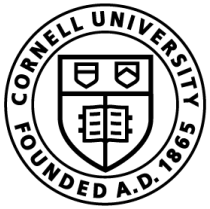


Cornell Bowers C·IS
College of Computing
and Information Science

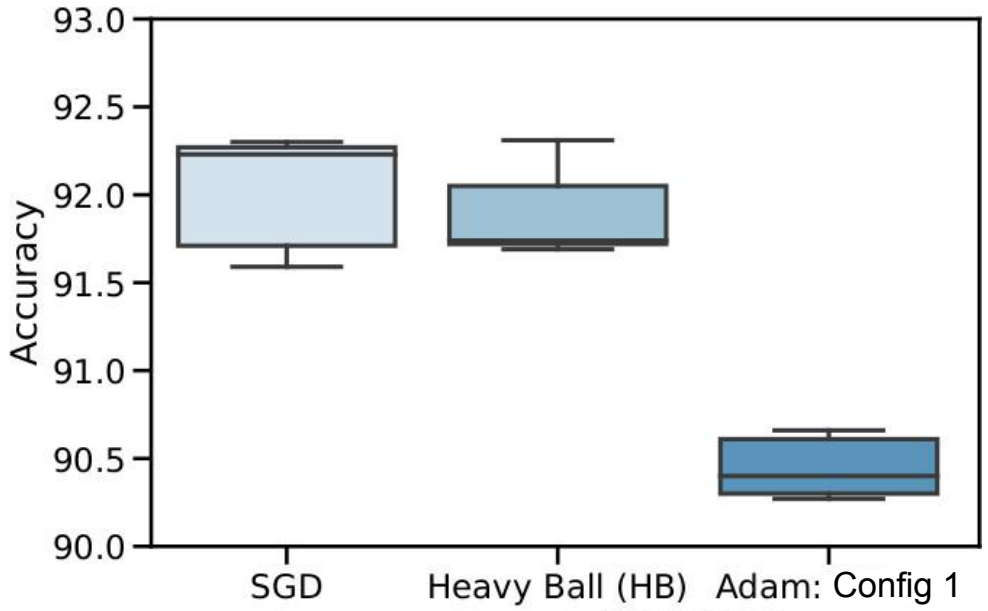
Hyperparameter Optimization Is Deceiving Us, and How to Stop It

A. Feder Cooper, Yucheng Lu, Jessica Zosa Forde, and Chris De Sa

It is well-known that Hyperparameter Optimization (HPO) greatly impacts algorithm performance

Using **grid search**, we find the best hyperparameter configuration for each optimizer

and conclude that non-adaptive optimizers outperform adaptive optimizers

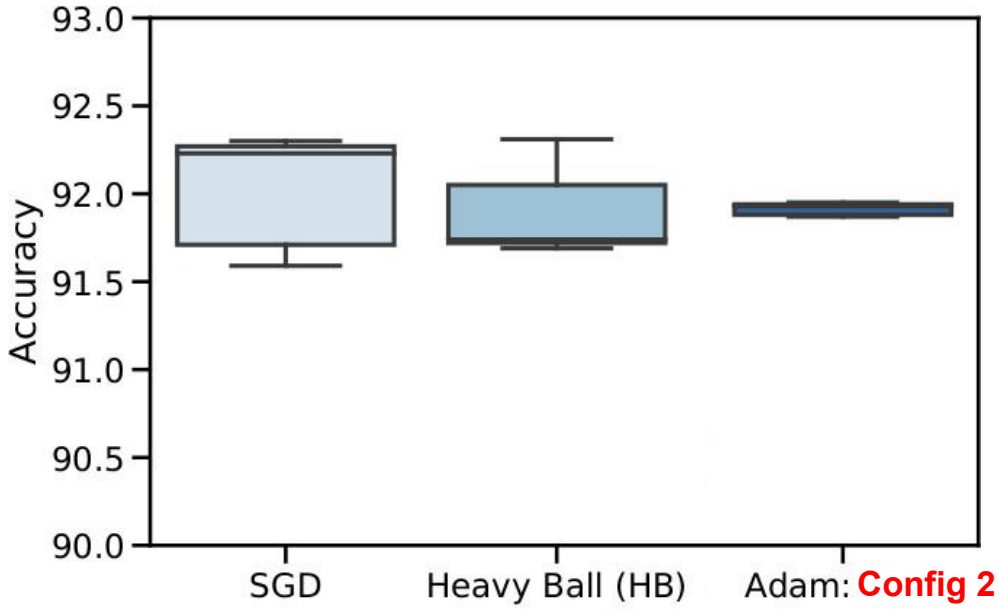


Optimizer test accuracy for VGG16 trained on CIFAR-10; n = 5 for each optimizer

It is well-known that Hyperparameter Optimization (HPO) greatly impacts algorithm performance

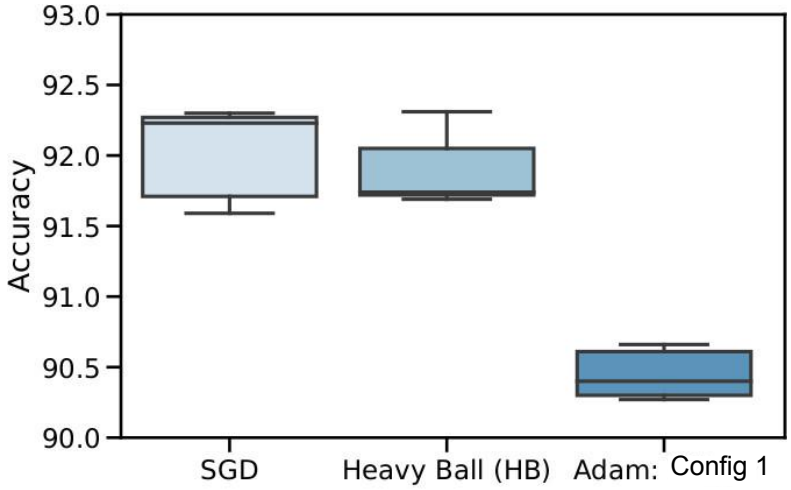
Changing HPO configuration can change our conclusions

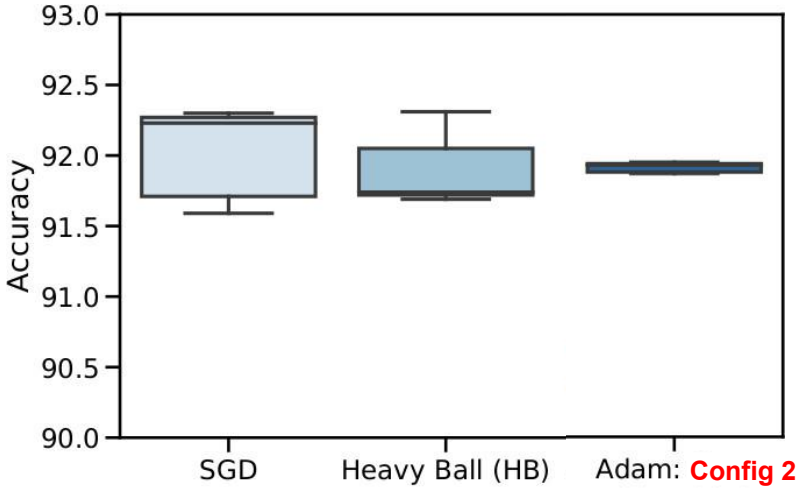
We instead conclude that non-adaptive optimizers do not outperform adaptive optimizers



Optimizer test accuracy for VGG16 trained on CIFAR-10; n = 5 for each optimizer

We want to prevent conclusions from depending on the underlying configuration of the HPO we perform



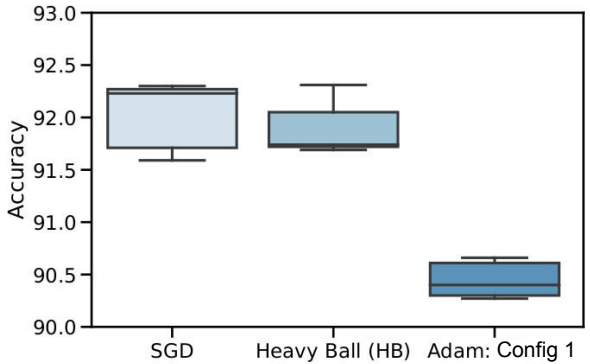


We therefore formalize the process of drawing conclusions from empirical studies using HPO

An HPO procedure

runs a **randomized algorithm** (e.g., grid search), with a particular **configuration** (e.g., grid points), to test a **set of hyperparameters** by running a **training algorithm** (e.g., SGD) to train a **model** (e.g., VGG16) on a **dataset**, using a **pseudorandom number generator**

and outputs the **chosen hyperparameter values** and a log documenting the run (which enables reproducibility)

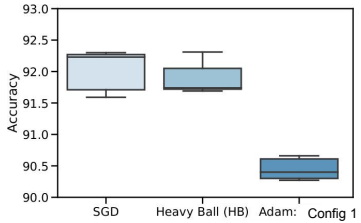


3 HPO procedures;
Each box plot corresponds to a log

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

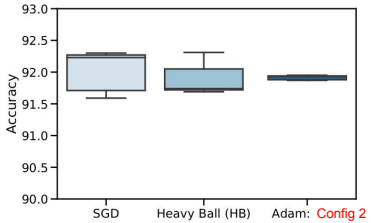
Epistemic Hyperparameter Optimization (EHPO) takes a **set of HPO procedures** and a function **F**, which maps a set of HPO procedure **logs** to **conclusions** about algorithm performance

Log set L1
(|L1| = 3)



F(L1) —> "Non-adaptive optimizers outperform adaptive ones"

Log set L2
(|L2| = 3)



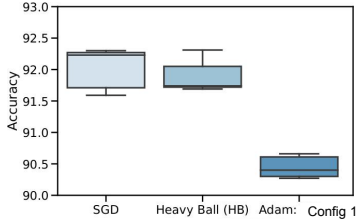
F(L2) —> "Non-adaptive optimizers do not outperform adaptive ones"

We therefore 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 procedure **logs** to **conclusions** about algorithm performance

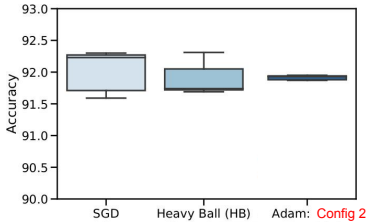
Logically inconsistent

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" = $\neg p$

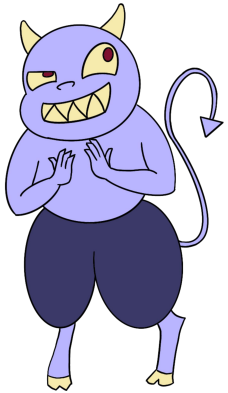
We frame drawing inconsistent conclusions from EHPO in terms of an adversary who can deceive us

Imagine an **evil demon** who

is trying to deceive us about relative algorithm performance via EHPO

maintains a set of log HPO logs which it can modify

presents us with a final log set, from which we can draw conclusions



We want to be sure that we will not be deceived about algorithm performance by any logs the demon ***could*** produce by changing HPO configurations

Modal logic is the standard way to formalize “could”

Modal logic extends propositional logic to allow us to reason about **possibility** introducing an additional operator

$\Diamond\phi$ eads, “It is **possible** that ϕ.”

Modal logic is the standard way to formalize “could”

Modal logic extends propositional logic to allow us to reason about **possibility** introducing an additional operator

$\Diamond\phi$ eads, “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 the outcomes of EHPO

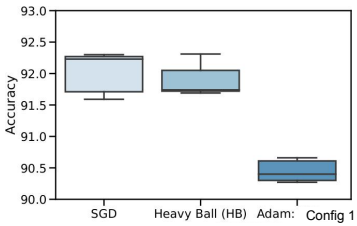
Expressing the possible outcomes of EHPO with $\Diamond_t p$

Syntax $\Diamond_t p$

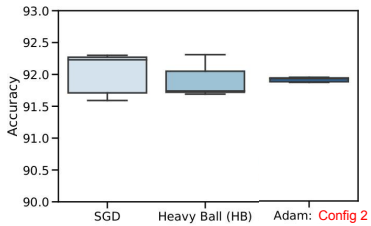
Intuition The demon could a adopt a strategy for running EHPO that is guaranteed to cause their desired outcome p in at most time *t* in expectation

Semantics A set of EHPO output logs **models** that it is possible pı time t

$\Diamond_t p$



$\Diamond_t \neg p$



Expressing our belief with $\mathcal{B}p$

Syntax $\mathcal{B}p$

Intuition We believe/ conclude p

Semantics The 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**

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

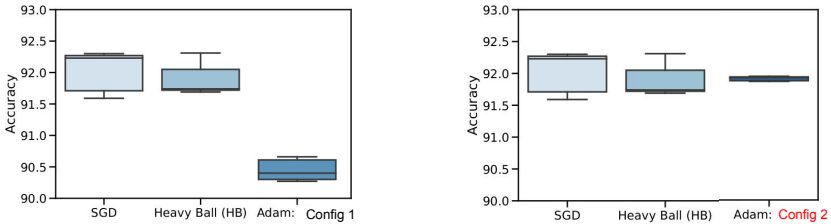
If it is **possible** for the **demon** can get us to **believe** p in time ***t***, then it is **not possible** for the **demon** to get us to **believe** $\neg p$ me ***t***

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

t-non-deceptive axiom

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

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** $\neg p$ me ***t***



Our motivating example is ***t*-deceptive**

Using this formalization, we can prove *t*-non-deceptive EHPO by showing they satisfy our axiom

Defense intuition:

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

Defense takeaway:

It is **always possible** to construct a t-defended EHPO

We suggest a concrete, *t*-non-deceptive EHPO

We describe a variation of **random search** that uses a produces a log with **K * R** random search trials

We divide the log into **R** logs each with **K** trials, and pass the divided log to an ensemble of **R naive reasoners**

The **defended reasoner** only concludes p if the R naive reasoners unanimously conclude p

In **sublinear-in-t time**, we can produce a log (of length **K * R**) that will allow our t-non-deceptive reasoner to reach conclusions

Takeaways

It is possible to construct t-defended EHPO, such as our t-defended random search

Any defense will depend on assumptions concerning how we configure underlying HPO, so researchers must be explicit about their configuration choices

Avoiding deception is just as important as ensuring reproducibility, as we want to ensure results are both replicable and correct