iJRASET
International Journal For Research in Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 14 **Issue:** IV **Month of publication:** April 2026

**DOI:** https://doi.org/10.22214/ijraset.2026.79764

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
*ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538*
*Volume 14 Issue IV Apr 2026- Available at www.ijraset.com*

# HiveMind AI: Emergent Intelligence through Multi-Agent Systems using PPO-based CTDE Architecture

Mohammed Muneeb[1], Syed Husamuddin[2], Mohammed Abdul Samad[3], Mr. K. Murlidhar[4]
[1, 2, 3]*Student, Department of Artificial Intelligence & Data Science, Methodist College of Engineering & Technology, Hyderabad, India*[123]
[4]*Assistant Professor, Department of Computer Science Engineering, Methodist College of Engineering & Technology, Hyderabad, India*

***Abstract:*** ***Multi-agent systems (MAS) represent an important paradigm for developing a model of distributed intelligence through interactions in a complex environment where several agents cooperate to complete a certain mission or accomplish common goals. However, due to some limitations of traditional solutions based on rule-based coordination and independent learning, the development of emergent intelligence is complicated by the lack of scalability, adaptability, and stability.***
***In this paper, the problem of emergent intelligence formation in a multi-agent system was solved by applying the concept of Deep Reinforcement Learning (DRL) to create a fully-fledged HiveMind AI simulation framework. In particular, the authors propose to use Proximal Policy Optimization in conjunction with the CTDE architecture to build a stable multi-agent system.***
***To implement the idea of emerging intelligence, the process of building the corresponding system went through several stages, which included not only designing an environment but also evaluating different approaches in order to choose the most effective and stable solution. Thus, both PPO and Soft Actor Critic algorithms were considered; in particular, the evaluation was performed in the same conditions and using the same criteria.***
***According to the experimental results, the implementation of PPO resulted in reaching 90% of successes compared to about 16% achieved when SAC was used. Therefore, PPO is a better choice when implementing CTDE-based multi-agent system. Moreover, the current implementation includes an interactive module allowing to estimate performance indicators such as reward, collision rate, synchronization level, etc.***
***Keywords: Multi-Agent Systems, Deep Reinforcement Learning, PPO, CTDE, Emergent Intelligence, Coordination, Simulation, Policy Optimization***

## I. INTRODUCTION

Multi-Agent Systems (MAS) refer to the class of intelligent computational systems characterized by multiple interacting autonomous agents operating in a joint environment towards the realization of specified goals. Such systems are extensively used in traffic optimization, swarm robots, distributed control systems, and large-scale simulations due to their applicability in modeling decentralized interactions and making decisions.

Traditional approaches to multi-agent programming involve either the use of pre-defined rules or centralized control mechanisms. In simple environments, such methods are efficient. However, they prove to be inadequate when applied to the real-life scenarios in which the number of interacting agents increases and environmental conditions become uncertain and constantly changing. In addition, the use of rules results in a system's inflexibility and inability to generalize to the unseen circumstances.

On the other hand, with the recent breakthroughs in Deep Reinforcement Learning (DRL), reinforcement learning agents can learn policies based on environmental interaction and not hardcoded instructions. However, applying DRL techniques to multi-agent settings raises additional challenges such as non-stationarity. Indeed, as other agents change their behavior, the environment becomes inconsistent for the agent under study. Finally, the agents' independent learning may lead to inefficient collaboration and learning processes.

In order to tackle these obstacles, this project presents HiveMind AI – a multi-agent framework based on the Proximal Policy Optimization method that utilizes Centralized Training with Decentralized Execution (CTDE). This technique allows one to achieve a balance between learning from all available information and taking into account only locally available data at execution time.

Thus, the purpose of the proposed research paper is to create a scalable MAS demonstrating collaborative capabilities through learning.

## II. PROBLEM STATEMENT

The fast evolution of distributed intelligent systems makes it necessary to develop flexible, scalable, and adaptive multi-agent systems able to function in ever-changing conditions. Conventional multi-agent systems mainly use either rule-based methods or central controllers, which reduce their flexibility and ability to operate in real-time environments. With increasing numbers of agents, conventional multi-agent systems become unable to coordinate effectively, affecting their performance negatively.

Innovations in Deep Reinforcement Learning (DRL) allow for designing learning-based methods of generating agents' behavior. At the same time, employing DRL in multi-agent problems leads to several key problems of non-stationarity, instability of learning, inefficient coordination, and failure to achieve convergence. In addition, independent learning is not always capable of accounting for relationships between agents, generating ineffective solutions.

Another important problem of currently used systems lies in their inability to provide an integrated platform for simulating and coordinating agents' actions, learning, testing, and visualization of results. Furthermore, there is a lack of experimental results obtained while comparing the effectiveness of different reinforcement learning algorithms in a multi-agent environment.

Thus, one of the primary issues to be solved in this project involves the creation of a scalable, stable, and efficient multi-agent learning framework enabling coordination of the agents' actions with reinforcement learning.

## III. LITERATURE SURVEY

In recent years, a lot of development has been seen in the area of MARL. Research efforts have been made towards improving MAS coordination, stability and scalability. The earliest approach to MAS was the rule-based one in which the actions of agents were programmed. While such systems produced deterministic outputs, they were inflexible and could not adapt to dynamic environments.

Independent reinforcement learning allowed each agent to learn their own policies; however, the downside was that it led to the problem of non-stationarity. As a result, the environment would be non-stationary relative to the other agents since each one constantly learns and updates their policy.

The way to deal with this challenge was through centralized training approaches in which a centralized critic knows about all aspects of the environment while learning an algorithm. CTDE is now widely accepted in MARL, as it combines centralized learning and decentralized execution.

Various RL algorithms have been applied within MARL frameworks. Among them are PPO and SAC. PPO is praised for its stability as the clipped loss makes sure the update to the policy is not overly large. In contrast, SAC adds an entropy-based term to improve exploration of the state space but is prone to causing instability.

Recently, the issue of designing the environment to allow agents to coordinate has attracted attention. One of the most important issues is choosing proper reward functions since incorrectly defined ones often result in undesired results.

There is no unified solution for MARL that would cover simulation, learning and evaluation and visualize results. Apart from that, there are not many comparisons of different algorithms in a multi-agent setting, especially when it comes to comparing PPO and SAC specifically.

## IV. RESEARCH GAP ANALYSIS

However, despite the advances in the field of multi-agent reinforcement learning, some essential problems remain unsolved and deserve further attention and discussion.

The first issue concerns the ability of traditional multi-agent reinforcement systems, which rely on pre-set rules or centralized control and cannot operate in a dynamically changing environment and adapt to changing circumstances. Thus, those systems cannot benefit from experience acquired through interactions and changes in their environment.

The second problem lies in the nature of independent reinforcement learning. It results in non-stationary conditions as the actions of agents influence each other's training in the process of learning. Moreover, this method leads to slower learning processes and weak coordination among actors. Some researchers suggest using centralized learning techniques; however, their use in practical solutions raises many questions.

Thirdly, even though algorithms like PPO and SAC have proven effective in single-agent reinforcement learning, there are no reliable comparative evaluations of their efficiency in multi-agent reinforcement learning settings. The main reason is that many studies lack proper metrics used for evaluating the performance of particular algorithms.

Furthermore, many of the frameworks discussed above only include simulations or learning components, without providing a combined system for testing the environment, algorithm implementation, learning process, evaluation of results, and visualization of findings.

Finally, the problem concerns advanced performance metrics, including coordination efficiency, synchronization, and behavior. In multi-agent reinforcement learning systems, such indicators play a key role in analyzing emergent intelligence of multiple agents.

The presented research will address the identified gaps and develop a unified framework for multi-agent reinforcement learning using PPO with CTDE architecture.

## V. PROPOSED WORK

The system design, HiveMind AI, involves the creation of a multi-agent learning architecture that utilizes reinforcement learning to induce emergent coordination among the agents. The system consists of several agents that act within a shared virtual environment, making observations about their respective local states and selecting actions by applying a shared policy network.

The central aspect of the system includes a PPO algorithm with CTDE architecture. In particular, during training, a centralized critic receives information about the global state to evaluate the overall action. This solves the issue of non-stationarity and ensures stable learning. In turn, individual agents use decentralized actors that rely on local observations when choosing an action based on a shared policy network.

Reward shaping in the system is performed using several types of rewards that encourage cooperative behavior in the environment. These include progress rewards, collision penalty, and other forms of encouragement for achieving goals in a cooperative manner.

An implementation plan involved several stages. First, a single agent environment was designed to check the effectiveness of the learning algorithm. Then, several agents in conjunction with CTDE architecture and various forms of rewards were added to the environment. Later, further optimizations were performed to improve reward and environment settings.

It is worth noting that both PPO and SAC algorithms were implemented in the environment for comparative purposes. As a result of experimentations, the PPO algorithm was chosen due to better performance characteristics.

## VI. SYSTEM ARCHITECTURE

This suggested HiveMind artificial intelligence system is based on a multi-level design where elements of multi-agent simulations, reinforcement learning, and real-time assessments are combined. Such a solution helps to provide stability and scalability in coordinating the actions of several agents at the same time, as well as efficiency in learning processes and integration of various components.

### *A. Overall System Architecture*

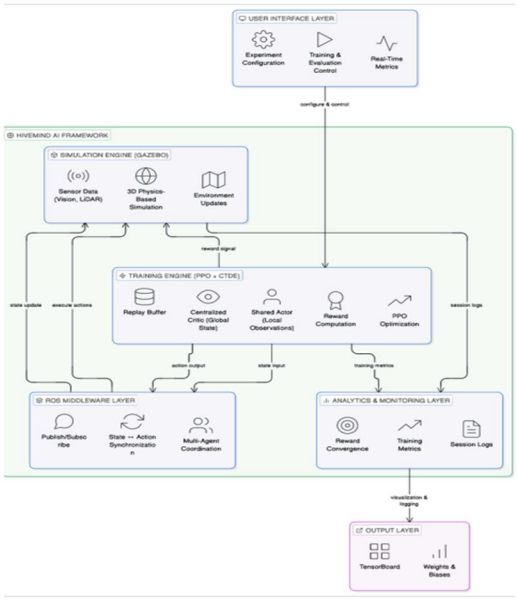



Fig 1. Architecture Workflow

This diagram presents a general overview of the overall architecture of the HiveMind AI framework, consisting of several connected parts – the user interface, training engine, multi-agent coordination module, and simulation environment.

Users work with the system via the web interface. Using current technologies for front-end development, it provides configuration of experiments, initiation of training process, and real-time monitoring of metrics. The back-end part manages evaluation processes and interaction with learning modules.

The base of the proposed system consists of a PPO training engine implementing the reinforcement learning pipeline. This component performs policy learning, optimization, and updating the parameters of a neural network. The system utilizes the Centralized Training and Decentralized Execution (CTDE) approach, where a centralized critic works during training but not during the execution stage when agents work independently.

A multi-agent coordination module allows the agents to use the same actor and critic networks. Local observations are received by an individual agent that makes actions according to their parameters. A centralized critic obtains global state information and evaluates the overall joint actions.

A simulation environment serves as a space where all agents act and receive observations, rewards, and perform interactions. It produces observations of current states of the system, computes rewards, and applies changes to the states depending on agent actions.

In this section, we describe the workflow of the proposed system. After initialization, agents start acting in the environment, obtaining local observations. They pass this information to an actor network that makes decisions about agent actions. Then, actions are applied to the environment resulting in updated states and rewards.

These states are evaluated by the centralized critic and advantages are computed using values of states. PPO algorithm optimizes policies using a clipped objective.

### *B. Learning Workflow (PPO-Based CTDE)*

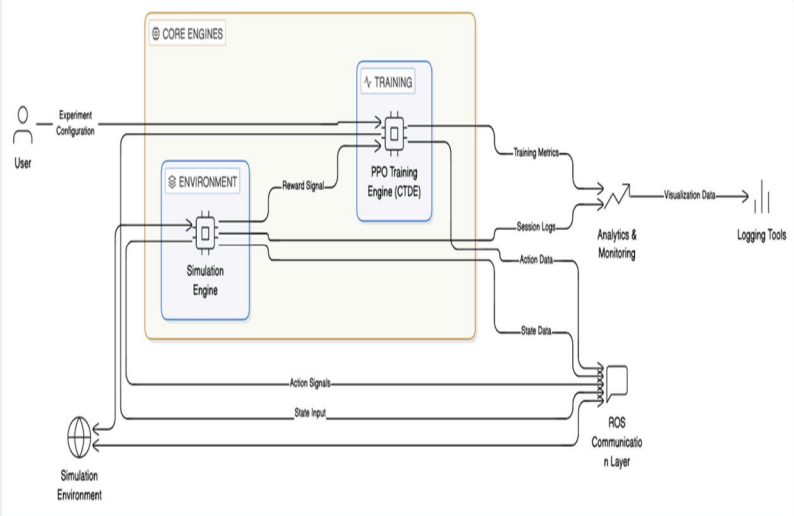



Fig 2. PPO based learning

The Learning workflow refers to how the process of reinforcement learning is carried out by the system. The agents in interactions with the environment receive trajectories containing information on states, actions, rewards, and next states.

Agents make use of their own observations for processing by means of actor network in order to produce actions. Agents perform actions in parallel and the global state received from actions is fed into the centralized critic, which assesses the state of all agents at once.

Advantages for each of agents in the environment can be estimated via the PPO algorithm, making use of value functions and rewards obtained. Advantages allow for updating the policy without exceeding a predetermined range due to the use of clipped objective function.

### *C. Multi-Agent Interaction Flow*

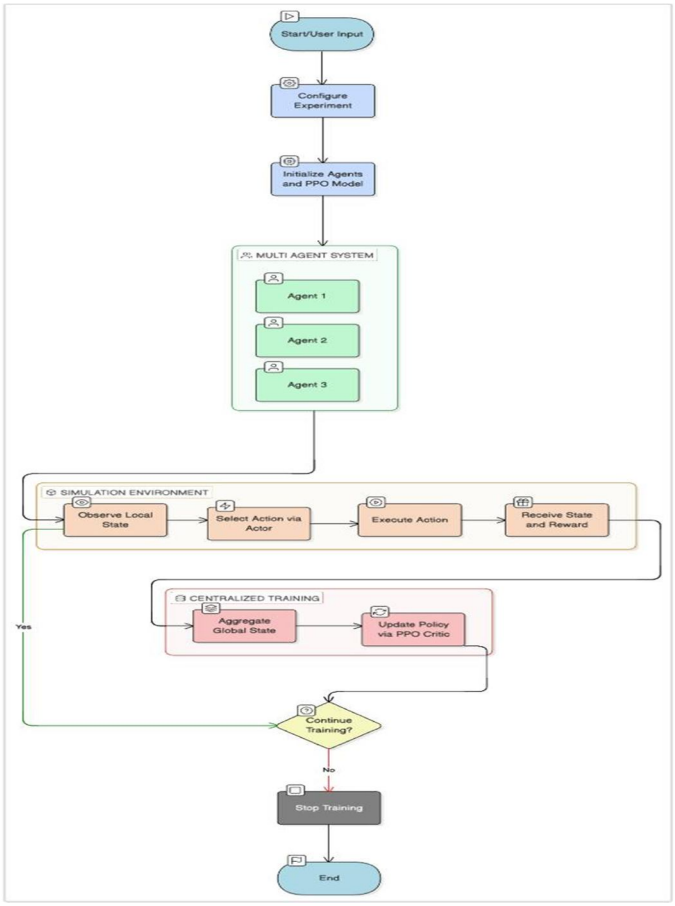



Fig 3. Multi Agent workflow

The multi-agent interaction loop explains how the agents interact with one another and behave in the environment. In the process, each agent individually senses its environment and takes actions based on the common policy model.

While there is no explicit communication between agents, there is coordination in the form of the reward scheme and centralized training mechanism. The reward model is created in such a way that agents work together by incentivizing them to make progress, punish them for colliding, and reward them when the task is completed successfully.

The role of the centralized critic becomes important in such a situation since it helps evaluate the combined state of the agents. It enables the system to recognize interactions among agents and learn to work together.

The interaction loop goes on as agents sense the environment, take actions, and learn from the environment..

### *D. System Data Flow*

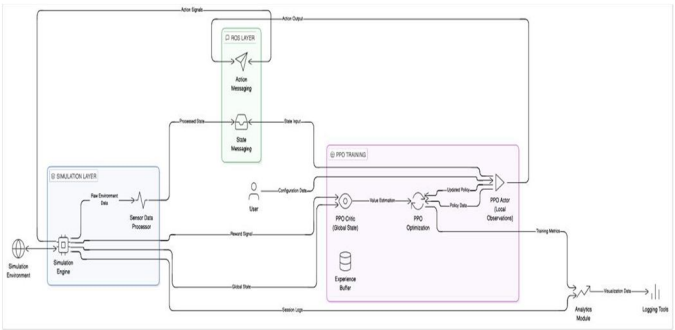


Fig 4. Dataflow Diagram

The system data flow refers to the manner in which information passes between various components within the system architecture. Initially, the environment provides state observations that are then sent to the actor network.

The actor network processes this information, making decisions based on observations, and sends out actions for every agent. The actions are sent back to the environment to be executed, and in turn, the state gets updated as well as calculated rewards according to the performed actions.

The global state is passed to the centralized critic, where it gets evaluated, providing the value of the current state to be considered. This information helps the PPO algorithm calculate the advantages and adjust the policy.

All necessary data, such as rewards, actions, and performance, are logged and visualized.

## VII. RESULTS

The HiveMind AI model has been successfully designed and tested in a controlled experiment setting. The outcomes of this test show significant advancements in multi-agent cooperation and robustness of learning stability..

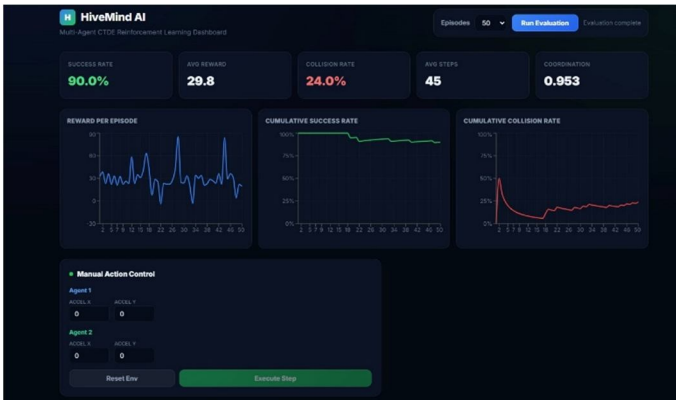



Fig 5. Evaluation metrics output

Final evaluation metrics:

- Success Rate: 90%
- Average Reward: 29.8
- Collision Rate: 0.24
- Average Steps: 44.8
- Coordination Score: 0.953
- Synchronization Score: 0.888

Reward convergence was noted as well as decreasing rates of collisions. Cooperation metrics show that emergent behavior exists, where agents learn how to cooperate without any communication..

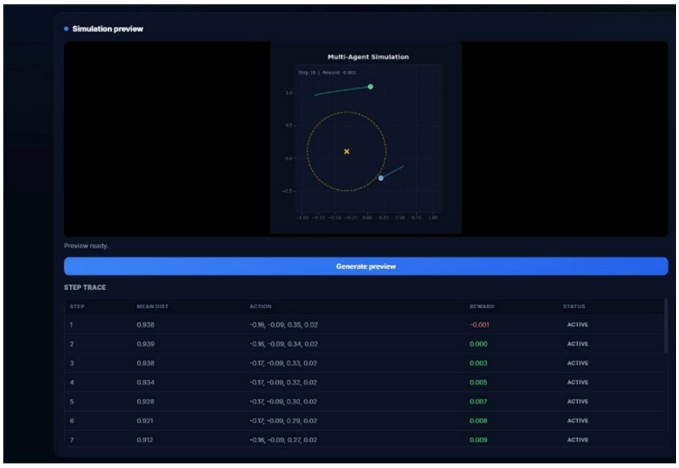
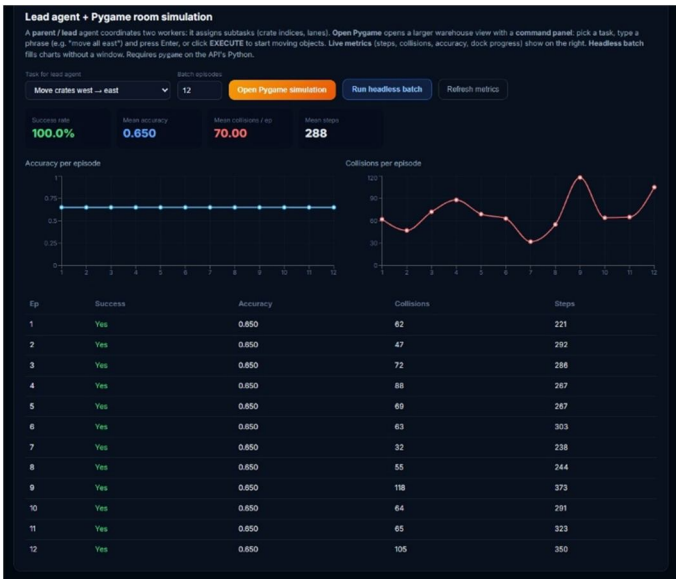



Fig 6. Agent convergence overtime

A user interface was made to visualize performance metrics and enable real-time evaluation, enhancing the usability and interpretability of the system.




| Ep | Success | Accuracy | Collisions | Steps |
|---|---|---|---|---|
| 1 | Yes | 0.650 | 62 | 221 |
| 2 | Yes | 0.650 | 47 | 292 |
| 3 | Yes | 0.650 | 72 | 286 |
| 4 | Yes | 0.650 | 88 | 267 |
| 5 | Yes | 0.650 | 69 | 267 |
| 6 | Yes | 0.650 | 63 | 303 |
| 7 | Yes | 0.650 | 32 | 238 |
| 8 | Yes | 0.650 | 55 | 244 |
| 9 | Yes | 0.650 | 118 | 373 |
| 10 | Yes | 0.650 | 64 | 291 |
| 11 | Yes | 0.650 | 65 | 323 |
| 12 | Yes | 0.650 | 105 | 350 |

Fig 7. Agent Simulation output in use case

Task assignment takes place manually, assigning the task to the parent AI, which coordinates its activities with other agents that are available for work. The division of tasks and execution take place efficiently within the coordination model created. At every instance of an activity in a controlled environment, the metrics are calculated using tensorboard and displayed on the dashboard.


Fig 8. Final Simulation Output

In order to present a visual simulation, we use pygame to show how each agent acts individually through a controlled coordination process. The collisions, accuracy, and success rate are displayed on the UI screen according to the real-time actions of success or failure and collisions that occur.

## VIII. DISCUSSION

The experiment results have shown the efficiency of using PPO CTDE approach for attaining stability in multi-agent coordination. Centralized critic helps improve learning stability by giving agents a global perspective in their learning process, whereas decentralized approach allows scaling.

In the course of implementation, PPO and SAC algorithms were analyzed within the same framework regarding the environment, reward scheme, and evaluation metrics. SAC algorithm was trained for longer timesteps and optimized through hyperparameters tuning but managed to achieve just about 16% success rate.

The SAC algorithm uses entropy exploration which made it unstable and hard for agents to develop any stable strategy towards achieving a coordinated solution.

It is vital to emphasize that proper selection of algorithm is crucial when dealing with multi-agent problems, and in this case, PPO was more appropriate than SAC.

## IX. CONCLUSION

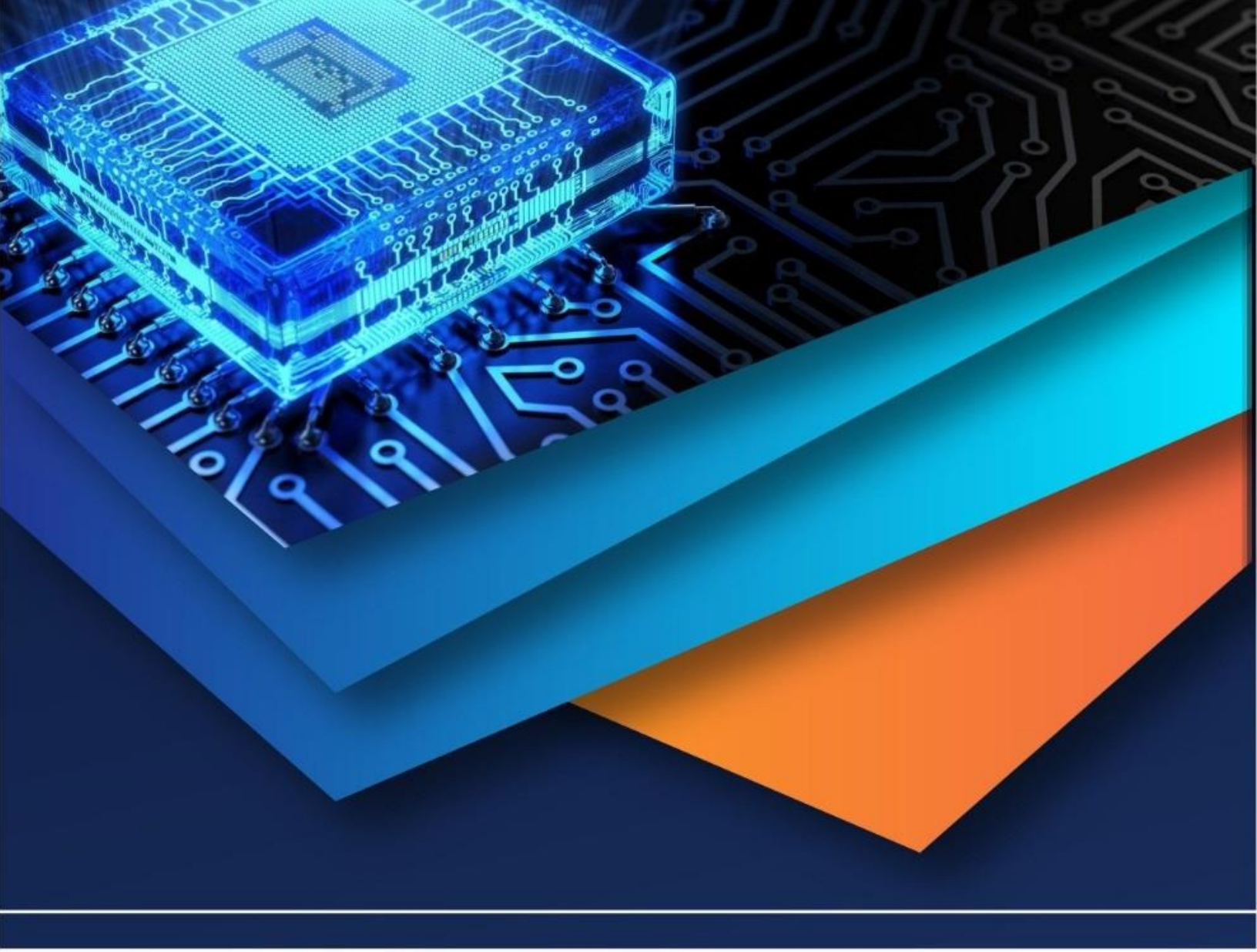
HiveMind AI is a multi-agent RL framework based on PPO CTDE architecture that aims at facilitating emergent intelligence. The framework has been successful in overcoming common problems such as instability, lack of coordination, and scalability in multi-agent settings.

Based on the well-planned implementation and thorough experiments carried out, the framework is highly efficient, showing a high level of success rate of 90% along with consistent behavior in coordination. Through comparative studies, PPO algorithm has proven to perform better than SAC algorithm in this context.

Future research could involve scalability in terms of increasing agents' numbers, introducing communication, and using the framework in practical situations.

## REFERENCES

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.
[2] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning," in Proc. International Conference on Machine Learning (ICML), 2018.
[3] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed., MIT Press, 2018.
[4] L. Busoniu, R. Babuska, and B. De Schutter, "A Comprehensive Survey of Multi-Agent Reinforcement Learning," IEEE Transactions on Systems, Man, and Cybernetics, 2008.
[5] C. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017.
[6] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual Multi-Agent Policy Gradients," in Proc. AAAI Conference on Artificial Intelligence, 2018.
[7] P. Sunehag et al., "Value-Decomposition Networks for Cooperative Multi-Agent Learning," in Proc. International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2018.
[8] M. Tan, "Multi-Agent Reinforcement Learning: Independent vs Cooperative Agents," in Proc. International Conference on Machine Learning (ICML), 1993.
[9] OpenAI, "Emergent Tool Use from Multi-Agent Interaction," OpenAI Research Blog, 2019.
[10] H. V. Hasselt, A. Guez, and D. Silver, "Deep Reinforcement Learning with Double Q-Learning," in Proc. AAAI Conference on Artificial Intelligence, 2016.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY