Benchmarking the Gap Between Theory and Practice in Multi-Agent Reinforcement Learning

Jana Angeloska
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, Macedonia
jana.angeloska@students.finki.ukim.mk

Dragana Usovikj
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, Macedonia
dragana.usovikj@students.finki.ukim.mk

***Abstract*—Multi-agent reinforcement learning has emerged as a critical framework for addressing complex cooperative tasks across robotics, resource management, and autonomous systems. [1] This study presents a systematic comparison of three representative multi-agent reinforcement learning algorithms across coordination-heavy and exploration-focused cooperative tasks. We evaluate Independent Proximal Policy Optimization, Multi-Agent Proximal Policy Optimization, and Multi-Agent Deep Deterministic Policy Gradient using the BenchMARL framework [2] integrated with the Vectorized Multi-Agent Simulator. [3] Our experiments systematically examine these algorithms' performance on the Balance and Sampling environments with agent team sizes ranging from three to seven agents. This reveals scalability characteristics and the practical value of centralized critics in different task contexts. By looking at learning dynamics, convergence behavior, and training stability across 300,000 environment interactions per trial, we establish task-dependent algorithmic selection guidelines that practitioners can use. This work contributes to bridging the gap between theoretical developments in multi-agent reinforcement learning and practical deployment considerations by providing comprehensive analysis and actionable recommendations for algorithm selection in cooperative multi-agent systems.**

***Index Terms*—Multi-agent reinforcement learning, MAPPO, IPPO, MADDPG, cooperative tasks, Balance, Sampling, BenchMARL, VMAS, algorithm benchmarking**

I. Introduction

As intelligent systems increasingly operate in multi-agent environments where coordinated decision-making is essential, the need for robust algorithmic solutions has become crucial. Applications ranging from autonomous vehicle coordination and unmanned aerial vehicle swarm operations to smart grid management and mobile edge computing demonstrate the broad relevance of cooperative multi-agent systems. [4], [5]

The Balance environment requires tight synchronization between agents to cooperatively transport a package against vertical gravity, presenting a coordination challenge where individual agent actions must be carefully aligned with teammates to maintain system stability. [3] In contrast, the Sampling environment involves parallel exploration and collection of spatially distributed resources, where agents can contribute to the collective objective through relatively independent actions in different regions of the workspace. [3] This orthogonal task structure enables us to investigate how algorithmic design choices, particularly the use of centralized critics and on-policy versus off-policy learning, affect performance across different coordination regimes.

A fundamental challenge that compounds the complexity of algorithm selection in multi-agent systems is the curse of dimensionality inherent to cooperative learning. As the number of agents increases, the joint state-action space grows exponentially, leading to severe scalability limitations that threaten both sample efficiency and computational tractability. [6] Understanding how different algorithmic paradigms handle the trade-off across team sizes is crucial for creating robust algorithm selection recommendations that work for three collaborating robots or hundreds of agents.

II. Related work

Multi-agent reinforcement learning (MARL) has seen significant advances in recent years, with several approaches emerging to address the challenges of coordinating multiple learning agents in shared environments.

A. Technologies and methods

Lowe et al. presented the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) as a foundational and basic technique for mixed cooperative-competitive contexts. The technique applies the actor-critic framework to multi-agent systems by utilizing a centralized training with decentralized execution (CTDE) paradigm. During training, each agent's critic has access to all agents' observations and actions, whereas actors only use local data during execution. This design contributes to addressing the non-stationary problem inherent in multi-agent learning. The environment seems non-stationary from each agent's perspective as a result of concurrent learning by others. [7]

Using policy gradient approaches, Yu et al. proved the "surprising effectiveness" of Multi-Agent Proximal Policy Optimization (MAPPO) in cooperative multi-agent games. Their work revealed that properly configured PPO-based methods, combined with techniques such as value normalization and shared parameters, can achieve strong performance and sample efficiency competitive with off-policy methods. The success of this relatively simple technique called into question the notion that cooperative multi-agent environments require domain-

specific algorithmic tweaks or complicated coordination protocols. [8]

Through Independent PPO (IPPO) in the context of the StarCraft Multi-Agent Challenge (SMAC), Schröder de Witt et al explored whether the centralized training mechanisms are essential for multi-agent learning. Their investigation into independent learning approaches, where agents learn without sharing information during training, provided insights into when the additional complexity of centralized training provides meaningful benefits. This work highlighted that in certain environments, independent learners with proper hyperparameter tuning can match or exceed the performance of more sophisticated joint learning methods like QMIX or MAVEN. These findings suggest that the choice of learning paradigm should be informed by the specific characteristics of the task at hand, as sharing extra sensory information can sometimes negatively interfere with learning. [9]

III. Methodology

We employ BenchMARL as the primary training library, integrated with the Vectorized Multi-Agent Simulator (VMAS) to provide a standardized and high-performance environment for multi-agent reinforcement learning (MARL). BenchMARL is the first MARL library created to enable standardized benchmarking across different algorithms, models, and environments while addressing the reproducibility crisis in the field. It is built with TorchRL as its backend and provides access to state-of-the-art implementations and high-performance execution. [2]

We utilize VMAS, an open-source framework with a vectorized 2D physics engine written in PyTorch, to model collective robotic learning. The Single Instruction Multiple Data (SIMD) execution paradigm, which VMAS uses to achieve speeds up to 100 times faster than non-vectorized simulators like OpenAI MPE, enables thousands of simultaneous environment simulations on accelerated hardware. [3]

Our focus is on the environments Balance and Sampling, which are orthogonal challenges for MARL algorithms. BenchMARL groups components into experiments that do not depend on specific implementations. This allows us to compare different algorithms fairly and systematically. [2], [3]

To ensure reproducibility, the full implementation and experiment configuration files are publicly available at the GitHub repository.

A. Algorithms

To evaluate algorithmic performance across environments with different coordination requirements, we selected three distinct multi-agent reinforcement learning algorithms that represent different approaches to the centralized training paradigm. These algorithms vary in their treatment of critic information and learning mechanisms, allowing us to investigate how different design choices affect performance in coordination-heavy versus exploration-heavy tasks.

1) IPPO: Independent Proximal Policy Optimization (IPPO) represents a decentralized approach to multi-agent reinforcement learning where each agent learns its own policy independently while treating other agents as part of the environment. IPPO extends the single-agent Proximal Policy Optimization (PPO) algorithm to the multi-agent setting by having each agent maintain its own local critic that estimates value functions based solely on local observations. Despite the theoretical limitations of independent learning, such as environment non-stationarity caused by concurrent learning of other agents, IPPO has demonstrated surprisingly strong empirical performance on cooperative tasks [9].

2) MAPPO: Multi-Agent Proximal Policy Optimization (MAPPO) extends PPO to the multi-agent setting by adopting a centralized training with decentralized execution (CTDE) paradigm. Unlike IPPO, MAPPO utilizes a centralized critic that has access to global state information during training, while maintaining decentralized actors that rely only on local observations during execution. This architectural choice allows the critic to provide more accurate value estimates by conditioning on the full environment state, which can significantly improve learning stability and coordination in tasks requiring tight synchronization between agents. Network parameters are typically shared across agents to improve sample efficiency. MAPPO has demonstrated strong empirical performance across various cooperative multi-agent benchmarks, often matching or exceeding the performance of off-policy methods while maintaining competitive sample efficiency [8].

3) MADDPG: Multi-Agent Deep Deterministic Policy Gradient (MADDPG) is an off-policy actor-critic algorithm designed for mixed cooperative-competitive multi-agent environments. MADDPG addresses the challenge of non-stationary environments in multi-agent settings by adopting a centralized training with decentralized execution framework. During training, each agent's critic has access to the observations and actions of all agents, allowing for more stable value function estimation in the presence of other learning agents. However, during execution, each agent's actor relies only on its own local observations, maintaining decentralized control. This approach enables MADDPG to handle both cooperative and competitive scenarios while mitigating the non-stationarity issues inherent in independent learning. The algorithm extends the DDPG framework to multi-agent settings through the use of experience replay and target networks for both actor and critic, which helps stabilize learning in complex multi-agent interactions [7].

The selection of these three algorithms enables us to systematically compare independent learning (IPPO), centralized critic with on-policy updates (MAPPO), and centralized critic with off-policy learning (MADDPG) across the Balance and Sampling environments. This comparison allows us to evaluate whether centralized critics provide advantages in coordination tasks like Balance, and whether off-policy learning offers benefits over on-policy methods in the exploration-focused Sampling environment.

B. Environments

Two environments, Balance and Sampling, were used to test the mentioned algorithms.

1) Balance: The Balance environment is a task in the Vectorized Multi-Agent Simulator (VMAS) that spawns N agents at the bottom of a world with vertical gravity. A line is drawn above the agents, and a package is randomly placed on top of it. The agents' primary objective is to cooperatively transport the package to a specific goal located at the top of the environment with a green circle. [3] This is shown in Figure 1.

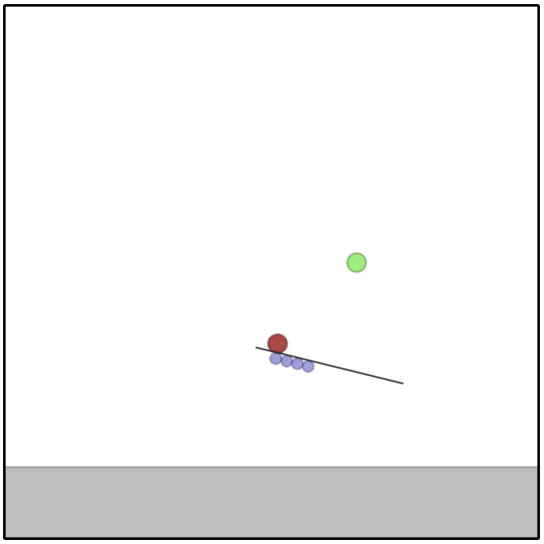

Fig. 1. The balance environment

When applied to the Balance task, the benchmark results from other studies show notable performance differences between various classes of algorithms. The best overall performance is achieved by actor-critic models that use centralized critics, such as MASAC, MADDPG, and MAPPO. The success of these models is attributed to the critic's ability to condition on global state information during the training phase, which simplifies the coordination required to move the package upward against vertical gravity. In contrast, Q-Learning algorithms such as IQL, VDN, and QMIX have been shown to perform poorly in this environment. [3]

2) Sampling: The Sampling environment is a task in the Vectorized Multi-Agent Simulator (VMAS) that spawns N agents randomly in a workspace with an underlying Gaussian density function. The workspace is discretized into a grid where each cell contains a sample value. The agents' primary objective is to collect samples by visiting grid cells, with each sample collected without replacement and given as reward to the whole team. Agents observe their position, velocity, nearby sample values in a 3×3 grid, and use lidar to sense each other. [3] This is shown in Figure 2.

When applied to the Sampling task, the benchmark results from other studies show contrasting performance characteristics compared to coordination-heavy environments. Independent learning approaches, particularly IPPO, demonstrate

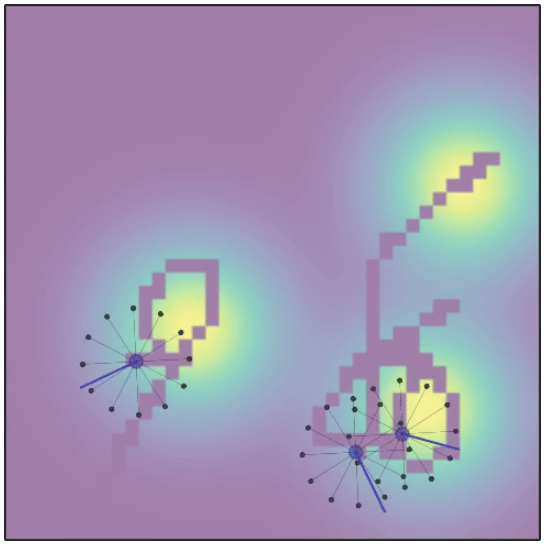

Fig. 2. The sampling environment

strong performance in this setting. The success of these methods stems from the task's inherent structure: agents can explore and collect samples in different regions of the workspace without requiring tight synchronization of their actions. The decentralized nature of sampling allows agents to contribute to the collective reward through parallel exploration rather than coordinated manipulation. In contrast, algorithms with centralized critics, such as MAPPO and MADDPG, do not provide significant advantages in this environment, as the added complexity of conditioning on global state information during training does not outweigh the benefits of independent exploration strategies. [3]

C. Experimental setup

For each run, we loaded `ExperimentConfig` instance from the default YAML configuration provided by BenchMARL and then adapted it to the needs of this study. The total training horizon was set to max_n_iters=100, with on_policy_collected_frames_per_batch=3000 environment steps per policy update. This results in 300,000 environment interactions overall per experiment, which, in reality, offered a good balance between computational expense and enabling the algorithms to achieve a plateau of stable performance.

Evaluation was activated and scheduled every 6000 environment steps, which provided sufficiently frequent checkpoints to monitor generalization without incurring excessive evaluation cost. Model checkpoints were saved every 12000 steps, enabling recovery and offline inspection of intermediate policies. To ensure a fair comparison across algorithms, the training configuration was kept as uniform as possible. For IPPO and MAPPO, it would have been straightforward to use relatively frequent evaluation and checkpointing (e.g., evaluation_interval=3000 and checkpoint_interval=6000). However, MADDPG relies on the off_policy_collected_frames_per_batch parameter, which by

TABLE I
COMPARISON OF MAPPO, IPPO, AND MADDPG ON THE VMAS BALANCE TASK WITH 3 AGENTS. ALL METRICS ARE AVERAGED OVER THE LAST 10 EVALUATIONS.

Algorithm	Final eval	Best eval	Mean eval	Mean episode length
IPPO	37.84	41.33	38.67	100.00
MADDPG	−4.02	29.59	13.37	81.22
MAPPO	37.04	44.06	40.79	100.00

default is set to 6000. In this off-policy setting, both evaluation_interval and checkpoint_interval are required to be integer multiples of off_policy_collected_frames_per_batch. Consequently, to keep the configuration consistent and comparable across all three algorithms, the intervals were set to 6000 for evaluation and 12000 for checkpointing for IPPO, MAPPO, and MADDPG alike.

We controlled the number of agents by overriding the n_agents field in the configuration. To investigate scalability with agent size, we ran experiments with $n \in \{3, 5, 7\}$ agents for each task. The set $\{3, 5, 7\}$ was chosen to (i) include a small agent size where coordination is relatively simple, (ii) a moderate regime where coordination becomes non-trivial, and (iii) a larger agent size that would need more communication. This progression allows observation of how algorithm performance and learning stability evolve as the joint action space and interaction complexity increase.

IV. RESULTS AND DISCUSSION

A. Balance performance

MAPPO and IPPO demonstrate strong performance on the Balance task, with MAPPO achieving the highest overall scores. In contrast, MADDPG shows significantly lower performance and training instability. We conclude this from table I, which summarizes the quantitative performance of all three algorithms on the Balance task with three agents. Both PPO-based methods achieve consistently high returns and effectively solve the task, as indicated by mean evaluation returns close to 40 and an average episode length of 100 steps, corresponding to the maximum episode horizon. MAPPO attains the highest best and mean evaluation return, suggesting slightly better asymptotic performance than IPPO under the same training. In contrast, MADDPG lags behind substantially. Although it occasionally reaches relatively high rewards (best return of 29.59), its final and mean returns remain much lower and its average episode length is noticeably shorter, indicating that episodes terminate earlier due to failures. These results already point to PPO-style on-policy methods, and MAPPO in particular, as more reliable and stable choices for this cooperative continuous-control setting.

Figure 3 reports the learning dynamics of MAPPO on the VMAS Balance task with three agents. Overall, MAPPO is able to solve the task reliably and with stable training behaviour. The mean training return increases rapidly from negative values to around 40 within the first ~ 25 iterations and then plateaus, indicating that the policy quickly discovers a successful balancing strategy and subsequently refines it. The evaluation curve follows a very similar trajectory, with a slightly higher mean return (around 32) over the full run, suggesting that the learned policy generalizes well to the evaluation rollouts and does not overfit to the particular trajectories seen during training.

The loss and entropy metrics provide further evidence of stable optimisation. The critic loss exhibits a sharp decrease during the early phase of learning and then fluctuates at a low level, which is consistent with the value function converging to a reasonable approximation of the true returns. The policy loss shows an initial descent followed by oscillations around zero, as expected for a policy-gradient method operating near a local optimum. At the same time, the policy entropy monotonically decreases over training, reflecting a gradual shift from exploratory behaviour towards more deterministic actions as the agents become confident about which behaviours are rewarded.

Finally, the episode length curve confirms that MAPPO learns to maintain balance for almost the entire episode horizon. After a brief initial phase with shorter episodes, the average episode length quickly saturates at the maximum value allowed by the environment and remains there for the remainder of training. Taken together, these curves indicate that MAPPO not only attains high returns on Balance, but does so in a sample-efficient and numerically stable manner, making it the strongest of the evaluated algorithms on this task.

B. Sampling performance

IPPO demonstrates superior performance on the Sampling task, achieving the highest overall scores. In contrast to the Balance task where MAPPO excelled, here IPPO clearly outperforms both MAPPO and MADDPG. We conclude this from Table II, which summarizes the quantitative performance of all three algorithms on the Sampling task with three agents. IPPO achieves consistently high returns and effectively solves the task, as indicated by a mean evaluation return above 21 and an average episode length of 100 steps, corresponding to the maximum episode horizon. IPPO attains the highest final and mean evaluation returns, suggesting strong asymptotic performance and stable learning. In contrast, MAPPO shows moderate performance, while MADDPG lags behind substantially. Although MADDPG occasionally reaches relatively high rewards (best return of 23.17), its final and mean returns remain lower, indicating high variance and training instability. Unlike Balance, all three algorithms maintain full episode lengths, suggesting that agents successfully navigate the exploration task without early termination. These results point to independent learning methods, and IPPO in particular, as the most reliable choice for this exploration-focused cooperative setting.

Figure 4 shows how IPPO learns on the VMAS Sampling task with three agents. The algorithm demonstrates stable and reliable learning throughout training. Training returns start low, but climb steadily to around 22 over approximately

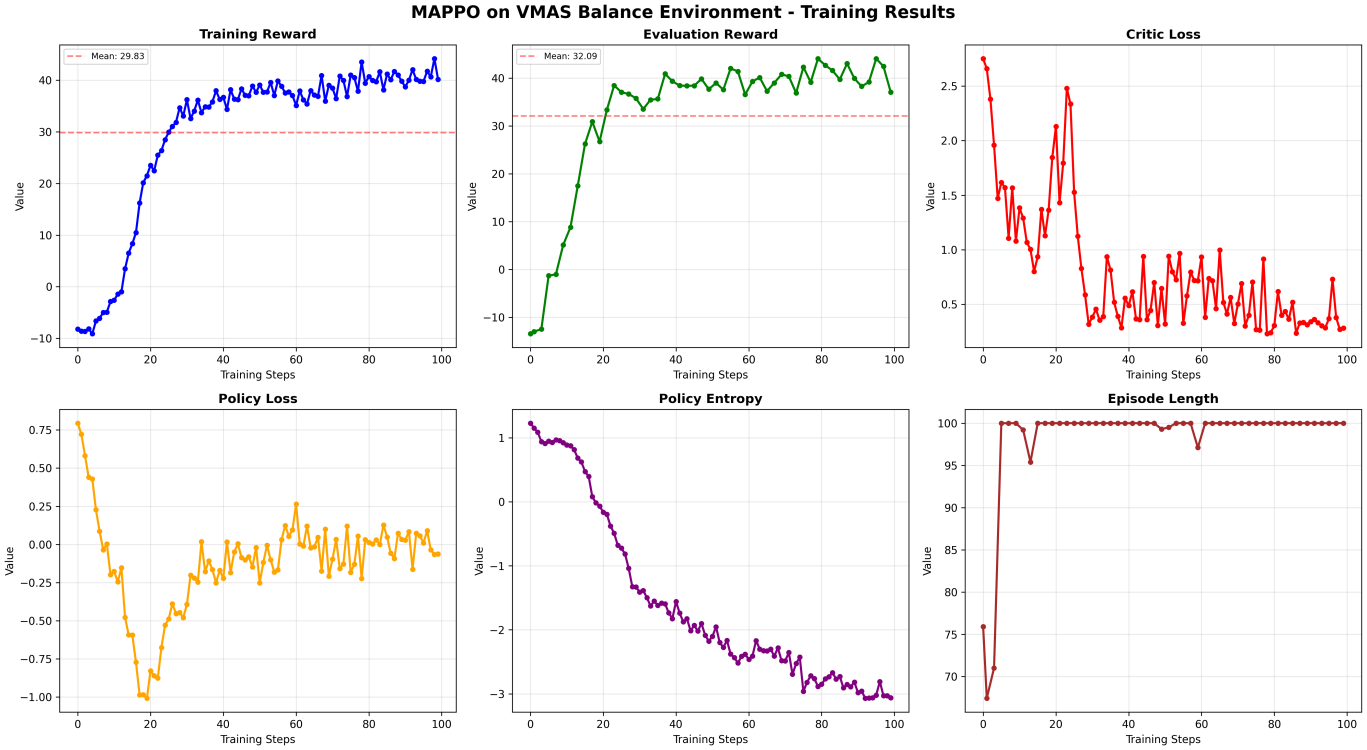

Fig. 3. Learning curves for MAPPO on the VMAS Balance environment (3 agents). The plots show the evolution of training and evaluation returns, critic and policy losses, policy entropy, and evaluation episode length over training iterations.

TABLE II
COMPARISON OF MAPPO, IPPO, AND MADDPG ON THE VMAS SAMPLING TASK WITH 3 AGENTS. ALL METRICS ARE AVERAGED OVER THE LAST 10 EVALUATIONS.

Algorithm	Final eval	Best eval	Mean eval	Mean episode length
IPPO	22.73	24.72	21.73	100.00
MADDPG	14.99	23.17	18.12	100.00
MAPPO	17.33	20.50	16.56	100.00

60 iterations, with continued gradual improvement afterward. This progression shows that agents are finding increasingly effective ways to explore the grid. Evaluation returns track this trend reasonably well, averaging around 16 but occasionally spiking above 20. The gap between training and evaluation suggests the policy maintains good exploration diversity rather than overfitting to specific patterns.

Looking at the loss and entropy curves reveals more about the learning process. The critic loss steadily rises and then levels off at moderate values - a pattern that makes sense given how the value function must adapt as agents discover new exploration strategies across the grid. Policy loss fluctuates around zero as the algorithm continuously tweaks the exploration behavior. The entropy curve, unlike the Balance task where entropy dropped quickly, here it decreases much more gradually and stays relatively high even late in training. This slower decline is actually helpful for Sampling, since agents benefit from maintaining some randomness to keep discovering uncollected samples in different grid regions.

Episode lengths remain at the maximum of 100 steps throughout training, showing that agents navigate the environment without issues from the very beginning. These learning curves demonstrate that IPPO achieves strong performance on Sampling while keeping exploration active and learning stable, which explains why it outperforms the other algorithms on this task

C. Scalability analysis

1) Balance with $n \in \{3, 5, 7\}$ agents: Having identified MAPPO as the strongest algorithm for the 3 agent Balance task, we extended our evaluation to team sizes of 5 and 7 agents to examine how performance scales with the number of cooperating agents. Table III summarizes this. Across all settings, the mean evaluation episode length over the last ten evaluations remains at the maximum value of 100 steps, indicating that MAPPO reliably learns policies that keep the system balanced for the full episode horizon regardless of team size. This suggests that increasing the number of agents does not make the task harder for MAPPO in terms of maintaining stability. Instead, the algorithm continues to produce solutions that avoid early termination.

However, the evaluation returns show a clear upward trend with the number of agents. For three agents, MAPPO achieves a final evaluation return of 37.04, with a best return of 44.06 and a mean return of 40.79 over the last ten evaluations. When the team size is increased to five agents, both the final

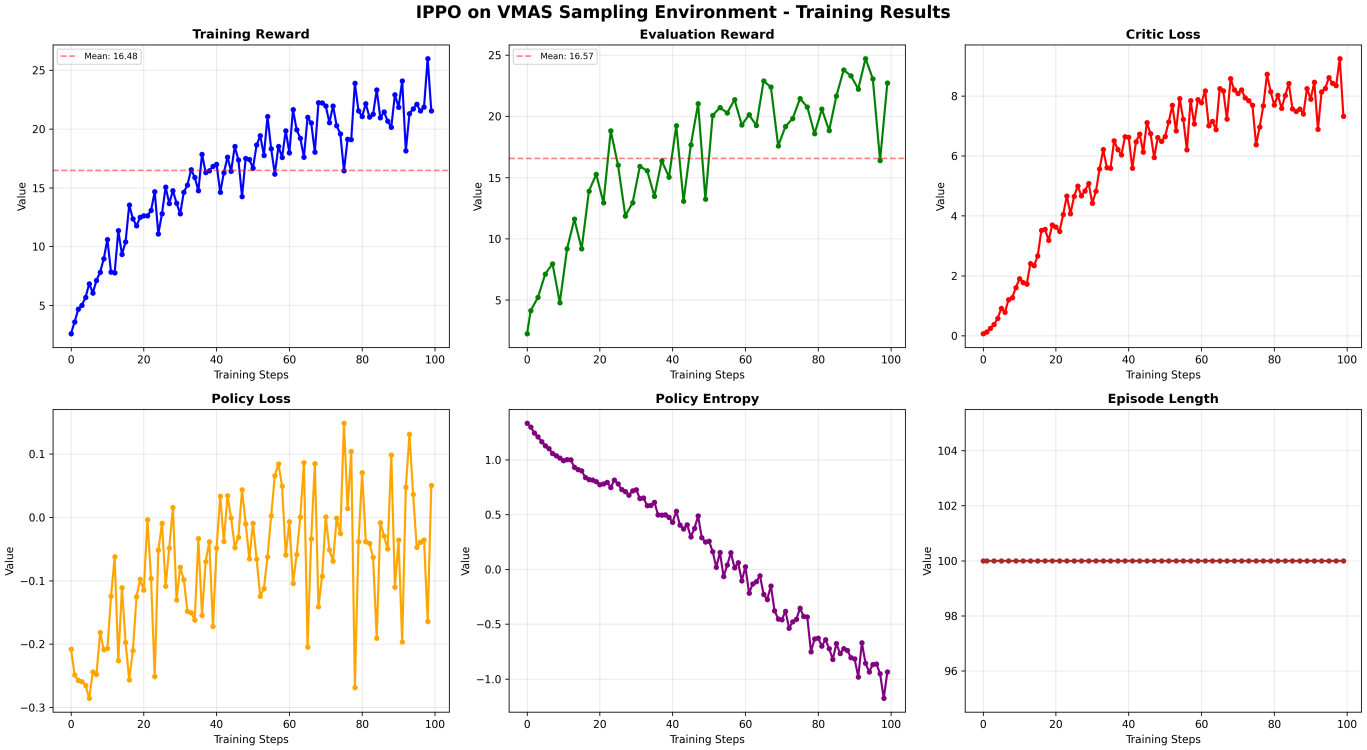

Fig. 4. Learning curves for IPPO on the VMAS Sampling environment (3 agents). The plots show the evolution of training and evaluation returns, critic and policy losses, policy entropy, and evaluation episode length over training iterations.

TABLE III
SCALABILITY OF MAPPO ON THE VMAS BALANCE TASK ACROSS DIFFERENT AGENT SIZES. ALL METRICS ARE AVERAGED OVER THE LAST 10 EVALUATIONS.

# Agents	Final eval	Best eval	Mean eval	Mean episode length
3	37.04	44.06	40.79	100.00
5	56.91	62.14	58.94	100.00
7	83.58	83.58	77.95	100.00

TABLE IV
SCALABILITY OF IPPO ON THE VMAS SAMPLING TASK FOR DIFFERENT TEAM SIZES. ALL METRICS ARE AVERAGED OVER THE LAST 10 EVALUATIONS.

# Agents	Final eval	Best eval	Mean eval	Mean episode length
3	22.73	24.72	21.73	100.00
5	20.06	23.44	20.92	100.00
7	19.83	20.71	18.29	100.00

and mean returns improve substantially (to 56.91 and 58.94, respectively), and the best observed return reaches 62.14. With seven agents, MAPPO attains its highest performance: the final return rises to 83.58 and the mean converged return to 77.95, while still maintaining full-length episodes. The fact that both the mean and best returns increase with team size indicates that MAPPO is able to effectively exploit the additional control capacity provided by more agents.

Overall, these results demonstrate favorable scalability of MAPPO on the Balance task. As the number of agents grows, the algorithm not only preserves stability (episodes of maximal length) but also achieves progressively higher returns, suggesting that the cooperative structure of Balance allows additional agents to contribute positively to performance when coordinated through a centralized critic and shared policy representation.

2) Sampling with $n \in \{3, 5, 7\}$ agents: Having established IPPO as the most effective algorithm for the 3-agent Sampling task, we conducted additional experiments with team sizes of 5 and 7 agents to investigate how performance scales as the number of exploring agents increases. Table IV presents these findings. Consistent with the Balance experiments, the mean evaluation episode length across all configurations remains at the maximum of 100 steps, confirming that IPPO consistently learns policies that enable agents to navigate the exploration task for the entire episode duration without encountering failures or early termination, regardless of team composition.

Unlike the Balance task, however, the evaluation returns reveal a contrasting pattern with respect to team size. With three agents, IPPO achieves a final evaluation return of 22.73, a best return of 24.72, and a mean return of 21.73 over the last ten evaluations. When scaling to five agents, performance exhibits a modest decline - the final return decreases to 20.06. This downward trend continues with seven agents, where IPPO records its lowest performance with a final return of 19.83. The progressive reduction in both mean and best returns as team

size increases suggests that adding more agents introduces coordination overhead that outweighs the potential benefits of parallel exploration in this environment.

This negative scaling behavior can be attributed to the nature of the Sampling task and the independent learning paradigm employed by IPPO. As more agents operate simultaneously in the shared workspace, they increasingly interfere with one another's exploration strategies - visiting already-sampled grid cells, clustering in high-density regions, or creating redundant coverage patterns. Since IPPO agents learn independently without explicit coordination mechanisms or shared critics, they cannot effectively partition the exploration space or coordinate their sampling strategies as the team grows. The result is diminishing returns per agent and lower overall collective performance, despite maintaining episode stability.

These findings highlight an important limitation of independent learning approaches in exploration tasks with shared resources. While IPPO excels at the Sampling task with small teams due to its simplicity and low coordination overhead, its lack of centralized information sharing becomes a liability as team size increases. This contrasts sharply with MAPPO's positive scaling on Balance, where the centralized critic enables effective coordination that grows more valuable with additional agents.

D. Algorithm Selection Guide and comparative analysis

Based on our evaluation across the Balance and Sampling environments, Table V summarizes practical guidelines for algorithm selection in cooperative multi-agent reinforcement learning tasks.

MAPPO is best suited for coordination-heavy tasks requiring tight synchronization between agents, such as cooperative manipulation or formation control, where the centralized critic's access to global state information enables effective coordination. IPPO performs optimally in exploration-focused tasks with parallel objectives and small to medium teams, where agents can contribute independently without requiring explicit coordination mechanisms. MADDPG, while designed for mixed cooperative-competitive settings and scenarios requiring off-policy learning, struggles with training stability and high variance in pure cooperative continuous control environments, making it less suitable for the tasks evaluated in this study.

Overall, algorithm performance is strongly task-dependent: MAPPO outperforms independent learners in coordination-heavy tasks like Balance, whereas IPPO achieves superior results in exploration-focused tasks such as Sampling due to lower coordination overhead. These findings highlight the importance of analyzing task structure, specifically the degree of required coordination, before selecting an algorithm.

PPO-based methods (MAPPO and IPPO) demonstrate clear practical advantages over MADDPG in cooperative settings, achieving stable training with minimal hyperparameter tuning while MADDPG exhibits high variance and poor convergence. The on-policy nature of PPO prevents destructive updates and maintains alignment between value estimates and current policies, avoiding the distribution mismatch issues that affect off-policy methods with experience replay in non-stationary multi-agent environments [10]. While MADDPG was designed for mixed cooperative-competitive scenarios, our results indicate that its benefits do not extend to pure cooperative continuous control tasks where on-policy methods provide superior stability.

TABLE V
ALGORITHM STRENGTHS AND RECOMMENDED USE CASES BASED ON TASK CHARACTERISTICS.

Algorithm	Strengths	Limitations
MAPPO	Centralized critic provides global state information; stable training; excellent scalability with agent count	Higher computational cost; may be unnecessary for independent exploration tasks
IPPO	Simple implementation; computationally efficient; strong performance when coordination is loose	Poor scalability with many agents; lacks explicit coordination mechanisms
MADDPG	Off-policy learning enables sample reuse; designed for heterogeneous agent interactions	High variance; difficult hyperparameter tuning; unstable training in pure cooperative settings

The contrasting scalability results between MAPPO on Balance and IPPO on Sampling reveal the context-dependent value of centralized critics. MAPPO's performance nearly doubled as agents increased from 3 to 7 on Balance, showing effective coordination through global state information. IPPO's performance degraded by 15% on Sampling as team size grew, since independent learners lack mechanisms to prevent redundant exploration. These results suggest that centralized critics are essential for tasks requiring tight coordination but may introduce unnecessary complexity for naturally parallelizable objectives.

E. Lessons learned

Through this comparative study, we gained several practical insights into multi-agent reinforcement learning algorithm selection and training.

1) Algorithm Selection is Task-Dependent

Choosing the right algorithm significantly impacts performance and training efficiency. For coordination-heavy tasks like Balance, the MAPPO's centralized critic provides meaningful advantages. However, for exploration tasks such as Sampling, the added complexity does not provide no benefit, and the simpler IPPO performs better.

2) PPO-based Methods Offer Better Practical Trade-offs

Our experiments confirm that PPO-based algorithms (MAPPO and IPPO) are significantly more robust than MADDPG for these cooperative tasks. They achieve better performance with minimal hyperparameter tuning. The on-policy nature handles multi-agent non-stationary more effectively than off-policy replay buffers.

3) Training Stability Requires Attention

We encountered several practical challenges: gradient explosion (requiring reduced learning rates), high variance in MADDPG (requiring multiple seeds), and the importance of entropy regularization for exploration tasks. Comparison against random baselines proved essential for interpreting negative rewards correctly.

F. Future work and Limitations

MADDPG Hyperparameter Sensitivity: Our experiments revealed that MADDPG performs poorly, particularly in the Balance environment compared to PPO-based methods. This observation aligns with other findings that off-policy multi-agent algorithms present considerable challenges in hyperparameter optimization. [11]

Furthermore, investigations into variational inequality methods for multi-agent reinforcement learning have highlighted that MADDPG training exhibits high sensitivity to random seed selection and hyperparameter choices, making reliable comparisons between methods challenging [12]. Future work should explore automated hyperparameter optimization techniques or other approaches specifically tailored to multi-agent off-policy algorithms to reduce the manual tuning burden and improve reproducibility.

Transfer Learning and Zero-Shot Generalization: The absence of evaluation on transfer learning scenarios where agents trained in one environment must adapt to new but related tasks without retraining provides ground for future work.

Future work should systematically investigate transfer learning capabilities across different algorithms by training agents in source environments and evaluating zero-shot or few-shot adaptation to target domains with modified team sizes, altered observation spaces, or different reward structures.

Limited Hyperparameter Exploration: A significant constraint of this study stems from limited computational resources, which restricted our ability to conduct extensive hyperparameter optimization across the evaluated algorithms.

Future work with access to greater computational capacity should systematically explore hyperparameter spaces for different algorithms using automated optimization frameworks. This may lead to potentially revealing configurations that substantially improve the performance of algorithms that appeared less effective in our study.

V. Conclusion

This study establishes empirical foundations for task-dependent algorithm selection in cooperative multi-agent reinforcement learning through systematic evaluation of IPPO, MAPPO, and MADDPG across coordination-heavy and exploration-focused environments. Our experiments demonstrate that algorithmic design choices, particularly centralized versus independent critics and on-policy versus off-policy learning, result in fundamentally different performance profiles depending on task structure.

The Balance environment revealed MAPPO's superiority in coordination-intensive scenarios, with performance improvements of 113% when scaling from three to seven agents. This success is directly due to to the centralized critic's capacity to leverage global state information for synchronized control. In contrast, the Sampling environment exposed the inefficiencies of centralized training for parallel exploration tasks. IPPO's independent learning paradigm achieved 30% higher returns than MAPPO while maintaining computational efficiency. These findings disprove the notion that centralized training universally benefits cooperative multi-agent systems. Instead, they highlight the critical importance of matching algorithmic mechanisms to coordination requirements.

When combined, our findings highlight the fact that greater performance in cooperative multi-agent systems is not always ensured by algorithmic complexity alone. This work highlights the necessity of informed, context-aware algorithm selection when transferring MARL approaches from benchmark environments to practical cooperative applications by illustrating distinct task-dependent trade-offs.

References

[1] G. Papadopoulos, A. Kontogiannis, F. Papadopoulou, C. Poulianou, I. Koumentis, and G. Vouros, "An extended benchmarking of multi-agent reinforcement learning algorithms in complex fully cooperative tasks," *arXiv preprint arXiv:2502.04773*, 2025.

[2] M. Bettini, A. Prorok, and V. Moens, "Benchmarl: Benchmarking multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 25, no. 217, pp. 1–10, 2024. [Online]. Available: http://jmlr.org/papers/v25/23-1612.html

[3] M. Bettini, R. Kortvelesy, J. Blumenkamp, and A. Prorok, "Vmas: A vectorized multi-agent simulator for collective robot learning," *The 16th International Symposium on Distributed Autonomous Robotic Systems*, 2022.

[4] Z. Ning and L. Xie, "A survey on multi-agent reinforcement learning and its application," *Journal of Automation and Intelligence*, vol. 3, no. 2, pp. 73–91, 2024.

[5] O. Mahjoub, S. Abramowitz, R. J. de Kock, W. Khlifi, S. V. Du Toit, J. Daniel, L. B. Nessir, J. C. Formanek, L. Beyers, L. Clark *et al.*, "Performant, memory efficient and scalable multi-agent reinforcement learning," 2024.

[6] A. Wong, T. Bäck, A. V. Kononova, and A. Plaat, "Deep multiagent reinforcement learning: Challenges and directions," *Artificial Intelligence Review*, vol. 56, no. 6, pp. 5023–5056, 2023.

[7] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *CoRR*, vol. abs/1706.02275, 2017. [Online]. Available: http://arxiv.org/abs/1706.02275

[8] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu, "The surprising effectiveness of MAPPO in cooperative, multi-agent games," *CoRR*, vol. abs/2103.01955, 2021. [Online]. Available: https://arxiv.org/abs/2103.01955

[9] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *CoRR*, vol. abs/2011.09533, 2020. [Online]. Available: https://arxiv.org/abs/2011.09533

[10] J. Foerster, N. Nardelli, G. Farquhar, T. Afouras, P. H. Torr, P. Kohli, and S. Whiteson, "Stabilising experience replay for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1146–1155.

[11] G. Papoudakis, F. Christianos, L. Schäfer, and S. V. Albrecht, "Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks," *arXiv preprint arXiv:2006.07869*, 2020.

[12] B. A. Sidahmed and T. Chavdarova, "Variational inequality methods for multi-agent reinforcement learning: Performance and stability gains," 2024.