Reproducibility

Ron Parr

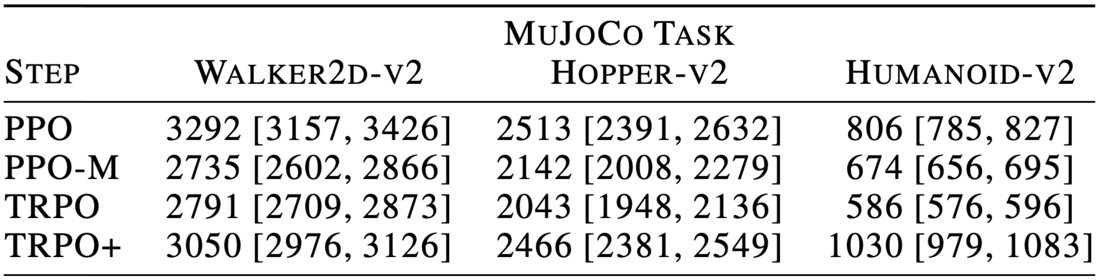
CSCI 2951-F

Brown University

Recall TRP vs PPO

- PPO originally introduced as a simpler alternative to TRPO
- Was also shown to perform better in many cases
- Engstrom et al. (IMPLEMENTATION MATTERS IN DEEP POLICY GRADIENTS: A CASE STUDY ON PPO AND TRPO) investigate this:
    - Find 9 optimizations in PPO not (clearly) documented as main improvements
    - "We find that much of the PPO's observed improvement in performance comes from seemingly small modifications to the core algorithm that either can be found only in a paper's original implementa- tion, or are described as auxiliary details and are *not* present in the corresponding TRPO baselines."
    - "Ultimately, we discover that the *PPO code-optimizations are more important in terms of final reward achieved* than the choice of general training algorithm (TRPO vs. PPO). "

Performance comparison



		MuJoCo Task	
Step	Walker2d-v2	Hopper-v2	Humanoid-v2
PPO	3292 [3157, 3426]	2513 [2391, 2632]	806 [785, 827]
PPO-M	2735 [2602, 2866]	2142 [2008, 2279]	674 [656, 695]
TRPO	2791 [2709, 2873]	2043 [1948, 2136]	586 [576, 596]
TRPO+	3050 [2976, 3126]	2466 [2381, 2549]	1030 [979, 1083]

[Engstrom et al., ICLR 19]

- PPO = full PPO algorithm
- PPO-M = PPO w/o 9 (seemingly secondary) optimizations
- TRPO = original TRPO algorithm
- TRPO+ = TRPO with PPO optimizations
- [,] = 95% confidence interval

Why reproducibility matters

- Scientific method helps us distinguish facts vs. theory/superstition/intuition etc.
- Scientific method is a process
- Failures:
  - Sow confusion
  - Waste time
  - Undermine public confidence in science
- But keep in mind:
  - We’re still human
  - We will make mistakes
  - That’s actually part of the process

How mistakes happen

- Honest mistakes
  - Clerical errors
  - Asking the wrong question/not checking the right thing
  - Unconscious biases (e.g., confirmation bias)
  - Statistical errors
- Misconduct
  - Falsification of data
  - Cherry picking
  - Reviewer misconduct

Is cherry picking ever OK?

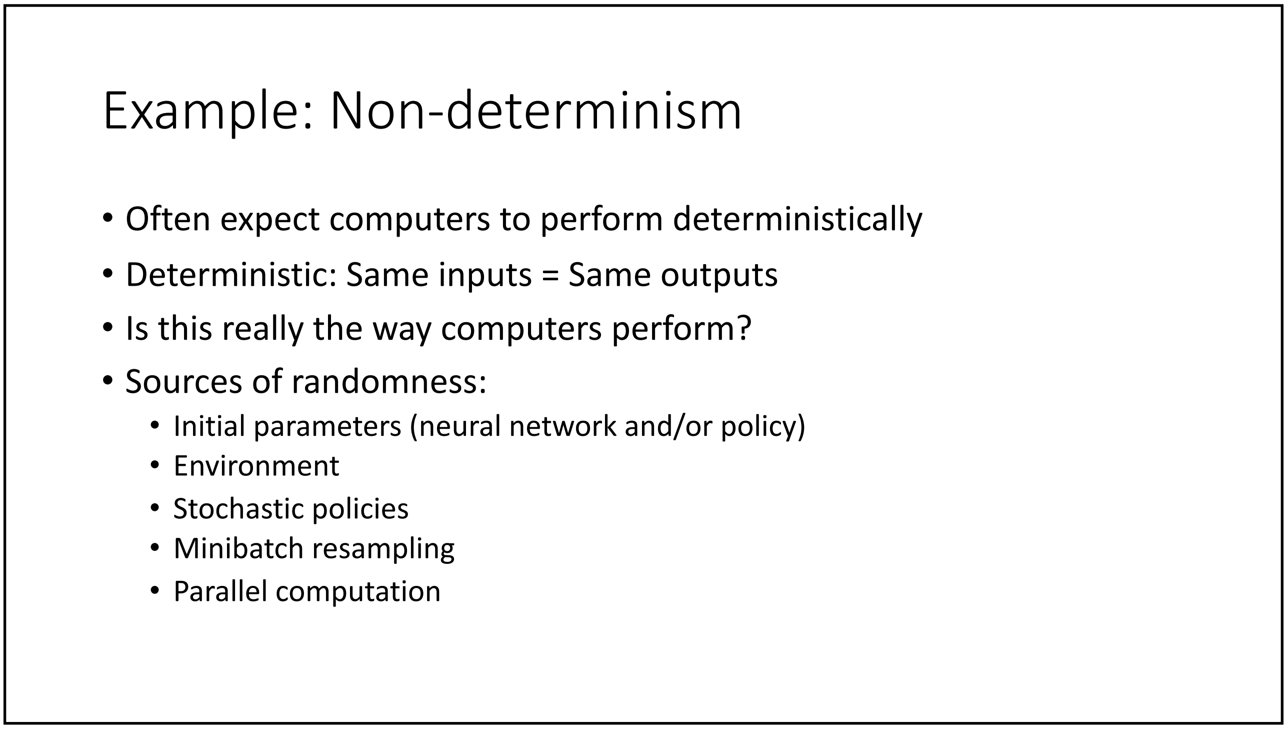
"If you teach a dog to talk, the reviewers won't complain that n=1."

Are things getting worse?

- Yes!
- Why?
- Reason 1 – Publication pressure
  - Rapidly growing community and high expectations for publication counts
  - Low reviewing quality, temptation
- Reason 2 - Deep learning:
  - Involves many random elements
  - Involves experiments that are expensive to repeat
  - Lack of awareness

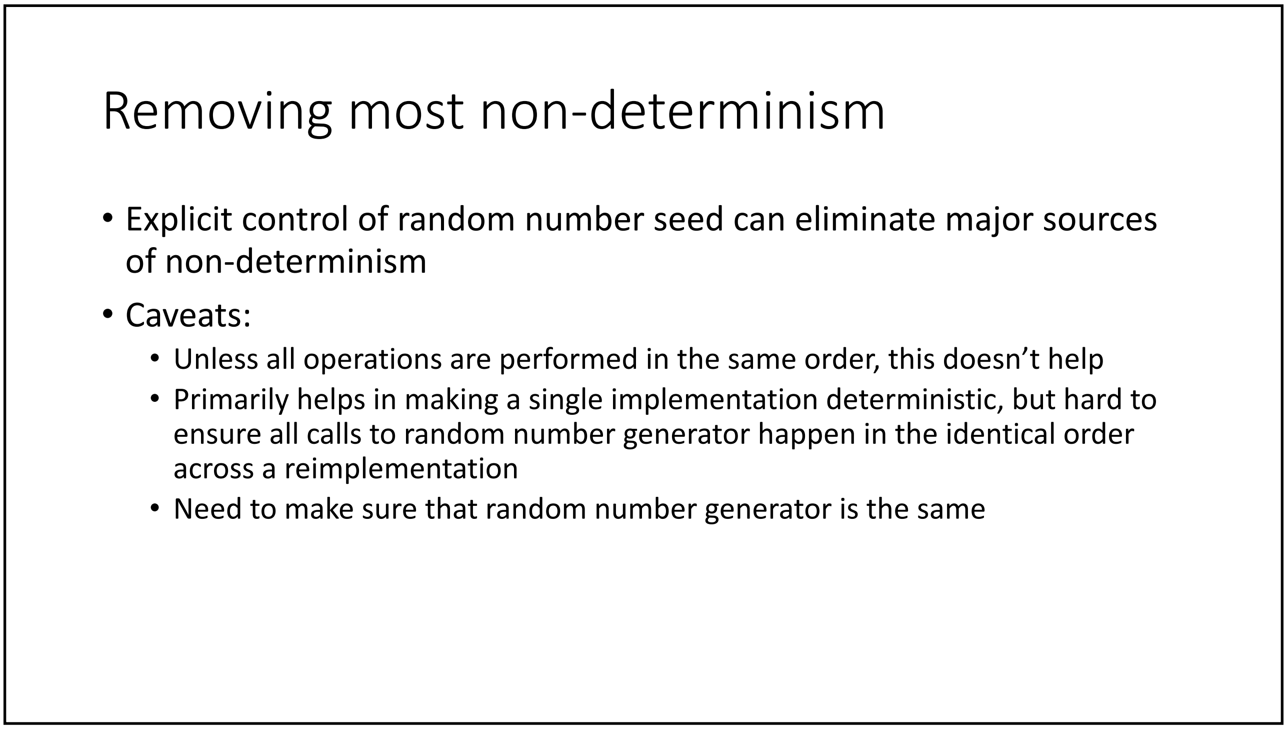
Is it worse for RL

- Yes!
- Why?
  - Experiments are particularly expensive (even by deep learning standards)
  - Variance is very high!



Example: Non-determinism

- Often expect computers to perform deterministically
- Deterministic: Same inputs = Same outputs
- Is this really the way computers perform?
- Sources of randomness:
  - Initial parameters (neural network and/or policy)
  - Environment
  - Stochastic policies
  - Minibatch resampling
  - Parallel computation



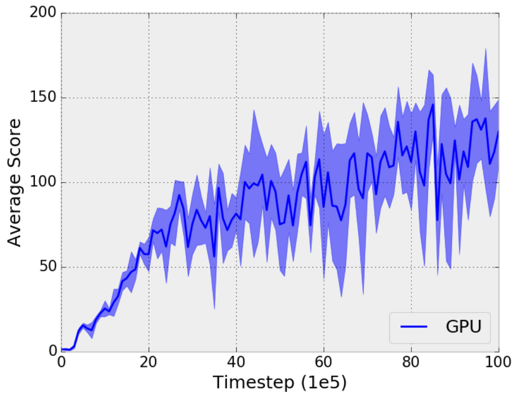
Removing most non-determinism

- Explicit control of random number seed can eliminate major sources of non-determinism
- Caveats:
  - Unless all operations are performed in the same order, this doesn’t help
  - Primarily helps in making a single implementation deterministic, but hard to ensure all calls to random number generator happen in the identical order across a reimplementation
  - Need to make sure that random number generator is the same

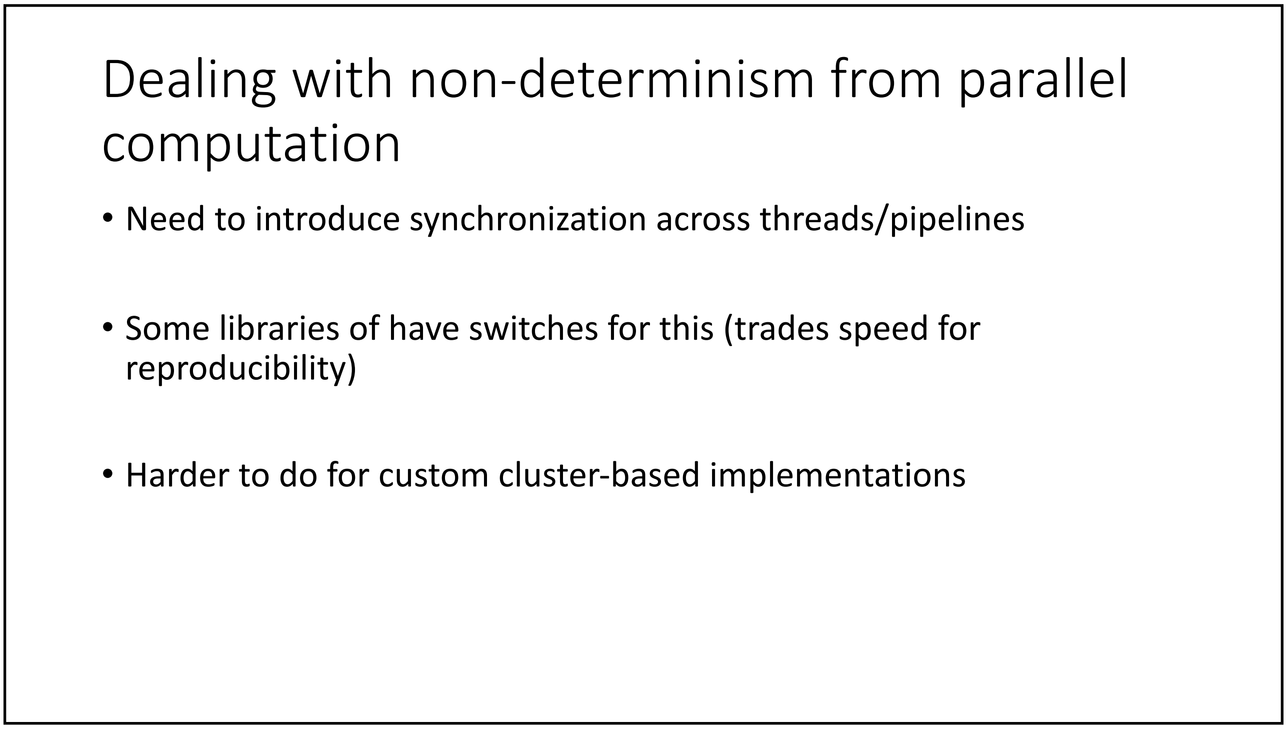
Non-determinism from parallel computation

- Some parallelized linear algebra or machine learning code is iterative, and based on *loosely coupled* parallel computations
- Often transparent to us because small non-determinisms may be below specified accuracy thresholds
- This issue can be magnified in Deep RL:
  - Most operations are done at low precision on GPUs
  - Tiny differences in influence action selection during exploration
  - A single different action choice can change what agent sees and change entire learning curve
- This issue gets even worse for algorithms that train in parallel across clusters of machines

Example of GPU variance

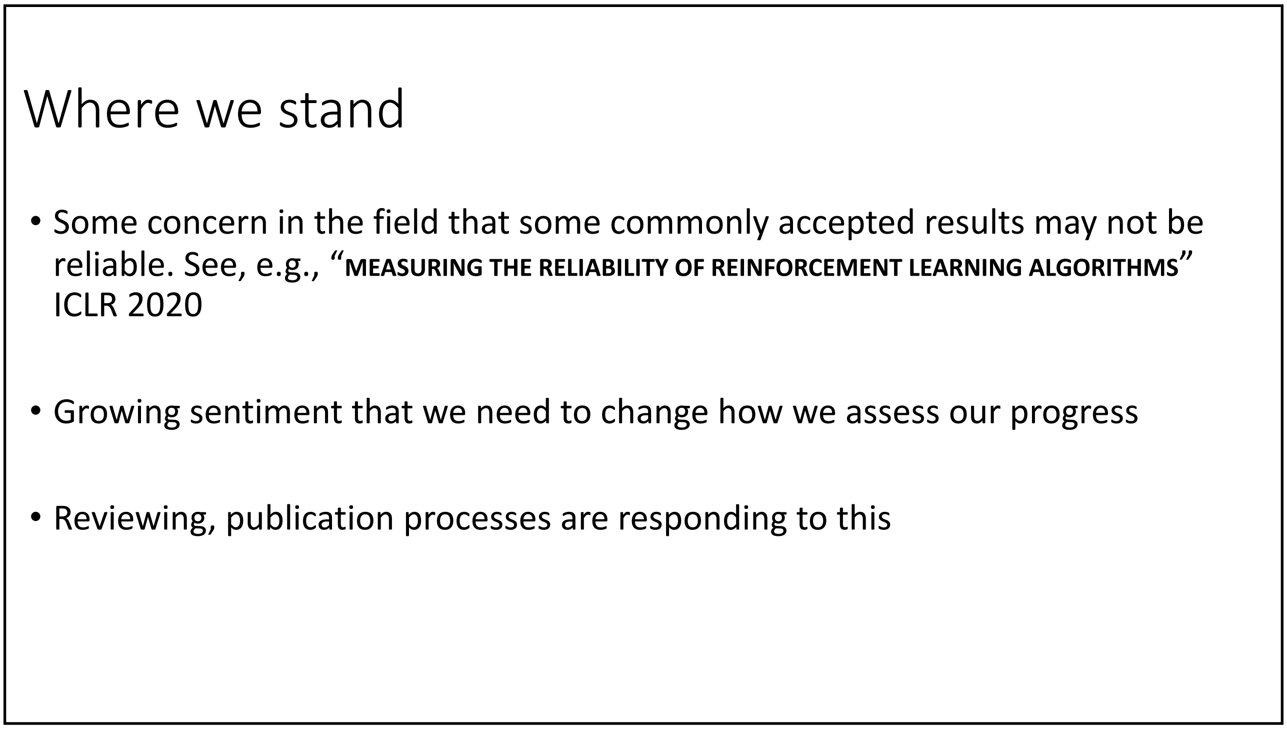


From Nagarajan et al. "The Impact of Nondeterminism on Reproducibility in Deep Reinforcement Learning"
Graph shows 1 SD



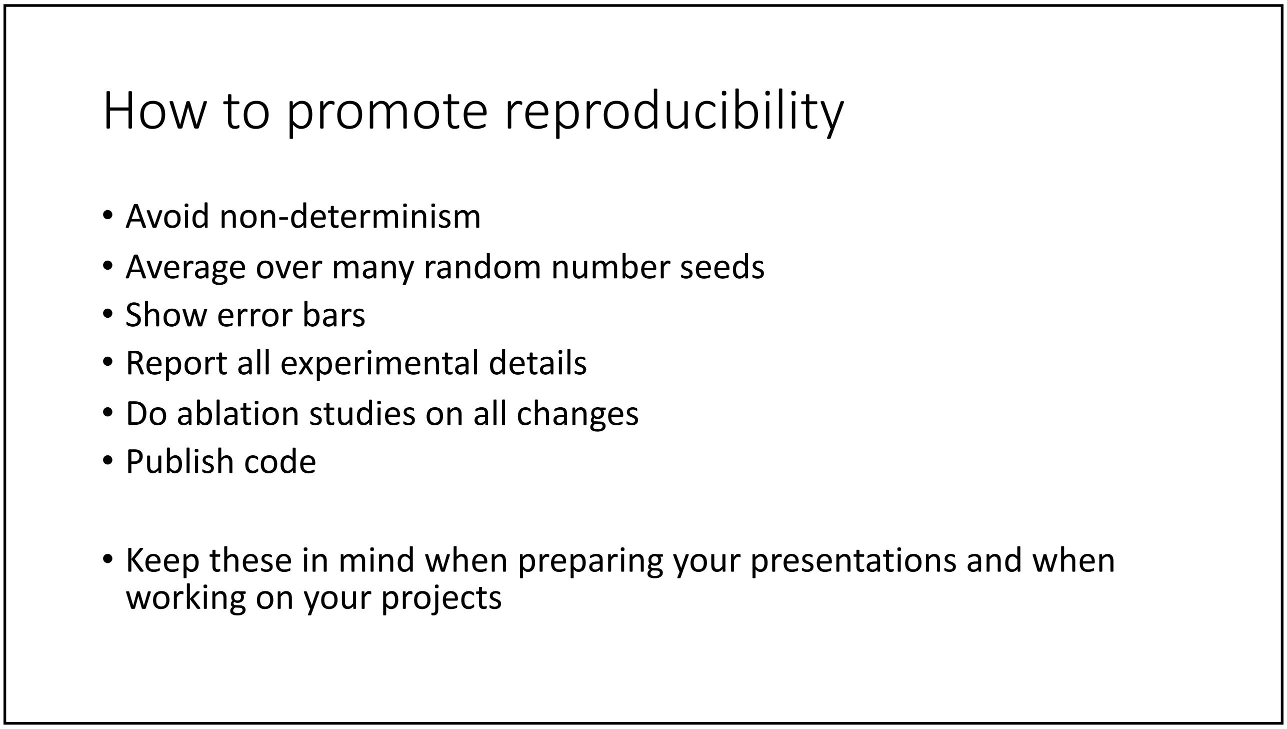
Dealing with non-determinism from parallel computation

- Need to introduce synchronization across threads/pipelines
- Some libraries of have switches for this (trades speed for reproducibility)
- Harder to do for custom cluster-based implementations



Where we stand

- Some concern in the field that some commonly accepted results may not be reliable. See, e.g., “MEASURING THE RELIABILITY OF REINFORCEMENT LEARNING ALGORITHMS” ICLR 2020
- Growing sentiment that we need to change how we assess our progress
- Reviewing, publication processes are responding to this



How to promote reproducibility

- Avoid non-determinism
- Average over many random number seeds
- Show error bars
- Report all experimental details
- Do ablation studies on all changes
- Publish code

- Keep these in mind when preparing your presentations and when working on your projects