Derek Deming

Email: derekdeming17@gmail.com
Website: http://derekdeming.io/ **Phone:** 760-525-0871
Github: https://github.com/derekdeming **LinkedIn**: https://www.linkedin.com/in/derek-deming/

Education

PhD in Computational Chemical Biophysics (dropped out to pursue tech career)
MS in Chemistry – *Emphasis in Computational Methods & Machine Learning*

- PNNL-WSU Distinguished Graduate Research Program Fellow
- Research Assistant: Radioactive Material and Engineering Fellowship @ Department of Energy (DoE)
- Research Gate: https://www.researchgate.net/profile/Derek-Deming-2

BA in Biology and Chemistry – *Emphasis in ML applied Chemical Biology Research*

- Bioinformatics Research Fellow at the Orthopedic Surgery Specialty Clinic
- Distinguished Presidential Scholar Undergraduate Research Fellow

Core Competencies

Languages: Proficient in C, C#, C++, CUDA, Python, Golang (GO), JavaScript, TypeScript, KQL, SQL, SCOPE, GraphQL, Cypher, TLC, YAML, and Bash
Compiler Expertise: LLVM, MLIR, TensorRT, CUDA, ONNX Runtime, TVM

Machine-Learning Architectures & Techniques

- **Frameworks & Tooling:** PyTorch, Jax, TensorFlow, CUDA, Triton, ONNX Runtime, HuggingFace Transformers, Keras, Unsloth, Prophet, Bluehound, AutoML, MLflow, Langchain, LlamaIndex, DeepSpeed, PySpark, TVM, TensorRT
- **Vision & Perception:** CNNs (YOLOv4/8/10, EfficientDet, SegFormer), OCR stacks (TrOCR, Donut), Vision Transformers, CLIP multi-modal fusion
- **Sequence & Time-Series:** RNN/LSTM/GRU, Temporal Convolutional Networks, Transformer encoders/decoders, RETNet recurrent transformers, Graph-temporal hybrids
- **LLMs & SLMs:** BERT, RoBERTa, GPT-2/3/4, Phi-3, Llama-3, Mistral, Adapter/PEFT (LoRA/QLoRA, Prefix-Tuning)
- **Optimization & Compression:** Quantization (INT8/FP8), pruning, structured sparsity, low-rank factorization, distillation, gradient checkpointing, efficient attention kernels, mixture-of-experts
- **Retrieval & Reasoning:** Retrieval-augmented generation (RAG), vector search (FAISS, pgvector), ColBERT late-interaction, self-reflection agents
- **Classical ML:** XGBoost, LightGBM, Random Forests, boosted linear/GLM, statistical forecasting

Cloud & Distributed Systems

- **Platforms:** Azure (Data Explorer, Synapse, AKS, AML, AI Search, Ev2), AWS (SageMaker, EKS, Lambda), GCP, Snowflake, Databricks Lakehouse, Ray, HPC clusters (InfiniBand)
- **Containerization & Orchestration:** Docker, Kubernetes, Karpenter autoscaling, Helm, Terraform IaC, Argo Workflows
- **MLOps & Observability:** MLflow, Kubeflow, GitHub Actions, Prometheus/Grafana, OpenTelemetry, Datadog, Weights & Biases
- **Certifications:** Certified AWS Databricks Platform Architect and in Databricks Lakehouse Platform Fundamentals

Experience

Senior Software Engineer, Machine Learning (CISO Security Research Org), Microsoft – Boston, MA Dec 2023 – Present

- **Graph‑Native Multi‑Modal Agents:** Designed and shipped an agent platform that ingests **>1 B nodes+edges** from Defender, Entra, MS Graph, and Sentinel, fusing structured telemetry (KQL), vision embeddings, and LLM signals. Built dynamic retrieval pipelines and prompt‑engineering patterns that cut triage time for threat hunters by **68 %**.
- **Post‑Training RL Threat‑Intel Optimization:** Built a PPO‑Clip/RLAIF loop that ingested 250 k analyst‑labeled triage sessions + 15M synthetic adversarial playbooks to fine-tune a 13 B‑param SecOps LLM. The pipeline lifted malicious‑indicator ranking MAP 28 %, cut false‑positive escalations 45 %, and trimmed GPU cost 32 % via value‑function distillation and LoRA merging.
- **Large‑Scale Graph Anomaly Detection:** Prototyped sliding‑window GNN + temporal LSTM hybrid detecting abnormal account/device pivots; live A/B shows **4 ×** improvement in true‑positive rate over baseline heuristics.
- Engineered a **PyTorch‑YOLOv4 and v8 malicious QR-code detection model**, exported to ONNX and deployed a near-real-time computer vision pipeline using **AKS + KEDA autoscaling**; processes **500 M+** MDO messages/day with **28 ms p95 CPU latency**, eliminating **$25 M** vendor COGS and raising catch‑rate **21 pp**.
- Architected a **90 M‑param Phi‑Mini SLM** for BEC/phish: Spark/ADLS pipeline over **4 B** labeled SecOps messages, LoRA/QLoRA fine-tuning, **TensorRT INT8 + KV‑cache** for **2 ×** throughput (55 k→110 k QPS) at **98 % recall**.
- Authored the **ONNX Predictor** NuGet library (C#/.NET) delivering **<30 ms p95** CPU inference over image detection models and **45 % memory savings** via OrtValue pooling; now powering 40+ Defender CV models. The **ONNX Predictor** library utilized GraphOptimizationLevel.ALL, IO‑binding, OrtValue pooling, and ORT_ENABLE_PREPACK; delivers **<30 ms p95** CPU inference & **45 % RAM savings**; shipped as NuGet to **8 Defender teams** powering 40+ CV models
- **Collaborations:** Worked with the Cortex AI Research group on embodied AI research which spans generalizable large action foundation models, dynamic AI safety from policy output RL, foundation models for robotic manipulation and formal methods for safe and interpretable control.

Open-Source ML Software Engineer, Independent Contractor Aug 2022 – Present

- **Bare-Metal ML OS in Zig** – Prototyped a custom micro-kernel in Zig (no_std, panic-less) targeting x86-64; cross-compiled with zig-cc and boot-tested in QEMU/KVM. Designed a zero-copy ring-buffer driver for NVMe and a minimal SysV ABI loader that boots a Tensor-RT INT8 inference shim directly from disk image—laying groundwork for sub-20 MB edge appliances with deterministic latency.
- **Model optimization** – Implemented speculative decoding with draft model pruning (70% parameter reduction) and quantization-aware training (QAT) using FakeQuantize ops during fine-tuning. Deployed Llama-2-7B + 1Bit-LLM (MSFT) draft pair with acceptance rate of 0.73, achieving 2.1× throughput improvement while maintaining BLEU score within 2% on code generation tasks. Applied INT8 QAT with asymmetric quantization and per-channel scaling, reducing inference memory by 45% on RTX 4090
- **RLHF for Personalized-Therapy LLM** – Wired together LangGraph orchestration, LlamaIndex retrievers, and custom Pydantic-validated reward logging to train a DeBERTa-v3 preference model and run PPO fine-tuning on A100 × 8. 40-step micro-batches, dynamic-KL control, and FP16 gradient clipping cut iteration time 40 % and raised helpfulness score 0.56 → 0.81.
- **Production Hybrid-RAG Stack** – FAISS HNSW (4 096-d) + BM25 sparse, ColBERT-v2 late-interaction, HyDE zero-shot expansion, and CoT-guided rerank (Phi-2-LoRA-CoT). Served via Triton 2.39 + TensorRT-LLM INT8, sustaining 8 k QPS @ <85 ms p95; top-1 MRR up 27 pp on security QA corpus.
- **Domain Embeddings & Tokenizer** – Trained MiniLM-SecEmbed (6-layer, 256-d) with contrastive NSSA loss over 120M SecOps docs; crafted SentencePiece 32 k tokenizer with byte-fallback. Result: vector size down 30 %, semantic recall increased 18 %
- **Edge-Ready LLMs** – Fine-tuned Phi-3-Mini, Llama-3-8B-Instruct, and Mistral-7B large language models using Unsloth LoRA/QLoRA-64. Quantized models with GPTQ 4-bit (g=32) and GGUF Q4_0 for Ollama deployment. Achieved token latency of under 150 ms on an RTX 4080 and under 320 ms on a Jetson Orin.
- **Kernel & Memory Efficiency** – Added FlashAttention-v2 and Triton fused rotary-GEMM kernels with 2:4

structured sparsity masks; delivered 1.8 × throughput and 22 % VRAM savings. OpenTelemetry spans on every inference hop trimmed P99 tails 33 %.

Data Platform ML Software Engineer, Securian Financial – REMOTE in Utah May 2023 – Dec 2023

- **End-to-End MLOps Fabric** – Authored Terraform + CloudFormation modules that spin up SageMaker, EKS, Glue, and Kinesis stacks in <15 min (↓ 90 %); GitHub Actions CI/CT/CD pushed hardened AMIs, triggered blue/green Canary in SageMaker Endpoints, and auto-rolled back on <1 % drift
- **High-Throughput Data Plane** – Orchestrated Glue ETL on EMR-Spark 3.4 + Iceberg tables (Z-order, Hudi compaction) via Step Functions, feeding 2.1 TB/day into credit-risk training jobs; Spot-Instance policy + multi-AZ shuffle routing cut compute spend 58 %
- **Streaming Fraud Shield** – Built a Kafka→Kinesis Data Streams→Flink pipeline (~82 K msgs/s) with Lambda SnapStart feature extraction and SageMaker Serverless batch-transform; end-to-end decision latency <450 ms p95, flagging new fraud events 8 h sooner than legacy nightly batch
- **Microservice-Scale Inference –** Dockerized model shards, deployed on EKS Fargate with Helm + Karpenter + HPA, using Bottlerocket nodes and GP3 CSI volumes; sustained 12 k RPS at 99.97 % SLA while GPU-based auto-scaler trimmed idle costs 35 %.
- **Observability & Governance –** Wired Prometheus + Grafana, AWS Distro for OpenTelemetry, and MLflow registry webhooks; unified lineage reports satisfy SOX & Model Risk-Mgmt audits four months ahead of schedule.

Founding ML Software Engineer, Thera AI – REMOTE in Utah Jan 2023 – July 2023

- Attempted bio-infrastructure startup on the side of full time job
- **Bio-RAG Engine** – Stitched LlamaIndex + pgvector (Postgres 15) into a retrieval-augmented pipeline over 48 M UniProt/GenBank records. FAISS IVF-PQ (nlist = 4096) + ProtBERT-BF16 embeddings delivered top-k recall with an increase of 31 pp while holding query latency <140 ms p95.
- **Sequence-Design Workbench** – Fine-tuned ProGen-2-1.5B with LoRA-64 on A100 × 4, blended with ESM-2 650 M function predictors; generated 2 000 novel peptides/week, 16 % of which passed downstream wet-lab binding assays—cutting design cycle > 4×.
- **Observability Hooks** – Wrote LangGraph callback handler that captures nested DAG spans, token-level costs, and GPU profiler traces to OpenTelemetry + Tempo; surfaced fine-grain evals (BLEU, perplexity) in Grafana dashboards.
- **Full-Stack Gen-AI Portal** – React 18 / Next.js 13 front-end, FastAPI + Strawberry GraphQL back-end, containerised with Docker-Compose → ECS Fargate + ALB; CI/CD via GitHub Actions & Pulumi. Streaming SSE endpoints keep chat UI at <100 ms TTFB even on mobile.
- **Domain-Aware Chatbots** – Deployed Phi-2-LoRA-CoT agents that parse natural-language experiment plans, surface literature summaries (BM25+hybrid rerank), and auto-draft Benchling protocols—reducing researcher search time ≈ 45 %.

ML Data Scientist II (MLOps Specialist), Swire Coca Cola – Salt Lake City, UT Aug 2022 – June 2023

- **Overview**: Started and grew a team of data scientists and machine learning engineers at a startup initiative inside Coca Cola to handle supply chain issues.
- **Machine Learning & Deep Learning:** Built and implemented advanced machine learning models for diverse business applications, including sales demand forecasting via cutting-edge deep learning techniques (at the time). We performed a ton of time-series analysis accounting for extreme seasonality due the dataset we were working with. I also designed and developed the inference infrastructure for a trade promotion optimization application using genetic algorithms and Streamlit.
- **MLOps & Data Engineering**: Developed and streamlined CI/CD pipelines via Jenkins and Azure DevOps for seamless ML model deployment. Used Kubernetes for orchestrating containerized applications and Git for version control. Enhanced security and monitoring practices for robust ML application reliability.
- **Real-world Business Solutions**: Led projects addressing key business challenges such as **sales demand forecast** using time series analysis with deep learning models, **customer churn prediction** and intervention strategies using NLP analysis of feedback, **customer segmentation** through NLP-based clustering, and **supply chain optimization** using reinforcement learning.
- **Technology Evaluation & Implementation:** Identified and incorporated new software technologies to improve

performance, maintainability, and reliability of ML systems. Tools included MLOps life cycles, Streamlit, MLflow, Delta Lakes, Docker, Cloud development, Snowpark, Snowflake database, Databricks, and Azure.

Research Scientist University of California, Irvine – Irvine, CA Aug 2020 – Sep 2022

- Applied statistical and machine learning techniques, including deep learning, to create scalable simulations for systems of interest. Employed deep learning models such as CNNs for the analysis of molecular structures, and unsupervised learning techniques for clustering and dimensionality reduction. Analyzed and understood large amounts of data for specific conditions and worked closely with collaborators to optimize the complexity of the simulations.
- Implemented atomic-scale molecular dynamics and multi-conformational Monte Carlo simulations, as well as machine learning techniques to simulate protein structures and optimized conformations of the protein structures. This required in-depth statistical analysis as well as dimensionality reduction analysis such as, KNN, regression, clustering, SVMs. The data analysis was performed in both Python and R scripting.
- Used a multiscale molecular simulation approach to gain atomic-level insight into the interprotein interactions that stabilize concentrated solutions of wild-type γ-crystallins and lead to the formation of aggregates in solutions of their cataract-related mutants. Utilized deep learning techniques for protein structure prediction and classification, enabling a better understanding of the underlying mechanisms.
- Translated statistical simulation analysis results to experimentalist collaborators to verify results and discuss further simulations and types of analyses that needed to be completed, including the potential integration of advanced machine learning methods for the interpretation of research findings and the identification of novel therapeutic targets.

Research Scientist, Department of Energy & Washington State University – Pullman, WA June 2018 – May 2020

- Spearheaded multiple research projects in collaboration with Pacific Northwest National Laboratory to develop a data pipeline between WSU and National Laboratory for continuous research analytics in our computational models.
- Applied various statistical methods in computational design of Metal-Organic Frameworks (MOFs), including regression analysis, Principal Component Analysis (PCA), cluster analysis, machine learning algorithms, Bayesian optimization, Monte Carlo simulations, molecular dynamics simulations, genetic algorithms, and artificial neural networks (ANNs) for property prediction and optimization.
- Utilized these techniques to effectively explore the vast MOF design space, identify structure-property relationships, and guide experimental synthesis efforts towards optimal material designs.
- Deployed genetic algorithms as an optimization technique based on the principles of natural selection and genetics, used to search for optimal MOF structures by evolving a population of candidate materials through selection, crossover, and mutation operations.
- Exploited synthetic crystallographic techniques to understand MOFs as extrapolating agents in solid-solvent phase extractions to improve radioactive waste separations. Leveraged Monte Carlo simulations paired with experimental data to design the most 'optimized' material.
- Mentored undergraduate students pursuing research by teaching them laboratory techniques, conceptual understanding of research tactics, and leading them on projects they found interesting, including the application of advanced machine learning techniques in their research.

Undergraduate Research Assistant, Concordia University, Irvine – Irvine, CA June 2015 – May 2018

- Developed a machine learning program using python to determine the difficulty of a college course based on previous grades and the current grade distribution.
- Gained experience with experimental molecular and biomolecular techniques (i.e., DNA isolation, PCR, sub-cloning, microbial transformation, solution/media preparation, aseptic techniques) as well as computational protein modeling and statistical models (utilized python and R).
- Synthesized, purified, and spectroscopically characterized chromium transition metal complexes with acetylacetone, chloride and bromide ligands. Complexes were synthesized using controlled conditions.

- Determined degradation pathway of Sphingomonas bacterium of antibiotic resistant bacteria through Spectroscopic Techniques. Compared the localization of human and yeast copper-zinc superoxide dismutase (SOD1) in Saccharomyces cerevisiae.

Projects + Hackathons

ONNX ML Inference Microsoft Internal

- I developed an ONNX ML Predictor library built in C# and .NET which serves as a CPU inference compute engine for our models. This library allowed us to modularly add and remove models as new ones were added to production as well as load in over 40+ models asynchronously. The library essentially maps inputs to outputs of a specified scenario and model. So we are able to have computer vision models as well as text based models mapped through the Predictor serving a near real time inference of sub 30 ms for image detection models.

Embodied Shield Microsoft Internal

- Built jailbreak-resistant safety layer for vision-language robots, integrating GAIA-2-style bidirectional world model with STL-CBF shields to ensure safe actions. Developed PromptGuard (LoRA on Gemma-7B), reducing jailbreak success from 14% to <1% on 5k adversarial prompts (Embodied-Jailbreak-Bench)
- Implemented RT-2 policy (1.3B params), BWM (0.9B), and real-time QP solver on Jetson Orin; added <8ms latency.
- Achieved 35% fewer collisions, 9% higher task success on 50-task safety suite; hopefully will turn into a research paper

Mechanistic Interpretability of GPT 2 Github: ML Interpretability

- This work was inspired by Chris Olah (Anthropic) and built on top of foundational work done by Neel Nanda. I looked into the emergent properties of positional embeddings in GPT-2 using TransformerLens (developed by Neel Nanda) and custom probing classifiers. Implemented data generation pipelines, logistic regression and neural network probes, and performed layer-wise analysis and attention head ablation studies. Demonstrated the ability to predict word positions from residual streams and identified specific layers and attention heads crucial for position encoding. This project contributes to LLM interpretability by providing insights into how positional information is encoded and processed within the model architecture.

Real-Time Malicious QR-Code Detection & QR-Code Decoding Microsoft Internal

- QR code detection is a serious project when it comes to security, especially when QR-codes can be embedded with loads of malicious content in them. I built a proprietary computer vision model which is composed of two main building blocks: a QR detector model trained to detect and segment QR codes and a QR code decoder. The decoder is built using Pyzbar, different image preprocessing techniques that maximize the decoding rate on difficult images.

Blackbox AI: Interpretability of YOLO Object Detection Github: YOLO Interpretability

- Basically this project was inspired by an internal project I worked on and I wanted to further investigate the YOLO model series and try to understand the “why” behind its verdicts.
- Layer-wise Relevance Propagation (LRP) is a technique used for explaining decisions of deep neural networks by propagating the prediction backward through the network using purposely designed rules. Researchers developed and analyzed various LRP rules (LRP-0, LRP-ε, LRP-γ) for different network layers, optimizing for both fidelity and understandability of explanations. They applied the Deep Taylor Decomposition framework to theoretically justify LRP rules and implemented efficient LRP algorithms using automatic differentiation in PyTorch, enhancing interpretability for complex AI models. This project was an implementation of this research paper: Layerwise Relevance Propagation

Rapidly Github: Rapidly

- This is a one stop shop for enterprise knowledge management software. Think of it as Glean or GlueAI but modern, dynamic, quick and reliable. We all know that one person who we turn to for all information in the company dating back 15+ years. Rapidly is the software that enables all employees to be knowledgeable over the entire stack without requiring 15+ years of experience. This knowledge management software deploys LLMs safely and reliably across the enterprise. Youtube [DEMO] Link

Finetune Phi 3.5 using Unsloth Github: LLM-stuff

- This project fine-tunes the Phi-3.5-mini-instruct model on various cybersecurity datasets using the Unsloth library. It processes and formats multiple datasets related to MITRE ATT&CK, cybersecurity tactics, and vulnerabilities, preparing them for training. The script implements efficient training techniques, including LoRA and 4-bit quantization, and includes features for model evaluation, saving, and optionally pushing to the Hugging Face Hub.
- Leveraged Unsloth library for advanced LLM fine-tuning, implementing Low-Rank Adaptation (LoRA) for efficient parameter updates and hardware-specific optimizations for NVIDIA GPUs. Utilized Automatic Mixed Precision (AMP) and gradient accumulation to accelerate training while maintaining accuracy. Employed Unsloth's memory-efficient "Tron" kernels and precise analytical backpropagation, significantly reducing memory usage and computational overhead in LLM training pipelines.

BioInformatics Research Assistant Github: BioIDE

- This project was in the early days of ChatGPT so I basically wanted to implement a domain specific personal research assistant. I had done quite a bit of computational biophysics research in the past, especially in grad school so I was building something I wish I had at the time. I noticed early on that ChatGPT was too general when it came to domain specific queries, so I figured if we combined the context of the latest research with the knowledge of ChatGPT then we could speed up the process of reading, discovering, and doing research.

Research Papers

- Graph‑Grounded Agents for Large‑Scale Threat Hunting – **Lead author**; Microsoft Security Research Series, 2025.
- Hybrid GNN‑LSTM Anomaly Detection on Defender Graphs – **Lead author**; Microsoft Security Research, 2025.
- High‑Throughput ONNX Inference Engine for Microsoft Defender Vision Models – **Lead author**; ML Systems for Security @ Microsoft, 2024
- Post‑Training Compression of SLMs for Security Telemetry – **Co‑author**; Microsoft Security Research, 2024
- Li X., Ding G., Hao L., Deming, D. A,[b] and Qiang Zhang (2020). *ACS Appl. Mater. Interfaces* https://doi.org/10.1021/acsami.0c04961
- Hao, L., Ding, G., Deming, D. A. and Zhang, Q. (2019), *Eur. J. Org. Chem.* doi:10.1002/ejoc.201901303
- Derek Deming et. al (2019) A Facile Method to Introduce Iron Secondary Metal Centers into Metal–Organic Frameworks, *J. Organ. Chem*. doi.org/10.1016/j.jorganchem.2019.06.0

Research Talks

Microsoft MLADS+ Responsible AI Conference: **Real-Time Malicious QR-Code Detection & QR-Code Decoding**

Microsoft FireCon (focused around Security, Incident Response, Reverse Engineering and more): **How to Make Computer Vision Models Less of a Black Box through Layerwise BackPropagation**