Evaluating Software Carbon Intensity of Foundation Models at Inference Stage

Dr. Ziliang Zong
Computer Science Department, Texas State University, San Marcos, Texas

Summary: Foundation models (e.g., BERT, GPT-3, CLIP, DALL-E 2) refer to large scale complex models that are trained on massive amounts of data and can be adapted to a wide range of downstream tasks. Despite the great potential of foundation models, they are extremely resource hungry and can generate significant carbon emissions. However, our knowledge about the carbon impact of different foundation models is very limited due to the lack of measurement tools, standard methodology, and evaluation metrics. To address this problem, this project conducts a comprehensive study on evaluating the carbon impact of several open-source foundation models (e.g., GPT-J 6B, GPT Neo 2.7B, GPT-NEO 1.3B, GPT-2) at inference stage using the Software Carbon Intensity (SCI) specification recently released by the Green Software Foundation. This study shows that (1) the quality and environmental impact of foundation models can be quantitatively measured and compared; (2) it is possible to replace carbon-intensive foundation models with more efficient ones without sacrificing model quality; (3) deploying foundation models on more efficient hardware (e.g. GPUs) can significantly reduce SCI; and (4) SCI is an effective metric to evaluate the carbon impact of different foundation models at the inference stage.

I. Introduction

In the past decade, we have witnessed the explosion of artificial intelligence (AI) applications in various domains such as natural language processing (NLP), computer vision, voice recognition etc. Meanwhile, the size and complexity of AI models have also increased exponentially. OpenAI reported that (plotted in Fig. 1 [1]) the required computing to train state-of-the-art deep learning models have increased 300,000x since 2012 and its growth rate has exceeded Moore's Law.

Recently, foundation models (e.g., BERT, GPT-3, CLIP, DALL-E 2) have shown their unprecedented power to be adapted to a wide range of downstream tasks after being trained on broad data at scale. These models will not only transform how future AI applications are built, but will also consume excessive power and generate far more carbon emissions than most people realize. For example, training the GPT-3 model consumed approximately 190,000 kWh of energy and produced 85,000 kg of CO_2 [2]. Therefore, it is paramount to fully study the environmental impact of foundation models at both the training and inference stage. In 2021, Stanford university released a report to urge developers and large-scale deployers of foundation models to consider how they can mitigate any unnecessary carbon emissions [3].

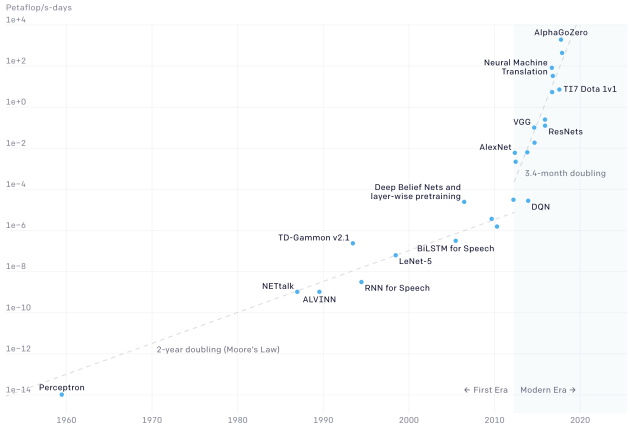

Fig. 1. Computing Trend to Train State-of-the-Art AI Models [1]

However, investigating how to reduce the carbon emissions of foundation models is not a trivial task for several reasons. First, most foundation models are currently not accessible to the research community. Second, only several IT giants have the computing resources and datasets needed to train these models. Third, the lacking methodology, metrics, and power measurement tools have impeded the research progress in understanding the carbon emissions of foundation models.

In this project, we conduct a comprehensive case study in exploring the carbon impact of foundation models at the inference stage. More specifically, we investigate several open-source foundation models, including GPT-J 6B [4], GPT-Neo 2.7B, GPT-Neo 1.3B, GPT-Neo 125M [5], and GPT-2 [6], which are released by the EleutherAI group [7] at Hugging Face. Since these models are already trained, we focus on evaluating their energy consumption and carbon emissions at the inference stage. Even though foundation models can generate large energy costs and carbon emissions during the training phase, their environmental impact can be larger when widely deployed to service millions of requests. To narrow down the scope of our research, we select a specific domain - Natural Language Processing (NLP) - in our study. We use the State of Texas Assessments of Academic Readiness (STAAR) [8] as the metric to quantitatively evaluate the quality of aforementioned foundation models and use the Software Carbon Intensity (SCI) specification (released by the Green Software Foundation in 2021) [9] as the metric to

quantitatively evaluate the carbon impact of these foundation models.

Our study makes the following contributions:

1) We demonstrate that SCI is an effective metric that can quantitatively evaluate the carbon impact of various foundation models.
2) We argue that evaluating the quality of foundation models is equally important while studying their carbon impact. In this study, we utilize State of Texas Assessments of Academic Readiness (STAAR) [8] as the metric to quantitatively evaluate the quality of foundation models. These models produce answers to the same set of well-selected questions and a Writing Performance Score (between 3 - 12 and a higher score indicates better quality) is assigned to each model according to the STAAR rubric. These scores can be used to determine higher quality outputs. For example, the Writing Performance Score of GPT-J 6B is 5.83, which is 12% lower than the GPT-Neo 2.7B with a score of 6.63.
3) We find that carbon intensive models do not necessarily yield better quality. For example, we observe that when being asked the same set of questions, the answers generated by GPT-Neo 1.3B have similar quality of answers generated by GPT-J 6B but GPT-Neo 1.3B only consumes 27% of energy. Replacing GPT-J 6B with GPT-Neo 1.3B will help mitigate energy requirements and reduce carbon waste without compromising quality.
4) We confirm that deploying foundation models on CPUs or GPUs will not affect model quality. However, leveraging GPU-acceleration has significant benefits on both response time and energy efficiency.

II. Related Work

Research in evaluating the carbon impact of AI models in general and foundation models in particular is still in its infancy. Among the limited works in this field, most of them focused on studying the energy expenditure during the training phase. The most substantially useful rationale for tracking energy consumption of AI is found in Henderson et al. as regards environmental impact [3], which makes the case that carbon impact from foundation models can, and should, be mitigated and offer several solutions during training, including selecting energy grids with minimal carbon emissions, using more efficient hardware to train models, using more efficient models, and distilling models to be more applicable.

While information about energy usage of existing AI models is notably sparse, there is information about energy expended during the initial training. Patterson et al. investigated methods to improve energy efficiency and carbon emission reduction for model training, including geographically locating to optimize carbon output, utilizing cloud providers as they may be more energy efficient, and being explicit about power consumption to aid end-users in selecting appropriate models to perform tasks [10]. Similarly, Anthony et al. provided carbontracker as a means to track and predict the energy usage and carbon emissions of training deep learning models [11]. This tool provided users of these models insight into their environmental impact during deployment. Gao et al. offered a method for modeling and characterizing AI and big data workloads called Data Motifs [12]. Eight motifs were identified, which is useful for determining the types of expected computational requirements for AI workloads. Labbe explained in detail the huge negative environmental impact incurred when training large AI models [13]. He used MegatronLM as an example, showing that training this model used as much energy as three homes during a calendar year. Toews gave insights as to the growing problem of AI training by demonstrating the difference between GPT-2 and GPT-3 [14]. The former took a dozen petaflop days to train while the latter required several thousand - a massive increase. He made the case that the problem will continue to get worse due to the greater reliance on AI models. DeWeerdt made a similar case using carbontracker [11] and recommends such tools as necessary to track energy usage of deployed AI models [15].

III. Foundation Model Selection

The first problem faced when comparing foundation models against each other is selecting something that can be easily understood without extensive expert knowledge. For example, if we were to compare various chess playing programs against each other, it might be difficult, if not impossible, to determine which results of two outputs were better unless we had these AI programs playing against Grand Masters, much like Deep Blue's performance was evaluated against Garry Kasparov in 1996 [16]. While it is mathematically possible to determine which chess move is better, this requires essentially using a vast amount of computing resources to compare those two moves. Hence, we prefer to select something that lends itself to easier "normal human" analysis, i.e. something that is not only more intuitive, but which can be evaluated by a non-expert.

A. Natural Language Processing

Given the above concerns about the required analysis, we select Natural Language Processing (NLP) for this study for three reasons:

1) Native speakers of a language will generally notice something is "off" about a particular sentence or response. This allows the typical researcher to more easily segregate initial outputs before a more lengthy in-depth analysis takes place.
2) There are a multitude of rubrics available to evaluate the English language that can concretely classify statements as "good" or "bad" grammar, allowing a more comprehensive analysis after the initial phase of selecting coherent statements.
3) Recently, several NLP implementations of foundation models have been made directly accessible and can be run on physical, commodity hardware. Accurate power measurement that might not be possible when using such models in the cloud or via remote systems is thus possible.

B. Selected Foundation Models

To evaluate the carbon impact of NLP implementations of state-of-the-art foundation models, we explore several models based on GPT-2 or GPT-3, which are the two most widely known foundation models used to perform NLP , both made available by OpenAI in 2019 and 2020, respectively [1]. Since the original GPT-3 model is accessible only via API [17], it does not allow us to deploy on local servers and collect power or carbon related data. Therefore, we select several pre-trained open-source foundation models released by EleutherAI [7] in our study. These models include GPT-2, GPT-3 variants, and GPT-J 6B, which is intended to supplant GPT-3 as an open-source version of that model. More importantly, we can deploy these models on our local system, build power and carbon profiling tools to collect information, and change deploy configurations if necessary. The details of the selected five foundation models can be found below:

- **GPT-J 6B**: GPT-J 6B is a 6 billion parameter model that, as previously stated, attempts to replicate GPT-3's capabilities while remaining open-source [4]. It was trained on the Pile, a 825 GB open-source language modelling data set consisting of 22 smaller sets combined [18]. Per Naik, GPT-J 6B compares very favorably to GPT-3, and could be further fine-tuned to produce even better outputs [19].
- **GPT-Neo 2.7B, GPT-Neo 1.3B, GPT-Neo 125M**: GPT-Neo is a family of models that were designed as implementations of GPT-3, also trained on the Pile [5]. The values after the family name refer to the number of parameters utilized in each model, which are 2.7 billion, 1.3 billion, and 125 million, respectively.
- **GPT-2**: GPT-2 is a 1.5 billion parameter transformer model that was first made available in 2019 [6]. It is included in this study because we want direct comparisons between an older foundation model and newer implementations. The model we use is from the HuggingFace repository [20].

C. NLP Prompt Selection

It is our belief that the quality of foundation models should be evaluated together with their carbon impact to make reasonable conclusions. A low-quality model will not be widely used even though it is carbon efficient. Our goal is to seek low carbon foundation models without compromising quality. However, evaluating the quality of foundation models itself is a daunting task. Since the context of our study focuses on NLP, we adopt the "ask a question, get a response" approach, which has been widely accepted for over 50 years since Joseph Weizenbaum releasing the ubiquitous ELIZA software in 1966 [21]. More specifically, we generate a list of 10 questions or sentence starters that the models would use as a basis to produce an output. While all of the prompts are open-ended, they vary in what sorts of answers are to be expected. For instance, when being asked how someone's day is going, we can expect a wide variety of valid responses. The foundation model would thus, too, be expected to produce answers that might wildly differ from each other. However, when asking a more concrete question such as "what is an elephant", a valid response might indeed include a story about how elephants were observed during an African safari, but a more objective reply would most likely list physical attributes and expected habitat. Hence, the prompts have an inherent bias for response type (direct answer vs. creative exposition), but also allow outputs outside the norm. In our study, we select the following 10 prompts:

1) What is an elephant?
2) What is a lemur?
3) How are you doing?
4) Guess my weight
5) Can you dance?
6) Good day to you, sir
7) What is a computer?
8) Who is Elon Musk?
9) Are you alive?
10) Mad Max is a

TABLE I
FOUNDATION MODEL SIZE COMPARISON

Model	Disk Usage
GPT-J 6B	22.5 GB
GPT-Neo 2.7B	9.5 GB
GPT-Neo 1.3B	4.6 GB
GPT-Neo 125M	502 MB
GPT-2	2.2 GB

IV. TESTING ENVIRONMENT

To fairly compare each model, it is critical to configure an environment that allows all foundation models to be run under identical circumstances. The following subsections explain how that environment is configured in our experiments.

A. Physical System

All experiments are conducted on the same server which contains an AMD Ryzen Threadripper 2950x processor (16 physical cores with hyperthreading support for 32 threads), 4 Nvidia RTX 2080TI GPUs, and 128GB of DDR4 Memory in quad channel configuration.

B. Supporting Libraries

The Pytorch [22] and Transformers [23] libraries are installed on our development system, and configured to allow direct interface to the HuggingFace repository via Python scripts. Additionally, CUDA [24] [25] is installed to support GPU acceleration. To minimize the impact of network delay and enable power profiling, all foundation models are downloaded to local storage, which consume approximately 39 GB of total disk space (See Table I for a breakdown of model size).

C. Power Profiling Tool

Two system profiling tools are developed to collect real-time information about CPU utilization, CPU power consumption as well as multiple GPUs' utilization and power consumption. To ensure the profiling tool is lightweight, we choose a sampling frequency of 1Hz, which has a negligible impact on the execution of foundation models. The profiling for GPUs is via the Nvidia Management Library (NVML). The power consumption data of the AMD Ryzen Threadripper processor is collected from the Model Specific Register (MSR) files. We can access different registers by seeking for the MSR number as an offset. For example, we can go to each $cpunum$ MSR directory and read 8 bytes starting from the offset $0xC001029A$ to get per CPU core energy. The package energy (i.e. total energy consumed on a single cpu chip) can be obtained from the offset $0xC001029B$. Since the recorded energy value is accumulative, we measure delta energy consumption over a sampling time period to calculate the average power consumption using the following equation:

$$CPU_{avgpower} = \Delta Energy * frequency \tag{1}$$

D. Python Scripting

A Python script utilizing the Pytorch and Transformers libraries is written to facilitate the testing. It selects a model, picks an option between deploying it on CPUs or GPUs, initiates the power profiling and logging collected data to CSV files at 1Hz sampling rate, then loads the model from the local storage and provides each prompt sequentially, along with two parameters: max_length and temperature, records the output and, finally stops the power recording and terminates the program.

The parameter **max_length** refers to the expected word count for the output and was set to 100 to allow significantly interesting results without taking an excessive amount of time. The parameter **temperature** [26] indicates how "open-ended" the expected answer should be. Temperature 0 is akin to a simple text search, i.e. more common words (or tokens) can be expected, and the same output occurs nearly every time. Temperature 1, however, is much more random, often producing extremely strange results. A temperature of 0.9 is selected for purposes of this study, as this results in more consistency and coherence, while ensuring interesting output (i.e. more human-like language). Each prompt is evaluated by a model three times, creating 30 total entries per run. The output is then written to a file for assigning writing performance scores (ref. Section VI.B).

V. Testing Procedure

After configuring the test environment, the following identical steps are performed to generate the experimental results for analysis:

1) Select a foundation model
2) Choose CPU or GPU deployment
3) Run the Python script with selected options
4) Match timestamps in CSV files with prompts to determine energy usage per prompt

To reduce the error rate and remove outliers, the Python script is executed five times under two separate scenarios:

1) CPU-only model evaluation
2) GPU-accelerated evaluation

This resulted in ten total runs, albeit with some failures. We observe that neither GPT-J 6B nor GPT-Neo 2.7B can be successfully executed on our GPUs because both models are too large to load into the RTX 2080TI GPU memory. In our experiments, we successfully collect 240 total outputs, resulting in approximately 106KB of total text.

VI. Evaluation Metrics and Results

To fully understand the quality and carbon impact of various foundation models at the inference stage, it is essential to comprehensively evaluate them from different key perspectives such as response time, power/energy, output quality, and carbon intensity. In this section, we present our evaluation metrics and results from these important perspectives.

A. Response Time, Power and Energy

Table II shows the average total power used by each output produced by the respective models. The results are broken down as follows:

- Time - The average amount of time in seconds taken for a prompt to be evaluated and an output generated. In the case of GPT-2, for instance, each prompt took roughly 27.5 seconds to be evaluated before the model generated and recorded a reply.
- Power - Average power consumed in watts over the total time each prompt was evaluated until terminating in an output.
- Total Energy - The amount of joules consumed during the entire prompt evaluation process. This is calculated by the formula $Watts * Seconds = Joules$.

From Table II, we can observe that the number of parameters is directly related to the response time and total energy consumed by each model. For example, GPT-J 6B is nearly twice as power-hungry per run than the next closest model (GPT-Neo 2.7B) when deployed in CPUs, using over 6,800 joules of energy to generate a single response. Further, GPT-J 6B takes nearly 18x as long to complete a single output as compared to GPT-Neo 125M. The power consumed by the models is fairly consistent. However, GPT-2 (with 1.5 billion parameters) appears to be an exception by using $\sim$13% more power than the next closest model (GPT-Neo 1.3B). However, GPT-2 takes less time to run thus still saves roughly 12% energy than GPT-Neo 1.3B.

Table III summarizes the results when deploying these models on GPUs, from which we can observe that both the response time and energy efficiency have been significantly increased for all three foundation models. While the CPU consumes between 15%-30% less power, the GPU utilizes a considerable amount, resulting in roughly 25% more total

TABLE II
AVERAGE FOUNDATION MODEL POWER USAGE - CPU ONLY

Model	Time (Seconds)	Power (Watts)	Total Energy (Joules)
GPT-J 6B	139.1	48.9	6802.9
GPT-Neo 2.7B	68.6	51.8	3553.5
GPT-Neo 1.3B	35.7	54.2	1934.9
GPT-Neo 125M	7.8	50.3	392.3
GPT-2	27.5	61.7	1696.8

TABLE III
AVERAGE FOUNDATION MODEL POWER USAGE - GPU ACCELERATED

Model	Time (Seconds)	CPU Power (Watts)	GPU Power (Watts)	Total Energy (Joules)
GPT-Neo 1.3B	2.1	42.6	36.8	166.7
GPT-Neo 125M	1.0	42.9	21.7	64.6
GPT-2	2.4	43.2	30.5	176.9

power on average when compared to the CPU only results. However, the time taken to produce each output is drastically reduced. For example, we see a 17x speedup for GPT-Neo 1.3B, which contributes to 91% overall energy reduction. Contrasting GPT-Neo 1.3B and GPT-2 again, their response time and energy efficiency are fairly comparable when deploying on GPUs. It can also be observed that the smallest model GPT-Neo 125M is 2 times faster and consumes much less energy but the quality of its output is a concern.

B. Writing Performance

As previously stated, we consider the quality of NLP outputs is as important as their energy and carbon efficiency. In fact, in some cases it might be even more important, depending on the circumstances. For example, an AI system utilizing NLP that guides a doctor through a complex medical procedure would need to be much more accurate than an NLP system that does general translation. Thus, a rigid evaluation on model quality is required. According to the Program for the International Assessment of Adult Competencies (PIAAC) in a 2012 study, the average adult in the United States reads at approximately an 8th grade level [27] [28]. As this work is concerned with NLP outputs of a more general nature, we target the 8th reading level as a baseline for output evaluation. In 2012, the Texas Education Agency adopted The State of Texas Assessments of Academic Readiness (STAAR) [8]. As part of this adoption, evaluative measures were made available to all Texas teachers to ensure a broadly applicable model that could assess various skills expected of students within the public school system, including reading and writing. Via STAAR, a rubric [29] and scoring guide [30] are available for English I, a class taken by all freshman-level (9th grade) students in Texas. STAAR rubrics are not available to assess 8th grade reading and writing, hence the selection of this rubric instead.

The rubric is divided into three portions:

- Organization/Progression - the clarity of organization, focus, and responsiveness of the output
- Development of Ideas - the output contains details and examples appropriate to the prompt, and also how engaging and interesting it is to read
- Use of Language/Conventions - precision of word choice, clear, correct grammar, and purposeful sentence structure

TABLE IV
AVERAGE FOUNDATION MODEL WRITING PERFORMANCE - CPU ONLY

Model	Writing Performance Score
GPT-J 6B	5.83
GPT-Neo 2.7B	6.63
GPT-Neo 1.3B	6.17
GPT-Neo 125M	3.57
GPT-2	5.83

TABLE V
AVERAGE FOUNDATION MODEL WRITING PERFORMANCE - GPU ACCELERATED

Model	Writing Performance Score
GPT-Neo 1.3B	5.87
GPT-Neo 125M	3.5
GPT-2	5.83

For each output generated by selected foundation models, the portions of the rubric are evaluated separately and assigned a numeric score between 1 and 4. These three portions are then added together, resulting in a total score between 3 and 12, with 12 being the highest possible score, i.e., the output is an example of superior writing. 3 denotes a nonsensical output, with poor grammar, little acknowledgement of the prompt, and lack of overall coherence. To eliminate as much bias as possible, all outputs are scored randomly, i.e., selected and scored with no indicator as to the model employed.

All 240 outputs are scored using the aforementioned rubric, and the final results are tallied and averaged in tables IV and V for comparisons. The average writing performance for each model is denoted, separated between CPU Only and GPU Accelerated results.

Comparing the CPU Only and GPU Accelerated results, it is immediately apparent that there is no statistical significance between the quality of output generated by either method. Thus, when possible, it would be best practice to always use the GPU if speed and power savings are valued. However, a very significant difference can be seen in the score differential amongst the more power-hungry models. As shown in table II, GPT-J 6B uses vastly more energy during the production of each output, but performs roughly on par with GPT-Neo 2.7B, GPT-Neo 1.3B, and GPT-2. GPT-Neo 2.7B has the highest writing performance score (12% better than GPT-J 6B) but consumes only 52% of power. There could be many reasons for this disparity, including the quality of the prompt selected, researcher bias when reading the outputs, or pure randomness. It is entirely possible that some models are simply better at

certain sorts of outputs than others.

Interestingly, GPT-Neo 1.3B and GPT-2 are very similar in writing performance, which aligns closely with their very similar energy consumption. When comparing these two models to GPT-Neo 2.7B, there is roughly a 2X energy increase for the at-best aforementioned 12% better score, and thus an important decision must be made: how valuable is that 12% increase in quality?

Finally, it must be mentioned that judging the actual quality of output from the various foundation models revealed the difficulty in reproducing natural language within a synthetic environment. Even though strict rules for English exist, there are countless instances where those rules might be broken for sake of clarity, emphasis, or even stylistic choice. What separates good English from bad English might be a matter of taste, and thus the criteria for tailoring output changes from domain to domain. Indeed, the mere 12% increase in quality of output might be extremely significant if an expert system is needed vs. an online chatbot that can be expected to spout gibberish in some instances.

C. Software Carbon Intensity

We leverage the Software Carbon Intensity (SCI) [9] specification to evaluate the carbon intensity of foundation models at the inference stage. SCI is a metric released by the Green Software Foundation in 2021 and it describes a methodology for calculating the rate of carbon emissions for a software system. Its purpose is to inform users and developers about the possible carbon impact of their software, services, and architectures, and assist them with making better choices in green software design. Per the specification, SCI is a rate of carbon emissions per one unit of R, and is represented by the following equation:

$$SCI = ((E * I) + M) \text{ per } R \tag{2}$$

Where:

- E = Energy consumed by a software system
- I = Location-based marginal carbon emissions
- M = Embodied emissions of a software system
- R = Functional unit (e.g. carbon per additional user, API-call, ML job, etc.)

R in this case refers to each individual run of the AI, i.e. evaluating a single prompt and producing an output. E per each R has already been calculated in tables II and III. As the test server is physically located in San Marcos, TX, power is supplied by Electric Reliability Council of Texas (ERCOT). Per the 2019 Grid Electricity Emissions Factors report, systems utilizing the ERCOT power grid generate 0.4784 $kgCO_2e$ per kWh [31], and is assigned as the value of I. M is calculated as follows:

$$M = TE * (TR/EL) * (RR/TR) \tag{3}$$

Where:

- TE = Total Embodied Emissions, the sum of LCA emissions for all hardware components

TABLE VI
CALCULATED SCI FOR ALL FOUNDATION MODELS

	GPT-J 6B	GPT-Neo 2.7B	GPT-Neo1.3B	GPT-Neo 125M	GPT-2
Total kWh - CPU	1285.26	1361.31	1424.35	1321.75	722.72
Total Runs - CPU	680,143	1,379,125	2,650,084	12,129,230	3,440,290
Total Emissions - CPU	5552.87	5589.25	5619.41	5570.32	5283.75
SCI - CPU	8.16	4.05	2.12	0.46	3.45
Total kWh - GPU	-	-	2086.13	1697.69	1937.06
Total Runs - GPU	-	-	45,051,428	94,608,000	39,420,000
Total Emissions - GPU	-	-	5936.01	5750.17	5864.69
SCI - GPU	-	-	0.13	0.06	0.15

- TR = Time Reserved, the length of time the hardware is reserved for use by the software
- EL = Expected Lifespan, the anticipated time that the equipment will be installed
- RR = Resources Reserved, the number of resources reserved for use by the software
- TR = Total Resources, the total number of resources available

TE is somewhat difficult to calculate, and thus we use the Boavizta dataset to estimate the TE for the test server utilized in these experiments [32] [33]. Per AMD, the Ryzen 2950x processor has a TDP of 180W [34]. The closest system in the Boavizta database with a similar processor TDP has a value of is a Dell PowerEdge R540 with a processor TDP of 165W. As the PowerEdge is a very similar high-end system, this is a good sample for approximating embodied emissions. gwp_total for the PowerEdge is 8230 $kgCO_2e$, and thus this value is used for TE in these calculations. Due to the lack of information available on carbon emissions during the GPU manufacturing process, TE (and subsequently SCI) is calculated under the assumption that the server only utilized the CPU when running models.

TR is selected as three years due to the length of actual usefulness of the system; this is a typical length of time a system of this sort is deployed before replacement is necessary to keep up with advancements in technology. EL, however, is set at five years as this is the average length of time a server is expected to be deployed before replacement. Essentially it is assumed that although a system of this sort can be expected to perform without hardware failure for five years, after three years the hardware will be insufficient to run contemporary jobs.

RR and TR are both set to 1. Here we assume most of the system resources will be utilized when executing large tasks submitted to these foundation models. Using the values above, the value of M would be calculated as:

$$M = 8230 kgCO_2e * (3/5) * (1/1) = 4938 kgCO_2e \tag{4}$$

For purposes of this calculation, joules of energy must be converted to kWh, with 3,600,000 joules equal to one kWh. Using GPT-J 6B, the total energy per model run would thus be 6802.9/3600000 = 0.00189 kWh. Further, we assume the foundation model service will be available for the entire

lifespan of the server (3 years) and thus we convert the time for each run into the total number of AI model runs over that time span: 3 years * 31,536,000 seconds/years / 139.1 seconds = 680,143.78 total runs. Hence, the total amount of energy utilized by GPT-J 6B over the lifespan of the server would be 680,143.78 * 0.00189 kWh = 1285.47 kWh. SCI for GPT-J 6B is then calculated as:

$$TotalEmissions = (1285.47kWh \times 0.4784kgCO_2e/kWh) + 4938kgCO_2e = 5552.97kgCO_2e \quad (5)$$

SCI is per run of each instance of the GPT-J 6B model and would be calculated simply as 5552.97/680,143.78 = 0.0082 $kgCO_2e$, or 8.2 gCO_2e.

We calculate the SCI of all foundation models and list the results in VI. It is apparent that the carbon expenditure during creation of the server itself has a massive impact on the overall SCI value. This indicates that selecting the device to run a foundation model is extremely important when considering carbon waste. Further, while GPT-J 6B is the most power hungry model for each run, the fact that the smaller GPT-Neo 2.7B and 125M models can run significantly more times during the server deployment time, which means those models could actually use more power and produce more total carbon waste if run continuously. Another significant finding is that using the GPU produces a much lower carbon impact from the ability to run the model many more times. SCI is substantially lower, but overall power usage is still higher.

VII. Conclusions and Future Work

This project aims to tackle the problem of evaluating the carbon impact of complex foundation models. We conduct a comprehensive study in exploring how to evaluate the quality and the carbon impact of foundation models at the inference stage. More specifically, we investigate several open-source foundation models, including GPT-J 6B [4], GPT-Neo 2.7B, GPT-Neo 1.3B, GPT-Neo 125M [5], and GPT-2 [6] and focus on evaluating their energy efficiency and carbon emissions at the inference stage. The following tentative conclusions can be made from this preliminary study:

The environmental impact of foundation models can be quantitatively measured and compared.

Per table VI, the SCI of each model can be calculated and compared to other models. GPT-J 6B, for instance, produces nearly 100% more carbon emissions than GPT-Neo 2.7B each time the models produce an output, demonstrating a significant difference in environmental impact. However, it must be noted that, as stated previously, employing GPT-J 6B may be a valid choice over other models if its value for a particular deployment outweighs its negative environment effects. SCI is thus a useful metric, but not an absolute indicator, of the validity of using a specific foundation model.

The quality of a foundation model can be found by objectively analyzing the output.

If an objective standard can be employed to evaluate outputs, the quality of a foundation model can be determined. In this study, we demonstrate how to use the STTAR rubric to generate objective Writing Performance Scores for the outputs generated by different foundation models. Most of the selected foundation models produce similar quality of output that is marginally above a mediocre 9th grade English student, albeit GPT-Neo 125M performs exceptionally poorly. This in itself provides some useful insights in ways to improve the models, perhaps by further analyzing the outputs and tailoring the model to target specific weak spots in the output generation. Applicability to other domains should be apparent. For instance, showing AI generated faces to graduate students and asking them to determine which face looks "more human" would be an example of a metric that could be employed to evaluate an image processing model. The students might be asked to score each face between 1 (a crude circle with dots) to 10 (appears to be an original photograph). Ultimately, whatever metric is chosen to score an AI model, it is still the responsibility of model developers to decide if the output matches expectations. They must decide which outputs to put forth for judgement and, in fact, what sorts of outputs to generate initially.

It is possible for a foundation model to be replaced by a more efficient model, mitigating energy requirements and reducing carbon waste while maintaining a similar level of expected output.

This may require more insights such as a cost-benefit analysis, but given the results of the study it does appear some models can indeed be migrated to a different model, depending on expected output. Specifically, if an NLP model is required to produce an output with a Writing Performance Score of 6, four of the five models evaluated would be viable candidates for such a task. While GPT-Neo 2.7B would be more attractive due to its higher score, if the need only requires a score of 6, the best option would be the much more efficient GPT-Neo 1.3B. In this particular case, the models both provide the expected level of quality, while GPT-Neo 1.3B is much cheaper to run, or about 27% the energy cost of GPT-J 6B. There might be instances where GPT-Neo 2.7B would be a better choice, such as in specific situations where the outputs for a particular type of language generation score are much higher than average. In this case the user making such a selection would need to make a choice based on a finer granularity than as determined by the initial evaluations provided herein. If minimizing carbon impact is a primary concern, SCI provides a concrete way to compare foundation models and easily determine which outputs are "greener".

There is no significant difference in the quality of output of foundation models when using the CPU-only or GPU-acceleration.

Referring to tables IV and V, the Writing Performance Score of an output is not significantly affected when only using the CPU or enabling GPU-acceleration. Further, up to 91% of energy is saved using the GPU over the CPU-only. Thus, employing GPUs is demonstrably a better practice.

Embodied carbon plays a dominating role in carbon intensity. Embodied carbon accounts for a large portion of

total carbon emissions. An effective way to minimize its impact on carbon intensity is to extend the lifetime of servers or increase the number of requests the server services.

It is worth noting that there are many limitations in this preliminary study, which need to be addressed in future work. First, we only select ten prompts to evaluate model quality, lump all the outputs together, and provide an overall average score. Broadening the scope of prompts and scoring each model according to specific types of prompts (informative, scientific, expository) could yield better approaches to evaluate model quality. However, given the enormity of time taken to perform this initial study, it should be evident that evaluating AI models can be as resource-hungry as the initial training process and is no light undertaking.

Another area for further research would be in evaluating GPT-J 6B and GPT-Neo 2.7B using GPU acceleration. It is speculated that similar performance increases and power mitigation would occur, as with the other models, but hard data would be useful.

It would also be of interest to run a wider variety of prompt evaluations, tailoring temperature and max_length to each prompt type. We expect that each prompt type would lend itself well to more comprehensive studies, perhaps one evaluating NLP implementations based on their ability to produce coherent scientific answers, and another that evaluates synthetic poetry.

The other limitation of this study is the lack of embodied carbon information for the testing server and real user usage data of the foundation models we test. To proceed with the study, we made several assumptions based on the best of our knowledge and the closest datasets we can find available. The accurate embodied carbon data and real user information will certainly help improve the quality of our results. We hope more of such information will be available to researchers for future studies.

REFERENCES

[1] "AI and Compute," https://openai.com/blog/ai-and-compute/, 2018.

[2] K. Quach, "Ai me to the moon...carbon footprint for training gpt-3 same as driving to our natural satellite and back," https://www.theregister.com/2020/11/04/gpt3_carbon_footprint_estimate/, 2020.

[3] R. Bommasani, D. A. Hudson *et al.*, "On the opportunities and risks of foundation models," 2021.

[4] B. Wang and A. Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," https://github.com/kingoflolz/mesh-transformer-jax, May 2021.

[5] S. Black, G. Leo *et al.*, "GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow," Mar. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5297715

[6] A. Radford, J. Wu *et al.*, "Language models are unsupervised multitask learners," 2019.

[7] EleutherAI, "Eleutherai hugging face repository," https://huggingface.co/EleutherAI, 2021.

[8] T. E. Agency, "Staar english i and english ii resources," https://tea.texas.gov/student-assessment/testing/staar/staar-english-i-and-english-ii-resources, 2021.

[9] G. S. Foundation, "Software carbon intensity (sci) specification," https://github.com/Green-Software-Foundation/software_carbon_intensity/blob/main/Software_Carbon_Intensity/Software_Carbon_Intensity_Specification.md, 2021.

[10] D. Patterson, J. Gonzalez *et al.*, "Carbon emissions and large neural network training," 2021.

[11] L. F. W. Anthony, B. Kanding, and R. Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," 2020.

[12] W. Gao, J. Zhan *et al.*, "Data motifs: A lens towards fully understanding big data and ai workloads," in *Proceedings of the 27th International Conference on Parallel Architectures and Compilation Techniques*, ser. PACT '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3243176.3243190

[13] M. Labbe, "Energy consumption of ai poses environmental problems," https://www.techtarget.com/searchenterpriseai/feature/Energy-consumption-of-AI-poses-environmental-problems, 2021.

[14] R. Toews, "Deep learning's carbon emissions problem," https://www.forbes.com/sites/robtoews/2020/06/17/deep-learnings-climate-change-problem/, 2021.

[15] S. DeWeerdt, "It's time to talk about the carbon footprint of artificial intelligence," https://www.anthropocenemagazine.org/2020/11/time-to-talk-about-carbon-footprint-artificial-intelligence/, 2020.

[16] L. Aung, "Deep blue: The history and engineering behind computer chess," https://illumin.usc.edu/deep-blue-the-history-and-engineering-behind-computer-chess/, 2010.

[17] Shreyak, "Text generation using gpt3," https://becominghuman.ai/text-generation-using-gpt3-781429c4169, 2021.

[18] L. Gao, S. Biderman *et al.*, "The Pile: An 800gb dataset of diverse text for language modeling," *arXiv preprint arXiv:2101.00027*, 2020.

[19] A. R. Naik, "Eleutherai's gpt-j vs openai's gpt-3," https://analyticsindiamag.com/eleutherais-gpt-j-vs-openais-gpt-3/, 2021.

[20] Huggingface, "Hugging face - gpt2," https://huggingface.co/gpt2, 2021.

[21] J. Weizenbaum, "Eliza-a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, jan 1966. [Online]. Available: https://doi.org/10.1145/365153.365168

[22] Pytorch, "Pytorch," https://pytorch.org/, 2021.

[23] HuggingFace, "Huggingface transformers," https://huggingface.co/docs/transformers/index, 2021.

[24] H. K. Kushwaha, "Running python script on gpu," https://www.geeksforgeeks.org/running-python-script-on-gpu/, 2021.

[25] NVIDA, "Cuda zone," https://developer.nvidia.com/cuda-zone, 2021.

[26] J. A. Kolar, "A simple guide to setting the gpt-3 temperature," https://algowriting.medium.com/gpt-3-temperature-setting-101-41200ff0d0be, 2020.

[27] M. G. (ETS), R. F. (ETS) *et al.*, "Literacy, numeracy, and problem solving in technology-rich environments among u.s. adults: Results from the program for the international assessment of adult competencies 2012," https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=2014008, 2012.

[28] A. Wylie, "What's the latest u.s. literacy rate?" https://www.wyliecomm.com/2021/08/whats-the-latest-u-s-literacy-rate/, 2021.

[29] T. E. Agency, "English i expository writing rubric," https://tea.texas.gov/sites/default/files/Rubric-EOC-Eng1-WrtgExpository.pdf, 2011.

[30] T. E. Agency, "English i expository scoring guide," https://tea.texas.gov/sites/default/files/2021-staar-english-1-scoring%20guide-tagged.pdf, 2021.

[31] Carbonfootprint.com, "2019 grid electricity emissions factors," https://www.carbonfootprint.com/docs/2019_06_emissions_factors_sources_for_2019_electricity.pdf, 2019.

[32] Boavizta, "Digital and environment: How to evaluate server manufacturing footprint, beyond greenhouse gas emissions?" https://www.boavizta.org/en/blog/empreinte-de-la-fabrication-d-un-serveur, 2021.

[33] Boavizta, "Environmental footprint data," https://github.com/Boavizta/environmental-footprint-data/blob/main/boavizta-data-us.csv, 2022.

[34] AMD, "Amd ryzen threadripper 2950x processor," https://www.amd.com/en/products/cpu/amd-ryzen-threadripper-2950x, 2022.