Towards True Reproducibility of Findings in Cybersecurity Research

Emma Tosch, Northeastern University, 6 December 2022

Disclaimer

Design with the spirit of LASER in mind

How to capture

the essence of a keynote

without making it feel like

a speech or lecture?

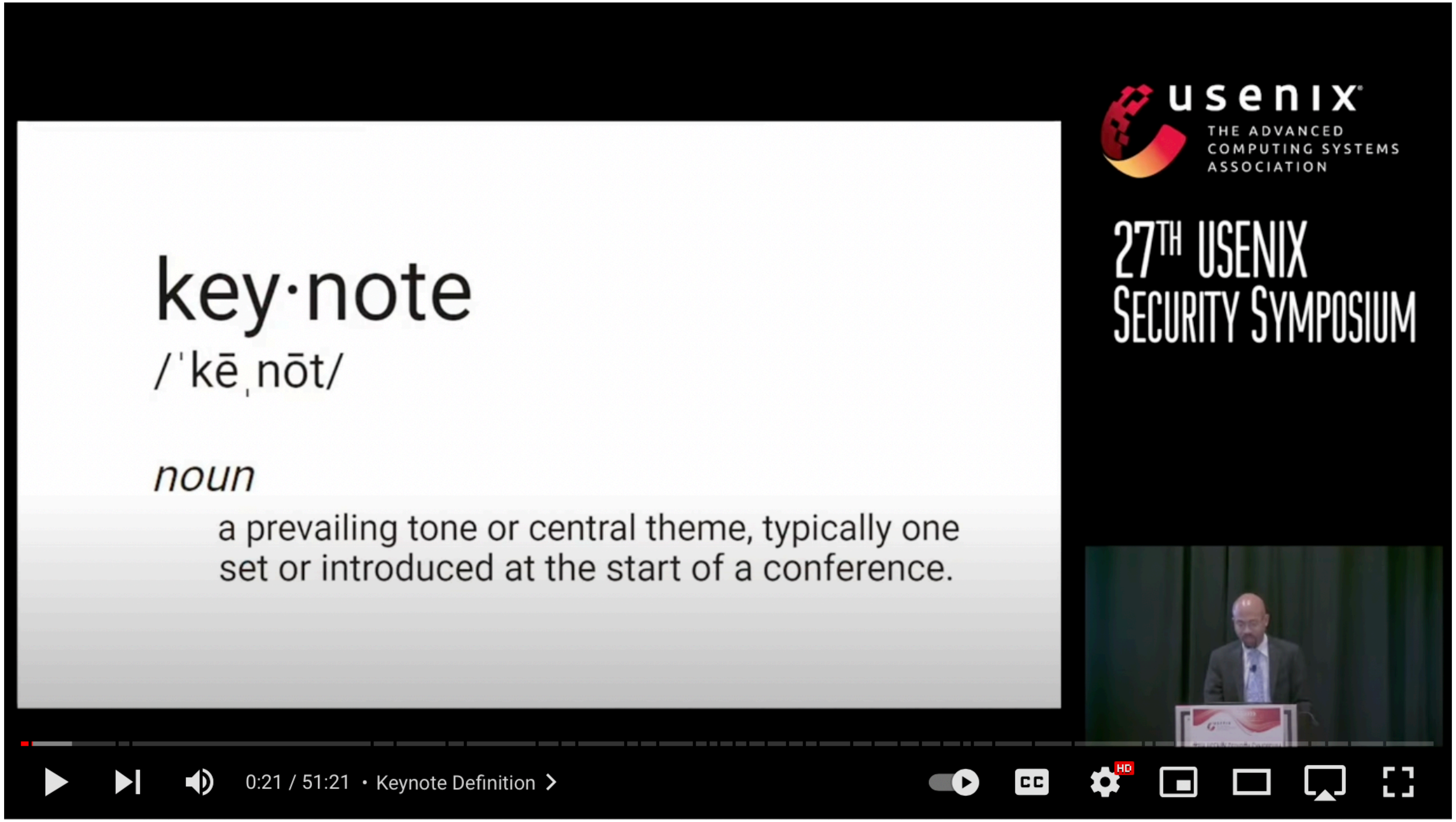
(c) NY state Department of Education

Disclaimer

Design with the spirit of LASER in mind

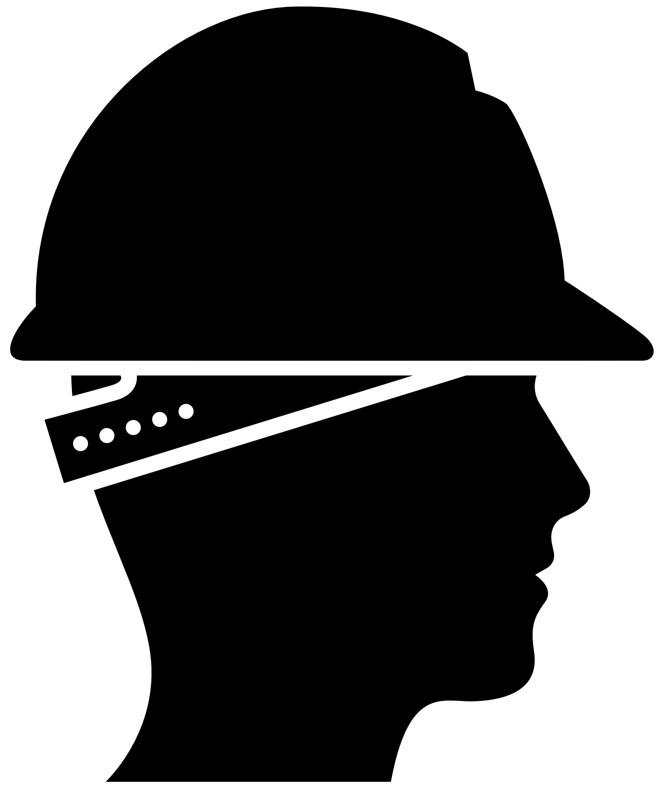
Disclaimer

Design with the spirit of LASER in mind

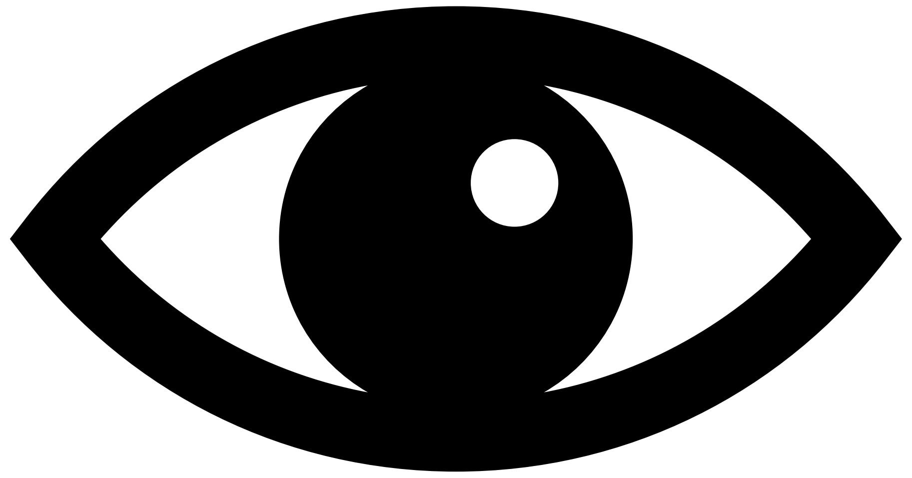


Disclaimer

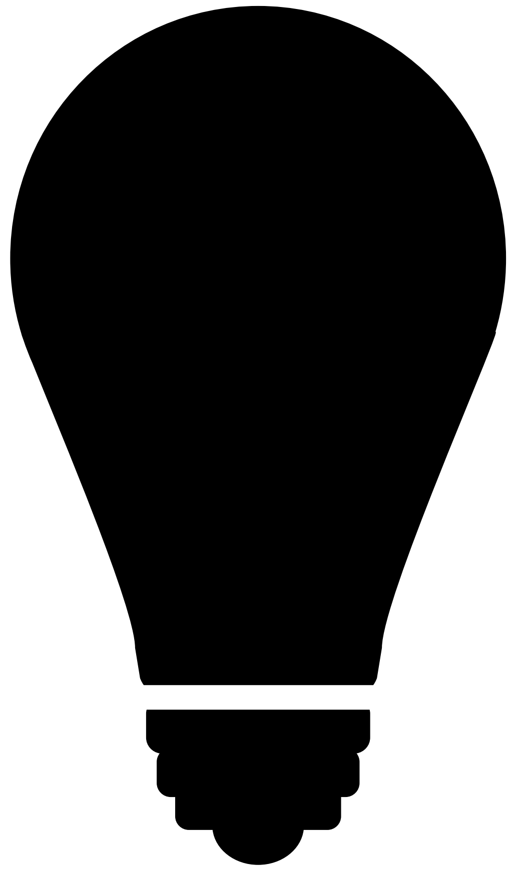
Design with the spirit of LASER in mind



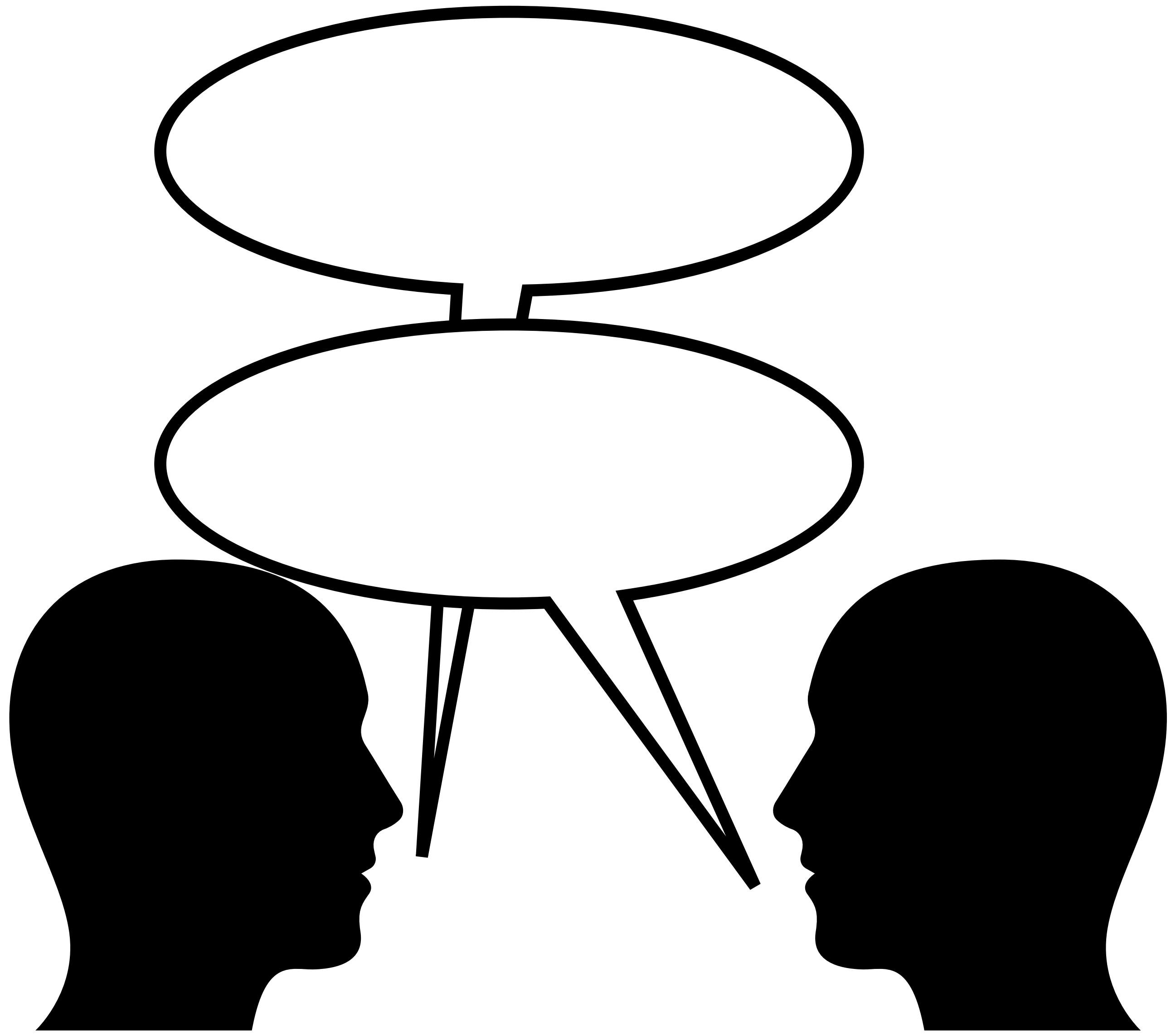
Work in progress



Speculation



Incomplete Ideas



Foster Discussion

Selfishly…

Outline

Oral History of Artifact Evaluation (student perspective)

Evaluators produce replicates

Language design for reproducibility

Formal languages unlock great power

Backstory

- September 2022: different workshop…
  - Full disclosure… completely forgot!
- 2013/14 — lab mates participated in one of the earliest AECs for SIGPLAN
- 2014 — submitted artifact (OOPSLA 2014)
- 2014 — began AEC review (POPL 2015)

HOWTO for AEC Submitters
(http://bit.ly/HOWTO-AEC)
(Last updated May 2022)

Dan Barowy (dbarowy@cs.umass.edu) - *now at Williams College*
Charlie Curtsinger (charlie@cs.umass.edu) - *now at Grinnell College*
Emma Tosch (etosch@cs.umass.edu) - *now University of Vermont*
John Vilk (jvilk@cs.umass.edu) - *now at Stripe*
of the PLASMA group (http://plasma.cs.umass.edu) at University of Massachusetts Amherst
with encouragement and support from Emery Berger (emery@cs.umass.edu)

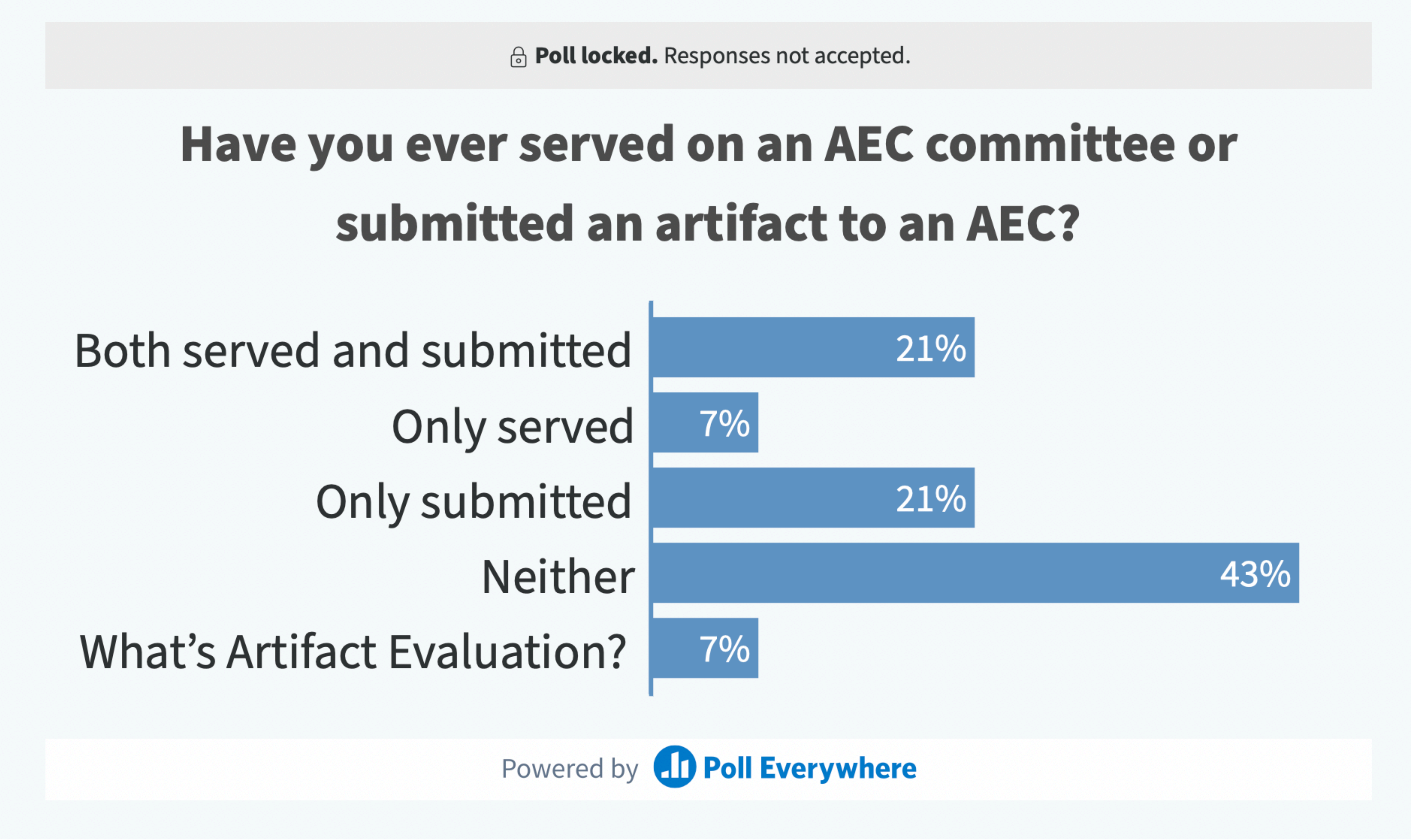
After serving on several Artifact Evaluation Committees and winning two Distinguished Artifact Awards, we put together this HOWTO document to help you submit an artifact that will pass the AEC process with flying colors.

How to Build a Good Software Artifact

1. Provide documentation with your artifact. We recommend that you prepare a Getting Started Guide. It should explain:
   a. how to download your artifact
   b. how to install your artifact
   c. how to run your artifact
   d. how to compare your artifact's outputs to outputs described in your paper.
2. Explicitly enumerate your claims in both your paper and in your artifact's documentation.
3. Provide a VM if possible, and when appropriate. VMs aid reproducibility because they help control for nuisance factors that are not central to an author's claims, significantly facilitating the review process. Nonetheless, reviewers may need to accept performance tradeoffs for VMs (e.g., because of the absence of special hardware). These tradeoffs are acceptable as long as authors explain to reviewers how and why they should adjust their expectations.
4. Provide step-by-step instructions, but make it easy for reviewers to supply their own inputs to your artifact. When reviewers can "play" with your artifact, it gives them confidence that your ideas were implemented robustly.

Source Code

1. If you are not bound by a nondisclosure agreement, make every effort to supply reviewers with source code. Good reviewers may read and modify your source code to learn the true capabilities of your artifact.
2. Document your code. You should sufficiently explain what is going on so that people who want to build on your work can do so.
3. If you discuss a new algorithm or unique implementation approach in your paper, have a reference to its implementation in the source code.


Poll locked. Responses not accepted.
Have you ever served on an AEC committee or submitted an artifact to an AEC?
Both served and submitted
21%
Only served
7%
Only submitted
21%
Neither
43%
What’s Artifact Evaluation?
7%
Powered by Poll Everywhere

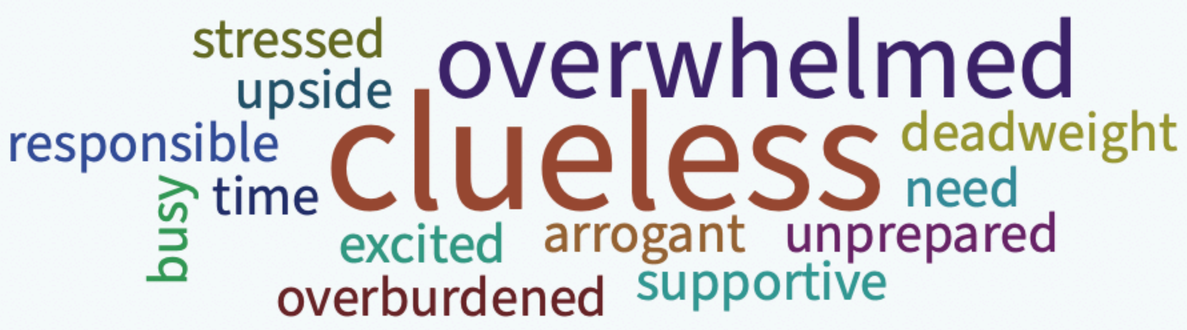
When poll is active, respond at **PollEv.com/emmatosch585**
Text **EMMATOSCH585** to **37607** once to join

The idea of submitting to artifact evaluation makes me feel...

Powered by Poll Everywhere

When poll is active, respond at **PollEv.com/emmatosch585**
Text **EMMATOSCH585** to **37607** once to join

The idea of serving on an artifact evaluation committee makes me feel..



Powered by **Poll Everywhere**

The rise and fall of expectations

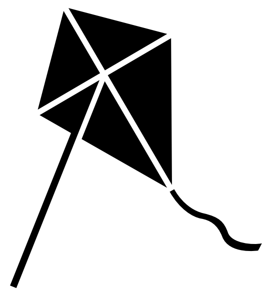
Student perspective

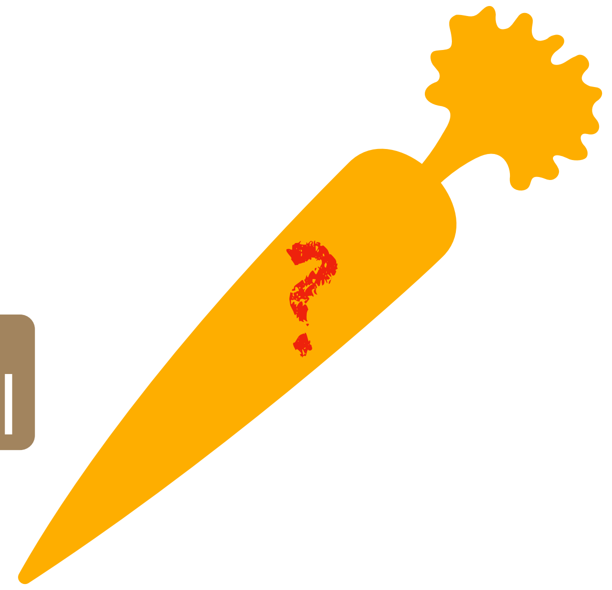
Process early on

1. Read abstract, note expectations *set by abstract*
2. Read paper, revise expectations, *in light of the paper*
3. Write out expected software components and datasets*
4. Sketch a plan for something novel to do with the software
5. Early days: no separate guide

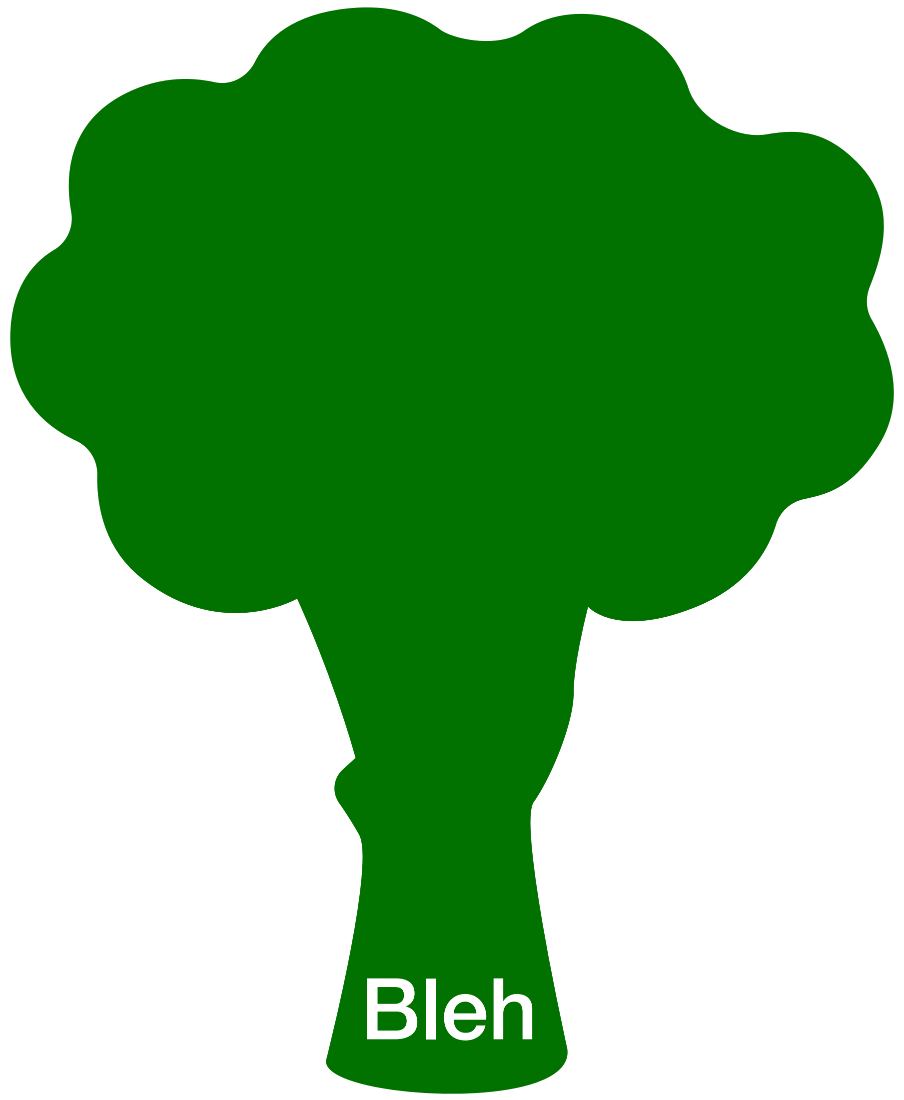
Student perspective

Reality





- Retrospective: assumed goal was reusability
    - *Then*: one badge. *Now*: Five
- Arguments in favor (at the time)
    - Promote best practices
    - Disincentivize “runs on my machine”
    - Temper reader’s expectations (inflated abstracts)



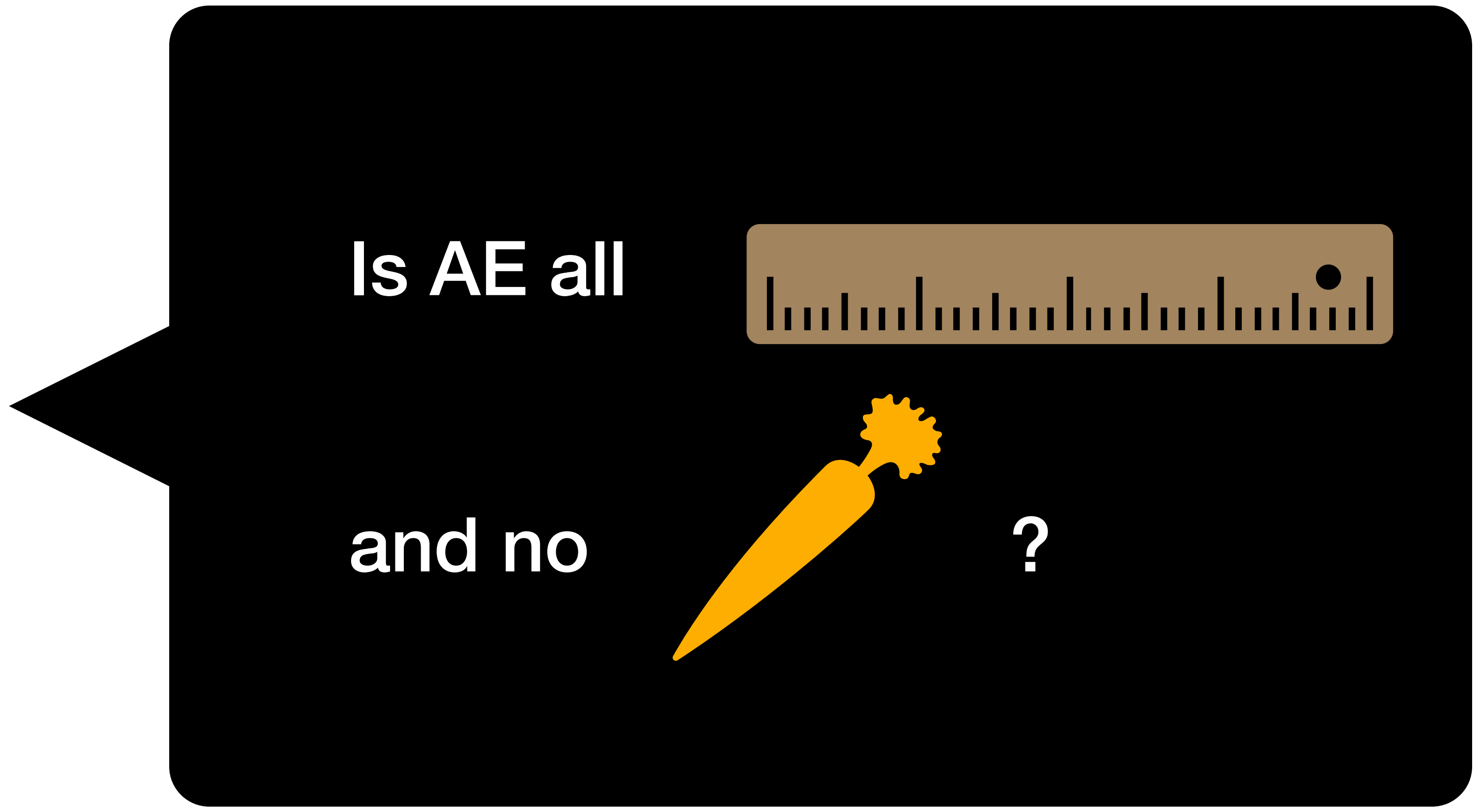
Only ever submitted once…

Why do we cite papers in the first place?

To please Reviewer #2.

Why do we cite papers in the first place?

- Findings
  - Don't *want* to have to start from scratch
- Contributions
  - New Software
  - New Datasets
  - New Methods
  - New Research Areas

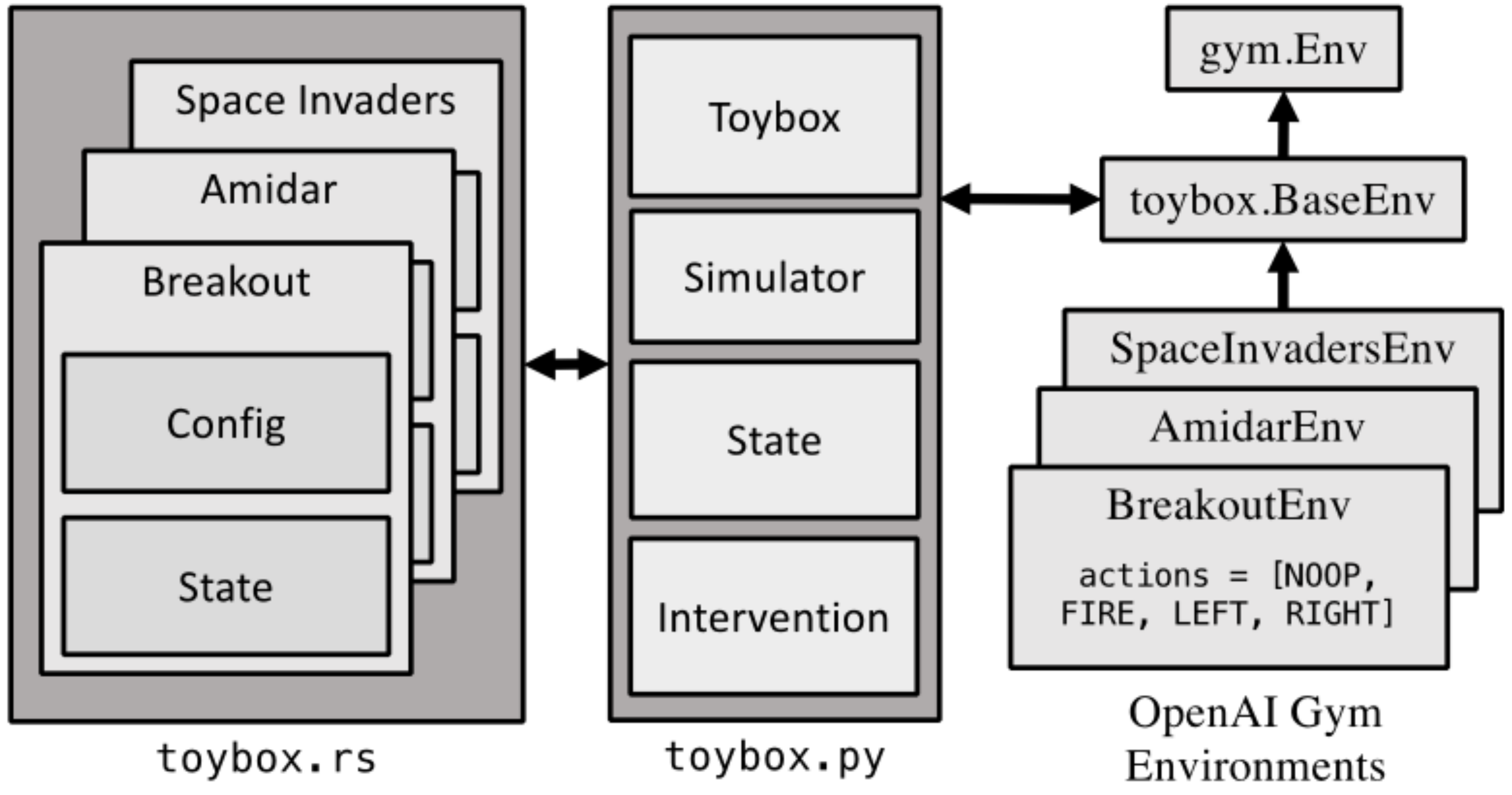


Software will be cited if it works*

…regardless of AE results

- Incentive: Public artifact
    - Don’t need artifact eval
    - Do we even want users?
        - Parable of SurveyMan

- Incentive: Good citizenship
    - Stand on the shoulder of giants!
    - Have you ever used someone else’s artifact? (Not repo)



Toybox

The Machine Learning Toybox for testing of Atari Reinforcement Learning Agents.

View My GitHub Profile

toybox.rs

Welcome to toybox.rs! This is the main organization and point of entry for using the Toybox platform for testing and experimentating with autonomous agents.

- Main repository with tests/experimentation support provided by a customized openai/baselines: toybox-rs/Toybox
- Core repository with implementations of the games: toybox-rs/toybox-rs. Releases available on PyPI: pypi package 0.5.0

What is Toybox?

Toybox is a set of *highly intervenable* environments for testing autonomous agents. While our efforts have focused on the efficient testing of deep RL agents, this work can be used in a variety of contexts that involve white-box testing of black-box agents.

If you use this code, or otherwise are inspired by our white-box testing approach, please cite our NeurIPS workshop paper:

Not able to convince collaborators to submit

What about student evaluators?

Student perspective

(Students: feel free to share your thoughts)

- I liked serving on AECs
    - I learned new technologies
    - Reading others' code makes your code better
    - Scalable training in methods
- Other incentives:
    - Be on a PC (now students officially on PCs)
    - Early on: part of something important
- Problem: evaluation is a lot of work

How to find more appealing carrots?

What do student stakeholders want out of the process?

Artifact Evaluators as contributors

Proposition 1

Submit… something else

Proposition 2

Outline

Oral History of Artifact Evaluation (student perspective)

Evaluators produce replicates

Language design for reproducibility

Where's the carrot?

Answer: In empirical evaluation.

Focus efforts on replication

Eval as improvement to the science

The value of replicates…

DISCLAIMER: Work in Progress

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Abstract

This paper presents a surprising result: changing a seemingly innocuous aspect of an experimental setup can cause a systems researcher to draw wrong conclusions from an experiment. What appears to be an innocuous aspect in the experimental setup may in fact introduce a significant bias in an evaluation. This phenomenon is called *measurement bias* in the natural and social sciences.

Our results demonstrate that measurement bias is significant and commonplace in computer system evaluation. By *significant* we mean that measurement bias can lead to a performance analysis that either over-states an effect or even yields an incorrect conclusion. By *commonplace* we mean that measurement bias occurs in all architectures that we tried (Pentium 4, Core 2, and m5 O3CPU), both compilers that we tried (gcc and Intel's C compiler), and most of the SPEC CPU2006 C programs. Thus, we cannot ignore measurement bias. Nevertheless, in a literature survey of 133 recent papers from ASPLOS, PACT, PLDI, and CGO, we determined that none of the papers with experimental results adequately consider measurement bias.

Inspired by similar problems and their solutions in other sciences, we describe and demonstrate two methods, one for detecting (causal analysis) and one for avoiding (setup randomization) measurement bias.

Categories and Subject Descriptors C. Computer Systems Organization [*C.4 Performance of Systems*]: Design studies

General Terms Experimentation, Measurement, Performance

Keywords Measurement; Bias; Performance

ASPLOS'09, March 7–11, 2009, Washington, DC, USA.

1. Introduction

Systems researchers often use experiments to drive their work: they use experiments to identify bottlenecks and then again to determine if their optimizations for addressing the bottlenecks are effective. If the experiment is biased then a researcher may draw an incorrect conclusion: she may end up wasting time on something that is not really a problem and may conclude that her optimization is beneficial even when it is not.

We show that experimental setups are often biased. For example, consider a researcher who wants to determine if optimization O is beneficial for system S. If she measures S and $S + O$ in an experimental setup that favors $S + O$, she may overstate the effect of O or even conclude that O is beneficial even when it is not. This phenomenon is called *measurement bias* in the natural and social sciences. This paper shows that measurement bias is commonplace and significant: it can easily lead to a performance analysis that yields incorrect conclusions.

To understand the impact of measurement bias, we investigate, as an example, whether or not *O3* optimizations are beneficial to program performance when the experimental setups differ. Specifically, we consider experimental setups that differ along two dimensions: (i) UNIX environment size (i.e., total number of bytes required to store the environment variables) because it affects the alignment of stack allocated data; and (ii) link order (the order of *.o* files that we give to the linker) because it affects code and data layout. There are numerous ways of affecting memory layout; we picked two to make the points in this paper but we have found similar phenomena with the others that we have tried.

We show that changing the experimental setup often leads to contradictory conclusions about the speedup of *O3*. By "speedup of *O3*" we mean run time with optimization level *O2* divided by run time with optimization level *O3*. To increase the generality of our results, we present data from two microprocessors, Pentium 4 and Core 2, and one simulator, m5 O3CPU [2]. To ensure that our results are not limited to gcc, we show that the same phenomena also appear when we use Intel's C compiler.

Evaluators produce replicates

Focus efforts on replication

Eval as improvement to the science

Two values; assign equal weight

The value of replicates…

Belief O

Specify as a causal graphical model

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

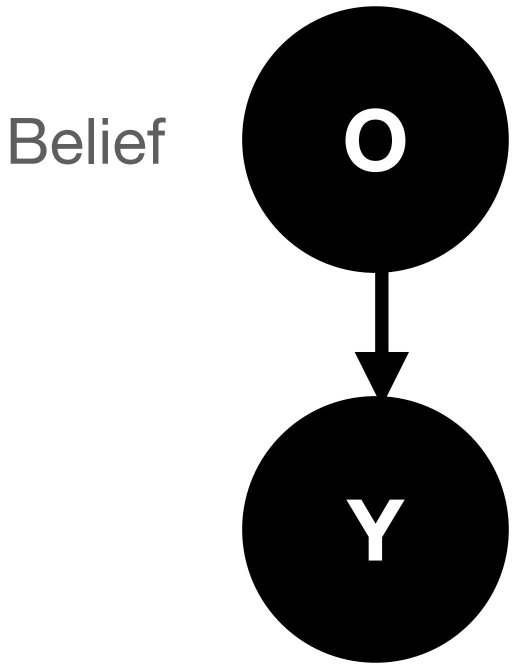
Evaluators produce replicates

Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model



Performance

Over what???

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

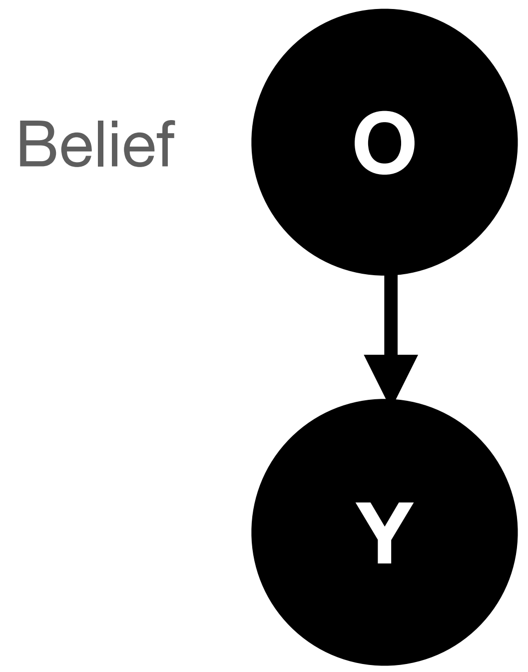
Evaluators produce replicates

Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model



Performance

Over what???

Population of all possible programs on my machine

Population of all possible programs on all suitable machines

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

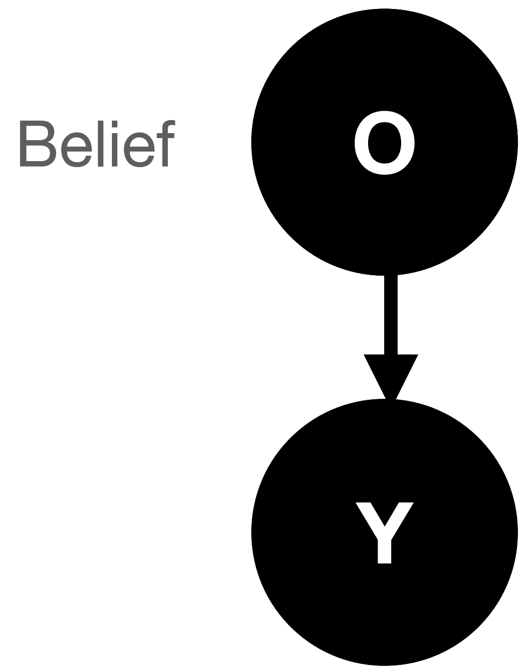
Evaluators produce replicates

Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model



Population of all possible programs on my machine

Population of all possible programs on all suitable machines

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

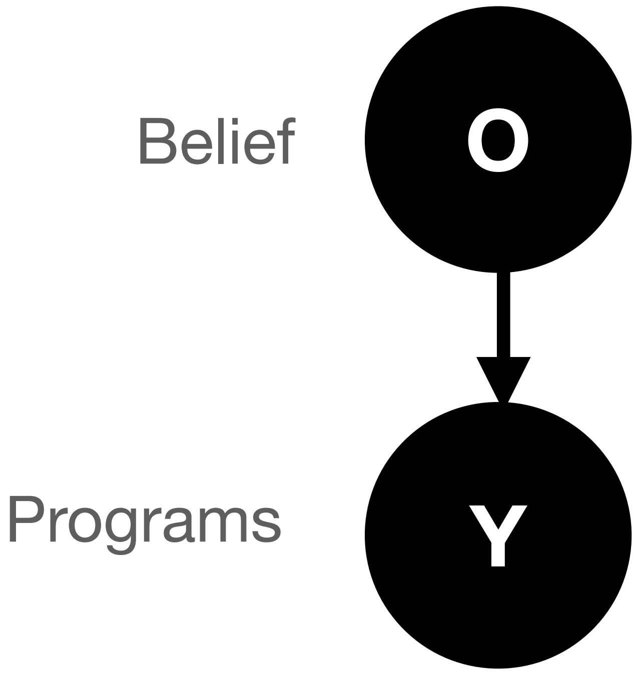
Evaluators produce replicates

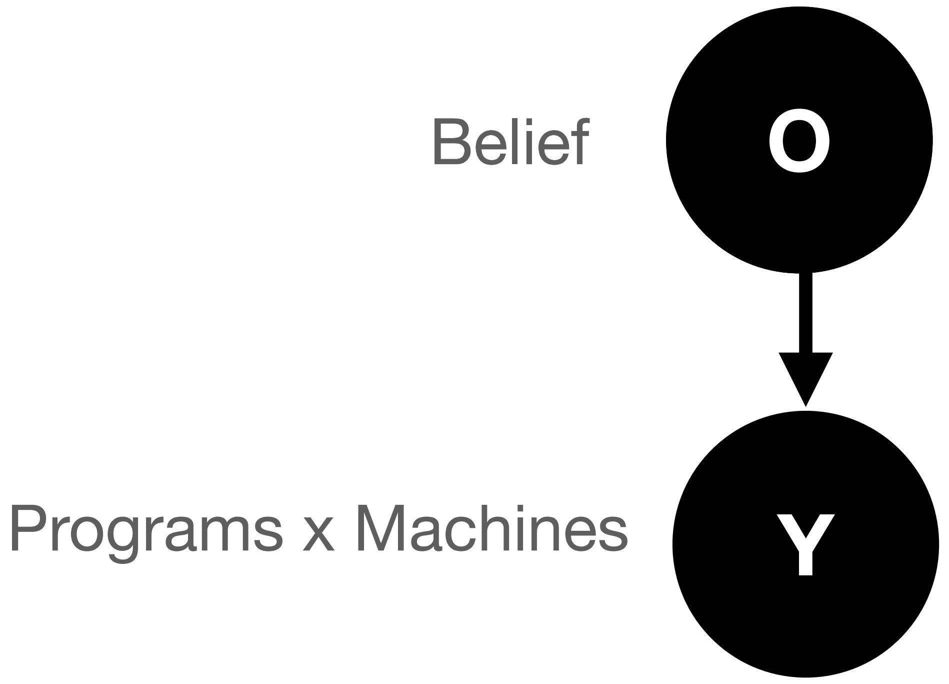
Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model





Population of all possible programs on my machine

Population of all possible programs on all suitable machines

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

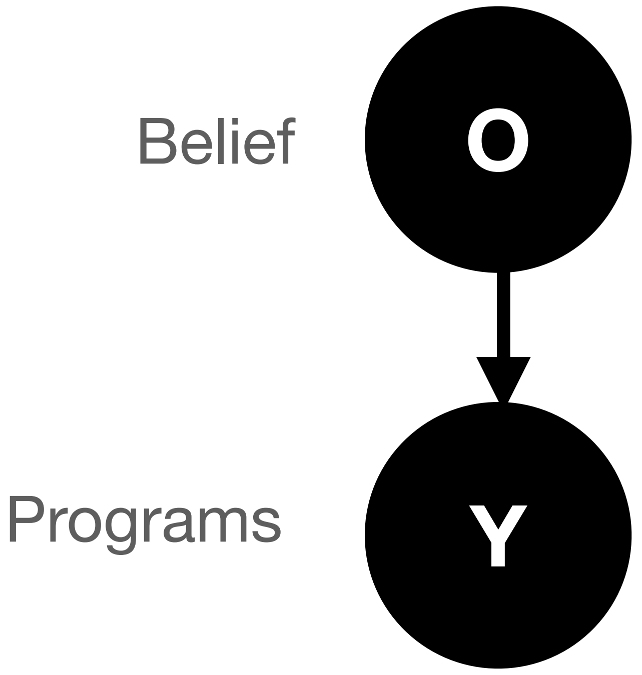
Evaluators produce replicates

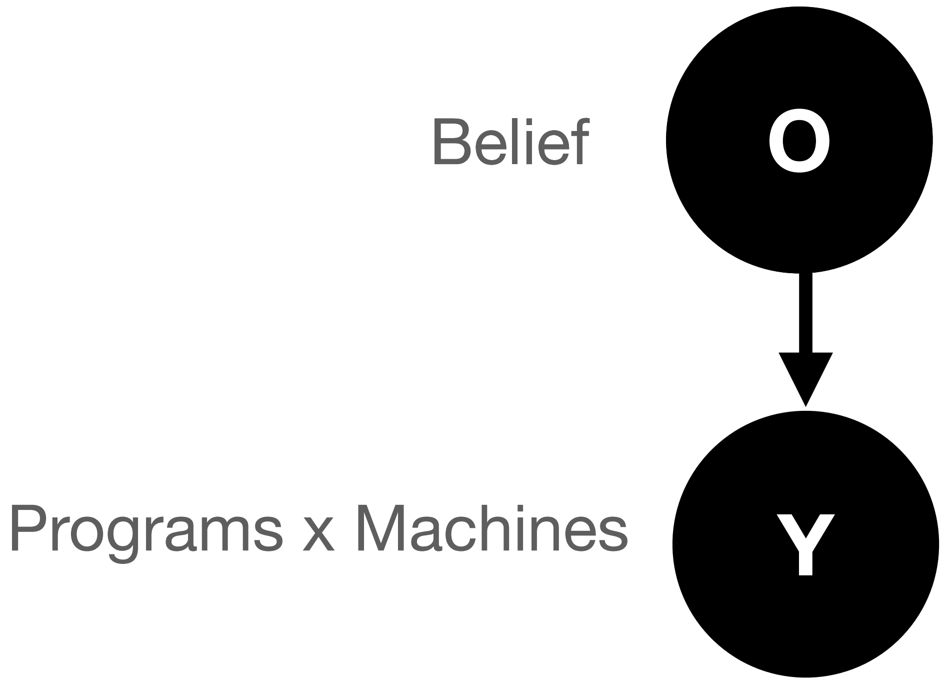
Focus efforts on replication

Eval as improvement to the science

The value of <u>replicates</u>…

Specify as a causal graphical model





Population of all possible programs on <u>my</u> machine

Population of all possible programs <u>on all suitable</u> machines

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

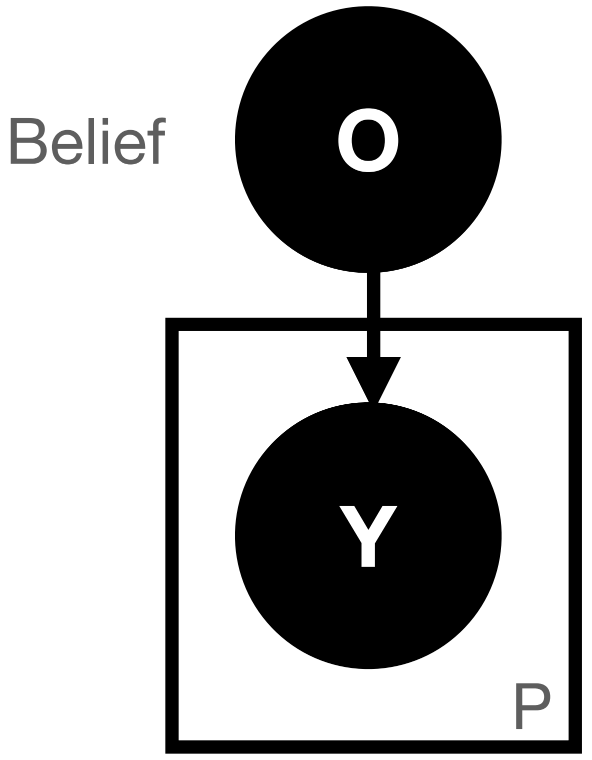
Evaluators produce replicates

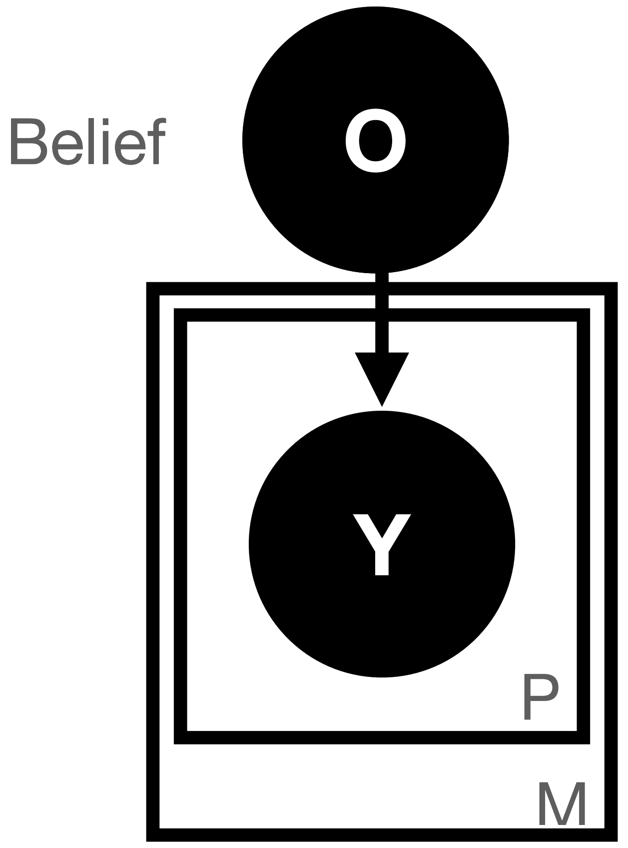
Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model





Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

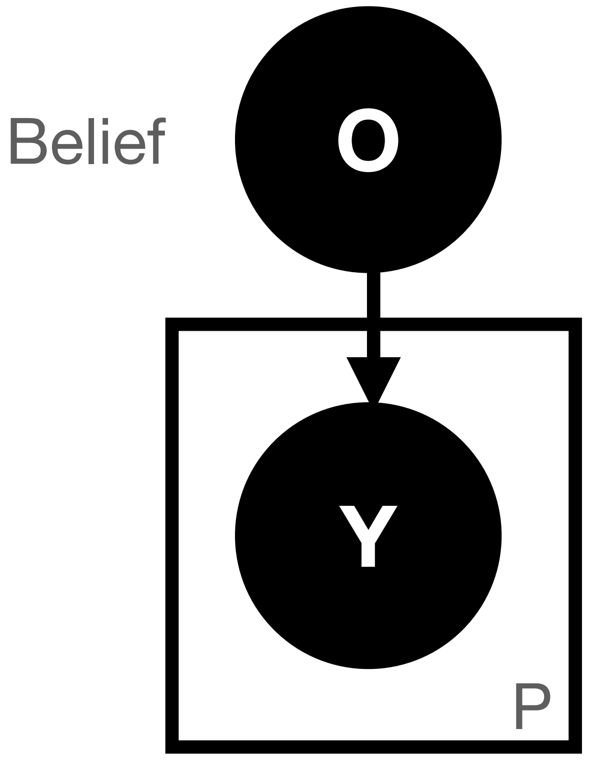
Focus efforts on replication

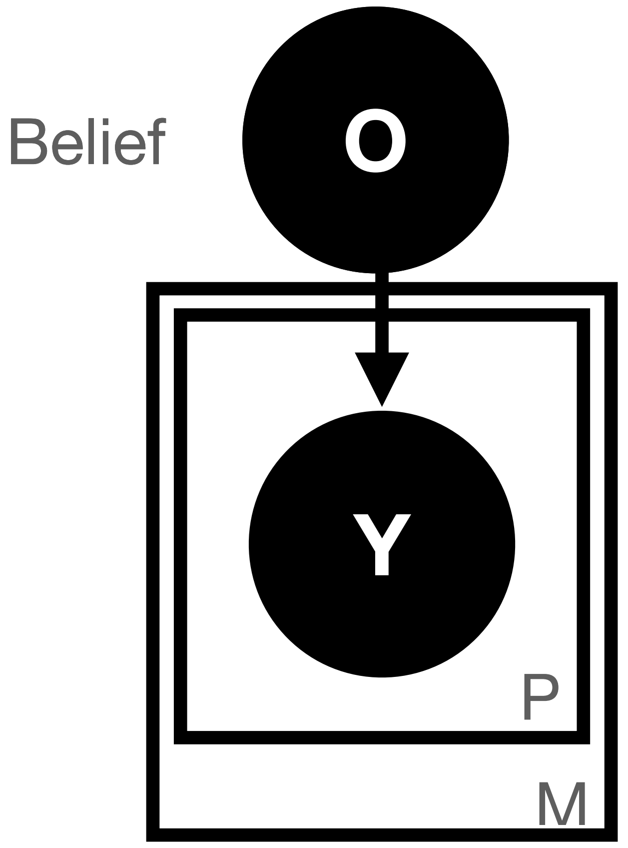
Eval as improvement to the science

Sample is fixed! (Benchmark)

The value of replicates…

Specify as a causal graphical model





Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

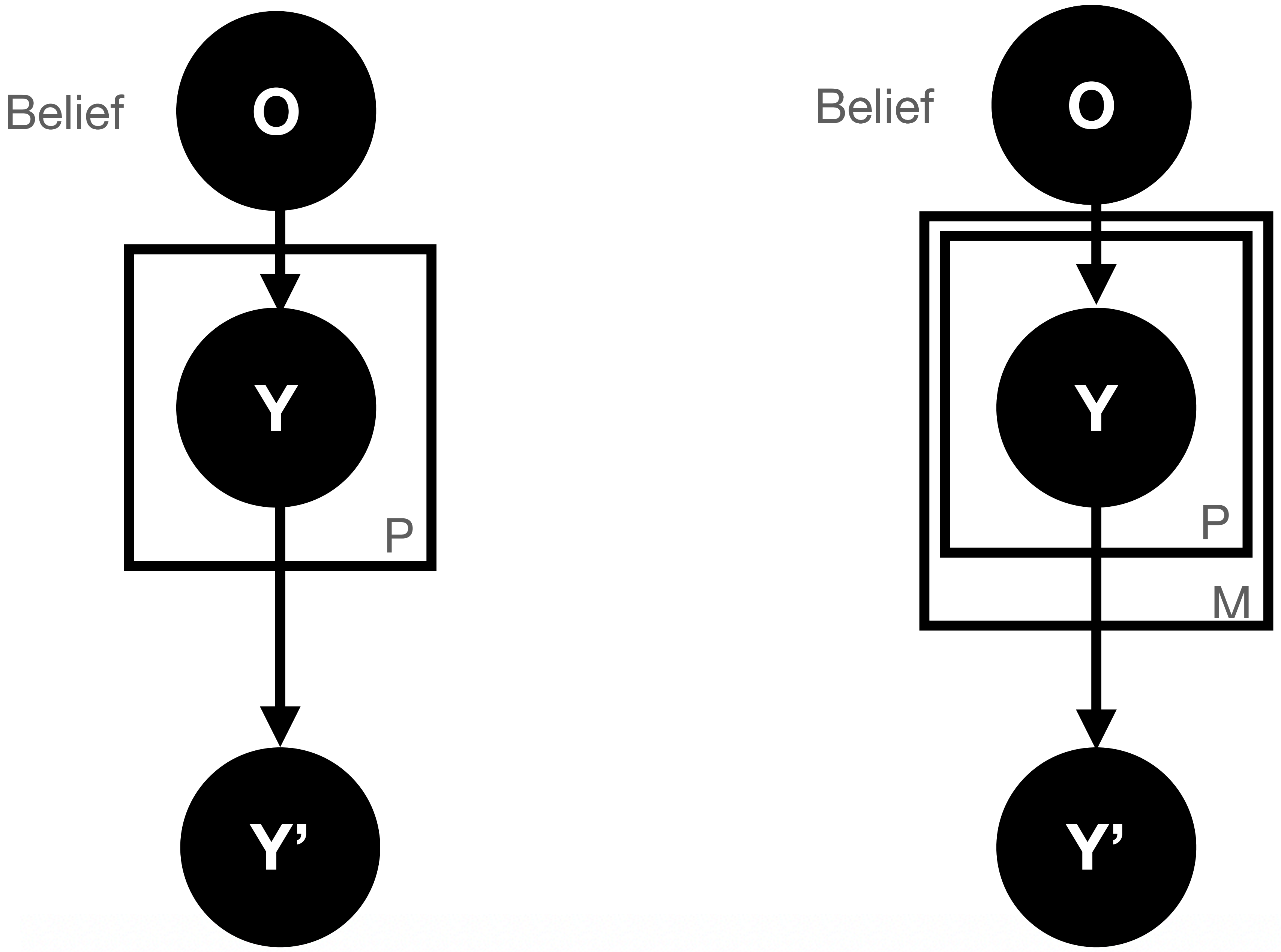
Focus efforts on replication

Eval as improvement to the science

Sample is fixed! (Benchmark)

The value of replicates…

Specify as a causal graphical model



Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

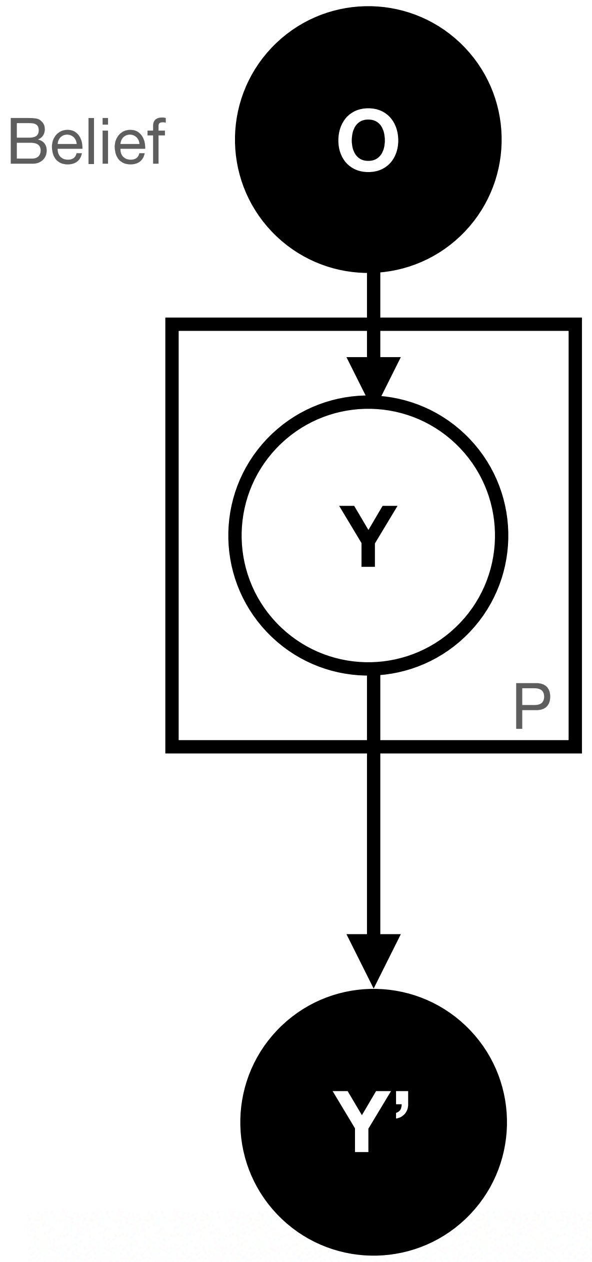
Focus efforts on replication

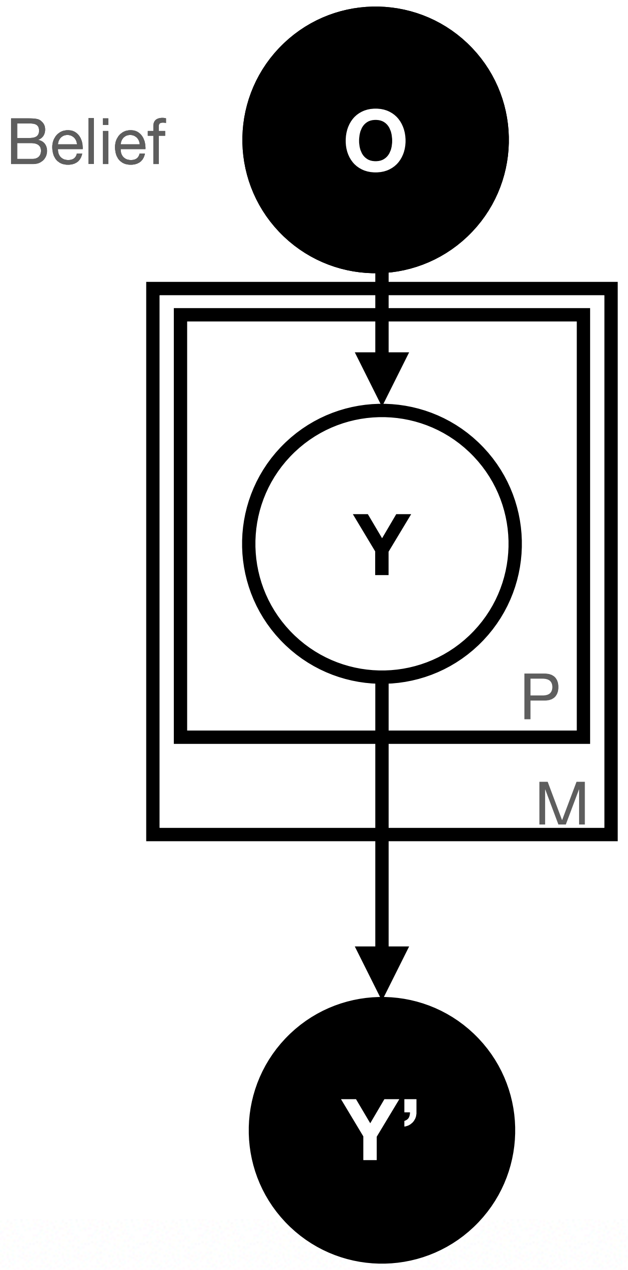
Eval as improvement to the science

Sample is fixed! (Benchmark)

The value of replicates…

Specify as a causal graphical model





Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

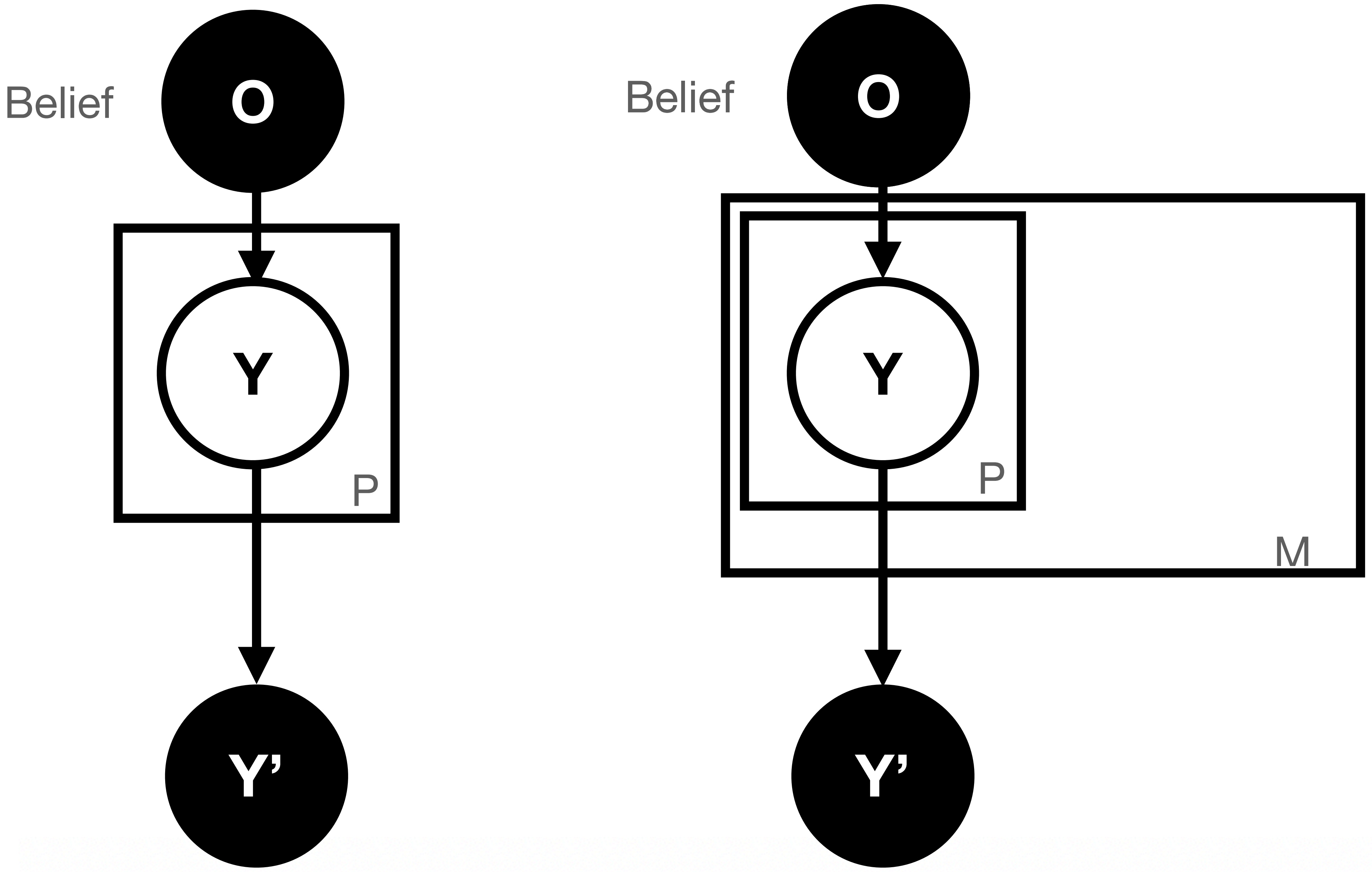
Evaluators produce replicates

Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model



Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

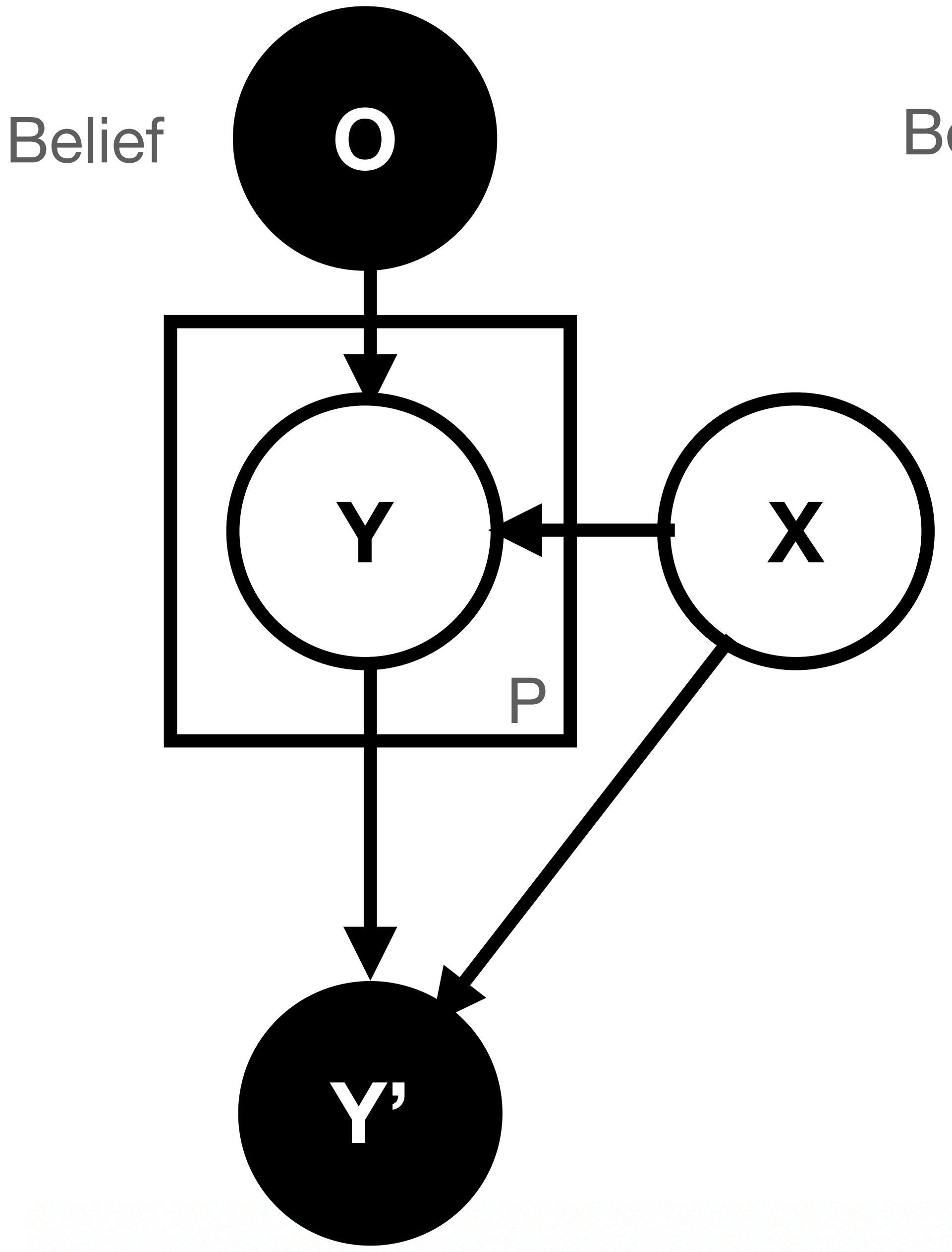
Focus efforts on replication

Eval as improvement to the science

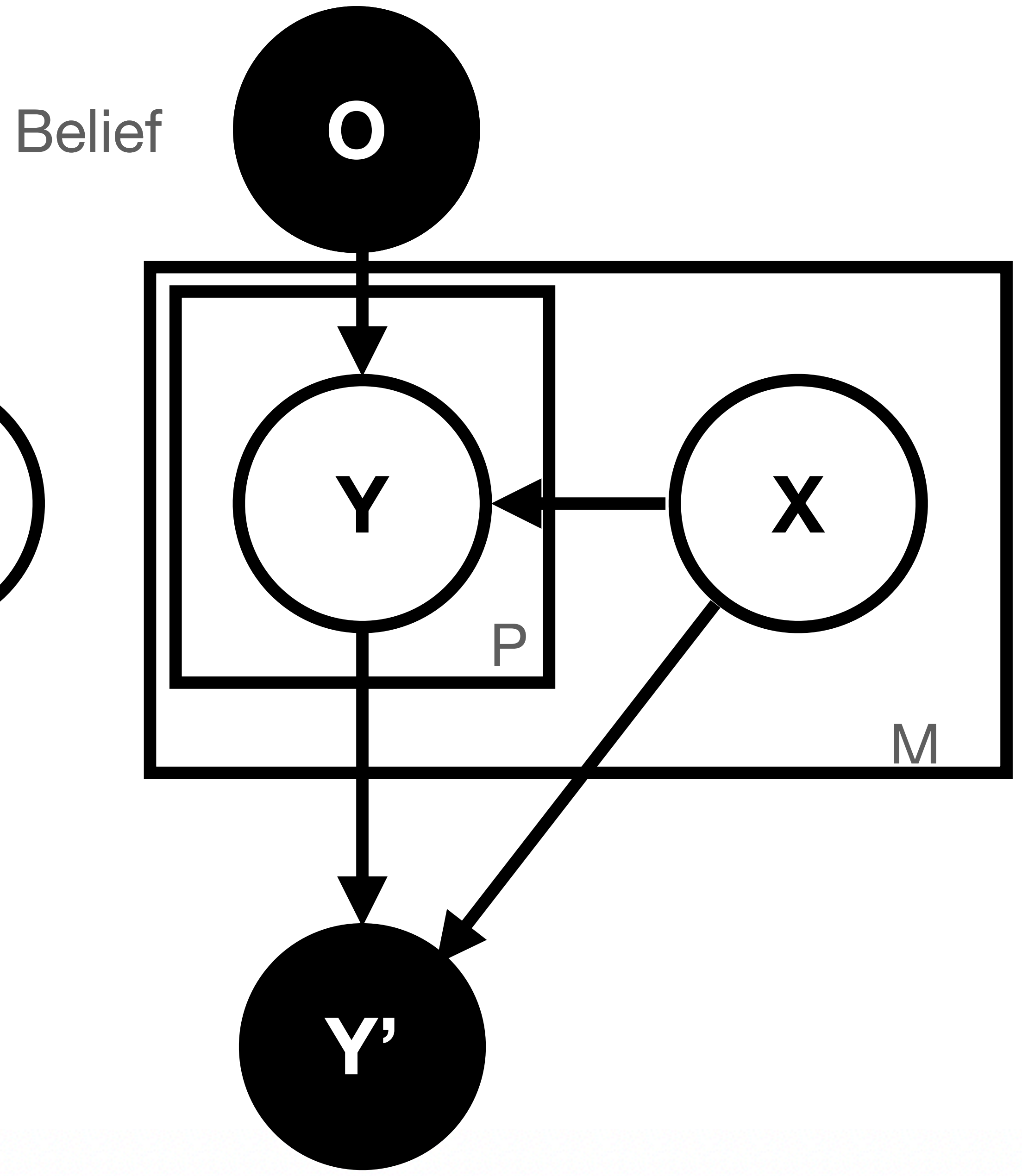
The value of <u>replicates</u>…

Specify as a causal graphical model

X has epistemic uncertainty



X has epistemic *and* aleatory uncertainty



Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

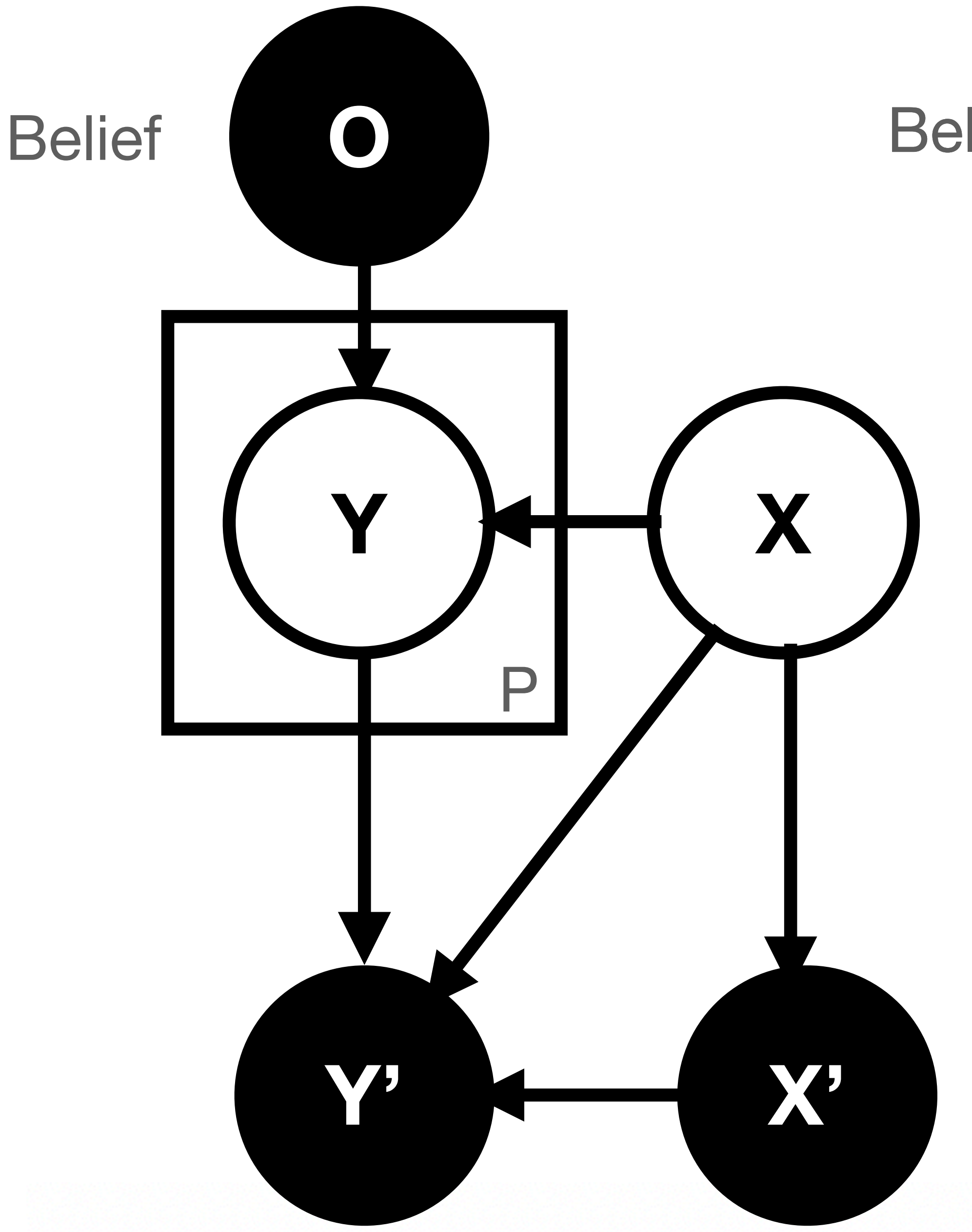
Focus efforts on replication

Eval as improvement to the science

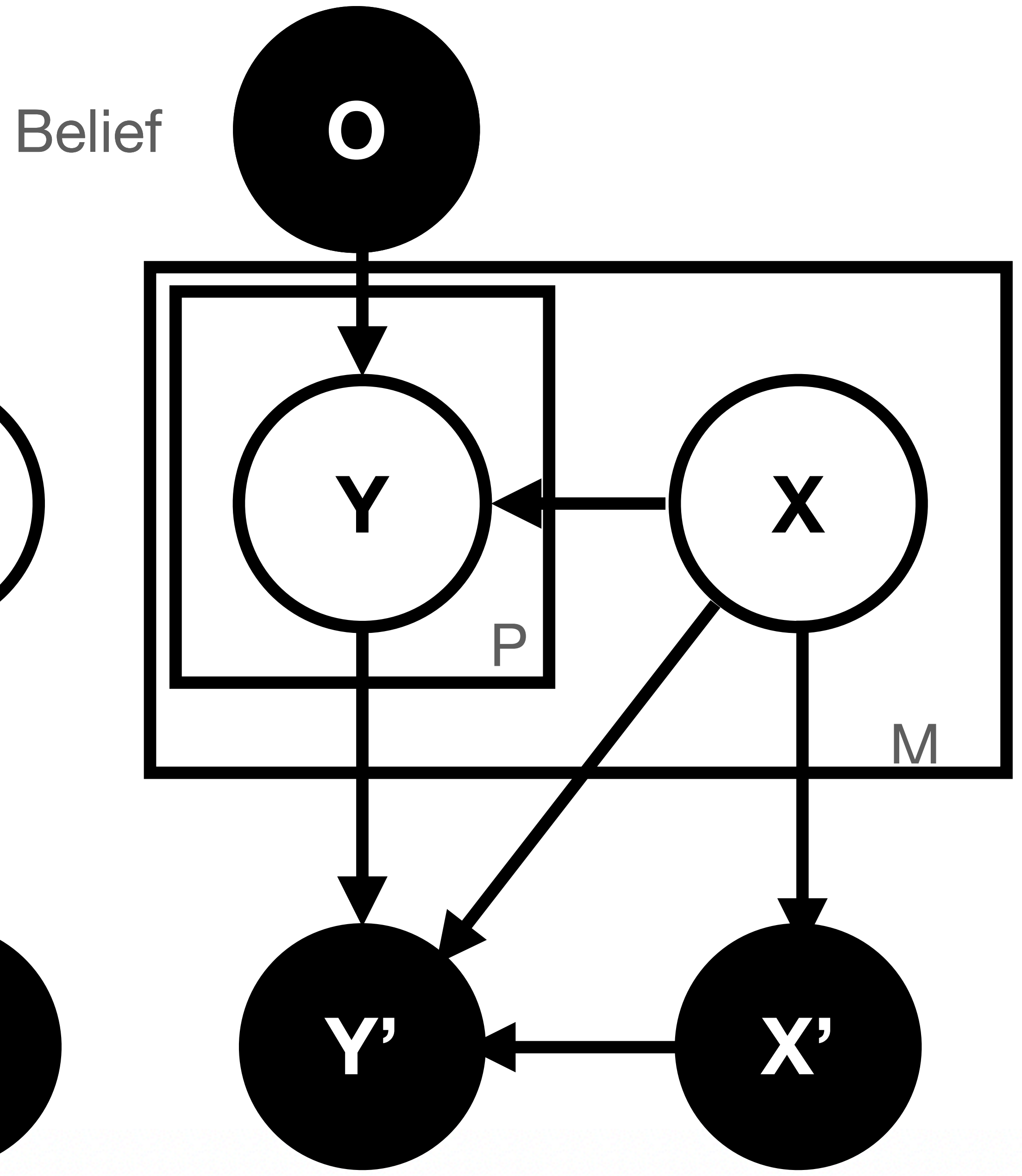
The value of <u>replicates</u>…

Specify as a causal graphical model

X has epistemic uncertainty



X has epistemic *and* aleatory uncertainty



Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

Focus efforts on replication

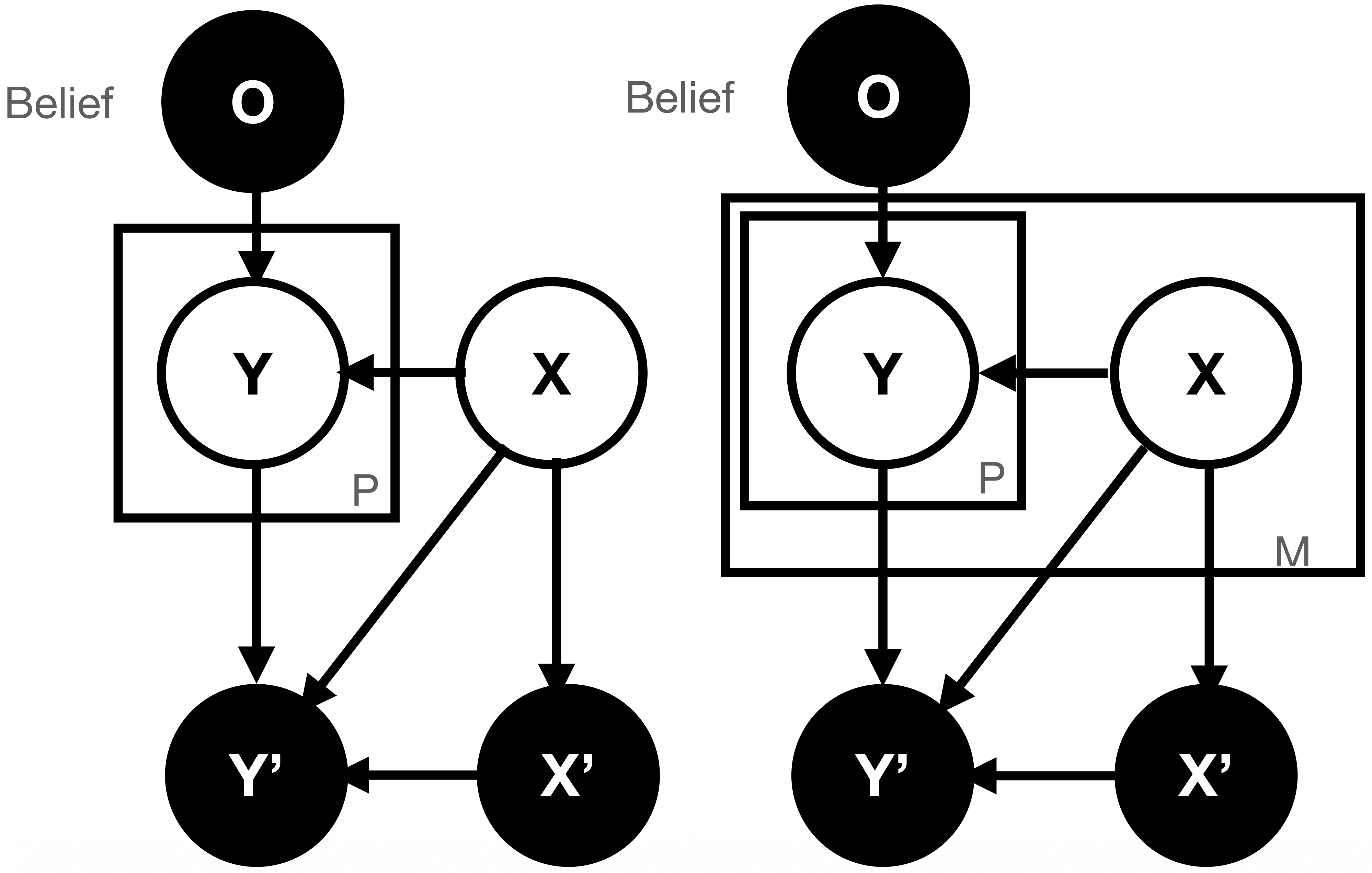
Eval as improvement to the science

The value of <u>replicates</u>…

Specify as a causal graphical model

X has epistemic uncertainty

X has epistemic *and* aleatory uncertainty



Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

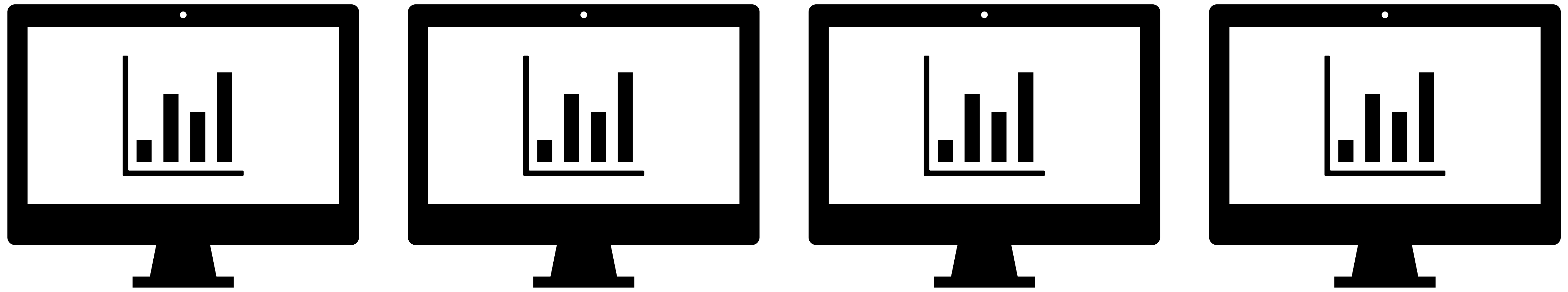
Evaluators produce replicates

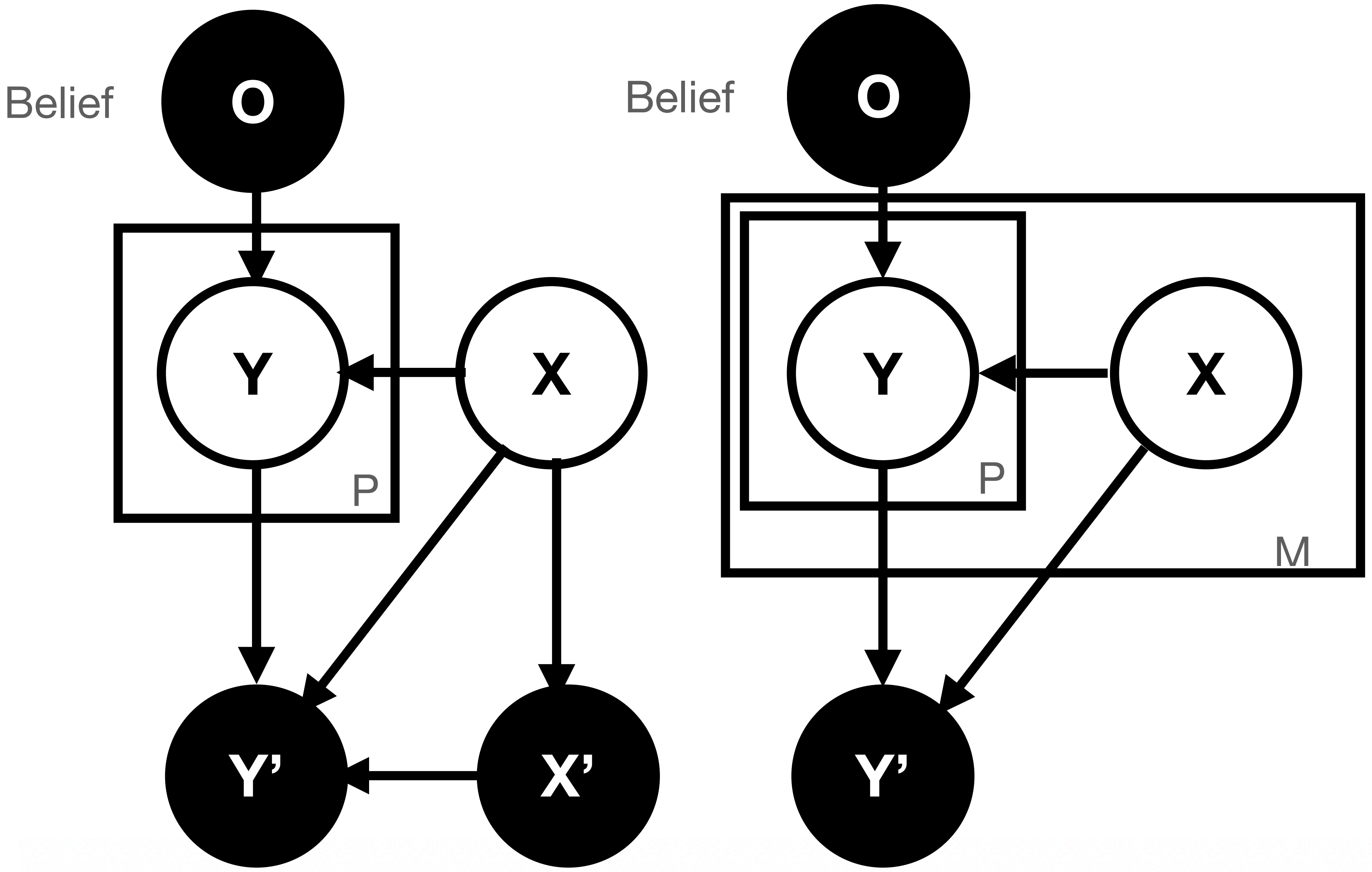
Focus efforts on replication

Eval as improvement to the science

The value of replicates…

Specify as a causal graphical model





Elements of M chosen arbitrarily (good enough)

Producing Wrong Data Without Doing Anything Obviously Wrong!

Todd Mytkowicz Amer Diwan
Department of Computer Science
University of Colorado
Boulder, CO, USA
{mytkowit,diwan}@colorado.edu

Matthias Hauswirth
Faculty of Informatics
University of Lugano
Lugano, CH
Matthias.Hauswirth@unisi.ch

Peter F. Sweeney
IBM Research
Hawthorne, NY, USA
pfs@us.ibm.com

Evaluators produce replicates

Outline

Oral History of Artifact Evaluation (student perspective)

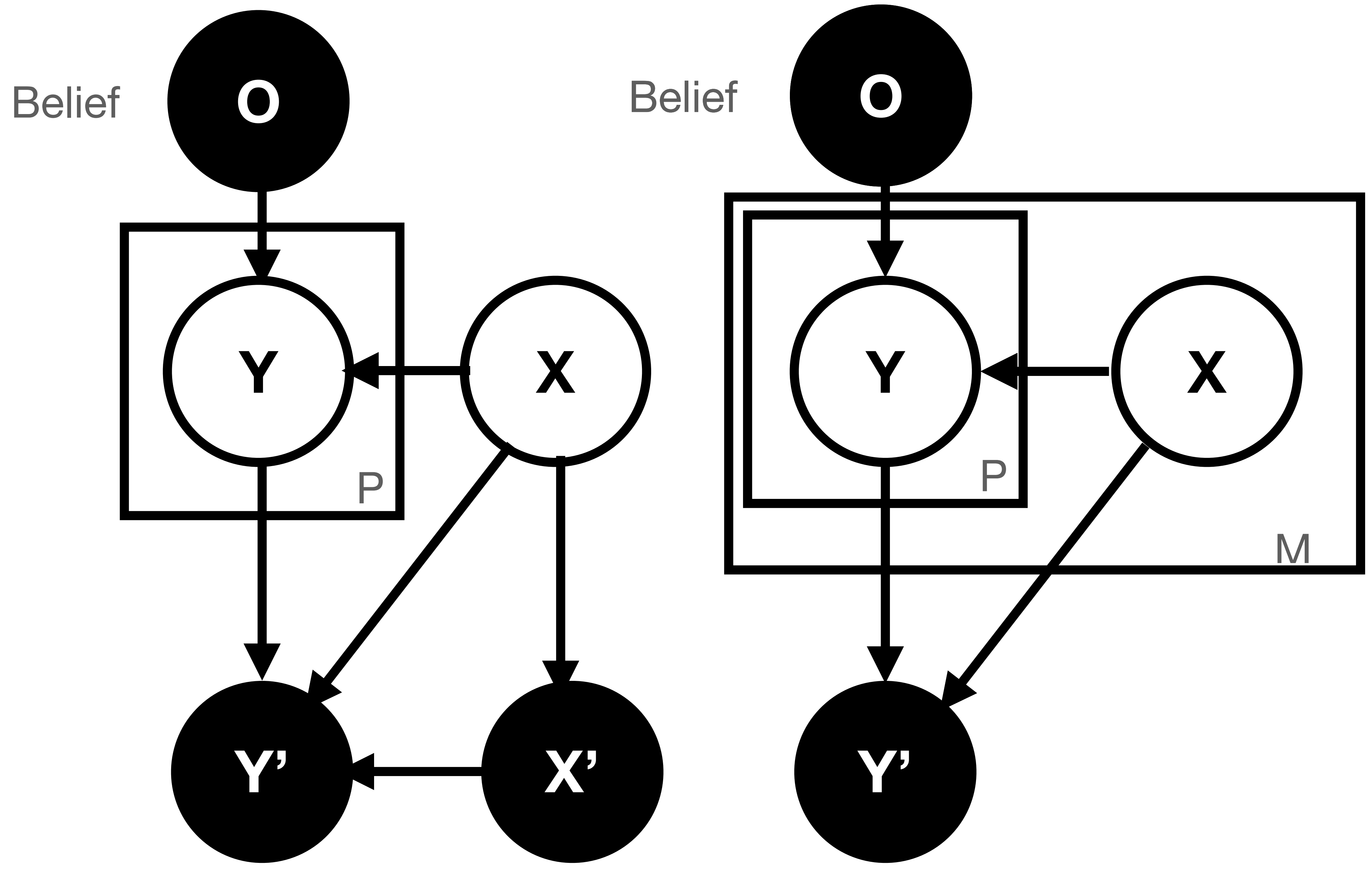
Evaluators produce replicates

Language design for reproducibility

Why this is interesting

CGMs are hard to get right

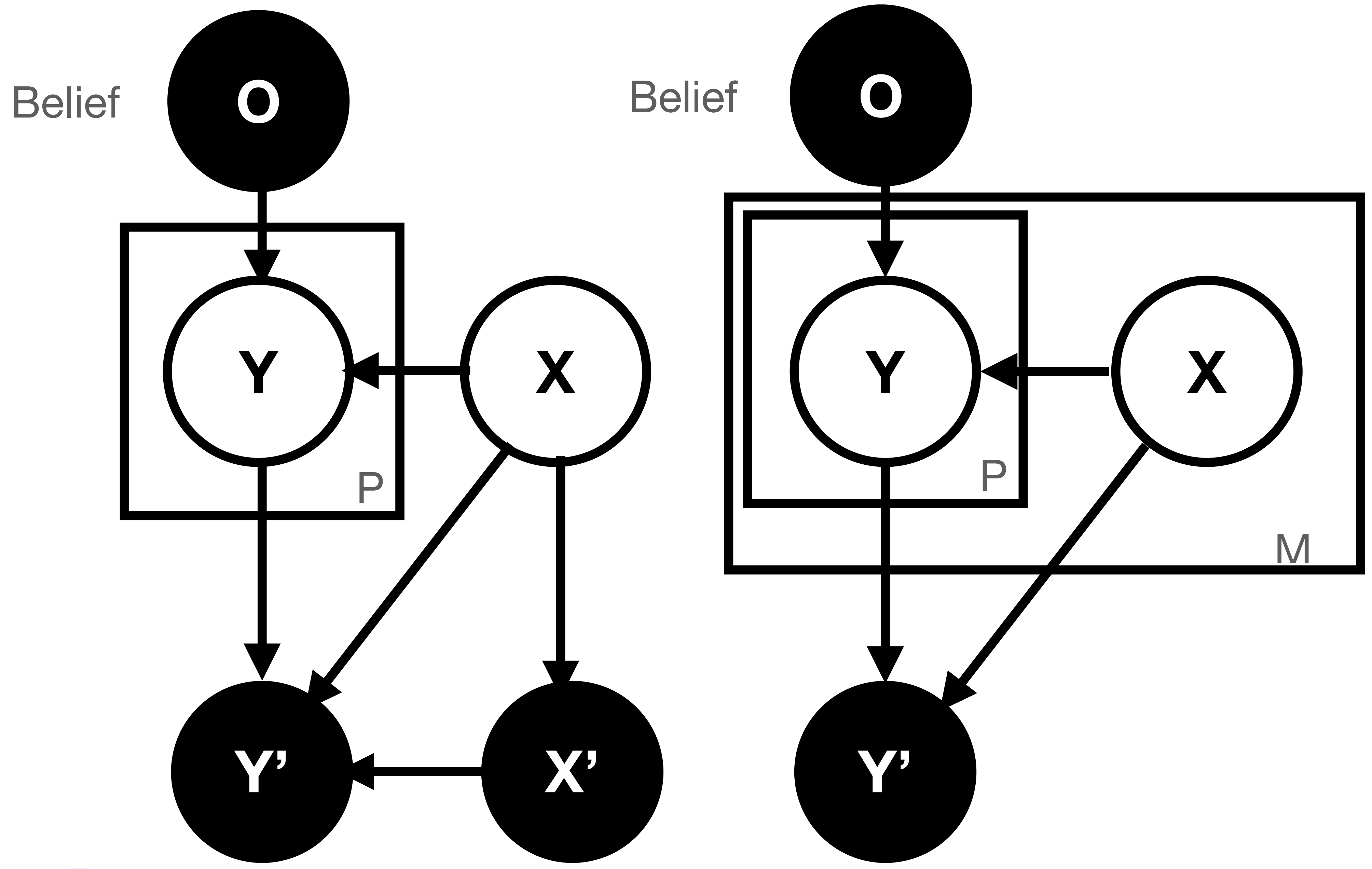
- Abuse of plate notation?
  - Y' is *not* randomly sampled
  - Should X' be a random variable?
  - Should we have a separate value for P?



Better: state assumptions in a language

Specifically, a hypothesis language

```
O : { "O2", "O3" }
Y : nat
(progid) Y <- 0
sharp (progid) assert (Y > 0)
Y_A = Y | O = "O2"
Y_B = Y | O = "O3"
(progid) assert (Y_A > Y_B)
```

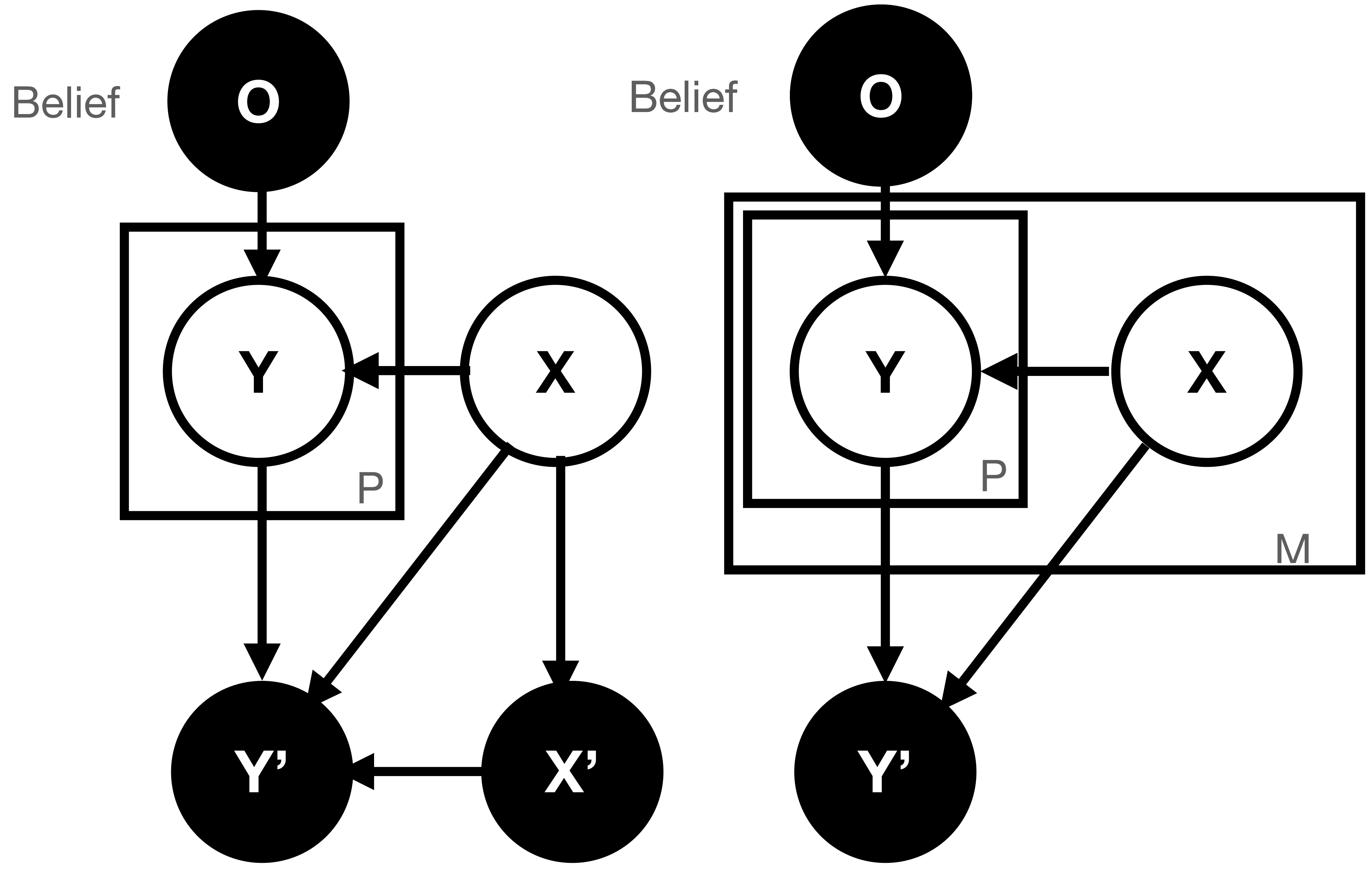



```
for trialid in repeat(progid, machineid) under complete:
    measure(Y)
```

Better: state assumptions in a language

Specifically, a hypothesis language

```
O : { "O2", "O3" }
Y : nat
E : nat
P : { "Pentium4", "Core2", "m5O3CPU" }
C : { "gcc", "intel" }
L : nat
(progid) Y <- O, L, E, C, P
Y_A = Y | O = "O2", L
Y_B = Y | O = "O3", L
(progid) assert (Y_A > Y_B)
(progid) Y_B >-> E
```



HyPL

$$
\begin{aligned}
op &::= \; = \mid > \mid < \\
coef &::= \; ? \mid n \\
sup &::= \texttt{nat} \mid \texttt{bool} \mid \{\mathbf{str}_1, \mathbf{str}_2, \ldots, \mathbf{str}_n\} \mid \texttt{real} \\
decl &::= X : \langle sup \rangle \mid X : \langle sup \rangle \; \texttt{of} \; (\mathbf{unitid}_i) \mid Y' = Y | (X_1 \; \langle op \rangle \; v_1, \ldots, X_n \; \langle op \rangle \; v_n) \\
hfn &::= \langle coef \rangle \mid \langle coef \rangle \; X \mid \langle coef \rangle \; X_1 \; X_2 \mid \langle coef \rangle \; \texttt{exp}(\langle hfn \rangle) \mid \langle hfn \rangle + \langle hfn \rangle \\
htype &::= \texttt{sharp} \; (\mathbf{unitid}_i) \mid (\mathbf{unitid}_i) \mid \texttt{belief} \\
bexp &::= \top \mid \bot \mid X \mid \; ! \, \langle bexp \rangle \mid \langle bexp \rangle \; \&\& \; \langle bexp \rangle \mid \langle bexp \rangle \; || \; \langle bexp \rangle \mid \langle hfn \rangle \; \langle op \rangle \; \langle hfn \rangle \mid \langle hyp \rangle \\
hyp &::= \underbrace{\langle htype \rangle \; Y := \langle hfn \rangle}_{SEM\ (strong\ causal)} \mid \underbrace{\langle htype \rangle \; Y <\!\!- X}_{weak\ causal} \mid \underbrace{\langle htype \rangle \; Y <\!\!- X|Z}_{weak\ causal\ conditional} \mid \underbrace{\langle htype \rangle \; Y >\!\!-\!\!> X}_{monotonic\ (associational)} \\
&\quad \mid \langle htype \rangle \; \texttt{assert} \; \langle bexp \rangle \mid \texttt{when} \; \langle bexp \rangle \; \texttt{then} \; \langle hyp+ \rangle \; \texttt{end} \\
stmt &::= \langle decl \rangle \mid \langle hyp \rangle \\
model &::= \langle stmt+ \rangle
\end{aligned}
$$

Language design for reproducibility

Why another PPL?

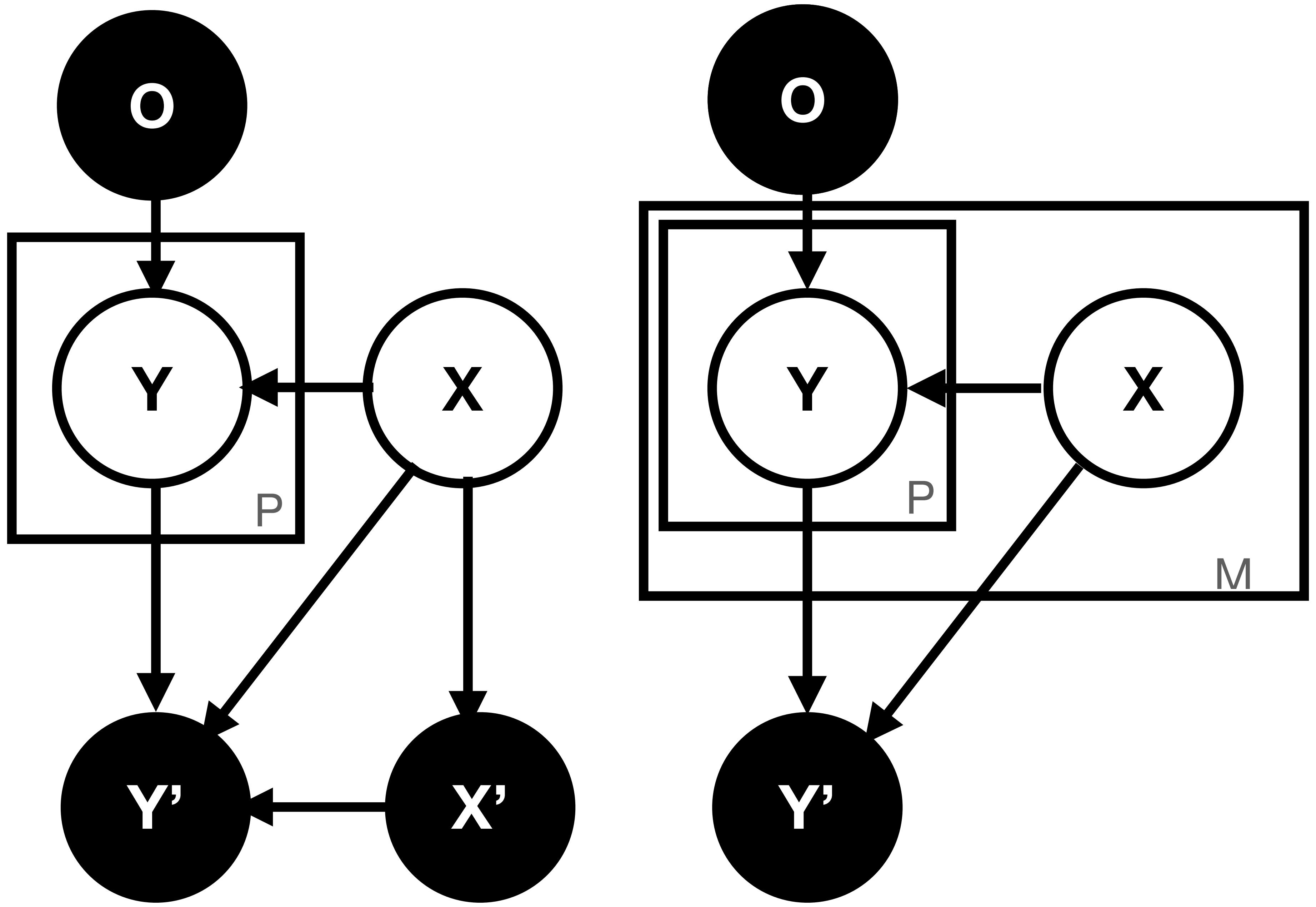
It’s *not* all about the parameters

Additional affordances via language-based approach

Enables: Structured Search

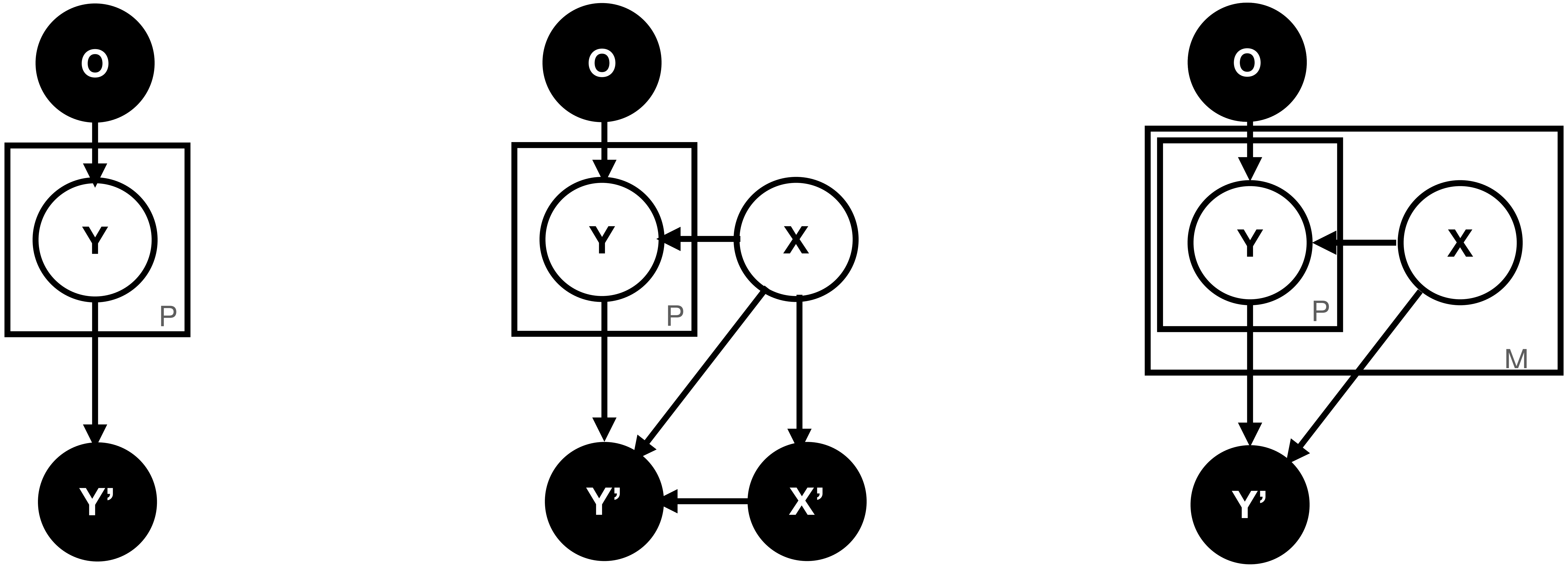
…or, search beyond keywords

```
O : { "O2", "O3" }
Y : nat
E : nat
P : { "Pentium4", "Core2", "m5O3CPU" }
C : { "gcc", "intel" }
L : nat
(progid) Y <- O, L, E, C, P
Y_A = Y | O = "O2", L
Y_B = Y | O = "O3", L
(progid) assert (Y_A > Y_B)
(progid) Y_B >-> E
```



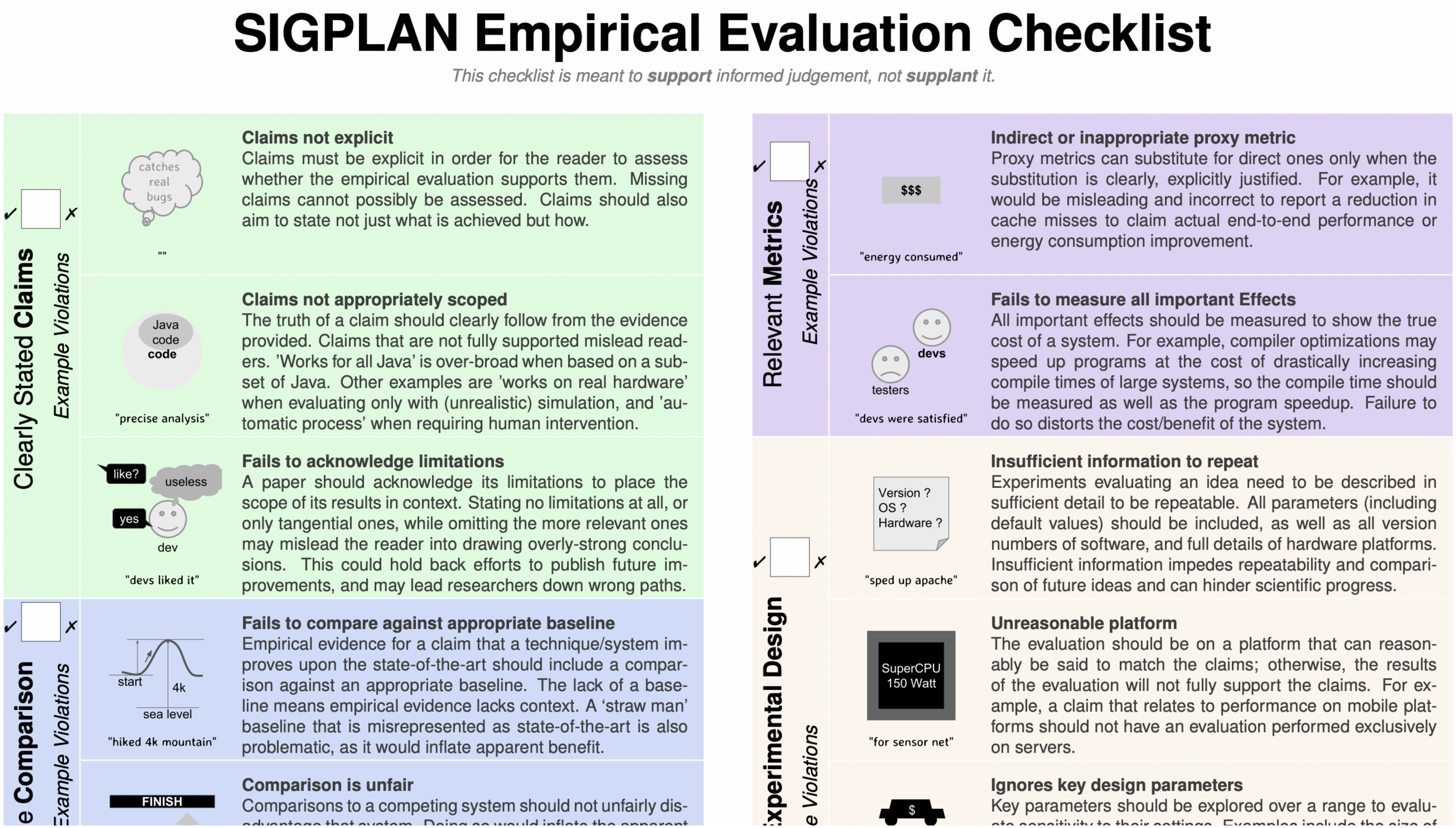
Enables: Continuous Auditing

…or, regression testing for past studies



Enables: Onboarding neophytes

Make adhering to best practices easier!



SIGPLAN Empirical Evaluation Checklist

*This checklist is meant to **support** informed judgement, not **supplant** it.*

Clearly Stated **Claims**

Example Violations

Claims not explicit
Claims must be explicit in order for the reader to assess whether the empirical evaluation supports them. Missing claims cannot possibly be assessed. Claims should also aim to state not just what is achieved but how.

Claims not appropriately scoped
The truth of a claim should clearly follow from the evidence provided. Claims that are not fully supported mislead readers. 'Works for all Java' is over-broad when based on a subset of Java. Other examples are 'works on real hardware' when evaluating only with (unrealistic) simulation, and 'automatic process' when requiring human intervention.

Fails to acknowledge limitations
A paper should acknowledge its limitations to place the scope of its results in context. Stating no limitations at all, or only tangential ones, while omitting the more relevant ones may mislead the reader into drawing overly-strong conclusions. This could hold back efforts to publish future improvements, and may lead researchers down wrong paths.

e **Comparison**

Example Violations

Fails to compare against appropriate baseline
Empirical evidence for a claim that a technique/system improves upon the state-of-the-art should include a comparison against an appropriate baseline. The lack of a baseline means empirical evidence lacks context. A 'straw man' baseline that is misrepresented as state-of-the-art is also problematic, as it would inflate apparent benefit.

Comparison is unfair
Comparisons to a competing system should not unfairly dis-

Relevant **Metrics**

Example Violations

Indirect or inappropriate proxy metric
Proxy metrics can substitute for direct ones only when the substitution is clearly, explicitly justified. For example, it would be misleading and incorrect to report a reduction in cache misses to claim actual end-to-end performance or energy consumption improvement.

Fails to measure all important Effects
All important effects should be measured to show the true cost of a system. For example, compiler optimizations may speed up programs at the cost of drastically increasing compile times of large systems, so the compile time should be measured as well as the program speedup. Failure to do so distorts the cost/benefit of the system.

xperimental **Design**

e Violations

Insufficient information to repeat
Experiments evaluating an idea need to be described in sufficient detail to be repeatable. All parameters (including default values) should be included, as well as all version numbers of software, and full details of hardware platforms. Insufficient information impedes repeatability and comparison of future ideas and can hinder scientific progress.

Unreasonable platform
The evaluation should be on a platform that can reasonably be said to match the claims; otherwise, the results of the evaluation will not fully support the claims. For example, a claim that relates to performance on mobile platforms should not have an evaluation performed exclusively on servers.

Ignores key design parameters
Key parameters should be explored over a range to evalu-

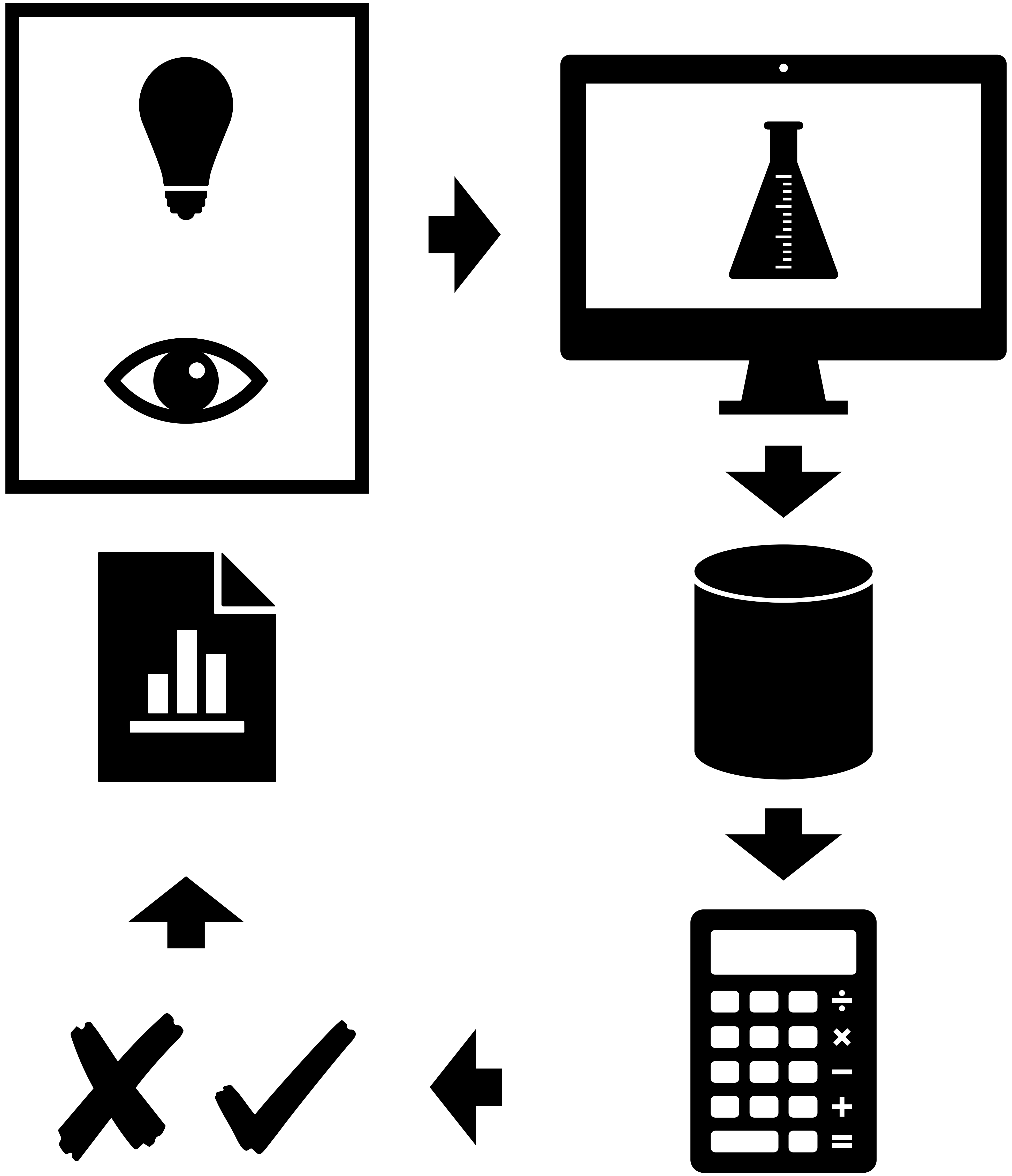
Challenges in application to cybersecurity

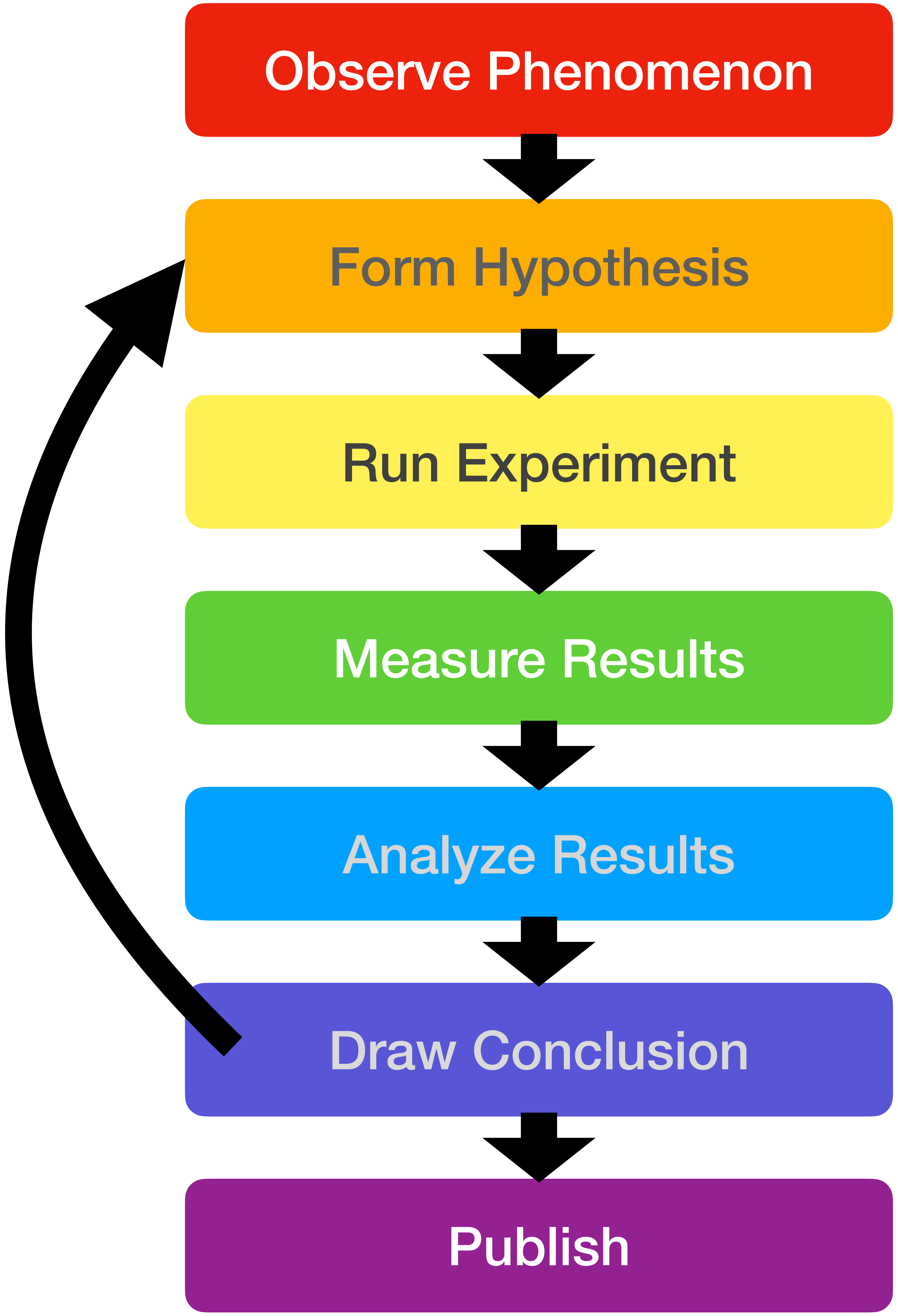
Extreme values

Interested in maxima or the long tail?

Need different methods!

Extreme values & Non-scientific knowledge





Not an end, but hopefully a