

TRUST: A Crowdsourcing Framework for Auditing Large Language Model Reasoning

Morris Yu-Chao Huang¹ Zhen Tan² Mohan Zhang¹
Pingzhi Li¹ Zhuo Zhang³ Tianlong Chen¹

¹UNITES Lab, University of North Carolina at Chapel Hill

²Arizona State University ³Columbia University

Corresponding author: morris@cs.unc.edu

Abstract

Large Language Models (LLMs) can produce complex reasoning chains, offering a window into their decision-making processes. However, verifying the quality (e.g., faithfulness and correctness) of these intermediate steps is a critical, unsolved challenge. Current auditing methods usually rely on a centralized auditor, either human or verifier model, to inspect full reasoning traces. This paper addresses three key challenges in those methods. **Robustness:** Centralized systems are single points of failure, vulnerable to attacks and systemic bias. **Scalability:** The lengthiness and complexity of reasoning traces create a severe bottleneck for human auditors. **IP confidentiality:** Model providers risk intellectual property theft or model distillation when exposing complete reasoning traces. To overcome these barriers, we introduce TRUST, a crowdsourcing framework for auditing LLM reasoning. TRUST makes the following contributions: First, it establishes a consensus mechanism among a diverse set of auditors that removes reliance on any *single trusted auditor*: under the modeled seat-corruption process the trace-level failure bound holds up to a 20% adversarial seat fraction, and empirically TRUST retains 69.0% accuracy with one compromised seat per segment, where centralized baselines collapse to $\sim 20\%$. Second, it introduces a scalable decomposition method that transforms reasoning traces into hierarchical directed acyclic graphs, enabling atomic reasoning steps to be audited in parallel by a crowd of independent auditors. Third, the framework limits exposure by distributing only partial segments of the reasoning trace to auditors. We provide theoretical guarantees for the security and economic incentives. Experiments across multiple LLMs (e.g., GPT-OSS, DeepSeek-r1, Qwen) and reasoning tasks (e.g., math, medicine, science, and humanities) demonstrate that TRUST is highly effective at identifying reasoning flaws and is significantly more resilient to corrupted and adversarial auditors than centralized baselines. Our work introduces an auditing framework for reasoning AI systems, enabling safer and secure deployment in high-stakes settings.

1 Introduction

Large language models (LLMs) have expanded from text generation to complex, multi-step reasoning, producing Large Reasoning Models (LRMs) that emit explicit reasoning traces (Wei et al., 2022). This exposes a model’s logical flow, but also its flaws: logical errors, unfaithfulness to the true internal state (Turpin et al., 2023), and safety vulnerabilities. Verifying these intermediate steps is a prerequisite for deploying LRMs in high-stakes domains such as medicine (Singhal et al., 2023), law (Chalkidis et al., 2021), and finance (Wang et al., 2023), and is made urgent by regulatory frameworks like the EU AI Act (COM, 2021) and the NIST AI RMF (AI, 2023), which mandate rigorous documentation and monitoring (OECD, 2019). However, even as recent works advance auditing for semi-structured reasoning (Leng

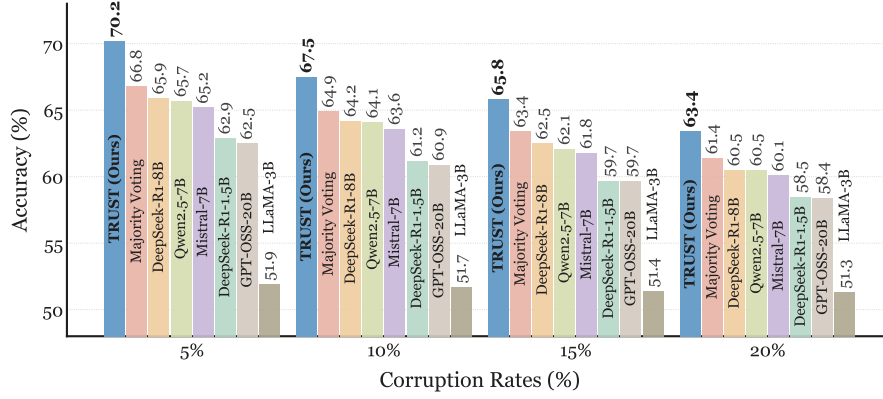


Figure 1: TRUST remains effective at auditing reasoning flaws in the MMLU-Pro-CoT-Train dataset (Lab, 2024) as the proportion of corrupted auditors increases. Metric: trace-level audit accuracy (%), $n = 200$ annotated traces, random segment-verdict corruption; Table 2c reports adaptive, correlated-bias and compromised-seat attacks.

et al., 2025) or propose new faithfulness metrics (Lanham et al., 2023), prevailing auditing methods remain misaligned with this paradigm. Their centralized, opaque, and unscalable nature creates unacceptable risks, as they either rely on a single trusted entity, cannot process the volume and complexity of reasoning traces, or force a dangerous trade-off between public transparency and the protection of proprietary models.

The inadequacy of current auditing systems stems from three interconnected challenges. First, a lack of **Robustness**: systems relying on a single auditor, human expert or LLM, are a “single point of failure”, vulnerable to prompt injection (Zou et al., 2023; Perez & Ribeiro, 2022) and to systemic bias (Bender et al., 2021; Liang et al., 2022). Second, a severe **Scalability** bottleneck: the volume and combinatorial complexity of traces from modern LLMs, especially under branching search (Lightman et al., 2023; Yao et al., 2023a), make comprehensive manual verification economically infeasible, as the human effort behind existing process-supervision datasets shows (Bai et al., 2022; Lightman et al., 2023). Additionally, simple crowdsourcing audit creating **IP confidentiality** issues since exposing complete reasoning traces for public audit risks the theft of valuable intellectual property through model distillation (Carlini et al., 2021) and increases the surface area for extracting sensitive training data (Nasr et al., 2023). We deliberately say *confidentiality* rather than *privacy* throughout: the property at stake is protection of the provider’s proprietary reasoning logic, not protection of end-user data (see the Ethics Statement).

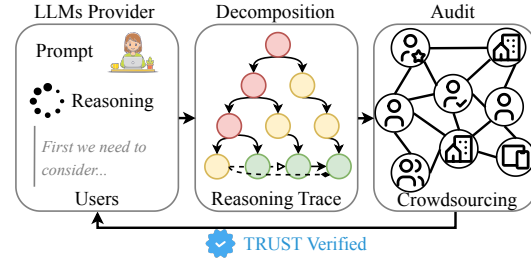


Figure 2: Reasoning traces (left) are decomposed into hierarchical segments (middle) and verified by a heterogeneous crowd of computer, LLM, and human auditors (right), whose crowdsourcing consensus produces a TRUST-verified audit.

Addressing these simultaneous challenges of robustness, scalability, and confidentiality demands a new approach to the auditing paradigm. Our work answer the following research questions:

RQ 1. How can we design an auditing system that is robust to malicious participants and systemic bias without relying on a single trusted auditor

RQ 2. How can this system scale to audit complex reasoning traces while preserving the intellectual property of the model provider and ensuring public transparency

Answering them leads to a framework integrating crowdsourcing, confidentiality-aware protocols, and a novel decomposition for reasoning itself. We introduce TRUST, a crowdsourcing framework for auditing LLM reasoning (see Figure 2). To achieve **robustness**,

TRUST establishes a consensus mechanism among a diverse, multi-tier set of auditors, drawing on principles from Byzantine Fault Tolerant systems (Castro et al., 1999; Lamport et al., 2019) to bound the trace-level failure probability under a modeled seat-corruption process. For **scalability**, the framework introduces a novel decomposition method that transforms reasoning traces into *Hierarchical Directed Acyclic Graphs* (HDAGs), a structured representation that permits parallel verification of atomic reasoning steps by a distributed network. To address **IP confidentiality**, the protocol distributes only partial, disconnected trace segments to individual auditors, raising the cost of full-trace reconstruction; Table 1 quantifies what a colluding coalition can still recover. The HDAG parser and the assignment platform remain trusted components: TRUST eliminates the single trusted *auditor*, not every trusted element of the pipeline (Section 5). We show empirically that the parser, although non-deterministic and inconsistent at the label level, does not move the trace verdict (Section E).

We prove a *Safety–Profitability Guarantee* theorem ensuring that, under configurable statistical and economic conditions, the system achieves a target audit safety level while making honest participation profitable and malicious behavior loss-making. Our empirical validation spans multiple LLMs (e.g., GPT-OSS (OpenAI, 2025), DeepSeek-r1 (DeepSeek-AI et al., 2025), Qwen (Yang et al., 2025)) and diverse reasoning tasks, with human-in-the-loop experiments validating the multi-tier design, and shows TRUST is markedly more resilient to coordinated attacks than centralized baselines.

In summary, our main contributions are:

- We introduce TRUST, to our knowledge the first crowdsourcing auditing system for reasoning traces that achieves confidentiality-aware verification without exposing the full trace of the proprietary model.
- We develop a systematic approach to decompose Chain-of-Thought reasoning into Hierarchical Directed Acyclic Graphs (HDAGs) that enable modular verification coupled with a multi-tier verification, routing simple problems to automated validators and complex problems to human experts.
- We state an explicit collusion threat model for crowdsourced reasoning audit and measure residual leakage along three axes (token recall, order recall, segment coverage), a property prior auditing work leaves implicit.
- We develop theoretically grounded incentive mechanisms which ensure that honest auditors profit while malicious actor incur losses, providing the foundation necessary for sustainable real-world deployment at scale.
- We conduct comprehensive experiments on diverse datasets (e.g., MMLU-Pro, GSM8K) and models (e.g., GPT-OSS, DeepSeek-r1), including human-in-the-loop studies, to demonstrate the effectiveness and robustness of TRUST against centralized baselines, under random corruption, adaptive targeted attack, correlated auditor bias and directly compromised auditor seats.

2 Related Works

Reasoning Model Verification. Chain-of-Thought (CoT) prompting has revolutionized LLM reasoning by exposing intermediate steps (Wei et al., 2022), evolving into sophisticated tree-based search methods (Yao et al., 2023a) and Large Reasoning Models that treat reasoning as a primary objective (Jaech et al., 2024; Guo et al., 2025). However, these advances lack systematic verification mechanisms for generated reasoning traces.

Auditing and Evaluation. Current auditing approaches range from centralized “LLM-as-a-judge” methods (Zheng et al., 2023) to Process Reward Models that provide step-by-step supervision (Lightman et al., 2023). Leng et al. (2025) audits semi-structured reasoners via formal verification, and Lanham et al. (2023) measures chain-of-thought faithfulness; both presume a centralized verifier. Related work on service-level integrity uses cryptographic verification (Sun et al., 2025) and model-substitution detection (Cai et al., 2025), addressing inference integrity rather than scalable semantic auditing by crowdsourcing consensus.

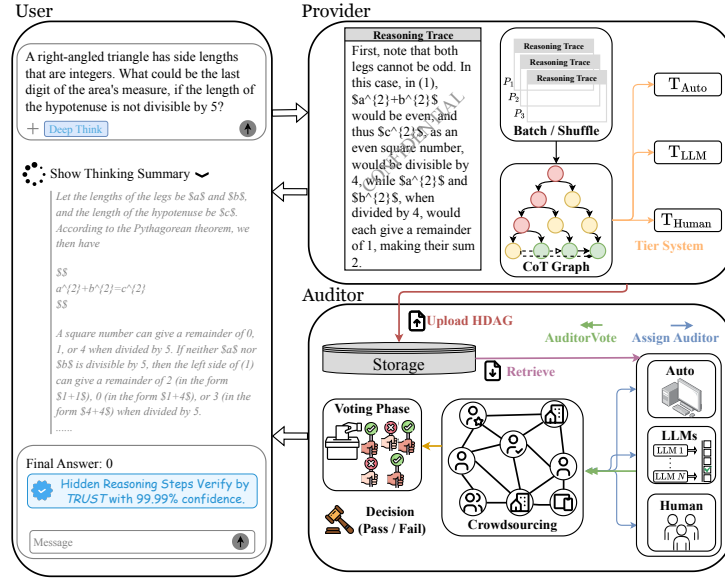


Figure 3: **Overview of TRUST.** The TRUST framework decomposes Chain-of-Thought reasoning traces into hierarchical directed acyclic graphs (DAGs) for modular verification across a three-tier auditor system. The process begins with a reasoning query (**left**) that generates intermediate reasoning steps, which are then decomposed into graph components and distributed across automated computers, LLM-based, and human auditors. Auditors verify reasoning segments independently, while reasoning trace decomposition limits each auditor’s exposure to proprietary model internals. The final verification result provides confidence guarantees for reasoning trace faithfulness and correctness for the end user.

Closest to our setting, RAudit (Chang & Geng, 2026) distributes individual reasoning steps to auditors without dependency context in order to keep the full trace hidden. We compare against a re-implementation of that protocol in Section 4.2 and find a +22.5 point trace-level gap, which isolates the contribution of contextual routing and consensus over blind per-step auditing.

Our work synthesizes these directions by introducing to our knowledge the first framework for crowdsourcing, confidentiality-aware semantic auditing of reasoning traces at scale. Due to space limitation, we provide more comprehensive related works in Section G.

3 Crowdsourcing Auditing for Large Reasoning Models

As illustrated in Figure 2, TRUST integrates both human or LLMs to audit *faithfulness*, *harmlessness*, and *logical consistency* of chain-of-thought (CoT) reasoning. By operating on intermediate traces rather than final outputs alone, TRUST enables earlier and more comprehensive detection of reasoning flaws. TRUST features the following key innovations:

❶ **Reasoning Trace Segmentation.** Traces from multiple providers are *batched* to anonymize source identity, then *segmented* into minimal auditable units stored so that each auditor sees only its assigned segment(s). ❷ **Auditing & Consensus.** Heterogeneous auditors (computational checkers, LLMs, humans) vote independently via a cryptographic *commit–reveal* protocol; segment-level quorums validate local steps and a trace-level aggregator combines weighted segment outcomes into the final decision. ❸ **Confidentiality-Aware Audit Infrastructure.** Audit metadata, assignments and outcomes are managed by the platform while raw reasoning content is compartmentalized and disclosed on a need-to-know basis, reducing exposure of full traces and supporting scalable human–AI collaboration.

3.1 Reasoning Trace Segmentation.

The core innovation of TRUST is the transformation of a linear reasoning trace into a **Hierarchical Directed Acyclic Graph (HDAG)** (see Figure 4 for an example). This representation abstracts reasoning into five levels: *Goal*, *Strategy*, *Tactic*, *Step*, and *Operation*. This hierarchy mirrors neural evidence integration (Sarafyazd & Jazayeri, 2019) and enables scalable, parallel verification.

HDAG Construction. TRUST constructs the HDAG through a three-stage pipeline. *Hierarchical Mapping* parses reasoning trace into nodes, where each node represents an auditable unit at one of the five abstraction levels. *Relational Extraction* defines edges to encode logical relationships, including structural links (DECOMPOSES_TO, REFINES) and logical dependencies (DEPENDS_ON, VALIDATES, CONTRADICTS). Finally, *Auditor Assignment* routes each node to an auditor type from {Computer, LLM, Human} based on its complexity; for example, deterministic tool calls are assigned to C, while high-level semantic logic is routed to H. Further implementation details and examples are provided in Sections H.1 and H.2.

The parser is a trusted component. The HDAG is itself an LLM output, so it is not trustless and we do not claim otherwise. Across 50 traces, independent parsers agree poorly on labels ($\kappa = 0.19$ abstraction level, $\kappa = 0.05$ auditor type; 79% of STEP labels differ), yet trace accuracy is unchanged at 72.0% under baseline, single-parser and consensus routing, and moves only 72.09% $\rightarrow$ 72.61% when 20% of routing decisions are corrupted: a parsing error matters only if it flips the verdict, and empirically it does not. We therefore claim TRUST removes the single trusted *auditor*, not every trusted authority (Section E).

3.2 Auditing & Consensus.

To handle the diversity of complexity and reasoning types across traces, TRUST routes each segment to the most appropriate auditor type from {Computer, LLM, Human}. This heterogeneous auditor pool improves accuracy by matching each segment to a specialized verifier and enhances robustness against adversarial actors because distinct auditor types have uncorrelated failure modes, limiting the damage any single compromised actor can cause. We formalize the consensus process across three hierarchical levels.

The computer tier is narrow. Exact arithmetic and symbolic-algebra checks fire on only 43 of 820 segments (5.2%) in our pool, at 81.5% raw accuracy; they remove cheap, high-confidence work from the LLM and human queues rather than carry the audit (Section F).

Seat Layer. Within a segment s , each of the $k_{t(s)}$ auditor seats votes independently, so individual errors do not systematically bias the aggregate verdict. Computer seats are modeled as noiseless ($\epsilon_C = 0$) since their verdicts come from deterministic formal checks, while LLM and human seats carry non-zero error rates ϵ_t . Human seats may additionally be adversarial with probability ρ_H , as humans can be deliberately incentivized to vote incorrectly; we set $\rho_C = \rho_L = 0$. We relax both idealizations empirically in Table 2c.

Segment Layer. We define the segment pass indicator $B_s = \mathbf{1}[\#\{\text{correct votes}\} \geq q_{t(s)}]$, where $q_t = \lceil \tau k_t \rceil$ is the quorum threshold for type t ; requiring a supermajority limits the influence of erroneous or adversarial votes. The exact pass probability p_t for a segment of type t with parameters $(k_t, \epsilon_t, \rho_t)$ is:

$$p_t = \Pr[B_s = 1] = \sum_{m=0}^{k_t} \binom{k_t}{m} \rho_t^m (1 - \rho_t)^{k_t - m} \left[\sum_{c=q_t}^{k_t - m} \binom{k_t - m}{c} (1 - \epsilon_t)^c \epsilon_t^{k_t - m - c} \right] \quad (3.1)$$

where m counts malicious seats that vote incorrectly and c counts correct votes cast by the remaining $k_t - m$ honest seats.

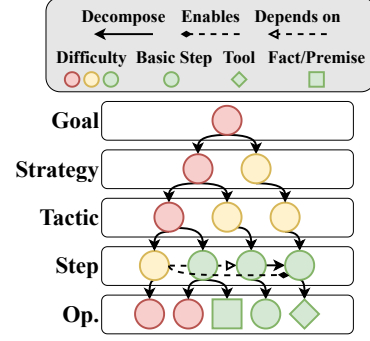


Figure 4: Example of an HDAG. Node color indicates the level of difficulty assigned to different types of auditors. Node shape denotes the relationships between nodes.

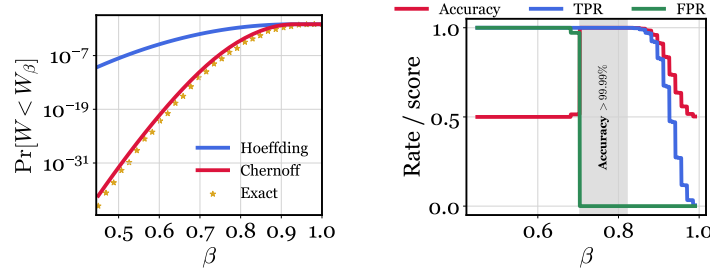


Figure 5: Parameters $\epsilon_C = 0$, $\epsilon_L = 0.05$, $\epsilon_H = 0.30$, $\rho_H = 0.1$. **(left)** Failure probability under the Hoeffding and Chernoff bounds vs. the exact solution of (3.2) over the trace-level quorum β . **(right)** TPR, FPR and accuracy over β ; the shaded band marks the quorum width achieving accuracy above 99.99%.

Table 1: Leakage under a colluding auditor coalition that pools every segment routed to its members; higher is worse. Token and segment recovery degrade quickly, trace *ordering* much more slowly.

Colluding seats	5%	10%	15%	20%	30%	50%
Token recall	32.6	49.6	61.6	73.6	86.1	98.3
Order recall	1.7	15.1	29.1	47.6	68.7	91.2
Segment coverage	32.4	49.7	61.9	73.7	86.3	98.2

Trace Layer. To aggregate segment outcomes into a trace-level decision we assign weights $w_{t(s)}$ reflecting each auditor type’s reliability, and define $W = \sum_{s=1}^S w_{t(s)} B_s$ and $W_\beta = \beta \sum_s w_{t(s)}$ with β the trace-level quorum threshold. We bound $\Pr[W < W_\beta]$ by the *tighter* of the Hoeffding and Chernoff inequalities, which dominate in different regimes:

$$\Pr[W < W_\beta] \leq \exp \left[\frac{-2(\mu_{\text{vote}} - W_\beta)^2}{\sigma_{\max}^2} \right] \wedge \min_{\lambda > 0} \exp \left(\lambda W_\beta + \sum_{s=1}^S \ln(p_s e^{-\lambda w_s} + (1 - p_s)) \right) \quad (3.2)$$

As shown in Figure 5, these bounds closely track the exact solution under representative parameters ($\epsilon_C = 0$, $\epsilon_L = 0.05$, $\epsilon_H = 0.30$, $\rho_H = 0.1$), so the tiered consensus mechanism keeps trace-level confidence guarantees robust even with individual auditor fallibility.

3.3 Confidentiality-Aware Audit Infrastructure.

The confidentiality property of TRUST is an intrinsic property of its structural design rather than an external enforcement mechanism. By decomposing the reasoning trace into an HDAG, TRUST compartmentalizes verification so that no single crowdsourced auditor can reconstruct the full reasoning chain. Each auditor receives only one or more atomic segments, without access to the broader context or final conclusion, because exposing the full trace to any individual worker risks intellectual property theft or model distillation. Audit outcomes are recorded as verification verdicts paired with opaque segment identifiers, providing an auditable log of completed work while keeping the underlying reasoning content compartmentalized.

Threat model and measured leakage. We assume a coalition of a c -fraction of auditor seats, honest-but-curious about verdicts, that pools every segment routed to any member and has no access to platform assignment metadata. We measure recovery along three axes: *token recall*, *order recall* (correctly recovered pairwise orderings) and *segment coverage*. At a 20% coalition token recall is 73.6%—segment *content* is not hidden—while order recall is 47.6%, so trace ordering and graph structure, which carry the reusable procedural knowledge a distillation attack wants, degrade far more slowly (Table 1). TRUST therefore raises the cost of reconstructing a usable trace; it is not a cryptographic guarantee, and leakage grows with coalition size. The layer is also a *configuration*, not a requirement: for open-weight models, publicly funded audits, and high-stakes domains where validation must dominate, it can be

disabled so credentialed experts receive full traces, leaving routing, consensus, escalation and incentives unchanged.

3.4 Economic Analysis

TRUST aligns auditor incentives through reputation, slashing and rewards. Each human auditor i keeps a reputation $r_i(t) \in [0, 1]$, updated after every segment as $r_i(t+1) = (1 - \gamma)r_i(t) + \gamma \mathbf{1}[\text{vote correct}]$; a low reputation raises the slashing probability on an incorrect vote, $p_{\text{slash}}(r) = p_{\min} + (p_{\max} - p_{\min})(1 - r)$, so persistent bad actors face escalating penalties. With per-segment payoff R for a correct vote, $-P$ for a slashed incorrect vote and 0 otherwise, an honest auditor’s expected payoff is $\mu_H(r) = (1 - \epsilon_H)R - \epsilon_H P p_{\text{slash}}(r)$. These dynamics are formalized below and confirmed empirically in Figure 9: honest auditors accumulate high reputation and steadily increasing profit, while malicious auditors and guessers are driven toward zero reputation and cumulative losses from repeated slashing.

Theorem 3.1 (Safety–Profitability (Informal; see Theorem I.4)). Fix horizon T , target failure ϵ_{target} , and $\delta \in (0, 1)$. If the **statistical dial** satisfies $\mu_{\text{vote}} - W_\beta \geq \sqrt{\frac{1}{2} \sigma_{\text{vote}}^2 \ln \frac{\lambda T}{\epsilon_{\text{target}}}}$ and the **economic dial** satisfies $\mu_{\min} := (1 - \epsilon_H)R - \epsilon_H P p_{\max} > 0$ and $p_{\min} \geq \frac{\delta}{1 - \alpha}$, then: **(a) Safety:** $\Pr[\text{fail in } [0, T]] \leq \epsilon_{\text{target}}$; **(b) Honest profit:** $\Pr[U_{\text{hon}}(T) \leq 0] \leq \exp(-c_1 T \mu_{\min}^2)$; **(c) Malicious loss:** $\Pr[U_{\text{mal}}(T) \geq 0] \leq \exp(-c_2 T (\delta P)^2)$ and $\mathbb{E}[U_{\text{mal}}(T)] \leq -\lambda T \delta P$.

The full derivation is given in Section I. For example, with $R = 6$, $P = 8$, $p_{\min} = 0.2$, $p_{\max} = 0.5$, $\epsilon_H = 0.30$ and $T = 1440$ segments, an honest seat earns $\mu_{\min} = 3.0$ per segment while a malicious seat loses $\mathbb{E}[X_{\text{mal}}] = -4.0$; over a 24-hour window Theorem I.4 gives failure probabilities of $\leq e^{-204} < 10^{-80}$ (honest unprofitable) and $\leq e^{-63.6} < 10^{-27}$ (malicious breaking even). We emphasize that this guarantee characterizes the *tolerance boundary* of the consensus rule under the modeled corruption process; it is a different quantity from the clean-data audit accuracy of Section 4, and neither implies the other.

4 Experimental Verification

4.1 Correctness and Faithfulness.

We use 200 samples from MMLU-Pro-CoT-Train (Lab, 2024), which annotates individual reasoning steps and final answers, letting us evaluate audits at both step and trace level. We compare TRUST against centralized approaches: (i) single-LLM auditors (DeepSeek-R1-8B, Qwen2.5-7B, Mistral-7B, GPT-OSS-20B, LLaMA-3B) and (ii) ensemble voting (majority, supermajority, weighted, unanimous), stress-testing robustness by flipping 5% to 20% of segment-level votes. At baseline TRUST achieves the highest accuracy (72.4%), above single auditors (DeepSeek-R1-8B, 67.7%) and ensembles (majority voting, 68.7%); under corruption all methods degrade but TRUST degrades most gracefully, staying above 63% at 20% corruption while ensembles drop below 61% and single auditors near 60%. The full per-method sweep is plotted in Figure 1 and tabulated in Table 7 (Section H).

Expanded, label-balanced evaluation. The 200-trace set is skewed toward correct traces (66%), so we choose the annotated pool to $n = 175$ traces whose added portion is close to balanced ($\approx 49\%$ correct) and recomputed every method on it (Table 2a). TRUST reaches 76.6% trace accuracy (bootstrap 95% CI [70.3, 82.9]); the paired gain over a single auditor is +18.0 points ([10.9, 25.9], $p < 0.001$), while the gain over the *best* ensemble is +2.1 points ([−10.4, 14.6]) and is **not** significant. On clean data a well-tuned ensemble is a strong baseline; heterogeneous consensus pays off under corruption, correlated bias and compromised auditors (Table 2c) and in escalation to humans (Section 4.4).

Auditor scale. To check these conclusions are not an artifact of small auditor models, we re-ran TRUST with three backbones spanning an order of magnitude in capability (Table 2b); the TRUST-over-single gap is positive for all three. GPT-5.1 and gpt-oss-20b agree on only

Table 2: **(a)** Label-balanced evaluation ($n = 175$; added traces $\approx 49\%$ correct vs. 66% originally), paired differences with bootstrap 95% CIs. **(b)** Stability across auditor backbones. **(c)** Adaptive top-weight corruption at a 20% budget, and violations of the independence / honest-seat assumptions (clean accuracy 72.1% in this harness). All numbers are trace-level audit accuracy (%) unless noted; **TRU.** = TRUST, **Sgl.** = best single auditor, **Ens.** = best ensemble, **Seg.** = segment-level accuracy.

(a) Label-Balanced eval ($n=175$)			(b) Auditor backbone				(c) Adaptive attack & assumptions			
	Acc.	Paired Δ [95% CI]	Backbone	TRU.	Sgl.	Seg.	Setting	TRU.	Sgl.	Ens.
TRUST	76.6	[70.3, 82.9]	gpt-oss-20b	76.6	58.6	76.2	Random @20%	63.4	52.7	62.1
vs. single	58.6	+18.0 [10.9, 25.9]***	qwen2.5-7b	60.2	59.4	76.1	Adaptive @20%	71.0	53.6	59.6
vs. best ens.	—	+2.1 [−10.4, 14.6] ^{n.s.}	GPT-5.1	73.3	70.0	78.2	Corr. bias $\rho=0.3$	70.9	—	—
							1 comp. seat	69.0	20.3	19.8
							2 comp. seats	29.5	20.3	19.8

*** $p < 0.001$ (one-sided paired test); n.s. = not significant.

53.5% of 316 compared segment verdicts, yet both land in the 73.3–76.6% trace band—segment-level disagreement cancels at the trace level, which is what the consensus layer is for.

Adaptive adversaries and broken assumptions. Random vote flipping is a weak threat model, and Section 3 assumes independent, non-adversarial seats; we test both (Table 2c). (i) *Targeted attack.* An attacker spending the same global budget on the highest-weight segments first reaches 71.0% against TRUST at a 20% budget, versus 63.4% under random flipping—targeting does not help. The reason is structural: flipping a trace verdict requires crossing the per-trace β -quorum, which typically needs several flips *inside the same trace*; a budget-constrained targeted attacker spreads its flips across high-weight segments of many traces, whereas random flipping occasionally clusters within one trace by chance. The same attack does hurt the 3-auditor ensemble (62.1% $\rightarrow$ 59.6%). (ii) *Correlated auditors.* Raising the intra-class correlation of auditor errors from 0 to 0.3 at a fixed mean error rate of 0.15 costs TRUST 1.2 points (72.1% $\rightarrow$ 70.9%), so partial violation of independence is tolerable. (iii) *Compromised seats.* With one compromised seat per segment TRUST retains 69.0% while single-auditor and ensemble baselines collapse to 20.3% and 19.8%; with two, TRUST also fails (29.5%). Under this configuration TRUST buys one seat of Byzantine tolerance, not unconditional robustness.

Output-Based Auditing is Insufficient. As shown in Table 3, output-only evaluation drops accuracy by 18% to 47% across all models compared to full-CoT auditing, confirming that correct final answers often mask flawed reasoning that only semantic auditing can detect.

Table 3: Baseline accuracy under Full-CoT vs. Output-Based evaluation.

	TRUST	R1-8B	Qwen-7B	Mistral-7B	R1-1.5B	OSS-20B	LLaMA-3B
Full-CoT	72.4	67.7	67.4	66.8	64.1	63.8	52.1
Output-Based	—	36.0	34.0	20.0	22.0	46.0	34.0

4.2 Comparison with Blind Per-Step Auditing.

The closest prior protocol for confidentiality-aware reasoning audit is RAudit (Chang & Geng, 2026), which distributes individual reasoning steps to auditors without dependency context. To isolate the contribution of *contextual routing plus consensus*, we re-implement RAudit-style blind step auditing and evaluate both on the 80 traces the two settings share. TRUST reaches 71.2% trace and 81.0% segment accuracy against 48.8% and

Table 4: TRUST vs. RAudit-style blind per-step auditing, on the 80 traces the two settings share. Accuracy in %.

Method	Trace	Seg.
TRUST	71.2	81.0
RAudit-style blind	48.8	54.8
Δ	+22.5	+26.2

54.8% (Table 4); the +22.5 point trace-level gap indicates that showing an auditor a step’s verified dependencies and implications, rather than the step in isolation, is where most of the benefit of decomposition lies (details in Section H).

4.3 Bias Mitigation.

Table 5: **Bias benchmark** (200 questions, 4 domains): segment-verdict accuracy in % on model-generated CoT, *not* comparable to the trace-level accuracies of Table 2. Bias score = (self-approval) – (other-approval); positive = favoritism, negative = criticism. “–” marks auditors that did not evaluate their own family’s traces, for which the score is undefined.

	Crowd. TRUST	Ensemble			Single LLM auditors						Sgl. avg.
		Super.	Major.	Weight.	GPT-OSS 20B	Qwen 7B	DS-R1 1.5B	Llama 3B	Mistral 7B	DS-R1 8B	
Acc. (%)	34.1	30.5	26.8	16.9	60.9	38.4	21.9	16.2	15.9	15.2	28.1
Bias	–	–	–	–	+44.4	–	–11.3	–	–	–	+5.5

Auditing systems are vulnerable to bias, where auditors may favor reasoning traces produced by their own model family or penalize outputs from competing models. This creates two common failure modes: (i) *self-favoritism*, where a model systematically approves its own reasoning, and (ii) *self-criticism*, where a model disproportionately rejects its own outputs.

TRUST is designed to mitigate such bias through three architectural features: (1) *Segment-level decomposition*, which breaks reasoning traces into atomic units; (2) *Multi-tier consensus*, combining human, LLM, and automated auditors; and (3) *Anonymous evaluation*, hiding the source model of each segment.

We construct a benchmark of 200 questions across four domains. For each, CoT traces from DeepSeek-R1-1.5B and GPT-OSS-20B are evaluated under three regimes: single auditors, ensemble auditors, and TRUST. In Table 5, single auditors vary in accuracy (15.2–60.9%) but show bias (avg. +5.5). Ensembles remove bias but perform poorly (16.9–30.5%). TRUST attains the best accuracy *among bias-free methods* (34.1%, vs. 16.9–30.5% for ensembles and a 28.1% single-auditor average). GPT-OSS-20B posts the highest raw number (60.9%) but carries a +44.4 self-favoritism score: much of that accuracy comes from approving its own traces, so it is not a usable auditor for a pool containing its own outputs. Absolute accuracies are low for every method here—model-generated traces with no step-level gold annotation—so the comparison of interest is the accuracy/bias trade-off within the table, not the level.

4.4 Human Experiment on Three-Tier Auditing with TRUST Variants

Table 6: Human Experiment on Three-Tier Auditing. Performance comparison across single LLM auditors, centralized human auditing, and TRUST variants. 30 Computer Science students audited 30 GSM8K math problems with CoT traces from DeepSeek-R1-8B and GPT-OSS-20B. Metrics are F1 score (higher better) and Brier score (lower better). Single LLMs perform poorly; a centralized human audit reaches F1=0.85. TRUST w/ HDAG is modestly higher on F1 (0.89) and substantially better calibrated (Brier 0.074 vs. 0.21). TRUST variants using random or fixed-length segmentation show degraded performance (F1=0.40), indicating that hierarchical decomposition, not segmentation per se, is what carries the gain.

Metric	Single LLM Auditors							TRUST Variants			
	DeepSeek R1-8B	Qwen2.5 7B	Mistral 7B	DeepSeek R1-1.5B	GPT-OSS 20B	LLaMA 3B	Human Audit	TRUST w/ HDAG	TRUST w/ Random Seg	TRUST w/ Fix Seg (Short)	TRUST w/ Fix Seg (Long)
F1	0.50	0.50	0.50	0.40	0.50	0.30	0.85	0.89	0.40	0.40	0.40
Brier Score	0.500	0.486	0.500	0.500	0.544	0.890	0.21	0.074	0.49	0.49	0.49

We recruited 30 computer science students as human auditors (example in Section H.3) and built a benchmark of 30 math problems from openai/gsm8k (Cobbe et al., 2021b), with CoT

traces from DeepSeek-R1-8B and GPT-OSS-20B. Auditors worked within TRUST alongside LLM and automated auditors; we compare against single-LLM auditors and a centralized human-only audit. Table 6 reports F1 and Brier scores. Single LLM auditors perform poorly when handed a whole trace to audit. Centralized human auditors achieve substantially stronger results ($F1 = 0.85$, against 0.50 for the best single LLM) and require approximately 1.4 minutes per trace on average (medium < 1 minute). They remain time-consuming, however, and susceptible to systematic bias and malicious behavior. In contrast, TRUST achieves a modestly higher F1 of 0.89 and a substantially lower Brier score of 0.074 against 0.21. The asymmetry matters: the accuracy gain over a human-only audit is small, whereas the calibration gain is large, and calibration is what makes an audit verdict usable downstream. We read this as feasibility and calibration evidence for the multi-tier design rather than a domain-general deployment study: it covers grade-school mathematics with student auditors, with no human study in the medical, legal or financial domains that motivate Section 1. For the TRUST variants (all using GPT-OSS-20B), random segmentation breaks the CoT at random between 2–6 sentences, fixed-short into two sentences per segment, and fixed-long into 5–7.

Human escalation is the high-stakes configuration. Since 76.6% is not sufficient for autonomous high-stakes use, we measure the accuracy–effort trade-off directly: escalating the top 10% most uncertain segments to the human tier raises trace accuracy to 74.42%, and 30% raises it to 77.52%. This is the intended deployment mode wherever human oversight exists—TRUST does not replace the human auditor, it decides which 10–30% of a trace deserves one.

5 Conclusion

We introduce TRUST, a crowdsourcing framework for auditing Large Reasoning Model traces that addresses robustness, scalability and IP-confidentiality. It integrates an HDAG decomposition of CoT reasoning, a multi-tier consensus mechanism routing verification to automated checkers, LLMs and human experts by complexity, and a confidentiality-aware infrastructure exposing only atomic segments per auditor. Its advantage is largest where centralized auditing is weakest—under random and adaptive corruption, correlated bias and compromised seats, where single auditors and ensembles fall to chance while TRUST retains 69.0%; on clean data its margin over a strong ensemble is small and not statistically significant. TRUST thus establishes crowdsourced semantic auditing as a pathway toward accountable reasoning-AI deployment—not as an autonomous certifier, but as a mechanism deciding which fraction of a trace deserves scarce human attention, while keeping the audit robust to any single auditor’s failure. The directions that follow most directly are an auditable parser and prompt-injection hardening (Section D).

Limitations. Two components of TRUST remain trusted: the HDAG parser is an LLM whose labels are unstable—independent parsers reach only $\kappa = 0.19$ on abstraction level and $\kappa = 0.05$ on auditor type, and 79% of STEP labels differ—so Section E establishes that the trace verdict is insensitive to that instability rather than that the graph is correct; and the confidentiality layer is weaker than a cryptographic guarantee, with token recall reaching 73.6% once a fifth of the auditor seats collude (Table 1). The empirical case is also narrower than the framing of Section 1. At 76.6% trace accuracy TRUST is not adequate for autonomous use in medicine, law or finance, where the appropriate configuration is human escalation (Section 4.4) with confidentiality disabled and credentialed experts given full traces; our human evidence is 30 computer science students on 30 GSM8K problems, which supports feasibility and calibration but says nothing about how domain experts would behave; and although we relax independence to $\rho = 0.3$ and test compromised seats, computer seats are still modeled as noiseless and LLM seats are never tested against prompt injection—the very attack we cite as motivation, and the most important experiment we have not run. Finally, Section C covers only the automated tier and the incentive dynamics of Section 3.4 are simulated, so strategic adaptation by real participants remains unobserved.

Ethics Statement

TRUST provides *confidentiality* for the provider’s proprietary reasoning logic. It does not protect end-user data: a trace containing personal information remains exposed to whichever auditor receives the segment carrying it, so deployments on user-facing traces require separate safeguards. The confidentiality layer is optional and must never take precedence over audit quality; Sections 3.3 and 5 state the configuration we would defend in high-stakes domains. Reputation and slashing run on a conventional database by default, since an on-chain instantiation would add material energy consumption and carbon footprint for a benefit that a permissioned append-only log already provides.

Acknowledgements

This research was partially supported by the XTX Markets AI for Math Fund, a Cisco Faculty Award, and an Amazon Research Award. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

- NIST AI. Artificial intelligence risk management framework (ai rmf 1.0). URL: <https://nvlpubs.nist.gov/nistpubs/ai/nist.ai>, pp. 100–1, 2023.
- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, pp. 610–623, 2021.
- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 17682–17690, 2024.
- Will Cai, Tianneng Shi, Xuandong Zhao, and Dawn Song. Are you getting what you pay for? auditing model substitution in llm apis. *arXiv preprint arXiv:2504.04715*, 2025.
- Nicholas Carlini, Florian Tramèr, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pp. 2633–2650, 2021.
- Miguel Castro, Barbara Liskov, et al. Practical byzantine fault tolerance. In *OsDI*, volume 99, pp. 173–186, 1999.
- Ilias Chalkidis, Abhik Jana, Dirk Hartung, Michael Bommarito, Ion Androutsopoulos, Daniel Martin Katz, and Nikolaos Aletras. Lexglue: A benchmark dataset for legal language understanding in english. *arXiv preprint arXiv:2110.00976*, 2021.
- Edward Y Chang and Longling Geng. Raudit: A blind auditing protocol for large language model reasoning. *arXiv preprint arXiv:2601.23133*, 2026.
- Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. Zkml: An optimizing system for ml inference in zero-knowledge proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pp. 560–574, 2024.

- Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *arXiv preprint arXiv:2211.12588*, 2022.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021a.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021b. URL <https://arxiv.org/abs/2110.14168>.
- EU COM. Laying down harmonised rules on artificial intelligence (artificial intelligence act) and amending certain union legislative acts. *Proposal for a regulation of the European parliament and of the council*, 2021.
- Victor Costan and Srinivas Devadas. Intel sgx explained. *Cryptology ePrint Archive*, 2016.
- Antonia Creswell, Murray Shanahan, and Irina Higgins. Selection-inference: Exploiting large language models for interpretable logical reasoning. *arXiv preprint arXiv:2205.09712*, 2022.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojuan Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shutong Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanbiao Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.

- Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. Pal: Program-aided language models. In *International Conference on Machine Learning*, pp. 10764–10799. PMLR, 2023.
- Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *Advances in Neural Information Processing Systems*, 30, 2017.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.
- Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *arXiv preprint arXiv:2212.10403*, 2022.
- Alon Jacovi, Yonatan Bitton, Bernd Bohnet, Jonathan Herzig, Or Honovich, Michael Tseng, Michael Collins, Roei Aharoni, and Mor Geva. A chain-of-thought is as strong as its weakest link: A benchmark for verifiers of reasoning chains. *arXiv preprint arXiv:2402.00559*, 2024.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Matt Gardner Johannes Welbl, Nelson F. Liu. Crowdsourcing multiple choice science questions. 2017.
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.
- Sourena Khanzadeh. Project ariadne: A structural causal framework for auditing faithfulness in llm agents. *arXiv preprint arXiv:2601.02314*, 2026.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.
- UW-Madison Lee Lab. Mmlu-pro-cot-train-labeled. <https://huggingface.co/datasets/UW-Madison-Lee-Lab/MMLU-Pro-CoT-Train-Labeled>, 2024. Accessed: 2025-09-24.
- Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. In *Concurrency: the works of leslie lamport*, pp. 203–226. 2019.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.
- Jixuan Leng, Cassandra A. Cohen, Zhixian Zhang, Chenyan Xiong, and William W. Cohen. Semi-structured llm reasoners can be rigorously audited, 2025. URL <https://arxiv.org/abs/2505.24217>.
- Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.
- Zhan Ling, Yunhao Fang, Xuanlin Li, Zhiao Huang, Mingu Lee, Roland Memisevic, and Hao Su. Deductive verification of chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 36:36407–36433, 2023.

- Tianyi Liu, Xiang Xie, and Yupeng Zhang. Zkcnn: Zero knowledge proofs for convolutional neural network predictions and accuracy. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2968–2985, 2021.
- Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4):1085–1115, 2024.
- Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.
- Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A Feder Cooper, Daphne Ippolito, Christopher A Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. Scalable extraction of training data from (production) language models. *arXiv preprint arXiv:2311.17035*, 2023.
- NVIDIA, :, Aarti Basant, Abhijit Khairnar, Abhijit Paithankar, Abhinav Khattar, Adithya Renduchintala, Aditya Malte, Akhiad Bercovich, Akshay Hazare, Alejandra Rico, Aleksander Ficek, Alex Kondratenko, Alex Shaposhnikov, Alexander Bukharin, Ali Taghibakhshi, Amelia Barton, Ameya Sunil Mahabaleshwarkar, Amy Shen, Andrew Tao, Ann Guan, Anna Shors, Anubhav Mandarwal, Arham Mehta, Arun Venkatesan, Ashton Sharabiani, Ashwath Aithal, Ashwin Poojary, Ayush Dattagupta, Balaram Buddharaju, Banghua Zhu, Barnaby Simkin, Bilal Kartal, Bitu Darvish Rouhani, Bobby Chen, Boris Ginsburg, Brandon Norick, Brian Yu, Bryan Catanzaro, Charles Wang, Charlie Truong, Chetan Mungekar, Chintan Patel, Chris Alexiuk, Christian Munley, Christopher Parisien, Dan Su, Daniel Afrimi, Daniel Korzekwa, Daniel Rohrer, Daria Gitman, David Mosallanezhad, Deepak Narayanan, Dima Rekesh, Dina Yared, Dmytro Pykhtar, Dong Ahn, Duncan Riach, Eileen Long, Elliott Ning, Eric Chung, Erick Galinkin, Evelina Bakhturina, Gargi Prasad, Gerald Shen, Haifeng Qian, Haim Elisha, Harsh Sharma, Hayley Ross, Helen Ngo, Herman Sahota, Hexin Wang, Hoo Chang Shin, Hua Huang, Iain Cunningham, Igor Gitman, Ivan Moshkov, Jaehun Jung, Jan Kautz, Jane Polak Scowcroft, Jared Casper, Jian Zhang, Jiaqi Zeng, Jimmy Zhang, Jinze Xue, Jocelyn Huang, Joey Conway, John Kamalu, Jonathan Cohen, Joseph Jennings, Julien Veron Vialard, Junkeun Yi, Jupiter Parmar, Kari Briski, Katherine Cheung, Katherine Luna, Keith Wyss, Keshav Santhanam, Kezhi Kong, Krzysztof Pawelec, Kumar Anik, Kunlun Li, Kushan Ahmadian, Lawrence McAfee, Laya Sleiman, Leon Derczynski, Luis Vega, Maer Rodrigues de Melo, Makesh Narsimhan Sreedhar, Marcin Chochowski, Mark Cai, Markus Kliegl, Marta Stepniewska-Dziubinska, Matvei Novikov, Mehrzad Samadi, Meredith Price, Meriem Boubdir, Michael Boone, Michael Evans, Michal Bien, Michal Zawalski, Miguel Martinez, Mike Chrzanowski, Mohammad Shoeybi, Mostofa Patwary, Namit Dhameja, Nave Assaf, Negar Habibi, Nidhi Bhatia, Nikki Pope, Nima Tajbakhsh, Nirmal Kumar Juluru, Oleg Rybakov, Oleksii Hrinchuk, Oleksii Kuchaiev, Oluwatobi Olabiyi, Pablo Ribalta, Padmavathy Subramanian, Parth Chadha, Pavlo Molchanov, Peter Dykas, Peter Jin, Piotr Bialecki, Piotr Januszewski, Pradeep Thalasta, Prashant Gaikwad, Prasoon Varshney, Pritam Gundecha, Przemek Tredak, Rabeeh Karimi Mahabadi, Rajen Patel, Ran El-Yaniv, Ranjit Rajan, Ria Cheruvu, Rima Shahbazyan, Ritika Borkar, Ritu Gala, Roger Waleffe, Ruoxi Zhang, Russell J. Hewett, Ryan Prenger, Sahil Jain, Samuel Krizan, Sanjeev Satheesh, Saori Kaji, Sarah Yurick, Saurav Muralidharan, Sean Narenthiran, Seonmyeong Bak, Sepehr Sameni, Seungju Han, Shanmugam Ramasamy, Shaona Ghosh, Sharath Turuvekere Sreenivas, Shelby Thomas, Shizhe Diao, Shreya Gopal, Shrimai Prabhumoye, Shubham Toshniwal, Shuoyang Ding, Siddharth Singh, Siddhartha Jain, Somshubra Majumdar, Soumye Singhal, Stefania Alborghetti, Syeda Nahida Akter, Terry Kong, Tim Moon, Tomasz Hliwiak, Tomer Asida, Tony Wang, Tugrul Konuk, Twinkle Vashishth, Tyler Poon, Udi Karpas, Vahid Noroozi, Venkat Srinivasan, Vijay Korthikanti, Vikram Fugro, Vineeth Kalluru, Vitaly Kurin, Vitaly Lavrukhin, Wasi Uddin Ahmad, Wei Du, Wonmin Byeon, Ximing Lu, Xin Dong, Yashaswi Karnati, Yejin Choi, Yian Zhang, Ying Lin, Yonggan Fu, Yoshi Suhara, Zhen Dong, Zhiyu Li, Zhongbo Zhu, and Zijia Chen. Nvidia nemotron nano 2: An accurate and efficient hybrid mamba-transformer reasoning model, 2025. URL <https://arxiv.org/abs/2508.14444>.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. Show your work: Scratchpads for intermediate computation with language models. 2021.

Forty-Two Countries Adopt New OECD. Principles on artificial intelligence, 2019.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL <https://arxiv.org/abs/2508.10925>.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In Gerardo Flores, George H Chen, Tom Pollard, Joyce C Ho, and Tristan Naumann (eds.), *Proceedings of the Conference on Health, Inference, and Learning*, volume 174 of *Proceedings of Machine Learning Research*, pp. 248–260. PMLR, 07–08 Apr 2022. URL <https://proceedings.mlr.press/v174/pal22a.html>.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. {LLMmap}: Fingerprinting for large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 299–318, 2025.

Zhizhi Peng, Taotao Wang, Chonghe Zhao, Guofu Liao, Zibin Lin, Yifeng Liu, Bin Cao, Long Shi, Qing Yang, and Shengli Zhang. A survey of zero-knowledge proof based verifiable machine learning. *arXiv preprint arXiv:2502.18535*, 2025.

Fábio Perez and Ian Ribeiro. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527*, 2022.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In Jian Su, Kevin Duh, and Xavier Carreras (eds.), *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 2383–2392, Austin, Texas, November 2016. Association for Computational Linguistics. doi: 10.18653/v1/D16-1264. URL <https://aclanthology.org/D16-1264>.

Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/IsPa*, volume 1, pp. 57–64. IEEE, 2015.

Morteza Sarafyazd and Mehrdad Jazayeri. Hierarchical reasoning by neural circuits in the frontal cortex. *Science*, 364(6441):eaav8911, 2019. doi: 10.1126/science.aav8911. URL <https://www.science.org/doi/abs/10.1126/science.aav8911>.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36: 68539–68551, 2023.

Christoph Schnabl, Daniel Hugenroth, Bill Marino, and Alastair R Beresford. Attestable audits: Verifiable ai safety benchmarks using trusted execution environments. *arXiv preprint arXiv:2506.23706*, 2025.

Zhouxing Shi, Yihan Wang, Fan Yin, Xiangning Chen, Kai-Wei Chang, and Cho-Jui Hsieh. Red teaming language model detectors with language models. *Transactions of the Association for Computational Linguistics*, 12:174–189, 2024.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023.

Tobin South, Alexander Camuto, Shrey Jain, Shayla Nguyen, Robert Mahari, Christian Paquin, Jason Morton, and Alex ‘Sandy’ Pentland. Verifiable evaluations of machine learning models using zksnarks. *arXiv preprint arXiv:2402.02675*, 2024.

- Guoheng Sun, Ziyao Wang, Bowei Tian, Meng Liu, Zheyu Shen, Shwai He, Yexiao He, Wanghao Ye, Yiting Wang, and Ang Li. Coin: Counting the invisible reasoning tokens in commercial opaque llm apis. *arXiv preprint arXiv:2505.13778*, 2025.
- Haochen Sun, Jason Li, and Hongyang Zhang. zkllm: Zero knowledge proofs for large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 4405–4419, 2024.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL <https://aclanthology.org/N19-1421>.
- Zhen Tan, Dawei Li, Song Wang, Alimohammad Beigi, Bohan Jiang, Amrita Bhattacharjee, Mansoor Karami, Jundong Li, Lu Cheng, and Huan Liu. Large language models for data annotation: A survey. *arXiv preprint arXiv:2402.13446*, 2024.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965, 2023.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.
- UW-Madison-Lee-Lab. Mmlu-pro-cot-train-labeled. <https://huggingface.co/datasets/UW-Madison-Lee-Lab/MMLU-Pro-CoT-Train-Labeled>, 2024.
- Neng Wang, Hongyang Yang, and Christina Dan Wang. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*, 2023.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- Meghan Wondra. Auditing large language models. *Available at SSRN 5675847*, 2025.
- Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL <https://arxiv.org/abs/2505.09388>.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023a.

- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.
- Maofan Yin, Dahlia Malkhi, Michael K Reiter, Guy Golan Gueta, and Ittai Abraham. Hotstuff: Bft consensus with linearity and responsiveness. In *Proceedings of the 2019 ACM symposium on principles of distributed computing*, pp. 347–356, 2019.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL <https://arxiv.org/abs/2203.14465>, 2203, 2022.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625*, 2022.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

Appendix

A	Impact Statement	19
B	The Use of Large Language Models (LLMs)	19
C	Latency Analysis	19
D	Future Work	20
E	HDAG Parser Quality and Verdict Robustness	20
F	Computer-Auditor Tier	20
G	More Related Works	21
G.1	Chain-of-Thought Reasoning	21
G.2	Auditing and Evaluation of LLM Reasoning	21
G.3	Decentralized Consensus and Privacy-Preserving Verification	22
H	Experimental Details	22
H.1	Implementation Details for Constructing HDAG	22
H.1.1	Step 1: Abstract Level Identification.	22
H.1.2	Step 2: Segment Within Each Abstraction Level.	23
H.1.3	Step 3: Extract Relationships.	23
H.1.4	Step 4: Assign Auditor Types.	24
H.1.5	Step 5: Refine and HDAG Construction.	24
H.2	HDAG Example	25
H.3	Human-Friendly Audit Example	26
H.4	Datasets	27
I	Theoretical Results	27
I.1	Consensus: Seat, Segment, and Trace Layers	27
I.2	Economic Layer: Reputation, Rewards, and Slashing	28
J	Motivating Example: Why Semantic Auditing Is Necessary	31
J.1	Three Critical Gaps That TRUST Addresses	33

A Impact Statement

This paper advances the auditing and verification of LLM reasoning, with the primary goal of improving transparency, accountability, and safety of AI systems deployed in high-stakes domains such as healthcare, law, and finance. The TRUST framework contributes to the trustworthiness of reasoning AI by enabling scalable, confidentiality-aware semantic audits that support emerging regulatory requirements for explainable AI oversight. We stress that TRUST is an oversight aid rather than a certification mechanism: at the accuracies reported here it is not a substitute for qualified human review in safety-critical decisions, and presenting a TRUST verdict to end users as a guarantee of correctness would overstate what the system establishes.

B The Use of Large Language Models (LLMs)

For *manuscript preparation*, this paper used Large Language Models exclusively as language-polishing tools: general proofreading and writing assistance, analogous to grammar checkers. They did not originate research ideas, generate scientific content, produce data or plots, or perform any evaluation reported here. For *the research itself*, LLMs are components and subjects of the system rather than authoring tools: they generate the reasoning traces under audit, construct the HDAG (Section H.1), and serve as one of the three auditor tiers. All such uses are described in Sections 3 and 4 and in the appendices, and the prompts are released with the code.

C Latency Analysis

In Figure 6, we present a latency analysis decomposing TRUST’s overhead into constituent components and comparing it against centralized baselines.

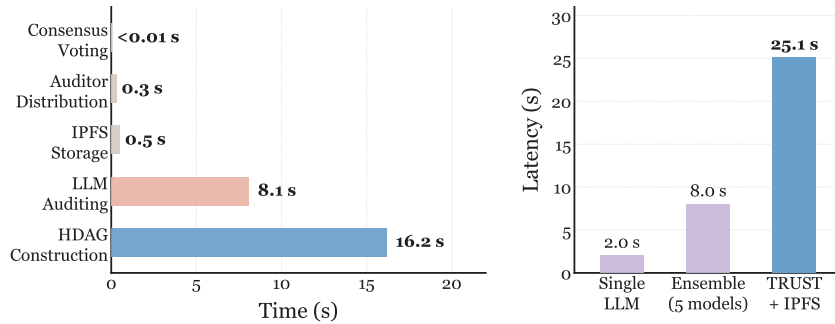


Figure 6: End-to-end latency analysis of TRUST. **(Left)** Breakdown of latency components for an average of 100 math problems. **(Right)** Comparison of total audit latency against centralized baselines.

Scope and Limitation. The latency figures reported here cover only the automated pipeline components: HDAG construction, segment routing, LLM auditing, and consensus aggregation. They do *not* include human auditor response time. In practice, when TRUST routes high-complexity segments to human auditors, end-to-end latency is bottlenecked by human response time rather than the automated components shown here. We therefore interpret Figure 6 as characterizing the latency of the automated tier only. Incorporating realistic human latency distributions into end-to-end latency modeling is an important direction for future work.

D Future Work

We identify four directions for extending TRUST. First, we will develop richer HDAG decomposition methods and adaptive auditor assignment strategies that leverage task-specific expertise, and in particular a parser whose output is itself auditable rather than merely verdict-stable. Second, we will harden the LLM auditor tier against prompt injection and against auditors that optimize for agreement with the consensus rather than for correctness; both are attacks that our current corruption model does not capture. Third, we plan to extend the framework to multi-agent settings, including cross-model reasoning consistency verification. Finally, the reputation, slashing and reward accounting described in Section 3.4 runs on a conventional database in our default configuration, and we treat that as the deployment target. A distributed ledger is one *optional* instantiation—segment commitments map to on-chain hashes, quorum voting to contract logic, reputation-weighted slashing to token incentives—but it is not required by the method, and we do not recommend it by default: it adds material energy consumption and carbon footprint in exchange for public verifiability of the audit log, a property a permissioned append-only log already provides at far lower environmental cost. Should such a deployment be pursued, benchmarking its cost-effectiveness should include an explicit energy and carbon accounting alongside latency and throughput.

E HDAG Parser Quality and Verdict Robustness

The HDAG is constructed by an LLM, so the graph on which every downstream audit operates is itself a model output. This section reports what we can and cannot say about it.

Protocol. We parse the same 50 reasoning traces with independent parsers spanning two model families and scales (an open-weight 20B model and GPT-5.1), under three routing modes: *baseline* (the routing shipped with the main experiments), *single-parser* (all segments routed by one parser’s assignment), and *consensus* (assignment by agreement across parsers). We additionally rerun a single parser three times at fixed decoding settings, and separately corrupt a fraction of routing decisions at random.

Parsers disagree at the label level. Inter-parser agreement is low: Cohen’s $\kappa = 0.19$ for the abstraction level assigned to a segment and $\kappa = 0.05$ for the auditor type, and 79% of STEP-level labels differ between parsers. We do not claim the parser is accurate, and we have no gold HDAG annotation against which accuracy could be defined.

The trace verdict does not move. Despite that disagreement, trace-level audit accuracy is 72.0% under all three routing modes, three reruns of a single parser produce zero variance in the trace verdict, and randomly corrupting 20% of routing decisions changes trace accuracy only from 72.09% to 72.61% (the direction of the change is within noise). The explanation is that a parsing error is consequential only if it changes which side of the per-trace β -quorum a trace falls on; most re-labelings move a segment between auditor tiers that reach the same verdict, and the quorum absorbs the rest.

Scope of the claim. These results support the claim that TRUST’s *output* is robust to parser variation across model scales and families. They do not support a claim that the framework is trustless: a parser that systematically mis-segments in a correlated way—for instance, one adversarially prompted to hide a particular class of error inside a single high-level node—is outside the variation we sampled here. We accordingly say “no single trusted *auditor*” rather than “no central trusted authority” throughout.

F Computer-Auditor Tier

The computer tier applies deterministic checks—exact arithmetic and symbolic algebra verification—to segments whose content admits them. It is deliberately narrow. On the

current trace pool it fires on 43 of 820 segments (5.2%), and its raw accuracy against the existing segment labels is 81.5%; the residual error is dominated by segments where the extracted expression is a faithful check of a mis-stated quantity, which a formal checker cannot detect. Semantic correctness is therefore established by the LLM and human tiers, and the computer tier’s role is to remove cheap, high-confidence work from their queue rather than to carry the audit.

G More Related Works

G.1 Chain-of-Thought Reasoning

Exposing intermediate reasoning steps, popularized by Chain-of-Thought (CoT) prompting (Wei et al., 2022), has become a cornerstone for enhancing the capabilities of Large Language Models (LLMs) (Kojima et al., 2022; Wang et al., 2022; Nye et al., 2021; Zhou et al., 2022; Zelikman et al., 2022). However, this paradigm has rapidly evolved, and the resulting complexity and length of reasoning traces present a critical, unsolved challenge for verification (Lightman et al., 2023; Jacovi et al., 2024; Ling et al., 2023; Chen et al., 2025). Initial research focused on eliciting reasoning and moving beyond simple linear chains to more structured representations (Creswell et al., 2022; Zhou et al., 2022; Besta et al., 2024; Chen et al., 2022). For instance, Tree-of-Thoughts (ToT) generalized CoT to a tree search, enabling explicit exploration and backtracking within the reasoning process (Yao et al., 2023a). Concurrently, a parallel line of work enabled models to offload complex calculations and external queries to tools, separating logical deduction from information retrieval (Schick et al., 2023; Yao et al., 2023b; Gao et al., 2023; Nakano et al., 2021). More recent efforts have focused on Large Reasoning Models (LRMs) that treat reasoning as a primary objective, allocating substantial computational resources to the process (Jaech et al., 2024; Guo et al., 2025; NVIDIA et al., 2025). These models are trained with large-scale reinforcement learning from process-level feedback and employ verifiers to guide multi-path search during inference (Huang & Chang, 2022). Despite these advances in generating sophisticated reasoning, the mechanisms for *auditing* these processes in a scalable, privacy-preserving, and decentralized manner have remained largely unexplored (Peng et al., 2025). In this paper, we address this gap by introducing a framework that decomposes complex reasoning traces into auditable graphs (HDAGs) and leverages a decentralized network for verification, making it suitable for the proprietary and intricate nature of modern LRMs.

G.2 Auditing and Evaluation of LLM Reasoning

As the complexity of reasoning in LLMs increases, methods for auditing its quality, faithfulness, and safety have become a critical area of research (Liang et al., 2022; Lanham et al., 2023; Wondra, 2025). A broader treatment of LLM auditing methodology situates this challenge across model, application, and governance layers (Mökander et al., 2024), providing useful context for understanding where reasoning-trace auditing sits within the wider auditing landscape. An initial and widely adopted approach relies on a centralized “LLM-as-a-judge,” which, while scalable, is known to suffer from inherent biases and constitutes a single point of failure (Zheng et al., 2023). A significant advancement was the shift from auditing final outcomes to verifying the reasoning *process* itself, primarily through Process Reward Models (PRMs) that provide step-by-step supervision (Lightman et al., 2023; Uesato et al., 2022; Khalifa et al., 2025). The limitations of this approach were subsequently highlighted by the discovery of unfaithful reasoning, where models generate plausible-looking steps that do not reflect their true decision-making process (Turpin et al., 2023); recent work on structural causal frameworks further sharpens this distinction by providing a causal account of faithfulness in LLM agents (Khanzadeh, 2026), complementing consensus-based semantic consistency checks with a principled notion of faithfulness auditing. More recently, research has expanded to address service-level integrity in opaque commercial settings, including work on detecting model substitution and using cryptographic methods to verify inference integrity (Shi et al., 2024; Cai et al., 2025; Pasquini et al., 2025; South et al., 2024). For example, CoIn audits “invisible” reasoning tokens via hash trees over embeddings, ensuring providers do not inflate billing without revealing proprietary content (Sun et al.,

2025). Closest to our privacy goals, RAudit proposes a blind auditing protocol for LLM reasoning that conceals trace content from auditors (Chang & Geng, 2026); while RAudit focuses on the privacy protocol design, TRUST contributes complementary machinery: hierarchical trace decomposition, heterogeneous crowdsourced consensus, and theoretically-grounded economic incentives that together ensure both scalability and auditor integrity. None of these works offer a unified framework for scalable *semantic* auditing that combines crowdsourced consensus to mitigate bias with economic guarantees to secure the auditor network. TRUST synthesizes these needs by performing process-level semantic auditing on partial traces, using a multi-tier crowdsourced auditor pool for robustness, and grounding network security in formal economic guarantees.

G.3 Decentralized Consensus and Privacy-Preserving Verification

Building systems that ensure integrity and privacy without a central trusted party has a long-standing foundation in decentralized consensus, cryptography, and confidential computing (Castro et al., 1999; Sabt et al., 2015; Costan & Devadas, 2016). These technologies provide the essential primitives for building auditable systems for proprietary models, where model internals and user data must remain confidential. Foundational work established Byzantine Fault Tolerant (BFT) consensus for robust agreement and public ledgers for tamper-evident records (Castro et al., 1999; Wood et al., 2014; Yin et al., 2019), while Trusted Execution Environments (TEEs) provided hardware-based confidential compute (Sabt et al., 2015). A prominent line of recent research focuses on applying these primitives directly to language models, particularly using Zero-Knowledge Proofs (ZKPs) to create verifiable attestations of model inference without revealing the model’s weights (Chen et al., 2024; Liu et al., 2021; Ghodsi et al., 2017). Specifically, zkLLM (Sun et al., 2024) introduced specialized cryptographic techniques to efficiently prove the correctness of attention mechanisms, demonstrating that even complex transformer computations can be cryptographically verified. Another approach uses TEEs to run auditable safety benchmarks on proprietary models, publishing hardware-signed attestations to a public log to prove that a specific model version passed certain tests (Schnabl et al., 2025). While this work is crucial for verifying *that* a specific computation was performed correctly by a specific model, it does not address the orthogonal problem of verifying the *semantic quality, faithfulness, and safety* of the reasoning trace through a scalable, human-in-the-loop consensus process. Our work provides a novel synthesis by using a public ledger and consensus not just for recording outcomes, but for orchestrating a decentralized, multi-tier network of auditors and managing economic incentives, creating the first system designed for end-to-end semantic auditing of model reasoning.

H Experimental Details

Table 7 tabulates the full per-method corruption sweep plotted in Figure 1.

H.1 Implementation Details for Constructing HDAG

This section provides the LLMs instruction of our auto data processing framework for creating the HDAG.

H.1.1 Step 1: Abstract Level Identification.

Below are the instructions for identifying different abstraction levels.

```
You are an abstraction level identification agent. Decompose a reasoning trace into:
- GOAL: Problem statement / objective
- STRATEGY: High-level approach
- TACTIC: Specific methods / key insights
- STEP: Individual reasoning moves
- OPERATION: Atomic calculations
```

```
Extract EXACT text spans. Preserve all math notation and line breaks.
```

Table 7: Performance comparison of TRUST (crowdsourcing) vs. centralized approaches across corruption rates. Best is **bold**, second-best is underlined. Metric: trace-level binary audit accuracy (%) on $n = 200$ annotated MMLU-Pro-CoT-Train traces under random segment-verdict corruption; mean $\pm$ s.d. over runs. Table 2a reports the same protocol on the expanded, label-balanced set.

Category	Method	Baseline	5% Corr.	10% Corr.	15% Corr.	20% Corr.
Crowdsourcing	TRUST	72.4$\pm$0.0	70.2$\pm$1.8	67.5$\pm$2.6	65.8$\pm$3.0	63.4$\pm$3.8
	<i>Ensemble Models</i>					
	Majority Voting	68.7 $\pm$ 0.0	66.8 $\pm$ 0.4	64.9 $\pm$ 0.7	63.4 $\pm$ 0.9	61.4 $\pm$ 1.1
	Supermajority	68.7 $\pm$ 0.0	66.8 $\pm$ 0.5	65.0 $\pm$ 0.7	63.2 $\pm$ 0.9	61.2 $\pm$ 0.9
	Weighted Voting	68.1 $\pm$ 0.0	66.4 $\pm$ 0.6	64.5 $\pm$ 0.7	62.7 $\pm$ 1.1	60.9 $\pm$ 0.9
Centralized	Unanimous	45.6 $\pm$ 0.0	46.1 $\pm$ 0.6	46.5 $\pm$ 0.9	46.8 $\pm$ 1.1	47.4 $\pm$ 1.0
	<i>Single Models</i>					
	DeepSeek-R1-8B	67.7 $\pm$ 0.0	65.9 $\pm$ 0.5	64.2 $\pm$ 0.7	62.5 $\pm$ 0.8	60.5 $\pm$ 1.0
	Qwen2.5-7B	67.4 $\pm$ 0.0	65.7 $\pm$ 0.6	64.1 $\pm$ 0.7	62.1 $\pm$ 0.9	60.5 $\pm$ 1.0
	Mistral-7B	66.8 $\pm$ 0.0	65.2 $\pm$ 0.6	63.6 $\pm$ 0.8	61.8 $\pm$ 1.1	60.1 $\pm$ 1.1
	DeepSeek-R1-1.5B	64.1 $\pm$ 0.0	62.9 $\pm$ 0.5	61.2 $\pm$ 0.8	59.7 $\pm$ 1.1	58.5 $\pm$ 0.8
	GPT-OSS-20B	63.8 $\pm$ 0.0	62.5 $\pm$ 0.6	60.9 $\pm$ 0.7	59.7 $\pm$ 1.1	58.4 $\pm$ 1.0
	LLaMA-3B	52.1 $\pm$ 0.0	51.9 $\pm$ 0.6	51.7 $\pm$ 0.7	51.4 $\pm$ 1.0	51.3 $\pm$ 1.1

Input: ## Problem Statement / ## Thinking Process / ## Final Output
Output: ## [LEVEL] blocks + # Block Structure JSON summary

H.1.2 Step 2: Segment Within Each Abstraction Level.

Below are the instructions for segmenting within each abstraction level.

You are a reasoning segmentation agent. Break each abstraction block into verifiable $\hookrightarrow$ segments.

Rules per level:

- GOAL: one complete statement per segment
- STRATEGY: one complete approach per segment
- TACTIC: break at technique boundaries
- STEP: one inference per segment
- OPERATION: one calculation per segment

Each segment includes:

- segment_id (G1, S1, T1, ST1, O1, ...)
- content (exact text)
- type
- summary (3-5 words)
- complexity (Low / Medium / High)

H.1.3 Step 3: Extract Relationships.

Below are the instructions for extracting relationships.

You are a semantic relationship analysis agent. Identify edges between segments:

- decomposes_to: higher abstraction level breaks into lower
- depends_on: logical prerequisite (A must hold for B to be valid)
- enables: A makes B possible
- validates: A confirms or verifies B
- exemplifies: A is a concrete instance of B

- refines: A adds detail or precision to B

Output: Hierarchical / Lateral / Support relationship lists + JSON summary with fields: from_segment, to_segment, relationship_type, strength, explanation.

H.1.4 Step 4: Assign Auditor Types.

Below are the instructions for assigning auditor types.

You are an auditor assignment agent. Assign to each segment:

- T_Auto: math calculations, formal logic, deterministic checks
- T_LLM: semantic coherence, factual accuracy, strategy assessment
- T_Human: ambiguous interpretation, domain expertise, high-stakes validation

Specify primary + secondary auditors with justification per segment.
Output: per-segment assignments + auditor distribution table + multi-auditor JSON.

H.1.5 Step 5: Refine and HDAG Construction.

Below are the instructions for refining and constructing the HDAG.

You are an HDAG construction agent. Synthesize all prior analyses into a final graph.

Validate:

- Acyclicity: no circular dependencies
- Connectivity: all nodes reachable from root
- Completeness: all reasoning preserved
- Auditability: each node has clear verification criteria

Output: finalized nodes + edges (with confidence scores) + verification flow
+ complete HDAG JSON with fields: nodes, edges, metadata.

H.2 HDAG Example

Problem: Alec is running for Class President. He thinks that if he can get
 ↳ three-quarters of the class to vote for him then there is no chance anyone else
 ↳ can beat him. Half of the class have already said they will vote for him but out
 ↳ of the remaining students, only 5 have said they are thinking about voting for
 ↳ him. He surveys the students who are thinking about voting for someone else, and
 ↳ changes his flyers to reflect the issues these students are concerned about. This
 ↳ results in a fifth of these students saying they'll vote for him. If Alec's class
 ↳ has 60 students and everyone who said they will vote for him does so, how many
 ↳ more votes does Alec need to reach his goal number of votes?

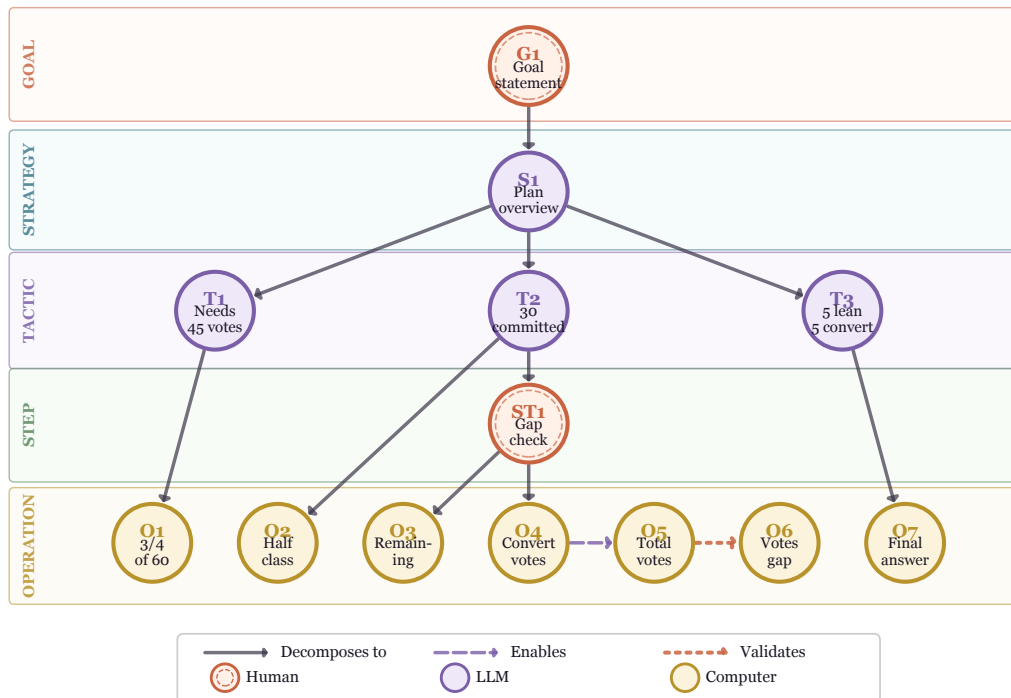


Figure 7: HDAG Example from Deepseek-R1-8B CoT.

H.3 Human-Friendly Audit Example

To illustrate how human auditors interact with TRUST, we walk through a representative audit case. Each segment is presented with four components: the **prompt context** (original problem), **dependencies** (verified prior steps), **current reasoning** (the step under evaluation), and **logical implications** (consequences that follow).

Prompt Context. Martha needs 4 cups of berries and 2 cups of heavy cream per quart of ice cream. She wants 1 quart each of strawberry and raspberry. Strawberries cost \$3.00 per 2-cup package, raspberries \$5.00 per 2-cup package, and heavy cream \$4.00 per 4-cup container. What is the total cost?

Verified Dependency (S1). Each quart requires 4 cups of berries and 2 cups of cream; she is making two quarts total.

Current Reasoning (T1). For two quarts combined: $2 \times 4 = 8$ cups of berries and $2 \times 2 = 4$ cups of cream are needed.

Logical Implication (T3). Eight cups of berries at 2 cups per package requires $8 \div 2 = 4$ packages total.

Self-Verification (ST1). Two quarts require 8 cups berries and 4 cups cream. Confirmed correct.

Human auditors evaluate each segment for logical consistency, mathematical accuracy, completeness, and clarity.

Ground Truth: Pass ✓. The step correctly doubles per-quart requirements, performs accurate arithmetic, and is confirmed by self-verification.

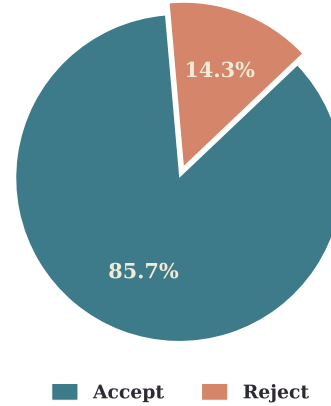


Figure 8: Human Audit Results.

H.4 Datasets

We evaluate TRUST on two datasets spanning multiple domains following recent surveys [Tan et al. \(2024\)](#).

Primary Dataset. MMLU-Pro-CoT-Train ([UW-Madison-Lee-Lab, 2024](#)) provides 200 samples with ground-truth annotations for individual reasoning steps and final answers across technical domains (engineering, mathematics, physics, chemistry, biology, and computer science), enabling evaluation at both step and trace levels.

Multi-Domain Bias Evaluation Dataset. We curate 300 questions (50 per domain) from established HuggingFace datasets across six domains: **Medical** (medmcqa, ([Pal et al., 2022](#))) covering clinical diagnosis and treatment; **Mathematics** (gsm8k, ([Cobbe et al., 2021a](#))) covering multi-step arithmetic and algebra; **Science** (sciq, ([Johannes Welbl, 2017](#))) covering physics, chemistry, and biology; **Common Sense** (commonsense_qa, ([Talmor et al., 2019](#))) covering everyday reasoning and social knowledge; **Humanities** (squad, ([Rajpurkar et al., 2016](#))) covering reading comprehension and factual inference; and a **Human Evaluation Subset** of 10 math problems from gsm8k used in human-in-the-loop experiments with 30 PhD-level auditors. All samples undergo ground-truth verification, format consistency checking, and deduplication.

I Theoretical Results

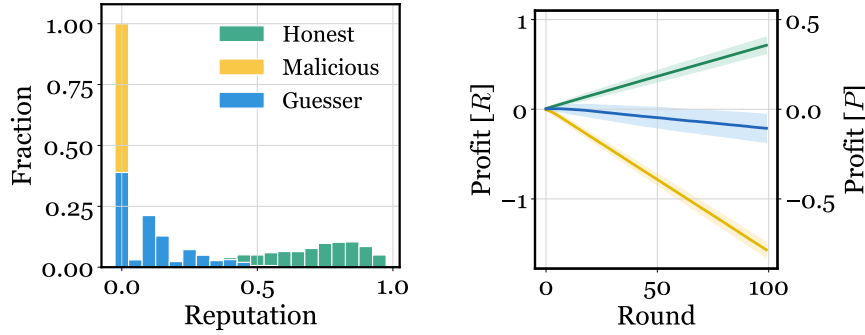


Figure 9: Simulation of the incentive layer of Section 3.4; same parameters as Figure 5. (left) Reputation separates honest auditors from malicious and random-guessing seats over rounds. (right) Profit curves diverge: honest auditors earn increasing rewards while malicious seats accumulate losses from repeated slashing.

I.1 Consensus: Seat, Segment, and Trace Layers

We provide the theoretical foundations of TRUST by analyzing statistical safety against malicious auditors and economic incentive/disincentive guarantees across three hierarchical layers.

Seat Layer. Within a segment s , each of the $k_{t(s)}$ seats votes independently, so that individual errors do not systematically bias the aggregate verdict. Computer seats are noiseless; LLM and human seats carry error rate $\epsilon_t > 0$; and human seats may be adversarial with probability ρ_H , since humans—unlike computers or LLMs—can be deliberately incentivized to vote incorrectly.

Segment Layer. To decide whether a segment passes, we define the segment pass indicator $B_s = \mathbf{1}[\#\{\text{correct votes}\} \geq q_{t(s)}]$, where $q_t = \lceil \tau k_t \rceil$ is the quorum threshold for type t . Requiring a supermajority limits the influence of erroneous or adversarial votes. The exact pass probability $p_s = \Pr[B_s = 1]$ is derived in Theorem I.1 below.

Trace Layer. To aggregate segment outcomes into a single trace-level decision, we assign reliability weights $w_{t(s)}$ to each segment type and define the weighted pass sum $W = \sum_{s=1}^S w_{t(s)} B_s$ and the trace-level quorum $W_\beta = \beta \sum_s w_{t(s)}$, where $\beta \in (0, 1)$ is the pass threshold. The trace fails if $W < W_\beta$. We bound this failure probability using the tighter of Hoeffding and Chernoff inequalities, as formalized in Theorem I.2.

We now derive the exact segment pass probability.

Lemma I.1 (Exact Pass Probability). For a segment of type t with parameters $(k_t, \epsilon_t, \rho_t)$, where $\rho_C = \rho_L = 0$, the pass probability is

$$p_t = \Pr[B_s = 1] = \sum_{m=0}^{k_t} \binom{k_t}{m} \rho_t^m (1 - \rho_t)^{k_t-m} \sum_{c=q_t}^{k_t-m} \binom{k_t-m}{c} (1 - \epsilon_t)^c \epsilon_t^{k_t-m-c}, \quad (\text{I.1})$$

where m counts adversarial seats that vote incorrectly and c counts correct votes among the remaining $k_t - m$ honest seats.

Proof. We first draw $m \sim \text{Binomial}(k_t, \rho_t)$ adversarial seats, all of whom vote incorrectly. Among the remaining $k_t - m$ honest seats, we then count $c \sim \text{Binomial}(k_t - m, 1 - \epsilon_t)$ correct votes. The segment passes iff $c \geq q_t$; summing over all m gives the result. $\square$

Trace-Level Moments. Since $B_s \sim \text{Bernoulli}(p_s)$ are independent with weights $w_s = w_{t(s)}$, the mean and variance of W are

$$\mu_{\text{vote}} = \mathbb{E}[W] = \sum_{s=1}^S w_s p_s, \quad (\text{I.2})$$

$$\sigma_{\text{vote}}^2 = \text{Var}(W) = \sum_{s=1}^S w_s^2 p_s (1 - p_s) \leq \sum_{s=1}^S w_s^2 =: \sigma_{\text{max}}^2. \quad (\text{I.3})$$

We use σ_{max}^2 as a conservative variance upper bound throughout.

Proposition I.2 (Hoeffding–Chernoff Bound). For trace-level quorum $W_\beta = \beta \sum_s w_s$,

$$\Pr[W < W_\beta] \leq \underbrace{\exp\left[-2(\mu_{\text{vote}} - W_\beta)^2 / \sigma_{\text{max}}^2\right]}_{\text{Hoeffding}} \wedge \underbrace{\min_{\lambda > 0} \exp\left(\lambda W_\beta + \sum_{s=1}^S \ln(p_s e^{-\lambda w_s} + 1 - p_s)\right)}_{\text{Chernoff}}. \quad (\text{I.4})$$

Taking the minimum of the two bounds is tighter because Hoeffding dominates when β is near μ_{vote} while Chernoff dominates in the tails.

Proof. For the Hoeffding bound, each summand $X_s := w_s B_s$ satisfies $0 \leq X_s \leq w_s$. Applying Hoeffding's inequality to $\sum_s (X_s - \mathbb{E}X_s)$ with range sum σ_{max}^2 yields the first term. For the Chernoff bound, for any $\lambda > 0$,

$$\Pr[W < W_\beta] \leq e^{\lambda W_\beta} \mathbb{E}[e^{-\lambda W}] = e^{\lambda W_\beta} \prod_{s=1}^S (p_s e^{-\lambda w_s} + (1 - p_s)),$$

using independence of the B_s . Minimizing over $\lambda > 0$ gives the second term. $\square$

I.2 Economic Layer: Reputation, Rewards, and Slashing

Notation. We collect the key symbols used throughout this section in Table 8. For convenience: R is the per-segment reward for a correct vote; P is the penalty when slashed; ϵ_H is the honest human error rate; $\delta \in (0, 1)$ is a design constant controlling the minimum guaranteed loss per malicious seat; λ is the segment arrival rate (segments per unit time); T is the time horizon; and $N_T \sim \text{Poisson}(\lambda T)$ is the random number of segments audited in $[0, T]$.

Reputation Update. Each human seat i maintains a reputation score $r_i(t) \in [0, 1]$, updated after every segment via

$$r_i(t+1) = (1 - \gamma) r_i(t) + \gamma \mathbf{1}[\text{vote correct}], \quad \gamma \in (0, 1]. \quad (\text{I.5})$$

The parameter γ controls how quickly reputation adapts to recent behavior: small γ gives a long memory while large γ reacts quickly to the most recent vote.

Reputation-Weighted Slashing. A low reputation should carry a higher risk of being penalized, to deter persistent misbehavior. We therefore define the slashing probability on an incorrect vote as

$$p_{\text{slash}}(r) = p_{\min} + (p_{\max} - p_{\min})(1 - r), \quad 0 < p_{\min} < p_{\max} \leq 1, \quad (\text{I.6})$$

which decreases linearly from $p_{\max}$ at $r = 0$ to $p_{\min}$ at $r = 1$.

Per-Segment Payoff. The net payoff $X_i \in \{-P, 0, R\}$ for seat i on one segment is R for a correct vote, $-P$ for an incorrect vote that triggers slashing, and 0 for an incorrect vote without slashing. With honest error rate ϵ_H , the expected payoff at reputation r is

$$\mu_H(r) := \mathbb{E}[X_i] = (1 - \epsilon_H)R - \epsilon_H P p_{\text{slash}}(r). \quad (\text{I.7})$$

Computer and LLM seats produce verifiable outputs and are therefore omitted from incentive analysis.

Variance and Range Bounds. To apply Bernstein-type moment-generating function (MGF) inequalities later, we need two constants. The range bound b on the centered increment $Y_i := X_i - \mathbb{E}[X_i]$ is set conservatively to

$$b := R, \quad (\text{I.8})$$

since the largest positive realization of X_i is R . The variance bound is the supremum of $\text{Var}[X_i]$ over all reputation states:

$$\sigma_H^2 := \sup_{r \in [0, 1]} \left[(1 - \epsilon_H)R^2 + \epsilon_H P^2 p_{\text{slash}}(r) - \mu_H(r)^2 \right]. \quad (\text{I.9})$$

These worst-case constants make the subsequent tail bounds reputation-agnostic, so they hold uniformly over all auditor histories.

Lemma I.3 (MGF of a Bounded Centered R.V.). Let W satisfy $\mathbb{E}[W] = 0$, $\text{Var}(W) = \sigma^2$, and $W \leq b$ a.s. with $b > 0$. Then for any $\theta \in (0, 3/b)$,

$$\mathbb{E}[e^{\theta W}] \leq \exp\left(\frac{\theta^2 \sigma^2}{2(1 - \theta b/3)}\right).$$

Proof. Follows from the standard Bernstein–Bernoulli MGF expansion; omitted for brevity. $\square$

Theorem I.4 (Safety–Profitability Guarantee). Fix horizon $T > 0$, target failure probability $\epsilon_{\text{target}} \in (0, 1)$, and design constant $\delta \in (0, 1)$. The system is governed by two tunable dials. The **statistical dial** controls how reliably the trace-level quorum is met. Let $\mu_{\text{vote}} := \mathbb{E}[W]$ and $\sigma_{\text{vote}}^2 := \sup \text{Var}(W)$ for one trace. Require the margin between the expected weighted pass sum and the quorum threshold to satisfy

$$\mu_{\text{vote}} - W_\beta \geq \sqrt{\frac{1}{2} \sigma_{\text{vote}}^2 \ln \frac{\lambda T}{\epsilon_{\text{target}}}}. \quad (\text{S1})$$

The **economic dial** controls auditor incentives. Choose $(R, P, p_{\min}, p_{\max})$ so that the minimum expected payoff for an honest seat is strictly positive and the minimum slashing rate

is large enough to guarantee malicious losses:

$$\mu_{\min} := (1 - \epsilon_H)R - \epsilon_H P p_{\max} > 0, \quad p_{\min} \geq \frac{\delta}{1 - \alpha}, \quad \alpha := \frac{P p_{\max}}{R + P p_{\max}}. \quad (\text{E1})$$

Under conditions (S1) and (E1), the following three guarantees hold simultaneously.

(a) Statistical safety. $\Pr[\text{trace fails in } [0, T]] \leq \epsilon_{\text{target}}$.

(b) Honest profitability. The cumulative payoff $U_{\text{hon}}(T) := \sum_{i=1}^{N_T} X_i$ of an honest seat remains positive with overwhelming probability:

$$\Pr[U_{\text{hon}}(T) \leq 0] \leq \exp\left[-\frac{\lambda T \mu_{\min}^2}{2\sigma_H^2 + \frac{2}{3}b \mu_{\min}}\right]. \quad (\text{I.10})$$

(c) Malicious loss. The cumulative payoff $U_{\text{mal}}(T)$ of a malicious seat is negative in expectation and remains non-negative only with exponentially small probability:

$$\Pr[U_{\text{mal}}(T) \geq 0] \leq \exp\left[-\frac{\lambda T (\delta P)^2}{2\sigma_H^2 + \frac{2}{3}b \delta P}\right], \quad \mathbb{E}[U_{\text{mal}}(T)] \leq -\lambda T \delta P. \quad (\text{I.11})$$

Proof. Let $N_T \sim \text{Poisson}(\lambda T)$ denote the random number of segments audited in $[0, T]$.

(a) Statistical Safety. A single trace's weighted pass sum W satisfies $\mathbb{E}[W] = \mu_{\text{vote}}$ and $\text{Var}(W) \leq \sigma_{\text{vote}}^2$. Applying Theorem I.2 with $a = \mu_{\text{vote}} - W_{\beta}$,

$$p_{\text{fail}} := \Pr[W < W_{\beta}] \leq \exp\left(-2(\mu_{\text{vote}} - W_{\beta})^2 / \sigma_{\text{vote}}^2\right).$$

Condition (S1) ensures the exponent is at most $-\ln(\lambda T / \epsilon_{\text{target}})$, so $p_{\text{fail}} \leq \epsilon_{\text{target}} / (\lambda T)$. Since traces arrive as a Poisson process, a union bound over $\mathbb{E}[N_T] = \lambda T$ traces gives

$$\Pr[\text{any trace fails in } [0, T]] \leq \lambda T \cdot p_{\text{fail}} \leq \epsilon_{\text{target}}.$$

(b) Honest Profitability. We bound the probability that the cumulative honest payoff is non-positive.

Step 1: Center the increments. Define $Y_i := X_i - \mu_{\min}$, so that $\mathbb{E}[Y_i] \geq 0$, $Y_i \leq b = R$, and $\text{Var}(Y_i) \leq \sigma_H^2$.

Step 2: MGF bound. By Theorem I.3, for any $\theta \in (0, 3/b)$,

$$\mathbb{E}[e^{\theta Y_i}] \leq \exp\left(\frac{\theta^2 \sigma_H^2}{2(1 - \theta b/3)}\right). \quad (\text{I.12})$$

Step 3: Chernoff bound over Poisson sum. Conditioning on $N_T = n$ and applying exponential Markov,

$$\Pr[U_{\text{hon}}(T) \leq 0 \mid N_T = n] \leq e^{-\theta n \mu_{\min}} (\mathbb{E}[e^{\theta Y_1}])^n.$$

Averaging over $N_T \sim \text{Poisson}(\lambda T)$ via the Poisson MGF $\mathbb{E}[z^{N_T}] = \exp(\lambda T(z - 1))$ with $z = e^{-\theta \mu_{\min}} \mathbb{E}[e^{\theta Y_1}]$ gives

$$\Pr[U_{\text{hon}}(T) \leq 0] \leq \exp\left(\lambda T \left\{-\theta \mu_{\min} + \frac{\theta^2 \sigma_H^2}{2(1 - \theta b/3)}\right\}\right). \quad (\text{I.13})$$

Step 4: Optimize over θ . Setting $g(\theta) = -\theta \mu_{\min} + \theta^2 \sigma_H^2 / [2(1 - \theta b/3)]$ and minimizing via the substitution $t = \theta b/3 \in (0, 1)$ yields the optimal value

$$\min_{\theta} g(\theta) = -\frac{\mu_{\min}^2}{2\sigma_H^2 + \frac{2}{3}b \mu_{\min}}. \quad (\text{I.14})$$

Substituting (I.14) into (I.13) completes the proof of (b).

(c) Malicious Loss. A malicious seat has $\mathbb{E}[X_i] \leq -\delta P < 0$ by condition (E1). The tail bound follows by repeating Steps 1–4 above with $\mu_{\min}$ replaced by δP and centering at $Z_i := X_i + \delta P$, yielding the stated exponential bound. The expected loss follows directly from linearity of expectation:

$$\mathbb{E}[U_{\text{mal}}(T)] = \lambda T \mathbb{E}[X_i] \leq -\lambda T \delta P. \quad \square$$

Table 8: Table of Notations.

Indices & random counts	
S	Total number of segments in a trace
$N_T \sim \text{Poisson}(\lambda T)$	# segments audited in horizon $[0, T]$
$t(s) \in \{C, L, H\}$	Auditor type of segment s
Per-segment vote variables (stat. layer)	
k_t	# seats of type t in a segment
$q_t = \lceil \tau k_t \rceil$	Per-type quorum (τ : vote threshold)
$B_s \in \{0, 1\}$	Segment pass indicator
$p_s = \Pr[B_s = 1]$	Segment pass probability
w_t	Weight of segment type t
Trace aggregation	
$W = \sum_s w_{t(s)} B_s$	Weighted pass total (one segment)
$W_\beta = \beta \sum_s w_{t(s)}$	Trace-level quorum threshold ($\beta \in (0, 1)$)
$\mu_{\text{vote}} = \mathbb{E}[W]$	Mean of W
σ_{vote}^2	$\sup \text{Var}(W)$ (single segment)
Human pay-off variables (econ. layer)	
R	Reward for a correct vote
P	Penalty if slashed
$p_{\text{slash}}(r)$	Slash prob. on error, reputation $r \in [p_{\min}, p_{\max}]$
ϵ_H	Honest human error rate
δ	Design constant: min. loss per malicious seat
$X_i \in \{-P, 0, R\}$	Net pay-off for seat i on one segment
$\mu_H(r)$	$\mathbb{E}[X_i]$ for honest seat with reputation r
$\mu_{\min}$	$\min_r \mu_H(r)$
b	Upper range used in Bernstein (default $b = R$)
σ_H^2	$\sup_r \text{Var}[X_i] \leq (R + P)^2 / 4$
Centred increments for MGF bounds	
$Y_i = X_i - \mu_{\min}$	Honest centred increment ($\mathbb{E}[Y_i] \geq 0, Y_i \leq b$)
$Z_i = X_i + \delta P$	Malicious centred increment ($\mathbb{E}[Z_i] \geq 0$)
Cumulative pay-offs	
$U_{\text{hon}}(T) = \sum_{i=1}^{N_T} X_i$	Total pay-off (honest seat) in $[0, T]$
$U_{\text{mal}}(T)$	Total pay-off (malicious seat) in $[0, T]$
Process rates	
λ	Segment intensity (segments per unit time)
T	Time horizon

J Motivating Example: Why Semantic Auditing Is Necessary

We present a clinical scenario in which two reasoning models produce *identical correct outputs* yet follow fundamentally different reasoning processes. This demonstrates that output-only auditing cannot distinguish sound evidence-based reasoning from error-prone reasoning that reaches the correct answer by coincidence.

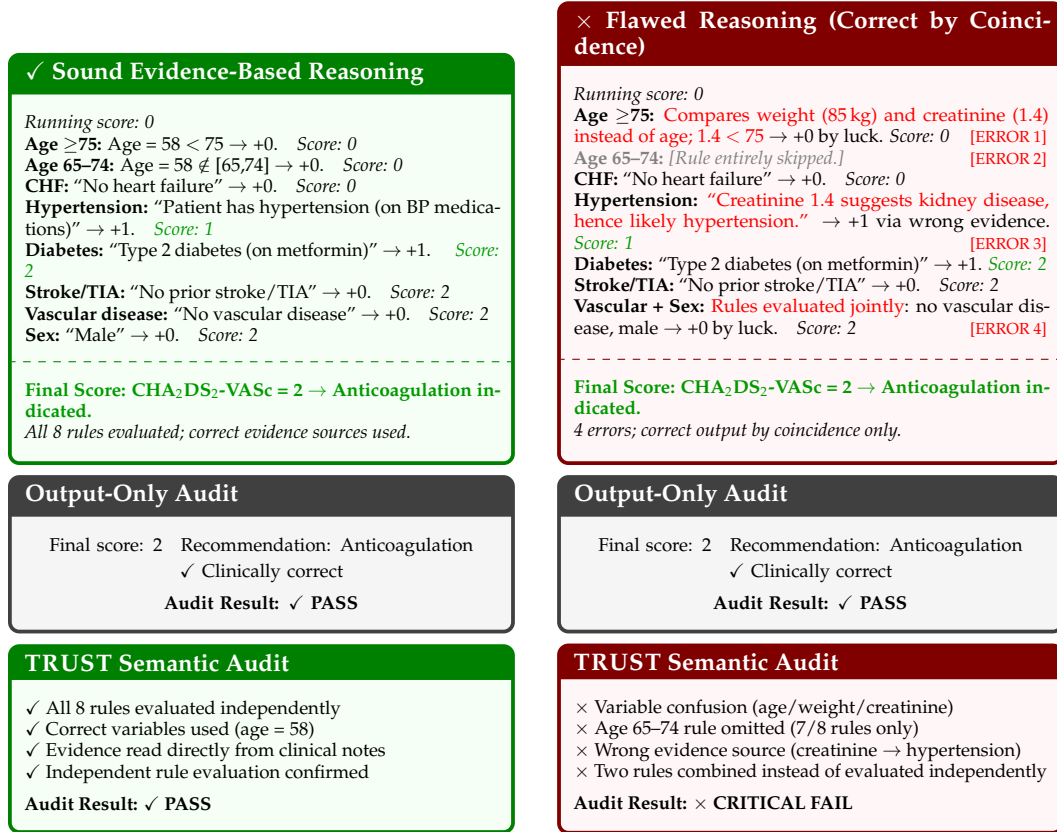


Figure 10: Two reasoning traces produce an identical, correct CHA₂DS₂-VASc score of 2, yet one is fundamentally flawed. Output-only auditing passes both. TRUST semantic auditing detects all four errors in the flawed trace and correctly rejects it. These errors cause catastrophic failures under distribution shift (e.g., when age changes to 75 or creatinine drops to 0.9).

Clinical Input
Patient Note: 58-year-old male. Weight: 85 kg; serum creatinine: 1.4 mg/dL; no active bleeding; no history of stroke. Admitted with atrial fibrillation. Clinical Note: Hypertension (on BP medications), type 2 diabetes (on metformin). No heart failure, no prior stroke/TIA, no vascular disease. Task: Calculate the CHA₂DS₂-VASc score. Scoring rules: congestive heart failure (+1), hypertension (+1), age ≥75 (+2), age 65–74 (+1), diabetes (+1), prior stroke/TIA (+2), vascular disease (+1), female sex (+1). Correct Answer: Score = 2 (Hypertension +1, Diabetes +1) → Anticoagulation recommended.

Figure 10 shows two complete reasoning traces that both arrive at score 2. The flawed trace reaches the correct answer only through four coincidences that happen to cancel: (1) confusing age (58) with weight (85 kg) and creatinine (1.4) still yields 0 age points since 58 < 65; (2) skipping the Age 65–74 rule is harmless for this patient; (3) inferring hypertension from elevated creatinine—rather than reading the clinical note—reaches the right conclusion; and (4) combining the vascular disease and sex rules still yields 0 for this male patient. Under output-only auditing, both traces pass. TRUST semantic auditing detects all four errors and correctly fails the flawed trace.

TRUST semantic auditing exposes all four hidden errors: variable confusion in age comparison, incomplete rule coverage (7 of 8 rules), wrong evidence extraction for hypertension, and improper combination of independent rules. This identifies brittle reasoning that succeeds only under specific lucky conditions and fails under distribution shift.

Why Flawed Reasoning Is Dangerous. The four errors cancel only for this particular patient profile. A small change—age 75 instead of 58, or creatinine 0.9 instead of 1.4—breaks each coincidence independently and produces incorrect scores, wrong treatment recommendations, and potential patient harm. Semantic auditing is therefore essential precisely because flawed reasoning is invisible to output-only evaluation.

Token Billing Without Value. The flawed trace also consumes more tokens than the sound trace, spending computation on wrong variable comparisons, unnecessary inferential reasoning, and combined rule evaluation. The deploying organization pays a premium for reasoning that is correct only by luck.

J.1 Three Critical Gaps That TRUST Addresses

The example above reveals three fundamental limitations of output-only auditing. First, it **cannot detect correct-by-coincidence reasoning**: both traces score 2, so both pass, even though one contains four systematic errors that will cause failures on slightly different patient profiles. TRUST catches these errors before deployment.

Second, it **cannot verify billing integrity**. When a flawed model consumes more tokens than a sound one, output-only auditing provides no means to determine whether those tokens represent genuine clinical reasoning or wasted computation. TRUST semantic auditing enables token-quality verification, ensuring organizations pay for sound reasoning rather than fortunate errors.

Third, it **cannot ensure regulatory compliance**. The FDA requires explainable clinical decision support systems with transparent reasoning trails. Output-only auditing can confirm a score but cannot explain how it was derived. TRUST produces a complete audit trail—each rule evaluated, each evidence source cited, each logical step recorded—that supports both regulatory approval and post-incident analysis.

These gaps are especially acute for commercial reasoning models such as GPT-4, Claude Extended Thinking, and OpenAI o1, whose internal reasoning is opaque. Without semantic auditing, hospitals cannot distinguish sound from coincidentally-correct outputs, and harm accumulates gradually as distribution shift erodes the lucky conditions that made flawed reasoning appear correct. TRUST closes this gap by detecting flawed reasoning patterns during pre-deployment testing, before any patients are affected.