Technical University of Denmark

Logical Theories for Uncertainty and Learning (02287)

# Epistemic Planning in Multi-Agent Reinforcement Learning

## Project Report

*Authors:*

Elle McFarlane (s222376)
Jonathan Mikler (s222962)
Anton Jørgensen (s194268)
Dipendra Bahadur Chand (s230006)

December 6, 2023

# Contents

# 1 Introduction

Multi-agent planning problems have a run-time directly proportional to the number of agents and the size of the problem instance, making them challenging to solve with *not-infinite* time or computational resources. *Traditional* approaches to transform the initial state into the goal state (that is, come up with a sequence of joint state-modifying actions), comes in the form of graph search algorithms, usually smart search Policy to find the *"path"* across the state-space. The challenges mainly concerned in this approaches are scale (bigger domains mean bigger state-space), stability (actually finding a solution, if one exists) and efficiency (coming as close to the optimal solution as possible). In this traditional approach, the entire planning strategy is, however clever, hard-coded. Aiming to simplify the problem by reducing the search space somehow, comes the proposal of solving a*simplified* version of the problem, where each agent (or a subset of agents) plans individually, and then some plan-merging strategy produces the whole solution. This implies that each individual agent operates on a *partially observable* world [RN10] with *local* information, and thus might produce an plan that is incompatible with those of other agents.

A different approach is learning the Policy; creating some kind of *state-transition* function which given some information about the (local) current state, the goal and perhaps some global information, provides action to transform the state into another closer to the goal. One take of this approach is using reinforcement learning (RL) to *deduce* (learn) this Policy with training on many (typically $1e^6 \leq$) iterations of problem instances (*"episodes"* in RL lingo) continuously improving the Policy, which takes the shape of a neural net which weights get updated from the performance of it in the episode. Broadly, we can define this Policy as $Q : S \times K \to A$, where the state $S$ and the additional information $K$ are fed into the Policy, and $A$ is the action most suited to be carried by the agent.

The decision on what information to feed the Policy comes from a different place than the challenges of the graph-search approaches; The assumption that all agents have access to global information all the time does not hold in real world applications, where changes in the environment or information about other agents' actions might also not be available. From the need to plan and operate with *partial-* or *no observability*, comes the challenge of training this Policy learning neural nets with partial information.

Modeling knowledge (or lack thereof) of agents and how this knowledge gets updated is the motivation behind epistemic modal logic. Perspective knowledge is modelled using local epistemic states and changes in this states are modelled with epistemic actions, which offer an interesting proposal to model the available information to be fed at a certain point to the *decision making* Policy $Q$.

This works aims to reproduce the results obtained by [Wal+23], which attempted to include what they call *epistemic priors* into the information bundle fed to the Policy in a partially observable state of an agent (local state in epistemic logic terms). In order to maintain the assumption of partial observability, such epistemic knowledge must be available in the local state from the beginning. Chapter 3 aims to present an epistemic formulation of the problem, presenting the epistemic state for each individual agent in the problem and how it evolves through time by epistemic actions carried out by her, others or some *"higher power"*. Chapter 4 formalizes the framework of the reinforcement learning implementation, then 5 present our results and our interpretation of them.

# 2 Problem Definition

Figure 1 displays an instance of the MA planning problem. Solid and dotted circles indicate starting and goal positions. The problem is, as in [Low+17], a cooperative navigation problem, except that the environment is only partially observable and the actions stochastic. DEL-based planning provides tools to solve such a problem, but we will later introduce variations that are not suited for the purely symbolic approach.

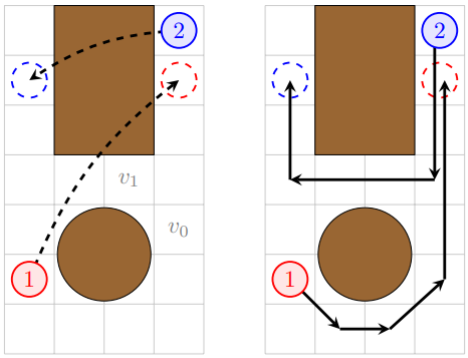



Figure 1: (a) An example instance of a cooperative navigation problem and (b) a solution for it.[Wal+23]

# 3 Dynamic epistemic model - or how do the agents update their knowledge or beliefs through time

Using the resources provided the the dynamic epistemic logic framework [Bol17] we can model an agent's local perspective for any given time, in which we encapsulate the knowledge available to her at that moment. using both *epistemic (local) states* and *epistemic actions*, a way to update the knowledge model for the agent, the evolution through a time line is modelled.

## 3.1 Epistemic (local) states

**Toy example** Consider agent 1's internal perspective in the context of the scenario presented in figure 1. If we consider a case were an agent knows **only** information about herself, her local epistemic state, the information available then would be her location in the world and whether she is at her goal or not. Naming the agent $i$, we would represent the prepositions "agent $i$ is at location $l$" and "agent $i$ is at her goal" with $@^i$ and $g^i$ respectively. Furthermore, we could represent the agent's initial local epistemic state $s_0^i$ as:

$$\bullet$$
$$@_0^1, \neg g^i$$

If we considered the case where the agent indeed has access some the location and goal of the other agent (agent 2) her local state would look like this (provided that the 2nd agent does not start at her goal location):

$$\bullet$$
$$@_0^1, \neg g^2, @_0^2, \neg g^j$$

If the information on the location of another agent is not definite, the local state could encode instead beliefs about other agents' position, provided that this knowledge of belief was made available somehow. Such a state where possible location of other agents are considered would look like this, for $N$ different consideration of the location of agent 2:

$$w1 : @^1 \wedge @_1^2 \qquad w_n @^1 \wedge @_n^2 \qquad w_N @^1 \wedge @_N^2$$
$$\bullet \text{---}_i \cdots \text{---}_i \bullet \text{---}_i \cdots \text{---}_i \bullet$$

This set of possible worlds, are indistinguishable from one another from agent $i$'s perspective, hence the accessibility relation between all of them. If no more information was provided, all of this

worlds would be equally plausible. If however, more information is made available, the plausibility distribution might change favoring some world(s) specially.

### 3.2 Epistemic state update through actions

Commonly referred to as *epistemic actions* are model-changing operators in accordance to the agents interaction with the world. In [Bol17] a formal definition is presented, along the mechanics of how they update a given state using the so-called *product update* ($\otimes$) operator. In short, given a state $s_0$ and an action $a$, the updated state $s_1$ is $s_0 \otimes a$. The way [Bol17] presents an action is as such:

$$a = \quad e1 :< pre, post > \quad \overset{i}{\bullet}\text{- - - - - - -} \cdots \text{- - - - - - -}\overset{i}{\bullet}\text{- - - - - - -} \cdots \text{- - - - - - -}\overset{i}{\bullet} \quad en :< pre, post > \quad eN :< pre, post >$$

Where $e$ is an event, which can be thought of as different configuration of the action. Any given state $s$, updated by this the application of an action would result in a new epistemic state, where the worlds would be those coming from the application of each individual event on each individual world (provided the *pre* condition was met in the original world). For a detailed example on this dynamic, please refer to [Bol17]. A description of the actions relevant for the model of the problem at hand is presented hereunder.

[Bol17] offers a classification for the different types of action. Relevant for this work are the *ontic* and *purely epistemic* actions, which are opposed to one another in the way they affect an agent's local epistemic state. *Purely epistemic* actions are those whose post condition can be implied directly from the precondition, while ontic actions add information to the epistemic state. An example of an purely epistemic action, relevant for this work is the so called *sensing* action, in which for the local state of an agent, if another agent should be within the "line of sight" (in a partially observable configuration), then the position of that other agent in that time step is now known by the agent in question. Later in the section, we will present an epistemic model for the problem at hand and the relevance of the actions mentioned here will become clearer.

The knowledge agent 1 has access to can change through time, and may or may not influence her decisions on what action to take herself. Even if she decided not to make any movement herself, her knowledge (or belief in the case of partial or no observability) can be updated by different kind of actions. While the *action* term can be interpreted as an *"active"* intentional move from the agent, *epistemic* actions are both action the agent could execute, actions other agent could execute that alter the agent's knowledge or belief, as well as public announcements; actions that do not alter reality, but might update the knowledge of an agent.

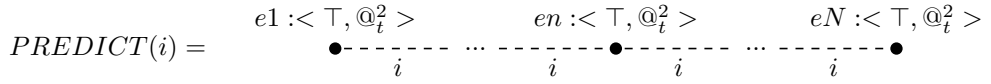
In the context of the multi-agent planning problem, the state-updating actions for any given agent are: $GO$, $SENSE$, and $PA$ (Public Announcement)[1].

**GO** The state for any given agent at a given time $t$, gets updated at least by two actions, one of them is *Go*, which might alter the location of the agent in the grid (up, down, left, right or No-Operation). The agent decides which event action to take based on some internal decision process, so that the state update does not expand into multiple possible world in the next time step, but changes into a single one with the updated location:

$$GO(i) = \quad e_{up} :< \top, up(@_t^i) > \quad e_{left} :< \top, left(@_t^2 >) \quad e_{no-op} :< \top, no-op(@_t^2 >)$$
$$\bullet \qquad \bullet \qquad \bullet \qquad \bullet \qquad \bullet$$
$$e_{down} :< \top, down(@_t^i) > \quad e_{right} :< \top, right(@_t^2 >)$$

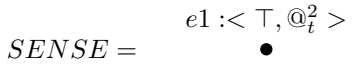
**PREDICT** The state for any given agent at a given time $t$, gets updated at least by two actions, the first one is GO, and the second one being $PREDICT$; an *purely-epistemic* action where the agent considers worlds where one of the possible GO actions (down, up, etc), changing her location. A representation of the local action, from the perspective of the agent *predicting*, would look as such:

[1] This definitions are very informal. A more thorough definition would be similar to this ones here following [Bol17]

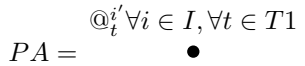



The dotted line indicate that agent $i$ has to consider all possibilities, at least until some indication of which one would be more likely.

**SENSE** The *SENSE* action models the event where another agent comes within the *line of sight* of the agent in question. In an agent $i$'s local state, if agent $j$ is seen by him, then $j$'s location is now a known fact. This action then looks like this:



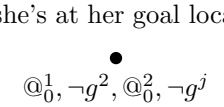

**PUBLIC ANNOUNCEMENT** The only *PA* action in the scope of the problem, is the one coming from the mentioned *Convention of Operation.* as such: all agents know each others starting positions, i.e. $k_i @_0^i$ for every $i \in I$ where $I$ is the agent set. Additionally, they also know each others goal destinations, but more precisely, each agent knows how other agents would act to get their respective goals in a *conventional* way. The prepositions that would follow from this come from a state transition model $C : S \times G- > A$, where $A$ would be a list of actions, out of which the *estimated* location at each time-step of another agent can be devised, based on the actions that would transform the given state $S_0$ into the state $S_t$. This information would all come from the knowledge given to the agents *prior* to run-time (at t=0), and hopefully help in the decision at each time step of what action to take, provided that there is no more information available. Formally the public announcement must be a set of prepositions that will hold true throughout run-time:




The apostrophe denotes estimation, in contrast to the real location of a given agent, which is not known for another given agent for $t \neq 0$.

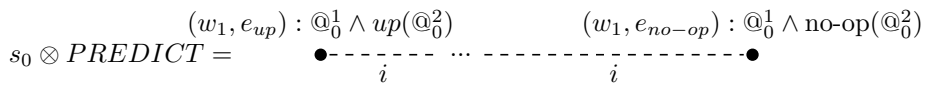
## 3.3 Decision making with available knowledge

We will now use the different ideas presented about dynamic epistemic logic to model the way an agent's knowledge evolves through time. At initial time ($t_0$), the local state for agent $i$ has the knowledge about her location, whether she's at her goal location, and similarly for the second agent:




the transition from time-step 0 to time-step 1 involves the serial product operation of the state with some actions. Before delving into the actions, it is important to remember that the agent has a goal she needs to reach, and therefore her decisions are oriented into reaching that goal. These decisions will be based on whatever information she has at the moment of deciding, which is the last of the actions taken before moving onto the next time-step.

The first action that updates the state is the *PREDICT* action, in which the state expands to have worlds with all the possible locations of the second agent, based on the possible *GO* actions she can perform:



The notations $up(@_0^2)$, $donw(@_0^2)$, etc. can be replaced by the common $@_1^{2'}$ since in each world they denote the same preposition. It is worth noting that all of this worlds are indistinguishable for agent 1 since she does not know the real location of agent 2, and their are all equally plausible.

The plausibility of each world is then adjusted by the second action in the transition, which comes in the shape of the public announcement preposition relevant for $t_1$, i.e. $@_1^{2'}$. The result of product updating $s_o \otimes PREDICT \otimes PA(1)$ is then:

$$s_0 \otimes Pred \otimes PA(1) = \overset{(w_1, e_{up}, pa_1) : @_0^1 \wedge @_1^2 \wedge @_1^{2'}}{\bullet} \overset{i}{\text{- - - - - - - - - - - - - -}} \cdots \overset{i}{\text{- - - - - - - - - - - - - -}} \overset{(w_1, e_{no-op}, pa_1) : @_0^1 \wedge @_1^2 \wedge @_1^{2'}}{\bullet}$$

This update in the local state comes with an additional. It updates not only the information but also the accessibility relationships by modifying the plausibility of the different worlds. The world where the *conventional* location and the predicted location are the same, will be selected as the one with the maximum plausibility, while the others will be assigned an equally corresponding low values, and therefore being selected to be the world that will continue to be updated towards $t_1$. We can reduce the state transformed so far, since the agent is keeping only the world with the highest plausibility:

$$s_0 \otimes Pred \otimes PA(1) = \overset{max_{plaus}(w_1, e_j, pa_1) : @_0^1 \wedge @_1^{2'}}{\bullet}$$

Lastly, comes the moment for the agent to decide her next move. In this section we will refer to this state-transition model as $Q : S\times \to A$ where $A$ is a specific GO action applicable by the agent. For clarity, we denote the partial transition state we have described so far as $s_{2/3} := s_0 \otimes Pred \otimes PA(1)$

$$s_1 = s_{2/3} \otimes Q(s_{2/3} =, @_g^2) \quad \overset{(max_{plaus}(w_1, e_j, pa_1), e_0^1) : @_1^1 \wedge @_1^{2'}}{\bullet}$$

# 4 Reinforcement Learning with Epistemic Priors

This section covers the limitations of DEL based planning as described above, and introduces alternative techniques for solving problems that are especially impacted by said limitations. Specifically, multi agent reinforcement learning with epistemic priors (MARL-EP) will be introduced as a way to compute high performing solutions to problems where the state space is too large for the search algorithms in DEL based planning, e.g. continuous state spaces.

## 4.1 Limitations of DEL based planning

The DEL framework presented in section 3 is a powerful decision making tool, allowing the space for representing state-transition systems to grow linearly with the number of propositions instead of exponentially. Some state spaces are however still too large for this approach, namely the continuous state spaces. Interestingly, they are not too large because of the number of propositions, but because of how the propositions must be defined. Consider an example single agent problem, taking place on the real line $\mathbb{R}$. The agent starts at position $p = 2$, and has goal state $p = 0$. The available actions are to move by adding a number $x \in [-1, 1]$ to the current position, after which a small random number $\epsilon \in \mathbb{R}$ is added as well. Clearly, the goal can never be reached in a finite amount of steps, and any attempt at adding non-trivial preconditions to actions leads to an uncountable number of propositions, except if we add propositions like $|p| < 0.01$, with which we effectively discretise the state space again. To address these environments we instead turn to methods that thrive on continuousness and differentiable functions.

## 4.2 Multi Agent Reinforcement Learning

The objective of multi agent reinforcement learning (MARL) is a preferably optimal solution to the MA planning problem as seen in Figure 1. In the DEL approach, such a solution would be found by applying a graph-search algorithm to the space of epistemic states induced by the epistemic model. In MARL, such a solution is a set of policies, one for each agent, $\pi_i$. The individual policies, $\pi_i(a_i|s_i, o_i)$, are the probabilities of choosing action $a_i$ given the current state $s_i$ and observation $o_i$. A number of approaches exist to compute these policies, but traditionally do not include the ability for agents to reason about the knowledge and behavior of other agents. Either they provide global observations to a central entity at training time, making them unsuitable for limited communication tasks, or they only condition $\pi_i$ on local state and observation, leading to diminished large-scale cooperation since large sets of global states with identical local states will be indistinguishable. One approach to alleviating these issues is to condition the policies on one additional feature, epistemic priors.

## 4.3 Introducing Epistemic Priors

Epistemic priors $e_i$ make it possible for agents to use their knowledge of other agents goals, as well as their knowledge of the knowledge of other agents, etc. This works by conditioning the policies on not just states and observations, but also the epistemic priors: $\pi_i(a_i|s_i, o_i, e_i)$. This enriches the local observations to more accurately estimate global state, leading to better cooperation.

To estimate $e_i$, it is assumed that all agents share common knowledge of a predetermined convention of operation. A convention of operation can be seen as a description of how agents are expected to behave, e.g. in Figure 1, since the detour around the brown circle is larger for Agent 2 than for Agent 1, the convention could be that the detours are taken by the least inconvenienced agent. This convention would be common knowledge, so no agent needs to plan for scenarios where the others do not know of it, though they might not follow it. Viewing it as a function on the state space, a convention of operation $C$ must be comprehensive, covering all states, deterministic, outputting the same action for the same input every time, and chainable, able to produce a trajectory by applying it repeatedly [Wal+23]. Such a convention can be constructed manually, but is in practice instantiated with a deterministic multi agent planner or RL policy. If using a policy, care must be taken to set the random seed at the right times to enforce the deterministic property.

## 4.4 Original MARL-EP Paper Results

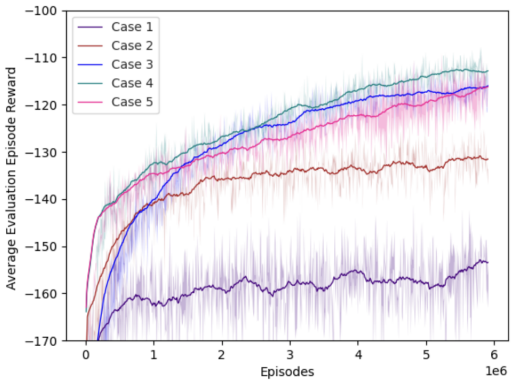



Figure 2: Evaluation of trained QMIX agents for different cases from MARL-EP paper [Wal+23]

The MARL-EP paper compares results between 5 different cases. Case 1 is no sensing, which is the baseline. Here agents do not have access to other agents location. Case 2 is limited sensing, where agents have access to other agents location when close. Case 3 is perfect sensing, meaning

agents always know the location of all other agents. Case 4 is limited sensing, with priors (QMIX-EP). Agents have access to other agents location when close, and use estimated location via priors otherwise. Case 5 is no sensing, with priors (QMIX-EP). This is similar to Case 1, but agents use estimated location of other agents. For a detailed discussion of these results, see [Wal+23].

# 5 Results

## 5.1 Experiment Setup

The authors were not able to open source their code at the time that we discussed with them, but one of them, Jaime Ide, was kind enough to explain what they did high-level and to confirm some low-level assumptions and details before we implemented.

The entire setup is explained in subsection A.1. However, there are differences between the original setup in [Wal+23] and this paper, which are discussed in the following. The original authors do not include the given agent's prior (joint step from epistemic plan) whereas we do. Including the agents own prior is conceptually equal to the agent knowing what it 'should do' according to the convention of operation. The original authors use a sub-optimal deterministic planner (CBS with bipartite reduction [WSF20]), whereas we use a pre-trained QMIX model with random seed set to ensure deterministic outputs. The original authors included the previous action as is normal when learning Q functions; the code we used did not do this by default and we did not see initial improvements for doing it, so we left it out, which means technically our neural nets learned Value functions as defined in reinforcement learning. They trained for 6 million steps whereas we trained for 4 million, and lastly we used a random seed of 1 since theirs was not included in the paper.

## 5.2 Results

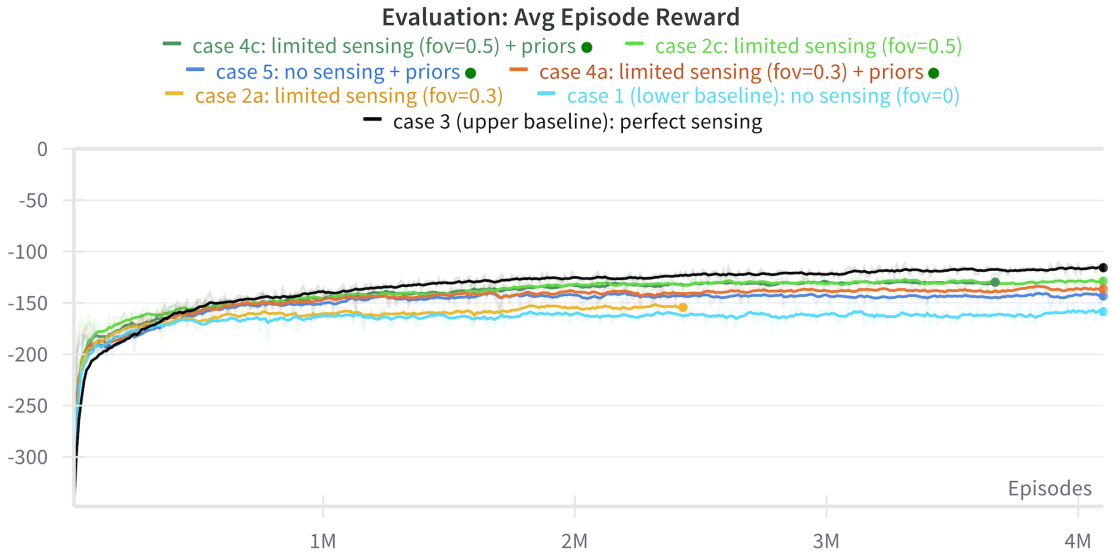



Figure 3: Reproducing MARL-EP Results with Pretrained QMIX as Planner and 4 Million Episodes. Numerical values at the 4 million mark are shown in the appendix in Figure 5

The results in Figure 3 are displayed such that the perfecting sensing case (equivalent to the performance of our pretrained QMIX planner) is black and the other cases are such that: the lighter colors have no priors and the darker colors have priors. For example, the no-sensing case (1) is light blue

and its equivalent priors case (5) is dark blue. For the limited sensing case, we tried field-of-view (FOV) sizes of 0.3 and 0.5

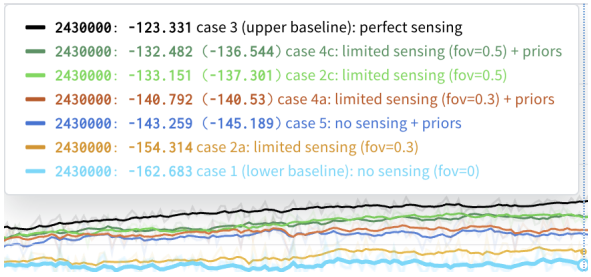



Figure 4: Performance Comparison of Cases at Episode ≈2.4 Million

The training for case 2a was unexpectedly cut short at 2,430,000 episodes (HPC admin interference) and we did not have time to continue training. Therefore, results at the ≈2.4 million mark can be seen in Figure 4.

**Takeaways** We see the best performance for perfect sensing and no priors (case 3), while the worst performance happens with no sensing and no priors (case 1). Generally, as FOV increases, performance also increases. Priors have higher impact on performance for lower FOV.

### 5.3 Discussion

The overall trends in the results align with expectations. Adding priors to an otherwise identical case increases performance, and likewise for increasing the FOV. There are however a few cases that do not behave as expected. First, adding priors when FOV $= 0.5$ does not increase performance at all. This is an extreme example of the general observation that as FOV gets higher, priors are less effective. A possible explanation of this is that as FOV increases, the amount of position estimations done by the priors decreases.

We also see no increase in learning per episode when priors are present, as seen in Figure 6. The authors of the original paper [Wal+23] saw a significant increase in early rate of learning, so the absence of this behavior in our results is surprising. Generally, when observing minor discrepancies between the results in Figure 3 and ours we have been able to chalk it up to differences in initial conditions and random seeds, but for such a noticeable feature to be missing across multiple cases, there must be a systematic reason.

## 6 Future Work

While we did reproduce the results of [Wal+23] to a large degree, there are several points of improvement to be adressed. Had time permitted, we would have averaged our results over multiple runs with different seeds, to improve validity. This could easily be done with our code as a starting point, and would answer whether some of the unexpected observations were simply due to random chance. Next, using the same solver as in [Wal+23] would make the results more one-to-one comparable. Starting from their code, this means integrating a C++ solver with a Python environment. Additionally, policy losses (Figure 9 did not converge to 0 except for the perfect sensing (planner) case, meaning more training will likely improve performance, if only by a marginal amount.

For additional experiments, one could test the relationship between FOV and effectiveness of adding priors, e.g. testing priors vs. no priors on FOV values in $\{0, 0.1, \ldots, 0.9, 1\}$. Our results would suggest an inverse relationship, but are currently unclear. Another approach could be to qualitatively evaluate the difference in behavior of agents with and without priors. Do their solutions become more creative, look more coordinated, or do they simply converge more towards the policy

given by the convention of operation? The latter is not necessarily a negative, as it still leads to increased global coordination.

# A Appendix

## A.1 Experiment setup

**Random seed** 1

**Hyperparameters** No changes from the defaults used in [VY22]

**Provider (planner) of convention of operation (joint-plan)** We trained a QMIX model in MPE Simple Spread environment on 6 million episodes where each episode the agents and landmarks started at random positions. We stopped training when the evaluation average reward was ≈-115, which mirrors the results in [VY22].

**Code & Architecture** We modified the code from [VY22] to include the epistemic priors for the MPE Simple Spread environment with 3 agents, 3 landmarks, RNN-based local Q-networks, and 1 environment (i.e. no parallelization since this is not allowed by the code for reasons unknown to us). The q network, and therefore, policy of each agent is shared by default, which does not work out in all MARL environments, but happened to in this case, likely due to the homogenous roles of the agents (all need to reach some landmark). The modified code is here: https://github.com/ellemcfarlane/logical_theories_marl [McF23]

**MARL-EP input space**

**Baselines (no-priors)** For 3 agents and 3 landmarks, the local observation space has 18 dimensions:

```
[
   current_agent_velocity,
   current_agent_position,
   all_landmark_positions,
   other_agent_positions,
   other_agents_communications
]
```

e.g., for agent 1 with 3 total agents and 3 landmarks:

```
[
    (agent1 vel x, agent1 vel y),
    (agent1 pos x, agent1 pos y),
    (landmark1 pos x, landmark1 pos y),
    (landmark2 pos x, landmark2 pos y),
    (landmark3 pos x, landmark3 pos y),
    (agent2 pos x, agent2 posy),
    (agent3 pos x, agent3 pos y),
    (agent2 comm1, agent2 comm2),
    (agent3 comm1, agent3 comm2)
]
```

Note: in the spread version of MPE, communication did not seem to be used as the values were always 0, so these were simply placeholder values that the agents ignored.

**Limited or no sensing** When other agents were outside of the current agent's field-of-view (FOV), the positions of the other agents were replaced with placeholder 0's.

**With priors** We appended the priors (relative positions of the other agents, including the given agent) to the original observation increasing dimensionaltiy by 2*n_agents (e.g. 6 in the 3-agent case for total of 24-dimensional local observations)

**Planner inputs** The same as the starting observation except that in the no/limited sensing case, the planner was always given the full observation (i.e. no placeholder values for agents outside of the FOV) to calculate the "ideal" plan

**Hardware** 1 GPU per experiment case: Tesla V100-SXM2-32GB from DTU's HPC; ≈ 1 hour per 1 million episodes in epistemic cases

## A.2 case values at episode 4 million

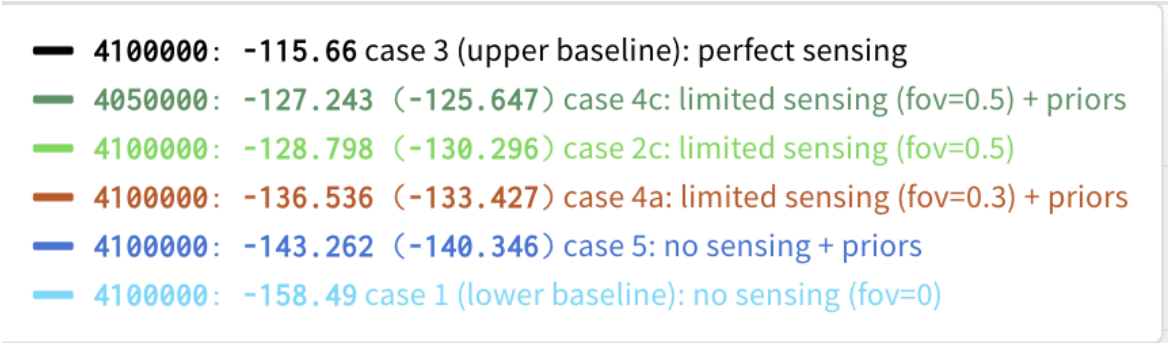



Figure 5: Final Results (minus case 2a)

## A.3 Learning rate increases for priors vs non-priors case

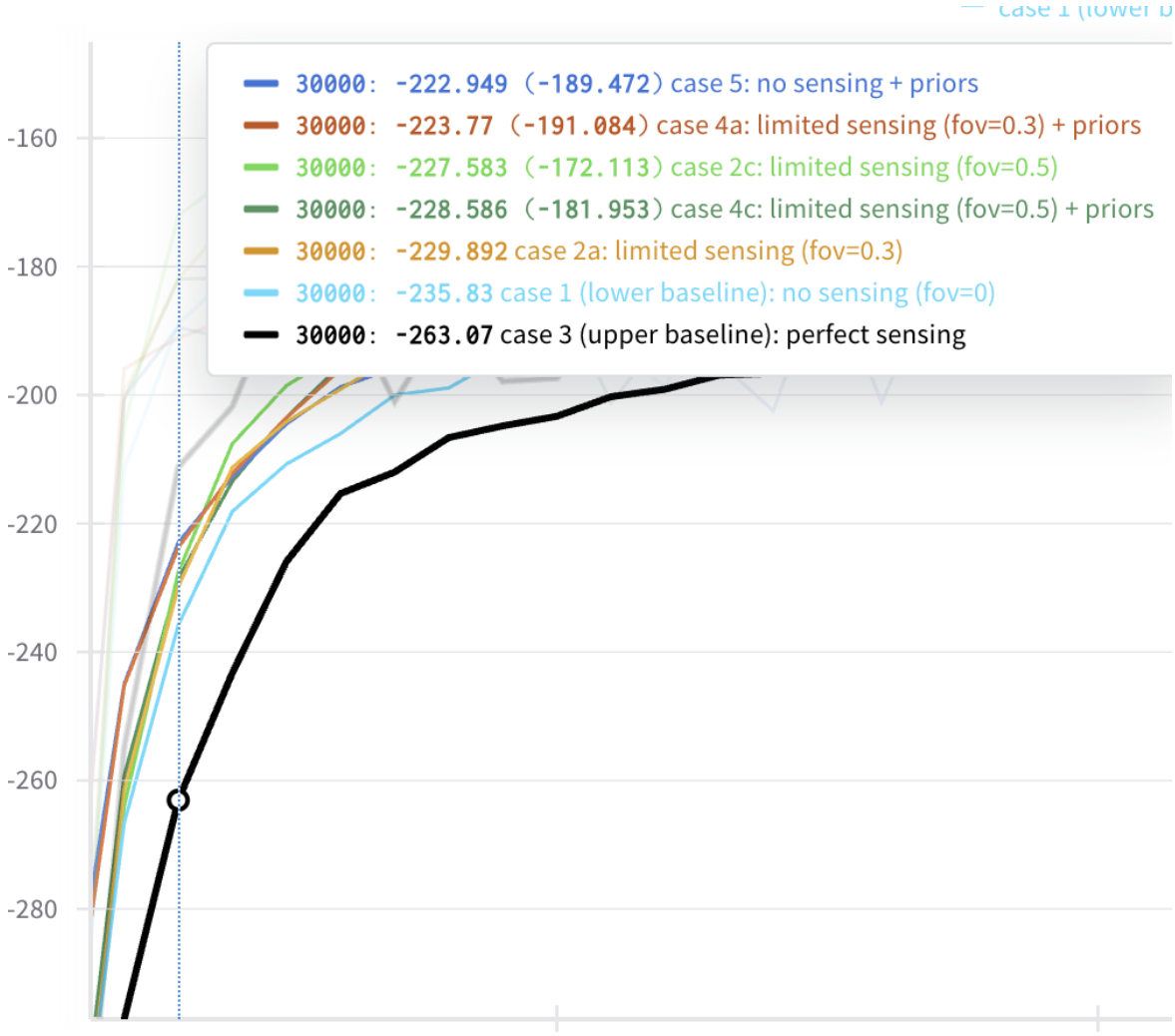



Figure 6: No Learning rate increases for priors vs non-priors case

All cases appeared to have a better learning start than the perfect sensing case and there were no noticeable differences between the priors and no-priors cases.

## A.4 Training Avg Rewards

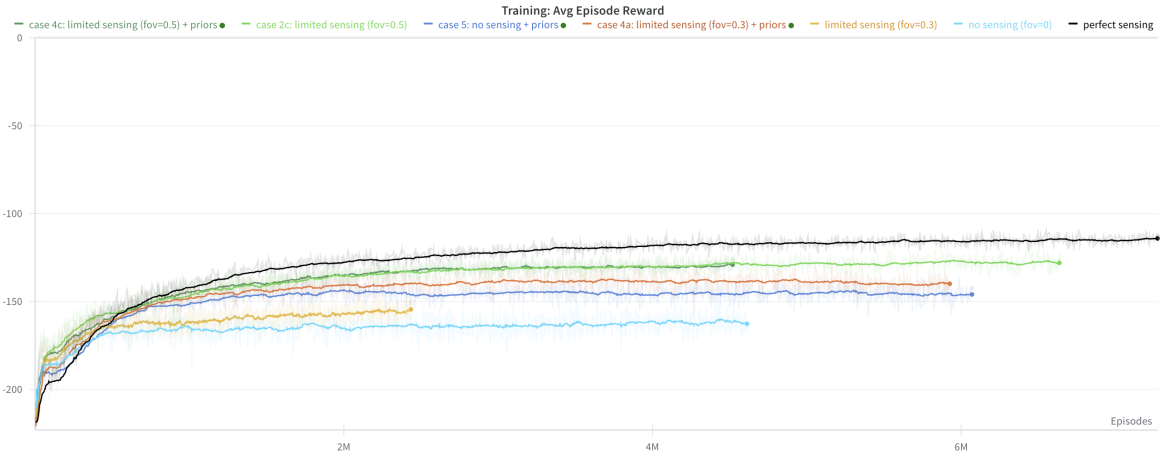



Figure 7: Training results

Note: as can be seen here, the different cases were trained for varying number of episodes, so for simplicity in the report, we just reported the 4 million evaluation mark for all cases (except 2a).

## A.5 Evaluation Rewards for Planner's Paths (Epistemic Plans)

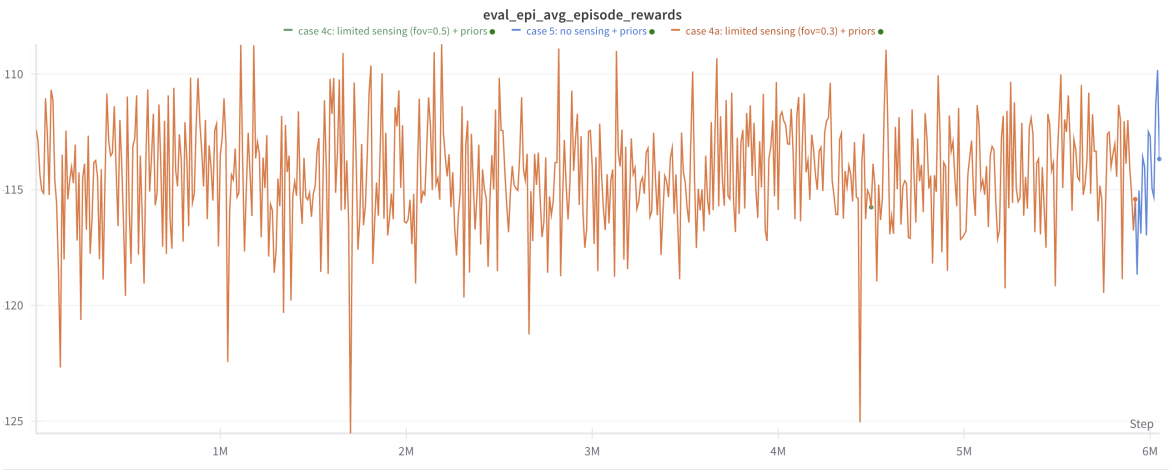



Figure 8: No Learning rate increases for priors vs non-priors case

We tracked the rewards of the planner to ensure it always solved the scenarios reasonably well (i.e. within a given reward range). Note the values exactly overlap in all cases since every case used the same random seed (i.e. same set of random scenarios) and the planner is deterministic. The blue shows an area where case 5 and case 4a have data but case 4c did not continue training to this point.

## A.6 Policy Losses

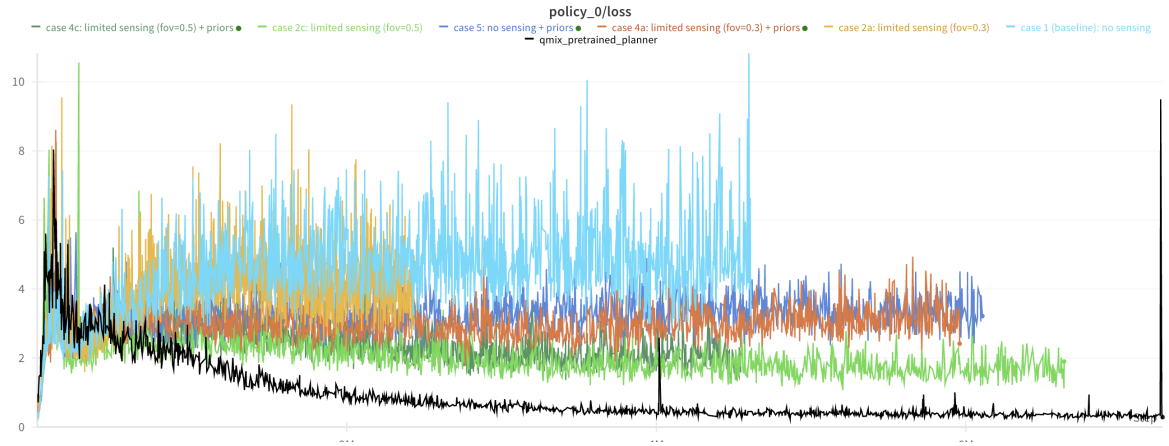



Figure 9: Policy losses

Note that the most stable loss and the one that converged closest to 0 is the pretrained planner (perfect sensing) case.

# References


[Bol17] Thomas Bolander. "A Gentle Introduction to Epistemic Planning: The DEL Approach". In: *Electronic Proceedings in Theoretical Computer Science* 243 (Mar. 2017), pp. 1–22. ISSN: 2075-2180. DOI: 10.4204/eptcs.243.1. URL: http://dx.doi.org/10.4204/EPTCS.243.1.

[Low+17] Ryan Lowe et al. "Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/68a9750337a418a86fe06c1991a1d64c-Paper.pdf.

[McF23] Elle McFarlane. *MARL-EP*. https://github.com/ellemcfarlane/marl_ep. 2023.

[RN10] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. 3rd ed. Prentice Hall, 2010.

[VY22] Akash Velu and Chao Yu. *Off-Policy Multi-Agent Reinforcement Learning (MARL) Algorithms*. https://github.com/marlbenchmark/off-policy. 2022.

[Wal+23] Thayne T. Walker et al. "Multi-Agent Reinforcement Learning with Epistemic Priors". In: *PRL Workshop Series – Bridging the Gap Between AI Planning and Reinforcement Learning*. 2023. URL: https://openreview.net/forum?id=5cWF3p2jDi.

[WSF20] Thayne T. Walker, Nathan R. Sturtevant, and Ariel Felner. "Generalized and Sub-Optimal Bipartite Constraints for Conflict-Based Search". In: *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 2020, pp. 7277–7284. DOI: 10.1609/AAAI.V34I05.6219. URL: https://doi.org/10.1609/aaai.v34i05.6219.