

CHANDRACHUD PATI

chandrachudp@iisc.ac.in · +91 7606844339 · Bengaluru, India

RESEARCH INTERESTS

Quantum compiler design and noise-adaptive compilation; fidelity analysis for NISQ hardware; hybrid quantum-classical algorithms; quantum computer systems and architecture; full-stack co-design bridging algorithm design and fault-tolerant deployment.

EDUCATION

National Institute of Technology Rourkela · Odisha, India

2019 – 2024

Integrated M.Sc. in Mathematics ·

CGPA: 7.72 / 10

Relevant coursework: Quantum Computation, Quantum Mechanics, Probability & Statistics, Linear Programming, Numerical Analysis, Stochastic Processes, Graph Theory, Number Theory

PUBLICATIONS

[1] The HBR Decomposition: Per-Phase Fidelity Attribution for Quantum Compiler Pipelines

Chandrachud Pati, Yogesh Simmhan. *Under review, IEEE Quantum Week (QCE) 2026*.
arXiv:2605.07876

[2] Fraud Detection using Hybrid Classical–Quantum Machine Learning

Chandrachud Pati. *Preprint, 2023*. DOI: 10.13140/RG.2.2.14787.92963

RESEARCH EXPERIENCE

Indian Institute of Science (IISc) · Bengaluru

Feb 2026 – Present

Project Associate – Quantum Computing · Supervisor: Prof. Yogesh Simmhan

- **HBR Decomposition (NQM-aligned):** Designed and implemented the HBR (High-level, Basis, Routing) decomposition framework to attribute per-phase fidelity loss across quantum compiler pipelines; results under review at IEEE QCE 2026 (arXiv:2605.07876).
- **Compiler fidelity analysis:** Evaluated fidelity degradation at each compiler stage — gate decomposition, basis translation, and qubit routing — for eight canonical quantum algorithms across three SDK stacks (Qiskit, PennyLane, TKET); work is aligned with India's National Quantum Mission.
- **Benchmark study:** Conducting a structured literature review and experimental scoping of GPU-accelerated noisy circuit simulation for multi-platform benchmarking; benchmarks on IBM Heron and IonQ Forte architectures in progress.

NIT Rourkela · Odisha

Sep 2023 – Jan 2024

Research Project – Hybrid Quantum-Classical Machine Learning · Supervisor: Prof. Snehashish Chakraverty

- **Architecture:** Designed a Hybrid Quantum-Classical Neural Network (HQNN) integrating 2-qubit parameterized quantum circuits (Qiskit + PennyLane) as layers within a Keras deep learning model.
 - **Results:** Applied the HQNN to financial fraud detection on a highly imbalanced credit card dataset; achieved AUC of 0.9433, outperforming a classical neural network baseline by ~2%.
 - **Key insight:** Observed that circuit fidelity under transpilation had a significant effect on model accuracy, motivating subsequent work on compiler-level fidelity analysis.
-

INDUSTRY EXPERIENCE

American Express · Gurugram, India

Aug 2024 – Jan 2026

Data Engineer II

- Built a GenAI-powered data architecture documentation system using LLM-based pipelines, reducing manual documentation overhead by ~60%.
- Engineered REST APIs in Python for infrastructure provisioning; worked with distributed data pipelines at enterprise scale.

IDFC First Bank · Mumbai, India

Feb 2024 – Jun 2024

Research Intern – NLP / ML

- Fine-tuned a BERT model for real-time transaction classification, achieving ~99% accuracy on an internal test set.
- Built ETL pipelines using PySpark and AWS to ingest and preprocess large-scale transaction records.

AWARDS & FELLOWSHIPS

- **ACM SIGMOD/PODS 2026 India Fellowship**
- **Winner** American Express Early Careers Hackathon 2025 (UK, US, India) — Synthetic Data Generation using GANs
- **Semi-finalist** Flipkart GRID 5.0 — Detection of Diabetic Retinopathy using Quantum Computing
- **Runner-up** Tally CodeBrewers 2022 — System Programming for Tagging Files
- **All India Rank 4** Unstop UG Leaders 2023 (national hackathon & quiz competition)

TECHNICAL SKILLS

Quantum SDKs: Qiskit, PennyLane, TKET, Cirq

Compiler & Simulation: Quantum circuit transpilation (Qiskit, TKET), noisy circuit simulation, cross-platform benchmarking

ML / Deep Learning: PyTorch, TensorFlow / Keras, LLM fine-tuning (LoRA), RAG, GANs

Languages & Tools: Python (PySpark), SQL, Git, AWS, Docker / Kubernetes, LaTeX