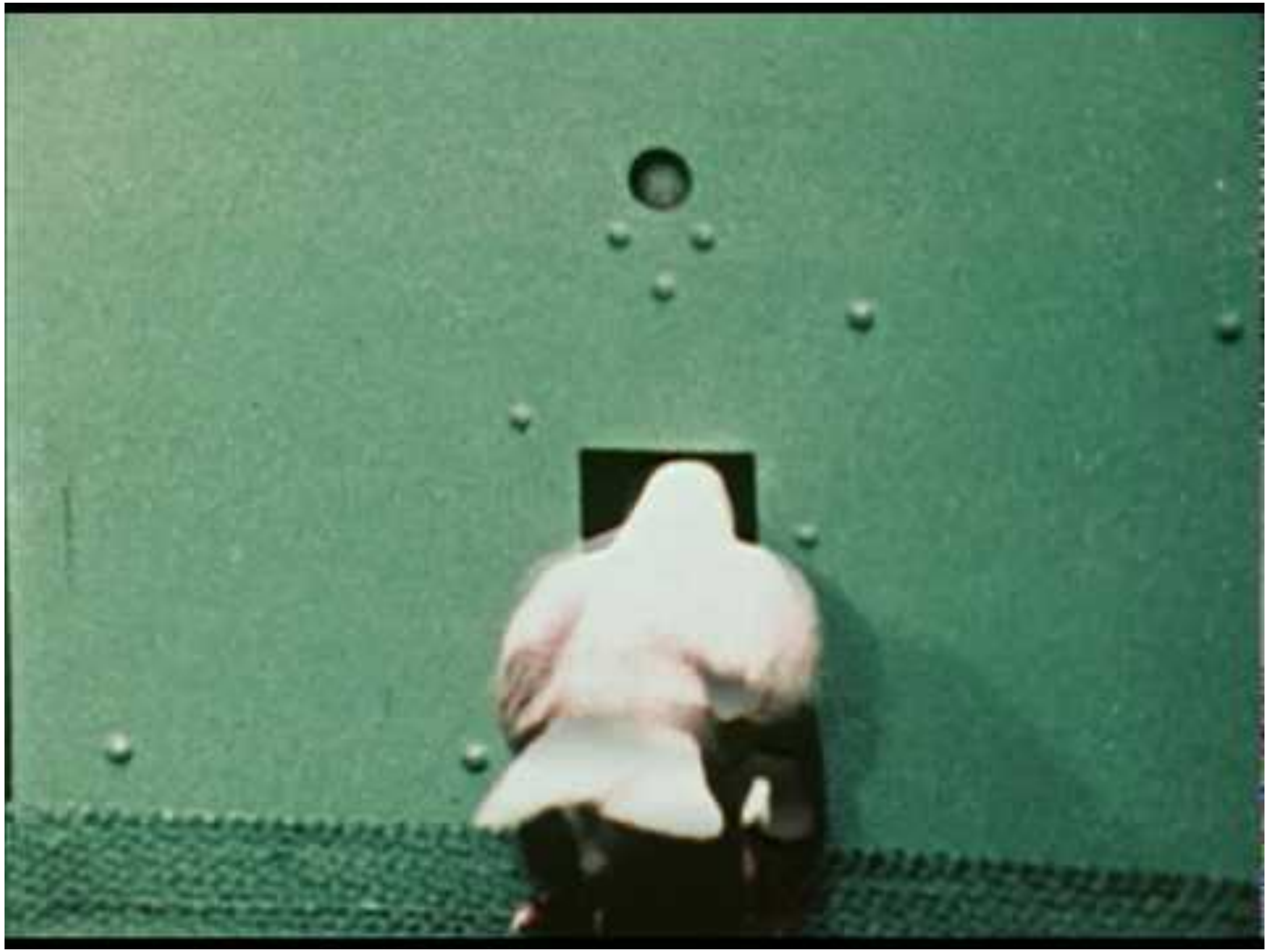
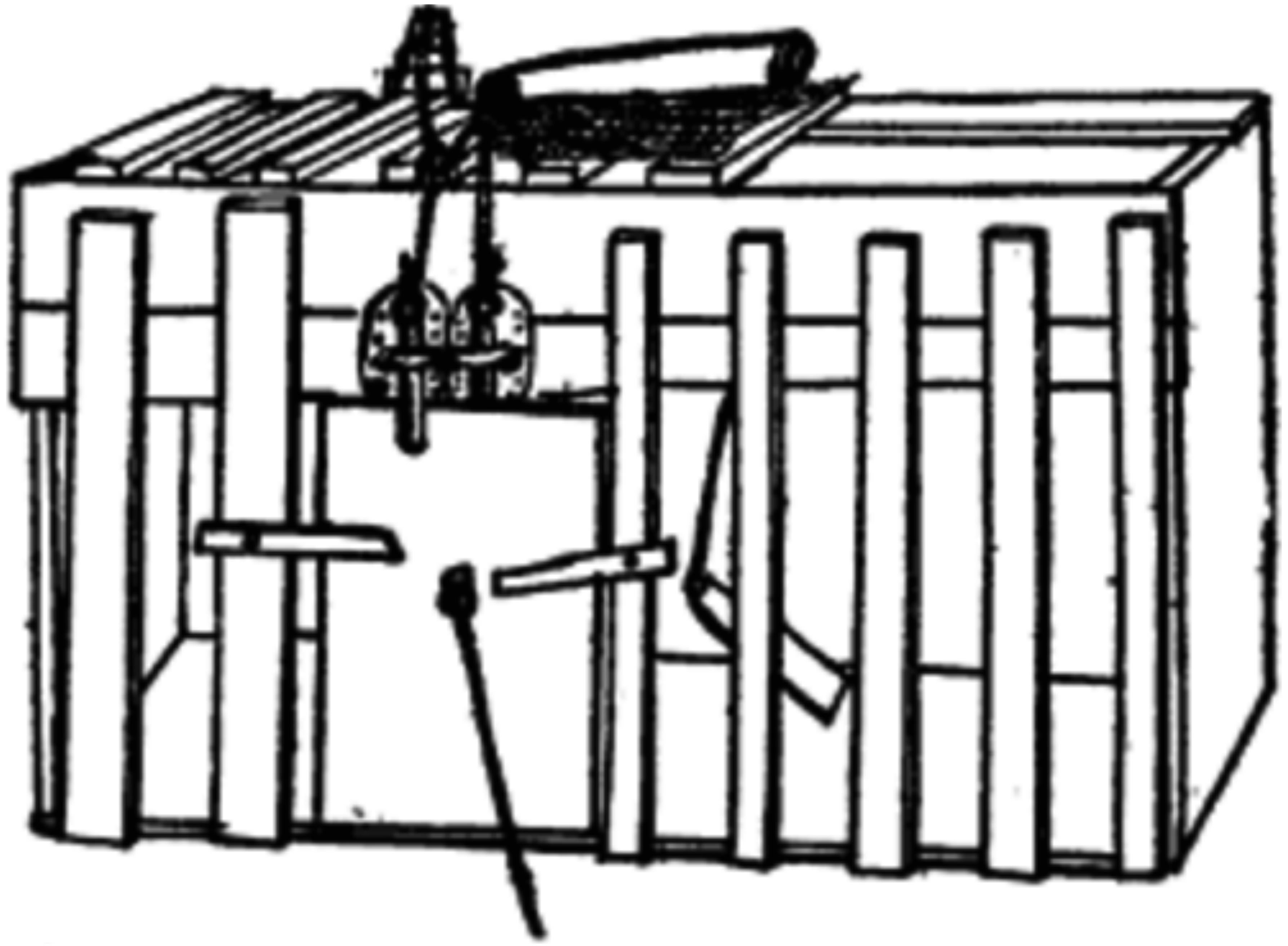
Learning machines



Shaping: teaching animals via the method of successive approximation

Learning machines



One of Thorndike's puzzle boxes.

Reprinted from Thorndike, Animal Intelligence: An Experimental Study of the Associative Processes in Animals, *The Psychological Review, Series of Monograph Supplements* II(4), Macmillan, New York, 1898.

Investigating operant conditioning

Start recording …

Admin

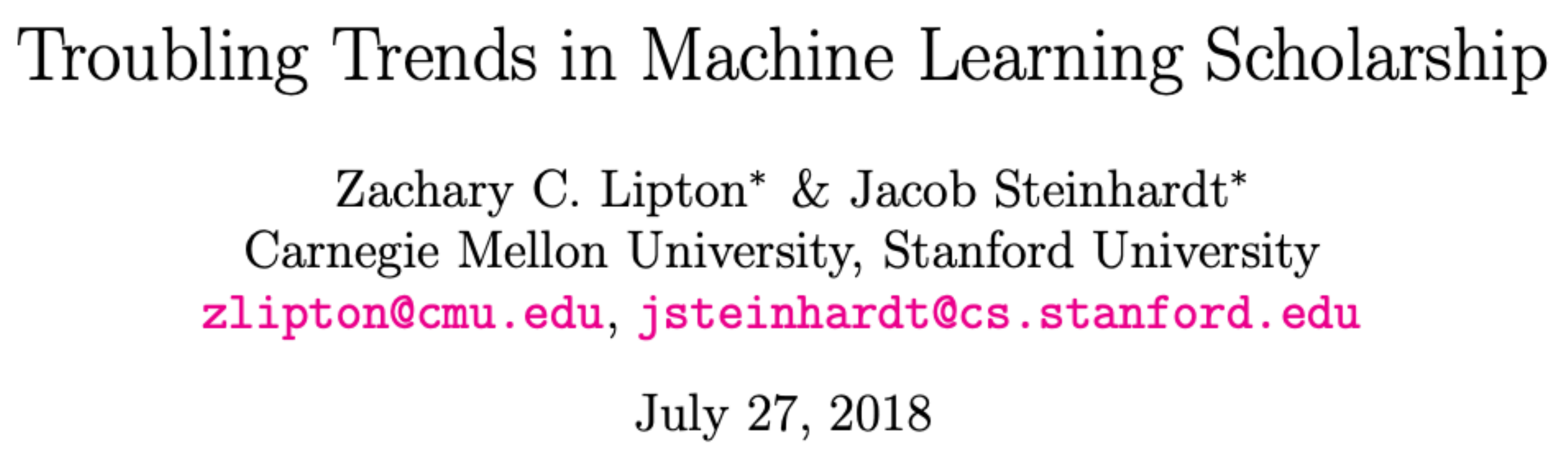
- Project team list sheet:
  - https://docs.google.com/spreadsheets/d/1f-QybvJk5V5dilsHOL9f6eNx5fAR0gFEuUekHS94MhE
- Session moderators for today: **Tata, Ganesh & Lo,Chunlok**
  - https://docs.google.com/spreadsheets/d/1dbmlvduupZUCDjxU4HW2_350OVrVG-g1FoEAG-uWhMk

A tale of two papers

~~The Good,~~ the bad, and the ugly

Two papers about methodology and scholarship in RL

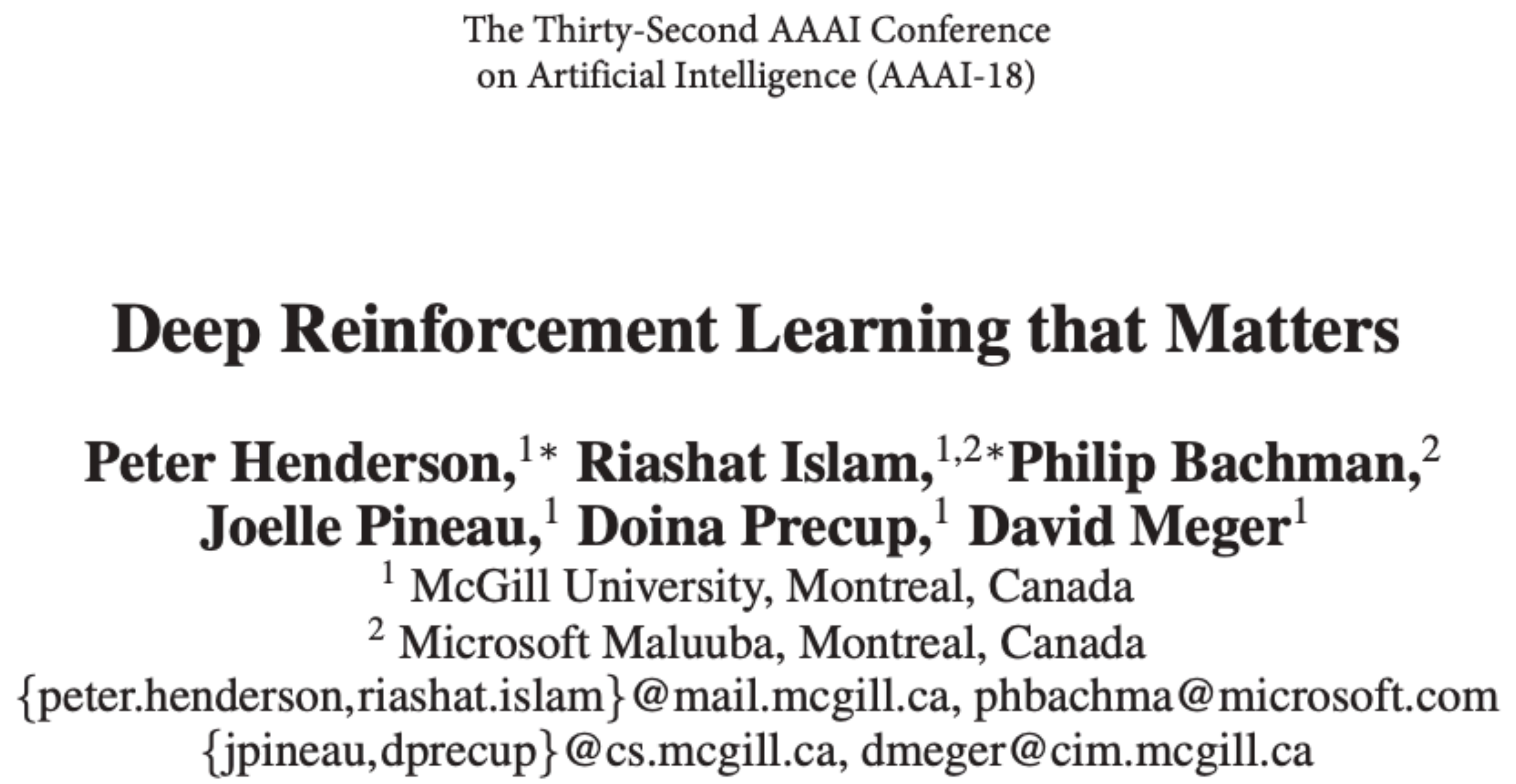
- Each paper focuses on a different sub community
  - Classical batch supervised learning
  - People who work in Continuous action RL, specifically Deep RL approaches
- Each paper has a different emphasis
  - Overall trends and motivations in the community
  - Whats wrong and how to do it better



Troubling Trends in Machine Learning Scholarship

Zachary C. Lipton* & Jacob Steinhardt*
Carnegie Mellon University, Stanford University
zlipton@cmu.edu, jsteinhardt@cs.stanford.edu

July 27, 2018



The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Deep Reinforcement Learning that Matters

Peter Henderson,[1*] **Riashat Islam,**[1,2*]**Philip Bachman,**[2]
Joelle Pineau,[1] **Doina Precup,**[1] **David Meger**[1]
[1] McGill University, Montreal, Canada
[2] Microsoft Maluuba, Montreal, Canada
{peter.henderson,riashat.islam}@mail.mcgill.ca, phbachma@microsoft.com
{jpineau,dprecup}@cs.mcgill.ca, dmeger@cim.mcgill.ca

889 citations between all three

Today’s focus is mostly about what is wrong

- Both papers focus on the most pressing problems
- They give specific examples—literally pointing to particular papers, codebases, statements, and experiments
- We will also talk about specific examples I have come across in RL
- These papers are a bit light on actionable fixes
- **Next lecture we will discuss three proposed ways to do better experiments in RL**

Troubling trends in ML (2018)

What we should be doing

- Researchers could have many goals:
    - Theoretically characterize what is learnable
    - Obtain understanding via rigorous experiments
    - Build a high performance (or high accuracy) system
- Any paper should aspire to do one of the following:
    - Provide understanding but not make claims not supported by the data
    - Use experiments to rule out hypotheses
    - Connect empirical claims with intuition and theory
    - Use language and terminology to minimize misunderstanding, conflation, unsupported claims, and hype

Troubling trends in ML (2018)

What we see in the literature

- Failure to distinguish between speculation and explanation
- Failure to identify sources of gain (improvement) in experiments
  - e.g., explaining architecture improvements vs of hyper-parameter tuning
- Using math to impress and confuse the reader
- Using language poorly: overload established terms or using fancy word with particular English meanings to suggest something about your algorithm
  - e.g., The *dreamer* agent is *curious* about its world …

Troubling trends in ML (2018)

Why is this happening

- “Strong results excuse weak arguments”
- ML and RL is growing rapidly, these things happen during periods of growth
- Less qualified reviewers due to growth
    - Way more lower quality submissions, more junior reviews proportionally
- Bad incentive structures
- These are symptoms of our success, not the cause of success
- Flawed papers get thousands of citations

Troubling trends in ML (2018)

The consequences

- Regardless of the reasons we should all care because ML is being deployed in the real world, and thus our papers are read by non-scientists too:
  - Students, application engineers, policy-makers, journalists
- We risk lab shutdowns, erosion of public and government trust
- In psychology, poor empirical standards have eroded public trust
- Even in AI this is an old and cyclic problem:
  - “Dermott (in 1976) chastised the AI community for abandoning self-discipline, warning prophetically that `if we can’t criticize ourselves, someone else will save us the trouble’.”

Disclaimers

This was written by insiders

- No students were hurt in the making of this paper

Explanation vs Speculation

Don’t pretend they are the same

- Example *covariate shift:* “It is well-known that a deep neural network is very hard to optimize due to the internal-covariate-shift problem.”
    - In the original paper this was an intuitive concept that was never technically defined nor was batch normalization ever clearly demonstrated to mitigate it
    - Later work suggested that this explanation was not correct
    - But the myth persists
- More generally claims without an experiment to support them
- Introducing terms that appear technical (but lack definition) and then using them to define other things

Explanation vs Speculation

Example from RL

- “In this work we show that an algorithm that supports continual learning—which takes inspiration from neurobiological models of synaptic consolidation—can be combined with deep neural networks to achieve successful performance in a range of challenging domains. In doing so, we demonstrate that current neurobiological theories concerning synaptic consolidation do indeed scale to large-scale learning systems. This provides prima facie evidence that these principles may be fundamental aspects of learning and memory in the brain”

Motivation, speculation and explanation can all be used

With care

- Tell the reader when you are motivating your ideas for outside inspirations
- Tell the reader when you are speculating
- Paper gives a nice example of how one paper talks at length about how dropout might be inspired by sexual reproduction
- Another example involves conveying uncertainty:
    - “Although such recommendations come. . . from years of experimentation and to some extent mathematical justification, they should be challenged. They constitute a good starting point. . . but very often have not been formally validated, leaving open many questions that can be answered either by theoretical analysis or by solid comparative experimental work”

Failure to identify sources of empirical gains

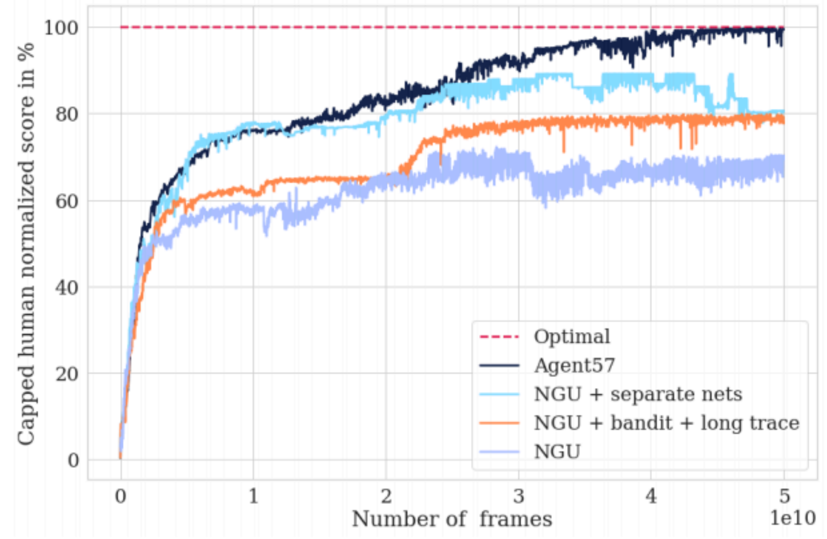
Do you really understand what is going on?

- Complex architectures and models are popular
- Advances often come from: simplifications, unifications, new problem formulations, and empirical insights
- Often advances come from the follow recipe, add:
    - Optimization heuristics, hyper-parameter tuning, data preprocessing, minor architecture changes, recent fancy algorithm adapted to your new environment
    - Outcome: SOTA performance!!
- Sometimes all these parts are needed, sometimes not. It’s our job to figure it out

Failure to identify sources of empirical gains

The mistakes

- Many tweets, tricks, and algorithmic changes but no ablations, not parameter studies
- If only one of those things matters, but you don’t clarify which thing matters you get credit for k novel contributions!!
  - The opposite is true: they didn’t do enough work!
- Example from the paper: claimed neural net architecture changes were not key for performance, it was hyper-parameter tuning
- Examples from RL:
  - Agent57: best across all Atari games compared to R2D2, NGU, MuZero…
    - Using dynamic discounting, adjusting T in T-BPTT, intrinsic rewards, new network architecture, meta-controller…all built on top of NGU…which combines UVFAs, re-trace, Double Q-learning, intrinsic rewards, many parallel exploration policies ….
    - Some ablations…but not tuning of the ablations



There are many ways to understand the gains

This often happens in followup work

- Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. *How does batch normalization help optimization? (no, it is not about internal covariate shift)*
- Implementation details matter: https://arxiv.org/abs/2005.12729
- Action dependent baselines don't do what you think: http://proceedings.mlr.press/v80/tucker18a.html
- Simple linear baselines are SOTA in AIGym: https://arxiv.org/abs/1703.02660

Mathiness

Paper needs more theory

- Sutton et al's Emphatic TD paper was rejected because the prose did no follow the usual lemma, theorem flow
- I have had papers rejected because the reviewers thought the theory was not interesting because the proof was not complex—they didn't even care about the statements
- Theory and formalism is an essential tool for expressing complex things clearly and compactly
  - See philosophy …
- Math and theory should aid the reader in understanding the paper, not the opposite
- Using unnecessary theory or math is like using complex (big) words or phrases to sound impressive

Mathiness

Theory can be used for evil too

- Weak arguments, bad ideas, weak empirical evidence propped up by complex math
- Spurious theorems:
  - that don’t support the main ideas of the paper
  - prove stability or convergence in a setting of little interest—assumptions too restrictive
- Famous example: paper introducing the Adam optimizer
  - Empirical paper with strong empirical support for the new method
  - Including a convergence proof—that turned out to be wrong
- Imprecise statements that suggest formal backing, other via citations

Poor use of language

Suggestive definitions

- Introduce a new technical term with a word that has an English meaning that is strongly suggestive of what you want the reader to think about your agent
    - “Curiosity Agent”, “Dreaming”
- Using such words to describe agent performance:
    - “Human-level”, “super-human”: false sense of current abilities—only true on games it was trained on
    - Popular articles continue to characterize modern image classifiers as “surpassing human abilities and effectively proving that bigger data leads to better decisions”
        - In practice you can make tiny changes to a stop sign and the agent will classify it as “40 MPH”

Poor use of language

Overloading

- Changing the established meaning of a technical term E.g.:
  - calling every Q-learning agent DQN
  - Generative models: models of the input distribution p(x) or the joint p(x,y)
    - Not any model that produces realistic-looking structured data
- And the opposite can happen, new terms introduced:
  - Artificial General Intelligence (AGI) vs Artificial Intelligence (AI)

Poor use of language

Suitcase Words

- Words that are use to refer to a broad range or collection of ideas
- Coined by Minsky (one of the creators of Reinforcement Learning)
- Words with no generally agreed-upon meaning
- Examples specific to RL:
    - “Model”: is it an estimate of the one step dynamics or just any NN?
    - “Optimizer”: step-size adaption algorithm? Concept form math? Name from tensorflow?
    - Not using language and notation to differentiate General Value Functions (GVF) and approximate learned GVFs

Deep RL that Matters

What is going on in AIGym and continuous control?

- Focused on continuous action, policy gradient methods
- Critical evaluation of current empirical practices
- Critical evaluation of repeatability, stability, and general usefulness of current methods

Deep RL that Matters

What is going on in AIGym and continuous control?

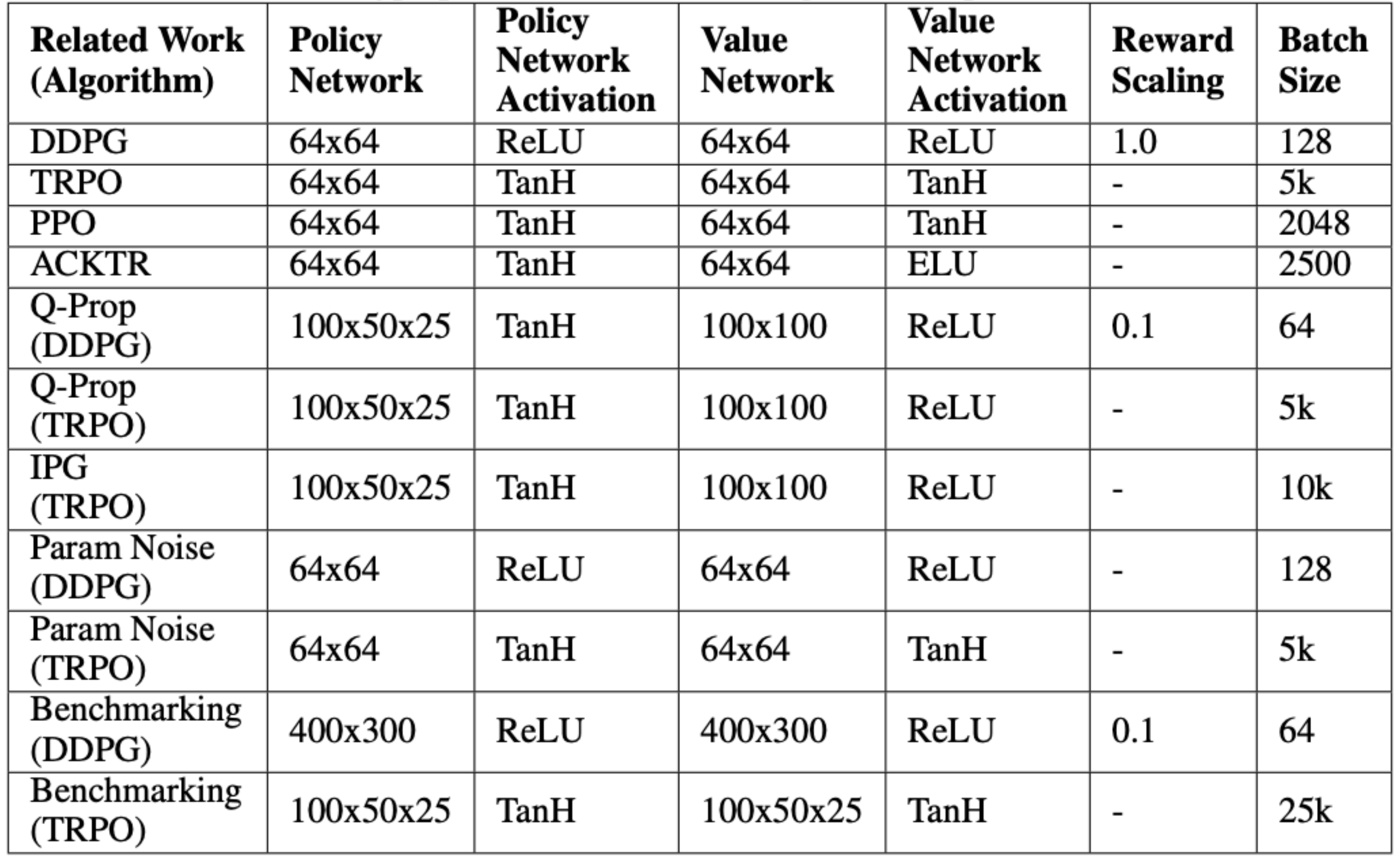
- AIGym domains require control of simulated robots with many degrees of freedom and high-dimensional inputs (joint angles and velocities)
- Motivated by conflicting empirical results found in the literature
- Reproducibility seems low priority and difficult
- Focused on continuous action, policy gradient methods
- Critical evaluation of current empirical practices
- Critical evaluation of repeatability, stability, and general usefulness of current methods

Dealing with hyper-parameters

…or not

- Has a big impact on performance of baselines
- Ranges of search (often informal) are not typically reported

Table 4: Evaluation Hyperparameters of baseline algorithms reported in related literature



Related Work (Algorithm)	Policy Network	Policy Network Activation	Value Network	Value Network Activation	Reward Scaling	Batch Size
DDPG	64x64	ReLU	64x64	ReLU	1.0	128
TRPO	64x64	TanH	64x64	TanH	-	5k
PPO	64x64	TanH	64x64	TanH	-	2048
ACKTR	64x64	TanH	64x64	ELU	-	2500
Q-Prop (DDPG)	100x50x25	TanH	100x100	ReLU	0.1	64
Q-Prop (TRPO)	100x50x25	TanH	100x100	ReLU	-	5k
IPG (TRPO)	100x50x25	TanH	100x100	ReLU	-	10k
Param Noise (DDPG)	64x64	ReLU	64x64	ReLU	-	128
Param Noise (TRPO)	64x64	TanH	64x64	TanH	-	5k
Benchmarking (DDPG)	400x300	ReLU	400x300	ReLU	0.1	64
Benchmarking (TRPO)	100x50x25	TanH	100x50x25	TanH	-	25k

Many design choices have significant impact on the performance of PG learners

Network architectures matter

…as do activation functions

- Dramatic performance differences are possible

- These things are interconnected and don't generalize across algorithms and environment

    - PPO with a large network may require tuning the trust region clipping or learning rate to compensate for the bigger net

.

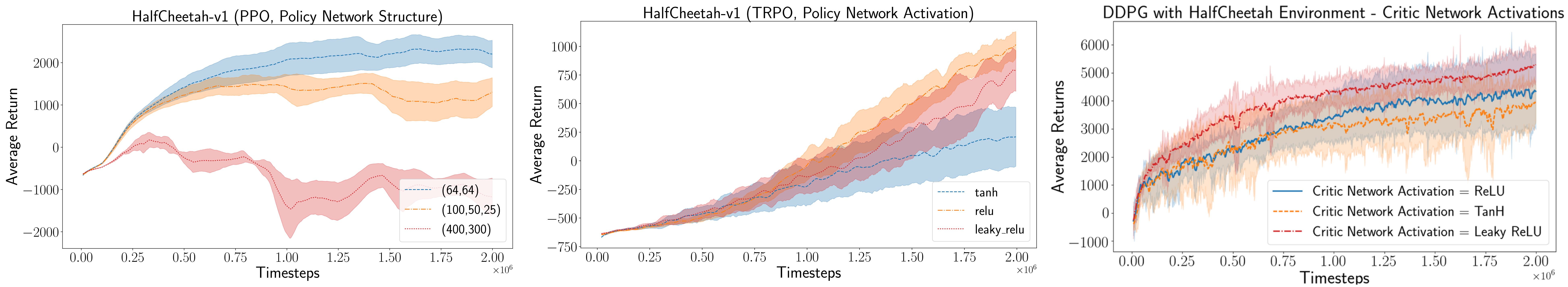


Figure 2: Significance of Policy Network Structure and Activation Functions PPO (left), TRPO (middle) and DDPG (right).

Reward scaling

…as do activation functions

- Multiplying the reward by a scalar during training
- Big effect but not consistent: sometimes failure to learn
- Neural Nets don’t like large magnitude targets (also the motivation for clipping in DQN)
- More principle approaches like PopArt (van Hasselt, 2016)

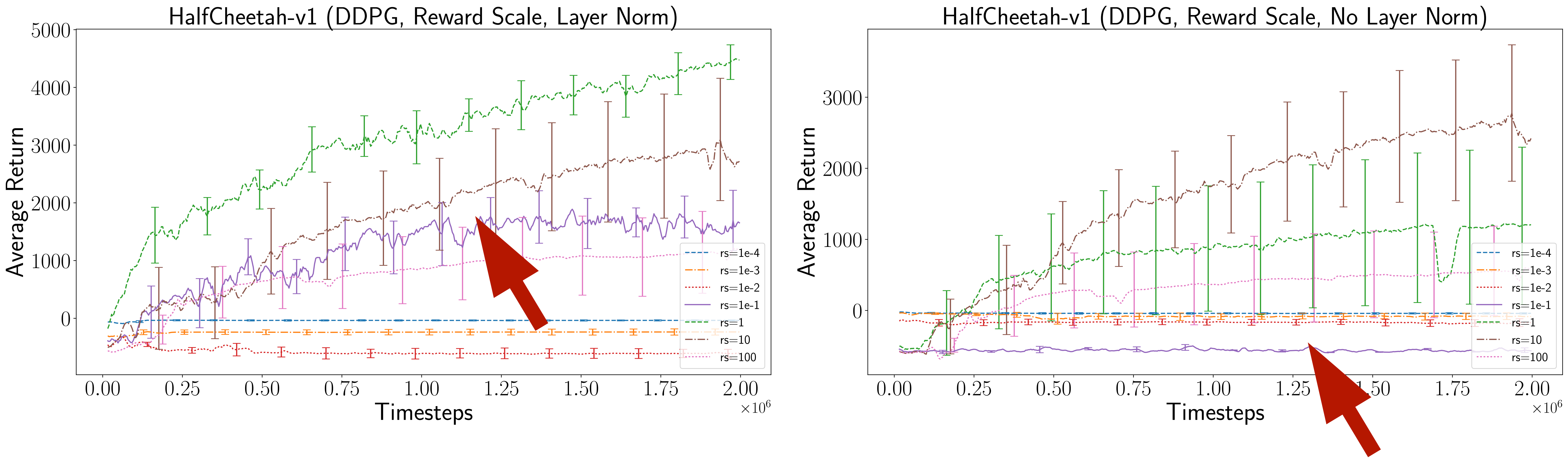
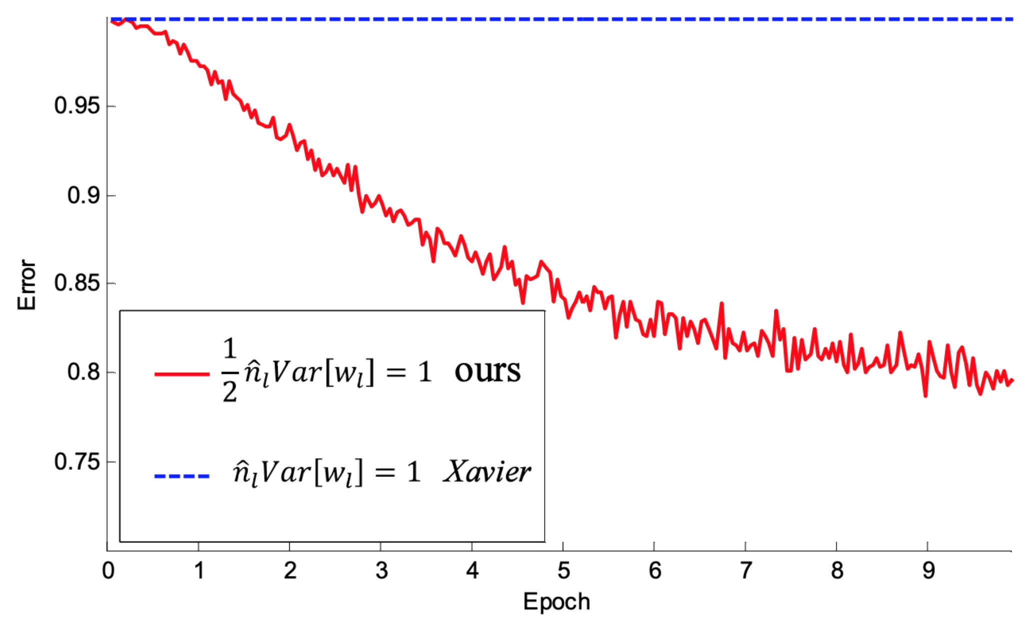


Figure 3: DDPG reward rescaling on HalfCheetah-v1, with and without layer norm.

Do seeds and number of runs matter?

Of course they do

- Neural networks require particular randomness to learn
- The environment, init, and policy can all be stochastic
- How many runs to we need?



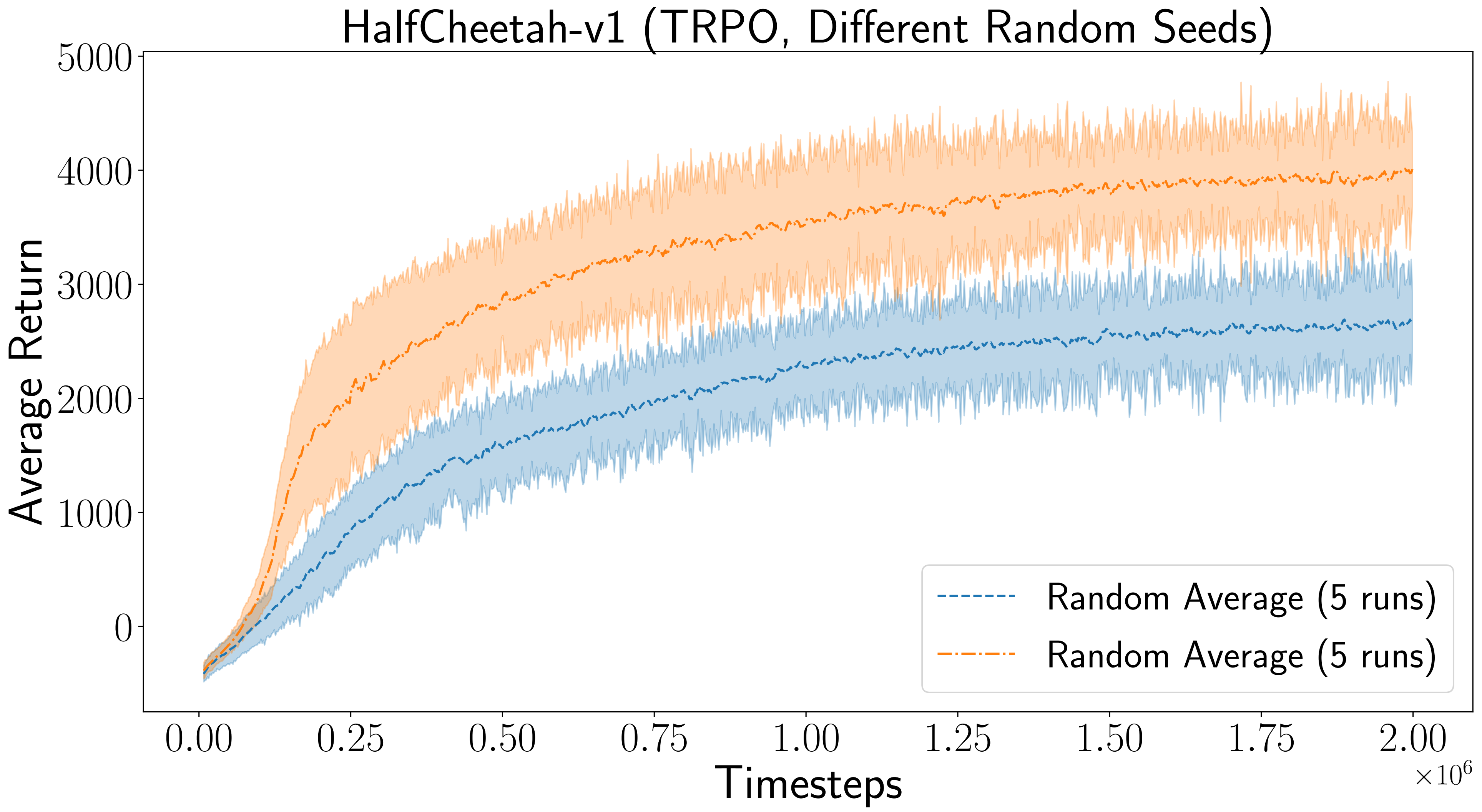
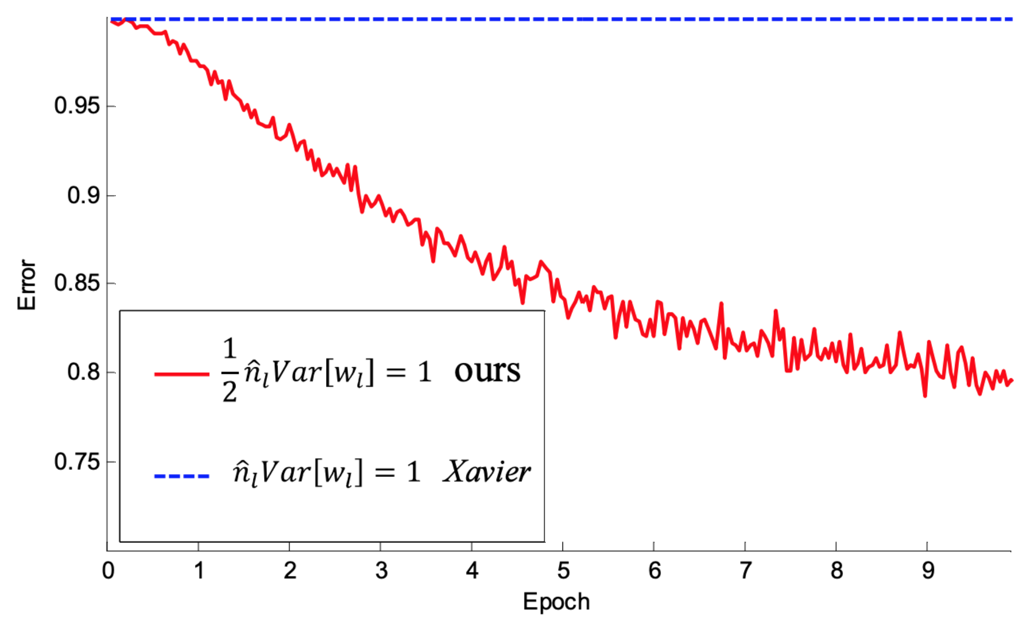


Figure 5: TRPO on HalfCheetah-v1 using the same hyperparameter configurations averaged over two sets of 5 different random seeds each. The average 2-sample t-test across entire training distribution resulted in $t = -9.0916$, $p = 0.0016$.

Do seeds and number of runs matter?

Of course they do



- Neural networks require particular randomness to learn
- The environment, init, and policy can all be stochastic
- How many runs to we need?
- What might be going on here?

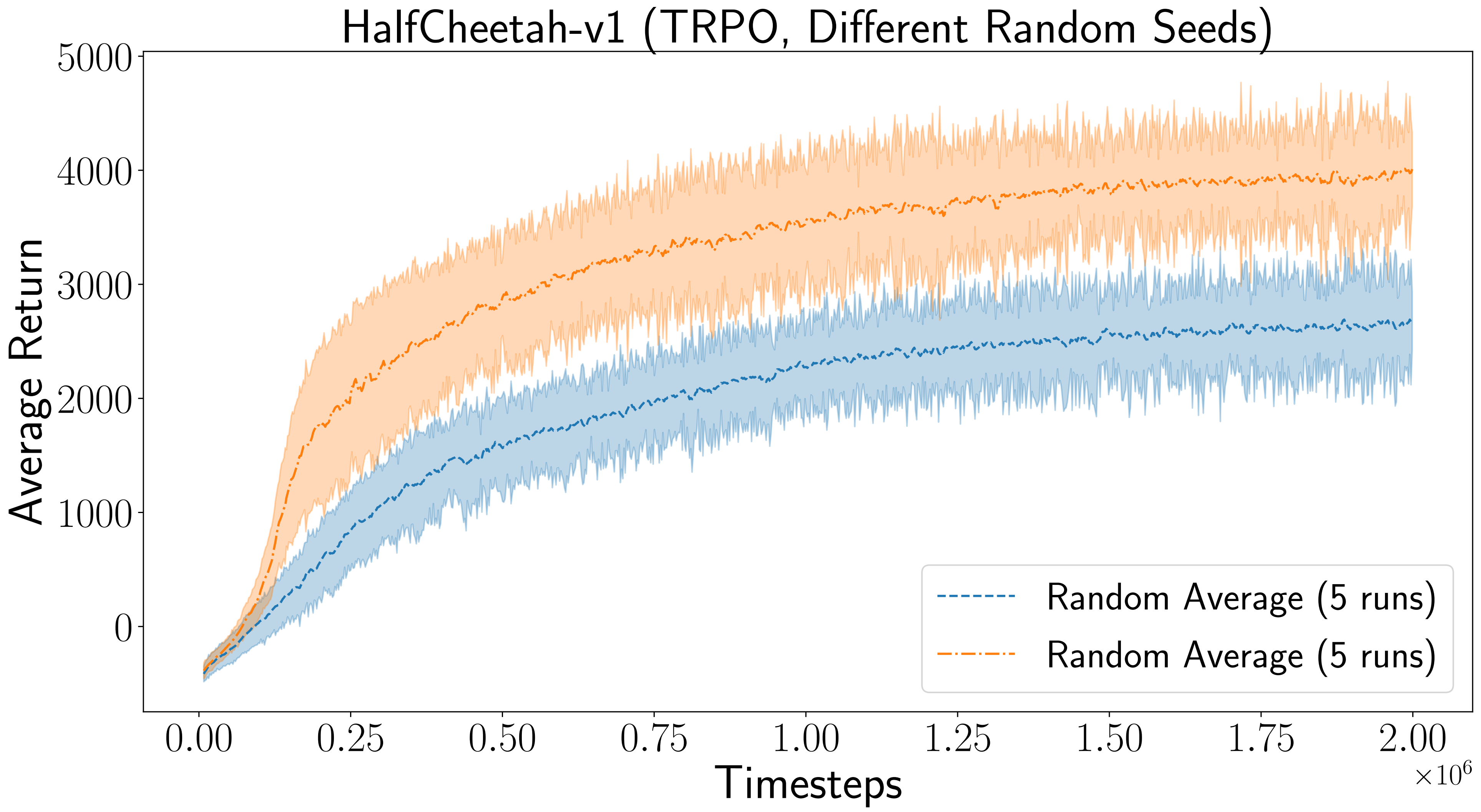


Figure 5: TRPO on HalfCheetah-v1 using the same hyperparameter configurations averaged over two sets of 5 different random seeds each. The average 2-sample t-test across entire training distribution resulted in $t = -9.0916$, $p = 0.0016$.

Do seeds and number of runs matter?

Common bad practices

- Top N runs among >N runs
- Max performance across runs
- Statistics ignoring “failure runs”
- Using a sub-set of an unspecified number of runs

Environments have a large impact

We are far from truly general agents

What is 130 return in swimmer? Curling up, flailing and not swimming

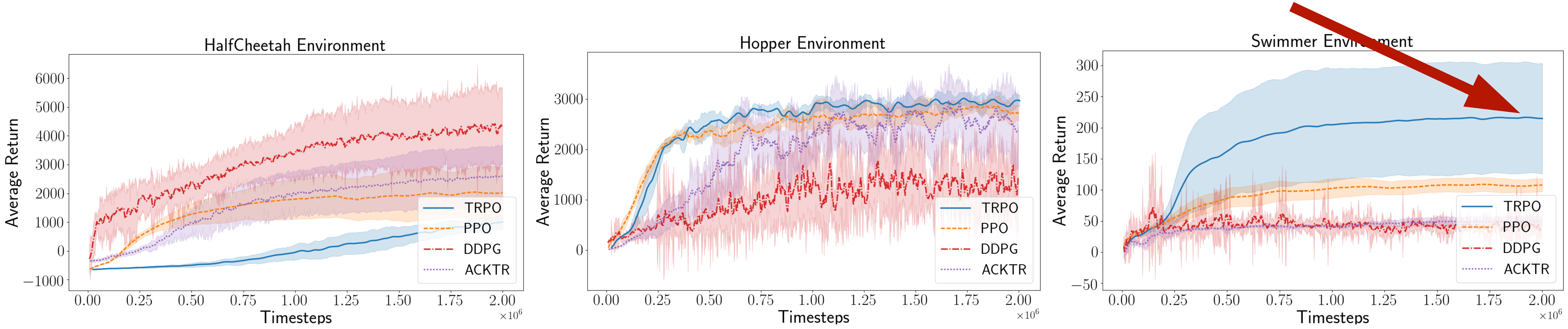


Figure 4: Performance of several policy gradient algorithms across benchmark MuJoCo environment suites

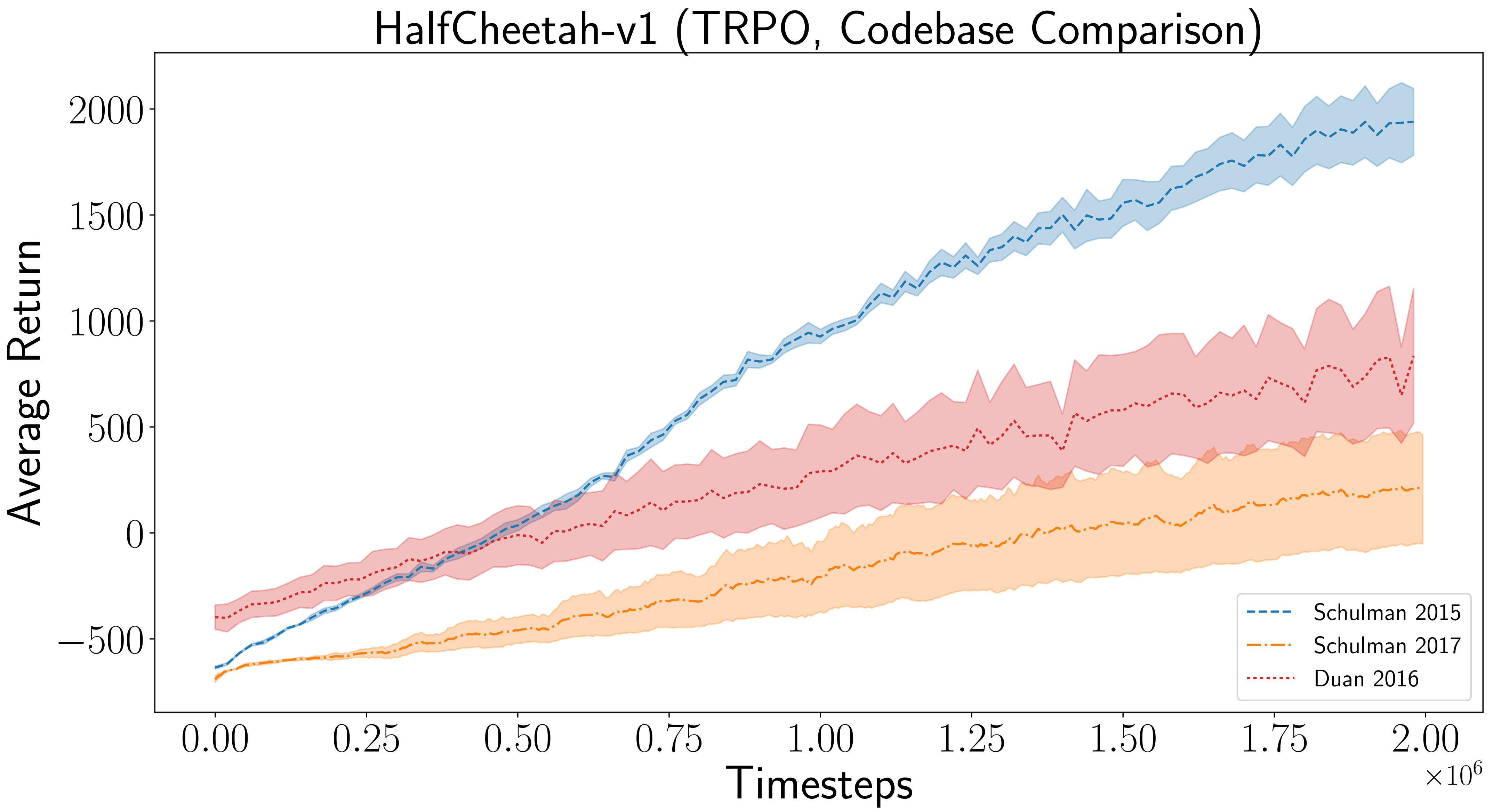
Environment	DDPG	ACKTR	TRPO	PPO
HalfCheetah-v1	5037 (3664, 6574)	3888 (2288, 5131)	1254.5 (999, 1464)	3043 (1920, 4165)
Hopper-v1	1632 (607, 2370)	2546 (1875, 3217)	2965 (2854, 3076)	2715 (2589, 2847)
Walker2d-v1	1582 (901, 2174)	2285 (1246, 3235)	3072 (2957, 3183)	2926 (2514, 3361)
Swimmer-v1	31 (21, 46)	50 (42, 55)	214 (141, 287)	107 (101, 118)

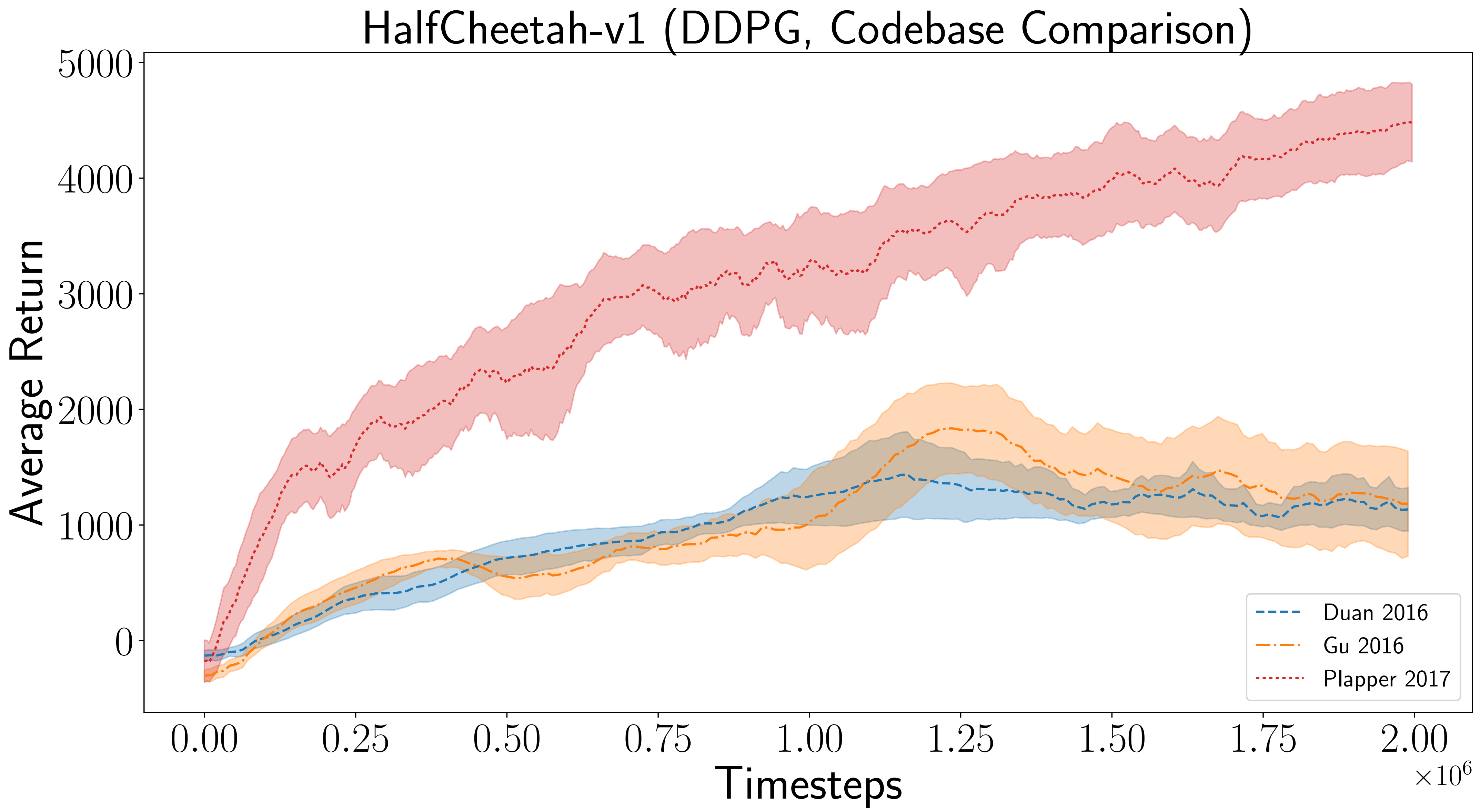
Table 3: Bootstrap mean and 95% confidence bounds for a subset of environment experiments. 10k bootstrap iterations and the pivotal method were used.

Code bases matter

Devil is in the details…or the SOTA is in the python …

- TRPO: OpenAI code, code from the paper, rl-lab tensor flow code
- DDPG: relax Theano code, OpenAI code
- Differences in the implementations are often not reported in the papers





Evaluation criteria matter

Many possible interesting questions to consider

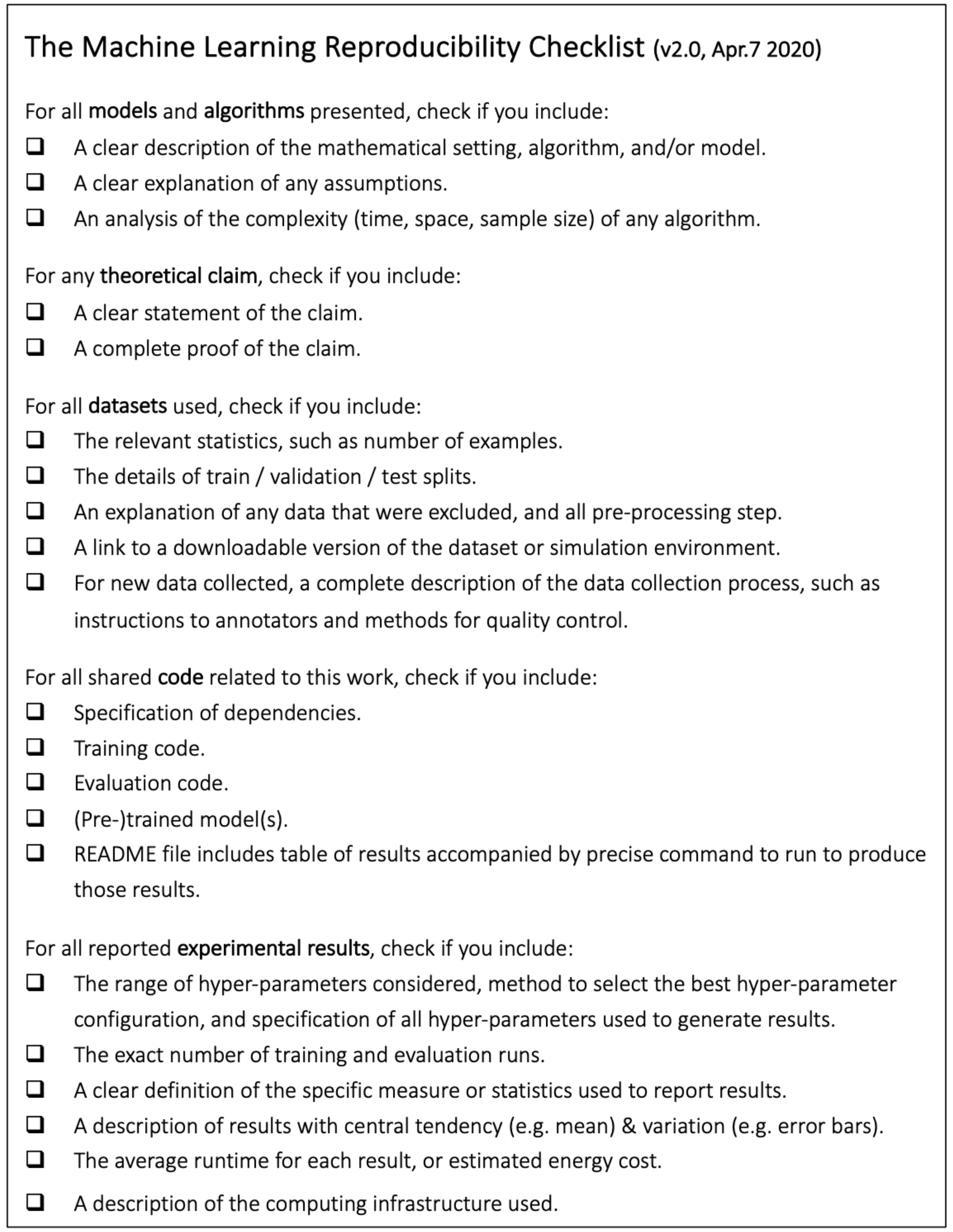
- Some algorithms are very unstable
- The mean can be very misleading (e.g., multimodal performance)
- What is our question: Online vs offline performance
    - Do we care about rewards as the agent’s learn?
    - Or the performance of the policy (without exploration) at the end?
- How do we measure variation and confidence?
    - Standard error and t-test, bootstrap CI, permutation test, sign test …
- What does the data even loop like anyway: what distributional assumptions are we making?

Henderson et al’s recommendations

We can do better

- Match the results in the literature as a first step
- Deal with hyper-parameters in a systematic way
- More runs
- Do significance tests
- Report all details: code optimizations, hyper parameter settings, setup, preprocessing, evaluation metrics for all algorithms tested
- We need algorithms that are less sensitive to their hyper-parameters
- Experiments should as a scientific question
- Maybe we should focus on real-world applications more (less game playing)?

Henderson et al motivated the creation of reproducibility checklists and requests for open sourcing code—what do you think?



The Machine Learning Reproducibility Checklist (v2.0, Apr.7 2020)

For all **models** and **algorithms** presented, check if you include:

- □ A clear description of the mathematical setting, algorithm, and/or model.
- □ A clear explanation of any assumptions.
- □ An analysis of the complexity (time, space, sample size) of any algorithm.

For any **theoretical claim**, check if you include:

- □ A clear statement of the claim.
- □ A complete proof of the claim.

For all **datasets** used, check if you include:

- □ The relevant statistics, such as number of examples.
- □ The details of train / validation / test splits.
- □ An explanation of any data that were excluded, and all pre-processing step.
- □ A link to a downloadable version of the dataset or simulation environment.
- □ For new data collected, a complete description of the data collection process, such as instructions to annotators and methods for quality control.

For all shared **code** related to this work, check if you include:

- □ Specification of dependencies.
- □ Training code.
- □ Evaluation code.
- □ (Pre-)trained model(s).
- □ README file includes table of results accompanied by precise command to run to produce those results.

For all reported **experimental results**, check if you include:

- □ The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results.
- □ The exact number of training and evaluation runs.
- □ A clear definition of the specific measure or statistics used to report results.
- □ A description of results with central tendency (e.g. mean) & variation (e.g. error bars).
- □ The average runtime for each result, or estimated energy cost.
- □ A description of the computing infrastructure used.