cmput 607: Empirical RL

Lecture 1

- Now recording

What is this class about?

Reinforcement learning and experments!

- RL is largely an empirical science
- It’s like regular science, except we design computational worlds to then deploy scientific analysis on
- The agent (algorithm), environment (problem), and experiment protocol (e.g., episodic vs continuing) produce a dynamical system that we ask questions about—that we seek to understand
- It is **easy** because we do it on computers; we control everything (unlike rabbits)
- It is **hard** because we typically compare multiple agents, that each generate their own data streams
- There are many ways to get this wrong and unfortunately bad experiments are common

Science is not about ranking numbers

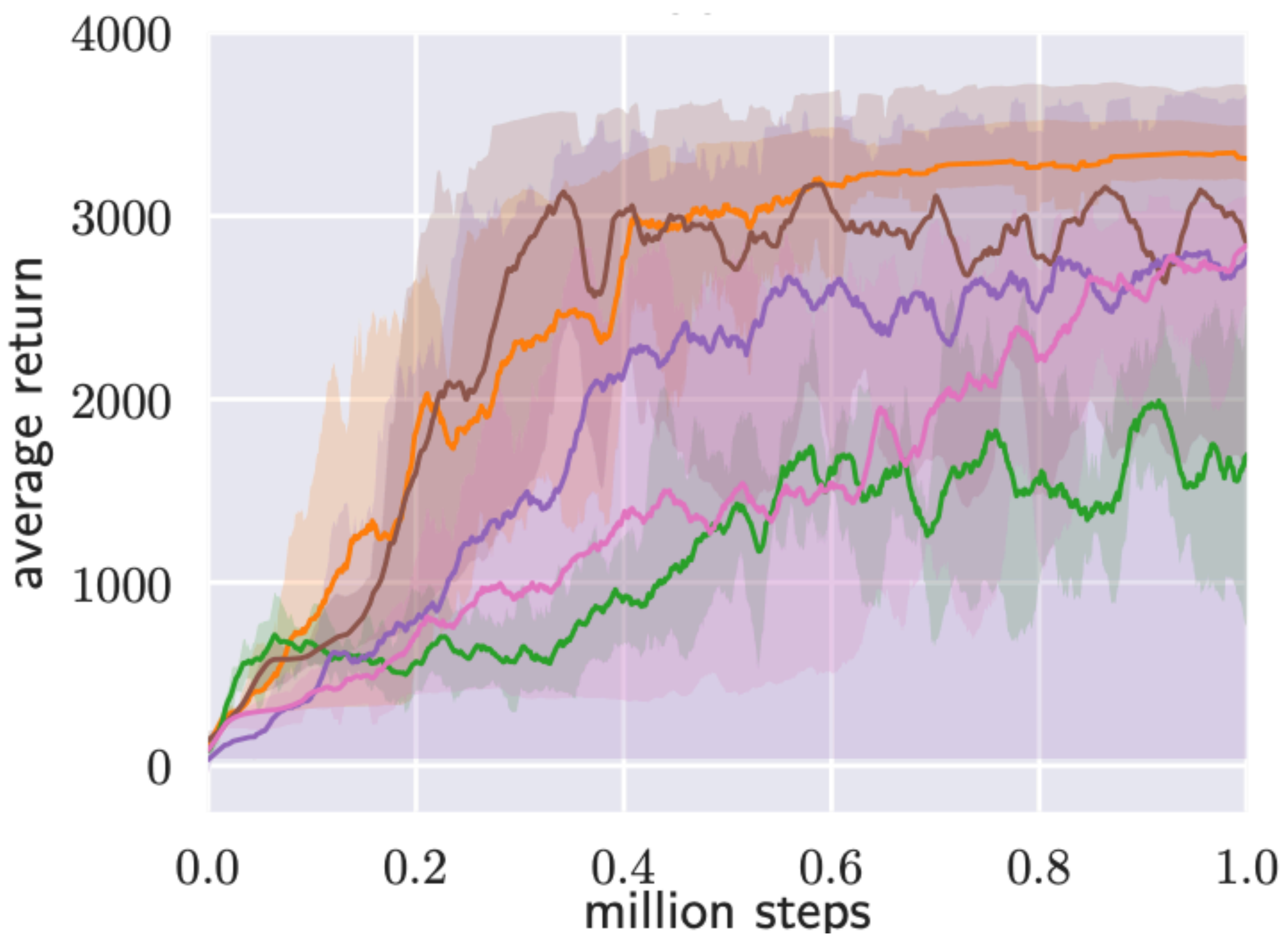
It is about insight and understanding

- You might want to better understand the strengths and weaknesses of your own algorithm
- Or you might want to understand the fundamental principles of learning and minds
- Leaderboard driven optimization of scores in but one type of question
  - “My number is bigger than your number is the lowest form of science” -Sutton
- Even ignoring the limitations of leader-boarding as a form of science, we do RL experiments really poorly!
  - And, the standard for claiming algorithm X > algorithm Y is really high!

Common mistakes in empirical RL

You will see every one of these mistakes in published papers. I have!

That does not mean its ok to do!!! These things are NOT OK!

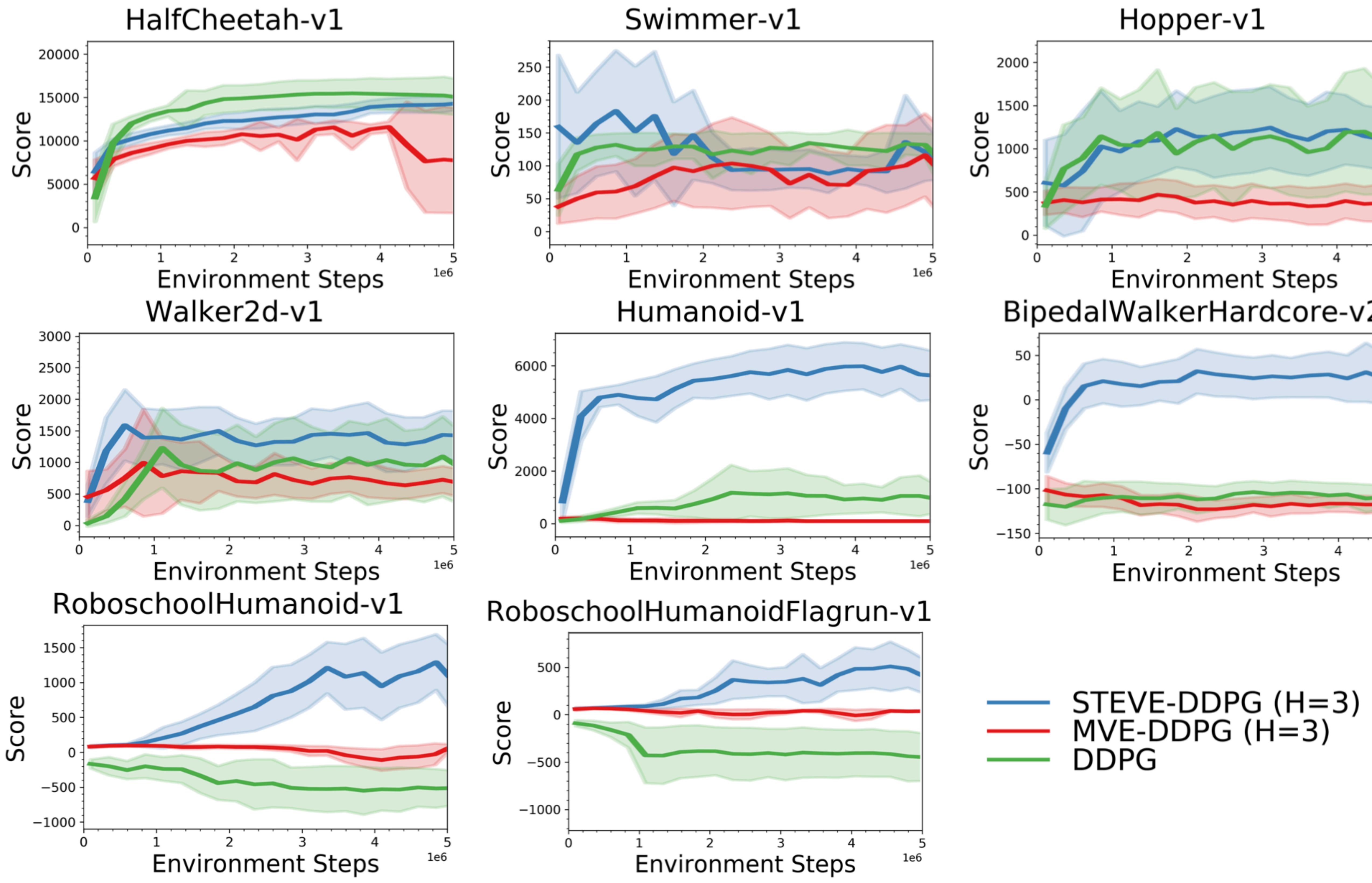


4000
3000
2000
1000
0
average return
0.0
0.2
0.4
0.6
0.8
1.0
million steps

Not enough runs

Insignificant results

- **Common practice**: run your new agent against several baselines you found on GitHub, average over 3 runs, plot learning curves
- **Problem**: 3 runs is almost never enough to support a valid statistically significant claim
  - It is not even enough to estimate the standard deviation of the data, and thus the “error bars” plotted are invalid…more on this in later lectures
- **Common reaction**: “but my agent/environment is huge and doing more runs requires too much compute”
- **Translation**: I want to run an experiment that I don’t have enough resources for, so can you just pretend with me that this result means something?
- **Solution**: only ask empirical questions for which you have the data, time (deadlines), and compute to answer


HalfCheetah-v1
Swimmer-v1
Hopper-v1
Walker2d-v1
Humanoid-v1
BipedalWalkerHardcore-v2
RoboschoolHumanoid-v1
RoboschoolHumanoidFlagrun-v1
Score
Environment Steps
1e6
STEVE-DDPG (H=3)
MVE-DDPG (H=3)
DDPG

Incorrect baselines

Picking the wrong competitors

- **Common practice**: compare your agent against a previous version of your algorithm, or some arbitrary agents you already have code for
- **Problem**: the choice of baseline method depends on the research question
  - Going after SOTA? better find the best method
  - In other cases the baseline should directly attempt to test the main idea underlying your method
- **Common reaction**: “it’s hard to implement that method”, “alg X is pretty close to SOTA in this environment”, “it’s not clear what the baseline should be?”—bad sign
- **Solution**: if it is not clear to you the experimenter what the correct baselines are, then you are in trouble. This should not be decided after the fact

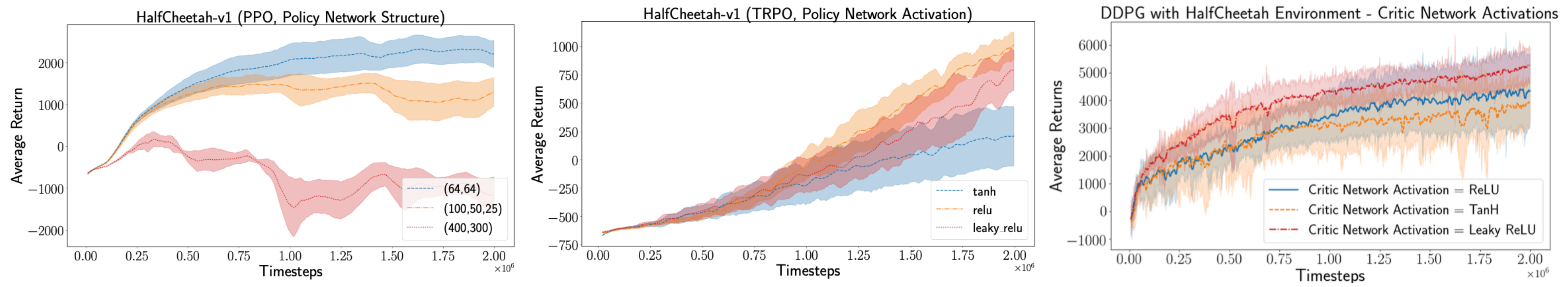


Figure 2: Significance of Policy Network Structure and Activation Functions PPO (left), TRPO (middle) and DDPG (right).

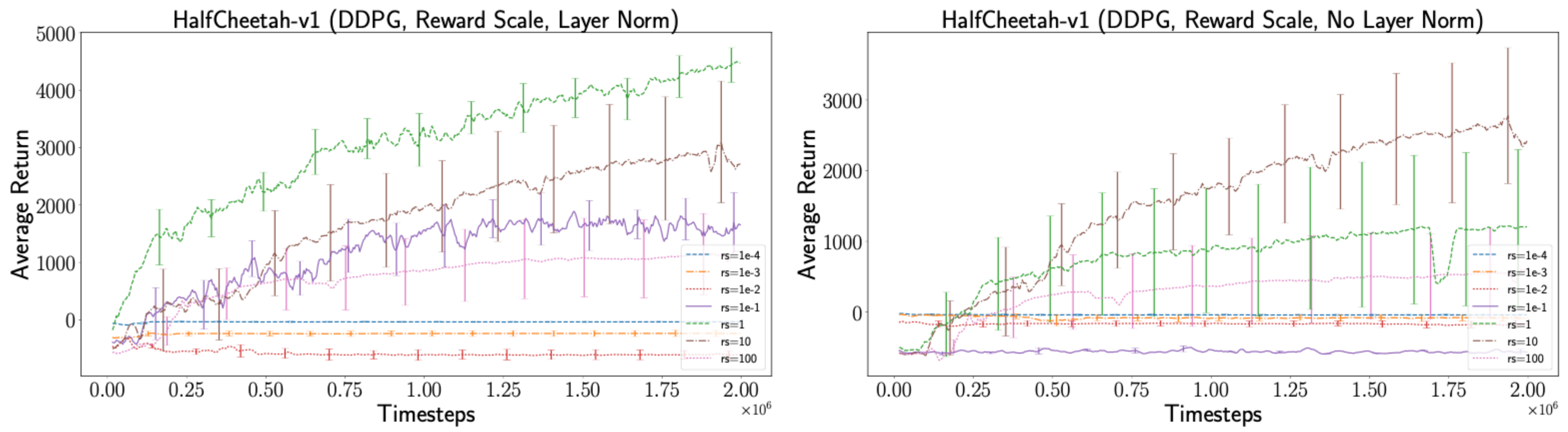
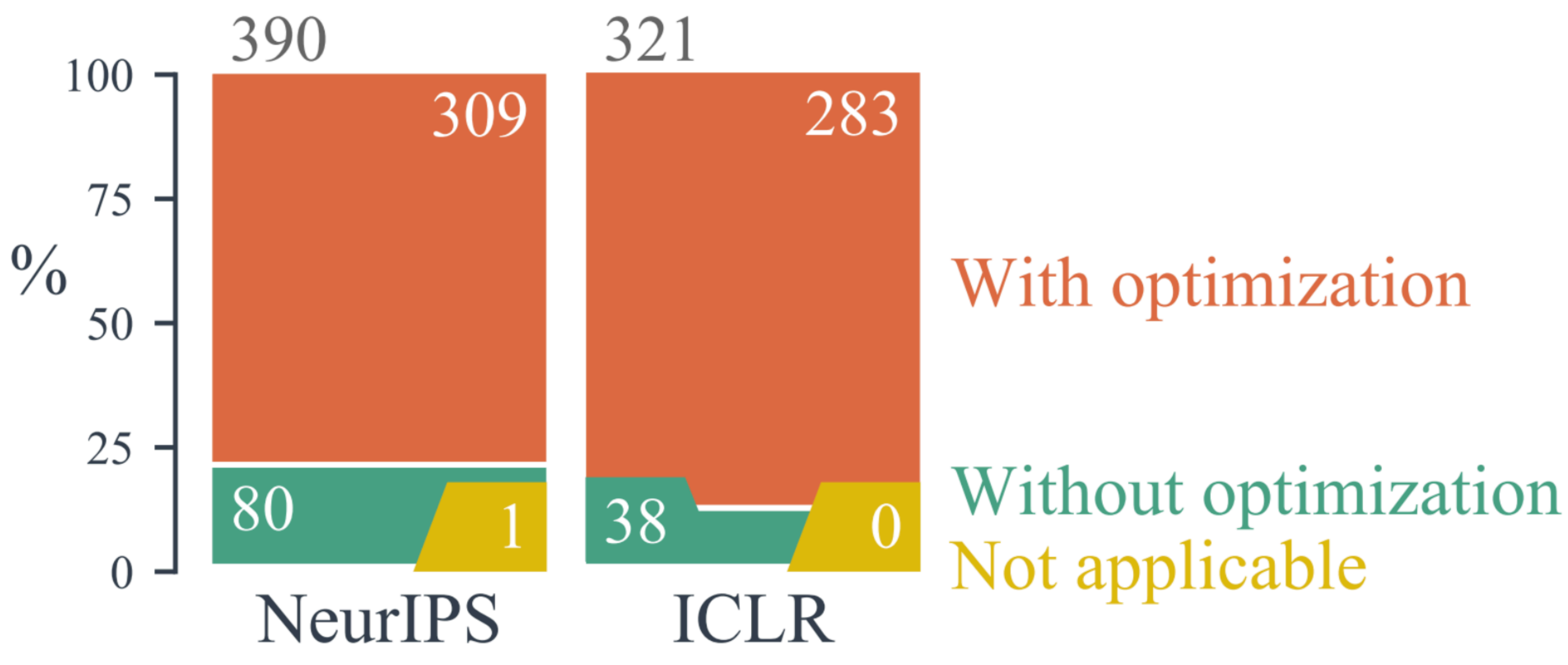
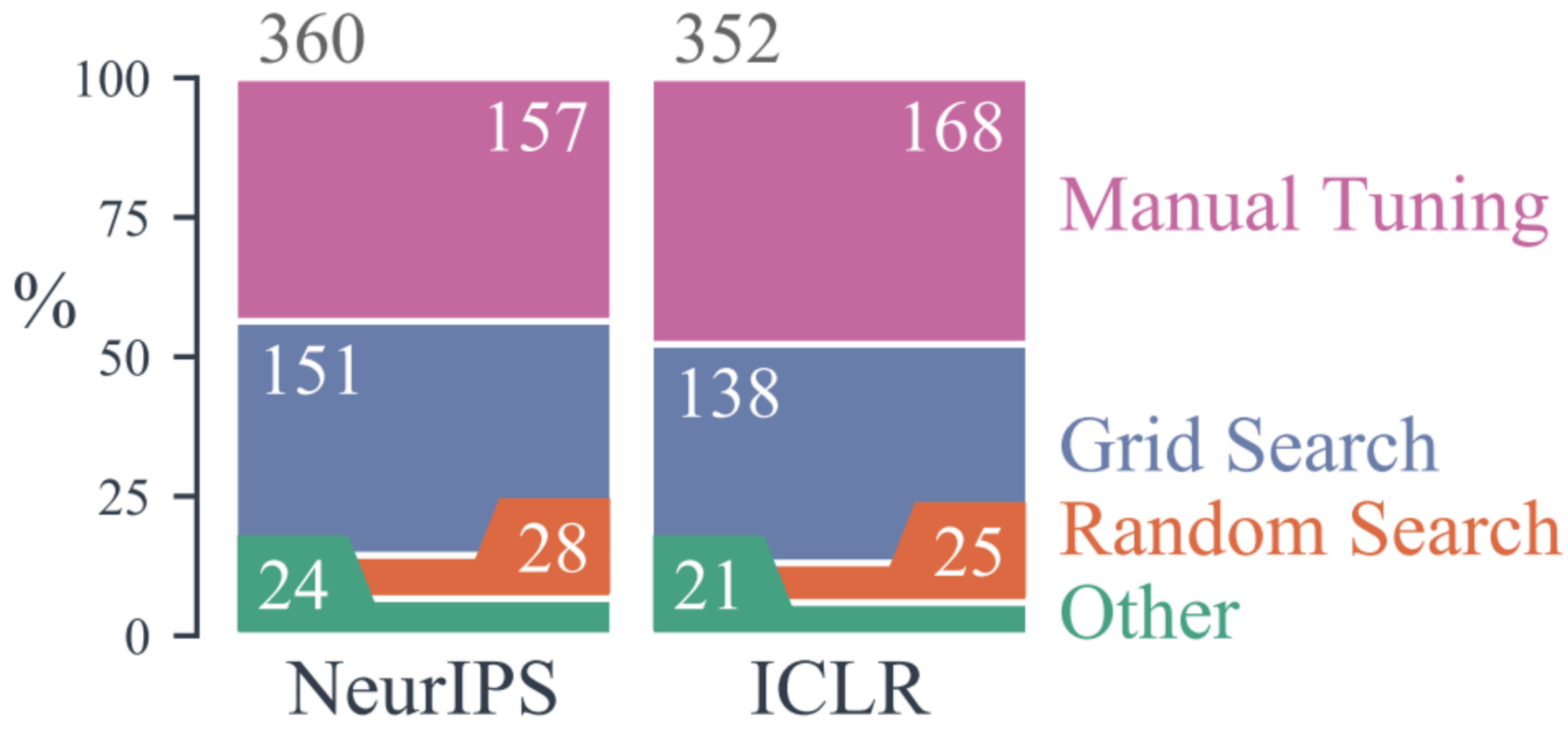


Figure 3: DDPG reward rescaling on HalfCheetah-v1, with and without layer norm.

Question: *Did you optimize your hyperparameters?*



Question: *If yes, you did optimize, how did you tune them?*



Untuned baselines

Misrepresenting the performance of other methods

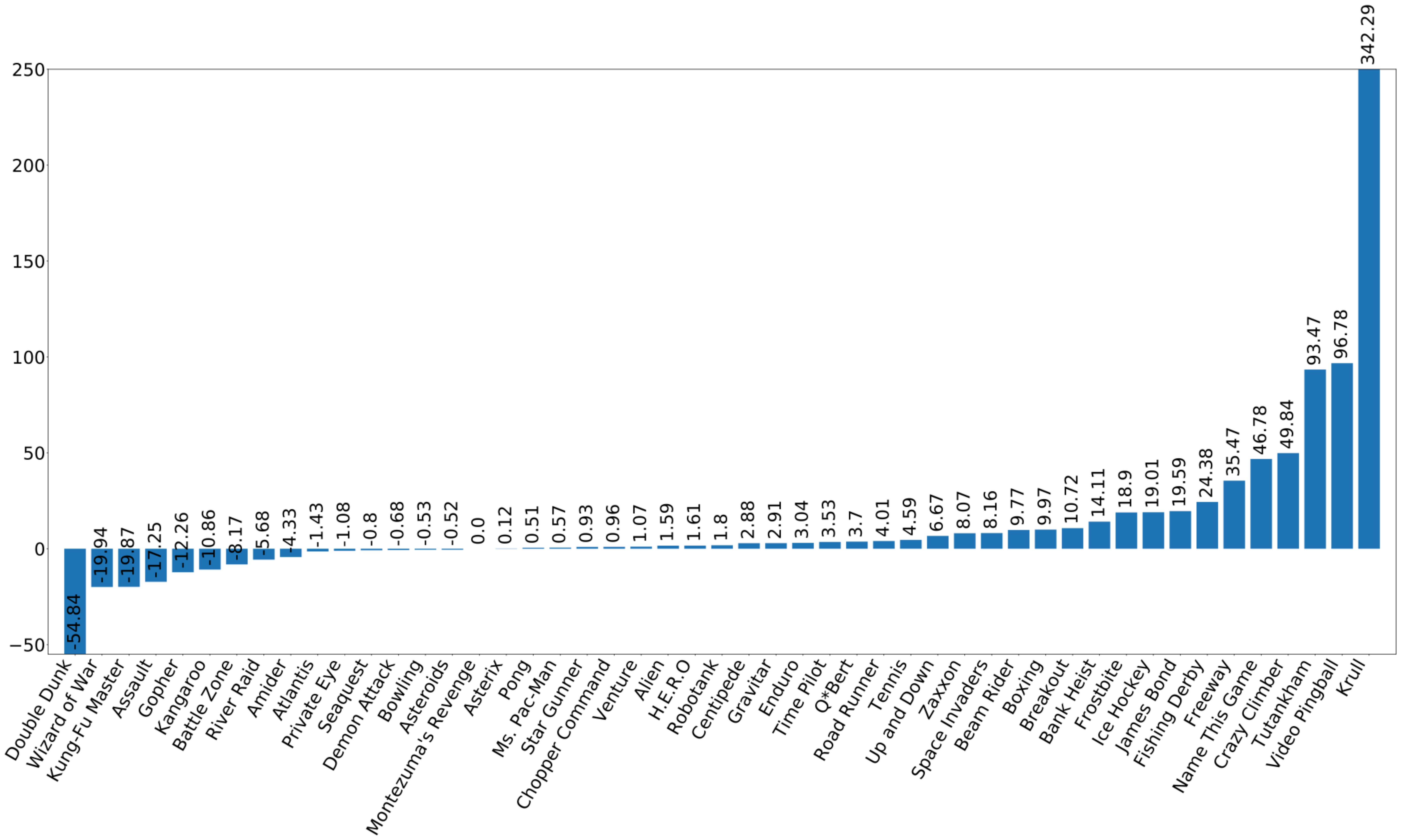
- **Common practice**: you test your agent on a new environment (gameX) you invented. You compare your agent to DQN on gameX. You simply use Nature DQN settings for the hyper-parameters and network architecture.
- **Problem**: DQN is tuned for Atari!! It will likely do much better on gameX if you retune the hyper-parameters
    - Do you think **your new algorithm** is tuned for gameX?
- **Common reaction**: “DNQ works on everything”, “nobody does that”, “it’s too much compute to do that”
- **Solution**: either tune DQN for gameX or use an environment for which DQN’s hyper-parameters have been tuned

Misleading ablations

Not tuning the ablations

- **Common practice**: imagine you invented DNQ+X+Y. Where X and Y are novel algorithmic components. You want to prove that both X and Y matter. You compare DQN, DQN+X, and DQN+Y
- **Problem**: Hopefully DQN was tuned for the environment. We know DQN+X+Y is tuned. What about DQN+X, and DQN+Y. They might perform even better if you tuned their hyper-parameters also
- **Common reaction**: “what?”
- **Solution**: tune each variant in the ablation study

From a hard exploration paper



Cherry picking

There are many ways this can happen:

- Reporting results on environments where your method wins, but not in environments where your method is bad
- Reporting performance measures where your method looks good (e.g., initial learning speed), but not measures that highlight weaknesses (e.g., stability)
- Reporting learning curves for the best hyper-parameter settings found from a sweep
  - Why might this be misleading?
- Not challenging your own idea—an important step is to try and break your method
- *What else? (Ie this would be a good time to unmute and talk!!)*
- **More generally**: trying different combinations of things until you find something that makes your approach look better. *In some sense everything we discussed is cherry picking!*
- **Solution**: you should have a clear hypothesis to test (including baselines and performance measures) before you run your experiment

Missing important details

What did you actually do

- Unspecified number of runs, number of steps, how the hyper-parameters were set, were episodes cutoff, …
- Why this environment? Why these baselines? Why this performance metric
- Missing labels in plots, undefined errorbars
- Which variant of an alg was used? What code base? What other implementation details were key for performance?
- Do we have all the detail relevant for demonstrating the scientific claim of the paper?
- **Solution**: be a sceptic of your own work, get others to read and comment

Conclusions NOT supported by the data

Dreaming and writing poorly…

- Examples:
  - All the error bars overlap; paper claims new method is better
  - Conducts significance test; fails to reject null hypothesis; claims improvement
  - Uses untuned baselines (or is missing baselines); claims improvement
  - Winning in the aggregate but performing poorly on important metrics/tasks
  - Other … ?

- **Solution**: follow the three step process to documenting your experiment

Three steps to writing up your experiment

Clarity and structure of writing matters!

- Describe the (1) problem and (in separate paragraphs) the (2) solution methods, and the (3) experiment setup & (4) performance metrics
- Plot the data and describe how it looks (without making conclusions about it). E.g., “As we can see in Figure 1, agent x accumulates no reward on average for the first 20 episodes and then increases to …”
  - At this stage we don’t make any subjective claims
- Finally, describe the high-level conclusions of the data. E.g., “Our experiment shows that LSTD can learn faster than TD”
  - Be careful to respect that these conclusions are **limited** to your experiment setup: function approx used, hyper-parameters swept, etc

These problems are becoming common

Why?

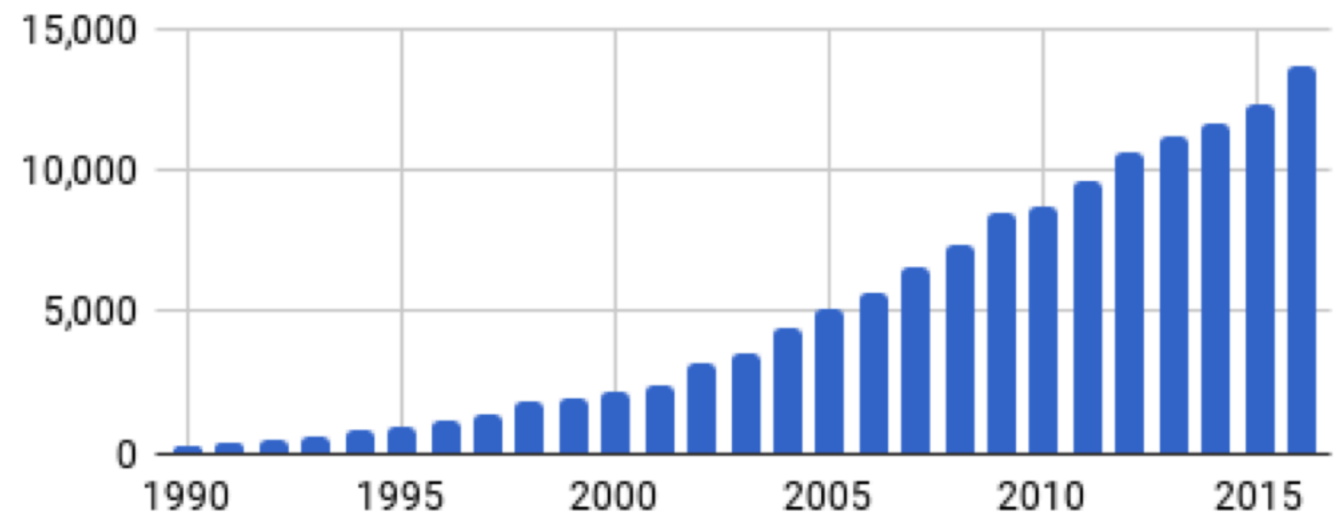


Figure 1: Growth of published reinforcement learning papers. Shown are the number of RL-related publications (y-axis) per year (x-axis) scraped from Google Scholar searches.

- Part of the problem is bad reviewers:
  - unrealistic expectations are common:
    - ``must have SOTA on Atari” for every paper etc
    - ``Where is DQN or SAC” i.e., some alg the reviewer knows well
  - They don’t know that non-system building, non-SOTA chasing papers are possible
    - One can propose new algorithmic ideas without proving it helps SOTA agents
  - In the end many reviews have difficulty understanding and seeing value in papers that are different than what their papers do and what their research looks like

These problems are becoming common

Low diversity of the current literature

- If you look at seminal older papers in RL, they contain a new idea and a small but well done experiment to show that idea matters
- Today we have papers with unclear math, incomplete explanations of the main idea and huge messy, potentially meaningless experiments
    - Most highly cited papers are about system building and leaderboards
- The reviewing system disproportionally accepts such papers
- Young researchers emulate what they see in the literature
- Thousands of researchers have joined the field recently, and they are not well trained on how to DO & EVALUATE good science
- My job in this course is to teach you to be better. To think scientifically. To help you run better experiments in RL!

This course is an apprenticeship in empirical RL research

Practice being a good scientist

- **This course is all about your project and the stages of research**
- At the end of the term, you (with 2 to 4 other group members) will submit your final project
- In **March** you will submit a draft of your project that includes preliminary experimental data
  - Each project will get three peer reviews from your classmates
  - You will peer review other's projects as part of your grade in this course
- Once during the semester you will make an in class presentation about either:
  - Your project
  - A paper from the literature (highlighting either good empirical practice or bad)

Practice being a good scientist
In other words

- At the end of the term, you will design, and conduct a good experiment
- You will write about it carefully and clearly
- You will practice how to review papers—learning to be critical and constructive
- You will practice integrating advice from reviewers—making your work better
- You will practice presenting

- How do we get there? What will the class look like?

Knowledge of RL is assumed

I won’t really teach RL

- The first few lectures I will give a very brief refresher on RL
- Knowing RL is required for this course:
    - You must have taken a university level course in RL before, or done the RL MOOC on your own
- **Next Monday we will have an in class quiz**
- It won’t count toward your grade
- It’s to help you understand if you have the background to take this class

This class will be project, student, and discussion driven

In other words

- The majority of lectures will be about organizing research, the scientific method, methodology, statistics of RL experiments, how to review, scientific writing and presentation
- After that, the rest of the course will be presentations from you!
- It is very import to gain practice practicing doing good research—that is what graduate school is all about
- Think of this class like CMPUT 603 but with a deep focus on the issues related to empirical RL
    - With perhaps more focus on a high-quality project

How will class-time work?

- Some classes I will lecture
- Some classes we will have 2 or more presentations (depending on # students)
- We want lots of discussion and questions regardless
- Each lecture **two students** will be assigned to be **discussion moderators**
- The discussion moderations will watch the chat and help bring questions to my attention and raise questions as well
    - Part of your participation mark will be based on this
- This class is designed to motivate interaction even though we are remote

In class presentations

- Could be about your project: the research question, related work and motivation if you go earlier in the term
  - If your presenting later in the term you might include some results
- Otherwise you present about a paper from the literature:
  - A paper introducing a new approach to empirical RL (e.g., https://arxiv.org/abs/2006.16958)
  - A paper discussing issues with current practice (e.g., https://arxiv.org/abs/1709.06560, https://arxiv.org/abs/1807.03341)
  - A regular paper that has **bad experiments**; you explain why and describe how the issues can be fixed
  - A regular paper that has **good experiments**; you explain why and describe another experiment that could be run to strengthen the paper
  - If you are not sure just ask me
- We will distribute sign up link soon. It will be first come first serve, but first presentations won't start until mid February

Project draft and peer review

- A draft of your project will be due March 24th
- Two weeks later your review of K other student drafts will be due
- Each project will receive feedback from the instruction team (me and the TAs) and K peer reviews
- We will discuss later how to write a good review
- Your peer-reviewing will count towards your final grade
- We will also flag any serious issues in your draft that must be fixed by the final submission
- **Your draft must include initial results**

Final project will be subject to desk rejection

- If you make a serious error in a conference paper it can be rejected without review
- For your final project if you make such an error you will loose 50% automatically
- The list of desk reject criteria will be discussed in lecture and highlighted in the draft review
    - Example: presenting results with only 3 runs
- This course is about the project. Take it seriously. You can start today!

How you can get help

No new algorithms. Empirical Projects only!

- Ideas from lecture and classmate presentations
- Ask questions in class time
- Discussion in class
- Course slack channel:
  - Discuss with other students and the instruction team
- Draft feedback

Admin summary

- Course webpage: https://amw8.github.io/EmpiricalRL/
  - Contains syllabus, resources, and schedule
- Classes will be on zoom
- We will use **chat** and **slido** for questions and interaction during class
- Class slack channel for discussion outside class time: empirical-rl.slack.com
- Sign-up sheet for presentations will be sent to everyone
- Main instructor: Adam White (amw8@)
- TAs: Andrew (ajjacobs@), Derek (xzli@), Archit (sakhadeo@)

Mark breakdown

- Project 50%
  - 10% for draft
  - 40% for final project
- In class presentation 20%
- Participation mark 30%
  - In class participation & session moderation 15%
  - Peer-review 15%

Academic Integrity

Know the rules! Don’t break them!

- The university has clear policies. It is your responsibility to know them:
  - https://www.ualberta.ca/current-students/academic-resources/academic-integrity/index.html
- Even in graduate courses with projects you can violate these rules: E.g.,
  - Fake your results
  - Plagiarism of others work
  - Generally misrepresenting other’s work as your own
- I must report all suspected cases to facility of Science
- If you are falling behind or suffering, then reach out to me—don’t cheat

Code of student behaviour

Know the rules! Don’t break them!

- The university has clear policies. It is your responsibility to know them:
  - https://www.ualberta.ca/governance/resources/policies-standards-and-codes-of-conduct/code-of-student-behaviour.html
- You will interact with your classmates in Zoom, Slido, and Slack
- You must adhere to the code in all these mediums
- Be respectful and Kind
- Understand that text is a bad medium for understanding intent:
  - Don’t assume bad intent in other’s messages
  - Think about how others might misunderstand the intent behind your messages
- **Bullying & Harassment will not be tolerated.** Please contact me, the TAs, or report here:
  - https://www.ualberta.ca/vice-president-finance/office-of-safe-disclosure-human-rights/about-online-reporting.html

Incomplete list of project ideas

No new algorithms. Empirical Projects only!

- Compare different forms of experience replay in a simple domain
- Compare experience replay with eligibility traces
- Investigate why DQN works poorly on cost-to-goal problems like Mountain Car
- Investigate different confidence intervals and hypothesis tests for comparing RL agents
- Compare Actor-critic with V-critic vs Q-critic
- Investigate the maximization bias of Sarsa variants
- Compare FF NNs and RNN on a fully observable problem
- Compare logistic Q-learning and Q-learning
- Compare PPO and TRPO on classic control domains
- Find a time-series dataset to further investigate Nexting and GVFs

Incomplete list of project ideas

No new algorithms. Empirical Projects only!

- Compare average reward methods to discounted variants
- Investigate the impact of delta update, unbiased average update and vanilla average reward Sarsa
- Random Representation vs NNs+backprop on a classic control domain
- Kernel methods vs NNs+backprop on a classic control domain
- Compare SAC and simple linear-gaussian AC in a continuous action task
- Compare SAC and action-discretization in a domain where discretization may hurt (pit world)
- Compare a couple exploration methods in a non-stationary grid world
- Investigate sparse activation function (LTA)
- *Take experiment from Sutton&Barto and take the next step …*
- *Take paper from the literature and improve one of its experiments …*

Why are you taking this class?

Expectation management is the key to life :)

- What is your background?
- What is your research area?
- What do you want to learn about in this class?
- What is your view of the current state of RL research?