You Only Ever Look Once

Experiments in…
Fine Tuning Domain-Specific Code Generating LLMs

Matt Robinson; Yao, Cai, Stone, Weiss
Data Science/Applied Statistics
University of California Berkeley ‘25

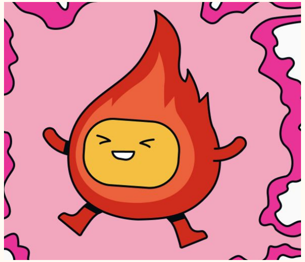
Bolts

1. **Why fine tune ⚡**
2. **How ⚡**
3. **Why experiments are more important than ever for our industry ⚡**

DOMAIN SPECIFIC

C++ Patterns

RAII, Concepts, Templates



Mojo/MAX

GPU programming

>= CUDA

Kubernetes

Dynamic clusters

Autoscalers

When it isn't simply

Secure Envs

Egress is NOT an option

BUT LLMS ARE ...?

- Perfetc enough
- LMBenchmark number go up
- Is already working
- Depends on what your definition of *is* is
- How to define what *is* is?

CUE: DESIGN THINKING

Interviewer: How’s it going?
Interviewee: Meh.

“I recently told my team they don’t have to use LLMs anymore, the time we spent reviewing generated code became too costly”

Engineering Manager
Series B InsurTech
Startup

“Cursor [an AI development environment] makes up documentation that does not even exist”

Senior Engineer
Major FinTech

“The GPT5 update broke all of my system prompts and I have to fall back to earlier and earlier versions to keep my workflows running”

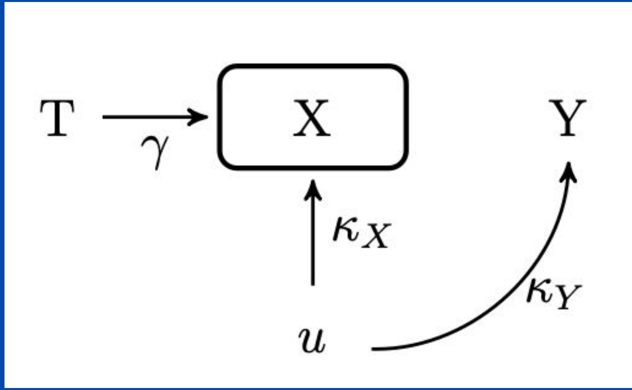
Engineering Manager
Not Google

“[Our] internal code agent is 90% correct. [I use] it [only] for writing documents because I am a bad writer. My domain knowledge is still absolutely necessary”

Senior Quant
Relentless.com

REACH FOR TOOLS TO...

Avoid benchmark overfitting ↓



Formal Hypothesis

Let

$$P_{default}$$

represent the performance of default coding LLMs and

$$P_{finetuned}$$

represent the performance of fine-tuned coding LLMs on language-specific tasks, where performance is measured as a score of execution validity, operational validity, and troubleshooting utility.

Null Hypothesis (H):

$$H_0 : P_{default} = P_{finetuned}$$

Alternative Hypothesis (H):

$$H_a : P_{default} \neq P_{finetuned}$$

Performance Evaluation Framework

Performance can be operationalized as a weighted score:

$$P = w_1 \cdot E_v + w_2 \cdot O_v + w_3 \cdot T_u$$

Where: -

$$E_v$$

= Execution validity (binary: 0 or 1) -

$$O_v$$

= Operational validity (binary: 0 or 1) -

$$T_u$$

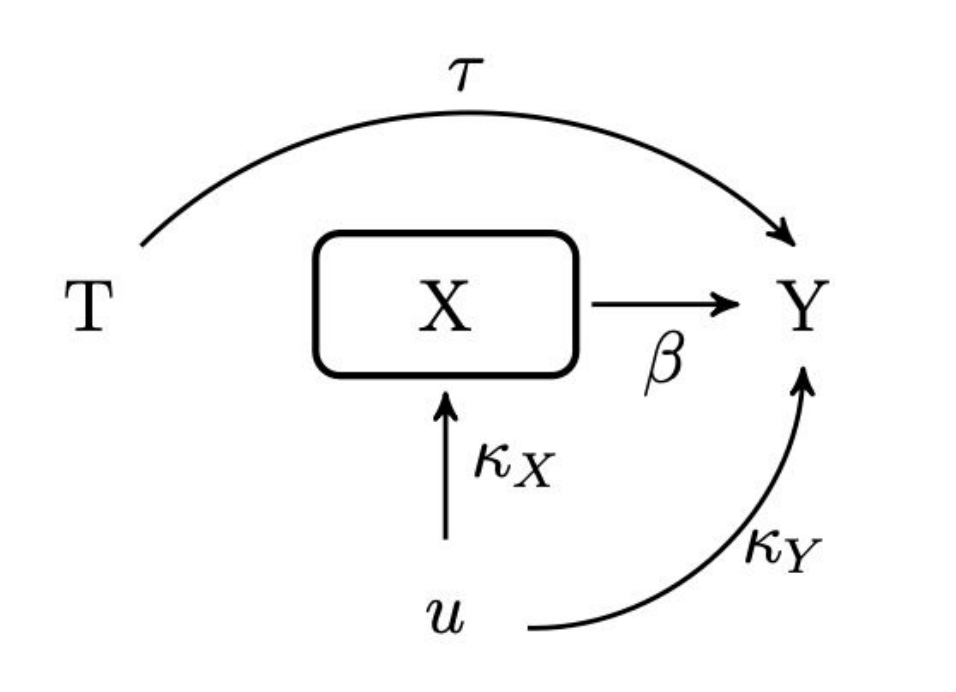
= Troubleshooting utility (binary: 0 or 1) -

$$w_1, w_2, w_3$$

= weights such that

$$w_1 + w_2 + w_3 = 1$$

EXPERIMENTS GOOD



Does X cause Y?
How much X causes Y? If at all.

Attributes of Proper Experiment:

- Treatment and Control
- Regression
- Likelihood Estimator
- Notion of Odds

probably…

RESULTS SNAPSHOT

stdout	return_code	flag	flag_detail

{
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"annotations": {
"kubectl.kubernetes.io/last-applied-
	0	false	Resource fulfills the task: it is a Namespa with only the requested simple organizati labels and no extra user-defined element

{
"apiVersion": "v1",
"kind": "Namespace",
"metadata": {
"annotations": {
"kubectl.kubernetes.io/last-applied-
	0	true	The provided Namespace resource lacks ResourceQuota that would enforce CPU a memory constraints, thereby failing the e task requirement.

FIN

You Only Ever Look Once

Fine Tuning Domain-Specific Code Generating LLMs

Matt Robinson, Yao, Cai, Stone, Weiss
Data Science/Applied Statistics
University of California Berkeley ‘25