Zicheng Ma

Objective: Seeking for 2026 Fulltime Software Engineer.

Email: zichengma@g.harvard.edu
Mobile(US):+1-217-693-2214
Linkedin: linkedin.com/in/zicheng-ma/
Github: github.com/ZichengMa

Education

- **Harvard University** — US
  M.S. in Computational Science and Engineering GPA: 3.8/4.0 — *May 2026 (Expected)*
- **Massachusetts Institute of Technology** — US
  Cross Registration in EECS Department — *May 2026 (Expected)*
- **University of Illinois Urbana-Champaign** — US
  B.S. in Computer Engineering ***GPA: 3.98/4.00*** — *June 2024 (Graduated with* ***Highest Honor****)*
- **Zhejiang University** — China
  B.E. in Electronic and Computer Engineering ***GPA: 3.98/4.00*** — *June 2024(Graduated with* ***Highest Honor****)*

Skills

- **Programming**: Python, Rust, C, C++, Go, SQL, CUDA, Java, JavaScript, x86 assembly, HTML/CSS, Bash
- **Frameworks**: Kubernetes, vLLM, SgLang, TensorRT, LMCache, Huggingface, Pytorch, ONNX, OpenVINO, React.js
- **Tools**: Git, Linux/UNIX, Docker, MySQL, Azure, Google Cloud(GCP), GDB, MongoDB, Neo4j, Zookeeper

Professional Experience

- **NVIDIA Dynamo Team** — Santa Clara, US
  Software Development Engineer Intern (Rust, Python, vLLM, AI Infra) — *May 2025 – Present*
  - **Fault tolerance and Observability**: Enhanced Dynamo's distributed runtime health monitoring by designing and implementing unified component-level HTTP endpoints (`/health`, `/metrics`). Led system design discussions collaborating across teams. Built backbone for the whole fault tolerance system.
  - **Aggregated Serving LMCache Integration**: Modified Dynamo's vllm Python client to integrate LMCache components, enabling efficient CPU offloading. This optimization **improved throughput by over 20%** and significantly reduced GPU memory usage pressure.
  - **Disaggregated Serving LMCache Integration**: Innovatively leveraged vllm's MultiConnector to connect NixlConnector and LMCacheConnector, allowing KV offloading and KV transfer happen simultaneously. Designed a seamless integration to support disaggregated serving aligned with vllm engine's scheduling and Dynamo's disaggregated serving logic. Achieved a **25% improvement in time-to-first-token** and successfully deployed in production as the team's highest priority.
- **Microsoft** — Beijing, China
  Software Development Engineer Intern(Python, ONNX, Huggingface, Pytorch) — *August 2023 - February 2024*
  - **LLM inference acceleration**: Accelerated the inference speed of LLM on Intel CPU, resulting in a **6700% speed increase**. Enabled more efficient real-time processing, reduced operational costs, and made inference on CPU feasible.
  - **Image generation acceleration**: Sped up StableDiffusion inference by optimizing memory allocation and CPU multi-core usage, achieving a **tenfold** speedup. Was invited to **give a speech to a team of over 50 engineers**.
  - **Automatic slot filtering**: Developed a pipeline to automatically parse user queries into different slots with LLM. Provided 50,000+ pieces of data for the product team. The effort led to an LLM parser, **now live on Bing Real Estate.**
  - **LLM4Feedback Agent**: Researched leveraging LLMs to analyze user feedback on software products, using few-shot demonstration techniques and data cleaning methods. Developed an AI agent based on the Microsoft TaskWeaver framework, automating analysis and leading to more accurate insights, which improved decision-making in product development
- **Verifiable cloud cluster controller framework** — Urbana-Champaign, US
  Research Assistant, collaborated with VMware Research.(Rust, Kubernetes, Go) — *March 2023 - November 2023*
  - **Develop k8s controllers for different application**: Developed controllers for mainstream applications, like ZooKeeper, RabbitMQ, and Cassandra on Kubernetes clusters using **Rust** and Verus (verified Rust), enhancing application management. Implemented mechanisms for **scaling, upgrading, updating, and booting up** clusters, ensuring robust management.
  - **Design a new framework to verify controller**: Designed a new framework for building Kubernetes controllers to verify liveness and safety, resulting in the creation of verifiable controllers and a descriptive model for controllers.
  - **Import Kubernetes APIs to the new framework**: Developed wrappers to describe the properties of various k8s APIs, facilitating the creation of specifications used in formal proofs, which facilitated formal proofs and ensured controllers adhered to defined properties, contributing to the overall system's verifiability.
  - **Unit-test**: Designed unit tests with Rust, connecting proof and actual operations to ensure system integrity.

Publications

- **Anvil: Verifying Liveness of Cluster Management Controllers**: Xudong Sun, Wenjie Ma, Tyler Gu, **Zicheng Ma**, Tej Chajed, Jon Howell, Andrea Lattuada, Oded Padon, Lalith Suresh, Adriana Szekeres, Tianyin Xu. (**OSDI 2024 best paper**)
- **AllHands: Ask Me Anything on Large-scale Verbatim Feedback via Large Language Models**: C. Zhang, **Z. Ma**, Y. Wu, S. He, S. Qin, X. Qin, Y. Liang, Y. Xue, Q. Lin, S. Rajmohan, D. Zhang, Q. Zhang. (ICDE 2025)
- **ElaLoRA: Elastic & Learnable Low-Rank Adaptation for Efficient Model Fine-Tuning**: Huandong Chang, **Zicheng Ma**, Mingyuan Ma, Zhenting Qi, Andrew Sabot, Hong Jiang, H. T. Kung. arXiv:2504.00254 (2025)

Honors and Awards

- Mathematical Contest In Modeling Finalist Award (Top 1% of the competitors worldwide – May 2022)