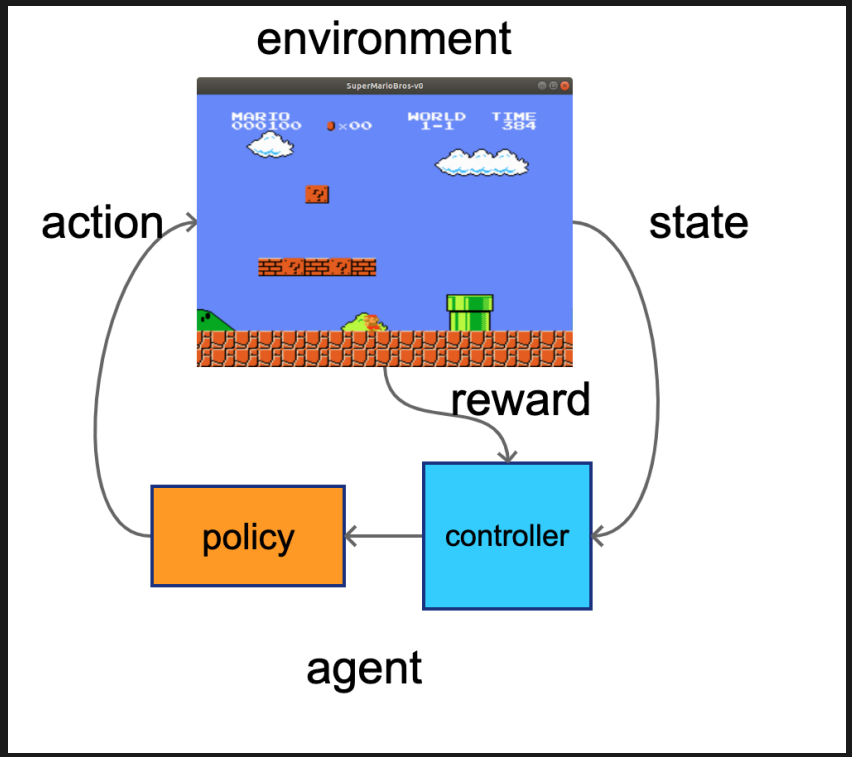
reinforcement learning current problems

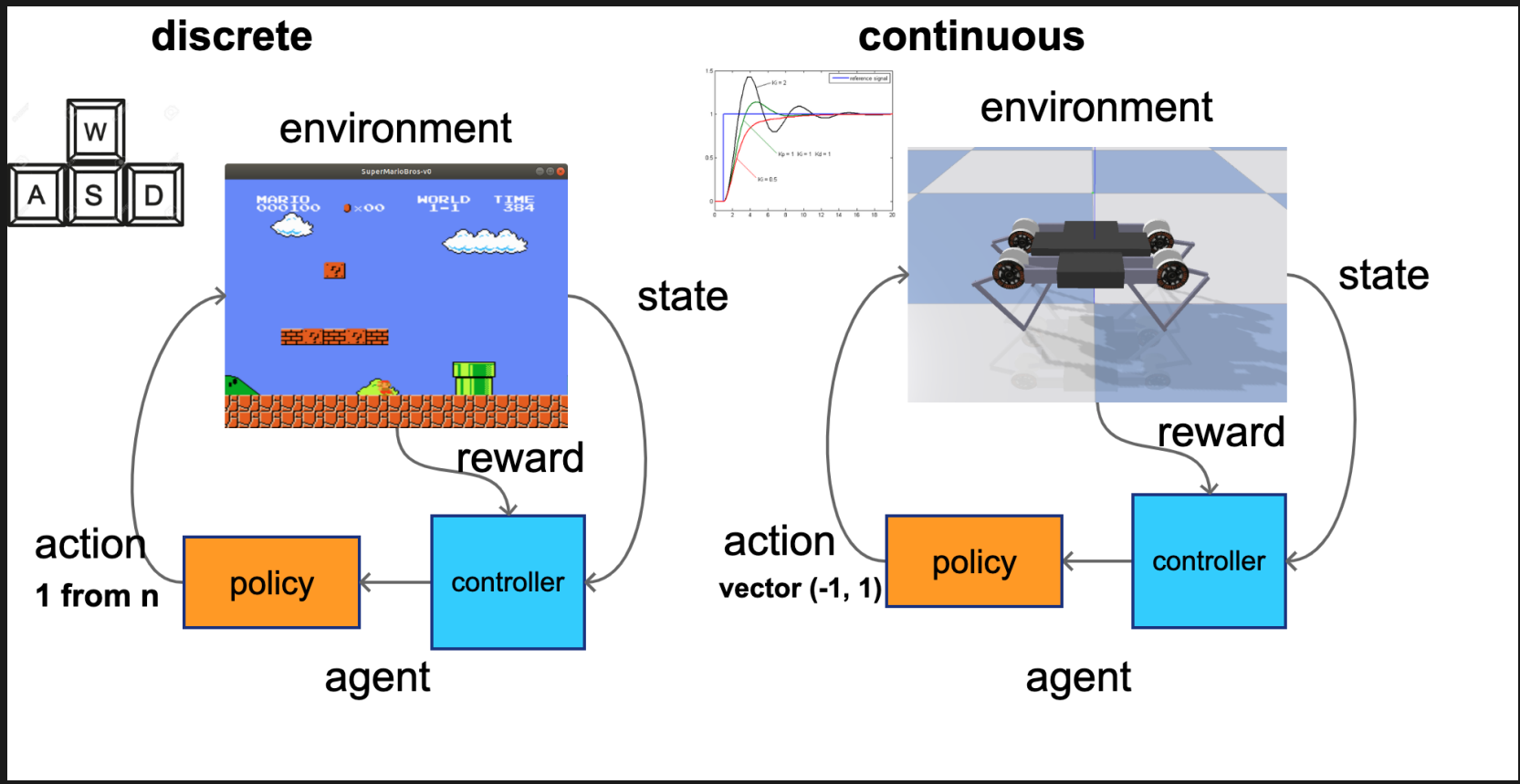
Michal CHOVANEC, PhD.

reinforcement learning



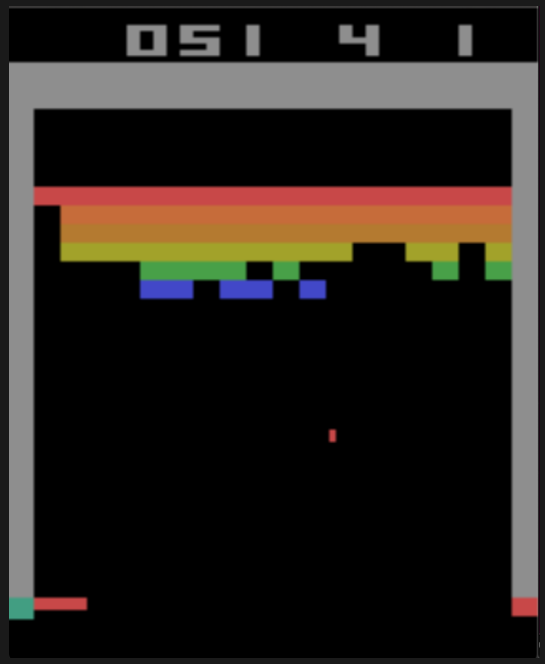
1. **obtain state** - observation
2. **choose action** - policy
3. **receive reward**
4. **learn from experiences**

action space



when it works ?

- **dense rewards**
- **small state space**
- **fully observable**



when it works - AlphaGO, AlphaZero - DeepMind

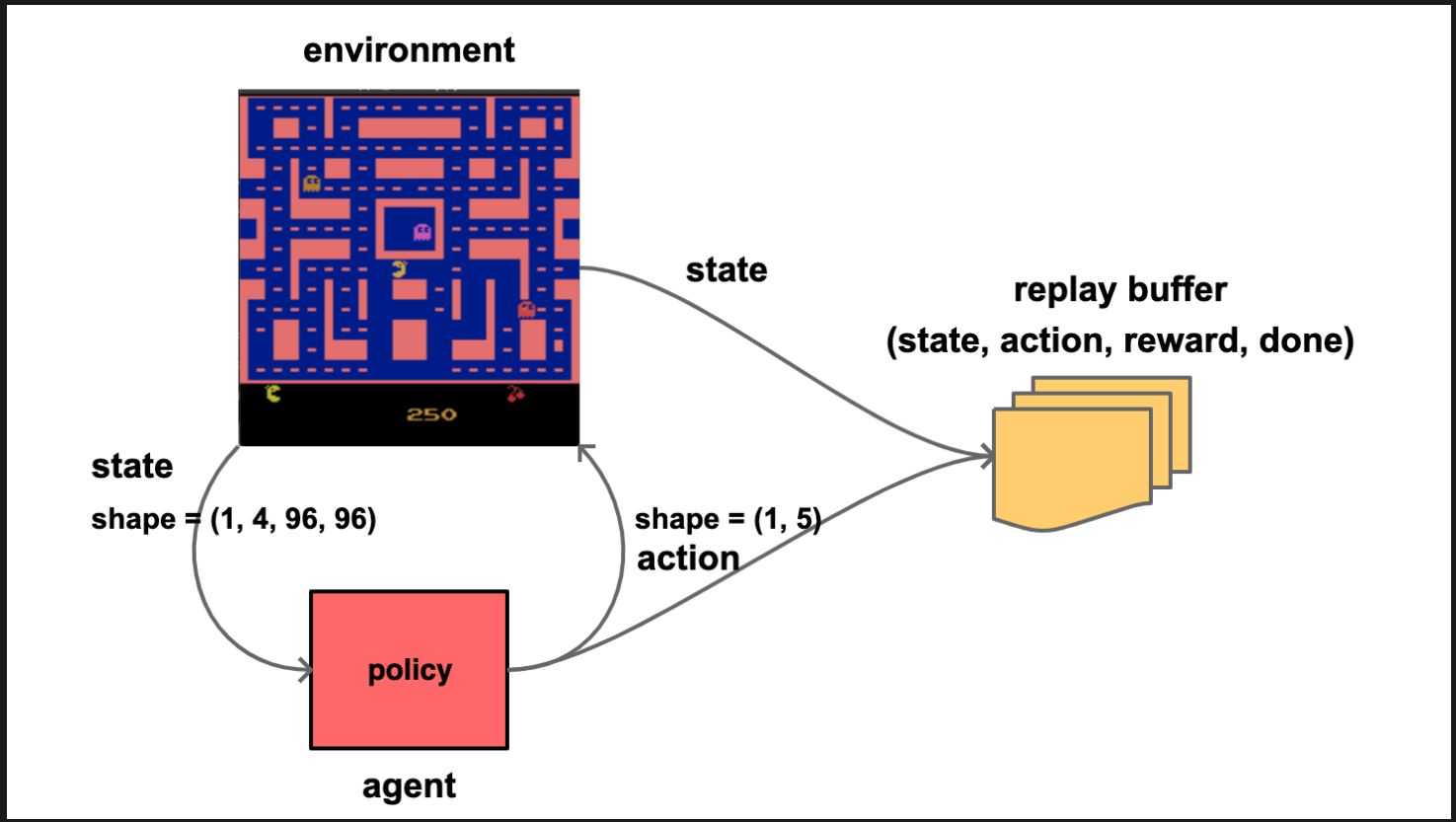
Chess with Suren, AlphaZero's Most Astonishing "Zugzwang" Game

most famous algorithms

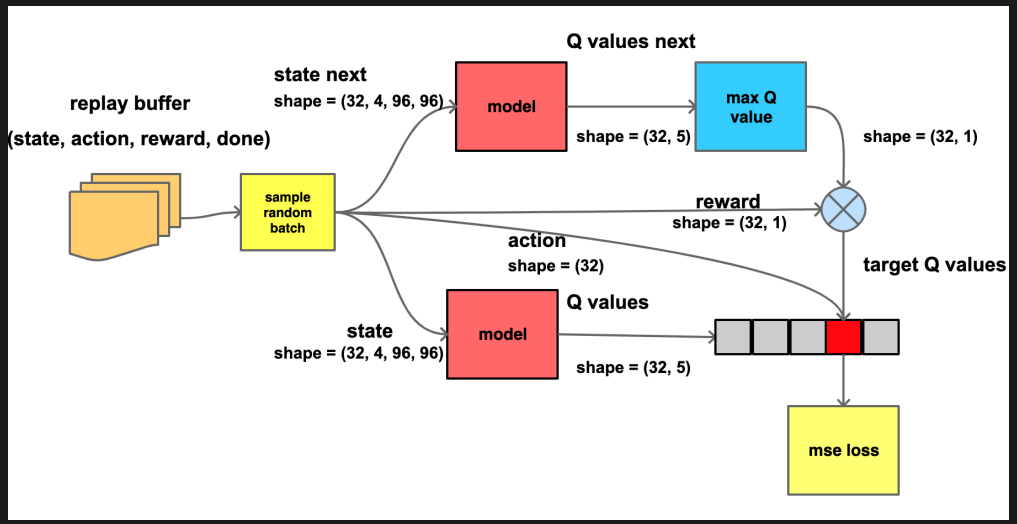
- deep Q network - **DQN** [1]
- deep deterministic policy gradient - **DDPG**, D4PG [2]
- advantage actor critic - AC, **A2C**, A3C
- proximal policy optimization - **PPO**, TRPO [3]

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

DQN - Playing Atari with Deep Reinforcement Learning



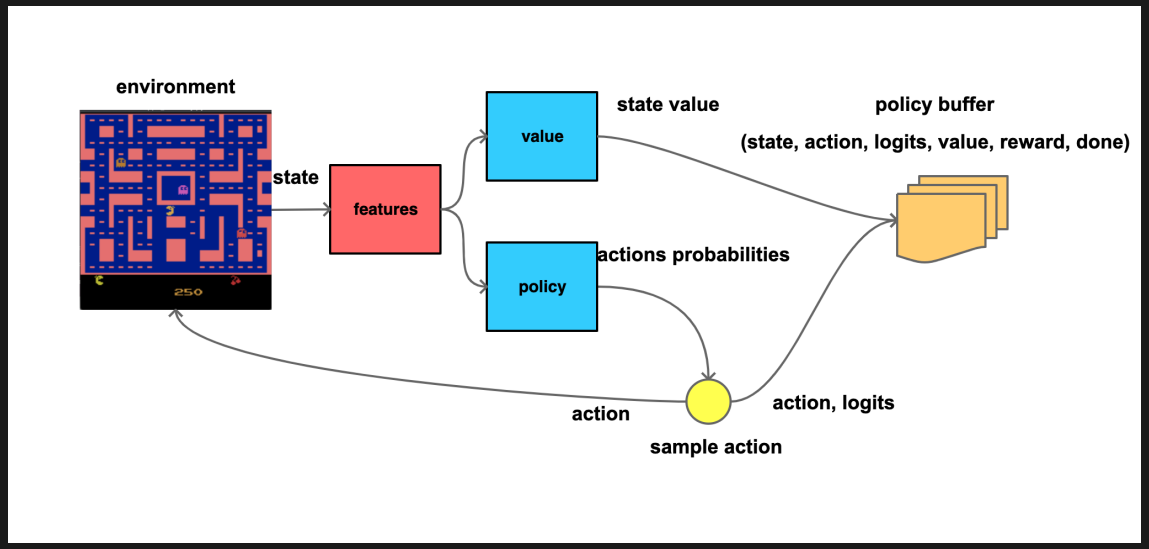
DQN - Playing Atari with Deep Reinforcement Learning



$$Q(s_n, a_n; \hat{\theta}) = R_n + \max_{\alpha \in \mathcal{A}} \hat{Q}(s_{n+1}, \alpha; \hat{\theta})$$

$$\mathcal{L}_\theta = (Q(s, a; \hat{\theta}) - Q(s, a; \theta))^2$$

PPO - Proximal Policy Optimization Algorithms



$$\mathcal{L} = log\pi(a_n|s_n)A_n^{\pi}$$

$$\mathcal{L} = \frac{1}{N}\sum_{n}^{N}\frac{\pi^{now}(a_n|s_n)}{\pi^{prev}(a_n|s_n)}A_n^{\pi^{old}}$$

- naive policy gradient - unstable
- minimize policy divergence
- clipping or KL-divergence

problems

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

Montezuma's revenge - 10 years of tears?

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

year	name	score
2013	Playing Atari with Deep Reinforcement Learning	0
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

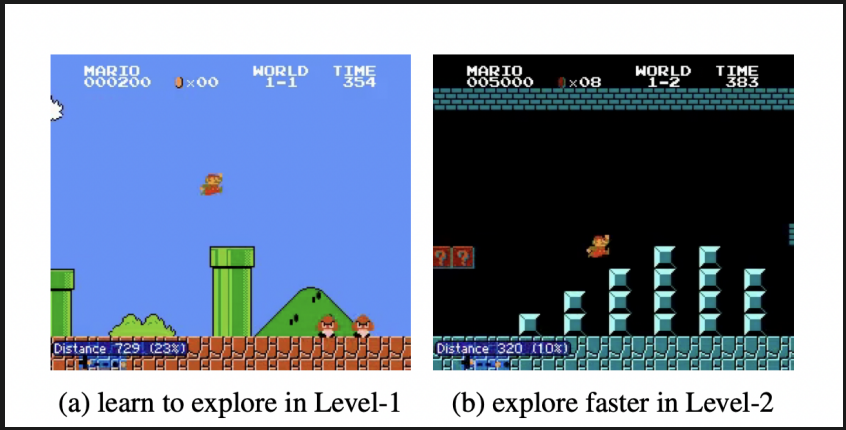
*** requires environment state saving/loading**

[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

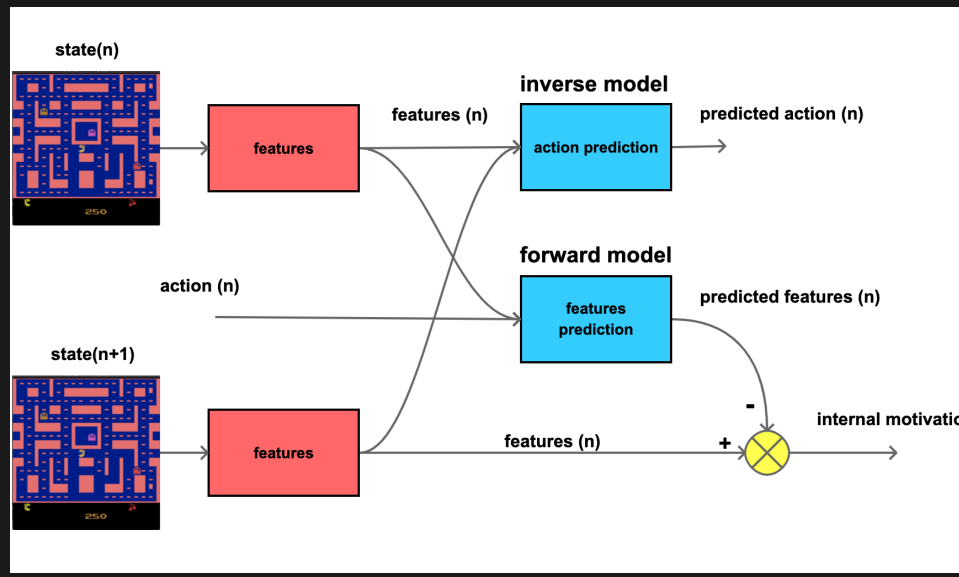
Curiosity-driven Exploration by Self-supervised Prediction [a]

[a] Pathak et al. 2017

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

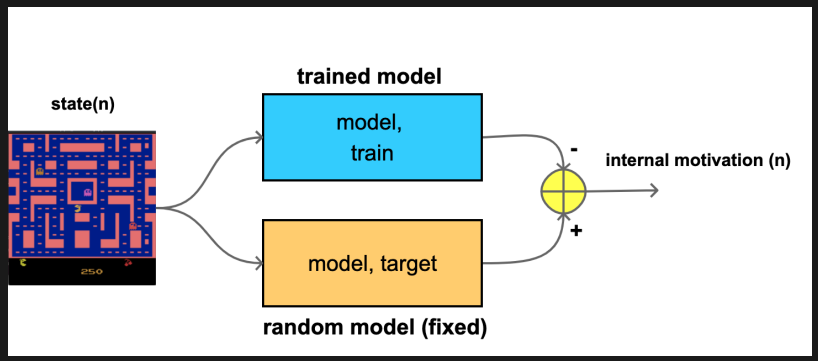


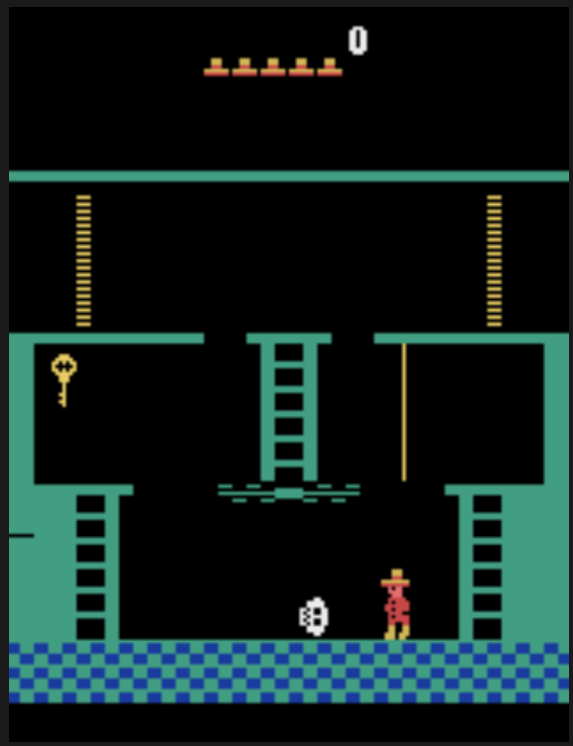
(a) learn to explore in Level-1 (b) explore faster in Level-2



Exploration by Random Network Distillation [a]

[a] Burda et al. 2018



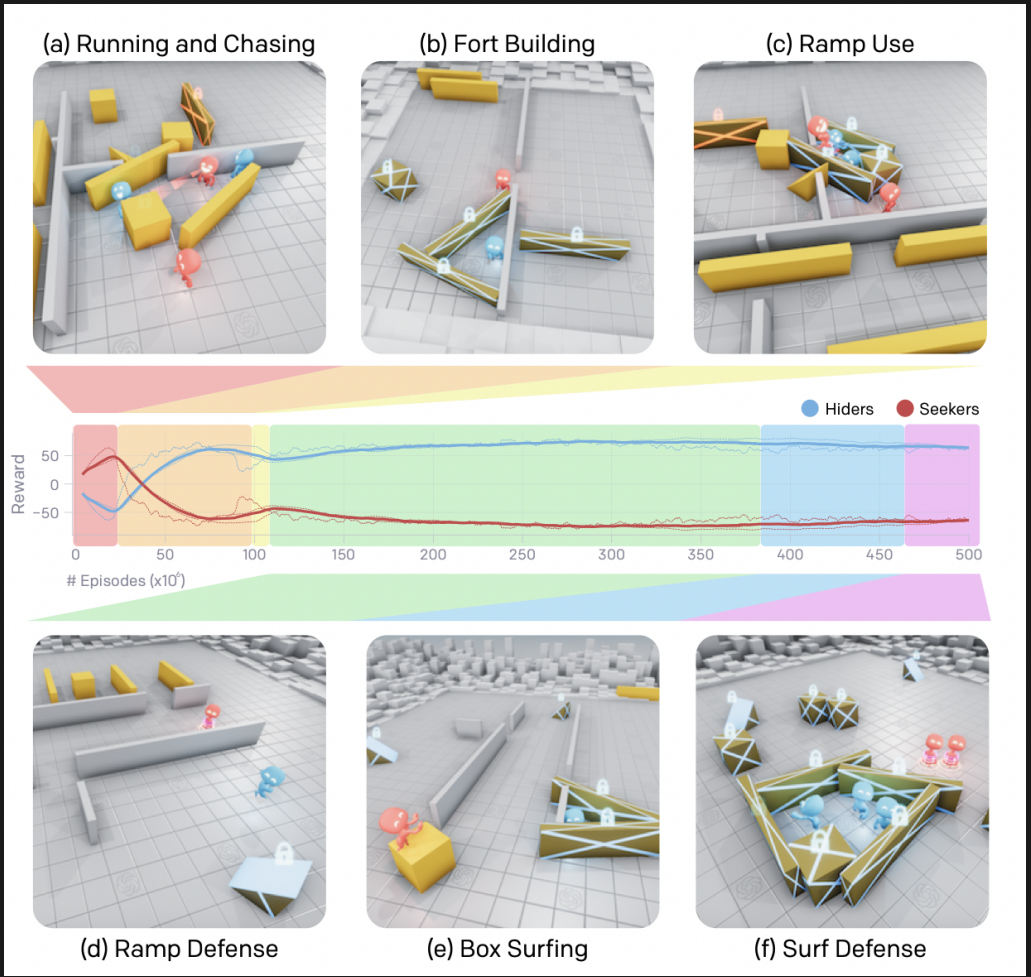


	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	**3,906**	**8,152**	-3	8,666	3,282	**1,859**
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209[1]	3,700[2]	**0**	**15,806**[2]	**12,380**[1]	**1,813**[3]
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

hide and seek [a]

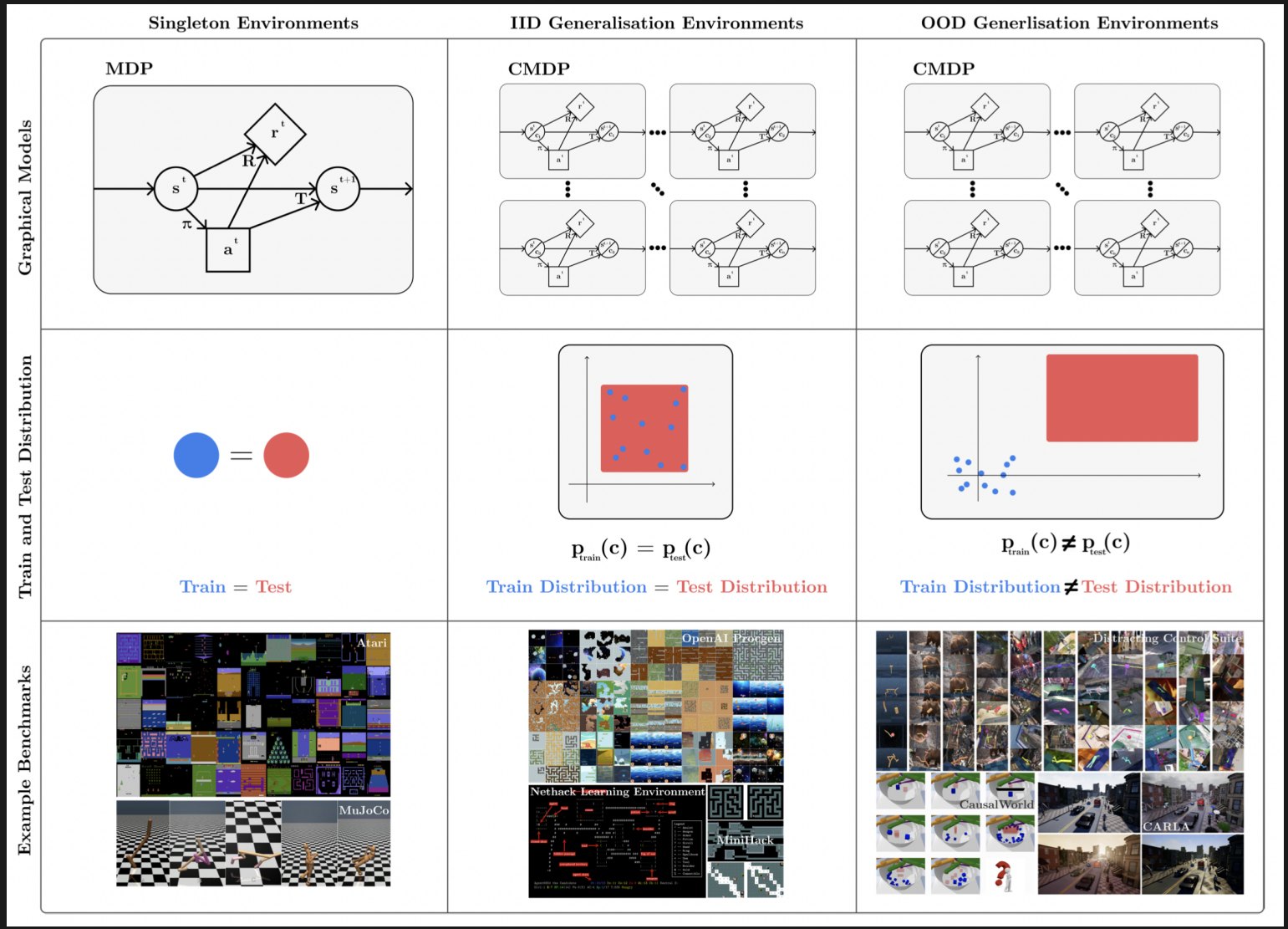
[a] Baker et al. 2020

OpenAI - Emergent Tool Use from Multi-Agent Interaction



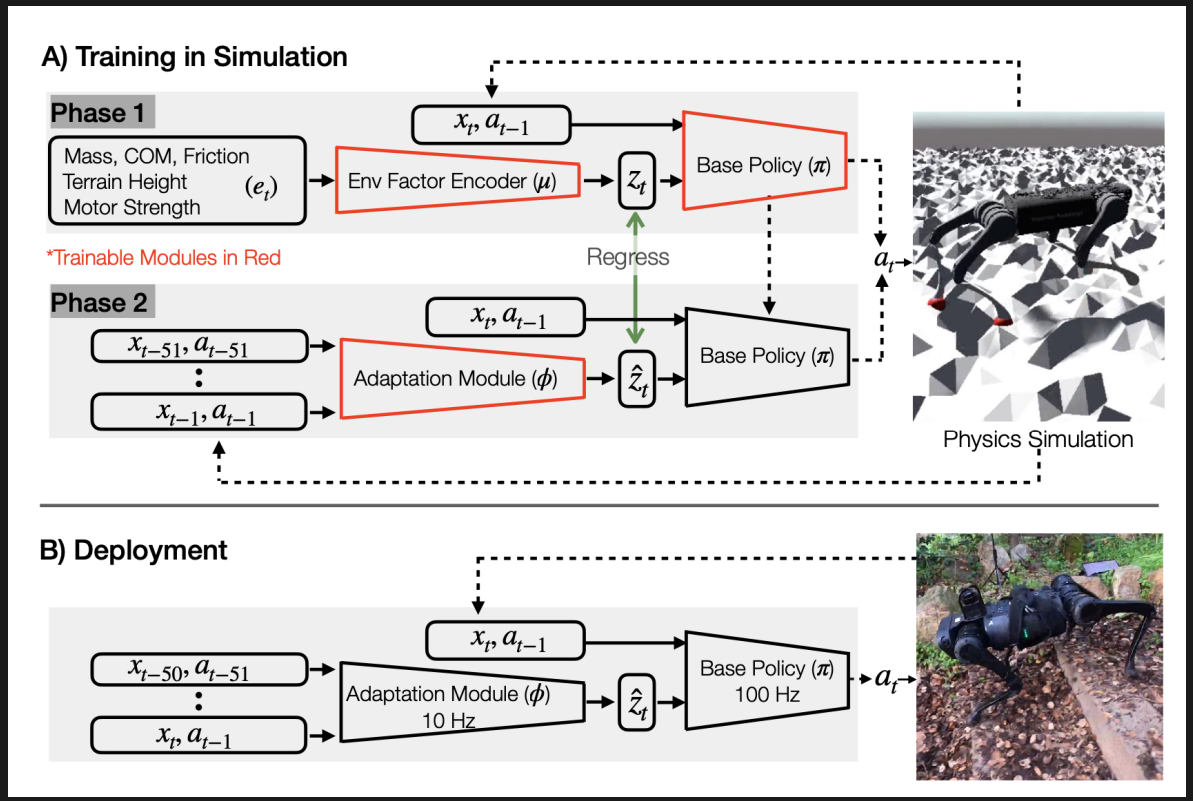
generalisation [a]

[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 [4] $4.5 * 10^9$ samples, score 8152 on MR
- Never give up [5] $3.5 * 10^{10}$ samples, score 10 000 on MR
- SND $1.28 * 10^8$ score 10 000 on MR

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

---

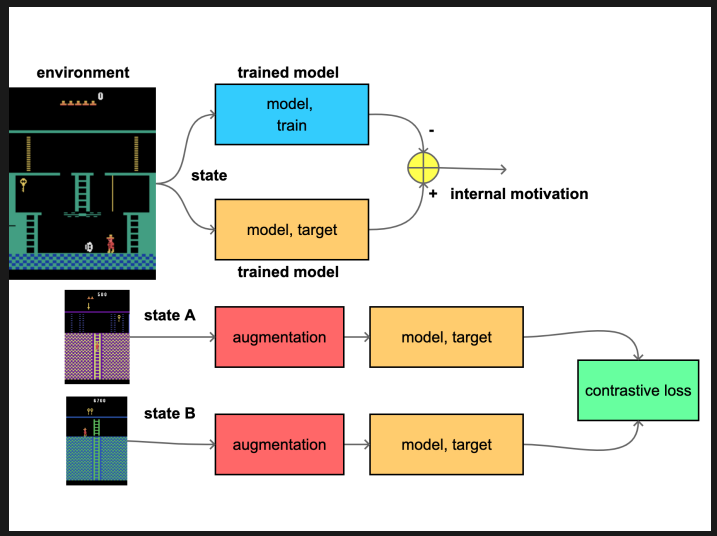
[4] Burda et al. 2018

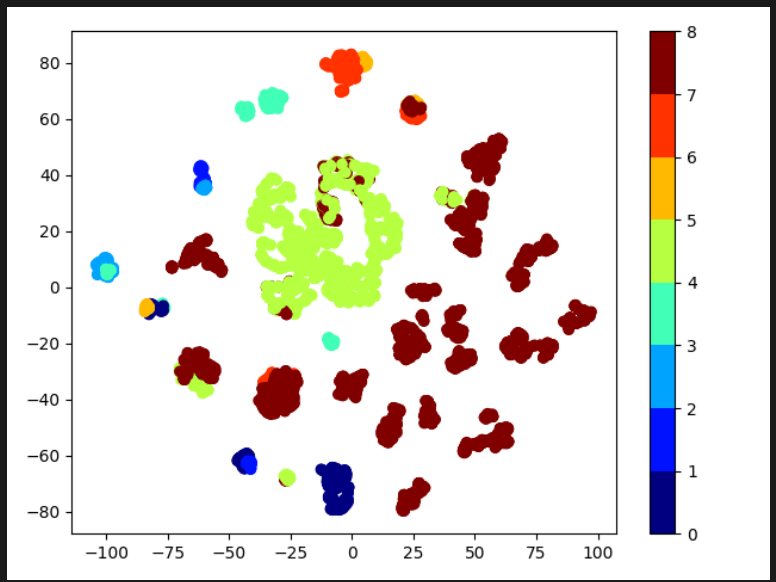
[5] Badia et al. 2020

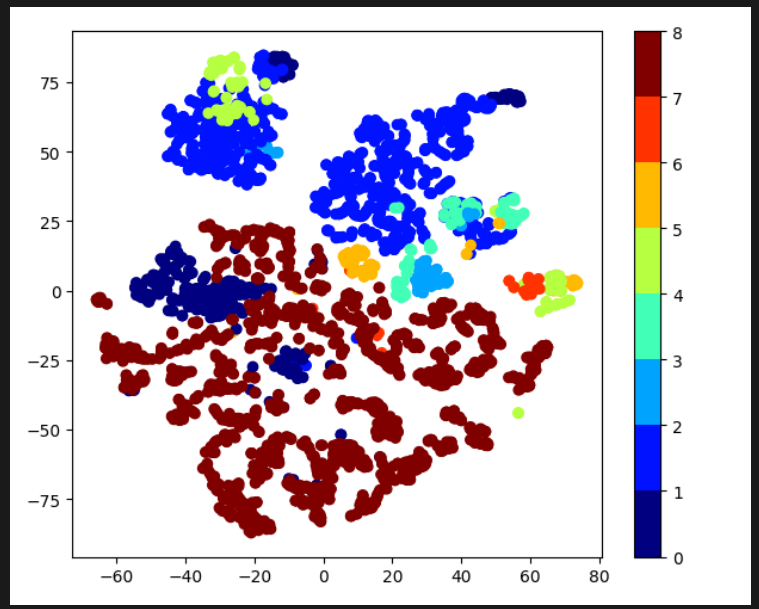
my current research

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

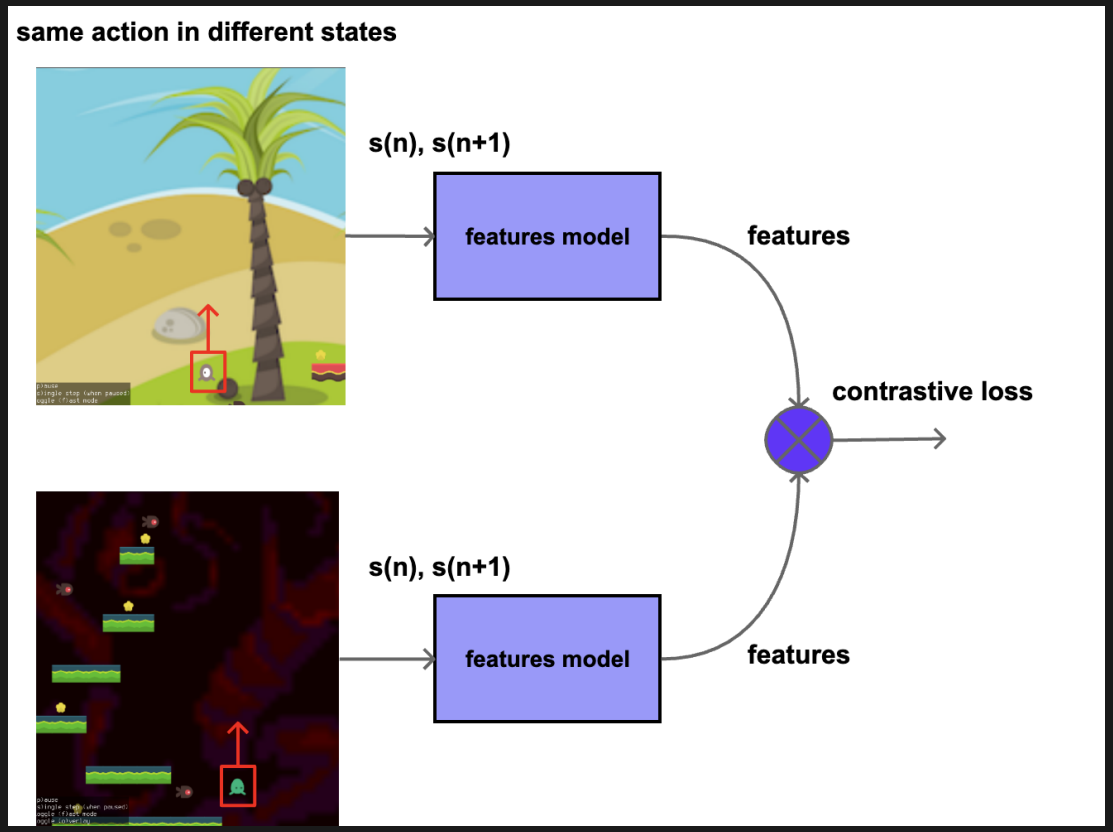
siamese network distillation







symmetry driven generalisation



$$\forall s(n), s(n+1) | a \in \mathbb{D} : f\Big(s(n), s(n+1); \theta\Big) = const_a$$

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