

AI as an Organizational Problem

Prasanna Tambe

Wharton School, U. Penn

tambe@wharton.upenn.edu

With Sharique Hasan (Duke) and David Hsu (Wharton)

LLM workflows require complementary investments

- Data pipelines
- Prompt/agent Skills
- Workflow redesign

Complements are interdependent; Returns are likely supermodular

Firms face a rugged configuration space—small adjustments can help or hurt depending on how other components are configured.

The Underlying, Foundational Technology Is Unstable / Rapidly Evolving

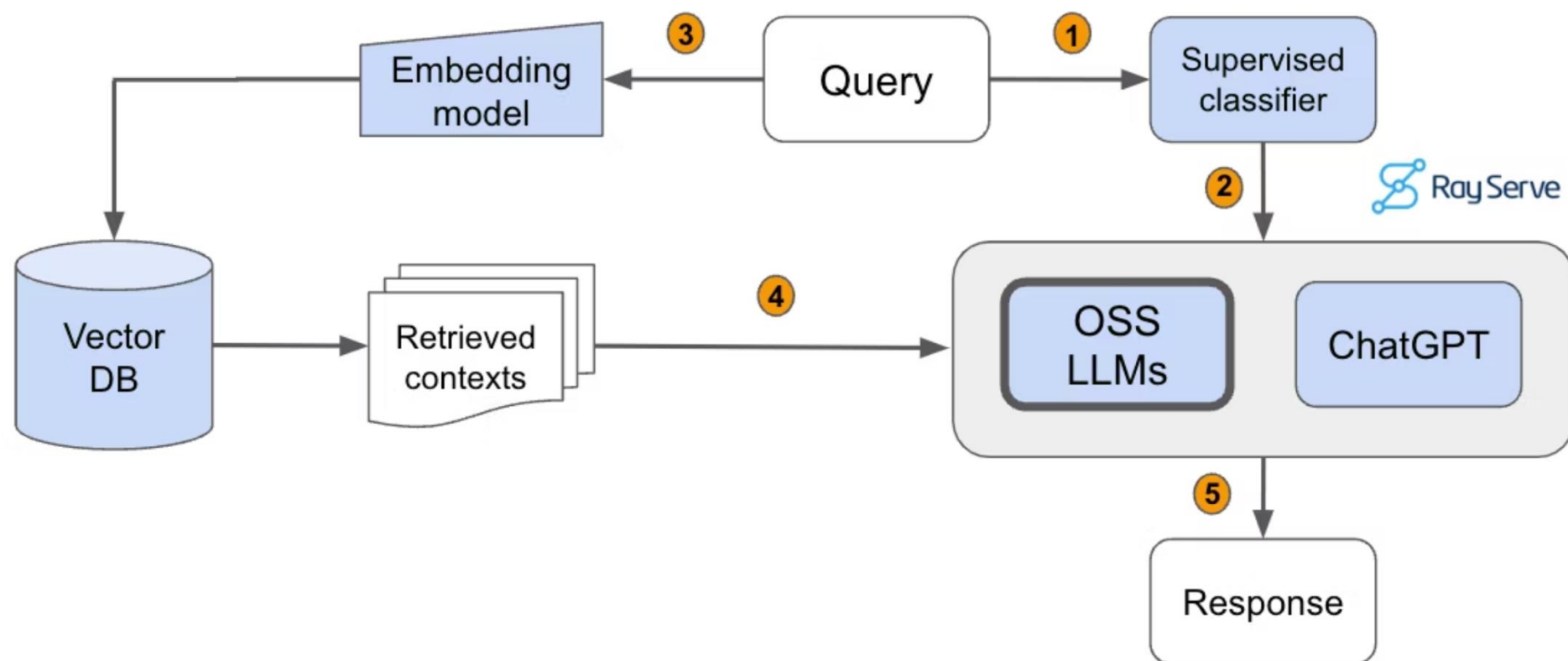
- new model releases
- shifting cost/performance ratios
- changes in guardrails, context length, memory, evals
- evolving integration stacks

This means firms must search across configurations, not just adopt a technology

Because configuration space is rugged:

- Firms are experimenting with pilots and prototypes
- Firms could get trapped on suboptimal peaks
- Escaping requires coordinated "long jumps" (bundled changes)

Consider an established LLM workflow. What are the costs of choosing a new provider?



Model "upgrades" have
already been associated
with workflow disruption


**GPT-5 Disrupted my entire workflow, and
here's what I aggressively switched to.**



Heatmap for LLM usage

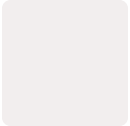


This "fragility" may pose a problem for investment in LLM-based workflows



Processes built on unstable foundations fail

If you redesign your patent evaluation workflow around a model that gives different answers each run, the entire process inherits that instability



Organizational change is expensive to reverse

Hiring, training, software purchases, and workflow changes are sunk costs; validating the AI before restructuring avoids costly rollbacks



Dependencies compound fragility

Downstream decisions (licensing negotiations, portfolio strategy, resource allocation) that depend on AI scores amplify any unreliability in the original evaluation

Research question:

How stable are LLM-based workflows when firms operate in a rugged, evolving configuration space?

Existing literatures from which we draw (and where we hope to contribute)

Extend the traditional IT-productivity literature on complements (landscapes there are viewed as static).

Organizational complementarity models: supermodular, but not dynamic or search-based.

AI-capability literature: focuses on capability effects, not configurational interdependence.

Our main insight:

LLMs create an "NK-style" problem

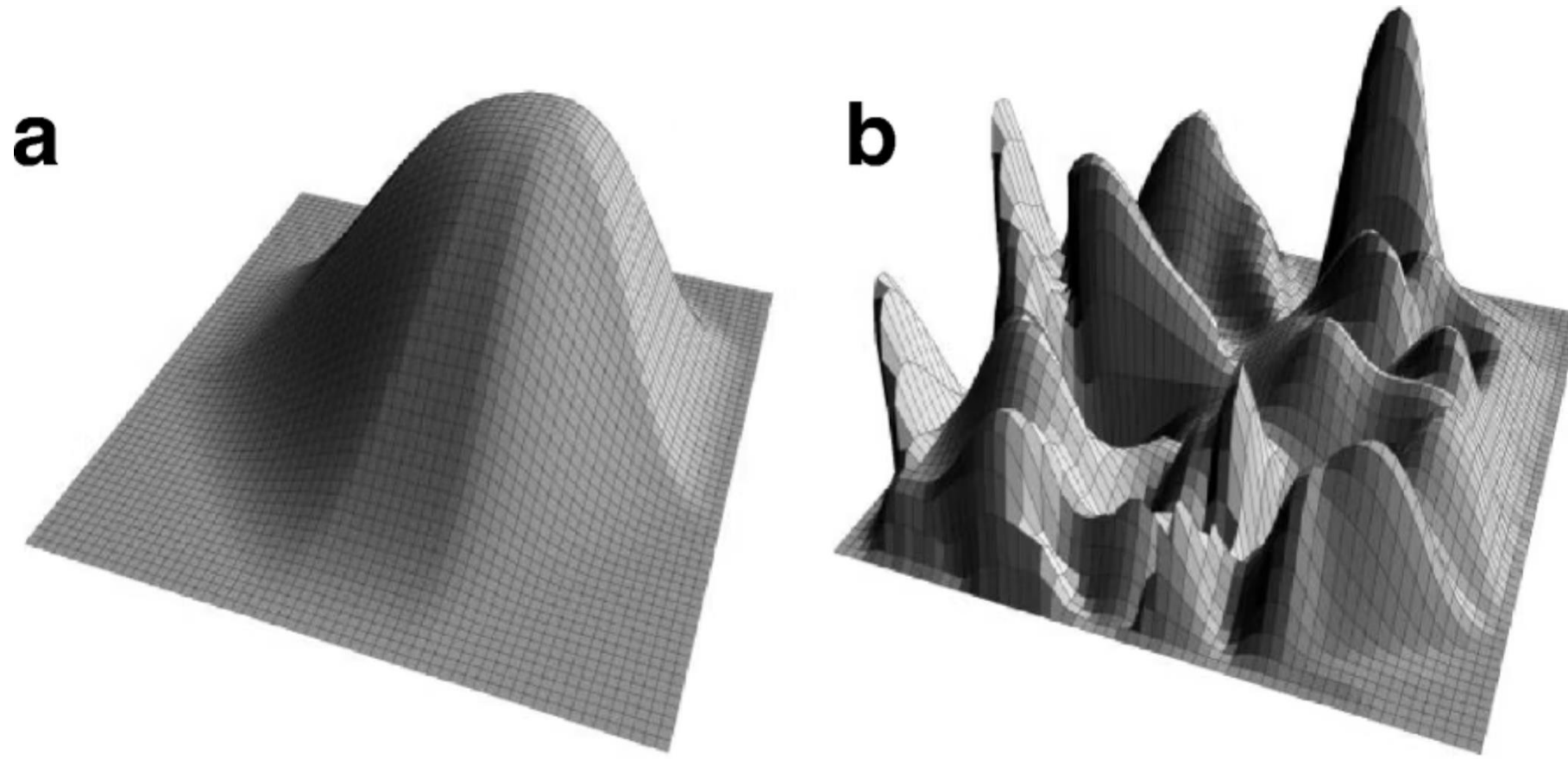
Traditional IT Investments

Architecture stays fixed, complementarities are stable, optimization is incremental

With LLMs

Managers face high interdependence (data, prompts, workflows, and human skills) *and* a moving landscape (model behavior shifts frequently)

Main point: LLM investments are rugged landscapes characterized by multiple, narrow peaks



Low K surface

High K surface

A rugged landscape for LLM investment means:

→ A model + complements is equivalent to climbing to a particular point on the fitness landscape

→ High accuracy on one run could mean you found a tall but precarious peak; one perturbation and you fall off

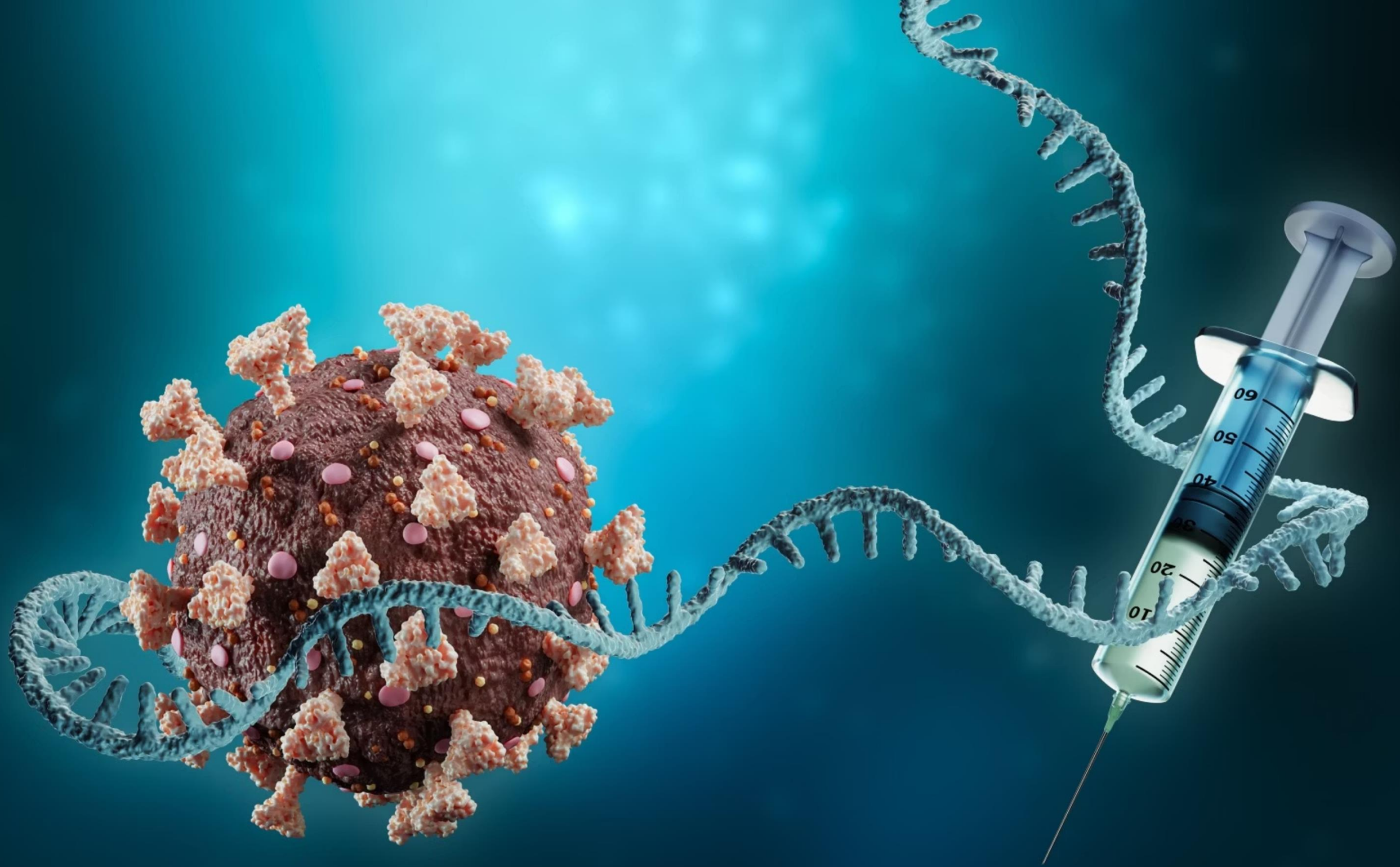
→ **The more tightly you couple your process to a particular model's characteristics, the more vulnerable you are when that model changes, degrades, or gets deprecated**

Our empirical approach

TASK: predict commercial value of university innovation

Data on university commercialization

This is a high-stakes, real-world task requiring complex reasoning using scientific and contextual information, with clear ground truth (licensing, deals, revenue), allowing objective evaluation of both accuracy and reliability. The data are proprietary (no public web contamination) so models cannot rely on web-trained priors (predictive power relies on reasoning and generalization).

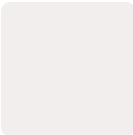


Empirical approach



Run each model twice on the same set of innovations

Same 300 innovations, same prompts, same conditions; only difference is the run itself



Compare model scores against licensing outcomes

Calculate AUC using ground truth (which patents were actually licensed vs. not)



Correlate Run 1 and Run 2 scores

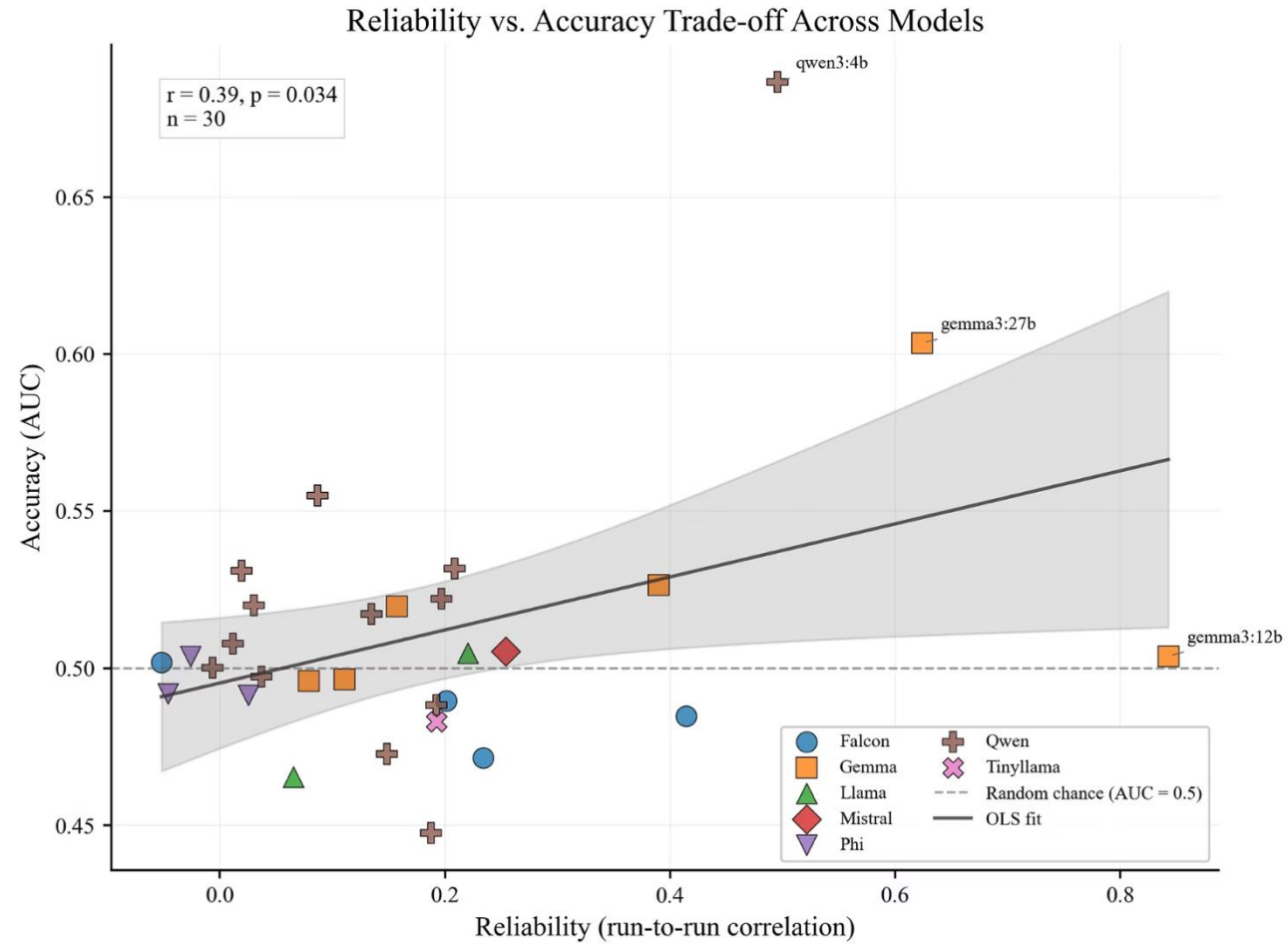
High correlation means the model gives consistent answers



Repeat across models

Different architectures, sizes, and providers to see which characteristics predict reliability and accuracy

We find that models trade off reliability and accuracy



Moving away from a fixed point (e.g., original model choice) drives decay in reliability

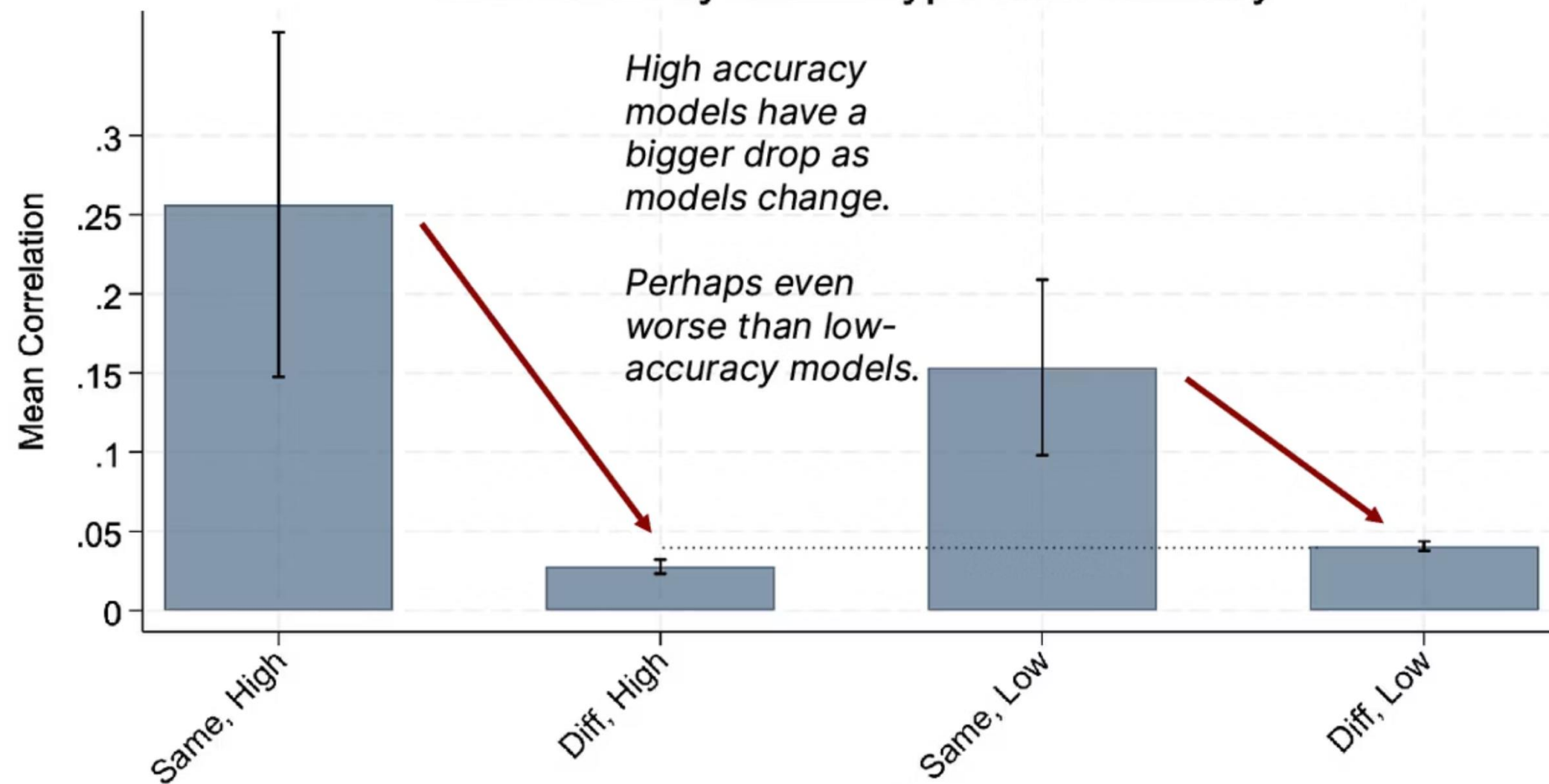
Especially when the origin is highly accurate.

Table 2: Reliability of AI Model Choices

	(1)	(2)	(3)	(4)	(5)	(6)
Diff. Model	-0.144*** (0.032)	-0.144*** (0.032)	-0.131*** (0.031)	-0.142*** (0.033)	-0.142*** (0.033)	-0.128*** (0.031)
Orig. Accuracy		-0.025 (0.058)	0.745* (0.424)		-0.022 (0.049)	0.789** (0.334)
Diff Model x Origin Accuracy			-0.783** (0.373)			-0.824** (0.321)
Constant	0.181*** (0.037)	0.181*** (0.037)	0.169*** (0.037)	0.179*** (0.032)	0.179*** (0.032)	0.166*** (0.030)
Origin Model FE	No	No	No	Yes	Yes	Yes
Destination Model FE	No	No	No	Yes	Yes	Yes
Observations	3,660	3,660	3,660	3,660	3,660	3,660

Standard errors in parentheses
Row and Column model fixed effects included where indicated. Standard errors clustered by row and column model.
* $p < 0.10$, ** $p < 0.05$, *** $p < 0.01$

Correlation by Model Type and Accuracy



Our regression shows high correlation within the same model across runs, but switching models drops correlation, particularly for high-accuracy models (~90% accurate), where correlation falls to zero.

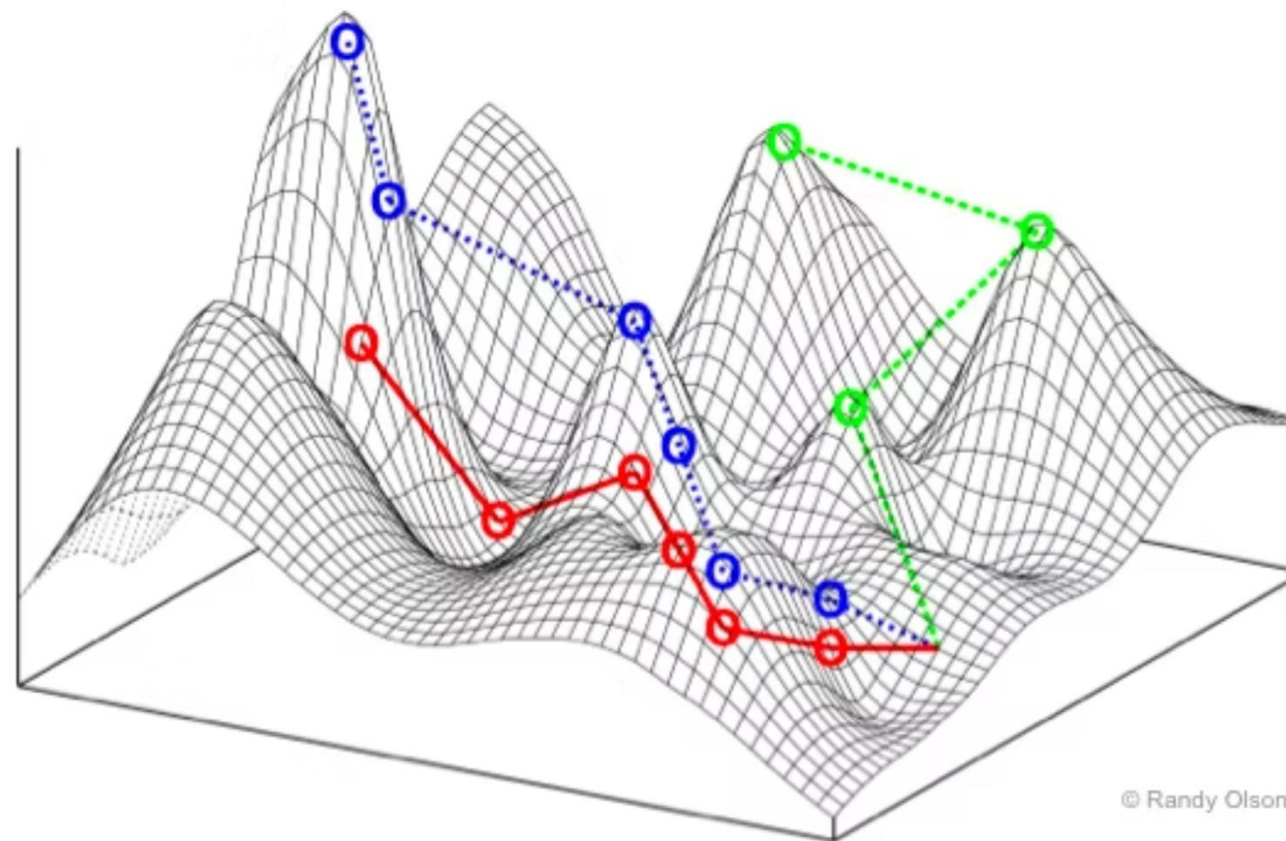
But even low-accuracy models (~10% accurate) show low correlations when switched.

This suggests models each occupy a distinct peak on a rugged fitness landscape, an effect most dramatic in high-performing models.

This ruggedness could affect model updating choices

Incremental change
= stay near **local optima**

Radical change
= may make you **worse off**



Firms' empirical outcomes could be heterogeneous

We should expect to observe:

- wide dispersion in performance outcomes
- early adopters with highly varied results
- complementarities explaining both big wins and notable failures

For managers:



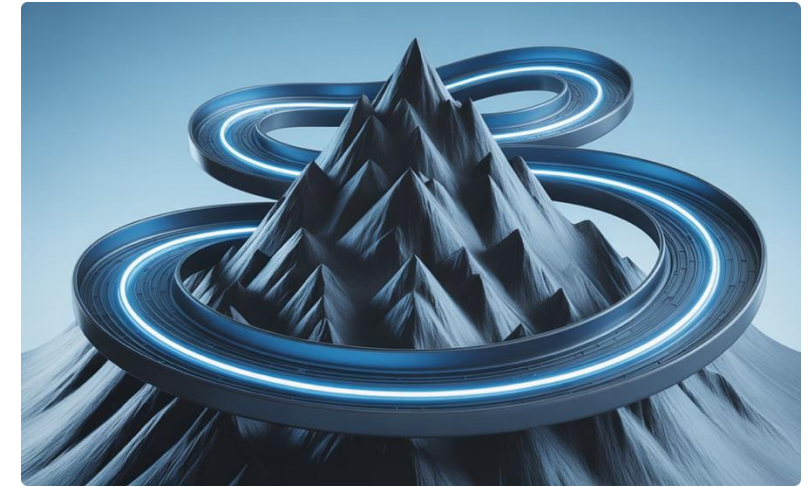
Minimize Coupling

Design decoupled elements such as standardized data formats, consistent prompt templates, and modular workflow steps to reduce complementarities.



Implement Buffer Layers

Introduce intermediate layers, retrieval modules, or supervised classifiers to absorb model volatility and stabilize outputs despite underlying changes.



Optimize for Adaptability

Prioritize cheaper re-configurations when models, guardrails, or context windows shift, treating this flexibility as an ongoing capability, not a one-off.

Thank you.
Comments
appreciated!

Prasanna (Sonny) Tambe.

tambe@wharton.upenn.edu.