Chun-Mao (Michael) Lai

☎ 858-373-7788 ✉ michaellai901026@gmail.com Chun-Mao Lai Mecoli1219 🏠 www.mecoli.net

Engineer focused on Systems and ML Systems, with experience building distributed infrastructure, contributing to open-source, and solving complex, real-world problems across data, compute, and coordination layers.

EDUCATION

University of California San Diego **09/2024 – Expected 12/2025**
MS. in Computer Science and Engineering *La Jolla, California*

- GPA: 3.93/4.0; Selected Courses: Data Systems for Machine Learning, Distributed Computing & Systems, Networked Services.

University of Illinois Urbana-Champaign **08/2023 – 12/2023**
Exchange Program in Electrical and Computer Engineering *Urbana, Illinois*

- GPA: 4.0/4.0; Selected Courses: Distributed Systems, Machine Learning.

National Taiwan University **09/2020 – 06/2024**
BSE. in Electrical Engineering *Taiwan*

- GPA: 3.97/4.0; Published 5 Machine Learning research papers, including 2 as first author and 3 in top-tier conferences (NeurIPS, ICML).

SKILLS/QUALIFICATIONS

Languages	Python, Javascript, Typescript, Go, C/C++, Java, Scala, Rust, Bash
ML/AI	PyTorch, CUDA, Triton, Liger-Kernel, HuggingFace, TensorFlow, Keras, Scikit-learn
Frameworks & Tools	Kubernetes, Docker, Apache Spark, Apache Kafka, Flyte, Ray, gRPC
Web Development	HTML/CSS, React, NextJS, Tailwind, ThreeJS, NodeJS, ExpressJS, Swagger, GraphQL, Prisma, Flask
Data & Storage	MySQL, PostgreSQL, MongoDB, Redis, Apache Hive, HDFS, OpenLineage
Platforms & DevOps	Linux, MacOS, Git, VS Code, IntelliJ, Jira

SELECTED WORK EXPERIENCE

LinkedIn Corporation (On-going) **06/2025 – 09/2025**
Systems & Infrastructure Engineering Intern | Scala, Java, Spark, Kafka, OpenLineage *Sunnyvale, CA*

- Exploring column-level data lineage for Spark jobs by analyzing logical plans via AST parsing, with the goal of enabling fine-grained lineage tracking across sources like Hive and HDFS.
- Prototyping a Kafka-based service to collect and process lineage metadata, supporting efficient and distributed ingestion and storage.

Taiwan Semiconductor Manufacturing Company (TSMC) **06/2023 – 07/2023**
Machine Learning Research Engineer Intern | Python, C, SQL, TensorFlow *Taiwan*

- Designed an innovative pairwise Style Transfer model for 3M-pixel super-resolution images, reducing error by 50%.
- Optimized image processing pipeline with MPI (Python), cutting processing time by 75% and achieving 5x faster training with TensorFlow on 2-node, 8-GPU cluster.

OPEN SOURCE CONTRIBUTIONS

Flyte (K8S Workflow Orchestration Platform For Data & ML Pipelines) **04/2024 – Present**
Open Source Committer | Go, Python, Docker, Kubernetes, gRPC *Remote*

Jupyter Notebook Support in Flytekit

- Enabled Jupyter Notebook support in Flytekit, allowing users to develop and run code remotely from notebook cells. (PR: #2733)
- Applied pickling techniques to solve Jupyter Notebook serialization issues and created comprehensive integration tests.
- Collaborated with Union.ai and the open-source community to refine system design. (PR: #2799)

Selected Contributions

- Enhanced resource efficiency by introducing Vertical Pod Autoscaling in Flyte, minimizing waste from failed tasks (PR: #6293)
- Designed and implemented system-wide support for Tuple and NamedTuple types in Flyte, including FlyteIDL schema and server-side updates, client library enhancements, and a formal RFC proposal. (RFC: #5699)
- Enabled default labels and annotations for the launch plans automatically created from workflow definitions. (PR: #2776)

Liger-Kernel (LLM Training Kernel) & HuggingFace TRL (Transformer Reinforcement Learning) **11/2024 - 02/2025**
Open Source Contributor | Python, Triton *Remote*

- Integrated Liger-Kernel's JSD, CPO, and SimPO algorithms into HuggingFace TRL, ensuring compatibility between both libraries.

SELECTED PROJECTS

Fault-Tolerant Distributed UDP Game Server **05/2025 – 06/2025**
Final Project - Distributed Computing & Systems | Golang *UC San Diego*

- Designed and built a novel fault-tolerant distributed system architecture using UDP—a rarely adopted approach for distributed systems—to support real-time, frame-synchronized multiplayer gameplay.
- Ensured no additional packet loss or latency during normal operations, with primary failure recovery incurring only negligible delay.

NTUEE Light Dance **09/2021 – 07/2023**
Software Leader | Typescript, GraphQL, Prisma, MySQL, Redis | https://www.youtube.com/@ntueelightdance6849 *Taiwan*

- Led a 25-member team responsible for developing the Light Dance editor service, managing a substantial codebase of 800,000 lines.
- Built the backend service from scratch to facilitate the storage of light dance data (up to 5GB) on a server and provide a co-editing environment.
- Refactored the storage schema using an SQL-based backend, which reduced client-side latency to under 1 second per operation.