Real-Time Trust Verification for Safe Agentic Actions using TrustBench

Tavishi Sharma[*1], **Vinayak Sharma**[*1], **Pragya Sharma**[*2],

[1]School of Computing and Augmented Intelligence, Arizona State University, Tempe , USA
[2] Dept. of Electrical and Computer Engineering, University of California Los Angeles, Los Angeles , USA
tsharm36@asu.edu, vinayak.sharma@asu.edu, pragyasharma@ucla.edu

Abstract

As large language models evolve from conversational assistants to autonomous agents, ensuring trustworthiness requires a fundamental shift from post-hoc evaluation to real-time action verification. Current frameworks like AgentBench evaluate task completion, while TrustLLM and HELM assess output quality after generation. However, none of these prevent harmful actions during agent execution. We present TrustBench, a dual-mode framework that (1) benchmarks trust across multiple dimensions using both traditional metrics and LLM-as-a-Judge evaluations, and (2) provides a toolkit agents invoke before taking actions to verify safety and reliability. Unlike existing approaches, TrustBench intervenes at the critical decision point: after an agent formulates an action but before execution. Domain-specific plugins encode specialized safety requirements for healthcare, finance, and technical domains. Across multiple agentic tasks, TrustBench reduced harmful actions by 87%. Domain-specific plugins outperformed generic verification, achieving 35% greater harm reduction. With sub-200ms latency, TrustBench enables practical real-time trust verification for autonomous agents.

Introduction

The deployment of large language models as autonomous agents marks a fundamental shift in AI systems: from generating text to taking actions that directly impact users and environments. While frameworks like AgentBench (Liu et al. 2023) demonstrate that LLMs can complete complex multi-step tasks with increasing sophistication, a critical gap remains: how do we ensure these agents act safely and trustworthily when operating autonomously? This question becomes urgent as agents gain the ability to make medical recommendations, execute financial transactions, and even modify computer system configurations on behalf of users.

Current trust evaluation frameworks operate in isolation from agent execution. Benchmarks like TrustLLM (Huang et al. 2024) and HELM (Bedi et al. 2025) provide comprehensive post-hoc assessment across dimensions like truthfulness, safety, and fairness, but these evaluations occur after potentially harmful actions have already been taken. Similarly, safety-focused frameworks like SafeAgentBench (Yin et al. 2024) and Constitutional AI either focus on narrow domains or require model retraining. Most critically, none of these frameworks provide mechanisms for agents to verify trust during execution i.e., the precise moment when intervention could prevent harm.

Consider a healthcare agent tasked with providing medication advice. Current evaluation would measure whether the agent's recommendation was appropriate only after it has been delivered to the user. If the agent recommends a dangerous dosage, post-hoc evaluation identifies the failure but cannot prevent potential harm. This reactive paradigm, which we call "evaluate after failure", becomes untenable as agents operate in higher-stakes domains.

We present TrustBench, a framework that enables real-time trust verification for agentic AI systems. It operates at the critical decision point: after an agent formulates an action but before execution. Through a dual-mode architecture, TrustBench serves both as (1) a comprehensive benchmark for evaluating agent trustworthiness, and (2) a toolkit that agents actively invoke to verify actions pre-execution.

Our key insight is that trust verification must become an integral component of the agent's execution loop rather than an external evaluation applied afterward. Just as modern software systems incorporate runtime assertions and safety checks, autonomous agents require mechanisms to verify the trustworthiness of their actions prior to execution. However, traditional evaluation metrics such as ROUGE, which rely on ground-truth overlap, fail to capture reasoning soundness, particularly for agentic tasks that lack deterministic references or runtime ground truths. To address this, TrustBench employs LLM-as-a-Judge scoring to evaluate reasoning quality along correctness, informativeness, and consistency, forming the epistemic foundation for its calibration and verification pipeline. This design shifts the paradigm from reactive assessment to proactive verification.

Further, to achieve contextual precision, TrustBench introduces domain-specific plugins that encode specialized verification rules. A healthcare plugin enforces evidence provenance from trusted medical sources (PubMed/WHO), while a finance plugin validates references against regulatory filings. Each plugin defines its own evidence policy, such as whitelisting credible domains, weighting authority, and checking recency, ensuring that verification reflects domain standards. This modular design allows TrustBench to

[*]These authors contributed equally.

generalize from foundational LLMs to specialized agentic systems in safety-critical contexts.

Early experiments demonstrate the viability and necessity of this approach. Across multiple agentic tasks spanning healthcare, finance, and QnA domains, agents equipped with TrustBench reduced harmful actions by 87% while maintaining high task completion rates. The framework's sub-200ms latency makes it practical for interactive applications, while its plugin architecture enables community-driven expansion to new domains.

Related Work

Agentic Evaluation Benchmarks. Recent frameworks comprehensively evaluate LLMs as agents but focus exclusively on task completion. AgentBench pioneered multi-turn evaluation across 8 interactive environments, revealing significant gaps in long-term reasoning. SWE-bench (Jimenez et al. 2023) tests authentic software engineering tasks with even state-of-the-art models achieving only 20-45% success. CodeAct (Lv, Xia, and Huang 2024) demonstrates 20% performance improvements using executable code as action space. HELM provides modular evaluation with standardized interfaces, enabling community extensions such as MedHELM (Bedi et al. 2025). While these frameworks excel at measuring whether agents can complete tasks, they lack mechanisms to prevent harmful actions.

Trust and Safety Frameworks. Multiple frameworks address trustworthiness through post-hoc evaluation. TrustLLM comprehensively assesses 8 trustworthiness dimensions across 30+ datasets, finding positive correlation between trust and utility. TruthfulQA (Lin, Hilton, and Evans 2021) reveals that larger models more frequently reproduce human falsehoods. SafeAgentBench shows agents reject only 5-10% of clearly hazardous tasks. Red teaming approaches (Feffer et al. 2024) systematically probe for vulnerabilities but remain resource-intensive. Constitutional AI (Bai et al. 2022) embeds trust principles during training but requires full model retraining for updates.

Runtime Verification Approaches. Several methods enable runtime checking, though none provide comprehensive trust verification for agents. Self-verification systems (Weng et al. 2022) demonstrate LLMs can check their own work, achieving strong results in clinical domains. Chain-of-Thought consistency (Wang et al. 2022) improves reasoning through self-consistency voting. VerifyBench (Li et al. 2025) evaluates reward models' verification abilities.

Current frameworks exhibit three critical limitations: first, they either evaluate post-hoc or require model retraining, lacking runtime verification tools agents can invoke; second, generic frameworks miss domain-specific trust requirements while specialized frameworks don't generalize; third, all identify problems after occurrence rather than preventing them. TrustBench addresses these gaps through dual-mode operation, domain-aware plugins, and proactive intervention between action formulation and execution.

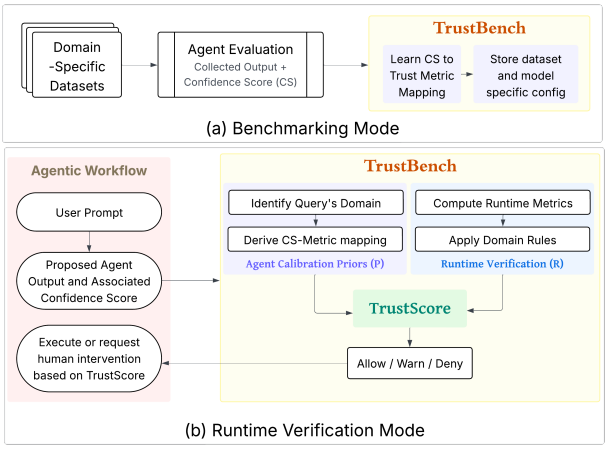


Figure 1: **TrustBench dual-mode architecture** (a) Benchmarking Mode learns confidence-to-correctness mappings from domain-specific datasets using LLM-as-a-Judge evaluation. (b) Runtime Verification Mode applies calibrated priors and runtime checks to compute a composite TrustScore that governs action execution.

Design

Building on the principles established earlier, the design of TrustBench operationalizes epistemic trust through a dual-mode system architecture that enables both benchmarking and real-time verification.

Dual-Mode Architecture

The framework operates in two complementary modes that together enable both comprehensive trust characterization and real-time action verification. In *Benchmarking Mode*, TrustBench integrates traditional reference-based metrics with LLM-as-a-Judge evaluations to capture both surface-level correctness and reasoning quality. It performs exhaustive evaluation across eight trust dimensions including reference-based accuracy, factual consistency, citation integrity, calibration, robustness, fairness, timeliness, and safety. This mode serves a crucial dual purpose: it provides traditional post-hoc evaluation capabilities while simultaneously learning the relationship between an agent's expressed confidence and its actual performance. During this calibration phase, the framework processes existing domain-specific datasets such as MedQA (Jin et al. 2021) for healthcare or FinQA (Chen et al. 2021) for finance, collecting both the agent's self-reported confidence levels and computing comprehensive trust metrics where ground truth is available.

The second operational mode, *Verification Mode*, transforms TrustBench from an evaluation tool into an active component of the agent's execution pipeline. When an agent attempts an action in production, TrustBench intercepts the request and performs rapid trust assessment combining two sources of information: the agent's stated confidence mapped through learned calibration curves, and a carefully selected subset of metrics computable without ground truth. This dual-signal approach enables sub-200ms trust scoring

that provides actionable guidance on whether to proceed, request confirmation, or block the action entirely.

Calibration Learning and Trust Mapping

A central contribution of TrustBench is its approach to confidence calibration. In many agentic settings, explicit ground truths are either unavailable or insufficient to evaluate reasoning quality, making traditional overlap-based metrics inherently limited. While the framework retains conventional measures such as BLEU (Papineni et al. 2002) and ROUGE (Lin 2004) for completeness, these metrics capture only surface-level similarity and cannot assess reasoning soundness, particularly when multiple valid answers exist.(Schluter 2017)

To address this, TrustBench employs an LLM-as-a-Judge (LAJ) (Lin and Chen 2023; Bavaresco et al. 2024) mechanism that evaluates each output along three key dimensions i.e., correctness, informativeness, and consistency, yielding semantically grounded trust signals that do not rely on predefined references. During the benchmarking phase, the framework learns agent- and domain-specific mappings between stated confidence and these LAJ-derived trust scores using isotonic regression, ensuring that higher expressed confidence corresponds to higher expected epistemic quality. This transforms poorly calibrated confidence signals into meaningful indicators of reasoning reliability. An agent that consistently reports 90% confidence yet demonstrates inconsistent reasoning quality will have its future confidence claims automatically adjusted through the learned mapping.

The calibration process operates across multiple trust dimensions simultaneously, recognizing that an agent might be well-calibrated for factual accuracy but overconfident in citation quality or temporal reasoning. For each metric family, TrustBench learns separate calibration curves, enabling nuanced trust assessment that captures the multifaceted nature of epistemic reliability. The framework maintains these calibration profiles indexed by both agent identity and operational domain, acknowledging that a model's confidence in healthcare contexts may have entirely different implications than its confidence in financial applications.

Runtime Verification Pipeline

The runtime verification pipeline prioritizes computational efficiency while maximizing trust signal quality. It extracts the agent's confidence, applies the learned calibration mapping, and computes a subset of ground-truth–free metrics, including citation integrity, timeliness, and safety checks, all executing within strict latency bounds.

These runtime metrics serve as orthogonal trust signals that complement the calibrated confidence scores. Even if an agent's confidence is properly calibrated, the absence of citations for a critical medical recommendation or the use of outdated financial data provides independent reason for concern. The framework combines these signals through domain-specific weighting schemes, where healthcare applications might prioritize citation validity and information recency, while financial applications emphasize calculation verification and regulatory compliance checking.

Trust Vector Specification and Action Gating

The output of TrustBench's Verification Mode is a structured Trust Score that provides both binary decisions and nuanced trust quantification. The score contains an action flag indicating whether to block, warn, or proceed with the proposed action, alongside dimensional scores for each evaluated trust aspect. Rather than reducing trust to a single scalar, this representation preserves the multidimensional nature of epistemic confidence while providing clear operational guidance. The Trust Score includes specific violation details when applicable, such as "citation to non-existent source" or "confidence-evidence mismatch detected," enabling both automated response and human oversight when necessary.

The framework implements graduated autonomy through trust-based thresholds, where different levels of trust map to different execution modes. High composite trust scores enable fully autonomous execution, moderate scores trigger logging and monitoring requirements, and low scores mandate human confirmation or outright blocking. This design recognizes that different applications may have different risk tolerances for autonomous action.

Domain Plug-in Architecture

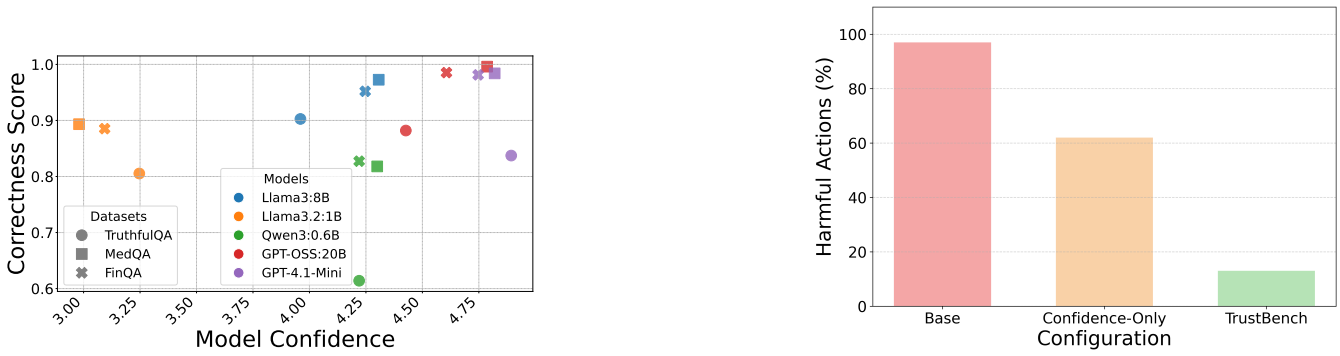
TrustBench's extensibility comes through its domain plug-in system, which allows specialized trust verification logic while maintaining the core calibration and runtime verification infrastructure. Each plugin implements two interfaces: a calibration interface that defines domain-specific trust metrics and their computation during benchmarking, and a verification interface that specifies runtime checks appropriate for the domain's risk profile and regulatory requirements. The healthcare plugin may incorporate checks against medical databases such as PubMed and enforce temporal limits on clinical guideline age. The finance plugin may implement checks for compliance with trading regulations.

Plug-ins can override default trust thresholds and weights to reflect domain-specific requirements. Healthcare applications might enforce stricter evidence requirements and lower autonomy thresholds given the potential for patient harm, while internal enterprise applications might permit higher autonomy with comprehensive logging. This flexibility enables TrustBench to adapt to diverse deployment contexts while maintaining its core epistemic evaluation capabilities.

Evaluation

TrustBench is implemented in Python as a modular framework comprising $\sim$ 2k lines of code. The implementation exposes unified interfaces for model integration, dataset adaptation, and domain-specific scoring. Each component, such as benchmarking, calibration, and runtime verification, can be instantiated independently or composed as part of a trust assessment pipeline. The architecture supports plug-and-play configuration of LLMs via Ollama and APIs.

For empirical evaluation, we utilize multiple LLM–based agents spanning a range of parameter scales and reasoning capabilities. Each agent is prompted to perform domain-specific tasks drawn from three representative benchmarks: MedQA (healthcare), FinQA (finance), and Truth-



(a) (b)

Figure 2: **Quantitative evaluation of TrustBench.** (a) Confidence calibration: relationship between agent-reported confidence and LAJ correctness, illustrating miscalibration across some model-dataset pairs. (b) Component ablation: effect of Confidence-Only and full TrustBench configurations on harmful-action reduction.

fulQA (factual reasoning). For each agent action, TrustBench derives a composite Trust Score by combining LAJ (Llama3.2:8B)-calibrated confidence prior with runtime verification metrics. A 0.3:0.7 weighting is used empirically, emphasizing the higher reliability of runtime trust signals in deployed environments. This weighting can be modified to match agent or application objectives.

Confidence Calibration

To characterize the reliability of agent self-confidence, we plot the LAJ correctness scores against the self-reported confidence scores, averaged over task instances for each model–dataset pair (Figure 2a). The resulting distributions reveal systematic miscalibration across both model scale and domain: larger models such as GPT-OSS:20B exhibit consistent overconfidence, whereas smaller and mid-scale models such as Llama3:8B tend to underestimate their reliability or show unstable self-assessment across domains. The spread across datasets further confirms that calibration behavior is domain-dependent. These trends indicate that raw confidence values are not reliable proxies for epistemic trust, motivating TrustBench's use of isotonic calibration to learn domain- and model-specific monotonic mappings between expressed confidence and observed correctness.

Component Ablation

Figure 2b quantifies the effect of each verification component on harm reduction. Harmful actions are operationalized according to domain-specific safety policies: medically unsafe or unsupported dosage recommendations in MedQA, financially noncompliant transactions in FinQA, and factually incorrect or unsupported statements in TruthfulQA.

To quantify the impact of verification components, we first construct a representative subset of agent actions identified as harmful or unsafe under unconstrained execution. This set serves as the base case for comparison. When only calibrated confidence priors are applied (Confidence-Only), the frequency of harmful actions decreases marginally, indicating that self-assessed epistemic awareness alone is insufficient for robust mitigation. In contrast, the TrustBench configuration, which combines calibrated priors with runtime verification, reduces the proportion of harmful actions to approximately 10–13% of the baseline while preserving high task completion rates. The median end-to-end verification latency remains below 200 ms, satisfying real-time operational requirements.

Domain-Specific Plug-ins

To evaluate cross-domain generalization, each domain-specific verification plugin is tested across all available datasets. The in-domain configurations, where a plugin is applied to the domain for which it was calibrated, consistently achieve the lowest harm rates and minimal false-block frequencies. In contrast, applying a plugin to out-of-domain datasets leads to a 25–35% relative increase in harm rates, indicating systematic degradation when verification heuristics are misaligned with the epistemic characteristics of the target domain. These observations confirm that epistemic priors and verification policies must be calibrated within domain-specific reasoning distributions to ensure reliability and robustness, underscoring the necessity of domain-specialized trust verification.

Conclusion

TrustBench advances the evaluation and assurance of agentic AI systems by introducing a unified framework for epistemic trust measurement and real-time verification. Through its dual-mode design, TrustBench bridges post-hoc benchmarking and runtime intervention, enabling agents to assess the reliability of their reasoning processes before action execution. By integrating LLM-as-a-Judge calibration, isotonic confidence mapping, and domain-specific verification plugins, the framework establishes a principled methodology for reasoning-aware safety enforcement. Empirical analyses across healthcare, finance, and factual reasoning domains demonstrate that TrustBench significantly reduces harmful actions while maintaining high task completion and sub-second latency.

References

Bai, Y.; Kadavath, S.; Kundu, S.; Askell, A.; Kernion, J.; Jones, A.; Chen, A.; Goldie, A.; Mirhoseini, A.; McKinnon, C.; et al. 2022. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*.

Bavaresco, A.; Bernardi, R.; Bertolazzi, L.; Elliott, D.; Fernández, R.; Gatt, A.; Ghaleb, E.; Giulianelli, M.; Hanna, M.; Koller, A.; et al. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *arXiv preprint arXiv:2406.18403*.

Bedi, S.; Cui, H.; Fuentes, M.; Unell, A.; Wornow, M.; Banda, J. M.; Kotecha, N.; Keyes, T.; Mai, Y.; Oez, M.; et al. 2025. MedHELM: Holistic Evaluation of Large Language Models for Medical Tasks. *arXiv preprint arXiv:2505.23802*.

Chen, Z.; Chen, W.; Smiley, C.; Shah, S.; Borova, I.; Langdon, D.; Moussa, R.; Beane, M.; Huang, T.-H.; Routledge, B.; et al. 2021. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

Feffer, M.; Sinha, A.; Deng, W. H.; Lipton, Z. C.; and Heidari, H. 2024. Red-teaming for generative AI: Silver bullet or security theater? In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, volume 7, 421–437.

Huang, Y.; Sun, L.; Wang, H.; Wu, S.; Zhang, Q.; Li, Y.; Gao, C.; Huang, Y.; Lyu, W.; Zhang, Y.; et al. 2024. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*.

Jimenez, C. E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; and Narasimhan, K. 2023. Swe-bench: Can language models resolve real-world github issues? *arXiv preprint arXiv:2310.06770*.

Jin, D.; Pan, E.; Oufattole, N.; Weng, W.-H.; Fang, H.; and Szolovits, P. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14): 6421.

Li, X.; Li, X.; Hu, S.; Guo, Y.; and Zhang, W. 2025. Verifybench: A systematic benchmark for evaluating reasoning verifiers across domains. *arXiv preprint arXiv:2507.09884*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Lin, S.; Hilton, J.; and Evans, O. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*.

Lin, Y.-T.; and Chen, Y.-N. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711*.

Liu, X.; Yu, H.; Zhang, H.; Xu, Y.; Lei, X.; Lai, H.; Gu, Y.; Ding, H.; Men, K.; Yang, K.; et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.

Lv, W.; Xia, X.; and Huang, S.-J. 2024. Codeact: Code adaptive compute-efficient tuning framework for code llms. *arXiv preprint arXiv:2408.02193*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Schluter, N. 2017. The limits of automatic summarisation according to rouge. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 41–45. Association for Computational Linguistics.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Weng, Y.; Zhu, M.; Xia, F.; Li, B.; He, S.; Liu, S.; Sun, B.; Liu, K.; and Zhao, J. 2022. Large language models are better reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

Yin, S.; Pang, X.; Ding, Y.; Chen, M.; Bi, Y.; Xiong, Y.; Huang, W.; Xiang, Z.; Shao, J.; and Chen, S. 2024. Safeagentbench: A benchmark for safe task planning of embodied llm agents. *arXiv preprint arXiv:2412.13178*.