Arunjay Agrawal

Madison, WI | agrawal54@wisc.edu | linkedin.com/in/arunjay-agrawal-aab235322 | github.com/Arunjay4213

Education

University of Wisconsin-Madison **Madison, WI**
B.S. Computer Science, B.S. Mathematics, B.S. Data Science | GPA: 3.82/4.0 *Expected May 2028*

Experience

AI Engineer Intern
Jun 2026 - August 2026
RIA Advisory *Miami, FL*

- Built an agentic AI system that converts unstructured client inputs into cited business requirements documents (BRDs) for an enterprise healthcare-payer Oracle ORMB implementation, cutting consultant **time-to-first-draft by 60%**.
- Architected a scoped GraphRAG retrieval layer over Qdrant and Neo4j indexing **4,000+ pages across 7 knowledge bases**, filtered by product, version, and functional-area metadata.
- Engineered a Docling ingestion pipeline that parses heterogeneous PDFs into typed structure, routing prose, tables, and figures to type-specific chunkers with vision-model figure descriptions, glossary extraction, and cross-document linking.
- Orchestrated a multi-agent LangGraph pipeline with an **LLM-critic loop** that revises drafts until they pass automated faithfulness checks against cited sources, plus MCP-exposed agent tools and human-in-the-loop review.

Software Engineer Intern
Feb 2026 - Present
UW-Madison Tech Exploration Lab *Madison, WI*

- Built a bilingual (Spanish/English) veterinary AI voice agent in Go serving **150+ daily queries** for a 10-veterinarian pilot across Wisconsin dairy farms, using few-shot prompts, strict non-diagnostic guardrails, and a 50-case safety test suite.
- Created a full-stack lab management dashboard (Python, JavaScript, PostgreSQL) on GCP with JWT authentication, automating mentor matching and resource allocation across **30 active ventures**, replacing a manual spreadsheet workflow.

Undergraduate Research Fellow
Feb 2026 - Present
UW-Madison Data Science Institute, Wallace Foundation $102M Initiative *Madison, WI*

- Designed data pipelines that transform coded qualitative transcripts into Social Network Analysis (SNA) adjacency matrices, enabling network-level collaboration insights across **8 urban school districts**.
- Built an interactive browser-based graph explorer (Cytoscape.js, D3.js, NetworkX) for SNA knowledge graphs supporting up to **1,300 nodes** with force-directed layout, multi-dimensional filtering, and neighborhood exploration.

Software Engineer Intern
May 2025 - Sep 2025
LogicPatterns *San Francisco, CA*

- Built a full-stack AI agent that converts natural language prompts into live software architecture diagrams, cutting diagram creation time from **30+ minutes to under 2 minutes**.
- Containerized the backend as FastAPI microservices with Docker, wrote Pytest tests covering inference, diagram synthesis, and frontend rendering, and optimized SVG rendering via a JSON-to-React bridge, cutting render latency to **1.3 seconds**.

Projects

Fused CUDA Kernels for Linear Attention | *CDIS & OpenAI Summer AI Lab*
Jun 2026 - Present

- Reverse-engineered the RACE Attention reference implementation (ICLR 2026). Found its shipped CUDA kernels **unused at runtime**, with the real bottleneck a PyTorch `cumsum` + matmul path materializing 5-D intermediate tensors.
- Engineering a fused CUDA forward kernel (shared-memory state, coalesced bf16 I/O, no atomics) targeting **2M-token forwards at 80%+ HBM bandwidth** on one H100, 15x past the reference's 131K wall.

Tripwire - Prompt Injection Evaluation Harness | *Open Source*
Jun 2026 - Present

- Led a team building a harness that measures whether tool-using AI agents leak planted secrets to attackers; published to PyPI (`tripwire-eval`) with a **351-test suite** and CI enforcing an attack-success-rate release gate.
- Built the harness so developers test any agent (raw loops, LangGraph, CrewAI) by writing one small adapter; implemented **13 attacks**, including adaptive ones that refine their injection based on the agent's responses.

Hybrid RAG Retrieval System | *Kaggle Competition, UW-Madison Environmental AI*
Oct 2025

- Designed a hybrid RAG retrieval system combining BM25 and dense embeddings (BGE-large) via reciprocal rank fusion, achieving **87% accuracy on 282 test queries** with only 40 training samples.
- Parsed 32 research PDFs with complex tables, normalized 15+ scientific units (CO2, kWh) into a consistent schema, and built the dense retrieval layer with cached embeddings and a persisted HNSW index with exact-search fallback.

Technical Skills

Languages: Python, TypeScript, JavaScript, Java, Go, C, CUDA, SQL, Bash
AI/ML: Machine Learning, PyTorch, RAG, GraphRAG, LangGraph, MCP, LLM APIs, Prompt Engineering
Data & Databases: Qdrant, Neo4j, PostgreSQL, MongoDB, Vector Databases, Knowledge Graphs
Frameworks & Backend: FastAPI, Flask, Node.js, React, REST APIs, WebSockets, Microservices
Infrastructure: Docker, Kubernetes, GCP, AWS, CI/CD (GitHub Actions), Git, Pytest