Exploration Strategies in Reinforcement Learning

Dongmin Lee

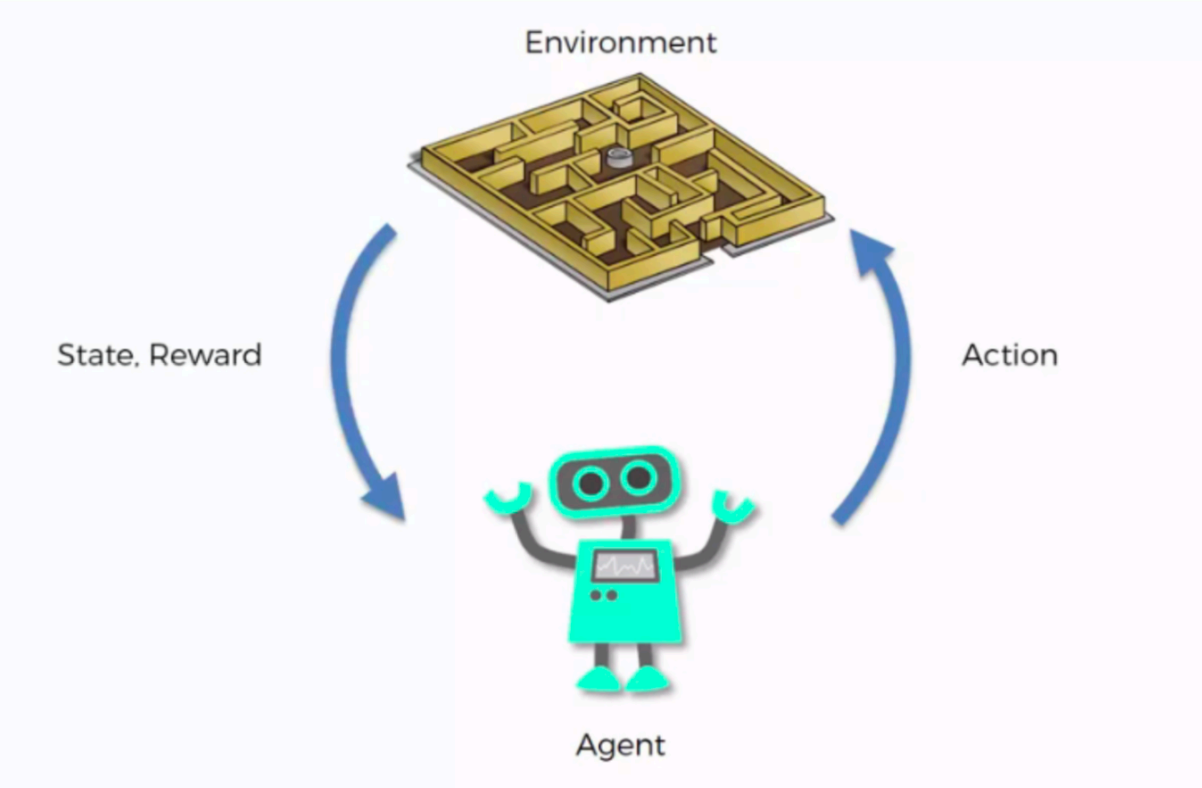
PR Study for ML

Jul, 2020

Outline

- Introduction
  - Reinforcement Learning
  - Deep Reinforcement Learning
  - Challenges of Deep RL
- Exploration Strategies in RL
- Entropy Regularization in RL
- Recent Trend

Reinforcement Learning

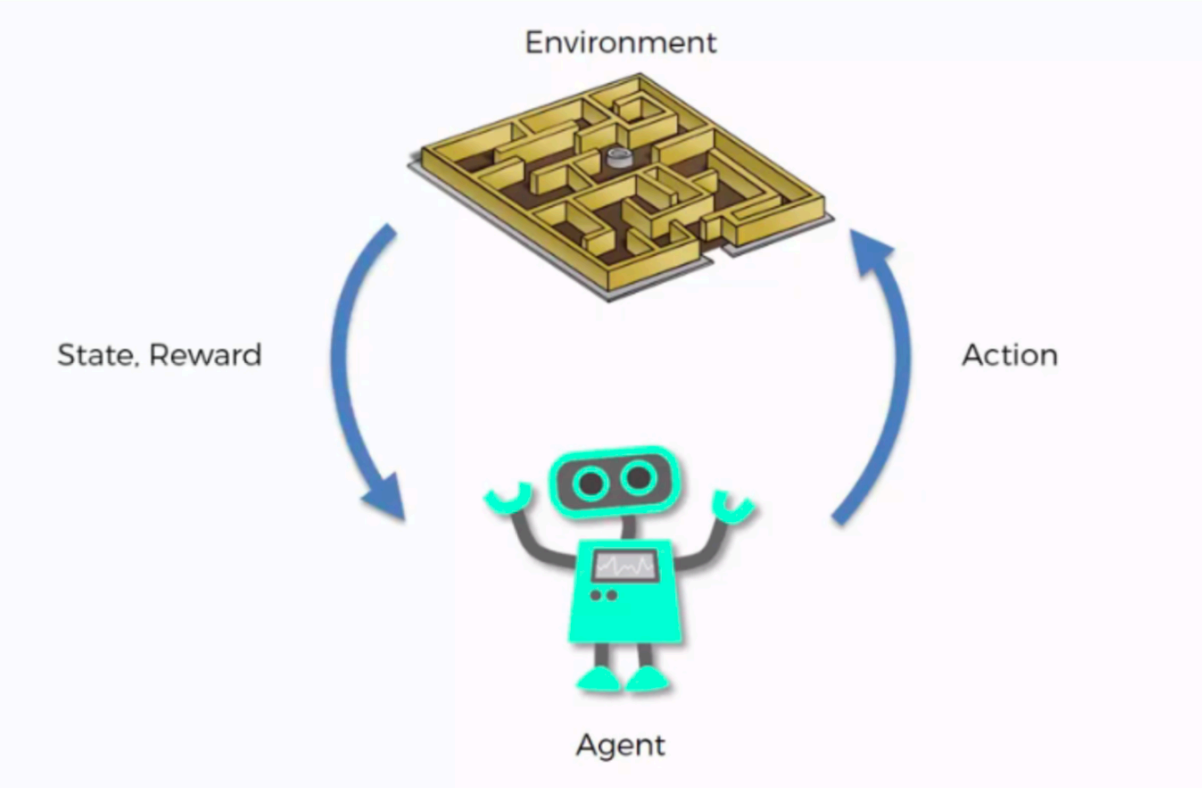


RL provides a formalism for behaviors

- Problem of a goal-directed agent interacting with an uncertain environment
- Interaction → adaptation

feedback & decision

Reinforcement Learning



Distinct Features of RL (★ ★ ★ ★ ★)

- There is no supervisor, only a reward signal
- Time really matters (sequential, non i.i.d data)
- Feedback is delayed, not instantaneous
- Trade-off between exploration and exploitation

Deep Reinforcement Learning

Reinforcement learning seems to work well in solving simple problems

- Structured environments
- Structured inputs (observations, states)
- Static environments
- Failure allowed

Deep Reinforcement Learning

Can reinforcement learning solve sequential decision making problems in real world ?

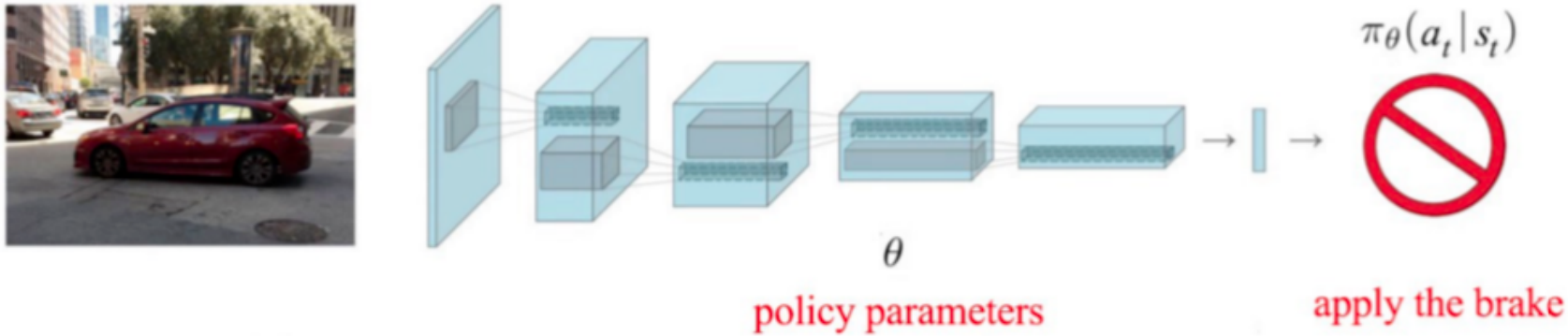
Many challenges:

- Unstructured environments
- Complex sensory inputs
- Fast adaptation
- Reliability, safety

Deep Reinforcement Learning

Deep RL = RL + Deep learning

- RL: sequential decision making interacting with environments
- Deep learning: representation of policies and value functions



Deep Reinforcement Learning

Some deep RL applications

Deep Reinforcement Learning

Some deep RL applications

- Game: Tetris, Chess, Go, Shogi, Dota2, StarCraft II

Deep Reinforcement Learning

Some deep RL applications

- Game: Tetris, Chess, Go, Shogi, Dota2, StarCraft II
- Robotics: Locomotion, Manipulation, Navigation

Deep Reinforcement Learning

Some deep RL applications

- Game: Tetris, Chess, Go, Shogi, Dota2, StarCraft II
- Robotics: Locomotion, Manipulation, Navigation
- Computer Vision: Object detection, Segmentation, Visual target tracking

Deep Reinforcement Learning

Some deep RL applications

- Game: Tetris, Chess, Go, Shogi, Dota2, StarCraft II
- Robotics: Locomotion, Manipulation, Navigation
- Computer Vision: Object detection, Segmentation, Visual target tracking
- NLP: Real-time machine translation, Question & Answering (Q&A), Dialogue generation

Deep Reinforcement Learning

Some deep RL applications

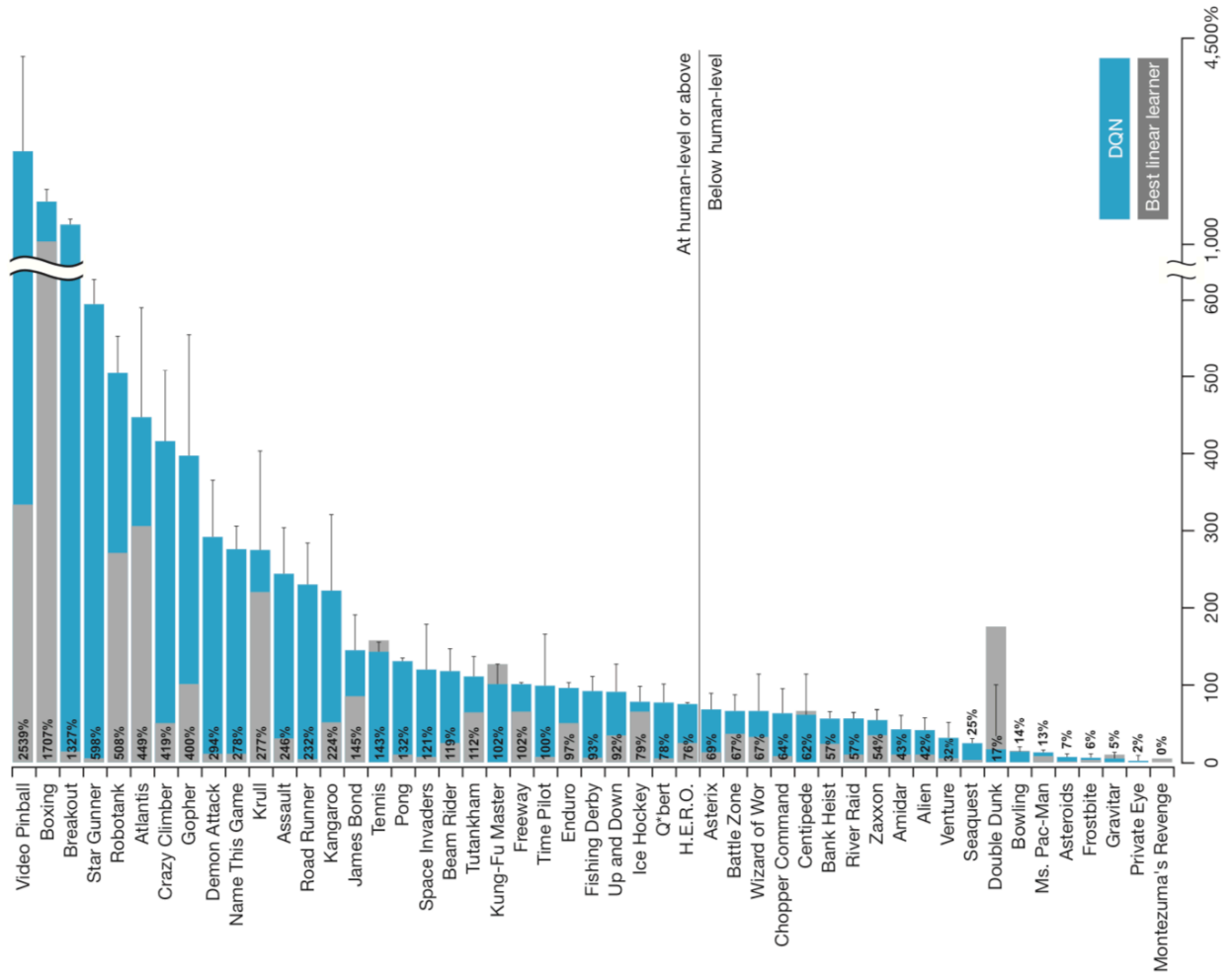
- Game: Tetris, Chess, Go, Shogi, Dota2, StarCraft II
- Robotics: Locomotion, Manipulation, Navigation
- Computer Vision: Object detection, Segmentation, Visual target tracking
- NLP: Real-time machine translation, Question & Answering (Q&A), Dialogue generation
- Recommender System: Interactive recommendation

Deep Reinforcement Learning

Deep RL applications: Atari breakout (2013)

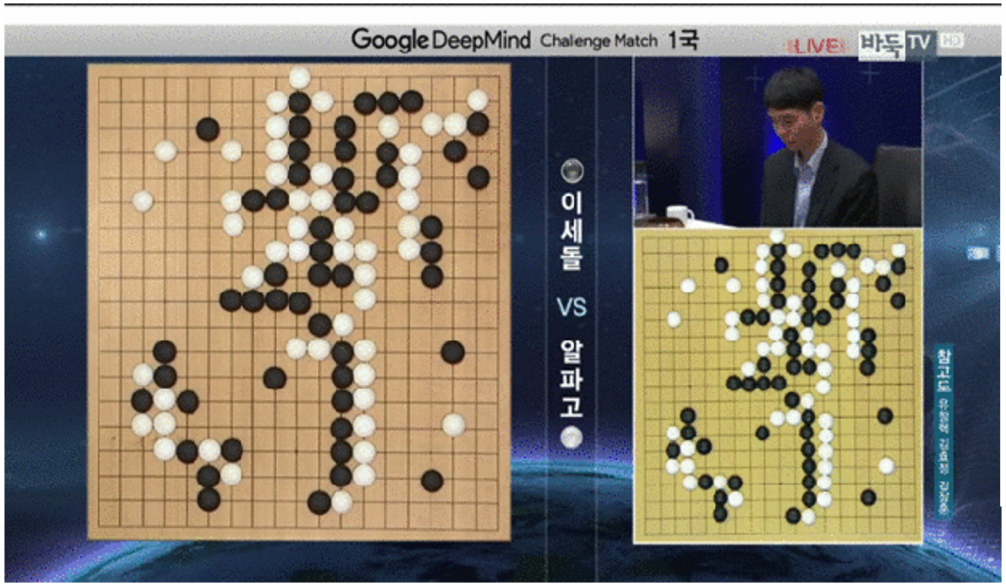
Deep Reinforcement Learning

Deep RL applications: Atari games (2015)



Deep Reinforcement Learning

Deep RL applications: AlphaGo (2016)

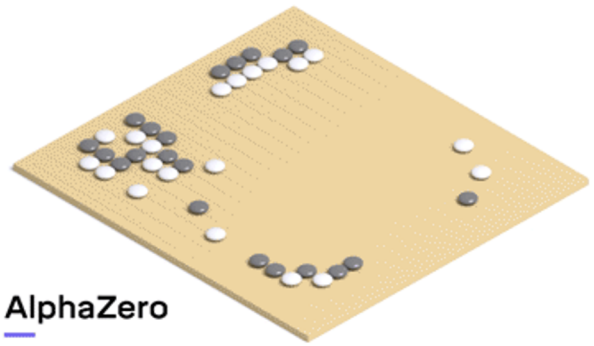


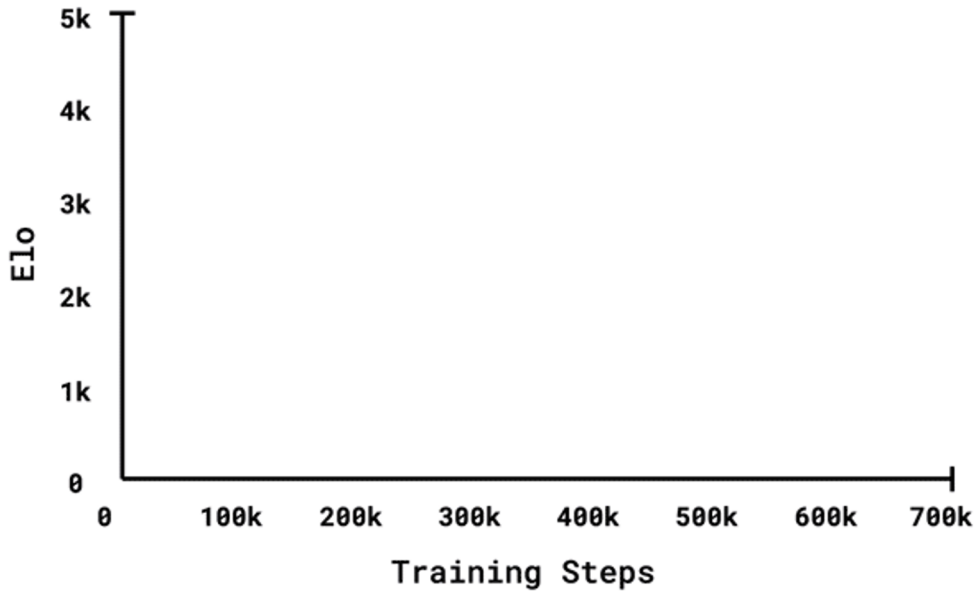
Deep Reinforcement Learning

Deep RL applications: Google Robotics (2017)

Deep Reinforcement Learning

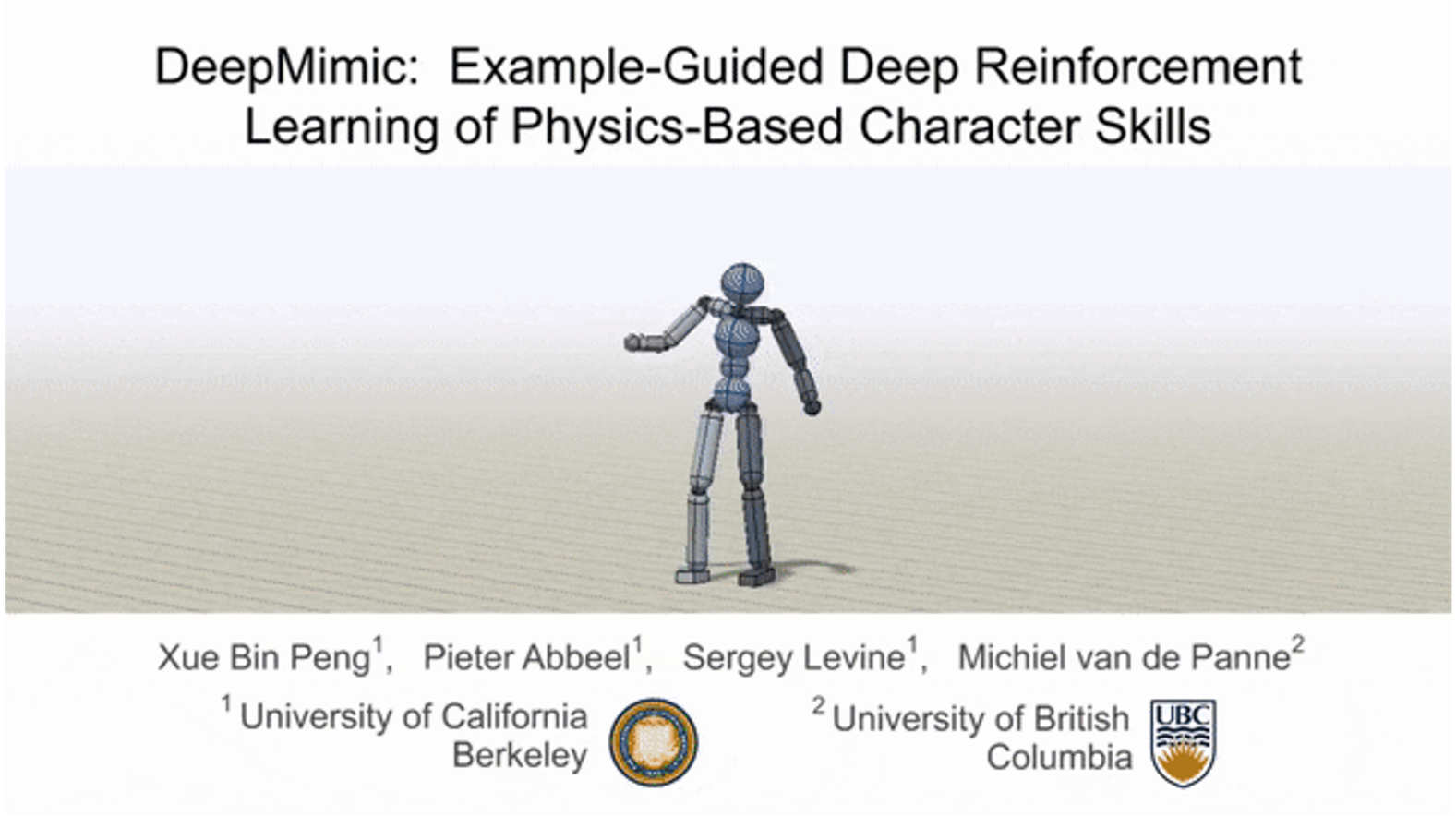
Deep RL applications: AlphaZero (2018)





Deep Reinforcement Learning

Deep RL applications: DeepMimic (2018)

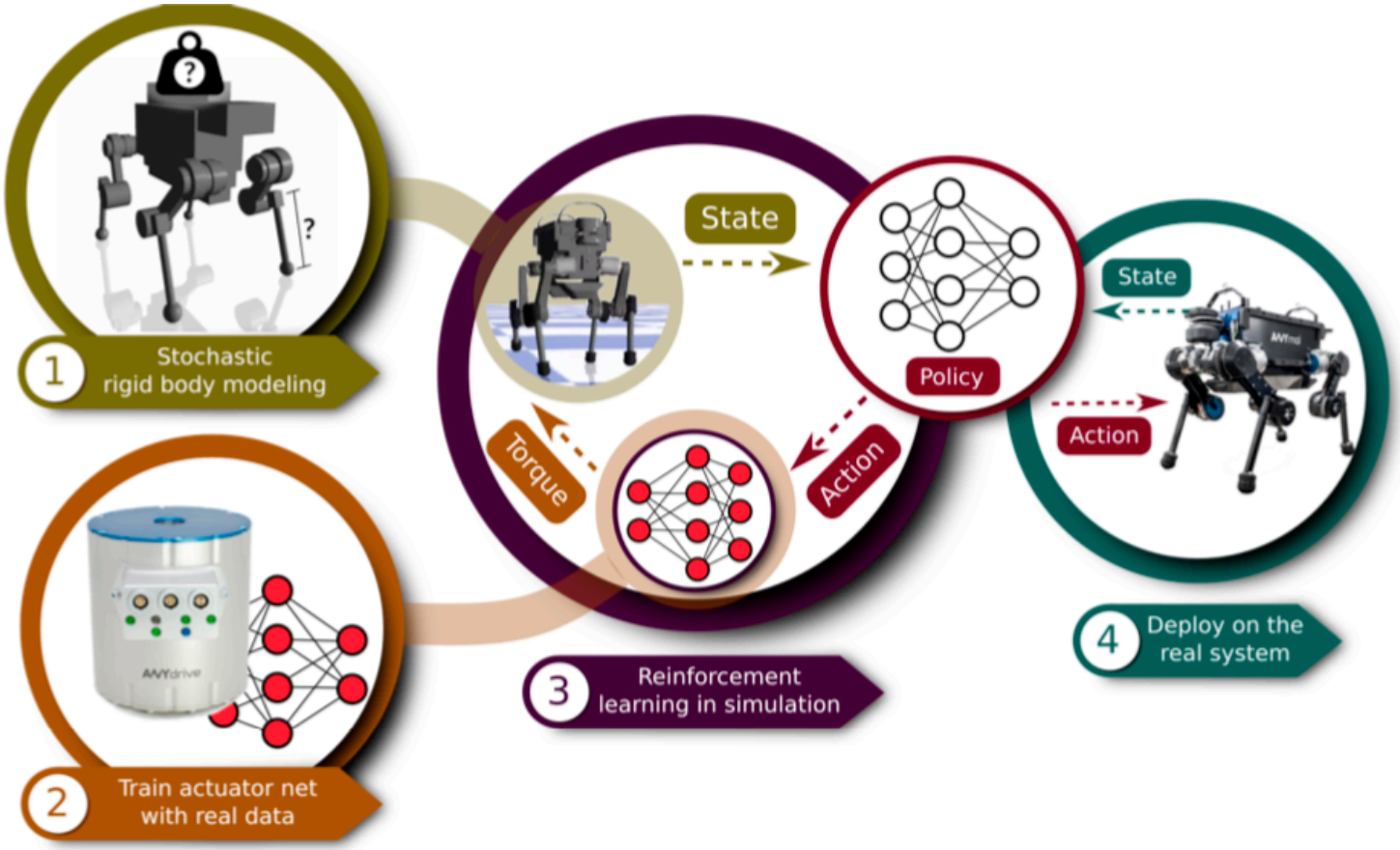


Deep Reinforcement Learning

Deep RL applications: Dota2 (2018)

Deep Reinforcement Learning

Deep RL applications: ANYmal (2018)

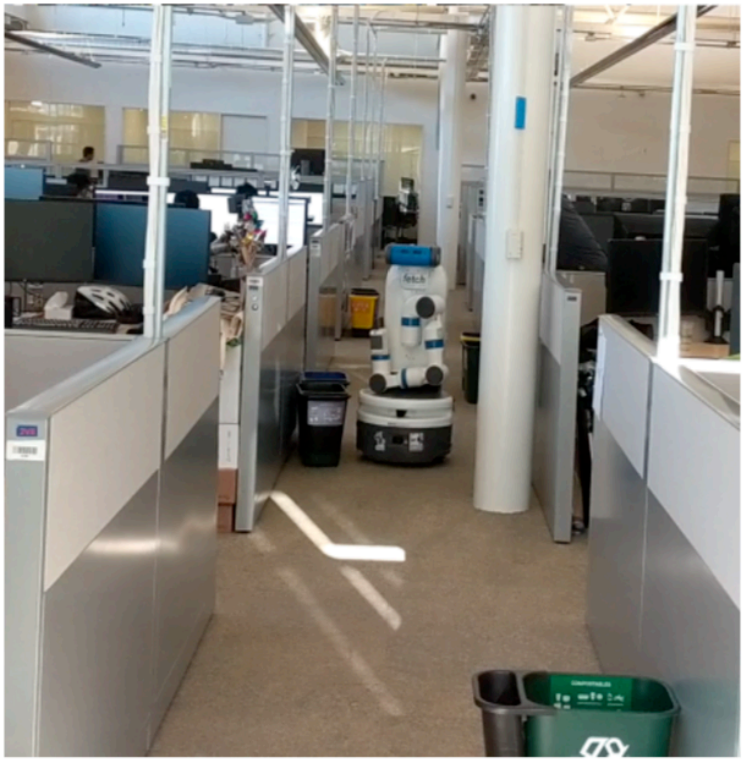


Deep Reinforcement Learning

Deep RL applications: AlphaStar (2019)

Deep Reinforcement Learning

Deep RL applications: Robot Navigation (2019)



Deep Reinforcement Learning

Deep RL applications: Solving Rubik's Cube (2019)

Challenges of Deep RL

Challenges of Deep RL

- Fast, stable learning

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- Exploration

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- Exploration
- Reward design

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- Exploration
- Reward design
- Sparse reward signals

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- Exploration
- Reward design
- Sparse reward signals
- Sim2Real gap

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- Exploration
- Reward design
- Sparse reward signals
- Sim2Real gap
- Safety / reliability

Challenges of Deep RL

- Fast, stable learning
- Hyperparameter tuning
- Sample efficiency: Huge # of samples
- **Exploration**
- Reward design
- Sparse reward signals
- Sim2Real gap
- Safety / reliability

Exploration

Common issue: Exploration

- Is there a value of exploring unknown regions of the environment?
- Which action should we try to explore unknown regions of the environment?

Exploration

Trade-off between exploration and exploitation

- Exploitation: Make the best decision given current information
- Exploration: Gather more information

Q) Why dilemma?

Exploration

Trade-off between exploration and exploitation

- Exploitation: Make the best decision given current information
- Exploration: Gather more information

Q) Why dilemma?

- The best long-term strategy may involve short-term sacrifices
- Need to gather enough information to make the best overall decisions

Examples

1. Restaurant selection
   - Exploitation: Go to your favorite restaurant
   - Exploration: Try a new restaurant

Examples

1. Restaurant selection
   - Exploitation: Go to your favorite restaurant
   - Exploration: Try a new restaurant
2. Online banner advertisements
   - Exploitation: Show the most successful advertisement
   - Exploration: Show a different advertisement

Examples

1. Restaurant selection
   - Exploitation: Go to your favorite restaurant
   - Exploration: Try a new restaurant
2. Online banner advertisements
   - Exploitation: Show the most successful advertisement
   - Exploration: Show a different advertisement
3. Oil drilling
   - Exploitation: Drill at the best known location
   - Exploration: Drill at a new location

Examples

1. Restaurant selection
   - Exploitation: Go to your favorite restaurant
   - Exploration: Try a new restaurant
2. Online banner advertisements
   - Exploitation: Show the most successful advertisement
   - Exploration: Show a different advertisement
3. Oil drilling
   - Exploitation: Drill at the best known location
   - Exploration: Drill at a new location
4. Game playing
   - Exploitation: Play the move you believe is best
   - Exploration: Play an experimental move

Exploration Strategies in RL

How can we algorithmically solve this issue exploration vs exploitation?

1. ϵ-Greedy
2. Optimism in the face of uncertainty
3. Thompson (posterior) sampling
4. Curiosity-driven exploration
5. Information theoretic exploration

Exploration Strategies in RL

How can we algorithmically solve this issue exploration vs exploitation?

1. ϵ-Greedy
2. Optimism in the face of uncertainty
3. Thompson (posterior) sampling
4. Curiosity-driven exploration
5. Information theoretic exploration

Q) What is a means of assessing these strategies?

Exploration Strategies in RL

What is regret?

- An additional means of assessing the algorithms' performance
- A loss that we incur due to time spent during the learning

Exploration Strategies in RL

What is regret?

- An additional means of assessing the algorithms' performance
- A loss that we incur due to time spent during the learning
- The expected cumulative regret:

$$\mathcal{R}_T \triangleq \mathbb{E}_\pi \left[\max_{a \in A} \sum_{t=0}^{T} r_t(a) - r_t(a_t) \right]$$

Exploration Strategies in RL

What is regret?

- An additional means of assessing the algorithms' performance
- A loss that we incur due to time spent during the learning
- The expected cumulative regret:

$$\mathcal{R}_T \triangleq \mathbb{E}_\pi \left[\max_{a \in A} \sum_{t=0}^{T} r_t(a) - r_t(a_t) \right]$$

Regret problem

- To find a strategy that minimizes the expected cumulative regret $\mathcal{R}_T$

Exploration Strategies in RL

What is regret?

- An additional means of assessing the algorithms' performance
- A loss that we incur due to time spent during the learning
- The expected cumulative regret:

$$\mathcal{R}_T \triangleq \mathbb{E}_\pi \left[\max_{a \in A} \sum_{t=0}^{T} r_t(a) - r_t(a_t) \right]$$

Regret problem

- To find a strategy that minimizes the expected cumulative regret $\mathcal{R}_T$

- **Maximize expected cumulative reward ≡ minimize expected cumulative regret!**

Exploration Strategies in RL

Method 1: ϵ-Greedy

Idea:

- Choose a greedy action $(\arg\max Q^{\pi}(s,a))$ with probability $(1-\epsilon)$
- Choose a random action with probability ϵ

Advantages:

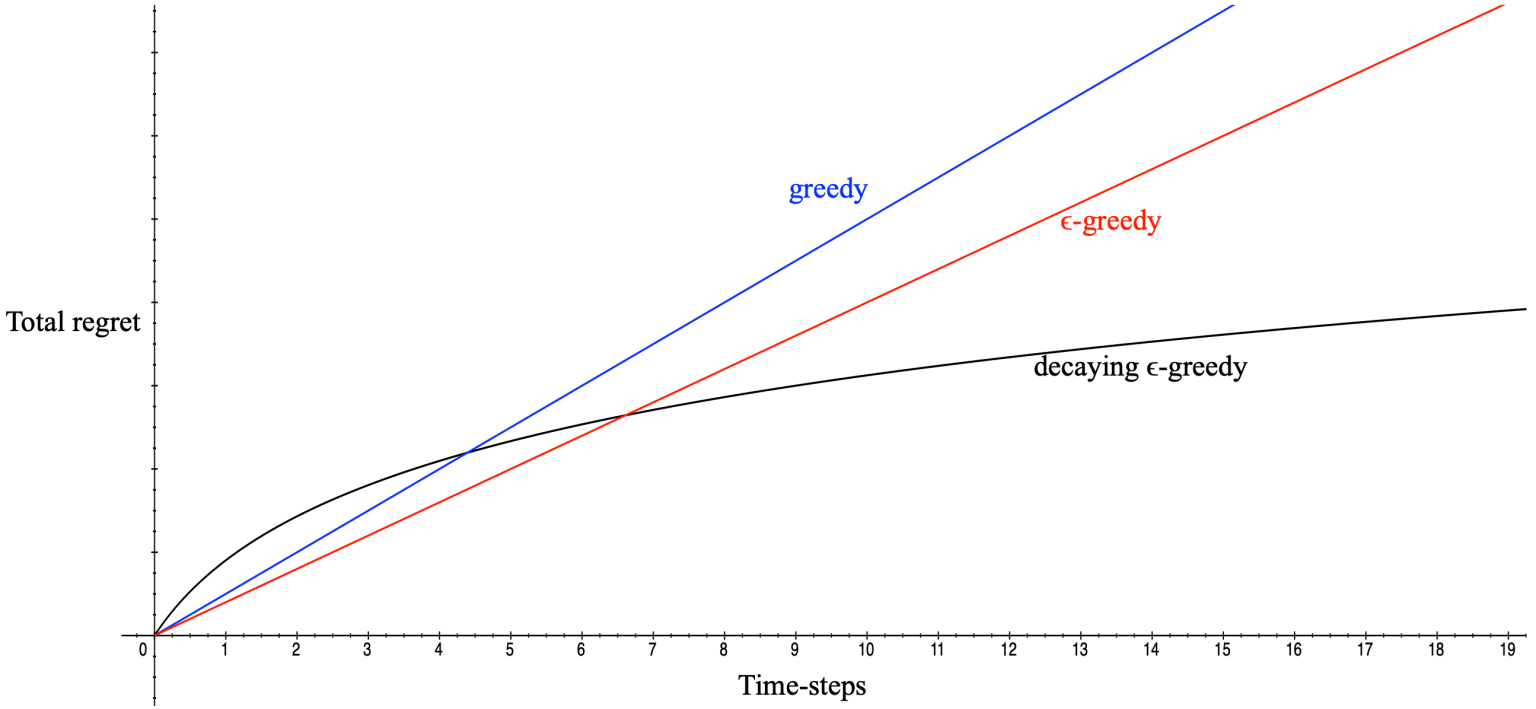
1. Very simple
2. Light computation

Disadvantages:

1. Fine tuning ϵ
2. Not quite systematic: No focus on unexplored regions
3. Inefficient

Exploration Strategies in RL

A comparison of greedy, ϵ-greedy, decaying ϵ-greedy



Exploration Strategies in RL

Method 2: Optimism in the face of uncertainty (OFU)

Idea:

1. Construct a confidence set for MDP parameters using collected samples
2. Choose MDP parameters that give the highest rewards
3. Construct an optimal policy of the optimistically chosen MDP

Advantages:

1. Regret optimal
2. Systematic: More focus on unexplored regions

Disadvantages:

1. Complicated
2. Intensive computation

Exploration Strategies in RL

Method 3: Thompson (posterior) sampling

Idea:

1. Sample MDP parameters from posterior distribution
2. Construct an optimal policy of the sampled MDP parameters
3. Update the posterior distribution

Advantages:

1. Regret optimal
2. Systematic: More focus on unexplored regions

Disadvantages:

1. Somewhat complicated

Exploration Strategies in RL

Method 4: Curiosity-driven exploration

Idea:

1. Use an intrinsic reward through Intrinsic Curiosity Module (ICM) with extrinsic reward
2. Formulate curiosity as the error to predict the action to be executed
3. Predict the next state given the executed action and a feature representation of the current state
4. Generate the intrinsic reward by defining loss function between the predicted next state and a feature representation of actual next state

Advantages:

1. Good empirical performance
2. Systematic: More focus on unexplored regions

Disadvantages:

1. Intensive computation
2. May be limited in the real world

Exploration Strategies in RL

Method 5: Information theoretic exploration

Idea:

- Use high entropy policy to explore more

Advantages:

1. Simple
2. Good empirical performance

Disadvantages:

1. More theoretic analyses needed

Entropy Regularization in RL

What is entropy?

Entropy Regularization in RL

What is entropy?

- A measure that represents **disorder or uncertainty** within a system

Entropy Regularization in RL

What is entropy?

- A measure that represents **disorder or uncertainty** within a system
- The well-known standard Shannon-Gibbs entropy:

$$H(P) \triangleq \mathop{\mathbb{E}}_{X\sim P}\left[-\log\left(P(X)\right)\right] = -\sum_{x\in X} P(x)\log P(x)$$

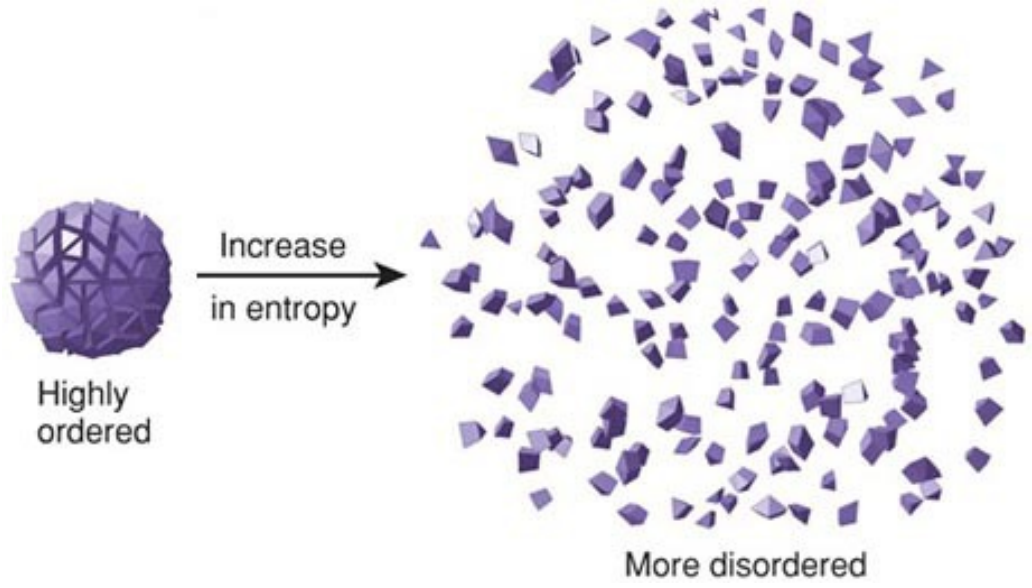
Entropy Regularization in RL

What is entropy?

- A measure that represents **disorder or uncertainty** within a system
- The well-known standard Shannon-Gibbs entropy:

$$H(P) \triangleq \mathop{\mathbb{E}}_{X\sim P}\left[-\log\left(P(X)\right)\right] = -\sum_{x\in X} P(x)\log P(x)$$

- More generally, entropy refers to **disorder or uncertainty**



Entropy Regularization in RL

What's the benefit of high entropy policy?

Entropy Regularization in RL

What's the benefit of high entropy policy?

- Entropy of policy:

$$H(\pi(\cdot\,|s)) \triangleq \mathbb{E}_{a\sim\pi}\big[-\log\big(\pi(a|s)\big)\big] = -\sum_{a}\pi(a|s)\log\pi(a|s)$$

- Higher disorder in policy π
- Try new risky behaviors: Potentially explore unexplored regions

MDPs

A Markov Decision Process (MDP) is a tuple $< S, A, p, r, \gamma >$, consisting of

- S: set of states (state space)
  e.g., $S = \{1, \dots, n\}$ (discrete), $S = \mathbb{R}^n$ (continuous)

- A: set of actions (action space)
  e.g., $A = \{1, \dots, m\}$ (discrete), $A = \mathbb{R}^m$ (continuous)

- p: state transition probability
  $p(s'|s, a) \triangleq \text{Prob}(s_{t+1} = s'|s_t = s, a_t = a)$

- r: reward function
  $r(s_t, a_t) = r_t$

- $\gamma \in (0,1]$: discount factor

Soft MDPs

Standard MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

Soft MDPs

Standard MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

Maximum entropy MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H\big(\pi(\cdot\,|s_t)\big)\right)\right]$$

where $H\big(\pi(\cdot\,|s_t)\big) = \mathbb{E}_{a\sim\pi}\big[-\log\big(\pi(a|s)\big)\big] = -\sum_a \pi(a|s)\log\pi(a|s)$.

Advantage of Maximum Entropy

What's the benefit of maximum entropy regularization?

- Computation:
  No maximization involved
- Exploration:
  Potentially explore unexplored regions using high entropy policy
- Structural similarity:
  Can combine it with many RL methods for standard MDP

Soft Actor-Critic (SAC)

Soft Actor-Critic:
Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

Tuomas Haarnoja [1] Aurick Zhou [1] Pieter Abbeel [1] Sergey Levine [1]

SAC (Jan 4 2018)

Soft Actor-Critic Algorithms and Applications

Tuomas Haarnoja*†‡ Aurick Zhou*† Kristian Hartikainen*† George Tucker‡
Sehoon Ha‡ Jie Tan‡ Vikash Kumar‡ Henry Zhu† Abhishek Gupta†
Pieter Abbeel† Sergey Levine†‡

SAC-AEA (Dec 13 2018)

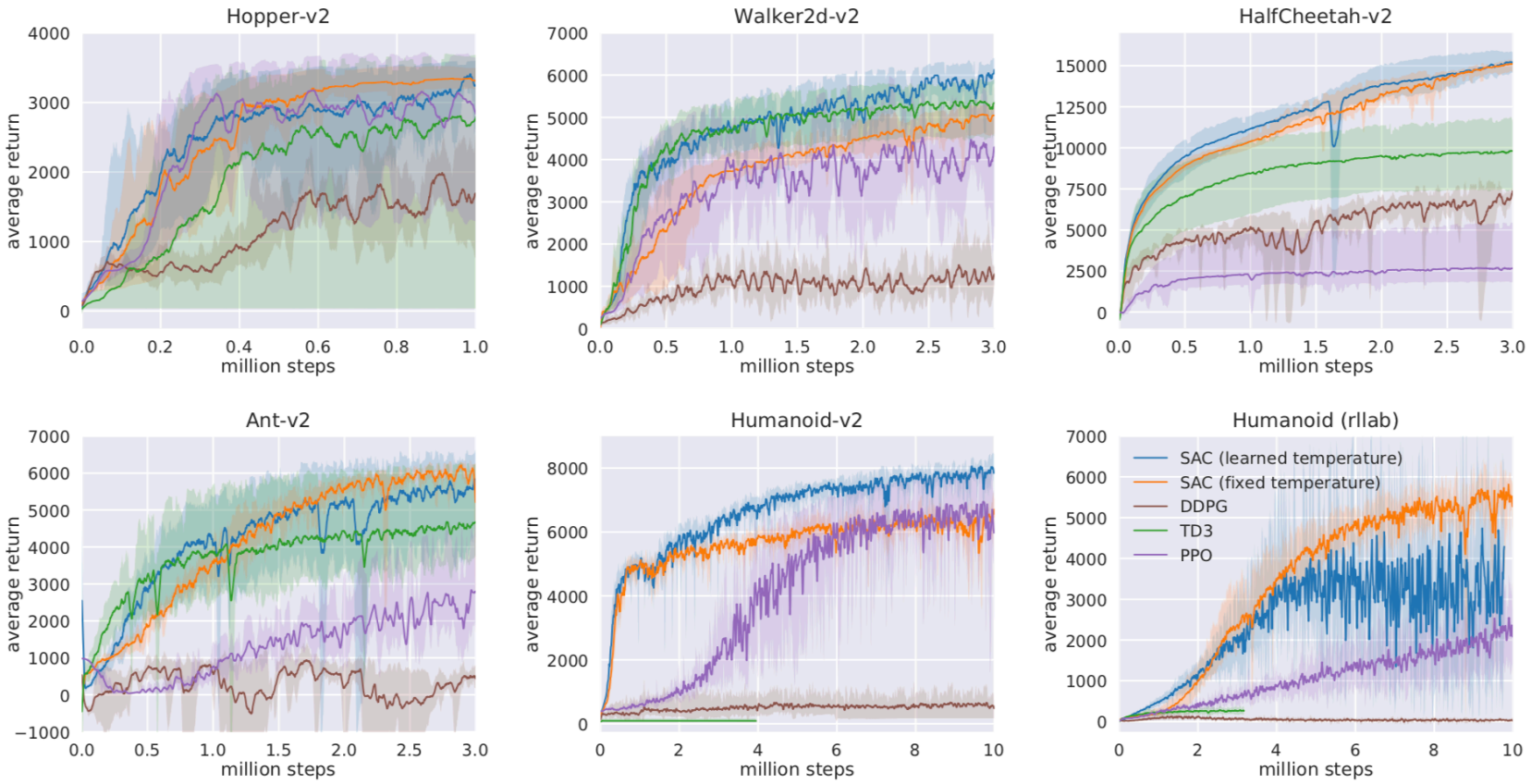
Idea:

- Maximum entropy + off-policy actor-critic

Advantages:

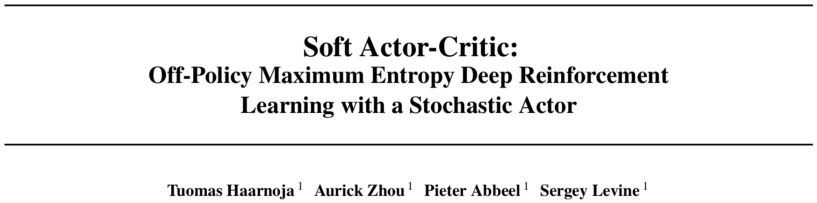
- Exploration
- Sample efficiency
- Stable convergence
- Little hyperparameter tuning

Result I

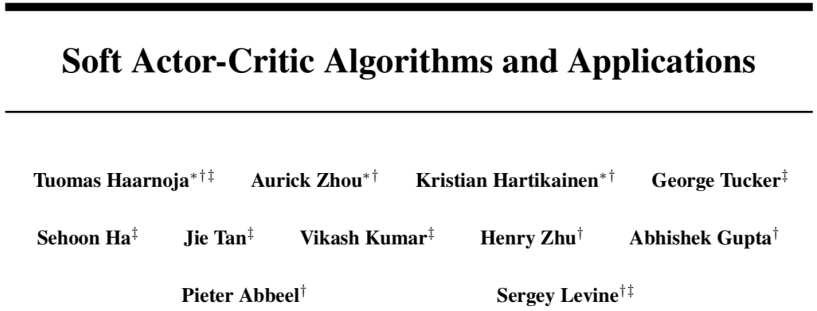


- Soft actor-critic (blue and yellow) performs **consistently** across all tasks and **outperforming** both on-policy and off-policy methods in the most challenging tasks.

Advantages and Disadvantages



SAC (Jan 4 2018)



SAC-AEA (Dec 13 2018)

Idea:

- Maximum entropy + off-policy actor-critic

Advantages:

- Exploration
- Sample efficiency
- Stable convergence
- Little hyperparameter tuning

Disadvantages:

- Performance loss
- Coefficient of entropy term

Advantages and Disadvantages

Soft Actor-Critic:
Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor

Tuomas Haarnoja [1] Aurick Zhou [1] Pieter Abbeel [1] Sergey Levine [1]

SAC (Jan 4 2018)

Soft Actor-Critic Algorithms and Applications

Tuomas Haarnoja*†‡ Aurick Zhou*† Kristian Hartikainen*† George Tucker‡
Sehoon Ha‡ Jie Tan‡ Vikash Kumar‡ Henry Zhu† Abhishek Gupta†
Pieter Abbeel† Sergey Levine†‡

SAC-AEA (Dec 13 2018)

Idea:

- Maximum entropy + off-policy actor-critic

Advantages:

- Exploration
- Sample efficiency
- Stable convergence
- Little hyperparameter tuning

Disadvantages:

- Performance loss
- Coefficient of entropy term

→ **Different types of entropy**

Tsallis Actor-Critic

Q) Can we use different types of entropy?

Tsallis Actor-Critic

Q) Can we use different types of entropy?

- Yes! They are unified by Tsallis entropy.

Tsallis Reinforcement Learning: A Unified Framework for Maximum Entropy Reinforcement Learning

Kyungjae Lee[1], Sungyub Kim[2], Sungbin Lim[3], Sungjoon Choi[3], and Songhwai Oh[1]
Dep. of Electrical and Computer Engineering, Seoul National University[1]
School of Computing, KAIST[2]
Kakao Brain[3]
kyungjae.lee@rllab.snu.ac.kr, sungyub.kim@kaist.ac.kr,
{sungbin.lim, sam.choi}@kakaobrain.com,
songhwai@snu.ac.kr

Maximum Entropy

Standard MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t)\right]$$

Maximum entropy MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H\big(\pi(\cdot\,|s_t)\big)\right)\right]$$

where $H\big(\pi(\cdot\,|s_t)\big) = \mathbb{E}_{a\sim\pi}\big[-\log\big(\pi(a|s)\big)\big] = -\sum_a \pi(a|s)\log\pi(a|s)$.

Maximum Tsallis Entropy

Maximum entropy MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha H\big(\pi(\cdot\,|s_t)\big)\right)\right]$$

Maximum Tsallis entropy MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha S_q\big(\pi(\cdot\,|s_t)\big)\right)\right]$$

Tsallis entropy:

$$S_q\big(\pi(\cdot\,|s_t)\big) = \mathbb{E}_{a\sim\pi}\big[-\ln_q(\pi(a|s))\big]$$

where, $\ln_q(x) = \begin{cases} \log(x), & if\ q = 1\ and\ x > 0 \\ \frac{x^{q-1}-1}{q-1}, & if\ q \neq 1\ and\ x > 0 \end{cases}$, q is called an *entropic index*

Tsallis Reinforcement Learning

What is the main idea?

Tsallis Reinforcement Learning

What is the main idea?

- A unified framework which **generalizes Shannon-Gibbs entropy maximization** to Tsallis entropy maximization in reinforcement learning

Tsallis Reinforcement Learning

What is the main idea?

- A unified framework which **generalizes Shannon-Gibbs entropy maximization** to Tsallis entropy maximization in reinforcement learning
- **The entropic index q controls the exploration-exploitation trade-off**: Different entropic index shows different exploration tendency

Tsallis Reinforcement Learning

What is the main idea?

- A unified framework which **generalizes Shannon-Gibbs entropy maximization** to Tsallis entropy maximization in reinforcement learning
- **The entropic index q controls the exploration-exploitation trade-off**: Different entropic index shows different exploration tendency
- An off-policy maximum Tsallis entropy actor-critic algorithm: **Tsallis actor-critic with a proper entropic index** outperforms exisiting actor-critic methods

Tsallis Entropy

Why is it called Tsallis entropy?

- From the years 2000 on, Tsallis entropy is widely used in the field of physics, information theory, social science

Tsallis Entropy

Why is it called Tsallis entropy?

- From the years 2000 on, Tsallis entropy is widely used in the field of physics, information theory, social science
- Tsallis entropy has been used to describe complex phenomena that cannot be explained by Shannon-Gibbs entropy

Tsallis Entropy

Why is it called Tsallis entropy?

- From the years 2000 on, Tsallis entropy is widely used in the field of physics, information theory, social science
- Tsallis entropy has been used to describe complex phenomena that cannot be explained by Shannon-Gibbs entropy
- Using Tsallis entropy, following phenomena have been explained
  - The fluctuation of the magnetic field in the solar wind
  - The velocity distributions in dissipative dusty plasma
  - The application of statistical mechanics to the study of thermodynamics of overdamped motion of interacting particles
  - Heavy tail distributions are derived from maximum Tsallis entropy problem

Maximum Tsallis Entropy

Maximum Tsallis entropy MDP problem:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(r(s_t, a_t) + \alpha S_q\big(\pi(\cdot\,|s_t)\big)\right)\right]$$

Tsallis entropy:

$$S_q\big(\pi(\cdot\,|s_t)\big) = \mathbb{E}_{a\sim\pi}\big[-\ln_q(\pi(a|s))\big]$$

where, $\ln_q(x) = \begin{cases} \log(x), & if\ q = 1\ and\ x > 0 \\ \frac{x^{q-1}-1}{q-1}, & if\ q \neq 1\ and\ x > 0 \end{cases}$, q is called an *entropic index*

What is intuition for Tsallis entropy?

- Entropy ↑ (α ↑ or q ↓): more exploration, not greedy
- Entropy ↓ (α ↓ or q ↑): more exploitation, greedy

Bandit with Maximum Tsallis Entropy

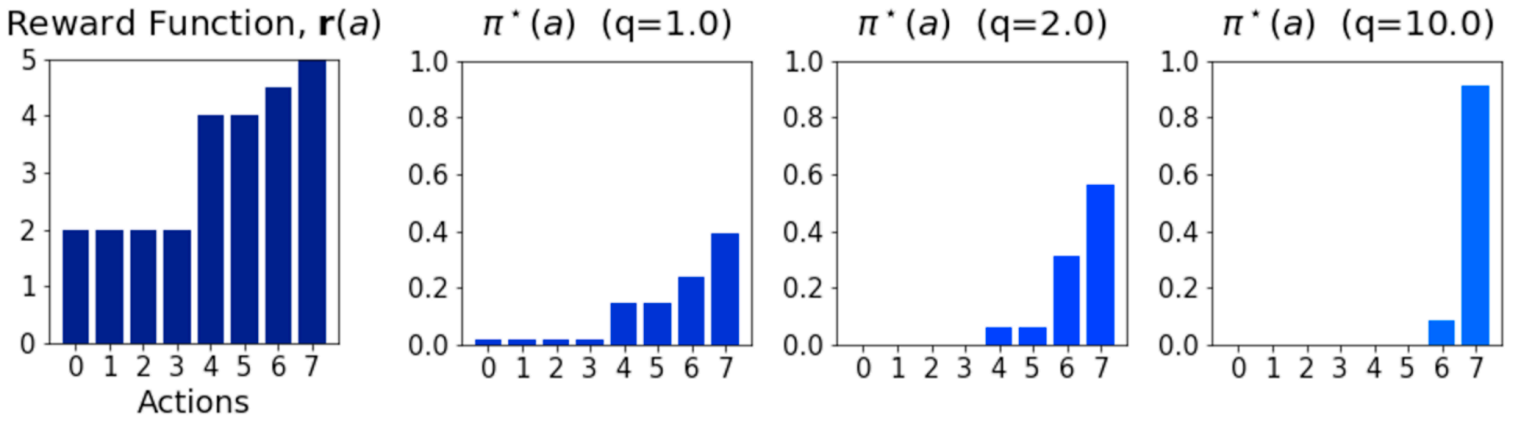
Stochastic multi-armed bandit (MAB) problem:

- Only action space and a reward function: $\{A, r\}$
- The reward only depends on an action: $r(a) = \mathbb{E}[\mathrm{R}|a]$

MAB with Tsallis entropy maximization:

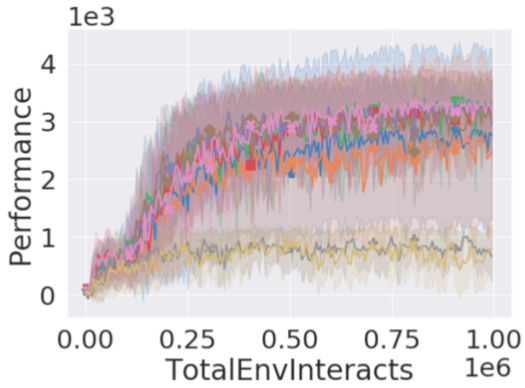
$$\max_{\pi} \mathbb{E}_{a\sim\pi}\left[\sum_{t=0}^{\infty}\left(r(a_t) + \alpha S_q(\pi)\right)\right]$$

Examples of π_q^* with different q values:

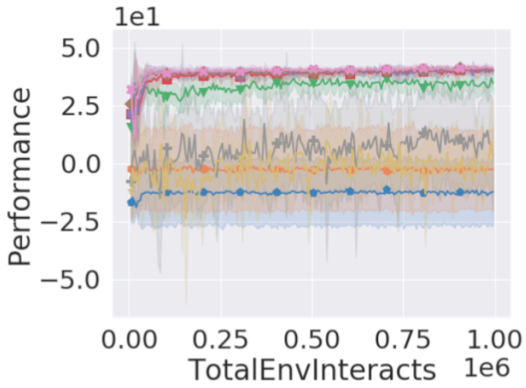


Result I

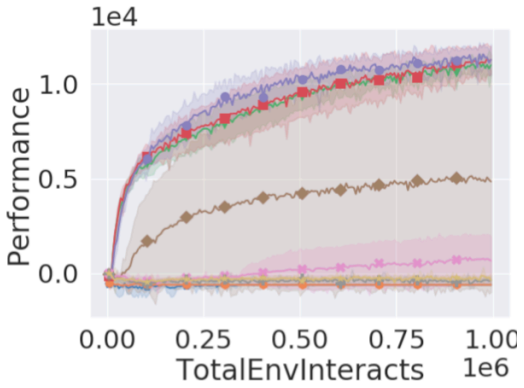
Different q values



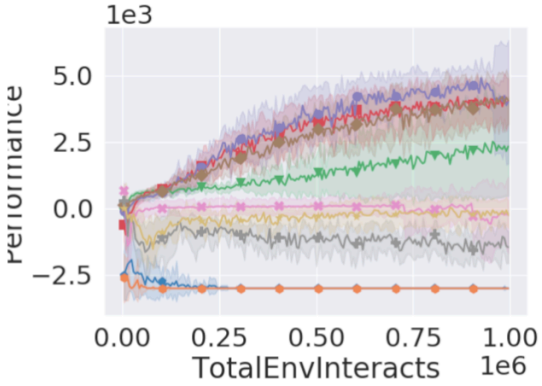
(a) Hopper-v2



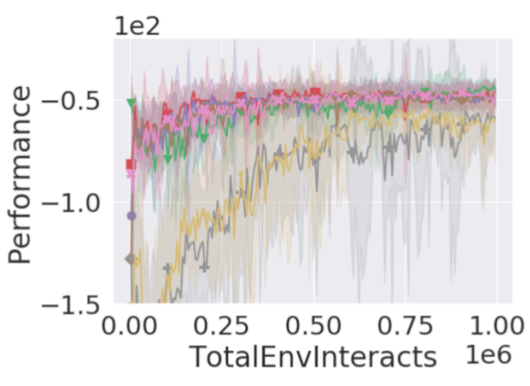
(b) Swimmer-v2



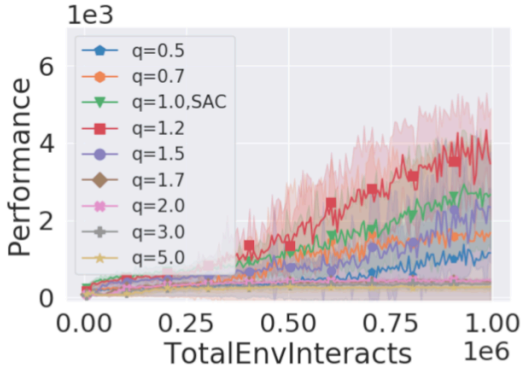
(c) HalfCheetah-v2



(d) Ant-v2



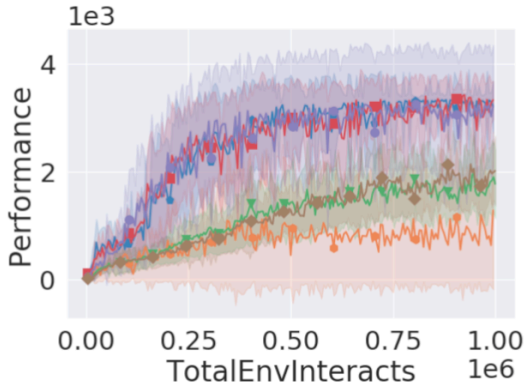
(e) Pusher-v2



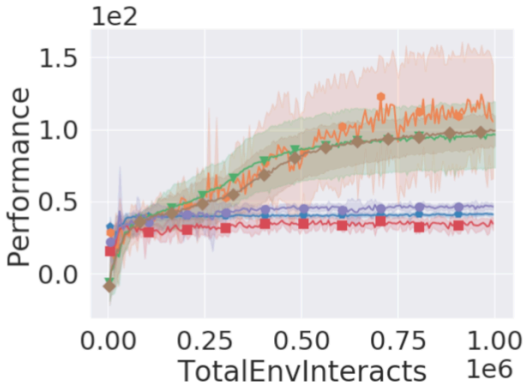
(f) Humanoid-v2

Result II

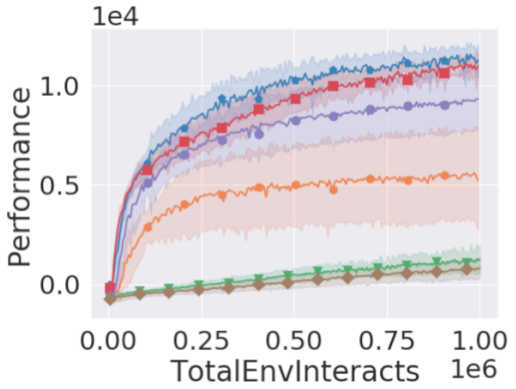
Comparison to existing methods



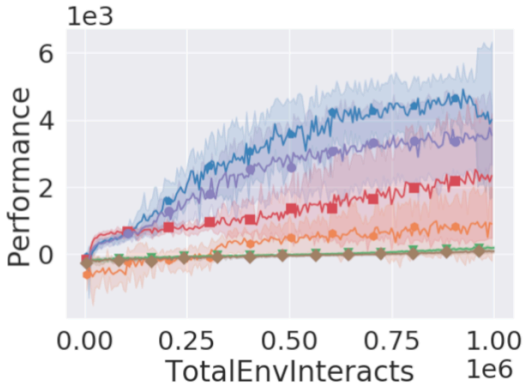
(a) Hopper-v2



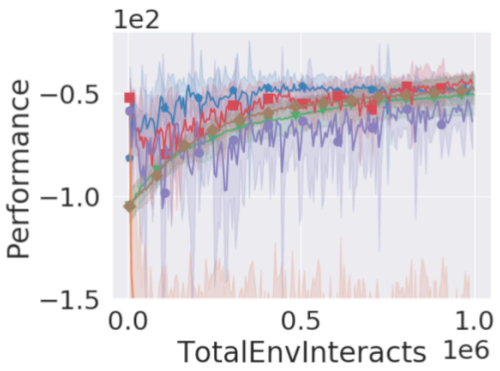
(b) Swimmer-v2



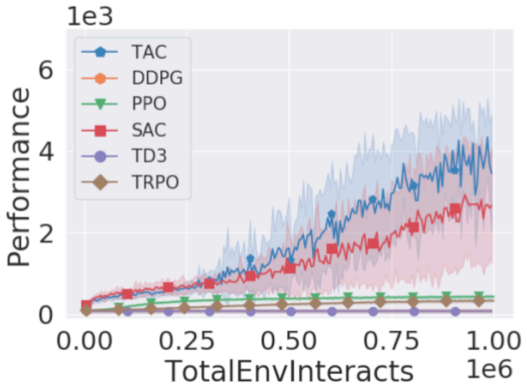
(c) HalfCheetah-v2



(d) Ant-v2



(e) Pusher-v2



(f) Humanoid-v2

Recent Trend I

Meta-RL

- The goal of meta-RL is to learn to adapt quickly to new task given a small amount of experience from previous tasks

Regular RL: learn policy for single task

$$\theta^* = \arg\max_\theta \mathbb{E}_{\pi_\theta(\tau)}[R(\tau)]$$

$$= f_{RL}(\mathcal{M})$$

MDP

Meta-RL: learn adaptation rule

$$\theta^* = \arg\max_\theta \sum_{i=1}^{n} \mathbb{E}_{\pi_{\phi_i}(\tau)}[R(\tau)]$$

meta-training / outer loop

where $\phi_i = f_\theta(\mathcal{M}_i)$

adaptation / inner loop

MDP for task i

$\mathcal{M}_1$ $\mathcal{M}_2$ $\mathcal{M}_3$

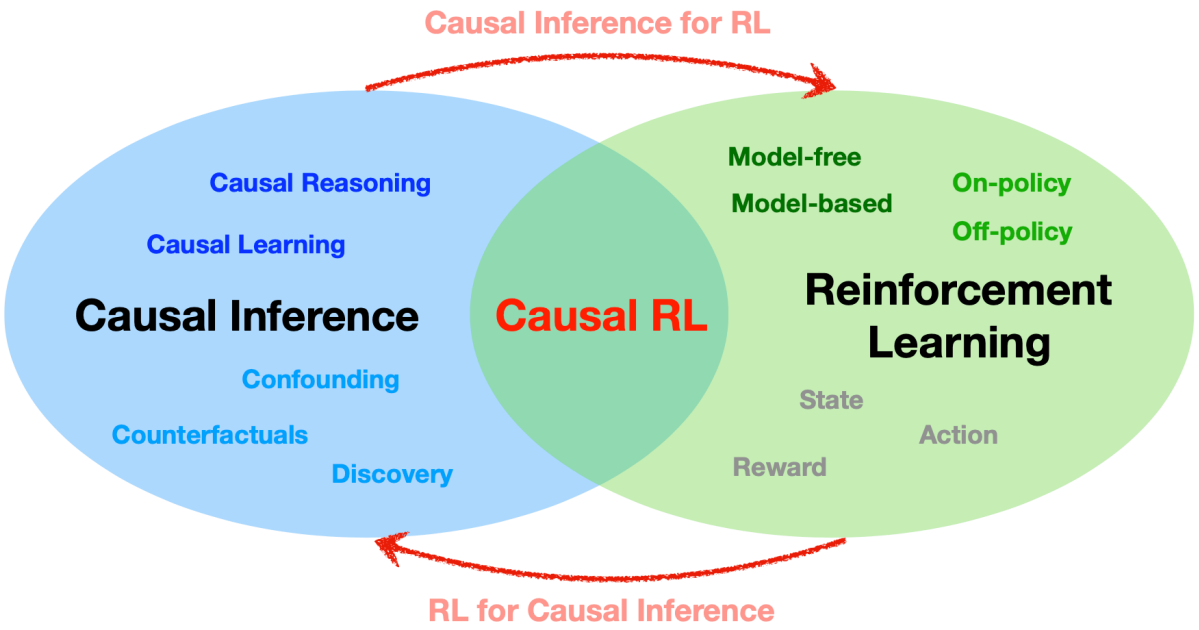
$\mathcal{M}_{test}$

Recent Trend II

Causal RL

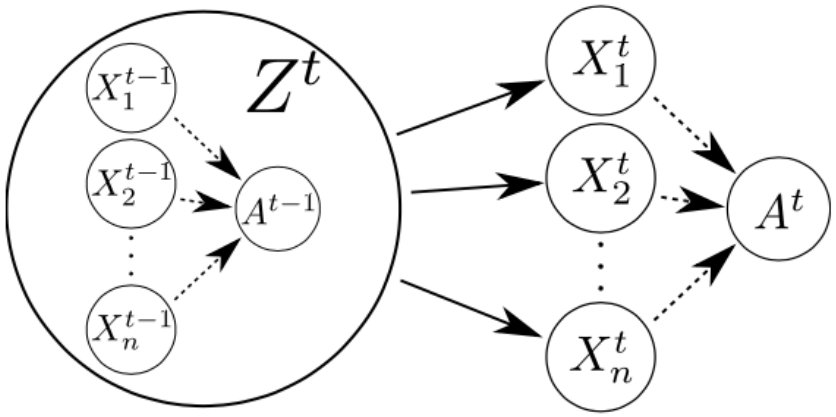
- RL is awesome at handling sample complexity and credit assignment
- Causal Inference (CI) is great at deducing cause-effect relationships across settings and conditions
- Can we have the best of both worlds? Yessss~!
- Simple solution:

Causal RL = CI + RL



Recent Trend II

Causal structure for causal RL

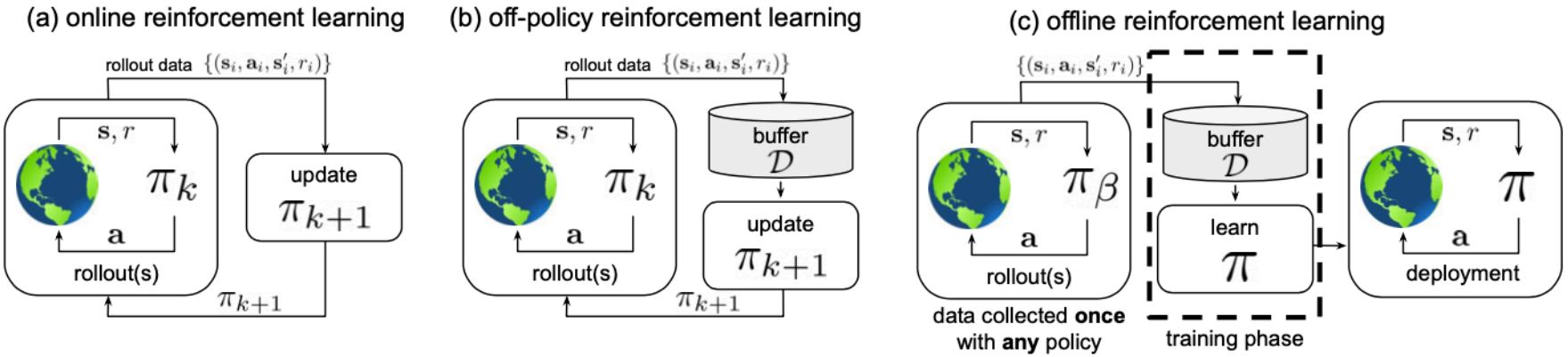


- The actions A^t are influenced by some information in the state observations X^t
- A **confounder** Z^t influences each state variable in X^t
- Some unknown subset of disentangled factors of X^t (**causes**) affect the actions, and rest (**nuisance variables**) do not
- **Intervention**:

$$P(X^t|do(X^t))$$

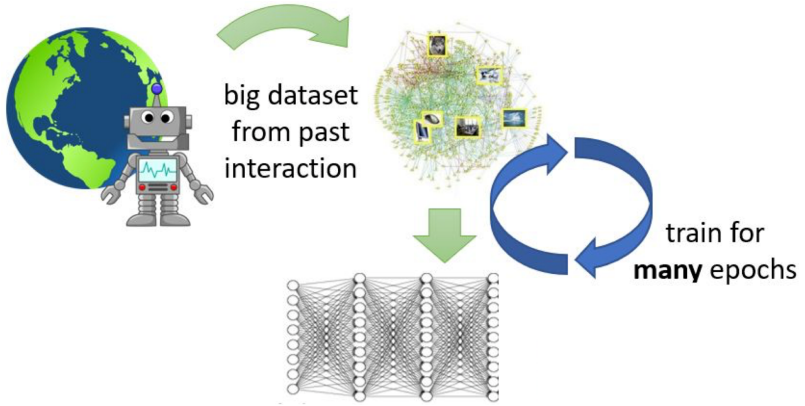
Recent Trend III

Offline RL



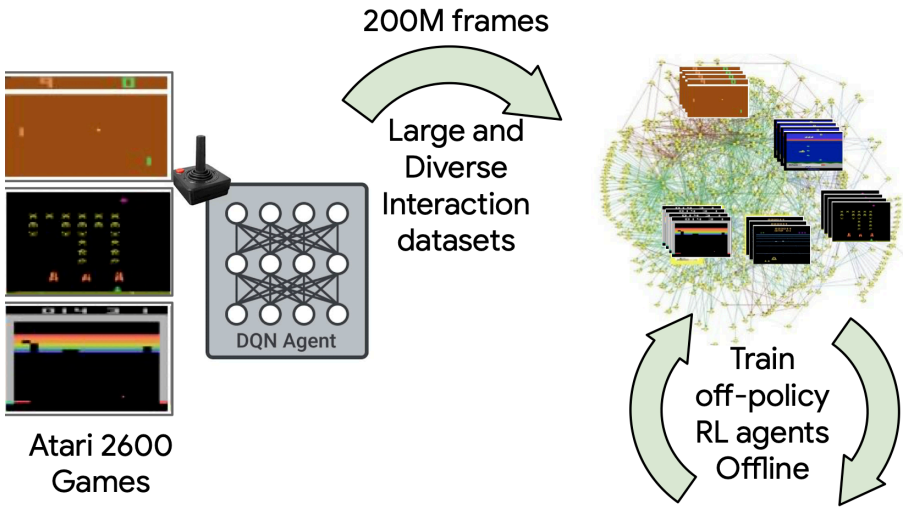
Reinforcement Learning with Online Interactions

Offline Reinforcement Learning



Recent Trend III

DQN replay dataset for offline RL



Thank You!

Any Questions?