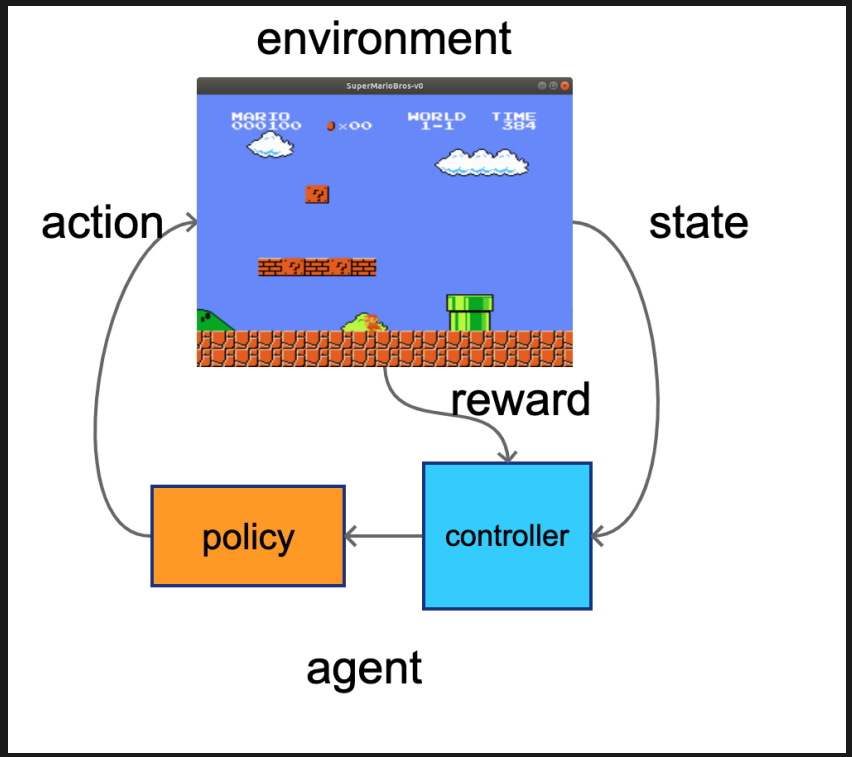
reinforcement learning current problems

Michal CHOVANEC, PhD.

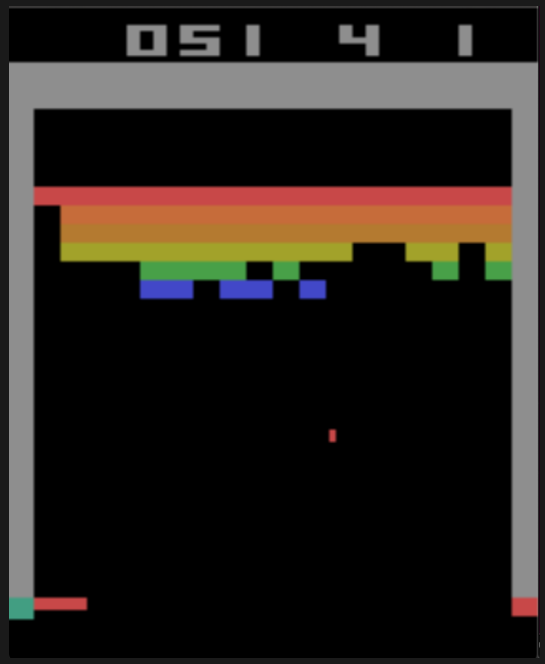
reinforcement learning



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

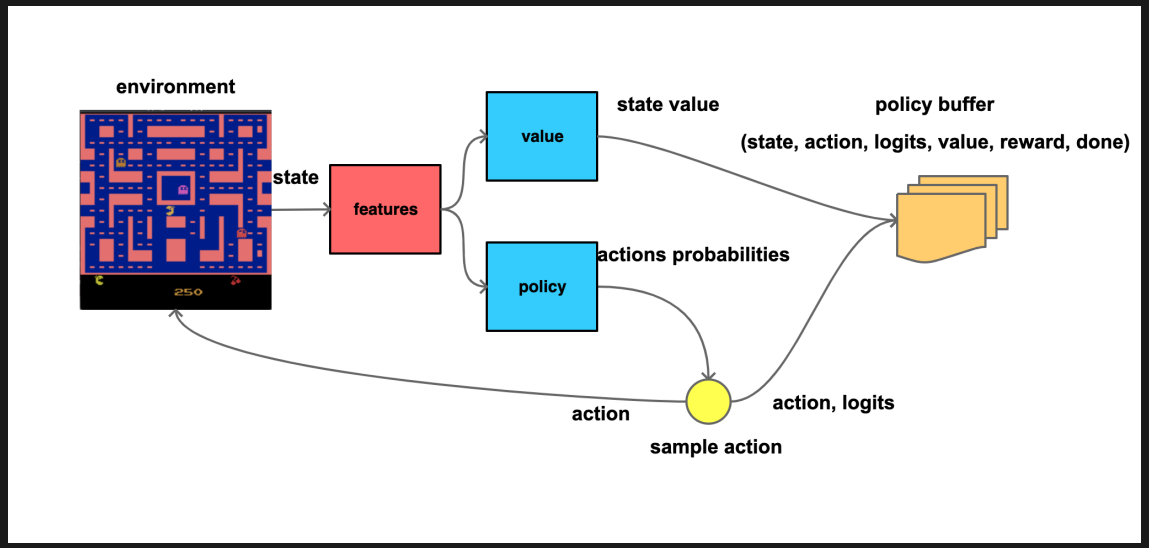
Chess with Suren, AlphaZero's Most Astonishing "Zugzwang" Game

most famous algorithms

- deep Q network - **DQN** [1]
- deep deterministic policy gradient - **DDPG**, D4PG [2]
- advantage actor critic - AC, **A2C**, A3C
- proximal policy optimization - **PPO**, TRPO [3]

[1] Mnih et al. 2013
[2] Lillicrap, Hunt et al. 2016
[3] Schulman, et al. 2017

PPO - proximal policy optimisation

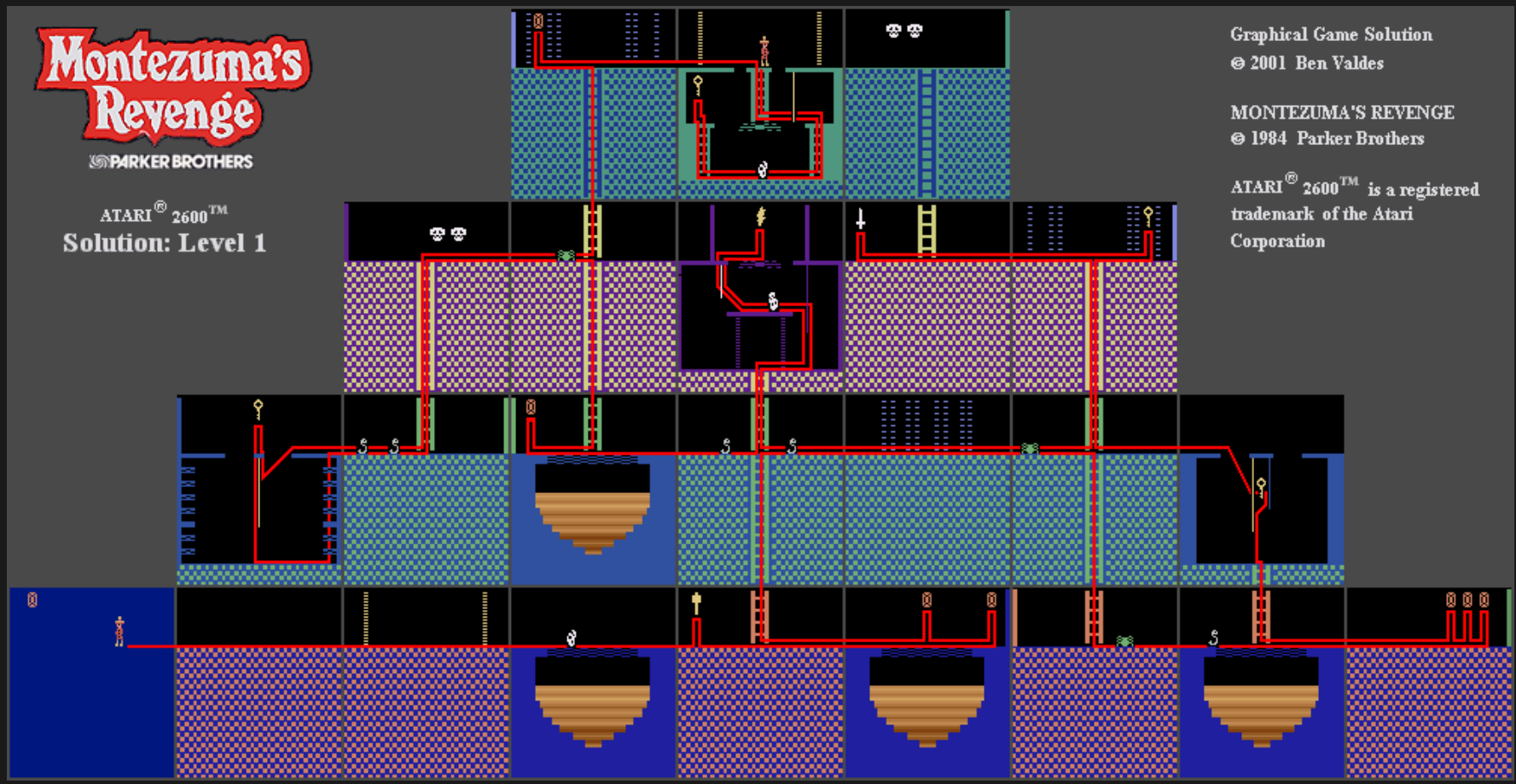


naive Actor-Critic vs PPO loss:

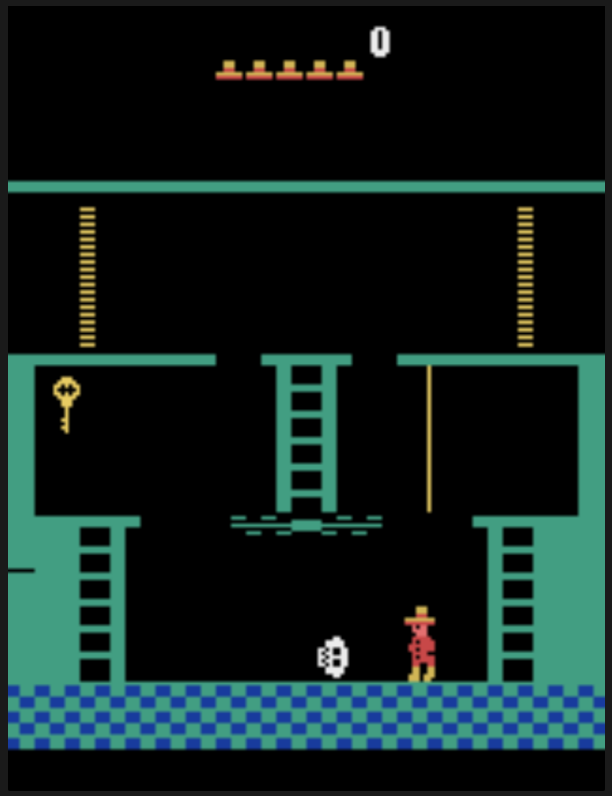
$$\mathcal{L}(\theta) = -log\pi(a_n|s_n;\theta)\Big(R(s_n,a_n) - V(s_n)\Big) - \beta\mathcal{H}(\pi)$$

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

Montezuma's Revenge



Montezuma's Revenge



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

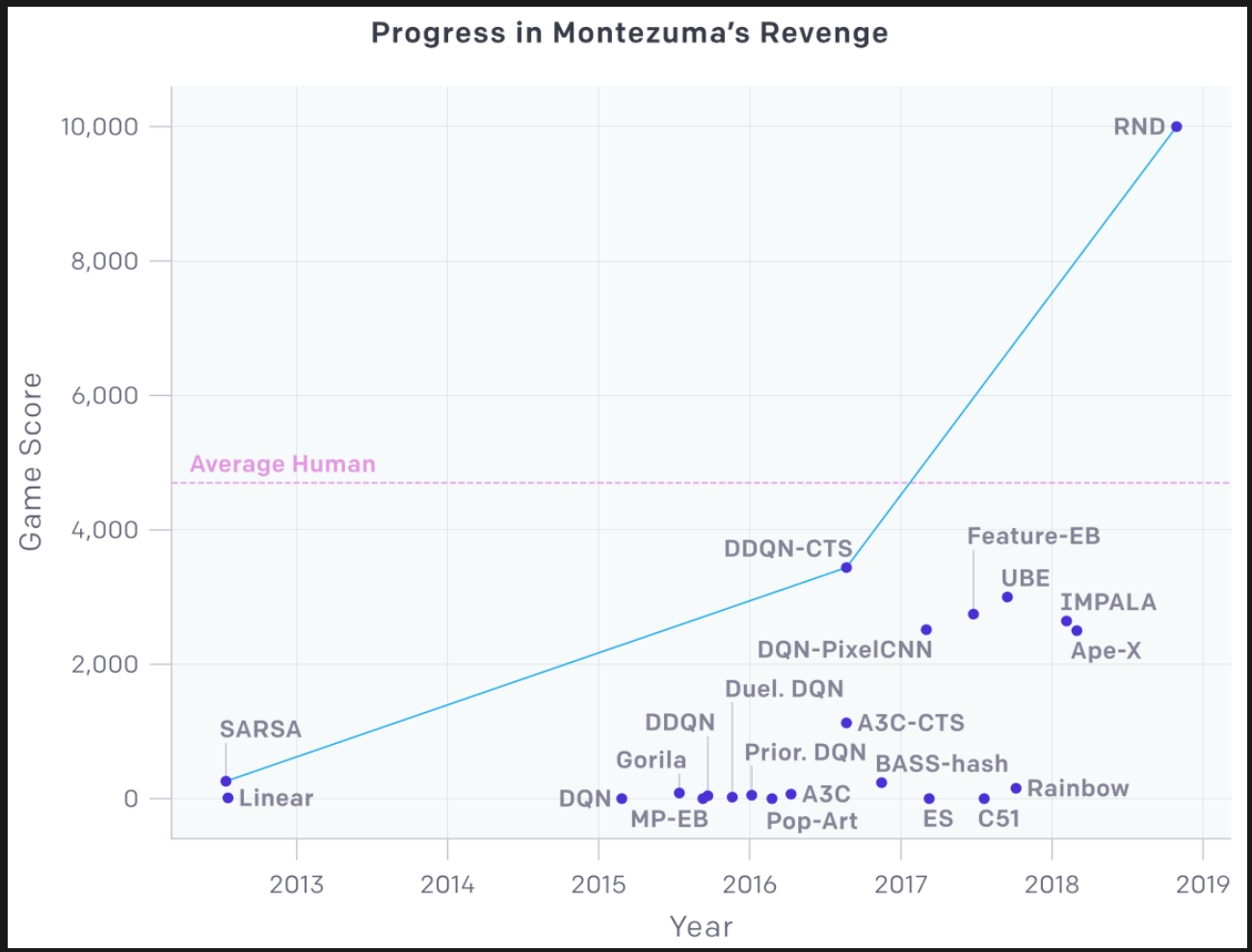
state of the art score

year	name	score
2015	Deep Reinforcement Learning with Double Q-learning	0
2017	Curiosity-driven Exploration [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
2022	Byol Explore [e]	13 518
2022/3	**Self Supervised Exploration**	**25 000**

*** 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
[e] https://arxiv.org/abs/2206.08332

state of the art score

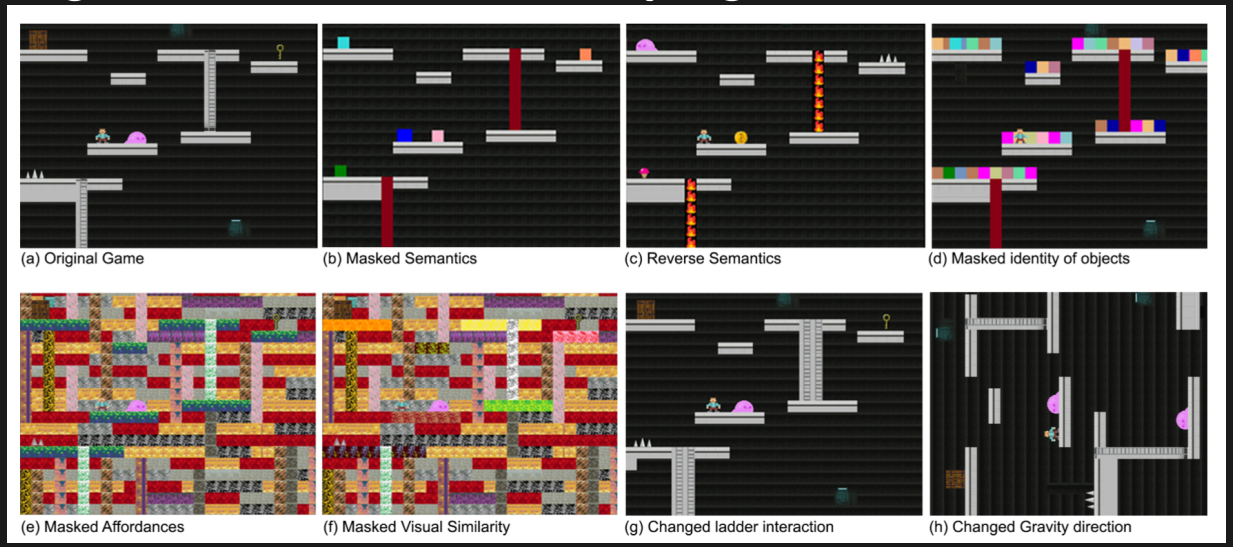


source :

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

why so hard ?

Investigating Human Priors for Playing Video Games [4]



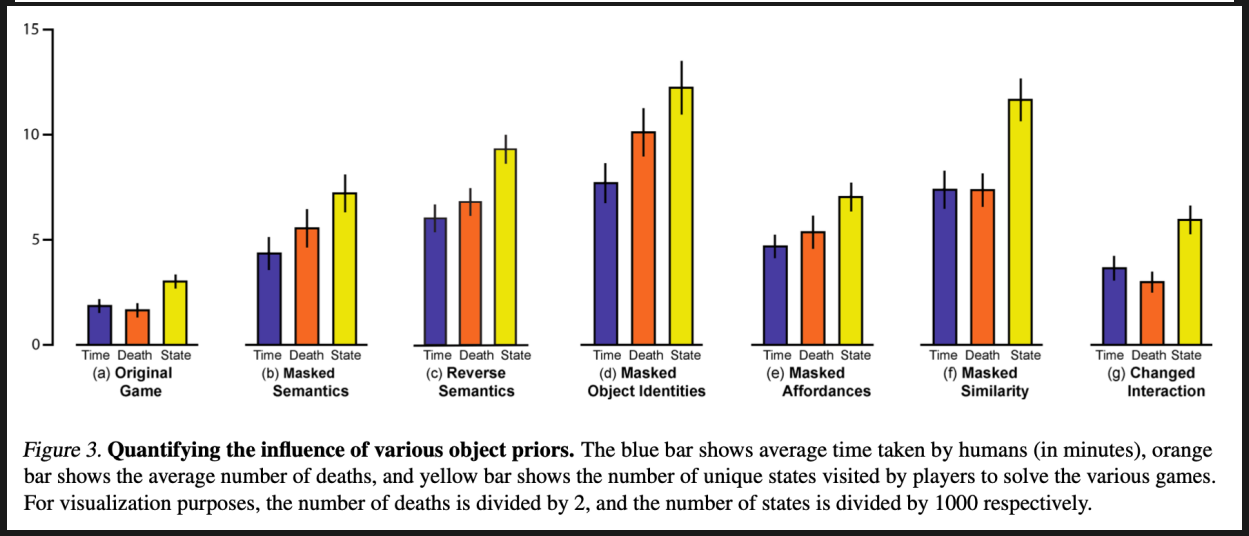


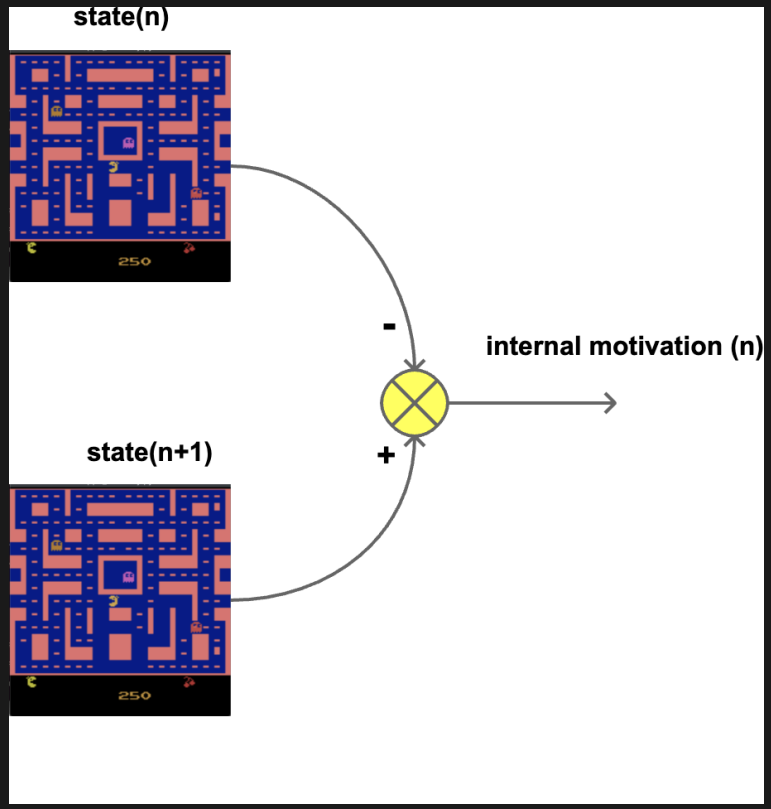
Figure 3. **Quantifying the influence of various object priors.** The blue bar shows average time taken by humans (in minutes), orange bar shows the average number of deaths, and yellow bar shows the number of unique states visited by players to solve the various games. For visualization purposes, the number of deaths is divided by 2, and the number of states is divided by 1000 respectively.

[4] Dubey et al. 2018

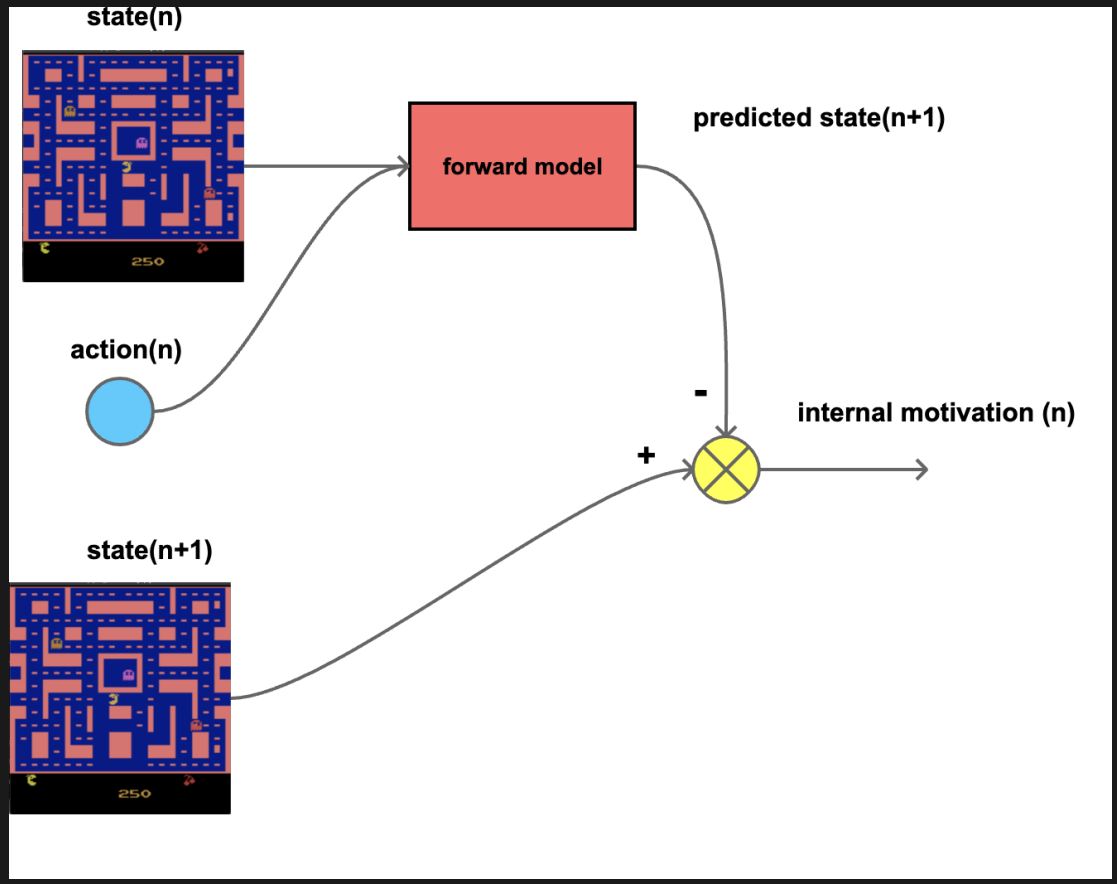
problems

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

pixel change motivation



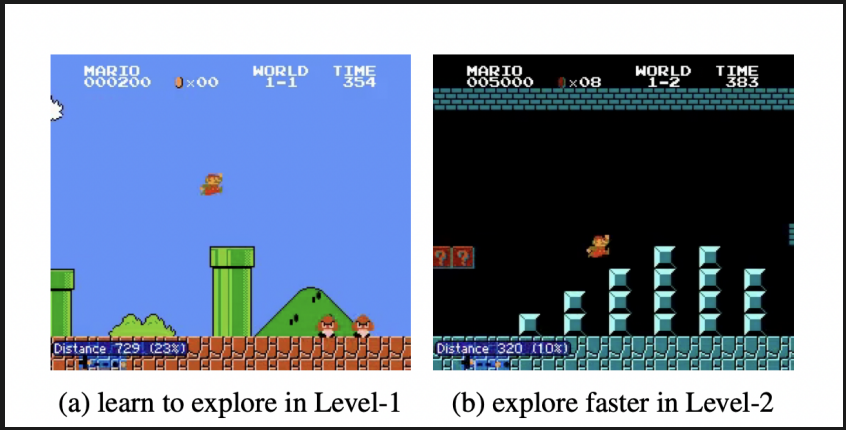
next state prediction



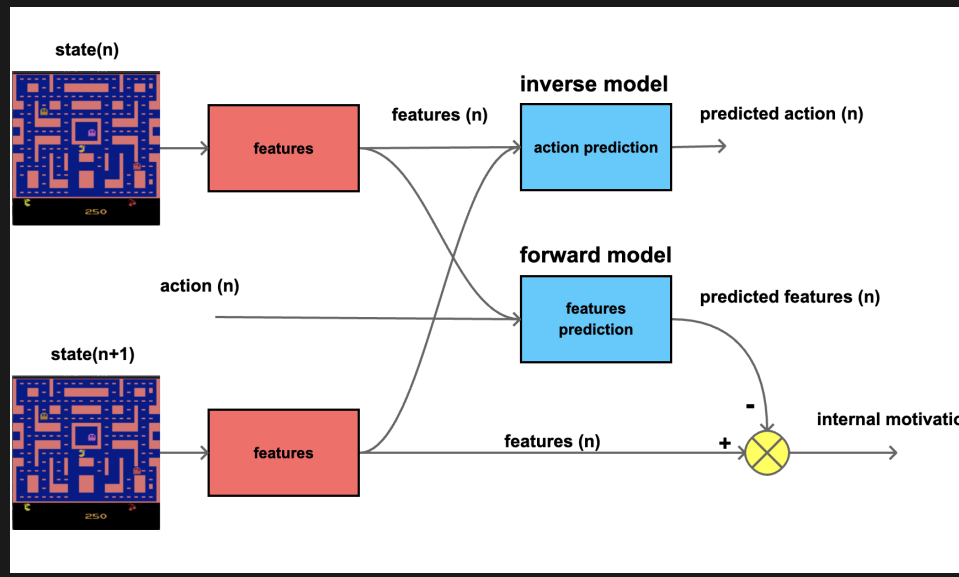
Curiosity-driven Exploration by Self-supervised Prediction [a]

[a] Pathak et al. 2017

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

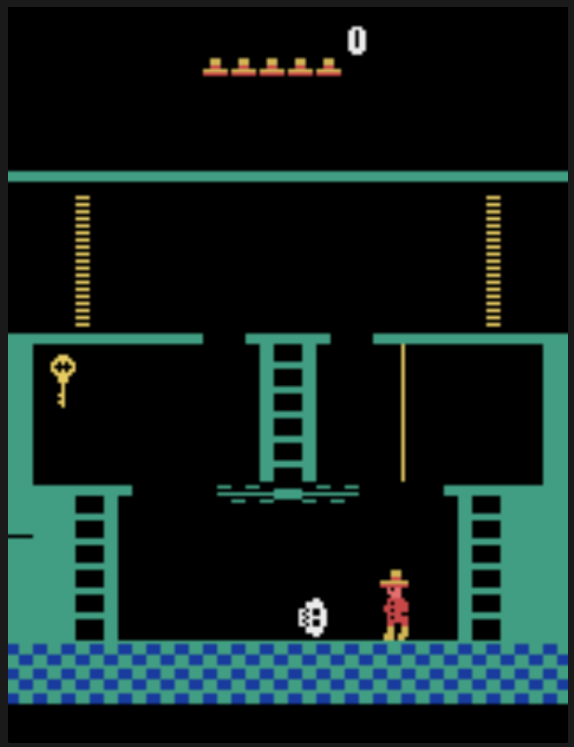


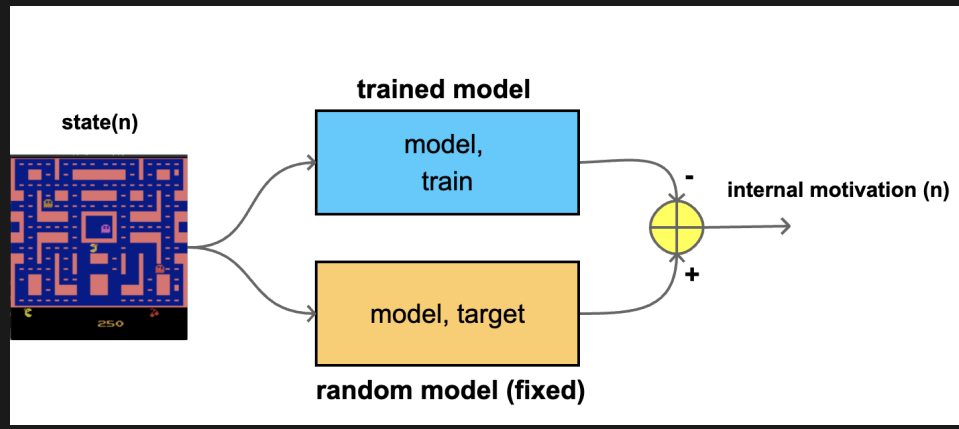
(a) learn to explore in Level-1 (b) explore faster in Level-2



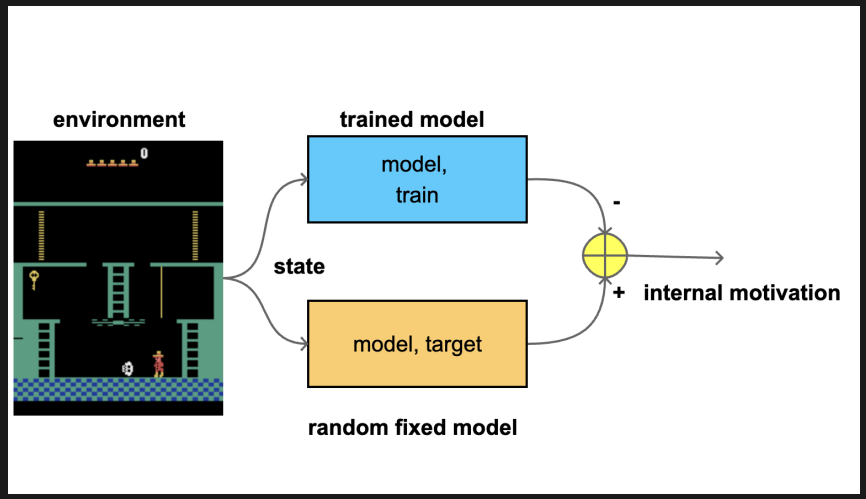
Exploration by Random Network Distillation [a]

[a] Burda et al. 2018



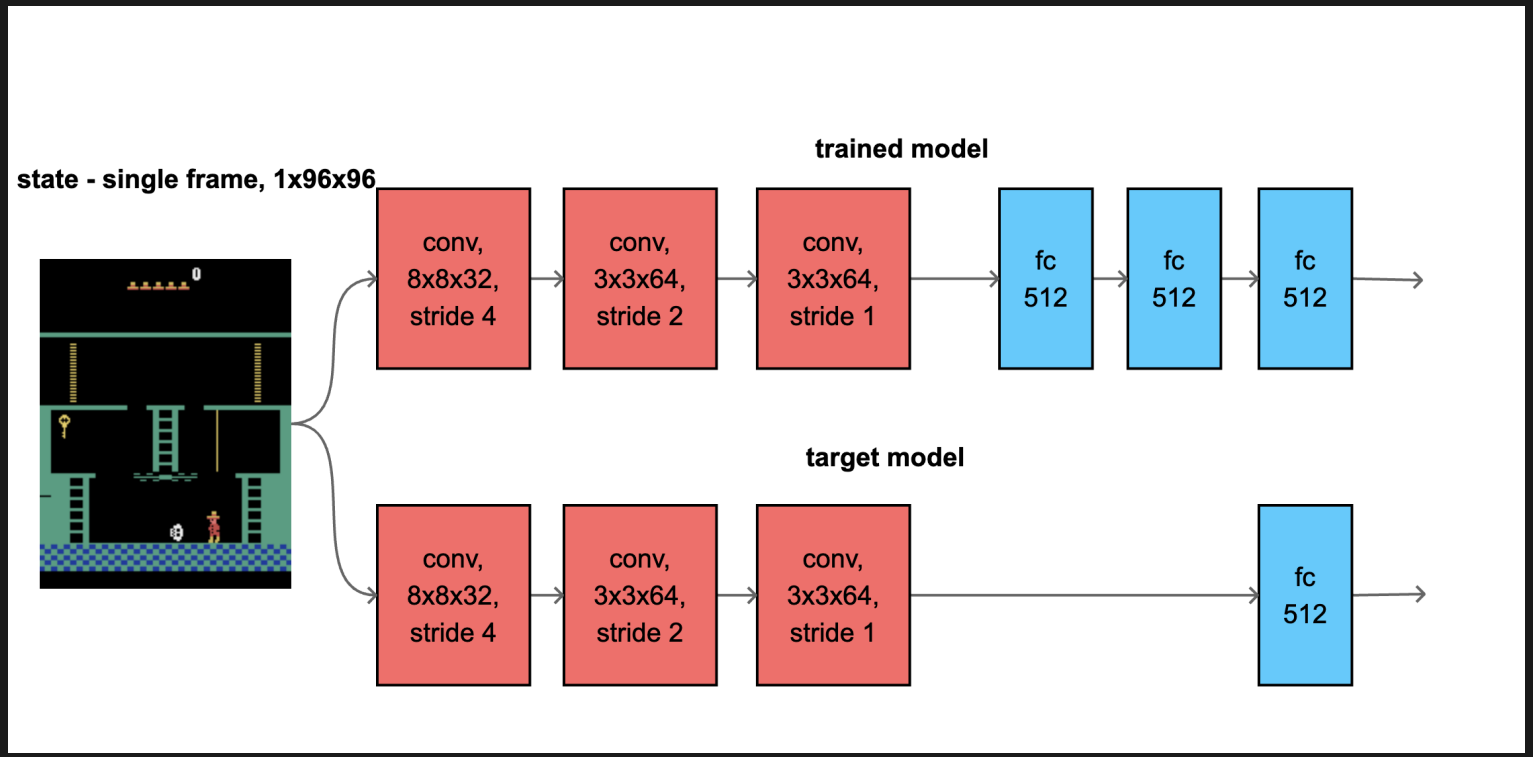


random network distillation

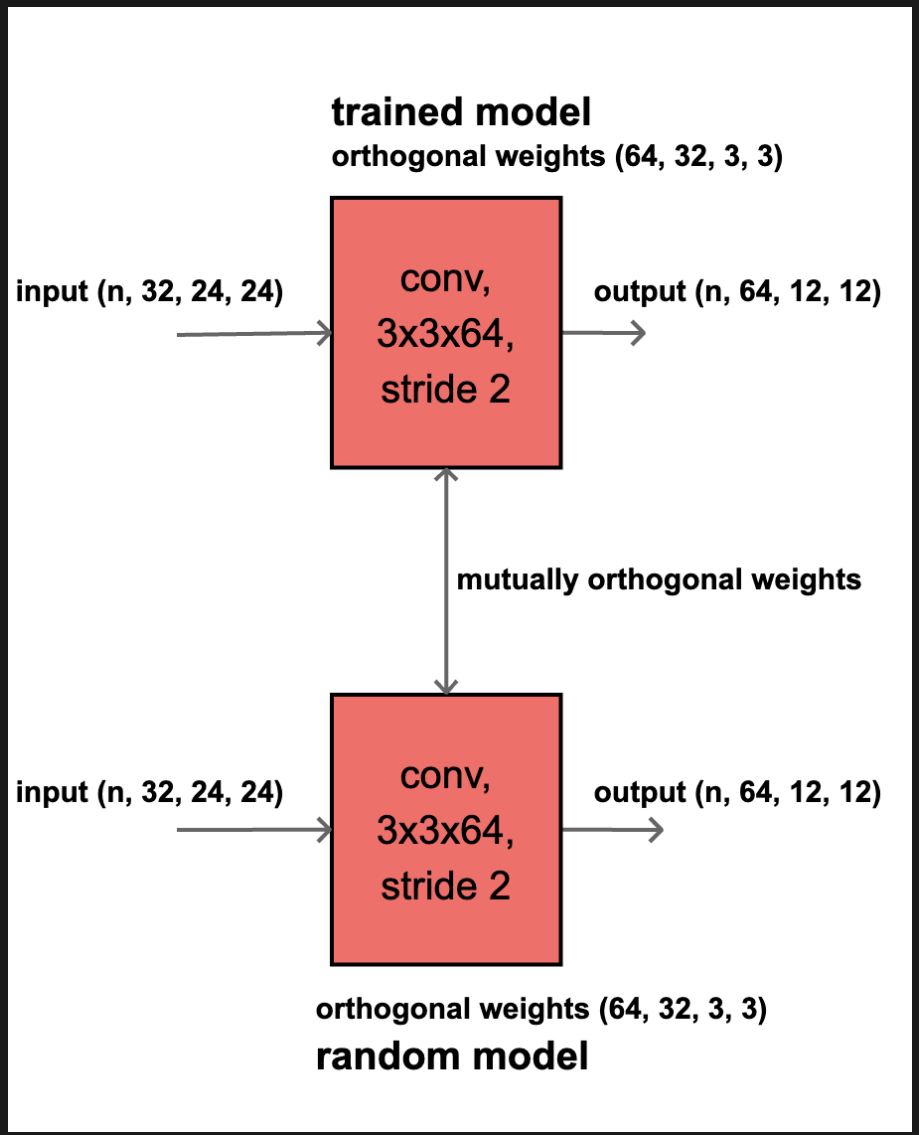


- neural network works as **novelty detector**
- model learns to imitate random (target) model
- **less visited states produce bigger motivation signal**
- orthogonal weights initialisation ($g = 2^{0.5}$) for strong signal
- lot of fully connected layers **to avoid generalisation**
- **coupled orthogonal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

random network distillation architecture



coupled RND architecture

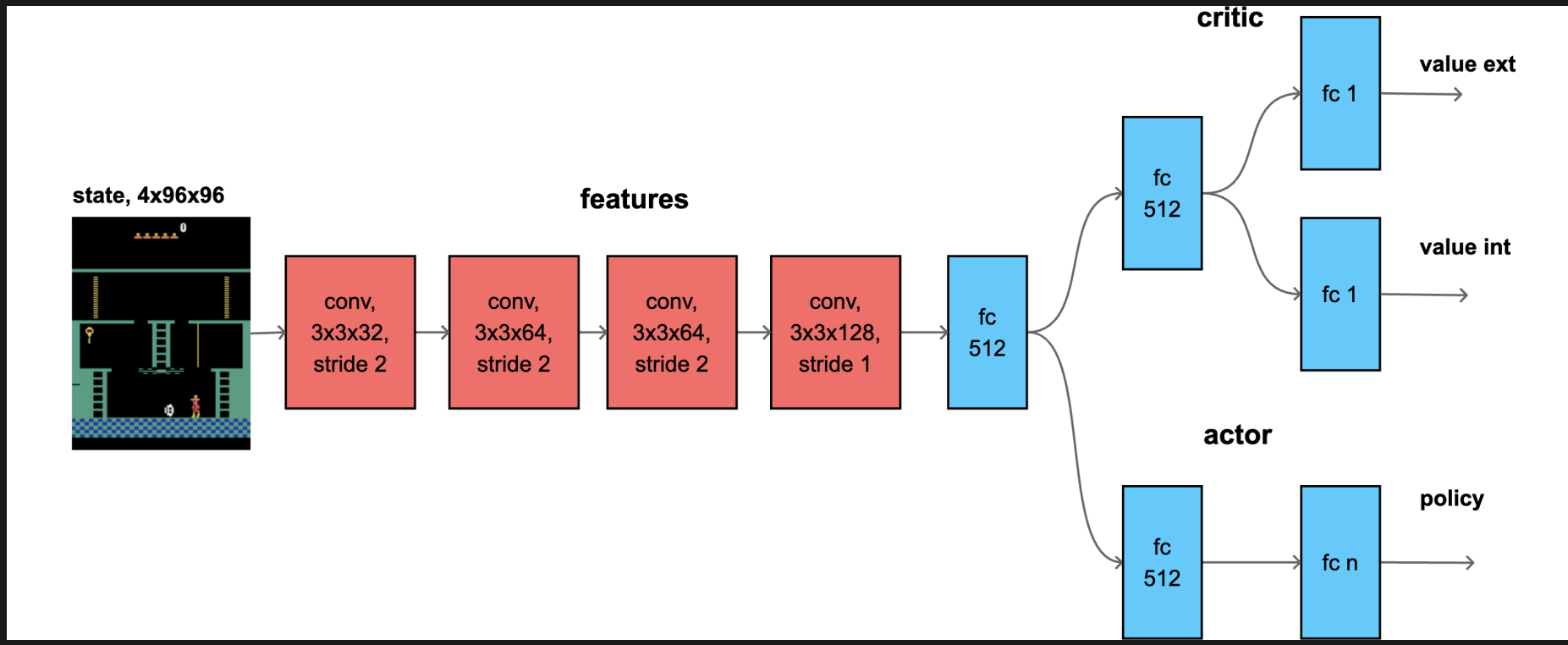



```python
def coupled_ortohogonal_init(shape, gain):
    w = torch.zeros((2*shape[0], ) + shape[1:])
    torch.nn.init.orthogonal_(w, gain)

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

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

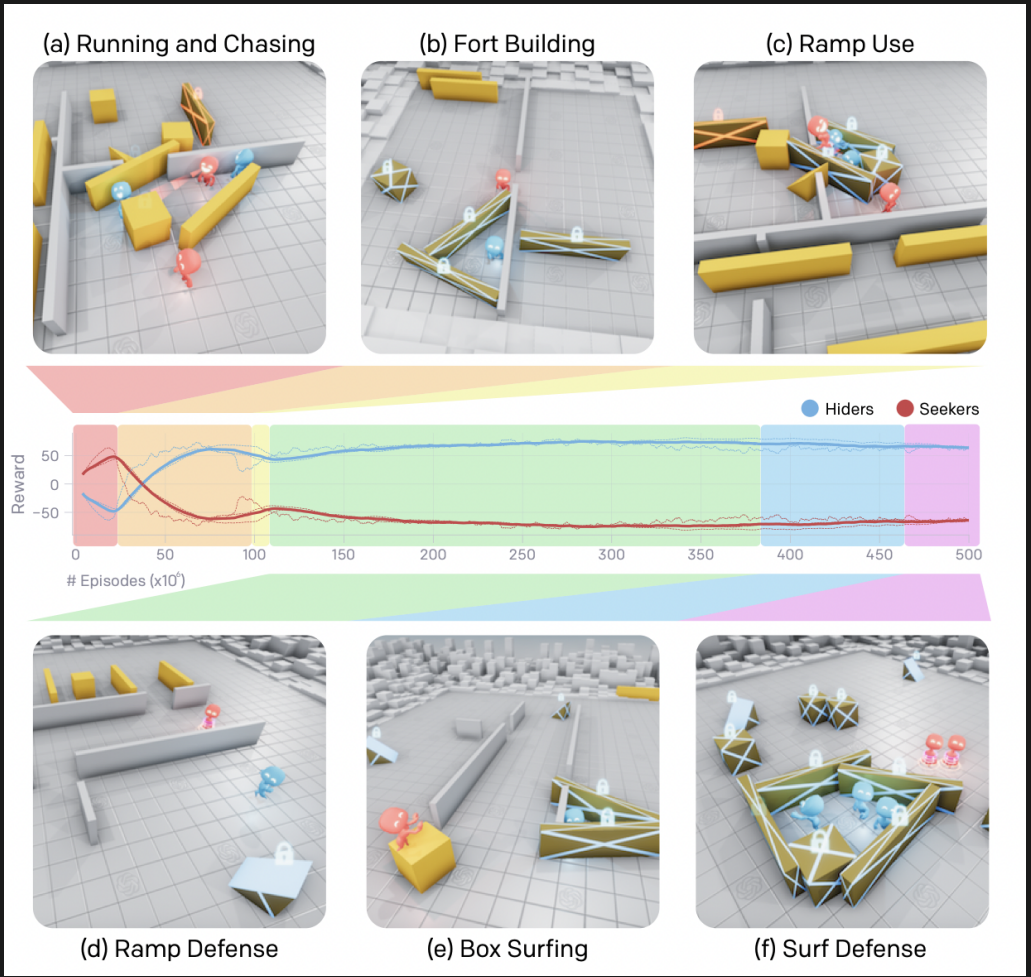
ppo model architecture



hide and seek [a]

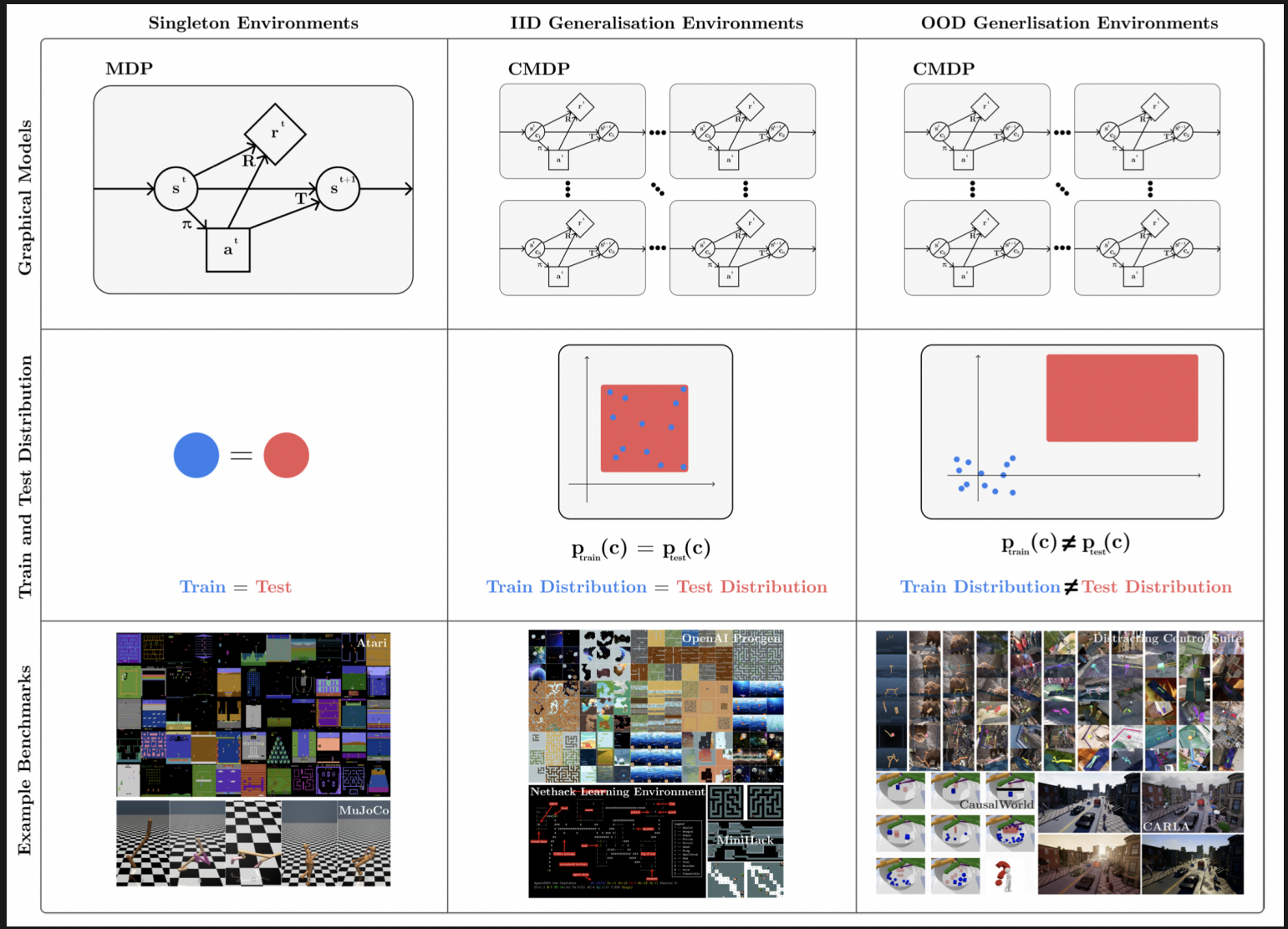
[a] Baker et al. 2020

OpenAI - Emergent Tool Use from Multi-Agent Interaction



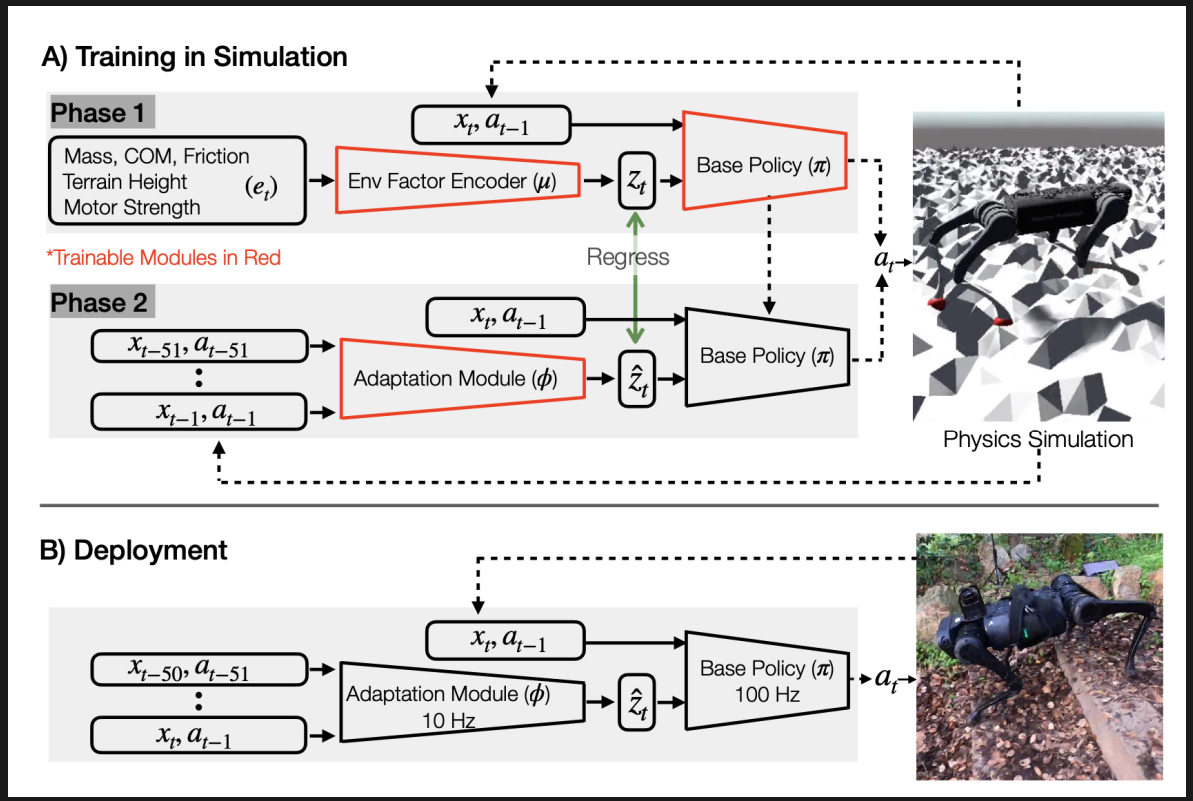
generalisation [a]

[a] Kirk et al. 2021



RMA: Rapid Motor Adaptation for Legged Robots [a]

[a] 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 [5] $4.5 * 10^9$ samples, score 8152 on MR
- Never give up [6] $3.5 * 10^{10}$ samples, score 10 000 on MR
- CND $1.28 * 10^8$ score 25 000 on MR

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

[5] Burda et al. 2018
[6] Badia et al. 2020

misleading papers - Curiosity-driven Exploration by Self-supervised Prediction [a]

[a] Pathak et al. 2017

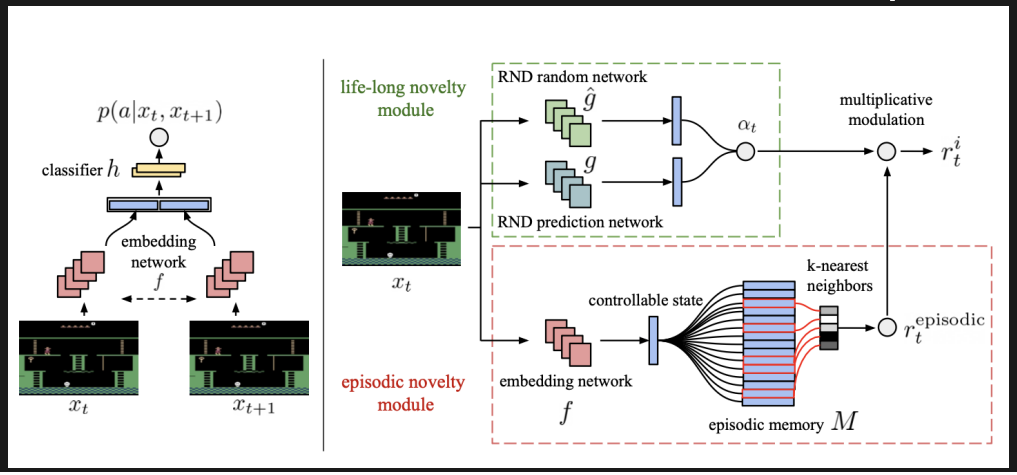
- not working on REAL hard exploration problems
- Super Mario is special case - moving forward is close to optimal policy
- inverse ICM model - why they didn't shown accuracy (my results arround 40% !!! even given policy)
- how predicted state looks ?

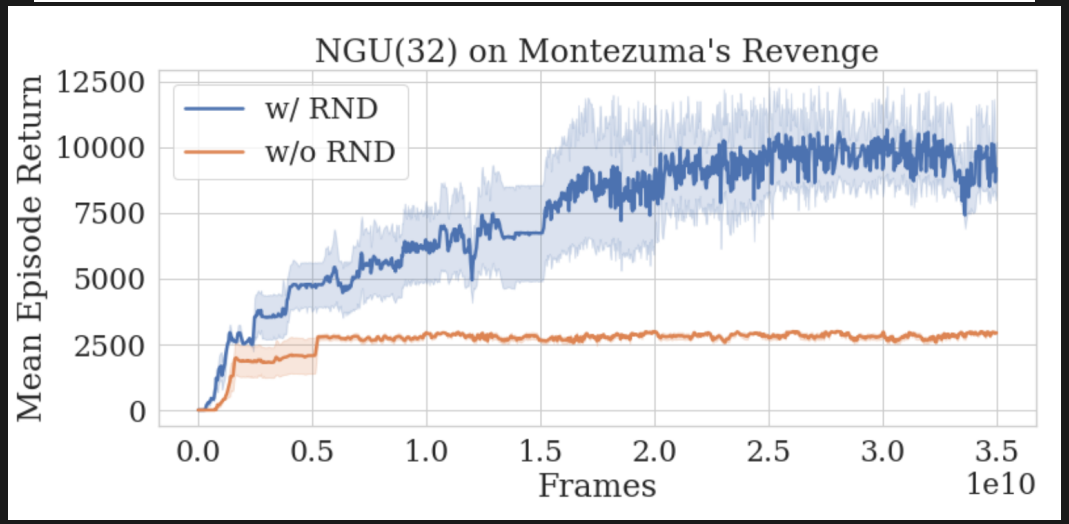
	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	**3,906**	**8,152**	-3	8,666	3,282	**1,859**
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209[1]	3,700[2]	**0**	**15,806**[2]	**12,380**[1]	**1,813**[3]
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

misleading papers - Never give up [a]

[a] Badia et al. 2020

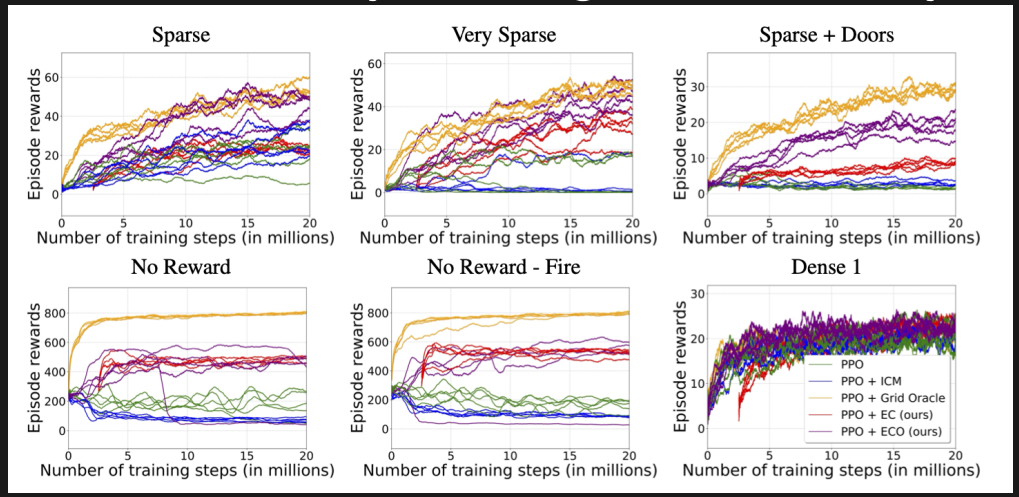
nice looking score, but on cost of $3.5 * 10^{10}$ samples !!!





other misleadings

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



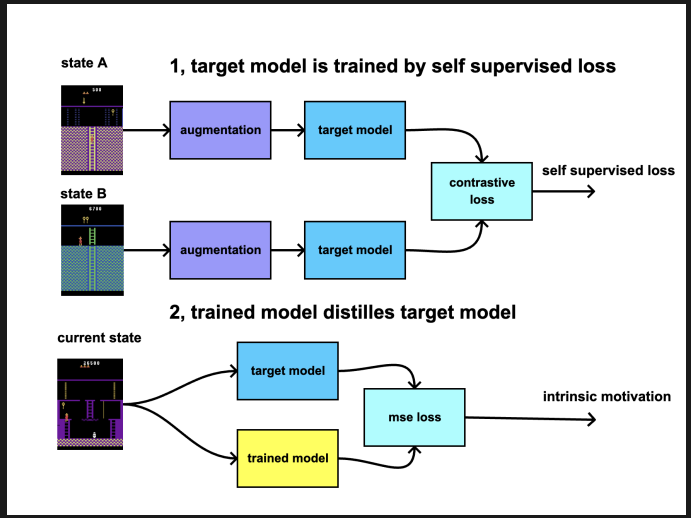
many other :

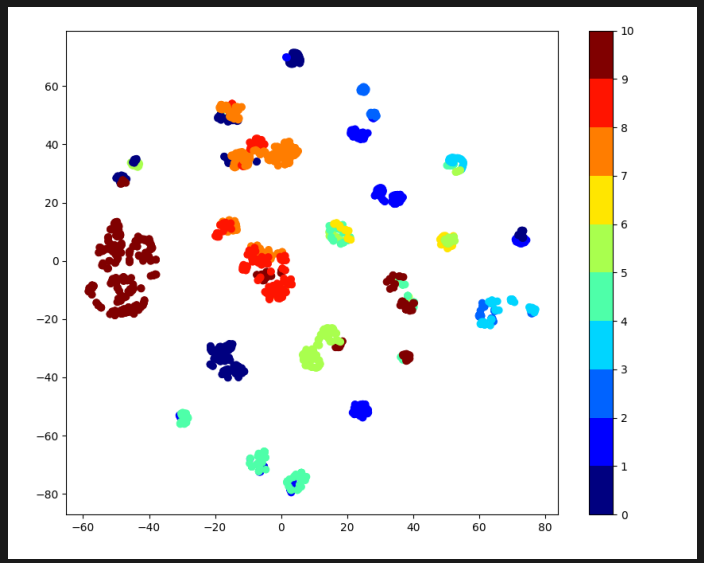
- simple gridworld or toy environment experiments
- providing key prior information (e.g. possition)
- selecting only "good" results

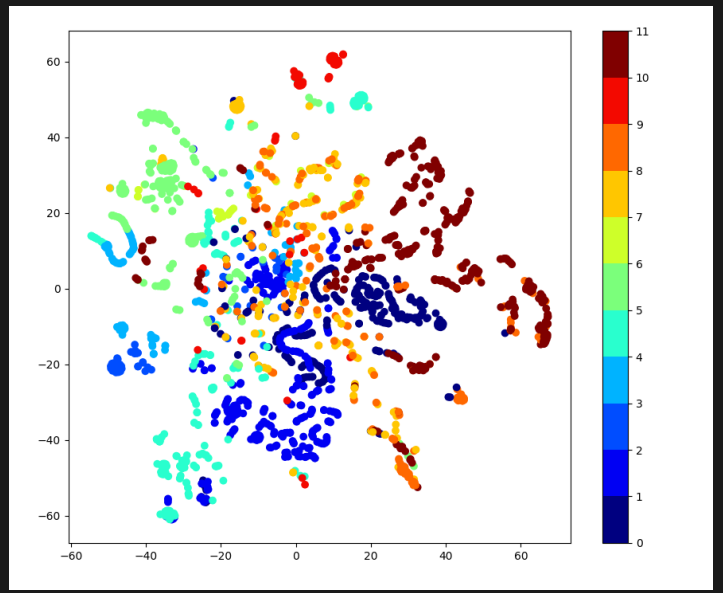
my current research

- 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







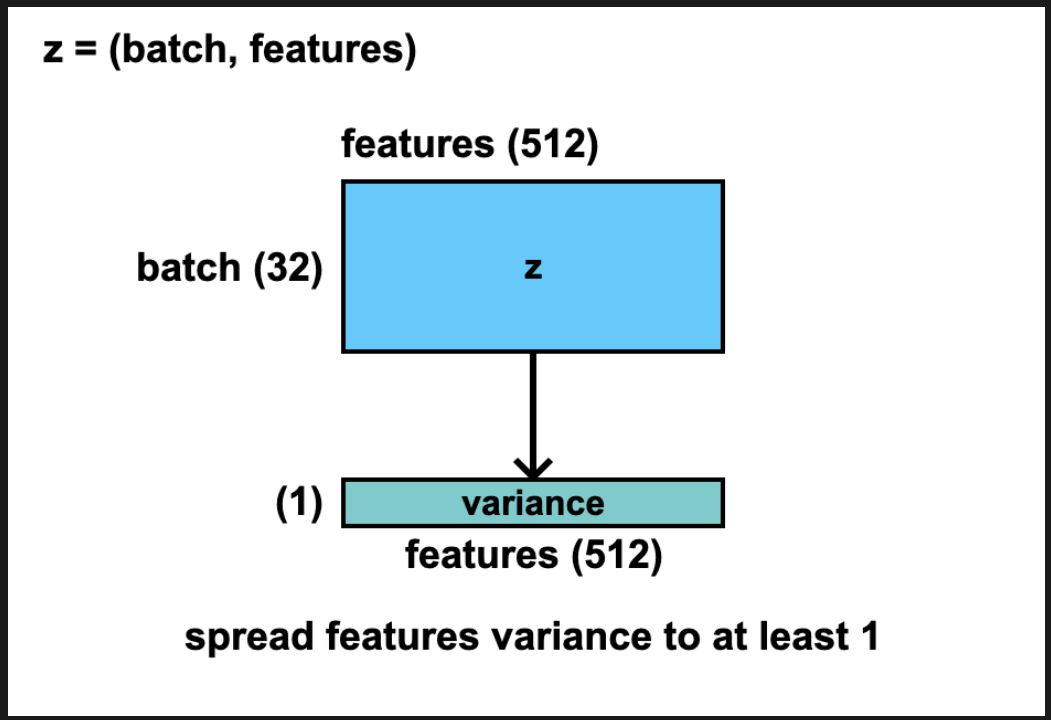
vicreg self supervised loss

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

$$
\begin{aligned}
\mathcal{L} =& \alpha \max\left(1 - \sum_{f}^{F} var_0(z_a) + var_0(z_b), 0\right) && \text{variance} \\
&+ \beta \sum_{n}^{N} \begin{cases} \sum_{f}^{F} (z_a - z_b)^2, & \text{if similar} \\ \max(1 - \sum_{f}^{F} (z_a - z_b)^2, 0), & \text{otherwise} \end{cases} && \text{invariance} \\
&+ \sum_{f}^{F} (1 - \mathbb{I}) \left((z_a^T z_a)^2 + (z_b^T z_b)^2 \right) && \text{covariance}
\end{aligned}
$$

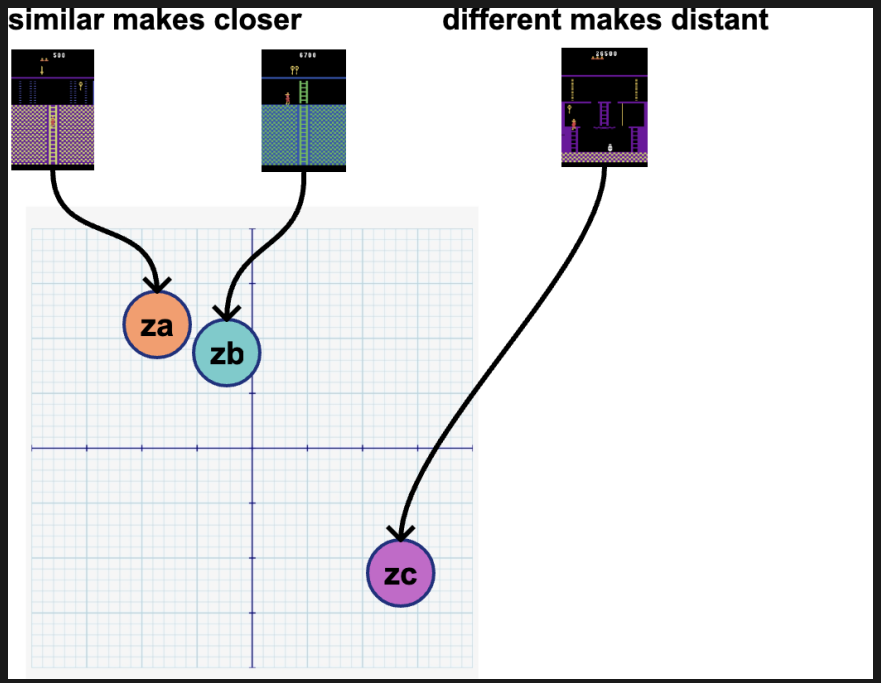
variance

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



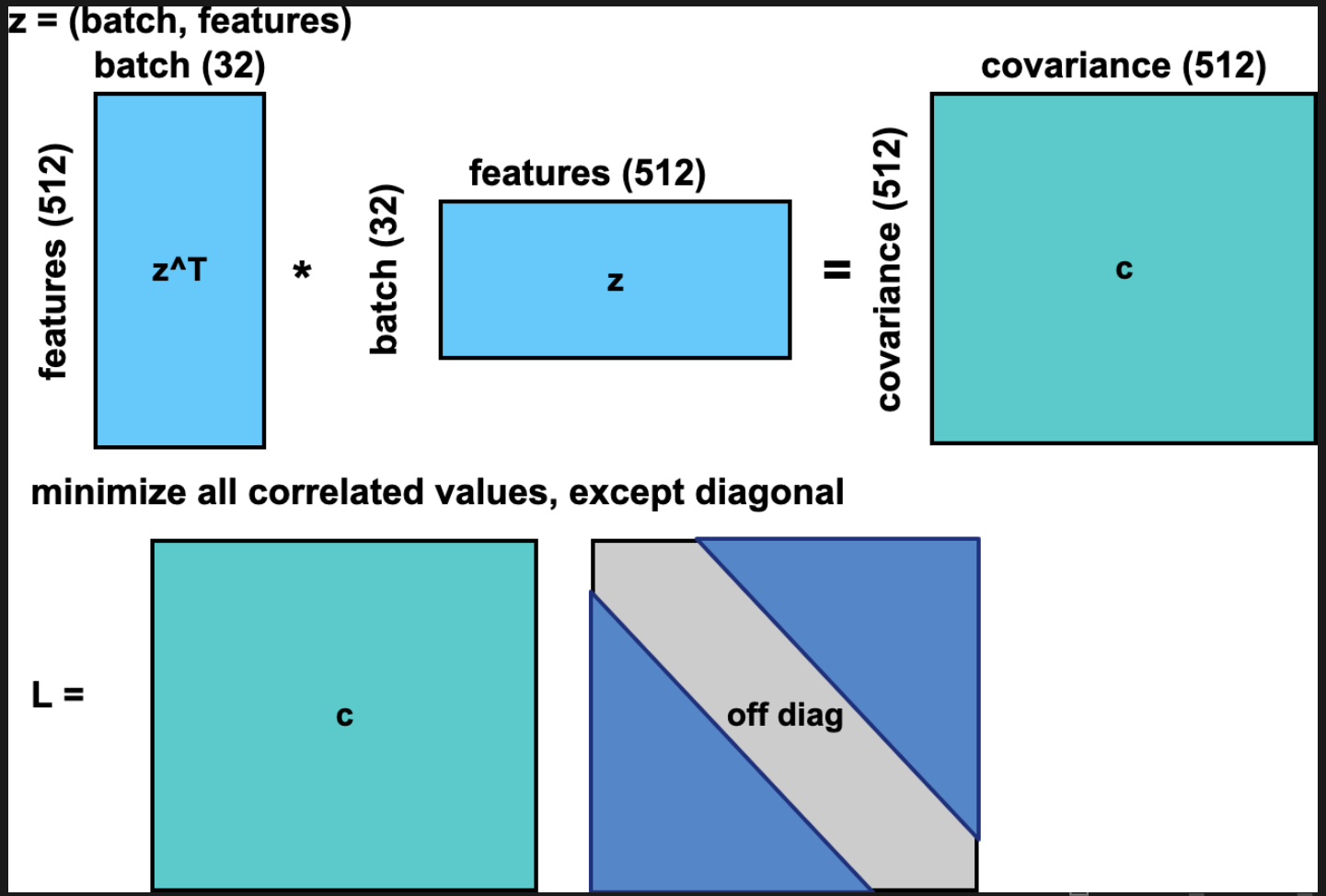
invariance

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

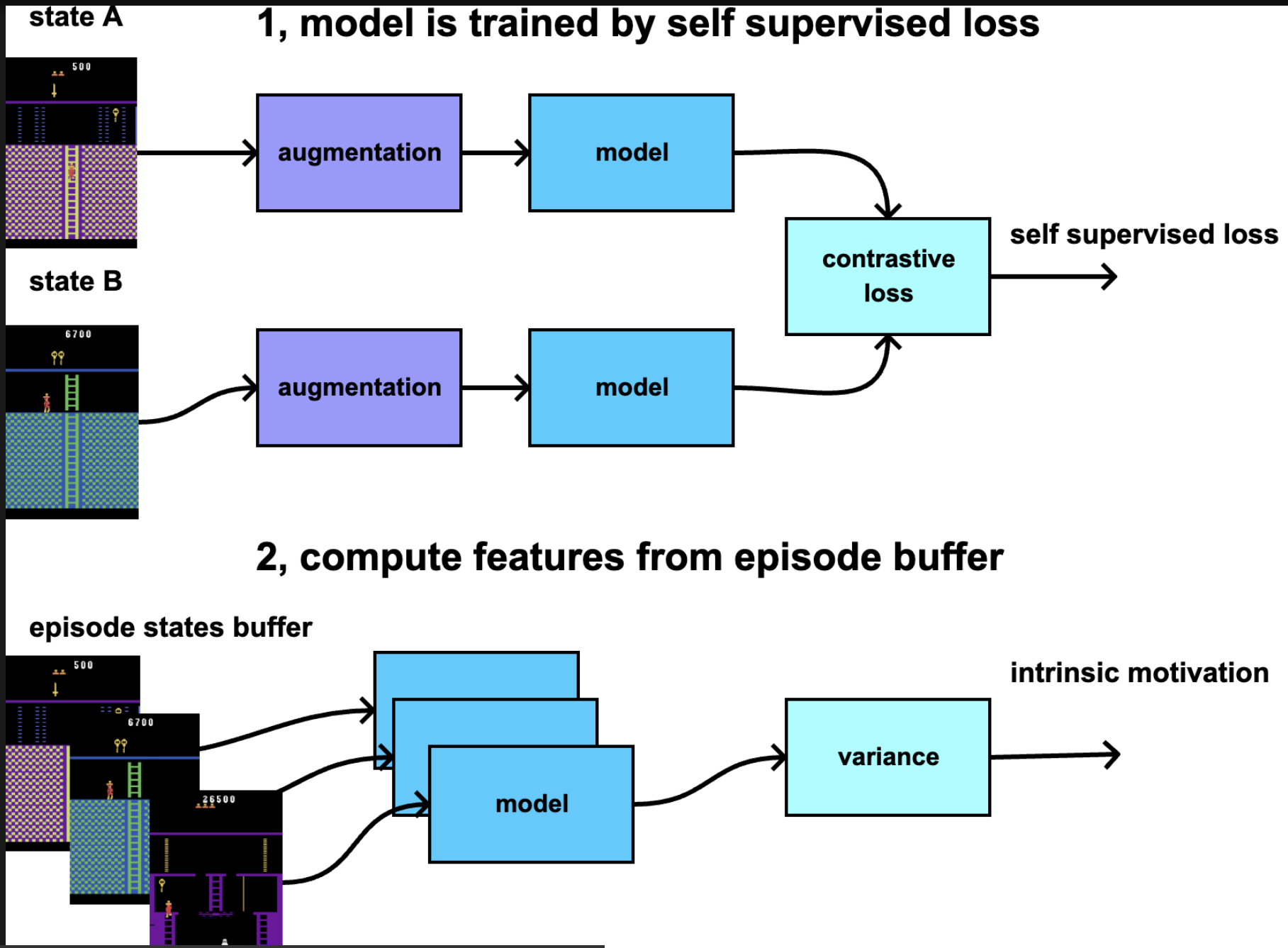


covariance

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

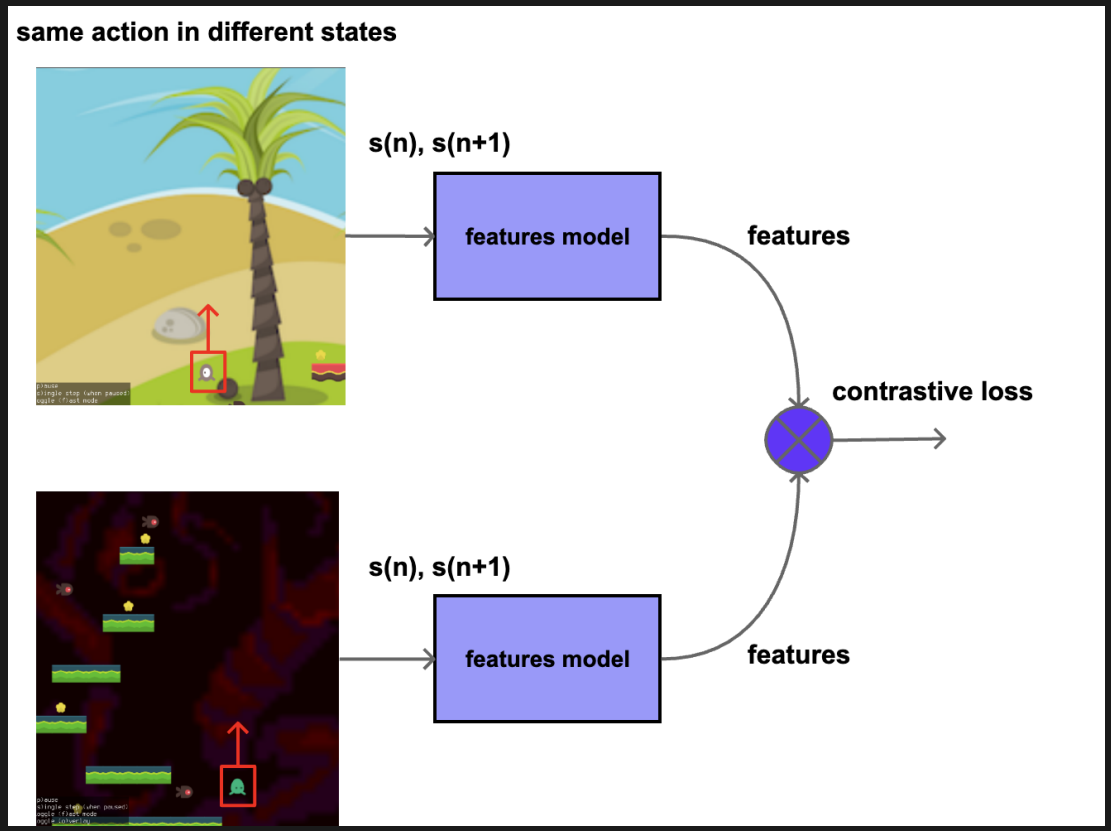


raising entropy driven exploration



symmetry driven generalisation [a]

[a] Bronstein et al. 2021 Geometric deep learning - some trivial hand-crafted symmetries



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

[7] 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