Unsupervised Reinforcement Learning

Pieter Abbeel
UC Berkeley EECS

Outline

- NN Architectures
- Representation Learning for Reinforcement Learning
- Active Pre-Training for Reinforcement Learning
- Human-in-the-loop Reinforcement Learning

Outline

- ***NN Architectures***
- Representation Learning for Reinforcement Learning
- Active Pre-Training for Reinforcement Learning
- Human-in-the-loop Reinforcement Learning

Language Models (e.g. OpenAI's GPT-2)

Prompt: A train carriage containing controlled nuclear materials was stolen in Cincinnati today. Its whereabouts are unknown.

GPT-2: *The incident occurred on the downtown train line, which runs from Covington and Ashland stations. In an email to Ohio news outlets, the U.S. Department of Energy said it is working with the Federal Railroad Administration to find the thief.*

'The theft of this nuclear material will have significant negative consequences on public and environmental health, our workforce and the economy of our nation,' said Tom Hicks, the U.S. Energy Secretary, in a statement. 'Our top priority is to secure the theft and ensure it doesn't happen again.'

The stolen material was taken from the University of Cincinnati's Research Triangle Park nuclear research site, according to a news release from Department officials.

The Nuclear Regulatory Commission did not immediately release any information.

According to the release, the U.S. Department of Energy's Office of Nuclear Material Safety and Security is leading that team's investigation.

'The safety of people, the environment and the nation's nuclear stockpile is our highest priority,' Hicks said. 'We will get to the bottom of this and make no excuses.'

Text Generatio… …'s GPT-2)

SYSTEM PROMPT (HUMAN-WRITTEN)

In a shocking fi… remote, previou… surprising to the … English.

MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)

The scientist nam… Ovid's Unicorn. Th… previously unknown …

Now, after almost tw… odd phenomenon is fi…

Dr. Jorge Pérez, an ev… La Paz, and several co… Mountains when they fou… or humans. Pérez noticed… a natural fountain, surr… snow.

Pérez and the others then … the time we reached the top… with some crystals on top,"…

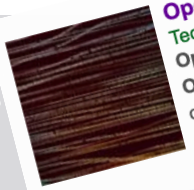
Pérez and his friends were as… These creatures could be seen … too much to see them — they we… horns.

…izarre creatures the scientists discovered … spoke some fairly regular English. Pérez … example, that they have a common …ke a dialect or dialectic."

…e unicorns may have originated in …ls were believed to be descendants of a …ed there before the arrival of humans …rica.

… unclear, some believe that perhaps …n a human and a unicorn met each …ivilization. According to Pérez, …ts seem to be quite common."

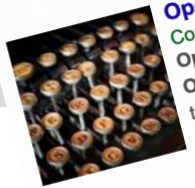
… that it is likely that the only …ns are indeed the descendants of … "But they seem to be able to … which I believe is a sign of … social organization," said the …



OpenAI built a text generator so good, it's considered too dangerous…
TechCrunch - 17 Feb 2019
OpenAI built a text generator **so** good, it's considered **too dangerous to release** …
OpenAI said its new natural **language model**, GPT-2, was trained to … said, it's only **releasing** a smaller version of the **language model**, citing its …
Scientists Developed an AI **So** Advanced They Say It's **Too Dangerous** …
ScienceAlert - 18 Feb 2019

AI text writing technology **too dangerous to release**, creators claim
The Drum - 17 Feb 2019
This technology could 'absolutely devastate' the internet as we know it
NEWS.com.au - 17 Feb 2019
This AI is **so** good at writing that its creators won't let you use it
In-Depth - CNN - 18 Feb 2019
Lord of The Rings, Celebrity Gossip: This AI is **So** Good at Writing That …
In-Depth - News18 - 18 Feb 2019
View all

When Is Technology Too Dangerous to Release to the Public?
Slate Magazine - 22 Feb 2019
If your knowledge of the model, called GPT-2, came solely on headlines … U.K. read, "Elon Musk-Founded **OpenAI** Builds Artificial Intelligence **So** … had trained a **language model** using text from 8 million webpages to predict …
AI Weekly: Experts say **OpenAI's** controversial **model** is a potential …
In-Depth - VentureBeat - 22 Feb 2019
View all



OpenAI's Text Model so Disruptive it's Deemed Too Dangerous To …
Computer Business Review - 15 Feb 2019
OpenAI's Text Model **so** Disruptive it's Deemed **Too Dangerous To Release** …
OpenAI has declined to **release** the full research due to concerns over … We've trained an unsupervised **language model** that can generate …
New AI fake text generator may be **too dangerous to release**, say …
Highly Cited - The Guardian - 14 Feb 2019
View all

[Radford et al, 2019]

Benchmarks – GPT-2

DATASET	METRIC	OUR RESULT	PREVIOUS RECORD	HUMAN
Winograd Schema Challenge	accuracy (+)	**70.70%**	63.7%	92%+
LAMBADA	accuracy (+)	**63.24%**	59.23%	95%+
LAMBADA	perplexity (-)	**8.6**	99	~1-2
Children's Book Test Common Nouns (validation accuracy)	accuracy (+)	**93.30%**	85.7%	96%
Children's Book Test Named Entities (validation accuracy)	accuracy (+)	**89.05%**	82.3%	92%
Penn Tree Bank	perplexity (-)	**35.76**	46.54	unknown
WikiText-2	perplexity (-)	**18.34**	39.14	unknown

Benchmarks -- BERT

GLUE Results

System	MNLI-(m/mm) 392k	QQP 363k	QNLI 108k	SST-2 67k	CoLA 8.5k	STS-B 5.7k	MRPC 3.5k	RTE 2.5k	**Average** -
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.9	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	88.1	91.3	45.4	80.0	82.3	56.0	75.2
$BERT_{BASE}$	84.6/83.4	71.2	90.1	93.5	52.1	85.8	88.9	66.4	79.6
$BERT_{LARGE}$	**86.7/85.9**	**72.1**	**91.1**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**81.9**

MultiNLI

Premise: Hills and mountains are especially sanctified in Jainism.
Hypothesis: Jainism hates nature.
Label: Contradiction

CoLa

Sentence: The wagon rumbled down the road.
Label: Acceptable

Sentence: The car honked down the road.
Label: Unacceptable

Might these pre-trained transformers be ***even*** more general?

Pretrained Transformers As Universal Computation Engines

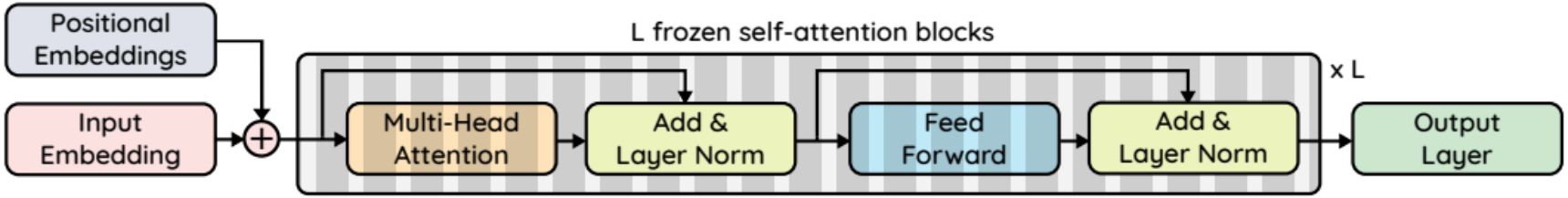
Kevin Lu
UC Berkeley
kzl@berkeley.edu

Aditya Grover
Facebook AI Research
adityagrover@fb.com

Pieter Abbeel
UC Berkeley
pabbeel@cs.berkeley.edu

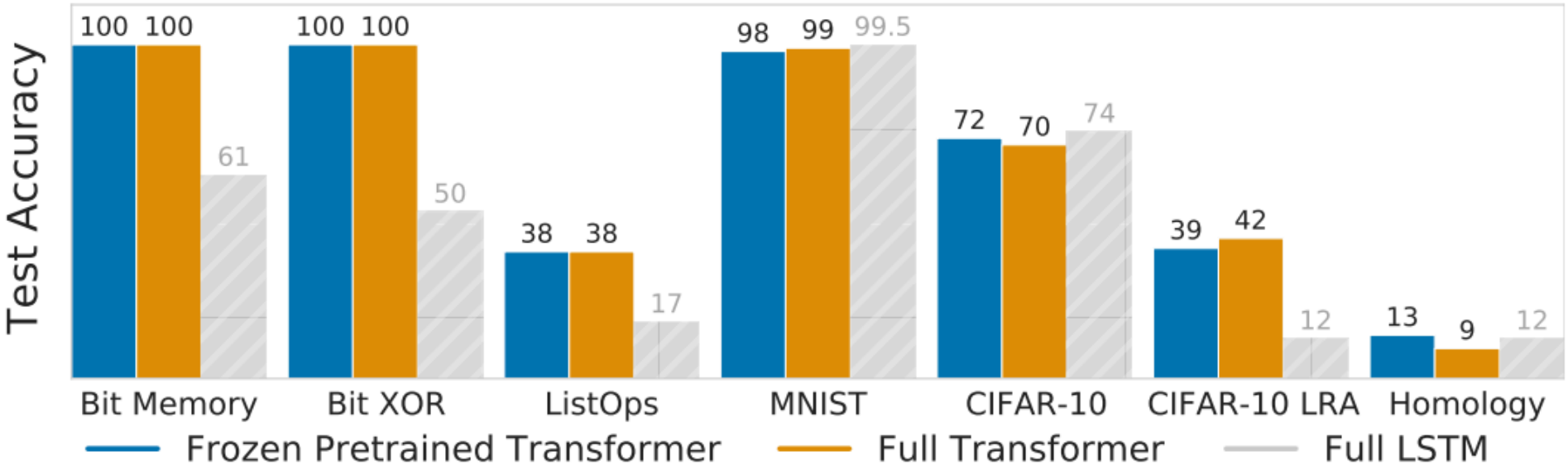
Igor Mordatch
Google Brain
imordatch@google.com

Pre-Trained Model + .1% finetune



- ***Pre-train:*** language corpus next-token prediction
- ***Minimally fine-tune:***
  - Bit memory
  - Bit XOR
  - ListOps
  - MNIST
  - CIFAR-10 and CIFAR-10 LRA
  - Remote homology detection

Can pretrained LMs transfer to new modalities?



What's the importance of the pretraining modality?

Model	Bit Memory	XOR	ListOps	MNIST	C10	C10 LRA	Homology
FPT	100%	100%	38.4%	98.0%	68.2%	38.6%	12.7%
Random	75.8%	100%	34.3%	91.7%	61.7%	36.1%	9.3%
Bit	100%	100%	35.4%	97.8%	62.6%	36.7%	7.8%
ViT	100%	100%	37.4%	97.8%	72.5%	43.0%	7.5%

Table 2: Test accuracy of language-pretrained (FPT) vs randomly initialized (Random) vs Bit Memory pretraining (Bit) vs pretrained Vision Transformer (ViT) models. The transformer is frozen.

Does performance scale with model size?

Model Size	# Layers	Total Params	Trained Params	FPT	Random
Small (Base)	12	117M	106K	68.2%	61.7%
Medium	24	345M	190K	69.8%	64.0%
Large	36	774M	300K	72.1%	65.7%

Table 6: Test accuracy of larger frozen transformer models on CIFAR-10.

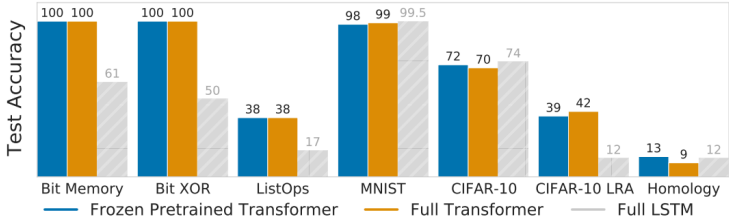
Does the trend hold across other transformer models?

Task	GPT-2 (FPT Default)	BERT	T5	Longformer
ListOps	38.4%	38.3%	15.4%	17.0%
CIFAR-10	68.2%	68.8%	64.7%	66.8%

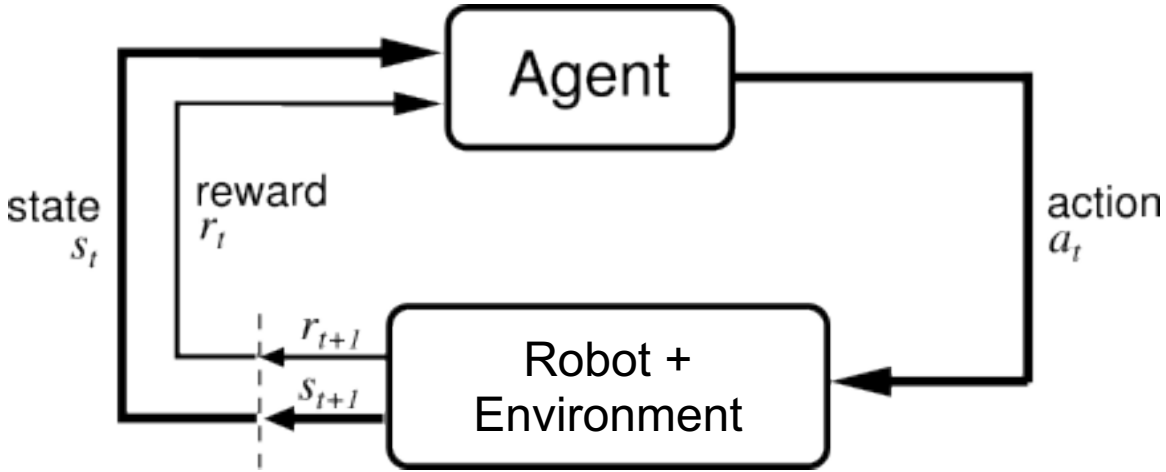
Table 9: Test accuracy for frozen pretrained transformer variants (base model sizes).

Outline

- NN Architectures: Pretrained Transformers as Universal Computation Engines
- ***Representation Learning for Reinforcement Learning***
- Active Pre-Training for Reinforcement Learning
- Human-in-the-loop Reinforcement Learning



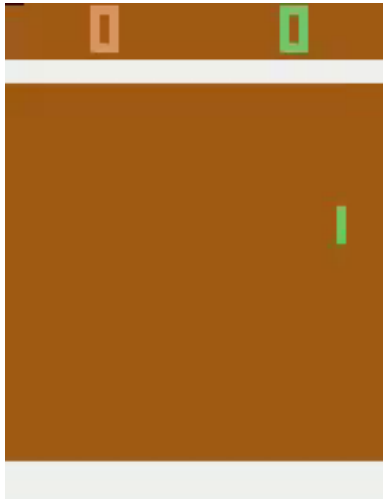
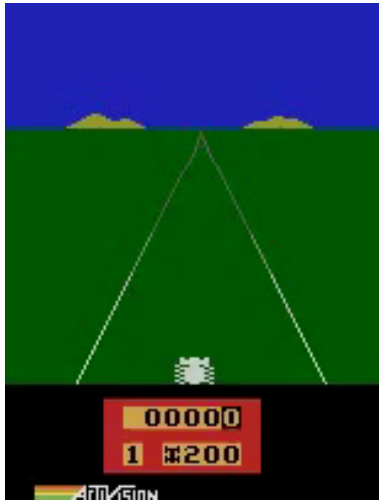
Reinforcement Learning (RL)

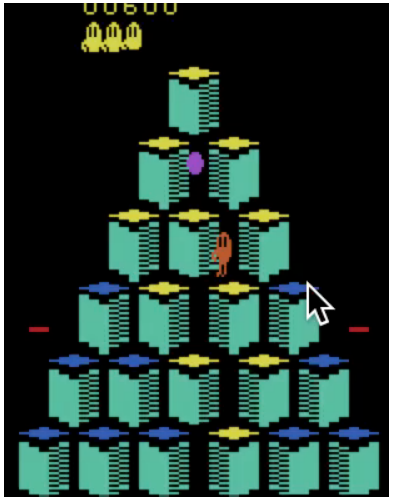


$$\max_{\theta} \quad \mathrm{E}[\sum_{t=0}^{H} R(s_t)|\pi_\theta]$$

[Image credit: Sutton and Barto 1998]

Deep RL: Atari




DQN Mnih et al, NIPS 2013 / Nature 2015

MCTS Guo et al, NIPS 2014; **TRPO** Schulman, Levine, Moritz, Jordan, Abbeel, ICML 2015; **A3C** Mnih et al, ICML 2016; **Dueling DQN** Wang et al ICML 2016; **Double DQN** van Hasselt et al, AAAI 2016; **Prioritized Experience Replay** Schaul et al, ICLR 2016; **Bootstrapped DQN** Osband et al, 2016; **Q-Ensembles** Chen et al, 2017; **Rainbow** Hessel et al, 2017; **Accelerated** Stooke and Abbeel, 2018; …

Deep RL: Go

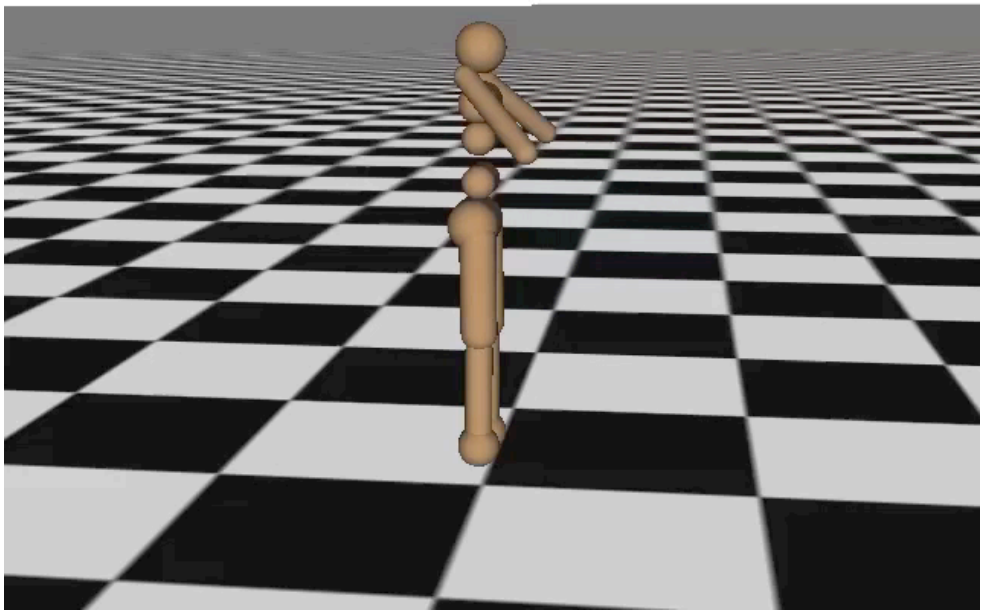
AlphaGo Silver et al, Nature 2015

AlphaGoZero Silver et al, Nature 2017

AlphaZero Silver et al, 2017

Tian et al, 2016; Maddison et al, 2014; Clark et al, 2015

Deep RL: Robot Locomotion



^ **TRPO** Schulman et al, 2015 + **GAE** Schulman et al, 2016

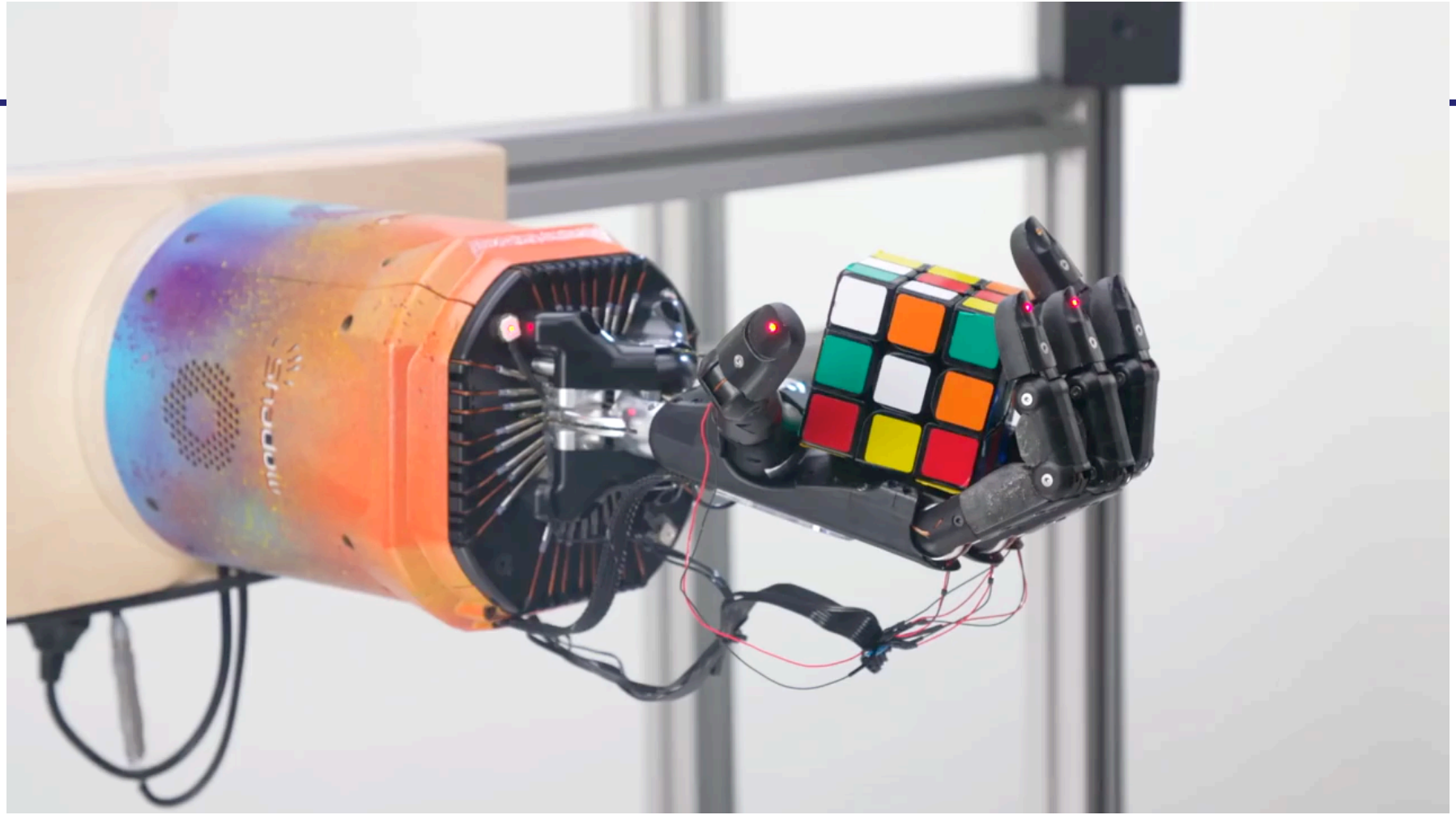
See also: **DDPG** Lillicrap et al 2015; **SVG** Heess et al, 2015; **Q-Prop** Gu et al, 2016; **Scaling up ES** Salimans et al, 2017; **PPO** Schulman et al, 2017; **Parkour** Heess et al, 2017;

Deep RL: Robot Locomotion

[Peng, Abbeel, Levine, van de Panne, 2018]

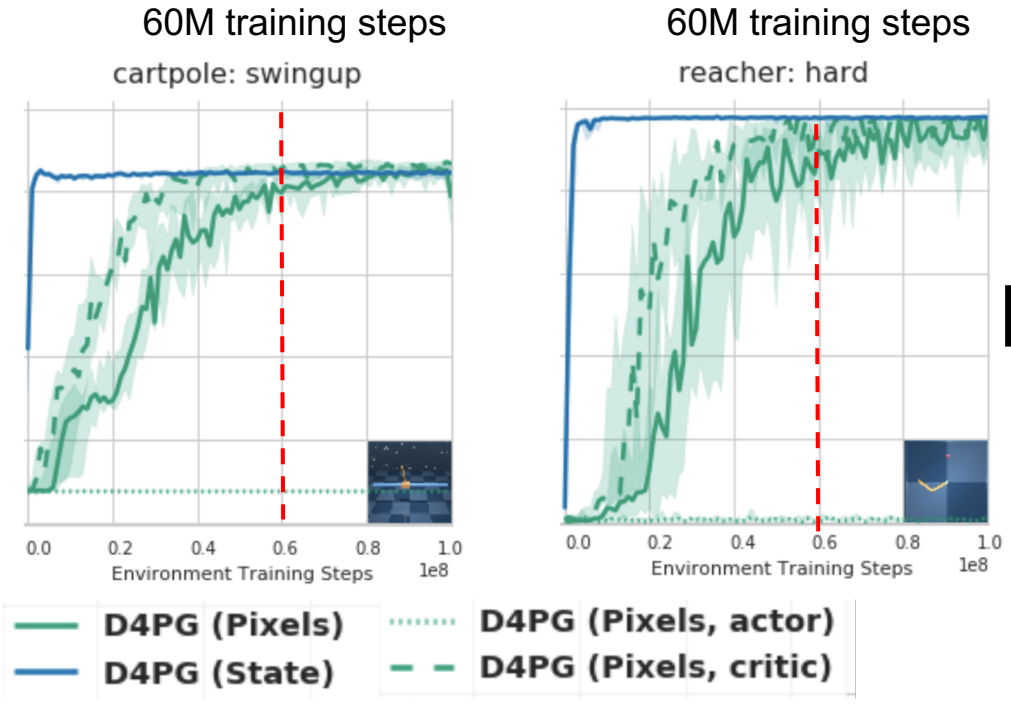
BRETT: Berkeley Robot for the Elimination of Tedious Tasks

[Levine*, Finn*, Darrell, Abbeel, JMLR 2016]



Can visual RL match data-efficiency state RL?

- State-based D4PG (blue) vs pixel-based D4PG (green)



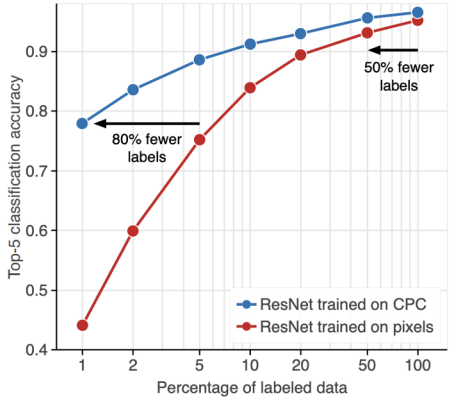
Pixel-based needs > 50M more training steps than state-based to solve same tasks

[Tassa et al., 2018] Tassa, Y., Doron, Y., Muldal, A., Erez, T., Li, Y., Casas, D.D.L., Budden, D., Abdolmaleki, A., Merel, J., Lefrancq, A. and Lillicrap, T DeepMind Control Suite, arxiv:1801.00690, 2018.

LeCake (Yann LeCun)

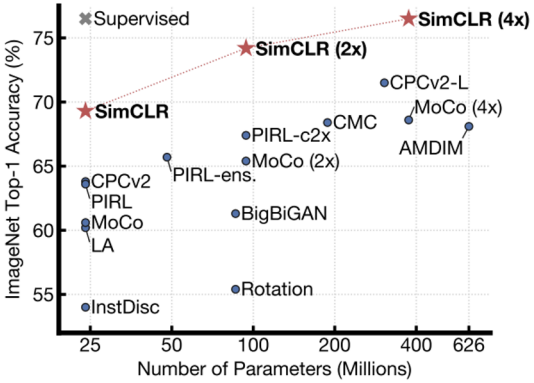
Contrastive learning: SOTA in computer vision

CPCv2 **top-5** ImageNet accuracy as function of labels



[Henaff, Srinivas et al., 2019]

SimCLR **top-1** ImageNet accuracy as function of # of parameters

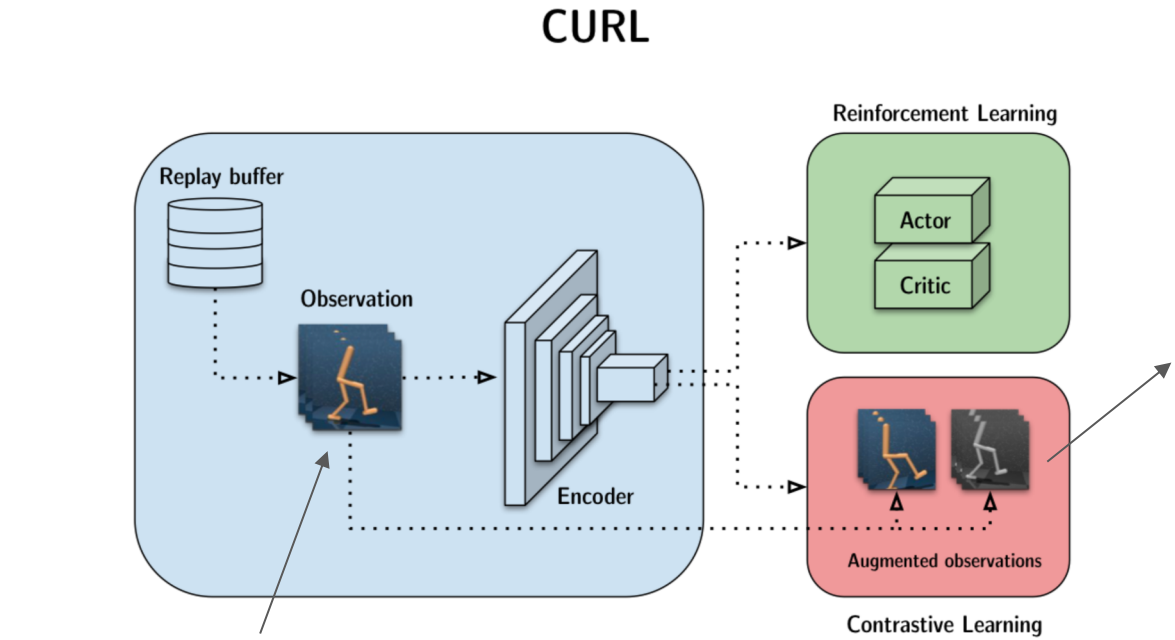


[Chen et al., 2020]

[Henaff et al., 2019] Olivier J. Hénaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S. M. Ali Eslami, Aaron van den Oord Data-Efficient Image Recognition with Contrastive Coding arxiv:1905.09272, 2019.

[Chen et al., 2020] Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. A Simple Framework for Contrastive Learning of Visual Representations arxiv:2002.05709, 2020.

Contrastive + RL



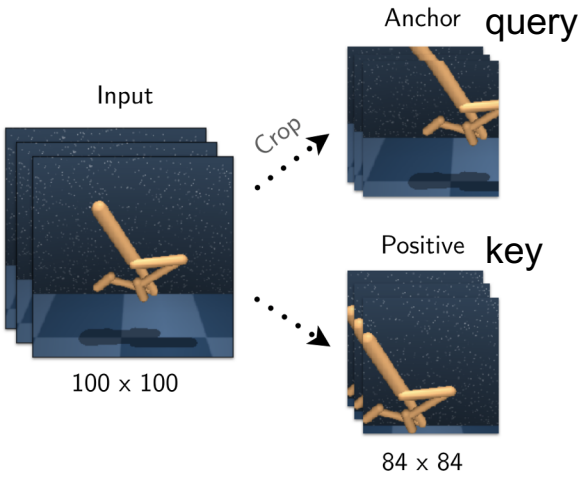
Need to define:

1. query / key pairs
2. similarity measure
3. architecture

Observations are stacked frames

[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

1. Query / key pairs: random crop

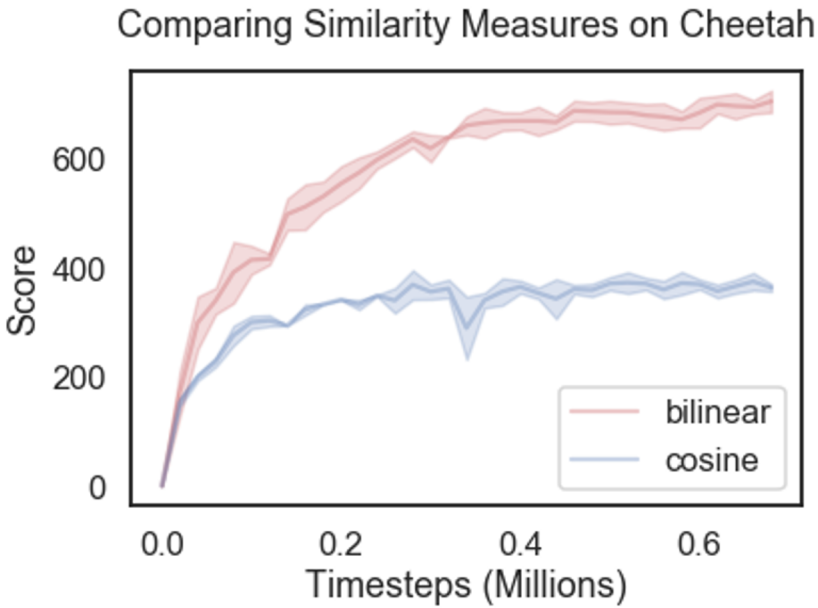


[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

2. Bilinear inner product with learned weight matrix

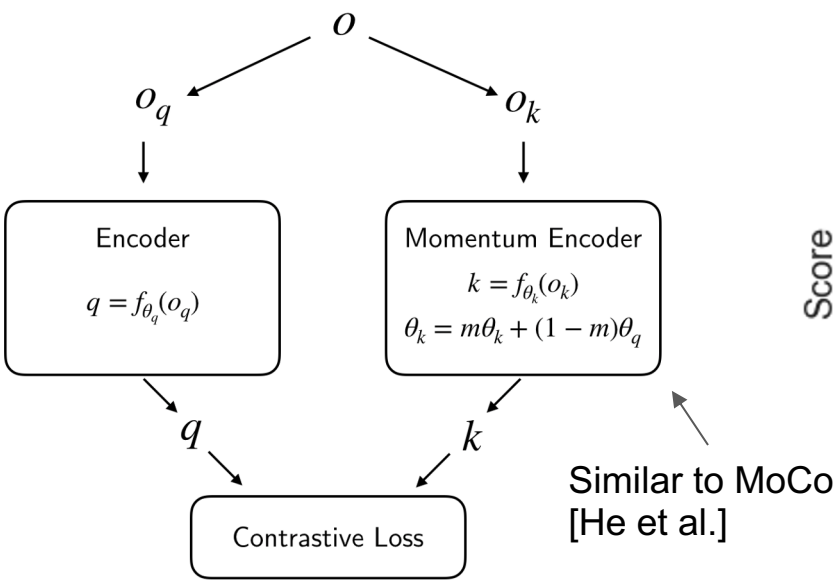
$$\begin{matrix} \text{logits} \\ \begin{bmatrix} q_0^T W k_0 & q_0^T W k_1 & \dots & q_0^T W k_j \\ q_1^T W k_0 & q_1^T W k_1 & \dots & q_1^T W k_j \\ \vdots & \vdots & \ddots & \vdots \\ q_j^T W k_0 & q_j^T W k_1 & \dots & q_k^T W k_j \end{bmatrix} \end{matrix} \quad \begin{matrix} \text{labels} \\ \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \end{matrix}$$

$$\mathscr{L}_q = \frac{\exp(q^T W k_+)}{\exp(q^T W k_+) + \sum_{i=0}^{K-1} \exp(q^T W k_i)}$$

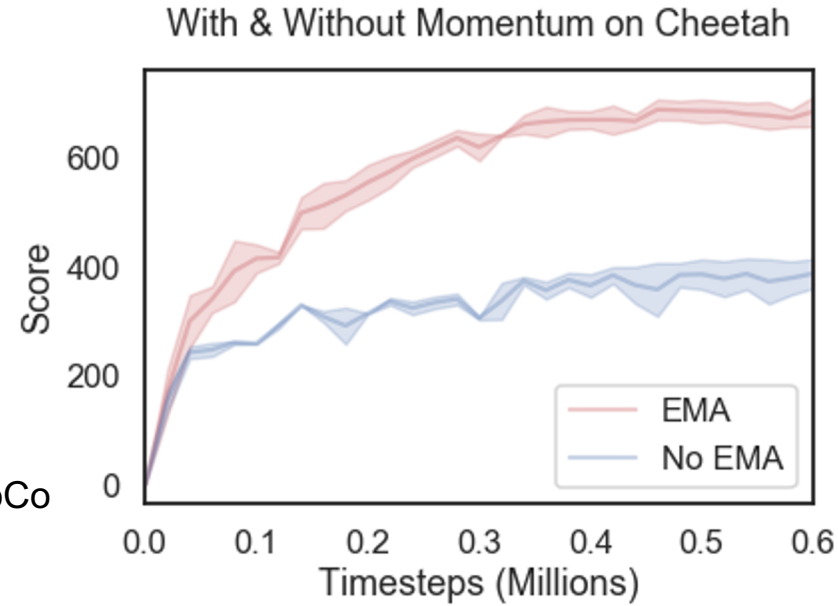


[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

3. Keys encoded with momentum



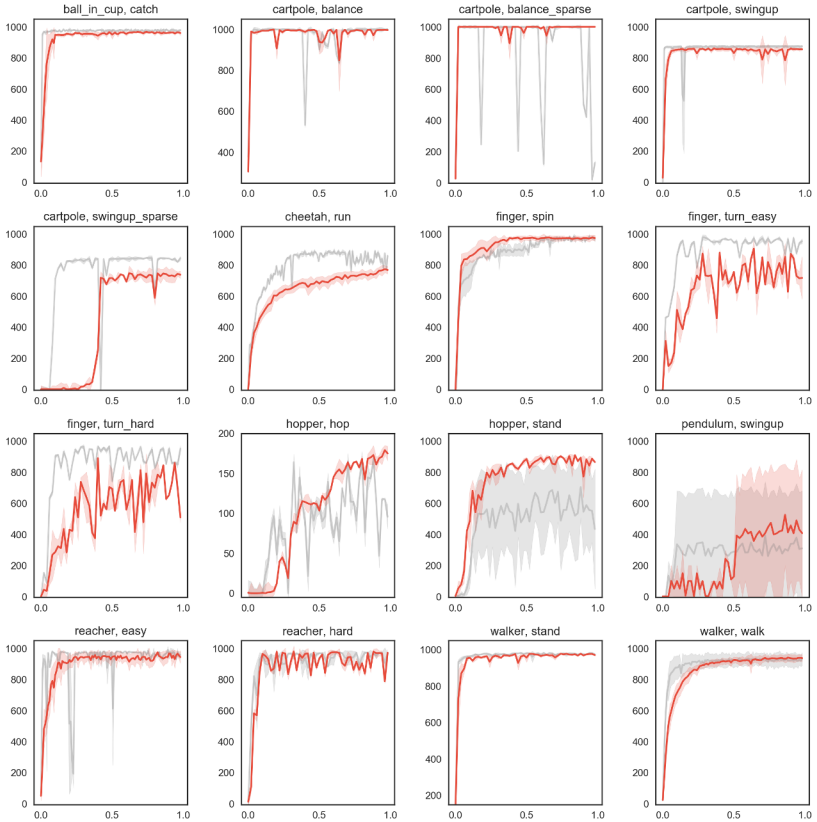
Similar to MoCo [He et al.]



[He et al., 2019] He, K., Fan, H., Wu, Y., Xie, S. and Girshick, R. Momentum Contrast for Unsupervised Visual Representation Learning arxiv:1911.05722, 2019.

CURL from pixels matches state-based SAC

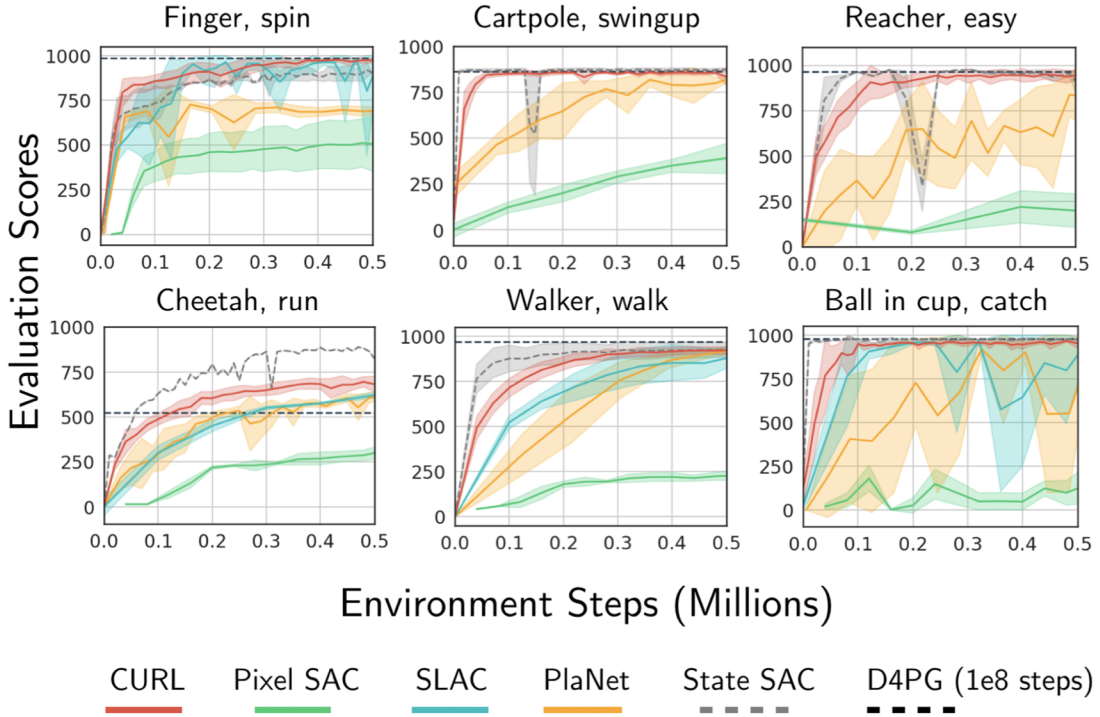
GRAY: SAC State
RED: CURL



[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

Pieter Abbeel

CURL Comparison: DeepMind Control Suite



[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

CURL Comparison: Atari

Atari performance benchmarked at 100K frames

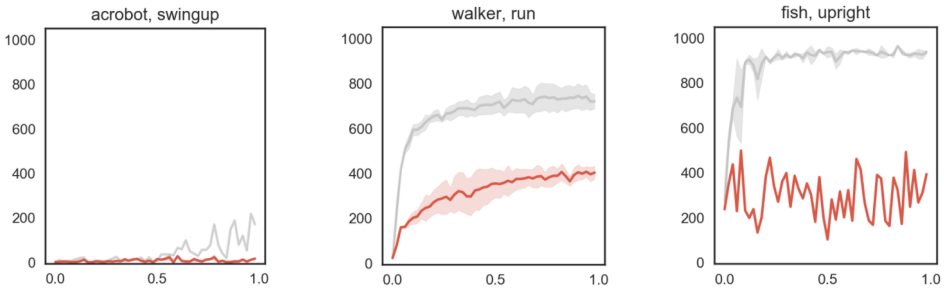
100K step scores	CURL Rainbow	SimPLe	Rainbow	Human	Random
Alien	**1148.2**	616.9	318.7	6875	184.8
Amidar	**232**	74.3	32.5	1676	11.8
Assault	473	**527.2**	231	1496	248.8
BattleZone	**11208**	4031.2	3285.71	37800	2895
Freeway	**27**	16.7	0	29.6	0
Frostbite	**924**	236.9	60.2	4335	74
Jamesbond	**400**	100.5	47.4	406.7	29.2
Qbert	**1352**	1288.8	123.46	13455	166.1
Seaquest	408	**683.3**	131.69	20182	61.1

[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

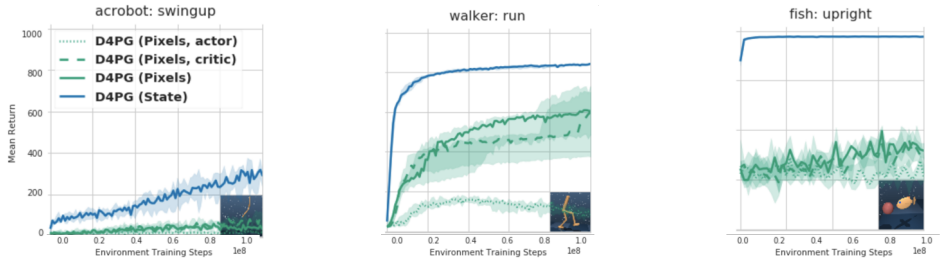
Hard Environments

GRAY: SAC State
RED: CURL

Environment training steps 1 = 1M



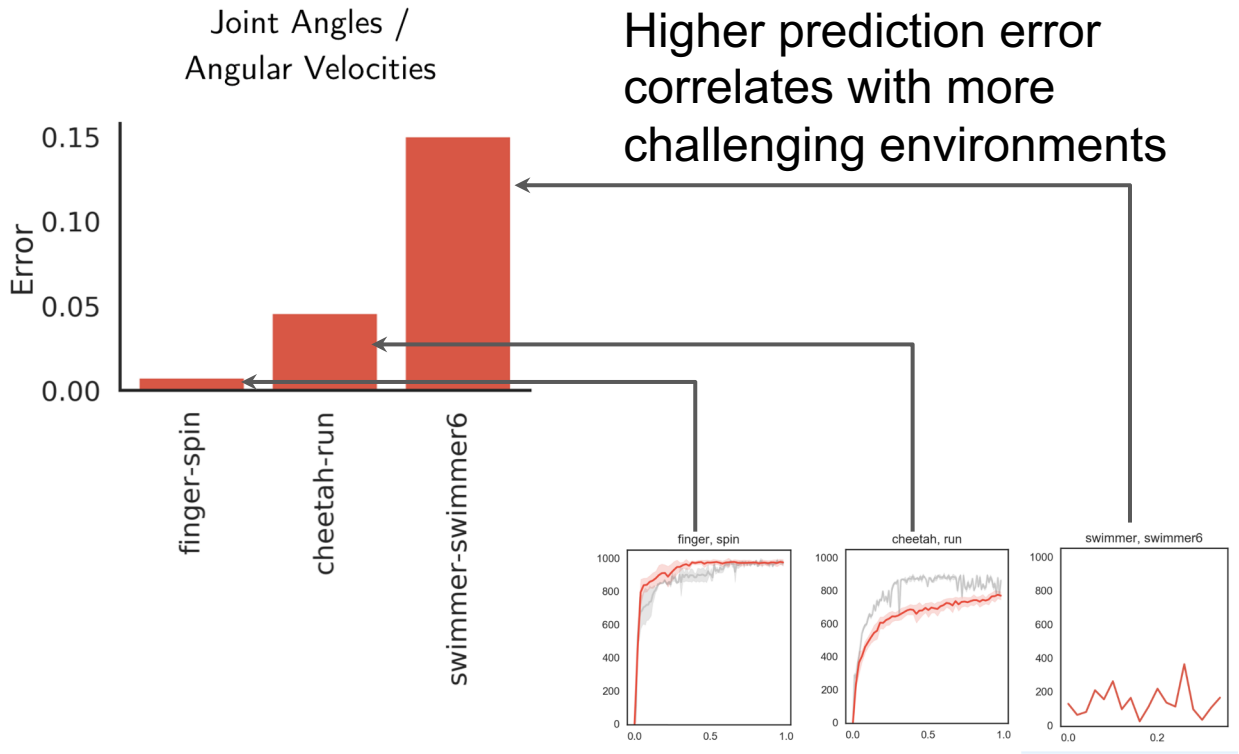
Environment training steps 1 = 100M



Observation:
similar struggle asymptotically for pixel-based Deep RL

[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

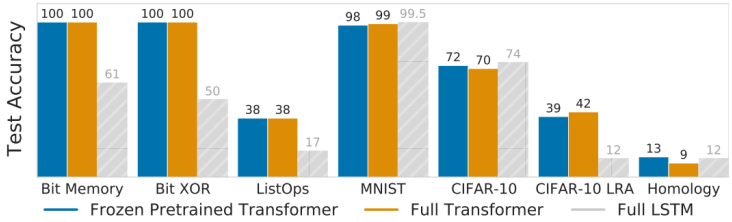
Predicting state from pixels

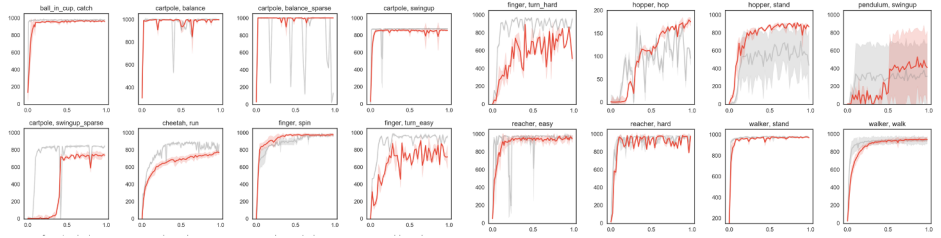


[CURL: A Srinivas*, M Laskin*, P Abbeel, 2020]

Outline

- NN Architectures: Pretrained Transformers as Universal Computation Engines
- Representation Learning for Reinforcement Learning
- ***Active Pre-Training for Reinforcement Learning***
- Human-in-the-loop Reinforcement Learning





Pieter Abbeel

RL with Unsupervised Pretraining:

- Agent is allowed to train for a long period:
  - ***with*** access to the environment
  - ***without*** access to the ultimate reward

RL with Unsupervised Pretraining

Algorithm	Objective	Visual Domain	Exploration	Off-policy	Pre-Trained model
MaxEnt	$\max \mathrm{H}(s)$	✗	✓*	✗	$\pi(a\|s)$
CBB	$\max \mathbb{E}_s[c(s)]$	✗	✓	✓	$\pi(a\|s)$
MEPOL	$\max \mathrm{H}(s)$	✗	✓*	✗	$\pi(a\|s)$
VISR	$\max -\mathrm{H}(z\|s)$	✓	✗	✓	$\psi(s,z), \phi(s)$
DIAYN	$\max -\mathrm{H}(z\|s) + \mathrm{H}(a\|z,s)$	✗	✓*	✓	$\pi(a\|s,z)$
EDL	$\max \mathrm{H}(s) - \mathrm{H}(s\|z)$	✗	✓*	✓	$\pi(a\|s,z)$
DADS	$\max \mathrm{H}(s) - \mathrm{H}(s\|z)$	✗	✗	✓	$\pi(a\|s,z), q(s'\|s,z)$
APT	$\max \mathrm{H}(s)$	✓	✓	✓	$\pi(a\|s), Q(s,a)$

$\psi(s,a)$: successor feature, $\phi(s)$: state representation, c(s): count-based bonus. *: only in state-based RL.

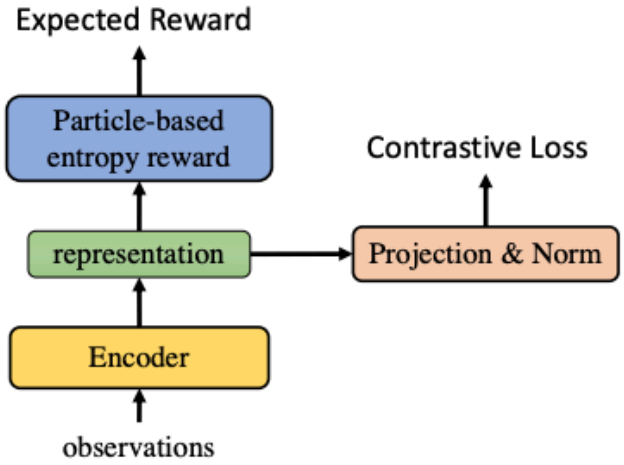
Table 1: Methods for pre-training RL in reward-free setting. DIAYN (Eysenbach et al., 2018), Count based bonus(CBB) (Bellemare et al., 2016), MEPOL (Mutti et al., 2020), VISR (Hansen et al., 2020), DADS (Sharma et al., 2019), EDL (Campos et al., 2020). Exploration: the method can explore efficiently. Visual: the method works well in visual RL. Off-policy: the method is off-policy RL.

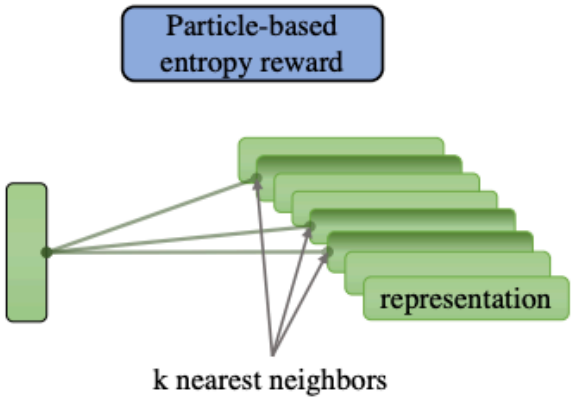
Our Approach (APT)

- During pre-training:

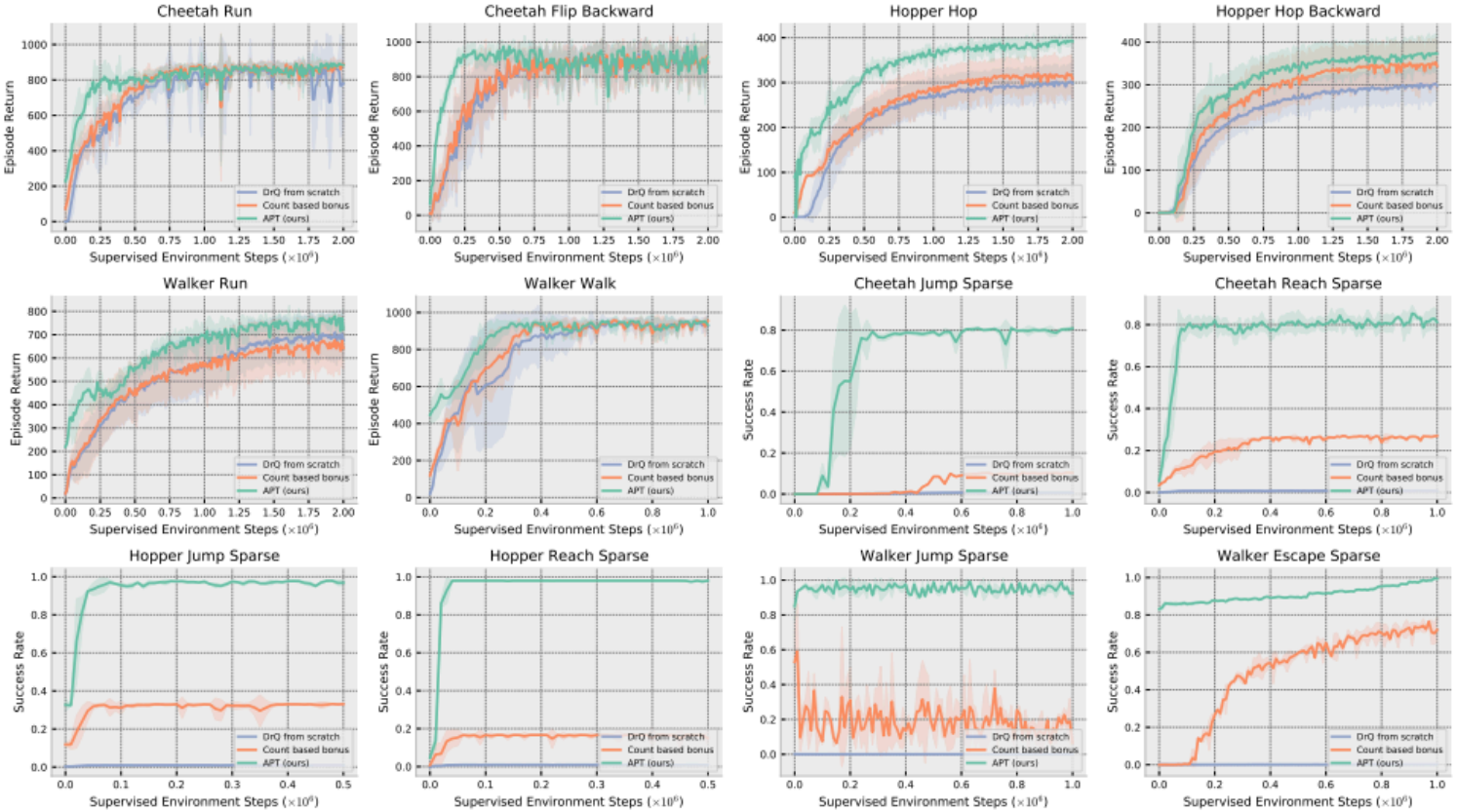
$$r(s, a, s') = \log\left(c + \frac{1}{k} \sum_{z^{(j)} \in \mathrm{N}_k(z)} \|z - z^{(j)}\|_{n_z}^{n_z}\right)$$

where $z = f_\theta(s)$





Experiments: Deepmind Control Suite



5MM pre-training steps

Experiments: Deepmind Control Suite

Methods	DrQ	CURL	CBB	DIAYN*	MEPOL*	APT (ours)	SAC (state)
Cheetah Run	660 ± 96	641 ± 51	113 ± 69	81 ± 29	156 ± 128	$\mathbf{671 \pm 89}$	772 ± 60
Hopper Hop	315 ± 103	289 ± 87	327 ± 121	171 ± 45	61 ± 29	$\mathbf{398 \pm 45}$	285 ± 95
Walker Run	713 ± 139	676 ± 127	381 ± 171	137 ± 43	275 ± 71	$\mathbf{789 \pm 58}$	801 ± 76
Cheetah Jump Sparse	$.0 \pm .0$	$.0 \pm .0$	$.18 \pm .08$	$.21 \pm .09$	$.05 \pm .02$	$\mathbf{.79 \pm .13}$	$.56 \pm .23$
Hopper Reach Sparse	$.13 \pm .05$	$.0 \pm .0$	$.17 \pm .31$	$.12 \pm .08$	$.13 \pm .06$	$\mathbf{.94 \pm .04}$	$.71 \pm .18$
Walker Turnover Sparse	$.0 \pm .0$	$.0 \pm .0$	$.67 \pm .14$	$.58 \pm .22$	$.43 \pm .12$	$\mathbf{.97 \pm .03}$	$.69 \pm .16$

Experiments: Atari

Algorithm	26 Game Subset Mdn	M	$>H$	Full 57 Games Mdn	M	$>H$
// Fully-Supervised Training						
SimPLe @100K	14.39	44.30	2	–	–	–
OTR @100K	20.40	26.42	1	–	–	–
PPO @500K	20.93	43.74	**7**	–	–	–
DQN @10M	27.80	52.95	**7**	8.61	27.55	7
SPR @100K	41.50	70.40	**7**	–	–	–
CURL @100K	17.50	38.10	2	–	–	–
DrQ @100K	28.42	35.70	2	–	–	–
// Unsupervised Pre-Training w/ Supervised Fine-Tuning						
DIAYN @100K	1.34	25.39	2	2.95	23.90	6
VISR @100K	9.50	**128.07**	**7**	6.81	102.31	11
GPI VISR @100K	6.59	111.23	**7**	8.99	**109.16**	**12**
CBB@100K	1.23	21.94	3	–	–	–
MEPOL@100K	0.34	17.94	2	–	–	–
APT(ours)@100K	**43.50**	62.06	**7**	**31.55**	43.59	**12**

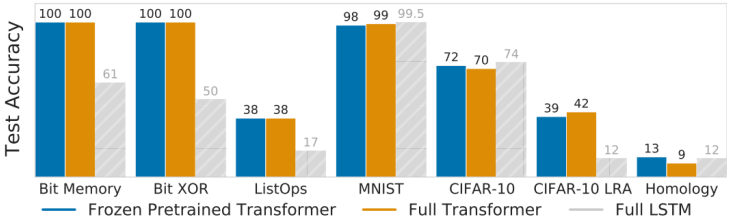
Take-aways:
- APT and VISR are strongest unsupervised pre training methods
- VISR stronger mean
- APT stronger median
- qualitatively: APT better on hard exploration games

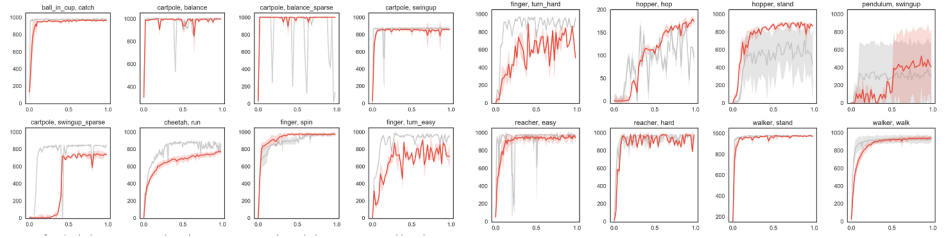
VISR <> APT

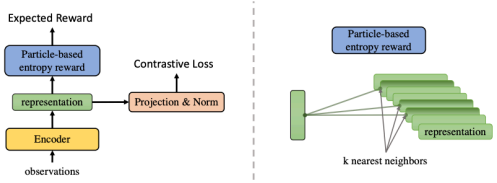
Game	Random	Human	VISR	APT
Alien	227.8	7127.7	364.4	**2614.8**
Amidar	5.8	1719.5	186.0	**211.5**
Assault	222.4	742.0	**1209.1**	891.5
Asterix	210.0	8503.3	**6216.7**	185.5
Asteroids	7191	47388.7	**4443.3**	678.7
Atlantis	12850.0	29028.1	**140542.8**	40231.0
Bank Heist	14.2	753.1	71.3	**416.7**
Battle Zone	2360.0	37187.5	**7072.7**	7065.1
Beam Rider	363.9	16826.5	1741.9	**3487.2**
Berzerk	123.7	2630.4	490.0	**493.4**
Bowling	23.1	160.7	**21.2**	-56.5
Boxing	0.1	12.1	13.4	**21.3**
Breakout	1.7	30.5	**17.9**	10.9
Centipede	2090.9	12017.1	**7184.9**	6233.9
Chopper Command	811.0	7387.8	**800.8**	317.0
Crazy Climber	10780.5	23829.4	**49373.9**	44128.0
Defender	2874.5	18688.9	**15876.1**	5927.9
Demon Attack	107805	35829.4	**8994.9**	6871.8
Double Dunk	-18.6	-16.4	-22.6	**-17.2**
Enduro	0.0	860.5	-3.1	**-0.3**
Fishing Derby	-91.7	-38.7	-93.9	**-5.6**
Freeway	0.0	29.6	-12.1	**29.9**
Frostbite	65.2	4334.7	230.9	**1796.1**
Gopher	257.6	2412.5	498.6	**2190.4**
Gravitar	173.0	3351.4	328.1	**542.0**
Hero	1027.0	30826.4	663.5	**6789.1**
Ice Hockey	-11.2	0.9	**-18.1**	-30.1
Jamesbond	29.0	302.8	**484.4**	356.1
Kangaroo	52.0	3035.0	**1761.9**	412.0
Krull	1598.0	2665.5	**3142.5**	2312.0
Kung Fu Master	258.5	22736.3	16754.9	**17357.0**
Montezuma Revenge	0.0	4753.3	0.0	**0.2**
Ms Pacman	307.3	6951.6	558.5	**2527.1**
Name This Game	2292.3	8049.0	**2605.8**	1387.2
Phoenix	761.4	7242.6	**7162.2**	3874.2
Pitfall	-229.4	6463.7	-370.8	**-12.8**
Pong	-20.7	14.6	-26.2	**-8.0**
Private Eye	24.9	69571.3	**98.3**	96.1
Qbert	163.9	13455.0	666.3	**17671.2**
Riverraid	1338.5	17118.0	**5422.2**	4671.0
Road Runner	11.5	7845.0	**6146.7**	4782.1
Robotank	2.2	11.9	10.0	**13.7**
Seaquest	68.4	42054.7	706.6	**2116.7**
Skiing	-17098.1	-4336.9	**-19692.5**	-38434.1
Solaris	1236.3	12326.7	**1921.5**	841.8
Space Invaders	148.0	1668.7	**9741.0**	3687.2
Star Gunner	664.0	10250.0	**25827.5**	8717.0
Surround	-10.0	6.5	-15.5	**-2.5**
Tennis	-23.8	-8.3	0.7	**1.2**
Time Pilot	3568.0	5229.2	**4503.6**	2567.0
Tutankham	11.4	167.6	50.7	**124.6**
Up N Down	533.4	11693.2	**10037.6**	8289.4
Venture	0.0	1187.5	-1.7	**231.0**
Video Pinball	0.0	17667.9	**35120.3**	2817.1
Wizard Of Wor	563.5	4756.5	853.3	**1265.0**
Yars Revenge	3092.9	54576.9	**5543.5**	1871.5
Zaxxon	32.5	9173.3	897.5	**3231.0**
Mean Human-Norm'd	0.000	1.000	68.42	**81.24**
Median Human-Norm'd	0.000	1.000	9.41	**66.89**
#Superhuman	0	N/A	11	**12**

Outline

- NN Architectures: Pretrained Transformers as Universal Computation Engines
- Representation Learning for Reinforcement Learning
- Active Pre-Training for Reinforcement Learning
- ***Human-in-the-loop Reinforcement Learning***





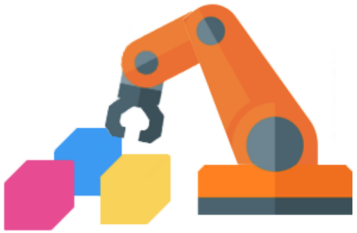


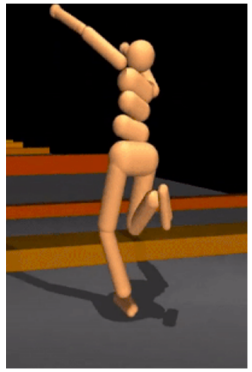
Pieter Abbeel

Challenge: Designing Suitable Reward

Decision
(actions)

Consequences
(observations,
rewards)



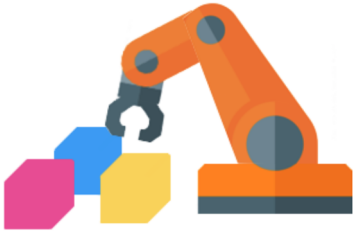


$$\hat{x}_t - 0.1||a_t||_2^2 - 3.0 \times (z_t - 1.3)^2$$

Challenge: Designing Suitable Reward

Decision (actions)

Consequences (observations, **rewards**)



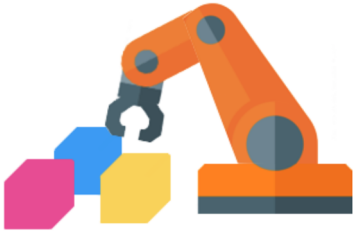
$$\hat{x}_t - 0.1||a_t||_2^2 - 3.0 \times (z_t - 1.3)^2$$

Hard tasks to define a reward (e.g. cooking)

Challenge: Designing Suitable Reward

Decision (actions)

Consequences (observations, **rewards**)



$$\hat{x}_t - 0.1||a_t||_2^2 - 3.0 \times (z_t - 1.3)^2$$

Hard tasks to define a reward (e.g. cooking)

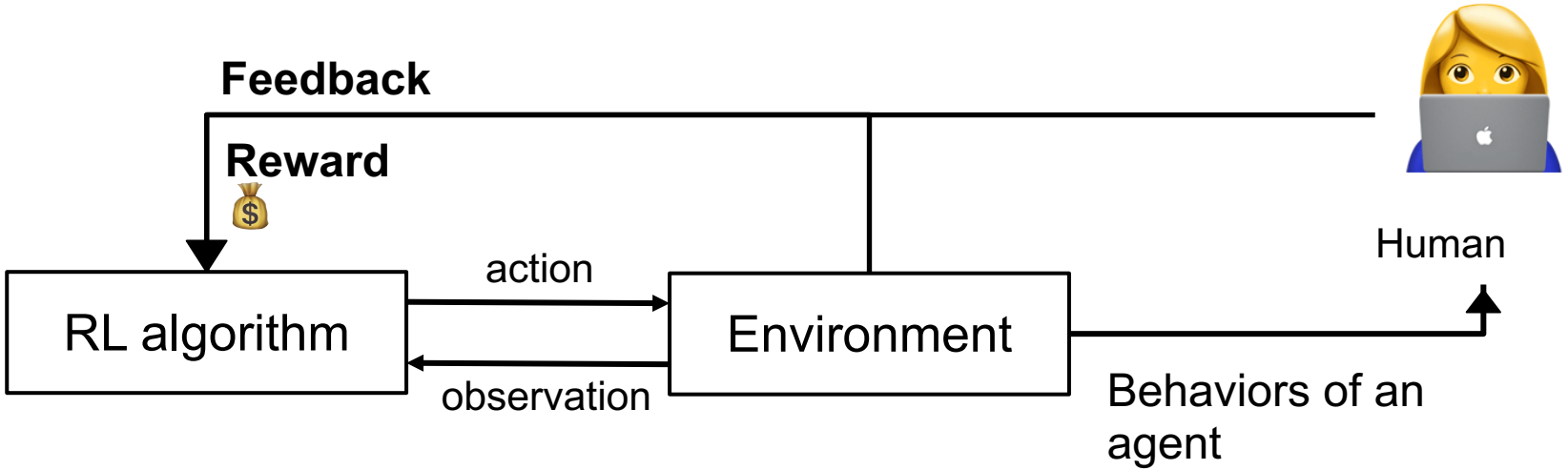
Reward exploitation

https://openai.com/blog/faulty-reward-functions/

What is Alternative Solution?

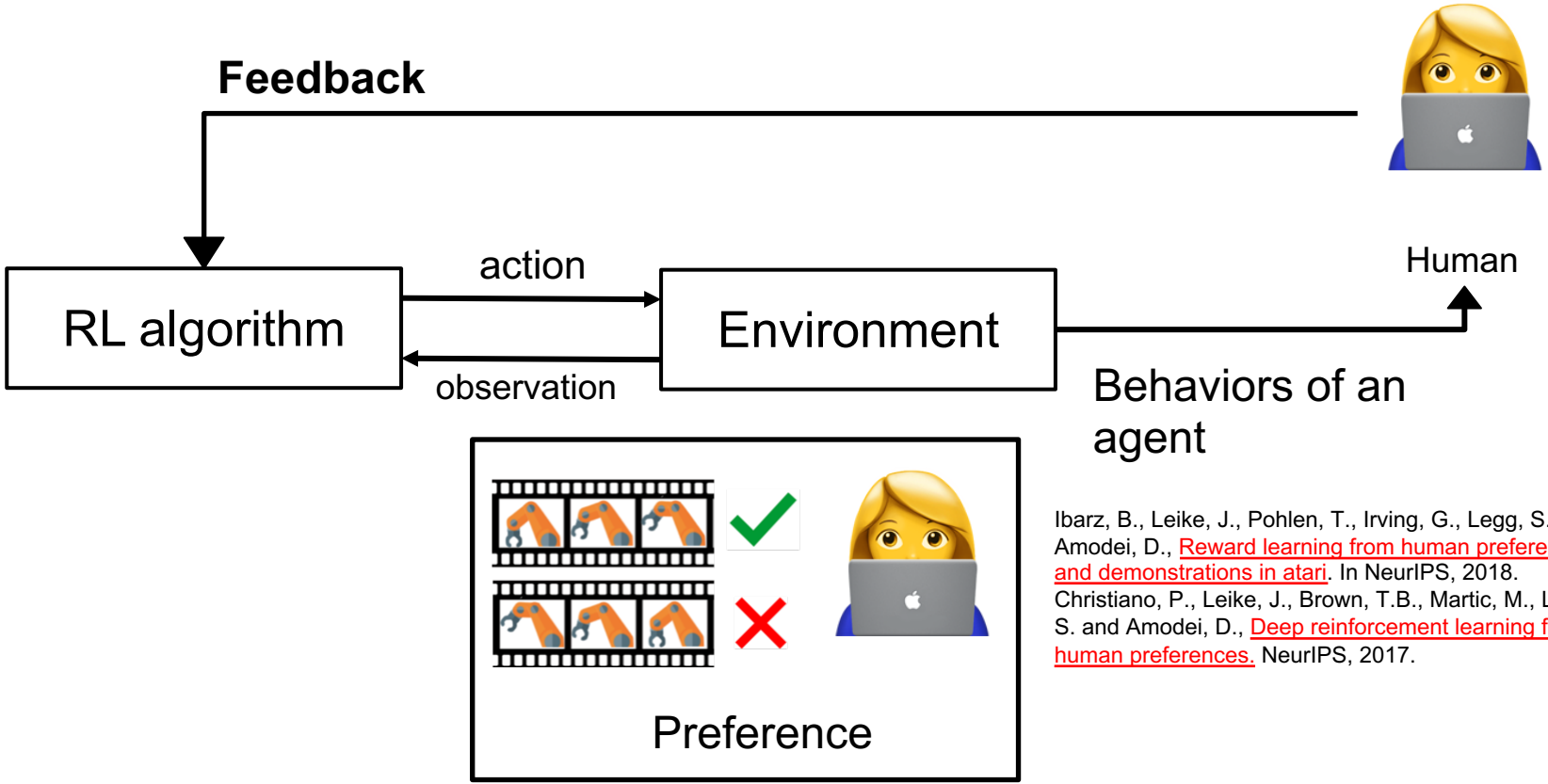
What is Alternative Solution?

- Putting (non-expert) humans into the agent learning loop!



What is Alternative Solution?

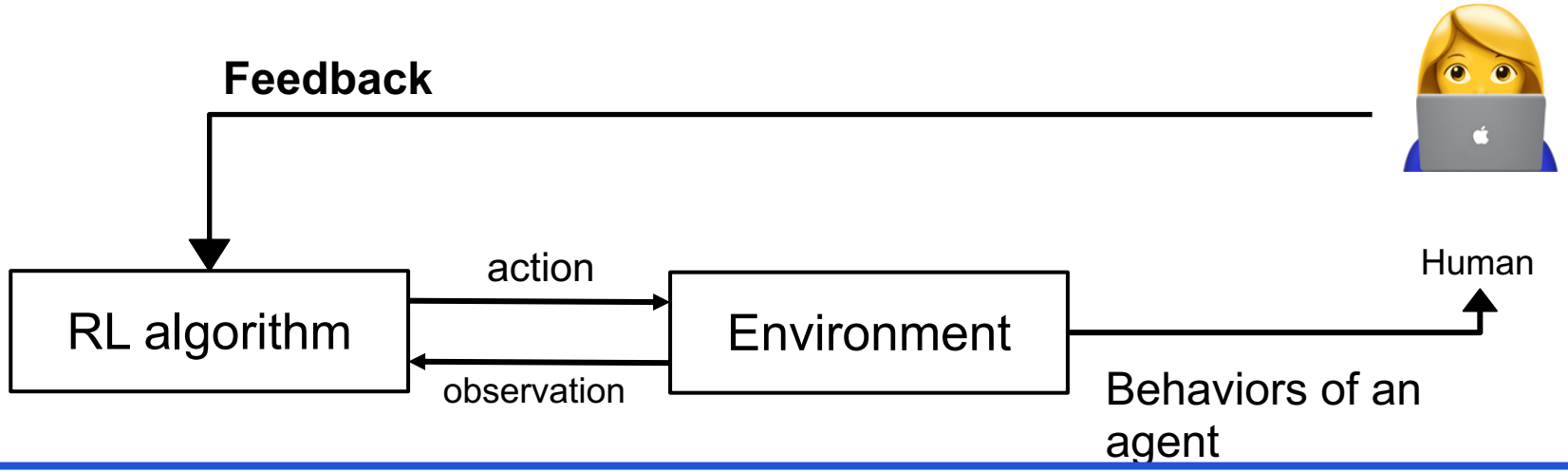
- Putting (non-expert) humans into the agent learning loop!



Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S. and Amodei, D., Reward learning from human preferences and demonstrations in atari. In NeurIPS, 2018.
Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.

What is Alternative Solution?

- Putting (non-expert) humans into the agent learning loop!



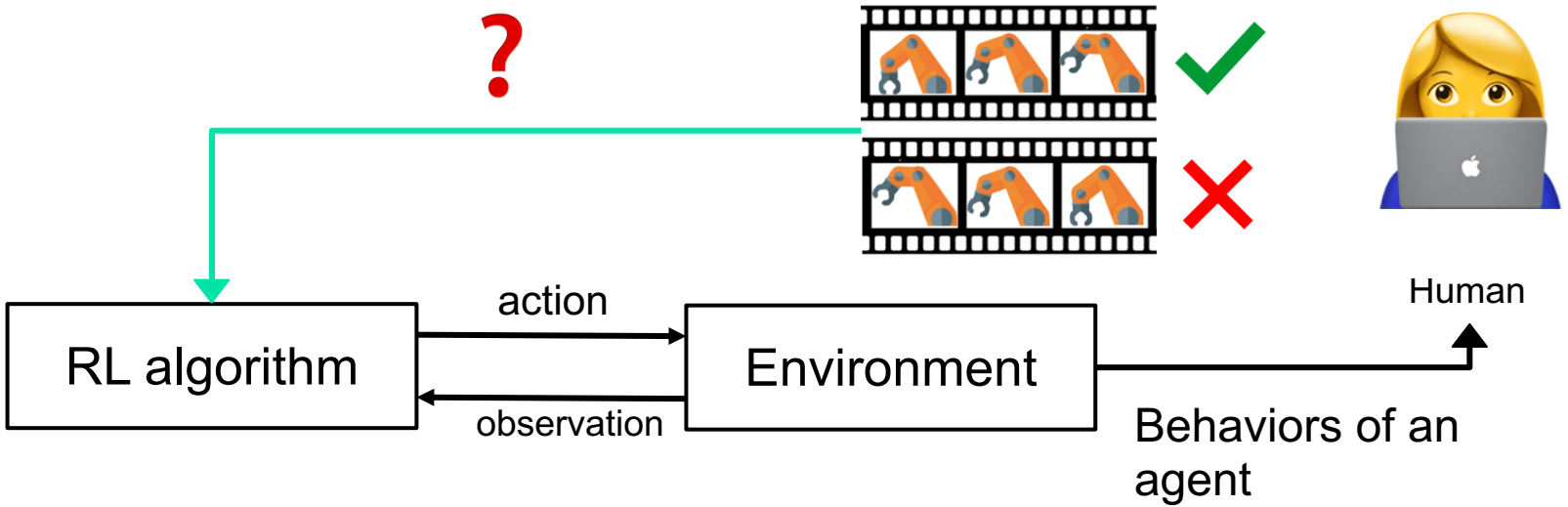
Human has ability to interactively guide agents according to their progress

→ Can teach more hard tasks, where we can't easily define the reward

→ Can avoid reward exploitation

Research Questions

Q1. How to train RL agents using preferences?



Research Questions

Q1. How to train RL agents using preferences?

Q2. How to design a feedback-efficient algorithm?

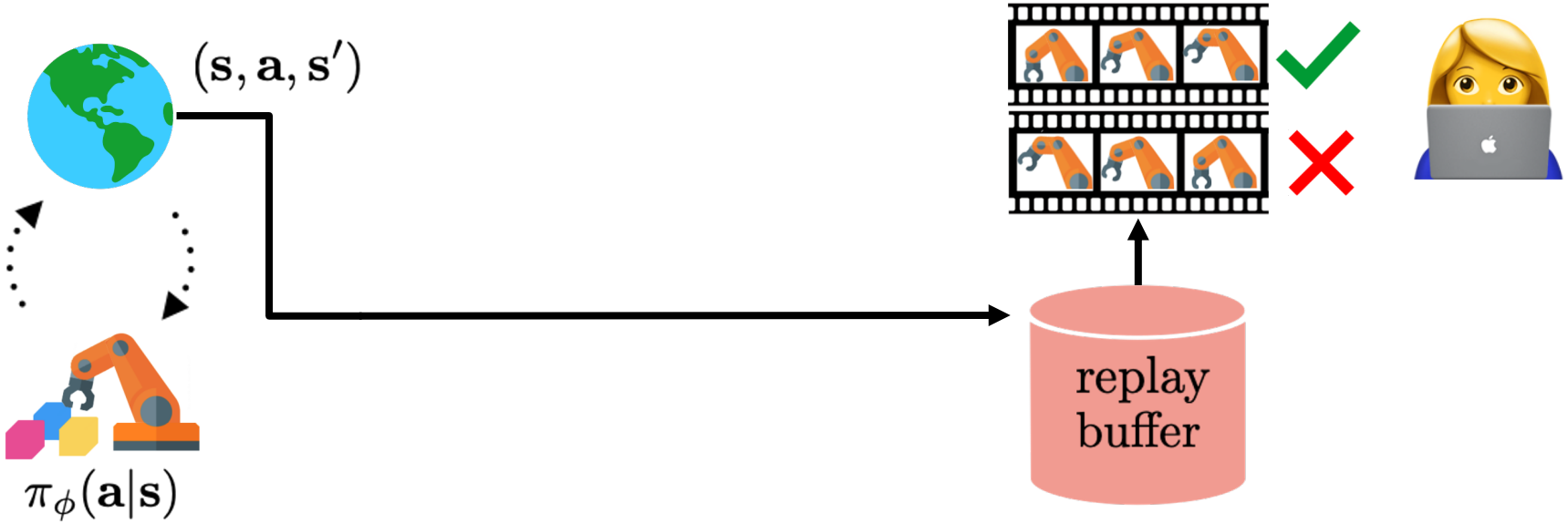
Large burden can be placed on the human!

Overall Framework

- Step 1. Collect samples via interactions with environment

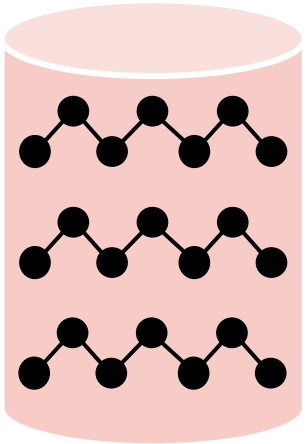
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences



Overall Framework

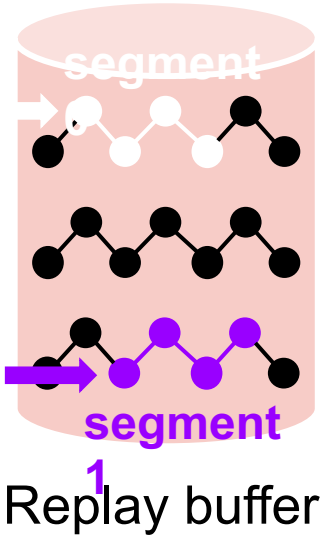
- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



Replay buffer

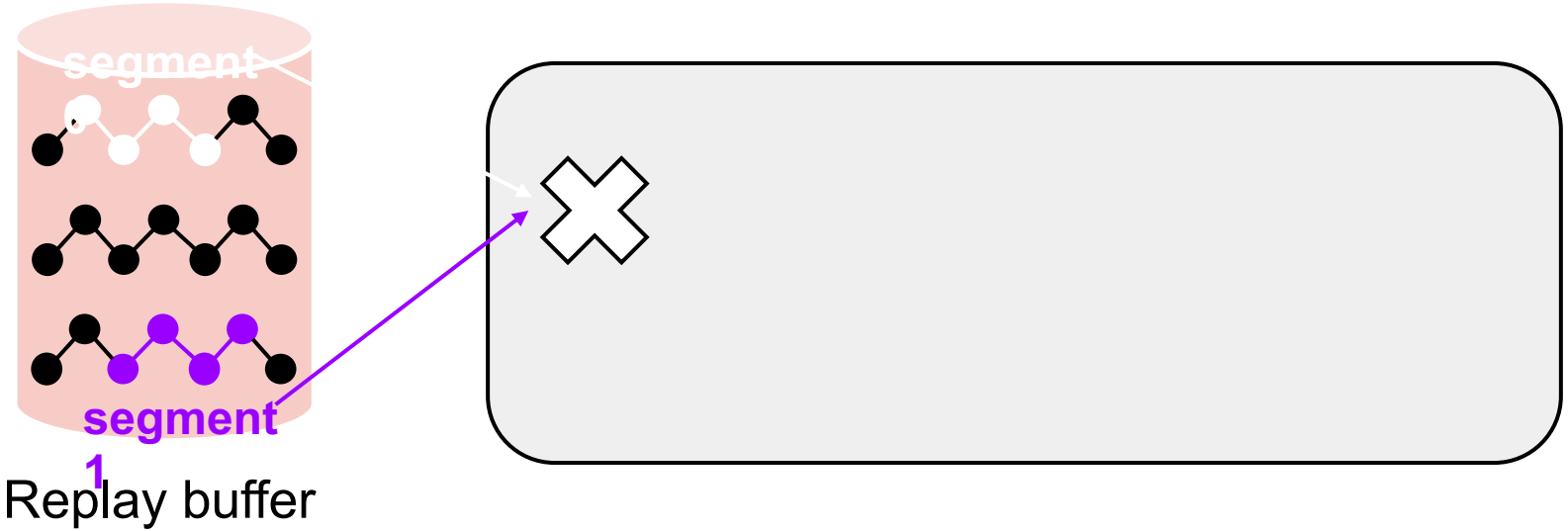
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



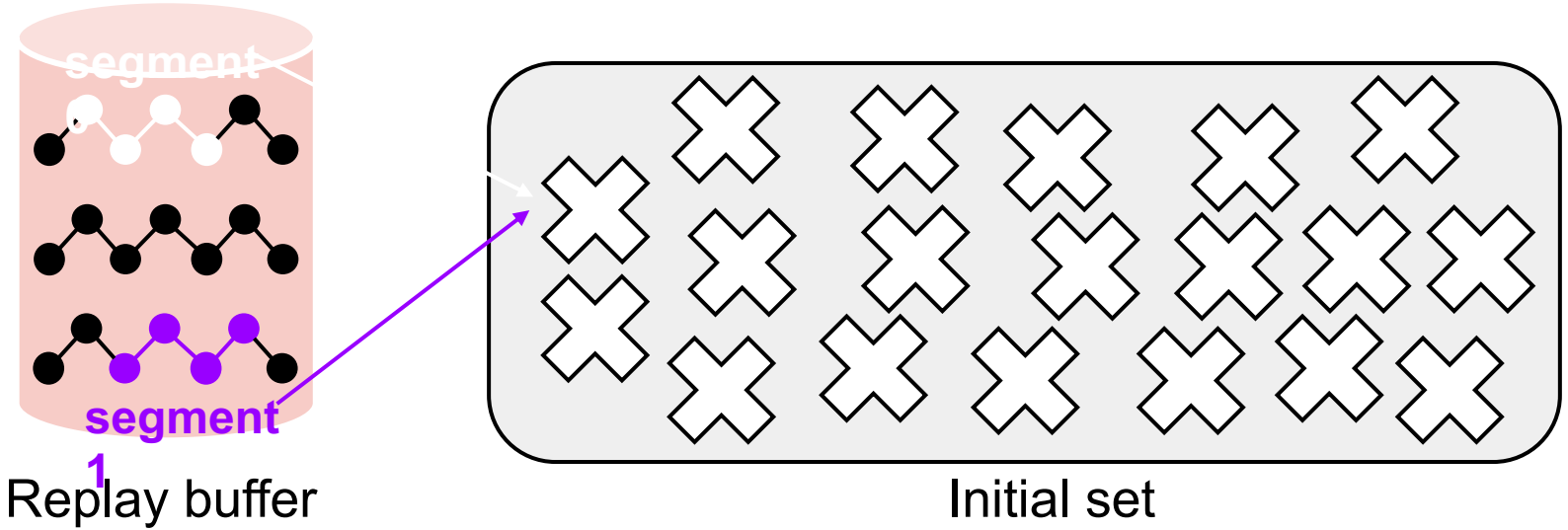
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



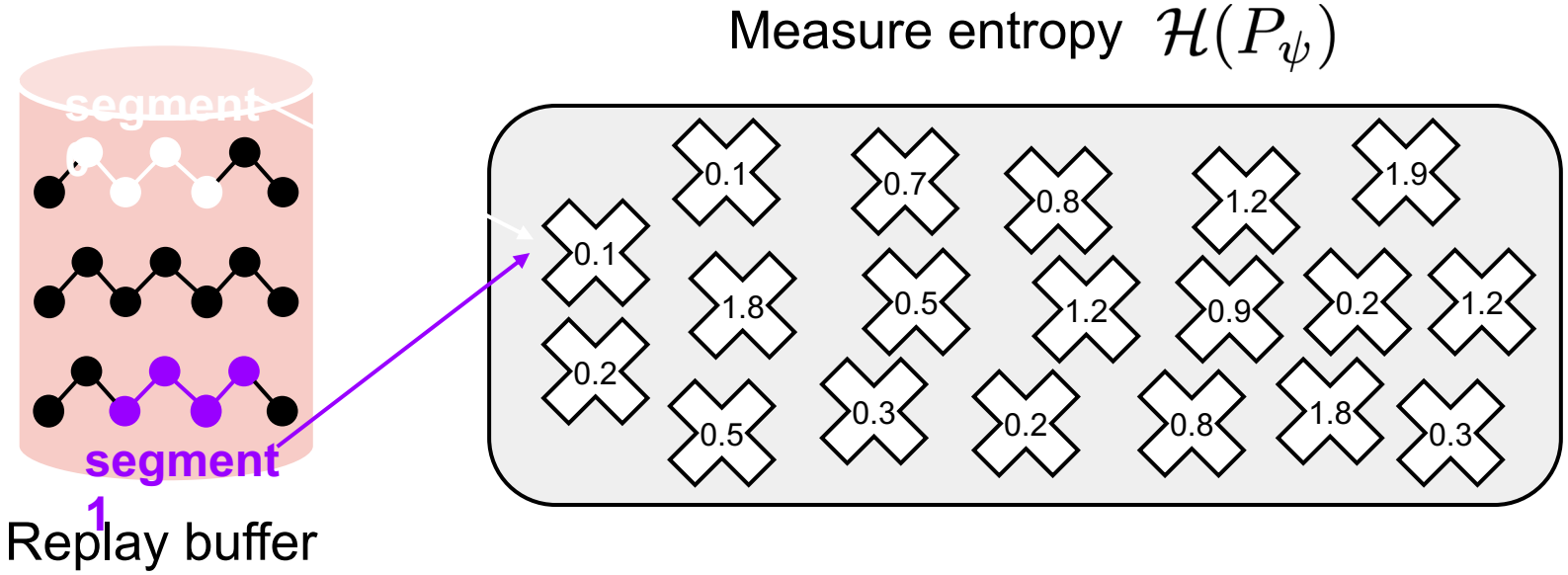
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



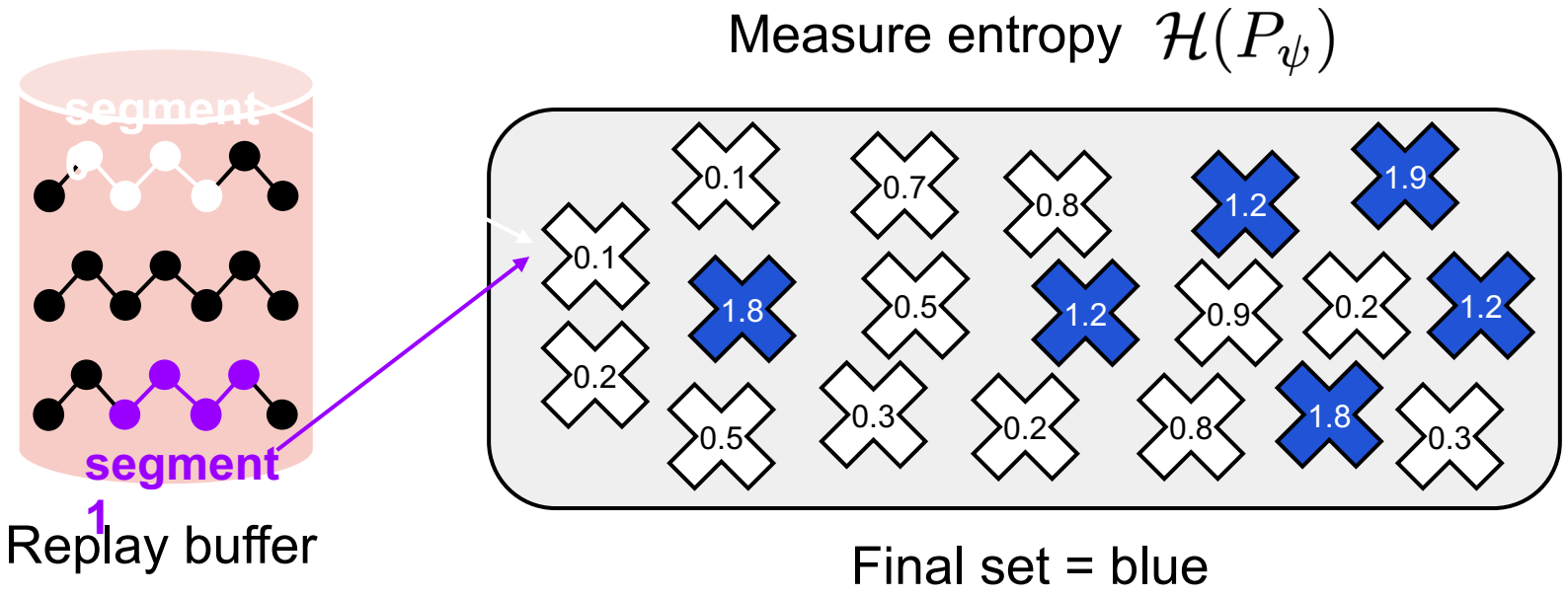
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



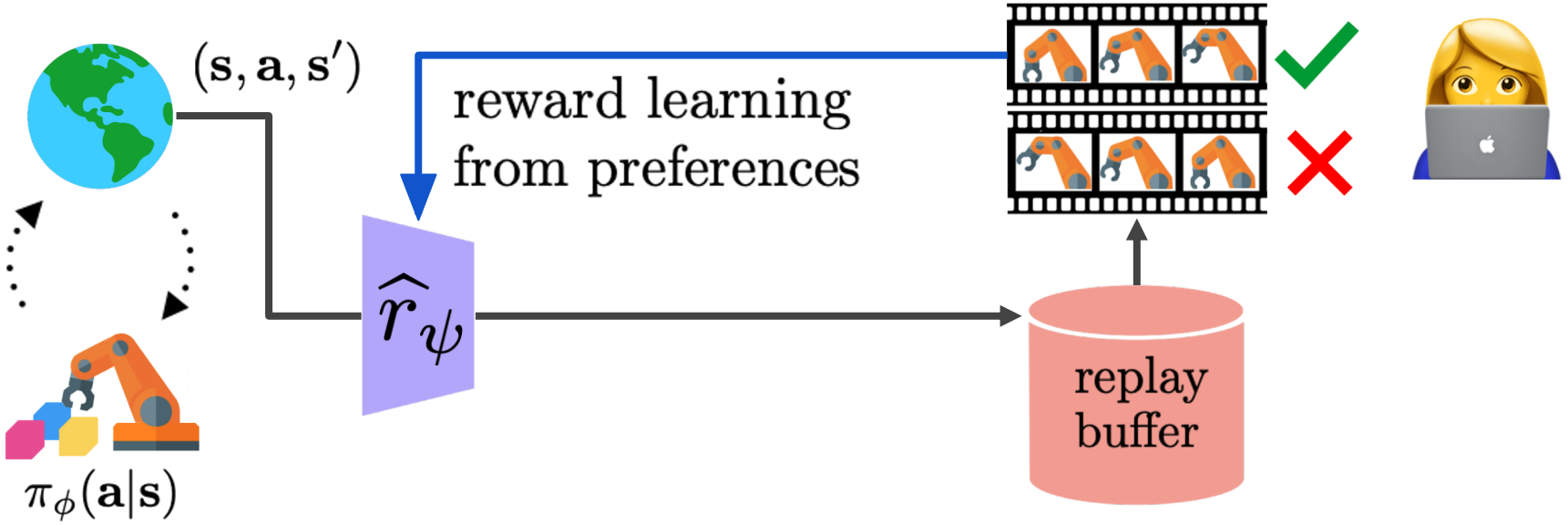
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
  - How to select two segments? Entropy sampling



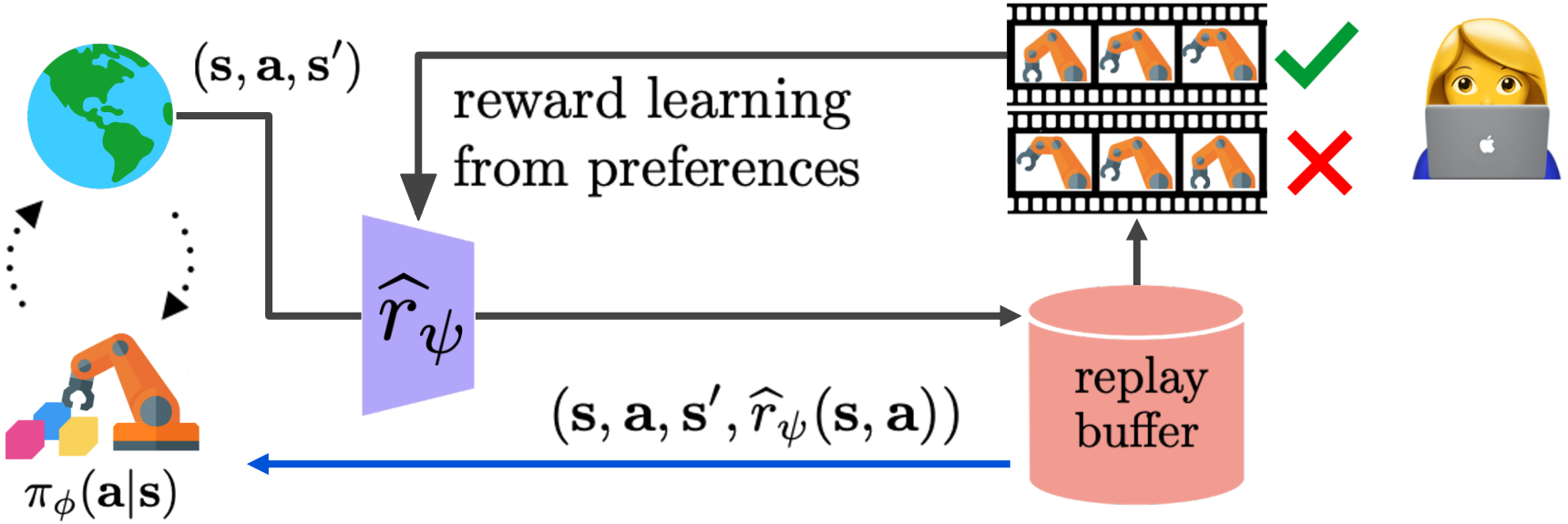
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
- Step 3. Optimize a reward model using cross entropy loss



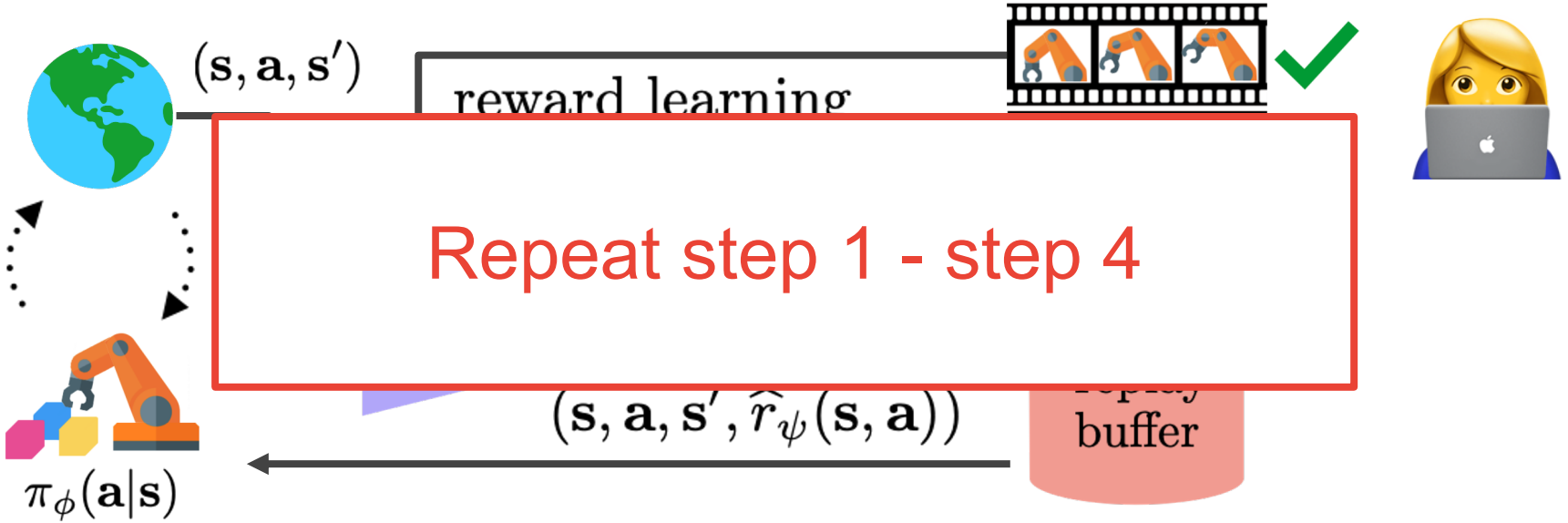
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
- Step 3. Optimize a reward model using cross entropy loss
- Step 4. Optimize a policy using off-policy algorithms



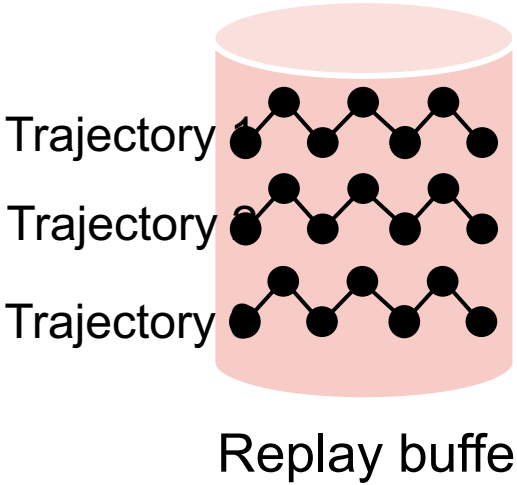
Overall Framework

- Step 1. Collect samples via interactions with environment
- Step 2. Collect human preferences
- Step 3. Optimize a reward model using cross entropy loss
- Step 4. Optimize a policy using off-policy algorithms



Learning Reward from Preferences

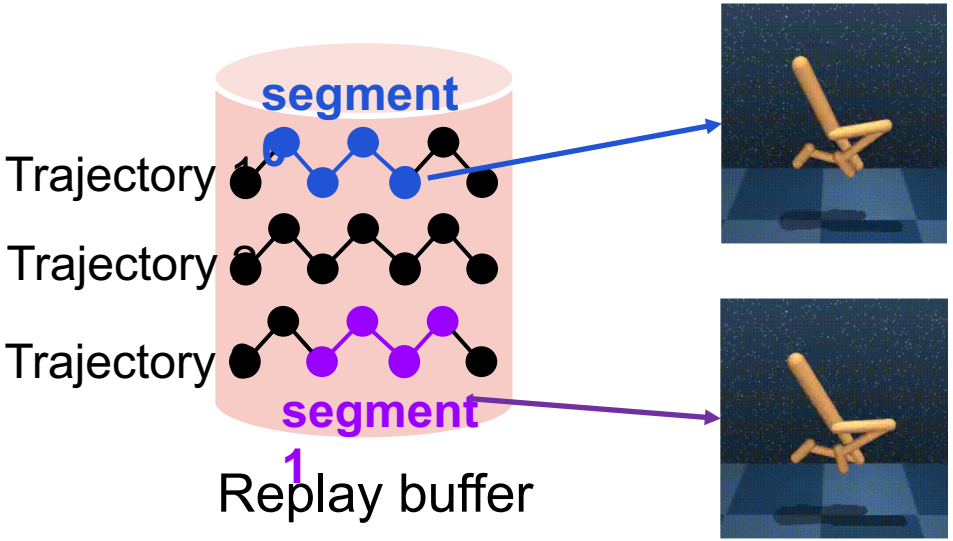
- Setup



Learning Reward from Preferences

- Setup
  - Select two segments. (σ^0, σ^1) , i.e. behaviors, from buffer

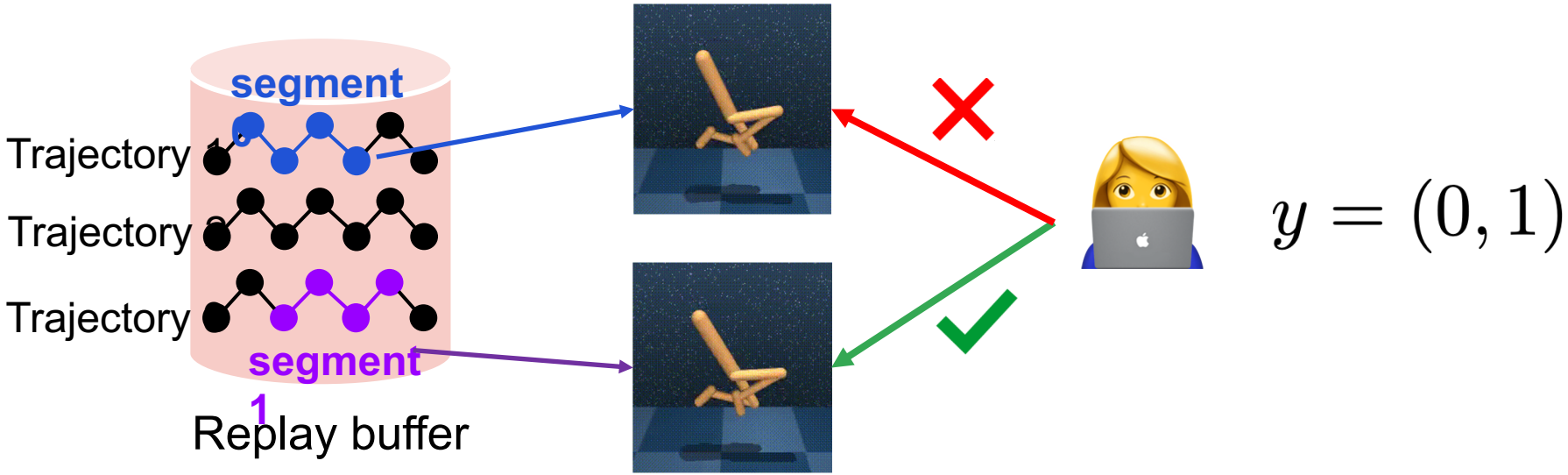
$$\sigma^i = \{(\mathbf{s}_t^I, \mathbf{a}_t^i), \cdots, (\mathbf{s}_{t+H}^I, \mathbf{a}_{t+H}^i)\}$$



Learning Reward from Preferences

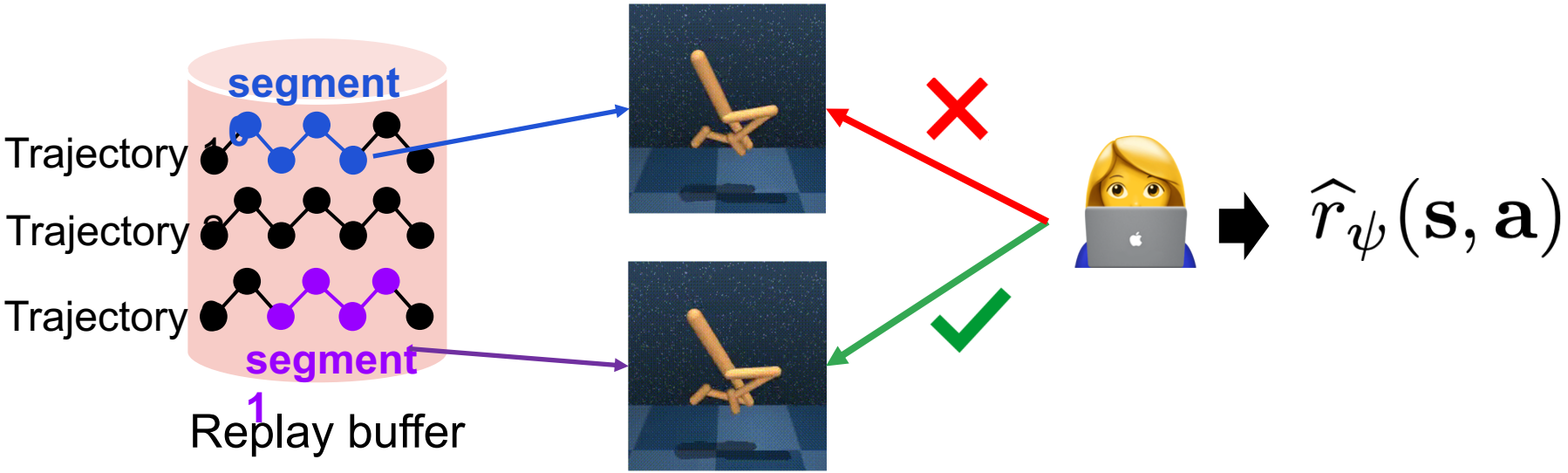
- Setup
  - Select two segments (σ^0, σ^1) , i.e. behaviors, from buffer
  - Human provides a preference as a binary label:

$$y \in \{(1,0),(0,1)\}$$



Learning Reward from Preferences

- Setup
  - Select two segments (σ^0, σ^1), i.e. behaviors, from buffer
  - Human provides a preference as a binary label
  - Goal: learning a reward model $\hat{r} : S \times A \to \mathbb{R}$



Learning Reward from Preferences

- Fitting a reward model [1]
  - Main idea: formulate this problem as a binary classification!

[1] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.
[2] Bradley, R.A. and Terry, M.E., Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika, 39(3/4), pp.324-345, 1952.

Learning Reward from Preferences

- Fitting a reward model [1]
  - Main idea: formulate this problem as a binary classification!
  - By following the Bradley-Terry model [2], we can model a **preference predictor** as follows:

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp \sum_t \widehat{r}(\mathbf{s}_t^1, \mathbf{a}_t^1)}{\sum_{i \in \{0,1\}} \exp \sum_t \widehat{r}(\mathbf{s}_t^i, \mathbf{a}_t^i)}$$

[1] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.
[2] Bradley, R.A. and Terry, M.E., Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika, 39(3/4), pp.324-345, 1952.

Learning Reward from Preferences

- Fitting a reward model [1]
  - Main idea: formulate this problem as a binary classification!
  - By following the Bradley-Terry model [2], we can model a **preference predictor** as follows:

$$P_\psi\left[\sigma^1 \succ \sigma^0\right] = \frac{\exp \sum_t \hat{r}(\mathbf{s}_t^1, \mathbf{a}_t^1)}{\sum_{i\in\{0,1\}} \exp \sum_t \hat{r}(\mathbf{s}_t^i, \mathbf{a}_t^i)}$$

Sum of rewards over segment 1

Event that segment 1 is preferable to segment 0

[1] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.
[2] Bradley, R.A. and Terry, M.E., Rank analysis of incomplete block designs: I. The method of paired comparisons. Biometrika, 39(3/4), pp.324-345, 1952.

Learning Reward from Preferences

- Fitting a reward model [1]
  - Main idea: formulate this problem as a binary classification!
  - By following the Bradley-Terry model [2], we can model a **preference predictor** as follows:

$$P_\psi[\sigma^1 \succ \sigma^0] = \frac{\exp \sum_t \widehat{r}(\mathbf{s}_t^1, \mathbf{a}_t^1)}{\sum_{i \in \{0,1\}} \exp \sum_t \widehat{r}(\mathbf{s}_t^i, \mathbf{a}_t^i)}$$

- We can learn a reward model by optimizing cross entropy:

$$\mathcal{L}^{\texttt{Reward}} = -\mathbb{E}_{(\sigma^0, \sigma^1, y)} \Big[y(0) \log P_\psi[\sigma^0 \succ \sigma^1] + y(1) \log P_\psi[\sigma^1 \succ \sigma^0] \Big].$$

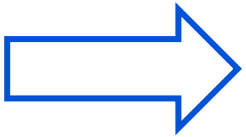
Unsupervised Pre-training: APT

- Obtaining a good initial state space coverage is important!
  - Human can’t convey much meaningful information to the agent

Unsupervised Pre-training: APT

- Obtaining a good initial state space coverage is important!
  - Human can't convey much meaningful information to the agent

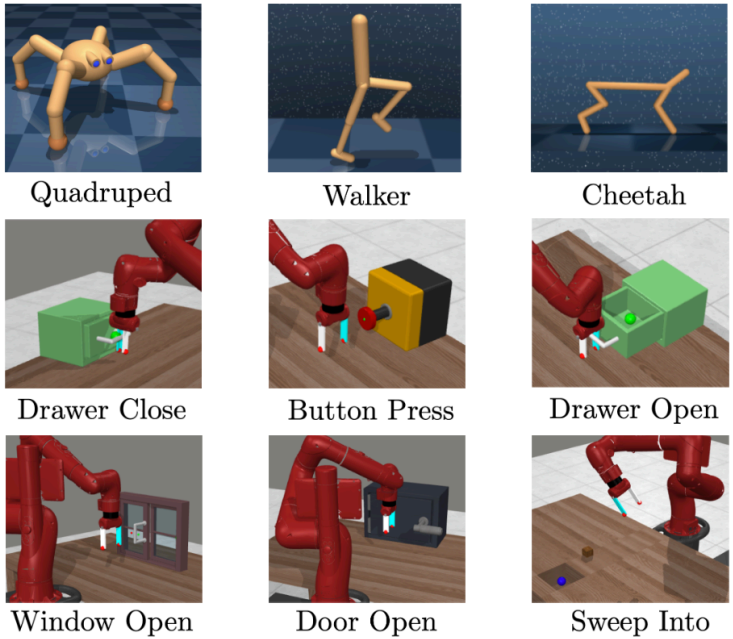
Behavior from random exploration policy



Behavior from pre-trained policy

Experiments

- Goal
  - Solving benchmark tasks **without** observing the true reward
- Tasks
  - Locomotion tasks from DMControl [1]
  - Robotic manipulation tasks from Meta-world [2]
- We will solve these tasks using as few queries as possible!
  - How to get preferences?

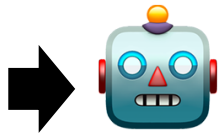


[1] Tassa, Y. et al. Deepmind control suite. arXiv preprint arXiv:1801.00690, 2018.
[2] Yu, T. et al, Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In CoRL, 2020.

Experiments

- We generate preferences using a scripted teacher [1, 2]:

$$\sigma^0 = \{(s_t^0, a_t^0), \cdots, (s_{t+H}^0, a_{t+H}^0)\}$$
$$\sigma^1 = \{(s_t^1, a_t^1), \cdots, (s_{t+H}^1, a_{t+H}^1)\}$$



$$y = 1 \quad \text{if} \sum_t r^* \left(\mathbf{s}_t^1, \mathbf{a}_t^1\right) > \sum_t r^* \left(\mathbf{s}_t^0, \mathbf{a}_t^0\right)$$

True task rewar

Preferences are immediately generated
→ more rapid experiments

We can evaluate the agent quantitatively by measuring the true average return

[1] Ibarz, B., Leike, J., Pohlen, T., Irving, G., Legg, S. and Amodei, D., Reward learning from human preferences and demonstrations in atari. In NeurIPS, 2018.
[2] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.

Tested Algorithms

- **PEBBLE (ours)**

  - Unsupervised pre-training (yellow)
    - Collect initial samples via unsupervised pre-training

Tested Algorithms

- **PEBBLE (ours)**

  - Unsupervised pre-training (yellow)
    - Collect initial samples via unsupervised pre-training
  - Learn a reward (red)
    - Get feedback (preference btw two behaviors) from a teacher
    - Learn a reward model

Tested Algorithms

- **PEBBLE (ours)**

  - Unsupervised pre-training (yellow)
    - Collect initial samples via unsupervised pre-training
  - Learn a reward (red)
    - Get feedback (preference btw two behaviors) from a teacher
    - Learn a reward model
  - Interaction (blue)
    - Interact with environment (e.g. 20 episodes)
    - Update policy using off-policy RL (SAC [1])

[1] Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S., Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In ICML, 2018.

Tested Algorithms

- **PEBBLE (ours)**

- **Preference-PPO [1]**

  - No unsupervised pre-training → random exploration (green)
  - Other parts (red & blue) are same except uses on-policy RL (PPO [2]) for updating policy

[1] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.
[2] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

Tested Algorithms

- **PEBBLE (ours)**

- **Preference-PPO [1]**

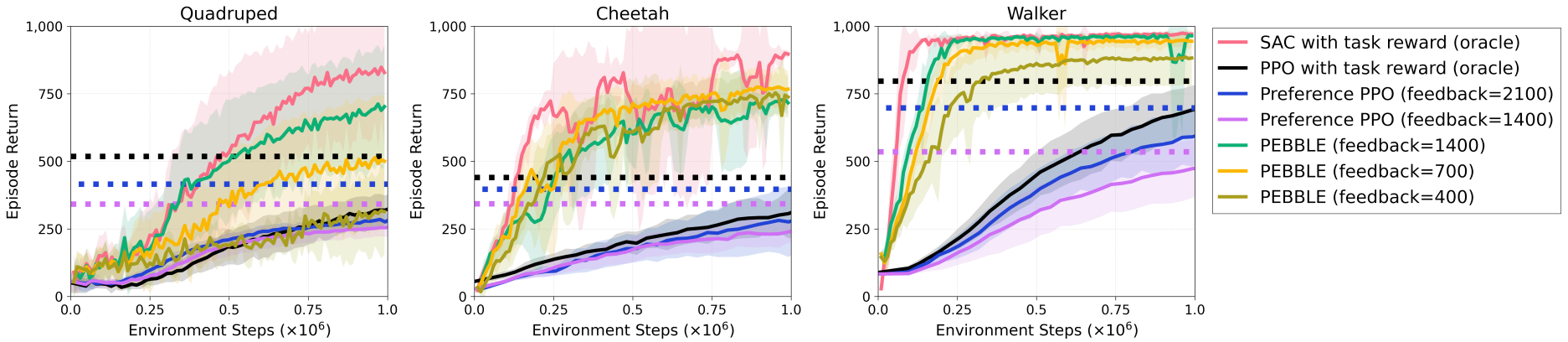
  - No unsupervised pre-training → random exploration (green)
  - Other parts (red & blue) are same except uses on-policy RL (PPO [2]) for updating policy

- **RL agents trained with true rewards (upper bound)**

[1] Christiano, P., Leike, J., Brown, T.B., Martic, M., Legg, S. and Amodei, D., Deep reinforcement learning from human preferences. NeurIPS, 2017.
[2] Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O., Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

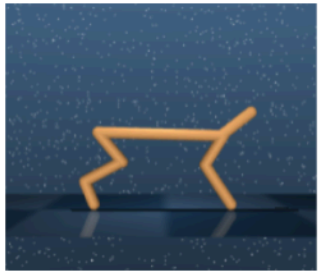
Locomotion Tasks

- Learning curves (10 random seeds)

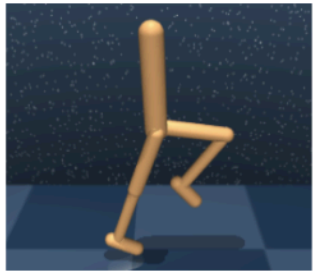


* Asymptotic performance of PPO and Preference PPO is indicated by dotted lines of the corresponding color

Quadruped



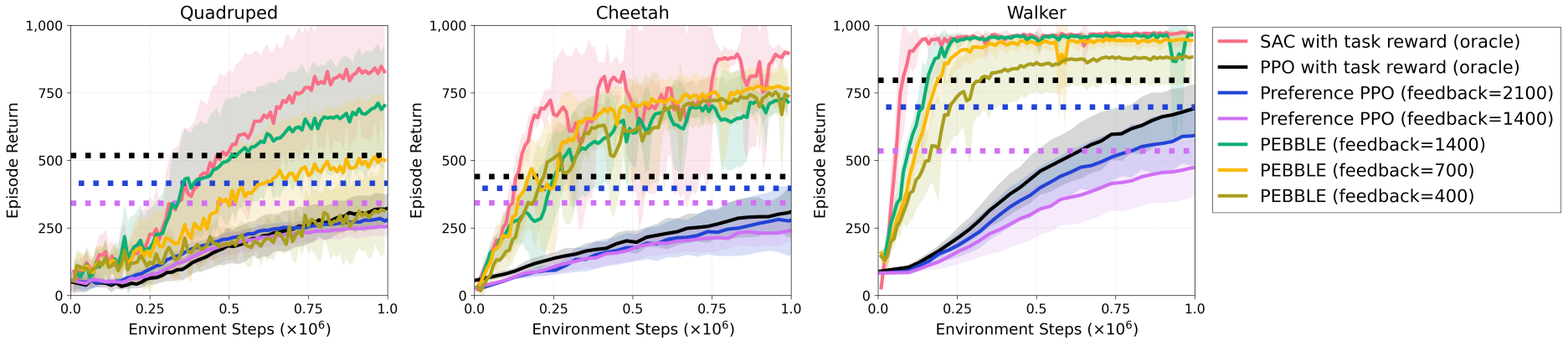
Cheetah



Walker

Locomotion Tasks

- Learning curves (10 random seeds)



* Asymptotic performance of PPO and Preference PPO is indicated by dotted lines of the corresponding color

Given a budget of 1400 queries,

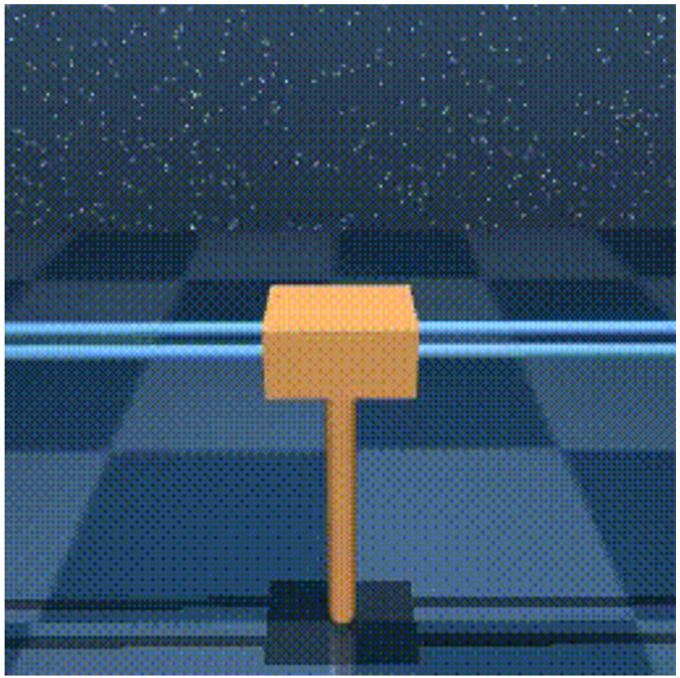
- PEBBLE (green) reaches the same performance as SAC (pink)
- Preference PPO (purple) is unable to match PPO (black)

PEBBLE is indeed more feedback-efficient

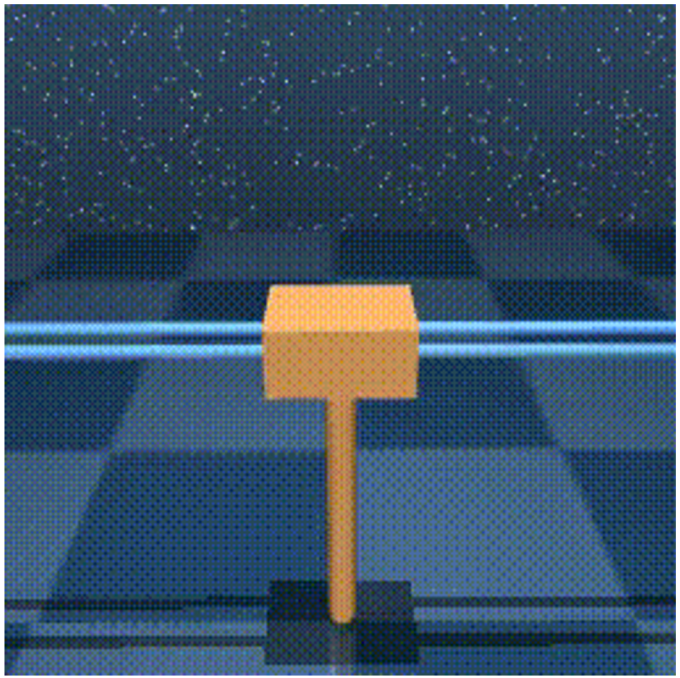
Can Human Teach Novel Behaviors?

Can Human Teach Novel Behaviors?

- 40 queries in less than 5 mins



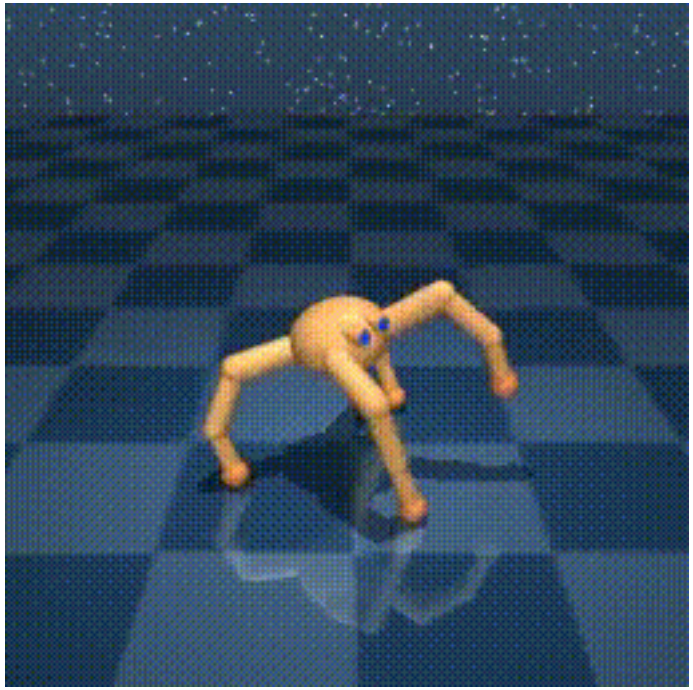
Counter clockwise



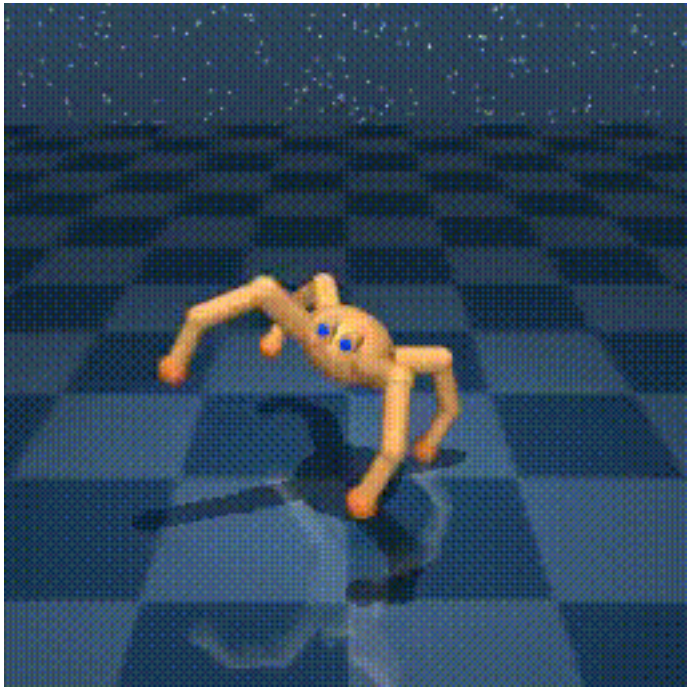
Clockwise

Can Human Teach Novel Behaviors?

- 200 queries in less than 30 mins



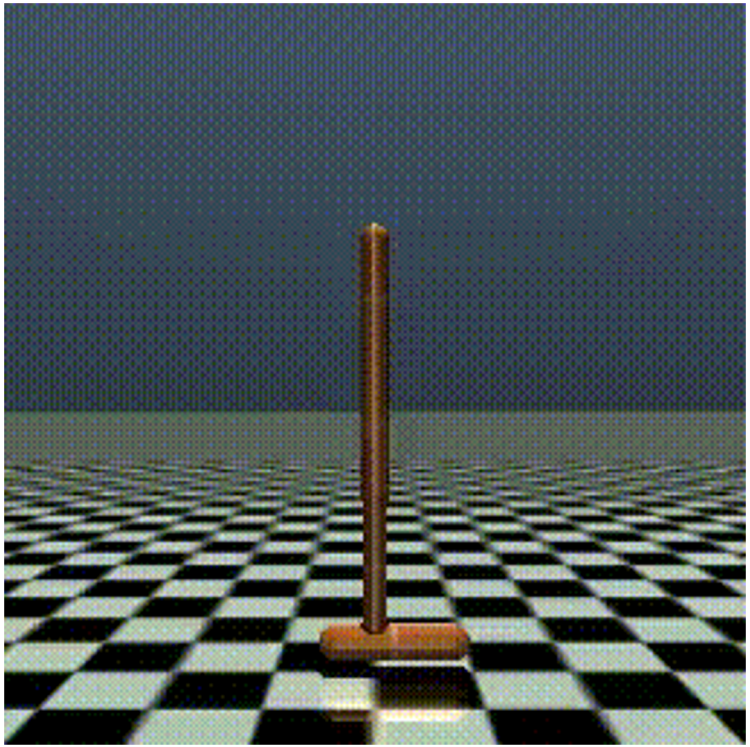
Waving left front leg



Waving right front leg

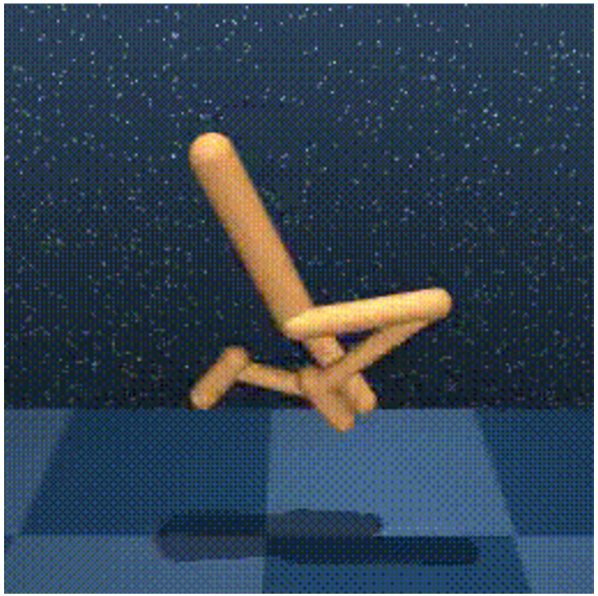
Can Human Teach Novel Behaviors?

- 400 queries in less than one hour



Can We Avoid Reward Exploitation?

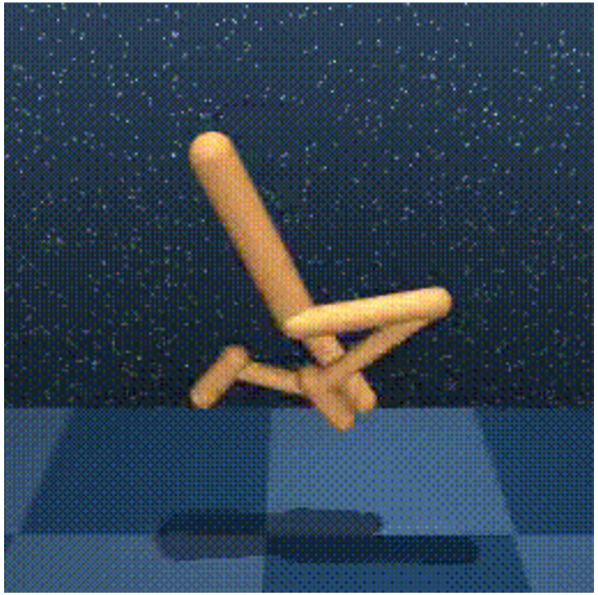
Can We Avoid Reward Exploitation?



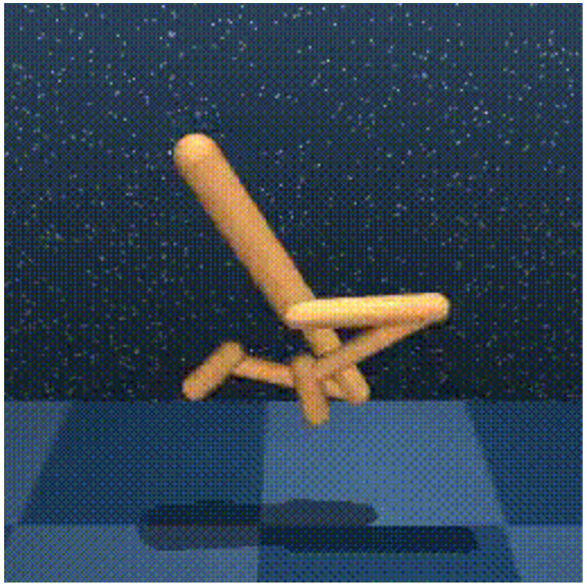
SAC with task reward on walker, walk
(use one leg even if score ~=1000)

Can We Avoid Reward Exploitation?

- 150 queries in less than 20 mins



SAC with task reward on walker, walk
(use one leg even if score ~=1000)

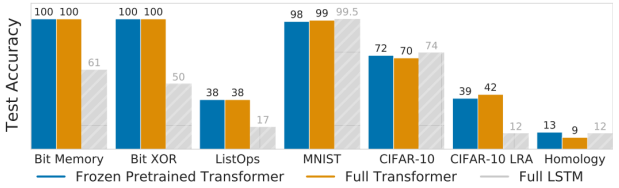


SAC trained with human feedback
(use both legs)

Summary

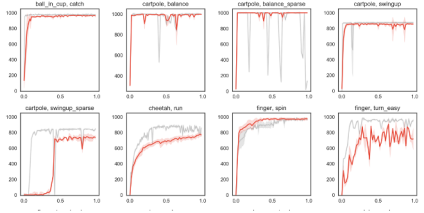
- NN Architectures

  Pretrained Transformers as Universal Computation Engines
  Kevin Lu, Aditya Grover, Pieter Abbeel, Igor Mordatch



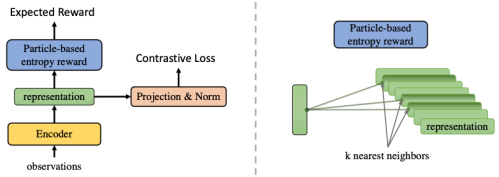
- Representation Learning for Reinforcement Learning

  CURL: Contrastive Unsupervised Representation Learning for RL
  Misha Laskin*, Aravind Srinivas*, Pieter Abbeel



- Unsupervised Skill Discovery

  Behavior from the Void: Unsupervised Active Pre-Training
  Hao Liu, Pieter Abbeel



- Human-in-the-loop Reinforcement Learning

  PEBBLE: Feedback-Efficient Interactive RL via Relabeling Experience and Unsupervised Pre-training
  Kimin Lee*, Laura Smith*, Pieter Abbeel