The Oracle Problem: Barriers and Progress

Bo Waggoner
University of Colorado, Boulder
and Gensyn

a16z crypto
July 20, 2026

Introducing Myself (Bo Waggoner)

Research examples:

- Prediction markets and AMMs (2015-)
- Algorithmic game theory and mechanism design (2012-)
- Group decisionmaking, e.g. how DAOs avoid capture (2022-)

General interests:

- All things blockchain, e.g. teach Intro to Blockchain
  with Matt Garnett (EF) and Danny Ryan (Etherealize)
- Public goods; coordination problems
- Value of information, forecasting
- Algorithms, randomness, theoretical CS broadly

Collaborators

Gensyn: AI / blockchain / prediction market startup

Gabriel Andrade, Gensyn

Raf Frongillo, CU Boulder / Gensyn

Mary Monroe, CU Boulder / Gensyn

Marx Zang, Gensyn

Outline

(1/3) Modeling

how should we model the oracle problem?

comparison to peer prediction

The Oracle Problem

Definition:

Make information from outside the ***execution layer*** *accessible to smart contracts within it.*

The Oracle Problem

Definition:
*Make information from outside the **execution layer** accessible to smart contracts within it.*

Example #1: lending.
Use a price oracle to determine the value of collateral.

Example #2: insurance.
Use an event oracle to assess if conditions for a payout are met.

Example #3: prediction markets.
Use an event oracle to resolve the market.

The general approach

Initial approach:

- Oracle is a smart contract
- The owner submits a transaction storing the answer

The general approach

Initial approach:

- Oracle is a smart contract
- The owner submits a transaction storing the answer

Elaborations:

- Owner is a *governed organization*
- Smart contract aggregates among *many contributors*
- Incentive structures
- Cryptographic magic? *(e.g. owner is a TEE)*

Oracle Problem - state of the art

Current **toolkit:**

- Bro code *(trust me, bro)*
- Reputation
- Agreement rewards, slashing
- Disputes and escalation
- Backstop with "governance" *(i.e. my bros)*
- Trusted execution environments (TEEs)

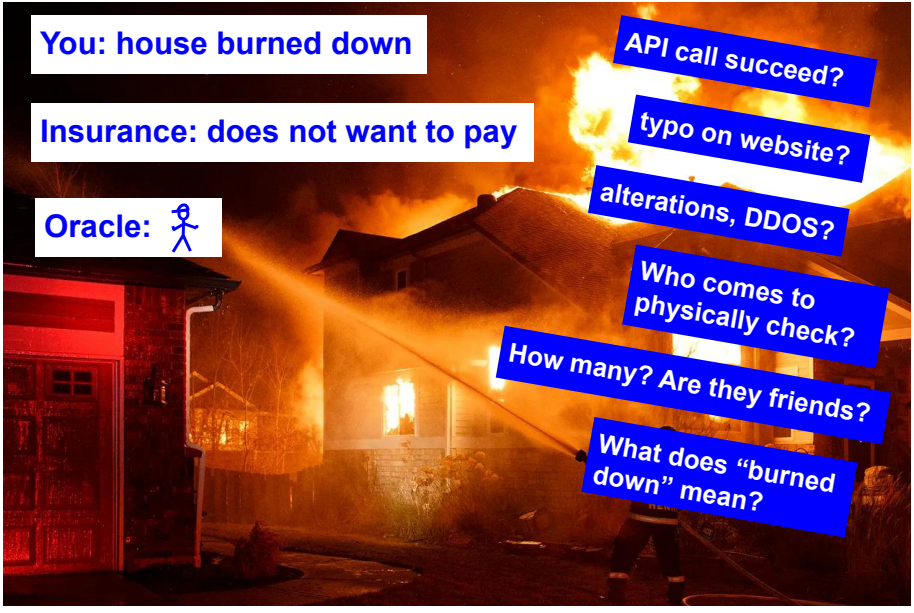
e.g. *Eskandari et al. (2021), Hassan et al. (2023), Cong et al. (2025), Caldarelli and Ornaghi (2026)*

Example: fire insurance

Example: fire insurance

Example: fire insurance

Example: fire insurance



Key Challenges for Oracles

1. **Manipulation**

2. **Free riders**

3. ~~**Ambiguity**~~ *(assume away for today)*

Key Challenges for Oracles

1. **Manipulation**
   - Attackers *expend resources*...
   - to *change the outcome* of the oracle...
   - or make it *unavailable*.

   note: attackers ∩ oracle

2. **Free riders**

3. ~~**Ambiguity**~~ *(assume away for today)*

Key Challenges for Oracles

1. Manipulation

- Attackers *expend resources*...
- to *change the outcome* of the oracle...
- or make it *unavailable*.

note: attackers ∩ oracle

2. Free riders

- Oracle participants can *independently investigate*...
- or not.

~~3. Ambiguity~~ *(assume away for today)*

Claim: Oracle Problem = Mechanism Design

The (Unambiguous) Oracle Problem:

- There is a bit that everybody knows, and
- we want to put it on the blockchain, but
- some want to change it. *Also, we're lazy.*

Claim: Oracle Problem = Mechanism Design

The (Unambiguous) Oracle Problem:

- There is a bit that everybody knows, and
- we want to put it on the blockchain, but
- some want to change it. *Also, we're lazy.*

Claim #1: we *can* use mechanism design.

- Blockchain provides: rules, messages, payments, commitment...

Claim: Oracle Problem = Mechanism Design

The (Unambiguous) Oracle Problem:

- There is a bit that everybody knows, and
- we want to put it on the blockchain, but
- some want to change it. *Also, we're lazy.*

Claim #1: we *can* use mechanism design.

Claim #2: we *cannot* use e.g. consensus.

- Everyone lives with the bit *written to the blockchain*
- Don't assume access to protocol-level primitives
- Off-chain activity still must *move on-chain*

Claim: Oracle Problem = Mechanism Design

The (Unambiguous) Oracle Problem:

- There is a bit that everybody knows, and
- we want to put it on the blockchain, but
- some want to change it. *Also, we're lazy.*

Claim #1: we *can* use mechanism design.

Claim #2: we *cannot* use e.g. consensus.

Claim #3: game-theoretic model of behavior is *apt*.

- Oracle as a for-profit org, participants as utility-maximizing
- Attackers as *having* resources and *spending* them

A “base” model

An oracle:

- Has n independent strategic participants
- Receives a query for $Y \in \{0, 1\}$ and fee $F > 0$ $Y \sim$ *Uniform*

A "base" model

An oracle:

- Has n independent strategic participants
- Receives a query for $Y \in \{0, 1\}$ and fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

A "base" model

An oracle:

- Has n independent strategic participants
- Receives a query for $Y \in \{0, 1\}$ and fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

Example:

- Agents submit a bit, output the majority vote
- Pay majority agents F/n each — *"output agreement"*

A "base" model

An oracle:

- Has n independent strategic participants
- Receives a query for $Y \in \{0, 1\}$ and fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

Example:

- Agents submit a bit, output the majority vote
- Pay majority agents F/n each — *"output agreement"*

Solution concepts:

- Simultaneous-move equilibrium — *all report together*
- Subgame perfect equilibrium — *report one at a time*

Comparison: Peer Prediction

Peer prediction: *ambiguous* version.

- Random variables $(X_1, \ldots, X_n, Y) \sim$ common prior
- Each agent i observes X_i *makes Bayesian update*
- *They participate in a mechanism.*
- Mechanism assigns payments

Comparison: Peer Prediction

Peer prediction: *ambiguous* version.

- Random variables $(X_1, \ldots, X_n, Y) \sim$ common prior
- Each agent i observes X_i *makes Bayesian update*
- *They participate in a mechanism.*
- Mechanism assigns payments

Dealbreakers:

1. Usually lacks output $\hat{Y}$ *poor epistemics*
2. Simultaneous moves *assumes no communication/coordination*
3. Doesn't model manipulation or exogenous incentives

A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

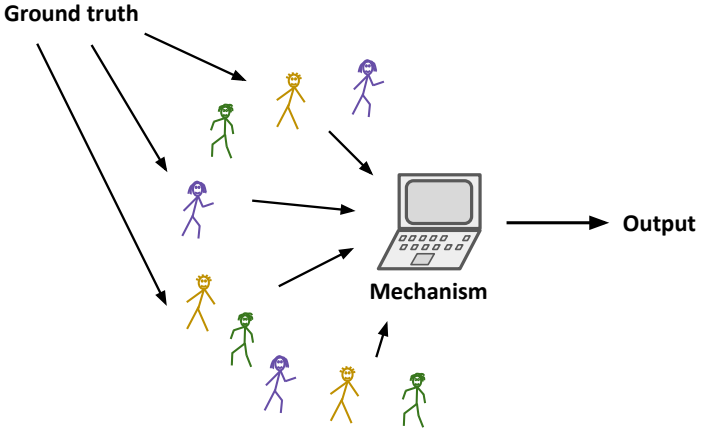
A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

Proof.

Agents can ignore their information and play some equilibrium. □



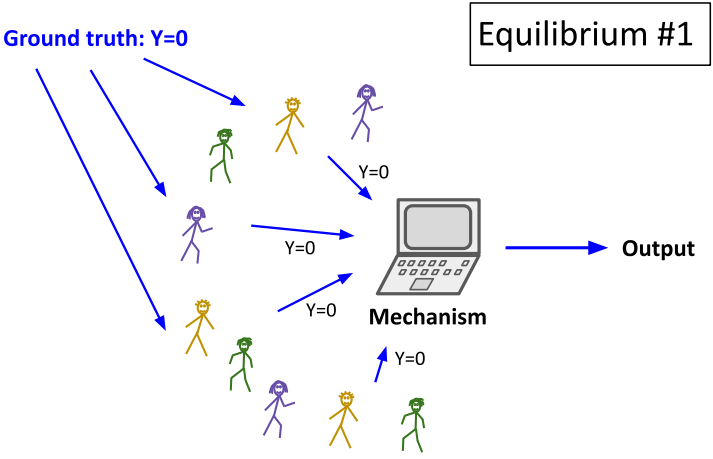
A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

Proof.

Agents can ignore their information and play some equilibrium. □



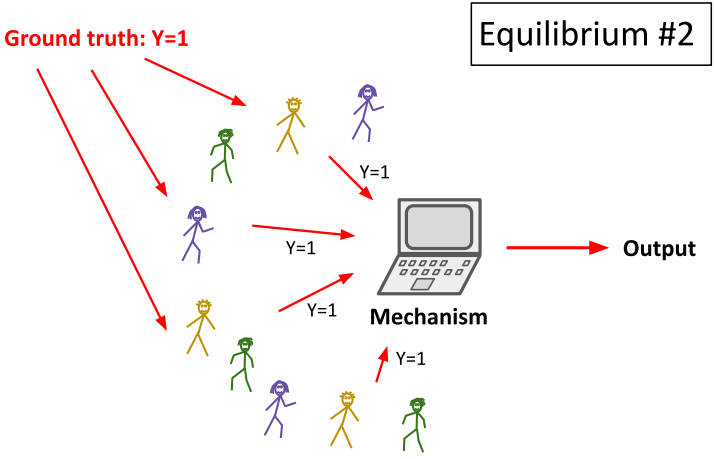
A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

Proof.

Agents can ignore their information and play some equilibrium. □



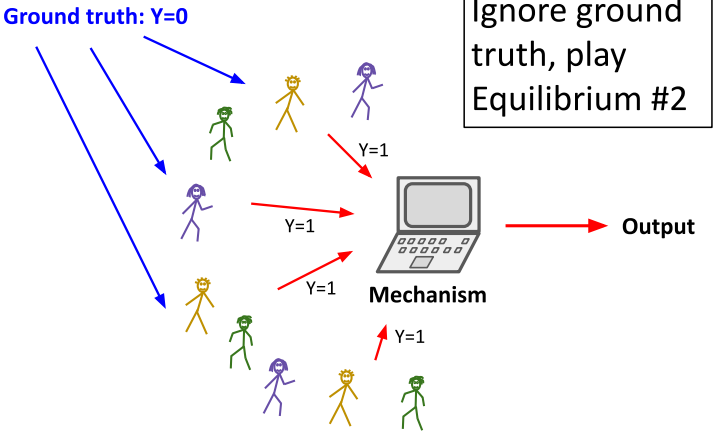
A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

Proof.

Agents can ignore their information and play some equilibrium. □



A generic impossibility result

Theorem (e.g. Chen & W. 2014)

In any mechanism, there is an equilibrium where $\hat{Y} \neq Y$ *w.prob.* $\geq \frac{1}{2}$.

Proof.

Agents can ignore their information and play some equilibrium. □

PP literature: make truthful the "best" eq.

Oracle problem: take lessons, but need different focus.

(2/3) Barriers

manipulation

free riding

Adding manipulation to base model

Base model:

- Query for $Y \in \{0, 1\}$, fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

Adding manipulation to base model

Base model:

- Query for $Y \in \{0, 1\}$, fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

Manipulation model:

- Attackers *expend resources*...
- to *change the outcome* of the oracle...
- or make it *unavailable*. — *attackers ∩ oracle*

Adding manipulation to base model

Base model:

- Query for $Y \in \{0, 1\}$, fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent observes Y — *unambiguous*
- *They participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*

Manipulation model:

- **Single attacker with pooled resources**
- **Resources are deployed as bribes to agents** — *with commitment*
- *Ignore for today: burning resources, unavailability*

Augmented model

- Query for $Y \in \{0, 1\}$, fee $F > 0$ — *$Y \sim$ Uniform*
- Every agent **and attacker** observes Y
- **Attacker commits to contingent payments** — *nonneg, sum to $\leq B$*
- *Agents participate in a mechanism* — *which does not know Y*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to F*
- **Attacker's payments execute** — *based on all actions*

Research question

Fix a mechanism (fee F) and attack (budget B), play an equilibrium.

- **Total success**: $\hat{Y} = Y$ in **all** equilibria. *for all attacks*
- **Partial success**: $\hat{Y} = Y$ in **some** equilibrium. *for all attacks*
- **Complete failure**: $\hat{Y} \neq Y$ w.pos.prob. in **all** equilibria. *some attack*

Research question

Fix a mechanism (fee F) and attack (budget B), play an equilibrium.

- **Total success**: $\hat{Y} = Y$ in **all** equilibria. *for all attacks*
- **Partial success**: $\hat{Y} = Y$ in **some** equilibrium. *for all attacks*
- **Complete failure**: $\hat{Y} \neq Y$ w.pos.prob. in **all** equilibria. *some attack*

Proposition (Simultaneous, from P.P. literature)

*For all $B \geq 0$, **no** simultaneous mechansim is a total success.*

Proposition (Sequential)

For all $B > 0$, every sequential mechanism is a complete failure.

Idea: when $Y = 0$, strictly incentivize the equilibrium for $Y = 1$ and vice versa.

Fix: reputation model

For rounds $t = 0, 1, 2, \ldots$:

- Request for $Y^t \in \{0, 1\}$, fee F
- conduct mechanism, output $\hat{Y}^t$
- **If $\hat{Y}^t = Y^t$: continue to next round**
- **If $\hat{Y}^t \neq Y^t$: stop**
- Agent utility is discounted sum: $\sum_{t=0}^{\infty} \delta^t \cdot \text{payment}_t$

Fix: reputation model

(Simplified) Reputation model: For rounds $t = 0, 1, 2, \ldots$:

- Request for $Y^t \in \{0, 1\}$, fee F
- conduct mechanism, output $\hat{Y}^t$
- **If $\hat{Y}^t = Y^t$: each i gets D_i = discounted sum**
- **If $\hat{Y}^t \neq Y^t$: stop**
- Agent utility is discounted sum: $\sum_{t=0}^{\infty} \delta^t \cdot \text{payment}_t$

Fix: reputation model

(Simplified) Reputation model: For rounds $t = 0, 1, 2, \ldots$:

- Request for $Y^t \in \{0, 1\}$, fee F
- conduct mechanism, output $\hat{Y}^t$
- **If $\hat{Y}^t = Y^t$: each i gets D_i = discounted sum**
- **If $\hat{Y}^t \neq Y^t$: stop**
- Agent utility is discounted sum: $\sum_{t=0}^{\infty} \delta^t \cdot \text{payment}_t$

Let $D := \sum_{i=1}^{n} D_i$ = total discounted future rewards

Go/no-go with reputation

Simple Majority mechanism:

- Each agent votes, $\hat{Y}$ = majority
- Split fee equally (regardless of votes)

Go/no-go with reputation

Simple Majority mechanism:

- Each agent votes, $\hat{Y} =$ majority
- Split fee equally (regardless of votes)

Theorem (Sequential)

In the reputation model with sequential equilibrium:

1. *If $B < \frac{1}{2}D$, Simple Majority is a total success.*
2. *If $B > D$, every mechanism is a complete failure.*

$B =$ attacker budget, $D =$ discounted sum of fees

Manipulation - summary

Conclusion: Incentive design is **not** effective against manipulation.

either simple things work, or nothing does.

Extensions: attacker burns money; makes sources unavailable

e.g. modeling TEEs

Free riders

Free riders

Model

- Query for $Y \in \{0, 1\}$, fee $F > 0$ — $Y \sim$ ***Bernoulli***(p)
- Every agent **can pay** c to observe Y — *"exerts effort"*
- *Agents participate in a mechanism*
- The mechanism outputs $\hat{Y} \in \{0, 1\}$ and payments — *which sum to* F

Free riders: go/no-go

Let $p \leq \frac{1}{2}$ WLOG.

Output Agreement mechanism:

- Each agent votes, $\hat{Y} =$ majority
- Agents get r iff they agree with majority

Free riders: go/no-go

Let $p \leq \frac{1}{2}$ WLOG.

Output Agreement mechanism:

- Each agent votes, $\hat{Y} =$ majority
- Agents get r iff they agree with majority

Theorem (Free riders, no reputation, simultaneous)

With simultaneous-move equilibrium:

- *If $F \geq 2\frac{c}{p}$, OA with 2 agents is a partial success.*
- *If $F < \frac{c}{p}$, every mechanism is a complete failure.*

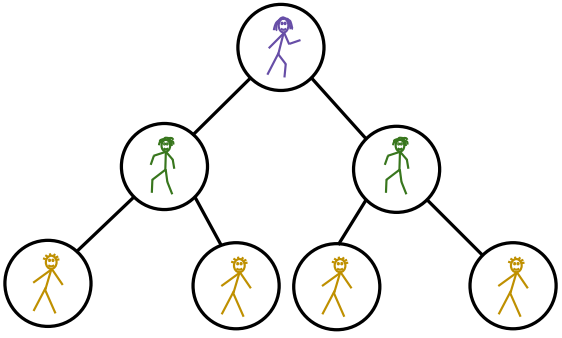
  Caveat: can spend less in expectation.

Free riders: go/no-go (cont'd)

Theorem (Free riders, no reputation, sequential)

With subgame perfect equil: ***every*** *mechanism is a complete failure.*

Idea: (1) at most one agent exerts effort, as later agents can just free ride.
(2) that agent should deviate to pretending.



Free riders with reputation

Centralized mechanism:

- $n = 1$ agent reports $\hat{Y}$
- That agent gets all the fees

Free riders with reputation

Centralized mechanism:

- $n = 1$ agent reports $\hat{Y}$
- That agent gets all the fees

Theorem (Free riders, reputation)

In the reputation free riders model:

- *With* $D > \frac{c}{\delta p}(1-\delta)(1-\delta+\delta p)$*, the centralized mechanism is a total success.*
- *With* $D < \frac{c}{\delta p}(1-\delta)(1-\delta+\delta p)$*, every mechanism is a complete failure.*

Holds for simultaneous and sequential.

Free Riders - summary

- Need to incentivize **information acquisition**
- Hard to incentivize **redundancy** in equilibrium

Sad, approx. truth: when any mechanism succeeds, **centralized** does.

(If time) more realistic mechanism

"Race to reveal" mechanism:

1. Hold a simultaneous-move output agreement vote.
   Only include first k agents to submit; use commit-reveal
2. Allow any agent to dispute the result. If so:
3. Hold a sequential-move vote among all n.

(If time) more realistic mechanism

"Race to reveal" mechanism:

1. Hold a simultaneous-move output agreement vote.
   Only include first k agents to submit; use commit-reveal
2. Allow any agent to dispute the result. If so:
3. Hold a sequential-move vote among all n.

Properties (with the right parameters):

- $> k$ agents exert effort and race to vote
- "guessing" early is not worth it
- manipulation is detected by effortful agents and disputed
- **(controversial)** in a dispute, agents exert effort in the vote
  off-equilibrium

(3/3) Progress

how to model and use AI?

A recent tool: AI agents

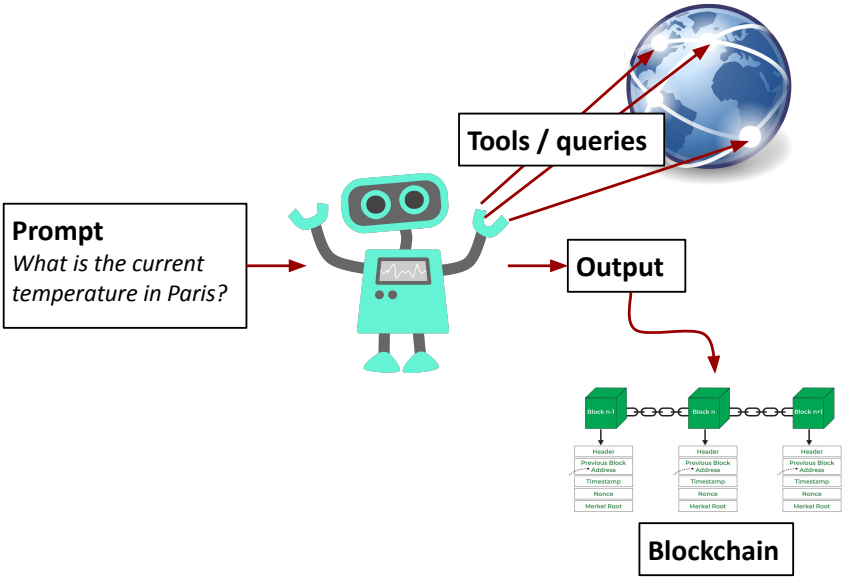
Idea (Gensyn, others): use an **AI agent** as the oracle.

Challenges:

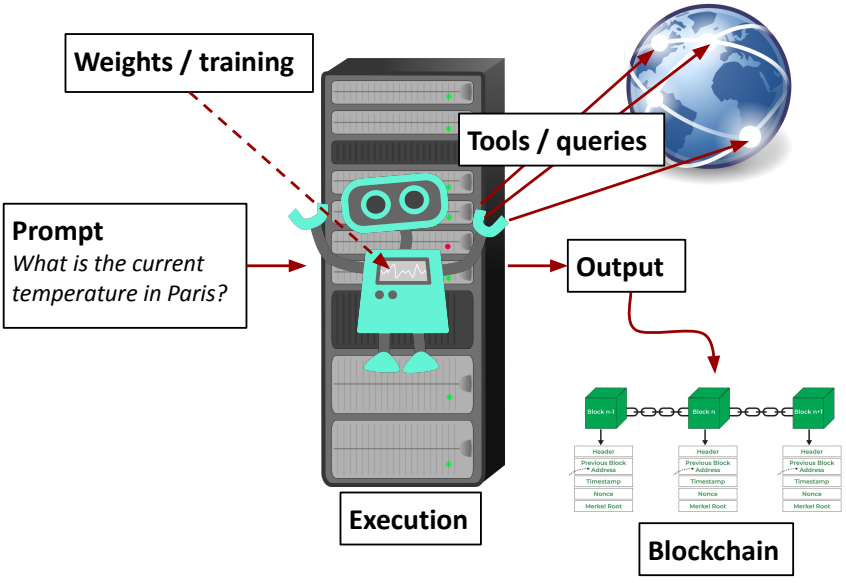
- Many vectors for manipulation
- Handling cases of inaccuracy, manipulation, or failure

Survey: *Calderalli (2025)*

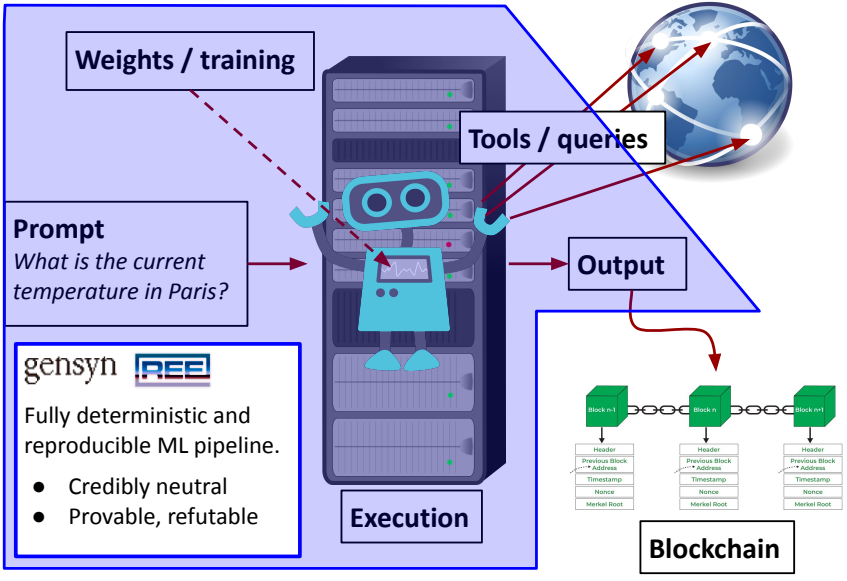
Manipulation vectors



Manipulation vectors



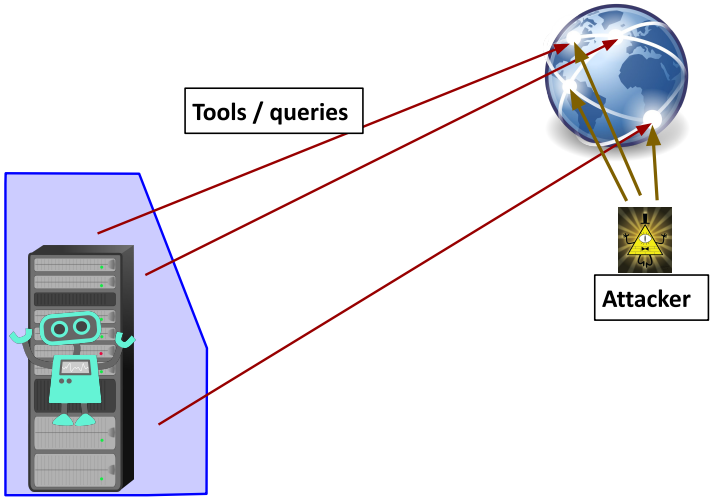
Manipulation vectors



Data source manipulation

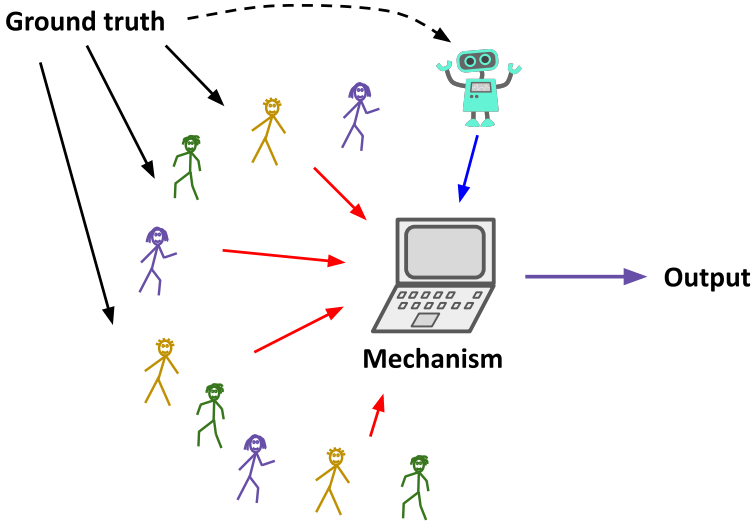
"Mini game": AI vs.
data source manipulation

(research in progress)



Modeling AI as a tool for oracles

Proposed model: AI = **independent signal of ground truth**.



Example integration

Optimistic escalation mechanism:

1. 🤖 reports $\hat{Y}^0$
2. any participant can dispute; if so:
3. hold a vote among n agents
4. $\hat{Y} = \begin{cases} \hat{Y}^0 & \text{at least } \alpha n \text{ votes for } \hat{Y}^0 \\ \neg\hat{Y}^0 & \text{otherwise} \end{cases}$

assume: $\Pr[\hat{Y}^0 = Y] = 1 - \epsilon$

$\alpha < \frac{1}{2}$

Example integration

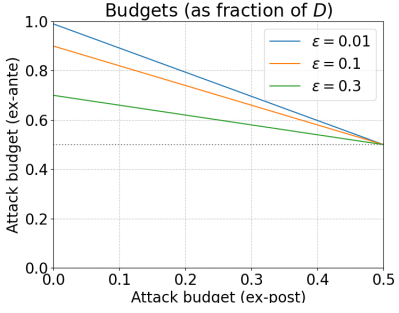
Optimistic escalation mechanism:

1. 🤖 reports $\hat{Y}^0$
2. any participant can dispute; if so:
3. hold a vote among n agents
4. $\hat{Y} = \begin{cases} \hat{Y}^0 & \text{at least } \alpha n \text{ votes for } \hat{Y}^0 \\ \neg\hat{Y}^0 & \text{otherwise} \end{cases}$

assume: $\Pr[\hat{Y}^0 = Y] = 1 - \epsilon$

$\alpha < \frac{1}{2}$

Results: by tuning α, can increase resources required for "ex-ante" attack while protecting against opportunistic "ex-post" attack.



Benefits of AI for ambiguity

Specific-general framing problem:

Give general query, let AI implement

ex: "temperature in Paris" → choose data sources

Benefits of AI for ambiguity

Specific-general framing problem:

Give general query, let AI implement

ex: "temperature in Paris" → choose data sources

Private or incomplete information:

Give AI reasoning and investigation budget

ex: "did house burn down" → adaptive search

Benefits of AI for ambiguity

Specific-general framing problem:

Give general query, let AI implement

ex: "temperature in Paris" → choose data sources

Private or incomplete information:

Give AI reasoning and investigation budget

ex: "did house burn down" → adaptive search

Subjectivity:

AI is (hopefully) **credibly neutral** and **predictable**

fixed model: anyone can test out scenarios

Wrapup

Summary

Claims:

- Oracle problem = mechanism design
- Main challenges: manipulation, freeriding, ~~ambiguity~~
- How to model them
- Barriers: go/no-go results
- Progress: using AI as a trusted input

Thanks!