A. Convergence Analysis

A.1. Preliminaries

Let $\mathfrak{J} = \phi \circ \boldsymbol{J} : \mathbb{R}^{N\times K} \to \mathbb{R}$ denote the *Social Welfare Objective* (SWO), as the composition of the social welfare function ϕ with the vectorial objective $\boldsymbol{J}$, where N is the number of agents and K denotes the dimension of the parameter $\boldsymbol{\theta}_i$ for the policy network of an agent $i \in [N]$.

Under a gradient descent optimization scheme, the direction of the update should follow the gradient at iteration k, i.e.,

$$\mathfrak{g}^{k,*} \coloneqq \nabla\mathfrak{J}(\boldsymbol{\theta}^k) = \nabla\phi(\boldsymbol{J}(\boldsymbol{\theta}^k))^\intercal \cdot \nabla\boldsymbol{J}(\boldsymbol{\theta}^k), \tag{10}$$

where $\nabla\boldsymbol{J}$ denotes the Jacobian of $\boldsymbol{J}$. While the subscript on $\boldsymbol{\theta}$ refers to the different agents, superscript k refers to the successive iterations of policy; it may be omitted below to lighten the notations.

Adapting the Policy Gradient Theorem (Sutton & Barto, 1998) to this fair-multi-agent context, leads us to the following gradient direction for agent i:

Lemma A.1.

$$\nabla_{\boldsymbol{\theta}_i}\mathfrak{J}(\boldsymbol{\theta}^k) = \beta_k \mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}^{k,SWF}(\boldsymbol{s},\boldsymbol{a}) \cdot \nabla_{\boldsymbol{\theta}_i} \log \pi^k_{\boldsymbol{\theta}_i}(a_i|o_i)\Big], \tag{11}$$

where $\boldsymbol{A}^{k,SWF} \coloneqq \nabla\phi(\boldsymbol{J}(\boldsymbol{\theta}^k))^\intercal \cdot \boldsymbol{A}^k : \mathcal{S}\times\mathcal{A} \to \mathbb{R}$, and β_k is the average length of an episode while following the policy $\boldsymbol{\pi}^k$ in the episodic case or $\beta_k = 1$ in the infinite horizon case.[4]

Proof. Let us look at $\boldsymbol{J}_{(j)}(\boldsymbol{\theta}) \coloneqq \boldsymbol{J}_{I_j}(\boldsymbol{\theta})$, the vectorial objective of agent i, and roughly follow the classical policy gradient theorem's proof. We immediately have:

$$\nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(s) = \sum_{a\in\mathcal{A}} \Big(\nabla_i \pi_{\boldsymbol{\theta}}(\boldsymbol{a} \mid \boldsymbol{s}) \boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s},\boldsymbol{a}) + \pi_{\boldsymbol{\theta}}(\boldsymbol{a} \mid \boldsymbol{s}) \nabla_i \boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s},\boldsymbol{a})\Big)$$

where ∇_i, $\boldsymbol{Q}_{(j)}$ and $\boldsymbol{V}_{(j)}$ stands for $\nabla_{\boldsymbol{\theta}_i}$, $\boldsymbol{Q}_{I_j}$ and $\boldsymbol{V}_{I_j}$ respectively. For the second term, we can develop $\boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}$ as $\sum_{s',r} P(s',r|s,a)(r + \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}'))$. Since nor the reward nor the transition probability depends on θ, it leads to, after simplifications:

$$\nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(s) = \psi_i(\boldsymbol{s}) + \sum_{a\in\mathcal{A}} \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) \sum_{s'\in\mathcal{S}} P(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a}) \nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}') = \psi_i(\boldsymbol{s}) + \sum_{s'\in\mathcal{S}} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',1) \nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}'), \tag{12}$$

where $\psi_{i,j}(\boldsymbol{s}) \coloneqq \sum_{a\in\mathcal{A}} \nabla_i \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) \boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s},\boldsymbol{a})$, and $\rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',k)$ is the probability of transitioning from state s to state s' while following the policy $\boldsymbol{\pi}$ after k steps. Notably $\rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',1) = \sum_{a\in\mathcal{A}} \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) P(\boldsymbol{s}'|\boldsymbol{s},\boldsymbol{a})$. The equation (12) enables us to smoothly unroll a recursive process:

$$\begin{aligned}
\nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(s) &= \psi_{i,j}(\boldsymbol{s}) + \sum_{s'\in\mathcal{S}} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',1) \left[\psi_{i,j}(\boldsymbol{s}') + \sum_{\boldsymbol{s}''\in\mathcal{S}} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}'\to\boldsymbol{s}'',1) \nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(s'')\right] \\
&= \psi_{i,j}(\boldsymbol{s}) + \sum_{s'\in\mathcal{S}} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',1)\psi_{i,j}(\boldsymbol{s}') + \sum_{s'\in\mathcal{S}}\sum_{\boldsymbol{s}''\in\mathcal{S}} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}'',2) \nabla_i \boldsymbol{V}^{\boldsymbol{\pi}}_{(j)}(s'') \\
&= \cdots \text{ (iterating) } \cdots \\
&= \sum_{\boldsymbol{s}'\in\mathcal{S}} \sum_{k=0}^{\infty} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}\to\boldsymbol{s}',k)\psi_{i,j}(\boldsymbol{s}'),
\end{aligned}$$

Coming back to the derivative of the vectorial objective of agent j:

$$\nabla_i \boldsymbol{J}_{(j)}(\boldsymbol{\theta}) = \sum_{\boldsymbol{s}\in\mathcal{S}} \chi^{\boldsymbol{\pi}}(\boldsymbol{s})\psi_{i,j}(\boldsymbol{s}) = \beta^{\boldsymbol{\pi}} \left(\sum_{\boldsymbol{s}\in\mathcal{S}} d^{\boldsymbol{\pi}}(\boldsymbol{s})\psi_{i,j}(\boldsymbol{s})\right),$$

[4] In the infinite horizon case, we can follow a nearly-identical proof, upon the assumption the underlying MDP is ergodic.

where $\chi^{\boldsymbol{\pi}}(\boldsymbol{s}) \coloneqq \sum_{k=0}^{\infty} \rho^{\boldsymbol{\pi}}(\boldsymbol{s}_0 \to \boldsymbol{s}, k)$, $d^{\boldsymbol{\pi}}(\boldsymbol{s}) = \frac{\chi^{\boldsymbol{\pi}}(\boldsymbol{s})}{\sum_{\boldsymbol{s}' \in \mathcal{S}} \chi^{\boldsymbol{\pi}}(\boldsymbol{s}')}$ is the stationary distribution and $\beta^{\boldsymbol{\pi}} := \sum_{\boldsymbol{s} \in \mathcal{S}} \chi^{\boldsymbol{\pi}}(\boldsymbol{s})$. Since $\pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) = \sum_{\boldsymbol{o}} \Omega(\boldsymbol{o}|\boldsymbol{s}) \prod_i \pi_{\boldsymbol{\theta}_i}(a_i|o_i)$:

$$\nabla_i \boldsymbol{J}_{(j)}(\boldsymbol{\theta}) = \beta^{\boldsymbol{\pi}} \sum_{\boldsymbol{s} \in \mathcal{S}} \sum_{\boldsymbol{a} \in \mathcal{A}} d^{\boldsymbol{\pi}}(\boldsymbol{s}) \boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}, \boldsymbol{a}) \nabla_i \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) = \beta^{\boldsymbol{\pi}} \sum_{\boldsymbol{s} \in \mathcal{S}} \sum_{a \in \mathcal{A}} d^{\boldsymbol{\pi}}(\boldsymbol{s}) \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s}) \boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}, \boldsymbol{a}) \nabla_i \log \pi_{\boldsymbol{\theta}}(\boldsymbol{a}|\boldsymbol{s})$$
$$= \beta^{\boldsymbol{\pi}} \mathbb{E}_{\substack{\boldsymbol{s} \sim d^{\boldsymbol{\pi}}(\bullet) \\ \boldsymbol{a} \sim \pi(\bullet|\boldsymbol{s}) \\ \boldsymbol{o} \sim \Omega(\bullet|\boldsymbol{s})}} \Big[\boldsymbol{Q}^{\boldsymbol{\pi}}_{(j)}(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big]$$

These computations may be equivalently rewritten with the advantage function:

$$\nabla_i \boldsymbol{J}(\boldsymbol{\theta}) = \beta^{\boldsymbol{\pi}} \mathbb{E}\Big[\boldsymbol{A}^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big]$$

Hence, returning to the SWO $\mathfrak{J}$, it boils down to the Lemma's claim:

$$\nabla_i \mathfrak{J}(\boldsymbol{\theta}) = \beta^{\boldsymbol{\pi}} \nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}))^T \cdot \mathbb{E}\Big[A^{\boldsymbol{\pi}}(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big] = \beta^{\boldsymbol{\pi}} \mathbb{E}\Big[A^{\boldsymbol{\pi}, SWF}(\boldsymbol{s}, \boldsymbol{a}) \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big]$$

□

As mentioned previously, since we adopt a fully decentralised framework with partial observability and since the advantage function is not known, we instead approximate our gradient update according to the following direction $\tilde{\mathfrak{g}}_i^k$ for an agent i:

$$\tilde{\mathfrak{g}}_i^k \coloneqq \mathbb{E}_{\boldsymbol{\theta}} \Big[\hat{\boldsymbol{A}}^{k,\text{SWF}}(\boldsymbol{o}, \boldsymbol{a}) \cdot \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big],$$

where $\hat{\boldsymbol{A}}^k = (\hat{\boldsymbol{A}}_i^k)_{i \in [N]}$ aggregates the estimate of the local advantage at step k for each agent i, $\hat{\boldsymbol{A}}_i^k : \mathcal{O}_i \times \mathcal{A}_i \to \mathbb{R}^{|I_i|}$, and $\hat{\boldsymbol{A}}^{k,\text{SWF}}(\boldsymbol{o}, \boldsymbol{a}) = \nabla \phi(\hat{\boldsymbol{J}}(\boldsymbol{\theta}^k))^\intercal \cdot \hat{\boldsymbol{A}}^k(\boldsymbol{o}, \boldsymbol{a})$.

Let $\hat{\mathfrak{g}}^k \in \mathbb{R}^{N \times K}$ denote the estimated gradient at step k, estimated from sampling a minibatch of transitions of size m. The update at step k of this stochastic gradient ascent can be written as:

$$\boldsymbol{\theta}^{k+1} \leftarrow \boldsymbol{\theta}^k + \alpha_k \hat{\mathfrak{g}}^k \tag{13}$$

Comparing the ideal and estimated gradients by introducing the error term $\boldsymbol{\epsilon}^k$, using (11):

$$\hat{\mathfrak{g}}^k = \overline{\mathfrak{g}}^k + \boldsymbol{\epsilon}^k = \beta_k \mathfrak{g}^{k,*} + \boldsymbol{\epsilon}^k, \quad \text{where} \quad \overline{\mathfrak{g}}_i^k := \mathbb{E}_{\boldsymbol{\theta}} \Big[\boldsymbol{A}^{k,\text{SWF}}(\boldsymbol{s}, \boldsymbol{a}) \cdot \nabla_{\boldsymbol{\theta}_i} \log \pi_{\boldsymbol{\theta}_i}(a_i|o_i) \Big]. \tag{14}$$

Let us decompose this error term into two parts:

$$\boldsymbol{\epsilon}^k = \boldsymbol{\nu}^k + \boldsymbol{\eta}^k, \quad \text{where } \boldsymbol{\nu}^k \coloneqq \hat{\mathfrak{g}}^k - \tilde{\mathfrak{g}}^k \quad \text{and} \quad \boldsymbol{\eta}^k \coloneqq \tilde{\mathfrak{g}}^k - \overline{\mathfrak{g}}^k \tag{15}$$

While the first term $\boldsymbol{\nu}^k$ tends towards zero when m tends towards infinity, some assumptions would be made below to control the second part of this error $\boldsymbol{\eta}^k$.

A.2. Assumptions

Here we detail and discuss the assumptions made in our theoretical analysis. The first part of the assumptions regarding the social welfare function ϕ has already been justified in the context of fairness considerations in Section 3.2. We simply assume in addition that the norm of the gradient of ϕ is bounded by $M_\phi \in \mathbb{R}^+$ on the image of $\boldsymbol{J}$.

Assumption 1. *ϕ is concave and non-decreasing in each argument. It is (sub-)differentiable[5] and the norm of its gradient is assumed to be bounded by $M_\phi \in \mathbb{R}^+$ on the image of $\boldsymbol{J}$. This implies that ϕ is L_ϕ-smooth on the image of $\boldsymbol{J}$, with $L_\phi \leq M_\phi$.*

Adopting the boundedness of the reward function is a common and legitimate hypothesis in the literature of policy gradient or actor-critic algorithms. If B_R is uniformly bounding the reward, in the discounted case, $\frac{B_R}{1-\gamma}$ would uniformly bound both the value or action-value function, and $\frac{2B_R}{1-\gamma}$ would bound the advantage function. From these considerations, it seems legitimate to assume the estimated advantage function to be bounded:

[5] We could adapt the proof to the case of sub-differentiability of ϕ and $\boldsymbol{J}$, yet their sub-gradient should satisfy Lipschitz-inequalities.

Assumption 2. *The estimated advantage function $\hat{A}^k$ is uniformly bounded by B_A, for all k.*

Here, we adopt classic assumptions concerning the vectorial objective function $\boldsymbol{J}$:

Assumption 3. *The function $\boldsymbol{J} : \mathbb{R}^{N\times K} \to \mathbb{R}^D$ is concave and Lipschitz-smooth with constant $L_{\boldsymbol{J}}$, i.e., its gradient is $L_{\boldsymbol{J}}$-Lipschitz continuous. These two hypotheses imply respectively:*

$$\forall i \in [D], \quad \boldsymbol{J}_i(\boldsymbol{\theta}') - \boldsymbol{J}_i(\boldsymbol{\theta}) \leq \langle \nabla \boldsymbol{J}_i(\boldsymbol{\theta}), \boldsymbol{\theta}' - \boldsymbol{\theta} \rangle$$

$$\|\nabla \boldsymbol{J}(\boldsymbol{\theta}') - \nabla \boldsymbol{J}(\boldsymbol{\theta})\| \leq L_{\boldsymbol{J}} \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|$$

This assumption notably covers the case where a linear approximation scheme is used. By Cauchy-Schwartz, integrating the last equation, and using the concavity hypothesis:

$$\boldsymbol{J}(\boldsymbol{\theta}') - \boldsymbol{J}(\boldsymbol{\theta}) \geq \nabla \boldsymbol{J}(\boldsymbol{\theta})^\intercal (\boldsymbol{\theta}' - \boldsymbol{\theta}) - \frac{L_{\boldsymbol{J}}}{2} \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|^2 \tag{16}$$

The previous assumptions imply:

Lemma A.2. (i) *Under Assumptions (1, 3), the SWO $\mathfrak{J}$ is concave.*

(ii) *Under Assumptions (1, 3), with the additional hypothesis that the gradient of the vectorial objective $\nabla \boldsymbol{J}$ is bounded by $M_{\boldsymbol{J}}$, then the SWO $\mathfrak{J}$ is $L_{\mathfrak{J}}-$smooth, with:*

$$L_{\mathfrak{J}} \leq L_\phi M_{\boldsymbol{J}}^2 + M_\phi L_{\boldsymbol{J}}.$$

Proof. (i) From Assumptions 1 and 3, both ϕ and $\boldsymbol{J}$ are concave, and ϕ is non-decreasing in each argument. It is then straightforward to prove that the fair objective $\mathfrak{J} = \phi \circ J$ is concave.

(ii) For the last assertion, we can show:

$$\begin{aligned}
\|\nabla \mathfrak{J}(\boldsymbol{\theta}') - \nabla \mathfrak{J}(\boldsymbol{\theta})\| &= \|\nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}'))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta}') - \nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta})\| \\
&\leq \|\nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}'))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta}') - \nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta}')\| + \|\nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta}') - \nabla \phi(\boldsymbol{J}(\boldsymbol{\theta}))^\intercal \nabla \boldsymbol{J}(\boldsymbol{\theta})\| \\
&\leq L_\phi \|\boldsymbol{J}(\boldsymbol{\theta}') - \boldsymbol{J}(\boldsymbol{\theta})\| M_J + L_J M_\phi \|\boldsymbol{\theta}' - \boldsymbol{\theta}\| \leq \left(L_\phi M_J^2 + L_J M_\phi\right) \|\boldsymbol{\theta}' - \boldsymbol{\theta}\|.
\end{aligned}$$

□

Let Θ^* denote the set of maximizers of $\mathfrak{J}$, i.e., $\Theta^* \coloneqq \operatorname{argmax}\{\mathfrak{J}(\boldsymbol{\theta}) \mid \boldsymbol{\theta} \in \mathbb{R}^{N\times K}\}$. The concavity of $\mathfrak{J}$ implies that the set $\boldsymbol{\theta}^*$ is convex. We furthermore assume:

Assumption 4. *The optimal set Θ^* is non empty and there exists a radius bound R_0 such that:*

$$\max_{\boldsymbol{\theta}^* \in \Theta^*} \max_{\boldsymbol{\theta}} \{\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\| \mid \mathfrak{J}(\boldsymbol{\theta}) \geq \mathfrak{J}(\boldsymbol{\theta}^0)\} \leq R_0,$$

where θ^0 is assumed to be the initial parameter value.[6]

A common condition on the learning rate in convex optimization is to assume it is smaller than the inverse of the Lipschitz constant. Here, we use a slight variant of this condition:

Assumption 5. *Given Lipschitz constant $L_{\boldsymbol{J}}$ from Assumption 3, and gradient bound M_ϕ from Assumption 1, the learning rate satisfies:*

$$\alpha_k \leq \frac{1}{L_{\boldsymbol{J}} M_\phi \beta_k},$$

where β_k is the average length of an episode while following the policy $\boldsymbol{\pi}^k$ in the episodic case or $\beta_k = 1$ in the infinite horizon case.

[6]However, if such hypothesis is not satisfied for $\boldsymbol{\theta}^0$, we may assume it is satisfied for another $\boldsymbol{\theta}^{i_0}$; it alters the inequality formulation in our theorem —as we need to shift the indices by i_0—, but not the convergence result itself.

Adopting classic assumptions regarding the policy, in the policy gradient or actor-critic literature of convergence analysis:[7]

Assumption 6. (i) *The policy $\boldsymbol{\pi_\theta}$ is differentiable with respect to $\boldsymbol{\theta}$, and $\nabla \log \boldsymbol{\pi_\theta}(\boldsymbol{a}|\boldsymbol{s})$, referred to as the score function, exists. Moreover, there exists a bound $M_{\boldsymbol{\pi}} \in \mathbb{R}^+$ which uniformly bounds $\|\nabla \log \boldsymbol{\pi_\theta}\|$ for all $\boldsymbol{\theta}$:*

$$\sup_{\boldsymbol{\theta},\boldsymbol{s},\boldsymbol{a}} \|\nabla \log \boldsymbol{\pi_\theta}(\boldsymbol{a}|\boldsymbol{s})\| = M_{\boldsymbol{\pi}}$$

(ii) *The variance of the estimated gradient is uniformly bounded:*

$$\sup_{\boldsymbol{\theta},\boldsymbol{s},\boldsymbol{a}} \mathrm{Var}(\hat{\mathfrak{g}}^{\boldsymbol{\theta}}(\boldsymbol{s},\boldsymbol{a})) \leq \sigma^{\sharp}$$

In our context of fully decentralized and partially observable multi-agent RL, we have to consider the error made by the local estimates of the advantage function, as mentioned in (7). Within such an approximation lies several sources of error that we will consider separately below.

First, we need to approximate the objective $\boldsymbol{J}$. Since we want to focus our analysis on the convergence of the policy[8], we make the following assumptions:

Assumption 7. *The approximation error of $\boldsymbol{J}$ is bounded in probability:*[9]

$$\forall k > 0, \quad \mathsf{P}\left(\left\|\hat{\boldsymbol{J}}(\boldsymbol{\theta}^k) - \boldsymbol{J}(\boldsymbol{\theta}^k)\right\| \leq \Xi_J\right) = 1$$

Then, we have to control the approximation error of the local advantage function, which stems from several sources (such as the finiteness of the sample set on which it would be estimated, and the expressiveness of the class of functions estimators used for $\hat{Q}$ and $\hat{V}$):

Assumption 8. *The approximation error of the local advantage $\boldsymbol{A}$ is uniformly bounded in probability:*

$$\forall k > 0, \quad \mathsf{P}\left(\left\|\hat{\boldsymbol{A}}^k(\boldsymbol{o},\boldsymbol{a}) - \boldsymbol{A}^k(\boldsymbol{o},\boldsymbol{a})\right\|_\infty \leq \Xi_A\right) = 1$$

Finally, the mismatch between the local and global advantage functions has to be taken into consideration:

Assumption 9. *The error stemming from the use of the decentralized local advantage is uniformly bounded in probability:*

$$\forall k > 0, \forall i \in [N], \forall j \in [N], j \neq i, \quad \mathsf{P}\left(\left\|\boldsymbol{A}^k_{j/i}(o_i,a_i)\right\|_\infty \leq \Xi_L\right) = 1$$

Or, less restrictively, after application of the SWF ϕ, we can assume instead:

$$\forall k > 0, \forall i \in [N], \quad \mathsf{P}\left(\left\|\boldsymbol{A}^{(-i),k,SWF}_{\bullet/i}(o_i,a_i)\right\|_\infty \leq \Xi_L\right) = 1, \tag{17}$$

with $\boldsymbol{A}^k_{j/i}(o_i,a_i) \coloneqq \mathbb{E}_{\substack{\boldsymbol{o}_{-i}\sim \mathsf{P}(\cdot|o_i),\\ \boldsymbol{a}_{-i}\sim \boldsymbol{\pi}^{-i}(\cdot|\boldsymbol{o}_{-i})}} \left[\boldsymbol{A}^k_j(\{o_i,\boldsymbol{o}_{-i}\},\{a_i,\boldsymbol{a}_{-i}\})\right]$ *and* $\left(\boldsymbol{A}^{(-i),k}_{\bullet/i}(o_i,a_i)\right)_j \coloneqq \begin{cases} \boldsymbol{A}^k_{j/i}(o_i,a_i) & j \neq i \\ 0 & j = i \end{cases}$.

Here, the notations $\boldsymbol{o}_{-i}$ resp. $\boldsymbol{a}_{-i}$ designate the set $\{o_j\}_{j\neq i}$ resp. $\{a_j\}_{j\neq i}$. The notation $\boldsymbol{A}_{j/i}$ stands for the advantage of the actions of agent i relative to another agent j. As made clear in the proof of Lemma A.3, when comparing $\boldsymbol{A}(\boldsymbol{s},\boldsymbol{a})$ and $\boldsymbol{A}(\boldsymbol{o},\boldsymbol{a})$, these inter-agent advantages $\boldsymbol{A}^{(-i)}_{\bullet/i}$ appear. This assumption tackles the discrepancy between each agent's local advantage and each agent's global advantage, which would take into consideration the benefit of the actions of other agents.

[7] Assumption 5.1 in (Xu et al., 2020), Assumption 3.1 in (Zhang et al., 2020), Assumption 4.1 in (Papini et al., 2018), Assumption 1 in (Kumar et al., 2019), Assumption 4.5 in (Qiu et al., 2021), etc.

[8] In Actor-Critic schemes, it is common to adopt a two-timescale analysis, in which we usually need to ensure that the actor update moves slowly while the critic update chases the slowly-moving target defined by the actor by enforcing a condition on the ratio of their learning rates (cf. (Konda & Tsitsiklis, 2000)). Here, we do not focus on bounding the error made from the advantage estimation in probability, in function of the size of the mini-batch m, but the analysis could be extended in later work.

[9] If we adopt GGF as SWF, this hypothesis could be relaxed, as we would only care that the estimate of $\boldsymbol{J}$ preserves the order of the different $J_i(\boldsymbol{\theta}^k)$ for each agent, not the actual value.

Looking at (17), let us remark that the benefits (positive or negative) of one agent action, may, upon application of the SWF ϕ, average out in some situations. Therefore, ignoring them may not have too much impact.

Note that the conditional probability arising in the above expected value satisfies, with the previous notations:

$$\mathsf{P}(\boldsymbol{o}_{-i} \mid o_i) = \sum_{\boldsymbol{s}} d^{\boldsymbol{\pi}}(\boldsymbol{s})\mathsf{P}(\boldsymbol{o}_{-i} \mid o_i, \boldsymbol{s}) = \sum_{\boldsymbol{s}} \frac{d^{\boldsymbol{\pi}}(\boldsymbol{s})\Omega(\boldsymbol{o} \mid \boldsymbol{s})}{\mathsf{P}(o_i \mid \boldsymbol{s})} = \sum_{\boldsymbol{s}} \frac{d^{\boldsymbol{\pi}}(\boldsymbol{s})\Omega(\boldsymbol{o} \mid \boldsymbol{s})}{\Omega(o_i \mid \boldsymbol{s})}$$

These assumptions enable to bound the error term $\boldsymbol{\eta}^k$ as defined in (15):

Lemma A.3. *Under assumptions 1,2,6,7,8, 9, $\boldsymbol{\eta}^k$ is bounded in probability:*

$$\forall k \quad \mathsf{P}\left(\left\|\boldsymbol{\eta}^k\right\| \leq \Xi\right) = 1, \quad \textit{with} \quad \Xi = (L_\phi B_A \Xi_J + M_\phi \Xi_A + M_\phi \Xi_L) M_{\boldsymbol{\pi}},$$

with the bounds as previously defined.

Proof. Splitting into three terms, from the definition of $\boldsymbol{A}^{SWF}$:

$$\begin{aligned}
\left\|\boldsymbol{\eta}_i^k\right\| = \left\|\tilde{\mathfrak{g}}_i^k - \overline{\mathfrak{g}}_i^k\right\| &= \left\|\mathbb{E}_{\boldsymbol{\theta}}\Big[\left(\hat{\boldsymbol{A}}^{k,\mathrm{SWF}}(\boldsymbol{o},\boldsymbol{a}) - \boldsymbol{A}^{k,\mathrm{SWF}}(\boldsymbol{s},\boldsymbol{a})\right)\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]\right\| \\
&\leq \left\|\mathbb{E}_{\boldsymbol{\theta}}\Big[\left(\nabla\phi(\hat{\boldsymbol{J}}(\boldsymbol{\theta}^k)) - \nabla\phi\left(\boldsymbol{J}(\boldsymbol{\theta}^k)\right)\right)^{\intercal}\cdot\hat{\boldsymbol{A}}^{\mathrm{k}}(\boldsymbol{o},\boldsymbol{a})\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]\right\| \\
&\quad + \left\|\mathbb{E}_{\boldsymbol{\theta}}\Big[\nabla\phi\left(\boldsymbol{J}(\boldsymbol{\theta}^k)\right)^{\intercal}\cdot(\hat{\boldsymbol{A}}^{\mathrm{k}}(\boldsymbol{o},\boldsymbol{a}) - \boldsymbol{A}^{\mathrm{k}}(\boldsymbol{o},\boldsymbol{a}))\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]\right\| \\
&\quad + \left\|\mathbb{E}_{\boldsymbol{\theta}}\Big[\nabla\phi\left(\boldsymbol{J}(\boldsymbol{\theta}^k)\right)^{\intercal}\cdot(\boldsymbol{A}^{\mathrm{k}}(\boldsymbol{o},\boldsymbol{a}) - \boldsymbol{A}^{\mathrm{k}}(\boldsymbol{s},\boldsymbol{a}))\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]\right\|
\end{aligned} \tag{18}$$

Since $\|\mathbb{E}(X)\| \leq \mathbb{E}(\|X\|)$:

$$\begin{aligned}
\left\|\boldsymbol{\eta}_i^k\right\| &\leq \Big(L_\phi B_A \left\|\boldsymbol{J}(\boldsymbol{\theta}^k) - \boldsymbol{J}(\boldsymbol{\theta})\right\| + M_\phi\Xi_A + M_\phi\Xi_L\Big)\mathbb{E}_{\boldsymbol{\theta}}\Big[\left\|\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\right\|\Big] \\
&\leq \Big(L_\phi B_A \Xi_J + M_\phi\Xi_A + M_\phi\Xi_L\Big) M_{\boldsymbol{\pi}},
\end{aligned} \tag{19}$$

with the previously defined bounds; for the first inequality we used the fact that ϕ is L_ϕ-smooth, and Assumptions 2 resp. 8 resp. 9 for the different terms; for the second inequality, we used Assumption 7 and the boundedness of $\|\nabla \ln \boldsymbol{\pi}\|$ from Assumption 6.

Let us justify the computations for the third term of (18). On the one hand, from the definition of $\boldsymbol{A}_{j/i}$, after some simplifications, we arrive at:

$$\mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}_j^{\mathrm{k}}(\boldsymbol{s},\boldsymbol{a})\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big] = \mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}_{j/i}^{\mathrm{k}}(o_i,a_i)\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]$$

On the other hand, the other terms may be simplified to:

$$\mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}_j^{\mathrm{k}}(o_j,a_j)\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big] = \begin{cases} 0 & j \neq i \\ \mathbb{E}_{\boldsymbol{\theta}}\Big[\boldsymbol{A}_i^{\mathrm{k}}(o_i,a_i)\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big] & j = i \end{cases}$$

Therefore, the third term of the right side in (18), boils down to:

$$\left\|\mathbb{E}_{\boldsymbol{\theta}}\Big[\nabla\phi\left(\boldsymbol{J}(\boldsymbol{\theta}^k)\right)^{\intercal}\cdot(\boldsymbol{A}_{\bullet/i}^{(-i),k}(o_i,a_i))\nabla_{\boldsymbol{\theta}_i}\log\pi_{\boldsymbol{\theta}_i}^k(a_i|o_i)\Big]\right\|,$$

where $\boldsymbol{A}_{\bullet/i}^{(-i),k}(o_i,a_i)$ denotes the vector whose j-th component is: $\left(\boldsymbol{A}_{\bullet/i}^{(-i),k}(o_i,a_i)\right)_j := \begin{cases} \boldsymbol{A}_{j/i}^k(o_i,a_i) & j \neq i \\ 0 & j = i \end{cases}$.

Assumption 9 enable us to conclude.

□

Locally concave Extension To extend our analysis to the locally concave scenario, two previous assumptions (Assumptions 3, 4) need to be reformulated as follows:

Assumption ($3'$)**.** *There exists a neighborhood $\mathcal{U} \subset \mathbb{R}^{N\times K}$ containing the initial parameter value θ^0, such that:*

- *The function $\boldsymbol{J} : \mathbb{R}^{N\times K} \to \mathbb{R}^D$ is concave on $\mathcal{U}$. It is Lipschitz-smooth on $\mathcal{U}$ with constant $L_{\boldsymbol{J}}$, i.e., its gradient is $L_{\boldsymbol{J}}$-Lipschitz continuous on $\mathcal{U}$.*
- *The set of optimal values in $\mathcal{U}$ $\Theta^{\mathcal{U},*}$ is non empty. There exists a radius bound $R_0^{\mathcal{U}}$ such that:*

$$\max_{\theta^*\in\Theta^{\mathcal{U},*}} \max_{\theta\in\mathcal{U}} \{\|\theta-\theta^*\| \mid \mathfrak{J}(\theta) \geq \mathfrak{J}(\theta^0)\} \leq R_0^{\mathcal{U}}.$$

This scenario covers the more general case where a neural network is used for the approximation. As previously, together with Assumption (1), (i) implies the local concavity of $\mathfrak{J}$:

Lemma A.4. *Under Assumptions (1, $3'$), the SWO $\mathfrak{J}$ is locally concave.*

A.3. Suboptimality Analysis

Lemma A.5. *Under Assumptions (1, 3, 5) the SWO between two consecutive update satisfies:*

$$\forall k>0, \quad \mathfrak{J}(\boldsymbol{\theta}^{k+1}) - \mathfrak{J}(\boldsymbol{\theta}^k) \geq \gamma_k \left\|\nabla\mathfrak{J}(\boldsymbol{\theta}^k)\right\|^2 - \delta_k \left\|\boldsymbol{\epsilon}^k\right\|^2, \quad \text{with } \gamma_k = \frac{\alpha_k\beta_k}{2}, \text{ and } \delta_k = \frac{\alpha_k}{2\beta_k}. \tag{20}$$

Proof. Since ϕ is assumed concave (Assumption 1):

$$\mathfrak{J}(\boldsymbol{\theta}^{k+1}) - \mathfrak{J}(\boldsymbol{\theta}^k) \geq \nabla\phi(\boldsymbol{J}(\boldsymbol{\theta}^k))^\intercal(\boldsymbol{J}(\boldsymbol{\theta}^{k+1}) - \boldsymbol{J}(\boldsymbol{\theta}^k)). \tag{21}$$

Since $\boldsymbol{J}$ is assumed $L_{\boldsymbol{J}}-$smooth and concave (Assumption 3), the following inequality holds:

$$\boldsymbol{J}(\boldsymbol{\theta}^{k+1}) - \boldsymbol{J}(\boldsymbol{\theta}^k) \geq \nabla\boldsymbol{J}(\boldsymbol{\theta}^k)^\intercal(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k) - \frac{L_{\boldsymbol{J}}}{2}\left\|\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k\right\|^2 \tag{22}$$

Since ϕ is assumed non decreasing in each component (Assumption 1), we can easily combine (21) and (22) to deduce:

$$\mathfrak{J}(\boldsymbol{\theta}^{k+1}) - \mathfrak{J}(\boldsymbol{\theta}^k) \geq \nabla\phi(\boldsymbol{J}(\boldsymbol{\theta}^k))^\intercal(\nabla\boldsymbol{J}(\boldsymbol{\theta}^k))(\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k)^\intercal - \left\|\phi(\boldsymbol{J}(\boldsymbol{\theta}^k))\right\|_1 \frac{L_{\boldsymbol{J}}}{2}\left\|\boldsymbol{\theta}^{k+1}-\boldsymbol{\theta}^k\right\|^2 \tag{23}$$

Using Assumption 1 to bound the gradient of ϕ, and given the definition of $\mathfrak{g}^{k,\star}$ (11), and the update rule (13) the above inequality turns into:

$$\mathfrak{J}(\boldsymbol{\theta}^{k+1}) - \mathfrak{J}(\boldsymbol{\theta}^k) \geq \alpha_k\langle\mathfrak{g}^{k,\star},\hat{\mathfrak{g}}^k\rangle - \frac{L_{\boldsymbol{J}}M_\phi}{2}\alpha_k^2\left\|\hat{\mathfrak{g}}^k\right\|^2 \tag{24}$$

Rewriting (24) with the error in (14), and using the condition on the learning rate (Assumption 5) would provide the result, since:

$$\begin{aligned}
\alpha_k\langle\mathfrak{g}^{k,\star},\hat{\mathfrak{g}}^k\rangle - \frac{L_{\boldsymbol{J}}M_\phi}{2}\alpha_k^2\left\|\hat{\mathfrak{g}}^k\right\|^2 \geq& \alpha_k\langle\mathfrak{g}^{k,\star},\hat{\mathfrak{g}}^k\rangle - \delta_k\left\|\hat{\mathfrak{g}}^k\right\|^2 \\
=&(\alpha_k\beta_k - \delta_k(\beta_k)^2)\langle\mathfrak{g}^{k,\star},\mathfrak{g}^{k,\star}\rangle + (\alpha_k - 2\delta_k\beta_k)\langle\mathfrak{g}^{k,\star},\boldsymbol{\epsilon}^k\rangle - \delta_k\left\|\boldsymbol{\epsilon}^k\right\|^2 \\
=&\gamma_k\langle\mathfrak{g}^{k,\star},\mathfrak{g}^{k,\star}\rangle - \delta_k\left\|\boldsymbol{\epsilon}^k\right\|^2,
\end{aligned}$$

where we set $\delta_k = \frac{\alpha_k}{2\beta_k}$, $\gamma_k = \frac{\alpha_k\beta_k}{2}$. The first inequality comes from Assumption 5, as it implies $\frac{L_{\boldsymbol{J}}M_\phi}{2}\alpha_k^2 \leq \delta_k$. As introduced previously, let us remind the constant β_k is the one stemming from the Policy Gradient Theorem, and corresponds in the episodic case to the average length of an episode. □

Before stating the main theorem, let us invoke a simple general lemma needed for evaluating the rate of convergence, in plain form and in probability.

Lemma A.6. (i) *If a sequence u_k satisfies an equation of the form:*

$$\forall k \leq k_0, \quad u_k - u_{k+1} \geq \mu u_k^2, \quad \textit{with } \mu \in \mathbb{R}^+, u_0 > 0, \tag{25}$$

Then, it has a sublinear rate of convergence:

$$\forall k \leq k_0, \quad u_k \leq \frac{u_0}{(\mu u_0)k + 1}$$

(ii) *If a sequence u_k satisfies:*

$$\forall k \leq k_0, \quad \mathsf{P}\left(u_k - u_{k+1} \geq \mu u_k^2\right) > 1 - \iota, \quad \textit{with } \mu \in \mathbb{R}^+, u_0 > 0. \tag{26}$$

Then, it has a sublinear rate of convergence with a certain probability:

$$\forall k \leq k_0, \quad \mathsf{P}\left(u_k \leq \frac{u_0}{(\mu u_0)k + 1}\right) \geq (1 - \iota)^k$$

Proof. (i) First, let us remark that the assumption (25) implies that (u_k) is a decreasing sequence. If the sequence reaches 0, the claim becomes trivial, so we would exclude this case in the proof, and focus on k such that $u_k > 0$. Exploiting the assumption in (25):

$$\frac{1}{u_k} - \frac{1}{u_{k-1}} = \frac{u_{k-1} - u_k}{u_{k-1}u_k} \geq \mu \frac{u_{k-1}}{u_k} \geq \mu. \tag{27}$$

By summing these inequalities, the following inequality holds:

$$\forall k \leq k_0, \quad \frac{1}{u_k} \geq \frac{1}{u_0} + k\mu \geq \frac{1 + k\mu u_0}{u_0}$$

It entails the claim (i) of the Lemma.

(ii) Transposing (27) to the probabilistic scenario, the assumption (26) provides the following inequality:

$$\mathsf{P}\left(\frac{1}{u_k} - \frac{1}{u_{k-1}} \geq \mu\right) > 1 - \iota$$

By considering the intersection of these events, we deduce:

$$\mathsf{P}\left(\frac{1}{u_k} - \frac{1}{u_0} \geq k\mu\right) \geq \mathsf{P}\left(\cap_{l=0}^{k}\left(\frac{1}{u_l} - \frac{1}{u_{l-1}} \geq \mu\right)\right) \geq (1 - \iota)^k$$

We can easily conclude. □

Let $\beta^\flat$ denote a lower bound for the average length of an episode, and $\alpha^\flat$ (resp. $\alpha^\sharp$) a lower (resp. upper bound) on the learning rate.

Theorem A.7. (i) *Under Assumptions (1, 2, 3, 4, 5, 6, 7, 8, 9), and assuming the size of the minibatch m tends towards infinity, the social welfare objective $\mathfrak{J}(\boldsymbol{\theta}^k)$ converges in probability and with a sub-linear convergence rate within a radius of convergence $\tilde{\mathfrak{r}}$ of the optimal value where:*

$$\tilde{\mathfrak{r}} = \frac{\Xi\left(2R_0 + \alpha^\sharp\right)}{2\beta^\flat}, \quad \textit{with} \quad \Xi \coloneqq (L_\phi B_A \Xi_J + M_\phi \Xi_A + M_\phi \Xi_L) M_{\boldsymbol{\pi}}.$$

More precisely, the suboptimality after k iterations may be bounded:

$$\lim_{m \to \infty} \mathsf{P}\left(\left(\mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^k)\right) \leq \max\left(\tilde{\mathfrak{r}}, \frac{\mathfrak{j}_0}{\kappa \mathfrak{j}_0 k + 1}\right)\right) = 1$$

$$\textit{with } \mathfrak{j}_0 \coloneqq \mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^0)\,,\ \kappa \coloneqq \frac{\upsilon - 1}{\upsilon}\frac{\alpha^\flat \beta^\flat}{2R_0^2}\,,\ \textit{and } \upsilon \coloneqq \left(1 + \frac{\alpha^\sharp}{2R_0}\right)^2$$

(ii) *Under Assumptions (1, 2, 3, 4, 5, 6, 7, 8, 9), the social welfare objective $\mathfrak{J}(\boldsymbol{\theta}^k)$ converges, with a certain probability, within the radius of convergence $\tilde{\mathfrak{r}}$ of the optimal value.*
More precisely, the suboptimality after k iterations may be bounded:

$$\forall \epsilon > 0, \forall k > 0, \quad \mathsf{P}\left(\mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^k) \leq \max\left(\tilde{\mathfrak{r}}_\epsilon, \frac{\mathfrak{j}_0}{\kappa \mathfrak{j}_0 k + 1}\right)\right) \geq 1 - k\iota_{\epsilon,m} + o(\iota_{\epsilon,m}), \tag{28}$$

$$\textit{with } \tilde{\mathfrak{r}}_\epsilon \coloneqq \frac{(\Xi + \epsilon)\left(2R_0 + \alpha^\sharp\right)}{2\beta^\flat}.$$

(iii) *Similarly, under Assumptions (1, 2, 3', 5, 6, 7, 8, 9), if $\boldsymbol{J}$ is simply locally concave, the SWO converges in probability and with a sublinear rate of convergence within the radius $\tilde{\mathfrak{r}}$ of a local optimum, under the limit where m tends toward infinity. Under these same assumptions, in the general case, we can, as in (28) bound the suboptimality relative to a local optima with a certain probability.*

This theorem implies that we can bound the number of iterations required to have $\left\|\mathfrak{J}(\boldsymbol{\theta}^k)\right\| < \epsilon$ in probability:

Corollary A.8. *Under Assumptions (1, 2, 3, 4, 5, 6, 7, 8, 9), with the previous notations:*

$$\forall \epsilon \; s.t \; \tilde{\mathfrak{r}} < \epsilon < \mathfrak{j}_0, \quad \forall k \geq k_\epsilon, \quad \lim_{m \to \infty} \mathsf{P}\left(\mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^k) < \epsilon\right) = 1, \quad \textit{with } k_\epsilon \coloneqq \frac{\upsilon}{\upsilon - 1} \frac{2R_0^2}{\alpha^\flat \beta^\flat}\left(\frac{1}{\epsilon} - \frac{1}{\mathfrak{j}_0}\right).$$

Assuming that the terms $\Xi_{\boldsymbol{J}}, \Xi_A, \Xi_L$ stemming from Assumptions 7, 8, 9 may be as small as desired, the convergence radius would tend towards 0, under the limit m tends to infinity. However, in case of interdependent agents, particularly in cooperative or competitive settings, it is not reasonable to neglect the term stemming from Assumption 9. In contexts where the interdependence between agents is tamer, these different advantages (of an agent depending on other agents actions) could average themselves out in (17).

Proof of Corollary A.8. Directly following from the assertion (i) of Theorem A.7, since it is straightforward to check that $\forall k \geq k_\epsilon, \frac{\mathfrak{j}_0}{\kappa \mathfrak{j}_0 k + 1} \leq \epsilon$. □

Proof of Theorem A.7. As previously stated in Lemma A.2, under Assumptions (1, 3), $\mathfrak{J}$ is concave. From the concavity of $\mathfrak{J}$, applying Cauchy−Schwartz inequality:

$$\mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^k) \leq \left\|\nabla\mathfrak{J}(\boldsymbol{\theta}^k)\right\| \left\|\boldsymbol{\theta}^* - \boldsymbol{\theta}^k\right\| \leq R_0 \left\|\nabla\mathfrak{J}(\boldsymbol{\theta}^k)\right\|, \tag{29}$$

where the last inequality stems from Assumption 4. Indeed, Assumption 4 is satisfied for $\boldsymbol{\theta}^0$ and it is easy to prove by recursion, that it holds for all the following $\boldsymbol{\theta}^k$'s; given (30), we can prove recursively $\mathfrak{J}(\boldsymbol{\theta}^k) \geq \mathfrak{J}(\boldsymbol{\theta}^0)$.

Combining (29) with Lemma A.5:

$$\mathfrak{J}(\boldsymbol{\theta}^{k+1}) - \mathfrak{J}(\boldsymbol{\theta}^k) \geq \kappa_k(\mathfrak{J}(\boldsymbol{\theta}^k) - \mathfrak{J}^*)^2 - \mathcal{E}_k = \kappa_k\left[(\mathfrak{J}(\boldsymbol{\theta}^k) - \mathfrak{J}^*)^2 - \mathfrak{d}_k^2\right] \quad \text{where} \begin{cases} \kappa_k &= \frac{\alpha_k \beta_k}{2R_0^2} \\ \mathcal{E}_k &= \frac{\alpha_k}{2\beta_k}\left\|\boldsymbol{\epsilon}^k\right\|^2 \\ \mathfrak{d}_k &= \sqrt{\frac{\mathcal{E}_k}{\kappa_k}} = \frac{\left\|\boldsymbol{\epsilon}^k\right\| R_0}{\beta_k} \end{cases}. \tag{30}$$

Rewriting (30), with $\mathfrak{j}_k = \mathfrak{J}^* - \mathfrak{J}(\boldsymbol{\theta}^k)$:

$$\mathfrak{j}_k - \mathfrak{j}_{k+1} \geq \kappa_k(\mathfrak{j}_k^2 - \mathfrak{d}_k^2), \tag{31}$$

Below, we first proceed to the convergence analysis while assuming the error term $\left\|\boldsymbol{\epsilon}^k\right\|$ is bounded by $\tilde{\Xi}$, in order to familiarise the reader with the proof's argument. In a second step, we refine this analysis, in order to work in probability.

Prior Analysis: In this prior analysis, we assume the error term is bounded by $\tilde{\Xi}$. It enables us to define an upper bound for $\mathfrak{d}_k$:

$$\mathfrak{d}_k \leq \mathfrak{r} \coloneqq \frac{\tilde{\Xi} R_0}{\beta^\flat} \tag{32}$$

From (30), we deduce that the sequence $(\mathfrak{J}(\boldsymbol{\theta}^{k+1}))_k$ is increasing at least until it reaches a distance $\mathfrak{r}$ from the optimal value $\mathfrak{J}^*$. Moreover, by setting $\mathfrak{r}^\upsilon \coloneqq \sqrt{\upsilon}\mathfrak{r}$, combining (31) and (32), we deduce:

$$\forall \upsilon > 1, \forall k, \quad \mathfrak{j}_k \geq \mathfrak{r}^\upsilon \Rightarrow \mathfrak{j}_k - \mathfrak{j}_{k+1} \geq \kappa_k \left(\mathfrak{j}_k^2 - \frac{(\mathfrak{r}^\upsilon)^2}{\upsilon} \right) \geq \frac{\upsilon - 1}{\upsilon} \kappa_k \mathfrak{j}_k^2 \geq \kappa^\upsilon \mathfrak{j}_k^2, \quad \text{with } \kappa^\upsilon \coloneqq \frac{\upsilon - 1}{\upsilon} \frac{\alpha^\flat \beta^\flat}{2R_0^2}. \tag{33}$$

From there, applying Lemma A.6 would enable us to roughly conclude that $\mathfrak{j}_k$ converges with a sublinear rate of convergence towards the optimal value, at least until it reaches $\mathfrak{r}$. More precisely, fixing k^υ such that $\mathfrak{j}_k > \mathfrak{r}^\upsilon$ for all $k < k^\upsilon$, Lemma A.6 asserts:

$$\forall k \leq k^\upsilon, \quad \mathfrak{j}_k \leq \frac{\mathfrak{j}_0}{\kappa^\upsilon \mathfrak{j}_0 k + 1} \tag{34}$$

Since this is valid for all $\upsilon > 1$, and $\lim_{\upsilon \to 1} \mathfrak{r}^\upsilon = \mathfrak{r}$, it suggests the sequence $\mathfrak{j}_k$ converges towards the radius $\mathfrak{r}$. However, this previous radius of convergence $\mathfrak{r}$ is not guaranteed to be stable. Indeed, once reached the radius $\mathfrak{r}$, the right side of (31) would be negative, and we can not conclude the sequence $(\mathfrak{j}_k)$ is still decreasing. However, we can rethink a stable radius of convergence. Let us choose k_1 —assuming it exists— such that $\mathfrak{j}_{k_1}$ lies within the radius $\mathfrak{r}$, but $\mathfrak{j}_{k_1 - 1}$ does not. Then, rewriting (31):

$$\mathfrak{j}_{k_1} - \mathfrak{j}_{k_1 - 1} \leq \kappa_{k_1 - 1} \left(\mathfrak{d}_{k_1 - 1}^2 - \mathfrak{j}_{k_1 - 1}^2 \right) \tag{35}$$

Therefore, from the previous definitions and assumptions:

$$\|\mathfrak{j}_{k_1}\| \leq \|\mathfrak{j}_{k_1 - 1}\| + \kappa_{k_1 - 1} \left\| \mathfrak{d}_{k_1 - 1}^2 - \mathfrak{j}_{k_1 - 1}^2 \right\| \leq \mathfrak{r} + \kappa_{k_1 - 1} \mathfrak{d}_{k_1 - 1}^2 = \mathfrak{r} + \mathcal{E}_{k_1 - 1} \leq \frac{\tilde{\Xi}}{\beta^\flat} R_0 + \frac{\alpha_{k_1 - 1}}{2\beta_{k_1 - 1}} \tilde{\Xi} \leq \tilde{\mathfrak{r}}, \tag{36}$$

where $\alpha^\sharp$ is an upper bound over the learning rate, and with our redefined radius of convergence:

$$\tilde{\mathfrak{r}} \coloneqq \frac{\tilde{\Xi}}{\beta^\flat} \left(R_0 + \frac{\alpha^\sharp}{2} \right).$$

By (36), $\mathfrak{j}_{k_1}$ lies within this radius. We claim that all the following values $(\mathfrak{j}_k)_{k > k_1}$ stay within the radius $\tilde{\mathfrak{r}}$. Indeed, by (31), we can deduce that if $\mathfrak{j}_{k_1} > \mathfrak{r}$, the successive values $(\mathfrak{j}_k)_{k > k_1}$ will be again decreasing at least until reaching the radius $\mathfrak{r}$. Therefore, even when some values in the sequence $(\mathfrak{j}_k)$ ventures out of the original radius of convergence $\mathfrak{r}$, they will stay within the new radius of convergence $\tilde{\mathfrak{r}}$.

Analysis in Probability In order to finalise the theorem proof, we will have to adapt the above arguments in probability. First, let us examine the error term, which, from the decomposition (14), satisfies:

$$\left\| \boldsymbol{\epsilon}^k \right\| \leq \left\| \boldsymbol{\nu}^k \right\| + \left\| \boldsymbol{\eta}^k \right\|$$

A consequence of the Chebyshev inequality, upon assumption 6 bounding the variance of the estimated gradient:

$$\forall \epsilon > 0, \quad \mathsf{P}\left(\left\| \boldsymbol{\nu}^k \right\| < \epsilon \right) = \mathsf{P}\left(\left\| \hat{\mathfrak{g}}^k - \tilde{\mathfrak{g}}^k \right\| < \epsilon \right) \geq 1 - \frac{(\sigma^\sharp)^2}{\epsilon^2 m},$$

where m is the size of the minibatch on which the gradient $\hat{\mathfrak{g}}^k$ is estimated. Meanwhile, as demonstrated in Lemma A.3, $\left\| \boldsymbol{\eta}^k \right\|$ may be bounded by Ξ in probability, under Assumptions 1, 2, 6, 7, 8, 9. Hence:

$$\forall \epsilon > 0, \quad \mathsf{P}\left(\left\| \boldsymbol{\epsilon}^k \right\| < \Xi_\epsilon \right) \geq 1 - \iota_{\epsilon, m}, \quad \text{with } \iota_{\epsilon, m} \coloneqq \frac{(\sigma^\sharp)^2}{\epsilon^2 m}, \text{ and } \Xi_\epsilon \coloneqq \Xi + \epsilon \tag{37}$$

In particular, upon taking the limit over m:

$$\forall \epsilon > 0, \quad \lim_{m \to \infty} \mathsf{P}\left(\left\| \boldsymbol{\epsilon}^k \right\| < \Xi_\epsilon \right) = 1. \tag{38}$$

Instead of the (32) from the prior analysis, we get the following probabilistic form:

$$\forall \epsilon > 0, \quad \mathsf{P}\left(\mathfrak{d}_k \leq \mathfrak{r}_\epsilon\right) \geq 1 - \iota_{\epsilon,m}, \quad \text{where } \mathfrak{r}_\epsilon \coloneqq \frac{\Xi_\epsilon R_0}{\beta^\flat}. \tag{39}$$

By setting $\mathfrak{r}_\epsilon^\upsilon \coloneqq \sqrt{\upsilon}\mathfrak{r}_\epsilon$, it induces the following —since $\mathfrak{j}_k^2 \geq \upsilon \mathfrak{d}_k^2 \Rightarrow \mathfrak{j}_k^2 - \mathfrak{d}_k^2 \geq \mathfrak{j}_k^2 \frac{\upsilon - 1}{\upsilon}$:

$$\forall \epsilon > 0, \forall \upsilon > 1, \forall k, \quad \mathsf{P}\left(\mathfrak{j}_k^2 - \mathfrak{d}_k^2 \geq \frac{\upsilon - 1}{\upsilon}\mathfrak{j}_k^2 \mid \mathfrak{j}_k \geq \mathfrak{r}_\epsilon^\upsilon\right) \geq \mathsf{P}\left(\mathfrak{j}_k^2 \geq \upsilon \mathfrak{d}_k^2 \mid \mathfrak{j}_k \geq \mathfrak{r}_\epsilon^\upsilon\right) \geq \mathsf{P}\left(\mathfrak{d}_k \leq \mathfrak{r}_\epsilon\right) \geq 1 - \iota_{\epsilon,m}.$$

With such an inequality in hand, we deduce from (31):

$$\forall \epsilon > 0, \forall \upsilon > 1, \quad \mathsf{P}\left(\mathfrak{j}_k - \mathfrak{j}_{k+1} \geq \kappa^\upsilon \mathfrak{j}_k^2 \mid \mathfrak{j}_k \geq \mathfrak{r}_\epsilon^\upsilon\right) \geq 1 - \iota_{\epsilon,m}, \text{ where } \kappa^\upsilon \coloneqq \frac{\upsilon - 1}{\upsilon}\frac{\alpha^\flat \beta^\flat}{2R_0^2} \tag{40}$$

Notably, passing to the limit over υ in (40), since $\mathfrak{r}_\epsilon = \lim_{\upsilon \to 1} \mathfrak{r}_\epsilon^\upsilon$:

$$\mathsf{P}\left(\mathfrak{j}_k \geq \mathfrak{j}_{k+1} \mid \mathfrak{j}_k \geq \mathfrak{r}_\epsilon\right) \geq 1 - \iota_{\epsilon,m} \tag{41}$$

The sequence $\mathfrak{j}_k$ is therefore decreasing until it reaches $\mathfrak{r}_\epsilon$, with a certain probability. At the limit where $m \to \infty$, $\mathfrak{j}_k$ is almost surely decreasing, until reaching the radius $\mathfrak{r}$, since we can tie tying ϵ to m such that $\lim_{m \to \infty, \epsilon \Rightarrow \infty} \iota_{epsilon,m} = 0$, e.g. with $\epsilon = m^{-\frac{1}{3}}$. Applying Lemma A.6 to the last inequality would enable us to roughly conclude that $\mathfrak{j}_k$ converges with a sublinear rate a convergence with a certain (bounded) probability until it reaches a radius $\mathfrak{r}_\epsilon$.

More precisely, let us fix k_ϵ^υ such that $\forall k < k_\epsilon^\upsilon, \mathfrak{j}_k > \mathfrak{r}_\epsilon^\upsilon$. Then, similarly to (34), Lemma A.6 leads to:

$$\forall k \leq k_\epsilon^\upsilon, \quad \mathsf{P}\left(\mathfrak{j}_k \leq \frac{\mathfrak{j}_0}{\mathfrak{j}_0 \kappa^\upsilon k + 1}\right) \geq \left(1 - \iota_{\epsilon,m}\right)^k = 1 - k\iota_{\epsilon,m} + o(\iota_{\epsilon,m}) \tag{42}$$

As this is valid for all $\upsilon > 1$ (although it changes the factor κ^υ of the convergence rate), it signifies the sequence $\mathfrak{j}_k$ is decreasing until k and converges towards $\mathfrak{r}_\epsilon$ with a certain probability.

However, as highlighted previously in (i), this radius $\mathfrak{r}_\epsilon$ is not guaranteed to be stable. Let us fix $k_{0,\epsilon}$ the first k such that $\mathfrak{j}_k \leq \mathfrak{r}_\epsilon$ and $k_{1,\epsilon}$ the first $k > k_{0,\epsilon}$, such that $\mathfrak{j}_k > \mathfrak{r}_\epsilon$ again —assuming they exists. Similarly to (36), from (35):

$$\mathfrak{j}_{k_{1,\epsilon}} \leq \mathfrak{j}_{k_{1,\epsilon}-1} + \kappa_{k_{1,\epsilon}-1}\left\|\mathfrak{d}_{k_{1,\epsilon}-1}^2 - \mathfrak{j}_{k_{1,\epsilon}-1}^2\right\| \leq \mathfrak{r}_\epsilon + \kappa_{k_{1,\epsilon}-1}\mathfrak{d}_{k_{1,\epsilon}-1}^2 = \mathfrak{r}_\epsilon + \mathcal{E}_{k_{1,\epsilon}-1}, \tag{43}$$

From the assumption on the error term and definition of $\mathcal{E}_k$, with the previous notations:

$$\forall k, \quad \mathsf{P}\left(\mathcal{E}_k < \frac{\alpha^\sharp}{2\beta^\flat}\Xi_\epsilon\right) > 1 - \iota_{\epsilon,m} \tag{44}$$

Combining (43) and (44), and introducing the new convergence radius, it implies:

$$\mathsf{P}\left(\mathfrak{j}_{k_{1,\epsilon}} \leq \tilde{\mathfrak{r}}_\epsilon\right) \geq 1 - \iota_{\epsilon,m}, \text{ where } \tilde{\mathfrak{r}}_\epsilon \coloneqq \frac{\Xi_\epsilon}{\beta^\flat}\left(R_0 + \frac{\alpha^\sharp}{2}\right).$$

By (40) and the initial assumption on $k_{1,\epsilon}$, the sequence $\mathfrak{j}_k$ is now decreasing with probability $(1 - \iota_{\epsilon,m})$. Thereupon, the following value $\mathfrak{j}_{k_{1,\epsilon}+1}$ satisfies:

$$\mathsf{P}\left(\mathfrak{j}_{k_{1,\epsilon}+1} \leq \tilde{\mathfrak{r}}_\epsilon\right) \geq \mathsf{P}\left(\mathfrak{j}_{k_{1,\epsilon}} \leq \tilde{\mathfrak{r}}_\epsilon\right)\mathsf{P}\left(\mathfrak{j}_{k_{1,\epsilon}+1} \leq \mathfrak{j}_{k_{1,\epsilon}}\right) \geq \left(1 - \iota_{\epsilon,m}\right)^2$$

Iterating such argument, would lead us to the following bound:

$$\forall k > k_{1,\epsilon}, \quad \mathsf{P}\left(\mathfrak{j}_k \leq \tilde{\mathfrak{r}}_\epsilon\right) \geq \left(1 - \iota_{\epsilon,m}\right)^{k - k_{1,\epsilon}} \geq \left(1 - \iota_{\epsilon,m}\right)^k. \tag{45}$$

This argument may be repeated for successive values of the sequence j_k, if such behaviors happens. Gathering the previous results, (42) and (45), leads us to the following claim:

$$\forall \epsilon > 0, \forall k > 0, \quad \mathsf{P}\Big[\mathsf{j}_k \leq \max\left(\tilde{\mathfrak{r}}_\epsilon, \frac{\mathsf{j}_0}{\kappa^\upsilon \mathsf{j}_0 k + 1}\right)\Big] \geq (1 - \iota_{\epsilon,m})^k = 1 - k\iota_{\epsilon,m} + o(\iota_{\epsilon,m}), \tag{46}$$

under the condition of υ satisfying:

$$\mathfrak{r}_\epsilon^\upsilon \leq \tilde{\mathfrak{r}}_\epsilon \iff \sqrt{\upsilon} R_0 \leq \left(R_0 + \frac{\alpha^\flat}{2}\right) \iff \upsilon \leq \left(1 + \frac{\alpha^\flat}{2R_0}\right)^2.$$

For the theorem, we adopted $\upsilon = \left(1 + \frac{\alpha^\flat}{2R_0}\right)^2$.

Limit case: As previously mentionned, by taking the limit $m \to \infty$, and $\epsilon \to 0$, such that $\epsilon^2 m \to \infty$ (e.g. $\epsilon = m^{-\frac{1}{3}}$), $\iota_{\epsilon,m}$ would tend towards 0. Hence, (46) reveals that j_k converges in probability within the radius $\tilde{\mathfrak{r}}$ with a sublinear rate of convergence, as claimed.

Locally Concave Case: The case of assuming $\boldsymbol{J}$ simply locally concave would lead to a similar convergence result, under the corresponding set of assumptions, yet for a local optimum on the neighborhood $\mathcal{U}$. Since the proof is almost identical to the previous case, it is left to the reader.

□

B. Additional Experimental Results

We present here the additional experimental results that were not included in the main paper. Through those experiments, we try to answer the following questions:

- **(A)** How do SOTO and FEN perform with unequal access to resources in a controlled domain?
- **(B)** Can SOTO cope with a different number of users and agents?
- **(C)** What if Basic could also learn from $\hat{\boldsymbol{A}}_i^{\text{IND}}$ then $\hat{\boldsymbol{A}}^{\text{SWF}}$ gradually with β?
- **(D)** What if we use the same architecture proposed in FEN to optimize a SWF?
- **(E)** Where does the bad performance of FEN in some domains come from?
- **(F)** Does SOTO still outperform FEN if it is also allowed to share scores to neighbors?
- **(G)** Is Basic capable enough to find fair solutions in small problems?

Question (A) In those experiments, we evaluate how our method performs when there is an unequal access to resources. We modified the Matthew Effect environment to restrict one of the 3 resources to be only collectible by the first two agents. They have an additional observation to determine whenever the resource is restricted. Other agents neither observe nor collect those restricted resources.

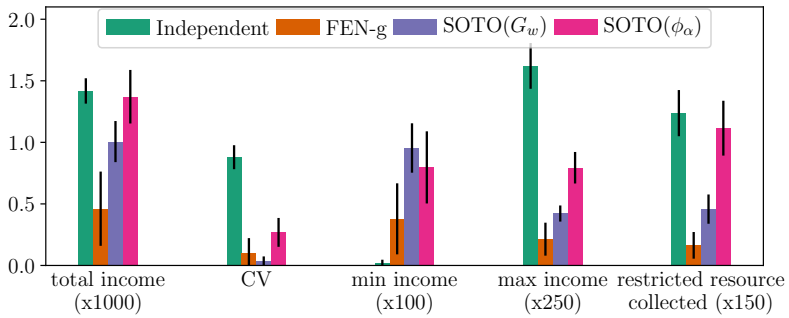

Figure 10: Comparison of Independent, FEN, and SOTO on Matthew Effect with unequal access to resources in the CLDE scenario. Higher total income is better, lower CV is better, higher min income is better, higher max income is better and higher restricted resource collected is better.

Compared to Matthew Effect without resource restriction (Figure 17), in Figure 10, as expected, Independent achieves a lower total income since it is more difficult to collect resources for the majority of the agents. FEN is Pareto-dominated by SOTO($G_{\boldsymbol{w}}$). Because of the unequal access to resources, the CV criterion has less importance in this experience: the first two agents should collect the maximum of restricted resources. This behavior is achieved by SOTO(ϕ_α), which collected a high number of restricted resources, which also Pareto-dominates FEN if we ignore the CV criterion. This shows that our method can cope with unequal access to resources whereas the first two agents of FEN do not collect more resources than the others.

Question (B) To answer (**B**), we turn to a more complex SUMO environment where we split the objectives per road instead of per intersection. The objective is to minimize the waiting time of the two roads of an intersection, hence each agent observes two users ($2N = D$). Accordingly, the output size of the critics in the SOTO architecture is two.

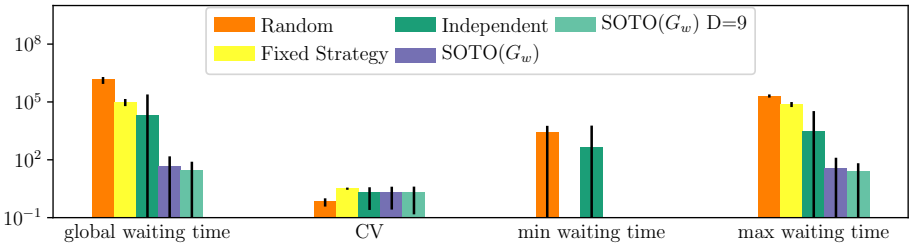

Figure 11: SUMO simulation with D=18 in the CLDE scenario. Lower global waiting time is better, lower CV is better, lower min waiting time is better and lower max waiting time is better.

In Figure 11, we can see that our method copes with a different number of users and agents. However, this artificial splitting does not allow for a fairer solution than the one when $D = 9$.

Question (C) To verify that the SOTO architecture is necessary and the benefits do not come only from training with β, we performed another control experiment where Basic is also trained with the two different objectives: $\hat{\boldsymbol{A}}_i^{\text{IND}}$ and then $\hat{\boldsymbol{A}}^{\text{SWF}}$.

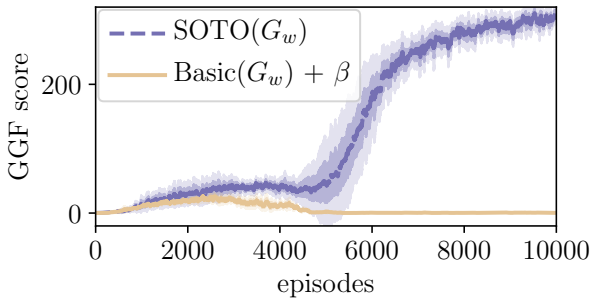

Figure 12: Comparison between SOTO and Basic updated by the two objectives with β on Matthew Effect.

Figure 12 can be analyzed in two parts. In the first stage, when $\hat{\boldsymbol{A}}_i^{\text{IND}}$ is mostly used to update the network, both approaches achieve an almost similar performance. However, the more $\hat{\boldsymbol{A}}^{\text{SWF}}$ is used, the more catastrophic forgetting intervenes. This demonstrates the importance of having two separate networks for each of the objectives.

Question (D) Here we want to verify if we could use the FEN architecture instead of the SOTO one. To do so, we extend FEN such that the meta-controller is rewarded by the current GGF value (FEN-g GGF). We also extended FEN such that the meta-controller is updated with $\hat{\boldsymbol{A}}^{\text{SWF}}$ (FEN-g A^{SWF}).

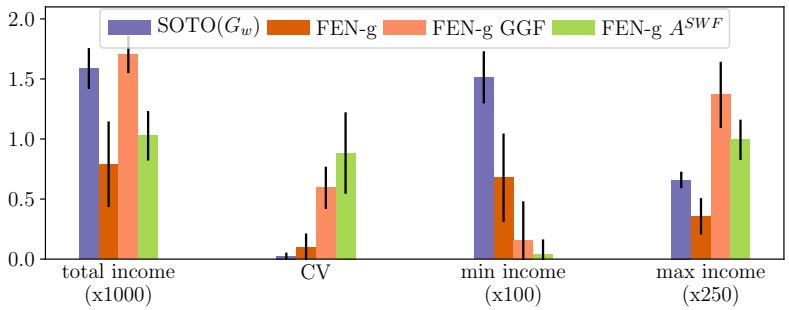

Figure 13: Comparison between SOTO, FEN without gossip, and FEN without gossip when it optimizes a SWF on the Matthew Effect. Higher total income is better, lower CV is better, higher min income is better, higher max income is better and higher restricted resource collected is better.

Figure 13 shows that both baselines are able to learn a good behavior (in terms of total income). However, none of them learn to achieve the better CV than FEN-g and SOTO. From this experiment, we conclude again that the SOTO architecture is essential to optimize a SWF without a centralized critic.

Question (E) Since FEN was not able to learn a good policy on the SUMO domain, we also tried to perform a pretraining of the first sub-policy as suggested by the authors for Plant Manufacturing. For the first 400 episodes, the first sub-policy is trained without interventions of the meta-controller.

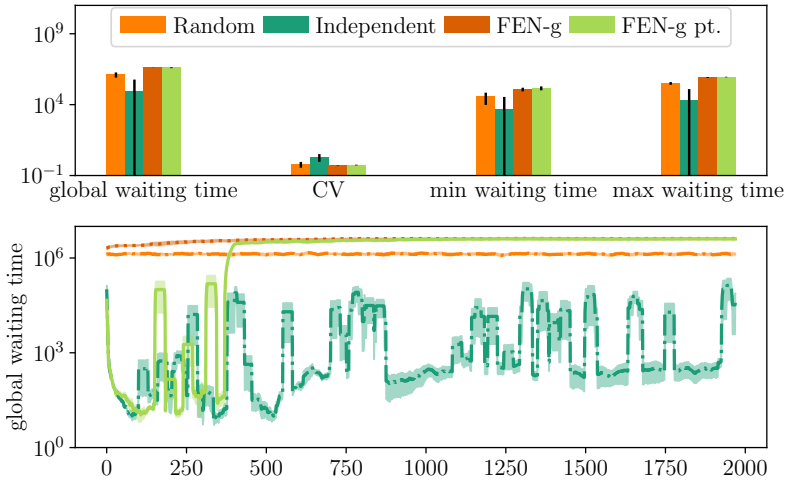

Figure 14: Comparison between FEN without gossip and FEN without gossip with pretraining (without gossip). Lower global waiting time is better, lower CV is better, lower min waiting time is better and lower max waiting time is better.

Figure 14 shows that during the first 400 episodes, the performance of FEN and Independent are indeed similar. However, when the meta-controller starts to switch other sub-policies the divergence begins. Note that when the pretraining is over, the first sub-policy is not updated anymore. So, if the performance drops it means that the meta-controller selects less and less often the first sub-policy. This is probably due to the fact that several agents select a wrong sub-policy at the same time, so the average performance is greatly reduced. Therefore, the other agents who chose the first sub-policy will also choose bad sub-policies for the next time to reduce the CV leading to a vicious circle until the worst possible average is reached. We observed this phenomenon in several domains.

Question (F) To verify that our better performances are not due to the fact that our networks have more input information, we provided FEN with the same additional inputs for its neural networks (i.e. the $\boldsymbol{J}(\boldsymbol{\theta})$ of the neighbors).

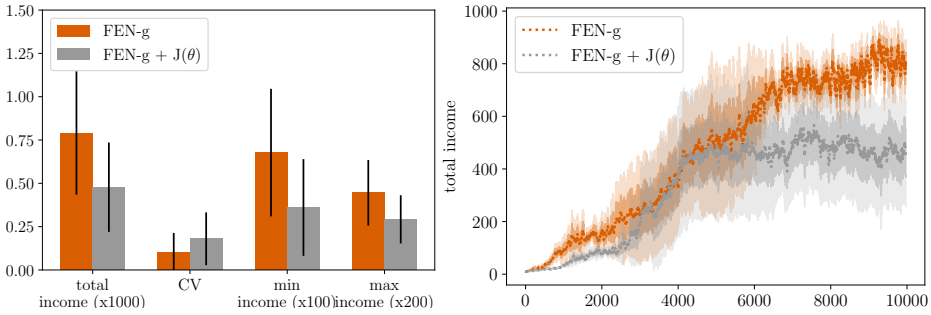

Figure 15: Comparison between FEN and FEN with more communications (without gossip).

From Figure 15, we can confirm that the additional communications are not helping FEN in Matthew Effect.

Question (G) In order to answer (G), we discuss the experimental results of Independent, FEN, Basic and SOTO on Job Scheduling in the CLDE scenario. Figure 16 illustrates the performances in terms of resource utilization, CV, minimum and maximum of utilities. As expected, Independent performs worse as it has the highest CV. Our algorithms achieve lower CV than FEN which indicates Basic and SOTO methods are able to find more fairer solution than FEN. Among all the algorithms, Basic($G_{\boldsymbol{w}}$) performs the best as it has the lowest CV. From these results, we can conclude that in simple domains Basic can find optimal solution without our proposed neural network architecture while in difficult task such as in Matthew Effect 17 and Plant Manufacturing 18 it performs worse.

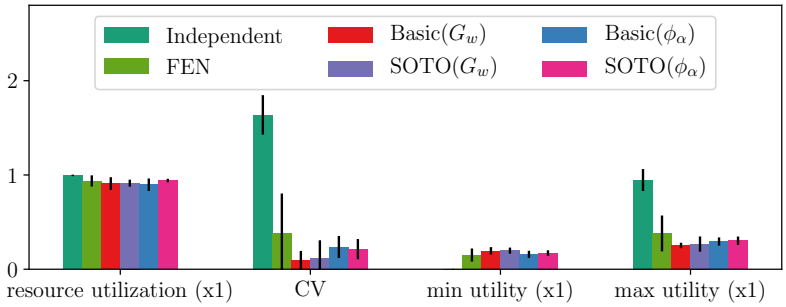

Figure 16: Comparison of Independent, FEN, Basic and SOTO on Job Scheduling in the CLDE scenario.

C. Complete Experimental Results Per Domain

In this part, we provide the results on the Job Scheduling, Matthew Effect and Plant Manufacturing environments (Table 1, Figure 17 and Figure 18). We also present an additional plot to visualize the solutions on Matthew Effect in the FD scenario (Figure 4).

C.1. Job Scheduling

Table 1: Job scheduling benchmark. The first part refers to the CLDE scenario. The second part refers to the FD scenario.

Method	Resource utilization	CV	min utility	max utility
Independent	1.00 ±0.00	1.64 ±0.19	0.00 ±0.00	0.95 ±0.11
FEN without gossip	0.93 ±0.07	0.37 ±0.42	0.15 ±0.07	0.37 ±0.19
Basic($G_{\boldsymbol{w}}$)	0.91 ±0.07	0.10 ±0.10	0.20 ±0.04	0.25 ±0.03
SOTO($G_{\boldsymbol{w}}$)	0.91 ±0.04	0.12 ±0.19	0.20 ±0.03	0.27 ±0.08
Basic(ϕ_α)	0.90 ±0.07	0.24 ±0.12	0.16 ±0.04	0.29 ±0.04
SOTO(ϕ_α)	0.94 ±0.02	0.23 ±0.10	0.16 ±0.03	0.31 ±0.04
Independent	1.00 ±0.00	1.64 ±0.19	0.00 ±0.00	0.95 ±0.11
FEN	0.72 ±0.24	0.84 ± 0.38	0.04 ±0.03	0.44 ±0.26
FD Basic($G_{\boldsymbol{w}}$)	0.79 ±0.22	1.12 ±0.46	0.02 ±0.03	0.57 ±0.28
FD SOTO($G_{\boldsymbol{w}}$)	0.87 ±0.12	1.07 ±0.42	0.02 ±0.04	0.60 ±0.23
FD Basic(ϕ_α)	0.71 ±0.31	1.36 ±0.42	0.01 ±0.02	0.61 ±0.34
FD SOTO(ϕ_α)	0.94 ±0.11	1.61 ±0.26	0.00 ±0.00	0.89 ±0.18

C.2. Matthew Effect

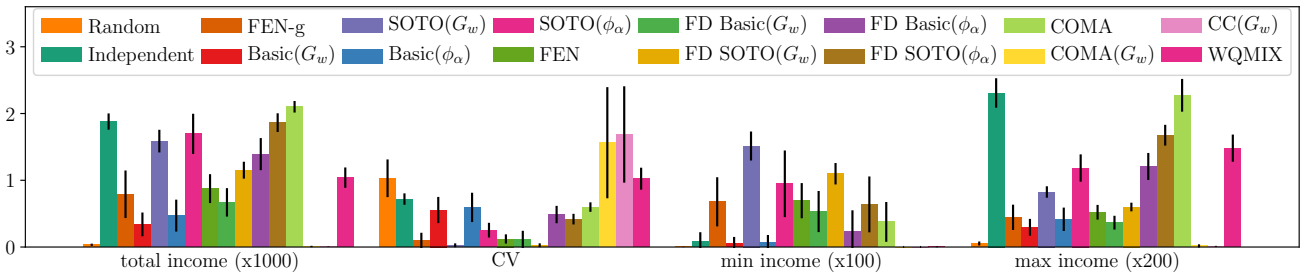

Figure 17: Comparison of different methods on Matthew Effect.

C.3. Plant Manufacturing

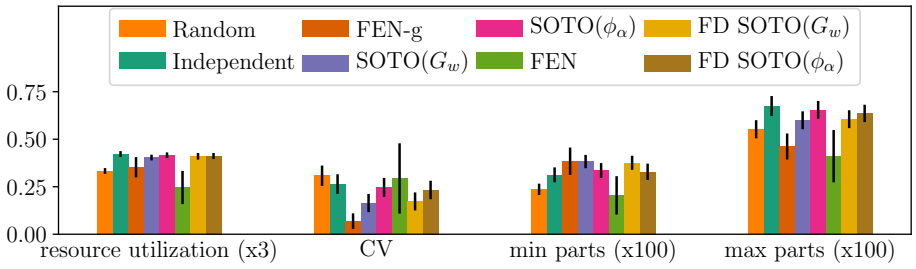

Figure 18: Comparison of different methods on Plant Manufacturing.

In Plant Manufacturing, Figure 18, in both scenarios, SOTO(G_w) and FEN are able to build the maximum number of products (equivalent to the minimum number of parts for each agent).

D. More details on experimentation

Note that compared to the results presented in (Jiang & Lu, 2019), our Independent method works almost twice better because we normalize the inputs, used Generalized Advantage Estimation (Schulman et al., 2016) instead of pure Monte-Carlo, and added bias units to the neural networks. All the presented methods benefit from those improvements.

In both scenarios (CLDE and FD), FEN is not restricted to send messages to neighbors for its gossip algorithm. To emphasize decentralization, all the agents are learning independent parameters without weights sharing (except for the COMA and WQMIX baselines).

The source code is available at `https://gitlab.com/AAAL/DFRL` with the configuration files used to generate our experimental results. The hyperparameters of the algorithms were optimized by grid search using Lightweight HyperParameter Optimizer (LHPO)[10], an open source library used to run asynchronous distributed experiments (Zimmer, 2018).

The complete SOTO architecture (including critics) is provided in Figure 19.

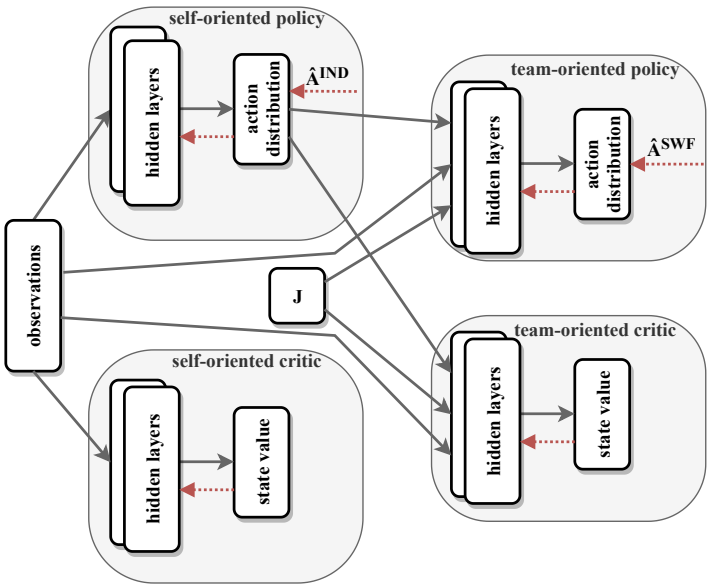

Figure 19: The complete SOTO architecture.

[10] `https://github.com/matthieu637/lhpo`

D.1. Hyperparameters

For all the experiments, we use PPO. The hidden layers of the neural networks are composed of two ReLU layers with 256 units. We used the ADAM (Kingma & Ba, 2015) optimization method with 2.5×10^{-4} as learning rate for the actor and 1×10^{-3} for the critic. The exploration bonus and the clipping ratio of the importance sampling in PPO are set to 0.03 and 0.1 respectively. In both scenarios, the FEN gossip algorithm uses $g = 10$ gossip rounds and $\tilde{k} = 3$ agents receiving the message. In every environment, we used Generalized Advantage Estimation with $\lambda = 0.97$ except in the Job Scheduling environment where we kept pure Monte-Carlo estimation to reproduce the state-of-the-art results. The discount factor γ is 0.98 for Plant Manufacturing, Matthew effect and Job Scheduling, 0.99 for SUMO and 0.95 for Iroko. Batch sizes are respectively 25, 50, 50, 50 and 128 in Job Scheduling, Matthew Effect, Plant Manufacturing, SUMO and Iroko. In continuous action space (Iroko), we tuned the standard deviation of the Gaussian distribution for each instance of algorithms instead of learning it (so it is independent of the state). The best standard deviations for Independent, FEN-g, SOTO(G_w) and FD SOTO(G_w) are 0.5, 0.5, 0.05, and 0.1 respectively.

D.2. Hardware

We performed the experiments on different computers equipped with 2 x Intel Xeon CPU E5-2678 v3 or Intel Core i7-8700 for SUMO and Iroko and 2 x Intel Xeon E5-2630 v3 or 2 x Intel Xeon Gold 6130 for the other domains.

Table 2: Comparison of the average computation times in hours.

Method	Job Scheduling	Matthew Effect	Plant Manufacturing	SUMO (D=9)	Iroko
Independent	9	27	8	20	5
FEN-g	13	37	64	60	6
Basic($G_{\boldsymbol{w}}$)	11	30			
SOTO($G_{\boldsymbol{w}}$)	11	34	12	14	7
Basic(ϕ_α)	11	23			
SOTO(ϕ_α)	14	29	11	27	
FEN	14	34	80+		
FD Basic($G_{\boldsymbol{w}}$)	11	26			
FD SOTO($G_{\boldsymbol{w}}$)	13	27	12	14	7
FD Basic(ϕ_α)	14	26			
FD SOTO(ϕ_α)	16	31	11		
Random		4	8	45	4
Fixed Strategy				45	4
COMA	13	23			
COMA($G_{\boldsymbol{w}}$)	15	29			
CC($G_{\boldsymbol{w}}$)	13	24			
WQMIX	24	80+		70	

In Table 2, the large computation time of FEN in Plant Manufacturing is due to the fact that the episode can last 10000 steps if all the 800 gems have not been collected.

D.3. Environments

Job Scheduling In Job Scheduling, a permanent and unique resource is placed on a grid with 4 agents, which they must learn to share it. In Job Scheduling, neighbors are limited to being one block away from the agent (the number of neighbors may vary over time). The different neighborhoods correspond to the grid observations.

Plant Manufacturing In Plant Manufacturing, 3 types of gems are randomly placed in a grid along with 5 agents. Once a gem is collected, it reappears at a random location with a random type. Each agent needs a specific combination of the gem types to build a part. Once each agent have a part, a product is built. However, the agents are not aware of the complete definition of that reward function, they are rewarded for each gem and part, not for the full product.

In Plant Manufacturing, neighbors are limited to being two blocks away from the agent (the number of neighbors may vary

over time). The different neighborhoods correspond to the grid observations.

The reported parameter of the Plant Manufacturing between the original paper and the code provided by Jiang & Lu (2019) differed[11]. Thus, we run all the algorithms with the following parameters: 800 gems are available for an episode, the reward bonus for collecting a gem is 0.01, the size of a minibatch is 50, and the type of resource is encoded by a one-hot vector.

Traffic Light Control We evaluate our approach in classic traffic light control problems. In such problems the goal is usually to minimize the accumulated waiting time of all vehicles over all agents. We consider the accumulated waiting time for each agent at each intersection separately. We use Simulation of Urban MObility (SUMO)[12] to simulate a 3x3 intersection grid that has total of 9 agents where each agent controls the traffic light phase of one intersection. Each intersection has 4 roads and a total of 8 lanes with different numbers of vehicles. A traffic light phase controls the traffic and specifies which lanes have the green light. In our setting, we assume that each intersection have 4 phases. Depending on the type of intersections, different types of phase pattern can be considered.

The state space is composed of the current traffic light phase and for each lane, its queue and density of cars stopped at the intersection. An action corresponds to choose the current traffic light phase. The environment simulates intersections for 5000 seconds, with an episode length of 500 decision steps. At each time step, new vehicles enter into the intersection with a fixed destination.

The minibatch size used is 50 and $\gamma = 0.99$. The neighborhood definition is based on the position on the grid (the number of neighbors vary from 2 to 4).

The reward function is defined as the total waiting time at the current intersection subtracted from the accumulated waiting time at the last step. This definition of reward function motivates agent to minimize the accumulated waiting time which is our objective in this domain. We have performed experiments in both cases where the reward is the accumulated waiting time for each agent ($D = N$) and where each agent's reward is further split into two components corresponds to two roads of the network ($D = 2N$).

Note that in this simulation if the agents' decisions are too bad, a situation of complete blockage can occur. If this is the case, then regardless of the actions taken by the agents for the rest of the episode, traffic will remain blocked everywhere.

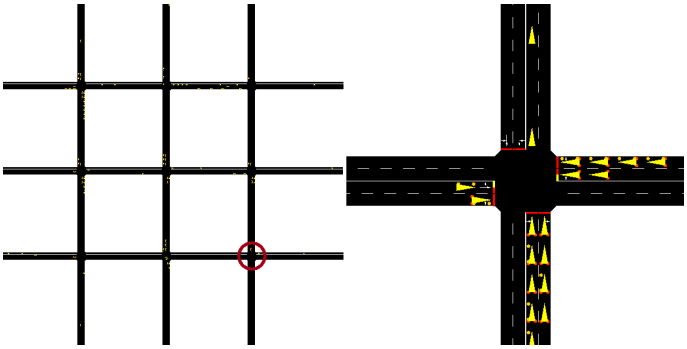

Figure 20: SUMO simulation with nine intersections in a grid. A zoom on one intersection is depicted at the bottom left. Cars can arrive from any direction and go to any of the 3 remaining ones.

Data Center Control In the Data Center Traffic Control problem, controllers manage the computer network that is shared by certain number of hosts. For the network topology, a fat-tree topology is considered (see Figure 21). In this topology, there are 16 hosts that are connected at the bottom with 20 switches. Each switch is connected with 4 ports, which results in a total of 80 queues in the network. We used Mininet[13] to simulate the network and Gobin[14] to generate network traffic. In classic data center control problems, the goal is usually to maximize the sum of host bandwidths in order to avoid the network congestion. However, we instead aim at maximizing the bandwidth of each host/agent in order to ensure fairness.

[11] Authors confirmed it in private communications. They also mentioned that a pretraining of the first sub-policy was required in this environment.

[12] https://github.com/eclipse/sumo

[13] https://github.com/mininet/mininet

[14] https://github.com/udhos/goben

The minibatch size used is 128 and the $\gamma = 0.95$. We consider the neighbors as the 3 closest hosts which can be linked with a maximum of two switches. The total number of agents are 16, where each agent is dedicated to each host.

The global state is a $n \times m$ matrix, where n is the number of ports in each switch and m is the number of network features collected by Goben. The continuous action corresponds to the allowed bandwidth for each host.

The D-dimensional reward vector is defined as follows:

$$\boldsymbol{r}_{\boldsymbol{a},s} = \boldsymbol{a} - 2 * \boldsymbol{a} * \max_i q_i(s)$$

where $\boldsymbol{a}$ is the vector action that represents the bandwidth allocation and $q_i(s)$ represents the i-th queue length. The reward is a vector, adapted from the (Ruffy et al., 2019), whose components are bandwidths per host penalized by the maximum of queue lengths. In the original environment, the average of $\boldsymbol{a}$ is taken to define a scalar reward.

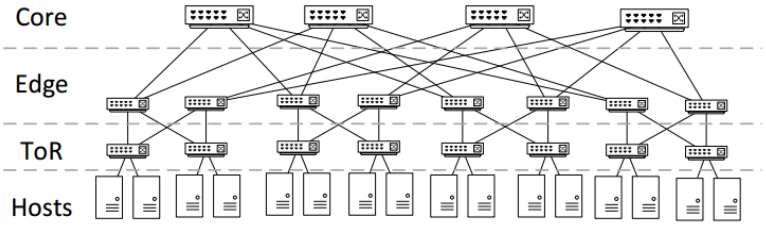

Figure 21: Network with a fat-tree topology from (Ruffy et al., 2019).