CHARACTERIZING THE CARBON IMPACT OF LLM INFERENCE

Edwin Lim [1] **Zhiyang Pan** [1] **Yang Zhou** [1]

ABSTRACT

The rapid rise of Large Language Models (LLMs) has raised significant concerns regarding their environmental impact. The increased language model capability that arises from scaling their size has encouraged the development of exponentially larger models, which increases the carbon footprint of both inference and training. We present a comprehensive study that measures the energy consumption of running LLM inference across both different generations of datacenter-grade GPU platforms and different model sizes. To compare the scaling of hardware capabilities with the scaling of model requirements, we monitor power consumption, memory usage, and performance metrics such as token latency and throughput. Then, we calculate sustainability metrics such as carbon per token and model FLOPs utilization (MFU) to derive insights on how to reduce both the operational and embodied carbon emissions of LLM serving applications. We find that on modern GPU platforms, increasing batch size can lead to a 11x smaller carbon emission per token with only a 2x increase in user-perceived latency per token. Additionally, we find that the total utilization of compute FLOPs on datacenter-grade GPUs running inference workloads is only 2%, identifying opportunities for hardware platforms with less compute capacity that could reduce embodied carbon emissions.

1 INTRODUCTION

Climate change is one of the most pressing challenges of our time, and projections currently estimate that Information and Communications Technologies (ICT) could constitute up to 20% of carbon emissions by 2030 (Jones et al., 2018). This rapid rise in ICT emissions is in part due to the exponential growth of both Deep Learning (DL) applications and their model size, which increases model accuracy and capability at the expense of a larger environmental footprint (Wu et al., 2022).

This carbon footprint can be attributed to both the operation of DL applications, which has been estimated to constitute 0.25% of the USA's annual power consumption from running recently sold GPUs alone (Chien, 2023), and the embodied footprint of manufacturing DL hardware, which includes the estimated sales of 3.2 million NVIDIA A100 GPUs in 2022 and 2023 (Chien, 2023).

While these specialized hardware platforms are necessitated by their higher energy efficiency and performance compared to more general-purpose options, their increasing impact on the environment should be systematically measured and minimized. To ensure environmentally friendly advancement of ML capability and usage, emerging ML hardware should be designed to be efficient in terms of overall silicon utilization, and ML systems should prioritize carbon-efficiency as an optimization metric.

To this end, we take a step towards designing sustainable systems and hardware for ML by first understanding modern LLM performance and energy characteristics on existing off-the-shelf GPUs. Alongside our operational carbon analysis, we measure the utilization of individual resources such as memory bandwidth, memory capacity, and compute capacity across various LLM workloads. We use our analysis to quantify and compare the embodied versus operational carbon costs of LLM inference, and identify potential methods to reduce overall carbon emissions.

2 MOTIVATION

The compute and memory requirements of training and serving large models is growing exponentially. For modern transformer-based models in particular, total training FLOPs and parameter count has been scaling by 750x/2yrs and 410x/2yrs, respectively (Gholami et al., 2024). The required FLOPs for inferencing these growing transformer models also follows this trend. Figure 1 demonstrates how to total TFLOPs required for inferencing LLMs scales linearly with respect to both the sequence length and the model size.

Aside from the rapidly increasing compute requirements,

[*]Equal contribution [1]Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, USA. Correspondence to: Edwin Lim <eglim@andrew.cmu.edu>, Zhiyang Pan <zhiyangp@andrew.cmu.edu>, Yang Zhou <yangzho6@andrew.cmu.edu>.

Final course project report for "15-642: Machine Learning Systems" Spring 2024, Pittsburgh, PA, USA, 2024.

DRAM bandwidth and capacity requirements have quickly become a bottleneck for efficient transformer inference. This is due to the requirement of storing model weights in DRAM, along with the inference optimization of KV-caching, which eliminates the need of recomputing "key" and "value" tensors of previous tokens at the expensive of storing them in DRAM. Figure 2(a) illustrates the scaling of required DRAM capacity for serving a single inference request with respect to model size and sequence length, with the required memory for KV-caching dominating as sequence lengths grows.

However, the capability of modern AI hardware platforms is struggling to keep pace. Figure 2(b) demonstrates the scaling of compute capacity, memory capacity, and memory bandwidth across different generations of NVIDIA GPUs. Peak TFLOPs of these platforms has been scaling by about 3.2x/2yrs, whereas HBM bandwidth and capacity have been scaling by only 1.6/2yrs and 1.7x/2yrs, respectively. The difference in scaling speeds of compute and memory, caused by the difficulties in technology scaling for memory devices, leads to an exponentially growing gap between the compute capacity of a device and its HBM capacity. For LLM inference in particular, which suffers from quickly growing memory overheads, this contributes to a growing memory bottleneck.

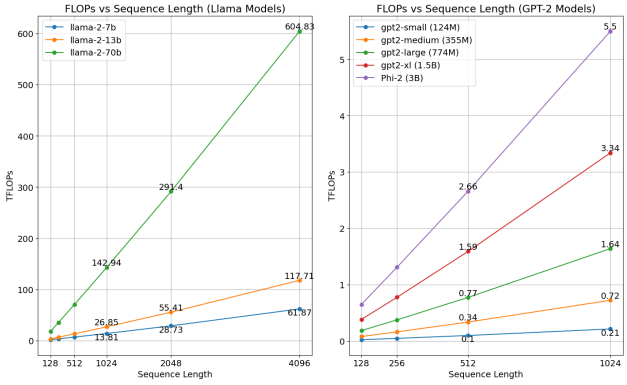

Figure 1. (a) Llama2; (b) GPT2 .

The growing gap between model requirements and hardware capabilities causes significant carbon emissions because (1) the number of GPUs required to train and serve a model grows larger and leads to more embodied carbon emissions, (2) the amount of FLOPs and data movement operations for training and serving also increases, causing more operational carbon emissions.

Our research seeks to answer the question of how the operational carbon emissions from LLM inference scales with model size, and whether we can trade off latency per token in order to reduce the operational carbon emission per token.

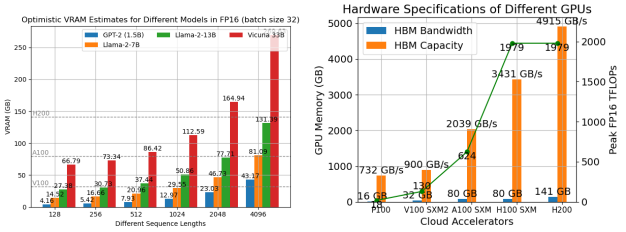

Figure 2. (a) The VRAM for models scales exponentially as the size of the model goes up; (b) GPU development trend over the last decade.

Additionally, we seek to identify opportunities for reducing the embodied carbon emissions of LLM inference by examining the overall Model Flops Utilization (MFU) of datacenter-grade GPUs under inference workloads, potentially highlighting the need for separate hardware platforms for inferencing and training.

3 Related Works

While modern DL-focused hardware platforms such as GPUs offer both competitive performance and the flexibility to accelerate a wide variety of DL applications, their silicon is both power-inefficient (Patel et al., 2023) and severely underutilized (Weng et al., 2022) in large-scale cloud deployments. While previous works have examined cluster-level scheduling techniques in order to keep coarse-grained GPU utilization high, our work instead examines the finer-grained silicon utilization metrics of overall compute unit utilization given a GPU with an available workload. Combined with efforts to keep cluster-level GPU utilization high, a high utilization efficiency of individual GPU silicon promises further reduction in the embodied emissions associated with serving large-scale models.

Additionally, we examine the energy-performance tradeoff on the granularity of individual platforms, whereas previous works analyze the methods to reduce inference energy on a cluster-level scale (Patel et al., 2024). Our work, while orthogonal to the coarser-grained power management techniques, presents further opportunities for operational carbon reduction of LLM inference when used in tandem with cluster-level power management techniques.

Previous work has also examined energy per token and overall GPU utilization numbers for LLM inference across both single-node and multi-node GPU configurations (Samsi et al., 2023), which demonstrates high utilization of GPU compute units from numbers reported by NVIDIA DCGM. Our work builds upon this by incorporating carbon emissions analysis on top of the energy usage analysis, and examining GPU utilization from the hardware perspective

of theoretical versus achieved FLOPs.

4 METHODOLOGY

4.1 System Overview

We propose and develop a carbon profiling framework, , to perform carbon and performance characterization of LLM workloads. Our framework consists of a workload generator which synthesizes inference workloads from the open-source prompt dataset for representative queries, a wrapper to run our workload generator with heterogenous GPU configurations, and real-time power monitors which measure GPU power usage at a second-scale granularity. Then an output parsing component that analyze the utilization and estimates carbon usage.

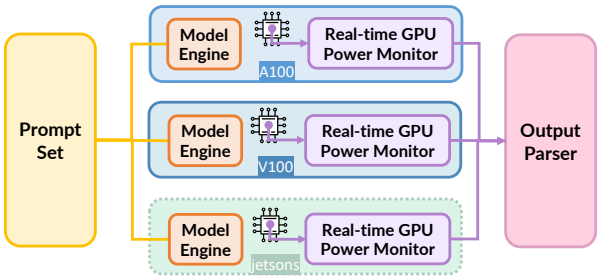

Figure 3. An overview of 's carbon profiling system of LLM inference on off-the-shelf GPUs.

4.2 Model Inference Framework: Tensorrt-LLM

We leverage NVIDIA's TensorRT-LLM libraries to run our experiments, which offers state-of-the-art engines for efficient LLM inference that can be utilized through a PyTorch-like Python API for NVIDIA GPUs. The engines built with TensorRT-LLM can be executed on a wide range of GPU configurations, utilizing Tensor Parallelism and Pipeline Parallelism to scale from single GPU nodes to multi-GPU nodes. We limit our analysis to models that can fit on single-GPU nodes due to the time constraints of a course project.

4.3 Power Measurement: NVSMI

We utilize Nvidia's command-line utility for querying and managing various states of NVIDIA GPU devices. While the workload generator is running, we utilize a separate script to continuously profile power, temperature, memory usage, and SM utilization on intervals of 1-3 seconds.

4.4 Carbon Modeling

Using power measurements from our experiments, we plot power against time and integrate to find the area under the discrete measurements and calculate the total energy (joules) consumed during the inferencing. We then use a carbon-intensity trace from the region in which the data center for the GPU is located, which converts units of energy into grams of CO2eq emissions depending on the amount of renewable energies available in the current geographic region.

Table 1. NVIDIA V100 Hardware Specification

	NVIDIA V100
GPU	Two NVIDIA GV100GL (Tesla V100S PCIe 32GB)
CPU	Two 32-core AMD 7542 at 2.9GH
RAM	512GB ECC Memory (16x 32 GB 3200MHz DDR4)
Disk	One 2TB 7200 RPM 6G SATA HDD
NIC	Dual-port Mellanox ConnectX-5 25 Gb NIC (PCIe v4.0) /BlueField2 100 Gb SmartNIC

Table 2. NVIDIA A100 Hardware Specification

	NVIDIA A100
GPU	NVIDIA HGX A100 GPU (4x 40GB A100 SXM4 GPUs)
CPU	Two 24-core AMD 7413 at 2.65GHz
RAM	512GB ECC Memory (16x 32 GB 3200MT/s RDIMMs)
Disk	Two 480 GB 6G SATA SSD/One 1.6 TB PCIe4 x4 NVMe SSD
NIC	Dual-port Mellanox ConnectX-6 DX 100Gb NIC

Given the total amount of output tokens generated during the inference process, we are then able to calculate the carbon emissions per token across our different experiments. estimates carbon emission via the following formula:

$$\text{Carbon Emission} = \left(\int_0^T Power \right) \cdot \text{Carbon Trace}$$

The carbon intensity traces are real-time data collected from an open-sourced platform "Electricity Maps" (ElectricityMaps, 2023).

then calculates carbon per token to understand the emission implication of each GPU hardware.

Carbon / Token = Carbon Emission / Total Generated Tokens

5 EXPERIMENTS

5.1 Hardware Platforms

We compare the model performance on two Nvidia Tensor Core GPUs: A100-SXM4-40GB and V100S-PCIE-32GB. The servers are located in CloudLab data centers, in Wisconsin. The hardware specifications are detailed in Table 1. A single V100-PCIE-32GB board advertises 112 TFLOPS of compute, and 32GB of HBM2 memory that supports 900 GB/s of bandwidth. An individual platform in the A100-SXM4-40GB pod boasts 624 TFLOPS of compute (5.6x of V100), along with 40GB of HBM2 capacity (1.25 of V100) that supports up to 1,555 GB/s of bandwidth (1.7x of V100).

5.2 Models & Hyperparameters

We characterize the performance and utilization of the Llama2 and GPT2 models with varying sizes, in order to

understand how carbon per token and GPU compute utilization scale across (1) GPU generations and (2) LLM model size.

For the scope of the course project, a knob that we choose to tune is batch size. Increasing the batch size of inference requests increases the throughput of tokens at the expense of potentially increasing the latency per token that a user would experience. Consequently, it presents an opportunity to reduce operational carbon emissions by outputting more total tokens per second at a similar power budget to single batch inference. We profile the models with TensorRT-LLM by consecutively varying the batch size from 1 until the GPU hardware platform encounters an out-of-memory ("OOM") error. This hard memory limit allows us to fully saturate the GPU capacity before drawing sustainability conclusions. The maximum achievable batch sizes for Llama2 7B and 13B are detailed in table 3, and the maximum achievable batch sizes for GPT2 137M, 812M, and 2B are detailed in table 4.

Table 3. The maximum batch size for each tested Llama2 model until OOM.

Model Size	V100 32GB Max. Batch Size	A100 80GB Max. Batch Size
7B	14	22
7B, WQ4bit	22	N/A
13B	4	8
13B, WQ4bit	12	N/A

Table 4. The maximum batch sizes for each tested GPT2 model until OOM.

Model Size	V100 32GB Max. Batch Size
137M	544
812M	128
2B	72

5.3 Prompt Set

In our research, we employed the ShareGPT prompt dataset (Team, 2023) to conduct profiling experiments. This dataset comprises authentic interactions between users and the closed-source ChatGPT model. To ensure data consistency and relevance, we refined the raw prompt data through several steps with following similar guidelines vLLM workload generator(Kwon et al., 2023). Firstly, we constrained the upper bound intput and output token limit to 1000 for Llama2 models, and 500 for GPT2 models. Secondly, we established a lower bound token limit of 4. Thirdly, we eliminated conversations comprising fewer than 2 rounds. Subsequently, for each experiment iteration, we randomly sampled 100 prompts from the refined dataset. These prompts were then fed into the model to emulate real-world serving workload scenarios.

6 Evaluations

6.1 Latency Throughput Trade-off

When serving LLMs, latency is often a critical optimization metric, as many companies may want their services or chatbots to feel as interactive as possible. Throughput is also an important consideration, as enabling higher tokens per second on a single GPU reduces the amount of GPUs required to run services. Increasing the throughput can be easily achieved by increasing the batch size of requests, but may come at the cost of an end user experiencing a higher per-token latency for their individual query. To quantify this tradeoff, we calculated the average tokens per second and average user-perceived seconds per token across all the batch sizes in our evaluation.

For unquantized Llama2-7B inference on a single A100 node, we show that by increasing our batch size from 1 to 22, we achieve a 10x improvement in tokens per second at a cost of only 2x higher user-perceived seconds per token. We observe similar trends across the different model types, and all user-perceived seconds per token remains under the average human token reading rate of around 5 tokens per second. As these models are relatively small compared to modern models that scale to hundreds of billions of parameters, memory issues arise before we are able to violate the human readability SLO. We leave the evaluation of larger models to future work to further examine this tradeoff.

6.2 GPU Utilization

To identify opportunities to reduce the carbon emissions of datacenter-grade GPUs for inference workloads, we calculate the total compute FLOPs utilization of our experiment platforms. We use the calflops tool (xiaoju ye, 2023) in order to calculate the total amount of model FLOPs necessary to compute every sequence of a given batch in our experiment with a specified model. Then, based on the latency of of the batch and the peak TFLOPS achievable by the test hardware platform in the duration of the batch completion, we calculate the model flops utilization (MFU) of the GPU given by the following definition.

MFU = achieved FLOPs / peak theoretical FLOPs

Our calculations show that for a batch size of 1, the A100 GPU is able to achieve only around 2.43% and 2.34% of peak FLOPs utilization of Llama 7B and 13B models respectively, whereas the V100 GPU is able to achieve only about 1.5% of peak FLOPs. This can be explained due to LLM inference being unable to utilize the Tensor Cores of the GPU due to its auto-regressive nature. Because the Tensor

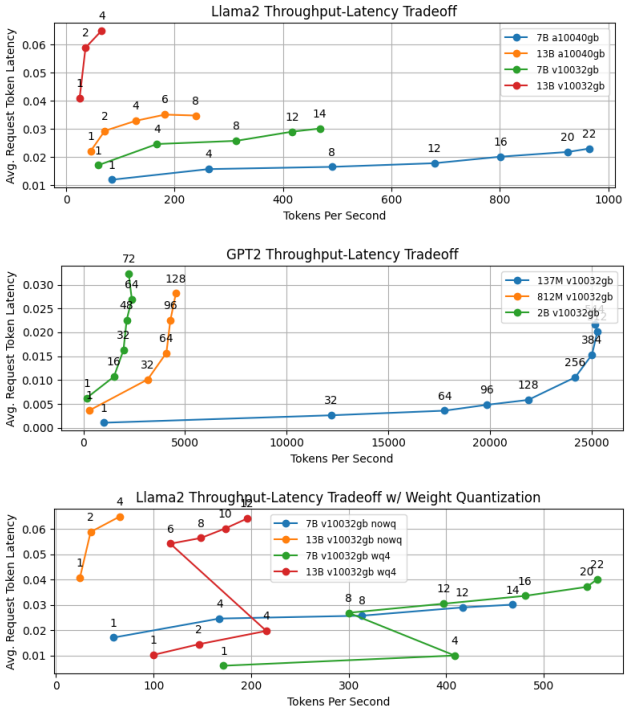

Figure 4. (a) throughput-latency tradeoff for Llama2 without weight quantization; (b) throughput-latency tradeoff for GPT2 without weight quantization; (c) throughput-latency tradeoff for Llama2 with weight quantization

Cores account for the majority of GPU FLOPs, the utilization of total compute during inference workloads is quite small. This identifies significant opportunities for embodied carbon reduction in inference-optimized LLM hardware. While GPUs can be used for both training and inference, many GPUs for a large service still must be dedicated inference nodes in order to meet request demand. Thus, if these GPUs opted for less tensor cores than their training-optimized counterparts, considerable overall die area (and thus embodied carbon emissions) could be saved.

6.3 Energy Consumption

As described in Section 4.3 that collects the power reading throughout the characterization experiments. Here, Fig. 6 (b) and (c) show the raw power reading during each experiment.

From the raw power reading, A100 demonstrates higher power consumption than V100 for inference of the same model. Take Llama2-7B as an example, the bright orange line on the top of Fig. 6 (b) is A100, whereas the corresponding V100 is the dark purple line in the medium range of Fig. 6 (b). This result aligns with our expectation as A100 has a higher number of FLOPs than V100, which consumes more

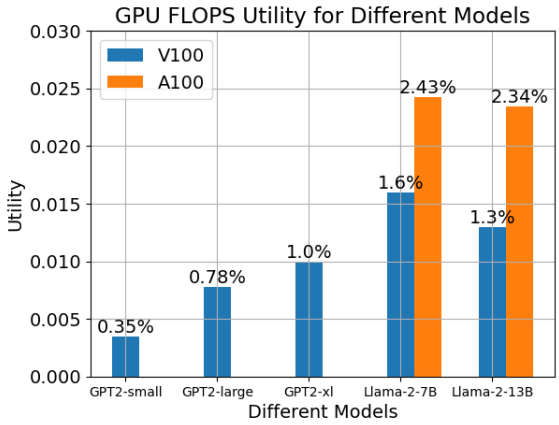

Figure 5. GPU model flops utilization (MFU) with batch size 1

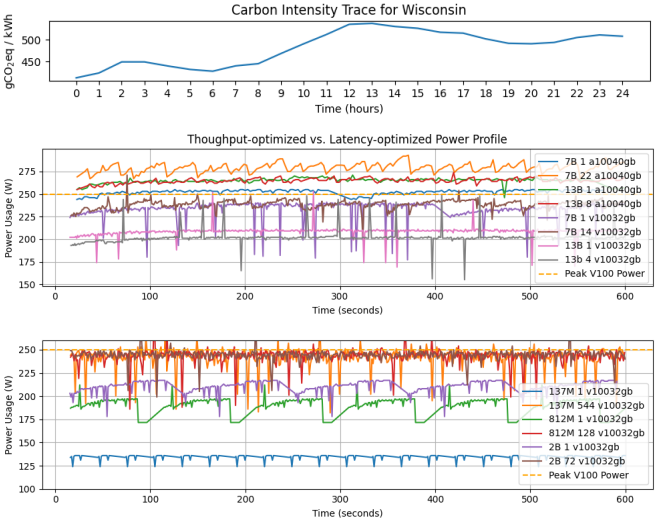

Figure 6. (a) The 24-hour carbon intensity trace for the Wisconsin region on May 3rd, 2024; (b) The raw power reading of Llama2 inference experiments; (c) The raw power reading of GPT2 inference experiments; The legend follows the format of:number_of_parameters batch_size GPU_memory.

energy than running V100. In that, higher-performing GPUs consume more energy per second than lower-performing GPUs.

Another dimension of the analysis is batch sizes. Again, from the raw power reading, we can see in Fig. 6 (c) that a bigger batch size uses more energy per second than a smaller batch size.

However, this is not accurate to say that higher-performing GPUs or bigger batch sizes consume more energy per second. One important factor that is not shown in Fig. 6 is the duration of the entire experiment. If we refer back to

Fig. 4, higher-performing GPUs and bigger batch sizes both result in lower average request token latency. Thus, we will examine the sustainability implication of LLM inference with the "per-token" metrics in the upcoming section.

6.4 Carbon Footprint

Now that we have the energy consumption numbers, the next step is to calculate the carbon emission of each LLM inference experiment. To collect real carbon intensity data, we took real-time carbon emission data from Wisconsin, the region where the GPUs' data center is located at as described in 5.1. For the scope of the project, we ran the experiment for less than an hour. Thus, we selected a single hourly carbon intensity data point 413 gCO_2eq/kWh from the Wisconsin region for the carbon emission calculation. Our analysis shows that for Llama-7B inference on the A100 node, we can reduce the carbon per token by 11x by increasing the batch size from 1 to 22. Given that this batch size increase only results in 2x higher user-perceived seconds per token, we demonstrate that prioritizing high batch sizes on GPU platforms is a promising avenue for **operational carbon emissions reduction** for LLM inference.

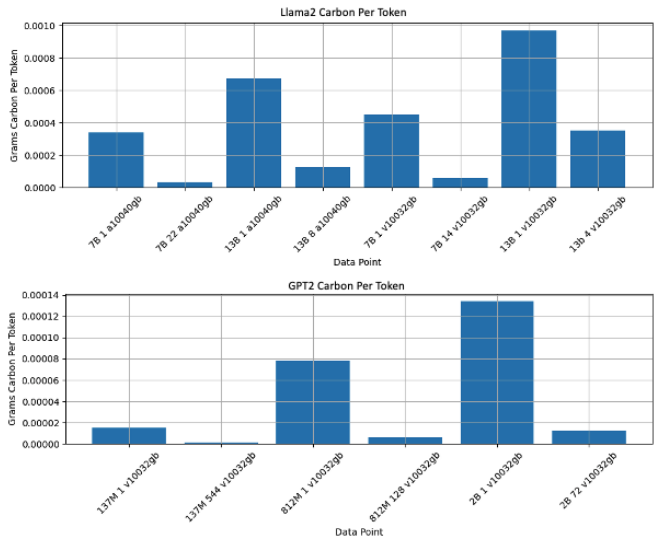

Figure 7. (a) The carbon emission per token for Llama2 inferences. The figure compares emissions at a batch size of 1 and at a maximum batch size; (b) The carbon emission per token for GPT2 inferences. The figure compares emissions at a batch size of 1 and at a maximum batch size; The legend follows the format of:number_of_parameters batch_size GPU_memory.

If we compare across GPU platforms, A100 is 1.3x more carbon emission in terms of carbon per token for Llama2-7B (batch size of 1) inference as shown in Fig. 7. In that, we demonstrated that higher performing GPUs are also more **operational carbon efficient** than lower performing GPUs.

7 Conclusion

Our work, , characterizes opportunities for reducing the carbon emissions of LLM inference on GPU platforms. We demonstrate that prioritizing large batch sizes can lead to an 11x decrease in operational carbon per token compared to a batch size of 1, meaning that schedulers that optimize for throughput during LLM inference have significant opportunities for carbon savings. We also highlight the issue of extremely low peak FLOPs utilization of GPU platforms (2%) in LLM inference that is caused by their auto-regressive nature, which identifies leaner compute platforms as opportunities for embodied carbon reduction for LLM inference. Due to the unique characteristics of LLM inference workloads, there is great opportunity to reduce its carbon emissions if sustainability is used as a first-order design metric. This work calls for awareness of building sustainable systems for machine learning, demonstrates opportunities in the design space, and serves as the first step of the authors' ongoing research project.

GitHub Link, for A100/V100: https://github.com/edwinlim0919/sustainable-deep-learning/tree/main
GitHub Link, for Jetsons: https://github.com/melishua/jetson-profiler
Youtube Link: https://www.youtube.com/watch?v=V6VK4kZLdO8
Presentation Slide Link: https://docs.google.com/presentation/d/1VdTn6PMlEFfs-Xid3dE7pKaQ0CeWJgRJY_TOWVBeGkk/edit?usp=sharing

Acknowledgements

We would like to thank our advisors Akshitha Sriraman, James Hoe, and Beidi Chen for supporting our research endeavors.

References

Chien, A. A. Genai: Giga$$$, terawatt-hours, and gigatons of co2. *Communications of the ACM*, 2023.

ElectricityMaps. Electricity Maps, 2023. URL https://github.com/electricitymaps/electricitymaps-contrib#data-sources/tree/master/parsers.

Gholami, A., Yao, Z., Kim, S., Hooper, C., Mahoney, M. W., and Keutzer, K. Ai and memory wall. *IEEE Micro*, 2024.

Jones, N. et al. How to stop data centres from gobbling up the world's electricity. *Nature*, 2018.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient

memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pp. 611–626, 2023.

Patel, P., Gong, Z., Rizvi, S., Choukse, E., Misra, P., Anderson, T., and Sriraman, A. Towards improved power management in cloud gpus. *IEEE Computer Architecture Letters*, 2023.

Patel, P., Choukse, E., Zhang, C., Goiri, Í., Warrier, B., Mahalingam, N., and Bianchini, R. Characterizing power management opportunities for llms in the cloud. In *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 3*, pp. 207–222, 2024.

Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., and Gadepally, V. From Words to Watts: Benchmarking the Energy Costs of Large Language Model Inference. *arXiv e-prints*, art. arXiv:2310.03003, October 2023. doi: 10.48550/arXiv.2310.03003.

Team, S. Sharegpt. https://sharegpt.com/, 2023.

Weng, Q., Xiao, W., Yu, Y., Wang, W., Wang, C., He, J., Li, Y., Zhang, L., Lin, W., and Ding, Y. {MLaaS} in the wild: Workload analysis and scheduling in {Large-Scale} heterogeneous {GPU} clusters. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, 2022.

Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., et al. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 2022.

xiaoju ye. calflops: a flops and params calculate tool for neural networks in pytorch framework, 2023. URL https://github.com/MrYxJ/calculate-flops.pytorch.

A PLEASE ADD SUPPLEMENTAL MATERIAL AS APPENDIX HERE

Put anything that you might normally include after the references as an appendix here, *not in a separate supplementary file*. Upload your final camera-ready as a single pdf, including all appendices.