David Antolick

(570) 926-0234 | david@antolick.ai | antolick.ai | linkedin.com/in/david-antolick | github.com/David-Antolick

Summary

AI/ML engineer building production agentic systems, RAG pipelines, and LLM evaluation infrastructure. Independently built and shipped a full-stack RAG platform with a custom ReAct agent, hybrid retrieval, and eval-driven iteration achieving 4.86/5.0 correctness across 230 benchmarked questions. Currently deploying multi-agent chatbots and data pipelines in regulated pharma manufacturing at J&J. MS in Computational Biology with hands-on protein language models, distributed GPU training, and molecular ML.

Technical Skills

AI/ML: Agentic AI (ReAct, tool calling, multi-agent orchestration), RAG (hybrid retrieval, BM25, RRF, reranking), LLM evaluation (LLM-as-judge, rubric scoring), prompt engineering, knowledge graphs, question generation pipelines
Languages: Python, TypeScript, JavaScript, SQL, Go (Hugo templates), HTML/CSS
Data Engineering: PySpark, Databricks, Delta Lake, Pandas, NumPy, ETL pipelines, JSON/Parquet data modeling
Backend: FastAPI, PostgreSQL, Pydantic, REST APIs, SSE streaming, OAuth/session auth, async Python
ML/DL: PyTorch (DDP, TorchScript), XGBoost, ESM-2, Hugging Face Transformers, scikit-learn, OpenMM
Frontend: Next.js 16, React 19, Tailwind CSS, SWR, SSE parsing, responsive/mobile-first design
Infrastructure: Docker, AWS (S3, CloudFront), Azure Blob Storage, Chroma, vLLM, Git, Linux, CUDA

Experience

ClimbSpeed — Independent Engineer Aug 2025 – Present
climbspeed.com — Production RAG platform for aviation ground school *Public launch April 2026*

- Built a production RAG-powered aviation ground school platform end-to-end as a solo project: 15,000+ lines Python, 6,300+ lines TypeScript, 750+ lines Jinja2 prompt templates, 9 PostgreSQL tables with migrations, 3-container Docker Compose architecture. Completed alpha testing with real student pilots; feedback drove the entire V3 roadmap.
- Designed a custom ReAct agent framework from scratch (no LangChain/LlamaIndex) with an abstract `BaseAgent`, tool-calling loop, state management, and extensible hook system. Iterated from a complex validation agent (78–86% pass) to a minimal 2-tool search+submit design (93–95% pass) — eval data proved measurement-driven simplification outperforms over-engineering.
- Implemented hybrid retrieval: semantic search (Chroma + SentenceTransformers) + BM25 keyword search + Reciprocal Rank Fusion, with agent-controlled multi-round search, parallel queries in a single LLM response, per-question chunk deduplication via content hashing, and round limits tuned by eval data (2-round answers scored 4.82/5 vs 5-round at 4.73/5).
- Built deterministic compute tools (aviation calculator with 9 functions, FAA calendar with 7 functions, 75 unit tests) to fix systematic LLM math failures identified in alpha audit. Tools run in a restricted Python sandbox. System prompt enforces mandatory tool use for numeric and date questions.
- Developed LLM-as-judge eval pipeline with producer-consumer ThreadPoolExecutor architecture, calibrated judging with full untruncated context pass-through (discovered truncating judge context caused 93.9% → 97.3% false-failure swing). Benchmarked at 4.86/5.0 correctness, 99.6% pass rate, 100% citation compliance across 230 questions.
- Engineered a 7-stage passage-grounded question generation pipeline: deterministic topic planning via concept graph, RAG-grounded generation from actual FAA text, 5-check hard verification gate with deterministic compute cross-checks, misconception-informed distractor overgenerate-and-rank, embedding-based semantic dedup (0.85 cosine threshold), and MCQ review. ˜2,900 V2 questions deployed; V3 design grounded in 6 peer-reviewed papers.
- Processed 45+ FAA publications into a searchable knowledge base using Docling hybrid chunking (structure-aware PDF parsing), sliding window tokenization (512 tokens, 128 overlap), and dual indexing into both Chroma (vector) and BM25 (sparse). Hand-curated supplemental FAQs for content that chunks poorly from PDFs.

AI and Platform Engineer Sep 2025 – Present
Johnson & Johnson — Carvykti CAR-T Cell Therapy Manufacturing *Remote*

- Designed and deployed a multi-agent chatbot on J&J's internal AI platform (Flowwise/AMP) enabling scientists and engineers to query lentiviral manufacturing data via natural language. Built a Claude 3.5 Sonnet routing agent with domain-aware routing rules, a custom JavaScript SQL executor against a Databricks SQL warehouse, safety guardrails (SELECT-only validation, auto LIMIT 1000), and async query polling.

- Built a PySpark data pipeline ingesting 17 heterogeneous parquet datasets from Azure Blob Storage across two manufacturing sites, normalizing schema conflicts via `unionByName(allowMissingColumns=True)`, constructing a material genealogy graph with iterative DFS (stack-based, memoized), and producing 4 structured JSON knowledge layers + 7 Delta Lake tables powering both the chatbot SQL interface and the Hugo portal.
- Developed a 3,900-line CrispML visualization pipeline transforming predictive ML model outputs (52 iterations across 6 rounds, 14 CQA outcomes, 7 model types) into hierarchical JSON with radar charts, scatter-regression plots, fishbone diagrams, and bi-directional bar chart visualizations. Built S3 discovery with multi-level fallback logic and a unified column resolver handling cross-round naming inconsistencies.
- Architected a tri-agent risk management chatbot (GPT-4o-mini) replacing traditional static intake forms with an AI-guided conversational interview. Built a 3-phase forward-only system (Intake → Assessment → Wrapup) with vector store RAG for similar-risk calibration from a 57-record risk register, 5×5 severity/likelihood heatmap scoring, and CEI statement generation.
- Built an automated chatbot evaluation harness scoring quality across 5 dimensions (correctness, grounding, completeness, tone, safety) with parallel ThreadPoolExecutor execution, progressive saving, graceful interrupt recovery, and automated 6-chart report generation.

Projects

REX Voice Assistant | *Python, Whisper, Silero VAD, FastAPI, asyncio, CUDA* 2024 – 2025

- Built a fully local streaming voice assistant for hands-free desktop control with sub-second perceived latency. Dual-mode audio pipeline: standard mode (Silero VAD end-of-speech detection) and low-latency mode (early intent matching on partial Whisper transcripts with safe/unsafe early-match gating). Integrated Spotify (OAuth), YouTube Music Desktop (companion API), and SteelSeries GG Moments (hotkey simulation).
- Shipped as an installable CLI tool (`pyproject.toml` entrypoint) with keyring-based secret storage, layered configuration (defaults → user overrides → env vars), latency/match-rate instrumentation via a FastAPI metrics dashboard with WebSocket streaming, and explicit Windows CUDA/cuDNN path handling.

Intron Retention Analysis Pipeline | *Python, Pandas, Biopython, Ensembl REST API, Docker* 2024 – 2025

- Built a multi-stage bioinformatics pipeline in the Robin Lee Lab (University of Pittsburgh) for downstream analysis of IRFinder output. Processes ˜300K rows/sheet across 4 experimental timepoints, enriches with Ensembl genomic annotations via batch REST API and BioMart queries, reconstructs intron-retained transcripts, and predicts NMD susceptibility using EJC distance rules.
- Designed modular OOP architecture with method-chaining fluent API (`SequenceProcessor`), pickle-based caching layer (50x load speedup), multi-replicate consensus filtering, strand-aware coordinate transforms, and publication-quality visualizations.

Education

University of Pittsburgh Pittsburgh, PA

M.S. Computational Biomedicine and Biotechnology *Aug 2024 – May 2025*

- Drug-kinase binding prediction: ESM-2 protein language model embeddings (3B parameters) + XGBoost with Bayesian HPO via scikit-optimize. Independent Spearman 0.624, ROC AUC 0.759. ESM-2 embeddings outperformed hand-crafted k-mer features.
- Molecular generation: SMILES variational autoencoder (GRU encoder/decoder, reparameterization trick, 1024-dim latent). TorchScript decoder export for dependency-light sampling.
- Scalable microscopy classification: ResNet-34 for 13 phenotype classes with DDP multi-GPU training (torchrun/NCCL), synchronized BatchNorm, and TorchScript export. Additional projects: protein-ligand MD (OpenMM, CUDA), variant effect prediction (ESM-2 + BLOSUM62, $R^2 = 0.73$), genomic motif prediction (HOMER PSSMs + XGBoost).

Rensselaer Polytechnic Institute Troy, NY

B.S. Computational Biology, Minor in Philosophy *Aug 2020 – May 2024*