2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Social Navigation for Mobile Robots in the Emergency Department

Angelique M. Taylor, Sachiko Matsumoto, Wesley Xiao, and Laurel D. Riek

2021 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9077-8/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICRA48506.2021.9561897

Abstract— **The emergency department (ED) is a safety-critical environment in which healthcare workers (HCWs) are overburdened, overworked, and have limited resources, especially during the COVID-19 pandemic. One way to address this problem is to explore the use of robots that can support clinical teams, e.g., to deliver materials or restock supplies. However, due to EDs being overcrowded, and the cognitive overload HCWs experience, robots need to understand various levels of patient acuity so they avoid disrupting care delivery. In this paper, we introduce the Safety-Critical Deep Q-Network (SafeDQN) system, a new acuity-aware navigation system for mobile robots. SafeDQN is based on two insights about care in EDs: high-acuity patients tend to have more HCWs in attendance and those HCWs tend to move more quickly. We compared SafeDQN to three classic navigation methods, and show that it generates the safest, quickest path for mobile robots when navigating in a simulated ED environment. We hope this work encourages future exploration of social robots that work in safety-critical, human-centered environments, and ultimately help to improve patient outcomes and save lives.**

I. Introduction

The emergency department (ED) is a high-stress, fast-paced safety-critical environment, which is frequently over-crowded, under-staffed, and under-funded, all of which has become worse during the COVID-19 pandemic. ED health-care workers (HCWs) are overworked and overstressed, leading to high rates of burnout and adverse mental health outcomes [1], [2]. Thus, many roboticists have explored the use of robots in clinical settings to help lessen the burden of HCWs, such as by engaging in non-value added tasks, like supporting material delivery and patient triage [3], [4], [5], [6], and providing support to patients at the bedside when providers are not available [7], [8], [5].

To perform these tasks, robots must understand the context of complex hospital environments and the people working around them. Furthermore, robots must make intelligent, acuity-aware navigation decisions that take this context into account. For example, a robot will encounter different groups of HCWs, some of whom may be performing safety-critical work on a high-acuity patient (e.g., severely ill, such as having a heart attack or stroke), while others may be conversing [9], [2], [10]. If a robot interrupts a clinical team treating a high-acuity patient, it could lead to adverse patient harm. As such, the goal of our work is to design robots that can navigate through these safety-critical environments, incorporating key clinical contextual information.

While there has been prior work in robotics in safety-critical navigation [11], social navigation [12] and deployment of robots in general hospital wards [5], [6], to our knowledge very few autonomous mobile robots have been deployed in the ED. This represents a major gap, as the ED is a unique type of safety-critical setting which presents novel navigation challenges for robots [1]. For example, patients are often placed in the hallways of the ED, making it a more challenging setting to navigate. Hallways are also often cluttered with carts and equipment, and HCWs often need to quickly move patients on gurneys through the ED. Also, patients often have high acuity conditions which require immediate medical attention.

In this paper, we address this gap by introducing the Safety-Critical Deep Q-Network (SafeDQN) system, which enables a robot to navigate in the ED while taking patient acuity level into account. We employ a reinforcement learning (RL) agent with the objective to deliver supplies to HCWs. As the agent generates a path, its objective is to avoid high-acuity patient areas as much as possible to maximize its cumulative rewards. SafeDQN uses a neural network (NN) to model state-action transitions as the agent explores the activities in the ED.

We estimate the acuity of patients in video data using two intuitions: 1) high-acuity patients tend to require fast, precise treatment which ultimately results in very dynamic motion by the HCWs and 2) high-acuity patients tend to require treatment from more HCWs than low acuity patients. When the robot visually observes a team in the ED, it detects the number of HCWs observed and tracks their average body movement to model the acuity level of a patient. By avoiding areas of high-acuity patients, we ensure that SafeDQN does not interrupt patient treatment.

The main contribution of our work is that we introduce a new computational model of patient acuity to enable robots to socially navigate without introducing additional causes of patient harm. To the best of our knowledge, this is the first work that presents an acuity-aware navigation method for robots in safety-critical settings. We hope this work inspires robotics researchers to enable robots to work safely in safety-critical, human-centered environments, and ultimately help improve patient outcomes, alleviate clinician workload, and support people in numerous other applications like supporting first responders.

II. Related Work

A. Social Navigation in Robotics

Navigation is a fundamental problem in robotics with the goal of generating the shortest path from a starting location to a goal location while avoiding obstacles [12], [13]. In the field of human-robot interaction (HRI) specifically, the

The authors are in Computer Science and Engineering, UC San Diego, USA. This work was supported by the National Science Foundation under Grant No. 1734482 and AFOSR Grant No. FA9550-18-1-0125

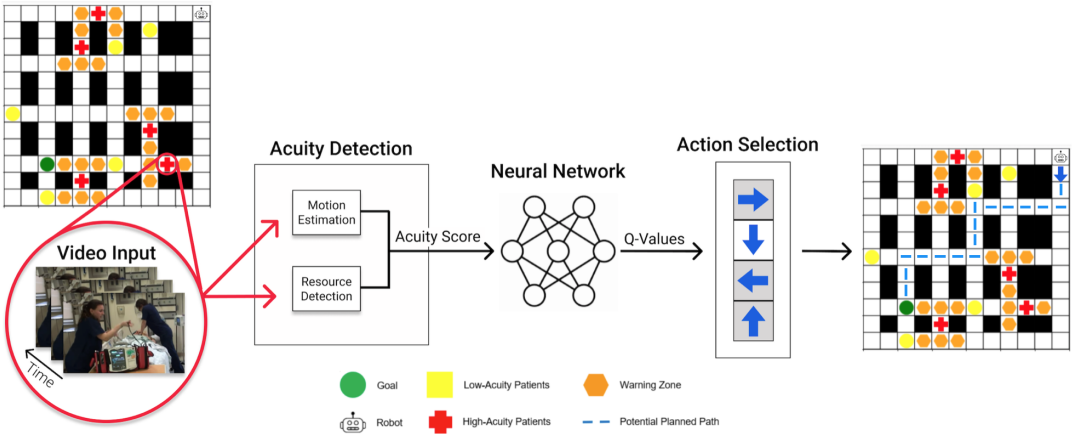

Fig. 1: SafeDQN System Architecture. SafeDQN performs acuity detection using Eq. 2. Then, it assigns an acuity score (AS) to each observed patient in the ED and uses the AS to assign a reward to the agent. The state-action value transitions for the agent exploration are modeled by a neural network, which predicts optimal Q-values.

objective is to generate a socially compliant path that obeys social norms dictated by human nature [12], [13]. For example, people often stand in group formations which robots can sense and use as part of how they plan [14], [15].

Some prior work uses context to improve a robot's social awareness during task execution [16], [17]. Some of these methods include environmental constraints in their planning such as robots that consider other agents' plans to avoid resource conflicts [18], and find paths that favor areas with high bandwidth or other resources [19]. Robots can also learn to avoid areas that might distract people from working, such as by imitating typical human paths using Hidden Markov Models (HMMs) [20] or by learning to avoid areas that afford working [21]. In contrast to prior work, we investigate social navigation in safety critical settings, where understanding the situational context can potentially save lives [22].

B. Navigation and Reinforcement Learning

Reinforcement learning (RL) is frequently used in planning. RL typically models an agent as a *Markov Decision Process (MDP)* which interacts with an environment E through sequential states s_t. The goal of the agent is to generate a policy π which maps states to actions, such that the agent maximizes its cumulative reward. This process obeys Bellman's equation, which generates the optimal action-value function. At each timestep, the agent takes an action $A = 1, \ldots, K$ and is given a reward r_t where the cumulative reward for all timesteps is $R_t = \sum_{t=0}^{\infty} \gamma^t r_t$ where t is the timestep where the agent's interaction with E ends. γ is the discount factor, which corresponds to a priority on future rewards given to the agent. For example, a high γ corresponds to a priority on future rewards and prioritizes immediate rewards. The agent's actions are chosen using Q-values to estimate the optimal action the agent should take to receive the highest cumulative reward.

In our work, we use a model-free approach, as it is challenging to model the ED, due to its frequently changing nature, sensor occlusion, noise, and the inability to mount sensors due to privacy concerns [22]. There are many model-free approaches in the literature, such as actor-critic and policy search methods, which search directly over a policy space to maximize the reward [23]. Another model-free method which has been extensively studied is Q-learning, a temporal difference learning algorithm that learns the action-value function to choose the optimal action at each timestep [24], [25]. We employ Q-learning, as it is well-suited for learning through exploration. It can efficiently learn in discrete action spaces, and has a useful exploration property which is beneficial for planning.

Mnih et al. [26] proposed a deep Q-network (DQN) to learn a policy for Atari 2600 games, which surpassed human performance. The authors entered the raw pixels from the game screen into convolutional neural networks (CNNs), which outputted a state-action value. This approach is beneficial because it uses an experience replay buffer to store the agent's experiences $e_t = (s_t, a, r_t, s_{t+1})$ in replay memory $M = (e_1, \ldots, e_t)$ [27]. Also, it maintains a learning network $Q(s, a; \theta)$ with updated parameters, and a target Bellman network with the original parameters θ^-.

$$L_i(\theta_i) \leftarrow \mathbf{E}[(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i))^2] \quad (1)$$

DQN networks are trained using samples from mini-batches in M and the parameters are updated using an optimizer such as stochastic gradient descent (SGD) or RMSProp [26], [28]. The agent's behavior is based on ϵ-greedy policy which employs a random strategy at probability ϵ and a greedy strategy with probability 1-ϵ. Many improvements have been proposed to improve DQN, such as Double Q-Learning [24] and Dueling Q-Networks [29]. The key difference between our work and DQN is that we aim to build a computational model of patient acuity togenerate the reward for the agent. Another approach, Safe Reinforcement Learning (SafeRL) [30], learns policies that maximize the expectation of accumulated return, while respecting security constraints in the learning and deployment process. Our work

is similar to SafeRL in that by modeling patient acuity, we are inherently modeling risk. However, to our knowedge, it does not explore visually modeling risk, particularly patient acuity, as we do.

III. Safety-Critical Deep Q-Network (SafeDQN)

We designed SafeDQN drawing on inspiration both from the Emergency Medicine literature as well as by engaging in several years of co-design work with ED staff, to help us understand the ED's operational context [9], [2]. For instance, ED HCWs explained that high-acuity patients typically require more HCWs to attend to them, and that they sometimes treat patients in the hallways when rooms are full. Fig. 1 and Algorithm 1 present an overview of SafeDQN.

When designing SafeDQN, we made several assumptions. First, we model the ED as a discrete 2D, grid-world environment, as commonly done in the literature [31], [32]. Second, typically in the ED, patients are being admitted, transferred, and discharged; therefore, we assume a discrete number of patients are being treated in the ED at any given time. Third, when the agent moves from one position to the next, it observes a video of patient treatment at the new position. Finally, we assume that patients have stationary positions on the map as they are not frequently moved in a typical ED.

SafeDQN models patient acuity using two intuitions: 1) high-acuity patients tend to require fast, precise treatment which ultimately results in very dynamic motion by the ED workers and 2) high-acuity patients tend to require treatment from more ED workers than low acuity patients. We use videos to represent high and low acuity patients. SafeDQN measures the maximum number of people P, which reflects the amount of resources needed to treat a patient. This is calculated by employing the object detector proposed by Redmond et al. [33] which determines the number of people in an image. Second, it calculates relative motion in a video $\vec{v}$, which reflects the chaotic motion of healthcare workers as they treat patients. Typically, high-acuity patients have time-sensitive conditions which requires healthcare workers to move quickly and dynamically. $\vec{v}$ is calculated using the average from one of two methods which include optical flow [34] and keypoint detector [35] over the course of a video.

SafeDQN estimates patient acuity, denoted Acuity Score ($AS \in [0, 1]$). The AS is inspired by Term Frequency Inverse Document Frequency (tf-idf) from the Natural Language Processing (NLP) community [36]. Tf-idf models the importance of words in documents relative to a collection of documents and has been used to predict topics of documents. We translate this equation to our problem domain but instead of placing importance on words, we place importance on the amount of resources needed to treat patients. Another key difference is that we model the number of patients being treated in the ED instead of documents. We weight this equation by the relative motion magnitude from a given video such that videos with more motion will generate a higher AS.

$$AS \leftarrow \vec{v}\frac{|P|}{1+|T|} \tag{2}$$

ALGORITHM 1: Safety-Critical Deep Q-Learning

Initialize Replay Memory M
Initialize Training Q-Network $Q(s, a; \theta)$
Initialize Target Q-Network $Q(s, a; \theta^-)$ with weights $\theta^- = \theta$
for *episode = 1:M* **do**
  Data generation phase:
  Initialize start state to s_t
  for *t=1:T* **do**
    a_t = $\max_a Q(s_t, a; \theta)$
    $rn \leftarrow$ random number between 0 and 1
    if $rn < \epsilon$ **then**
      $a_t \leftarrow$ select random a_t
    end
    s_{t+1}, $r_t \leftarrow$ ExecuteAction(a_t)
    Store transition (s_t, a_t, r_t, s_{t+1}) in M
  end
  Training phase:
  Randomize a memory M for experience replay
  for *k=1:n* **do**
    Sample random minibatch B from M
    while $B \neq \{\}$ **do**
      r_k = RewardFunction()
      y_k = $\begin{cases} \text{r}_k, \text{ if step } k+1 \text{ terminal} \\ \text{r}_k + \gamma y_k - Q(s_k, a_k; \theta^-), \text{ otherwise} \end{cases}$
      Perform gradient descent on loss
      $(y_k - Q(s_k, a_k; \theta))^2$ w.r.t the network parameters
    end
  end
  After every C-episodes sync θ^- with θ
end

- $\vec{v}$ average velocity of all image frames from a motion estimation method in a given video.
- $|P|$ is the maximum number of people in a given video.
- $|T|$ number of patients being treated in the ED.

SafeDQN employs a neural network (NN) where the input layer is the size of the state-space (i.e., size of the 2D grid), followed by two hidden layers and a Rectified Linear Unit layer (ReLU) where the final hidden layer uses ReLU as the activation function which is the size of our action-space (i.e., number of actions) (See Fig. 1). We use a mean squared error loss function and we conduct ablation studies to explore the use of one of three optimizers which include Adam, RMSProp, and Stochastic Gradient Descent (SGD). SafeDQN outputs the Q-values and selects the action with the maximum Q-value, where higher Q-values correspond to the optimal actions. The SafeDQN agent employs a discrete set of actions which include "move up", "move down", "move left", "move right" (See Fig 1).

The reward function, r, captures the necessary behaviors for our agent to navigate in an environment while taking into account the acuity of patients being treated. r prevents the agent from going outside the grid ($r = .05 * R * C$ where R is number of rows and C is number of columns) or re-visiting a cell ($r = -.25$); enables it to avoid navigating in areas of patient rooms ($r = -.75$), low-acuity patients ($r = -.15$), and high-acuity patients ($r = -.30$); and ensures it reaches its goal ($r = +1$). In all other cases, ($r = -.04$).

IV. Implementation

For our experimental testbed, we deployed our agent in a simulation environment that we designed in OpenAI Gym [37], a standard library used to test RL systems. Each position on the map corresponds to an observation from our dataset (see Section IV-A). We encode random locations of

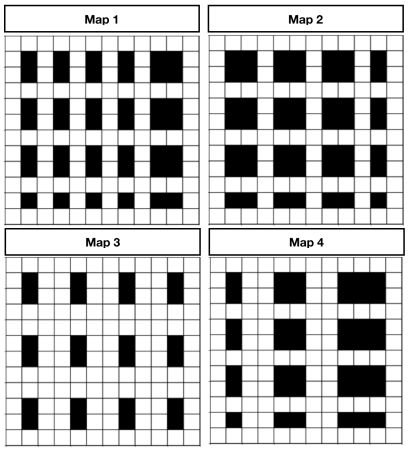

Fig. 2: Maps of the environment used to train SafeDQN. We used a single block easy (Map 1) and hard map (Map 2), and a double block easy (Map 3) and hard map (Map 4).

clinical teams treating low and high-acuity patients. Also, we generate blank videos (zero-padded) for empty locations on the map.

We expected certain maps would be more difficult for the agent to navigate than others (See Fig. 2). For instance, Map 3 has more space than the other maps, so it should be easier for the agent to find a path to its goal. On the other hand, Map 2 has less space than Maps 3 and 4 and fewer hallways than Map 1, so it might be harder for an agent to find a path that does not interfere with a clinical team.

A. Dataset

For a robot to function within the ED, it needs an accurate representation of real-world observations. However, to the best of our knowledge, there are no existing video datasets with real-world data. Thus, we constructed our own dataset, composed of videos that reflect realistic conditions in an ED. The videos show day-to-day tasks of HCWs, and reflect how dynamic, chaotic, and crowded it can be (See Fig. 3). We used the following search terms on YouTube: "emergency medicine trauma," "gunshot wound trauma," "heart attack trauma," "traumatic injuries emergency department," "Untold Stories of the E.R.," "Trauma: Life in the E.R.," "Emergency Room: Life + Death at VGH," "Boston Med," "Hopkins," "Save my life," "NY Med," "The Critical Hour," "Boston EMS," "Big Medicine," and "medical documentaries."

After collecting videos, we reviewed them to ensure they contained clinical activities. We then removed image sequences that did not show patient treatment. Next, we split all videos to create fixed, uniform segments of 100 images each. Segments consisting of less than 100 images were adjusted to the appropriate length by applying circular padding. The final dataset contains 689 videos.

Each video was labeled with the maximum number of people along with two velocity measures, optical flow (OF) [34] and a keypoint detector (KD) [35] as these are commonly used methods in the literature. In order to generate a velocity measurement using the keypoint detector, we use the temporal difference of keypoints throughout the video.

Fig. 3: To train SafeDQN, we constructed a new dataset containing real world video from acute care settings that reflect realistic ED conditions, showing how dynamic, chaotic, and crowded it can be.

We calculated the number of people in each video using the pedestrian detector from [33], which we used to classify videos as low or high-acuity. 550 videos were labeled as low-acuity and 139 videos were labeled as high-acuity.

B. Training Procedure

We trained SafeDQN using a data generation procedure and training procedure, similar to [28]. In the data generation procedure, the agent explores the ED environment with constant $\epsilon = 0.1$ in which the agent takes a random action at probability ϵ and a greedy strategy at probability ϵ-1. This process generates a sequence of tuples (s_t, r_t, a_t, s_{t+1}) that are stored in replay memory M. In the training procedure, the agent samples from the replace buffer without replacement and trains the Q-Network to minimize the loss (See Section III). We randomly sampled 5 high-acuity and 5 low-acuity videos from our dataset and assigned them random locations on the map.

We trained SafeDQN on K80 GPUs using one of three optimizers, Stochastic Gradient Descent (SGD), Adam, and RMSProp, as done in [40], [23]. We conducted experiments with each optimizer and we report the results for SGD as this achieved the best performance. We trained SafeDQN for a maximum of 10k episodes or until convergence where we iteratively trained the NN using SGD with a momentum=0 and learning rate of 0.001 using a sample size of 32 with a mini-batch size of 1.

V. Evaluation

We seek to test the SafeDQN's ability to navigate among low- and high-acuity patients in our simulated environment. To facilitate this, we follow evaluation procedures from [41], [12] which employs similar methods such as random walk, greedy search, among others (described below) [42], [43].

- **Random Walk**: baseline method that employs a simple heuristic for navigation. The agent randomly selects an action until it reaches the goal.
- **A* Search**: a classic graph traversal method that uses a simple heuristic to generate the shortest path [38]. This heuristic considers the path that it has already traveled and an approximated distance to the goal in order to generate the shortest path.
- **Dijkstra**: a shortest path method that models the environment as an undirected graph. It aims to find a series of nodes with shortest distance [39].

Method	Map 1								Map 2							
	Avg. Path Length ↓		**Avg. Reward ↑**		**Avg. HA Penalties ↓**		**Avg. LA Penalties ↓**		**Avg. Path Length ↓**		**Avg. Reward ↑**		**Avg. HA Penalties ↓**		**Avg. LA Penalties ↓**	
	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD
Random Walk	243.6	231.0	-54.8	-53.3	5.9	5.6	15.4	25.0	231.6	240.6	-49.9	-54.5	3.1	10.9	11.3	11.3
A*	12.6	11.7	-2.4	-2.5	0.1	0.2	0.4	0.9	**11.2**	11.9	-2.1	-2.3	**0**	0	0.3	**0.1**
Dijkstra	11.6	10.4	-2.4	-2.1	0.1	0.3	**0.2**	**0.4**	12.0	12.0	-2.6	-2.4	0.1	0	**0.2**	0.4
SafeDQN	**11.3**	**9.4**	**-0.6**	**-0.6**	**0**	**0.1**	0.7	0.7	17.2	**10.6**	**-1.7**	**-0.5**	0.1	0	0.6	0.4
Method	Map 3								Map 4							
	Avg. Path Length ↓		**Avg. Reward ↑**		**Avg. HA Penalties ↓**		**Avg. LA Penalties ↓**		**Avg. Path Length ↓**		**Avg. Reward ↑**		**Avg. HA Penalties ↓**		**Avg. LA Penalties ↓**	
	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD	OF	KD
Random Walk	247.9	215.9	-54.2	-46.6	6.7	2.1	4.6	7.8	225.6	215.8	-50.3	-48.7	4.7	10.4	13.1	6.0
A*	10.9	11.7	-2.1	-2.3	0	0.1	0.1	0	11.6	**10.7**	-2.4	-2.4	0.1	1.0	1.1	0.2
Dijkstra	10.2	11.4	-2.1	-2.3	0	0.1	0.1	0	11.6	12.3	-2.3	-2.7	0.1	1.5	**0.5**	**0.1**
SafeDQN	**10.1**	**10.6**	**-0.5**	**-0.6**	0	**0**	0.1	1.0	**10.5**	11.5	**-0.6**	**-0.8**	**0**	**0.3**	1.1	1.4

TABLE I: Comparison between SafeDQN variants and Random Walk, A* search [38], and Dijkstra's [39] shortest path methods over 100 iterations where ↑ means higher is better and ↓ means lower is better. We compare the optical flow (OP) [34] and keypoint detection (KD) [35] methods which we used to determine the number of low and high acuity patients for each map. We ran all methods on four maps which have single and double block hallways (See Fig. 2).

We employ several metrics that are consistent with prior work to evaluate SafeDQN [41], [12]. To understand the overall performance of each method we report descriptive statistics for average path length, average cumulative reward achieved in each episode, and average high- and low-acuity penalties (See Fig. I). To determine which method performed the best, we focused our analysis on path length (PL), which indicates path efficiency, and high-acuity penalties (HAP), which indicate the safest path. We performed a three-way multivariate analysis of variance (MANOVA) test on HAP and PL. The independent variables we tested were navigation method (A*, Dijkstra, or SafeDQN), motion estimation method (optical flow or keypoint detector), and map (1-4) (See Table I).

Mauchly's test indicated that the assumption of sphericity had been violated for all main effects except for HAPs for map ($p > .05$) $\chi^2(2)$ = 10.3, PL for map ($p > .05$) $\chi^2(2)$ = 8.6, and PL for map*motion ($p > .05$) $\chi^2(2)$ = 1.4. As a result, we correct the degrees of freedom using Greenhouse–Geisser estimates of sphericity (ϵ = .96 for HAP for the main effect of map, ϵ = .97 for PL for the main effect of map, ϵ = 1.0 for PL for the main effect of map*motion) with Bonferroni adjustment. All effects are reported as significant at $p < .05$. Table I and Fig. 4 summarizes the results of the experiments.

1) Navigation Method: There was a significant main effect of the navigation method on HAP, F(1.15, 114.12) =605.17, $p < 0.001$, $r = 0.859$. Contrasts between navigation methods revealed that for HAP, SafeDQN performed significantly better than Random Walk, F(1,99) = 688.475, $p < 0.001$, $r = 0.874$; A*, F(1,99) = 9.032, $p < 0.005$, $r = 0.084$; and Dijkstra, F(1,99) = 15.954, $p < 0.001$, $r = 0.139$.

There was also a significant main effect of the navigation method on PL, F(1.0, 99.65) =2,697.030, $p < 0.001$, $r = 0.965$. Contrasts between navigation methods revealed that SafeDQN performed significantly better than Random Walk, F(1,99) = 2,739.96, $p < 0.001$, $r = 0.965$, A*, F(1,99) = 90.44, $p < 0.001$, $r = 0.477$, and Dijkstra, F(1,99) = 104.67, $p < 0.001$, $r = 0.514$.

2) Motion Estimation Method: There was a significant main effect of the motion estimation method on HAP, F(1, 99) = 42.15, $p < 0.001$, $r = 0.299$, but not for PL, F(1, 99) = 1.13, $p > 0.05$, $r = 0.011$. Contrasts revealed that for HAP, OF, F(1,99) = 42.15, $p < 0.001$, $r = 0.299$, performed better than KD.

3) Map: There was a significant main effect of the map on HAP, F(3, 276) = 16.45, $p < 0.001$, $r = 0.142$, but not for PL, F(3, 282) = 0.70, $p > 0.05$, $r = 0.007$. Contrasts revealed that for HAP, Map 1, F(1,99) = 17.60, $p < 0.001$, $r = 0.151$; Map 2, F(1,99) = 6.74, $p < 0.05$, $r = 0.064$; and Map 3, F(1,99) = 44.93, $p < 0.001$, $r = 0.312$, performed better than Map 4.

*4) Navigation Method * Motion Estimation Method*: There was a significant interaction between the navigation method and map in HAP, F(1, 116) = 17.98, $p < .001$, $r = .154$ and insignificant for PL, F(1, 100) = 2.3, $p > .05$, $r = .022$ (See Fig. 4). Contrasts were performed to compare all navigation methods to SafeDQN and all motion estimation methods which revealed SafeDQN performed significantly better than A* for HAP (F(1, 99) = 6.5, $r = .062$, $p < .05$) and Dijkstra for HAP (F(1, 99) = 12.9, $r = .115$, $p < .05$) for OF and KD. Contrasts also revealed that SafeDQN achieves a significantly lower PL than A* (F(1, 99) = 23.3, $r = .190$, $p < .001$) and Dijkstra (F(1, 99) = 20.0, $r = .168$, $p < .001$) for OF and KD.

*5) Navigation Method * Map*: There was a significant interaction between the navigation method and map for HAP, F(3, 320) = 7.7, $r < .072$, $p < .001$ and a insignificant interaction for PL F(2, 285) = 1.1, $p > .05$ $r < .011$. Contrasts were performed to compare all navigation methods to SafeDQN and all maps and showed a significant difference in all interactions except HAP for Random Walk compared to SafeDQN for Map 2 compared to Map 4 F(1, 99) = .35, $r = .004$, $p > .05$, PL for Random Walk compared to SafeDQN for Map 1 compared to Map 4 F(1, 99) = 6.5, $r = .062$, $p > .05$, and Map 2 compared to Map 4 F(1, 99) = .35, $r = .004$, $p > .05$, and Map 3 compared to Map 4 F(1, 99) = .876, $r = .009$, $p > .05$ Additional insignificant

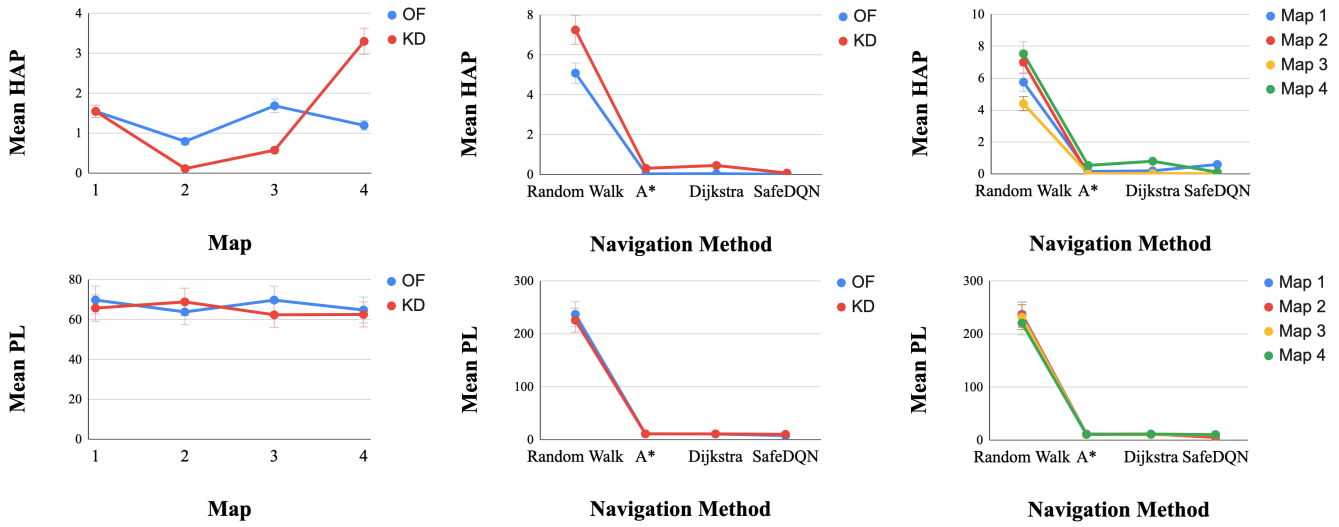

Fig. 4: Interactions between navigation methods for each map and compare the path length (PL) and high-acuity patients (HAP) for optical flow (OF) and keypoint detection (KD) methods. The y-axis shows the mean high and low acuity penalties.

interactions include PL for A* compared to SafeDQN for Map 3 compared to Map 4 F(1, 99) = 1.2, $r = .013$, $p > .05$, PL for Dijkstra compared to SafeDQN for Map 1 compared to Map 4 F(1, 99) = .82, $r = .008$, $p > .05$, and PL for Dijkstra compared to SafeDQN for Map 3 compared to Map 4 F(1, 99) = .61, $r = .006$, $p > .05$.

*6) Navigation Method * Motion Estimation Method * Map*: There was a significant three-way interaction for HAP F(3,314) = 38.2, $p < .001$, $r = .279$ and a insignificant interaction for PL, F(2, 290) = 38.2, $p > .05$, $r = .011$. This indicates navigation method and motion estimation had different effects on HAP and PL depending on the map being used. This is reflected in the interaction graphs in Fig. 4.

Overall, Random Walk had the worse performance of all methods $p < .001$. Table I shows that SafeDQN achieved the highest reward in all test cases.

VI. Discussion

We presented SafeDQN, one of the first acuity-aware navigation methods for mobile robots in the ED. Our results show that SafeDQN outperforms classic navigation methods in terms of avoiding high-acuity patients and generating the most efficient, safest path. Another key feature of our system is that it has an understanding of patient acuity inferred from live video, while others do not, making it well-suited for use by robots in the ED and other clinical environments.

Our results indicate that the motion estimation method used in our experiments had an impact on high-acuity and low-acuity penalties. This may be because the keypoint detector generated more noisy motion estimates than OF. Another potential cause of this is the threshold we applied to our dataset to generate the high and low acuity videos. The threshold needs to be tuned based on how frequently high-acuity patients visit the ED. For instance, if an ED frequently has many high-acuity patients, the robot may need to have a higher threshold for what it considers high-acuity, so it can still find a path while avoiding the most critical patients.

SafeDQN has several benefits for mobile robots working in safety-critical environments. It enables robots to visually infer patient acuity, so it will not disturb healthcare providers during critical care procedures. This may also be useful in other safety critical settings, such as disaster response. Furthermore, other researchers can easily build on our work to explore the benefits of other types of neural networks. For example, SafeDQN is easily expandable to other Deep Q-Learning approaches such as Double Q-Learning and Dueling Q-Networks.

In addition to the benefits of SafeDQN, it is also important to discuss its limitations. The primary limitation is that we evaluated SafeDQN in a simulation environment; real-world environments and mobile robot deployments will no doubt present many challenges. For example, as we have discussed previously [22], real-world perception is challenging for mobile robots in the ED due to occlusion, noise, and privacy concerns, making the use of visual sensors difficult. As SafeDQN relies on vision to determine patient acuity, this creates a question as to how well it will succeed in real-world settings. However, many recent privacy-preserving methods have been explored recently within the computer vision and ubiquitous sensing communities [44], [45], [46], suggesting a path forward for SafeDQN as well.

In the future, we plan to deploy SafeDQN on a physical robot in a medical simulation training center to evaluate its effectiveness in real-world settings. Our robot will deliver supplies to medical students as they treat patients during their clinical training sessions. Some areas of improvement that we plan to explore include ways to reduce training time as DQN methods are well known for being computationally expensive to train [26]. Also, we plan to explore how we might employ SafeDQN in a continuous 3D environment (e.g., [47]) for visual navigation in safety-critical settings. Finally, it would be exciting to explore SafeDQN within other safety-critical contexts, such as search and rescue.

REFERENCES

[1] L. D. Riek, "Healthcare robotics," *Communications of the ACM*, vol. 60, no. 11, pp. 68–78, 2017.

[2] A. Taylor, D. Chan, , and L. Riek, "Robot-centric perception of human groups," *Transactions on Human Robot Interaction*, vol. 9, no. 3, pp. 1–21, 2020.

[3] D. M. Wilkes, S. Franklin, E. Erdemir *et al.*, "Heterogeneous artificial agents for triage nurse assistance," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2010, pp. 130–137.

[4] M. Gombolay, S. Hayes *et al.*, "Robotic assistance in coordination of patient care," *Robotics, Science, and Systems (RSS)*, 2018.

[5] V. H. Wang and T. F. Osborne, "Social robots and other relational agents to improve patient care," *Using Technology to Improve Care of Older Adults*, p. 166, 2017.

[6] A. Thomaz and V. Chu, "Diligent robotics," 2019. [Online]. Available: https://diligentrobots.com/

[7] H. S. Lee and J. Kim, "Scenario-based assessment of user needs for point-of-care robots," *Healthcare informatics research*, vol. 24, no. 1, pp. 12–21, 2018.

[8] S. Jeong, D. E. Logan, M. S. Goodwin *et al.*, "A social robot to mitigate stress, anxiety, and pain in hospital pediatric care," in *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*, 2015.

[9] A. Taylor, H. R. Lee, A. Kubota, and L. D. Riek, "Coordinating clinical teams: Using robots to empower nurses to stop the line," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, p. 221, 2019.

[10] A. Taylor and L. D. Riek, "Robot perception of human groups in the real world: State of the art," in *2016 AAAI Fall Symposium*, 2016.

[11] S. Singh, M. Friesen, and R. McLeod, "Optimized path planning for a mobile real time location system (mrtls) in an emergency department," in *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)*. The Steering Committee of The World Congress in Computer Science, Computer, 2014, p. 1.

[12] A. Pokle, R. Martín-Martín, P. Goebel, V. Chow, H. M. Ewald, J. Yang, Z. Wang, A. Sadeghian, D. Sadigh, S. Savarese *et al.*, "Deep local trajectory replanning and control for robot navigation," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5815–5822.

[13] C. Mavrogiannis, A. M. Hutchinson, J. Macdonald, P. Alves-Oliveira, and R. A. Knepper, "Effects of distinct robot navigation strategies on human behavior in a crowded environment," in *2019 14th ACM/IEEE Intern. Conference on Human-Robot Interaction (HRI)*. IEEE, 2019.

[14] J. Rios-Martinez, A. Spalanzani, and C. Laugier, "From proxemics theory to socially-aware navigation: A survey," *Intern. Journal of Social Robotics*, vol. 7, no. 2, pp. 137–153, 2015.

[15] M. Vázquez, A. Steinfeld, and S. E. Hudson, "Parallel detection of conversational groups of free-standing people and tracking of their lower-body orientation," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, pp. 3010–3017.

[16] A. Nigam and L. D. Riek, "Social context perception for mobile robots," in *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2015, pp. 3621–3627.

[17] E. S. Short, A. Allevato, and A. L. Thomaz, "Sail: simulation-informed active in-the-wild learning," in *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2019.

[18] A. W. Ter Mors, C. Witteveen, J. Zutt, and F. A. Kuipers, "Context-aware route planning," in *German Conference on Multiagent System Technologies*. Springer, 2010, pp. 138–149.

[19] C.-Y. Wang and R.-H. Hwang, "Context-aware path planning in ubiquitous network," in *International Conference on Ubiquitous Intelligence and Computing*. Springer, 2009, pp. 54–67.

[20] S. Sehestedt, S. Kodagoda, and G. Dissanayake, "Robot path planning in a social context," in *2010 IEEE Conference on Robotics, Automation and Mechatronics*. IEEE, 2010, pp. 206–211.

[21] L. Piyathilaka and S. Kodagoda, "Affordance-map: A map for context-aware path planning," in *Australasian Conference on Robotics and Automation, ACRA*, 2014.

[22] A. Taylor, S. Matsumoto, and L. Riek, "Situating robots in the emergency department," *Spring Symposium on Applied AI in Healthcare: Safety, Community, and the Environment*, 2020.

[23] J. Kulhánek, E. Derner, T. de Bruin, and R. Babuška, "Vision-based navigation using deep reinforcement learning," in *2019 European Conference on Mobile Robots (ECMR)*. IEEE, 2019, pp. 1–8.

[24] H. V. Hasselt, "Double q-learning," in *Advances in neural information processing systems*, 2010, pp. 2613–2621.

[25] V. Mnih, A. P. Badia, M. Mirza *et al.*, "Asynchronous methods for deep reinforcement learning," in *Intern. conference on machine learning*, 2016, pp. 1928–1937.

[26] V. Mnih, K. Kavukcuoglu, D. Silver *et al.*, "Human-level control through deep reinforcement learning," *Nature*, 2015.

[27] L. Lin, "Reinforcement learning for robots using neural networks," Carnegie-Mellon Univ Pittsburgh PA School of Computer Science, Tech. Rep., 1993.

[28] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Robot gains social intelligence through multimodal deep reinforcement learning," in *2016 IEEE-RAS 16th Intern. Conference on Humanoid Robots (Humanoids)*. IEEE, 2016, pp. 745–751.

[29] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," *arXiv preprint arXiv:1511.06581*, 2015.

[30] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[31] L. Garrote, M. Torres, T. Barros, J. Perdiz, C. Premebida, and U. J. Nunes, "Mobile robot localization with reinforcement learning map update decision aided by an absolute indoor positioning system," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[32] H. Bae, G. Kim, J. Kim, D. Qian, and S. Lee, "Multi-robot path planning method using reinforcement learning," *Applied Sciences*, vol. 9, no. 15, p. 3057, 2019.

[33] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[34] S. S. Beauchemin and J. L. Barron, "The computation of optical flow," *ACM computing surveys (CSUR)*, vol. 27, no. 3, pp. 433–466, 1995.

[35] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[36] J. Ramos *et al.*, "Using tf-idf to determine word relevance in document queries," in *Proceedings of the first instructional conference on machine learning*, vol. 242. Piscataway, NJ, 2003, pp. 133–142.

[37] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[38] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.

[39] E. W. Dijkstra *et al.*, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[40] A. H. Qureshi, Y. Nakamura, Y. Yoshikawa, and H. Ishiguro, "Show, attend and interact: Perceivable human-robot social interaction through neural attention q-network," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 1639–1645.

[41] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi, "Target-driven visual navigation in indoor scenes using deep reinforcement learning," in *2017 IEEE intern. conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 3357–3364.

[42] A. Pfrunder, P. V. K. Borges, A. R. Romero, and A. Catt, G.and Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3d lidar," in *International Conference on Intelligent Robots and Systems (IROS)*. IEEE/RSJ, 2017, pp. 2601–2608.

[43] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum, "Look, listen, and act: Towards audio-visual embodied navigation," in *Proc. of the International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9701–9707.

[44] A. Haque, A. Milstein, and L. Fei-Fei, "Illuminating the dark spaces of healthcare with ambient intelligence," *Nature*, vol. 585, no. 7824, pp. 193–202, 2020.

[45] A. Kubota, T. Iqbal, J. A. Shah, and L. D. Riek, "Activity recognition in manufacturing: The roles of motion capture and sEMG+inertial wearables in detecting fine vs. gross motion," in *International Conference on Robotics and Automation*. IEEE, 2019.

[46] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[47] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik *et al.*, "Habitat: A platform for embodied ai research," in *Proc. of the International Conference on Computer Vision*. IEEE, 2019, pp. 9339–9347.