Multi-Agent Common Knowledge Reinforcement Learning

Christian Schroeder de Witt, Jakob Foerster, Gregory Farquhar, Philip H.S. Torr, Wendelin Boehmer, Shimon Whiteson

schroederdewitt@gmail.com

Motivation

- Deep multi-agent reinforcement learning for cooperative multi-agent tasks such as autonomous driving requires learning of policies that allow agents to coordinate their actions.
- Intractably large joint action spaces, partial observability and communication constraints necessitate the learning of decentralised policies even if learning is centralized.
- The absence of communication channels forces agents to coordinate solely based on *implicit communication* through observations.
- Common knowledge allows agents to coordinate *simultaneously* without the need for explicit communication.

The Challenge: How to construct a deep multi-agent reinforcement learning algorithm that benefits from centralized training in the presence of common knowledge, while allowing for fully decentralized execution?

Common Knowledge

- Common knowledge of a group of agents refers to facts that all members know, and that "*each individual knows that all other individuals know it, each individual knows that all other individuals know that all the individuals know it, and so on*" [Osborne & Rubinstein (1994)]
- Notions for imperfect common knowledge, such as *probabilistic common knowledge* exist [Krasucki 1991].

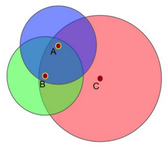


Figure 1: Three agents and their fields of view. A and B's locations are common knowledge to A and B as they are within each other's fields of view. Although C can see A and B, it shares no common knowledge with them.

MACKRL

- Novel actor-critic algorithm with centralised critic and hierarchically factorised joint agent policy.

$$\nabla_\theta J_t = \underbrace{\Big(r(s_t, \mathbf{u}_{\text{env},t}) + \gamma V(s_{t+1}, \mathbf{u}_{\text{env},t}) - V(s_t, \mathbf{u}_{\text{env},t-1})\Big)}_{\text{sample estimate of the advantage function}} \nabla_\theta \log\Big(\underbrace{\pi(\mathbf{u}_{\text{env},t}|\{\tau_t^a\}_{a\in\mathcal{A}},\xi)}_{\text{JOINT\_POLICY}(\mathbf{u}_{\text{env},t}|\mathcal{A},\{\tau_t^a\}_{a\in\mathcal{A}},\xi)}\Big),$$

- Allows end-to-end training while exploiting common knowledge

Algorithm 2 Compute joint policies for a given $\mathbf{u}_{\text{env}}^{\mathcal{G}} \in \mathcal{U}_{\text{env}}^{\mathcal{G}}$ of a group of agents $\mathcal{G}$ in MACKRL

function JOINT_POLICY($\mathbf{u}_{\text{env}}^{\mathcal{G}}|\mathcal{G}, \{\tau_t^a\}_{a\in\mathcal{G}}, \xi$) ▷ random seed in ξ is common knowledge
  $a' \sim \mathcal{G}$; $\mathbf{I}^{\mathcal{G}} := \mathcal{I}^{\mathcal{G}}(\tau_t^{a'}, \xi)$ ▷ common knowledge $\mathbf{I}^{\mathcal{G}}$ is identical for every agent $a' \in \mathcal{G}$
  $p_{\text{env}} := 0$ ▷ initialise probability for choosing environmental joint action $\mathbf{u}_{\text{env}}^{\mathcal{G}}$
  for $\mathbf{u}^{\mathcal{G}} \in \mathcal{U}^{\mathcal{G}}$ **do** ▷ add probability to choose $\mathbf{u}_{\text{env}}^{\mathcal{G}}$ for all outcomes of $\pi^{\mathcal{G}}$
    if $\mathbf{u}^{\mathcal{G}} = \mathbf{u}_{\text{env}}^{\mathcal{G}}$ **then** ▷ if $\mathbf{u}^{\mathcal{G}}$ is the environmental joint action in question
      $p_{\text{env}} := p_{\text{env}} + \pi^{\mathcal{G}}(\mathbf{u}_{\text{env}}^{\mathcal{G}}|\,\mathbf{I}^{\mathcal{G}})$
    if $\mathbf{u}^{\mathcal{G}} \notin \mathcal{U}_{\text{env}}^{\mathcal{G}}$ **then** ▷ if $\mathbf{u}^{\mathcal{G}} = \{\mathcal{G}^1, \ldots, \mathcal{G}^k\}$ is a set of disjoint subgroups
      $p_{\text{env}} := p_{\text{env}} + \pi^{\mathcal{G}}(\mathbf{u}^{\mathcal{G}}|\,\mathbf{I}^{\mathcal{G}}) \prod_{\mathcal{G}'\in\mathbf{u}^{\mathcal{G}}} \text{JOINT\_POLICY}(\mathbf{u}_{\text{env}}^{\mathcal{G}'}|\,\mathcal{G}', \{\tau_t^a\}_{a\in\mathcal{G}'}, \xi)$
  return p_{env} ▷ return probability that controller $\pi^{\mathcal{G}}$ would have chosen $\mathbf{u}_{\text{env}}^{\mathcal{G}}$

Algorithm 1 Decentralised action selection for agent $a \in \mathcal{A}$ in MACKRL

function SELECT_ACTION(a, τ_t^a, ξ) ▷ random seed in ξ is common knowledge
  $\mathcal{G} := \mathcal{A}$ ▷ initialise the group $\mathcal{G}$ of all agents
  $\mathbf{u}_t^{\mathcal{G}} \sim \pi^{\mathcal{G}}(\,\cdot\,|\,\mathcal{I}^{\mathcal{G}}(\tau_t^a, \xi))$ ▷ $\mathbf{u}_t^{\mathcal{G}}$ is either a joint environmental action in $\mathcal{U}_{\text{env}}^{\mathcal{G}}$...
  while $\mathbf{u}_t^{\mathcal{G}} \notin \mathcal{U}_{\text{env}}^{\mathcal{G}}$ **do** ▷ ... or a set of disjoint subgroups $\{\mathcal{G}_1, \ldots, \mathcal{G}_k\}$
    $\mathcal{G} := \{\mathcal{G}'\,|\,a \in \mathcal{G}', \mathcal{G}' \in \mathbf{u}_t^{\mathcal{G}}\}$ ▷ select subgroup containing agent a
    $\mathbf{u}_t^{\mathcal{G}} \sim \pi^{\mathcal{G}}(\,\cdot\,|\,\mathcal{I}^{\mathcal{G}}(\tau_t^a, \xi))$ ▷ draw an action for that subgroup
  return u_t^a ▷ return environmental action $u_t^a \in \mathcal{U}_{\text{env}}^{\mathcal{G}}$ of agent a

Pairwise MACKRL

Pairwise MACKRL restricts the joint policy to a three-level hierarchical controller, where the top level is a pair selector conditioning on common knowledge between all the agents, the second level consists of pair controllers conditioning on pair-wise common knowledge. The second level can choose to delegate to the lowest level, which consists of independent learners.

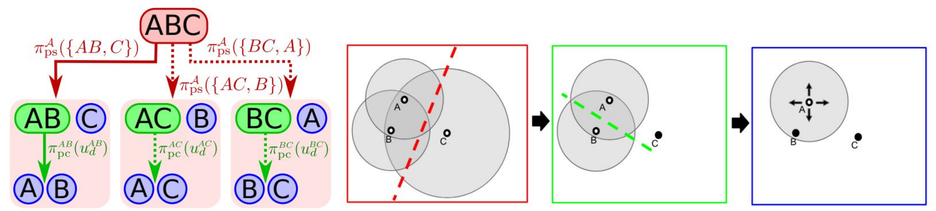


Figure 2: An example for Pairwise MACKRL. [Left]: the full hierarchy for 3 agents (dependencies on common knowledge are omitted for clarity). Only solid arrows are computed during decentralised sampling with Algorithm 1, while all arrows must be computed recursively during centralised training (see Algorithm 2). [Right]: the (maximally) 3 steps of decentralized sampling from the perspective of agent A. (i) Pair selector $\pi_{\text{ps}}^{\mathcal{A}}$ chooses the partition $\{AB, C\}$ based on the common knowledge of all agents $\mathcal{I}^{ABC}(\tau^A, \xi) = \varnothing$. (ii) Based on the common knowledge of pair A and B, $\mathcal{I}^{AB}(\tau^A, \xi)$, the pair controller π_{pc}^{AB} can either choose a joint action $(u_{\text{env}}^A, u_{\text{env}}^B)$, or delegate to individual controllers by selecting u_d^{AB}. (iii) If delegating, the individual controller π^A must select the action u_{env}^A for the single agent A. All steps can be computed based on A's history τ^A.

Level	Policy / Controller	#π
1	$\pi_{\text{ps}}(u^{\text{ps}}\|\mathcal{I}_{s_t}^{\mathcal{A}}, u_{t-1}^{\text{ps}}, h_{t-1}^{\text{ps}})$	1
2	$\pi_{\text{pc}}^{aa'}(u^{aa'}\|\mathcal{I}_{s_t}^{aa'}, u_{t-1}^{aa'}, h_{t-1}^{aa'}, aa')$	3
3	$\pi^a(u^a\|z_t^a, h_{t-1}^a, u_{t-1}^a, a)$	3

$$\begin{aligned}\boldsymbol{\pi}_\theta(u_{\text{env}}^1, u_{\text{env}}^2, u_{\text{env}}^3) = {}& \pi_{\text{ps},\theta}^{\mathcal{A}}(u_{\text{ps}}^{\mathcal{A}}{=}\{\{1\},\{2,3\}\}|\mathcal{I}_s^{1,2,3}) \cdot \pi_\theta^1(u_{\text{env}}^1|\tau^1) \\ & \cdot \Big(\pi_{\text{pc},\theta}^{2,3}(u_{\text{env}}^{2,3}|\mathcal{I}_s^{2,3}) + \pi_{\text{pc},\theta}^{2,3}(u_d^{2,3}|\mathcal{I}_s^{2,3}) \cdot \pi_\theta^2(u_{\text{env}}^2|\tau^2) \cdot \pi_\theta^3(u_{\text{env}}^3|\tau^3)\Big) \\ & + \pi_{\text{ps},\theta}^{\mathcal{A}}(u_{\text{ps}}^{\mathcal{A}}{=}\{\{2\},\{1,3\}\}|\mathcal{I}_s^{1,2,3}) \cdot \pi_\theta^2(u_{\text{env}}^2|\tau^2) \\ & \cdot \Big(\pi_{\text{pc},\theta}^{1,3}(u_{\text{env}}^{1,3}|\mathcal{I}_s^{1,3}) + \pi_{\text{pc},\theta}^{1,3}(u_d^{1,3}|\mathcal{I}_s^{1,3}) \cdot \pi_\theta^1(u_{\text{env}}^1|\tau^1) \cdot \pi_\theta^2(u_{\text{env}}^3|\tau^3)\Big) \\ & + \pi_{\text{ps},\theta}^{\mathcal{A}}(u_{\text{ps}}^{\mathcal{A}}{=}\{\{3\},\{1,2\}\}|\mathcal{I}_s^{1,2,3}) \cdot \pi_\theta^3(u_{\text{env}}^3|\tau^3) \\ & \cdot \Big(\pi_{\text{pc},\theta}^{1,2}(u_{\text{env}}^{1,2}|\mathcal{I}_s^{1,2}) + \pi_{\text{pc},\theta}^{1,2}(u_d^{1,2}|\mathcal{I}_s^{1,2}) \cdot \pi_\theta^1(u_{\text{env}}^1|\tau^1) \cdot \pi_\theta^2(u_{\text{env}}^2|\tau^2)\Big).\end{aligned}$$

Top: Policy hierarchy for Pairwise MACKRL. Right: Explicitly Written-out joint policy for Pairwise MACKRL with three agents.

Results - StarCraft II

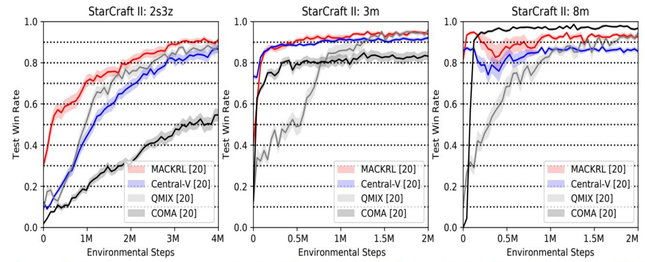
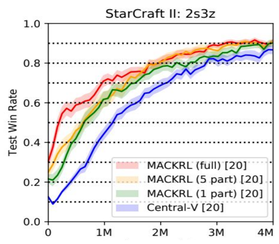


Figure 4: Win rate at test time across StarCraft II scenarios: 2 Stalkers & 3 Zealots [left], 3 Marines [middle] and 8 Marines [right]. Plots show means and their standard errors with [number of runs].

- Challenging unit micromanagement tasks under partial observability constraints with circular field of view.
- MACKRL outperforms its Central-V baseline in sample efficiency and limit performance on all maps.
- MACKRL also significantly outperforms state-of-the-art COMA and QMIX algorithms in terms of sample efficiency.

Scalability, Robustness & Introspection



- MACKRL can exploit *probabilistic* common knowledge arising from noisy sensor using correlated sampling.
- MACKRL can scale to many agents by restricting pair selector to random subsets
- Pair controller introspection verifies use of common knowledge

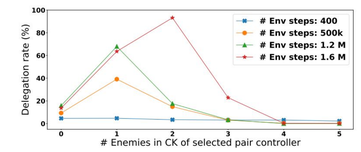
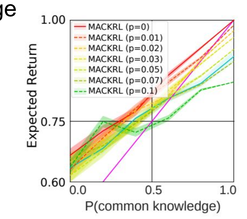


Figure 8: Delegation rate vs. number of enemies (2s3z) in the common knowledge of the pair controller over training.



Results - Matrix game with common knowledge

- We investigate a two-agent single-step matrix game where the reward matrix is chosen between A and B at random. The choice of matrix played is available as common knowledge with probability *p*
- We demonstrate that Pairwise MACKRL outperforms both Joint Action-Learning (JAL) and Independent Learning (IL) under decentralised execution if common knowledge Is available.

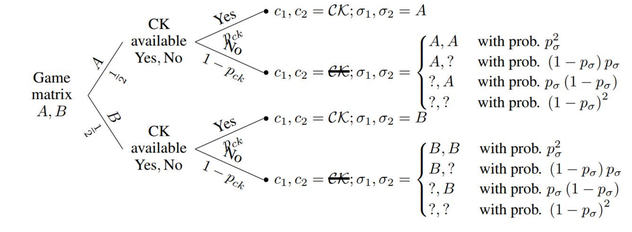
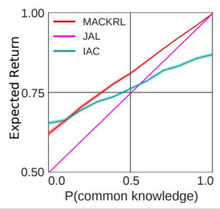


Figure 7: Probability tree for our simple single-step matrix game. The game chooses randomly between matrix A or B, and whether common knowledge is available or not. If common knowledge is available, both agents can condition their actions on the game matrix chosen. Otherwise, both agents independently only have a random chance of observing the game matrix choice. Here, p_{ck} is the probability that common knowledge exists and p_σ is the probability that an agent independently observes the game matrix choice. The observations of each agent 1 and 2 are given by tuples (c_1, σ_1) and (c_2, σ_2), respectively, where $c_1, c_2 \in \{\mathcal{CK}, \cancel{\mathcal{CK}}\}$ and $\sigma_1, \sigma_2 \in \{A, B, ?\}$.



$$\frac{1}{5}\begin{bmatrix}0&0&1&0&5\\0&0&2&0&0\\1&2&4&2&1\\0&0&2&0&0\\5&0&1&0&0\end{bmatrix} \qquad \frac{1}{5}\begin{bmatrix}5&0&0&2&0\\0&1&2&4&2\\0&0&0&2&0\\0&0&0&1&0\\0&0&0&0&5\end{bmatrix}$$

Above: The two game reward matrices A and B. Left: Expected return of MACKRL, JAL and IAC

Discussion and Future Work

- MACKRL uses a novel hierarchical policy structure that allows to exploit common knowledge between agents during fully decentralized execution. This results in outperformance of strong baselines and even state-of-the-art on StarCraft II. We demonstrate MACKRL's ability to scale to many agents and its robustness to sensor noise.
- Future work will address off-policy variants of MACKRL and investigate how limited-bandwidth communication can be exploited in the presence of common knowledge.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement number 637713), the National Institutes of Health (grant agreement number R01GM114311) and EPSRC/MURI grant EP/N019474/1. This work is linked to and partly funded by the project Free the Drones (FreeD) under the Innovation Fund Denmark and Microsoft. It was also supported by the Oxford-Google DeepMind Graduate Scholarship and a generous equipment grant from NVIDIA.