Return-Based Contrastive Representation Learning for Reinforcement Learning

Guoqing Liu[1,*], Chuheng Zhang[2,*], Li Zhao[3], Tao Qin[3], Jinhua Zhu[1], Jian Li[2], Nenghai Yu[1], Tie-Yan Liu[3]

1. University of Science and Technology of China
2. IIIS, Tsinghua University
3. Microsoft Research Asia

Microsoft Research 微软亚洲研究院

清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University

Motivation

- Various **auxiliary tasks** have been proposed to accelerate **state representation learning**, thus improving sample efficiency (DRL).
- Existing auxiliary tasks **do not take the characteristics of RL problems into considerations** and **are unsupervised/self-supervised**.
- By **leveraging returns**, **the most important feedback signals in RL**, we propose a novel auxiliary task that **forces the learnt representations to discriminate state-action pairs with different returns**.

Method

1. To formally characterize the desired representation.

Z^{π}-irrelevance abstractions

2. To learn Z^{π}-irrelevance abstractions with sampled returns.

Z-Learning algorithm

3. To derive a practical algorithm with more balanced labels.

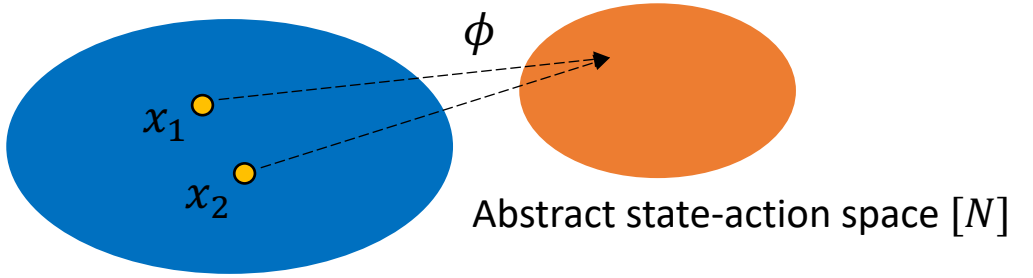
Return-based Contrastive Learning for RL (RCRL)

Z^{π}-irrelevance Abstraction

- Intuition: Leverage returns as supervision signals to design state abstractions.
- Bring us a new form of abstraction, Z^{π}-irrelevance.

  ***Definition**: Given a policy π, Z^{π}-irrelevance abstraction is denoted as $\phi: \mathcal{X} \to [N]$, for any $x_1, x_2 \in \mathcal{X}$ with $\phi(x_1) = \phi(x_2)$, we have $\mathbb{Z}^{\pi}(x_1) = \mathbb{Z}^{\pi}(x_2)$.*

- Z^{π}-irrelevance abstraction aggregates state-action pairs with similar return distributions under a certain policy π.



$$\phi(x_1) = \phi(x_2)\ implies\ Z^{\pi}(x_1) = Z^{\pi}(x_2).$$

Benefits

- Better reduction of state-action space (a coarser abstraction).

Proposition 1. *Given a policy π and the parameter for return discretization K, $N_{\pi,K}$ denotes the minimum N such that a Z^{π}-irrelevance exists, we have $N_{\pi,K} \leq N_{\pi,\infty} \leq |\phi_B(\mathcal{S})||\mathcal{A}|$ for any π and K, where $|\phi_B(\mathcal{S})|$ is the number of abstract states for the coarsest bisimulation.*

Smaller abstract state-action space.

- Approximate the Q-values arbitrarily accurately.

Proposition 2. *Given a policy π and any Z^{π}-irrelevance $\phi: \mathcal{X} \to [N]$, there exists a function $Q: [N] \to \mathbb{R}$ such that $\left|Q\left(\phi(x)\right) - Q^{\pi}(x)\right| \leq \frac{R_{max} - R_{min}}{K}, \forall x \in \mathcal{X}$.*

ϕ can sufficiently represent Q-values. When $\pi \to \pi^*$, Q-value is exactly $Q^*(s, a)$.

Z-learning

- To learn the Z^{π}-irrelevance abstractions, we propose Z-learning with a contrastive loss based on a dataset $\mathcal{D}$.

$$\min_{\phi \in \Phi_N, w \in \mathcal{W}_N} \mathcal{L}(\phi, w; \mathcal{D}) \coloneqq \mathbb{E}_{(x_1, x_2, y) \sim \mathcal{D}} \left[\left(w\left(\phi(x_1), \phi(x_2)\right) - y \right)^2 \right]$$

  - ϕ: $\mathcal{X} \to [N]$ represents the encoder (state-action representation).
  - w: $[N] \times [N] \to [0,1]$ represents the discriminator.
  - y (binary label) indicates whether x_1, x_2's sampled returns belong to same bin.

Z-learning

- Z-learning can learn Z^{π}-irrelevance abstraction provably efficiently.

Theorem 1. *Given the encoder $\hat{\phi}$ returned by Z-learning algorithm, the following inequality holds with probability $1-\delta$ and for any $x' \in \mathcal{X}$:*

$$\mathbb{E}_{x_1 \sim d, x_2 \sim d}\left[\mathbb{I}\left[\hat{\phi}(x_1) = \hat{\phi}(x_2)\right] \left| \mathbb{Z}^{\pi}(x')^{T}\left(\mathbb{Z}^{\pi}(x_1) - \mathbb{Z}^{\pi}(x_2)\right)\right|\right]$$
$$\leq \sqrt{\frac{8N}{n}\left(3 + 4N^2 lnn + 4\ln|\Phi_N| + 4\ln\left(\frac{2}{\delta}\right)\right)},$$

where $|\Phi_N|$ is the cardinality of encoder function class and n is the size of the dataset.

1. whenever $\hat{\phi}$ maps two state-actions x_1, x_2 to the same value, $|Z^{\pi}(x_1) - Z^{\pi}(x_2)|$ up to an error proportional to $\frac{1}{\sqrt{n}}$, where n is the size of the dataset.

Corollary 1. *The encoder $\hat{\phi}$ returned by Z-learning algorithm with $n \to \infty$ is a Z^{π}-irrelevance, i.e., for any $x_1, x_2 \in \mathcal{X}$, $\mathbb{Z}^{\pi}(x_1) = \mathbb{Z}^{\pi}(x_2)$ if $\hat{\phi}(x_1) = \hat{\phi}(x_2)$.*

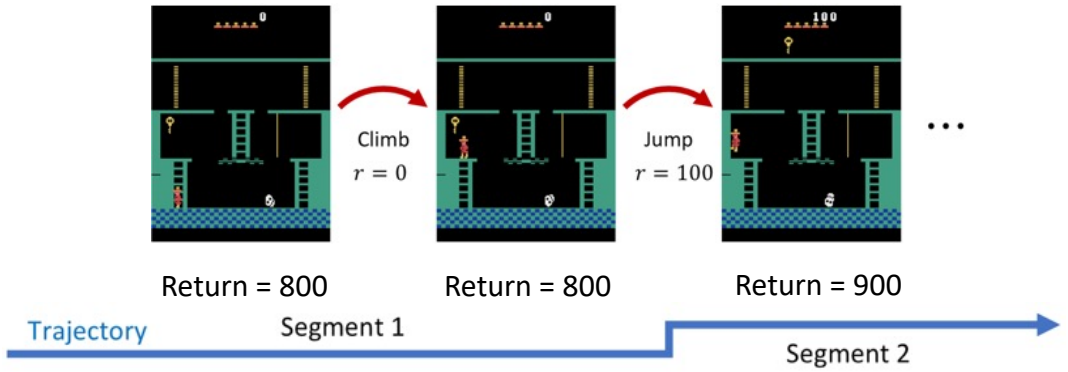
2. $\hat{\phi}$ becomes a Z^{π}-irrelevance abstraction when $n \to \infty$.

Return-based Contrastive Learning (RCRL)

1. The labels in the dataset may be unbalanced in practice, which may prevent the discriminator from learning properly.
2. Manually determine bins on the return distribution as prior.

⇒ Segmenting! ☺

- Cut the trajectories into segments, where each segment contains state-action pairs with the same or similar returns.



An example from *Montezuma's Revenge* in Atari.

Experiments --- Atari-100K benchmark (26 Games)

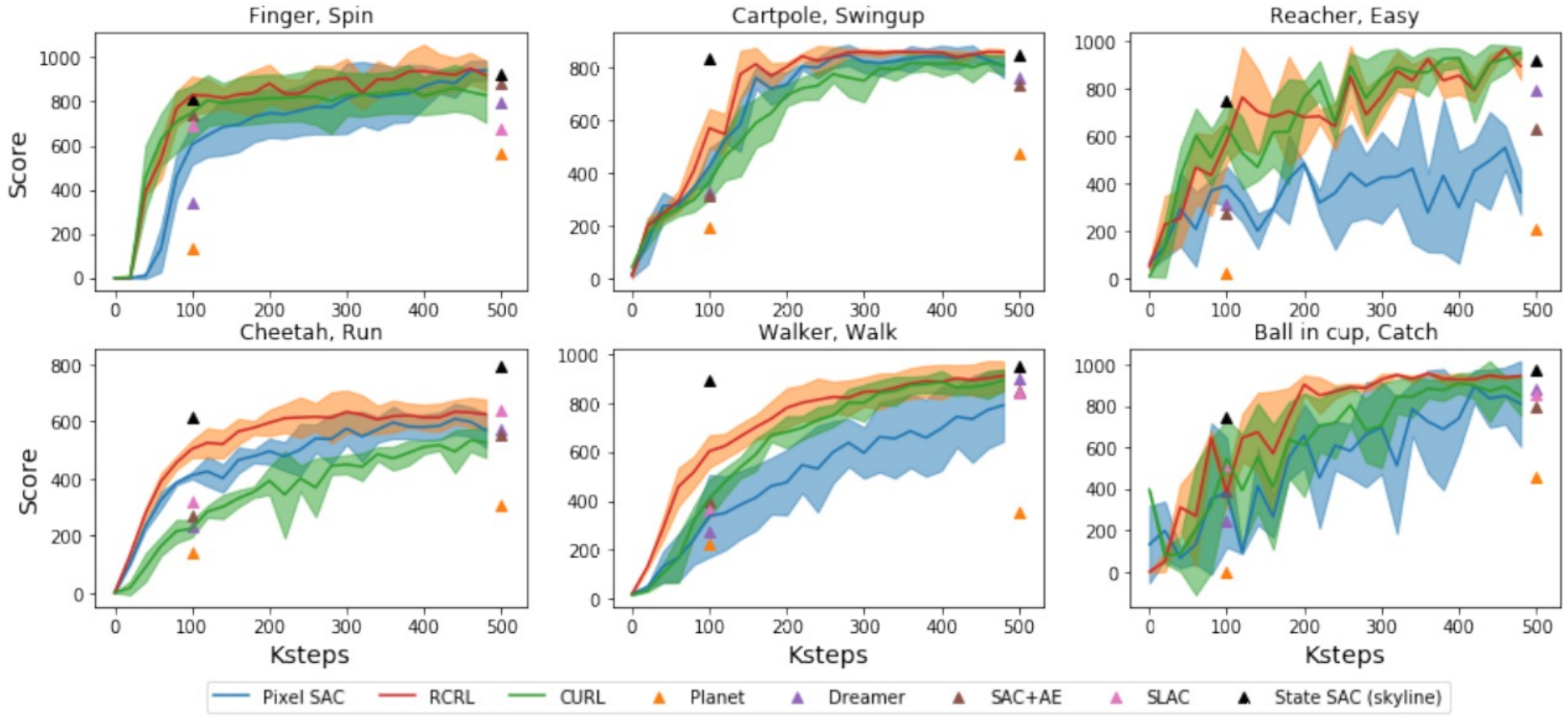
	Human	SimPLe	Rainbow	ERainbow	ERainbow-sa	CURL	RCRL	RCRL+CURL
ALIEN	7127.7	616.9	318.7	739.9	813.8	558.2	854.2	**912.2**
AMIDAR	1719.5	88.0	32.5	**188.6**	154.2	142.1	157.7	125.1
ASSAULT	742.0	527.2	231.0	431.2	576.2	**600.6**	569.6	588.4
ASTERIX	8503.3	**1128.3**	243.6	470.8	697.0	734.5	799.0	683.0
BANK HEIST	753.1	34.2	15.6	51.0	96.0	**131.6**	107.2	99.0
BATTLE ZONE	37187.5	5184.4	2360.0	10124.6	13920.0	14870.0	14280.0	**17380.0**
BOXING	12.1	**9.1**	-24.8	0.2	2.2	1.2	2.7	6.7
BREAKOUT	30.5	**16.4**	1.2	1.9	3.4	4.9	4.3	4.0
CHOPPER COMMAND	7387.8	1246.9	120.0	861.8	1064.0	1058.5	**1262.0**	1008.0
CRAZY CLIMBER	35829.4	**62583.6**	2254.5	16185.3	21840.0	12146.5	15120.0	15032.0
DEMON ATTACK	1971.0	208.1	163.6	508.0	768.0	**817.6**	790.4	618.3
FREEWAY	29.6	20.3	0.0	**27.9**	26.5	26.7	26.6	25.4
FROSTBITE	4334.7	254.7	60.2	866.8	1472.0	1181.3	1337.6	**1516.6**
GOPHER	2412.5	**771.0**	431.2	349.5	384.8	669.3	429.6	458.8
HERO	30826.4	2656.6	487.0	6857.0	4787.9	6279.3	6454.1	**7647.4**
JAMESBOND	302.8	125.3	47.4	301.6	308.0	471.0	314.0	**503.0**
KANGAROO	3035.0	323.1	0.0	779.3	732.0	872.5	842.0	**932.0**
KRULL	2665.5	**4539.9**	1468.0	2851.5	2740.0	4229.6	2997.5	3905.8
KUNG FU MASTER	22736.3	**17257.2**	0.0	14346.1	11914.0	14307.8	9762.0	11856.0
MS PACMAN	6951.6	1480.0	67.0	1204.1	1384.5	1465.5	**1555.2**	1336.8
PONG	14.6	**12.8**	-20.6	-19.3	-18.3	-16.5	-16.9	-18.72
PRIVATE EYE	69571.3	58.3	0.0	97.8	80.0	218.4	102.6	**282.3**
QBERT	13455.0	**1288.8**	123.5	1152.9	893.5	1042.4	1121.0	942.0
ROAD RUNNER	7845.0	5640.6	1588.5	**9600.0**	5392.0	5661.0	6138.0	5392.0
SEAQUEST	42054.7	**683.3**	131.7	354.1	402.0	384.5	375.6	489.6
UP N DOWN	11693.2	3350.3	504.6	2877.4	3235.2	2955.2	**4210.2**	3127.8
Median HNS	100.0%	14.4%	0.0%	16.1%	16.7%	17.5%	18.5%	**19.6%**

Table 1: Scores of different algorithms/baselines on 26 games for Atari-100k benchmark. We show the mean score averaged over five random seeds.

- Outperform strong baselines, such as Rainbow, SimPLe, CURL.
- Achieve even better performance when combined with existing auxiliary task, e.g. CURL.

Experiments --- DMControl

- Outperform strong baselines, such as pixel SAC, CURL.
- Comparable even to State SAC (The Skyline).



Thank You!

Algorithm

Param #1 *Param #2*

Algorithm 2: Return based Contrastive learning for RL (RCRL)

Initialize the embedding $\phi_\theta : \mathcal{X} \to \mathbb{R}^d$ and a discriminator $w_\vartheta : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$

Param #3 Initialize the parameters φ for the base RL algorithm that uses the learned embedding ϕ_θ

Given a batch of samples $\mathcal{D}$, the loss function for the base RL algorithm is $\mathcal{L}_{\mathrm{RL}}(\phi_\theta, \varphi; \mathcal{D})$

A replay buffer $\mathcal{B} = \emptyset$

foreach *iteration* **do**

  Rollout the current policy and store the samples to the replay buffer $\mathcal{B}$

  Draw a batch of samples $\mathcal{D}$ from the replay buffer $\mathcal{B}$

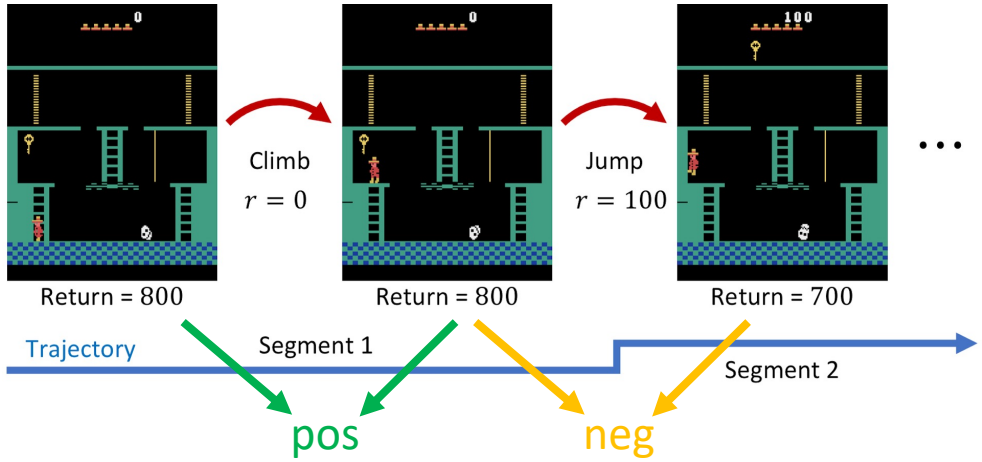
  Update the parameters with the loss function $\mathcal{L}(\phi_\theta, w_\vartheta; \mathcal{D}) + \mathcal{L}_{\mathrm{RL}}(\phi_\theta, \varphi; \mathcal{D})$

end

return *The learned policy*

Sample data & Update all parameters.

Experiments



- Analysis on learned representation
  - Rainbow (w/o aux. task) RCRL (w/ aux. task)
  - pos (state-actions within the same segment) neg (otherwise)
  - Better performance results from better representation