MANSA: Learning Fast and Slow in Multi-Agent Systems

Anonymous Authors[1]

Abstract

In multi-agent reinforcement learning (MARL), independent learning (IL) often shows remarkable performance and easily scales with the number of agents. Yet, using IL can be inefficient and runs the risk of failing to successfully train, particularly in scenarios that require agents to coordinate their actions. Using centralised learning (CL) enables MARL agents to quickly learn how to coordinate their behaviour but employing CL everywhere is often prohibitively expensive in real-world applications. Besides, using CL in value-based methods often needs strong representational constraints (e.g. individual-global-max condition) that can lead to poor performance if violated. In this paper, we introduce a novel plug & play IL framework named **M**ulti-**A**gent **N**etwork **S**election **A**lgorithm (MANSA) which selectively employs CL only at states that require coordination. At its core, MANSA has an additional agent that uses *switching controls* to quickly learn the best states to activate CL during training, using CL only where necessary and vastly reducing the computational burden of CL. Our theory proves MANSA preserves cooperative MARL convergence properties, boosts IL performance and can optimally make use of a fixed budget on the number CL calls. We show empirically in Level-based Foraging (LBF) and StarCraft Multi-agent Challenge (SMAC) that MANSA achieves fast, superior and more reliable performance while making 40% fewer CL calls in SMAC and using CL at only 1% CL calls in LBF.

1. Introduction

Multi-agent reinforcement learning (MARL) has emerged as a powerful framework that enables autonomous agents to complete various tasks in areas such as autonomous driving (Zhou et al., 2020), swarm robotics (Mguni et al., 2018; 2019) and smart grids (Wang et al., 2021; Qiu et al., 2021; 2022). 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 way of training agents in a decentralised manner. Since the agents ignore other agents, IL can be trained quickly as each agent's learning process is contingent on only its local observations and own actions. This is efficient in scenarios that require only weak interactions between agents (Kok & Vlassis, 2004).

Despite these apparent benefits, training MARL using IL has several formidable drawbacks: with no ability to observe the actions of other agents, random occurrences of successful coordination among IL agents are improbable, causing IL methods to sometimes struggle in tasks that require coordination (Hernandez-Leal et al., 2017). Also, ignoring other agents' influence on the system means from the agent's perspective, the environment can appear non-stationary which precludes convergence guarantees (Yang & Wang, 2020).

On the other hand, MARL learners can be trained in simulated environments in which agents can be provided with other agents' observations and other state information. Centralised training and decentralised execution (CT-DE) (Kraemer & Banerjee, 2016; Foerster et al., 2018; McAleer et al., 2022) is a framework that uses a centralised critic that exploits global information during training while performing execution in a decentralised fashion. With this added information during training, agents can learn to condition their policies on other agents' actions which mitigates the appearance of non-stationarity. The CT-DE 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., 2021b) 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 is a strong coordination component that produces a need for global observations during training (Sharma et al., 2021). Nevertheless, CT-DE suffers from an explosive growth in complexity since the joint action-state space grows exponentially with the number of agents (Deng

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

et al., 2021). Consequently, CT-DE methods can require large numbers of samples to complete training. In regions in which the agents do not strongly interact, this added complexity can prove to be an unnecessary burden as agents do not benefit from global information (Kok & Vlassis, 2004). Fig. 1 shows an example scenario in which the agents are required to coordinate only at a small subregion.

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 that can produce suboptimal value approximation.

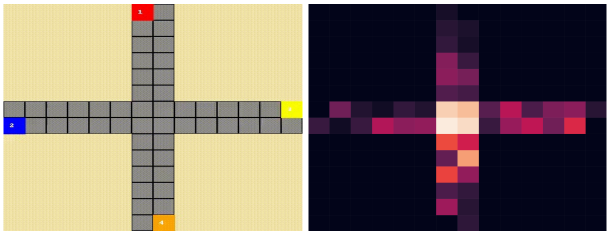


Figure 1. Left. In this Traffic Junction scenario, to avoid collisions agents (coloured squares) need only coordinate at the intersection. Before, their actions do not affect others so using IL at these states is sufficient. *Right.* Heatmap of MANSA's CL calls. MANSA activates CL most at the intersection where coordination is needed.

To tackle these issues, we introduce a general plug & play MARL framework, MANSA which optimally selects where in the environment to call on centralised learners to boost IL during training. MANSA involves a decentralised learning method, and, an *adaptive* reinforcement learning (RL) agent that presides over when CL or IL is used. Specifically, the additional agent determines at which states to activate CL while IL is used at all other states. This is in contrast to current MARL methods that use solely either CL or IL at all states throughout training. A key feature of MANSA is the novel combination of RL and a form of policy known as *switching controls* (Mguni et al., 2022; 2021a;b) . Switching controls are policies that introduce a switch mechanism that affects some control process in a dynamical system (Mguni, 2018). In our case, as we show, this enables the adaptive RL agent to quickly determine where to switch to CL while the IL and CL (off-policy) learners concurrently train. This allows the benefits of both algorithm classes to be leveraged while overcoming some of the issues of any one class.

Since CL calls are expensive, it can be useful to consider enforcing a fixed budget on the number of CL calls during training. To this end in Sec. 5, we extend MANSA to enable it to solve MARL problems while respecting a budgetary constraint of the number of allowed CL calls during training.

Overall, MANSA has several advantages:

• By switching to CL only at the set of states in which it is beneficial while leveraging the benefits of IL, MANSA increases the learning efficiency of CT-DE (see Sec. 6.1).
• 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 failure (see Sec. 6.2.2).
• 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 CL methods (see Sec. 6.2.1). Additionally, MANSA allows for a fixed budget for calls of CL (see Sec. 6.3).
• MANSA is a plug & play framework which seamlessly adopts any MARL algorithm (see Sec. 6.2).

To enable MANSA to perform successfully, we tackle several challenges. First, including a new adaptive RL agent that learns while the N MARL agents are training can occasion convergence issues. Second, the adaptive RL agent uses switching controls which differs from the frameworks of standard RL. To this end, we prove MANSA preserves the MARL convergence properties (Theorem 1) and boosts the performance of IL agents (Prop. 1). We then characterise the optimal CL activation points with an online condition enabling it to quickly determine where switching to CL is beneficial during the agents' training phase (Prop. 2).

When the problem includes budgetary constraints on the number of allowed CL calls, as the number of CL calls accumulates there is less freedom to execute more CL activations further on during training. Therefore to make optimal use of the allowed number CL calls, it is necessary to learn a policy that optimally decides whether to activate CL calls *given its remaining budget*. We resolve this by using a *state augmentation* technique which treats the remaining budget as a state component (Theorem 2). State augmentation techniques originated in control theory (Daryin & Kurzhanski, 2005) and have recently been adapted to single agent RL (Sootla et al., 2022; Mguni et al., 2022).

2. Related Work

A key aim of the CT-DE framework is to ensure the policies it generates are consistent with the desired system goal. This principle is known the Individual-Global-Max (IGM) principle (Son et al., 2019). To realise this in the CT-DE framework, QMIX (Rashid et al., 2018) and VDN (Sunehag et al., 2017) propose two sufficient conditions of IGM to factorise the joint action-value function. Crucially, such decompositions and limited by the action-value function classes they can represent and can perform badly in 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 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 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, in some tasks, MAPPO has been shown to be outperformed by IL, specifically, PPO (Schulman et al., 2017) with the latter only needing modest hyperparameter tuning (de Witt et al., 2020). Consequently, in this paper, we realise our framework within value-based methods. Nevertheless, MANSA's plug & play facility means that actor-critic methods can be accommodated naturally in extensions.

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 that require the global coordination requirements of the system to be known beforehand. Moreover, other works centered 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 both a priori unknown and noisy.

3. MANSA

A fully cooperative multi-agent system is modelled by a decentralised-Markov decision process (dec-MDP). A dec-MDP is an augmented MDP involving two or more agents $\{1,\ldots,N\} =: \mathcal{N}$ with a common goal that each independently decide actions to take which they do so simultaneously over many time steps. Formally, a 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 $i \in \mathcal{N}$. 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* $\boldsymbol{a}_t = (a_t^1, \ldots, a_t^N) \in \boldsymbol{\mathcal{A}} \equiv \times_{i=1}^N \mathcal{A}_i$ produces an immediate reward $r \sim R(s_t, \boldsymbol{a}_t)$ where $R : \mathcal{S} \times \boldsymbol{\mathcal{A}} \to \mathcal{P}(D)$ is the team reward function that all agents jointly seek to maximise and where D is a compact subset of $\mathbb{R}$ and $\mathcal{P}$ is some distribution on $\mathbb{R}$. Lastly, $P : \mathcal{S} \times \boldsymbol{\mathcal{A}} \times \mathcal{S} \to [0, 1]$ is the probability function describing the system dynamics. We consider a partially observable system 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. To decide its action each agent samples its *Markov policy* $\pi_{i,\theta_i} : \mathcal{Z}_i \times \mathcal{A}_i \to [0, 1]$ which is parameterised by the vector $\theta_i \in \mathbb{R}^d$ and is contained in Π_i. We occasionally drop the parameter θ_i and write π_i and we denote by $\boldsymbol{\Pi} := \times_{i\in\mathcal{N}} \Pi_i$. For any agent and for any joint policy $\boldsymbol{\pi} \in \boldsymbol{\Pi}$, the state value and state-action value function are: $v(s|\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r \middle| s_0 = s, \boldsymbol{a} \sim \boldsymbol{\pi}\right]$ and $Q(s, \boldsymbol{a}|\boldsymbol{\pi}) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r \middle| s_0 = s, \boldsymbol{a_0} = \boldsymbol{a}; \boldsymbol{a} \sim \boldsymbol{\pi}\right]$ respectively.

We now describe the core details of MANSA, 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.

3.1. Framework

To tackle the challenges described, we equip each MARL agent with access to both a centralised learner, which we call Central and an independent learner, which we call Independent. MANSA includes an additional RL agent, Global, i.e., the switching controller, that decides on the states to activate Central during the agents' training phase while using Independent as the learning algorithm everywhere else.

Fig. 2 shows a schematic representation of MANSA. Global observes the global state s_t of the environment and samples the discrete policy of the switching controller $g_t \sim \mathfrak{g} : \mathcal{S} \to \{0, 1\}$. If $g_t = 0$, each of the N agents in the environment use their respective local observations of the environment to generate actions $\boldsymbol{a}_t$ from the policy of Independent. If $g_t = 1$, $\boldsymbol{a}_t$ is generated from the policy of Central using the global state. The agents' actions $\boldsymbol{a}_t$ are executed in the environment and the loop repeats. The trajectories generated by this process are stored in a replay buffer from which Global, Independent, and Central are trained.

The Global agent is endowed with its own objective which captures its goal to improve the learning process and max-

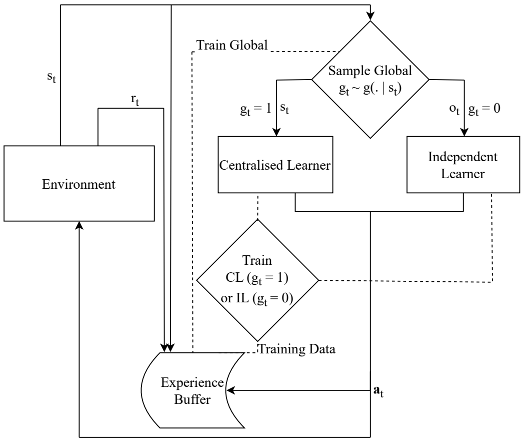


Figure 2. MANSA schematic.

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

```
Input: Independent policies $\boldsymbol{\pi}^i$, centralised policies $\boldsymbol{\pi}^c$,
    Global policy $\mathfrak{g}_0$, independent learning algorithm
    $\Delta^i$, centralised learning algorithm $\Delta^c$, learning algorithm for Global $\Delta^g$, experience buffer $\boldsymbol{B}$
Output: Optimised policies $\boldsymbol{\pi}^{i\star}$, $\boldsymbol{\pi}^{c\star}$, and $\mathfrak{g}^\star$
for $t = 1, T$ do
    Given environment state $s_t$ evaluate $g_t \sim \mathfrak{g}(\cdot|s_t)$
    if $g_t = 1$ then
        Sample action using global state $\boldsymbol{a}_t \sim \boldsymbol{\pi}^c(\cdot|s_t)$ Use Central
    else
        Sample action using local observations $\boldsymbol{a}_t \sim \boldsymbol{\pi}^d(\cdot|\boldsymbol{\tau}_t)$ Use Independent
    Apply action $\boldsymbol{a}_t$ to environment to obtain $s_{t+1}$, $\boldsymbol{\tau}_{t+1}$
     and $\boldsymbol{r}_{t+1} := \sum_{i \in \mathcal{N}} r_{i,t+1}$
    Store $(s_t, \boldsymbol{\tau}_t, \boldsymbol{a}_t, r_{t+1}, s_{t+1}, \boldsymbol{\tau}_{t+1})$ in $\boldsymbol{B}$
    if $g_t = 1$ then
        Sample $\boldsymbol{B}$ to obtain $(s_i, \boldsymbol{a}_i, \boldsymbol{r}_i, s_{i+1})$ and update $\boldsymbol{\pi}^c$
         with $\Delta^c$ (Discard $\boldsymbol{\tau}_t, \boldsymbol{\tau}_{t+1}$)
    else
        Sample $\boldsymbol{B}$ to obtain $(\boldsymbol{\tau}_i, \boldsymbol{a}_i, \boldsymbol{r}_i, \boldsymbol{\tau}_{i+1})$ and update
         $\boldsymbol{\pi}^i$ with $\Delta^i$ (Discard $\boldsymbol{s}_t, \boldsymbol{s}_{t+1}$)
    Sample B to obtain $(s_i, g_i, \boldsymbol{r}_i, s_{i+1})$ and update $\mathfrak{g}$ with
     $\Delta^g$ (Discard $\boldsymbol{a}_t, \boldsymbol{\tau}_t, \boldsymbol{\tau}_{t+1}$)
```

imise the performance of the system of the N MARL agents through its decisions of where to activate Central. To induce Global to selectively choose when to perform an activation, each activation incurs a fixed cost for Global which is quantified by a fixed constant $c > 0$. These costs ensure that any activation of the CL critic must be beneficial to the performance of the system either at the current or subsequent states. The objective for Global is:

$$v_G(s|\boldsymbol{\pi}, \mathfrak{g}) = \mathbb{E}_{g\sim\mathfrak{g}}\left[\sum_{t=0}^{\infty} \gamma^t \left(r - c \cdot \mathbf{1}(g(s_t))\right) \Big| s_0 = s; \boldsymbol{a} \sim \boldsymbol{\pi}\right],$$

and Global's action-value function is $Q_G(s, \boldsymbol{a}|\boldsymbol{\pi}, \mathfrak{g}) = \mathbb{E}_{g\sim\mathfrak{g}}\left[\sum_{t=0}^{\infty} \gamma^t (r - c \cdot \mathbf{1}(\mathfrak{g}(s_t)))|s_0 = s, \boldsymbol{a_0} = \boldsymbol{a}; \boldsymbol{a} \sim \boldsymbol{\pi}\right]$.

With this objective, Global's goal is to maximise the system performance by activating Central at the required set of states to enable the agents to solve $\mathfrak{M}$ with the minimal number of CL activations. Therefore, by learning an optimal $\mathfrak{g}$, Global acquires the optimal policy for activating Central.

Adding the agent Global with an objective distinct from the N agents results in a non-cooperative Markov game $\mathcal{G} = \langle \mathcal{N} \times \{G\}, \mathcal{S}, ((\mathcal{A}_i)_{i\in\mathcal{N}}, \mathcal{A}_G), P, (R, R_G), \gamma \rangle$ where G, $\mathcal{A}_G := \{0, 1\}$ and $R_G(s, a, g) := R(s, a) - c \cdot \mathbf{1}(g)$ denote the Global agent, its action set and its reward function respectively. In MARL, having multiple learners with a payoff structure that is neither zero-sum nor a team game can occasion convergence issues (Shoham & Leyton-Brown, 2008). Moreover, unlike standard MARL frameworks, MANSA incorporates switching controls used by Global. Nevertheless in Sec. 4 we prove the convergence of MANSA under standard assumptions.

Details on Architecture

MANSA's components. We now describe a concrete realisation of MANSA's core components which consist of N MARL agents, a CL RL algorithm as Central, an IL RL algorithm as Decentral and a switching control RL algorithm as Global. Each (MA)RL component can be replaced by various other (MA)RL algorithms.

- *N* **MARL agents**. Each agent has two value-based policies. That is, each agent has (1) a policy induced by a value function that takes as input agent's *global observation* which includes the joint action and global state, and (2) an action policy induced by a value function that takes as input only the agent's local observation.
- **Independent Q-Learning (IQL)**. In this paper, we use IQL (Tan, 1993) to train Decentral. IQL is a popular RL algorithm which is off-policy.
- **QMIX**. For training Central, we use QMIX (Rashid et al., 2018), an off-policy MARL value-based method that accommodates only action value functions that adhere to a monotonicity constraint in the combination of the agents' individual value functions.
- **Switching Control Policy** (Mguni et al., 2022). A soft actor-critic (SAC) (Haarnoja et al., 2018) agent called Global whose policy's action set consists of 2 actions: 1) use the centralised policy (perform CL updates), 2) do not use the centralised policy (perform IL updates). Global updates its policy $\mathfrak{g}$ while each agent learns their individual policy.

The MANSA framework includes a feature that enables it to restrict the CL updates to only when Global executes a CL call (i.e. when $g_t = 1$). In this way, communication occurs between the CL agents solely when Global performs a CL activation (no information in shared between IL and CL).

This ensures the communication burden between agents is strictly limited during training.

Note also 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).

4. Convergence and Optimality of MANSA

We now show that the MANSA framework, which induces an $N + 1$ *non-cooperative Markov game*, converges to the solution that both maximises the Global agent's value function and the Global agent's objective. With this, the Global agent learns to activate CL only at the set of states at which doing so improves the system performance of the MARL agents. The result is achieved through several steps: Theorem 1 shows MANSA learns the optimal solution for the Global agent so that it activates CL only when it is profitable to do so over the horizon of the problem (recall that each activation incurs a CL cost). Prop. 1 proves the MANSA framework leads to higher system performance as compared to training the underlying base MARL method on its own. Finally, we characterise the optimal CL activation points and show that Global can use a condition on its action-value function that can be evaluated online to determine when to activate CL (for the case when Global uses a Q-learning variant). All our results are built under Assumptions 1 - 7 (Sec. 15 of the Appendix) which are standard in RL and stochastic approximation theory.

To prove our first result, we now show that for a fixed set of joint IL and CL policies, the solution of Global's problem is a limit point of a sequence of Bellman operations acting on a value function. We then show that the system in which both the IL, CL and Global agents train concurrently within the MANSA framework converges to the solution.

Theorem 1. ***i)*** *Let $v_G : \mathcal{S} \to \mathbb{R}$ then for any fixed joint policies $\boldsymbol{\pi}^c, \boldsymbol{\pi} \in \boldsymbol{\Pi}$ the solution of Global's problem is given by $\lim_{k\to\infty} T_G^k v_G(\cdot|\boldsymbol{\pi}, \mathfrak{g}) = \max_{\hat{\mathfrak{g}}} v_G(\cdot|\boldsymbol{\pi}, \hat{\mathfrak{g}})$, where T_G is given by $T_G v_G := \max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c} Q_G, \max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left[R_G + \gamma \sum_{s'\in\mathcal{S}} P(s'; \cdot) v_G(s')\right]\right\}$ and $\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c} Q_G(s, \boldsymbol{a}|\cdot) := Q_G(s, \boldsymbol{\pi}^c(s)|\cdot) - c$ which measures the expected return for Global following a switch to the CL joint policy minus the intervention cost c.*

ii) *Given a system of convergent MARL learners of $\mathcal{M}$, MANSA ensures the convergence of the system $\mathcal{G}$ when Global uses a Q-learning variant.*

Therefore, Theorem 1 proves the solution to Global's problem in which Global optimally selects the set of states to activate CL can be obtained by computing the limit of a dynamic programming procedure. Secondly, it proves the MANSA system of $N + 1$ agents converges to the solution of $\mathcal{G}$ (when Global uses a Q-learning variant).

Next we show MANSA improves performance outcomes:

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

The result shows that using the MANSA framework leads to improvements in the underlying MARL algorithm (as compared to training the MARL algorithm on its own). Note that *a fortiori* Prop. 1 implies $v(s|\tilde{\boldsymbol{\pi}}) \geq v(s|\boldsymbol{\pi})$, $\forall s \in \mathcal{S}$.

The following result characterises Global's policy $\mathfrak{g}$:

Proposition 2. *For any $s_t \in \mathcal{S}$ and for all $\boldsymbol{a}_t \in \boldsymbol{\mathcal{A}}$, the policy $\mathfrak{g}$ is given by: $\mathfrak{g}(\cdot|s_t) = \mathbf{1}_{\mathbb{R}_+}\left(\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c} Q_G(s_t, \boldsymbol{a}_t|\cdot) - \max_{\boldsymbol{a}_t\in\boldsymbol{\mathcal{A}}} Q_G(s_t, \boldsymbol{a}_t|\boldsymbol{\pi}, \mathfrak{g})\right)$, where $\mathbf{1}$ is the indicator function.*

Prop. 2 provides characterisation of where Global should activate Central. The condition allows for the characterisation to be evaluated online during the learning phase.

5. MANSA with a CL Call Budget

So far we have considered the case in which the aim is to solve the problem $\mathfrak{M}$ while using the minimum number of CL calls. We now introduce a variant of MANSA, namely MANSA-B that aims to solve the problem while respecting a budgetary constraint of the number of allowed CL calls during training. We show that by tracking its remaining budget the MANSA-B framework is able to learn a policy that makes optimal usage of its CL budget while respecting the budget constraint almost surely.

The problem in which Global now faces a fixed budget on the number of CL calls gives rise to the following constrained problem setting:

$$\max_{\mathfrak{g}} v_G(s|\boldsymbol{\pi}, \mathfrak{g}) \;\text{ s. t. } n - \sum_{k<\infty} \sum_{t_k\geq 0} \mathbf{1}(\mathfrak{g}(\cdot|s_{t_k})) \geq 0, \forall s \in \mathcal{S},$$

where $n \geq 0$ is a fixed value that represents the budget for the number CL activations and the index $k = 1, \ldots$ represents the training episode count. As in (Sootla et al., 2022; Mguni et al., 2022), we introduce a new variable

x_t that tracks the remaining number of activations: $x_t := n - \sum_{t\geq 0} \mathbf{1}(\mathfrak{g}(s_t))$ where the variable x_t is now treated as the new state variable which is a component in an augmented state space $\mathcal{X} := \mathcal{S} \times \mathbb{N}$. We introduce the associated reward functions $\widetilde{R} : \mathcal{X} \times \boldsymbol{\mathcal{A}} \to \mathcal{P}(D)$ and $\widetilde{R}_G : \mathcal{X} \times \boldsymbol{\mathcal{A}} \to \mathcal{P}(D)$ and the probability transition function $\widetilde{P} : \mathcal{X} \times \boldsymbol{\mathcal{A}} \times \mathcal{X} \to [0, 1]$ whose state space input is now replaced by $\mathcal{X}$ and the Global value function for the game $\widetilde{\mathcal{G}} = \langle \mathcal{N} \times \{G\}, \mathcal{S}, ((\mathcal{A}_i)_{i\in\mathcal{N}}, \mathcal{A}_G), \tilde{P}, \tilde{R}, \tilde{R}_G, \gamma\rangle$. We now prove MANSA-B ensures maximal performance for a given number of CL calls (CL call budget).

Theorem 2. *Consider the budgeted cooperative problem $\widetilde{\mathcal{G}}$, then For any $\widetilde{v} : \mathcal{X} \to \mathbb{R}$, the solution of $\widetilde{\mathcal{G}}$ is given by $\lim_{k\to\infty} \tilde{T}_G^k \widetilde{v}^{\boldsymbol{\pi}} = \max_{\mathfrak{g}} \widetilde{v}^{\boldsymbol{\pi},\mathfrak{g}}$, where Global's optimal policy takes the Markovian form $\widetilde{\mathfrak{g}}(\cdot|\boldsymbol{x})$ for any $\boldsymbol{x} \equiv (x, s) \in \mathcal{X}$.*

Theorem 2 shows MANSA converges under standard assumptions to the solution of Global's problem (and the dec-POMDP) when Global faces a CL call budget constraint.

6. Experiments

We performed a series of experiments to test whether MANSA **1.** Enables MARL to solve multi-agent problems while reducing the number of CL calls **2.** Improves the performance of IL and reduces its failure modes **3.** Learns to optimise its use of CL under a CL call budget. We used the code accompanying the MARL benchmark study of Papoudakis et al. (2021) for the baselines. For these experiments, we tested MANSA in 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 cases to IL as well as a broad range of attributes as we describe below. 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, dark lines represent averages over 3 seeds and the shaded regions represent 95% confidence intervals.

Level-based Foraging (LBF). In LBF an agent controls units of particular levels and there are apples of particular levels scattered around the map. Each agent's 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 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. LBF has the option of enforcing coordination (map names suffixed with "coop") by making the food level such that at least two agents are required to coordinate to collect any food. LBF tasks are designed to sometimes require coordination to solve the problem, while other times needing little interaction between agents.

StarCraft Multi-Agent Challenge (SMAC). The goal in SMAC is for a team of units under an agent's control to defeat a team of units under an opponent's control. Different maps in SMAC vary along several dimensions including heterogeneity of units, number of units, and terrain. These differences result in agents having to adopt varying degrees of coordination to solve different maps. For example, in *so_many_baneling*, *zealots* under the agent's control face a larger army of enemy *banelings*. As banelings can cause significant 'splash' damage, it is critical for units under the agent's control to cooperate and space out so as to minimise damage. Conversely, in *corridor*, such cooperation may not be needed. Here, a small army of zealots under the agent's control face off against a large army of zerglings. The optimal strategy is for the zealots to wall-off a choke point and avoid getting surrounded. While it may seem that significant coordination is required to solve this map (i.e., all zealots converge to the choke point), in fact, it is not necessary. Due to location of the choke-point, the optimal actions for a zealot acting independently mirror those of a coordinated group – IL is as good as CL in this case. Thus, the design of SMAC sometimes befits IL algorithms and sometimes CL algorithms.

6.1. Can MANSA learn to use CL less frequently in settings where CL is not required? For this experiment, we first studied MANSA in two normal-form (matrix) games: a *coordination* game (specifically the Assurance Game) and the Non-Monotonic Team Game presented in Rashid et al. (2018). We modified the reward function of the Assurance Game with a parameter $\alpha \in [0, 1]$, as shown in Table 1. For $\alpha = 0$, the reward function degenerates to the reward function of the standard Assurance game, while for $\alpha = 1$, each agent gets a reward of 10 irrespective of the other agent's action, that is, the game is completely decoupled. Similarly, in the non-monotonic team game, α parameterises the degree to which the reward structure of the game is non-monotonic. In this modified game, $\alpha = 0$ represents a normal form game with a monotonic reward while $\alpha = 1$ represents a non-monotonic reward function.

	Up	Down
Up	$5(1+\alpha), 5(1+\alpha)$	$10\alpha, 10\alpha$
Down	$10\alpha, 10\alpha$	$10, 10$

	A	B
A	$2\alpha, 2\alpha$	1,1
B	1,1	8,8

Table 1. Modified reward functions of Assurance Game (*top*), and non-monotonic team game (*bottom*).

Fig. 3 shows plots of α versus the number of calls to CL. In both games, higher values of α ought to result in less

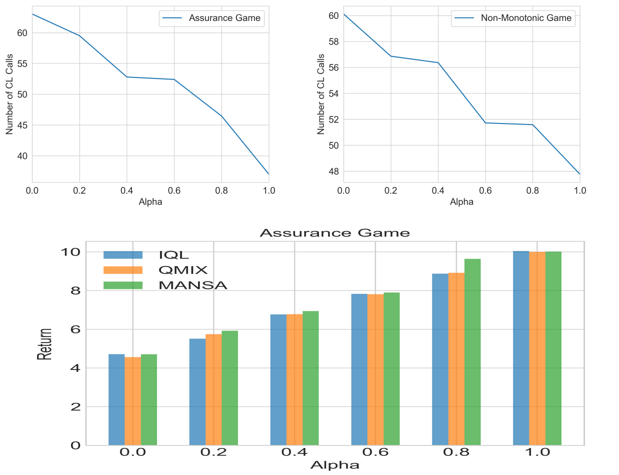


Figure 3. **Normal form games.** Total number of CL calls by MANSA in the Assurance Game (top left) and the non-monotonic team game (top right) and end-of-training returns for MANSA, QMIX and IQL for various values of α (bottom). As the rewards in Assurance Game become more decoupled ($\alpha \to 1$) so the requirement for coordination becomes weaker, MANSA reduces the number of CL calls it makes during training. In the Non-Monotonic game, as the extent of the monotonicity in the reward decreases ($\alpha \to$1), MANSA similarly reduces the number of CL (QMIX) calls. Note, in both cases MANSA makes a small number of calls to CL as Global initially explores both CL and IL. Despite MANSA reducing its dependence on CL as $\alpha \to 1$, it achieves returns that are better or the same as the baselines for all α.

usage of CL, and as expected, as α increases, calls to the CL decrease and MANSA shows greater dependence on IL for training. This suggests MANSA is capable of selectively using CL with a high degree of granularity. It also provides strong evidence MANSA exercises thriftiness in its usage of CL in environments with no strong coordination aspect.

We next investigated MANSA's ability modulate its use of CL in LBF Foraging-8x8-2p-2f-coop-v1, a dynamic setting with many states and agents. To do this, we isolated three configurations of the LBF task that have strongly, medium and weakly coupled reward functions i.e. for the agents to solve the task, each case requires a specific level of coordination by the agents. The weakest case is a setting in which each food item can be collected by just one agent; in the medium level, collecting each food item requires two agents to coordinate while in the strongest level, collecting each food item needs all agents to coordinate. For each case we measured the total number of CL calls made by MANSA over the course of training. As shown in Fig. 4, as the level of required coordination increases (from weak to strong), MANSA increases the number of CL calls to promote learning policies capable of coordination among the agents during their training phase.

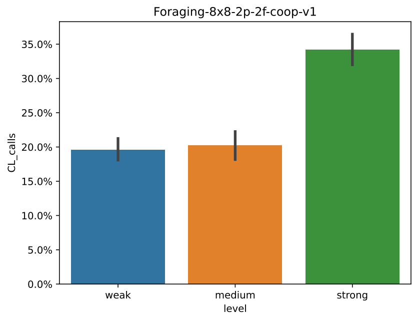


Figure 4. Number of CL calls within LBF maps with varying degrees of coupling within the reward functions.

Lastly, to confirm the usefulness of MANSA's switching control component, Section 9.2 of the Appendix gives an ablation study in which we replaced the Global agent with a random switching controller. compared to simply activating Central at random (line labelled ”random policy”). As is shown, removing MANSA's switching control aspect leads to significant degredation in overall performance as compared with MANSA with its adaptive RL agent Global.

6.2. Can MANSA improve the overall performance of IL and reduce failure modes? We first examined this claim in LBF; Fig. 5 shows aggregated (normalised) area under the curve AUC performance curves of the tested algorithms (for individual plots see Sec. 12 in the Appendix). MANSA outperforms both IQL and QMIX by a notable margin in half the maps (4 of 8). Moreover, even in maps where QMIX performs poorly, e.g., Foraging-10x10-3p-5f-v2, Foraging-10x10-5p-3f-v2, MANSA is able to use QMIX to significantly outperform IQL (compare performance of vanilla QMIX versus MANSA in plots in Sec. 12). This is due to MANSA correctly identifying states that benefit from CL (and those that do not) and there activating CL to achieve significant performance gains. The empirical results serve to validate MANSA's preservation of MARL convergence properties and its ability to leverage both CL and IL to deliver higher performance. In Sec. 9 of the Appendix, we show the results of an ablation study of the switching cost parameter. So long as the value of this hyper-parameter is roughly in the correct order of magnitude, MANSA per-

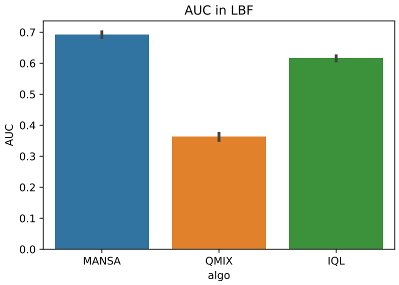


Figure 5. Aggregate (normalised) area under the curve (AUC) results across 10 LBF tasks. MANSA has superior aggregate performance, markedly outperforming the CL method (QMIX) and either matching or outperforming the IL method (IQL) on all tasks.

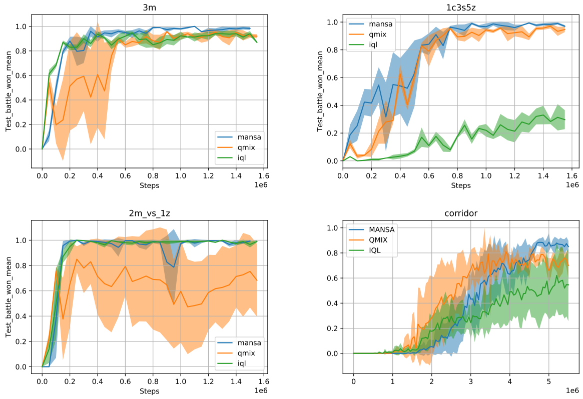


Figure 6. Learning curves in some individual SMAC maps. While QMIX fails to learn effective policies on all maps, and IQL on two maps, MANSA achieves high performance across the tasks.

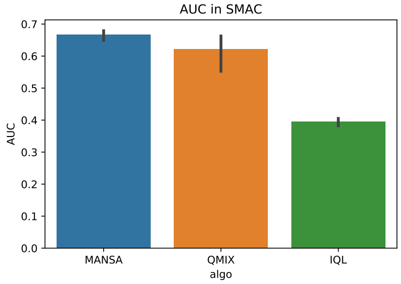


Figure 7. Aggregate normalised AUC results across 9 SMAC maps. MANSA has superior performance and is not susceptible to learning failures unlike the base CL (QMIX) and IL methods (IQL).

forms well and thus is easy to tune.

We next examined the claim in SMAC. Fig. 7 shows the aggregated normalised AUC results across a range of SMAC maps (for full set of plots of individual maps see Sec. 12 in the Appendix). MANSA's aggregate AUC performance is superior to both baselines. It also outperforms all baselines in all maps except *3s5z_vs_3s6z*. MANSA's flexibile choice of MARL method allows it to avoid the failures of IQL in maps such as *1c3s5z*, *3s5z*, *2s3z*, and *MMM2* without heavily relying on CL (MANSA's CL call rates are shown in Table 5 of the Appendix). Similarly, MANSA avoids the failures of QMIX in *2m_vs_1z* and *corridor*.

To validate the claim MANSA can reduce failure rates, we plotted the failure rates of each algorithm (i.e. on how many tasks each algorithm failed by the total number of tasks) in Fig. 8. We define a failure as achieving an end-of-training win rate of less than 0.8 on SMAC. IL and CL failed in 44% (4 of 9) and 22% (2 of 9), respectively, of the SMAC maps while MANSA did not fail at all.

MANSA is a Plug & Play IL Enhancement Framework. To validate our claim that MANSA easily adopts MARL algorithms, we ran further experiments with a stronger CL baseline to test if MANSA is still beneficial when the CL baseline is stronger than the IL baseline (see Sec. 10 in the Appendix). Even when the CL is a stronger algorithm, MANSA still outperforms the baselines by switching from IL to CL when CL is the better option. Moreover, as is explained in Sec 10, MANSA increases the number of CL queries when the underlying CL is stronger.

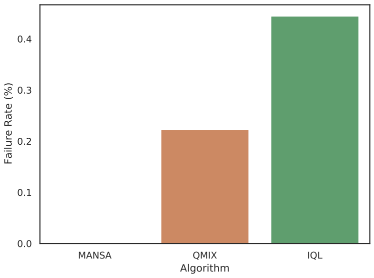


Figure 8. Failure rates (number of failed tasks/total number of tasks) of each algorithm across all SMAC maps.

Can MANSA optimise its use of CL under a CL budget? To validate our claim that MANSA-B optimises CL calls under a fixed budget, we ran MANSA-B in 4 SMAC maps with a varying CL call budget. Table 6 in the Appendix shows the Win rates comparing MANSA-B with various CL call budgets against MANSA (original). In 2 out of the 4 maps, MANSA achieves win rates of above 98% despite a cap of 10% on the original CL calls. As the budget increases to 50%, MANSA achieves above 65% win rates on all maps.

7. Conclusion

In this paper, we presented MANSA, a novel MARL framework tool for enhancing performance of IL training under a limited number of CL calls. MANSA combines IL and CL in a way that enables IL to leverage the benefits of CL while minimising the complexity burden and limitations of representational constraints suffered by CL methods. Similarly, MANSA mitigates the issues suffered by IL such as its inability to efficiently solve coordination tasks and lack of convergence guarantees. In so doing, MANSA provides a framework that leverages each algorithm class and removes the split between IL and CL MARL training methods. Our theoretical results show that MANSA preserves MARL convergence guarantees and improves outcomes for IL and our empirical analyses presents a detailed suite of experimental results on normal form games, LBF, and SMAC. In all these domains, MANSA improves performance, reduces failure modes all the meanwhile minimising its use of CL. In future work, we will consider the natural extension of the framework to encompass switching between various CL methods to leverage the benefits of their various factorisations (as well as IL). We envisage the ideas in this paper can be extended more broadly across machine learning wherever there exists a tradeoff between optimisation that needs expensive computation and expedience and inexpensiveness.

References

Daryin, A. and Kurzhanski, A. Nonlinear control synthesis under double constraints. *IFAC Proceedings Volumes*, 38 (1):247–252, 2005.

de Witt, C. S., Gupta, T., Makoviichuk, D., Makoviychuk, V., Torr, P. H., Sun, M., and Whiteson, S. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533*, 2020.

Deng, X., Li, Y., Mguni, D. H., Wang, J., and Yang, Y. On the complexity of computing markov perfect equilibrium in general-sum stochastic games. *arXiv preprint arXiv:2109.01795*, 2021.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. PMLR, 2018.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

Jaakkola, T., Jordan, M. I., and Singh, S. P. Convergence of stochastic iterative dynamic programming algorithms. In *Advances in neural information processing systems*, pp. 703–710, 1994.

Kok, J. R. and Vlassis, N. Sparse cooperative q-learning. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 61, 2004.

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

Lowe, R., Wu, Y. I., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in neural information processing systems*, pp. 6379–6390, 2017.

Macua, S. V., Zazo, J., and Zazo, S. Learning parametric closed-loop policies for markov potential games. *arXiv preprint arXiv:1802.00899*, 2018.

Mahajan, A., Rashid, T., Samvelyan, M., and Whiteson, S. Maven: Multi-agent variational exploration. *arXiv preprint arXiv:1910.07483*, 2019.

McAleer, S., Farina, G., Lanctot, M., and Sandholm, T. Escher: Eschewing importance sampling in games by computing a history value function to estimate regret. *arXiv preprint arXiv:2206.04122*, 2022.

Mguni, D. A viscosity approach to stochastic differential games of control and stopping involving impulsive control. *arXiv preprint arXiv:1803.11432*, 2018.

Mguni, D. Cutting your losses: Learning fault-tolerant control and optimal stopping under adverse risk. *arXiv preprint arXiv:1902.05045*, 2019.

Mguni, D., Jennings, J., and de Cote, E. M. Decentralised learning in systems with many, many strategic agents. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Mguni, D., Jennings, J., Macua, S. V., Sison, E., Ceppi, S., and de Cote, E. M. Coordinating the crowd: Inducing desirable equilibria in non-cooperative systems. *arXiv preprint arXiv:1901.10923*, 2019.

Mguni, D., Sootla, A., Ziomek, J., Slumbers, O., Dai, Z., Shao, K., and Wang, J. Timing is everything: Learning to act selectively with costly actions and budgetary constraints. *arXiv preprint arXiv:2205.15953*, 2022.

Mguni, D. H., Jafferjee, T., Wang, J., Perez-Nieves, N., Slumbers, O., Tong, F., Li, Y., Zhu, J., Yang, Y., and Wang, J. Ligs: Learnable intrinsic-reward generation selection for multi-agent learning. *arXiv preprint arXiv:2112.02618*, 2021a.

Mguni, D. H., Wang, J., Jafferjee, T., Perez-Nieves, N., Song, W., Tong, F., Chen, H., Zhu, J., Yang, Y., and Wang, J. Learning to shape rewards using a game of two partners. 2021b.

Papoudakis, G., Christianos, F., Schäfer, L., and Albrecht, S. V. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS)*, 2021.

Peng, B., Rashid, T., de Witt, C. A. S., Kamienny, P.-A., Torr, P. H., Böhmer, W., and Whiteson, S. Facmac: Factored multi-agent centralised policy gradients. *arXiv preprint arXiv:2003.06709*, 2020.

Qiu, D., Wang, J., Wang, J., and Strbac, G. Multi-agent reinforcement learning for automated peer-to-peer energy trading in double-side auction market. In *IJCAI*, pp. 2913–2920, 2021.

Qiu, D., Wang, J., Dong, Z., Wang, Y., and Strbac, G. Mean-field multi-agent reinforcement learning for peer-to-peer multi-energy trading. *IEEE Transactions on Power Systems*, 2022.

Rashid, T., Samvelyan, M., Schroeder, C., Farquhar, G., Foerster, J., and Whiteson, S. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 4295–4304. PMLR, 2018.

Rashid, T., Farquhar, G., Peng, B., and Whiteson, S. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning. *arXiv preprint arXiv:2006.10800*, 2020.

Samvelyan, M., Rashid, T., De Witt, C. S., Farquhar, G., Nardelli, N., Rudner, T. G., Hung, C.-M., Torr, P. H., Foerster, J., and Whiteson, S. The starcraft multi-agent challenge. *arXiv preprint arXiv:1902.04043*, 2019.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

Sharma, P. K., Fernandez, R., Zaroukian, E., Dorothy, M., Basak, A., and Asher, D. E. Survey of recent multi-agent reinforcement learning algorithms utilizing centralized training. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications III*, volume 11746, pp. 117462K. International Society for Optics and Photonics, 2021.

Shoham, Y. and Leyton-Brown, K. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E., and Yi, Y. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, pp. 5887–5896. PMLR, 2019.

Sootla, A., Cowen-Rivers, A. I., Jafferjee, T., Wang, Z., Mguni, D., Wang, J., and Bou-Ammar, H. SAUTE RL: Almost surely safe reinforcement learning using state augmentation, 2022.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K., et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

Tsitsiklis, J. N. and Van Roy, B. Optimal stopping of markov processes: Hilbert space theory, approximation algorithms, and an application to pricing high-dimensional financial derivatives. *IEEE Transactions on Automatic Control*, 44(10):1840–1851, 1999.

Wang, J., Ren, Z., Liu, T., Yu, Y., and Zhang, C. Qplex: Duplex dueling multi-agent q-learning. *arXiv preprint arXiv:2008.01062*, 2020.

Wang, J., Xu, W., Gu, Y., Song, W., and Green, T. C. Multi-agent reinforcement learning for active voltage control on power distribution networks. *Advances in Neural Information Processing Systems*, 34:3271–3284, 2021.

Yang, Y. and Wang, J. An overview of multi-agent reinforcement learning from game theoretical perspective. *arXiv preprint arXiv:2011.00583*, 2020.

Yu, C., Velu, A., Vinitsky, E., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of mappo in cooperative, multi-agent games. *arXiv preprint arXiv:2103.01955*, 2021.

Zhou, M., Luo, J., Villella, J., Yang, Y., Rusu, D., Miao, J., Zhang, W., Alban, M., Fadakar, I., Chen, Z., et al. Smarts: Scalable multi-agent reinforcement learning training school for autonomous driving. *arXiv preprint arXiv:2010.09776*, 2020.

Part I

Appendix

8. Assurance Game Construction

In Sec. 6.1 we introduce a variant of the classic coordination game, the Assurance game. The matrix game presented in Sec. 6.1 is the supposition of the Assurance Game with a non-strategic component resulting in a new game whose entries are the supposition of the above components. The composition of the new game is calibrated by a parameter α which runs from 0 to 1. At its extreme points 0 and 1, the game degenerates into the Assurance game and the entirely non-strategic game. We begin by stating the reward function $R_i(a_i, a_j) : \mathcal{A}_i \times \mathcal{A}_j \to \mathbb{R}$ for the agents $i, j \in \{1, 2\}$.

$$R_i(a_i, a_j) = \alpha(\mathcal{R}_i(a_i) + \mathcal{R}_j(a_j)) + (1-\alpha)\mathfrak{R}_i(a_i, a_j), \quad i, j \in \{1, 2\}, \tag{1}$$

where $R_i : \mathcal{A}_i \to \mathbb{R}$ and $R_j : \mathcal{A}_j \to \mathbb{R}$ are bounded, real valued functions and $\mathcal{A}_i$ and $\mathcal{A}_j$ are compact sets.

We assume $\mathfrak{R}_i$ in (1) can't be decoupled into a function of the form $\mathfrak{R}_i(a_i, a_j) = f(a_i) + g(a_j)$. From (1), we see that when $\alpha = 0$, $R_i(a_i, a_j) = \mathfrak{R}_i(a_i, a_j)$ meaning that the game is strongly coupled and that as $\alpha \to 1$, $R_i(a_i, a_j) \to \mathcal{R}_i(a_i) + \mathcal{R}_j(a_j)$ meaning that the game is decoupled (the agents have no effect on other agents' rewards).

In what follows, the payoff matrix of $\mathfrak{R}_i(a_i, a_j)$ is denoted by A: This represents the coupled part of the reward in (1), i.e. $\mathfrak{R}_i(a_i, a_j)$. We construct a second matrix corresponding to the independent part of the reward in (1) and denote this matrix by B. Notice in this payoff matrix, the actions of the other agents have no effect on the agent's own reward (whenever an agent plays an action its reward is identical regardless of the other agents action). Thus, to construct the matrix game corresponding to (1), we simply compute the weighted sum entry-wise. Denote this by C. Now we vary the value of α within the interval $[0, 1]$ and plot the number of CL calls used during training (this is the number of times the Global agent performs a switch) vs the value of α.

$A =$

	Up	Down
Up	5,5	0,0
Down	0,0	10,10

$B =$

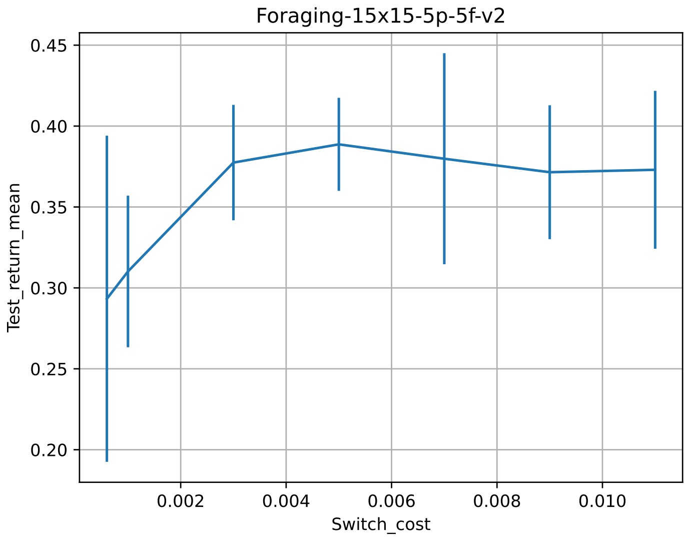
	Up	Down
Up	10,10	10,10
Down	10,10	10,10

$C =$

	Up	Down
Up	$5(1+\alpha), 5(1+\alpha)$	$10\alpha, 10\alpha$
Down	$10\alpha, 10\alpha$	$10, 10$

9. Ablation studies

9.1. A.1 Switching Cost Parameter



We ran a simple study to ascertain the sensitivity of MANSA to the *switching cost* parameter. We picked a random map in LBF and ran MANSA with a range of values for the switching cost. As shown in the plot, while within a given order of magnitude (here 10^{-2}), MANSA's performance is largely robust to the switching cost. However, there is a deterioration in performance if the switching cost is set too low, and thereby does not penalize usage of CL enough.

9.2. Importance of Switching Controls

A key component of MANSA is the switching control mechanism. This enables the Global agent to select the states in which activating Central leads to performance improvements. To evaluate the impact of the switching control component, we compared the performance of MANSA with a version of MANSA which has the switching control replaced with an equal-chances Bernoulli Random Variable (i.e., at any given state, the Global decides whether or not to activate Central with equal probability) (note that always activating Central degenerates to QMIX and similarly, never activating Central degenerates to IQL). Figure 9 shows the comparison of the performances of the variants. We examined the performance of the variants of MANSA in the LBF task described in Section 7. As can be seen in the plot, incorporating the ability to learn an optimal switching controls in MANSA (labelled ”MANSA (OW-QMIX+IQL”) leads to much better overall performance compared to simply activating Central at random (line labelled ”random_policy”).

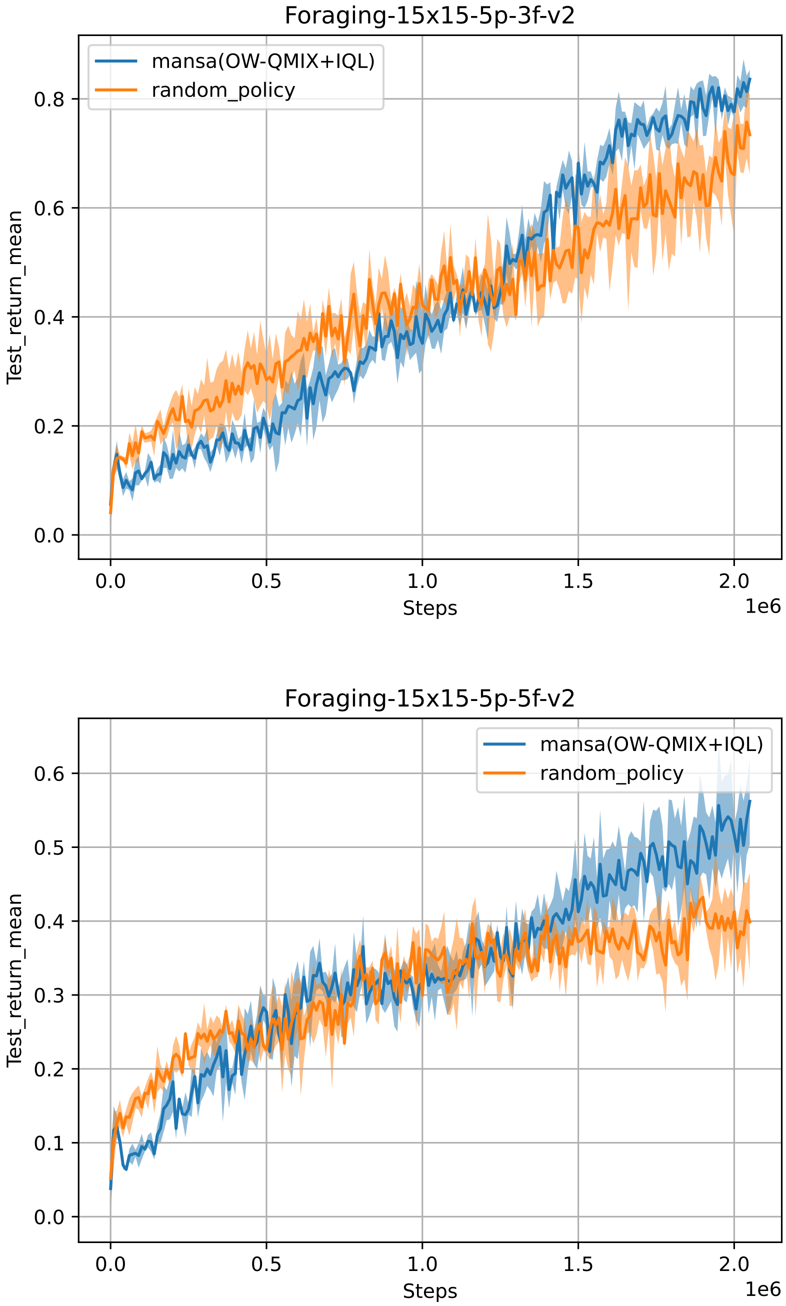
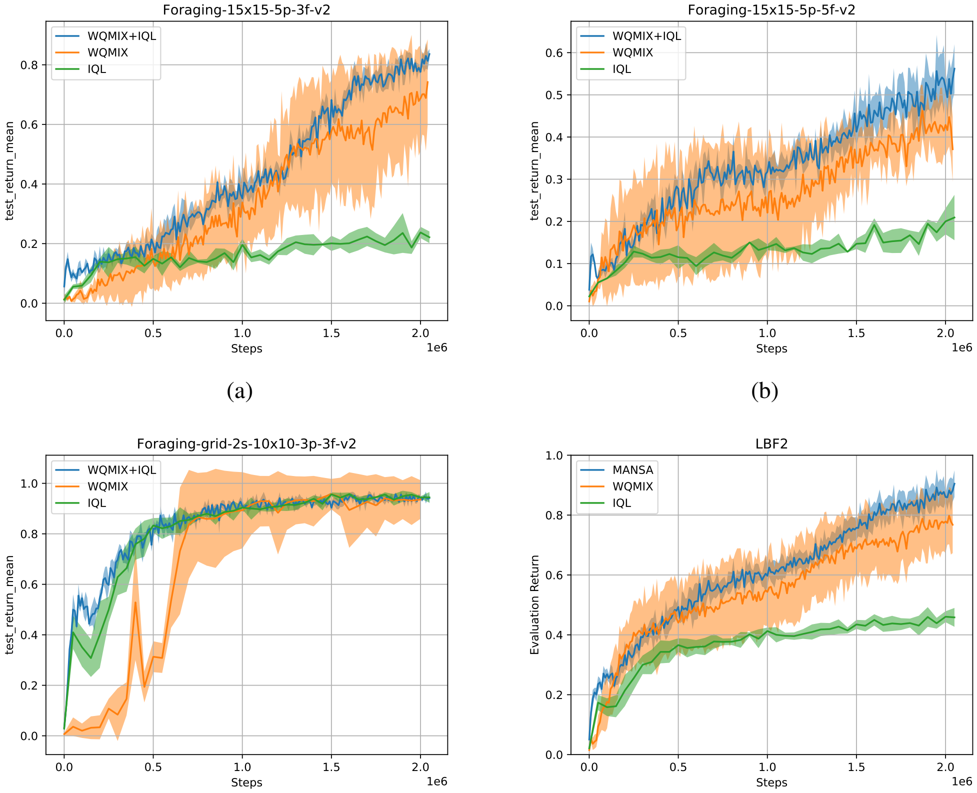

Foraging-15x15-5p-3f-v2
mansa(OW-QMIX+IQL)
random_policy
Test_return_mean
0.0
0.2
0.4
0.6
0.8
Steps
0.0
0.5
1.0
1.5
2.0
1e6
Foraging-15x15-5p-5f-v2
mansa(OW-QMIX+IQL)
random_policy
Test_return_mean
0.0
0.1
0.2
0.3
0.4
0.5
0.6
Steps
0.0
0.5
1.0
1.5
2.0
1e6

Figure 9.

10. MANSA is a Plug & Play Enhancement Tool



(a)

(b)

(c)

(d)

Figure 10. Learning curves on LBF with a stronger CL algorithm, W-QMIX. W-QMIX outperforms IQL on the selected maps, but MANSA is still able to leverage the advantages of both IL and CL to outperform the baselines.

Table 3. Percentage of CL calls in MANSA on the maps shown in 10.

LBF Map	Percentage of CL (W-QMIX) calls
Foraging-15x15-5p-3f-v2	53.15%
Foraging-15x15-5p-5f-v2	87.91%
Foraging-grid-2s-10x10-3p-3f-v2	5.86%

In this experiment, we replaced QMIX in MANSA with a stronger CL component, W-QMIX, to test if MANSA is able to successfully delivery performance benefits even if the CL baseline is stronger than the IL baseline. Figure 10 shows learning curves where, unlike in Figure 5, IQL is outperformed by a CL algorithm, W-QMIX. For MANSA to achieve reasonable performance here, the switching controller ought to opt to use CL more frequently than IL even if this incurs a switching cost. Indeed, we see that in all maps, MANSA significantly outperforms the baselines, and from Table 3 we see that MANSA uses CL much more in these maps than the maps indicated in Table 4. Moreover, as with previous experiments, MANSA seems to have correctly identified states that benefit from CL (and those that do not) and only utilising CL ther to achieve significant performance gains.

11. MANSA CL Call Analysis

11.1. MANSA CL Calls

One of our key claims is that MANSA reduces the number of CL calls made during training. In this section, we present the CL call percentages made by MANSA in both Level-based foraging (LBF) and StarCraft Multi-Agent Challenge (SMAC). In the case of LBF, MANSA successfully solves the tasks (recall that MANSA outperforms all baselines in all SMAC maps except *3s5z_vs_3s6z* and MANSA outperforms both IQL and QMIX by a notable margin in most of the maps (4 of 8) in LBF, see Section 12 for detailed performance plots).

LBF Map	Percentage of CL calls
Foraging-8x8-3p-3f-v2	4.76%
Foraging-10x10-3p-5f-v2	5.23%
Foraging-10x10-5p-3f-v2	1.85%
Foraging-15x15-5p-5f-v2	0.76%
Foraging-5x5-2p-1f-coop-v2	3.46%
Foraging-8x8-2p-2f-coop-v2	16.90%
Foraging-10x10-5p-1f-coop-v2	0.71%
Foraging-10x10-8p-1f-coop-v2	0.16%

Table 4. Percentage of calls to CL in MANSA in LBF.

SMAC Map	Percentage of CL calls
1c3s5z	81.67%
2m_vs_1z	69.01%
2s3z	79.70%
3m	59.22%
3s5z	82.19%
8m	62.96%
corridor	80.19%
MMM2	80.78%
so_many_baneling	74.83%

Table 5. Percentage of calls to CL in MANSA in SMAC.

11.2. MANSA-B CL calls under Budgetary Constraints

End-of-training win-rates of MANSA-B under various CL call budget constraints. Here, the percentages shown on the top row indicate CL calls proportionate to the number of CL calls that was made by MANSA (blue), e.g., 10% means we only allow MANSA-B total number of CL calls equal to 10% of the calls of MANSA. The performance of QMIX (orange) and IQL (green) are also shown for reference. In this table we see further evidence of MANSA's remarkably granular control over using CL. In the map *so_many_baneling*(which as we described in Section 6 requires high levels of coordination), performance improves with each budget increment of CL calls.

	Original/ QMIX/IQL	10%	20%	50%	75%
3m	98.00 ± 1.00 92.00 ± 1.63 87.00 ± 0.82	98.00 ±1.00	97.67 ±1.15	99.33 ±0.58	99.00 ±1.00
2s3z	97.00 ± 2.00 96.33 ± 0.58 77.67 ± 5.86	92.33 ±5.13	96.00 ±3.46	90.00 ±5.29	96.33 ±0.57
2m vs 1z	99.00 ± 0.00 68.33±28.75 99.00 ± 0.82	100.00 ±0.00	100.00 ±0.00	100.00 ±0.00	99.67 ±1.00
so many baneling	97.00 ± 1.00 86.00 ± 3.61 92.34 ± 5.03	84.50 ±11.84	98.00 ±2.51	95.50 ±2.64	93.50 ±5.13

Table 6. End-of-training win-rates of MANSA-B under various CL call budget constraints.

12. Detailed Performance Plots

12.1. Level-Based Foraging

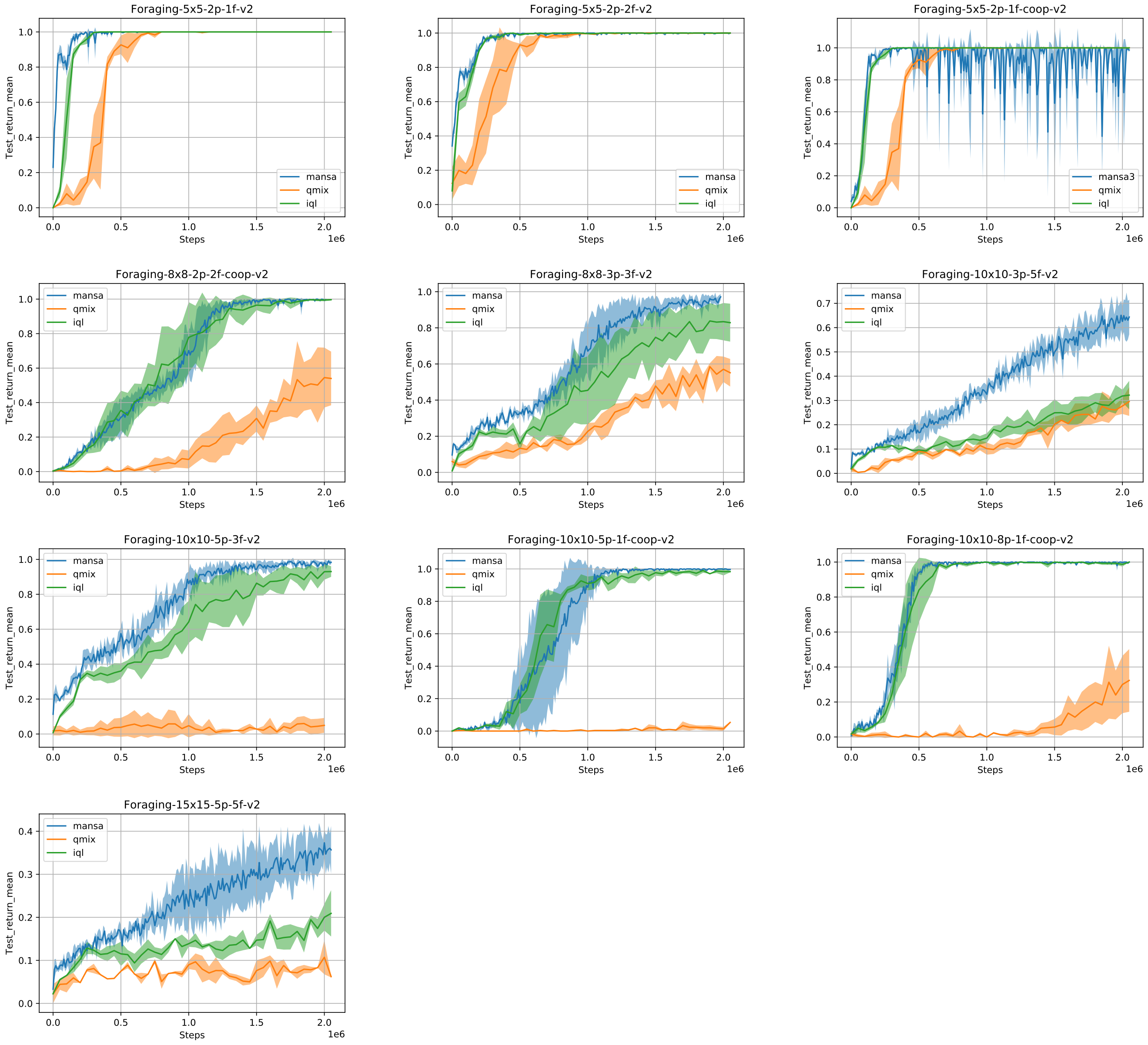


Figure 11. Learning curves on individual LBF maps.

12.2. StarCraft Multi-Agent Challenge

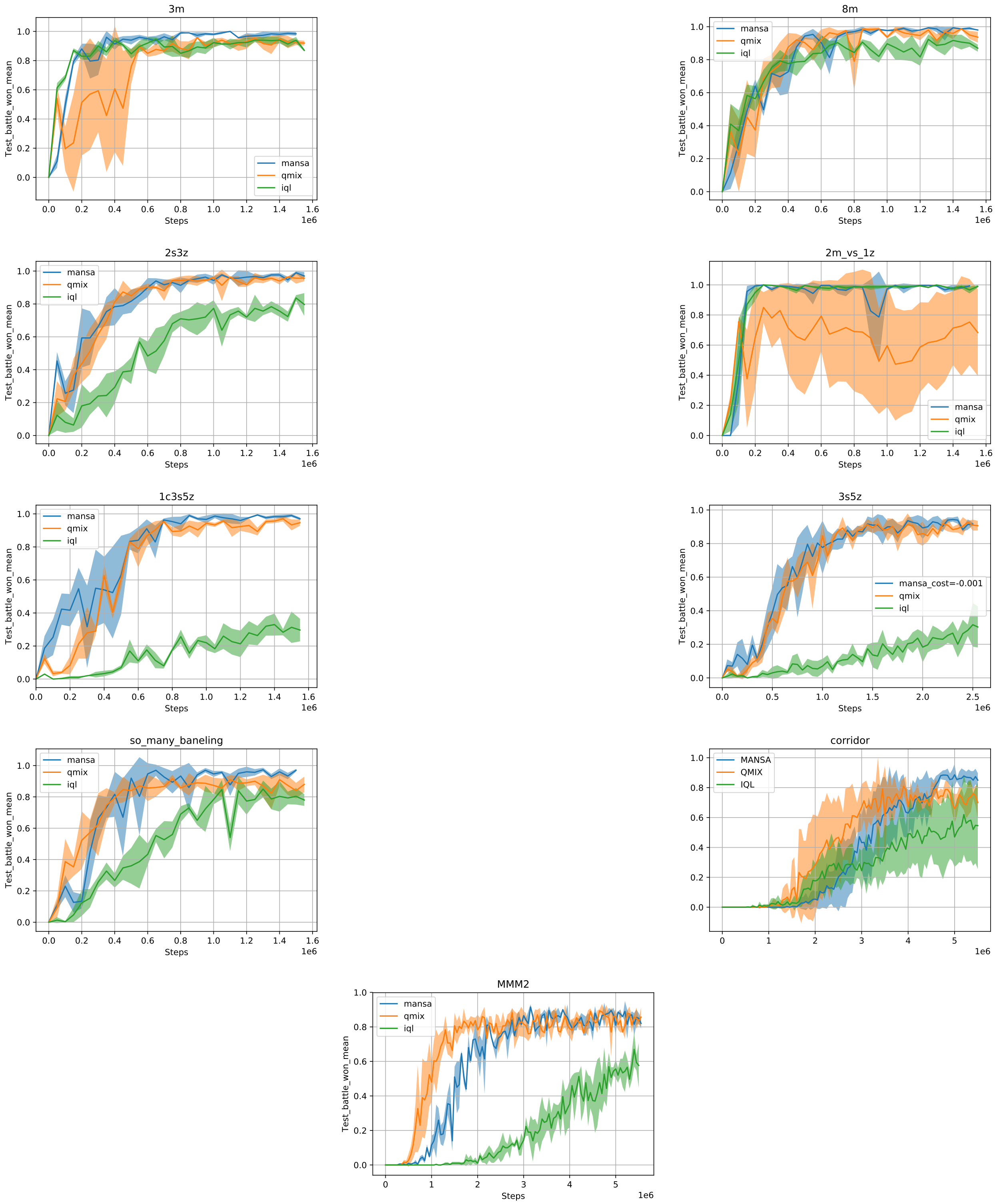


Figure 12. Learning curves on individual SMAC maps. While QMIX fail to learn effective policies on two maps, and IQL fails on four maps, MANSA does not exhibit any failure cases. We define failure as achieving a win-rate of less than 80%.

13. MANSA with CL Update Restriction

MANSA includes a feature that imposes the condition that CL updates can only occur when the Global agent makes a CL call (i.e. when $g = 1$). In this section we provide training plots display the results for MANSA with this CL training restriction (MANSA_CLR) against the baselines. As before, MANSA_CLR substantially outperforms the baselines on all tested LBF tasks. Similarly, in SMAC, MANSA_CLR outperforms the baselines on the majority tasks and matches their performance on others.

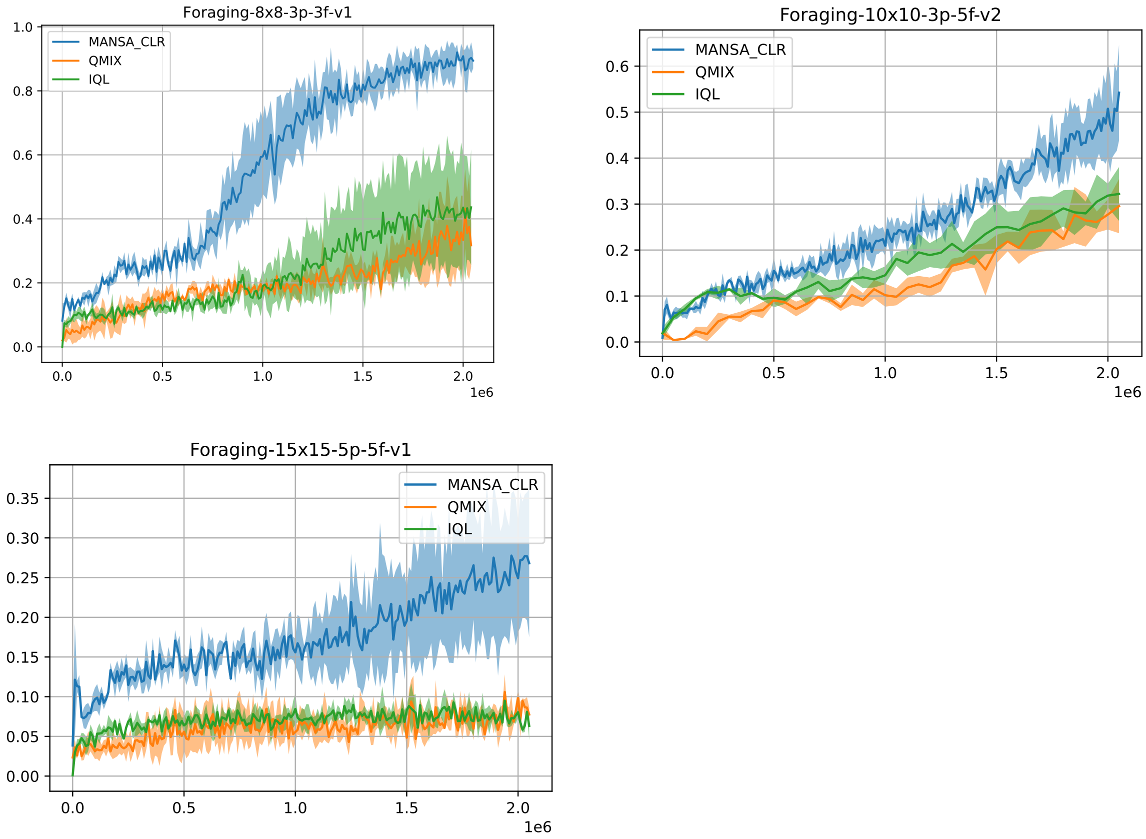


Figure 13. End-of-training returns of MANSA with CL update restriction (MANSA_CLR) in Level-Based Foraging (LBF).

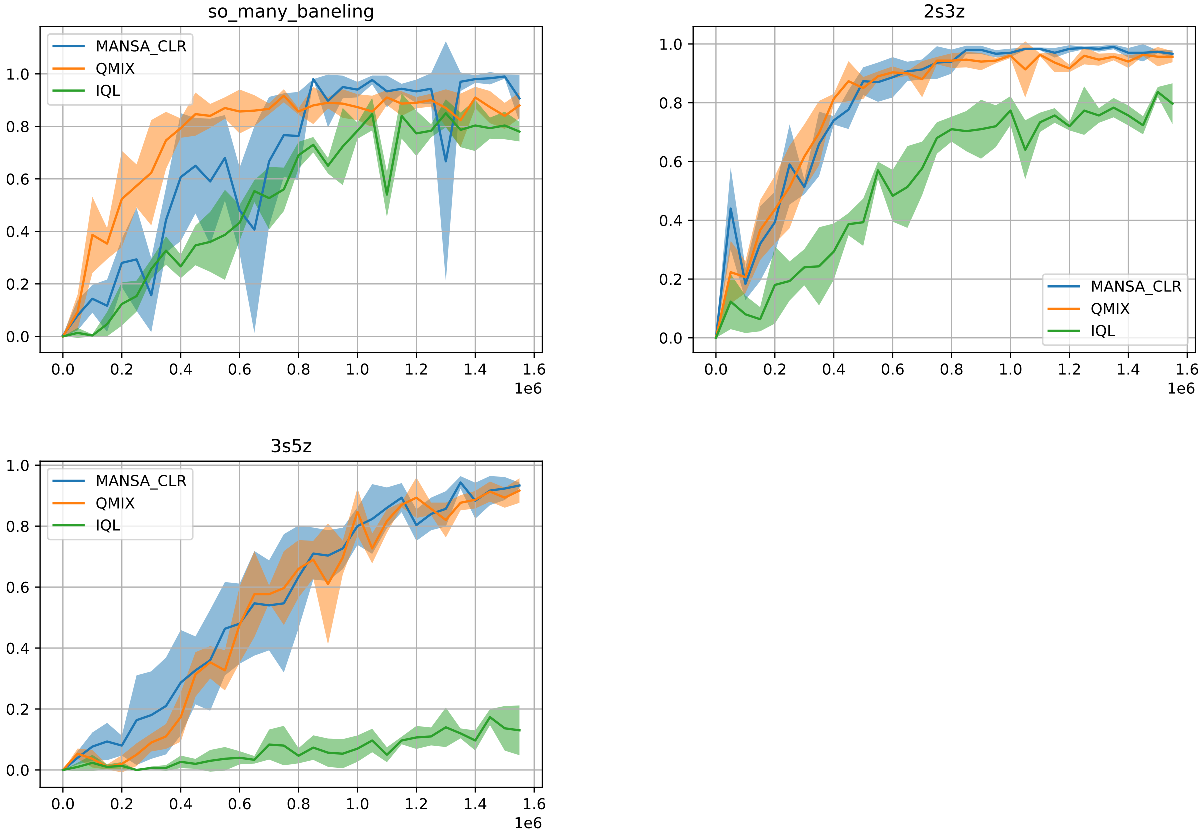


Figure 14. End-of-training win-rates of MANSA with implementation with CL update restriction (MANSA_CLR) in StarCraft Multi-Agent Challenge (SMAC).

13.1. MANSA-Budget with CL Update Restriction

In Section 13. we provided results for MANSA_CLR which imposes the restriction that CL updates can only occur when the Global agent makes a CL call (i.e. when $g = 1$). In this section, Table 7 displays the results for MANSA-B_CLR which imposes the CL update restriction on the MANSA-B framework (i.e. MANSA which has a budget constraint on the number of CL calls). As before, MANSA_CLR outperforms IQL when given a budget of just 20% CL calls and outperforms QMIX on 2m_vs_1z with just a 10% CL budget. In 2s3z MANSA outperforms QMIX when it has a budget of 75% for its CL calls; i.e. it outperforms QMIX even though it is forced to make 25% fewer CL calls than QMIX.

	Original/QMIX/IQL	**10%**	**20%**	**50%**	**75%**
2m_vs_1z	98.00 ± 1.00 92.00 ± 1.63 87.00 ± 0.82	100.00 ±0.00	99.67 ±0.57	96.67 ±3.05	99.00 ±0.00
2s3z	96.67 ± 1.24 95.67 ± 1.8 79.67 ± 6.69	82.00 ±1.41	82.33 ±5.18	81.67 ±1.69	96.33 ±0.47

Table 7. End-of-training win-rates of MANSA-B with CL update restriction and various CL call budget constraints against baselines.

14. Hyperparameter Settings

In the table below we report all hyperparameters used in our experiments. Hyperparameter values in square brackets indicate ranges of values that were used for performance tuning.

Clip Gradient Norm	1
γ_E	0.99
λ	0.95
Learning rate	$1\text{x}10^{-4}$
Number of minibatches	4
Number of optimisation epochs	4
Number of parallel actors	16
Optimisation algorithm	Adam
Rollout length	128
Sticky action probability	0.25
Use Generalized Advantage Estimation	True
Coefficient of extrinsic reward	[1, 5]
Coefficient of intrinsic reward	[1, 2, 5, 10, 20, 50]
Global discount factor	0.99
Probability of terminating option	[0.5, 0.75, 0.8, 0.9, 0.95]
L function output size	[2, 4, 8, 16, 32, 64, 128, 256]

15. Notation & Assumptions

We assume that $\mathcal{S}$ is defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and any $s \in \mathcal{S}$ is measurable with respect to the Borel σ-algebra associated with $\mathbb{R}^p$. We denote the σ-algebra of events generated by $\{s_t\}_{t\geq 0}$ by $\mathcal{F}_t \subset \mathcal{F}$. In what follows, we denote by $(\mathcal{Y}, \|\|)$ any finite normed vector space and by $\mathcal{H}$ the set of all measurable functions. Where it will not cause confusion (and with a minor abuse of notation) for a given function h we use the shorthand $h^{(\pi^i,\pi^{-i})}(s) = h(s, \pi^i, \pi^{-i}) \equiv \mathbb{E}_{\pi^i,\pi^{-i}}[h(s, a^i, a^{-i})]$.

The results of the paper are built under the following assumptions which are standard within RL and stochastic approximation methods:

Assumption 1 The stochastic process governing the system dynamics is ergodic, that is the process is stationary and every invariant random variable of $\{s_t\}_{t\geq 0}$ is equal to a constant with probability 1.

Assumption 2 The agents' reward function R is in L_2.

Assumption 3 For any positive scalar c, there exists a scalar μ_c such that for all $s \in \mathcal{S}$ and for any $t \in \mathbb{N}$ we have: $\mathbb{E}\left[1 + \|s_t\|^c | s_0 = s\right] \leq \mu_c(1 + \|s\|^c)$.

Assumption 4 There exists scalars C_1 and c_1 such that $|R(s, \cdot)| \leq C_2(1 + \|s\|^{c_2})$ for some scalars c_2 and C_2 we have that: $\sum_{t=0}^{\infty} \left|\mathbb{E}\left[R(s_t, \cdot)|s_0 = s\right] - \mathbb{E}[R(s_0, \cdot)]\right| \leq C_1 C_2(1 + \|s_t\|^{c_1 c_2})$.

Assumption 5 There exists scalars e and E such that for any $s \in \mathcal{S}$ we have that: $|R(s, \cdot)| \leq E(1 + \|s\|^e)$.

Assumption 6 For any Global policy $\mathfrak{g}$, the total number of interventions is $K < \infty$.

16. Proof of Technical Results

We begin the analysis with some preliminary results and definitions required for proving our main results.

Definition 1. *A.1 Given a norm $\|\cdot\|$, an operator $T:\mathcal{Y}\to\mathcal{Y}$ is a contraction if there exists some constant $c\in[0,1[$ for which for any $J_1,J_2\in\mathcal{Y}$ the following bound holds: $\|TJ_1-TJ_2\|\leq c\|J_1-J_2\|$.*

Definition 2. *A.2 An operator $T:\mathcal{Y}\to\mathcal{Y}$ is non-expansive if $\forall J_1,J_2\in\mathcal{Y}$ the following bound holds: $\|TJ_1-TJ_2\|\leq\|J_1-J_2\|$.*

Lemma 1. *(Mguni, 2019) For any $f:\mathcal{Y}\to\mathbb{R}:\mathcal{Y}\to\mathbb{R}$, we have that the following inequality holds:*

$$\left\|\max_{a\in\mathcal{Y}}f(a)-\max_{a\in\mathcal{Y}}g(a)\right\|\leq\max_{a\in\mathcal{Y}}\|f(a)-g(a)\|. \tag{2}$$

Lemma 2. *A.4(Tsitsiklis & Van Roy, 1999) The probability transition kernel P is non-expansive so that if $\forall J_1,J_2\in\mathcal{Y}$ the following holds: $\|PJ_1-PJ_2\|\leq\|J_1-J_2\|$.*

Proof of Theorem 1

Proof. The proof of the Theorem proceeds by first proving that for any two fixed set of joint policies $\boldsymbol{\pi}^d,\boldsymbol{\pi}^c\in\boldsymbol{\Pi}$, the Global agent's learning process, which involves switching controls converges. Recall, that the Global agent presides over an activation that deactivates $\boldsymbol{\pi}^d$ and activates $\boldsymbol{\pi}^c$.

Prove that the solution to Markov Team games (that is games in which both players maximise *identical objectives*) in which one of the players uses switching control is the limit point of a sequence of Bellman operators (acting on some test function)

Therefore, the scheme of the proof is summarised with the following steps:

A) Prove that for any fixed Central and Decentral policies $\boldsymbol{\pi}^c$ and $\boldsymbol{\pi}^d$, Global's switching control policy converges to a solution of Global's problem.

B) Prove that the MG $\mathcal{G}$ has a dual representation as a *Markov Team Game* whose solution is obtained by computing the solution of a team Markov game.

C) Prove that all agents solve the same problem.

We begin by recalling the definition of the intervention operator $\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}$ for any $s\in\mathcal{S}$ and for a given $\boldsymbol{\pi}^c$:

$$\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s,\boldsymbol{a}|\cdot):=Q_G(s,\boldsymbol{\pi}^c(s)|\cdot)-c \tag{3}$$

Secondly, recall that the Bellman operator for the game $\mathcal{G}$ is given by:

$$T_g v_G(s_{\tau_k}):=\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_{\tau_k},\boldsymbol{a}),\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left[R_G(s_{\tau_k},\boldsymbol{a},g)+\gamma\sum_{s'\in\mathcal{S}}P(s';\boldsymbol{a},s_{\tau_k})v_G(s')\right]\right\} \tag{4}$$

To prove (i) it suffices to prove that T is a contraction operator. Thereafter, we use both results to prove the existence of a fixed point for $\mathcal{G}$ as a limit point of a sequence generated by successively applying the Bellman operator to a test value function. Therefore our next result shows that the following bounds holds:

Lemma 3. *The Bellman operator T is a contraction so that the following bound holds: $\|T\psi-T\psi'\|\leq\gamma\|\psi-\psi'\|$.*

In the following proofs we use the following notation: $\mathcal{P}^{\boldsymbol{a}}_{ss'}=:\sum_{s'\in\mathcal{S}}P(s';\boldsymbol{a},s)$ and $\mathcal{P}^{\boldsymbol{\pi}}_{ss'}=:\sum_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\boldsymbol{\pi}(\boldsymbol{a}|s)\mathcal{P}^{\boldsymbol{a}}_{ss'}$.

To prove that T is a contraction, we consider the three cases produced by (4), that is to say we prove the following statements:

i) $$\left|\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_t,\boldsymbol{a},g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_t}v_G(s')\right)-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_t,\boldsymbol{a},g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_t}v'_G(s')\right)\right|\leq\gamma\|v_G-v'_G\|$$

ii) $$\left\|\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G - \mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q'_G\right\| \leq \gamma \left\|v_G - v'_G\right\|, \qquad .$$

iii) $$\left\|\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G - \max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left[R_G(s_t,\boldsymbol{a},g) + \gamma\mathcal{P}^{\boldsymbol{a}}v'_G\right]\right\| \leq \gamma \left\|v_G - v'_G\right\|.$$

We begin by proving i).

Indeed, for any $\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ and $\forall s_t \in \mathcal{S}, \forall s' \in \mathcal{S}$ we have that

$$\begin{aligned}
&\left|\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left(R_G(s_t,\boldsymbol{a},g) + \gamma\mathcal{P}^{\pi}_{s's_t}v_G(s')\right) - \max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left(R_G(s_t,\boldsymbol{a},g) + \gamma\mathcal{P}^{\boldsymbol{a}}_{s's_t}v'_G(s')\right)\right| \\
&\leq \max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left|\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_t}v_G(s') - \gamma\mathcal{P}^{\boldsymbol{a}}_{s's_t}v'_G(s')\right| \\
&\leq \gamma \left\|Pv_G - Pv'_G\right\| \\
&\leq \gamma \left\|v_G - v'_G\right\|,
\end{aligned}$$

using the non-expaniveness of the operator P and Lemma 1.

We now prove ii). Using the definition of $\mathcal{M}$ we have that for any $s_\tau \in \mathcal{S}$

$$\begin{aligned}
&\left|(\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G - \mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q'_G)(s_\tau,\boldsymbol{a}_\tau)\right| \\
&= \left|R_G(s_\tau,\boldsymbol{\pi}^c,g) - c + \gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{\pi}^c}v_G(s_\tau) - \Big(R_G(s_\tau,\boldsymbol{\pi}^c,g) - c + \gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{\pi}^c}v'_G(s_\tau)\Big)\right| \\
&\leq \max_{\boldsymbol{a}_\tau,g\in\boldsymbol{\mathcal{A}}\times\{0,1\}}\left|R_G(s_\tau,\boldsymbol{a}_\tau,g) - c + \gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s_\tau) - \left(R_G(s_\tau,\boldsymbol{a}_\tau,g) - c + \gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v'_G(s_\tau)\right)\right| \\
&= \gamma\max_{\boldsymbol{a}_\tau,g\in\boldsymbol{\mathcal{A}}\times\{0,1\}}\left|\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s_\tau) - \mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v'_G(s_\tau)\right| \\
&\leq \gamma \left\|Pv_G - Pv'_G\right\| \\
&\leq \gamma \left\|v_G - v'_G\right\|,
\end{aligned}$$

using the fact that P is non-expansive. The result can then be deduced easily by applying max on both sides.

We now prove iii). We split the proof of the statement into two cases:

Case 1: First, assume that for any $s_\tau \in \mathcal{S}$ and $\forall\boldsymbol{a} \in \boldsymbol{\mathcal{A}}$ the following inequality holds:

$$\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a}) - \max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}} \left(R_G(s_\tau,\boldsymbol{a}_\tau,g) + \gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right) < 0. \tag{5}$$

We now observe the following:

$$
\begin{aligned}
&\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\\
&\leq\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\\
&\leq\Bigg|\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}\\
&\qquad-\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}\\
&+\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\Bigg|\\
&\leq\Bigg|\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}\\
&\qquad-\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}\Bigg|\\
&\qquad\quad+\left|\max\left\{\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right),\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})\right\}-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\right|\\
&\leq\gamma\max_{a\in\mathcal{A}}\left|\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s')-\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v'_G(s')\right|+\left|\max\left\{0,\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\right\}\right|\\
&\leq\gamma\left\|Pv_G-Pv'_G\right\|\\
&\leq\gamma\|v_G-v'_G\|,
\end{aligned}
$$

where we have used the fact that for any scalars a,b,c we have that $|\max\{a,b\}-\max\{b,c\}|\leq|a-c|$ and the non-expansiveness of P.

Case 2: Let us now consider the case:

$$
\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\geq 0.
$$

For this case, first recall that $c>0$, hence

$$
\begin{aligned}
&\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\\
&\leq\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G(s_\tau,\boldsymbol{a})-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)+c\\
&\leq\left(R_G(s_\tau,\boldsymbol{a},g)-c+\gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s')\right)|^{\boldsymbol{a}\sim\boldsymbol{\pi}^c}-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)-c+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\\
&\leq\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a},g)-c+\gamma\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}v_G(s')\right)-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left(R_G(s_\tau,\boldsymbol{a}_\tau,g)-c+\gamma\mathcal{P}^{\boldsymbol{a}}_{s's_\tau}v'_G(s')\right)\\
&\leq\gamma\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left|\mathcal{P}^{\boldsymbol{\pi}}_{s's_\tau}\mathcal{P}^{\boldsymbol{a}}\left(v_G(s')-v'_G(s')\right)\right|\\
&\leq\gamma\left|v_G(s')-v'_G(s')\right|\\
&\leq\gamma\left\|v_G-v'_G\right\|,
\end{aligned}
$$

using the non-expansiveness of the operator P. Hence we have that

$$
\left\|\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G-\max_{\boldsymbol{a}\in\boldsymbol{\mathcal{A}}}\left[R_G(\cdot,\boldsymbol{a})+\gamma\mathcal{P}^{\boldsymbol{a}}v'_G\right]\right\|\leq\gamma\left\|v_G-v'_G\right\|. \tag{6}
$$

Gathering the results of the three cases gives the desired result.

To prove the theorem, we make use of the following result:

Theorem 3 (Theorem 1, pg 4 in (Jaakkola et al., 1994))**.** *Let $\Xi_t(s)$ be a random process that takes values in $\mathbb{R}^n$ and given by the following:*

$$\Xi_{t+1}(s) = (1 - \alpha_t(s))\,\Xi_t(s)\alpha_t(s)L_t(s), \tag{7}$$

then $\Xi_t(s)$ converges to 0 with probability 1 under the following conditions:

i) $0 \leq \alpha_t \leq 1, \sum_t \alpha_t = \infty$ *and* $\sum_t \alpha_t < \infty$

ii) $\|\mathbb{E}[L_t|\mathcal{F}_t]\| \leq \gamma\|\Xi_t\|$, *with* $\gamma < 1$*;*

iii) $\text{Var}\left[L_t|\mathcal{F}_t\right] \leq c(1 + \|\Xi_t\|^2)$ *for some* $c > 0$*.*

Proof. To prove the result, we show (i) - (iii) hold. Condition (i) holds by choice of learning rate. It therefore remains to prove (ii) - (iii). We first prove (ii). For this, we consider our variant of the Q-learning update rule:

$$\begin{aligned}Q_{t+1}(s_t,\boldsymbol{a}_t) &= Q_t(s_t,\boldsymbol{a}_t)\\ &\quad + \alpha_t(s_t,\boldsymbol{a}_t)\left[\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q(s_{\tau_k},\boldsymbol{a}), R(s_{\tau_k},\boldsymbol{a},g) + \gamma\max_{\boldsymbol{a'}\in\boldsymbol{\mathcal{A}}} Q_G s_{t+1},\boldsymbol{a'})\right\} - Q_t(s_t,\boldsymbol{a}_t)\right].\end{aligned}$$

After subtracting $Q^\star(s_t,\boldsymbol{a}_t)$ from both sides and some manipulation we obtain that:

$$\begin{aligned}&\Xi_{t+1}(s_t,\boldsymbol{a}_t)\\ &= (1-\alpha_t(s_t,\boldsymbol{a}_t))\Xi_t(s_t,\boldsymbol{a}_t)\\ &\quad + \alpha_t(s_t,\boldsymbol{a}_t))\left[\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g) + \gamma\max_{a'\in\mathcal{A}} Q_G(s',\boldsymbol{a'})\right\} - Q^\star(s_t,\boldsymbol{a}_t)\right],\end{aligned}$$

where $\Xi_t(s_t,\boldsymbol{a}_t) := Q_t(s_t,\boldsymbol{a}_t) - Q^\star(s_t,\boldsymbol{a}_t)$.

Let us now define by

$$L_t(s_{\tau_k},\boldsymbol{a}) := \max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g) + \gamma\max_{a'\in\mathcal{A}} Q_G(s',\boldsymbol{a'})\right\} - Q^\star(s_t,a).$$

Then

$$\Xi_{t+1}(s_t,\boldsymbol{a}_t) = (1-\alpha_t(s_t,\boldsymbol{a}_t))\Xi_t(s_t,\boldsymbol{a}_t) + \alpha_t(s_t,\boldsymbol{a}_t))\left[L_t(s_{\tau_k},a)\right]. \tag{8}$$

We now observe that

$$\begin{aligned}\mathbb{E}\left[L_t(s_{\tau_k},\boldsymbol{a})|\mathcal{F}_t\right] &= \sum_{s'\in\mathcal{S}} P(s';a,s_{\tau_k})\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g) + \gamma\max_{a'\in\mathcal{A}} Q_G(s',\boldsymbol{a'})\right\} - Q^\star(s_{\tau_k},a)\\ &= T_G Q_t(s,\boldsymbol{a}) - Q^\star(s,\boldsymbol{a}).\end{aligned} \tag{9}$$

Now, using the fixed point property that implies $Q^\star = T_G Q^\star$, we find that

$$\begin{aligned}\mathbb{E}\left[L_t(s_{\tau_k},\boldsymbol{a})|\mathcal{F}_t\right] &= T_G Q_t(s,\boldsymbol{a}) - T_G Q^\star(s,\boldsymbol{a})\\ &\leq \|T_G Q_t - T_G Q^\star\|\\ &\leq \gamma\left\|Q_t - Q^\star\right\|_\infty = \gamma\left\|\Xi_t\right\|_\infty.\end{aligned} \tag{10}$$

using the contraction property of T established in Lemma 3. This proves (ii).

We now prove iii), that is

$$\text{Var}\left[L_t|\mathcal{F}_t\right] \leq c(1 + \|\Xi_t\|^2). \tag{11}$$

Now by (9) we have that

$$\begin{aligned}
\mathrm{Var}\left[L_t|\mathcal{F}_t\right] &= \mathrm{Var}\left[\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g)+\gamma\max_{a'\in\mathcal{A}}Q_G(s',\boldsymbol{a'})\right\}-Q^\star(s_t,a)\right]\\
&= \mathbb{E}\Bigg[\Bigg(\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g)+\gamma\max_{a'\in\mathcal{A}}Q_G(s',\boldsymbol{a'})\right\}\\
&\qquad\qquad -Q^\star(s_t,a)-(T_G Q_t(s,\boldsymbol{a})-Q^\star(s,\boldsymbol{a}))\Bigg)^2\Bigg]\\
&= \mathbb{E}\left[\left(\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g)+\gamma\max_{a'\in\mathcal{A}}Q_G(s',\boldsymbol{a'})\right\}-T_G Q_t(s,\boldsymbol{a})\right)^2\right]\\
&= \mathrm{Var}\left[\max\left\{\mathcal{M}^{\mathfrak{g},\boldsymbol{\pi}^c}Q_G s_{\tau_k},\boldsymbol{a}), R_G(s_{\tau_k},\boldsymbol{a},g)+\gamma\max_{a'\in\mathcal{A}}Q_G(s',\boldsymbol{a'})\right\}-T_G Q_t(s,\boldsymbol{a}))^2\right]\\
&\leq c(1+\|\Xi_t\|^2),
\end{aligned}$$

for some $c > 0$ where the last line follows due to the boundedness of Q (which follows from Assumptions 2 and 4). This concludes the proof of part (i) of the Theorem (i.e. **[A]**).

□

□

Proof of Part B

To prove Part **B**, we prove the following result[1] :

Proposition 3. *For any $\pi \in \Pi$ and for any Global policy $\mathfrak{g}$, there exists a function $B^{\boldsymbol{\pi},\mathfrak{g}} : \mathcal{S}\times\{0,1\}\to\mathbb{R}$ such that*

$$v(s|\boldsymbol{\pi})-v(s|\boldsymbol{\pi}') = B(s|\boldsymbol{\pi},\mathfrak{g})-B(s|\boldsymbol{\pi},\mathfrak{g}'),\ \ \forall s\in\mathcal{S} \tag{12}$$

$$v_G(s|\boldsymbol{\pi},\mathfrak{g})-v_G(s|\boldsymbol{\pi},\mathfrak{g}') = B(s|\boldsymbol{\pi},\mathfrak{g})-B(s|\boldsymbol{\pi},\mathfrak{g}''),\ \ \forall s\in\mathcal{S} \tag{13}$$

$$v_G(s|\boldsymbol{\pi},\mathfrak{g})-v_G(s|\boldsymbol{\pi}',\mathfrak{g}) = B(s|\boldsymbol{\pi},\mathfrak{g})-B(s|\boldsymbol{\pi},\mathfrak{g}'),\ \ \forall s\in\mathcal{S} \tag{14}$$

where in particular the function B is given by:

$$B(s|\boldsymbol{\pi},\mathfrak{g}) = \mathbb{E}\left[\sum_{t=0}^{\infty}\gamma^t r\right], \tag{15}$$

for any $s\in\mathcal{S}$.

Proof. This is manifest from the construction of B and Assumption 6. □

Proof of Proposition 1

Proof of Prop. 1. We split the proof into two parts:

i) We first prove that $v^{\tilde{\boldsymbol{\pi}}}(s)\geq v^{\boldsymbol{\pi}}(s),\ \forall s\in\mathcal{S}$ where we use $\tilde{\boldsymbol{\pi}}$ to denote the N agents' joint policy induced under the influence of the Global.

ii) Second, we prove that there exists a finite integer M such that $v^{\tilde{\boldsymbol{\pi}}_m}(s)\geq v^{\boldsymbol{\pi}_m}(s)$ for any $m\geq M$.

The proof of part (i) is achieved by proof by contradiction. Denote by $v^{\boldsymbol{\pi},\mathfrak{g}\equiv 0}$ the value function for the Controller for the system *without the Global*. Indeed, let $(\hat{\boldsymbol{\pi}},\mathfrak{g})$ be the policy profile at the stable point of the system (Markov perfect equilibrium) and assume that Global's interventions lead to a decrease in total system returns. Then by construction $v^{\hat{\boldsymbol{\pi}},\mathfrak{g}}(s) < v^{\boldsymbol{\pi},\mathfrak{g}\equiv 0}(s)$ which is a contradiction since $(\hat{\boldsymbol{\pi}},\mathfrak{g})$ is a stable point (MPE profile).

[1] This property is analogous to the condition in Markov potential games (Macua et al., 2018; Mguni et al., 2021b)

We now prove part (ii).

By part (i) we have that $v^{\tilde{\boldsymbol{\pi}}}(s) = \lim_{m\to\infty} v^{\tilde{\boldsymbol{\pi}}_m}(s) \geq v^{\boldsymbol{\pi}}(s) = \lim_{m\to\infty} v^{\boldsymbol{\pi}_m}(s)$. Since $v^{\boldsymbol{\pi}}(s)$ is maximal in the sequence $v^{\boldsymbol{\pi}_1}(s), v^{\boldsymbol{\pi}_2}(s), \ldots, v^{\boldsymbol{\pi}}(s)$ we can deduce that $v^{\boldsymbol{\pi}}(s) \geq v^{\boldsymbol{\pi}_n}(s)$ for any $n \leq \infty$. Hence for any n there exists a $c \geq 0$ such that $\lim_{m\to\infty} v^{\tilde{\boldsymbol{\pi}}_m}(s) = v^{\boldsymbol{\pi}_n}(s) - c$. Now by construction $v^{\tilde{\boldsymbol{\pi}}_m}(s) \to v^{\tilde{\boldsymbol{\pi}}}(s)$ as $m \to \infty$, therefore the sequence $\tilde{v}^{\boldsymbol{\pi}_1}, \tilde{v}^{\boldsymbol{\pi}_2}, \ldots,$ forms a Cauchy sequence. Therefore, there exists an M such that for any $\epsilon > 0$, $v^{\tilde{\boldsymbol{\pi}}_n}(s) - (v^{\boldsymbol{\pi}_n}(s) - c) < \epsilon \; \forall n \geq M$. Since ϵ is arbitrary we can conclude that $v^{\tilde{\boldsymbol{\pi}}_n}(s) - (v^{\boldsymbol{\pi}_n}(s) - c) = 0 \; \forall n \geq M$. Since $c \geq 0$, we immediately deduce that $v^{\tilde{\boldsymbol{\pi}}_n}(s) \geq v^{\boldsymbol{\pi}_n}(s), \forall n \geq M$ which is the required result. □

Proof of Proposition 2

Proof. We begin by re-expressing the *activation times* at which the Global agent activates Central. In particular,an activation time τ_k is defined recursively $\tau_k = \inf\{t > \tau_{k-1} | s_t \in A, \tau_k \in \mathcal{F}_t\}$ where $A = \{s \in \mathcal{S}, g(s_t) = 1\}$. The proof is given by deriving a contradiction. Let us there suppose that $\mathcal{M}v_G(s_{\tau_k}) \leq v_G(s_{\tau_k})$ and that the activation time $\tau_1' > \tau_1$ is an optimal activation time. Construct the $\mathfrak{g}'$ and $\mathfrak{g}$ policy switching times by $(\tau_0', \tau_1', \ldots,)$ and $(\tau_0', \tau_1, \ldots)$ respectively. Define by $l = \inf\{t > 0; \mathcal{M}v_G(s_t) = v_G(s_t)\}$ and $m = \sup\{t; t < \tau_1'\}$. By construction we have that

$$\begin{aligned} &v_G(s|\boldsymbol{\pi}, \mathfrak{g}') \\ &= \mathbb{E}\left[R_G(s_0, \boldsymbol{a}_0, g) + \mathbb{E}\left[\ldots + \gamma^{l-1}\mathbb{E}\left[R(s_{\tau_1-1}, \boldsymbol{a}_{\tau_1-1}, g) + \ldots + \gamma^{m-l-1}\mathbb{E}\left[R_G(s_{\tau_1'-1}, \boldsymbol{a}_{\tau_1'-1}, g) + \gamma\mathcal{M}^{\boldsymbol{\pi},\mathfrak{g}'}v_G(s_{\tau_1}|\boldsymbol{\pi}, \mathfrak{g}')\right]\right]\right]\right] \\ &< \mathbb{E}\left[R_G(s_0, \boldsymbol{a}_0, g) + \mathbb{E}\left[\ldots + \gamma^{l-1}\mathbb{E}\left[R_G(s_{\tau_1-1}, \boldsymbol{a}_{\tau_1-1}, g) + \gamma\mathcal{M}^{\boldsymbol{\pi},\tilde{\mathfrak{g}}}v_G(s_{\tau_1}|\boldsymbol{\pi}, \mathfrak{g}')\right]\right]\right] \end{aligned}$$

We make use of the following observation

$$\mathbb{E}\left[R_G(s_{\tau_1-1}, \boldsymbol{a}_{\tau_1-1}, g) + \gamma\mathcal{M}^{\boldsymbol{\pi},\tilde{\mathfrak{g}}}v_G(s_{\tau_1}|\boldsymbol{\pi}, \mathfrak{g}')\right] \tag{16}$$

$$\leq \max\left\{\mathcal{M}^{\boldsymbol{\pi},\tilde{\mathfrak{g}}}v_G(s_{\tau_1}|\boldsymbol{\pi}, \mathfrak{g}'), \max_{\boldsymbol{a}_{\tau_1}\in\boldsymbol{\mathcal{A}}}\left[R_G(s_{\tau_1}, \boldsymbol{a}_{\tau_1}, g) + \gamma\sum_{s'\in\mathcal{S}} P(s'; a_{\tau_1}, s_{\tau_1})v_G(s'|\boldsymbol{\pi}, \mathfrak{g})\right]\right\}. \tag{17}$$

Using this we deduce that

$$\begin{aligned} &v_G(s|\boldsymbol{\pi}, \mathfrak{g}') \leq \mathbb{E}\Bigg[R_G(s_0, \boldsymbol{a}_0, g) + \mathbb{E}\Bigg[\ldots \\ &+ \gamma^{l-1}\mathbb{E}\left[R_G(s_{\tau_1-1}, \boldsymbol{a}_{\tau_1-1}, g) + \gamma\max\left\{\mathcal{M}^{\boldsymbol{\pi},\tilde{\mathfrak{g}}}v_G(s_{\tau_1}|\boldsymbol{\pi}, \mathfrak{g}'), \max_{\boldsymbol{a}_{\tau_1}\in\boldsymbol{\mathcal{A}}}\left[R_G(s_{\tau_k}, \boldsymbol{a}_{\tau_k}, g) + \gamma\sum_{s'\in\mathcal{S}} P(s'; a_{\tau_1}, s_{\tau_1})v_G(s'|\boldsymbol{\pi}, \mathfrak{g}),\right]\right\}\right]\Bigg]\Bigg] \\ &= \mathbb{E}\left[R_G(s_0, \boldsymbol{a}_0, g) + \mathbb{E}\left[\ldots + \gamma^{l-1}\mathbb{E}\left[R_G(s_{\tau_1-1}, \boldsymbol{a}_{\tau_1-1}, g) + \gamma\left[T_G v_G(s_{\tau_1}|\boldsymbol{\pi}, \tilde{\mathfrak{g}})\right]\right]\right]\right] = v_G(s|\boldsymbol{\pi}, \tilde{\mathfrak{g}}), \end{aligned}$$

where the first inequality is true by assumption on $\mathcal{M}$. This is a contradiction since π' is an optimal policy for Player 2. Using analogous reasoning, we deduce the same result for $\tau_k' < \tau_k$ after which deduce the result. Moreover, by invoking the same reasoning, we can conclude that it must be the case that $(\tau_0, \tau_1, \ldots, \tau_{k-1}, \tau_k, \tau_{k+1}, \ldots,)$ are the optimal switching times. This completes the proof. □

17. Proof of Theorem 2

Proof. The proof of the Theorem is straightforward since by Theorem 1, Global's problem can be solved using a dynamic programming principle. The proof immediately by application of Theorem 2 in (Sootla et al., 2022).

□