John Olafenwa

Scientist & ML Engineer | Post-Training Adaptive Agentic Models

Website: https://johnolafenwa.com
Email: johnolafenwa@gmail.com
LinkedIn: https://uk.linkedin.com/in/olafenwajohn
GitHub: https://github.com/johnolafenwa
YouTube: https://www.youtube.com/@John.Olafenwa

About

I lead research on post-training adaptive agentic models for Microsoft Teams. At the core of my work is applying reinforcement learning with rubrics and on-policy distillation to train frontier models that learn from their interactions with people and adapt their behaviour through long-term memory and in-context adaptation. My focus is on efficient reasoning, tool calling, memory and value alignment.

Outside of work, I study the mathematics of deep learning, especially information theory and calculus, and share insights from my learnings on my blog and YouTube. I am an active coder and I love building things. Using a team of Codex and Claude Code agents, I built and currently maintain Aura, an open source Pythonic compiled language for high-performance ML systems.

Key Skills

- LLM Post-Training (RL, Rubric Generation from Values, Alignment and On-Policy Distillation)
- Large scale synthetic data generation
- Building Agentic systems (tool calling, reasoning, agent evaluation)
- Deep Learning Fundamentals (Scaling Laws, Network Efficiency, Regularization, Vector Quantization)
- Transformers (Dense and Banded Attention, Positional Encoding and MoEs)
- Experienced Python Developer with good knowledge of Python Internals
- PyTorch + Triton (Distributed Training, Mixed Precision Training, Kernel Optimization)
- Engineering Skills (Docker, Kubernetes, Redis, Linux)

Experience

Senior Scientist | Microsoft

Sep 2025 - Present

- Post-training frontier OpenAI and MAI models for Microsoft Teams. This involves working in large research codebases, designing rewards for various tasks including personality and safety, and training pipelines using novel post-training algorithms and frontier training compute infrastructure, large scale data preparation including both real and synthetic data and creating evaluation systems that align with product needs.
- Developed a novel training approach that converts text-only LLMs into native text + audio models by self-distilling the base model into a text + audio adapter variant.
- Co-developed advanced agents for Microsoft Teams, consulting with partner teams, delivering deep dives on tool calling systems and building reference implementations for sequential and parallel tool use.
- Built a reference RL post-training library with support for GRPO for post-training open source LLMs.

Scientist 2 | Microsoft

Sep 2023 - Sep 2025

- Led development of a team monorepo for training GPTs and working with agents; built inference and supervised finetuning frameworks for open source LLMs.
- Built data labelling and metrics tools for annotating evaluation data for Microsoft Facilitator agent. This was used to label real data, enabling evaluation of off the shelf and fine-tuned models on diverse real world scenarios.
- Developed multimodal fine tuning workflows, LLM as judge evaluation, synthetic data generation pipelines, and a single API abstraction spanning Azure OpenAI and HuggingFace.
- Incubated research prototypes that matured into standalone repositories adopted by partner teams.

- Led the transition of my wider science team from research in mixed reality to working on LLM training and agent building; ran learning sessions on transformers, multimodal models, fine tuning, pre-training, and using LLM APIs.
- Ran small scale pretraining experiments with 3B parameter model on up to 100+ GPUs to train language models from scratch. This included large scale real and synthetic data preparation, model architecture design and training with accelerate + deepspeed.
- Co-designed and implemented agents in Teams, working across prototyping, fine tuning, prompt design, and system evaluation.

Scientist I | Microsoft
Jun 2021 - Sep 2023

- Designed and trained the audio driven avatar animation model used in Microsoft Teams avatars; scaled training using horovod with pytorch for multi node and multi-gpu training and moved all data preparation to azure machine learning, greatly speeding up research iteration.
- Replaced spectrogram based CNNs with a lightweight 1D convolution architecture operating on raw audio, shrinking model size from several tens of MB to a few hundred KB while improving quality and meeting strict client side latency constraints.
- Built inference runtime in both python and javascript to accelerate product integration. The inference runtimes were based on onnxruntime.
- Led development and maintenance of post launch metrics infrastructure, enabling reproducible evaluation of errors, latency, and fairness metrics; I developed a new metric based on Pearson correlation between 3D face landmark points over an entire audio-video sequence, enabling us better measure lip sync quality.

Software Engineer | Microsoft
Jan 2020 - Jun 2021

- Worked on 3D object detection in the Azure Object Understanding team, contributing to Azure Object Anchors.
- Built data processing and training workflows and added support for mixed precision training using NVIDIA AMP.
- Initiated collaboration with the Cambridge Mixed Reality team on training models for avatar lip sync from audio signals.

CTO and Co Founder | DeepQuest AI
Aug 2018 - Dec 2019

- Led development of DeepStack, an on-device AI engine providing APIs for face detection, face recognition, scene recognition, and custom object detection. DeepStack was built using Go and Python.
- Built a high performance inference engine and trained custom models running across kubernetes and local devices with GPUs, CPUs, Raspberry Pi, NVIDIA Jetson (using TensorRT), and Intel Neural Compute Stick (using OpenVino).
- Leveraged off the shelf models and designed new variants optimized for low compute devices.
- Built and maintained the company's website and cloud infrastructure.
- Scaled adoption to 10M plus Docker Hub installs; featured by NVIDIA and Docker.

Presentations

- Deep Dive into Pretraining and RL Maths and Code from an Information Theory Perspective: https://www.youtube.com/watch?v=MaZWafi4gYY
- How SORA Works: https://youtu.be/IqZXkBjKb2E
- RL vs SFT: On-Policy vs Off-Policy Learning: https://youtu.be/JN_jtfazJic
- More on: https://www.youtube.com/@John.Olafenwa/videos

Select Articles

- The Log Trick: https://johnolafenwa.com/blog/the-log-trick/

- On the Subject of Thinking Machines: https://medium.com/data-science/on-the-subject-of-thinking-machines-c3ba65a7105
- Review of Deep Residual Learning for Image Recognition: https://medium.com/deepreview/review-of-deep-residual-learning-for-image-recognition-a92955acf3aa

OpenSource

AI Safety Atlas (2026)
https://aisafetyatlas.com
An open reference guide to AI safety and the techniques used today. Covers:
- Foundational concepts and the case for AI safety
- Value alignment, mechanistic interpretability, chain-of-thought monitoring and red teaming
- Curated papers, research organizations and people to follow
- A living document, open to community contributions on GitHub

Aura (2026)
https://github.com/johnolafenwa/Aura
A compiled language with Python-like syntax, built for scalable and highly performant ML systems. Features include:
- Compile-time ownership and mutability checking
- Generics, traits, pattern matching and structured concurrency
- A standard library spanning files, networking, JSON and TOML, and observability
- Native compilation with a formatter, test runner, language server and C FFI

RLFusion (2026)
https://github.com/johnolafenwa/RLFusion
A minimalistic LLM posttraining library supporting both training and evaluation. Initial set of features includes:
- Supervised finetuning with support for prompt masking
- Reinforcement Learning with verified rewards using GRPO
- On-Policy distillation using reverse KL divergence.
- General purpose evaluator, to run LLM evaluation with any benchmark.

TorchFusion (2018)
https://github.com/johnolafenwa/TorchFusion
Inspired by Keras for TensorFlow, I created TorchFusion as a general purpose training framework built on PyTorch. It provides several features to simplify training neural networks, filling in features that were missing from the official PyTorch API at the time, including.
- Initialisers and NN modules for 1d, 2d and 3D convolutions and pooling layers.
- Mixed precision training
- A general trainer API that allows configuring the step functions, callbacks and logging.
- Specialised trainers for a number of GAN algorithms.
- Dataset classes for a number of computer vision datasets

Publications

- YouTube: https://www.youtube.com/@John.Olafenwa
- Substack Blog: https://johnolafenwa.substack.com
- DeepReview: https://medium.com/deepreview
- Medium Blog: https://medium.com/johnolafenwa
- Introduction to Deep Computer Vision: https://github.com/johnolafenwa/DeepVision
- FastNet: https://arxiv.org/abs/1802.02186