

TEGANMOSIBINEBA 'TEGAN' JEGEDE

AI Safety Researcher | Evaluations & Empirical Failure-Mode Research

+234 813 488 1182 | tegan.jegade@yahoo.com | Abuja, Nigeria



EDUCATION

MSc Computer Science | Nile University of Nigeria, Abuja

Oct 2024 – Jul 2026

- Completed 2 July 2026. Thesis on evaluating and mitigating failure in agentic systems via reward design; empirical, multi-seed evaluation methodology.

B.Eng. Electrical & Electronics Engineering | University of Abuja, FCT

Nov 2015 – Feb 2022

PUBLICATIONS & TECHNICAL WRITING

- Jegade, T. O. (2026). *Sparse Binary Reward Degrades Agent Uncertainty Quantification in Multi-Seed Evaluation*. ICML 2026 Workshop on Statistical Frameworks for Uncertainty in Agentic Systems (AgenticUQ), Seoul. **[Accepted]** [\[OpenReview\]](#)
- Jegade, T. O. (2026). *Reward Instability in Binary Reinforcement Learning for Multimodal Language Agents: Evidence from Multi-Seed Evaluation*. **[Under review]**
- Technical writing:** Electricity Load Forecasting: AI vs Classical Regression, a clear technical explainer translating an empirical modelling comparison for a general audience. [\[Read on Medium\]](#)

EMPIRICAL EVALUATION EXPERIENCE

AI Researcher (MSc Thesis) | Nile University Abuja

Oct 2024 – Present

- Benchmark methodology:** Designed a novel evaluation showing that a widely used practice, single-seed reporting of RL-trained agents, is fragile and misleading: built the eval end to end, ran it across eight seeds, and demonstrated confidence intervals extending outside the feasible parameter space.
- Qualitative to quantitative:** Turned qualitative observations into quantitative measures: scanned model transcripts to identify degenerate failure patterns (incoherent token repetition), then operationalised them as cross-seed variance and confidence-interval metrics.
- Red-teaming:** Red-teamed my own results, isolating the mechanism (reward density collapse) through a five-configuration ablation, and stated limitations explicitly rather than overclaiming, the standard required for evidence a regulator or reviewer could act on.
- Engineering:** Built the full eval and training pipeline in Python (PyTorch, LLaVA-1.5-7B, LoRA, quantisation) against frozen benchmarks and analysed large, noisy multi-seed output sets on constrained hardware.

Research Intern (NLP & LLM Prototyping) | Nile University Abuja

Oct 2024 – Present

- Built LLM evaluation harnesses (LangChain, spaCy, NLTK) and ran systematic assessments of frontier models (GPT-4, LLaMA 2, Mistral), producing reproducible comparative results.
- Daily power-user of frontier models and coding agents (Claude Code / Codex-style workflows) to build and iterate on evals quickly.

SELECTED PROJECTS

- Agent tooling:** Autonomous job-application agent (LangGraph, Flask, SSE) with a real-time monitoring dashboard instrumenting a multi-step agentic workflow for live behavioural observation.
- Grounding:** Retrieval-Augmented Generation system (LangChain, Streamlit) grounding outputs in external knowledge to reduce hallucination.

AI SAFETY TRAINING & FELLOWSHIPS

- BlueDot Impact:** Technical AI Safety Project Fellowship (ongoing); completed the 30-hour Technical AI Safety course (Jun 2026), covering safety techniques, kill-chain threat modelling, and contribution planning.
- TRI AI Saturdays x Google DeepMind:** Participant, 16-week Google DeepMind AI Research Foundation course (Cohort 10).

TECHNICAL SKILLS

- Python, PyTorch, Scikit-Learn; eval harnesses, LLM APIs
- Evaluation: benchmarks, multi-seed CIs, ablations, failure-mode analysis
- Agents & LLMs: LangChain, LangGraph, LlamaIndex, Hugging Face
- **AI Governance:** Familiar with the EU AI Act, including its risk-based framework and Articles 5 and 92-93 enforcement provisions.
- Data analysis on large, noisy datasets; statistics, effect sizes
- Coding agents (Claude Code / Codex) as daily tooling
- Git, Jupyter, GPU workflows, experiment tracking

CERTIFICATIONS

- BlueDot Impact: Technical AI Safety (Jun 2026)
- Project Management Professional (PMP)
- Data Analysis Certification
- Python Programming

References available upon request.