**Aditya's Modern LLM + Memory Research Course**

**24 Weeks | 200+ Videos | Pure Research Frontier**

**No old shit. Pure 2024-2025 LLM research.**

**📱 CONTENT FORMAT PER WEEK**

**Each week = 1 major topic with:**

* **5-7 Deep Videos (15-25 min each) - YouTube**
* **10-15 Shorts (60s each) - Instagram/YouTube Shorts**
* **1 Long Blog Post**
* **Working Code**

**Total: ~150 deep videos + 250+ shorts over 24 weeks**

**🎯 PART 1: MODERN TRANSFORMER ARCHITECTURE (Weeks 1-3)**

**Week 1: Attention Mechanisms (7 videos)**

**Video 1 (20 min): "Self-Attention Explained"**

* **Pure attention mechanism**
* **Q, K, V mathematics**
* **Attention formula breakdown**
* **Visual explanations**

**Video 2 (18 min): "Multi-Head Attention"**

* **Why multiple heads**
* **Parallel processing**
* **Head specialization**
* **Implementation**

**Video 3 (22 min): "Grouped-Query Attention (GQA)"**

* **Multi-Query Attention (MQA)**
* **GQA architecture**
* **Memory savings**
* **Llama 2/3 usage**

**Video 4 (20 min): "Multi-Head Latent Attention (MLA)"**

* **DeepSeek innovation**
* **Latent compression**
* **KV cache reduction**
* **Performance gains**

**Video 5 (18 min): "Sliding Window Attention"**

* **Local attention windows**
* **Mistral architecture**
* **Memory efficiency**
* **Implementation**

**Video 6 (25 min): "Flash Attention 1, 2, 3"**

* **IO-aware attention**
* **Tiling strategy**
* **Memory hierarchy**
* **Integration guide**

**Video 7 (20 min): "Coding All Attention Variants"**

* **Standard attention**
* **GQA implementation**
* **Sliding window**
* **Flash Attention integration**

**Shorts (15 clips):**

1. **"Self-Attention in 60s"**
2. **"GQA vs MQA vs MHA"**
3. **"Why Flash Attention is 4x Faster"**
4. **"Sliding Window Attention"**
5. **"MLA: DeepSeek's Secret"**
6. **"Attention Memory Cost"**
7. **"Multi-Head Visualization"**
8. **"Flash Attention Explained"**
9. **"GQA in Llama 3"**
10. **"Attention Complexity"**
11. **"KV Cache Problem"**
12. **"Local vs Global Attention"**
13. **"Sparse Attention Patterns"**
14. **"Attention Optimization"**
15. **"Which Attention to Use?"**

**Code: modern\_attention.py (all variants)**

**Blog: "Modern Attention Mechanisms: 2025 Guide"**

**Week 2: Positional Encodings (6 videos)**

**Video 1 (18 min): "RoPE: Rotary Position Embeddings"**

* **Rotary matrix mathematics**
* **Relative position encoding**
* **Why it works**
* **Implementation**

**Video 2 (20 min): "ALiBi: Attention with Linear Biases"**

* **Linear bias mechanism**
* **Extrapolation capability**
* **Train short, test long**
* **Comparison with RoPE**

**Video 3 (22 min): "YaRN: Context Extension"**

* **RoPE frequency scaling**
* **NTK-aware interpolation**
* **Extending to 128k+**
* **YaRN parameters**

**Video 4 (16 min): "NoPE: No Positional Embeddings"**

* **Why some models skip positions**
* **Implicit position learning**
* **SmolLM3 approach**
* **Trade-offs**

**Video 5 (20 min): "Position Encoding Comparison"**

* **RoPE vs ALiBi vs YaRN**
* **Extrapolation tests**
* **Memory costs**
* **Which to choose**

**Video 6 (25 min): "Implementing All Position Encodings"**

* **RoPE from scratch**
* **ALiBi implementation**
* **YaRN scaling**
* **Benchmarking**

**Shorts (12 clips):**

1. **"RoPE Explained"**
2. **"ALiBi vs RoPE"**
3. **"YaRN Context Extension"**
4. **"NoPE: No Positions?"**
5. **"RoPE Frequency Scaling"**
6. **"Position Extrapolation"**
7. **"RoPE Math Simplified"**
8. **"ALiBi Train-Short-Test-Long"**
9. **"YaRN Parameters"**
10. **"Which Position Encoding?"**
11. **"Context Length Extension"**
12. **"Position in Modern LLMs"**

**Code: position\_encodings.py**

**Blog: "Position Encodings: RoPE, ALiBi, YaRN Compared"**

**Week 3: Modern Architecture Components (7 videos)**

**Video 1 (18 min): "Layer Normalization Evolution"**

* **LayerNorm basics**
* **RMSNorm (Llama, Mistral)**
* **Pre-norm vs Post-norm**
* **Normalization placement**

**Video 2 (20 min): "Advanced Feed-Forward Networks"**

* **Standard FFN**
* **SwiGLU activation**
* **GeGLU variants**
* **MoE FFN preview**

**Video 3 (22 min): "Residual Connections & Skip"**

* **Why residuals matter**
* **Gradient flow**
* **Scaling residuals**
* **Modern practices**

**Video 4 (25 min): "Complete Transformer Block"**

* **Attention layer**
* **FFN layer**
* **Normalization**
* **Putting it together**

**Video 5 (20 min): "Decoder-Only Architecture"**

* **Why decoder-only won**
* **GPT architecture**
* **Llama architecture**
* **Modern trends**

**Video 6 (18 min): "Model Initialization"**

* **Weight initialization**
* **Scaling strategies**
* **muP (maximal update parameterization)**
* **Stable training**

**Video 7 (28 min): "Building Complete GPT"**

* **Full architecture code**
* **Training setup**
* **Generation**
* **Debugging**

**Shorts (14 clips):**

1. **"RMSNorm vs LayerNorm"**
2. **"SwiGLU Activation"**
3. **"Why Residuals Work"**
4. **"Decoder-Only Architecture"**
5. **"Pre-Norm vs Post-Norm"**
6. **"Transformer Block Breakdown"**
7. **"GPT Architecture"**
8. **"FFN Variants"**
9. **"Weight Initialization"**
10. **"Model Stability"**
11. **"Gradient Flow"**
12. **"Modern Architecture"**
13. **"Complete Transformer"**
14. **"Training Setup"**

**Code: modern\_transformer.py (complete implementation)**

**Blog: "Building Modern Transformers: 2025 Edition"**

**🔥 PART 2: TRAINING & OPTIMIZATION (Weeks 4-6)**

**Week 4: Training Fundamentals (6 videos)**

**Video 1 (20 min): "Modern Optimizers"**

* **AdamW deep dive**
* **Lion optimizer**
* **Sophia optimizer**
* **Adafactor**
* **Which to use**

**Video 2 (22 min): "Learning Rate Schedules"**

* **Warmup strategies**
* **Cosine annealing**
* **Constant LR with warmup**
* **Linear decay**
* **WSD schedule**

**Video 3 (18 min): "Gradient Clipping & Stability"**

* **Global gradient clipping**
* **Per-parameter clipping**
* **Gradient norm monitoring**
* **Stability tricks**

**Video 4 (25 min): "Mixed Precision Training"**

* **FP16 vs BF16**
* **Loss scaling**
* **Gradient accumulation**
* **Memory savings**

**Video 5 (20 min): "Batch Size & Gradient Accumulation"**

* **Effective batch size**
* **Gradient accumulation**
* **Large batch training**
* **Critical batch size**

**Video 6 (24 min): "Complete Training Loop"**

* **Training harness**
* **Checkpointing**
* **Logging (WandB)**
* **Monitoring**

**Shorts (12 clips):**

1. **"AdamW vs Adam"**
2. **"Lion Optimizer"**
3. **"LR Warmup Explained"**
4. **"Cosine Annealing"**
5. **"Gradient Clipping"**
6. **"FP16 vs BF16"**
7. **"Gradient Accumulation"**
8. **"Batch Size Matters"**
9. **"Loss Scaling"**
10. **"Training Stability"**
11. **"Checkpointing Strategy"**
12. **"Monitoring Metrics"**

**Code: training\_modern.py**

**Blog: "Modern LLM Training: Complete Guide"**

**Week 5: Scaling Laws & Compute (6 videos)**

**Video 1 (22 min): "Chinchilla Scaling Laws"**

* **Optimal compute allocation**
* **Parameter vs token scaling**
* **20 tokens per parameter**
* **Compute budget**

**Video 2 (20 min): "Data Scaling"**

* **Data quantity vs quality**
* **Data mixing ratios**
* **Deduplication impact**
* **Data curriculum**

**Video 3 (18 min): "Model Scaling Predictions"**

* **Predicting loss**
* **Performance estimation**
* **When to stop training**
* **Scaling experiments**

**Video 4 (25 min): "Compute-Optimal Training"**

* **Budget allocation**
* **Parameter selection**
* **Token count planning**
* **Practical application**

**Video 5 (20 min): "Post-Training Scaling"**

* **Inference-time compute**
* **Test-time scaling**
* **o1-style reasoning**
* **Thinking tokens**

**Video 6 (22 min): "Scaling Law Calculator"**

* **Building prediction tools**
* **Loss estimation**
* **Budget optimizer**
* **Practical examples**

**Shorts (10 clips):**

1. **"Chinchilla Laws Explained"**
2. **"20 Tokens per Parameter"**
3. **"Compute Budget Allocation"**
4. **"Data vs Parameters"**
5. **"Scaling Predictions"**
6. **"When to Stop Training"**
7. **"Post-Training Compute"**
8. **"Test-Time Scaling"**
9. **"o1 Reasoning"**
10. **"Scaling Calculator"**

**Code: scaling\_laws.py**

**Blog: "Chinchilla Scaling Laws: Practical Guide"**

**Week 6: Data Processing (7 videos)**

**Video 1 (20 min): "Data Collection Strategies"**

* **Web scraping**
* **CommonCrawl**
* **FineWeb-Edu**
* **Curated datasets**

**Video 2 (22 min): "Data Filtering & Quality"**

* **Quality filters**
* **Perplexity filtering**
* **Classifier-based filtering**
* **Heuristic filters**

**Video 3 (25 min): "Deduplication"**

* **Exact deduplication**
* **Fuzzy deduplication**
* **MinHash LSH**
* **Document-level vs line-level**

**Video 4 (18 min): "Tokenization at Scale"**

* **BPE training**
* **Vocabulary selection**
* **Multilingual tokenization**
* **Special tokens**

**Video 5 (20 min): "Data Mixing & Curriculum"**

* **Domain mixing**
* **Upsampling/downsampling**
* **Curriculum learning**
* **Annealing strategies**

**Video 6 (22 min): "Data Pipeline Architecture"**

* **Distributed processing**
* **Sharding strategies**
* **Storage optimization**
* **Streaming data**

**Video 7 (24 min): "Building Complete Data Pipeline"**

* **End-to-end implementation**
* **Quality checks**
* **Processing pipeline**
* **Monitoring**

**Shorts (14 clips):**

1. **"Data Quality > Quantity"**
2. **"FineWeb-Edu Dataset"**
3. **"Deduplication Explained"**
4. **"MinHash LSH"**
5. **"Data Filtering"**
6. **"Tokenizer Training"**
7. **"Data Mixing Ratios"**
8. **"Curriculum Learning"**
9. **"Data Pipeline"**
10. **"Perplexity Filtering"**
11. **"Quality Metrics"**
12. **"Data Streaming"**
13. **"Sharding Strategy"**
14. **"Data Processing"**

**Code: data\_pipeline/ (complete pipeline)**

**Blog: "LLM Data Processing: Production Pipeline"**

**🧠 PART 3: MEMORY SYSTEMS (Weeks 7-12)**

**Week 7: Long Context Problem (6 videos)**

**Video 1 (20 min): "The Context Length Crisis"**

* **O(n²) complexity**
* **Memory bottleneck**
* **Attention cost analysis**
* **Why 1M context is hard**

**Video 2 (22 min): "KV Cache Deep Dive"**

* **What is KV cache**
* **Memory calculation**
* **Cache management**
* **Optimization strategies**

**Video 3 (18 min): "Lost in the Middle"**

* **Long context challenges**
* **Information retrieval**
* **Position bias**
* **Mitigation strategies**

**Video 4 (25 min): "Long Context Benchmarks"**

* **RULER benchmark**
* **Needle in haystack**
* **LongBench**
* **ZeroSCROLLS**

**Video 5 (20 min): "Memory Profiling"**

* **VRAM tracking**
* **Attention cost**
* **Throughput analysis**
* **Bottleneck identification**

**Video 6 (24 min): "Context Length Experiments"**

* **Scaling experiments**
* **Quality degradation**
* **Cost analysis**
* **Practical limits**

**Shorts (12 clips):**

1. **"O(n²) Problem Visual"**
2. **"KV Cache Explained"**
3. **"Lost in Middle"**
4. **"1M Context Cost"**
5. **"Memory Bottleneck"**
6. **"Attention Complexity"**
7. **"Long Context Benchmarks"**
8. **"Needle in Haystack"**
9. **"Context Window"**
10. **"Memory Profiling"**
11. **"VRAM Breakdown"**
12. **"Context Limits"**

**Code: long\_context\_profiler.py**

**Blog: "Long Context: The Memory Crisis"**

**Week 8: Transformer-XL & Recurrence (6 videos)**

**Video 1 (22 min): "Transformer-XL Architecture"**

* **Segment-level recurrence**
* **Memory mechanism**
* **Addressing scheme**
* **Benefits**

**Video 2 (20 min): "Relative Positional Encodings"**

* **Why relative positions**
* **Mathematics**
* **Implementation**
* **Comparison**

**Video 3 (25 min): "Segment Caching Strategy"**

* **Cache management**
* **Rolling memory**
* **Segment size**
* **Trade-offs**

**Video 4 (18 min): "Training Transformer-XL"**

* **Training modifications**
* **Backpropagation through time**
* **Gradient flow**
* **Stability**

**Video 5 (22 min): "Compressive Transformers"**

* **Compression mechanism**
* **Old memory compression**
* **Compression functions**
* **Quality trade-offs**

**Video 6 (28 min): "Implementing Recurrent Memory"**

* **Complete implementation**
* **Caching logic**
* **Compression**
* **Benchmarking**

**Shorts (12 clips):**

1. **"Transformer-XL Overview"**
2. **"Segment Recurrence"**
3. **"Relative Positions"**
4. **"Rolling Cache"**
5. **"Memory Compression"**
6. **"Compressive Transformers"**
7. **"Segment Size"**
8. **"Memory Trade-offs"**
9. **"Training Modifications"**
10. **"Recurrent Attention"**
11. **"Cache Management"**
12. **"Context Extension"**

**Code: transformer\_xl.py, compressive\_transformer.py**

**Blog: "Transformer-XL & Compressive Memory"**

**Week 9: Retrieval-Augmented Memory (7 videos)**

**Video 1 (20 min): "RAG Fundamentals"**

* **Retrieval-augmented generation**
* **Architecture overview**
* **When to use RAG**
* **Benefits and limitations**

**Video 2 (22 min): "kNN-LM: Non-Parametric Memory"**

* **kNN language models**
* **Datastore creation**
* **Nearest neighbor search**
* **Interpolation**

**Video 3 (25 min): "RETRO: Retrieval-Enhanced Transformers"**

* **RETRO architecture**
* **Chunked cross-attention**
* **Retrieval mechanism**
* **Training**

**Video 4 (20 min): "FAISS for Billion-Scale Search"**

* **FAISS library**
* **Index types**
* **Scaling to billions**
* **Optimization**

**Video 5 (18 min): "Embedding Models for Retrieval"**

* **Dense retrievers**
* **Contrastive learning**
* **Embedding quality**
* **Model selection**

**Video 6 (22 min): "Retrieval Strategies"**

* **Top-k retrieval**
* **Reranking**
* **Hybrid search**
* **Query augmentation**

**Video 7 (28 min): "Building Complete RAG System"**

* **End-to-end implementation**
* **Datastore building**
* **Retrieval pipeline**
* **Evaluation**

**Shorts (14 clips):**

1. **"RAG Explained"**
2. **"kNN-LM Overview"**
3. **"RETRO Architecture"**
4. **"FAISS for Search"**
5. **"Non-Parametric Memory"**
6. **"Retrieval vs Generation"**
7. **"Building Datastore"**
8. **"Embedding Models"**
9. **"Top-k Retrieval"**
10. **"Hybrid Search"**
11. **"Reranking"**
12. **"RAG Benefits"**
13. **"When to Use RAG"**
14. **"RAG Limitations"**

**Code: rag\_system/ (complete RAG)**

**Blog: "Retrieval-Augmented Generation: Complete Guide"**

**Week 10: Product-Key Memory (PKM) (6 videos)**

**Video 1 (22 min): "Product-Key Memory Architecture"**

* **External memory module**
* **Product quantization**
* **A×B codebook**
* **Why it works**

**Video 2 (20 min): "Product Quantization Math"**

* **Quantization theory**
* **Product space**
* **Key selection**
* **Value retrieval**

**Video 3 (25 min): "Training PKM"**

* **Integration with Transformer**
* **Loss functions**
* **Gradient flow**
* **Stability**

**Video 4 (18 min): "Collision Analysis"**

* **Key collisions**
* **Utilization metrics**
* **Addressing strategies**
* **Quality impact**

**Video 5 (22 min): "PKM Ablation Studies"**

* **Memory size**
* **Number of heads**
* **Top-k selection**
* **Temperature**

**Video 6 (28 min): "Implementing PKM from Scratch"**

* **Complete implementation**
* **Training loop**
* **Evaluation**
* **Optimization**

**Shorts (12 clips):**

1. **"PKM Explained"**
2. **"Product Quantization"**
3. **"A×B Codebook"**
4. **"Trainable Memory"**
5. **"Key Selection"**
6. **"Memory Collisions"**
7. **"PKM vs Attention"**
8. **"External Memory"**
9. **"Memory Scaling"**
10. **"PKM Training"**
11. **"Ablation Results"**
12. **"When to Use PKM"**

**Code: product\_key\_memory.py**

**Blog: "Product-Key Memory: Trainable External Memory"**

**Week 11: KV Cache Optimization (7 videos)**

**Video 1 (20 min): "KV Cache as Bottleneck"**

* **Memory analysis**
* **Scaling issues**
* **Inference costs**
* **Optimization motivation**

**Video 2 (22 min): "SnapKV: Attention-Based Eviction"**

* **Importance scoring**
* **Head-wise selection**
* **Top-p retention**
* **Quality metrics**

**Video 3 (25 min): "StreamingLLM: Sink Tokens"**

* **Attention sink phenomenon**
* **Sink + window**
* **Rolling KV cache**
* **Infinite length**

**Video 4 (20 min): "KV Cache Quantization"**

* **8-bit quantization**
* **4-bit quantization**
* **Per-channel/per-token**
* **Quality degradation**

**Video 5 (18 min): "H2O: Heavy Hitters"**

* **Local + heavy hitters**
* **Dynamic selection**
* **Memory savings**
* **Performance**

**Video 6 (22 min): "Unified KV Policy Framework"**

* **Policy interface**
* **Comparison framework**
* **Benchmarking**
* **Best practices**

**Video 7 (28 min): "Implementing All KV Policies"**

* **SnapKV implementation**
* **StreamingLLM**
* **Quantization**
* **H2O**
* **Evaluation**

**Shorts (14 clips):**

1. **"KV Cache Bottleneck"**
2. **"SnapKV Explained"**
3. **"Attention Sink"**
4. **"StreamingLLM"**
5. **"KV Quantization"**
6. **"4-bit KV Cache"**
7. **"H2O Policy"**
8. **"Heavy Hitters"**
9. **"Cache Eviction"**
10. **"Memory Savings"**
11. **"Quality Trade-offs"**
12. **"Best KV Policy"**
13. **"Infinite Context"**
14. **"KV Optimization"**

**Code: kv\_policies/ (all policies)**

**Blog: "KV Cache Optimization: Complete Survey"**

**Week 12: Infini-Attention & Modern Memory (6 videos)**

**Video 1 (22 min): "Infini-Attention Architecture"**

* **Compressive memory**
* **Local + global attention**
* **Memory updates**
* **Google's approach**

**Video 2 (20 min): "Memory as Associative Array"**

* **Key-value memory**
* **Update rules**
* **Retrieval mechanism**
* **Delta rule**

**Video 3 (25 min): "Implementing Infini-Attention"**

* **Architecture details**
* **Memory module**
* **Training**
* **Evaluation**

**Video 4 (18 min): "RMT: Recurrent Memory Transformer"**

* **Segment-based processing**
* **Memory tokens**
* **Gradient flow**
* **Long context**

**Video 5 (22 min): "Memory Systems Comparison"**

* **Transformer-XL**
* **Compressive**
* **PKM**
* **Infini-Attention**
* **Trade-off analysis**

**Video 6 (28 min): "Building Hybrid Memory System"**

* **Combining approaches**
* **Best of each**
* **Implementation**
* **Benchmarking**

**Shorts (12 clips):**

1. **"Infini-Attention"**
2. **"Compressive Memory"**
3. **"Local + Global"**
4. **"Memory Updates"**
5. **"Associative Memory"**
6. **"RMT Overview"**
7. **"Memory Tokens"**
8. **"Hybrid Memory"**
9. **"Memory Comparison"**
10. **"Best Memory System"**
11. **"Memory Trade-offs"**
12. **"Modern Memory"**

**Code: infini\_attention.py, memory\_systems.py**

**Blog: "Modern Memory Systems: Complete Survey"**

**⚡ PART 4: EFFICIENCY & SYSTEMS (Weeks 13-15)**

**Week 13: Flash Attention & Kernels (6 videos)**

**Video 1 (22 min): "GPU Memory Hierarchy"**

* **HBM vs SRAM**
* **Memory bandwidth**
* **IO bottleneck**
* **Roofline model**

**Video 2 (25 min): "Flash Attention Algorithm"**

* **Tiling strategy**
* **Online softmax**
* **Block-sparse attention**
* **Mathematics**

**Video 3 (20 min): "Flash Attention 2"**

* **Improvements over v1**
* **Work partitioning**
* **Performance gains**
* **Adoption**

**Video 4 (22 min): "Flash Attention 3"**

* **Latest optimizations**
* **Producer-consumer overlap**
* **Warp specialization**
* **H100 optimization**

**Video 5 (18 min): "Integrating Flash Attention"**

* **PyTorch integration**
* **xFormers**
* **Memory savings**
* **Speed improvements**

**Video 6 (28 min): "Custom CUDA Kernels"**

* **Triton basics**
* **Kernel optimization**
* **Profiling**
* **Best practices**

**Shorts (12 clips):**

1. **"HBM vs SRAM"**
2. **"Flash Attention Explained"**
3. **"4x Speedup How?"**
4. **"Tiling Strategy"**
5. **"Flash Attention 2"**
6. **"Flash Attention 3"**
7. **"Memory Bandwidth"**
8. **"IO Bottleneck"**
9. **"Triton Kernels"**
10. **"CUDA Optimization"**
11. **"Kernel Profiling"**
12. **"xFormers Library"**

**Code: flash\_attention/, triton\_kernels/**

**Blog: "Flash Attention: Deep Dive"**

**Week 14: Parallelism & Distributed Training (7 videos)**

**Video 1 (22 min): "Data Parallelism (DDP)"**

* **Distributed data parallel**
* **AllReduce**
* **Gradient synchronization**
* **Scaling efficiency**

**Video 2 (25 min): "Tensor Parallelism"**

* **Model sharding**
* **Megatron-LM**
* **Column/row parallelism**
* **Communication**

**Video 3 (20 min): "Pipeline Parallelism"**

* **Micro-batching**
* **Pipeline stages**
* **GPipe and variants**
* **Bubble mitigation**

**Video 4 (22 min): "Fully Sharded Data Parallel (FSDP)"**

* **ZeRO optimizer**
* **Sharding strategies**
* **Memory savings**
* **Performance**

**Video 5 (18 min): "3D Parallelism"**

* **Combining DP, TP, PP**
* **Optimal configuration**
* **Real-world examples**
* **Scaling laws**

**Video 6 (20 min): "DeepSpeed & Accelerate"**

* **DeepSpeed features**
* **Accelerate library**
* **Configuration**
* **Best practices**

**Video 7 (28 min): "Multi-Node Training Setup"**

* **Cluster setup**
* **NCCL configuration**
* **Debugging**
* **Monitoring**

**Shorts (14 clips):**

1. **"Data Parallelism"**
2. **"Tensor Parallelism"**
3. **"Pipeline Parallelism"**
4. **"FSDP Explained"**
5. **"ZeRO Optimizer"**
6. **"3D Parallelism"**
7. **"DeepSpeed Features"**
8. **"Accelerate Library"**
9. **"Multi-Node Training"**
10. **"AllReduce Operation"**
11. **"Model Sharding"**
12. **"Pipeline Bubbles"**
13. **"NCCL Configuration"**
14. **"Distributed Debugging"**

**Code: distributed\_training/**

**Blog: "Distributed Training: Complete Guide"**

**Week 15: Inference Optimization (6 videos)**

**Video 1 (20 min): "vLLM & PagedAttention"**

* **PagedAttention algorithm**
* **KV cache as virtual memory**
* **Continuous batching**
* **Throughput optimization**

**Video 2 (22 min): "Speculative Decoding"**

* **Draft model approach**
* **Verification**
* **Speedup analysis**
* **Implementation**

**Video 3 (25 min): "Quantization for Inference"**

* **INT8 quantization**
* **GPTQ**
* **AWQ**
* **Quality preservation**

**Video 4 (20 min): "Model Compilation"**

* **TorchScript**
* **ONNX export**
* **TensorRT**
* **Optimization**

**Video 5 (18 min): "Batching Strategies"**

* **Static batching**
* **Dynamic batching**
* **Continuous batching**
* **Request scheduling**

**Video 6 (28 min): "Production Inference System"**

* **End-to-end setup**
* **vLLM deployment**
* **Monitoring**
* **Scaling**

**Shorts (12 clips):**

1. **"vLLM Explained"**
2. **"PagedAttention"**
3. **"Continuous Batching"**
4. **"Speculative Decoding"**
5. **"INT8 Quantization"**
6. **"GPTQ vs AWQ"**
7. **"Model Compilation"**
8. **"TensorRT"**
9. **"Batching Strategies"**
10. **"Inference Throughput"**
11. **"KV Cache Management"**
12. **"Production Serving"**

**Code: inference\_systems/**

**Blog: "LLM Inference: Production Guide"**

**🎯 PART 5: MIXTURE OF EXPERTS (Weeks 16-17)**

**Week 16: MoE Fundamentals (7 videos)**

**Video 1 (22 min): "Mixture of Experts Overview"**

* **Sparse MoE concept**
* **Router mechanism**
* **Expert specialization**
* **Benefits**

**Video 2 (25 min): "MoE Architecture Details"**

* **Top-k routing**
* **Expert layers**
* **Load balancing**
* **Capacity factor**

**Video 3 (20 min): "Router Design"**

* **Router types**
* **Gating function**
* **Expert selection**
* **Training dynamics**

**Video 4 (22 min): "Load Balancing in MoE"**

* **Load balancing loss**
* **Auxiliary losses**
* **Router z-loss**
* **Expert capacity**

**Video 5 (18 min): "MoE Training Challenges"**

* **Expert collapse**
* **Load imbalance**
* **Gradient issues**
* **Stabilization**

**Video 6 (20 min): "Fine-Grained MoE"**

* **Mixtral architecture**
* **Granular experts**
* **Modern designs**
* **Performance**

**Video 7 (28 min): "Implementing MoE from Scratch"**

* **Router implementation**
* **Expert layers**
* **Load balancing**
* **Training loop**

**Shorts (14 clips):**

1. **"MoE Explained"**
2. **"Sparse vs Dense"**
3. **"Top-k Routing"**
4. **"Expert Specialization"**
5. **"Load Balancing"**
6. **"Mixtral Architecture"**
7. **"MoE Training"**
8. **"Expert Collapse"**
9. **"Router Design"**
10. **"Capacity Factor"**
11. **"MoE Benefits"**
12. **"When to Use MoE"**
13. **"MoE Challenges"**
14. **"Modern MoE"**

**Code: mixture\_of\_experts/**

**Blog: "Mixture of Experts: Complete Guide"**

**Week 17: Advanced MoE (6 videos)**

**Video 1 (22 min): "DeepSeek-V3 MoE"**

* **Multi-head latent attention**
* **MoE design**
* **Auxiliary-loss-free**
* **Performance**

**Video 2 (20 min): "Llama 4 MoE"**

* **Architecture details**
* **Expert configuration**
* **Training approach**
* **Results**

**Video 3 (25 min): "Expert Specialization Analysis"**

* **What experts learn**
* **Domain specialization**
* **Visualization**
* **Interpretability**

**Video 4 (18 min): "MoE Inference Optimization"**

* **Expert caching**
* **Batching strategies**
* **Memory management**
* **Throughput**

**Video 5 (22 min): "Soft MoE & Alternatives"**

* **Soft merging**
* **Expert pruning**
* **Distillation**
* **Trade-offs**

**Video 6 (28 min): "Production MoE System"**

* **Training pipeline**
* **Inference serving**
* **Monitoring**
* **Optimization**

**Shorts (12 clips):**

1. **"DeepSeek-V3 MoE"**
2. **"Llama 4 Experts"**
3. **"Expert Specialization"**
4. **"MoE Inference"**
5. **"Soft MoE"**
6. **"Expert Caching"**
7. **"MoE Distillation"**
8. **"Production MoE"**
9. **"MoE Monitoring"**
10. **"Expert Pruning"**
11. **"MoE Trade-offs"**
12. **"Modern MoE Designs"**

**Code: advanced\_moe/**

**Blog: "Advanced MoE: DeepSeek & Llama 4"**

**🔄 PART 6: ALIGNMENT & RLHF (Weeks 18-19)**

**Week 18: Supervised Fine-Tuning (6 videos)**

**Video 1 (20 min): "SFT Fundamentals"**

* **Instruction tuning**
* **Dataset creation**
* **Loss functions**
* **Hyperparameters**

**Video 2 (22 min): "Instruction Dataset Design"**

* **Prompt formats**
* **Quality criteria**
* **Diversity**
* **Coverage**

**Video 3 (25 min): "Chat Templates"**

* **Message formatting**
* **System prompts**
* **Special tokens**
* **Best practices**

**Video 4 (18 min): "Multi-Turn Conversations"**

* **Context management**
* **Turn-level training**
* **Masking strategies**
* **Quality**

**Video 5 (20 min): "Domain-Specific SFT"**

* **Code generation**
* **Math reasoning**
* **Tool use**
* **Specialization**

**Video 6 (28 min): "Complete SFT Pipeline"**

* **Data preparation**
* **Training loop**
* **Evaluation**
* **Deployment**

**Shorts (12 clips):**

1. **"SFT Explained"**
2. **"Instruction Tuning"**
3. **"Chat Templates"**
4. **"Prompt Formatting"**
5. **"Multi-Turn Training"**
6. **"Dataset Quality"**
7. **"System Prompts"**
8. **"Domain SFT"**
9. **"Code Tuning"**
10. **"Math Reasoning"**
11. **"Tool Use"**
12. **"SFT Best Practices"**

**Code: sft\_pipeline/**

**Blog: "Supervised Fine-Tuning: Complete Guide"**

**Week 19: RLHF & Alignment (7 videos)**

**Video 1 (22 min): "RLHF Overview"**

* **Reinforcement learning basics**
* **Human feedback loop**
* **Reward modeling**
* **Policy optimization**

**Video 2 (25 min): "Reward Model Training"**

* **Preference data**
* **Bradley-Terry model**
* **Model architecture**
* **Training process**

**Video 3 (20 min): "PPO for LLMs"**

* **Proximal policy optimization**
* **KL divergence constraint**
* **Value function**
* **Implementation**

**Video 4 (22 min): "DPO: Direct Preference Optimization"**

* **Avoiding RL**
* **Direct optimization**
* **Mathematics**
* **Advantages**

**Video 5 (18 min): "RLAIF: AI Feedback"**

* **Constitutional AI**
* **Self-improvement**
* **Scalable oversight**
* **Results**

**Video 6 (20 min): "Online vs Offline RL"**

* **Trade-offs**
* **Sample efficiency**
* **Stability**
* **Best practices**

**Video 7 (28 min): "Complete RLHF Pipeline"**

* **Preference collection**
* **Reward training**
* **PPO/DPO implementation**
* **Evaluation**

**Shorts (14 clips):**

1. **"RLHF Explained"**
2. **"Reward Modeling"**
3. **"PPO Algorithm"**
4. **"DPO vs PPO"**
5. **"RLAIF Overview"**
6. **"Preference Data"**
7. **"KL Constraint"**
8. **"Constitutional AI"**
9. **"Online RL"**
10. **"Offline RL"**
11. **"Alignment Tax"**
12. **"Safety Alignment"**
13. **"RLHF Challenges"**
14. **"Best Alignment"**

**Code: rlhf\_pipeline/**

**Blog: "RLHF & DPO: Complete Alignment Guide"**

**🧪 PART 7: INDIC RESEARCH (Weeks 20-24)**

**Week 20: Benchmark Design (6 videos)**

**Video 1 (22 min): "LLCE-Indic: Motivation"**

* **Why Indic languages**
* **Long-context challenges**
* **Existing gaps**
* **Our contribution**

**Video 2 (25 min): "Task Taxonomy"**

* **Needle in haystack**
* **Temporal reasoning**
* **Cross-lingual QA**
* **Aggregation tasks**
* **Contradiction resolution**

**Video 3 (20 min): "Data Sources"**

* **Government documents (RBI, SEBI)**
* **Legal corpus**
* **Educational materials**
* **Licensing**

**Video 4 (22 min): "Schema Design"**

* **Metadata structure**
* **Annotation format**
* **Quality checks**
* **Validation**

**Video 5 (18 min): "Multilingual Challenges"**

* **Code-mixing**
* **Script variations**
* **Translation quality**
* **Evaluation metrics**

**Video 6 (28 min): "Benchmark Infrastructure"**

* **Scraping pipeline**
* **Storage**
* **API design**
* **Access**

**Shorts (12 clips):**

1. **"LLCE-Indic Overview"**
2. **"Indic Languages"**
3. **"Long Context Tasks"**
4. **"Government Data"**
5. **"Temporal Reasoning"**
6. **"Cross-Lingual QA"**
7. **"Code-Mixing"**
8. **"Benchmark Design"**
9. **"Data Quality"**
10. **"Multilingual Eval"**
11. **"Task Types"**
12. **"Why This Matters"**

**Code: llce\_indic/schema/**

**Blog: "Introducing LLCE-Indic Benchmark"**

**Week 21: Data Collection (6 videos)**

**Video 1 (20 min): "Web Scraping Infrastructure"**

* **Scrapy framework**
* **Rate limiting**
* **Error handling**
* **Monitoring**

**Video 2 (22 min): "PDF Extraction"**

* **OCR for scanned docs**
* **Layout preservation**
* **Table extraction**
* **Quality**

**Video 3 (25 min): "Text Cleaning Pipeline"**

* **Noise removal**
* **Format normalization**
* **Language detection**
* **Validation**

**Video 4 (18 min): "Deduplication Strategies"**

* **Near-duplicate detection**
* **Fuzzy matching**
* **Threshold tuning**
* **Efficiency**

**Video 5 (20 min): "Data Statistics"**

* **Corpus analysis**
* **Language distribution**
* **Length statistics**
* **Quality metrics**

**Video 6 (28 min): "Complete Data Pipeline"**

* **End-to-end flow**
* **Monitoring**
* **Quality gates**
* **Deployment**

**Shorts (12 clips):**

1. **"Web Scraping"**
2. **"PDF Extraction"**
3. **"OCR for Indic"**
4. **"Text Cleaning"**
5. **"Deduplication"**
6. **"Quality Checks"**
7. **"Data Pipeline"**
8. **"Corpus Stats"**
9. **"Language Detection"**
10. **"Noise Removal"**
11. **"Format Normalization"**
12. **"Data Quality"**

**Code: llce\_indic/pipeline/**

**Blog: "Building LLCE-Indic: Data Pipeline"**

**Week 22: Annotation (7 videos)**

**Video 1 (22 min): "Annotation Guidelines"**

* **Task instructions**
* **Quality criteria**
* **Edge cases**
* **Training annotators**

**Video 2 (25 min): "LLM-Assisted Annotation"**

* **Prompt design**
* **Quality control**
* **Human verification**
* **Efficiency gains**

**Video 3 (20 min): "Temporal Task Creation"**

* **Update chains**
* **Version tracking**
* **Current vs outdated**
* **Validation**

**Video 4 (22 min): "Cross-Lingual Tasks"**

* **Question in one language**
* **Answer in another**
* **Translation quality**
* **Difficulty levels**

**Video 5 (18 min): "Quality Assurance"**

* **Inter-annotator agreement**
* **Validation sets**
* **Error analysis**
* **Iterative improvement**

**Video 6 (20 min): "Annotation Tools"**

* **Web interface**
* **Keyboard shortcuts**
* **Progress tracking**
* **Export formats**

**Video 7 (28 min): "Final Dataset Analysis"**

* **Task distribution**
* **Difficulty analysis**
* **Coverage metrics**
* **Release preparation**

**Shorts (14 clips):**

1. **"Annotation Process"**
2. **"LLM Assistance"**
3. **"Quality Control"**
4. **"Temporal Tasks"**
5. **"Cross-Lingual QA"**
6. **"IAA Metrics"**
7. **"Annotation Tools"**
8. **"Task Distribution"**
9. **"Difficulty Levels"**
10. **"Edge Cases"**
11. **"Validation"**
12. **"Dataset Quality"**
13. **"Release Prep"**
14. **"Annotation Best Practices"**

**Code: llce\_indic/annotation/**

**Blog: "LLCE-Indic Annotation Strategy"**

**Week 23: Experiments (7 videos)**

**Video 1 (22 min): "Baseline Models"**

* **Llama 3.1**
* **Mistral**
* **DeepSeek**
* **Gemma**
* **Setup**

**Video 2 (25 min): "Memory Systems Evaluation"**

* **Transformer-XL**
* **Compressive**
* **PKM**
* **kNN-LM**
* **Comparison**

**Video 3 (20 min): "KV Cache Policies"**

* **SnapKV**
* **StreamingLLM**
* **H2O**
* **Quantization**
* **Results**

**Video 4 (22 min): "Positional Encoding Impact"**

* **RoPE**
* **ALiBi**
* **YaRN**
* **Extrapolation tests**

**Video 5 (18 min): "Ablation Studies"**

* **Segment length**
* **Memory size**
* **Retrieval k**
* **Cache policy params**

**Video 6 (20 min): "Cross-Lingual Analysis"**

* **Hindi performance**
* **Tamil performance**
* **Code-mixed**
* **Translation quality**

**Video 7 (28 min): "Results Analysis"**

* **Recall@distance**
* **Temporal accuracy**
* **Quality vs memory**
* **Best configurations**

**Shorts (14 clips):**

1. **"Baseline Results"**
2. **"Llama 3.1 Performance"**
3. **"PKM Results"**
4. **"kNN-LM Results"**
5. **"SnapKV Performance"**
6. **"StreamingLLM"**
7. **"Recall@Distance"**
8. **"Temporal Accuracy"**
9. **"Cross-Lingual"**
10. **"Best Memory System"**
11. **"Ablation Insights"**
12. **"Quality Trade-offs"**
13. **"Hindi Performance"**
14. **"Final Results"**

**Code: llce\_indic/experiments/**

**Blog: "LLCE-Indic: Experimental Results"**

**Week 24: Paper & Release (6 videos)**

**Video 1 (25 min): "Paper Walkthrough: Introduction"**

* **Motivation**
* **Contributions**
* **Related work**
* **Overview**

**Video 2 (28 min): "Paper Walkthrough: LLCE-Indic"**

* **Benchmark design**
* **Data collection**
* **Task taxonomy**
* **Statistics**

**Video 3 (22 min): "Paper Walkthrough: Methods"**

* **Memory systems**
* **Implementations**
* **Configurations**
* **Baselines**

**Video 4 (25 min): "Paper Walkthrough: Results"**

* **Main results**
* **Ablations**
* **Analysis**
* **Discussion**

**Video 5 (20 min): "Reproducibility Package"**

* **Code release**
* **Dataset access**
* **Documentation**
* **Docker setup**

**Video 6 (30 min): "Journey Recap & Future Work"**

* **6-month journey**
* **Key learnings**
* **Future directions**
* **Call to action**

**Shorts (12 clips):**

1. **"Paper Released!"**
2. **"LLCE-Indic Open Source"**
3. **"Key Findings"**
4. **"Best Memory System"**
5. **"Dataset Released"**
6. **"Code Available"**
7. **"Reproducibility"**
8. **"6-Month Journey"**
9. **"Key Learnings"**
10. **"Future Work"**
11. **"Try It Yourself"**
12. **"Research Impact"**

**Code: Complete GitHub release**

**Blog: "LLCE-Indic: Complete Release & Journey"**

**Twitter Thread: "6 months of research in public 🧵"**

**📊 FINAL STATISTICS**

**Total Output:**

* **150+ Deep Videos (15-28 min each)**
* **280+ Short Clips (60s each)**
* **24 Blog Posts**
* **Complete Codebase (20+ modules)**
* **Research Paper**
* **Dataset (LLCE-Indic)**
* **24 Twitter Threads**

**Coverage: ✅ Modern Transformer Architecture  
✅ Training & Optimization  
✅ Memory Systems (7 major techniques)  
✅ Efficiency & Systems  
✅ Mixture of Experts  
✅ Alignment & RLHF  
✅ Indic Research & Benchmark**

**⚡ WEEKLY SCHEDULE**

**Monday: Record 2-3 deep videos  
Tuesday: Edit videos, create shorts  
Wednesday: Record remaining videos  
Thursday: Code implementation  
Friday: Blog writing, final edits  
Saturday: Schedule all content  
Sunday: Rest, plan next week**

**Batch record shorts: Every 2 weeks, record 20-30 shorts in one session**

**🎯 THE ADITYA BHATT BRAND**

**"Modern LLM research, explained and built in public"**

* **Deep technical content**
* **Working code for everything**
* **Research frontier coverage**
* **Indic language focus**
* **Production systems perspective**

**You're not copying anyone. This is YOUR research journey documented.**