

Chufeng Jiang

✉ chufeng.jiang@utexas.edu | ☎ +1-925-860-9229 | 🌐 <https://chufeng-jiang.github.io/>

Summary

I am a Ph.D. Candidate in Computer Science at the Graduate Center, City University of New York (CUNY), and expect to graduate between *July and December 2027*. I am co-advised by Dr. Feng Gu at CUNY and Dr. Defu Cao at the University of Southern California (USC). My research focuses on efficient AI systems, with particular interests in LLM inference and serving. I am currently conducting research on runtime optimization for Mixture-of-Experts (MoE) models, including execution-context-aware backend selection, efficient kernel execution, and adaptive inference systems.

Education

The City University of New York	Ph.D in Computer Science	Aug.2024 - Dec.2027 (Expected)	USA
The University of Texas at Austin	M.Sc. in Computer Science	Jan.2024 - Aug.2026	USA
Nankai University	BEng in Computer Science and Technology	Sep.2020 - Jun.2023	CN
Hunan University	BMgt in Accounting	Sep.2013 - Jun.2017	CN

Technical Skills

Languages	Python, C/C++, CUDA C++, Java, Go, Rust, Kotlin, JavaScript, Bash
LLM / ML Systems	PyTorch, Transformers, vLLM, Triton, FlashInfer, CUTLASS, Hugging Face, ONNX
Inference Optimization	MoE, KV Cache, Continuous Batching, Paged Attention, Prefix Caching, Quantization
GPU Performance	CUDA, Triton Kernels, Tensor Cores, CUDA Graphs, Kernel Fusion, Nsight Compute, Nsight Systems
Distributed Systems	NCCL, MPI, Docker, Kubernetes, Kafka, Spark, gRPC, REST APIs
Cloud / Infrastructure	AWS, Linux, CI/CD, GitHub Actions, Docker Compose, Kubernetes, Prometheus, Grafana
Backend / Frameworks	Spring Boot, Django, Flask, FastAPI, Node.js, JUnit, Gradle, Maven
Databases / Search	PostgreSQL, MySQL, MongoDB, Redis, OpenSearch, CockroachDB, Oracle

Selected Publications

- Chufeng Jiang, Feng Gu. “RouteFlow: Execution-Context-Aware Backend Selection for Efficient MoE Inference”. **In preparation**
- Chufeng Jiang, Heng Li, Zhiqi Wang, Hongda Ma, Defu Cao, Feng Gu. “One Backend Does Not Fit All: Routing-Aware GPU Execution for Sparse MoE Recommenders.” **In preparation**
- Chufeng Jiang, Heng Li, Zhiqi Wang, Hongda Ma, Defu Cao, Feng Gu. “When Agents Pander to Tests: An Empirical Study of Validation Misalignment in Software Engineering Agents”. Submitted to ICSE 2027
- Chufeng Jiang, Heng Li, Zhiqi Wang, Hongda Ma, Defu Cao*, Feng Gu*. “ViRuleEval: Interpretable Neuro-Symbolic Evaluation of Text-to-Video Generative Models”. 2026 IEEE the 5th International Conference on Artificial Intelligence, Human-Computer Interaction and Robotics (To Appear)
- Chufeng Jiang*, Heng Li. “Tri-LaneNet: Enhancing Intelligent Transportation Systems Perception Capabilities”. 50th IEEE Conference on Computers, Software, and Applications.
- Chufeng Jiang, Neng-Fa Zhou*. “Identifying Good Rules for Efficient SAT Encodings of Single Constant Multiplication Using Machine Learning”. The 42nd International Conference on Logic Programming.
- Neng-Fa Zhou*, Chufeng Jiang, Hendrik Bierlee, Peter Stuckey. “Dynamic Programming and Tabled Logic Programming for Encoding Single-Constant Multiplication into SAT”. 18th International Symposium on Functional and Logic Programming.
- Su, Jing, Chufeng Jiang, Xin Jin, Yuxin Qiao, Tingsong Xiao, Hongda Ma, Rong Wei, Zhi Jing, Jiajun Xu, and Junhong Lin. “Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review”. arXiv:2402.10350 (2024).

Work Experience

Backend Developer, Prospect Equities®	July. 2026 - Present	Remote
- Developed Java/Spring Boot REST APIs for sales and rental map search, supporting building-level marker grouping, filtering, and paginated retrieval of property listings. - Redesigned the OpenSearch indexing and aggregation pipeline by integrating PostgreSQL building identifiers and implementing composite aggregations with search-after pagination, removing an inaccurate 500-hit grouping limit and improving scalability for large result sets. - Diagnosed an OpenSearch scalability bottleneck in map-search queries, identifying unbounded requests that returned over 25,000 markers, generated approximately 4.87 MB responses, and reached 4.45 s latency; currently redesigning query, aggregation, and response paths to reduce latency and payload size. - Built JUnit/Mockito unit and controller tests covering sales and rental APIs, filtering, aggregation, invalid mappings, and validation failures; maintained backward-compatible OpenAPI contracts and collaborated with frontend engineers on API integration and performance debugging.		
Adjunct Lecturer, CUNY	Jan. 2026 - Present	NY, USA
- Independently led end-to-end course delivery for Medgar Evers College–<i>CIS 310 Computer Graphics</i> and Bronx Community College–<i>DAT 47 Java Programming</i>, including curriculum design, lectures, and assessment. - Designed assignments and projects evaluations, developed grading rubrics, and ensured fair and consistent assessment. - Mentored students in debugging and project development, addressing complex programming and system design challenges.		

Research Projects

RouteFlow: Execution-Context-Aware Backend Selection for MoE LLM Serving Aug. 2026 – NY, USA

- Leading an **ongoing research project** on execution-context-aware runtime optimization for **MoE LLM serving**, developing **RouteFlow** with **vLLM**, **Triton**, and **FlashInfer CUTLASS** on an **NVIDIA RTX 5090** to enable adaptive GPU backend selection beyond conventional isolated profiling.
- Built a **1,024-context** exact workload-matched benchmark with **Granite 4.0 H Tiny**, spanning **32 prompts**, **8 MoE layers**, and **4 prefill/decode positions**, and replayed MoE inputs under natural LLM execution and isolated profiling.
- Engineered a reproducible GPU profiling pipeline with **fresh-process isolation**, exact tensor capture/replay, **CUDA**-event timing, candidate warmup, counterbalanced backend ordering, hash-based correctness validation, and independent runtime-variability calibration.
- Current results show that natural and isolated execution disagree on the raw fastest backend in **62.30%** matched contexts; under an independently calibrated **31.29% conservative variability threshold**, **15.23%** retain a resolved backend-winner inversion, with substantially stronger mismatch during **decode** and in **early MoE layers**.
- Ongoing work includes **independent fresh-process replication**, execution-context feature analysis, **context-aware backend prediction and selection**, latency-regret and selector-overhead evaluation, end-to-end serving experiments, and generalization across additional **MoE models**, **numerical formats**, and **GPU architectures**.

RouteMoE: Context-Preserving GPU Optimization for Sparse MoE Inference April. 2026 – NY, USA

- Identified **execution-context-dependent backend ranking** in sparse MoE inference: identical explicit workloads favored different GPU backends under natural versus isolated execution, exposing a limitation of conventional fixed-trace profiling for runtime backend selection.
- Conducted root-cause analysis using **Nsight Systems**, controlled cache eviction, and activation-locality experiments; identified **pre-MoE activation residency** as the dominant factor, explaining approximately **98%** of the context benefit and reducing Triton FC1 kernel time by **25.3%** under hidden-resident execution.
- Designed a lightweight **context-preserving backend selector** that preserves the natural execution path during offline calibration, eliminating fixed-trace context distortion without online profiling.
- Achieved **100% oracle agreement** and **zero measured regret** across **168 preregistered blind cases**, improving geometric-mean inference latency by **1.26%** over the best static backend with only **0.2 μ s** of decision overhead.
- Built an end-to-end GPU inference and benchmarking system for a BF16 sparse MoE model trained on **TencentGR-1M (1.0M users, 90.2M events)**; implemented execution paths using **PyTorch**, **Triton**, **CUDA/C++**, and **cuBLAS**, including device-resident dispatch, grouped FC1/FC2, fused FP32 combine, padded BMM, and grouped GEMM.

A Portable LLM Inference System for Heterogeneous GPUs Jan. 2026 - Jun. 2026 NY, USA

- Built an end-to-end **LLM inference engine** that integrates model loading, device management, operator dispatch, transformer execution, and autoregressive generation; validated full inference of **DeepSeek-R1-Distill-Qwen-1.5B** with all **339** model weights loaded through the custom execution stack.
- Designed a modular **heterogeneous execution architecture** that decouples common inference logic from vendor-specific GPU runtimes, enabling shared device, memory, and operator abstractions across **NVIDIA CUDA** and **MetaX MACA**.
- Implemented core LLM operators including **Add**, **RMSNorm**, **RoPE**, **SwiGLU**, **Embedding**, **Argmax**, **Linear**, and **Self-Attention**, enabling shared GPU implementations and backend-specific optimization across accelerators.
- Developed a **Triton operator backend** with implementations of key LLM kernels and systematic comparisons against **PyTorch** and native GPU kernels across tensor shapes, data types, and execution configurations, exposing both numerical and performance portability issues.
- Built a GPU benchmarking and profiling workflow using **Triton**, **Nsight Compute**, and **SASS analysis** to study memory access, launch configuration, instruction behavior, and kernel decomposition, achieving up to **12% kernel-level speedup** through vectorization and launch-policy optimization.

Qwen2.5 Reasoning Alignment via Supervised Fine-Tuning Sep. 2025 - Dec. 2025 NY, USA

- Developed an end-to-end reasoning alignment pipeline for **Qwen2.5-7B**, combining multi-source CoT supervised fine-tuning with **GRPO**-based reinforcement learning.
- Engineered and unified **50K+ reasoning samples** from multi-source datasets, applying filtering, standardization, and **<think>** annotations to improve data quality and structured reasoning.
- Designed multi-objective reward functions incorporating answer correctness, CoT structure, numerical coverage, and length constraints, enabling stable and structured policy optimization.
- Achieved reward improvements from **0.99** to **1.50** over 1,000 training steps, with low KL divergence across most iterations.

LiveOak Compiler University of Texas at Austin Jun. 2024 - Aug. 2024 TX, USA

- Developed a full compiler for a Java-like language, conducting **lexical/syntactic analysis** to build an AST and **semantic analysis** including type checking, formals/actuals validation, and class/method verification.
- Implemented **register allocation** and generated optimized machine code for two target architectures: a stack-based virtual machine (**SaM**) and **x86 assembly**.

Awards and Fellowships

1. 2026 ERI/ PS2 Summer Public Research Fellows, City University of New York
2. 2024 Graduate Center Fellowship, City University of New York
3. 2022 Outstanding Student Award, Nankai University