Satvik Saluja

Noida, India | satviksaluja2507@gmail.com | github.com/SatvikSaluja | Portfolio | LeetCode

Education

Jaypee Institute of Information Technology — Noida, India
*B.Tech Biotechnology, Minor in Artificial Intelligence & Machine Learning | CGPA: **9.02*** — *Expected 2028*

Technical Skills

Languages: C, C++, Python, JavaScript

Core CS: Data Structures & Algorithms, Object-Oriented Programming, Mathematical Modeling, Numerical Methods

Scientific / ML: Neural ODEs, PyTorch, torchdiffeq, PyTorch Geometric, NumPy, SciPy, ONNX Runtime

Web & Tools: React (Vite), Tailwind CSS, FastAPI, Git, GitHub, HTML/CSS

Projects

HNN Simulation Validation & Testing Infrastructure | *Python, NumPy, SciPy, pytest*

- Engineered a modular **validation pipeline** for HNN-style neural simulation outputs, separating execution logic from integrity checks to mirror HNN-Core's proposed testing architecture (Projects 15 & 16).
- Implemented **peak latency validation** and waveform integrity checks against reference baselines, enabling regression-style reproducibility testing across simulation parameter sweeps.
- Designed a **structured output schema** for simulation results, enforcing strict field contracts (timing, amplitude, synaptic weights) to facilitate automated cross-run comparison and CI integration.
- Decoupled synaptic behavior assertions into composable validator modules, enabling targeted unit testing of individual drive types (evoked, rhythmic, Poisson) independent of full simulation runs.
- Benchmarked numerical consistency of solver outputs across tolerance settings, confirming reproducibility of stochastic drive parameters under fixed RNG seeds.

Neural ODE Cognitive State Simulator | *PyTorch, torchdiffeq, FastAPI, NumPy*

- Built a **continuous-time dynamical system** $dS/dt = f(S, U, \theta)$ modeling EEG-derived cognitive states using Neural ODEs; parameterized with band-power features (delta, alpha, beta, gamma).
- Integrated **adaptive ODE solvers** (dopri5, RK45) via `torchdiffeq` with configurable step-size tolerances, validating solver convergence and numerical stability across trajectory lengths.
- Exposed **simulation endpoints** via FastAPI supporting parameterized queries over initial conditions and drive inputs, enabling systematic exploration of state-space dynamics.
- Structured the **modular simulation pipeline** into isolated stages (feature extraction → ODE integration → state decoding), mirroring reproducible scientific workflow design.

Full-Stack Metabolic Pathway Simulator | *PyTorch Geometric, GATv2, FastAPI, KEGG API*

- Modeled biochemical reaction networks (Glycolysis, TCA cycle, PPP) as **bipartite graphs** with GATv2 + FiLM convolutions over KEGG-sourced metabolite-enzyme topology for flux state prediction.
- Designed a **structured simulation output** contract (per-node flux estimates, attention weights, pathway tags) consumed by a React dashboard — enforcing reproducibility across KEGG data versions.
- Implemented graph-level **validation checks** comparing predicted flux distributions against literature reference ranges, exposing per-pathway deviation metrics via FastAPI structured responses.
- Separated pathway data ingestion, GNN inference, and output serialization into independently testable modules, enabling targeted regression testing of biological simulation components.

AirAware — Browser-Native AQI Inference | *ONNX Runtime, React (Vite), Tailwind CSS*

- Exported a trained AQI regression model to **ONNX** and deployed client-side inference via `onnxruntime-web`, eliminating backend latency while preserving numerical parity with Python outputs.
- Validated cross-environment output consistency (Python vs. browser ONNX) using structured numerical comparison, confirming sub-0.1% deviation across pollutant feature ranges.

Interests

Open Source: Active contributor mindset; drawn to scientific simulation codebases requiring rigorous numerical validation and modular refactoring — directly aligned with HNN-Core GSoC 2026 Projects 15 & 16.

Research Areas: Computational neuroscience, neural mass models, ODE-based dynamical systems, reproducible simulation pipelines, and testing infrastructure for scientific Python libraries.