Seminar on Bayesian Machine Learning

Multi-agent Reinforcement Learning

Learning to Cooperate & Avoiding Non-stationarity

Arsenii Ashukha

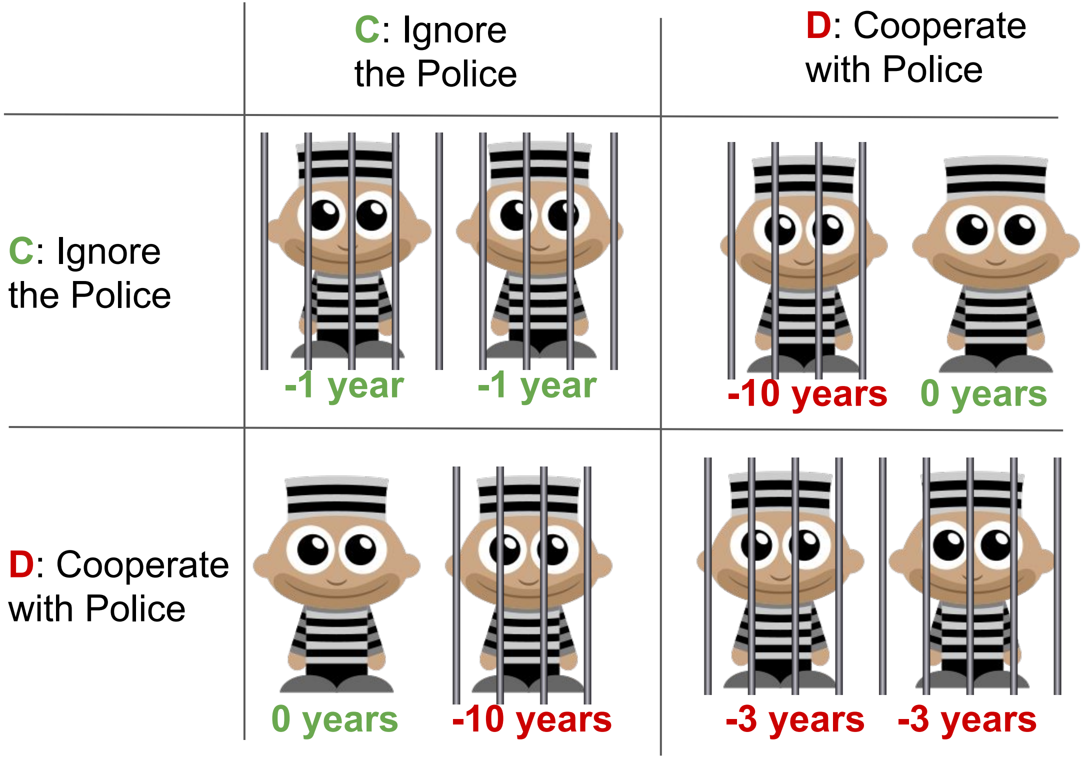
p(B|A)yesgroup.ru
UNIVERSITY OF AMSTERDAM
SAMSUNG AI Research

April 13th, 2017

Why should AI researchers think about MA systems?

- Compete, Cooperate, Communicate
- Louse individual reward in order to get a high joint reward
- Achieve global goals form local actions

Cooperation Study, Prisoner's Dilemma



Situation where:

- any individual may profit from selfishness
- unless too many agents do
- then the whole group loses

	C	D
C	-1, -1	-10, 0
D	0, -10	-3, -3

Matrix Game Social Dilemmas

	Cooperate	Defect
Cooperate	R, R	S, T
Defect	T, S	P, P

R - Reward
P - Penalty
T - Temptation
S - Sucker

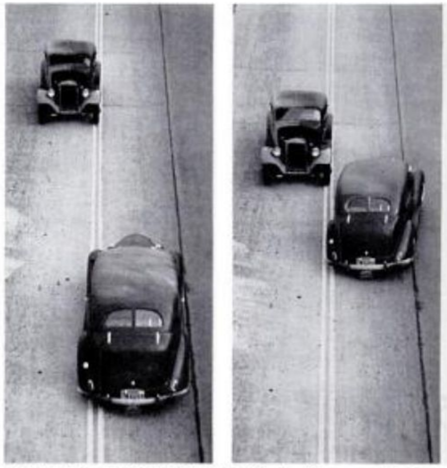
- mutual cooperation is preferred to mutual defection, $R > P$
- mutual cooperation is preferred to being exploited, $R > S$
- mutual cooperation is preferred to coop and defect $2R > T + S$
- one out of two:
    - **Greed**: Exploding cooperation prefer to mutual coop $T > R$
    - **Fear**: Mutual defection is preferred to being exploited $P > S$

Socially Undesirable Nash Equilibria

Chiken

T > R > S > P

3, 3	**1, 4**
4, 1	0, 0

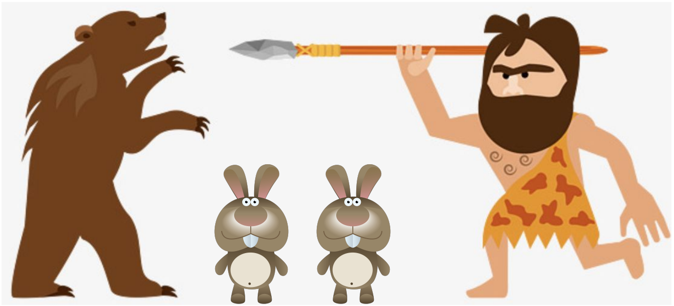


Greed drives defection

Strug Hunt

R > T > P > S

4, 4	0, 2
2, 0	**1, 1**



Fear drives defection

Prisoner's Dilemma

T > R > P > S

-1, -1	-10, 0
0, -10	**-3, -3**

Greed and Fear drives defection

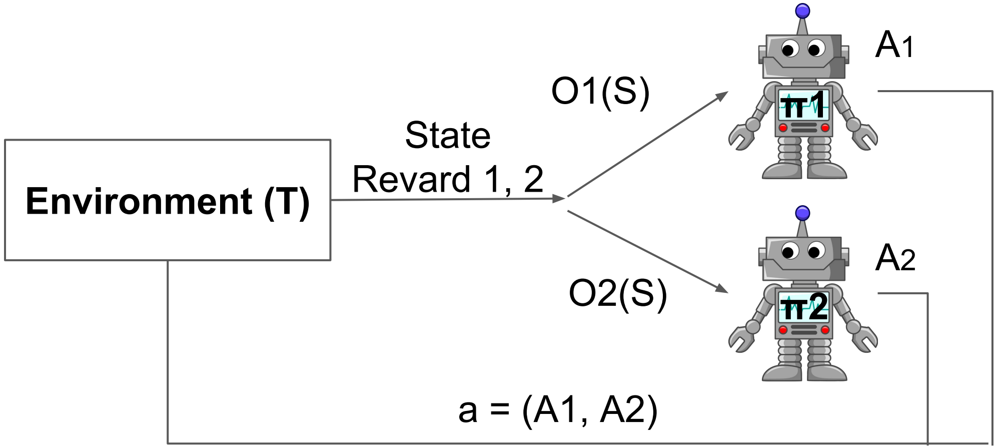
Reinforcement Learning Recup

- Real worlds social dilemmas are temporally extended
- Cooperativeness is graded quantity
- C/D have to be applied to policies, not just to single actions
- The 1st player’s action can affect the 2nd player's decision

Sequential Social Dilemmas (SSDs) address this issues.

Multi-agent Reinforcement Learning in Sequential Social Dilemmas https://arxiv.org/abs/1702.03037

Sequential Social Dilemmas (SSDs)



- When |S| = 1, A in {C, D}, O(s) = s the **Markov game** is **Matrix game**
- Long-term playoff, for a pair of policies π = (π1, π2)

$$V_i^{\vec{\pi}}(s_0) = \mathbb{E}_{\vec{a}_t \sim \vec{\pi}(O(s_t)), s_{t+1} \sim \mathcal{T}(s_t, \vec{a}_t)} \left[\sum_{t=0}^{\infty} \gamma^t r_i(s_t, \vec{a}_t) \right]$$

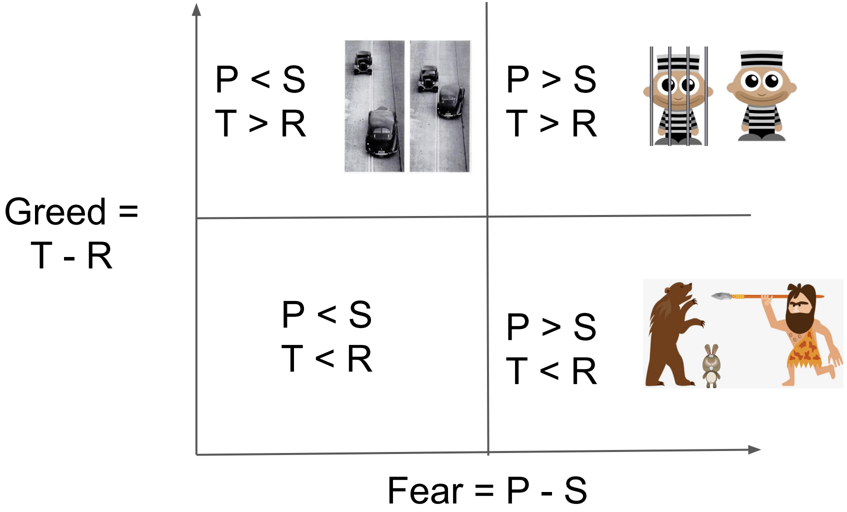
- The outcome for cooperative and defecting policies result in a matrix game

Sequential Social Dilemmas (SSDs)

- Given two set of policies π_C and π_D, evaluate them

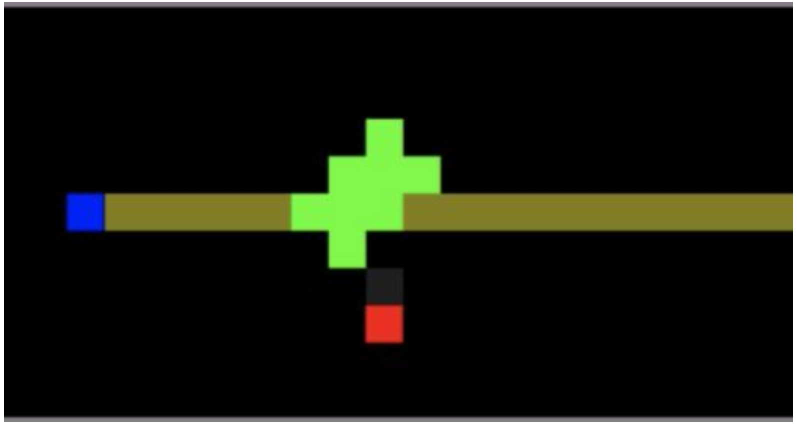
$$R(s) := V_1^{\pi^C,\pi^C}(s) = V_2^{\pi^C,\pi^C}(s) \qquad S(s) := V_1^{\pi^C,\pi^D}(s) = V_2^{\pi^D,\pi^C}(s)$$
$$P(s) := V_1^{\pi^D,\pi^D}(s) = V_2^{\pi^D,\pi^D}(s) \qquad T(s) := V_1^{\pi^D,\pi^C}(s) = V_2^{\pi^C,\pi^D}(s)$$

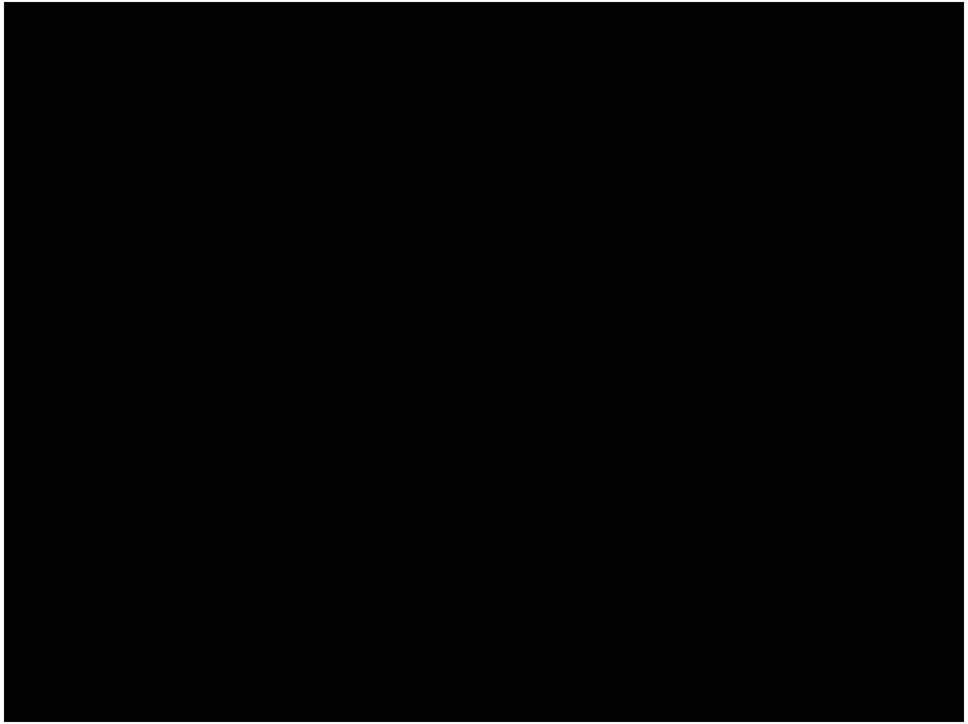
- For every outcome compute Greed = T - R and Fear = P - S



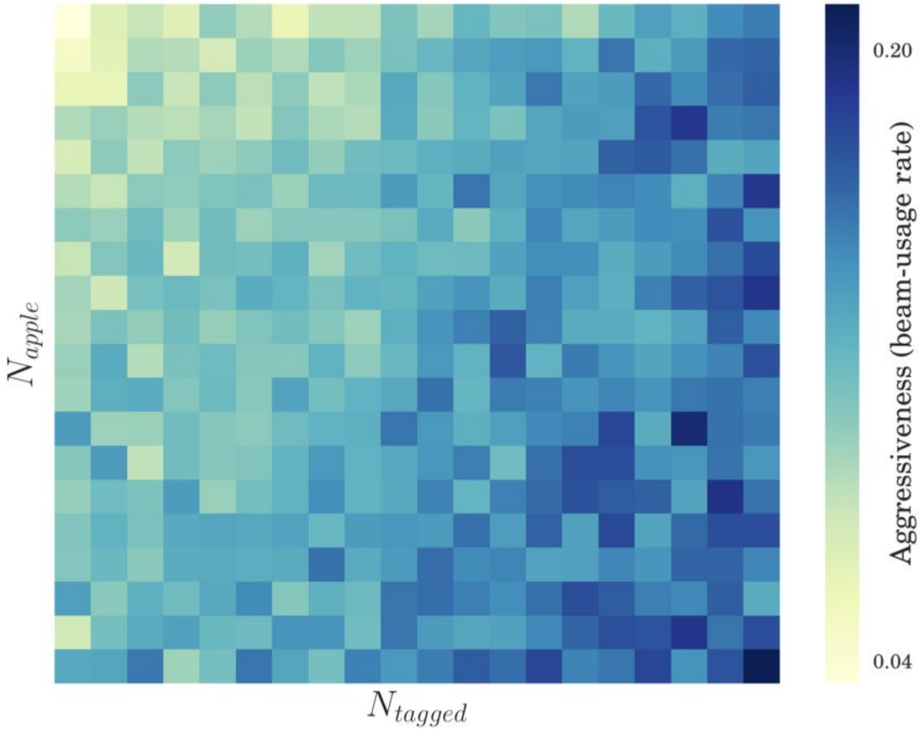
R - Reward
T - Temptation
P - Penalty
S - Sucker

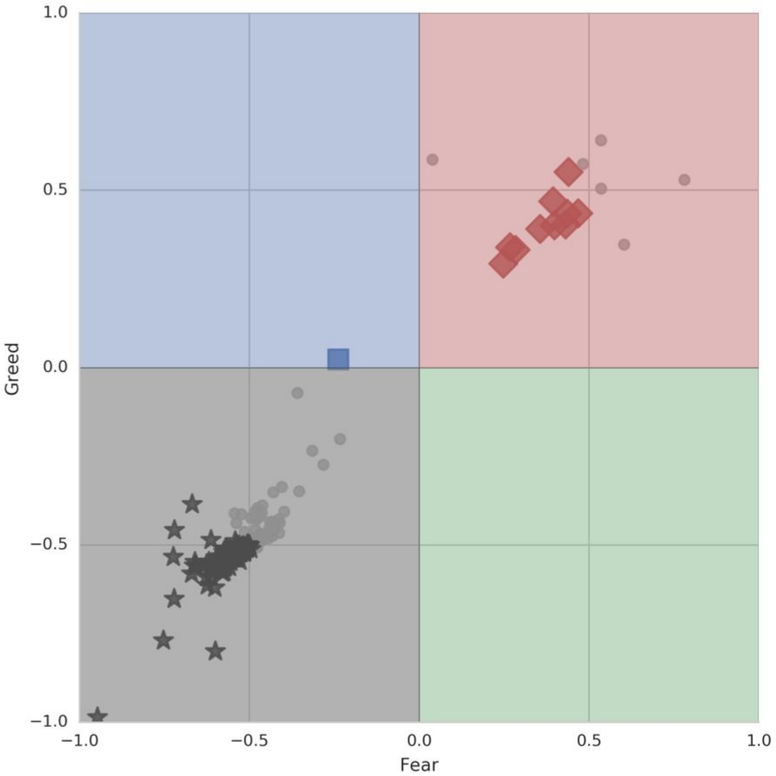
Gathering Game



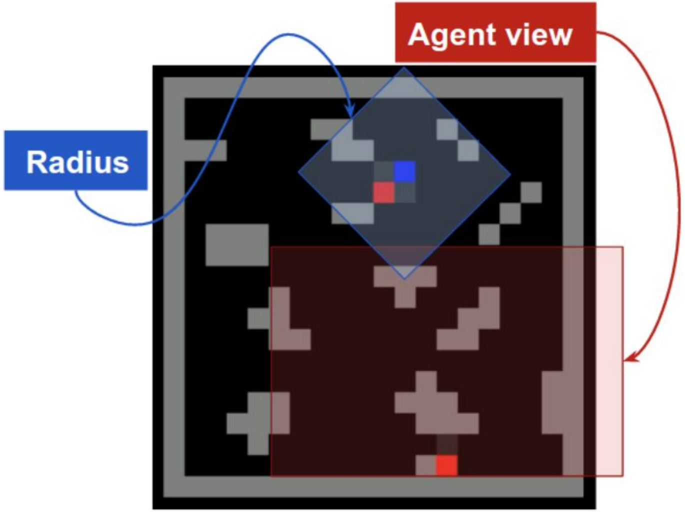


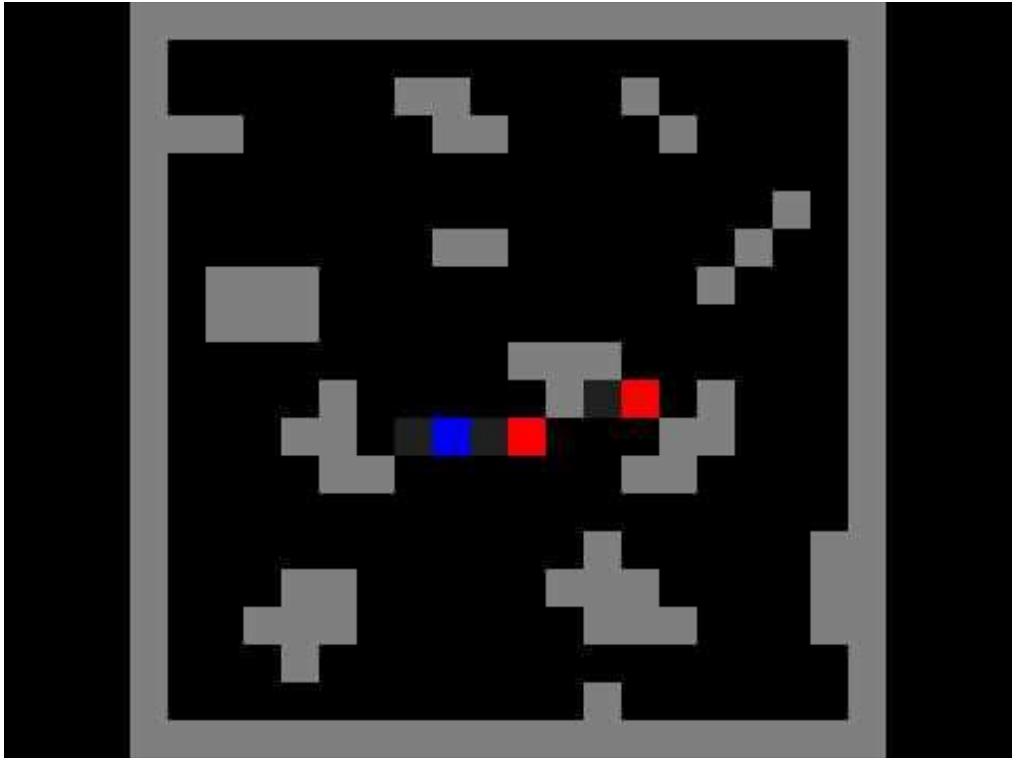
Gathering Game



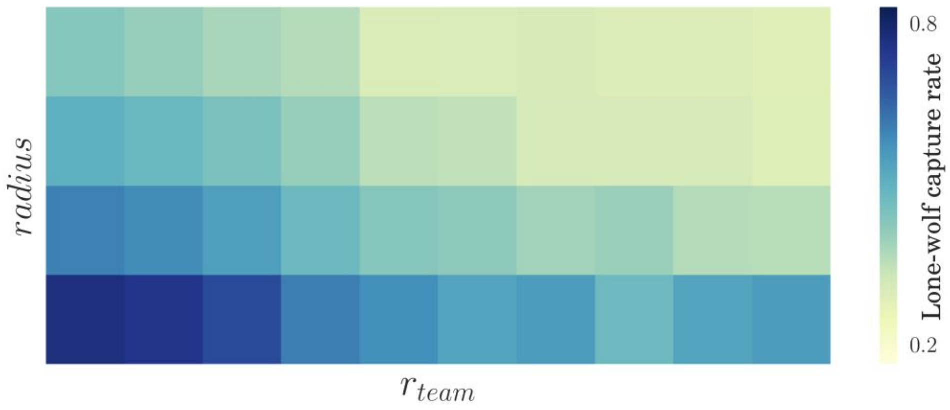


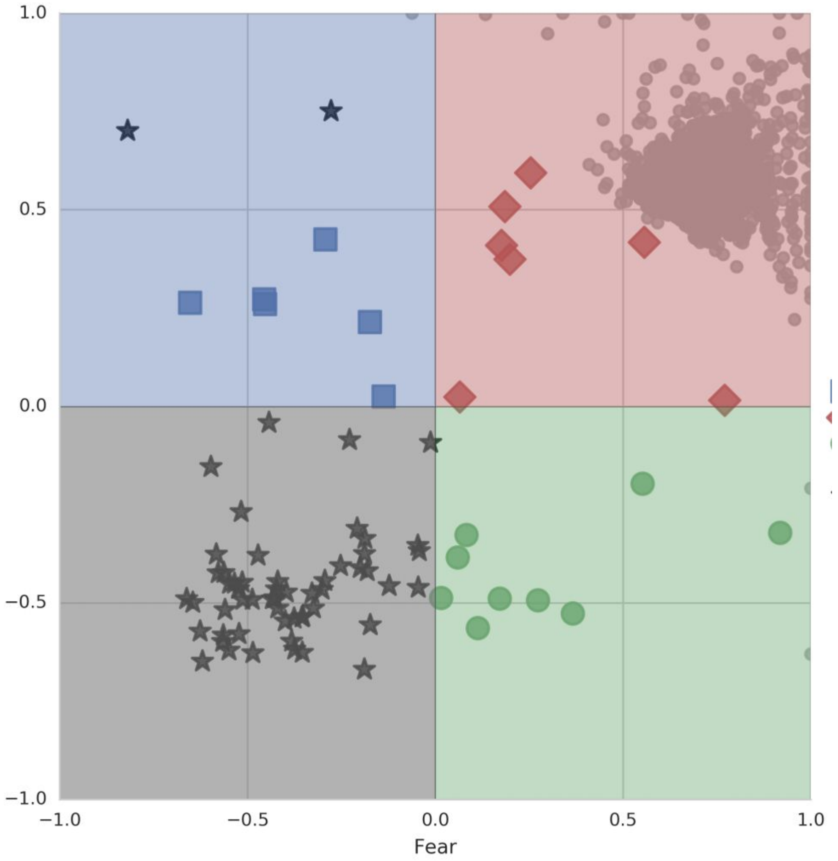
Wolfpack game: two wolves to chase a prey





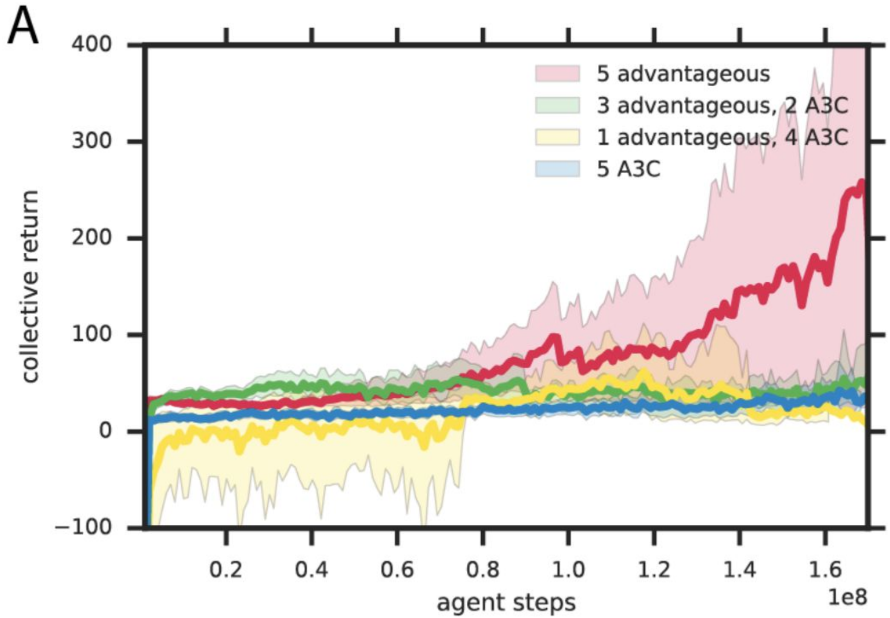
Gathering Game





SSDs Learning: MA-SSDs and Inequity Aversion

$$U_i(r_i, \dots r_N) = r_i - \frac{\alpha_i}{N-1} \sum_{j \neq i} \max(r_j - r_i, 0) - \frac{\beta_i}{N-1} \sum_{j \neq i} \max(r_i - r_j, 0)$$



Inequity aversion resolves intertemporal social dilemmas, https://arxiv.org/abs/1803.08884

Deep RL Recup

- Bellman Optimality equation

$$Q^*(s,u) = r(s,u) + \gamma \sum_{s'} P(s'|s,u) \max_{u'} Q^*(s',u')$$

- Q-Learning and Deep Q-Networks (DQN)

$$\mathcal{L}(\theta) = \mathbb{E}_{s,a,r,s'}[(Q^*(s,a|\theta) - y)^2], \qquad \text{where} \qquad y = r + \gamma \max_{a'} \bar{Q}^*(s',a')$$

- Policy Gradient (PG) Algorithms

$$J(\theta) = \mathbb{E}_{s\sim p^{\boldsymbol{\pi}}, a\sim\boldsymbol{\pi}_\theta}[R] \qquad \nabla_\theta J(\theta) = \mathbb{E}_{s\sim p^{\boldsymbol{\pi}}, a\sim\boldsymbol{\pi}_\theta}[\nabla_\theta \log \boldsymbol{\pi}_\theta(a|s) Q^{\boldsymbol{\pi}}(s,a)]$$

- Deterministic Policy Gradient (DPG) Algorithms

$$J(\theta) = \mathbb{E}_{s\sim p^{\boldsymbol{\mu}}}[R(s,a)] \qquad \nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}[\nabla_\theta \boldsymbol{\mu}_\theta(a|s) \nabla_a Q^{\boldsymbol{\mu}}(s,a)|_{a=\boldsymbol{\mu}_\theta(s)}]$$

Multi-Agent Actor-Critic

$$P(s'|s, a_1, ..., a_N, \boldsymbol{\pi}_1, ..., \boldsymbol{\pi}_N) = P(s'|s, a_1, ..., a_N) = P(s'|s, a_1, ..., a_N, \boldsymbol{\pi}'_1, ..., \boldsymbol{\pi}'_N)$$

- Multi-Agent Actor-Critic

$$\nabla_{\theta_i} J(\theta_i) = \mathbb{E}_{s \sim p^{\boldsymbol{\mu}}, a_i \sim \boldsymbol{\pi}_i}[\nabla_{\theta_i} \log \boldsymbol{\pi}_i(a_i|o_i) Q_i^{\boldsymbol{\pi}}(\mathbf{x}, a_1, ..., a_N)]$$

- The centralized Q function is updated as

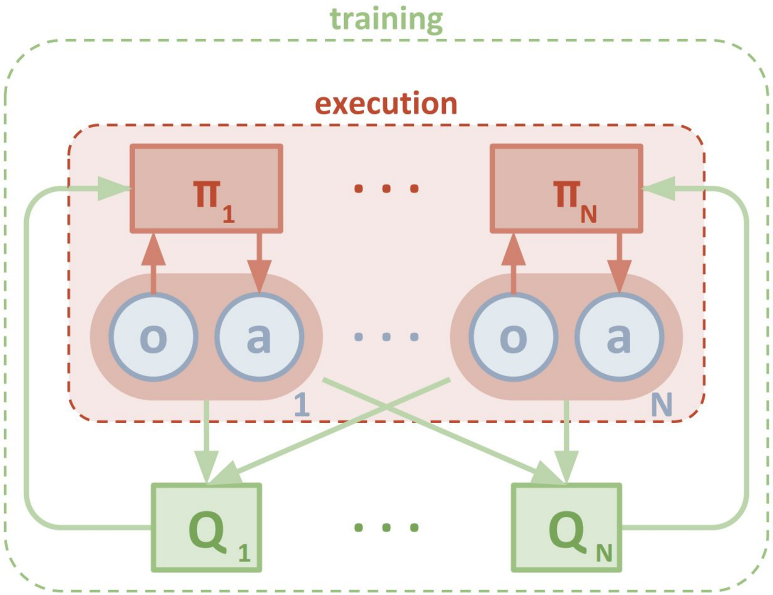
$$\mathcal{L}(\theta_i) = \mathbb{E}_{\mathbf{x},a,r,\mathbf{x}'}[(Q_i^{\boldsymbol{\mu}}(\mathbf{x}, a_1, \ldots, a_N) - y)^2]$$

$$y = r_i + \gamma\, Q_i^{\boldsymbol{\mu}'}(\mathbf{x}', a'_1, \ldots, a'_N)\big|_{a'_j = \boldsymbol{\mu}'_j(o_j)}$$

- Inferring Policies of Other Agents

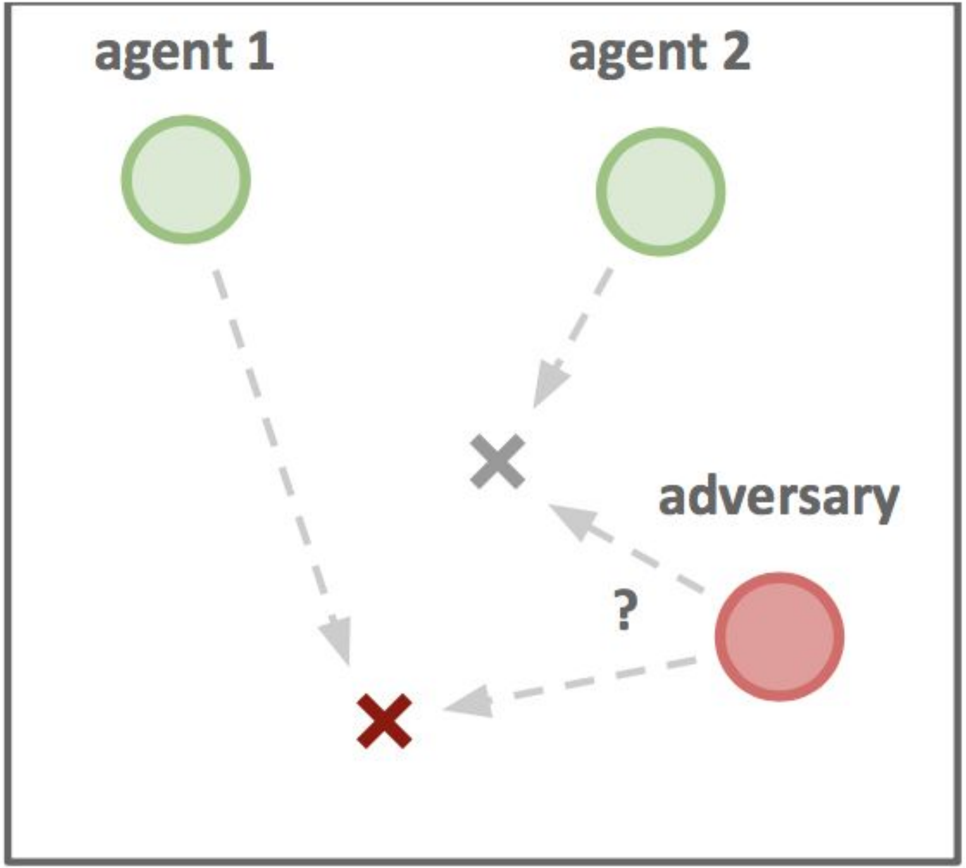
$$\hat{y} = r_i + \gamma Q_i^{\boldsymbol{\mu}'}(\mathbf{x}', \hat{\boldsymbol{\mu}}_i'^1(o_1), \ldots, \boldsymbol{\mu}'_i(o_i), \ldots, \hat{\boldsymbol{\mu}}_i'^N(o_N))$$

$$\mathcal{L}(\phi_i^j) = -\mathbb{E}_{o_j, a_j}\left[\log \hat{\boldsymbol{\mu}}_i^j(a_j|o_j) + \lambda H(\hat{\boldsymbol{\mu}}_i^j)\right]$$

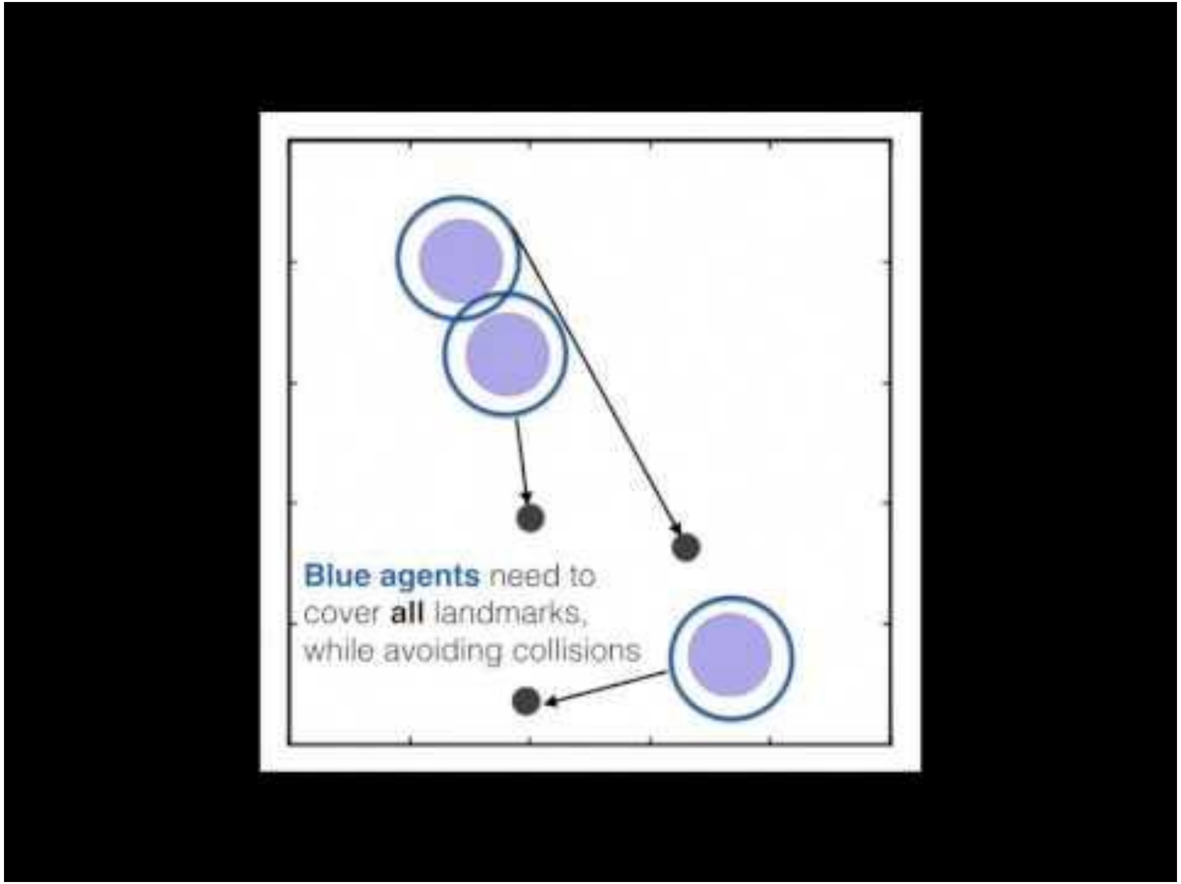


Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, https://arxiv.org/abs/1706.02275

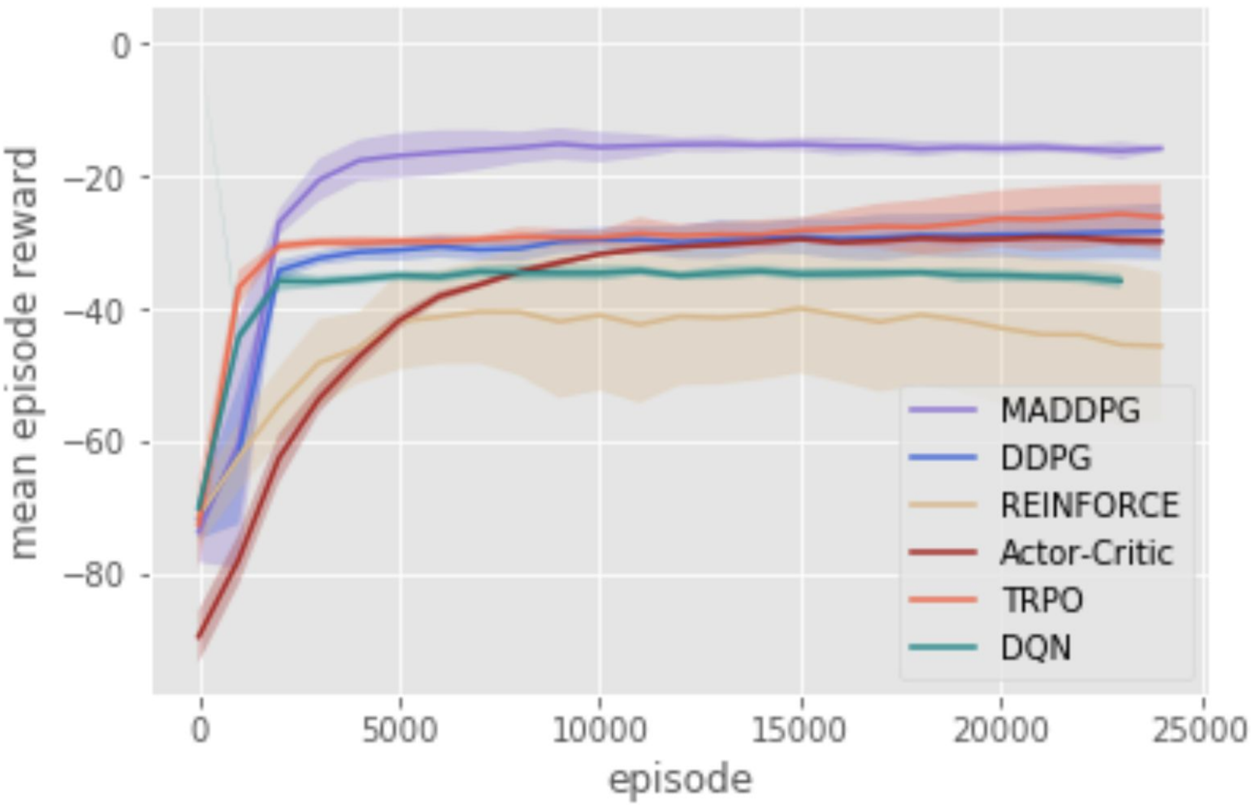
Multi-Agent Actor-Critic: Physical Deception



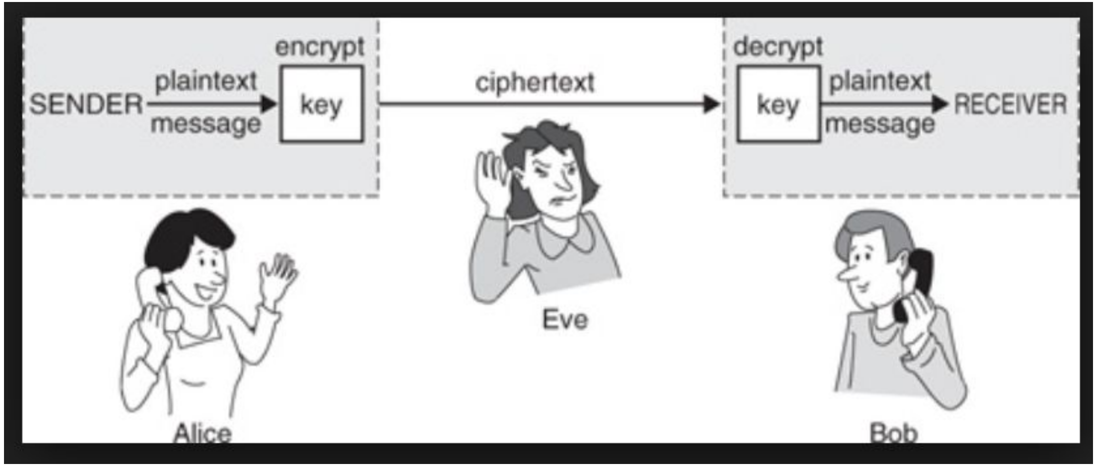
Multi-Agent Actor-Critic



Multi-Agent Actor-Critic: cooperative communication



Multi-Agent Actor-Critic: cooperative communication



Stabilising Experience Replay for Deep MARL

Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning

Jakob Foerster [*1] **Nantas Nardelli** [*1] **Gregory Farquhar** [1] **Triantafyllos Afouras** [1]
Philip. H. S. Torr [1] **Pushmeet Kohli** [2] **Shimon Whiteson** [1]

Abstract

Many real-world problems, such as network packet routing and urban traffic control, are naturally modeled as multi-agent *reinforcement learning* (RL) problems. However, existing multi-agent RL methods typically scale poorly in

multi-agent systems. Unfortunately, tackling such problems with traditional RL is not straightforward.

If all agents observe the true state, then we can model a cooperative multi-agent system as a single meta-agent. However, the size of this meta-agent's action space grows exponentially in the number of agents. Furthermore, it is not

- Q-function conditioned on other policies

$$Q_a^*(s, u_a|\boldsymbol{\pi}_{-a}) = \sum_{\mathbf{u}_{-a}} \boldsymbol{\pi}_{-a}(\mathbf{u}_{-a}|s)\left[r(s, u_a, \mathbf{u}_{-a}) + \gamma \sum_{s'} P(s'|s, u_a, \mathbf{u}_{-a}) \max_{u'_a} Q_a^*(s', u'_a)\right]$$

- Q-function conditioned on other policies

$$\langle s, u_a, r, \pi(\mathbf{u}_{-a}|s), s' \rangle^{(t_c)} \qquad \mathcal{L}(\theta) = \sum_{i=1}^{b} \frac{\boldsymbol{\pi}_{-a}^{t_r}(\mathbf{u}_{-a}|s)}{\boldsymbol{\pi}_{-a}^{t_i}(\mathbf{u}_{-a}|s)}[(y_i^{DQN} - Q(s, u; \theta))^2]$$

Conclusion

- Multi-agent RL just has started growing up
- MA systems have more in common with real environments

- Classical Cooperation study can be applied to Markov Games
- Non-stationarity avoiding methods work under strong assumptions
- We will probably see much more breakthrough research in this area

[1] Multi-agent Reinforcement Learning in Sequential Social Dilemmas, https://arxiv.org/abs/1702.03037
[2] Inequity aversion resolves intertemporal social dilemmas https://arxiv.org/abs/1803.08884
[3] Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments, https://arxiv.org/abs/1706.02275
[4] Stabilising Experience Replay for Deep Multi-Agent Reinforcement Learning, https://arxiv.org/abs/1702.08887
[5] The Role of Multi-Agent Learning in Artificial Intelligence Research at DeepMind, https://youtu.be/CvL-KV3IBcM