Remembering the Markov Property in Cooperative MARL

Kale-ab Abebe Tessera[1], **Leonard Hinckeldey**[1], **Riccardo Zamboni**[2], **David Abel**[1], **Amos Storkey**[1]
{k.tessera,l.hinckeldey,david.abel,a.storkey}@ed.ac.uk,
riccardo.zamboni@polimi.it

[1]**University of Edinburgh, Edinburgh, UK**
[2]**Politecnico di Milano, Milano, Italy**

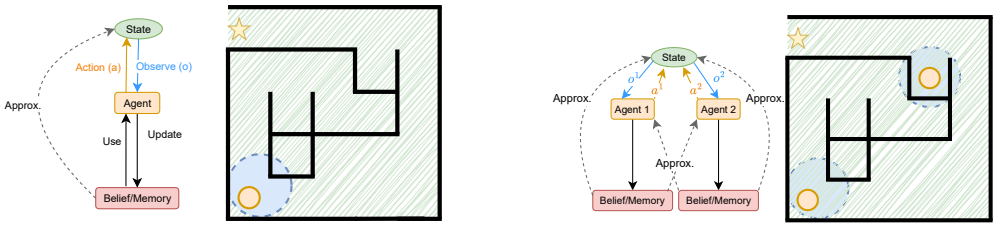
Abstract

Cooperative multi-agent reinforcement learning (MARL) is typically formalised as a Decentralised Partially Observable Markov Decision Process (Dec-POMDP), where agents must reason about the environment and other agents' behaviour. In practice, current model-free MARL algorithms use simple recurrent function approximators to address the challenge of reasoning about others using partial information. In this position paper, we argue that the empirical success of these methods is not due to effective Markov signal recovery, but rather to learning simple conventions that bypass environment observations and memory. Through a targeted case study, we show that co-adapting agents can learn brittle conventions, which then fail when partnered with non-adaptive agents. Crucially, the same models can learn grounded policies when the task design necessitates it, revealing that the issue is not a fundamental limitation of the learning models but a failure of the benchmark design. Our analysis also suggests that modern MARL environments may not adequately test the core assumptions of Dec-POMDPs. We therefore advocate for new cooperative environments built upon two core principles: (1) behaviours grounded in observations and (2) memory-based reasoning about other agents, ensuring success requires genuine skill rather than fragile, co-adapted agreements.

1 Introduction

In many real-world scenarios, teams of agents must make decisions and cooperate under uncertainty without accessing the full conditions of the environment they act upon. This is remarkably the central challenge of multi-agent learning: contending with imperfect information. Decentralised Partially Observable Markov decision processes (Dec-POMDPs, Bernstein et al., 2002; Oliehoek et al., 2016) are the prominent decision-making model in these scenarios, as each agent perceives only their own actions and observations.

As such, they are the more common generalisation of single-agent Partially Observable Markov Decision Processes (POMDP, Åström, 1965; Kaelbling et al., 1998). Yet, despite their ubiquity in practice, our understanding of MARL (Albrecht et al., 2024) in such settings remains limited. This is somewhat expected since even in single-agent settings, planning and learning under partial observability suffer from well-known computational and statistical hardness results (Papadimitriou & Tsitsiklis, 1987; Lusena et al., 2001), and the presence of multiple agents hinders the ability to build a full history of the performed actions and perceived observations, or to recover a distribution over the latent state of the environment, known as *belief* (Kaelbling et al., 1998). In principle, each



(a) **POMDP:** *If you can't see, you must remember (Kaelbling et al., 1998).*

(b) **Dec-POMDP:** *If you can't see, you must predict (Oliehoek et al., 2016).*

Figure 1: **Hidden State Requirements in (Dec-)POMDPs. (a)** In POMDPs, the agent uses memory or beliefs to approximate the state. **(b)** In Dec-POMDPs, each agent must additionally predict the behaviour of other agents under uncertainty.

agent should build and update a belief over the joint state and other agents' policies (or equivalently individual histories) – "multi-agent belief" – to recover a **Markovian signal** (Oliehoek et al., 2016). As visually represented in Fig. 1, agents should approximate the environment state and be able to predict the behaviour of other agents to act optimally.

However, exact multi-agent belief-state computation in Dec-POMDPs is known to be NEXP-complete (Bernstein et al., 2002). As a result, practical model-free MARL methods rely on finite-memory or recurrent policy representations (e.g., RNNs or GRUs), originally used to handle partial observability in single-agent RL (Hausknecht & Stone, 2015). These are usually instantiated in the *centralised training with decentralised execution* (CTDE) (Oliehoek et al., 2008; Kraemer & Banerjee, 2016) paradigm, where agents have access to additional information (sometimes from other agents) during training, to improve efficiency.

While the practical success of model-free MARL methods (among others, Yu et al., 2022; Papoudakis et al., 2020) might suggest that these methods are adequately recovering the Markov signal by reasoning about the environment and other agents. In this paper, we argue to the contrary. Through a focused case study (Section 4), we demonstrate that co-adapting agents often sidestep true state recovery by converging on brittle conventions that depend neither on grounded observations nor on the recurrent hidden state. Yet, the same architectures can learn state-grounded policies once the task is not amenable to conventions, hinting that the shortfall lies not in the learning or modelling capacity of the models but in the environment design.

Additionally, we highlight this misalignment in modern benchmarks such as Hanabi (Bard et al., 2020), MaBrax (Rutherford et al., 2023; Peng et al., 2021) and SMAX (Rutherford et al., 2023), where either memoryless feed-forward policies can paradoxically outperform or match their recurrent counterparts or the learned policies do not adequately require dependence on observations or history (Section 5). Such results imply that many MARL tasks do not in fact demand the kind of temporal reasoning and belief maintenance that Dec-POMDP theory considers essential.

Collectively, these findings expose a gap between the reasoning abilities agents possess and the behaviours that current environments test for. Whereas prior work has proposed algorithms to mitigate conventions (Hu et al., 2020; Foerster et al., 2019; Hu et al., 2021), we reframe the emergence of conventions as a diagnostic signal: when a benchmark allows effortless coordination through non-generalisable shortcuts, it is the benchmark, rather than the algorithm, that requires re-evaluation. We therefore advocate for new cooperative environments built upon two core principles: **(1) behaviours grounded in observations** and **(2) memory-based reasoning** about other agents, ensuring agents must recover and exploit the true Markov structure of MARL problems to succeed.

2 Related Work

Conventions. Prior work has shown that co-trained MARL agents can form conventions that can be brittle to new unseen partners and propose augmenting learning algorithms to tackle this (Hu et al., 2020; Foerster et al., 2019; Hu et al., 2021). In contrast, we reframe conventions as a diagnostic signal that the benchmark itself might be ill-posed and fail to test for the intended Dec-POMDP reasoning. Using mutual information metrics, we show that agents can learn to ignore their observations entirely, yet these same methods are capable of learning reliable, grounded policies when the task design necessitates it. Furthermore, although we show some cases of zero-shot coordination failures in Section 4, our primary focus (Section 5) is the standard MARL setting where agents are trained and evaluated together.

Environments. SMAC V2 (Ellis et al., 2023) showed that many original SMAC (Samvelyan et al., 2019) maps could be solved by open-loop policies that ignored local observations (only conditioned on time steps) and introduced "meaningful partial observability" to mitigate this flaw. We show that other modern MARL environments such as MaBrax (Rutherford et al., 2023) suffer from an even stronger variant of the same pathology: "blind" agents that receive *no* observations (even without time steps) still obtain non-trivial returns on a range of configurations (Fig. 8 in the Appendix). We extend this line of inquiry by analysing a broader range of environments and looking beyond just partial observability to evaluate the need for grounded, memory-based policies.

Agent Modelling. Agent modelling techniques aim to predict other agents' actions, goals, or beliefs to improve coordination or competition (Albrecht & Stone, 2018). Modern methods focus on learning latent representations from observation histories, often using auto-encoders with auxiliary prediction losses, to modelling the behaviour of other agents (Papoudakis et al., 2021; Zintgraf et al., 2021; Rabinowitz et al., 2018; Xie et al., 2021). While such architectures are powerful, model-free recurrent policies remain the dominant baseline in cooperative MARL (Yu et al., 2022; Papoudakis et al., 2020). Recent partner modelling work showed that model-free RNNs can encode teammates' abilities when the environment enables influence over them, in two-player Overcooked environments (Mon-Williams et al., 2025). They focused on the ad-hoc teamwork settings with a single controllable agents. Our investigation complements their results as we show memory-based reasoning and grounded policies can emerge when the environment requires it; however, we focus on MARL settings where all agents are co-trained together.

3 Background

We next introduce the main concepts that will be covered throughout the paper.

Interaction Protocol. As a base model for interaction, we consider a discounted Dec-POMDP (Bernstein et al., 2002), defined by the tuple $\mathcal{M} = (\mathcal{N}, \mathcal{S}, \mathbb{T}, \mathbb{O}, \mu, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\mathcal{O}^i\}_{i\in\mathcal{N}}, R, \gamma)$. Here, $\mathcal{N}$ is the set of $N \in \mathbb{N}$ agents and $\mathcal{S}$ is the set of global states. At each time step t, the system is in some state $s_t \in \mathcal{S}$. Each agent $i \in \mathcal{N}$ selects an action $a_t^i \in \mathcal{A}^i$, forming a joint action $\mathbf{a}_t = (a_t^1, \dots, a_t^N)$ in the joint action space $\mathcal{A} = \prod_{i=1}^N \mathcal{A}^i$. This action leads to a state transition according to the probability function $\mathbb{T}(s_{t+1}|s_t, \mathbf{a}_t)$ and a shared reward $R(s_t, \mathbf{a}_t)$. Agents do not observe the global state s_t, instead they receive a local observation $o_t^i \in \mathcal{O}^i$. The joint observation $\mathbf{o}_t$ is drawn according to the observation function $\mathbb{O}(\mathbf{o}_t|s_t, \mathbf{a}_{t-1})$. The goal is to learn a joint policy $\boldsymbol{\pi}$ that maximises the expected discounted return, given an initial state distribution $\mu \in \Delta(\mathcal{S})$ and a discount factor $\gamma \in [0, 1)$: $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}} \mathbb{E}_{s_0\sim\mu,\, \mathbf{a}_t\sim\boldsymbol{\pi}} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, \mathbf{a}_t)\right]$.

Markov Property. A Dec-POMDP has the Markov property if the current state contains all relevant information for predicting the future (Sutton & Barto, 1998; Oliehoek et al., 2016). Formally, this means the transition dynamics do not depend on the full history, $\mathbb{T}(s_{t+1} \mid s_t, \mathbf{a}_t) = \mathbb{T}(s_{t+1} \mid$

$s_t, \mathbf{a}_t, s_{t-1}, \mathbf{a}_{t-1}, \ldots, s_0, \mathbf{a}_0)$, where s_t is the environment state and $\mathbf{a}_t$ is the joint action. While the dynamics are Markovian in the joint state, individual agents cannot observe this state directly and must act based on partial observations, typically attempting to recover a Markovian signal by forming beliefs or approximations over the state and other agents' actions (Oliehoek et al., 2016).

Mutual Information. To study the information embedded in our agent's policies, we propose metrics based on mutual information (MI) $\mathbb{I}(X;Y)$, measuring the information shared between two discrete random variables X and Y, defined as $\mathbb{I}(X;Y) = \sum_{X,Y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}$. We measure the MI between an agent's observation and action, $\mathbb{I}(O;A)$, and its recurrent hidden state and action, $\mathbb{I}(H;A)$, to quantify dependence on sensory input and memory, respectively. We estimate these using the k-NN estimator (Kraskov et al., 2004; Ross, 2014; Garcin et al., 2025), while averaging all values across agents.

4 On the Recovery of Markovian Information in MARL

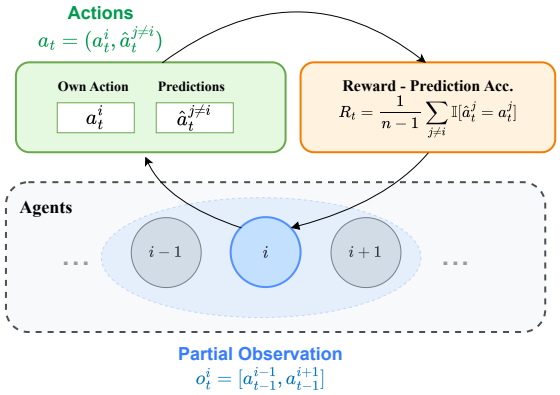
A plethora of recent works (among others, Yu et al., 2022; Papoudakis et al., 2020) provided extensive evidence of the empirical effectiveness of MARL in addressing a wide range of tasks. Such a success story might implicitly suggest that these algorithms do indeed manage to recover essential information for decision-making in multi-agent environments, namely a Markov signal reconstructing the environment's state and other agent's actions. In this section, we address the following fundamental question:

Are deep MARL policies truly recovering a Markov state? If no, what allows for their success?

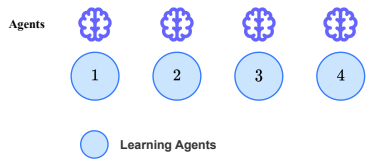
In the following, we will control for the information agents have access to so as to analyse the interplay between their ability to recover or extract essential features of the environment and the other agents as well as their ability to solve the tasks. Wijmans et al. (2023); Mon-Williams et al. (2025) provided extensive evidence that purely egocentric information, i.e., changes in one's location or orientation, can still give rise to emergent goal-navigation and implicit partner-modelling. We address push this idea further: We construct *blind* environment instantiations, in which agents have access to *no* observation at all. In other words, these blind agents receive no sensory input about the environment, their teammates or even their past actions, and thus they have to rely solely on reward feedback to learn. Concurrently, we will control for their ability to recover a Markovian signal by analysing two different instantiations of Independent PPO (IPPO, De Witt et al., 2020): **MLP (feed-forward)**, a standard multilayer perceptron that processes each observation independently, and **GRU (recurrent)**, a gated recurrent unit encoder capable of integrating information over time to form an implicit history representation. All methods do not use parameter sharing.

Warm-up Environment: Prediction Game. As a first experiment instantiation, we design the **Prediction Game** (see Fig. 2a): A cooperative task involving agents that act in an environment where at each timestep t, an agent i receives a partial observation o_t^i consisting of the previous actions of its two immediate neighbours, namely a_{t-1}^{i-1} and a_{t-1}^{i+1}. Esch agent then selects a multi-discrete action $a_t^i = (\tilde{a}_t^i, \hat{a}_t^{-i})$, made of two components: own action ($\tilde{a}_t^i$) being the agent's actual action; action prediction ($\hat{a}_t^{-i}$) being a vector of predictions for the actions of the set of other agents $-i = \{j \in \mathcal{N} \neq i\}$. Finally, agents share the same reward R_t explicitly defined as its prediction accuracy at timestep t, in other words $R_t = \frac{1}{N-1} \sum_{i \in \mathcal{N}} \mathbb{I}[\hat{a}_t^{-i} = \tilde{a}_t^{-i}]$, where $\mathbb{I}$ is the indicator function. By means of this reward function, Prediction Game ensures that agents will have to accurately model and predict the behaviour of others in order to succeed. In this way, the problem explicitly models Dec-POMDPs instances where recovering the Markovian signal requires agents to not only estimate the environment's state but also to predict the behaviour of others.

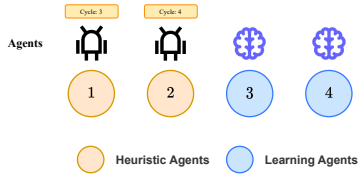
Emergence of Conventions in Concurrently Learning Agents. First, we addressed homogeneous settings where four IPPO agents learn concurrently (see Fig. 2b). As shown in Fig. 3a,



(a) The general agent interaction model, showing the observation-action-reward loop for a single agent within its local neighborhood.

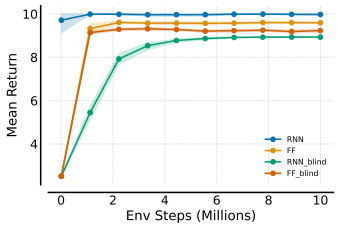


(b) A homogeneous setup with four learning agents.

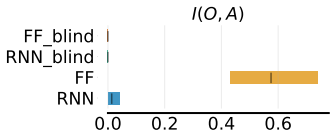


(c) A heterogeneous setup with two learning and two heuristic agents.

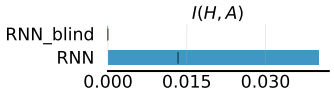
Figure 2: Overview of Predictive Game environment and the specific agent configurations used in the experiments.



(a) Mean Return, 95% CI.



(b) $I(O, A)$



(c) $I(H, A)$

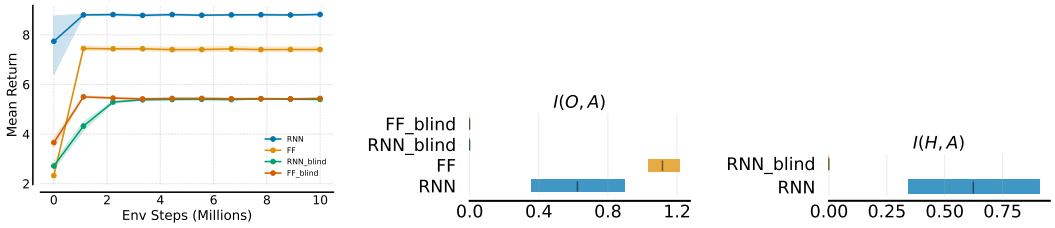
Figure 3: Concurrent Learning Agents Experiment: (a) Learning performances ; (b) Mutual Information between actions and observations; (c) Mutual Information between agents' actions and histories. We report the mean and bootstrapped 95% Confidence Intervals (CI) over 10 seeds.

learning both recurrent (**RNN**) and feed-forward (**FF**) policies lead to near-optimal performances. Strikingly, the *blind* agents, i.e., the ones lacking both memory and observations, also converge to high-performing policies. We claim that this surprising success does not stem from sophisticated, grounded reasoning, but rather agents learn to ignore their observations and form simple, synchronised policies. Indeed, Figures 3b, 3c show that the MI between observations and actions $I(O, A)$ and hidden state of the recurrent network H and actions $I(H, A)$ is low, as the maximum MI possible for these experiments is 8. In other words, the agents are not learning to rely on their grounded observations or hidden state, but rather simple, emergent conventions that bypass the need for complex reasoning (we refer to Figure 6 in Appendix for action distributions plots).

Failure of Conventions in Concurrently Learning Agents. To test the robustness of the conventions emerging in the previous case, we test resulting policies by substituting two of the agents with fixed-policy agents (we report details on such fixed policies in Appendix). Indeed, this zero-shot coordination task results in breaking the learned conventions and requires agents to infer the partners' behaviour from observation or history alone: The results in Table 1 show a drastic drop in performance for all agent types. This demonstrates that the learned policies are indeed brittle and fail to generalise. The conventions learned during concurrent learning are a shortcut that completely bypasses the challenge of robustly modelling other agents, failing immediately when partners behave unexpectedly.

Table 1: Performance Comparison: Baseline vs Adding Heuristic Agents

Scenario	RNN	FF	RNN_blind	FF_blind
Baseline	9.97 (9.92, 10.03)	9.59 (9.55, 9.63)	8.94 (8.91, 8.97)	9.23 (9.12, 9.34)
Add Heuristic	4.80 (4.41, 5.20)	4.04 (3.26, 4.81)	4.57 (4.18, 4.95)	4.62 (4.25, 4.99)



(a) Mean Return, 95% CI. (b) $I(O, A)$ (c) $I(H, A)$

Figure 4: Partially Concurrent Learning Agents Experiment: (a) Learning performances; (b) Mutual Information between actions and observations; (c) Mutual Information between agents' actions and histories. We report the mean and bootstrapped 95% Confidence Intervals (CI) over 10 seeds.

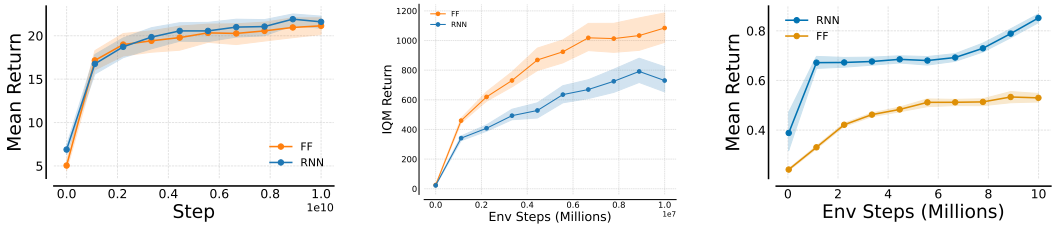
On the Necessity of Grounded Policies. Finally, we investigate whether agents can learn robust policies when the task requires it. To do so, we train agents from scratch in the environment with two fixed partners, deterministically selecting different actions in a repetitive way by following specific cycles (see Appendix A.1 for details). Since these agents start each episode in a random phase of their cycle, a learning agent cannot rely on a pre-arranged convention and is forced to infer the hidden state (the current phase and phase length) of its partners from observations. Interestingly, RNN policies do learn effective policies, demonstrating a successful use of memorization to identify the cycles and predict future actions In contrast, memoryless MLP policies perform worse, and blind agents fail to model the non-learning agents (Fig. 4a). Importantly, in both cases the MI between observations and actions (Fig. 4b) and the hidden state and actions (Fig. 4c) is now significantly higher. This indicates that, unlike in the previous setting, the agent's policy is now actively and necessarily grounded in its observation history: the mechanism for success has changed.

5 Do Modern Environments Require Complex Reasoning Relying on Markovian Information?

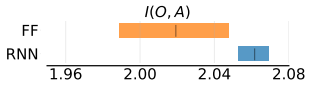
Section 4 revealed a crucial principle: When an environment is structured to prevent convention-based shortcuts, agents are capable of learning the desired, more complex behaviours of extracting Markovian signals from the environment. This naturally leads to a critical question for the broader MARL community:

*Do modern MARL environments **actually** require (1) behaviours grounded in observations and (2) memory-based reasoning about other agents?*

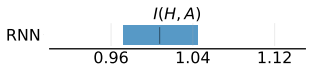
The answer to this question is paramount: Demanding behaviour that is grounded in observation is necessary to prevent agents from learning brittle conventions, while requiring memory-based reasoning ensures a task truly embodies the challenges of a Dec-POMDP. Without environments that enforce both, we risk measuring progress on benchmarks that can be solved with the same non-generalisable shortcuts identified in our case study. In this section, we investigate a small sample of popular MARL benchmarks to explore this question.



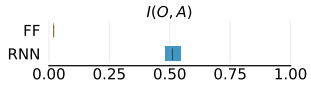
(a) Hanabi (Two-Players). (b) MaBrax. (c) SMAX (5 Units).



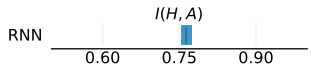
(d) Hanabi ($\mathbb{I}(O;A)_{max} \approx 3$)



(e) Hanabi ($\mathbb{I}(H;A)_{max} \approx 3$)



(f) SMAX ($\mathbb{I}(O;A)_{max} \approx 2.30$)



(g) SMAX ($\mathbb{I}(H;A)_{max} \approx 2.30$)

Figure 5: (a–c) Sample-efficiency (interquartile-mean) across Hanabi, MaBrax, and SMAX. (d–g) Mutual information between observations and actions $\mathbb{I}(O;A)$ and between hidden state and actions $\mathbb{I}(H;A)$, stacked per environment for Hanabi (d–e) and SMAX (f–g).

Agent Modeling: Hanabi. As a first environment instance, we consider Hanabi (Bard et al., 2020)[1]. It is a partially observable cooperative MARL environment based on the card game, where players see their teammates' cards but not their own. To succeed, players need to exchange clues and use them to infer information about the cards in their possession. The core challenge moves beyond a clue's literal meaning to inferring the teammate's intent—that is, why that specific clue was given. This has made Hanabi a popular benchmark when testing theory of mind, agents' modelling or ad-hoc coordination (Hu et al., 2020; Foerster et al., 2019; Nekoei et al., 2023; Hu et al., 2021).

Since an agent's only source of information comes from the clues provided by its partner, this design encourages policies to be grounded in observation. Our findings confirm this: the MI analysis shows that both RNN and FF policies learn to actively use their observations (Fig. 5d) or the RNN hidden state (Fig. 5e). However, our results also show that FF achieves nearly identical performance to the RNN, suggesting that memory provides no significant advantage in this specific task instantiation (see Fig. 5a). This implies that while Hanabi successfully necessitates grounded policies, it does not adequately test for complex, memory-based reasoning about other agents over time. The task can be solved with no notion of history, revealing a limitation in its ability to evaluate the full spectrum of reasoning required in complex Dec-POMDPs.[2]

Continuous Control: MABrax. To explore our central questions in more complex, continuous control MARL, we next evaluate our policies on five difference instances of Multi-Agent Brax (MABrax, Rutherford et al., 2023). MABrax is a JAX-accelerated version of the MaMuJoCo (Peng et al., 2021), where a robot's body parts are controlled by different agents. Each agent receives only ego-centric observations, such as its own joint angles and velocities, along with those of its immediate neighbours, with the goal being to collaborate to move forward.

The results reported in Figure 5b present a potentially surprising result:[3] memoryless feed-forward architectures outperform their RNN counterparts. We posit that this is because in locomotion-style tasks, current proprioception fully determines the optimal torque, making memory less relevant for these tasks. Furthermore, limited observability of distant agents also reduces any incentive for com-

[1] In the following, we use the two-player version of this game, where the maximum score is 25.

[2] As a side note, we highlight that throughout the experiments agents' observations included the entire discard pile and not just the top card. Different observation functions might call for the need for memorization.

[3] The interested reader can refer to Figure 7 in Appendix for the detailed plots.

plex partner modelling. As a side note, we highlight that while MaMuJoCo-inspired environments are currently ubiquitous as multi-agent benchmarking, they hide a crucial limitation: agents completely unaware of others can *still* achieve non-trivial performances in a set of instances. We reported these results in Fig. 8 in the Appendix, unlike previous results by Ellis et al. (2023), we also removed remove time steps from agents' observations.

Overall, these results provide strong evidence that popular locomotion benchmarks may not adequately evaluate challenging multi-agent learning. We also caution against using the fully-observable variants of these environments found in some recent work (Wang et al., 2023; Zhong et al., 2024), as they deviate from the Dec-POMDP problem structure.

Meaningful Partial Observability: SMAX. As a final experiment, we evaluate IPPO agents on SMAX (Rutherford et al., 2023), a JAX-accelerated version of the SMAC (Samvelyan et al., 2019). Specifically, we focus on SMAC-v2 (Ellis et al., 2023), which was introduced to address limitations in the original SMAC, such as a simple open-loop policies could succeed while ignoring observations. In contrast, SMAC-v2 incorporates stochastic starting positions and enforcing "meaningful partial observability", where agents must infer critical information held by their teammates.

Our results confirm that these changes successfully create a memory-dependent task. As shown in Figure 5c, RNN policies outperform their FF counterparts by a large margin, a finding also noted by Ellis et al. (2023). However, a deeper analysis reveals a more nuanced picture. When we look at $\mathbb{I}(O; A)$ and $\mathbb{I}(H; A)$, we see a moderate dependence between these variables and actions, which correspond to approximately 22% and 33% of the theoretical maximum $H(A) = \ln 10 \approx 2.30$ for a 10-action space, which is significantly lower than in coordination-centric benchmarks like Hanabi (69% and 32% of their maximum). This suggests that while SMAC-v2 maps are a significant improvement, there is still potential to design environments that demand an even stronger reliance on history-based reasoning to fully capture the complexity of Dec-POMDPs.

6 Conclusions and Takeaway

In this paper, we provide empirical evidence that suggests robust and generalisable MARL systems are fundamentally gated by the design of evaluation environments: These environments must be designed to necessitate and reward policies grounded in the known hardness of Dec-POMDPs (Fig. 1). By removing the possibility of convention-based shortcuts, we showed in Section 4 that agents can be encouraged to develop more meaningful and grounded policies. Additionally, we showed in Section 5, that many common benchmarks do not require temporal reasoning or grounded policies.

Takeaway. We advocate for the design and adoption of benchmarks that compel agents to develop policies built upon two core principles: (1) **behaviours grounded in observations** and (2) **memory-based reasoning** about other agents. We claim that enforcing these properties is essential to ensure that environments capture the true complexities of Dec-POMDPs and drive the development of more reliable multi-agent systems. While modern benchmarks show promise, our analysis indicates they too can be improved to more rigorously test these foundational capabilities.

Next-steps. We acknowledge several avenues for future research. Our analysis focused on Independent PPO (IPPO) with non-shared parameters, and an important next step would be to investigate how these findings extend to algorithms with parameter sharing and methods with centralised critics such as MAPPO (Yu et al., 2022). Furthermore, while our study spanned several distinct environments, extending this diagnostic approach across an even wider range of MARL benchmarks would help to build a more comprehensive understanding of the MARL evaluation landscape.

7 Acknowledgements

We would like to thank Aris Filos-Ratsikas for fruitful discussions on early versions of this work. An author on this project has received funding towards this work from the European Union's Horizon Europe research and innovation programme under grant agreement No. 101120726.

References

Stefano V Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018.

Stefano V. Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-Agent Reinforcement Learning: Foundations and modern approaches*. MIT Press, 2024.

Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965.

Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:37567–37593, 2023.

Jakob Foerster, Francis Song, Edward Hughes, Neil Burch, Iain Dunning, Shimon Whiteson, Matthew Botvinick, and Michael Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 1942–1951. PMLR, 2019.

Samuel Garcin, Trevor McInroe, Pablo Samuel Castro, Prakash Panangaden, Christopher G Lucas, David Abel, and Stefano V Albrecht. Studying the interplay between the actor and critic representations in reinforcement learning. *arXiv preprint arXiv:2503.06343*, 2025.

Matthew J Hausknecht and Peter Stone. Deep recurrent q-learning for partially observable mdps. In *AAAI fall symposia*, volume 45, pp. 141, 2015.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Hengyuan Hu, Adam Lerer, Brandon Cui, Luis Pineda, Noam Brown, and Jakob Foerster. Off-belief learning. In *International Conference on Machine Learning*, pp. 4369–4379. PMLR, 2021.

Leslie Pack Kaelbling, Michael L Littman, and Anthony R Cassandra. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 69(6):066138, 2004.

Christopher Lusena, Judy Goldsmith, and Martin Mundhenk. Nonapproximability results for partially observable Markov Decision Processes. *Journal of artificial intelligence research*, 14:83–103, 2001.

Ruaridh Mon-Williams, Max Taylor-Davies, Elizabeth Mieczkowski, Natalia Velez, Neil R Bramley, Yanwei Wang, Thomas L Griffiths, and Christopher G Lucas. Partner modelling emerges in recurrent agents (but only when it matters). *arXiv preprint arXiv:2505.17323*, 2025.

Hadi Nekoei, Xutong Zhao, Janarthanan Rajendran, Miao Liu, and Sarath Chandar. Towards few-shot coordination: Revisiting ad-hoc teamplay challenge in the game of hanabi. In *Conference on Lifelong Learning Agents*, pp. 861–877. PMLR, 2023.

Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized pomdps. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

Frans A Oliehoek, Christopher Amato, et al. *A concise introduction to decentralized POMDPs*, volume 1. Springer, 2016.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of Markov Decision Processes. *Mathematics of Operations Research*, 12(3):441–450, 1987.

Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V Albrecht. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. *arXiv preprint arXiv:2006.07869*, 2020.

Georgios Papoudakis, Filippos Christianos, and Stefano Albrecht. Agent modelling under partial observability for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:19210–19222, 2021.

Bei Peng, Tabish Rashid, Christian Schroeder de Witt, Pierre-Alexandre Kamienny, Philip Torr, Wendelin Böhmer, and Shimon Whiteson. Facmac: Factored multi-agent centralised policy gradients. *Advances in Neural Information Processing Systems*, 34:12208–12221, 2021.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Brian C Ross. Mutual information between discrete and continuous data sets. *PloS one*, 9(2): e87357, 2014.

Alexander Rutherford, Benjamin Ellis, Matteo Gallici, Jonathan Cook, Andrei Lupu, Gardar Ingvarsson, Timon Willi, Akbir Khan, Christian Schroeder de Witt, Alexandra Souly, et al. Jaxmarl: Multi-agent rl environments in jax. *arXiv preprint arXiv:2311.10090*, 2023.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder De Witt, Gregory Farquhar, Nantas Nardelli, Tim GJ Rudner, Chia-Man Hung, Philip HS Torr, Jakob Foerster, and Shimon Whiteson. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, Cambridge, MA, 1998.

Xihuai Wang, Zheng Tian, Ziyu Wan, Ying Wen, Jun Wang, and Weinan Zhang. Order matters: Agent-by-agent policy optimization. In *The Eleventh International Conference on Learning Representations*, 2023.

Erik Wijmans, Manolis Savva, Irfan Essa, Stefan Lee, Ari S Morcos, and Dhruv Batra. Emergence of maps in the memories of blind navigation agents. In *The Eleventh International Conference on Learning Representations*, 2023.

Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on robot learning*, pp. 575–588. PMLR, 2021.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

Yifan Zhong, Jakub Grudzien Kuba, Xidong Feng, Siyi Hu, Jiaming Ji, and Yaodong Yang. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 25(32): 1–67, 2024.

Luisa Zintgraf, Sam Devlin, Kamil Ciosek, Shimon Whiteson, and Katja Hofmann. Deep interactive bayesian reinforcement learning via meta-learning. *In International Conference on Autonomous Agents and Multi-Agent Systems*, 2021.

A Appendix

A.1 Prediction Game: Heuristic Agents

A heuristic (non-learning) agent i follows a simple periodic policy. At environment step t its action is

$$a^i(t) = \Big((i \bmod A) \; + \; \Big\lfloor \frac{t}{k_i} \Big\rfloor + \phi_i \Big) \bmod A,$$

where

- $A = |\mathcal{A}|$ is the number of discrete actions;
- $k_i \in \mathbb{N} > 0$ is the **cycle length**: the agent repeats the same action for k_i steps before advancing to the next in a modulo-A loop;
- $\phi_i \sim \mathrm{Uniform}\{0, \ldots, k_i - 1\}$ is a fresh **initial phase** drawn at the start of every episode, so each episode begins at a random point in the cycle.

Example. Assume $A = 4$ (actions $\{0, 1, 2, 3\}$).

- **Agent 0** with $k_0 = 3$ and $\phi_0 = 1$ executes the sequence

$$(1, 1, 1, \; 2, 2, 2, \; 3, 3, 3, \; 0, 0, 0, \ldots).$$

- **Agent 2** with $k_2 = 2$ and $\phi_2 = 0$ executes

$$(2, 2, \; 3, 3, \; 0, 0, \; 1, 1, \; 2, 2, \ldots).$$

This produces predictable but non-trivial periodic behaviour. Furthermore, learning agents only have $1/4$ of selecting the first action correctly since there is no previous action to help them make this selection and no information from previous episodes that can be used to determine the starting point in a cycle.

A.2 Prediction Game: Additional Plots

In Figure 6, we show the action histograms of our concurrently learning agents during evaluation. We see that they form conventions and stick to specific action patterns.

A.3 MaBrax: Additional Results

Here we show more detailed results of the individual MaBrax tasks in figure 7. In 8 we show a comparison of results of blind agents who don't receive an observation at all.

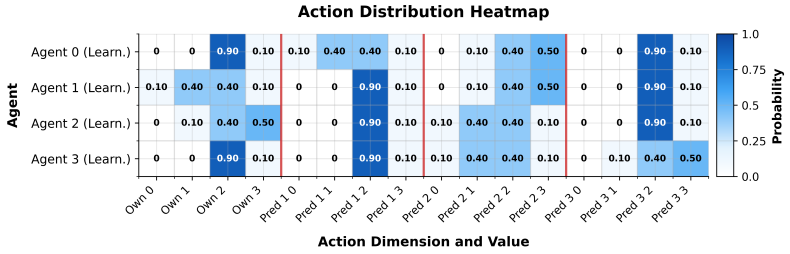
A.4 Hyperparameters

A.4.1 Prediction Game

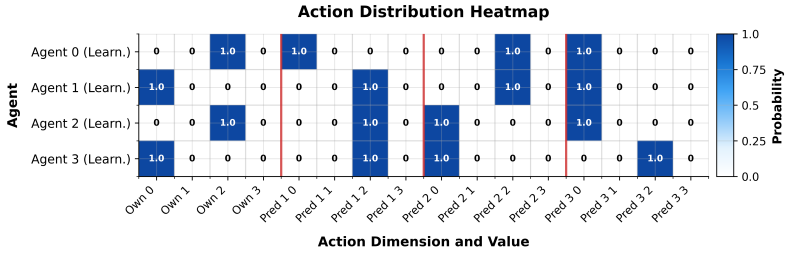
For each experiment in the Prediction Game, we perform a sweep over the following hyperparameters: learning rate (LR $\in \{1\times10^{-4}, 3\times10^{-4}, 5\times10^{-4}, 1\times10^{-3}\}$), clipping epsilon (CLIP_EPS $\in \{0.1, 0.2, 0.5\}$), and whether to anneal the learning rate (ANNEAL_LR $\in$ {True, False}). We report the best performance per method in Table 2.

A.4.2 MaBrax

For each MaBrax environment, we perform random sweeps over 32 different learning rates, where LR $\in [0.0001, 0.01]$. Each selected learning rate is evaluated across 5 random seeds, separately for both the RNN and feed-forward (FF) implementations. The final selected hyperparameters are provided in Table 3.



(a) FF Action Histogram



(b) RNN Action Histogram

Figure 6: Action Histograms for FF and RNN IPPO agents on the Prediction Game, evaluated over 1000 episodes at the end of training.

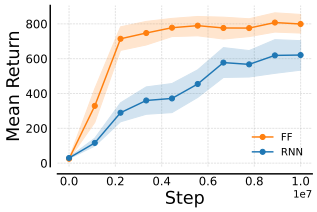
Table 2: Hyperparameters used for Prediction Game

Hyperparameter	**RNN**	**FF**
Network parameters		
Agent parameter sharing	False	False
Embedding dimension	128	-
GRU hidden dimension	128	-
Actor hidden dimension	-	128
Critic hidden dimension	-	128
Activation function	relu	relu
Training parameters		
Total time steps	1.0×10^7	1.0×10^7
Number of steps	128	128
Number of environments	16	16
Number of evaluation episodes	-	-
Number of seeds	10	10
Update epochs	4	4
Number of minibatches	4	4
Learning rate annealing	False	True
Learning rate	5.0×10^{-4}	2.5×10^{-4}
Entropy coefficient	1.0×10^{-2}	1.0×10^{-2}
Clipping epsilon	0.2	0.2
Scale clipping epsilon	False	-
Ratio clipping epsilon	-	-
Gamma	0.99	0.99
GAE lambda	0.95	0.95
Value function coefficient	0.5	0.5
Max gradient norm	0.5	0.5

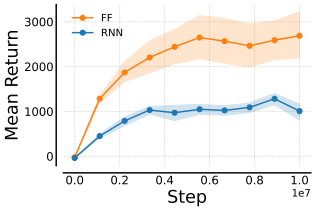
Table 3: Hyperparameters used for MaBrax Experiments

Hyperparameter	**RNN**	**FF**
Network parameters		
Agent parameter sharing	False	False
Embedding dimension	128	-
GRU hidden dimension	128	-
Actor hidden dimension	-	128
Critic hidden dimension	-	128
Activation function	tanh	tanh
Training parameters		
Total time steps	1.0×10^7	1.0×10^7
Number of steps	64	64
Number of environments	256	256
Number of evaluation episodes	32	32
Number of seeds	16	16
Update epochs	4	4
Number of minibatches	4	4
Learning rate annealing	False	False
Learning rate Ant 2×4	3.0×10^{-3}	3.0×10^{-4}
Learning rate Half-Cheetah 6×1	2.1×10^{-4}	1.0×10^{-3}
Learning rate Hopper 3×1	1.0×10^{-3}	2.5×10^{-3}
Learning rate Humanoid 8|9	1.8×10^{-3}	8.5×10^{-4}
Learning rate Walker2d 2×3	9.5×10^{-4}	3.4×10^{-3}
Entropy coefficient	1.0×10^{-4}	1.0×10^{-4}
Clipping epsilon	0.2	0.2
Scale clipping epsilon	False	False
Ratio clipping epsilon	False	False
Gamma	0.99	0.99
GAE lambda	0.95	0.95
Value function coefficient	1.0	1.5
Max gradient norm	0.5	0.5
Adam epsilon	1.0×10^{-8}	1.0×10^{-8}
Advantage unroll depth	8	8

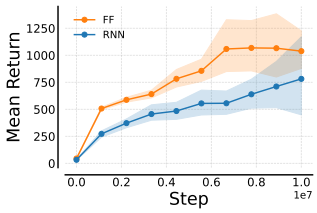
(a) MABrax environments (Peng et al., 2021; Rutherford et al., 2023)



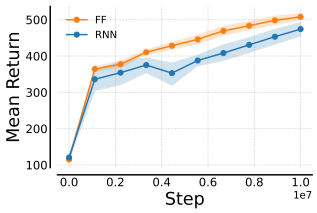
(b) Ant 4×2



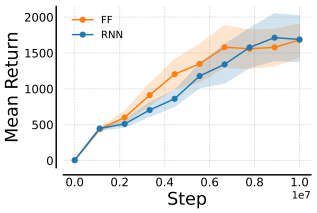
(c) Half Cheetah 6×1



(d) Hopper 3×1

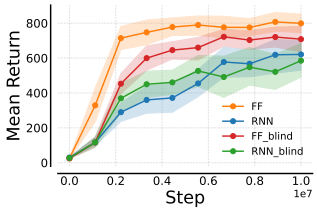


(e) Humanoid 8|9

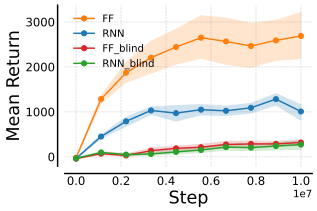


(f) Walker 2×3

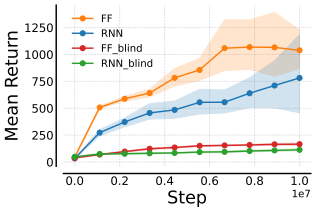
Figure 7: Top: MABrax environment suite and sample tasks. Bottom: means and 95% bootstrapped confidence intervals across 16 seeds for each environment.



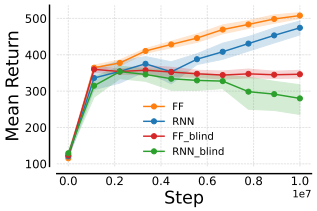
(a) Ant 4×2



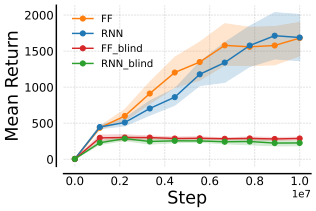
(b) Half Cheetah 6×1



(c) Hopper 3×1



(d) Humanoid 8|9



(e) Walker 2×3

Figure 8: MABrax environments comparing partially-observable, to blind performances showcasing mean returns and 95% bootstrapped confidence intervals across 16 seeds

A.4.3 Hanabi

For Hanabi we use the same hyperparameter settings as JaxMARL (Rutherford et al., 2023), these can be found in Table 4 below.

A.4.4 SMAX

For SMAX we use the hyperparameters from JAxMARL Rutherford et al. (2023), as shown in Table 5.

Table 4: Hyperparameters used for Hanabi Experiments

Hyperparameter	**FF**	**RNN**
Network parameters		
Agent parameter sharing	False	False
Embedding dimension	-	128
GRU hidden dimension	-	128
Actor hidden dimension	128	-
Critic hidden dimension	128	-
Activation function	tanh	tanh
Training parameters		
Total time steps	1.0×10^{10}	1.0×10^{10}
Number of steps	128	128
Number of environments	1024	1024
Number of evaluation episodes	128	128
Number of seeds	8	8
Number of checkpoints	256	256
Update epochs	4	4
Number of minibatches	4	4
Learning rate annealing	True	True
Learning rate	5.0×10^{-4}	5.0×10^{-4}
Entropy coefficient	1.0×10^{-2}	1.0×10^{-2}
Clipping epsilon	0.2	0.2
Scale clipping epsilon	False	False
Ratio clipping epsilon	False	False
Gamma	0.99	0.99
GAE lambda	0.95	0.95
Value function coefficient	1.0	1.0
Max gradient norm	0.5	0.5
Adam epsilon	1.0×10^{-8}	1.0×10^{-8}
Advantage unroll depth	8	8

Table 5: Hyperparameters used for SMAX Experiments

Hyperparameter	**FF**	**RNN**
Network parameters		
Recurrent	False	True
GRU hidden dimension	-	128
Fully connected dimension	-	128
Activation function	relu	relu
Training parameters		
Total time steps	1.0×10^7	1.0×10^7
Number of steps	128	128
Number of environments	128	128
Update epochs	4	2
Number of minibatches	4	2
Learning rate annealing	True	True
Learning rate	4.0×10^{-3}	4.0×10^{-3}
Entropy coefficient	0.0	0.0
Clipping epsilon	0.1	0.2
Scale clipping epsilon	-	False
Gamma	0.99	0.99
GAE lambda	0.95	0.95
Value function coefficient	0.5	0.5
Max gradient norm	0.5	0.5
Seed	30	30
Map name	smacv2_5_units	smacv2_5_units
See enemy actions	True	True
Walls cause death	True	True
Attack mode	closest	closest
Max steps	100	100