Researcher-llama-3.2-3B for Video Anomaly Detection Research

By: Thomas Foltz

This project aimed to fine-tune a large language model, creating Researcher-llama-3.2-3B, and evaluate its use as a research approach generator for video anomaly detection research. The following sections will discuss how the dataset was curated, the llm and the fine-tuning process, the evaluation of Researcher-llama-3.2-3B's effectiveness, and the reflection on the lessons learned and the challenges encountered.

Dataset construction

Dataset Generation via Perplexity LLM

The dataset was generated using Perplexity's pAI-7B-2025Q2 model through carefully engineered prompts to produce novel video anomaly detection (VAD) research questions and technical approaches. A structured prompt guided the LLM to create CSV-formatted pairs that combined practical scenarios (e.g., "How to detect AI-assisted cheating in online exams?") with technically detailed solutions. Prompts enforced diversity across many domains like healthcare, retail, and industrial IoT while requiring specific technical components such as modern architectures (Swin Transformers, 3D CNNs), sensor fusion techniques, and deployment constraints. The LLM produced 128 examples, each pairing a VAD challenge with an actionable solution spanning computer vision, edge computing, and domain-specific physics.

Data Preprocessing for Model Training

The raw CSV output underwent formatting for optimal fine-tuning. A Python script transformed each row into instruction-response pairs by prefixing questions with "Question: " and approaches with "Approach: ". This explicit task framing conditioned models to recognize the QA structure while maintaining technical precision. By keeping training (93 examples), validation (16), and test (16) splits separate during transformation, the pipeline prevented data leakage. The final format aligned with standard seq2seq training protocols, enabling models to learn mappings between problem statements and technical solutions while accommodating variable input lengths through token-based padding strategies.

LLM selection and training details

Model Selection

The foundation model selection and training methodology prioritized computational efficiency while maintaining domain-specific relevance. The Researcher-llama-3.2-3B architecture builds on Meta's Llama-3.2-3B base model, chosen for its balanced 3.2-billion parameter count and demonstrated performance in technical reasoning tasks. To enable training on consumer-grade hardware, 4-bit quantization was employed via Hugging Face's BitsAndBytes library, reducing VRAM consumption to just ~4GB during both training and inference. This optimization allowed full fine-tuning workflows to run on an NVIDIA RTX 4070 Ti Super (16GB VRAM), democratizing access to LLM specialization for video anomaly detection research.

Training Details

The adaptation strategy combined Low-Rank Adaptation (LoRA) with quantized training, targeting 7 critical layers: query/key/value projections and feed-forward networks (gate_proj, down_proj, up_proj). With LoRA rank=8 and alpha=16, only 25.4% of parameters remained trainable while keeping 74.6% frozen. The 4-bit loading configuration used NF4 quantization with double quantization for weight storage efficiency, paired with float16 compute dtype to preserve numerical stability during gradient updates.

The 50-epoch training regimen employed:

- Paged AdamW 8-bit optimizer for memory-efficient weight updates
- 32-step gradient accumulation to simulate effective batch_size=32
- 5e-5 learning rate with cosine decay to prevent catastrophic forgetting
- Per-device batch_size=1 to fit within VRAM constraints

Despite the small 93-example training set, this configuration decreased validation loss from approximately 5 to 2.4 without overfitting. The explicit "Question: ... Approach: ..." formatting established clear task boundaries during preprocessing, while architectural constraints in the training data (Swin Transformers, 3D CNNs) steered the model toward computer vision-centric solutions. The entire pipeline demonstrates how 4-bit quantization combined with parameter-efficient fine-tuning enables LLM specialization on consumer GPUs while maintaining technical rigor.

Evaluation metrics and experiments

Experiment 1: Semantic Similarity & Context Relevance

To quantify response quality, we computed semantic similarity (alignment with ground-truth approaches) and context relevance (adherence to the original question) using all-mpnet-base-v2 sentence embeddings. The fine-tuned model achieved a 74.8% improvement in semantic similarity over the base Llama-3.2-3B (tuned: **0.445** vs. original: 0.255). Context relevance remained stable (tuned: 0.383 vs. original: 0.390), confirming specialization without overfitting to spurious patterns.

Experiment 2: ROUGE-L Precision

ROUGE-L scores measured the overlap of technical terminology between generated and reference approaches. The tuned model doubled performance in this metric (**0.073** vs. 0.041), reflecting its improved ability to reproduce domain-specific phrases like "spatio-temporal graph convolutional networks" and "multi-modal fusion transformers."

Experiment 3: LLM Subjective Evaluation

To get a better idea of how the fine-tuned model performs at generating reasonable approaches for novel research challenges, it was necessary to get a more subjective evaluation. Therefore, an assessment was employed using Perplexity's pAI-7B-2025Q2 model, which scored responses on four criteria (weighted total = 100) to the ground truth:

Comparison to ground truth:

Criteria	Weight	Original LLM	Tuned LLM
Technical Accuracy	40%	12.2/40	**26.8/40**
Completeness	30%	9.4/30	**15.3/30**
Relevance	20%	7.5/20	**9.2/20**
Practicality	10%	**6.5/10**	3.4/10

Comparison to model's approach

It was also interesting to see the scores improve when allowing the pAI-7B-2025Q2 model to generate a score based on its information retrieval, as seen in the table below.

Criteria	Weight	Original LLM	Tuned LLM
Technical Accuracy	40%	15.0/40	**28.0/40**
Completeness	30%	12.0/30	**18.0/30**
Relevance	20%	10.0/20	**16.0/20**
Practicality	10%	5.0/10	**8.0/10**

Key Observations

1. Technical Accuracy
    - Original LLM: Frequently provided generic or unrelated answers (e.g., multiple-choice options for technical questions).
    - Tuned LLM: Demonstrated improved technical alignment (e.g., mentioning federated learning for multi-mall systems) but lacked specificity (e.g., omitted YOLOv7 for PPE detection).
2. Completeness
    - Original LLM: Failed to address multi-step solutions (e.g., answered "fuzzy logic" for virtual trespassing detection without details).
    - Tuned LLM: Covered broader concepts (e.g., synthetic data for rare shoplifting tactics) but missed critical components like Schrödinger bridge diffusion.
3. Relevance
    - Original LLM: 65% of responses were irrelevant (e.g., discussing geophysical surveys for construction alerts).
    - Tuned LLM: 80% relevance, with gaps in aligning with ground truth methods (e.g., omitted causal transformers for temporal lag).
4. Practicality

- Original LLM: Proposed impractical solutions (e.g., manual NDVI analysis for satellite monitoring).
- Tuned LLM: Addressed edge deployment and real-time processing but overlooked privacy-preserving techniques like homomorphic encryption.

The evaluation demonstrates that Researcher-llama-3.2-3B generates better approaches than its base model while maintaining inference efficiency (~4GB VRAM).

Lessons learned and challenges

Lessons

This project provided valuable insights into efficient LLM fine-tuning and the complexities of specialized model development. Through implementing Low-Rank Adaptation (LoRA), I learned how decomposing weight matrices into smaller trainable components enables parameter-efficient tuning while preserving base knowledge. Combined with 4-bit quantization using NF4 encoding and quantization, this approach reduced VRAM requirements to ~4GB, demonstrating how modern techniques democratize LLM specialization on consumer hardware. However, balancing LoRA's rank and alpha parameters required careful experimentation to maintain technical accuracy while avoiding overfitting to the small 93-example dataset.

Challenges

Several key challenges emerged during the process. First, dataset curation proved unexpectedly demanding - despite using advanced prompt engineering with Perplexity's model, the generated 128-example dataset still contained completeness gaps that limited solution practicality scores. Second, achieving effective learning required meticulous hyperparameter tuning: the 5e-5 learning rate with cosine decay and 32-step gradient accumulation prevented catastrophic forgetting, but the model's tendency to produce longer, more computationally intensive solutions revealed inherent tensions between technical completeness and deployment efficiency. Finally, evaluation complexities became apparent - while automated metrics (semantic similarity, ROUGE-L) showed clear improvements, subjective LLM assessments highlighted persistent issues in practical applicability that quantitative metrics alone couldn't capture. This underscores the need for hybrid evaluation frameworks combining automated scoring with domain-expert validation when fine-tuning LLMs for technical research applications.