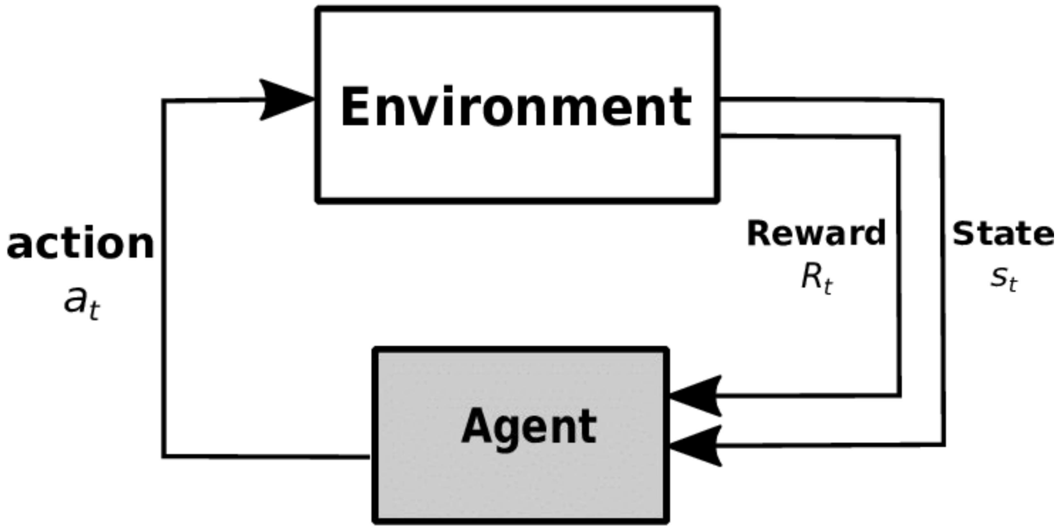
2017 Curiosity-driven Exploration by Self-supervised Prediction

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, Trevor Darrell

Reinforcement Learning Needs Reward

Reward in Real Life is Sparse



Curiosity

1. Curiosity helps an agent explore its environment for new knowledge
2. Curiosity helps an agent learn skills helpful in future scenarios

Two Challenges to Measure the Novelty

1. Difficult to build statistical model of predicting the next state based on current state in high-dimensional continuous state

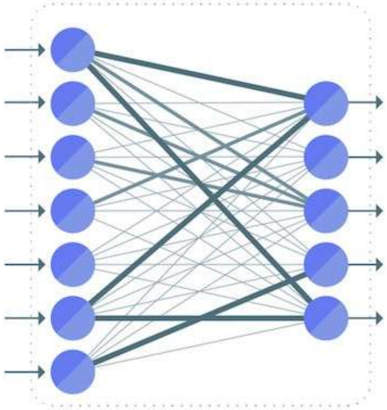
2. Inherent stochasticity in the environment and noise in the agent’s actuation

One solution is to only reward the agent when it encounters states that are hard to predict but learnable. Yet learnability is hard to determine.

Proposed Solution

Not making predictions in the raw sensory space

Transform the sensory input into a feature space where only the information relevant to the action performed by the agent is represented

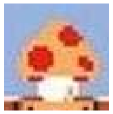


Why?

1. Predicting all the pixels is hard

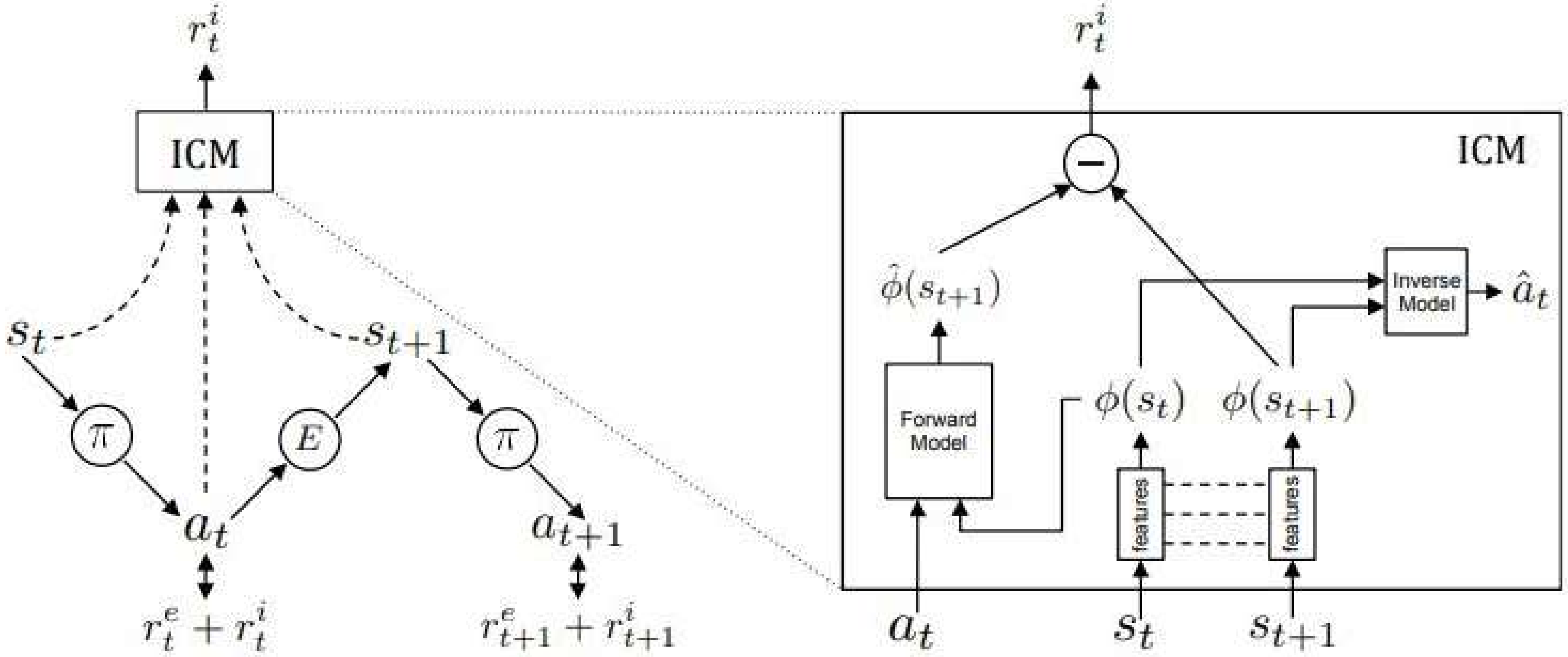
2. It is wrong

    Things controlled by the agent

    Thing out of control but affect agent 

    Things out of control and not affect agent

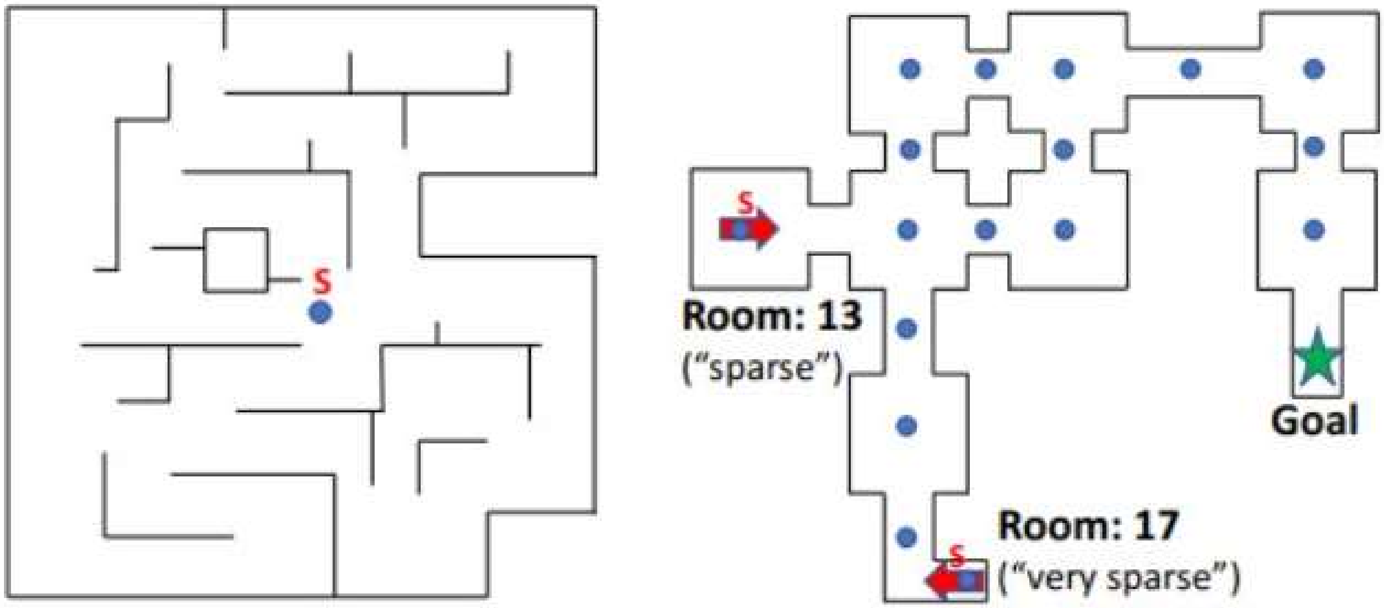
Intrinsic Curiosity Module (ICM)



Evaluation (VizDoom)

a 3-D navigation game with action space consisting of four actions: move forward, move right, move left, and no action. The game ends when the agent finds the vest in the 9 rooms connected by corridors.

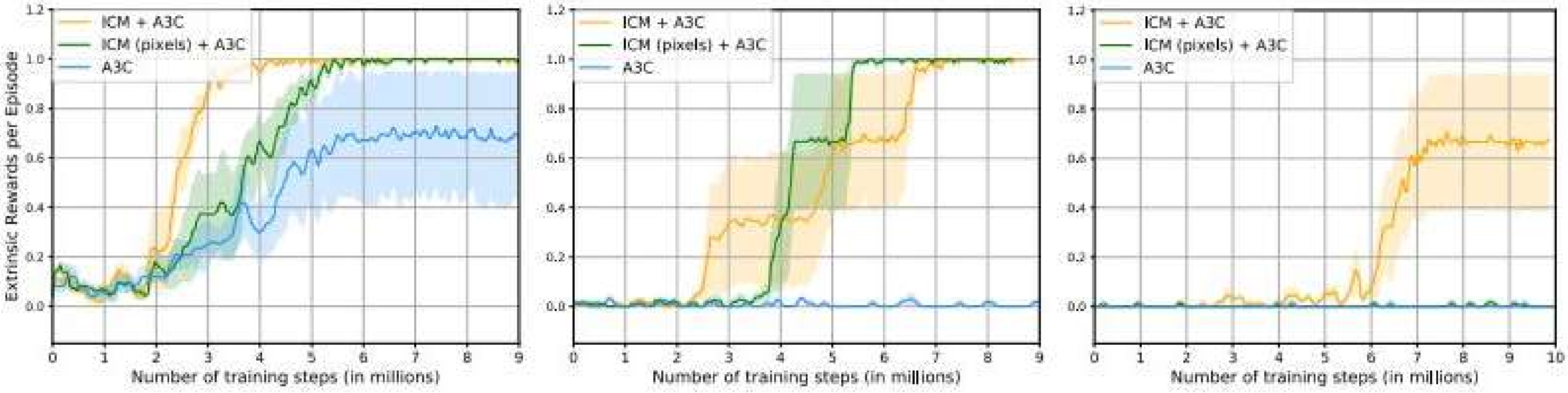
Three Setups



(a) Train Map Scenario

(b) Test Map Scenario

Result

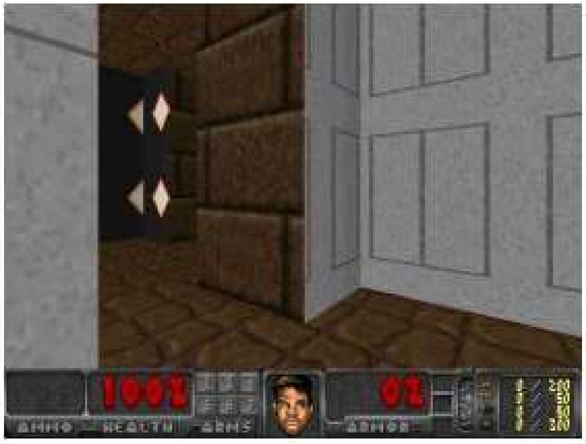


(a) "dense reward" setting

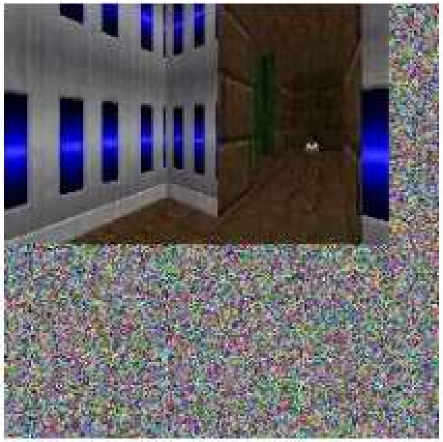
(b) "sparse reward" setting

(c) "very sparse reward" setting

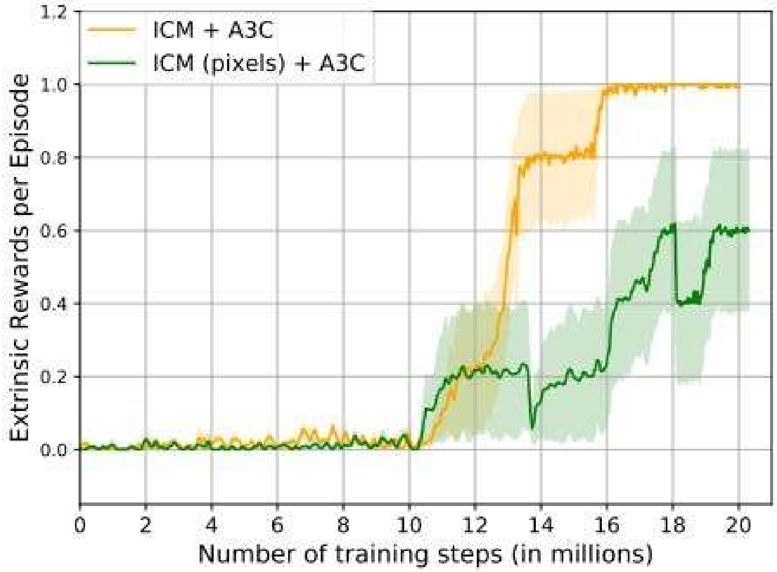
Adding Noise



(a) Input snapshot in VizDoom



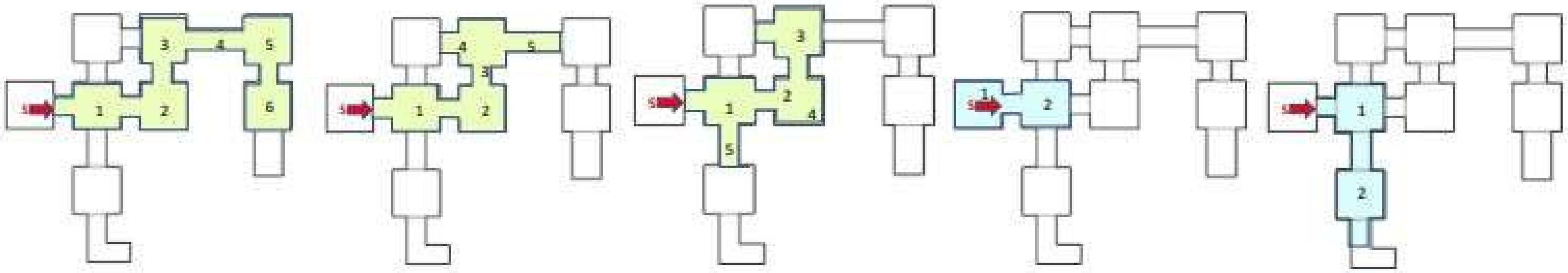
(b) Input w/ noise



No Reward

Blue: Without ICM

Yellow: ICM



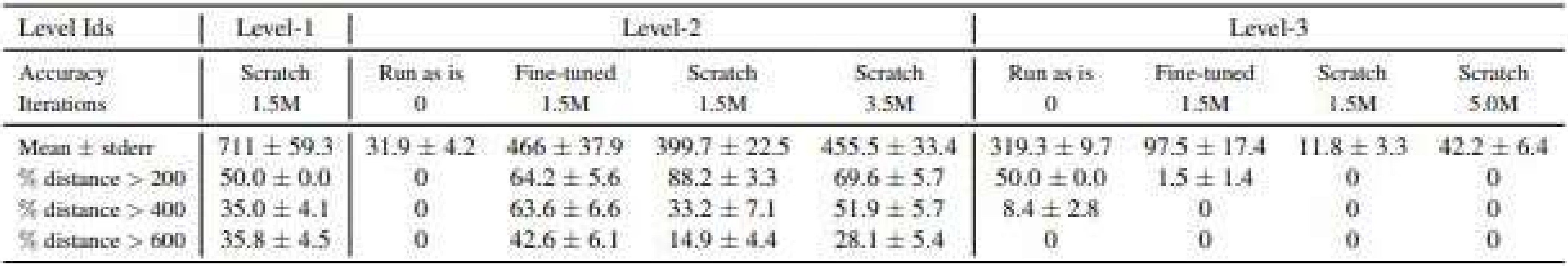
Evaluation (Mario)

1. Mario learn to cross over 30% of level 1
2. The agent receive no reward for dodging or killing enemy, but automatically discovered those behaviors
3. Because getting killed result seeing a small part of the game space.
4. To remain curious, the agents learn to avoid death.

Testing Generalization

1. Apply policy as it is to a new scenario
2. Adapt the policy by fine tuning with curiosity reward only
3. Adapt the policy to maximize some extrinsic rewards

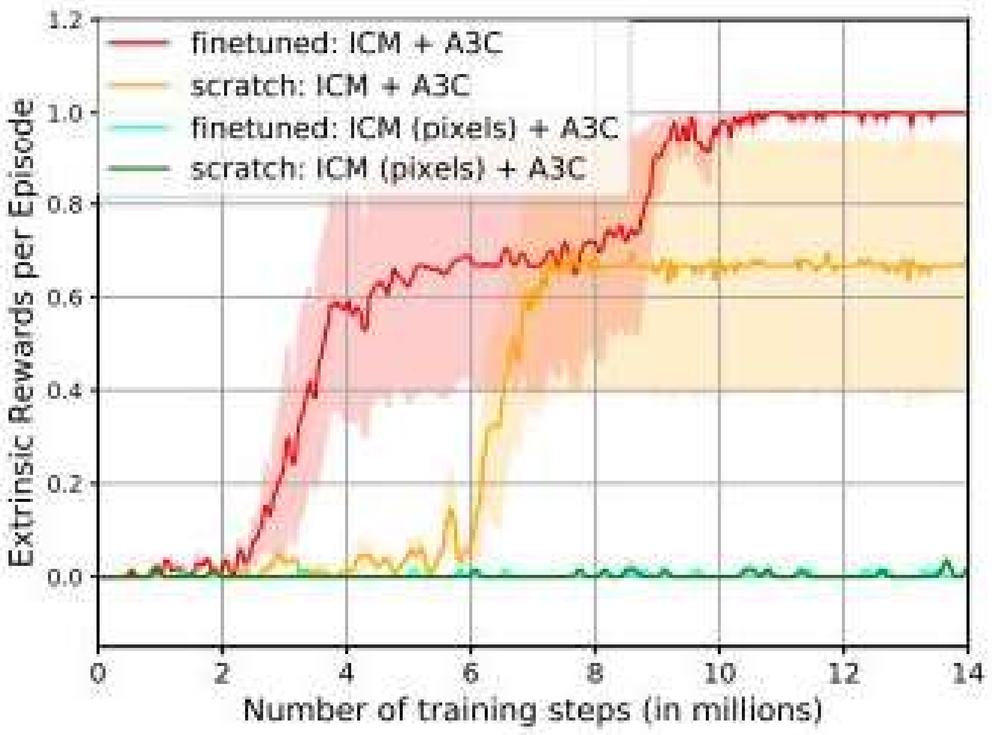
Result



Level Ids	Level-1	Level-2				Level-3			
Accuracy	Scratch	Run as is	Fine-tuned	Scratch	Scratch	Run as is	Fine-tuned	Scratch	Scratch
Iterations	1.5M	0	1.5M	1.5M	3.5M	0	1.5M	1.5M	5.0M
Mean ± stderr	711 ± 59.3	31.9 ± 4.2	466 ± 37.9	399.7 ± 22.5	455.5 ± 33.4	319.3 ± 9.7	97.5 ± 17.4	11.8 ± 3.3	42.2 ± 6.4
% distance > 200	50.0 ± 0.0	0	64.2 ± 5.6	88.2 ± 3.3	69.6 ± 5.7	50.0 ± 0.0	1.5 ± 1.4	0	0
% distance > 400	35.0 ± 4.1	0	63.6 ± 6.6	33.2 ± 7.1	51.9 ± 5.7	8.4 ± 2.8	0	0	0
% distance > 600	35.8 ± 4.5	0	42.6 ± 6.1	14.9 ± 4.4	28.1 ± 5.4	0	0	0	0

1. As is works for level 3 but not level 2
2. Finetune makes level 2 better
3. Level 3 is boring (curiosity blockade)

Result With Reward



Discussion

I wonder why the paper did the no reward settings experiment. The author claims that a good exploration policy is to allow the agent to visit as many states as possible even without any goals. I think the goal should contain 'arriving at the final flag' at least. In this case, some movements would not make sense or be not worthwhile exploring. E.g. you will never head to left when you born at the Mario world. From this view, I don't think it is valuable do no reward settings experiment.

Discussion

One thing that stuck out to me in this paper was the discussion about why pre training on level 1 then fine tuning on level 2 obtained better performance than training from scratch on level 2.

They seemed to suggest that the easier/friendlier environment of level 1 was more conducive to learning useful skills like jumping/moving. On top of that, they mention that the agent can only make it about 30% of the way through level 1 because there is a pit 38% of the way through.

I'm wondering if it could be useful to scale learning environment complexity as the agent learns. For example, present a “safer” situation where mario could learn the “long jump” behavior that would clear the pit without dying. Then once curiosity rewards saturate, add complexity to the environment. My worry would be that this would be cumbersome and involve a lot of human design across the environments, which sort of takes away from the idea of self-supervised learning in complex environments.