

Jincheng Ma

Software Engineer In Test | AI Agents & Test Infra

San Jose, CA | (872) 239-9245 | majincheng990128@gmail.com | LinkedIn | GitHub

Skills

Languages: Go (Golang), Python, TypeScript, JavaScript, Java, SQL

AI & LLM: LLM Agents, Multi-Agent Systems, Agent Orchestration, LLM Evaluation & Benchmarking, Model Context Protocol (MCP), LLM API Integration, AI Coding Tools (Claude Code, Cursor), LLM Observability

Backend & Distributed Systems: RESTful APIs, gRPC, Microservices, Distributed Systems, System Design, Kafka, Redis, PostgreSQL, MySQL, Spring Boot, Next.js, React

Cloud & DevOps: AWS, GCP, Docker, Kubernetes, Linux, Git, GitHub Actions, GitLab CI, CI/CD, Observability, Agile/Scrum

Testing & Quality: Test Automation, API Testing, Integration Testing, E2E Testing, Regression Testing, Playwright, Pytest, JUnit, Postman, Code Review

Work Experience

TikTok Inc.

Jul 2025 – Sep 2026

Software Engineer In Test

San Jose, CA

- Architected and shipped a **CI-native multi-agent code review system** reviewing every merge request across **30+** internal repositories, orchestrating **OpenCode** agent sessions behind a **composable plugin architecture** isolating the VCS provider, the model provider, and internal engineering-standard checks.
- Engineered a **coordinator agent** that fans out to **7 concurrent domain specialists** (security, performance, code quality, documentation, release, internal compliance), each in an isolated session, then adjudicates their output into one **schema-validated** review: deduplicating findings, re-categorizing misfiled ones, filtering speculative noise, and **blocking the merge** on any critical-severity issue.
- **Calibrated the adjudication layer** against reviewer judgment so severity calls stayed correct rather than merely consistent, using streaming **JSONL** telemetry for per-task **token and cost attribution** to compare model tiers and configurations on quality, latency, and cost per review.
- Cut per-review cost with **risk-tiered orchestration** scaling agent count and model tier to diff size, file count, and security-sensitive paths; **diff noise filtering** and a **shared context file** under prompt caching removed the 7x context duplication of a naive fan-out and reduced token spend **over 30%**.
- Hardened the system for CI with per-model **circuit breakers and fallback chains**, multi-level timeouts, hung-session detection, and **input sanitization** closing prompt-injection paths into the coordinator.
- Owned end-to-end **test automation** for Trust & Safety products, building backend **API/integration** and frontend **E2E** suites wired into **CI/CD release gates**, lifting P0/P1 regression automation from **80.21% to 95.82%**; drove **service reliability** from production metrics and on-call trend analysis, cutting production issues by **30%** and lifting inspection stability from **93.94% to 99.99%**.
- Led a **cross-region test-gap analysis** across application code, runtime configuration, middleware, and third-party dependencies, cataloging **19 region-specific scenarios** and **60 configuration sets** and turning them into a layered validation strategy of **configuration diff and snapshot checks**, **API/RPC assertions**, and targeted **E2E tests**.

INFL.US

Jan 2025 – Jun 2025

Software Engineer In Test

Chicago, IL

- Built an internal **data anomaly inspection platform** (Next.js, TypeScript) running **150+ scheduled inspection jobs daily** over **50+ tables** across multiple databases, with a dashboard for continuous visibility into data health and a **webhook-based notification** path pushing detected anomalies to team channels in real time.
- Built a **Playwright** E2E automation framework and **Android** workflows integrated with **CI/CD**, automating regression across core flows to improve release confidence.

Carrefour China (Suning.com)

Jul 2021 – Mar 2022

Backend Software Engineer (Management Trainee Program)

Shanghai, China

- Developed and optimized **RESTful** query APIs in **Go** (net/http) for a retail **data-middle-platform** delivering self-service **data query and BI**, reaching **80 ms** average response and **35%** higher throughput via endpoint and concurrency tuning.
- Owned the platform's **access-control** module, designing **role-based permission groups (RBAC)** with **OAuth 2.0** and **JWT** to govern which teams could query which datasets and dashboards.
- Accelerated retrieval with **Redis** caching (go-redis, **85%** hit rate) and **PostgreSQL** tuning via **pgx** and strategic indexing, cutting critical query time by **40%**; built **CI/CD** in **GitLab CI** (build, golangci-lint, go test, Docker) with automated API tests, cutting release cycle time **30%**.

Education

Illinois Institute of Technology – M.S. Information Technology & Management; GPA 4.0/4.0

April 2022 – Dec 2024

Nanjing University of Technology – B.E. Material Science and Engineering;

Sep 2017 – Jun 2021