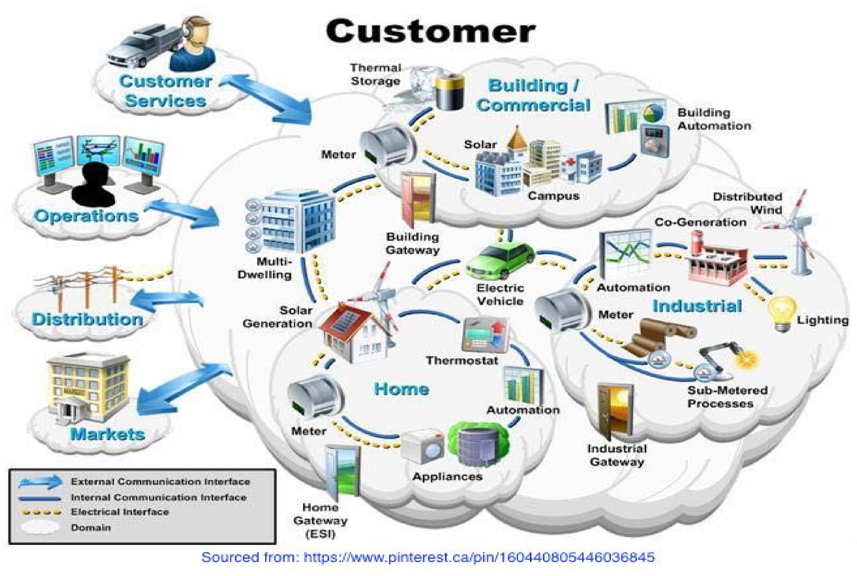
# Reinforcement Learning for Mean-field Teams

Jayakumar Subramanian, Raihan Seraj & Aditya Mahajan

ECE & CIM, McGill University and GERAD

## Mean-field Teams


## Motivation

- Systems with large number of exchangeable agents:
  - Smart grids
  - Cellular networks
  - Computer networks
  - Economic organizations
- Difficulty:
  - Global state not available/too expensive to share.
  - Curse of dimensionality: solution concept scales exponentially or double exponentially with number of agents.
- Objective: Low complexity RL algorithm.

## Model

- **State & action of agent** $i \in N$: $X^i_t \in \mathcal{X}$ & $U^i_t \in \mathcal{U}$.
- All agents are **partially exchangeable** $\implies$ the state evolution of a generic agent depends on the states and actions of other agents **only through the mean-fields** of the states.
- **Mean-field of** $X$: $Z_t = \frac{1}{n}\sum_{i\in N}\delta_{X^i}$.
- The system has **mean-field sharing information-structure**, i.e., the information available to agent $i$ is given by $I^i_t = \{X^i_t, Z_t\}$.
- We assume that all agents use identical (stochastic) control law: $\mu_t \colon \mathcal{X}\times\mathcal{Z} \to \Delta(\mathcal{U})$
- **Per-step reward**: $R_t \sim r(\mathbf{X}_t, \mathbf{U}_t)$.
- **Independent initial states**: $\mathbb{P}(X_0 = x_0) = \prod_{i\in N}\mathbb{P}(X^i_0 = x^i_0) =: \prod_{i\in N} P_0(x^i_0)$.
- **Controlled Markov evolution**: $\mathbb{P}(X_{t+1} \mid X_{0:t}, U_{0:t}) = \mathbb{P}(X_{t+1} \mid X_t, U_t)$.
- **Mean-field effect**:
  $\mathbb{P}(X_{t+1} \mid X_t, U_t) = \prod_{i\in N}\mathbb{P}(X^i_{t+1} \mid X^i_t, U^i_t, Z_t) =: \prod_{i\in N} P(x^i_{t+1} \mid x^i_t, u^i_t, z_t)$.
- **Resulting dynamics**:
  $\mathbb{P}(X_{t+1} = x_{t+1} \mid X_{0:t} = x_{0:t}, U_{0:t} = u_{0:t}) = \prod_{i\in N} P(x^i_{t+1} \mid x^i_t, u^i_t, z_t)$.
- **Performance**: $J(\mu) = \mathbb{E}^{\mu}\Big[\sum_{t=0}^{\infty}\gamma^t R_t\Big]$.

**Main idea behind solution approach**

**Existence of a planning solution for the model specified $\implies$ basis for developing an RL approach for the model.**

## Planning solution [Arabneydi and Mahajan, 2014]

- Prescription: $h_t(x) = \mu_t(x, z_t), \quad \forall x \in \mathcal{X}$.
- Per-step reward: $\mathbb{E}[r(\mathbf{X}_t, \mathbf{U}_t)|Z_{1:t}, H_{1:t}] = \mathbb{E}[r(\mathbf{X}_t, \mathbf{U}_t)|Z_t, H_t] =: \tilde{r}(Z_t, H_t)$.

**Theorem**

Unique bounded fixed point: $V(z) := \max_{h\in\mathcal{H}}\mathbb{E}[\tilde{r}(z,h) + \gamma V(Z_{t+1})|Z_t = z, H_t = h]$
with $\arg\max$ of the right hand side : $\psi(z)$.
$\implies$ optimal policy: $\mu(x, z) = \psi(z)(x)$.

## Reinforcement learning solution

- Assumption: we have access to a simulator for $P(\cdot \mid x^i_t, u^i_t, z_t)$ and $\hat{r}(x^i_t, u^i_t, z_t)$.
- Using $n$ copies of this simulator, we create a simulator for the mean-field dynamics.
- $(Z_{t+1}, R_t)$ obtained from averaging $(X^i_{t+1}, R^i_t) \implies$ simulator with internal state $Z_t$.

**Use of standard RL algorithms in this $Z_t$ state simulator**

TRPO [Schulman et al., 2015], PPO[Schulman et al., 2017] & NAFDQN [Gu et al., 2016].

## Demand response example

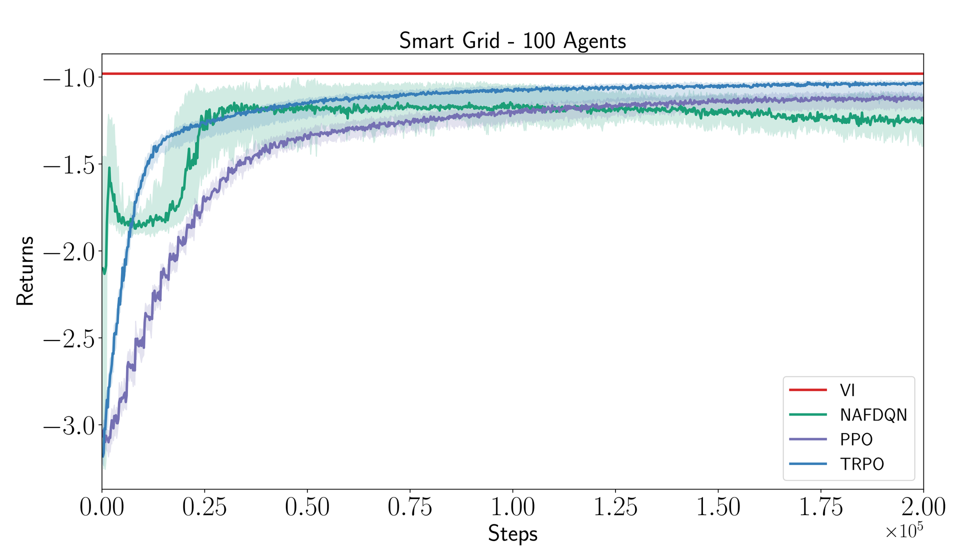
System with $n$ agents, where $\mathcal{X} = \{0, 1\}$, $\mathcal{U} = \{\emptyset, 0, 1\}$. The dynamics are given by:

$$P(\cdot \mid \cdot, \emptyset, z) = M,$$
$$P(\cdot \mid \cdot, 0, z) = (1-\varepsilon_1)\left[\begin{smallmatrix}1 & 0\\ 1 & 0\end{smallmatrix}\right] + \varepsilon_1 M,$$
$$P(\cdot \mid \cdot, 1, z) = (1-\varepsilon_2)\left[\begin{smallmatrix}0 & 1\\ 0 & 1\end{smallmatrix}\right] + \varepsilon_2 M.$$

Per-step reward:

$$R_t = -\Big(\tfrac{1}{n}\sum_{i\in N}\Big(c_0 \mathbb{1}_{\{U^i_t=0\}} + c_1 \mathbb{1}_{\{U^i_t=1\}}\Big) + KL(Z_t \| \zeta)\Big).$$

**Performance**




## Malware spread example

System consists of $n$ agents where $\mathcal{X} = [0, 1]$, $\mathcal{U} = \{0, 1\}$. The dynamics are given by:

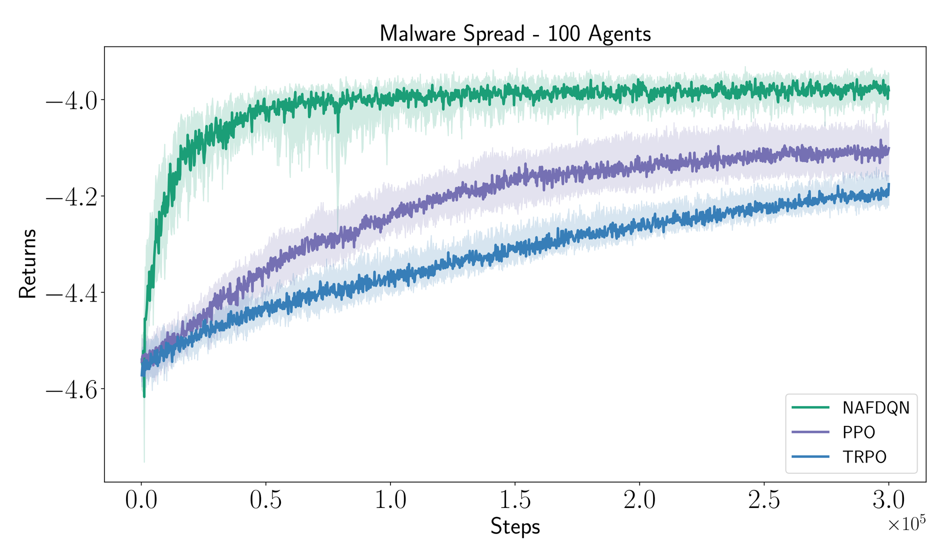
$$X^i_{t+1} = \begin{cases} X^i_t + (1 - X^i_t)\omega_t, & \text{for } U_t = 0,\\ 0 & \text{for } U_t = 1,\end{cases}$$

where $\omega_t \sim \texttt{Uniform}[0, 1]$.

Per-step reward:

$$R_t = -\Big(\tfrac{1}{n}\sum_{i\in N}(k + \langle Z_t\rangle)X^i_t + \lambda U^i_t\Big).$$

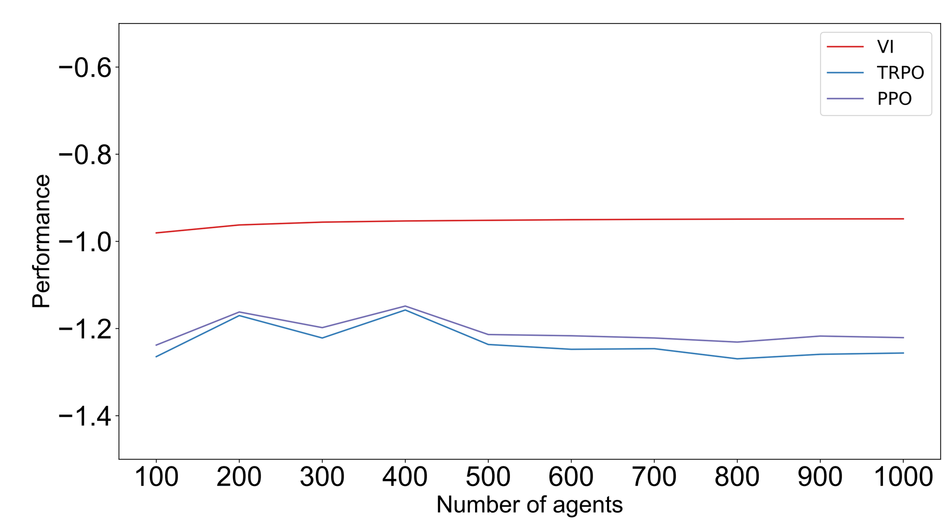
**Performance**




## Mean-field approximation

- Approximate a large population system with an infinite population system, find the optimal policy for the infinite population system and use that policy in the finite population system.
- We use MFT-RL for $m = 100$ agents and use the resultant policy in the systems with $n > 100$ agents.

**MFT-RL for $m = 100$ agents in larger systems**




## Conclusion

- There are many results in the Dec-POMDP/decentralized control literature where a team optimal solution can be obtained using dynamic programming.
- Our central thesis is that for such models one can easily translate the dynamic program to a reinforcement learning algorithm.
- We illustrate this point by using mean-field teams as an example. This allows us to use standard off-the-shelf RL algorithms to obtain solutions for some MARL setups.

LaTeX TikZposter