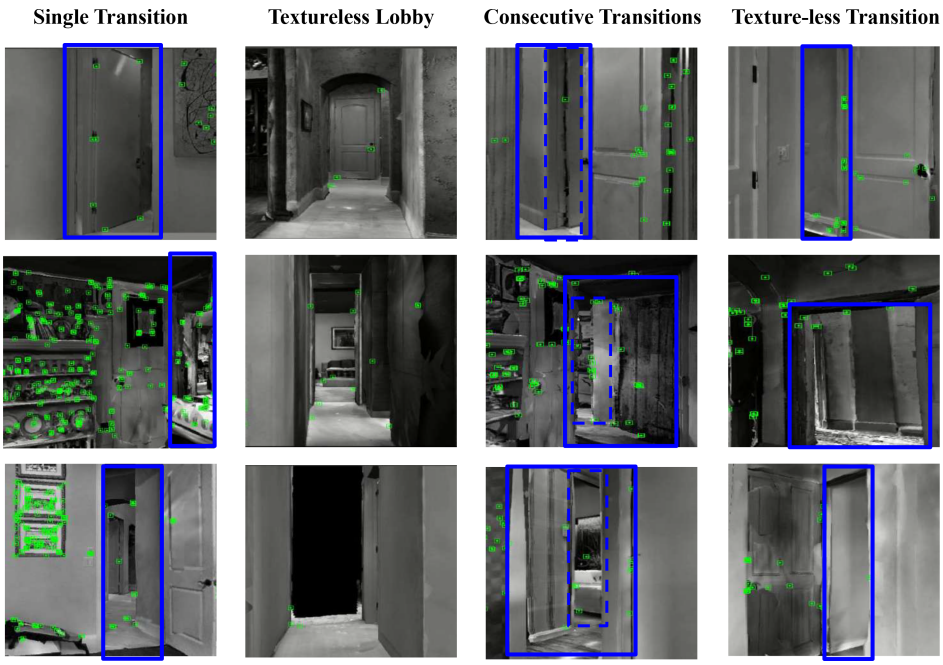

Fig. 2. *Imitation-SLAM dataset (large-scale, multichallenge dataset):* We release a comprehensive 50-h dataset, which includes a variety of difficult tracking situations for ORB-SLAM3, including single and consecutive transitions, texture-less lobbies, and room entries or exits where there are very limited features ahead of the door/transition (highlighted as blue boxes). An expert human plans the trajectory in these scenarios to complete the episode without tracking failure. In addition to the human supervision, this dataset also comes with 6 DoF ground truth pose and depth images.

TABLE I
IMITATION LEARNING DATASETS: WHAT CAN WE LEARN FROM IMITATION LEARNING DATASETS DEVELOPED FOR SDV? LARGE MANUAL DRIVING DATASETS ARE REQUIRED

Approach	Year	Manual Driving Data
LeCun [22]	2005	127000 frames = $\sim$ 2.35 hours
NVIDIA [13]	2016	3 days
BDDV [18]	2017	417 days
Waymo [14]	2018	60 days
Sauer et al. [19]	2018	240000 frames = $\sim$ 4.44 hours
Uber [15]	2020	10 days

Finally, to assess how much manual supervision is required, we fine-tune our imitation learning framework using automatic but unsupervised exploration inside the simulator. Interestingly, now equipped with expert human policy, the imitation learning framework outperforms itself with this additional unsupervised experience. More concretely, an imitation learning framework with only 20 h of manual data and additional unsupervised fine-tuning within a simulator outperforms an imitation learning framework with 50 h of manual expert data. This fusion of offline human supervision and online simulator exploration makes active view planning a manageable task, even for novel tracking scenarios.

Contributions

1) We show that RL-driven active view planning encounters challenges due to sparse rewards in transition scenarios. When dealing with sparse rewards, further RL training can result in inefficient improvements.
2) We propose an efficient imitation learning-based active view planning approach that achieves state-of-the-art results in visual tracking in challenging situations (e.g., transitions).
3) We release our dataset of 50 h of manual training, which can be used as expert policy (π_b) for offline training.

II. RELATED WORK

There are two main approaches to deal with failures in visual tracking. The first approach consists of classic and well-established methods to counter tracking failures (postfailure). The second and more recent approach uses learning methods to pre-empt tracking failure and try to avoid it.

Loop closing is perhaps the most widely used approach to recover from track losses [1], [4], [23]. Planning the path in a manner that consists of frequent revisits of a particular location adds robustness to visual tracking [24]. Initially, loop closing was used for regaining the track [4], [25]. More recently, a multiple map approach has been used, where a new map is started after track loss, and upon revisiting a particular location, maps are fused [1]. Next best view (NBV) is another classic approach that plans the next motion after accessing the current state, e.g., 3-D map. In certain NBV methods, an action is chosen if it ensures that the current map remains visible from the prospective future position [26]. In the case of exploring a new region, tracking the current map might not be feasible; instead, the potential to acquire substantial new features is perhaps a better strategy to maintain tracking.

Planning the NBV is nontrivial in transition scenarios since the map of the current scene (room) will not be visible after the transition. Also, the content of the scene beyond the transition may not be visible; hence, it is difficult to forecast potential

future features required to maintain tracking. Deng et al. [27] applied the NBV approach around corridor corners, which allows relatively smoother transitions as compared to doors. Similarly, the option of using loop closures to regain tracking is not possible when transitions into an unexplored area are involved. Salas et al. [28] indicated the fundamental limitations of visual tracking across transitions. Both feature-based [4], [25] and direct [29], [30] SLAM methods are greatly affected by occlusion indoors. Salas et al. [28] suggested a high-level scene layout understanding [31], [32] to account for such failures. However, even in their case, tracking was maintained manually while transiting the room.

There are few learning-based approaches to predict feature tracking failures in outdoor scenarios. Hartmann et al. [33] used a learning-based approach to determine which features are generally reliable and matchable in the future. Similarly, Rabiee et al. [34] learned to select reliable features while taking specular reflection, lens flare, and shadows into account. However, these approaches [33], [34] do not take into account heavy occlusion, which occurs across transitions indoors, and even reliable features will not match under occlusion.

Saxena et al. [35] used a classifier-based approach that learned moves to avoid tracking failures for drone navigation through the woods. Closeups, tree shadows, and texture-less regions (plain land) were the main cause of tracking failures. In this case, a single move helped avoid tracking failures (one step to the right might avoid being blinded by a tree trunk). Transitions indoors might require a policy based on a sequence of actions that guarantees delayed rewards. Habibi et al. [36] proposed a virtual navigation support tool that controls the virtual camera automatically through feature-rich areas inside 3-D models. However, feature tracking is not their objective.

Safe-SLAM [37] proposed a Q-learning-based approach to learn long-term policies to avoid tracking failures across sharp turns. DI-SLAM [2] fused this Q-learning approach with the deep learning methods and offered a large-scale evaluation of visual SLAM in indoor scenarios. In this work, we show that this deep RL-based approach [2] considerably improves tracking stability if the mobile agent navigates inside a single room. However, it has fundamental limitations across transitions. We propose a novel active view planning-based imitation learning approach (Imitation-SLAM) to overcome the limitations of this deep RL approach.

Recently, Dai et al. [38] proposed a generative adversarial imitation learning (GAIL) [39] approach to improve visual tracking for RGBD SLAM. GAIL-SLAM [38] overcomes the limitations of reward function [40] present in RL approaches. Differently to [38], we aim to handle the transitions (crossing the door, turning a corner), whereas their focus is only on texture-less surfaces while utilizing the GAIL technique.

Similar to DI-SLAM [2] and GAIL-SLAM [38], we use a photo-realistic simulator [41] for evaluating our approach. These simulators allow the evaluation of visual SLAM approaches without human involvement at the testing stage. Well-known visual SLAM datasets are mainly gathered for outdoor scenes (KITTI [42]) or have limited transition scenarios indoors (TUM-RGBD [43], TartanAir [3]). Furthermore, we demonstrate the efficacy of our approach in real-world experiments.

The SLAM community is long aware of this challenge of crossing transitions, especially doors and texture-less lobbies [28], [44]. However, failures across transitions have largely eluded attention. In our opinion, the well-known benchmarks are gathered by a human operator controlling the camera motion (KITTI [42], TUM-RGBD [43], TartanAir [3]). Even before these benchmarks, visual SLAM solutions were evaluated for a sequence gathered by a handheld camera [45]. Therefore, these sequences were captured using human intelligence and, perhaps, resulted in a limited number of tracking failures. In this work, we propose to leverage this human expertise in training the visual SLAM agent to cross these transition barriers. The human expert in any scenario will be responsible for actively deciding which action (forward/backwards, clockwise/anticlockwise rotation, etc.) will result in the tracking failure and thus will be able to take the best action to adjust the camera view of the robot for the best possible feature tracking.

There are a few interesting human–robot collaborative mapping approaches [46], [47]. However, these collaborations are limited to a specific scene, whereas our learning-based approach generalizes over different scenes.

III. Imitation-SLAM: Our Training Dataset

As indicated in Fig. 1, crossing challenging transition scenarios without tracking failure is an open challenge for the state-of-the-art monocular visual SLAM leveraging traditional perspective cameras. The same is also true for monocular visual SLAM trained with a deep Q-network (DQN) [2]. The way forward, as proposed in this work, is to use human expertise to train the agent to avoid tracking failures. Unfortunately, no dataset is available for this purpose, containing human users expertly navigating in an environment to avoid tracking failure (see Table II).

With this work, we also release our training dataset, which is a 50-hour-long dataset containing 100 challenging sequences (see Fig. 2). We call our dataset the Imitation-SLAM dataset. Our dataset includes challenging tracking scenarios where the expert SLAM user manages to complete the episode without tracking failure. Different scenarios include the following.

1) *Transitions*: Agent entering/exiting the scene with little content visible from the target room (door crossing).
2) *Consecutive transitions*: Agent moving on the path having consecutive transitions (doors) with little interval to gather features.
3) *Texture-less lobby*: Environments/long lobbies with lack of features/texture.
4) *Texture-less transitions*: Agent entering the scene with texture-less entrance.

This dataset also includes 6 DoF ground truth pose and depth images. In addition, our dataset contains long trajectories (a path length of around 280 m) (see Fig. 3), spanning multiple rooms. This path length is longer than the maximum path length covered in well-known indoor datasets, e.g., TUM-RGBD [43] and EuRoC MAV [54], having a maximum trajectory length of 150 m (see Table III), while staying in the same room/region during the dataset collection.

TABLE II
IMITATION-SLAM (OURS) DATASET—A QUANTITATIVE ANALYSIS: OUR DATASET CONTAINS SUCCESSFULLY COMPLETED EPISODES BY SLAM EXPERTS IN CHALLENGING TRACKING SCENARIOS

Dataset	Environment	Frames	Resolution (Mpx)	Sequences	Ground truth (pose)	SLAM applicability	Adequate transition scenarios	Suitable for imitation training
KITTI [42]	outdoor	41k	0.5	22	✓	✓	✗	✗
ActiveVision [48]	indoor	30k	0.3	15	✗	✗	✗	✗
Fordcampus [49]	outdoor	7k	1.0	2	✗	✓	✗	✗
TUM-RGBD [43]	indoor	65k	0.3	27	✓	✓	✗	✗
Malaga [50]	outdoor	38k	0.8	6	✓	✓	✗	✗
Newcollege [51]	outdoor	51k	0.2	1	✗	✓	✗	✗
NYU Depth [52]	indoor	0.14k	0.3	26	✓	✓	✗	✗
Multiview RGBD [53]	indoor	0.67k	2	9	✓	✓	✗	✗
TartanAir [3]	mixed	233k	–	1037	✓	✓	✗	✗
EuRoC MAV [54]	indoor	–	0.3	11	✓	✓	✗	✗
ICL-NUM [55]	indoor	0.92k	0.3	8	✓	✓	✗	✗
SceneNet [56]	indoor	5M	0.07	17	✗	✓	✗	✗
RobotCar [57]	outdoor	0.3M	1.2	100	✓	✓	✗	✗
Northcampus [58]	outdoor	–	1.92	27	✓	✓	✗	✗
Imitation-SLAM **(Ours)**	**indoor**	**180k**	**2**	**100**	✓	✓	✓	✓

Other datasets exhibit limited multiple scene transition scenarios.

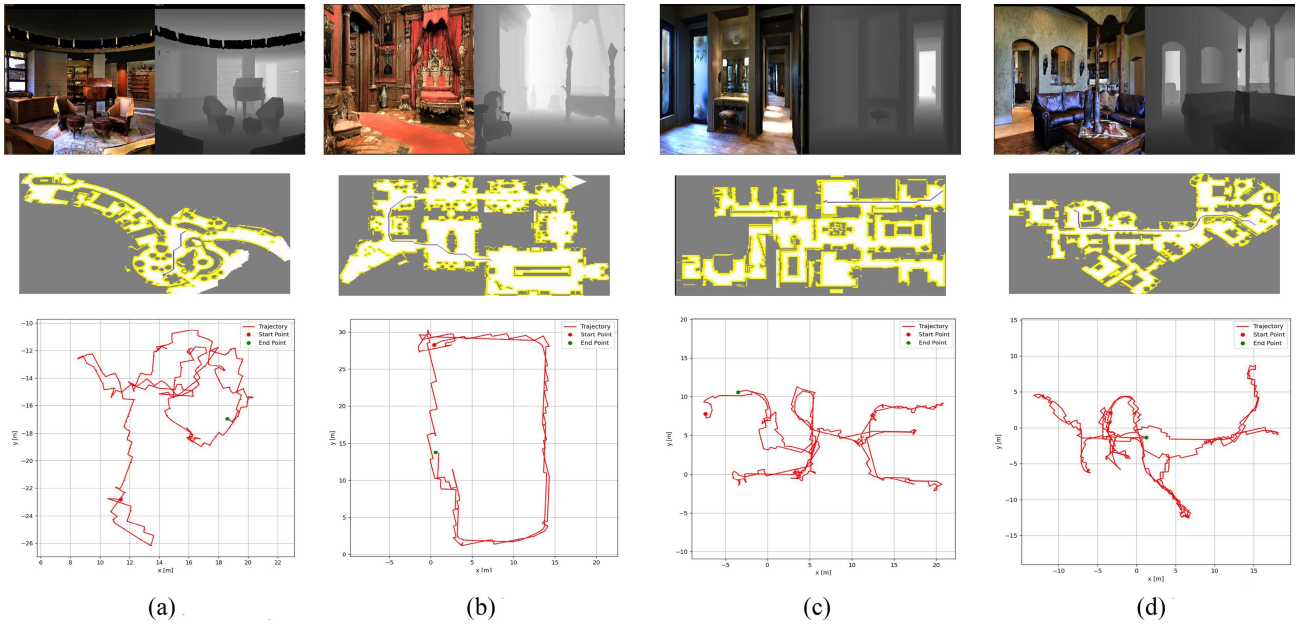

Fig. 3. *Sample sequences from our dataset:* We present four examples of sequences contained in our dataset. The top row represents an example RGB and depth image, the middle row represents the area's complete map, and the bottom row represents the ground truth trajectory. (a) Path length: 110.00 m. (b) Path length: 178.60 m. (c) Path length: 231.98 m. (d) Path length: 276.86 m.

TABLE III
COMPARATIVE ANALYSIS OF PATH LENGTHS: WE PRESENT A COMPARISON OF OUR IMITATION-SLAM DATASET WITH TWO OF THE FAMOUS EXISTING DATASETS

Dataset	Frames	Maximum path length	Multiple rooms	Textureless lobbies	Multiple transitions
TUM-RGBD [43]	65k	150m	✗	✗	✗
EuRoC MAV [54]	-	150m	✗	✗	✗
Imitation-SLAM (Ours)	180k	280m	✓	✓	✓

A. Demo by Human Operator

How good are humans at avoiding tracking failure if they are asked to control the camera motion? Fig. 4 shows an example of how the human controls the camera to avoid tracking failure while entering the room. Initially, only a limited part of the room is visible through the open door [blue box in Fig. 4(a)], and no features from the target room are part of the tracker. Therefore, the human operator moves across the door, hoping to capture the features from that room [see Fig. 4(b) and (c)]. However, limited features have been captured [negligible features inside the red box in Fig. 4(c)]. Hence, moving toward the room is dangerous. The operator steps back [see Fig. 4(d)] and again moves across the door [see Fig. 4(e)], this time capturing features nicely spread across the target room [substantial features inside white box in Fig. 4(e)]. Now, it is safe to enter the room without tracking failure [see Fig. 4(f)]. Our experiments show that at least ∼9 h (350 episodes) of human demonstration data is required so that the monocular visual SLAM agent has a nonzero success rate of crossing the transitions without tracking failures.

Avoiding tracking failure may seem trivial at first glance, but even humans fail to make this transition over several attempts

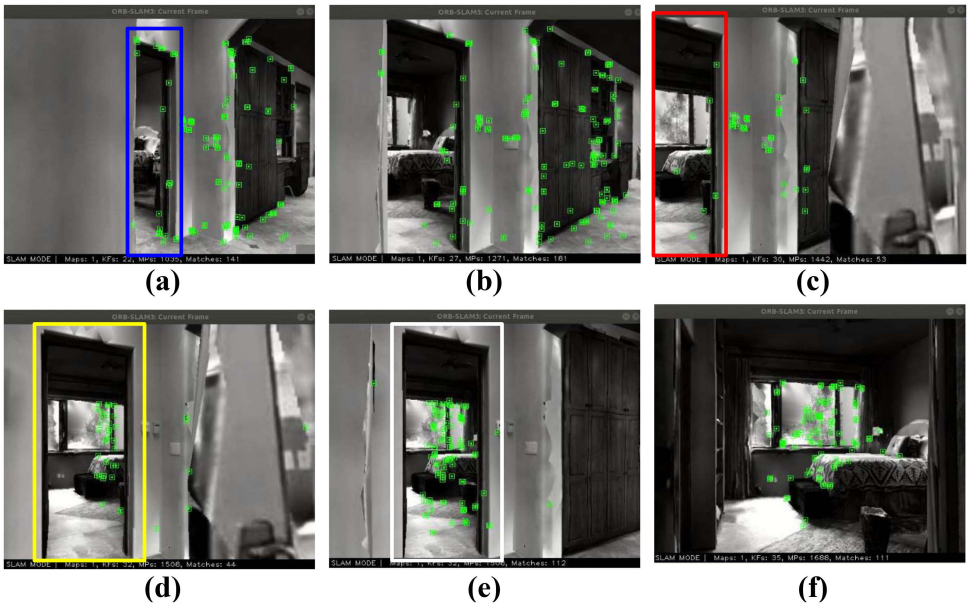

Fig. 4. *Successful demo by human operator:* Human operator successfully crosses the door without tracking failure. The human aims to enter the room on the left [blue box in (a)], but the contents of this room are not visible, which poses a problem. To capture enough features (green dots) from the target room, the human operator decides to first move across the door (b)–(c) but is only able to capture a few features, as shown by the limited features in the red box in (c). The operator then decides to step back (d) and try again, moving across the door (e). This second attempt successfully captures enough features from the target room. With sufficient features captured, it is now safe to enter the room (f). We encourage the reader to view the supplementary video (timestamp: 01:23–01:48) for better understanding.

TABLE IV
JUDGING HUMAN EXPERTISE: EVEN EXPERIENCED SLAM USERS SUFFER TRACKING FAILURE DURING TRANSITIONS

	Human 1	Human 2	Human 3	Human 4	Human 5
SLAM Experience	✓	✓	✗	✗	✓
Attempts for Successful Transition	3	2	4	3	1

despite having prior SLAM experience (see Table IV). Fig. 5 shows an example where the aim is to enter the room on the right. The target room's contents are visible through the door [blue box in Fig. 5(a)], and they are texture-rich. However, limited features have been captured on the right side of the screen [see Fig. 5(a)], and therefore, it is dangerous to move toward the right as dictated by optimal path planning. Instead, the human operator steps backwards [see Fig. 5(b)]. This helps to keep the current features tracked, and hopefully, a larger portion of the room will be visible by stepping back, resulting in more features being captured on the right side. A few new features have been captured but are negligible from the targeted room. The operator makes a lateral right [see Fig. 5(c)] and left [see Fig. 5(d)] motion, but still, no additional features have been captured from the targeted room.

The operator gets closer to the door [see Fig. 5(e)] to have a closer look at the contents of the targeted room. Then, it moves laterally right [see Fig. 5(f)] and left [see Fig. 5(g)] to capture features of the goal room. Few new features have been captured. However, moving toward the room results in tracking failure [see Fig. 5(h)].

Gathering expert episodes is considerably more demanding than gathering SDV episodes. However, this effort of ∼50 h (100 episodes) allows us to imitate this policy, and SLAM manages to move across the transitions without track loss. Even though a dedicated effort of ∼50 h of training leads us to emulate the human policy, it is crucial to fine-tune the trained agent to leverage the full potential of trained DQN for an optimized performance [59].

IV. IMITATION LEARNING METHODOLOGY

The environment can be modeled as a Markov decision process, where an agent sequentially takes actions to maximize the sum of discounted future rewards. In a Markov decision process with the state-space $S := \{1, 2, \ldots, |S|\}$ and action-space $A := \{1, 2, \ldots, |A|\}$, the agent selects an action $a \in A$ at the current state $s \in S$, then the state transits to the next state s' with the state transition probability $P(s'|s, a)$, and the transition incurs a reward $r(s, a, s')$, where $r(s, a, s')$ is the reward function. For convenience, we simply write $r(s_t, a_t, s_{t+1}) =: r_t, t \in \{0, 1, \ldots\}$.

We aim to imitate this policy π_b used by humans to decide an appropriate action, $a_t \in A$, given any current state, $s_t \in S$ (current image observed by the agent), such that the tracking failures can be avoided at the time of transitions. This is made possible by utilizing imitation learning. The agent is first trained under the guidance of a human expert before it can be deployed in an unknown environment. A detailed explanation of our approach is presented in Fig. 6 and Algorithm 1.

The agent is first trained using offline Q-learning under human supervision, where given any state $s_t \in S$, the next appropriate

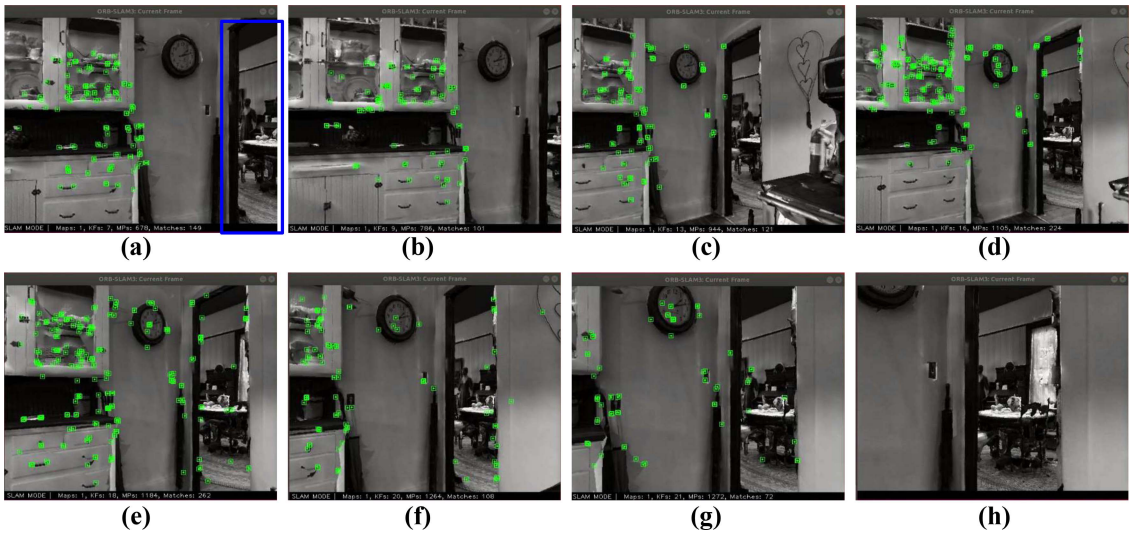

Fig. 5. *Tracking failures during human operator's demo:* The human operator encounters tracking failures while trying to cross the door. The goal is to enter the room on the right [blue box in (a)]. However, the right side of the frame in (a) lacks features (green dots), making this move risky. To address this, the operator steps back (b) and performs lateral motions to the right (c) and left (d), but no features are captured from the target room. The operator then moves closer to the door (e) and repeats the lateral motions to the right (f) and left (g). This time, a few features from the target room are captured (g). Despite this, attempting to enter the room still leads to tracking failure (h).

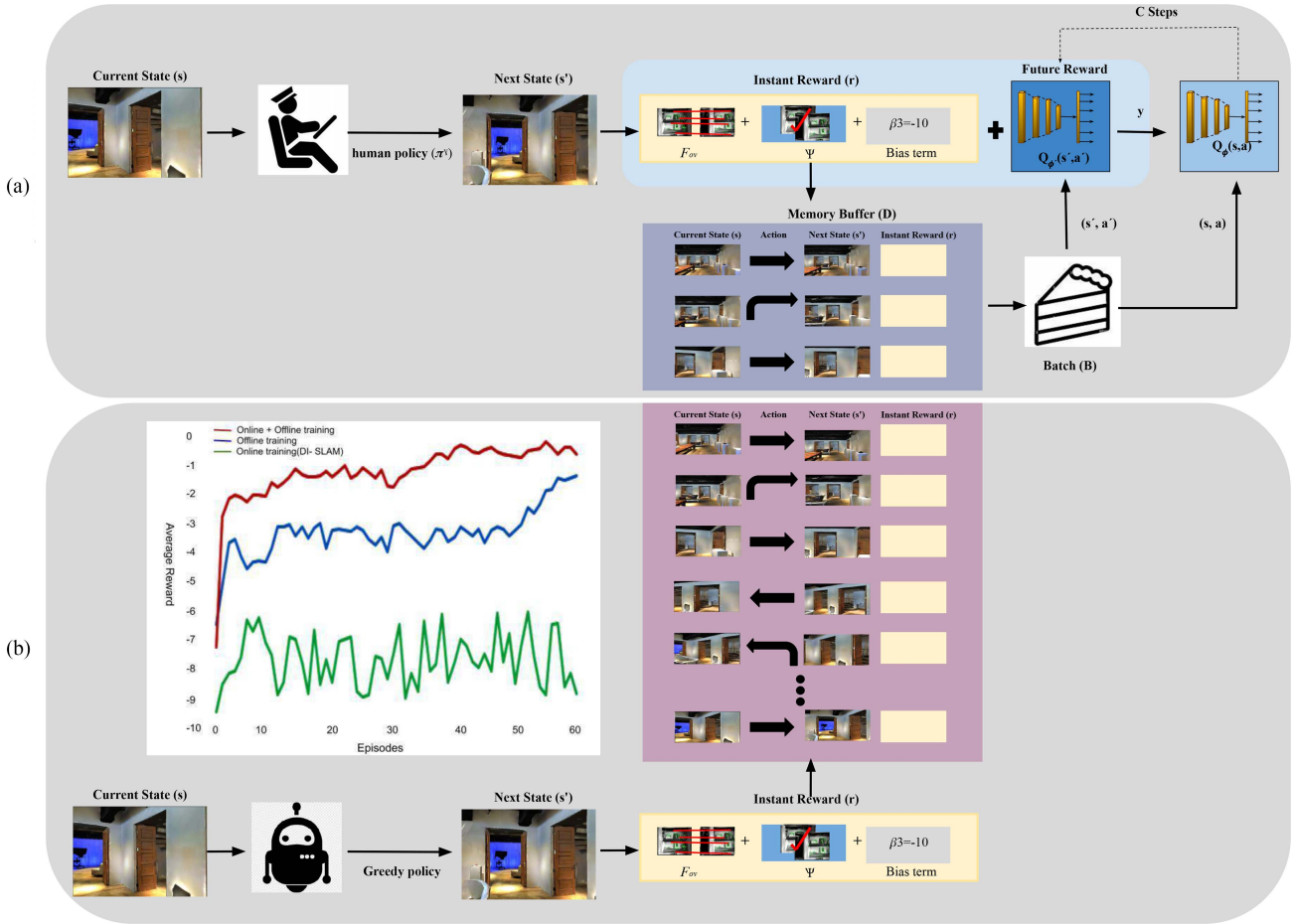

Fig. 6. *Our proposed imitation DQN-based SLAM:* The DQN is trained offline via a human-generated policy first during the offline training phase. Later, this network is fine-tuned via online Q-learning as implemented by Naveed et al. ([2]). The graph clearly shows higher rewards during training when the network is fine-tuned (online) after offline training. (a) Offline training. (b) Online training.

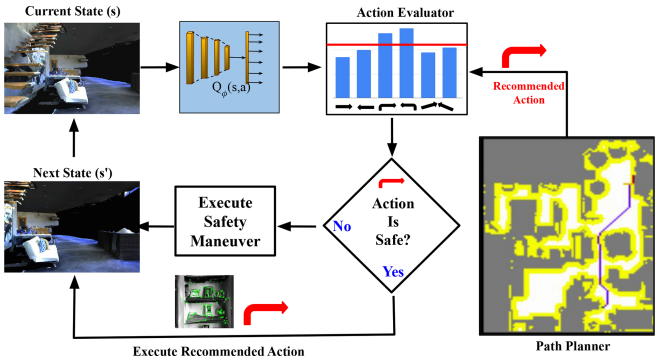

Fig. 7. *Deployment phase of Imitation-SLAM:* During the deployment phase, the agent observes the current image; this is utilized by the trained network DQN to generate Q values (bar graph) against each of the possible six actions, i.e., forward/backward, clockwise/anticlockwise rotation, and strafe left/right translation. These actions are then inspected by the action evaluator block to decide if the Q value of the suggested action (by path planner) is safe or not. If the action is recommended unsafe, the agent executes a safety maneuver in its attempt to avoid SLAM failure.

action $a_t \in A$ is selected by the human expert [see Fig. 6(a)]. This results in a reward r_t and the next state s_{t+1}, which is stored in an experience replay buffer D as a tuple [see Fig. 6(a)]. After a certain number of steps, a random mini-batch of these tuples is used to train the target network [see Fig. 6(a)]. We release our 50-h offline training data for the SLAM community, which contains ∼180 k frames and around 100 sequences. To ensure the optimality of the trained policy, the offline trained Q-network is further trained via online RL as implemented by Naveed et al. [2] 6(b).

Once the training is completed, the agent is deployed in the environment, where it utilizes the trained Q-network to evaluate the quality of an action (to be taken) in a current situation (see Fig. 7). The action in the current state will be recommended by the path planner. If the Q-value of the recommended action is above a defined threshold the action is deemed safe; otherwise, the agent will perform a safety maneuver $a_t \in A$. We evaluate multiple strategies to choose the safety maneuver.

A. Offline Training of DQN

1) Designing the Reward: If the tracker manages to track a large number of its current features, then the reward is positive. Alternatively, if the tracking fails, then the reward is negative. The reward is calculated using

$$r(s,a) = \beta_1 F_{ov} + \beta_2 \psi + \beta_3 \tag{1}$$

where F_{ov} represents the number of overlapping tracking features between two consecutive images, β_1 is the regulation constant for F_{ov} with a value of $10/400$, where 400 represents the upper limit of overlapping tracking features, β_3 is a constant term with a value of -10 so that the reward remains negative at all times, and ψ is the SLAM failure indicator, with the usual value of zero. Its value is raised to 1 when the agent faces a SLAM tracking failure. It is then multiplied with a constant $\beta_2 = -10$, thereby making the reward function highly negative. This high negative reward helps in negative reinforcement,

Algorithm 1: Imitation Q-Learning.

Phase 1 – Offline learning via human policy

Initialize experience replay buffer D
Initialize ϕ and ϕ' to random values
for *Episode* $j = 0, 1, \ldots$ **do**
Observe s_0
for *Time* $t = 0, 1, \ldots, \tau - 1$ **do**
Take an action $a_t \sim \pi_b(\cdot|s_t)$ following the human-generated policy, observe the next state s_{t+1} and reward r_t, and store (s_t, a_t, r_t, s_{t+1}) in the replay buffer D
Uniformly sample a random mini-batch B from D.
Set the target

$$y = \begin{cases} r & \text{if the episode terminates at } s' \\ r + \gamma Q_{\phi'}(s', a') & \text{otherwise} \end{cases}$$

Minimize the mean square error loss through a gradient descent step with respect to the online variable ϕ on the loss function

$$L(\phi) = \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (y - Q_\phi(s,a))^2$$

Every C steps, update the target variable: $\phi' \leftarrow \phi$
Set $s_{t+1} \leftarrow s_t$

Phase 2 – Fine-tuning (Online training)

Restore the buffer D from the offline learning
Restore the trained network ϕ and ϕ' to values obtained from the offline training (phase 1)
for *iteration* $t = 0, 1, \ldots$ **do**
Take action $a_t \in A$ using the epsilon-greedy policy, not from the human policy
Observe s_{t+1}, r_t, store (s_t, a_t, r_t, s_{t+1}) in D, and uniformly sample a random mini-batch B from D.
Minimize the mean square error loss through a gradient descent step with respect to the online variable ϕ on the loss function

$$L(\phi) = \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (y - Q_\phi(s,a))^2$$

Every C steps, $\phi' \leftarrow \phi$ and $s_{t+1} \leftarrow s_t$

Phase 3 – Deployment phase

Activate the trained network
for *mission* M **do**
Observe current state s_i
Observe a_i from path planner
if $Q(s_i, a_i) <$ *threshold* **then**
a_i = safety maneuver (a $\in A$)
execute a_i

which, in turn, ensures SLAM failure avoidance. We have used the same weights as used in [2] and [37].

2) Training the DQN: While the actions a_t for any current state s_t are generated by a human-generated policy, it is important to train the network for autonomous deployment in the future over generalized situations where the network can successfully implement the policy π_b. Therefore, the so-called experience replay buffer D is created, which stores the state transition (s_t, a_t, r_t, s_{t+1}) at each time instant (in phase 1 of

Algorithm 1) where s_t is current state (the current image frame), a_t is the action, r_t is the reward and s_{t+1} is the next state.

To learn a sophisticated policy, instantaneous reward r_t is not enough. A suboptimal move, at the current stage, might lead to success later on. This is represented by the future rewards. For this purpose, we select a mini-batch B of size $|B|$ from the replay buffer D and estimate the future expected return through the target network

$$y = \begin{cases} r & \text{if the episode terminates at } s' \\ r + \gamma Q_{\phi'}(s', a') & \text{otherwise} \end{cases}$$

where $\gamma \in (0, 1)$ is the discount factor, which decides the weights given to future rewards. The Q network is trained by taking the gradient descent step on the mean squared loss function of the temporal-difference error $y - Q_\phi(s, a)$

$$L(\phi) = \frac{1}{|B|} \sum_{(s,a,r,s') \in B} (y - Q_\phi(s, a))^2.$$

After every C steps, Q_ϕ replaces $Q_{\phi'}$ to update the target network. In our training, C is set to 100 steps for each episode.

B. Fine-Tuning (Online Training)

Although the offline DQN has the ability to imitate a near-optimal policy, it suffers from the shortcoming of logged data coverage [60]. In simpler words, the optimality of the policy is highly dependent upon the quality of the available scenarios and how well the offline policy interacts with the environment. Unfortunately, with offline training, it is not possible to verify how optimal the policy is, thereby demanding an online fine-tuning of DQN, which allows the network to interact with a variety of scenarios online while exploiting the already present policy experience of trained DQN. For online fine-tuning, we utilize the deep Q-learning (phase 2 of Algorithm 1) presented in [2]. Here, we show that a DQN trained via a human policy for ∼20 h, when fine-tuned for ∼45 more hours, supersedes the performance of a network that is trained for ∼50 h, only via the human policy π_b (see Fig. 12)

C. Deployment Phase

The DQN, after the fine-tuning, is ready for deployment. Similar to Naveed et al. [2], the DQN $Q_\phi(s, a)$ evaluates the quality of action a given the current state s (phase 3 of Algorithm 1 and Fig. 7).

We have currently utilized the autonomy stack of the MINOS simulator [41]. We provide the start and the goal position within the simulator, and MINOS has the inbuilt path planner, which provides the optimal collision-free path. The path planner output is in the form of a sequence of actions, as highlighted in Fig. 7, rather than a path consisting of multiple way-points.

We evaluate the quality of the action at every step using the expert policy π_b. At each current state, the trained network generates Q values against action. An action is deemed unsafe if the Q value is higher than a defined threshold. We have adopted the value of this threshold from Naveed et al. [2]. For nonsafe action, we disagree with the path planner and choose a safety maneuver to avoid tracking failure (phase 3 of Algorithm 1 and Fig. 7). The path is replanned from that point to the goal, and the above process repeats until the mission is completed or SLAM failure occurs. We are considering only six actions/moves, i.e., $A = \{$forward/backward, clockwise/anticlockwise rotation, strafe left/right translation$\}$.

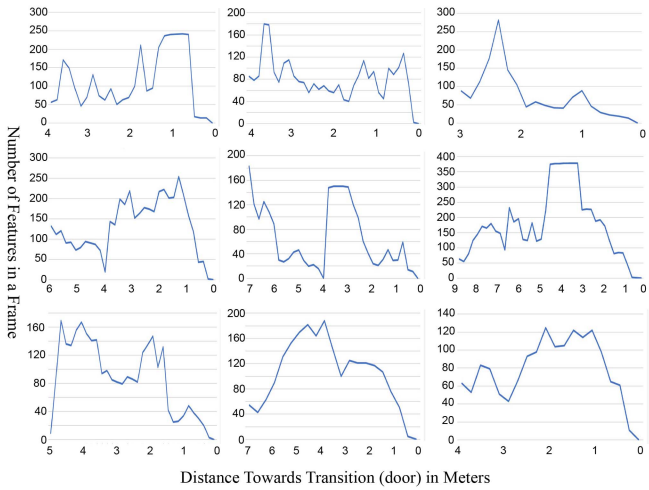

Fig. 8. *Tracking feature analysis*: The number of features tracked drops to zero as we approach the transition (door) in nine different sequences.

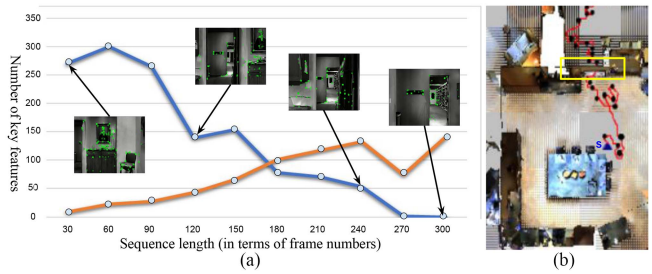

Fig. 9. *Key feature behavior at transitions:* (a) We make an attempt to understand how the features vary when the agent performs a successful transition from one room to another via a door. The blue line represents the number of features (out of the total tracked map features) selected from the current room, and the orange line represents the number of features of the room after the transition which are visible and being tracked while being in the first room. It is not the total number of features tracked but rather the features from the perspective room for a successful transition. (b) Path followed to cross the door.

D. Novel Reward

We updated the reward proposed in DI-SLAM [2] with an additional incentive to reach the goal

$$r(s, a) = \beta_1 F_{\text{ov}} + \beta_2 \psi + \beta_3 + \beta_4 D_g + \beta_5 \tag{2}$$

where D_g is the difference between the previous distance to the goal and the current distance to the goal, β_4 is a constant that is assigned a value of 1, and β_5 is assigned a big positive value (10) if the agent manages to reach the goal, otherwise it stays 0 to ensure that the overall reward is negative.

V. Experimental Results

We use several baselines for comparison with our approach (Imitation-SLAM). In this section, we explain our baselines,

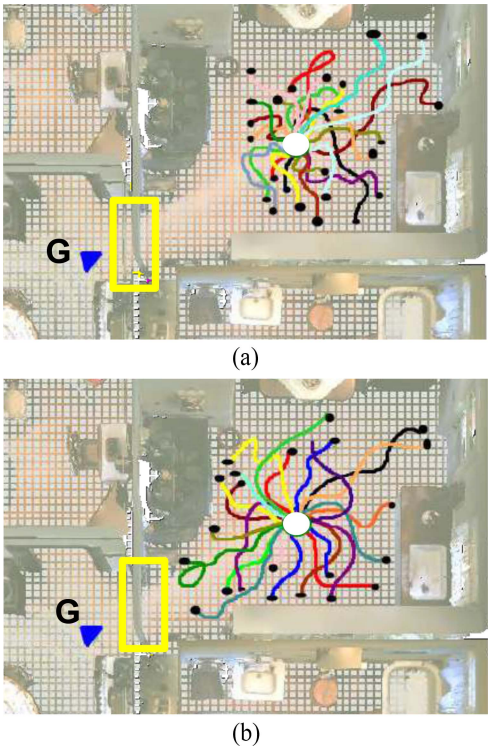

Fig. 10. *Understanding limitations of DI-SLAM [2]:* (a) DI-SLAM aims to avoid tracking failure. In a transition case (exiting a door), this agent surprisingly backtracks instead of moving toward the goal. These trajectories represent the paths taken by DI-SLAM during the training phase, and the black dot at the end of each trajectory indicates the point of tracking failure. Our analysis reveals that DI-SLAM provides no incentive [see (1)] to move toward the goal. (b) We updated the reward of DI-SLAM and included an incentive to move toward the goal [see (2)]. However, even in this case, self-supervised training results in failure to cross the door despite advancing toward the door.

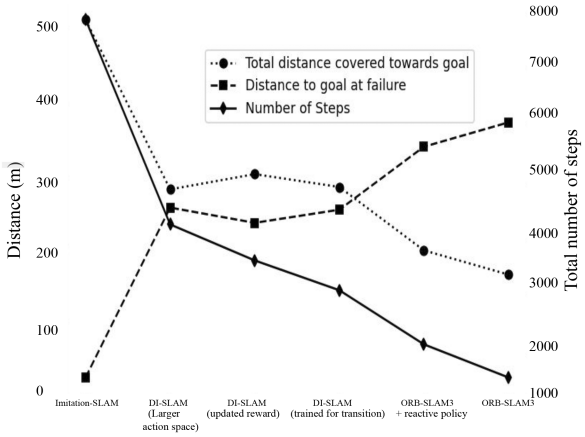

Fig. 11. *Comparison with baselines:* Our Imitation-SLAM takes the most steps and covers the largest distance compared with baselines. It also fails closest to the goal, whereas other baselines fail when they are still far from the goal. The data represents the sum of all 40 navigation episodes.

followed by their performance comparison with our approach in a large-scale evaluation. We have conducted a total of 40 novel experiments, which are divided into transitions, texture-less lobbies, and longer routes (more than 30 m). Each experiment contains at least one or two door transitions in addition to longer routes, featureless walls, and texture-less lobbies. In our experiments, we specify the start and goal locations within the MINOS simulator for each episode. The simulator's inbuilt planner then provides the optimal path based on these defined points. This approach allows us to evaluate the navigation performance systematically within each episode.

A. Adjusting Tracking Parameters of ORB-SLAM3

In contemporary feature-based visual SLAM systems like ORB-SLAM3 [1], a threshold is typically employed to identify tracking failures. If the number of tracked features falls below this threshold, tracking failure is flagged. To investigate whether adjusting or relaxing this threshold is of any use, we analyzed the number of tracked features in transition scenarios (see Fig. 8). As we approach the transition (door in this case), the number of tracked features decreases considerably. In cases where the tracking failure takes place (nine cases in Fig. 8), the number of tracked features falls to zero. This indicates relaxing the tracking threshold will be of little use. This corroborates the findings reported in [27] that adjusting thresholds offers a limited advantage.

We also analyzed the case where visual SLAM successfully managed to cross the transition (see Fig. 9). Here again, the number of tracked features decreases considerably (blue line in Fig. 9). However, visual SLAM manages to capture a few features from the room across the door (orange line in Fig. 9) long before it reaches the transition. Therefore, the visual tracker needs to develop a high-level understanding of the scene, which helps it to seek features from the next room. In our opinion, developing such a high level of understanding may be difficult using the sparse maps generated from visual SLAM. Hence, the active SLAM approaches [26], [27], [61], which rely on these sparse maps, might not be as effective as the deep RL approaches, which take the entire image, having dense information, as input.

B. Baselines

1) ORB-SLAM3 [1]: This is the state-of-the-art visual SLAM. We evaluate the tracking robustness of standard ORB-SLAM3 for the first time in challenging transition scenarios. In the related work [2], [37], analysis was limited to ORB-SLAM [4]. This serves as our first *off-the-shelf* baseline.

2) ORB-SLAM3 With Reactive Policy: If along the path, the number of tracked features falls below a certain threshold the agent is advised to step back and take a safety maneuver. The path is replanned toward the goal before the agent moves forward. We allow the agent multiple attempts until the tracking fails. This policy is inspired by [27].

3) DI-SLAM [2] Trained for Transitions: This deep RL-based approach has been proposed recently to avoid tracking failures in indoor scenarios. In this approach, ORB-SLAM [4] is assisted with the DQN, which flags dangerous actions. First, we replaced ORB-SLAM [4] with ORB-SLAM3 [1]. Second, we retrained (∼12 h, 700 episodes) this approach, especially for transition scenarios, for a fair comparison. Large-scale evaluation results are reported in Section V-E. It is interesting to note an interesting limitation of this method here.

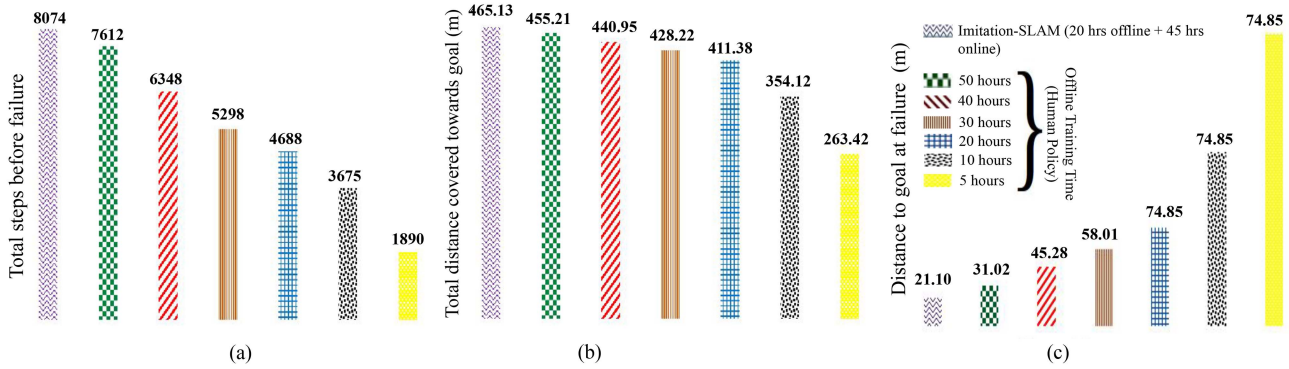

Fig. 12. *Comparison of varying offline training hours with human policy:* The agent trained with human policy for less time (20 h) when fine-tuned with 45 h of unsupervised training produces the best results. All approaches were given 3 chances to complete each episode, and the maximum distance covered in any of the three attempts was considered. (a) Total number of steps. (b) Total distance covered towards goal. (c) Distance to goal at failure.

Analysis of the training episodes revealed interesting findings [see Fig. 10(a)]. The agent is reluctant to move toward the goal, which is just next to the door. Surprisingly, it tries to move backwards in the majority of attempts (episodes). In our opinion, the reason behind this conservative approach is related to the reward [see (1)] of the agent. The main aim of this reward is to avoid tracking failure. There is no incentive to reach the goal. Therefore, the agent is reluctant to move toward the door since it results in a closeup scenario with limited features, which penalizes its reward. It is always safe to take a step back and have a panoramic view of the room.

4) DI-SLAM With Updated Reward: We retrained the DQN with updated reward [see (2)] for approximately ∼22 h, having 1219 episodes in 11 different houses. The agent was trained in challenging transition scenarios [see Fig. 10(b)].

With the updated reward, the agent showed the urgency to move toward the door (and the goal). However, the agent was unable to reach the goal even once during training for ∼12 h (710 episodes). To keep the task simple, the agent was trained on the same path for these episodes. We utilized the pretrained weights of DI-SLAM [2] for initializing our training. DI-SLAM [2] was trained for ∼60 h in similar scenes. Even in this case, the agent backtracks. Perhaps self-supervised learning [7], [16], [62] of the agent requires a substantial amount of training. In this work, we propose an efficient approach based on imitation learning to overcome this barrier.

5) DI-SLAM With Larger Action Space: We adopted the action space reported in the literature (see [2], [35]). In this section, we have expanded the action space from six to eight actions to evaluate the tracking robustness over a large action space. The new action set $A = \{$forward/backward, look up/down (changing roll), clockwise/anticlockwise rotation, lateral left/right translation$\}$.

We retrained the entire DQN with this expanded action set from scratch for ∼50 h for challenging scenarios and longer routes. However, we observed that these additional actions resulted in degraded performance (see Fig. 11). The agent was not able to cross the transition even once and, hence, could not reach the goal either.

This behavior suggests that increasing the action space makes the task even more challenging for the visual SLAM agent. This finding advocates for approaches (including our approach), which add robustness to visual SLAM methods.

C. Training Strategies: Offline or Fine-Tuning

Generating a significantly large amount of data for training the Q-network with human policy (offline) is a challenging task. In this section, we experimentally test if longer offline (human policy) training is necessary. Fig. 12 shows a comparison of results with incremental training duration (via human policy). The performance of the system improves with additional hours of training, which is natural. However, our Imitation-SLAM, with only 20 h of offline training with human policy and 45 h of unsupervised online training, outperforms 50 h of training with human policy. This is a significant reduction in the time requirement for human-assisted training. It must be noted that unsupervised online training is significantly easier where the agent moves within a simulator and keeps collecting experiences.

D. Qualitative Evaluation: Imitation Learning

To evaluate our imitation learning approach, we select a challenging route [see Fig. 13(e)] where the agent needs to cross a low textured corridor [yellow box in Fig. 13(e)] and is required to turn right and finally needs to cross the door to reach the goal. The agent takes a rather longer route to reach the goal [see Fig. 13(e)]. This house was not part of the training episodes.

We analyze the Q-values [box with broken border in Fig. 13(a)] as well as tracking the state of ORB-SLAM3 during multiple instants [see Fig. 13(a)–(d)] along the route. As mentioned earlier, at any given instant, if the action recommended by the path planner is considered prone to tracking failure according to our approach, a safety maneuver is performed. Actions having a Q-value below −28 (indicated by the horizontal red line in Fig. 13) are considered unsafe. The path is replanned after the execution of the safety maneuver.

At the entrance of the corridor [see Fig. 13(a)], the path planner recommends going straight, as it is the shortest path to the goal. However, it is considered unsafe, according to the Q-value of our approach. It can be observed that the majority of the tracked features are on the left side of the frame [see

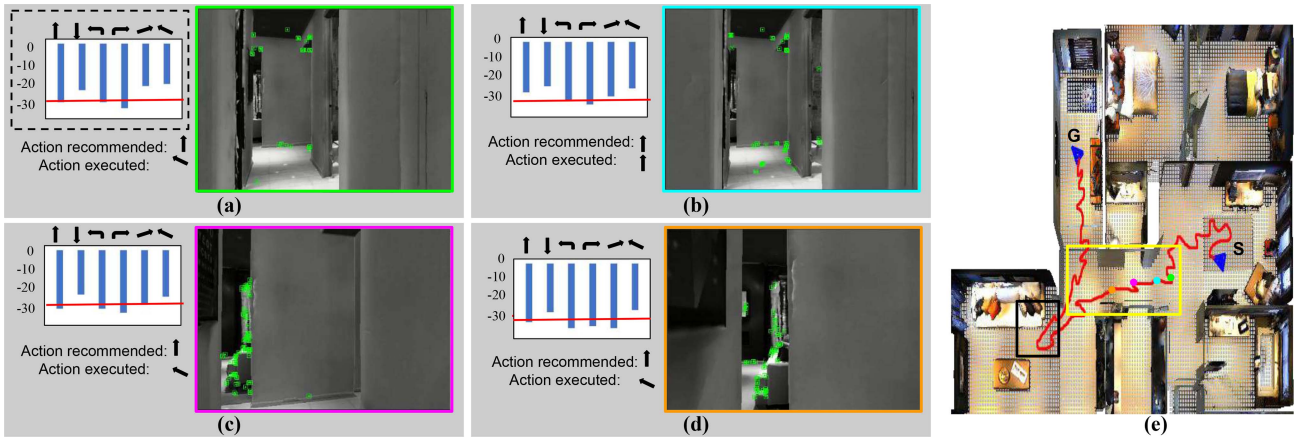

Fig. 13. Qualitative evaluation of imitation learning (crossing a texture-less corridor). See text for details.

Fig. 13(a)]. Our approach disagrees with the path planner, so a safety maneuver is performed. Note that we are not selecting the action having the highest Q-value but instead performing the safety maneuver. We discuss various strategies for safety maneuver in the next section. In the next step [see Fig. 13(b)], most of the tracked features are in the centre of the frame; hence, the action recommended by the path planner is considered safe and executed. In the next two instances [see Fig. 13(c) and (d)] agent needs to move forward, however, it faces a texture-less wall and the majority of features are on the left of the frame. Therefore, the recommended straight actions are ignored.

Instead of making a sharp right turn, which is unsafe in a texture-less corridor, the agent goes ahead and rotates gradually in the textured room [see black box in Fig. 13(e)]. After rotating, the agent exits this room and laterally advances toward the door, which allows it to have sneak peeks at the target room, which allows it to initialize features due to lateral motion. As we observed in the demo by humans, they also tried to capture features before entering the room. Our agent manages to imitate that strategy.

Our approach also manages to reach the goal in longer routes (∼23 m) spanning over several rooms consisting of multiple turns, texture-less lobbies, and multiple doors [see Fig. 14]. Previously active mapping [27] was attempted on shorter routes (∼17 m) having a single turn (around a corner) without any transition (door). Here again, the characteristic lateral motion is evident before crossing each door.

Surprisingly, our imitative learning approach manages to learn strategies not explicitly demonstrated by human trainers. In a few cases, the agent manages to exit the room by walking backwards, which allows it to view the room (instead of the door), and the agent manages to avoid the closeup view while approaching the door. We encourage the reader to view our supplementary video (timestamp: 05:12–05:20).

We also investigate the cases in which our imitation learning approach failed to make the transitions (see Fig. 15). The failure frames (see second row in Fig. 15) indicate that these are mainly close-ups or texture-less scenarios with negligible features. Any action at this stage will result in track loss. Human experts managed to operate ORB-SLAM3 successfully without track loss (blue paths in the first row in Fig. 15). Analyzing the path followed by human experts while operating SLAM indicates that they managed to avoid these close-ups due to their foresight and long-term planning. Our (IL+RL) approach has outperformed

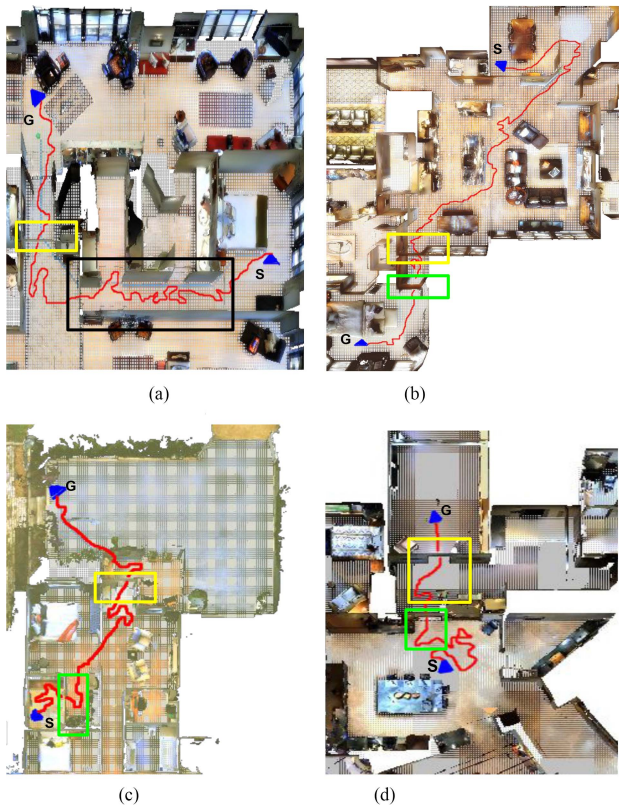

Fig. 14. *Imitation-SLAM (ours) is resilient:* Our approach manages to successfully complete routes red line) extended over several rooms and consisting of multiple turns, texture-less lobbies, and door transitions (doors indicated by black, green, and yellow boxes, respectively). (a) Path Length: 19.73 m with a lobby and a door transition. (b) Path Length: 23.9 m with two door transitions. (c) Path Length: 20.1 m with two door transitions. (d) Path Length: 10.11 m with two door transitions.

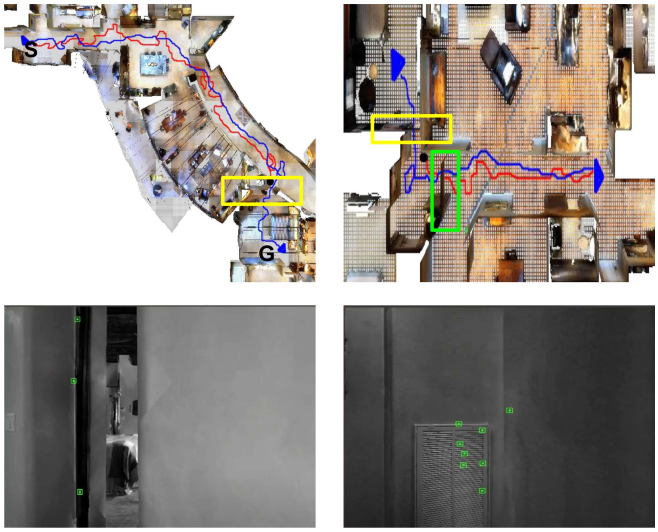

Fig. 15. *Understanding limitations of our imitation learning approach:* Our imitation agent (red track in the top row) ends up in a closeup situation (shown in the bottom row), and hence, the tracking fails. Human operators manage to reach the goal (blue track in the top row) by anticipating these closeups earlier in the journey.

other baselines. However, there is still a performance gap between an expert SLAM operator and our approach, which is understandable.

E. Large-Scale Evaluation

We collected 40 novel navigation episodes, all containing at least one transition, textureless lobby, or a longer path. The episodes varied in path length (distance to goal at the start) from 2 to 36 m, with a total path length of all episodes ∼540 m.

1) Navigation Success: Anderson et al. [63] introduced a performance measuring metric for navigating agents known as SPL, i.e., success weighted by path length as given in the following equation:

$$\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)} \tag{3}$$

where N is the total number of episodes, S_i is the binary indicator of success or failure (0,1) in episode i, l_i is the total distance (optimal) to goal in episode i, and p_i is the distance (optimal or otherwise) taken by the agent to reach the goal. The metric gives decisive importance to navigation success, i.e., reaching a predefined goal, and in case of success, it also evaluates if the path taken by the agent is optimal (it penalizes the agent if it takes a longer suboptimal path). SPL equal to 1 indicates the agent has reached the goal in all episodes and has taken optimal path in all of them. Our Imitation-SLAM reaches the goal in 34 episodes (out of 40) and gets an SPL of 0.387, while the SPL for all other baselines is zero (they never reached the goal in any episode).

2) Resilience to Track-Loss: Even in episodes where our Imitation-SLAM could not reach the goal, it showed considerably more resilient tracking and covered more distance as compared to baselines. Fig. 16 shows the distance covered by all baselines and goal positions in each of the 40 episodes. It must be noted that the distance covered toward the goal is calculated by finding the distance remaining toward the goal at failure and subtracting it from the optimal distance from start to goal. This is done for all approaches. For cumulative understanding, results are shown in the average form in Fig. 11. Our Imitation-SLAM covers the most distance before tracking failure. To emphasize that our agent is moving toward the goal (and not just covering distance), we also show distance to goal at failure [see Fig. 12(c)], which is lowest for Imitation-SLAM.

TABLE V
CHOICE OF SAFETY MANEUVER: A COMPARISON OF SAFETY MANEUVERS IS PRESENTED FOR 40 EXTENSIVE EXPERIMENTS

Method	Success rate	Number of steps	Short routes	Textureless routes	Sharp turns	Transitions Crossed
Number of tries: 1						
Highest Q value action	11/40	9102	✓	✓	✗	11/68
Highest Q value action (excluding backward action)	32/40	7123	✓	✓	✓	49/68
Random action	25/40	8023	✓	✓	✓	35/68
Number of tries: 3						
Highest Q value action	12/40	9699	✓	✓	✗	12/68
Highest Q value action (excluding backward action)	34/40	7489	✓	✓	✓	57/68
Random action	32/40	8853	✓	✓	✓	48/68

3) Choice of Safety Maneuver: Our initial assessment was that the safest action at any stage is the backward motion, which is quite understandable. In the case of exploration, as is happening in our scenario, backward motion means that the agent returns to a scenario, which is known and, hence, safer. This is evident in Fig. 13, where we display Q values of all the actions at different stages. Visual inspection of histograms reveals that backward motion is almost always the most preferred motion with the highest score.

If we choose the backward motion as the safety maneuver, we will get stuck in a loop, since the path planner provides the same path as its looking for the shortest path to the goal. Results (see Table V) support our initial assessment. Selecting the action with the highest Q value (safest action), according to our network, consumes the largest number of steps without progressing toward the goal.

Second, in simpler cases, there is only a single dangerous action (flagged by our network), while the remaining actions are safe. Therefore, random safety action selection allows avoiding the backward action, while avoiding tracking failures. However, in difficult scenarios, e.g., back-to-back transitions, random action also suffers from failures (see Table V) and is unable to reach the goal.

We have leveraged our network to decide the safety maneuver instead of random action in such scenarios. We select the action with the highest Q value sans backward action (see Table V). This strategy proves the most successful.

F. Timing Analysis

Our imitation learning approach is real-time and processes the input image within a few milliseconds (see Table VI). Our

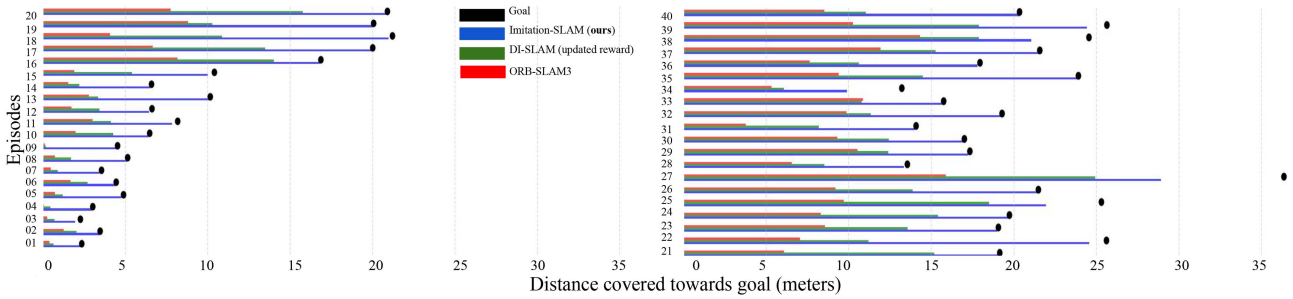

Fig. 16. *Comparison with baselines:* Our Imitation-SLAM reaches the goal in 34 out of 40 episodes, whereas other baselines fail to reach the goal in any episode. All approaches were given 3 chances to complete each episode, and the maximum distance covered in any of the three attempts was considered.

Fig. 17. *Peoplebot:* The robot is built on PIONEER 3-DX and used for real-world experiments.

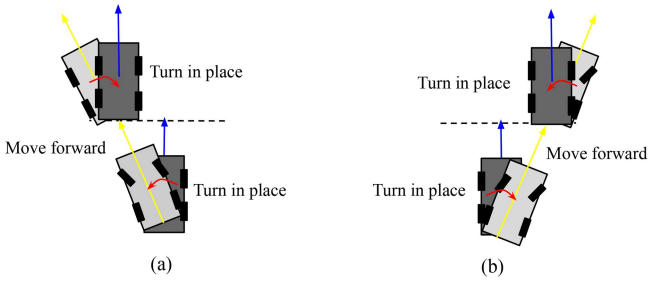

Fig. 18. *Converting strafe action into combination of rotation and translation:* Peoplebot (the robot utilized) is a differential drive robot, it is essential to translate the holonomic actions into nonholonomic actions. The agent rotates slightly, and moves forward in order to imitate the strafe action. (a) Strafe left motion. (b) Strafe right motion.

TABLE VI
TIMING ANALYSIS: OUR IMITATION-SLAM IS REAL TIME

Image	1	2	3	4	5	6	7	8	9	10
Time (msec)	2.10	1.80	1.90	2.0	1.77	2.01	2.12	2.17	1.98	1.73

It takes 2 ms per frame to generate a Q-value.

TABLE VII
REAL-WORLD EXPERIMENTS: OUR IMITATION-SLAM SUCCESSFULLY REACHES GOAL AS COMPARED TO OTHER BASELINES IN PHYSICAL DEMONSTRATIONS

Method	Success rate	Single transition	Double transition	Longer distances
ORB-SLAM3	0/10	✗	✗	✗
DI-SLAM	1/10	✗	✗	✗
Imitation-SLAM (Ours)	8/10	✓	✓	✓

system details include the CPU: Intel(R) Core(TM) i5-9300H CPU at 2.40 GHz and GPU: GeForce GTX 1650.

We have used ResNet-50 [64] architecture as the backbone of our $Q_\phi(s, a)$ network. The input image is resized to $224 \times 224 \times 3$ before it is fed to the network. The size of the output layer is 2048. The output (extracted features) is fed to an MLP with 1 hidden layer (of size 512) to predict the Q values against each of the six actions.

G. Real-World Deployment

In this section, we demonstrate how our approach performs in the real-world. To this end, we deployed our approach to a Peoplebot built over a two-wheeled differential drive PIONEER 3-DX (see Fig. 17). The robot is equipped with an integrated webcam of a connected laptop. This facilitates the robot with real-time visual input. We deployed our solution on the camera feed in live mode. The robot is tele-operated for path planning and obstacle avoidance. In the case of RL+IL-based navigation, the robot can ignore the provided action command and perform a safety maneuver, if the recommended action is considered dangerous.

Our Imitation-SLAM agent is trained for a discrete set of actions (six actions) with a constant step size in the simulator. In our setup, we have defined six discrete actions: forward, backwards, turn left, turn right, strafe left, and strafe right. We have deployed a nonholonomic robot (similar to [65]); therefore, performing strafe actions is not possible. To this end, we implemented a strategy to replicate the strafing actions, as depicted in Fig. 18. The agent rotates slightly and moves forward in order to imitate the strafe action.

The real-world robot motion takes place in the continuous space of actions with a different step size. In addition, the momentum of the robot does not allow an abrupt change in state, e.g., coming to a stop instantly. As shown below, even with these differences, our deep network manages to guide the robot across the transition safely, indicating the robustness of our approach. In our opinion, using the image as input, as compared to sparse inputs (features tracked, sparse point cloud), enables our system to outperform the baselines. Furthermore, there is inherent robustness in visual SLAM, which can cope with slight changes in step sizes and deviations from discrete actions.

The approach was tested in a real-world indoor environment, where the mission was to start from a given point and reach the

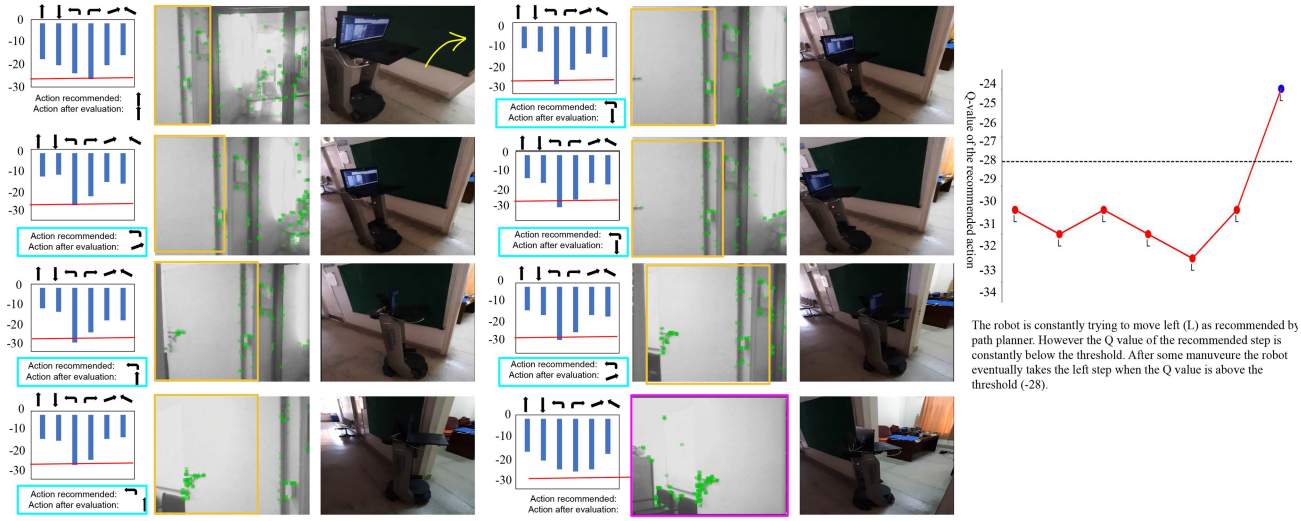

Fig. 19. *Real-world experiment*: The aim of the robot is to enter the room through a door to the robot's left (highlighted by a yellow arrow). The left action recommended by the planner is constantly deemed unsafe by our method, forcing the robot to take random steps, eventually taking the left step and entering the room.

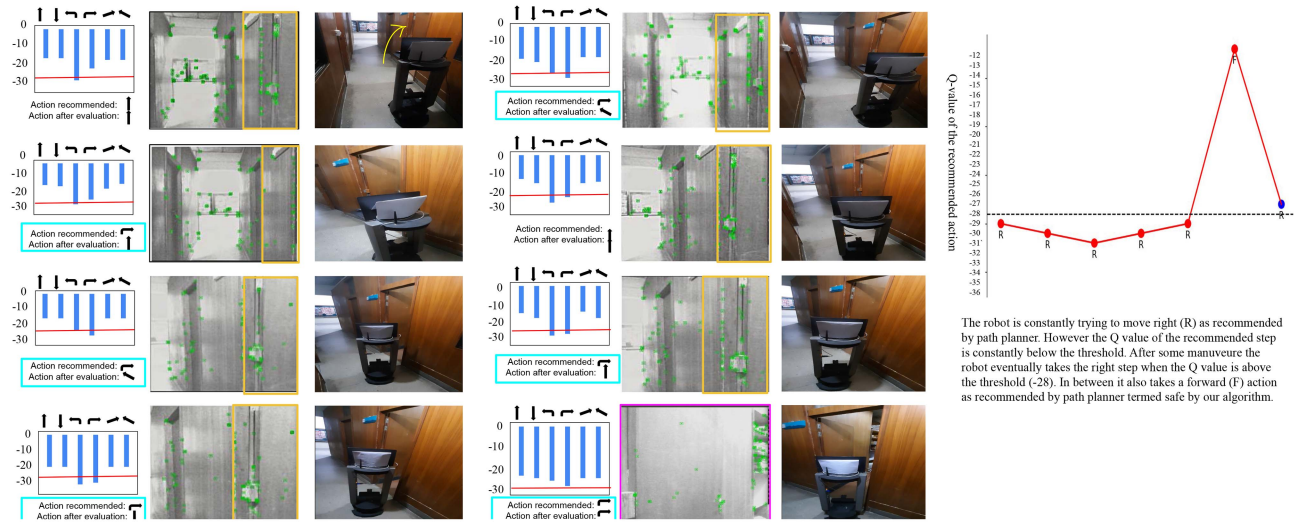

Fig. 20. *Real-world experiment:* The aim of the robot is to enter the room through a door to the robot's right (highlighted by a yellow arrow). The right action recommended by the planner is constantly deemed unsafe by our method, forcing the robot to take random steps, eventually taking the right step and entering the room.

goal position while avoiding visual SLAM failures, especially when crossing the door. During its course, the robot utilized our approach to decide if the recommended action by the path planner is safe to execute. As shown, our approach considerably disagreed with the proposed actions (highlighted by blue boxes in Figs. 19 and 20), by the path planner, near transitions. Instead, safety maneuvers were executed at critical states. Leveraging the advice from our approach, the robot managed to cross multiple transitions, which also included longer distances. We also show that our approach reaches the goal in all scenarios where ORB-SLAM3 and DI-SLAM fail to reach the goal even once, as shown in Table VII. We advise the reader to watch the real-world experiment video uploaded on our GitHub page.[1]

[1]https://tinyurl.com/478maz6u

VI. Conclusion

In this work, we have shown for the first time that even with deep RL-based approaches, visual SLAM still finds it very difficult to move across transitions without tracking failure. We propose a novel imitation learning approach where the dataset is made public for the SLAM community to serve as a head-start to achieve a faster convergence toward optimal tracking performance and to assist the community in developing robust SLAM solutions.

Multisensor SLAM, involving sensor redundancy, seems an obvious way forward. In this supplementary video (timestamp: 01:13–01:19), we demonstrate that even stereo SLAM (ORB-SLAM2 [66]) suffers from tracking failures in commonly occurring scenarios. Proprioceptive sensors (IMU, Gyro) may be

deployed to increase SLAM robustness. However, there is no guarantee that multisensor setup will not fail. It has been shown that gyros [67] and accelerometers [68] malfunction in the presence of intentional environmental noise. Such noise is likely to exist in the environment naturally (mobile phone ringtone). We advocate that improving the robustness of individual sensors ultimately improves the robustness of multisensor setups. Avoiding understanding the failures of individual sensors by leveraging sensor redundancy does not guarantee a resilient system. This work provides a platform to evaluate failures of visual systems extensively.

REFERENCES

[1] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[2] K. Naveed, M. L. Anjum, W. Hussain, and D. Lee, "Deep introspective SLAM: Deep reinforcement learning based approach to avoid tracking failure in visual SLAM," *Auton. Robots*, vol. 46, no. 6, pp. 705–724, 2022.

[3] W. Wang et al., "Tartanair: A dataset to push the limits of visual SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2020, pp. 4909–4916.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

[5] A. Nair, B. McGrew, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Overcoming exploration in reinforcement learning with demonstrations," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 6292–6299.

[6] Y. Zhu et al., "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 3357–3364.

[7] M. Savva et al., "Habitat: A platform for embodied AI research," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 9339–9347.

[8] A. Khosla, B. An An, J. J. Lim, and A. Torralba, "Looking beyond the visible scene," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3710–3717.

[9] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik, "Cognitive mapping and planning for visual navigation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2616–2625.

[10] S. Brahmbhatt and J. Hays, "Deepnav: Learning to navigate large cities," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5193–5202.

[11] T. Sattler, Q. Zhou, M. Pollefeys, and L. Leal-Taixe, "Understanding the limitations of CNN-based absolute camera pose regression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3302–3312.

[12] M. Raghu, C. Zhang, J. Kleinberg, and S. Bengio, "Transfusion: Understanding transfer learning for medical imaging," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[13] M. Bojarski et al., "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.

[14] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," 2018, *arXiv:1812.03079*.

[15] N. Djuric et al., "Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 2095–2104.

[16] V. Mnih et al., "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[17] D. A. Pomerleau, "Alvinn: An autonomous land vehicle in a neural network," in *Proc. Adv. Neural Inf. Process. Syst.*, 1988, vol. 1.

[18] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2174–2182.

[19] A. Sauer, N. Savinov, and A. Geiger, "Conditional affordance learning for driving in urban environments," in *Proc. Conf. Robot Learn.*, 2018, pp. 237–252.

[20] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 4693–4700.

[21] D. Wang, C. Devin, Q.-Z. Cai, F. Yu, and T. Darrell, "Deep object-centric policies for autonomous driving," in *Proc. Int. Conf. Robot. Autom.*, 2019, pp. 8853–8859.

[22] U. Muller, J. Ben, E. Cosatto, B. Flepp, and Y. Cun, "Off-road obstacle avoidance through end-to-end learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 18, 2005.

[23] M. H. Ikram, S. Khaliq, M. L. Anjum, and W. Hussain, "Perceptual aliasing++ : Adversarial attack for visual SLAM front-end and back-end," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4670–4677, Apr. 2022.

[24] V. Indelman, L. Carlone, and F. Dellaert, "Planning in the continuous domain: A generalized belief space approach for autonomous navigation in unknown environments," *Int. J. Robot. Res.*, vol. 34, no. 7, pp. 849–882, 2015.

[25] G. Klein and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2009, pp. 83–86.

[26] C. Mostegel, A. Wendel, and H. Bischof, "Active monocular localization: Towards autonomous monocular exploration for multirotor mavs," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 3848–3855.

[27] X. Deng, Z. Zhang, A. Sintov, J. Huang, and T. Bretl, "Feature-constrained active visual slam for mobile robot navigation," in *Proc. Int. Conf. Robot. Autom.*, 2018, pp. 7233–7238.

[28] M. Salas, W. Hussain, A. Concha, L. Montano, J. Civera, and J. Montiel, "Layout aware visual tracking and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 149–156.

[29] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2320–2327.

[30] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 834–849.

[31] A. Flint, D. Murray, and I. Reid, "Manhattan scene understanding using monocular, stereo, and 3D features," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 2228–2235.

[32] M. W. Hussain, J. Civera, and L. Montano, "Grounding acoustic echoes in single view geometry estimation," in *Proc. AAAI Conf. Artif. Intell.*, Jun. 2014, vol. 28, no. 1, pp. 2760–2766.

[33] W. Hartmann, M. Havlena, and K. Schindler, "Predicting matchability," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 9–16.

[34] S. Rabiee and J. Biswas, "IVOA: Introspective Vision for Obstacle Avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Macau, China, 2019, pp. 1230–1235, doi: 10.1109/IROS40897.2019.8968176.

[35] D. M. Saxena, V. Kurtz, and M. Hebert, "Learning robust failure response for autonomous vision based flight," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 5824–5829.

[36] Z. Habibi, G. Caron, and E. M. Mouaddib, "3D model automatic exploration: Smooth and intelligent virtual camera control," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 612–626.

[37] V. Prasad et al., "Learning to prevent monocular SLAM failure using reinforcement learning," in *Proc. 11th Indian Conf. Comput. Vis., Graph. Image Process.*, 2018, pp. 1–9.

[38] X.-Y. Dai, Q.-H. Meng, S. Jin, and Y.-B. Liu, "Camera view planning based on generative adversarial imitation learning in indoor active exploration," *Appl. Soft Comput.*, vol. 129, 2022, Art. no. 109621.

[39] J. Ho and S. Ermon, "Generative adversarial imitation learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016.

[40] H.-T. L. Chiang, A. Faust, M. Fiser, and A. Francis, "Learning navigation behaviors end-to-end with autorl," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 2007–2014, Apr. 2019.

[41] M. Savva, A. X. Chang, A. Dosovitskiy, T. Funkhouser, and V. Koltun, "MINOS: Multimodal indoor simulator for navigation in complex environments," 2017, *arXiv:1712.03931*.

[42] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 3354–3361.

[43] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2012, pp. 573–580.

[44] F. A. A. Cheein, C. De La Cruz, T. F. Bastos, and R. Carelli, "Slam-based cross-a-door solution approach for a robotic wheelchair," *Int. J. Adv. Robot. Syst.*, vol. 6, no. 3, pp. 239–248, 2009.

[45] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[46] A. Kowdle, Y.-J. Chang, D. Batra, and T. Chen, "Scribble based interactive 3D reconstruction via scene co-segmentation," in *Proc. IEEE Int. Conf. Image Process.*, 2011, pp. 2577–2580.

[47] H. Ahmad, S. M. Usama, W. Hussain, and M. L. Anjum, "A sketch is worth a thousand navigational instructions," *Auton. Robots*, vol. 45, pp. 313–333, 2021.
[48] P. Ammirato, P. Poirson, E. Park, J. Košecká, and A. C. Berg, "A dataset for developing and benchmarking active vision," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 1378–1385.
[49] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and LiDAR data set," *Int. J. Robot. Res.*, vol. 30, no. 13, pp. 1543–1552, 2011.
[50] J. Čech, J. Sanchez-Riera, and R. Horaud, "Scene flow estimation by growing correspondence seeds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3129–3136.
[51] M. Smith, I. Baldwin, W. Churchill, R. Paul, and P. Newman, "The new college vision and laser data set," *Int. J. Robot. Res.*, vol. 28, no. 5, pp. 595–599, 2009.
[52] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Comput. Vis.–ECCV 2012: 12th Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.
[53] G. Georgakis, M. A. Reza, A. Mousavian, P.-H. Le, and J. Košecká, "Multiview RGB-D dataset for object instance detection," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 426–434.
[54] M. Burri et al., "The euRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.
[55] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and slam," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 1524–1531.
[56] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison, "Scenenet RGB-D: Can 5 m synthetic images beat generic imagenet pre-training on indoor segmentation?," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2678–2687.
[57] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *Int. J. Robot. Res.*, vol. 36, no. 1, pp. 3–15, 2017.
[58] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of michigan north campus long-term vision and lidar dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.
[59] Y. Luo, J. Kay, E. Grefenstette, and M. P. Deisenroth, "Finetuning from offline reinforcement learning: Challenges, trade-offs and practical solutions," 2023, *arXiv:2303.17396*.
[60] A. Wagenmaker and A. Pacchiano, "Leveraging offline data in online reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 35300–35338.
[61] M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, "Active SLAM: A review on last decade," *Sensors*, vol. 23, no. 19, 2023, Art. no. 8097.
[62] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50 k tries and 700 robot hours," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3406–3413.
[63] P. Anderson et al., "On evaluation of embodied navigation agents," 2018, *arXiv:1807.06757*.
[64] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[65] T. Soualhi, N. Crombez, A. Lombard, S. Galland, and Y. Ruichek, "Curiosity-driven learning for visual control of nonholonomic mobile robots," in *Proc. 21st Int. Conf. Ubiquitous Robots*, 2024, pp. 273–280.
[66] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[67] Y. Son et al., "Rocking drones with intentional sound noise on gyroscopic sensors," in *Proc. 24th USENIX Secur. Symp. (USENIX Secur. 15)*, 2015, pp. 881–896.
[68] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2017, pp. 3–18.

Kanwal Naveed received the bachelor's degree in electronics engineering and the master's degree in electrical engineering from International Islamic University Islamabad, Islamabad, Pakistan, in 2011 and 2013, respectively. She is currently working toward the Ph.D. degree in methods to avoid tracking failures in visual SLAM with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan.

Her research interests include machine learning, robotics, and control systems.

Wajahat Hussain received the bachelor's degree in electrical engineering from the National University of Sciences and Technology (NUST), Islamabad, Pakistan, in 2007, the master's degree in signal processing from Technical University Munich, Munich, Germany, in 2010, and the Ph.D. degree in robotics from the University of Zaragoza, Zaragoza, Spain, in 2016.

He is currently an Associate Professor with NUST. His research interests include data driven scene understanding for vision, robotics and graphics.

Irfan Hussain received the second level master's degree in automatica and control technologies from Politecnico Di Torino, Turin, Italy, in 2012 and the Ph.D. degree in robotics from the University of Siena, Siena, Italy, in 2017.

He is currently an Associate Professor in Robotics and Mechanical Engineering with Khalifa University, Abu Dhabi, UAE. He has authored or coauthored more than 100 peer-reviewed papers in his research field. His research interests include embodied and cognitive intelligence in robots with applications in marine, healthcare, and industrial robotics.

Dr. Hussain is currently an Associate Editor of proceedings of the Institution of Mechanical Engineers, Part C. Research Topic Editor on "Wearable Robots and Sensorimotor Interfaces: Augmentation, Rehabilitation, Assistance or substitution of human sensorimotor function" (ID 21096) and *Frontiers in Neurorobotics*. He is also an Associate Editor for IEEE Robotics and Automation Society (RAS) International Conference on Robotics and Automation (ICRA) 2023, ICRA-22, ICRA-21, 2018 RAS/EMBS International Conference on Biomedical Robotics and Biomechatroncis.

Donghwan Lee (Member, IEEE) received the B.S. degree in electronic engineering from Konkuk University, Seoul, South Korea, in 2008, the M.S. degree in electrical and electronic engineering from Yonsei University, Seoul, South Korea, in 2010, and the M.S. degree in mathematics and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2017.

He was a Postdoctoral Research Associate with the Coordinated Science Laboratory, University of Illinois at Urbana–Champaign, Champaign, IL, USA, from 2017 to 2019. He is currently an Assistant Professor with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea. His current research interests include reinforcement learning, control theory, and optimization.

Muhammad Latif Anjum received the bachelor's degree in electrical engineering from UET Lahore, Lahore, Pakistan, in 2007, the M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, South Korea, in 2010, and the Ph.D. degree in mechatronics engineering from Politecnico di Torino, Turin, Italy, in 2015.

He is currently an Assistant Professor of Electrical Engineering with the School of Electrical Engineering and Computer Science, National University of Sciences and Technology, Islamabad, Pakistan, where he is also heading Robotics and Machine Intelligence (ROMI) Lab. His research interests include visual navigation, path planning, autonomous systems, and machine intelligence.