Agenda

Adapting to the changing world

Introduction to reinforcement learning and contextual bandits

Personalizer overview

Apprentice mode

Making contextual bandits work in practice

Q&A

Microsoft

Making contextual bandits work in practice

Paul Mineiro
Microsoft Research
@PaulMineiro

Debugging Learning Systems

How do you debug something that is expected to make mistakes?

Property Based Testing

Find something that is true in a correct system

Search for counterexamples

Property of Learning Algorithms

Performance of learned policy should not be too much worse than performance of best representable policy

Property of Learning Algorithms

<u>Performance</u> of learned policy should not be too much worse than <u>performance</u> of best representable policy

Evaluation

Evaluation is critical

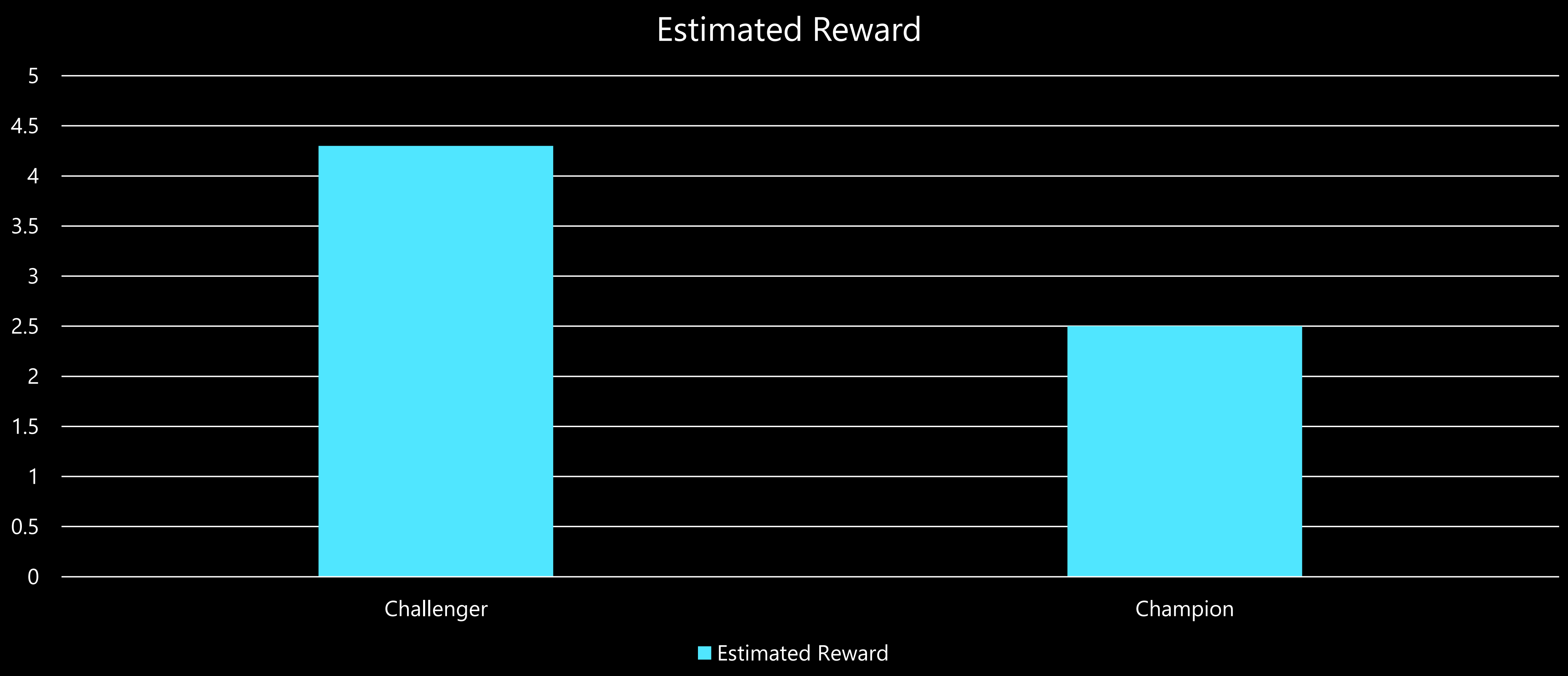
- Online evaluation is the gold standard
- Offline evaluation is critical for rapid iteration

Offline Evaluation

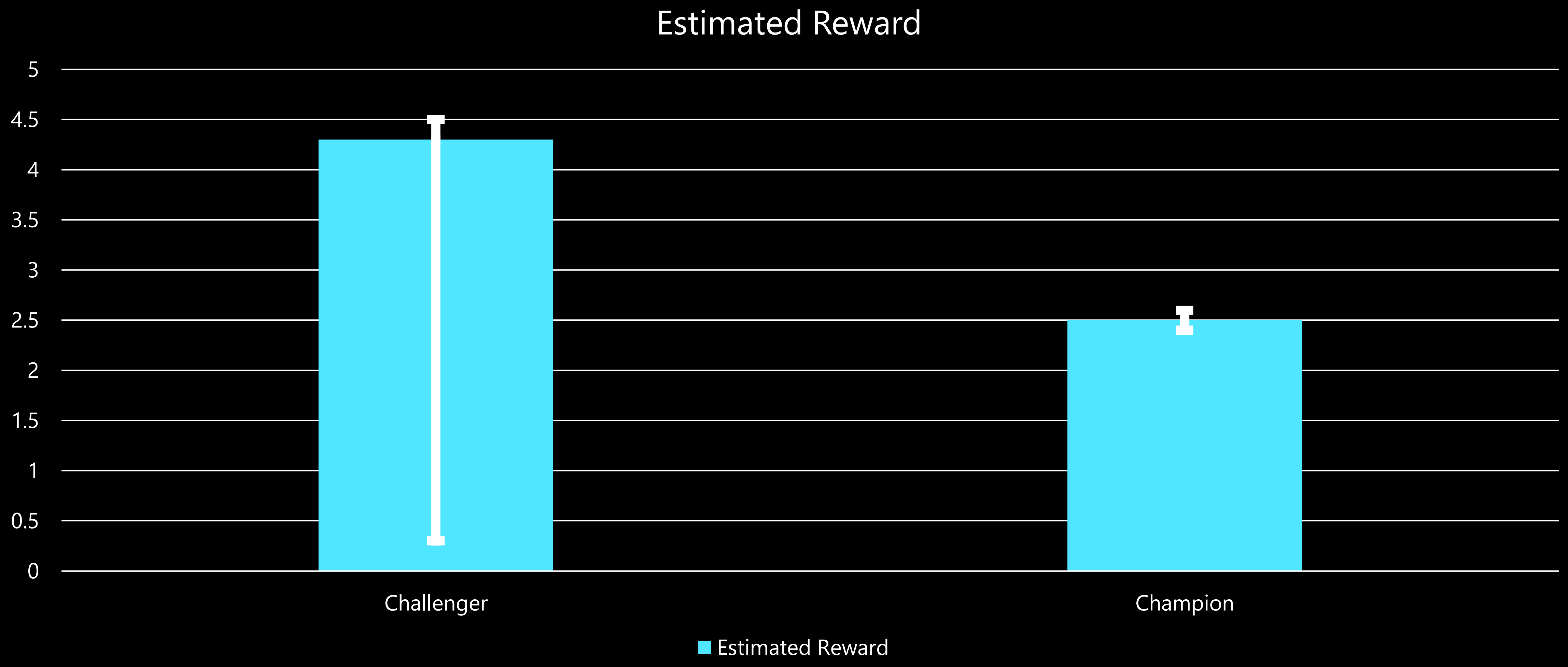
- Temporal: Respect time in train-test splits.
- CB: Must use a counterfactual estimator.
- Confidence Intervals: No point estimates ever!

Rant against point estimates

Which one is better?



How about now?



Always use confidence intervals

- Previous slide is typical in CBs but not supervised learning
- Challenger policy has better point estimate due to optimization, but
- Variance of CB estimate is higher for challenger policy.
- Tip: optimize lower bound with VW via `--cb_dro`

Bootstrap: Easy way to get CIs on (almost) anything

```
from numpy.random import choice as rc
from numpy import quantile
# resample testSet and compute estimate
samples = [ evaluate(rc(testSet, size=len(testSet)))
            for _ in range(100)
          ]
# get empirical quantiles from resampling
ninetyPctCI = quantile(testSet, q=[0.05, 0.95])
```

Property of Learning Algorithms

Performance of learned policy should not be too much worse than performance of best representable policy

I don’t know the performance of the best representable policy

Baseline

- A representable policy which you know is not very good.
- “not much worse than the best” implies “beat the baseline”.
- Example: choose an action uniformly at random.

Representable (?)

- To beat a baseline you must be able to mimic the baseline.
- Frequent fail: baseline has access to more information.
- Example: “Personalizer baseline”.

How to represent any baseline

- Use the prediction of the baseline as input to the new system.
- Performance jump → representation issue
- Performance the same → some other problem

Property of Learning Algorithms

Performance of learned policy should not be <u>too much worse</u> than performance of best representable policy

How much data is required for success?

How much data: theory answer

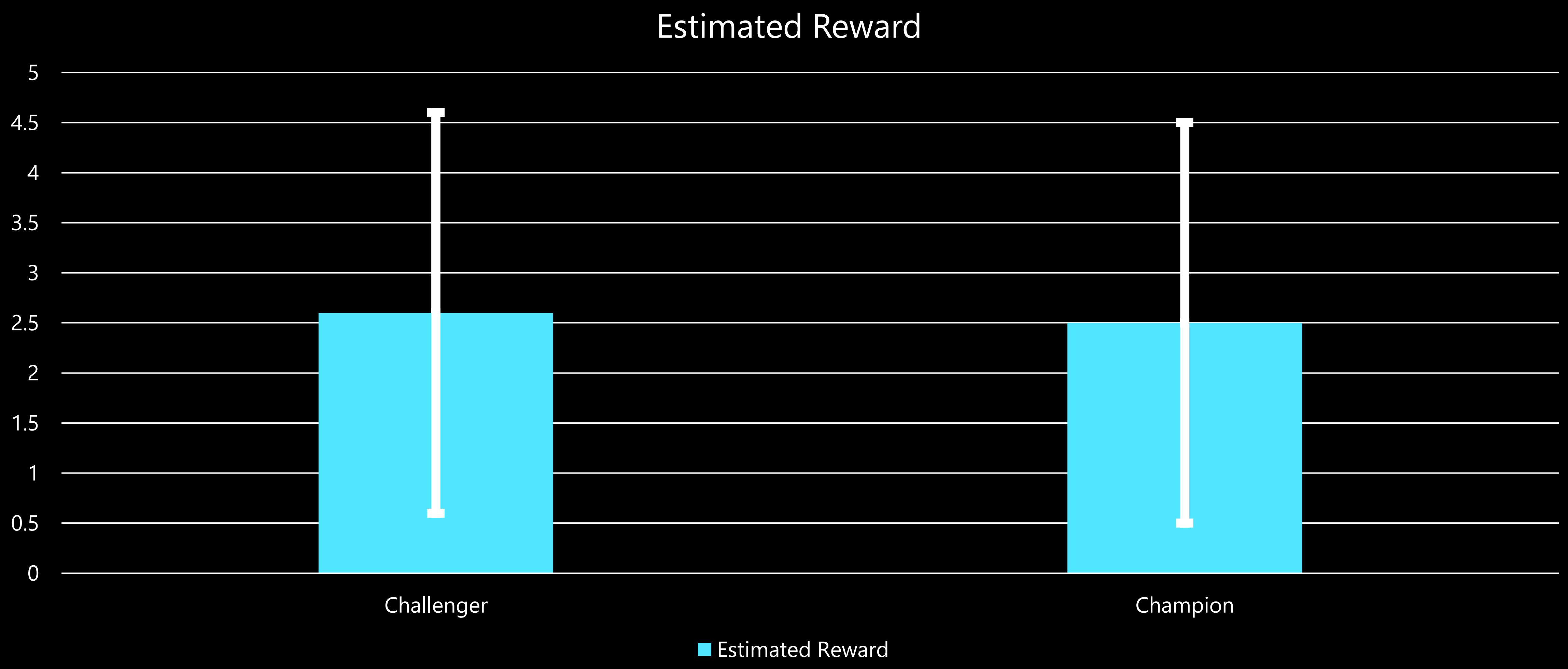
- More actions → more data required
- More reward variance → more data required
- Smaller lift → more data required

Do I have a bug or not enough data?

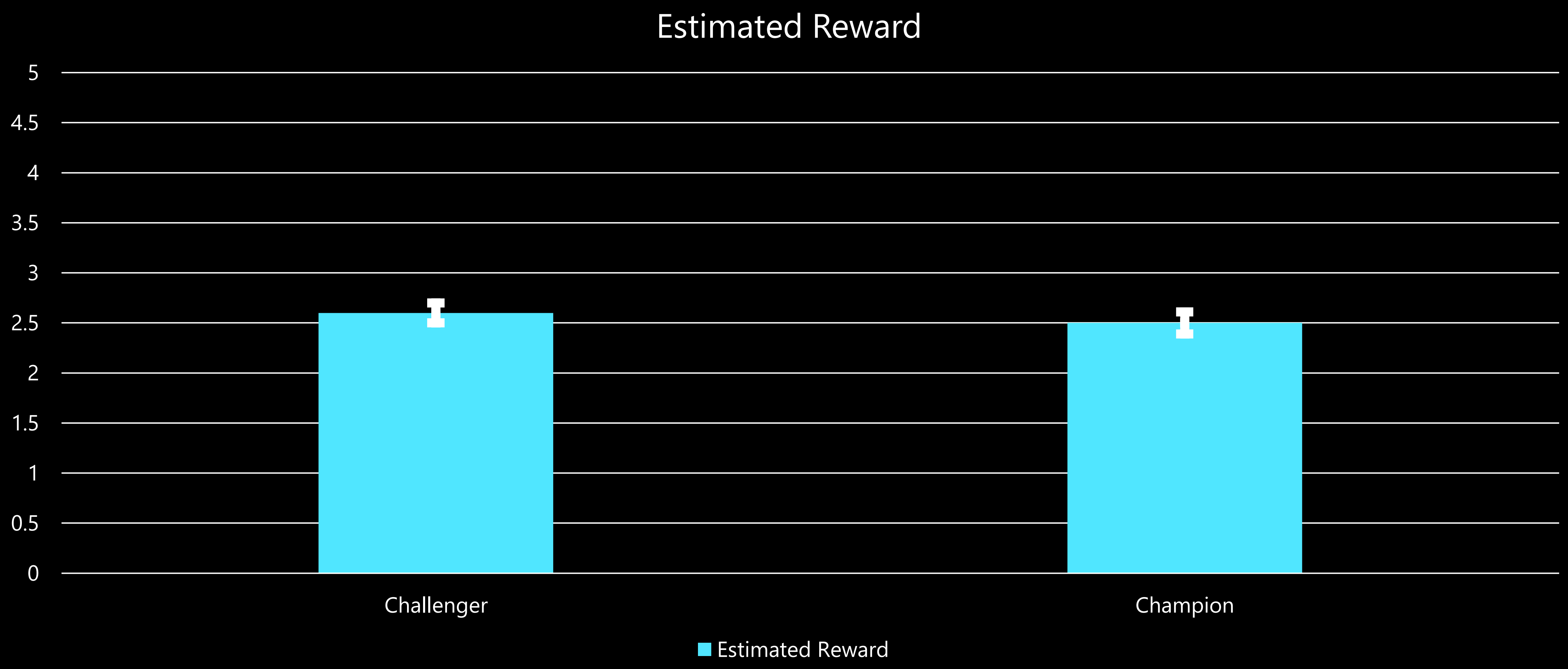
How much data: practical answer

- Confidence intervals are your friend.

You may need more data if you see this …



More data will not help if you see this …



Summary

Get the evaluation correct

Always use confidence intervals

Define baselines you can beat

Recap

Adapting to the changing world

Introduction to reinforcement learning and contextual bandits

Personalizer overview

Apprentice mode

Making contextual bandits work in practice

Q&A