Rapporteur Report: Scalable AI Safety via Doubly-Efficient Debate

Aarushi Lakhi

November 2026

1 Paper summary

1.1 Background: AI Safety via Debate

This paper by Brown-Cohen, Irving, and Piliouras [1] is a follow-up to AI Safety via Debate by Irving, Christiano, and Amodei [2], which we discussed in the previous class. In the Irving et al. proposal, two copies of an AI argue opposite sides of a question. If one of them lies, the other attacks the weakest part of the lie, and the argument narrows until it reaches a small claim that a human judge can evaluate. The judge reads the whole exchange and picks a winner, and training rewards whichever model the judge picked. On the theory side, the 2018 paper showed that with a judge who only does polynomial work, debate can in principle decide any problem in PSPACE, the class of problems that can be solved with polynomial memory even when they take exponential time. Determining who wins a perfectly played game of chess is a standard example.

That result left three gaps, which this paper tries to address. First, the honest debater might need exponential compute to win. To know where to attack the liar's argument, it might have to explore an exponentially large game tree, so the theorem says the truth can win, but a real model with bounded resources would be unable to find the winning moves. Second, Barnes and Christiano [4] later identified the obfuscated arguments problem, where a dishonest debater builds a long argument with one subtle error hidden among many reasonable-looking sub-claims. The honest side may suspect that something is wrong without being able to locate the error in a reasonable amount of time, and a debater who can only say that there is a problem somewhere will lose. Third, the 2018 analysis treated human judgment as deterministic, so the same question always gets the same answer.

1.2 Motivation

The paper motivates the problem with a language model that drafts a long legal contract. Small details in contracts can be really important, and the authors cite a case where a misplaced comma cost the US government $38.4 million [1]. Training such a model with human feedback would mean paying an expert to read the entire contract for every training example, which is not realistic. The goal is a training signal for the whole contract that only asks a human to read a few sentences, where that number of sentences does not grow as the contract gets longer.

The authors compare this to AlphaZero. No human could tell AlphaZero which Go move was best, but it still reached superhuman play through self-play, because at the end of a game it is easy to check who won. It is much harder to check language tasks this way since verifying a long plan can be as hard as writing it,

or harder. The paper presents debate as a way to make "who won?" easier to check for language tasks, so that the same kind of self-play training becomes possible.

1.3 The doubly-efficient debate setup

A debate has two debaters, *A* and *B*, where *A* argues that the answer is yes and *B* argues that it is no, and a verifier *V* that applies the rules and declares a winner. Nobody knows in advance which debater is honest. Human judgment can be called through an oracle *O*, a black box that answers yes-or-no questions such as "does this study have a clear methodological flaw?" Both the debaters and the verifier can query the oracle, but the queries work differently depending on whether it's a debater or the verifier requesting it. When the verifier queries it, a real human rater is asked. When a debater queries it, the debater predicts what a human would say from what it already knows about people. The debaters need these predictions because some steps of the task are human judgments, such as whether a particular study is flawed, and a debater cannot write out the task without filling in those steps. Sending each of these questions to a real person would make human effort grow with the length of the task, which is what the paper is trying to avoid (Figure 1).

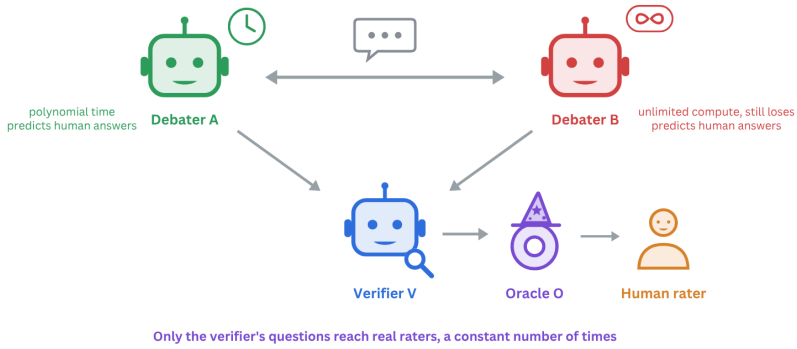


Figure 1. The doubly-efficient debate setup. Both debaters predict human answers from what they already know, while only the verifier's questions go to real human raters. Green and red mark the honest and dishonest roles, which are not known in advance.

A protocol is measured by three quantities: *Ptime*, the computation the honest debater needs; *Vtime*, the computation the verifier needs; and q, the number of questions sent to real humans. "Doubly-efficient" refers to keeping both the honest debater and the verifier efficient, whereas the 2018 paper allowed both debaters unlimited compute. If T is the time needed to solve the problem directly and l is the length of one human question, the targets are roughly $Ptime = T \log T$ for the honest debater, $Vtime = l \log T$ for the verifier, and $q = O(1)$ human questions. The dishonest debater is allowed unlimited compute and should still lose.

The paper also requires two guarantees (Definition 4.1). Completeness says that if the claim is true, the honest debater wins with probability at least c against any opponent. Soundness says that if the claim is false, the dishonest debater wins with probability at most s, even with unbounded time. The paper requires $c > 1/2 > s$, so the debate does better than a coin flip and can be repeated with a majority vote until the truth wins almost every time. In the deterministic protocols $c = 1$ and $s = 0$, and in the stochastic protocol $c = 3/5$ and $s = 2/5$.

1.4 Deterministic debate

All of the protocols rely on writing a computation as a transcript, which is the sequence of steps a machine M takes on an input x, ending in the answer. For a language model following instructions, the transcript is the sequence of tokens it writes, so in practice it corresponds to the model's chain of thought. Each step only depends on a few earlier cells, which the paper calls $I(t)$, so checking whether step t is correct does not require reading the whole transcript. Some steps are ordinary computation, and some are oracle queries that ask a human for a judgment.

The first protocol (Theorem 5.1) is a binary search. A commits to the state of the computation halfway through, and B says which half contains the mistake, meaning either the midpoint does not follow from the start or the end does not follow from the midpoint. They repeat on that half, and after about log T rounds they reach a single step, which the verifier checks directly with at most a constant number of human questions. The honest debater only needs to run the computation once, but the verifier has to read a full intermediate state every round, and those states can be as large as the memory S that the computation uses. The resulting costs are $O(T \log T)$ for the honest debater, $O(S \log T)$ for the verifier, and $O(1)$ human questions.

The second protocol (Theorem 5.3) removes the verifier's dependence on S using cross-examination, an idea from Barnes and Christiano [3]. In a debate with cross-examination, the parties can question independent copies of a debater that cannot see the rest of the conversation. For language models this is cheap, because the same model can be rerun with the earlier messages removed from its context. The verifier can then ask a fresh copy of A what it wrote at a given position, and A cannot tailor its answer to the state of the debate. The protocol has three steps. A writes out the full transcript, B points to a single step t along with the cells $I(t)$ it depends on, and the verifier reads only those cells and recomputes step t, asking a human at most once. If the step is correct A wins, and otherwise B wins (Figure 2). The costs are $O(T \log T)$ for the honest debater, $O(l \log T)$ for the verifier, and $O(1)$ human questions. This version barely looks like a debate, because the rounds of the binary search have been compressed into a single move by B. The adversarial pressure now comes entirely from B's freedom to choose which step to challenge, and cross-examination is what lets the verifier read just that step without trusting A's account of the rest.

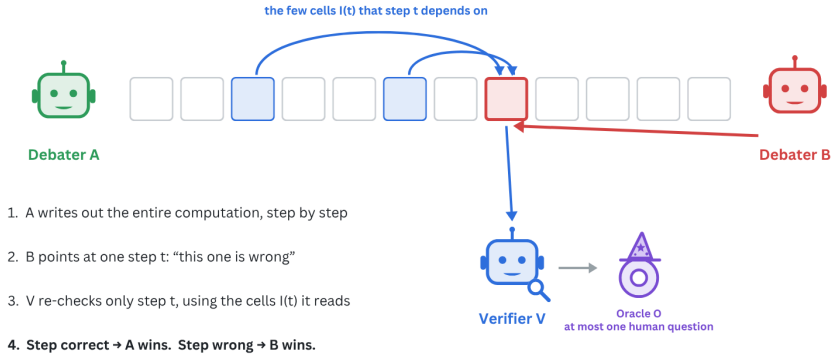


Figure 2. The cross-examination protocol (Theorem 5.3). A writes out the full transcript, B points to one step t, and the verifier recomputes that step from the cells I(t) it depends on, asking a human at most once.

The honest side always wins this protocol. If A is honest, every step of its transcript is correct, so wherever B points, the step checks out. If A is lying about the answer, its transcript has to break somewhere, since correct steps cannot lead to a wrong conclusion. There is therefore a first broken step, and an honest B can find it by running the computation itself in time T, with no exponential search (Figure 3). This is how the paper avoids obfuscated arguments, although it does so by restricting attention to computations whose full transcript has polynomial length and could in principle be checked by a human. The authors note in their conclusion that allowing both debaters to be unbounded brings the obfuscation problem back [1].

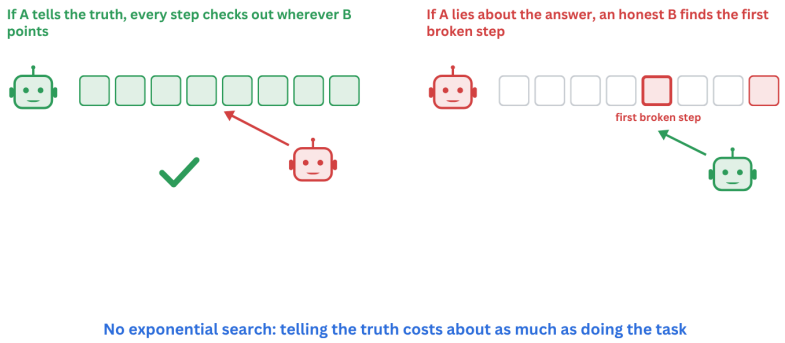


Figure 3. Why the honest side wins. If A is honest, any step B points to checks out. If A lies about the answer, an honest B can find the first broken step by running the computation itself.

1.5 Stochastic debate

The paper's running example for this section is a researcher who wants to know whether caffeine improves learning in college students. The plan, which plays the role of the machine M, is to search the literature, keep only studies that have no clear methodological flaw and are randomized controlled trials measuring final exam performance after at least six weeks, run a random-effects model, and check heterogeneity with meta-regression. Only the step about methodological flaws requires human judgment, so it is the step that becomes an oracle query.

Because humans disagree on things, the paper models the oracle as a weighted coin, where each query is like asking a random person. This creates a problem, because small errors in estimating human answers can add up over a long computation. If a task takes the average of a thousand answers, a few unusual answers barely change the result, but if a task computes the parity of those answers (whether the number of yeses is odd or even), one person changing their mind flips it. No one would run a meta-analysis that computes a parity, but milder versions of the same sensitivity are realistic, such as a conclusion that depends on whether one borderline study is included.

The paper handles this by restricting to K-Lipschitz tasks (Definition 6.1). If every human's probability of answering yes shifts by at most ε, the probability of the final answer shifts by at most $K\varepsilon$. For example, if the rate at which raters call study 17 flawed moves from 75% to 76% and $K = 5$, the probability of the final conclusion moves by at most five percentage points. Parity has a K that grows with the size of the task. A majority vote, which asks whether the average answer is above one half, has a small K when most people agree, but near a 50/50 split a small shift in each person's probability can flip the outcome, and K

grows with the square root of the number of voters (about 8 for 100 raters). Every time-T computation is K-Lipschitz for $K = O(T)$, so the result only gives savings when K is much smaller than $\sqrt{T}$, and the case the authors focus on is K constant.

In the stochastic protocol (Theorem 6.2), A goes through the computation step by step, and at each step it states the probability that the step comes out to 1, for example that a human would call the study flawed with probability 0.75. A and B then flip a coin together to decide the step's actual value. Each contributes a random number between 0 and 1, the two are added modulo 1, and the step is set to 1 if the sum is at most 0.75. As long as one of the two numbers is random, the sum is random, so neither side can rig the outcome, and cross-examination ensures that neither sees the other's number first. If B thinks one of A's probabilities is wrong, its only move is to object at that step, which ends the debate. The verifier then asks about K^2 real people the same question and checks whether the fraction who say yes is close to A's claimed probability. If it is close A wins, and otherwise B wins. If nobody objects, the verifier reads off the final answer (Figure 4).

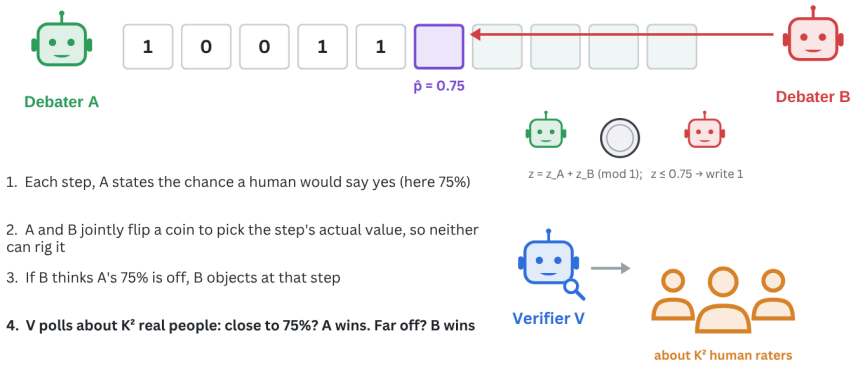


Figure 4. The stochastic protocol (Theorem 6.2) at a step where a human says yes with probability 0.75. A states the probability, A and B jointly flip a coin to fix the step's value, and if B objects, the verifier polls about K^2 real raters on that one question.

In the caffeine example, A might estimate that 75% of people would call study 17 flawed and exclude it, and B might object that the study looks fine. The question would then go to real raters, and if (hypothetically) 72% of them say the study is flawed, A wins. The costs are $O(K^2\, T \log T)$ for the honest debater, $O(K^2 + l \log T)$ for the verifier, and $O(K^2)$ human questions, with completeness 3/5 and soundness 2/5. When K is a constant, the human effort does not grow with the length of the task. The authors also formalized this theorem in the Lean 4 proof assistant [1].

1.6 Debate with a witness

The last two results drop the assumption that a human wrote the plan. Instead, the researcher could ask, “Does caffeine improve learning in college students? Produce a high-level plan to answer this question, give a step-by-step justification of the correctness of the plan, and then execute the plan.” The plan together with its justification is called a witness. It can be hard to come up with, but it can be checked step by step, like a solution to an NP problem. The machine M then becomes “check every step of the plan and

its justification, then execute the plan," and the earlier protocols are run on the question together with the witness. Combining the witness with cross-examination gives Theorem 7.1, and combining it with the stochastic protocol gives Theorem 7.2. In both cases, a human only has to check a constant number of steps, and the debate itself is the same as before: *A* commits to the plan and carries it out, and *B* chooses where to object. The authors argue that this is the setting that matters most for safety, since it corresponds to an AI proposing a plan and arguing that carrying it out will have the desired outcome.

1.7 Training and deployment

The paper also describes how the protocols would be used in practice. During training, *A* and *B* are trained through self-play to win the debate game, and when a debater queries the oracle, it is predicting a human answer from its pretrained knowledge or from what it has learned during training. Only the verifier's queries go to real human raters, so the number of human labels per debate stays constant while the debaters' computation can grow. At deployment, no human feedback is used, and *A*'s output is trusted as long as *B* does not object to a step.

2 My thoughts

The most interesting idea to me in this paper was how it changes where debate gets its power. In the 2018 paper, the debaters had unbounded compute, which is what made debate powerful. With unbounded debaters, a judge doing only polynomial work could decide anything in PSPACE, including problems no human could check directly. This paper gives up that reach to make the protocol efficient. Its theorems cover computations that already have a polynomial-length transcript a careful human could check step by step, and the contribution is that the human only has to check one of those steps. That makes the result useful for tasks like the 300-page contract, where the expense is in the reading, but it does say anything about reasoning that cannot be broken into human-checkable steps at all, which is the case scalable oversight is most concerned with. The authors raise this themselves in their open problems [1].

Because of this, much of the difficulty moves into the machine *M*. Each step has to be small enough for a person to judge on its own, and the guarantees only hold relative to the steps *M* defines. A question like "is study 17 flawed?" seems like a single step, but answering it well may depend on context spread across the study and the field, and a plan whose most important judgment never appears as an explicit step would pass every check. In the witness setting, the AI writes the plan, so it also chooses what the human is eventually asked. I think this is closely related to Professor Aaronson's objection in class, described in Section 3.3, that real disagreements are usually about which framework to use.

A related detail is that when the verifier queries the human oracle, the query is built from cells that *A* wrote. If *A* is lying, the dishonest debater writes the exact text the human reads. The paper treats the oracle as a fixed function of its input, so wording has no effect beyond meaning, but a debater with unbounded compute could search over phrasings for one that a rater will approve. There is already evidence of this kind of pressure from training. Wen et al. [7] found that RLHF made models better at convincing time-limited human evaluators without making them more correct, raising evaluators' false positive rate by 24.1% on a question-answering task and 18.3% on a programming task. In practice, the oracle may also be an LLM judge, in which case a debater who writes the cells the judge reads controls

part of the judge's prompt. The protocol narrows a disagreement down to one step, but it does not ensure that the step is presented fairly.

I also think the stochastic model is more forgiving than real human judgment. Each rating is treated as an independent draw from a weighted coin, so disagreement averages out once enough people are polled. Real raters are often wrong in the same direction, for example, when they share a framing, and polling more of them only makes the verifier more confident in the shared mistake. The K-Lipschitz condition adds a second limit. As Professor Aaronson pointed out in class, it means the protocol will not work in many contexts, and I suspect that many questions we would want AI help with, such as policy decisions, are more sensitive to individual judgments than a meta-analysis that averages studies. The constants are also large. Taken literally, the protocol has the verifier collect around 20 million ratings to settle one objection even when $K = 1$, although the authors note that their Lean formalization uses different constants.

Finally, the guarantees say that the honest strategy wins if a debater plays it, but not that training will produce such a debater. This matters most at deployment, where A's answer is trusted unless B objects, since a B that missed an error will respond the same as a B that checked carefully and found nothing. A later safety case sketch by Buhl, Pfau, Hilton, and Irving [8] lists these conditions as assumptions, including that training reaches an approximate equilibrium of the debate game and that debaters do not avoid high-reward strategies during training, and the authors conclude that current techniques cannot yet establish them.

3 Class discussion

3.1 Obfuscated arguments and Lean

A student proposed a test case for obfuscated arguments. Suppose one debater submits a 100-page Lean proof, and the other believes it exploits a bug in the Lean kernel, but setting up the exploit takes many lines, so there is no single line to point to. Several students noted that the paper does not allow a general claim that something is wrong somewhere, since the challenger has to name a specific step t and the cells $I(t)$ it depends on, and Professor Aaronson agreed that the setup requires this. The class did not settle whether this requirement solves the problem or only defines it away.

Something very similar happened in July 2026, when Ramana Kumar posted an AI-assisted Lean "disproof" of the Collatz conjecture that used no sorry placeholders or extra axioms. Lean accepted it because it exploited a bug in how the kernel handled nested inductive types, along with a separate bug in the independent checker Nanoda, and the Lean team patched the kernel within hours of the report [9]. Every step of that proof was valid under the rules being checked, so the error was in the rules, which in the paper's terms is an error in M. The protocol can only locate a wrong step relative to a fixed definition of a correct step, so it has no way to represent this kind of failure. This also bears on a piece of news Professor Aaronson had opened class with, namely OpenAI's claimed disproof of the Exponential Time Hypothesis, which came with a Lean certificate. The certificate makes the proof much easier to trust, but that trust now rests on the kernel and on whether the formal statement matches the intended one.

3.2 Debate between current models

I asked whether the protocol could be run on current reasoning models. Several students had tried informal versions, such as asking Claude for a proof and ChatGPT to find the error, or using two models to debug the same code. Depending on the phrasing, the critic either found pedantic issues with notation or agreed with the other model, and one student said that models now tend to find the same bug, which makes them less useful as independent checkers. Another student pointed out that hidden chains of thought make the transcript-based protocols hard to try at all.

These experiences match recent research. Wynn, Satija, and Hadfield [10] found that in multi-agent debate, models often switch from correct answers to incorrect ones after seeing their peers' reasoning, partly because they favor agreement over challenging flawed reasoning. Debate has done best on reading comprehension. Khan et al. [5] found that non-expert judges answered correctly 88% of the time with debate, compared with 60% without it, and Kenton et al. [11] found that debate helped weak LLM judges most when the debaters could read a text the judge could not, with mixed results on math, coding, and logic. Doubly-efficient debate is aimed at this second kind of task, where the judge has all the information and the bottleneck is effort. The informal debates my classmates described also had no rule forcing the critic to name a single step and no limit on the check to that step, which is the structure the paper's guarantees depend on.

3.3 Disagreements about frameworks

Professor Aaronson argued that most real-life debates do not turn on a long computation that could be checked step by step. They are usually about which assumptions and which framework are reasonable, with many factual disagreements following from the more basic one, and he was not sure how these protocols would help. A student offered an analogy that points the other way, where two political factions prepare policy plans privately and then debate them in public. Under the paper's assumptions, the public would only need to check the one step that a faction attacks, and an honest faction, if there is one, is guaranteed to win.

The caffeine example falls between these two views. Its plan includes choices a skeptic could dispute, such as keeping only randomized trials that last at least six weeks. In the basic protocol, the plan is M, so those choices are fixed before the debate starts. In the witness version, they can be challenged, but only by asking whether a justification step is clearly correct, which is an awkward question to ask about a choice of framework. Political disagreements have the same problem, since they are often about what a plan should aim for, and no single step of either plan can settle that. Debate seems best suited to disputes where both sides already agree on what evidence would decide the question.

3.4 Hidden reasoning and honest mistakes

I asked how useful the protocols could be if models can hide or edit their chain of thought. Professor Aaronson distinguished a model's private scratchpad from the public argument it gives in the debate, and a student added that as long as the public transcript is committed before anyone inspects it, edits to the private reasoning do not affect the protocol. I think this distinction works in debate's favor. Anthropic found that Claude 3.7 Sonnet mentioned a hint it had used in its reasoning only about 25% of the time,

and DeepSeek R1 about 39% of the time [12], which is a problem for methods that rely on reading a model's chain of thought. Debate checks the public transcript as a computation, so if every step is correct, the answer is correct regardless of how the model arrived at it.

I also asked whether a model that makes one mistake should count as a liar, since a single wrong step loses the debate even when the final answer is right. Professor Aaronson argued that a debate aimed at the truth has to let a side retract a point without conceding everything, as academics do in exchanges of letters, but that retractions should be limited, since a debater that restarts its argument every turn never had a reason for its position. Michael et al. [13] found that in debates between humans, 46% of errors came from mistakes by the honest debater, the largest single source, so this is a central issue in practice. The theory assumes a perfectly honest and reliable debater, while the safety case sketch [8] targets a small error rate instead, and prover-estimator debate [6] lets a debater assign probabilities to subclaims, which gives it a way to be uncertain without being treated as a liar.

3.5 Shared blind spots

I raised the case where both debaters believe the same false thing, which seems likely if they are copies of one model. In that case nobody objects, and the verifier never sees the bad step. Professor Aaronson compared this to the Einstein and Bohr debates, which philosophers of science still try to score even though both sides were missing Bell's theorem. When I asked whether the debaters should be different models, he was unsure, since one model can take on many personas with different prompts.

I am not convinced that personas solve this. A persona changes what a model says, but its mistakes come from its training. Goel et al. [14] found that language models' mistakes are becoming more similar as their capabilities increase, and that LLM judges score models more favorably when those models are similar to themselves. For this paper's guarantees, what matters is whether the debaters' errors are correlated, and assigning two copies of a model to opposite sides does not change that.

3.6 Predicting human judgments

The last question I raised was how well debaters can predict human answers, since they never talk to real people. A student suggested a loophole for the contract example, where a debater's single step is "a human reviewed the whole contract and approved it," which puts the full burden back on the oracle. Others responded that B would have to point to the specific part it thinks is wrong. This led to a question the class did not resolve, which is how many words one step can contain.

I think this question is central to the paper's main claim. The verifier's cost depends on l, the length of a query, but nothing in the theorems limits how much a human has to read or think to answer one. A protocol that asks one question about an entire contract meets the constant-question guarantee without reducing human effort, so the real measure of effort is the size of a step, which the paper leaves to whoever writes M.

References

[1] J. Brown-Cohen, G. Irving, and G. Piliouras. Scalable AI Safety via Doubly-Efficient Debate. arXiv:2311.14125, 2023 (ICML 2024).

[2] G. Irving, P. Christiano, and D. Amodei. AI Safety via Debate. arXiv:1805.00899, 2018.

[3] B. Barnes and P. Christiano. Writeup: Progress on AI Safety via Debate. AI Alignment Forum, 2020.

[4] B. Barnes and P. Christiano. Debate Update: Obfuscated Arguments Problem. AI Alignment Forum, 2020.

[5] A. Khan et al. Debating with More Persuasive LLMs Leads to More Truthful Answers. ICML 2024.

[6] J. Brown-Cohen, G. Irving, and G. Piliouras. Avoiding Obfuscation with Prover-Estimator Debate. arXiv:2506.13609, 2025.

[7] J. Wen, R. Zhong, A. Khan, E. Perez, J. Steinhardt, M. Huang, S. R. Bowman, H. He, and S. Feng. Language Models Learn to Mislead Humans via RLHF. arXiv:2409.12822, 2024.

[8] M. D. Buhl, J. Pfau, B. Hilton, and G. Irving. An Alignment Safety Case Sketch Based on Debate. arXiv:2505.03989, 2025.

[9] Gigazine. An AI-assisted Collatz “disproof” exposed a bug in the Lean kernel. August 2026.

[10] A. Wynn, H. Satija, and G. Hadfield. Talk Isn't Always Cheap: Understanding Failure Modes in Multi-Agent Debate. arXiv:2509.05396, 2025.

[11] Z. Kenton et al. On Scalable Oversight with Weak LLMs Judging Strong LLMs. NeurIPS 2024.

[12] Anthropic. Reasoning Models Don't Always Say What They Think. April 2025.

[13] J. Michael et al. Debate Helps Supervise Unreliable Experts. arXiv:2311.08702, 2023.

[14] S. Goel et al. Great Models Think Alike and This Undermines AI Oversight. ICML 2025.