# **Comparative Analysis of LLM Reasoning vs. Nsight Compute Ground Truth for GPU Kernel Bottlenecks**

**Date:** October 25, 2025

## **Executive Summary**

This study presents a comparative evaluation between GPU kernel performance bottleneck predictions made by the **Gemini 2.0 Flash** large language model (LLM) and empirical ground truth derived from NVIDIA’s **Nsight Compute** profiling tool.

An **11-kernel CUDA benchmark suite** was profiled using ncu --set full, targeting diverse bottleneck categories (Compute-, Memory-, and Latency-Bound). Ground truth classifications were determined by analyzing *Streaming Multiprocessor (SM) Throughput* and *DRAM Throughput* percentages. In contrast, the LLM inferred bottlenecks purely from CUDA source code reasoning.

The comparison revealed an **overall accuracy of 64%** for the LLM’s predictions relative to Nsight Compute results. Gemini 2.0 Flash demonstrated **perfect accuracy (100%)** in detecting *Latency-Bound* kernels, moderate success for *Memory-Bound* (57% precision, 80% recall), but significant difficulty in distinguishing *Compute-Bound* and *Mixed-Bottleneck* cases.

Notable misclassifications included naive\_matmul (incorrectly labeled Memory-Bound) and high\_reg\_pressure (incorrectly labeled Compute-Bound). The findings suggest that while Gemini 2.0 Flash exhibits a foundational understanding of GPU performance patterns, it lacks the nuanced quantitative reasoning required for precise classification of complex kernels.

## **1. Introduction**

Efficient GPU kernel optimization relies on identifying the primary performance bottleneck—whether computation, memory bandwidth, or latency is the limiting factor.

* **Compute-Bound kernels** saturate the GPU’s arithmetic pipelines (high ALU usage, low memory activity).
* **Memory-Bound kernels** are constrained by limited DRAM bandwidth or memory transaction latency.
* **Latency-Bound kernels** experience stalls due to synchronization, data dependencies, or control flow divergence.

**NVIDIA Nsight Compute (ncu)** provides fine-grained performance counters that reveal how well a kernel utilizes available compute and memory resources. However, interpreting these results requires domain expertise.

This project explores whether a **Large Language Model (LLM)**—specifically **Gemini 2.0 Flash**—can replicate expert reasoning to predict bottlenecks directly from CUDA source code. Its predictions are then quantitatively compared against Nsight Compute ground truth classifications derived from measured throughput data.

## **2. Methodology**

### **2.1 Kernel Test Suite**

An 11-kernel CUDA suite was developed, each kernel designed to exhibit specific bottleneck characteristics:

| Category | Kernels |
| --- | --- |
| **Memory-Bound** | saxpy, branch\_divergence, high\_reg\_pressure, parallel\_reduction |
| **Compute-Bound** | naive\_matmul, tiled\_matmul |
| **Latency-Bound** | naive\_transpose, bank\_conflict, strided\_global, atomic\_histogram |
| **Mixed/Optimized** | tiled\_transpose |

This diversity ensures coverage of typical GPU execution bottlenecks, including compute-intensive arithmetic kernels, memory bandwidth-limited operations, and latency-dominated synchronization-heavy tasks.

### **2.2 Ground Truth Generation (Nsight Compute Profiling)**

1. **Profiling Execution:** Each kernel was compiled within a C++ harness and executed with problem size ( N = 2048 ). Nsight Compute collected performance data using:

* ncu --set full -o report\_full\_with\_tiled.ncu-rep .\harness.exe

1. **Data Export:** The binary .ncu-rep file was converted to CSV:

* ncu --import report\_full\_with\_tiled.ncu-rep --csv --page details > ncu\_report\_details.csv

1. **Metric Extraction and Labeling:** Two key metrics were extracted:
   * **SM Throughput (%)** – fraction of compute unit utilization.
   * **DRAM Throughput (%)** – fraction of memory subsystem utilization.

* Classification rules were applied as follows:

| Category | SM Throughput | DRAM Throughput |
| --- | --- | --- |
| **COMPUTE-BOUND** | > 60% | < 30% |
| **MEMORY-BOUND** | < 30% | > 60% |
| **LATENCY-BOUND** | < 40% | < 40% |
| **MIXED (Compute/Memory)** | > 50% | > 50% |
| **Likely COMPUTE-BOUND** | SM > DRAM (intermediate) |  |
| **Likely MEMORY-BOUND** | DRAM > SM (intermediate) |  |
| **MIXED/OTHER** | Both between 40–50% |  |

1. **Output:** The derived results were saved to derived\_ground\_truth.csv, containing per-kernel SM/DRAM throughput and the final classification label.

### **2.3 LLM Prediction Generation**

1. **Model Used:** **Gemini 2.0 Flash** (Google’s latest LLM as of 2025) was accessed via a Python script (predict\_llms.py).
2. **Prompt Context:** The model was informed of the GPU’s peak theoretical performance (9.0 TFLOPs FP32, 192 GB/s bandwidth) and given each kernel’s full source code.

* It was then asked to:
  + Identify the dominant bottleneck.
  + Provide reasoning (JSON output).

1. **Output:** Predictions were saved to llm\_predictions.csv, including rationale text and error flags (API\_ERROR, ERROR) if any.

### **2.4 Comparison and Evaluation**

The processed ground truth and LLM predictions were merged and analyzed using Python’s scikit-learn for classification metrics and confusion matrix visualization. The **performance profile plot** (performance\_profile\_comparison.png) mapped each kernel’s SM vs DRAM throughput, color-coded by ground truth and shaped by LLM prediction for direct visual comparison.

## **3. Results**

### **3.1 Overall Accuracy**

Gemini 2.0 Flash achieved **64% accuracy**, correctly classifying 7 out of 11 kernels.

### **3.2 Classification Metrics**

precision recall f1-score support  
  
 COMPUTE-BOUND 0.50 1.00 0.67 1  
 LATENCY-BOUND 1.00 1.00 1.00 2  
 Likely COMPUTE-BOUND 0.00 0.00 0.00 1  
 MEMORY-BOUND 0.57 0.80 0.67 5  
 MIXED(Compute/Memory) 0.00 0.00 0.00 1  
 MIXED/OTHER 0.00 0.00 0.00 1  
  
 accuracy 0.64 11  
 macro avg 0.35 0.47 0.39 11  
 weighted avg 0.49 0.64 0.55 11

**Observations:**

* *Latency-Bound:* Perfect precision and recall (2/2 correctly identified).
* *Memory-Bound:* Moderate accuracy; several false positives.
* *Compute-Bound:* Correctly detected tiled\_matmul but missed naive\_matmul.
* *Mixed Cases:* Consistently underperformed—no correct detections.

### **3.3 Confusion Matrix**

[[1 0 0 0 0 0] # True COMPUTE-BOUND  
 [0 2 0 0 0 0] # True LATENCY-BOUND  
 [0 0 0 1 0 0] # True Likely COMPUTE-BOUND -> Predicted MEMORY  
 [1 0 0 4 0 0] # True MEMORY-BOUND -> 4 Correct, 1 Predicted COMPUTE  
 [0 0 0 1 0 0] # True MIXED(Compute/Memory) -> Predicted MEMORY  
 [0 0 0 1 0 0]] # True MIXED/OTHER -> Predicted MEMORY

Labels: COMPUTE-BOUND, LATENCY-BOUND, Likely COMPUTE-BOUND, MEMORY-BOUND, MIXED(Compute/Memory), MIXED/OTHER.

### **3.4 Performance Profile Visualization**

The **scatter plot** (performance\_profile\_comparison.png) illustrates SM throughput (x-axis) vs DRAM throughput (y-axis):

* **Bottom-left (Low SM, Low DRAM):** Latency-Bound kernels (atomic\_histogram, bank\_conflict) — correctly identified.
* **Upper-left (High DRAM, Low SM):** Memory-Bound kernels (saxpy, branch\_divergence).
* **Diagonal region:** Mixed kernels (parallel\_reduction, tiled\_transpose).
* **Right region (High SM, Low DRAM):** Compute-Bound kernels (tiled\_matmul, naive\_matmul).

Marker mismatches visually indicate incorrect LLM classifications.

## **4. Mismatch Analysis**

### 1. **high\_reg\_pressure**

* **Ground Truth:** MEMORY-BOUND (SM: 12.4%, DRAM: 92.4%)
* **LLM Prediction:** COMPUTE-BOUND
* **Explanation:** The LLM overemphasized arithmetic instruction count while ignoring the effect of register spilling and occupancy loss—factors that lead to memory bottlenecks.

### 2. **naive\_matmul**

* **Ground Truth:** Likely COMPUTE-BOUND (SM: 98.8%, DRAM: 37.0%)
* **LLM Prediction:** MEMORY-BOUND
* **Explanation:** The model attempted to compute *operational intensity (OI)* and erroneously found a low value (0.25 FLOPs/Byte). It failed to consider data reuse in matrix multiplications, misidentifying this classic compute-intensive kernel as memory-limited.

### 3. **parallel\_reduction**

* **Ground Truth:** MIXED(Compute/Memory) (SM: 73.6%, DRAM: 69.3%)
* **LLM Prediction:** MEMORY-BOUND
* **Explanation:** Although Gemini noted both compute and memory phases, it ultimately classified the kernel as memory-limited. The reasoning disregarded the significant compute workload during reduction stages.

### 4. **tiled\_transpose**

* **Ground Truth:** MIXED/OTHER (SM: 71.1%, DRAM: 49.8%)
* **LLM Prediction:** MEMORY-BOUND
* **Explanation:** Nsight shows moderate DRAM and high SM throughput, but the model simplified the interpretation to a purely memory-bound case, neglecting shared memory optimization effects.

## **5. Discussion**

### **5.1 Strengths**

* **Latency Detection:** Gemini 2.0 Flash showed perfect precision and recall for latency-heavy kernels, correctly associating synchronization and atomics with latency-bound performance.
* **Contextual Awareness:** The model’s explanations included advanced GPU terminology (e.g., “shared memory bank conflicts,” “global memory transactions”), indicating a conceptual grasp of GPU execution.

### **5.2 Weaknesses**

1. **Mixed Bottleneck Blind Spot:** Kernels with balanced compute and memory activity (e.g., parallel\_reduction) were consistently oversimplified as memory-bound.
2. **Operational Intensity Misuse:** The LLM misapplied OI calculations, leading to wrong conclusions for kernels with high data reuse.
3. **Neglect of Hardware Constraints:** High register usage or warp divergence—key performance limiters—were often overlooked.
4. **Qualitative Over Quantitative Reasoning:** The LLM’s reasoning emphasized textual code features over numerical inference from provided GPU specifications.

### **5.3 Interpretation**

The results highlight an essential trade-off: **Gemini 2.0 Flash excels at semantic reasoning** but lacks the numerical precision required for quantitative performance analysis. Its predictions align well for kernels with clear bottlenecks (either compute- or latency-dominated) but falter when complex hardware interactions emerge.

## **6. Conclusion**

The comparative study between **Gemini 2.0 Flash** predictions and **Nsight Compute** ground truth across 11 CUDA kernels demonstrates the potential and limitations of LLM-based GPU performance reasoning.

Key takeaways:

* **Overall Accuracy:** 64%
* **Perfect Accuracy:** Latency-Bound kernels
* **Common Error:** Misclassification of mixed and compute-bound workloads as memory-bound

While Gemini 2.0 Flash provides insightful first-pass analysis, it cannot yet replace precise, metric-driven tools such as Nsight Compute. The model’s reasoning lacks quantitative rigor and misinterprets operational intensity in cases of significant data reuse or hardware contention.

Future work should involve:

* Fine-tuning LLMs with labeled kernel datasets.
* Incorporating explicit metric prompts (SM %, DRAM %, occupancy).
* Developing hybrid AI-profiling systems combining LLM reasoning with counter-based inference.

## **Appendix: Supporting Data and Files**

| File | Description |
| --- | --- |
| ncu\_report\_details.csv | Raw Nsight Compute export |
| derived\_ground\_truth.csv | Processed metrics with bottleneck labels |
| llm\_predictions.csv | Gemini 2.0 Flash predictions with rationale |
| mismatched\_predictions.csv | Kernels where LLM and ground truth differ |
| confusion\_matrix.png | Visual representation of classification accuracy |
| performance\_profile\_comparison.png | SM vs DRAM throughput scatter plot |