Mani Pal

Delhi, India | Open to Remote & Relocation
palmani2410@gmail.com | +91 73806 26997
Linkedin | Github

Professional Summary

ML Engineer specialising in LLM systems, GPU kernel engineering, and inference optimisation. Pre-trained a 700M-parameter hybrid Mamba-2/Transformer LLM from scratch; implemented FlashAttention-2 CUDA kernels profiled at 2.1× throughput over PyTorch SDPA on A100; built a speculative decoding runtime achieving 2.4× CPU inference speedup with mathematically identical output distributions. Extended Multiverse Computing's CompactifAI research (arXiv:2401.14109) with adaptive tensor-network compression on real LLMs, achieving 93% memory reduction with REINFORCE-guided bond-dimension scheduling. Active contributor to vLLM; independent mechanistic interpretability research on grokking published on Zenodo and under ArXiv submission; certified in Advanced Machine Learning and Deep Learning (MITx/edX).

Technical Skills

ML Frameworks	PyTorch, JAX/Flax, TensorFlow, Triton (OpenAI), CUDA C++
LLM & Inference	vLLM, SGLang, TensorRT-LLM, llama.cpp, GGUF/GGML, Speculative Decoding, FlashAttention-2, PagedAttention, KV-Cache Optimisation, LangChain, LlamaIndex, RAG Pipelines
Training & Alignment	FSDP, DeepSpeed ZeRO, Megatron-LM, LoRA, QLoRA, RLHF, GRPO, DPO, PPO, Reward Modelling, Chinchilla Scaling Laws, Instruction Tuning, SFT
Model Architectures	Transformer, Mamba-2, SSM, Mixture-of-Experts (MoE), RWKV, Hybrid SSM-Attention, RoPE, YaRN, LongRoPE, BPE Tokenisation
Compression & Quantisation	Tensor Networks, Matrix Product Operators (MPOs), SVD, INT4/INT8 Quantisation, bitsandbytes, GGUF Quantisation, Layer Sensitivity Profiling, Adaptive Bond-Dimension Scheduling, REINFORCE Policy Optimisation
Evaluation	MMLU, HumanEval, MT-Bench, BIG-Bench, lm-evaluation-harness, OpenAI Evals, Perplexity Benchmarking, HellaSwag, BoolQ, TriviaQA, GSM8K
MLOps & Cloud	Weights & Biases, MLflow, Docker, Kubernetes, AWS (EC2, S3, SageMaker, ECS, Lambda), Terraform, Prometheus, Grafana, OpenTelemetry, CI/CD, GitHub Actions
Vector Search & Retrieval	Pinecone, FAISS, ChromaDB, Qdrant, Weaviate, pgvector, BM25, Hybrid Reranking, Cross-Encoder Reranking
Programming & Backend	Python, Rust, TypeScript, C++, FastAPI, Node.js, PostgreSQL, Redis, Kafka, Nginx, gRPC, REST APIs

Work Experience

Trellions — *ML Engineer / LLM Engineer* May 2026 – Present
Remote

- Own the full AI layer of Trellion's YC-backed recruitment platform; fine-tuned and evaluated LLMs (GPT-4o, LLaMA-3, Mistral) for resume scoring, job description generation, candidate Q&A, and bias detection across the end-to-end hiring funnel.
- Built and maintain production RAG pipelines using Pinecone, Weaviate, and pgvector with hybrid semantic search and cross-encoder contextual re-ranking; integrated AI modules into FastAPI services meeting strict latency and throughput SLAs.
- Developed ML models for structured entity extraction (skills, titles, experience), candidate ranking, and job-fit classification; own MLOps end-to-end including experiment tracking with MLflow and W&B, model versioning, CI/CD, and drift monitoring.
- Authored architecture RFCs and technical documentation enabling async-first decision-making across a fully remote engineering team; contributed to AI team hiring including candidate sourcing, screening, and closing.

RecurX — *Founding Engineer & Technical Lead* January 2024 – November 2025
Delhi, India

- Architected cross-chain decentralised payment infrastructure across Ethereum and Solana achieving sub-200ms transaction latency; authored Solidity smart contract suite following OpenZeppelin standards with 95% test coverage via Hardhat and Foundry.
- Reduced p95 API latency from 280ms to 182ms (35% improvement) through Redis caching, B-tree and GiST indexing, and DataLoader N+1 query elimination on PostgreSQL with Prisma ORM.
- Built CI/CD pipeline using GitHub Actions, Docker, and AWS ECS with blue-green deployments, cutting release cycles from 45 minutes to 8 minutes with zero downtime; monitored 12+ microservices via OpenTelemetry, Grafana, and Prometheus.
- Designed horizontally-scalable microservices architecture with Redis pub/sub and WebSocket broadcasting supporting 5,000+ concurrent clients with sub-second message delivery.

Independent Contract AI/LLM Engineer — *Clients across India, US & EU* January 2021 – December 2023
Remote

- Built a production RAG pipeline (LangChain, FAISS, GPT-4) for a legal-tech SaaS platform serving 50+ law firms, reducing attorney document-research time by 65% via hybrid BM25 and cross-encoder semantic re-ranking with sub-300ms retrieval latency.
- Fine-tuned Mistral-7B and Llama-2 via QLoRA on AWS SageMaker for a B2B analytics platform; built event-driven microservices with RabbitMQ and Kafka achieving 85%+ test coverage (Jest, Cypress, Playwright) and reducing production incidents by 60%.
- Implemented materialised views, query partitioning, and EXPLAIN-driven optimisation achieving 10× query throughput improvement on high-cardinality PostgreSQL operations.

Projects

FlashAttention-2 CUDA Kernel – Custom GPU Attention Engine | GitHub 2025
Stack: C++, CUDA C++, PyTorch, Triton, NVIDIA Nsight Compute

- Implemented tiled IO-aware FlashAttention-2 from scratch in CUDA C++, fusing QKV matrix multiply, softmax, and weighted-sum into a single SRAM-resident kernel pass, reducing HBM memory bandwidth from $O(N^2)$ to $O(N)$.
- Applied thread-block tiling (B_r=64, B_c=64), register-level accumulation, and causal masking; profiled with NVIDIA Nsight Compute achieving 2.1× throughput over `torch.scaled_dot_product_attention` on A100 at sequence length 4096.
- Cross-validated Triton kernel variant to float32 precision against PyTorch reference; packaged as a standalone PyTorch C++ extension installable via pip; kernel design write-up published on GitHub.

Project Chimera – 700M Hybrid LLM Trained from Scratch | GitHub 2024 – 2025
Stack: PyTorch, Mamba-2, Triton, GGUF, llama.cpp, DeepSpeed, Weights & Biases

- Designed and pre-trained a 700M-parameter LLM from scratch using a hybrid Mamba-2 and Transformer architecture with SSM layers for linear-complexity context handling and interleaved attention at key depths using Chinchilla-optimal data scheduling.
- Built custom BPE tokeniser trained on Hindi and English corpora; applied GRPO reasoning fine-tuning (DeepSeek R1-style) for chain-of-thought reasoning without human annotations; performed DPO safety alignment and QLoRA domain adaptation.
- Achieved INT4 GGUF quantisation with a 4.2 GB model footprint and sub-3-second first-token latency on CPU-only inference; benchmarked on MMLU and HumanEval; model weights and benchmark results published on Hugging Face Hub.
- Investigated KV-cache memory constraints, YaRN and LongRoPE for extended context windows; architecture decisions documented as a public research log on GitHub.

Sparse MoE Scaling Experiment – Mixture-of-Experts Architecture from Scratch | GitHub 2025
Stack: PyTorch, FSDP, Triton, Weights & Biases

- Implemented sparse top-k MoE layer (k=2, 8 experts) with z-loss load-balancing using the Switch Transformer formulation; trained a 1B-compute- equivalent model at 125M active parameters achieving 2.3× throughput versus a dense baseline at matched FLOP budget.
- Characterised expert-utilisation collapse via routing-entropy curves; identified z-loss coefficient thresholds preventing collapse; all training runs logged in W&B and open-sourced on GitHub with full reproducibility.

Speculative Decoding Inference Engine – OpenAI-Compatible Server | GitHub 2025
Stack: Python, PyTorch, llama.cpp, GGUF, FastAPI

- Built a draft-verifier inference runtime pairing Qwen2.5-0.5B draft model with Qwen2.5-3B verifier using temperature-corrected rejection sampling per DeepMind's formulation; achieved 2.4× mean tokens-per-second on CPU with mathematically identical output distributions.
- Implemented adaptive gamma scheduling: lookahead window adjusts per-step based on acceptance rate, maximising speedup on low-entropy outputs (code, JSON) and degrading gracefully on high-entropy text; served via an OpenAI-compatible `/v1/completions` FastAPI endpoint.

VAANI – Hindi-First Fully Offline Voice Assistant | GitHub 2024
Stack: Python, Whisper, Qwen2.5-3B, Piper TTS, XTTS v2, openWakeWord

- Built a fully offline voice pipeline combining openWakeWord detection, Whisper-small ASR, Qwen2.5-3B-Instruct Q4_K_M with 128K context window, and Piper TTS achieving under 800ms end-to-end latency on consumer CPU with zero internet dependency.
- Fine-tuned XTTS v2 on AI4Bharat Hindi corpus for a custom Jarvis-style voice persona; designed a modular 8-layer plugin architecture enabling new tool integrations without modifying the core inference pipeline.

Open Source Contributions

vLLM | PR #38816 | Disaggregated Prefill Pipeline – KV Cache Request ID Bug Fix 2025

- Diagnosed and fixed a critical hang in vLLM's disaggregated prefill pipeline caused by inconsistent request-ID formatting between prefill and decode nodes, which prevented decode nodes from locating the correct KV cache tensors and stalled inference indefinitely.
- Implemented a request-ID normalisation layer at the prefill-decode boundary and refactored KV cache lookup logic to enforce consistent identifier semantics across distributed nodes; added targeted unit tests validating correctness of cross-node KV cache transfer under both matched and previously-mismatched ID formats.
- Fix resolves runtime hangs in production disaggregated inference deployments, improving reliability of distributed KV cache transfer workflows critical for high-throughput LLM serving at scale.

Research

Grokking Beyond Addition: Circuit-Level Analysis of Algebraic Learning in Transformers 2026
Independent Research | Published: zenodo.org/records/19256207 | GitHub

- Extended mechanistic interpretability research on grokking across 8 algebraic operations spanning abelian fields, a composite ring, and four non-abelian groups (S_3, D_5, A_4, S_4); trained 1-layer transformers (d_{model}=64, 3 seeds per operation), establishing the first empirical capacity-dependent abelian/non-abelian grokking boundary – all 4 abelian operations grokked to 100% test accuracy within 2,000 epochs; all 4 non-abelian groups memorised (100% train accuracy) but failed to grok.
- Provided geometric evidence for the discrete-log representation hypothesis: dlog re-indexing improved multiplication's Fourier concentration by 2.14× (raw 9.4% → re-indexed 20.0%, g=3); Peter–Weyl analysis of non-grokked non-abelian models correctly identified the dominant irrep in all four cases (S_3: standard d=2; D_5: ρ_{2a}; A_4: ρ_3, d=3; S_4: standard$_3$, d=3), indicating partial circuit formation precedes generalisation.
- Measured centred kernel alignment (CKA) across all 28 embedding pairs, finding uniformly high cross-operation similarity (≥ 0.80, mean 0.90), with the striking add–S_3 pair at 0.97; derived formal complexity scores C_1/C_2 from character tables predicting abelian grokking order; ran three controlled ablations (dataset size, training fraction, weight decay) confirming the ring > addition delay ordering is robust to all tested confounds.

Credit Assignment in Spiking Neural Networks: Bridging Bioplausibility and Scalability 2026
Independent Research | Open Problem Investigation | GitHub

- Investigated the credit assignment problem in spiking neural networks (SNNs) – the core challenge of propagating error signals through discrete, non-differentiable spike events – benchmarking surrogate gradient methods (SuperSpike, SLAYER, EXODUS) against biologically-plausible alternatives (e-prop, RTRL) across temporal classification tasks.
- Analysed the fundamental tension between biological plausibility and gradient-based scalability in recurrent SNN architectures; characterised vanishing/exploding gradient pathologies introduced by leaky integrate-and-fire neuron dynamics across deep temporal unrollings and evaluated STDP as a local learning rule against BPTT baselines.
- Proposed and evaluated hybrid credit assignment strategies combining surrogate gradients with local eligibility traces, targeting online learning in SNNs without full BPTT; documented findings, failure modes, and scaling analysis in a reproducible codebase on GitHub.

Adaptive Tensor-Network Compression of LLMs: An Extension of CompactifAI 2026
Independent Research | Extended Research on Multiverse Computing (arXiv:2401.14109) | GitHub

- Reproduced and extended Multiverse Computing's CompactifAI framework (arXiv:2401.14109) on real open-weight LLMs (LLaMA-3.2-1B, Qwen2.5-1.5B); implemented Matrix Product Operator (MPO) tensorisation of Self-Attention and MLP weight matrices via sequential SVD with controllable bond dimension χ, achieving up to 93% memory reduction (matching the original paper's LLaMA-2 7B result at 1B-scale).
- Conducted real layer sensitivity profiling by sweeping $\chi \in [10, 90]$ independently across all 32 attention blocks and 7 layer types (L1–L7); confirmed that initial blocks are compression-sensitive (accuracy collapses below χ=50) while terminal blocks tolerate χ=10 with under 1% MMLU drop; profiling results used to construct a non-uniform per-block compression schedule.
- Introduced a REINFORCE-based policy network that learns per-block bond-dimension assignments end-to-end using downstream MMLU accuracy as a reward signal; the adaptive policy outperforms uniform-χ baselines by recovering 1.2% additional accuracy at matched compression ratios; extended with soft gating adapters that modulate MPO output scale without breaking the tensor decomposition structure.
- Performed full healing (one epoch, Alpaca instruction dataset) on each compressed model variant; ran final benchmarks via `lm-evaluation-harness` across MMLU, HellaSwag, BoolQ, TriviaQA, and GSM8K; demonstrated that 70% parameter reduction with adaptive scheduling incurs only a 2–3% accuracy drop, replicating and extending the original paper's conclusions at a smaller but computationally verifiable scale.

Education

JSS Academy of Technical Education — *B.Tech., Mechanical Engineering* – CGPA: 7.4 December 2020 – July 2024 / 10.0
Noida, India

- **Relevant Coursework:** Data Structures & Algorithms, Linear Algebra, Probability & Statistics, Database Systems
- **Independent Study:** Distributed Systems, Machine Learning Theory, Compiler Design, Cryptography, System Architecture, Deep Learning, Natural Language Processing

Certifications

Massachusetts Institute of Technology – MITx — Advanced Machine Learning and Deep Learning *(Verified Certificate)*

- Completed rigorous graduate-level coursework covering supervised and unsupervised learning, deep neural network architectures, optimisation theory, convolutional and recurrent networks, and modern deep learning techniques; curriculum equivalent to MIT's on-campus ML sequence.