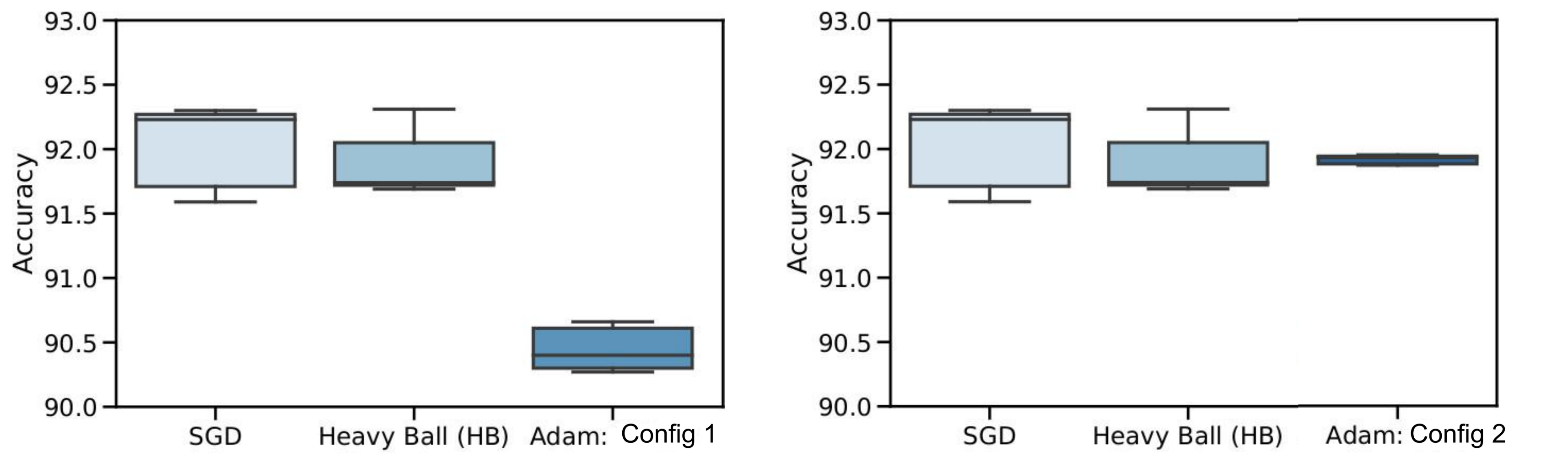
Hyperparameter Optimization Is Deceiving Us, and How to Stop It

A. Feder Cooper[1], Yucheng Lu[1], Jessica Zosa Forde[2], and Chris De Sa[1]
Cornell University[1], Brown University[2]

We want to prevent our conclusions about algorithm performance from depending on the underlying configuration of the **hyperparameter optimization (HPO)** that we perform

We do not know the ground truth



non-adaptive optimizers outperform adaptive optimizers	**non-adaptive optimizers do not outperform adaptive optimizers**
p AND	$\neg p$ → **FALSE**

It is **fine** to **accept either** to form conclusions

It is **fine** to **accept neither** to form no conclusions

It is **not fine** to **accept both** to form inconsistent conclusions

We do not want it to be ***possible*** to form inconsistent conclusions because we want to derive **reliable knowledge** about algorithm performance

This is challenging because

- **the process of picking HPO configurations to test is vague**
- **whether we believe our conclusions or not is also vague**

We remove the vagueness from the problem by pinning down

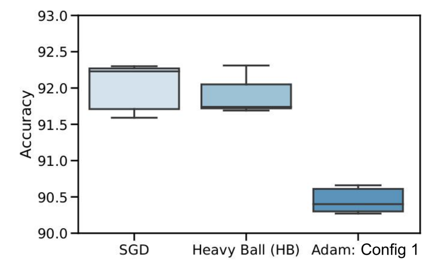
- a concrete formalization for the
- **possible outcomes of running HPO**
- A concrete formalization for our **belief in conclusions**

To do so, we define HPO to return a **log**, which records all the choices and measurements made during an HPO run (enabling reproducibility)

We then formalize the process of drawing conclusions from empirical studies using HPO:

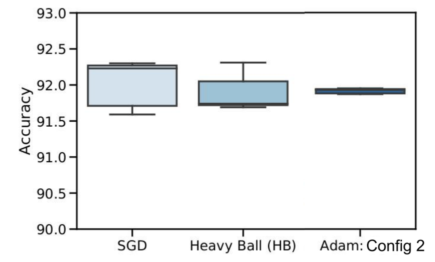
Epistemic Hyperparameter Optimization (EHPO) takes a set of HPO procedures and a **function F**, which maps a set of HPO **logs to conclusions** about algorithm performance

Log set L1 (|L1| = 3)



F(L1) → "Non-adaptive optimizers outperform adaptive ones" = p

Log set L2 (|L2| = 3)



F(L2) → "Non-adaptive optimizers do not outperform adaptive ones" = $\neg p$

We imagine a demon trying to deceive us about algorithm performance via EHPO

The demon maintains a set of HPO logs that it can modify, and presents us with a final log set, from which we can draw conclusions

The demon ***could*** produce L1 or ***could*** produce L2, and then could discard L2

Modal logic is the standard way to formalize ***could*** by extending propositional logic to allow us to reason about possibility by introducing an additional operator

$\Diamond\phi$ reads, "It is possible that ϕ"

We combine **two modal operators** so that we can capture the idea that **it is not possible** to adopt **inconsistent beliefs** about experimental outcomes

$\Diamond_t p$ means a demon could adopt strategy guaranteed to cause **desired outcome *p***, taking time at most ***t*** in expectation; a set of EHPO output logs models that it is possible ***p*** in time ***t***

$\mathcal{B}p$ means we believe/conclude ***p***; a set of EHPO output logs models our belief in ***p***

With both of these operators, we can formalize the problem of **hyperparameter deception**

***t*-non-deceptive axiom**

$$\neg(\Diamond_t \mathcal{B}p \wedge \Diamond_t \mathcal{B}\neg p)$$

If it is **possible** for the demon can get us to **believe *p*** in time ***t***, then it is **not possible** for the demon to get us to **believe *not p*** in time ***t***

We use this formalization to **prove non-trivial theorems** about whether a HPO procedure is **defended against deception**

We can do this by proving that an EHPO satisfies our ***t*-non-deceptive axiom**

Intuitively, our defense works as follows: Given some **naive reasoner**, we construct a **defended reasoner** that is **always more skeptical** than the naive reasoner

If the naive reasoner is ***t*-non-deceptive**, then any more skeptical reasoner is also ***t*-non-deceptive**

The important takeaway is that it is ***always possible*** to construct a *t*-defended EHPO

We describe a **defended variation of random search**, which is defended against deception for ***t***, but it is able to draw conclusions using up compute budgets that are **O(√t)**

	p	$\neg p$	$1-\delta$	Conclude
SGD vs. Adam	0.213	0.788	0.75	$\neg p$
			0.8	Nothing
			0.9	Nothing
HB vs. Adam	0.168	0.832	0.75	$\neg p$
			0.8	$\neg p$
			0.9	Nothing

It makes conclusions using **much fewer resources** than the total compute budget for which it is defended against deception