Multi Drone RL Strategy for Swarm Formation, Survival and Narrow-Alley Navigation

Ananya Gandhi 20319

The following research report discusses upon a novel reward shaping and zone-division structure, and a training strategy in multi agent reinforcement learning for the purpose of enabling adaptivity, self-organizing navigation and autonomy in drone swarms. We discuss of of developing robust multi-agent behaviours which are capable of dynamic formations, obstacle avoidance, and smooth leadership transfers in complex dynamic environments, while maintaining our swarms function in a decentralized manner. This research hopes to find its direct applications in a multitude of missions like search-and-rescue, emergency food delivery and other distributed pay-load deliveries where we want autonomous swarm behaviours to be intelligent enough to trust them at the stake of the lives/materials of the parties involved.

I. INTRODUCTION

With the emergence of autonomous robotic systems coming into play, as well as the the various researches gaining inspiration from the natural collective intelligence and distributed problem-solving from the biological as well as the computational perspectives, we try to have some fun by trying to develop artificial swarm algorithms in a fresh fashion. We try to make our swarms fairly equal to natural swarms in terms of the capability of autonomy and intelligence when it comes to complex navigation, obstacle avoidance and the integrity of the swarm as an unit while including elements which might or might not pertain to the reward-desiring nature of the *homo sapiens*. Simply put, we might make these swarms work for us for missions for the collective good or for collecting goods.

Integrating swarm strategies into drones, will give them the power of emergent behaviours: where individual agents can collectively exhibit behaviours that transcend their individual capabilities. We shall delve into the very interesting world of the *Multi-agent Reinforcement Learning (MARL)*, using it to leverage our swarms to be able to navigate complex and dynamic environments while maintaining swarm integrity and resilience, not just blindly, but intelligently.

The fundamental challenge in creating such intelligent swarm systems is not merely about programming an individual agent behaviour but to do so with a comprehensive framework which will enable the emergent, collective behaviour.

The traditional control mechanisms for swarm systems have been majorly centralized systems rendering them vulnerable to single points of failure, size-constraints due to exponential computational overload, thus reducing scalability. These systems struggle to respond dynamically to dynamic environments. While one might propose using physio-metric potential fields, these still hold their limitations for dynamical environments, higher computational complexities, as well as lacking complex collective behaviour component we aim for. Another disadvantage is that agents have higher chances of getting stuck in local minimas and find unable to escape them. Algorithmically speaking, intelligence during dynamically changing surroundings is positively correlated with performance during exploration missions in new, unknown environments.

In contrast, decentralized systems hold potential for creating robust, adaptive systems capable of autonomous decision-making and self-organization as we bestow on our swarms the blessing of continual-learning through MARL.

Our research here, addresses this challenge through a novel approach to reward-shapong and zone-division structures in reinforcement learning. By reimagining how drone agents perceive and interact with their environment, we aim to develop multi-agent behaviours that can seamlessly adapt to complex scenarios. The proposed methodology goes beyond the conventional swarm navigation techniques by introducing mechanisms for dynamic formation maintenance, smart obstacle avoidance, and smooth leadership transitions.

The motivation for this research stems from critical real-world applications where autonomous swarm behaviour could potentially save lives or enable us to overcome bodily or logistical limitations. Search-and-rescue missions, for instance, often involve navigating treacherous, unpredictable environments where traditional human-controlled methods are inefficient or dangerous. Emergency food delivery in disaster-stricken regions, payload distribution in challenging terrains, and reconnaissance in hazardous zones are just few of the example scenarios where intelligent, autonomous drone swarms could very well revolutionise the operational strategies.

Moving on to the technical aspects, our approach leverages the Proximal Policy Optimisation (PPO), a Reinforcement Learning Algorithm; within our own customised environment (till now, at least) in the Unity Game Engine. We used the ML-Agents Library and C# language to create a framework allowing drone agents to learn and adapt through sophisticated reward mechanisms. We take into consideration, also the hardware limitations in terms of sensors in drone machines as well. By implementing a unique zone-division structure, we will enable our drones to develop a nuanced understanding of proximity, collision risks and collaborative behaviours. The core innovation lies in the Intra-Swarm Rebound Reward Component- a mechanism that not only maintains our swarm integrity but also facilitates exploration and we hypothesis for it to enable a naturally smooth leadership transitions in our corresponding future work.

Succinctly put, the research contributes to the field by addressing several critical challenges in multi-agent systems by:

1. Developing decentralized control mechanisms that maintain collective coherence.

2. Creating adaptive navigation strategies that work in complex, dynamic environments.

3. Implementing robust leadership transfer protocols without centralized coordination.
4. Designing reward structures that encourage both collective stability and individual exploration.

Moreover, our methodology introduces a novel solution to the persistent challenge of handling variable drone observations in neural network architectures.

In the subsequent sections, we will cover our methodology, present empirical results, discuss more on our implementation strategies, and explore the broader implications of our approach, followed by the future works and finally, the conclusion.

II. METHODOLOGY

Research Framework Overview

Our research employs a multi-agent reinforcement learning (MARL) approach to develop an autonomous drone swarm navigation system. The methodology is structured around four primary components:

1. Training Environment Design
2. Reward Shaping Mechanism
3. Neural Network Architecture
4. Training Strategy

A. Training Environment Design

We developed a custom training environment using the Unity ML-Agents framework, specifically designed to simulate complex navigation scenarios. The environment is not just a simulation, but a sophisticated platform that:

- Introduces randomized obstacle configurations
- Provides multiple training areas to enhance learning convergence
- Implements variable drone sensing zones
- Incorporates dynamic environmental constraints

The core motivation is to create a learning space that mirrors real-world complexity, allowing our drone swarms to develop adaptive strategies that transcend their individual capabilities.

B. Reward Shaping Mechanism: The Heart of Intelligent Learning

Our reward shaping strategy is the most novel aspect of this research. We've developed a comprehensive reward structure categorized into multiple components that go beyond traditional binary reward systems:

1. Spatial Reward Components:

- Proximity-based rewards
- Zone-based reward mechanisms (good region rewards, bad region penalties)
- Sensing zone rewards that encourage strategic positioning

2. Collision Management Rewards:

- Obstacle collision penalties
- Intra-swarm collision Penalties
- Boundary Collision Penalties

3. Swarm Integrity Rewards:

- Swarmation reward for cohesive formation
- Intra-Swarm Rebound Reward Component - a groundbreaking mechanism that:
  - Enables collective exploration
  - Facilitates natural leadership transitions
  - Maintains swarm structural integrity

The Intra-Swarm Rebound Reward is particularly innovative. When one drone receives a reward, other drones in the swarm receive a proportional "rebound" reward, promoting collective learning and exploration.

C. Neural Network Architecture: Handling Complexity Dynamically

We employ a decentralized neural network architecture that addresses critical challenges in multi-agent systems:

- Variable drone observation handling
- Randomized input mechanisms to mitigate ordering biases
- Fixed-size input preservation
- Utilizing Proximal Policy Optimization (PPO) algorithm
- Centralised Learning with one global neural network shared by all agents
- Each agent, although has its own observation space
- Shared learning parameters across the swarm

A unique innovation is our approach to handling variable neural network inputs. Instead of using zero-fillers, which can mislead the network, we leverage the relativistic nature of neural networks by introducing randomized inputs during training.

D. Training Strategy: Progressive Skill Development

Our training methodology follows a structured, multi-phase approach:

1. *Initial Training Phase:*

- Deploy drones in random configurations
- Allow exploratory random movements
- Establish basic swarm formation capabilities

2. *Advanced Training Phase:*

- Introduce waypoint navigation
- Implement leader-follower dynamics
- Gradually increase environmental complexity

E. Observation Space

- Relative positions of neighboring drones
- Obstacle configurations
- Environmental boundaries
- Current swarm formation status

F. Action Space

- Continuous movement vectors
- Rotation capabilities
- Formation maintenance actions

G. Computational Considerations

- Utilized Unity ML-Agents framework
- Proximal Policy Optimization implementation

III. IMPLEMENTATION SPECIFICS

The implementation focused on creating a simulation environment, defining drone configurations, and developing robust reward mechanisms to achieve the desired swarm behavior. Each component of the implementation was designed to replicate real-world conditions and challenges.

A. Simulation Setup

The simulation environment was built using the Unity ML-Agents framework, a Gaming Engine. The key features of the simulation setup included:

Randomized Obstacles: Obstacles of varying sizes and shapes were placed randomly across the environment. This required drones to dynamically adjust their trajectories, enhancing their obstacle-avoidance capabilities.

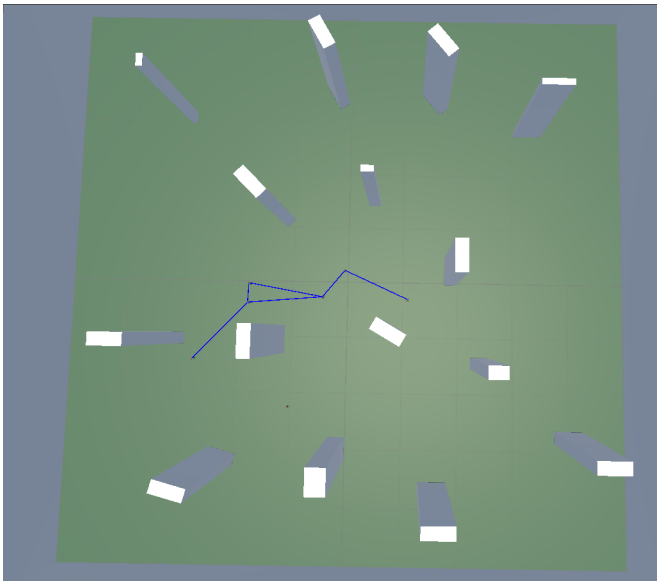

FIG. 1. Randomized environment

Dynamic Constraints: The simulation incorporated constraints like restricted communication ranges. For instance, each drone's sensing radius was capped, ensuring decentralized decision-making. These constraints tested the drones' adaptability to operate in confined and unpredictable environments, such as urban areas or disaster-stricken regions.

Training Zones: To ensure robust learning, the environment was divided into multiple zones, each with unique characteristics.

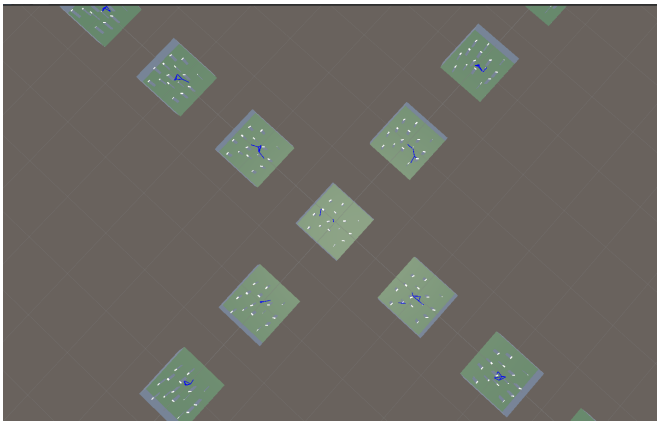

FIG. 2. Multiple Training Areas

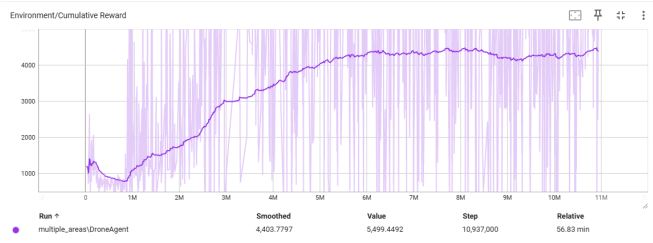

FIG. 3. Faster Convergence achieved by using multiple Training Arears

B. Drone Configuration

Each drone in the swarm was configured with predefined physical and functional properties tailored for autonomous decision-making:

Sensing Radius (r_s): This parameter defined the range within which a drone could detect other drones or obstacles. It ensured that decision-making was based on local information, fostering decentralized behaviour.

Zone Definitions: The environment was divided into spatial zones:

- **Good Region (r_{in} to r_{out}):** It is A reward zone promoting optimal inter-drone spacing to maintain swarm integrity.
- **Bad Region:** This region is sitauted between the circles with radii r_{in} and $r_{tooclose}$ with the drone at its center. I is a penalty zone where drones are at a comparative risk of collision in their next steps or inefficiencies due to overcrowding.
- **Too Close Region:** This region is situated inside the circle of $r_{tooclose}$ with the drone at its centre. It is the area immediately surrounding the drone where collisions must be avoided in order to keep the drone safe.

Action and Observation Spaces: Each drone operated with an independent neural network that processed observations like relative positions of neighbours, obstacle proximity, and boundary distances. The action space included movement adjustments in the $x, 1, z$ axes, enabling 2D navigation.

The above spatial configuration and zone-division for each individual drone allows the drone to have a nuanced understanding of its proximity to other objects: other drones as well as obstacles. Also towards collision risks, and collaborative zones while navigating through the complex environment as part of a swarm.

IV. REWARD MECHANISMS

The success of a swarm-based multi-drone system relies heavily on well-structured **reward mechanisms** that govern the learning process. In this work, a comprehensive reward-shaping framework was designed to train a swarm of drones

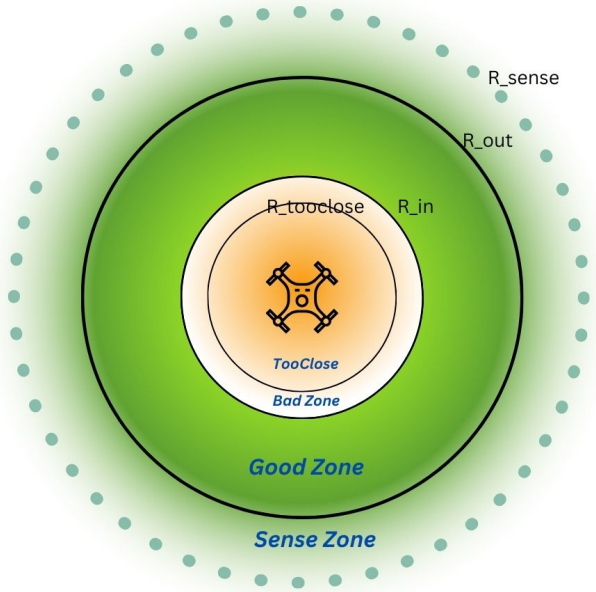

FIG. 4. Spacial Drone Configuration of an individual drone

using Reinforcement Learning (RL), ensuring stability, effective navigation, and autonomous adaptability. Below are the detailed components of the reward mechanism, incorporating equations and diagram references.

A. Reward Shaping Framework

Reward shaping optimizes drone behaviors by applying specific penalties and rewards based on environmental interactions. The primary reward components are categorized into:

- **Proximity and Collision Rewards**:
  - **TooClosePenalty** (R_{tooclose}): Applied when a drone is too close to another drone or an obstacle.
  - **ObstacleCollisionPenalty** (R_{ocp}): Imposed when a drone collides with an obstacle.
  - **Intra-SwarmCollisionPenalty** (R_{iscp}): Designed to penalize collisions between drones in the swarm.
- **Region-Based Rewards**:
  - **GoodRegionReward** (R_{good}): A reward for staying within an optimal region around the leader.
  - **BadRegionPenalty** (R_{bad}): Penalizes drones for being in undesirable areas.
- **Exploration and Formation**:
  - **SwarmationReward** (R_{swarm}): A high-value reward for maintaining swarm formation.

– **ReboundReward** (R_{rebound}): A cascading reward shared across the swarm when one drone earns the SwarmationReward.

B. Reward Functions

1. Proximity and Collision Management

The **TooClosePenalty** ensures safe drone spacing:

$$R_{\text{tooclose}} = \begin{cases} -1 & \text{if } d_{ij} < r_{\min} \\ 0 & \text{otherwise} \end{cases}$$

where d_{ij} is the distance between drones i and j, and $r_{\min}$ is the minimum safe distance.

2. Obstacle Penalties

Obstacle-related penalties are defined as:

$$R_{\text{ocp}} = -\alpha, \quad R_{\text{iscp}} = -\beta$$

where $\beta \gg \alpha$. This ensures that intra-swarm collisions are prioritized for avoidance over obstacle collisions.

3. Region-Based Feedback

To reward or penalize based on location:

$$R_{\text{good}} = +\gamma, \quad R_{\text{bad}} = -\delta$$

with $\gamma > \delta$. This incentivizes drones to stay in optimal zones while avoiding undesired areas.

```
// Reward Valuations in High to Low Order
private readonly float swarmationReward = 200.0f;
private readonly float insideGoodRegionReward = 40.0f;
private readonly float insideSensingZoneReward = 10.0f;
private readonly float intraSwarmCollisionPenalty = -80.0f;
private readonly float obstacleCollisionPenalty = -60.0f;
private readonly float boundaryCollisionPenalty = -60.0f;
private readonly float tooClosePenalty = -30.0f;
private readonly float insideBadRegionPenalty = -10.0f;
private readonly float rr_factor = 0.5f;

private bool collidedWithObstacle = false;
private bool collidedWithDrone = false;
private bool collidedWithBoundary = false;
```

FIG. 5. Reward Valuations as set in the code

4. Swarm Formation and Leadership Dynamics

Swarm Rewards: The Swarmation Reward incentivized formation behavior by rewarding drones for entering each other's sensing zones. The Intra-Swarm Rebound Reward, a novel mechanism, propagated rewards across the swarm when one drone achieved a positive outcome. This fostered collective exploration and enhanced cohesion.

The **SwarmationReward** (R_{swarm} or R_{s}) promotes cohesive movement:

$$R_{\text{swarm}} = +\eta \quad \text{if swarm integrity is maintained.}$$

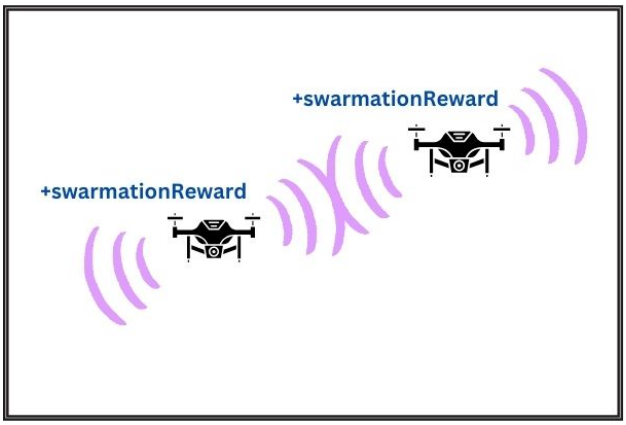

FIG. 6. Swarmation Reward Depiction

Additionally, the **ReboundReward** is distributed across the swarm:

$$R_{\text{rebound}} = \sum_{i=1}^{N} w_i R_{\text{swarm}}$$

where w_i is the weight based on the drone's position in the swarm. We set all $w_i = 1$.

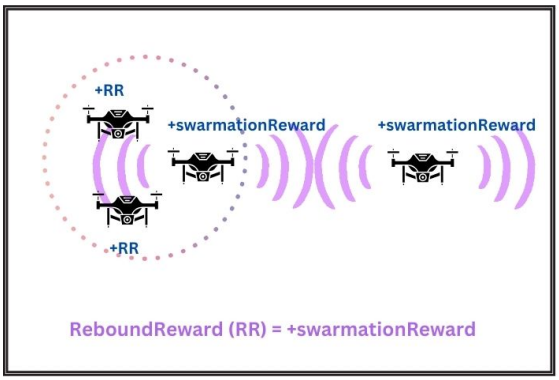

FIG. 7. Rebound Rewards Depiction

C. Prioritization of Rewards

The **Swarmation Reward** is set to a very high value, much greater than the **Collision Penalty** and **Bad Zone Penalty**. This ensures that the swarm prioritizes disintegration over the loss of its member drone. The prioritization is expressed as:

$$|R_{\text{swarm}}| > |R_{\text{collision}}| > |R_{\text{good}}|$$

To ensure proper navigation in narrow alleys, the penalties are prioritized as follows:

$$|R_{\text{bad}}| < |R_{\text{tooclose}}| < |R_{\text{ocp}}| \ll |R_{\text{iscp}}|$$

The much greater intra-swarm collision penalty (R_{iscp}) compared to the drone-obstacle collision penalty (R_{ocp}) ensures that a drone would prioritize taking itself out rather than risking collision with another drone in the swarm during extreme situations.

This hierarchy ensures that swarm integrity is prioritized over other factors, such as individual drone survival.

D. Diagram Annotations

- **Reward Zones**: The diagram illustrates sensing zones and how $R_{\text{good}}, R_{\text{bad}}$, and R_{tooclose} are applied depending on the position of drones relative to these zones.
- **Reward Hierarchy**: A flowchart highlights the decision-making logic based on the rewards and penalties.
- **Training Visualization**: Progress snapshots show improved swarm cohesion and navigation over training iterations.

E. Novelty of the Approach

The incorporation of **ReboundReward** introduces resilience in swarm dynamics, enabling:

1. Natural leadership takeover during leader drone failures.
2. Enhanced exploratory capabilities for individual drones.
3. Reduced penalty impact through cooperative learning.

The system's adaptability in high-density, narrow-alley environments is a key highlight of this methodology. By implementing this reward framework, the swarm achieves a balance between exploration, safety, and task efficiency, forming a scalable and robust system for autonomous multi-drone operations.

V. EMPIRICAL RESULTS

The empirical evaluation of the multi-drone reinforcement learning system demonstrates the effectiveness of the proposed reward mechanisms and training methodology. The results are analyzed based on several key performance indicators, supported by visualizations and statistical data.

A. Training Performance and Convergence

The training process displayed steady improvement in drone behavior over time. The convergence trends are evident in the reward progression plots:

- **Key Observations:**
  - Periodic dips in cumulative rewards signify the discovery and adoption of better policies.
  - The reward curve stabilizes as the model converges, highlighting the success of the training strategy.

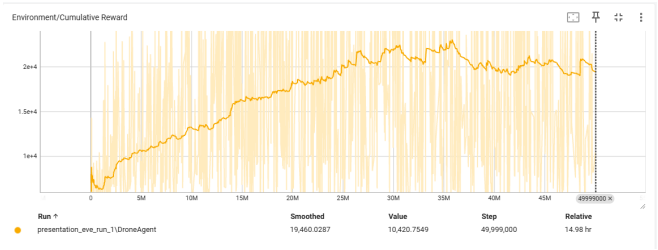

FIG. 8. Cumulative Rewards over time

B. Swarm Formation and Maintenance

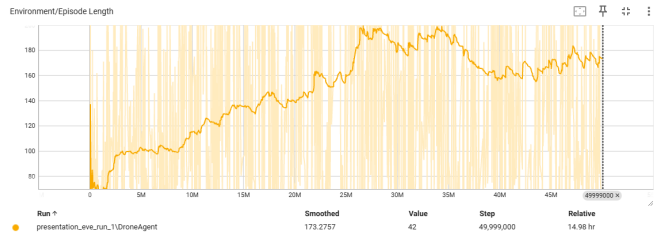

FIG. 9. Episode Length over Time

The swarm maintained its formation effectively, even under dynamically changing scenarios. This is a direct result of the **SwarmationReward** and **ReboundReward** mechanisms.

- **Visual Evidence:**
  - Training stage snapshots showcase drones learning to cluster cohesively around the leader while avoiding collisions.
  - Diagram depicting leader-following behavior confirms the robustness of swarm dynamics.

You can access the simulation videos using the following links:
https://drive.google.com/file/d/18VSLQUJHz-Gh989HCR7sdSMEIQpFEVaa/view?usp=drive$_{l}ink$
https://drive.google.com/file/d/19H5hGQXe10Tlk0RcuBX7lgwL$_{r}eS06-q/view?usp = drive_{l}ink$

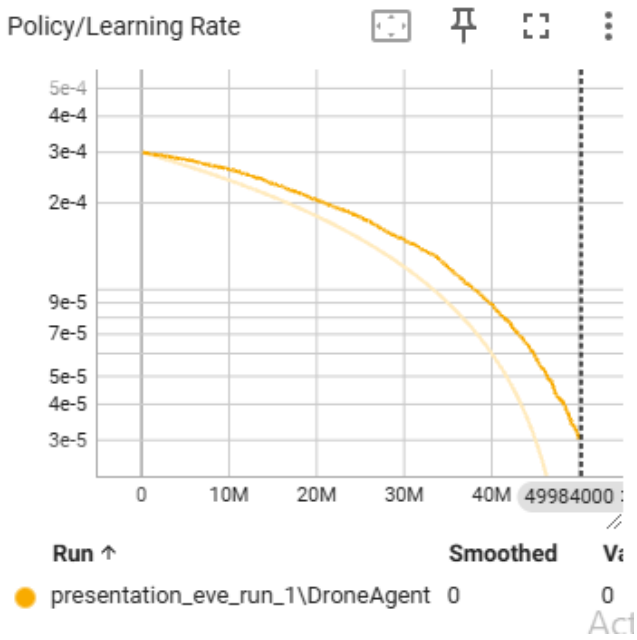

FIG. 10. Decrease in Learning Rate over time

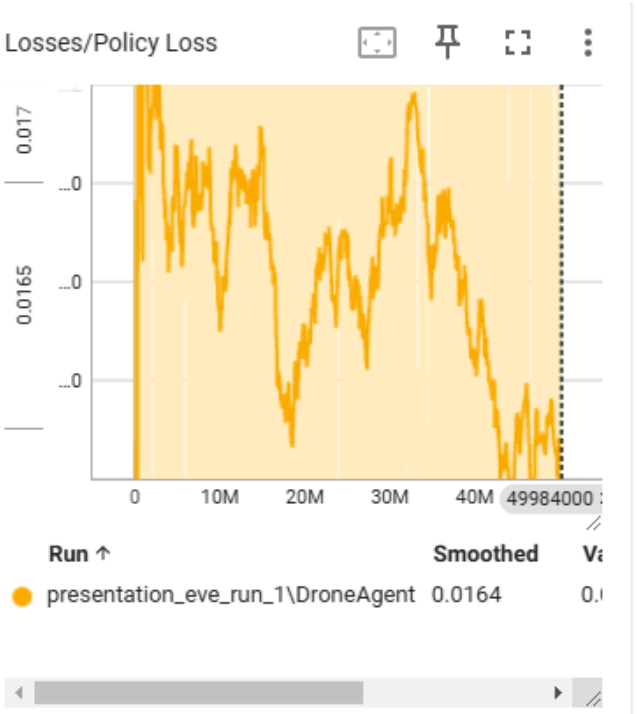

FIG. 11. Policy Loss Graph

C. Navigation in Narrow Alleys

To validate the system's capability in constrained environments, narrow-alley navigation was tested. The following observations were made:

- The swarm avoided obstacles while maintaining integrity, as shown in simulation results.
- The penalty prioritization ($R_{\text{bad}} < R_{\text{tooclose}} < R_{\text{ocp}} \ll R_{\text{iscp}}$) played a critical role in ensuring drones avoided intra-swarm collisions even in extreme scenarios.

VI. FUTURE WORKS

The proposed methodology for multi-drone swarm formation, navigation, and resilience was analyzed under diverse environmental conditions to evaluate its performance. The key areas of analysis include scalability, adaptability, computational efficiency, and robustness against failures.

Adaptability was tested in varied environments, including high-density urban areas and open spaces:

- The swarm successfully navigated narrow alleys without compromising integrity
- Drones maintained formation while dynamically reconfiguring based on the leader's trajectory.
- The swarm autonomously formed a swarm seamlessly.
- The **ReboundReward** facilitated faster reorganization and exploration post-failure.

The incorporation of **SwarmationReward** and **ReboundReward** mechanisms ensures robust swarm behavior, making this methodology well-suited for complex real-world applications.

Building on the promising results of this research, several avenues for future work have been identified to extend and refine the proposed framework:

Hardware Integration: One of the primary next steps is to implement the proposed methodology on physical UAV platforms, such as CrazyFlies or other small-scale drones. Real-world testing will validate the robustness of the framework under practical conditions, including sensor noise, hardware limitations, and environmental uncertainties.

Enhanced User Interfaces: Virtual reality (VR) integration for leader drone control is an exciting area of development. This will allow operators to interact intuitively with the swarm, monitor its progress in real time, and intervene if necessary.

Adaptive Learning in Real-Time: Future work will explore the potential of on-the-fly learning, where drones can update their policies during deployment. This approach will enable them to adapt to entirely novel environments without retraining in simulations.

Energy Efficiency and Sustainability: Research into optimizing energy consumption during swarm operations will address practical deployment challenges, particularly for missions requiring long durations or limited access to recharging facilities.

VII. CONCLUSION

In this work, we have proposed a robust and scalable methodology for multi-drone swarm navigation and formation using reinforcement learning. The approach incorporates advanced reward mechanisms, including the **SwarmationReward** and **ReboundReward**, to achieve high levels of performance in various environments, including urban areas with narrow alleys.

References

https://arxiv.org/abs/2206.08881v1