A PROOF OF THEORETICAL RESULTS

A.1 Proof of Theorem 3.4

THEOREM 3.4. *The value of the optimal identical Blue team policy ϕ^*_{finite} obtained from the finite population game is within ϵ of the value of the optimal identical Blue team policy ϕ^* obtained from the equivalent zero-sum coordinator game. Formally, for all joint states $\mathbf{x}^{N_1}$ and $\mathbf{y}^{N_2}$,*

$$\min_{\psi^{N_2}} J^{N,\phi^*_{\text{finite}},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) - \min_{\psi^{N_2}} J^{N,\phi^*,\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \leq \epsilon, \quad (8)$$

where $\epsilon = O(1/\sqrt{\underline{N}})$ and $\underline{N} = \min\{N_1, N_2\}$.

We have the following definition of the lower game value for the finite-population ZS-MFTG:

$$\underline{J}^{N*}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) = \max_{\phi^{N_1}\in\Phi^{N_1}} \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^{N_1},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}). \quad (A.1)$$

Note that the maximization for the Blue team is being performed over the set of all team policies Φ^{N_1}, including identical as well non-identical team policies. If we restrict ourselves to the set of identical team policies $\Phi \subseteq \Phi^{N_1}$ it follows immediately that

$$\underline{J}^{N*}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \geq \max_{\phi^{N_1}\in\Phi} \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^{N_1},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}). \quad (A.2)$$

Suppose that ϕ^*_{finite} is the optimal identical Blue team policy obtained from the finite population game. It follows from (A.1) that

$$\underline{J}^{N*}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \geq \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*_{\text{finite}},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}). \quad (A.3)$$

Furthermore, let $\phi^* \in \Phi$ be the optimal identical Blue team policy obtained from the equivalent zero-sum (infinite-population) coordinator game and

$$\min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*_{\text{finite}},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \geq \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*,\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}). \quad (A.4)$$

Using Theorem 3.3, and using (A.4), yields

$$\begin{aligned}\underline{J}^{N*}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) &\geq \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*_{\text{finite}},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \\ &\geq \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*,\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \\ &\geq \underline{J}^{N*}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) - O\Big(\frac{1}{\sqrt{\underline{N}}}\Big),\end{aligned}$$

where $\underline{N} = \min\{N_1, N_2\}$. Upon rearranging terms, we finally have,

$$\min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*_{\text{finite}},\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) - \min_{\psi^{N_2}\in\Psi^{N_2}} J^{N,\phi^*,\psi^{N_2}}(\mathbf{x}^{N_1},\mathbf{y}^{N_2}) \leq \epsilon,$$

where $\epsilon = O(1/\sqrt{\underline{N}})$.

A.2 Proof of Proposition 5.1

PROPOSITION 5.1. *With initial conditions $\mu_{t=0} = [1,0,0]^\intercal$ and $\nu_{t=0} = [0,1,0]^\intercal$, all mean-field optimal trajectories satisfy $\mu^*_t = \nu^*_t = [\frac{1}{3},\frac{1}{3},\frac{1}{3}]^\intercal$ for all $t \geq 2$, and $\mu^*_1 = [0, 1-\eta, \eta]^\intercal$ where $\eta \in [\frac{1}{3},\frac{2}{3}]$ and $\nu^*_1 = [0,\frac{2}{3},\frac{1}{3}]^\intercal$. Furthermore, the unique game value is given by $-\frac{1}{3}$.*

For the constrained RPS game under the stated initial condition, we cannot obtain the target distribution $\begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}^\intercal$ after a single time step but this may be possible for $t \geq 2$. To this end, consider the following candidate trajectory respecting the transition dynamics:

$$\begin{aligned}\mu_0^\intercal &= \begin{bmatrix}1 & 0 & 0\end{bmatrix}, \\ \mu_1^\intercal &= \begin{bmatrix}1-x_1 & x_1 & 0\end{bmatrix}, \\ \mu_2^\intercal &= \begin{bmatrix}1-x_1-x_2 & x_1+x_2-x_3 & x_3\end{bmatrix}, \\ \mu_t^\intercal &= \begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}, \quad \forall t > 2. \\ \nu_0^\intercal &= \begin{bmatrix}0 & 1 & 0\end{bmatrix}, \\ \nu_1^\intercal &= \begin{bmatrix}0 & 1-y_1 & y_1\end{bmatrix}, \\ \nu_2^\intercal &= \begin{bmatrix}y_3 & 1-y_1-y_2 & y_1+y_2-y_3\end{bmatrix}, \\ \nu_t^\intercal &= \begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}, \quad \forall t > 2.\end{aligned}$$

In order to respect the simplex structure for μ_t and ν_t, we have the following constraints at all times:

$$0 \leq x_1, x_2, x_3 \leq 1, \quad x_2 \leq 1 - x_1, \quad x_3 \leq x_1.$$

Similarly,

$$0 \leq y_1, y_2, y_3 \leq 1, \quad y_2 \leq 1 - y_1, \quad y_3 \leq y_1.$$

For the distribution at $t = 2$ to be $\begin{bmatrix}\frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{bmatrix}^\intercal$ for both teams, we get the additional constraints

$$\begin{aligned} x_3 = y_3 &= \frac{1}{3}, \\ x_1 + x_2 - x_3 = y_1 + y_2 - y_3 &= \frac{1}{3}, \\ \implies x_1 + x_2 = y_1 + y_2 &= \frac{2}{3}, \end{aligned}$$

which implies that

$$x_1, y_1 \leq \frac{2}{3},$$

since $x_2, y_2 \geq 0$. The constraints now take the form

$$\frac{1}{3} \leq x_1 \leq \frac{2}{3}, \quad (A.5)$$

and similarly,

$$\frac{1}{3} \leq y_1 \leq \frac{2}{3}. \quad (A.6)$$

The objective function for cRPS is given by

$$J^{N,\phi,\psi}(\mu_0,\nu_0) = \mathbb{E}_{\phi,\psi}\Big[\sum_{t=1}^{T} \mu_t^\intercal A \nu_t \Big| \mu_0, \nu_0\Big], \quad (A.7)$$

which leads to the optimization problem

$$\max_{\phi^t}\min_{\psi^t} \quad J^{N_1,N_2,\phi^t,\psi^t}(\mu_0,\nu_0) = \mu_1^\intercal A \nu_1 \quad . \quad (A.8)$$

Substituting $\mu_1^\intercal = [1-x_1, x_1, 0]$ and $\nu_1^\intercal = [0, 1-y_1, y_1]$ results in the following expression for the maximizing Blue team:

$$\begin{aligned}\max_{\phi^t} \quad J^{N_1,N_2,\phi^t,\psi^t}(\mu_0,\nu_0) &= x_1 + 2y_1 - 3x_1y_1 - 1 \\ &= x_1(1-3y_1) + (2y_1 - 1).\end{aligned}$$

Since this equation is linear in x_1, the solution to the maximization problem subject to the constraint (A.5) is

$$x_1 = \frac{1}{3}, \quad y_1 > \frac{1}{3}, \tag{A.9}$$

$$x_1 = \frac{2}{3}, \quad y_1 < \frac{1}{3}, \tag{A.10}$$

$$x_1 \in [\frac{1}{3}, \frac{2}{3}], \quad y_1 = \frac{1}{3}. \tag{A.11}$$

Following the same approach for the minimizing Red team, we get the following objective,

$$\min_{\psi^t} \quad J^{N_1,N_2,\phi^t,\psi^t}(\mu_0, \nu_0) = x_1 + 2y_1 - 3x_1y_1 - 1$$
$$= y_1(2 - 3x_1) + (x_1 - 1),$$

subject to the constraint (A.6), with the solution being:

$$y_1 = \frac{1}{3}, \quad x_1 < \frac{2}{3}, \tag{A.12}$$

$$y_1 = \frac{2}{3}, \quad x_1 > \frac{2}{3}, \tag{A.13}$$

$$y_1 \in [\frac{1}{3}, \frac{2}{3}], \quad x_1 = \frac{2}{3}. \tag{A.14}$$

Constraint (A.5) ensures that (A.13) cannot hold, while constraint (A.6) similarly prevents (A.10) from holding.

Consider now the case when $y_1 > \frac{1}{3}$. From (A.9) it follows that $x_1 = \frac{1}{3}$. Conversely, if the Blue team commits to a distribution with $x_1 = \frac{1}{3}$, the Red team's best response given by (A.12) gives $y_1 = \frac{1}{3}$, resulting in an incentive for the Red team to deviate from $y_1 > \frac{1}{3}$. Thus, (A.9) does not constitute an optimal solution. Following a similar argument, it can be shown that (A.14) is not an optimal solution either, as illustrated below.

Assume that $x_1 = \frac{2}{3}$. From (A.14), $y_1 \in [\frac{1}{3}, \frac{2}{3}]$. Now, if the Red team announces that it will deploy the distribution $y_1 \in [\frac{1}{3}, \frac{2}{3}]$, the Blue team's response for x_1 follows from (A.9) and (A.11). We have already established that (A.9) is not an optimal solution. This implies that $x_1 \in [\frac{1}{3}, \frac{2}{3}]$ can be a possible response to the Red team. However it violates (A.14), where $x_1 = \frac{2}{3}$ follows from strict equality. Thus, (A.14) does not constitute an optimal solution as the Blue team has an incentive to deviate.

Now, suppose the Blue team announces a distribution where $x_1 \in [\frac{1}{3}, \frac{2}{3}]$. In this case, the Red team's optimal response, derived from (A.12) and (A.11), is $y_1 = \frac{1}{3}$. Conversely, if the Red team announces that its distribution will be $y_1 = \frac{1}{3}$, the Blue team will still follow $x_1 \in [\frac{1}{3}, \frac{2}{3}]$. Since neither team has an incentive to deviate from these distributions, they form an optimal trajectory. Thus, the solution to the bilinear optimization problem for two-time step convergence takes the form:

$$\mu_1^* = \begin{bmatrix} 1 - x_1 \\ x_1 \\ 0 \end{bmatrix} \quad \text{and} \quad \nu_1^* = \begin{bmatrix} 0 \\ \frac{2}{3} \\ \frac{1}{3} \end{bmatrix}, \tag{A.15}$$

such that $x_1 \in [\frac{1}{3}, \frac{2}{3}]$, leading to a game value of $-\frac{1}{3}$. This establishes the distribution at $t = 1$ and confirms the existence of a two-time step optimal trajectory, thereby proving the first part of the proposition.

Now note the following:

(1) The original objective function (A.7) can be expressed in a bilinear form (similar to the expressions for μ_0, μ_1, μ_2 using x_1, x_2, x_3). This makes it concave in the first argument and convex in the second argument.
(2) The mean-fields μ and ν lie on a simplex and are hence, compact and convex.

Thus, by the generalized version of von Neumann's minimax theorem [21], we conclude that the game value is unique, proving the second part of the proposition [1].

B RPS AND cRPS SETUP

B.1 State Space

We have three states in this representation of the game: rock, paper and scissors. We denote this state space as $\mathcal{S} = \{\mathsf{R}, \mathsf{P}, \mathsf{S}\}$. The empirical distribution of the Blue team is denoted by $\mu \in \mathcal{P}(\mathcal{S})$ and that of the Red team is denoted by $\nu \in \mathcal{P}(\mathcal{S})$. Since we have three states for each team, both EDs lie in a three-dimensional simplex denoted by $\mathcal{P}(\mathcal{S})$.

B.2 Action Space

B.2.1 RPS. At each state, we define three actions denoted by $\mathcal{A} = \{\mathtt{CW}, \mathtt{CCW}, \mathtt{Stay}\}$. These actions represent the ability of the agents to move from one state to the other in the following fashion:

(1) CW denotes a clockwise cyclic action from one state to the other, i.e., from $\mathsf{R} \to \mathsf{P}$, $\mathsf{P} \to \mathsf{S}$, $\mathsf{S} \to \mathsf{R}$.
(2) CCW denotes a counterclockwise cyclic movement, i.e., from $\mathsf{R} \to \mathsf{S}$, $\mathsf{S} \to \mathsf{P}$, $\mathsf{P} \to \mathsf{R}$.
(3) Stay denotes the idle action (remain in the same state as before).

B.2.2 cRPS. At each state we have two actions denoted by $\mathcal{A} = \{\mathtt{CW}, \mathtt{Stay}\}$. These actions represent the ability of the agents to move from one state to the other in the following fashion:

(1) CW denotes a clockwise cyclic action from one state to the other, i.e., from $\mathsf{R} \to \mathsf{P}$, $\mathsf{P} \to \mathsf{S}$, $\mathsf{S} \to \mathsf{R}$.
(2) Stay denotes the idle action (remain in the same state as before).

Thus, we cannot directly jump from R to S within a single step, but must go via P. Mathematically, S does not lie in the reachable set of R. The reachable set $\mathcal{R}(s)$ for each state s at a given time step under this modified action space is as follows

$$\mathcal{R}(\mathsf{R}) = \{\mathsf{R}, \mathsf{P}\}, \quad \mathcal{R}(\mathsf{P}) = \{\mathsf{P}, \mathsf{S}\}, \quad \mathcal{R}(\mathsf{S}) = \{\mathsf{S}, \mathsf{R}\}.$$

B.3 Dynamics and Transition Probabilities

For both RPS and cRPS, we consider deterministic transitions $\mathcal{T}(s, a, s')$, which implies that given a state-action pair (s, a), the agent reaches a unique next state s' with certainty (no distribution over the reachable states). Thus, for state R and action Stay, the transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \{0, 1\}$ is given as:

$$\mathcal{T}(\mathsf{R}, \mathtt{Stay}, \mathsf{R}) = 1, \quad \mathcal{T}(\mathsf{R}, \mathtt{Stay}, \mathsf{P}) = 0, \quad \mathcal{T}(\mathsf{R}, \mathtt{Stay}, \mathsf{S}) = 0.$$

[1] Note: The optimal infinite horizon trajectory itself need not be unique (we have shown that x_1 can take a range of values).

This implies that an agent in the state R upon taking the Stay action remains in state R. Similarly,

$$\mathcal{T}(\mathsf{R}, \mathsf{CW}, \mathsf{R}) = 0, \quad \mathcal{T}(\mathsf{R}, \mathsf{CW}, \mathsf{P}) = 1, \quad \mathcal{T}(\mathsf{R}, \mathsf{CW}, \mathsf{S}) = 0.$$

represent the method to transition from R to P .

B.4 Reward Structure

In a two player RPS game, the reward matrix for Player 1 is defined as:

$$A = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}.$$

We extend this two-player framework to the multi-agent team game formulation. Define the pairwise reward for an agent at state $x \in \mathcal{S}$ within the Blue team and at state $y \in \mathcal{S}$ from the Red team as

$$r(x, y) \triangleq A_{xy},$$

where A_{xy} represents the element from the reward matrix A corresponding to the states x (row player) and y (column player). In lieu of the zero-sum structure, the reward for the agent at y with respect to x becomes $-r(x, y)$. Thus, for each player $x_i \in \mathcal{S}$ and $i = 1, 2, \ldots, N_1$ in the Blue team and $y_j \in \mathcal{S}$ and $j = 1, 2, \ldots, N_2$ in the Red team, the reward for the Blue team can be defined as

$$R_{\text{Blue}}(\mathbf{x}, \mathbf{y}) = \frac{1}{N_1} \sum_{i=1}^{N_1} \underbrace{\left[\frac{1}{N_2} \sum_{j=1}^{N_2} r(x_i, y_j) \right]}_{\text{reward for agent } i}.$$

Rewriting the term inside the square brackets as

$$\begin{aligned} \frac{1}{N_2} \sum_{j=1}^{N_2} r(x_i, y_j) &= \frac{1}{N_2} \sum_{y \in \mathcal{S}} \sum_{j=1}^{N_2} r(x_i, y) \mathbf{1}_{y_j = y} \\ &= A_{x_i s_0} \sum_{j=1}^{N_2} \frac{1}{N_2} \mathbf{1}_{y_j = s_0} + A_{x_i s_1} \sum_{j=1}^{N_2} \frac{1}{N_2} \mathbf{1}_{y_j = s_1} \\ &\quad + A_{x_i s_2} \sum_{j=1}^{N_2} \frac{1}{N_2} \mathbf{1}_{y_j = s_2} \\ &= A_{x_i s_0} \nu(s_0) + A_{x_i s_1} \nu(s_1) + A_{x_i s_2} \nu(s_2) \\ &= A(x_i) \nu, \end{aligned} \tag{B.16}$$

where $A(x_i)$ is the row of the reward matrix corresponding to state x_i. Using (B.16), the total Blue reward can be expressed as

$$\begin{aligned} R_{\text{Blue}}(\mathbf{x}, \mathbf{y}) &= \frac{1}{N_1} \sum_{i=1}^{N_1} A(x_i) \nu \\ &= \left(\frac{1}{N_1} \sum_{x \in \mathcal{S}} \sum_{i=1}^{N_1} A(x_i) \mathbf{1}_{x_i = x} \right) \nu \\ &= \Big(A(s_0)\mu(s_0) + A(s_1)\mu(s_1) + A(s_2)\mu(s_2) \Big) \nu \\ &= \mu^{\top} A \nu. \end{aligned}$$

B.5 Implementation Details and Hyperparameters

The state distributions are represented as arrays that are concatenated together to form the global observation. This becomes the input to the critic network which consists of a single hidden layer of 64 neurons and two tanh activation functions. The output is a single value that is equal to the estimated value function. On the other hand, the actor-network consists of a single MLP layer of 64 neurons that is concatenated with the local agent observation. Additionally, the logits are converted to a probability distribution through a softmax layer. The dimension scales with $|\mathcal{A}|$. Both the actor and critic networks are initialized using orthogonal initialization [7].

The single-stage RPS game is trained for 5,000 time steps with the actor and critic learning rates set to 0.0005 and 0.001, respectively, which remain constant throughout training. The networks are updated using the ADAM optimizer [8] every 50 time steps for 10 epochs and a PPO clip value of 0.1. The entropy is decayed from 0.01 to 0.001 geometrically. We use an episode length of 1 after which the rewards are bootstrapped.

Moreover, since we have a single "team" buffer and the input/output dimensions are small, we do not use a mini-batch based update. For cRPS we use an episode length of 10 after which the rewards are bootstrapped. cRPS is trained using 200,000 time steps (=20,000 episodes) and is updated every 100 time steps. The algorithm was trained on a single NVIDIA GeForce RTX 3070 GPU and the training times are given in Tables 1 and 2.

C BATTLEFIELD SETUP

C.1 State and Action Space

We consider a large-scale two-team (Blue and Red) ZS-MFTG on an $n \times n$ grid world. The state of the i^{th} Blue agent is defined as the pair $x_i = (p_i^x, s_i^x)$ where $p_i^x \in \mathcal{S}_{\text{position}}$ denotes the position of the agent in the grid world and $s_i^x \in \mathcal{S}_{\text{status}} = \{0, 1\}$ defines the status of the agent: 0 being inactive and 1 being active. Similarly, we define the state of the Red agent as $y_i = (p_i^y, s_i^y)$. The state spaces for the Blue and Red teams are denoted by $\mathcal{X} = \mathcal{Y} = \mathcal{S}_{\text{position}} \times \mathcal{S}_{\text{status}}$, respectively. The mean-fields of the Blue (μ) and Red (ν) teams are distributions over the joint position and status space, i.e., $\mu, \nu \in \mathcal{P}(\mathcal{S}_{\text{position}} \times \mathcal{S}_{\text{status}})$. The action spaces are given by $\mathcal{U} = \mathcal{V} = \{\mathsf{Up}, \mathsf{Down}, \mathsf{Left}, \mathsf{Right}, \mathsf{Stay}\}$ for both teams, representing discrete movements in the grid world. The learned identical team policy assigns actions based on an agent's local position and status, as well as the observed mean-fields of both teams. In the following subsections, we elaborate on the weakly coupled transition dynamics and reward structure introduced in the game, followed by a detailed discussion of the training procedure and network architecture for MF-MAPPO in this example.

C.2 Interaction Between Agents

The transitions between states for agents belonging to both teams are characterized by their dynamics. These dynamics are probabilistic and depend on interactions among agents and are weakly coupled through their mean-field distributions. The weak coupling dynamics is keeping in line with the assumption in [4].

An agent at a given grid cell can be deactivated by the opponent team with a nonzero probability if the empirical mean-field of the opponent team at the grid cell supersedes that of the agent's own team. Similarly, a deactivated agent can be revived if the empirical mean-field of the agent's team is greater than the opponent's. This is referred to as numerical advantage.

The total transition probability from a state (p, s) to (p', s') by taking an action a is given by the product of transitioning from $(p'|(p, s), a, \mu, \nu)$ and $(s'|(p, s), a, \mu, \nu)$. For simplicity, we consider that the position transition does not depend on the mean-field and the status transition does not depend on the action taken. For agent i belonging to the Blue team, the expression is formulated as

$$\begin{aligned}\mathbb{P}\big(x_i' \mid x_i, a_i, \mu, \nu\big) &= \mathbb{P}\big((p_i^{x'}, s_i^{x'}) \mid (p_i^x, s_i^x), a_i, \mu, \nu\big) \\ &= \mathbb{P}\big(p_i^{x'} \mid (p_i^x, s_i^x), a_i\big)\mathbb{P}\big(s_i^{x'} \mid (p_i^x, s_i^x), \mu, \nu\big).\end{aligned}$$

Here, $\mathbb{P}\big(p_i^{x'} \mid (p_i^x, s_i^x), a_i\big)$ is given by

$$\mathbb{P}\big(p_i^{x'} \mid (p_i^x, s_i^x), a_i\big) = \begin{cases} 1, & s_i^x = 1 \text{ and no boundary} \\ 0, & \text{otherwise.} \end{cases}$$

Calculating $\mathbb{P}\big(s_i^{x'} \mid (p_i^x, s_i^x), \mu, \nu\big)$ yields

$$\begin{aligned}\mathbb{P}\big(1 \mid (p_i^x, 1), \mu, \nu\big) &= 1 - \alpha_x\big(\nu(p_i^x) - \mu(p_i^x)\big), \\ \mathbb{P}\big(0 \mid (p_i^x, 1), \mu, \nu\big) &= \alpha_x\big(\nu(p_i^x) - \mu(p_i^x)\big),\end{aligned}$$

and

$$\begin{aligned}\mathbb{P}\big(1 \mid (p_i^x, 0), \mu, \nu\big) &= \beta_x\big(\mu(p_i^x) - \nu(p_i^x)\big), \\ \mathbb{P}\big(0 \mid (p_i^x, 0), \mu, \nu\big) &= 1 - \beta_x\big(\mu(p_i^x) - \nu(p_i^x)\big),\end{aligned}$$

where α_x and β_x are parameters that control the amount of activation and deactivation and $\big(\nu(p_i^x) - \mu(p_i^x)\big)$ and $\big(\mu(p_i^x) - \nu(p_i^x)\big)$ are the numerical advantages of the Red (over Blue) and Blue (over red) teams at position p^x respectively. The values are clipped between 0 and 1. The Red team, being the defending team, is given a slight advantage in terms of higher deactivation power. This enables the possibility of capturing Blue team agents. However, to avoid degeneracy, the Red team agents are not allowed to penetrate the target , that is,

$$\mathbb{P}\big((p_i^{y'} = \text{Target}) \mid (p_i^y, s_i^y), v_i\big) = 0 \quad p_i^y \neq \text{Target}.$$

For our experiments, we assume $\alpha_x = 15$, $\alpha_y = 5$, and $\beta_x = \beta_y = 0$.

C.3 Reward Structure

The team rewards only depend on the mean-fields of the two teams. For the battlefield scenario, the Blue team agents receive a positive reward corresponding to the fraction of agents that reach the target alive. This is a one-time reward that depends on the change in the fraction of the population of the agents at the target, i.e., if $\mu_t|_{\text{target}} = \mu_{t+1}|_{\text{target}}$, then the team does not receive any positive reward. Each agent in the team receives an identical "team reward." The reward function is mathematically formulated as

$$R_{\text{Blue},t+1}(\mu, \nu) = \kappa\, \Delta\mu_{t+1}|_{\text{target}},$$

where,

$$\Delta\mu_{t+1}|_{\text{target}} = \mu_{t+1}(p^x = \text{Target}, s^x = 1) - \mu_t(p^x = \text{Target}, s^x = 1).$$

We have chosen $\kappa = 100$ in our simulations (heavier emphasis on reaching the target). The Red team's reward is the negative of the Blue team since we have a zero-sum game. Each team aims to maximize its own expected reward.

C.4 Implementation and Hyperparameters

The state distribution for a grid world of size $n \times n$ is represented as a three-dimensional array of size $(2, n, n)$ for each team. The first layer depicts the mean-field of the agents over an $n \times n$ grid that are alive and active, while the second layer gives information about the team's deactivated population. Each team's distribution is then concatenated together to form the global observation. This becomes the common information that is the input to the critic network which in our case is of size $(4, n, n)$ as we have two teams. Both neural networks consist of two main parts: a convolutional block and a fully connected block.

For the critic, the first CNN layer is the input layer that takes the 4 channels and outputs 32 channels, with a kernel size of 3x3, stride of 1, and padding of 1. Followed by ReLU activation, we have a hidden layer that takes 32 channels and outputs 64 channels, with the same kernel size, stride, and padding. Lastly, after another ReLU activation, we have the output layer that takes 64 channels and outputs 64 channels, again with the same kernel size, stride, and padding. After another ReLU layer, the output of the CNN is passed through an MLP. Namely, a fully connected (dense) layer takes the flattened output of the convolutional block and reduces it to 128 units. Between the input and the output layers, we have a single tanh activation function.

On the other hand, the input to the actor-network is split into two CNN blocks: one to process the common information and one to process the local information. The local information channel, is an array of size $(1, n, n)$ that locates the position of the agent with value +1 if it is active and -1 if it has been deactivated. This local information is passed through a single CNN layer that outputs 16 channels with a kernel size of 3x3, stride of 1, and padding of 1 while the common information is passed through two such layers with the output of 32 channels. Both outputs are then followed by a ReLU activation function and the latent representation of the common information combined with the local agent observation is then passed through an MLP architecture.

A fully connected (dense) layer takes the flattened output of the convolutional block and reduces it to 512 units. We have a single hidden layer that reduces the dimension further to 128 and then the output logits. The layers are separated by the tanh activation functions. Finally, the logits are converted to a probability distribution through a softmax layer. Both the actor and critic networks are initialized using orthogonal initialization [7]. The architectures of the shared-team actor and minimally-informed critic networks for this example are shown in Figures C.1 and C.2 respectively.

All maps are trained using a single NVIDIA GeForce RTX 3070 GPU. The actor and critic learning rates are set to 0.0005 and 0.001 and both decay geometrically by a factor of 0.999. The networks are updated using the ADAM optimizer [8] with two mini-batches for 10 epochs and a PPO clip value of 0.1. The entropy coefficient is initialized to 0.01 and decays with a factor of 0.995.

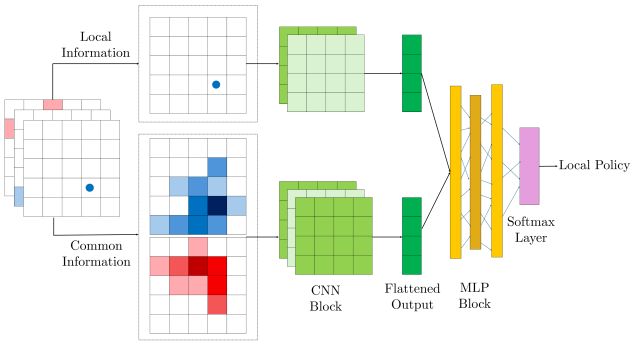

Figure C.1: MF-MAPPO: Shared-team actor for battlefield

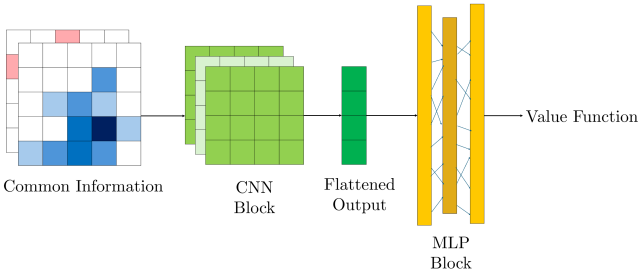

Figure C.2: MF-MAPPO: Minimally-informed critic for battlefield

Maps 1 and 2 which are 4×4 grid worlds are trained for 5×10^6 and 4.5×10^6 time steps, respectively, and in both cases, the episode length is 20 time steps and the update frequency is every 500 time steps. The total training period is about one day. On the other hand, Map 3 being 8×8 in dimension, has an episode length of 64, is trained for 9×10^6 time steps and its network is updated every 1,000 time steps. The total training period is approximately three days.

D ADDITIONAL RESULTS

In this section, we present additional simulation results from the zero-sum battlefield game.

D.1 Validation Cases for MF-MAPPO

The following subsection qualitatively discusses the battlefield game for different map layouts. For these results, both teams are deploying policies trained using MF-MAPPO.

Map A. The first map is a simple 4×4 grid world with a single target that we use to validate our algorithm. The target is partially blocked by an obstacle, see Figure D.3. For the initial condition in Figure D.3, the Blue team is initially split into two equal groups. The Blue team decides to merge the two sub-groups of agents into a single group. With this formation, the Red team has zero numerical advantage over the Blue team when they encounter in (g), resulting in all Blue agents safely arriving at the target. In comparison, if the two Blue subgroups do not merge but move toward the target one at a time, it will lead to 50% of the Blue team population being deactivated (first subgroup), followed by the remaining 50% (second subgroup). This demonstrates how the observation of mean-field distributions guides rational decision-making.

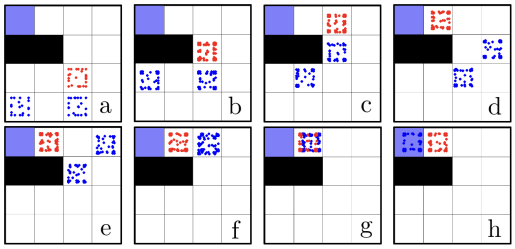

Figure D.3: Red is concentrated; Blue is evenly split.

Map B. This map is identical to the one presented in Section 5.3. In the first scenario (Figure D.4), 70% of the Blue agents start at cell [2, 2], and 30% at cell [1, 1], while the Red team is evenly split between cells [0, 1] and [3, 3]. Half of the Red team at [0, 1] successfully blocks the 30% Blue agent group from entering the left corridor due to its numerical advantage, which forces the Blue agents to opt for the right corridor. At the same time, the larger Blue group with 70% of the population utilized their numerical advantage over the half Red team at the top right and deactivated all the Red agents as shown in (c) and reached the target at time step (d). This allowed the smaller 30% group to follow through the same corridor without losing agents.

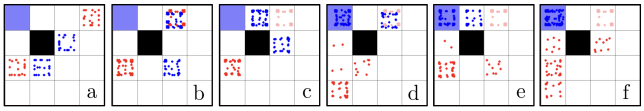

Figure D.4: Red is evenly split; 30% Blue are at $[1, 1]$ and 70% are at $[2, 2]$.

In the second scenario (Figure D.5), with the Red team evenly distributed at the corners, 30% of the Blue agents start at cell [2, 2] and 70% at cell [3, 1]. The Red team's numerical advantage at [3, 3] forces the Blue agents to move around and regroup (Figures D.5(b)-D.5(f)). Once united, the Blue team's numerical advantage forces the Red subgroup at [0,1] to disperse to avoid deactivation, allowing Blue to reach the target.

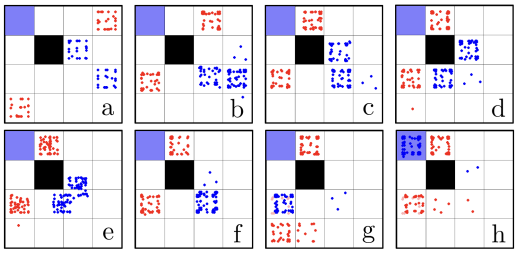

Figure D.5: Red is evenly split; 70% Blue are at $[3, 1]$ and 30% are at $[2, 2]$.

Map C. Subsection 5.3 presented scenarios featuring structured initial configurations for both teams over an 8×8 grid. It is important to emphasize, however, that the algorithm is trained on a diverse set of initial conditions for a given map, ranging from agents concentrated within a few selected cells to agents distributed randomly across the grid world. The following examples demonstrate that the teams are able to accomplish their objectives even in scenarios where agents are dispersed across the environment rather than clustered into just 1-2 subgroups.

In Figure D.6, the Blue team is initialized randomly, and local subgroups of agents emerge and coordinate to reach the target. This behavior is particularly pronounced near the upper target, where a greater numerical advantage facilitates successful coalition formation (Figures D.6(c)–(e)).

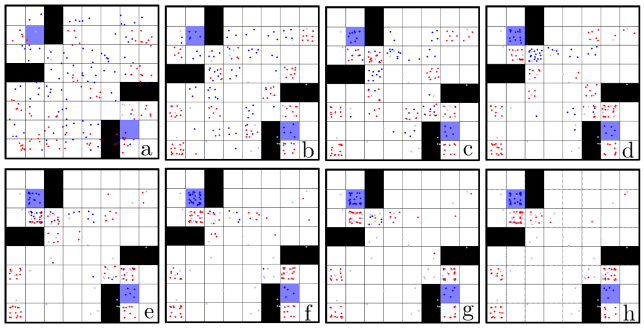

Figure D.6: Blue team is randomly spread around the map.

Turning to the randomly distributed Red team agents in Figure D.7, it is observed that they concentrate near the two target entrances and successfully neutralize most Blue team subgroups.

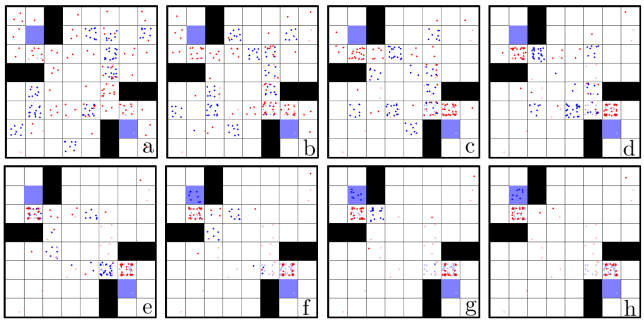

Figure D.7: Red team is randomly spread around the map.

D.2 Comparison with Baseline

Initial Condition 1. We focus on the same initial conditions as in Figure 9, but now pit the Blue team against the MF-MAPPO defenders instead of the DDPG-MFTG defenders (Figures 9(c) and D.8). The results align with those in Figure 8, where MF-MAPPO Blue agents effectively leverage their numerical advantage, enabling a larger number of agents to reach the target.

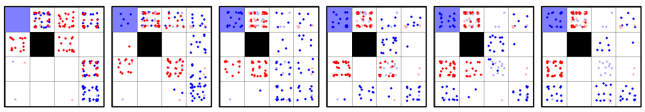

Figure D.8: MF-MAPPO Red vs. DDPG-MFTG Blue.

Initial Condition 2. In Figures D.9 and 10(c), the defenders deactivate the Blue agents under both algorithms. However, similar to the results in Figure 10, MF-MAPPO agents actively learn to block the targets.

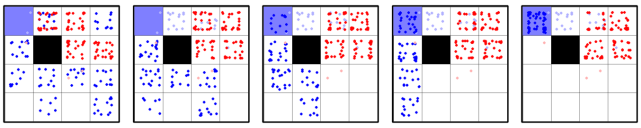

Figure D.9: MF-MAPPO Blue vs. DDPG-MFTG Red.

Initial Condition 3. We present a final initial condition for the 4×4 Battlefield game (Figure D.10, where, using similar arguments as in the previous two cases, we can establish the superiority of MF-MAPPO agents over the baseline DDPG-MFTG, whether MF-MAPPO serves as the attacker or the defender.

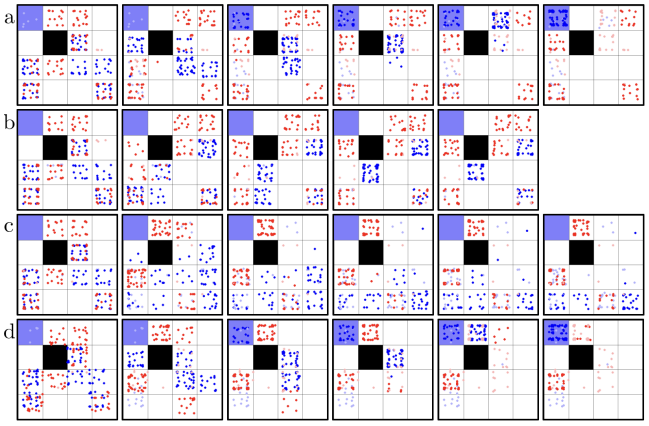

Figure D.10: a. MF-MAPPO Blue vs. DDPG-MFTG Red; b. DDPG-MFTG Blue vs. DDPG-MFTG Red; c. MF-MAPPO Red vs. DDPG-MFTG Blue; d. MF-MAPPO Red vs. MF-MAPPO Blue.