Aryan Dua

DETAILED CURRICULUM VITAE

AI research & engineering

aryan.iitd1@gmail.com +91 8830639139 tan90cot0.github.io GitHub / tan90cot0 LinkedIn / aryandua

Research and applied work in biomedical language models, retrieval, multi-agent systems, evaluation, robustness, and asynchronous convex optimization.

EDUCATION

Indian Institute of Technology Delhi

2020 - 2025

B.Tech + M.Tech in Computer Science

B.Tech GPA: 9.00/10 | M.Tech GPA: 8.80/10
Institute merit list, top 7% in Semesters 1 and 2; branch change to CSE in 2021.

BIOMEDICAL AI RESEARCH

AI-Powered Personalized Treatment Pathways

A Comparative Analysis of Biomedical Vision-Language Models
Master's thesis / Gen AI & NLP Research Intern, IIT Delhi & Wipro
Jan 2024 - May 2025 | Thesis submitted Jun 2025 | Advisor: Prof. Mausam

Investigated how domain adaptation, biomedical retrieval, multimodal information extraction, and specialized agents can support patient-specific diagnosis and treatment reasoning. Compared MEDITRON, MedAlpaca, LLaVA-Med, OpenBioLLM, Llama, and DeepSeek model families using medical question answering and clinical cases.

Benchmarks included MedQA, PubMedQA, MedMCQA, and MMLU-Medical. The research examined both aggregate scores and failures in patient-specific reasoning, psychiatric risk detection, and guideline adherence.

Parameter-efficient fine-tuning

Fine-tuned MEDITRON-7B using 8-bit quantization and LoRA on 200,000 artificially labeled PubMedQA examples. Used rank 16, alpha 32, dropout 0.1, and attention projection targets. Training used AdamW, a 2e-5 learning rate, three epochs, a 2,048-token sequence limit, mixed precision, gradient accumulation, and gradient checkpointing.

PubMedQA accuracy reported in the thesis

Configuration	Accuracy	Difference
MEDITRON-7B baseline	43.8%	-
LoRA fine-tuned MEDITRON-7B	73.8%	+30.0 percentage points

Adaptation and compute trade-offs

Evaluated LoRA ranks 4-64. Rank 16 achieved 73.8% PubMedQA accuracy in 8.4 hours; rank 32 reached 74.1% in 14.6 hours, showing the small incremental gain relative to additional training time.

> Experimental scope: the best revised agent improved peak F1 by 0.015 absolute over the single-model baseline. The study uses curated cases and a limited expert panel; benchmark gains do not establish prospective clinical benefit. Answer-conditioned teacher results are separate from independent diagnosis performance.

BIOMEDICAL AI

Retrieval, agents & evaluation

Master's research, continued / Retrieval, specialized agents, and evaluation.

Biomedical retrieval pipeline

Built a PubMed-based RAG pipeline using PubMedBERT embeddings, FAISS dense search, BM25 sparse retrieval, and cross-encoder re-ranking. The documented setup used 512-token chunks with 50-token overlap, 768-dimensional normalized embeddings, a 70/30 dense-sparse weighting, and top-5 retrieval.

Retrieval results and component ablations

For fine-tuned MEDITRON-7B, the thesis reports average benchmark accuracy increasing from 62.4% to 71.3% with retrieval (+8.9 points). Dense-only retrieval yielded +5.2 points, hybrid retrieval +7.8, and re-ranking +8.9. Factual hallucination rates in the clinical evaluation fell from 8.3% to 4.8%, approximately a 42% relative reduction.

Diagnostic architecture and iteration

Designed DeepSeek R1-8B agents for symptom analysis, medical history, and clinical results, with a diagnostic integration agent receiving structured summaries and original case information. Revised agent inputs and output requirements to address missed psychiatric risks, medication safety, urgency, severity, and differential diagnosis.

Diagnostic agent ablations / peak F1, threshold k < 8.5

Configuration	Peak F1
Single diagnosis model	0.475
Original agents + original case	0.455
Original agents, without original case	0.240
Revised agents + original case	0.490
Revised agents, without original case	0.460
Revised agents + original case + ICD list	0.420
Revised agents + original case, 3-letter evaluation	0.325

Human-aligned evaluation and robustness

Compared FrugalScore, BERTScore, GPT-4 judging, and BioBERT v2 with human diagnostic-similarity judgments. Reported agreement was 65%, 71%, 76%, and 82%, respectively. Analyzed confidence thresholds, missing case information, diagnostic-code classes, patient age, inaccurate clinical assumptions, and psychiatric-risk failures.

Reasoning and multimodal directions

Explored Llama fine-tuning on DeepSeek-70B reasoning traces; answer-only and reasoning-plus-answer variants both reached 40% precision and 39% recall in the reported comparison. Investigated TabTransformer/DePlot components for biomedical tables and plots, and a five-candidate diagnosis mechanism with supporting/counter evidence and final arbitration.

OPTIMIZATION RESEARCH

Asynchronous projective splitting

Zuse Institute Berlin / Research Intern

Jun - Aug 2022 | Advisors: Zev Woodstock and Sebastian Pokutta

Contributed to a general-purpose Julia implementation of the Combettes-Eckstein asynchronous, block-iterative primal-dual projective splitting algorithm for fully nonsmooth convex optimization. The work bridges existing convergence theory and practical implementation in machine learning and image recovery.

Asynchronous implementation

Used Distributed.jl to distribute proximal computations across workers, with ProximalOperators.jl interoperability and support for custom proximal operators. Supported variable-size blocks, matrix or functional linear operators and their adjoints, full/cyclic activation, configurable delay bounds, and iteration-dependent proximal parameters.

Instrumentation and experimentation

Recorded objective values, iterate differences, convergence proxies, epochs, iterations, and proximal-operator calls. Ran experiments on a Linux HPC system managed by Slurm. Compared synchronous delay bound D=0 with asynchronous D=5 under simulated proximal-computation latency, averaging ten runs.

Sparse classifier training

Studied latent group lasso with overlapping groups: 10,000 dimensions, 1,000 measurements, 1,429 groups, and 25% simulated label noise. A compressed representation stored each group's ten active coordinates rather than a dense vector.

Stereoscopic image recovery

Formulated recovery of multiple blurred and noisy images using wavelet sparsity, a quadratic fidelity term, and consistency between horizontally shifted views. Used a 5x5 averaging blur, Gaussian noise, and 300 recovery iterations.

Runtime under simulated latency / current public PDF, Table 1

Workers	Synchronous	Asynchronous	Reduction
2	881.144 s	728.214 s	17.4%
3	923.442 s	701.419 s	24.0%

Scaling and hyperparameter findings

Compared two to eight workers; three workers produced the fastest asynchronous runtime in the reported sweep, with additional workers increasing overhead. Used logarithmic grid search followed by ternary search and compared constant, linear-decrease, nonlinear-decrease, and random proximal-parameter schedules. Decreasing schedules outperformed constant ones in the reported classifier study.

RESEARCH OUTPUTS

A note on asynchronous Projective Splitting in Julia

Utkarsh Sharma, Kashish Goel, Aryan Dua, Sebastian Pokutta, Zev Woodstock (repository listing).

Optimization Online preprint, posted Apr 2025; updated May 2026. Timings follow the current public PDF.

Paper: Optimization Online | Current PDF | Code: AsyncProx

APPLIED AI

From evidence to working systems

Building retrieval, agent runtimes, evaluation infrastructure, and usable AI applications.

Lymph Labs / Co-Founder & CTO

Jul 2026 - Present

Medical evidence-search platform; MVP used by 2,000+ doctors, including clinicians from AIIMS and JIPMER.

- **Retrieval and provenance:** Built medical RAG across textbooks, clinical guidelines, and research literature, combining ingestion, OpenSearch/Qdrant hybrid retrieval, biomedical search, and deduplication with passage-level provenance and traceable citations.
- **LLM runtime:** Engineered multi-provider LangGraph/LangChain workflows for clinical research and medical education, with parallel tool execution, structured outputs, citation validation, token budgets, streaming, and bounded retries/fallbacks.
- **Durable operations:** Designed FastAPI/PostgreSQL API and worker services with transactional outbox dispatch via Cloud Tasks, idempotent operations, replayable SSE, cancellation, workspace authorization, rotating sessions, and usage metering.
- **Web and mobile:** Built Next.js/React and React Native/Expo iOS/Android clients with evidence-linked answers, OCR-backed uploads, clinical document workflows, editable medical mind maps, and admin tools for users, operations, analytics, and spend.
- **Evaluation and observability:** Built Langfuse tracing, versioned prompt management, evaluation harnesses, and retrieval replay/profiling tools to investigate source grounding, answer coverage, latency, token costs, and provider failures.
- **Infrastructure and delivery:** Provisioned Google Cloud Run and Cloudflare Pages infrastructure with Terraform. Built CLI release pipelines with immutable artifacts, independent component releases, validation gates, smoke checks, and rollback workflows.

AGENT PLATFORMS

Abacus.ai / ML Platform Engineer

Jul 2025 - Jul 2026

- Built the infrastructure framework enabling users to create their own skills and use them with their own DeepAgent instances.
- Built an internal AI software engineer capable of autonomous engineering tasks across the stack.
- Refactored core LLM infrastructure for native tool calling across providers including OpenAI, Anthropic, and Google, supporting tool-using agents through different model interfaces.
- Implemented end-to-end text-to-speech for DeepAgent using OpenAI, Hume, ElevenLabs, and Gemini providers.
- Designed S3-backed persistent storage for user-uploaded files on AppLLM, including presigned upload/download URLs and multipart uploads supporting files up to 5 TB.

EXPERIENCE & TECHNICAL BREADTH

Teaching, industry & foundations

IIT Delhi / Teaching Assistant, Parallel Programming — Jul 2024 - Jun 2025

- Designed assignments covering parallel computing and multithreading; mentored 150+ students in OpenMP, MPI, CUDA, and parallel algorithm optimization.
- Developed automated assignment evaluation frameworks on HPC clusters using PBS batch job scheduling.

Quadeye / Quant Intern, High-Frequency Trading — May - Jul 2024

- Researched equity-derivative arbitrage and cash-and-carry opportunities with minimal delta exposure; analyzed volume, trades, and order-book dynamics to generate real-time entry/exit signals using statistical analysis.

JackandJill.ai / Founding Engineer Intern — May - Jun 2025

- Built automated inbound-email attachment processing for diverse file formats and metadata extraction. Developed HTML scraping pipelines for job application links and integrated search APIs and LLM services for metadata extraction.

Ripik.ai / Software Developer / ML Intern — May - Jul 2023

- Built APIs for Ripik Optimus, an AI-driven pharmaceutical production planner for SunPharma. Developed graph-based scheduling algorithms and pipelines for plant-data preprocessing and feature engineering.

TECHNICAL SKILLS

Model adaptation
PyTorch, Hugging Face Transformers, TensorFlow, supervised fine-tuning, LoRA/PEFT, quantization, distillation, few-shot prompting.

Retrieval and agents
RAG, PubMedBERT, FAISS, BM25, OpenSearch, Qdrant, cross-encoder re-ranking, LangGraph, LangChain, multi-agent systems, native tool calling.

Evaluation and optimization
Benchmarking, ablations, precision/recall/F1, robustness and hallucination analysis, confidence thresholds, BERTScore, BioBERT, Langfuse, convex/proximal optimization.

Languages and compute
Python, C/C++, Julia, Java, R, SQL, Bash, JavaScript, HTML/CSS; CUDA, MPI, OpenMP, HPC, Slurm, PBS.

Backend and infrastructure
FastAPI, Django, PostgreSQL, Redis, S3, Docker, Git, Linux, Cloud Tasks, Google Cloud Run, Terraform, Cloudflare Pages, CI/CD.

Applications and data
Next.js, React, React Native/Expo, document ingestion, OCR workflows, web scraping, feature engineering, OpenCV, scikit-learn.

HONORS & DISTINCTIONS

- 5th place, Dream11 Gameathon ML competition among premier engineering institutes (2025).
- Among 11 students in a batch of 1,200 to secure a branch change to IIT Delhi Computer Science; institute merit list, top 7% in Semesters 1 and 2 (2021).
- JEE Advanced AIR 516; JEE Main AIR 534; KVPY AIR 677; top 100 in BITSAT; qualified for INChO; Maharashtra PRMO rank 12 and RMO qualifier (2020).
- Regional bronze medal, World Robot Olympiad (2017).

NLP RESEARCH / 01

Detecting Wikipedia controversy

Leveraging LLMs for Enhanced Detection of Controversies On Wikipedia

Rudhar Pratap Singh, Aryan Dua, and Ananya Aakriti
Co-authored IIT Delhi project report

Read report (PDF) | Google Drive copy

Question and dataset

Can language models identify controversial Wikipedia topics more effectively than edit-history and lexical signals? We manually annotated a balanced dataset of 142 topics (71 controversial and 71 non-controversial), using group discussion to establish labels. Preprocessing extracted article titles, edit counts, article content, summaries, controversy scores, and labels.

Classical and revision-history baselines

Evaluated logistic regression on edit counts, Naive Bayes on article text, logistic regression combining CountVectorizer features with edit counts, and a controversy measure based on mutual reverts. The strongest baseline combined text and edit counts, reaching 93.1% accuracy; the other reported results were 75.9%, 86.2%, and 77.4%, respectively.

Language-model and prompt experiments

Compared BART NLI zero-shot classification (65%) and a RoBERTa-base few-shot configuration (44%). Adding edit counts reached 80%; Pegasus summaries reached 81%. Grouping articles into nine thematic subtopics and supplying related examples reached 87%, the best reported prompting result. BERT sentence embeddings and cosine similarity were also used to select in-context examples, reaching 84%.

Evaluation and interpretation

Evaluation used only 29 examples. Classical text-plus-edit features outperformed the language-model configurations; no LLM advantage or broad generalization is established. Larger and more diverse evaluation was identified as future work.

Research record

Co-authored the project report with Rudhar Pratap Singh and Ananya Aakriti. Conducted at IIT Delhi under Prof. Abhijnan Chakraborty, Oct-Nov 2023.

NLP RESEARCH / 02

Graphs to natural language

Language from Links: Optimizing Graph Embeddings for Text Generation using Large Language Models

Samidha Verma and Aryan Dua
Co-authored IIT Delhi project report

Read report (PDF) | Google Drive copy

Task and benchmark

Studied conditional generation of natural-language descriptions from DBpedia knowledge graphs using WebNLG. The project adapted decoder-only GPT-2 and compared text-linearized graph inputs with graph-embedding approaches. Evaluation distinguished seen DBpedia categories from categories absent during training.

My contributions

The report credits me with implementing the GPT-2 task from scratch using recent packages, implementing the first graph-embedding method, preprocessing input files, and co-writing the report and presentation. Samidha Verma implemented the adaptations to the supplied repository, the second embedding method, and the global graph construction; I describe those as team experiments.

GPT-2 adaptation and fine-tuning

The project introduced head/relation/tail markers, relation-label vocabulary entries, left padding, and an instruction to convert triples into English. Training used Adam at 3e-5 with linear learning-rate decay. Batch sizes of 2, 4, and 8 and beam widths of 1, 3, and 5 were selected by development BLEU. Effective changes from the custom implementation were integrated into the original codebase, with output trimming and redundant-fragment removal.

Graph encoder experiment I implemented

Converted linearized triples into NetworkX graphs and PyTorch Geometric objects, encoded nodes with a graph convolutional network, and mean-pooled node embeddings to condition GPT-2 alongside text. This experiment did not yield satisfactory generation, motivating further architecture and training work.

Broader team investigation

The team also explored a heterogeneous global graph and graph-transformer/MLP conditioning. This was the collaborator-led second method; testing was incomplete and generated text remained repetitive or nonsensical.

Results and generalization

The final text-input pipeline reports BLEU 23.00 overall, 30.87 on seen categories, and 12.91 on unseen categories. Overall ROUGE-1/2/L F1 scores are 0.4268/0.2171/0.3227. Published BART/T5 baseline scores reproduced in the report are substantially higher; they are reference results, not models we trained. The seen/unseen gap and failed embedding runs expose limits in generalization and structural conditioning.

PROJECT RECORD

Language models, ML & systems

01 / NLP **Graph-to-text generation with GPT-2** Report

Implemented GPT-2 from scratch, input preprocessing, and a GCN graph-embedding pipeline. Team results: BLEU 23.00 overall, 30.87 seen, 12.91 unseen. Full methods, contribution attribution, and unsuccessful embedding experiments appear on page 7.

02 / NLP **Hate span detection with BERT & CRF** Code

Combined BERT representations and conditional random fields for sequence labeling, identifying hateful spans within sentences. The task targets the portions of a sentence containing hateful content rather than assigning only a sentence-level label.

03 / NLP **Controversy detection using LLMs** Report

Co-authored a study comparing Wikipedia revision signals, lexical classifiers, and language-model prompts. Best prompting: 87%; text-plus-edit-count baseline: 93.1%, on 29 test examples. Full dataset, comparisons, and limitations appear on page 6.

04 / ML **Text and image classification**

Implemented Naive Bayes for movie-review sentiment prediction and constructed a neural network for Fashion-MNIST classification, using 28x28 grayscale images across ten clothing categories. Supervised by Prof. Parag Singla, Sep-Oct 2022.

05 / ML **CUDA image processing & LeNet-5** Code

Built a C++/CUDA image-processing library and implemented LeNet-5 for GPU-accelerated MNIST handwritten-digit recognition. Connected neural-network operations to lower-level parallel execution on the GPU.

06 / SYSTEMS **Distributed graph algorithms with MPI** Code

Built a 64x64 perfect-maze generator and solver using MPI for parallel processing. Explored BFS, DFS, Kruskal's, and Dijkstra's algorithms in graph generation and solving workflows.

07 / SYSTEMS **CRDT collaborative notebook editor** Code

Built a collaborative Jupyter notebook-style editor using conflict-free replicated data types. Supported real-time collaboration and concurrent edits through distributed data structures designed to reconcile changes without conflicts.

08 / SYSTEMS **Linux kernel programming** Code

Implemented Linux kernel components involving scheduling, memory management, and device drivers. Added process-monitoring and context-switch-tracking system calls and developed thread-safe kernel data structures. Operating Systems course project, Feb-Apr 2023.

RELEVANT COURSEWORK

Machine Learning (COL774), Advanced AI (COL864), NLP (ELL884), Advances in LLMs (AIL821), Artificial Intelligence (COL333), and Data Structures (COL106).