535514: Reinforcement Learning

Lecture 0 — Logistics

Ping-Chun Hsieh

February 19, 2024

Course Info

- Graduate-level *seminar-like* course
  - Present a few classic or popular latest RL papers every week

- No textbook (but quite a lot of references will be provided). We will use *slides* for most of the time

- This course will be highly interactive!

- Throughout this course, we will primarily focus on the theoretical foundations of RL (70%), with some discussions on the practical aspects (30%)

Specifically, we focus on RL algorithms with theoretical guarantees in terms of:

Optimality and Convergence

Disclaimers!

1. There will be ***quite a few*** Theorems and Proofs…

2. You will also be asked to prove some key theorems in the written assignments

- Interestingly, most of the proofs require only **probability, calculus**, and **basic algebra**

Prerequisites

- Some math maturity:
  - Familiar with calculus and probability (measure-theoretic background not required)
  - Basic understanding of optimization would help
- Programming skills:
  - Python
  - Familiarity with Tensorflow or PyTorch would help

Logistics

- This course will be taught in English
  - Questions can be asked in either English or Mandarin
- Slides and assignments will be posted on E3
- Office hours: Mondays, 1pm-2pm @EC418 (starting on 2/26)
- TAs:
  - Kai-Jie Lin (林楷傑): kjl0508.sc10@nycu.edu.tw

Course Registration

- Default limit: 50 students
- Manual Registration: At most 15 students
  - Due to the limitations of human resource and my bandwidth, I would prefer to stick to this limit
  - For those who needs manual registration, please fill out the Google form: https://forms.gle/NEfwojGAdcno723m7
  - Submit your registration requests by 9pm this Thursday (2/22)
  - Decisions will be made this Friday (2/23)
  - You will NOT be able to drop the course after being manually added to this course
- Auditing is allowed (as long as there are seats available)
  - Send an email to the TAs if you need access to E3

Topics That Will Be Covered

1. Fundamentals: MDPs and planning (3 weeks)
2. Model-free RL: Policy optimization (6 weeks)
   - E.g.: PG, Actor-Critic, DPG, DDPG, Natural PG, TRPO, PPO, SAC…
   - We will also briefly discuss offline model-free RL
3. Model-free RL: Value-based approach (3 weeks)
   - E.g.: Q-learning, Sarsa, Double-Q, DQN, Distributional RL (C51), QR-DQN
4. Model-Based RL (1-2 weeks)
5. Imitation Learning and RLHF (2 weeks)
   - GAIL, IQ-Learn, etc
   - RLHF

Topics That Will *Not* Be Covered

- Hierarchical RL
- Multi-task RL
- Multi-objective RL
- Meta RL
- Multi-agent RL
- RLHF
- … and so on

Grading

- Warm-up Assignments: 35%
- Theory Project: 30%
  - Hackmd Blogpost: 20%
  - Reviews: 10%
- Team Implementation Project: 35%
  - Oral Presentation: 15%
  - Technical Report: 20%

Warm-Up Assignments

- Each homework will be a mixture of proofs and mini-programming tasks (roughly 100~200 lines of code)
  - HW1 (12%): MDPs, planning, and D4RL (Week 3)
  - HW2 (11%): Policy optimization and policy gradient (Week 6)
  - HW3 (12%): DPG, DDPG, TD3, TRPO, and PPO (Week 9)

- Main purposes:
  - For those not familiar with tensorflow/pytorch/OpenAI-gym: you will get a chance to quickly pick up these tools!
  - For those not familiar with theoretical analysis: you will get the flavor of this interesting and delightful task :)

Late-Submission Policy for Assignments

- You might be busy with your research, so we decide to run a "linear penalty scheduler"
  - Suppose the assignment is X days late
  - X in (0,1]: 1/6 of total score deducted
  - X in (1,2]: 2/6 of total score deducted
  - X in (2,3]: 3/6 of total score deducted
  - X in (3,4]: 4/6 of total score deducted
  - X in (4,5]: 5/6 of total score deducted
  - X > 5: No credit

Policy on Large Language Models (LLM)

- You can use LLMs (e.g., ChatGPT) to revise your reports
- However, please do NOT generate a report directly by LLMs
  - Based on the current ChatGPT, that would typically give you a report full of obvious errors

Yann LeCun

1 小時 ·

My unwavering opinion on current (auto-regressive) LLMs
1. They are useful as writing aids.
2. They are "reactive" & don't plan nor reason.
3. They make stuff up or retrieve stuff approximately.
4. That can be mitigated but not fixed by human feedback.

Project: *Offline RL Challenge*

How Much Data Needed in RL?

- Question: How much data is needed for a typical RL algorithm to solve Atari games?

How Much Data Needed in RL?

- Question: How much data is needed for a typical RL algorithm to solve Atari games?

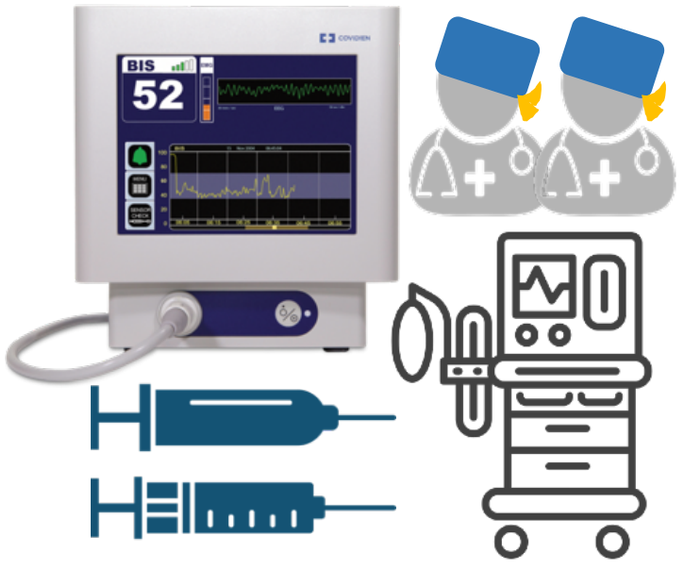
- RL: 100+ millions frames (~1000 hours of game time)
- Humans: Several hours is enough!

Fact: RL is known to require a large number of data samples

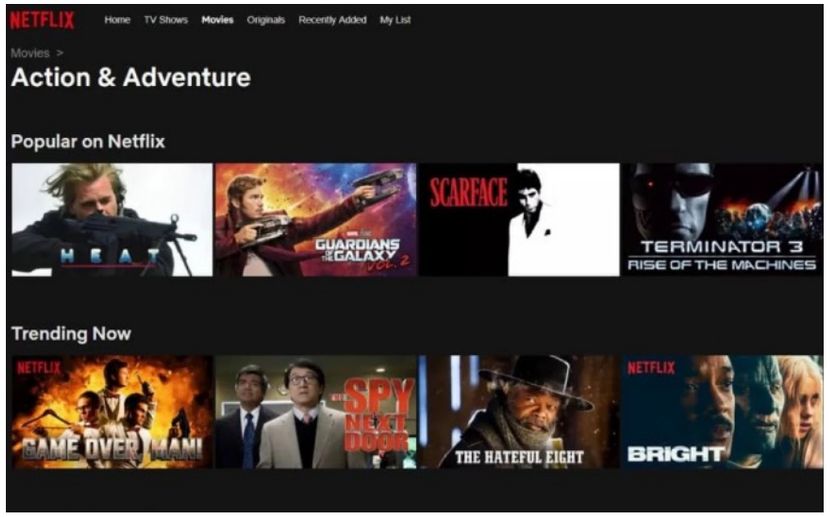
Data Collection Issue in Real-World Problems

Robotics

Healthcare

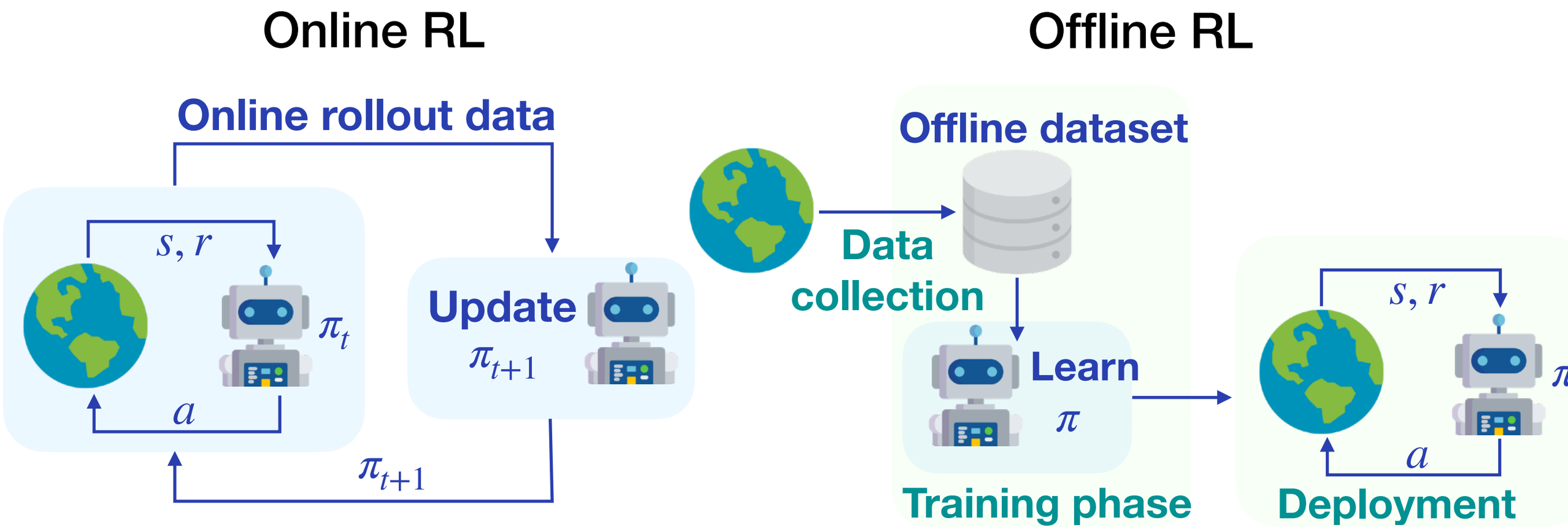


Recommender Systems



- RL deployment is challenging if data collection is either costly or dangerous

Online RL vs Offline RL



- Offline RL: Leverage pre-collected data to learn a control strategy, *with no online interaction*
- Offline RL has been one of the most important topics since 2020
- And it is still a growing research field in RL!

You will have the opportunity to join this interesting and practical RL field through this RL course!

"Offline RL Challenge!"

Offline RL Challenge

1. Theory project

- Write a reader-friendly digest for a recent paper on offline RL theory

2. Team implementation project

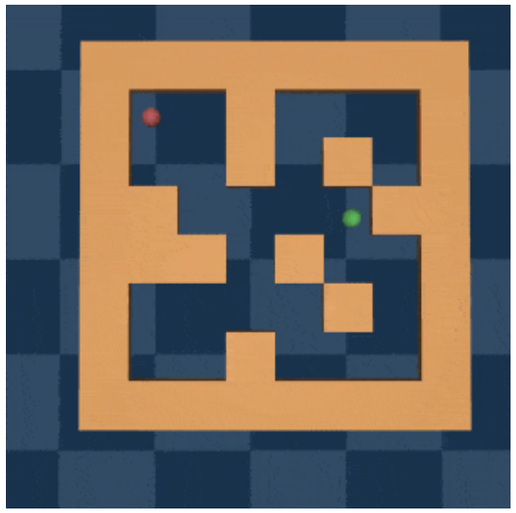
- Implement and improve an offline RL algorithm
- Test it on interesting and practical RL tasks

Part 1: Team Implementation Project

D4RL: A Popular Offline RL Benchmark

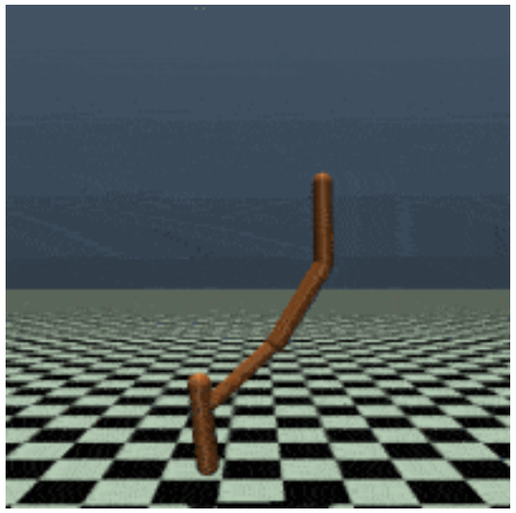
- D4RL: A popular dataset for offline RL across multiple domains

(Most widely used in RL research)

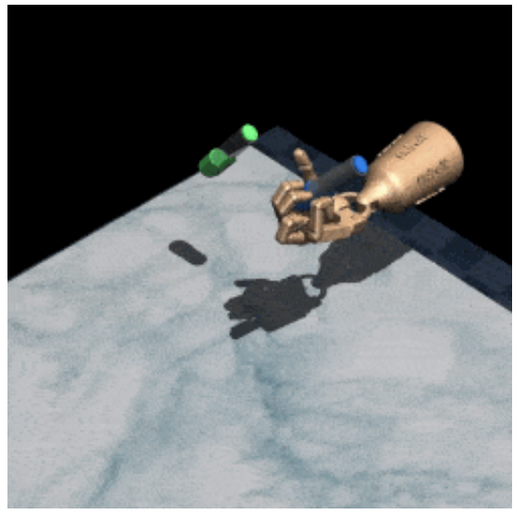


Maze2D

AntMaze

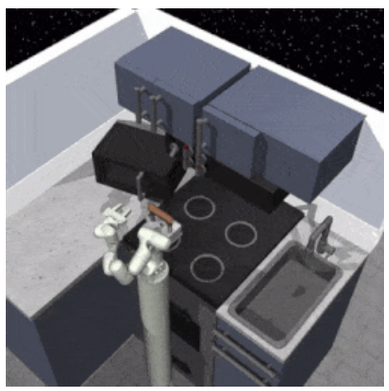


MuJoCo



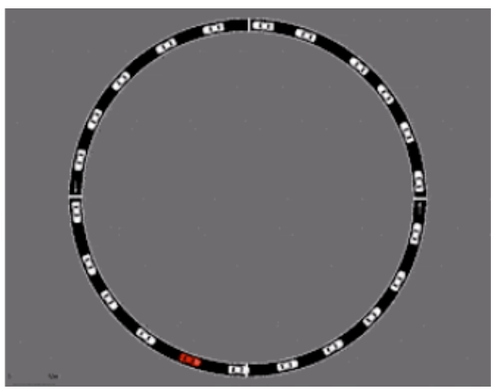
Adroit

(Not that widely used but still interesting)



FrankaKitchen

CARLA



Flow

“Collaborative” Offline RL Challenge

- This project will be done in groups of 3-4 students
- Select an Offline RL algorithm published in a top ML venue
  - E.g.: ICML, NeurIPS, ICLR, AAAI, IJCAI, UAI, AISTATS, KDD…
  - A list of recommended offline RL methods will be provided to you (see next few pages)
- Reproduce, improve, and evaluate your algorithm on D4RL
- Each team will deliver a presentation (posters or orals, TBD) and a technical report
- We will make all the reports, slides, and source code available to the students in this class via a shared Github repo

“Collaborative” Offline RL Challenge (Cont.)

- **3 Tasks to be achieved!**

  1. Reproduce: Get familiar with the algorithm and the open-source source code (if available), and then reproduce the experimental results on D4RL (e.g., MuJoCo, Maze2D, or AntMaze)

  2. Design: Improve the algorithm by adding / removing building blocks (examples in the next few pages) and handle model selection issue

  3. Let’s Shoot the Moon!: Crack the interesting and challenging “Adroit” tasks in D4RL

Step 1: Reproduce

- Get familiar with an open-source implementation (see next few pages)
- Reproduce the results and report them in a table

Example:

Task Name	BC	CQL	PBRL	SAC-10 (Reproduced)	EDAC (Paper)	EDAC-10 (Reproduced)	RORL (Ours)
halfcheetah-random	2.2±0.0	**31.3±3.5**	11.0±5.8	**29.0±1.5**	28.4±1.0	13.4 ± 1.1	28.5±0.8
halfcheetah-medium	43.2±0.6	46.9±0.4	57.9 ±1.5	64.9±1.3	**65.9±0.6**	64.1±1.1	**66.8±0.7**
halfcheetah-medium-expert	44.0±1.6	95.0±1.4	92.3±1.1	107.1±2.0	106.3±1.9	**107.2±1.0**	**107.8±1.1**
halfcheetah-medium-replay	37.6±2.1	45.3±0.3	45.1±8.0	**63.2±0.6**	61.3±1.9	60.1±0.3	**61.9±1.5**
halfcheetah-expert	91.8±1.5	97.3±1.1	92.4±1.7	104.9±0.9	**106.8±3.4**	104.0±0.8	**105.2±0.7**
hopper-random	3.7±0.6	5.3±0.6	**26.8±9.3**	25.9±9.6	25.3±10.4	16.9±10.1	**31.4±0.1**
hopper-medium	54.1±3.8	61.9±6.4	75.3±31.2	0.8±0.2	101.6±0.6	**103.6±0.2**	**104.8±0.1**
hopper-medium-expert	53.9±4.7	96.9±15.1	**110.8±0.8**	6.1±7.7	110.7±0.1	58.1±22.3	**112.7±0.2**
hopper-medium-replay	16.6±4.8	86.3±7.3	100.6±1.0	**102.9±0.9**	101.0±0.5	**102.8±0.3**	**102.8±0.5**
hopper-expert	107.7±9.7	106.5±9.1	**110.5±0.4**	1.1±0.5	110.1±0.1	77.0±43.9	**112.8±0.2**
walker2d-random	1.3±0.1	5.4±1.7	8.1±4.4	1.5±1.1	**16.6±7.0**	6.7±8.8	**21.4±0.2**
walker2d-medium	70.9±11.0	79.5±3.2	89.6±0.7	46.7±45.3	**92.5±0.8**	87.6±11.0	**102.4±1.4**
walker2d-medium-expert	90.1±13.2	109.1±0.2	110.1±0.3	**116.7±1.9**	114.7±0.9	115.4±0.5	**121.2±1.5**
walker2d-medium-replay	20.3±9.8	76.8±10.0	77.7±14.5	89.6±3.1	87.1±2.3	**94.0±1.2**	**90.4 ± 0.5**
walker2d-expert	108.7±0.2	109.3±0.1	108.3±0.3	1.2±0.7	**115.1±1.9**	57.8±55.7	**115.4 ± 0.5**
Average	49.7	70.2	74.4	50.8	**82.9**	71.2	**85.7**
Total	746.1	1052.8	1116.5	761.6	**1243.4**	1068.7	**1285.7**

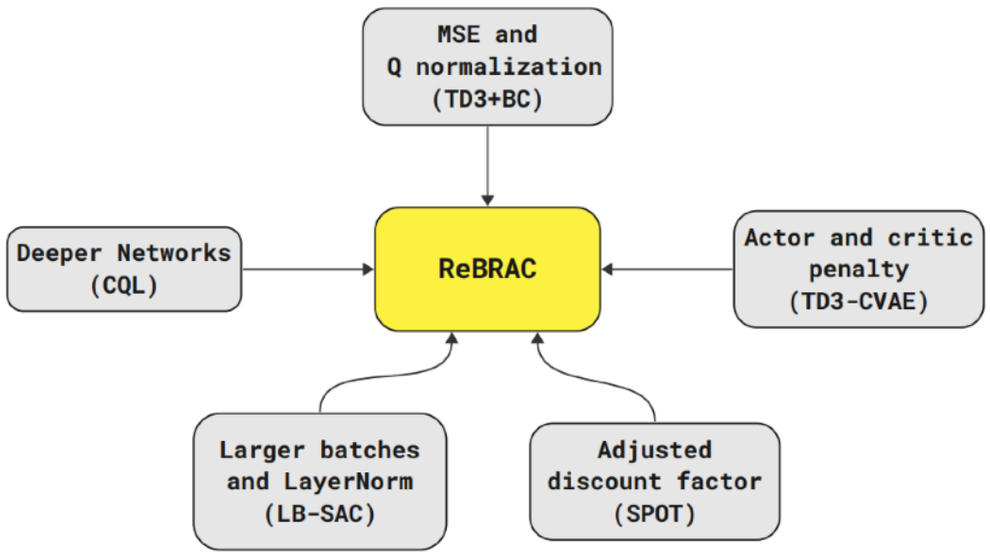
RORL: Robust Offline Reinforcement Learning via Conservative Smoothing (https://arxiv.org/pdf/2206.02829.pdf)

Step 2: Design

- Improve the algorithm by adding / removing blocks or tricks
- Ablation study

Example: ReBrac (NeurIPS 2023)

Seemingly minor design choices can have a huge impact!



Modification	TD3+BC	CQL	EDAC	MSG	CNF	LB-SAC	SAC-RND
Deeper networks	✗	✓	✓	✓	✓	✓	✓
Larger batches	✗	✗	✗	✗	✓	✓	✓
Layer Normalization	✗	✗	✗	✗	✗	✓	✓
Decoupled penalization	✗	✗	✗	✗	✗	✗	✓
Adjusted discount factor	✗	✗	✗	✗	✗	✗	✓

Ablation	Gym-MuJoCo	AntMaze	Adroit	All
TD3+BC, paper	-	27.3	0.0	-
TD3+BC, our	63.4	18.5	52.3	52.2
TD3+BC, tuned	71.8 (-10.9%)	27.9 (-62.9%)	53.5 (-25.9%)	58.3 (-19.2%)
TD3+BC w/ γ change	-	17.5 (-76.7%)	-	-
TD3+BC w/ LN	71.4 (-11.4%)	35.6 (-52.7%)	55.6 (-4.1%)	60.2 (-16.6%)
TD3+BC w/ large batch	14.4 (-82.1%)	0.0 (-100.0%)	1.6 (-97.2%)	7.9 (-89.0%)
TD3+BC w/ layer	71.2 (-11.6%)	44.1 (-41.4%)	56.4 (-2.7%)	61.9 (-14.2%)
ReBRAC w/o large batch	75.9 (-5.8%)	-	-	-
ReBRAC w large batch	-	41.0 (-45.6%)	55.4 (-4.6%)	-
ReBRAC w/o γ change	-	21.0 (-72.1%)	-	-
ReBRAC w/o LN	59.2 (-26.5%)	0.0 (-100.0%)	25.1 (-56.7%)	38.0 (-47.3%)
ReBRAC w/o layer	78.5 (-2.6%)	18.1 (-75.9%)	59.0 (+1.7%)	61.9 (-14.2%)
ReBRAC w/o actor penalty	22.8 (-71.7%)	0.1 (-99.8%)	0.0 (-100.0%)	11.4 (-84.2%)
ReBRAC w/o critic penalty	81.1 (+0.6%)	72.2 (-4.1%)	56.9 (-1.8%)	71.5 (-0.9%)
ReBRAC w/o decoupling	79.8 (-0.9%)	76.9 (+2.1%)	56.7 (-2.2%)	71.6 (-0.8%)
ReBRAC	80.6	75.3	58.0	72.2

Revisiting the Minimalist Approach to Offline Reinforcement Learning (https://arxiv.org/abs/2305.09836)

Another Example of Ablation Study: Rainbow DQN

Rainbow: Combining Improvements in Deep Reinforcement Learning [AAAI 2018]

Matteo Hessel DeepMind
Joseph Modayil DeepMind
Hado van Hasselt DeepMind
Tom Schaul DeepMind
Georg Ostrovski DeepMind
Will Dabney DeepMind
Dan Horgan DeepMind
Bilal Piot DeepMind
Mohammad Azar DeepMind
David Silver DeepMind

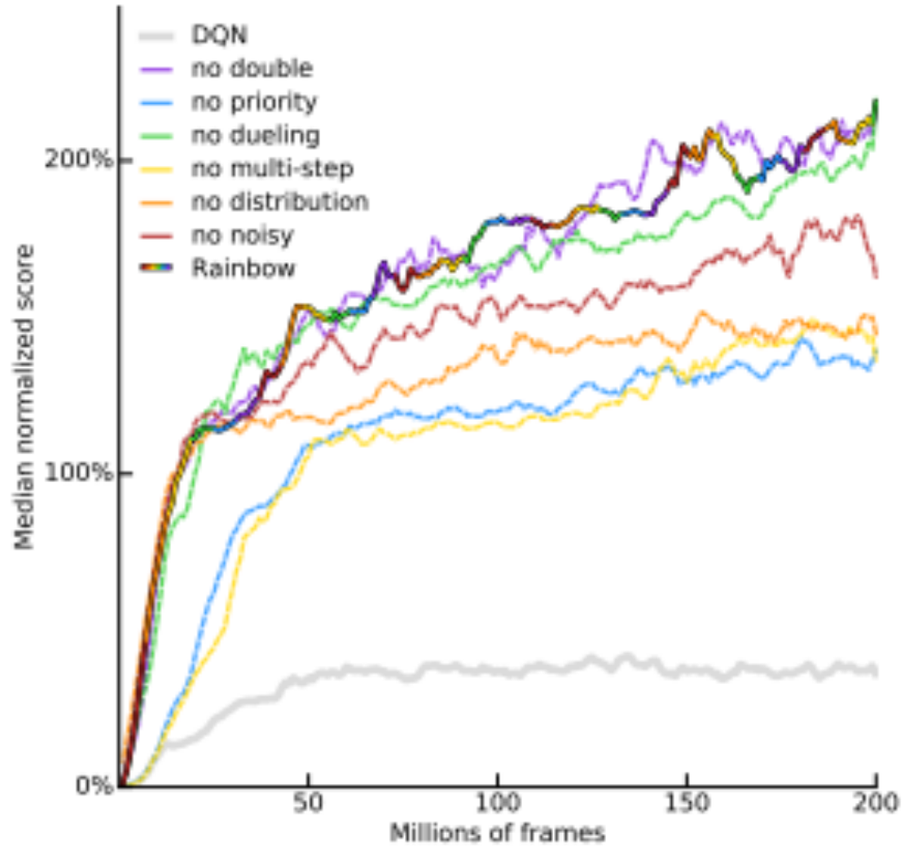
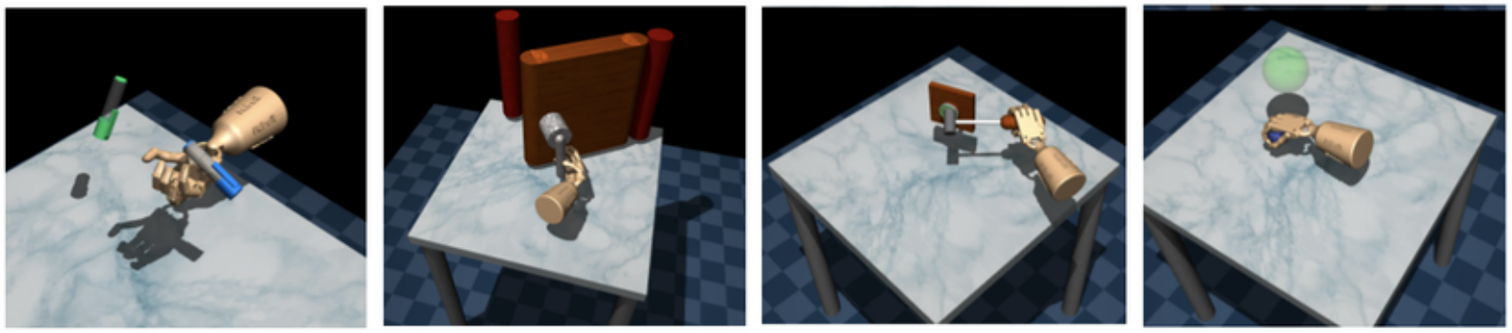


Figure 3: **Median human-normalized performance** across 57 Atari games, as a function of time. We compare our integrated agent (rainbow-colored) to DQN (gray) and to six different ablations (dashed lines). Curves are smoothed with a moving average over 5 points.

…Rainbow integrates several different ideas into a single agent, we conducted additional experiments to understand the contribution of the various components… we performed ablation studies. *In each ablation, we removed one component from the full Rainbow combination…*

Step 3: Tackle the Adroit Tasks!

Let's tackle this interesting robot control problem together!



Pen Door Hammer Relocate

Most existing methods can only get nearly 0 normalized score on these tasks

Task Name	BC	TD3+BC	IQL	CQL	SAC-RND	ReBRAC, our
pen-human	34.4	81.8 ± 14.9	81.5 ± 17.5	37.5	5.6 ± 5.8	**103.5** ± 14.1
pen-cloned	56.9	61.4 ± 19.3	77.2 ± 17.7	39.2	2.5 ± 6.1	**91.8** ± 21.7
pen-expert	85.1	146.0 ± 7.3	133.6 ± 16.0	107.0	45.4 ± 22.9	**154.1** ± 5.4
door-human	0.5	-0.1 ± 0.0	3.1 ± 2.0	**9.9**	0.0 ± 0.1	0.0 ± 0.0
door-cloned	-0.1	0.1 ± 0.6	0.8 ± 1.0	0.4	0.2 ± 0.8	**1.1** ± 2.6
door-expert	34.9	84.6 ± 44.5	**105.3** ± 2.8	101.5	73.6 ± 26.7	104.6 ± 2.4
hammer-human	1.5	0.4 ± 0.4	2.5 ± 1.9	**4.4**	-0.1 ± 0.1	0.2 ± 0.2
hammer-cloned	0.8	0.8 ± 0.7	1.1 ± 0.5	2.1	0.1 ± 0.4	**6.7** ± 3.7
hammer-expert	125.6	117.0 ± 30.9	129.6 ± 0.5	86.7	24.8 ± 39.4	**133.8** ± 0.7
relocate-human	0.0	-0.2 ± 0.0	0.1 ± 0.1	**0.2**	0.0 ± 0.0	0.0 ± 0.0
relocate-cloned	-0.1	-0.1 ± 0.1	0.2 ± 0.4	-0.1	0.0 ± 0.0	**0.9** ± 1.6
relocate-expert	101.3	**107.3** ± 1.6	106.5 ± 2.5	95.0	3.4 ± 4.5	106.6 ± 3.2
Average w/o expert	11.7	18.0	20.8	11.7	1.0	**25.5**
Average	36.7	49.9	53.4	40.3	12.9	**58.6**

A List of *Model-Based* Offline RL Methods

- **MOPO**: Model-based Offline Policy Optimization
- **COMBO**: Conservative Offline Model-Based Policy Optimization
- **RAMBO**: Robust Adversarial Model-Based Offline Reinforcement Learning
- **MOBILE**: Model-Bellman Inconsistancy Penalized Offline Reinforcement Learning
- **ARMOR**: A Model-based Framework for Improving Arbitrary Baseline Policies with Offline Data

- You can start from the OfflineRL-Kit open-source implementation:
  - https://github.com/yihaosun1124/OfflineRL-Kit

A List of *Model-Free* Offline RL Methods

- **BCQ**: Batch Constrained Q-learning
- **BEAR**: Bootstrapping Error Accumulation Reduction
- **CQL**: Conservative Q-Learning
- **AWAC**: Advantage Weighted Actor-Critic
- **TD3+BC**: A Minimalist Approach to Offline RL
- **IQL**: Implicit Q-Learning
- **DT**: Decision Transformer
- **ReBRAC**: Revisiting the Minimalist Approach to Offline RL
- **SAC-N**: Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble
- **EDAC**: Uncertainty-Based Offline Reinforcement Learning with Diversified Q-Ensemble

- You can start from the CORL open-source implementation:
  - https://github.com/corl-team/CORL

Some Remarks

- The final project report will be graded
  - Mainly by “how much insight and improvement made on the base algorithm”
  - Only minimally by “how good the scores on D4RL are”
- Therefore, it does not matter much whether you choose a strong or a not-that-strong offline RL method to begin with
  - Choosing a state-of-the-art method may indicate little improvement possible

Team Project: Some Milestones (Tentative)

- Team Project Milestones:
  - Find your team members: by 3/15 (Week 4)
  - Submit your algorithm preference: by 4/2 (Week 7)
    - We will finalize the topics by 4/5
  - 1st Team-TA Meetup: 5/1-5/3 (Week 11)
  - 2nd Team-TA Meetup: 5/27-5/29 (Week 15)
  - Poster/Oral presentations: 6/11-6/13 (2.5-hour sessions, TBD)
  - Submission of technical report: by 6/17

Part 2: Theory Project

Offline RL Theory Project

- Goal: Explain an offline RL theory paper by writing a blogpost

Step 0. Each student select a theory paper

Step 1. Fully digest the analysis and explain them using your own words

Step 2. Identify all the assumptions and explain why they are needed

Step 3. Try your best to make a critique on the paper:

- What are the main contributions of this paper?
- Are the assumptions reasonable or too strong?
- Why are the results interesting?
- What are the main intuitions behind the theorems and the proofs?
- What are the potential weaknesses of the paper?
- If you were the author, would you design something different or similar?

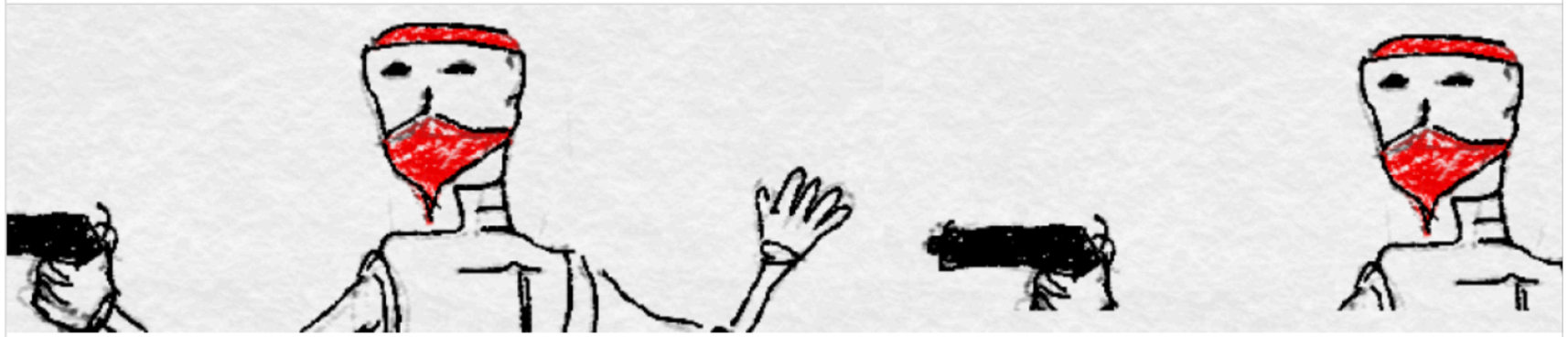
Offline RL Theory Project (Cont.)

Step 4. Compile your thoughts into a self-contained hackmd blogpost

- You shall explain the algorithm, theorems, and proofs by using your own words
- Markdown language supports LaTeX-like syntax, which would make it easier for you to type math symbols and equations
- No page limit. Your hackmd note will be graded based on quality, not quantity.

IMPORTANT: Please use your own words. Do not copy the paper verbatim. (解釋而非轉述)

Exemplar Blogs on RL Theory



Stochastic Linear Bandits and UCB

October 19, 2016 18 Comments

Recall that in the adversarial contextual K-action bandit problem, at the beginning of each round t a context $c_t \in \mathrm{C}$ is observed. The idea is that the context c_t may help the learner to choose a better action. This led us to change the benchmark in the definition of regret. In this post we start with reviewing how contextual bandit problems can be defined in the stochastic setting. We use this setting to motivate the introduction of stochastic linear bandits, a fascinatingly rich model with much structure and which will be the topic of a few of the next posts. Besides defining stochastic linear bandits we also cover how UCB can be generalized to this setting.

Stochastic contextual bandits

In the standard K-action stochastic contextual bandit problem at the beginning of round t the learner observes a context $C_t \in \mathrm{C}$. The context may or may not be random. Next, the learner chooses its action $A_t \in [K]$ based on the information available. So far these is no difference to the adversarial setting. The difference comes from the assumption that the reward X_t which is incurred satisfies

$$X_t = r(C_t, A_t) + \eta_t \,,$$

where $r : \mathrm{C} \times [K] \to \mathbb{R}$ is the so-called *reward function*, which is unknown to the learner, while η_t is random noise.

Search...

- About
- Download book

Recent Posts

- Bayesian/minimax duality for adversarial bandits
- The variance of Exp3
- First order bounds for k-armed adversarial bandits
- Bandit Algorithms Book
- Bandit tutorial slides and update on book

Recent Comments

- Tor Lattimore on Ellipsoidal Confidence Sets for Least-Squares Estimators
- Zeyad on Ellipsoidal Confidence Sets for Least-Squares Estimators
- Tiancheng Yu on Bayesian/minimax duality for adversarial bandits
- Claire on Ellipsoidal Confidence Sets for Least-Squares Estimators
- Tor Lattimore on Ellipsoidal Confidence Sets for Least-Squares Estimators

Csaba Sepesvari

Tor Lattimore

https://banditalgs.com/2016/10/19/stochastic-linear-bandits/

Exemplar Blogs on RL Theory (Cont.)

MARL Theory

Homepage

Recent Progresses in Multi-Agent RL Theory

Nov 14, 2021 • Yu Bai, Chi Jin

Reinforcement learning (RL) has made substantial empirical progresses in solving hard AI challenges in the past few years. A big portion of these progresses—Go, Dota 2, Starcraft, economic simulation, social behavior learning, and so on—come from *multi-agent RL*, that is, sequential decision making involving more than one agents. While the theoretical study of (single-agent) RL has a long history and a vastly growing recent interest, multi-agent RL theory is arguably a newer and less developed field, with its own unique challenges and opportunities that we feel very excited about.

In this extended blog post, we present a brief overview of the basics of multi-agent RL theory, along with some recent theoretical developments in the past few years. We will focus on learning Markov games, and cover the basic formulations, learning goals, planning algorithms, as well as recent advances in sample-efficient learning algorithms under the interactive (exploration) setting.

The majority of this blog post will assume familiarity with the basics of RL theory in the single-agent setting (for example, materials in this fantastic book). We will focus on "what is different" when it comes to multi-agent, and discuss the various recent developments and opportunities therein.

Chi Jin

Yu Bai

https://yubai.org/blog/marl_theory.html

Offline RL Theory Project (Cont.)

- Theory paper = A paper with at least 1 non-trivial theorem
- 1 student per team
- A list of recommended papers will be posted on E3 and would be updated continually
- Feel free to recommend a paper that is NOT on the list, but is of the most interest to you

Theory Project: Peer Discussions!

1. Research is about

"presenting your novel ideas and shaking existing beliefs of others"!

2. Research is also about

"making factual critiques to helping people improve their works"

- We'd like to encourage more peer discussions!

(Figure source: *Chuka Ichiban*!)

Theory Project: Peer Discussions!

- Let's do "peer reviews"!

- Each student serves as a reviewer of 2 hackmd notes
  - 1 assigned, the other 1 is selected by yourself
  - Reviewing = provide review comments and ask questions
  - Review template will be provided
  - Reviews can be written either in English or Mandarin
  - Bonus points will be given if you review more than 2 hackmd notes

- All the reviews and hackmd notes will be made public to all the students in our class (with reviewer anonymized)

Theory Project: Some Milestones (Tentative)

- Theory project milestones:
  - Select a paper: by 4/2 (Week 7)
  - Submit your hackmd note: by 5/15 (Week 13)
  - Peer reviews: 5/16-5/27

Shall I Take This Course?

- This course will be a good fit:
  - if you want to understand RL from a theoretical viewpoint
  - if you'd like to search for fundamental research topics on RL
  - if you kind of enjoy mathematical analysis
  - if you are comfortable with research-oriented projects
- This course would NOT be a good fit:
  - if you expect to learn hands-on implementation / engineering tricks (which are VERY important to deep RL community)
  - if you are a passionate practitioner who focus on making things to work
  - if you prefer application-oriented RL projects

Your “Homework” For Today

- Check E3 and see if you have access to the material

- Team Project: Start forming your project teams

- Manual Registration: For those who need manual registration, please fill out the Google form: https://forms.gle/NEfwojGAdcno723m7

Useful References

Reinforcement Learning

- [SB] Richard Sutton and Andrew Barto, *Reinforcement Learning: An Introduction*, 2nd edition, 2019.
- [AJK] Alekh Agarwal, Nan Jiang Sham M. Kakade, *Reinforcement Learning: Theory and Algorithms*, 2020 (https://rltheorybook.github.io/rl_monograph_AJK.pdf)
- [KWW] Mykel Kochenderfer, Tim Wheeler, Kyle Wray, *Algorithms for Decision Making*, 2020 (https://algorithmsbook.com/)

Optimization

- [BCN] Léon Bottou, Frank E. Curtis, and Jorge Nocedal, *Optimization Methods for Large-Scale Machine Learning* (https://arxiv.org/abs/1606.04838)
- [NW] Jorge Nocedal and Stephen Wright, *Numerical optimization, 2006*

Any Questions?