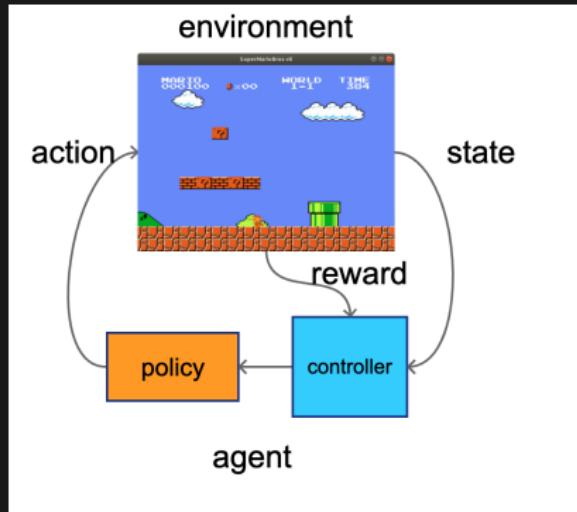
A painting depicting a traditional Aztec ceremony. In the center, a priest in elaborate feathered headdress and ceremonial attire stands on a platform, his arms raised in a gesture of offering or sacrifice. He holds a small object in one hand. Below him, a vast crowd of people, mostly men in traditional dress, 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

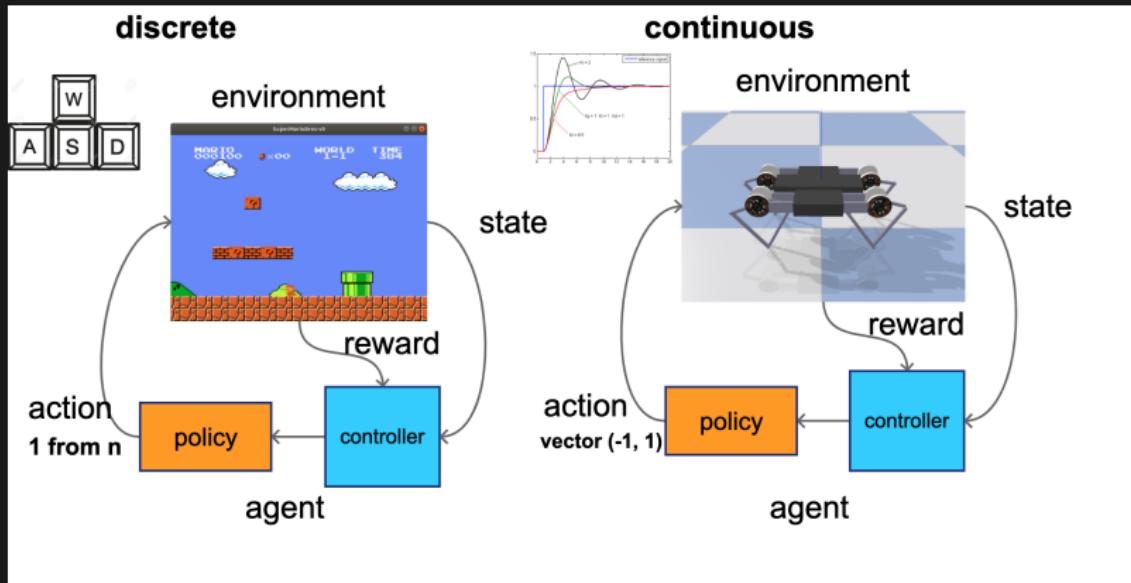
## Michal CHOVANEC, PhD.

# reinforcement learning



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

# 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**<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>

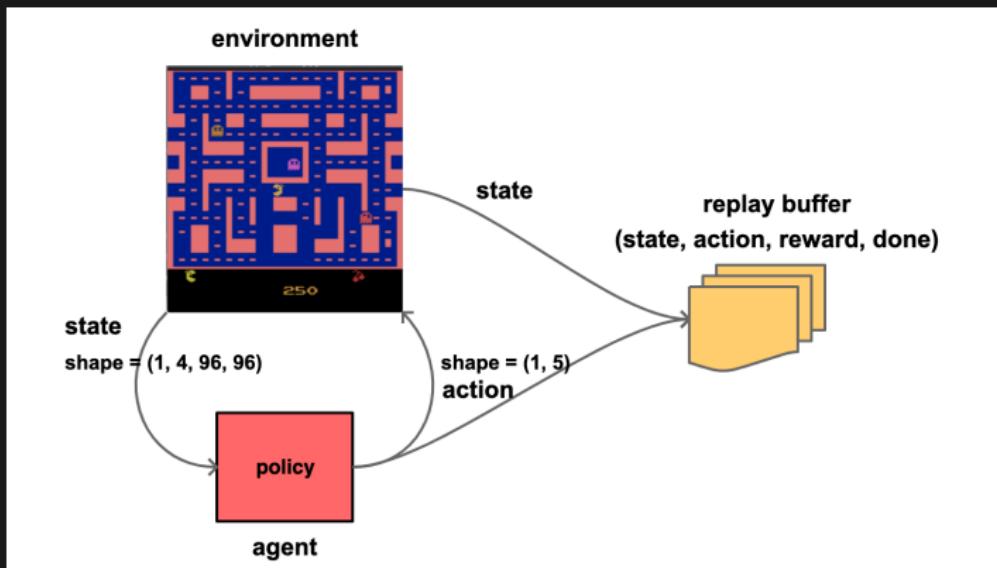
---

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

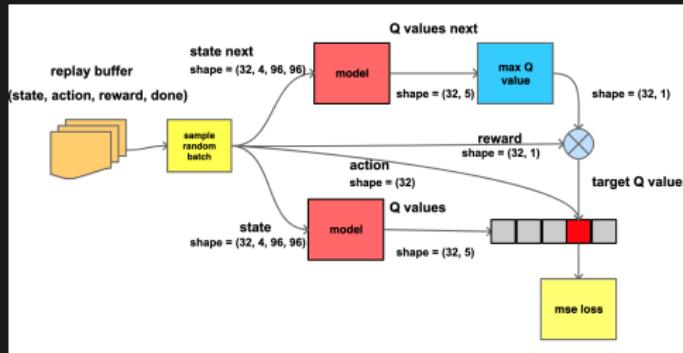
<sup>2</sup>Lillicrap, Hunt et al. 2016

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

# DQN - Playing Atari with Deep Reinforcement Learning



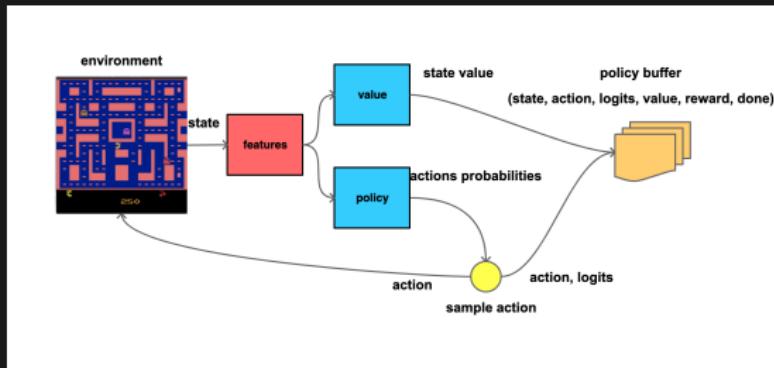
# DQN - Playing Atari with Deep Reinforcement Learning



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

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

# 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 <sup>a</sup>	0
2021	MuZero	2500
2018	Count-Based Exploration with Neural Density Models <sup>b</sup>	3705
<b>2019</b>	<b>Exploration by Random Network Distillation <sup>c</sup></b>	<b>8152</b>
2021	GoExplore* <sup>d</sup>	43 000

\* requires environment state saving/loading

---

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

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

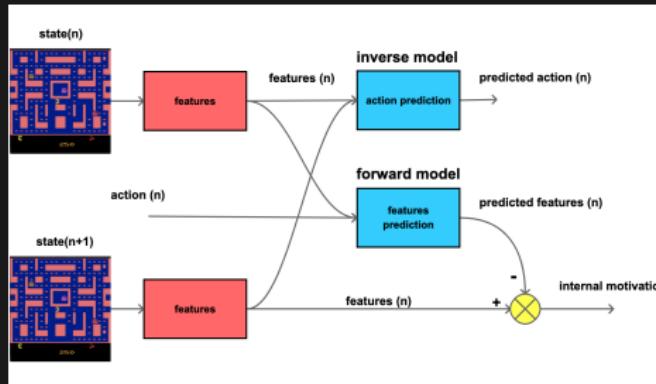
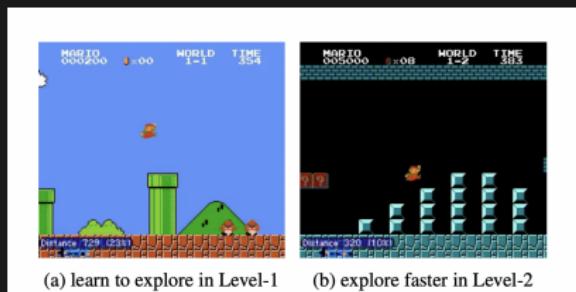
<sup>c</sup><https://arxiv.org/abs/1810.12894>

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

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

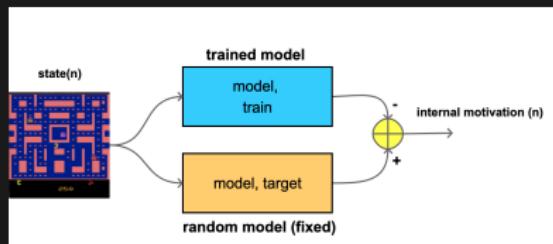
<sup>a</sup>Pathak et al. 2017

- 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

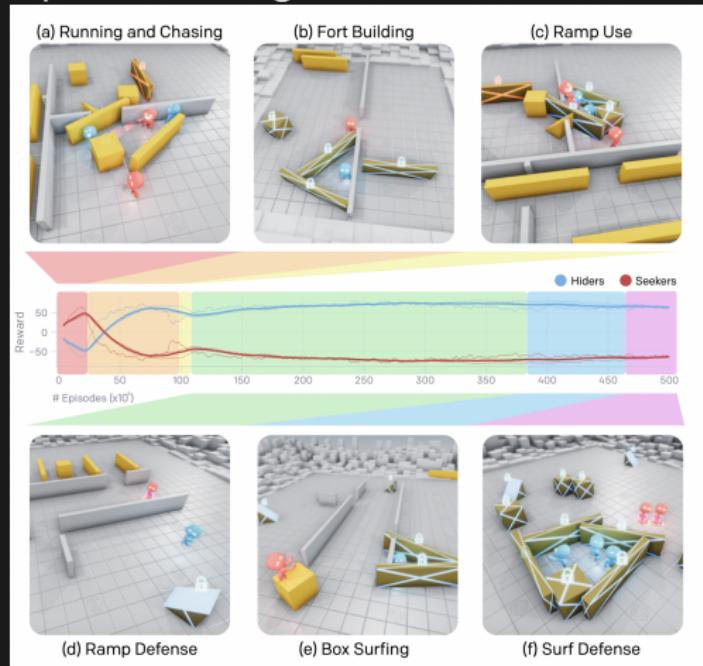


	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	<b>3,906</b>	<b>8,152</b>	-3	8,666	3,282	<b>1,859</b>
PPO	3,426	2,497	0	105	3,387	0
<b>Dynamics</b>	<b>3,371</b>	<b>400</b>	<b>0</b>	<b>33</b>	<b>3,246</b>	<b>1,712</b>
SOTA	2,209 <sup>1</sup>	3,700 <sup>2</sup>	<b>0</b>	<b>15,806<sup>2</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

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

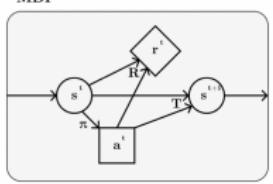
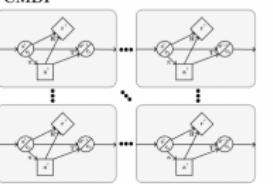
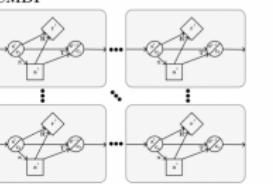
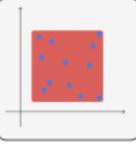
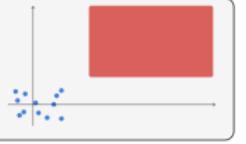
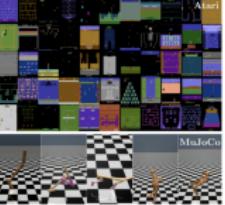
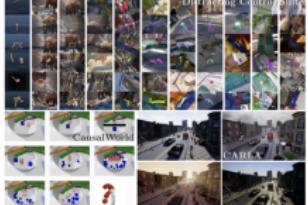
<sup>a</sup>Baker et al. 2020

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



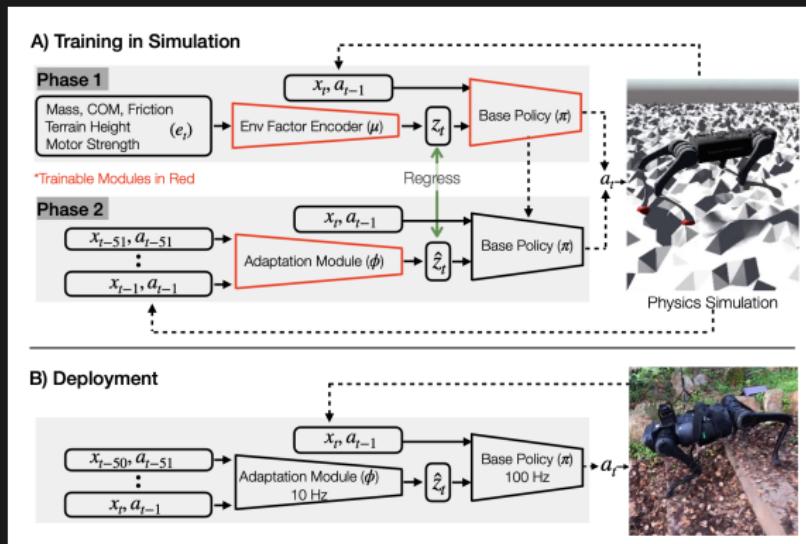
# generalisation <sup>a</sup>

<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 pointing 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 an action node $a^i$ . An arrow from $a^i$ points to $s^i$ .	 A diagram of a Causal Markov Decision Process (CMDP). It shows two parallel causal chains. The top chain starts with a state node $s^i$ , followed by a reward node $r^i$ , then a transition node $T^i$ , and finally the next state node $s^{i+1}$ . The bottom chain starts with a state node $s^j$ , followed by a reward node $r^j$ , then a transition node $T^j$ , and finally the next state node $s^{j+1}$ . Arrows indicate causal relationships between nodes in each chain.	 A diagram of a Causal Markov Decision Process (CMDP) under Out-of-Distribution (OOD) generalisation. It shows two parallel causal chains, similar to the IID case, but with different causal structures or distributions.
Train and Test Distribution	 A blue dot equals a red dot, indicating that Train = Test distribution.	 A scatter plot showing a uniform distribution of blue dots within a red rectangular boundary, representing Train Distribution = Test Distribution.	 A scatter plot showing blue dots in the lower-left and a large red rectangle in the upper-right, representing Train Distribution $\neq$ Test Distribution.
Example Benchmarks	 A grid of small images showing various environments: Alari, OpenAI Gym, and MuJoCo.	 A grid of small images showing environments from the Nothack Learning Environment, including OpenAI Gym, MuJoCo, and various 3D scenes.	 A grid of small images showing environments from the CAGLAD dataset, including Diving, Control, Snake, Casual World, 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>4</sup>  $4.5 * 10^9$  samples, score 8152 on MR
  - Never give up <sup>5</sup>  $3.5 * 10^{10}$  samples, score 10 000 on MR
  - SND  $1.28 * 10^8$  score 10 000 on MR

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

---

<sup>4</sup>Burda et al. 2018

<sup>5</sup>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

# 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