

Ameya Panchal

(781) 870-7474 • ameyapanchalo11@gmail.com • ameya-bit.github.io • github.com/Ameya-bit • linkedin.com/in/ameya-bit

Education

The Pennsylvania State University

B.S. Computer Science, Minors in Physics and AI • Schreyer Honors College • GPA 3.71 / 4.00

University Park, PA

Expected May 2028

- Coursework: Linear Algebra, Probability, Algorithms, Discrete Math, Vector Calculus; now: AI, Neural Networks, Statistics

Publications

A. Panchal and A. ud-Doula. "Systematic Bias from Assuming Isotropic Surface Emission in Neutron-Star Pulse-Profile Modeling." *Astrophysics and Space Science*, under review.

Submitted Aug 2026

Research Experience

Do Residual-Stream Probes Survive Degrading Chain-of-Thought Legibility?

Supervised Program for Alignment Research (SPAR) • Mentee Researcher • mentor: Marios Tsatsos

Sep 2026 – Present

- To test whether activation probes still work when chain-of-thought turns illegible, sampled 1,816 Qwen3-4B traces on 400 borderline prompts (XSTest, OR-Bench) the model both complies with and refuses, so a probe must read the reasoning, not the prompt
- Labelled comply/refuse with two LLM judges (88% agreement, $\kappa = 0.76$), since keyword checks misread soft refusals

The J-lens offset is the model's token frequency: z-scoring helps [post]

Independent • PyTorch, jacobian-lens • Qwen3, Pythia, GPT-2, Gemma-2

Aug 2026 – Sep 2026

- Traced the per-token offset that buries the J-lens's readout of a model's activations to the model's own word frequency (Spearman 0.48); subtracting it, the obvious fix, makes readouts $3.5-7\times$ worse
- Z-scoring took secret-word elicitation from zero at every layer to the only working method on Qwen3-1.7B (pre-registered)
- Code: github.com/Ameya-bit/quantify-jlens-offset

A Dead-ReLU Absorbing State — Resolving an Open Memorization Anomaly [LessWrong]

Independent • PyTorch • LessWrong

Jul 2026

- Explained why one small transformer in a published memorization challenge could barely learn: its ReLU units died in training and cut off the gradient, a training failure, not a capacity limit
- Confirmed it with one controlled experiment (a bias after the hidden layer restored full capacity, 1024 facts; before it, none, 488) and hand-built weights storing 672 facts; *overturned* the authors' explanation
- Code: github.com/Ameya-bit/mem-toy-scratch

An Induction Head in Disguise — Circuit Analysis of a Character-Level Transformer [LessWrong]

Independent • PyTorch, TransformerLens • LessWrong

Jul 2026

- Trained a 6-layer character-level transformer on Nietzsche's prose and located the one attention head that copies earlier text forward
- *Ruled out* quote and parenthesis matching; the head has half a bracket-matching mechanism: what to write, not where to look
- Code: github.com/Ameya-bit/QuotesByNiche

Reading GPT-2's IOI Circuit Three Ways [writeup]

Independent • TransformerLens

Jul 2026

- Reproduced GPT-2 small's indirect-object-identification circuit (Wang et al.) with direct logit attribution, activation patching, and path patching, each implemented by hand
- Showed each method is blind to what the next reveals: attribution ranks name mover L9H9 first (+3.54) and misses S-inhibition head L8H6, which patching makes the brightest head on the grid
- Isolated the L8H6-to-L9H9 edge with path patching (L9H9's query alone recovers 0.10 of the effect) and confirmed both roles in the attention patterns
- Code: github.com/Ameya-bit/IOI

Systematic Bias from Assuming Isotropic Surface Emission in Neutron-Star Pulse-Profile Modeling [poster]

Penn State Multi-Campus REU • advised by Prof. Asif ud-Doula • Python

Oct 2025 – Aug 2026

- Simulated X-ray photons scattering through a neutron star's atmosphere to measure the error a common assumption introduces
- Validated it against an exact analytic solution (reduced $\chi^2 = 0.70$) and published waveforms (0.11% peak error)
- Measured a +0.14 pulsed-fraction bias for PSR J0740+6620, then showed the error relocates rather than vanishing: into waveform shape when geometry saturates the pulsed fraction, and again when spin shrinks it

- Groundwork: wrote 1D and 3D Monte Carlo photon-transport codes and trained PyTorch neural-network emulators to stand in for simulation runs
- Code: github.com/Ameya-bit/mc-radiative-transfer, github.com/Ameya-bit/Monte-Carlo

Presentations

R. Havre, A. Panchal, and A. Tang. "CrashAI: A Crash Map That Explains Itself." Poster, Penn State AI Showcase, Innovation Hub, Penn State. Sep 2026

A. Panchal and A. ud-Doula. "Where the Error Hides: How a Common Assumption Biases Pulse Fraction for Neutron Stars." Poster, Penn State Multi-Campus REU Research Exhibition, online. Jul 2026

Honors & Awards

Dean's List, every semester Fall 2024 – Spring 2026

Penn State Multi-Campus REU Scholarship (\$5,000) Summer 2026

1st of 40 teams, Nittany AI Challenge 2026

Teaching

Penn State Department of Physics University Park, PA
Physics Learning Assistant • PHYS 211 Aug 2025 – Dec 2025

- Taught calculus-based mechanics in problem sessions, labs, and exam prep

Experience

Nittany AI Advance Internship Program State College, PA

Application Specialist • Team of 5 • Client: Penn State Office of the Physical Plant Dec 2025 – Apr 2026

- Built a tool that pulls records (project name, ID) out of scanned documents using text recognition (Tesseract OCR), an LLM, and regex
- Processed 1,000 of the client's documents, speeding up manual data entry with a person still reviewing the results
- Raised accuracy up to 10% by merging LLM and regex results and benchmarking GPT-4 vs. GPT-5 on Azure; ran 5× faster in parallel

Projects

CrashAI — Crash-Risk Analysis Platform

Founder & Technical Lead • Team of 3 • 1st of 40 teams, Nittany AI Challenge 2026 • Scoped with PennDOT Feb 2026 – Present

- Built the machine-learning crash-severity model (XGBoost, 81 road and driver features) on 110,813 Pennsylvania crashes from 2024, merged from eight PennDOT datasets, with per-site SHAP explanations and what-if simulation
- Validated it against multinomial logistic regression in leakage-controlled five-fold cross-validation: lower Brier score in every fold (0.5992 vs 0.6031), killed-or-serious-injury AUC up from 0.813 to 0.822
- Led a team building the React / TypeScript frontend and Python services (33-tool agent loop, retrieval over engineering manuals, eval harness), served by a FastAPI backend with sub-5 ms road-data lookups

Peira — An Agent-Native Instrument for Benefit Cliffs ^[GitHub]

Solo • OpenAI WebMCP Challenge entry • TypeScript, React, Zod • Python, FastAPI Aug 2026 – Sep 2026

- Built a web app where a person and a ChatGPT agent investigate US benefit cliffs together: 11 agent-callable tools to sweep income, trace the rule behind a cliff, compare households, and find the smallest policy fix
- Kept each tool reply to ~1 KB of derived numbers; Zod validation on every input, a server-side whitelist for policy edits, rate limiting; 21 backend tests and an end-to-end script; live on Netlify and Render

Skills

Machine Learning: Python, PyTorch, HuggingFace Transformers, TransformerLens, NumPy, scikit-learn, Monte Carlo simulation, Git

Mechanistic Interpretability: Circuits, OV/QK analysis, activation and path patching, direct logit attribution, Jacobian and logit lenses

Languages & Frameworks: Python, TypeScript / JavaScript, Rust, Java, SQL • React, Vite, Node, FastAPI, Flask

References

Available on request.