Deep RL at the edge of the statistical Precipice

NeurIPS 2021 (Oral)

agarwl.github.io/rliable

Google Research Mila

Google Research

TL;DR

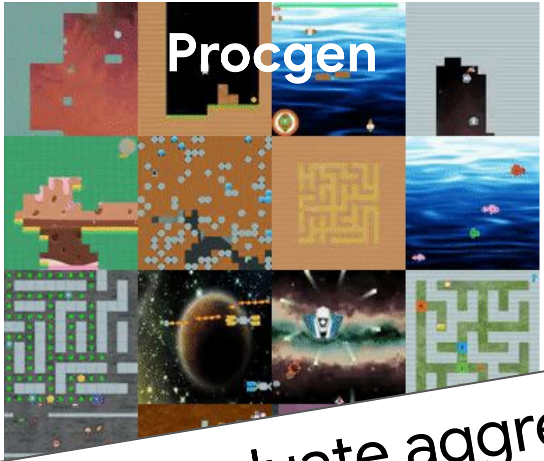
This work calls for a change in how we evaluate performance on reinforcement learning benchmarks, for which we present more reliable protocols, easily applicable with **even a handful of runs**, to prevent unreliable results from stagnating the field.

Few extra lines of code for reliable evaluation:

github.com/google-research/rliable

Google Research

Assessing Progress in Deep RL



Evaluate aggregate performance on a suite of tasks.

Point estimates are prevalent.

Distributional RL (ICML'17)

	Mean	Median	> H.B.	> DQN
DQN	228%	79%	24	0
DDQN	[illegible]	[illegible]	[illegible]	43
DUEL.	[illegible]	[illegible]	[illegible]	50
PRIOR.	434%	124%	39	48
PR. DUEL.	592%	172%	39	44
C51	**701%**	**178%**	**40**	**50**
UNREAL†	880%	250%	-	-

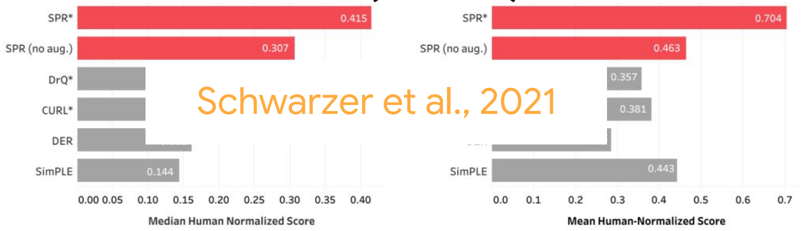
Bellemare et al., 2017

Offline RL (ICML'20)

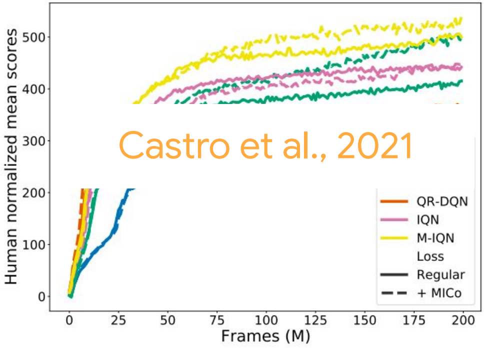
Offline agent	Median	> DQN
DQN (Nature)	83.4%	17
DQN ([illegible])	[illegible]	41
Ensem[illegible]	[illegible]	39
Averag[illegible]	[illegible]	43
QR-DQN	118.9%	45
REM	**123.8%**	**49**

Agarwal et al., 2020

Self-predictive representations (ICLR'21)



MiCo State Abstraction (NeurIPS'21)



Point estimates are prevalent.

Distributional RL (ICML’17)

	Mean	Median	> H.B.	> DQN
DQN	228%	79%	24	0
DDQN	[illegible]	[illegible]	[illegible]	43
DUEL.	[illegible]	[illegible]	[illegible]	50
PRIOR.	434%	124%	39	48
PR. DUEL.	592%	172%	[illegible]	[illegible]
C51	**701%**	[illegible]	[illegible]	[illegible]
UNREAL†	[illegible]	[illegible]	[illegible]	[illegible]

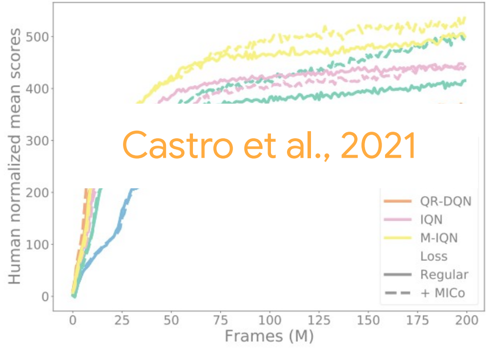
Bellemare et al., 2017

Self-predictive representations (ICLR’21)

Schwarzer et al., 2021

Ignores statistical uncertainty in results.

State Abstraction (NeurIPS’21)



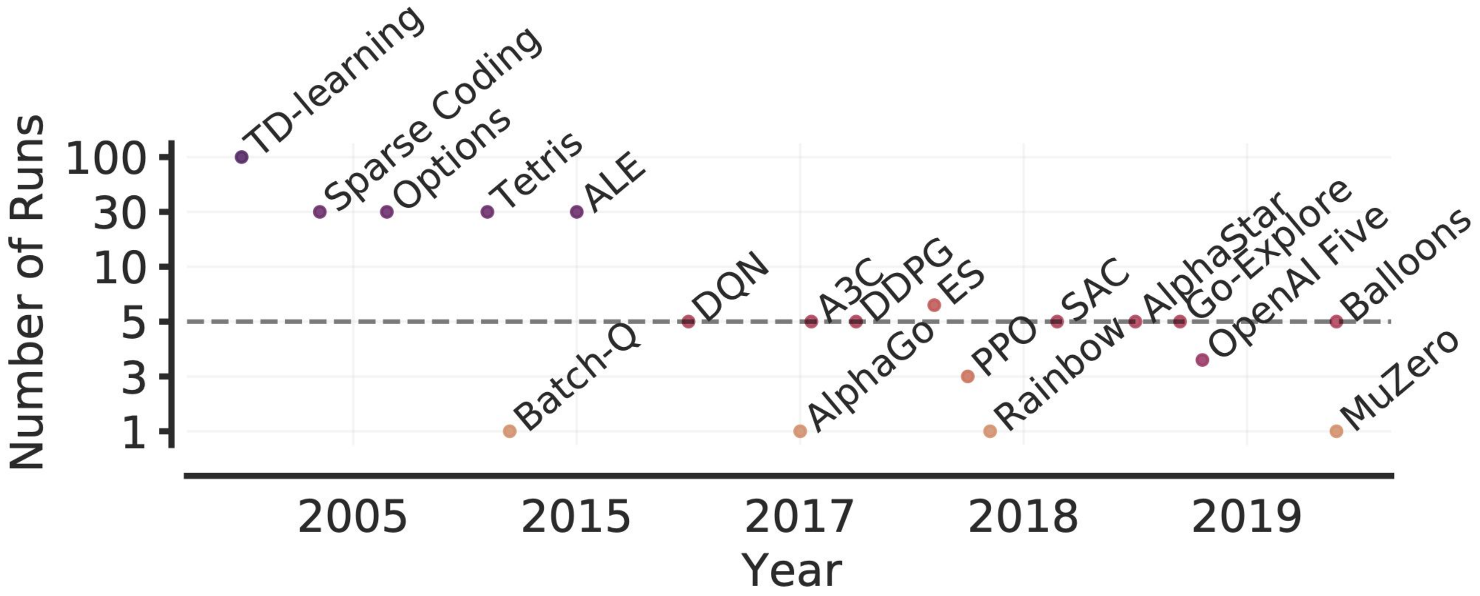
Castro et al., 2021

Of[illegible] RL (ICML’20)

Offline agent	Median	> DQN
DQN (Nature)	83.4%	17
DQN ([illegible]	[illegible]	41
Ensem[illegible]	[illegible]	39
Averag[illegible]	[illegible]	43
QR-DQN	118.9%	45
REM	**123.8%**	**49**

Agarwal et al., 2020

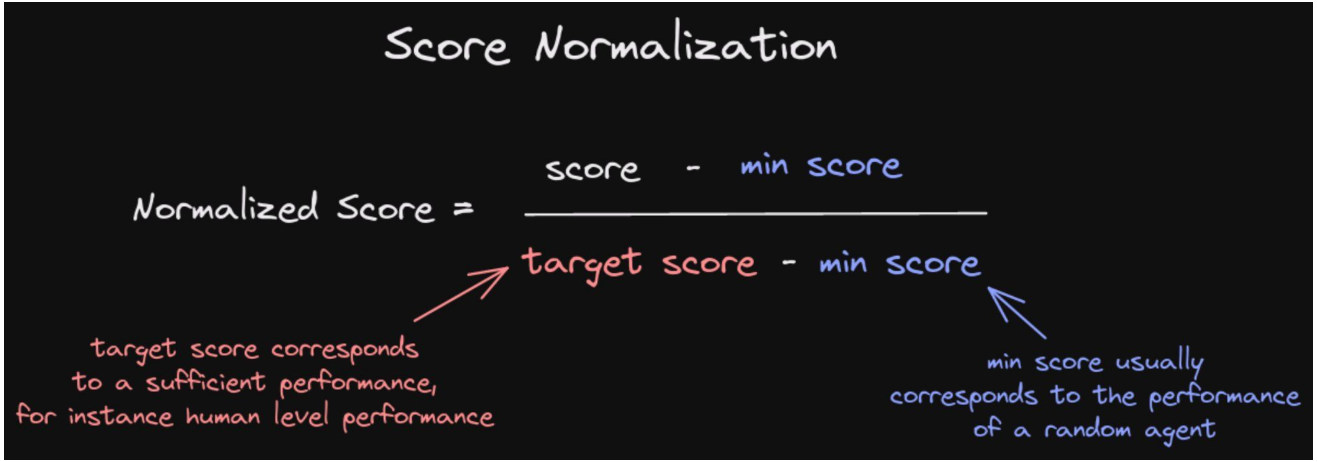
Statistical uncertainty exacerbated by small number of runs in Deep RL



So, what could go wrong with ignoring statistical uncertainty?

Case Study: Atari 100k benchmark

- Evaluate performance after 100k training steps (~ 2-3 hrs of gameplay)
  - Aggregate results on 26 Atari games



Source: [A visual introduction to RLiable](#) by Antonin Raffin

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., ... & Michalewski, H. (2019). Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.

Case Study: Atari 100k benchmark

- Evaluate performance when trained for 100k interactions (~ 2-3 hrs of gameplay)
  - Aggregate results on 26 Atari games
- Comparison using **Median Human Normalized** Scores
  - Typically 3-5 runs per game
  - $$\text{Median}(\frac{1}{N}\sum_{n=1}^{N} \underbrace{x_{1,n}}_{\text{Score on game 1 on run `n`}}, \frac{1}{N}\sum_{n=1}^{N} x_{2,n}, \cdots, \frac{1}{N}\sum_{n=1}^{N} x_{M,n},)$$

Kaiser, L., Babaeizadeh, M., Milos, P., Osinski, B., Campbell, R. H., Czechowski, K., ... & Michalewski, H. (2019). Model-based reinforcement learning for atari. *arXiv preprint arXiv:1903.00374*.

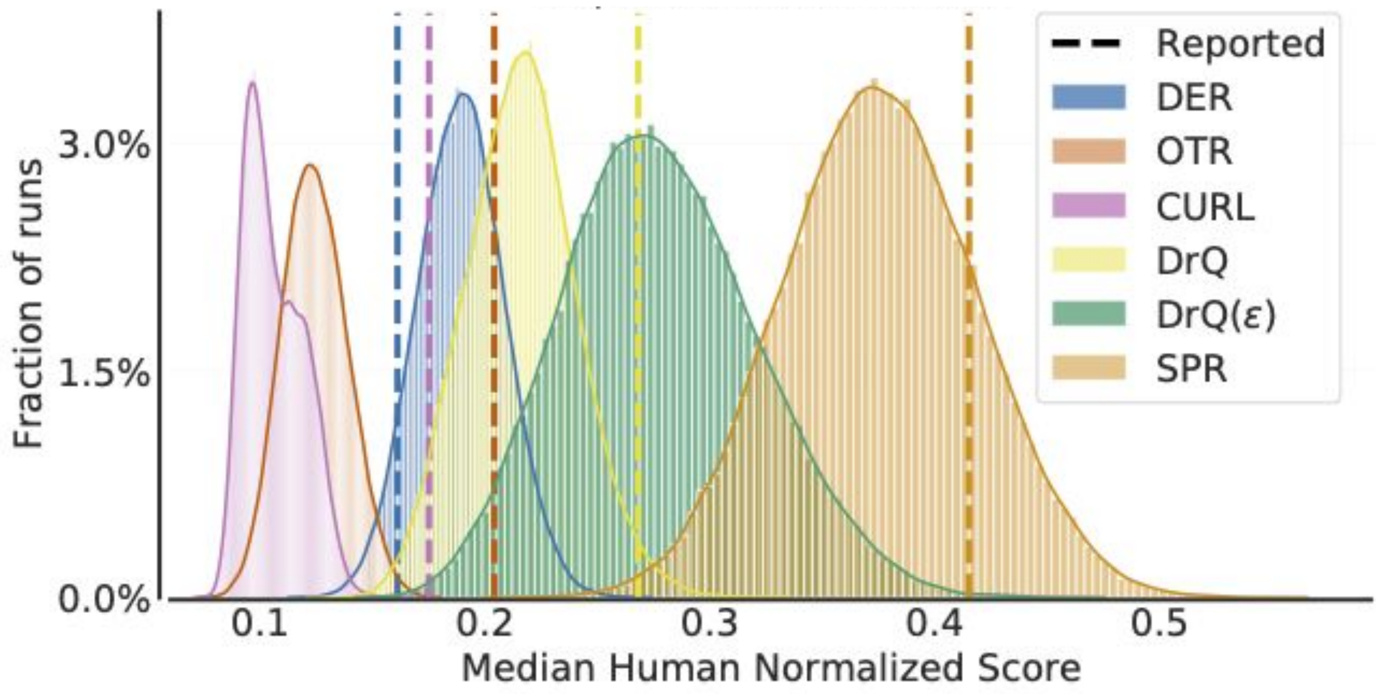
Case Study: Experimental Setup

- Evaluate **100 independent runs** for 5 algorithms:
  - DER, OTR, DrQ, CURL, and SPR

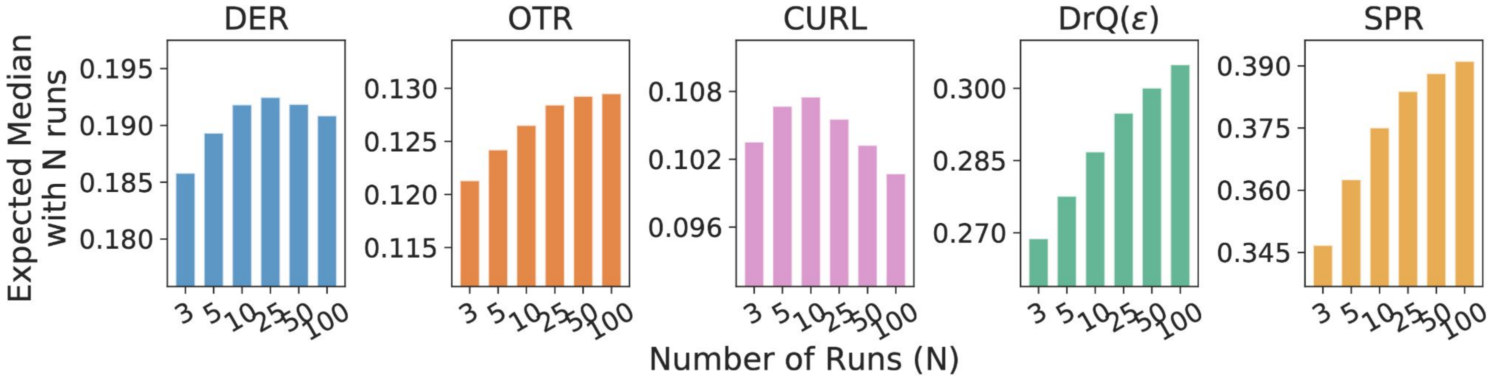
- We have **26 games × 100 scores/game** per algorithm.
  - Subsample scores with replacement to 3–100 runs.

1. van Hasselt, Hado, Matteo Hessel, and John Aslanides. "When to use parametric models in reinforcement learning?." ***NeurIPS* (2019).**
2. Kielak, Kacper. "Do recent advancements in model-based deep reinforcement learning really improve data efficiency?." *arXiv preprint arXiv:2003.10181* (2020).
3. Kostrikov, Ilya, Denis Yarats, and Rob Fergus. "Image augmentation is all you need: Regularizing deep reinforcement learning from pixels." ***ICLR* (2021).**
4. Srinivas, Aravind, Michael Laskin, and Pieter Abbeel. "CURL: Contrastive unsupervised representations for reinforcement learning." ***ICML* (2020).**
5. Schwarzer, Max, Ankesh Anand, Rishab Goel, R. Devon Hjelm, Aaron Courville, and Philip Bachman. "Data-efficient reinforcement learning with momentum predictive representations." ***ICLR* (2021).**

What if I report performance using a different set of runs?

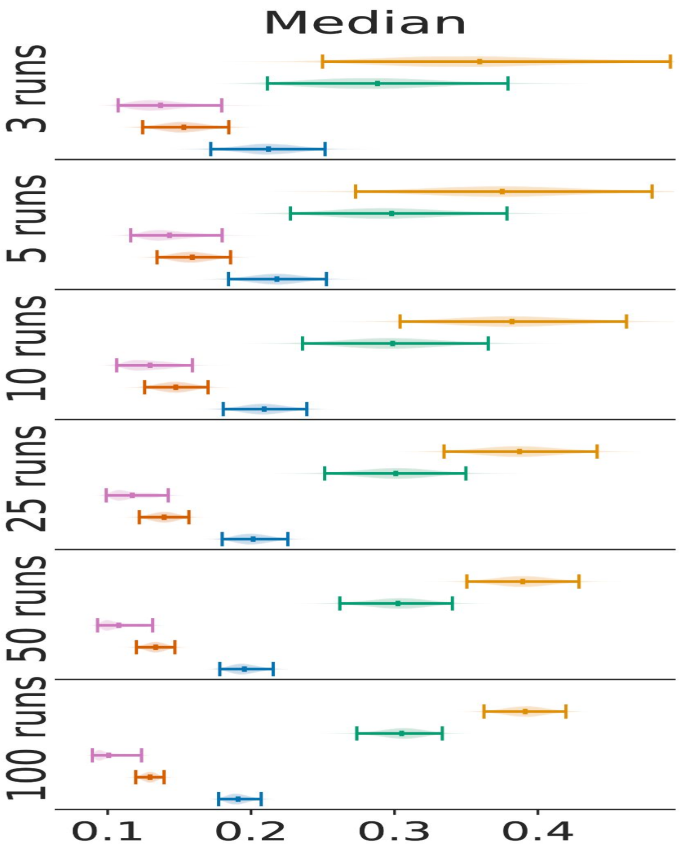


Median scores are substantially biased!



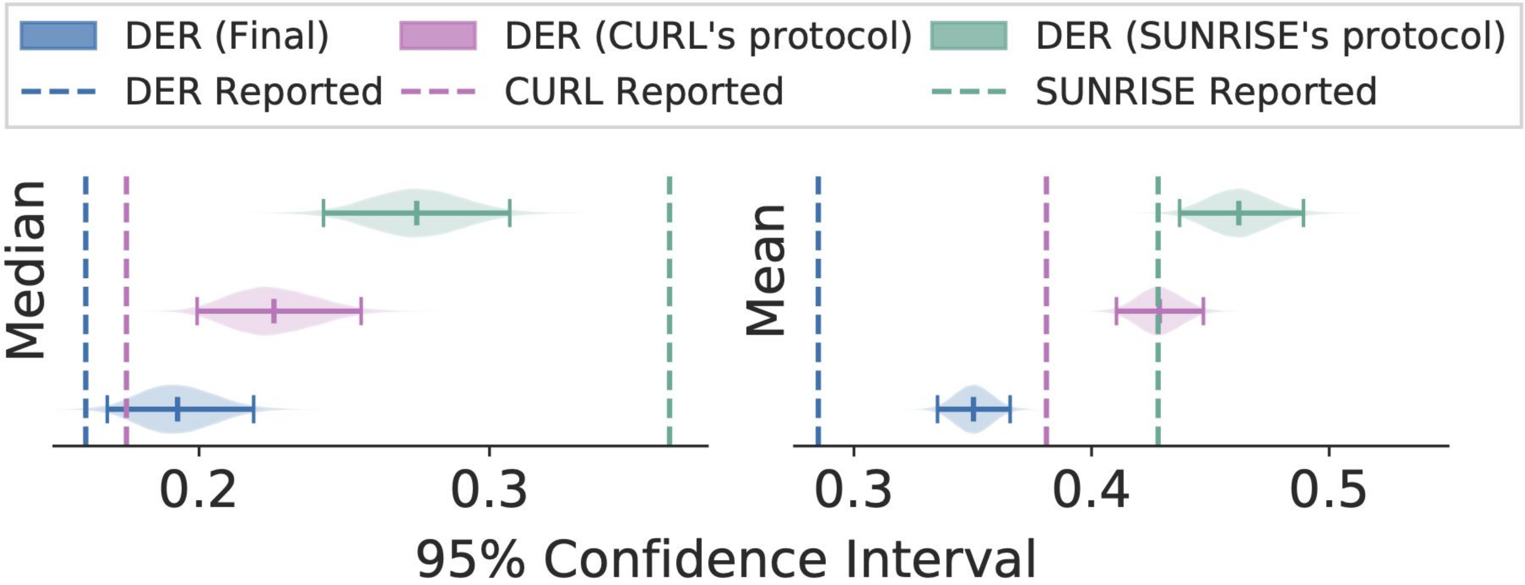
$$\mathrm{Median}\big(\mathbb{E}[X_1], \cdots, \mathbb{E}[X_M]\big) \neq \mathbb{E}\Big[\mathrm{Median}\big(\frac{1}{N}\sum_{n=1}^{N} X_{1,n}, \cdots, \frac{1}{N}\sum_{n=1}^{N} X_{M,n}\big)\Big]$$

How many runs for negligible uncertainty?



Even 30-50 runs may not suffice for certain comparisons.

Changes in evaluation protocols invalidates comparisons to prior work.



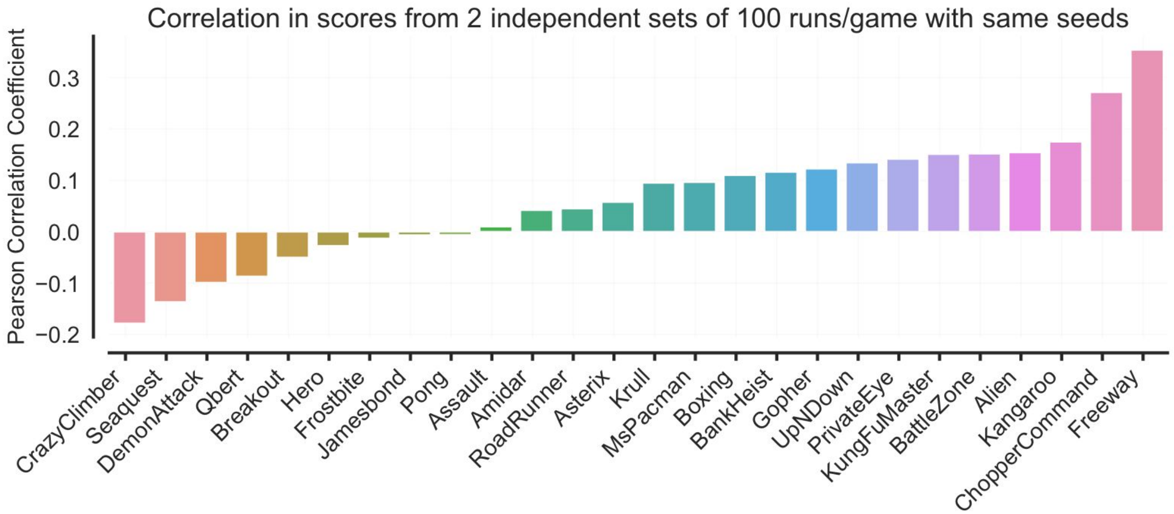
Also *see*: Mauro Birattari and Marco Dorigo. How to assess and report the performance of a stochastic algorithm on a benchmark problem: mean or best result on a number of runs? Optimization letters, 2007.

How to reliably evaluate performance?

How to reliably evaluate performance?

~~**Just Fix Random Seeds?**~~ Not a solution.

- Why prefer one set of seeds over another?
- Often can't fix randomness in practice (different hardware, non-determinism in GPUs)



How to reliably evaluate performance?

~~**Evaluate More Runs?**~~ Not feasible.

- 5 runs on 50 Atari games for 200M frames takes 1000+ GPU days.
- More complex RL benchmarks -- quite expensive to evaluate even a few runs.

How to reliably evaluate performance *with a handful of runs*?

~~Is statistical significance testing the solution~~? Not really.

- Dichotomous (significant vs not significant)
- Widely misinterpreted.
- Often hide effect sizes (such as size of improvement over baseline).

Fun fact: Main statistics journal in USA bans thresholding p-values!

[1] Amrhein, Valentin, Sander Greenland, and Blake McShane. "**Scientists rise up against statistical significance**." Nature (2019) .
[2] Wasserstein, Ronald L., Allen L. Schirm, and Nicole A. Lazar. "**Moving to a world beyond "p< 0.05"**." The American Statistician (2019).

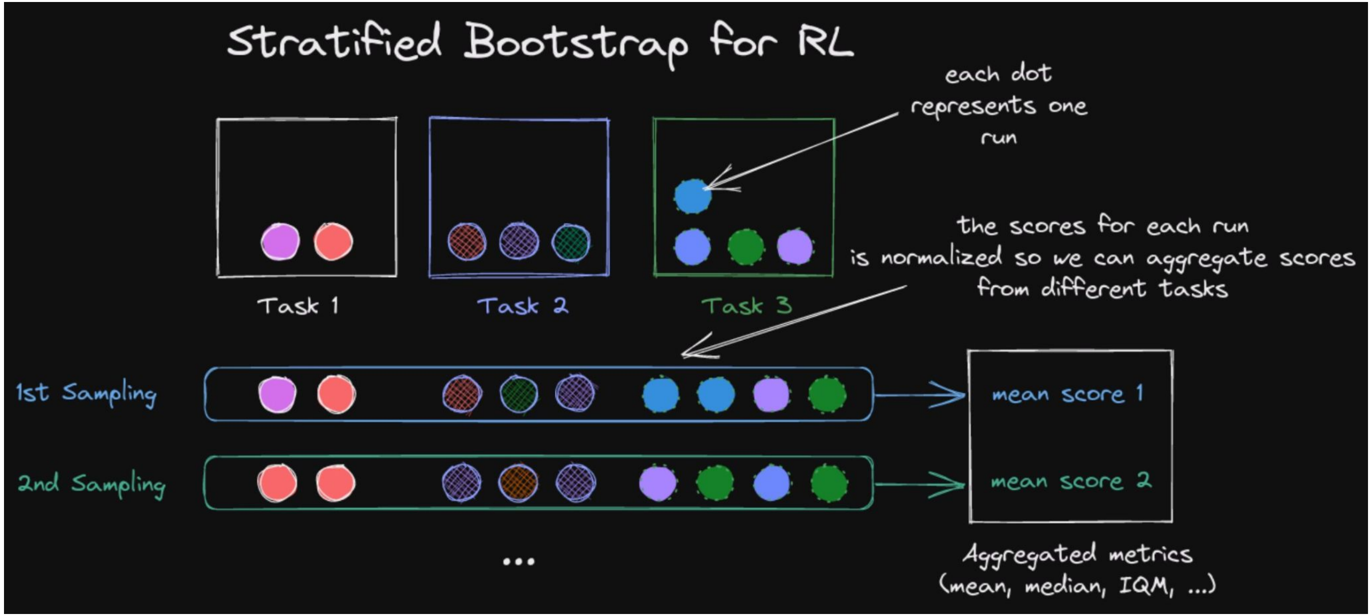
How to reliably evaluate performance *with a handful of runs*?

Desiderata	Current evaluation approach	Our recommendation
Uncertainty in aggregate performance	Point estimates	**Interval estimates**
Variability in performance across tasks and runs	Tables with task mean scores	**Performance Profiles**
Aggregate metrics for overall performance	Mean / Median	**Interquartile Mean (IQM), Prob. of Improvement**

Interval Estimates: Stratified Bootstrap Confidence Intervals

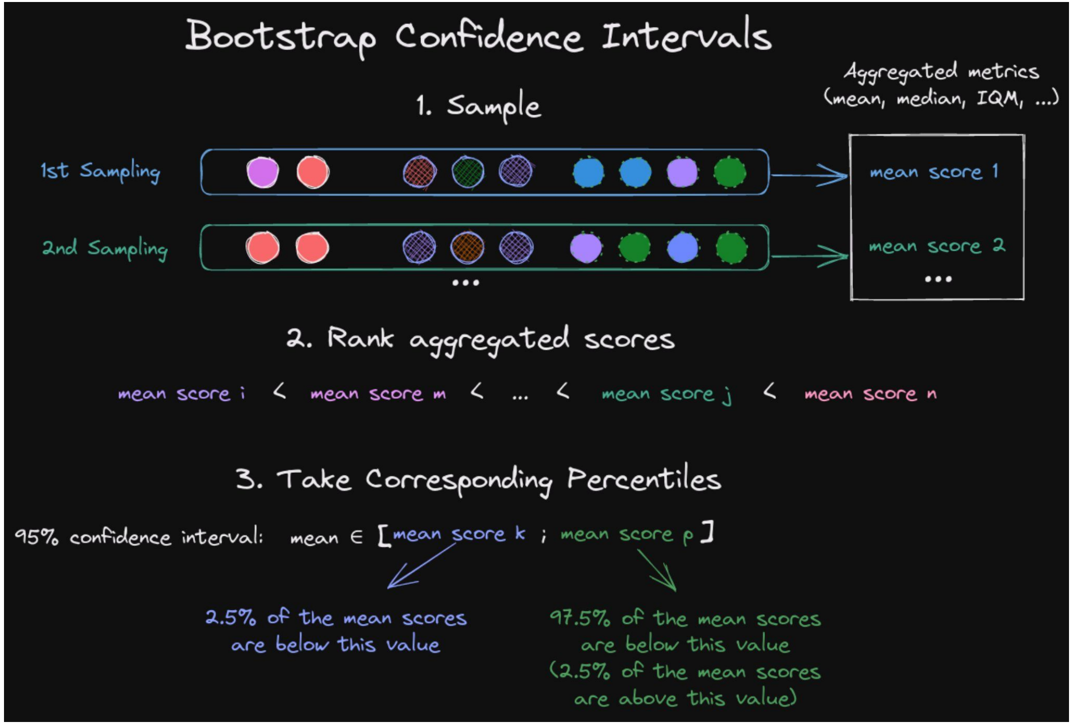
- “If we repeat the experiment with different runs, what aggregate score are we expected to get?”

Interval Estimates:
Stratified Bootstrap Confidence Intervals



Source: [A visual introduction to RLiable](#) by Antonin Raffin

Stratified Bootstrap Confidence Intervals: How does it work?



Source: [A visual introduction to RLiable](#) by Antonin Raffin

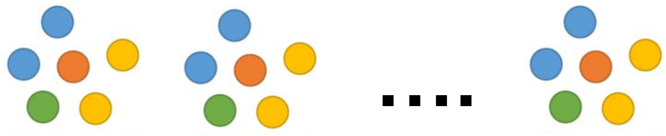
Interval Estimates:
Stratified Bootstrap Confidence Intervals

Single task with N runs

Task 1

- Only N random samples
- Bootstrapping CIs don't make sense with N ≤ 5!

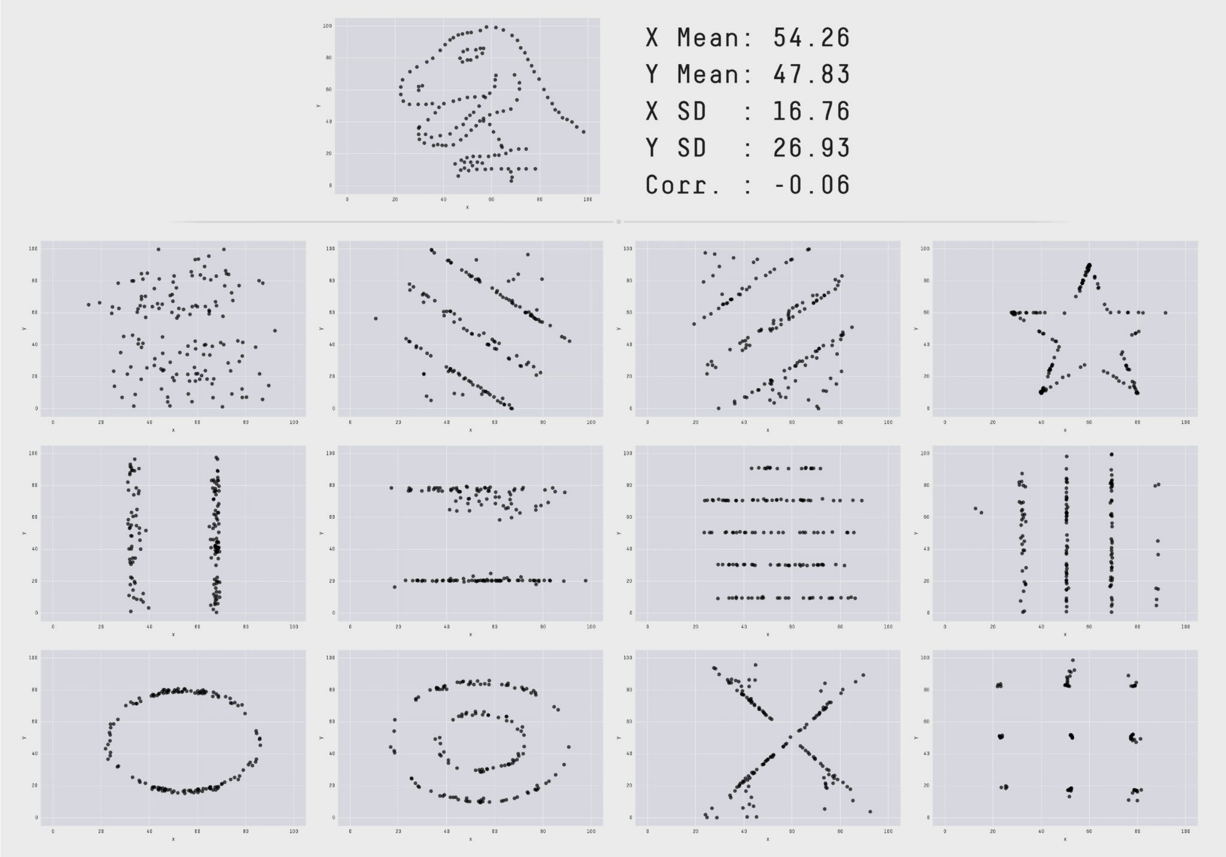
M tasks with N runs



Task 1 Task 2 Task M

- N*M random samples
- Bootstrapping results in reasonably accurate CIs with N ≥ 5!

Aggregate metrics hide task variability!



Source: Same Stats, Different Graphs. https://www.autodesk.com/research/publications/same-stats-different-graphs.

Performance Variability: Tables with per-task scores?

Google Research

Game	Random	Human	SimPLe	DER	OTRainbow	CURL	DrQ	SPR (no Aug)	SPR
Alien	227.8	7127.7	616.9	739.9	824.7	558.2	771.2	**847.2**	801.5
Amidar	5.8	1719.5	88.0	**188.6**	82.8	142.1	102.8	142.7	176.3
Assault	222.4	742.0	527.2	431.2	351.9	600.6	452.4	**665.0**	571.0
Asterix	210.0	8503.3	**1128.3**	470.8	628.5	734.5	603.5	820.2	977.8
Bank Heist	14.2	753.1	34.2	51.0	182.1	131.6	168.9	**425.6**	380.9
BattleZone	2360.0	37187.5	5184.4	10124.6	4060.6	14870.0	12954.0	10738.0	**16651.0**
Boxing	0.1	12.1	9.1	0.2	2.5	1.2	6.0	12.7	**35.8**
Breakout	1.7	30.5	16.4	1.9	9.8	4.9	16.1	12.9	**17.1**
ChopperCommand	811.0	7387.8	**1246.9**	861.8	1033.3	1058.5	780.3	667.3	974.8
Crazy Climber	10780.5	35829.4	**62583.6**	16185.3	21327.8	12146.5	20516.5	43391.0	42923.6
Demon Attack	152.1	1971.0	208.1	508.0	711.8	817.6	**1113.4**	370.1	545.2
Freeway	0.0	29.6	20.3	**27.9**	25.0	26.7	9.8	16.1	24.4
Frostbite	65.2	4334.7	254.7	866.8	231.6	1181.3	331.1	1657.4	**1821.5**
Gopher	257.6	2412.5	771.0	349.5	**778.0**	669.3	636.3	774.5	715.2
Hero	1027.0	30826.4	2656.6	6857.0	6458.8	6279.3	3736.3	5707.4	**7019.2**
Jamesbond	29.0	302.8	125.3	301.6	112.3	**471.0**	236.0	367.2	365.4
Kangaroo	52.0	3035.0	323.1	779.3	605.4	872.5	940.6	1359.5	**3276.4**
Krull	1598.0	2665.5	**4539.9**	2851.5	3277.9	4229.6	4018.1	3123.1	3688.9
Kung Fu Master	258.5	22736.3	**17257.2**	14346.1	5722.2	14307.8	9111.0	15469.7	13192.7
Ms Pacman	307.3	6951.6	**1480.0**	1204.1	941.9	1465.5	960.5	1247.7	1313.2
Pong	-20.7	14.6	**12.8**	-19.3	1.3	-16.5	-8.5	-16.0	-5.9
Private Eye	24.9	69571.3	58.3	97.8	100.0	**218.4**	-13.6	52.6	124.0
Qbert	163.9	13455.0	**1288.8**	1152.9	509.3	1042.4	854.4	606.6	669.1
Road Runner	11.5	7845.0	5640.6	9600.0	2696.7	5661.0	8895.1	10511.0	**14220.5**
Seaquest	68.4	42054.7	**683.3**	354.1	286.9	384.5	301.2	580.8	583.1
Up N Down	533.4	11693.2	3350.3	2877.4	2847.6	2955.2	3180.8	6604.6	**28138.5**

- Overwhelming beyond a few tasks
- Standard deviations frequently omitted
- Mean scores present incomplete picture for non-gaussian distributions!

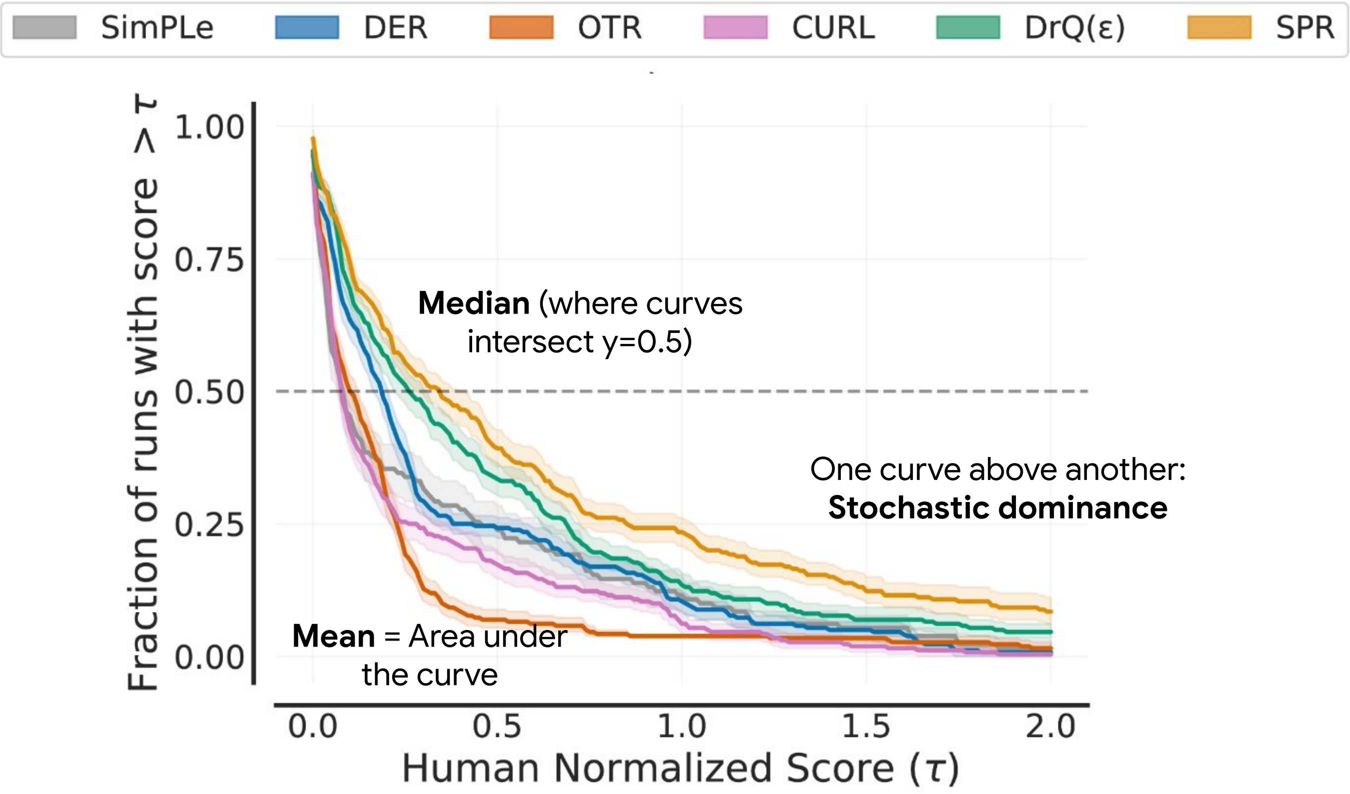
A better approach: Performance profiles with CIs

$$p(\tau) = \frac{1}{NM} \sum_{m=1}^{M} \sum_{n=1}^{N} \mathbf{1}[X_{n,m} > \tau]$$

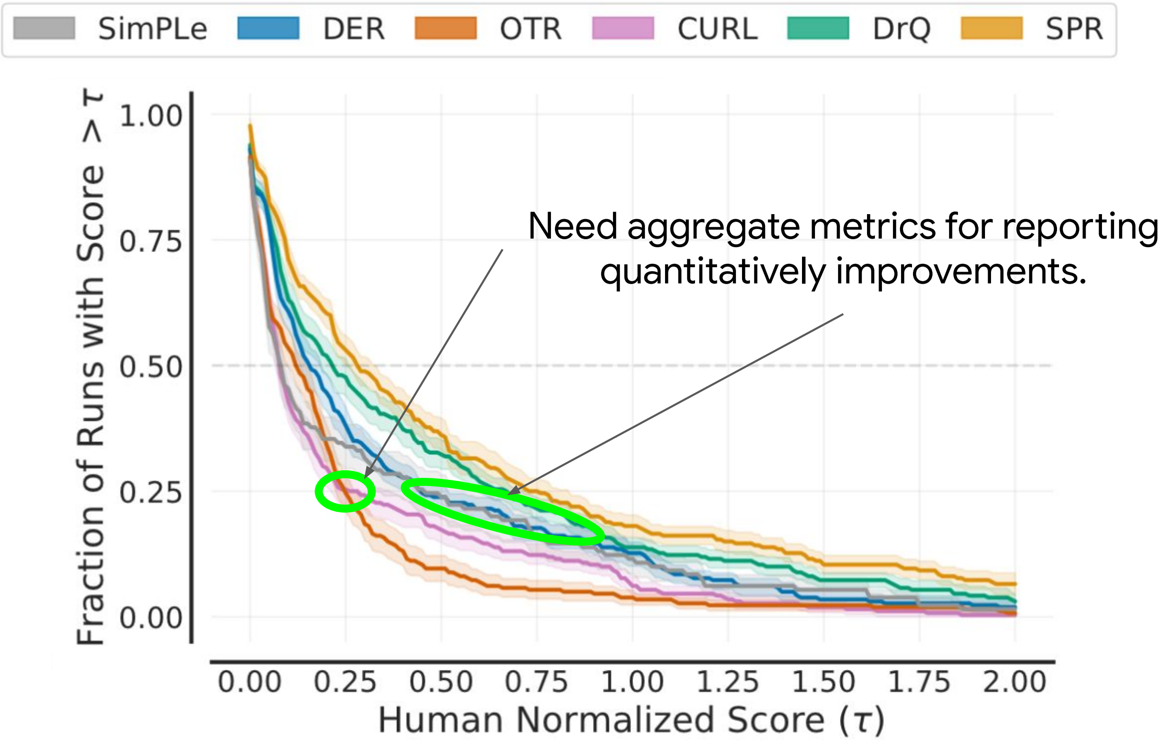
- Typically used for comparing solve times of different optimization methods.
- Robust to outlier runs/tasks.
- Robust to small changes in performance across all tasks.

Dolan, E. D., & Moré, J. J. (2002). Benchmarking optimization software with performance profiles. *Mathematical programming*.

Performance Profiles for a bird's-eye view!



What if one algorithm doesn't dominate another?



Existing Metrics are Deficient!

Google Research

- **Median**: High variability and not robust -- score of 0 on half of the tasks does not change it.

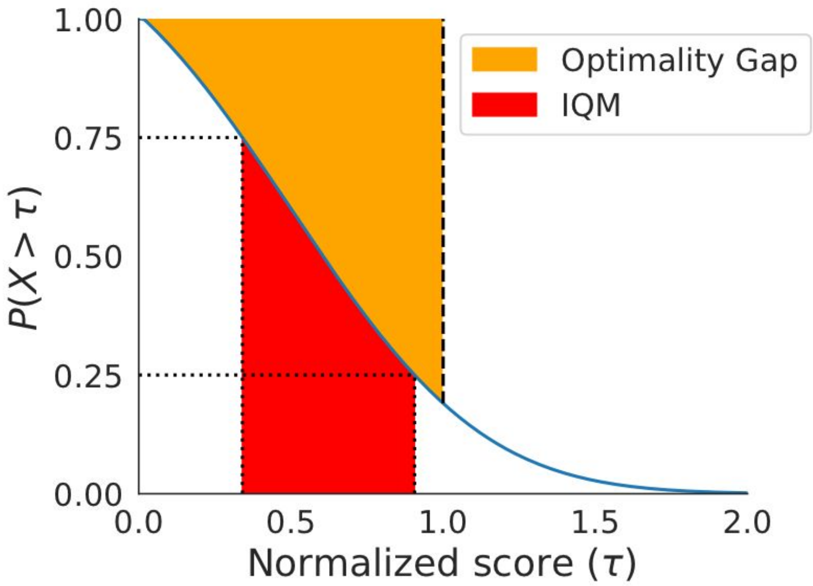
- **Mean**: Easily dominated by a few outlier tasks.

Need better aggregate metrics that are robust, not dominated by outliers and have small uncertainty.

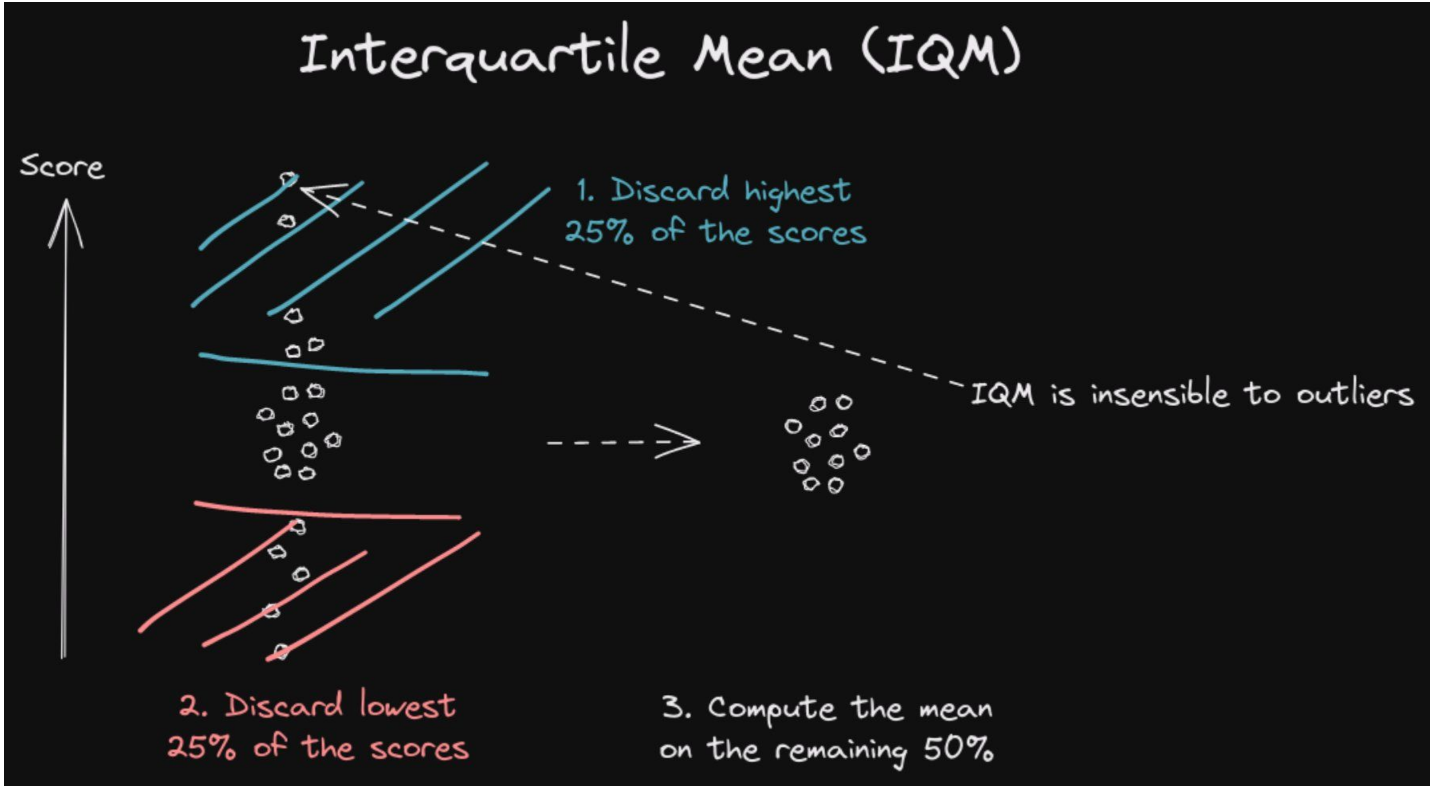
Robust and Efficient Aggregate Metrics

- Median → **Interquartile Mean (IQM)**
  - Averages middle 50% scores across all runs and tasks
  - Best of both worlds: Median, Mean: 50%, 0% trimmed mean

- Mean → **Optimality Gap**
  - How far an algorithm is from optimal performance

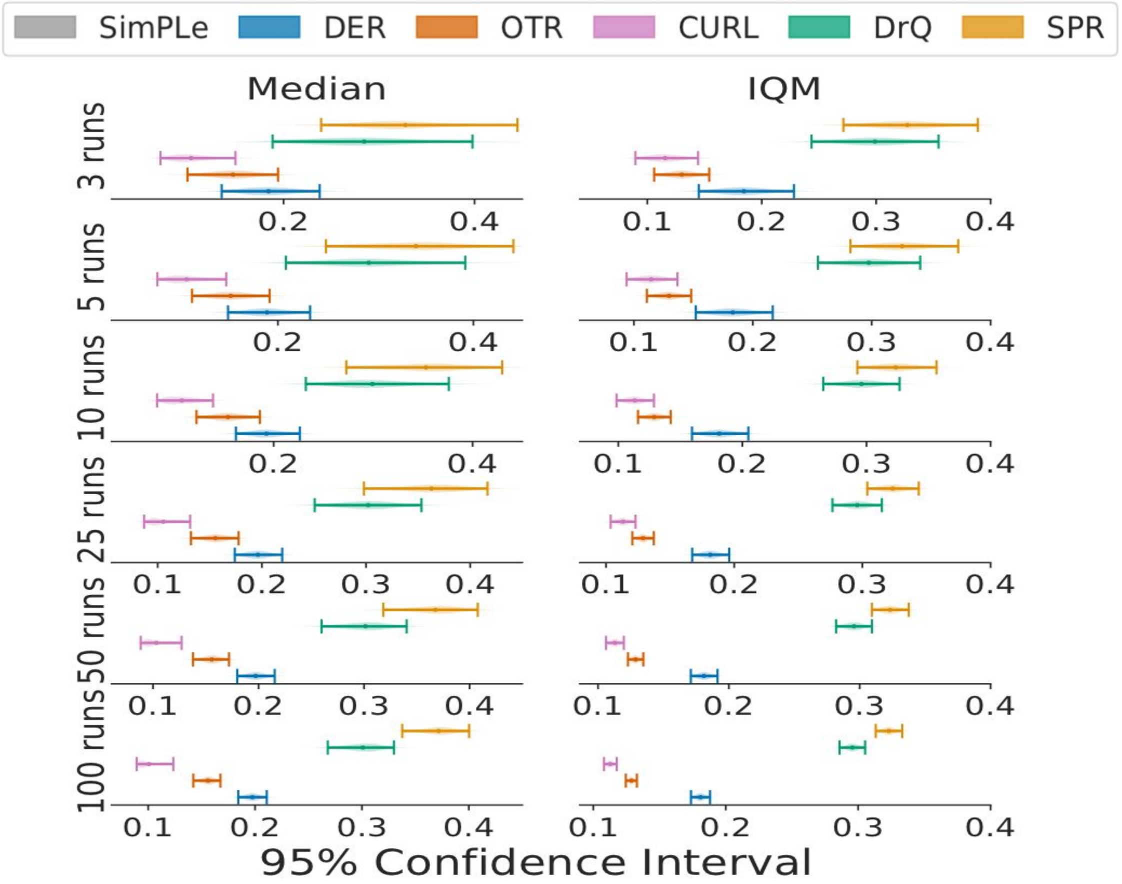


Visual introduction to IQM



Source: *A visual introduction to RLiable by Antonin Raffin*

IQM leads to smaller confidence intervals



Google Research

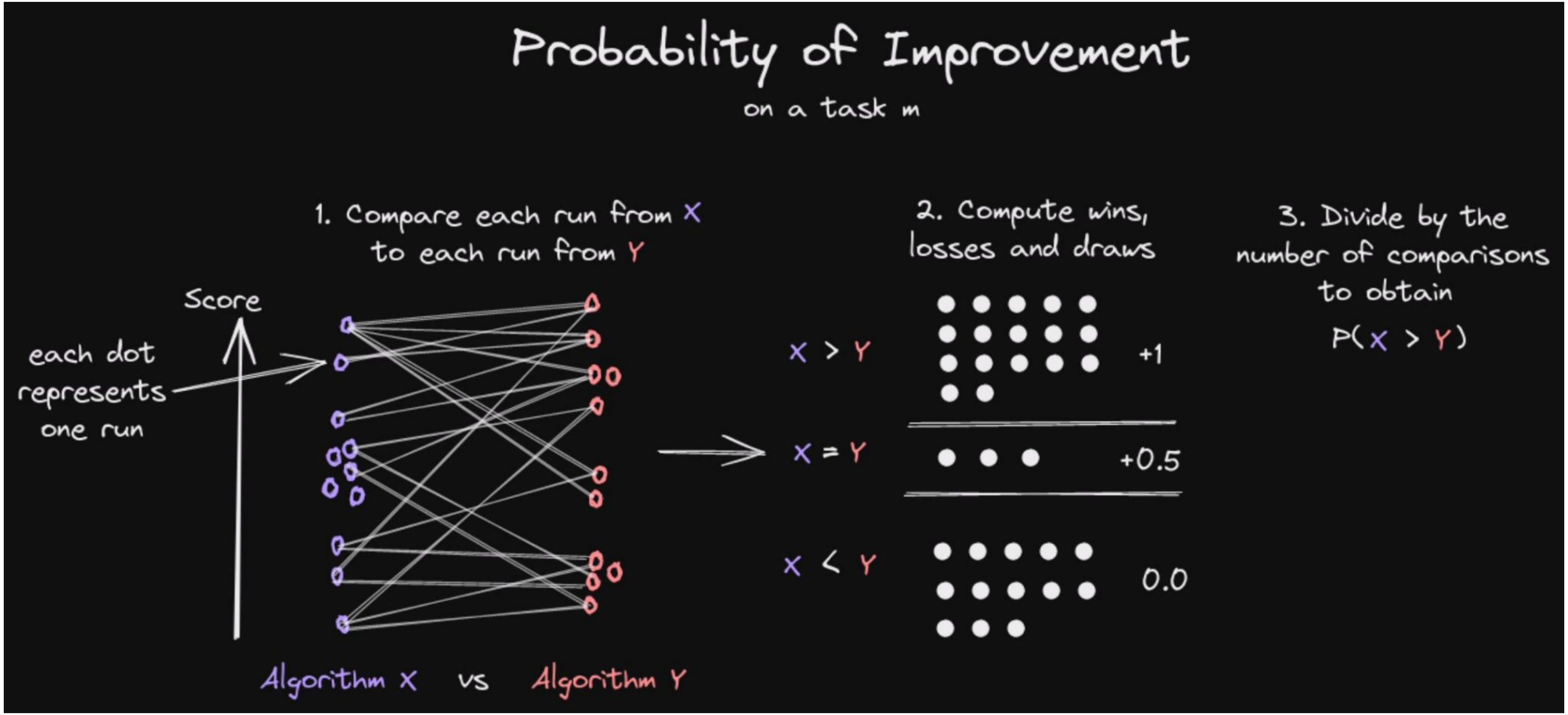
Am I better than the baseline?

We can compute probability of improvement of algorithm X over Y.

$$P(X > Y) = \frac{1}{M} \sum_{m=1}^{M} P(X_m > Y_m)$$

Performance on task X.

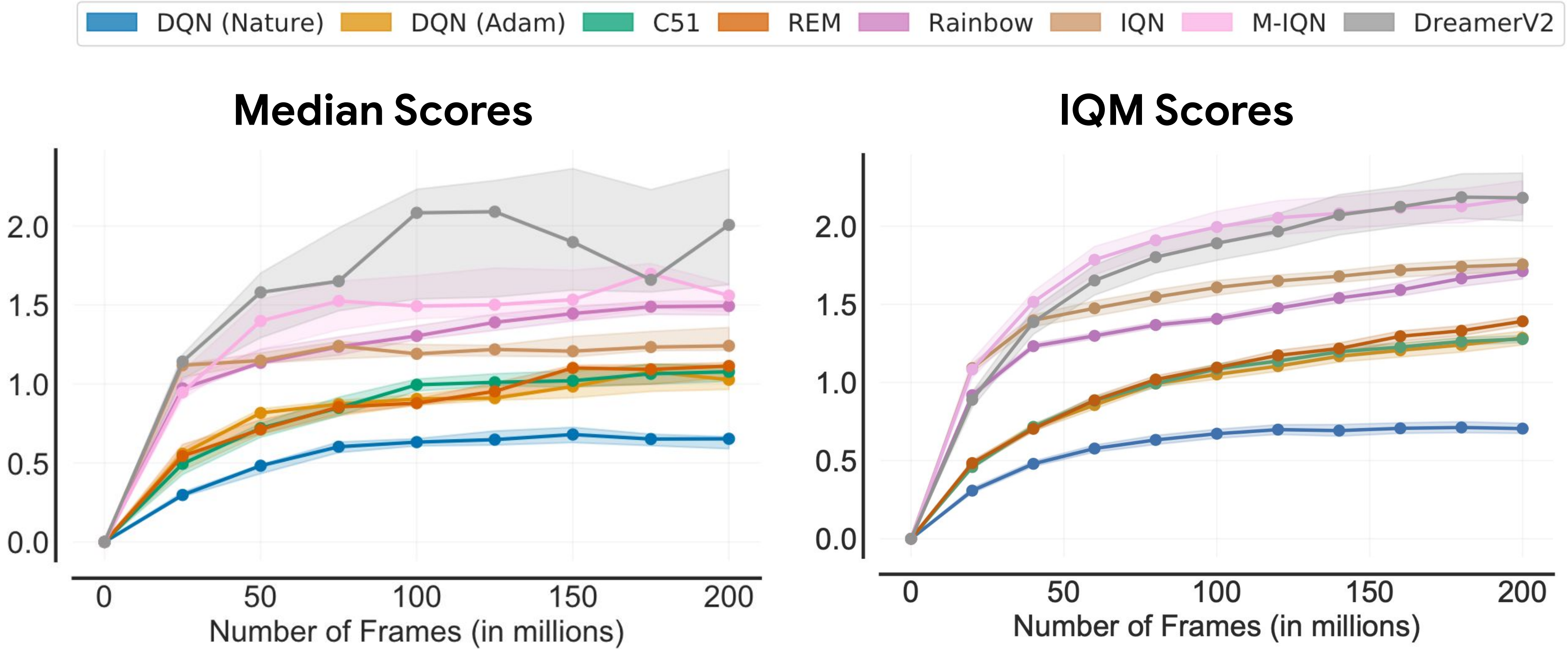
Probability of Improvement



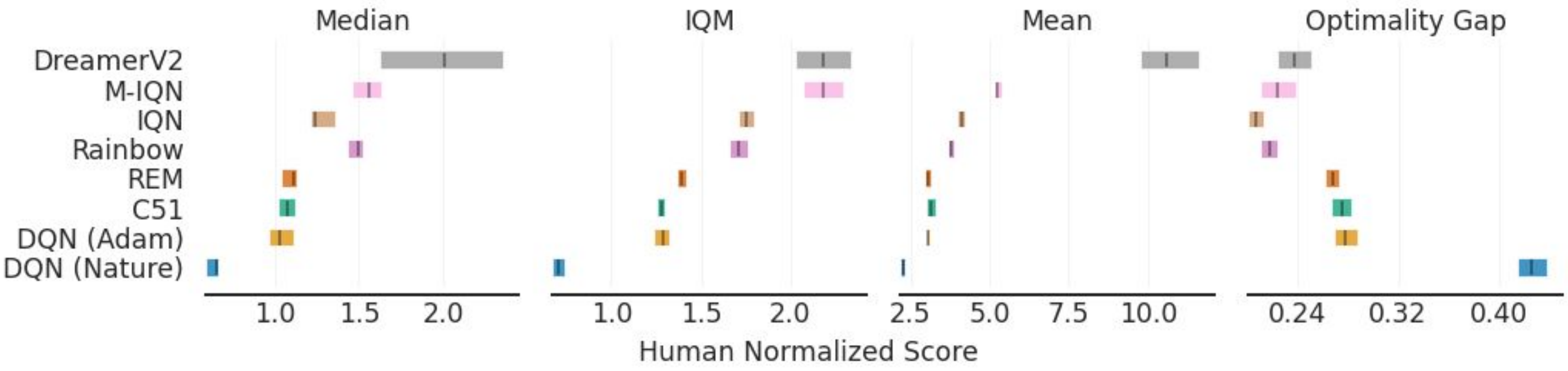
Source: A visual introduction to RLiable by Antonin Raffin

Re-evaluating Evaluation

Re-evaluating algorithms on ALE



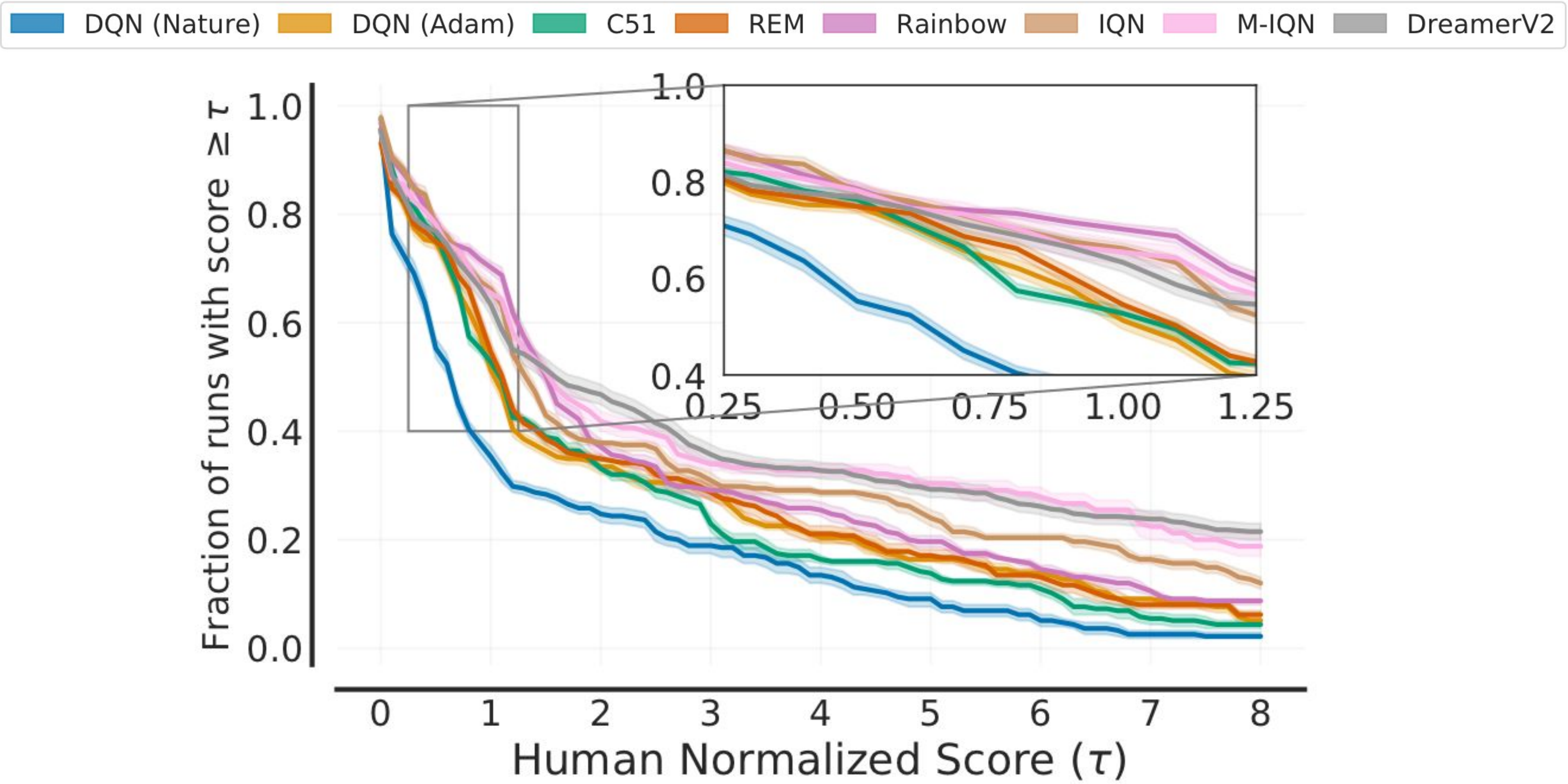
ALE: Interval estimates



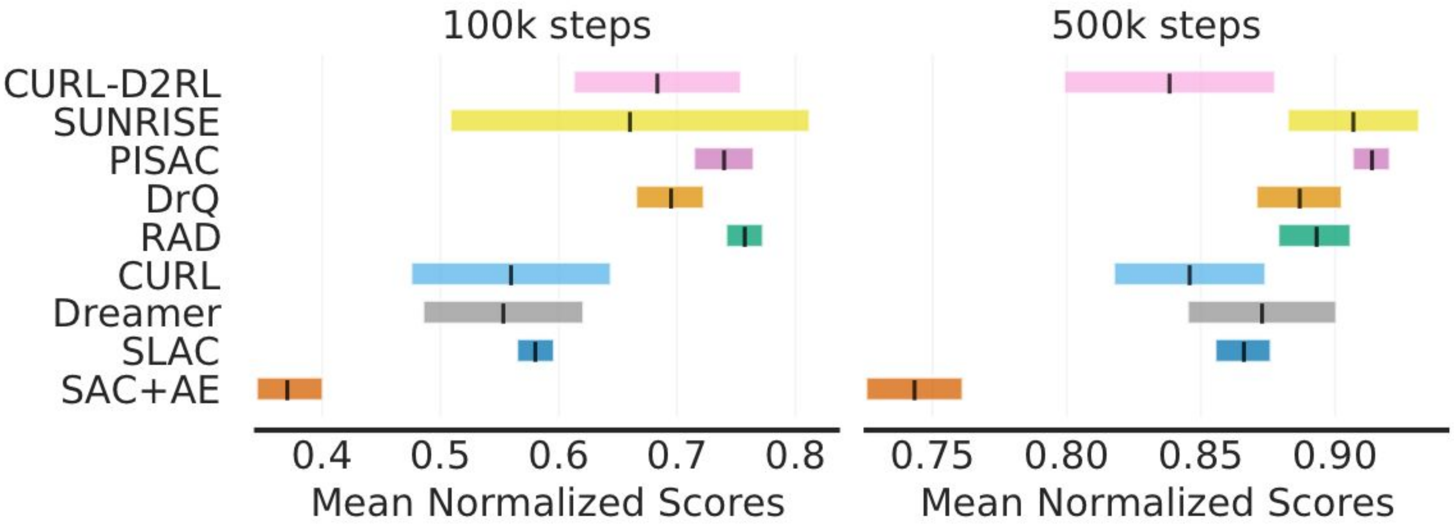
Performance Ranking changes depending on the metric!

ALE: Performance Profiles

Google Research

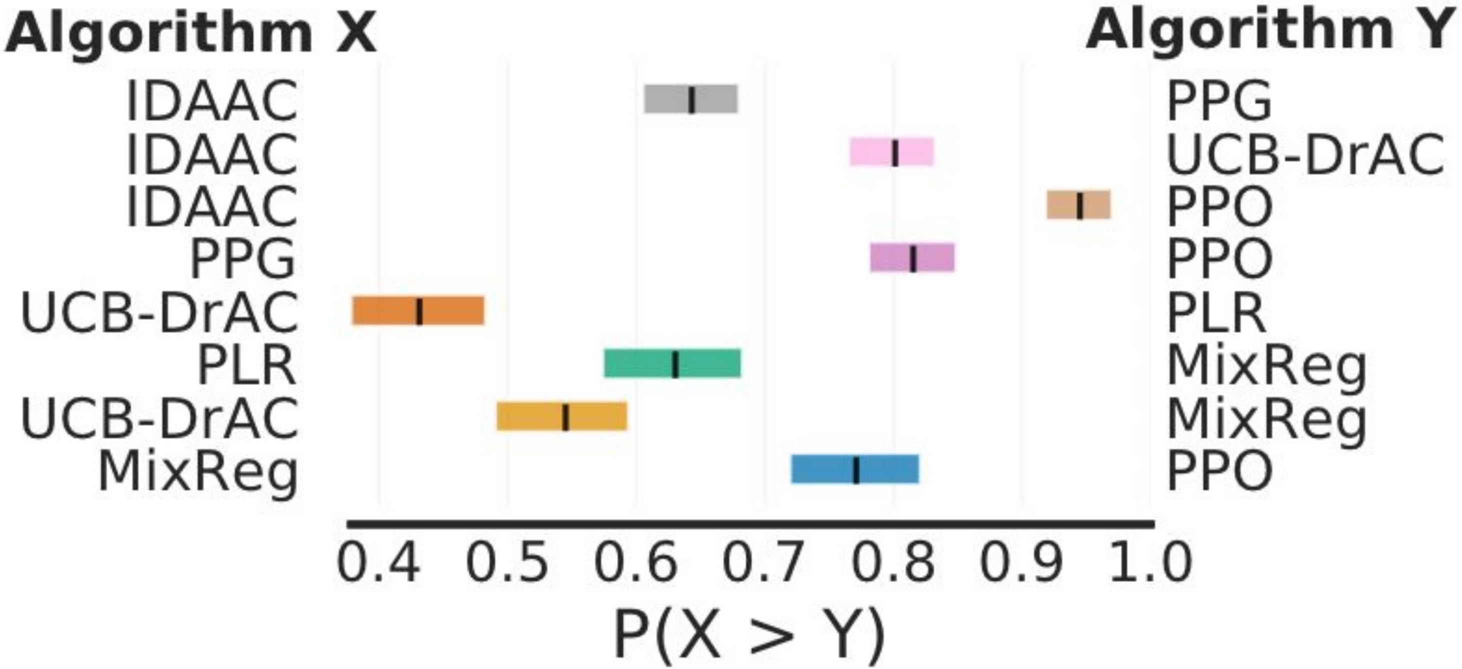


Re-evaluating algorithms on DM Control



1 - Optimality Gap

Procgen: Average Probability of Improvement



Takeaways

Google Research

- **Use interval estimates** as opposed to point estimates.
- More is more: **Performance profiles** for qualitative summarization.
- Use better aggregate performance measures such as **interquartile mean** (IQM) and prob. of improvement.
- Provide individual runs for better statistical comparisons.

See bit.ly/statistical_precipice_colab for jumpstart. Thank you!

Just as a rock-climber can skirt the edge of the steepest precipices, it seems likely that ongoing progress in RL will require greater experimental discipline.

See agarwl.github.io/rliable.