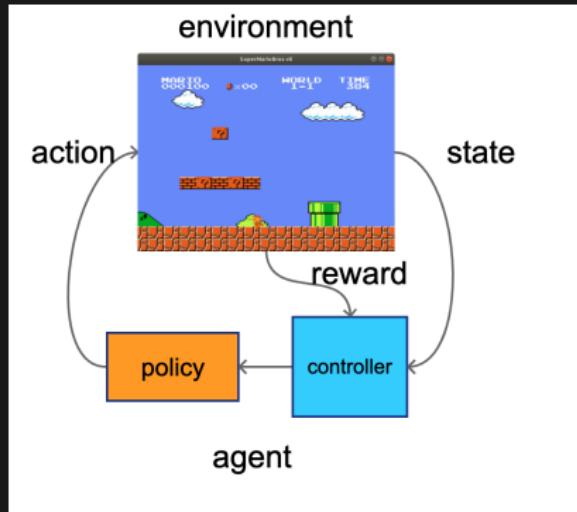
The background of the slide is a painting depicting a traditional Aztec ceremony. A priest in elaborate feathered headdress and ceremonial attire stands on a platform, performing a ritual over a large, crowded, and noisy scene. In the foreground, a person in a green feathered headdress looks up at the priest. Other figures, including a knight in armor, are visible in the shadows. The scene is set against a backdrop of ancient stone structures and pyramids.

reinforcement learning current problems

Michal CHOVANEC, PhD.

reinforcement learning



- ① **obtain state** - observation
- ② **choose action** - policy
- ③ **receive reward**
- ④ **learn from experiences**

when it works ?

- dense rewards
- small state space
- fully observable



when it works - AlphaGO, AlphaZero - DeepMind



Chess with Suren, AlphaZero's Most Astonishing "Zugzwang" Game



most famous algorithms

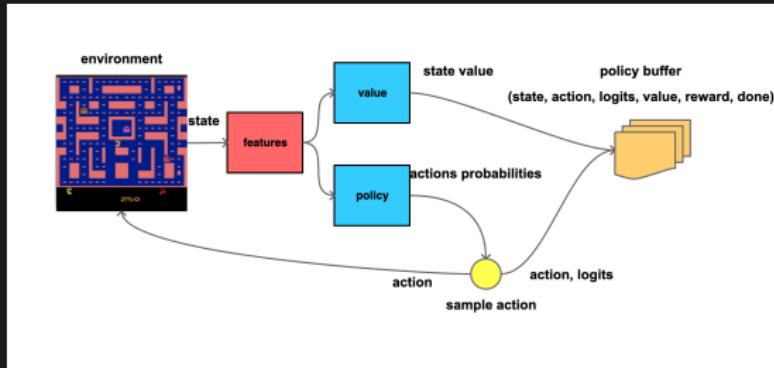
- deep Q network - **DQN**¹
- deep deterministic policy gradient - **DDPG**, D4PG²
- advantage actor critic - AC, **A2C**, A3C
- proximal policy optimization - **PPO**, TRPO³

¹Mnih et al. 2013

²Lillicrap, Hunt et al. 2016

³Schulman, et al. 2017

PPO - proximal policy optimisation



naive Actor-Critic vs PPO loss:

$$\mathcal{L}(\theta) = -\log \pi(a_n|s_n; \theta) \left(R(s_n, a_n) - V(s_n) \right) - \beta \mathcal{H}(\pi)$$

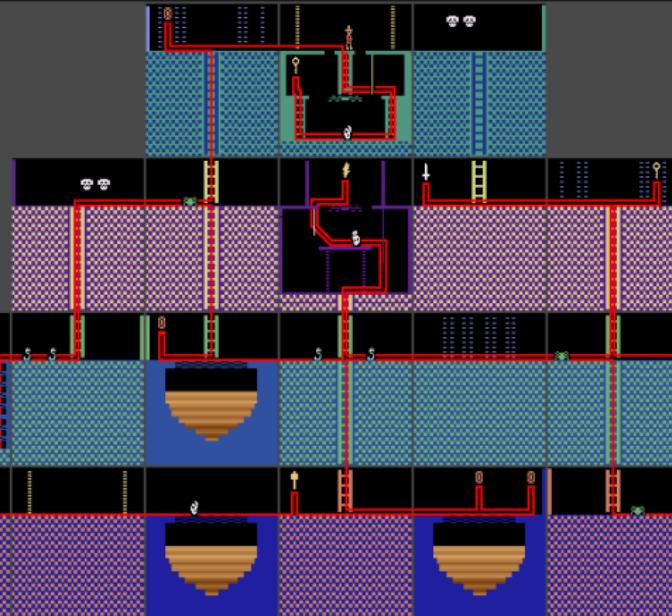
$$\mathcal{L}(\theta) = -\log \frac{\pi(a_n|s_n; \theta)}{\pi(a_n|s_n; \theta_{old})} \left(R(s_n, a_n) - V(s_n) \right) - \beta \mathcal{H}(\pi)$$

Montezuma's Revenge



© PARKER BROTHERS

ATARI® 2600™
Solution: Level 1



Graphical Game Solution

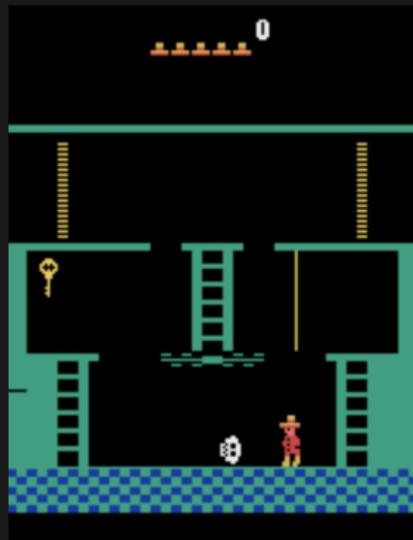
© 2001 Ben Valdes

MONTEZUMA'S REVENGE

© 1984 Parker Brothers

ATARI® 2600™ is a registered trademark of the Atari Corporation

Montezuma's Revenge



- **very sparse rewards** - hundreds of steps
- **huge state space**
- **hard exploration**
- **needs returns back**

state of the art score

source : <https://paperswithcode.com/sota/atari-games-on-atari-2600-montezumas-revenge>

year	name	score
2015	Deep Reinforcement Learning with Double Q-learning	0
2017	Curiosity-driven Exploration by Self-supervised Prediction ^a	0
2021	MuZero	2500
2018	Count-Based Exploration with Neural Density Models ^b	3705
2019	Exploration by Random Network Distillation ^c	8152
2021	GoExplore* ^d	43 000

* requires environment state saving/loading

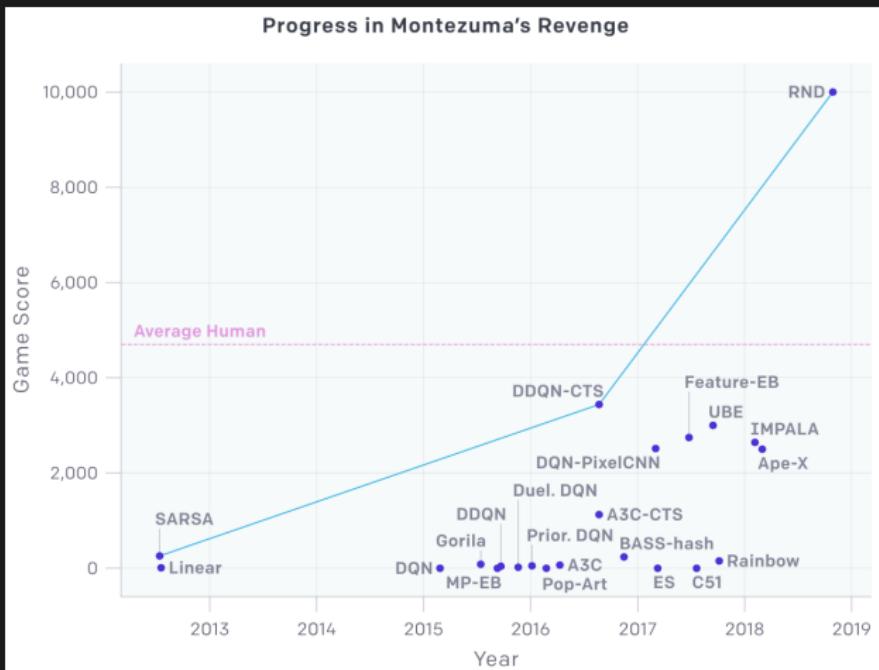
^a<https://arxiv.org/abs/1705.05363>

^b<https://arxiv.org/abs/1703.01310>

^c<https://arxiv.org/abs/1810.12894>

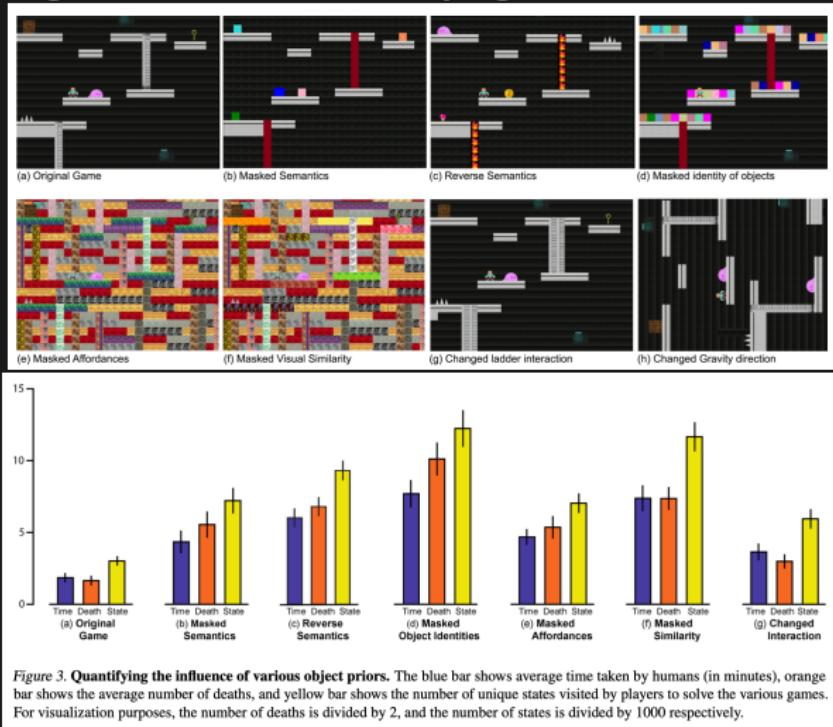
^d<https://arxiv.org/abs/2004.12919>

state of the art score



why so hard ?

Investigating Human Priors for Playing Video Games ⁴

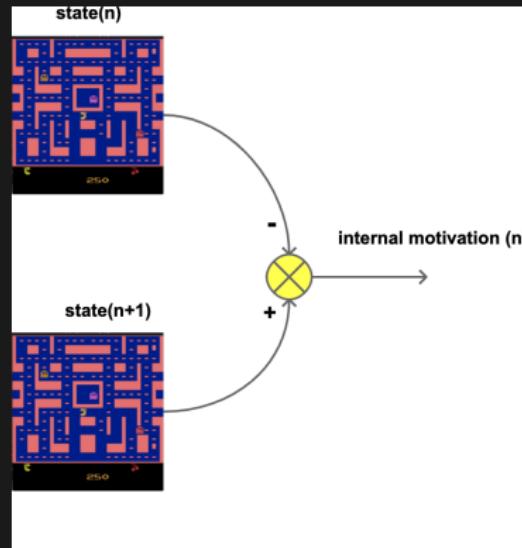


⁴Dubey et al. 2018

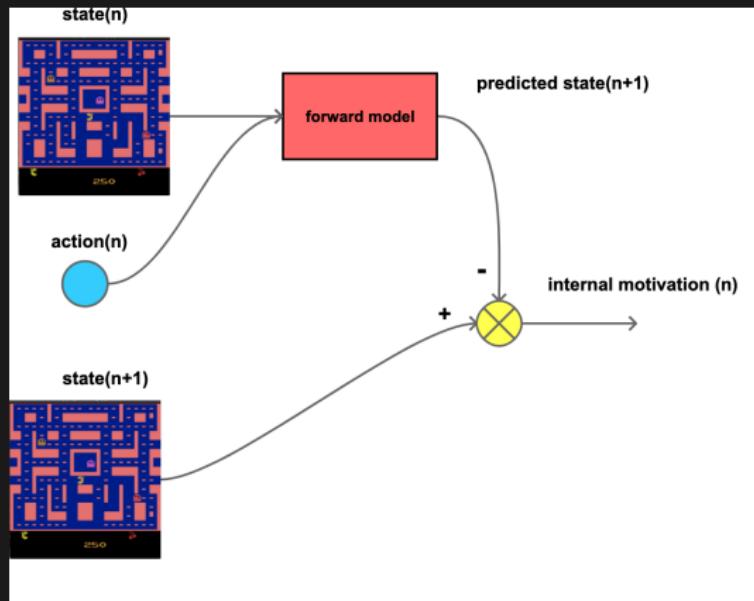
problems

- hard exploration tasks (sparse rewards)
- generalisation
- sample efficiency

pixel change motivation



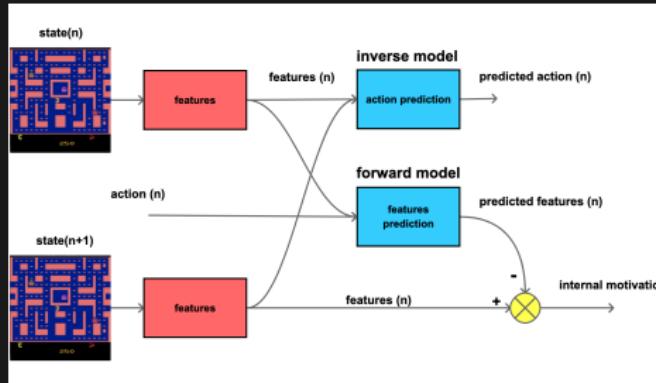
next state prediction



Curiosity-driven Exploration by Self-supervised Prediction ^a

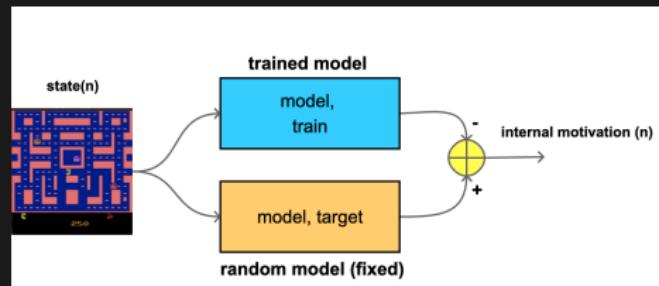
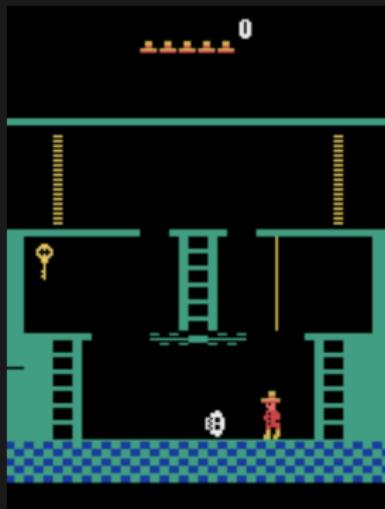
^aPathak et al. 2017

- predict **next state**, forward model
- **motivation == prediction error**
- not working ...

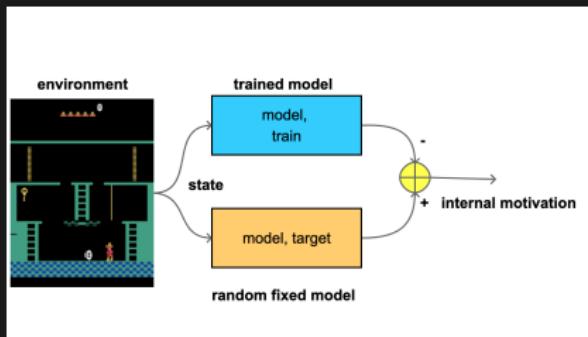


Exploration by Random Network Distillation ^a

^aBurda et al. 2018

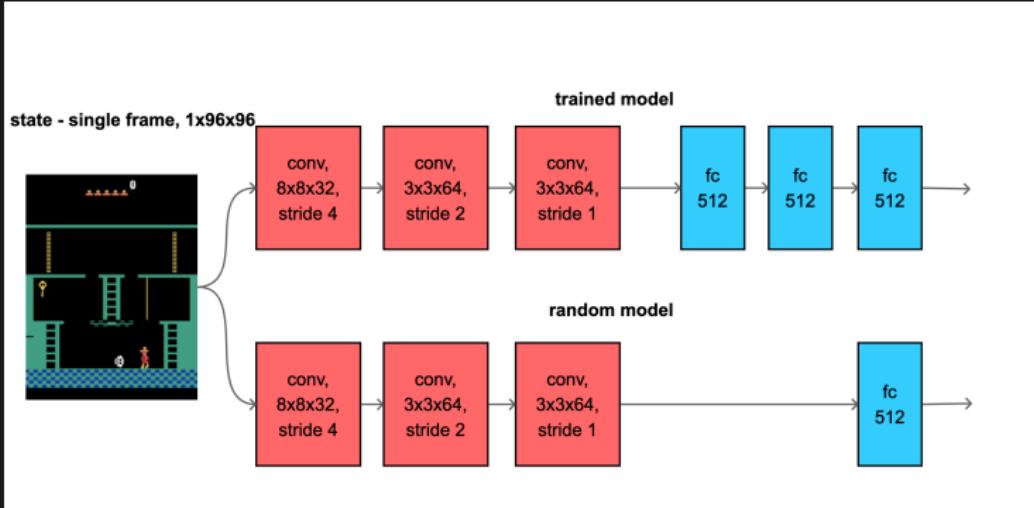


random network distillation

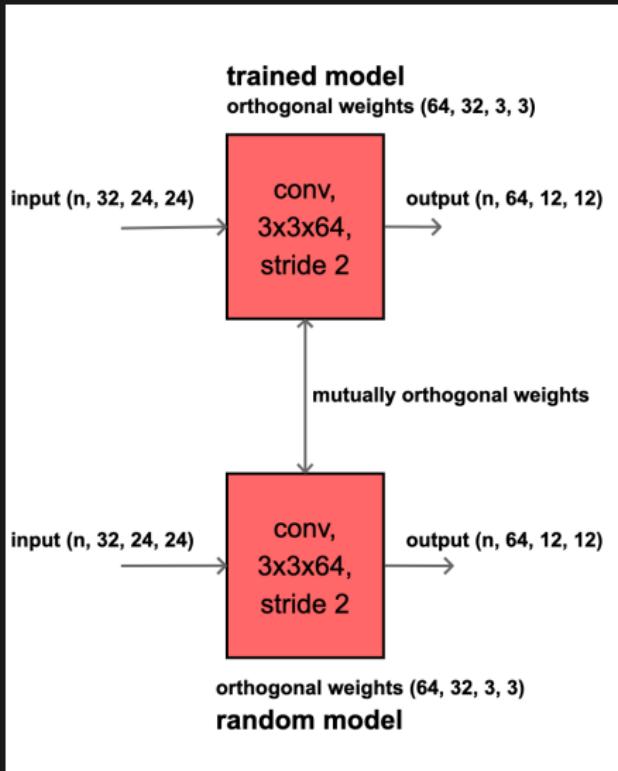


- neural network works as **novelty detector**
- model learns to imitate random (target) model
- **less visited states produce bigger motivation signal**
- orthogonal weights initialisation ($g = 2^{0.5}$) for strong signal
- lot of fully connected layers **to avoid generalisation**
- **coupled orthogonal models**

random network distillation architecture



coupled RND architecture

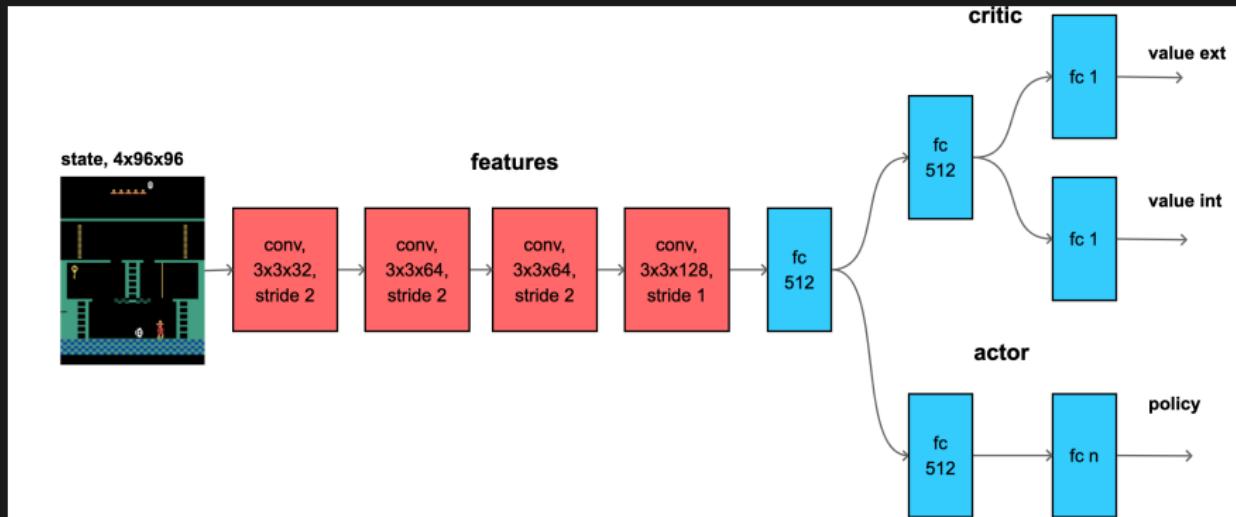


```
def coupled_orthoogonal_init(shape, gain):
    w = torch.zeros((2*shape[0], ) + shape[1:])
    torch.nn.init.orthogonal_(w, gain)

    w = w.reshape((2, ) + shape)
    return w[0], w[1]

wa, wb = coupled_orthoogonal_init((64, 32, 3, 3), 2.0**0.5)
```

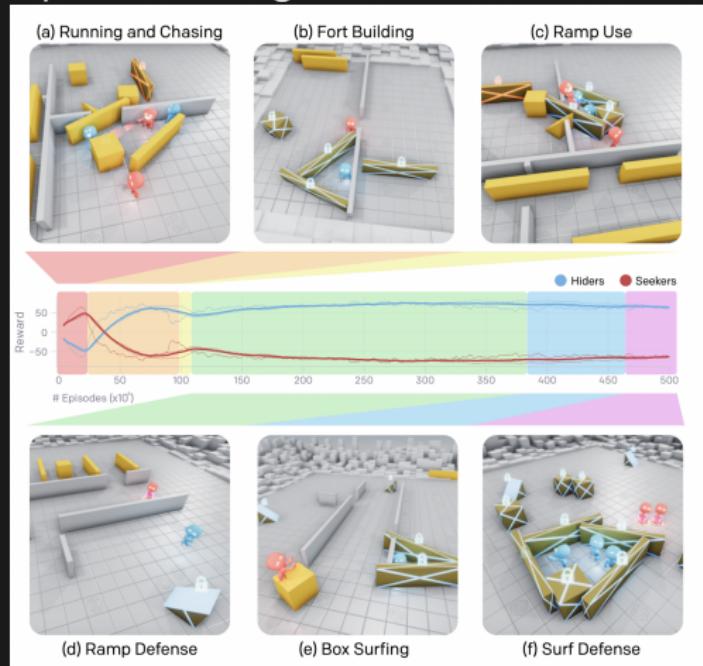
ppo model architecture



hide and seek ^a

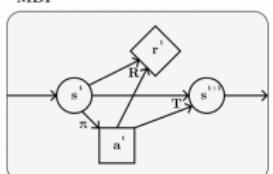
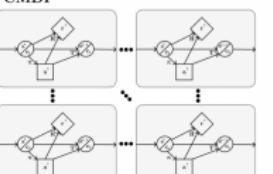
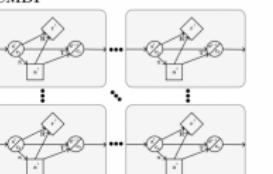
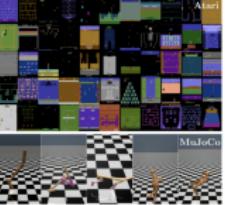
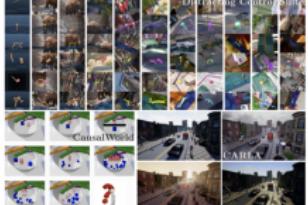
^aBaker et al. 2020

OpenAI - Emergent Tool Use from Multi-Agent Interaction



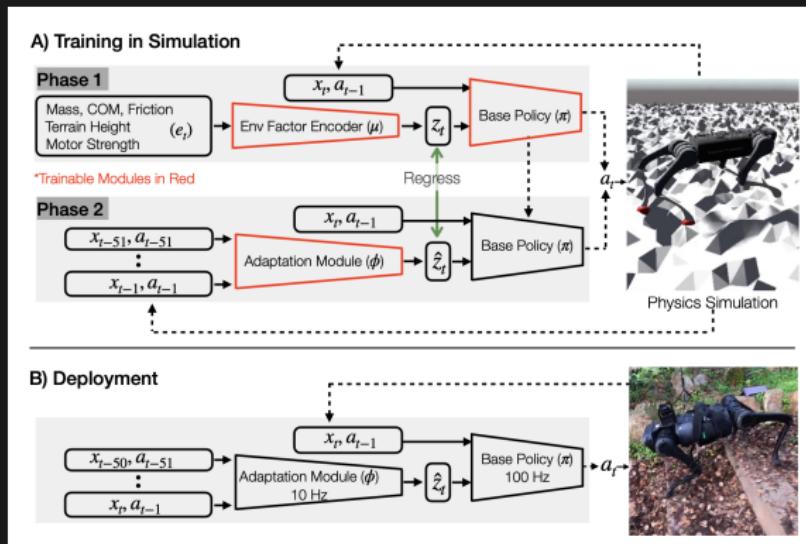
generalisation ^a

^aKirk et al. 2021

	Singleton Environments	IID Generalisation Environments	OOD Generalisation Environments
Graphical Models	 A diagram of a Markov Decision Process (MDP). It shows a state node s^i with an arrow to a reward node r^i . From r^i , an arrow points to a transition node T^i . From T^i , an arrow points to the next state node s^{i+1} . A policy node π has an arrow pointing to action node a^i . Action node a^i has arrows pointing to both s^{i+1} and r^i .	 A diagram of a Causal Markov Decision Process (CMDP). It shows two parallel causal paths from state s^i to s^{i+1} . The top path goes through a reward node r^i and a transition node T^i . The bottom path goes through an action node a^i . Both paths have policy nodes π preceding them.	 A diagram of a Causal Markov Decision Process (CMDP) under Out-of-Distribution (OOD) generalisation. It shows two parallel causal paths from state s^i to s^{i+1} . The top path goes through a reward node r^i and a transition node T^i . The bottom path goes through an action node a^i . Both paths have policy nodes π preceding them.
Train and Test Distribution	 A blue dot equals a red dot. Train = Test	 A scatter plot showing a uniform distribution of blue dots within a red rectangular boundary. $p_{\text{train}}(c) = p_{\text{test}}(c)$ Train Distribution = Test Distribution	 A scatter plot showing a uniform distribution of blue dots on the left and a red rectangular boundary on the right. $p_{\text{train}}(c) \neq p_{\text{test}}(c)$ Train Distribution \neq Test Distribution
Example Benchmarks	 Screenshots from the Alari and MuJoCo benchmarks.	 Screenshots from the OpenAI Gym and NoFluff Learning Environment benchmarks.	 Screenshots from the DivingBench, CasualWorld, and CARLA benchmarks.

RMA: Rapid Motor Adaptation for Legged Robots ^a

^aKumar et al. 2021



sample efficiency

- intuitive unit **one Montezuma experiment**
 - 128M samples runs **65hours**
 - eats less than **4G memory**
 - fits **3 experiments** into single GPU (12G)
-
- RND ⁵ $4.5 * 10^9$ samples, score 8152 on MR
 - Never give up ⁶ $3.5 * 10^{10}$ samples, score 10 000 on MR
 - SND $1.28 * 10^8$ score 10 000 on MR

NGU on my machine means **740 days !!!**

⁵Burda et al. 2018

⁶Badia et al. 2020

misleading papers - Curiosity-driven Exploration by Self-supervised Prediction ^a

^aPathak et al. 2017

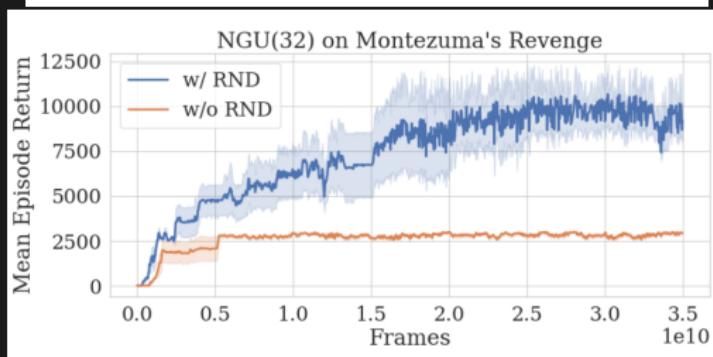
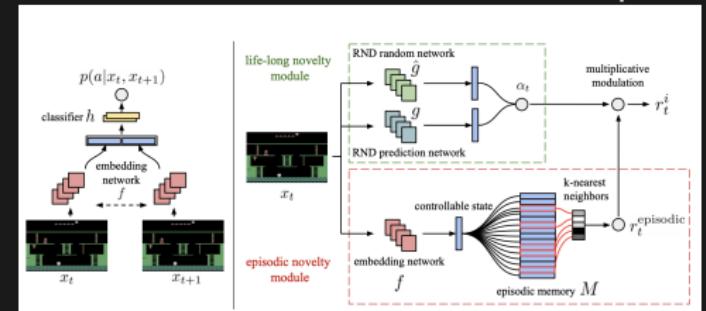
- not working on REAL hard exploration problems
- Super Mario is special case - moving forward is close to optimal policy
- inverse ICM model - why they didn't show accuracy (my results around 40% !!! even given policy)
- how predicted state looks ?

	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	3,906	8,152	-3	8,666	3,282	1,859
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209 ¹	3,700 ²	0	15,806²	12,380¹	1,813³
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

misleading papers - Never give up ^a

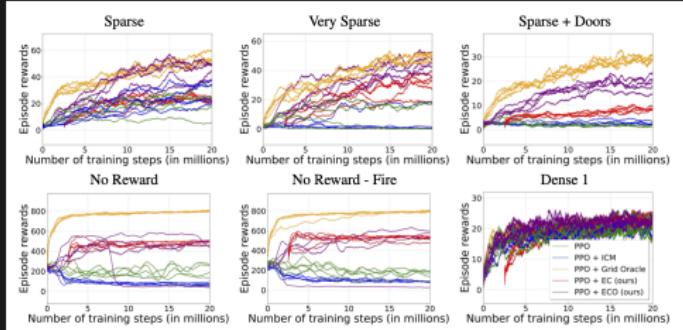
^aBadia et al. 2020

nice looking score, but on cost of $3.5 * 10^{10}$ samples !!!



other misleadings

avoiding comparing with SOTA or common benchmarks
results : Episodic Curiosity Through Reachability, Savinov, 2019



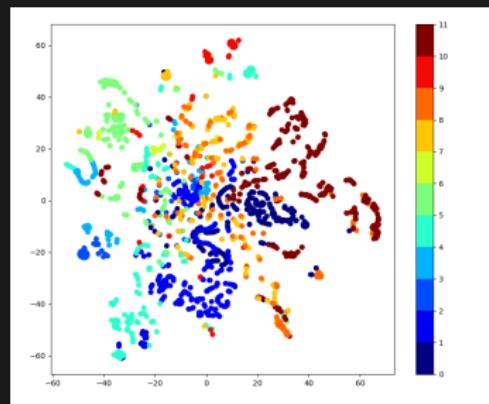
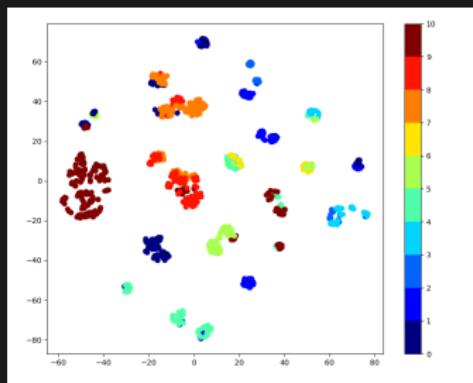
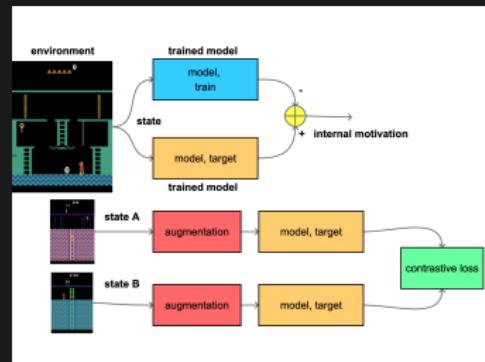
many other :

- simple gridworld or toy environment experiments
- providing key prior information (e.g. position)
- selecting only "good" results

my current research

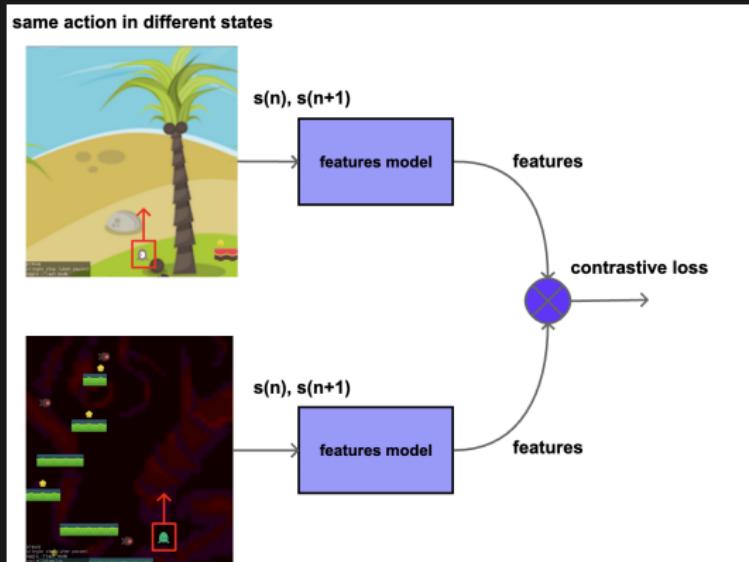
- siamese network distillation - reached SOTA score with 1/100 samples
- symmetry driven generalisation, Noether's theorem in RL

siamese network distillation



symmetry driven generalisation ^a

^aBronstein et al. 2021 Geometric deep learning - some trivial hand-crafted symmetries



$$\forall(s_n, s_{n+1}|a) : f(s_n, s_{n+1}; \theta) = const_a^7$$

⁷avoid trivial collapse

recommended sources

- book : Maxim Lapan, 2020, Deep Reinforcement Learning Hands-On second edition
- book : Enes Bilgin, 2020, Mastering Reinforcement Learning with Python
- youtuber : Yannic Kilcher, link
- youtuber : Two Minute Papers, link
- web : Paper With Code, link
- web : Intellabs, link

Q&A



- <https://github.com/michalnand/>
- michal.nand@gmail.com