Laser Learning Environment: A new environment for coordination-critical multi-agent tasks

Yannick Molinghen
MLG, Université Libre de Bruxelles and
AI lab, Vrije Universiteit Brussel
yannick.molinghen@ulb.be

Raphaël Avalos
AI lab, Vrije Universiteit Brussel
raphael.avalos@vub.be

Mark Van Achter
KU Leuven and
Strides
mark.vanachter@kuleuven.be

Ann Nowé
AI lab, Vrije Universiteit Brussel
ann.nowe@vub.be

Tom Lenaerts
MLG, Université Libre de Bruxelles,
AI lab, Vrije Universiteit Brussel and
Center for Human Compatible AI, UC Berkeley
tom.lenaerts@ulb.be

Abstract

We introduce the Laser Learning Environment (LLE), a collaborative multi-agent reinforcement learning environment in which coordination is central. In LLE, agents depend on each other to make progress (*interdependence*), must jointly take specific sequences of actions to succeed (*perfect coordination*), and accomplishing those joint actions does not yield any intermediate reward (*zero-incentive dynamics*). The challenge of such problems lies in the difficulty of escaping state space bottlenecks caused by interdependence steps since escaping those bottlenecks is not rewarded. We test multiple state-of-the-art value-based MARL algorithms against LLE and show that they consistently fail at the collaborative task because of their inability to escape state space bottlenecks, even though they successfully achieve perfect coordination. We show that Q-learning extensions such as prioritised experience replay and n-steps return hinder exploration in environments with zero-incentive dynamics, and find that intrinsic curiosity with random network distillation is not sufficient to escape those bottlenecks. We demonstrate the need for novel methods to solve this problem and the relevance of LLE as cooperative MARL benchmark.

1 Introduction

Many problems, ranging from societal to technological, are inherently multi-agent and often require coordination (or cooperation) among the agents to achieve individually and collectively defined goals (Cao et al., 2013; Klima et al., 2018). While evolution has provided humans with the skills to deal with such tasks, researchers and engineers have to train the artificial agents in a much shorter time span to also manage such tasks.

We are interested in Reinforcement Learning (Sutton and Barto, 2018, RL) as one of the branches of Machine Learning that holds the promise of training such agents by interacting with their environment.

16th European Workshop on Reinforcement Learning (EWRL 2023).

Deep RL has made dazzling progress in recent years with deep Q-learning, showing human or superior to human performance in a wide range of single-agent situations (Mnih et al., 2015), and this progress influenced Multi-Agent Reinforcement Learning (Panait and Luke, 2005, MARL).

In comparison to single-agent RL, centralised MARL faces the issue of the exponential growth of state and joint action spaces with the number of agents, making this approach intractable even for relatively small problems. Decentralised MARL approaches avoid the exponential growth of the action space at the cost of nonstationarity (Laurent et al., 2011): Since multiple learning agents adapt their policy over time, each agent continuously has to adapt to the changing policy of the other agents, making it more challenging to acquire robust and general policies. To mitigate that effect, Oliehoek et al. (2008a) introduce the paradigm of Centralised Training with Decentralised Execution (CTDE) that has demonstrated how successful it can be in complex cooperative multi-agent tasks (Sunehag et al., 2018; Rashid et al., 2018; Avalos et al., 2022).

In the last five years, a variety of environments have been developed for cooperative MARL. Among others, the Multi-agent Particle Environment (Lowe et al., 2017; Mordatch and Abbeel, 2017, MPE), the StarCraft Multi-Agent Challenge (Samvelyan et al., 2019, SMAC), the Hanabi Learning Environment (Bard et al., 2020, HLE) and Overcooked (Wu et al., 2021) respectively aim at studying different aspects of cooperative multi-agent problem-solving such as the ability to infer the intentions of other agents or the emergence of basic compositional language.

Contributions In this work, we focus on fully cooperative multi-agent problems where agents optimise a single shared reward. We identify a category of such problems that is not well studied, introduce the Laser Learning Environment (LLE) that fits in that category and test state-of-the-art CTDE methods against it. To the best of our knowledge, LLE exhibits a unique combination of three properties: 1) *perfect coordination*: failing to coordinate can be fatal; 2) *interdependence*: agents need each other to progress; 3) *zero-incentive dynamics*: key steps toward success are not rewarded. We then show how agents successfully achieve perfect coordination but are unable to overcome the zero-incentive dynamics and escape state space bottlenecks, even when using Q-learning extensions such as prioritised experience replay (Schaul et al., 2016) and n-steps return (Watkins, 1989). Finally, we show that intrinsic curiosity with Random Network Distillation (Burda et al., 2018) does not overcome the state space bottlenecks created by the interdependence of the agents. Together, our results demonstrate that LLE is a relevant benchmark for future work and aim to point researchers in new relevant directions of cooperative MARL research.

The code for the environment is publicly available at `https://github.com/yamoling/lle`.

2 Background

2.1 Multi-agent Markov Decision Process

A Multi-agent Markov Decision Process (Boutilier, 1996, MMDP) is described as a tuple $\langle n, S, A, T, R, s_0, s_f, \gamma \rangle$ where n is the number of agents, S is the set of states, $A \equiv A_1 \times \cdots \times A_n$ is the set of joint actions and A_i is the set of actions of agent i, $T\colon S \times A \to \Delta_S$ is a function that gives the probability of transitioning from state s to state s' by taking action a, $R : S \times A \times S \to \mathbb{R}$ is the function that gives the reward obtained by transitioning from s to s' by performing joint action $\boldsymbol{a}$, $s_0 \in S$ is the initial state, s_f is the final state and $\gamma \in [0, 1)$ is a discount factor.
A transition is defined as $\tau = \langle s, \boldsymbol{a}, r, s' \rangle$ with $s, s' \in S, \boldsymbol{a} \in A, r \in \mathbb{R}$. An episode of length l is a sequence of transitions $\tau_1, \ldots, \tau_l$ such that $\tau_1 = \langle s_0, \boldsymbol{a}, r, s' \rangle$ and $\tau_l = \langle s_l, \boldsymbol{a}, r, s_f \rangle$. Each agent i acts according to a policy $\pi_i\colon S \to \Delta_{A_i}$ and their objective is to find the joint policy $\boldsymbol{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$ that maximises their expected discounted reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R_t | s = s_0, \boldsymbol{\pi}\right]$. The action-value function of a policy measures the expected return obtained by taking action $\boldsymbol{a}$ is state s by following policy $\boldsymbol{\pi}$. In particular, we define the joint action-value function $Q^{\boldsymbol{\pi}} : S \times \boldsymbol{A} \to \mathbb{R}$.
Scenarios where agents receive individual observations of the state instead of the full state are referred to as Decentralised Partially Observable Markov Decision Processes (Oliehoek and Amato, 2016).

2.2 *Q*-value factorisation

Laurent et al. (2011) showed that multi-agent systems suffer from a non-stationarity problem (Tuyls and Weiss (2012) also refer to it as the multi-agent moving target problem) because learning agents

perceive the other learning agents as part of the environment. Claus and Boutilier (1998) have shown that naive implementations of Independent Q-Learning (IQL) were often unsuccessful, even for very simple tasks, partly because of this non-stationarity.

To tackle this non-stationarity problem, Sunehag et al. (2018) introduce Value Decomposition Network (VDN), an algorithm based on the concept of Q-value factorisation (Oliehoek et al., 2008b) in which each agent i has its own utility function $Q_i : S \times A_i \to \mathbb{R}$. VDN decomposes the joint Q-value into a simple sum of the agents' utility. This factorisation allows for decentralised execution thanks to the Individual Global Max property (Son et al., 2019, IGM) that ensures consistency in action selection between the centralised training and the decentralised execution. QMIX (Rashid et al., 2018) extends VDN by allowing more complex factorisation using a monotonically increasing hype-network conditioned on the state.

2.3 Cooperative multi-agent environments

In the last few years, several cooperative multi-agent environments have emerged to study different aspects of the cooperative multi-agent problem and we give here an overview of popular environments with discrete action spaces.

The StarCraft Multi-Agent Challenge (Samvelyan et al., 2019, SMAC) comes with a wide range of maps and offers a complex cooperative partially observable problem where individual agents of the same team have to defeat the opponent team in a short skirmish. This environment has been introduced to study whether agents could learn complex behaviours such as kiting.

Overcooked (Wu et al., 2021) is a cooperative environment in which multiple agents receive recipes and have to cook them before serving them on a plate. On top of spatial movements, this environment has been introduced to study the ability of agents to infer the hidden intentions of others as well as the ability of agents to learn when to split the tasks amongst themselves (divide and conquer) and when to work on the same task (cooperate).

Hanabi Learning Environment (Bard et al., 2020, HLE) is an environment that comes from a board game with the same name. In Hanabi, every player plays one after the other and has to decide whether to play a card or give a clue to a teammate. This environment is well suited to study reasoning, beliefs and intentions of other players.

The Multi-agent Particle Environment (Lowe et al., 2017; Mordatch and Abbeel, 2017, MPE) is a set of cooperative and competitive 2D tasks with a dense reward signal. It has both partially and fully observable variants and offers scenarios that involve communication. This environment has been introduced to study the emergence of a basic compositional language.

3 The Laser Learning Environment

We introduce here the Laser Learning Environment (LLE), a multi-agent grid world populated by agents of different colours (see Figure 1) and with different types of tiles: floor, start, wall, laser, laser source, gem, and exit tiles. The main objective in LLE is for each agent to reach an exit tile, while additional points can be gathered by collecting gems along the way. The game is cooperative since agents must help each other to pass laser beams: if an agent's colour doesn't match the colour of a laser beam, this agent dies and the game ends. However, when an agent of matching colour stands in line with a beam, it blocks the laser and allows other agents to reach areas of the map that they could not access by themselves.

LLE comes with 6 different maps (Appendix E) and can also generate random solvable maps of arbitrary sizes based on multiple parameters (number of agents, number of gems, number of lasers and wall density). Randomly generated maps can also be constrained to some degree of difficulty depending on how many steps of *perfect coordination* (see Section 3.1) are required to complete the level. While future work will address the issue of curriculum learning (Parker-Holder et al., 2022), in this work, we focus on the configuration visualised in Figure 1.

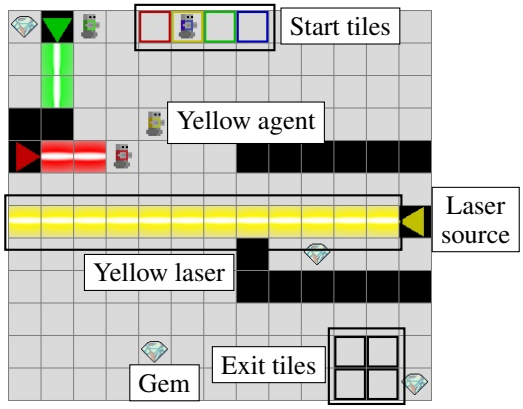


Figure 1: Level 6 of LLE, which has 4 agents, 3 lasers and 4 gems. Agent red blocks the red laser, making it possible for the other agents to pass to the lower part of the grid world. Additional blocking of the yellow laser is required for them to all pass and reach the exit tiles. The red and green agents need to coordinate their actions to achieve the collective goal of this game.

3.1 Motivations

As explained in Section 2.3, there already exist several multi-agent cooperative environments that are suitable for studying various fields of MARL. However, LLE aims at studying a new range of problems. As far as we were able to find the information, LLE introduces new complexities in the study of cooperative problem solving due to a combination of three properties: *perfect coordination*, *interdependence* and *zero incentive dynamics*.

Perfect coordination is defined as the property of a policy in which agents simultaneously take a specific sequence of actions such that any agent that singlehandedly deviates from this policy in state s_t would directly lead to a penalty in state s_{t+1} and possibly the early termination of the game.

Unlike all the environments presented in Section 2.3, LLE requires agents to achieve perfect coordination when blocking lasers. Focussing on agents red and yellow in Figure 1, it requires perfect coordination for agent yellow to cross the red laser: if agent yellow goes down and agent red releases the laser, agent yellow would die and the game would terminate with a punishment. Although Hanabi has the comparable property of directly punishing agents that fail to coordinate, the environment is such that agents play one after the other. For that reason, Hanabi does not require *perfect coordination*.

Interdependence expresses how much a set of agents relies on another set of agents to perform a particular sequence of actions in order to make progress in the collective task.

Interdependence introduces *bottlenecks in the state space* in the steps that require coordination. A high level of interdependence means that from the start state, the end state that maximises the collective expected discounted return is located behind such interdependence bottlenecks. Inversely, a low level of interdependence means that agents can explore most of the state space without relying on each other.

The maps presented in Appendix E have been designed with increasing levels of interdependence in mind, with level 6 having the highest level of interdependence. In comparison, SMAC, MPE and HLE challenge agents with situations where any agent can explore the state space, sometimes even finish the game, regardless of their teammates' actions.

Overcooked explicitly uses the concept of sub-tasks in the form of recipes that also enclose the reachable state space until the current sub-task is solved. However, agents can most of the time accomplish those sub-tasks by themselves and are therefore not interdependent. Arguably, there is *interdependence* in completely split maps where agents do not have access to every kitchen tool (e.g.: full-divider map) and must pass items over the counter to the other to complete recipes.

Zero-incentive dynamics defines the property of an environment in which overcoming bottlenecks in the state space is not rewarded. Intuitively, an environment with zero-incentive dynamics does not reward agents for succeeding at key (cooperative) dynamics.

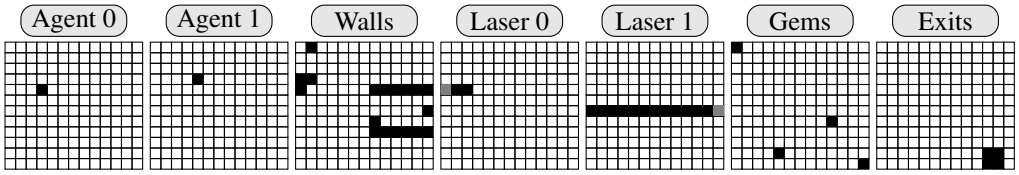


Figure 2: Representation of the state shown in Figure 1. The layers "Agent 2", "Agent 3", "Laser 2" and "Laser 3" were omitted for the sake of conciseness. Each layer encodes the location of a specific type of object of the grid world (walls, agents' locations, ...). White squares represent 0s, black squares are 1s and grey squares are -1.

As detailed in Section 3.4, blocking lasers in LLE does not provide any reward, and laser-blocking are state space bottlenecks caused by interdependence. Consequently, LLE has zero-incentive dynamics. Overcooked on the other hand provides rewards for finishing recipes and therefore does not have zero-incentive dynamics, even in the full-divider map that does have interdependence. We discuss in Section 4.1 that the combination of interdependence and zero-incentive dynamics makes it difficult for agents to escape regions surrounded by lasers during the training process, which makes LLE a challenging exploration task.

We distinguish zero-incentive dynamics from temporal credit assignment (Sutton, 1984), which is related to the ability of a policy to determine which action takes credit for some later reward, and not a property of the environment.

3.2 States

The state of LLE encodes the location of grid-world elements in a layered fashion illustrated in Figure 2. These items are namely agents, walls, lasers, laser sources, gems and exits locations. There is one layer per agent and one layer per laser colour. This representation has the advantage of being very generic as long as the size of the map and the number of agents remain identical, allowing future work in the field of generalisation and curriculum learning Parker-Holder et al. (2022).

3.3 Actions

The action space of LLE is discrete and the possible actions are `NORTH`, `EAST`, `SOUTH`, `WEST` and `STAY`, which is required for laser-blocking purposes. Actions are identical for all agents.

LLE prevents agents from entering invalid tiles by providing the set of available actions for each agent at each time step. An agent cannot enter walls, move beyond the grid boundaries or enter a tile that is currently occupied by another agent. This prevents multiple types of conflicts that can occur in grid world problems such as edge, following and swapping conflicts (Stern et al., 2017). Additionally, once an agent enters an exit tile, it cannot leave it anymore and the only action allowed is `STAY`.

Vertex conflicts – when two or more agents enter the same tile – can unfortunately not be prevented with information on action availability. As a result, when agents provoke a vertex conflict by trying to move to the same tile, their actions are replaced by `STAY`.

3.4 Rewards

The reward function of LLE has a single scalar output as the team reward, which is the result of the joint action of the agents at a given time step. Collecting a gem or entering an exit tile provides a collective reward of $+1$. Finishing the game, i.e. when all agents are on an exit tile, also provides an additional reward of $+1$. However, if any agent dies at a time step, the reward is set to -1 times the number of agents that have died.

3.5 Metrics

Two metrics are used here to evaluate the performance of the LLE agents: the score and the exit rate.

The **score** refers to the undiscounted sum of rewards throughout an episode. This is a natural metric to analyse since agents try to maximise the *discounted* sum of rewards over the course of an episode.

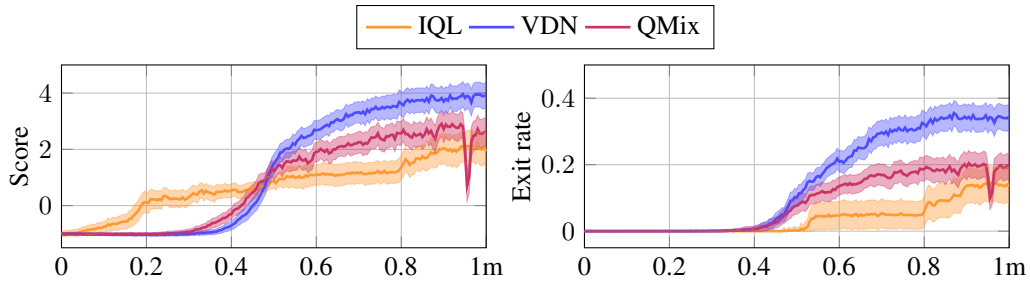


Figure 3: Training score and exit rate over time for IQL, VDN and QMIX on level 6 (Figure 1). The maximal achievable score is 9. Results averaged on 20 different seeds and shown with 95% confidence interval, capped by the minimum and maximum.

The maximal score of an LLE environment can be calculated as follows: given a map with n agents and g gems, we can compute the maximum score as maximum score $= n + g + 1$.

The **exit rate** is defined as the proportion of agents that reach an exit tile at the end of an episode. Since the objective in every LLE is to exit the level, this metric gives an estimation of how close agents are to the objective. An exit rate of 1 means that all agents have successfully exited the level.

The combination of those two metrics gives insight into the agents' behaviour. As long as the exit rate is below 1, it means that the agents are not able to finish the level. If the exit rate is 1, then analysing the score allows us to know if agents have collected all the gems.

3.6 Implementation

LLE is implemented in Rust which makes it an extremely fast environment. LLE comes with a strongly type hinted Python interface for seamless integration with common MARL libraries and is extensively tested in both Python and Rust.

4 Experiments

We analyse the score and exit rate on level 6[1] shown in Figure 1 with Independent deep Q-Learning (Mnih et al., 2015, IQL), Value Decomposition Network (Sunehag et al., 2018, VDN) and QMIX (Rashid et al., 2018). Then, we discuss in Section 5 the results with regard to perfect coordination, interdependence and zero-incentive dynamics.

In their respective papers, VDN and QMIX have been introduced in the scope of partially observable environments while LLE is fully observable. We treat the fully observable case as the particular case of Dec-POMDP where the individual observations are equal to the state. We motivate the usage of QMIX by the fact that it has demonstrated its superior representational capability in comparison to VDN with a fully observable example referred to as *Two-Step game* (Rashid et al., 2018). Therefore, we use a feed-forward neural network instead of a recurrent one.

Experimental setup In our experiments, agents interact with the environment for one million time steps and policies are updated every 5 steps on a batch of 64 transitions. Agents use an ϵ-greedy policy linearly annealed from 1 to 0.05 over 500k time steps and use a replay memory of 50k transitions. We use Double Q-learning for all algorithms. The target network is updated via soft updates with $\tau = 0.01$ (Lillicrap et al., 2016) which has experimentally provided better results in our experiments than periodic hard updates of the target network. All hyperparameters can be found in Appendix B.

The utility values Q_i of each agent $i \in \{1, \ldots, n\}$ are estimated with a convolutional neural network (LeCun et al., 1998) to take advantage of the spatial nature of the layered observations. Since the observations are designed such that every *pixel* gives information, the stride of the convolutions is 1. Appendix C provides a more detailed description of the neural network architecture. Since LLE is fully observable and agents share the same neural network weights, we concatenate the flattened

[1]Results for each level can be found in Appendix F.

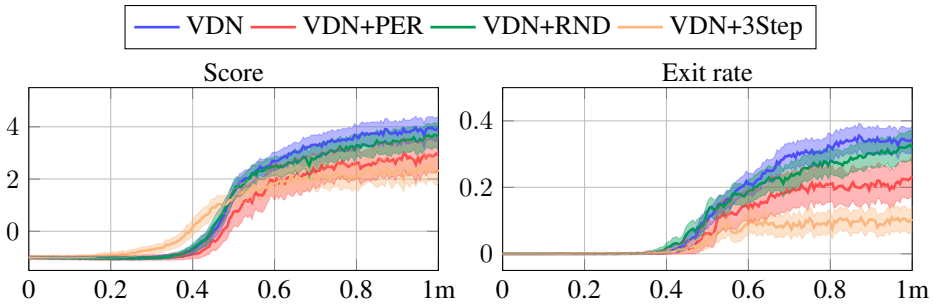


Figure 4: Training score and exit rate over training time for VDN, VDN with PER, VDN with RND and VDN with 3-step return on level 6. The maximal score that agents can reach on level 6 of an episode is 9. Results are averaged on 20 different seeds and shown with 95% confidence intervals

output of the CNN with a one-hot encoding of the agent id. This allows the neural network to output different Q_i-values for each agent.

We define the maximal episode duration to be $\lfloor \frac{width \times height}{2} \rceil$ steps after which the episode is truncated and taken to an end. This heuristic for episode duration allows the agents to discover a lot of dynamics of the environment without polluting the replay memory too much with useless transitions. For instance in level 6, if agents red and yellow reach the exit tiles but did not open the way for blue and green, then the remaining time steps of the episode would just be the latter two waiting behind the beam.

Foreanalysis Level 6 (Figure 1) is of size 12×13 and has 4 agents and 4 gems. The maximal score is hence $4 + 4 + 1 = 9$ as explained in Section 3.5. The optimal policy in level 6 is the following: **i)** Agent green should collect the gem in the top left corner; **ii)** Agent red should block the red laser and wait for every other agent to cross; **iii)** Agent yellow should cross the red laser and collect the gem that only he can collect near the yellow source; **iv)** Agent yellow should block the laser for every agent to cross; **v)** Agents should collect the remaining gems on the bottom half; **vi)** Agents should go to the exit tiles. The length of such an episode is ≈ 30 time steps, well below the time limit of $\lfloor \frac{12 \times 13}{2} \rceil = 78$ steps.

4.1 Baseline results

Figure 3 shows the mean score and exit rate over the course of training on level 6 (Figure 1). VDN performs best on this map. That being said, none of the algorithms ever reaches the highest possible score of 9 and at most half of the agents ever reach the end exit tiles.

Looking into the results, the best policy learned only completes items i), iii) and v). Agents red and yellow escape the top half of the map, collect gems on that side and reach the exit, while agents green and blue are not waited for. This policy yields a score of 6 and an exit rate of 0.5. We could introduce reward shaping in order to drive the agents towards a better solution more easily. However, reward shaping is notoriously difficult to achieve properly and can drive agents towards unexpected (and likely undesired) behaviours (Amodei et al., 2016).

4.2 Results with Q-learning extensions

When a policy is not successful enough, there exists a few common approaches to try and improve the learning of the policy. We take VDN as our baseline since it provides the best results in our experiments and we combine it with Prioritised Experience Replay (Schaul et al., 2016, PER), n-step return (Watkins, 1989) and intrinsic curiosity (Schmidhuber, 1991) on top of it and analyse their impact on the learning process.

Prioritised Experience Replay is a technique used in off-policy reinforcement learning to enhance learning efficiency by prioritising experiences and sampling them based on their informativeness. The intuition is to sample past experiences whose Q-values are poorly estimated more often and hope

that when agents discover a better policy than their current one, this policy would be prioritised. In our setting, we hope that if agents ever complete the level, this experience would be prioritised.

As Schaul et al. suggest, we use the temporal difference error as the priority. We have performed a hyperparameter search on α and β that respectively control the exponential scale of priorities and the exponential scale of importance sampling weights with values ranging from 0.3 to 0.8 and have found the best values to be $\alpha = 0.6$ and $\beta = 0.5$, where β is annealed from 0.5 to 1 on the course of the training (1m steps).

Figure 4 shows that prioritised sampling performs worse than uniform sampling overall. We hypothesise that PER hinders exploration in the early stages of the training because of the zero-incentive dynamics of the game and discuss this further in Section 5.2 and Section 5.3 with regard to interdependence and zero-incentive dynamics respectively.

***N*-step return** is a technique that aims at fastening the bootstrapping process (Sutton and Barto, 2018) by propagating rewards up to n-steps into the past. In the scope of LLE, the idea here is to propagate the reward for collecting gems faster to the step of laser blocking. We have tried values for $n \in \{3, 5, 7, 9\}$ and found $n = 3$ to give the highest score on average.

Figure 4 show that 3-step return yields worse results than any other variant. We relate this poor performance to the high probability of dying from lasers while exploring, resulting in agents learning more conservative policies. We relate this phenomenon to the zero-incentive dynamics of the game and discuss this topic further in Section 5.3.

Intrinsic curiosity aims at introducing bias in the learning process to encourage agents to explore unknown states. During learning, intrinsic curiosity adds an extra reward referred to as the intrinsic reward, thereby altering the updates of the Q-function. With this method, we hope that agents would learn faster about laser-blocking and therefore allow them to escape state space bottlenecks.

Intrinsic curiosity with Random Network Distillation (Burda et al., 2018, RND) has proven to work well in single-agent environments with zero-incentive dynamics such as Montezuma's Revenge, where the agent must first collect a torch (unrewarded) to be able to light up a dark room (unrewarded) and in the end, collect a treasure (rewarded). Similarly, the objective of using RND is to quicken the discovery of the laser-blocking dynamic and encourage agents to explore the state space.
We linearly anneal the intrinsic reward from a factor 2 down to 0 over the course of the training (1m steps), clip the intrinsic reward to be lower or equal to 5 and warm up the RND for 64 updates before issuing any intrinsic reward different from 0.

Figure 4 shows that RND performs very similarly to the baseline and does not enable the agents to escape more state space bottlenecks and therefore to learn significantly better policies.

5 Discussion

5.1 Perfect coordination

We analyse the policy that agents learn with the VDN baseline and show their ability to achieve perfect coordination. VDN has been chosen over QMIX for interpretability reasons and because it has been the most successful algorithm in our tests.

Level 6 Analysing the policy of the four agents after training for 1m steps, we can see that the agents understand the laser dynamics: they learn low Q_i-values for walking into deadly lasers and for releasing a laser beam on another agent (killing it). Agents red and yellow also learn to cooperate and block lasers for each other in order to reach the bottom half of the map, collect the gems and reach the exit tiles. We argue that agents learn perfect coordination and illustrate that behaviour in a simpler toy example illustrated in Figure 5, where agents must also block a laser.

Toy example Consider the toy example depicted in Figure 5 with one laser and two agents. We train the agents for 160k steps and then analyse the Q_i-values during the steps concerned by perfect coordination in Table 1.

(a) (b) (c)

Figure 5: Four consecutive states of an episode. Agent red blocks the laser for agent yellow and waits for the latter to have left the range of the blocked beam.

Table 1: Q_i-values from the successive states depicted in Figure 5. The highest Q_i-values are in bold.

State	Agent	North	South	West	East	Stay
5a	Red	-1.29	-0.97	2.52	2.53	**2.56**
	Yellow	0.78	**0.94**	0.67	0.91	0.85
5b	Red	1.44	1.51	1.48	1.51	**1.56**
	Yellow	1.77	**2.13**	1.18	1.41	1.45
5c	Red	1.15	**1.36**	1.16	1.19	1.25
	Yellow	0.71	1.42	**1.49**	1.42	1.41

Starting with step 5a, the red agent understands that releasing the laser beam has a much lower value than the other actions because it would likely kill agent yellow. The Q_i-values suggest that the *credit* for killing an agent (and hence losing the game) is assigned to the one releasing the beam, not to the one walking in the laser span. At step 5b, the Q_i-values of agent yellow show that the yellow agent has a clear preference for the south action to collect one gem and close the gap with the exit. Meanwhile, agent red keeps blocking the laser as long as agent yellow stands in the range of the laser beam. With such behaviour similar to the one of level 6 for agent red and yellow, we conclude that agents achieve perfect coordination successfully.

5.2 Interdependence

As discussed in Section 3.1, agents interdependence introduces bottlenecks in the state space, making that category of environment challenging exploration tasks. We hypothesise that this difficulty of randomly stumbling on good policies contributes to the failure of PER: if agents never come across good policies, PER can never prioritise it. Across the 20 seeded runs with PER on level 6, we checked that at no single point in time, any episode ever reached an exit rate above 0.5. With RND however, agents *very* occasionally reached an exit rate of 0.75, suggesting that RND might be pushed further.

5.3 Zero-incentive dynamics

We argue that the zero-incentive dynamics of LLE have a detrimental effect on n-step return and on PER because they accentuate the punishment of dying from lasers each in their own way.

With regard to PER, since walking into a laser either yields no reward or a punishment (as the definition of zero-incentive dynamics implies), PER is likely to give a higher priority to transitions where punishment has occurred rather than to transitions with a null reward. As a result, we hypothesise that in the early stages of the game, PER emphasises the fact that lasers can be deadly which results in agents being more reluctant to walk into lasers overall.

Looking into the poor performance of n-step return, we observe that agents are very likely to die by walking into a laser in their exploration phase. It is also very likely that this punishment is the only non-null reward signal within the n last steps because of the reward sparsity and the zero-incentive dynamics of the environment. Consequently, when this "bad" experience (caused by a poor policy or by random exploration) is then sampled from the replay buffer, the agents learn that the n last actions eventually lead to punishment and their policy is therefore updated to be "safer". Our experimental results in Appendix D support this explanation, as the mean score decreases when n increases.

6 Conclusion

In this paper, we introduced the Laser Learning Environment (LLE), a new cooperative multi-agent grid world that, to the best of our knowledge, exhibits a unique combination of three properties: *perfect coordination*, *interdependence* and *zero-incentive dynamics*. We discussed that the interdependence property induces state space bottlenecks that are difficult to overcome.

We tested IQL, VDN and QMIX against LLE and showed that those algorithms struggled at completing the cooperative task. Our experiments demonstrated that agents successfully achieve perfect coordination but also showed that due to interdependence and zero-incentive dynamics, agents fail at long-term coordination and thus never complete the task. We highlighted that prioritised experience replay and n-steps return hinder the agents' performance because of the zero-incentive dynamics of the environment. Intrinsic curiosity with random network distillation also did not provide enough incentive to escape state space bottlenecks and did not enable agents to learn significantly better policies.

Overall, our experiments demonstrated that current state-of-the-art value-based methods fail in LLE and reveal that new benchmarks are needed in cooperative Multi-Agent Reinforcement Learning. This is why we look forward to seeing how the MARL community will approach the Laser Learning Environment and use it to tackle the challenges discussed in this paper and other topics such as generalisation, curriculum learning and inter-agent communication.

Acknowledgements

Raphaël Avalos was supported by the FWO (Research Foundation – Flanders) under the grant 11F5721N. Tom Lenaerts is supported by an FWO project (grant nr. G054919N) and two FRS-FNRS PDR (grant numbers 31257234 and 40007793). His is furthermore supported by Service Public de Wallonie Recherche under grant n° 2010235–ariac by digitalwallonia4.ai. Ann Nowé and Tom Lenaerts are also suported by the Flemish Government through the AI Research Program and TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215.

References

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul F. Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *CoRR*, abs/1606.06565, 2016. URL `http://arxiv.org/abs/1606.06565`.

Raphaël Avalos, Mathieu Reymond, Ann Nowé, and Diederik M. Roijers. Local Advantage Networks for Cooperative Multi-Agent Reinforcement Learning. In *AAMAS '22: Proceedings of the 21st International Conference on Autonomous Agents and MultiAgent Systems (Extended Abstract)*, 2022.

Nolan Bard, Jakob N. Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H. Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, Hugo Larochelle, Marc G. Bellemare, and Michael Bowling. The hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020. ISSN 00043702. doi: 10.1016/j.artint.2019.103216. URL `http://arxiv.org/abs/1902.00506`.

Craig Boutilier. Planning, learning and coordination in multiagent decision processes. *Proceedings of the 6th Conference on Theoretical Aspects of Rationality and Knowledge*, page 195–210, 1996.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018. URL `http://arxiv.org/abs/1810.12894`.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9(1): 427–438, 2013. doi: 10.1109/TII.2012.2219061.

Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98*, 1998.

Richard Klima, Daan Bloembergen, Rahul Savani, Karl Tuyls, Alexander Wittig, Andrei Sapera, and Dario Izzo. Space debris removal: Learning to cooperate and the price of anarchy. *Frontiers in Robotics and AI*, 5, 2018. ISSN 2296-9144. doi: 10.3389/frobt.2018.00054. URL https://www.frontiersin.org/articles/10.3389/frobt.2018.00054.

Guillaume J. Laurent, Laëtitia Matignon, and N. Le Fort-Piat. The world of independent learners is not markovian. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 15 (1):55–64, 2011. ISSN 18758827, 13272314. doi: 10.3233/KES-2010-0206. URL https://www.medra.org/servlet/aliasResolver?alias=iospress&doi=10.3233/KES-2010-0206.

Yann LeCun, Leon Bottou, Yoshua Bengio, and Patrick Ha. Gradient-based learning applied to document recognition, 1998.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2016. URL http://dblp.uni-trier.de/db/conf/iclr/iclr2016.html#LillicrapHPHETS15.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. ISSN 1476-4687. doi: 10.1038/nature14236. URL https://www.nature.com/articles/nature14236. Number: 7540 Publisher: Nature Publishing Group.

Igor Mordatch and Pieter Abbeel. Emergence of grounded compositional language in multi-agent populations. *arXiv preprint arXiv:1703.04908*, 2017.

Frans A. Oliehoek and Christopher Amato. *A Concise Introduction to Decentralized POMDPs*. SpringerBriefs in Intelligent Systems. Springer International Publishing, 2016. ISBN 978-3-319-28929-8. doi: 10.1007/978-3-319-28929-8. URL http://link.springer.com/10.1007/978-3-319-28929-8.

Frans A. Oliehoek, Matthijs T. J. Spaan, and Nikos Vlassis. Optimal and approximate q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008a. ISSN 1076-9757. doi: 10.1613/jair.2447. URL http://arxiv.org/abs/1111.0062.

Frans A Oliehoek, Matthijs T J Spaan, and Shimon Whiteson. Exploiting locality of interaction in factored dec-POMDPs. *Int. Joint Conf. on Autonomous Agents and Multi-Agent Systems*, pages 517–524, 2008b. URL http://hdl.handle.net/10993/11029.

Liviu Panait and Sean Luke. Cooperative multi-agent learning: The state of the art. *Autonomous Agents and Multi-Agent Systems*, 11(3):387–434, 2005. ISSN 1573-7454. doi: 10.1007/s10458-005-2631-2. URL https://doi.org/10.1007/s10458-005-2631-2.

Jack Parker-Holder, Minqi Jiang, Michael Dennis, Mikayel Samvelyan, Jakob Foerster, Edward Grefenstette, and Tim Rocktäschel. Evolving curricula with regret-based environment design. In *International Conference on Machine Learning*, pages 17473–17498. PMLR, 2022.

Tabish Rashid, Mikayel Samvelyan, and Christian Schroeder. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning, 2018.

Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philiph H. S. Torr, Jakob Foerster, and Shimon Whiteson. The StarCraft Multi-Agent Challenge. *CoRR*, abs/1902.04043, 2019.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, pages 1–21, 2016.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In Jean-Arcady Meyer, editor, *From Animals to Animats*, pages 222–227. The MIT Press, international conference on simulation adaptive behavior: from animals to animats edition, 1991. ISBN 978-0-262-25667-4. doi: 10.7551/mitpress/3115.003.0030. URL https://direct.mit.edu/books/book/3865/chapter/162771/a-possibility-for-implementing-curiosity-and.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning, 2019. URL http://arxiv.org/abs/1905.05408.

Roni Stern, Nathan R Sturtevant, Ariel Felner, Sven Koenig, Hang Ma, Thayne T Walker, Jiaoyang Li, Dor Atzmon, Liron Cohen, T K Satish Kumar, Eli Boyarski, and Roman Bart. Multi-agent pathfinding: Definitions, variants, and benchmarks, 2017.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. Value-decomposition networks for cooperative multi-agent learning based on team reward. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 3: 2085–2087, 2018. ISSN 15582914. ISBN: 9781510868083.

Richard S. Sutton and Andrew G. Barto. Reinforcement learning: an introduction, 2018.

Richard Stuart Sutton. *Temporal Credit Assignment in Reinforcement Learning*. PhD thesis, University of Massachusetts Amherst, 1984. AAI8410337.

Karl Tuyls and Gerhard Weiss. Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, 33(3):41, 2012. ISSN 2371-9621, 0738-4602. doi: 10.1609/aimag.v33i3.2426. URL https://ojs.aaai.org/index.php/aimagazine/article/view/2426.

Christopher Watkins. *Learning From Delayed Rewards*. PhD thesis, University of Cambridge, 1989.

Sarah A. Wu, Rose E. Wang, James A. Evans, Joshua B. Tenenbaum, David C. Parkes, and Max Kleiman-Weiner. Too many cooks: Coordinating multi-agent collaboration through inverse planning. *Topics in Cognitive Science*, 2021. doi: https://doi.org/10.1111/tops.12525. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/tops.12525.

A Code repository

All the code for the Laser Learning Environment is publicly available on the repository `https://github.com/yamoling/lle`.

B Hyperparameters

Hyperparameter search was performed with VDN on a combination of batch sizes (32, 64 and 128 transitions) and memory sizes (50k, 100k and 200k transitions). Then, we have tried training intervals of 1 and 5.

Then, we performed a hyperparameter search for prioritised experience replay on a combination of α (0.3, 0.4, 0.5, 0.6, 0.7, 0.8) and β (0.3, 0.4, 0.5, 0.6, 0.7, 0.8).

For random network distillation, we have explored update ratios $p = 0.25, 0.5$ and 1, then enabled or disabled annealing, and finally tried clipping the intrinsic reward to 1 or to 5.

Table 2: Hyperparameters used across all the experiments

Parameter	**Value**	**Comment**
Memory size	50 000	Transitions
Batch size	64	Transitions
Train interval	5 time steps	
Optimiser	Adam	
α (Learning rate)	0.0005	For both Q-network and mixer
Grad norm clipping	10	Clips both Q-network and mixer
γ	0.95	Discount factor
τ	0.01	Soft update rate
ϵ_{start}	1	
ϵ_{min}	0.05	
ϵ annealing	500k time steps	Linearly annealed
α	0.6	PER
β	0.5	PER
β annealing	1m time steps	PER
p	0.25	Randomly mask error from RND with probability p
IR factor	$2 \rightarrow 0$	Intrinsic Reward linearly annealed over 1m steps
IR clip	5	IR is clipped to 5 maximum
IR warmup	64	The RND is optimised 64 times before issuing any intrinsic reward

C Neural network architecture

The Q-network is made of two parts with an interconnection: a convolutional neural network of three layers, an interconnect that flattens the CNN, and finally a neural network of three linear layers. This is depicted in Table 3.

Table 3: Q-network architecture

Layer type	Activation	Output shape	Stride	Kernel
Input		$11 \times 13 \times 15$		
Conv2D	ReLU	$32 \times 11 \times 13$	1	(3×3)
Conv2D	ReLU	$64 \times 9 \times 11$	1	(3×3)
Conv2D	ReLU	$32 \times 7 \times 9$	1	(3×3)
Flatten		2016		
Concat		2020		
Linear	ReLU	64		
Linear	ReLU	64		
Linear		5		

D Results of *n*-steps returns with VDN

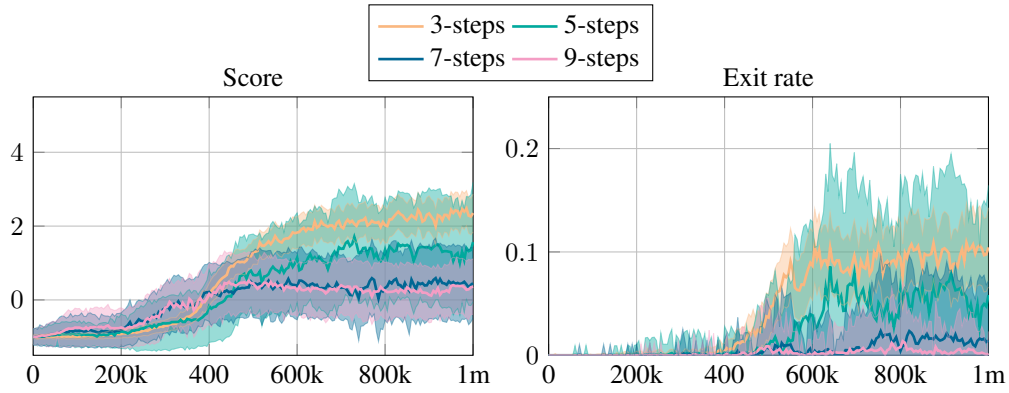
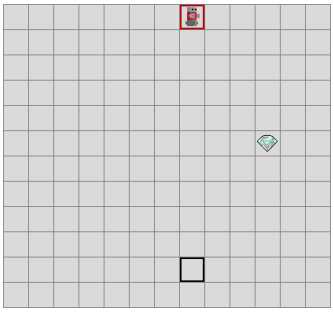
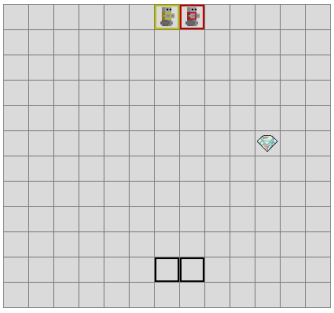


Figure 6: Scores and exit rates for different values of n in n-steps return. Results are shown with the mean in bold $\pm$ standard deviation, capped by minimum and maximum.

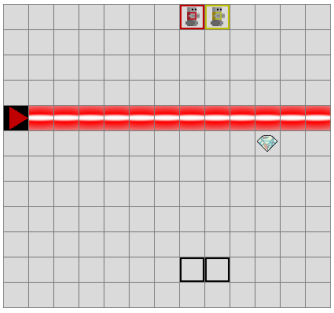
E Maps provided by LLE



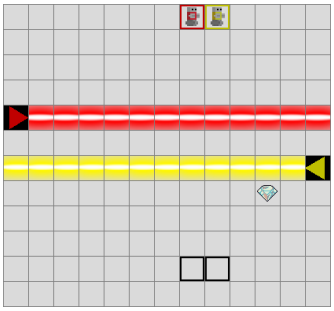
(a) Level 1 for debugging purposes.



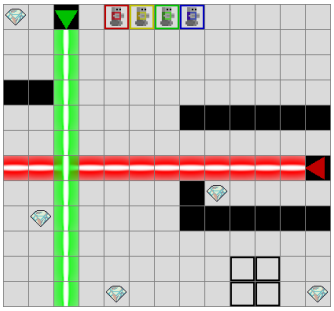
(b) Level 2, first step into multi-agent problems, almost no interdependence.



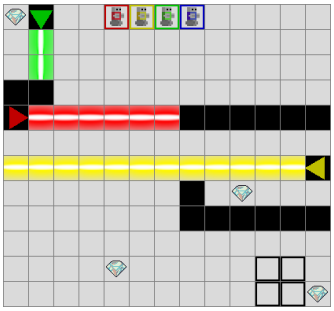
(c) Level 3, introduces the dynamic of laser-blocking, which increases interdependence.



(d) Level 4 introduces two lasers such that each agent successively has to block it for the other for even more interdependence.



(e) Level 5 also has 2 lasers but has 4 agents, which increases interdependence.



(f) Level 6, four agents and three lasers, the most difficult level introduced with the highest level of interdependence.

Figure 7: Six standard levels of LLE. The levels have been designed with incremental level of interdependence in mind. Level 6 is the main level studied in this work.

F Results on standard maps

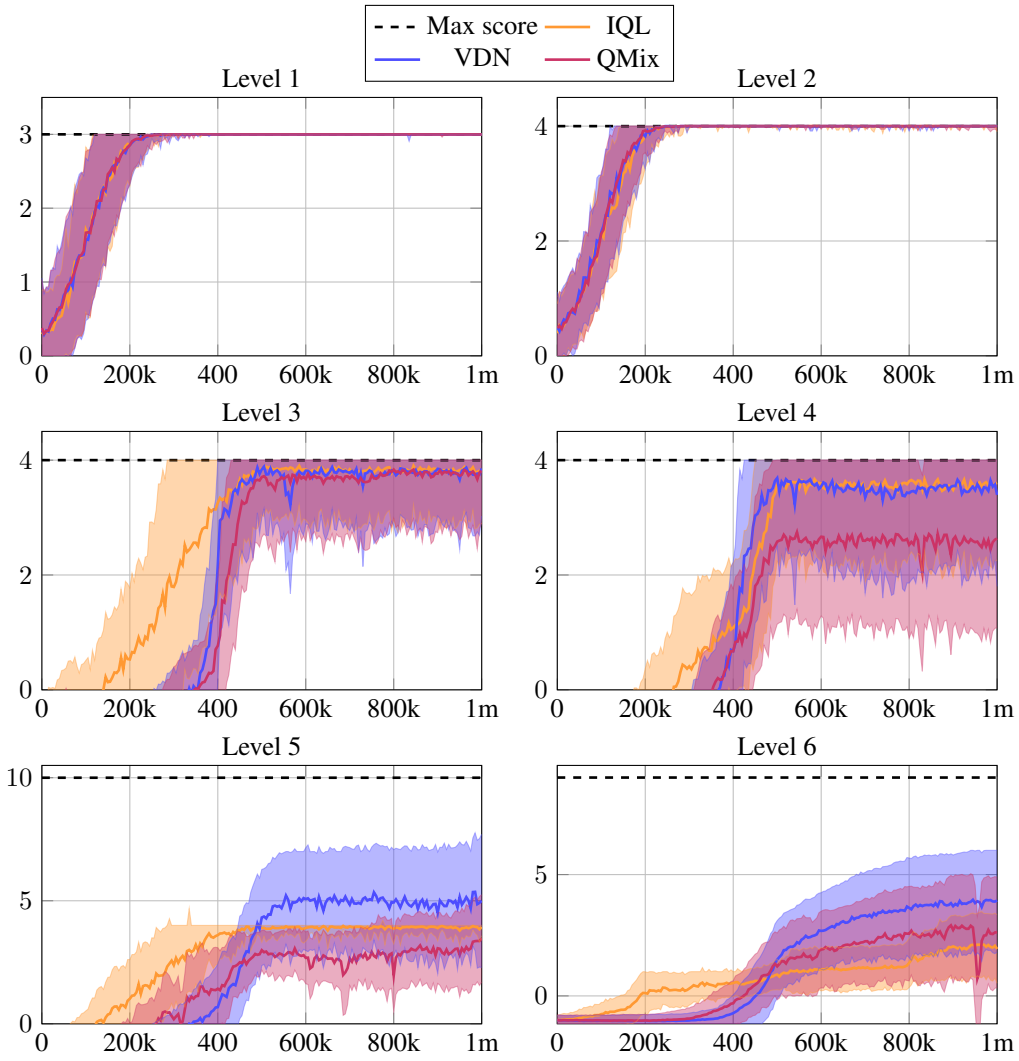


Figure 8: Scores on standard maps over training time. Maximum score achievable is shown as a black dotted line. These results show the mean in bold $\pm$ the standard deviation, capped by minimum and maximum.