Yuntianyi Chen

Email: yuntianc@uci.edu   Homepage: yuntianyi-chen.github.io

EDUCATION

- **Ph.D. Candidate in Software Engineering** — Irvine, CA
  University of California, Irvine — *Sep 2021 - Present*
  Donald Bren School of Information and Computer Sciences

- **Bachelor of Computer Science and Technology** — Wuhan, China
  Wuhan University — *Sep 2016 - Jun 2020*
  School of Computer

PUBLICATIONS

[**FSE 2025**] Yuntianyi Chen, Yuqi Huai, Yirui He, Shilong Li, Changnam Hong, Qi Alfred Chen, Joshua Garcia. "A Comprehensive Study of Bug-Fix Patterns in Autonomous Driving Systems". Proc. ACM Softw. Eng. 2, FSE (2025).

[**FSE 2024**] Yuntianyi Chen, Yuqi Huai, Shilong Li, Changnam Hong, Joshua Garcia. "Misconfiguration Software Testing for Failure Emergence in Autonomous Driving Systems". Proc. ACM Softw. Eng. 1, FSE (2024), 1913–1936.

[**TSE 2023**] Yuqi Huai, Sumaya Almanee, Yuntianyi Chen, Xiafa Wu, Qi Alfred Chen, Joshua Garcia. "scenoRITA: Generating Diverse, Fully Mutable, Test Scenarios for Autonomous Vehicle Planning". IEEE Trans. Software Eng. 49, 10 (2023), 4656–4676.

[**ICSE 2023**] Yuqi Huai, Yuntianyi Chen, Sumaya Almanee, Tuan Ngo, Xiang Liao, Ziwen Wan, Qi Alfred Chen, Joshua Garcia. "Doppelgänger Test Generation for Revealing Bugs in Autonomous Driving Software". In 45th IEEE/ACM International Conference on Software Engineering, ICSE 2023, Melbourne, Australia, May 14-20, 2023. IEEE, 2591–2603.

[**APSEC 2019**] Yongfeng Gu, Yuntianyi Chen, Xiangyang Jia, Jifeng Xuan. "Multi-Objective Configuration Sampling for Performance Ranking in Configurable Systems". In 26th Asia-Pacific Software Engineering Conference, APSEC 2019, Putrajaya, Malaysia, December 2-5, 2019. IEEE, 150–157.

[**SOFL+MSVL 2019**] Yuntianyi Chen, Yongfeng Gu, Lulu He, and Jifeng Xuan. "Regression Models for Performance Ranking of Configurable Systems: A Comparative Study". In Structured Object-Oriented Formal Language and Method - 9th International Workshop, SOFL+MSVL 2019, Shenzhen, China, November 5, 2019. Springer, 243–258.

EXPERIENCE

- **Autoware Foundation** — Irvine, CA
  Collaborating Researcher | Open-Source Developer — *Aug 2023 - Present*
  - **Configuration Architecture Refactoring**: Contributed to **10%** of all refactoring of Autoware ROS nodes. Developed a tool for automated configuration refactoring in Autoware ROS nodes, enabling developers to accelerate their workflow.
  - **Scenario Record Analyzer**: Developed an automated tool to detect **9** types of violations in driving scenario records.
  - **DevOps Dojo**: Joined the Autoware Open AD Kit working group for **DevOps Dojo** development, which aims to accelerate optimized hardware and software solutions for autonomous driving.

- **Software Aurora (SORA) Lab - University of California, Irvine** — Irvine, CA
  Graduate Student Researcher — *Sep 2021 - Present*
  - **ADS Testing**: Conducted foundational research in autonomous driving systems, focusing on scenario generation and configuration testing to detect bug-revealing violations. Developed **4 open-source projects** shared with the community.
  - **Bug-Fix Pattern Benchmark**: Open-sourced a **benchmark** of **1,331** bug-fix instances for bug-fix pattern study in ADSes. Built an interactive tool using **Node.js**, **MongoDB**, and **TypeScript** to enable classification and collaborative labeling.
  - **SE4ADS Workshop Organization**: Helped organize the **first international workshop** of SE4ADS at **ICSE**, a top-tier conference in software engineering, promoting discussions on software engineering challenges in autonomous driving systems.

- **Centre of Software Testing, Analysis and Reliability (CSTAR) - Wuhan University** — Wuhan, China
  Research Assistant — *Jul 2020 - Jul 2021*
  - **Transfer Learning**: Proposed a method that uses the labeled data of other projects to help optimize the target projects through feature-level transfer learning in the performance prediction, which can greatly reduce the data measurement cost.
  - **Performance Prediction and Optimization**: Developed a **multi-objective** configuration sampling method that optimizes performance ranking while reducing measurement cost, achieving up to **5x reduction** in sampling cost.

SELECTED PROJECTS

- **A Comprehensive Study of Bug-Fix Patterns in Autonomous Driving Systems**

  Autonomous Driving Systems | Empirical Software Engineering | Bug-Fix Pattern   *Jul 2023 - Sep 2024*
  - Conducted the first empirical study of bug-fix patterns in autonomous driving systems (ADS), analyzing **1,331 bug fixes** from two major open-source ADS projects, Apollo and Autoware.
  - Introduced a novel **taxonomy** of **15 syntactic** and **27 semantic** bug-fix patterns, providing a structured framework for understanding and categorizing ADS bug repairs.
  - Developed a modularization granularity **hierarchy**, mapping bug-fix patterns to ADS modules, sub-modules, algorithms, and code statements for systematic debugging and repair.
  - Created a public **benchmark** of 1,331 bug-fix instances, offering researchers and practitioners valuable data for automated bug detection and repair in ADS.

- **Misconfiguration Software Testing for Failure Emergence in Autonomous Driving Systems**

  Autonomous Driving Systems | Search-Based Software Testing | Software Configuration   *Aug 2022 - Mar 2024*
  - Developed ConfVE, the **first automated configuration testing framework** for autonomous driving systems (ADS), enabling the discovery of bug-revealing violations without modifying existing scenario-generation techniques.
  - Implemented **9 test oracles** and designed a **multi-objective optimization** genetic algorithm to detect system failures and configuration-induced violations.
  - Introduced a duplicate violation eliminator, leveraging an **unsupervised machine learning** approach (i.e., clustering) to minimize redundant failure detections and identify truly unique violations.
  - Evaluated ConfVE on Apollo and Autoware ADSes, running 124,950 virtual tests over 990 hours, leading to the discovery of 1,818 unique violations across 9 violation types. Demonstrated superiority over existing approaches, achieving up to a **65.88% improvement** in unique violation detection compared to ConfVD and **74.19% reduction** in duplicate elimination.

- **scenoRITA: Generating Diverse, Fully Mutable, Test Scenarios for Autonomous Vehicle Planning**

  Autonomous Driving Systems | Search-Based Software Testing | Unsupervised Machine Learning   *Dec 2022 - May 2023*
  - Developed scenoRITA, a **search-based test generation** framework for autonomous vehicle (AV) software that leveraged the evolutionary algorithm and domain-specific constraints for **multi-objective optimization**, ensuring automated scenario generation while dynamically adapting obstacles to expose safety-critical situations.
  - Implemented a **novel gene representation**, enabling obstacles to **evolve dynamically**, modifying attributes such as start and end location, type, speed, size, and mobility during test generation.
  - Designed **5 test oracles** to assess AV safety and comfort violations, covering collision detection, speeding detection, unsafe lane change, fast acceleration, and hard braking.
  - Executed 79,051 virtual tests on Baidu Apollo 7.0 using 4 high-definition city maps. Compared with state-of-the-art approaches (AV-FUZZER, AutoFuzz), demonstrating **4.95x** to **8.22x** more scenario generation efficiency and detecting **1,146** unique violations versus only 7 and 37 from competing techniques.

- **Doppelgänger Test Generation for Revealing Bugs in Autonomous Driving Software**

  Autonomous Driving Systems | Search-Based Software Testing   *Jun 2022 - Sep 2022*
  - Developed DoppelTest, a methodology where **multiple ADS instances** control all vehicles in a scenario, ensuring every violation is caused by ADS misbehavior, making it bug-revealing.
  - Designed **smart obstacles**, ensuring traffic rule compliance and defensive driving to eliminate unrealistic, non-bug-revealing violations caused by unintelligent obstacles.
  - Discovered **123 bug-revealing violations** in Apollo 7.0, categorized into 8 unique bug types, including failure to recognize slow dynamic obstacles and unsafe lane borrow decisions. Proposed and submitted a fix for one bug type discovered by DoppelTest, which was **accepted by Apollo developers**, demonstrating real-world impact.
  - Achieved a **100% bug-revealing rate**, compared to baseline random testing, which generated over 1,100 violations but none that exposed actual ADS bugs. Executed 240 hours of testing, demonstrating runtime efficiency, with DoppelTest **reducing analysis time by 77.11%** compared to prior methods like scenoRITA.

- **Multi-Objective Configuration Sampling in Configurable Systems**

  Multi-objective Optimization | Performance Prediction | Software Configuration   *Jun 2019 - Aug 2019*
  - Developed MoConfig, a **multi-objective** configuration sampling method that optimizes performance ranking in configurable systems by balancing measurement cost and ranking difference. Introduced three alternative objectives to effectively estimate ranking difference without requiring prior performance data, enabling an **unsupervised sampling** approach.
  - Conducted experiments on **20** real-world datasets, demonstrating that MoConfig achieves better ranking accuracy while requiring fewer sampled configurations than the baseline.
  - Validated MoConfig with four multi-objective optimization algorithms (NSGA-II, eMOEA, IBEA, and DBEA), finding that **NSGA-II** consistently performs well, ensuring reliable ranking predictions.
  - Achieved up to **5x reduction in sampling cost**, showing that MoConfig maintains ranking accuracy while significantly lowering computational overhead compared to existing approaches.

- **Regression Models for Performance Ranking of Configurable Systems: A Comparative Study**

  Supervised Machine Learning | Performance Prediction | Software Configuration   *May 2018 - Jul 2019*

- Conducted a comparative study on **4 regression methods** (CART, SVR, GPR, GBRT) for performance ranking in configurable systems, evaluating their effectiveness on **21** evaluation scenarios across **16** real-world systems.
- Demonstrated that CART and GBRT **outperform** other methods in performance ranking, achieving better minimum actual ranks (MAR) in the majority of scenarios. Showed that SVR and GPR require fewer sampled configurations, **reducing the cost** of performance ranking but at the expense of ranking accuracy.
- Implemented an iterative sampling strategy, effectively reducing the number of required performance measurements, leading to up to a **50% reduction in sampling effort** compared to exhaustive methods.

SERVICES

Organization

- **Web & Publicity Chair** — Workshop
  1st International Workshop on Software Engineering for Autonomous Driving Systems (ICSE-SE4ADS 2025) — *Sep 2024*
- **Local Arrangements Chair** — Workshop
  The Workshop of Intelligent Real-time Methods and Technologies of Quality Improvement based on Co-programming — *Apr 2021*

Program Committee

- **Program Committee Member** — Workshop
  1st International Workshop on Software Engineering for Autonomous Driving Systems (ICSE-SE4ADS 2025) — *Sep 2024*
- **Program Committee Member** — Artifacts Evaluation
  47th IEEE/ACM International Conference on Software Engineering (ICSE 2025) — *Sep 2024*
- **Program Committee Member** — Artifacts Evaluation
  31st ACM Conference on Computer and Communications Security (CCS 2024) — *May 2024*
- **Program Committee Member** — Artifacts Evaluation
  21st International Conference on Software Architecture (ICSA 2024) — *Mar 2024*

Reviewer

- **Reviewer** — Journal
  IEEE Transactions on Intelligent Transportation Systems (T-ITS) — *Apr 2025*
- **Reviewer** — Artifacts Evaluation
  47th IEEE/ACM International Conference on Software Engineering (ICSE 2025) — *Jan 2025*
- **Reviewer** — Journal
  ACM Transactions on Software Engineering and Methodology (TOSEM) — *Dec 2024*
- **Reviewer** — Workshop
  1st International Workshop on Software Engineering for Autonomous Driving Systems (ICSE-SE4ADS 2025) — *Nov 2024*
- **Reviewer** — Journal
  IEEE Transactions on Software Engineering (TSE) — *Nov 2024*
- **Reviewer** — Journal
  ACM Transactions on Software Engineering and Methodology (TOSEM) — *Jul 2024*
- **External Reviewer** — Conference
  33rd ACM SIGSOFT International Symposium on Software Testing and Analysis (ISSTA 2024) — *May 2024*
- **Reviewer** — Artifacts Evaluation
  31st ACM Conference on Computer and Communications Security (CCS 2024) — *May 2024*
- **External Reviewer** — Conference
  47th IEEE/ACM International Conference on Software Engineering (ICSE 2025) — *Apr 2024*
- **Reviewer** — Artifacts Evaluation
  21st International Conference on Software Architecture (ICSA 2024) — *Mar 2024*
- **Reviewer** — Journal
  IEEE Robotics and Automation Letters (RA-L) — *Jan 2024*
- **Delegate Reviewer** — Conference
  41st IEEE International Conference on Robotics and Automation (ICRA 2024) — *Oct 2023*
- **External Reviewer** — Conference
  32nd ACM International Conference on the Foundations of Software Engineering (FSE 2024) — *Oct 2023*
- **Sub-reviewer** — Conference
  20th IEEE International Conference on Software Architecture (ICSA 2023) — *Dec 2022*
- **Sub-reviewer** — Journal
  27th IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2020) — *Nov 2019*

Volunteer

- **Student Volunteer** — Conference
  31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2023) — *Dec 2023*
- **Student Volunteer** — Symposium
  Southern California Software Engineering Symposium (SuCSES 2023) — *May 2023*

Talks

- **Conference Paper Presentation** Hangzhou, China
  "Regression Models for Performance Ranking of Configurable Systems: A Comparative Study" *Nov 2019*
  The Annual Conference on Software Analysis, Testing and Evolution (SATE 2019)

Honors and Awards

- ACM SIGSOFT CAPS Award in 2024
- NSF Student Travel Award in 2023
- Chair's Award, University of California, Irvine in 2021
- Scholarship for Outstanding Students (Top 5%) in 2017-2018, 2016-2017, and 2018-2019
- Chinese Undergraduate Computer Design Contest (Provincial Level, Second Prize) in 2018