Deciphering the Past

Presented by [#G3], March 18, 2026

The Epigraphic Bottleneck:

Why Ancient Inscriptions Remain Untranslated?

- The Manual Constraint:
  - Translating ancient inscriptions currently requires rare domain expertise and weeks of painstaking manual labour per artifact.
- Visual Noise & Degradation:
  - Physical media suffers from millennia of weathering, surface erosion, and inconsistent 3D carving depths, causing standard OCR models to fail.
- Linguistic Complexity:
  - Extinct Tamil scripts (Tamili and Vatteluttu) were written as continuous strings without modern word spacing, utilizing complex morphological Puṇarcchi rules.
- The Data Gap:
  - There is a severe lack of massive, annotated, and digitized datasets for extinct South Indian typography to train supervised models.

சிகை கொற்றன் வந்த கண்டன்

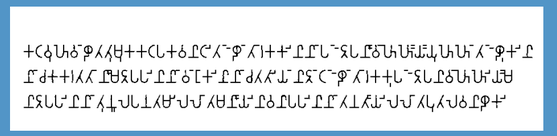
Sigai Kottran came and inspected

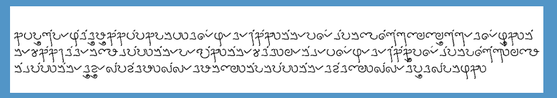
Data Engineering:

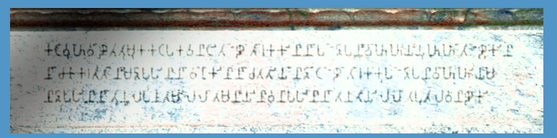
Bridging the Epigraphic Data Gap via Synthetic Corpus Generation

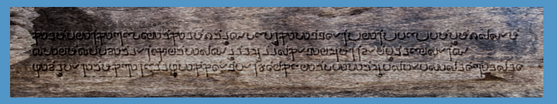
- **The Data Scarcity Problem:** No large-scale, annotated visual datasets exist for extinct South Indian scripts, rendering traditional supervised learning impossible.
- **Typographical Pipeline:** We engineered a programmatic pipeline to convert continuous strings of classical Tamil literature into clean, high-contrast Vatteluttu and Tamili glyph images and text (image-text pairs) using aksharamukha python package.
- **Physical Degradation Simulation:** To bridge the visual gap between digital text and physical artifacts, we utilized the “augraphy” library to mathematically blend the glyphs onto diverse stone textures.

கடவுள்வாழ்த்துமுக்கட்பகவனடிதொழாதார்க்
கின்னாபொற்பனைவெள்ளையுள்ளாழ
கின்னாசக்கரத்தானைமறப்பின்னாவாங்கின்
னாசத்தியான்றாடொழாதார்க்குற்பனவெள்
ளியைமன்றப்பின்னாதுநூல்பந்தமில்லாதம
னையின்வனப்பின்னாதந்தையில்லாதபுதல்
வனழகி









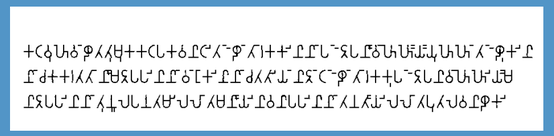
Data Engineering:

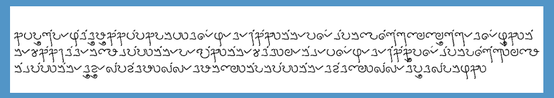
Bridging the Epigraphic Data Gap via Synthetic Corpus Generation

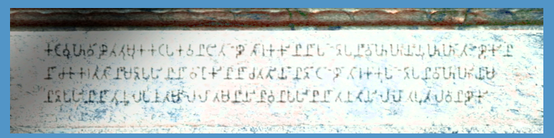
- **Augmentation Hyperparameters:** Applied specific algorithms to mimic millennia of weathering to 1000 test images and trained model with 5,000 black&white images:
  - Surface Noise: Simulating pitting and erosion on the rock face.
  - Lighting Variance: Replicating uneven shadows caused by 3D chisel depths.
  - Carving Bleed: Altering stroke thickness to mimic physical carving anomalies.

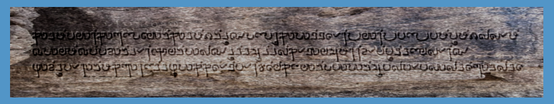
- **Hardware-Constrained Stratification:** Generated 28,000 synthetic text-image pairs, strategically stratified and down-sampled to 6000 pairs to strictly adhere to the 20GB VRAM cluster limits without sacrificing morphological variance.

கடவுள்வாழ்த்துமுக்கட்பகவனடிதொழாதார்க்
கின்னாபொற்பனைவெள்ளையுள்ளாழ
கின்னாசக்கரத்தானைமறப்பின்னாவாங்கின்
னாசத்தியான்றாடொழாதார்க்குற்பனவெள்
ளியைமன்றப்பின்னாதுநூல்பந்தமில்லாதம
னையின்வனப்பின்னாதந்தையில்லாதபுதல்
வனழகி





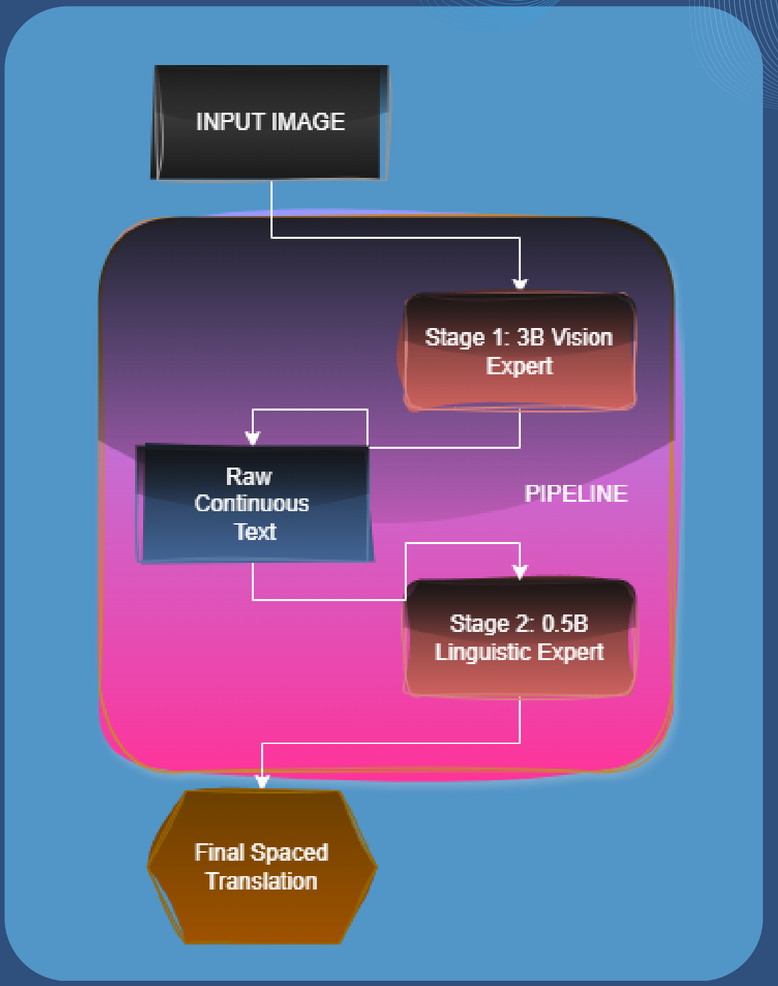




Methodology:

A Cascaded Multi-Modal Architecture

- ***The Monolithic Limitation:*** Forcing a single Vision-Language Model to simultaneously perform zero-shot OCR on degraded stone and parse complex morphological Punarcchi rules leads to high VRAM overhead and suboptimal convergence.
- **The Decoupled Solution:** We engineered a two-stage pipeline, encapsulated by an orchestrator script, dividing the problem into two highly specialized domain experts.
- **Stage 1 (The Vision Expert):** A 3-billion parameter VLM (Qwen2.5-VL) fine-tuned via 4-bit QLoRA to extract continuous, unspaced character strings directly from the severe visual noise of the stone.
- **Stage 2 (The Linguistic Expert):** A highly efficient, lightweight 0.5-billion parameter causal language model dedicated purely to resolving word boundaries and inserting modern spacing.
- **Computational Efficiency:** By isolating the text-segmentation task, we drastically reduce inference costs and bypass the need to load the heavy 3B vision model for purely textual tasks.

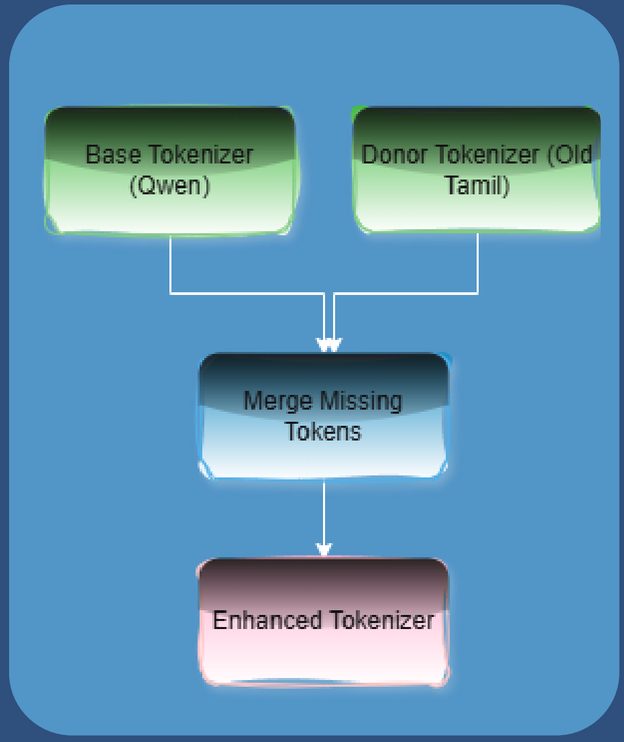


Stage 1 – Vision Expert

Configuration & Hardware Engineering

Model & Vocabulary Injection :

- **Base Architecture**: Qwen2.5-VL-3B-Instruct
- **The Tokenizer Problem**: The base model lacked specific ancient Tamil characters.
- **The Fix**: Programmatically extracted custom Tamil tokens from a custom tokeniser, injected them into the 3B model, and dynamically resized the embedding matrix.



VRAM & Cluster Survival Hacks :

- **Memory Fragmentation:** Bypassed CUDA OOM crashes by setting PYTORCH_ALLOC_CONF = "expandable_segments:True".
- **Resolution Bounding:** Capped the processor's dynamic resolution to a maximum of 384 * 28 * 28 pixels to strictly control sequence-length memory spikes.
- **Checkpoint Resilience:** Engineered a custom resume script to automatically recover from university VPN drops and cluster timeouts without losing gradient progress.

Stage 1 – Vision Expert ...

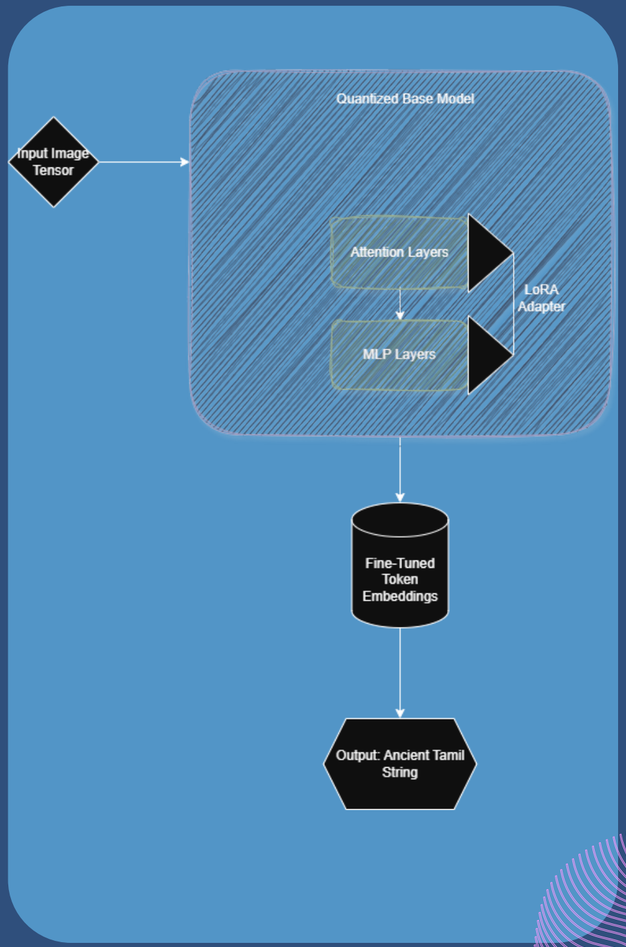
Configuration & Hardware Engineering

Quantization & LoRA Strategy:

- **Quantization:** 4-bit NormalFloat (NF4) with Double Quantization (bfloat16 compute dtype).
- **PEFT Configuration:** Rank 16, Alpha 32
- **Target Modules:** Targeted attention heads (q_proj, v_proj, etc.) and MLPs (gate_proj, up_proj).
- **Critical Save:** Explicitly saved embed_tokens and lm_head in the LoRA config to ensure the newly injected Tamil vocabulary was preserved.

Data Pipeline & Augmentation:

- **Stratified Split:** 5,000 Train / 1,000 Val, strictly stratified by script_type (Tamili vs. Vatteluttu) to prevent class imbalance.
- **On-the-Fly Augmentation:** 20% random probability of ImageOps.invert() during batching to simulate different rock shadings and ink rubbings.
- **Effective Batching:** Batch Size 1 with 16 Gradient Accumulation steps, optimizing via adamw_8bit.



Stage 2 – The "Grammarian"

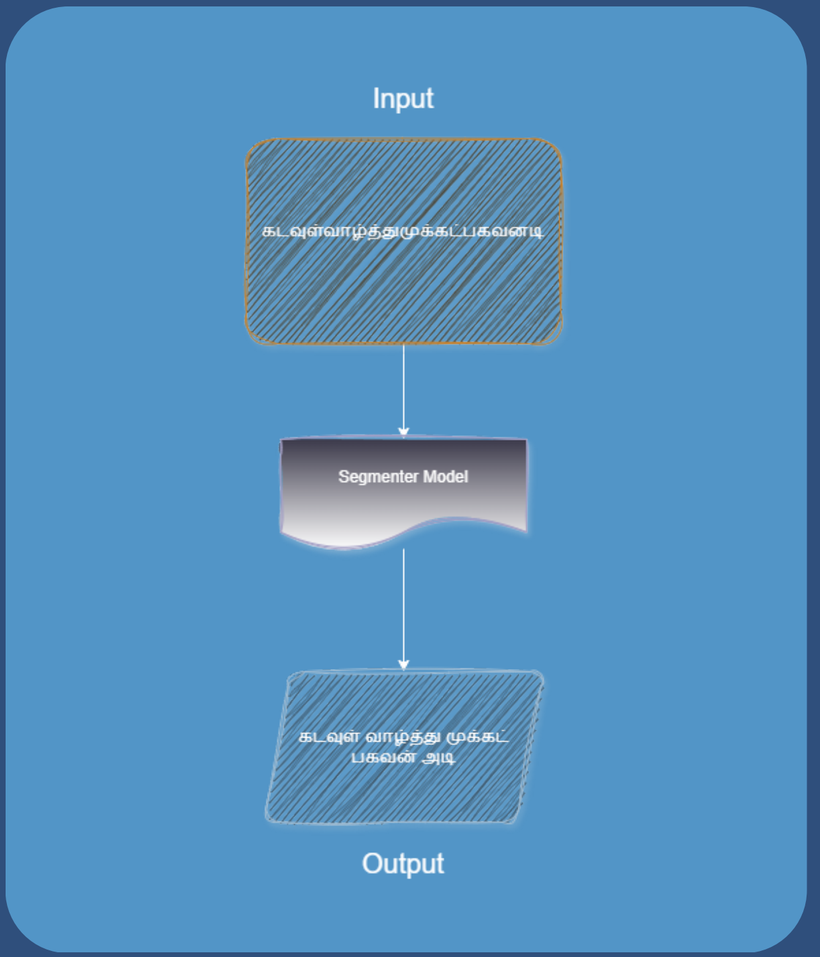
Overcoming Ancient Tamil PunarchiRules

The Linguistic Hurdle:

- **The Problem:** Ancient Tamil operates on complex morphophonological fusion rules (Sandhi). Words do not merely sit next to each other; they mutate and merge when connected, eliminating standard word boundaries.
- **The Requirement:** A standard dictionary lookup fails here. The system requires contextual, grammatical reasoning to untangle the fused characters into modern spaced syntax.

Model Selection: The Ultra-Lightweight Approach:

- **Base Architecture:** Qwen2.5-0.5B.
- **The Rationale:** Text segmentation is a highly localized syntactic task. It does not require the vast "world knowledge" of a 7B or 72B model. By deploying a 0.5B model, we drastically reduced inference latency and memory overhead, making the final API highly responsive.

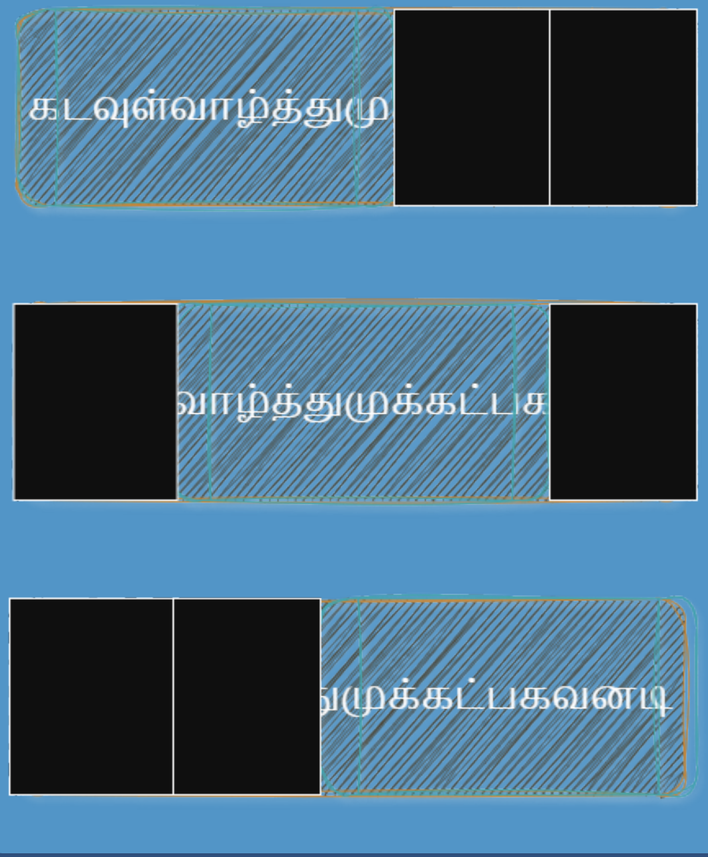


Stage 2 – The "Grammarian"...

Overcoming Ancient Tamil PunarchiRules

Data Strategy: The Sliding Window Algorithm:

- An LLM's attention mechanism dilutes over massive text blocks. To force hyper-efficient learning, the data had to be surgically engineered.
- **The Algorithm:** Implemented a sliding window chunking script across large epigraphic corpora.
- **The Output:** Generated 25,611 bite-sized conversational pairs.
- **The Constraint:** Strictly limited inputs to 15-word micro-prompts. This artificially restricted the 0.5B model's attention span, forcing it to concentrate entirely on local word boundaries rather than hallucinating overarching semantic narratives.



Results Comparison

Analyzing the output pipeline effectiveness

The pipeline demonstrates significant progress, transitioning from raw continuous Tamil strings to accurately segmented text, showcasing the efficiency of our two-stage AI architecture.

Evaluation Target	BLEU Score	ROUGE-1 (Unigram)	ROUGE-L (Sequence)
Stage 1: Clean Script OCR (Baseline)	95.40	0.9881	0.9838
Stage 1: Noisy Stone OCR (Stress Test)	63.02	0.8796	0.8032
Stage 2: Morphological Segmentation	72.02	0.8471	0.8469

Engineering Resilience

Overcoming Obstacles in Development

The Hardware Bottleneck: CUDA Out-Of-Memory (OOM)

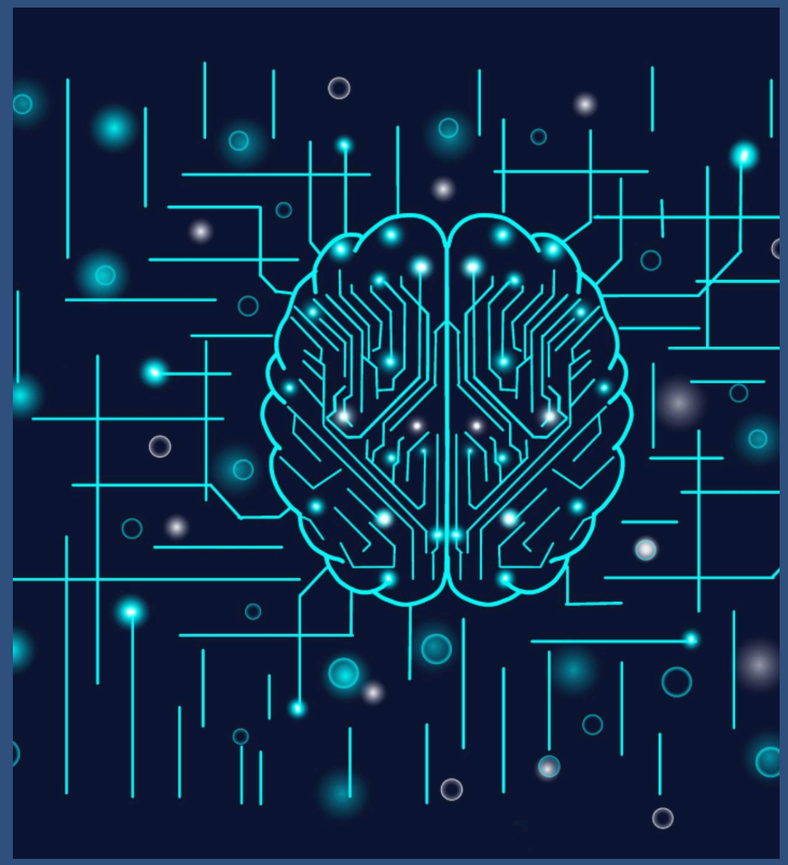
- **The Problem:** High-resolution stone chunks caused massive sequence-length spikes, instantly crashing the university cluster's strict 20GB VRAM limit.
- **The Engineering Fix:** Enabled expandable_segments:True to prevent PyTorch memory fragmentation.
  - Offloaded state memory to CPU via the paged_adamw_8bit optimizer.
  - Capped dynamic resolution and applied 4-bit NF4 double quantization.
- **The Result:** Squeezed a full 3B Vision-Language training loop into an 18.5GB footprint.

Engineering Resilience...

Overcoming Obstacles in Development

The Cognitive Risk: Catastrophic Forgetting

- **The Problem:** The base Qwen model had no concept of ancient Tamil. However, forcing it to learn a new alphabet risked overwriting its pre-trained spatial reasoning (making it "forget" how to see shadows and shapes).
- **The Engineering Fix:** Executed a surgical "Vocab Stealing" script to safely inject missing tokens.
  - Strictly gated the gradients using LoRA: Froze the core MLP layers while explicitly unfreezing only the embed_tokens and attention projection layers.
- **The Result:** The model successfully learned a dead script while retaining 100% of its foundational visual intelligence.



Future Horizons

The Field Prototype

Dataset Expansion: From Synthetic to Real-World

- **The Current State:** The models were trained and validated on a highly engineered synthetic dataset of 6000/28,000 images.
- **The Next Phase:** Partnering with epigraphy departments to ingest real, historically verified photographs of Vatteluttu and Tamili inscriptions.
- **The Method:** Using the current 3B Vision model as a baseline to pseudo-label unread real-world inscriptions, human experts will verify the outputs, creating a high-quality feedback loop for continuous fine-tuning.

The V1 Prototype & Next-Generation Scaling

- The Proof of Concept: The current 3B / 0.5B decoupled architecture successfully validates the two-stage methodology under strict academic hardware constraints (20GB VRAM).
- Scaling the "Engine": With the architectural blueprint proven, future R&D will swap the lightweight models for heavier, state-of-the-art foundation models.
- Next-Gen Integration: Upgrading Stage 1 to a 7B+ parameter Vision-Language Model or utilizing upcoming architectures (e.g., Qwen3.5). These larger models will close the BLEU score gap on highly degraded, real-world stone textures.

Key Takeaways

Architectural Decoupling Succeeds:

- Replaced monolithic, error-prone OCR attempts with a specialized Two-Stage Agent. Isolating visual extraction (3B) from linguistic grammar (0.5B) drastically reduced hallucinations and improved overall accuracy.

High Performance Under Hard Constraints

- Proved that state-of-the-art Vision-Language fine-tuning can be achieved on strict academic hardware (20GB VRAM) through aggressive 4-bit quantization, matrix resizing, and optimizer offloading.

Solving the Continuous Script Bottleneck

- Successfully reconstructed complex Sandhi (புணர்ச்சி) word boundaries using an ultra-lightweight 0.5B model, achieving a 72.02 BLEU score via an engineered sliding-window dataset.