Qingyao Sun

Google Scholar – nalzok@GitHub – Website

qs234@cornell.edu
+1 (872)212-9451

Education

Cornell University — Ithaca, US
Ph.D. in Computer Science — *Aug. 2023 – Present*

University of Chicago — Chicago, US
M.S. in Statistics; GPA: 3.9 — *Sept. 2021 – Mar. 2023*

University of California, Berkeley — Berkeley, US
Exchange Student; A+ in CS 61[A-C] — *Aug. 2018 – Dec. 2018*

East China Normal University — Shanghai, China
B.S. in Statistics; GPA: 3.78 (Major GPA: 3.96; Rank: 1/71) — *Sept. 2015 – Jul. 2020*

Research

Mikado: Sparse Linear Algebra for Graph Computation, *Cornell University* — Feb. 2026 – Present

- Reformulated genotype representation graph (GRG) matrix multiplication as SpTRSV using the reachability matrix $(I - A)^{-1}$
- Designed Mikado, a dependency-driven blocked SpTRSV pipeline; implemented CPU/GPU versions with MKL/cuSPARSE
- Achieved 452× over single-threaded C++ baseline and 3.2× speedup over optimized GPU traversal on A100 80 GB
- Built BOLT-LMM-inf benchmarks and statistical validation; Mikado GPU delivered 18.7× speedup and 11.7× cost saving
- Co-first author of the bioRxiv preprint *Sparse Linear Algebra Accelerates Genotype Representation Graph Computation at Biobank Scale*; submitted to Nature Methods. Available: https://www.biorxiv.org/content/10.64898/2026.09.10.750583.

GRG-Enabled Algorithms for Computational Genomics, *Cornell University* — Sept. 2025 – Present

- Developed a SNP-space gene–environment reformulation that reuses genotype products across L environments, reducing the dominant cost from $\Theta(LMNk)$ to $\Theta(MNk)$; studying estimator variance, probe distributions, and batched GRG multiplication
- Investigating sparse matrix exponential evaluation and continuous-time Markov simulation for `mrpast` demographic inference

LLM Quantization with QTIP, *Cornell University* — Mar. 2024 – Aug. 2024

- Achieved 8× nominal weight compression and 3.36× FP16 throughput (2-bit Llama 2-7B, batch 1, RTX 6000 Ada)
- Improved perplexity over QuIP# by applying the random permutation trellis encoder to incoherence-processed LLM weights
- Implemented high-performance CUDA kernels on NVIDIA {P40, 3090, A6000, A100, H100} GPUs for QTIP dequantization
- Led to the paper *QTIP: Quantization with Trellises and Incoherence Processing.* NeurIPS 2024 Spotlight. Available: https://arxiv.org/abs/2406.11235.

LLM Quantization with QuIP#, *Cornell University* — Nov. 2023 – Feb. 2024

- Improved the numerical stability and speed of QuIP# quantization with matrix factorization
- Implemented CUDA dequantization; inference reached 56.8% peak memory bandwidth (2-bit Llama 2-70B, RTX 4090)
- Led to the paper *QuIP#: Even Better LLM Quantization with Hadamard Incoherence and Lattice Codebooks.* ICML 2024 Poster. Available: https://arxiv.org/abs/2402.04396.

Test–Time Label–Shift Adaptation, *Google Research* — Jun. 2022 – Aug. 2023

- Addressed spurious correlation in classification by developing {JAX, PyTorch} implementations for an EM algorithm to calculate the MAP estimator for test-time adaptation on target label distribution with both neural networks and gradient-boosted trees
- Validated improvements over invariant and ERM baselines on TPU VMs and NVIDIA A100 GPUs
- First author of *Beyond Invariance: Test-Time Label-Shift Adaptation for Distributions with "Spurious" Correlations.* NeurIPS 2023 Poster. Available: https://arxiv.org/abs/2211.15646.

Iterative Random Forest, *Lawrence Berkeley National Laboratory* — Jul. 2019 – Oct. 2019

- Halved the execution time of the R package `iRF` for iterative random forests by rewriting critical subroutines in C++
- Created the R package `tree.interpreter` in C++/Armadillo which implements the MDI-oob algorithm for random forests

Internships

Applied Scientist Intern
Amazon Web Services, Inc. — *May. 2026 – Present*

- Optimizing Hessian-aware quantization and variable-bit allocation using numerical optimization and stochastic estimation.

Test Systems Engineering Intern
Tesla, Ltd — *May. 2025 – Aug. 2025*

- Developed a Python controller for Raichu (next-generation Tesla internal power supply) to orchestrate its hardware components
- Enabled asynchronous operations with Trio which provides a $\sim 50\%$ speed gain and robust handling for hardware failure
- Implemented mock CAN/UDS I/O for offline integration testing; set up GitHub Actions for linting and unit/integration tests

Professional Service

Reviewer: AISTATS 23, NeurIPS 2[3-6], ICML 2[3-6], ICLR 2[4-7], AAAI 25, SPIGM 25, TMLR — Online
Reviewer: IEEE Transactions on Circuits and Systems for Video Technology — Online
Reviewer: IEEE Transactions on Information Theory — Online