

深層強化学習入門 –逆強化学習と生成的模倣学習–

内部英治

国際電気通信基礎技術研究所（ATR）

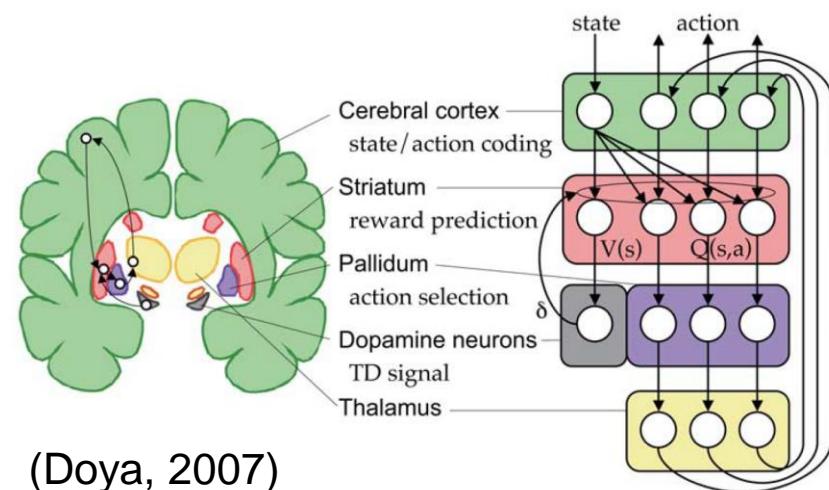
脳情報研究所 ブレインロボットインターフェース研究室



ともに究め、明日の社会を拓く

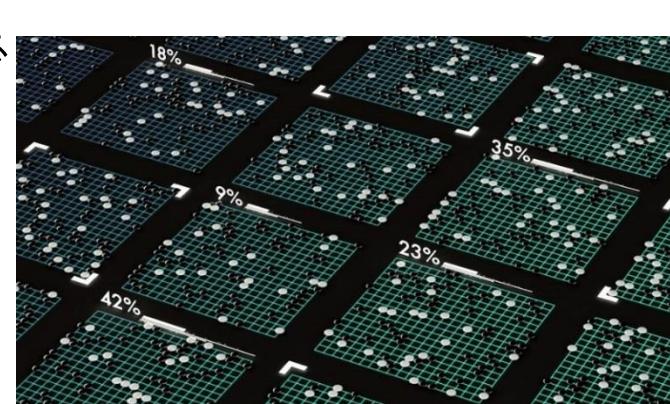
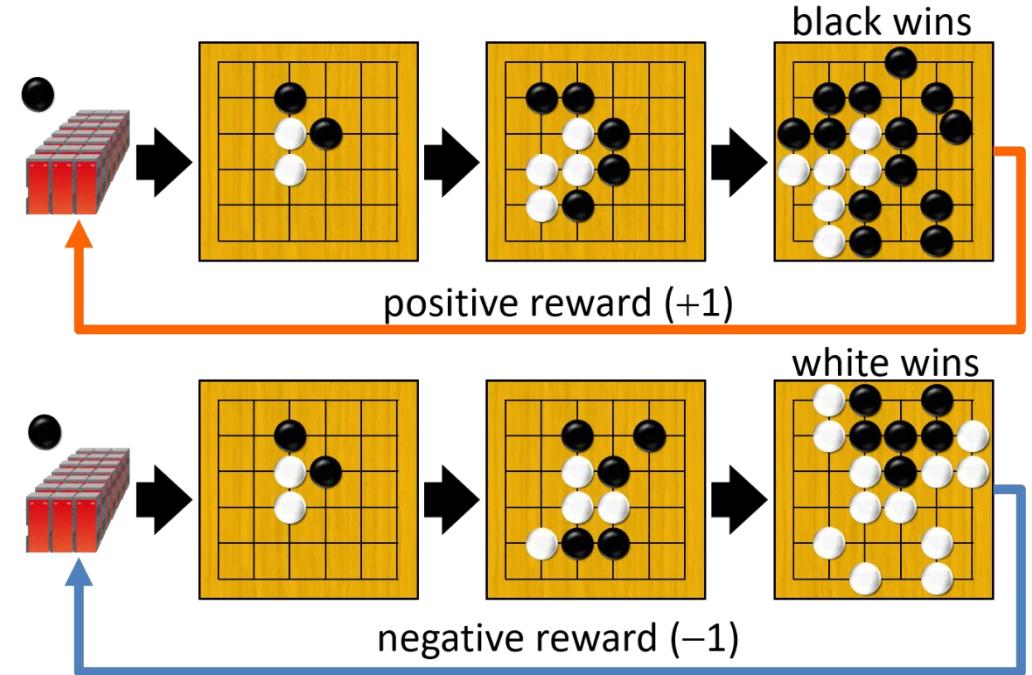
強化学習とは

- 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



報酬設計の困難さ

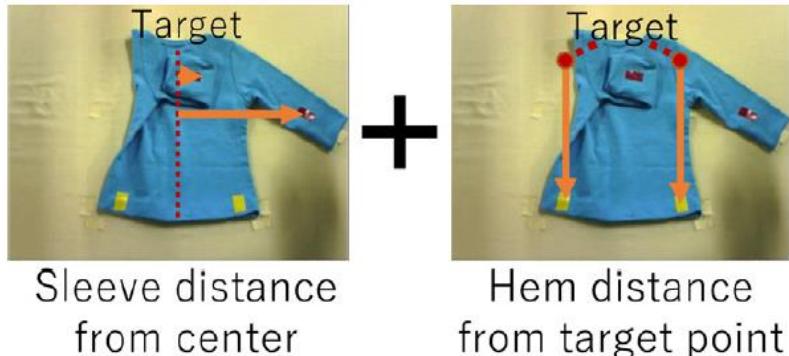
- 状態 s における行動 a の即時評価である報酬を人が設計
- 囲碁の場合
 - 勝敗に応じて正または負の報酬
 - 対戦中に与えられる報酬は0
- AlphaGo Zero (Silver et al., 2017)は3日間で490万回、40日間で2900万回の自己対戦による大量学習データが必要



柔軟物の操作の学習における報酬

Reward

The reward function is designed to trigger an action to fold the hem after folding the sleeve. The processing is shown in Algorithm 3.



Samples : 0

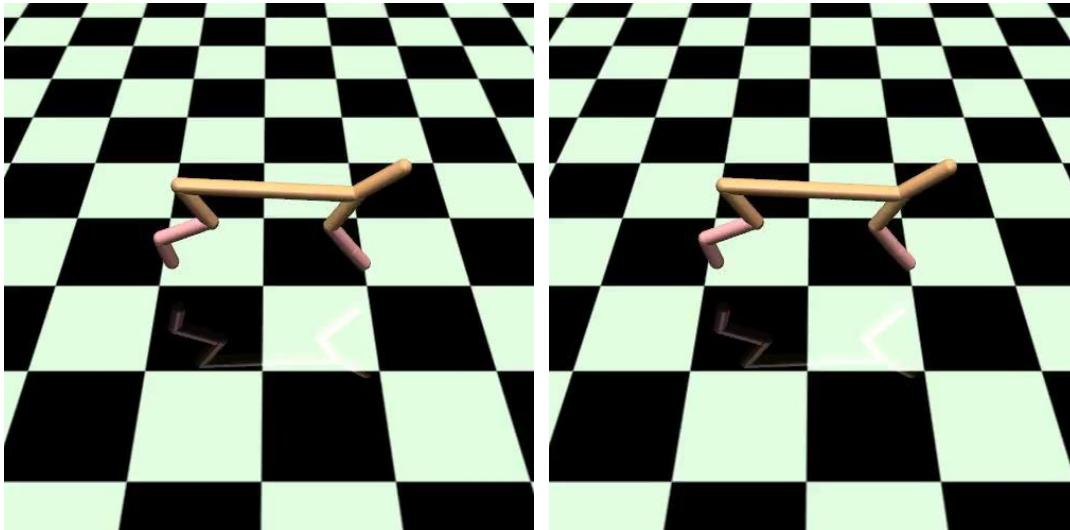
Training time : 0

Algorithm 3: Reward function of t-shirt folding task

```
Initialize  $InitHemR = [0.675, 0.8]$ ,  $InitHemL = [0.325, 0.8]$ 
Initialize  $TargetHemR = [0.675, 0.208]$ ,
 $TargetHemL = [0.325, 0.208]$ 
Function HemReward( $SleevePoint$ ,  $CenterHem$ ):
    Initialize reward = 0
    reward = -Sum( $|SleevePoint - CenterHem|$ )
    return reward
Function SleeveReward( $HemPoint$ ,  $InitHem$ ,  $TargetHem$ ):
    Initialize reward = 0
    Initialize Distance =  $|InitHem - TargetHem|$ 
    reward = Sum(Distance -  $|HemPoint - TargetHem|$ )
    return reward
Function ShirtReward():
    Initialize reward = 0
    Update color marker
    Get  $HemPointR$ ,  $HemPointL$ ,  $SleevePointR$ ,  $SleevePointL$ 
    if Detect hem marker then
         $CenterHem = (HemPointR + HemPointL)/2$ 
        reward = SleeveReward( $SleevePointR$ ,  $CenterHem$ ) +
        SleeveReward( $SleevePointL$ ,  $CenterHem$ )
    else
        reward = 1
    if Detect sleeve marker then
        reward = reward +
        HemReward( $HemPointR$ ,  $InitHemR$ ,  $TargetHemR$ ) +
        HemReward( $HemPointL$ ,  $InitHemL$ ,  $TargetHemL$ )
    return reward
```

四脚ロボットの走行課題

- タスク: できるだけ速く前進する



[Deep Reinforcement Learning Doesn't Work Yet]

- 上下反転することは想定していなかったため、いったんこの方策を学習すると、そこから元に戻れない

即時報酬

$$r(s, a) = v_x - 0.05\|a\|_2^2$$

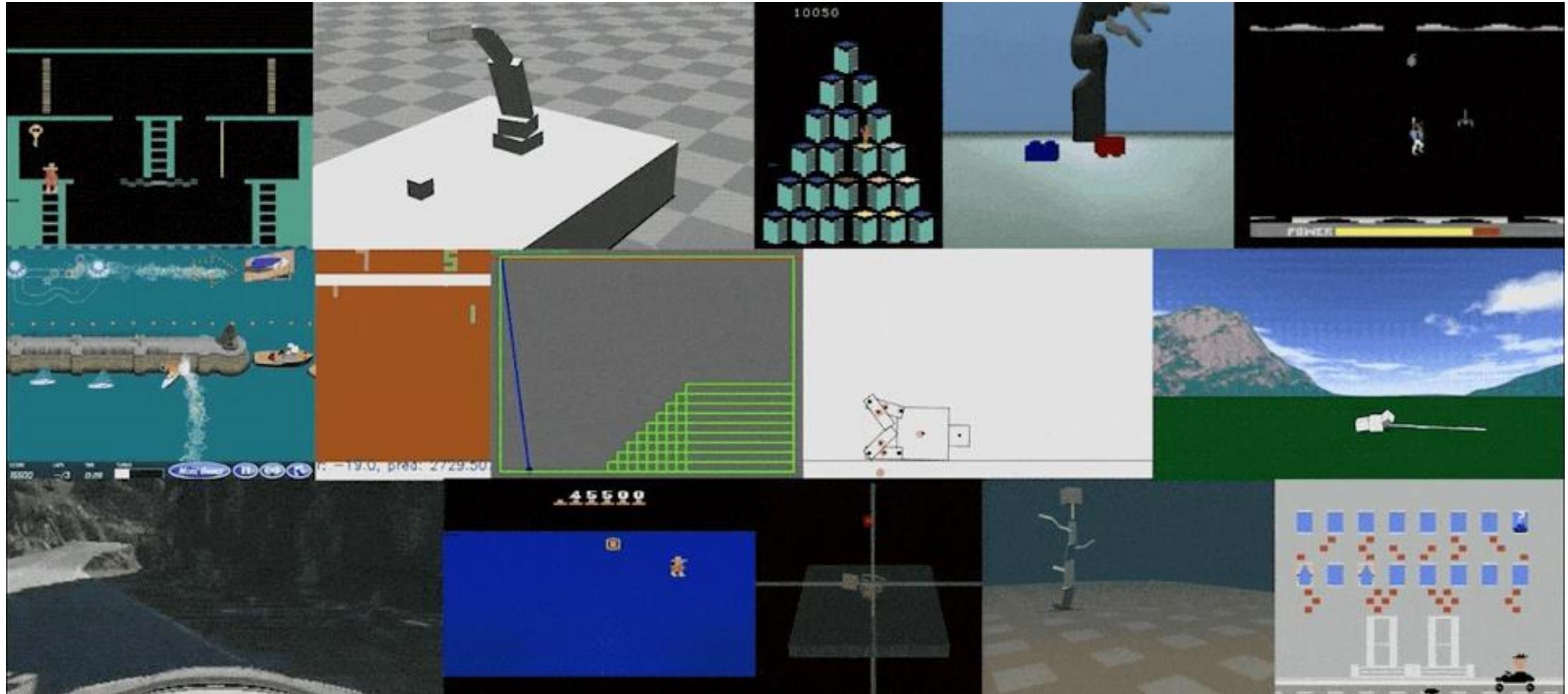
前進速度 トルクの二乗
ノルム

意図しない解を発見する強化学習

- 目的:ボートレースを早く、そして他のプレイヤーより先にゴールする
 - プレイヤーのコース進行に直接の報酬はなく、コース上に設置されたターゲットにぶつかることで高いスコアを獲得
 - 完走しなくても高得点を獲得できる方策を学習



意図しない方策を学習することは多い

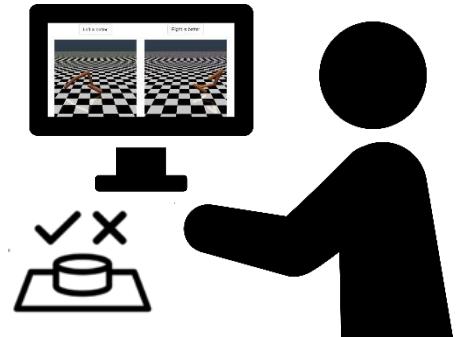
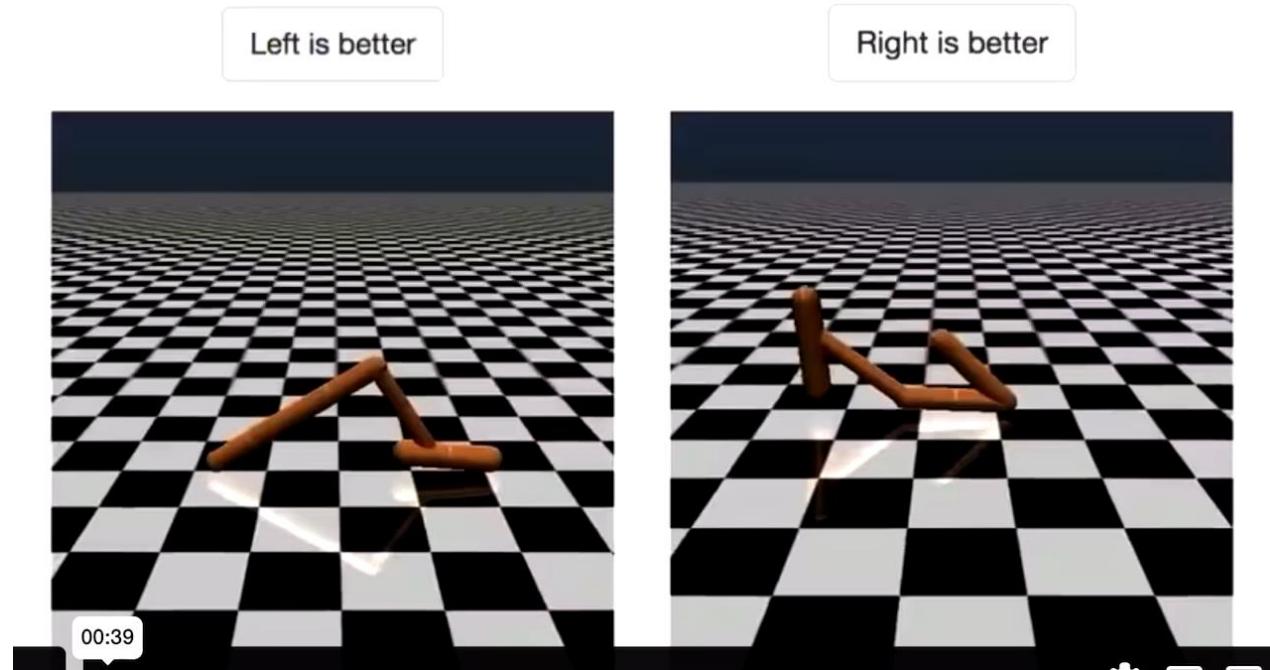


報酬をどうやって設計するか？

- 「人間からのフィードバックによる強化学習」で用いられている報酬モデルの学習
 - ChatGPTで用いられている技術で、人手でどの状態行動系列が良いかをラベル付けされたデータから報酬を推定
- 逆強化学習
 - 正解となる状態行動系列から報酬を推定
- 敵対的生成的模倣学習
 - 正解となる状態行動系列から報酬を推定し、さらに強化学習によって方策を学習

報酬モデルの学習

- どちらの動作が良いかを人が判定（ラベル付け）



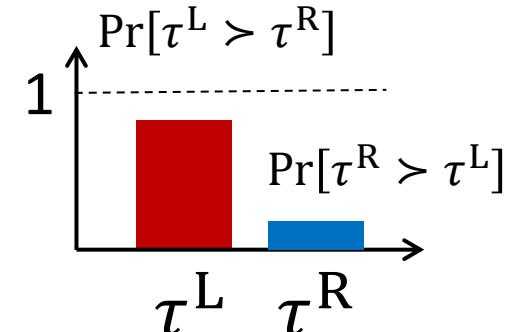
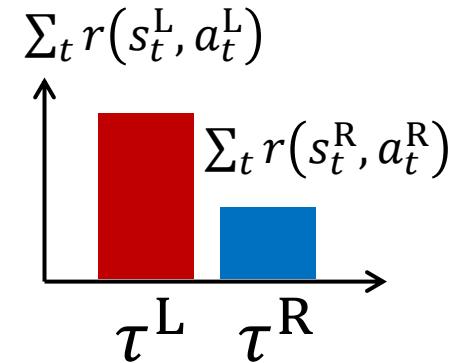
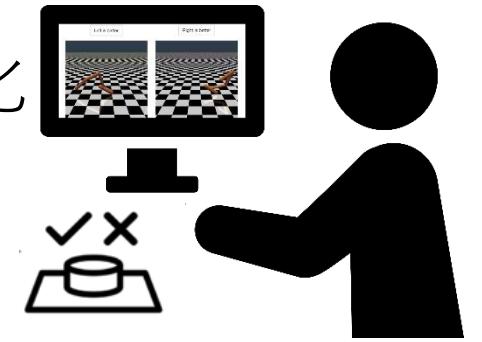
$$\tau^L = \left((s_0^L, a_0^L), (s_1^L, a_1^L), \dots, (s_{T-1}^L, a_{T-1}^L) \right)$$

$$\tau^R = \left((s_0^R, a_0^R), (s_1^R, a_1^R), \dots, (s_{T-1}^R, a_{T-1}^R) \right)$$

報酬モデルの学習

- $r(s, a; w)$: 状態 s , 行動 a の報酬を重み w でパラメータ化
- 状態行動系列 τ の総報酬は $\sum_t r(s_t, a_t; w)$
- データは $\mathcal{D} = \{(\tau_j^L, \tau_j^R, \mu_j)\}_{j=1}^N$ の形式で与えられる
 - $\mu_j = \begin{cases} 1 & \tau_j^L > \tau_j^R \\ 0 & \tau_j^L \leq \tau_j^R \end{cases}$
- 左の動作が良いと識別する確率 (Bradley-Terry モデル)

$$\Pr[\tau^L > \tau^R] = \frac{\exp(\sum_t r(s_t^L, a_t^L; w))}{\exp(\sum_t r(s_t^L, a_t^L; w)) + \exp(\sum_t r(s_t^R, a_t^R; w))}$$

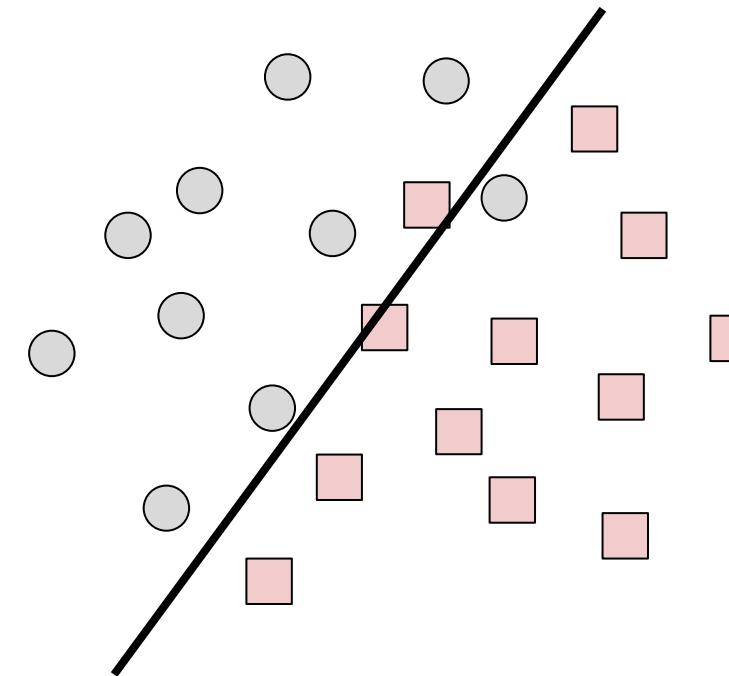


報酬モデルの学習

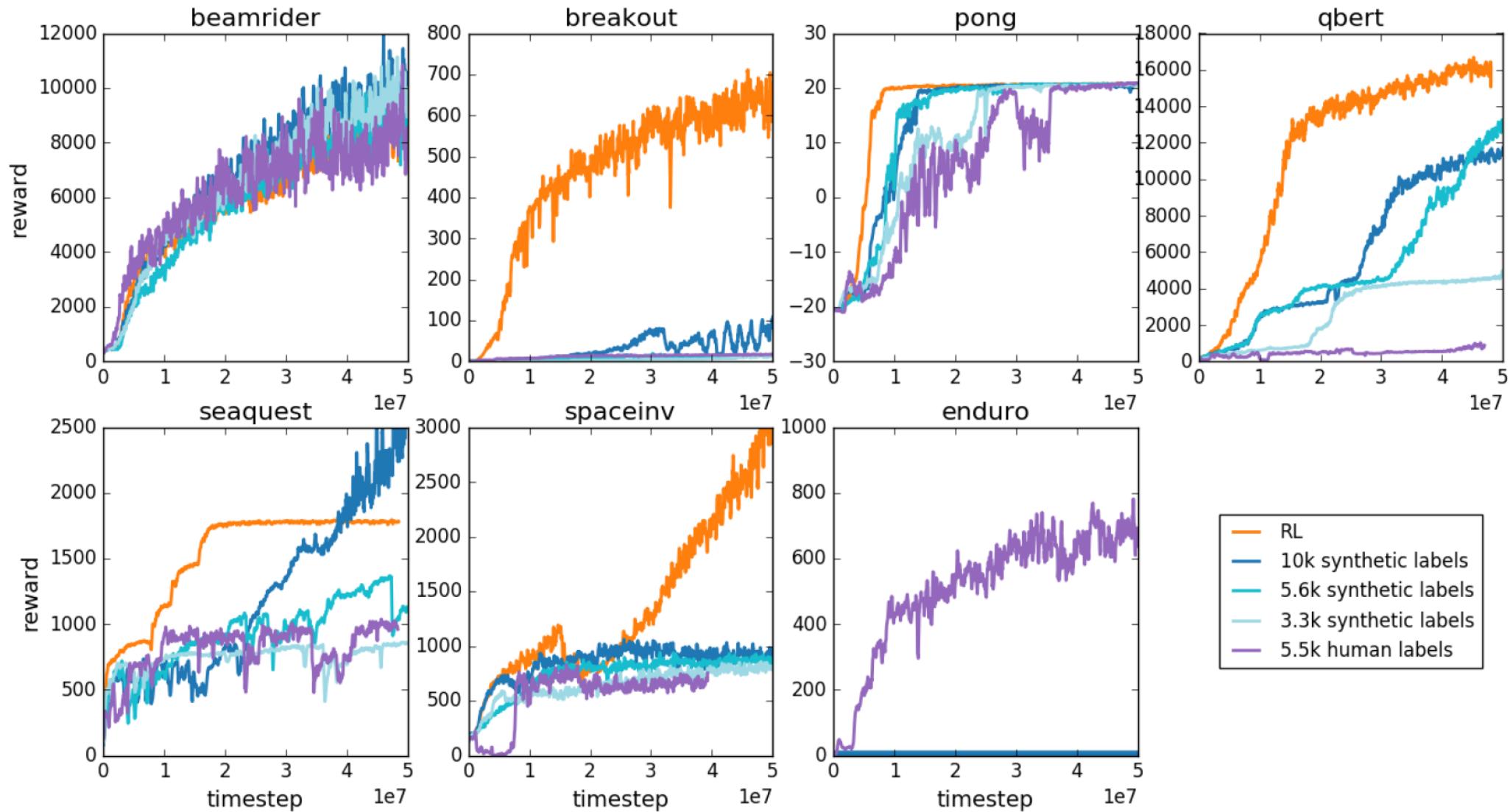
- 交差エントロピーを最小化して w を学習

$$L(w) = - \sum_{(\tau_j^L, \tau_j^R, \mu_j) \in \mathcal{D}} \mu_j \log \Pr[\tau_j^L > \tau_j^R] + (1 - \mu_j) \log(1 - \Pr[\tau_j^L > \tau_j^R])$$

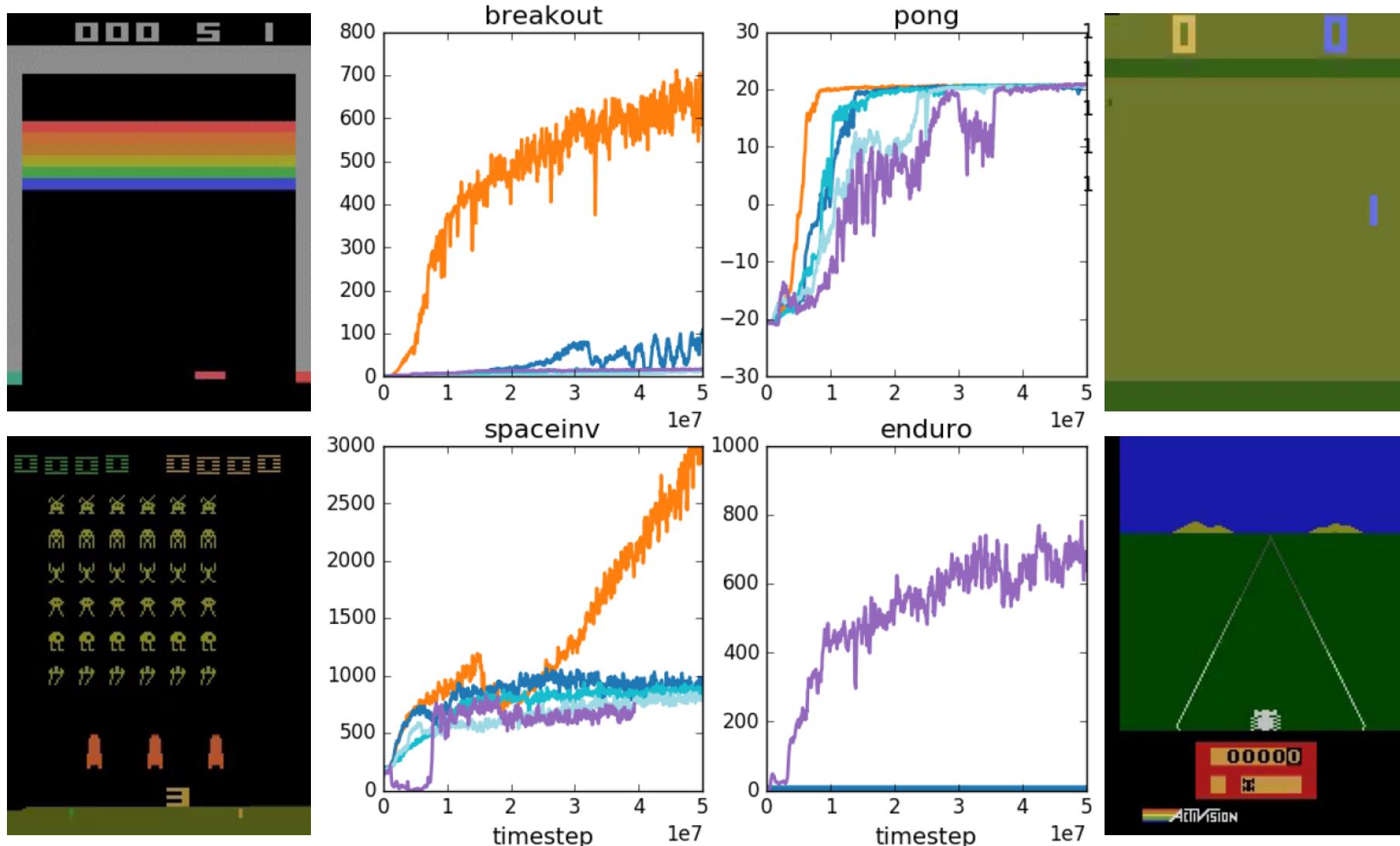
- 二値分類問題



Atariゲームでの検証



Atariゲームでの検証



報酬モデルの学習の問題点

- $\mathcal{D} = \{(\tau_j^L, \tau_j^R, \mu_j)\}_{j=1}^N$ のようなデータの準備はかなり大変
- 人手によるラベル付け μ_j は曖昧
 - ChatGPTのような自然言語と異なり、ロボットの動作の良し悪しの判定は人には向いていない？
- 報酬を学習した後で通常の強化学習を適用するため、学習効率はかなり悪い

報酬をどうやって設計するか？

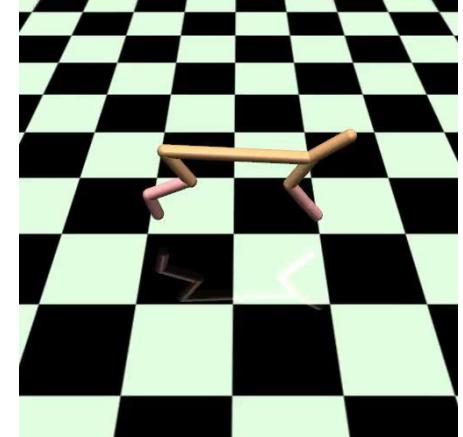
- 「人間からのフィードバックによる強化学習」で用いられている報酬モデルの学習
 - ChatGPTで用いられている技術で、人手でどの状態行動系列が良いかをラベル付けされたデータから報酬を推定
- 逆強化学習
 - 正解となる状態行動系列から報酬を推定
- 敵対的生成的模倣学習
 - 正解となる状態行動系列から報酬を推定し、さらに強化学習によって方策を学習

逆強化学習とは

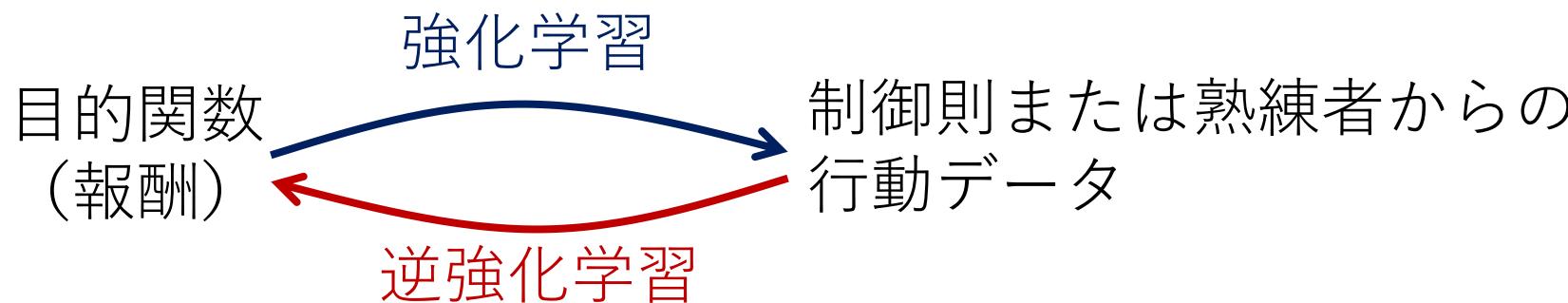
- 単純な報酬を使うと膨大な学習データと計算時間が必要
- 詳細な報酬を事前に設計するのは困難
 - 意図とは異なる行動を学習
- 熟練者の行動データをもとに報酬を推定する技術が逆強化学習



[OpenAI Blog. Faulty Reward ...]



[Sorta Insightful (Blog)]



素朴な模倣学習（行動クローニング）との関係

- エキスパートが提供する状態行動対（デモンストレーション）から行動を直接学習する

$$\mathcal{D}^E = \{(s_t^i, a_t^i)\}_{i=1}^{N^E}$$

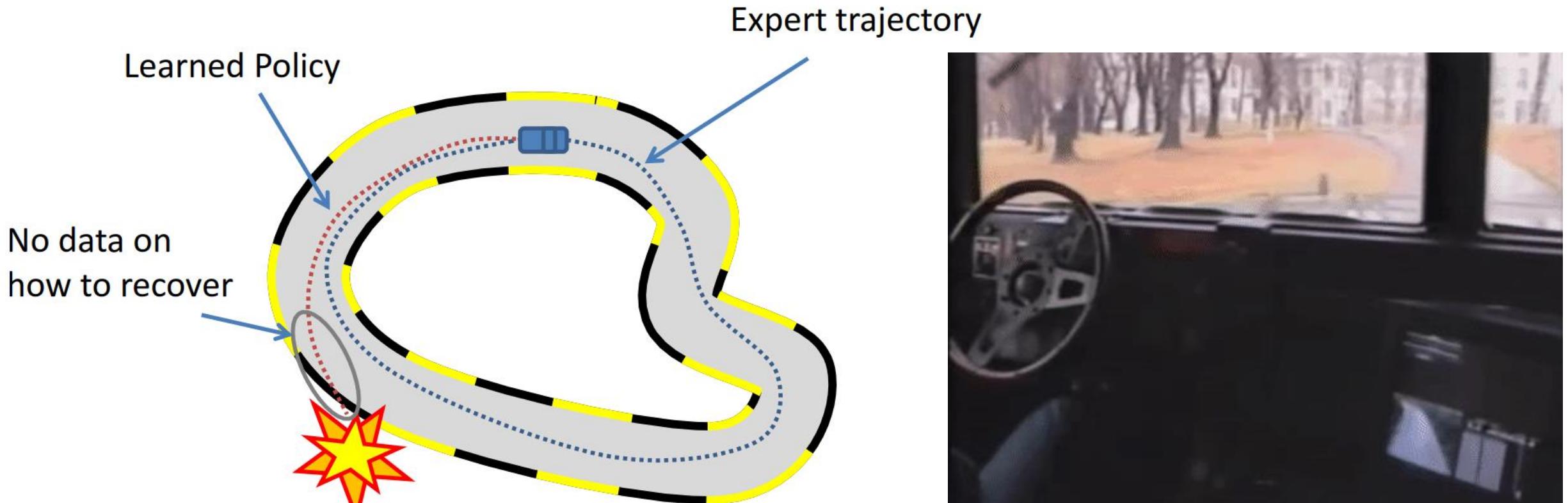


- パラメータ θ で表した決定論の方策を二乗誤差を最小にするように学習

$$\min_{\theta} L = \frac{1}{N^E} \sum_{i=1}^{N^E} (a_t^i - \pi(s_t^i; \theta))^2$$

ナイーブな模倣学習はうまくいかない

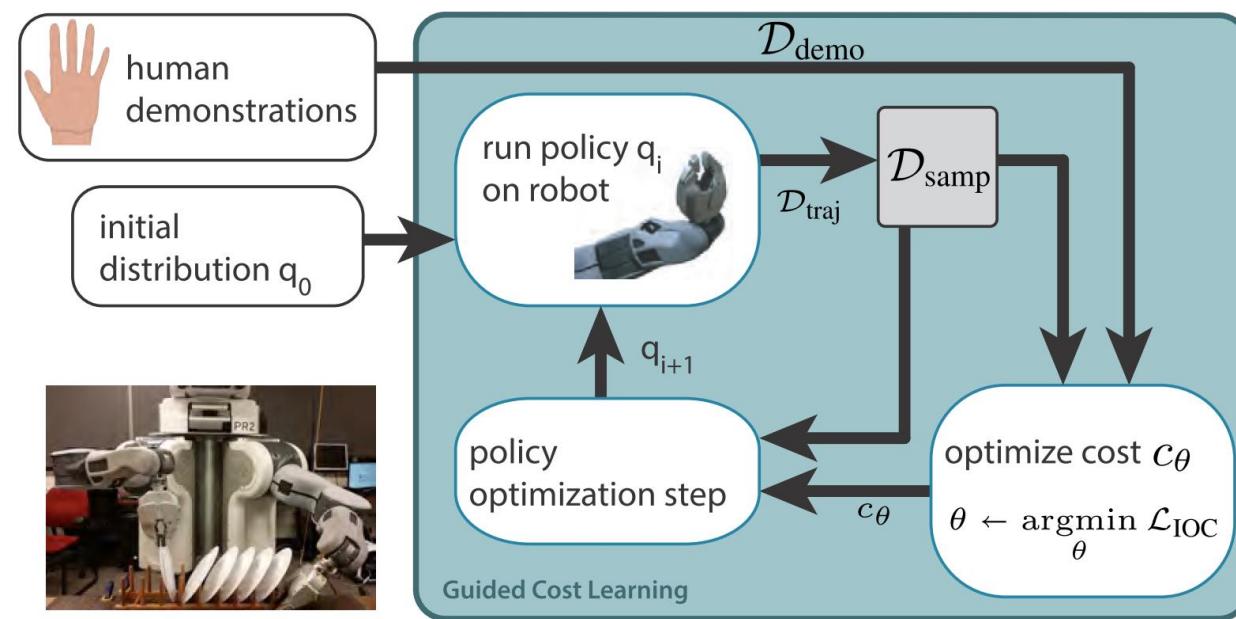
- 次第にエキスパートの軌道から外れ、誤差が大きくなる



- もとの軌道に戻るメカニズムがない

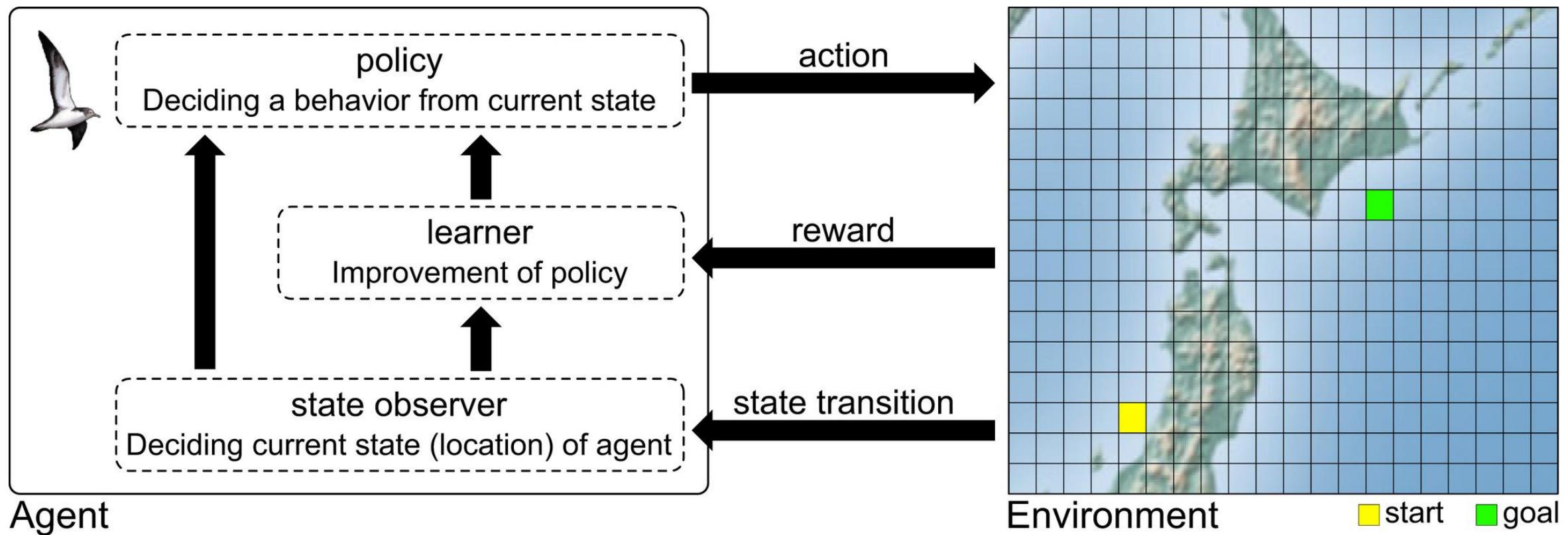
ロボット制御への応用

- Kinesthetic demonstrationにより生成されたエキスパートデータを敵対的模倣学習
 - 報酬推定と方策の改善の繰り返し



海鳥（オオミズナギドリ）の飛行経路予測

- モデルベース逆強化学習MaxEnt IRL (Ziebart et al., 2010)を適用
- 環境を量子化し、離散状態・離散行動MDP環境を作成

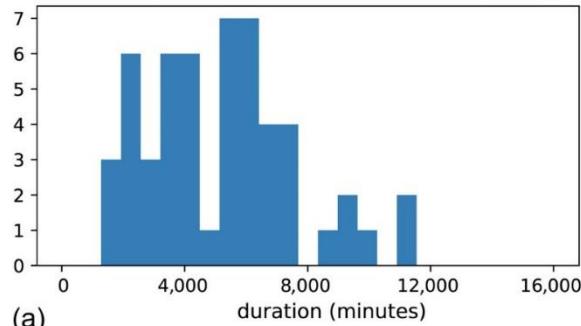
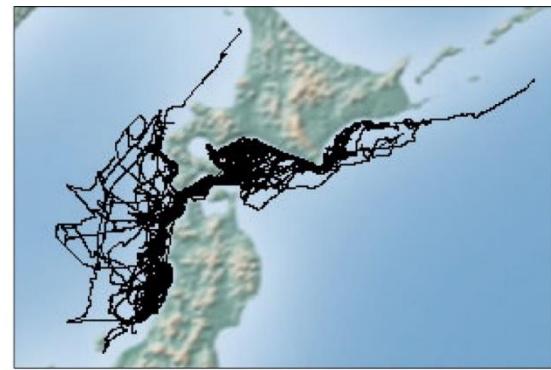


Hirakawa, T., Yamashita, T., Tamaki, T., Fujiyoshi, H., Umezu, Y., Takeuchi, I., Matsumoto, S., and Yoda, K. (2018). [Can AI predict animal movements? Filling gaps in animal trajectories using inverse reinforcement learning](#). Ecosphere.

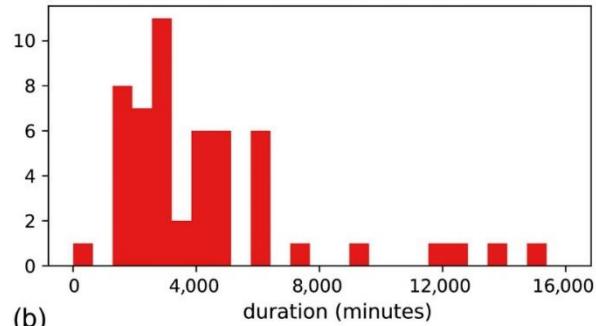
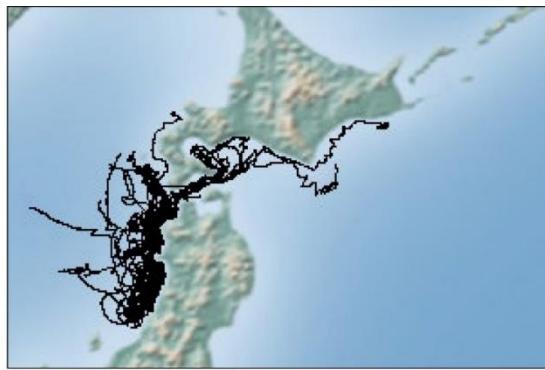
計測データ

- 106 trajectories (53 males and 53 females)

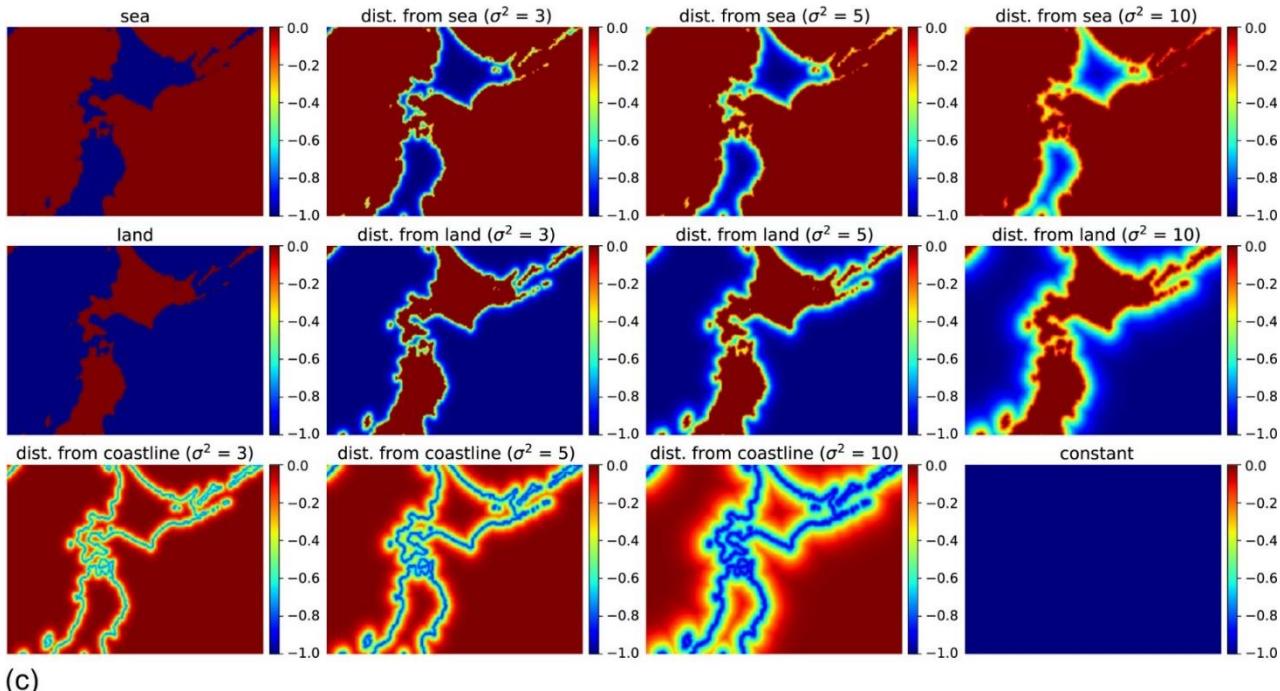
オス



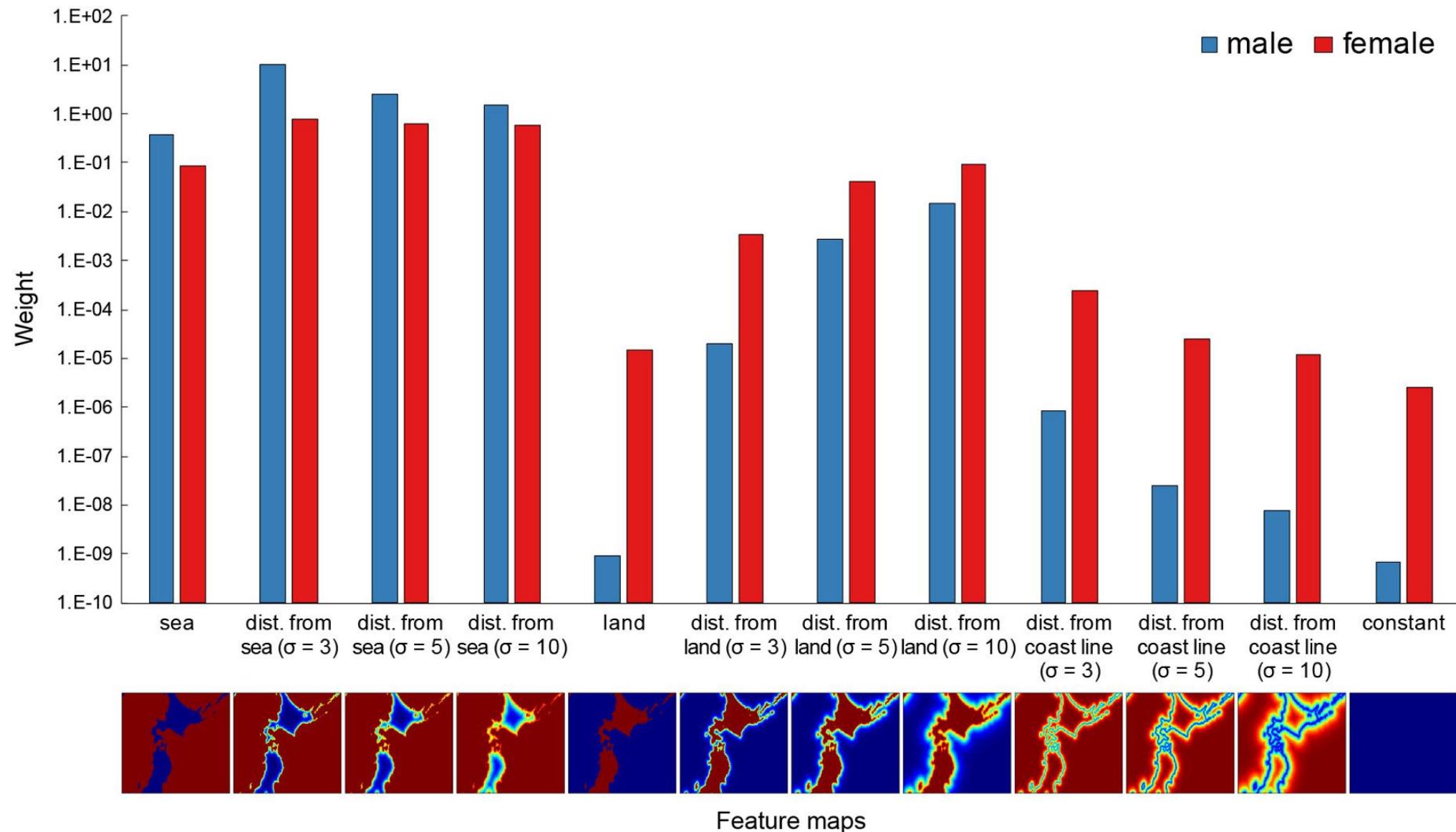
メス



報酬を表現する特徴量



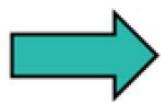
推定された報酬関数



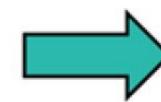
線虫*C. elegans*の温度走性行動

- 以下の二つのモードから構成されていることを発見
 - 効率的に成育温度に向かうモード
 - 同じ温度の等温線に沿って移動するモード

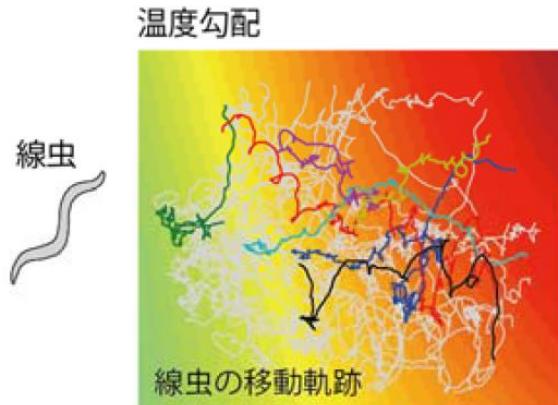
行動時系列データ



逆強化学習法



行動戦略

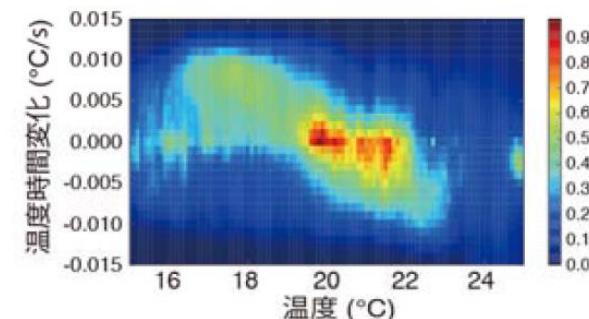


線虫は育成された温度を好むように、また
飢餓を経験した温度を避けるように移動する。

行動戦略を示す数式

$$\pi(s' | s) = \frac{p(s' | s) \exp(v(s'))}{\sum_x p(x | s) \exp(v(x))}$$

推定された価値関数： $v(s)$



動物が行動していく遭遇する各状況が、戦略上
どれくらいの価値があるのかを示している。

卓球動作の解析

- 初心者や熟練者の行動を報酬の観点から分類

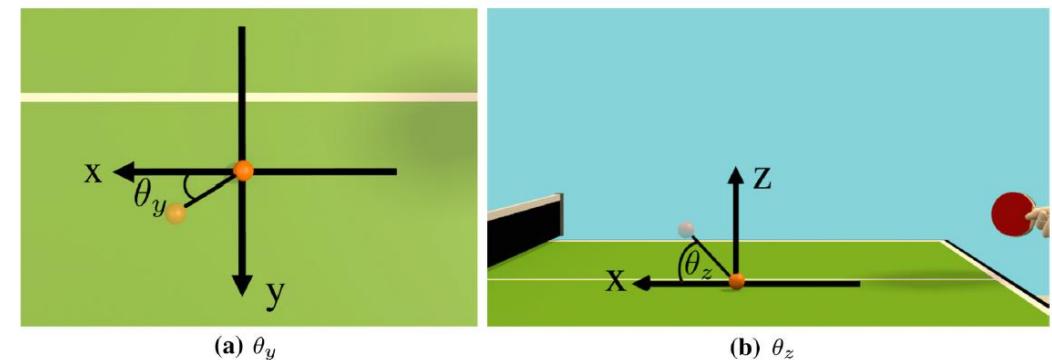
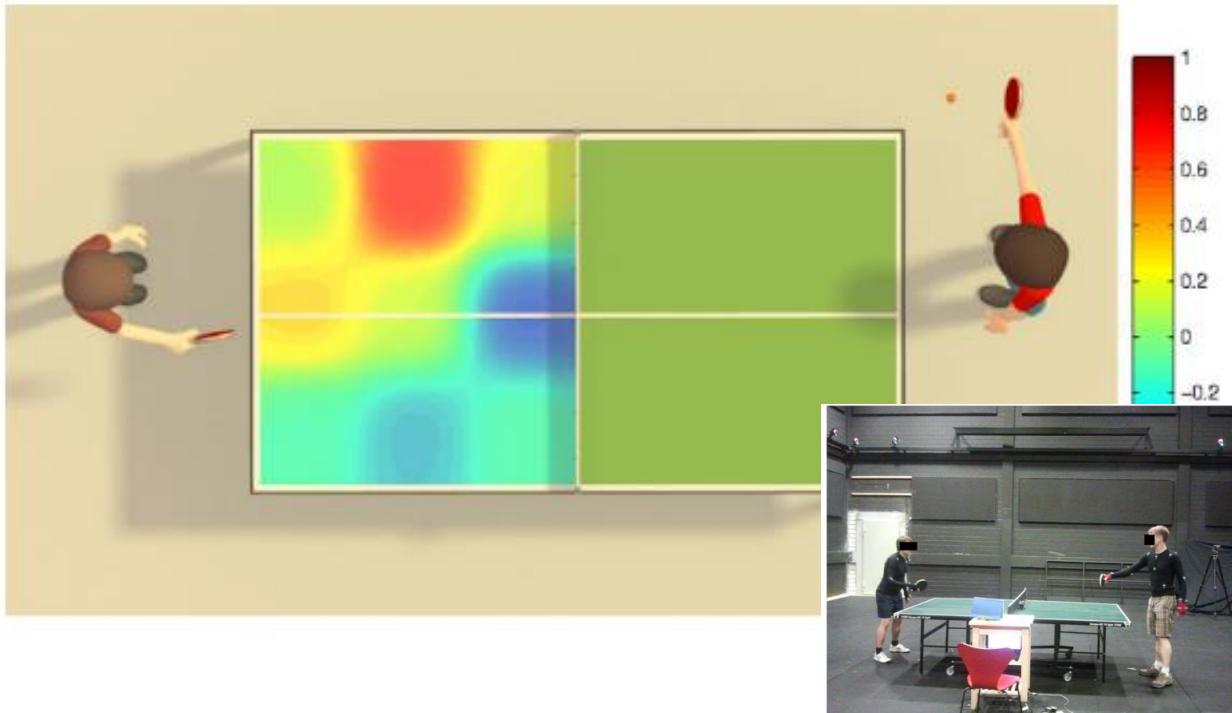
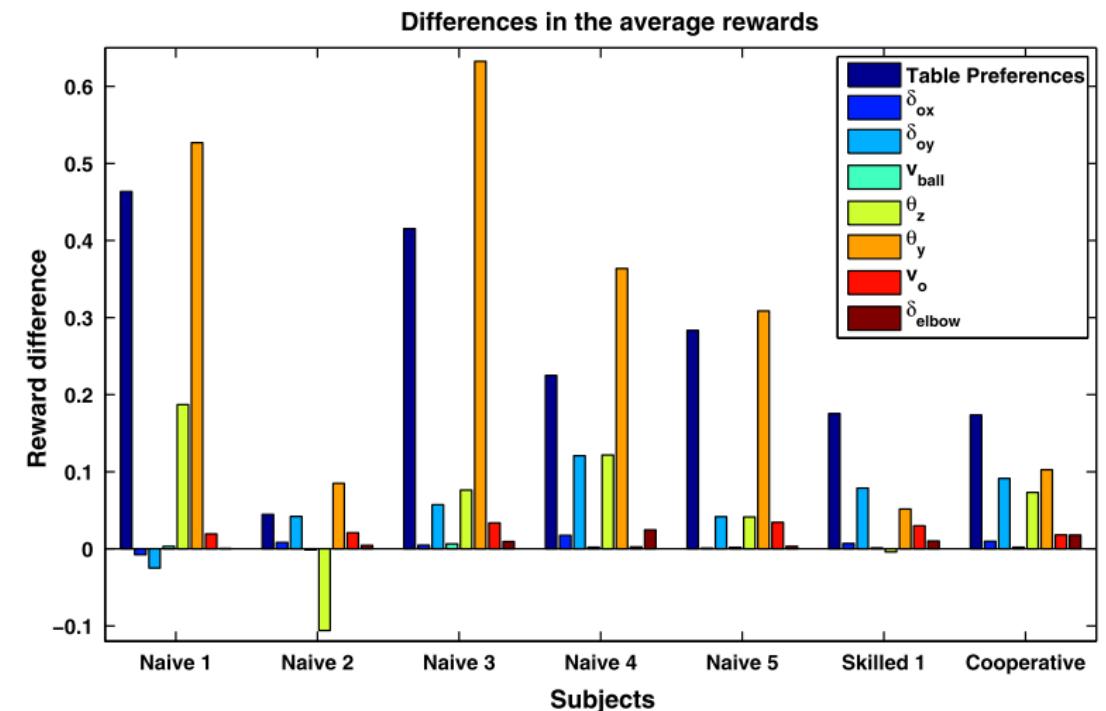
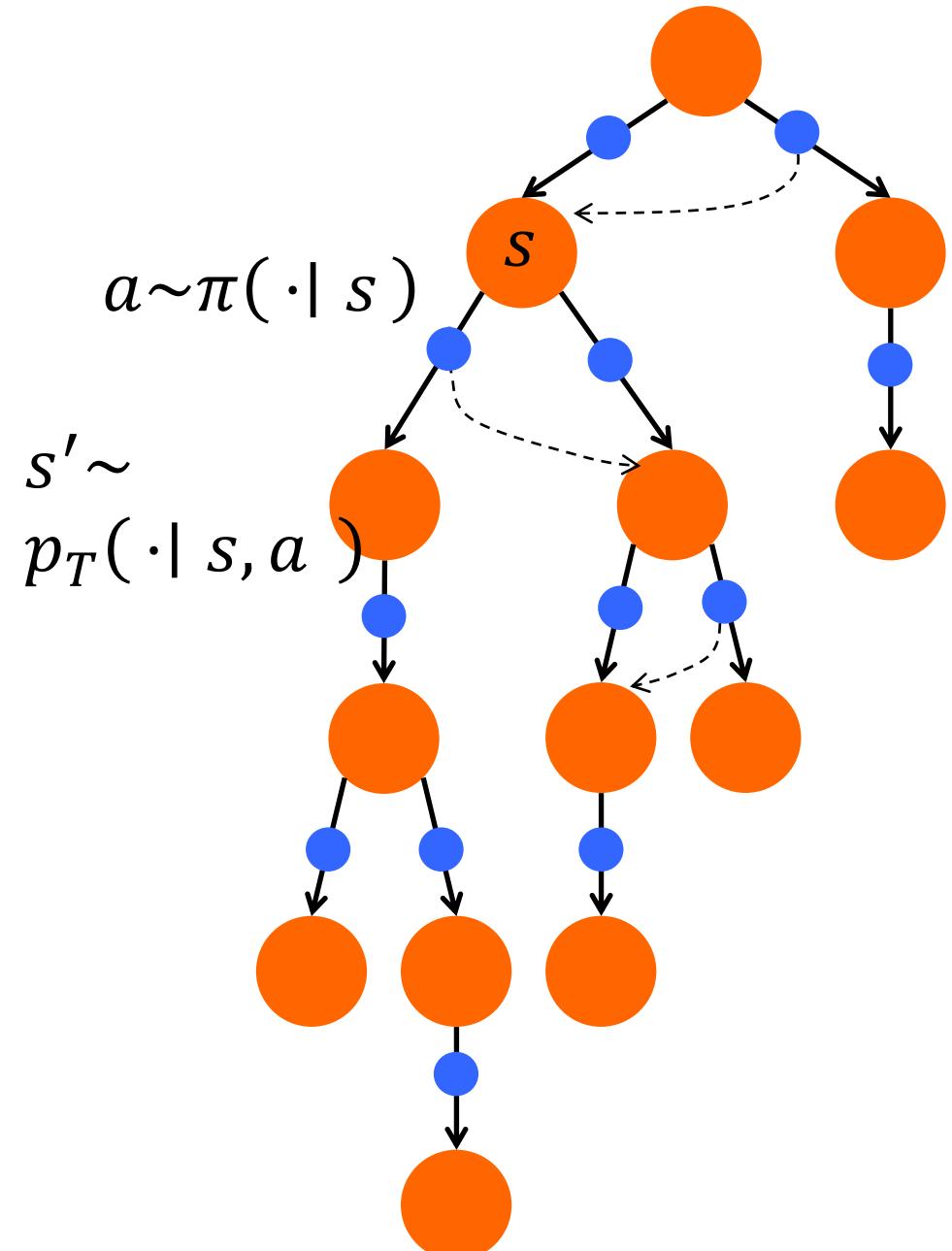


Fig. 5 The bouncing angles θ_y and θ_z in the xy - and xz -surface define the orientation of the ball. While θ_z corresponds to the horizontal bouncing angle, θ_y corresponds to the direction of the ball and thereby defines if the ball is played cross to the left, cross to the right or straight



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 (世界モデル)
$\pi(a s)$	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)



逆強化学習の目的

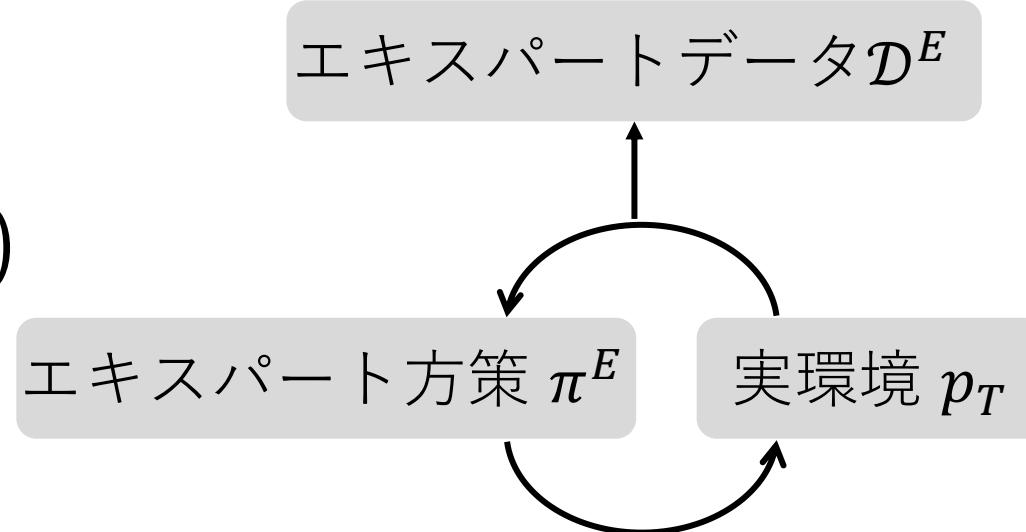
- エキスパート方策 $\pi^E(a|s)$ から「報酬なしの」状態行動系列

$$\mathcal{D}^E = \{\tau^i\}_{i=1}^{N^E}$$

$$\tau^i = (s_0^i, a_0^i, s_1^i, a_1^i, \dots, s_{T-1}^i, a_{T-1}^i, s_T^i)$$

が与えられている

- π^E, p_T は未知



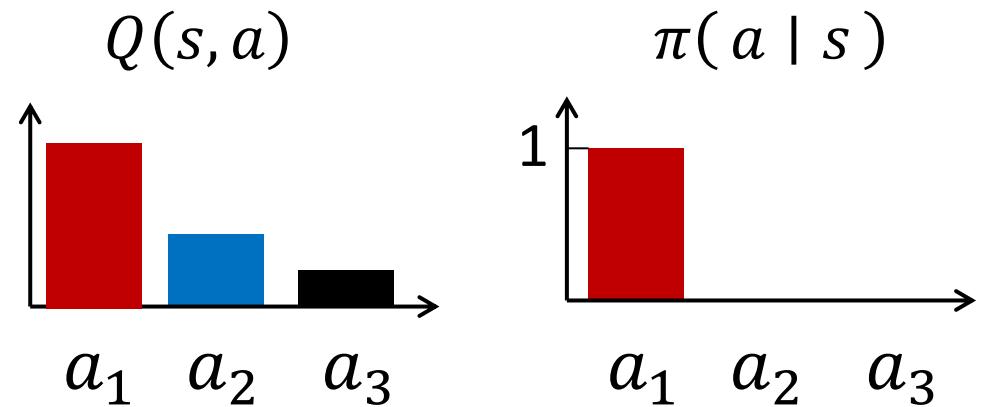
- エキスパートが報酬 $r(s, a; w)$ を使って π^E を学習したと仮定
 - w : 報酬を表すネットワークの重み

逆強化学習の困難さ

- エキスパート方策 π^E が「ある報酬の下で強化学習によって得られた」ものであるなら、先週説明した近似修正方策反復法の解となっているはずである

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

- $\pi^E(a | s)$ は Q 関数の大小のみで決定され、微分不可能



逆強化学習の目的

- エキスパート方策 $\pi^E(a|s)$ から「報酬なしの」状態行動系列

$$\mathcal{D}^E = \{\tau^i\}_{i=1}^{N^E} \quad \tau^i = (s_0^i, a_0^i, s_1^i, a_1^i, \dots, s_{T-1}^i, a_{T-1}^i, s_T^i)$$

が与えられている

- \mathcal{D}^E を生成できる報酬 $r(s, a; w)$ を推定する

- w : 報酬を表すネットワークの重み

- Markov性の仮定の下

$$\Pr(s_0, a_0, \dots, s_T, a_T, s_{T+1}) = p(s_0) \prod_{t=0}^{T-1} p_T(s_{t+1} | s_t, a_t) \pi^E(a_t | s_t)$$

正則化を導入した近似修正方策反復法による アプローチ

- $\pi^E \in \mathcal{G}_{\pi_k}^{\lambda, \tau}(Q_k) = \arg \max_{\pi} \sum_a \pi(a | s) Q(s, a) - \lambda \text{KL}(\pi \| \pi_k) + \tau \mathcal{H}(\pi)$

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

$$\mathcal{T}_{\pi | \mu}^{\lambda, \tau}(Q) \triangleq r(s, a)$$

$$+ \gamma \sum_{s'} p_T(s' | s, a) \left[\sum_{a'} \pi(a' | s') Q(s', a') - \lambda \text{KL}(\pi \| \mu) + \tau \mathcal{H}(\pi) \right]$$

最大エントロピ逆強化学習

- $\lambda = 0$ としたときの最適方策

$$\begin{aligned}\pi^E(a | s) &= \arg \max_{\pi} \sum_a \pi(a | s) Q(s, a) + \tau \mathcal{H}(\pi) \\ &= \frac{\exp(\tau^{-1} Q(s, a))}{\sum_a \exp(\tau^{-1} Q(s, a))}\end{aligned}$$

- エキスパート方策 π^E をQの関数として表現し、 \mathcal{D}^E を使って学習する
- 他の状態 $s' \neq s$ のことを考慮していない
 - 状態についての関数 $F(s)$ を引いても影響がない

$$Q'(s, a) = Q(s, a) - F(s)$$

最大エントロピ逆強化学習

- Q関数をロールアウトで置き換えとしたときの最適方策

$$\tau^{-1}Q'(s, a) \approx \sum_{t=0}^{T-1} \gamma^t r(s_t, a_t; w) \triangleq G(w)$$

- エキスパート方策をエキスパート軌道に変換

$$\Pr(\tau | s_0; w) = \frac{\exp(\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t; w))}{\sum_{\tau} \exp(\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t; w))} = \frac{1}{Z(w)} \exp\left(\sum_{t=0}^{T-1} \gamma^t r(s_t, a_t; w)\right)$$

- $\tau = (a_0, s_1, a_1, \dots, s_{T-1}, a_{T-1}, s_T)$
- とりうる τ が2種類しかない場合、これは報酬モデルの学習で用いたものと同じ

最大エントロピ逆強化学習

- あとは \mathcal{D}^E を使って最尤推定すればよいが...

$$\nabla_w \ln \Pr(\tau | s_0) = \mathbb{E}_{\pi^E}[\nabla_w G(w)] - \nabla_w \ln Z(w)$$
$$= \underbrace{\mathbb{E}_{\pi^E}[\nabla_w G(w)]}_{\text{エキスパートのデータから得られる総報酬の勾配}} - \underbrace{\mathbb{E}_{\Pr(\tau|s_0;w)}[\nabla_w G(w)]}_{\text{報酬 } r(s, a; w) \text{ で学習した最適方策を使って得られる総報酬の勾配}}$$

- 最大エントロピ逆強化学習は第2項をまじめに解く

正規化定数 $Z(w)$ の評価方法の違い

- $Z(w)$ を別の分布 $b(\tau)$ を使った重点サンプリングで近似

$$\ln Z(w) = \ln \sum_{\tau} \exp(G(w)) = \ln \sum_{\tau} b(\tau) \frac{\exp(G(w))}{b(\tau)} = \ln \mathbb{E}_b \left[\frac{\exp(G(w))}{b(\tau)} \right]$$

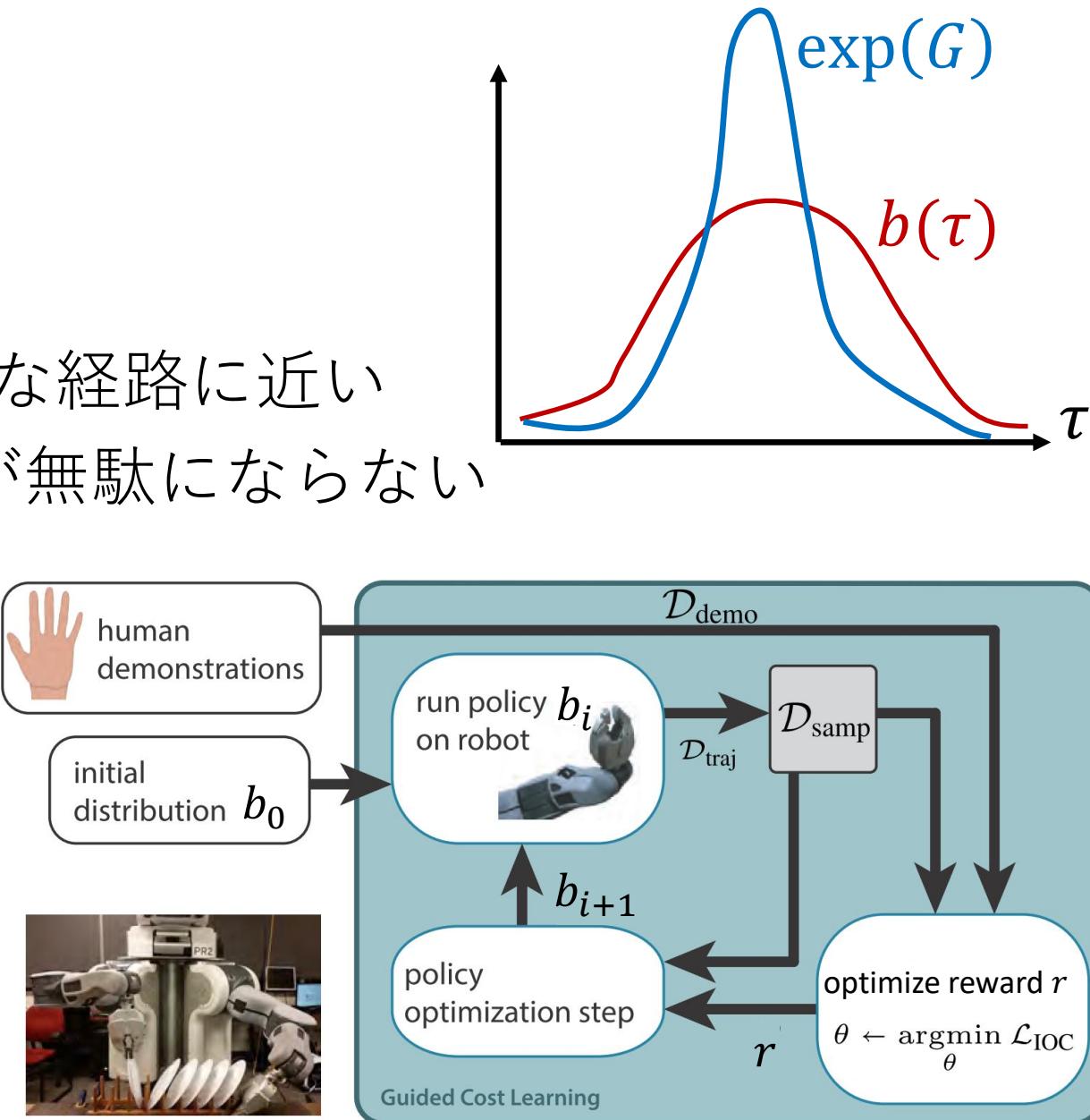
- RELENT (Boularias et al., 2011): $b(\tau)$ は一様分布。エキスパート分布と大きく異なり、推定が不安定
- PI_LOC (Kalakrishnan et al., 2013): エキスパートデータを撮動したデータを用いるが、重点サンプリングによる補正なし
- PI_IOC (Aghasadeghi and Bretl, 2011): 推定した報酬をもとに学習した方策によって生成された系列を用いるが、重点サンプリングによる補正なし

Guided Cost Learning

- $b(\tau)$ としてどんなものが良い？
- $\exp(G(\tau))$ が大きい、つまり最適な経路に近いほど $b(\tau)$ は大きい方がサンプルが無駄にならない

$$b(\tau) \propto \exp(G(w))$$

- 報酬の推定値を使って最適方策を求めることが等価
- 報酬推定と方策更新を繰り返す



最大エントロピ逆強化学習の問題点

- (同じ長さの) 状態行動系列をデータの要素とするので,
 - 正解となるエキスパートのデータの収集が大変
 - 系列が長くなると, 推定される重みの分散が増加
- 最大エントロピ逆強化学習はモデルベースで, 勾配の計算のたびに動的計画法を解く必要がある
- Guided Cost Learningはモデルフリーだが, $b(\tau)$ の更新の仕方の妥当性があいまい

報酬をどうやって設計するか？

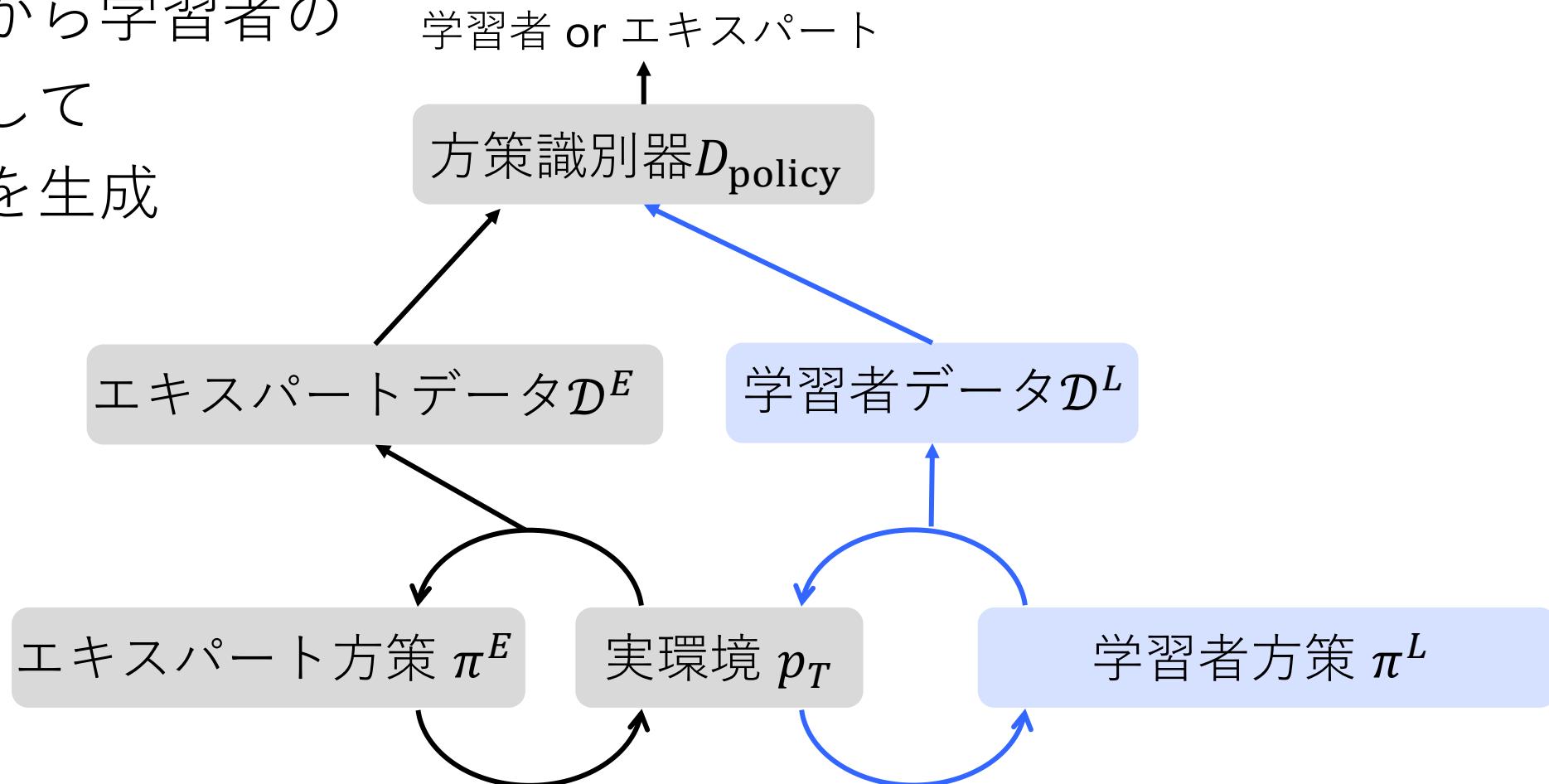
- 「人間からのフィードバックによる強化学習」で用いられている報酬モデルの学習
 - ChatGPTで用いられている技術で、人手でどの状態行動系列が良いかをラベル付けされたデータから報酬を推定
- 逆強化学習
 - 正解となる状態行動系列から報酬を推定
- 敵対的生成的模倣学習
 - 正解となる状態行動系列から報酬を推定し、さらに強化学習によって方策を学習

順強化学習と逆強化学習の組み合わせによる報酬推定

- Guided Cost Learning (Finn et al., 2016a) は
 - 報酬推定のために正規化定数 Z (の勾配) の評価が必要
 - ある分布からのサンプルを使って近似するために重点サンプリングが必要
 - 分布を改善するために強化学習を利用
- 結果としてエキスパートデータから報酬と方策を求める枠組みになる
➡ 敵対的生成ネットワークと同等

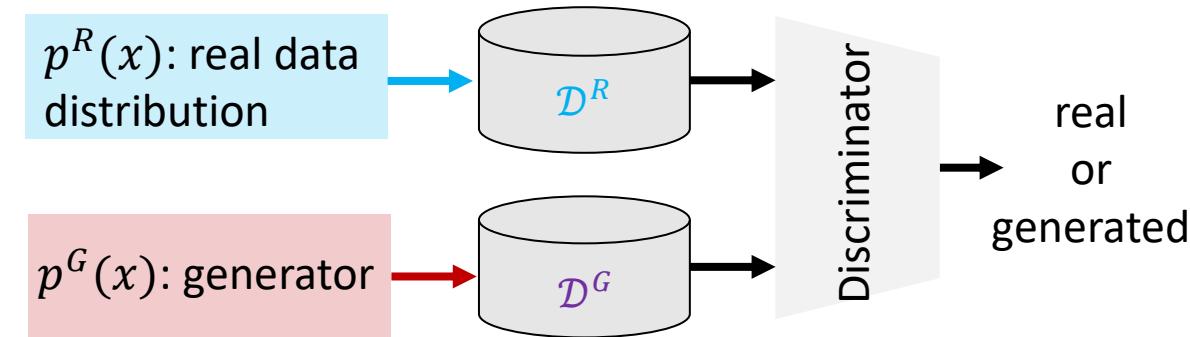
敵対的生成的模倣学習

1. エキスパートの軌道と学習者が生成した軌道を比較
2. 軌道の違いから学習者の方策を更新して異なる軌道を生成



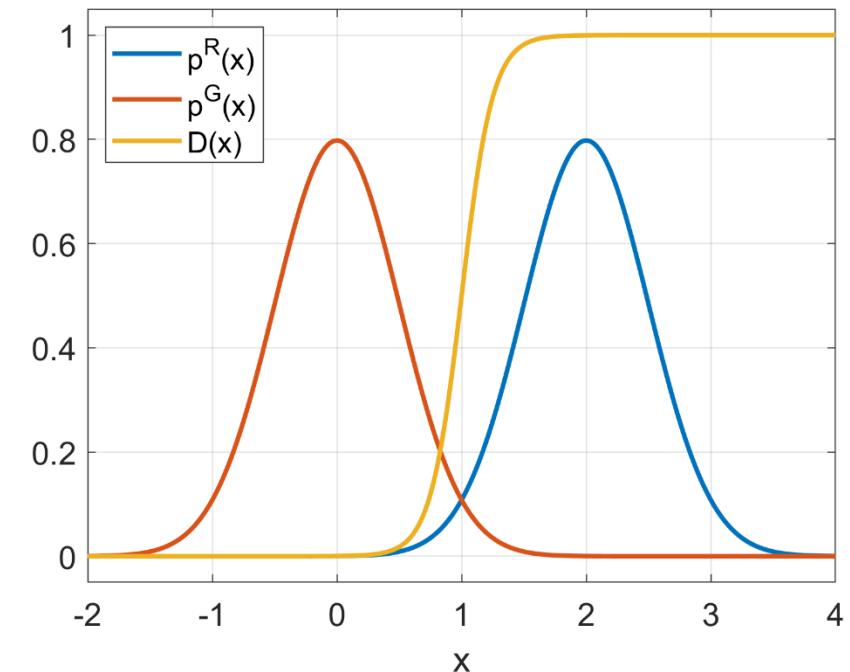
Reminder: Generative Adversarial Networks (GANs)

- Game theoretic framework for solving generative modeling problems (Goodfellow et al., 2014)



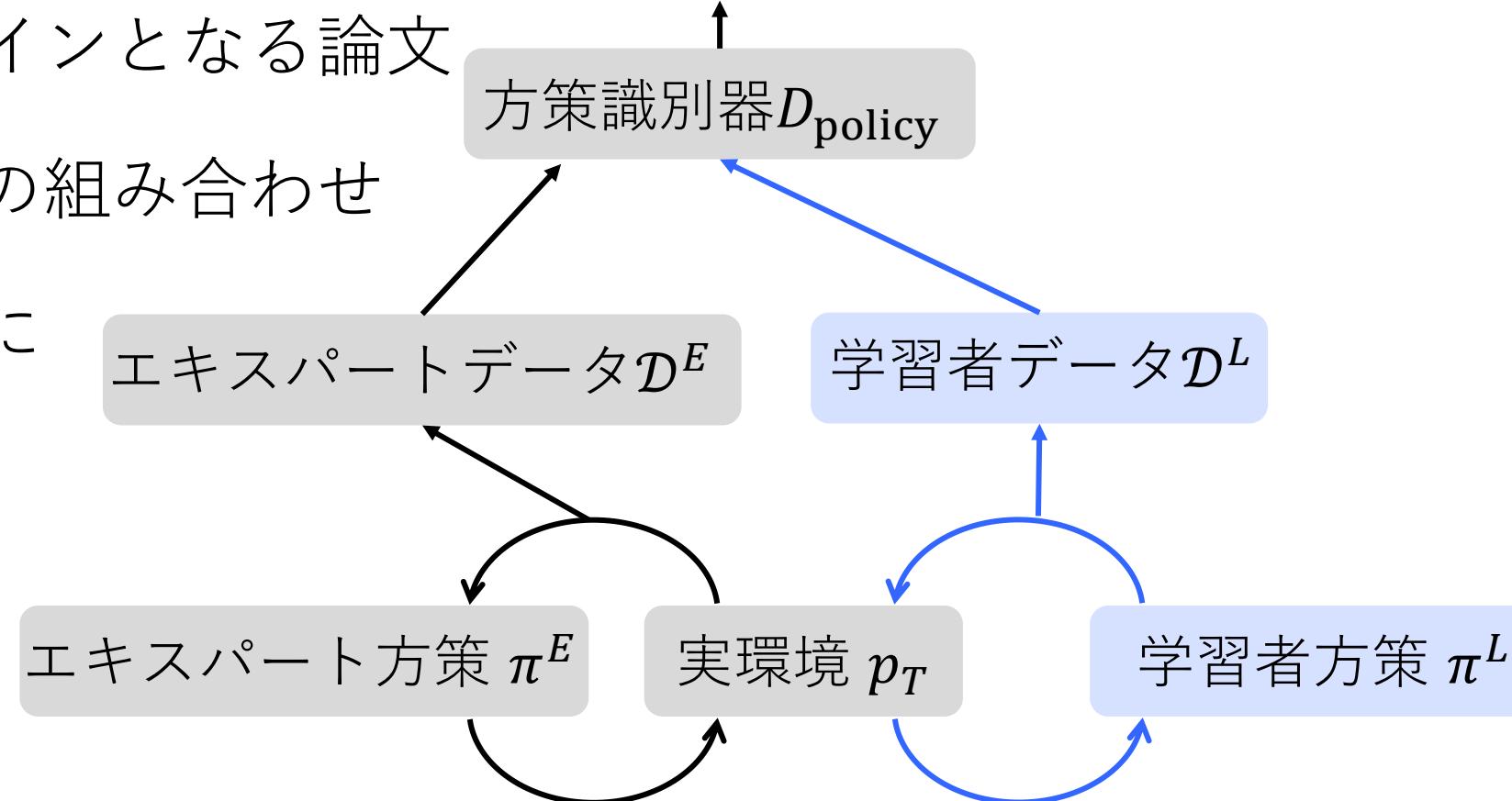
- The generator generates fake data to fool the discriminator
- The discriminator tries to find out whether data is generated or real
- The optimal discriminator is

$$D^*(x) = \frac{p^R(x)}{p^G(x) + p^R(x)}$$



敵対的生成模倣学習 (Generative Adversarial Imitation Learning; GAIL)

- 模倣学習を敵対的生成ネットワーク(GAN)として定式化
学習者 or エキスパート
- 模倣学習でベースラインとなる論文
- 生成器が方策と環境の組み合わせ
- 識別器が報酬の推定に相当



GAILの目的関数

- $D(s, a)$ は s, a がエキスパートデータか生成されたデータか判定

$$D(s, a) = \begin{cases} 1 & (s, a) \text{が実データ} \\ 0 & (s, a) \text{が生成データ} \end{cases}$$

– 定義は論文によって異なるので注意

- 目的関数

$$\min_{\pi} \max_D \mathbb{E}_{(s,a) \sim \pi^E} [\ln(1 - D(s, a))] + \mathbb{E}_{(s,a) \sim \pi} [\ln(D(s, a))] - \lambda \mathcal{H}(\pi)$$

– $\mathbb{E}_{(s,a) \sim \pi^E} [\cdot]$ は $\pi^E(a | s)$ のもとで得られる
state-action-visitation分布のもとでの期待値

$$\pi^E(s, a) = \pi^E(a | s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi^E)$$

識別器の目的関数

- D は J^D を最大化

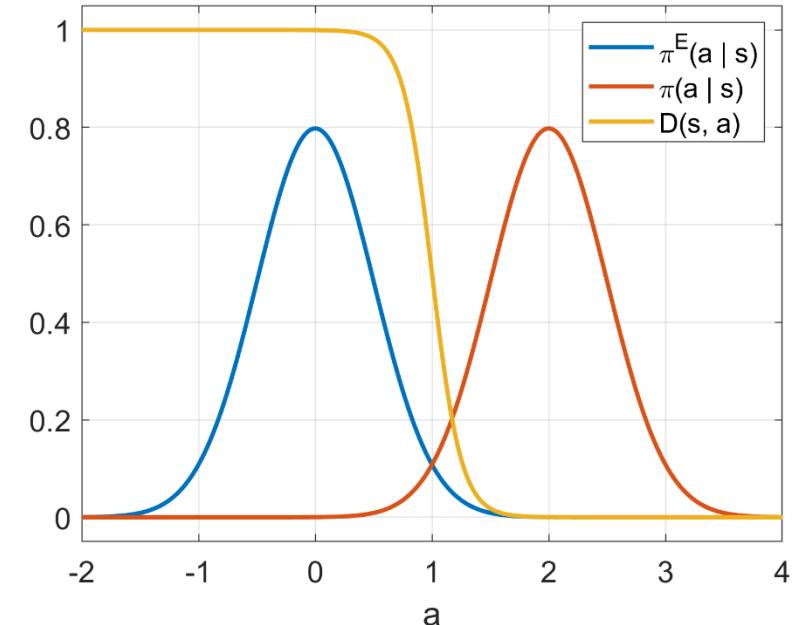
$$J^D = \mathbb{E}_{(s,a) \sim \pi^E} [\ln(1 - D(s, a))] + \mathbb{E}_{(s,a) \sim \pi} [\ln(D(s, a))]$$

- 最適な識別器 (Goodfellow et al., 2004)

$$D^*(s, a) = \frac{\pi(a | s)}{\pi^E(a | s) + \pi(a | s)}$$

$$J^D(D^*) = JS(\pi \| \pi^E) - 2 \ln 2$$

- 識別器はエキスパートと学習者の方策を Jensen-Shannon ダイバージェンスで近似



生成器の学習の補足

- 生成器の目的関数

$$\min_{\pi} J^{\pi}(\pi), J^{\pi}(\pi) = \mathbb{E}_{(s,a) \sim \pi} [\ln(D(s,a))]$$

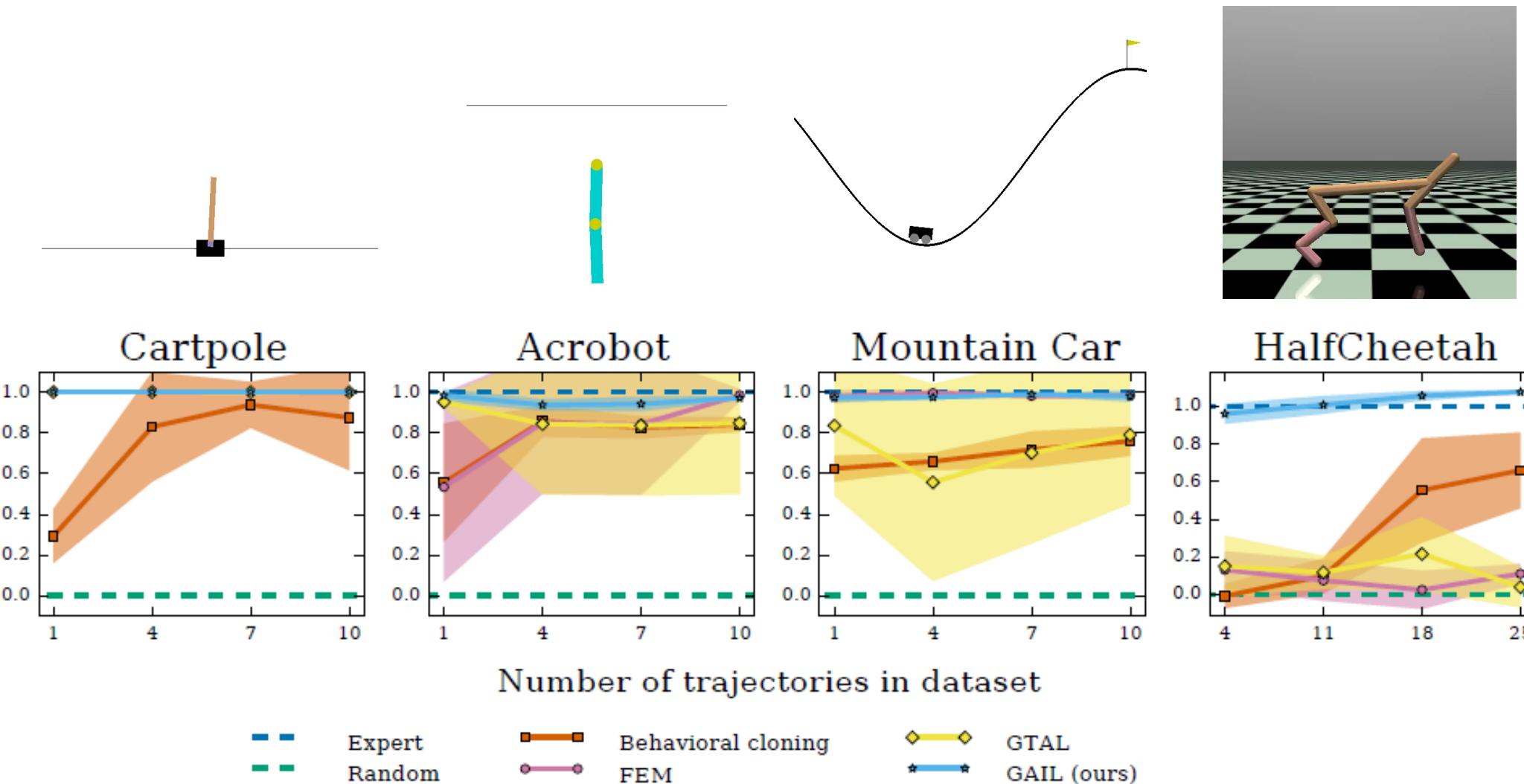
– ただし $(s, a) \sim \pi(s, a) = \pi(a | s) \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \pi)$

- 識別器から計算される報酬に対して強化学習を実行して方策 π を求める

$$\max_{\pi} \mathbb{E}_{(s,a) \sim \pi} [r(s,a)], \quad r(s,a) = -\ln(D(s,a))$$

– Ho and Ermon (2016) では Trust Region Policy Optimization (TRPO) (Schulman, et al., 2015) を使用

エキスパートデータ数に対する性能比較

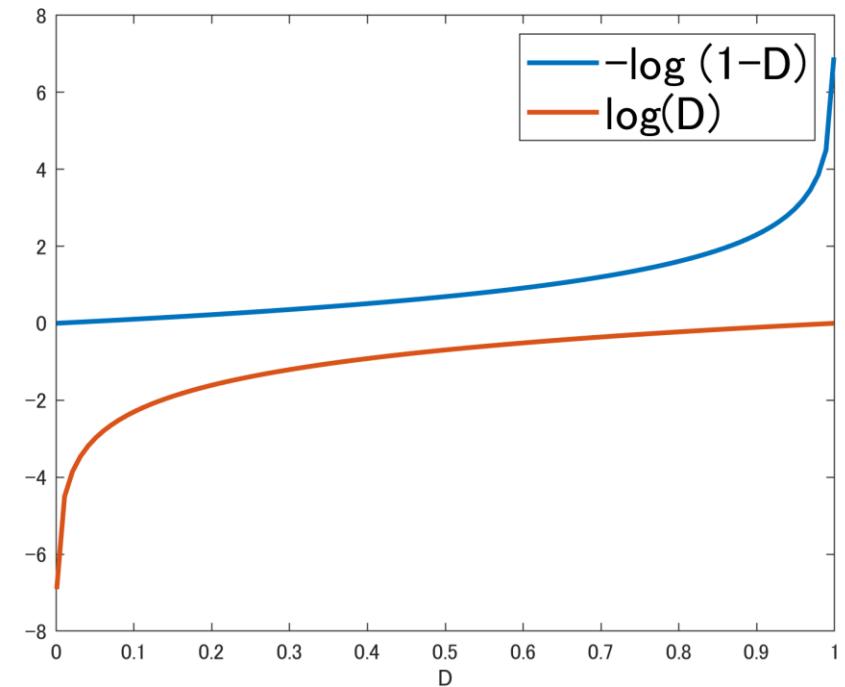


GAILの様々な拡張

- 行動が観測できない場合への対応 IRLGAN (Henderson et al., 2018), AGAIL (Sun & Ma, 2019), GAIffO (Torabi, et al., 2019)
- マルチタスクへの応用 OptionGAN (Henderson et al., 2018), InfoGAIL (Li, et al., 2017)
- サンプル効率の改善 DAC (Kostrikov, et al., 2019), (Sasaki et al., 2019), SAM (Blondé & Kalousis, 2019), MF-ERIL (Uchibe and Doya, 2021)
- モデルベース MGAIL (Baram et al., 2017), MB-ERIL (Uchibe, 2022)
- 識別器の構造化 AIRL (Fu, et al., 2018), LogReg-IRL (Uchibe, 2018)

GAILの問題点

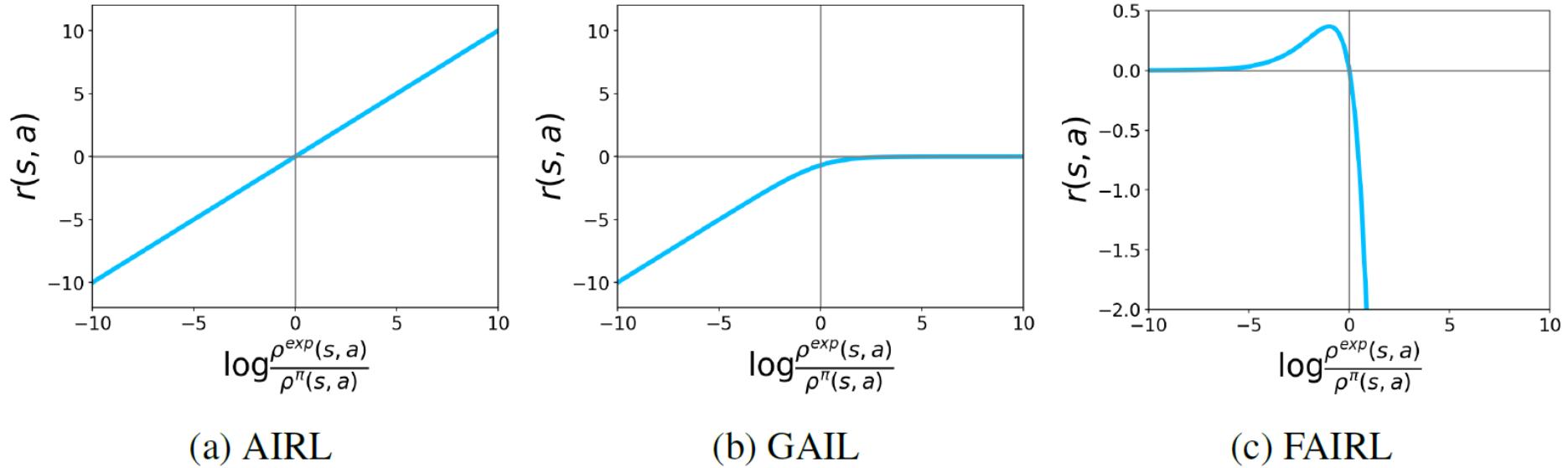
- 方策オン型強化学習(TRPOやProximal Policy Optimization)によって方策を学習するため、相互作用回数に関してサンプル効率は悪い
 - 方策オフ型にすることで、ある程度の改善は可能(Kostrikov et al., 2019; Blondé and Kalousis, 2019)
- $\ln D(s, a)$ は偏りのある報酬になっている
(Kostrikov et al., 2019)
 - 識別器の出力は $[0, 1]$ であるため報酬は実数の全区間を表現しない
$$r(s, a) = -\ln(D(s, a))$$



f-divergenceによる一般化

Method	objective
Behavior Cloning	$\mathbb{E}_{\pi_E(s)}[\text{KL}(\pi_E(a s) \parallel \pi(a s))] = -\mathbb{E}_{\pi_E(s,a)}[\ln \pi(a s)] + C$
DAgger (Ross et al., 2011)	$\mathbb{E}_{p_{agg_{1:n}}}[\text{KL}(\pi_E(a s) \parallel \pi(a s))]$ at iteration $n+1$
AIRL (Fu et al., 2018)	$\text{KL}(\pi(s,a) \parallel \pi_E(s,a)) = -\mathbb{E}_{\pi(s,a)}[\ln \pi_E(s,a)] - \mathcal{H}(\pi(s,a))$
GAIL (Ho and Ermon, 2016)	$D_{\text{JS}}(\pi(s,a) \parallel \pi_E(s,a)) - \mathcal{H}(\pi)$
FAIRL (Ghasemipour et al., 2019)	$\text{KL}(\pi_E(s,a) \parallel \pi(s,a)) = -\mathbb{E}_{\pi_E(s,a)}[\ln \pi(s,a)] - \mathcal{H}(\pi_E(s,a))$
Symmetric f-div (Ho and Ermon, 2016)	$D_{f-\text{symm}}(\pi(s,a) \parallel \pi_E(s,a)) - \mathcal{H}(\pi)$
f-MAX (Ghasemipour et al., 2019)	$D_f(\pi(s,a) \parallel \pi_E(s,a))$

f-divergenceによる一般化



Method	Halfcheetah		Ant		Walker		Hopper	
	Det	Stoch	Det	Stoch	Det	Stoch	Det	Stoch
BC	-62 ± 182	-126 ± 218	82 ± 124	19 ± 70	1804 ± 1286	1293 ± 480	1435 ± 78	764 ± 129
AIRL	8043 ± 237	7377 ± 482	6024 ± 155	4598 ± 65	3979 ± 323	3846 ± 319	3393 ± 7	2561 ± 331
FAIRL	7924 ± 318	7453 ± 640	6607 ± 139	5525 ± 287	4297 ± 71	4225 ± 34	3379 ± 10	3061 ± 170
BC	641 ± 70	285 ± 166	258 ± 292	23 ± 69	656.4 ± 72	594 ± 37	2543 ± 328	1673 ± 375
AIRL	8132 ± 143	6914 ± 313	5811 ± 208	5027 ± 287	4499 ± 68	4355 ± 92	3417 ± 4	2530 ± 260
FAIRL	8275 ± 24	7900 ± 25	6267 ± 312	5473 ± 73	4824 ± 3	4778 ± 13	3429 ± 27	3335 ± 38
BC	872 ± 640	302 ± 288	147 ± 59	94 ± 11	726 ± 33	578 ± 25	2253 ± 433	1135 ± 407
AIRL	8347 ± 37	7061 ± 324	5984 ± 58	4406 ± 506	4433 ± 166	4284 ± 218	3425 ± 14	2524 ± 363
FAIRL	8302 ± 15	7522 ± 406	6365 ± 128	5442 ± 147	4807 ± 6	4764 ± 28	3428 ± 27	3415 ± 21
DAgger	8418 ± 14	6646 ± 1209	6978 ± 11	6011 ± 201	4874 ± 34	4071 ± 1073	3460 ± 5	2962 ± 157

まとめ

- 報酬モデルの学習, 最大エントロピ逆強化学習, 敵対的生成模倣学習の紹介
 - 背後にはエントロピ正則強化学習
- GAILを起点として様々なアルゴリズムが提案
 - f -divergenceに一般化することで多様なアルゴリズムが導出できる
 - 近年は, 方策への正則化をエントロピではなく, 凸関数をつかった一般化も可能 (Jeon et al., 2020)

最大エントロピ逆強化学習の別表現

- 対数尤度の勾配

$$\nabla_w \ln \Pr(\tau | s_0) = \mathbb{E}_{\pi^E}[\nabla_w G(w)] - \mathbb{E}_{\Pr(\tau|s_0)}[\nabla_w G(w)]$$

- 実は最大エントロピ逆強化学習は次のように定式化できる

$$L = \max_r \left\{ J(r, \pi_E) - \max_{\pi} J(r, \pi) \right\}$$

$$J(r, \pi) = \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t (r(s_t, a_t) + \mathcal{H}(\pi(\cdot | s_t))) \right]$$