Rebuttal Material for Paper # 4515

This document contains the following items:

- A new modified Assurance game and new MANSA CL call plot
- Area under curve results for StarCraft Multi-Agent Challenge
- Area under curve results for Level-Based Foraging
- MANSA SMAC maps with extra seeds
- Win rate training curves for MANSA with restriction on CL updates and baselines
- Win rates for MANSA-B with restriction on CL updates and baselines
- Pseudocode for MANSA

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

Table 1: Modified reward functions of Assurance Game.

	AUC	std
MANSA	0.667676	0.013114
QMIX	0.621228	0.048804
IQL	0.3949	0.009503

Table 2: Mean and standard deviation of each algorithm at each step in each map from SMAC.

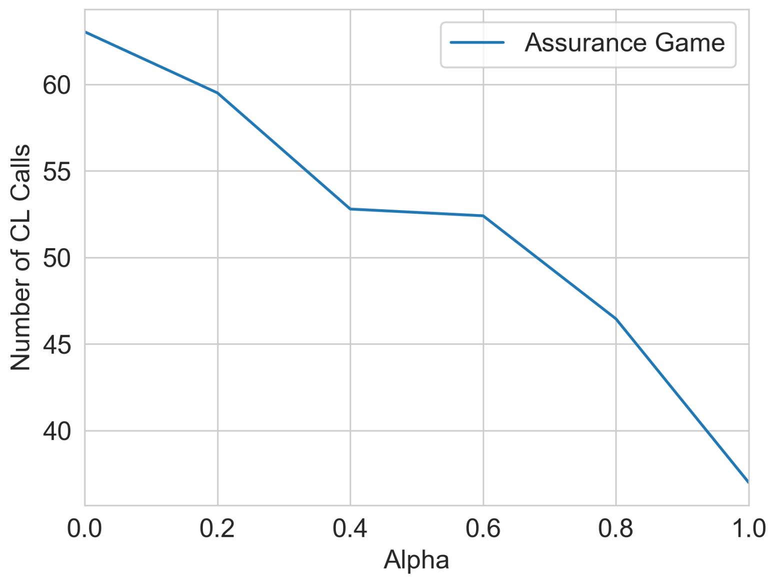

Figure 1: MANSA CL calls in Modified Assurance game in Table 1.

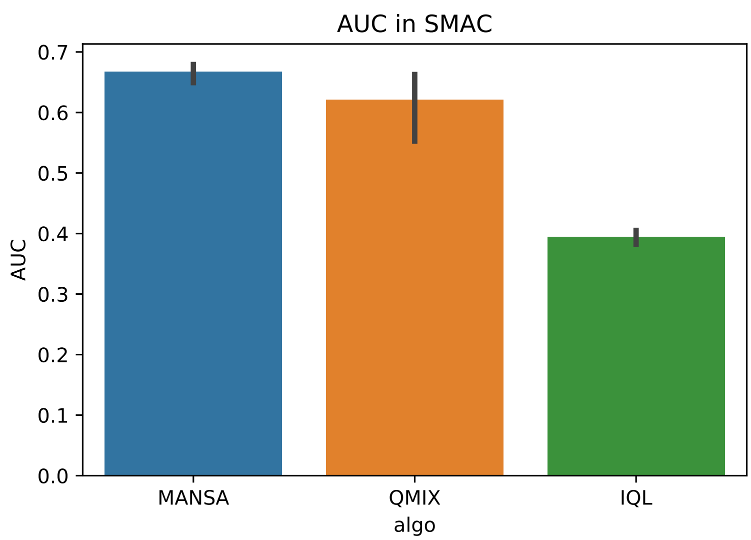

Figure 2: Area under the curve (normalised) results in all tested StarCraft Multi-agent Challenge maps.

	AUC	std
MANSA	0.692759	0.007762
QMIX	0.363176	0.009317
IQL	0.617038	0.006738

Table 3: Mean and standard deviation of each algorithm at each step in each map from LBF.

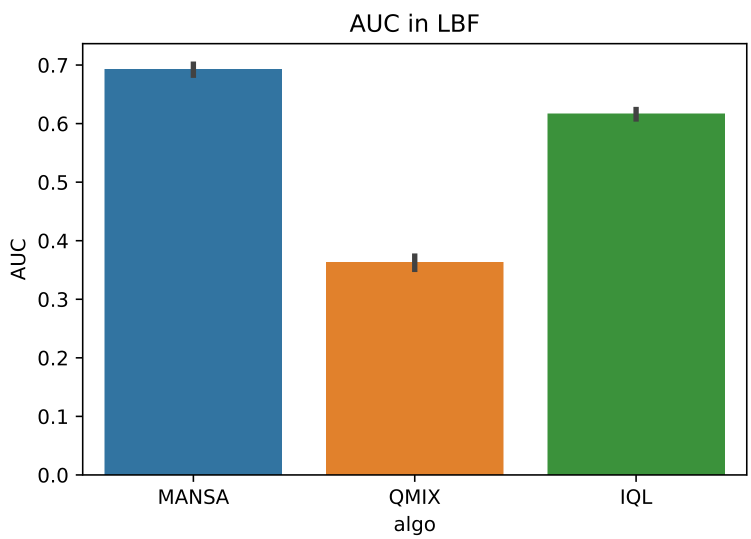

Figure 3: Area under the curve (normalised) results in all tested Level-Based Foraging maps.

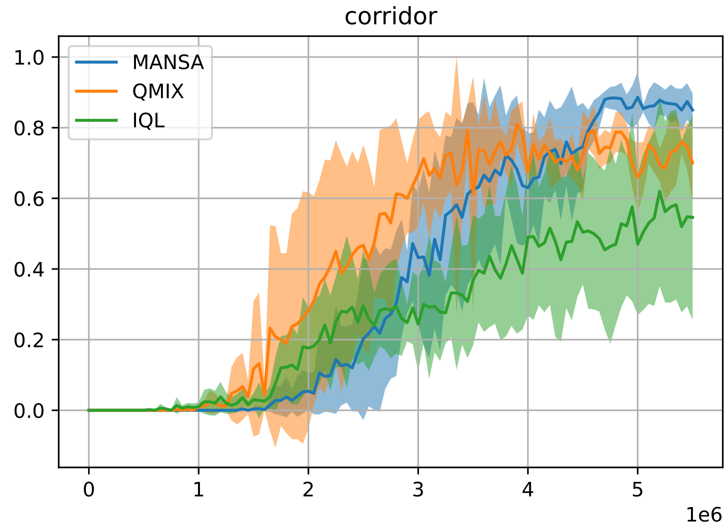

Figure 4: MANSA and baselines with 5 seeds in the StarCraft Multi-Agent Challenge (SMAC) map *Corridor*.

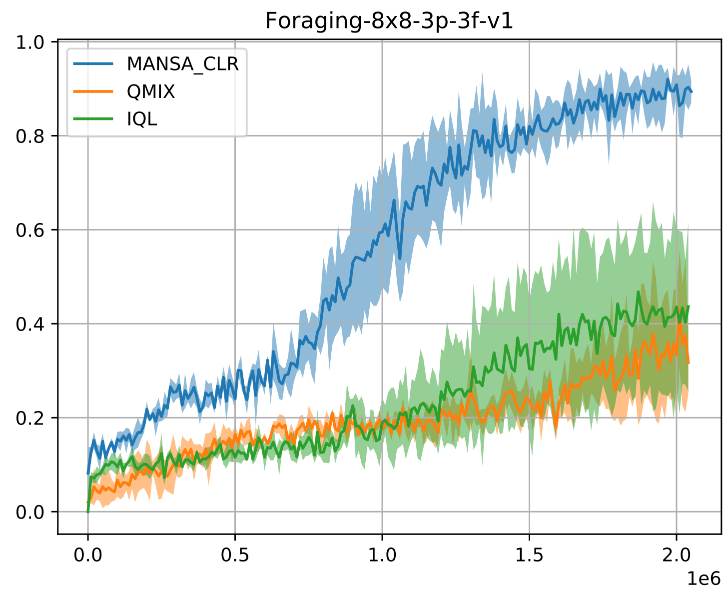

Figure 5: End-of-training win-rates of MANSA with implementation with CL update restriction (MANSA_CLR) in Level-Based Foraging (LBF).

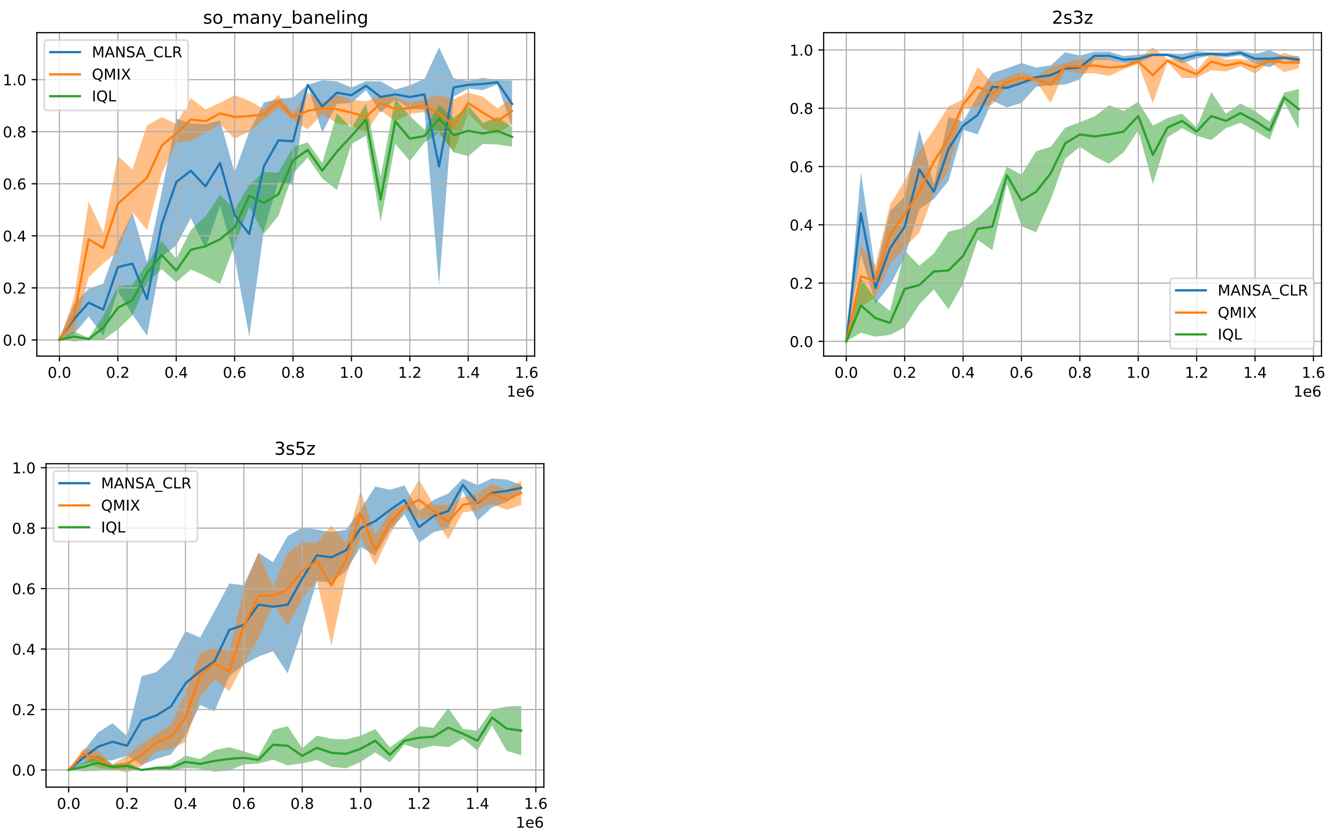

Figure 6: End-of-training win-rates of MANSA with implementation with CL update restriction (MANSA_CLR) in StarCraft Multi-Agent Challenge (SMAC).

	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

Table 4: End-of-training win-rates of MANSA-B with implementation with CL update restriction and various CL call budget constraints against baselines.

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 Δ^{i}, centralised learning algorithm Δ^{c}, learning algorithm for Global Δ^{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 Δ^{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 Δ^{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 Δ^{g} **(Discard** $\boldsymbol{a}_t, \boldsymbol{\tau}_t, \boldsymbol{\tau}_{t+1}$)

Figure 7: Pseudocode for MANSA. This includes a centralised learning update restriction.