EPFL

Curiosity-driven exploration

Alireza Modirshanechi

Outline

Part 1. **Exploration bonus in tabular RL**

- Multi-Armed Bandits (MAB)
- Markov Decision Processes (MDP)

Part 2. **Curiosity-driven RL**

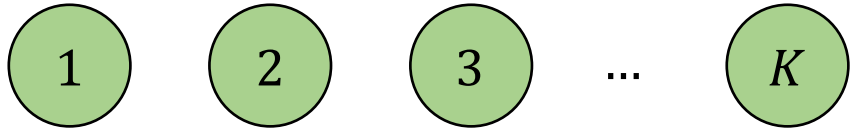
- Intelligent behavior in the absence of 'reward'
- Surprise, Novelty, and Information-gain in Deep RL
- Meta-learning of the reward function

Part 3. **Noisy TV problem**

- Being curious in the presence of stochasticity
- Noisy TV problem in curiosity driven Deep RL
- Over-optimism in humans and distraction by stochasticity.

Multi-Armed Bandits (MAB)

- We have K possible actions:



What to choose at time t?

- With true average reward:

$$\mu_i = E[r|a = i]$$

μ_1 μ_2 μ_3 ... μ_K

Optimal policy: $a_t = \arg\max_i \mu_i$

- Naïve estimates of averages:

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

$\hat{\mu}_1^{(t)}$ $\hat{\mu}_2^{(t)}$ $\hat{\mu}_3^{(t)}$ ... $\hat{\mu}_K^{(t)}$

$$T_i^{(t)} = \{\tau \leq t : a_\tau = i\}$$

Not optimal: $a_t = \arg\max_i \hat{\mu}_i^{(t)}$

Solutions based on random exploration:
- Epsilon-greedy
- Softmax
- Thompson sampling

- Comments for the previous slide:

- If we knew the exact average reward $\mu_i = E[r|a = i]$ of each arm, then the optimal solution would trivially be to choose the arm with highest average reward: $a_t = \arg\max_i \mu_i$

- A naïve approach is to estimate the average reward by the empirical averages and greedily choose the action with maximum estimated average reward: $a_t = \arg\max_i \hat{\mu}_i^{(t)}$

- The naïve greedy policy is prone to fail in finding the best action.

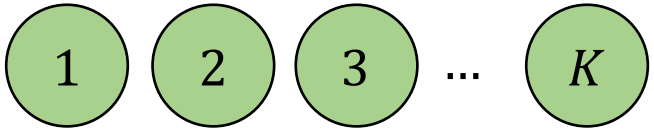
- You have seen epsilon-greedy and the softmax policy as two approaches for dealing with this problem by adding randomness to the action-selection. Another approach to exploration is Thompson sampling (Thompson 1933 in Biometrika). We do not discuss Thompson sampling in this lecture.

- Our focus will be on "directed exploration" by using exploration bonuses.

How to evaluate an exploratory policy?

- MAB with K possible actions:

$\mu_i = E[r|a = i]$ Highest reward rate: $\mu^* = \max_i \mu_i$



- "Regret" of algorithm A (ϵ-greedy):

$$R_A(T) = E_A\left[\sum_{t=1}^{T} \mu^* - \mu_{a_t}\right]$$

The best you could choose

What you chose

- Consistent algorithms:

$$\lim_{T\to\infty} \frac{R_A(T)}{T} = 0 \quad \Longrightarrow \quad \lim_{T\to\infty} \frac{E_A[\sum_{t=1}^{T} \mu_{a_t}]}{T} = \mu^*$$

- **Theorem 1 of Lai and Robbins 1985:**

  Under specific conditions, if algorithm A is consistent, then, loosely speaking, $R_A(T)$ is at least proportional to $\log T$.

a loose notion of optimality

- Comments for the previous slide:

- Before discussing how differently one can deal with exploration-exploitation dilemma, we discuss a common method for evaluating different algorithms in multi-armed bandits.

- A key notion to evaluate an algorithm A is regret $R_A(T)$ measuring the expected difference between the choices of the algorithm and the best possible actions, summed over the first T steps.

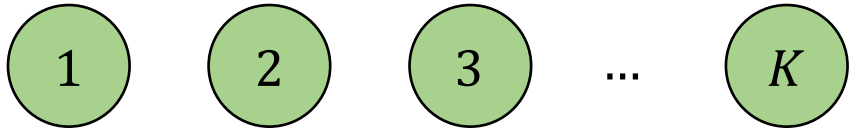
- An algorithm is called consistent, if its average regret $\frac{R_A(T)}{T}$ vanishes over time.

- It is proven (under certain conditions; see Lai and Robbins 1985 in Advances in Applied Mathematics) that the regret of a consistent algorithm scales at least logarithmically with time T.

- This introduces a loose notion of optimality: An optimal algorithm is a consistent algorithm whose regret scales logarithmically with time T.

An example of optimal algorithms

- MAB with K possible actions:



- Reminder: greedy algorithm

$$\hat{\mu}_i^{(t)} = \frac{\sum_{\tau \in T_i^{(t)}} r_\tau}{\left|T_i^{(t)}\right|}$$

$\hat{\mu}_1^{(t)}$ $\hat{\mu}_2^{(t)}$ $\hat{\mu}_3^{(t)}$ ... $\hat{\mu}_K^{(t)}$

$$a_t = \arg\max_i \hat{\mu}_i^{(t)}$$

- Upper Confidence Bound (UCB1 in Auer et al. 2002):

$$U_i^{(t)} = \hat{\mu}_i^{(t)} + \sqrt{\frac{2\log t}{\left|T_i^{(t)}\right|}}$$

The naïve estimate of average reward

Bonus for exploration (the same as what you saw for Monte Carlo Tree Search)

$U_1^{(t)}$ $U_2^{(t)}$ $U_3^{(t)}$ ... $U_K^{(t)}$

$$a_t = \arg\max_i U_i^{(t)}$$

Theorem 1 of Auer et al. 2002:
$R_{\mathrm{UCB1}}(T) \propto \log T + \mathrm{const.}$

- Comments for the previous slide:

- A smart optimal algorithm is Upper Confidence Bound (UCB; proposed by Auer et al. 2002 in Machine Learning) that computes a confidence bound index $U_i^{(t)}$ for each action and chooses the one with highest index.

- The index is equal to the naïve estimate average reward $\hat{\mu}_i^{(t)}$ plus an exploration bonus that is (i) a decreasing function of how many times an arm has been chosen $\left|T_i^{(t)}\right|$ but (ii) an increasing function of how many actions have been taken in total (i.e. t).

- The regret for the UCB algorithm scales logarithmically with T, hence it is an "optimal" algorithm. The constants of the regret can be fine-tuned by some variations of the algorithm (see Auer et al. 2002).

Outline

Part 1. **Exploration bonus in tabular RL**

✓- Multi-Armed Bandits (MAB)

- Markov Decision Processes (MDP)

Part 2. **Curiosity-driven RL**

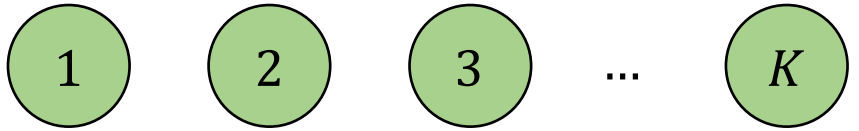
- Intelligent behavior in the absence of 'reward'
- Surprise, Novelty, and Information-gain in Deep RL
- Meta-learning of the reward function

Part 3. **Noisy TV problem**

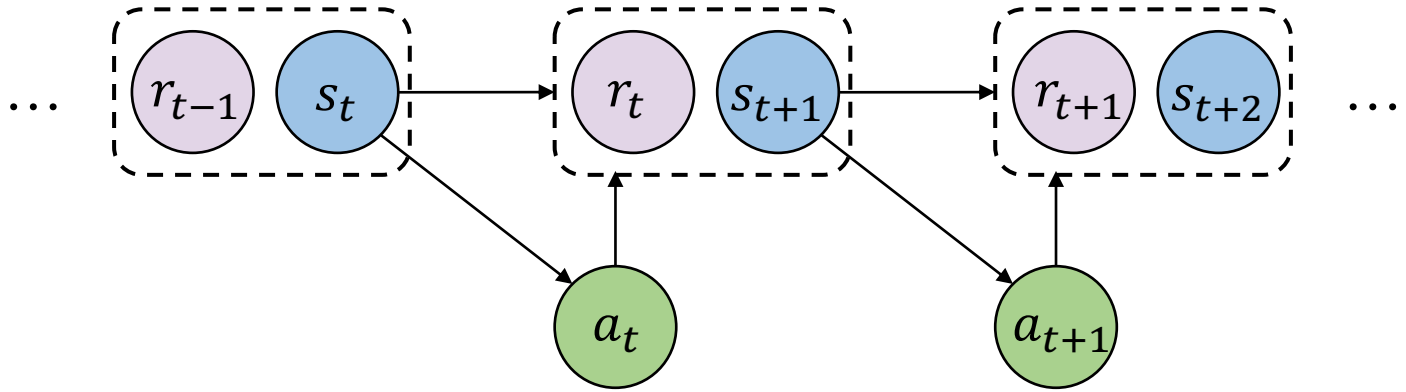
- Being curious in the presence of stochasticity
- Noisy TV problem in curiosity driven Deep RL
- Over-optimism in humans and distraction by stochasticity.

Beyond MAB

- MAB with K possible actions:



- Markov Decision Processes (MDP):
  - P: transition probabilities, e.g. $P(s'|s,a)$
  - R: expected reward, e.g. $R(s,a)$



Exploration bonus in MDPs

- Dynamic programming with true $P(s'|s,a)$ and $R(s,a)$:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a') \qquad a_t = \arg\max_a Q^*(s_t,a)$$

- Naïve model-based (MB) RL:

$$\hat{Q}^{(t)}_{\mathrm{MB}}(s,a) = \hat{R}^{(t)}(s,a) + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}^{(t)}_{\mathrm{MB}}(s',a')$$

$$\hat{R}^{(t)}(s,a) = \frac{\sum_{\tau \in T^{(t)}_{s,a}} r_\tau}{\left|T^{(t)}_{s,a}\right|} \qquad \hat{P}^{(t)}(s'|s,a) = \frac{\left|T^{(t)}_{s,a,s'}\right|}{\left|T^{(t)}_{s,a}\right|}$$

$$a_t = \arg\max_a \hat{Q}^{(t)}_{\mathrm{MB}}(s_t,a)$$

The exploration-exploitation trade-off is even more serious in MDPs than MABs.

Any trick similar to UCB?

$$T^{(t)}_{s,a} = \{\tau \le t : a_\tau = a, s_\tau = s\} \qquad T^{(t)}_{s,a,s'} = \{\tau \le t : a_\tau = a, s_\tau = s, s_{\tau+1} = s'\}$$

- Comments for the previous slide:

- Similar to the bandit setting, if we have access to the true transition probabilities and reward functions, then the optimal policy would be to use Dynamic Programming, solve the optimal Bellman equations, and use a greedy policy on the resulting Q-values: $a_t = \arg\max_a Q^*(s_t, a)$

- In the absence of the complete knowledge of the environment, a naïve model-based approach is to approximate the transition probabilities and the reward values, solve the optimal Bellman equations by using these estimates, and use a greedy policy on the resulting Q-values: $a_t = \arg\max_a \hat{Q}^{(t)}_{\mathrm{MB}}(s_t, a)$

- The naïve model-based approach is prone to be stuck in some parts of the environment and never find the optimal policy. You have seen epsilon-greedy and the softmax policy as to approaches to deal with this issue by adding randomness to the action-selection. Here, we ask whether we can find a directed exploration approach like UCB for MDPs.

MBIE+EB (Strehl and Littman 2008)

- Dynamic programming with true $P(s'|s,a)$ and $R(s,a)$:

$$Q^*(s,a) = R(s,a) + \gamma \sum_{s'} P(s'|s,a) \max_{a'} Q^*(s',a')$$

- Naïve model-based (MB) RL:

$$\hat{Q}_{\mathrm{MB}}^{(t)}(s,a) = \hat{R}^{(t)}(s,a) + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}_{\mathrm{MB}}^{(t)}(s',a')$$

- Model-based interval estimation with exploration bonus (MBIE+EB in Strehl and Littman 2008):

$$\hat{Q}_{\mathrm{MB}}^{(t)}(s,a) = \hat{R}^{(t)}(s,a) + \frac{\beta}{\sqrt{T_{s,a}^{(t)}}} + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}_{\mathrm{MB}}^{(t)}(s',a')$$

The naïve estimate of average reward

Bonus for exploration (different from UCB regarding $\log t$)

- Comments for the previous slide:

- Model-based interval estimation with exploration bonus (MBIE+EB; proposed by Strehl and Littman 2008 in the Journal of Computer and System Sciences) uses the exact same procedure as the naïve model-based approach except that it adds an exploration bonus to the reward function.

- The exploration bonus is a decreasing function of how many times a specific action is taken in a specific state, so it motivates taken actions that have been taken less often.

Adding the exploration bonus is "good"

- Model-based interval estimation with exploration bonus (MBIE+EB in Strehl and Littman 2008):

$$\hat{Q}_{\mathrm{MB}}^{(t)}(s,a) = \hat{R}^{(t)}(s,a) + \frac{\beta}{\sqrt{T_{s,a}^{(t)}}} + \gamma \sum_{s'} \hat{P}^{(t)}(s'|s,a) \max_{a'} \hat{Q}_{\mathrm{MB}}^{(t)}(s',a')$$

- Theorem 2 in Strehl and Littman 2008:

  MBIE+EB is Probably Approximately Correct for MDPs (= it is **PAC-MDP**).

  = loosely speaking, its choices are **good enough** with **high probability**.

- Alternative: Bayesian Exploration Bonus (BEB) by Kolter and Ng 2009

$$\text{Bonus} = \frac{\beta}{1+T_{s,a}^{(t)}}$$

It is not PAC-MDP but is near-Bayesian.

Theorem 2. Exploration based on a bonus proportional to $\left(T_{s,a}^{(t)}\right)^{-p}$ is not PAC-MDP if $p > 0.5$.

- Comments for the previous slide:

- MBIE+EB is proven to be PAC-MDP (see Strehl and Littman 2008): In short and loosely speaking, this means that, with high probability, most of the actions take by MBIE+EB are close to the actions that would have been taken by the optimal policy.

- Alternative exploration bonuses are possible, but they have different properties. For example, an exploration bonus proportional to one over $T_{s,a}^{(t)}$ is not PAC-MDP but is "near Bayesian" (i.e., another notion of optimality; see Kolter and Ng in ICML 2009).

Part 1: Quiz

?

- A consistent learning algorithm eventually achieves a zero *average* regret in Multi-Armed Bandits (MAB).

?

- An optimal algorithm in MABs achieves a constant *total* regret.

?

- $\frac{\beta}{\sqrt{T_{s,a}^{(t)}}}$ is always better exploration bonus for MDPs than $\frac{\beta}{T_{s,a}^{(t)}}$.

Summary

- Adding exploration bonus provably improves the performance of RL algorithms.
- Hence, to optimally seek a reward signal, one may benefit from seeking a modification of that reward signal.
- There is, however, not a single approach to
    - define a exploration bonus
    - evaluate its performance.

Outline

Part 1. **Exploration bonus in tabular RL**

✓- Multi-Armed Bandits (MAB)

✓- Markov Decision Processes (MDP)

Part 2. **Curiosity-driven RL**

- Intelligent behavior in the absence of 'reward'
- Surprise, Novelty, and Information-gain in Deep RL
- Meta-learning of the reward function

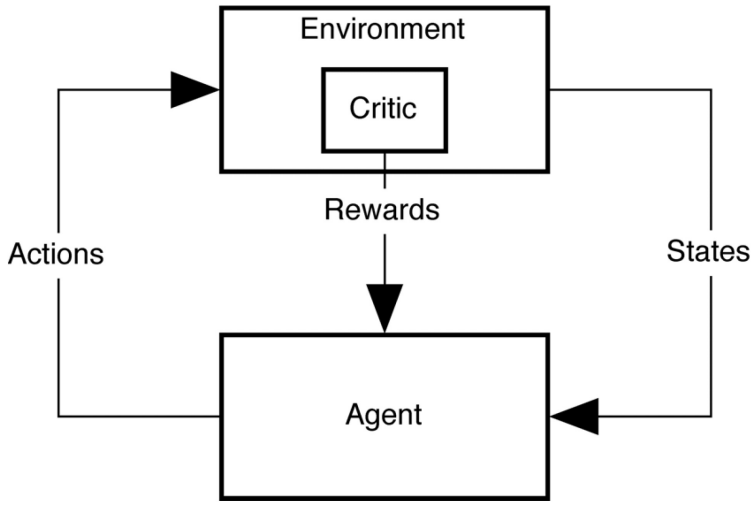
Part 3. **Noisy TV problem**

- Being curious in the presence of stochasticity
- Noisy TV problem in curiosity driven Deep RL
- Over-optimism in humans and distraction by stochasticity.

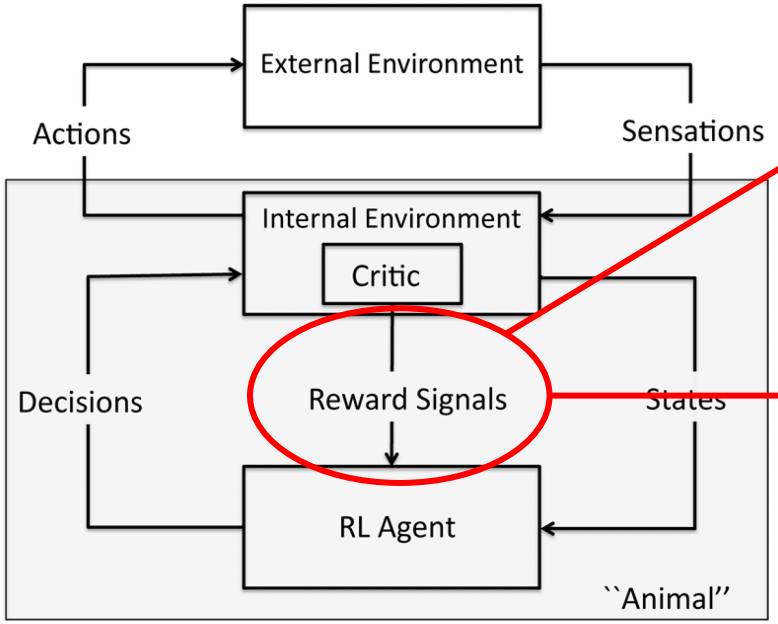
Intelligent behavior in the absence of 'reward'

- Intrinsically motivated RL (Singh et al. 2010)

- In the traditional RL models, rewards are always "external":



- Alternative: Reward signal is essentially internal



Extrinsic component:
- Nutrition
- Money
- etc.

Intrinsic component:
- Information/Knowledge
- Surprise
- Novelty
- etc.

What results in curiosity-driven behavior

- Comments for the previous slide:

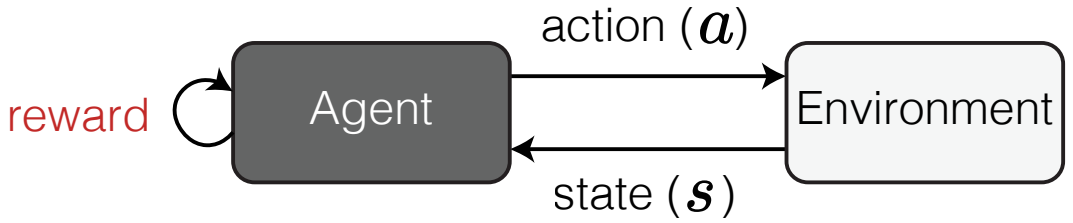
- In the traditional view on RL, the reward signal is always external and is given to the agent by the environment.

- However, humans and animals receives only sensory inputs from their environment, and whether these inputs feel rewarding are judged internally. Based on this argument, an alternative view on RL has been proposed to consider the reward signal generated internally by the agent.

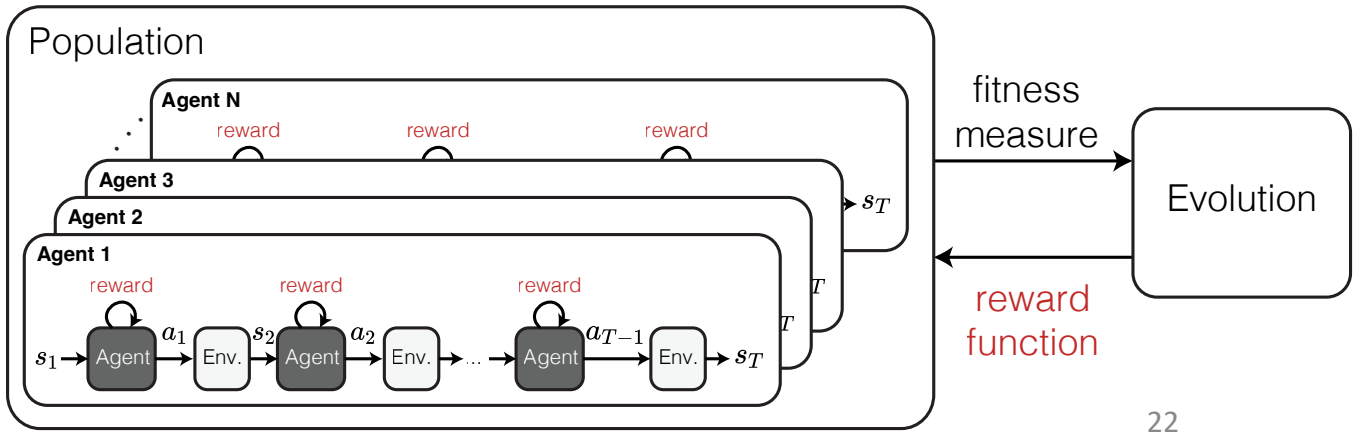
- The internally generated reward signal consists of an extrinsic and an intrinsic component. Typical examples for the intrinsic component of the reward signal are novelty, surprise, information, etc. This component is believed to drive curiosity-driven behavior in humans and animals.

Two approaches to modeling curiosity

- How should we characterize the intrinsic reward function?
- Bottom-up views: Starting from **"What are we curious about?"**



- Top-down views: Starting from **"Why are we curious?"**



[Figures from Modirshanechi et al. (in preparation)]

- Comments for the previous slide:

- There is an open question on how to characterize the internal reward signal to either have the best performance in a specific (set of) task(s) (in machine learning) or have the best model of curiosity-driven behavior in humans and animals (in neuroscience, psychology, and cognitive science).

- Attempts to address this question can be classified into two categories. Bottom-up approaches start with the question of "What are we curious about?" and define the internal reward based on some heuristic reasoning (e.g., it is good to seek novelty if you want to explore all states in an environment).

- Top-down approaches start with the question of "Why are we curious?" and define reward function as the evolutionary solution to the optimization of a fitness measure (e.g., survival rate). See Singh et al. 2010 in IEEE Transactions on Autonomous Mental Development for more details and discussions.

Different intrinsic reward signals in bottom-up views

- State novelty:

$$N^{(t)}_{s_t,a_t \to s_{t+1}} = -\log P^{(t)}_N(s_{t+1})$$ or, e.g., $\frac{1}{T^{(t)}_{s_t}}$.

- Transition surprise:

$$S^{(t)}_{s_t,a_t \to s_{t+1}} = -\log P^{(t)}(s_{t+1}|s_t, a_t)$$

- Information gain or progress rate:

$$IG^{(t)}_{s_t,a_t \to s_{t+1}} = d\left(P^{(t)}(.|s_t, a_t), P^{(t+1)}(.|s_t, a_t)\right)$$

Including classic bonuses, e.g., $\frac{1}{T^{(t)}_{s_t,a_t}}$.

- Others, e.g., empowerment (Klyubin et al. 2005).

- The main challenge in deep RL:

Very high dimensional state spaces…

- There has been TONS of work on defining, finetuning, and using intrinsic rewards in Deep RL.

- Today, we cover 3 examples.

You can see these reviews for many more:
- Aubret et al. 2019 and 2022 on arXiv
- Ladosz et al. 2022 in Information Fusion

- Comments for the previous slide:

- In the bottom-up approach, there are many different choices of intrinsic rewards. We focus on three classes:
  1. State novelty drives agents to explore the least visited parts of the environment.
  2. Surprise drives agents to explore parts of the least predictable parts of the environment.
  3. Information-gain or progress rate drives agents to parts of the environment where they have highest rate in learning/improving their model of the environment.

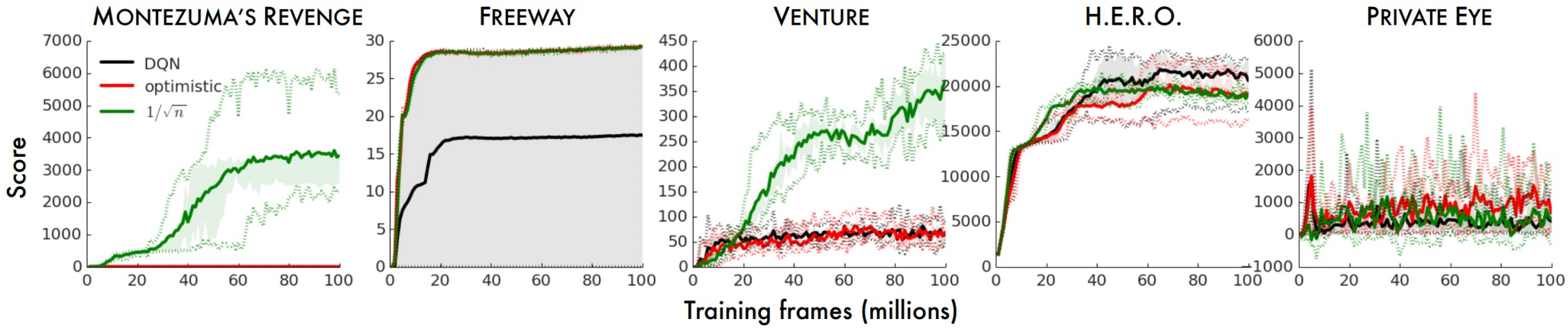
- The information-gain and progress rate usually behave similarly to the classic exploration bonuses and are decreasing functions of $T_{s_t,a_t}^{(t)}$.

- The challenge of implementing these methods in Deep RL is the very high dimensional state spaces in the problems of Deep RL. The simple counts do not have any meaning in such spaces.

- There has been TONS of work on defining, finetuning, and using intrinsic rewards in Deep RL, but we focus only on three examples of these methods.

Example 1: Novelty-seeking in Deep RL

- Bellemare et al. in NeurIPS 2016 (> 1'000 citations):
  - A parameterized model to define and learn $P_N^{(t)}(s_{t+1})$ on the pixel space.
  - Define the pseudo-count $\hat{T}_{s_{t+1}}^{(t)}$ based on $P_N^{(t)}(s_{t+1})$ (**beautiful theory**).
  - Double DQN with $r_{t+1} = r_{t+1}^e + \frac{0.05}{\sqrt{\hat{T}_{s_{t+1}}^{(t)} + 0.01}}$



- Comments for the previous slide:

- A classic example of implementing novelty-seeking in Deep RL considers a parameterized distribution for defining and learning the state frequency (= called "density model" in the paper).

- The state frequency is used to define pseudo counts (that considers similarities between different states) which are then used for exploration.

- Description of the methodology for the empirical results (copied from the original paper):

6.1 Exploration in Hard Atari 2600 Games

From 60 games available through the Arcade Learning Environment we selected five "hard" games, in the sense that an ϵ-greedy policy is inefficient at exploring them. We used a bonus of the form

$$R_n^+(x,a) := \beta(\hat{N}_n(x) + 0.01)^{-1/2}, \tag{4}$$

where $\beta = 0.05$ was selected from a coarse parameter sweep. We also compared our method to the optimistic initialization trick proposed by Machado et al. (2015). We trained our agents' Q-functions with Double DQN (van Hasselt et al., 2016), with one important modification: we mixed the Double Q-Learning target with the Monte Carlo return. This modification led to improved results both with and without exploration bonuses (details in the appendix).

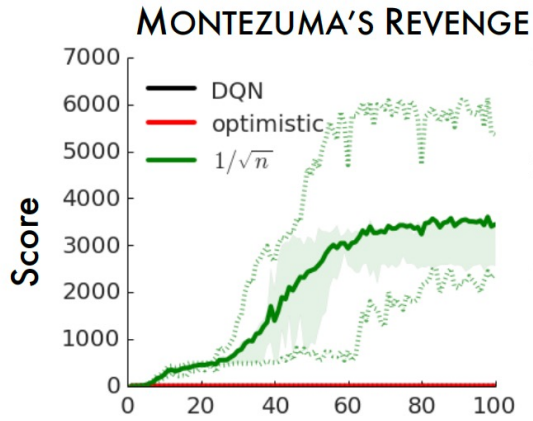
- Figure caption (copied from the original paper):

Figure 2: Average training score with and without exploration bonus or optimistic initialization in 5 Atari 2600 games. Shaded areas denote inter-quartile range, dotted lines show min/max scores.

Example 1: Novelty-seeking in Deep RL

- Bellemare et al. in NeurIPS 2016 (> 1'000 citations):

- Novelty-seeking
  - enables efficient exploration and
  - results in higher extrinsic rewards



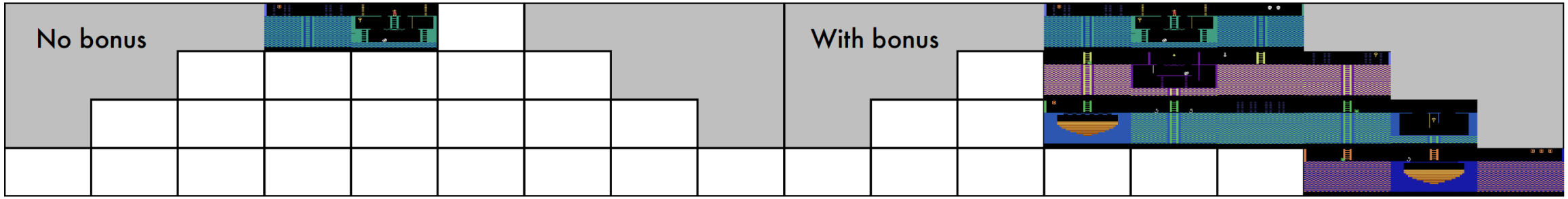


Figure 3: "Known world" of a DQN agent trained for 50 million frames with (**right**) and without (**left**) count-based exploration bonuses, in MONTEZUMA'S REVENGE.

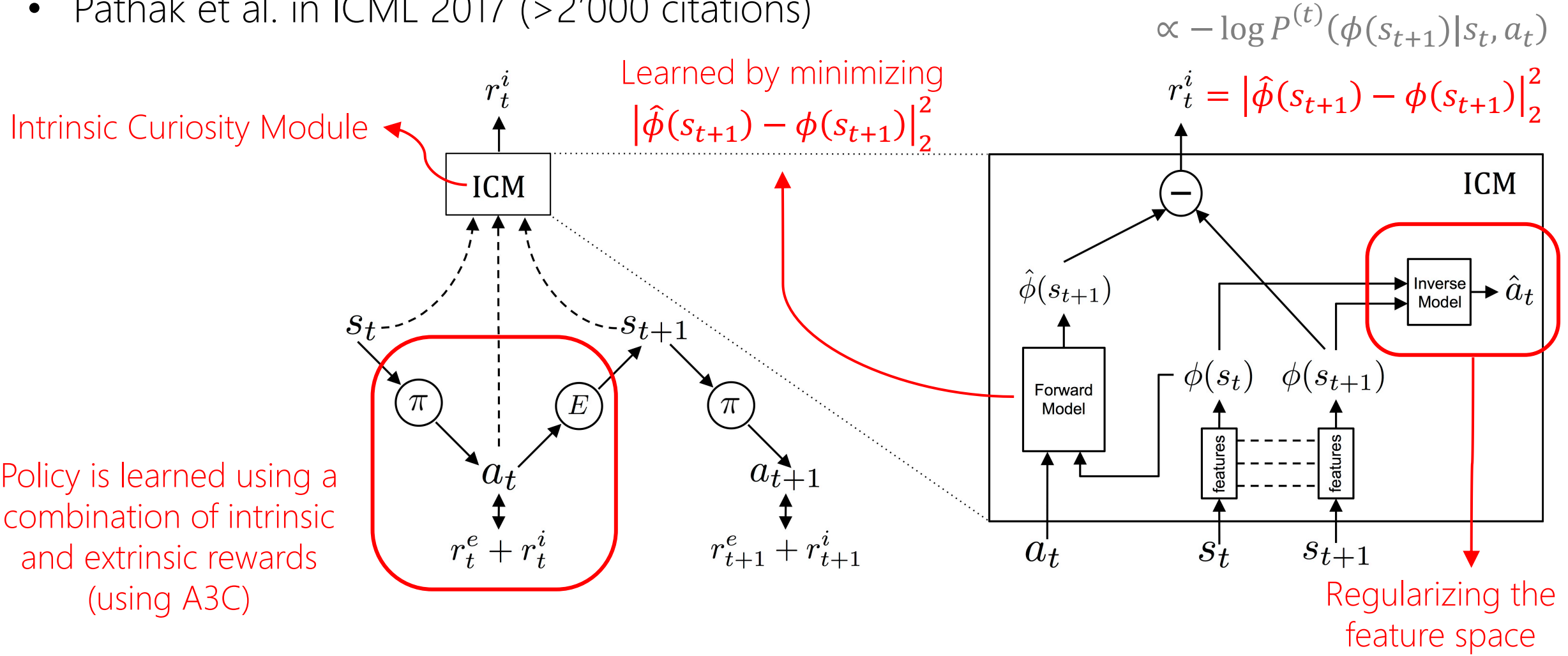
- Comments for the previous slide:

- Description of the figure in the text:

> MONTEZUMA'S REVENGE is perhaps the hardest Atari 2600 game available through the ALE. The game is infamous for its hostile, unforgiving environment: the agent must navigate a number of different rooms, each filled with traps. Due to its sparse reward function, most published agents achieve an average score close to zero and completely fail to explore most of the 24 rooms that constitute the first level (Figure 3, top). By contrast, within 50 million frames our agent learns a policy which consistently navigates through 15 rooms (Figure 3, bottom). Our agent also achieves a score higher than anything previously reported, with one run consistently achieving 6600 points by 100 million frames (half the training samples used by Mnih et al. (2015)). We believe the success of our method in this game is a strong indicator of the usefulness of pseudo-counts for exploration.[1]

Example 2: Surprise-seeking in Deep RL

- Pathak et al. in ICML 2017 (>2'000 citations)



- Comments for the previous slide:

- Figure caption (copied from the original paper):

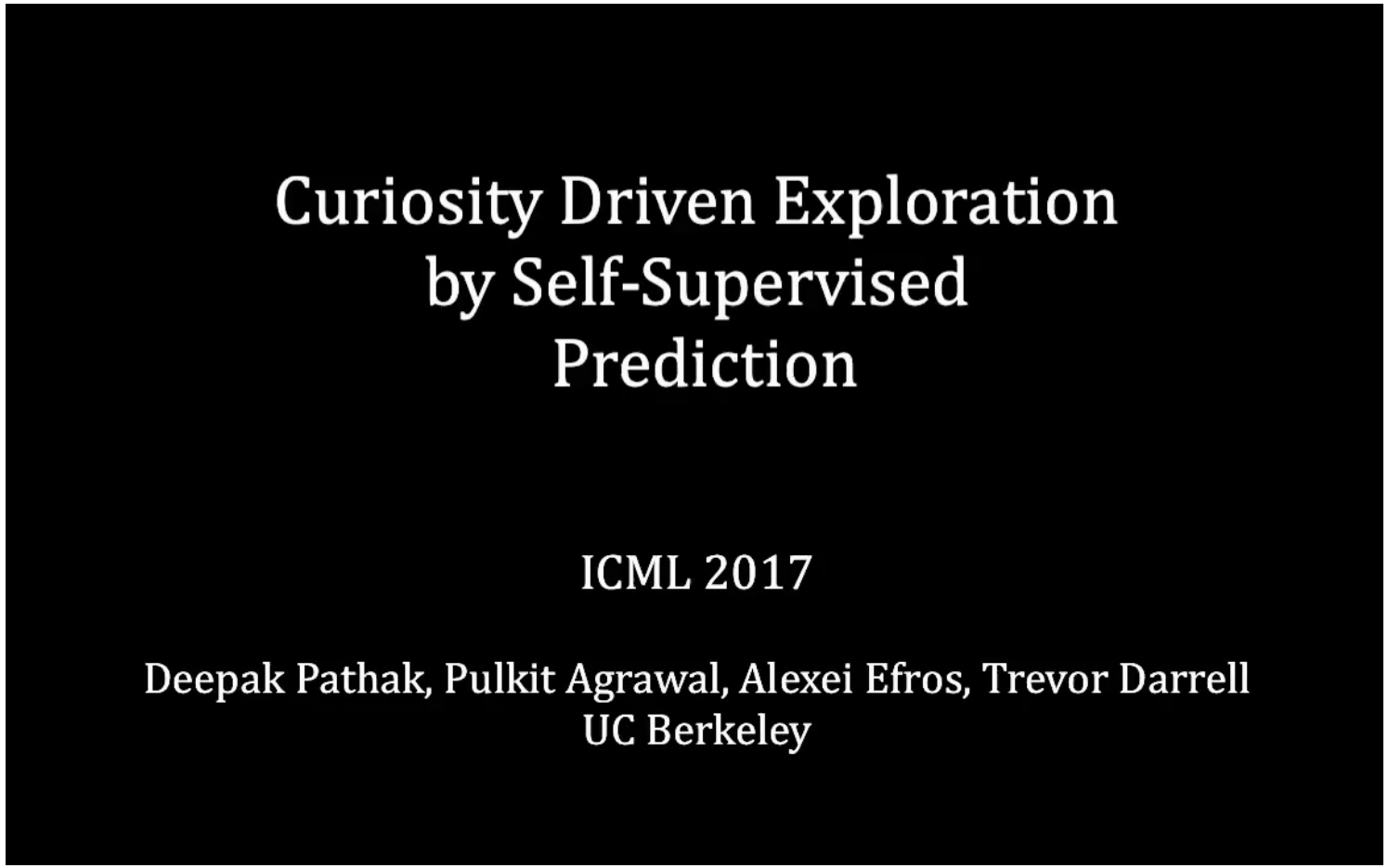
> *Figure 2.* The agent in state s_t interacts with the environment by executing an action a_t sampled from its current policy π and ends up in the state s_{t+1}. The policy π is trained to optimize the sum of the extrinsic reward (r_t^e) provided by the environment E and the curiosity based intrinsic reward signal (r_t^i) generated by our proposed Intrinsic Curiosity Module (ICM). ICM encodes the states s_t, s_{t+1} into the features $\phi(s_t), \phi(s_{t+1})$ that are trained to predict a_t (i.e. inverse dynamics model). The forward model takes as inputs $\phi(s_t)$ and a_t and predicts the feature representation $\hat{\phi}(s_{t+1})$ of s_{t+1}. The prediction error in the feature space is used as the curiosity based intrinsic reward signal. As there is no incentive for $\phi(s_t)$ to encode any environmental features that can not influence or are not influenced by the agent's actions, the learned exploration strategy of our agent is robust to uncontrollable aspects of the environment.

- The proposed intrinsic motivation is proportional to $-\log P^{(t)}(\boldsymbol{\phi}(\boldsymbol{s_{t+1}})|\boldsymbol{s_t}, \boldsymbol{a_t})$ if we consider $P^{(t)}(\boldsymbol{\phi}(\boldsymbol{s_{t+1}})|\boldsymbol{s_t}, \boldsymbol{a_t})$ to be Gaussian distribution with $\hat{\boldsymbol{\phi}}(\boldsymbol{s_{t+1}})$ as its mean and a (scaled) identity matrix as its covariance matrix.

- Without the inverse model, training the forward model can results in a representation collapse: $\boldsymbol{\phi}(\boldsymbol{s_{t+1}})$ becomes independent of $\boldsymbol{s_{t+1}}$ which results in a trivially minimum loss for the forward model.

Example 2: “Intelligent“ behavior with seeking surprise

- Pathak et al. in ICML 2017 (>2’000 citations)

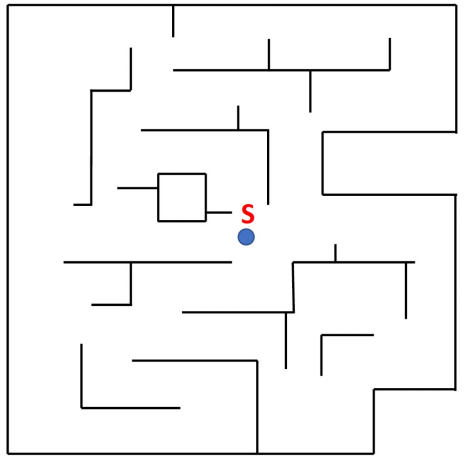


- Seeking surprise enable agents to learn the task in the absence of any extrinsic reward.
- Seeking surprise enable learning skill that are generalizable to other tasks.

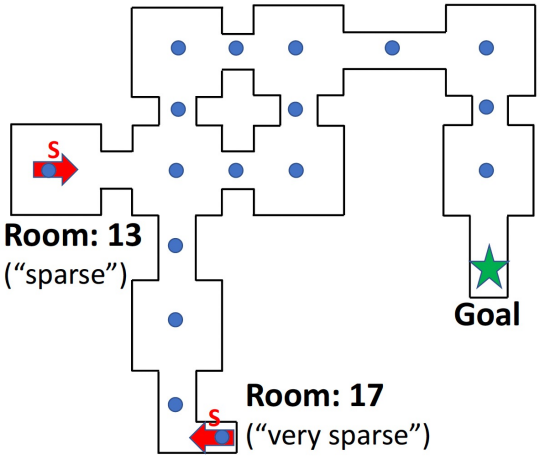
Link to the video:
https://youtu.be/J3FHOyhUn3A

Example 2: “Intelligent“ behavior with seeking surprise

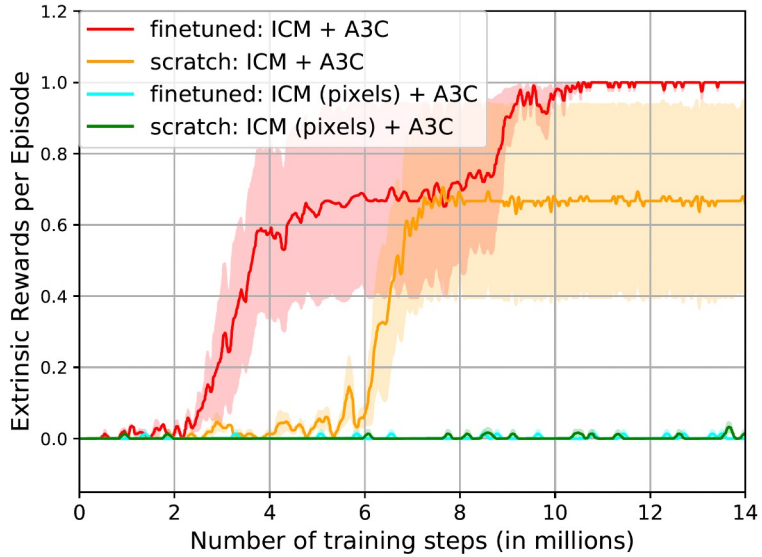
- Pathak et al. in ICML 2017 (>2’000 citations)
- Seeking surprise enables learning exploration strategies that are useful in other environments.



(a) Train Map Scenario



(b) Test Map Scenario



- Comments for the previous slide:
- Figure 4 caption (copied from the original paper):

Figure 4. Maps for VizDoom 3-D environment: (a) For generalization experiments (c.f. Section 4.3), map of the environment where agent is pre-trained only using curiosity signal without any reward from environment. 'S' denotes the starting position. (b) Testing map for VizDoom experiments. Green star denotes goal location. Blue dots refer to 17 agent spawning locations in the map in the "dense" case. Rooms 13, 17 are the fixed start locations of agent in "sparse" and "very sparse" reward cases respectively. Note that textures are also different in train and test maps.

- Figure 8 caption (copied from the original paper):

Figure 8. Performance of ICM + A3C agents on the test set of *VizDoom* in the "very sparse" reward case. Fine-tuned models learn the exploration policy without any external rewards on the training maps and are then fine-tuned on the test map. The scratch models are directly trained on the test map. The fine-tuned ICM + A3C significantly outperforms ICM + A3C indicating that our curiosity formulation is able to learn generalizable exploration policies. The pixel prediction based ICM agent completely fail. Note that textures are also different in train and test.

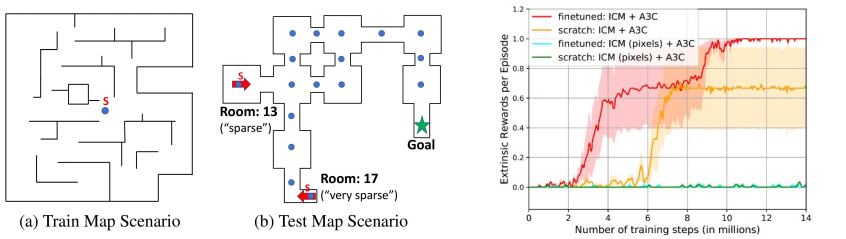


Figure 4 Figure 8

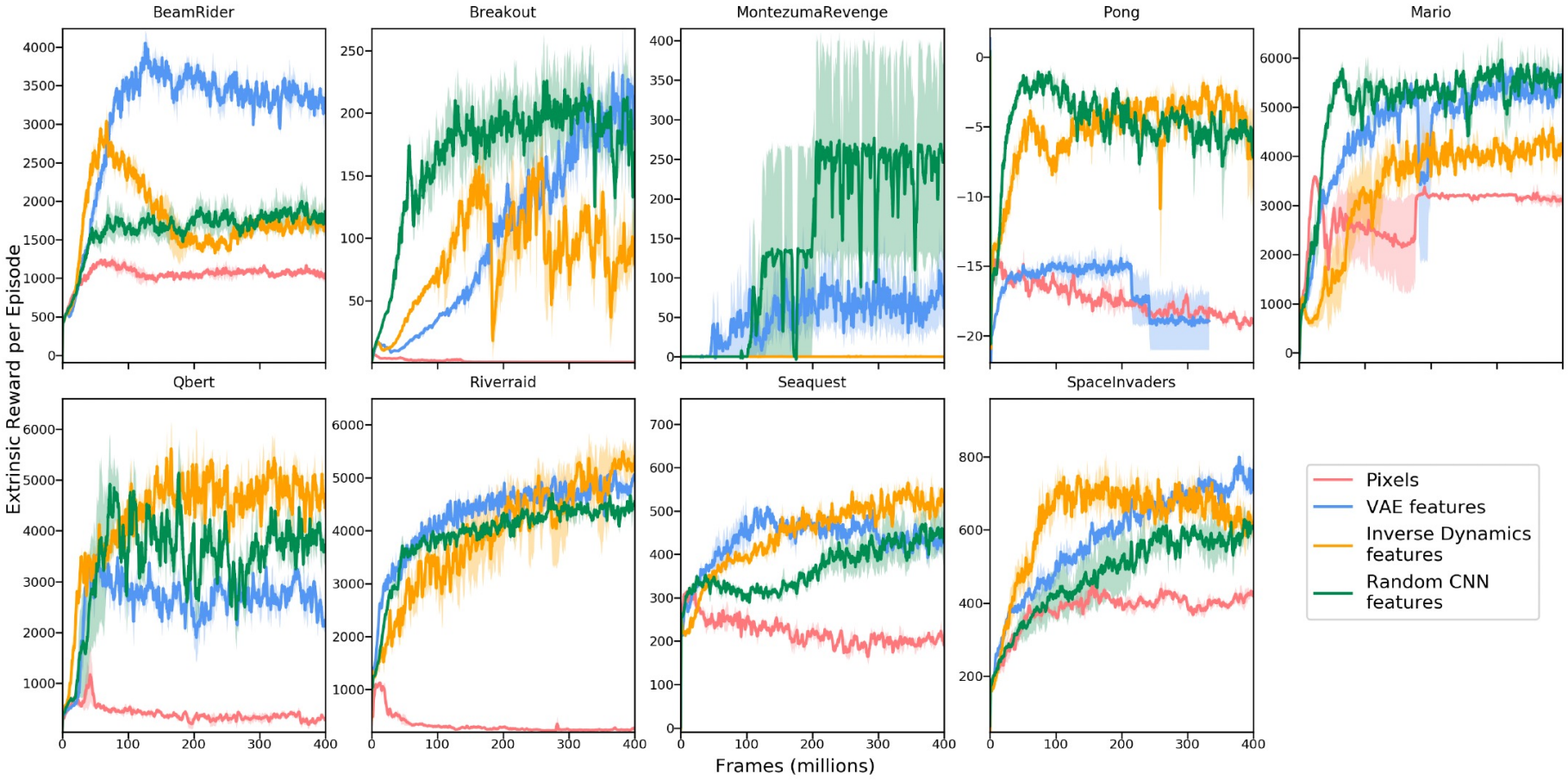
- Description of the results (copied from the original paper):

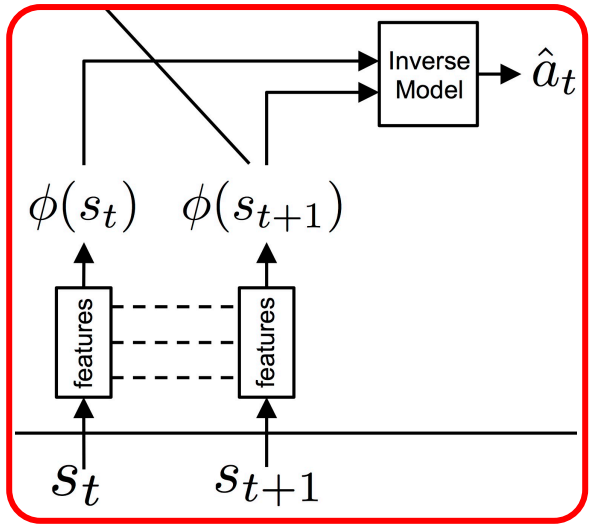
Fine-tuning with extrinsic rewards: If it is the case that the agent has actually learned useful exploratory behavior, then it should be able to learn quicker than starting from scratch even when external rewards are provided by environment. We perform this evaluation on *VizDoom* where we pre-train the agent using curiosity reward on a map showed in Figure 4a. We then test on the "very sparse" reward setting of 'DoomMyWayHome-v0' environment which uses a different map with novel textures (see Figure 4b) as described earlier in Section 4.1.

Results in Figure 8 show that the ICM agent pre-trained only with curiosity and then fine-tuned with external reward learns faster and achieves higher reward than an ICM agent trained from scratch to jointly maximize curiosity and the external rewards. This result confirms that the learned exploratory behavior is also useful when the agent is required to achieve goals specified by the environment. It is also worth noting that ICM-pixels does not generalize to this test environment. This indicates that the proposed mechanism of measuring curiosity is significantly better for learning skills that generalize as compared to measuring curiosity in the raw sensory space.

Example 2: ICM follow-up by Burda et al. in ICLR 2019

- 54 Atari game: Learning with *no extrinsic reward!*





- Comments for the previous slide:

- Figure caption (copied from the original paper):

Figure 2: A comparison of feature learning methods on 8 selected Atari games and the Super Mario Bros. These evaluation curves show the mean reward (with standard error) of agents trained purely by curiosity, without reward or an end-of-episode signal. We see that our purely curiosity-driven agent is able to gather rewards in these environments without using any extrinsic reward at training. Results on all of the Atari games are in the appendix in Figure 8. We find curiosity model trained on pixels does not work well across any environment and VAE features perform either same or worse than random and inverse dynamics features. Further, inverse dynamics-trained features perform better than random features in 55% of the Atari games. An interesting outcome of this analysis is that random features for modeling curiosity are a simple, yet surprisingly strong baseline and likely to work well in half of the Atari games.

Intermediate summary

- Reward can be seen as an internal signal consisting of an extrinsic and an intrinsic component.

- Seeking surprise and novelty as intrinsic rewards can lead to
    - efficient exploration for finding sources of extrinsic rewards.
    - self-supervised learning of complex behavior even in the absence of rewards.

Outline

Part 1. **Exploration bonus in tabular RL**

✓- Multi-Armed Bandits (MAB)

✓- Markov Decision Processes (MDP)

Part 2. **Curiosity-driven RL**

✓- Intelligent behavior in the absence of 'reward'

✓- Surprise, Novelty, ~~and Information-gain~~ in Deep RL

- Meta-learning of the reward function

Part 3. **Noisy TV problem**

- Being curious in the presence of stochasticity
- Noisy TV problem in curiosity driven Deep RL
- Over-optimism in humans and distraction by stochasticity.