Deep Reinforcement Learning Course Project

2024 Spring

April 2024

0 Overview

This is the documentation for the **course project of Deep Reinforcement Learning in 2024 Spring**. You are required to finish a group project related to some sub-field of deep reinforcement learning. We have provided some tracks for you to explore, and we are also open to your own super cool ideas beyond our proposed tracks (you can choose the Open Track if you want to work on your own idea).

In this documentation, we will first explain our logistics (1), resources (2) and timeline (3), and **the requirements of different research tracks are attached at the end of this documentation**.

Good luck and enjoy your experience in deep reinforcement learning research! :)

1 Logistics

1.1 Rules

- This is a group project and each group consists of **2-4 students**. Please fill in this google sheet after you have found your group. **No later than April 24th!**
- You will need to submit a 1-page project proposal, some milestone materials and a report to summarize your project (3-page minimum and 8-page maximum).
- This project weighs 35% in the whole score of this course. **The project proposal, oral /poster session and the report occupy 5%, 15% and 15% respectively.**
- The milestone materials, which can be of any format (e.g., report/demo video/live demo/website/slides) that can demonstrate your results, are only used for selecting oral presentations. You should submit your milestone material some time before the presentation day (e.g., Week 15).
- Poster and oral sessions are expected to be in English. Requirements on the presentation's duration will be announced later.
- We expect each group member to contribute to the project genuinely; contributions should be listed clearly.
- We expect to host the presentation day sometime near the end of the semester, before the exam weeks (e.g., Week 16).

1.2 Report Format

Please use the provided NeurIPS latex template to complete your report.
Your report should include the following sections:

- A short abstract that summarizes your work;
- An introduction section that explains your motivation, proposed method, and results;
- A related work section that briefly surveys the past literature;
- A method section that describes how you solve the problem in detail;
- A results section that demonstrates the performance of your proposed method;
- (Optional) Additional ablative experiments to understand your proposed method;
- (Optional) A discussion section to elaborate on your thoughts on your project.

Here is a detailed list of how we will evaluate your report:

- Writing, format, and style of your report: 10%;
- Literature review: 10%;
- Novelty (if the project is your own idea) or depth that you investigate the idea (if you select one of the ideas we provided): 20%. You can also make your own variation over the idea we provided, we will evaluate this term in a case-by-case fashion;
- Results, how well does your solution solve the task, and how impressive are the final results: 50%;
- Logical clarity, how well your method and your results are delivered, and whether it's clear to draw the conclusion you provided from your results: 10%;

1.3 Poster Format

Your poster should include the following sections:

- A short abstract that summarizes your work;
- A motivation section that briefly explains why you want to work on this idea;
- A problem formulation section to introduce your task setting in an RL context;
- A method section to demonstrate your key improvement/contribution over existing methods/dataset/simulator, etc;
- A result section that shows your main experiment figures, curves and tables.

Here is a detailed list of how we will evaluate your poster:

- Visual Appearance of your poster: 30%;
- Clarity of your method and results: 30%;
- Faithfulness and integrity compared to your report: 30%;
- Guidance or inspiration for future research: 10%.

You can design your poster through LaTeX (template), PowerPoint (template), or any image editing tool you find convenient. Only the final figure (in png, jpg, pdf, etc.) is required. **The aspect ratio of your poster should be 16:9.** Those selected for oral don't need to provide a poster.

1.4 Oral Format

Your oral presentation should include the following sections:

- A motivation section that briefly explains why you want to work on this idea;
- A problem formulation section to introduce your task setting in an RL context;
- A method section to demonstrate your key improvement/contribution over existing methods/dataset/simulator, etc;
- A result section that shows your main experiment figures, curves and tables.

Here is a detailed list of how we will evaluate your presentation:

- Clarity of your oral presentation: 30%;
- Visual appearance of your slides: 20%;
- Literature review: 10%;
- Presentation of methods and results: 30%;
- Guidance or inspiration for future research: 10%.

Note: Results are best viewed with **comparison videos**, e.g., your RL agent successfully wins a game while others cannot.

1.5 Tips

- Properly identify the scope of your research problem. Something like "Applications of Deep RL" is too broad, and "PPO speedup on a 3090 Ti GPU" is probably too narrow;
- Finding a small number of high-quality reference papers will save you a lot of time. It is a good idea to start with top-tier journals or conference papers with high citations;
- Start from a working/high-star repo in GitHub. RL is fragile; always start with something that already works;
- Rehearse your talk!
- If your project is super cool, please consider extending it and submitting it to an AI conference.

2 Resources

2.1 RL Algorithms Implementation

- Stable Baselines3
- Tianshou
- ElegantRL
- Spinningup
- RLlib
- Dopamine

2.2 Popular RL Simulators & Environments

Simulators:

- MuJoCo
- PyBullet
- Isaac Gym
- Sapien
- Gazebo
- SMAC
- PlasticineLab
- TaiChi Lang

Environments:

- Gymnasium(Openai Gym)
- DeepMind Control Suite
- Meta-World
- IsaacGymEnvs
- RLBench
- robosuite
- HumanoidBench

2.3 Useful tools:

- WandB
- TensorBoard
- Hydra

2.4 Reference Resources

- arXiv: search for relevant categories or keywords
- Google Scholar: search for relevant keywords or authors
- NeurIPS/ICML/ICLR: proceedings of top AI conferences
- Websites of research labs or organizations, such as OpenAI, Google DeepMind, etc.

3 Timeline

Week 8:

- Project starts!
- Team up (fill in the google sheet in one week, you are not required to decide your track at that time)
- Documentation released

Week 10:

- 1-page project proposal
- Idea selected or idea proposed
- Team members and your track
- [redacted]
- Key literature survey (list 3 key papers about your selected idea)

Week 15:

- Mid of this week (Wednesday): Milestone materials due
- End of this week (Friday): Notification of oral and poster

Week 16:

- First 45 mins: Oral presentations
- Second 45 mins: Poster session

After Week 16:

- Final report due (refer to Section 1.2 for requirements)
- Deadline will be announced later (will try to avoid the final exam weeks).

Standard RL Track

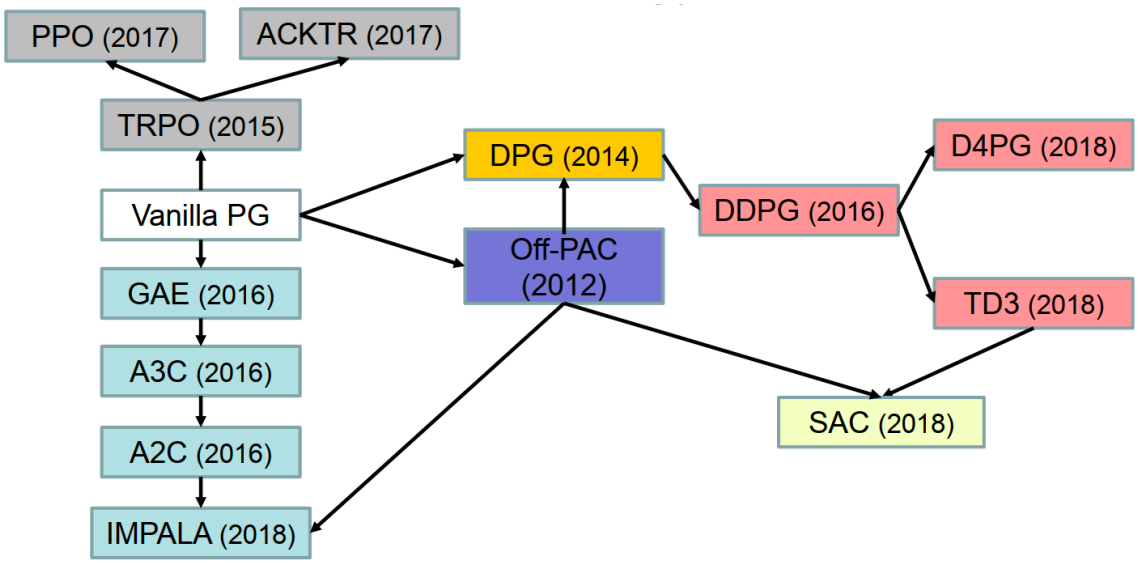

Figure 1: **Advanced Standard RL Approaches**: development pathways of classic RL algorithms including the SOTA methods we require.

1 Project Description

Stated-based Reinforcement Learning offers two distinct paths for exploration and advancement, each addressing specific aspects of RL challenges. The first path involves enhancing the Proximal Policy Optimization (PPO) [RL1] algorithm, incorporating insights and methodologies from recent developments like Muesli [RL2]. The second path focuses on improving Actor-Critic methods, specifically through Twin Delayed Deep Deterministic policy gradient (TD3) [RL3] and Soft Actor-Critic (SAC) [RL4], with innovations highlighted in BEE [RL5] and ACE [RL6] algorithms.

Each team should start with a thorough literature review of their chosen path, replicate the baseline models, and then propose novel techniques to enhance the selected RL framework's efficiency and effectiveness.

Notice: Despite years of efforts to refine PPO/SAC, finding strategies that markedly surpass their performance remains difficult. Therefore, while both paths in Standard RL may not demand high computational resources, attempts to improve upon them could present considerable challenges.

2 Requirements

In this section, we will provide a reference technology roadmap for your project.

2.1 Understanding Basic Concepts and Algorithms

Provide a comprehensive literature review on standard RL, identify current challenges, and decide whether to begin your project with TD3/SAC or PPO. If you want to start your project with TD3 [RL3] / SAC [RL4], provide a concise overview of the technical aspects of TD3 /

SAC, as well as one of BEE [RL5] and ACE [RL6]. Alternatively, if you choose to start with PPO, offer a brief introduction to the technical details of PPO [RL1] and Muesli [RL2].

2.2 Running Baselines

For this project, you are required to conduct experiments in the Mujoco [RL7], which is a widely used and relatively easy benchmark. You should reproduce the results in the paper in the environment **Halfcheetah**, **Walker2d** and **Ant** environments.

2.3 Algorithm Design

Enhance the algorithm's performance on the chosen tasks(you can add more tasks if you want). You should compare your algorithm with our proposed baselines and an improvement on any prospective(e.g. final performance, sample efficiency) over the chosen baseline is expected.

Of course, you are also encouraged to challenge the new SOTA BEE/ACE and Muesli in the final performance, sample efficiency, robustness for random seeds, etc. If you successfully beat them, congratulations! You can try to submit your project to a top conference!

2.4 Potential Directions

If you are interested in this project but have no idea where to start, might as well have a try at these directions:

- **Attempt to reproduce SOTA algorithms**: Reproducing SOTA algorithms represents a foundational step in pioneering new research. Algorithms such as BEE [RL5], ACE [RL6], and Muesli [RL2] have not been open-sourced. Begin your project by attempting to reconstruct these algorithms based on published results and methodologies.
- **Balance between exploration and exploitation**: Balancing exploration and exploitation remains a pivotal challenge in reinforcement learning. Strategies for exploration can broadly be divided into two categories. The first, uncertainty-oriented exploration, utilizes techniques like the upper confidence bound (UCB) to navigate based on uncertainty in value estimates [RL8, RL9, RL10, RL11]. The second category, intrinsic motivation-oriented exploration, encourages exploration through intrinsic rewards derived from prediction errors or state novelty [RL12,RL13,RL14,RL15]. You are encouraged to integrate some of the methods into your algorithm or propose your own idea.
- **Causal Reinforcement Learning**: Recently, recognizing the substantial capabilities of causality in addressing data inefficiency and interpretability challenges within RL, there has been a surge of research in the domain of causal reinforcement learning [RL16,RL17]. More information and related works can be found on the website. You can try to add causal inference to your RL algorithm.

Note: After implementing your own ideas/tricks, you can always first try the **Halfcheetah** environment (a simpler environment) to quickly verify whether your implementation is correct!

3 About Grading

We provide a reference grading policy for this project as follows:

- **[60 pts]**: Report the baseline results, and try a trick that others have implemented but failed.

- **[80 pts]**:
  - Report the baseline results and successfully replicate one of the BEE/ACE/Muesli algorithms.
  - Report the baseline results, conduct thorough experiments on all the provided directions, and analyze the results in detail (it can be okay if all of them fail).
  - Report the baseline results, propose and implement your own idea, and analyze the results in detail (it can be okay if it fails).
- **[100 pts]**: Make a great improvement on TD3, SAC or PPO.
- **Try conference submission**: Beat BEE, ACE or Museli in performance.

4 Contact

If you have any questions, please feel free to contact our TA Guowei Xu via WeChat or email: xgw23@mails.tsinghua.edu.cn.

RL Reference

[RL1] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[RL2] Matteo Hessel, Ivo Danihelka, Fabio Viola, Arthur Guez, Simon Schmitt, Laurent Sifre, Theophane Weber, David Silver, and Hado van Hasselt. Muesli: Combining improvements in policy optimization, 2022.

[RL3] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods, 2018.

[RL4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, 2018.

[RL5] Tianying Ji, Yu Luo, Fuchun Sun, Xianyuan Zhan, Jianwei Zhang, and Huazhe Xu. Seizing serendipity: Exploiting the value of past success in off-policy actor-critic. *arXiv preprint arXiv:2306.02865*, 2023.

[RL6] Tianying Ji, Yongyuan Liang, Yan Zeng, Yu Luo, Guowei Xu, Jiawei Guo, Ruijie Zheng, Furong Huang, Fuchun Sun, and Huazhe Xu. Ace : Off-policy actor-critic with causality-aware entropy regularization, 2024.

[RL7] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[RL8] Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Emilie Kaufmann, Edouard Leurent, and Michal Valko. Fast active learning for pure exploration in reinforcement learning. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 7599–7608. PMLR, 2021.

[RL9] Pierre Ménard, Omar Darwiche Domingues, Xuedong Shang, and Michal Valko. UCB momentum q-learning: Correcting the bias without forgetting. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 7609–7618. PMLR, 2021.

[RL10] Emilie Kaufmann, Pierre Ménard, Omar Darwiche Domingues, Anders Jonsson, Edouard Leurent, and Michal Valko. Adaptive reward-free exploration. In *ALT*, volume 132 of *Proceedings of Machine Learning Research*, pages 865–891. PMLR, 2021.

[RL11] Xiyao Wang, Ruijie Zheng, Yanchao Sun, Ruonan Jia, Wichayaporn Wongkamjan, Huazhe Xu, and Furong Huang. Coplanner: Plan to roll out conservatively but to explore optimistically for model-based rl, 2023.

[RL12] Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *ICML*, volume 119 of *Proceedings of Machine Learning Research*, pages 8583–8592. PMLR, 2020.

[RL13] Adrià Puigdomènech Badia, Pablo Sprechmann, Alex Vitvitskyi, Zhaohan Daniel Guo, Bilal Piot, Steven Kapturowski, Olivier Tieleman, Martín Arjovsky, Alexander Pritzel, Andrew Bolt, and Charles Blundell. Never give up: Learning directed exploration strategies. In *ICLR*. OpenReview.net, 2020.

[RL14] Mirco Mutti, Riccardo De Santi, and Marcello Restelli. The importance of non-markovianity in maximum state entropy exploration. In *ICML*, volume 162 of *Proceedings of Machine Learning Research*, pages 16223–16239. PMLR, 2022.

[RL15] Qisong Yang and Matthijs T. J. Spaan. CEM: constrained entropy maximization for task-agnostic safe exploration. In *AAAI*, pages 10798–10806. AAAI Press, 2023.

[RL16] Samuel J Gershman. Reinforcement learning and causal models. *The Oxford handbook of causal reasoning*, 1:295, 2017.

[RL17] James Bannon, Brad Windsor, Wenbo Song, and Tao Li. Causality and batch reinforcement learning: Complementary approaches to planning in unknown domains. *arXiv preprint arXiv:2006.02579*, 2020.

Offline RL Track

1 Project Description

Offline Reinforcement Learning [ORL1] aims to learn a policy from a fixed dataset of previously collected data, without the need for online data collection. This is a challenging problem, as the data may be biased, suboptimal, or not fully representative of the environment.

While traditionally offline RL studies have focused on improving the learning algorithm given a fixed dataset, the offline RL paradigm can also be viewed as a special case of supervised learning problem, of which the selection of the dataset is crucial. Inspired by the recent attempt to optimize the dataset, instead of the learning algorithm in the field of multimodal learning [ORL2], we propose a new track for the offline RL challenge, where the goal is to optimize the dataset for offline RL.

2 Requirements

In this section, we will provide a reference technology roadmap for your project.

2.1 Understanding Basic Concepts and Algorithms

Provide a comprehensive literature review on offline RL, identify current challenges, and give a brief introduction to the technical details of the offline RL algorithms, including BCQ [ORL3], CQL [ORL4], IQL [ORL5].

2.2 Running Baselines

For this project, you are required to use the offline RL algorithms from the d3rlpy library and implement another offline RL algorithm. You should first set up the environment appropriately and reproduce the results in the **medium-replay-v0** dataset of the three environments: **halfcheetah**, **hopper**, and **walker2d**. Then you can choose an algorithm from Extreme Q-Learning [ORL6], SAC-N [ORL7], and Q-learning Decision Transformer [ORL8] to implement and merge it into the original d3rlpy codebase. Note that you may first clone the d3rlpy codebase and add new scripts to do your implementation.

2.3 Optimizing the Dataset

In this part, you are going to optimize the medium-replay dataset for offline RL. The medium-replay dataset is a dataset collected by the D4RL benchmark [ORL9], which contains samples from an RL agent's replay buffer. It's important to note that you have **no** access to the underlying environment in this task, and you can only manipulate the dataset itself.

One primary approach is utilizing the trajectory return information and trimming the dataset to remove the low-return trajectories. However, you need to balance the trade-off between the dataset size and the quality of the dataset, and the trajectory return is not the only factor that affects the quality of the dataset. Other factors, such as the diversity of the dataset, the coverage of the state space, etc, may have a significant impact as well.

An important side note is that you are not restricted to **trim** the dataset. Although you have no access to the environment in this track, it's possible to **augment** the dataset, or **generate** *fake* data as well. For example, you can add noise to the state or action, or use some kind of generative model to generate new samples following the data distribution.

2.4 Potential Directions

If you are interested in this project but have no idea on where to start, might as well have a try at these directions:

- **Trim the dataset with self-supervised criteria**: You can try to remove some out-of-distribution trajectories or samples from the dataset, similar approach can be found in data-cleaning literature like [ORL10].
- **Trim the dataset from the feedback of agents**: Inspired by the effective approach in [ORL2], you can train an offline RL or imitation learning agent and use its likelihood of taking the action given states from the dataset as a criterion, and remove the transitions with low likelihood.
- **Trim the dataset according to curiosity**: You can use a curiosity-driven model to evaluate the novelty of the state-action pairs in the dataset, and remove the low-curiosity samples, as in exploration literature like [ORL11].
- **Augment the dataset with generated data**: You can use a generative model (VAE, diffusion model, etc) to generate new samples following the data distribution, and add them to the dataset to improve the data coverage. One example can be found in [ORL12].
- **Play with the reward signal of the dataset**: Surprising results for [ORL13] show that offline RL algorithms aren't so sensitive to the reward signal, and they can be manipulated to improve the performance. You can try to modify the reward signal of (part of) the dataset as well.

3 About Grading

We provide a reference grading policy for this project as follows:

- [**60 pts**]: Report the baseline results of Section 2.2. Use the three algorithms (BCQ, CQL, and IQL) implemented by d3rlpy to test the dataset optimization results from now on. Try the "primary approach" mentioned above and analyze the results in detail. You should test with at least 10 different trimming percentages. If you can't get enough computational resources, you can just report the results on the **halfcheetah** environment.
- [**80 pts**]: Conduct thorough experiments through either trimming or augmenting the dataset, and analyze the results in detail. You can use the ideas mentioned above, or use your proposed ones. You should compare your algorithm with the best of the "primary results" and show an improvement in the performance of the three algorithms.
- [**100 pts**]: You can combine different kinds of operations on the dataset, and show a greater improvement in the performance of the three algorithms.
- **Try conference submission**: Verify the versatility of your method on more environments, like Atari from d3rlpy.

4 Contact

If you have any questions, please feel free to contact our TA Kaizhe Hu via WeChat or email: hkz22@mails.tsinghua.edu.cn.

Offline RL Reference

[ORL1] Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020.

[ORL2] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, Eyal Orgad, Rahim Entezari, Giannis Daras, Sarah Pratt, Vivek Ramanujan, Yonatan Bitton, Kalyani Marathe, Stephen Mussmann, Richard Vencu, Mehdi Cherti, Ranjay Krishna, Pang Wei Koh, Olga Saukh, Alexander Ratner, Shuran Song, Hannaneh Hajishirzi, Ali Farhadi, Romain Beaumont, Sewoong Oh, Alex Dimakis, Jenia Jitsev, Yair Carmon, Vaishaal Shankar, and Ludwig Schmidt. Datacomp: In search of the next generation of multimodal datasets, 2023.

[ORL3] Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration, 2019.

[ORL4] Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning, 2020.

[ORL5] Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning, 2021.

[ORL6] Divyansh Garg, Joey Hejna, Matthieu Geist, and Stefano Ermon. Extreme q-learning: Maxent reinforcement learning without entropy. 2023.

[ORL7] Gaon An, Seungyong Moon, Jang-Hyun Kim, and Hyun Oh Song. Uncertainty-based offline reinforcement learning with diversified q-ensemble, 2021.

[ORL8] Taku Yamagata, Ahmed Khalil, and Raul Santos-Rodriguez. Q-learning decision transformer: Leveraging dynamic programming for conditional sequence modelling in offline rl, 2023.

[ORL9] Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning, 2020.

[ORL10] Fabian Gröger, Simone Lionetti, Philippe Gottfrois, Alvaro Gonzalez-Jimenez, Ludovic Amruthalingam, Labelling Consortium, Matthew Groh, Alexander A. Navarini, and Marc Pouly. Selfclean: A self-supervised data cleaning strategy, 2023.

[ORL11] Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation, 2018.

[ORL12] Guanghe Li, Yixiang Shan, Zhengbang Zhu, Ting Long, and Weinan Zhang. Diffstitch: Boosting offline reinforcement learning with diffusion-based trajectory stitching, 2024.

[ORL13] Anqi Li, Dipendra Misra, Andrey Kolobov, and Ching-An Cheng. Survival instinct in offline reinforcement learning, 2023.

Visual RL Track

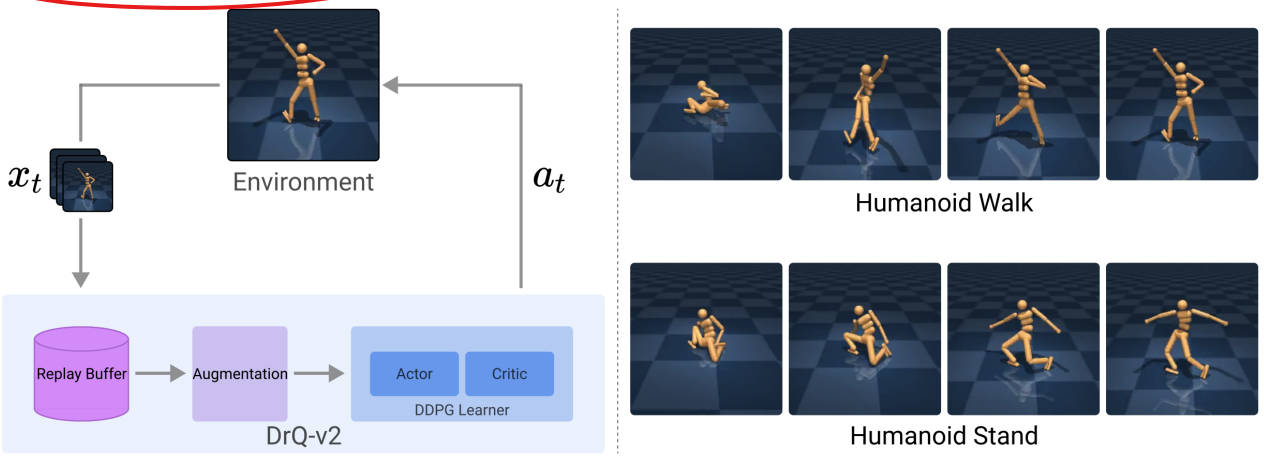

Figure 2: **DrQ-v2** is an actor-critic algorithm for image-based RL. It alleviates encoder overfitting by applying random shift augmentation to pixel observations sampled from the replay buffer.

1 Project Description

Visual Reinforcement Learning (Visual RL), coupled with high-dimensional observations, has consistently confronted the long-standing challenge of high training difficulty and low sample efficiency.

Your team should first conduct a thorough literature review on this field, and understand how the existing approaches tackle this high-dimensional input problem. Additionally, you should read the paper of some state-of-the-art algorithms (DrQ-v2 [VRL1], DrM [VRL2]) (an improved method built on DrQ-v2) and their codebase thoroughly. Then you should reproduce the results from the SOTA methods, and propose your own techniques to improve the performance.

Notice: Visual RL often takes quite long time to train (from several hours to several days) and it requires computation resources. It may be unsuitable for you if you only have a local machine or you plan to rush this project in one week!

2 Requirements

In this section, we will provide a reference technology roadmap for your project.

2.1 Understanding Basic Concepts and Algorithms

Provide a comprehensive literature review on visual RL, identify current challenges, and give a brief introduction to the technical details of DrQ-v2 [VRL1] and DrM [VRL2].

2.2 Running Baselines

For this project, you are required to conduct experiments in the DeepMind Control Suite [VRL3], which is one of the default environments in the original DrQ-v2 and DrM paper.

You should reproduce the results in the paper in the environment **Cartpole Balance** (Easy), **Hopper Hop** (Medium) and **Dog Stand** (Hard).

2.3 Algorithm Design

Enhance the algorithm's performance on the chosen tasks (you can add more medium/hard tasks if you want). You should compare your algorithm with our proposed baselines and an improvement on any prospective (e.g. final performance, sample efficiency) over the DrQ-v2 baseline is expected.

Of course, you are also encouraged to challenge our new SOTA DrM in final performance, sample efficiency, robustness for random seeds, etc. If you successfully beat DrM, congratulations! You can try submit your project to a top conference!

2.4 Potential Directions

If you are interested in this project but have no idea on where to start, might as well have a try on these directions:

- **Apply more useful data augmentations**: Data augmentation is a key trick in the success of DrQ-v2, and many successors have tried other data augmentation methods to improve its performance [VRL4, VRL5]. You can try to combine some novel data augmentation methods or propose a new one.
- **Use novel pre-trained visual encoders**: Pre-trained visual encoders are widely used in computer vision, reinforcement learning or robotics tasks [VRL6, VRL7, VRL8, VRL9]. You can replace the original visual encoder with some pre-trained ones to test their performance.
- **Propose some new representation learning approach for visual encoding**: Many previous works focus on how to design a proper visual encoder (pre-)training objective to better arouse visual encoder for robotics tasks [VRL9, VRL8]. Please refer to relevant projects and propose your own idea.
- **Leverage whatever model-free RL tricks**: We have learned many tricks in class, such as double Q network [VRL10], TD(λ) [VRL11], prioritized experience replay [VRL12]. Some of these tricks have been applied to DrQ-v2, since it is an algorithm built on DDPG [VRL13], but there remain some other tricks, and also there are some novel methods proposed for RL recently [VRL14]. You can try some tricks after performing an additional literature survey or put forward your own design.

Note: After implementing your own ideas/tricks, you can always first try the Cartpole environment to quickly verify whether your implementation is correct!

3 About Grading

We provide a reference grading policy for this project as follows:

- **[60 pts]**: Report the baseline results, and try a trick that others proposed but haven't been implemented in visual RL (e.g. prioritized experience replay).
- **[80 pts]**:
  - Report the baseline results, conduct thorough experiments on at least 8 tricks (2 per direction), and analyze the results in detail (it can be okay if all of them fail).

  - Report the baseline results, propose and implement your own idea, and analyze the results in detail (it can be okay if it fails).
- **[100 pts]**: Make a great improvement on DrQ-v2.
- **Try conference submission**: Beat DrM in performance.

4 Contact

If you have any questions, please feel free to contact our TA Pu Hua via WeChat or email: huap20@mails.tsinghua.edu.cn.

Visual RL Reference

[VRL1] Denis Yarats, Rob Fergus, Alessandro Lazaric, and Lerrel Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. In *International Conference on Learning Representations*, 2021.

[VRL2] Guowei Xu, Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Zhecheng Yuan, Tianying Ji, Yu Luo, Xiaoyu Liu, Jiaxin Yuan, Pu Hua, et al. Drm: Mastering visual reinforcement learning through dormant ratio minimization. In *The Twelfth International Conference on Learning Representations*, 2023.

[VRL3] Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, et al. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

[VRL4] Guozheng Ma, Zhen Wang, Zhecheng Yuan, Xueqian Wang, Bo Yuan, and Dacheng Tao. A comprehensive survey of data augmentation in visual reinforcement learning. *arXiv preprint arXiv:2210.04561*, 2022.

[VRL5] Guozheng Ma, Linrui Zhang, Haoyu Wang, Lu Li, Zilin Wang, Zhen Wang, Li Shen, Xueqian Wang, and Dacheng Tao. Learning better with less: Effective augmentation for sample-efficient visual reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[VRL6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[VRL7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[VRL8] Suraj Nair, Aravind Rajeswaran, Vikash Kumar, Chelsea Finn, and Abhinav Gupta. R3m: A universal visual representation for robot manipulation. In *Conference on Robot Learning*, pages 892–909. PMLR, 2023.

[VRL9] Yecheng Jason Ma, Shagun Sodhani, Dinesh Jayaraman, Osbert Bastani, Vikash Kumar, and Amy Zhang. Vip: Towards universal visual reward and representation

via value-implicit pre-training. In *The Eleventh International Conference on Learning Representations*, 2022.

[VRL10] Hado Van Hasselt, Arthur Guez, and David Silver. Deep reinforcement learning with double q-learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 30, 2016.

[VRL11] Harm Seijen and Rich Sutton. True online td(lambda). In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 692–700, Bejing, China, 22–24 Jun 2014. PMLR.

[VRL12] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*, 2015.

[VRL13] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[VRL14] Tianying Ji, Yu Luo, Fuchun Sun, Xianyuan Zhan, Jianwei Zhang, and Huazhe Xu. Seizing serendipity: Exploiting the value of past success in off-policy actor-critic. *arXiv preprint arXiv:2306.02865*, 2023.

Imitation Learning Track

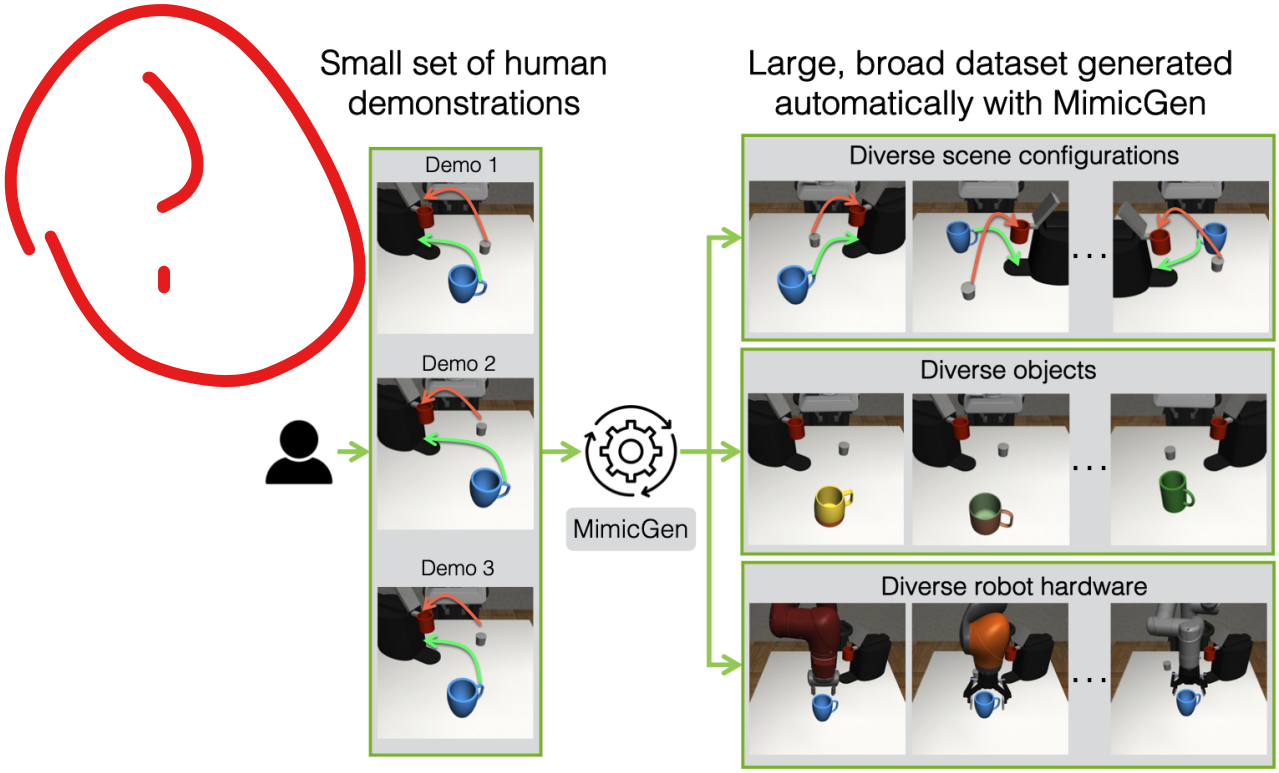

Figure 3: **MimicGen** is a data generation system that can produce large diverse datasets from a small number of human demonstrations by re-purposing the demonstrations to make them applicable in new settings.

1 Project Description

Imitation learning from human demonstrations has become an effective paradigm for robot learning. One popular approach to acquiring those demonstrations is to have human operators teleoperate robot arms to complete tasks [IL1, IL2], but this method suffers costly and time-consuming human labor, especially when the number of required data scales up. To tackle this problem, MimicGen [IL3] provides another direction to get demonstrations, which is to generate a large number of data from a small number of human-collected data.

Your team should first conduct a thorough literature review on this field, and discover how the number of demonstrations can affect the preformance of imitation learning algorithms. Additionally, you should read the paper of MimicGen [IL3] and its relevant codebase (mimicgen, robomimic, robosuite) thoroughly. Then you should reproduce the results from MimicGen in some environments, since the demo editing method is not open-sourced, and we are also open to your own ideas.

Notice: This project could be quite clear in its technical roadmap but hard in coding because you will not have the open-sourced code for the core method. It will be better if you have some knowledge or experience in robotics, such as homogeneous transformation and inverse kinematics.

2 Requirements

In this section, we will provide a reference technology roadmap for your project. Note that you are not required to finish all the steps in this roadmap.

2.1 Understanding Basic Concepts and Algorithms

Provide a comprehensive literature review on imitation learning and data collection methods, give a brief introduction to the technical details of BC (Behaviour Cloning), BC-RNN and Diffusion Policy (SOTA) [IL4], and compare the number of demonstrations required in these algorithms (state-based and visual-based) for some task in robosuite.

2.2 Running Baselines

For this project, you are first required to download the source and core data, and set up the environment following the instructions in the repo of MimicGen. Then run the BC-RNN algorithm (state-based/visual-based) on the tasks **Stack** (Easy) and **Square** (Medium) with original data and generated data respectively. Demonstrate your comparison between the performance from less human-collected data and from more generated data on both tasks.

2.3 Generating New Demonstrations w/o Sub-task Parsing

You can first try to generate some new demonstrations without the sub-task parsing step, which means you can consider each task as the smallest unit that cannot be decomposed. A problem you may meet with is that in such the smallest task, only one object can be randomized, because you should edit the whole trajectory based on such random transformation. But in fact there are two different objects in both required environments. A possible way to solve this problem can be to **randomize these two objects with the same transformation**, which means to make the relative pose between the two objects unchanged.

You can **start from only applying translation (without rotation)** to each object to simplify the problem, and then further apply rotation to the transformation.

Then you can **train BC-RNN with your generated data**, and compare the results with those trained with the original human-collected data and data generated by data released by MimicGen.

2.4 Understanding the Data Generation Rate (Bonus 10 pts)

Theoretically, if the demo generation pipeline is implemented correctly, you should always get successful generated trajectories. But in fact the data generation rate will not always be 1, because there will be problems with the precision of inverse kinematics computation and collision. However, you will find the data generation rate suffer from a large drop from 1, indicating that there must be some other thing "wrong" besides the two problems aforementioned (so it is okay if your implementation does not always output successful demos). Please give your own reason and analysis.

Hint: You should look into the code of the OSC controller in robosuite.

2.5 Generating New Demonstrations w/ Sub-task Parsing

After you finish your generation pipeline without sub-task parsing, you can go on to add it back. With sub-task parsing, you can apply different randomization to each object in the

scene, and then increase the diversity of your generated data. Then you can perform another training on this new generated data and conduct comparison with the previous results.

2.6 Exploration

You can refer to the Limitations section in the original paper of MimicGen and try any trick you can come up with to improve it. If you manage to extend MimicGen to another level, it is very likely to be a submission to a top conference in robot learning!

3 About Grading

We provide a reference grading policy for this project as following:

- **[60 pts]**: Report the baseline results (BC-RNN on two tasks with two kinds of data).
- **[80 pts]**: Report the baseline results, and successfully implement generation pipeline without sub-task parsing. Additionally deliver the video of a generated demonstration besides curves.
- **[100 pts]**: Report the baseline results, and successfully implement generation pipeline with sub-task parsing. Additionally deliver the video of a generated demonstration besides curves.
- **Try conference submission**: Improve MimicGen.

4 Contact

If you have any questions, please feel free to contact our TA: Pu Hua via WeChat or his email address huap20@mails.tsinghua.edu.cn.

Imitation Learning Reference

[IL1] Cheng Chi, Zhenjia Xu, Chuer Pan, Eric Cousineau, Benjamin Burchfiel, Siyuan Feng, Russ Tedrake, and Shuran Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[IL2] Ajay Mandlekar, Yuke Zhu, Animesh Garg, Jonathan Booher, Max Spero, Albert Tung, Julian Gao, John Emmons, Anchit Gupta, Emre Orbay, et al. Roboturk: A crowdsourcing platform for robotic skill learning through imitation. In *Conference on Robot Learning*, pages 879–893. PMLR, 2018.

[IL3] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Conference on Robot Learning*, pages 1820–1864. PMLR, 2023.

[IL4] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023.

Your Own Idea

1 Project Description

You have the option to choose your own topic in the course. For this project, You must follow these tips:

- Your proposed project should be related to our course.
- **You must discuss your proposal with TAs and get their approval, no matter whether you are proposing your brand new idea or just working on the project in your own lab.**
- **You cannot use the papers that you have already posted on arXiv or have published.**

2 Requirements

In this section, we will provide a reference technology roadmap for your project. Note that it is just a reference and we can help you to build and adjust your roadmap based on your specific idea.

2.1 Basic concepts and algorithms

Provide a comprehensive review of the literature on this field or the related work of your own solution.

2.2 Run Baselines

Run the baselines in your setting.

2.3 Algorithm Design

Design your own algorthm.

3 About Grading

We will discuss with you on a reference grading policy after you decide to work on your own project and get our approval.

4 Contact

If you choose this track, please first contact anyone of our TAs via Wechat or email, and we will schedule a discussion with you on your idea.