MANSA: LEARNING FAST AND SLOW IN MULTI-AGENT SYSTEMS WITH A GLOBAL SWITCHING AGENT

Anonymous authors
Paper under double-blind review

ABSTRACT

In multi-agent systems, independent learners (IL) often show remarkable performance and easily scale with the number of agents. Yet, training IL can sometimes be inefficient particularly in states that require coordinated exploration. Using observations of other agents' actions through centralised learning (CL) enables agents to quickly learn how to coordinate their behaviour but employing CL at all states is prohibitively expensive in many real-world applications. Besides, applying CL often needs strong representational constraints (such as individual-global-max condition) that can lead to poor performance if violated. In this paper, we introduce a novel IL framework named **M**ulti-**A**gent **N**etwork **S**election **A**lgorithm (MANSA) that selectively employs CL only at states that require coordination. Central to MANSA is the additional reinforcement learning (RL) agent that uses *switching controls* to quickly learn when and where to activate CL so as to boost the performance (and using CL only where necessary) while using only IL everywhere else. Our theory proves MANSA, which can seamlessly adopt any existing multi-agent RL (MARL) algorithms, preserves MARL convergence properties in cooperative settings. We prove MANSA can improve performance and maximise performance given a limited budget of CL calls. We show empirically in Level-based Foraging and StarCraft Multi-agent Challenge that MANSA achieves fast, superior training performance through its minimal selective use of CL.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has emerged as a powerful tool to enable autonomous agents to solve various tasks such as ride-sharing (Zhou et al., 2020) and swarm robotics (Mguni et al., 2018). Among MARL methods are a class of algorithms known as independent learners (IL) e.g. independent Q learning (Tan, 1993).

IL decomposes a MARL problem with N agents into N decentralised single-agent problems. In this way, each agent treats other agents as part of the environment which provides a straightforward method of performing decentralised learning. Since the agents ignore other agents, IL can induce quick training since each agent's training of its policy is contingent on its local observations and own actions and not the actions of others. This approach is efficient in (sub-)systems in which the interaction between agents is weak (Kok & Vlassis, 2004). However, ignoring other agents' influence on the system results in the loss of RL convergence guarantees and may produce in oscillatory learning behaviour since from the agent's perspective, the environment can appear non-stationary. Secondly, IL is unable to distinguish randonmess due to systemic stochasticity and that produced by other learning agents' exploration further hindering learning in some environments. An issue is that in multi-agent systems (MAS), agents are typically required to coordinate to solve the task. This presents a difficulty for IL since random occurrences of successful coordination are improbable without observations of other agents. These limitations mean IL can struggle to tackle scenarios where coordination is required (Hernandez-Leal et al., 2017).

On the other hand, MARL learners are often trained in simulated environments. As such, MARL agents can be provided with other agents' observations and other state information during training. With this added information, agents can condition their policies on other agents' actions which eradicates the appearance of non-stationarity.

Centralised training and decentralised execution (CT-DE) (Kraemer & Banerjee, 2016; Foerster et al., 2018) is a framework that uses a centralised critic for training while performing execution in a decentralised fashion. This framework has become a central MARL paradigm and is the basis of popular methods such as QMIX (Rashid et al., 2018), SPOT-AC (Mguni et al., 2021a) and COMA (Foerster et al., 2018). Various studies have conjectured that CT-DE can speed up training by fostering cooperative behaviour and stabilising training. This is useful when there exists a tight coupling between agent interactions which necessitates global observations during training (Sharma et al., 2021). Despite these apparent advantages, CT-DE suffers from an explosive growth in computational burden since the joint action-state space grows exponentially with the number of agents (Yang et al., 2020). As a consequence, CT-DE methods require large numbers of samples to complete training. In (sub-)systems in which a tight coupling between agent interactions does not exist everywhere, centralisation can introduce a computational burden without providing advantages (Kok & Vlassis, 2004) (e.g. Fig. 1). To mitigate the explosive growth in complexity and enable CT-DE to scale, various CT-DE algorithms such as QMIX (Rashid et al., 2018), VDN Sunehag et al. (2017) decompose the joint value function into factors that depend only on individual agents. The representational constraints needed to achieve such decompositions can lead to provably poor exploration and suboptimality (Mahajan et al., 2019). For example, QMIX requires a monotonicity constraint which can produce suboptimal value approximation.

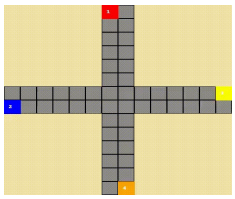

Figure 1: In this driving scenario, the agents (shown as 1 - 4) only interact at the intersection and need only coordinate there. Prior to arrival at the intersection, their actions do not affect other agents.

To tackle these issues, we introduce a general MARL framework, MANSA which optimally selects when to call on centralised learners to boost training using IL. MANSA involves a decentralised critic network Decentral, a centralised critic network, Central and, an *adaptive* RL agent Global. Global determines at which states to activate Central while the decentralised network Decentral is used at all other states. A key feature of MANSA is the novel combination of RL and *switching controls* (Mguni, 2018) which enables Global's to quickly tackle its task while the two networks are in training concurrently. This enables the benefits of both algorithm classes to be leveraged while overcoming some of the issues presented by any one algorithm class.

An integral component of MANSA is a novel combination of RL and switching controls (Mguni, 2018). This enables Global to determine useful states to learn to activate CL (for example the states in which the agents are required to coordinate their actions) and minimise unnecessary CL calls during training. This is in contrast to current MARL methods that use solely either CL or IL at all states throughout training. The binary decision space for determining whether CL and IL should be activated means that the Global agent can quickly determine the states where CL is beneficial while the MARL agents learn.

Overall, MANSA has several advantages:
• By switching to a centralised critic only at the set of states in which CL is required while leveraging the benefits of IL, MANSA increases the computational efficiency of CT-DE methods.
• MANSA activates CL when (and only when) required resulting in MANSA boosting IL performance and enabling IL to tackle tasks which using IL would otherwise lead to coordination failures.
• MANSA minimises the number of times that CL is called (and hence the global information is used during training) while either matching or improving the performance of fully CT methods. Additionally, the MANSA framework allows for a fixed budget for calls of CL.
• MANSA is a plug & play framework which seamlessly adopts any MARL algorithm.

To enable the framework to perform successfully, we tackle several challenges. Firstly including a new agent, Global that learns while the N MARL agents are training can occasion convergence issues. Secondly, unlike standard RL, Global's learning process uses a form of RL policy known as

switching controls. We prove that MANSA, which now includes Global induces a learning process that converges and preserves the MARL learners' convergence properties.

	Centralised Training	*(Fully) Decentralised Training*
Convergence guarantees	Yes	No
Scalability	Combinatorial (requires representational constraints)	Constant
Required observation inputs	Global	Local

2 Related Work

A key principle in the CT-DE framework is the Individual-Global-Max (IGM) principle (Son et al., 2019) which ensures that CT-DE learning process generates policies that are consistent with the desired system goal. In order to realise the IGM principle in the CT-DE framework, QMIX and VDN propose two sufficient conditions of IGM to factorize the joint action-value function. Crucially however, such decompositions and limited by the joint action-value function class they can represent and can perform badly is systems that do not adhere to these conditions (Wang et al., 2020).

Several methods have been proposed to address this structural limitation. QPLEX (Wang et al., 2020) uses a dueling network architecture to factor the joint action-value function avoiding representational restrictions. Nevertheless, QPLEX has been shown to fail in simple tasks with non-monotonic value functions (Rashid et al., 2020). QTRAN (Son et al., 2019) formulates the MARL problem as a constrained optimisation problem with L2 penalties for decentralisation. Nevertheless, QTRAN has been shown to scale poorly in complex MARL tasks such as SMAC (Peng et al., 2020). WQMIX (Rashid et al., 2020) considers a weighted projection which is weighted towards better performing joint actions. At the core of these techniques are heuristics that do not guarantee the IGM consistency. Consequently, achieving full expressiveness of the IGM function class with scalability remains an open challenge for MARL.

Actor-critic methods such as COMA (Foerster et al., 2018) and MADDPG (Lowe et al., 2017) are popular methods within MARL. These methods involve a centralised critic but nonetheless do not impose restrictions to represent the joint-action value function. Nevertheless, these methods are significantly outperformed by value based methods such as QMIX on standard MARL benchmarks e.g. StarCraft Multi-Agent Challenge (SMAC) (Peng et al., 2020). MAPPO (Yu et al., 2021) which is a leading actor-critic method with a centralised value function, extends a popular single-agent RL method, Proximal Policy Optimization (Schulman et al., 2017) to MARL. Nevertheless, MAPPO has been shown in (de Witt et al., 2020) to be outperformed by single agent learners (and QMIX), specifically PPO in some tasks while requiring modest hyperparameter tuning.

Several papers have explored the issue of exploiting localility of the agents' interactions in different ways. Early works such as (Kok & Vlassis, 2004) tackle the problem in learning in systems with sparse subregions. Such works make stringent assumptions which require the global coordination requirements of the system to be known beforehand. Moreover, other works centred on detecting where in the state space global or extra information is required to obtain a good policy. These works take the approach of detecting the influence of other agents on the reward signal. This approach is highly limited in our setting where the reward signal is allowed to be subject to noise.

3 Preliminaries

A fully cooperative MAS is modelled by a decentralised-Markov decision process (dec-DEC-MDP) (Yang & Wang, 2020). A dec-DEC-MDP is an augmented MDP involving two or more agents $\{1, \dots, N\} =: \mathcal{N}$ with a common goal that independently decide actions to take which they do so simultaneously over many rounds. Formally, a dec-DEC-MDP is a tuple $\mathfrak{M} = \langle \mathcal{N}, \mathcal{S}, (\mathcal{A}_i)_{i \in \mathcal{N}}, P, R, \gamma \rangle$ where $\mathcal{S}$ is the finite set of states, $\mathcal{A}_i$ is an action set for agent $i \in \mathcal{N}$ and $R : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathcal{P}(D)$ is the team reward function that all agents jointly seek to maximise where D is a compact subset of $\mathbb{R}$ and lastly, $P : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ is the probability function de-

scribing the system dynamics where $\boldsymbol{\mathcal{A}} := \times_{i=1}^{N} \mathcal{A}_i$. In this paper, we consider a partially observable MAS so that given the system is in the state $s_t \in \mathcal{S}$, each agent $i \in \mathcal{N}$ makes only local observations $\tau_{t,i} = O(s_t, i)$ where $O : \mathcal{S} \times \mathcal{N} \to \mathcal{Z}_i$ is the observation function and $\mathcal{Z}_i$ is the set of local observations for agent i. Each agent $i \in \mathcal{N}$ uses a *Markov policy* $\pi_{i,\theta_i} : \mathcal{Z}_i \times \mathcal{A}_i \to [0, 1]$ to decide its actions, where the policy is parameterised by the vector $\theta_i \in \mathbb{R}^d$. Throughout the paper, we occasionally drop the policy parameter and write π_{i,θ_i} as π_i. At each time $t \in 0, 1, \ldots,$ the system is in state $s_t \in \mathcal{S}$ and each agent $i \in \mathcal{N}$ takes an action $a_t^i \in \mathcal{A}_i$. The *joint action* $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^N) \in \boldsymbol{\mathcal{A}}$ produces an immediate reward $r_i \sim R(s_t, \boldsymbol{a}_t)$ for agent $i \in \mathcal{N}$ and influences the next-state transition which is chosen according to P. For each agent $i \in \{1, \ldots, N\}$, the state value function $v^{\boldsymbol{\pi}}$ and the state-action value function $Q^{\boldsymbol{\pi}}$ are given by: $v^{\boldsymbol{\pi}}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s, \boldsymbol{a}) \middle| s_0 = s, \boldsymbol{a} \sim \boldsymbol{\pi}\right]$ and $Q^{\boldsymbol{\pi}}(s, \boldsymbol{a}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s, \boldsymbol{a}) \middle| s_0 = s, \boldsymbol{a_0} = \boldsymbol{a}; \boldsymbol{a} \sim \boldsymbol{\pi}\right]$ respectively. We denote by Π_i each agent's compact Markov policy space and write $\boldsymbol{\Pi} := \times_{i \in \mathcal{N}} \Pi_i$.

4 The MANSA Framework

We now describe the details of the MANSA framework and how it learns to determine when to use a centralised learning process and how it improves learning and performance. We then describe the agents' objectives and learning processes.

To tackle the challenges described above, we introduce to the system of N MARL agents Global, an *adaptive* RL agent with its own objective. Global's role is to determine at each state whether the N MARL agents must use a centralised critic or a decentralised critic. Using observations of the joint actions played by the N agents, the goal of Global is to improve the learning process and maximise of the team performance by performing activations of the centralised critic. To do this, Global learns how to select between two critics, Central and Decentral. Global's policy space consists of a form of policies known as *switching controls* (Mguni et al., 2021b) which enable Global to decide at which states to activate CL. At each state Global first makes a *binary decision* to decide to *activate* Central.

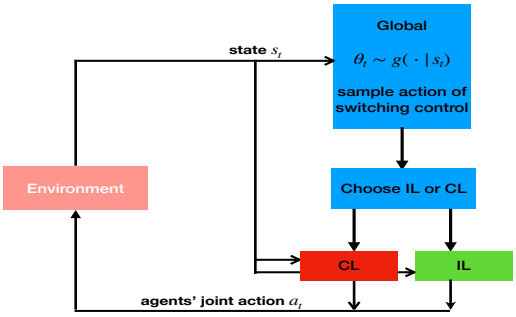

Figure 2: MANSA schematic.

Algorithm 1 **M**ulti **A**gent **N**etwork **S**election **A**lgorithm (**MANSA**)

Input: Initial agent policies $\pi_0^1, \ldots, \pi_0^N$, Global policies $\mathfrak{g}_{c_0}, g_0$, RL learning algorithm Δ
Output: Optimised agent joint policy $\boldsymbol{\pi}^\star$
for $t = 1, T$ **do**
  Given environment state s_t evaluate $g_t \sim \mathfrak{g}(\cdot|s_t)$ **if** $g = 1$ **then**
    $\boldsymbol{a}_t \sim \boldsymbol{\pi}^c(\cdot|s_t)$ **Use** Central
  else
    $\boldsymbol{a}_t \sim \boldsymbol{\pi}^d(\cdot|\tau_{t,i})$ **Use** Decentral
  Apply action $\boldsymbol{a}_t$ to obtain s_{t+1} and $\boldsymbol{r}_{t+1} := \sum_{i \in \mathcal{N}} r_{i,t+1}$ by applying $\boldsymbol{a}_t$ to environment.
  Update $(\pi^i)_{i \in \mathcal{N}}$, $(\mathfrak{g}, g)$ using $(s_t, \boldsymbol{a}_t, r_i, s_{t+1})$ and $(s_t, \boldsymbol{a}_t, (r_i)_{i \in \mathcal{N}}, s_{t+1})$ resp. and Δ **// Learn the individual policies**

To induce Global to selectively choose when to activate Central, each activation incurs a fixed cost for Global. In this case, the objective for Global is: $v_G^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, \boldsymbol{a}_t) - c\sum_{k\geq 1}\delta^t_{\tau_k}\right]$, where $c \in \mathbb{R}_{>0}$ is a fixed positive constant and $\delta^t_{\tau_k}$ is the Kronecker-delta function which is 1 whenever $t = \tau_k$ and 0 otherwise, assigns a (strictly negative) cost for each activation. The cost ensures Global activates Central only when doing so delivers a performance improvement. With this objective, Global seeks to maximise the system performance by activating CL at the required set of states while, due to the cost of doing so, minimising the number of unnecessary CL calls.

We now describe how at each state the decision to activate Central is determined. At any state, the decision to activate Central is decided by a (categorical) policy $\mathfrak{g} : \mathcal{S} \to \{0, 1\}$ which acts according to Global's objective. In particular, during training, first, Global makes an observation of the state $s_k \in \mathcal{S}$ and the joint action $\boldsymbol{a}_k$ and using $\mathfrak{g}$, Global decides whether or not to activate Central. We denote by $\{\tau_k\}$ the times that an activation takes place, for example, if the activation of CL is first made at state s_5 then again at s_7, then $\tau_1 = 5$ and $\tau_2 = 7$. Recalling the role of $\mathfrak{g}$, $\{\tau_k\}$ obey the expression $\tau_k = \inf\{t > \tau_{k-1} | s_t \in \mathcal{S}, \mathfrak{g}(s_t) = 1\}$ and are therefore[1] ***rules*** *that depend on the state*. Hence, by learning an optimal $\mathfrak{g}$, Global learns the optimal switching control policy.

MANSA's components

We now describe MANSA's core components which consists of CL, DL and the switching control agent. Each component can be replaced by various other MARL algorithms.

• N MARL agents. Each agent has two value-based policies that is, each agent has (i) an action policy with a critic that takes as input agent's *global observation* which includes the joint action and global state, and (ii) an action policy with a critic that takes as input only the agent's local observation.

• **Independent Q-Learning (IQL)**. In this paper, we use IQL (Tan, 1993) to learn the decentralised critic policies. IQL is a popular RL algorithm which is off-policy.

• **QMIX**. For the CL, we use QMIX (Rashid et al., 2018) which is a multi-agent value-based method that can train decentralised policies in a decentralised manner and guarantees consistency between the centralised and decentralised policies.

• **Switching Control Policy**. A PPO agent called Global whose policy's action set consists of two actions: use centralised policy, do not use centralised policy. Global updates its policy $\mathfrak{g}$ while the agents $\{1, \ldots, N\}$ learn their individual policies $\{\pi_1, \ldots, \pi_N\}$.

Discussion on Computational Aspect

The switching control mechanism results in a framework in which the problem facing Global has a markedly reduced computational complexity as compared with that facing the Central and Decentral (though the learners share the same experiences). Crucially, the decision space for Global is $\mathcal{S} \times \{0, 1\}$ i.e at each state it makes a binary decision. Consequently, the learning process for $\mathfrak{g}$ is much quicker than either Central or Decentral's policy which must optimise over a decision space which is $|\mathcal{S}||\mathcal{A}|$ (choosing an action from its action space at every state) and $|\mathcal{S}||\mathcal{A}|^N$ respectively. This results in Global rapidly learning its optimal policy (relative to the base MARL learners).

5 Convergence and Optimality of MANSA

The addition of Global's RL switching control process during learning can produce convergence issues (Zinkevich et al., 2006). We now show that MANSA converges. To do this we study the problem for Global and show its problem admits a stable point that can be computed using standard RL methods. We secondly show that the solution to MANSA ensures weakly higher performing agent policies than what would be achieved by solving $\mathfrak{M}$ directly. In particular, we prove the following:

1. MANSA converges to the system solution and does so with (linear) function approximators.

2. MANSA leads to higher overall return.

3. MANSA-B ensures maximal performance for a given number of CL calls (CL call budget).

[1] More precisely, $\{\tau_k\}_{k\geq 0}$ are *stopping times* (Øksendal, 2003).

The results are built under Assumptions 1 - 7 (Sec. 12 of the Appendix) which are standard in RL and stochastic approximation theory.

We now give our first result that shows that the solution of the system $\mathcal{G}$ can be computed as a limit point of a sequence of Bellman operations. Second, the result shows the convergence of MANSA and, it converges with (linear) function approximators. In what follows, we define a *projection* Π on a function Λ by: $\Pi\Lambda := \underset{\bar{\Lambda}\in\{\Psi r | r\in\mathbb{R}^p\}}{\arg\min} \left\|\bar{\Lambda} - \Lambda\right\|$:

Theorem 1. ***i)*** *Let $V : \mathcal{S} \to \mathbb{R}$ then the solution of $\mathcal{G}$ is given by $\lim_{k\to\infty} T^k V^{\boldsymbol{\pi},\mathfrak{g}} = \max_{\hat{\boldsymbol{\pi}}\in\boldsymbol{\Pi},\hat{\mathfrak{g}}} V^{\hat{\boldsymbol{\pi}},\hat{\mathfrak{g}}} = V^{\boldsymbol{\pi}^\star,\mathfrak{g}^*}$, where $(\boldsymbol{\pi}^\star, \mathfrak{g}^*)$ and T is a stable policy profile and the Bellman operator of $\mathcal{G}$ resp.*

ii) *MANSA converges to the stable point of $\mathcal{G}$, moreover, given a set of linearly independent basis functions $\Psi = \{\psi_1, \ldots, \psi_p\}$ with $\psi_k \in L_2, \forall k$, MANSA converges to a limit point $r^\star \in \mathbb{R}^p$ which is the unique solution to $\Pi\mathfrak{F}(\Psi r^\star) = \Psi r^\star$ where $\mathfrak{F}$ is defined by: $\mathfrak{F}\Lambda := \hat{R}_1 + \gamma P \max\{\mathcal{M}\Lambda, \Lambda\}$ where $r^\star$ satisfies: $\|\Psi r^\star - Q^\star\| \leq (1-\gamma^2)^{-1/2} \|\Pi Q^\star - Q^\star\|$.*

Part i) of the Theorem proves the system in which Global and the N agents jointly learn has a stable point which is the limit of a dynamic programming procedure. Crucially, the limit point corresponds to the solution of the MDP $\mathcal{M}$. Part ii) establishes the solution to $\mathcal{G}$ can be computed using MANSA. The result also establishes the convergence of MANSA to the solution using (linear) function approximators and that the approximation error is bounded by the smallest error achievable given the basis functions.

Proposition 1. *There exists some finite integer N such that $v^{\tilde{\boldsymbol{\pi}}_m}(s) \geq v^{\boldsymbol{\pi}_m}(s),\ \forall s \in \mathcal{S}$ for any $m \geq N$ where $\tilde{\pi}_m$ and π_m are the respective agent joint policies after the m^{th} learning iteration with and without Global's influence.*

Note that *a fortiori* Prop. 1 implies $v^{\tilde{\boldsymbol{\pi}}}(s) \geq v^{\boldsymbol{\pi}}(s),\ \forall s \in \mathcal{S}$. Prop. 1 shows that Global (weakly) improves joint system outcomes. Additionally, the proposition indicates that the introduction of Global never leads to a reduction to the total (environment) return.

Theorem 2. *Consider the c-SG $\widetilde{\mathcal{G}}$ for the problem* **??** *then: a) The Bellman equation is satisfied that is $\exists \tilde{V}_G^{*,\boldsymbol{\pi},\mathfrak{g}}$ such that $\tilde{V}_G^{*,\boldsymbol{\pi},\mathfrak{g}}(\boldsymbol{z}) = \max_{\boldsymbol{a}\in\mathcal{A}} \left(\widetilde{R}(\boldsymbol{z},\boldsymbol{a}) + \gamma \mathbb{E}\left[\tilde{V}_G^{*,\boldsymbol{\pi},\mathfrak{g}}(\boldsymbol{z}')\right]\right)$, where Global's optimal policy takes the form $\mathfrak{g}^*(\cdot|z)$. b) Given a $\widetilde{V} : \mathcal{Z} \to \mathbb{R}$, the solution of $\widetilde{\mathcal{G}}$ is $\lim_{k\to\infty} \tilde{T}^k \widetilde{V}^{\boldsymbol{\pi}} = \max_{\hat{\boldsymbol{\pi}}\in\boldsymbol{\Pi},\hat{\mathfrak{g}}} \widetilde{V}^{\hat{\boldsymbol{\pi}},\hat{\mathfrak{g}}} = \widetilde{V}^{*,\boldsymbol{\pi},\mathfrak{g}^*}$, where $(\boldsymbol{\pi}^*, \mathfrak{g}^*)$ and is a stable policy profile of $\widetilde{\mathcal{G}}$ and $\tilde{T}$ is the Bellman operator of $\widetilde{\mathcal{G}}$.*

The result has several important implications. The first is that we can use our MARL based method, MANSA to obtain the solution of Global's problem while guaranteeing convergence (under standard assumptions). Secondly, our state augmentation procedure admits a Markovian representation of Global's optimal policy.

6 Experiments

We performed a series of experiments to test MANSA's ability to successfully learn a switching control policy between CL and IL that improves performance. Specifically, we tested if MANSA: 1. Performs CL calls less often in strongly decoupled settings. 2. preserves MARL convergence properties. 3. Reduces the failure modes of each of these game classes. 4. Can improve overall performance. For these experiments, we used Normal-form games, Level-based Foraging (LBF) (Papoudakis et al., 2021) and StarCraft Multi-Agent Challenge (SMAC) (Samvelyan et al., 2019). These environments have specific features which in some cases are advantageous to CL, and in some cased to IL as we describe below.

We used the code accompanying the MARL benchmark study of Papoudakis et al. (2021) for the baselines. We implemented MANSA on top QMIX (Rashid et al., 2018) (as the CL) and IQL (Tan, 1993) (as the IL). We used SAC (Haarnoja et al., 2018) to learn the switching control policy itself. In all plots, the dark lines represent averages over three seeds and the shaded regions represent 95% confidence intervals.

Normal Form Games

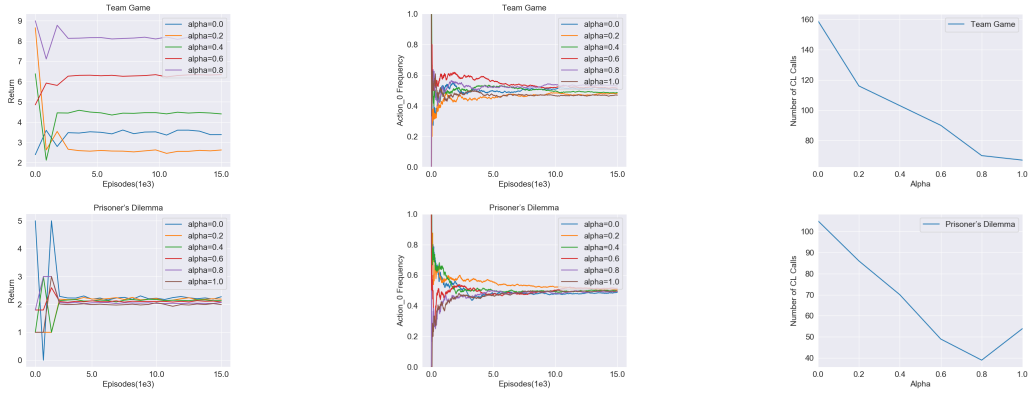

Figure 3: Normal form games. The top row shows results on Team games and the bottom row shows results on Prisoner's dilemma. Right hand plot shows that when the rewards of the agents as the rewards of the agents becomes more decoupled as $\alpha \to 1$ i.e. the strategic interaction becomes weak, MANSA reduces the number of CL calls it makes during training.

We tested two different examples of two-player normal form games (matrix games). We slightly modified the standard normal form game set up by parameterising the reward functions of the game by a parameter $\alpha \in [0, 1]$: α quantifies how strongly coupled the interaction between the two agents is. That is, the degree to which the actions of an agent affects the rewards of the other agent. For $\alpha = 0$, the games are strongly coupled and for $\alpha = 1$, the games are completely decoupled. We investigated the behaviour of MANSA for various values of α within the interval $[0, 1]$, in particular, we test how often MANSA calls on the CL class of learners for different values of α. These are state-less environments, and precise details of the pay-off matrices are given in Appendix 8.

We test the following cases: (1) Team games $\mathfrak{R}_i(a_i, a_j) = \mathfrak{R}_j(a_j, a_i)$ (Fig. 3 top row), and (2) Nonzero-sum games with a strictly dominant strategy, namely,the *prisoner's dilemma* (Fig. 3 bottom row). Fig. 3 shows plots of return (left column), frequency of usage of IL (centre column), and number of calls to CL (right column). As expected, as α increases and the reward function becomes more decoupled, calls to the CL decrease in both environments. Similarly, in both environments, as shown in the centre column, increasing α results in higher frequency of using "Action_0" of the switching controller, i.e., IL. These plots show a smooth modulation of calls to CL and usage of IL with respect to α. This suggests that MANSA is capable of picking the best of CL and IL with high degree of of granularity. Furthermore, this experiment provides empirical evidence that MANSA can identify and call CL less often as the environment becomes more decoupled.

Level-based Foraging (LBF). As depicted in Fig. 4, in LBF, an agent controls units of particular levels (in the example, both units are level 1), and there are apples of particular levels (in the example, all apples are level 2) scattered around the map. The agents goal is to collect as much food as possible. Crucially, the agents can only collect a food if the cumulative level of the agents adjacent to the food that are executing the "collect" action is greater than or equal to the level of the food. As the agent levels and the food levels are randomly assigned, some food may be collectable by a single agent, while some food may require the coordination of all agents. That is, LBF has attributes such that in some circumstances coordination among the agents is required, while in some subregions the agents sparsely interact. Finally, this environment gives us the option of enforcing coordination (these map names are suffixed with "coop") by making the food level such that at least two agents are required to coordinate to collect any food in the environment.

Fig. 4 shows performance curves of the tested algorithms. The implementation of MANSA in these experiments uses QMIX for CL and IQL for IL (i.e., it uses the same baselines against which it is compared). As can be seen, MANSA outperforms these baselines by a notable margin in almost half the maps (four out of ten). This suggests MANSA, which identifies states that benefit from centralised training (and the states that do not) yields significant performance gains in addition to providing robustness against failure modes.

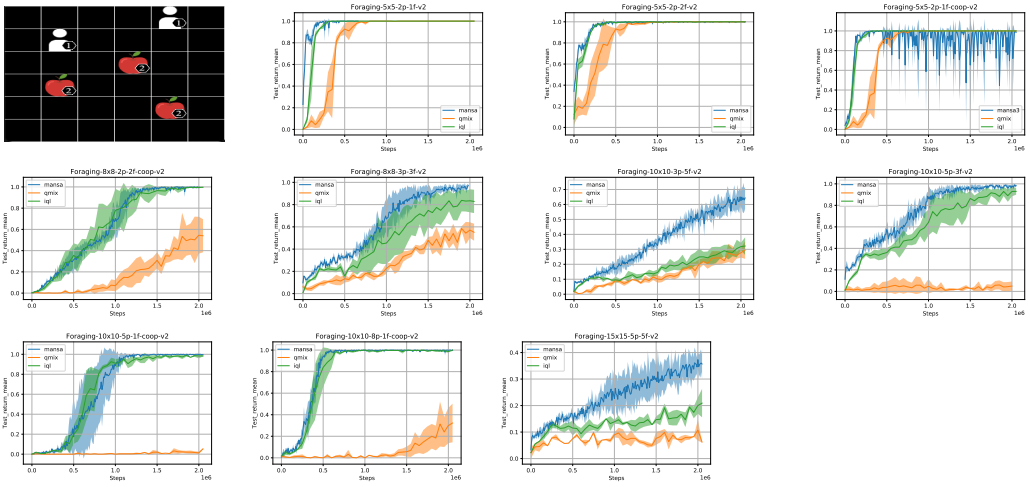

Figure 4: Level Based Foraging Environment (top left) and Learning curves on LBF. MANSA outperforms baselines and demonstrates strong robustness across the range of maps.

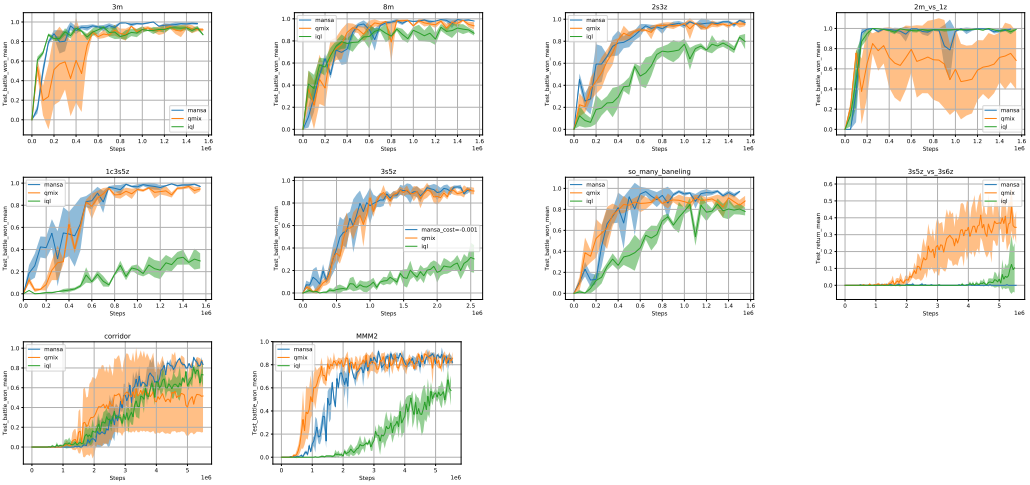

Figure 5: Learning curves on SMAC. MANSA demonstrates robust performance across and is not susceptible to failure cases unlike the base CL method (QMIX) or the base IL method (IQL).

Surprisingly the IL baseline (IQL) generally outperformed the CL baseline (QMIX), even in maps that have reward structures that require strong coordination between agents: *Foraging-5x5-2p-1f-v2*, *Foraging-10x10-5p-1f-coop-v2*, and *Foraging-10x10-8p-1f-coop-v2*. In particular, in these maps, agents must coordinate to enact the "collect" action simultaneously in order to capture the food. Despite this requirement, as the results show, contrary to assumptions, IL outperformed CL. The benefit of MANSA is evident — by not requiring any a priori commitment to either CL or IL, MANSA performs robustly in these maps. Our empirical results in LBF validate MANSA's preservation of the convergence properties of MARL and its ability to leverage both CL and IL to deliver higher performance. In Section 9 of the appendix, we show that while the switching cost parameter does impact MANSA's performance, it is relatively easy to tune this parameter.

StarCraft Multi-Agent Challenge (SMAC). The goal in SMAC is for a team of units under the control of the agent to defeat a team of units under an opponent's control. Different maps available in SMAC vary along several dimensions: number of units, diversity of units, degree of coordination required, and terrain, to name a few. In different SMAC maps (and indeed under different circumstances within a map) varying levels of coordination are needed. For example, in the map *so_many_baneling*, zealots under the agent's control face off a larger army of enemy banelings. As banelings attack by

exploding on contact with their opponents and cause significant "splash" damage, it is critical for units under the agent's control to space out so as to minimise the effects of "splash" damage. On the other hand, in the map *corridor*, such coordination may not be needed. Here, a small army of zealots under the agent's control face off against a large army of zerglings. Ideally, the zealots ought to wall-off a choke-point and take down the enemy and must avoid getting surrounded. While it may seem like significant coordination is required to solve this map (i.e., all zealots converge to the choke-point), in fact this is not necessary. Due to location of the choke-point, the optimal actions for a zealot acting independently mirror those of the coordinated group. That is, IL is as good as CL in this case. These two examples illustrate that SMAC is not a "one-algorithm-fits-all" environment. The design of SMAC sometimes requires coordination and sometimes does not, and a robust algorithm is one which is not impacted by this requirement for flexibility.

Fig. 5 shows 'Test_battle_won_mean', vs 'Steps' for a range of SMAC maps for MANSA and the baselines. As can be seen, MANSA converges to performance that is as good as the top-performing baseline in all maps except *3s5z_vs_3s6z*. In particular, MANSA's flexibility allows it to avoid the failures of IQL in maps such as *1c3s5z*, *3s5z*, *2s3z*, and *MMM2* without resorting to using only CL (we present tabulated % of calls to CL in Section 10). Similarly, it also avoids the failures of QMIX (centralised policy) in *2m_vs_1z* and *corridor*. Therefore, MANSA is robust to failures due to the representational constraints of CL and the failure modes of IL.

7 Conclusion

In this paper, we presented MANSA, a novel framework to solve multi-agent systems. MANSA combines IL and CL in a way such that it enables one to benefit from the "best-of-both-worlds". It allows one to utilise the computational benefits of IL, while not suffering from convergence issues due to its CL component. We proved that MANSA converges. Finally, we presented a detailed suite of experimental results on normal form games, LBF, and SMAC. In all these domains, MANSA performed robustly and had no failure modes, and indeed, to our surprise sometimes even strongly out-performed the baselines (despite using them as its components). For future work, we are exploring methods to develop versions of MANSA with an explicit budget for to CL.