On Public Permissionless Decentralized Ledger Oracles

UNIVERSITY OF TORONTO

Andreas Veneris[*,†], Ryan Berryhill[*], Neil Veira[*], Yuxi Cai[*], Georgios Fragkos[‡], Eirini Tsiropoulou[‡], John Adler[*] and Marco Merlini[*]

[*] Department of Electrical and Computer Engineering, University of Toronto

[†] Department of Computer Science, University of Toronto

[‡] Department of Electrical and Computer Engineering, University of New Mexico

Outline

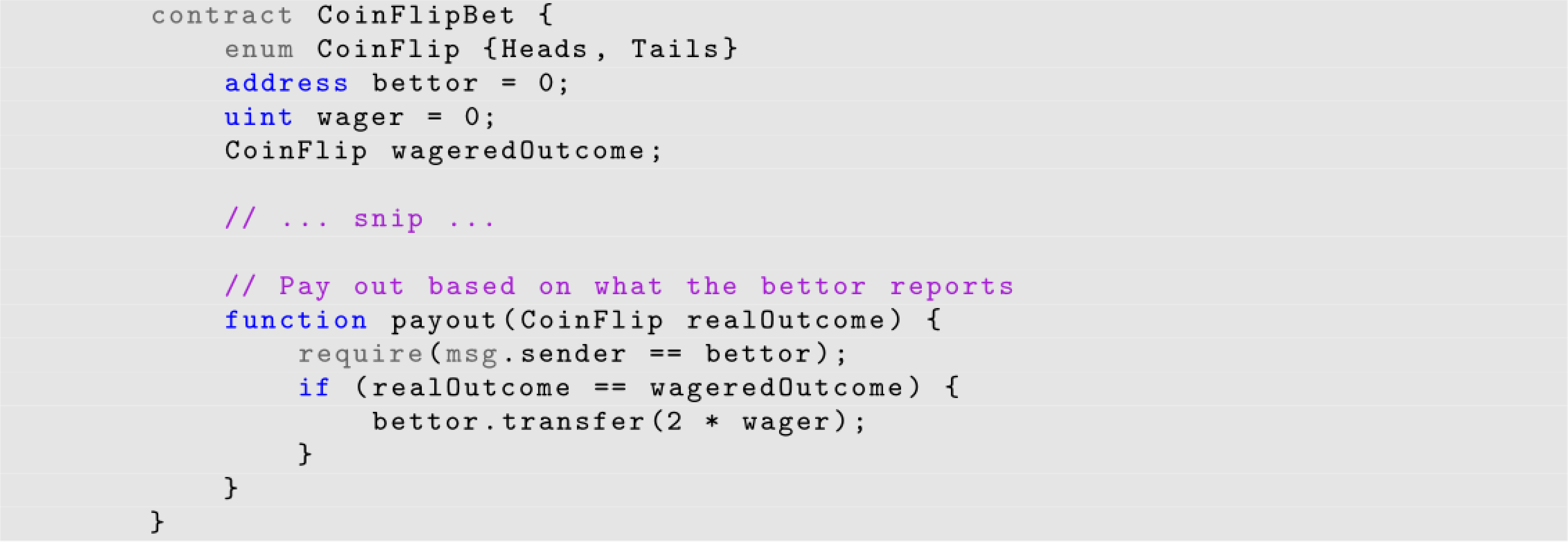
- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

Introduction

- Consider a betting smart-contract for the coin flip before the Superbowl
- We can take a bet, but how do we pay out a winning bet?



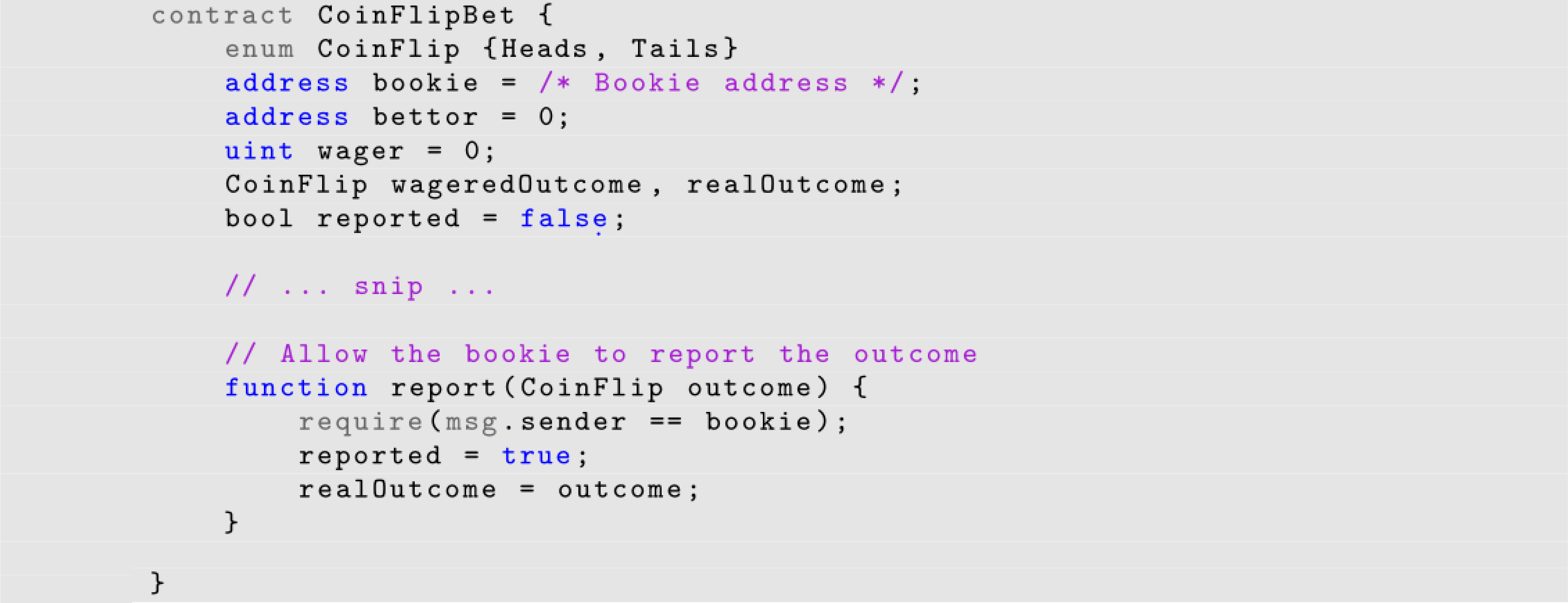
```
contract CoinFlipBet {
    enum CoinFlip {Heads, Tails}
    address bettor = 0;
    uint wager = 0;
    CoinFlip wageredOutcome;

    // ... snip ...

    // Pay out based on what the bettor reports
    function payout(CoinFlip realOutcome) {
        require(msg.sender == bettor);
        if (realOutcome == wageredOutcome) {
            bettor.transfer(2 * wager);
        }
    }
}
```

Introduction

- We can’t trust the bettor to report the outcome of the coin toss



```
contract CoinFlipBet {
    enum CoinFlip {Heads, Tails}
    address bookie = /* Bookie address */;
    address bettor = 0;
    uint wager = 0;
    CoinFlip wageredOutcome, realOutcome;
    bool reported = false;

    // ... snip ...

    // Allow the bookie to report the outcome
    function report(CoinFlip outcome) {
        require(msg.sender == bookie);
        reported = true;
        realOutcome = outcome;
    }

}
```

Introduction

The Gateway Problem

- If the bookie is trusted, then why use a decentralized smart contract?
- If you need a blockchain to interact with the real world, you have a big problem – *Blockchains are blind to real-life world events!*
  - e.g., prediction markets, insurance, managing financial assets, adjudication
- Solution: ***query a decentralized oracle!***

Other benefits of decentralized oracles:

- Data collection and annotation via crowd-sourcing
- Ensuring data availability

Introduction

Current oracle solutions – they all require *"centralized trust"*

Oraclize.it

- Fetches data from specified web source
- Requires "trust" to a central server – can deny requests or collude with website owners

Town Crier

- Similar + trusted hardware proofs (e.g., Intel's SGX) verify authenticity
- Also requires "trust" to a central server and Intel Corp.

Introduction

Current oracle solutions – they all require *“centralized trust”*

Chainlink

- Aims to provide a cross-chain portal to internet-available information i.e., data available on websites
- Although with multiple information sources, selection and aggregation mechanisms are proposed by the user

Augur

- Token holders report answers or challenge reports
- Requires “trust” to a *designated reporter* – a privileged (centralized) user who reports first

Introduction

Trustless and decentralized oracle markets

- **Decentralized = permissionless + equiprivileged:**
  - Any member of the public can answer questions
  - Needs proper game-theoretical incentives for honest reporting

The lazy equilibrium

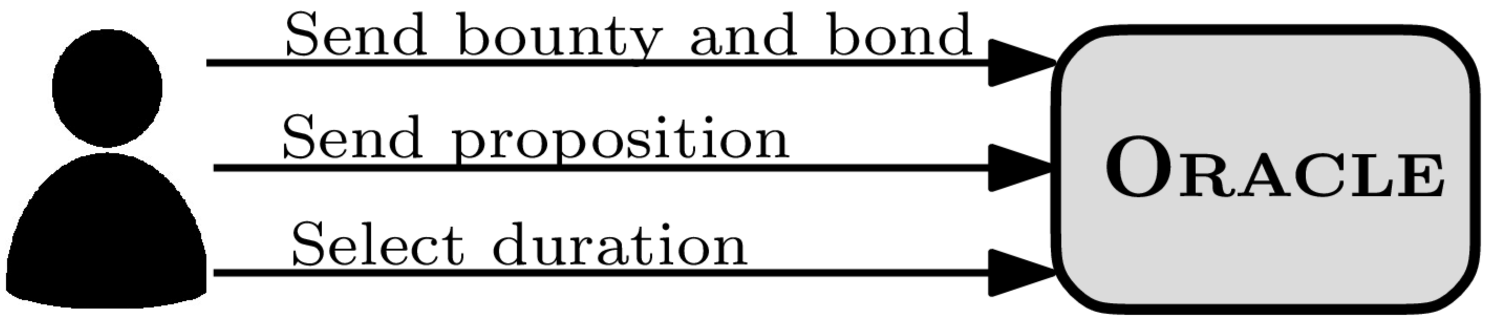
- Why wouldn’t everyone just always vote True?
- Easier than trying to figure out the “correct” answer
- A Nash equilibrium – analogous to the Verifier’s Dilemma

Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

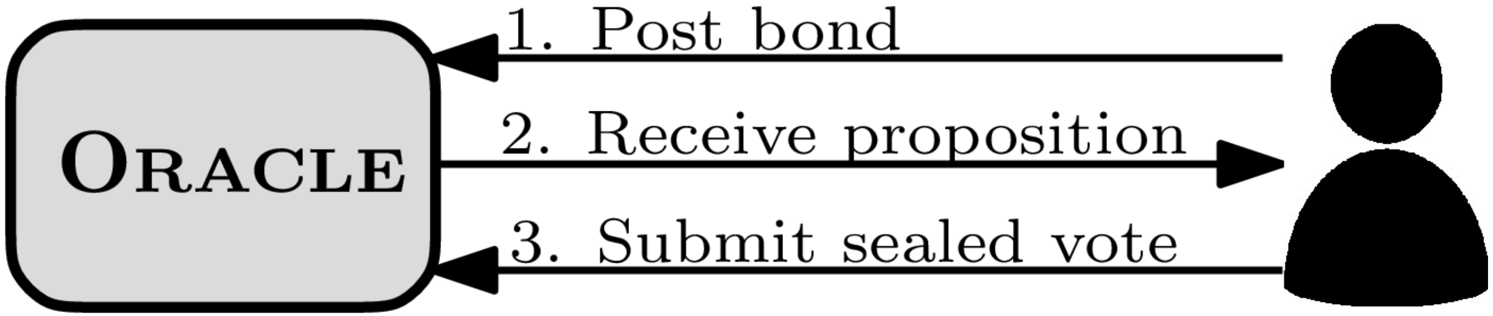
Decentralized Oracle Model

- Smart contract maintains pool of active **Boolean** (True or False) propositions $p_1, p_2, p_3, \ldots$
- Users can submit new propositions at any time
- Must also submit:
  - **Bounty** to pay for participation
  - **Bond** for incentives
  - **Duration** of proposition



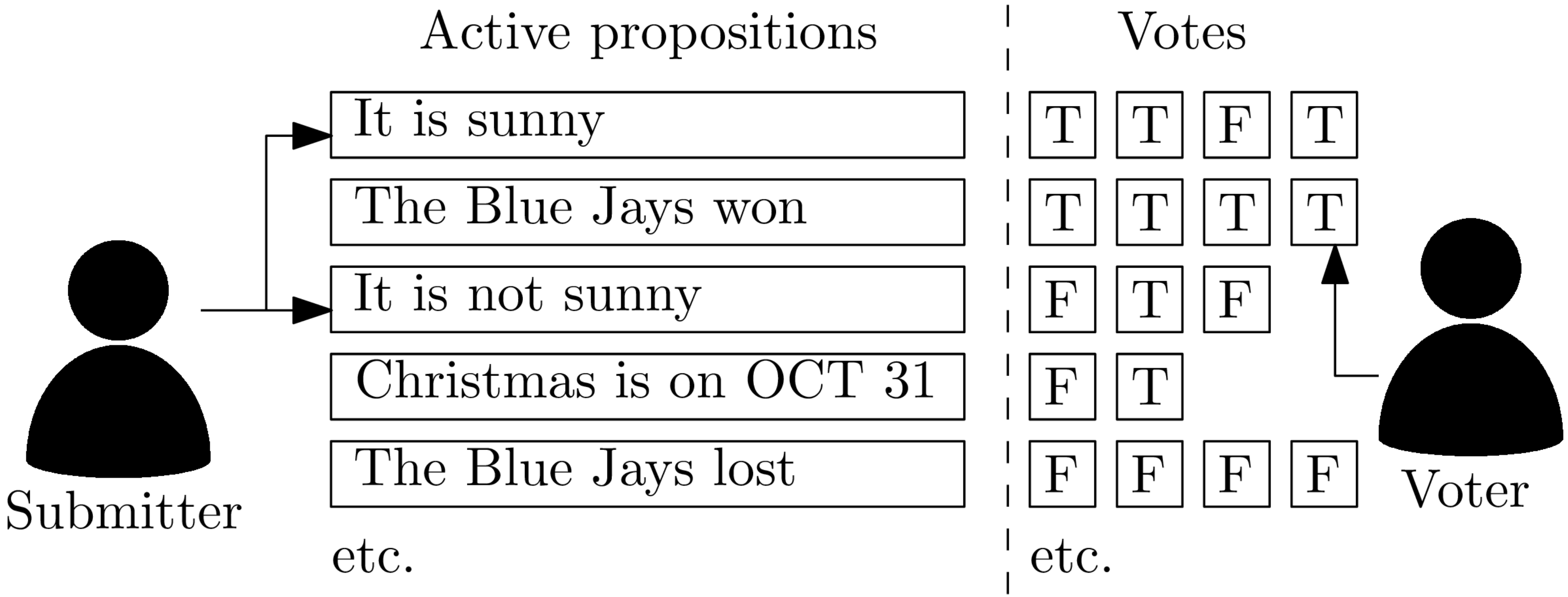
Decentralized Oracle Model

- **Voter**: any user that requests participation by posting a bond
- Receives a **randomly** chosen proposition
- Submits sealed vote of True or False



Decentralized Oracle Model

- Each proposition is *randomly* assigned to multiple voters
- Votes are tallied to determine output when proposition expires



Active propositions	Votes			
It is sunny	T	T	F	T
The Blue Jays won	T	T	T	T
It is not sunny	F	T	F	
Christmas is on OCT 31	F	T		
The Blue Jays lost	F	F	F	F
etc.	etc.			

Decentralized Oracle Model

- **Private opinion ($\boldsymbol{PO_{ij}}$):** Opinion of voter v_i on proposition p_j (True/False)
  - Honest voters keep their PO unknown to other voters
  - Dishonest voters may collude and share their PO (i.e., may vote differently to POij)

- **Voting strategy**: $\sigma_{ij}\left(PO_{ij}\right) =$ answer that v_i reports on p_j
  - If honest, then $\sigma_{ij}\left(PO_{ij}\right) = PO_{ij}$

- **Most Probable Private Opinion ($\boldsymbol{MPPO_j}$)** : Majority PO on p_j (True/False)
  - Serves as the 'ground truth' or the 'correct' answer
  - We want the decentralized oracle (market) to output $MPPO_j$

Decentralized Oracle Model

Definitions

- $\boldsymbol{c_i}$ = voter v_i's *perceived* probability of agreeing with $MPPO$
  - Note that v_i generally does not know other voters' PO

- $\boldsymbol{c}$ = probability that randomly selected voter reports $MPPO$
  - Measure of "degree of contention" of proposition
  - $\mathrm{c} = 1$ → everyone agrees
  - $c = 0.5$ → maximum disagreement
  - $0.5 \leq c \leq 1$ *if everyone is honest*

Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

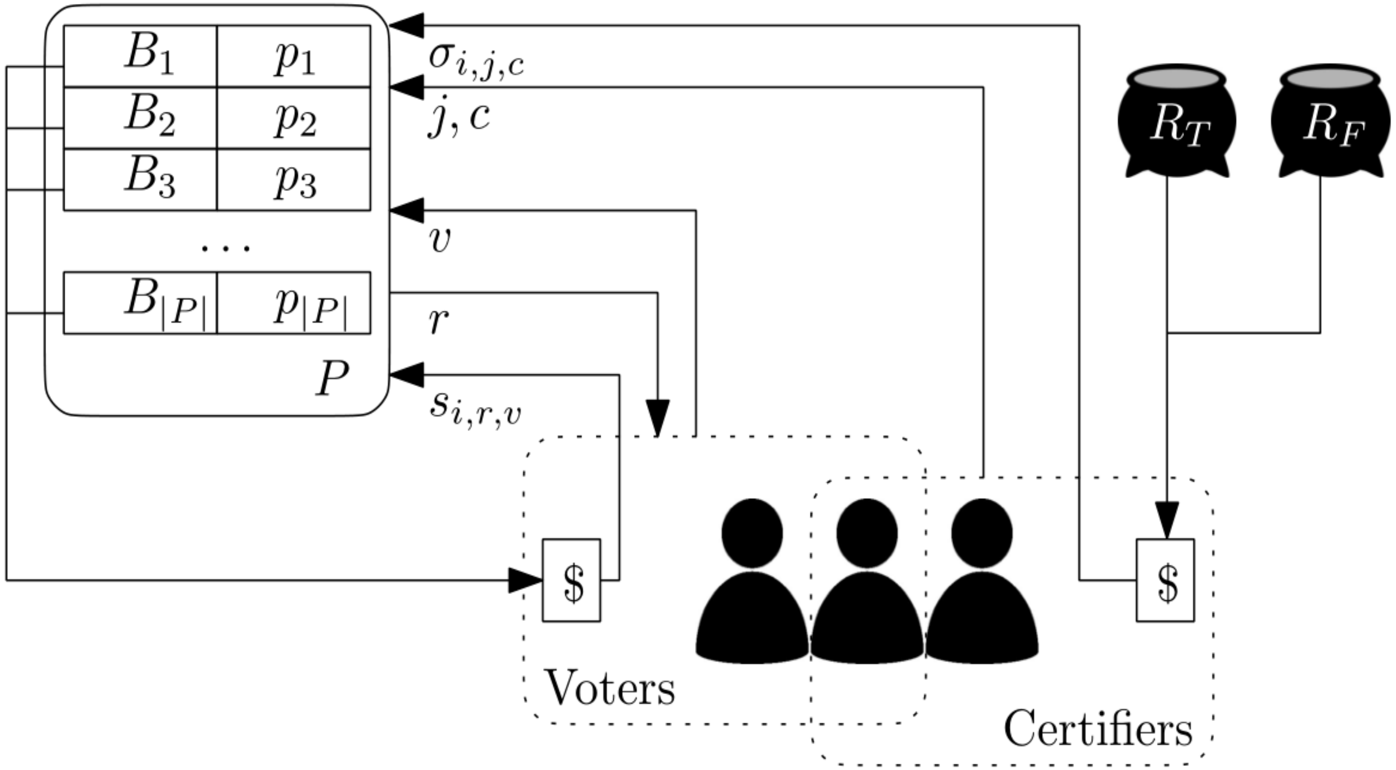
Double-Player Protocol

- First decentralized oracle protocol
- Two types of voters: **voters** and **certifiers**

	Bond and Perceived Reward Value	**Proposition Assignment**	**Payoff Rules**	**Risk interpretation**
Voters	Small	Random	Reward: agree with the majorities of **both** voters and certifiers; Penalty: vote against **both** majorities	Low risk, low reward
Certifiers	Large	Chosen by certifier	Reward: agree with the majorities of **both** voters and certifiers; Penalty: vote against **either** majorities	High risk, high reward

Double-Player Protocol

- **Rewards:** payment for voting correctly
- Paid from a **reward pool** which depends on the vote value (True/False)
- Pools are funded by bounties and forfeited bonds
- After a proposition is decided True/False, reward pool for the opposite value increases
- Always voting the same way is not the most profitable strategy (because the opposite pool increases)



Double-Player Protocol - Analysis

- Assume each player's strategy directly depends on <u>only</u> PO_i
- Voting and certification can be seen as two independent series of Bernoulli trials
- the probability of $MPPO_j$ being selected by the majority of n_j voters on proposal p_j if all voters are honest is denoted by majority function M_v:

$$M_v(n_j, MPPO_j) = 1 - B(\left\lfloor \frac{n_j}{2} \right\rfloor, n_j, q_j).$$

- Similarly, for certifiers:

$$M_c(m_j, MPPO_j) = 1 - B(\left\lfloor \frac{m_j}{2} \right\rfloor, m_j, q_j);$$

Double-Player Protocol - Analysis

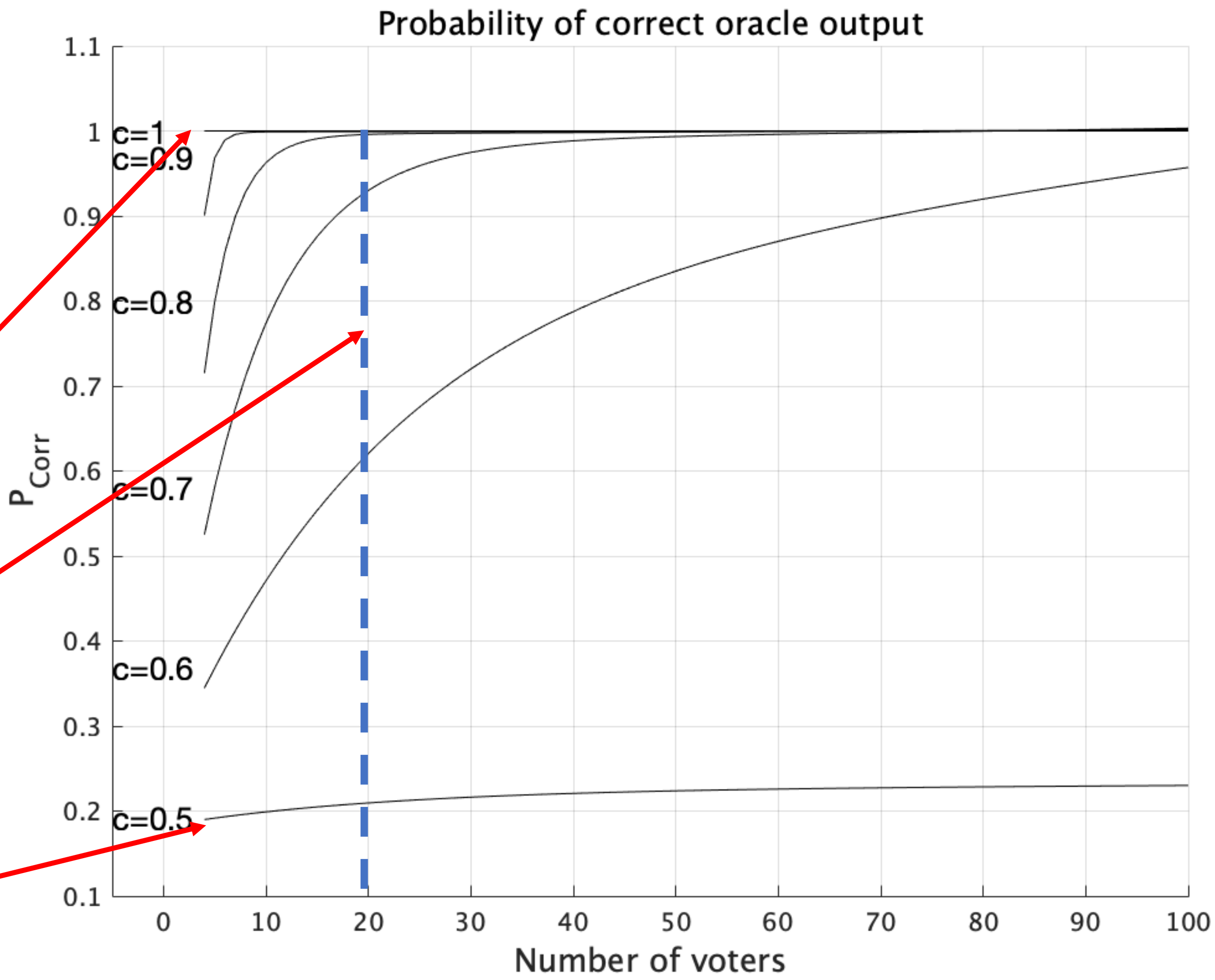
Probability of oracle correctness

P_{Corr} = probability that majority of voters answer the MPPO

At c = 1.0, everyone agrees.
Oracle will always be correct.

P_{corr} improves with more voters (ideally 10-20) as long as c > 0.5.

At c = 0.5, P_{corr} cannot be better than random ($0.5^2 = 0.25$)



Double-Player Protocol - Adversaries

Probability of successful manipulation

Assume there are in total 100 propositions, adversaries try to manipulate p_j

#voter increases → probability of successful manipulation decreases

#voter /prop.	Total stake	Adversarial stake	c	Pr(o_j=¬MPPO$_j$) (voter)
20	$2000 \times s_{max}$	0	0.8	0.0006
			0.95	$< 1 \times 10^{-9}$
		$100 \times s_{max}$ (0.05 of total)	0.8	0.0028
			0.95	$< 1 \times 10^{-6}$
		$500 \times s_{max}$ (0.25 of total)	0.8	0.1275
			0.95	0.0123
100	$10000 \times s_{max}$	0	0.8	$< 1 \times 10^{-11}$
			0.95	≈ 0
		$500 \times s_{max}$ (0.05 of total)	0.8	$< 1 \times 10^{-8}$
			0.95	≈ 0
		$2500 \times s_{max}$ (0.25 of total)	0.8	0.0168
			0.95	$< 1 \times 10^{-5}$

Pr(o_j=¬MPPO$_j$) decreases significantly from #voter = 20 to #voter = 100

s_{max}: maximum stake of voter

Double-Player Protocol - Adversaries

Probability of successful manipulation

Assume there are in total 100 propositions, adversaries try to manipulate p_j

More costly to manipulate with certifiers

Adversaries need a significant amount of stake when #voter and c are high

With the addition of certifiers, adversaries need to stake $500c_{min}$ more to manipulate certifying while $\Pr(o_j = \neg MPPO_j)$(final) is now $< 1 \times 10^{-8}$

s_{max}: maximum stake of voter
c_{min}: minimum stake of certifer
$c_{min} > s_{max}$

#voter /prop.	Total stake	Adversarial stake	c	$\Pr(o_j=\neg MPPO_j)$ (voter)	$\Pr(o_j=\neg MPPO_j)$ (certifier)
20	$2000 \times s_{max}$	0	0.8	0.0006	0.548
			0.95	$< 1 \times 10^{-9}$	$< 1 \times 10^{-7}$
		$100 \times s_{max}$ (0.05 of total)	0.8	0.0028	0.2438
			0.95	$< 1 \times 10^{-6}$	$< 1 \times 10^{-4}$
		$500 \times s_{max}$ (0.25 of total)	0.8	0.1275	≈ 1
			0.95	0.0123	0.7100
100	$10000 \times s_{max}$	0	0.8	$< 1 \times 10^{-11}$	$< 1 \times 10^{-9}$
			0.95	≈ 0	≈ 0
		$500 \times s_{max}$ (0.05 of total)	0.8	$< 1 \times 10^{-8}$	$< 1 \times 10^{-6}$
			0.95	≈ 0	≈ 0
		$2500 \times s_{max}$ (0.25 of total)	0.8	0.0168	0.8156
			0.95	$< 1 \times 10^{-5}$	0.0002

Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

Paired-Question Protocol

- Voters **answer one proposition**
- Submitter must **submit two binary propositions** p and p' and **a bond**
- Bond is returned iff the final outputs for p and p' are complementary
- p and p' should be designed to have different answers
  - Easiest method: make p' the converse statement of p
- Voters are only rewarded for answering p and p' if the final outputs of the oracle on p and p' are complementary

Paired-Question Protocol

Intuition

- Voting the same way on both p and p' yields no rewards
- Solves the lazy equilibrium problem
- Submitters are incentivized to submit pairs that clearly have opposite answers
- Each voter will believe that approximately 50% of propositions are True
- Results in stronger voter incentives

Paired-Question Protocol: Analysis

Expected Voter Payoffs

- Voter v_i receives reward on p if
  - They agree with the majority
  - Output of p and p' differ
- Voter v_i receives penalty on p if
  - They disagree with the majority
  - Output of p and p' differ
- We set $\text{Reward amount} = \text{Penalty amount} \ = 1$
- If output of p and p’ are the same, voters receive their bounties and submitter gets penalized by losing his bond

Paired-Question Protocol

Assumptions

- Every voter is able to answer any question (i.e., assigned randomly)
- Sufficiently many voters are available (ideally 10-20)
- Voter v_i's strategy directly depends on only PO_i
    - Doesn't depend on other voters' PO
    - Doesn't directly depend on the question statement
    - Precludes strategies such as "guess which of p and p' is the converse, and vote False on the converse"
    - Assumption is relaxed later when analyzing adversaries

Paired-Question Protocol: Analysis

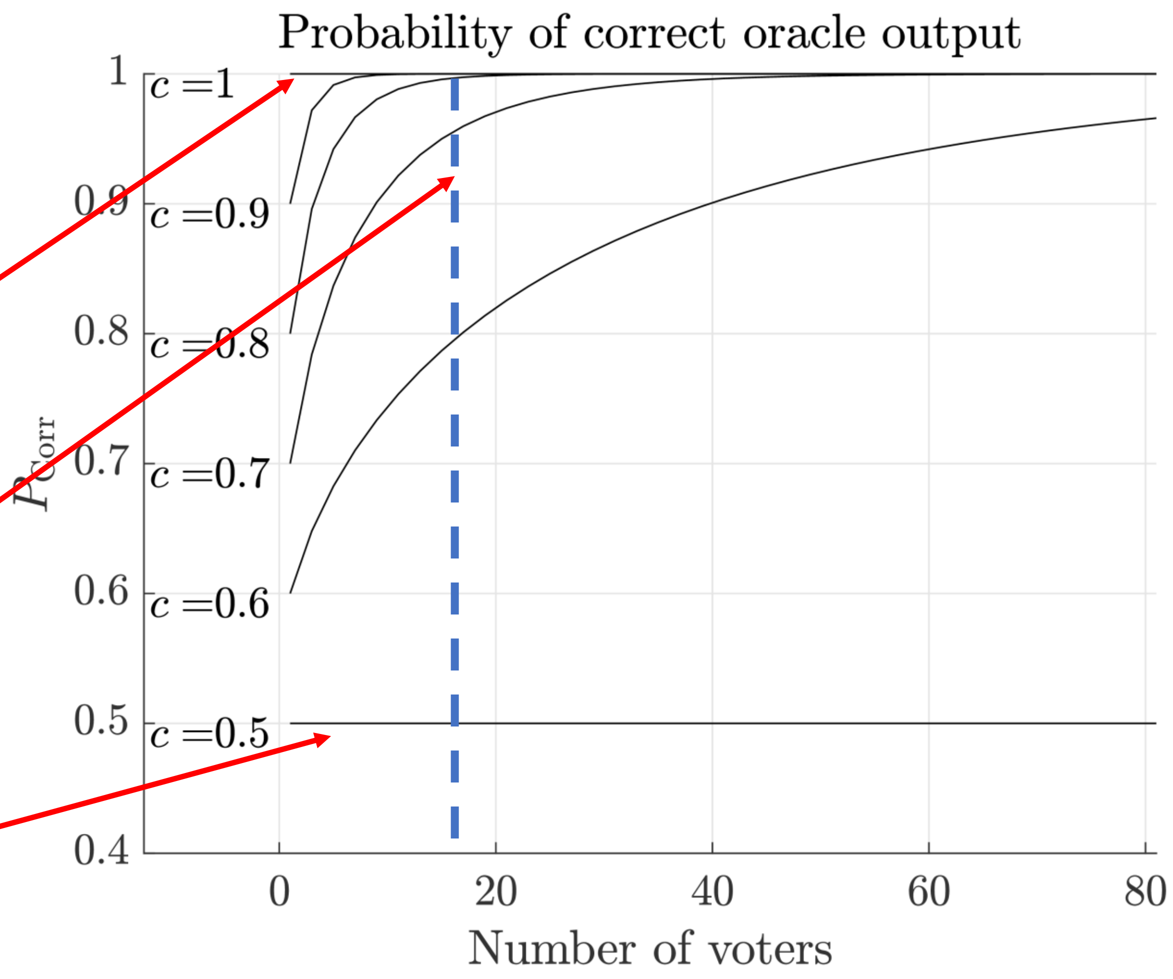
Probability of oracle correctness

P_{Corr} = probability that majority of voters answer the MPPO

At c = 1.0, everyone agrees.
Oracle will always be correct.

P_{corr} improves with more voters (ideally 10-20) as long as c > 0.5.

At c = 0.5, there is no agreement.
P_{corr} cannot be better than random.

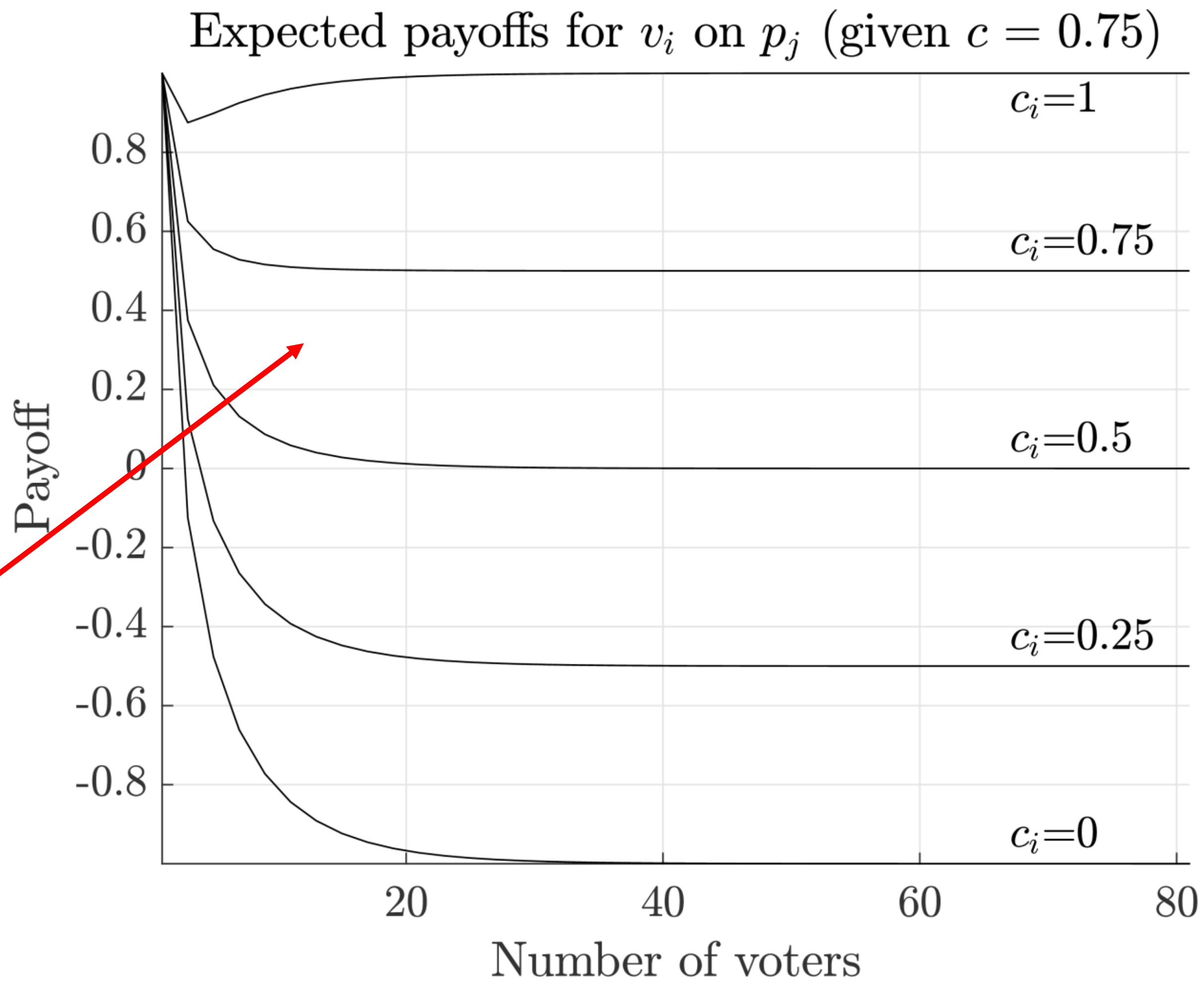


Paired-Question Protocol: Analysis

Expected Voter Payoffs

- Let $c = 0.75$ (overall degree of contention on p_j)

Payoff improves with probability of agreeing with the MPPO (c_i)



Paired-Question Protocol: Analysis

Lazy voting cannot do better than honest voting

- Everyone votes the same way → p and p' will never have different outputs
- No penalties, but no rewards either (Expected payoffs = 0)
- If you're reasonably accurate, it's much better to vote honestly!

Honest voting is a Nash equilibrium, oracle disincentivizes "lying"

- No mixed strategy can do better than pure honesty

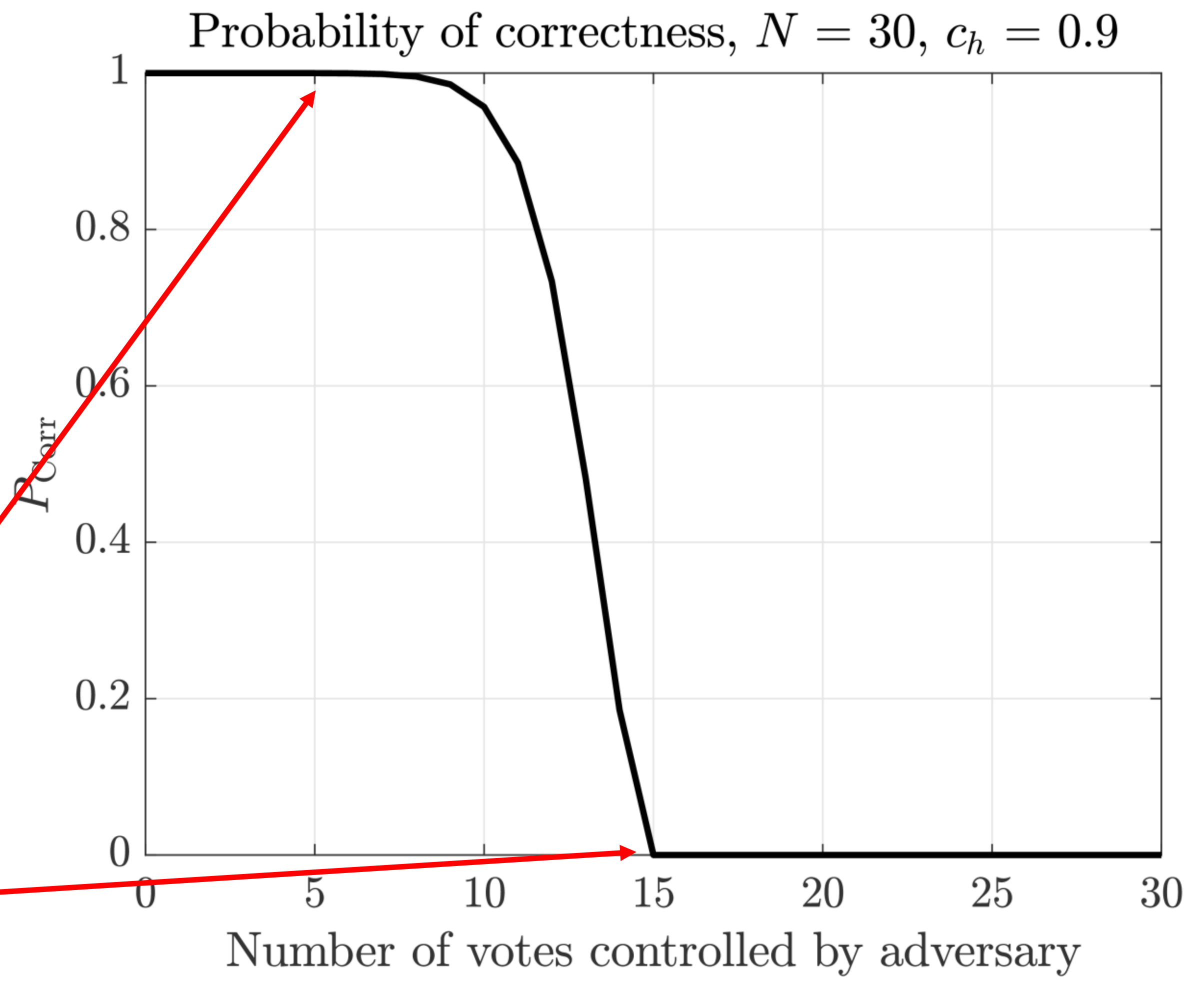
Paired-Question Protocol: Adversaries

Adversarial model

- Adversary controls n_a voters
- n_h honest voters
- $N = n_a + n_h$ = total # of voters
- $c_h = c_i$ of honest voters
- Suppose adversary tries to force incorrect output

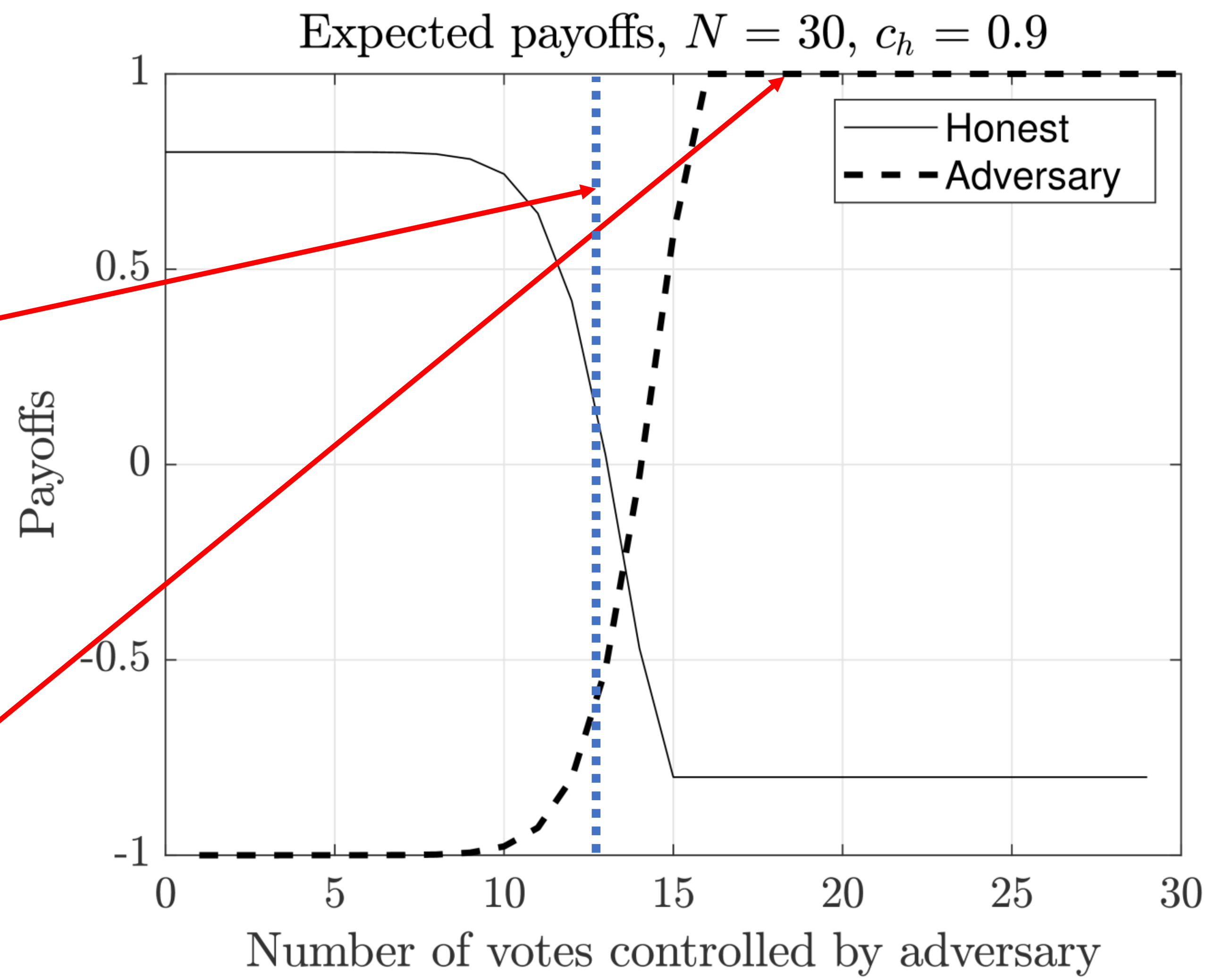
With few adversarial votes, output is MPPO with high probability

With majority of votes, adversary has complete control of output



Paired-Question Protocol: Adversaries

Expected Voter Payoffs



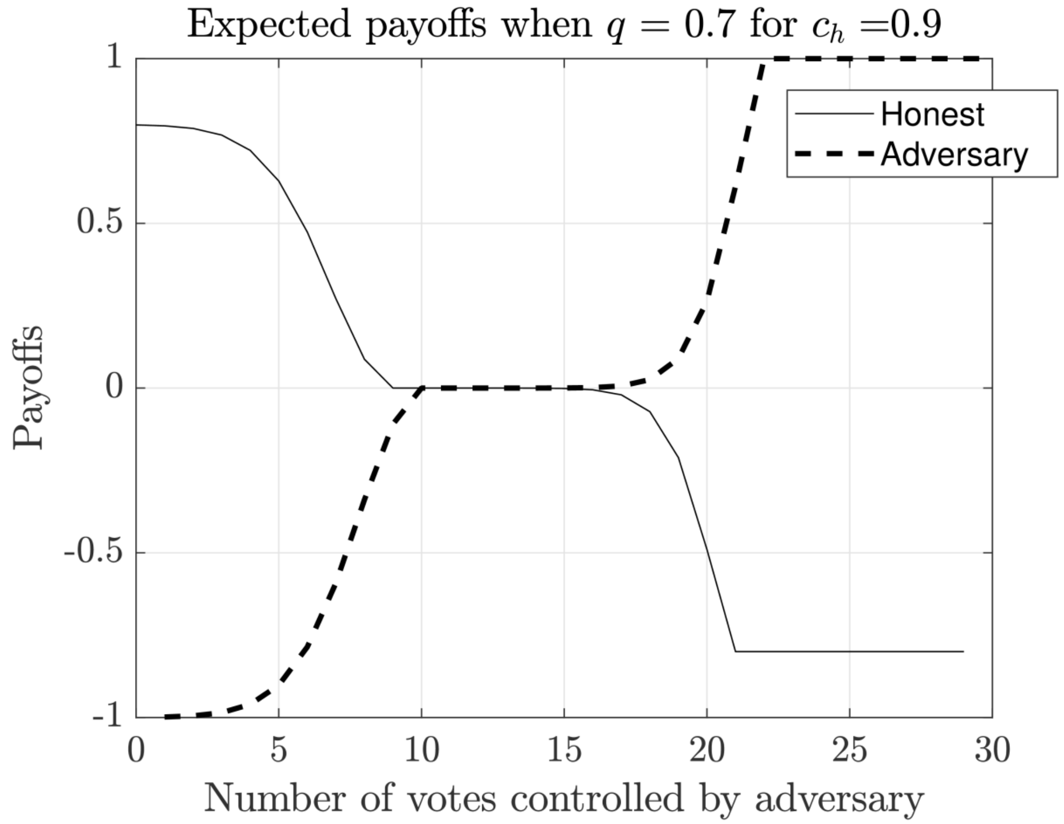
Payoffs for honesty are good as long as:

$$n_h > \frac{N}{2c_h}$$

Adversary profits when it outnumbers honest voters

Paired-Question Protocol: Adversaries

- **Quorum size (q):** minimum fraction of votes required to establish an output
- Increasing q can diminish the adversary's influence (N=30)

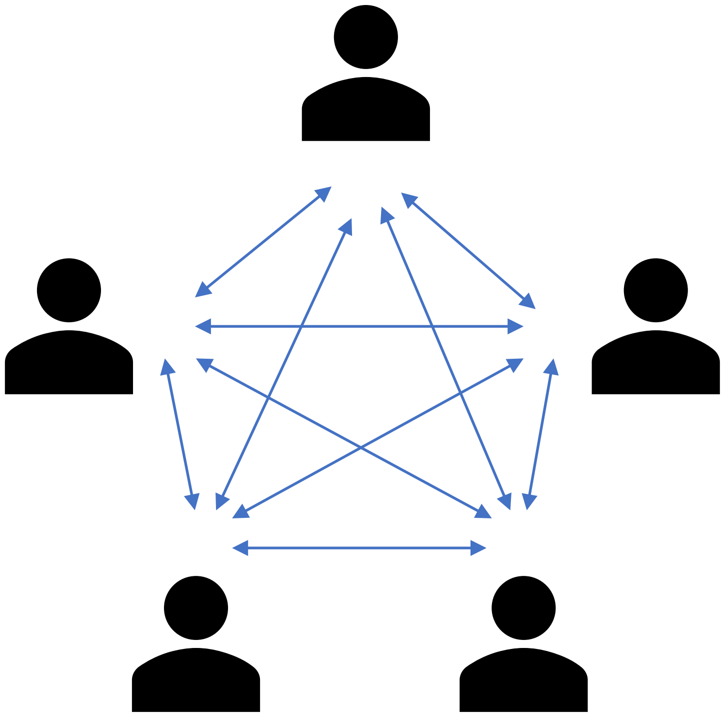


Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

Peer Prediction Protocol

- Previous approaches cannot verify if
  - the output is the "correct" answer, nor
  - distinguish between noise / honest voting
    - *i.e.,* Is MPPO wrong? Correct?
- What if the question is difficult or not likely to have a common opinion?
- Peer prediction leverages those problems by assigning scores to each opinion based on reported ***prediction on popularity***
  - The higher the score is, the more likely it is truthful



Peer Prediction Protocol - RBTS

- The protocol is based on the idea of **Robust Bayesian Truth Serum***
- RBTS is the first peer prediction mechanism that does not rely on knowledge of the common prior to provide **strict incentive compatibility** for every number of agents n > 3
- Each *agent-i* submit a binary **information report** and a numerical **prediction report** to a proposal
  - **Information report (x_i)** represents a revealed opinion of the agent
    - *i.e.*, the proposal is True/False
  - **Prediction report (y_i)** reflects the agent's belief about the distribution of information reports in the population
    - *i.e.*, 95% of all agents believe the proposal is True

* Witkowski, J., and Parkes, D. C. 2012b. A robust bayesian truth serum for small populations. In Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12).

Peer Prediction Protocol - RBTS

- Score for each *agent-i* is determined by comparing their two reports with two other randomly selected *agent-j* and *agent-k* selected as follows:
  - **Reference agent *(j = (i + 1) % n):*** whose prediction report y_j is used
  - **Peer agent *(k = (i + 2) % n):*** whose information report x_k is used
  - **n** = total # of agents

- The final RBTS score for *agent-i* is determined by summing up the **information score** and **prediction score**

Peer Prediction Protocol – adopted RBTS

- RBTS score varies by ordering of the agents therefore may not be consistent

Therefore, to make the scores more “fair”:

- *General idea*: Instead of scores based on the reports from two other agents, takes the mean of all agents excluding agent i
- Use majority of information report as x_k
- This guarantees consistency of score without changing the incentive compatibility

Peer Prediction Protocol

Overall Protocol:

- Submitters submit complementary pairs of proposals p and p'
- Voters submit an **information vote** and a **prediction** for each assigned proposal
- When the proposal is closed, score is assigned to every agent based on all the submitted reports
- Based on the average score of Truth-voting and False-voting voters, an outcome is determined for p
- Similar to paired-question protocol, voters are only rewarded for answering p and p' if the final outputs of the oracle on p and p' are complementary

Peer Prediction Protocol

Model Assumptions

- All voters are Bayesian thinkers – they maintain a belief in the form of a probabilistic distribution over several possible states on the proposal
  - *i.e.*, Picasso is the greatest modern artist – Every voter is equally confident in that there are 30% or 80% of the population agree with this statement
- All voters update their prediction belief based on private opinion PO_i
  - *i.e.,* a voter thinks that Picasso is indeed the greatest modern artist – the voter updates their belief so that that they are more confident that more of the population are in favor of this idea
- All voters are risk-neutral and seek to maximize their expected score
  - *i.e.,* if honest reporting is an equilibrium, they will report honestly

Peer Prediction Protocol

Reporting Process of an honest *voter-i*:

Before processing a proposal, ***voter-i***

- has a prediction belief PB_i on how popular the proposal is

When processing the proposal, ***voter-i***

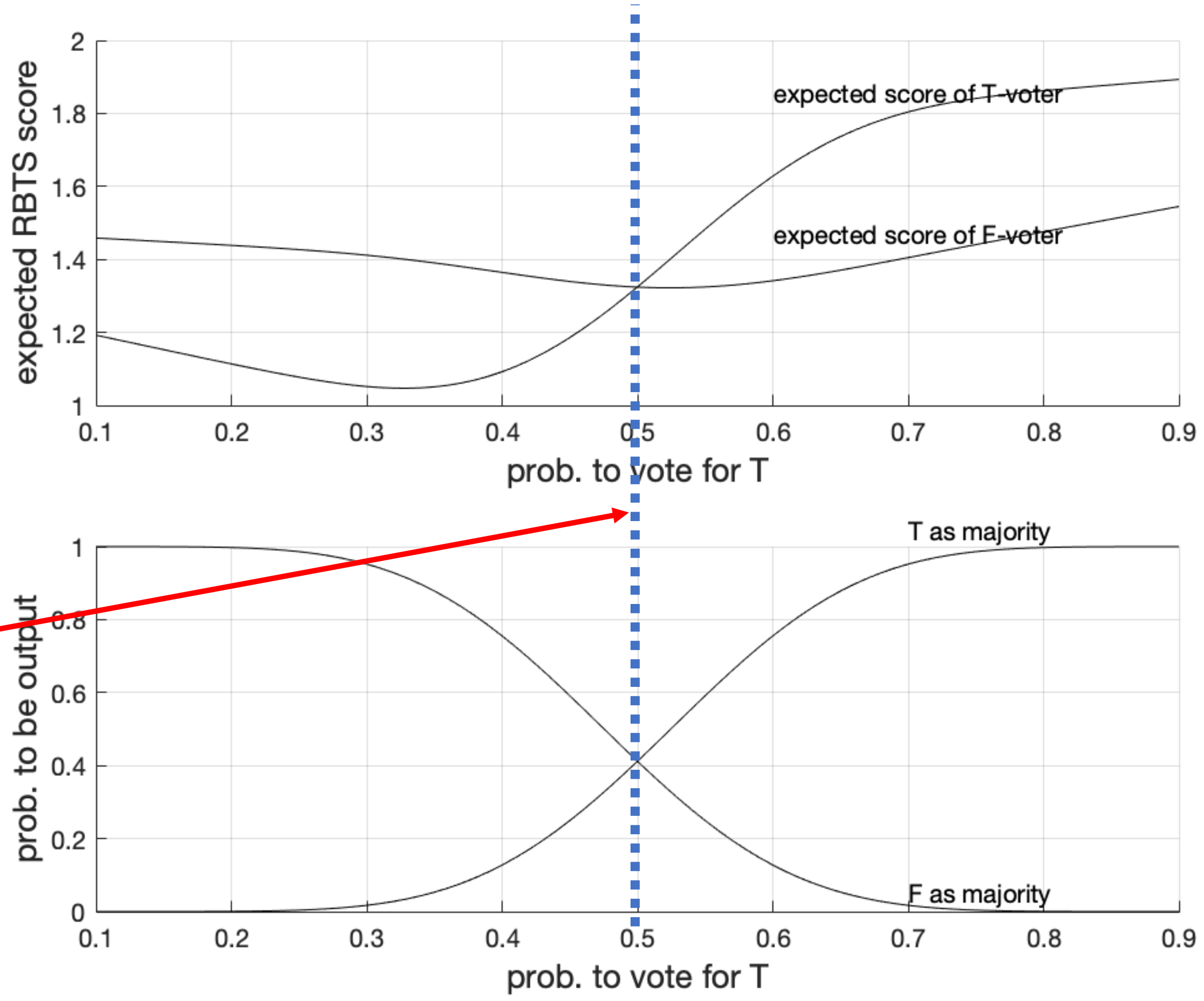
- comes up with private opinion PO_i, which is a random variable with value *{T, F}* and agrees with *MPPO* with probability *q*
- updates their prediction belief PB_i to PB_i' based on PO_i
- reports an answer v_i based on PO_i, and a prediction p_i, based on PB_i'

Peer Prediction Protocol - Analysis

When prediction belief doesn't favor either oracle outcome (*i.e.*, $PB_i(T) \approx PB_i(F)$)

- By definition of MPPO, T is MPPO when Pr(vote for T) > 0.5

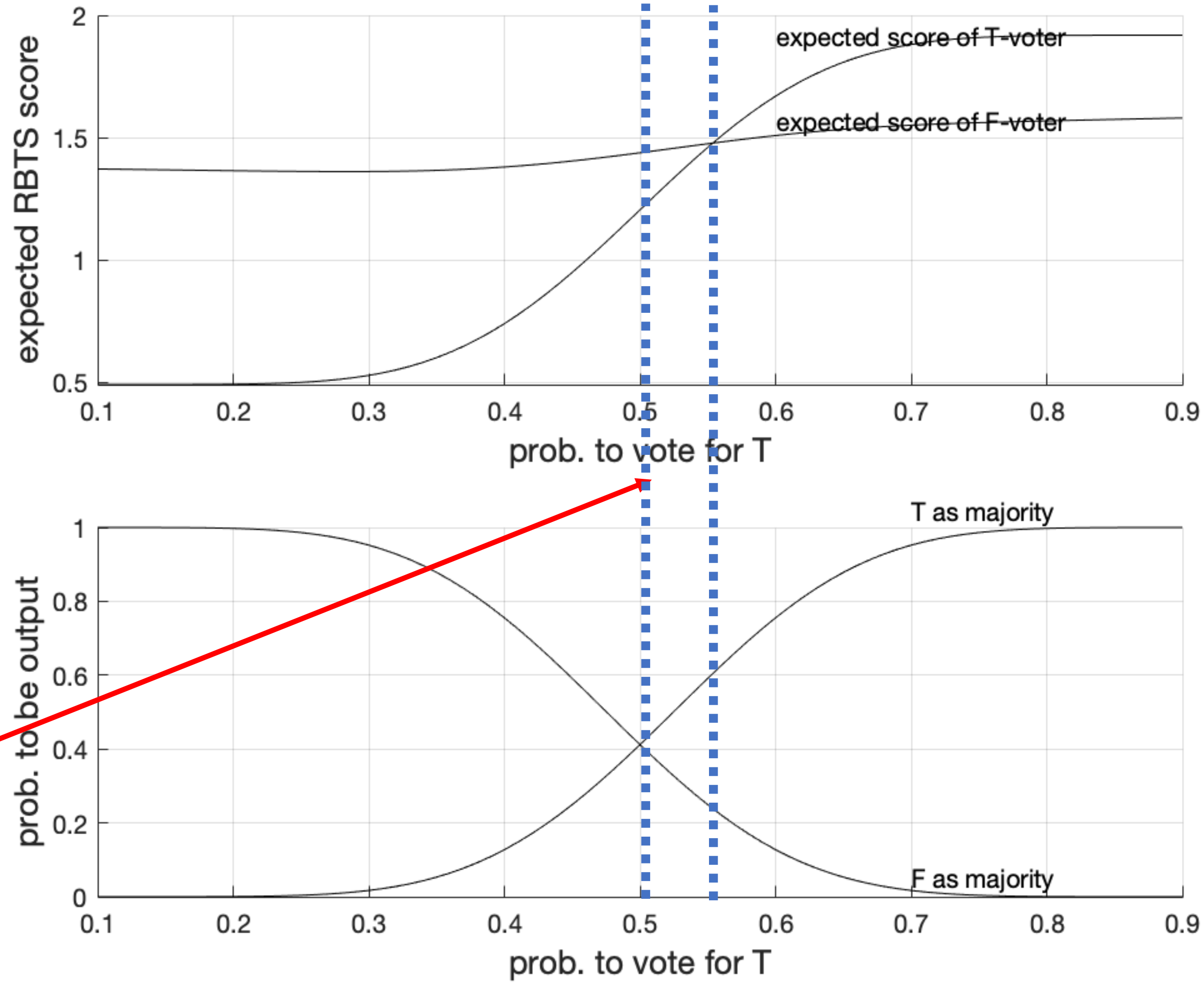
When there exists an MPPO, the expected score is higher for choosing MPPO



Peer Prediction Protocol - Analysis

When prediction belief favors T (*i.e., $PB_i(T) > PB_i(F)$*)

- By definition of MPPO, T is MPPO when Pr(vote for T) > 0.5
- The expected break-even point shifts toward an higher probability of T



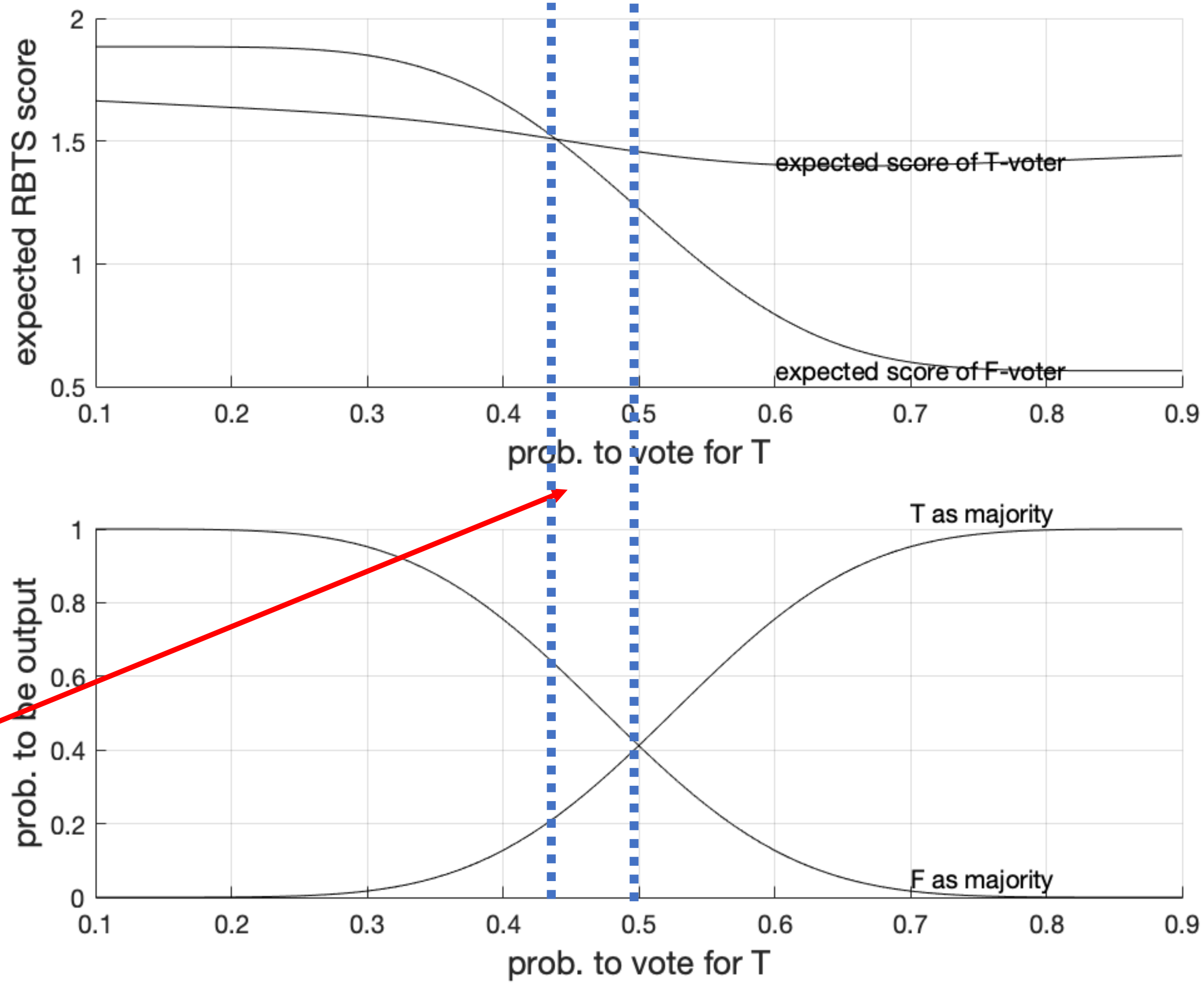
There exists an interval where the expected outcome disagrees with MPPO

Peer Prediction Protocol - Analysis

When prediction belief favors F (*i.e.*, $PB_i(F) > PB_i(T)$)

- By definition of MPPO, F is MPPO when Pr(vote for T) < 0.5
- The expected break-even point shifts toward an lower probability of T

There exists an interval where the expected outcome disagrees with MPPO



Peer Prediction Protocol - Analysis

Why shifts expected outcome away from MPPO?

- incentivizes voters to vote honestly without yielding to popularity
  - *i.e.*, even if PO_i is not the majority opinion, honest *voter-i* still expects a chance to receive higher score and hence reward
- If PB_i is biased toward outcome *o*, relax the required popularity of ¬*o*
- If PB_i is biased toward opinion ¬*o*, relax the required popularity of *o*
- Pair-question guarantees complementary outcomes of *p* and *p’*

Outline

- Introduction
- Decentralized Oracle Model
- Astraea I: Double-Player Protocol
- Astraea II: Paired-Question Protocol
- Astraea III: Peer Prediction Protocol
- Comparison

Comparison – Astraea I: Double-player protocol

Advantages:

- Incentivizes players with different incentive level to participate in the system

Disadvantages:

- Does not discourage lazy voting
- It is hard to analyze the incentive of the players
- Output only depends on the popularity

Comparison – Astraea II: Paired-question protocol

Advantages:

- Stronger guarantees and incentives for honesty than Astraea I
- Questions are balanced (approx. 50% True, 50% False)
    - Lazy equilibrium may be harder to reach
- Only powerful adversaries can manipulate the output

Disadvantages:

- Output only depends on the popularity

Comparison – Astraea III: Peer prediction protocol

Advantages:

- Takes prediction belief as a measure to break-even
- Adversarial attack is more difficult in some cases considering prediction belief

Disadvantages:

- Requires voters to be knowledgeable of the popularity
- Attack may be easier in some cases considering prediction belief

Conclusion and Future Work

- Improve on staked voting-based decentralized oracle protocol
- Honest voting is Bayes-Nash Incentive Compatible
- Future work: implementation and deployment on blockchain
  - Verify whether empirical performance matches theoretical analysis
  - Introducing varying rewards for the Peer Prediction Model
  - Introduction of reputation systems
  - Introduction of multiple adjudication (dispute) rounds and randomization