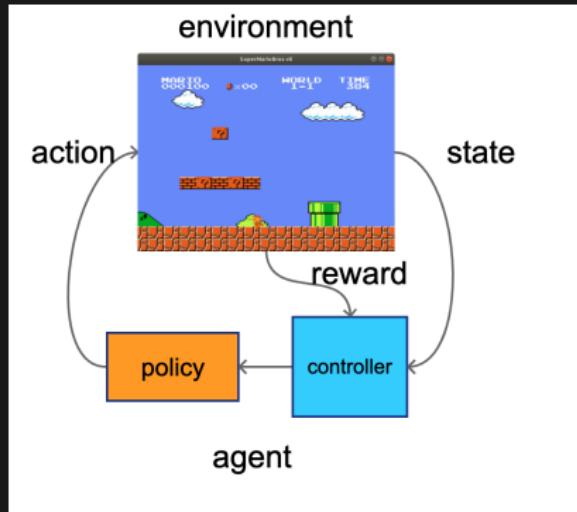
A painting depicting a traditional Aztec ceremony. In the center, a priest in elaborate feathered headdress and ceremonial attire stands on a platform, his arms raised in a gesture of offering or sacrifice. He holds a small object in his right hand. Below him, a vast crowd of people, mostly men in traditional tunics, looks up in awe. To the right, another figure in a detailed headdress and armor stands near a red banner with gold symbols. The background features a large, ornate pyramid with multiple levels and steps. The overall atmosphere is one of a solemn, historical event.

# 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**<sup>1</sup>
- deep deterministic policy gradient - **DDPG**, D4PG<sup>2</sup>
- advantage actor critic - AC, **A2C**, A3C
- proximal policy optimization - **PPO**, TRPO<sup>3</sup>

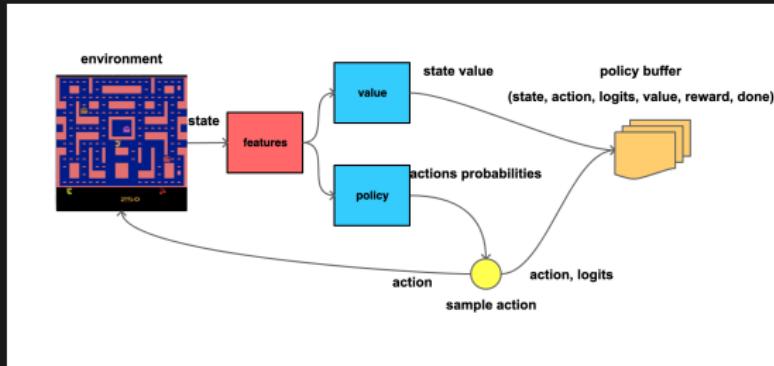
---

<sup>1</sup>Mnih et al. 2013

<sup>2</sup>Lillicrap, Hunt et al. 2016

<sup>3</sup>Schulman, et al. 2017

# PPO - proximal policy optimisation



naive Actor-Critic vs PPO loss:

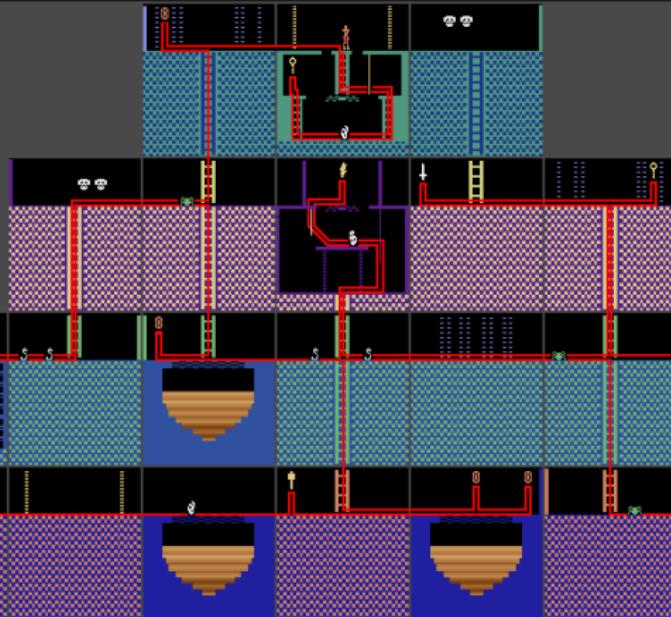
$$\mathcal{L}(\theta) = -\log \pi(a_n|s_n; \theta) \left( R(s_n, a_n) - V(s_n) \right) - \beta \mathcal{H}(\pi)$$

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

# Montezuma's Revenge



ATARI<sup>®</sup> 2600™  
Solution: Level 1

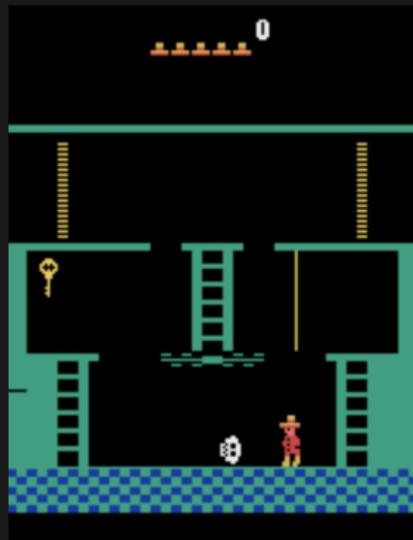


Graphical Game Solution  
© 2001 Ben Valdes

MONTEZUMA'S REVENGE  
© 1984 Parker Brothers

ATARI<sup>®</sup> 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

year	name	score
2015	Deep Reinforcement Learning with Double Q-learning	0
2017	Curiosity-driven Exploration <sup>a</sup>	0
2021	MuZero	2500
2018	Count-Based Exploration with Neural Density Models <sup>b</sup>	3705
2019	Exploration by Random Network Distillation <sup>c</sup>	8152
2021	GoExplore* <sup>d</sup>	43 000
2022	Byol Explore <sup>e</sup>	13 518
<b>2022/3</b>	<b>Self Supervised Exploration</b>	<b>25 000</b>

\* requires environment state saving/loading

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

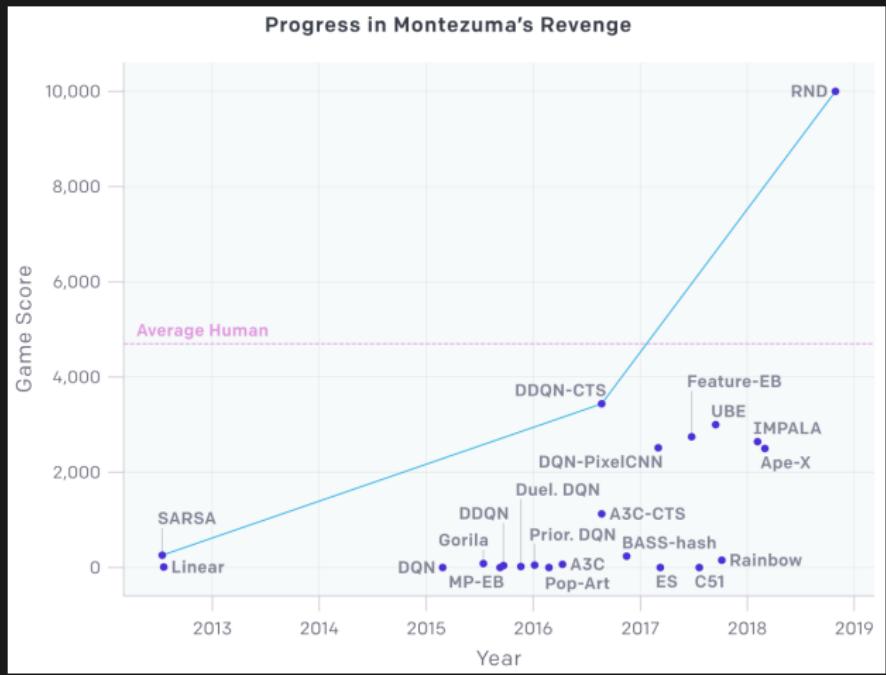
<sup>b</sup><https://arxiv.org/abs/1703.01310>

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

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

<sup>e</sup><https://arxiv.org/abs/2206.08332>

# state of the art score

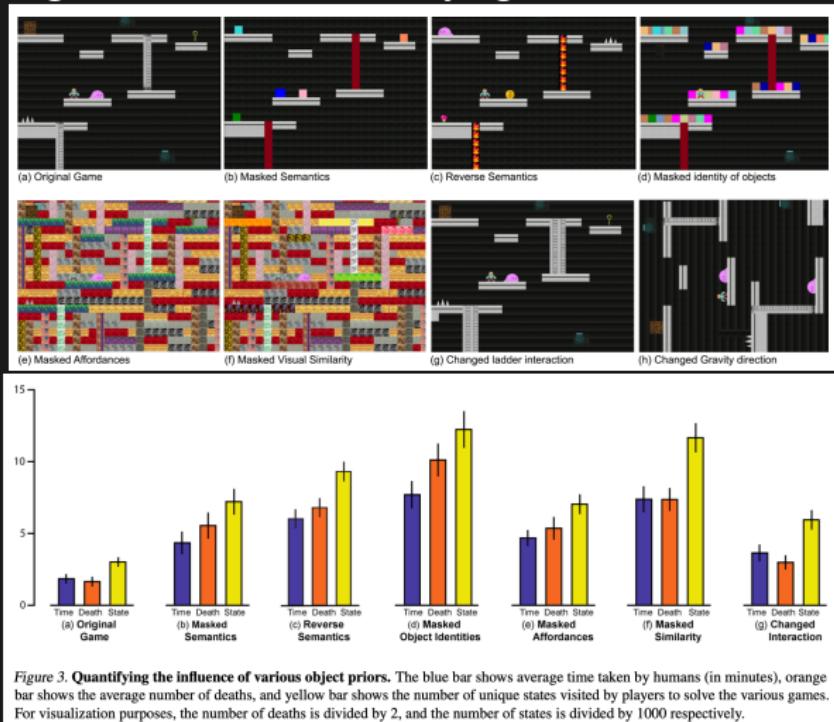


source :

[https://paperswithcode.com/sota/  
atari-games-on-atari-2600-montezumas-revenge](https://paperswithcode.com/sota/atari-games-on-atari-2600-montezumas-revenge)

# why so hard ?

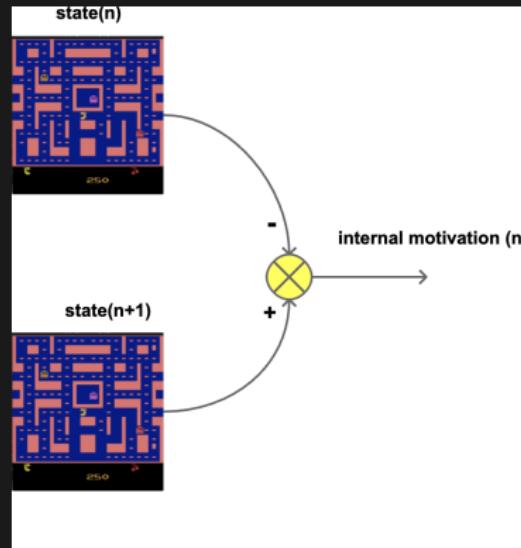
## Investigating Human Priors for Playing Video Games <sup>4</sup>



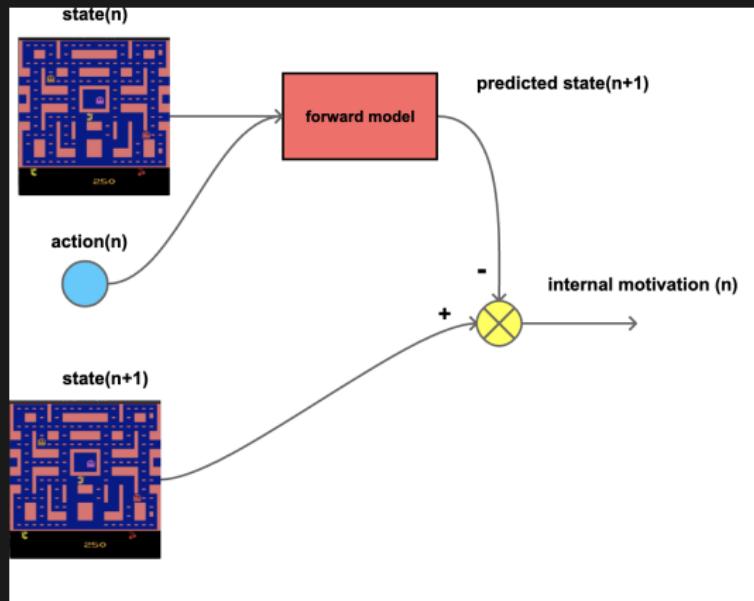
# problems

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

# pixel change motivation



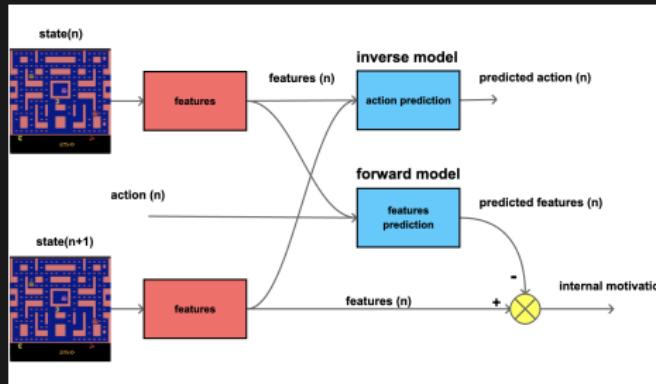
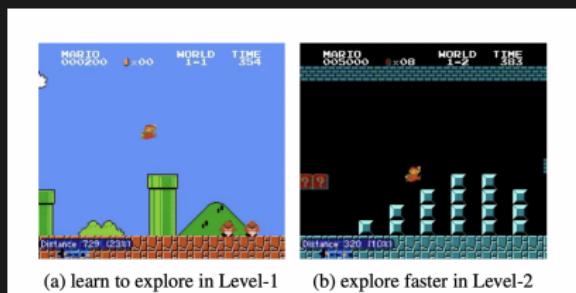
# next state prediction



# Curiosity-driven Exploration by Self-supervised Prediction <sup>a</sup>

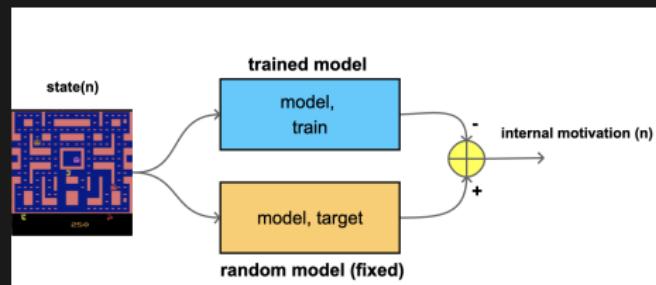
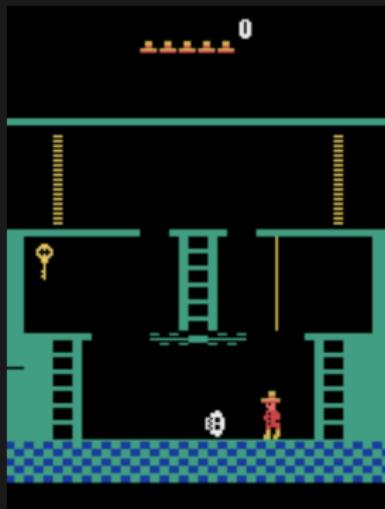
<sup>a</sup>Pathak et al. 2017

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

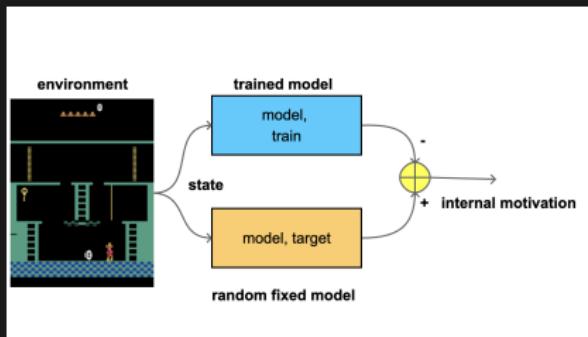


# Exploration by Random Network Distillation <sup>a</sup>

<sup>a</sup>Burda et al. 2018

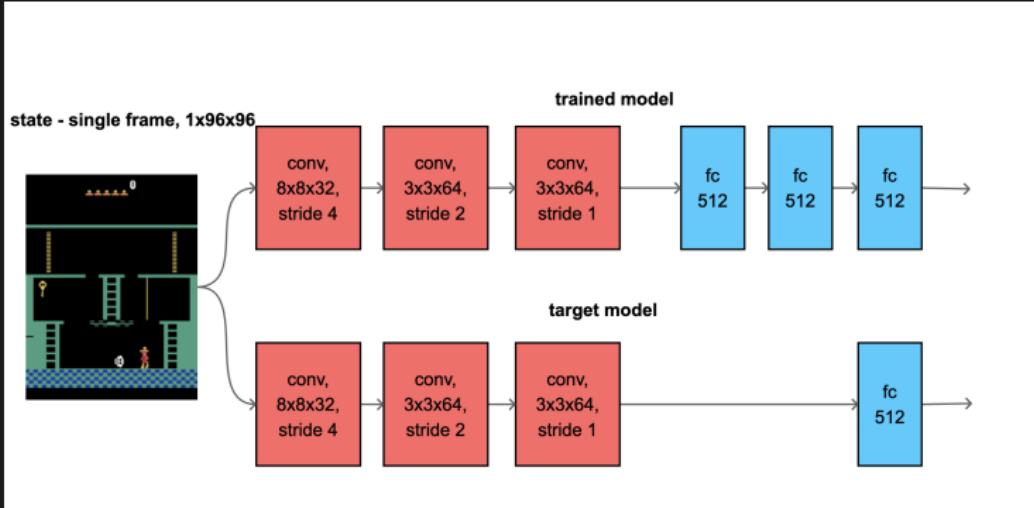


# random network distillation

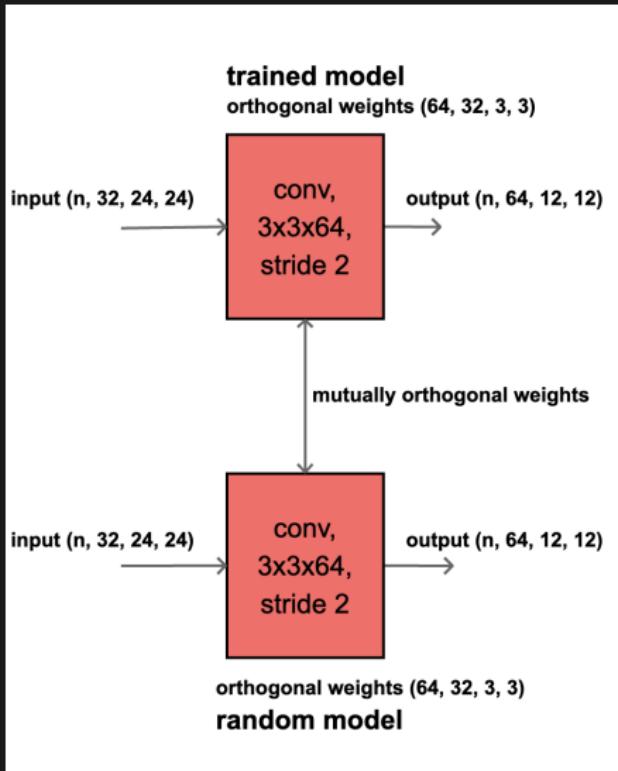


- neural network works as **novelty detector**
- model learns to imitate random (target) model
- **less visited states produce bigger motivation signal**
- orthogonal weights initialisation ( $g = 2^{0.5}$ ) for strong signal
- lot of fully connected layers **to avoid generalisation**
- **coupled orthogonal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

# random network distillation architecture



# coupled RND architecture

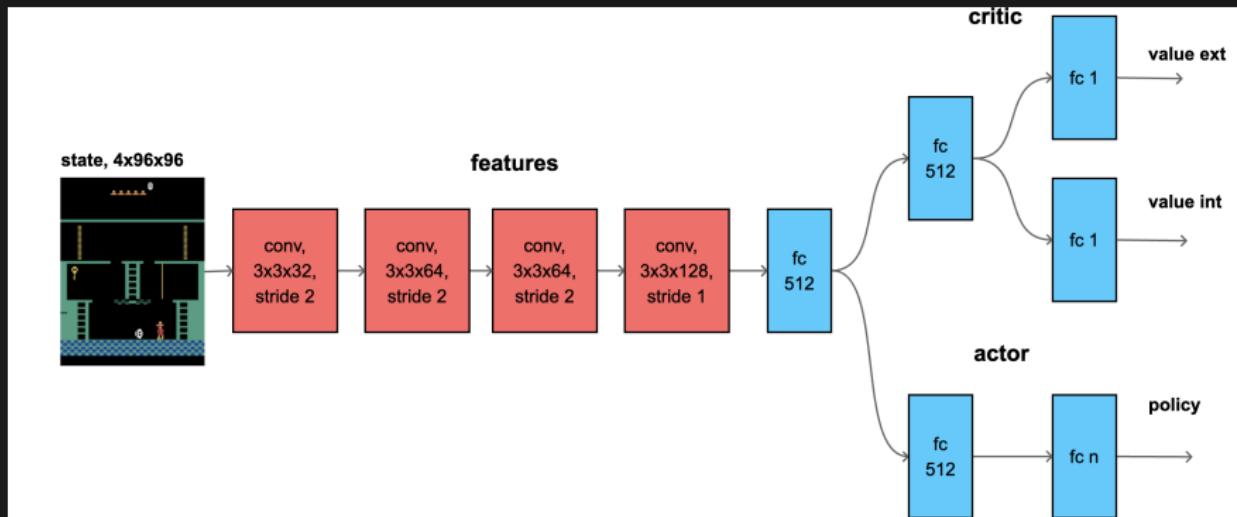


```
def coupled_orthoogonal_init(shape, gain):
    w = torch.zeros((2*shape[0], ) + shape[1:])
    torch.nn.init.orthogonal_(w, gain)

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

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

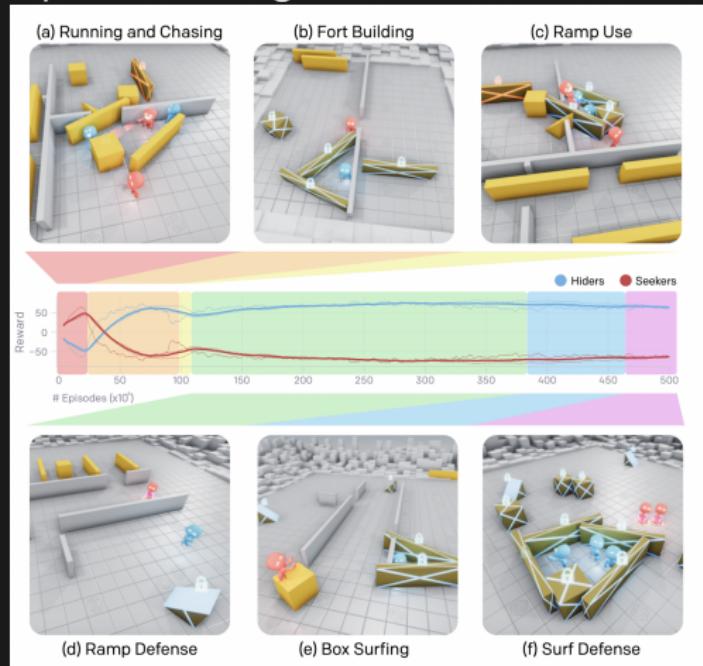
# ppo model architecture



# hide and seek <sup>a</sup>

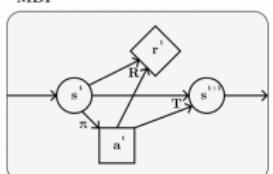
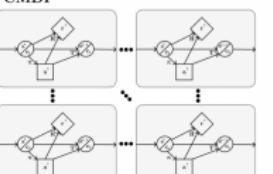
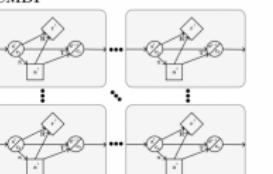
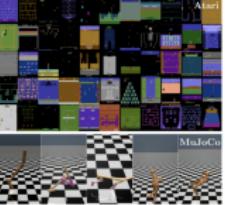
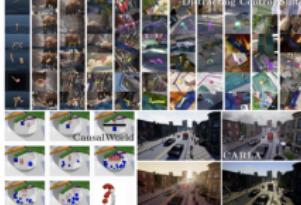
<sup>a</sup>Baker et al. 2020

## OpenAI - Emergent Tool Use from Multi-Agent Interaction



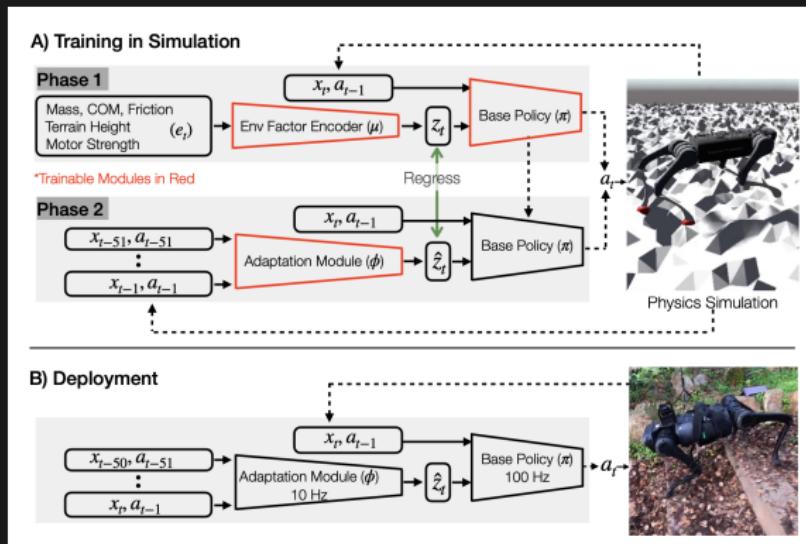
# generalisation <sup>a</sup>

<sup>a</sup>Kirk 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 $\pi$ 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 $\pi$ 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 $\pi$ preceding them.
Train and Test Distribution	 A blue dot equals a red dot. $p_{\text{train}}(c) = p_{\text{test}}(c)$ 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 blue dots on the left and a large red rectangle on the right, indicating a mismatch between the train and test distributions. $p_{\text{train}}(c) \neq p_{\text{test}}(c)$ Train Distribution $\neq$ Test Distribution
Example Benchmarks	 A grid of small images showing various environments: Alasri, OpenAI Progeny, Nothack Learning Environment, and MuJoCo.	 A grid of small images showing various environments: OpenAI Progeny, Nothack Learning Environment, and MuJoCo.	 A grid of small images showing various environments: CausalWorld and CARLA.

# RMA: Rapid Motor Adaptation for Legged Robots <sup>a</sup>

<sup>a</sup>Kumar 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 <sup>5</sup>  $4.5 * 10^9$  samples, score 8152 on MR
  - Never give up <sup>6</sup>  $3.5 * 10^{10}$  samples, score 10 000 on MR
  - CND  $1.28 * 10^8$  score 25 000 on MR

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

---

<sup>5</sup>Burda et al. 2018

<sup>6</sup>Badia et al. 2020

# misleading papers - Curiosity-driven Exploration by Self-supervised Prediction <sup>a</sup>

<sup>a</sup>Pathak et al. 2017

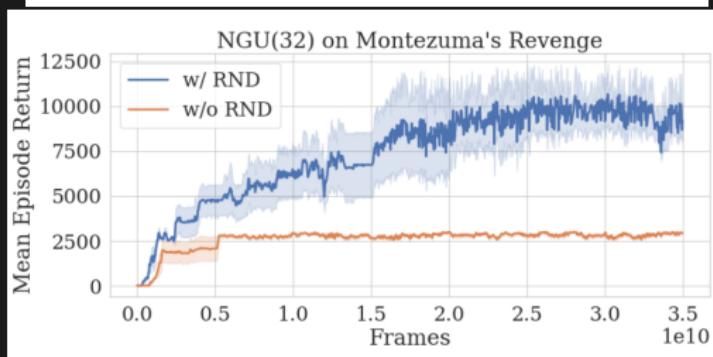
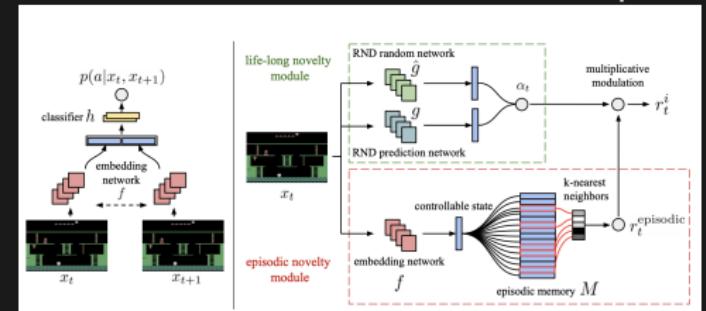
- not working on REAL hard exploration problems
- Super Mario is special case - moving forward is close to optimal policy
- inverse ICM model - why they didn't show accuracy (my results around 40% !!! even given policy)
- how predicted state looks ?

	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	<b>3,906</b>	<b>8,152</b>	-3	8,666	3,282	<b>1,859</b>
PPO	3,426	2,497	0	105	3,387	0
<b>Dynamics</b>	<b>3,371</b>	<b>400</b>	<b>0</b>	<b>33</b>	<b>3,246</b>	<b>1,712</b>
SOTA	2,209 <sup>1</sup>	3,700 <sup>2</sup>	<b>0</b>	<b>15,806<sup>2</sup></b>	<b>12,380<sup>1</sup></b>	<b>1,813<sup>3</sup></b>
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

# misleading papers - Never give up <sup>a</sup>

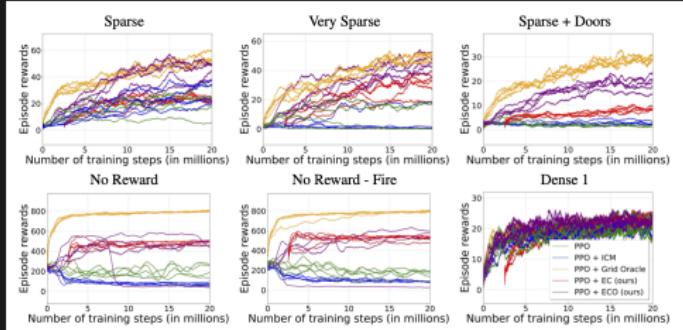
<sup>a</sup>Badia et al. 2020

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



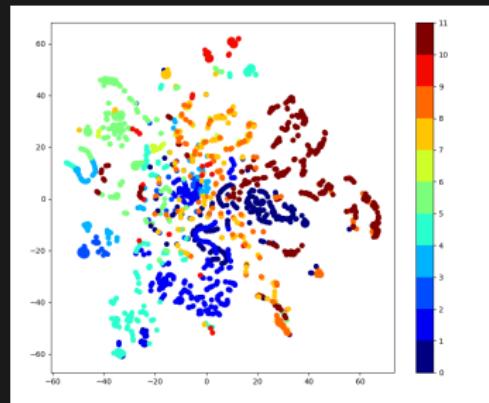
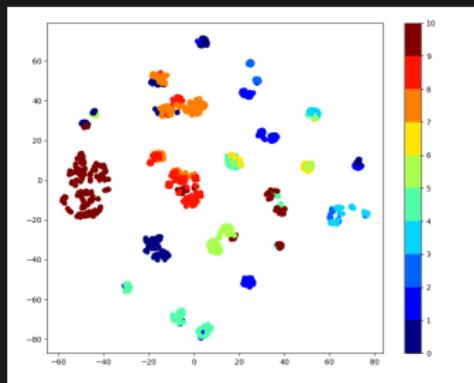
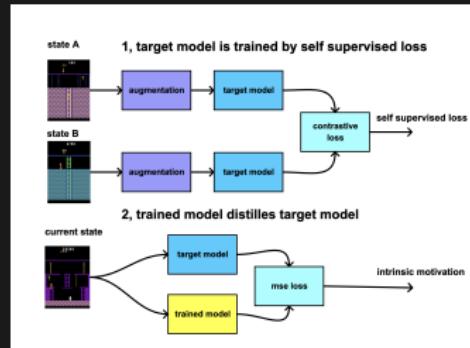
many other :

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

# my current research

- self supervised network distillation - reached SOTA score with 1/100 samples
- raising entropy driven exploration
- symmetry driven generalisation, Noether's theorem in RL

# exploration by self supervised network distillation



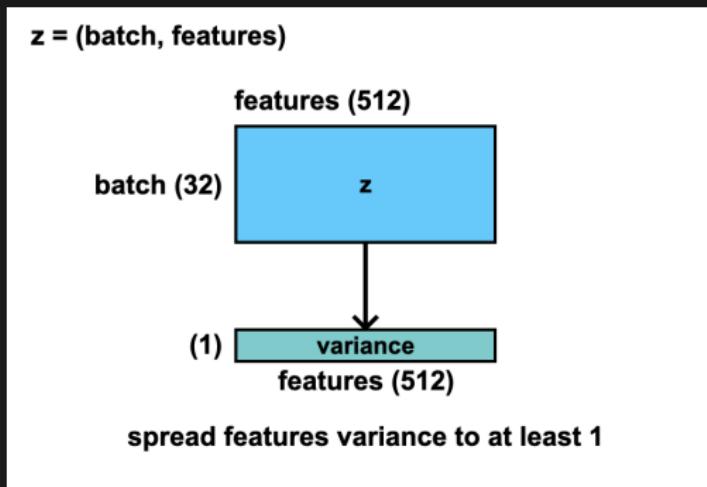
# vicreg self supervised loss

- variance : maximize batch-wise variance
- invariance : closer features for similar, distant for different states
- covariance : minimize features covariance

$$\begin{aligned}\mathcal{L} = & \alpha \max \left( 1 - \sum_f^F \text{var}_0(z_a) + \text{var}_0(z_b), 0 \right) && \text{variance} \\ & + \beta \sum_n^N \begin{cases} \sum_f^F (z_a - z_b)^2, & \text{if similar} \\ \max(1 - \sum_f^F (z_a - z_b)^2, 0), & \text{otherwise} \end{cases} && \text{invariance} \\ & + \sum_f^F (1 - \mathbb{I}) \left( (z_a^T z_a)^2 + (z_b^T z_b)^2 \right) && \text{covariance}\end{aligned}$$

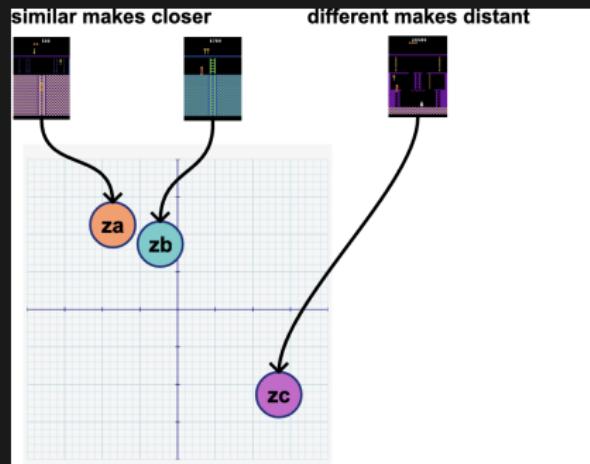
# variance

$$\mathcal{L}_{variance} = \alpha \max \left( 1 - \sum_f^F var_0(z_a) + var_0(z_b), 0 \right)$$



# invariance

$$\mathcal{L}_{invariance} = \beta \sum_n^N \begin{cases} \sum_f^F (z_a - z_b)^2, & \text{if similar} \\ \max(1 - \sum_f^F (z_a - z_b)^2, 0), & \text{otherwise} \end{cases}$$



# covariance

$$\mathcal{L}_{\text{covariance}} = \gamma \sum_f^F (1 - \mathbb{I}) \left( (z_a^T z_a)^2 + (z_b^T z_b)^2 \right)$$

$z = (\text{batch, features})$

batch (32)

features (512)

$z^T$

\*

batch (32)

features (512)

$z$

$\parallel$   
covariance (512)  
covariance (512)

covariance (512)

c

minimize all correlated values, except diagonal

$L =$

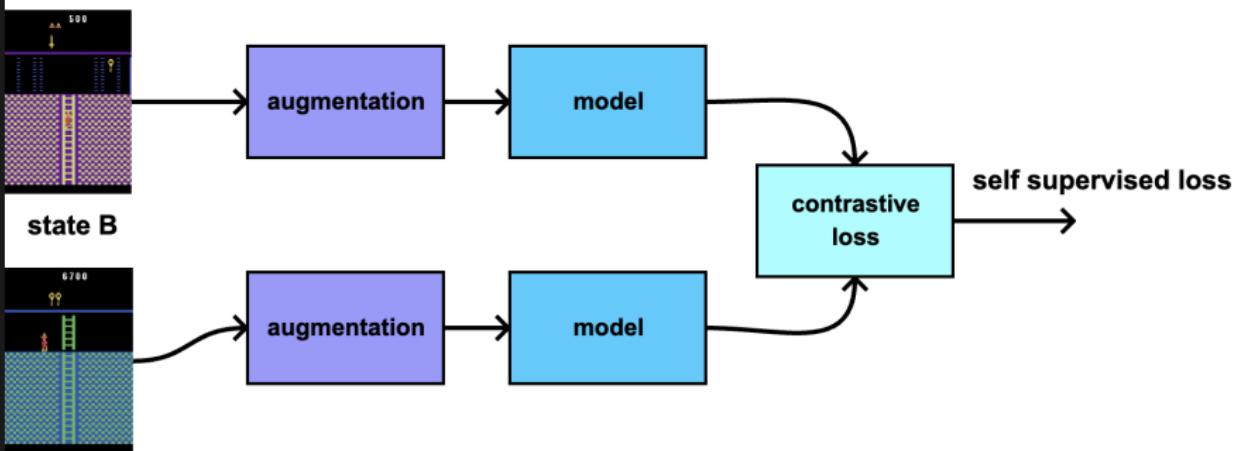
c

off diag

# raising entropy driven exploration

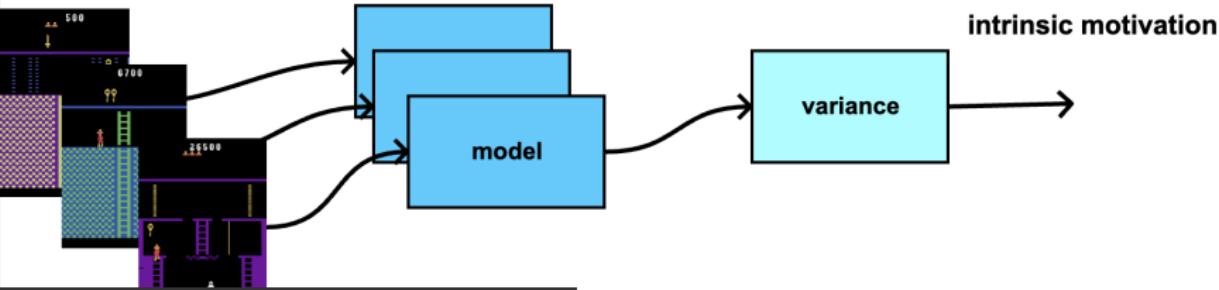
state A

## 1, model is trained by self supervised loss



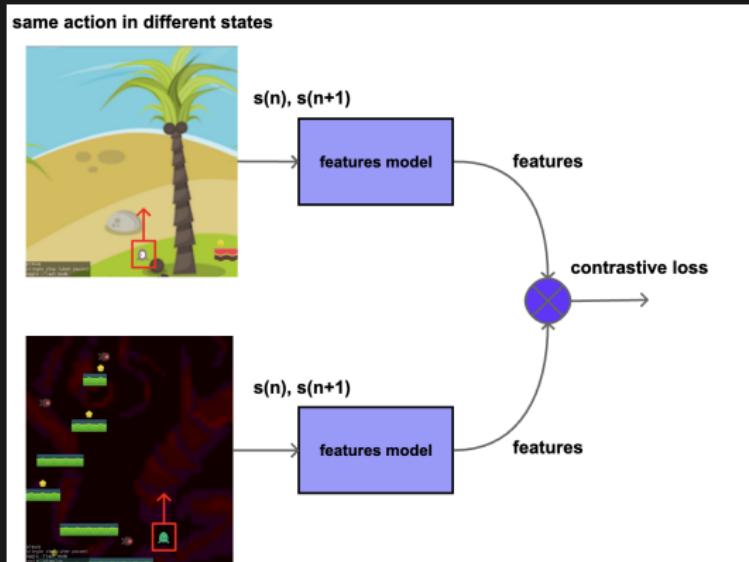
## 2, compute features from episode buffer

episode states buffer



# symmetry driven generalisation <sup>a</sup>

<sup>a</sup>Bronstein et al. 2021 Geometric deep learning - some trivial hand-crafted symmetries



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

<sup>7</sup>avoid 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