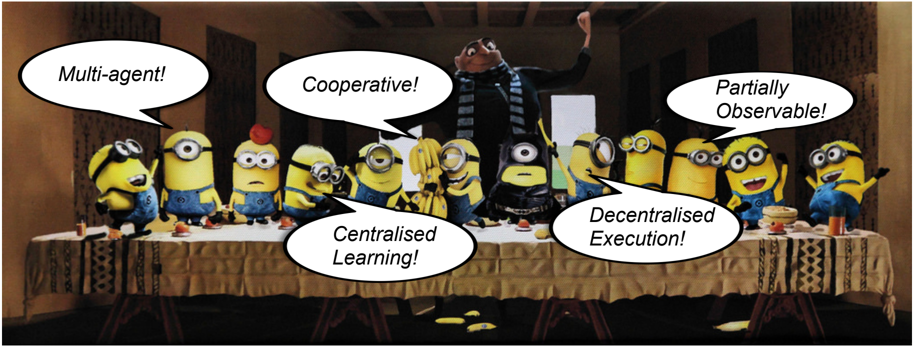
Cooperative Multi-Agent Reinforcement Learning

Shimon Whiteson
Dept. of Computer Science
University of Oxford

joint work with Jakob Foerster, Gregory Farquhar,
Triantafyllos Afouras, Nantas Nardelli, Tabish Rashid,
Mikayel Samvelyan, and Christian Schroeder de Witt

July 13, 2018

Setting



(Figure by Jakob Foerster)

Multi-Agent MDP

- All agents see the global state s
- Individual actions: $u^a \in U$
- State transitions: $P(s'|s, \mathbf{u}) : S \times \mathbf{U} \times S \rightarrow [0, 1]$
- Shared team reward: $r(s, \mathbf{u}) : S \times \mathbf{U} \rightarrow \mathbb{R}$
- Equivalent to an MDP with a factored action space

Dec-POMDP

- Observation function: $O(s, a) : S \times A \rightarrow Z$
- Action-observation history: $\tau^a \in T \equiv (Z \times U)^*$
- Decentralised policies: $\pi^a(u^a|\tau^a) : T \times U \rightarrow [0, 1]$
- Natural decentralisation: communication and sensory constraints
- Artificial decentralisation: coping with joint action space
- Centralised learning of decentralised policies

Single-Agent Policy Gradient Methods

- Optimise π_θ with gradient ascent on expected return:

$$J_\theta = \mathbb{E}_{s \sim \rho^\pi(s), u \sim \pi_\theta(s,\cdot)} \left[r(s,u) \right]$$

- Good when *greedification* is hard, e.g., continuous actions

- Policy gradient theorem [Sutton et al. 2000]:

$$\nabla_\theta J_\theta = \mathbb{E}_{s \sim \rho^\pi(s), u \sim \pi_\theta(s,\cdot)} \left[\nabla_\theta \log \pi_\theta(u|s) Q^\pi(s,u) \right]$$

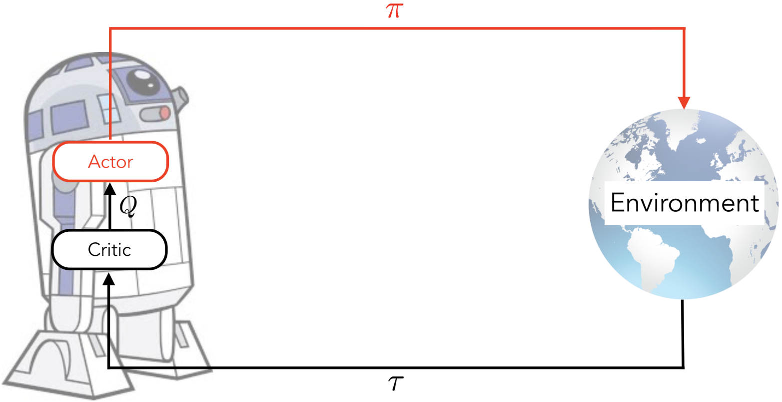
- REINFORCE [Williams 1992]:

$$\nabla_\theta J_\theta \approx g(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t|s_t) R_t$$

Single-Agent Actor-Critic Methods [Sutton et al. 00]

- Reduce variance in $g(\tau)$ by learning a *critic* $Q(s, u)$:

$$g(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t|s_t) Q(s_t, u_t)$$



Single-Agent Baselines

- Further reduce variance with a *baseline* $b(s)$:

$$g(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t|s_t)(Q(s_t, u_t) - b(s_t))$$

- $b(s) = V(s) \implies Q(s, u) - b(s) = A(s, u)$, the *advantage function*:

$$g(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t|s_t)A(s_t, u_t)$$

- *TD-error* $r_t + \gamma V(s_{t+1}) - V(s)$ is an unbiased estimate of $A(s_t, u_t)$:

$$g(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t|s_t)(r_t + \gamma V(s_{t+1}) - V(s_t))$$

Single-Agent Deep Actor-Critic Methods

- Actor and critic are both deep neural networks
    - Convolutional and recurrent layers
    - Actor and critic share layers

- Both trained with stochastic gradient descent
    - Actor trained on policy gradient
    - Critic trained on TD(λ) or Sarsa(λ)

Independent Actor-Critic

- Inspired by *independent Q-learning* [Tan 1993]
  - Each agent learns independently with its own actor and critic
  - Treats other agents as part of the environment

- Speed learning with *parameter sharing*
  - Different inputs, including a, induce different behaviour
  - Still independent: critics condition only on τ^a and u^a

- Limitations:
  - Nonstationary learning
  - Hard to learn to coordinate
  - Multi-agent credit assignment

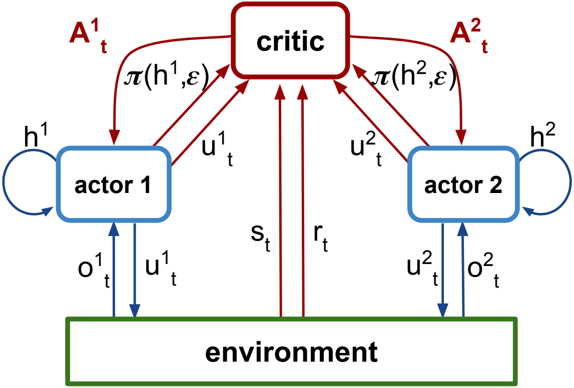
Counterfactual Multi-Agent Policy Gradients

- Centralised critic: stabilise learning to coordinate
- Counterfactual baseline: tackle multi-agent credit assignment
- Efficient critic representation: scale to large NNs

Centralised Critic

Centralisation → Hard greedification → actor-critic

$$g_a(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t^a|\tau_t^a)(r_t + \gamma V(s_{t+1}) - V(s_t))$$



Wonderful Life Utility [Wolpert & Tumer 2000]

Difference Rewards [Tumer & Agogino 2007]

- Per-agent shaped reward:

$$D^a(s, \mathbf{u}) = r(s, \mathbf{u}) - r(s, (\mathbf{u}^{-a}, c^a))$$

where c^a is a *default action*

- Key property:

$$D^a(s, (\mathbf{u}^{-a}, \dot{u}^a)) > D^a(s, \mathbf{u}) \implies r(s, (\mathbf{u}^{-a}, \dot{u}^a)) > r(s, (\mathbf{u}^{-a}, a))$$

Estimating Counterfactuals

- How to estimate counterfactual $r(s, (\mathbf{u}^{-a}, c^a))$?
- Extra simulations are expensive
- Learn a model of $r(s, \mathbf{u})$ instead [Proper & Tumer 2012] [Colby et al. 2016]
- COMA can just use the centralised critic $Q(s, \mathbf{u})$

Choosing c^a

- *Aristocrat utility* [Wolper & Tumer 2002] uses expectation instead:

$$D^a(s, \mathbf{u}) = r(s, \mathbf{u}) - \sum_{u^a} \pi^a(u^a|\tau^a) r(s, (\mathbf{u}^{-a}, u^a))$$

  but introduces *self-consistency* problems

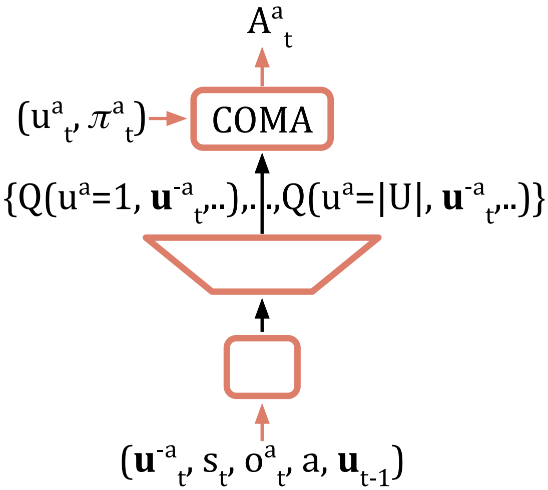
- COMA uses a *counterfactual baseline* instead:

$$g_a(\tau) = \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta(u_t^a|\tau_t^a) A^a(s_t, \mathbf{u}_t)$$

$$A^a(s, \mathbf{u}) = Q(s, \mathbf{u}) - \sum_{u^a} \pi^a(u^a|\tau^a) Q(s, (\mathbf{u}^{-a}, u^a))$$

  leaving gradient unbiased and ensuring self-consistency

Efficient Critic Representation



A^a_t

(u^a_t, π^a_t)

COMA

$\{Q(u^a=1, \mathbf{u}^{-a}_t,..),..,Q(u^a=|U|, \mathbf{u}^{-a}_t,..)\}$

$(\mathbf{u}^{-a}_t, s_t, o^a_t, a, \mathbf{u}_{t-1})$

Starcraft

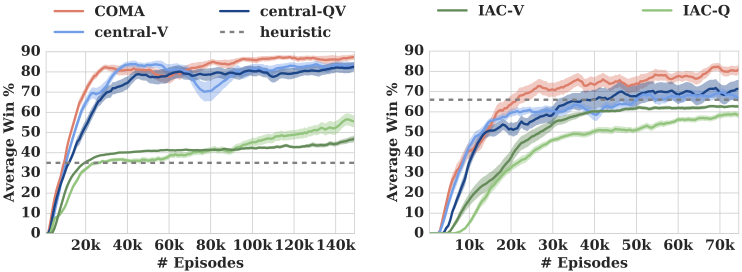
Starcraft Micromanagement [Synnaeve et al. 2016]

Decentralised Starcraft Micromanagement

Baseline Algorithms

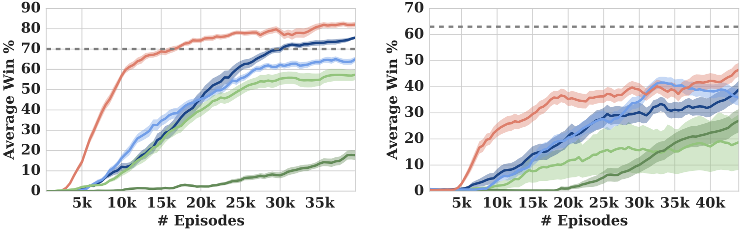
- *IAC-V:* independent actor-critic with $V(\tau^a)$ (TD error)
- *IAC-Q:* independent actor-critic with $A(\tau^a, u^a) = Q(\tau^a, u^a) - V(\tau^a)$
- *Central-V:* centralised critic $V(s)$ (TD error)
- *Central-QV:*
  - Centralised critics $Q(s, \mathbf{u})$ and $V(s)$
  - Advantage gradient $A(s, \mathbf{u}) = Q(s, \mathbf{u}) - V(s)$
  - COMA but with $b(s) = V(s)$

COMA Results vs. Baselines (Average Performance)



(a) 3 Marines

(b) 5 Marines



(c) 5 Wraiths

(d) 2 Dragoons & 3 Zealots

COMA Results vs. Centralised (Best Agents)

Map	COMA	Heuristic	DQN	GMEZO
3 Marines	98	74	-	-
5 Marines	95	98	99	100
5 Wraiths*	98	82	70	74
2 Dragoons & 3 Zealots	65	68	61	90

COMA Paper

Counterfactual Multi-Agent Policy Gradients

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras,
Nantas Nardelli, and Shimon Whiteson

The Outstanding Student Paper of AAAI-18

Factored Joint Value Functions

- Independent learners: no model of joint value function
- COMA: monolithic model of joint value function
- *Factored* joint value functions can improve scalability

Value Decomposition Networks

- VDNs [Sunehag et al., 2017] factor per agent:

$$Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \sum_{a=1}^{n} Q_i(\tau^a, u^a; \theta^a)$$

- Added benefit of decentralising the arg max:

$$\operatorname*{arg\,max}_{\mathbf{u}} Q_{tot}(\boldsymbol{\tau}, \mathbf{u}) = \begin{pmatrix} \arg\max_{u^1} Q_1(\tau^1, u^1) \\ \vdots \\ \arg\max_{u^n} Q_n(\tau^n, u^n) \end{pmatrix}$$

- No more hard greedification $\implies$ Q-learning, not actor-critic

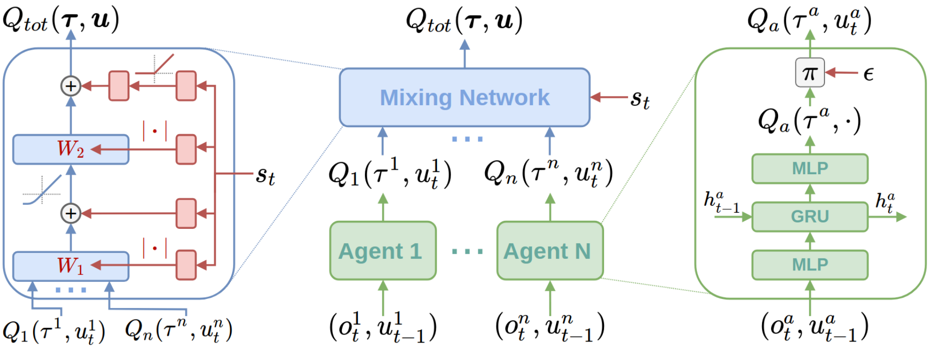
QMIX

- To decentralise arg max, it suffices to enforce:

$$\frac{\partial Q_{tot}}{\partial Q_a} \geq 0, \ \forall a \in A$$

- Use three networks:
  1. Agent network: represents $Q_i(\tau^a, u^a; \theta^a)$
  2. Mixing network: represents $Q_{tot}(\boldsymbol{\tau})$ using nonnegative weights
  3. Hypernetwork: generates weights of hypernetwork based on global s

QMIX Networks



Representational Capacity

		Agent 2	
		A	B
Agent 1	A	0	1
	B	1	2

linear &
monotonic

VDN & QMIX

		Agent 2	
		A	B
Agent 1	A	0	1
	B	1	8

nonlinear &
monotonic

just QMIX

		Agent 2	
		A	B
Agent 1	A	2	1
	B	1	8

nonlinear &
nonmonotonic

neither

Does it matter?

Two-Step Game

		Agent 2	
		A	B
Agent 1	A	7	7
	B	7	7

State 2A

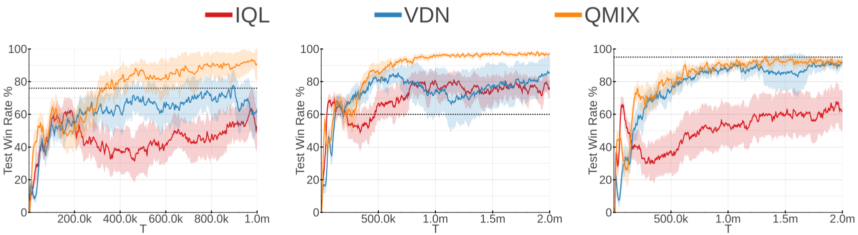
		Agent 2	
		A	B
Agent 1	A	0	1
	B	1	8

State 2B

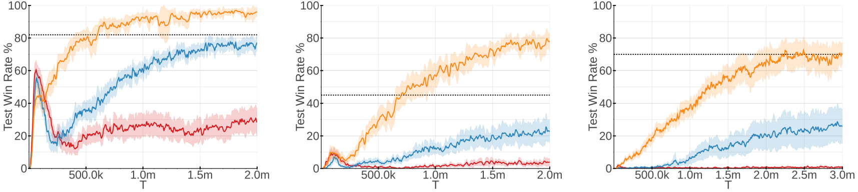
		State 1		State 2A		State 2B	
		A	B	A	B	A	B
VDN	A	6.94	6.94	6.99	7.02	-1.87	2.31
	B	6.35	6.36	6.99	7.02	2.33	6.51
		A	B	A	B	A	B
QMIX	A	6.93	6.93	7.00	7.00	0.00	1.00
	B	7.92	7.92	7.00	7.00	1.00	8.00

Decentralised Starcraft II Micromanagement

QMIX Results

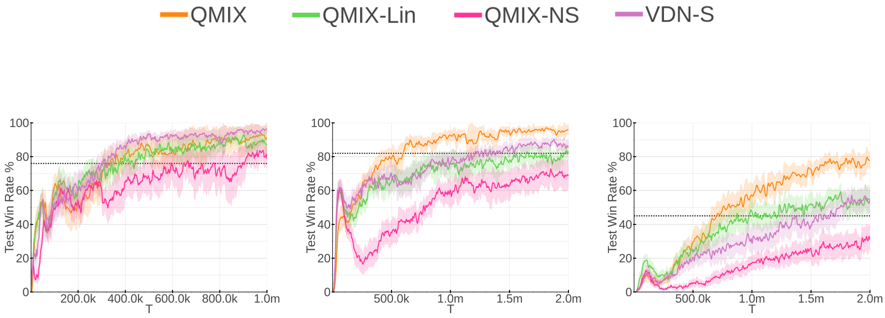


(e) 3 Marines (f) 5 Marines (g) 8 Marines



(h) 2 Stalkers & 3 Zealots (i) 3 Stalkers & 5 Zealots (j) 1 Col., 3 Stalk. & 5 Zeal.

QMIX Ablations



(k) 3 Marines (l) 2 Stalkers & 3 Zealots (m) 3 Stalkers & 5 Zealots

QMIX Paper

QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt,
Gregory Farquhar, Jakob Foerster, and Shimon Whiteson

ICML-18

Conclusions

- Multi-agent learning is tractable in the right setting
- Centralised learning of decentralised policies is such a setting
- Deep learning give new hope for scalable factored value functions
- Increasing reliance on critics $\rightarrow$ exploration is the next frontier