DEEP LEARNING JP
[DL Papers]

強化学習の再定式化について:
Beyond Reward Based End-to-End RL

Presenter: Yusuke Iwasawa, Matsuo Lab

概要

- 通常の強化学習：期待報酬を最大化するような方策を得ることが目的
  - Optimization：REINFORCE
  - Dynamic Programing：Q学習，TD Learning

- 問題点
  - 報酬だけからの学習は多くの場合困難（特にスパースリワード）
  - サンプル効率が悪い
  - 汎化性能も一般には高くない（特定タスクを解くだけでも大量のサンプルが必要）
  - （総括すると）新しいタスクに簡単に適用できるような状況ではない

- 報酬最大化という枠組み自体の再検討（再定式化）が必要では？
  - => トピック1：表現学習の活用
  - => トピック2：データ最適化（＋教師あり学習）

本発表に関係する文献

表現学習

- “CURL: Contrastive Unsupervised Representations for Reinforcement Learning”, ICML2020
- “Decoupling Representation Learning from Reinforcement Learning”, ICLR2021 (Under review)
- “Unsupervised State Representation Learning in Atari”, NeurIPS2019
- “Dynamics-Aware Embeddings”, ICLR2020
- “Deep Reinforcement and Infomax Learning”, NeurIPS2020
- “Data-Efficient Reinforcement Learning with Self-Predictive Representations”, ICLR2021 (Under review)

データセット最適化（１つ目メイン）

- “Reinforcement learning is supervised learning on optimized data”, Berkeley Blog
- “Hindsight Experience Replay”, NIPS2017
- “Training Agents using Upside-Down RL”, arxiv
- “Reward-Conditioned Policies”, arxiv
- “Rewriting History with Inverse RL: Hindsight Inference for Policy Improvement”, NeurIPS2020

前提知識のリンク集

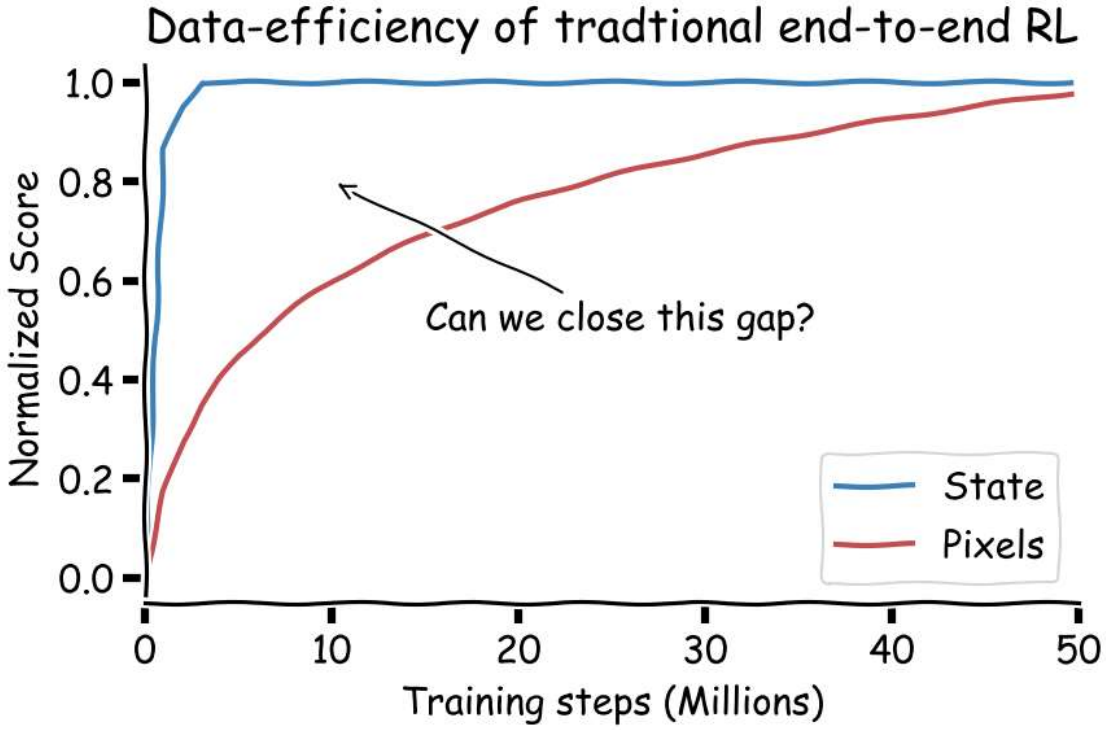
（知らなくてもある程度わかるように話す予定です）

- 通常の強化学習
  - 強化学習の基礎と深層強化学習
- 教師なし学習（自己教師あり学習）の進展
  - [DL輪読会]相互情報量最大化による表現学習
  - 最近の自己教師あり学習とcontrastive learningについて

表現学習の活用について

全体像

- 通常の強化学習
  - 報酬を最大化
  - End-to-Endで最適化
  - Pixelからの学習は難しい（右図）
- 表現学習を活用
  - 報酬以外の学習信号を活用
  - c.f. 世界モデル系の研究

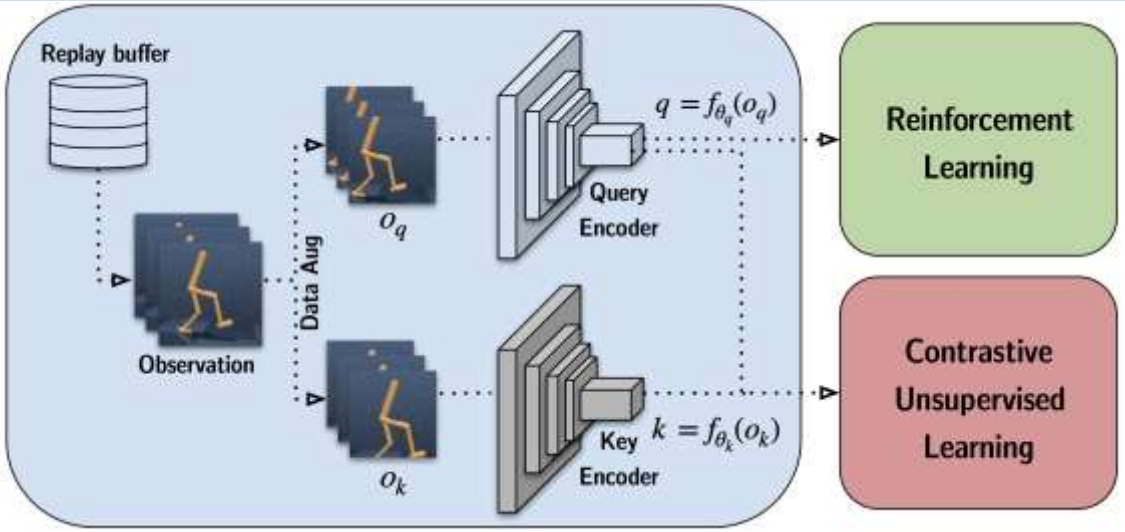


https://bair.berkeley.edu/blog/2020/07/19/curl-rad/

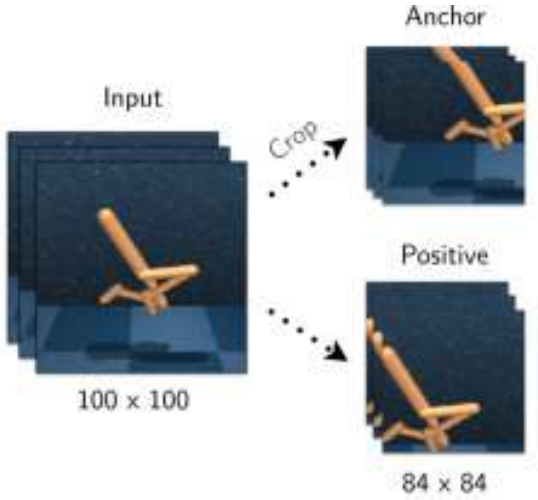
CURL: Contrastive Unsupervised Representations for Reinforcement Learning (ICML2020)

Srinivas, Aravind, Michael Laskin, and Pieter Abbeel

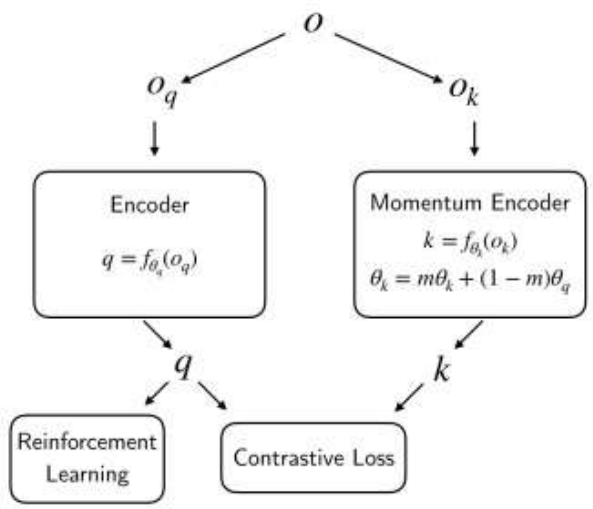
全体像



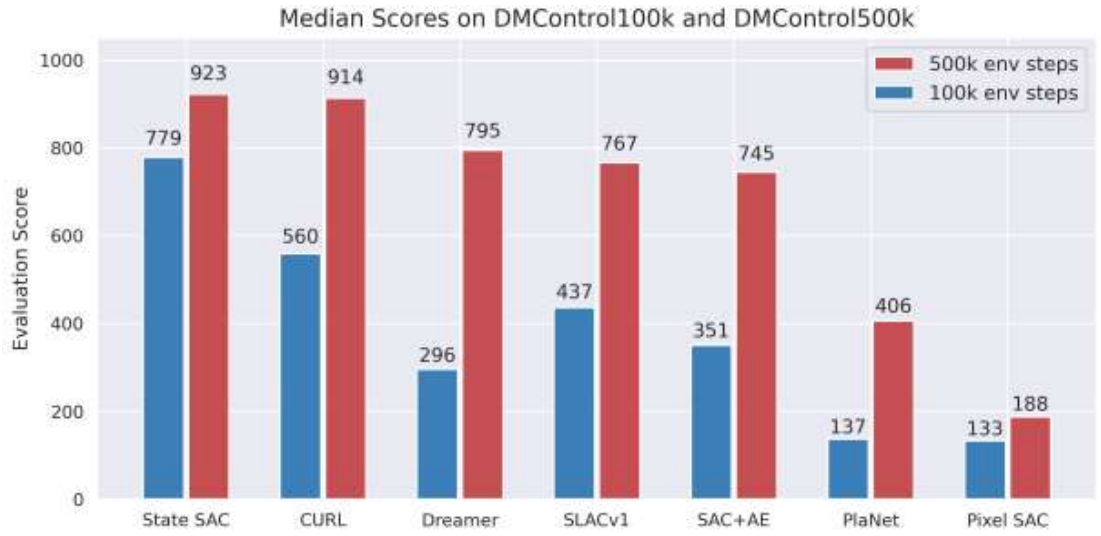
正例の作り方



Momentum Encoder



- 対照推定による損失を追加（CPCと同じinfoNCE）
  - 正例：同じ画像に異なるデータ拡張
  - 負例：異なるデータ
- Momentum Encoderを利用
- RLアルゴリズムはタスク依存で選べる
- 画像ベースでやるより大幅に良い
- DreamerやSLACなどよりもよい



CURLの定性的な結果

We show that pixel-based RL with CURL nearly the matches data-efficiency of RL from state

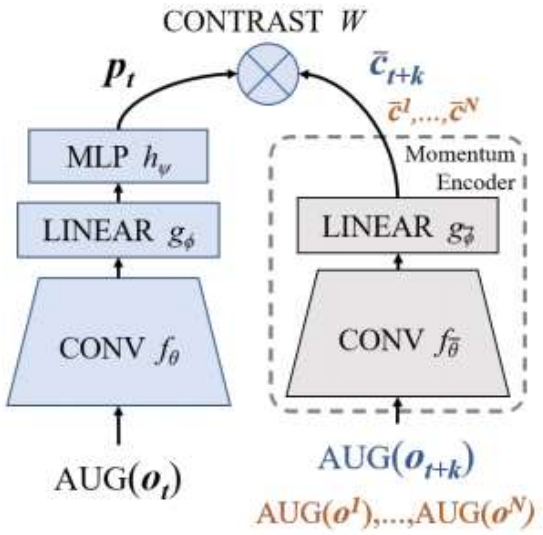
CURL on Atari

Game	Human	Random	Rainbow	SimPLe	OTRainbow	Eff. Rainbow	CURL
Alien	7127.7	227.8	318.7	616.9	**824.7**	739.9	558.2
Amidar	1719.5	5.8	32.5	88.0	82.8	**188.6**	142.1
Assault	742.0	222.4	231	527.2	351.9	431.2	**600.6**
Asterix	8503.3	210.0	243.6	**1128.3**	628.5	470.8	734.5
Bank Heist	753.1	14.2	15.55	34.2	**182.1**	51.0	131.6
Battle Zone	37187.5	2360.0	2360.0	5184.4	4060.6	10124.6	**14870.0**
Boxing	12.1	0.1	-24.8	**9.1**	2.5	0.2	1.2
Breakout	30.5	1.7	1.2	**16.4**	9.84	1.9	4.9
Chopper Command	7387.8	811.0	120.0	**1246.9**	1033.33	861.8	1058.5
crazy_climber	35829.4	10780.5	2254.5	**62583.6**	21327.8	16185.3	12146.5
demon_attack	1971.0	152.1	163.6	208.1	711.8	508.0	**817.6**
freeway	29.6	0.0	0.0	20.3	25.0	**27.9**	26.7
frostbite	4334.7	65.2	60.2	254.7	231.6	866.8	**1181.3**
gopher	2412.5	257.6	431.2	771.0	**778.0**	349.5	669.3
hero	30826.4	1027.0	487	2656.6	6458.8	**6857.0**	6279.3
jamesbond	302.8	29.0	47.4	125.3	112.3	301.6	**471.0**
kangaroo	3035.0	52.0	0.0	323.1	605.4	779.3	**872.5**
krull	2665.5	1598.0	1468	**4539.9**	3277.9	2851.5	4229.6
kung_fu_master	22736.3	258.5	0.	**17257.2**	5722.2	14346.1	14307.8
ms_pacman	6951.6	307.3	67	**1480.0**	941.9	1204.1	1465.5
Pong	14.6	-20.7	-20.6	**12.8**	1.3	-19.3	-16.5
Private Eye	69571.3	24.9	0	58.3	100.0	97.8	**218.4**
Qbert	13455.0	163.9	123.46	**1288.8**	509.3	1152.9	1042.4
Road_Runner	7845.0	11.5	1588.46	5640.6	2696.7	**9600.0**	5661.0
seaquest	42054.7	68.4	131.69	**683.3**	286.92	354.1	384.5
Up_n_Down	11693.2	533.4	504.6	**3350.3**	2847.6	2877.4	2955.2

“Decoupling Representation Learning from Reinforcement Learning” (ICLR2021 Under Review)

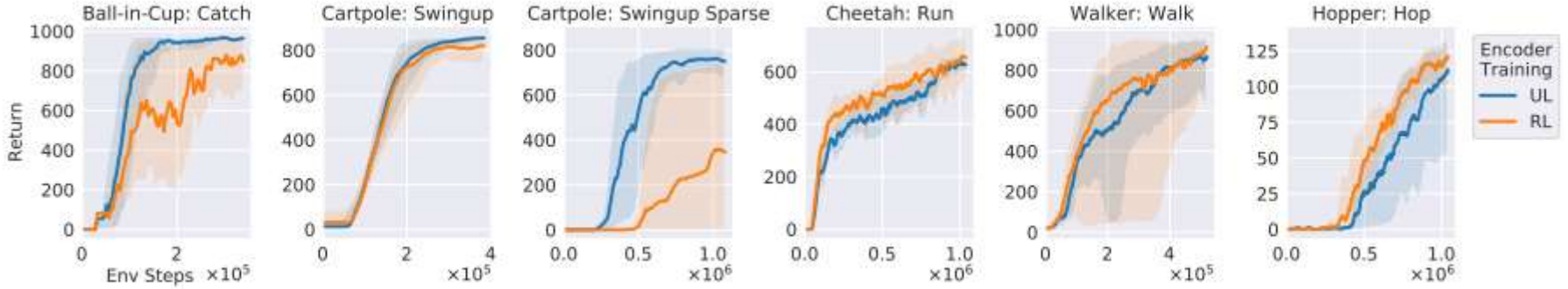
Stooke, Adam, Kimin Lee, Pieter Abbeel, and Michael Laskin

手法

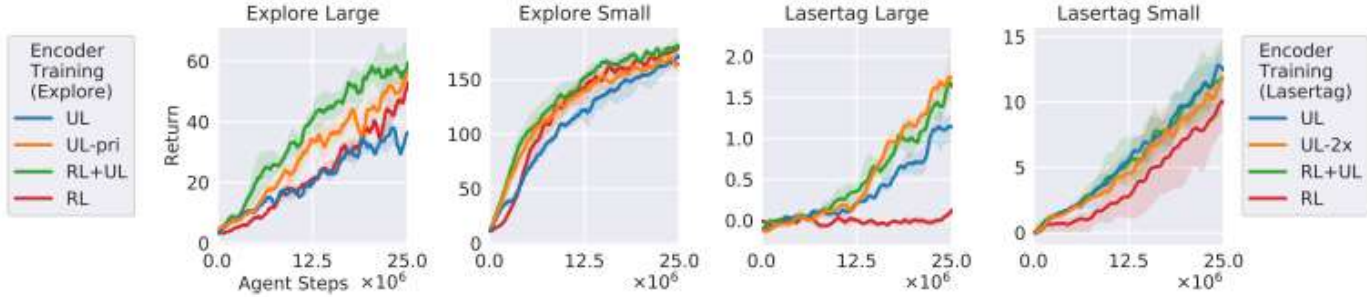


- 時間方向を考慮した対照推定(infoNCE)
- Momentum Encoderを利用
- Residual MLP:$p_t = h_\psi(c_t) + c_t$
- SACやPPOと組み合わせ

結果：DMControl

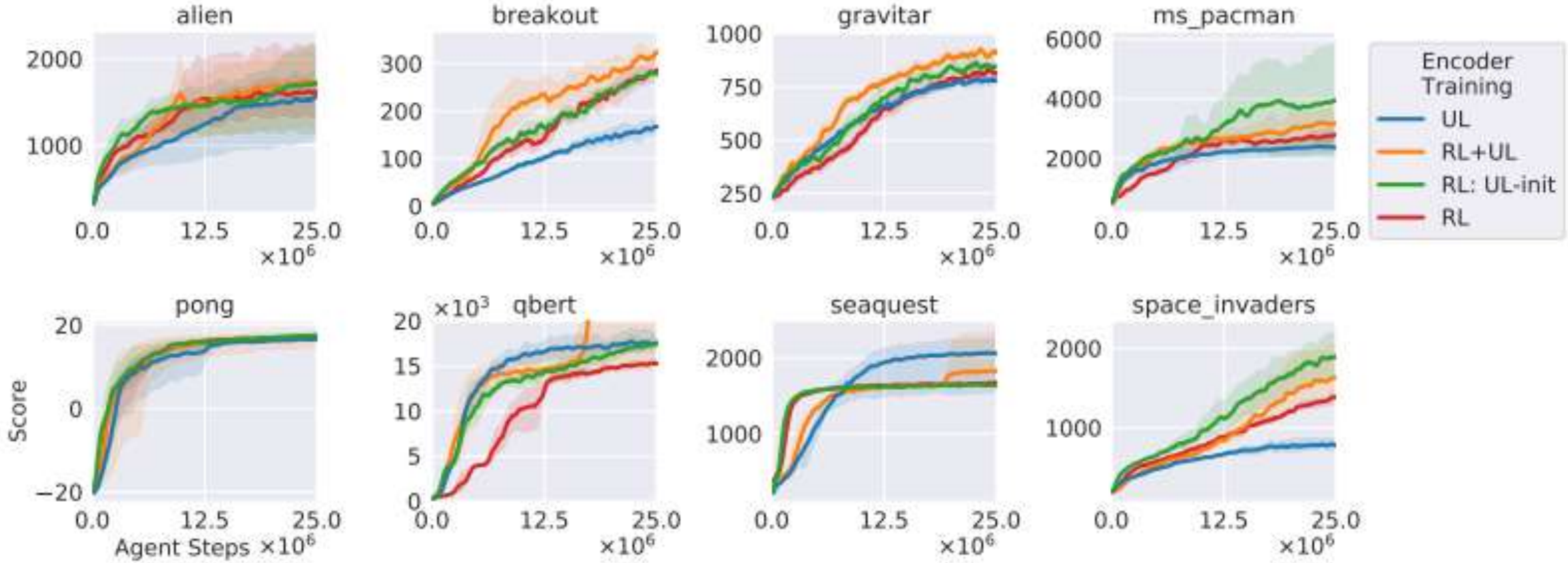


結果：DMLab



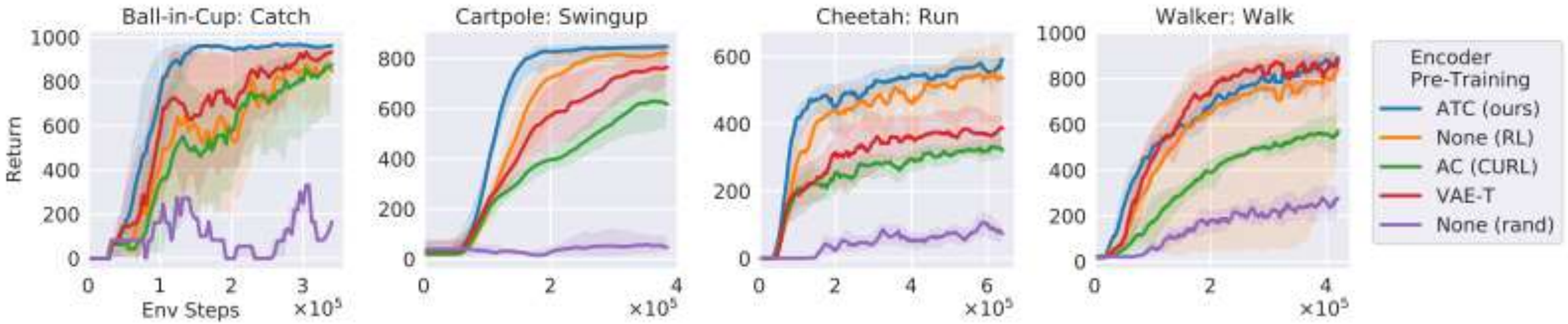
- 様々な環境でUL（エンコーダを対照推定だけで学習）がRL（エンコーダを報酬で学習した場合）と比べて同程度
  => 報酬なしでエンコーダを学習できた（新規の主張）
- DMLabでは細かい工夫を入れている

Atariでの結果



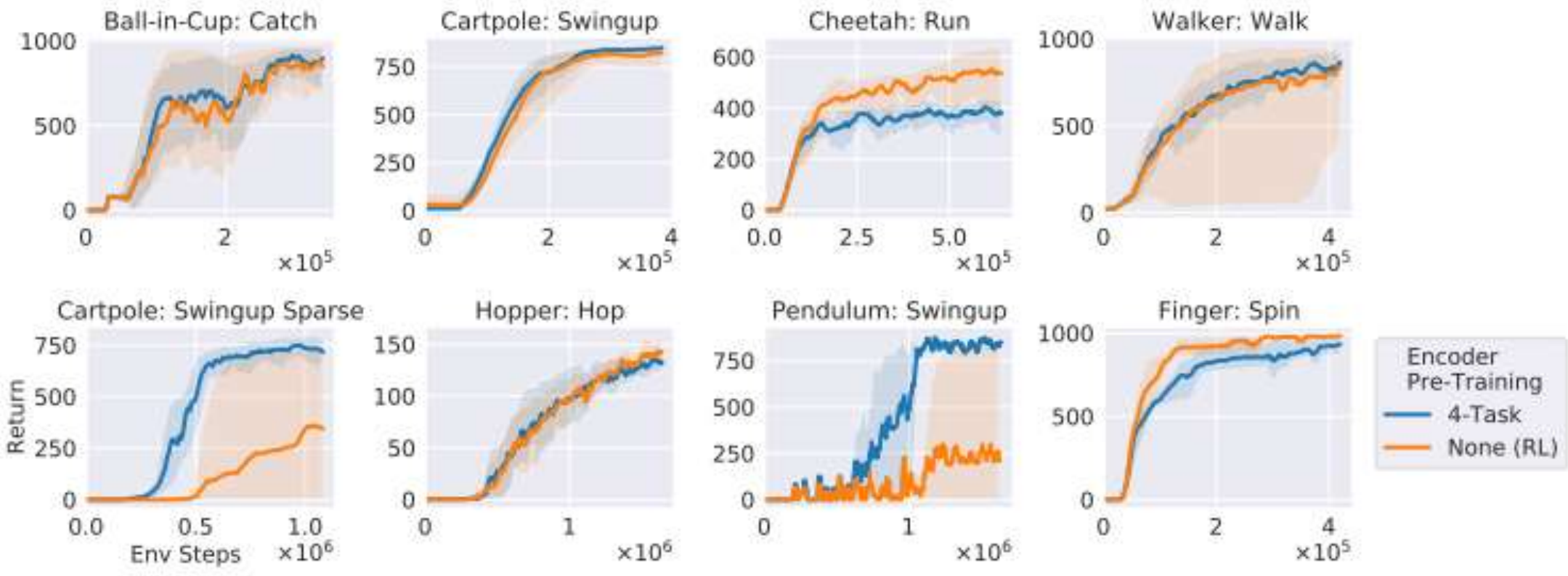
- AtariでもULは多くの場合うまくいくが，breakoutやspace invadersでは微妙
  - ただし，補助タスクとして使うとうまくいく（RL+UL）
- 初期値の学習としてULを使うことも効果的

Encoder Pre-training (DMControl)



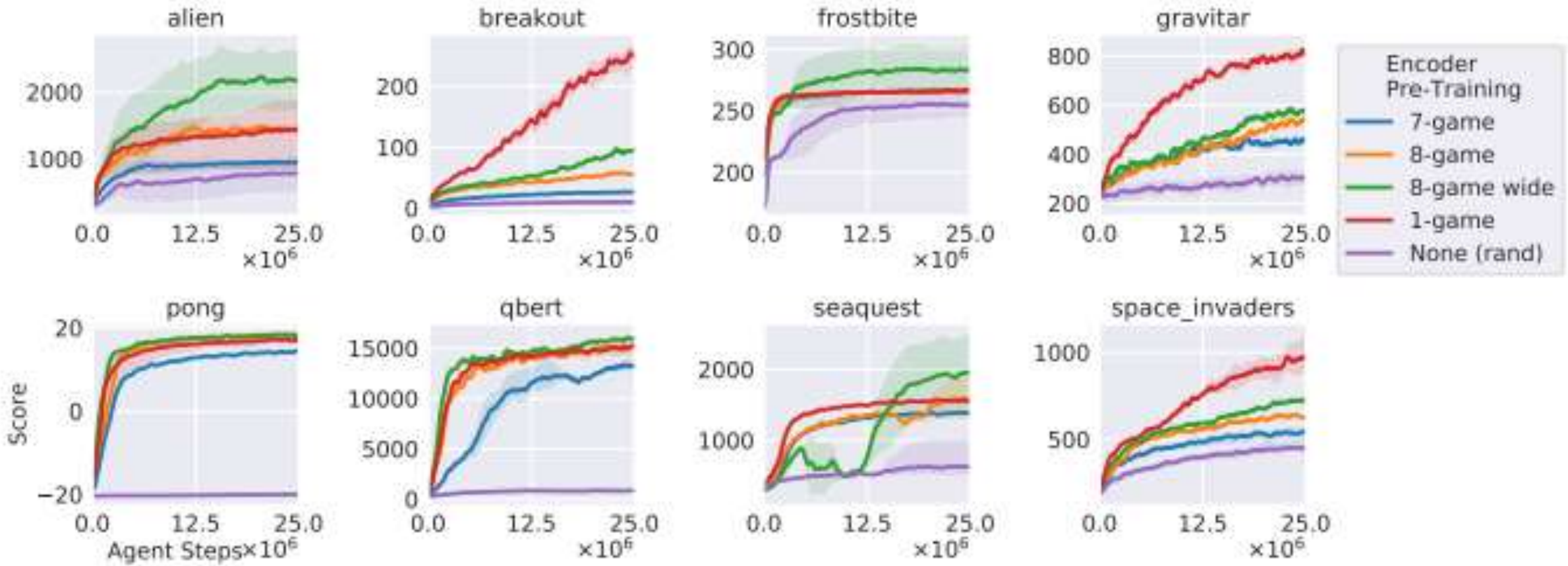
- エキスパート軌道（オフラインデータ）を使った事前学習における比較
  - オフラインデータを使ってエンコーダを事前学習し，RLの学習時にはフリーズする
- CURLやVAE-t（多分先を予測するVAE）と比較しても提案法が良いことが多い
  => 手法としてもおそらく良いものになっている

Multi-Task Encoders (DMControl)



- 上4つの環境のみでエンコーダを事前学習して別のタスクに転移
  - つまり，HopperやPendulum，Fingerはエンコーダの学習に利用していない
- なんと転移できる（！！！）
  - 特に報酬がスパースな場合，RLより良い

Multi-Task Encoders (Atari)



- Atariでは必ずしもうまくいくわけじゃない
  - 7-game: 自分以外の7個の環境で事前学習，8-game: 前環境で事前学習，1-game: 同じ環境で事前学習
- 転移できるものとできないものがある（当たり前だが）
- エンコーダを大きくするとある程度うまくいく（8-game wide）

まとめとその他の研究の簡単な紹介

- 強化学習に表現学習を組み込む動き
  - 報酬だけでなく，よい状態を学習してから学習する
  - 報酬をエンコーダの学習に使わなくても報酬ベースと同等以上の性能が出るような報告も

- “Unsupervised State Representation Learning in Atari”, NeurIPS2019
  - 時間方向と空間方向どちらにも対照推定
  - Decouple論文でAtariの方でベンチマークとして使われている（ACTの方が良い）
- “Dynamics-Aware Embeddings”, ICLR2020
  - 将来を予測できるような抽象状態と抽象行動を学習，それを方策の学習に使う
- “Deep Reinforcement and Infomax Learning”, NeurIPS2020
  - 将来を予測するように（将来の状態についての情報をよく持つように）学習する
  - 未知の環境への適応可能性が上がる（継続学習ができる）．PacManで検証．
  - 複雑なタスクで性能が上がる．Procgenで検証．
- “Data-Efficient Reinforcement Learning with Self-Predictive Representations”, ICLR2021 (Under review)
  - 対照推定ではなく素直な先読み
  - BYOL版という感じ（素直でよさそう）

データ最適化 (+ 教師あり学習)

元記事

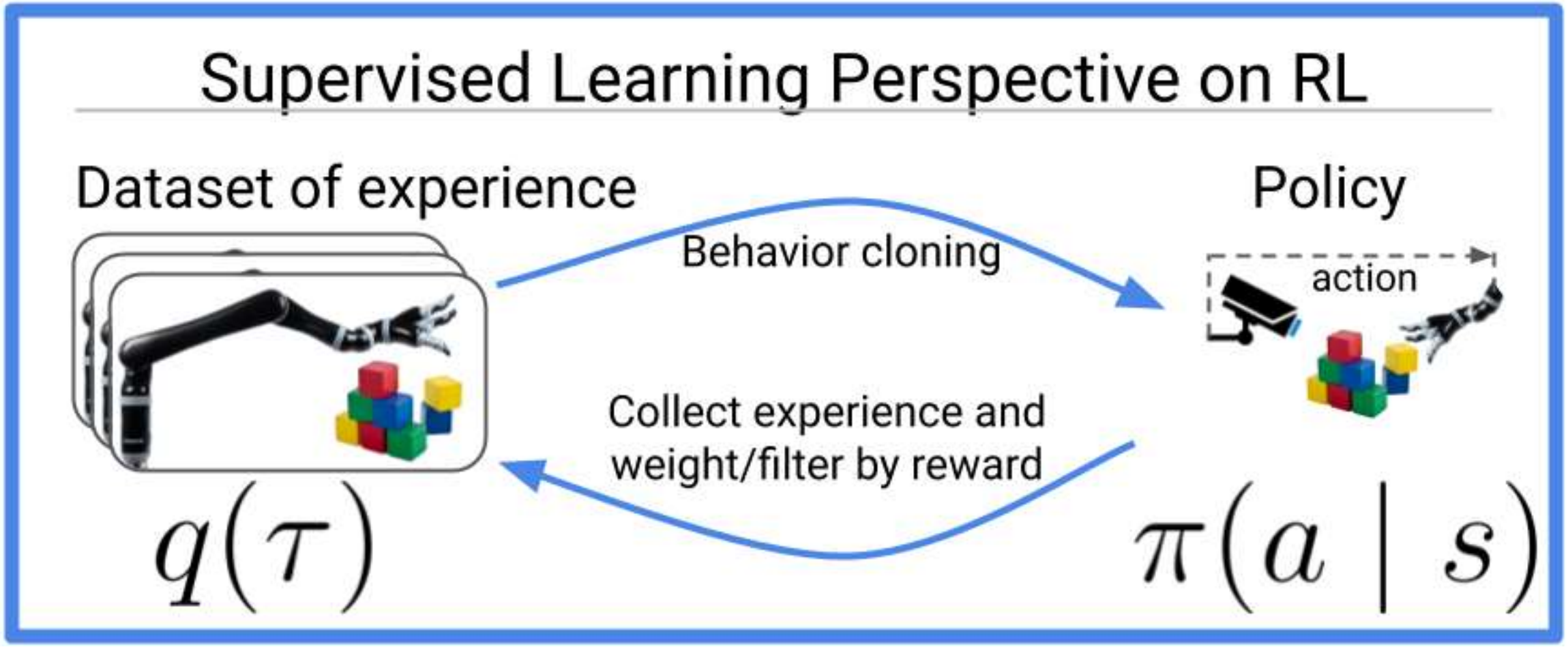
Reinforcement learning is supervised learning on optimized data

Ben Eysenbach and Aviral Kumar and Abhishek Gupta Oct 13, 2020

The two most common perspectives on Reinforcement learning (RL) are **optimization** and **dynamic programming**. Methods that compute the gradients of the non-differentiable expected reward objective, such as the REINFORCE trick are commonly grouped into the optimization perspective, whereas methods that employ TD-learning or Q-learning are dynamic programming methods. While these methods have shown considerable success in recent years, these methods are still quite challenging to apply to new problems. In contrast deep supervised learning has been extremely successful and we may hence ask: *Can we use supervised learning to perform RL?*

https://bair.berkeley.edu/blog/2020/10/13/supervised-rl/

教師あり学習としての強化学習：概念的な説明



教師あり学習としての強化学習：形式的な説明

期待報酬最大化の下界

$$\log J(\theta) = \log \mathbb{E}_{\pi(\tau)}\left[R(\tau)\right]$$
$$\geq \mathbb{E}_{q(\tau)}\left[\log R(\tau) + \log \pi_\theta(\tau) - \log q(\tau)\right] := F(\theta, q)$$

※ イェンセンの不等式

方策の改善フェーズ

$$\max_\theta F(\theta, q) = \max_\theta \mathbb{E}_{\tau\sim q(\tau)}\left[\sum_{s_t, a_t \in \tau} \log \pi_\theta(a_t \mid s_t)\right] + \text{const.}$$

与えられた軌道の元での教師あり学習

データの最適化

$$\max_q F(\theta, q) = \max_q \mathbb{E}_{q(\tau)}\left[\log R(\tau)\right] - D_{\mathrm{KL}}\left(q(\tau) \,\|\, \pi(\tau)\right).$$

τ~q() が高い報酬をとるようにqを変更
（２項目は方策からの軌道との乖離を防ぐ役割）

通常の強化学習の考え方との比較

	Optimization Perspective	Dynamic Programming Perspective	Supervised Learning Perspective
What are we optimizing?	policy (π_θ)	Q-function (Q_θ)	policy (π_θ) and data ($q(\tau)$)
Loss	Surrogate loss $\tilde{L}(\theta, \tau \sim \pi_\theta)$	TD error	Lower bound $F(\theta, q)$
Data used in loss	collected from current policy	arbitrary	optimized data

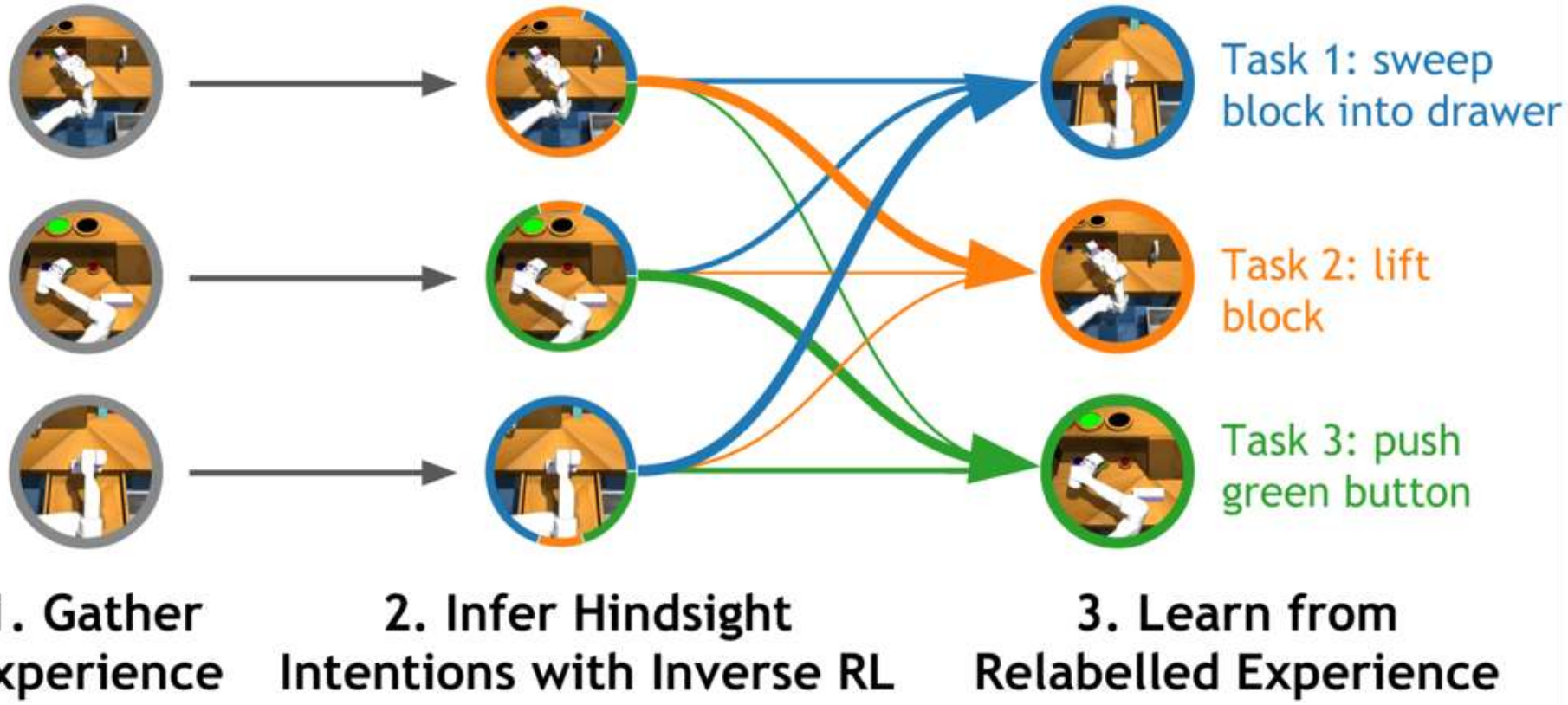
データ分布を最適化する方法

- アプローチ
  - 報酬の高い系列のみを残す
  - 軌道最適化
  - 報酬で重みづけ

- q自体も様々な形で表すことができる
  - 経験分布，生成モデル，etc…

- 様々な手法がこの枠組みをとっているとみなすことができる
  - Advantage Weighted Regression, Self-Imitation Learning, MPO
  - （手法の詳細追ってないので割愛します）

最近の流れ：マルチタスク化

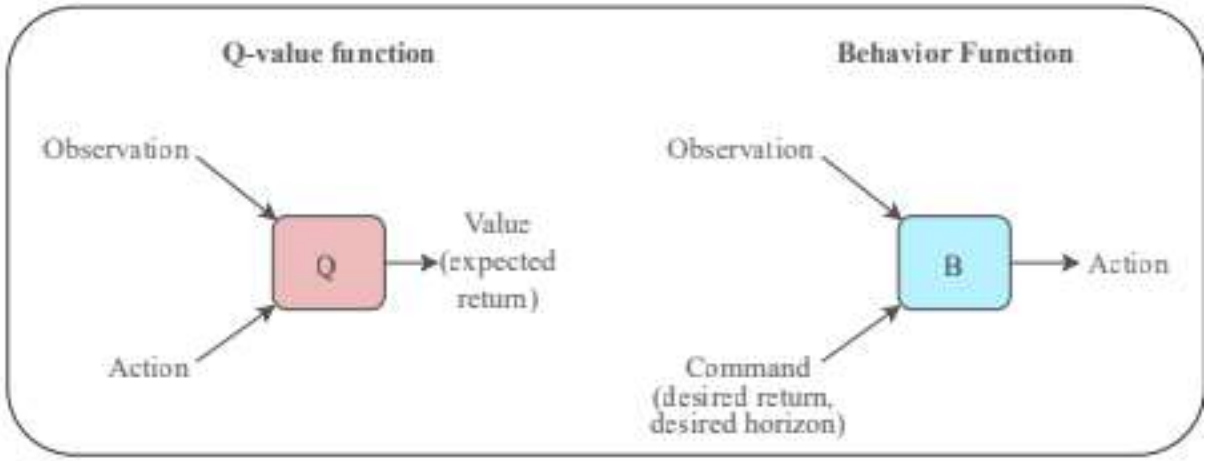


- 得られた軌道が達成しているゴールを，達成したかったゴールだとみなす
- すると，マルチタスクな状況ではある軌道が別のタスクにとって役立つということが起こる（つまり，Relabelingにより$q(\tau)$を最適化できる
- 参考：Hindsight Experience Replay（DL輪読会 中村君資料，松嶋君資料）

マルチタスクの考え方は単一タスクのRLにも拡張できる

Algorithm 1 Generic Algorithm for Reward-Conditioned Policies (RCPs)

$\theta_1 \leftarrow$ random initial parameters
$\mathcal{D} \leftarrow \emptyset$
$\hat{p}_1(Z) \leftarrow$ initial value distribution
for iteration $k = 1, \ldots, k_{\max}$ **do**
  sample target value $\hat{Z} \sim \hat{p}_k(Z)$.
  roll out trajectory $\tau = \{\mathbf{s}_t, \mathbf{a}_t, r_t\}_{t=0}^{T}$, with policy $\pi_{\theta_k}(\cdot|\mathbf{s}_t, \hat{Z})$
  for each step t, label $(\mathbf{s}_t, \mathbf{a}_t)$ with observed value Z_t
  store tuples $\{\mathbf{s}_t, \mathbf{a}_t, Z_t\}_{t=0}^{T}$ in $\mathcal{D}$
  $\theta_{k+1} \leftarrow \arg\max_\theta \mathbb{E}_{\mathbf{s},\mathbf{a},Z\sim\mathcal{D}}[\log \pi_\theta(\mathbf{a}|\mathbf{s}, Z)]$
  $\hat{p}_{k+1} \leftarrow$ update target value distribution using $\mathcal{D}$
end for



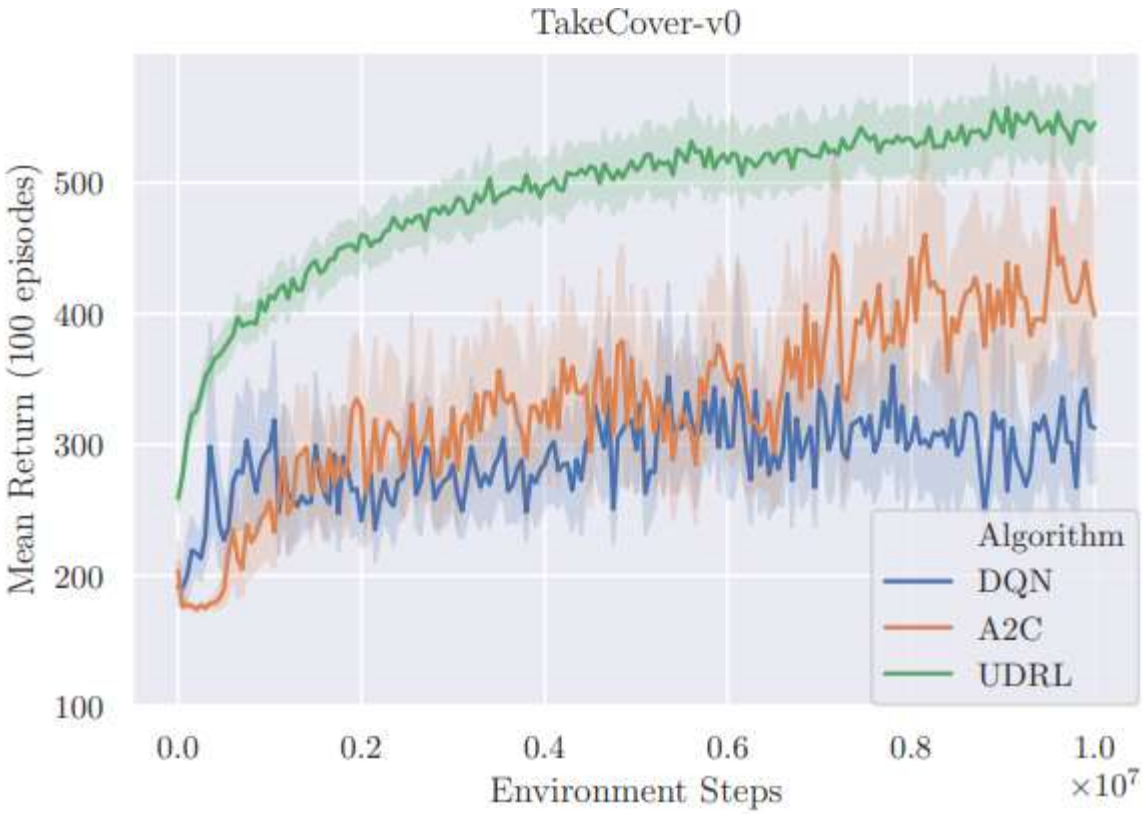
“Reward-Conditioned Policies”

- 報酬で条件づけた方策を作る
- 達成した報酬を，達成したかった報酬だと考える
  - すると，ある報酬を達成する軌道が手に入る
- あとはこの軌道を達成するような方策を学習すればよい

“Training Agents using Upside-Down Reinforcement Learning”

- 報酬とそれを達成する時間（軌道の長さ）で条件づける
- つまり報酬を予測するのではなくコマンドとして使う
- あとは同様

結果



- Upside Downの論文より
- 報酬がスパースな場合に特にうまくいく

後半まとめ

- 強化学習を，教師あり学習の観点で再解釈する試み
  - 強化学習は最適化されたデータの上で教師あり学習をしている
- 特に，再ラベリングによりデータ分布を最適化するような方法が出現している
  - Hindsight Experience Replay
  - Generalized Hindsight Experience Replay
  - Reward Conditioned Policy

全体まとめ・感想

- 強化学習における
  - 前半：表現学習の活用
  - 後半：教師あり学習としての再解釈
- どちらも画像領域での成功をより積極的に取り入れる動き
  - 報酬がスパースな場合への対応，マルチタスク，高速な適応など様々な恩恵が得られる可能性がある

- 感想
  - 世界モデルは両方のブリッジになるかもしれない
    （モデルを学習すること自体データ分布を作っているのと同じ）
    - 例えばモデルからMPCでサンプル作るとか？
  - Control as InferenceやAction and Perception as Divergence Minimizationとの関係が気になる