Multi-Agent Reinforcement Learning for DeepRacer Multi-Car Racing

Kaiqing Zhang

Amazon AI Labs Intern

Mentor: Sahika Genc

2019.08.07

Overview of My Contributions

	Stage 1: Object-Avoidance	Stage 2: Racing with Fixed-Policy Opponent	Stage 3: Multi-Car Racing	Longer Term
Science	Design reward function		Formulate Robust MARL motivated from DeepRacer	Organize DeepRacer MARL reading group
	Warm start to improve sample-efficiency		Develop theory for Robust MARL	Present and brainstorm ideas on MARL
Engineering	Real track test	Build MADDPG [Lowe '17] with one learner car in Coach	TO DO: Test algorithms	Build MARL framework in Coach codebase
Documentation	Report & Quip doc		Paper draft	Idea Bank for DeepRacer

Outline

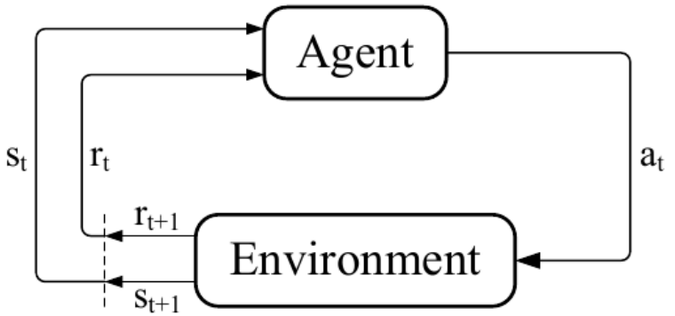
- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms & Theoretical Results
- MARL for Coach
- Other contributions

Outline

- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms & Theoretical Results
- MARL for Coach
- Other contributions

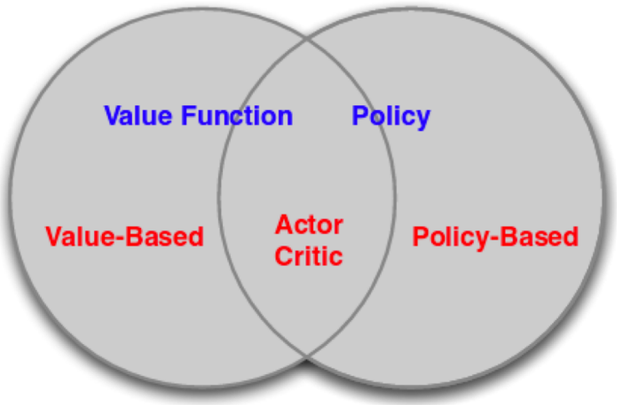
Reinforcement Learning (RL)

- Reinforcement Learning: solving Markov decision processes/optimal control problem without knowing the model
- (Stationary) policy: $\boldsymbol{\pi}(s,a) : \mathcal{S} \times \mathcal{A} \to [0,1]$
- Goal: maximize accumulated/time-average reward



$$\max_{\boldsymbol{\pi}} \quad J(\boldsymbol{\pi}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| \boldsymbol{\pi}\right]$$

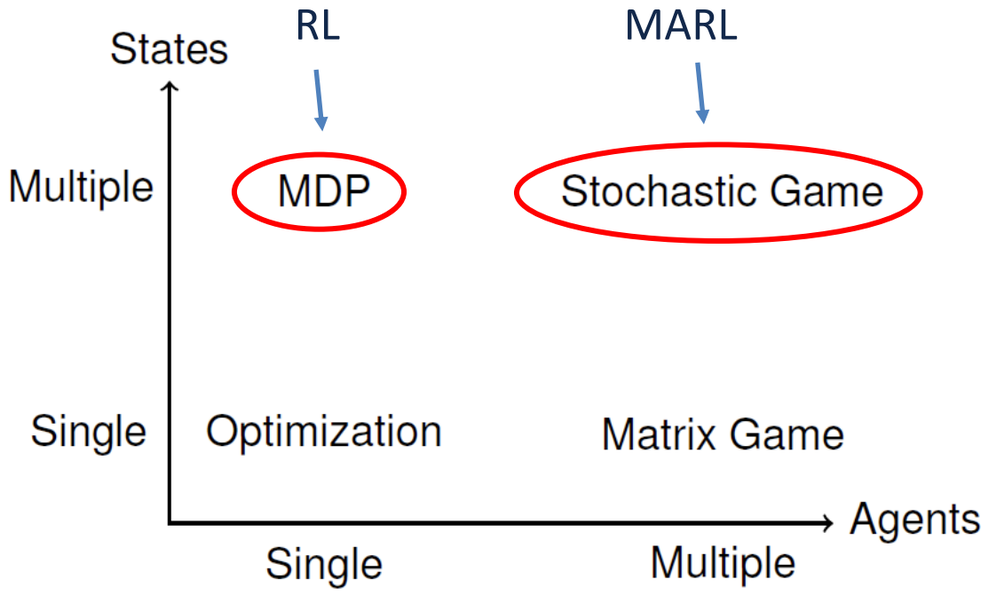
- Algorithms:
  - Critic-only/value-based: e.g., Q-learning, SARSA, DQN, DDQN
  - Actor-only/policy-based: e.g., policy gradient, PPO, TRPO
  - Actor-critic enjoys both advantages



Source: *Reinforcement learning Lecture Notes*, David Silver, 2016

Multi-Agent RL (MARL)

- Many practical applications involve multiple agents
  - Robotics, autonomous vehicles
  - Sensor networks
  - Video games, game of Go
- Settings:
  - Cooperative
  - Competitive
  - Mixed of the two
- Algorithms:
  - Tabular case/linear function approximation, with convergence guarantee
  - Neural-nets for function approximation, mostly purely empirical
- See my previous literature review slides for more details



Modeling of MARL

- Framework: Stochastic/Markov Games [Littman '94] $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, P, \{R^i\}_{i\in\mathcal{N}}, \gamma \rangle$

$$R^i(s, a^1, \cdots, a^N, s') : \mathcal{S} \times \mathcal{A}^1 \times \cdots \times \mathcal{A}^N \times \mathcal{S} \to \mathbb{R} \qquad P(s' \,|\, s, a) : \mathcal{S} \times \mathcal{A}^1 \times \cdot \times \mathcal{A}^N \times \mathcal{S} \to [0, 1]$$

  - Policy: $\boldsymbol{\pi}^i(s, a^i) : \mathcal{S} \times \mathcal{A}^i \to [0, 1]$, joint policy $\boldsymbol{\pi}(s, a) := \prod_{i\in\mathcal{N}} \boldsymbol{\pi}^i(s, a^i) : \mathcal{S} \times \mathcal{A} \to [0, 1]$
  - Goal:

$$\max_{\boldsymbol{\pi}^i} \quad J^i(\boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i}) := \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t^i \middle| \boldsymbol{\pi}^i, \boldsymbol{\pi}^{-i} \right]$$

  - Solution concept: Nash equilibrium (NE) $\pi_* = (\pi_*^1, \cdots, \pi_*^N)$

$$\forall i \in \mathcal{N}, \quad J^i(\pi_*^i, \pi_*^{-i}) \geq J^i(\pi^i, \pi_*^{-i})$$

  - Special cases: team & zero-sum

- Challenges:
  - Non-stationarity [Tan '93]
  - Curse of dimensionality: with # of agents
  - Learning goal [Shoham '06]

MARL Algorithms

- Value-based:
  - Minimax-Q [Littman '94], Nash-Q [Hu and Wellman '00], Friend-Foe-Q [Littman '01]

$$Q_{t+1}(s_t, a_t) = (1-\alpha_t)Q_t(s_t, a_t) + \alpha_t \left[r_t + \beta \max_a Q_t(s_{t+1}, a) \right]$$

$$Q^i_{t+1}(s, a^1, \ldots, a^n) = (1-\alpha_t)Q^i_t(s, a^1, \ldots, a^n) + \alpha_t \left[r^i_t + \beta Nash Q^i_t(s') \right]$$

  - Neural networks-based: Use NN to approximate Q-function
- Policy-based: parametrize $\pi^i \rightarrow \pi^i_{\theta^i}$
  - Multi-agent policy gradient: $\nabla_{\theta^i} J^i(\theta) = \mathbb{E}\left[\nabla \log \pi^i_{\theta^i}(a^i \mid s) \cdot Q^i_{\pi_\theta}(s, a^1, \cdots, a^N) \right]$
  - Each agent improves $\pi^i_{\theta^i}(a^i \mid s)$ following its own policy gradient
- Actor-critic: additionally parametrize $Q^i_{\pi_\theta}(\cdot,\cdot) = Q^i_{\pi_\theta}(\cdot,\cdot;\omega^i)$
  - [Gupta et al., '17; Lowe et al., '17; Srinivasan et al., '19]

Multi-Car Racing & MARL

- Stage 1: Object-avoidance
  - Essentially single-agent RL, with reward re-design
  - Single-agent RL, e.g., PPO, suffices

- Stage 2: Racing with fixed-policy opponent
  - Minimum viable product (MVP) case
  - Essentially also single-agent RL, with reward re-design $\mathcal{M}^i := \langle \mathcal{S}, \mathcal{A}^i, \widetilde{R}^i, \widetilde{P}^i, \gamma \rangle$

    with $\widetilde{R}^i(s, a^i) := \int_{\mathcal{A}^{-i}} R^i(s, a^i, a^{-i}) \pi^{-i}(a^{-i}|s) da^{-i}$ and $\widetilde{P}^i(\cdot|s, a^i) := \int_{\mathcal{A}^{-i}} P(\cdot|s, a^i, a^{-i}) \pi^{-i}(a^{-i}|s) da^{-i}$
  - Either single agent RL, or MARL with only one agent updating the policy

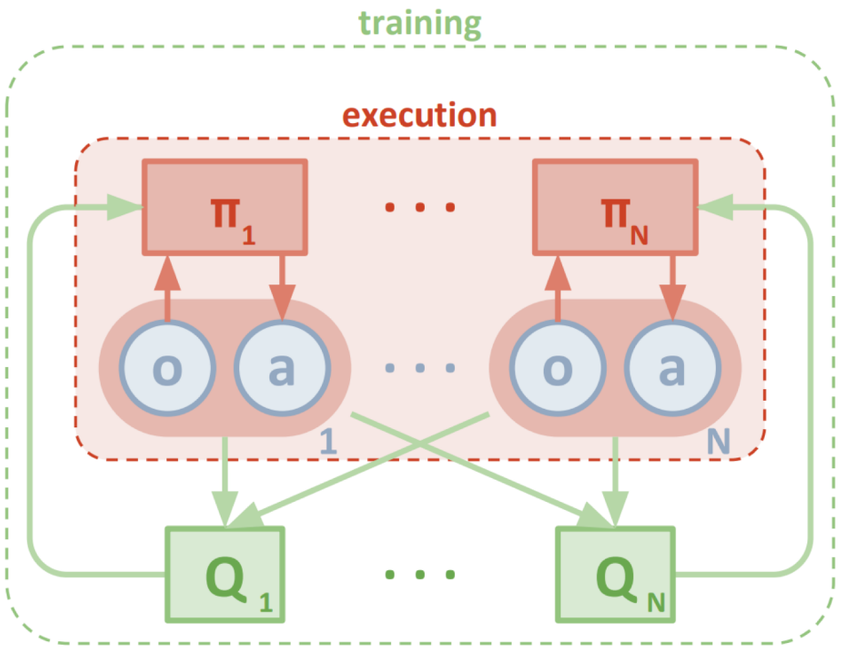
- Stage 3: Multi-car racing
  - Actual MARL framework: general-sum Markov games
  - State-of-the-art MARL algorithms should work, e.g., MADDPG in [Lowe et al., 17’]

Multi-Car Racing & MARL

- Modeled as a Markov game with *partial observability* (POMG)
  - State: joint position $s = (s^1, s^2)$, with $s^i = (x^i, y^i)$
  - Action: velocity $a^i = v^i = (v_x^i, v_y^i)$
  - Transition: physical model $s_{t+1}^i = s_t^i + \Delta t * (a_t^i)$
  - Reward: finish as fast as possible + avoid collision $R^i(s, a^1, a^2)$
  - Observation: image from on-car camera $O^i(s', a^1, a^2)$
- In general, POMG is NEXP-complete in theory [Bernstein et al., ’00]
- Two solutions (see report [Zhang and Genc, ’19])
  - *God camera*: partial observability to full observability (Markov game)
    - Optimal in theory
    - Existing MARL algorithms can apply
  - Heuristic *approximation*: use “joint observation” as “state”, i.e., $Q^i(o^1, \cdots, o^N, a^1, \cdots, a^N)$ and $V^i(o^1, \cdots, o^N)$

Multi-Car Racing & MARL

- Algorithmic structure for Stages 2 & 3 (Idea from MADDPG [Lowe et al., '17])
  - Centralized critic (in simulator) + decentralized actor (in sim2real)
  - Obs. and actions of other agents can be used to ease training, since $Q^i(o^1, \cdots, o^N, a^1, \cdots, a^N)$
  - But only *local* observation is used for execution, since $\pi^i(a^i | o^i)$
  - Actor-critic, also DPG, PPO, TRPO, fit in this structure
  - For Stage 2: only one agent updates



- "Multi-car Arena" service for DeepRacer
  - Multi-car racing environment with customized reward
  - Customer models can be submitted as a payment
  - Models are collected to improve the arena

Source: [Lowe et al., 2017]

Outline

- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms & Theoretical Results
- MARL for Coach
- Other contributions

Motivation

- What if different multi-car arenas use different customized rewards?
  - Each customer trains the Nash equilibrium policy for its agent
  - NE policies of different games may not constitute an equilibrium anymore
  - Trained policy should be robust w.r.t. this reward uncertainty

- Formally, robust RL + MARL
  - Model: robust Markov games [Kardeş et al., ’11]

$$\bar{\mathcal{G}} := \langle \mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \{\bar{\mathcal{R}}_s^i\}_{(i,s)\in\mathcal{N}\times\mathcal{S}}, \{\bar{\mathcal{P}}_s\}_{s\in\mathcal{S}}, \gamma \rangle$$

  - Uncertainty sets of reward functions $\bar{\mathcal{R}}_s^i \subseteq \mathbb{R}^{|\mathcal{A}|}$ and transitions $\bar{\mathcal{P}}_s$
  - “Distribution-free” way to model uncertainty, as in *robust optimization* [Nemirovski et al., ’09]

Formulation

- Bellman equation
  - Fact: an agent plays to optimize the worst-case of “current-stage cost” + “worst-case cost from current-stage on”

$$\bar{V}^i(s) = \max_{\pi^i(\cdot|s)} \min_{\substack{\bar{R}^i_s \in \bar{\mathcal{R}}^i_s \\ \bar{P}(\cdot|s,\cdot)\in\bar{\mathcal{P}}_s}} \sum_{a\in\mathcal{A}} \prod_{j=1}^{N} \pi^j(a^j|s)\left(\bar{R}^i(s,a) + \gamma \sum_{s'\in\mathcal{S}} \bar{P}(s'|s,a)\bar{V}^i(s')\right)$$

- Interesting view: the uncertainty is the decision made by a “nature” agent
- Solution concept: robust Markov perfect Nash equilibrium (RMPNE)
  - Joint policy $\pi_* = (\pi_*^1, \pi_*^2, \cdots, \pi_*^N)$, if for any $s \in \mathcal{S}$ and $i \in [N]$, there exists $\bar{V} = (\bar{V}^1, \cdots, \bar{V}^N)$

$$\pi^i_*(\cdot|s) \in \operatorname*{argmax}_{\pi^i(\cdot|s)} \min_{\substack{\bar{R}^i_s \in \bar{\mathcal{R}}^i_s \\ \bar{P}(\cdot|s,\cdot)\in\bar{\mathcal{P}}_s}} \sum_{a\in\mathcal{A}} \pi^i(a^i|s) \prod_{j\neq i} \pi^j_*(a^j|s)\left(\bar{R}^i(s,a) + \gamma \sum_{s'\in\mathcal{S}} \bar{P}(s'|s,a)\bar{V}^i(s')\right)$$

Existence of the Equilibrium

Proposition 1: with finite state action spaces $\mathcal{S}$ and $\mathcal{A}$, and compact uncertain sets $\bar{\mathcal{R}}_s^i$ and $\bar{\mathcal{P}}_s$, there always exists a RMPNE.

- Proof based on Kakutani fixed-point theorem, following [Kardeş et al., '11]
- For DeepRacer, transition is certain, i.e., $\bar{\mathcal{P}}_s = \{P(\cdot|s,\cdot)\}$, then Bellman equation

$$\bar{V}^i(s) = \max_{\pi^i(\cdot|s)} \min_{\pi^{0,i}(\cdot|s)} \mathbb{E}_{\bar{R}_s^i \sim \pi^{0,i}(\cdot|s), a^i \sim \pi^i(\cdot|s), a^{-i} \sim \pi^{-i}(\cdot|s)} \left(\bar{R}^i(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \bar{V}^i(s') \right)$$

- Nature policy $\pi^0 := \{\pi^{0,i}\}_{i \in \mathcal{N}}$, with $\pi^{0,i} : \mathcal{S} \to \Delta(\bar{\mathcal{R}})$
- Equivalent solution concept: RMPNE with nature (NRMPNE) $\widetilde{\pi}_* = (\pi_*^0, \pi_*^1, \cdots, \pi_*^N)$

$$\left(\pi_*^i(\cdot|s), \pi_*^{0,i}(\cdot|s)\right) \in \underset{\pi^i(\cdot|s), \pi^{0,i}(\cdot|s)}{\text{argmaxmin}} \sum_{\bar{R}_s^i \in \bar{\mathcal{R}}_s^i} \pi^{0,i}\left(\bar{R}_s^i \middle| s\right) \sum_{a \in \mathcal{A}} \pi^i(a^i|s) \prod_{j \neq i} \pi_*^j(a^j|s) \left(\bar{R}^i(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) \bar{V}^i(s') \right)$$

- Existence of RMPNE $\Longleftrightarrow$ existence of NRMPNE with deterministic $\pi_*^0 = \left\{\pi_*^{0,i}\right\}_{i \in \mathcal{N}}$

Outline

- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms & Theoretical Results
- MARL for Coach
- Other contributions

Value-based Algorithms

- Known model: value iteration

$$\bar{V}^i_{t+1}(s) = \max_{\pi^i(\cdot|s)} \min_{\bar{R}^i_s \in \bar{\mathcal{R}}_s} \sum_{a\in\mathcal{A}} \prod_{j=1}^{N} \pi^j(a^j|s)\left(\bar{R}^i(s,a) + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)\bar{V}^i_t(s')\right)$$

$$=: \mathcal{T}^i_{\mathcal{V}}(\bar{V}^i_t)$$

  - Converges if $\mathcal{T}^i_{\mathcal{V}}$ is a contracting mapping

- Unknown model: Q-learning
  - Bellman equation for Q-value: $\bar{Q}^i(s,a) := \pi^{0,i}_*(s)[a] + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) \sum_{a'} \prod_{j=1}^{N} \pi^j_*(a'^j|s')\bar{Q}^i(s',a')$

  - Learning update $\bar{Q}^i_{t+1}(s_t,a_t) := (1-\alpha_t)\cdot \bar{Q}^i_t(s_t,a_t) + \alpha_t \cdot [\bar{R}^i_{*,t} + \gamma \bar{Q}^i_t(s_{t+1},a_{t+1})],$

$$\text{with } \bar{R}^i_{*,t} = \pi^{0,i}_{*,t}(s_t)[a_t], \quad a^i_{t+1} \sim \pi^i_{*,t}(\cdot|s_{t+1}),$$

where $\pi^{0,i}_{*,t}$ and $\pi^i_{*,t}$ are equilibrium policies that solve *stage-game* $\bar{Q}_t = (\bar{Q}^1_t, \cdots, \bar{Q}^N_t)$

Value-based Algorithms

Theorem 2 (Informal): Under the same assumptions on the learning rate α_t, the exploration policy, and the equilibrium-point properties as Nash-Q learning[Hu and Wellman '00], Q-learning for robust MARL converges to the RMPNE almost surely.

- Proof borrows the idea from [Hu and Wellman '00]
- However, all agents' Q-value estimates $\bar{Q}_t = (\bar{Q}_t^1, \cdots, \bar{Q}_t^N)$ needs to be maintained
- Cannot have "decentralized execution"

Policy-based Algorithms

- Parameterize the policies $\widetilde{\pi}_\theta := (\pi_{\theta^0}, \pi_{\theta^1}, \cdots, \pi_{\theta^N})$, with $\pi_{\theta^0} := \{\pi_{\theta^{0,i}}\}_{i\in\mathcal{N}}$
- Value functions given $\widetilde{\pi}_\theta$

$$\bar{V}^i_{\widetilde{\pi}_\theta}(s) = \sum_{a\in\mathcal{A}} \prod_{j=1}^{N} \pi_{\theta^j}(a^j|s)\left(\pi_{\theta^{0,i}}(s)[a] + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a)\bar{V}^i_{\widetilde{\pi}_\theta}(s')\right)$$

$$\bar{Q}^i_{\widetilde{\pi}_\theta}(s,a) = \pi_{\theta^0}(s)[a] + \gamma \sum_{s'\in\mathcal{S}} P(s'|s,a) \sum_{a'\in\mathcal{A}} \prod_{j=1}^{N} \pi_{\theta^j}(a'^j|s')\bar{Q}^i_{\widetilde{\pi}_\theta}(s',a')$$

Theorem 3 [Policy Gradient Theorem for Robust MARL] The policy gradients w.r.t. each agent's parameter have the following form:

$$\nabla_{\theta^i} J^i(\theta) = \mathbb{E}_{s\sim\rho^{s_0}_{\widetilde{\pi}_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\nabla \log \pi_{\theta^i}(a^i|s)\cdot \bar{Q}^i_{\widetilde{\pi}_\theta}(s,a)\right],$$

$$\nabla_{\theta^{0,i}} J^i(\theta) = \mathbb{E}_{s\sim\rho^{s_0}_{\widetilde{\pi}_\theta}, a\sim\pi_\theta(\cdot|s)}\left[\nabla \pi_{\theta^{0,i}}(s)[a]\right]$$

Actor-Critic for Robust MARL

- Also parameterize $\bar{Q}^i_{\widetilde{\pi}_\theta}$ by $\bar{Q}_{\omega^i}$
- Two timescale *online* updates: $\lim_{t\to\infty} \beta_t/\alpha_t = 0$

Critic

$$\delta^i_t = \pi_{\theta^{0,i}_t}(s_t)[a_t] + \gamma \bar{Q}_{\omega^i_t}(s_{t+1}, a_{t+1}) - \bar{Q}_{\omega^i_t}(s_t, a_t)$$

$$\omega^i_{t+1} = \omega^i_t + \alpha_t \cdot \delta^i_t \cdot \nabla \bar{Q}_{\omega^i_t}(s_t, a_t)$$

Actor

$$\theta^i_{t+1} = \theta^i_t + \beta_t \cdot \nabla \log \pi_{\theta^i_t}(a^i_t | s_t) \cdot \bar{Q}_{\omega^i_t}(s_t, a_t)$$

$$\theta^{0,i}_{t+1} = \theta^{0,i}_t - \beta_t \cdot \nabla \pi_{\theta^{0,i}_t}(s_t)[a_t],$$

- Fits in “centralized critic + decentralized actor” framework

Actor-Critic for Robust MARL

- Pseudocode for batch version:

Algorithm 1 Actor-Critic for Robust Multi-Agent RL:

Initialization of Q-value parameters $\{\omega_0^i\}_{i\in\mathcal{N}}$, and policy parameters $\{\theta_0^i\}_{i\in\mathcal{N}}$ and $\theta_0^0 := \{\theta_0^{0,i}\}_{i\in\mathcal{N}}$.
for episode = 1 to M **do**
  Receive an initial state s
  for $t = 1, \cdots T$ **do**
    For each agent i, sample action $a^i \sim \pi_{\theta_t^i}$ with current policy $\pi_{\theta_t^i}$
    Execute joint $a = (a^1, \cdots, a^N)$, and observe new state s'
    Store (s, a, s') in replay buffer $\mathcal{D}$, let $s' \leftarrow s$
    for agent $i = 1$ to N **do**
      Sample a random mini-batch of S samples of (s_t, a_t, s_{t+1}) from $\mathcal{D}$
      Set

$$y_t = \pi_{\theta^{0,i}}(s_t)[a_t] + \gamma \bar{Q}_{\omega^i}(s_{t+1}, a_{t+1}^1, \cdots, a_{t+1}^N)\big|_{a_{t+1}^i \sim \pi_{\theta^i}(\cdot|s_{t+1})},$$

      Update critic by minimizing the loss $\mathcal{L}(\theta^i) = \frac{1}{S}\sum_{t=1}^{S}\left(y_t - \bar{Q}_{\omega^i}(s_t, a_t)\right)^2$
      Update actor using the sampled policy gradient

$$\nabla_{\theta^i} J^i(\theta) \approx \frac{1}{S}\sum_{t=1}^{S} \nabla \log \pi_{\theta^i}(a_t^i|s_t)\bar{Q}_{\pi_\theta}^i(s_t, a_t),\ \theta'^i = (1-\tau)\theta^i + \tau\nabla_{\theta^i} J^i(\theta)$$

$$\nabla_{\theta^{0,i}} J^i(\theta) \approx \frac{1}{S}\sum_{t=1}^{S} \nabla \pi_{\theta^{0,i}}(s_t)[a_t], \qquad \theta'^{0,i} = (1-\tau)\theta^{0,i} + \tau\nabla_{\theta^{0,i}} J^i(\theta)$$

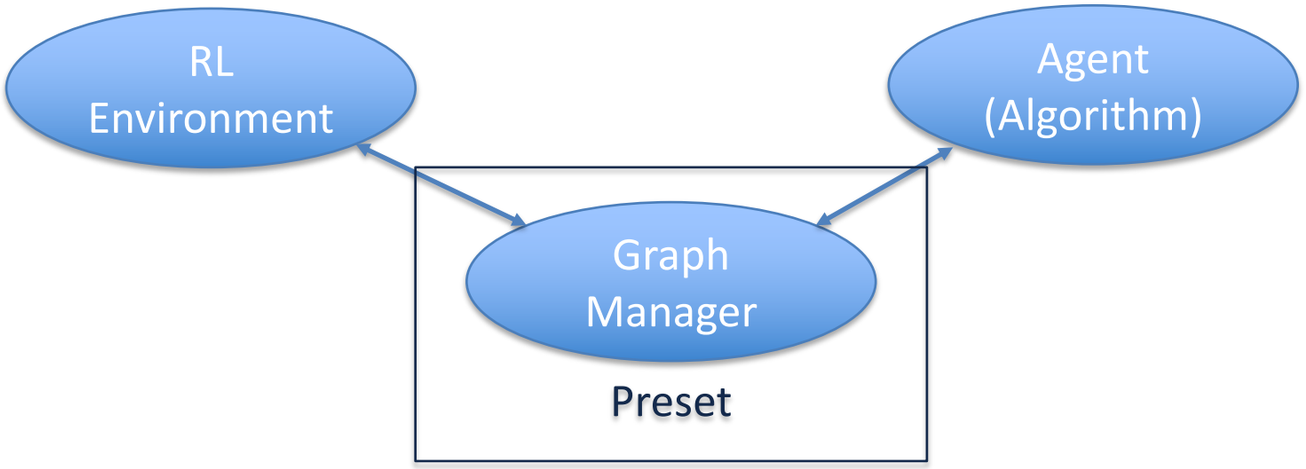
    end for
  end for
end for

Outline

- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms & Theoretical Results
- MARL for Coach
- Other contributions

Code Structure

- Code structure in Coach RL



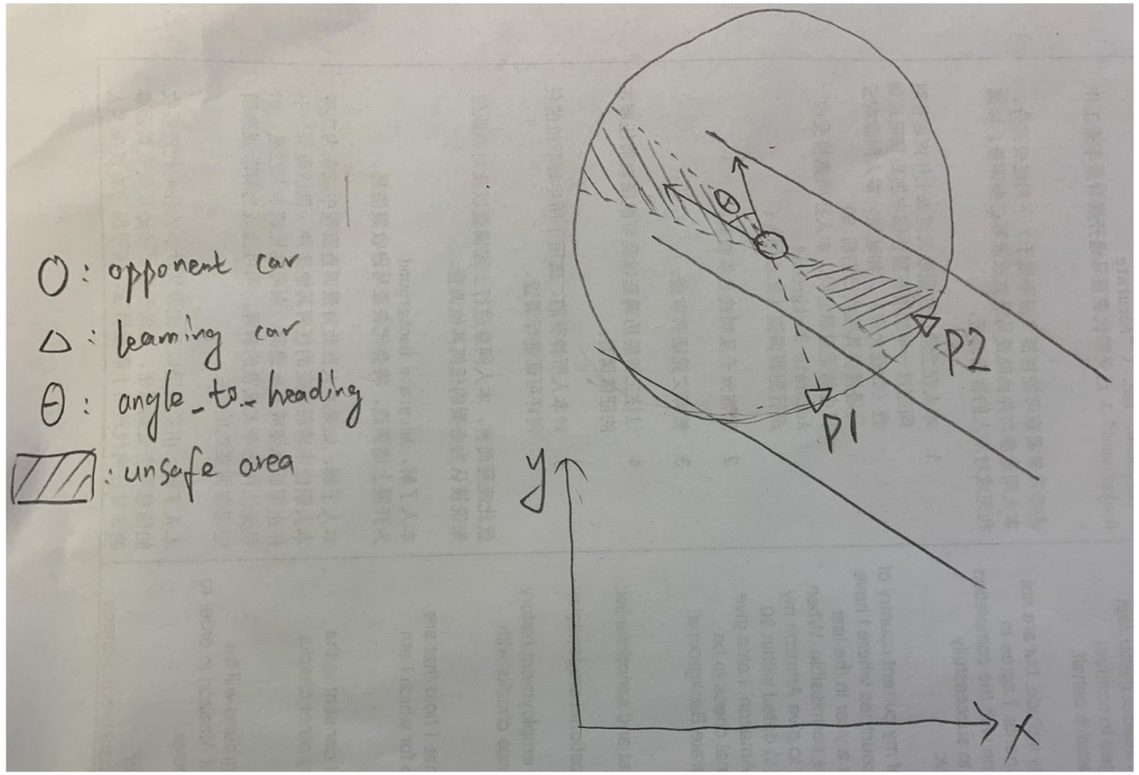
- Modifications
  - Graph Manager: *multiagent_step()*, with joint *act_n* and *obs_n* as inputs, joint *env_response_n* as outputs
  - Agent: *maddpg_agent()*, with *agents and agent_index* as input; *train_multiagent()*, concatenate the *observation* and *action* to prepare *batch; learn_from_batch(),* update NN weights for *critic* using *_n* data, and for *actor* using local *observation*
  - Environment: GymVectorEnvironment.level to support graph_manager
  - Preset: combine everything

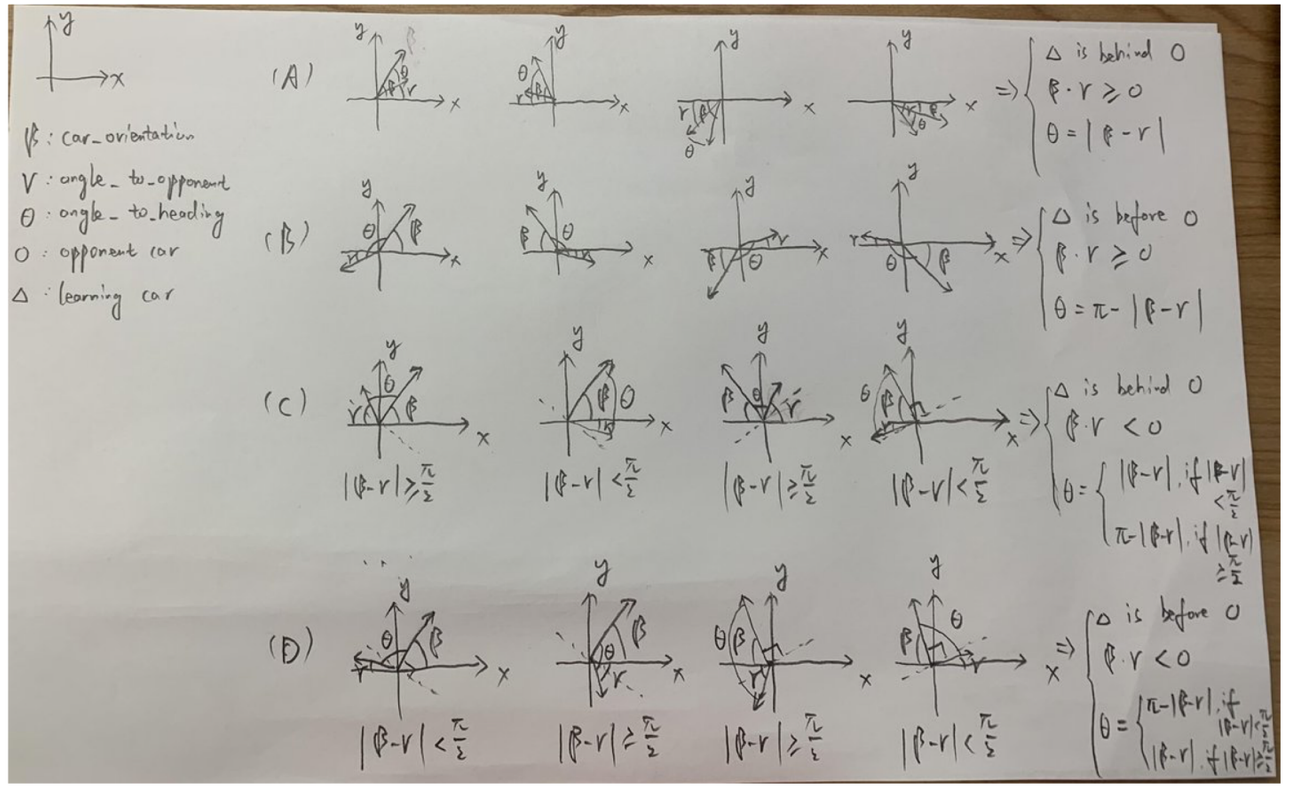
Outline

- Multi-Agent RL & Multi-Car Racing
- New problem: Robust MARL
- Algorithms
- Theoretical Results
- MARL for Coach
- Other contributions

Other Contributions

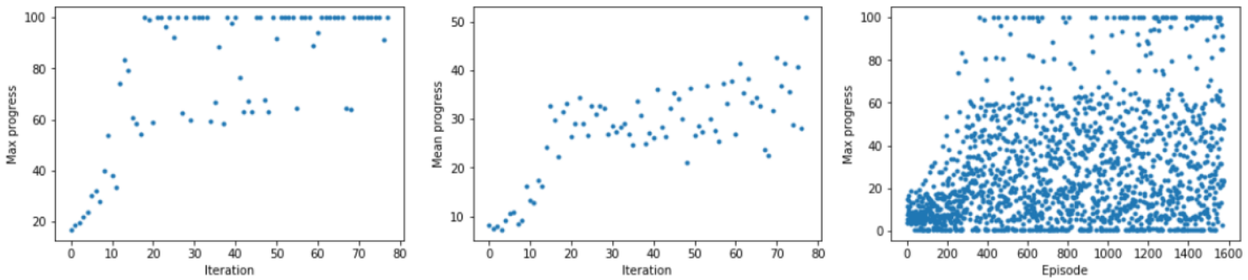
- Design reward for object-avoidance and MVP case: https://quip-amazon.com/fpfoA9DUYyG1
  - Idea: use relative information between cars, e.g., position, velocity, heading, etc.



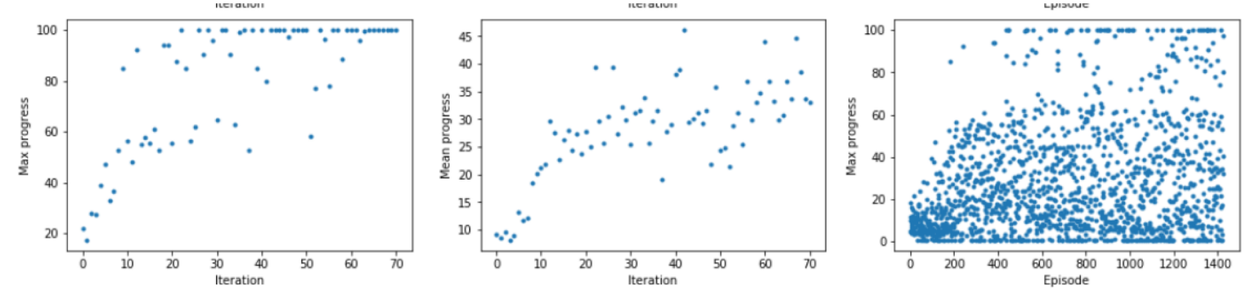


Other Contributions

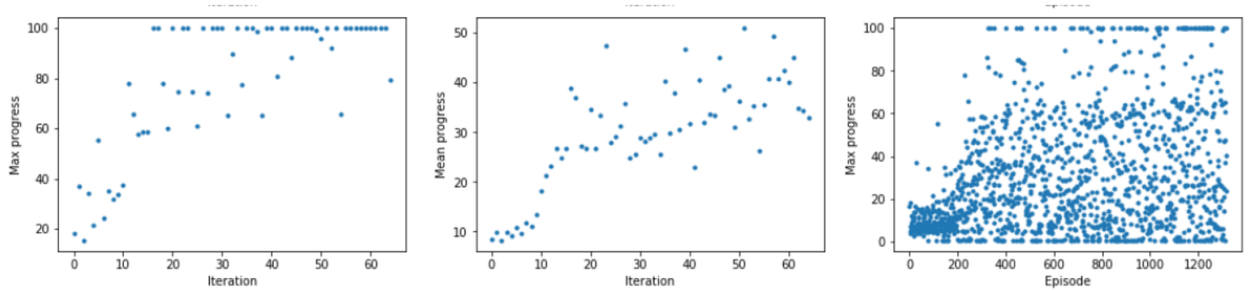
- Design reward for object-avoidance and MVP case: https://quip-amazon.com/fpfoA9DUYyG1
  - Performance improved (with Tao)



Baseline with simple “bubble” reward, 51% at iter. 77



Safe_Angle = 45, 46.1% at iter. 42

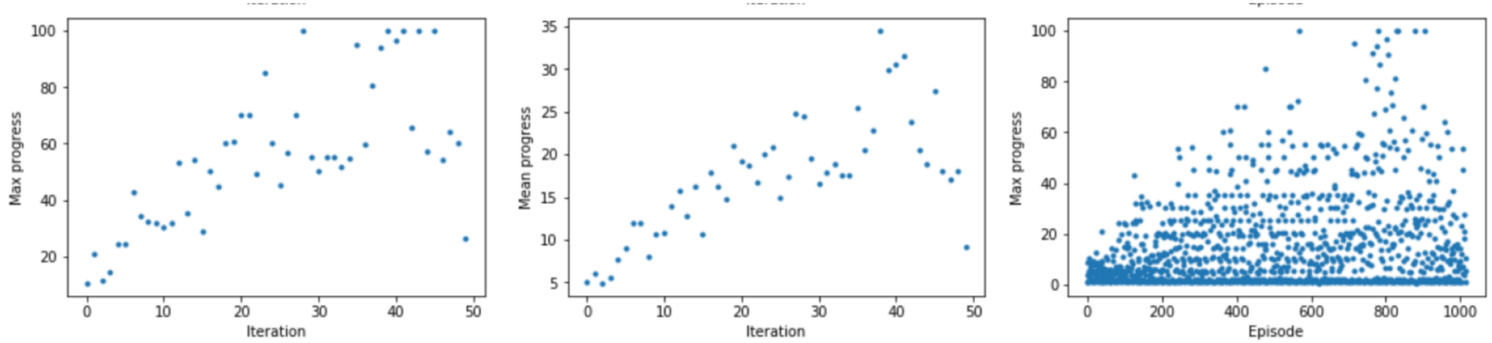


Safe_Angle = 30, 51% at iter. 51

Other Contributions

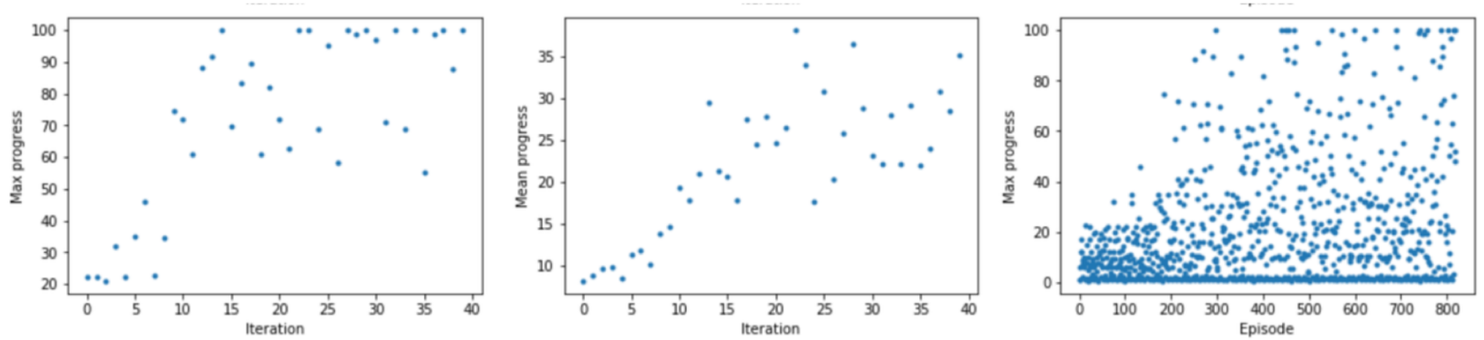
- "Warm start" idea to improve sample efficiency
  - Idea: use the pretrained model from single-car racing, only need to "learn" avoiding
  - Performance improved (with Tao & Yunzhe)

Cold start



cold start

Warm start with "inner lane model"



warm start

Other Contributions

- Start the MARL reading group and a reading list: https://quip-amazon.com/coenAWLl4el4

- Start an “Idea bank” on RL for DeepRacer (with Sunil): https://quip-amazon.com/mSV6AnnwFpQp

Highlights & Conclusions

- Three stages are required to achieve actual multi-car racing
- Studied robust MARL for the first time, motivated from DeepRacer competition (potentially a research paper)
- Centralized critic + decentralized actor can be the training framework for multi-car racing
- Implemented this framework in Coach for later MARL training

References

- M. L. Littman. “Markov games as a framework for multi-agent reinforcement learning,” *In Machine learning proceedings,* pages 157–163. Elsevier, 1994.
- C. Boutilier. “Planning, learning and coordination in multi-agent decision processes,” *In Conference on Theoretical Aspects of Rationality and Knowledge,* pages 195–210, 1996.
- M. L. Littman. “Friend-or-foe Q-learning in general-sum games,” *In International Conference on Machine Learning,* volume 1, pages 322–328, 2001.
- Y. Shoham, R. Powers, and T. Grenager. “Multi-agent reinforcement learning: A critical survey,” *Technical Report,* 2003.
- X. Wang and T. Sandholm. “Reinforcement learning to play an optimal Nash equilibrium in team Markov games,” *In Advances in Neural Information Processing Systems*, pages 1603–1610, 2003.
- J. Hu and M.P. Wellman. “Nash Q-learning for general-sum stochastic games,” *Journal of Machine Learning Research,* 4(11):1039–1069, 2003.
- T. Smith and R. Simmons. “Heuristic search value iteration for POMDPs,” *In Proceedings of the Conference on Uncertainty in Artificial Intelligence*, pages 520–527, 2004.
- D. Silver and J. Veness. “Monte-Carlo planning in large POMDPs,” *In Advances in Neural Information Processing Systems,* pages 2164–2172, 2010.
- S. Kar, J.M. Moura, and H.V. Poor. “QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations,” *IEEE Transactions on Signal Processing*, 61(7):1848–1862, 2013.
- J. Foerster, Y.M. Assael, N. Freitas, and S. Whiteson. “Learning to communicate with deep multi-agent reinforcement learning,” *In Advances in Neural Information Processing Systems*, pages 2137–2145, 2016.

References

- J. Perolat, F. Strub, B. Piot, and O. Pietquin. “Learning Nash equilibrium for general-sum Markov games from batch data.,” arXiv preprint arXiv:1606.08718, 2016.
- R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. “Multi-agent actor-critic for mixed cooperative competitive environments,” *arXiv preprint arXiv:1706.02275*, 2017.
- G. Arslan and S. Yüksel, “Decentralized Q-learning for stochastic teams and games,” *IEEE Transactions on Automatic Control*, vol. 62, no. 4, pp. 1545–1558, 2017.
- A. Mathkar and V. S. Borkar. “Distributed reinforcement learning via gossip,” *IEEE Transactions on Automatic Control* , 62(3):1465–1470, 2017.
- K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Başar. “Fully decentralized multi-agent reinforcement learning with networked agents,” *In International Conference on Machine Learning*, pages 5872–5881, 2018.
- L. Lin, K. Zhang, Z. Yang, Z. Wang, T. Başar, R. Sandhu, and J. Liu. “A Communication-Efficient Multi-Agent Actor-Critic Algorithm for Distributed Reinforcement Learning,” in *Proceedings of IEEE Conference on Decision and Control*, IEEE, 2019.
- J. K. Gupta, M. Egorov, and M. Kochenderfer. “Cooperative multi- agent control using deep reinforcement learning,” *In International Conference on Autonomous Agents and Multiagent Systems,* pages 66–83. Springer, 2017.
- E. Kardeş, O. Fernando, and W. H. Randolph. “Discounted robust stochastic games and an application to queueing control,” *Operations Research* 59.2, pages 365-382, 2011
- E. Kardeş. “On discounted stochastic games with incomplete information on payoffs and a security application,” *Operations Research Letters* 42.1: pages 7-11, 2014

References

- E. V. Mazumdar, M. I Jordan, and S. S. Sastry. “On the convergence of gradient-based learning in continuous games,” *arXiv preprint arXiv:1804.05464,* 2018.
- E. V. Mazumdar, M. I Jordan, and S. S. Sastry. “On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games,” *arXiv preprint arXiv:1901.00838,* 2019.
- C. Jin, P. Netrapalli, and M. I Jordan. “Minmax optimization: Stable limit points of gradient descent ascent are locally optimal,” *arXiv preprint arXiv:1902.00618*, 2019.
- K. Zhang, Z. Yang, and T. Başar. “Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games,” *arXiv preprint arXiv:1906.00729*, 2019.
- H. He, J. Boyd-Graber, K. Kwok, and H. Daumé III. “Opponent modeling in deep reinforcement learning,” *In International Conference on Machine Learning*, pages 1804–1813, 2016.
- M. Lanctot, V. Zambaldi, A. Gruslys, A. Lazaridou, K. Tuyls, J. Perolat, D. Silver, and T Graepel. 2017. “A unified game theoretic approach to multiagent reinforcement learning.” In *Advances in Neural Information Processing Systems* pages 1603–1610, 2018.

Thank You!

Related Work

- Related work (by algorithms):
  - Theoretical results:
    - Cooperative: independent Q learning (IQL) [Tan ’93], decentralized Q [Kar et al., ’13], decentralized actor-critic [Zhang et al., ’18,’19]
    - Non-cooperative: Minimax-Q [Littman ’94], Nash-Q [Hu et al. ’00], Friend-Foe-Q [Littman ’01], actor-critic for general-sum SG [Perolat et al. ’16], policy gradient (PG)-based learning [Mazumdar et al., ’18 ’19; Jin et al., ’19; Zhang et al., ’19]
    - Mixed: factorized LSPI [Lagoudakis & Parr ’03], decentralized fitted Q-iteration [Zhang et al., ’18]
  - Empirical results (with deep NN):
    - Cooperative: deep IQL [Tampuu et al., ’15] and variants [Foerster et al., ’16,’17], learning to communicate [Foerster et al., ’16], variants for partially-observed (PO) setting [Gupta et al., ’17]
    - Non-cooperative: opponent modeling +DQN [He et al., ’16], Learning with opponent-learning awareness (LOLA) [Foerster et al., ’18], PG-based robust adversarial RL (RARL) [Pinto et al. ’17],
    - Mixed: centralized critic + decentralized actor [Lowe et al., ’17], a unified game-theoretic framework for PO setting [Lanctot et al., ’18]

Other Open Problems in MARL

- Theory for MARL with function approximation
  - Why FA: state-action spaces grow exponentially with # of agents
  - Naïve policy gradient fails to converge [Mazumdar et al. ’19, Jin et al. ’19]
- Partially observable setting: decentralized POMDP (Dec-POMDP)
  - Much more practical
  - But most existing algorithms are either not scalable or lack of theory
- Safe/Robust MARL
  - Robots, UVAs, self-driving fleets are safety-critical systems
  - Incorporate current safe RL results into the multi-agent setting
- Model-based MARL