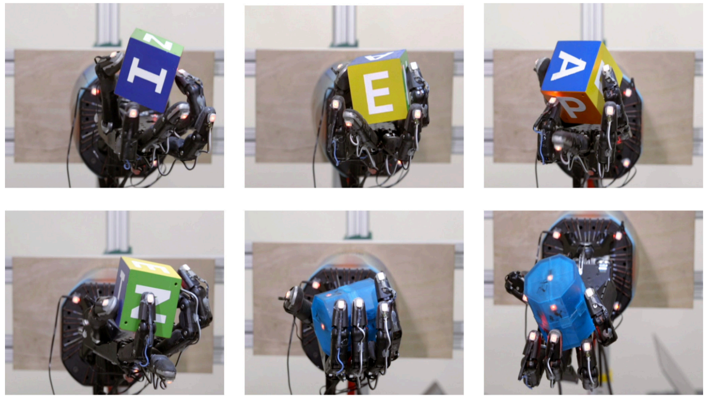
APS: Active Pretraining with Successor Features

Hao Liu, Pieter Abbeel

Berkeley
UNIVERSITY OF CALIFORNIA

BAIR
BERKELEY ARTIFICIAL INTELLIGENCE RESEARCH

Big picture and key challenges



Reinforcement Learning: task specific, difficult to generalize to new tasks

https://openai.com/blog/solving-rubiks-cube/
https://openai.com/projects/five/
https://deepmind.com/blog/article/Agent57-Outperforming-the-human-Atari-benchmark

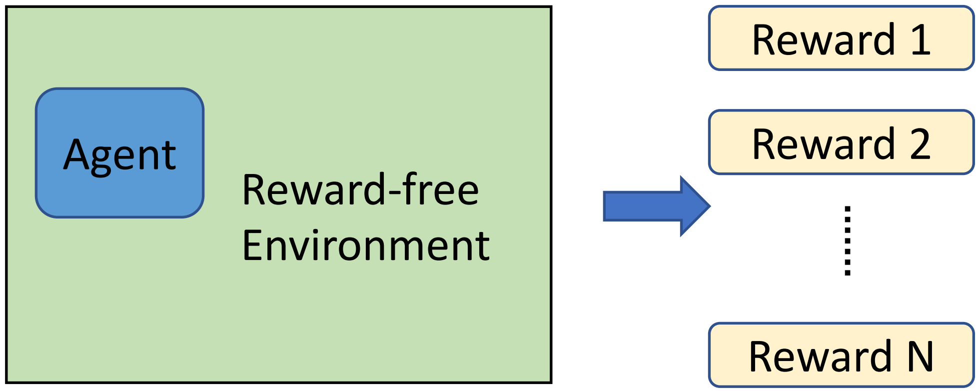
Generalization to new tasks

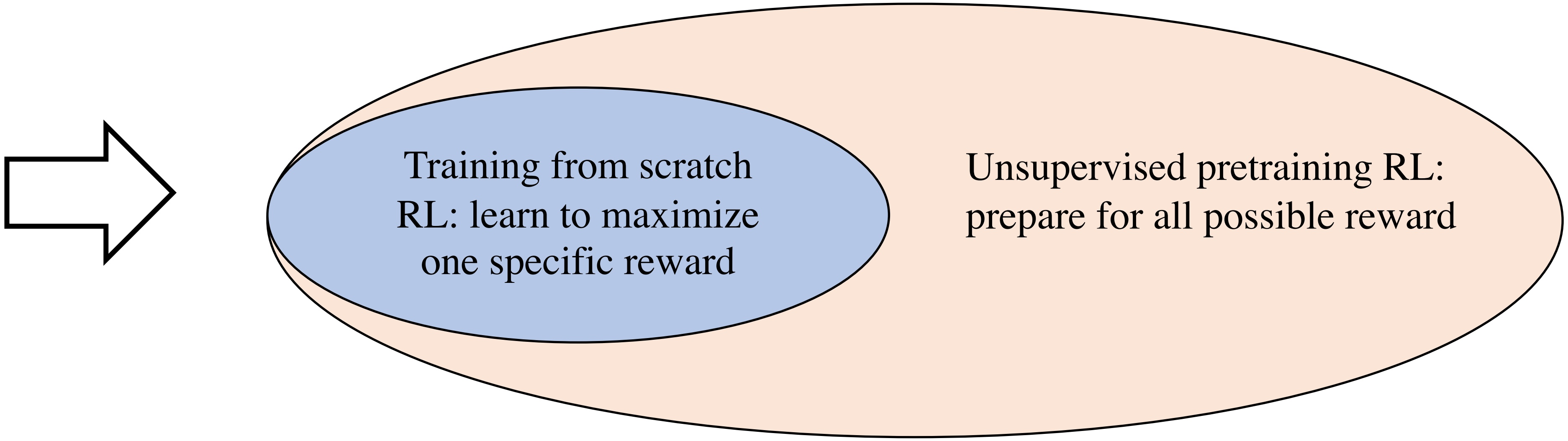
- Vision (SimCLRv2, DINO):
  - Pre-train on ImageNet -> finetune for other tasks
- NLP (GPT-*, BERT):
  - Pre-train on internet text -> finetune for other tasks
- Reinforcement Learning:
  - ???????????????????? -> finetune for other tasks

Problem setting: Open-ended Environments

- Pretraining without accessing environment reward function

=> Finetuning on different downstream reward functions





Variational Approximation as Intrinsic Reward

- Prior work aim to maximize MI between states and some conditioning variables

$$I(s; z) = \mathcal{H}(z) - \mathcal{H}(z|s)$$

$$-\mathcal{H}(z|s) \geq \mathbb{E}_{\pi_z}[\log q(z|s)]$$

Gregor, K., Rezende, D. J., & Wierstra, D. (2016). Variational intrinsic control.

Insufficient Exploration

- For usual decomposition of mutual information

$$I(s;z) = \mathcal{H}(z) - \mathcal{H}(z|s)$$

⇨

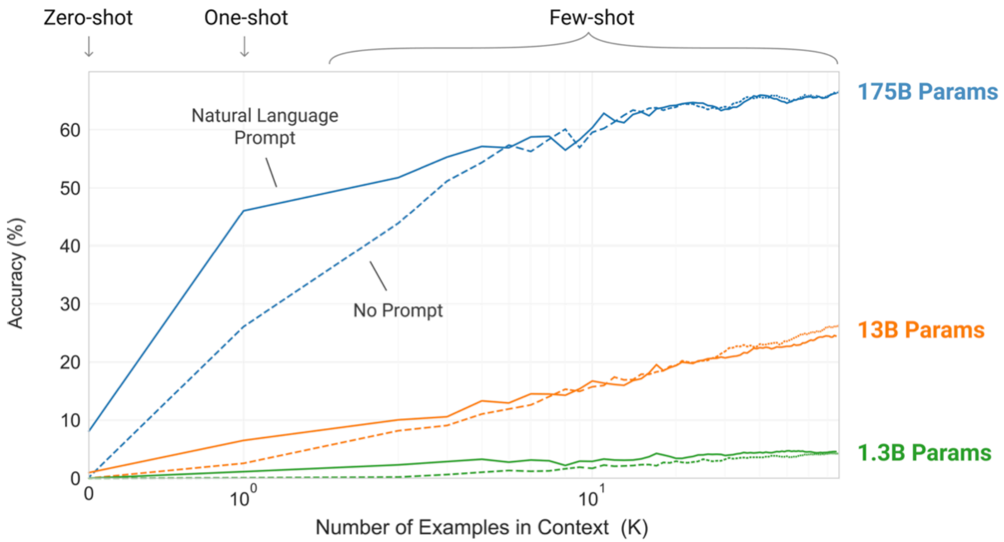
Predict latent variables from states

No incentive to explore

Gregor, K., Rezende, D. J., & Wierstra, D. (2016). Variational intrinsic control.

Scaling Law

- Large amount of data is important for unsupervised pretraining



Explore the environment

But how does the agent know where to explore

Brown, Tom B., et al. "Language models are few-shot learners."

Intrinsic Reward for Entropy Maximization

Incentivizing exploration by introducing intrinsic rewards based on a measure of state novelty

State entropy as intrinsic reward

$$\mathcal{H}(s) = -\mathbb{E}_{s\sim p(s)}\left[\log p(s)\right]$$

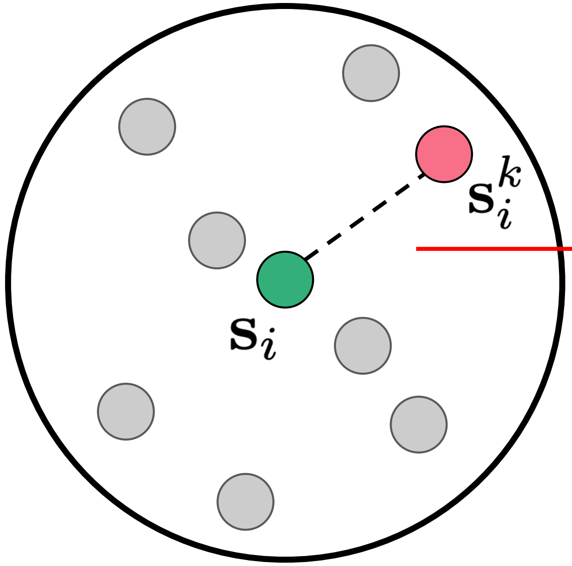
⇨ **Measuring state entropy is intractable** to compute in most setting

K-Nearest-Neighbor Entropy Estimation

- K-nearest entropy estimator [1], asymptotically consistent and unbiased

$$\mathcal{H}(s) = -\mathbb{E}_{s\sim p(s)}\left[\log p(s)\right]$$

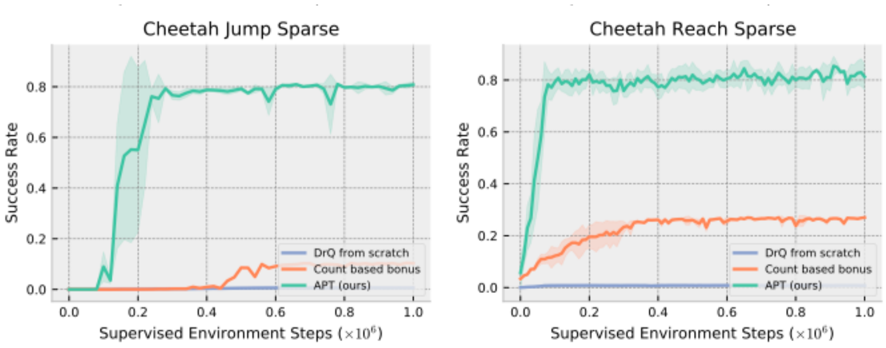
$$\widehat{\mathcal{H}}(\mathbf{s}) \propto \sum_i \log(||\mathbf{s}_i - \mathbf{s}_i^k||)$$

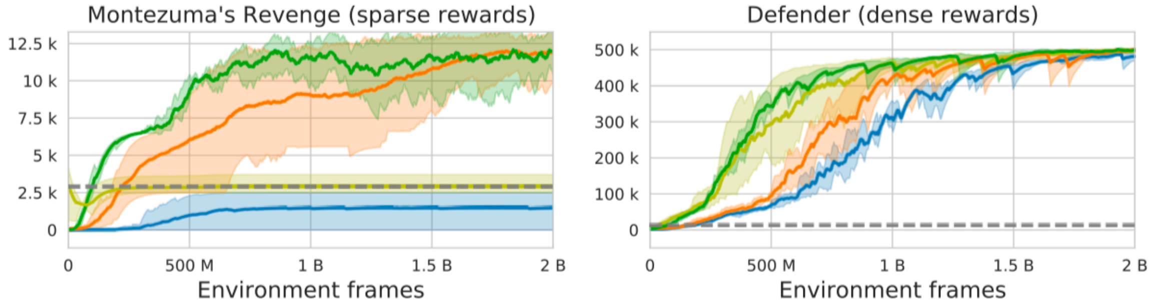


[1] Singh, H., et al., 2003. Nearest neighbor estimates of entropy. *American journal of mathematical and management sciences*, *23*(3-4), pp.301-321.

K-NN Entropy Pretraining is Powerful

- Finetuning last few layers of pretrained model significantly outperform training from scratch [1, 2, 3, 4, 5]





[1] APT: Behavior From the Void: Unsupervised Active Pre-Training, Liu & Abbeel, 2020

[2] MEPOL: Task-Agnostic Exploration via Policy Gradient of a Non-Parametric State Entropy Estimate, Mutti et al, 2020

[3] CPT: Coverage as a Principle for Discovering Transferable Behavior in Reinforcement Learning, Campos et al, 2021

[4] ProtoRL: Reinforcement Learning with Prototypical Representations, Yarats et al, 2021

[5] RE3: State Entropy Maximization with Random Encoders for Efficient Exploration, Seo*, Chen*, et al, 2021

One pretrained model for many tasks

- Prior work on entropy maximization pretraining finetune the model for each task

Finetuning for each downstream task reward function is expensive and inefficient

Explicit Entropy Maximization in MI

- Decomposing mutual information into explicit exploration and exploitation

$$I(s;z) = \mathcal{H}(s) - \mathcal{H}(s|z)$$

  - Exploring by particle-based entropy
  - Learning latent variable conditioned policy

- Intrinsic reward consists of entropy exploration and learning latent skills

$$\text{intrinsic reward} = r_{\text{entropy}} + r_{\text{sklll}}$$

Successor Features

- Using successor features [1, 2] as parameterization

$$-\mathcal{H}(s|z) \geq \mathbb{E}_{\pi_z}[\log q(s|z)] = \mathbb{E}_{\pi_z}[\phi(s)^\top z]$$

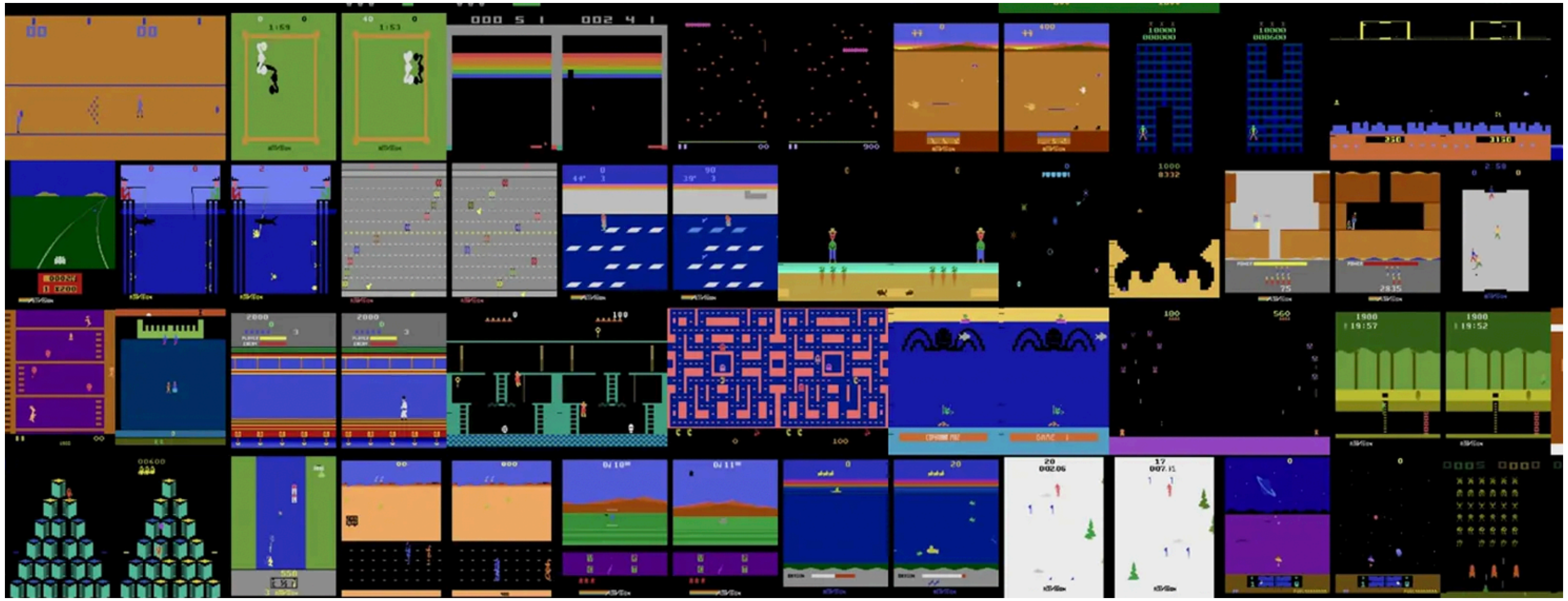
$$\begin{aligned} Q^\pi(s,a) = \ & \mathbb{E}_{a_t=s, a_t=a}\left[\sum_{i=t}^{\infty} \gamma^{i-t} \phi\big(s_{i+1}, a_{i+1}, s'_{i+1}\big)\right]^\top z \\ & \equiv \psi^\pi(s,a)^\top z \end{aligned}$$

Quickly adapt to new reward by identifying downstream task by linear regression

[1] Successor features for transfer in reinforcement learning. Barreto et al.
[2] Fast Task Inference with Variational Intrinsic Successor Features. Hansen et al.

Evaluation Setting

- Unsupervised pretraining for 200M steps per env without environment reward function
- Data efficiency benchmark: agents are allowed only 100k steps which is ~2 hours of real-time gameplay



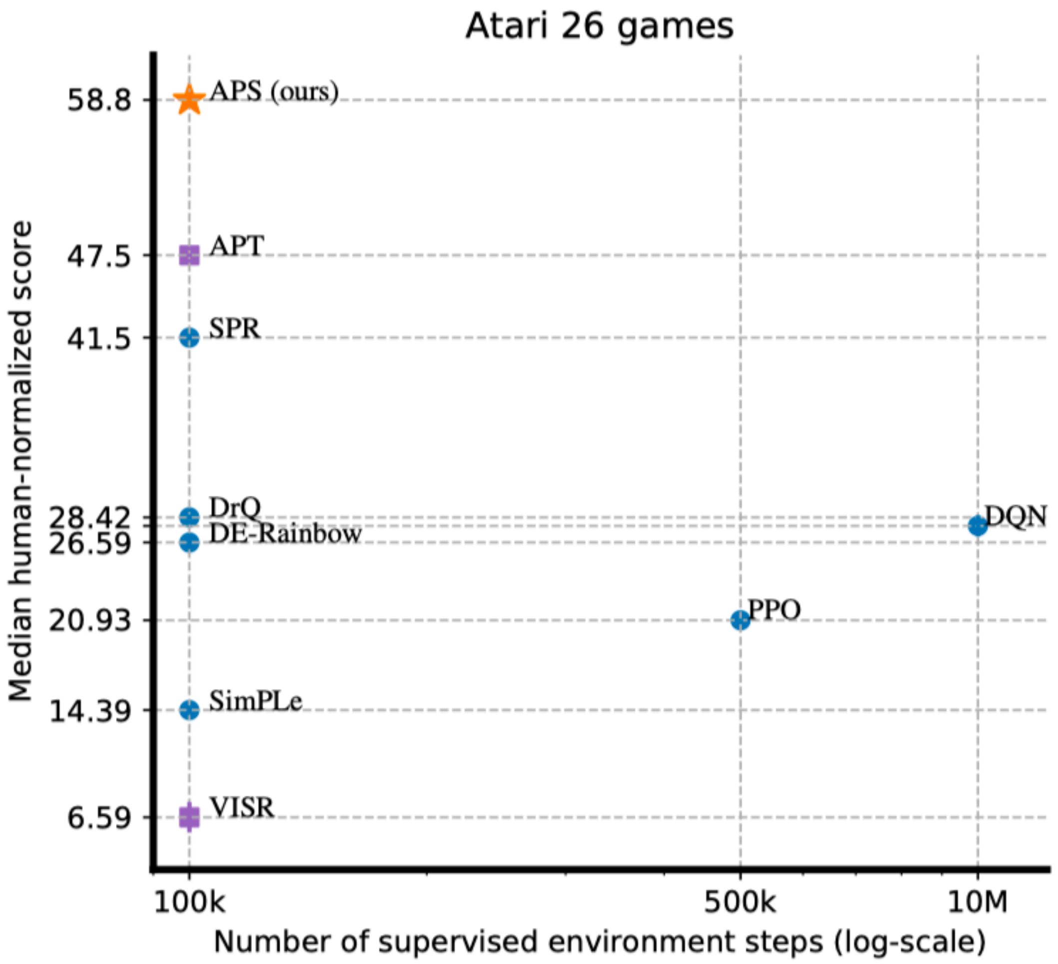
Results

SOTA RL train fro $I(s; z) = \mathcal{H}(z) - \mathcal{H}$ $\mathcal{H}(s)$ based based Ours

Game	Random	Human	SimPLe	DER	CURL	DrQ	SPR	VISR	APT	APS (ours)
Alien	227.8	7127.7	616,9	739.9	558.2	771.2	801.5	364.4	**2614.8**	934.9
Amidar	5.8	1719.5	88.0	188.6	142.1	102.8	176.3	186.0	**211.5**	178.4
Assault	222.4	742.0	527.2	431.2	600.6	452.4	571.0	**12091.1**	891.5	413.3
Asterix	210.0	8503.3	1128.3	470.8	734.5	603.5	977.8	**6216.7**	185.5	1159.7
Bank Heist	14.2	753.1	34.2	51.0	131.6	168.9	380.9	71.3	**416.7**	262.7
BattleZone	2360.0	37187.5	5184.4	10124.6	14870.0	12954.0	16651.0	7072.7	7065.1	**26920.1**
Boxing	0.1	12.1	9.1	0.2	1.2	6.0	35.8	13.4	21.3	**36.3**
Breakout	1.7	30.5	16.4	1.9	4.9	16.1	17.1	17.9	10.9	**19.1**
ChopperCommand	811.0	7387.8	1246.9	861.8	1058.5	780.3	974.8	800.8	317.0	**2517.0**
Crazy Climber	10780.5	23829.4	62583.6	16185.2	12146.5	20516.5	42923.6	49373.9	44128.0	**67328.1**
Demon Attack	107805	35829.4	62583.6	16185.3	12146.5	20516.5	42923.6	**8994.9**	5071.8	7989.0
Freeway	0.0	29.6	20.3	27.9	26.7	9.8	24.4	-12.1	**29.9**	27.1
Frostbite	65.2	4334.7	254.7	866.8	1181.3	331.1	**1821.5**	230.9	1796.1	496.5
Gopher	257.6	2412.5	771.0	349.5	669.3	636.3	715.2	498.6	**2590.4**	2386.5
Hero	1027.0	30826.4	2656.6	6857.0	6279.3	3736.3	7019.2	663.5	6789.1	**12189.3**
Jamesbond	29.0	302.8	125.3	301.6	471.0	236.0	365.4	484.4	356.1	**622.3**
Kangaroo	52.0	3035.0	323.1	779.3	872.5	940.6	3276.4	1761.9	412.0	**5280.1**
Krull	1598.0	2665.5	**4539.9**	2851.5	4229.6	4018.1	2688.9	3142.5	2312.0	4496.0
Kung Fu Master	258.5	22736.3	17257.2	14346.1	14307.8	9111.0	13192.7	16754.9	17357.0	**22412.0**
Ms Pacman	307.3	6951.6	1480.0	1204.1	1465.5	960.5	1313.2	558.5	**2827.1**	2092.3
Pong	-20.7	14.6	**12.8**	-19.3	-16.5	-8.5	-5.9	-26.2	-8.0	12.5
Private Eye	24.9	69571.3	58.3	97.8	218.4	-13.6	**124.0**	98.3	96.1	117.9
Qbert	163.9	13455.0	1288.8	1152.9	1042.4	854.4	669.1	666.3	17671.2	**19271.4**
Road Runner	11.5	7845.0	5640.6	9600.0	5661.0	8895.1	**14220.5**	6146.7	4782.1	5919.0
Seaquest	68.4	42054.7	683.3	354.1	384.5	301.2	583.1	706.6	2116.7	**4209.7**
Up N Down	533.4	11693.2	3350.3	2877.4	2955.2	3180.8	**28138.5**	10037.6	8289.4	4911.9
Mean Human-Norm'd	0.000	1.000	44.3	28.5	38.1	35.7	70.4	64.31	69.55	**99.04**
Median Human-Norm'd	0.000	1.000	14.4	16.1	17.5	26.8	41.5	12.36	47.50	**58.80**
# Superhuman	0	N/A	2	2	2	2	7	6	7	**8**

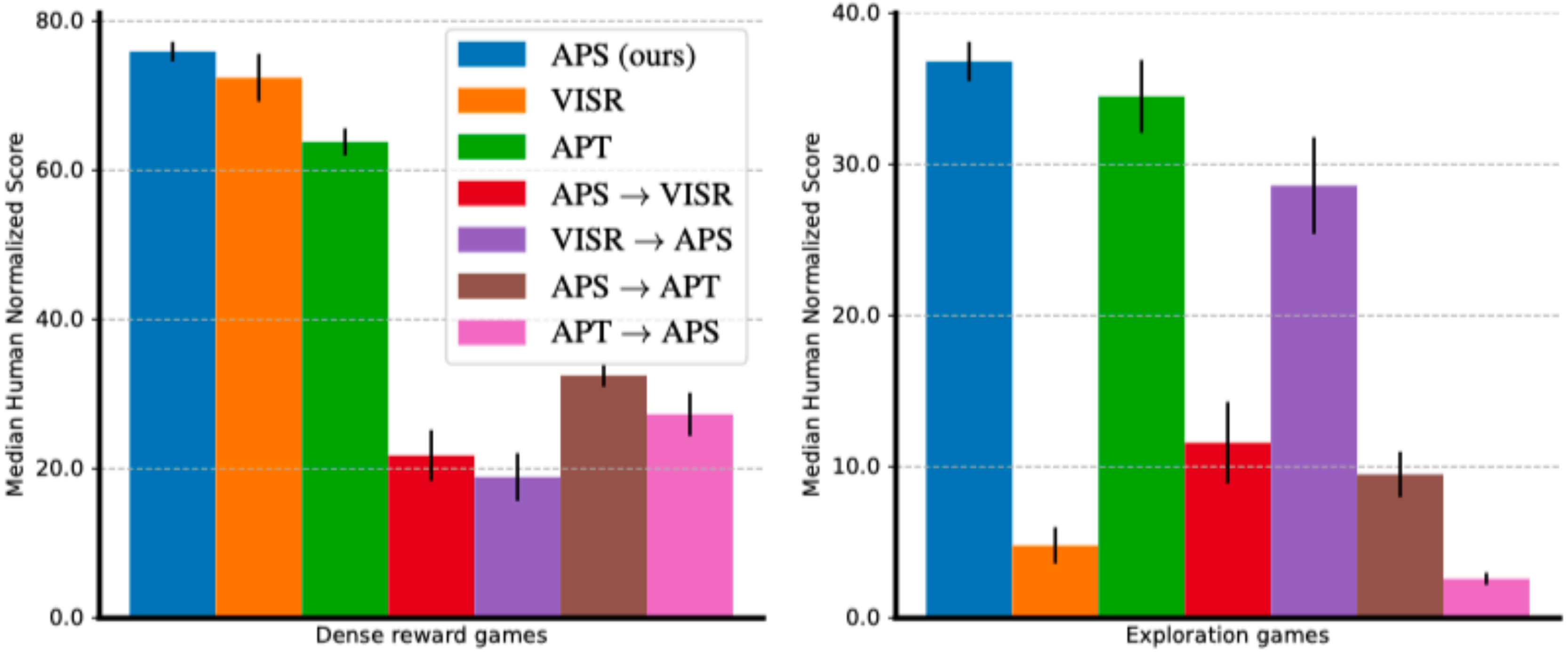
Result

- Unsupervised pretraining outperforms training from scratch using fewer number of interactions

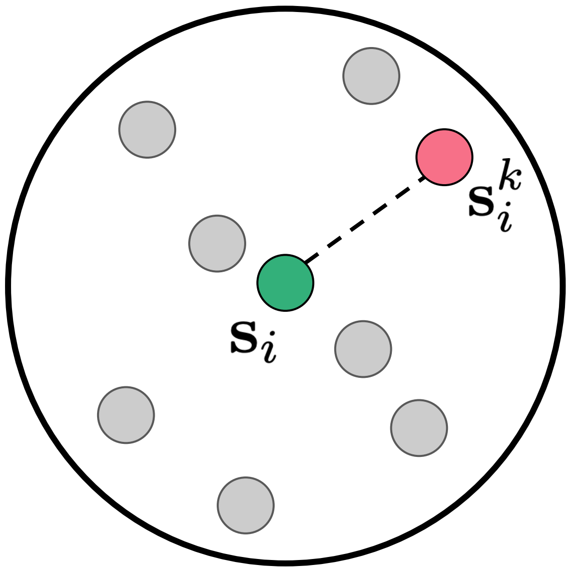


Analysis

- Particle-based entropy maximization is important for exploration
- Successor features and conditioning on latent variables helps quick adaptation



Summary



- Particle-based entropy maximization is effective in exploration
- Conditioning policy on latent variables helps adaptation
- Our method is simple yet effective in combining the best of both world

$$I(s;z) = \mathcal{H}(s) - \mathcal{H}(s|z)$$

Future Work

- Incorporate the prompt design to harvest the few-shot ability of our method
- Better representation learning for particle-based exploration
- Supervised training alternatives for unsupervised pretraining?