

深層強化学習入門 －エントロピ正則強化学習－

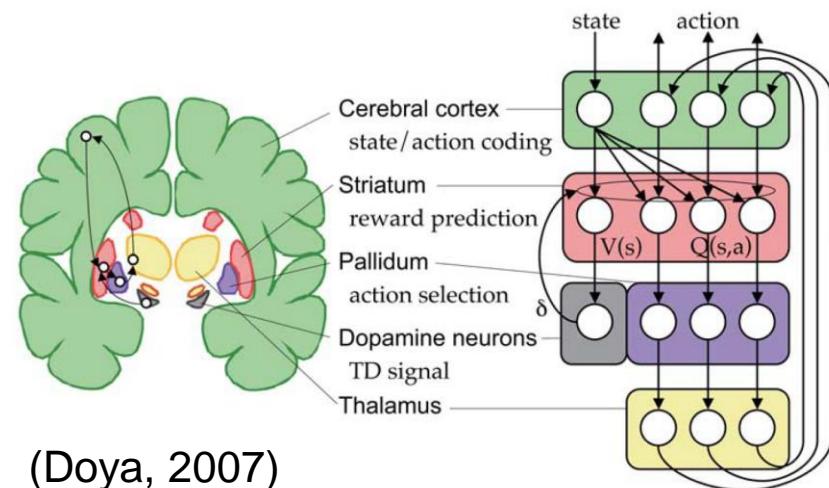
内部英治

国際電気通信基礎技術研究所

脳情報研究所 ブレインロボットインターフェース研究室

強化学習とは

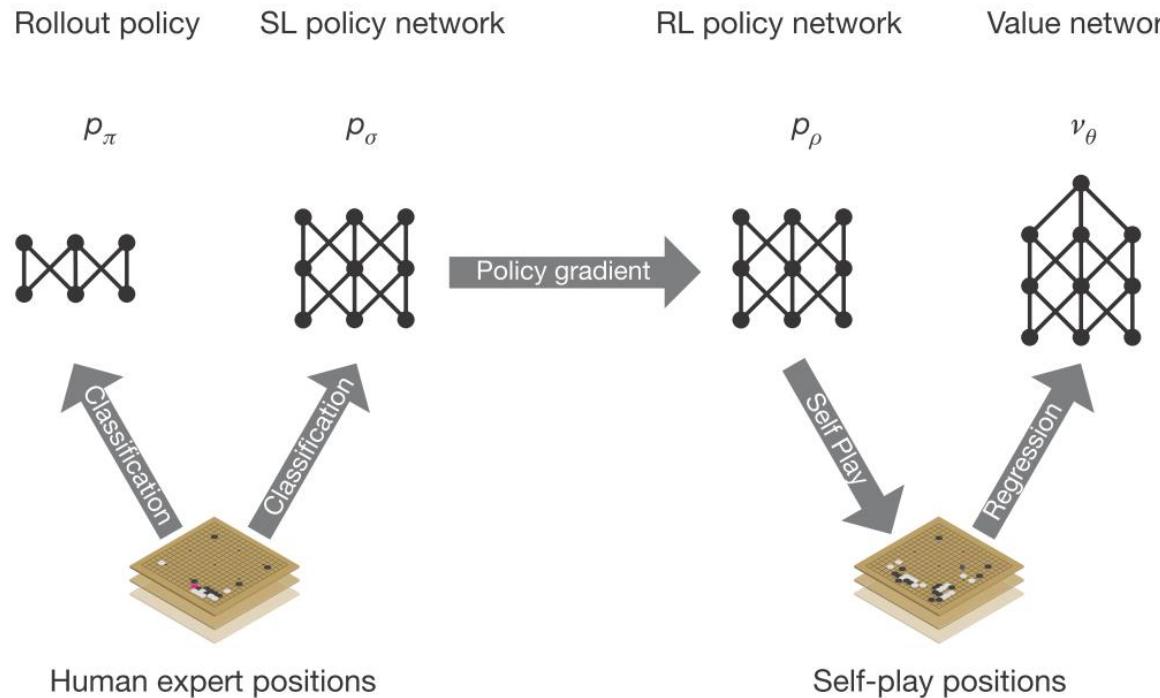
- 試行錯誤を通して方策（行動ルール）を学ぶ人工知能技術
- 囲碁のチャンピオンに勝利したアルファ碁は強化学習とディープラーニングの組み合わせ
→ ロボットなどの制御へ応用
- ヒトや動物の意思決定のモデルとしても注目
→ 脳科学の観点からの説明



AlphaGo

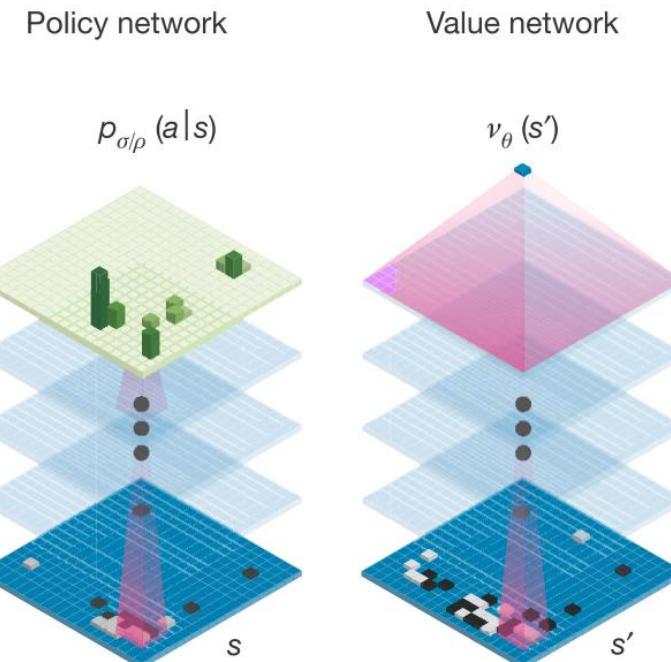
- 棋譜データから方策を学習したあと
強化学習

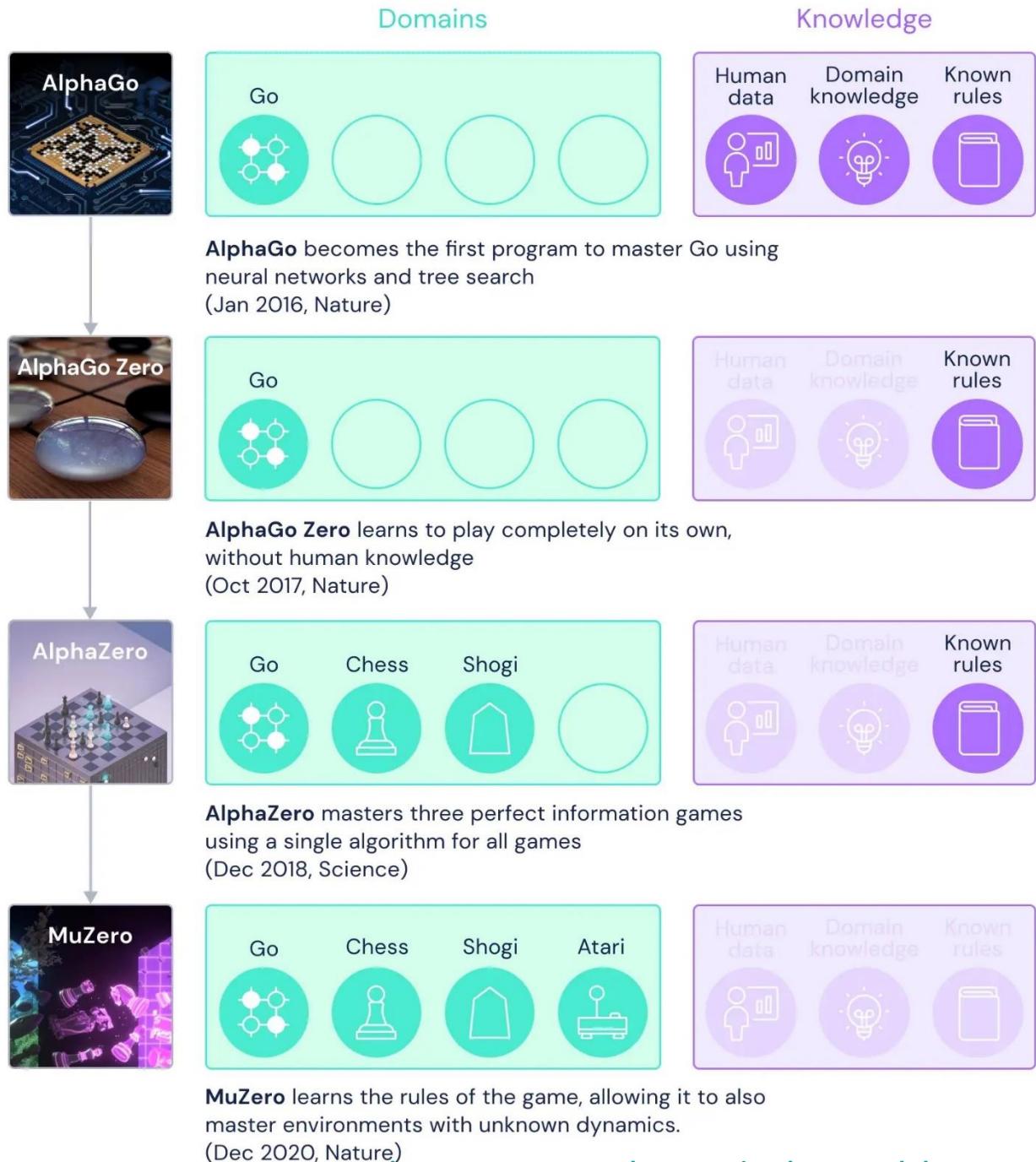
a



Go ratings				
Rank	Name	♂ ♀	Flag	Elo
1	Ke Jie	♂		3615
2	Google AlphaGo			3585
3	Park Jungwhan	♂		3569
4	Iyama Yuta	♂		3532
5	Lee Sedol	♂		3519

b





アルファシリーズの歴史

- AlphaGo Zeroは棋譜データなし、強化学習だけで実現
- AlphaZeroは碁だけでなく、チェスや将棋にも適用
- MuZeroはモデル（ルール）も学習

InstructGPT (ChatGPTの前身)

- 大規模モデルのファインチューニングに報酬学習 + 強化学習

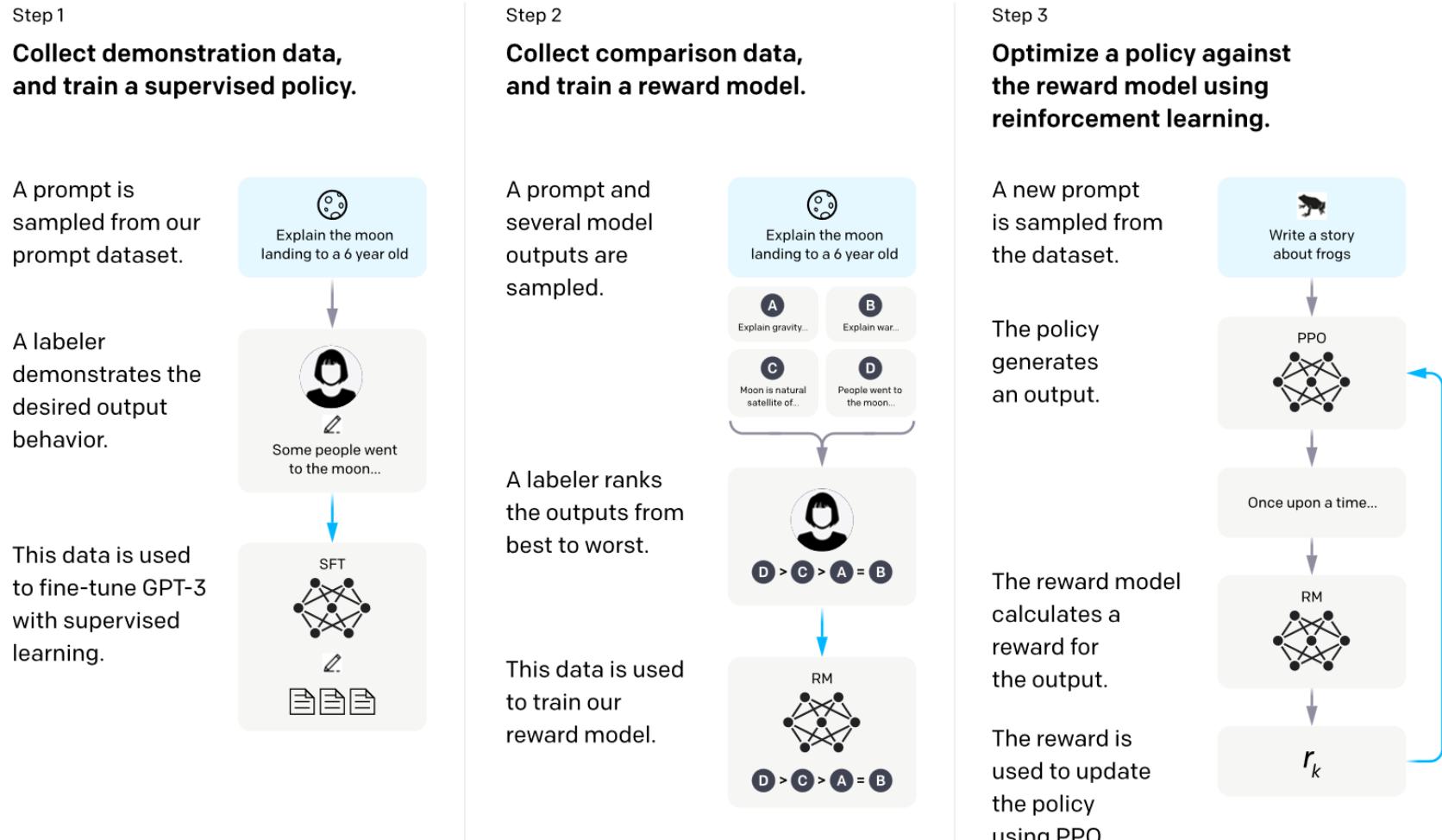
- Step2でランク付け

データから

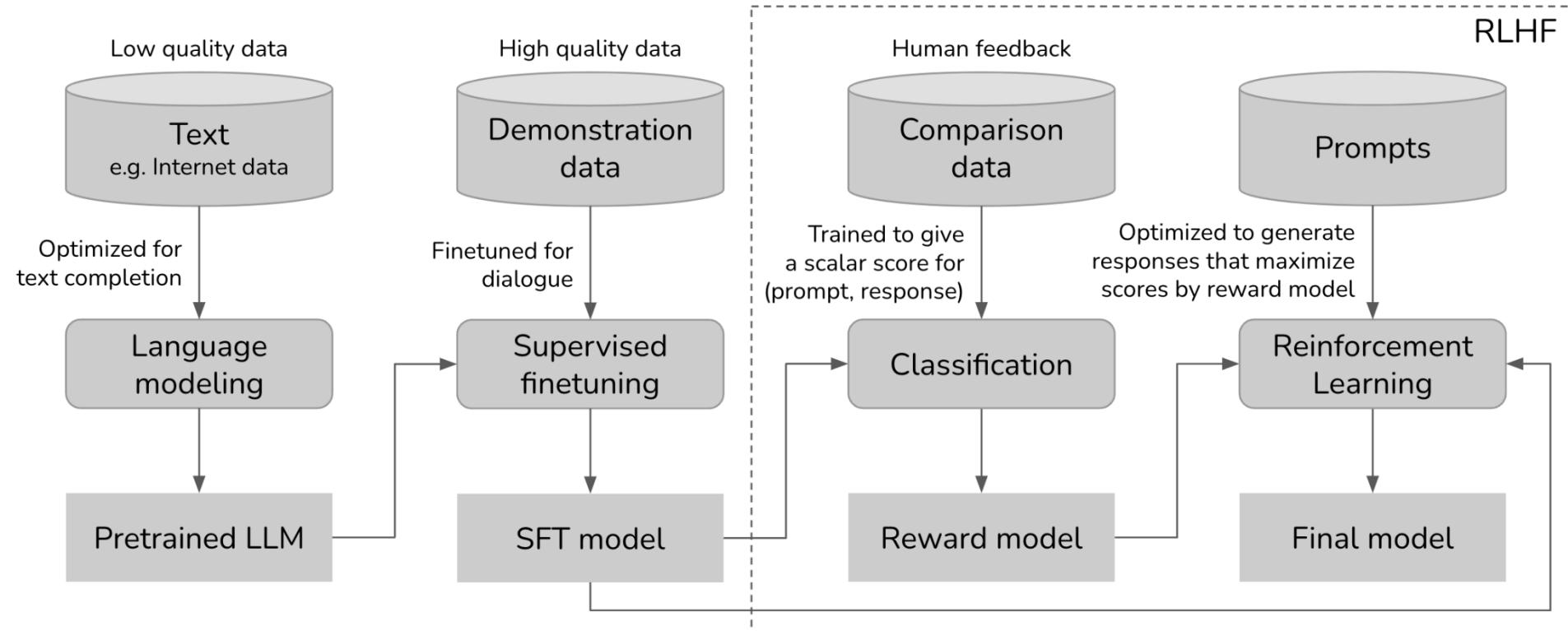
報酬学習

- Step3でエントロピ[°]
正則強化学習

Proximal Policy
Optimization (PPO)
を使用



RLHF: Reinforcement Learning from Human Feedback



Scale
May '23

>1 trillion
tokens

10K - 100K
(prompt, response)

100K - 1M comparisons
(prompt, winning_response, losing_response)

10K - 100K
prompts

Examples
Bolded: open
sourced

GPT-x, Gopher, **Falcon**,
LLaMa, **Pythia**, Bloom,
StableLM

Dolly-v2, **Falcon-Instruct**

InstructGPT, ChatGPT,
Claude, **StableVicuna**

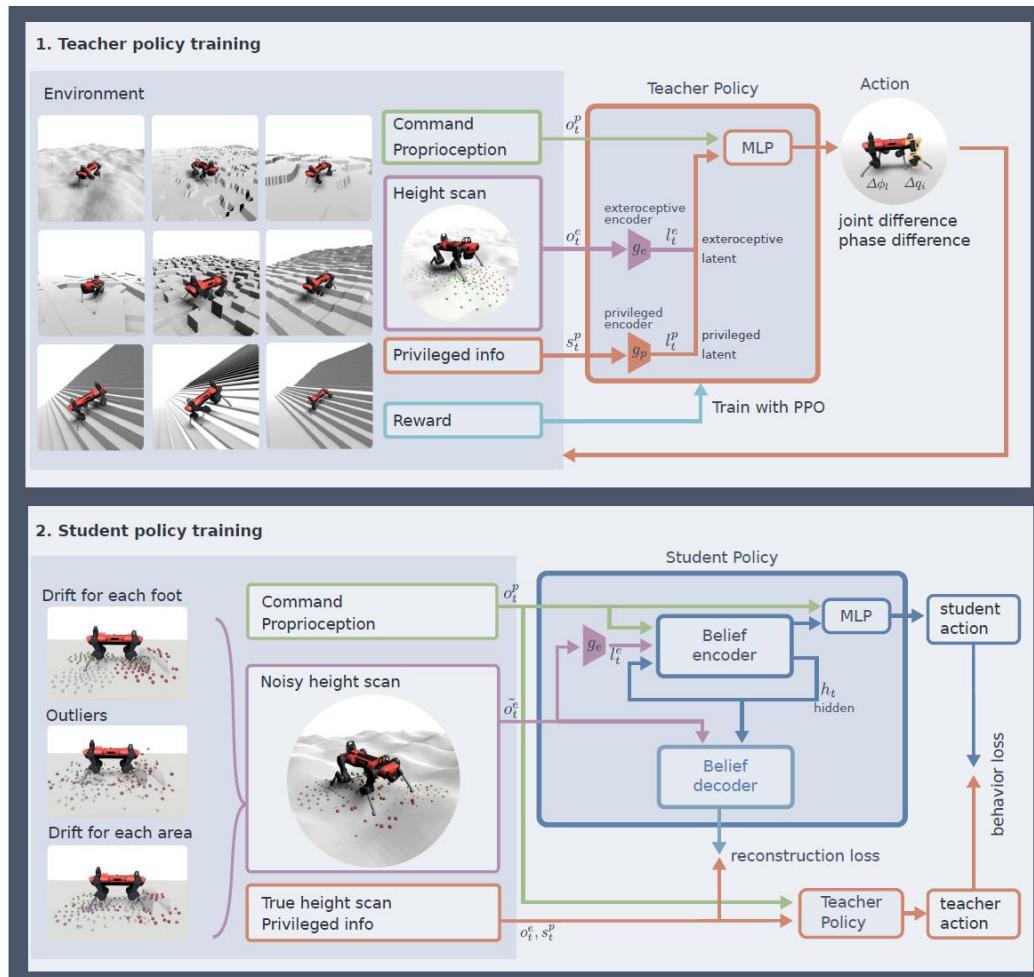
四脚ロボットの不整地走行



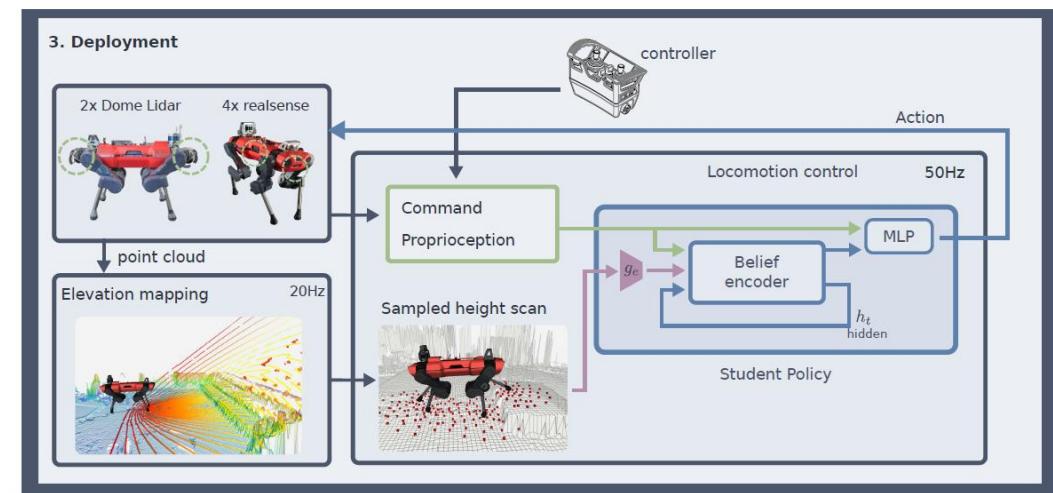
Learning robust perceptive locomotion
for quadrupedal robots in the wild

T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. (2022). [Learning robust perceptive locomotion for quadrupedal robots in the wild](#). *Sci. Robot.*, vol. 7, no. 62, p. eabk2822.
<https://leggedrobotics.github.io/rl-perceptiveloco/>

四脚ロボットの不整地走行



- 教師方策は特権情報をもとに PPOで方策を学習
- 生徒方策は実際に利用可能なセンサ情報だけを使って教師を模倣



1対1の大型ロボットによるサッカー

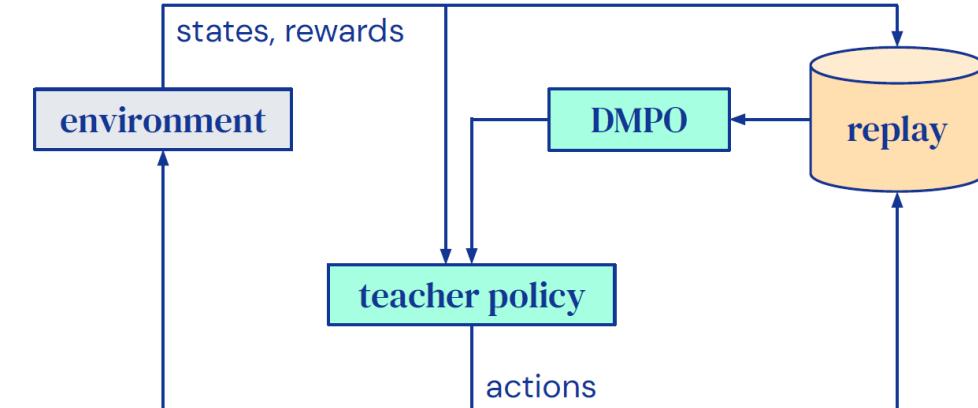


T. Haarnoja et al. (2023). [Learning Agile Soccer Skills for a Bipedal Robot with Deep Reinforcement Learning](#). arXiv.
<https://www.youtube.com/watch?v=chMwFy6kXhs>

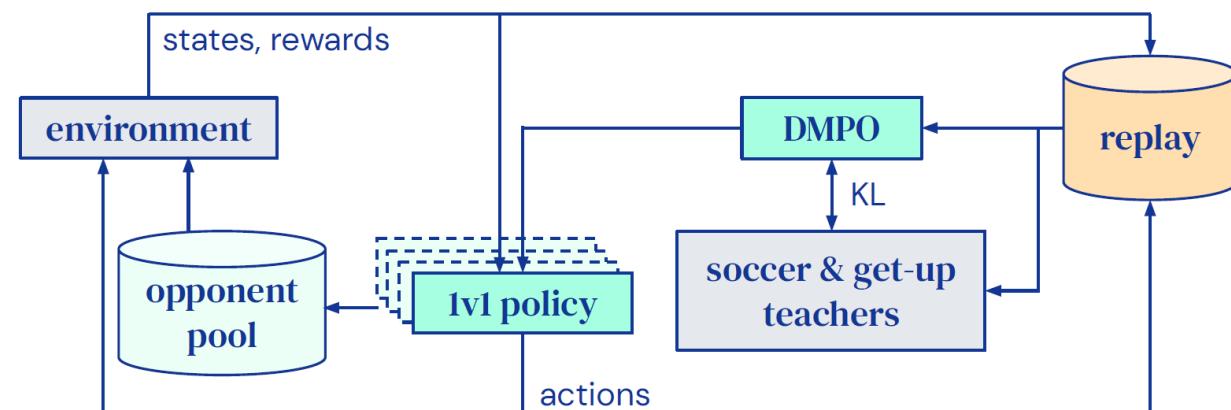
1対1の人性ロボットによるサッカー

- 第1段階では訓練されていない敵を相手にシュートする行動と転倒状態からの立ち上がり動作を学習
 - 方策評価に分布型クリティックを使った Distributional MPO (DMPO)
- 第1段階で学習した方策から逸脱しないようにKL正則を追加したDMPO

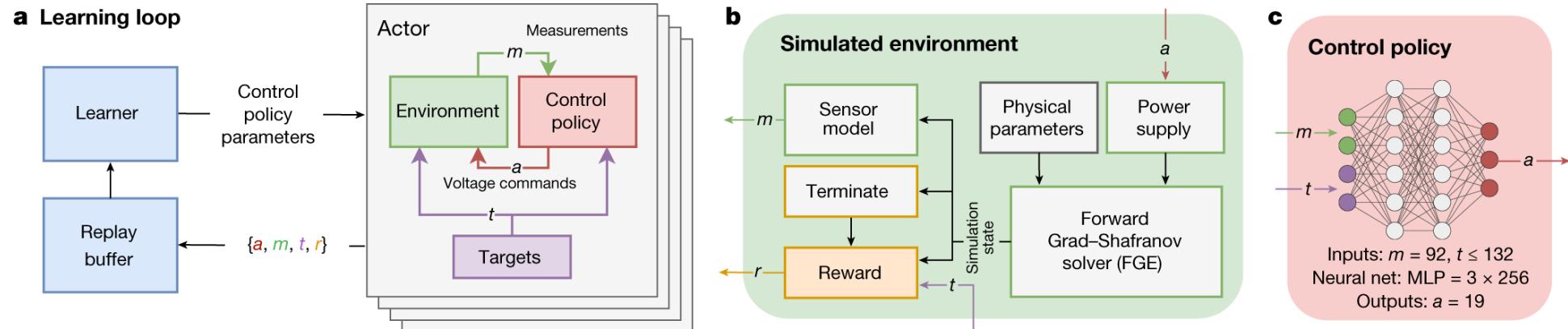
Stage 1: train teacher policies for soccer and get-up



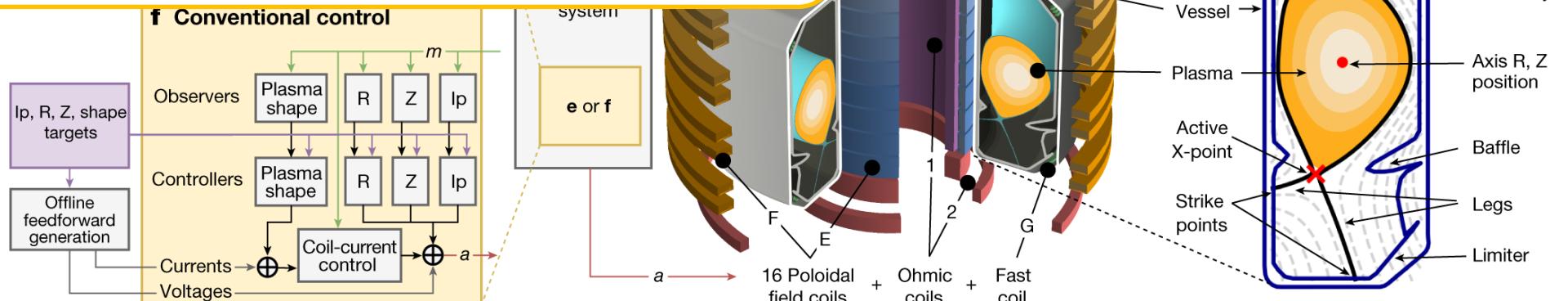
Stage 2: train the 1v1 policy with regularization to teachers and self-play



トカマク型核融合炉のプラズマ制御



使用されているアルゴリズム Maximum a posteriori Policy Optimization (MPO)は
エントロピ正則強化学習



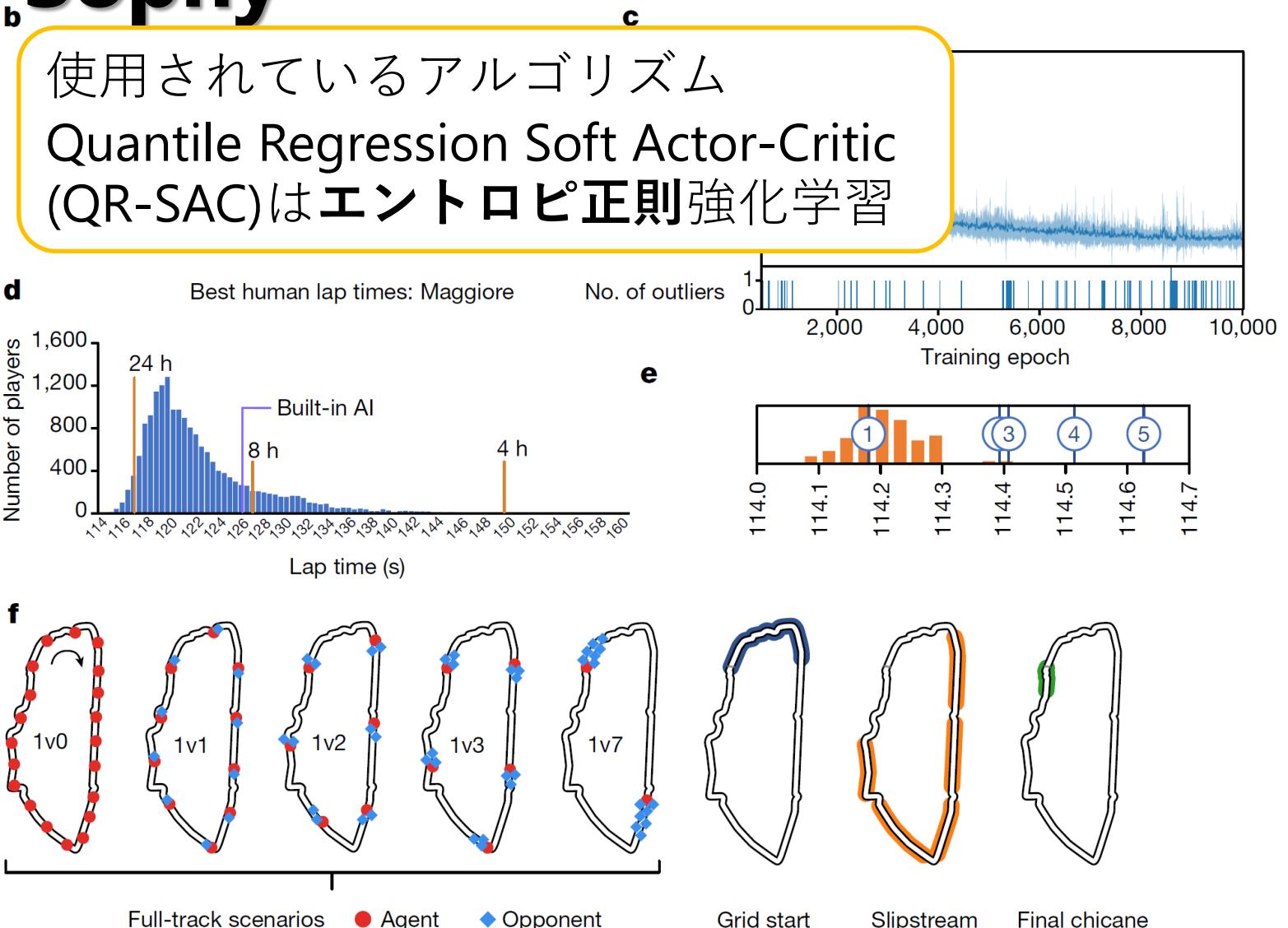
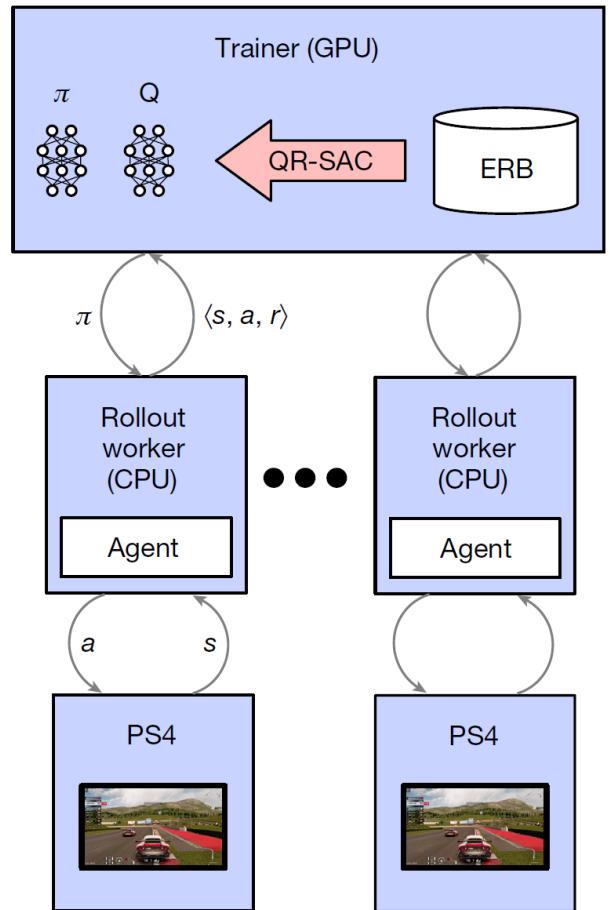
Gran Turismo Sophy

- エキスパートプレイヤーに勝利



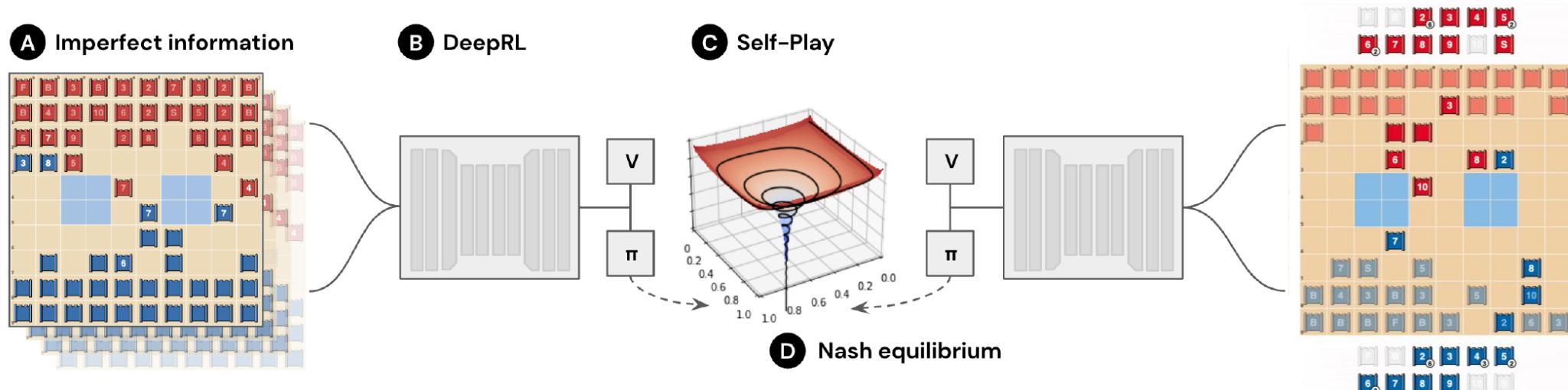
Captured on PS4. Game progression required to access most vehicles.

Gran Turismo Sophy



DeepNash: 不完全戦略ゲームStratego (ヨーロッパの軍人将棋) に応用

- モデルフリー深層強化学習+自己対戦+ナッシュ均衡
 - AlphaGoなどでは考慮されなかったゲーム理論を導入することで、自己対戦の効率を大幅に改善

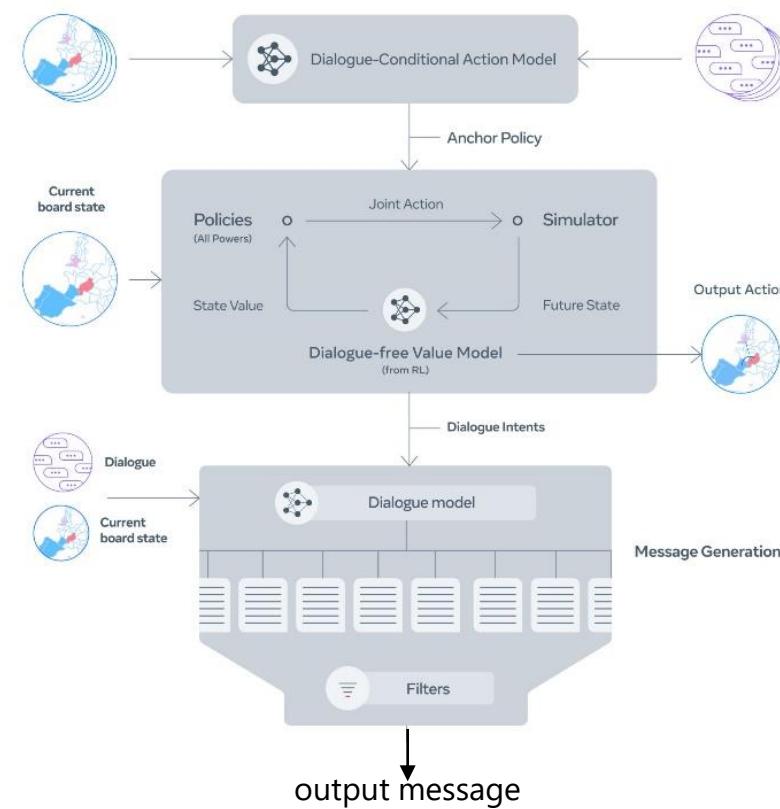
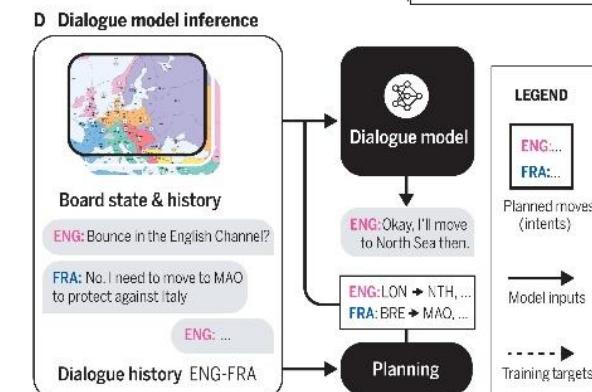
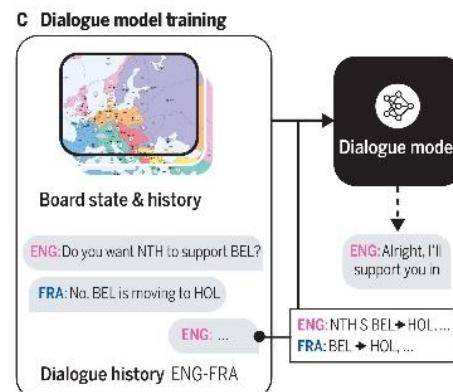
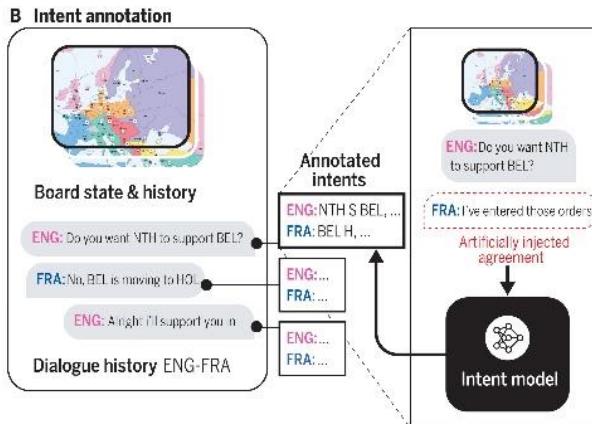
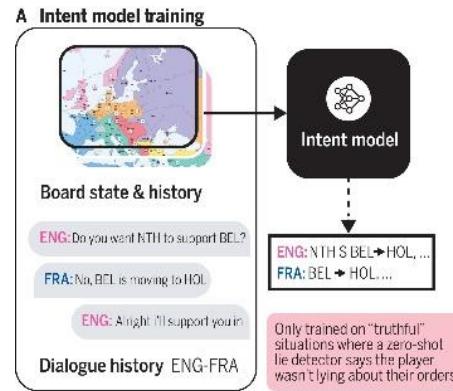


Replicator dynamics: $\frac{d}{d\tau} \pi_\tau^i(a^i) = \pi_\tau^i(a^i) [Q_{\pi_\tau}^i(a^i) - \sum_b \pi_\tau^i(b) Q_{\pi_\tau}^i(b)]$

Reward transformation: $r^i(\pi^i, \pi^{-i}, a^i, a^{-i}) = r^i(a^i, a^{-i}) - \eta \log \left(\frac{\pi^i(a^i)}{\pi_{\text{reg}}^i(a^i)} \right) + \eta \log \left(\frac{\pi^{-i}(a^{-i})}{\pi_{\text{reg}}^{-i}(a^{-i})} \right)$

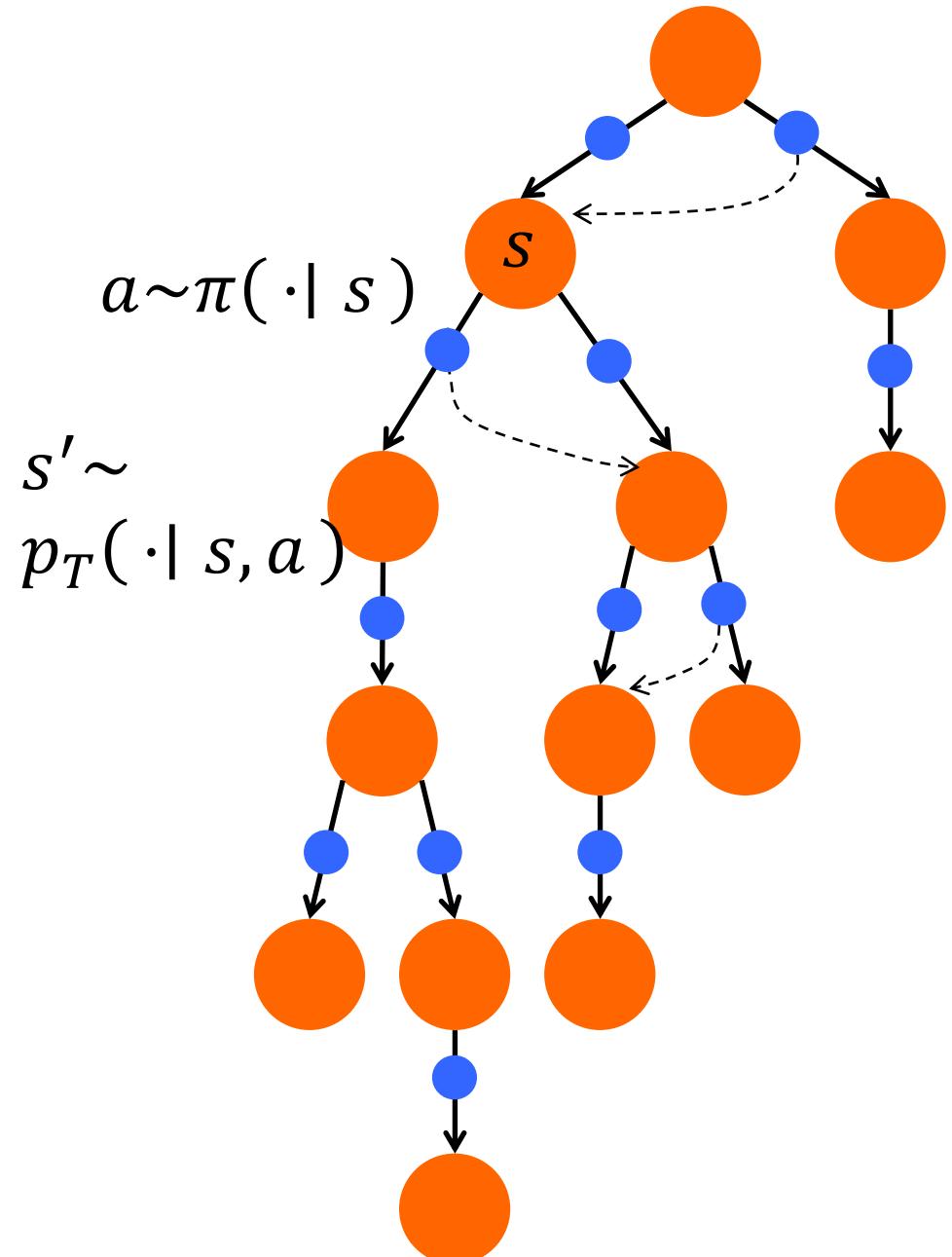
Cicero: 不完全戦略ゲームDiplomacyに応用

- 第1次世界大戦時のヨーロッパを舞台にした7人制のボードゲーム
- Ciceroは人間プレイヤーと対戦し上位10%の成績を収めた



記号について

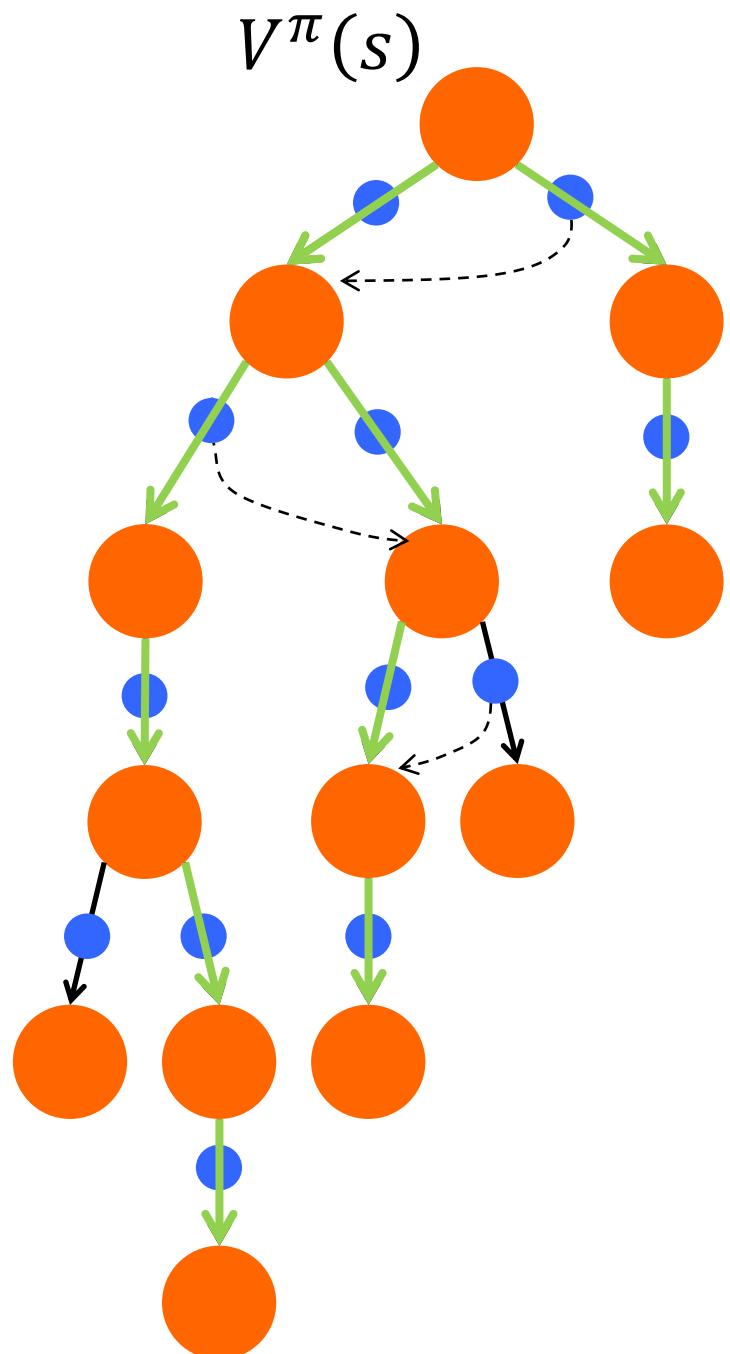
s, \mathcal{S}	状態および状態集合
a, \mathcal{A}	行動および行動集合
$p_T(s' s, a)$	状態 s で行動 a を実行したとき 状態 s' に遷移する確率
$\pi(a s)$	状態 s で行動 a を実行する確率 (方策)
$r(s, a)$	状態 s で行動 a を実行したとき の評価値(報酬)



状態価値関数

- 状態 s の「価値」
- 状態 s からスタートし、以降は方策 π に従って行動したときに得られる総報酬

$$G_t = \sum_{t=0}^{\infty} \gamma^k r(s_t, a_t) \quad \gamma \in [0, 1]: \text{割引率}$$



状態価値関数

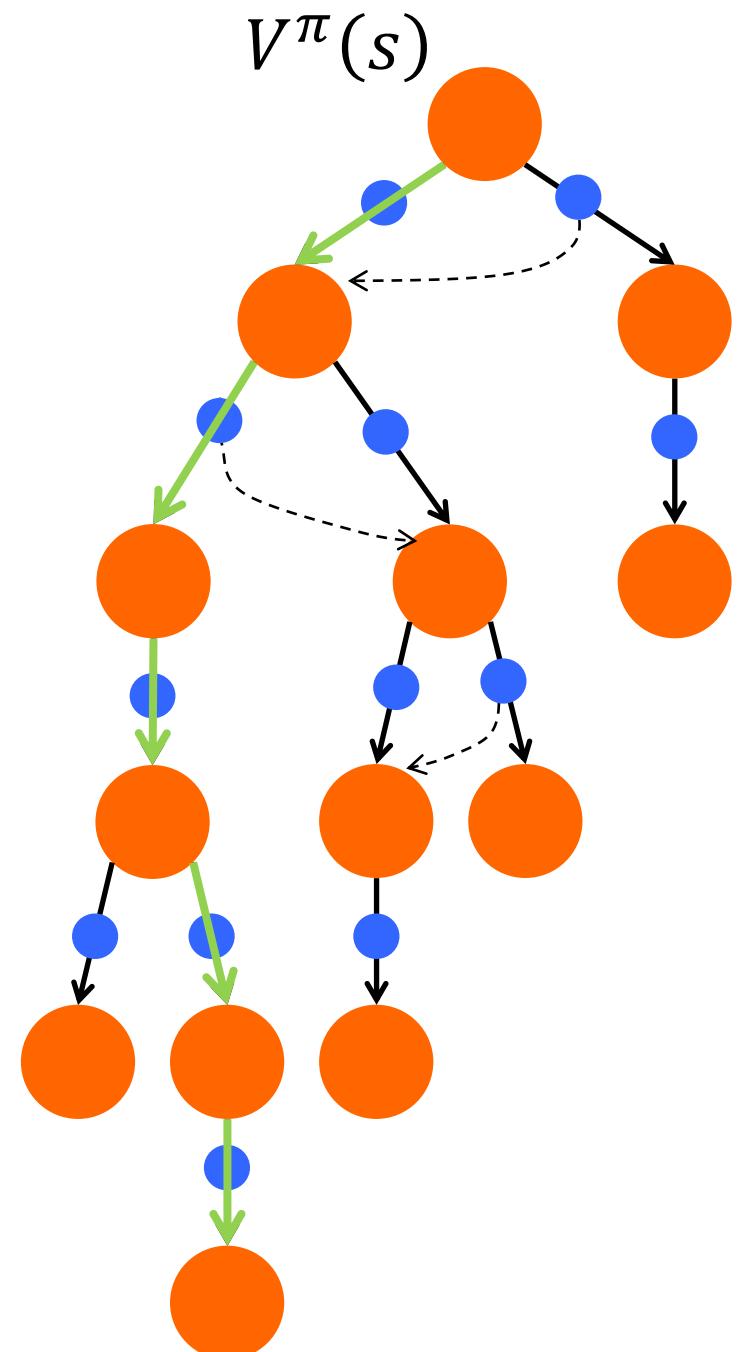
- 状態 s の「価値」
- 状態 s からスタートし、以降は方策 π に従って行動したときに得られる総報酬

$$G_t = \sum_{t=0}^{\infty} \gamma^k r(s_t, a_t) \quad \gamma \in [0, 1]: \text{割引率}$$

- 状態価値関数：総報酬の期待値

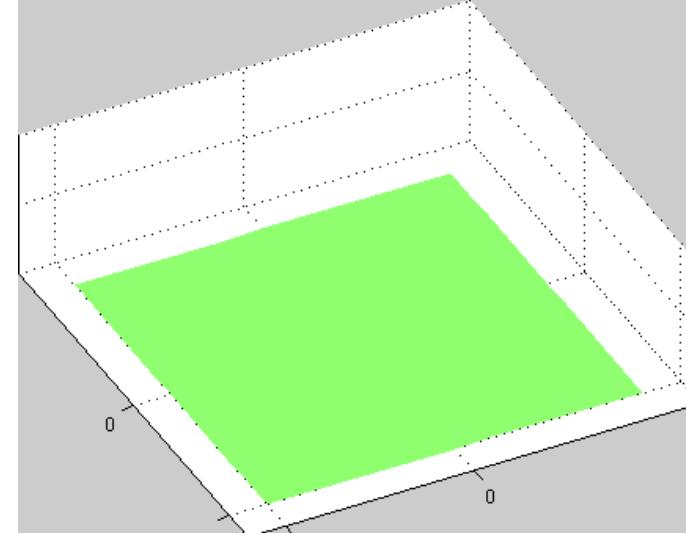
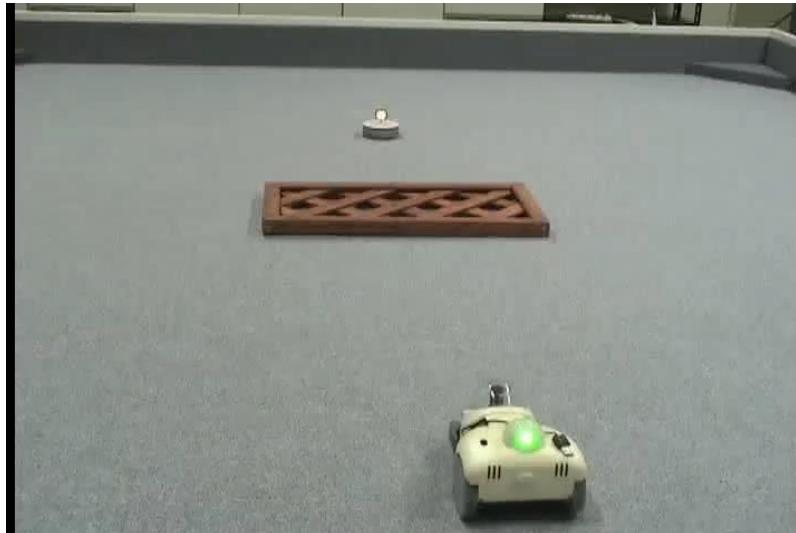
$$V^\pi(s) = \mathbb{E}_{p_T, \pi}[G_t \mid s_t = s]$$

$$\begin{aligned} p(a_t, s_{t+1}, a_{t+1}, s_{t+2}, \dots \mid s_t) \\ = \prod_{k=0}^{\infty} \pi(a_{t+k} \mid s_{t+k}) p_T(s_{t+k+1} \mid s_{t+k}, a_{t+k}) \end{aligned}$$



状態価値関数のイメージ

- 障害物との衝突を避けながら、電池パックを捕獲する課題
 - 電池パックを捕獲したときに大きな正の報酬
 - 障害物と衝突したときに大きな負の報酬
 - 停止以外の行動を選択したとき小さな負の報酬

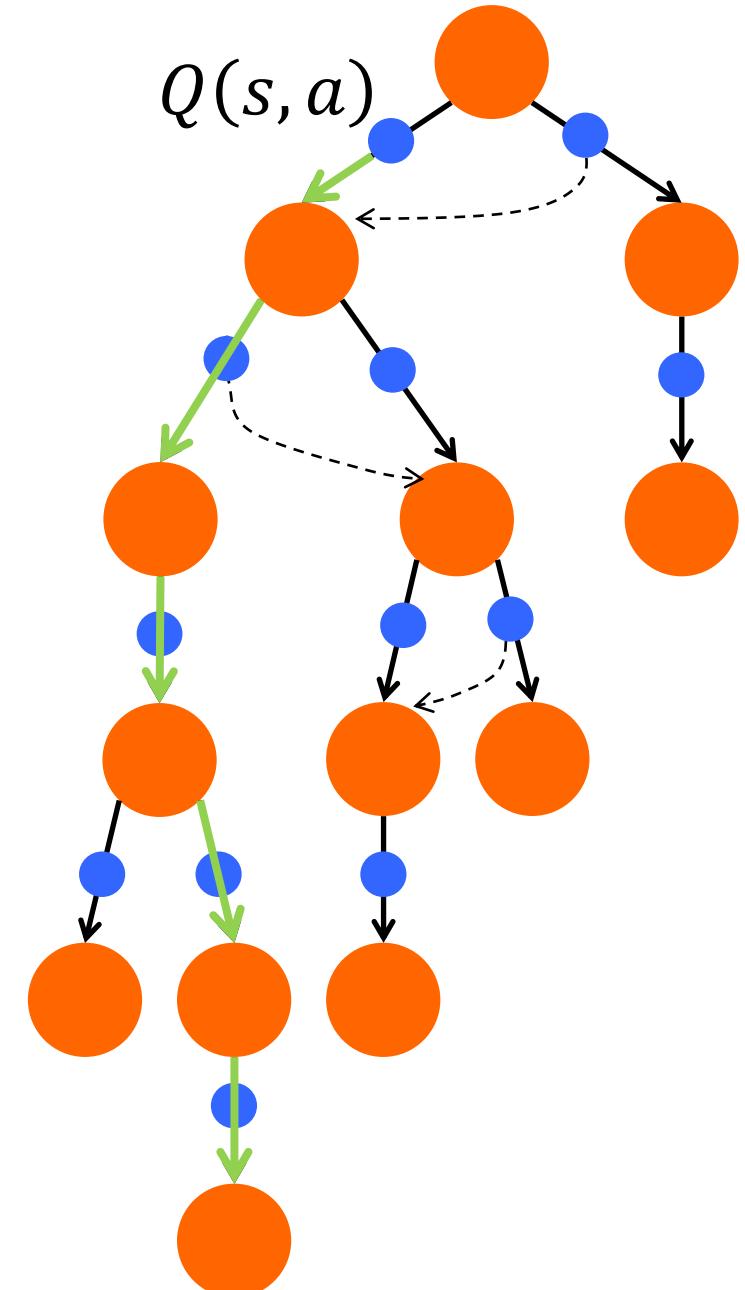


状態行動価値関数

- 状態 s と行動 a の組の「価値」
- 状態行動価値関数:
状態 s からスタートし、以降は方策 π に従って行動したときに得られる総報酬の期待値

$$Q^\pi(s, a) = \mathbb{E}_{\pi, p_T}[G_t \mid s_t = s, a_t = a]$$

$$\begin{aligned} & p(s_{t+1}, a_{t+1}, s_{t+2}, \dots \mid s_t, a_t) \\ &= \prod_{k=0}^{\infty} p_T(s_{t+k+1} \mid s_{t+k}, a_{t+k}) \pi(a_{t+k+1} \mid s_{t+k+1}) \end{aligned}$$



V^π と Q^π の関係

- $$V^\pi(s) = \sum_a \pi(a | s) Q^\pi(s, a)$$

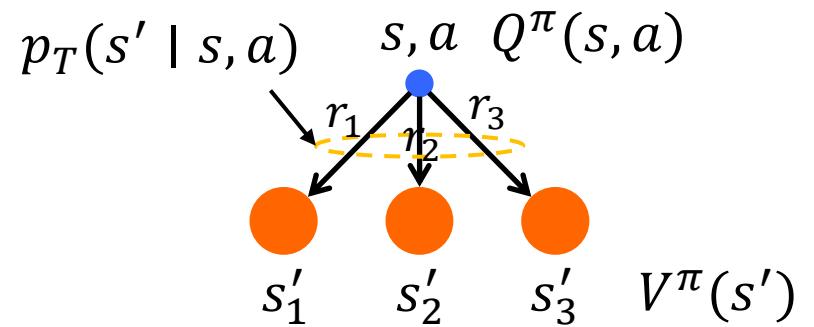
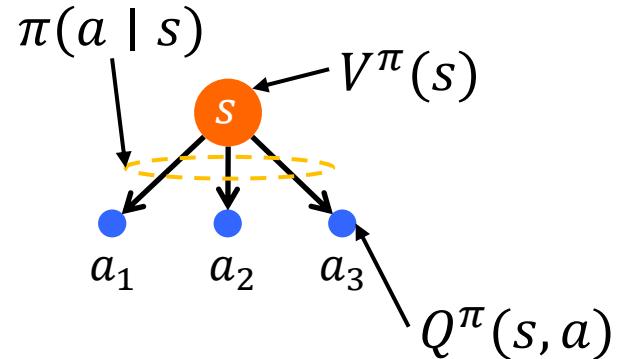
$$= \mathbb{E}_\pi[Q^\pi(s, a)]$$

- $$Q^\pi(s, a) = \sum_{s'} p_T(s' | s, a) [r + \gamma V^\pi(s')]$$

- アドバンテージ関数

$$A^\pi(s, a) \triangleq Q^\pi(s, a) - V^\pi(s)$$

- 定義より $\mathbb{E}_\pi[A^\pi(s, a)] = 0$



Bellman期待方程式

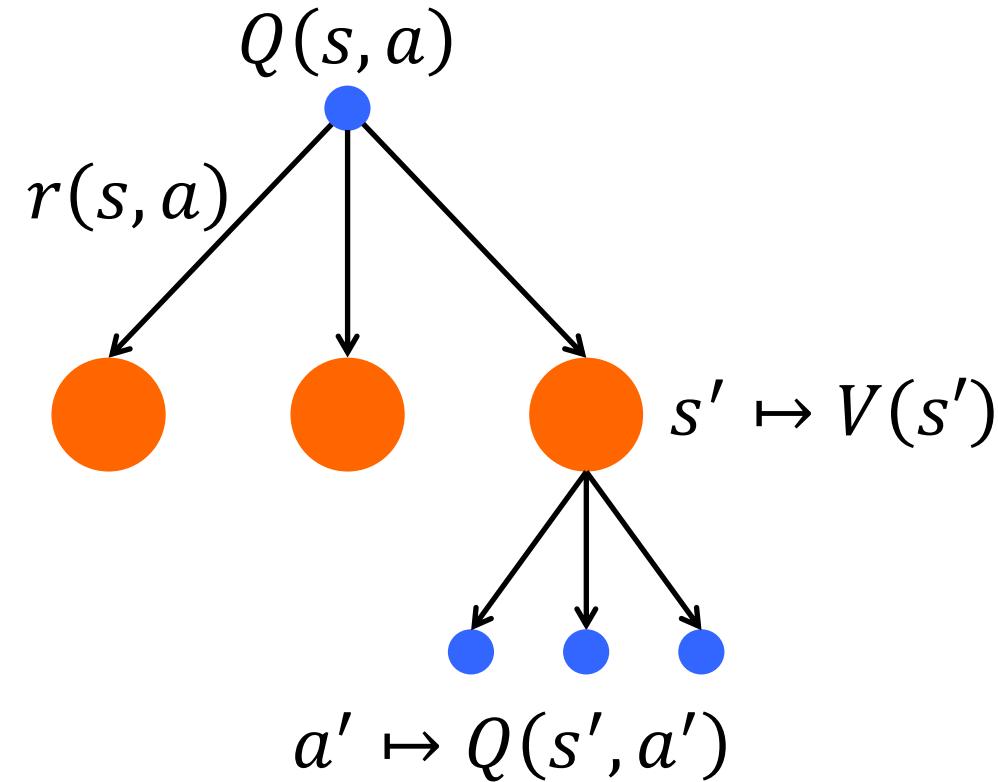
- Q^π についての再帰式
- 任意の方策 π と状態・行動の組に対して

$$Q^\pi(s, a) = \mathbb{E}_\pi[G_t \mid s_t = s, a_t = a]$$

$$= r(s, a) + \gamma \mathbb{E}_{p_T}[V(s')]$$

$$= r(s, a) + \gamma \mathbb{E}_{p_T}[\mathbb{E}_\pi[Q^\pi(s', a')]]$$

$$= r(s, a) + \gamma \sum_{s'} p_T(s' \mid s, a) \sum_{a'} \pi(a' \mid s') Q^\pi(s', a')$$



Bellman期待作用素

- Bellman方程式の右辺をもとに、関数 $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ を変換する作用素 \mathcal{T}_π を定義

$$\mathcal{T}_\pi Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') \right]$$

- 解の一意性
Bellman期待作用素の不動点は唯一であり、それは状態行動価値関数



Q^π は Bellman 方程式 $Q(s, a) = \mathcal{T}_\pi Q(s, a)$ を満たす唯一の関数

Bellman期待作用素（モデルフリーの場合）

- \mathcal{T}_π は状態遷移確率（モデル）と報酬が既知の場合に定義可能

$$\mathcal{T}_\pi Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') \right]$$

- モデルが未知（モデルフリー）で報酬も観測された値の場合
観測したサンプル (s, a, r, s', a') を使った近似作用素を用いる

$$\hat{\mathcal{T}}_\pi Q(s, a) = r + \gamma Q(s', a')$$

最適状態行動価値関数

- 方策 π によって状態行動価値関数 Q^π は異なる
→ 価値が最大のものを最適とする
- 数学的な定義

$$Q^*(s, a) = \max_{\pi} Q^\pi(s, a)$$

最適状態行動価値関数と最適方策

- 方策 π によって状態行動価値関数 Q^π は異なる

➡ 価値が最大のものを最適とする

- 数学的な定義

$$Q^*(s, a) = \max_{\pi} Q^{\pi}(s, a)$$

- 最適行動は「最大の価値を持つ」

$$\pi^*(a | s) = \begin{cases} 1 & \text{if } a = \arg \max_{a \in \mathcal{A}} Q^*(s, a) \\ 0 & \text{otherwise} \end{cases}$$

➡ $\pi^* \in \arg \max_{\pi} \sum_a \pi(a | s) Q^*(s, a)$

Q は唯一でも π^* は複数存在する可能性がある

Bellman最適作用素

- \mathcal{T}_π の右辺の Q についての期待値を最適なものに置き換え

$$\mathcal{T}_\pi Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') \right]$$



$$\mathcal{T}_* Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi^*(a' | s') Q(s', a') \right]$$

Bellman最適作用素

- \mathcal{T}_π の右辺の Q についての期待値を最適なものに置き換える

$$\mathcal{T}_\pi Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') \right]$$



$$\mathcal{T}_* Q(s, a) = r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\max_{a'} Q(s', a') \right]$$

- Bellman最適作用素の不動点は唯一であり、それは最適状態行動価値関数
- Q^* は Bellman 最適方程式 $Q(s, a) = \mathcal{T}_* Q(s, a)$ を満たす唯一の関数

方策反復（方策評価と方策改善）

- 方策評価

与えられた方策 π_k に対して、状態行動価値 Q_k を求める

while $|Q_{k+1} - Q_k| > \text{threshold}$

$$Q_{k+1} = \mathcal{T}_{\pi_k} Q_k$$

- 方策改善

$$\pi_{k+1} = \arg \max_{\pi} \sum_a \pi(a | s) Q_k(s, a)$$

複数の方策が最大となる場合
でもどれか一つを選ぶ

- 繰り返し計算によって Q^*, π^* が求められる

価値反復

- 方策評価と方策改善を同時に実施

状態行動価値 Q_k を求める

while $|Q_{k+1} - Q_k| > \text{threshold}$

$$Q_{k+1} = \mathcal{T}_* Q_k$$

Q_k が Q^* の良い近似になるまで繰り返す

- 最適方策の計算

$$\pi^* = \arg \max_{\pi} \sum_a \pi(a | s) Q^*(s, a)$$

- 繰り返し計算によって Q^* を求めた後で π^* を計算

近似修正方策反復法 (Approximate Modified Policy Iteration)

- 最適方策と対応する価値関数を推定するモデルベース法

$$\left[\begin{array}{l} \pi_{k+1} \in \mathcal{G}(Q_k) \triangleq \arg \max_{\pi} \sum_a \pi(a | s) Q_k(s, a) \\ Q_{k+1} = (\mathcal{T}_{\pi_{k+1}})^m Q_k + \epsilon_{k+1} \end{array} \right]$$

– ϵ はBellman作用素を適用したときに生じる誤差

- $m = 1$: 近似価値反復法 (Approximate Value Iteration)
- $m = \infty$: 近似方策反復法 (Approximate Policy Iteration)

M. L. Puterman et al. (1978). [Modified Policy Iteration Algorithms for Discounted Markov Decision Problems](#). Management Science. 24(11): 1127-1137.

B. Scherrer et al. (2015). [Approximate Modified Policy Iteration and its Application to the Game of Tetris](#). Journal of Machine Learning Research. 16(49):1629–1676.

近似修正方策反復法 (Approximate Modified Policy Iteration)

- 最適方策と対応する価値関数を推定するモデルベース法

$$\left[\begin{array}{l} \pi_{k+1} \in \mathcal{G}(Q_k) \triangleq \arg \max_{\pi} \sum_a \pi(a | s) Q_k(s, a) \\ Q_{k+1} = (\mathcal{T}_{\pi_{k+1}})^m Q_k + \epsilon_{k+1} \end{array} \right]$$

– ϵ はBellman作用素を適用したときに生じる誤差

- 様々な強化学習アルゴリズムが方策評価と方策改善に
エントロピ正則化を導入することで導出される

方策改善ステップに正則化を導入

- エントロピとKullback-Leiblerダイバージェンスを追加

$$G_{\mu}^{\lambda, \tau}(Q) \triangleq \arg \max_{\pi} \sum_a \pi(a | s) Q(s, a) - \lambda \text{KL}(\pi \| \mu) + \tau \mathcal{H}(\pi)$$

- $\text{KL}(\pi \| \mu) = \sum_a \pi(a | s) \ln \frac{\pi(a | s)}{\mu(a | s)}$ λ, τ : ハイパーパラメータ

 ベースライン方策 μ から逸脱するのを防ぐ

- $\mathcal{H}(\pi) = - \sum_a \pi(a | s) \ln \pi(a | s)$

 最適方策が決定論的になるのを防ぐ

方策評価ステップに正則化を導入

- 方策改善ステップと同様に修正

$$\begin{aligned} \mathcal{T}_{\pi|\mu}^{\lambda,\tau}(Q) &\triangleq r(s, a) \\ &+ \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') - \lambda \text{KL}(\pi \parallel \mu) + \tau \mathcal{H}(\pi) \right] \end{aligned}$$

正則化を導入した近似修正方策反復法 (Approximate Modified Policy Iteration)

- 最適方策と対応する価値関数を推定するモデルベース法

$$\begin{cases} \pi_{k+1} = \mathcal{G}_{\pi_k}^{\lambda, \tau}(Q_k) \\ Q_{k+1} = \left(\mathcal{T}_{\pi_{k+1} | \pi_k}^{\lambda, \tau} \right)^m Q_k + \epsilon_{k+1} \end{cases}$$

$\lambda > 0$ または $\tau > 0$ であれば
方策は唯一に決定できる

- ベースライン μ として一つ前の方策 π_k を用いる

導出されるエントロピ正則強化学習

	エントロピのみ	KLのみ	エントロピとKL両方
正則ありの評価ステップ	Soft Q-learning (Fox et al., 2016; Haarnoja et al., 2017), Soft Actor-Critic (Haarnoja et al., 2018), Mellowmax (Asadi and Littman, 2017)	Dynamic Policy Programming (Azar et al., 2012), Speedy Q-learning (Azar et al., 2011).	Conservative Value Iteration (Kozuno et al., 2019), Advantage Learning (Baird, 1999; Bellemare et al., 2013)
正則なしの評価ステップ	Softmax DQN (Zhao et al., 2019)	Trust Region Policy Optimization (Schulman et al., 2015). Maximum a Posteriori Opt. (Abdolmaleki et al., 2018) Politex (Abbasi-Yadkori et al., 2019), Momentum Value Iteration (Vieillard et al., AISTAT 2020)	Softened LSPI (Pérolat et al., 2016) Momentum DQN (Vieillard et al., AISTAT 2020)

M. Geist et al. (2019). [A Theory of Regularized Markov Decision Processes](#). In Proc. of ICML, 2160-2169.

N. Vieillard et al. (2020). [Leverage the Average: an Analysis of KL Regularization in RL](#). NeurIPS 33, 12163-12174.

Soft Actor-Critic (SAC)の導出

- SACは $m = 1$, エントロピ正則あり, KL正則なしのアルゴリズム

$$\begin{aligned} \mathcal{T}_{\pi_{k+1}|-}^{0,\tau}(Q) &\triangleq r(s, a) \\ &+ \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi_{k+1}(a' | s') (Q(s', a') - \tau \ln \pi_{k+1}(a' | s')) \right] \end{aligned}$$

- ソフトBellmanバックアップオペレータ(Haarnoja et al., 2018)そのもの

Soft Actor-Critic (SAC) の導出

- SACはエントロピあり、KLなしのアルゴリズム

$$\begin{aligned} \mathcal{G}_-^{0,\tau}(Q) &\triangleq \arg \max_{\pi} \sum_a \pi(a | s)(Q(s, a) - \tau \ln \pi(a | s)) \\ &= \frac{\exp(\tau^{-1} Q(s, a))}{\sum_a \exp(\tau^{-1} Q(s, a))} \triangleq \pi_{k+1}^*(a | s) \end{aligned}$$

- Lagrangeの未定乗数法を使って求める

正則付き方策改善の計算

- $\arg \max_{\pi} \sum_a \mathbb{E}_{\pi}[Q] - \lambda \text{KL}(\pi \parallel \mu) + \tau \mathcal{H}(\pi)$ を計算
- $\lambda \text{KL}(\pi \parallel \mu) - \tau \mathcal{H}(\pi)$ が π に関して凸
- 結果

$$\max_{\pi} \sum_a \mathbb{E}_{\pi}[Q] - \lambda \text{KL}(\pi \parallel \mu) + \tau \mathcal{H}(\pi) = (\lambda + \tau) \left(\ln \sum_a \mu(a \mid s)^{\frac{\lambda}{\lambda + \tau}} \exp \frac{q(s, a)}{\lambda + \tau} \right)$$

$$\arg \max_{\pi} \sum_a \mathbb{E}_{\pi}[Q] - \lambda \text{KL}(\pi \parallel \mu) + \tau \mathcal{H}(\pi) = \frac{\mu(a \mid s)^{\frac{\lambda}{\lambda + \tau}} \exp \frac{q(s, a)}{\lambda + \tau}}{\sum_{a'} \mu(a' \mid s)^{\frac{\lambda}{\lambda + \tau}} \exp \frac{q(s, a')}{\lambda + \tau}}$$

Soft Actor-Critic (SAC) の導出

- SACはエントロピあり、KLなしのアルゴリズム

$$\begin{aligned} \mathcal{G}_-^{0,\tau}(Q) &\triangleq \arg \max_{\pi} \sum_a \pi(a | s)(Q(s, a) - \tau \ln \pi(a | s)) \\ &= \frac{\exp(\tau^{-1} Q(s, a))}{\sum_a \exp(\tau^{-1} Q(s, a))} \triangleq \pi_{k+1}^*(a | s) \end{aligned}$$

– Lagrangeの未定乗数法を使って求める

- SACは行動が連続の場合に用いることが多いので、サンプルしやすいパラメトリックな分布で近似

$$\pi_{k+1}(a | s) = \arg \min_{\pi} \mathbb{E}_s[\text{KL}(\pi \| \pi_{k+1}^*)]$$

実際のSAC

- Q関数を学習する損失関数

- 価値関数を二つ学習

$$L(Q_j, \mathcal{D}) = \mathbb{E}_{(s,a,r,s',d) \sim \mathcal{D}} \left[\left(Q_j(s, a) - y(r, s', d) \right)^2 \right] \quad j = 1, 2$$

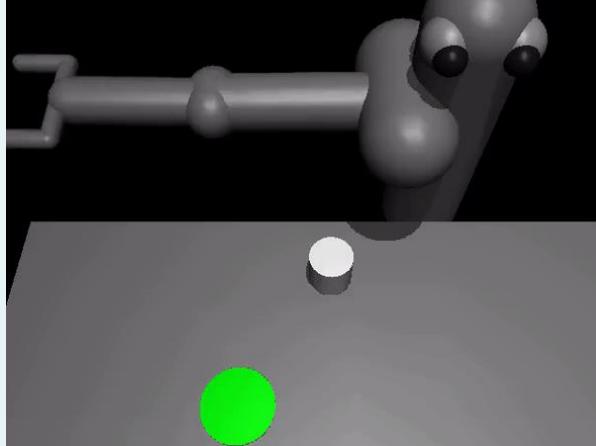
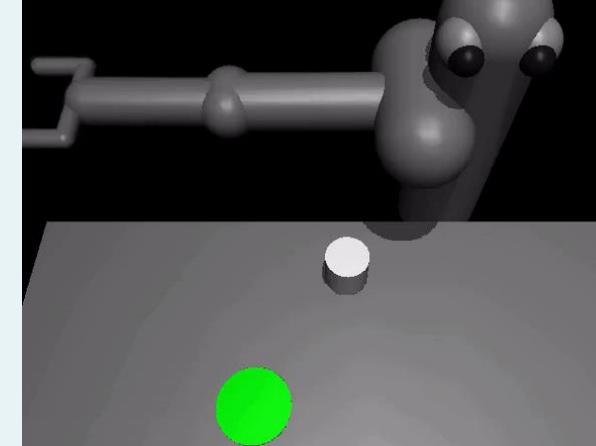
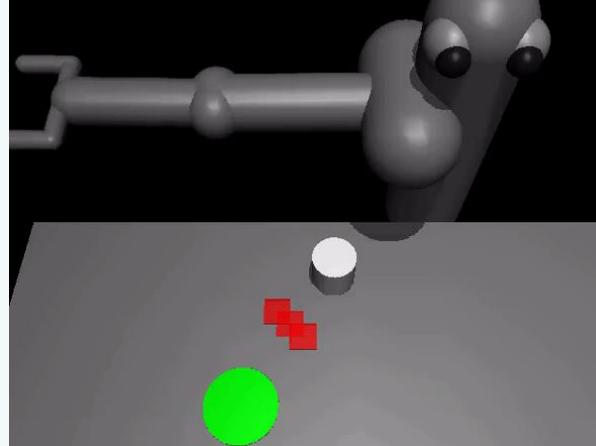
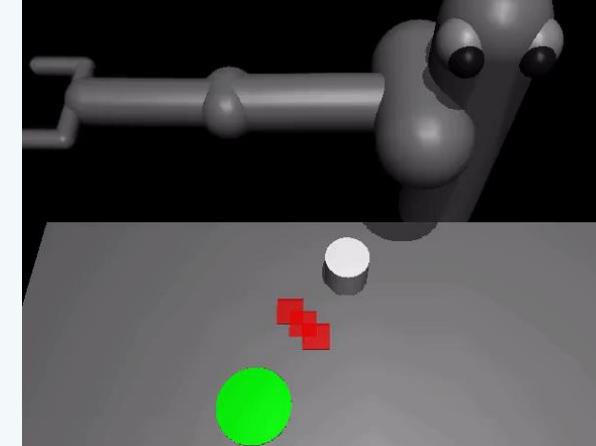
経験再生

- 目標値 (Bellman作用素)

$$y(r, s', d) = r + \gamma(1 - d) \left(\min_{j=1,2} \underbrace{Q_{\text{target},j}(s', \tilde{a}') - \tau \ln \pi(\tilde{a}' | s)}_{\text{ターゲットネットワーク}} \right)$$

過大評価を回避するための技巧

SACのようなアルゴリズムはある種のロバスト 強化学習である

	Standard RL	MaxEnt RL
Trained and evaluated without the obstacle:		
Trained without the obstacle, but evaluated with the obstacle:		

SACのようなアルゴリズムはある種の ロバスト強化学習

- エントロピ正則のため、できるだけランダムに行動しながら報酬を最大化する

➡ 方策が環境にノイズを注入し、外乱から回復するよう学習
- 理論的にも報酬や環境の摂動に対してロバスト

$$\max_{\pi} \min_{\tilde{r} \in \tilde{\mathcal{R}}(\pi)} \mathbb{E} \left[\sum_t \tilde{r}(s_t, a_t) \right]$$

$$\tilde{\mathcal{R}}(\pi) = \left\{ \tilde{r}(s_t, a_t) \mid \mathbb{E}_{\pi} \left[\sum_t \ln \sum_a \exp(r(s_t, a) - \tilde{r}(s_t, a)) \right] \leq \epsilon \right\}$$

Soft Q-learningの導出

- SACと同様の条件だが、最適作用素に正則化を導入

$$\begin{aligned}\mathcal{T}_{*,\pi_{k+1}}^{0,\tau} Q(s, a) &= r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \\ &\quad \times \left[\max_{\pi} \sum_{a'} \pi(a | s) (Q(s', a') - \tau^{-1} \ln \pi(a | s)) \right] \\ &= r(s, a) + \gamma \sum_{s'} p_T(s' | s, a) \left[\tau \ln \sum_{a'} \exp(\tau^{-1} Q(s', a')) \right]\end{aligned}$$

– Soft Q-Iteration (Haarnoja et al., 2017)そのもの

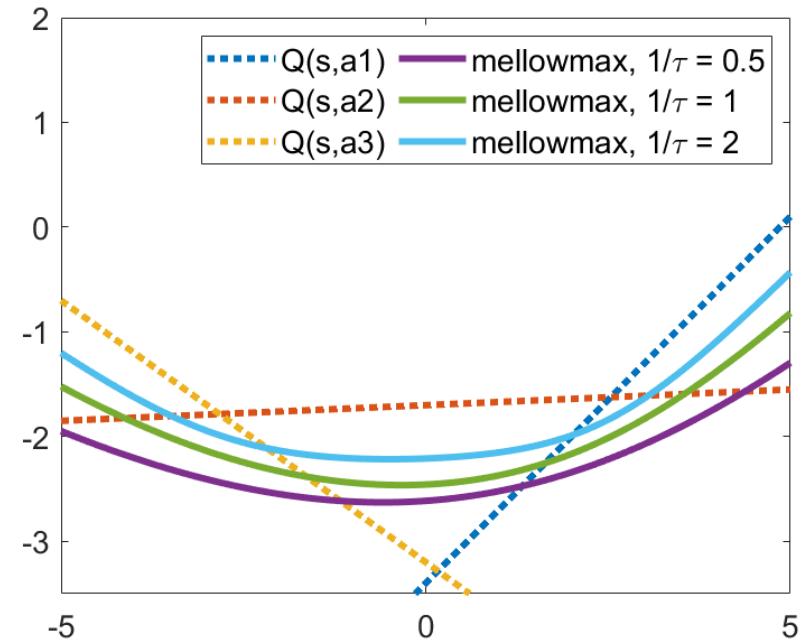
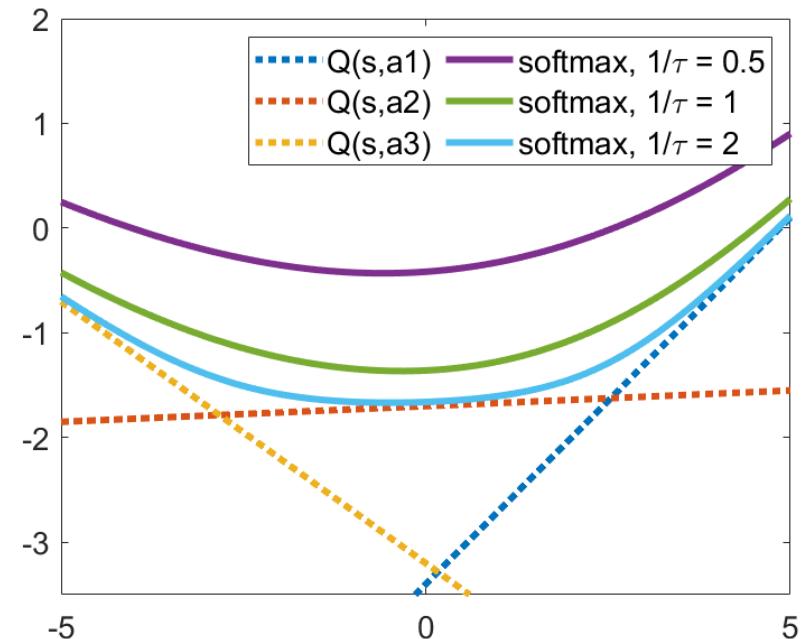
Log-sum-exp関数について

- Softmax (log-sum-exp)関数はmax関数より大きい

$$\begin{aligned}\max_a Q(s, a) &\leq \tau \ln \sum_a \exp(\tau^{-1} Q(s, a)) \\ &\leq \max_a Q(s, a) + \log |\mathcal{A}|\end{aligned}$$

- Mellowmax (log-average-exp)関数
(Asadi and Littman 2017)

$$\begin{aligned}\text{mellowmax}_a Q(s, a) &\\ \triangleq \tau \ln \left\{ \frac{1}{|\mathcal{A}|} \sum_a \exp(\tau^{-1} Q(s, a)) \right\}\end{aligned}$$



Mellowmaxの導出

- エントロピ項なし、一様方策分布 $\pi_U(a|s) = 1/|\mathcal{A}|$ をベースラインとするKL正則化ありのアルゴリズム

$$\begin{aligned} \mathcal{T}_{*,\pi_{k+1}|\pi_U}^{\lambda,0}(Q) &\triangleq r(s, a) \\ &+ \gamma \sum_{s'} p_T(s' | s, a) \left[\max_{\pi} \sum_{a'} \pi_{k+1}(a' | s') \left(Q(s', a') - \tau \ln \frac{\pi_{k+1}(a' | s')}{\pi_U(a' | s')} \right) \right] \end{aligned}$$

$\underbrace{\phantom{\max_{\pi} \sum_{a'}}}_{\text{mellowmax}}_{a'} Q(s', a')$

- ターゲットネットワークを使わなくても学習できると実験的に報告されている

K. Asadi and M. L. Littman (2017). [An Alternative Softmax Operator for Reinforcement Learning](#). Proc. of ICML, pp. 243-252.

S. Kim, K. Asadi, M. Littman, and G. Konidaris. (2019). DeepMellow: Removing the Need for a Target Network in Deep Q-Learning. Proc. of IJCAI.

TRPOの導出

- $m = \infty$, 方策評価は正則なし

$$Q_{k+1} = (\mathcal{T}_{\pi_{k+1}}^{0,0})^\infty Q_k + \epsilon_k = Q_{\pi_{k+1}} + \epsilon_k$$

実際にはMonte Carloロールアウトを使って Q 関数を計算

- 方策改善はエントロピ正則なし, KL正則化あり

$$\pi_{k+1} \in \mathcal{G}_{\pi_k}^{\lambda,0}(Q_k) = \arg \max_{\pi} \mathbb{E}_s \left[\sum_a \pi(a | s) \left(Q(s, a) - \tau \ln \frac{\pi(a | s)}{\pi_k(a | s)} \right) \right]$$

TRPOの導出

- 実際には制約付き問題として方策改善を実装

$$\pi_{k+1} \in \mathcal{G}_{\pi_k}^{\lambda, 0}(Q_k) = \arg \max_{\pi} \mathbb{E}_s \left[\sum_a \pi(a | s) \left(Q(s, a) - \tau \ln \frac{\pi(a | s)}{\pi_k(a | s)} \right) \right]$$


$$\pi_{k+1} = \arg \max_{\pi} \mathbb{E}_s [\mathbb{E}_{a \sim \pi}[Q(s, a)]] \quad \text{subject to} \quad \mathbb{E}_s [\text{KL}(\pi_k \| \pi)] \leq \epsilon$$

- 重点サンプリングの使用


$$\pi_{k+1} = \arg \max_{\pi} \mathbb{E}_s \left[\mathbb{E}_{a \sim \pi_k} \left[\frac{\pi(a | s)}{\pi_k(a | s)} Q(s, a) \right] \right] \quad \text{subject to} \quad \mathbb{E}_s [\text{KL}(\pi_k \| \pi)] \leq \epsilon$$

PPOの導出

- TRPOの制約付き方策改善を正則化の方式に修正

$$\pi_{k+1} = \arg \max_{\pi} \mathbb{E}_s \left[\mathbb{E}_{a \sim \pi_k} \left[\frac{\pi(a | s)}{\pi_k(a | s)} Q(s, a) \right] \right] \text{ subject to } \mathbb{E}_s [\text{KL}(\pi_k \| \pi)] \leq \epsilon$$



$$\pi_{k+1} = \arg \max_{\pi} \mathbb{E}_s \left[\mathbb{E}_{a \sim \pi_k} \left[\frac{\pi(a | s)}{\pi_k(a | s)} A(s, a) \right] - \lambda \text{KL}(\pi_k \| \pi) \right]$$

- エントロピ正則強化学習では $\text{KL}(\pi \| \pi_k)$ を使っていることに注意
 - Optimistic RL (Kobayashi, 2022)

まとめ

- 深層強化学習の最近の進捗状況を紹介した
- 近似修正方策反復法を紹介した
- エントロピ正則の導入した最近の学習アルゴリズムを紹介した
 - いくつかのアルゴリズムを簡単に導出した