IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 21 July 2022, accepted 18 August 2022, date of publication 26 August 2022, date of current version 2 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3202208

RESEARCH ARTICLE

Autonomous Drone Swarm Navigation and Multitarget Tracking With Island Policy-Based Optimization Framework

SULEMAN QAMAR[1,2], SADDAM HUSSAIN KHAN[1,2,3], MUHAMMAD ARIF ARSHAD[1,4], MARYAM QAMAR[5,6], JEONGHWAN GWAK[7,8,9,10], AND ASIFULLAH KHAN[1,2,11]

[1]Pattern Recognition Laboratory (PRLab), Department of Computer and Information Sciences, Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad 45650, Pakistan
[2]PIEAS Artificial Intelligence Center (PAIC), Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad 45650, Pakistan
[3]Department of Computer Systems Engineering, University of Engineering and Applied Sciences (UEAS), Swat, Khyber Pakhtunkhwa 19060, Pakistan
[4]Centres of Excellence in Science & Applied Technologies, Islamabad Capital Territory, Islamabad 44090, Pakistan
[5]Department of Computer Science, University of Azad Jammu and Kashmir Muzaffarabad, Azad Kashmir 13100, Pakistan
[6]Department of Computer Science and Engineering, Kyung Hee University, Seoul 02447, South Korea
[7]Department of Software, Korea National University of Transportation, Chungju 27469, South Korea
[8]Department of Biomedical Engineering, Korea National University of Transportation, Chungju 27469, South Korea
[9]Department of AI Robotics Engineering, Korea National University of Transportation, Chungju 27469, South Korea
[10]Department of IT and Energy Convergence (BK21 FOUR), Korea National University of Transportation, Chungju 27469, South Korea
[11]Deep Learning Laboratory, Center for Mathematical Sciences (CMS), Pakistan Institute of Engineering and Applied Sciences, Nilore, Islamabad 45650, Pakistan

Corresponding authors: Asifullah Khan (asif@pieas.edu.pk) and Jeonghwan Gwak (jgwak@ut.ac.kr)

This work was supported by PAIC, PIEAS, IT Endowment under Pakistan Higher Education Commission (HEC) and "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-001 (1345341783)) and the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (Grant No. NRF-2020R1I1A3074141).

ABSTRACT Swarm intelligence has been applied to replicate numerous natural processes and relatively simple species to achieve excellent performance in a variety of disciplines. An autonomous approach employing deep reinforcement learning is presented in this study for swarm navigation. In this approach, complex 3D environments with static and dynamic obstacles and resistive forces such as linear drag, angular drag, and gravity are modeled to track multiple dynamic targets. In this regard, a novel island policy optimization model is introduced to tackle multiple dynamic targets simultaneously and thus make the swarm more dynamic. Moreover, new reward functions for robust swarm formation and target tracking are devised to learn complex swarm behaviors. Since the number of agents is not fixed and has only the partial observance of the environment, swarm formation and navigation become challenging. In this regard, the proposed strategy consists of four main components to tackle the aforementioned challenges: 1) Island policy-based optimization framework with multiple targets tracking 2) Novel reward functions for multiple dynamic target tracking 3) Improved policy and critic-based framework for the dynamic swarm management 4) Memory. The dynamic swarm management phase translates basic sensory input to high-level commands and thus enhances swarm navigation and decentralized setup while maintaining the swarm's size fluctuations. While in the island model, the swarm can split into individual sub-swarms according to the number of targets, thus allowing it to track multiple targets that are far apart. Also, when multiple targets come close to each other, these sub-swarms have the ability to rejoin and thus form a single swarm surrounding all the targets. Customized state-of-the-art policy-based deep reinforcement learning neuro-architectures are employed to achieve policy optimization. The results show that the proposed strategy enhances swarm navigation by achieving a high cumulative reward and a low policy loss. The simulations show that the proposed framework can efficiently track multiple static and dynamic targets in complex environments.

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

INDEX TERMS Navigation, swarm robotics, deep reinforcement learning, obstacle avoidance, target tracking, multi-agent.

I. INTRODUCTION

Deep reinforcement learning exploits the ideas of deep learning [1], [2] and reinforcement learning [3]. It has been used for learning advantageous behaviors for agent training [4]. The technique of integrating simple activities in such a manner that helps in developing a more sophisticated behavior is known as swarm intelligence [5], [6]. It has allowed us to replicate numerous natural processes by relatively simple species cooperating and completing complex tasks to achieve excellent results in a variety of disciplines while ostensibly conducting simple activities. As a result of its utility, engineering artificial multi-agent systems with swarm intelligence has become a burgeoning research field. Swarm intelligence has a wide range of uses [7], [8], including high-level monitoring of dynamic networks, adaptive routing in telecommunications, distributed sensing technology [9], surveillance [10], data processing, cluster analysis, search and rescue missions [11], [12],advertisement, and drone used as a delivery bot [13]. This is especially important in current times of pandemic, and there are also potential applications such as using nanobots within the body of a cancer patient to kill tumors [14]. Swarm intelligence is also being employed by NASA in planetary imaging [15]. Swarm intelligence is a natural phenomenon in which the activities of numerous dispersed and simple self-organized organisms combine to produce ''intelligent'' global behavior. Swarm awareness can be observed in nature in several subtle and awe-inspiring ways, such as when simple species acting independently interact to generate complex global behavior. Bee colonies, schools of fish, flocks of birds, ant colonies, hawks hunting, animal herding, and bacterial growth are all examples of swarm intelligence [16]. To fulfill common objectives like foraging, group coercion, and alignment control, these swarm systems employ the notions of ''quantity'' and ''coordination'' [17], [18]. Swarm robotics [19], [20] is a technique for coordinating many robots as part of a larger structure made up mostly of basic physical robots [21]. It is expected that robot-robot interactions and their interactions with the environment result in intended reciprocal behavior [22]. Artificial swarm intelligence, as well as biological observations of flies, ants, and other natural systems that display swarm behavior, inspired this work. However, most artificial swarm systems find it challenging to represent such a mix of behaviors displayed by natural swarms since doing so adds to the problem's complexity [23], [24]. When the drones are controlled manually, an operator must be present to operate them. Furthermore, traditional machine learning [25], [26] requires manual feature engineering, which is tedious and less flexible. This research work addresses the problem of the development of an end-to-end model for detecting targets in various settings and autonomous navigation [27] of drones tracking the targets while avoiding obstacles and maintaining stable agent formations.

This research aims to create artificial swarms for the purpose of navigation in unseen environments and tracking targets. Five key swarm behaviors are modeled: (1) swarm formation and organization, (2) dynamic obstacle avoidance, (3) locating single and multiple targets, (4) Navigation towards the target using the shortest path while sustaining swarm formation, and (5) tracking multiple targets by dividing the swarm into sub-swarms and tracking each target with a single sub-swarm. The research is particularly focused on finding multiple targets in complex environments resembling real-world scenarios by training swarm agents using Unity 3D. The proposed framework is shown in Fig.1. To sum up, this work has the following contributions:

1) A policy-based deep reinforcement learning framework named Island policy-based optimization framework (IPOF) is proposed, enabling the drone swarm to navigate autonomously while avoiding obstacles and tracking multiple targets. To prepare the drone swarm for real-life situations [28], complex 3D environments with dynamic obstacles having distinct morphology are created. In addition, resistive forces like linear drag, angular drag, and gravity are added to make the environments more realistic and complicated.
2) Novel reward functions have been introduced that allow the swarm to avoid barriers and track multiple targets while traversing the shortest path and maintaining a stable swarm structure. Both static and mobile targets can be efficiently tracked.
3) Improved policy and critic-based framework for the dynamic swarm management is introduced, thus increasing swarm efficiency.
4) To improve swarm navigation and decentralized setup while preserving the swarm's size variations, a mechanism that converts basic sensory input to high-level commands is employed. The concept of memory is also added to aid drone swarms in remembering the best paths.

II. RELATED WORK

Researchers have long been interested in extracting and implementing the principles that regulate these amazing biological swarm systems since their performance regularly surpasses individual biological organisms'. Swarm simulations and manual inspections have previously achieved substantial results [29], [30], [31], [32]. Mimicking the swarm behavior of animals manually has been studied extensively, for example, in [33] the laws that control enormous prey recovery in insects were employed to explain the realization of swarm behavior in robots. Minaeian *et al.* [34] developed SLAM algorithms for a set of unmanned ground and aerial vehicles to map and monitor crowds. Pheromone-based localization of dispersed targets by a swarm of virtual agents operating

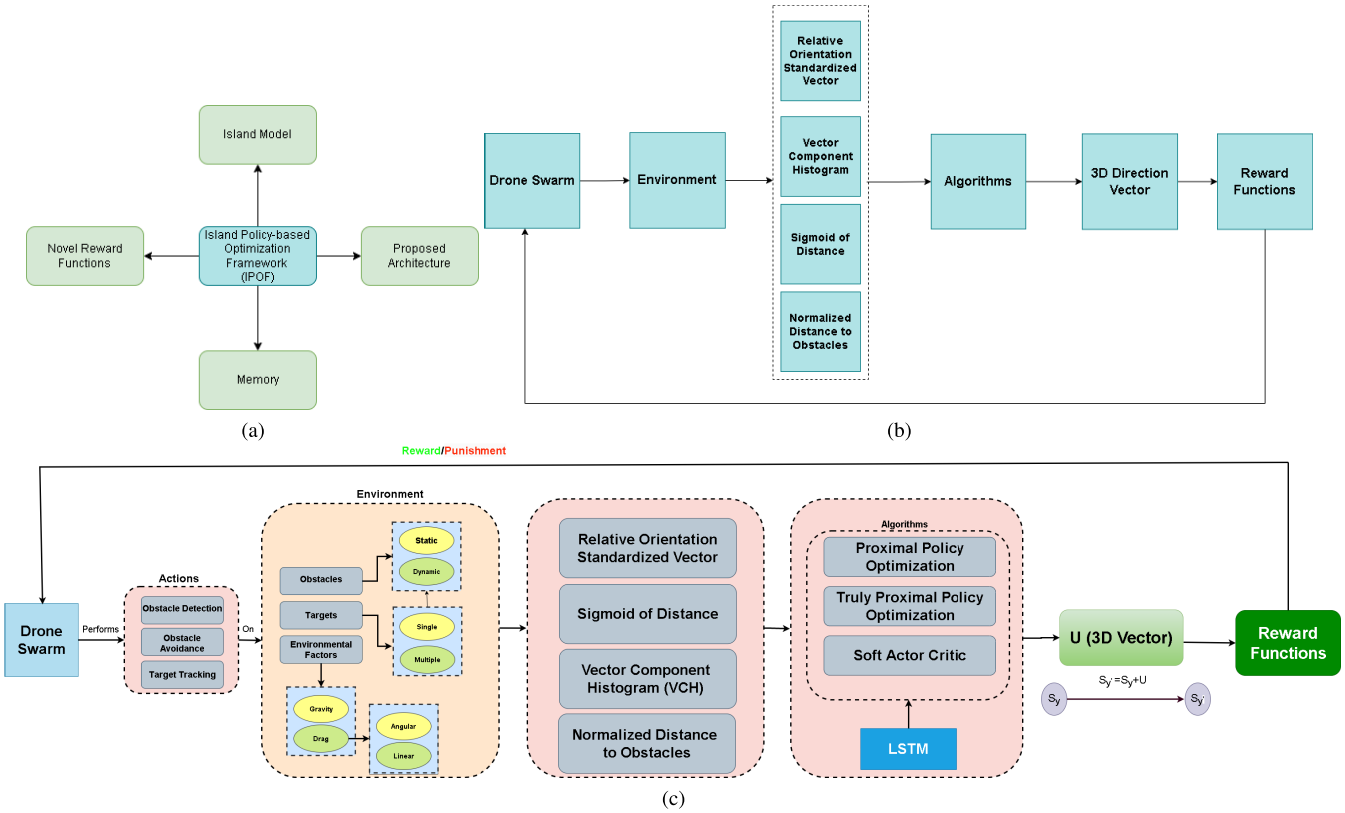


FIGURE 1. **Drone swarm navigation and tracking, (a) Island policy-based optimization framework (b) Drone swarm navigation and tracking block diagram (c) Flow Diagram representing major components, their relationship and arrangement.**

in a simulated discretized environment was studied in [35]. In their research, mini-UAVs are viewed as swarm agents, and they may have imperfections while detecting targets. Swarm behaviors, such as aggregation, foraging, creation, and monitoring, were studied in [36] and algorithms were developed to replicate such behavior.

When the problem's complexity grows exponentially, the time and effort required to solve it also grows dramatically. So, the time and effort spent inspecting, formulating, and solving a problem must be reduced. Q-Learning was developed by Watkins *et al.* [37] in 1992 as a strategy for iteratively training agents to behave optimally to maximize reward. With sufficient samples, Q-Learning converges to produce optimum action-value pairings in Markovian domains with a probability of one. Google Deep Mind built the first deep learning model based on Q-Learning (DQN) in 2013 [38], it could play a variety of Atari 2600 games using only pixels as feedback. However, in the Q-Learning algorithm, state space is continuous, but action space is discrete, so it can't be used in problem domains where action space also needs to be continuous.

To handle the problem of continuous action space, Lillicrap *et al.* [39] proposed a deep deterministic policy gradient-based actor-critic algorithm (DDPG) that borrows heavily from DQN in terms of simple architecture, including mini-batch updates and the Ornstein-Uhlenbeck process [40] as exploratory noise. Swarm formation and mutual localization was explored in [41], [42], with a few modifications, they utilized the DDPG [43] algorithm. Actor-network and critic-network input was adapted to make the swarm machine work. A novel technique was utilized in which they gave the critic network entire state data while only giving the actor network partial state data. As a result, the critic network uses global state data to modify the parameters of the actor network, whereas the actor network simply uses local state data.

In the technique proposed by Akhloufi *et al.*, a deep learning method is provided to anticipate the behavior of agents tracking a travelling drone [44]. A single agent was trained by [45] to maneuver in a dynamic maze-like environment using deep reinforcement learning. In their research, information from location sensing devices was employed as feedback to train the model with memory. Not only they used a single drone but also they worked on the navigation problem so there was no target involved at all.

An iterative technique called Trust Region Policy Optimization (TRPO) [46] was introduced that operated similarly to natural policy gradient methods and could optimize large non-linear policies. Thus, it could be effectively used for neural networks but required second-order derivative calculations. Proximal Policy Optimization (PPO) [47] from Google Deep Mind became a state-of-the-art solution to train

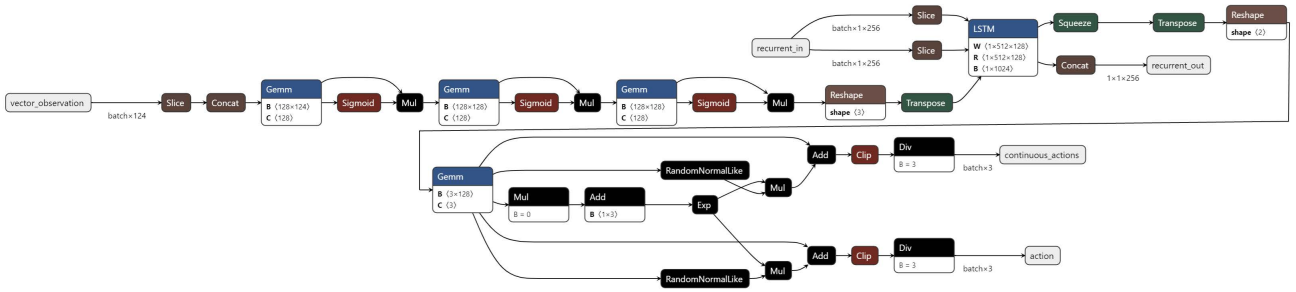


FIGURE 2. Proposed architecture for training describing all components of the network architecture for the purpose of training the model.

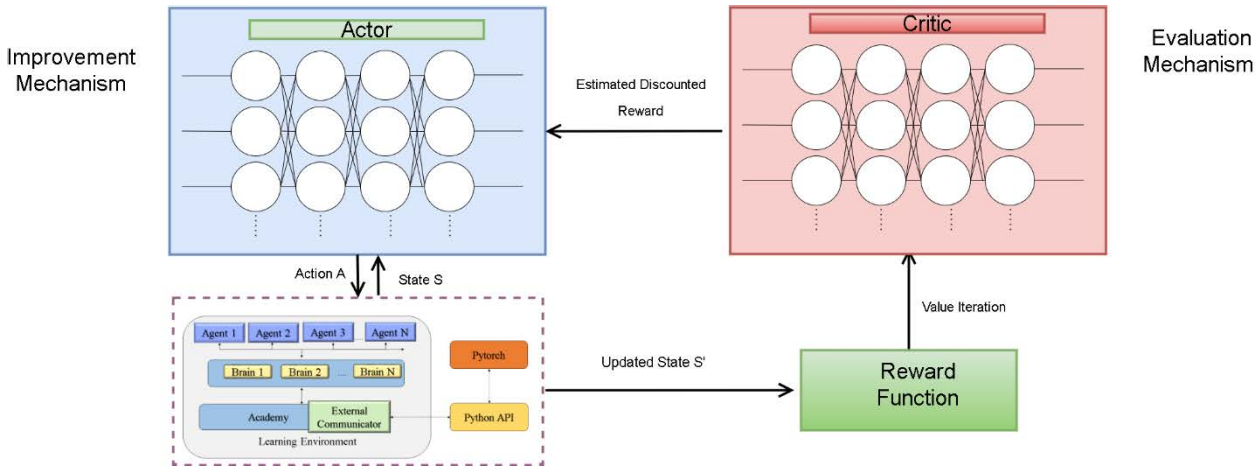
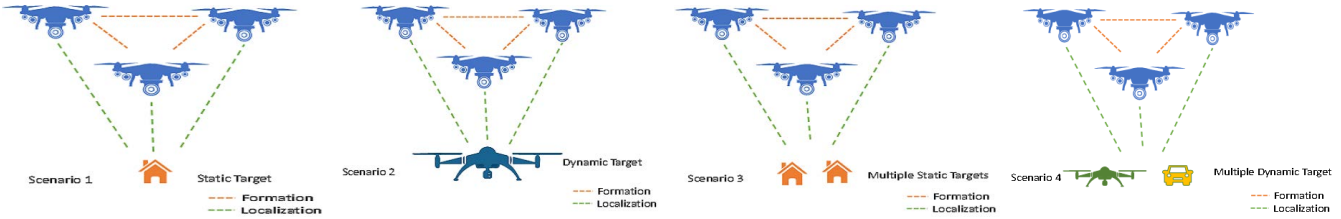


FIGURE 3. Actor-Critic along with simulation environment.

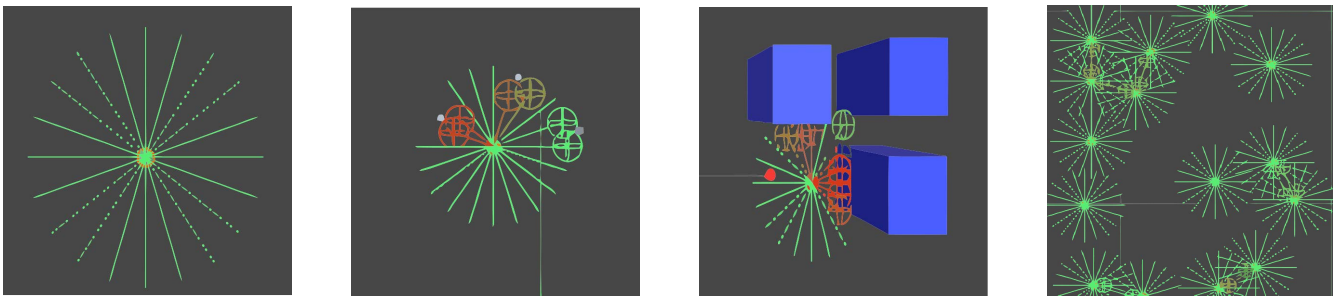
agents because of its sample reliability. It largely followed the concept behind TRPO but reduced calculations from second-order derivative to the first-order derivative. It used a stochastic gradient ascent to optimize a ''surrogate'' clipped objective function. In the Atari domain, the PPO algorithm out-performed Advantage Actor Critic (A2C) [48]and Actor Critic with Experience Replay (ACER) [49]. PPO algorithms were used to measure the efficiency of OpenAI Gym on high-dimensional controls, such as humanoid running and steering. In [50] code level optimizations for the PPO algorithm to work properly were summarized. A variant of the PPO algorithm named IEM-PPO [51] was presented with improved sample efficiency, better stability, and robustness, yielding comparatively higher cumulative reward, but took more time to train. PPO algorithm along with incremental curriculum learning [52] and long-short-term memory (LSTM) [53][55] was utilized to implement an adaptable navigation algorithm. The Truly Proximal Policy Optimization (TPPO) [54] modifies the PPO algorithm to perform slightly better in terms of stability and sample efficiency. Hämäläinen *et al.* [55] argued that in PPO, the variance of exploration prematurely shrinks, which makes progress slower, and proposes PPO-CMA to dynamically increase or decrease the variance of exploration. A new surrogate learning objective featuring an adaptive clipping mechanism named as PPO-λ is introduced in [56]. It iteratively improves policies based on a theoretical goal for adaptive policy change. PPO algorithm was employed in [57] to create drone swarms using multiple sensors per agent to reach target information while avoiding obstacles. Wang *et al.* [58] worked on an autonomous multi-agent [59] target reinforcement model using UAVs for searching and tracking. Camera, IMU, and GPS were used as sensors. Since their work was based on patrol, they only considered a constrained environment. Qu *et al.* [60] leveraged the concept of association for multiple target tracking. Their research centered on intelligent sensors that can distinguish readings based on targets using Grideye, an infrared sensor with the ability to calculate target location and surface temperature. Multi-target tracking on MOT15 and MOT16 datasets is performed by Ren *et al.* [61]. Although they were challenging datasets but the work was based on tracking humans.

PPO, despite having good performance suffered from sample efficiency. It showed good performance On small sample spaces, however low sample efficiency caused a lot of problems on large sample spaces. TPPO enhanced sample efficiency but improvements were still limited. To counter this problem, an off-policy method known as Soft Actor-Critic



(a) (b) (c) (d)

FIGURE 4. Drone Swarm surrounding and tracking, (a) Single static target (b) Single dynamic target (c) Multiple static targets (d) Multiple dynamic targets.



(a) (b) (c) (d)

FIGURE 5. (a) Agent with its sensing area (b) Three levels of danger identified by agent, "Green" represents slight danger; no immediate action required, "Brown" represents intermediate danger; try to minimize it without sacrificing goal; "Red" represents extreme danger; prioritize safety (c) Agent's interaction with obstacles (d) all agents maintaining safe distance with each other.

(SAC) was introduced in [62] that focused on maximizing reward with maximum possible entropy, which is the measure of randomness. SAC required extensive hyperparameter tuning in some cases but achieved state-of-the-art results.

III. METHODOLOGY

Due to the complexity of the current world and the comparatively poor sample efficiency of algorithms in the field of deep reinforcement learning, it is difficult to directly create a real-life model. Furthermore, explicitly training our model in the real world might result in mishaps. Thus, models are trained using simulations, utilizing Unity3D engine since it provides the required tools needed to construct complex 3D environments. Another reason is that it includes the ml-agents library, which allows Python to be used as a back-end for deep learning tasks. Architecture components for the training module are represented in Fig. 2 while general Actor-Critic along with rewards and relation with the learning environment and different scenarios that can occur during tracking targets is shown in Figs. 3 and 4.

A. SIMULATION ENVIRONMENT

It is crucial to design environments that support agents' ability to train successfully and efficiently. Therefore, a variety of scenarios are constructed for training the drone swarm to determine the best conditions. Agents with their sensing area and different levels of danger identification by agents are visualized in Fig. 5. Basic environments are visualized in Fig. 6(a) whereas agents within environment is given in Fig. 6(b). Several settings are used during training to extend our model and assess the efficacy of different training circumstances. All models include a 3D environment with a variable number of obstacles that cover various cubical volumes, as indicated in Table 1. Obstacle locations are chosen at random to make our method adaptable to changing surroundings. The number of multi-agent training targets varies across simulations, ranging from two to sixteen. Table 2 lists the environments utilized for multi-target training. A summary of actions that can be taken by agents is listed in Table 3.

TABLE 1. Single target environments.

Volume in units^3	Length of each axis in units	Number of obstacles	Length of obstacles in units	Obstacle positions
1,000,000	100	100	Ranging between 1 to 10	Random
500,000	50	60	Ranging between 1 to 5	Random
100,000	10	30	Ranging between 1 to 7	Random

The drawback of using only static targets throughout the training period caused the model to memorize their locations resulting in the loss of generalization. Counteracting this, the position of targets in the environment is also randomized by employing the random tick, which involves changing the positions of target points every 100 ticks to introduce time-decoupled uncertainty. To generate ticks, a random floating-point value is generated at each time-step. The threshold for ticks being considered is 0.85.

TABLE 2. Multi-target environments.

Volume in units^3	Axis Length	No. of obstacles	Obstacles' Sizes	Obstacle positions	No of Targets	Target Type
1,000,000	100	20	Ranging between 1 to 10	Static	2	Static
500,000	50	50	Ranging between 1 to 5	Random	2	Dynamic
100,000	10	30	Ranging between 1 to 7	Random	4	Static
100,000	10	10	Ranging between 2 to 8	Static	8	Dynamic
700,000	70	100	Ranging between 5 to 10	Random	16	Static
1000,000	100	200	Ranging between 1 to 10	Random	16	Dynamic

TABLE 3. Allowed actions of agents.

Action	Possibilities
Motion	9 Directions (up, down, right, left, forward, backward, pitch, roll, yaw)
Rotation	[0-360)
Hover over target	Yes/No

Agents must also begin each simulation in a random place, which necessitated the creation of a position randomizer. It generates a random position in the environment from all available places. Obstacles, targets, and drone agents that have already been created are all eliminated from prospective locations. Each agent utilizes distance sensors in a circular pattern around them to detect and avoid obstacles. Proposed model is faster and uses fewer resources because these sensors eliminate the need for significant processing, which is common in camera-based techniques.

Every agent has knowledge about the targets and their immediate surroundings. At each time-step, the positions of the targets are assumed to be known. A sigmoid of the normalized geodesic distance between the targets' and the agents' positions is taken to generate a compact representation of the targets' locations. For ease of use, all agents and targets have an aligned coordinate system. Also, their rotational characteristics are locked. The following formulae are used to calculate the relative orientation standardized vector $\boldsymbol{P}$ as shown in (1). The distance between target locations $\boldsymbol{i}$ and agent y are then computed by taking their sigmoid as shown in (2).

$$\mathbf{P} = \left(\frac{\mathbf{S_i} - \mathbf{S_y}}{||\mathbf{S_i} - \mathbf{S_y}||} \right) \tag{1}$$

$$\boldsymbol{\sigma}^{y}_{b_i} = \left(\frac{||\mathbf{S_i} - \mathbf{S_y}||}{1 + ||\mathbf{S_i} - \mathbf{S_y}||} \right) \tag{2}$$

where S_i and S_y denote the target points and agent y's location, respectively. The location information related to all agents present in the nearby zone and obstacle sensor outputs are included in the information state vector. As agent numbers in a neighborhood might fluctuate and deep neural networks (DNNs) have finite input capacity, feeding location data from nearby agents directly into the network can be difficult. To address this problem, a 3D histogram approach is proposed. In the first step, vectors are computed by subtracting the position of agent y from all agents surrounding it, and then these vectors are mapped into J bins in each axis. Then the axes are concatenated into a single vector and given to DNN as input. Vector component **<k>** histogram computation for every agent is calculated using an algorithm called VCH presented as follows:

Algorithm 1 VCH

$His_j^{<K>} \leftarrow 0, j = 1, 2, 3, \ldots J$ (Initialize)
Input:
x: number of agents; D_c:Communication Radius
Output:
Vector Component Histograms
For x=1 to W do

$$j = \left[\frac{< K >_y^x + D_c}{D_c \times 2} \right], g \in \{x, y, z\}$$

$$\mathrm{D}_s \leq b_y^x \leq D_c \qquad His_j^{<K>} = \frac{1 - \sigma_y^x + D_c}{W \times 3}$$

Otherwise 0

where x denotes all agents except agent y, the difference in **k** components of position vectors of agents x and y is referred to as $< k >_y^x$. Distance between agent x and y is denoted by b_y^x; D_s is safe distance parameter while D_c represents a communication area. Obstacle detection sensors denoted by (D_{sen}) are considered time-of-flight (ToF) sensors. Output of all sensors is **Ti** represented by (3).

$$T_i = 1 - \begin{cases} \dfrac{c_i}{D_{sen}}, & \text{if if an obstacle is sensed by ``i''} \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

After that, the data from all the sensors are aggregated and submitted to the DNN.

B. ACTION GENERATION

A delta vector $\boldsymbol{u}$ composed of three variables, representing 3D coordinates, is generated. The vector **u** and the agent's present position vector S_i is then summed to get new position S_i' as shown in (4).

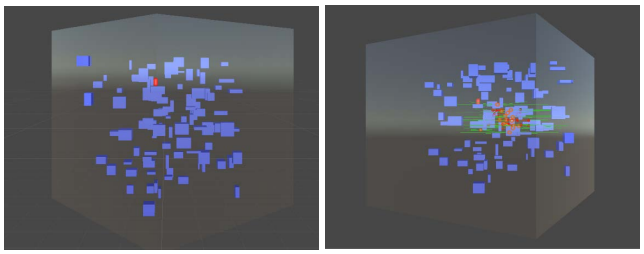
$$\mathbf{S'_y} = \mathbf{S_y} + \mathbf{u} \tag{4}$$

This action generation strategy shown in Fig. 7 broadens the applications of the proposed approach.

C. PERFORMANCE METRICS

The performance metrics for assessing successful training of the model are explained in this section.

Mean Cumulative Reward (MCR) is an assessment of total reward which indicates an increasing trend during effective training.



(a) (b)

FIGURE 6. Environment visualization: grey box represents outer boundary (a) obstacles are blue colored while target is represented by a red colored capsule (b) Agents with their obstacle lines (green in color) are visualized.

Value Loss (VL) is the performance indicator for assessing policy change, registers a declining trend during effective training.

Policy Loss (PL) is an accurate state value prediction which exhibits growing trend until the reward is stable, then it starts to drop during effective training.

Entropy (E) represents the unpredictability of the model's decisions and exhibits a diminishing trend during effective training.

D. MODELING REWARDS

Training data is acquired by running multiple simulations in parallel and then optimizing our novel reward function using PPO, TPPO, and SAC. R_n in (5) is the navigation reward that allows a swarm of drones to calculate the shortest path between the target and the swarm in real-time, even if the target is behind a large obstacle with no direct path to it. The shortest path is determined at each time-step, while the target is moving and drones are in flight so that drones may surround the target as quickly as possible. R_o in (6) denotes the reward for assisting in the construction and organization of drone agents. It enables them to form swarms in real-time while remaining at a safe distance yet close enough to communicate. With the assistance of R_s, the swarm can avoid obstacles (7). In addition, if an agent leaves the environment or if the agent is destroyed, then a negative reward as punishment is generated, and the destroyed agent is respawned at some random place inside the environment. Total reward by a single swarm is represented using RS in (8) which is a combination of individual rewards like R_n, R_o, R_s. R_{ms} (9) rewards all swarms that are present in the environment and are cooperating. Swarm divides into sub-swarms and similarly multiple swarms combine to form a single swarm. The number of targets determines how the swarm subdivision is done. If there is one target with a swarm tracking it, and another target is introduced, the swarm will split to track both targets. Similarly, increasing the number of targets increases the number of swarm sub-divisions. Also, if two swarms are tracking two targets and one of the targets is eliminated, the two swarms merge to form a single swarm.

$$R_n = \begin{cases} 1 - \sigma\left(b_y^i - D_s\right), & \text{if } b_y^i \in [D_s, \infty) \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

$$R_o = \sum_{x=1}^{X} \begin{cases} \frac{1-\sigma\left(b_y^i - D_s\right)}{x \times 3}, & \text{if } b_y^x \in [D_s, D_c) \\ -1, & \text{otherwise} \end{cases} \tag{6}$$

$$R_s = 1 - \sigma\left(\sum_{p=1}^{P} U_p\right) \tag{7}$$

$$RS = R_n + R_o + R_s \tag{8}$$

$$R_{ms} = \frac{\sum_{n=1}^{N} RS_n}{N} \tag{9}$$

E. HYPERPARAMETERS

The policy and critic networks, each with two dense layers of 64 neurons, are employed by the PPO, TPPO and SAC with simulation hyperparameters given in Table 4 while training

TABLE 4. Simulation hyperparameters (PPO, TPPO and SAC).

S.No.	Hyperparameter	Value
01	Simulation Instances	28
02	Agent Quantity in Simulations (S)	23
03	Number of steps per episode	900 steps
04	Radius for Communicating (D_c)	9 units
05	Histogram Bins per Axis (K)	32
06	Obstacle Sensors per Agent (J)	18
07	Sensor Range $(D_s en)$	7 units
08	Safe Region (D_s)	3 units

TABLE 5. Training hyperparameters (PPO and TPPO).

S.No.	Hyperparameter	Value
01	Total steps	55 millions
02	Time Horizon	512 steps
03	Size of Batch	1024
04	Buffer Size	10,240 steps
05	Rate of Learning	0.0007
06	Policy update penalty (beta)	0.007
07	Clipping Value	0.3
08	Lambda	0.96
09	Epochs	2
10	Learning Rate Decay	Linear

TABLE 6. Training hyperparameters (SAC).

S.No.	Hyperparameter	Value
01	Total steps	55 millions
02	Time Horizon	512 steps
03	Size of Batch	256
04	Buffer Size	10,240 steps
05	Rate of Learning	0.0007
06	Policy update penalty (beta)	0.007
07	Clipping Value	0.3
08	Lambda	0.96
09	Epochs	2
10	Buffer Initial Steps	12
11	Initial Entropy Coefficient	0.9
12	Save Replay Buffer	True
13	Tau	0.005
14	Steps per Update	3
15	Reward Signal Number Update	3

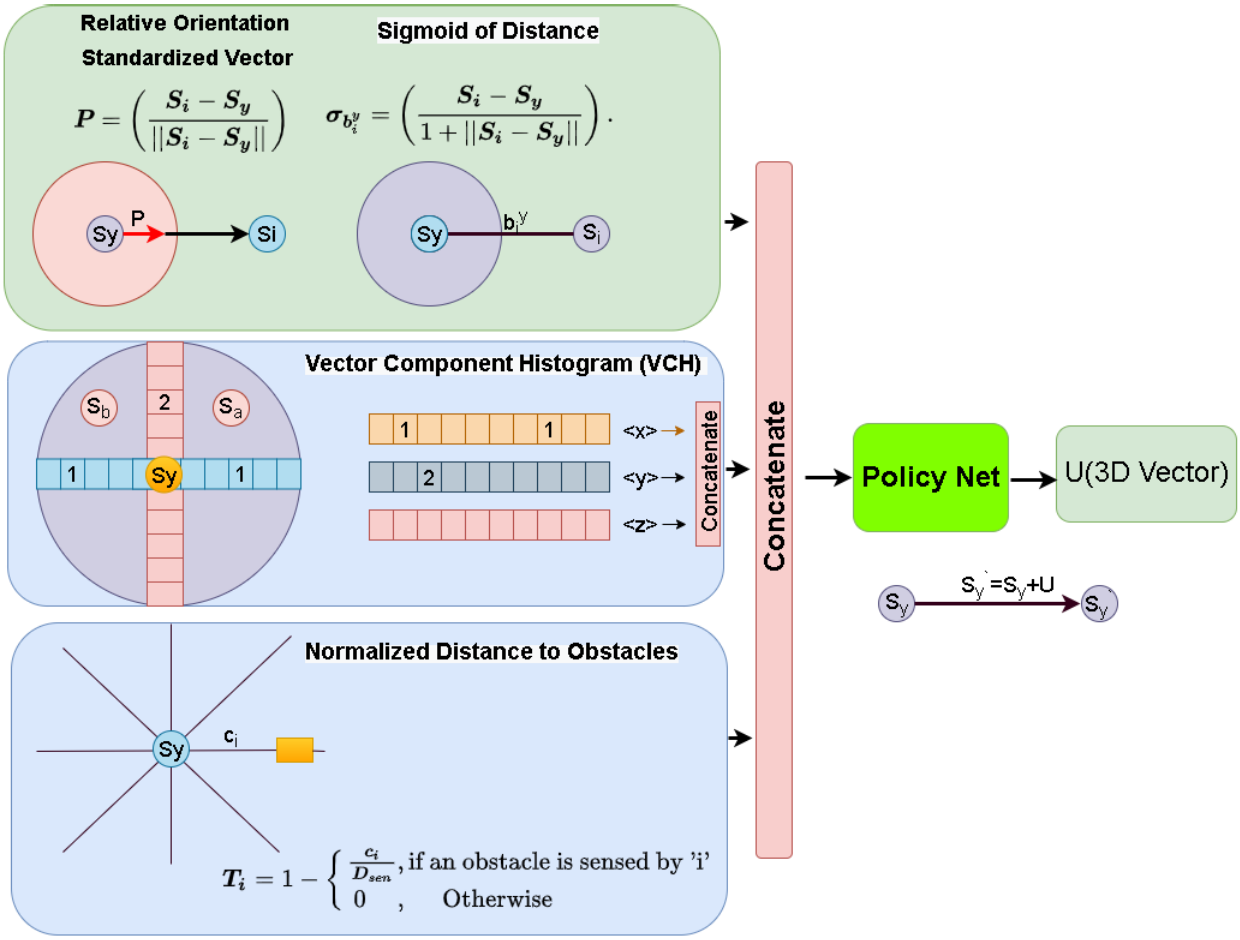


FIGURE 7. Creation of U (3D Vector) representing next location of drone in the 3D environment by concatenating Relative Orientation Standardized Vector, Sigmoid of Geodesic Distance, Vector Component Histogram (VCH), and Normalized Distance to Obstacles.

hyperparameters given in Table 5 and Table 6. Architectures are optimized to facilitate faster convergence at about 10 million steps. Moreover, only two epochs were used to further cut down on training time.

IV. RESULTS AND DISCUSSION

Performance curves for PPO, TPPO and SAC are obtained for 55 million training steps.

Comparative Cumulative Reward (CR) is provided in Fig. 8(a). The increasing trend shows successful training. SAC performs best due to it being sample efficient, while PPO and TPPO have comparable results. Policy Loss (PL) and Value Loss (VL) curves are given in Fig. 8(b) and Fig. 8(c), the performance of SAC can be visually verified. In Fig. 8(d), Entropy, change in policy during training, is shown. Entropy shows decreasing trend which correlates to successful learning.

A. RESULTS

Spheres indicate the agents, while target is represented by black and sometimes red color. Everything else is included in obstacles. When two or more agents are in each other's communication zone, a green-colored edge forms between them to visually demonstrate communication. The drone agents not only effectively form formations in the demos, but they are also able to navigate the complicated terrain, avoiding different sized and shaped obstacles while maintaining a swarm-like structure.

Furthermore, it successfully reaches and surrounds the targets. Even if the target is constantly moving, the swarm continues to encircle it. When several dynamic targets are traveling in various directions, the swarm splits into sub-swarms, with each sub-swarm having the same number of agents following a single target object. Sub-swarms merge to form a bigger swarm when targets approach closer to one another.

A visual assessment of the swarm's activity is required to demonstrate the feasibility of our strategy. The visual depictions of several aspects of a swarm's activity are given in experiment section (Demonstrations can be accessed here[1]).

[1] www.youtube.com/playlist?list=PLq0872kWvR0VqTcnWDrudsDRELvtFY7w

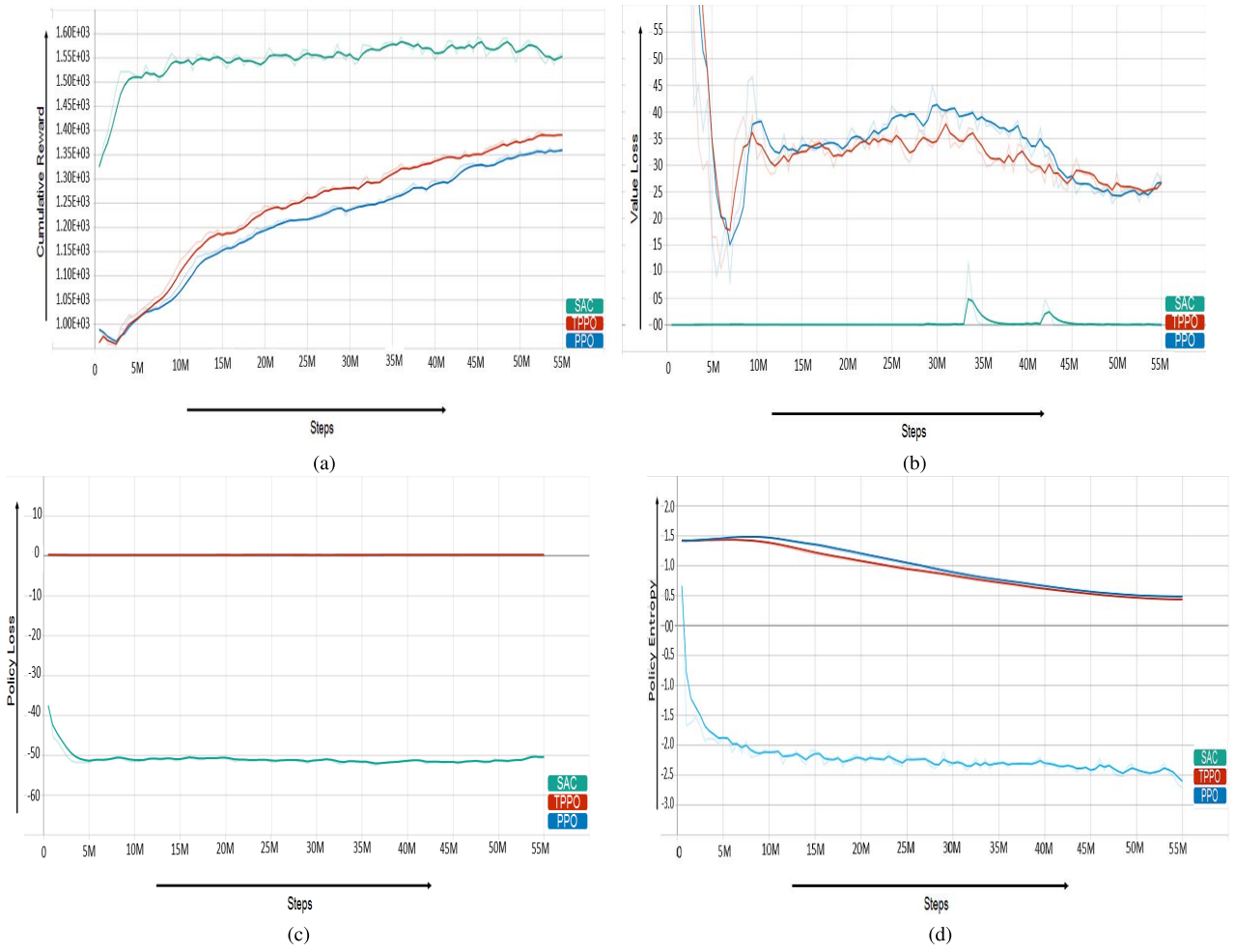


FIGURE 8. Comparison of SAC, PPO and TPPO in terms of (a) Mean cumulative reward (MCR) (b) The value loss (y-axis) against 55 million training steps (x-axis) (c) Policy loss.

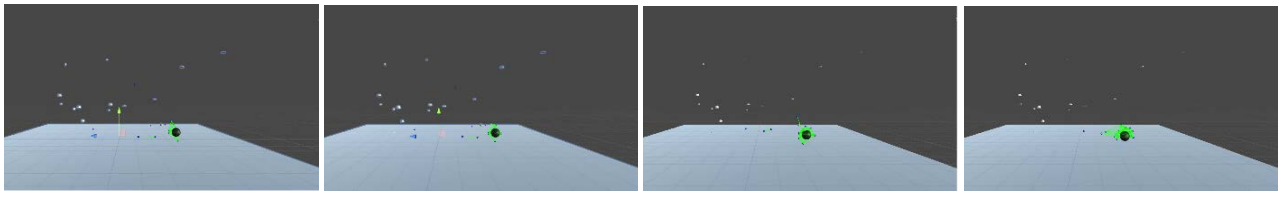


FIGURE 9. Swarm formation, organisation, maintenance while surrounding the target and tracking the moving target in obstacle-free environment.

V. EXPERIMENTS AND DISCUSSION

Experiments are carried out to test and develop our models. The complexity of the environment is gradually raised, and model improvements are made through gaining insights from the model's undesirable actions. Experiment summary is given in Table 7. Single target localization and tracking can be done by single drone although not efficiently. Furthermore, swarm is needed in case there are multiple targets. As shown in demonstrations, complex swarm behaviors are learnt automatically.

A. EXPERIMENT 1: SWARM ORGANIZATION AND OBSTACLE AVOIDANCE

For evaluation of swarm formation and observe swarm-like behavior, a $1000 units^3$ obstacle-free environment is created as shown in Fig.9(a). Drones successfully formed a swarm and surrounded the black target visualized in Fig 9(b). It is important to observe that the target was a static object in these cases.

In the next experiment, target was made mobile with a constant speed while keeping all other factors the same as

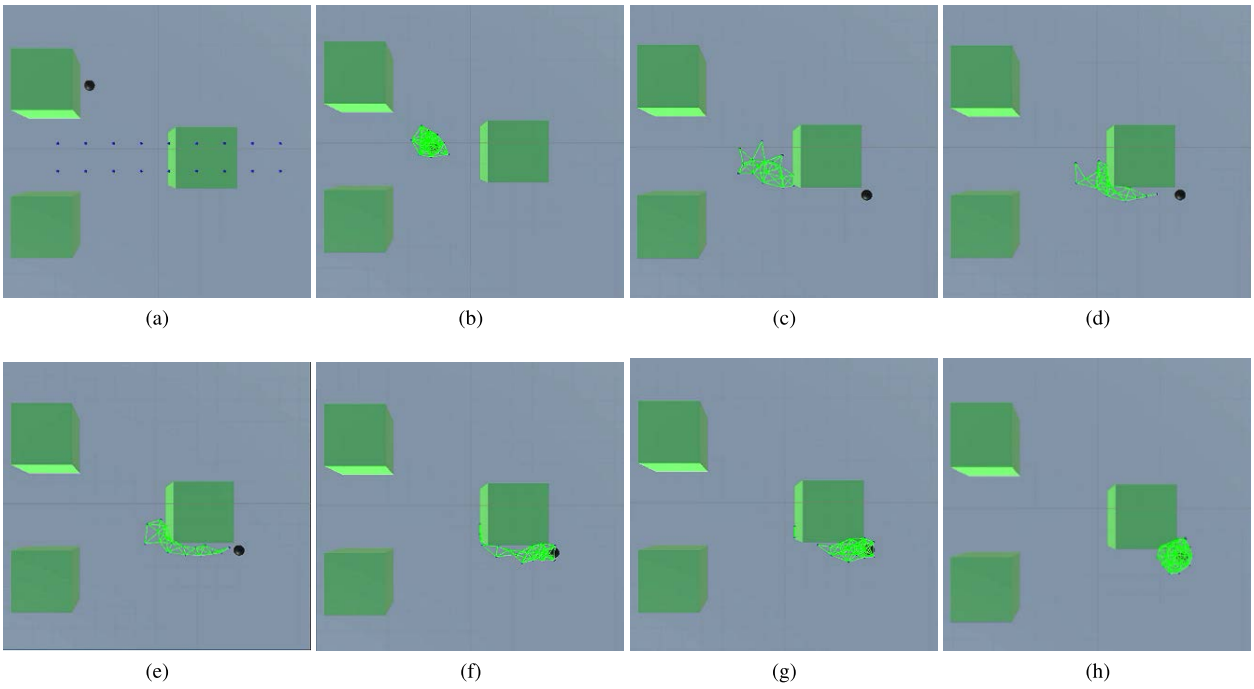


FIGURE 10. Swarm organization and obstacle avoidance.

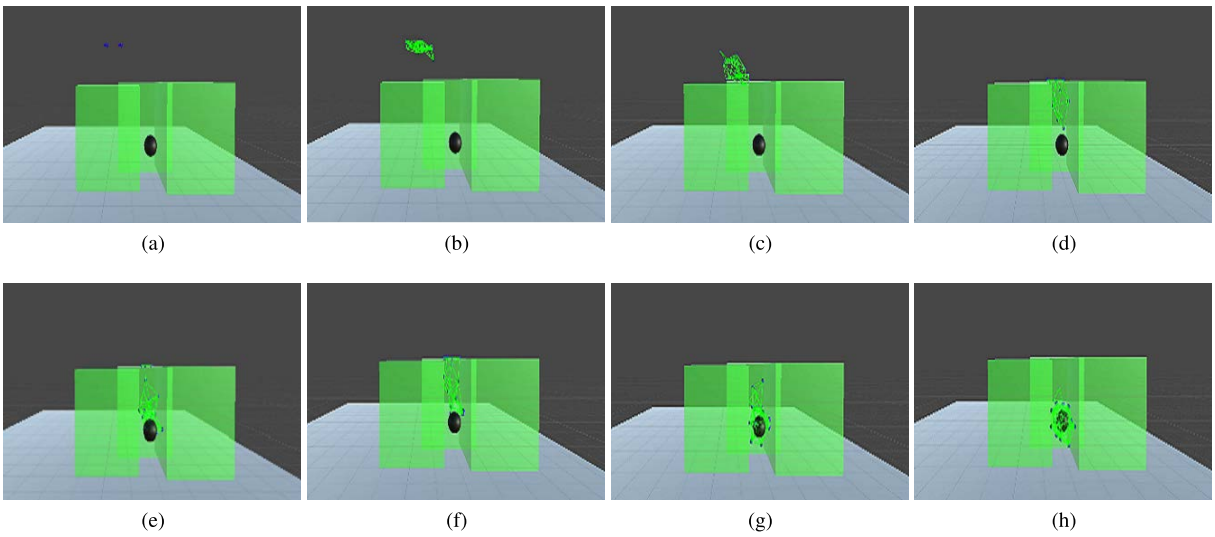


FIGURE 11. Target surrounded by obstacles by three sides, swarm has only one path traversable that can lead to target.

before, swarm again surrounded the moving target as shown in Fig.9(d). Swarm kept hovering over the target by virtue of each drone trying to minimize distance between itself and the target while avoiding collisions with other drones and also the target.

In the next experiment, three large green obstacles, a black target and 18 blue colored drones are placed in the environment in such a way that drones can travel through a small opening at the top or moving all the way to the other side and then entering from there but travelling this way will have the effect of covering more distance for the swarm. Drones form a swarm first and then use the shortest available path, that is, from the top hole successfully surround the target visualized in Fig.10(a) and 10(b), respectively. Also, the reason behind relocation of target behind an obstacle was to further evaluate the obstacle avoidance mechanism of swarm as seen in Fig.10(c).

Swarm's ability to avoid obstacles while maintaining swarm formation, surrounding the target and tracking dynamic target can been observed in Fig.10(d-h).

B. EXPERIMENT 2: TARGET SURROUNDED BY OBSTACLES

In this experiment, the target object was placed behind a series of obstacles (transparent green in color) and given

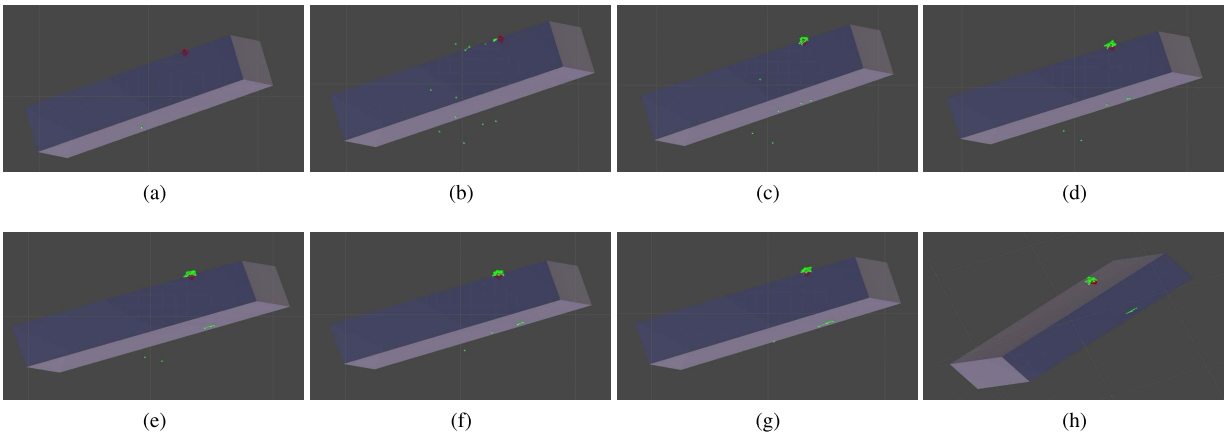


FIGURE 12. Target behind an obstacle with no visible path using Euclidean distance, swarm can't reach the target.

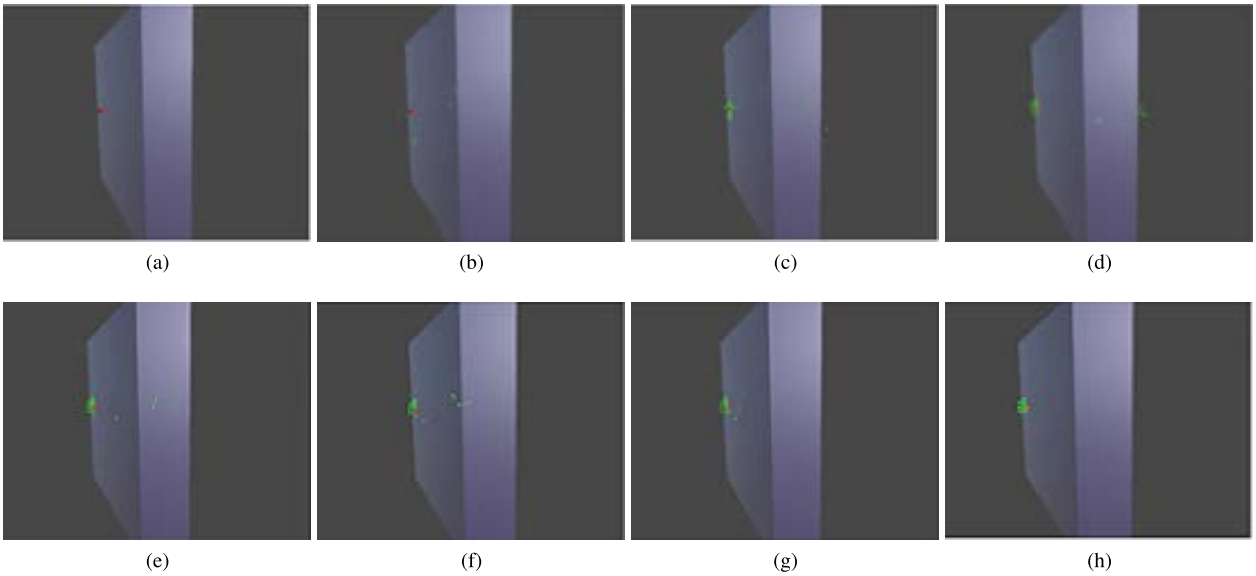


FIGURE 13. Target behind an obstacle with no visible path using Geodesic distance, swarm can reach the target by passing through one side of the obstacle.

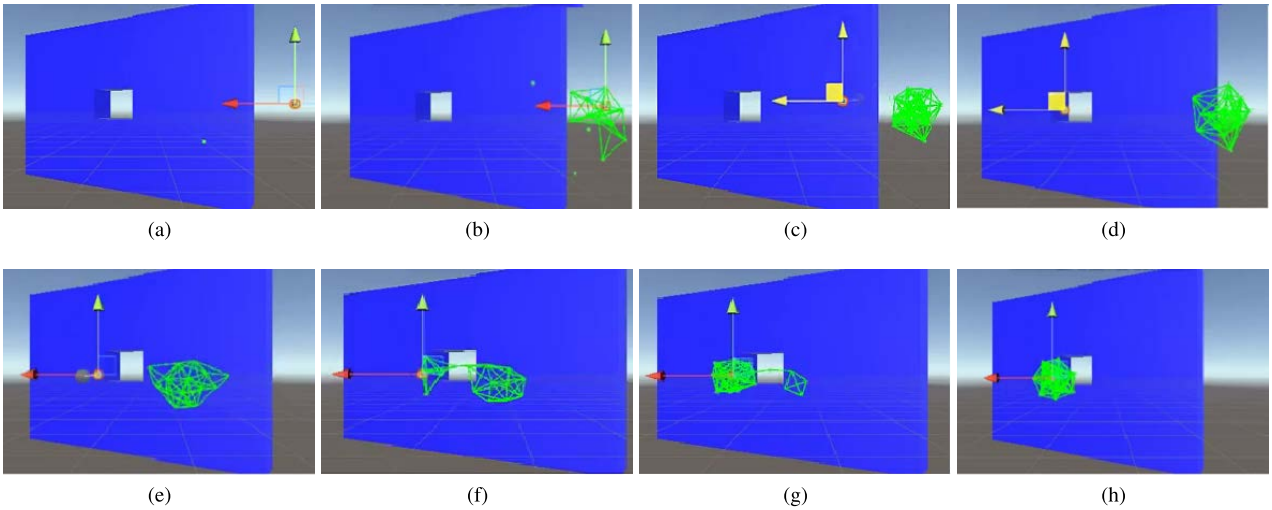


FIGURE 14. Swarm traversing through a hole in single file.

a single convoluted path to transverse in order to reach the target shown in Fig.11(a-b). The swarm's trajectory demonstrates intelligent activity. The swarm body moves around obstacles trying to find an opening and when found,

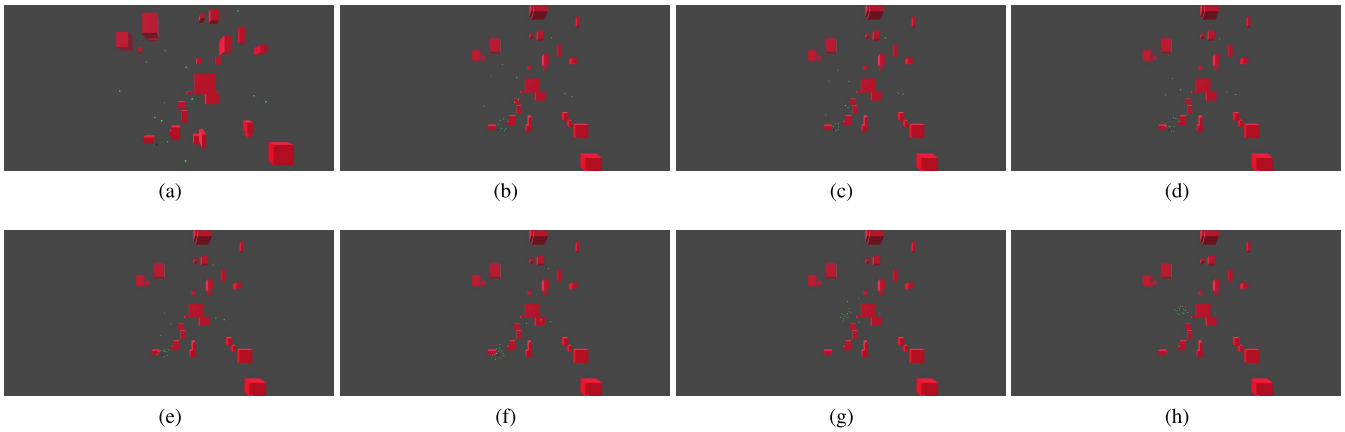


FIGURE 15. Multiple Priority-based targets' tracking in complex environment.

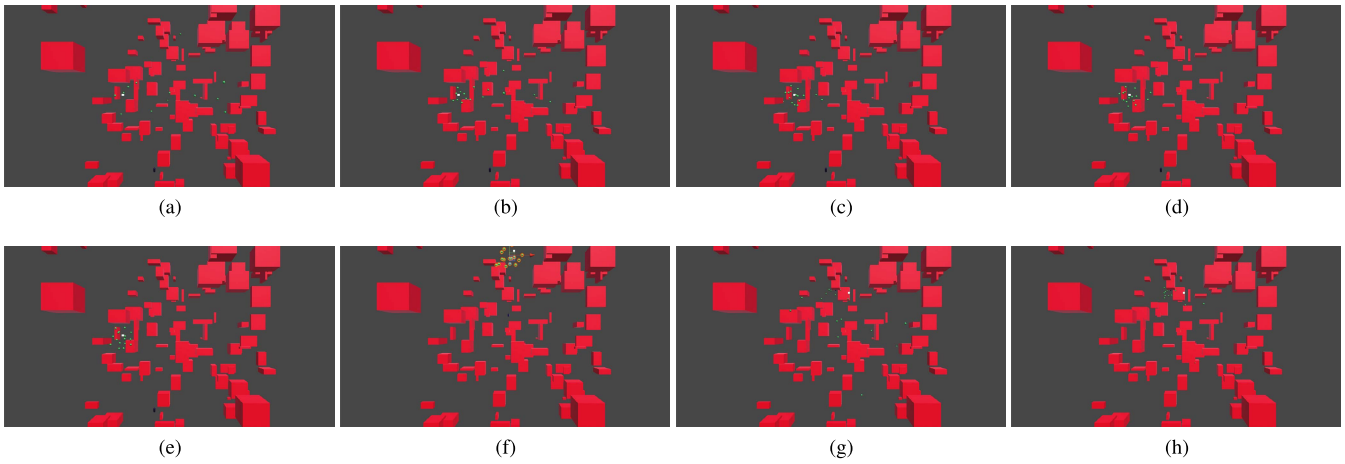


FIGURE 16. Multiple alternating Priority-based targets' tracking in complex environment.

swarm quickly surrounds the target visualized in Fig.11(c) and 11(d-h), respectively.

C. EXPERIMENT 3: TARGET BEHIND AN OBSTACLE

In this experiment, drones' robustness is tested if they can find a target hidden completely behind an obstacle. The first experiment failed to provide ideal results since the swarm was unable to reach the destination as shown in Fig.12. The issue was that the swarm was using Euclidean distance, which does not account for obstructions or alternative pathways. So, geodesic distance was introduced which calculates shortest distance along the manifold. Same experiment was re-performed with geodesic distance successfully (Fig.13). As seen in Fig.13(c-f), swarm moves around the wall to reach the target while traversing shortest possible distance.

D. EXPERIMENT 4: SWARM TRAVERSING THROUGH A HOLE IN SINGLE FILE

In this experiment, a hole was added in the wall which was present in the previous experiment. As the path through the whole has become the shortest path, the idea is to check if the model would make the swarm pass through the hole or if it would still let the drone swarm pass through the nearest corner of the object. To further increase the difficulty, hole was made small enough so that only one drone can pass through it at a time. Target was placed on one side of the wall (Fig.14(a) and drones were allowed to converge around it and form a swarm (Fig.14(b-c)). Then target was moved to other side of the hole (Fig.14(d)). Swarm started to move through the hole, one drone at a time, and surrounded the target on the other side (Fig.14(e-h)).

E. EXPERIMENT 5: MULTIPLE TARGETS

Two targets were given distinct priorities in this experiment. Targets were visually distinguished based on colour; a black coloured square had a greater priority (level one) than a blue coloured square (level two). The swarm swiftly approached the higher priority target, as seen in Fig.15, while the lower priority target was ignored since advancing towards it would deprive the swarm of the higher priority target reward. Moving away from the objective also adds a negative reward, which is additional motivation to stick to the higher priority target.

In the next iteration of this experiment, priority between targets was altered after a certain time as seen in Fig.16, the swarm always abandoned the lower priority target and went towards the higher priority target.

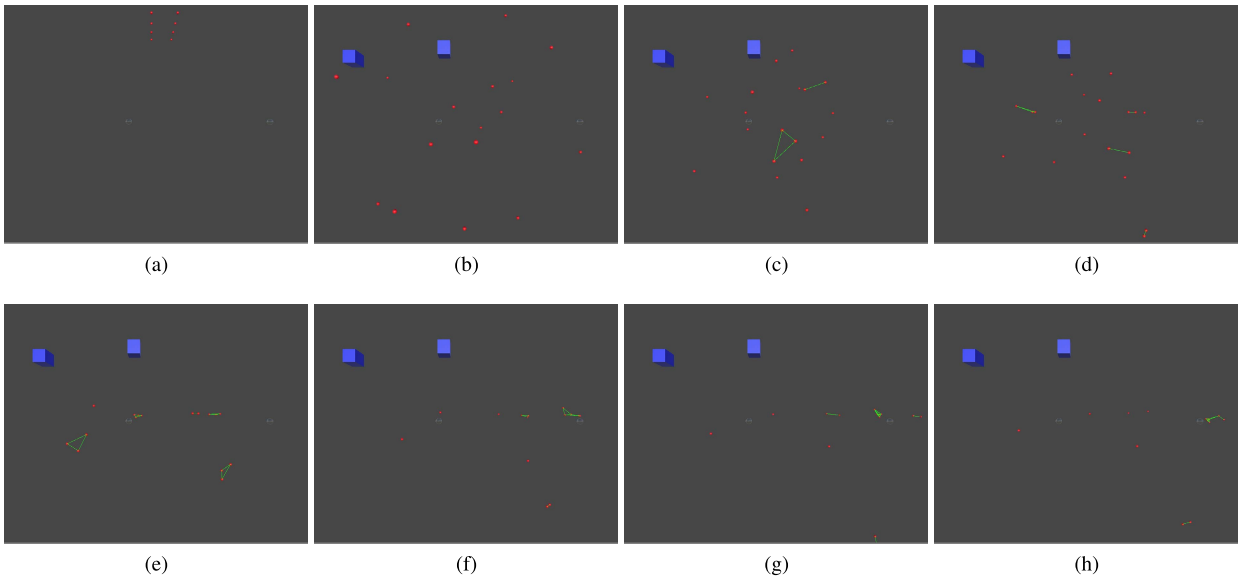


FIGURE 17. Multiple targets' tracking with same priority in simple environment.

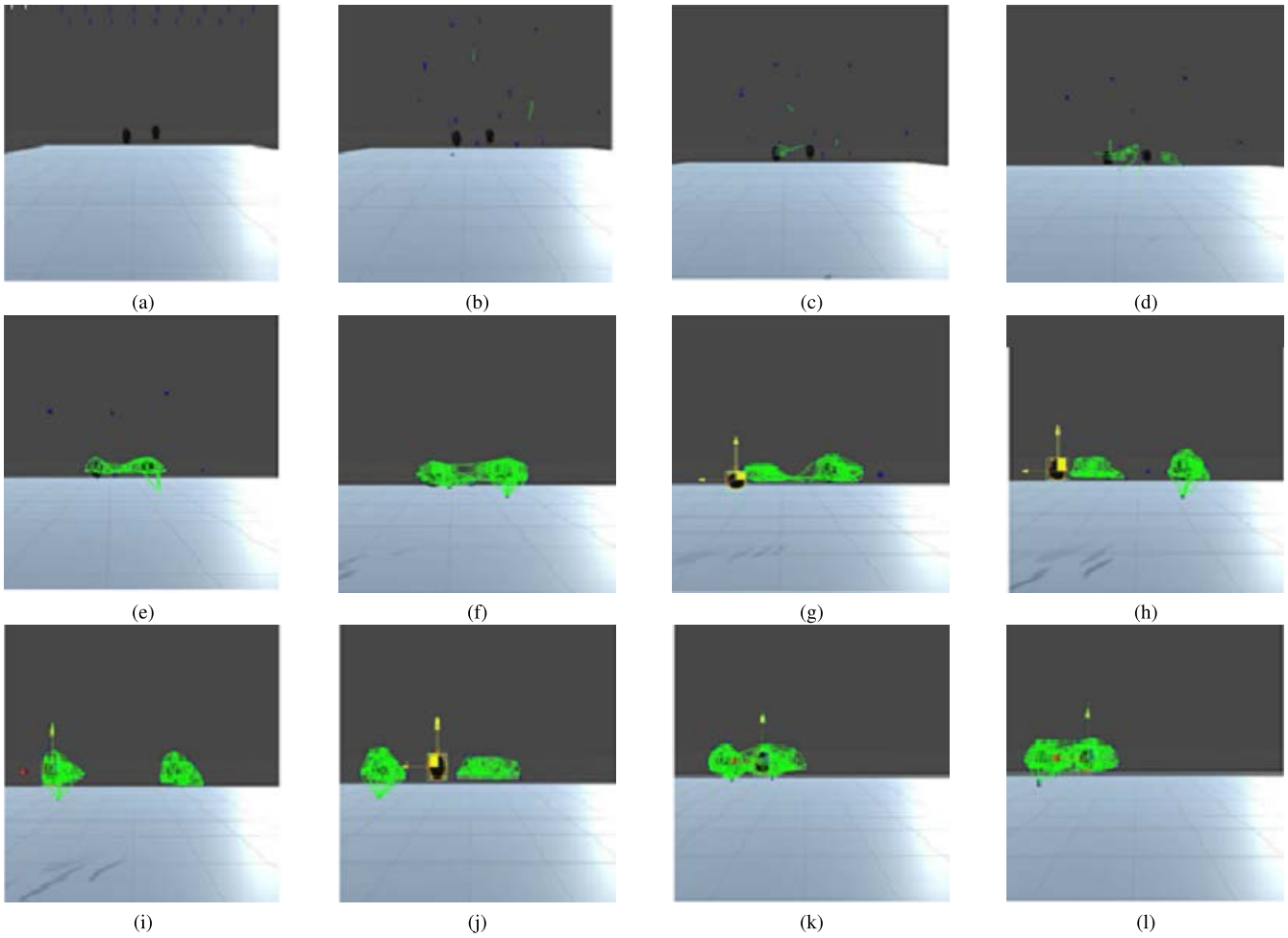


FIGURE 18. Single swarm surrounding multiple static targets.

When both objectives were assigned the same priority in the third iteration, the swarm got confused, especially when the majority of the drones were between the two targets, shown in Fig.17. This was due to the fact that advancing towards one goal resulted in positive reward, while moving away from the other target resulted in negative reward. As a

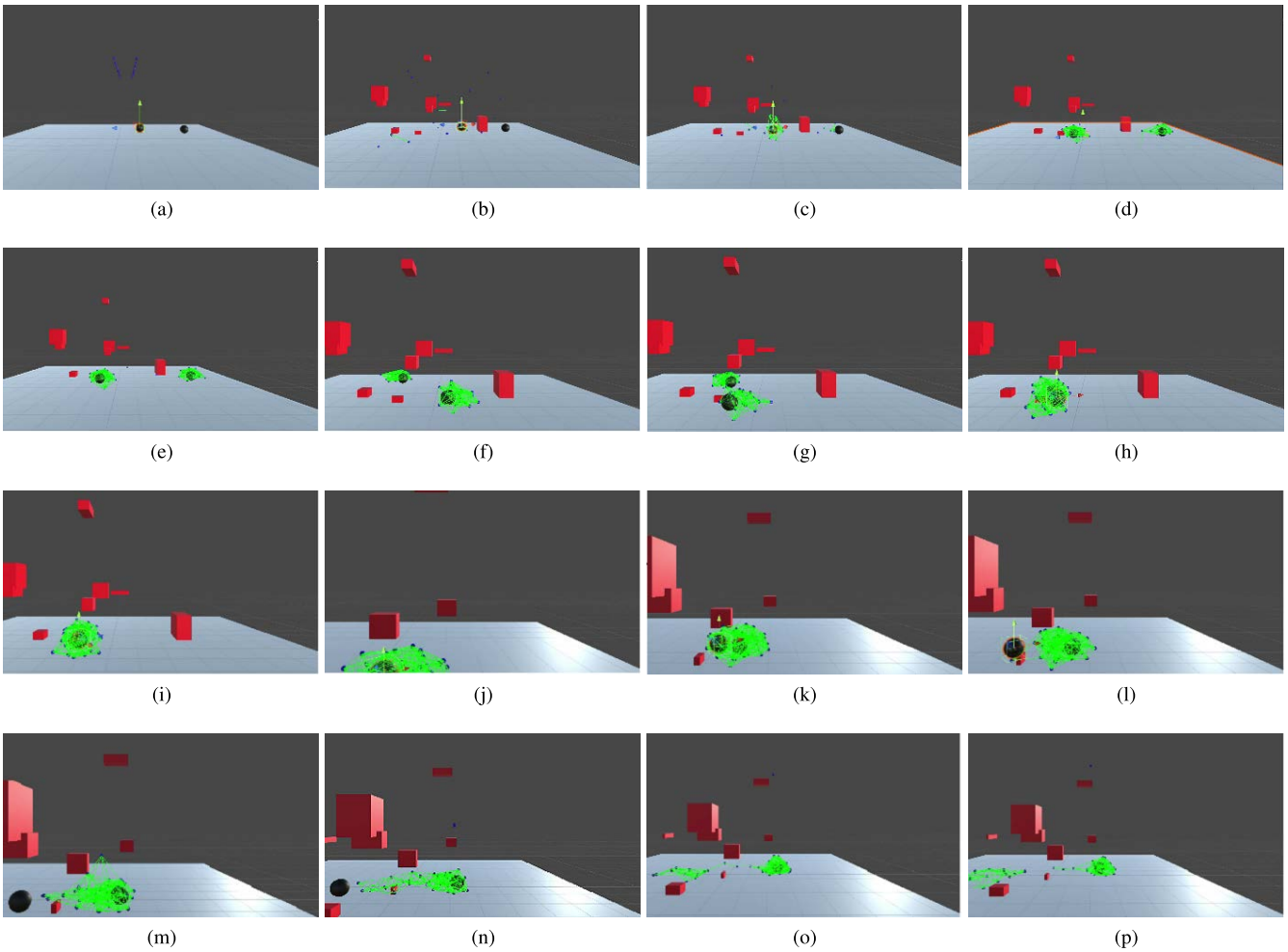


FIGURE 19. Dynamic swarm tracking multiple mobile targets in a complex environment.

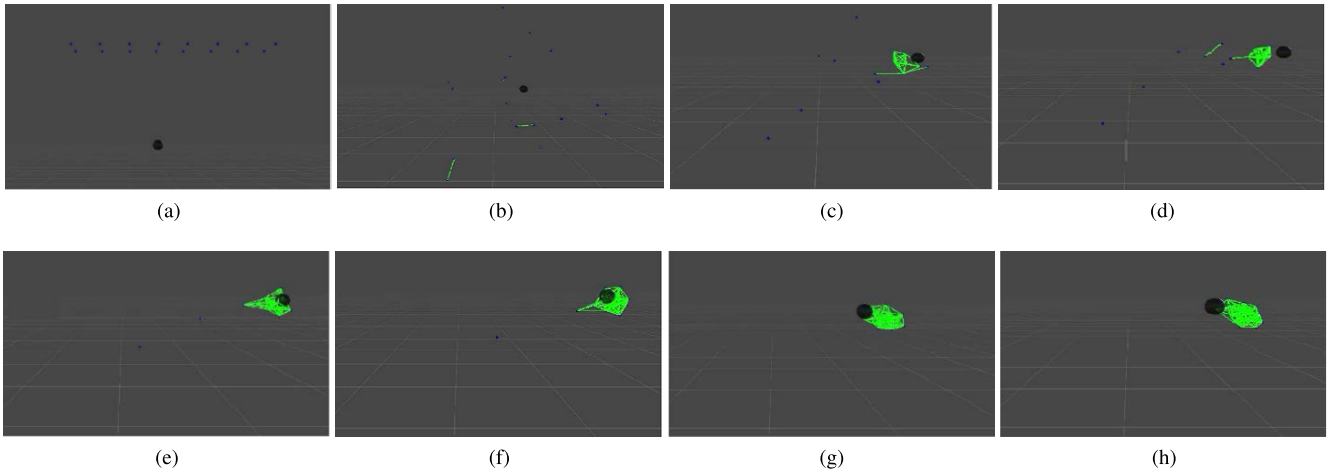


FIGURE 20. Simulating Environmental Factors like Gravity, Linear and Angular drag and their effect on drone swarm.

result, swarm performed poorly in this scenario. The notion of sub-swarms was established to address these concerns.

F. EXPERIMENT 5: MULTIPLE SWARMS AND TARGETS

In this experiment, two static targets were placed close to each other in the environment whose volume is $1000units_3$ (Fig.18(a)). For visual reasons, the size of the targets was enhanced, but, for our swarm, they were simply dots in three-dimensional space in all experiments so 1 unit out of 1000,000,000 units. Swarm was successful in locating both targets and surrounding them (Fig.18(b-f)). To visualize the concepts of large single swarm (Huge Swarm composed

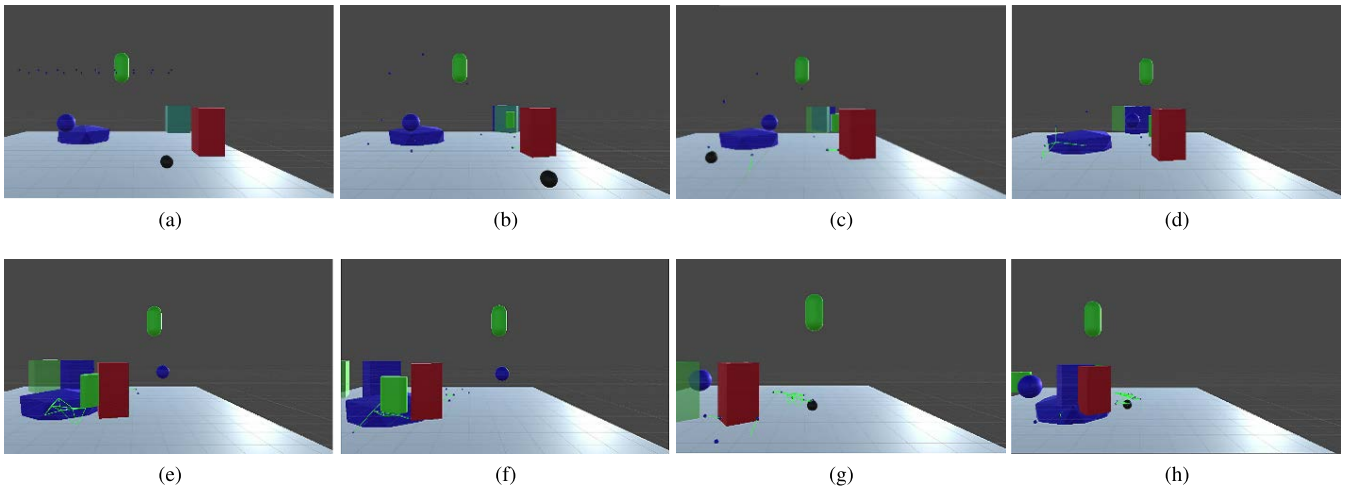


FIGURE 21. Dynamic Obstacles with Fast Moving Target, drone swarm shows impressive evasion properties.

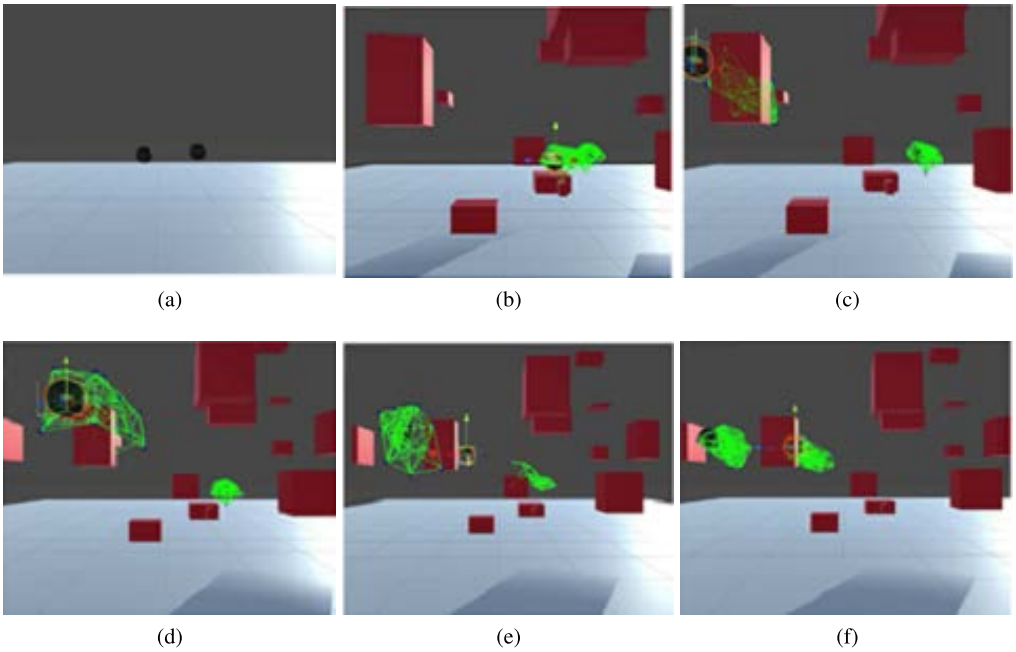


FIGURE 22. Complex Environment containing multiple dynamic targets in a cluttered environment while environmental factors like gravity, linear and angular drag are in effect.

of many sub-swarms), dividing that into small swarms and vice versa, distance between the two targets was increased (Fig.18(g)). Swarm was able to divide itself into two smaller swarms (Fig.18(h-i)). To recreate a single swarm by combining these sub-swarms, targets were brought close together (Fig.18(j)). Sub-swarms tailing targets got close to each other and as a result a single swarm was formed (Fig.18(k-l)). As shown in Fig.19, two dynamic targets were added to the environment and, after the swarm localized them, they were made to move away from each other. The swarm divided itself into smaller swarms to track both targets going in different directions. Then, both targets were moved closer to each other. This caused the sub-swarms to gradually merge again to form a single swarm.

G. EXPERIMENT 6: MULTIPLE TARGETS IN A COMPLEX ENVIRONMENT WITH DYNAMIC OBSTACLES AND ENVIRONMENTAL FACTORS

In this experiment, to check our model's robustness against environmental factors. Factors like gravity, linear drag, and angular drag were modeled in this study. Gravity was modeled at a value of $9.81 m/s^2$ to mimic the earth's surface environment. Linear and angular drag values were set at 0.25N and 0.15Nm, respectively. Detailed dynamics will

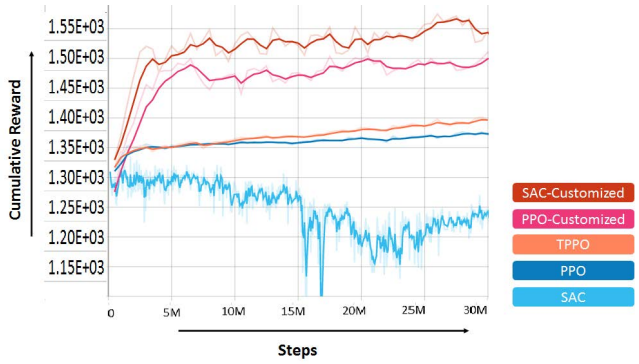


FIGURE 23. Comparison between existing state-of-the-art and our customized models.

be addressed in future work. Despite the addition of these factors, our model navigated the environment successfully with minimal difference compared to its absence as shown in Fig.20.

Dynamic obstacles were introduced as shown in Fig.21(a). Ground layer here was only present to make visualization better and doesn't have colliders. Target velocity was set as very high and even obstacle movement velocity was identical to the velocity of drone swarm in order to observe behavior in chaotic environments. Drones still exhibited swarm-like formation and behavior (Fig.21(f-g)), tried to track target but didn't have a lot of success due to high velocity of target. Interestingly drones avoided collisions most of the time using backward movement (Fig.21(e)) and even moving away from incoming obstacles when necessary (Fig.21(f)).

Furthermore, the number of obstacles and targets is increased, object morphology is changed, and some additional environmental factors are added to validate the scalability of the proposed approach, as seen in Fig.22. Gravitational value is kept at 9.81 m/s^2, but the linear and angular drag force is increased to a value of 0.37N and 0.25Nm, respectively. Swarm slowed down at drag values exceeding 0.5.Our model was robust enough to tackle this complex environment and found targets and tracked them successfully.

VI. COMPARATIVE ANALYSIS

Comparison between PPO, TPPO, and SAC original architectures and with our customized architectures is presented in Table 8. The number of layers and neurons in each layer are increased to deal with the complexity of the training environment while other hyper-parameters are optimized for training efficiency. SAC is highly dependent on hyper-parameters, so a number of different arrangements were tested, and the best was selected for the final experiments. Furthermore, LSTM is leveraged to help the swarm remember the best paths and aid in obstacle avoidance which slightly improves the overall approach. Also, comparative results shown in Fig 23 show that our technique performs better than existing techniques.

TABLE 7. Experiments summary.

S.No.	Experiments	Description
$01-a$	Swarm behavior and target localization	Forming a Swarm Organization and maintaining it while locating the target object.
$01-b$	Obstacle avoidance	Agents avoid complex obstacles in a 3D environment.
02	Target surrounded by obstacles	Obstacles surround the target object from five sides. Only one path is available for drone agents to reach the target object.
03	Navigation through a small hole	Drone agents passing through a hole single-file.
04	Target behind an obstacle	The target object is behind a wall and drone agents must find the shortest path to reach it.
$05-a$	Multiple targets (Priority-based)	Single swarm, multiple targets, priority based, one target is given higher priority then the other.
$05-b$	Multiple targets (Alternating Priority)	Single swarm, multiple targets, priority based, similar to last experiment but priority is changed between target after certain time.
$05-c$	Multiple targets (Same Priority)	Instead of assigning differing priority-based targets, same priority is given to all targets.
$06-a$	Multiple targets	Multiple targets are present, drone swarm needs to surround all of them.
$06-b$	Swarm Subdivision and Combination	Swarm divides itself into smaller swarms known as sub-swarms in other to track multiple targets going in different directions and sub-swarms can combine again when they come close to each other.
$07-a$	Multiple Dynamic Target objects	Targets are all moving, drone swarm not only needs to surround but must keep surrounding it while they are on the move.
$07-b$	Environmental Factors	Drone agents' robustness to environmental factors like gravity, linear drag, angular drag
$07-c$	Fast moving obstacles	Drone swarm shows impressive evasion properties even when obstacles are moving at high speeds and still manages to maintain somewhat of a formation while trying to track extremely fast moving target
$07-d$	Multiple target objects in a complex environment	Dynamic multiple target tracking but in a complex environment with a lot of obstacles and environmental factors added in. The number of obstacles is increased along with target objects to check the scalability of our proposed methodology.

TABLE 8. Performance comparison with Existing algorithms.

Algorithm	MCR	No. of Layers	Neurons
PPO	1348	2	64
TPPO	1327	2	128
Customized-PPO	1448	3	128
SAC	1223	2	256
Customized-SAC	1550	2	128

VII. CONCLUSION

An efficient methodology to train homogeneous swarm agents is presented for obstacle avoidance and navigation towards multiple targets in complex dynamic 3D

environments. A compact vector representation is proposed for presenting state data to our network. This generalizes the behaviors of our drones independent of their proximity to other drones. Furthermore, an appropriate incentive mechanism employing reward functions was developed to carefully design collision-free navigation while maintaining the swarm's connection. Also, the problem of multi-target tracking, where our swarm can track multiple targets while maintaining formation and communication within the swarm, was tackled. The concept of dynamic swarms was introduced, where a swarm can be divided to track more than one target simultaneously, and also if targets are removed, sub-swarms can combine to form a single larger swarm. Also, even when there are multiple targets in the environment in close proximity to each other, sub-swarms can combine into a single swarm for that duration. When targets move away, the swarm can again sub-divide. The results demonstrate the approach's universality by testing it in various situations that a swarm may face. Our framework achieves an improvement in the cumulative reward (as compared to the existing techniques considered for comparison) from 1100-1300 to 1500-1600 range. Furthermore, we also incorporated multi-target tracking as a featured part of our framework. This strategy can be scaled up to be employed for real-world swarm applications. It can find multiple uses, for example, in search and rescue, food delivery, etc. It can also be used against terrorists, for instance, tracking their vehicles and autonomous bombardment of their bases. Similarly, it finds both constructive uses, such as food and medicine delivery, etc., and destructive uses in war, such as aerial poison bombing. It can also be used as a hovering swarm to either protect an entity or hinder it from making any movements.

REFERENCES

[1] S. H. Khan, A. Sohail, M. M. Zafar, and A. Khan, "Coronavirus disease analysis using chest X-ray images and a novel deep convolutional neural network," *Photodiagnosis Photodyn. Therapy*, vol. 35, Sep. 2021, Art. no. 102473.

[2] A. Sohail, A. Khan, N. Wahab, A. Zameer, and S. Khan, "A multi-phase deep CNN based mitosis detection framework for breast cancer histopathological images," *Sci. Rep.*, vol. 11, no. 1, p. 6215, Dec. 2021.

[3] V. Francois-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, "An introduction to deep reinforcement learning," 2018, *arXiv:1811.12560*.

[4] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018, pp. 1–8.

[5] G. Beni, "Swarm intelligence," in *Complex Social and Behavioral Systems: Game Theory and Agent-Based Models*. Springer, 2020, pp. 791–818.

[6] A. Chakraborty and A. K. Kar, "Swarm intelligence: A review of algorithms," in *Nature-Inspired Computing and Optimization*. Cham, Switzerland: Springer, 2017, pp. 475–494.

[7] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[8] O. Ertenlice and C. B. Kalayci, "A survey of swarm intelligence for portfolio optimization: Algorithms and applications," *Swarm Evol. Comput.*, vol. 39, pp. 36–52, Apr. 2018.

[9] G. Chmaj and H. Selvaraj, "Distributed processing applications for UAV/drones: A survey," in *Progress in Systems Engineering*. Cham, Switzerland: Springer, 2015, pp. 449–454.

[10] W. Power, M. Pavlovski, D. Saranovic, I. Stojkovic, and Z. Obradovic, "Autonomous navigation for drone swarms in GPS-denied environments using structured learning," in *Proc. IFIP Int. Conf. Artif. Intell. Appl. Innov.*, vol. 584. Cham, Switzerland: Springer, 2020, pp. 219–231.

[11] J. Hu, H. Niu, J. Carrasco, B. Lennox, and F. Arvin, "Voronoi-based multi-robot autonomous exploration in unknown environments via deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 14413–14423, Oct. 2020.

[12] K. A.-R. Youssefi and M. Rouhani, "Swarm intelligence based robotic search in unknown maze-like environments," *Expert Syst. Appl.*, vol. 178, Sep. 2021, Art. no. 114907.

[13] W.-C. Chiang, Y. Li, J. Shang, and T. L. Urban, "Impact of drone delivery on sustainability and cost: Realizing the UAV potential through vehicle routing optimization," *Appl. Energy*, vol. 242, pp. 1164–1175, May 2019.

[14] U. R. Devasena, D. P. Brindha, and R. Thiruchelvi, "A review on DNA nanobots: A new techniques for cancer treatment," *Asian J. Pharmaceutical Clin. Res.*, vol. 11, no. 6, pp. 61–64, 2018.

[15] D. St-Onge, M. Kaufmann, J. Panerati, B. Ramtoula, Y. Cao, E. B. J. Coffey, and G. Beltrame, "Planetary exploration with robot teams: Implementing higher autonomy with swarm intelligence," *IEEE Robot. Autom. Mag.*, vol. 27, no. 2, pp. 159–168, Jun. 2020.

[16] L. Rosenberg, D. Baltaxe, and N. Pescetelli, "Crowds vs swarms, a comparison of intelligence," in *Proc. Swarm/Hum. Blended Intell. Workshop (SHBI)*, Oct. 2016, pp. 1–4.

[17] R. Xiao and Y. Wang, "Labour division in swarm intelligence for allocation problems: A survey," *Int. J. Bio-Inspired Comput.*, vol. 12, no. 2, pp. 71–86, 2018.

[18] A. C. Stan and O. Mihaela, "Petri Nets based coordination mechanism for cooperative multi-robot system," *J. Elect. Eng., Electron., Control Comput. Sci.*, vol. 6, no. 2, pp. 7–14, 2020.

[19] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm Evol. Comput.*, vol. 50, Nov. 2019, Art. no. 100565.

[20] S.-J. Chung, A. A. Paranjape, P. Dames, S. Shen, and V. Kumar, "A survey on aerial swarm robotics," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 837–855, Aug. 2018.

[21] J. Hu, P. Bhowmick, and A. Lanzon, "Two-layer distributed formation-containment control strategy for linear swarm systems: Algorithm and experiments," *Int. J. Robust Nonlinear Control*, vol. 30, no. 16, pp. 6433–6453, Nov. 2020.

[22] J. Hu and A. Lanzon, "An innovative tri-rotor drone and associated distributed aerial drone swarm control," *Robot. Auto. Syst.*, vol. 103, pp. 162–174, May 2018.

[23] L. Cao, Y. Cai, and Y. Yue, "Swarm intelligence-based performance optimization for mobile wireless sensor networks: Survey, challenges, and future directions," *IEEE Access*, vol. 7, pp. 161524–161553, 2019.

[24] M. Schranz, G. A. Di Caro, T. Schmickl, W. Elmenreich, F. Arvin, A. Şekercioğlu, and M. Sende, "Swarm intelligence and cyber-physical systems: Concepts, challenges and future trends," *Swarm Evol. Comput.*, vol. 60, Feb. 2021, Art. no. 100762.

[25] S. H. Khan, M. H. Yousaf, F. Murtaza, and S. Velastin, "Passenger detection and counting for public transport system," *NED Univ. J. Res.*, vol. 17, no. 2, pp. 35–46, Mar. 2020.

[26] B. Taha and A. Shoufan, "Machine learning-based drone detection and classification: State-of-the-art in research," *IEEE Access*, vol. 7, pp. 138669–138682, 2019.

[27] H.-Y. Lin and X.-Z. Peng, "Autonomous quadrotor navigation with vision based obstacle avoidance and path planning," *IEEE Access*, vol. 9, pp. 102450–102459, 2021.

[28] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, "Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks," *IEEE Access*, vol. 7, pp. 59069–59080, 2019.

[29] Q.-V. Pham, T. Huynh-The, M. Alazab, J. Zhao, and W.-J. Hwang, "Sum-rate maximization for UAV-assisted visible light communications using NOMA: Swarm intelligence meets machine learning," *IEEE Internet Things J.*, vol. 7, no. 10, pp. 10375–10387, Oct. 2020.

[30] E. Chen, J. Chen, A. W. Mohamed, B. Wang, Z. Wang, and Y. Chen, "Swarm intelligence application to UAV aided IoT data acquisition deployment optimization," *IEEE Access*, vol. 8, pp. 175660–175668, 2020.

[31] M. Y. Arafat and S. Moh, "Localization and clustering based on swarm intelligence in UAV networks for emergency communications," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8958–8976, Oct. 2019.

[32] F. Dai, M. Chen, X. Wei, and H. Wang, "Swarm intelligence-inspired autonomous flocking control in UAV networks," *IEEE Access*, vol. 7, pp. 61786–61796, 2019.
[33] C. Kube and E. Bonabeau, "Cooperative transport by ants and robots," *Robot. Auto. Syst.*, vol. 30, no. 1, pp. 85–101, 2000.
[34] S. Minaeian, J. Liu, and Y. J. Son, "Vision-based target detection and localization via a team of cooperative UAV and UGVs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 7, pp. 1005–1016, Jul. 2016.
[35] A. L. Alfeo, M. G. C. A. Cimino, N. De Francesco, A. Lazzeri, M. Lega, and G. Vaglini, "Swarm coordination of mini-UAVs for target search using imperfect sensors," *Intell. Decis. Technol.*, vol. 12, no. 2, pp. 149–162, Mar. 2018.
[36] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications-swarm intelligence algorithms," *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 1004–1015, Mar. 2018.
[37] C. J. C. H. Watkins and P. Dayan, "*Q*-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, 1992.
[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.
[39] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.
[40] O. E. Barndorff-Nielsen and N. Shephard, "Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics," *J. Roy. Stat. Soc., B, Stat. Methodol.*, vol. 63, no. 2, pp. 167–241, May 2001.
[41] M. Hüttenrauch, A. Šošić, and G. Neumann, "Guided deep reinforcement learning for swarm systems," 2017, *arXiv:1709.06011*.
[42] M. Hüttenrauch, S. Adrian, and G. Neumann, "Deep reinforcement learning for swarm systems," *J. Mach. Learn. Res.*, vol. 20, no. 54, pp. 1–31, 2019.
[43] B. Li and Y. Wu, "Path planning for UAV ground target tracking via deep reinforcement learning," *IEEE Access*, vol. 8, pp. 29064–29074, 2020.
[44] M. A. Akhloufi, S. Arola, and A. Bonnet, "Drones chasing drones: Reinforcement learning and deep search area proposal," *Drones*, vol. 3, no. 3, p. 58, Jul. 2019.
[45] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.
[46] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1889–1897.
[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017, *arXiv:1707.06347*.
[48] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1928–1937.
[49] Z. Wang, V. Bapst, N. Heess, V. Mnih, R. Munos, K. Kavukcuoglu, and N. de Freitas, "Sample efficient actor-critic with experience replay," 2016, *arXiv:1611.01224*.
[50] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep RL: A case study on PPO and TRPO," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.
[51] J. Zhang, Z. Zhang, S. Han, and S. Lü, "Proximal policy optimization via enhanced exploration efficiency," 2020, *arXiv:2011.05525*.
[52] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman, "Teacher-student curriculum learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 9, pp. 3732–3740, Sep. 2020.
[53] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[54] Y. Wang, H. He, and X. Tan, "Truly proximal policy optimization," in *Uncertainty in Artificial Intelligence* (Proceedings of Machine Learning Research), Israel, 2020, pp. 113–122.
[55] P. Hamalainen, A. Babadi, X. Ma, and J. Lehtinen, "PPO-CMA: Proximal policy optimization with covariance matrix adaptation," in *Proc. IEEE 30th Int. Workshop Mach. Learn. Signal Process. (MLSP)*, Sep. 2020, pp. 1–6.
[56] G. Chen, Y. Peng, and M. Zhang, "An adaptive clipping approach for proximal policy optimization," 2018, *arXiv:1804.06461*.
[57] M. Shahbaz and A. Khan, "Autonomous navigation of swarms in 3D environments using deep reinforcement learning," in *Proc. Int. Symp. Recent Adv. Electr. Eng. Comput. Sci.*, Oct. 2020, pp. 1–6.
[58] J. Wang, Y. Tang, J. Kavalen, A. F. Abdelzaher, and S. P. Pandit, "Autonomous UAV swarm: Behavior generation and simulation," in *Proc. Int. Conf. Unmanned Aircr. Syst. (ICUAS)*, Jun. 2018, pp. 1–8.
[59] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28573–28593, 2018.
[60] D. Qu, B. Yang, and N. Gu, "Indoor multiple human targets localization and tracking using thermopile sensor," *Infr. Phys. Technol.*, vol. 97, pp. 349–359, Mar. 2019.
[61] L. Ren, J. Lu, Z. Wang, Q. Tian, and J. Zhou, "Collaborative deep reinforcement learning for multi-object tracking," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 586–602.
[62] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 1861–1870.

SULEMAN QAMAR received the B.S. degree in computer science with majors in artificial intelligence and the M.S. degree in computer science with majors in deep reinforcement learning.

He is currently working as a Research Assistant at the CIPMA Laboratory, Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan. His research interests include deep reinforcement learning, deep neural networks, autonomous navigation and tracking, swarm intelligence, biomedical informatics, and medical image analysis utilizing machine learning techniques. He received Gold Medal for both B.S. and M.S. degrees.

SADDAM HUSSAIN KHAN received the bachelor's degree from the University of Engineering and Technology (UET) Peshawar, the master's degree from the UET Taxila, and the Ph.D. degree from the Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, Pakistan. He is currently an Assistant Professor with the Department of Computer System Engineering, University of Engineering and Applied Science (UEAS), Swat, Pakistan. His research interests include computer vision, deep neural networks, machine learning, medical image analysis, deep learning in cyber security, and deep reinforcement learning.

MUHAMMAD ARIF ARSHAD received the bachelor's degree in computer systems engineering from the Islamia University of Bahawalpur (IUB), in 2019, and the master's degree in computer science from the Pakistan Institute of Engineering and Applied Sciences (PIEAS), Islamabad, in 2021. He is currently pursuing his research with the CESAT. His research interests include artificial intelligence, computer vision, deep learning, and machine learning. He was the Bronze Medalist in bachelor's degree. He was the Silver Medalist.

MARYAM QAMAR received the B.S. degree in computer science from The University of AJ&K, Muzaffarabad, Azad Jammu and Kashmir, Pakistan, in 2013, and the M.S. degree in computer science from the National University of Sciences and Technology, Islamabad, Pakistan, in 2017. She is currently pursuing the Ph.D. degree in computer science with Kyung Hee University, South Korea.

Since 2019, she has been a Lecturer with the Department of Computer Science and Information Technology, The University of AJ&K. Her research interests include artificial intelligence, machine learning, image and video processing, and computer vision.

JEONGHWAN GWAK received the Ph.D. degree in machine learning and artificial intelligence from the Gwangju Institute of Science and Technology (GIST), Gwangju, South Korea, in 2014. From 2002 to 2007, he worked for several companies and research institutes as a researcher and a chief technician. From 2014 to 2016, he worked as a Postdoctoral Researcher with the GIST, and from 2016 to 2017, he was a Research Professor. From 2017 to 2019, he was a Research Professor with the Department of Radiology, Biomedical Research Institute, Seoul National University Hospital, Seoul, South Korea. In 2019, he joined the Korea National University of Transportation (KNUT), as an Assistant Professor, and since 2021, he has been an Associate Professor. He is currently the Director of the Applied Machine Intelligence Laboratory. His current research interests include deep learning, computer vision, image and video processing, the AIoT, fuzzy sets and systems, evolutionary algorithms, optimization, and relevant applications of medical and visual surveillance systems.

ASIFULLAH KHAN became a Full Professor, in 2016. He has more than 21 years of research experience and is currently working as a Professor with the PIEAS. He has been the Head of the Department of Computer and Information Sciences, PIEAS, from 2016 to 2019, where he is also the Head of the PIEAS AI Center. In the field of machine learning and pattern recognition, he has 110 international journals, 53 conferences, and nine book chapter publications with more than 6000 citations to his credit. He has successfully supervised 20 Ph.D. scholars so far and is on the panel of reviewers of 53 ISI international journals. He has won eight research grants as a principal investigator. His research interests include machine learning, deep neural networks, image processing, and pattern recognition. He has been awarded the President's Award for Pride of Performance in 2018. In addition, he has received four HECś Outstanding Research Awards and one Best University Teachers Award. He has also received PAS-COMSTECH Prize 2011 in computer science & IT. He has received Research Productivity Awards from the Pakistan Council for Science and Technology (PCST) in 2012, 2013, 2014, 2015, and 2016.