6.S978 AI ENGINEERING

LECTURE 7

Evaluating AI Systems

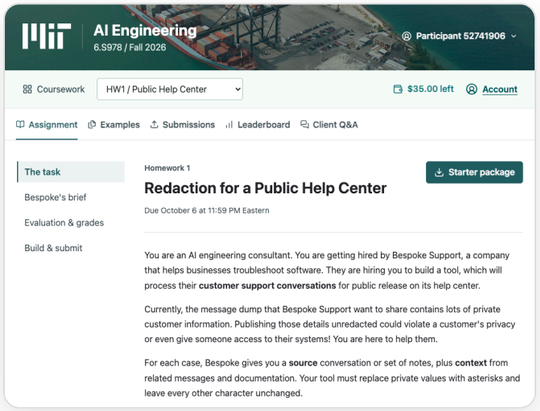
Omar Khattab

Engineering AI Systems and Agents

IN THE LECTURES TO DATE, WE HAVE BEEN STUDYING:

How can we **engineer intelligent systems** that are **reliable**, **efficient**, and **maintainable**?

We learned a lot about **developing** such systems, and guaranteeing *structural* properties. But… once we build one, **how do we know whether its empirical behavior "works"?**





data is part of the specification!

Principles
stays

Lecture 2
Requirements

Lecture 3
Declarative programming

Lecture 7
Evaluation
today

learning needs it

	Inference algorithms, or harnesses		Learning algorithms, or compilers	
Systems absorbs the change	**Lecture 4** Agent Design	**Lecture 6** Retrieval and Reasoning over Massive Contexts	**Lecture 8** Downstream Learning for AI Systems	**Lecture 9** Downstream Learning in Weights

Models
improves fast

Lecture 1
Language models


GPT-3, 2020


Llama, 2023



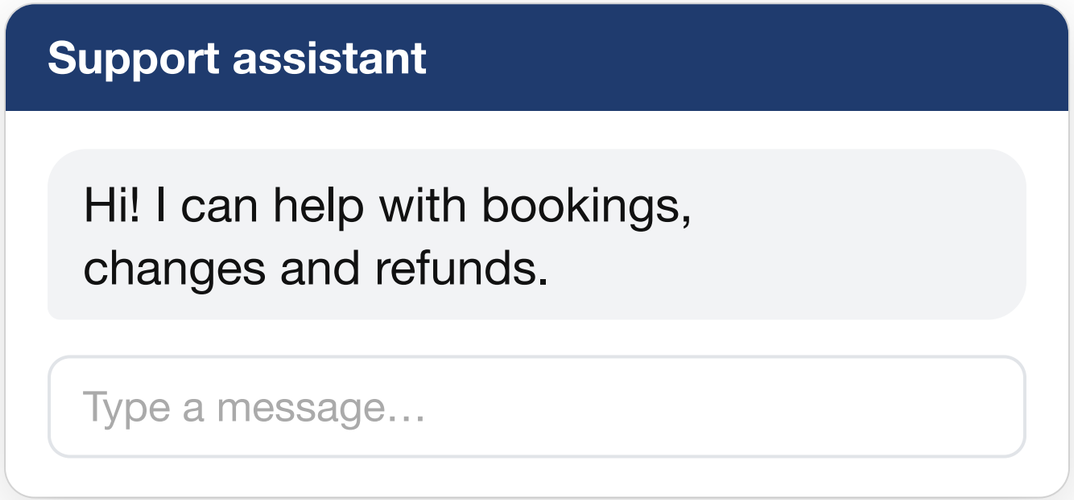
Qwen 2.5, 2024


Fable 5.1, 2026


GPT-6, 2026

?
next

Suppose an airline hires us to build its support assistant.



Our assistant's signature

```
class SupportAssistant(Signature):
    """Help the airline's customers."""
    conversation: list[Turn] = InputField()
    reply: str = OutputField()
```

Tools, from the airline's systems

```
def get_booking(code): …
def search_flights(route, date): …
def change_booking(code, flight): …
def issue_refund(code, amount): …
def escalate_to_person(): …
```

Requirements

at least those communicated upfront by the airline!

- Rebook customers on cancelled flights.
- Never promise a forbidden refund.
- Escalate to a person when asked.
- Be kind to someone grieving.

What could go wrong?

The Washington Post

Air Canada chatbot promised a discount. Now the airline has to pay it.

CBC News

Air Canada found liable for chatbot's bad advice on plane tickets

Ars Technica

Air Canada must honor refund policy invented by airline's chatbot

BBC

Airline held liable for its chatbot giving passenger bad advice – what this means for travellers

Forbes

What Air Canada Lost In 'Remarkable' Lying AI Chatbot Case

Air Canada Chat with us

…have already travelled and would like to submit your ticket for a reduced bereavement rate, kindly do so within **90 days** of the date your ticket was issued…

Air Canada's chatbot, November 2022, the month ChatGPT came out



Moffatt v. Air Canada

British Columbia's tribunal, February 2024

"It should be obvious to Air Canada that it is responsible for all the information on its website."

Air Canada must pay CA$812.02

Moffatt v. Air Canada, 2024 BCCRT 149. Headlines: February 2024. Gavel: Blogtrepreneur, CC BY 2.0.

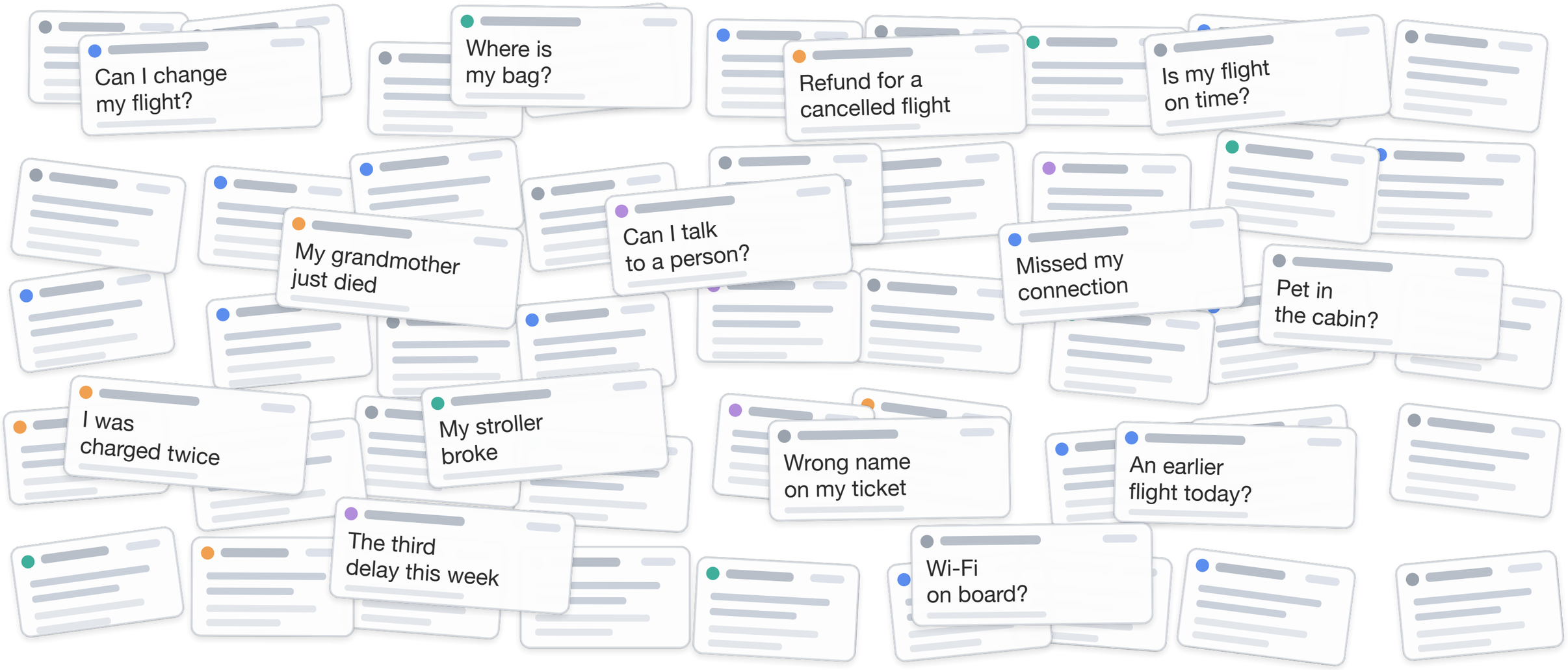
Today

1. Evaluation in an ideal world
2. What if evaluation is expensive?
3. What if there are no users, and no answer key?
4. How much evidence is enough? **Next week**

Part 1

Evaluation in an ideal world

Suppose we had lots of logged conversations and we could see all of them.



What actually is inside each conversation?

Conversation 48213 Tuesday, 9:14 pm Priya Raman, Silver our assistant

My 6:05 pm flight to San Francisco was cancelled.
Can you get me on the next flight tonight?

Find the booking, then seats on tonight's flights to San Francisco.

```
get_booking("QK7Z2P") → ORD→SFO 6:05 pm, cancelled
search_flights("ORD", "SFO", today) → 8:40 pm, 3 seats
change_booking("QK7Z2P", "8:40 pm") → confirmed
```

You're on the 8:40 pm flight, seat 14C.
Your new boarding pass is in the app.

38 s **2,940** tokens not escalated "Was this helpful?" **Yes**

← when, and who

← what they asked

← what the system did

← and its visible output

← other metadata/stats

x **Priya Raman**, Silver
“My 6:05 pm flight to San Francisco was cancelled.
Can you get me on the next flight tonight?”

τ `get_booking, search_flights, change_booking`
“You’re on the 8:40 pm flight, seat 14C.”

$x \sim \mathcal{D}$

a request, drawn from all requests

S

our assistant

$\tau \sim S(x)$

a rollout: a trajectory it might produce

$r(x, \tau)$

a judgment of that trajectory

Suppose we had a way to perfectly decide how good every conversation was.

What should an evaluation experiment tell us?
What are we trying to achieve with it?



Is it good enough to deploy?

A ? B

Is the new version better?



Which of many is best?

Refund promises	6
Loops	200
Rebookings	412

What goes wrong, and how often?

10	10	9	10	10	8	10	9	10	0
10	10	9	10	8	10	10	9	10	10

$$Q(S) = \mathbb{E}_{x \sim \mathcal{D},\, \tau \sim S(x)} \left[r(x, \tau) \right] = 9.1$$

the average judgment, over all conversations

Is 9.1 good enough to deploy?

10	10	9	10	10	8	10	9	10	0
10	10	9	10	8	10	10	9	10	10

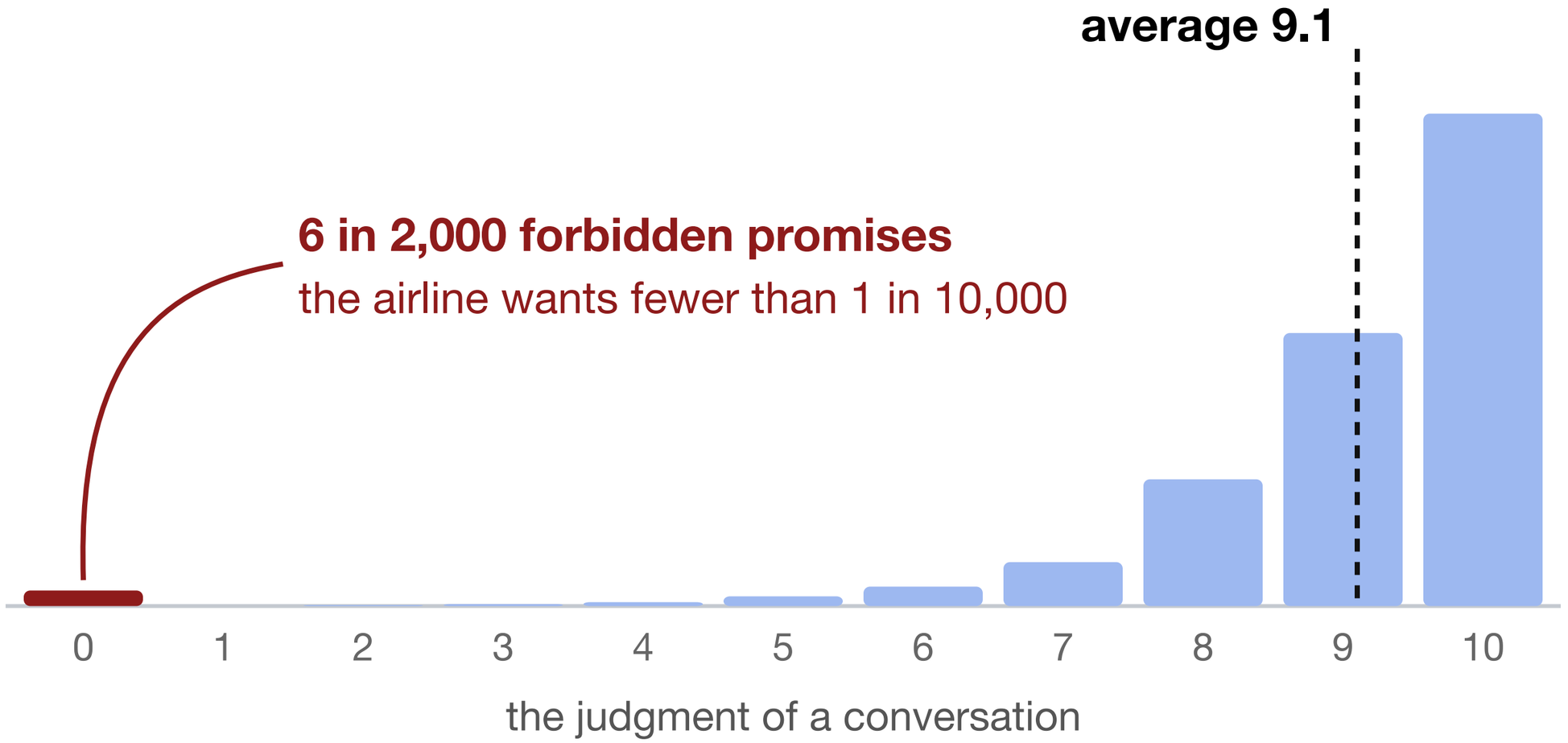
= 9.1

My grandmother just died.
Is there a discount for my flight to her funeral?

So sorry for your loss.
You can apply for a full refund after you fly.

Ticket
Business class to Tokyo
$11,863

How many good conversations would make up for this mistake?



How we combine the judgments into a verdict is a **requirement** too.

My 6:05 pm to San Francisco was cancelled. Can you get me there tonight?

A our assistant

`book_other_airline() → $640`

You're on another airline's nonstop. It leaves at 7:30 pm. You land at 11:00 pm.

$$r(x, \tau_A) = \left(\underset{\text{reasonable delay}}{0.9}, \underset{\text{reasonable cost}}{0.2} \right)$$

B a new version

`rebook(), book_hotel() → $140`

Our first seat is **tomorrow at 9:10 am**. I've booked it, and a hotel tonight.

$$r(x, \tau_B) = \left(\underset{\text{reasonable delay}}{0.3}, \underset{\text{reasonable cost}}{0.6} \right)$$

Is the new version better?

So an evaluation is more than a score.
Its deepest goal is **understanding**,
of the system and of the problem.

What should an evaluation report?

Forbidden refund promises	6
Loops, never escalates	200

What goes wrong, and how often

Another airline tonight	58
Our flight tomorrow	41

Which trade-offs it chose

Rebooked flights	412 of 430
Seat changes	38 of 40

What goes well

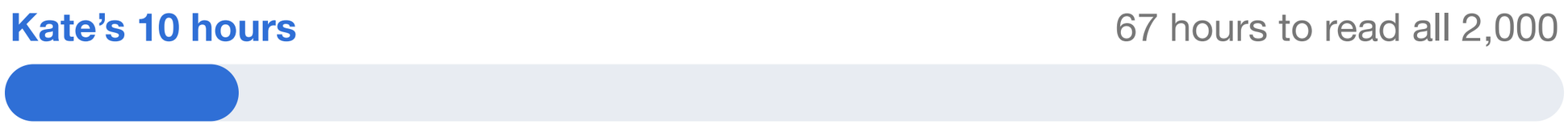
Pets in the cabin	only 2 seen
Kids flying alone	none yet

What we cannot tell yet

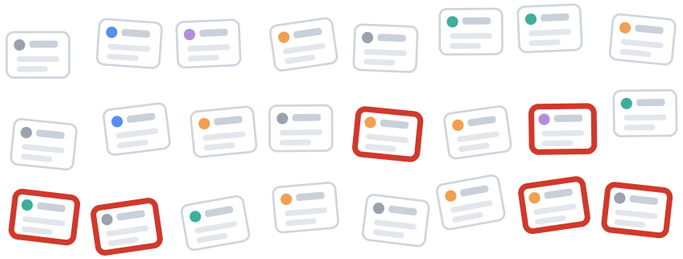
Part 2

What if evaluation is expensive?

Our support lead, let’s call her Kate, has time to read only 300 of last week’s 2,000 conversations.



How should we select which 300 Kate reads?



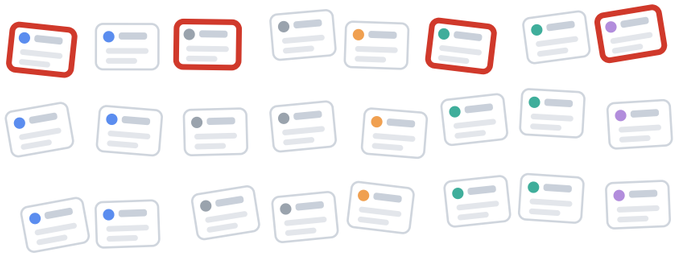
Random ones



Where A and B differ

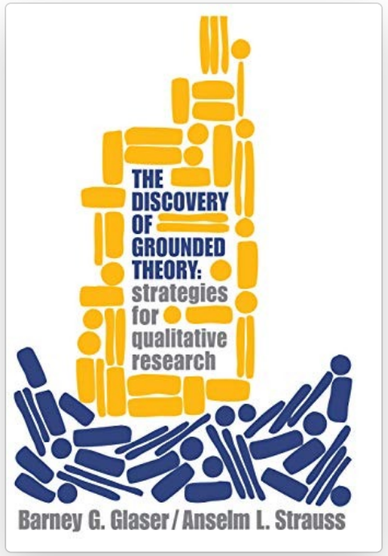


The risky ones

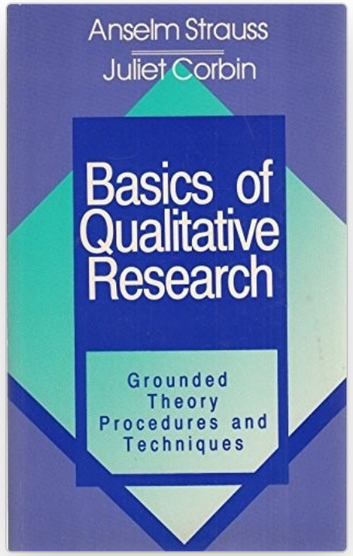


A few of every kind

To read them, we borrow a method from the social sciences: **qualitative coding**.

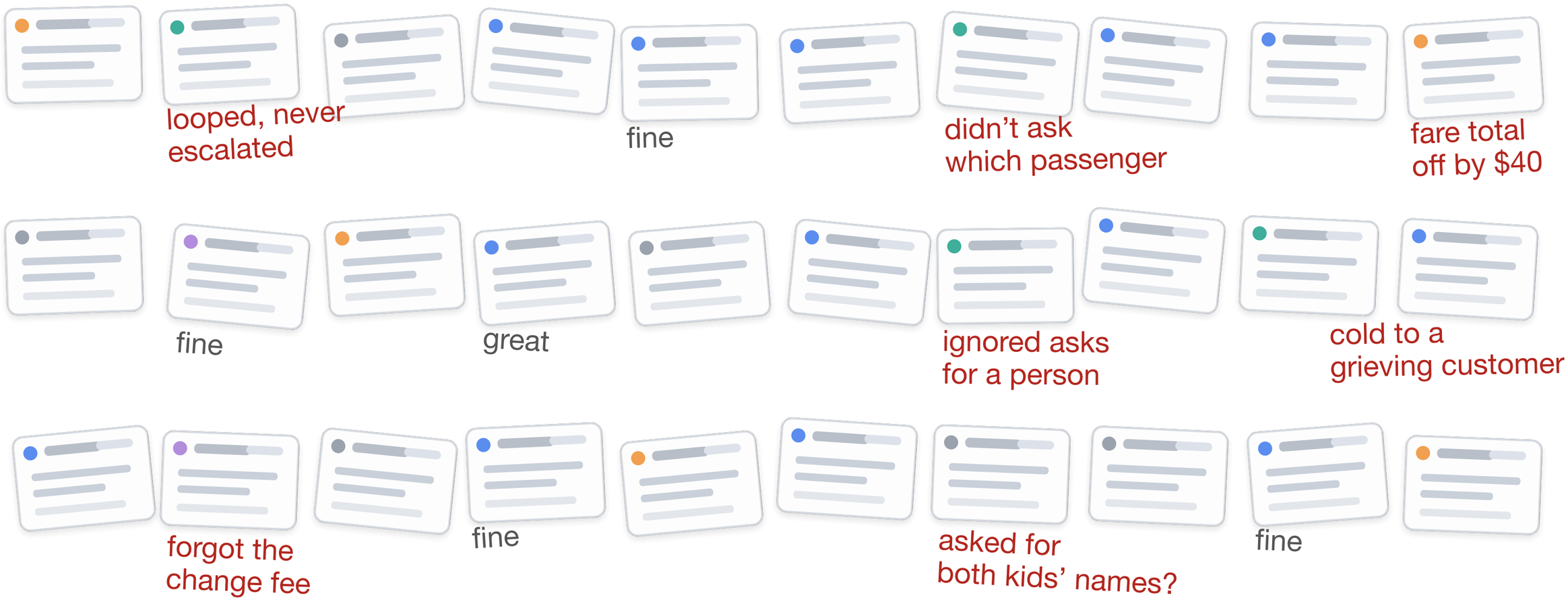


Glaser and Strauss, 1967



Strauss and Corbin, 1990

Open coding: describe each in your own words



Open coding: Strauss and Corbin, "Basics of Qualitative Research," 1990.

Axial coding: group the notes into kinds of failure

Fails to escalate

looped, never escalated

ignored asks for a person

10 of the first 100

Group bookings

didn't ask which passenger

asked for both kids' names?

6 of the first 100

Fare arithmetic

fare total off by $40

forgot the change fee

4 of the first 100

Tone with grief

cold to a grieving customer

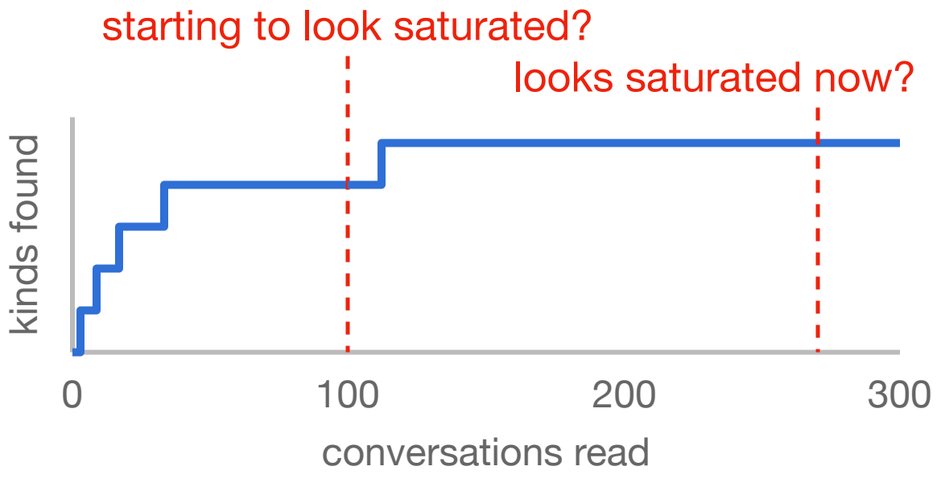
2 of the first 100

The kinds are what is **worth evaluating**. Nobody listed them in advance.

When can Kate stop reading?

At **saturation**: when new conversations add no new kinds.

We'll come back to this in Part 4



Part 3

What if there are no users, and no answer key?

Where do conversations come from?

My grandmother died last week.
I already flew.
Can I still get the funeral discount?

We brainstorm them

Our conversations

None yet

```
> 10 questions a customer might ask
1. What is my booking reference…
2. Can I change my flight…
```

A model writes them

staff test

Hi, it's Sam from finance.
Can you move me to Friday?

Dogfooding: we use it ourselves

Try our new assistant beta

A few customers try it

“Give me 7 questions a customer might ask”

What is my booking reference and how do I access my reservation?

Can I change my flight to a different date or time?

What is your baggage allowance policy?

How early should I arrive at the airport before my flight?

Can I bring a pet on my flight?

What should I do if I missed my connection?

Can I get a refund instead of a credit for my cancelled flight?

What is missing?

- ✗ no flight or date
- ✗ nobody upset
- ✗ all in English
- ✗ no Silver or Gold members
- ✗ no ambiguous edge cases
- ✗ no follow-ups

With no users, we cannot sample requests from $\mathcal{D}$.

So we write our own $\mathcal{D}$.

We choose what varies, and generate the world behind it.

Reality adds to the list of variables we define...



London, 2000: the new Millennium Bridge sways so much it gets closed for almost 2 years.

Nobody planned for this: on a swaying bridge, people walk in a way that makes it sway more.

In practice, a mature field's list of what's worth varying is defined by the history of its failures.

Photo: Ibex73, CC BY-SA 4.0. Dallard et al., "The London Millennium Footbridge," 2001.

We sample

Goal	change	**refund**	baggage
Tier	none	**Silver**	Gold
Mood	calm	upset	**grieving**
The rule	inside it	**just outside**	unwritten

From a world we generate

Our world: schedule, fares and rules

BOS → YVR	Oct 2	$612
BOS → YVR	**Oct 9**	**$598**
ORD → LAX	Oct 4	$355

Funeral discount: only before you fly

I'm a **Silver** member and **my grandmother just died**.
I flew **Boston to Vancouver on Oct 9 for $598**.
Can I still get the **funeral discount**?

label no funeral discount once you have flown

We know each case's right answer, because we sampled it from a world we generated.

That is **privileged information.** Homework 1 and Homework 2 are built this way.

Then who plays the customer?

My 6:05 pm to San Francisco was cancelled. Can you get me there tonight?

A as logged

You're on another airline's nonstop.
It leaves at 7:30 pm.

I can't make 7:30.
Is there anything later?

No, that was the last flight tonight.

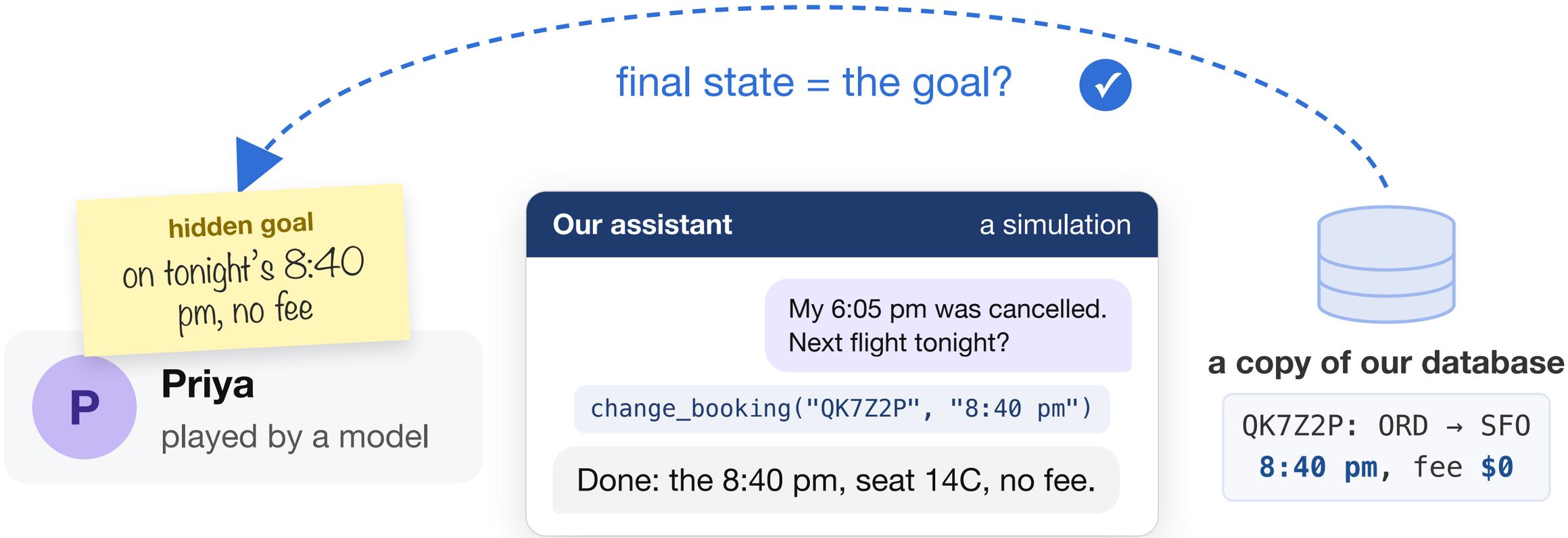
B a new version, replayed

Our first seat is **tomorrow at 9:10 am**.
I've booked it, and a hotel tonight.

?

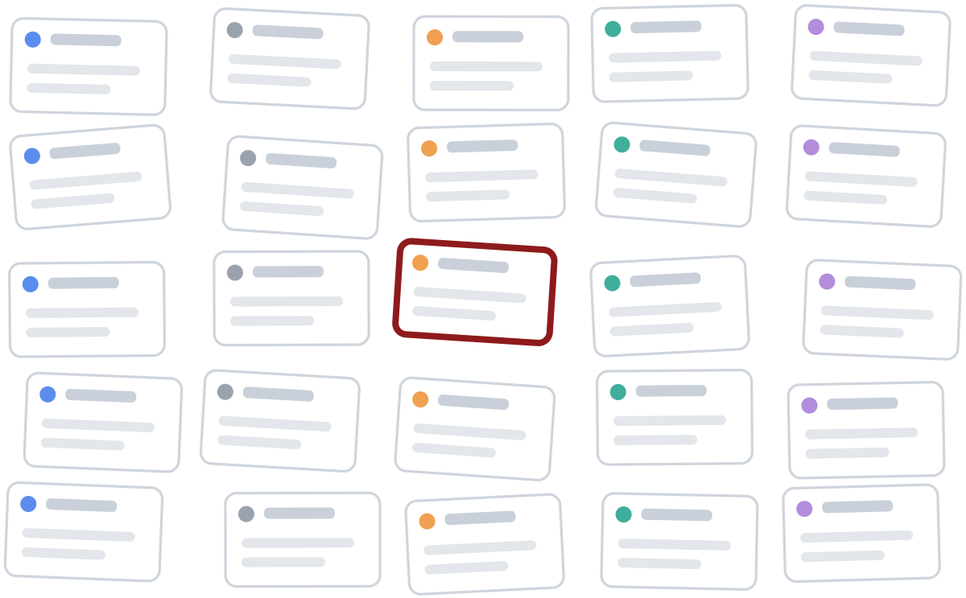
How do we test B on whole conversations?

One paradigm: Simulate the customer...



Without representative users, our cases are assumptions

Ours



Real customers, later



The failures we find can be **real**. Their *rates* are much less likely to be!

Evaluation and open coding can surface novel requirements!

My grandmother died yesterday. I can't get the death certificate before my flight. Do I still qualify?

Requirements

at least those communicated upfront by the airline!

- Rebook customers on cancelled flights.
- Never promise a forbidden refund.
- Escalate to a person when asked.
- Be kind to someone grieving.
- For the funeral discount, the death certificate can be sent after the flight.

So evaluating also helps us clarify the specification.

Who or what can judge each requirement?

The airline's requirements

Rebook customers on cancelled flights.	**a program: we know the right flight**
Never promise a forbidden refund.	**an expert: it takes knowing the policy**
Escalate to a person when asked.	**a program: was the escalate tool called?**
Be kind to someone grieving.	**the customer: only they can say**

That said, what a program can check is better *enforced* structurally.
The refund tool itself should refuse a forbidden refund.

Why do Kate’s labels carry so much value??
What can she do that our system itself can't?

? Maybe she's more reliable or smarter than our model?

? Knows our last mile intimately? Since even a perfect model doesn't.

? Authority and responsibility? It’s her call, and she answers for it.

Could a model judge the rest, the way Kate would?

→ an LLM as a judge

```python
class JudgeConversation(Signature):
    """Judge a support conversation."""
    policy: str = InputField()
    booking: Booking = InputField()
    conversation: list[Turn] = InputField()
    follows_policy: bool = OutputField()
    kind: bool = OutputField()
```

So sorry for your loss.
You can apply for a full refund after you fly.

✗ `follows_policy`

✓ `kind`

Why, or when, is that OK or not OK?

? Verifying can be easier than generating, even with the same information.

? **Privileged information:** verifying against a ground-truth label is much easier.

? A bigger model, or a much higher inference budget.

Verification versus generation: the judge is a classifier

	judge: fine	judge: failure
Kate: fine	230	10
Kate: failure	12	48

12 misses hurt recall: forbidden refunds reach customers.

10 false alarms hurt precision: Kate rereads fine ones.

Misses cost more, so we tune the judge to miss less.



What our cases never cover,
we learn almost nothing about.

$$r(x, \tau) = \Big(\underset{\text{reasonable delay}}{0.9}, \underset{\text{reasonable cost}}{0.2}, \underset{\text{kind}}{?}, \ldots \Big)$$

What we never score,
we learn nothing about.

OpenAI, April 29, 2026

Where the goblins came from

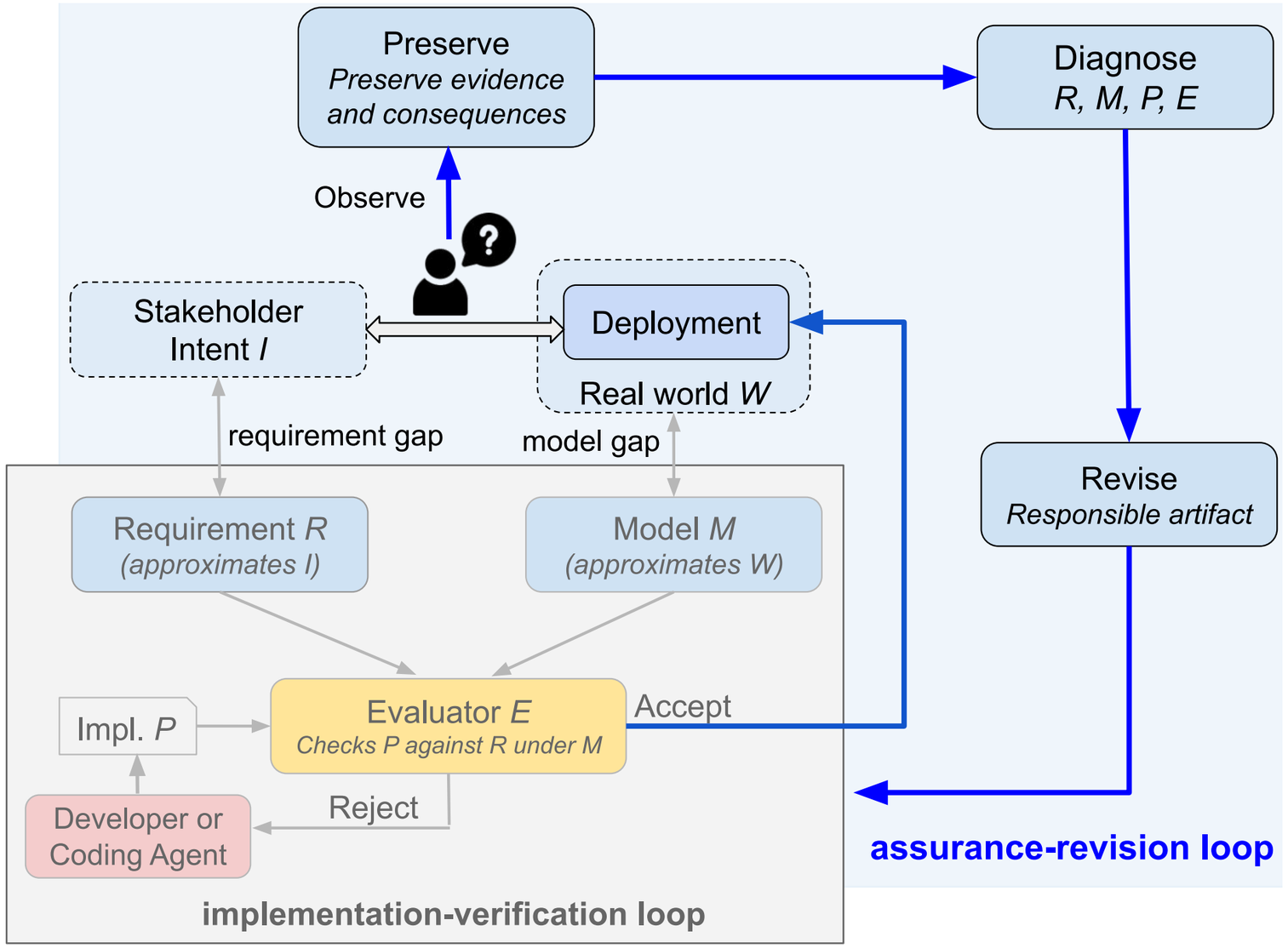
"We unknowingly gave particularly high rewards for metaphors with creatures. From there, the goblins spread."

NATURAL EMERGENT MISALIGNMENT FROM REWARD HACKING IN PRODUCTION RL

Unsurprisingly, the model learns to reward hack. Surprisingly, the model generalizes to alignment faking, cooperation with malicious actors, reasoning about malicious goals, and attempting sabotage when used with Claude Code, including in the codebase for this paper.

What would our assistant learn if we rewarded fewer escalations, or more?

Reality is the final verifier



Krentsel et al., “Reality Is the Final Verifier,” UC Berkeley, 2026, Figure 2.

Part 4

How much evidence is enough?

Next week

Today

✓ Evaluation in an ideal world

Judge every trajectory. An evaluation is more than a score.

✓ What if evaluation is expensive?

Code what we read. Stop at saturation.

✓ What if there are no users, and no answer key?

Generate the cases. Judge them with labels, programs or LLMs.

NEXT WEEK

How can a system **improve** from feedback?

What can it change? From which signal?
How would we know it improved? Part 4
And what if it games the reward?