

Michael Mansour

Los Angeles, CA — (818) 823-5726 — michael@suminnovation.xyz
SUMChain: MMwxCo6U35xubBLi6N3ouktKiT7FM2gFE

EDUCATION

University of California, Los Angeles (UCLA)

Spring 2026

B.S. Mathematics of Computation — GPA: 3.70

Relevant Coursework: Network/Graph Theory, Optimization, Numerical Analysis, Deep Learning, Algorithms/Complexity, Real Analysis, Probability Theory and Statistics, Signal Processing/Fourier Analysis, Cryptography

Pasadena City College

Spring 2024

Mathematics & Computer Science — GPA: 4.00

EXPERIENCE

UCLA, Mathematics Directed Research

2026 – Present

Researcher

- Conducting DARPA-backed research on inexact attributed subgraph matching over knowledge graphs. Developing and evaluating approximate subgraph isomorphism algorithms to identify laundering typologies in large-scale, noisy financial networks from a DARPA dataset. Directed by Prof. Paolo Climaco and Prof. Andrea Bertozzi.

Caltech, Lunar Trailblazer

2023 – 2024

Mission Operations and Ground Data Systems Intern

- Managed spacecraft simulated maneuvers and hardware/software troubleshooting for NASA-led lunar mission. Developed telemetry applications and dashboards using Python, InfluxDB, and OpenMCT — used by dozens of operators to monitor hundreds of telemetry streams in real time.
- Analyzed quaternion/spatial rotation telemetry for attitude determination and control system telemetry analysis, specifically to monitor Keep-Out Zone Violations of maneuvers.
- **Publication:** E. Scire, B. Ehlmann, ..., M. Mansour *et al.*, “Lunar Trailblazer Ground System Development,” *Proc. SPIE 13098*, 130981K (2024). [DOI: [10.1117/12.3027493](https://doi.org/10.1117/12.3027493)]

PORTFOLIO

SUM: Layer-1 Blockchain & Web3 Ecosystem ([SUMChain](#) / [SUMail](#) / [SNIP](#))

2025 – Present

- Co-designing a Layer-1 PoA/PoS blockchain entirely in Rust. Proposed a round-robin validator rotation for deterministic block production and protocol-level token standards.
- Fully architected a [Chrome Extension](#) using TypeScript for secure signing/authentication, featuring a private vault where keys exist only in memory after Argon2id decryption. Developed a React/Vite-based interface and a SwiftUI mobile wallet, integrating protocol-level endpoints to retrieve, verify, and display on-chain data directly from SUMChain.
- Storage Node Interface Protocol (SNIP): Designed and implemented a native decentralized file storage protocol in Rust with deterministic 3× replication, BLAKE3 content addressing, Proof-of-Retrievability challenges, parallel upload/download orchestrators, a self-healing MarketSync worker, and garbage collection. Built the P2P networking layer on libp2p: QUIC (UDP) as the default transport with TCP+Noise+Yamux fallback for Circuit Relay v2, AutoNAT for NAT-status detection, and Kademlia DHT for WAN peer discovery.

GNN for Ethereum Anti-Money Laundering Detection

2026

- Adapted IBM’s Multi-GNN architecture from edge-level Bitcoin AML classification to node-level Ethereum sanctions detection using transfer learning with extreme class imbalance (14,140:1 ratio) and weighted cross-entropy loss. [[training](#)]
- Engineered a Rust data pipeline with crash recovery to collect, label, and aggregate blockchain transactions into weekly directed graphs in Parquet format (90M+ transactions, 33M+ addresses). [[preprocessing](#), [modeling](#), [visualization](#)]

[Activation-Aware Pruning for Graph Neural Networks](#)

2026

- Adapted a weight pruning method built for large language models to graph neural networks, adding two variants that score weights by graph structure (node degree and class balance).
- Benchmarked 5 pruning methods across GCN, GAT, GraphSAGE, and GPR-GNN on 17 datasets at 10–90% sparsity using Python, PyTorch, and PyTorch Geometric. Neither structure-aware variant beat simple magnitude pruning.

[B-Plus-Tree SQL Engine](#) (C++)

2024

- Built a relational database engine from scratch — including a SQL parser (state machines implementation), a B-Plus-Tree indexing layer with binary file storage, and a query execution pipeline supporting SELECT, INSERT, and conditional filtering (Shunting-Yard/Boolean RPN).

Languages/Tools: C++, Rust, Python (PyTorch, NumPy, Pandas, scikit-learn, Matplotlib), SQL, Swift, JavaScript/TypeScript (React, Tailwind), Docker/Kubernetes, GCP (BigQuery, Dataflow, Cloud Composer), Git