Profiling Computational Intensity of Deep Neural Network Training on CUDA

Austin Pham, *Columbia University*, April 2024

***Abstract*—In this project, we investigate the performance characteristics of three popular deep learning models: ResNet50, ViT (Vision Transformer), and MobileNet trained on three distinct NVIDIA GPU architectures: Tesla Turing 4 (T4), NVIDIA Ada Lovelace 4 (L4), and NVIDIA Tesla Volta 100 (V100). Our objective is to analyze and compare the performance of these models across different GPU architectures, focusing on metrics such as training time, memory bandwidth, and computational efficiency. The experimental methodology involves training the deep learning models on each GPU architecture while collecting performance data using hardware counters and profiling tools. The study aims to identify the strengths and limitations of each GPU architecture for various deep learning workloads. Our findings can guide practitioners in selecting the most suitable GPU for their specific deep learning workloads, considering factors such as performance, cost, and scalability.**

I. Introduction

Deep learning has revolutionized various domains, from computer vision and natural language processing to speech recognition and reinforcement learning [1]. As deep learning models continue to grow in complexity and size, the demand for efficient hardware architectures to train and deploy these models has become increasingly important. Graphics Processing Units (GPUs) have emerged as the primary hardware accelerators for deep learning workloads, offering massive parallelism and high computational throughput.

However, not all GPU architectures are created equal, and their performance can vary significantly depending on the deep learning model and workload characteristics. This study aims to investigate the performance characteristics of three widely-used deep learning models: ResNet50, Vision Transformer (ViT), and MobileNet. We train these models on a subset of ImageNet, a commonly used dataset for large-scale object recognition [2].

The choice of deep learning models for this study is motivated by their diverse network topology and computational requirements. ResNet50 and ViT represent computationally intensive models that can potentially benefit from the parallelization capabilities of GPUs. In particular, the attention mechanisms in ViT involve highly parallelizable operations that could leverage the GPU's resources effectively. In contrast, MobileNet is a lightweight model designed for efficient inference on mobile devices, and its performance on GPUs may not be as pronounced.

The chosen GPU architectures span a range of computational capabilities and memory configurations. The Tesla T4, based on the Turing architecture, is a cost-effective GPU with relatively lower CUDA core count and memory bandwidth, potentially making it more suitable for lightweight models like MobileNet. In contrast, the L4, built on the newer Ada Lovelace architecture, offers higher TFLOP performance and improved memory bandwidth over the T4, which could benefit more computationally intensive workloads like ResNet50 and ViT. Lastly, the Tesla V100, a high-end GPU with significantly higher TFLOP performance and memory bandwidth, is expected to showcase its strengths for computationally intensive models like ViT and ResNet50. By evaluating these diverse GPU architectures, we aim to uncover their strengths and limitations for different deep learning models and workload characteristics.

II. Methodology

For each deep learning architecture and hardware accelerator, we collect performance data during the training process. The training is performed with a batch size of 64, and a total of 256 images are used, resulting in 4 steps per epoch. A warmup epoch is executed first to allow the GPU to reach a steady state. During the second epoch, metrics are recorded for all 4 steps but then averaged after to ensure a more consistent benchmark.

To gather performance data, the NVIDIA Nsight Compute CLI profiling tool is used [3]. The ncu command is executed to collect a set of metrics capturing the number of executed instructions for different arithmetic operations (additions, multiplications, fused multiply-adds) across different data types (float, double, half-precision), as well as memory DRAM bandwidth utilization and the overall GPU time duration.

After collecting data from the profiling run, a series of post-processing steps are performed to extract the relevant metrics from the generated csv file. The raw profiling data is processed using a Python script, which sums up the values for each metric across all kernel functions executed during the training step. The *analysis.ipynb* notebook calculates aggregate metrics such as total DRAM bandwidth utilization, floating-point operations per second (FLOPS) for different data types, and the total GPU time duration using the following formulas:

$$time\ (sec) = \frac{gpu\_time\_duration.sum}{1e9 * 4} \tag{1}$$

$$DRAM\ (GB/s) = \frac{dram\_read + dram\_write}{1e9 * 4 * time} \tag{2}$$

$$FLOPS = \frac{op\_fadd + op\_fmul + 2 * op\_ffma}{4} \tag{3}$$

$$GLOP/sec = \frac{FLOPS}{1e9 * sec} \tag{4}$$

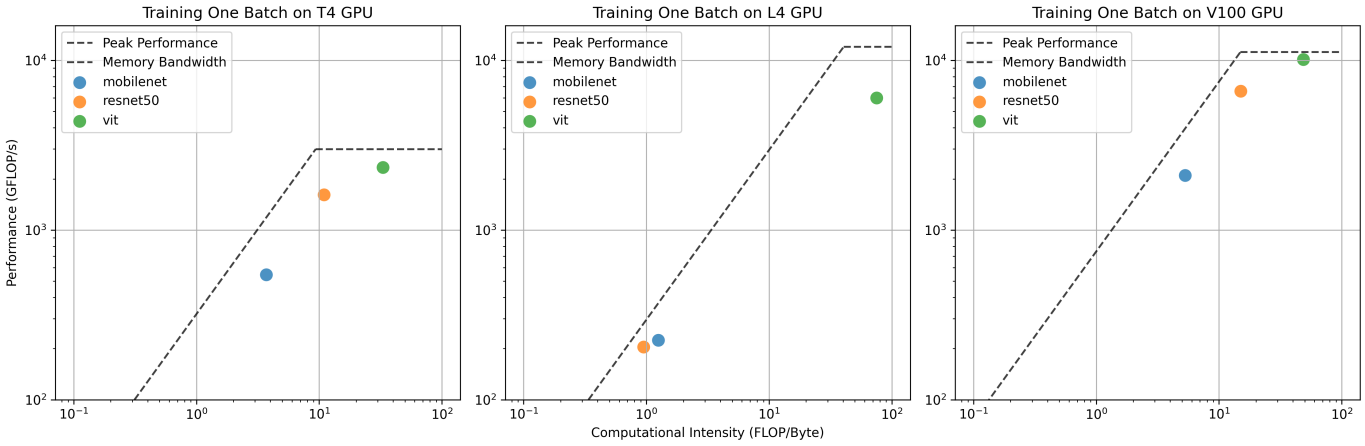

Fig. 1. Roofline models the 3 different hardware accelerators: NVIDIA Tesla T4, NVIDIA L4, and NVIDIA Tesla V100. For each, we graphed performance ($GFLOP/s$) as a function of computational intensity ($FLOP/byte$) for each of the tested models: ResNet50, MobileNet and ViT. The inflection point where hardware memory bandwidth meets peak computational performance determines whether we are memory bound (left) vs. compute bound (right).

$$Comp.\ Intensity\ (Flop/Byte) = GLOP/DRAM \quad (5)$$

III. Theoretical Calculations

To gain insights into the expected performance characteristics of the deep learning models on different GPU architectures, we can perform theoretical calculations based on the computational requirements of the models and the specifications of the GPU accelerators. These calculations can provide rough estimates of the training time for a single step, allowing us to establish a baseline for comparison with the empirical results obtained from our experiments.

A. Computational Requirements

Each model's architectural design defines their respective computational requirements. To help with our analysis we use Flop Counter for PyTorch models, a tool developed by Facebook research. The *fvcore* package provides both an operator-level and module-level flop counts, which was used in our theoretical calculations [4].

1) MobileNet: MobileNet is a lightweight convolutional neural network designed specifically for efficient inference on mobile and embedded devices. Its architecture is characterized by the use of depthwise separable convolutions, which decouple the spatial and channel-wise convolutions, resulting in a significant reduction in computational complexity [5]. According to the FLOP analysis, MobileNet requires approximately **15.31 GFLOPs** for a single forward and backward pass during training.

2) ResNet50: ResNet50 is a deeper and more complex convolutional neural network architecture, widely used for various computer vision tasks. It employs standard convolutional layers, along with residual connections that allow for effective training of deeper networks [6]. The FLOP count reveals that ResNet50 requires approximately **270 GFLOPs** for a single training step.

3) Vision Transformer (ViT): The Vision Transformer (ViT) is a transformer-based model that adapts the self-attention mechanism, originally designed for natural language processing, to handle image data. Unlike convolutional architectures, ViT divides the input image into non-overlapping patches and processes them through multi-head attention layers and feed-forward networks [7]. The FLOP analysis indicates we require around **280 GFLOPs** for a single training step, making it the most computationally intensive model among the three architectures studied.

B. GPU Accelerator Specifications

The performance of the GPU accelerators can be characterized by their peak floating-point performance (TFLOPS), memory bandwidth, and other architectural features.

Architecture	Peak TFLOPS	Memory Bandwidth
NVIDIA Tesla T4	8.1	320
NVIDIA L4	30.3	300
NVIDIA Tesla V100	15.7	900

1) NVIDIA Tesla T4: The Tesla T4, based on the Turing architecture, has a peak single-precision (FP32) performance of 8.1 TFLOPS and a memory bandwidth of 320 GB/s [8].

2) NVIDIA L4: The L4, built on the newer Ada Lovelace architecture, offers a peak FP32 performance of 30.3 TFLOPS and a memory bandwidth of 300 GB/s [8].

3) NVIDIA Tesla V100: The Tesla V100 is a high-end GPU with a peak FP32 performance of 15.7 TFLOPS and a memory bandwidth of 900 GB/s [9].

C. Training Time Estimates

Based on the computational requirements of the models and the specifications of the GPU accelerators, we can estimate the theoretical training time for a single step using the following formula:

$$t_{train} = \frac{FLOPs}{Peak\ TFLOPS} \quad (6)$$

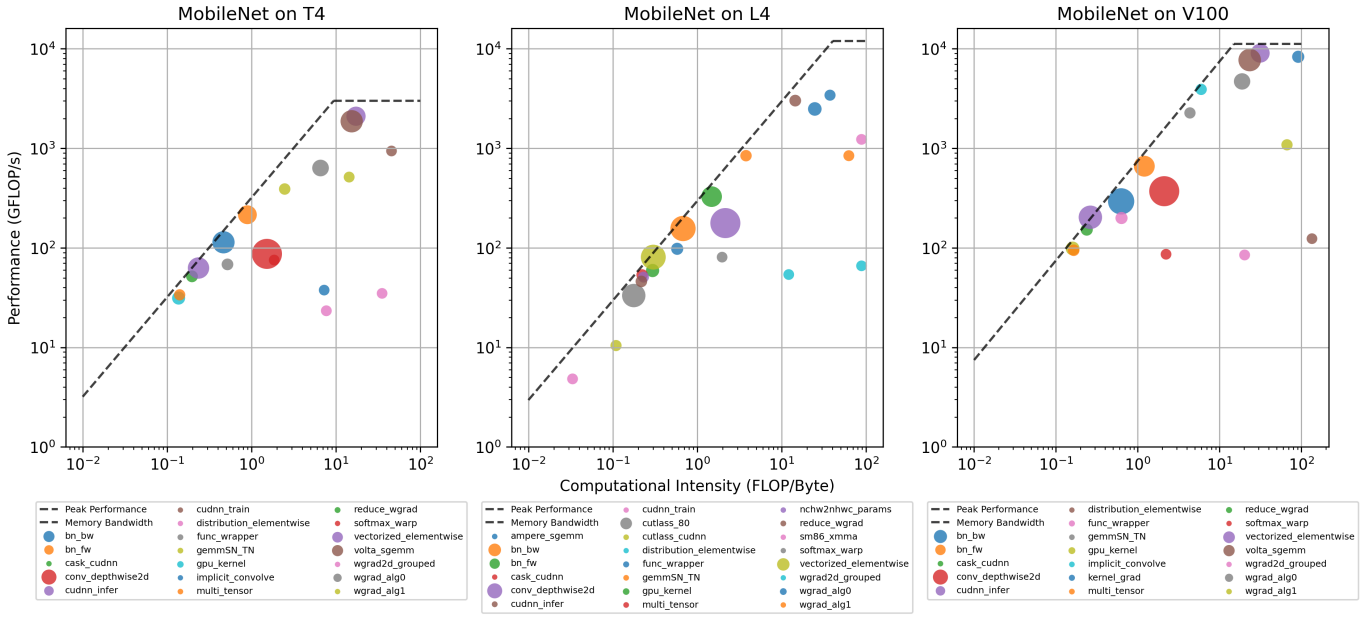

Fig. 2. We aggregate similar functioning kernels among the MobileNet architecture. The size of each point is determined by its relative GPU run time, showing the overall impact a particular cluster has on the entire run job. We see the longest running kernels are within both the **memory bound** regime

Assuming perfect scaling and no memory bandwidth limitations, the theoretical training times for a single step are as follows:

- ResNet50
  - Tesla T4: $\frac{270\times10^9}{8.1\times10^{12}} \approx 33.33$ seconds
  - L4: $\frac{270\times10^9}{30.3\times10^{12}} \approx 8.91$ seconds
  - Tesla V100: $\frac{270\times10^9}{15.7\times10^{12}} \approx 17.20$ seconds
- MobileNet
  - Tesla T4: $\frac{15.31\times10^9}{8.1\times10^{12}} \approx 1.89$ seconds
  - L4: $\frac{15.31\times10^9}{30.3\times10^{12}} \approx 0.51$ seconds
  - Tesla V100: $\frac{15.31\times10^9}{15.7\times10^{12}} \approx 0.98$ seconds
- Vision Transformer (ViT)
  - Tesla T4: $\frac{280\times10^9}{8.1\times10^{12}} \approx 34.57$ seconds
  - L4: $\frac{280\times10^9}{30.3\times10^{12}} \approx 9.24$ seconds
  - Tesla V100: $\frac{280\times10^9}{15.7\times10^{12}} \approx 17.83$ seconds

IV. Hypothesis

Due to MobileNet's lightweight design, we expect it to perform relatively well across all GPU architectures. However, it may not fully utilize the computational capabilities of more powerful GPUs. We may expect a memory bandwidth bottleneck across all accelerators, given limited computational requirements.

A deeper and more complex convolutional neural network, like ResNet50, may be expected to benefit more from the increased computational capabilities of the L4 and Tesla V100 GPUs. Theoretical calculations suggest that ResNet50 may perform better on the L4 compared to the Tesla V100, as the L4 has a higher peak TFLOPS performance, potentially resulting in shorter training times.

As for Vision Transformers, the most computationally intensive model among the three, its performance is expected to be heavily influenced by the computational capabilities of the GPUs. Based on the theoretical calculations, the L4 GPU is likely to provide the best performance for ViT due to its high peak TFLOPS and relatively high memory bandwidth.

Overall, based on the theoretical calculations and architectural considerations, the L4 GPU is likely to provide the best performance for the computationally intensive models like ResNet50 and ViT. The Tesla V100 may also perform well for these models, although its performance might be slightly lower than the L4.

V. Empirical Results

Model	GPU	DRAM	Time	GLOP/s	CI
ResNet50	T4	147.8	0.7590	1615	10.92
	L4	215.4	0.9508	204.8	0.9235
	V100	436.2	0.1867	6570	15.06
MobileNet	T4	146.7	0.2204	544.5	3.711
	L4	179.6	0.1174	2224	12.47
	V100	396.5	0.0612	2094	5.282
ViT	T4	70.47	0.7306	2336	33.15
	L4	79.67	0.2777	5992	75.21
	V100	207.6	0.1688	10113	48.72

The table above shows the aggregated metrics for a single training step. Throughout all three models, we see that as expected the DRAM memory throughput measured in GB/s increases hardware performance, moving from the T4 to L4 to V100 accelerator. For the MobileNet and Vision Transformer model, we also find the expected increase in peak performance measured in $GLOPS/s$ as we increase accelerator capabilities. Interesting, for the ResNet50 architecture, we actually see a decrease in peak performance going from T4 to L4, while V100 still performs the best.

This discrepancy could be explained based on architectural differences between the GPUs and their ability to efficiently

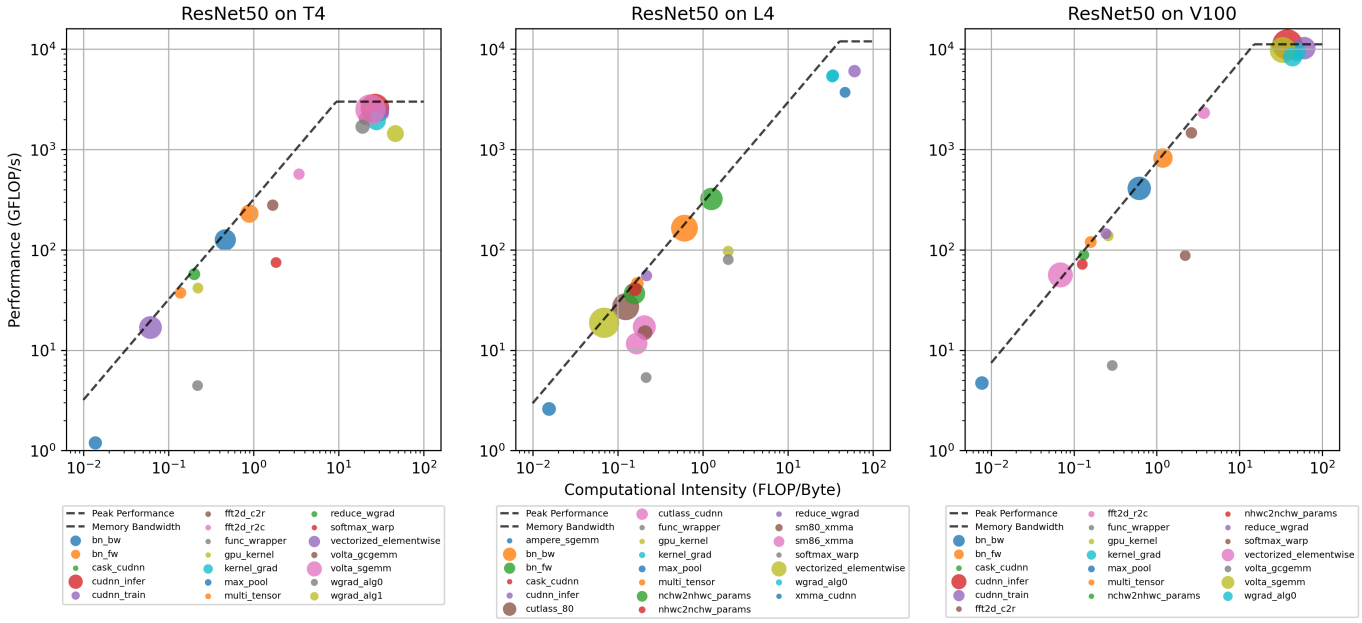

Fig. 3. We aggregate similar functioning kernels among the ResNet50 architecture. The size of each point is determined by its relative GPU run time, showing the overall impact a particular cluster has on the entire run job. We see the longest running kernels are within both the **compute and memory bound** regime

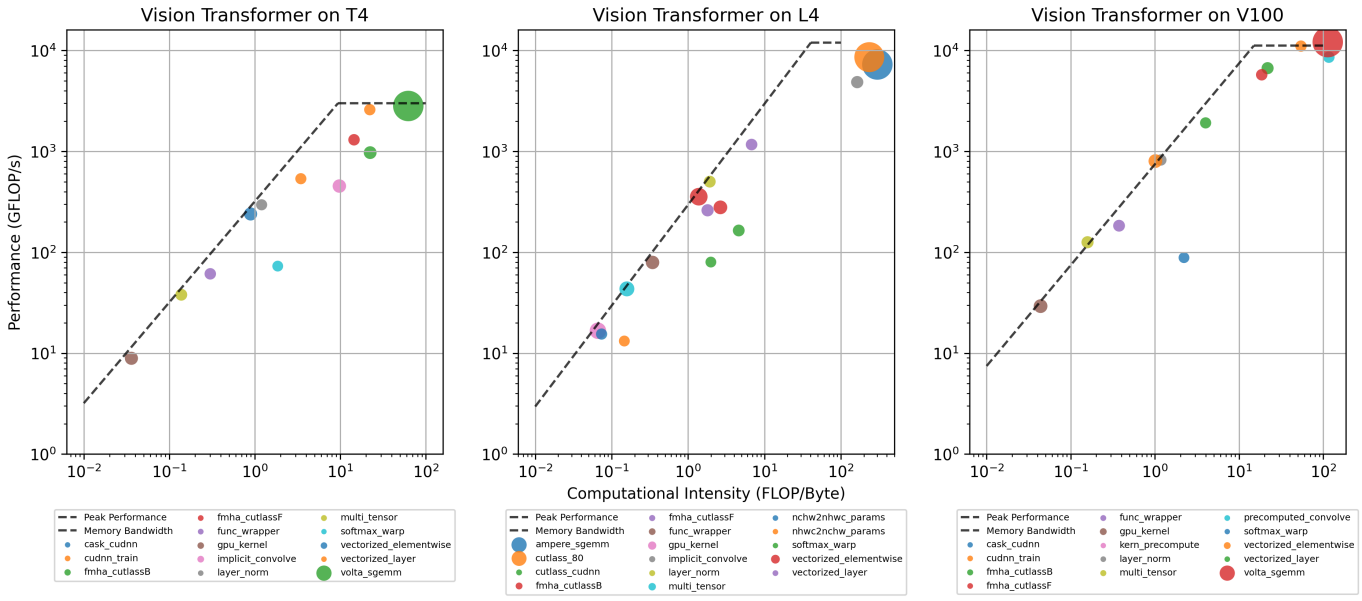

Fig. 4. We aggregate similar functioning kernels among the Vision Transformer architecture. The size of each point is determined by its relative GPU run time, showing the overall impact a particular cluster has on the entire run job. We see the longest running kernels are within both the **compute bound** regime

handle the computational patterns and memory access patterns of the ResNet50 model. The T4 GPU, based on the older Turing architecture, might be better optimized for the specific workloads and data patterns of ResNet50, leading to higher peak performance compared to the L4 GPU. Despite having lower overall computational capabilities than the L4, the T4 architecture may be better suited for convolutional and pooling operations prevalent in ResNet50, resulting in more efficient utilization of its resources. On the other hand, the L4 GPU, based on the newer Ada Lovelace architecture, is designed to handle a wider range of workloads, including attention-based models like Transformers [10]. While it offers higher computational power and memory bandwidth, the L4's architecture might not be as well-optimized for the specific computational patterns of ResNet50, leading to a lower peak performance compared to the T4. Regardless, as a result of these general trends, we see training times tend to decrease as we improve hardware performance.

A. *Computational Intensity*

We find T4 and V100 architectures generally more computationally efficient for ResNet50 and L4 more suited for

MobileNet and Vision Transformer. This behavior could be attributed to the balance between computational power and memory bandwidth in these architectures.

Given that the L4 has the highest computational capabilities ($30.3\ TFLOPS$) relative to its memory bandwidth ($300\ GB/s$), it may be better optimized for computationally intensive workloads like Vision Transformers, rather than convolutional neural networks like ResNet50. Despite having higher computational capabilities, the L4 might not be as well-suited for the specific computational patterns and memory access patterns of ResNet50, leading to suboptimal utilization of its resources.

MobileNet, being a lightweight model, might benefit from the L4's higher computational capabilities, allowing it to better leverage the parallel processing power for its depthwise separable convolutions. The Vision Transformer, with its attention mechanisms and dense matrix operations, could take advantage of the L4's high computational throughput, resulting in better computational intensity compared to the T4 and V100.

VI. Roofline Models

The roofline analysis across the three GPU architectures provides valuable insights into the performance characteristics of the evaluated models. While the lightweight MobileNet remains predominantly memory-bound across all GPUs, the more computationally intensive models, such as ResNet50 and Vision Transformer, can better leverage the increased computational capabilities and memory bandwidth of the V100 and L4 GPUs, leading to a more balanced distribution of compute-bound and memory-bound kernels.

A. MobileNet

The roofline plots for MobileNet reveal that this lightweight model is primarily memory-bound on all three GPU architectures. This behavior is expected since MobileNet is designed for efficient inference on resource-constrained devices and may not fully utilize the computational resources of high-end GPUs.

B. ResNet50

On the T4 GPU, ResNet50 appears to be predominantly memory-bound, however, on the V100 and L4 GPUs, we see it near the middle of the memory-bound and compute-bound regimes. This observation aligns with the expectation that more powerful GPUs can better exploit the computational capabilities of ResNet50, a deeper and more complex convolutional neural network.

C. Vision Transformer

The roofline plots for the Vision Transformer model further reinforce the observations about ResNet50. On all three GPU accelerators, the aggregated training step appears to be computationally bound, which isexpected for the computationally intensive Transformer model, which can better leverage the computational resources of high-end GPUs.

VII. Kernel Level Analysis for each Model

Additionally, we plot kernel level performance metrics for each model architecture to highlight which operations most heavily impacted the overall training performance. To simplify analysis, we grouped kernels based on similar names as displayed in the legend of Figure 2-4. The size of each plotted kernel groups are based off of the collected *gpu_time_duration.sum* NVIDIA metric, which is an indication of how much each kernel contributed to the overall calcualted metrics for the entire training step.

A. MobileNet

From the MobileNet roofline plots in Figure 2, we observe that the majority of kernels are clustered near the memory bandwidth roof, indicating that the overall performance is primarily memory-bound for this lightweight model. The kernels that stand out as being more compute-bound include *cuDNN_train*, *volta_sgemm*, and *vectorized_elementwise*. These kernels are likely associated with the depthwise separable convolutions and other computationally intensive operations in the MobileNet architecture, which would make sense to stand out as individual compute bound kernels among a predominantly memory bound training regime. Notably, we see *conv_depthwise2d* taking a significant portion of each training step. We expect this low computation operation to be memory bound, which is shown in each roofline.

B. ResNet50

In the case of ResNet50, the roofline plots in Figure 3 reveal a more diverse distribution of kernels across the compute-bound and memory-bound regions, particularly on the T4 and V100 GPUs. *vectorized_elementwise* and *bn_bw* appear memory bound while *volta_sgemm* and *cudnn_infer* appear compute bound. Batch normalization is a technique used to improve the training stability and performance of deep neural networks, involving the computation of the mean and variance of the input data and normalizing the activations based on these statistics. This process can be memory-intensive, as it requires loading and storing intermediate tensors, especially for smaller batch sizes.

On the other hand, *volta_sgemm* is a kernel that performs general matrix-matrix multiplication (GEMM) operations, heavily used in the convolutional and fully connected layers of ResNet50. These operations are computationally intensive and can benefit from the parallel processing capabilities of GPUs, making them compute-bound on powerful architectures like the V100.

C. Vision Transformer

For the T4 and V100 architectures, we see the dominant *volta_sgemm* operation that is used frequently within how multi-head attention and feed-forward layers structure operations in the Transformer architecture. As for the L4 architecture, kernels such as *ampere_sgemm* and *cutlass_80* dominate the training step runtime. The later is a kernel related to the

CUTLASS (CUDA Templates for Linear Algebra Subroutines) library, which is a collection of high-performance CUDA kernels for linear algebra operations [11]. This aligns with the expectation that the attention mechanisms and dense matrix operations in Transformers are computationally intensive and can benefit from powerful GPU architectures.

VIII. Conclusion

The empirical results and roofline analysis presented in this study provide valuable insights into the performance characteristics of different deep learning models and GPU architectures. Through an in-depth evaluation of MobileNet, ResNet50, and Vision Transformer models on T4, V100, and L4 GPUs, several important observations were made.

Firstly, the computational intensity and memory requirements of the models played a crucial role in determining their performance on different GPU architectures. Lightweight models like MobileNet were predominantly memory-bound across all GPUs, indicating that their performance was limited by the available memory bandwidth rather than the computational capabilities. In contrast, more complex models like ResNet50 and Vision Transformer exhibited a mix of memory-bound and compute-bound behavior, especially on the more powerful V100 and L4 GPUs. The Vision Transformer model consistently demonstrated higher computational intensity due to its attention mechanisms and dense matrix operations. These findings highlight the importance of carefully considering the model architecture and its computational requirements when selecting the appropriate GPU hardware for efficient training and inference.

References

[1] I. H. Sarker, "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions," *SN Computer Science*, vol. 2, no. 6, p. 420, 2021.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[3] Nvidia. (2024) 4. nsight compute cli. [Online]. Available: https://docs.nvidia.com/nsight-compute/NsightComputeCli/

[4] facebookresearch, "fvcore," https://github.com/facebookresearch/fvcore/tree/main, 2024.

[5] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[7] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[8] Nvidia. (2024) Nvidia l4 tensor core gpu. [Online]. Available: https://www.nvidia.com/en-us/data-center/l4/

[9] ——. (2024) Nvidia tesla v100 the first tensor core gpu. [Online]. Available: https://www.nvidia.com/en-gb/data-center/tesla-v100/

[10] J. Morra. (2022) Nvidia's lovelace gpu upgrades graphics with ai. [Online]. Available: https://www.electronicdesign.com/technologies/embedded/article/21251092/electronic-design-nvidias-lovelace-gpu-upgrades-graphics-with-ai

[11] NVIDIA, "Cutlass 3.5," https://github.com/NVIDIA/cutlass, 2024.