# TAVE LA ROCQUE

Vancouver, BC | tave@tavelarocque.com | tavelarocque.com | github.com/tavelarocque | linkedin.com/in/tave-la-rocque

## SUMMARY

UBC engineering student on full-time research leave. I build systems whose results can be checked: hypotheses registered before the data that decides them, independent implementations that must agree, planted bugs that prove the tests can fail, runs that reproduce bit for bit, and formal proofs of hardware.

## EDUCATION

**University of British Columbia** | Vancouver, BC *Expected June 2028*
*Bachelor of Applied Science in Engineering | C Programming, Engineering Design, Mechanics, Calculus I–III, Linear Algebra*

## TECHNICAL SKILLS

**Languages: Python (NumPy, Pandas, SciPy, scikit-learn, TensorFlow, Keras, matplotlib, Streamlit), C++20, OCaml (Jane Street Core, dune, ppx_expect, Quickcheck), Rust (rayon, cargo), SQL, TypeScript, JavaScript, Java**
**Quantitative methods: SVI volatility smiles, Breeden-Litzenberger densities, Carr-Madan static replication, implied binomial trees, Black-76, Greeks under sticky-strike and sticky-moneyness, Avellaneda-Stoikov market making, markout / order-flow toxicity analysis, counterfactual regret minimization, exploitability and best-response analysis**
**Research discipline: pre-registration, locked holdouts, CSCV, deflated Sharpe, Holm / BHY multiple-testing corrections, bootstrap inference, property-based and known-answer testing**
**Math: Calculus I–III, Linear Algebra, MATH 152 (UBC Engineering Math), probability and statistics (HarvardX, online)**
**Software: React, Node.js, NestJS, REST APIs, Git, GitHub Actions, Supabase Postgres, Jupyter, dune, pytest**
**Enterprise & AI tooling: Excel (advanced, SUMIFS/SUMPRODUCT), Microsoft Copilot, Power Platform, Azure, Ollama (local model deployment), openpyxl, python-docx**
**Languages (spoken): English (native), Japanese, French (basic), Spanish (basic)**

## PROJECTS

**ITCH Volatility Replication | Pre-Registered Test of Maitrier & Bouchaud (2026) | osf.io/z4t7x** Oct 2026

- Ran a pre-registered out-of-sample test of a 2026 Quantitative Finance framework linking square-root impact, order imbalance and volatility, on 8 full NASDAQ trading days. Registered on OSF with the code frozen at a git tag before any held-out stock was touched.
- One Holm family of 8 tests: P-F is rejected in both tick-size bins, which is evidence against the informed-flow regime only; the other three predictions are not rejected. No result was computed before the registration closed.
- The P-F rejection is robust: negative in all five confirmatory liquidity cells and under four registered robustness arms. The confirmatory run reproduces bit for bit across 2,823 fields, with registered hashes verifiable in the public repository. Paper in preparation.

*C++20 · Python · OSF pre-registration · Holm correction · local Whittle · bootstrap / jackknife*

**itchbook | NASDAQ L3 Order Book Engine & Microstructure Studies | github.com/tavelarocque/itchbook** 2026

- Built a C++20 engine rebuilding NASDAQ's full order-by-order book from raw TotalView-ITCH 5.0 at 92 ns/message over 423M messages/day: 3.3x faster than CppTrader's L3 book on the same machine, with output byte-identical to an independent Python reference written so the cross-check could not share a bug.
- Showed that 85% of cancelled volume leaves from behind a resting order across 7.5M orders, so standard L2 queue models overfill backtests. Behind 10 to 24 orders, roughly 1 in 6 simulated fills never happened.
- Found DeepLOB beats an order-flow-imbalance baseline on macro-F1 (0.53 vs 0.45) across 7 out-of-time days, but no model clears costs: 0.1 to 0.4 ticks of edge against at least 1.6 ticks of spread and fees.

*C++20 · CMake · GoogleTest · RapidCheck · libFuzzer · ASan / UBSan / TSan · MIT, full history public*

**protocol-emulator | Programmable Protocol CPU, Jane Street ASIC Competition | github.com/tavelarocque/protocol-emulator** Nov 2026

- Designed a two-thread programmable protocol-emulator ASIC in Hardcaml (OCaml) for Jane Street's ASIC competition, targeting Tiny Tapeout's IHP 130 nm shuttle at 6x4 tiles. Protocols are firmware loaded over SPI after fabrication rather than fixed logic. Current layout: 18,817 standard cells, 28.3% utilisation, 0 DRC and 0 LVS errors, passes Tiny Tapeout precheck, timing met with about 80 ns of slack at the 10 MHz rating.
- Formally verified the whole chip pins-to-pins against an independently written specification, from every state, as a SAT equivalence proof with Hardcaml_verify and Z3. Separately proved in CI with Yosys that the post-layout gate netlist equals the RTL on every output pin from the first cycle after reset, by induction over 1,840 paired signals.
- Built a verification stack that catches all 58 planted bugs, kept as code so the build fails if any survives: lockstep differential testing against an independently written OCaml reference model over 400 random two-thread programs and 120,000 cycles with full state compared every cycle, coverage that fails the build on any empty bin, and 24 cocotb tests on both the RTL and the post-layout netlist against 7 third-party device models.

*OCaml · Hardcaml · Hardcaml_verify · Z3 · Yosys · cocotb · Icarus Verilog · Tiny Tapeout / LibreLane · static timing analysis · mutation testing*

**spr-calibration | Exact Minimax Values for an Open Problem in Online Calibration | github.com/tavelarocque/spr-calibration-code** 2026

- Computed exact minimax values of the sign-preservation-with-reuse game underlying the sequential-calibration bounds of Dagan et al. (STOC 2025), certified in both directions: lower bounds from exact rational strategy values, upper bounds from Farkas certificates.
- Wrote a 33-page paper on the result; a coauthor of the original bound engaged with the work and requested the code.

*Rust · Python · exact rational arithmetic · linear programming / Farkas certificates*

**Implied Odds | Prediction Markets vs Option-Implied Probabilities | github.com/tavelarocque/implied-odds-research** 2026

- Live research system comparing Kalshi S&P 500, Nasdaq-100 and Bitcoin contracts against option-implied probabilities, using Breeden-Litzenberger on arbitrage-checked SVI smiles with Deribit for Bitcoin. Runs 24/7 on GitHub Actions and paper-trades under fixed rules.
- Every test is pre-registered, judged on a locked holdout and adjusted with Holm/BHY. Kalshi's S&P contracts are well calibrated, and a Kalshi plus realized-volatility blend beat Kalshi's own price on the sealed holdout.

*Python · NumPy / SciPy · SVI / Breeden-Litzenberger · GitHub Actions · Holm / BHY*

**ocaml-contracts | Contract Language for Prediction Markets | github.com/tavelarocque/ocaml-contracts** 2026

- Implemented Peyton Jones, Eber & Seward's Composing Contracts to price Kalshi Bitcoin contracts from Deribit-implied densities. Three independent pricers agree with a Python reference to 2.7x10^-12 across 13,804 contracts.
- Includes Greeks, delta hedging, an Avellaneda-Stoikov market maker and markout analysis of order-flow toxicity.

*OCaml 5.4 · Jane Street Core · dune · ppx_expect · Quickcheck*

**UBC AISES InSSEP Team | SSEP Mission 22 Microgravity Experiment** Sept 2026 – Present

- Co-designed a protein crystallization experiment proposed for flight to the International Space Station, comparing lysozyme crystals grown by counter-diffusion in microgravity against gel and liquid ground controls.
- Wrote the proposal's materials and handling section to Rhodium Scientific and NCESSE hardware specifications, covering sample formulation, chamber volumes, thermal requirements and the crew interaction timeline.

*Protein crystallization · counter-diffusion · NASA JSC 26895 hazard assessment · Rhodium RhFET-01 · proposal submitted, not selected for flight*

## EXPERIENCE

**Vancity | Indigenous Project Assistant Intern, Corporate Real Estate** *May 2026 – Aug 2026*

*Selected 47 of 8,500 applicants (top 0.55%) for the inaugural Indigenous internship cohort. Audited a 14-tab legacy capital tracking system (500+ rows, 400+ GL codes) and designed a 4-layer modernization architecture for a $20M+ capital portfolio.*

- Rebuilt the Complaints Risk Classification System across 2,370 cases with a two-stage classify-and-validate architecture; the validation layer caught a 10x reporting error and an over-triggering keyword that had inflated one risk band by ~2,200 cases.
- Shipped an extraction-first agent workflow for complaint summarization validated against live closed cases, plus a data reconciliation MVP that distinguished complete, inferred and ambiguous sessions rather than silently filling gaps.

**MODUX | Co-Founder & Lead Engineer** *Apr 2026 – Present*

*Native iOS training app for hybrid athletes (modux.ca), submitted to Apple App Store, in review.*

- Solo-built end-to-end in 10 days: React Native 0.81 + Expo SDK 54 + TypeScript, Supabase Postgres with Row Level Security + Edge Functions in Deno, Strava + Apple HealthKit integrations.
- Designed the Interference Score system: an 11-dimensional fatigue vector with 72-hour decay modeling strength vs endurance training conflict, surfaced as a 0-100 readiness score (ACWR-based with confidence scoring) and TRIMP-based training stress separation for lift vs run load.
- Built per-muscle recovery map (front/back body diagram, react-native-svg) and cross-source data deduplication between Strava and HealthKit via fuzzy matching plus a UNIQUE PARTIAL index for DB-level enforcement.

**Current at UBC | Co-Founder & CEO** *Feb 2026 – Present*

- Co-founded a student-to-industry event running April 2027: 300+ students, 3 keynote speakers and 50 Vancouver professionals, with all net proceeds to BC Children's Hospital Foundation.

*Built and own the full production stack: currentatubc.com on Netlify, Zapier automation pipelines (form submissions to Google Sheets, Slack alerts, custom CRM logging), Cloudflare DNS, and end-to-end deployment and uptime.*

**Tave La Rocque Media (@tavelarocque) | Founder & Creator** **June 2026 – Present**

- Grew a short-form content brand from 1,040 to 200,000+ followers in five months across Instagram (100K), TikTok (50K) and Facebook (50K): 15+ videos past 1M views, 40+ past 100K, 11M+ peak.
- Ran it as a measurement loop rather than a feed: logged per-video metadata and retention curves into a structured dataset, tested one variable at a time (hook, first frame, pacing, caption) against watch-through, and cut skip rate from ~60% to sub-20%.

**KS Society | Director of Philanthropy** *May 2026 – Present*

- Promoted to Director after leading External Recruitment; own philanthropic strategy, partner outreach, and event programming for the society.

*Member of Kappa Sigma Fraternity; active in chapter philanthropy and external partnerships alongside a full engineering course load and two co-founded ventures.*