IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 June 2025, accepted 13 July 2025, date of publication 18 July 2025, date of current version 31 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3590587

SURVEY

Generative Models, Attention Mechanisms, and Adaptive Methods for Robot Navigation in Complex Environments–A Survey

HARINATH SRIDHARAN[1], NAMBALA RAMSAI[2], AND BHASKAR VUNDURTHY[3]
[1]Deloitte Consulting India Private Ltd., Bengaluru 560103, India
[2]Robotics Laboratory, Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600036, India
[3]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Corresponding author: Harinath Sridharan (harinath.sridharan@gmail.com)

ABSTRACT Autonomous mobile robots, equipped with multiple sensors, have been traditionally used to perform search, rescue and other tasks. A number of new scenarios for application of mobile robots have emerged in the last decade. These include automated logistics handling in warehouse-like environments (in the context of e-commerce) and robot-assisted personal care. In these scenarios, there is a need for highly accurate object recognition and semantic knowledge. Also, enhanced safety requirements have come up as robots attempt to interact more with humans and make efforts to recognize their gestures and movements. Mobile robots also increasingly operate in malls and other zones which are pedestrian-rich. Thus, a need has arisen for a relook at navigation strategies for mobile robots. Classical approaches are typically inadequate in these new settings. This survey is aimed at studying the role of contemporary approaches in artificial intelligence in enabling successful robotic navigation in a variety of complex environments. In particular, we discuss how generative models, attention mechanisms and adaptive methods have helped mobile robots navigate in cluttered, uneven and even unknown indoor and outdoor environments. We also point to several interesting possibilities in the future.

INDEX TERMS Mobile robot navigation, complex environments, generative adversarial networks, variational autoencoders, attention mechanisms, transformers, diffusion models, normalizing flow models, deep reinforcement learning, imitation learning, graph neural networks and knowledge graphs, current trends.

I. INTRODUCTION

Autonomous mobile robots operate in a variety of environments [1]. These include the factory floor where they are used for transporting materials and products. They have also been traditionally used for search and rescue tasks. Figure 1 presents a scenario for navigation of mobile robots. In particular, it depicts a factory environment where the robot navigates amidst a wide range of static objects and humans moving between different work stations. The task of the robot here is to assist humans in transporting items from one location to another in the factory floor.

The associate editor coordinating the review of this manuscript and approving it for publication was Yangmin Li.

Classical robotic navigation involves point to point transfer amidst obstacles. Collision avoidance is a key issue. Several geometric algorithms have been designed in the context of navigation [2]. They include model-based and sensor-based solutions. Various approaches have been adopted including visual odometry, feature matching, and stereo vision.

During the last decade, a number of new scenarios for application of mobile robots have emerged. These include warehouse environments [3] in the context of e-commerce, retail stores for surveying [4] and domestic environments [5] to assist the elderly. Robots often face enormous challenges in especially highly constrained spaces [6]. In these scenarios, there is also a need for highly accurate object recognition and semantic knowledge. Also, enhanced safety requirements have come up as robots attempt to interact more with

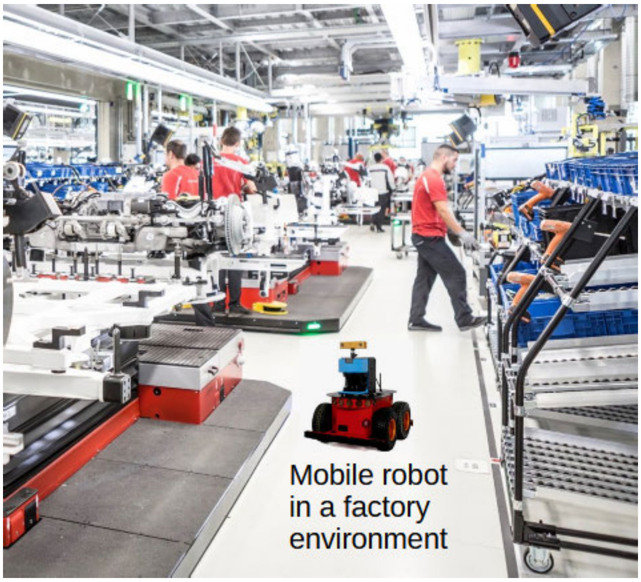

FIGURE 1. **Robot navigation in a factory setting.**

humans. Socially-aware navigation has become increasingly important. For example, robots are currently used in homes and other places such as exhibition halls to guide humans [7]. Robots are required to make efforts to recognize the gestures and movements of humans in these environments. Mobile robots also increasingly operate in malls and other zones which are pedestrian-rich. Thus, a need has arisen for a re-look at navigation strategies for mobile robots. Conventional navigation strategies are typically inadequate in these new settings and are now being largely replaced by methods that involve learning of the environment in which a robot navigates.

Different learning methods have been explored by researchers. Some involve teaching the robot to handle obstacles. Other scenarios that a robot may encounter in practice can also be provided as inputs to train a robot. Broadly, these fall into a data driven approach for navigation. During the period from (approximately) 2012 to 2019, considerable effort was on using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [8] for various robotic tasks. CNNs have been used for performing classification and detection tasks. RNNs have been particularly valuable for tasks such as change detection (such as recognizing the movement of furniture in a domestic setting) and shortest path calculation. However, CNNs as well as RNNs typically work well when a large amount of training data is available and it may be difficult to provide this for robotic navigation tasks.

Recently, *generative models* [9], [10], [11] have been proposed for various tasks in computer vision. These models are capable of generating new data from (limited amount of) input data. Further, while the performance of CNN is comparable to humans in object detection tasks (even with scaling), the intensity of the corresponding features in the feed-forward CNN layers is weak when we have partially or heavily occluded objects (compared to the completely visible objects) [12]. Recently, the notion of attention mechanisms [13] has been proposed for natural language processing tasks and variants of it have been developed for computer vision [14]. These handle non-ideal scenarios in object detection well and are valuable in the context of robot navigation seeking, for example crockery, in a domestic setting. It is worth noting that crockery/cutlery that is sought by the robot may be partially occluded hence methods that handle this occlusion well are valuable. A third category of methods, based on robots learning optimal behaviour by trial and error and adjusting actions dynamically, has also been highly effective in navigation. Examples include reinforcement learning and knowledge graph-based methods. These are broadly referred to as adaptive approaches.

We focus on study of works that use generative models, attention mechanisms and adaptive methods for robot navigation in complex environments in this paper. The complex environments we consider in this survey include the following: (i) cluttered environments (ii) unseen environments (iii) environments with rough terrain and (iv) environments with one or more dynamic obstacles. To our knowledge, prior reviews on robot navigation for such environments have not explicitly discussed the role of attention mechanisms, generative models and the various adaptive methods (besides reinforcement learning). The present survey is motivated by the following questions.

- What approaches have been used for handling complex environments prior to the advent of contemporary learning methods ?
- What are the capabilities achieved using *attention mechanisms* ?
- How well do various *generative models* perform ?
- Can *adaptive methods* be used in addition (to transformers or generative models) to enhance the performance ?
- What is the role of attention mechanisms and generative models to enhance the performance of an adaptive strategy ?

The remainder of this survey is organized as follows. In the next section, we present the basic terminology including tree diagrams showing different types of generative models, attention mechanisms and adaptive methods. We also indicate the methodology adopted for the survey. Section III describes classical works in the decade preceding the active introduction of deep learning methods. The role of early deep learning methods such as convolutional neural networks and recurrent neural networks in robot navigation is discussed in section IV. Section V discusses works in the domain of generative models. Attention mechanisms for robot navigation are discussed in section VI. This is followed by adaptive methods in section VII. A summary of key findings and potential areas for further exploration are presented in section VIII. Section IX concludes the survey.

II. TERMINOLOGY AND METHODOLOGY FOR THE SURVEY

The survey focusses on three contemporary AI-based approaches for robot navigation. In this section, a brief description of these three approaches, namely generative models, attention mechanisms and adaptive methods, is presented. Tree diagrams showing the various types of generative models, attention mechanisms and adaptive methods are also presented.

A. TERMINOLOGY

1) OVERVIEW OF GENERATIVE MODELS

Generative models are approaches that model the distribution of inputs as well as outputs [10]. It is possible to generate synthetic data points in the input space. In other words, they can generate new output samples that are different from those used during training of the model.

Generative models have been very effective at generation of images and text [9] for applications such image inpainting and text completion. Completion of text, for instance, is realized by the model with the knowledge of just the statistics of the language. In the context of robot navigation, generative models have been developed for various purposes. These include crowd motion prediction for safe robot navigation [15], footstep planning for humanoid robots [16] as well as generation and evaluation of multiple paths [17] for a mobile robot (and choosing ones that meet some constraints).

A tree diagram showing the various types of generative models used for robot navigation is presented in Figure 2.

2) OVERVIEW OF ATTENTION MECHANISMS

The origins of attention mechanisms can be traced to the work by Bahdanau et al. [18]. The authors in [18] proposed it as an extension to recurrent neural networks for machine translation. An attention mechanism can be thought of as an approach to access encoded inputs without having to compress the entire input into a single fixed-length representation [11]. Attention mechanisms allow a network to give different weights to different inputs [10]. Further, the weighting coefficients can themselves depend on the input values.

Attention mechanisms have been effectively applied to natural language processing applications [13]. The authors in [13] eliminate recurrence structures and show excellent performance. In the context of robot navigation, attention mechanisms have been used in various ways. For example, in off-road navigation and intelligent transportation, attention mechanisms have been valuable to segment the free-space and predicting the traversible paths effectively [19], [20]. They have also been useful for inferring target location in object-goal navigation.

Several types of attention mechanisms have been proposed. These include transformers, graph attention networks, group-wise attention mechanisms and attention score maps, A tree depicting the attention mechanisms used in robot navigation is shown in Figure 3.

3) OVERVIEW OF ADAPTIVE METHODS

Adaptive methods in machine learning comprise of those where the model parameters are not fixed. Instead, they get updated with time. A model's state at a given time is used to update the parameters. Adaptive methods have been studied from approximately mid-1970s.

A widely used adaptive method is reinforcement learning (RL) [21]. Its mathematical foundation is provided by Markov decision processes (MDPs). Merging basic reinforcement learning with deep learning algorithms has led to deep reinforcement learning [22], [23]. Recent surveys on the subject include [24], [25], and [26]. A few other adaptive methods have also been studied in the context of robot navigation. These include graph neural networks and knowledge graphs [9].

A tree diagram showing the various types of adaptive methods used in robot navigation in complex environments is presented in Figure 4.

B. REVIEW METHODOLOGY

This review examines the current state of mobile robot navigation with focus on complex environments of different kinds, viz. environments with dynamic obstacles, uneven terrain, occluded objects etc. The terrains include indoor as well as outdoor scenarios. Outdoor environments are not restricted to roads. Both wheeled as well as legged robots are considered. We also discuss works that employ a robotic manipulator on a mobile base. The articles collected were based primarily on search using Google Scholar, IEEE Xplore, and Semantic Scholar. The focus was on journals and top conferences. Keywords used for the search included "robot navigation", "complex environments", "artificial intelligence techniques". To narrow down the search, some additional keywords such as "generative models", "attention mechanisms" and "adaptive methods" were provided. Filtering of the results was done to focus on the work in the last five years. For classical methods (discussed in section III), the search period was primarily from 2000 to 2012.

III. CLASSICAL METHODS FOR ROBOT NAVIGATION IN COMPLEX ENVIRONMENTS

In this section, we discuss classical methods for robot navigation in unknown environments, dynamic environments and ones with certain constraints (such as dead ends). We restrict our attention to primarily the decade preceding 2012 when the Alexnet model was introduced [27]. Table 1 lists the various works. We will discuss these works briefly next.

A combination of model-based and reactive methods for autonomous navigation in office-like environments (with desks, tables etc.) is presented in [28]. A topological map based on sensory gradient is constructed and is combined with a reactive method that employs potential fields. The

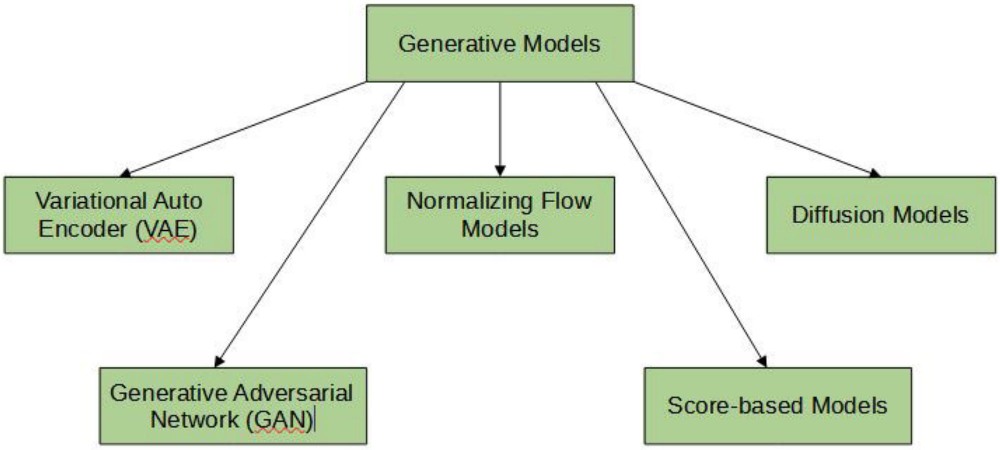

FIGURE 2. Various generative models.

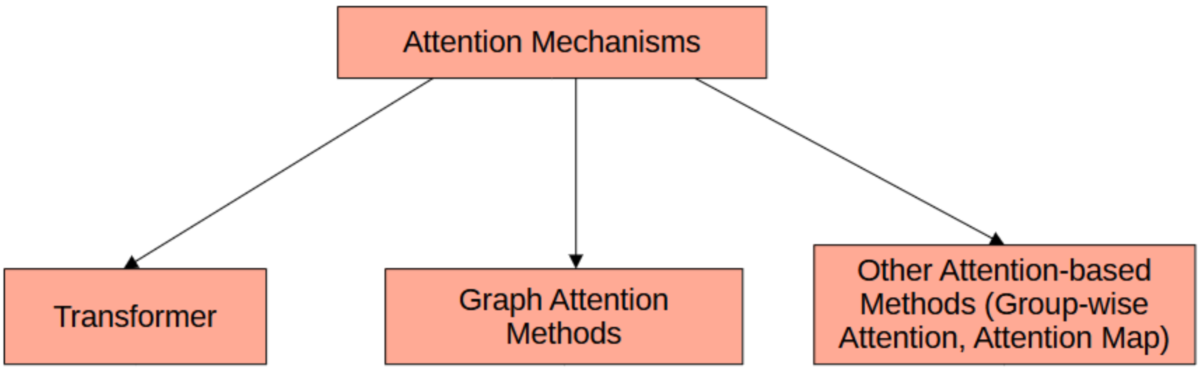

FIGURE 3. Various attention mechanisms.

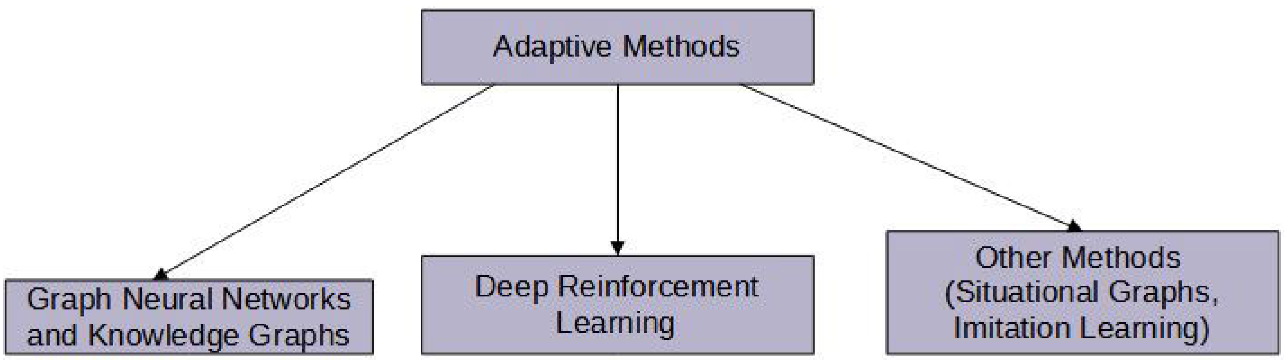

FIGURE 4. Various adaptive methods.

authors describe experiments in a static environment. A fuzzy logic-based approach to robot navigation applicable to a rough outdoor terrain is described in [29]. On-board terrain sensing and traversability analysis are reported in [29] and the authors indicate that incorporation of a global map-based path planner would be valuable to enhance their approach.

A method for navigation in an unknown indoor environment or a partially known environment is presented in [30]. The approach is based on fuzzy inference and two types of obstacle avoidance behaviours, one for the convex case and the other for the non-convex case, are reported. Experiments in a polygonal environment are reported. Navigation and mapping in a large unstructured environment are studied in [31]. A hybrid metric map combining feature maps and other metric representations is proposed in [31]. A consistency investigation of Simultaneous Localization And Mapping (SLAM) is also presented. Robustness to localization and operation in crowded environments are potential areas for further investigations.

Traditional navigation algorithms focus on transferring a robot to a predefined destination avoiding obstacles. However, in domains such as personal care, there is a need to consider human interaction during navigation. With this in view, the authors in [32] report the development of dynamical systems-based approach for indoor navigation. The approach relies on a topological map and attempts to have the robot maintain a formation with humans. Enhancing the complexity of formations and robot engagement in conversations would be interesting further steps. Another map-based approach for navigation is described in [33]. The authors develop a traversability region model keeping in view laboratory-type environments. Extensions to outdoor scenarios would be valuable.

An early effort on studying the development of learning algorithms for navigation is presented in [34]. Global strategy entropy is used as a measure for evaluating progress of learning. Simulation studies are reported. Another learning-based strategy for navigation especially in an unknown environment is reported in [35]. The authors propose a neurofuzzy-based approach coordinating sensor information and robot motion.

An intelligent navigation method for service robots is presented in [36]. The focus is on indoor smart environments with similarity in patterns. Environments with multiple obstacles and increased sophistication would be interesting to explore. Navigation in highly cluttered environments is studied in [37]. Performance-based fuzzy behaviors are defined and investigated and the work is a step towards navigation in unknown dense environments. Another fuzzy logic-based strategy is reported in [38] for an indoor environment with dead ends. A memory grid is defined and a minimum-risk method is proposed. Extension to an outdoor setting would be interesting. An optimal control-based strategy for navigation of a mobile robot is described in [39]. The authors present simulation results and point to the possibility of real-time implementation.

A visually complex environment with limited field of view is studied in [40] in the context of robot navigation. Bidirectional information exchange between robot and human is investigated. Simultaneous localization and mapping handling 3D navigation of humanoids (to handle uneven surfaces like steps for instance) would increase the range of applications. The authors in [41] present a fuzzy behaviour-based architecture for navigation in an unknown indoor environment. Optimizing the fuzzy behaviours and enhancing behaviour integration would be valuable extensions. Another fuzzy logic-based approach for an environment with complex traps in described in [42]. The authors compare Bayesian inference with their fuzzy logic approach and show the advantages with respect to speed and handling imprecision using the latter. Extension to outdoor environments would be valuable. An approach based on generating 2D occupancy grid maps from 3D point clouds is presented for outdoor environments with varying illumination in [43]. The challenges here rest on getting accurate information which is difficult when there are dynamic obstacles. Further, memory requirements are fairly high. A human-centered sensitive navigation method is proposed in [44] wherein a robot is intended to harmoniously coexist with humans. Increasing the accuracy of human tracking would be a valuable enhancement to the approach in [44]. A reinforcement learning-based strategy is presented in [45] for robot navigation in a dynamic environment. The authors in [45] focus on reducing the size of the Q-table to get a low-complexity realization. Incorporating pedestrian behaviours and handling crowded environments would be valuable extensions to the work in [45].

A fuzzy logic-based strategy for navigation avoiding local traps is reported in [46]. The authors focus on a static environment. A Voronoi diagram-based approach for navigation in a 3D unknown indoor environment is presented in [47]. Extensions to outdoor environments would be of interest. A spiral maze-like environment is considered for robot navigation in [48] and a neural network-based strategy is described. Accuracy enhancement of the approach in [48] would be a valuable further step.

To summarize, we note that several classical works that have handled complex environments are based on fuzzy logic. Some works use a simple neural network (with small number of hidden layers) to tune parameters of membership functions that are part of the fuzzy logic strategy. Highly crowded outdoor environments have not been investigated in detail. Further, there is a need to devise methods for navigation that explicitly consider safety aspects while humans are also part of the environment. In addition, applications that call for recognizing an object automatically with high accuracy (in the context of social robotics or logistics handling) have not been considered.

IV. EARLY DEEP LEARNING-BASED APPROACHES FOR ROBOT NAVIGATION

In the period from approximately 2012 to 2019, research on deep learning for robot navigation was predominantly using Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). We briefly summarize the efforts here. For additional information on early works on deep learning and reinforcement learning for robotics in general,

TABLE 1. Overview of classical robot navigation approaches in complex environments.

Reference	Environment	Approach	Contributions	Potential Enhancement
Maravall et al. [28]	Office-like environments with desks, tables, doors	Hybrid navigation	Reactive navigation algorithm and automatic generation of topological maps	Strategies that incorporate safety considerations
Seraji et al. [29]	Rough outdoor terrain near river bed and containing rocks	Fuzzy logic and traversability index	Behaviour-based navigation scheme with on-board terrain sensing and traversability analysis	Incorporation of global map-based path planner
Maaref et al. [30]	Unknown indoor environment or partially known environment	Fuzzy inference	Local method for unknown environment, fusion of local and global methods for partially known environment	Handling non-polygonal and dynamic environments
Guivant et al. [31]	Large unstructured outdoor environments	Fusion of feature and grid maps	Design of a hybrid metric map, development of a path planning technique and investigation of consistency of SLAM	Increasing speed and accuracy of SLAM, enhancing robustness of localization
Althaus et al. [32]	Indoor with interacting humans	Dynamical systems-based making use of a topological map	Design of a state diagram and control for maintaining a formation with humans	More sophisticated formations as well as engaging in conversations
Castejon et al. [33]	Cluttered indoor environments such as laboratories	Voronoi-like map based	Traversibility region model development using 3D laser scanner data	Handling uneven terrains and outdoor environments
Zhuang [34]	Dynamic environments and environments with multiple obstacles	Reinforcement learning	Analysis of strategy entropy and development of learning algorithm with self-adaptive learning rate	Generalization of strategy entropy from discrete state space to continuous state space
Zhu et al. [35]	Unknown indoor environments and dynamic environments	Neurofuzzy-based approach	Development of linguistic fuzzy rules and a learning algorithm to tune parameters of membership functions	Schemes for navigation in environments with lighting constraints
Park et al. [36]	Indoor smart environments (with similar patterns)	Semantic map-based	Localization and path planning using semantic map	Environments with multiple obstacles
Selekwa et al. [37]	Highly cluttered environments	Preference-based fuzzy behaviour	Design of fuzzy behaviour system using multivalued logic and application to navigation	Dynamic environments
Wang et al. [38]	Unknown environment with dead ends	Fuzzy logic-based	Development of a grid-based map termed memory grid and a minimum risk method for behaviour-based navigation	Long distance navigation and outdoor environments
Kokosy et al. [39]	Unknown indoor environment	Nonlinear dynamic programming	Path planning with nonholonomic constraints via optimal control formulation	Realtime implementation and testing on unicycle type robots
Carff et al. [40]	Visually complex environments with limited field of view	Bidirectional information exchange between robot and human	Design of a human-robot team navigation system with mixed reality displays and virtual viewpoints	Simultaneous localization and mapping handling 3D navigation (of humanoids)
Qing-yong et al. [41]	Unknown indoor environment	Fuzzy behaviour based architecture	Design of four basic behaviors and integration via a behaviour controller	Optimizing fuzzy behaviors and improving behaviour integration
Ayari et al. [42]	Dynamic and uncertain environment with complex traps	3D fuzzy logic	Fuzzy rules for linear and angular velocity and experiments with limited sensory support	Outdoor environments and solutions that consider safety aspects
Irie et al. [43]	Outdoor environment with varying illumination	2D Occupancy grid from 3D point clouds	Robot pose estimation on grid maps	Reducing memory requirement and handling dynamic objects
Lam et al. [44]	Environment with humans and other robots	Human-centred sensitive navigation	Development of harmonious rules and generation of sensitive fields	Increasing accuracy of human tracking
Jaradat et al. [45]	Dynamic environment	Reinforcement learning	Modified state space to reduce size of Q-table	Incorporating pedestrian behaviours and handling crowded environment
Qian et al. [46]	Environment with local traps	Fuzzy logic	Fusing obstacle avoidance and wall following behaviours	Dynamic environments
Ren et al. [47]	3D Unknown indoor environment	Voronoi diagram	Hybrid navigation strategy using deliberative planning and local optimization	Outdoor environments
Zhao et al. [48]	Spiral maze-like environment	Neural network	Single hidden layer network with back propagation	Accuracy enhancement

TABLE 2. Navigation Task versus Type of Neural Network; CNN denotes convolutional neural network while RNN denotes recurrent neural network.

Task \ Type of Network	CNN	RNN
Vision-based navigation by wheeled robot	[52], [53]	
Quadruped Path Control	[61]	
Shortest path planning for wheeled robot		[63]
Path planning of Centaur-like robot	[60]	
Robot relocalization for wheeled robot	[54]	
Heuristic map learning for walking robot	[59]	
Change detection for wheeled robot		[5]
Dynamic path planning for wheeled robot		[62]
Object Tracking for humanoid	[65]	
Terrain classification by wheeled robot	[55]	[64]
Facade cleaning by wall-climbing robot	[58]	
Target following by wheeled robot	[66], [67]	
Weed management by agricultural robot	[57], [56], [68]	

we refer the reader to the extensive surveys in [49], [50], and [51].

CNNs have been extensively used for image-based robot navigation approaches. These include vision-based navigation [52], [53], relocalization [54] and terrain classification [55] with wheeled robots. They have also been used in agricultural robotics for weed management [56], [57]. Applications for legged robots with CNNs include facade cleaning [58], heuristic map-learning [59] and path planning of Centaur-like robot [60]. Path control of a quadruped via accurate classification of furniture in the environment has also been accomplished using CNNs in [61].

The authors in [5] consider the task of automated change detection in a domestic environment and point to the use of a recurrent neural network for effective navigation when the environment involves furniture or other gadgets with changes in position. A gated recurrent unit-recurrent neural network is proposed for dynamic path planning of a mobile robot in [62]. The authors in [63] present the use of a recurrent neural network for shortest path planning. Another scenario where a time-based approach such as RNN is useful (and outperforms other approaches such as support vector machines with respect to accuracy) is for vibration-based ground classification [64] that is valuable for mobile robot navigation.

Figure 5 illustrates an application of robot navigation arising in weed detection. Conventional approaches to handle this problem require not only color image data but also additional cues such as near infra-red information [56]. With deep networks *high accuracy* is obtained merely with RGB data. Relevant works based on CNNs in this direction include [56], [69]. Contemporary approaches for this task are via transformers [70], and offer enhanced performance due to a notion called self-attention.

As observed from Table 2, CNN and RNN have been used for navigation of wheeled and legged robots in various settings. As indicated earlier, CNN and RNN-based methods require fairly large amount of training data which may often be difficult to provide for robot navigation. Further, CNN does not extract features when there are occlusions [12] and this can present challenges in social robotics settings when a mobile robot is tasked with detecting crockery/cutlery to help adults in a domestic setting. Generative models are capable of creating new data from existing ones and this is valuable in path finding applications. In particular, generative approaches such as diffusion models enable generation of multiple paths for a robot from a given starting point allowing the robot to evaluate each and choose the best for navigation (additional details are provided in section V-E). Further attention mechanisms such as transformers have the advantage that transfer learning is highly effective [10] and this allows use (in robotics) of datasets created for other domains.

Pure CNN and RNN-based approaches have now been largely superseded by generative models, attention mechanisms and adaptive methods for robot navigation. However, it is worth noting that CNN and variants of RNN (such as Long Short Term Memory (LSTM) network) are still used as a component of some generative models for navigation. Further, they augment attention mechanisms to obtain enhanced performance in certain scenarios. In the sections to follow, we examine how the current landscape is in the context of learning for robot navigation.

V. GENERATIVE MODELS

One of the powerful tools of machine learning frameworks is generative models, which generate new and realistic data from the existing data. They excel in understanding complex patterns or complex data. In the context of navigation, the generative models can generate trajectories for path planning, predict the next states (for instance, human movement during robot navigation), complete a discontinuous navigation path, model occluded objects and so on.

The study of generative models is divided into five subsections. We start with Variational Auto Encoders (VAEs), which are simple and easily integrated into latent spaces. They use a probabilistic framework but produce blurry images. Then we explore Generative Adversarial Networks (GANs), which generate high-quality, realistic data with the help of adversarial training. GANs do not have any likelihood function, sometimes leading to unstable training. To

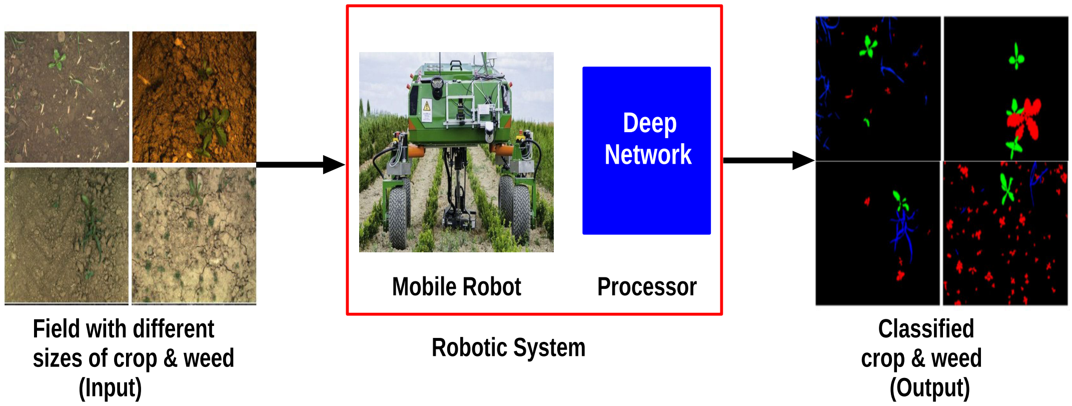

FIGURE 5. Classification of crop and weed in agricultural robotics via deep learning;green indicates crop while red indicates weed.

overcome the disadvantages of VAEs and GANs, a normalization flow-based model is introduced where high-quality data or images are generated, exact likelihoods are computed, and stable training is done. In recent years, there has also been a new type of model named score-based models, and a special case of this is diffusion models, which have become quite popular as they show good performance while handling high-dimensional data. In the following subsections, we will explain these generative models and their application to robotics, particularly for navigation tasks.

A. VARIATIONAL AUTO ENCODER (VAE)

One of the early, more straightforward and foundational generative models is VAE, as they introduce latent variable modelling using a probabilistic framework, which is generally regarded as a good starting point for learning generative models. It was introduced by Kingma and Welling in 2013 [77]. By encoding high-dimensional data into probabilistic latent space, which uses mean and variance, the VAEs convert the input data into efficient low-dimensional data by enabling reconstruction and uncertainty estimation and using decision-making skills. VAEs depend on Stochastic Gradient Variational Bayes (SGVB) estimation, which allows them to learn a latent space representation across varied datasets efficiently. VAEs have various applications for robotic tasks such as localization, motion planning, navigation and scene understanding. VAE learns a probability distribution (mean and variance) and samples a latent vector from this distribution. Further, VAE optimizes a combination of reconstruction loss and Kullback-Leibler (KL) divergence to ensure the latent space follows a smooth distribution.

Figure 6 shows how a variational autoencoder can be used for robot navigation in complex environments. The robot utilizes the captured images and laser scans for map construction, self localization and path planning in its environment. The application of VAE makes the system adaptive to varying environmental conditions.

Self-localization is one of the key tasks in robotic navigation, and VAEs have been utilized in several works for this purpose. One such work is reported in [71] where VAE is used for designing a self-localization model, i.e. finding the position of an electric wheelchair robot. The robot is equipped with a depth camera, and the features extracted from these camera images are given to VAE. The robot is tested on pre-defined trajectories, and the Euclidian distance error obtained is in the range [0.03, 0.33 m] in simple environments, but it is quite high (1.52 m) when the environment is cluttered. One more variant of VAE, named laserVAE, is developed in [72]. The robot is able to perform global self-localization tasks both in indoor and outdoor environments. The robot is equipped with LiDAR, and its data is sent to VAE. It reconstructs the laser scan data, but the normal VAE produces smoothly interpolated data, which leads to loss of step-edges, which are critical in determining obstacles or surfaces. To overcome this problem, a step-edge classifier is applied at the decoder output so that the sharp transitions can be restored, thereby increasing model robustness even in dynamic environments with noise. This laserVAE has achieved a success rate in self-localization of 92.5% and 95.2% in indoor and outdoor environments respectively.

A new framework, named Dynamic AMorphous Obstacle Navigation (DAMON), is proposed in [73] for efficient robot motion planning. The high-dimension (11-D) sensory data, which includes obstacle position (3-D), robot joint angles (7-D) and status of collision(1-D), are given to the encoder, which consists of three fully connected (FC) layers. It converts to 2-D sampled latent state representation, which separates colliding states from non-colliding states. Then, using a decoder with three FC layers but in the reverse order, the encoder converts this vector into a 11D state vector. The K-Nearest Neighbors (KNN) algorithm is used to create a collision-free path in static environments, and in case of dynamic obstacles Gilbert-Johnson-Keerthi

TABLE 3. VAE.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environments tested	References
COOR-IMG VAE	Self-localization	Electric wheelchair robot	Robust against sensor noise	Limited generalization	Combining with path planning algorithms	Indoor	[71]
LaserVAE	Self-localization	Mercury	Stable performance across environments	Only static environment is assumed	Extending to 3D maps	Indoor and outdoor	[72]
DAMON	Trajectory planning	Franka Emika Panda	Adaptive dynamic obstacle avoidance	Requires very large dataset	Integrating with RL algorithms	Indoor (On table)	[73]
VAE	Navigation	Robot with camera	Creating path with image sequences	Discontinuous path generated	Combining VAE with GAN	Simulation (Video)	[74]
VAE-HA,HU, ND,NG	Personalized robot navigation	Kobuki Turtlebot 2	Safer and efficient navigation	Requires very high quality training data	Sensor fusion for better understanding of environment	Indoor	[75]
VAE,SAE, WAE	Autonomous navigation	ANYmal	WAE better classified the terrains	Limited to specific geographic regions	Extend to dynamic environment	Outdoor (Park, desert)	[76]

(GJK) algorithm and Gaussian Mixture Model (GMM) are used to predict its path, and adjust accordingly. DAMON achieves 98.6% and 97.8% success rates in the case of static and dynamic environments, which is 20% higher than the Rapidly-exploring Random Tree (RRT) method, and trajectory smoothness improves by 24% and latency improves by 85% compared to RRT.

The VAE is used in [74] for robot navigation in an indoor environment. A video of the entire indoor environment (here home) is taken and 1000 images are extracted from the video to train the VAE. The encoder, which has a Convolutional Neural Network (CNN), converts the images into 4-D vectors where each dimension corresponds to different transformations of the scene, thereby representing the camera movement, i.e. move forward or backwards, move left or right, tilt up and down and tilt left/right. The shortest path is computed in the latent space as initial and destination points are given to the encoder. The decoder creates the images from the path that the robot should follow to reach the destination. As the VAE is not so good at capturing local features, blurry images are obtained from the decoder, and the generated path is not continuous, leading to abrupt jumps between two points. This shortcoming can be overcome by using GANs along with VAE.

The VAE is used in [75] for personalized robot navigation tasks with the help of depth sensors. The unique characteristic of this robot is it follows the path preferred by the user rather than just optimal paths in both static and dynamic environments. The human operator controls the robot using a VR headset and a joystick, and the robot collects the data from the RGB-D camera, which is given to VAE to convert to low dimensional latent space. The authors propose a total of five variants of VAE. The first three of them are VAE-Human Aware (HA), VAE-Human Unaware (HU) and VAE-No Demonstration (ND). VAE-HA and VAE-HU are trained with human demonstrations, but only VAE-HA optimizes for human preference but later doesn't optimize, and VAE-ND is not at all trained with human demonstrations. The fourth variant is VAE-FOV-120, where the depth camera field of view(FOV) is 120° (rather than 90°), while the fifth variant named VAE-No Goal Distance (NG) is one where goal distance is not taken as input. All the five models are tested with Long Short Term Memory (LSTM)-Human Predictor (HP), which uses LSTM to model human movements and VAE for robot path planning. Overall, VAE-HA achieves a higher success rate, while LSTM-HP achieves a lesser collision rate.

The models with VAE have achieved good performance when the robot's navigation is on plain surfaces, as terrain variations are minimal when different terrains are used. VAEs capture brightness levels instead of rich features like textual or structural variations. So Wasserstein distance-based AutoEncoder (WAE) and Spherical constraint-based AutoEncoder (SAE) are used in [76] where the robot is tested on highly varied terrains in the Spanish Tabernas desert. VAE learns from a probabilistic distribution, but the latent space in WAE tries to minimize the Wasserstein distance between the learned latent distribution and a fixed prior distribution. Meanwhile, in SAE, the latent space is constrained to the unit sphere's surface, which makes embeddings lie on a hyperspherical manifold rather than being distributed freely in Euclidean space. WAE is the best model out of the three models as it has 14.7% higher differential entropy (which measures uncertainty/variations in the distribution), and SAE has 13% higher differential entropy when compared to VAE. Table 3 summarizes the work in robotic navigation using VAEs. We now discuss the role of generative adversarial networks.

B. GENERATIVE ADVERSARIAL NETWORK (GAN)

GANs were introduced in 2014 [84]. A new way of training called adversarial training become popular and it is used in

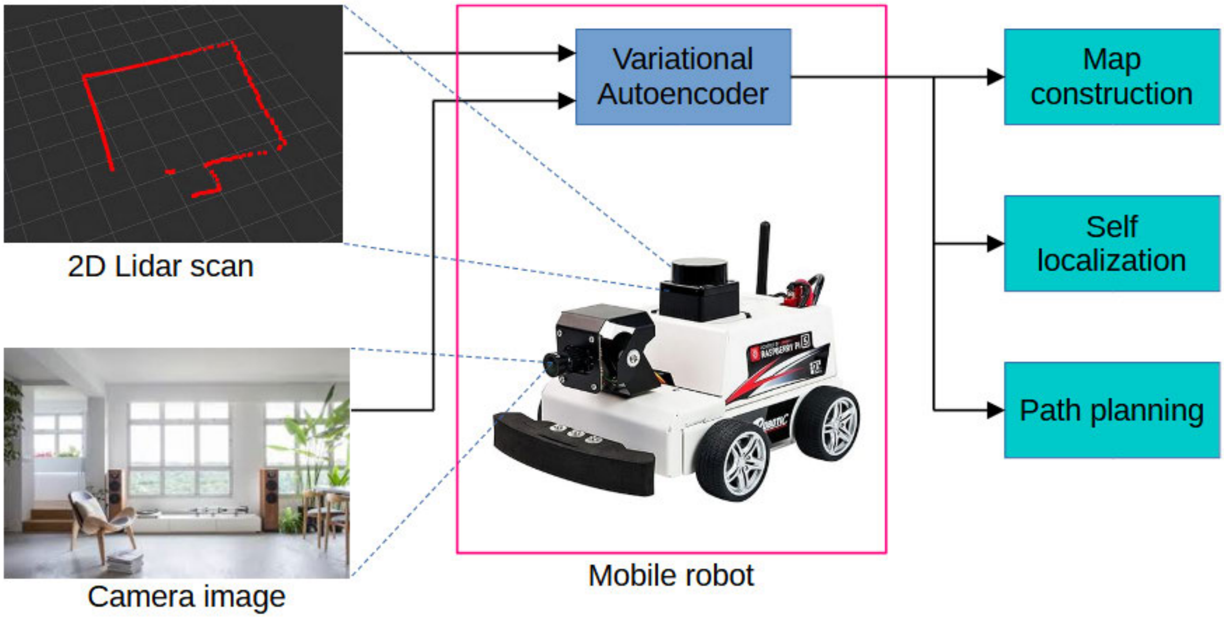

FIGURE 6. **Variational autoencoder for robot navigation.**

GAN. This approach helps to generate high-quality (new) data. It is a two-network model where the generator and discriminator compete with each other in a mini-max game. The generator tries to create new and realistic data while the discriminator classifies these data into either real or fake data. Any neural network model performance depends on the dataset and its training, but in a few applications such as medicine, autonomous driving, robot navigation and anomaly detection, the dataset availability is minimal. So GANs are used to increase the dataset, ensuring it is as realistic as possible.

With respect to navigation, GANs help create realistic maps, generating multiple possible path-planning trajectories that are feasible and efficient while considering the constraints, obstacles, and safety perspective. Research works on GAN include robot path planning and navigation in complex indoor environments [16], [78], dynamic environments [15], [79], [80] and perception and mapping-based navigation [82], [83].

The U-net-based generator of GAN in [78] generates multiple possible paths, and the discriminator acts as a binary classifier that chooses the best path generated. The model takes images as input and utilises local and global path planning to enable real-time feedback. It is tested in Gazebo simulation environment. The inference time is reduced by 73% and 83% in curved shorter (50m) and longer paths (450m), respectively, when compared to the A^* algorithm, and the path replanning time is also reduced significantly.

The authors in [16] use GAN for foot-step planning-based navigation for humanoid robots in complex indoor environments. The robot-captured images are given to a CNN-based GAN, similar to U-Net, with skip connections but with minor modifications. The generator outputs the path image, which is converted to a grid-based representation. If any dynamic object enters the path, the GAN is re-run to generate a new path image. The model is trained with $10K$ images for $50K$ epochs and achieves an accuracy of 93.6%. When tested on the Gazebo simulator, it took shorter planning time and resulted in shorter path length when compared to Dijkstra's and A^* algorithms.

Static Obstacles Probability Description (SOPD)-GAN is proposed in [79] for robot path-planning in dynamic environments. The robot is equipped with a LiDAR and point cloud data is obtained from it. With the help of Normal Distributions Transform (NDT) and Euclidean clustering algorithms, the objects in the environment are classified into static objects (such as pillars and poles) and dynamic objects (like people or pedestrians walking around). The SOPD module, which consists of an LSTM network, considering the static obstacles, gives a probability distribution of various paths for pedestrians, and the generator network of GAN produces realistic and collision-free trajectories. The robot is controlled using these trajectories with the help of the Improved Dynamic Window Approach (IDWA) algorithm.

A GAN network termed Social GAN which predicts human motion, is used along with real-time human motion prediction and trajectory planning for a robot in [80]. An RGB-D camera captures human movements, and based on past movements, the Social GAN predicts future human movements. The generator consists of an LSTM-based encoder-decoder network, and it is connected through a pooling module, that processes the interactions between

TABLE 4. GAN.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environments tested	References
CNN-based GAN	Path planning	Turtlebot4	Takes lower path planning time	Requires high-end computational resources(10 GPUs)	Deployment of model in real world	Gazebo simulator	[78]
CNN-based GAN	Foot step planning	Nao (Humanoid)	Finds shortest, smoother path	Not applicable to uneven terrains	Enabling modelling for vision based object detection	Gazebo simulator	[16]
SOPD-GAN	Autonomous navigation	Ackerman type robot	Smoother and safer navigation in dynamic environment	Global path planning is not considered	Optimizing prediction problem	Indoor (Corridor)	[79]
SocialGAN	Socially compliant navigation	Mobile service robot	Maintains better safer distance from humans	Adapting to highly unpredictable human movements	Multi-robot coordination	Gazebo simulator	[80]
GAN-MP	Navigation under localization attacks	Holonomic robot	Robust to different attacks	High computational load	Modelling hardware failures too.	Simulation	[81]
GAN-ATM	Navigation across terrains	UAV-UGV Collabrative system	Generates shorter trajectories	Requires different models for different terrains	Extending to varying outdoor terrains	Outdoor	[82]
CNN based GAN	Path planning	DJI Tello drone	Enhanced visualization of navigation with help of AR	Not generalized for unknown environments	Incorporate illumination changes in environment	Indoor	[83]

multiple persons, thereby avoiding unrealistic predictions. The robot pose is estimated with the help of data from LiDAR and IMUs. By using the Motion Prediction-Timed Elastic Band (MP-TEB) algorithm, the human trajectory predictions by Social GAN are integrated into TEB optimization, thereby producing efficient, smoother and collision-free robot trajectories. This is tested on a Gazebo simulator, and the proposed framework achieved reduced time for reaching the target, reduced path length and maintained higher safety distance from humans when compared to the TEB framework. Social-GAN is also tested on ballbot robot in [15] for human motion prediction. The social-GAN model has a 33% lesser Average Displacement Error when compared to the constant velocity model, leading to better prediction. But when this social-GAN is practically deployed for robot navigation control with a Model Predictive Controller (MPC), the navigation is practically not improved despite better human motion prediction. This is perhaps due to safety-efficiency issues or the prediction leading to lower optimization or insufficient practical data in the training dataset.

Most motion planning schemes involving robot localization (i.e., position estimation) assume they are foolproof or resilient to attacks. However, the robots are practically vulnerable if localization attacks happen, including Inertial Measurement Unit (IMU) injection, Global Positioning System (GPS) spoofing or LiDAR replay attacks. The authors in [81] proposed a GAN-based motion planning method resilient to these localization attacks. The GAN model takes the previous robot's position, velocity and attacked position and gives the corrected robot position. This position is given to the motion planning module, which consists of LSTM and a deep-Q learning-based model. When the system is attack-free, this model and the deep reinforcement learning (DRL) model have the same success rate of 98%, but when the system is under attack, the DRL model's success rate drops to 84%, but the proposed model has 97% success rate, demonstrating a high success rate amidst attacks.

GAN is also used in [82] for Active Terrain Mapping (ATM) for collaborative air-ground robotic systems. The system consists of an Autonomous Aerial Vehicle (AAV), which captures images and sends them to the ground station, which controls the Autonomous Ground Vehicle (AGV). A CNN is used to classify terrain into one of three categories: grass, pavement, or concrete. As the terrain data is limited, the three GAN models, namely GAN_{grass}, GAN_{pave} and $GAN_{concrete}$ augment the data, thereby increasing the dataset, which increases the CNN classifier accuracy thereby providing efficient path planning for UGV which uses Rapidly Exploring Random Tree (RRT) algorithm. The average path length used by this method is smaller than others, and a CNN accuracy of 90.35% is achieved.

GAN and multiple steps AI-driven pipeline are used in [83] for real-time robot navigation and augmented reality visualization. A Micro Aerial Vehicle (MAV) captures its surrounding images, and with the help of AirSim, a 3D virtual simulation is created. Here, the GAN is used for domain transfer between the real and virtual environments, i.e., mapping the images between them. The shortest path in a virtual environment is computed by the A^* algorithm, and it is further made efficient and faster with the help of 3D Navigation Meshes, which reduces the waypoints. Each waypoint is encoded with metadata like speed, time left and obstacle position. GAN transfers the data in a virtual environment to a Jetson Nano, which controls the MAV.

Table 4 captures the works on GANs. We now move on to normalizing flow models.

C. NORMALIZING FLOW (NF) MODELS

Normalizing Flows are also generative models but they can learn complex probability distributions by utilizing a series of invertible and differentiable transformations. The NF models produce the exact likelihood estimation, making these models useful in practical environments, more precisely, for probabilistic modelling tasks. Other generative models like VAEs will produce only the approximate likelihood, while GANs cannot compute likelihoods explicitly. Traditional SLAM algorithms approximate the uncertainties in localization problems with the help of Gaussian distributions. However, practical environments also contain non-Gaussian noise due to odometry slip/grip error modelling and multimodal data processing. So NF models play a crucial role in robot navigation where the exact likelihood of uncertainty in navigation tasks is computed, and the robot safety is ensured by avoiding those specific points in the path.

A hybrid approach using normalizing flows is proposed in [85] for robot navigation in a dynamic environment. A Mecanum rover with LiDAR is used, and pedestrians are detected using Distance Robust Spatial Attention and Auto-regressive Model (DR-SPAAM) [93]. Two navigation approaches are used. One is traditional learning based, i.e., uses a deep RL algorithm, and the other is based on the Optimal Reciprocal Collision Avoidance (ORCA) algorithm [94], which is a rule-based algorithm. The graph-NF model makes a decision on when to switch between these two algorithms. This model computes the likelihood of the present situation with training data; if it is above the threshold, deep RL is used, or else the ORCA algorithm is used. This switching ensures that there is no collision with pedestrians. When the model is tested in simulation, the proposed NF model has a 34% higher success rate, and when practically tested, the deep RL method has only a 16% success rate whereas the proposed NF-based switching model has an 84% success rate.

Anomaly detection approach is used in [86] for safer robot navigation in unknown outdoor environments. A quadrupedal robot named ANYmal is used and teleoperated over various terrains for data collection with its RGB-D camera. Initially, Resnet-18 [95] is used for feature extraction. The NF model used is real-valued Non-Volume Preserving (Real-NVP) to learn the terrain's probability distribution. The areas with anomalies are detected based on scores of logarithmic likelihood, and the robot should avoid these places. This Real-NVP-based model achieved an area under receiver operating characteristic curve (AUROC) of 0.85, which is 21% and 13% higher than using autoencoder and support vector data description (SVDD) based models, respectively.

Normalizing-flows-incremental-smoothing-and-mapping (NF-iSAM) is proposed in [88]. The authors model the SLAM problem as a factor graph (where landmarks and robot position are represented as nodes and robot sensor data are represented by edges). Initially, the robot trajectory is modelled to follow Gaussian distribution, and then a series of invertible transformations are applied so that it is converted into a more complex distribution that can capture real-world disturbances. Finally, posterior distribution, i.e. robot trajectory, is obtained, and a Bayes tree is used to update it from time to time while the robot is navigating. However, this model has computational challenges, and it is tested only on small datasets. So the inference process is further optimized in [89] by introducing lightweight NF models and also updating only the affected branches in the Bayes tree. It is worth noting that the model works on large datasets.

Traditional sampling-based motion planners (SBP) for robots like RRT^*, $Bi - RRT^*$, and $Informed - RRT^*$ have limited performance as they perform random sampling and also, there are chances of mode collapse or posterior collapse due to the high varying nature of configuration space and motion plan configurations. To overcome these challenges, PlannerFlows are introduced in [91], which uses NF to make the motion planner follow a conditional probability distribution. This improves the model's learning about the environment as high-quality samples are produced.

NF is integrated with soft actor critic (SAC), an RL algorithm, in [92] for controlling the robot for navigation tasks. With the help of NF, the problem of the model converging to sub-optimal local minima is avoided as it uses a multimodal policy instead of a unimodal Gaussian policy. As invertible transformations are applied because of NF, the model can learn and adapt to the environment so that exploration and learning efficiency are improved. The model is tested on six MuJoCo (Multi-Joint dynamics with Contact) and three PyBullet Roboschool tasks, which are simulation environments. This SAC-NF model uses fewer parameters and achieves higher performance (i.e. higher reward) when compared to the SAC model alone. We now move on to score-based models.

D. SCORE-BASED MODELS (SBM)

Unlike VAEs, which learn a direct relation or mapping between input and output or GANs, which perform adversarial training, SBMs estimate the probability density gradient, which is the data score. SBMs generate new data by filtering out noisy inputs via iterative denoising. This approach allows the SBMs to capture more complex features and model the uncertainty effectively. Hence, SBMs try to learn the score function, which is defined as the gradient of the logarithmic probability density of the data distribution. The newly generated data tries to move towards a higher probability region by learning this score function. As SBMs operate on probabilistic frameworks, they also provide

TABLE 5. Normalizing Flow (NF) and Score-based Models.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environments tested	References
Graph-NF	Navigation in dynamic environment	Mecanum rover	Improved reliability	Sensitive to sensor noise, perception errors	Adjusting switching between models online	Indoor	[85]
Real-NVP (Normalizing Flow)	Anomaly detection in robot navigation	ANYmal	Does not require manual labeling of training data	Only local navigation strategies considered	Integration with global path planning algorithms	Outdoor terrains (forest, farm, park).	[86]
Conditioned score model	Trajectory planning	Planar robot	Shorter and smoother trajectories	Requires high training data	Extending it to 3D navigation tasks	Simulation	[87]
NF-iSAM	SLAM for robot	Navigational robot	Reduced error(ATE) and failure rate	Computationally expensive	Can be extended to multi-agent SLAM	Simulation	[88], [89]
HAM-Nav (Score-based Model)	Navigation using hand drawn maps	Mobile robot	Good performance in unknown environment	Requires large scale models(GPT-4)	Extension to multi robot system	Indoor	[90]
Planner Flows	Motion planning	Jaco arm robot	Faster convergence, lower collision rate	Limited performance in highly unstructured environments	Enhancing robustness in partial known environments	Simulation (2D,4D,6D Configuration spaces)	[91]
SAC-NF	Navigation in complex environment	Walker2-2d, Antv2	Stable navigation, higher rewards	Higher training complexity	Evaluating in real world environment	Simulation (MuJoCo, PyBullet Roboschool)	[92]

uncertainty estimation inherently, making them suitable for safety applications such as path planning or navigation in dynamic environments. Some recent works on SBMs are discussed next.

A conditioned score based model is proposed in [87] for robot's collision free trajectory generation. The surrounding environment is represented as a black and white occupancy map, and initial and target positions are included in it. A signed distance function (SDF) loss is used to obtain the information of obstacles using a CNN-based encoder. The proposed model consists of a temporal U-Net, which is trained with denoising score matching (DSM), and the score function of all trajectories is computed. These trajectories are further smoothened by refining, i.e. by removing noisy or collision trajectories with the help of Langevin Dynamics and SDF values. Finally, the shortest feasible path is considered for robot navigation. The success rate of this model is 90%, i.e. the generated trajectories of this model are 90% collision-free trajectories.

A novel architecture of hand-drawn map navigation (HAM-Nav), which uses vision language models (VLM), is proposed in [90] for robot navigation in various diverse environments. The input to the model is a simple freehand sketch of a hand-drawn map where a robot path is drawn. This map is converted to a graph representation with the help of k-means clustering and various other networks, including Google Cloud Vision API, grounding-DINO [96] and Grounded-Segment Anything Model (G-SAM) [97]. The robot position is estimated using a localization engine module, which consists of selective visual association prompting (SVAP) and a GPT-4o VLM. This module uses score-based prompting and selects the most likely robot position. Then, the navigation planning engine (NPE) module is used to choose the actions of the robot, i.e. moving forward/left/right, which also uses score-based prompting. This model is practically tested on a jackal-wheeled robot, and it achieved a success rate of 78% with success weighted by path length (SPL) of 0.714. Table 5 captures the work on normalizing flows and score-based models. We now move on to diffusion models.

E. DIFFUSION MODELS

One of the most widely used generative models today is the diffusion model, which is a special case of score-based models. They are inspired by the principle of non-equilibrium thermodynamics. The model consists of neural network architectures like U-Net or transformer, and the features given as input to these models are extracted from encoders. The training of these models consists of two stages, i.e., forward and reverse processes. In the forward diffusion process, the noise, generally Gaussian noise, is iteratively added step by step over time. The reverse diffusion process, indicated by the name, tries to predict the noise in data and removes it from the data. This ability makes the diffusion model learn complex, multimodal data to generate well-organized/structured data like generating trajectories for robot motion planning. The main advantage of the diffusion model is that it generates diverse and high-quality data, i.e. high-quality, optimal and feasible collision-free trajectories while maintaining stability during the training. As previously stated, in SBM, the score function is the gradient of the logarithmic probability density; in diffusion models, the score function is calculated at each time step t, i.e., the gradient of the logarithmic probability density at each t. By further imposing conditions on the generative model, it can be used in challenging environments. Some of the recent works are described below in this subsection.

Figure 7 depicts robot navigation using the diffusion model-based approach. Diffusion methods are used to generate various paths for the robot to reach the destination. The best among these paths is then selected.

The authors in [98] used RGB images captured from the robot to generate trajectories with the help of a diffusion model and transformer encoder with a masked attention approach. This encoder allows the robot to switch its task to either navigation (if a target image is given) or exploration. This model achieves 98% and 90% success rates in exploration and navigation, respectively, but has limited performance when there are occlusions and has higher inference time. So the Husky robot in [99] is equipped with LiDAR and uses diffusion based trajectory generation (DTG) model. The LiDAR readings, velocity data from odometry and target information are given to the perception encoder. The obtained feature representation vectors are given to the diffusion model, which has Conditional RNN, where sequential data is processed, and waypoints are generated. The model is robust to occlusion and achieves higher real-time performance, i.e., 29% higher distance ratio and almost 50% reduction in inference time when compared to [98].

Similarly, Resnet-18 is used as an encoder in [100] and a denoising diffusion probabilistic model (DDPM) whose structure is similar to U-net (i.e. rather than convolutional layers, feature-wise linear modulation layers are used). This model achieves 87% success rate in navigation. The trajectories are generated 23 times faster than traditional A^* algorithm but only experience difficulties during extreme navigation cases, i.e., the path is either too short or too long.

One of the advantages of the diffusion model is that it can generate multiple possible trajectories which can be used for robot path planning. One such example is [101] where multiple possible motion trajectory paths are generated. This is a scenario where traditional Conditional Variational Autoencoders (CVAEs) may not perform well. The diffusion model produces samples iteratively and optimizes the trajectories, leading to a collision-free optimal path for the robot by denoising the sampled trajectories over multiple stages. Although the training requires more computational resources, the inference time is low and high-quality trajectories are generated. This is further enhanced by introducing a Conditional Diffusion Transformer in [102]. The proposed Navigation World Model (NWM) generates realistic robot movements based on the previous robot path and target position despite restrictions or constraints enforced on robot movements thereby preventing unnecessary deviations. Therefore, this leads to 40% reduction in Absolute Trajectory Error (ATE) and 52% reduction in Relative Pose Error (RPE) when compared to the General Navigation Model (GNM) in [107]. However, this model may experience difficulties with predicting dynamic obstacles such as pedestrians walking on the road.

One significant issue in robot navigation is the movement of the robot in dynamic environments where humans are also present. Some of the works include [17], [103], [104], [105]. A joint multi-agent interaction diffusion model (JMID) is proposed in [103] for human motion prediction and Bilevel Model Predictive Control (MPC) for robot path planning without human collisions. These are tested on a clearpath jackal robot in an indoor environment with up to three persons moving around, and it achieves faster response, i.e., the robot reaches the target quicker with at least 10-15% fewer deviations when compared to the other models like AgentFormer or Constant Velocity Guess models. A local diffusion model, i.e. denoising diffusion probabilistic model (DDPM), is used in [104] for obstacle avoidance navigation planning in various environments such as static, dynamic, zigzag and maze-like scenarios. This model achieves a very high success rate of 95% and 91% in static and maze-like scenarios, almost double that of the LSTM-GMM model. This model achieved high performance in local trajectory planning but encounters difficulties with long-term planning as it has limited observation horizons.

Linear Temporal Logic on Finite Traces (LTLf), a formal logic framework where constraints that are dependent on the time are used, along with diffusion model is developed in [17] for a quadruped robot. This approach achieves efficient and safe navigation in a constraint-driven environment where both static and temporal constraints are handled. The satisfaction rate (how good the generated trajectories are following the given constraints) using this model is 35% higher when compared to using the diffusion model alone. Global information-guided conditional diffusion model (GI-CDM) is proposed in [105], which improves state action trajectories, i.e., improves off-line reinforcement learning used for robot navigation. This global diffusion model is best used for long-range navigation applications. Initially, the observational encoder extracts the features from the dataset, which contains robot sensor data (i.e. observations), robot actions and global information. A U-net-styled diffusion model is used, and a noise scheduler is used strategically to add or remove the noise during the model training. The generated trajectories from the diffusion model are improved using a decision transformer and refined using reinforcement learning. This model has shown 30% and 2% higher success rates in navigation in static and dynamic environments, respectively when compared to the Gaussian Mixture Model (GMM).

There are also some efforts that go beyond a single robot. One example is the multi-robot multi-model planning diffusion (MMD) model proposed in [106]. Initially, all the robots are fed with maps, input, and target positions, and each robot uses an identical but separate diffusion model

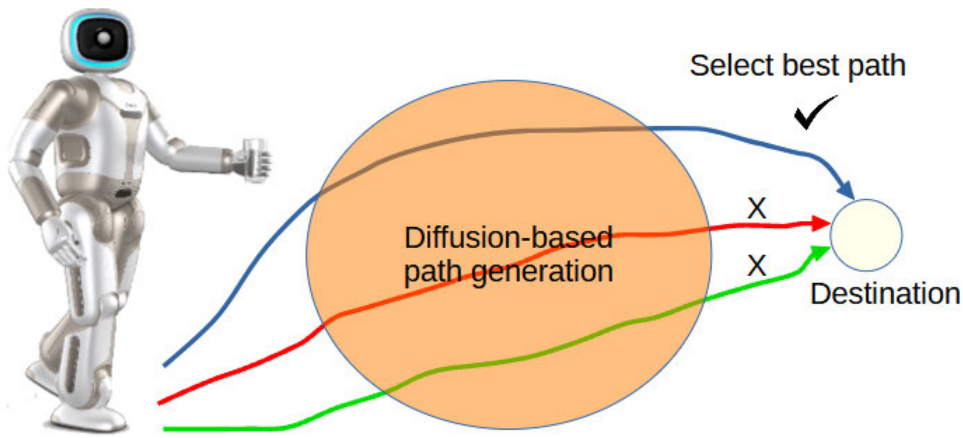

FIGURE 7. Illustration for robot navigation via a Diffusion-based approach.

TABLE 6. Diffusion models.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environments tested	References
NoMaD	Navigation in unknown environments	LoCoBot	Lower collision rate with fewer parameters	The goal can only be in form of an image	Can also include other input modalities	Indoor, outdoor	[98]
DTG	Mapless navigation	Husky robot	Higher performance, lower inference time than NoMaD	Adding confidence based trajectory selection	Extending to dynamic obstacles	Outdoor	[99]
DDPM	Autonomous navigation	Quadrupedal robot (boston dynamics spot, unitree go1)	Faster generation of paths	Limited performance in very smaller, longer trajectories	Further optimization of model to make light weight model	Indoor	[100]
Temporal U-net based model	Motion planning, manipulation tasks	Franka emika panda robot	Better motion planning than CVAE	Higher inference time	Exploring faster sampling techniques	Indoor (Shelf)	[101]
NWM	Visual goal directed navigation task	Clearpath jackal	Better performance even in unknown environments	Mode collapse in unknown environments	To better capture dynamic obstacles	Real world video datasets	[102]
JMID	Safe navigation of robot along with moving humans	Clearpath jackal	The human deviations are reduced.	Computationally expensive	Extending to unstructured environments	Indoor	[103]
LDP	Autonomous navigation	Ackermann steering robot	Lower collision rate	Slower in sampling process	Optimizing model for lowering inference time	Indoor	[104]
LTLDoG	Navigation under temporal constraints	Quadrupedal (unitree go2 Edu)	Well operates even with unseen ltlf constraints	Requires large datasets for training	Integration with interpolant based methods	Indoor	[17]
GI-CDM	Navigation for service robots	Agilex Hunter 2.0	High quality of trajectory generation	Limited to structured environments	Combining the model with local planners	Indoor	[105]
MMD	Generating motion trajectories in shared map	Multiple holonomic ground robots	Higher scalability (upto 40 robots)	Limited to fixed duration trajectories	Extension to perform collaborative tasks	Simulated ware house	[106]

that produces a trajectory for each robot. Then, a search algorithm named extended and Enhanced Conflict-Based Search (xECBS) algorithm is used to ensure a collision-free path is designed and scaled up to 40 robots. It performs better in comparison to other approaches such as like Motion-Planning-Diffusion-Composite (MPD-Composite) which do not work well beyond six robots. The approach in [106] is further improved by using a sequencing model for long trajectories. This approach generates continuous and dynamic trajectories rather than discrete grid-based approaches. This can be used in warehouse automation as well as search and rescue applications.

Table 6 presents a summary of the works on diffusion models. We next move on to discuss about attention mechanisms for robot navigation.

VI. ATTENTION MECHANISMS FOR ROBOT NAVIGATION IN COMPLEX ENVIRONMENTS

Recently, attention-based models have garnered attention for robot navigation in a variety of complex environments. As indicated earlier, when the complexity of the environment increases (for example, a cluttered environment or dynamic environment with people moving or other robots also moving), and when semantic goals are given, the classical algorithms often do not perform well. In this context, attention mechanisms [13], [18] have become a valuable tool. The attention-based models have been used in various navigation tasks, which mainly include audio or vision-based navigation, object goal navigation, autonomous exploration, and socially aware navigation, i.e., navigation when humans are moving.

The attention mechanism allows the robot to focus on relevant information like semantic, spatial or temporal information in the high dimensional input data. This allows the robots to prioritize certain things over others (for instance, human movement in the scene). The models using attention mechanisms are generally better generalizable to unseen environments or new targets as they change or adapt their focus to new environments dynamically. Although attention mechanisms have few limitations like requirement of high computational cost and requirement of large datasets, they are widely used for navigation tasks as they offer higher capabilities than traditional methods especially in complex environments.

This section focuses on recent developments in applications of attention mechanisms in robotic navigation, which are primarily categorized into three sub-sections: transformer-based models, graph attention-based approaches and a few other attention-based methods.

A. TRANSFORMER-BASED METHODS

Although transformers were originally developed for natural language processing (NLP) tasks since they are good at capturing long-term dependencies, they are also widely used in robotic vision and navigation applications. Unlike CNNs or RNNs and their variants, which process the data hierarchically or sequentially, transformers can directly model the relationship between any two elements in the input data regardless of the distance between them. Transformers employ a self-attention mechanism. This allows the transformer to capture effectively global spatial and temporal dependencies across input features making them very suitable for complex navigation tasks. It is worth noting that complex navigation tasks require the visual perception of identifying target objects or obstacles along with long-term planning including path planning. For example, for vision-based navigation tasks, the vision transformer can be used for enhanced path planning by integrating the transformer (for modelling the environment) with classical path planning techniques. The transformers described in this section include spatio-temporal transformer [108], [109], [110], Vision transformer [14], [111] and Crossmap transformer(CMT) [112]. Although transformers require high computational cost (as self-attention scales quadratically with sequence length), there are recent efforts to develop lightweight transformer architectures for deployment on resource-constrained edge devices.

Figure 8 depicts robot navigation using transformers. A robot with a transformer implementation is capable of segmenting the wild road in the outdoor environment image and predict a path in it.

He et al. [108] proposed a spatio-temporal transformer-based policy with an optimization algorithm to capture human-robot interaction in the context of social robot navigation. A spatio-temporal graph is constructed where robots and humans are represented by nodes (also called agents), spatial edges correspond to the relationship between agents at any point in time while temporal edges represent the movement of agents over time. Three RNNs separately process the information of nodes, spatial edges, and temporal edges. The spatial and temporal features from the RNNs are fused using a gated embedding mechanism and given to a multihead attention transformer, followed by a soft actor-critic reinforcement learning (RL) algorithm to optimize the navigation policy. This framework achieves 99.2% success rate with the average time taken by the robot being 10.1 seconds for reaching the target in the presence of five humans.

A hybrid spatio-temporal graph transformer is designed in [109] for socially aware navigation, i.e. robot navigation in the presence of moving humans. The robot is equipped with an RGB-D camera, and humans are detected with the help of YOLO [113] and the DeepSort algorithm. A spatio-temporal graph similar to the one in [108] is constructed. The spatial transformer, which consists of multi-head attention and a graph convolution network (GCN), creates a spatial attention map that learns the spatial relationship between agents. The temporal transformer, which has multi-head attention, creates a temporal attention map that captures the agent's trajectories or path over time. A multi-modal transformer, which contains a self-attention transformer and a cross-modal transformer, fuses spatial and temporal attention maps to capture human behaviour. Finally, a fully connected layer acts as a decoder, and the robot selects the action based on the soft actor critic (SAC) RL algorithm. The authors introduced a new social score metric, which includes navigation time, distance

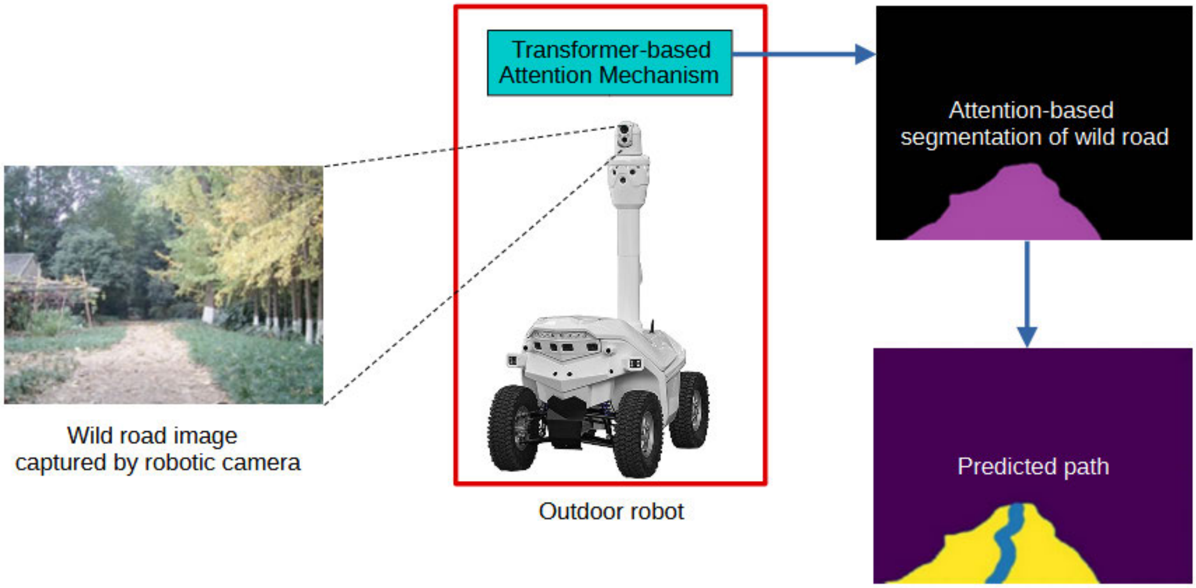

FIGURE 8. Transformer-based approach for robot navigation.

violations and failure rate. This spatio-temporal graph transformer achieved a social score of 95.8, whereas Optimal Reciprocal Velocity Obstacles (ORCA) and structural-RNN have 22.6 and 81.2, respectively, when the experiment is conducted in the presence of 20 humans.

Another spatial-temporal transformer is proposed in [110] for robotic navigation in a crowded environment. Here, the robot and person's states (position, velocity, orientation) over three time steps/time frames are given in sequence to the model. The feature extractor (linear projection layer) converts the sequence to an embedded feature vector and gives it to the network. It consists of a global spatial state encoder, a global temporal state encoder, and residual connections between them. Finally, a value-based reinforcement learning policy is used for selecting the optimal path. This model has been tested on the Collision-Avoidance-with-Deep-Reinforcement-Learning (CADRL) simulator and has achieved double success rate and a higher reward than the ORCA method within a shorter time.

For navigation tasks, various paths have to be analyzed and the relation between different locations has to be analyzed. So, representing the robot's surrounding environment in the form of a graph/map encodes the geometrical and topological relations between different locations in the map. Also, classical path planning algorithms can be integrated with the map. Vision Transformer (ViT) [14] along with A^* is used in [111] for enhanced path planning for quadrupedal robots. Initially, the environment is represented using a 2-D map where 1 represents the presence of an obstacle (and otherwise, 0). This map is divided into patches, pre-processed, and given to ViT. The ViT with its multiple self-attention layers, learns the complex spatial relations in the map, and a guidance map is constructed. Further, a differentiable A^* algorithm is used for finding the optimal path. The path planning time is drastically reduced when compared to the A^* algorithm alone (15-60% reduction; 15% in a maze and 60% in a mall).

The authors in [114] propose a navigation transformer for the object-goal navigation problem. The robot is equipped with an RGB-D camera, and YOLOv8 is used for object detection. This information is further enriched with semantic embeddings, and 3-d coordinates are obtained from depth data. With the obtained information, a neighbourhood map is created in the form of a binary grid. The transformer uses this information to learn inter-object relationships, and then LSTM is used to store this memory over time. Finally, asynchronous advantage actor-critic (A3C) reinforcement learning maps the state to actions, and the action with the highest reward is selected. This navigation transformer achieves 10% higher average success rate than the attention-based semantic similarity network (SSNet) [115].

A Map Attention with Semantic Transformer (MaAST) is developed in [116] for efficient robot navigation tasks with limited training data. The robot collects RGB-depth images and semantic information. A 3-layer CNN extracts visual features from fused RGB and depth images. The transformer is used for processing the egocentric semantic map, and a Gated Recurrent Unit (GRU) is used for temporal processing and maintaining memory of fused visual and map features. Finally, the proximal policy optimization (PPO) algorithm is used for decision making for robotic navigation. This MaAST framework achieves 13% higher success weighted by path length (SPL) with 80% reduction in training steps when compared to the model [117], which only uses RGB-D information.

A Relationwise Transformer Network(RTNet) along with a heterogeneous zone graph (HZG), is proposed in [118] for indoor object-goal navigation. The Detection Transformer (DETR) [119] detects the object categories and visual features from the captured RGB image. With the help of HZG, the scene is reconstructed as a graph with nodes and edges. The RTNet consists of a Node-to-Node (N2N) encoder with a self-attention mechanism and an Edge-to-Node (E2N) decoder with a cross-attention mechanism. The asynchronous advantage actor-critic (A3C) reinforcement learning uses the features from RTNet to enable the robot to make the right decision. This model achieves a 36% higher success rate than using the A3C RL algorithm alone (The success rate (SR) of A3C is 52% while that of RTNet is 88%).

A Crossmap transformer (CMT) is proposed in [112] for vision and language navigation tasks. The language instructions for navigation are given to the language encoder, which has a MiniLM network and a transformer. The surrounding environment and the robot's previous actions are provided in images to the visual encoder with a ResNet-152 network. The action decoder computes the likelihood of each action. These language and visual encoders and decoders make the CMT model the relationship between the instruction and navigation path. The key contribution is double back translation(DBT), where the CMT and CrossMap Speaker (CMS) are trained. The CMS generates the path, and this path is also learned to improve generalisation to unseen environments. This model achieved 34% and 33% higher success rates in seen and unseen environments when compared to the Seq2Seq model [120].

The authors in [121] examine the role of causal understanding for vision-guided robot navigation. Causality-Aware Transformer (CAT) networks are designed in such a way that navigation is not treated like a language model, where RNNs and transformers are used in such a way to process sequential data, where predicting output equally depends on all past inputs. CAT ensures that only one-step causality is taken, i.e., the next state is predicted by only the current state and action, not all past states. Although the past states indirectly influence the next state, their effect is already present in the present state. Initially, features are extracted from the RGB image with the help of Contrastive-Language-Image-Pretraining (CLIP) ResNet-50, and the embeddings are given to the transformer. The causal understanding module enforces single-step causality. The model is trained with reinforcement learning and has achieved superior performance compared to standard transformer and RNN models, with only 10% of training steps (CAT took 20M steps while RNNs took 200M steps for training).

A NavFormer is designed in [122] for target-driven navigation in disaster-like situational scenes, i.e., unknown and dynamic environments (where a global map is not available). The NavFormer framework takes images captured by the robot and the target image as input, and the data is encoded with the help of dual encoders. The static encoder extracts static environment features like walls and furniture while the general encoder extracts features of dynamic or moving obstacles. Here, the Jackal robot is used as the main robot while the Turtlebot3 robots are used as dynamic obstacles. The extracted feature vectors are processed by a causal transformer, which is GPT-2 based. This model is also tested for a real-world unseen environment with multiple robots, i.e. three jackal robots in one experiment, and a jackal robot along with three Turtlebot3 robots in another. The NavFormer achieves almost double the success rate and half the collision rate compared to the decision transformer.

Classical reinforcement learning-based approaches typically treat the goal state as a condition. Instead of taking this view, the authors in [20] treat the goal state as an input and develop a transformer-enabled reinforcement learning scheme for autonomous-ground vehicle navigation. The goal point coordinates and the most recent four images are converted to embeddings and given to the goal-guided transformer (GT) (modified visual transformer). From the transformer, a goal-aware representation of the scene is created and given to the SAC RL algorithm, which generates robot actions. This GT-based RL has been shown to outperform Visual-Transformer-based RL and CNN-based RL.

Normally, the entire (robot) navigation is treated as part of a modular pipeline in classical control methods and perception and control modules are decoupled. The authors in [123] take a different view and propose a Control Transformer that treats goal-directed navigation as a sequential decision-making task. The data from the camera, LiDAR, and relative goal positions are embedded and sent to the control transformer as tokens in sequence form. During the training phase, the environment is represented using a probabilistic road map (PRM). This model has been applied on a Turtlebot3 and has achieved 96% and 71% success rates in reaching the target in simulation and in a real-world experiment, respectively.

A Pathformer (transformer-based framework) is proposed in [19] for vision-based navigation in various off-road environments(various terrains). Initially, Resnet-50 is used for feature extraction, and then a transformer encoder-decoder produces a segmentation mask of the image. The masks help to classify various terrains such as rock, concrete, gravel and grass. Then the masks are further refined, and a safe path for navigation is generated with the help of waypoints. The advantage of this model is that it only uses RGB images and does not depend on additional sensors like LiDAR or GPS.

Sample inefficiency (the requirement of large datasets) and limited computational resources are the two main problems for implementing deep learning models on edge devices. So a vision transformer pre-trained with Self-Distillation-with-NO-labels (DINO) [124] is used in [125] for line follower and obstacle avoidance tasks. The model is fine-tuned with just

70 labelled images, and the model's resolution is adjustable between 240 ∗ 240 and 960 ∗ 960 so as to balance segmentation detail with inference speed. This ViT model has been tested on the Duckiebot on the Duckietown platform. The 1-block ViT and 3-block ViT are compared with CNNs of 24,32 layers, and 1-block ViT has offered greater balance with regard to detection accuracy and speed, whereas 3-block ViT achieved higher detection accuracy.

The authors in [126] were among the earliest to integrate the transformer architecture into a reinforcement learning framework for multiple robot path planning tasks in which the robots do not explicitly communicate with each other. The transformer models the interactions between the agents and also the temporal dependencies. Imitation learning, which accelerates the training, is combined with a reinforcement learning module consisting of a double deep Q-network (DDQN). This model has achieved a higher success rate and fewer steps are required to navigate than the existing models (Decision Causal Communication (DCC) [127], PRIMAL [128]) when tested in simulation with up to 64 robots. It has also been validated in an indoor environment with three robots.

The authors in [129] present a group-wise attention mechanism for identifying the navigability of different terrains from RGB images/videos. In particular, the authors identify different terrain groups in off-road and unstructured outdoor terrains. The captured RGB image features are extracted using the backbone of modified mixed transfer (MiT). The image is divided into patches and sent to a series of encoders with the help of a multi-head self-attention (MHSA) mechanism, which generates feature maps of various resolutions. These maps are fused using a groupwise attention head and further passed to the segmentation head to get a coarse-grained semantic segmentation map. A robot path is obtained with this map and Terrain Elevation-based Robot Path planning (TERP). This model has achieved, on an average 10% higher success rate and 37% reduction in false positive rate of forbidden regions.

The transformer-based memory model is used in [130] for interactive visual navigation in cluttered environments. The navigation task is formulated as a Partially Observable Markov Decision Process (POMDP). The robot captures RGB and depth images, and CNN is used as a feature extractor. These RGB and depth features, along with goal position, are given to the transformer encoder, i.e., the believe state encoder, which encodes the past observations into a context-aware belief state. It consists of a stack of multi-head attention blocks. It uses casual and local attention to maintain temporal consistency and to focus on recent interactions, respectively. The soft actor-critic (SAC) algorithm is used to optimize the path. This model achieved 17% higher success rate than using SAC alone in unseen environments in Gazebo simulator. Table 7 captures the works on transformers. We now move on to graph attention-based methods.

B. GRAPH ATTENTION-BASED APPROACHES

For robotic navigation tasks, particularly in a dynamic environment (involving humans) or multi-robot tasks, the surrounding environment can be represented as a graph so that the model can capture the interactions between the robots themselves or human-robot interactions. The nodes in the graph represent the robots or humans while the edges represent interactions between the nodes. Graph attention is effective for crowd navigation due to relational modeling. Some of the tools used for robotic navigation tasks include graph convolutional network (GCN) [131] and Graph Attention neTwork (GAT) [132], [133], [134], [135]. GAT uses an attention mechanism which uses attention weights so that it can focus on most relevant neighbours.

Figure 9 illustrates how graph attention-based approaches are useful for robot navigation. In a complex multi-agent environment, collision avoidance is performed by applying a graph attention network to extract the interactions between robots.

Graph Attention neTwork (GAT) is used in [133] for robot navigation in a dynamic environment where other robotic agents also move. The attention mechanism in GAT computes the effects of nearby agents, and reinforcement learning is used for learning navigation policy. Further, Optimal Reciprocal Collision Avoidance (ORCA) is used to enhance short-term safety, i.e. collision avoidance. The combined GAT, RL and ORCA model has achieved a success rate of 98% in simulation when five other agents were present.

One more application of GAT can be seen in the hierarchical motion planning framework in [134] for multi-robot navigation tasks. The robot is equipped with a 2-D laser scanner, and the most recent three scans are sent as input. Depending on the rule-based algorithm with the sensor data, the motion selector chooses a collision avoidance policy if there is an obstacle; otherwise, a target-driven policy is selected. The target-driven policy guides the robot toward its goal. The collision avoidance policy uses GAT with the Proximal Policy Optimization (PPO) algorithm, where GAT uses a soft attention mechanism for modelling interactions between the robots. This model has achieved a success rate of 100% when tested on a circular simulation setting, whereas CNN-based policy completely fails, and ORCA achieves only a 60% success rate.

A group-aware robot navigation framework is proposed in [135] for socially compliant robot navigation. This framework combines a group awareness mechanism (GAM) with a spatio-temporal graph attention network (GAT). The robot is equipped with RGB-D camera and 3D LiDAR. GAM consists of YOLOv5, which is used for people detection; Kalman filters are used to find their positions with the help of LiDAR data, and groups of people are represented using convex hulls. The robot and people are represented as a graph, and spatio-temporal GAT consists of a graph convolutional network (GCN) and LSTM network for modelling the spatial and temporal interaction, respectively.

TABLE 7. Transformer-based models.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environment tested	References
PathFormer	Autonomous off-road navigation	UGV	Handles various terrains, only camera is used	Not implemented in real time	Incorporating temporal information to the model	Real world dataset	[19]
Spatio-temporal transformer	Robot crowd navigation	Social robot	Efficient, safer and faster navigation	Static obstacles are not considered/modelled	Incorporating SLAM algorithm for accurate localization	Gazebo Simulator	[108]
Goal guided transformer	Navigation in unseen office environment	SCOUTMINI	Good generalization for unseen environments	Limited performance in presence of dynamic obstacles	Adding multi-modal inputs like LiDAR	Indoor	[20]
Vision transformer	Legged locomotion indoors	Boston dynamics spot, Unitree Go1	Faster planning time in complex environments	Accepts only binary occupancy maps as input	Extending the model by also taking images as input	Indoor (Complex)	[111]
Control transformer	Navigation in unknown environments	Turtlebot3	Zeroshot sim2real transfer, without additional real-world data	Limited real time performance due to sensor noise, drift	Extending to dynamic obstacles like humans	Indoor (Maze)	[123]
Transformer based IRL	Multi-robot coordinated path planning	AGVs-3	Higher success rate, lower average steps in navigation	Limited performance when number of robots increases	Extending to larger maps/environment	Indoor	[126]
Crossmap transformer	Vision & language navigation	Domestic service robots	Improved generalization on unseen environments	Shortest path is not optimized	Integrating with RL, transfer to real world	Simulation (Matterport3D)	[112]
Navigation transformer	Object goal navigation	Autonomous robot	Requires fewer steps to find the target	Need heavy computational resources	Include robust fall back policies so that robot recovers from failure	AI2-THOR simulator	[114]
Vision Transformer	Lane following	Duckiebot	Shallow ViT outperformed deeper ViT	Not suitable for high resolution inputs	Extending to unstructured environment	Indoor (Duckietown platform)	[125]
Map Attention with Semantic Transformers (MaAST)	Point goal navigation	Autonomous agent	Higher success rate with fewer training steps/time (only 20%)	Assumes idealized GPS sensor, compass (not practical)	Extending to outdoor environments	Habitat simulator	[116]
Causality-aware transformer	Object & point navigation	Service robot	Similar performance with only 10% training steps	Assumes one-step causality (not applicable to dynamic environments)	Optimize the model for edge devices	RoboTHOR, Habitat simulator	[121]
NavFormer	Target driven navigation	Jackal robot, Turtlebot3	Generalized with unseen environments, dynamic obstacles	Assumption of static target	Incorporating multi-modal sensing	Gazebo simulator, Indoor	[122]
Hybrid spatio-temporal graph transformer	Socially-aware navigation	Service robot	Superior efficiency and safety	Assumes robot not visible to humans (not practical)	Incorporating real time feedback from humans (testing phase)	Indoor	[109]
Spatial temporal state transformer	Crowd-aware autonomous navigation	Mobile robot	Faster, safer, smoother navigation paths	Not deployed on real robots	Extending to multi-robot navigation	CADRL Simulator	[110]
Relation-wise transformer network	Object goal navigation	Mobile robot	Good generalization to unseen environment	The model struggles with corner traps, small targets	Incorporating visual semantic perceptions	Indoor (Real world AVD)	[118]
GA-Nav	Autonomous navigation	Clearpath Jackal, Husky	Improved success rate and computationally efficient	Requires consistent training data labelling	Extend to even extreme weather conditions	Outdoor (Unstructured terrains)	[129]
Scene Memory Transformer(SMT)	Interactive visual navigation	Embodied agent	Enhanched performance, stability, better decision making	Tested only in simulation, lower effort efficiency	Cost aware planning to improve effort efficiency	iGibson simulator	[130]

A Message Aware Graph Attention neTwork (MA-GAT) is proposed in [132] for multiple robot path planning on a large scale. Initially, the environment is represented in the form of a map, and CNN with Resnet block is used for feature extraction. A dynamic communication graph is constructed where nodes are represented by robots and the robots communicating are represented by edges. The robots share feature vectors with the neighbour robots via graph neural networks, and an attention mechanism is applied to select the more relevant features. Finally, the policy network decides the robotic action that needs to be taken. This model is trained on a $20 * 20$ grid with 10 robots and, when tested on $50 * 50$ grid with 100 robots and $200 * 200$ grid with 1000 robots, achieved success rates of 95% and 82% respectively, showing strong generalizability.

Classical robot navigation systems based on traditional reinforcement learning tend to consider human-robot interactions in a unidirectional fashion. However, the authors in [136] have considered and modelled both human-robot and human-human interactions for robot navigation in crowded environments and proposed a Local Map-Social Attentive Reinforcement Learning (LM-SARL) model. This model

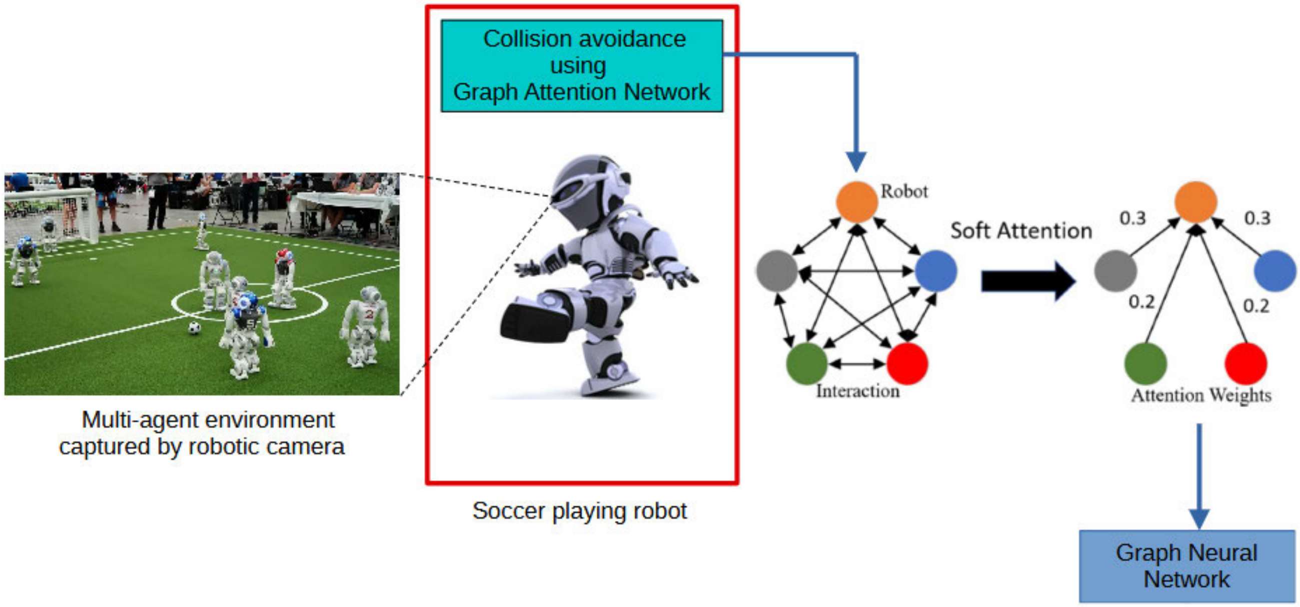

FIGURE 9. Attention Graph-based approach for robot navigation.

consists of an interaction module, a pooling module and a planning module. The human-robot interactions are modelled by the interaction module with the help of coarse-grained local maps. The pooling module, which consists of a self-attention mechanism, is used to compute the attention score for each person and aggregate the interactions into an embedding vector. Finally, the planning module estimates the value function of the robot and crowd states, which are optimal for the robot's safe navigation. This model achieved double success rate and a higher reward than the ORCA method within a shorter time.

A Heterogeneous Graph Transformer (HGT) along with a task-driven attention mechanism is proposed in [137] for hierarchical relational object navigation task. From the RGB and depth images of the environment, a scene graph is constructed. This graph is given to HGT with an attention mechanism, and the extracted features are fused with those extracted from RGB and depth images. This model achieved a success rate of 88%, which is 30% higher than using RGB-D features alone. Liu et al. [138] proposed recurrent graph neural networks with an attention mechanism for robot navigation in dense environments by enabling the robot to be socially and intentionally aware. Initially, a person's future trajectory is predicted either by the Gumbel Social Transformer (GST) or by the constant velocity model. A spatio-temporal interaction graph is then constructed with nodes as agents and edges corresponding to interactions. The recurrent graph neural network includes two attention mechanisms: one for modelling human-human interactions and the other for robot-human interactions. It also contains a Gated Recurrent Unit (GRU) to capture temporal dependencies across time. This model achieved a success rate of 89% in simulation and 83.33% in an indoor environment when tested on the TurtleBot 2i robot.

Robot navigation, even with deep learning models, becomes difficult when the crowd size increases. Recognizing this, learned attention is incorporated into a graph-based reinforcement learning network in [131] for robot navigation in crowds. The model, termed Gaze modulated Graph Convolutional Network-based RL (G-GCNRL), consists of attention, crowd aggregation and value networks. Both attention and crowd aggregation networks consist of a two-layer graph convolutional network where the attention network learns (which) people to focus on based on human gaze data, and these attention weights are given to the graph convolutional network of the crowd aggregation network to model the interactions between people and the robot. Finally, the value network consists of reinforcement learning for selecting the optimal robot path. This G-GCNRL model has achieved 11% higher success rate and navigation was 16% faster when compared to the socially-aware RL model.

MultiSoc is proposed in [139] for goal-directed robot navigation with multi-robot implicit coordination (where robots explicitly do not communicate with each other) in crowded environments. MultiSoc integrates graph neural networks (with attention mechanism) into the reinforcement learning method for modelling the interactions (i) between humans and (ii) between humans and robots. It consists of two graph neural networks: an edge selector that constructs a sparse graph that prioritizes the most important interactions and a crowd coordinator that modifies nodes based on its neighbourhood influences. The model uses trajectory prediction and density factors to adapt to varying crowds. This model achieved an 81% success rate when tested in the MultiCrowdNav simulator with six humans and six robots.

A novel framework, named MARVEL (Multi-Agent Routing in Variable Environments with Learning), is designed in [140] for multi-robot navigation in uncertain topological environments. The model is designed to improve the group of robots' stochastic on-time arrival (SOTA) probability, where edge traversability is unknown and only revealed to the robot when it arrives at the edge's starting node. The model reformulates the problem into a Partially Observable Markov Decision Process (POMDP). This model combines dynamic adaptive graph embedding with entropy-based online experts so that robots can collectively explore and adjust the paths accordingly. This model has achieved a SOTA probability of 63%, whereas other baseline models have only achieved 30-55%. Table 8 lists the various graph attention-based works. We now proceed to discuss works that use other types of attention mechanisms.

C. OTHER ATTENTION-BASED METHODS

Although transformer-based attention models and graph attention-based models are widely popular, a few other attention-based approaches have been developed recently. These methods often include developing customized architectures or integrating with existing reinforcement learning methods, path planners or spatiotemporal data representations. Most of these models focus on handling sensor input, which can be used in path planning, i.e. focusing on task-specific spatial regions or temporally varying patterns. This improves the decision-making capabilities of robots, particularly in dynamic or cluttered environments.

An Attention-based Value Classification Actor-Critic (AVCAC) architecture is proposed in [141] for safe and efficient robot navigation in unknown environments. Initially, an encoder stores the input sensor data from LiDAR, RGB-D camera, and a local attention module extracts key features like goals and obstacles. The key component in this model is a Value-Classified Rollout Replay (VCRR) buffer, which acts as an experience buffer, i.e., it classifies and stores the experiences in reinforcement learning. To avoid future collisions, a lookahead multi-step prediction reward function is used. This model is tested on an automated guided vehicle and has achieved a 95% success rate.

A LiDAR-based place recognition system is proposed in [142] for large-scale outdoor unstructured environments like orchards where conventional global navigation satellite systems are unreliable and often fail. The autonomous guided vehicle is equipped with a LiDAR and Spatial Binary Pattern (SBP) descriptor that encodes the environment's spatial structure and point density and converts 3-D LiDAR scan into cylindrical voxel bins and then projects them on to a 2-D bird's-eye view matrix. Then, an attention score map is used to highlight the region of importance. Hierarchical two-stage matching is used to detect loop closures. The first stage is fast attention score map candidates search, which eventually reduces the potential loop closure candidates. The second stage is overlap estimations re-identification, which is used to confirm loop closures in the first stage by ensuring geometrically consistent matches are only accepted. This model achieved improved localization accuracy.

A Spatio-Temporal Attention Deep Reinforcement Learning (DRL) model (STA-DRL) is proposed in [143] for LiDAR-based robotic navigation. The model uses A^* path planner, and the next five upcoming waypoints and the LiDAR scan data are given as input to the model. Next, in the pre-processed stage, with the help of a Temporal Accumulation Group Descriptor (TAGD), which uses Iterative Closest Point (ICP) alignment, the dynamic obstacles are focussed by removing the noise and compressing the LiDAR data. The attention-based feature extractor has two parallel streams/modules. Firstly, a spatial attention module detects static obstacles and focuses on risky LiDAR scans, i.e., possible collisions. Secondly, a temporal attention module is used for detecting dynamic obstacles. The spatial and temporal attention module features are fused and fed into a Deep Deterministic Policy Gradient (DDPG) network to achieve smooth and collision-free robot navigation.

The authors in [144] proposed attention-based deep reinforcement learning for robotics navigation tasks in highly dynamic environments. The self-attention mechanism is integrated into the Twin Delayed Deep Deterministic (TD3) policy gradient algorithm for context-aware decision-making during navigation. This attention-based TD3 model achieves 4.8% and 2.5% higher success rates in navigation in static and dynamic scenes, respectively, when compared with using TD3 alone. Table 9 lists the other attention-based approaches.

VII. ADAPTIVE APPROACHES FOR ROBOT NAVIGATION

While attention mechanisms and generative models have been of tremendous interest for robot navigation during the last few years, adaptive methods have been studied for much longer. Early efforts are based on reinforcement learning [21]. The reinforcement learning paradigm relies on the notions of agent, action, reward and penalty. In the context of robot navigation, a reward is for reaching the goal (or reducing the distance to the goal) while a penalty is applied when a robot gets close to an obstacle (or moves farther from the goal). Recently, an extension named deep reinforcement learning that combines classical reinforcement learning with deep networks (for example, convolutional neural networks) has been proposed to enhance performance. Other adaptive methods have also emerged.

In this section, we survey the recent literature on adaptive methods relating to graph neural networks and knowledge graphs, deep reinforcement learning as well as other approaches such as imitation learning and situational graphs. *It turns out that adaptive methods can be readily combined with attention mechanisms [143] to enhance the performance of navigation strategies.*

TABLE 8. Graph attention based models.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environment tested	References
LM-SARL	Crowd-aware robot navigation	Segway Loomo robot	Improved anticipation of crowd dynamics	Assumes holonomic robot dynamics	Extending to outdoor environments	Indoor	[136]
MultiSoc	Multi-agent implicit coordination in crowd	Multiple social robots	Higher scalability, robustness	Limited performance when humans have limited FOV	Better modelling of different groups of human behaviour	MultiCrowdNav simulator	[139]
HGT	Hierarchical relational object navigation tasks	Fetch robot	Higher success rate, faster training time	Assumes static obstacles only	Extending it to manipulation tasks	iGibson 2.0 simulator	[137]
MA-GAT	Multi-robot path planning	Mobile robots	Excellent generalizability w.r.t map size, no. of robots	Not tested on continuous spaces	Incorporating RL instead of depending on expert data	Simulation	[132]
GNN	Intention aware robot navigation	TurtleBot 2i	Improved success rate, safety and social awareness	Assumes non-reacting humans towards the robot	Extending to non-holonomic robots	Indoor	[138]
GAT	Robot navigation	Holonomic robot	Improved robustness in dynamic environment	Real time uncertainty is not considered	Extending to non-holonomic robots	Simulation (Python)	[133]
GAT	Multi-robot navigation	Differential drive robots	Higher path efficiency and scalability	Robot speed is slow	Integration with global path planning algorithms	ROS Stage, OpenAI Gym Simulators	[134]
ST-GAT	Group aware robot navigation	Mobile robot	Higher navigation efficiency, social compliance	Limited to 4 people in a group	Incorporating active path clearing strategy	Indoor	[135]
G-GCNRL	Robot navigation in crowded environments	Unicycle robot	Smoother and faster trajectories	Model assumes structured crowds	Integrating with self-supervised gaze prediction	Simulation	[131]
MARVEL	Multi-robot navigation amidst uncertainities	Pioneer 3-DX robot	Higher on-time arrival probability	Computationally complex, expensive	Enhanching generalization, computational efficiency	Indoor (Maze)	[140]

TABLE 9. Other Attention-based Models.

Model	Purpose	Robot	Findings	Limitations	Potential enhancements	Environment tested	References
AVCAC	Autonomous navigation in dynamic environments	AGV	Reduced collision rates	Highly dependent on local observations	Integration with SLAM	Gazebo simulator, Indoor	[141]
SBP-ASM	Autonomous navigation for agricultural tasks	AGV	Improved recall rate, localization accuracy	Computationally expensive when LiDAR range increased	Integrating with semantic segmentation	Outdoor (Orchards)	[142]
STA-DRL	Navigation in dynamic environments	Kobuki Turtlebot 2	Improved robust to sensor noise, ICP misalignment	People are modelled as cuboids, so minor artifacts occur	Extending to outdoor environment	Indoor	[143]
Attention based DRL	Navigation in composite environments	Mobile robot	Good generalization to unseen environments	Requires longer training time	Incorporating multi frame LiDAR data	Gazebo simulator	[144]

A. GRAPH NEURAL NETWORKS AND KNOWLEDGE GRAPHS

Graph neural networks (GNNs) are neural architectures to process graphs [9]. One type of graph of particular interest in robot navigation is a knowledge graph. Knowledge graphs encode a set of facts about objects by defining relations between them [9]. In other words, knowledge graphs are structured representations of facts and consist of entities, relationships and semantic descriptions [145], [146], [147], [148]. Graph neural networks have been extensively studied for solving multi-robot navigation problems. Knowledge graphs are valuable from the point of view of semantic navigation and object-goal navigation.

Figure 10 depicts the use of knowledge graph for robot navigation. In the case of service robots deployed indoors, an RGB-D image of environment is captured and object detection is performed on this image. A knowledge graph is constructed using the detected objects and their distances. The knowledge graph is valuable for capturing semantic information. It also serves as a means to construct movement relationship among entities [149] and to design action plans adaptable to run-time [150].

Early works utilizing graph neural networks for motion planning and navigation [151], [152] directly represent the robot's configuration space. Recent work [153] involves utilizing knowledge graphs to obtain an intermediary representation for compartmentalizing the workspace. A framework termed *RoboPlanner* is presented in [150] to generate action plans in autonomous robots. Knowledge property graphs are used to model the knowledge of world models, capabilities of the robot as well as task templates. The authors present simulation results. Extension of the work

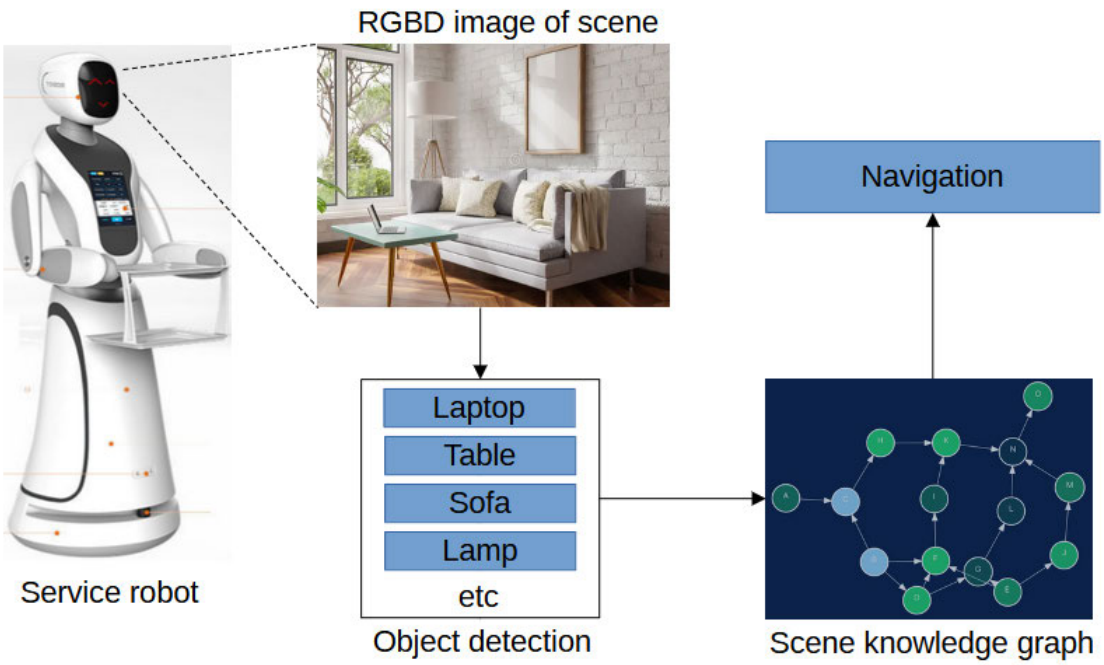

FIGURE 10. Knowledge graphs for robot navigation.

in [150] to physical robots would be valuable. The authors in [154] consider the role of representation and reasoning about semantic knowledge in robotics. In particular, they discuss the use of knowledge graph embeddings as semantic representations and identify the challenges with respect to incremental updation of the representation. The work in [154] can be extended to incorporate data from physical robots in a dynamic setting.

Graph neural networks have been valuable for decentralized multi-robot path planning and navigation [155], [156]. These networks work in conjunction with other deep neural networks such as CNN. For instance, the features extracted by CNN are communicated among robots via graph neural networks [155]. The work in [155] assumes zero delay in communication between robots. Studying the effectiveness of their approach relaxing this assumption would be valuable. Graph neural networks also play a valuable role in robot inference with other robots and router nodes [156]. The performance of the approach in [156] can be enhanced by incorporating memory units and attention modules.

Another use of graph neural networks is for learning control admissibility models for multi-robot navigation [157]. The approach in [157] is scalable but does not permit coordination and communication between the robots. This can lead to critical situations (such as deadlocks). This is an area that can be further explored via modification of the architecture.

An approach to robot planning based on behaviour tree and knowledge graph is proposed in [7]. The authors describe building of a semantic network for planning via an ontology knowledge base, a scene knowledge base, an environment-aware knowledge base and a task knowledge base. Using these, task behaviours are created and reasoning is performed. A technique for navigating to a remote object invisible in the current view is presented in [158]. The authors construct a scene knowledge graph and then design a reasoner based on probability models and a navigator for moving to the object of interest. However, the navigation length and navigation time are somewhat high. There is scope for reduction of these. Further, the success of the navigation task itself can be enhanced. The task of inspecting substations autonomously is studied in [159]. Multisource data fusion is considered and the role of knowledge graph is investigated for summarizing and visualizing a five-way manual of substation inspection. Capturing semantic information via knowledge graphs in the context of SLAM has also been studied [149]. The approach has been tested in simulations and experiments. The approach in [149] can be enhanced to obtain better real-time performance and accuracy.

An approach to efficiently reach predefined goal objects with just few steps is presented in [160]. The authors align the knowledge graph with visual perception for this task. Another work on object-goal navigation using knowledge graphs is reported in [161]. The method in [161] is based on three modules, namely an object-goal navigation module, a cognitive memory module and an interaction module. The approach in [161] has been tested in experiments but there are limitations in semantic segmentation that may sometimes lead to misidentification of personal objects of the same class.

A technique for visual navigation based on knowledge graphs and a value regularization policy is reported in [162]. The approach in [162] can be extended to enhance the performance for dynamic environments. A method based on knowledge graphs to increase the search accuracy of Robot Operating System (ROS) packages is described in [163].

Some works use graph neural networks and knowledge graphs along with other contemporary artificial intelligence approaches such as transformers. The authors in [164] present a goal-oriented approach to visual semantic navigation by combining the knowledge graph with transformers. Agents are made to learn how to make a series of action decisions based on visual input to find a specified goal. Several directions for further study of the approach in [164] are possible. The authors indicate that probabilistic relationships between co-occurent objects may be considered for making the navigation scheme more robust. Table 10 lists methods that use knowledge graphs. We now proceed to the next adaptive method, namely deep reinforcement learning.

B. DEEP REINFORCEMENT LEARNING

Deep reinforcement learning (DRL) was introduced as a paradigm for playing Atari 2600 games in [22]. A deep Q-network was proposed for human level control by the same research group in [23]. Since then, deep reinforcement learning has been applied to a wide range of tasks. We focus on its utility for robotic navigation.

A picture illustrating an application of deep reinforcement learning to navigation in agricultural robotics is shown in Figure 11. The robot arm is assumed to be mounted on a mobile platform which traverses the field where the apples are grown. The input to the machine learning method running on the robot is an image. The robotic arm (on the mobile base) is trained to go from its initial position to the final position. This process is repeated for many samples and this leads to update of the model parameters. Through images, the robotic arm adapts and moves to pluck the ripe fruits. We now discuss various efforts reported in the robot navigation literature on using deep reinforcement learning.

Works on robot navigation using deep reinforcement learning can be classified based on the nature of the environment. Navigation in pedestrian-rich environments has been studied by a number of research groups [136], [166], [167], [168], [169], [170]. Some works look at the problem as a crowd-aware navigation task. Different networks have been used in these works in combination with the basic reinforcement learning paradigm. In particular, Long Short Term Memory (LSTM) network [136], [166] and Convolutional Neural Network (CNN) [168], [170] are frequently used as part of the policy network. We now briefly discuss the contributions on crowd-aware navigation.

The authors in [166] develop a collision avoidance with reinforcement learning algorithm that takes advantage of the GPU-based asynchronous advantage actor-critic scheme in [189] for policy learning. Experiments with a mobile robot as well as multiple multirotors are reported. A crowd-aware robot navigation scheme is presented in [136] which incorporates an attention mechanism. The goal of the attention mechanism is to learn the collective importance of neighboring humans for socially compliant navigation. The authors report experiments with the Segway platform and demonstrate that their approach is time efficient. An approach to get a robot to unfreeze in dense pedestrian crowds is presented in [167] and it takes into account the coordination between robot and humans. However, the method experiences challenges due to the use of only 2D LiDAR information and can be enhanced with semantic knowledge as well as vision data. Robot navigation in a crowded environment is handled by a combination of imitation learning and reinforcement learning in [168]. The method in [168] processes information about static and dynamic objects separately and this enables the robot to move differently when aproaching static obstacles than when encountering moving pedestrians. The method has a somewhat higher rate of collisions with agents in the field of view as well as those outside the field of view and this is an area where further enhancements are possible. The authors in [169] present an approach that uses graph neural networks in conjunction with reinforcement learning to learn local interactions between different pairs (such as object and robot, object and object). A hybrid scheme that combines the dynamic window approach to collision avoidance [190] with deep reinforcement learning is studied in [170].

Navigation with the reinforcement learning paradigm by *legged robot structures* has also been explored by a few research groups [171], [172], [191],. A walking pattern generator becomes an important component of these designs. Deep reinforcement learning is shown to be an effective method to address the limitations of model-based control for the walking robots. The authors in [171] identify the challenges that arise including (imprecise) tuning of award functions and absence of guarantees with regard to safe operation. They then develop a constrained guided policy optimization framework for tracking base velocity commands. The sophistication of the environment considered in [171] is not high and the authors indicate the possibility of using perception data in the future for handling more challenging terrains. An approach that combines model-based and model-free methods for quadrupedal locomotion in challenging terrains in developed in [172]. The components of the design include a terrain-aware planner and a foothold plus base motion controller. Excellent results are shown for walking over a set of large gaps with different sizes. Humanoid locomotion is studied in [191] by formulating a Markov decision process that aids in safe and data-efficient learning.

DRL has also been valuable to perform navigation under non-ideal conditions (such as varying illumination). An example is the work in [177] where the authors introduce the notion of generalized computation graphs for self-supervised DRL. Grid-based navigation using DRL has been studied by a few groups [175], [176]. In this, the use of a

TABLE 10. Graph Neural Networks (GNNs) and Knowledge Graph-based Methods.

Model	Purpose	Findings	Validation	Potential enhancements	References
Knowledge graph	Navigation of warehouse robots	Design time action plans adaptable to run-time	Simulations	Extension of framework to physical robots	[150]
GNN	Human-aware robot navigation	Very effective to model disruption and highly scalable	Simulations	User profiling and personalization	[165]
Knowledge graph	Visual SLAM for navigation	Constructing apriori movement relationship among entities	Simulations and Experiments	Improving real-time performance and accuracy	[149]
GNN	Decentralized multi-robot path planning	Effective communication (among robots) of features extracted by CNN	Simulations	Incorporate (non-zero) delay in communication between robots	[155]
Knowledge graph	Object-goal navigation	Efficient search for target objects	Simulations and Experiments	Handle misidentification of personal objects of same class	[161]
GNN	Navigation in dynamic environments	Effective plan generation	Simulations	Incorporating safe intervals to increase success rate on hard instances	[152]
Knowledge graph	Semantics-based navigation	Continual learning for updating representations	Simulations	Adaptation of continual learning principles including data from physical robots	[154]
GNN	Decentralized unlabelled navigation	Effective robot inference with other robots and router nodes	Simulations and Experiments	Enhance performance using memory units and attention modules	[156]
Knowledge graph	Navigation for intelligent inspection	Multisource fusion for patrol robots	Simulations	Enhance obstacle recognition ability	[159]
GNN	Multirobot navigation avoiding deadlocks	Learning control admissibility models	Simulations	Handling situations that require inter-robot interactions	[157]
Knowledge graph	Object-goal navigation	Obtaining accurate and coherent scene descriptions	Simulations	Enhancing performance in dynamic environments	[160]
Knowledge graph	Visual navigation	Narrowing down search for an object by direction association	Simulations	Enhancing performance for dynamic environments	[162]
Knowledge graph	Remote object navigation	Effective scene understanding and adaptation to different scenes	Simulations and Experiments	Reduce the time and increase the success of navigation towards object and	[158]
Knowledge graph	Robotic guide for exhibition hall	Effective welcoming of guests, leading the way and adaptation based on relationships	Simulations and Experiments	Improve flexibility of AND-OR graph construction	[7]

variation of LSTM, namely Gated Recurrent Unit (GRU) has also been studied [176]. Navigation seeking a specific object has also been approached via deep reinforcement learning. The authors in [181] use color as a means for the robot to search for an object. It is worth noting that this problem (referred to also as object goal navigation) has also been explored using attention mechanisms.

The idea of using context for navigation has been explored from a deep reinforcement learning perspective in [180]. This is compared by the authors with the dynamic window-based approach for navigation used in [170]. The benefits of the high-level semantics-based approach in [180] are in terms of safety and robustness. The work in [180] has potential for further improvement by addition of more semantic classes (including long corridors, doors) into the training environment.

Navigation in rough outdoor terrains has also been studied using the deep reinforcement learning approach [178], [179].

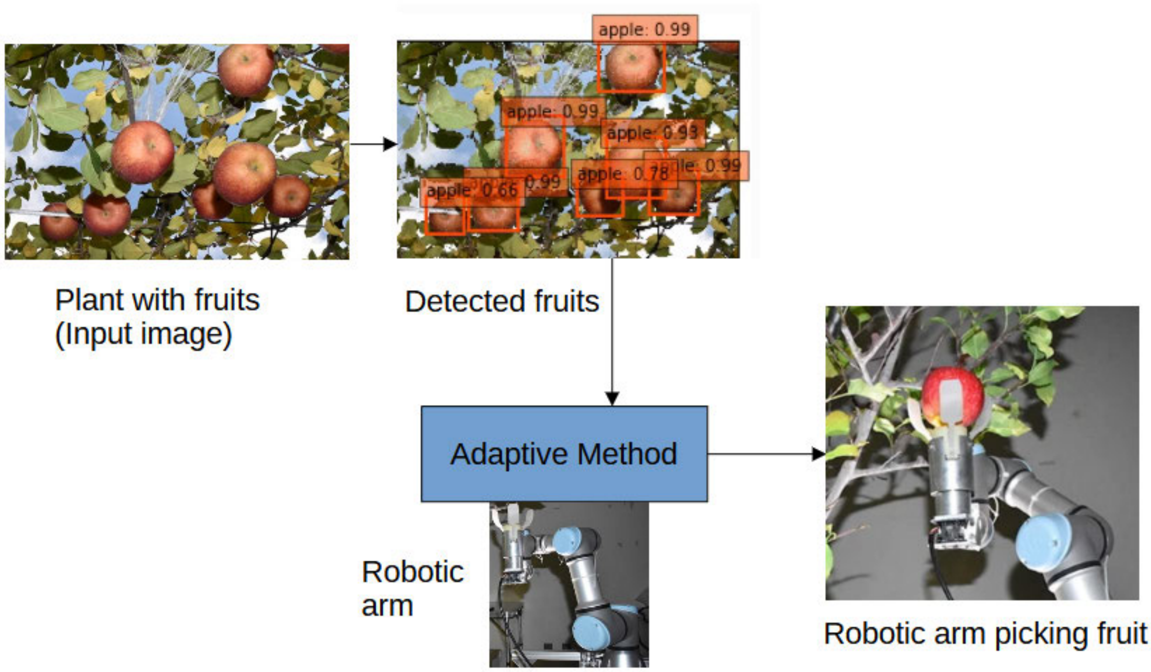

FIGURE 11. **Deep reinforcement learning for agricultural robotics.**

TABLE 11. **Deep Reinforcement Learning-based methods (Sim denotes Simulation while Expt denotes experiments with robots).**

Setting	References	Networks/Tools Used	Validation
Navigation in pedestrian-rich environment	[166], [167], [136], [168], [169], [170]	LSTM [166], [136] CNN [168], [170]	Sim + Expt (All the works)
Legged robot locomotion in uneven terrain	[171], [172]	—	Sim [171] Sim + Expt [172]
Multirobot collision-avoidance and navigation	[173], [174]	Hierarchical relational graph [174]	Sim [173] Sim + Expt [174]
Grid-based navigation	[175], [176]	GRU [176]	Sim + Expt
Navigation with changing illumination	[177]	Generalized computation graph	Sim + Expt
3D Rough Terrain Navigation	[178], [179]	CNN [178] LSTM [179]	Sim + Expt [178] Sim [179]
Semantics-guided navigation in dynamic environment	[180]	Multilayer Perceptron (4 Hidden Layers)	Sim + Expt
Navigation seeking an object	[181]	CNN	Sim + Expt
Navigation in a maze-like indoor environment	[182], [183], [184], [185], [186], [187], [188]	LSTM [182], [188] CNN [184], [187] Fuzzy Logic [186]	Sim + Expt

A sim-to-real pipeline for navigation is proposed in [178]. The simulated environment is used to provide simulated robot sensory data which serves as input to the deep reinforcement learning module. It is shown that the DRL approach in [178] has a success rate of 87% and further, the DRL approach leads to the shortest cumulative distance travelled. The authors in [179] report a DRL network that uses raw sensor data from the robot's onboard sensors to determine a series of local navigation actions for the robot. It is worth pointing out that a mapless navigation strategy is particularly valuable in this outdoor setting and hence DRL becomes a natural choice. DRL has also been extended to the scenario when multiple mobile robots perform point-to-point navigation. Approaches based on hierarchical relational graph [173] and hybrid control [174] have been studied for this purpose. Table 11 captures the various works on navigation using deep reinforcement learning. We now briefly discuss a few works on other adaptive methods.

C. OTHER ADAPTIVE METHODS FOR ROBOT NAVIGATION

Recently, the notion of *Situational Graphs* has been used for robot navigation. These graphs enable the robot to understand the environment in which it operates. Situational graphs are designed to adapt to changing environmental conditions. They bridge LiDAR-based geometric simultaneous localization and mapping with scene graphs [192]. Studies with legged robots are reported in [192]. The current experiments assume a Manhattan world (corresponding to the existence of three mutually orthogonal directions). Relaxing this would be a valuable extension to the work. Situational graphs have also been studied for robot localization in [193].

Another adaptive approach for robot navigation is imitation learning [194], [195], [196]. Imitation learning is slightly different from reinforcement learning. Here again, an agent is involved. However, imitation learning refers to an agent's acquisition of behaviours by observing a teacher (expert) demonstrating a certain task. An approach for target-driven visual navigation in indoor environments based on imitation learning is presented in [187]. The method is based on learning a variational generative module from expert demonstrations. The power of the technique is shown via improved training data efficiency and helps to perform map-less navigation.

VIII. KEY FINDINGS AND FUTURE RESEARCH DIRECTIONS

In this section, we first present a summary of the important findings of this survey. This is followed by a discussion on potential topics for further study.

A. SUMMARY OF MAIN FINDINGS OF THIS SURVEY

As surveyed in section III, several classical works are based on fuzzy logic. Some works have used simple neural networks (with small number of hidden layers) to tune parameters of membership functions that are part of the fuzzy logic strategy. Challenges in crowd-aware navigation have largely been not explored in classical works. In general, robots have performed navigation in environments of low sophistication. Further, the adaptability has been limited leading to difficulties in handling highly challenging scenarios.

During the last few years, there has been a concerted effort on using contemporary learning models for robot navigation in complex environments. In particular, generative models, attention mechanisms and advanced adaptive methods have been used to handle a wide variety of scenarios in robot navigation.

In the initial stages, VAEs and GANs constituted the primary generative models of interest in robot navigation. VAEs have found applications in self-localization and scenarios involving navigation on plain surfaces. GANs have been useful for predicting next states corresponding to human movements during robot navigation. They have also enabled completing a discontinuous navigation path. In recent years, additional generative models have been proposed. One of them is the normalizing flow-based model which has enabled calculation of exact likelihood of uncertainty in navigation tasks thereby contributing to safety of navigation. Score-based models are also useful in uncertainty estimation thereby enabling safe robot navigation in dynamic environments. A special type of score-based model, named diffusion model, has been useful to generate multiple paths enabling choice of one that is best suited for a mobile robot in point to point navigation.

Two types of attention mechanisms have been predominantly used for robot navigation tasks. The first, based on transformers, is a deep learning model just like convolutional neural networks. However, transformers have the capability to model the relationship between two elements in input data regardless of the distance between them. This feature enables transformers to capture global spatial and temporal dependencies across input features which contributes to enhanced visual perception for effective target-driven robot navigation. The second type of attention mechanism, based on graph attention networks, involves a graph representation where the nodes correspond to humans/robots while the edges correspond to the interactions between robots and humans (or between robots). As a consequence, graph attention networks play a valuable role in socially-aware robot navigation and in multi-robot navigation.

Adaptive approaches have a long history starting with basic reinforcement learning for robot navigation. These have been enhanced and currently, graph neural network-based approaches and deep reinforcement learning play an important role in navigation in a wide range of interesting scenarios. Graph neural networks (along with knowledge graphs) have also played an important role in multi-robot navigation similar to graph attention-based methods. Further, semantic navigation has been specifically addressed via knowledge graphs. Navigation in dynamic environments via deep reinforcement learning has taken off recently in a major way extending some work prior to 2010 (on basic reinforcement learning).

Examination of the efficacy of combining different styles, for example adaptive methods with attention mechanisms, has also been studied with a view to understand performance in unknown environments.

B. FUTURE RESEARCH DIRECTIONS

We have listed potential enhancements to specific works in the tables earlier. In this section, we present some broad directions for further investigations.

1) ROBOTIC NAVIGATION BASED ON SITUATIONAL AWARENESS

The role of deep network models for localization and object detection during navigation has been explored. We have also discussed adaptive methods such as deep reinforcement learning to handle complex environments (such as a maze) for navigational tasks. Another promising direction for further studies is based on examination of cognitive and developmental systems for navigation. In particular, it would be useful to look at cognitive and experience map constructions [197] and take advantage of such representations during navigation. The authors in [197] describe a situational awareness fitting network which helps for navigation. The position of a robot in a path is mapped to a situational awareness value. It would be useful to investigate if attention mechanisms can be combined with the notion of situational awareness to enhance the performance of robot navigation in sophisticated environments.

2) ROBOT NAVIGATION WITH QUANTUM MACHINE LEARNING

Quantum machine learning [198] has been actively researched during the last decade. Efforts on using quantum computing and quantum machine learning to robotics are also underway. The authors in [199] study *quantum deep reinforcement learning* for robot navigation tasks. While the efforts are currently limited to simple systems, this area holds tremendous promise for more detailed studies focusing on complex navigation tasks.

3) OTHER EMERGING PARADIGMS FOR MOBILE ROBOT NAVIGATION

Recently, a method for synthesizing novel views of complex scenes has been proposed in [200]. It is based on representing the scenes as neural radiance fields. Neural radiance fields have been applied to vision-only robot navigation in [201]. It would be of interest to examine if generative models could be combined with scene representation methods to enhance the success of the navigation task.

In the context of approximate computing and when energy efficiency is the goal, the neuromorphic computation paradigm has proven to be useful. This has a natural bearing on mobile robot tasks (since the on-board power support is limited). Neuromorphic hardware accelerators for reinforcement learning to perform obstacle avoidance for mobile robots are being actively researched [202]. Additional investigations in this area particularly in the context of *handling uneven terrains* and deployment of energy-efficient solutions on mobile robots would be valuable.

4) ENVIRONMENTS WITH LIMITED ILLUMINATION

One of the challenges with robot navigation in outdoor environments is variation in lighting. This can happen during different times of the day as well as during different seasons (rainy period or winter with fog for instance). Some of the early approaches [43] relied on stereo vision to get a map which could then be used for navigation. However, challenges have existed with regard to the accuracy of the approaches besides the computational resources (memory in particular). With the advent of contemporary learning methods based on generative models and deep reinforcement learning, there has been significant advancement in this direction. One example is the work in [203] on using generative adversarial networks for visual SLAM in low-light conditions. Further work in this area would be valuable especially to handle a highly cluttered environment.

5) BEYOND SINGLE ROBOT NAVIGATION

We have discussed largely the penetration of various current learning methods into the world of mobile robots operating standalone. However, there are a number of scenarios where multiple robots are involved.

Multiple mobile robots typically rely on a network for communication and accomplishment of the task in a coordinated way. However, in practice, the network may not be reliable or there may be disruption in communication due to the presence of obstacles in the environment. Under these circumstances, learning strategies come in handy. There has been limited work on use of contemporary learning models when multiple robots are involved.

6) HYBRID METHODS FOR ROBOT NAVIGATION

There have also been efforts where the starting point is an adaptive method and generative models have been incorporated to enhance performance. One example is the work on target-driven visual navigation via generative imitation learning [187]. Extension to arbitrary targets and completely unseen environments remains a challenge.

IX. CONCLUSION

This review has been directed to understand the different ways in which contemporary learning approaches based on generative models, attention mechanisms and adaptive methods have been used for robot navigation in complex environments. It has been observed that these approaches have enhanced the success of robot navigation in a variety of highly challenging environments.

ETHICAL APPROVAL

This article does not contain any studies with human participants performed by any authors.

CONSENT TO PARTICIPATE

Not Applicable

CONSENT TO PUBLISH

Not Applicable

AUTHORS CONTRIBUTIONS

All the three authors participated in collection of references. Harinath Sridharan and Nambala Ramsai prepared the initial draft of the article. All the three participated in critique and rewriting.

FUNDING

Not Applicable

COMPETING INTERESTS

The authors declare that they have no competing interests.

AVAILABILITY OF DATA AND MATERIALS

Not Applicable

ACKNOWLEDGMENT

Harinath Sridharan was with the School of Mechanical Engineering, SASTRA Deemed University, Thanjavur, India. Bhaskar Vundurthy was with the Robotics Laboratory, Electrical Engineering Department, Indian Institute of Technology Madras.

REFERENCES

[1] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Cambridge, MA, USA: MIT Press, 2004.

[2] H. Choset, K. M. Lynch, S. Hutchinson, G. A. Kantor, W. Burgard, L. E. Kavraki and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. Cambridge, MA, USA: MIT Press, 2007.

[3] R. Keith and H. Manh La, "Review of autonomous mobile robots for the warehouse environment," 2024, *arXiv:2406.08333*.

[4] M. Paolanti, L. Romeo, M. Martini, A. Mancini, E. Frontoni, and P. Zingaretti, "Robotic retail surveying by deep learning visual and textual data," *Robot. Auto. Syst.*, vol. 118, pp. 179–188, Aug. 2019.

[5] B. Ko, H.-J. Choi, C. Hong, J.-H. Kim, O. Chul Kwon, and C. D. Yoo, "Neural network-based autonomous navigation for a homecare mobile robot," in *Proc. IEEE Int. Conf. Big Data Smart Comput. (BigComp)*, Feb. 2017, pp. 403–406.

[6] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: A survey," *Auto. Robots*, vol. 46, no. 5, pp. 569–597, Jun. 2022.

[7] Y. Zhou, S. Zhu, W. Song, J. Gu, J. Ren, X. Xi, T. Jin, and Z. Mu, "Robot planning based on behavior tree and knowledge graph," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2022, pp. 827–832.

[8] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[9] S. J. D. Prince, *Understanding Deep Learning*. Cambridge, MA, USA: MIT Press, 2023.

[10] C. M. Bishop and H. Bishop, *Deep Learning : Foundations and Concepts*. Cham, Switzerland: Springer, 2024.

[11] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, *Dive Into Deep Learning*. Cambridge, U.K.: Cambridge Univ. Press, 2023.

[12] R. O'Reilly, D. Wyatte, S. Herd, B. Mingus, and D. Jilk, "Recurrent processing during object recognition," *Frontiers Psychol.*, vol. 4, p. 124, Jul. 2013.

[13] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 30, Jun. 2017, pp. 5998–6008.

[14] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–22.

[15] S. Poddar, C. Mavrogiannis, and S. S. Srinivasa, "From crowd motion prediction to robot navigation in crowds," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 6765–6772.

[16] P. Mishra, U. Jain, S. Choudhury, S. Singh, A. Pandey, A. Sharma, R. Singh, V. K. Pathak, K. K. Saxena, and A. Gehlot, "Footstep planning of humanoid robot in ROS environment using generative adversarial networks (GANs) deep learning," *Robot. Auto. Syst.*, vol. 158, Sep. 2022, Art. no. 104269.

[17] Z. Feng, H. Luan, P. Goyal, and H. Soh, "LTLDoG: Satisfying temporally-extended symbolic constraints for safe diffusion-based planning," *IEEE Robot. Autom. Lett.*, vol. 9, no. 10, pp. 8571–8578, Oct. 2024.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2015, pp. 1–15.

[19] B. Hassan, N. A. Madjid, F. Kashwani, M. Alansari, M. Khonji, and J. Dias, "PathFormer: A transformer-based framework for vision-centric autonomous navigation in off-road environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2024, pp. 7718–7725.

[20] W. Huang, Y. Zhou, X. He, and C. Lv, "Goal-guided transformer-enabled reinforcement learning for efficient autonomous navigation," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 1832–1845, Feb. 2024.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Peterson, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Playing Atari with deep reinforcement learning," in *Proc. NIPS Deep Learn. Workshop*, Jan. 2013, pp. 1–9.

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[24] H. Le, S. Saeedvand, and C.-C. Hsu, "A comprehensive review of mobile robot navigation using deep reinforcement learning algorithms in crowded environments," *J. Intell. Robotic Syst.*, vol. 110, no. 4, pp. 1–22, Nov. 2024.

[25] H. Jiang, H. Wang, W.-Y. Yau, and K.-W. Wan, "A brief survey: Deep reinforcement learning in mobile robot navigation," in *Proc. 15th IEEE Conf. Ind. Electron. Appl. (ICIEA)*, Nov. 2020, pp. 592–597.

[26] D. Damjanović, P. Biočić, S. Prakljačić, D. Činčurak, and J. Balen, "A comprehensive survey on SLAM and machine learning approaches for indoor autonomous navigation of mobile robots," *Mach. Vis. Appl.*, vol. 36, no. 3, pp. 1–21, May 2025.

[27] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] D. Maravall, J. De Lope, and F. Serradilla, "Combination of model-based and reactive methods in autonomous navigation," in *Proc. ICRA. Millennium Conf. IEEE Int. Conf. Robot. Autom. Symposia*, vol. 3, Jun. 2000, pp. 2328–2333.

[29] H. Seraji and A. Howard, "Behaviour-based robot navigation on challenging terrain: A fuzzy logic approach," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 308–321, Mar. 2002.

[30] H. Maaref and C. Barret, "Sensor-based navigation of a mobile robot in an indoor environment," *Robot. Auto. Syst.*, vol. 38, no. 1, pp. 1–18, Jan. 2002.

[31] J. Guivant, E. Nebot, J. Nieto, and F. Masson, "Navigation and mapping in large unstructured environments," *Int. J. Robot. Res.*, vol. 23, nos. 4–5, pp. 449–472, Apr. 2004.

[32] P. Althaus, H. Ishiguro, T. Kanda, T. Miyashita, and H. I. Christensen, "Navigation for human–robot interaction tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 2, Nov. 2004, pp. 1894–1900.

[33] C. Castejon, D. Blanco, B. L. Boada, and L. Moreno, "Voronoi extraction of free-way areas in cluttered environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Jun. 2005, pp. 406–411.

[34] X. Zhuang, "The strategy entropy of reinforcement learning for mobile robot navigation in complex environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, Nov. 2005, pp. 1742–1747.

[35] A. Zhu and S. X. Yang, "Neurofuzzy-based approach to mobile robot navigation in unknown environments," *IEEE Trans. Syst., Man Cybern., C, Appl. Rev.*, vol. 37, no. 4, pp. 610–621, Jul. 2007.

[36] J.-H. Park, S.-H. Baeg, and M.-H. Baeg, "An intelligent navigation method for service robots in the smart environment," in *Proc. Int. Conf. Control, Autom. Syst.*, 2007, pp. 494–499.

[37] M. F. Selekwa, D. D. Dunlap, D. Shi, and E. G. Collins, "Robot navigation in very cluttered environments by preference-based fuzzy behaviors," *Robot. Auto. Syst.*, vol. 56, no. 3, pp. 231–246, Mar. 2008.

[38] M. Wang and J. N. K. Liu, "Fuzzy logic-based real-time robot navigation in unknown environment with dead ends," *Robot. Auto. Syst.*, vol. 56, no. 7, pp. 625–643, Jul. 2008.

[39] A. Kokosy, F.-O. Defaux, and W. Perruquetti, "Autonomous navigation of a nonholonomic mobile robot in a complex environment," in *Proc. IEEE Int. Workshop Saf., Secur. Rescue Robot.*, Oct. 2008, pp. 102–108.

[40] J. Carff, M. Johnson, E. M. El-Sheikh, and J. E. Pratt, "Human–robot team navigation in visually complex environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 3043–3050.

[41] Q.-Y. Bao, S.-M. Li, W.-Y. Shang, and M.-J. An, "A fuzzy behavior-based architecture for mobile robot navigation in unknown environments," in *Proc. Int. Conf. Artif. Intell. Comput. Intell.*, vol. 2, Nov. 2009, pp. 257–261.

[42] E. Ayari, S. Hadouaj, and K. Ghedira, "A fuzzy logic method for autonomous robot navigation in dynamic and uncertain environment composed with complex traps," in *Proc. 5th Int. Multi-conference Comput. Global Inf. Technol.*, Sep. 2010, pp. 18–23.

[43] K. Irie, T. Yoshida, and M. Tomono, "Mobile robot localization using stereo vision in outdoor environments under various illumination conditions," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 5175–5181.
[44] C.-P. Lam, C.-T. Chou, K.-H. Chiang, and L.-C. Fu, "Human-centered robot navigation—Towards a harmoniously human–robot coexisting environment," *IEEE Trans. Robot.*, vol. 27, no. 1, pp. 99–112, Feb. 2011.
[45] M. A. Kareem Jaradat, M. Al-Rousan, and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Robot. Comput.-Integr. Manuf.*, vol. 27, no. 1, pp. 135–149, Feb. 2011.
[46] K. Qian and A. Song, "Autonomous navigation for a mobile robot based on a sonar ring and its implementation," in *Proc. 8th IEEE Int. Symp. Instrum. Control Technol. (ISICT)*, Oct. 2012, pp. 1–4.
[47] X. Ren, J. Wang, Z. Cai, and M. Gu, "Navigation and measurement control in complex environment," in *Proc. 2nd Int. Conf. Instrum., Meas., Comput., Commun. Control*, Dec. 2012, pp. 280–284.
[48] T. Zhao and Y. Wang, "A neural-network based autonomous navigation system using mobile robots," in *Proc. 12th Int. Conf. Control Autom. Robot. Vis. (ICARCV)*, Dec. 2012, pp. 1101–1106.
[49] H. A. Pierson and M. S. Gashler, "Deep learning in robotics: A review of recent research," *Adv. Robot.*, vol. 31, no. 16, pp. 821–835, Aug. 2017.
[50] S. Cebollada, L. Payá, M. Flores, A. Peidró, and O. Reinoso, "A state-of-the-art review on mobile robotics tasks using artificial intelligence and visual data," *Expert Syst. Appl.*, vol. 167, Apr. 2021, Art. no. 114195, doi: 10.1016/j.eswa.2020.114195.
[51] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1238–1274, Sep. 2013.
[52] W. Chen, T. Qu, Y. Zhou, K. Weng, G. Wang, and G. Fu, "Door recognition and deep learning algorithm for visual based robot navigation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2014, pp. 1793–1798.
[53] D. H. Dos Reis, D. Welfer, M. A. De Souza Leite Cuadros, and D. F. T. Gamarra, "Mobile robot navigation using an object recognition software with RGBD images and the YOLO algorithm," *Appl. Artif. Intell.*, vol. 33, no. 14, pp. 1290–1305, Dec. 2019.
[54] H. Sinha, J. Patrikar, E. G. Dhekane, G. Pandey, and M. Kothari, "Convolutional neural network based sensors for mobile robot relocalization," in *Proc. 23rd Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2018, pp. 774–779.
[55] F. Vulpi, A. Milella, R. Marani, and G. Reina, "Recurrent and convolutional neural networks for deep terrain classification by autonomous robots," *J. Terramechanics*, vol. 96, pp. 119–131, Aug. 2021, doi: 10.1016/j.jterra.2020.12.002.
[56] A. Milioto, P. Lottes, and C. Stachniss, "Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in CNNs," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 2229–2235.
[57] C. McCool, T. Perez, and B. Upcroft, "Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1344–1351, Jul. 2017.
[58] M. Kouzehgar, Y. K. Tamilselvam, M. V. Heredia, and M. R. Elara, "Self-reconfigurable façade-cleaning robot equipped with deep-learning-based crack detection based on convolutional neural networks," *Autom. Construct.*, vol. 108, Dec. 2019, Art. no. 102959.
[59] R. Terasawa, Y. Ariki, T. Narihira, T. Tsuboi, and K. Nagasaka, "3D-CNN based heuristic guided task-space planner for faster motion planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 9548–9554.
[60] V. Suryamurthy, V. S. Raghavan, A. Laurenzi, N. G. Tsagarakis, and D. Kanoulas, "Terrain segmentation and roughness estimation using RGB data: Path planning application on the centauro robot," in *Proc. IEEE-RAS 19th Int. Conf. Humanoid Robots*, Oct. 2019, pp. 289–296.
[61] A. Pajaziti, X. Bajrami, and A. Paliqi, "Path control of quadruped robot through convolutional neural networks," *IFAC-PapersOnLine*, vol. 51, no. 30, pp. 610–615, 2018.
[62] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, "A novel GRU-RNN network model for dynamic path planning of mobile robot," *IEEE Access*, vol. 7, pp. 15140–15151, 2019.
[63] M. H. Woo, S.-H. Lee, and H. M. Cha, "A study on the optimal route design considering time of mobile robot using recurrent neural network and reinforcement learning," *J. Mech. Sci. Technol.*, vol. 32, no. 10, pp. 4933–4939, Oct. 2018.
[64] S. Otte, C. Weiss, T. Scherer, and A. Zell, "Recurrent neural networks for fast and robust vibration-based ground classification on mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5603–5608.
[65] V. N. Ferreira, S. R. J. Neto, A. C. Almeida, and R. A. D. C. Bianchi, "A visual memory system for humanoid robots," in *Proc. Latin Amer. Robotic Symp., Brazilian Symp. Robot. (SBR) Workshop Robot. Educ. (WRE)*, Nov. 2018, pp. 426–431.
[66] N. G. Shakev, S. A. Ahmed, V. L. Popov, and A. V. Topalov, "Recognition and following of dynamic targets by an omnidirectional mobile robot using a deep convolutional neural network," in *Proc. Int. Conf. Intell. Syst. (IS)*, Sep. 2018, pp. 589–594.
[67] K. Koide, J. Miura, and E. Menegatti, "Monocular person tracking and identification with on-line deep feature selection for person following robots," *Robot. Auto. Syst.*, vol. 124, Feb. 2020, Art. no. 103348.
[68] M. F. Kazerouni, N. T. M. Saeed, and K.-D. Kuhnert, "Exploration of autonomous mobile robots through challenging outdoor environments for natural plant recognition using deep neural network," in *Proc. IEEE 15th Int. Conf. Intell. Comput. Commun. Process. (ICCP)*, Sep. 2019, pp. 279–285.
[69] C. McCool, J. Beattie, J. Firn, C. Lehnert, J. Kulk, O. Bawden, R. Russell, and T. Perez, "Efficacy of mechanical weeding tools: A study into alternative weed management strategies enabled by robotics," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 1184–1190, Apr. 2018.
[70] F. Garibaldi-Márquez, D. A. Martínez-Barba, L. E. Montañez-Franco, G. Flores, and L. M. Valentín-Coronado, "Enhancing site-specific weed detection using deep learning transformer architectures," *Crop Protection*, vol. 190, Apr. 2025, Art. no. 107075.
[71] A. Abe, G. Pongthanisorn, S.-I. Kaneko, and G. Capi, "Enhancing robot self-localization accuracy and robustness: A variational autoencoder-based approach," in *Proc. 5th Int. Conf. Control Robot. (ICCR)*, Nov. 2023, pp. 145–149.
[72] S. Wakita, T. Nakamura, and H. Hachiya, "Laser variational autoencoder for map construction and self-localization," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2018, pp. 3993–3998.
[73] A. Dastider and M. Lin, "DAMON: Dynamic amorphous obstacle navigation using topological manifold learning and variational autoencoding," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 251–258.
[74] K. Hu and P. O'Connor, "Learning a representation map for robot navigation using deep variational autoencoder," 2018, *arXiv:1807.02401*.
[75] J. de Heuvel, N. Corral, B. Kreis, J. Conradi, A. Driemel, and M. Bennewitz, "Learning depth vision-based personalized robot navigation from dynamic demonstrations in virtual reality," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 6757–6764.
[76] L. Puck, M. Schik, T. Schnell, T. Buettner, A. Roennau, and R. Dillmann, "AutoExplorers: Autoencoder-based strategies for high-entropy exploration in unknown environments for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 16648–16654.
[77] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.
[78] K. K. Borkar, M. K. Singh, R. K. Dasari, A. Babbar, A. Pandey, U. Jain, and P. Mishra, "Path planning design for a wheeled robot: A generative artificial intelligence approach," *Int. J. Interact. Design Manuf.*, vol. 19, no. 2, pp. 1–12, Feb. 2025.
[79] H. Li, D.-H. Qian, G.-Y. Liu, Z. Cui, and J.-T. Lei, "Pedestrian trajectory prediction based on SOPD-GAN used for the trajectory planning and motion control of mobile robot," *IEEE Access*, vol. 11, pp. 131376–131393, 2023.
[80] T. N. Canh, X. HoangVan, and N. Y. Chong, "Enhancing social robot navigation with integrated motion prediction and trajectory planning in dynamic human environments," in *Proc. 24th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2024, pp. 731–736.
[81] W. Tang, Y. Zhou, H. Sun, Y. Zhang, Y. Liu, Z. Ding, J. Liu, and J. He, "GAN-based robust motion planning for mobile robots against localization attacks," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1603–1610, Mar. 2023.
[82] J. Chen, Z. Chen, M. Fang, J. Li, Z. Ming, and S. Wang, "A GAN-based active terrain mapping for collaborative air-ground robotic system," in *Proc. IEEE 4th Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Jul. 2019, pp. 622–627.
[83] J. Maldonado-Romo, M. A. Perez, and A. R. Molina, "Path planning generator with metadata through a domain change by GAN between physical and virtual environment," *Sensors*, vol. 21, p. 7667, Jun. 2021.
[84] I. Goodfellow, J. P. Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[85] K. Matsumoto, Y. Hyodo, and R. Kurazume, "Crowd-aware robot navigation with switching between learning-based and rule-based methods using normalizing flows," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2024, pp. 4823–4830.
[86] L. Wellhausen, R. Ranftl, and M. Hutter, "Safe robot navigation via multi-modal anomaly detection," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1326–1333, Apr. 2020.
[87] J. Carvalho, M. Baeirl, J. Urain, and J. Peters, "Conditioned score-based models for learning collision-free trajectory generation," in *Proc. NeurIPS Workshop Score-Based Models*, 2022, pp. 1–10.
[88] Q. Huang, C. Pu, D. Fourie, K. Khosoussi, J. P. How, and J. J. Leonard, "NF-iSAM: Incremental smoothing and mapping via normalizing flows," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 1095–1102.
[89] Q. Huang, C. Pu, K. Khosoussi, D. M. Rosen, D. Fourie, J. P. How, and J. J. Leonard, "Incremental non-Gaussian inference for SLAM using normalizing flows," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1458–1475, Apr. 2023.
[90] A. Hao Tan, A. Fung, H. Wang, and G. Nejat, "Mobile robot navigation using hand-drawn maps: A vision language model approach," 2025, *arXiv:2502.00114*.
[91] T. Lai and F. Ramos, "PlannerFlows: Learning motion samplers with normalising flows," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 2542–2548.
[92] B. Mazoure, T. Doan, A. Durand, J. Pineau, and R. D. Hjelm, "Leveraging exploration in off-policy algorithms via normalizing flows," in *Proc. 3rd Conf. Robot Learn. (CoRL)*, Jan. 2019, pp. 1–15.
[93] D. Jia, A. Hermans, and B. Leibe, "DR-SPAAM: A spatial-attention and auto-regressive model for person detection in 2D range data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10270–10277.
[94] J. V. D. Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal n-body collision avoidance," in *Proc. 14th Int. Symp. Robot. Res.*, Jan. 2011, pp. 3–19.
[95] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[96] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, J. Zhu, and L. Zhang, "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," 2023, *arXiv:2303.05499*.
[97] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang, "Grounded SAM: Assembling open-world models for diverse visual tasks," 2024, *arXiv:2401.14159*.
[98] A. Sridhar, D. Shah, C. Glossop, and S. Levine, "NoMaD: Goal masked diffusion policies for navigation and exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 63–70.
[99] J. Liang, A. Payandeh, D. Song, X. Xiao, and D. Manocha, "DTG : Diffusion-based trajectory generation for mapless global navigation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2024, pp. 5340–5347.
[100] J. Liu, M. Stamatopoulou, and D. Kanoulas, "DiPPeR: Diffusion-based 2D path planner applied on legged robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 9264–9270.
[101] J. Carvalho, A. T. Le, M. Baierl, D. Koert, and J. Peters, "Motion planning diffusion: Learning and planning of robot motions with diffusion models," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 1916–1923.
[102] A. Bar, G. Zhou, D. Tran, T. Darrell, and Y. LeCun, "Navigation world models," 2024, *arXiv:2412.03572*.
[103] S. Samavi, A. Lem, F. Sato, S. Chen, Q. Gu, K. Yano, A. P. Schoellig, and F. Shkurti, "Safe diffusion model predictive control for interactive robotic crowd navigation," in *Proc. Workshop Safe Robust Robot Learn. Oper. Real World*, 2024, pp. 1–12.
[104] W. Yu, J. Peng, H. Yang, J. Zhang, Y. Duan, J. Ji, and Y. Zhang, "LDP: A local diffusion planner for efficient robot navigation and collision avoidance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2024, pp. 5466–5472.
[105] Y. Wu, J. Li, and J. Ji, "A diffusion model augmented indoor robot local navigation planner based on decision transformer," in *Proc. IEEE 4th Int. Conf. Softw. Eng. Artif. Intell. (SEAI)*, Jun. 2024, pp. 276–282.
[106] Y. Shaoul, I. Mishani, S. Vats, J. Li, and M. Likhachev, "Multi-robot motion planning with diffusion models," 2024, *arXiv:2410.03072*.
[107] D. Shah, A. Sridhar, A. Bhorkar, N. Hirose, and S. Levine, "GNM: A general navigation model to drive any robot," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 7226–7233.
[108] H. He, H. Fu, Q. Wang, S. Zhou, W. Liu, and Y. Chen, "Spatio-temporal transformer-based reinforcement learning for robot crowd navigation," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2023, pp. 1–7.
[109] W. Wang, R. Wang, L. Mao, and B.-C. Min, "NaviSTAR: Socially aware robot navigation with hybrid spatio-temporal graph transformer and preference learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 11348–11355.
[110] Y. Yang, J. Jiang, J. Zhang, J. Huang, and M. Gao, "ST^2: Spatial–temporal state transformer for crowd-aware autonomous navigation," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 912–919, Feb. 2023.
[111] J. Liu, S. Lyu, D. Hadjivelichkov, V. Modugno, and D. Kanoulas, "ViT—A*: Legged robot path planning using vision transformer A*," in *Proc. IEEE-RAS 22nd Int. Conf. Humanoid Robots (Humanoids)*, Dec. 2023, pp. 1–6.
[112] A. Magassouba, K. Sugiura, and H. Kawai, "CrossMap transformer: A crossmodal masked path transformer using double back-translation for vision-and-language navigation," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6258–6265, Oct. 2021.
[113] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[114] Q. Mao, J. Wang, M. Xu, and Z. Chen, "NavTr: Object-goal navigation with learnable transformer queries," *IEEE Robot. Autom. Lett.*, vol. 9, no. 12, pp. 11738–11745, Dec. 2024.
[115] X. Li, S. Zhang, Y. Lu, K. Dang, and L. Ran, "Zero-shot object goal visual navigation with class-independent relationship network," 2023, *arXiv:2310.09883*.
[116] Z. Seymour, K. Thopalli, N. Mithun, H.-P. Chiu, S. Samarasekera, and R. Kumar, "MaAST: Map attention with semantic transformers for efficient visual navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 13223–13230.
[117] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied AI research," 2019, *arXiv:1904.01201*.
[118] Y. He and K. Zhou, "Relation-wise transformer network and reinforcement learning for visual navigation," *Neural Comput. Appl.*, vol. 36, no. 21, pp. 13205–13221, Jul. 2024.
[119] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comp. Vis.*, Jan. 2020, pp. 213–229.
[120] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. Van Den Hengel, "Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments," 2017, *arXiv:1711.07280*.
[121] R. Wang, Y. Liu, Y. Cao, and L. Yao, "Causality-aware transformer networks for robotic navigation," 2024, *arXiv:2409.02669*.
[122] H. Wang, A. H. Tan, and G. Nejat, "NavFormer: A transformer architecture for robot target-driven navigation in unknown and dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 8, pp. 6808–6815, Aug. 2024.
[123] D. Lawson and A. H. Qureshi, "Control transformer: Robot navigation in unknown environments through PRM-guided return-conditioned sequence modeling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 9324–9331.
[124] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin, "Emerging properties in self-supervised vision transformers," 2021, *arXiv:2104.14294*.
[125] M. Saavedra-Ruiz, S. Morin, and L. Paull, "Monocular robot navigation with self-supervised pretrained vision transformers," in *Proc. 19th Conf. Robots Vis. (CRV)*, May 2022, pp. 197–204.
[126] L. Chen, Y. Wang, Z. Miao, Y. Mo, M. Feng, Z. Zhou, and H. Wang, "Transformer-based imitative reinforcement learning for multirobot path planning," *IEEE Trans. Ind. Informat.*, vol. 19, no. 10, pp. 10233–10243, Oct. 2023.
[127] Z. Ma, Y. Luo, and J. Pan, "Learning selective communication for multi-agent path finding," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 1455–1462, Apr. 2022.
[128] G. Sartoretti, J. Kerr, Y. Shi, G. Wagner, T. K. S. Kumar, S. Koenig, and H. Choset, "PRIMAL: Pathfinding via reinforcement and imitation multi-agent learning," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2378–2385, Jul. 2019.

[129] T. Guan, D. Kothandaraman, R. Chandra, A. J. Sathyamoorthy, K. Weerakoon, and D. Manocha, "GA-nav: Efficient terrain segmentation for robot navigation in unstructured outdoor environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 8138–8145, Jul. 2022.

[130] W. Li, R. Hong, J. Shen, L. Yuan, and Y. Lu, "Transformer memory for interactive visual navigation in cluttered environments," *IEEE Robot. Autom. Lett.*, vol. 8, no. 3, pp. 1731–1738, Mar. 2023.

[131] Y. Chen, C. Liu, B. E. Shi, and M. Liu, "Robot navigation in crowds by graph convolutional networks with attention learned from human gaze," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2754–2761, Apr. 2020.

[132] Q. Li, W. Lin, Z. Liu, and A. Prorok, "Message-aware graph attention networks for large-scale multi-robot path planning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 3, pp. 5533–5540, Jul. 2021.

[133] T. Zhang, T. Qiu, Z. Pu, Z. Liu, and J. Yi, "Robot navigation among external autonomous agents through deep reinforcement learning using graph attention network," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9465–9470, 2020.

[134] X. He, X. Shi, J. Hu, and Y. Wang, "Multi-robot navigation with graph attention neural network and hierarchical motion planning," *J. Intell. Robotic Syst.*, vol. 109, no. 2, pp. 1–12, Oct. 2023.

[135] X. Lu, A. Faragasso, Y. Wang, A. Yamashita, and H. Asama, "Group-aware robot navigation in crowds using spatio-temporal graph attention network with deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 10, no. 4, pp. 4140–4147, Apr. 2025.

[136] C. Chen, Y. Liu, S. Kreiss, and A. Alahi, "Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 6015–6022.

[137] M. Lingelbach, C. Li, M. Hwang, A. Kurenkov, A. Lou, R. Martín-Martín, R. Zhang, L. Fei-Fei, and J. Wu, "Task-driven graph attention for hierarchical relational object navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 886–893.

[138] S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, D. L. McPherson, J. Geng, and K. Driggs-Campbell, "Intention aware robot crowd navigation with attention-based interaction graph," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 12015–12021.

[139] E. Escudie, L. Matignon, and J. Saraydaryan, "Attention graph for multi-robot social navigation with deep reinforcement learning," 2024, *arXiv:2401.17914*.

[140] Z. Yu, H. Guo, A. H. Adiwahono, J. Chan, B. S. W. Tynn, C.-M. Chew, and W.-Y. Yau, "Multi-robot reliable navigation in uncertain topological environments with graph attention networks," 2024, *arXiv:2411.16134*.

[141] C. Sun, X. Wu, Y. Wang, and C. Sun, "Attention-based value classification reinforcement learning for collision-free robot navigation," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 11, pp. 6898–6911, Nov. 2024, doi: 10.1109/TIV.2024.3391007.

[142] F. Ou, Y. Li, and Z. Miao, "Place recognition of large-scale unstructured orchards with attention score maps," *IEEE Robot. Autom. Lett.*, vol. 8, no. 2, pp. 958–965, Feb. 2023.

[143] J. de Heuvel, X. Zeng, W. Shi, T. Sethuraman, and M. Bennewitz, "Spatiotemporal attention enhances LiDAR-based robot navigation in dynamic environments," *IEEE Robot. Autom. Lett.*, vol. 9, no. 5, pp. 4202–4209, May 2024.

[144] Y. Wei, X. Sun, Q. Zhang, and L. Xu, "Attention-based deep reinforcement learning approach for robot navigation in dynamic environments," in *Proc. 36th Chin. Control Decis. Conf. (CCDC)*, May 2024, pp. 1832–1837.

[145] S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu, "A survey on knowledge graphs: Representation, acquisition, and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 2, pp. 494–514, Feb. 2022.

[146] L. Lin and F. Wang, "Adaptive learning system based on knowledge graph," in *Proc. 9th Int. Conf. Educ. Training Technol.*, Apr. 2023, pp. 1–7.

[147] S. Zhang, X. Wang, Y. Ma, and D. Wang, "An adaptive learning method based on knowledge graph," *Frontiers Educ. Res.*, vol. 6, no. 6, pp. 112–115, 2023.

[148] M. K. Bergman. (2019). *A Common Sense View of Knowledge Graphs*. Adapt. Inf., Adapt. Innov., Adapt. Infrastructure. [Online]. Available: http://www.mkbergman.com/2244

[149] B. Fang, G. Mei, X. Yuan, L. Wang, Z. Wang, and J. Wang, "Visual SLAM for robot navigation in healthcare facility," *Pattern Recognit.*, vol. 113, May 2021, Art. no. 107822.

[150] A. Kattepur and P. Balamuralidhar, "RoboPlanner: Autonomous robotic action planning via knowledge graph queries," in *Proc. 34th ACM/SIGAPP Symp. Appl. Comput.*, Apr. 2019, pp. 953–956.

[151] A. Khan, A. Ribeiro, V. Kumar, and A. G. Francis, "Graph neural networks for motion planning," 2020, *arXiv:2006.06248*.

[152] R. Zhang, C. Yu, J. Chen, C. Fan, and S. Gao, "Learning-based motion planning in dynamic environments using GNNs and temporal encoding," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 30003–30015.

[153] W. Liu, K. Eltouny, S. Tian, X. Liang, and M. Zheng, "KG-planner: Knowledge-informed graph neural planning for collaborative manipulators," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 5061–5071, 2025.

[154] A. Daruna, M. Gupta, M. Sridharan, and S. Chernova, "Continual learning of knowledge graph embeddings," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1128–1135, Apr. 2021.

[155] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 11785–11792.

[156] X. Ji, H. Li, Z. Pan, X. Gao, and C. Tu, "Decentralized, unlabeled multi-agent navigation in obstacle-rich environments using graph neural networks," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2021, pp. 8936–8943.

[157] C. Yu, H. Yu, and S. Gao, "Learning control admissibility models with graph neural networks for multi-agent navigation," in *Proc. 6th Conf. Robot Learn. (CoRL)*, Jan. 2022, pp. 1–12.

[158] Y. Li, Y. Ma, X. Huo, and X. Wu, "Remote object navigation for service robots using hierarchical knowledge graph in human-centered environments," *Intell. Service Robot.*, vol. 15, no. 4, pp. 459–473, Sep. 2022.

[159] B. Tang, X. Huang, Y. Ma, H. Yu, L. Tang, Z. Lin, D. Zhu, and X. Qin, "Multi-source fusion of substation intelligent inspection robot based on knowledge graph: A overview and roadmap," *Frontiers Energy Res.*, vol. 10, pp. 1–12, Aug. 2022.

[160] N. Xu, W. Wang, R. Yang, M. Qin, Z. Lin, W. Song, C. Zhang, J. Gu, and C. Li, "Aligning knowledge graph with visual perception for object-goal navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 5214–5220.

[161] C.-T. Chen, S. J. Koh, F.-H. Chang, Y.-S. Huang, and L.-C. Fu, "Object-goal navigation of home care robot based on human activity inference and cognitive memory," *IEEE Trans. Hum.-Mach. Syst.*, vol. 54, no. 6, pp. 808–817, Dec. 2024.

[162] K. Zhou, C. Guo, W. Guo, and H. Zhang, "Learning heterogeneous relation graph and value regularization policy for visual navigation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 11, pp. 16901–16915, Nov. 2024.

[163] S. Wang, X. Mao, S. Yang, M. Wu, and Z. Zhang, "ROS package search for robot software development: A knowledge graph-based approach," *Frontiers Comput. Sci.*, vol. 19, no. 6, Jun. 2025, Art. no. 196320.

[164] Z. Wang and G. Tian, "Goal-oriented visual semantic navigation using semantic knowledge graph and transformer," *IEEE Trans. Autom. Sci. Eng.*, vol. 22, pp. 1647–1657, 2025.

[165] P. Bachiller, D. Rodriguez-Criado, R. R. Jorvekar, P. Bustos, D. R. Faria, and L. J. Manso, "A graph neural network to model disruption in human-aware robot navigation," *Multimedia Tools Appl.*, vol. 81, no. 3, pp. 3277–3295, Jan. 2022.

[166] M. Everett, Y. F. Chen, and J. P. How, "Collision avoidance in pedestrian-rich environments with deep reinforcement learning," *IEEE Access*, vol. 9, pp. 10357–10377, 2021.

[167] T. Fan, X. Cheng, J. Pan, P. Long, W. Liu, R. Yang, and D. Manocha, "Getting robots unfrozen and unlost in dense pedestrian crowds," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1178–1185, Apr. 2019.

[168] L. Liu, D. Dugas, G. Cesari, R. Siegwart, and R. Dubé, "Robot navigation in crowded environments using deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5671–5677.

[169] Z. Liu, Y. Zhai, J. Li, G. Wang, Y. Miao, and H. Wang, "Graph relational reinforcement learning for mobile robot navigation in large-scale crowded environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 8, pp. 8776–8787, Aug. 2023.

[170] U. Patel, N. K. S. Kumar, A. J. Sathyamoorthy, and D. Manocha, "DWA-RL: Dynamically feasible deep reinforcement learning policy for robot navigation among mobile obstacles," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 6057–6063.

[171] S. Gangapurwala, A. Mitchell, and I. Havoutis, "Guided constrained policy optimization for dynamic quadrupedal robot locomotion," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3642–3649, Apr. 2020.

[172] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "DeepGait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 3699–3706, Apr. 2020.
[173] T. Wang, X. Du, M. Chen, and K. Li, "Hierarchical relational graph learning for autonomous multirobot cooperative navigation in dynamic environments," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 42, no. 11, pp. 3559–3570, Nov. 2023.
[174] T. Fan, P. Long, W. Liu, and J. Pan, "Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 856–892, Jun. 2020.
[175] G. Chen, L. Pan, Y. Chen, P. Xu, Z. Wang, P. Wu, J. Ji, and X. Chen, "Robot navigation with map-based deep reinforcement learning," in *Proc. IEEE Int. Conf. Netw., Sens. Control (ICNSC)*, Oct. 2020, pp. 1–6.
[176] H. Quan, Y. Li, and Y. Zhang, "A novel mobile robot navigation method based on deep reinforcement learning," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 3, pp. 1–11, May 2020.
[177] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5129–5136.
[178] H. Hu, K. Zhang, A. H. Tan, M. Ruan, C. Agia, and G. Nejat, "A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6569–6576, Oct. 2021.
[179] K. Zhang, F. Niroui, M. Ficocelli, and G. Nejat, "Robot navigation of environments with unknown rough terrain using deep reinforcement learning," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Aug. 2018, pp. 1–7.
[180] L. Kästner, C. Marx, and J. Lambrecht, "Deep-reinforcement-learning-based semantic navigation of mobile robots in dynamic environments," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2020, pp. 1110–1115.
[181] A. Staroverov, D. A. Yudin, I. Belkin, V. Adeshkin, Y. K. Solomentsev, and A. I. Panov, "Real-time object navigation with deep neural networks and hierarchical reinforcement learning," *IEEE Access*, vol. 8, pp. 195608–195621, 2020.
[182] H. Shi, L. Shi, M. Xu, and K.-S. Hwang, "End-to-end navigation strategy with deep reinforcement learning for mobile robots," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2393–2402, Apr. 2020.
[183] R. Raj and A. Kos, "Intelligent mobile robot navigation in unknown and complex environment using reinforcement learning technique," *Sci. Rep.*, vol. 14, no. 1, p. 22852, Oct. 2024.
[184] X. Ruan, D. Ren, X. Zhu, and J. Huang, "Mobile robot navigation based on deep reinforcement learning," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2019, pp. 6174–6178.
[185] J. Wang, S. Elfwing, and E. Uchibe, "Modular deep reinforcement learning from reward and punishment for robot navigation," *Neural Netw.*, vol. 135, pp. 115–126, Mar. 2021, doi: 10.1016/j.neunet.2020.12.001.
[186] Y. Yin, Z. Chen, G. Liu, J. Yin, and J. Guo, "Autonomous navigation of mobile robots in unknown environments using off-policy reinforcement learning with curriculum learning," *Expert Syst. Appl.*, vol. 247, Aug. 2024, Art. no. 123202.
[187] Q. Wu, X. Gong, K. Xu, D. Manocha, J. Dong, and J. Wang, "Towards target-driven visual navigation in indoor scenes via generative imitation learning," *IEEE Robot. Autom. Lett.*, vol. 6, no. 1, pp. 175–182, Jan. 2021.
[188] M. Dobrevski and D. Skočaj, "Deep reinforcement learning for map-less goal-driven robot navigation," *Int. J. Adv. Robotic Syst.*, vol. 18, no. 1, pp. 1–13, Jan. 2021.
[189] M. Babaeizadeh, I. Frosio, S. Tyree, J. Clemons, and J. Kautz, "Reinforcement learning through asynchronous advantage actor-critic on a GPU," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Jan. 2016, pp. 1–8.
[190] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Autom. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.
[191] J. Ahn, J. Lee, and L. Sentis, "Data-efficient and safe learning for humanoid locomotion aided by a dynamic balancing model," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4376–4383, Jul. 2020.
[192] H. Bavle, J. L. Sanchez-Lopez, M. Shaheer, J. Civera, and H. Voos, "Situational graphs for robot navigation in structured indoor environments," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 9107–9114, Oct. 2022.
[193] M. Shaheer, J. A. Millan-Romera, H. Bavle, J. L. Sanchez-Lopez, J. Civera, and H. Voos, "Graph-based global robot localization informing situational graphs with architectural graphs," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2023, pp. 9155–9162.
[194] S. Schall, "Learning from demonstration," in *Proc. Adv. Neural Inf. Process. Syst.*, 1997, pp. 1040–1046.
[195] S. Schaal, "Is imitation learning the route to humanoid robots?" *Trends Cogn. Sci.*, vol. 3, no. 6, pp. 233–242, Jun. 1999.
[196] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surv.*, vol. 50, no. 2, pp. 1–35, 2017.
[197] X. Liu, Z. Cao, Y. Yu, G. Ren, J. Yu, and M. Tan, "Robot navigation based on situational awareness," *IEEE Trans. Cogn. Develop. Syst.*, vol. 14, no. 3, pp. 869–881, Sep. 2022.
[198] L. Alchieri, D. Badalotti, P. Bonardi, and S. Bianco, "An introduction to quantum machine learning: From quantum logic to quantum deep learning," *Quantum Mach. Intell.*, vol. 3, no. 2, pp. 1–30, Dec. 2021.
[199] H. Hohenfeld, D. Heimann, F. Wiebe, and F. Kirchner, "Quantum deep reinforcement learning for robot navigation tasks," *IEEE Access*, vol. 12, pp. 87217–87236, 2024.
[200] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," *Commun. ACM*, vol. 65, no. 1, pp. 99–106, Jan. 2022.
[201] M. Adamkiewicz, T. Chen, A. Caccavale, R. Gardner, P. Culbertson, J. Bohg, and M. Schwager, "Vision-only robot navigation in a neural radiance world," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4606–4613, Apr. 2022.
[202] A. Amaravati, S. B. Nasir, J. Ting, I. Yoon, and A. Raychowdhury, "A 55-nm, 1.0–0.4V, 1.25-pJ/MAC time-domain mixed-signal neuromorphic accelerator with stochastic synapses for reinforcement learning in autonomous mobile robots," *IEEE J. Solid-State Circuits*, vol. 54, no. 1, pp. 75–87, Jan. 2019.
[203] A. Savinykh, M. Kurenkov, E. Kruzhkov, E. Yudin, A. Potapov, P. Karpyshev, and D. Tsetserukou, "DarkSLAM: GAN-assisted visual SLAM for reliable operation in low-light conditions," in *Proc. IEEE 95th Veh. Technol. Conf. (VTC-Spring)*, Jun. 2022, pp. 1–6.

HARINATH SRIDHARAN received the bachelor's degree in mechanical engineering from SASTRA Deemed University, Thanjavur, India, in 2022. He was an Intern with the Indian Institute of Technology Madras. He is currently a Data Analyst with Deloitte Consulting India Private Ltd., Bengaluru, India. He has published in the area of deep networks for soccer playing robots. His research interests include robotics, production planning, machine learning, and operations research.

NAMBALA RAMSAI received the bachelor's degree in electrical engineering from the National Institute of Technology Warangal, India, in 2016, and the master's degree in electrical engineering from the National Institute of Technology Rourkela, India, in 2020. He is currently pursuing the Ph.D. degree in electrical engineering with the Indian Institute of Technology Madras. He has published in the area of sensor fusion and deep networks. His research interests include robotics, FPGA-based system design, and neural networks.

BHASKAR VUNDURTHY received the Ph.D. degree from the Indian Institute of Technology Madras, in 2019. He is currently a Project Scientist with the Robotics Institute, Carnegie Mellon University, USA. Prior to this, he was associated with the Department of Electrical Engineering, University of Notre Dame, and MathWorks, India. He has published several articles in different areas in robotics. His research interests include multirobot systems and machine learning.