

Nicholas Fidalgo

+1 (479) 531-3651 | nicom.fidalgo@gmail.com | Cambridge, MA

Relevant Links: [LinkedIn](#) | [GitHub](#) | [Google Scholar](#) | [Portfolio](#)

Programming: Python, C++, C, JavaScript/TypeScript, SQL, Java, C#, PyTorch, TensorFlow, NumPy, OpenCV

Technologies: React, Next.js, Node.js, Express, PostgreSQL, Supabase, MongoDB, Firebase, Docker, AWS, GCP, Azure, Git

Other: Stripe, Verilog, VHDL, WebSockets, Vercel, HTML, CSS | **Languages:** English (Fluent), Spanish (Intermediate)

EDUCATION

Harvard University

August 2022 – May 2026

- M.S. Computer Science | B.A. Computer Science (Honors) Cambridge, MA
- Affiliations: ColorStack, Hispanic Scholarship Fund, Harvard Tech for Social Good, Harvard Data Analytics Group

WORK EXPERIENCE

Netflix Playback Data Systems *Software Engineer Intern*

Los Gatos, CA | June 2025 – August 2025

Brill *Co-Founder*

Cambridge, MA | January 2025 – Present

- Co-founded an AI-powered SAT preparation platform offering personalized tutoring, adaptive practice, full-length Digital SAT simulations, serving 10+ paying customers with a 4.9/5 user rating.
- Developed full-stack web application using React, Next.js, Supabase, and Stripe to build dynamic, responsive interfaces with LaTeX math rendering, user authentication, payment processing, and multi-table statistics tracking.
- Integrated OpenAI and Gemini APIs to generate and validate 2,000+ questions and curate 15+ practice tests.

Massachusetts General Hospital *Data Scientist Intern* | [Publication](#)

Boston, MA | August 2023 – July 2024

- Applied PCA, z-score standardization, and KNN imputation reducing data dimensionality by 86% and retaining 90% of variance, accelerating model training by 300%.
- Developed 5 custom features boosting polynomial regression model accuracy by 70% up to 94% for predicting MRI lesion counts, enabling insights into GLP-1 agonists' effects on multiple sclerosis.
- Automated clinical data processing pipeline for 49 patient records from MGH's registry, integrating Python and SQL, co-authoring a peer-reviewed publication.

PROJECTS

Learned LSM-Trees with Predictive Filtering | [GitHub](#) | [Paper](#)

March 2025 – May 2025

- Integrated two novel ML-based enhancements to LSM-trees: a classifier to skip unnecessary Bloom filter checks achieving up to $2.28\times$ GET latency reduction and learned Bloom filters reducing memory usage by 70–80%.
- Built a full testing pipeline with realistic workloads and instrumentation to measure latency, memory, and error rates; trained gradient-boosted models with 45 engineered features for predictive query optimization.

Simulating Evolvability as a Learning Algorithm | [GitHub](#) | [Paper](#)

March 2025 – May 2025

- Implemented first large-scale empirical investigation of evolvability to simulate evolution as a constrained learning process across 6 Boolean function classes and first exploration of majority and general conjunctions/disjunctions.
- Discovered majority and general conjunctions/disjunctions can evolve under aggregate-only feedback (previously unknown in theory), validated theory knowledge such as evolvability in monotone case and non-evolvability of parity.
- First empirical investigation of different distributions, neutral mutation disallowance, and initialization constraints.

Scalable LSM-Tree Storage Engine | [GitHub](#) | [Paper](#)

March 2025 – May 2025

- Engineered high-performance LSM-tree in C++ using skip list memtable, Bloom filters with variable FPR, fence pointers, achieving sub-linear latency scaling from 100MB to 10GB datasets and up to 40% improved write throughput.
- Implemented novel hybrid compaction strategy combining tiering, lazy leveling, and leveling maintaining throughput under write-heavy workloads and improvements in 80/20 Zipfian skewed key and query distributions.
- Achieved near-linear scalability up to 16 threads and 32 concurrent clients, delivering $12\times$ latency reduction and $25\times$ throughput increase compared to single-threaded baselines, testing 8-dimensions of evaluation.

Multimodal AI for Forensic Sketch Generation | [GitHub](#) | [Paper](#)

November 2024 – December 2024

- Designed and implemented a multimodal pipeline integrating LoRA fine-tuned CLIP and Stable Diffusion v1.5, achieving 21% higher SSIM and 25% higher PSNR over baseline models in forensic sketch generation.
- Fine-tuned CLIP using LoRA on self- and cross-attention layers, improving text-sketch alignment by 9% and reducing perceptual error (LPIPS) over iterations validated through ablation studies.
- Developed iterative refinement process with dynamic embedding updates, enhancing quality across 5 refinement cycles.

Enhanced U-Net for Semantic Segmentation | [GitHub](#) | [Paper](#)

November 2024 – December 2024

- Engineered enhanced U-Net architectures with residual blocks, batch normalization, and dilated convolutions, improving segmentation accuracy by 3.6–4.7% over the baseline model on urban scene datasets.
- Developed hybrid U-Net-FCN model with skip connections, achieving 0.93 Dice for sky and 0.91 for road segmentation.
- Optimized loss functions using a combined Dice and Cross-Entropy approach, mitigating class imbalance and improving segmentation accuracy for irregular objects like sidewalks (18%) and buildings (37%).