# Haris Mahmood

AI Research Engineer

Islamabad, Pakistan | +92 335 5065740 | haris@harismahmood.dev
harismahmood.dev | linkedin.com/in/hmahmood24 | github.com/hmahmood24

## Summary

AI research engineer with six years of work across document understanding, machine learning framework transpilers, and LLM agents, now focused on post-training, continual learning, and agent harnesses. Most recently at Unify (YC W23), built much of the platform under its AI teammates, from the RAG pipeline to generative UI and app integrations, and led a ground-up redesign of its open-source agent harness, after co-founding Ivy and leading its source-to-source ML transpiler, which Kornia merged upstream.

## Experience

### Unify (YC W23)
Jan 2025 – Oct 2026

*Founding AI Engineer* — *Remote*

Startup that built Unify Interfaces, an observability and evaluation platform for LLM applications, then Unify Teammates, AI agents that companies onboarded like new employees and worked with over chat, email, and live calls.

- Led a ground-up redesign of Unify's open-source agent harness around one code-execution tool: on Continual-ARC, an unpublished continual-learning benchmark, it solved more puzzles at every model reasoning effort than the old harness and than OpenClaw, Hermes Agent, and Prime Agent.
- Redesigned the agent's memory for both memoization and generalisation: the agent wrote up each task as a Python function, and an agent mesh of a writer, a librarian, and sandboxed sub-agents kept the library.
- Built the RAG pipeline behind Unify Teammates: documents chunked along their own headings, tables kept as typed data the agent queried in code, and ingestion on GKE workers that scaled with the Pub/Sub backlog.
- Built generative UI for the agents: from a plain-English request a teammate wrote a React app on a component kit built for it, checked it in a real browser, and wired actions such as emailing a report, inside a sandbox where the server validated, confirmed, and ran each action exactly once.
- Built the agents' OAuth and API-key app integrations with progressive disclosure: agents searched for apps and tools from their own code, loaded each schema only when calling it, and every call passed per-account permissions that allowed, asked about, or blocked it.
- Built, tested, and rolled out installable agent workflows that switched on once a customer connected the apps they needed, and let a recurring task be distilled into Python that called a model only for its judgement steps.
- Built the Next.js front end of two products: the tables and dashboard editor through which teams explored their LLM traces and evaluations in Unify Interfaces, and the Teammates app shell that kept chat, live activity, and calls running across navigation.

### Ivy (Unify spin-out)
2024 – Jan 2025

*Co-founder and CEO* — *Remote*

Company built around an open-source, source-to-source ML transpiler that rewrote PyTorch code as TensorFlow and JAX.

- Co-founded Ivy with colleagues from Unify and led it as CEO and technical lead, running technical demonstrations for enterprise machine learning teams and setting the roadmap from what their models needed.
- Led its source-to-source ML transpiler, rebuilt at the spin-out on an architecture first laid down at Unify, which rewrote PyTorch models and libraries as readable TensorFlow or JAX with their control flow intact, so a PyTorch model could run on Google's TPUs as editable JAX source.
- Built its translation cache, which stored translations as serialisable syntax trees, and its handling of PyTorch's in-place operations in immutable TensorFlow and JAX, and validated converted Hugging Face models by loading their weights, comparing outputs, and training them in the target framework.
- Brought it to Kornia, the PyTorch computer-vision library: about 90% of a large set of its functions and classes transpiled, and the maintainers merged the integration and shipped it in Kornia 0.7.4.

### Unify (YC W23)
Mar 2023 – Apr 2024

*Machine Learning Research Engineer II* — *Remote*

Unify was then building Ivy, an open-source framework for making machine learning code portable across frameworks.

- Led the team building Ivy's graph tracer and transpiler, which traced a model into its computation graph and rebuilt it in another framework, and made traces carry Python values as well as tensors, so shapes such as `x.shape[0]` stayed dynamic instead of being frozen into constants.
- Extended the Ivy framework that served as the common representation between frameworks: its PyTorch,

TensorFlow, and JAX frontends, the backends beneath them, and its stateful API, where converted PyTorch modules became trainable Keras models, tied weights and LSTMs included.
- Adapted a fork of TensorFlow's AutoGraph so built-ins such as `len` appeared in the trace, made `jax.jit` work on transpiled Hugging Face models, and cut transpile time on very large graphs, such as FasterRCNN to JAX, from about 65 to about 25 minutes.

**CureMD** 2020 – Mar 2023
*Machine Learning Engineer* *Lahore, Pakistan*
Healthcare software company.

- Built SmartRecruit with one other engineer before large language models: a recruitment system, used for CureMD's own hiring, that parsed resumes of any layout with over 90% accuracy and parsed and scored each in 2–3 seconds, against about three minutes for a recruiter to read one.
- Linked each extracted entity to its education or work entry by drawing the entities onto the rendered page and detecting entries with a custom YOLOv5 model, so dates matched the right employer even in multi-column layouts.
- Chose headings by a vote among twelve candidate segmentations and recovered columns and reading order from them, then scored candidates with an explainable model over O*NET-normalised roles and about 13,000 skill terms.

**N-ovative Health Technologies** Jul 2020 – Sep 2020
*Embedded Developer* *Islamabad, Pakistan*

- Designed the electronic circuit and control system of the NUST Powered Air Purifying Respirator and worked on the software and hardware interfacing of the N-Saviour ventilator during the COVID-19 equipment shortage.

## Education

**National University of Sciences and Technology (NUST)** 2016 – 2020
Bachelor's in Electrical Engineering Islamabad, Pakistan
CGPA 3.69/4.00

## Selected Open Source

- Unify agent harness: open-source harness in which agents act by writing code; led its ground-up redesign, with an agent mesh and a memory built for memoization and generalisation.
- Ivy: the transpiler's source, from the traced-graph version to the source-to-source rebuild, published on PyPI.
- Kornia: co-authored the integration, merged in October 2024, that gave Kornia TensorFlow, JAX, and NumPy support through Ivy's transpiler.
- TensorLy: co-mentored the Google Summer of Code 2023 project to make TensorLy usable with every framework Ivy supported, by switching its backends through Ivy.

## Technical Skills

**Languages:** Python, TypeScript, SQL, Bash
**Machine learning:** PyTorch, TensorFlow, JAX, Keras, Hugging Face Transformers, PEFT (LoRA), vLLM, YOLOv5, spaCy, HunFlair, scispaCy
**LLM agents:** Code-executing agents, multi-agent systems, agent memory, RAG, tool calling, prompt caching, sandboxing (bubblewrap, seccomp), OpenAI and OpenRouter APIs
**Program transformation:** Python AST transformation, static analysis, graph tracing, TensorFlow AutoGraph
**Documents and data:** Docling, PyMuPDF, Polars, openpyxl, OpenCV
**Backend:** FastAPI, Flask, PostgreSQL (query planning, JSONB), SQLAlchemy, Pydantic, MongoDB, TypeDB, RabbitMQ
**Frontend:** Next.js (App Router), React, generative UI, TanStack Query, TanStack Table, Zustand
**Cloud and infrastructure:** Google Cloud (GKE, Pub/Sub, Cloud Run, Cloud Build), AWS (ECS, S3), Kubernetes, Docker, MLflow

## References

**Daniel Lenton**, Director and CEO, Unify AI Ltd dan@unify.ai