I am a **computer systems** researcher. My research explores how *computer systems* can both leverage and advance *artificial intelligence (AI)* to improve *system reliability*. Specifically, I develop techniques that automate operational tasks (e.g., detection, diagnosis, and mitigation) in large-scale systems, sitting at the intersection of systems, AI, and software engineering (SE), all toward building *reliable, intelligent,* and *secure systems*.

Cloud systems are becoming ever more critical — worldwide public cloud spending is expected to hit $723 billion in 2025. Yet failures are the norm in the cloud: outages or incidents happen every day, downtime for large systems can run over $500,000 per hour. Despite massive investments in automation, today's cloud operations still rely heavily on human engineers. My research tackles this fundamental question: how can we embed intelligence into systems so they can autonomously detect, diagnose, and recover from failures safely, efficiently, and at scale?

My PhD research advances the reliability of large-scale cloud systems by developing new techniques and platforms for the full lifecycle of incident management: 1) **detection:** I developed automatic fault-injection and testing methods [3, 8] that expose hidden failure modes and uncover reliability issues; 2) **diagnosis:** I designed techniques based on large language models (LLM) [4, 7, 10] that analyze heterogeneous telemetry to pinpoint root causes, and 3) **mitigation:** I built agentic AI systems [1, 9] that can execute safe recovery or mitigation automatically. To support this research agenda and ensure rigorous evaluation, I built evaluation frameworks [2, 6] for AIOps approaches across realistic cloud environments. I have published broadly not only in *systems* [2, 3, 4, 9], but also in *AI* [1, 6], *SE* [7, 8, 10], and *security* [5, 11, 12] venues.

My view is in the notion of *endogenous* system intelligence: given high-level requirements (e.g., oracles for system health, safety properties), system intelligence can drive continuous discovery of solutions and autonomously act on them. This view began with my key observation: cloud system failures emerge from the complex interplay of code, configuration, workload, and runtime dynamics, often driven by evolving conditions that static models or pre-defined rules hardly capture. Operationalizing this insight, I make systems more instrumented (to expose rich and precise telemetry for AI reasoning), interpretable (so humans can understand and verify AI decisions), and safe to operate (e.g., write actions are reversible and bounded in scope). These principles enable AI to become an active yet controlled participant in system operation rather than a passive observer.

Impact and recognition. My root-cause analysis system, RCACopilot, has been deployed in Microsoft's production to support live incident management, and I had two internships there to transfer my research into practice. My work on reliability testing has discovered 70+ and fixed 50+ new bugs in open-source cloud systems. My evaluation system, ITBench, received the *Oral* and *Spotlight* at ICML 2025. Beyond technical contributions, my research has shaped education and outreach: it has been featured in graduate classes at UIUC and the Indian Institute of Science, and presented in tutorial sessions at top-tier venues such as SOSP, DSN, and FSE. My research has been widely covered by the media, including Microsoft Research Blog, IBM Research Blog, CIO, The New Stack, The Weekend Read, and MarkTechPost; and AIOpsLab was recognized as one of the "Best AI Agent Papers of 2024" by Juteq. In addition, I received the Mavis Future Faculty Fellow award at UIUC.

Future work. My long-term vision is to make AI an effective and trustworthy partner in operating real-world systems, not limited to cloud systems. In the near term, I will make clouds more "AI-operable", developing agentic AI systems that can not only debug and resolve issues, but also manage and optimize cloud systems end-to-end. I will continue collaborating closely with industry partners (e.g., Microsoft, IBM, and Meta) to build evaluation platforms that mirror production-scale complexity, enabling AI evaluation under realistic, large-scale conditions. Looking ahead, I aim to extend these ideas to broader classes of cyber-physical and socio-technical systems, including IoT, financial, and robotics systems, where we need to rethink how agentic AI can effectively help.

1 Main Research

Testbed for Evaluating Agentic AI in Live System Operation [2, 6]

AI's rapid progress owes much to benchmarks, and there are already many. However, evaluating AI, especially agentic AI, for real-world system operations remains hard: most site reliability engineering (SRE) benchmarks use static, pre-collected data and cannot capture interactive, evolving behavior of cloud environments and the agents. SRE agents must reason and act as the system changes across a large action space, where failures can occur at any layer, and valid mitigations may span many controls. This raises a big-picture question: how effective and safe are such agents in live clouds? My work has taken the first step by building two open-source platforms for interactive, reproducible agent evaluation in clouds.

At the core, my evaluation platforms, AIOpsLab [2] (with Microsoft) and ITBench [6] (with IBM), contribute: (1) a live orchestration layer that deploys microservice applications, injects faults in real time, generates workloads, provides observability of the heterogeneous telemetry, and exposes a principled agent–cloud interface (ACI) for easy and safe interaction; (2) 200+ task-oriented scenarios that evaluate agents separately on four types of tasks: detection, localization, root-cause analysis, and mitigation, enabling fine-grained attribution of agents' strengths and weaknesses; and (3) a portable, extensible framework that unifies metrics and supports easy integration of user-defined services, workloads, and fault models, so evaluations generalize beyond our reference applications. Beyond SRE, ITBench extends to 50 SecurityOps and two FinOps scenarios. Under these, state-of-the-art SRE agents resolve up to 13.8% of the problems in ITBench, showing significant potential to improve.

Detecting System Defects [3, 7, 8]

Proactive defect detection is the first guardrail before failures surface. Modern applications increasingly depend on cloud services (e.g., AWS S3, Azure Storage) for various functionalities. Despite the benefits, such "cloud native" practice imposes emerging reliability challenges introduced by the fault models of opaque cloud backends and less predictable connections between the application and cloud services. My research tries to address the emerging reliability challenges by building a "push-button" reliability testing tool named Rainmaker [3], as a basic SDK utility for any cloud-backed application. Rainmaker helps developers easily and systematically test their applications' correctness, in the face of various errors under the cloud-based fault model. The core of Rainmaker is its automatic fault injection policies that define what faults to inject and where (e.g., at which REST calls) to inject faults. Rainmaker's fault injection policies are guided by a bug taxonomy: It considers transient error(s) that can occur during one REST API call initiated by the application (and the corresponding retries by the SDK); yet, it captures common bug patterns and shows that error (mis)handling of even one REST call can have major impacts on application correctness. Rainmaker has detected 73 bugs (51 fixed) in 11 popular cloud-backed applications.

We further applied the testing technique to service emulators [8], revealing 94 discrepancies across 255 APIs from five Azure and AWS services. In total, 37% of the tested cloud APIs behaved differently in the emulator versus the real service. These discrepancies lead to inconsistent testing results, threatening deployment safety, introducing false alarms, and creating debuggability issues.

Beyond API-level testing, I built Ciri [7], an LLM-empowered validator that detects misconfigurations in the system. Ciri turns a config file or diff into a prompt with few-shot examples drawn from valid and misconfigured cases, automatically retrieves relevant code snippets, queries one or more LLMs, and enforces configuration validation. On ten widely used systems, Ciri achieved file- and parameter-level F1 scores of up to 0.79 and 0.65, respectively, where file-level F1 measures the

ability to identify misconfigured files, and parameter-level F1 measures the accuracy of pinpointing specific wrong parameters. Ciri detected 45 of 51 real-world misconfigurations, outperforming other baselines.

Diagnosing the Causes of Failures [4, 10]

Despite early prevention, production failures are inevitable. When they occur, we resort to root cause analysis (RCA) to diagnose the issue. However, RCA is challenging because the relevant evidence, including logs, metrics, traces, and deployment history, is often scattered and noisy. When conducting RCA, engineers rely on troubleshooting guides, which are static documents that hard-code the debugging steps for engineers to refer to. These guides are slow to apply, labor-intensive, and quickly out of date. My work, RCACopilot [4], leverages LLMs to do RCA within a structured, two-stage pipeline: (1) an incident-matched handler automatically collects incident telemetry, and (2) RCACopilot then retrieves similar historical incidents by performing a similar neighbor search in the embedding space, uses an LLM to summarizes the incoming and retrieved incidents, reasons over them, and predicts a root cause label with explanation. The incident handler is inspired by my observation that the on-call debugging process follows a decision-tree pattern. The handler requires engineers to define it once, then automatically aggregates the diagnostic information when new alerts come. In evaluation, RCACopilot can achieve RCA accuracy up to 0.766 and has been integrated into Microsoft's production system across more than 30 teams.

Moreover, I conducted the first large-scale empirical study [10] of incidents in generative AI (GenAI) cloud services. By analyzing incidents from the cloud provider hosting OpenAI's models, we uncovered unique reliability challenges there, including invalid inference, quality degradation, and privacy risks. The study shows that GenAI incidents are harder to detect and resolve: they take 1.8x longer to mitigate and are 3x more often detected by humans rather than automated monitors.

Towards Safe and Autonomous Recovery [1, 9]

Failure recovery is usually the last and most difficult stage of incident management because it requires state-changing actions that, if unsafe, can worsen failures. To address this challenge, I designed STRATUS [1], a multi-agent system that automates detection, diagnosis, and safe mitigation. Its core principle, Transactional No-Regression (TNR), ensures that every mitigation plan is undoable and that the system never regresses below its original state after the mitigation procedure. STRATUS realizes TNR through undoable actions, sandboxed execution, and bounded risk windows. In experiments on AIOpsLab and ITBench's benchmarks, STRATUS achieved mitigation success rates of 69.2% and 50.0%, significantly higher than prior SRE agents.

2 Other Research [5, 11, 12]

Beyond addressing reliability issues, I have also tackled external cyber threats that put systems at risk. I helped develop Shadewatcher [12], a real-time cyber threat detection system, by mapping security concepts of system entity interactions to recommendation concepts of user-item interactions. Also, I contributed to the development of a system, Watson [11], to automatically abstract low-level audit logs into high-level user behaviors. It can cluster semantically similar behaviors to reduce analyst effort by over two orders of magnitude during attack investigation. In a separate endeavor, I co-authored a survey [5] on comprehensive systematization of the system auditing literature, with a focus on data provenance techniques.

3 Future Research

While my prior work has shown how AI can enhance existing systems when applied as an afterthought, my future research aims to rethink the design of AI-native systems that integrate intelligence as a core part of their operation.

Making multi-model and multi-agent architectures more efficient. My earlier work [1] introduced multi-agent designs for site reliability engineering (SRE). Going forward, I see efficiency as a central challenge in scaling multi-agent and multi-model coordination. While much recent work has optimized single-model inference and serving efficiency, little attention has been given to the higher-level coordination efficiency across multiple agents and models. Open problems include: how agents can synchronize effectively without incurring excessive or unordered communication, and how to decide when to invoke large models versus smaller ones in latency-sensitive scenarios. I plan to explore adaptive mechanisms for agent coordination and model selection, to enable responsive and resource-efficient intelligence.

Safety assurance and execution confinement in operations. Safety is the cornerstone of trustworthy autonomous operation. Agents interact with external systems through API calls, for instance, via the Model Context Protocol (MCP), to retrieve information or modify external system states. Without proper safeguards, such agent actions can be hazardous, potentially driving the system into worse states. I will continue to design mechanisms that make agent actions reversible and interpretable. Specifically, I plan to integrate rollback-aware execution to remove side effects whenever necessary and develop validation oracles that continuously assess post-action system health, deciding whether the agent should retry, reflect, or escalate. These designs will extend my earlier concept of Transactional No-Regression (TNR) [1], providing safety guarantees for AI-driven actions in production systems.

Extending autonomous operation beyond cloud systems. Drawing from my work on autonomous cloud operations, I will explore how intelligent systems can manage cyber-physical or socio-technical environments such as IoT infrastructures, financial platforms, and robotic systems. Unlike cloud environments, these domains have their own challenges: actions have physical or economic consequences, sensing is noisy, decisions must be both timely and safe, and computation may be limited. These differences call for a fundamental rethinking of how intelligence in operations perceives, reasons, and acts when embedded within other real-world systems.

Human–AI cooperative operation. Even as operations become increasingly automated, human insight and intervention remain invaluable. I plan to build collaborative autonomy, where humans and AI agents share control, context, and responsibility in systems. I will design interactive frameworks in which agents can explain their reasoning, quantify uncertainty, and request guidance when confidence is low, while humans can provide corrective feedback that the agents incorporate through continual learning.

Toward holistic evaluation platforms. Building scalable and reproducible evaluation platforms remains a central goal of my research. Built upon my previous experience with open-source frameworks, AIOpsLab [2] and ITBench [6], I plan to develop more holistic platforms that evaluate agentic AI systems under realistic workloads and diverse fault types, so that we can approach deployment of the agents in real, production environments more closely. Beyond operational effectiveness, the new evaluation system will also assess security and trustworthiness: examining whether agents can withstand adversarial manipulation and detect or mitigate security incidents, e.g., DDoS attacks. My goal is to create a rigorous foundation for developing dependable AI agents.

In summary, my future research aims to establish a principled foundation for effective, safe, and efficient systems with agentic AI.

References

[1] **Yinfang Chen**, Jiaqi Pan, Jackson Clark, Yiming Su, Noah Zheutlin, Bhavya Bhavya, Rohan Arora, Yu Deng, Saurabh Jha, and Tianyin Xu. "STRATUS: A Multi-agent System for Autonomous Reliability Engineering of Modern Clouds". In: *Proceedings of the Thirty-Ninth Annual Conference on Neural Information Processing Systems (NeurIPS'25)*. Dec. 2025.

[2] **Yinfang Chen**, Manish Shetty, Gagan Somashekar, Minghua Ma, Yogesh Simmhan, Jonathan Mace, Chetan Bansal, Rujia Wang, and Saravan Rajmohan. "AIOpsLab: A Holistic Framework to Evaluate AI Agents for Enabling Autonomous Clouds". In: *Proceedings of the Eighth Annual Conference on Machine Learning and Systems (MLSys'25)*. May 2025.

[3] **Yinfang Chen**, Xudong Sun, Suman Nath, Ze Yang, and Tianyin Xu. "Push-Button Reliability Testing for Cloud-Backed Applications with Rainmaker". In: *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation (NSDI'23)*. 2023.

[4] **Yinfang Chen**, Huaibing Xie, Minghua Ma, Yu Kang, Xin Gao, Liu Shi, Yunjie Cao, Xuedong Gao, Hao Fan, Ming Wen, Jun Zeng, Supriyo Ghosh, Xuchao Zhang, Chaoyun Zhang, Qingwei Lin, Saravan Rajmohan, Dongmei Zhang, and Tianyin Xu. "Automatic Root Cause Analysis via Large Language Models for Cloud Incidents". In: *Proceedings of the Nineteenth European Conference on Computer Systems (EuroSys'24)*. 2024.

[5] Muhammad Adil Inam, **Yinfang Chen**, Akul Goyal, Jason Liu, Jaron Mink, Noor Michael, Sneha Gaur, Adam Bates, and Wajih Ul Hassan. "SoK: History is a Vast Early Warning System: Auditing the Provenance of System Intrusions". In: *2023 IEEE Symposium on Security and Privacy (S&P'23)*. 2023.

[6] Saurabh Jha, Rohan Arora, Yuji Watanabe, Takumi Yanagawa, **Yinfang Chen**, Jackson Clark, Bhavya Bhavya, Mudit Verma, Harshit Kumar, Hirokuni Kitahara, Noah Zheutlin, Saki Takano, Divya Pathak, Felix George, Xinbo Wu, Bekir O. Turkkan, Gerard Vanloo, Michael Nidd, Ting Dai, Oishik Chatterjee, Pranjal Gupta, Suranjana Samanta, Pooja Aggarwal, Rong Lee, Pavankumar Murali, Jae-wook Ahn, Debanjana Kar, Ameet Rahane, Carlos Fonseca, Amit Paradkar, Yu Deng, Pratibha Moogi, Prateeti Mohapatra, Naoki Abe, Chandrasekhar Narayanaswami, Tianyin Xu, Lav R. Varshney, Ruchi Mahindru, Anca Sailer, Laura Shwartz, Daby Sow, Nicholas C. M. Fuller, and Ruchir Puri. "ITBench: Evaluating AI Agents across Diverse Real-World IT Automation Tasks". In: *Proceedings of the 42nd International Conference on Machine Learning (ICML'25)*. July 2025.

[7] Xinyu Lian*, **Yinfang Chen***, Runxiang Cheng, Jie Huang, Parth Thakkar, and Tianyin Xu. "Large Language Models as Configuration Validators". In: *Proceedings of the 47th International Conference on Software Engineering (ICSE'25)*. Apr. 2025.

[8] Anna Mazhar, Saad Sher Alam, William Zheng, **Yinfang Chen**, Suman Nath, and Tianyin Xu. "Fidelity of Cloud Emulators: The Imitation Game of Testing Cloud-based Software". In: *Proceedings of the 47th International Conference on Software Engineering (ICSE'25)*. Apr. 2025.

[9] Manish Shetty, **Yinfang Chen**, Gagan Somashekar, Minghua Ma, Yogesh Simmhan, Xuchao Zhang, Jonathan Mace, Dax Vandevoorde, Pedro Las-Casas, Shachee Mishra Gupta, Suman Nath, Chetan Bansal, and Saravan Rajmohan. "Building AI Agents for Autonomous Clouds: Challenges and Design Principles". In: *Proceedings of the 15th ACM Symposium on Cloud Computing (SoCC'24)*. Nov. 2024.

[10] Haoran Yan*, **Yinfang Chen***, Minghua Ma, Ming Wen, Shan Lu, Shenglin Zhang, Tianyin Xu, Rujia Wang, Chetan Bansal, Saravan Rajmohan, Chaoyun Zhang, and Dongmei Zhang **(*: Equal Contributions)**. "An Empirical Study of Production Incidents in Generative AI Cloud Services". In: *Proceedings of the 36th IEEE International Symposium on Software Reliability Engineering (ISSRE'25)*. Oct. 2025.

[11] Jun Zeng, Zheng Leong Chua, **Yinfang Chen**, Kaihang Ji, Zhenkai Liang, and Jian Mao. "WATSON: Abstracting Behaviors from Audit Logs via Aggregation of Contextual Semantics". In: *Network and Distributed System Security Symposium (NDSS'21)*. 2021.

[12] Jun Zeng, Xiang Wang, Jiahao Liu, **Yinfang Chen**, Zhenkai Liang, Tat-Seng Chua, and Zheng Leong Chua. "Shadewatcher: Recommendation-guided cyber threat analysis using system audit records". In: *2022 IEEE Symposium on Security and Privacy (S&P'22)*. 2022.