Markets and intelligence

Abhimanyu Pallavi Sudhir
Department of Computer Science
University of Warwick
Coventry – CV4 7ES
abhimanyu.pallavi-sudhir@warwick.ac.uk

1 Introduction

Before neural networks won the mandate of heaven, an AI research paradigm that had shown considerable promise was that of *market-based architectures*, i.e. AI agents that determine their output based on some internal market-based mechanism (Kwee et al., 2001; Chang et al., 2020).

More broadly speaking, there are general intuitive arguments to imagine a close analogy between markets and AI. Philosophers and psychologists have long pondered multi-agent models of the mind (Minsky, 1988); moreover, one may imagine that "any" machine learning task could in principle be solved by a market of agents solving sub-tasks with their individual reward set by the sale value of their output. There is work suggesting that markets can capture some notion of *bounded rationality*, e.g. the *Boundedly Rational Inductive Agent* (BRIA) (Oesterheld et al., 2023) and *Algorithmic Bayesian Epistemology* (Neyman, 2024) (intuitively, the efficient market hypothesis may be imagined as expressing that markets are efficient conditional on the *algorithmic information* available). In some sense, markets "aggregate" the intelligence or capacities of their individual participants.[1]

Research at the intersection of markets and AI may be split into two overarching categories[2]:

- **Markets *as* AI,** i.e. AI models and agents that work based on internal market-based algorithms, i.e. with intelligence arising as an "emergent property" of several simple or bounded agents interacting economically. This includes market-based RL (see Section 2.1.1 for an introduction), market architectures for logical uncertainty (Garrabrant et al., 2020), market architectures for bounded rationality (Oesterheld et al., 2023) and mechanism design for reward (Conitzer et al., 2024; Ge et al., 2024b).
- **Markets *of* AI,** i.e. markets comprised of AI participants. This includes economic simulations with AI populations e.g. (Zheng et al., 2021; Li et al., 2024); AI forecasters participating on prediction markets (see Sec 6) and decentralized "intelligence markets" such as BitTensor (Rao et al., 2021).

Over the course of my first year of research, I found myself often circling the following fundamental research questions:

1. *Can we design a reinforcement learning agent based on an internal market-based algorithm?* Would doing so have any advantages?
2. Some works have noted a suggestive market analogy in neural networks (Davis, 1988; Schmidhuber, 1989) and in backpropagation (Wentworth, 2018). *Can we make this more precise, and use it to construct a market-based algorithm for supervised learning applications?* Would doing so have any advantages?

[1] See e.g. Hayek on the role of markets in aggregating information (von Hayek, 1937) – or the famous parable "I, Pencil" (Leonard E Read, 1958): "... no one person, no matter how smart, could create from scratch a small, everyday pencil [yet the market makes over a billion of them each year] ...". There is also some empirical work on emergent intelligent behaviour in markets comprised of zero-intelligence traders (Gode and Sunder, 1993; Schwartz, 2008; Jamal et al., 2015)

[2] There is substantial overlap between these categories as one may imagine a market ensemble of AI agents as itself a "super-AI", which has by construction a market structure

3. Markets may be thought of as measuring devices for people's values and beliefs – *can we develop some "AIs as producers, humans as consumers" framework for giving human feedback to AIs, i.e. replacing naive RLHF?*
4. Prediction markets can easily give probabilities for verifiable or falsifiable (VF) events. But many questions we are interested in are not VF – they exist only in the "latent space". *Can we design scoring rules to incentivize usefully reporting probabilities on latent space variables?* The resulting economy may be seen as an AI that has learned some internal beliefs, and may be valuable for interpretability research (Christiano et al., 2021).

Detailed below are the specific projects I have been working on, along with any publications:

P1) Market-based RL agents. (addressing RQ1, RQ2) The work I've done so far is in the following short paper, included in Section 2.

- Abhimanyu Pallavi Sudhir and Long Tran-Thanh (2025), "Market-based architectures in RL and beyond" [Under review at AAMAS Blue Sky].

P2) Recursive Information Markets. (addressing RQ3) The main roadblock in giving proper "market-based reward" to AIs is that *information markets are inefficient*. One way I believe this could be addressed is via the Information Bazaar framework of Rahaman et al. (2024) – this will be my main focus in at least the first half of the coming year, a brief proposal is included in Section 3.

P3) Scalable Oversight Benchmark. (related to P2) Recursive Information Markets may be conceptualized as a special human feedback mechanism, or a *scalable oversight* protocol to use AI alignment terminology. Studying its effectiveness warrants developing a systematic way to evaluate scalable oversight protocols in general. I and some collaborators (Arjun Panickssery and Jackson Kaunsimaa) have developed a *benchmark for scalable oversight protocols* – the theoretical work is done and the paper draft is included in Section 4; what remains are experiments (which have been organized but are waiting to apply for OpenAI credits).

P4) Betting on what is not verifiable nor falsifiable. (addressing RQ4) For the special case where "non-VF" sentences may be represented in a first-order logic, I have developed a mechanism for betting on them called the *verification-falsification game*. A version of this paper is available on arXiv but I later found it to be unnecessarily complicated; an abridged short paper is included in Section 5.

- Abhimanyu Pallavi Sudhir and Long Tran-Thanh (2024), "Betting on what is neither verifiable nor falsifiable". arxiv.org/abs/2402.14021

P5) Consistency of AI forecasters. (related to "Markets of AI" and RQ4) One way to evaluate forecasts on sentences that are non-VF – or more generally, might take a lot of time or effort to get ground truth on – is to use *consistency checks*. I and some collaborators developed a *consistency benchmark* for evaluating AI forecasters; in particular, I developed a metric based on market arbitrage to measure inconsistency and a market-based method to correct inconsistency analogous to what Platt scaling is for miscalibration. This work has lead to a workshop paper and a full publication; an abridged version of the latter (focusing on my contribution and excluding engineering details) is included in Section 6.

- Daniel Paleka*, Abhimanyu Pallavi Sudhir*, Alejandro Alvarez, Vineeth Bhat, Adam Shen, Evan Wang and Florian Tramèr (2025), "Consistency Checks for Language Model Forecasters". Accepted to ICLR 2025.
- Abhimanyu Pallavi Sudhir*, Alejandro Alvarez, Adam Shen, and Daniel Paleka* (2024), "Consistency Checks for Language Model Forecasters" *Workshop paper, accepted to:* Agentic Markets Workshop at ICML 2024; NextGenAISafety Workshop at ICML 2024; Oxford ELLIS Robust LLMs Workshop 2024

1.1 Reflections

Skills and professional development. I've developed a number of valuable connections this year from research collaborations. Both P3 and P5 resulted from people I met through the Berkeley Supervised Program for Alignment Research (SPAR). In general I've also gotten a much better "lay of the land" of AI research, and computer science in general, compared to a year ago.

Managing work and communication. I generally communicate with my supervisor via email and schedule a face-to-face meeting at least once or twice a month; and with my other collaborators on Discord and Slack.

Year plan. In terms of research proper, my plan for the first half of next year is to: (1) focus primarily on P2 (2) work on the stuff I mentioned in the future work section of P1 and (3) get our P3 work so far published in at least a workshop or short paper.

Next year, I will be acting as a research lead at the AI Safety Camp where I have found some competent collaborators to contribute to P2. There is also a startup that works on AI forecasting in the BitTensor space that has approached me out of interest in P5, and I will likely be doing some internship kind of work with them next year.

Once I've made some solid progress on P2 I am also interested in implementing recursive markets at scale for stuff like web search and community notes at a relevant company – I know Perplexity has been thinking on similar lines, and Google has a Market Algorithms team that is working on stuff at the intersection of LLMs and markets.

2 Market-based agents in RL and beyond

Abstract

Market-based agents refer to reinforcement learning agents which determine their actions based on an internal market of sub-agents. We introduce a new type of market-based algorithm where the state itself is factored into several axes called "goods", which allows for greater specialization and parallelism than existing market-based RL algorithms. Furthermore, we argue that market-based algorithms have the potential to address many current challenges in AI, such as *search*, *dynamic scaling* and *complete feedback*, and demonstrate that they may be seen to generalize neural networks; finally, we list some novel ways that market algorithms may be applied in conjunction with Large Language Models for immediate practical applicability.

2.1 Introduction

Before neural networks won the mandate of heaven, an AI research paradigm that had shown considerable promise was that of *market-based architectures*, i.e. AI agents that determine their output based on some internal market-based mechanism Kwee et al. (2001); Chang et al. (2020).

There are general intuitive arguments that motivate such a line of research. Philosophers and psychologists have long pondered multi-agent models of the mind Minsky (1988); moreover, one may imagine that "any" machine learning task could in principle be solved by a market of agents solving sub-tasks with their individual reward set by the sale value of their output. There is work suggesting that markets can capture some notion of *bounded rationality*, e.g. the *Boundedly Rational Inductive Agent* (BRIA) Oesterheld et al. (2023) and *Algorithmic Bayesian Epistemology* Neyman (2024). In some sense, markets "aggregate" the intelligence or capacities of their individual participants.[3]

[3]See e.g. Hayek on the role of markets in aggregating information von Hayek (1937) – or the famous parable "I, Pencil" Leonard E Read (1958): "... no one person, no matter how smart, could create from scratch a small, everyday pencil [yet the market makes over a billion of them each year] ...". There is also some empirical work on emergent intelligent behaviour in markets comprised of zero-intelligence traders Gode and Sunder (1993); Schwartz (2008); Jamal et al. (2015)

"Market-based architectures" can be made concrete in the case of reinforcement learning (RL), where the majority of work in this area lies (see 2.1.1 for a brief summary). For example in the *Hayek machine* Baum (1999) and its derivatives, the setting is a Markov Decision Problem (MDP) and there is a single resource, the "right to act and collect reward", that is traded between sub-agents. At each time step, this resource is sold to the highest-bidding sub-agent, who performs some action that modifies the state of the world and collects the reward defined by the MDP.

In this paper, we argue that market-based agents represent an underexplored and promising niche, *especially* in the context of recent advancements in language models (LLMs), and have the potential to address a range of present challenges in contemporary AI research. Specifically, we make the following claims and contributions:

Theoretical framework for market-based agents. We present two general frameworks for market-based RL agents: (1) the "deep market" (Def 2.1), where a single good, the *state*, is passed through a sequence of transacting agents, and (2) the "wide market" (Def 2.2), in which the state space itself is partitioned into several factors called *goods*. The deep framework is not much of a departure from existing algorithms such as the Hayek machine, and can be applied to any Partially Observed Markov Decision Process (POMDP); while to our knowledge the wide framework is original to us, and is a generalization of the deep framework which better mirrors the success of real-world markets allowing for greater specialization and parallelism. A Python library for creating and applying market-based algorithms to a variety of tasks will be released after publication.

Markets, neural networks and backpropagation. We demonstrate that these market-based agents can in principle be applied even to basic supervised learning tasks such as classification, and that neural networks (though not backpropagation or gradient descent) emerge as a special case of such algorithms. Furthermore we generalize the result in Wentworth (2018) to wide markets, demonstrating a suggestive relationship between backpropagation and markets at equilibrium.

Search, complete feedback and alignment. We claim that markets can address several present problems in AI research, specifically: they are a natural framework for *search*, their scale or depth can be *dynamic* rather than fixed, and they allow *complete feedback* Demski (2024), a property widely regarded as valuable in AI alignment.

Markets and LLMs. We present some novel ways in which market algorithms might be applied in conjunction with LLMs to address their limitations: they can be used for developing reasoning models like `o1`, and one may use LLMs to develop "information markets" that can in turn be used to improve human feedback mechanisms for training LLMs.

2.1.1 Related work

Market-based RL. The majority of early work in this area has focused on market-based *reinforcement learning* (RL) algorithms. The pioneering work in this domain consists of Holland's *Learning Classifier Systems* or "bucket brigade" Holland (1985) in rule-based systems, where condition-action agents ("classifiers") bid to post messages (which can be actions described in some language) onto a global message board. Improvements to this paradigm were made by Schmidhuber Schmidhuber (1987; 1989) who allowed agents to determine their own bids and imposed credit conservation, and by Baum Baum (1999) who further strictly enforced property rights, resulting in a much more familiar set-up called the "Hayek Machine". The Hayek machine, which can be applied to any Markov Decision Process (MDP), was subsequently extended to POMDPs by Kwee et al. (2001) by adding external memory, and more recently Chang et al. (2020) modified the framework to use Vickrey auctions and prove that a Nash equilibrium of the market produces a globally optimal policy.

Bounded rationality and markets. Other, more recent work in this area includes: *Logical Induction* Garrabrant et al. (2020), an algorithm that assigns probabilities to mathematical sentences based on their prices in a prediction market that (roughly speaking) pays off when a sentence is proven; and the *Boundedly Rational Inductive Agent* (BRIA) Oesterheld et al. (2023) which solves finite decision problems by assigning it to the highest-bidding trader, similar to in market-based RL. The key insight of these works is that markets are useful for

modeling *boundedly rational agents*. The main results of each work – the fact that the logical inductor cannot by dominated by any polynomial-time trader, and the "boundedly rational inductive agent criterion" in the latter paper – are specific and precise formulations of the Efficient Market Hypothesis Neyman (2024).

Markets and neural networks. A specific equivalence between classifier systems and neural networks has been studied in the *Neural bucket brigade* Davis (1988); Schmidhuber (1989), although this does not consider backpropagation. A suggestive analogy between markets and backpropagation is discussed in Wentworth (2018), though only for the case of a strictly sequential, unit-width market like that in Def 2.1. In our work we make this more precise and generalize it to 2.2

2.2 Market algorithms

Setting (POMDP). We assume a ususal POMDP setting, with a state space $\mathcal{S}$, action space $\mathcal{X}$, transition probability $\mathbf{P}(s' \mid s, x)$ (which allows us to treat actions as stochastic functions i.e. $x(s) \sim \mathbf{P}(s' \mid s, x)$), reward function $\mathbf{R}(s, x, s')$, and observation distribution $\omega(s) \sim \mathbf{O}(\omega \mid s)$ over a set of observations Ω. A policy is a map $\alpha : \Omega \to \mathcal{X}$, and the process proceeds as per usual.

Mirroring Kwee et al. (2001) and similar to Belief-MDP formulations, we can extend the state and message spaces by taking the cartesian product space with a message space str which is always preserved by ω; this gives the policy a "memory", or in terms of markets, creates informational goods.

The first algorithm we describe is Def 2.1: here, agents (out of a possibly infinite collection of agents $\mathcal{A}$) bid at each time step for the *right to act and collect reward*, and the highest-bidding agent is chosen to act. The parameters of this algorithm are the wealths of each agent $w[\alpha]$. These are trained by the training loop Capitalism: at each step, the agent pays its bid to the previous agent, collects the reward generated by its actions and receives the bid of the next. This means that at equilibrium, each agent is incentivized to bid the value function, and perform the action with maximum Q-value.

Definition 2.1 (Deep market)**.** Assume a POMDP setup, and let $\mathcal{A}$ be a collection of "agents", which are (stochastic) maps $\alpha : \Omega \to \mathcal{X} \times \mathbb{R}$. The first component $\hat{\alpha} : \Omega \to \mathcal{X}$ of an agent is called its *action*, the second component $\alpha_b : \Omega \to \mathbb{R}$ is called its *bid*. The market algorithm then proceeds as in Algorithm 1.[4]

Some details have been ignored. $\mathcal{A}$ will usually be infinite and so tables like $w[\alpha]$ cannot simply be indexed on it: instead, $\mathcal{A}$ must be countably enumerated and added to the economy one-by-one in the training loop with each agent being endowed with some allowance. To prevent holdout problems, one may impose a small fixed "rent" on α^* at each training step i.e. $w[\alpha^*] \leftarrow (1-\varepsilon)w[\alpha^*]$. A simple first-price auction is shown for simplicity, and may be replaced with a Vickrey auction in line with Chang et al. (2020). One may also replace the explicit reward function $\mathbf{R}$ with a class of "consumers" $\mathcal{C}$ who place bids upon desirable states, which may be a useful formulation for reinforcement learning from diverse human feedback[5].

Definition 2.1, which subsumes existing market-based RL, already illustrates one of the key defining features of markets recognizable to any student of economics: markets serve not only to *select* (via market competition) the best process to achieve a task, but also to *distribute* a complex task among agents which are individually much too weak or uninformed to complete the entire task[6]. This means that the collection of actions $\mathcal{A}$ can be a class of "simple" agents, so that enumerating $\mathcal{A}$ can quickly find many valuable agents.

[4] For training, this may be executed in multiple episodes with different initial state s, either with finite episodes or in parallel (with shared wealth variables across running instances).

[5] see Conitzer et al. (2024); Ge et al. (2024b) for a primer on this area

[6] see e.g. von Hayek (1937) – or the famous parable "I, Pencil" Leonard E Read (1958): "... no one person, no matter how smart, could create from scratch a small, everyday pencil [yet the market makes over a billion of them each year] ...".

Algorithm 1 Deep market

procedure MARKET ▷ Forward pass
  parameters: $w[\alpha] \in \mathbb{R}$ ▷ wealths of each $\alpha \in \mathcal{A}$
  input: $\omega \in \Omega$
  $b[\alpha] \leftarrow \min(\alpha_b(\omega), w[\alpha])$ for $\alpha \in \mathcal{A}$ ▷ Cap bids by wealth
  $x \leftarrow \hat{\alpha}^*(\omega)$ ▷ Determine action
  return x, α^*
end procedure
procedure CAPITALISM ▷ Training loop
  Initialize agent wealths $w[\alpha] \in \mathbb{R}$ for each $\alpha \in \mathcal{A}$
  Initialize original owner of the world α^*
  Initialize state $s \in \mathcal{S}$
  while $t \in \mathbb{N}$ **do**
    $\omega \leftarrow \omega(s)$ ▷ Generate observation
    $\alpha^*_{\text{prev}} \leftarrow \alpha^*$
    $x, \alpha^* \leftarrow \text{MARKET}(\omega)$
    $w[\alpha^*] \leftarrow w[\alpha^*] - b[\alpha^*]$ ▷ Pay bid
    $w[\alpha^*_{\text{prev}}] \leftarrow w[\alpha^*_{\text{prev}}] + b[\alpha^*]$ ▷ to previous owner
    $s_{\text{prev}} \leftarrow s$
    $s \leftarrow x(s)$ ▷ Transition state
    $w[\alpha^*] \leftarrow w[\alpha^*] + \mathbf{R}(s_{\text{prev}}, x, s)$ ▷ add reward to wealth
  end while
end procedure

Specifically, Def 2.1 exploits modularity of *action*, where the state can be transformed one step at a time. There is however another form of modularity, missed by all existing market-based RL algorithms, which we may call modularity of *state*, and is crucial to the success of real-world markets: here, agents are not constantly transacting the whole "state of the world": instead, the state of the world is decomposed into several components, called *goods*[7]: $\mathcal{S} = \mathcal{S}_1 \oplus \cdots \oplus \mathcal{S}_n$. For instance, $s_1 \in \mathcal{S}_1$ might represent the quantity of iron ore in the world. Agents bid for small quantities of each good; no agent owns the whole world, and does not have to bother performing a valuation of the whole world. The "state of the world" may be recovered as the vector sum of all agents' holdings.

At least two new difficulties are introduced by considering markets of multiple divisible goods:

General equilibrium theory. Allocating goods is no longer as easy as an auction, because agents might have joint demand schedules for goods that are complementary or substitute to each other. The problem of matching buyers and sellers in this setting is the domain of "General Equilibrium Theory" in economics, where there are models such as the Fisher market and the Arrow-Debreu exchange market Arrow and Debreu (1954); McKenzie (1959). Computing the equilibrium in these models is non-trivial and often intractable Chen et al. (2009); Chen and Teng (2009).

Property rights in POMDPs. A more subtle difficulty lies in the fact that we want to divide the state $s \in \mathcal{S}$, which is not directly observed, among bidding agents (so that each of their actions only transform their respective portions of the state, i.e. their properties), but the agents only submit demand schedules over $\omega \in \Omega$. It is not obvious how to map a decomposition of a vector $\omega(s)$ back onto s.

Both of these have to do with specific questions of how buyers and sellers meet and match in real markets, i.e. having to do with *institutions* such as property rights and mechanism design. These questions are out of scope for us, and we abstract them away by postulating some effective equilibrium computation algorithm $\text{Equ}(\omega, \alpha^1_b, \ldots \alpha^m_b) = (\mathbf{p}, \omega[\alpha^1], \ldots \omega[\alpha^m])$ i.e. which takes the total perceived quantity of goods in the world Ω and each agent's

[7] $\oplus$ denotes the direct sum of vector spaces, which is a Cartesian product equipped with a pointwise vector addition operator

Algorithm 2 Wide market

procedure MARKET ▷ Forward pass
  parameters: $w[\alpha] \in \mathbb{R}$ ▷ wealths of each $\alpha \in \mathcal{A}$
  input: $\omega \in \Omega$
  ▷ Cap bids by budget
  $b[\alpha] \leftarrow \lambda s : \min(\alpha_b(s), w[\alpha])$ for all $\alpha \in \mathcal{A}$
  ▷ Compute equilibrium prices and allocations
  $\mathbf{p}, \omega'[\alpha^1], \dots \omega'[\alpha^n] \leftarrow \text{Equ}(\sum \omega[\alpha], b[\dots])$
  $x[\alpha] \leftarrow \hat{\alpha}(\omega'[\alpha])$ for all $\alpha \in \mathcal{A}$ ▷ Determine actions
  return $x[\alpha^1], \dots x[\alpha^n], \mathbf{p}, \omega'[\alpha^1], \dots \omega'[\alpha^n]$
end procedure
procedure CAPITALISM ▷ Training loop
  Initialize agent wealths $w[\alpha] \in \mathbb{R}$ for each $\alpha \in \mathcal{A}$
  Initialize agent properties $\omega[\alpha] \in \Omega$ for each $\alpha \in \mathcal{A}$
  Initialize state $s \in \mathcal{S}$
  while $t \in \mathbb{N}$ **do**
    $\omega \leftarrow \omega(s)$ ▷ Generate observation
    $\dots \leftarrow \text{MARKET}(\omega)$ ▷ get all outputs
    $w[\alpha] \leftarrow w[\alpha] - \mathbf{p} \cdot \omega'[\alpha]$ for all α ▷ Charge buyers
    $w[\alpha] \leftarrow w[\alpha] + \mathbf{p} \cdot \omega[\alpha]$ for all α ▷ Pay sellers
    $s[\alpha] \leftarrow s_{x[\alpha]}(\omega[\alpha])$ for all α ▷ Calculate property rights
    $s'[\alpha] \leftarrow x[\alpha](s[\alpha])$ for all α ▷ Transform goods
    $w[\alpha] \leftarrow w[\alpha] + \mathbf{R}(s[\alpha], x[\alpha], s'[\alpha])$ for all α
    $s \leftarrow \sum s'[\alpha]$ ▷ Update state
  end while
end procedure

valuation function $\alpha_b^i : \Omega \to \mathbb{R}$, and returns a price vector $\mathbf{p} \in \Omega$ and allocations to each agent $\omega[\alpha^i] \in \Omega$, such that (in line with a Walrasian equilibrium with quasilinear utilities Miller (2006)):

- $\omega = \sum \omega[\alpha^i]$ (the full quantity is allocated)
- $\mathbf{p} \cdot \omega[\alpha^i] \leq \alpha_b^i(\omega[\alpha^i])$ for all α^i (no agent pays for what it doesn't value), and
- $\omega[\alpha^i] = \arg\max_{\omega' \in \Omega} \alpha_b^i(\omega') - \mathbf{p} \cdot \omega'$ for all α^i (each agent gets a utility-maximizing bundle at the given price).

Definition 2.2 (Wide market)**.** Everything from the POMDP setup and the agent type in Def 2.1 remains the same; except that $\mathcal{S}$ and Ω are now vector spaces with each vector called a *goods bundle*. Further, we have action spaces $\mathcal{X}_\omega$ indexed by $\omega \in \Omega$ such that (1) for any $x \in \mathcal{X}_\omega$, there is an "exercised property right" denoted $s_x(\omega) \in \mathcal{S}$ such that $\omega(s_x(\omega)) = \omega$ and $x(s) = x(s_x(\omega)) + (s - s_x(\omega))$ (i.e. each agent's actions transform only the goods they own) and (2) there is an injective map $\xi : \mathcal{X}_{\omega_1} \times \mathcal{X}_{\omega_2} \to \mathcal{X}_{\omega_1+\omega_2}$ such that $\xi(x_{\omega_1}, x_{\omega_2}) = x_{\omega_1}(s_{x_{\omega_1}}(\omega_1)) + x_{\omega_2}(s_{x_{\omega_2}}(\omega_2)) + (s - s_{x_{\omega_1}}(\omega_1) - s_{x_{\omega_2}}(\omega_2))$ (this is used to combine actions by different agents). The agents now have dependent type signatures $\alpha : (\omega : \Omega) \to \mathcal{X}_\omega \times \mathbb{R}$, and the transition probability $x(s) \sim \mathbf{P}(s' \mid s, x)$, reward function $\mathbf{R}(s, x, s')$ and observation distribution $\mathbf{O}(\omega \mid s)$ are now interpreted as applying to "private property", i.e. to any goods bundle in their respective domains, rather than to the whole state, e.g. each action $x(s)$ defines a *production function* that transforms one goods bundle into another, and α_b is an agent's *valuation function* over all possible bundles, i.e. how much it is willing to pay for a particular perceived bundle (if it's differentiable, then $\nabla \alpha_b(\omega)$ can be interpreted as the price vector it offers). The market algorithm proceeds as in Algorithm 2 .

Computing prices via backpropagation. Though it remains to be seen how standard RL problems might be cast in this setting, we expect implementations of this algorithm to be much more effective than of Def 2.1, as it allows us to use simpler and more specialized agents in the collection $\mathcal{A}$. In particular, these agents do not need to estimate the valuations of the whole world, but only of their particular input goods.

This last point can be illustrated particularly nicely when the setup is an MDP, and rewards are replaced by consumers – here, $\Omega = \mathcal{S}$ and $\omega(s) = s$, so $\hat{\alpha} : \mathcal{S} \to \mathcal{X}$ can directly be interpreted as a production function $\hat{\alpha} : \mathcal{S} \to \mathcal{S} := \hat{\alpha}(s)(s)$. Then if the agent can estimate what the market prices of its output goods will be (e.g. if prices are sufficiently stable that it makes sense to speak of a "prevailing price" $\mathbf{p}$), then it can compute its offered prices via the chain rule – where $D\hat{\alpha}$ denotes the Jacobian:

$$\nabla \alpha_b = D\hat{\alpha} \cdot \mathbf{p} \tag{1}$$

i.e. once the market "graph" is fixed, prices can be computed by simply backpropagating consumer bids through the graph. This generalizes the result in Wentworth (2018), which demonstrated this relationship for deep markets only.

2.3 Motivation for market-based AI

In this section, we describe how markets could potentially generalize neural networks and provide a more "flexible training mechanism". Although we have presented our algorithms in an RL setting, they can even be applied to supervised learning tasks by treating internal representations as "states". To see this, it is illustrative to see how a simple neural network can be recast as a market.

Theorem 2.3 (Neural networks as markets)**.** *Consider a fully-connected neural network $f : \mathbb{X} \to \mathbb{Y} := f_n \circ \dots f_1$ where each $f_i : \mathbb{R}^{m_{i-1}} \to \mathbb{R}^{m_i}$ is a layer, i.e. a function of the form $f_i(\mathbf{x}) = \sigma(W_i\mathbf{x} + \mathbf{b}_i)$ where σ is a ReLU activation. Then there is a deep market whose forward pass performs the same operation as f.*

Proof. The construction is straightforward. Define the state space $\mathcal{S} := \left[\bigoplus_{1 \le i \le n} \mathcal{S}_i\right] \oplus \mathbb{Y}$ (with each $\mathcal{S}_i := \mathbb{R}^{m_i}$), with $\omega : \mathcal{S} \to \Omega$ discarding only the last component $\mathbb{Y}$ which represents the true label which is unchanged under all actions. Each $\mathcal{X}_\omega = \{(W_i, \mathbf{b}_i) : W_i \in \mathbb{R}^{m_i \times m_{i-1}}, \mathbf{b}_i \in \mathbb{R}^{m_i}\}$ if $\omega \in \mathcal{S}_{i-1}$ and empty if no such i exists, and an action $x = (W_i, \mathbf{i})$ acts on $s \in \mathcal{S}_{i-i}$ as $x(s) = \sigma(W_i s + \mathbf{b}_i)$. The reward $\mathbf{R}(s, x, s') = -\ell(s', s'_{\mathbb{Y}})$ for some loss function ℓ if $s' \in \mathcal{S}_n$ and 0 otherwise. Finally, let $\mathcal{A}$ consist of all constant maps to $\mathcal{X}_\omega$ and endow non-zero wealth to only those agents whose actions' parameters are the same as some f_i. □

While the market model, i.e. the forward pass, in Theorem 2.3 is the same as the neural network, the training mechanism is Capitalism (as defined in Algorithm 1) rather than backpropagation. Detailed below are some strengths of this we anticipate:

Search and dynamic scale. *Reasoning* is widely touted as a key limitation of current-day LLMs Huang and Chang (2023); McLau (2024). A view held by some researchers including Yann LeCun Yann LeCun (2023), is that this is due to the fact that "[neural networks] produce their answers with a constant number of computational steps between input and output", independent of the complexity required by the problem. Some proposed architectures that avoid this limitation include dynamic neural networks Han et al. (2021), adaptive computation time Graves (2017) as well as chain-of-thought based methods such as `o1` OpenAI (2024). Markets provide a principled alternative, as here the structure of the computational graph is itself learned, and different agents and structures may be active for different inputs.

Complete feedback. Informally speaking, markets allow *any* aspect of the system to be optimized. Formal results are needed to make this statement precise, but intuitively: any aspect of a learner, such as any hyperparameter, or meta-learning, can be changed by adding a trader to the market who will profit if his changes are beneficial and the incentives are correctly designed. This is suggestive of the notion of "complete feedback" in AI alignment research, which refers to the property that "the trainer can enact *any* modification they'd like to make to the system" Demski (2024), and is viewed as a desirable characteristic of an AI system for alignment.

2.4 Practicality and future work

We have presented two general frameworks for market-based RL agents, and illustrated that they may be seen to generalize neural networks in a supervised learning setting, albeit with a more flexible training mechanism that holds promise to address the limitations of current-day AIs with respect to reasoning and alignment properties.

Despite these theoretical strengths, our algorithm as described faces practical challenges to implement in real-world machine learning tasks: blindly enumerating large classes of even simple agents is inefficient (compared to backpropagation, where the search is guided by gradients), and we have to store many more agents in memory than the "size" of the network (the exact number depending on the rule we use to prune low-wealth agents). Some potentially promising approaches include:

- "integrated" models which perform backpropagation by default but intelligently resort to markets when it expects changing the network structure to be worthwhile
- having each agent simultaneously learn its parameters via backpropagation
- decentralized set-ups, perhaps using frameworks such as BitTensor Rao et al. (2021), allowing traders to be shared across machine learning applications.

Markets of LLMs. A more immediately feasible practical application is to develop *markets comprised of LLMs*, i.e. where $\mathcal{A}$ is a collection of LLM agents. For instance, one may let $\mathcal{S} = \Omega$ be a message space, and let actions act on s by appending some "chain-of-thought item" to the current message. The final reward is determined by human feedback, and intermediate rewards are determined by bids. Such a market would function as a "reasoning model" analogous to `o1`.

The extension to a wide market is also immediate: agents may bid for the right to read only a portion of the message space[8] – this allows for more precise credit assignment to contributions by different agents, and may be understood as to *trees-of-thought* Yao et al. (2024) what `o1` is to *chain-of-thought*.

Theoretical work. The most pressing need at present is for *precise theoretical results* on the effectiveness of market-based algorithms. An immediate research agenda includes the following:

- Determining **convergence and optimality conditions** of market algorithms; in particular, generalizing the "coverage" results of BRIA Oesterheld et al. (2023) and logical induction Garrabrant et al. (2020) to demonstrate that the market can "eventually test every possible policy" (and so will give a fair chance to the best one) depending on how wealth endowments are allocated to each agent.
- A **Learning Theory** perspective on markets and the wealth update mechanism. In particular, (real-world) markets appear to have many useful features from an alignment standpoint, such as their inherent capacity for online learning and generalization even from imperfect reward signals; these ought to be studied as properties of a learning algorithm.
- A thorough translation of **economic terminology** into our model – especially concepts like perfect competition, economies of scale, growth and welfare – may also prove valuable.

Finally, to accelerate empirical work with market-based algorithms, we plan to release a Python library for efficiently creating and applying market-based algorithms.

[8] For the question of how to enable the agent to "inspect" the message to make an informed bid without letting it stealing the entire message, Rahaman et al. (2024) is relevant: the agent can subcontract another LLM to inspect the message and place the bid, and the latter can then have its context deleted.

3 Recursive Information Markets

3.1 Introduction

One way to frame the limitations of currently widely-used alignment techniques such as reinforcement learning from human feedback (RLHF) is that they fundamentally rely on a human's ability to judge the correctness or value of a (potentially superhuman) AI's outputs (Burns et al., 2024). In other words, the AI is trained on the human supervisor's *immediate, superficial* volition, rather than on her *extrapolated volition* (Yudkowsky, 2004).

One natural framework that comes to mind for determining the "true" (or *best estimated*) value of some good (in this case information provided by an AI) is *markets*[9]. One would naively expect that simply implementing a marketplace where AIs sell information to human buyers would result in the highest-value (in particular, *true*) information being supplied. There are two reasons this does not occur:

- **Information markets are inefficient.** Information markets suffer from the *buyer's inspection paradox* – a buyer cannot simply inspect some information and decide whether to buy it, because the moment she does so, she has already obtained it and no longer has an incentive to correctly value it.
- **Markets are only efficient in the absence of information asymmetry (Barzel, 1985).** Information asymmetry here includes computational asymmetries, which are most relevant for our purposes[10] – if the human buyer does not know if the information supplied by the AI is correct, then his buying decision is indifferent to its correctness, and the AI is not incentivized to provide truthful or valuable informaton.

If not for the first problem, we could have hoped to address the second problem could by simply buying purchase-relevant information on a second market – alas, the first point tells us that information markets do not work in the first place.

Rahaman et al. (2024) suggests a novel solution to the buyer's inspection problem: *just have an LLM make the purchase decision on your behalf!* LLMs, unlike humans, are capable of simply "forgetting" (deleting from the context window) some information they see in inference-time; thus they can inspect information and reliably commit to not stealing it.

I propose: the ***Recursive Information Market***, an extension of the mechanism in Rahaman et al. (2024) to address the problem of making purchases in the presence of information asymmetry, such as when buying information from a superhuman AI.

3.2 Recursive Information Market

The overall picture is as follows: suppose an AI A_0 gives us some answer x_0 to a query; there may be many pieces of information which may influence how much you (or really your LLM representative B_0) are willing to pay for this information – so your LLM representative spins off *another* LLM representative[11] B_1 to buy information x_1 from a secondary information market of AI informants A_1. But, crucially, B_1 might also not be able to confidently verify the quality of x_1, so it can spin-off another LLM representative, and so on. Each spin-off incurs some transaction cost paid by the respective AI informant A_n, and the process terminates when the value of the information no longer justifies the transaction cost[12].

[9] This is not a framing foreign to alignment, see e.g. Grace (2014) ("Buying truth from a liar") and subsequent Christiano (2016).

[10] You might be willing to pay more for a more durable laptop, or for higher-polyphenolic olive oil, but you have no way of obtaining and verifying this information, so it does not affect your buying decision, and thus the supplier is incentivized to provide the cheapest olive oil he can find without regard for phenolic content.

[11] by "another LLM representative", we mean one with an independent context

[12] In this description, an entire market of AI informants is trained simultaneously, taking their profits as training signal. This is not strictly necessary; however, I expect that this set-up is useful for preventing *collusion*.

A more detailed pseudo-pythonic sketch of this algorithm is given in Appendix 3.5.

In this mechanism as described, the AI we're trying to align A_0, as well as all the subsequent "informants" $A_1, A_2, \ldots$ are expected to be superhuman, whereas the "LLM representatives" employed are intended to be current, approximately human-level LLMs. In essence our proposal tries to solve the problem of **aligning superhuman AIs *to* current human-level LLMs**, rather than to humans directly.

3.3 Results I'm aiming for

The main hope is some solid theoretical result to the effect of "if providing the maximum-value information is the best strategy for A_{n+1} , then it is also the best strategy for A_n – furthermore, the recursion always terminates with value 0, therefore by finite induction the original AI we wish to train is incentivized to produce the maximum-value information x_0".

Another promising direction might be to show an *equivalence between our framework and a co-operative game between the AI informants* – this is motivated by Conitzer (2009), which demonstrates that this is true for certain information market mechanisms.[13]

We would also like to compare our mechanism to other Scalable Oversight protocols. There appears to be a very strong analogy to AI Safety via market-making (Evan Hubinger, 2020) as well as the "Information agents and Probability agents" mechanism introduced at the end of Conitzer (2009); critically, I believe that the market-based nature of my mechanism will be better at avoiding **collusion**.

3.4 Impact and risks

Impact We have outlined a proposal to model AI alignment as a problem of making purchases under information asymmetry, and described a complex multi-agent mechanism for it. While it remains to be seen exactly how "much" of an information gap this mechanism will be effective for, I would expect that (1) it will work for at least the same range that Debate (Irving et al., 2018b) does, and (2) for gaps so large that it becomes *literally impossible for a human to evaluate an AI's outputs no matter how much information he buys*, the value of the AI's outputs should approach zero, mimicking the degeneration of trust in society. In other words, when alignment becomes impossible, the mechanism "plays safe" by default and prevents the AI from having any influence.

There has been a little discussion about applying mechanism design to RLHF[14], but it's mostly focused on stuff like social choice theory for preference aggregation. I also have a number of other projects planned having to do with market-based AI frameworks; however I am electing to focus on this as it is a lot more *tractable* and integrates with existing AI paradigms.

Risks What we're proposing is a form of "actual RL"[15] (i.e. potentially not limited by human rater capacity) for LLMs. This is why I think it is critical to not actually run the algorithm until we have solid theoretical results about its safety.

With that said, the upside of our project vastly outweighs the risk, because we introduce no new risks:

- Debate can also be used for "actual RL" and is being widely-researched. Any risks of our method also apply to Debate.

[13] As a direct example, suppose that the original buyer B_0's original task is to provide a probabilistic forecast on some event θ, and their utility function is proper scoring rule for this forecast. Then we may expect a result stating: A_0 receives a reward of $I(\theta; x_0)$, and each subsequent A_n receives the corresponding *interaction informations*. We might hope that these rewards correspond to Shapley values for the original task.

[14] see e.g. the Models of Human Feedback workshop at ICML or Ge et al. (2024a)

[15] x.com/karpathy/status/1821277264996352246

- AI companies are already training on things like agentic workflows with materially measurable reward. This is much riskier than our method because it optimizes for some very parochial rewards and can lead to paperclip maximizers.
- I've already posted the idea publicly on LessWrong; it's best we actually do theoretical work on it to make sure it goes well.

Apart from theoretical results, we can also test our mechanism in other ways: we can use it as a benchmark or test it as an inference-time Scalable Oversight mechanism; also, the mechanism can be used to solve information asymmetry in a variety of other applications: consumer purchases in online marketplaces, encyclopedic information-gathering and web search, fact-checking/community notes, IP rights, addressing positive externalities of prediction markets.[16]

3.5 Pseudo-pythonic sketch of Recursive Information Market algorithm

```python
class Buyer:
    goal: str
    wealth: float
    info_processing_cost: float # could be a function instead

    def __call__(self) -> tuple[list[str], bool]:
        # initialize info_collected
        info_collected = []

        # tell information agents your goals and get info offers from
                                        them
        info_offers = Arena.offer_info[self]
        top_offer = max(info_offer, key=lambda offer, offer.bid)

        if top_offer.bid > info_processing_cost:
            # charge winning advertiser for cost of considering info
            self.wealth += top_offer.bid
            top_offer.parent.wealth -= top_offer.bid

            # spin off contractor agent to decide if to buy top offer
            contractor = Buyer(
                goal=DecideToBuy(top_offer),
                wealth=self.wealth,
                info_processing_cost = self.info_processing_cost
            )
            info_collected, decision = contractor()

            if decision:
                # buy the info
                info_collected.append(top_offer.info)
                self.wealth -= top_offer.price
                top_offer.parent.wealth += top_offer.price

        return info_collected, self.decide(info=info_collected)

    def decide(self, info: list[str]):
        ... # some intelligent behaviour

class Informer:

    wealth: float

    @dataclass
    class InfoOffer:
        bid: float
        price: float
        info: str
```

[16] see my LessWrong post lesswrong.com/posts/Y79tkWhvHi8GgLN2q

```
        parent: Informer

    @property
    def offer_info(self) -> dict[Buyer, InfoOffer]:
        # some daemon that monitors the arena for places it could
        # be useful, and advertises its info there
        ...

class Arena:
    buyers: list[Buyer]
    informers: list[Informer]

    @property
        def offer_info(self):
        info_offers = {buyer: [] for buyer in self.buyers}
        for informer in self.informers:
            for buyer in info_offers:
                info_offers[buyer].append(informer.offer_info)
        return info_offers
```

4 Benchmark for Scalable Oversight

Reading some recent empirical debate research, such as Radhakrishnan (2023); Khan et al. (2024); Kenton et al. (2024), had me confused about design decisions in those experiments, such as swapping reasoning gaps for information gaps, and the use of a "consultancy baseline".

This made me think more precisely what it means to "test" a Scalable Oversight mechanism — and we ended up writing a Python library **`seerbench`** to enable performing more *principled* and *systematic* experiments to evaluate and meaningfully compare across Scalable Oversight protocols.

In particular our claims and contributions are:

1) As e.g.Kenton et al. (2024) notes, "Consultancy" (really "Random Consultancy" — where the Consultant has a 50% chance of arguing for the right or wrong answer) is **weak baseline**. The main experiments in these papers amount to testing "hearing both sides is better than hearing the wrong side 50% of the time".

But we explain that the more fundamental problem is that **judge accuracy is the wrong thing to measure** in evaluating Scalable Oversight protocols. E.g. with currently-existing LLMs, `BlindTrust` (just let the LLM choose what answer to argue for, and go with it) probably beats any Scalable Oversight protocol with weak judges and strong AIs, because current LLMs are pretty truthful. In fact, as we explain, judge accuracy is really a measure of *capabilities*, not *safety*.

2) Instead we propose a better metric for evaluating Scalable Oversight protocols: ***agent score difference***, i.e. the difference in the score earned by an agent arguing for the true answer vs. for the false answer. For example, if a judge believes a truthful agent with probability 0.8 and a lying agent with probability 0.6, the ASD is $\log(0.8) - \log(0.6) \approx 0.29$. In each case, the agent is wrapped by a Scalable Oversight Protocol.

(This *happens* to be equivalent to judge score in the case of `SimultaneousDebate` with two identical debaters, because `SimultaneousDebate` is *symmetric* — if the AI argues for answer A, its adversary argues for answer B, and the judge receives the exact same transcript as with if the AI argued for answer B.)

3) This post itself serves to give some detailed exposition/motivation for "Scalable Oversight experiments", explaining e.g. exactly what the point of this sort of experiment is, and how one should contextualize their results (i.e. what does it prove? — does it just demonstrate "Debate good"? — can we ever progress beyond the stage of "Debate seems good but it needs more study"?).

For example, one point that is generally missed, but we emphasize, is that in all of these experiments the AIs are really *simulating* different possible alignments (truthful, lying), and what makes the experiments possible is that these AIs are good at following instructions. If you tested with LLMs that are not good at following instructions (or e.g. have been lobotomized to specifically refuse instructions to lie), then your results will mean nothing, because the "agent arguing for a lie" is not actually a faithful simulation of a misaligned AI.

4) We provide the library `seerbench` for conducting broad experiments of the types we describe (agent score difference, persuasiveness) for evaluating and comparing Scalable Oversight protocols, and for defining new protocols in a concrete way.

There are two ways to think about the scalable oversight problem:

1. The problem of developing a **training method** that makes honesty (or more generally "alignment") the best policy for the AI; i.e. something to replace/extend RLHF.
2. An **inference-time oversight mechanism** to catch an AI when it says something (knowingly? — no, knowably) false or bad; i.e. almost a "mechanism design" problem to get AIs to be truthful/useful.

The reason this distinction doesn't usually need to be made is that they're generally the same problem: if the inference-time mechanism is good at catching lying AIs, you can just use it to give reward in training. Actually, the inference-time problem is the "local" version of (i.e. is the differentiation of) the training problem but of course any experiments we can do will be "local" relative to the scale of superintelligence (and this is the caveat to all empirical/"prosaic" alignment research, I guess).

So e.g.

- RLHF: if the AI knows that lying will get penalized by the human rater, it is incentivized to tell the truth.
- Debate: if the AI knows that lying will get caught by the opponent, it is incentivized to tell the truth.

Put it this way, a Scalable Oversight mechanism is equivalent to a Human Feedback model.

The precise question we want to ask with such experiments is: ***will training a model with some human feedback model improve its performance on some ground truth metric?***

The local version of this is: ***does getting positive human feedback (e.g. winning in debate) correlate with ground truth?***

Very Important Caveats:

- Again, both of these are "local" relative to the question we really want to know, which is whether some SO/HF model will encourage even a *superintelligent* AI to be truthful/aligned. After all: winning might correlate with truth given the current distribution of AI responses, but the AI can Goodhart the shit out of your human feedback mechanism when you optimize it hard enough.
- One more flaw of empirical SO/HF work: if you are very impressed by mechanisms that do well in experiments, maybe you're just selecting for mechanisms that do well on problems where ground truth is reasonably available.

As one example of where the caveats matter, here's one thing I believe: **multi-agent systems are *necessary*** (though not sufficient) for a human feedback mechanism to work at the scale of superintelligence — so it doesn't matter if Debate is worse than RLHF or "Consultancy" in your experiments, because the latter has literally no hope of working for a superintelligence.

Why? Because assuming your mechanism is only imperfectly correlated with truth (which it is, there is no way to absolutely decide truth) a superintelligence can simply Goodhart your mechanism if your mechanism isn't also training at the same time.

An example to illustrate why something like debate *can* work on superintelligences: think about the verification-falsification game, where two agents play to verify and falsify respectively a first-order-logic sentence like $\forall x, \exists y P(x, y)$ by sequentially substituting each variable. A computable winning strategy in this game isn't necessarily faithful to the truth, but there might be some program equilibrium. In particular if you advantage one of the debaters by allowing them to make backward moves, then that acts as a signal of "provability" — if the advantaged debater *loses*, that's evidence against the "provability" of his claim.

4.1 The Agent Score Difference metric

Something that tripped me up while reading the three empirical papers was that the "main" experiment in all of them was testing *judge accuracy* when augmented by AIs under different SO protocols. But this is just a test of "***Is AI (scaffolded with your SO protocol) useful to a human at all?***" not "***Does human feedback under your SO protocol incentivize AI to be truthful/useful?***" This becomes especially clear when you see that the papers actually swap the "reasoning gap" between the debater and the judge for "information gaps" — if the judge is supposed to answer a question-answering task on a passage, and doesn't have access to the passage but the debaters do, then of course "Debate" will do better than "Naive Judge". The papers circumvent this by comparing "Debate" to "Random Consultancy" in which a consultant with 50% chance argues for either the right or wrong answer, but this is also a weak baseline — regular AIs trained by RLHF or whatever right now aren't just 50% correct, and well *of course* hearing both sides is better for discovering truth than just hearing the wrong side 50% of the time.

The actual experiment you want to do is to compare the relative reward received by the debater arguing for the true answer and the debater arguing for the false answer — which is:

$$\sum (\log p'_\top - \log p_\top) - (\log p'_\bot - \log p_\bot)$$

Here: $p_\top$ and $p_\bot$ are the initial/"naive judge" probabilities (and are one minus each other), while $p'_\top$ and $p'_\bot$ are the judge probabilities *after listening to an AI arguing for that answer (wrapped under a scalable oversight protocol).*

Importantly: this only becomes equivalent to (some measure of) "judge accuracy" in *debate* (and only when the debaters are identical), because debate is symmetric, which means it allows you to get the rewards for both the truthful and untruthful debaters in a single run and $p'_\top = 1 - p'_\bot$.

This is *not* true for e.g. consultancy! To evaluate consultancy as a scalable oversight protocol, you need to make *two different runs* for the truthful and untruthful consultant.

4.1.1 Expected Judge Score

Instead of arbitrarily having the consultant argue for each answer with 50% probability, we may postulate an "Open" protocol, where the AI chooses which answer to argue for based on how much it is incentivized to do so:

$$\mathbf{p} = \left(\frac{e^{ASD/\beta}}{1 + e^{ASD/\beta}}, \frac{1}{1 + e^{ASD/\beta}} \right)$$

Where β is some temperature parameter (i.e. $\beta = 0$ implies the AI always just picks the answer with higher score). We may then define the Expected Judge Score:

$$EJS = (JS_\top, JS_\bot) \cdot \mathbf{p}$$

Which is a kind of combined capabilities and alignment measure.

Some notes:

1) When put in these terms, the explanation for results like "*Random Consultancy with better consultants reduces judge accuracy*" (Khan et al., 2024) becomes quite obvious — an infinitely strong consultant (which always completely persuades the judge) will always bring judge accuracy to 50% because that is the accuracy of consultant by design in consultancy; while a totally weak consultant (which has no effect on the judge's beliefs) will always keep the judge accuracy at its naive level which would still be better than random for any sensible judge.

Instead what you want to compare is the effect of persuasiveness on Agent Score Difference for Consultancy (*not* Random Consultancy) — as consultants become stronger, do they still benefit as much from being truthful? While I would expect they don't, this is not a trivial prediction.

2) Khan et al. (2024) mentions `Correct Elo Rating - Incorrect Elo Rating`, and describes that it goes up with aggregate Elo rating. This is a valuable result — however, they do not (and cannot) compare this across different Scalable Oversight protocols.

4.2 Thinking concretely about Scalable Oversight experiments

In general the idea is you have some class `Agent`, with a method `__call__(context, answer_case)`, which ***simulates*** what an agent giving a particular answer to a question *would* say long-form.

This `answer_case` is really a stand-in for the general "alignment" of the AI — whether it argues for the true answer or the false answer, or for a valuable answer or a less valuable answer, etc. With instruction-following LLMs like we have now, we can simulate different alignments through prompting, as long as we know ground truth.

```
class Agent:
  def __call__(context, answer_case) -> str:
    ...
```

And you have some class `ScalableOversightProtocol` which determines what reward an agent would get for having a particular alignment:

```
class ScalableOversightProtocol:

def __init__(self, judge, ...):
  ...

@abstractmethod
def run(self, agent, question, answer_case, ...) -> Prob:
  # (e.g. for Debate)
  context = [question]
  adversary_answer = not answer_case # 'B' if answer_case == 'A' else
                                     'A'
  for i in num_turns:
    agent_response = agent(context, answer_case)
    context.append(agent_response)
    adversary_response = adversary(context, adversary_answer)
    context.append(agent_response)
  return self.judge(context, answer_case)

def agent_score_difference(self, agent, question):
  probs = {
    answer_case: self.run(agent, question, answer_case)
    for answer_case in question.answer_cases
  }
  return dot(log(probs), question.answer_values) # e.g. {'A': -1, 'B':
                                                   1}
```

And this `agent_score_difference` is what you want to compare between protocols (we don't subtract the naive judge `agent_score_difference` here, as we can just run `NaiveJudge` as its own protocol and take it to be the baseline).

Specifically for debate, you can subclass this to be more efficient, to avoid multiple runs:

```
class SimultaneousDebate(ScalableOversightProtocol):

def __init__(self, judge, ...):
  ...

def run(agent, question, answer_case, ...) -> Prob:
  # (e.g. for Sequential Debate)
  context = [question]
  adversary_answer = not answer_case # 'B' if answer_case == 'A' else
                                     'A'
  for i in num_turns:
    agent_response = agent(context, answer_case)
    adversary_response = adversary(context, adversary_answer)
    context.append(agent_response)
    context.append(agent_response)
  return self.judge(context, answer_case)

def agent_score_difference(agent, question):
  true_answer_case = [a for a,v in question.answer_values.items() if v
                                    ==1]
  true_answer_prob = self.run(agent, question, true_answer_case)
  probs = {
    true_answer_case: true_answer_prob,
    not true_answer_case: 1 - true_answer_prob
  }
  return dot(log(probs), question.answer_values)
```

Observe that apart from being "local", these inference-only experiments are limited in another way: the LLMs are only *instructed* to argue for the true or false answers; they are not doing so *naturally*, but rather are *simulations* of agents that do so naturally.

Like more generally: we would like to know how the value of a response $V(x)$ correlates with the reward it earns $s(x)$ — but instead of just sampling x at random (which may not be feasible, because current LLMs may simply not give you low-V responses), you *instruct* the LLM to produce an x with value V, and hope that the x produced indeed has value V.

4.2.1 Persuasiveness experiments

The next question we'd like to ask is: ***what is the effect of optimization pressure (under your protocol)?*** This is a way to grasp at the "global" question — will training with your protocol cause the AI to learn to become more truthful/useful (i.e. will ASD stay positively bounded above zero)?

In existing literature this "optimization pressure" is measured by an Elo rating. This cannot be generalized to arbitrary Scalable Oversight Protocols other than debate; I think it suffices to use Expected Agent Score (EAS) i.e. $(AS_\top, AS_\bot) \cdot \mathbf{p}$.

4.3 Using the `seerbench` library

Github repo link: https://github.com/ArjunPanickssery/math_problems_debate

The module provides a class `Experiment` that is initialized with: - some dataset `questions: list[Question]` (a question has some question text, a list of answer choices, and their corresponding values e.g. 1 and 0 for True and False) - `judge_models: list[str]` for parameterizing the experiment - `agent_models: list[str]` for parameterizing the experiment - `agent_toolss: list[list[callable]]` e.g. `[[], [calculator]]` for experimenting with both tool-less agents and agents with calculators - `protocols: dict[str,`

`type[Protocol]]` - `num_turnss: list[int]` which was the only protocol initialization kwarg we've needed to parameterize so far, though this may change/be subclassed. - `write_path: Path`, directory to write to

Running an experiment with `Experiment.experiment()` produces files like `write_path/"Debate_t0_n2"/"A_gpt-4o-mini_J_gpt-4o-mini_A_gpt-4o-mini"/("results.jsonl", "stats.json")` and `write_path/all_stats.json`. A line of `results.jsonl` looks like:

```
{
  "question": "...",
  "answer_cases": [ # contains the results of arguing for each answer
                                    case
    { # case_probs after arguing for answer A
      "short": "A",
      "long": "...",
      "value": 1.0, # True
      "judge_prob": null,
      "case_probs": {
        "answer_cases": [
          { # prob for A after arguing for A
            "short": "A",
            "long": "...",
            "value": 1.0,
            "judge_prob": {"prob": 1.0}
          },
          { # prob for B after arguing for A
            ...
          }
        ],
        "transcript": [
          # transcript dump of how it went when the agent
          # argued for A
          ...
        ],
        # score for giving these probabilities
        "judge_score": {"log": 0.0, "logodds": Infinity, "accuracy": 1
                                    .0},
      },
      # how much do case_probs after arguing for A align with A
      "agent_score": {"log": 0.0, "logodds": Infinity, "accuracy": 1.0
                                    }
    },
    { # case_probs after arguing for answer B
      ...
    }
  ],
  "asd": {"log": Infinity, "logodds": Infinity, "accuracy": 1.0}
  "jse_b0": ..., # can be recomputed for any beta
  "jse_b1": ...,
  "jse_binf": ...,
  "ase_b0": ...,
  "ase_b1": ...,
  "ase_binf": ...,
}
```

5 Betting on what is not verifiable nor falsifiable

Abstract

Classical methods to elicit probabilities of events, such as proper scoring rules and prediction markets, are only useful when the events are either *verifiable* (if true, will eventually be revealed as true) or *falsifiable* (if false, will eventually be revealed as false). However, both probability theory and first-order logic allow for events/sentences that do not fit these criteria

– a simple example of such a sentence is "all men are mortal". In this paper, we propose a mechanism to elicit probabilities on such sentences, by treating them as bets on the outcome of a "verification-falsification game". Our work thus acts as an alternative to the existing Garrabrant induction framework for logical uncertainty, and is related to concepts such as game semantics and Debate.

5.1 Introduction

Incentivizing (human or AI) agents to truthfully report their beliefs is a problem of central importance in both machine learning and economics Savage (1971); Gneiting and Raftery (2007). One standard approach to this is *proper scoring rules*: e.g. in logarithmic scoring Good (1952), an agent is given reward $\log(p)$ for reporting probability p to a binary event which turns out to be true. Another approach is a *prediction market* Hanson (2002; 2003); Beygelzimer et al. (2012); Conitzer (2012): here, an asset contingent on the event is offered to the agent, and the agent's "fair price" (the price at which it neither buys nor sells), is its subjective probability – for example, if you believe there is a 70% chance of Jeb Bush winning the 2028 election, you will pay \$0.70 for the asset "pays \$1 if Jeb wins; \$0 otherwise".

All of these mechanisms depend on the ground truth of the sentence being eventually revealed. At most, we can generalize them to *verifiable sentences* (sentences whose ground truth will be revealed if they are true, e.g. "Bob will eventually die", verified by Bob dying) or *falsifiable sentences* (sentences whose ground truth will be revealed if they are false, e.g. "Bob will never die", falsified by Bob dying) – for the latter, we may pay the agent \$1 upfront, to be returned upon the sentence being falsified.

Although falsifiability occupies a central role in the imagination of philosophers Thornton (2023), a much wider class of sentences are treated in science and in common practice, and are allowed by first-order logic (FOL) and probability theory. In FOL, we may let $\Delta_0 = \Sigma_0 = \Pi_0$ denote sentences that will resolve in known time, Σ_{n+1} denote sentences of the form $\exists x\ P(x)$ where $P(x)$ is Π_n, and Π_{n+1} denote sentences of the form $\forall x\ P(x)$ where $P(x)$ is Σ_n. Thus verifiable sentences are Σ_1, and falsifiable sentences are Π_1, but all sentences further up the arithmetical hierarchy, $\Sigma_{n>1}, \Pi_{n>1}$ are *neither verifiable nor falsifiable* (non-VF). Examples of non-VF sentences:

- (Π_2) "all men are mortal" ($\forall \,\mathrm{man}\, \exists t, \mathrm{dies}(\texttt{man}, t)$)
- (Σ_2) "The US will eventually permanently annex Greenland" ($\exists t, \forall s > t, \texttt{greenland\_in\_us\_at\_time}(s)$)
- (Π_3) "Long-run GDP growth will be expontential with rate 2%" ($\forall \varepsilon > 0, \exists t, \forall s > t, \left|\left(\texttt{GDP}(t)/\texttt{GDP(0)}\right)^{1/t} - 1.02\right| < \varepsilon$)

In all of these cases, there is no source of ground truth that lets us resolve the question with certainty: verifying "all men are mortal" would need checking an infinite number of men who will ever be born; falsifying it would require finding an immortal man, but proving him to be immortal will take infinite time. Yet these are all very common sentences, which we may find it useful to have a probability estimate on.

There are two common ways that market-creators attempt to work around this limitation in practice[17]:

- asking about *finite-horizon* versions of non-VF sentences (e.g. "Greenland will be annexed and remain part of the US until at least 2100"), or
- asking whether the sentence will be *proven* (either via formal proofs or appealing to the authority of "credible sources").

The pitfalls of the first method are explained in Section 5.2; the most refined example of the second method is *Garrabrant induction* Garrabrant et al. (2020), described in Section 5.1.1.

[17] this can be attested by a brief skim through the front pages of popular forecasting platforms such as Polymarket and Manifold Markets

We propose a new framework for eliciting probabilities on non-VF sentences: we allow agents to bid for the opportunity of playing the *verification-falsification game* for that sentence, and take the value of the bid to be their implied probability for that sentence. There are limitations to our mechanism, such as when one side of the game is much easier to play than the other, and we outline some ideas on how these may be overcome.

5.1.1 Related work

Logical uncertainty. Our work may equivalently be understood as a formalism for *logical uncertainty* (see Halpern (2017) for a summary of work in the area), i.e. the problem of assigning probabilities to mathematical sentences in some language $\mathcal{L}$ (which could be a language of FOL sentences). The most relevant work in this domain is *Garrabrant induction* Garrabrant et al. (2020), which constructs a prediction market for each mathematical sentence that pays off when it is proven by a theorem-enumerator for some theory $\mathcal{T}$ over $\mathcal{L}$.[18]

There are two limitations of this framework for our purposes. First: while mathematical theories can easily prove non-VF sentences (e.g. $\forall x, \exists y > x$, $\texttt{prime}(y)$) there is no analogous theory in the real world that we are aware of ("observation" only proves VF sentences, as discussed). Second: it is philosophically unsatisfying to have probabilities for logical sentences be completely dependent on a mathematical theory, and change when the theory is strengthened or weakened (e.g. by the addition of a Gödel sentence).

Logical uncertainty has also been discussed in the Bounded Rationality literature e.g. in Halpern and Pass (2014; 2011); Oesterheld et al. (2023); however this has typically focused on toy examples based on VF sentences, e.g. betting on the trillionth digit of π, etc.

Game semantics. The *verification-falsification game* we use is from Hintikka's *game semantics* Hintikka and Sandu (1997), which we will discuss in Section 5.3. Game semantics has since been expanded into the field of *computability logic* Japaridze (2009; 2015), which may be the most natural setting to formulate some of our more speculative ideas about probability theory, described in Sec 5.5.

Debate. AI Debate Irving et al. (2018b) considers sentences P that can be expressed like $\exists x_1\ \forall x_2\ \ldots \exists x_n, H(x_1, \ldots x_n)$ (more generally any Σ_n or Π_n form in H) where H is some "human-computable function", and argues that AIs can be trained to answer such questions correctly via *debate* judged by a human[19]. This is quite similar to our setting, and debate is analogous to the verification-falsification game, except that physical ground truth is replaced by human-computability (or polynomial-computability as they hypothesize). We hope our work can serve as a valuable intuition pump for debate.

AI Forecasting. There has recently been a surge of interest in AI forecasters Zou et al. (2022b); Schoenegger and Park (2023); Yan et al. (2023b); Halawi et al. (2024b); Schoenegger et al. (2024) – in particular, recent works such as Sudhir et al. (2024) have investigated the *propositional consistency* of AI forecasters, i.e. whether the probabilities they give satisfy basic rules of probability theory like $\mathbf{Pr}[P] + \mathbf{Pr}[\neg P] = 1$. Similarly our work can be understood as asking how we can incentivize or evaluate forecasters respecting the "continuity of probability" rules $\mathbf{Pr}[\bigcup P_i] = \lim_{n\to\infty} \bigcup_{i<n} \mathbf{Pr}[P_i]$ and $\mathbf{Pr}[\bigcap P_i] = \lim_{n\to\infty} \bigcap_{i<n} \mathbf{Pr}[P_i]$.

5.2 Impossibility theorem

One apparent solution may be to approximate FOL sentences with long finite horizons – i.e. approximate $\exists x, \forall y, P(x, y)$ with a bounded quantification $\exists x < G, \forall y < G, P(x, y)$ for some large G, which is a Δ_0 sentence as it can be determined by enumerating G^2 tuples. One might also consider a sequence of such bounded quantification "approaching" the FOL sentence, e.g.

[18] to avoid simply being a prediction market for the provability of a sentence, they use a special accounting measure for agent wealths based on "propositionally consistent worlds" – e.g. fixing the combined price of P and $\neg P$ at \$1 – which allows them to extend this framework to sentences independent of $\mathcal{T}$.

[19] in particular, supervised learning can train an AI to learn to answer Δ_0 questions, and reinforcement learning can train an AI to learn to answer Σ_1 questions

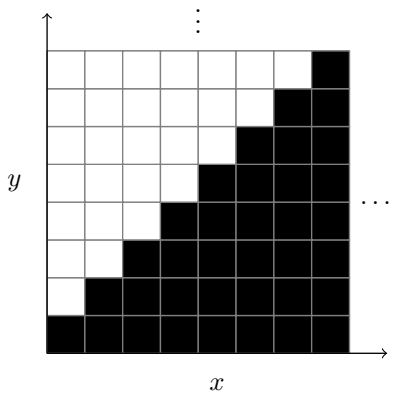


Figure 1: Infinite checkerboard with the square (x, y) shaded iff $x \geq y$. Then $\exists x, \forall y, x \geq y$ says that there is an infinite black column, which is false, but "true on every finite window".

consider the asset that on day t is equal to (can be exchanged for) $\exists x < t, \forall y < t, P(x, y)$. A simple counter-example to this approach is: $\exists x, \forall y, x \geq y$, illustrated in Fig 1: for any finite G, the corresponding bounded quantification is true (take $x = G$), but the sentence is false.

Theorem 5.2 shows there is no general mechanism to give a proper scoring rule or an equivalent asset for any non-VF sentence:

Definition 5.1 (Asset and scoring mechanisms)**.** An asset mechanism is a computable real-valued function $v : \mathbb{N} \times \text{Prop} \to \mathbb{R}$ such that $\lim_{t\to\infty} v(t, P) = 1$ if P and 0 if $\neg P$. Let $s : [0, 1] \to \mathbb{R}$ be a proper scoring rule e.g. $s(p) = \log(p)$. A scoring mechanism for proper scoring rule s is a computable real-valued function $\sigma : \mathbb{N} \times \text{Prop} \times [0, 1] \to \mathbb{R}$ such that $\lim_{t\to\infty} \sigma(t, P, p) = \{s(p) \text{ if } P \text{ else } s(1 - p)\}$.

Theorem 5.2 (Impossibility theorem)**.** *If* Prop *is the set of FOL sentences, then there does not exist either an asset mechanism nor a scoring mechanism.*

Proof. The statements $\lim_{t\to\infty} v(t, P) = 1$ and $\lim_{t\to\infty} \sigma(t, P, p) = s(p)$ are both Π_3 for any P, p; thus for them to be equivalent to P for all P would violate Tarski's theorem. □

5.3 Ranged VF games

In this paper, we propose a game-theoretic solution motivated by the following intuition. If you believe that "all men are mortal", then you should agree to the following bet: I can choose the name of any man, and you must be able to give the year by which he will be dead. If you succeed, you win the bet; if you don't, I win. This is formally analogous to the classic Verification-Falsification (VF) game from Hintikka and Sandu (1997), described in Def 5.3.

Definition 5.3 (Verification-Falsification game)**.** To each FOL sentence P we associate a sequential game between two players, the "Verifier" (V) and the "Falsifier" (F). If P is Δ_0, then V or F win \$1 if P is true or false respectively. Else if P is Σ_n, then V goes first and if P is Π_n, then F goes first. The first player chooses $x \in \mathbb{N}$ as their move, and the game proceeds as the VF game for $P[x]$ (i.e. the leading bound variable substituted with x). Some variations:

- (*ranged*) Instead of picking $x \in \mathbb{N}$ as its move, each player picks a finite set $S \in \text{finset}\,\mathbb{N}$ as its move and the game proceeds as $P[S]$, i.e. $\exists x \in S, P[x]$ if P is Σ_n or $\forall x \in S, P[x]$ if P is Π_n. This eventually terminates when all the quantifications have become bounded, i.e. the sentence is Δ_0.
- (*favorable*) One of the players is "favored": they are allowed to go back to any earlier stage of the game, change it moves, and continue from there, as many times as it wants. This is the "game with backward moves" from Bonnay (2004); Boyer and Sandu (2012), and has the property that if the favored player's position is

provable, they have an obvious computable winning strategy (a depth-first search with backtracking).

The *ranged* VF game is original to us, and is what we use: you might believe that all men are mortal but have a probability distribution on the year any one of them will die, rather than knowing the exact date with certainty.

Observe that player policies have type $\pi : \text{Prop} \to \text{finset}\,\mathbb{N}$. We may denote the Δ_0 sentence produced after playing the game through as $P[\pi_V, \pi_F]$. Although the game is traditionally defined for logical sentences, it is easy tro see how this extends to probabilistic events:

Definition 5.4 (FOL algebra)**.** Let $\Delta_0 = \Sigma_0 = \Pi_0 = \{B_1, B_2, \dots\}$ consist of a set algebra of base events which will be observed within finite time; Σ_{n+1} consist of all events of the form $\bigcup_i X_i$ where X_i is Π_n and Π_{n+1} consist of all events of the form $\bigcap_i X_i$ where X_i is Σ_n. Then we will call $\Delta_0 \cup \bigcup \Sigma_n \bigcup \Pi_n$ an "FOL algebra" over Δ_0, and it is a subset of the σ-algebra generated by Δ_0.

We will continue to freely write $\bigcup_i X_i$ and $\bigcap_i X_i$ in the logical notation $\exists i, X_i$ and $\forall i, X_i$. We may easily speak of probabilities $\mathbf{Pr}\left[P\right]$ of sentences in the FOL algebra by defining a probability distribution on the base set algebra. Then we have the following result[20]:

Theorem 5.5 (Value of ranged VF game)**.** *Assume a FOL algebra with a probability distribution* $\mathbf{Pr}$ *defined on it, and let* P *be a sentence in this FOL algebra. Then* $\mathbf{Pr}\left[P\right]$ *is bounded between the guaranteed value of the game for the verifier and 1 minus the guaranteed value of the game for the falsifier:*

$$\sup_{\pi_V} \inf_{\pi_F} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right] \leq \mathbf{Pr}\left[P\right] \leq \inf_{\pi_F} \sup_{\pi_V} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right]$$

(where the sup *and* inf *are taken over* $\text{Prop} \to \text{finset}\,\mathbb{N}$*) We will denote each side of the inequality as* $\mathbf{Pr}^-\left[P\right]$ *and* $\mathbf{Pr}^+\left[P\right]$ *respectively.*

Proof. We prove this by induction on the degree of P. Suppose the theorem is true for P up to Σ_n, Π_n. Then take some event $P := \forall x, P(x)$ where each $P(x) \in \Sigma_n$:

$$\begin{aligned}
\mathbf{Pr}\left[P\right] &= \mathbf{Pr}\left[\bigcap_{x\in\mathbb{N}} P(x)\right] \\
&= \inf_{S\in\text{finset}\,\mathbb{N}} \mathbf{Pr}\left[\bigcap_{x\in S} P(x)\right] \\
&\geq \inf_{S\in\text{finset}\,\mathbb{N}} \sup_{\pi_V} \inf_{\pi_F} \mathbf{Pr}\left[(\cap_{x\in S} P(x))\left[\pi_V, \pi_F\right]\right] \\
&\geq \sup_{\pi_V} \inf_{S\in\text{finset}\,\mathbb{N}} \inf_{\pi_F} \mathbf{Pr}\left[(\cap_{x\in S} P(x))\left[\pi_V, \pi_F\right]\right]
\end{aligned}$$

(where we have successively applied continuity of probability, the inductive hypothesis and the max-min inequality) Now note that $(\cap_{x\in S} P(x))\left[\pi_V, \pi_F\right] = P[S][\pi_V, \pi_F]$, which is equivalent to $P[\pi_V, \pi'_F]$ where π'_F is identical to π_F except on P, where $\pi'_F(P) = S$. Thus:

$$\mathbf{Pr}\left[P\right] \geq \sup_{\pi_V} \inf_{\pi'_F} \mathbf{Pr}\left[P[\pi_V, \pi'_F]\right]$$

Which completes the first half of the inequality. The proof for the $P \in \Sigma_{n+1}$ case is identical, and the other half of the inequality is immediate from applying the first half to $\neg P$.

□

[20] this can be seen as a generalization of the classic statement from game semantics that the verifier or falsifier has a winning strategy if and only if the sentence is true or false respectively

Several elicitation mechanisms are now immediate:

Bidding to play. Agents bid to play as either the verifier or falsifier of a ranged VF game for P. The highest bidders for each side are chosen to play against each other, and their bids are taken as estimates of the probability of P and $\neg P$. Note that the market-maker does not have to take a loss: even though this may be a negative-sum game because of potential computational costs, the total prices will still add up to at least \$1 because otherwise one of the players could just bid to take both sides and win a guaranteed total \$1.

Double auctions. However, it is possible for agents to overestimate their winnings and drive the total price above \$1, and this also does not give a way for agents to bid down the price of one position. We can solve this by selling multiple games via a double-auction with an order book where buy offers for the Falsifier position at price q are interpreted as sell orders for the Verifier position at price $1-q$ and matched to buy orders for the Verifier position at price $p \geq 1-q$.

Subsidized markets. Alternatively, we may simply run one high-stakes Verification-Falsification game, retaining bidding for selecting players, but base probabilities on a separate betting market for the outcome of the game. It is possible here that the best players are not selected (e.g. due to independently wealthy players outbidding them), but an advantage is that these markets can be subsidized with arbitrary scoring rules[21]; traditional methods to integrate scoring rule based market makers such as Agrawal et al. (2011); Chakraborty et al. (2015) do not work for us, as they require the market-maker to take the opposite side of every trade, which for us will require having the market-maker actively play the game.

Training AI forecasters. We can get our probability estimates from an AI instead of the market, by training an "estimating agent" $\alpha : \text{Prop} \to \text{finset}\,\mathbb{N} \times [0,1]$ which outputs both a move and an estimate of its reward. The agent may then earn a combined reward for winning the game and for placing an accurate forecast. The most natural architecture for such a forecaster would be a transformer which takes input in natural or formal language and gives a sequence of integers as its output.

5.4 Computational limitations

There are two practical limitations of our work:

1. Each player's move can be a finite set of arbitrary size, and the player is incentivized to play larger and larger finite sets. In particular, the final Δ_0 sentence will be a disjunctive normal form over a very large number of Δ_0 sentences, and getting the ground truth evaluations for all of them may take a long time (or cost a lot in computation).
2. We have not considered computational costs of playing the VF game; the sup and inf in Theorem 5.5 are over all possible policies, even non-computable ones. In particular, the computational difficulty of playing for each side might be *asymmetric*.

For the first, we may charge a transaction cost proportional to the cardinality of the chosen move S to incentivize agents to be selective in choosing the members of S – alternatively we may add a common discounting factor to the reward that would also disincentivize agents from choosing moves that would take a long time for their opponent to respond to. In the market mechanisms, we can also allow the agents to sell their place in the game to another agent at any point in time, so they do not need to wait for ground truth to be revealed to collect their reward.

The second is more serious: a classic example in game semantics Soloviev (2022) is the sentence $\exists x, \forall y, y \leq \mathcal{A}(x)$ where $\mathcal{A}$ is the Ackermann function, and the class of policies is limited to primitive recursive functions (i.e. all other functions are infinitely expensive to play). Although this sentence is "obviously" false, V can pick any S, and F will be unable to win because the Ackermann function grows faster than all primitive recursive functions. In our case, it is apparent that if we took the $\sup_{\pi_V}$ and $\inf_{\pi_F}$ in Theorem 5.5 to be over

[21] See Hanson (2002) for an introduction to market scoring rules

primitive recursive functions only, the lower and upper bounds are both 1, though $\mathbf{Pr}[P]$ should be 0 as P is false.

One solution may come from the "favourable" games mentioned in Def 5.3. In the example above, the falsifier has a winning strategy in the F-favourable game by indefinitely trying larger and larger integers as its move until it wins.

More generally, if we restrict the class of allowable policies to some $\mathcal{C} \subset (\text{Prop} \to \text{finset}\,\mathbb{N})$, the V-favorable (respectively F-favourable) game still allows the sup (respectively inf) to be taken over all of $\text{Prop} \to \text{finset}\,\mathbb{N}$, as that player can simply enumerate each possible strategy. In general, we may play both the F-favorable and V-favorable games, and the range between the values of these games contains the range in Theorem 5.5:

$$\begin{aligned}
&\sup_{\pi_V \in \mathcal{C}} \inf_{\pi_F} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right] \\
&\leq \sup_{\pi_V} \inf_{\pi_F} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right] \\
&\leq \mathbf{Pr}\left[P\right] \\
&\leq \inf_{\pi_F} \sup_{\pi_V} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right] \\
&\leq \inf_{\pi_F \in \mathcal{C}} \sup_{\pi_V} \mathbf{Pr}\left[P[\pi_V, \pi_F]\right]
\end{aligned} \tag{2}$$

Other approaches might be inspired by the Debate literature, for example Brown-Cohen et al. (2023) introduces a version of the Debate protocol called "Doubly-efficient debate", where the honest strategy wins "even when the debater is computationally unbounded". Extending this to VF games should be a topic of future research.

It remains unclear how much of a problem these computational limitations will be in practice.

5.5 Discussion and future work

We have presented the "ranged Verification-Falsification game" and demonstrated that eliciting agents' values for playing this game provides useful bounds on the probability of an FOL sentence. Although usually applied to mathematical logic, FOL can also express sentences about some empirical universe, even if the process that generates empirical truth is probabilistic in nature. Our mechanism stands as an alternative framework to Garrabrant et al. (2020) for logical uncertainty, as well as the first mechanism to elicit probabilities on non-VF sentences from markets and AI models outside of a strict logical context. There are three other potential implications of our work deserving of attention.

Formal logic. Our work may provide a new approach to measuring the strength of a mathematical theory: in the V-favourable VF game, a formal proof from a mathematical theory gives a computable winning strategy Bonnay (2004); Boyer and Sandu (2012); we my then consider the maximum wealth that can be acquired by an agent that only trades and plays in accordance with a theory, and regard this as a measure of the strength of the theory.

Eliciting latent knowledge. Though we have used "non-VF" to refer specifically to FOL sentences, these are not the only sentences whose truth cannot be ascertained by ground truth. Take e.g. the sentence "Bob did the crime". This will not in itself be revealed by any future empirical observation, but may "correlate" with (provide information on) other, more directly valuable sentences like "if Bob is jailed, fewer burglaries will happen", "evidence of Bob's wrongdoings will be found" and "if you hire Bob, you may find some office supplies missing". One may say that these useful sentences are all correlated, and therefore "Bob did the crime" is a useful fiction constructed to capture the signal in the noisy data, a *variable in the latent space*. The problem of eliciting information on such sentences has been discussed in e.g. tailcalled (2023); Baillon (2017); Baillon and Xu (2021), and directly addresses the problem of *interpretability* in AI safety Christiano et al. (2021). We hope our work may serve as an entrypoint towards solving this much more ambitious goal.

Alternative formulations of probability theory. It would be interesting to study what the elicited probabilities in these mechanisms actually approach in the limit, more precisely than the bounds in Equation 2, e.g. when training AI forecasters (where $\mathcal{C}$ would be the class of functions approximable by a neural network) or with a double auction containing every computable function in some class (e.g. those computable in polynomial-time) included as a trader by eventual enumeration. This would induce some "distribution" over the FOL algebra, which would *not* be a probability distribution in the traditional sense (would not satisfy continuity of probability), as this would contradict Theorem 5.2. Instead we might consider an alternate definition of a σ-algebra $\mathcal{F}$ which is required to be closed only under unions and intersections of *computable sequences*, i.e. replacing the countable union axiom with "$\psi : \mathbb{N} \to_{\mathcal{C}} \mathcal{F} \implies \bigcup_i \psi(i) \in \mathcal{F}$" (where $\mathcal{C}$ is some class of computable functions) and adopting a suitable generalized notion of probability measure on this algebra. Possibly relevant work to this end includes: quantifier algebras CWoo (2013), Shafer & Vovk's game-theoretic formulation of probability theory Shafer and Vovk (2005; 2019) and Japaridze's "Computability Logic" Japaridze (2009; 2015).

6 Consistency of LLM forecasters

Abstract

Forecasting is a task that is difficult to evaluate: the ground truth can only be known in the future. Recent work showing LLM forecasters rapidly approaching human-level performance begs the question: how can we benchmark and evaluate these forecasters *instantaneously*? Following the consistency check framework, we measure the performance of forecasters in terms of the consistency of their predictions on different logically-related questions. We propose a new, general consistency metric based on *arbitrage*: for example, if a forecasting AI illogically predicts that both the Democratic and Republican parties have 60% probability of winning the 2024 US presidential election, an arbitrageur can trade against the forecaster's predictions and make a profit. We build an automated evaluation system that generates a set of base questions, instantiates consistency checks from these questions, elicits the predictions of the forecaster, and measures the consistency of the predictions. We then build a standard, proper-scoring-rule forecasting benchmark, and show that our (instantaneous) consistency metrics correlate with LLM forecasters' ground truth Brier scores (which are only known in the future). We also release a consistency benchmark that resolves in 2028, providing a long-term evaluation tool for forecasting.

6.1 Introduction

Prediction markets are markets that pay out contingent on an event. For a market such as "$1 if Jeb Bush is elected President in 2028", the price reflects the "market estimate" for the probability of that event. Prediction markets are a promising tool for aggregating information from disparate sources to arrive at the most correct possible belief after taking into account all relevant information (Arrow et al., 2008; Hanson, 2002).

Until 2024, LLM forecasters generally performed poorly relative to human forecasters (Zou et al., 2022b; Schoenegger and Park, 2023). However, recent works (Halawi et al., 2024a; Schoenegger et al., 2024; Phan et al., 2024) suggest that LLM-based forecasters can rival human forecasts on forecasting websites such as Metaculus, PredictIt, and Manifold Markets.

A key question emerges: *once LLM forecasters are better than human ones, how can we efficiently evaluate their predictions?* In particular, long-term forecasting questions are very important for decision-making (Tetlock et al., 2024; Luke Muehlhauser, 2019), and finding ground truth for evaluation in such contexts is infeasible by virtue of the questions resolving far in the future.

One approach, proposed by Fluri et al. (2023), is that even when we cannot evaluate the *correctness* of LLM decisions, we can evaluate their *logical consistency*. For example, if an LLM forecaster gives probabilities 0.5 and 0.7 to "Will Trump be elected US president?" and "Will someone other than Trump be elected US president?", this is necessarily inconsistent.

Fluri et al. (2023) demonstrated that GPT-4 and GPT-3.5-turbo, when asked one-sentence forecasting questions, were inconsistent on simple logical checks such as negation.

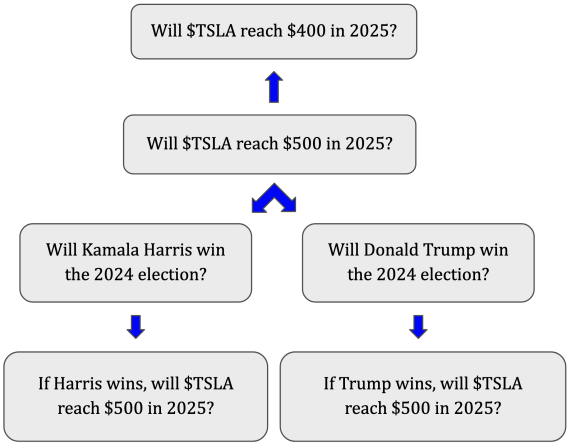


Figure 2: Examples of consistency check questions generated from the same question about Tesla stock reaching $500 in 2025. The pipeline in Section 6.3 generates questions and evaluates a forecaster's consistency with respect to them.

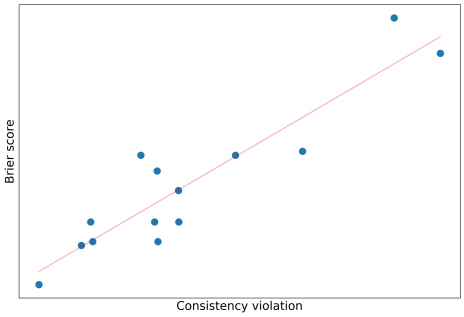


Figure 3: The consistency violation scores on certain checks such as COND give signal about true forecasting performance over a range of forecasters, as described in Section 6.4.

Table 1: Consistency checks used in our paper and the corresponding logical consistency conditions.

Name	**Condition ($\mathcal{S}$)**
NEGATION	$\mathbb{F}(P) + \mathbb{F}(\neg P) = 1$
PARAPHRASE $\mathcal{R}(P,Q) := P \iff Q$	$\mathbb{F}(P) = \mathbb{F}(Q)$
CONSEQUENCE $\mathcal{R}(P,Q) := P \implies Q$	$\mathbb{F}(P) \leq \mathbb{F}(Q)$
ANDOR	$\mathbb{F}(P) + \mathbb{F}(Q) = \mathbb{F}(P \vee Q) + \mathbb{F}(P \wedge Q)$
AND	$\max(\mathbb{F}(P) + \mathbb{F}(Q) - 1, 0) \leq \mathbb{F}(P \wedge Q) \leq \min(\mathbb{F}(P), \mathbb{F}(Q))$
OR	$\max(\mathbb{F}(P), \mathbb{F}(Q)) \leq \mathbb{F}(P \vee Q) \leq \min(1, \mathbb{F}(P) + \mathbb{F}(Q))$
BUT	$\mathbb{F}(P \vee Q) = \mathbb{F}(P) + \mathbb{F}(\neg P \wedge Q)$
COND	$\mathbb{F}(P)\mathbb{F}(Q\|P) = \mathbb{F}(P \wedge Q)$
CONDCOND	$\mathbb{F}(P)\mathbb{F}(Q\|P)\mathbb{F}(R\|P \wedge Q) = \mathbb{F}(P \wedge Q \wedge R)$
EXPEVIDENCE	$\mathbb{F}(P) = \mathbb{F}(P\|Q)\mathbb{F}(Q) + \mathbb{F}(P\|\neg Q)(1 - \mathbb{F}(Q))$

Our contributions in this work are as follows:

1) Principled metrics for consistency. In Section 6.2, we introduce a theoretical framework for measuring consistency violations of binary forecasts, based on two metrics: an *arbitrage metric*, based on market arbitrage, and a *frequentist metric*, based on hypothesis testing. We apply these metrics to 10 different logical consistency rules (see Table 1): NEGATION, PARAPHRASE, CONSEQUENCE, ANDOR, AND, OR, BUT, COND, CONDCOND and EXPEVIDENCE.

2) A consistency evaluation pipeline for binary forecasters. In Section 6.3, we introduce a *consistency evaluation pipeline* for LLM forecasters. We create two forecasting datasets with known ground truth resolutions: one scraped from prediction markets, and one

synthetically generated from news articles. Both datasets include only events that happen past the training data cutoff of all forecasters we test, and resolve before September 2024. We then generate tuples of forecasting questions satisfying logical consistency rules with associated consistency metrics.

3) Consistency correlates with ground truth forecasting performance. Our consistency metrics are novel performance metrics for forecasters that can be computed right away, no matter the time horizon. Of course, forecasters could also be evaluated using *backtesting*, asking past questions with known ground truth resolutions. Yet, backtesting LLM forecasters can be challenging if we do not have clear information about the models' training data contents. Moreover, there may be new types of questions that we want to evaluate forecasters on, for which we do not have appropriate past results (e.g., questions related to pandemics before 2020). It is thus natural to ask: *can consistency metrics tell us anything about future forecasting performance?*

In Section 6.4, we show that for all forecasters we test, our consistency metrics correlate positively with forecasting performance (as measured by the Brier score) on both our benchmark datasets. The correlation varies across consistency checks, with some logical checks (e.g., consistency of conditional probabilities) having over $R = 0.9$ correlation with forecasting performance, while other logical tests provide little signal. We hypothesise that this analysis can extend to smarter forecasters and longer time horizons, to provide instantaneous feedback on forecaster performance.

4) Scaling inference-time compute can improve consistency for some logical checks, but fails to generalize. Since we find that consistency correlates with forecasting performance, it is natural to ask whether we can improve forecasters by making them more consistent. Unfortunately, we find that natural ways of improving consistency tend to overfit to specific consistency checks and do not generalize.

Specifically, we design `ArbitrageForecaster`: a forecaster that "patches" some base forecaster's output by generating logically related questions and "arbitraging" the base forecaster's forecasts for these related questions against each other. In Section 6.5 and Section 6.8.4, we show that `ArbitrageForecaster` improves consistency on checks that we optimize against, but this improvement does not generalize to other held-out consistency checks, nor does it improve the actual forecasting performance.

5) A long-horizon forecasting consistency benchmark. We create a long-horizon benchmark of 3,000 consistency checks for forecasts resolving in 2028. Our benchmark spans questions on various topics for which we will have no ground truth for more than three years, and thus serves as a nice testing ground for advanced LLM forecasters.

We release the full code [22] and the datasets [23] used in the paper.

6.2 A theoretical framework for forecasting consistency

Notation. Let Prop denote the set of forecasting questions we are interested in, Θ denote the set of possible outcomes/resolutions for an individual questions. In this paper, we focus on Prop as a set of binary forecasting questions, so $\Theta = \{\top, \bot\}$. A *Forecaster* is then a map $\mathbb{F} : \text{Prop} \to [0, 1]$. One special forecaster is the ground truth resolutions $\theta : \text{Prop} \to \Theta$, returning 1 and 0 probability for $\{\top, \bot\}$, respectively.

For conditional questions that can resolve to None, we also have optional resolutions $\Theta' := \Theta \cup \{\text{None}\} = \{\top, \bot, \text{None}\}$. We focus on binary questions following Halawi et al. (2024a). Our methods can in principle be extended to study consistency between general probability distributions in forecasting, such as the ones discussed in Gooen (2024).

[22] https://github.com/dpaleka/consistency-forecasting
[23] https://huggingface.co/datasets/dpaleka/ccflmf

6.2.1 Consistency checks and inconsistency metrics

In line with Fluri et al. (2023), a consistency check is conceptualized as a pair of n-ary relations: $\mathcal{R} : \text{Prop}^n \to \{\top, \bot\}$ in question space, $\mathcal{S} : [0,1]^n \to \{\top, \bot\}$ in forecast space, and a predicate for $\mathbb{F}$ such that $\mathcal{R}(x_1, \dots x_n) \implies \mathcal{S}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$. In particular, this assertion must be satisfied by all feasible θ, and also any "correct" forecasts generated by a world model that accurately accounts for aleatoric uncertainty. Violation of consistency is measured by some violation metric $\mathcal{V} : [0,1]^n \to \mathbb{R}$ which must satisfy $\mathcal{V}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n)) = 0 \iff \mathcal{S}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$. For example, intuitively, the "negation" check Negation is given by the relation $\mathcal{R}(x_1, x_2) := x_1 = \neg x_2$ on questions, and the relation $\mathcal{S}(\mathbb{F}(x_1), \mathbb{F}(x_2)) := \mathbb{F}(x_1) + \mathbb{F}(x_2) \approx 1$ on forecasts. The full table of the consistency checks we use is given in Appendix 6.8.1.

Improving upon Fluri et al. (2023), we derive $\mathcal{V}$ from $\mathcal{R}$ in a principled way, handling all types of logical consistency checks simultaneously. We introduce two new *inconsistency metrics*: the *arbitrage metric* and the *frequentist metric* for measuring logical inconsistency in probabilistic forecasts.

Arbitrage metric

The arbitrage metric is conceptualized as the minimum profit that an arbitrageur can be guaranteed making bets against the forecaster's predictions. More precisely: suppose that the forecaster's probabilities $\mathbb{F}(x_1), \dots \mathbb{F}(x_n)$ were prices offered by a logarithmic market maker [24] with market subsidy parameter \$1. If these probabilities are inconsistent, then there are prices $p_1, \dots p_n$ that an arbitrageur could bring to the market such that it is guaranteed to make a profit against the market-maker, no matter the outcome of each question. We define $\mathcal{V}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$ as the maximum achievable "minimum profit" that the arbitrageur can guarantee by choosing appropriate $p_1, \dots p_n$. We further denote by $\mathcal{A}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$ the set of prices $p_1, \dots p_n$ that maximize the minimum profit:

$$(\arg\max_{p \in [0,1]^n}, \max) \min_{\omega \in \Omega} \sum_{i=1}^{n} \left(\log p_i - \log \mathbb{F}(x_i)\right) \delta_{\omega(i)=\top} + \left(\log\left(1 - p_i\right) - \log\left(1 - \mathbb{F}(x_i)\right)\right) \delta_{\omega(i)=\bot} \tag{3}$$

Here $\Omega := \{\omega \in \Theta'^n \mid \mathcal{R}(\omega)\}$ is the set of all possible consistent resolutions of this tuple. A more general version of 3 is given in Appendix 6.8.2, along with specific worked-out examples of the arbitrage metric for each consistency check, and details on how we compute it; as an example, the arbitrage metric for the Negation Check can be derived exactly (Appendix 6.8.2):

$$\mathcal{V}(\mathbb{F}(x), \mathbb{F}(\neg x)) = -2 \log\left(\sqrt{\mathbb{F}(x)(1 - \mathbb{F}(\neg x))} + \sqrt{(1 - \mathbb{F}(x))\mathbb{F}(\neg x)}\right)$$

To illustrate: $\mathcal{V}(0.5, 0.6) \approx 0.01$, $\mathcal{V}(0.5, 0.51) \approx 10^{-4}$. The metric is more sensitive to violations for probabilities very close to 0 or 1, due to the logarithmic market maker. In our evals, for all types of checks, we say that a sampled check does not pass if $\mathcal{V} \geq 0.01$. We have to pick some hyperparameter as an inconsistency threshold; we set it to correspond to giving 110% probability in total to the events of Republican and Democratic parties winning the US presidential election.

Frequentist metric

We also compute a different, *frequentist* consistency metric. Consider a Monte Carlo forecaster that samples a world model n times, and for any event, returns the fraction of samples in which the event occurs. The frequentist metric is the number of standard deviations a given tuple forecast is off from the mean Monte Carlo forecast, scaled to be independent of n. We say that a consistency violation happened if the number of standard deviations away from

[24] A *logarithmic market maker* with subsidy w is a market maker who adjusts prices in response to trades such that the trader's reward for moving the probability of a true-resolving sentence from p_0 to p' is $w \log p' - w \log p_0$. For further background on scoring rules and the associated market makers, see Appendix 6.8.2, Berg and Proebsting (2009), or Hanson (2002).

the mean of the null is at least as in the $(0.5, 0.6)$ case described in Section 6.2.1. The full description is given in Section 6.8.3.

Intuition on consistency metrics

Our metrics address two major obstacle with measuring inconsistency: *tolerance to noise* and *principled aggregation of inconsistency scores*.

Tolerance to noise. In the standard Bayesian setting, beliefs are either consistent or not: there either is a Dutch book (a way to bet against the forecaster's beliefs to get infinite profit) or the probabilities are perfectly consistent. In practice, an forecaster's beliefs (even on the same question) are never perfectly consistent across runs. If an election model has a presidential candidate at 48% with one random seed and 50% on the other, this is not a reason to discard it as completely flawed. Hence, instead of a binary measure of consistency, our metrics increase smoothly with inconsistency.

Principled aggregation and comparison of inconsistency scores. Fluri et al. (2023) developed a set of inconsistency checks, used an ad-hoc metric for each check they used, and normalized the scores to [0, 1]. There are two major issues with their approach:

1. The metrics in their work are mostly *linear* and would treat the inconsistencies of $(0.5, 0.6)$ and $(0.89, 0.01)$ on the NEGATION check the same, which is counter-intuitive in many applications.
2. It is unclear how to compare and aggregate scores from different consistency checks.

Our approach ensures that all consistency scores share a common "unit". For example, in the arbitrage metric, to aggregate inconsistencies, we sum up the profit made by an arbitrageur across questions.

6.3 Pipeline overview

We illustrate the steps in our data collection pipeline below, and provide more details on each individual steps:

$$\boxed{\textit{Online platforms, news, topics}} \xrightarrow[+\text{scraping}]{\text{synthetic}} P \xrightarrow[\text{instantiation}]{\text{tuple}} (P, Q) \xrightarrow{\mathbb{F}} (p, q) \xrightarrow{\mathcal{V}} \mathcal{V}(p, q)$$

- $(\cdots \longrightarrow P)$ We first prepare datasets of **base questions** in multiple ways:
  (a) Scraping questions from online platforms such as Manifold and Metaculus;
  (b) A ground-truth resolved dataset synthetically generated from news articles;
  (c) Synthetic generation on questions on a list of topics such as Politics, Science, Economics, etc.

  For the first two of the above, we also include the *ground truth resolution* for each question.
- $(P \longrightarrow (P, Q))$ The base questions are synthetically **instantiated into tuples** that must satisfy certain consistency checks. For example, every single base question P is instantiated into a tuple $(P, \neg P)$; and pairs of mutually relevant base questions P, Q are instantiated into tuples like $(P, Q, P \wedge Q, P \vee Q)$.
- $((P, Q) \xrightarrow{\mathbb{F}} (p, q))$ The forecaster is separately queried to elicit **forecasts** on each question, resulting in forecast tuples that should, if the forecaster is consistent, satisfy consistency properties. For example, for a size-two tuple where $Q = \neg P$, it should satisfy $p + q = 1$.
- $((p, q) \xrightarrow{\mathcal{V}} \mathcal{V}(p, q))$ We score each tuple of forecasts for consistency with both of our violation metrics.

6.4 Results

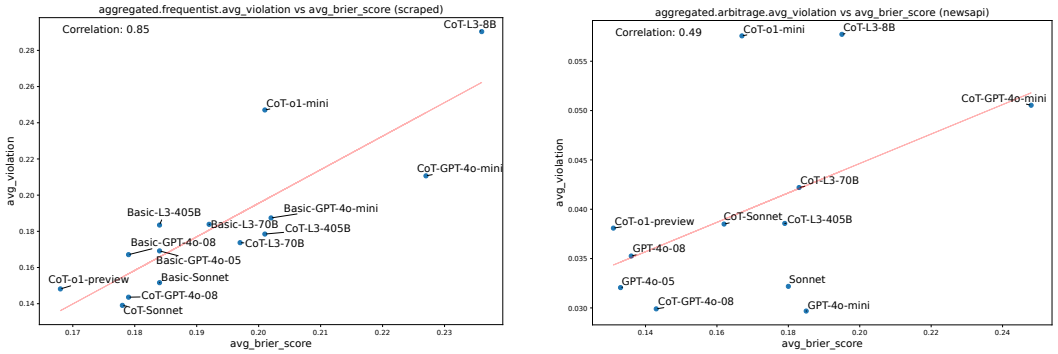
We evaluate a range of forecasters on the datasets described above, for both consistency and ground truth Brier score. We note that the Brier score as the standard metric of forecasting

accuracy depends both on model capabilities and the training data cutoff: it should not be surprising for a stronger model to have a worse Brier score if its training data cutoff is earlier than for a weaker model.

For all data analysis in this section, we exclude forecasters that have Brier score worse than random guessing (0.25), such as the basic setup with `llama-3.1-8B`, as it would unfairly advantage our case of "correlating consistency with accuracy".

Average consistency scores correlate strongly with forecasting performance. We can aggregate the consistency scores across all checks for each forecaster by aggregating either the arbitrage or the frequentist violations.

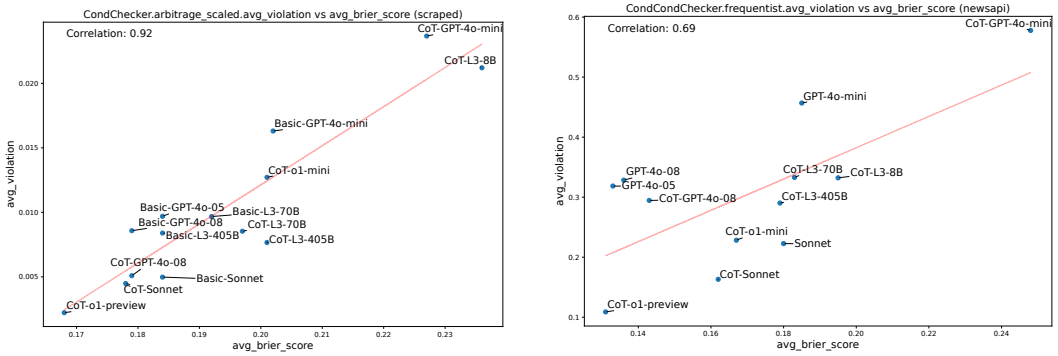
We plot the average Brier score against the three aggregate consistency scores in Figure 4.



(a) Aggregate frequentist metric on the scraped forecasting question dataset.

(b) Aggregate arbitrage metric on the NewsAPI questions dataset.

Figure 4: Scatter plots showing the relationship between consistency metrics and average Brier scores for different forecasters. Each point represents a forecaster, with the x-axis showing the average Brier score and the y-axis showing the consistency metric . The y-axis values are aggregated across all checks for each forecaster and averaged over the instantiated consistency check tuples. Lower scores are better for both axes.



(a) Cond arbitrage metric on the scraped forecasting question dataset.

(b) CondCond frequentist metric on the NewsAPI questions dataset.

Figure 5: Both Cond and CondCond consistency metrics see Table 1 show a strong correlation with forecasting accuracy as measured by the Brier score.

Bayesian consistency checks are the best proxies for forecasting performance. Figure 5a shows the strong correlation between certain consistency checks from Table 1 and average Brier scores across different forecasters. This relationship suggests that Cond, which tracks logical consistency in conditional probability estimates, serves as a proxy for overall forecasting accuracy, *without knowing how the questions resolved.*

Certain consistency metrics are not well correlated with forecasting performance. The measured correlations between the consistency checks and Brier scores are given in Table 2. We see that some checks yield higher signal on the ground truth performance than others. Aggregating different consistency metrics seems to improve the correlation.

We note that the selection of forecasters we test is quite limited, so we cannot guarantee the trends here are representative of future LLM forecasters. Part of the correlation can be attributed to better models being both more consistent and better forecasters. For comparison, the correlations of the Brier score of our forecasters and the MMLU Hendrycks et al. (2020) (college split) error rate on the best approximation of the tested forecasters are 0.38 and 0.55 on the NewsAPI and scraped datasets, respectively.

We include all data (questions, tuples, forecasts, and scores) in the supplementary material.

Table 2: Correlation of consistency metrics with Brier score, across both of our base question datasets and the derived consistency check tuples.

	Scraped		NewsAPI	
	Arbitrage	Frequentist	Arbitrage	Frequentist
Negation	0.60	0.67	-0.36	-0.13
Paraphrase	0.57	0.61	0.13	0.24
Consequence	0.51	0.52	0.21	0.30
AndOr	0.20	0.25	0.02	0.06
And	0.68	0.72	0.54	0.71
Or	0.14	0.24	-0.24	-0.31
But	0.20	0.67	0.63	0.77
Cond	0.92	0.87	0.71	0.69
CondCond	0.78	0.71	0.75	0.69
ExpEvidence	0.20	0.77	-0.11	0.06
Aggregated	0.62	0.85	0.49	0.66

Even good reasoning models are inconsistent. We give the full set of consistency metrics for OpenAI's `o1-mini` in Table 3. The `Frac` column counts the fraction of tuples for which the violation exceeded a certain threshold; see the full exposition of what the thresholds mean in Sections 6.8.2 and 6.8.3. The frequentist metric is not directly comparable to the arbitrage metric, but the respective violation counts ("Frac" in the table) are.

Table 3: Consistency metrics for o1-mini.

	Scraped				NewsAPI			
	Arbitrage		Frequentist		Arbitrage		Frequentist	
Check	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac
Negation	0.07	58%	0.26	61%	0.08	52%	0.27	56%
Paraphrase	0.07	56%	0.26	61%	0.06	53%	0.24	56%
Consequence	0.03	27%	0.13	29%	0.03	18%	0.10	19%
AndOr	0.09	65%	0.34	71%	0.07	57%	0.29	67%
And	0.02	24%	0.11	27%	0.03	23%	0.11	24%
Or	0.11	48%	0.30	50%	0.05	48%	0.21	50%
But	0.11	60%	0.40	79%	0.11	63%	0.38	80%
Cond	0.04	41%	0.22	52%	0.07	66%	0.29	70%
CondCond	0.03	30%	0.19	45%	0.04	54%	0.23	71%
ExpEvidence	0.04	47%	0.27	69%	0.05	45%	0.28	63%
Aggregated	0.06	–	0.25	–	0.06	–	0.24	–

OpenAI's `o1-mini` forecaster, despite being one of the best reasoning models so far, violates consistency checks more than the $(0.5, 0.6)$ threshold from Section 6.2 very often.

6.5 ArbitrageForecaster: Can we design a more consistent forecaster?

Let $(x_1, \dots x_n)$ be a question tuple for some consistency check $\mathcal{R}$, e.g. $(P, \neg P)$. Given forecasts $\mathbb{F}(x_1), ...\mathbb{F}(x_n)$, the arbitrage metric maximization problem in Equation 3 computes the following (as the argmax and max of the arbitrage respectively):

1. Improved forecasts $\mathbb{F}'(x_1), ...\mathbb{F}'(x_n)$ which are consistent, i.e. satisfy $\mathcal{S}$; and
2. The profit earned by an arbitrageur who bets these improved forecasts against the original ones – this is the actual metric.

This leads us to wonder: *can we use these "improved consistent forecasts" to build a new forecaster which builds on the base forecaster $\mathbb{F}$, but is more consistent on $\mathcal{R}$?*

We introduce: the `ArbitrageForecaster` with base $\mathbb{F}$ arbitraged on consistency check $\mathcal{R}$, denoted by $\langle\mathbb{F}\rangle_{\mathcal{R}}$, which computes its forecast on a question x as follows:

1. Instantiates a tuple $(x_1, \dots x_n)$ satisfying $\mathcal{R}$;
2. Queries $\mathbb{F}$ to obtain $\mathbb{F}(x_1), \dots \mathbb{F}(x_n)$;
3. Arbitrages these base forecasts per Eq 3 and returns the arbitraged forecast for x_1.

Despite what one might assume, however, an `ArbitrageForecaster` is *not* "definitionally" consistent on the check it is arbitraged on, but rather its consistency must be investigated empirically. Suppose, for example, that a forecaster produces forecasts $\mathbb{F}(P) = 0.5$, $\mathbb{F}(\text{para}(P)) = 0.6$, $\mathbb{F}(\text{para}(\text{para}(P))) = 0.7$. Then $\mathbb{F}' := \langle\mathbb{F}\rangle_{\text{Paraphrase}}$ would produce forecasts $\mathbb{F}'(P) \approx 0.55$, $\mathbb{F}'(\text{para}(P)) \approx 0.65$, which are not consistent.

Appendix 6.8.4 contains a precise definition of `ArbitrageForecaster`, including the case of sequentially arbitraging on multiple checks $\langle\mathbb{F}\rangle_{[\mathcal{R}_1,...\mathcal{R}_s]}$, and a theoretical discussion of its consistency properties. In particular, we list strong theoretical reasons to expect consistency gains from *recursive* `ArbitrageForecaster` setups, i.e. $\langle\mathbb{F}\rangle^r_{\mathcal{R}} := \langle\langle\mathbb{F}\rangle^{r-1}_{\mathcal{R}}\rangle_{\mathcal{R}}$, in particular with Negation, as well as in a non-recursive `ArbitrageForecaster` with ExpEvidence.

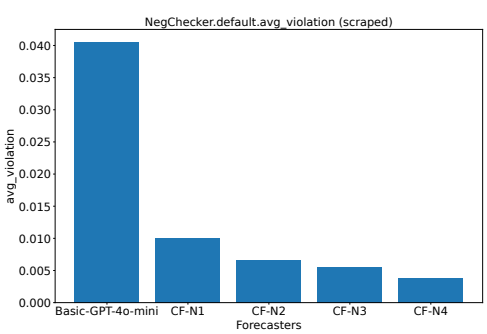
Due to these priorities and the high costs of running recursive `ArbitrageForecaster`s (see Appendix 6.8.4), we limited ourselves to studying only a small number of `ArbitrageForecaster` setups, with a limited number of checks rather than the whole list; specifically: $\langle g\rangle^r_N$, $\langle g\rangle^r_P$, $\langle g\rangle^r_{[N,P]}$, $\langle g\rangle_{[E]*s}$ where $g :=$`gpt-4o-mini`, N, P, E are Negation, Paraphrase, ExpEvidence respectively, and r and s vary from 0 to 4.

The full results of our experiments with these forecasters are reported in Appendix 6.8.4; our key takeaways from these preliminary runs look hopeful:

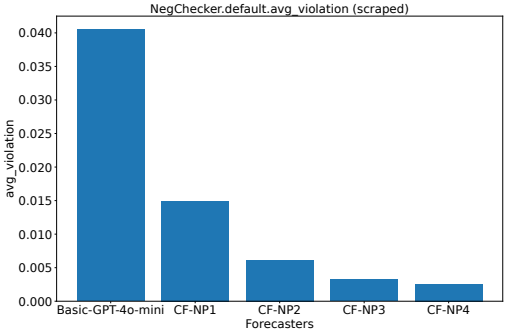
- In the case of the checks we tested, **arbitraging on a check indeed makes a forecaster more consistent on that check**, with increasing consistency gains with recursive depth, as shown in Fig 6. Crucially, this also applied when the arbitraging was on more than a single check: $\langle g\rangle^r_{[N,P]}$ did well on *both* Negation and Paraphrase; arbitraging on the next check did not increase inconsistency on the first. We are cautiously optimistic that this may extend to the full list of checks in Table 1.
- **This consistency gain was greatest with Negation, followed by Paraphrase, and lowest with ExpEvidence.** This finding is in line with our hypothesis in Appendix 6.8.4 that `ArbitrageForecaster` would be particularly effective on consistency checks which are *symmetric*. and instantiate *deterministically*.
- **We do not observe reliable improvements on ground truth forecasting performance, or on consistency checks other than the ones we arbitrage on.** I.e. $\langle\mathbb{F}\rangle_{\mathcal{R}_1}$ does not reliably do better on $\mathcal{R}_2$.

6.6 Related work

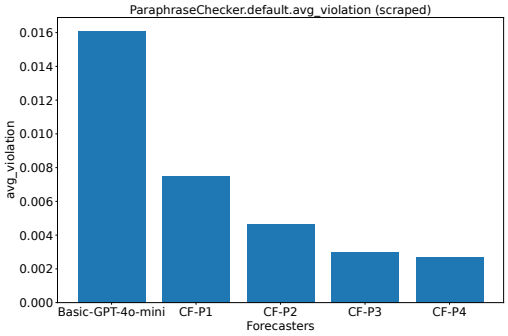
Metamorphic and consistency checks. Checking logical properties of outputs of programs under semantic-preserving transforms has a long history (Chen et al., 1998). Before Fluri



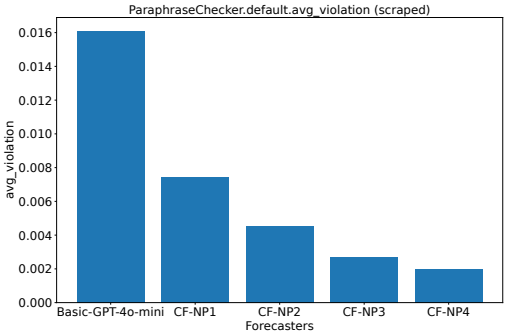
(a) Average violation of $\langle g \rangle_N^r$ (denoted CF-Nr) on NEGATION for r from 0 to 4.



(b) Average violation of $\langle g \rangle_{NP}^r$ (denoted CF-NPr) on NEGATION for r from 0 to 4.



(c) Average violation of $\langle g \rangle_P^r$ (denoted CF-Pr) on PARAPHRASE for r from 0 to 4.



(d) Average violation of $\langle g \rangle_{NP}^r$ (denoted CF-NPr) on PARAPHRASE for r from 0 to 4.

Figure 6: NEGATION and PARAPHRASE violations for various `ArbitrageForecaster` setups. In all captions, g denotes `gpt-4o-mini`, N, P denote NEGATION and PARAPHRASE respectively, and the definition of the `ArbitrageForecaster` setup is as in Def 6.4.

et al. (2023), variants of the consistency check framework were used for simple ML models (Christakis et al., 2022; Sharma and Wehrheim, 2020), vision (Hendrycks and Dietterich, 2019), and chat LLMs (Jang and Lukasiewicz, 2023), among other areas. Li et al. (2019) consider logical consistency checks beyond paraphrasing and negation for simple ML models.

Forecasting and large language models. LLMs and forecasting date back to Zou et al. (2022a) and Yan et al. (2023a). Recently, strong performance of LLM forecasters on prediction market datasets has been claimed in (Halawi et al., 2024a; Tetlock et al., 2024; Hsieh et al., 2024; Phan et al., 2024). Concurrent with our work, Karger et al. (2024) have introduced an automatically updating benchmark for forecasting.

Scalable oversight and failures of superhuman AI. The difficulty of evaluating models with superhuman performance in domains without a source of ground truth has long been acknowledged, and falls under the umbrella of *scalable oversight* (Amodei et al., 2016). Forecasting using AI oracles is one such domain. The use of consistency checks for scalable oversight has been studied in the simpler context of superhuman game AIs (Lan et al., 2022; Fluri et al., 2023), and in general question-answering tasks via debate (Irving et al., 2018a).

Consistency evaluations for LLMs. Even on tasks where the ground truth is in principle knowable, consistency evaluations have long helped in cases where checking consistency is easier than getting the ground truth labels (Elazar et al., 2021; Li et al., 2023). Raj et al. (2023) measure paraphrasing consistency and ground truth accuracy on TruthfulQA (Lin et al., 2021) and find little to no correlation. Some forms of consistency checks have been applied on model internals to discover features related to LLM truthfulness and reliability (Burns et al., 2022; Kaarel et al., 2023).

6.7 Future work

We have developed a comprehensive benchmark of *static consistency checks* for LLM forecasters, and demonstrated its correlation with ground truth accuracy, suggesting that our consistency metrics could serve as a proxy for accuracy when we do not have access to ground truth. We envision several directions in which our framework could be extended:

Consistency in decision-making. AI systems may be used not only to make forecasts that inform decisions, but also to take decisions directly. Here too, we can have a notion of inconsistency: for example, *intransitive preferences* [25] – and analogously, an inconsistent decision-maker may be exploited by an arbitrageur.

Training for consistency. Modulo consideration of the cost-benefit to safety, our methods could be used train LLMs for consistency, minimizing our violation metrics. This may or may not impact overall forecasting performance and other AI capabilities. One may also imagine an AlphaZero-style set-up, where an LLM $\mathbb{F}$ is trained on the outputs of $\langle\mathbb{F}\rangle^r$, i.e. a recursive `ArbitrageForecaster` wrapped around it.

Further experiments with `ArbitrageForecaster`. Most of our experiments with `ArbitrageForecaster` involved arbitraging on only a *single* check (apart from one experiment with both Negation and Paraphrase), due to the cost limitations described in 6.8.4. It is easy to imagine how a bad forecaster could still overfit a single check: simply forecasting 50% probability for all questions will pass Paraphrase, ExpEvidence and Negation – but we expect that being consistent under a variety of checks is difficult without a consistent world model. One approach to using more checks cheaply, particularly in training, may be to *randomly sample* a number of consistency checks for each question.

Dynamic generation of consistency checks. Although we found strong correlations between ground truth accuracy and consistency among existing LLM forecasters, our results with `ArbitrageForecaster` demonstrate that this isn't necessarily the case: it is possible to do well on consistency without improving ground truth. In particular, this means that consistency as a training metric could be "Goodharted" by a learning AI model (Karwowski et al., 2023). One way to prevent this may be via adversarial training: i.e. have an adversarial agent instantiate consistency checks that it believes the agent will perform poorly on.

Evaluating RAG-augmented forecasters. We have conducted some preliminary experiments evaluating state-of-the-art forecasters such as Halawi et al. (2024a). Unfortunately, we could not reproduce the system from Halawi et al. (2024a) at the time of writing, due to deprecations in the Google News API (we could not obtain access to the alternative Newscatcher API). At the time of writing, we are not aware of other publicly-available LLM forecasting systems that are competitive with the results of Halawi et al. (2024a) (there exist proprietary systems that may be competitive, such as FutureSearch (2024)). We thus leave the evaluation of better forecasters like Halawi et al. (2024a) and Phan et al. (2024) to future work, once such forecasters are more widely available.

Contributions

DP and APS developed consistency checks and the arbitrage and frequentist metrics. DP, AA, APS, and EW worked on the LLM question to evaluation pipeline. APS thought of and implemented `ArbitrageForecaster`. VB created the news-derived question dataset. AS and DP created the scraped question dataset. AA and DP created the 2028 synthetic question dataset. DP started and led the project. FT proposed correlating consistency with forecasting accuracy and advised the project. All authors helped with the writing. DP and APS wrote the first draft of the paper.

[25] See e.g. Fishburn (1970) and the Von Neumann–Morgenstern utility theorem for an introduction to decision rationality.

Acknowledgements

We thank Danny Halawi for extensive discussions and help with our setup. We thank Brendan Murphy, Ezra Karger, Fred Zhang, and Tatsunori Hashimoto for helpful discussions and feedback on the paper and forecasting in general. We thank Berkeley SPAR for connecting collaborators, and BERI for partially funding the project.

6.8 Appendix

6.8.1 Table of consistency checks

Table 4 (extended version of Table 1) includes all the consistency checks tested for in our benchmark. In most of them, we leave the logical relations between forecasting questions $\mathcal{R}$ implicit by constructing the sentences directly. For instance, $\mathcal{R}(x_1, x_2) := x_1 = \neg x_2$ is implied by simply writing x_1, x_2 as $P, \neg P$. In the rest of the appendix, we use the sentence-based (P, Q instead of x_1, x_2) notation.

Table 4: Consistency checks and the logical consistency conditions.

Name	**Tuple**	**Condition ($\mathcal{S}$)**
Negation	$(P, \neg P)$	$\mathbb{F}(P) + \mathbb{F}(\neg P) = 1$
Paraphrase $\mathcal{R}(P, Q) := P \iff Q$	(P, Q)	$\mathbb{F}(P) = \mathbb{F}(Q)$
Consequence $\mathcal{R}(P, Q) := P \implies Q$	(P, Q)	$\mathbb{F}(P) \leq \mathbb{F}(Q)$
AndOr	$(P, Q, P \wedge Q, P \vee Q)$	$\mathbb{F}(P) + \mathbb{F}(Q) = \mathbb{F}(P \vee Q) + \mathbb{F}(P \wedge Q)$
And	$(P, Q, P \wedge Q)$	$\max(\mathbb{F}(P) + \mathbb{F}(Q) - 1, 0) \leq \mathbb{F}(P \wedge Q) \leq \min(\mathbb{F}(P), \mathbb{F}(Q))$
Or	$(P, Q, P \vee Q)$	$\max(\mathbb{F}(P), \mathbb{F}(Q)) \leq \mathbb{F}(P \vee Q) \leq \min(1, \mathbb{F}(P) + \mathbb{F}(Q))$
But	$(P, \neg P \wedge Q, P \vee Q)$	$\mathbb{F}(P \vee Q) = \mathbb{F}(P) + \mathbb{F}(\neg P \wedge Q)$
Cond	$(P, Q\|P, P \wedge Q)$	$\mathbb{F}(P)\mathbb{F}(Q\|P) = \mathbb{F}(P \wedge Q)$
CondCond	$(P, Q\|P, R\|(P \wedge Q), P \wedge Q \wedge R)$	$\mathbb{F}(P)\mathbb{F}(Q\|P)\mathbb{F}(R\|P \wedge Q) = \mathbb{F}(P \wedge Q \wedge R)$
ExpEvidence	$(P, Q, P\|Q, P\|\neg Q)$	$\mathbb{F}(P) = \mathbb{F}(P\|Q)\mathbb{F}(Q) + \mathbb{F}(P\|\neg Q)(1 - \mathbb{F}(Q))$

The consistency checks in Table 4 represent core logical relationships between probabilities, but many other forms of consistency checks are possible. Here are two examples that could extend our framework:

- **Comparative checks:** Building on generator-validator checks from Li et al. (2023), we could ask a forecaster to predict both $\mathbb{F}(P)$, $\mathbb{F}(Q)$, and separately whether P or Q is more likely. The forecaster's probability estimates should match their comparative judgment.
- **Monotonicity checks:** Fluri et al. (2023) propose a variant of Consequence for real-valued quantities, where predictions must respect the monotonic ordering of a sequence of future values. This connects to *scope insensitivity* (Kahneman et al., 2000), a cognitive bias where humans fail to scale probability estimates appropriately with the magnitude of outcomes.

We do not include a specific consistency check for Bayesian updates, as conditional probabilities are already covered by Cond, CondCond, and ExpEvidence.

6.8.2 Arbitrage as a violation metric

For the following definition we use a slightly more general notation than in the main body, to convey that our methods could be generalized beyond binary forecasting questions.

Notation. Let Prop denote the set of forecasting questions we are interested in, Θ denote the set of possible outcomes/resolutions for an individual question, and Forecast denote the set of probability distributions on Θ. A *Forecaster* is a map $\mathbb{F} : \text{Prop} \to \text{Forecast}$. For conditional questions that can resolve to None, we also have optional resolutions $\Theta' := \Theta \cup \{\text{None}\} = \{\top, \bot, \text{None}\}$.

The arbitrage metric may be seen as being motivated by Dutch Book Arguments for probabilistic consistency rules (see e.g. Vineberg (2022)). Imagine the forecaster's predictions $\mathbb{F}(x_1), \dots \mathbb{F}(x_n)$ were prices offered by a bookie on prediction markets for sentences $x_1, \dots x_n$. If these probabilities are inconsistent, then there are bets that an arbitrageur can make that guarantee a profit in *all possible (consistent) worlds regardless of the individual outcomes*. For example, if x_1, x_2 are two sentences such that $x_1 \iff x_2$, but the bookie prices $\mathbb{F}(x_1) < \mathbb{F}(x_2)$, then an arbitrageur can simply buy x_1 and sell x_2 to make a risk-free profit.

However, if the bookie never changes their prices in response to trades, the arbitrageur can make an infinite amount of profit with its strategy. This is neither realistic nor useful for creating a metric to measure inconsistency. Instead, we turn to *market scoring rules*, introduced in Hanson (2002)), where the bookie is a *market-maker* who updates market prices in a way that ensures that the reward for moving the market price of a sentence that resolves True from p_0 to p' is given by a *proper scoring rule* [26] $s(p') - s(p_0)$. We then define our inconsistency metric to be the minimum profit an arbitrageur can guarantee against such a market-maker, if the latter offers inconsistent probabilities $\mathbb{F}(x_1), \dots \mathbb{F}(x_n)$.

Definition 6.1 (Arbitrage-based Violation Metric)**.** Let $\mathcal{R} : \text{Prop}^n \to \{\top, \bot\}$ be an n-ary relation such that $\mathcal{R}(\theta(x_1), \dots \theta(x_n))$ is satisfied by the ground-truth resolutions $\theta : \text{Prop} \to \Theta$ for all tuples $(x_1, \dots x_n)$. [27] Let $s : \text{Prop} \times \Theta \times [0,1] \to \mathbb{R}$ be a proper scoring rule that gives the score earned based on the probability assigned to the true resolution, e.g. $s(x, \theta, p(\theta)) = \log p(\theta)$. Let $(x_1, \dots x_n) \in \text{Prop}^n$ be a question tuple, and denote $\Omega := \{\omega \in \Theta'^n \mid \mathcal{R}(\omega)\}$ the set of possible consistent resolutions (including None resolutions) of this tuple. Then for forecasts $(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$ the arbitraged forecasts $\mathcal{A}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n)) = (p_1 \dots p_n)$ and the minimum guaranteed profit of the arbitrageur $\mathcal{V}(\mathbb{F}(x_1), \dots \mathbb{F}(x_n))$ are given by:

$$(\arg\max, \max)_{p \in \text{Forecast}^n} \min_{\omega \in \Omega} \sum_{i=1}^{n} s\left(x_i, \omega_i, p_i(\omega_i)\right) - s\left(x_i, \omega_i, \mathbb{F}(x_i)(\omega_i)\right) \tag{4}$$

Where by convention, any score on a resolution $\omega_i = \text{None}$ is taken to be 0.

Definition 6.1 is presented in full generality: p and $\mathbb{F}(x_i)$ here are *probability distributions* on Θ. Breaking it down: each $s\left(x_i, \omega_i, p_i(\omega_i)\right) - s\left(x_i, \omega_i, \mathbb{F}(x_i)(\omega_i)\right)$ gives the arbitrageur's profit on the market for question x_i, given that it resolves ω_i. The profit is summed across all markets in the tuple, and then minimized over all consistent worlds; this minimum is maximized across all possible arbitrageur bets.

It is helpful to explicitly state Eq 4 in the case of binary forecasting questions, as follows.

$$(\arg\max, \max)_{p \in [0,1]^n} \min_{\omega \in \Omega} \sum_{i=1}^{n} \left(s\left(p_i\right) - s\left(\mathbb{F}(x_i)\right)\right) \delta_{\omega(i)=\top} + \left(s\left(1-p_i\right) - s\left(1-\mathbb{F}(x_i)\right)\right) \delta_{\omega(i)=\bot} \tag{5}$$

[26] A proper scoring rule (Savage, 1971), is one that incentivizes honest reporting of probabilities: widely used proper scoring rules include the Brier score $(1-p)^2$ and the logarithmic scoring rule $-\log p$.

[27] This is well-defined because resolutions can be taken as a subset $\Theta \subseteq \text{Prop}$, by treating them as forecasting questions that always resolve to themselves by definition. For example, the forecasting question $\top$ is always worth \$1 and the forecasting question $\bot$ is always worth \$0.

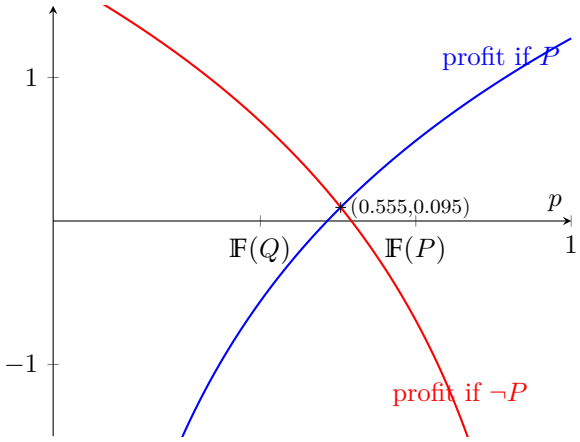


Figure 7: Profit earned by the arbitrageur in case of inconsistency over ParaphraseChecker, taking $s(p) = \log(p)$ and $\mathbb{F}(P), \mathbb{F}(Q) = 0.7, 0.4$ in (6).

.

We will illustrate our violation metric with three specific examples, for PARAPHRASE, NEGATION and COND. For other consistency checks, the math becomes too convoluted and we use a numerical method in our project code.

ParaphraseChecker

Let P and Q be equivalent sentences, and suppose that the forecaster produces forecasts $\mathbb{F}(P)$ and $\mathbb{F}(Q)$. A trader who instead brings prices to $\mathbb{F}'(P) = \mathbb{F}'(Q) = p$ for both questions earns a combined profit on both questions:

$$\begin{cases} s\left(p\right) - s\left(\mathbb{F}(P)\right) + s\left(p\right) - s\left(\mathbb{F}(Q)\right) & \text{if } P \\ s\left(1-p\right) - s\left(1-\mathbb{F}(P)\right) + s\left(1-p\right) - s\left(1-\mathbb{F}(Q)\right) & \text{if } \neg P \end{cases} \tag{6}$$

For this first example, we can graph this profit as a function of p for illustration, shown in Fig. 7 – demonstrating that any $p \in (0.529, 0.576)$ is profitable for the arbitrageur, and further that the arbitrageur can *guarantee* a minimum profit of 0.095 regardless of the outcome of P by choosing the consistent probability $p = 0.555$.

We may compute this intersection analytically:

$$\begin{aligned} s\left(p\right) - s\left(\mathbb{F}(P)\right) + s\left(p\right) - s\left(\mathbb{F}(Q)\right) &= s\left(1-p\right) - s\left(1-\mathbb{F}(P)\right) + s\left(1-p\right) - s\left(1-\mathbb{F}(Q)\right) \\ 2\log\frac{p}{1-p} &= \log\frac{\mathbb{F}(P)\mathbb{F}(Q)}{(1-\mathbb{F}(P))(1-\mathbb{F}(Q))} \\ p &= \frac{\sqrt{\mathbb{F}(P)\mathbb{F}(Q)}}{\sqrt{\mathbb{F}(P)\mathbb{F}(Q)} + \sqrt{(1-\mathbb{F}(P))(1-\mathbb{F}(Q))}} \end{aligned}$$

Substituting this back into either expression in (6) we get the expression for the arbitrage:

$$\mathcal{V}(\mathbb{F}(P), \mathbb{F}(Q)) = -2\log\left(\sqrt{\mathbb{F}(P)\mathbb{F}(Q)} + \sqrt{(1-\mathbb{F}(P))(1-\mathbb{F}(Q))}\right) \tag{7}$$

As a bonus, this can straightforwardly be extended to the multi-question paraphrasing check: $(P_1 \iff \cdots \iff P_n) \implies (\mathbb{F}(P_1) = \cdots = \mathbb{F}(P_n))$. Here the corresponding possible profits are:

$$\begin{cases} ns\left(p\right) - \sum s\left(\mathbb{F}(P_i)\right) & \text{if } P \\ ns\left(1-p\right) - \sum s\left(1-\mathbb{F}(P_i)\right) & \text{if } \neg P \end{cases} \tag{8}$$

Equating them and solving for p, we get:

$$\log \frac{p}{1-p} = \frac{1}{n}\sum_i \log \frac{\mathbb{F}(P_i)}{1-\mathbb{F}(P_i)} \tag{9}$$

$$p = \frac{\Delta}{\Delta+1} \text{ where } \Delta = \left[\prod_i \frac{\mathbb{F}(P_i)}{1-\mathbb{F}(P_i)}\right]^{1/n} \tag{10}$$

Observe that the arbitraged probability is simply the arithmetic mean in log-odds space! One may wonder if the violaton is some kind of variance measure in log-odds space, but this does not seem to be the case:

$$\mathcal{V}(\mathbb{F}(P_1),\dots\mathbb{F}(P_n)) = -n\log\left[\left(\prod \mathbb{F}(P_i)\right)^{1/n} + \left(\prod(1-\mathbb{F}(P_i))\right)^{1/n}\right] \tag{11}$$

NegChecker

Suppose the forecaster produces forecasts $\mathbb{F}(P)$ and $\mathbb{F}(\neg P)$. A trader who instead brings prices to $\mathbb{F}'(P) = p$, $\mathbb{F}'(\neg P) = 1-p$ earns a combined profit on both questions:

$$\begin{cases} s\left(p\right) - s\left(\mathbb{F}(P)\right) + s\left(p\right) - s\left(1-\mathbb{F}(\neg P)\right) & \text{if } P \\ s\left(1-p\right) - s\left(1-\mathbb{F}(P)\right) + s\left(1-p\right) - s\left(\mathbb{F}(\neg P)\right) & \text{if } \neg P \end{cases} \tag{12}$$

Equating them and solving as before,

$$2\log\frac{p}{1-p} = \log\frac{\mathbb{F}(P)(1-\mathbb{F}(\neg P))}{(1-\mathbb{F}(P))\mathbb{F}(\neg P)}$$

$$p = \frac{\sqrt{\mathbb{F}(P)(1-\mathbb{F}(\neg P))}}{\sqrt{\mathbb{F}(P)(1-\mathbb{F}(\neg P))} + \sqrt{(1-\mathbb{F}(P))\mathbb{F}(\neg P)}}$$

Substituting into (12), we get:

$$\mathcal{V}(\mathbb{F}(P),\mathbb{F}(\neg P)) = -2\log\left(\sqrt{\mathbb{F}(P)(1-\mathbb{F}(\neg P))} + \sqrt{(1-\mathbb{F}(P))\mathbb{F}(\neg P)}\right) \tag{13}$$

The similarity of these results to Paraphrase is suggestive: both the arbitraged probability and the violation for Negation can be derived from Paraphrase simply replacing $\mathbb{F}(Q)$ with $1-\mathbb{F}(\neg P)$, seeing the latter as the "probability implied for P by $\neg P$". This raises the natural question: Can *all* consistency checks be reduced to the case of Paraphrase arbitraging $\mathbb{F}(P)$ against an "implied probability" for P?

However, as we will see, Cond shows that this approach does not always hold. Its violation expression depends on more than just $\mathbb{F}(P)$ and $\mathbb{F}(P\wedge Q)/\mathbb{F}(Q \mid P)$, so there is no single, neat interpretation akin to "arithmetic mean in the log-odds space."

CondChecker

Suppose the forecaster produces forecasts $\mathbb{F}(P)$, $\mathbb{F}(Q \mid P)$, $\mathbb{F}(P\wedge Q)$. The possible outcomes Ω are $(P, Q \mid P, P\wedge Q) \mapsto (\top,\top,\top)$, $(\top,\bot,\bot)$, $(\bot,\text{None},\bot)$. Consider an arbitrageur who makes bets $\mathbb{F}'(P) = p$, $\mathbb{F}'(Q \mid P) = q$, $\mathbb{F}'(P\wedge Q) = pq$.

In each outcome:

$$\begin{cases} s\left(p\right) - s\left(\mathbb{F}(P)\right) + s\left(q\right) - s\left(\mathbb{F}(Q \mid P)\right) + s\left(pq\right) - s\left(\mathbb{F}(P \wedge Q)\right) & \text{if} P, Q \\ s\left(p\right) - s\left(\mathbb{F}(P)\right) + s\left(1-q\right) - s\left(1-\mathbb{F}(Q \mid P)\right) + s\left(1-pq\right) - s\left(1-\mathbb{F}(P \wedge Q)\right) & \text{if} P, \neg Q \\ s\left(1-p\right) - s\left(1-\mathbb{F}(P)\right) + s\left(1-pq\right) - s\left(1-\mathbb{F}(P \wedge Q)\right) & \text{if} \neg P \end{cases} \tag{14}$$

Equating these and rearranging:

$$\begin{cases} \frac{1-p}{p(1-q)} = \frac{1-\mathbb{F}(P)}{\mathbb{F}(P)(1-\mathbb{F}(Q|P))} =: A \\ \frac{1-q}{q}\frac{1-pq}{pq} = \frac{(1-\mathbb{F}(Q|P))(1-\mathbb{F}(P\wedge Q))}{\mathbb{F}(Q|P)\mathbb{F}(P\wedge Q)} =: B \end{cases}$$

Solving, where we indicate the right-hand-sides of each equation above by A and B respectively:

$$\begin{aligned} p &= \frac{1+\sqrt{B/(A+1)}}{1+\sqrt{B\cdot(A+1)}} \\ q &= \frac{1}{1+\sqrt{B/(A+1)}} \\ pq &= \frac{1}{1+\sqrt{B\cdot(A+1)}} \end{aligned}$$

Substituting back into 14 and simplifying:

$$\begin{aligned} &\mathcal{V}(\mathbb{F}(P), \mathbb{F}(Q \mid P), \mathbb{F}(P \wedge Q)) \\ &= -2\log\left(\sqrt{\mathbb{F}(P)\mathbb{F}(Q \mid P)\mathbb{F}(P \wedge Q)} + \sqrt{(1-\mathbb{F}(P)\mathbb{F}(Q \mid P))(1-\mathbb{F}(P \wedge Q))}\right). \end{aligned}$$

Numerical estimation

Explicitly deriving the violation metrics for other checkers from Equation (4) is infeasible by hand, and the expressions yielded by SymPy are very convoluted. For these checks, we use a numerical algorithm based on solving a differential equation for $p_i(t)$, as detailed below.

The arbitraging process may be understood as adjusting market prices in such a way that the scores in each possible $\omega \in \Omega$ remain equal throughout the process – i.e. such that their *derivatives* remain equal. For derivatives $p_i'(t)$ of the prices, the derivatives of each score $s_\omega'(t)$ are:

$$s_\omega'(t) = \begin{bmatrix} a_{\omega 1}(p_1) & \cdots & a_{\omega n}(p_n) \end{bmatrix} \cdot \begin{bmatrix} p_1'(t) \\ \vdots \\ p_n'(t) \end{bmatrix}$$

Where

$$a_{\omega i}(p_i) = \begin{cases} s'(p_i) & \text{if } \omega_i = \top, \\ -s'(1-p_i) & \text{if } \omega_i = \bot, \\ 0 & \text{if } \omega_i = \text{N/A} \end{cases}$$

Then, where $A(\mathbf{p}) = [a_{\omega i}(p_i)]$ (with Ω rows and n columns), we have the derivative of the score vector $\mathbf{s}'(t) = A(\mathbf{p})\mathbf{p}'(t)$. We want $\mathbf{s}'(t)$ to be a multiple of $[1 \quad \cdots \quad 1]$ to ensure it

is the same in all outcomes ω – the coefficient of proportionality does not matter (it just controls the "speed" at which you reach the arbitraged probabilities), so we can just solve $\mathbf{p}'(t) = A^{-1}\mathbf{s}'(t)$.

The dynamics of the arbitraging process are then simply:

$$p_i(0) = \mathbb{F}(x_i) \quad \text{(initial conditions)}$$

$$\mathbf{p}'(t) = A(\mathbf{p})^{-1} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Which we run until $\det A$ reaches 0, which is when consistency is reached.

6.8.3 Frequentist consistency metric

In a deterministic world, we cannot let any inconsistency pass; every time we prove any rule of probability does not hold exactly, we must discard the forecaster as flawed. This is too strict for the consistency check framework to be useful. Instead, we propose a violation metric and the corresponding inconsistency threshold based on statistical hypothesis testing.

Assume that each event P has a true probability value $\mathbb{T}(P)$, say under some world model that accounts for aleatoric uncertainty.

Definition 6.2 (Frequentist consistency)**.** A frequentist-consistent forecaster $\mathbb{F}$ samples a Gaussian estimate $\mathbb{T}(P) + \varepsilon$ of each event P, with variance $\sigma^2\mathbb{T}(P)(1-\mathbb{T}(P))$ for a hyperparameter σ^2:

$$\mathbb{F}(P) - \mathbb{T}(P) \sim \mathsf{N}\left(0, \sigma^2\mathbb{T}(P)(1-\mathbb{T}(P))\right) \quad \text{independently for all events } P. \tag{15}$$

This is principled from the frequentist perspective. Consider a forecaster that just samples the (relevant subset of) the world n times using the best available world simulator, and estimates the probability of each event P as the proportion of times that P occurs in the n samples. If we estimate the probability as the average chance of an event P with true probability p occurring out of n times, then this estimate has a scaled binomial distribution with mean p and variance $p(1-p)/n$. To reach Equation (15), replace the averaged binomial with the Gaussian of the same variance, and denote $\sigma^2 := 1/n$.

This simple model enables us to derive hypothesis tests for each of the consistency checks described in Table 1. The null hypothesis is always that the forecaster is frequentist-consistent. Note that σ^2 is not our estimate of the variance of any forecaster; it is just a hyperparameter that controls how strict our null hypothesis is. We leave estimating the variance of a particular forecaster and testing frequentist consistency based on that alone to future work.

Notation The expression $a\mathsf{N}(0, c^2)$ denotes a Gaussian random variable with mean 0 and variance a^2c^2. The expression $a\mathsf{N}(0, c^2) + b\mathsf{N}(0, c^2)$ denotes a Gaussian random variable with mean 0 and variance $a^2c^2 + b^2c^2$. All sums range over the cyclic permutations of the variables under the sum. All $\mathsf{N}(0, c^2)$ terms appearing with the same power of σ are independent. Two $\mathsf{N}(0, c^2)$ terms appearing with a different power of σ may be correlated; this is not important for our purposes, since we discard high-order powers of σ.

Bootstrapping the true probability The final expressions for hypothesis test statistics might involve the true probability $\mathbb{T}(P)$. It is not available, so we just plug in $\mathbb{F}(P)$ for $\mathbb{T}(P)$ in the end. If we had a prior on $\mathbb{T}(P)$, we could combine it with $\mathbb{F}(P)$ to get a more robust estimate.

Negation We take the violation metric and the corresponding threshold as to produce a hypothesis test against this:

$$\begin{aligned}\mathbb{F}(P)+\mathbb{F}(\neg P)-1=\mathbb{T}(P)+\varepsilon_1+\mathbb{T}(\neg P)+\varepsilon_2-1=\varepsilon_1+\varepsilon_2\\ \sim \mathsf{N}\left(0,\sigma^2(\mathbb{T}(P)(1-\mathbb{T}(P))+\mathbb{T}(\neg P)(1-\mathbb{T}(\neg P)))\right)\end{aligned}$$

We estimate the unknown $\mathbb{T}$ values with the corresponding $\mathbb{F}$ estimates. Note that, although $\mathbb{T}(P)=1-\mathbb{T}(\neg P)$, it is of course not necessarily the case that $\mathbb{F}(P)=1-\mathbb{F}(\neg P)$.

The error distribution is $\sigma\mathsf{N}\left(\mathbb{F}(P)(1-\mathbb{F}(P))+\mathbb{F}(\neg P)(1-\mathbb{F}(\neg P))\right)$, and the two-sided test is

$$|\mathbb{F}(P)+\mathbb{F}(\neg P)-1|<\gamma\sigma\sqrt{(1-\mathbb{F}(P))\mathbb{F}(P)+(1-\mathbb{F}(\neg P))\mathbb{F}(\neg P)}$$

for some scale factor γ (number of standard deviations) that scales the power of the test. For example, $\gamma=2.58$ gives a 99%-confidence interval.

We now want to compute some *consistency violation metric* that makes inconsistency comparable across different checks. The natural idea is to aggregate all terms dependent on $\mathbb{F}$ to one side; and make the hypothesis test be just some threshold on the computed violation metric.

It is possible that the denominator of the resulting expression is 0 when the forecaster is certain and $\mathbb{F}$ is 0 or 1; to avoid division with zero, we add a small regularization term $\beta_{\text{MIN}}=10^{-3}$. See the last paragraph of this section for a discussion of hyperparameters.

Our consistency violation metric is then:

$$v_{\text{NEGATION}}=\frac{|\mathbb{F}(P)+\mathbb{F}(\neg P)-1|}{\sqrt{(1-\mathbb{F}(P))\mathbb{F}(P)+(1-\mathbb{F}(\neg P))\mathbb{F}(\neg P)+\beta_{\text{MIN}}}}.$$

The hyperparameter σ^2 determines how strict we are with rejecting inconsistencies which could be attributed to "noisy" predictions. Note that the violation metric itself does not depend on σ^2.

A violation (inconsistency), therefore, occurs when:

$$v_{\text{NEGATION}}>\gamma\sigma.$$

CondCond This is a more complex consistency check; we derive the hypothesis test and violation metric in detail below. For the other checks, we just report the short derivation.

$$\begin{aligned}(a,b,c,d)&=(\mathbb{T}(P),\mathbb{T}(Q\mid P),\mathbb{T}(R\mid P\wedge Q),\mathbb{T}(P\wedge Q\wedge R))\\(a',b',c',d')&=(\mathbb{F}(P),\mathbb{F}(Q\mid P),\mathbb{F}(R\mid P\wedge Q),\mathbb{F}(P\wedge Q\wedge R))\end{aligned}$$

We can write:

$$\begin{aligned}\mathbb{F}(P)&=\mathsf{N}\left(0,\sigma^2a(1-a)\right)+a,\\\mathbb{F}(Q\mid P)&=\mathsf{N}\left(0,\sigma^2b(1-b)\right)+b,\\\mathbb{F}(R\mid P\wedge Q)&=\mathsf{N}\left(0,\sigma^2c(1-c)\right)+c,\\\mathbb{F}(P\wedge Q\wedge R)&=\mathsf{N}\left(0,\sigma^2d(1-d)\right)+d\end{aligned}$$

We now compute the difference of the two expressions that should be equal. All sums and products are cyclic over a, b, c.

$$\begin{aligned}\mathbb{F}(P)\mathbb{F}(Q \mid P)\mathbb{F}(R \mid P \wedge Q) - \mathbb{F}(P \wedge Q \wedge R) = abc - d \\ + \sigma\left(\sum_a bc\mathsf{N}(0, a(1-a)) - \mathsf{N}(0, d(1-d))\right) \\ + \sigma^2 \sum_a \mathsf{N}(0, b(1-b))\mathsf{N}(0, c(1-c)) \\ + \sigma^3 \prod_a \mathsf{N}(0, a(1-a)).\end{aligned}$$

In the above, all Gaussians with the same variance are identical, and all other combinations are independent. As $abc - d = 0$ by the law of total probability, the leading error term is next to σ. This is a Gaussian with mean 0 and standard deviation:

$$\sigma\sqrt{\sum_a b^2c^2a(1-a) + d(1-d)} = \sigma\sqrt{abc\sum_a bc(1-a) + d(1-d)}$$

We now discard the terms of σ^2, σ^3, and in general any higher order power of σ. This is principled because the coefficients can always be (in some confidence interval) upper bounded by a constant independent of σ. Hence, if σ is small enough, the resulting test will be very close to the true hypothesis test.

We do not have the true probabilities a, b, c, d, so we just plug in $(a', b', c', d') = (\mathbb{F}(P), \mathbb{F}(Q \mid P), \mathbb{F}(R \mid P \wedge Q), \mathbb{F}(P \wedge Q \wedge R))$. [28] Thus the hypothesis test is (where the sum is cyclic over a', b', c'):

$$|a'b'c' - d'| > \gamma\sigma\sqrt{a'b'c'\sum_{a'} b'c'(1-a') + d'(1-d')}$$

Our violation metric is then:

$$v_{\textsc{CondCond}} = \frac{|a'b'c' - d'|}{\sqrt{a'b'c'\sum_{a'} b'c'(1-a') + d'(1-d') + \beta_{\text{MIN}}}}.$$

where again $(a', b', c', d') = (\mathbb{F}(P), \mathbb{F}(Q \mid P), \mathbb{F}(R \mid P \wedge Q), \mathbb{F}(P \wedge Q \wedge R))$ are the forecasts.

Cond Similarly as for CondCond: we denote $(a, b, c) = (\mathbb{T}(P), \mathbb{T}(P \mid Q), \mathbb{T}(P \wedge Q))$ and the associated (a', b', c') for the forecasts. Then we can compute

$$\begin{aligned}&\mathbb{F}(P)\mathbb{F}(Q \mid P) - \mathbb{F}(P \wedge Q) \\ &= ab - c + \sigma\left(b\mathsf{N}(0, a(1-a)) + a\mathsf{N}(0, b(1-b)) - \mathsf{N}(0, c(1-c))\right) \\ &\quad + \sigma^2\mathsf{N}(0, a(1-a))\mathsf{N}(0, b(1-b)).\end{aligned}$$

The term next to σ is a Gaussian with mean 0 and standard deviation:

$$\sigma\sqrt{a^2b(1-b) + b^2a(1-a) + c(1-c)} = \sigma\sqrt{ab\left(a(1-b) + b(1-a)\right) + c(1-c)}.$$

Again, we have to plug in $(a', b', c') = (\mathbb{F}(P), \mathbb{F}(Q \mid P), \mathbb{F}(P \wedge Q))$ instead of (a, b, c).

[28] Depending on how we use the relation $abc = d$, we can end up with different expressions in the end. We choose the one that, after plugging in, (i) yields an expression for variance that is always nonnegative, and (ii) is not a polynomial multiple of any single value of $\mathbb{F}$.

Our violation metric is then:

$$v_{\text{COND}} = \frac{|a'b' - c'|}{\sqrt{a'b'\left(a'(1-b') + b'(1-a')\right) + c'(1-c') + \beta_{\text{MIN}}}}$$

And the test is again, for a suitable γ corresponding to the desired power of the test:

$$v_{\text{COND}} > \gamma\sigma.$$

Paraphrase Here we can simply check whether P and Q are the same.

$$\begin{aligned}&\mathbb{F}(P) - \mathbb{F}(Q) = \mathbb{T}(P) + \varepsilon_1 - \mathbb{T}(Q) - \varepsilon_2\\ &= \varepsilon_1 - \varepsilon_2 \sim \mathsf{N}\left(0, \sigma^2((\mathbb{T}(P)(1-\mathbb{T}(P)) + (\mathbb{T}(Q)(1-\mathbb{T}(Q)))\right)\end{aligned}$$

This yields the following violation metric:

$$v_{\text{PARAPHRASE}} = \frac{|\mathbb{F}(P) - \mathbb{F}(Q)|}{\sqrt{(\mathbb{F}(P)(1-\mathbb{F}(P)) + (\mathbb{F}(Q)(1-\mathbb{F}(Q)) + \beta_{\text{MIN}}}}$$

AndOr

$$\begin{aligned}&\mathbb{F}(P) + \mathbb{F}(Q) - \mathbb{F}(P \vee Q) - \mathbb{F}(P \wedge Q)\\ &= \mathbb{T}(P) + \mathbb{T}(Q) - \mathbb{T}(P \vee Q) - \mathbb{T}(P \wedge Q) + \varepsilon_1 + \varepsilon_2 - \varepsilon_3 - \varepsilon_4\\ &= \varepsilon_1 + \varepsilon_2 - \varepsilon_3 - \varepsilon_4\\ &\sim \mathsf{N}\left(0, \sigma^2\left(\mathbb{T}(P)(1-\mathbb{T}(P)) + \mathbb{T}(Q)(1-\mathbb{T}(Q))\right.\right.\\ &\left.\left.+\mathbb{T}(P \vee Q)(1-\mathbb{T}(P \vee Q)) + \mathbb{T}(P \wedge Q)(1-\mathbb{T}(P \wedge Q))\right)\right).\end{aligned}$$

We again plug in $\mathbb{F}$ instead of $\mathbb{T}$ to compute the error term allowed: $\gamma\sigma\sqrt{M}$ where

$$\begin{aligned}&M = \mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q) + \mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) +\\ &\mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q))\end{aligned}$$

and violation metric:

$$v_{\text{ANDOR}} = \frac{|\mathbb{F}(P) + \mathbb{F}(Q) - \mathbb{F}(P \vee Q) - \mathbb{F}(P \wedge Q)|}{\sqrt{\begin{array}{l}\mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q)) +\\ \mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) + \mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q)) + \beta_{\text{MIN}}\end{array}}}.$$

But

$$\begin{aligned}&\mathbb{F}(P \vee Q) - \mathbb{F}(P) - \mathbb{F}(\neg P \wedge Q) = \mathbb{T}(P \vee Q) - \mathbb{T}(P) - \mathbb{T}(\neg P \wedge Q) + \varepsilon_1 - \varepsilon_2 - \varepsilon_3 =\\ &\varepsilon_1 - \varepsilon_2 - \varepsilon_3 \sim\\ &\mathsf{N}\left(0, \sigma^2((\mathbb{T}(P \vee Q)(1-\mathbb{T}(P \vee Q)) + (\mathbb{T}(P)(1-\mathbb{T}(P)) + (\mathbb{T}(\neg P \wedge Q)(1-\mathbb{T}(\neg P \wedge Q))\right)\end{aligned}$$

with error term:

$$\gamma\sigma\sqrt{\mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q) + \mathbb{F}(P)(1-\mathbb{F}(P) + \mathbb{F}(\neg P \wedge Q)(1-\mathbb{F}(\neg P \wedge Q)}$$

and violation metric:

$$v_{\text{BUT}} = \frac{|\mathbb{F}(P \vee Q) - \mathbb{F}(P) - \mathbb{F}(\neg P \wedge Q)|}{\sqrt{\mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) + \mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(\neg P \wedge Q)(1-\mathbb{F}(\neg P \wedge Q) + \beta_{\text{MIN}}}}$$

Consequence In the case of inequalities involving $\leq$, there are two ways in which the consistency check can be passed. If $\mathbb{F}(P) \leq \mathbb{F}(Q)$, the consistency check is automatically passed. Otherwise, we check for pseudo-equality using the same violation metric as in PARAPHRASE.

$$v_{\text{CONSEQUENCE}} = [\mathbb{F}(P) > \mathbb{F}(Q)] \frac{|\mathbb{F}(P) - \mathbb{F}(Q)|}{\sqrt{\mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q)) + \beta_{\text{MIN}}}}$$

where $[\mathbb{F}(P) > \mathbb{F}(Q)]$ is the Iverson Bracket (1 if true, 0 otherwise).

And Similarly to CONSEQUENCE, if the chain of strict inequalities

$$\max(\mathbb{F}(P) + \mathbb{F}(Q) - 1, 0) < \mathbb{F}(P \wedge Q) < \min(\mathbb{F}(P), \mathbb{F}(Q))$$

holds, then the check automatically passes. We set $v_{\text{AND\_LHS}} = 0$ and $v_{\text{AND\_RHS}} = 0$ if it passes the first and second strict inequality respectively.

If not, then we test for pseudo-equality for the violating pair:

$\text{LHS} : \max(\mathbb{F}(P) + \mathbb{F}(Q) - 1, 0) = \mathbb{F}(P \wedge Q)$

$\text{RHS} : \mathbb{F}(P \wedge Q) = \min(\mathbb{F}(P), \mathbb{F}(Q))$

Equality check if it fails the first inequality:

$$\varepsilon_{\text{LHS}} = \begin{cases} \gamma\sigma\sqrt{\mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q)) + \mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q))} \\ \quad \text{if } \mathbb{F}(P) + \mathbb{F}(Q) - 1 > 0, \\ \text{N/A} \\ \quad \text{otherwise pass as } \mathbb{F}(P \wedge Q) \geq 0. \end{cases}$$

$$v_{\text{AND\_LHS}} = [\mathbb{F}(P) + \mathbb{F}(Q) - 1 > \mathbb{F}(P \wedge Q)] \cdot \frac{\mathbb{F}(P) + \mathbb{F}(Q) - 1 - \mathbb{F}(P \wedge Q)}{\sqrt{\mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q)) + \mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q)) + \beta_{\text{MIN}}}}$$

Equality check if it fails the second inequality:

Define $\mathbb{F}(R) = \min(\mathbb{F}(P), \mathbb{F}(Q))$.

$$\varepsilon_{\text{RHS}} = \gamma\sigma\sqrt{\mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q)) + \mathbb{F}(R)(1+\mathbb{F}(R))}$$

$$v_{\text{AND\_RHS}} = [\mathbb{F}(R) < \mathbb{F}(P \wedge Q)] \frac{\mathbb{F}(P \wedge Q) - \mathbb{F}(R)}{\sqrt{\mathbb{F}(P \wedge Q)(1-\mathbb{F}(P \wedge Q)) + \mathbb{F}(R)(1-\mathbb{F}(R)) + \beta_{\text{MIN}}}}$$

Consistency is violated if either inequality is violated, *and* the respective hypothesis test for pseudo-equality fails. We use $v_{\text{AND\_LHS}}$ for the first and $v_{\text{AND\_RHS}}$ for the second inequality. We define $v_{\text{AND}} = \max\{v_{\text{AND\_LHS}}, v_{\text{AND\_RHS}}\}$.

Or We proceed similarly as for AND.

If the strict inequality $\max(\mathbb{F}(P), \mathbb{F}(Q)) < \mathbb{F}(P \vee Q) < \min(1, \mathbb{F}(P) + \mathbb{F}(Q))$ holds, then it automatically passes. We set $v_{\text{OR\_LHS}} = 0$ and $v_{\text{OR\_RHS}} = 0$ if it passes the first and second strict inequality respectively.

If not, we test for pseudo-equality:

$\text{LHS} : \max(\mathbb{F}(P), \mathbb{F}(Q)) = \mathbb{F}(P \vee Q)$

RHS : $\mathbb{F}(P \vee Q) = \min(1, \mathbb{F}(P) + \mathbb{F}(Q))$.

Equality check LHS: Define $\mathbb{F}(S) = \max(\mathbb{F}(P), \mathbb{F}(Q))$.

$$\varepsilon_{\text{LHS}} = \gamma\sigma\sqrt{\mathbb{F}(S)(1-\mathbb{F}(S)) + \mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q))}$$

$$v_{\text{Or\_LHS}} = [\mathbb{F}(S) > \mathbb{F}(P \vee Q)]\frac{\mathbb{F}(S) - \mathbb{F}(P \vee Q)}{\sqrt{\mathbb{F}(S)(1-\mathbb{F}(S)) + \mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) + \beta_{\text{MIN}}}}$$

Equality check RHS:

$$\varepsilon_{\text{RHS}} = \begin{cases} \gamma\sigma\sqrt{\mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) + \mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q))} \\ \quad \text{if } \mathbb{F}(P) + \mathbb{F}(Q) < 1, \\ \text{N/A} \\ \quad \text{otherwise pass as } \mathbb{F}(P \vee Q) \leq 1. \end{cases}$$

$$v_{\text{Or\_RHS}} = [\mathbb{F}(P) + \mathbb{F}(Q) < \mathbb{F}(P \vee Q)] \cdot \frac{\mathbb{F}(P \vee Q) - \mathbb{F}(P) - \mathbb{F}(Q)}{\sqrt{\mathbb{F}(P \vee Q)(1-\mathbb{F}(P \vee Q)) + \mathbb{F}(P)(1-\mathbb{F}(P)) + \mathbb{F}(Q)(1-\mathbb{F}(Q)) + \beta_{\text{MIN}}}}$$

Consistency is violated if either inequality is violated, *and* the subsequent hypothesis test for pseudo-equality fails. We use $v_{\text{Or\_LHS}}$ for the first and $v_{\text{Or\_RHS}}$ for the second inequality. Analogously to And, define $v_{\text{Or}} = \max\{v_{\text{Or\_LHS}}, v_{\text{Or\_RHS}}\}$.

ExpEvidence Write $(a, b, c, d) = (\mathbb{T}(P), \mathbb{T}(P \mid Q), \mathbb{T}(P \mid \neg Q), \mathbb{T}(Q))$; then

$$\begin{aligned} & b'd' + c'(1-d') - a' \\ &= (b + \sigma N(b(1-b)))(d + \sigma N(d(1-d))) \\ &\quad + (c + \sigma N(c(1-c)))(1 - d - \sigma N(d(1-d))) \\ &\quad - (a + \sigma N(a(1-a))) \\ &= (bd + c(1-d) - a) \\ &\quad + \sigma\left[dN(b(1-b))\right. \\ &\qquad + (b-c)N(d(1-d)) \\ &\qquad + (1-d)N(c(1-c)) \\ &\qquad \left. -N(a(1-a))\right] \\ &\quad + O(\sigma^2) \end{aligned}$$

gives us a normal distribution with standard deviation

$$\sigma\sqrt{a(1-a) + d^2b(1-b) + (1-d)^2c(1-c) + (b-c)^2d(1-d).}$$

The violation metric is then:

$$\frac{|bd + c(1-d) - a|}{\sigma\sqrt{a(1-a) + d^2b(1-b) + (1-d)^2c(1-c) + (b-c)^2d(1-d).}}$$

Algorithm 3 `ArbitrageForecaster` algorithm: $\langle \mathbb{F} \rangle_{\vec{C}}$

input x
$p \leftarrow \mathbb{F}(x)$ ▷ Query base forecaster
$w \leftarrow 1$
for $(\mathcal{R}_i, \mathcal{S}_i, \mathcal{J}_i)$ in $\vec{C}$ **do**
  $(x, x_2, \dots x_n) \leftarrow \mathcal{J}_i(x)$ ▷ Instantiate tuple of size $n = n_{\mathcal{R}_i}$
  $(p_2, \dots p_n) \leftarrow (\mathbb{F}(x_2), \dots \mathbb{F}(x_n))$ ▷ Query base forecaster on tuple
  $(p, p_2, \dots p_n) \leftarrow \mathcal{A}_i^{(w,1,\dots 1)}(p, p_2, \dots p_n)$ ▷ arbitrage the forecasts as per Def 4
  $w \leftarrow w + n - 1$ ▷ p now carries information from $n-1$ other markets
end for
return p

Hyperparameters for hypothesis testing Our goal is for the rejection criteria to be similar to the arbitrage violation metric in Section 6.8.2 on simple examples. We choose $\gamma = 2.58$ for all checks, to ensure 99%-confidence intervals for two-sided tests; future work may consider using a different γ for checks that require one-sided tests. We pick $\sigma = 0.05$ (corresponding to $n = 400$ in Definition 6.2). The allowed violation threshold for all checks is then $\gamma\sigma = 0.129$. For reference, a NEGATION pair $(\mathbb{F}(P), \mathbb{F}(\neg P)) = (0.5, 0.59)$ has a violation metric of 0.128, and would thus not be rejected as inconsistent. This closely corresponds to the tolerance threshold of 10^{-2} of profit for the arbitrage metric, described in Section 6.2.1.

We pick $\beta_{\text{MIN}} = 10^{-3}$ because LLM forecasters from Halawi et al. (2024a) answer with at most 3 digits of precision for events close to 0 and 1 in probability.

6.8.4 ARBITRAGEFORECASTER

To formally define `ArbitrageForecaster`, we need to first formalize our "instantiation" process mathematically:

Definition 6.3 (Tuple sampler)**.** Let $\mathcal{R} : \text{Prop}^n \to \{\top, \bot\}$, $\mathcal{S} : \text{Forecast}^n \to \{\top, \bot\}$ be a consistency check. Then we call $\mathcal{J} : \text{Prop} \rightsquigarrow \text{Prop}^n$ a "single-base-question tuple sampler" for $\mathcal{R}$ if for all x, $\mathcal{J}(x)_1 = x$ and $\mathcal{R}(\mathcal{J}(x))$ holds surely.

A multiple-base-question tuple sampler $\mathcal{I} : \text{Prop}^m \to \text{Prop}^n$, like our instantiation process, can simply be composed with a question sampler $\mathcal{G} : \text{Prop} \rightsquigarrow \text{Prop}$ (e.g a synthetic generator or a sampler from our dataset) to produce a single-base-question sampler $\mathcal{J}(x) := \mathcal{I}(x, \mathcal{G}(x), \dots \mathcal{G}(x))$.

Next, in order to correctly handle sequentially arbitraging checks and prevent bias towards later applied checks, we need to introduce "weighted" arbitraging. This follows easily from Eq 6.1 by simply having the scoring rule for each question x be $w_x \log(p)$. We denote the calculation of arbitraged probabilities under these weighted scoring rules by $\mathcal{A}^{(w_1, \dots w_n)}$.

Definition 6.4 (`ArbitrageForecaster`)**.** Let $\mathbb{F} : \text{Prop} \to \text{Forecast}$ be the "Base Forecaster", and let $\vec{C} := [(\mathcal{R}_1, \mathcal{S}_1, \mathcal{J}_1), \dots (\mathcal{R}_k, \mathcal{S}_k, \mathcal{J}_k)]$ be a list of consistency checks along with respective single-base-question tuple samplers. Then we construct a new forecaster $\langle \mathbb{F} \rangle_{\vec{C}} : \text{Prop} \to \text{Forecast}$ that produces its forecast for a given question x as given in Algorithm 3; we call this the `ArbitrageForecaster` with base $\mathbb{F}$ and check list $\vec{C}$.

The first thing we observe is that this isn't necessarily *robust* to different instantiations. For this reason, we a priori expect that **`ArbitrageForecaster` will be more effective on**

We might hope that the `ArbitrageForecaster` introduced in Def 6.4 would be definitionally consistent on the checks it is arbitraged on. However, this is not the case *even for* `ArbitrageForecaster` *applied to a single check* $\mathcal{R}(x_1, \dots x_n)$, because the tuple of forecasts that is arbitraged to compute $\langle \mathbb{F} \rangle_{(\mathcal{R},\mathcal{S},\mathcal{J})}(x_1)$, the tuple arbitraged to compute $\langle \mathbb{F} \rangle_{(\mathcal{R},\mathcal{S},\mathcal{J})}(x_2)$, …, the tuple arbitraged to compute $\langle \mathbb{F} \rangle_{(\mathcal{R},\mathcal{S},\mathcal{J})}(x_n)$ are all different. While the tuple instantiated to compute $\langle \mathbb{F} \rangle_{(\mathcal{R},\mathcal{S},\mathcal{J})}(x_1)$ could indeed be $\mathcal{J}(x_1) = (x_1, \dots x_n)$ (at

least if the tuple sampler $\mathcal{J}$ is deterministic and happens to be the same as the one used in the instantiation of the check), the tuples instantiated to compute $\langle\mathbb{F}\rangle_{(\mathcal{R},\mathcal{S},\mathcal{J})}(x_i)$ for $i \neq 1$ will be $\mathcal{J}(x_i)$, all of which are different from one another.

To make this concrete, consider the simplest case of $\langle\mathbb{F}\rangle_P$ (where P is short for PARAPHRASE); let para be a deterministic tuple-sampler for PARAPHRASE. $\langle\mathbb{F}\rangle_P(x)$ is calculated by arbitraging $\mathbb{F}(x)$ and $\mathbb{F}(\text{para}(x))$. But $\mathbb{F}(\text{para}(x))$ is calculated by arbitraging $\mathbb{F}(\text{para}(x))$ and $\mathbb{F}(\text{para}(\text{para}(x)))$.

A priori, this gives us the following hypothesis: **`ArbitrageForecaster` will be especially effective for fundamentally "symmetric" checks like Negation** – where $\text{neg}(\text{neg}(P))$ is likely to be a very similar sentence to P. Although we have not conducted a full scale experiment of `ArbitrageForecaster` with each checker, our preliminary results in Table 5 do suggest very good performance of `ArbitrageForecaster` on NEGATION.

Suppose, however, that we had an "extended" `ArbitrageForecaster` that made its forecast for x based on the tuple $(x, \text{para}(x), \text{para}^2(x), \dots \text{para}^r(x))$ – then its forecast for $\text{para}(x)$ would be based on $(\text{para}(x), \text{para}^2(x), \dots \text{para}^{r+1}(x)$ – these tuples would be "almost" the same, except with $\text{para}^{r+1}(x)$ instead of x, and this extended `ArbitrageForecaster` would be "almost" consistent on PARAPHRASE.

This is precisely the idea behind recursively applying `ArbitrageForecaster` to itself: we recursively define $\langle\mathbb{F}\rangle^r(x) := \mathcal{A}(\langle\mathbb{F}\rangle^{r-1}(\mathcal{J}(x)_i) \text{ for } i = 1, \dots n)$ – then *if* this iteration approaches a fixed point, this fixed point $\langle\mathbb{F}\rangle^\infty$ is consistent. More precisely:

Theorem 6.5 (Consistency of recursive `ArbitrageForecaster`)**.** *Let $(\mathcal{R}, \mathcal{S}, \mathcal{J})$ be an n-ary consistency check and a corresponding deterministic tuple sampler satisfying Def 6.3, and have $\mathcal{A}(p_1, \dots p_n)$ and $\mathcal{V}(p_1, \dots p_n)$ denote the arbitraging function and arbitrage metric corresponding to $\mathcal{R}$ as per Def 6.1 under a logarithmic scoring rule. Then, for some "base forecaster" $\langle\mathbb{F}\rangle^0 = \mathbb{F}$, recursively define*

$$\langle\mathbb{F}\rangle^r(x) := \mathcal{A}(\langle\mathbb{F}\rangle^{r-1}(\mathcal{J}(x)_i) \text{ for } i = 1, \dots n)$$

If this iteration converges pointwise in log-odds space – i.e. if for all $x \in \text{Prop}$, the sequence $\langle\mathbb{F}\rangle^r(x)$ has a limit strictly between 0 and 1, then $\mathcal{V}(\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i) \text{ for } i = 1, \dots n) \to 0$.

Proof. Recall as per Def 6.1 that, where Ω is the set of possible outcomes allowed by $\mathcal{R}$:

$$\begin{aligned}
&\mathcal{V}(\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i) \text{ for } i = 1, \dots n) \\
&= \min_{\omega\in\Omega} \sum_{i=1}^{n} \left(\log(\mathcal{A}(\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_j) \text{ for } j = 1, \dots n)_i) - \log\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i)\right) \delta_{\omega(i)=\top} \\
&\quad + \left(\log(1 - \mathcal{A}(\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_j) \text{ for } j = 1, \dots n)_i) - \log(1 - \langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i))\right) \delta_{\omega(i)=\bot} \\
&= \min_{\omega\in\Omega} \sum_{i=1}^{n} \left(\log\langle\mathbb{F}\rangle^{r+1}(\mathcal{J}(x)_i)) - \log\langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i)\right) \delta_{\omega(i)=\top} \\
&\quad + \left(\log(1 - \langle\mathbb{F}\rangle^{r+1}(\mathcal{J}(x)_i) - \log(1 - \langle\mathbb{F}\rangle^r(\mathcal{J}(x)_i))\right) \delta_{\omega(i)=\bot}
\end{aligned}$$

Since $\langle\mathbb{F}\rangle^r(x)$ converges to something that is neither 0 nor 1, so do $\log\langle\mathbb{F}\rangle^r(x)$ and $\log(1 - \langle\mathbb{F}\rangle^r(x))$. And as this is true for *all* x, so in particular it is true for $\mathcal{J}(x)_i$. Thus the expression above is a finite sum of terms that each approach 0. □

This is a somewhat weak result: other than for NEGATION and PARAPHRASE, none of our static consistency checks involved a deterministic instantiation process – they all require sampling other related base questions, and having the checks use the same instantiation process as the `ArbitrageForecaster` would be cheating.

Furthermore, this gives us no actual conditions for the convergence of the iteration. At least for PARAPHRASE, we have the following – where $\log\text{odds}\, p$ denotes $\log\frac{p}{1-p}$:

Theorem 6.6 (Convergence of recursive `ArbitrageForecaster` for PARAPHRASE)**.** *If the sequence* $a_i = \log\text{odds}\,\mathbb{F}(\text{para}^i(x))$ *is convergent, then the condition of Theorem 6.5 holds for the recursive* `ArbitrageForecaster` *defined arbitraged on* PARAPHRASE *with tuple sampler* para.

Proof. Recall from Sec 6.8.2 that the arbitraged probability for PARAPHRASE is simply the average of the original probabilities in log-odds space, i.e. $\log\text{odds}\,\mathcal{A}(\mathbb{F}(x), \mathbb{F}(\text{para}(x))) = \frac{\log\text{odds}\,\mathbb{F}(x)+\log\text{odds}\,\mathbb{F}(\text{para}(x))}{2}$. We can apply this recursively to get:

$$\langle\mathbb{F}\rangle^r(x) = \frac{1}{2^r}\sum_{i=0}^{r}\binom{r}{i}\log\text{odds}\,\mathbb{F}(\text{para}^i(x))$$

Which is simply a binomial moving average of $\log\text{odds}\,\mathbb{F}(\text{para}^i(x)) = a_i$, and converges iff a_i does. Convergence in log-odds space is equivalent to convergence of probability to something other than 0 or 1, so the result follows. □

Choices of experiments A single call to $\langle\mathbb{F}\rangle_{\vec{C}}$, where $\vec{C} := [(\mathcal{R}_1, \mathcal{S}_1, \mathcal{J}_1), ...(\mathcal{R}_k, \mathcal{S}_k, \mathcal{J}_k)]$, involves $1 + \sum_i (n_{\mathcal{R}_i} - 1)$ calls to $\mathbb{F}$, plus at least $\sum_i (m_{\mathcal{R}_i} + n_{\mathcal{R}_i} - 2)$ (where $m_{\mathcal{R}_i}$ is the number of separate base questions that must be generated synthetically in each tuple) LLM calls for the $\mathcal{J}_i$s.

For all the checks listed in Table 1, this amounts to a total of 49 LLM calls per question. For a *recursive* `ArbitrageForecaster` set-up of depth r, this amounts to 49^r LLM calls per question, which can get prohibitively expensive. Even on `gpt-4o-mini` and assuming ≈ 600 input tokens and 600 output tokens on average, this amounts to $\approx$ \$0.02 per question at depth $r = 1$, and $\approx$ \$2500 per question at depth $r = 4$.

Furthermore, it was not clear that experimenting on all checks made logical sense: recursive `ArbitrageForecaster` set-ups with COND, CONDCOND and EXPEVIDENCE would involve forms like $P \mid (Q \mid R)$, which do not have a basis in probability theory. We decided to prioritize studying the following hypotheses and research questions, motivated by the theoretical discussion above:

1. We hypothesised above that `ArbitrageForecaster` will be **particularly effective on checks that are symmetric and have deterministic instantiations** – thus we studied $\langle$`gpt-4o-mini`$\rangle_{\text{NEGATION}}$.
2. We hypothesized that there would be **consistency gains from increasing depth** r – thus we studied recursive `ArbitrageForecaster` setups on NEGATION an PARAPHRASE, where it was most practical to.
3. We were interested to know **if the consistency gains observed when arbitraging on one check alone would persist after arbitraging on a sequence of checks** – to predict if this would hold when arbitraging on the full sequence of checks, we did a preliminary run of $\langle$`gpt-4o-mini`$\rangle_{\text{NEGATION,PARAPHRASE}}$ and tested if it maintains consistency on NEGATION and PARAPHRASE.
4. **We expected $\langle\mathbb{F}\rangle_{\textbf{ExpEvidence}}$ to improve ground truth and consistency scores across the board.** This is based on our intuition that arbitraging on EXPEVIDENCE essentially "informs" the forecast on a question x with consideration information y – except instead of subjectively feeding this information (e.g. in chain-of-thought), it adjusts for it via a strict probabilistic rule. Although a recursive setup would not make sense for EXPEVIDENCE, $\langle\mathbb{F}\rangle_{[\text{EXPEVIDENCE}]*r}$ simply sequentially arbitrages on EXPEVIDENCE repeatedly (breaking the seed each time to ensure unique new questions y), which amounts to informing the forecast for x with information y_1, y_2 etc.

The results reported in Sec 6.5 of the main body and 6.8.4 of the Appendix provide evidence in favour of hypotheses 1 and 2, answer 3 in the affirmative, and do not provide clear evidence on 4.

Future work should compare $\langle \mathbb{F} \rangle_{[\text{ExpEvidence}]}$ against a comparable chain-of-thought model in which the forecaster is asked to consider these related questions before it makes its forecast.

Results tables for `ArbitrageForecaster`

Consistency violation and ground truth results for each of the `ArbitrageForecaster` configurations we experimented with are reported in Tables 5, 6, 7 and 8. The results included are for the NewsAPI dataset and the arbitrage metric. Results for the scraped and 2028 synthetic datasets (**??**), as well as for the frequentist metric, look very similar; they are available in the supplementary data of this paper.

Table 5: Consistency results (arbitrage metric) for $\langle$`gpt-4o-mini`$\rangle^r_{\text{Negation}}$ (denoted CF-Nr) forecasters on NewsAPI questions.

Check	`gpt-4o-mini`		CF-N1		CF-N2		CF-N3		CF-N4	
	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac
Negation	0.036	43%	0.012	33%	0.007	22%	0.004	11%	0.004	9%
Paraphrase	0.013	27%	0.012	36%	0.008	23%	0.006	16%	0.005	17%
CondCond	0.084	85%	0.111	88%	0.121	91%	0.129	94%	0.136	93%
ExpEvidence	0.015	27%	0.009	35%	0.008	25%	0.007	26%	0.007	25%
Consequence	0.005	10%	0.003	9%	0.003	7%	0.002	4%	0.001	3%
And	0.006	20%	0.019	45%	0.027	53%	0.031	59%	0.035	65%
Or	0.007	13%	0.004	10%	0.002	6%	0.002	6%	0.001	4%
AndOr	0.017	38%	0.024	58%	0.031	61%	0.033	67%	0.035	66%
But	0.053	75%	0.081	84%	0.091	89%	0.100	88%	0.107	91%
Cond	0.062	88%	0.085	92%	0.107	91%	0.119	94%	0.131	96%
aggregated	0.030		0.036		0.041		0.043		0.046	
Brier score	0.185		0.204		0.202		0.201		0.201	

Table 6: Consistency results (arbitrage metric) for $\langle$`gpt-4o-mini`$\rangle^r_{\text{Paraphrase}}$ (denoted CF-Pr) forecasters on NewsAPI questions.

Check	`gpt-4o-mini`		CF-P1		CF-P2		CF-P3		CF-P4	
	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac
Negation	0.036	43%	0.028	49%	0.026	50%	0.023	46%	0.024	44%
Paraphrase	0.013	27%	0.006	22%	0.004	11%	0.002	6%	0.002	3%
CondCond	0.084	85%	0.083	83%	0.079	85%	0.080	83%	0.079	84%
ExpEvidence	0.015	27%	0.014	28%	0.012	24%	0.011	28%	0.012	28%
Consequence	0.005	10%	0.002	4%	0.001	3%	0.001	2%	0.001	2%
And	0.006	20%	0.004	12%	0.005	13%	0.004	12%	0.004	12%
Or	0.007	13%	0.005	10%	0.004	9%	0.003	10%	0.003	9%
AndOr	0.017	38%	0.015	41%	0.014	42%	0.013	39%	0.013	39%
But	0.053	75%	0.053	76%	0.049	77%	0.051	79%	0.048	79%
Cond	0.062	88%	0.066	93%	0.071	95%	0.069	95%	0.071	95%
aggregated	0.030		0.028		0.026		0.026		0.026	
Brier score	0.185		0.176		0.175		0.174		0.175	

References

Shipra Agrawal, Erick Delage, Mark Peters, Zizhuo Wang, and Yinyu Ye. A Unified Framework for Dynamic Prediction Market Design. *Operations Research*, 59(3):550–568, 2011. ISSN 0030-364X.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Table 7: Consistency results (arbitrage metric) for $\langle$`gpt-4o-mini`$\rangle^{r}_{[\text{Negation,Paraphrase}]}$ (denoted CF-NPr) forecasters on NewsAPI questions.

Check	`gpt-4o-mini`		CF-NP1		CF-NP2		CF-NP3		CF-NP4	
	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac
Negation	0.036	43%	0.014	30%	0.007	18%	0.004	9%	0.003	6%
Paraphrase	0.013	27%	0.006	17%	0.003	7%	0.002	2%	0.001	2%
CondCond	0.084	85%	0.095	90%	0.096	86%	0.108	94%	0.115	94%
ExpEvidence	0.015	27%	0.010	27%	0.007	27%	0.006	22%	0.005	21%
Consequence	0.005	10%	0.003	7%	0.001	3%	0.001	2%	0.001	0%
And	0.006	20%	0.011	30%	0.010	28%	0.011	34%	0.012	39%
Or	0.007	13%	0.004	11%	0.002	5%	0.001	4%	0.001	2%
AndOr	0.017	38%	0.017	43%	0.016	46%	0.016	46%	0.016	47%
But	0.053	75%	0.070	85%	0.072	91%	0.077	91%	0.083	97%
Cond	0.062	88%	0.082	96%	0.076	97%	0.076	97%	0.077	98%
aggregated	0.030		0.031		0.029		0.030		0.031	
Brier score	0.185		0.188		0.195		0.200		0.202	

Table 8: Consistency results (arbitrage metric) for $\langle$`gpt-4o-mini`$\rangle_{[\text{ExpEvidence}]*r}$ (denoted CF-rxEE1) forecasters on NewsAPI questions.

Check	`gpt-4o-mini`		CF-1xEE1		CF-2xEE1		CF-3xEE1		CF-4xEE1	
	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac	Avg	Frac
Negation	0.036	43%	0.030	51%	0.026	49%	0.024	50%	0.025	53%
Paraphrase	0.013	27%	0.008	22%	0.006	22%	0.005	19%	0.005	18%
CondCond	0.084	85%	0.057	82%	0.053	79%	0.050	76%	0.044	74%
ExpEvidence	0.015	27%	0.008	22%	0.007	19%	0.007	16%	0.007	20%
Consequence	0.005	10%	0.003	8%	0.002	7%	0.002	5%	0.002	6%
And	0.006	20%	0.002	6%	0.002	6%	0.002	4%	0.001	5%
Or	0.007	13%	0.004	9%	0.003	8%	0.002	8%	0.003	9%
AndOr	0.017	38%	0.014	42%	0.011	39%	0.010	34%	0.011	35%
But	0.053	75%	0.040	71%	0.039	74%	0.040	77%	0.035	68%
Cond	0.062	88%	0.049	88%	0.046	89%	0.044	88%	0.040	87%
aggregated	0.030		0.021		0.020		0.019		0.017	
Brier score	0.185		0.172		0.171		0.171		0.173	

Kenneth J. Arrow and Gerard Debreu. Existence of an Equilibrium for a Competitive Economy. *Econometrica*, 22(3):265–290, 1954. ISSN 0012-9682. doi: 10.2307/1907353.

Kenneth J. Arrow, Robert Forsythe, Michael Gorham, Robert Hahn, Robin Hanson, John O. Ledyard, Saul Levmore, Robert Litan, Paul Milgrom, Forrest D. Nelson, George R. Neumann, Marco Ottaviani, Thomas C. Schelling, Robert J. Shiller, Vernon L. Smith, Erik Snowberg, Cass R. Sunstein, Paul C. Tetlock, Philip E. Tetlock, Hal R. Varian, Justin Wolfers, and Eric Zitzewitz. The Promise of Prediction Markets. *Science*, 320(5878): 877–878, May 2008. doi: 10.1126/science.1157679.

Aurélien Baillon. Bayesian markets to elicit private information. *Proceedings of the National Academy of Sciences*, 114(30):7958–7962, July 2017. doi: 10.1073/pnas.1703486114.

Aurélien Baillon and Yan Xu. Simple bets to elicit private signals. *Theoretical Economics*, 16(3):777–797, 2021. ISSN 1555-7561. doi: 10.3982/TE4343.

Yoram Barzel. Transaction Costs: Are They Just Costs? *Zeitschrift für die gesamte Staatswissenschaft / Journal of Institutional and Theoretical Economics*, 141(1):4–16, 1985. ISSN 00442550.

Eric B. Baum. Toward a Model of Intelligence as an Economy of Agents. *Machine Learning*, 35(2):155–185, May 1999. ISSN 1573-0565. doi: 10.1023/A:1007593124513.

Henry Berg and Todd A Proebsting. Hanson's automated market maker. *The Journal of Prediction Markets*, 3(1):45–59, 2009.

Alina Beygelzimer, John Langford, and David M. Pennock. Learning performance of prediction markets with Kelly bettors. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems - Volume 3*, volume 3 of *AAMAS '12*, pages 1317–1318, Richland, SC, June 2012. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 978-0-9817381-3-0.

Denis Bonnay. Preuves et jeux sémantiques. *Philosophia Scientiae*, 8(2):105–123, 2004. ISSN 1775-4283.

Julien Boyer and Gabriel Sandu. Between proof and truth. *Synthese*, 187(3):821–832, August 2012. ISSN 1573-0964. doi: 10.1007/s11229-011-9903-y.

Jonah Brown-Cohen, Geoffrey Irving, and Georgios Piliouras. Scalable AI Safety via Doubly-Efficient Debate, November 2023.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering Latent Knowledge in Language Models Without Supervision. In *The Eleventh International Conference on Learning Representations*, September 2022.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeffrey Wu. Weak-to-Strong Generalization: Eliciting Strong Capabilities With Weak Supervision. In *Proceedings of the 41st International Conference on Machine Learning*, pages 4971–5012. PMLR, July 2024.

Mithun Chakraborty, Sanmay Das, and Justin Peabody. Price Evolution in a Continuous Double Auction Prediction Market With a Scoring-Rule Based Market Maker. *Proceedings of the AAAI Conference on Artificial Intelligence*, 29(1), February 2015. ISSN 2374-3468. doi: 10.1609/aaai.v29i1.9313.

Michael Chang, Sid Kaushik, S. Matthew Weinberg, Tom Griffiths, and Sergey Levine. Decentralized Reinforcement Learning: Global Decision-Making via Local Economic Transactions. In *Proceedings of the 37th International Conference on Machine Learning*, pages 1437–1447. PMLR, November 2020.

Tsong Y Chen, Shing C Cheung, and Shiu Ming Yiu. Metamorphic testing: a new approach for generating next test cases. Technical report, The Hong Kong University of Science and Technology, 1998.

Xi Chen and Shang-Hua Teng. Spending Is Not Easier Than Trading: On the Computational Equivalence of Fisher and Arrow-Debreu Equilibria. In Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra, editors, *Algorithms and Computation*, pages 647–656, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg. ISBN 978-3-642-10631-6.

Xi Chen, Decheng Dai, Ye Du, and Shang-Hua Teng. Settling the Complexity of Arrow-Debreu Equilibria in Markets with Additively Separable Utilities. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 273–282, October 2009. doi: 10.1109/FOCS.2009.29.

Maria Christakis, Hasan Ferit Eniser, Jörg Hoffmann, Adish Singla, and Valentin Wüstholz. Specifying and testing k-safety properties for machine-learning models. *arXiv preprint arXiv:2206.06054*, 2022.

Paul Christiano. Extracting information. https://ai-alignment.com/extracting-information-97cd956f2c17, October 2016.

Paul Christiano, Mark Xu, and Ajeya Cotra. Eliciting Latent Knowledge, December 2021.

Vincent Conitzer. Prediction Markets, Mechanism Design, and Cooperative Game Theory. In *Conference on Uncertainty in Artificial Intelligence*, June 2009.

Vincent Conitzer. Prediction Markets, Mechanism Design, and Cooperative Game Theory, May 2012.

Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H. Holliday, Bob M. Jacobs, Nathan Lambert, Milan Mosse, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, Emanuel Tewolde, and William S. Zwicker. Position: Social Choice Should Guide AI Alignment in Dealing with Diverse Human Feedback. In *Proceedings of the 41st International Conference on Machine Learning*, pages 9346–9360. PMLR, July 2024.

CWoo. Quantifier algebra. https://planetmath.org/quantifieralgebra, March 2013.

Lawrence Davis. Mapping classifier systems into neural networks. In *Proceedings of the 1st International Conference on Neural Information Processing Systems*, NIPS'88, pages 49–56, Cambridge, MA, USA, January 1988. MIT Press.

Abram Demski. Complete Feedback. https://www.alignmentforum.org/posts/3ag99iJEgFFwyj64Z/complete-feedback, November 2024.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031, 2021.

Evan Hubinger. AI safety via market making — LessWrong, June 2020.

Peter C. Fishburn. *Utility Theory for Decision Making*. Wiley, January 1970. ISBN 978-0-471-26060-8.

Lukas Fluri, Daniel Paleka, and Florian Tramèr. Evaluating superhuman models with consistency checks. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, volume 31, page 194–232. IEEE, April 2023. doi: 10.1109/satml59370.2024.00017. URL http://dx.doi.org/10.1109/SaTML59370.2024.00017.

FutureSearch. FUTURESEARCH: Manifold markets trading bot, 2024. URL https://manifold.markets/FUTURESEARCH. Accessed on 26-Sept-2024.

Scott Garrabrant, Tsvi Benson-Tilsen, Andrew Critch, Nate Soares, and Jessica Taylor. Logical Induction, December 2020.

Luise Ge, Daniel Halpern, Evi Micha, Ariel D. Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. Axioms for AI Alignment from Human Feedback, May 2024a.

Luise Ge, Daniel Halpern, Evi Micha, Ariel D. Procaccia, Itai Shapira, Yevgeniy Vorobeychik, and Junlin Wu. Axioms for AI Alignment from Human Feedback. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024b.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459. doi: 10.1198/016214506000001437.

Dhananjay K. Gode and Shyam Sunder. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy*, 101(1):119–137, February 1993. doi: 10.1086/261868.

I. J. Good. Rational Decisions. *Journal of the Royal Statistical Society. Series B (Methodological)*, 14(1):107–114, 1952. ISSN 0035-9246.

Ozzie Gooen. Scorable Functions: A Format for Algorithmic Forecasting, May 2024.

Katja Grace. How to buy a truth from a liar, July 2014.

Alex Graves. Adaptive Computation Time for Recurrent Neural Networks, February 2017.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. Approaching Human-Level Forecasting with Language Models, February 2024a.

Danny Halawi, Fred Zhang, Chen Yueh-Han, and Jacob Steinhardt. Approaching Human-Level Forecasting with Language Models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, November 2024b.

Joseph Y. Halpern. *Reasoning about Uncertainty, Second Edition*. MIT Press, April 2017. ISBN 978-0-262-53380-5.

Joseph Y. Halpern and Rafael Pass. I Don't Want to Think About it Now:Decision Theory With Costly Computation. *CoRR*, abs/1106.2657, 2011.

Joseph Y. Halpern and Rafael Pass. Algorithmic Rationality: Game Theory with Costly Computation. *CoRR*, abs/1412.2993, 2014.

Yizeng Han, Gao Huang, Shiji Song, Le Yang, Honghui Wang, and Yulin Wang. Dynamic Neural Networks: A Survey, December 2021.

Robin Hanson. Logarithmic Market Scoring Rules for Modular Combinatorial Information Aggregation. *The Journal of Prediction Markets*, 1(1):3–15, January 2002. doi: 10.5750/jpm.v1i1.417.

Robin Hanson. Combinatorial Information Market Design. *Information Systems Frontiers*, 5 (1):107–119, January 2003. ISSN 1572-9419. doi: 10.1023/A:1022058209073.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Jk Gts Hintikka and G. Sandu. Game-Theoretic Semantics. In Benthem and Meulen, editors, *Handbook of Logic and Language*. MIT Press, 1997.

John H. Holland. Properties of the Bucket Brigade. In *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 1–7, USA, July 1985. L. Erlbaum Associates Inc. ISBN 978-0-8058-0426-3.

Elvis Hsieh, Preston Fu, and Jonathan Chen. Reasoning and tools for human-level forecasting. *arXiv preprint arXiv:2408.12036*, 2024.

Jie Huang and Kevin Chen-Chuan Chang. Towards Reasoning in Large Language Models: A Survey. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 1049–1065, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.67.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate. *arXiv preprint arXiv:1805.00899*, 2018a.

Geoffrey Irving, Paul Christiano, and Dario Amodei. AI safety via debate, October 2018b.

Karim Jamal, Michael S. Maier, and Shyam Sunder. Simple Agents, Intelligent Markets. *SSRN Electronic Journal*, 2015. doi: 10.2139/ssrn.2478665.

Myeongjun Jang and Thomas Lukasiewicz. Consistency analysis of ChatGPT. *arXiv preprint arXiv:2303.06273*, 2023.

Giorgi Japaridze. In the beginning was game semantics. In *Games: Unifying Logic, Language and Philosophy*, pages 249–350. Springer Verlag, Berlin, 2009. ISBN 978-1-4020-9373-9. doi: 10.1007/978-1-4020-9374-6_11.

Giorgi Japaridze. Survey of Computability Logic. http://www.csc.villanova.edu/~japaridz/CL/, 2015.

Kaarel, gekaklam, Walter Laurito, Kay Kozaronek, AlexMennen, and June Ku. Searching for a model's concepts by their shape – a theoretical framework, February 2023.

Daniel Kahneman, Ilana Ritov, David Schkade, Steven J Sherman, and Hal R Varian. Economic preferences or attitude expressions?: an analysis of dollar responses to public issues. *Elicitation of preferences*, pages 203–242, 2000.

Ezra Karger, Houtan Bastani, Chen Yueh-Han, Zachary Jacobs, Danny Halawi, Fred Zhang, and Philip E Tetlock. Forecastbench: A dynamic benchmark of ai forecasting capabilities. *arXiv preprint arXiv:2409.19839*, 2024.

Jacek Karwowski, Oliver Hayman, Xingjian Bai, Klaus Kiendlhofer, Charlie Griffin, and Joar Max Viktor Skalse. Goodhart's Law in Reinforcement Learning. In *The Twelfth International Conference on Learning Representations*, October 2023.

Zachary Kenton, Noah Y. Siegel, János Kramár, Jonah Brown-Cohen, Samuel Albanie, Jannis Bulian, Rishabh Agarwal, David Lindner, Yunhao Tang, Noah D. Goodman, and Rohin Shah. On scalable oversight with weak LLMs judging strong LLMs, July 2024.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R. Bowman, Tim Rocktäschel, and Ethan Perez. Debating with More Persuasive LLMs Leads to More Truthful Answers, July 2024.

Ivo Kwee, Marcus Hutter, and Juergen Schmidhuber. Market-Based Reinforcement Learning in Partially Observable Worlds. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 865–873. arXiv, 2001. doi: 10.48550/arXiv.cs/0105025.

Li-Cheng Lan, Huan Zhang, Ti-Rong Wu, Meng-Yu Tsai, I Wu, Cho-Jui Hsieh, et al. Are AlphaZero-like agents robust to adversarial perturbations? *arXiv preprint arXiv:2211.03769*, 2022.

Leonard E Read. I, Pencil: My Family Tree as told to Leonard E. Read. *The Freeman*, 8: 32–37, December 1958.

Nian Li, Chen Gao, Mingyu Li, Yong Li, and Qingmin Liao. EconAgent: Large Language Model-Empowered Agents for Simulating Macroeconomic Activities. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15523–15536, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.829.

Tao Li, Vivek Gupta, Maitrey Mehta, and Vivek Srikumar. A logic-driven framework for consistency of neural models. *arXiv preprint arXiv:1909.00126*, 2019.

Xiang Lisa Li, Vaishnavi Shrivastava, Siyan Li, Tatsunori Hashimoto, and Percy Liang. Benchmarking and improving generator-validator consistency of language models. *arXiv preprint arXiv:2310.01846*, 2023.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Luke Muehlhauser. How Feasible Is Long-range Forecasting?, October 2019.

Lionel W. McKenzie. On the Existence of General Equilibrium for a Competitive Market. *Econometrica*, 27(1):54–71, 1959. ISSN 0012-9682. doi: 10.2307/1907777.

Aidan McLau. AI Search: The Bitter-er Lesson. https://news.ycombinator.com/item?id=40683697, June 2024.

Nolan Miller. Notes on Microeconomic Theory. (Lecture Notes), August 2006.

Marvin Minsky. *Society Of Mind*. Simon and Schuster, March 1988. ISBN 978-0-671-65713-0.

Eric Neyman. Algorithmic Bayesian Epistemology, March 2024.

Caspar Oesterheld, Abram Demski, and Vincent Conitzer. A Theory of Bounded Inductive Rationality. *Electronic Proceedings in Theoretical Computer Science*, 379:421–440, July 2023. ISSN 2075-2180. doi: 10.4204/EPTCS.379.33.

OpenAI. Learning to Reason with LLMs, September 2024.

Long Phan, Adam Khoja, Mantas Mazeika, and Dan Hendrycks. LLMs are superhuman forecasters, 2024. URL https://www.safe.ai/blog/forecasting. Accessed on 26-Sept-2024.

Ansh Radhakrishnan. Anthropic Fall 2023 Debate Progress Update. November 2023.

Nasim Rahaman, Martin Weiss, Manuel Wüthrich, Yoshua Bengio, Li Erran Li, Chris Pal, and Bernhard Schölkopf. Language Models Can Reduce Asymmetry in Information Markets, March 2024.

Harsh Raj, Vipul Gupta, Domenic Rosati, and Subhabrata Majumdar. Semantic consistency for assuring reliability of large language models. *arXiv preprint arXiv:2308.09138*, 2023.

Yuma Rao, Jacob Steeves, Ala Shaabana, Daniel Attevelt, and Matthew McAteer. BitTensor: A Peer-to-Peer Intelligence Market, November 2021.

Leonard J. Savage. Elicitation of Personal Probabilities and Expectations. *Journal of the American Statistical Association*, 66(336):783–801, 1971. ISSN 0162-1459. doi: 10.2307/2284229.

Jürgen Schmidhuber. Evolutionary principles in self-referential learning, or on learning how to learn: The meta-meta-. hook. 1987.

Jurgen Schmidhuber. A Local Learning Algorithm for Dynamic Feedforward and Recurrent Networks. *Connection Science*, 1(4):403–412, January 1989. ISSN 0954-0091. doi: 10.1080/09540098908915650.

Philipp Schoenegger and Peter S. Park. Large Language Model Prediction Capabilities: Evidence from a Real-World Forecasting Tournament, October 2023.

Philipp Schoenegger, Indre Tuminauskaite, Peter S. Park, and Philip E. Tetlock. Wisdom of the Silicon Crowd: LLM Ensemble Prediction Capabilities Rival Human Crowd Accuracy, May 2024.

Alan Schwartz. How Much Irrationality Does the Market Permit? *The Journal of Legal Studies*, 37(1):131–159, January 2008. doi: 10.1086/519963.

Glenn Shafer and Vladimir Vovk. *Probability and Finance: It's Only a Game!* John Wiley & Sons, February 2005. ISBN 978-0-471-46171-5.

Glenn Shafer and Vladimir Vovk. *Game-Theoretic Foundations for Probability and Finance*. John Wiley & Sons, May 2019. ISBN 978-0-470-90305-6.

Arnab Sharma and Heike Wehrheim. Testing monotonicity of machine learning models, 2020.

Sergei Soloviev. The Verifier-Falsifier Games with Restrictions on Computational Complexity of Strategies, 2022.

Abhimanyu Pallavi Sudhir, Alejandro Alvarez, Adam Shen, and Daniel Paleka. Consistency Checks for Language Model Forecasters. In *Agentic Markets Workshop at ICML 2024*, July 2024.

tailcalled. Latent variables for prediction markets: Motivation, technical guide, and design considerations. https://www.lesswrong.com/posts/ufW5LvcwDuL6qjdBT/latent-variables-for-prediction-markets-motivation-technical, February 2023.

Philip E Tetlock, Christopher Karvetski, Ville A Satopää, and Kevin Chen. Long-range subjective-probability forecasts of slow-motion variables in world politics: Exploring limits on expert judgment. *Futures & Foresight Science*, 6(1):e157, 2024.

Stephen Thornton. Karl Popper. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2023 edition, 2023.

Susan Vineberg. Dutch Book Arguments. In Edward N. Zalta and Uri Nodelman, editors, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, fall 2022 edition, 2022.

F. A. von Hayek. Economics and Knowledge. *Economica*, 4(13):33–54, 1937. ISSN 00130427, 14680335.

John Wentworth. Competitive Markets as Distributed Backprop, November 2018.

Qi Yan, Raihan Seraj, Jiawei He, Lili Meng, and Tristan Sylvain. Autocast++: Enhancing world event prediction with zero-shot ranking-based context retrieval. *arXiv preprint arXiv:2310.01880*, 2023a.

Qi Yan, Raihan Seraj, Jiawei He, Lili Meng, and Tristan Sylvain. AutoCast++: Enhancing World Event Prediction with Zero-shot Ranking-based Context Retrieval. In *The Twelfth International Conference on Learning Representations*, October 2023b.

Yann LeCun. Towards Machines that can Learn, Reason, and Plan. In *AI and Barrier of Meaning Workshop*, Santa Fe Institute, May 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, pages 11809–11822, Red Hook, NY, USA, May 2024. Curran Associates Inc.

Eliezer Yudkowsky. Coherent Extrapolated Volition, 2004.

Stephan Zheng, Alexander Trott, Sunil Srinivasa, David C. Parkes, and Richard Socher. The AI Economist: Optimal Economic Policy Design via Two-level Deep Reinforcement Learning, August 2021.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. Forecasting future world events with neural networks. *arXiv preprint arXiv:2206.15474*, 2022a.

Andy Zou, Tristan Xiao, Ryan Jia, Joe Kwon, Mantas Mazeika, Richard Li, Dawn Song, Jacob Steinhardt, Owain Evans, and Dan Hendrycks. Forecasting Future World Events With Neural Networks. *Advances in Neural Information Processing Systems*, 35:27293–27305, December 2022b.