Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model

Alexandra Sasha Luccioni, Sylvain Viguier, Anne-Laure Ligozat

Journal of Machine Learning Research 24 (2023) 1-15 Submitted 1/23; Revised 5/23; Published 6/23

Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model

Alexandra Sasha Luccioni SASHA.LUCCIONI@HUGGINGFACE.CO
Hugging Face
Montréal, Canada

Sylvain Viguier SYLVAINV@GRAPHCORE.AI
Graphcore
London, UK

Anne-Laure Ligozat ANNE-LAURE.LIGOZAT@LISN.UPSACLAY.FR
LISN & ENSIIE
Paris, France

Editor: Shakir Mohamed

Abstract

Progress in machine learning (ML) comes with a cost to the environment, given that training ML models requires computational resources, energy and materials. In the present article, we aim to quantify the carbon footprint of BLOOM, a 176-billion parameter language model, across its life cycle. We estimate that BLOOM's final training emitted approximately 24.7 tonnes of CO_2eq if we consider only the dynamic power consumption, and 50.5 tonnes if we account for all processes ranging from equipment manufacturing to energy-based operational consumption. We also carry out an empirical study to measure the energy requirements and carbon emissions of its deployment for inference via an API endpoint receiving user queries in real-time. We conclude with a discussion regarding the difficulty of precisely estimating the carbon footprint of ML models and future research directions that can contribute towards improving carbon emissions reporting.

Keywords: carbon footprint, language modeling, life cycle assessment, machine learning.

1. Introduction

Climate change is one of our generation's biggest challenges, impacting ecosystems and livelihoods across the world; estimating and reducing our carbon emissions is an important part of mitigating its impacts (Masson-Delmotte et al., 2018). According to recent estimates, the global CO_2 emissions of the information and communications technology (ICT) sector account for around 2% of global CO_2 emissions, but this figure is hard to estimate precisely given the distributed nature of global computing infrastructure (International Telecommunication Union, 2020; Malmodin and Lundén, 2018; Copenhagen Centre on Energy Efficiency, 2020). The infrastructure used for training and deploying machine learning (ML) models contributes to this number, but the exact extent of this contribution is also unclear. In order to get a better grasp of the carbon footprint of the field, it is

therefore important to start systematically tracking the carbon footprint of ML models and algorithms and the main sources of emissions.

Large language models (LLMs) are among the biggest ML models, spanning up to hundreds of billions of parameters, requiring millions of GPU hours to train, and emitting carbon in the process. As these models grow in size – which has been the trend in recent years – it is crucial to understand to also track the scope and evolution of their carbon footprint. The current study describes the first attempt to estimate the broader carbon footprint of an LLM, including the emissions produced by manufacturing the computing equipment used for its training as well as the model deployment via an API. The goal of our study is not to hone in on an exact number for the emissions produced, but to provide estimates of the relative contribution of each step of the deployment process towards the overall emissions of the model. We conclude with a discussion about the carbon emissions of different LLMs as well as the BigScience workshop overall, and propose directions for future work to both quantify and report these emissions.

2. Related Work

There are different aspects of the environmental impact of computing in general and machine learning in particular that are relevant to our study; we briefly describe existing relevant work in the paragraphs below.

Empirical Studies on ML CO_2 Emissions Most of the existing work in this area has been done on estimating the CO_2 emissions incurred during model training. Starting with the seminal work of Strubell et al., who looked at the carbon footprint of training a Transformer model (2019), more recent studies have also looked at other model architectures and their ensuing emissions (Patterson et al., 2021; Naidu et al., 2021). Other studies have pursued a broader analysis of trends in terms of the energy requirements and CO_2 emissions of ML models in general (Thompson et al., 2020; Wu et al., 2021; Patterson et al., 2022). While some studies predict a growth in terms of carbon emissions of ML models (Thompson et al., 2020), others have predicted that emissions will shrink in coming years (Patterson et al., 2022); further work is therefore needed to get additional estimates from a broader variety of models and use cases.

Tools for Estimating Carbon Impact Another relevant research direction has pursued the development of tools for estimating the CO_2 emissions of training ML models, resulting in several tools created for this purpose. Some of these run in parallel to model training code and track its energy consumption and CO_2 emissions (e.g. Schmidt et al. (2021); Anthony et al. (2020)), while others can be used post-training in order to produce a more high-level estimate of emissions (e.g. Lacoste et al. (2019)). However, these tools remain seldom used for reporting the CO_2 emissions in ML publications, and a recent study has found that they vary significantly in terms of the estimates that they produce (Bannour et al., 2021).

Additional Factors Complementary work has also been done on other contributions to the overall carbon footprint of ML, ranging from the carbon footprint of in-person versus virtual conference attendance (Skiles et al., 2021) to the manufacturing of computing hardware (Gupta et al., 2021) as well as the life cycle analysis of the entire ML development and deployment cycle (Ligozat et al., 2021) and the certification of ML systems according to their social and environmental impacts (Gupta et al., 2020). Increasingly, scholars have adopted a broader perspective on considering the environmental impacts of ML models, going above and beyond only the CO_2 emissions of model training and

considering aspects such as equipment manufacturing and deployment (Wu et al., 2021; Kaack et al., 2022). However, there is still a need for a common approach in terms of estimating and comparing the carbon emissions of ML models which spans these different parts of the model life cycle.

3. Background and Methodology

3.1 The BLOOM Model

The BigScience Large Open-science Open-access Multilingual Language Model (BLOOM) is a 176 billion parameter language model. It was trained on 1.6 terabytes of data in 46 natural languages and 13 programming languages as part of the BigScience workshop, a year-long initiative that lasted from May 2021 to May 2022 and brought together over a thousand researchers from around the world. The BigScience workshop was granted access to the computing resources of the Institut du développement et des ressources en informatique scientifique (IDRIS) of the Centre national de la recherche scientifique (CNRS) in France, which meant that model training was carried out on the Jean Zay computer cluster of IDRIS. We present some key numbers about BLOOM model training in Table 1 below, and refer readers to Le Scao et al. (2022) for additional information about model architecture and training.

Total training time	118 days, 5 hours, 41 min
Total number of GPU hours	1,082,990 hours
Total energy used	433,196 kWh
GPU models used	Nvidia A100 80GB
Carbon intensity of the energy grid	57 gCO_2eq/kWh

Table 1: Key statistics about BLOOM model training – for more details about our methodology, see Section 4.2.

While training the model was the culmination of the BigScience project, many other efforts were needed to achieve this goal. This includes initiatives such as: data sourcing, collection and processing, tokenization, architecture engineering and evaluation. Additionally, in the months preceding the final BLOOM training, several smaller-scale experiments were launched in order to evaluate different model sizes and architectures, which helped converge on the final BLOOM architecture. In the results that presented in Section 4, we report the carbon emissions produced by the final 176B parameter BLOOM model, whereas the emissions of intermediate model training and evaluation carried out within the scope of the BigScience project are presented in Section 5.2.

3.2 Methodology

While there is no universally-accepted approach for assessing the environmental impacts of ML models, we strive towards adopting the widely-used Life Cycle Assessment (LCA) methodology, which aims to cover all stages of the life cycle of a product or process (Klöpffer, 1997). While we do not have all of the necessary information to carry out a "cradle-to-grave" assessment of BLOOM

(which would consider the environmental impacts of all processes from raw material extraction to disposal), we focus on the steps for which we do have sufficient information, which range from manufacturing the equipment used for training the model to model deployment (see Fig 1). In fact, recent work has proposed a more specific framework for categorizing ML's effects on greenhouse gas (GHG) emissions, consisting of 3 categories: (A) computing-related impacts, (B) immediate impacts of deploying ML and (C) system-level impacts on other domains (Kaack et al., 2022). While this framework has not yet been widely adopted in our field, we believe it is particularly useful given the specificity of the ML life cycle. In the current study, we focus on category (A) and briefly discuss deployment and system-level impacts in Section 5.3.

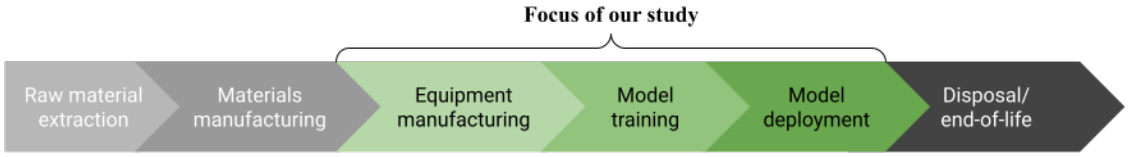

Figure 1: While the LCA approach encompasses all the stages of the product life cycle, we focus on those stages indicated in green, which range from equipment manufacturing to model deployment.

Given that the LCA approach aims to account for all possible sources of GHG emissions (e.g. methane, carbon dioxide, nitrous oxide, etc.), it is necessary to convert these different gasses to a single unit of measure in order to meaningfully sum them up. The standardized measure that is often used for this is *carbon dioxide equivalents* (CO_2eq), which are calculated based on comparing the global-warming potential (GWP) of different greenhouse gasses to that of carbon dioxide (CO_2). For instance, methane has a 100-year GWP 25 times that of CO_2– this means that one tonne of methane is equal to 25 tonnes of CO_2eq. In the sections below, we be using this unit of measure to estimate BLOOM's emissions throughout its life cycle.

4. Results: Carbon Emissions of the BLOOM Model

Our study aims to bring together the different elements contributing to the overall carbon footprint of training BLOOM and to compare the relative contribution of each one towards BLOOM's total emissions. While we will predominantly focus on model training, we will also take into account the emissions produced by manufacturing the computing equipment used for running the training, the energy-based operational emissions, as well as the carbon footprint of model deployment and inference. All of the code and data used for our analyses are available in our Github repository.

4.1 Embodied Emissions

The *embodied emissions* are those emissions associated with the materials and processes involved in producing a given product, such as the computing equipment needed to train and deploy ML models. While the production of these emissions is exclusively limited to the manufacturing process, this total amount is usually spread over the time during which equipment is used by dividing the total embodied emissions by the time of use. The BLOOM model was trained on HPE Apollo 6500 Gen10 Plus servers containing Nvidia A100 GPUs. The closest comparable computing equipment that provides LCA information is HPE's ProLiant DL345 Gen10 Plus server, which is similar to the Apollo 6500

and has a production footprint of approximately 2500 kg of CO_2eq (HPE, 2021). This does not include the embodied emissions of the GPUs which are used in the server, whose embodied emissions must be calculated separately. While Nvidia does not currently disclose the carbon footprint of its GPUs, recent estimates put the lower bound of this amount at approximately 150 kg of CO_2eq (Davy, 2021), which is the number we will use for our embodied emissions estimates.

Assuming a replacement rate of 6 years and 85% average usage (which are the figures provided to us by IDRIS), the figures above translate to an embodied carbon footprint of approximately 0.056 kg of CO_2eq for each hour of server time and 0.003 kg of CO_2eq for each hour of GPU time. Given that BLOOM training lasted a total of 1.08 million hours using, on average, 384 GPUs across 48 computing nodes, we can estimate that the embodied emissions associated to BLOOM training represent approximately 7.57 tonnes for the servers and 3.64 tonnes for the GPUs, adding a total of 11.2 tonnes of CO_2eq to its carbon footprint. This does not include the embodied emissions of the rest of the computing infrastructure (e.g. the network switches, cooling equipment and other devices that power the network), which are difficult to quantify given that we do not have the necessary information regarding their distribution and usage.

4.2 Dynamic Power Consumption

As described in Section 2, most of the existing research on the carbon footprint estimation of ML models has focused on estimating the CO_2 emissions produced by generating the electricity necessary for powering model training – this is typically referred to as *dynamic consumption*. This is typically calculated by multiplying the number of GPU hours used by the thermal design power (TDP) of those GPUs and the carbon intensity of the energy grid used to power the hardware. TDP remains an upper bound of GPU power consumption, but it is often used as a proxy given when access to real-time GPU power consumption is impossible. A grid's carbon intensity depends on the electricity source that powers it – for instance, coal-powered grids result in more carbon emissions per kWh of electricity compared to grids powered by hydroelectricity or solar power. Also, while many compute providers carry out post hoc carbon offsetting or heat recycling, we do not take this into account in our estimation, given that we are focusing on the direct carbon emissions linked to dynamic power consumption [1].

As reported in Table 1, training the BLOOM model required a total of 1.08 million GPU hours on a hardware partition constituted of Nvidia A100 SXM4 GPUs with 80GB of memory, which have a TDP of 400W (NVIDIA, 2022). While we were not able to track real-time power consumption for the entire experiment, empirical observations noted that GPU utilization was typically very high, nearing 100%. Also, we do not consider the power usage of CPUs, which consume approximately 40 times less energy than GPUs and which are typically not as solicited during the model training process. This represents an electrical consumption of 433,196 kWh of electricity during training; multiplied by the carbon intensity of the energy grid used, which is approximately 57 gCO_2eq/kWh (Aurora Energy Research, 2020), this results in a total of 24.69 tonnes of CO_2eq emitted due to dynamic energy consumption.

1. In fact, a percentage of the heat produced by the Jean Zay computing cluster is recuperated to supply the heat and cold exchange network of the Paris-Saclay urban campus.

4.3 Idle Power Consumption

So far, the emphasis in the ML community has been on estimating the energy consumption and ensuing carbon emissions of the energy used to power specialized hardware such as GPUs. However, it is important to keep in mind that the broader infrastructure that maintains and connects this hardware also requires large amounts of energy to power it – this is referred to as *idle consumption*. The quantity of energy needed for this depends on the efficiency of the computing cluster that is being used and the configuration of the devices on the cluster. Taking into account the idle power consumption during the model training time is important to estimate the overhead costs that are added on top of the energy consumed by the hardware itself. This can be reflected in part by factoring in the PUE (Power Usage Effectiveness) of the data centers used for training these models, which is the approach adopted by Patterson et al. for estimating the carbon emissions of ML models such as T5 and GPT-3 (2021). However, the way in which PUE is calculated (dividing the total amount of energy used by a computer data center facility by the energy delivered to computing equipment) does not account for the totality of energy consumed by the data center infrastructure, leaving out certain key aspects of idle consumption that a more fine-grained analysis encompasses (Brady et al., 2013; Kurpicz et al., 2018), which is why we adopt a different method, based on empirical measurement.

Computing Mode	**Power consumption**	**Percentage of total**
Infrastructure consumption	27 kW	13.5%
Idle consumption	64 kW	32%
Dynamic consumption	109 kW	54.5%
Total consumption	**200 kW**	**100%**

Table 2: Breakdown of power consumption of the A100 partition of the Jean Zay cluster. *Infrastructure mode* measures the power consumed by networking systems, datacenter maintenance and cooling systems (i.e., servers are turned off). *Idle* measures the additional power consumed by servers turned on but unused. *Dynamic* measures the additional power consumed by servers actively training BLOOM.

In order to estimate the idle consumption of the computing infrastructure that we used for training BLOOM, we ran a series of experiments to compare the total energy consumption of idle devices on the Jean Zay computing cluster (e.g. network, GPUs, storage, cooling/heating and computation nodes) to the total consumption of the same devices while running the model training code. As we show in Table 2, we found that for the A100 partition of the cluster used for training the model, in *Infrastructure mode* (with the computing nodes turned off but the network, storage and cooling turned on), the power consumption was 27 kW; in *Idle mode* (with network, storage and compute nodes on, but no processes running), the additional power consumption was 64 kW. During BLOOM training, the dynamic power consumption averaged at over 109 kW, for total of 200kW for the whole system. This indicates that only around 54% of the power consumption can be attributed to running the code (i.e. the dynamic power consumption described in Section 4.2), whereas the remaining 46% is used for keeping the computing nodes on. Extrapolating this to the total model training time, this adds a further 256,646 kWh of idle power consumption on top of the dynamic power used for training BLOOM, and 14.6 tonnes of CO_2eq to the overall carbon footprint of model training.

Process	CO_2emissions (CO_2eq)	Percentage of total emissions
Embodied emissions	11.2 tonnes	22.2 %
Dynamic consumption	24.69 tonnes	48.9 %
Idle consumption	14.6 tonnes	28.9 %
Total	**50.5 tonnes**	**100.00%**

Table 3: Breakdown of CO_2 emissions from different sources of the BLOOM model life cycle

While it may seem excessive to add such a large overhead to BLOOM's carbon footprint, taking embodied and idle emissions into account is a much better reflection of the true emissions of model training than solely considering the dynamic consumption of GPUs, as it also considers the network overhead and larger computing infrastructure without which training cannot take place. The figures from Table 3 are similar to those provided in product carbon footprint estimations for computing equipment (such as the one for the HPE servers used in Section 4.1), which estimate that the embodied emissions account for approximately 20-30% of life cycle emissions, whereas use (i.e. the emissions of both dynamic and idle consumption) are 70-80% of the total footprint. However, our estimations thus far have only been limited to BLOOM training – in the following section, we aim to go further by doing an case study analysis of the energy consumption and ensuing carbon emissions of model deployment.

4.4 Deployment and Inference

In order to attempt to estimate the carbon emissions incurred by deploying BLOOM, we ran the CodeCarbon package (Schmidt et al., 2021) on a Google Cloud Platform (GCP) instance with 16 Nvidia A100 40GB GPUs, where BLOOM was deployed via an inference API, and tracked the energy usage of the instance over a period of approximately 18 days. The model received an average of 558 requests per hour, which were handled in real time (i.e. without any batching), for 230,768 requests in total. While this is not necessarily representative of all deployment use cases, it is an example of real-time deployment of LLMs in applications such as chatbots, where they are expected to respond to a constant, varying flux of user queries. It also provides a useful data point for starting to measure the carbon emissions of ML model inference, which has not been the focus of much research to date.

As it can be seen in Figure 2, during the 18 day period for which we carried out our analysis, the power consumed by the compute instance running the BLOOM model fluctuated between 1252 W to 2735 W – divided by the 16 GPUs that were used, this amounts to 78-171W per GPU, which is significantly less than the TDP of this type of GPUs (400W). This indicates that the GPUs are not being used at maximum capacity, which is coherent with the nature of API deployment – since inference requests are unpredictable, optimization of GPU memory using techniques such as batching and padding, which are the norm during training, is not possible, and the GPUs remain idle in between user requests.

In total, the instance used for the BLOOM model API consumed 914 kWh of electricity – of this amount, 22.7% was consumed by the RAM (207.2 kWh), 2% by the CPU (18.5 kWh) and 75.3% by the GPU (688.38 kWh). It is hard to disaggregate this number into idle versus dynamic consumption

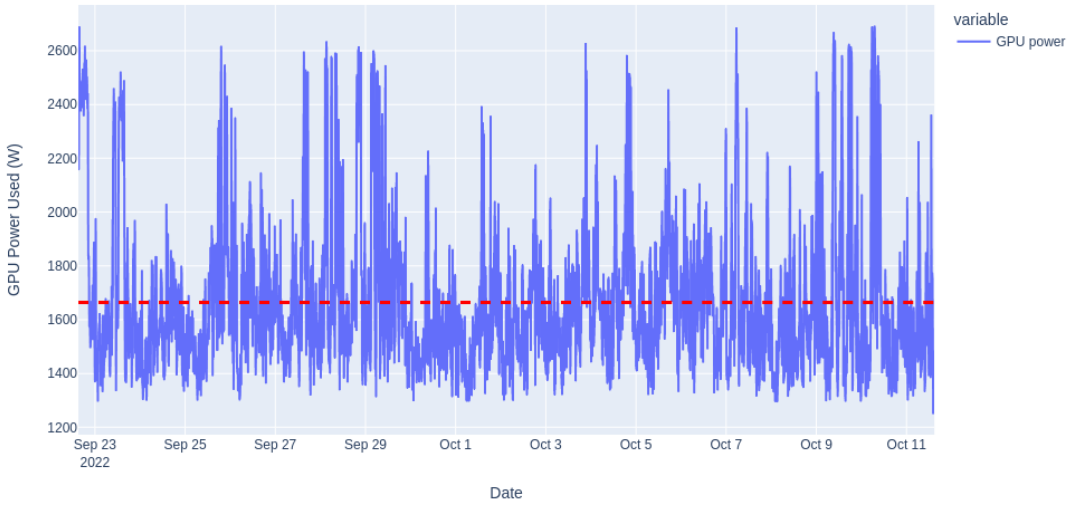

Figure 2: The fluctuation of mean power used to power the 16 Nvidia A100 GPUs running the BLOOM model API, with the mean power consumption in red (1664W) in dotted red.

because we do not have access to the GCP platform as we did for Jean Zay, but we can nonetheless compare the energy consumed by the instance versus the number of requests that it received. We do so in Figure 3, where we plot the number of incoming requests to the BLOOM inference API and the energy consumption of the GCP instance where it is running. It can be seen in the Figure that even when there are almost no incoming requests during a 10 minute interval, there is still ∼0.28kWh of energy that is consumed during this interval, which represents the energy consumption of the model when it is not responding to any user requests. While more experimentation is needed in order to further disaggregate these numbers, we believe it is worth noting the high proportion of energy dedicated to maintaining a LLM like BLOOM in memory (approximately 75% of the total energy consumed by the instance), without it being used. Going further, given that the GCP instance used for deploying the BLOOM model is running in the `us-central1` region, which has a carbon intensity of 394 gCO_2eq/kWh (Google, 2022), this results in approximately 19 kgs of CO_2eq emitted per day of API deployment, or 340 kg over the total period during which we were tracking emissions.

Given the many different combinations of configurations that can be used for deploying ML models, ranging from the hardware used for deployment to the batch size of inferences and the region where the model is running, the use case that we describe above is one among many. However, it is a useful starting point to estimate the carbon emissions involved in deploying ML models, which are lacking in the field. While a 2019 article estimated that 80–90% of Nvidia's ML workload is inference processing (Leopold, 2019) – a figure that was cited in a recent article by Patterson et al (2021), a recent publication by Meta reported that inference accounted for approximately one-third of their end-to-end ML carbon footprint while the remainder is produced by data management, storage, and training (Wu et al., 2021). We hope that the rough estimates we provide above shed some light on this question and plan on pursuing this avenue of research further in our future research endeavors, which we discuss in more detail in Section 5.3.

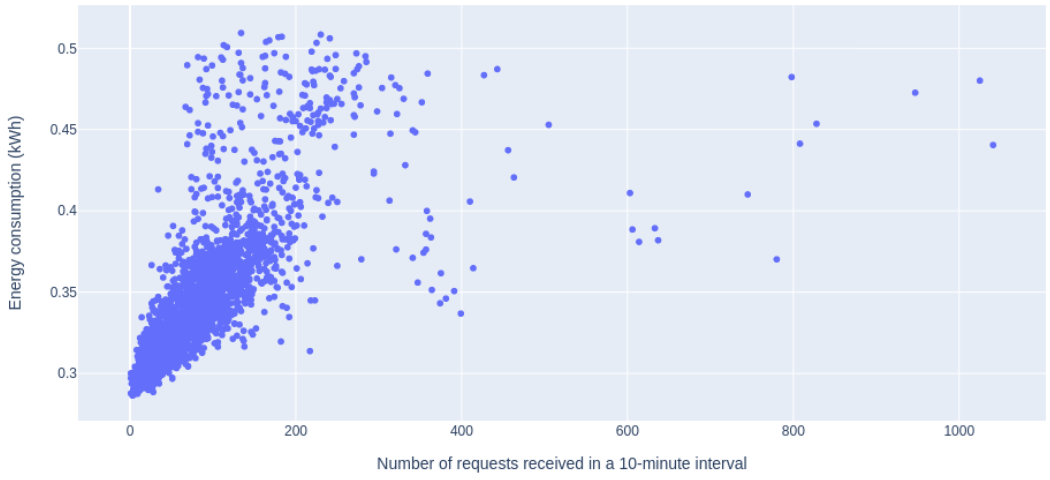

Figure 3: The quantity of energy used by the GCP instance (on the y axis) versus the number of requests received by the instance in a 10 minute interval (on the x axis). It can be seen that even when zero requests are received by the instance in this time span (bottom left of the graph), the energy consumption remains at approximately 0.28 kWh.

5. Discussion and Future Work

The main contribution of the present article is to define and connect the different sources of carbon emissions involved in training and deploying BLOOM, a 176B parameter language model. While we try to be as precise as possible in our calculations, they remain an estimate based on the information available to us. In the current, final section of our article, we will compare our estimate to that of recent similar LLMs, attempt to estimate the dynamic consumption of all processes run within the scope of the BigScience workshop and discuss next steps and improvements that can be made to our approach to guide future work in the area.

5.1 Comparisons with other LLMs

A few recent LLM papers reported the carbon footprint of model training, including notable models such as OPT-175B (Zhang et al., 2022), GPT-3 (Patterson et al., 2021) and Gopher (Rae et al., 2021). However, since the accounting methodologies for reporting carbon emissions are not standardized, it is hard to precisely compare the carbon footprint of BLOOM to that of these models. In this section, we will try to disentangle the different factors for each model: (1) the energy consumption of model training, (2) the CO_2eq emissions produced by dynamic consumption during training, and (3) the CO_2eq emissions produced via dynamic consumption while taking into account datacenter PUE (i.e. overhead) as well. We present these numbers in Table 4, in which numbers in *italics* indicate numbers that have been inferred based on the information provided in the papers accompanying these models, without being stated explicitly in articles or accompanying documentation. All of the information provided regarding the carbon intensity of the grids used for training was extracted from the articles themselves.

Model name	Number of parameters	Datacenter PUE	Carbon intensity of grid used	Energy consumption	CO_2eq emissions	CO_2eq emissions $\times$ PUE
GPT-3	175B	1.1	429 gCO_2eq/kWh	1,287 MWh	*502 tonnes*	552 tonnes
Gopher	280B	1.08	330 gCO_2eq/kWh	*1,066 MWh*	*352 tonnes*	380 tonnes
OPT	175B	*1.09* [2]	*231gCO_2eq/kWh*	*324 MWh*	70 tonnes	*76.3 tonnes* [3]
BLOOM	176B	1.2	57 gCO_2eq/kWh	433 MWh	25 tonnes	30 tonnes

Table 4: Comparison of carbon emissions between BLOOM and similar LLMs. Numbers in *italics* have been inferred based on data provided in the papers describing the models.

We can see that BLOOM training resulted in less than half of the emissions of the closest comparable model, OPT (which emitted 70 tonnes compared to BLOOM's 25 tonnes), and 20 times less than GPT-3 (502 tonnes). This can be explained in large part by the carbon intensity of the energy source used for training, given that the carbon intensity of the electric grid powering Jean Zay is 57 gCO_2eq/kWh, compared to 231 gCO_2eq/kWh for OPT, 429 gCO_2eq/kWh for GPT-3 and 330 gCO_2eq/kWh for Gopher. Comparing the raw energy consumption of the models is interesting as well because we can see that BLOOM actually consumed slightly more energy than OPT: 433 MWh compared to OPT's 324 MWh, despite their proximity in size and training set up. Of course, there are also other factors that should be considered when comparing the energy consumption of models, such as the type of hardware used, the number of tokens seen, the model architecture, etc., so an exact comparison is difficult, and it is useful to consider all of the characteristics described above when comparing models.

Finally, as we mentioned in Section 4.3, the carbon footprint accounting approach proposed by Patterson et al. (2021) includes datacenter PUE, which is not always taken into account by other models. In order to allow a fair comparison, we attempt to disaggregate model carbon emissions with and without taking PUE into account in Table 4. Since the PUEs of datacenters used for training ML models are relatively efficient and very similar (ranging from 1.08 to 1.2), their contribution to the overall carbon footprint of model training is relatively small. However, as we have shown in Section 4, these numbers represent a small part of the actual carbon emissions and environmental impacts of training ML models, given that they reflect neither the embodied emissions nor the emissions due to model inference and deployment. In the next section, we attempt to go one step further by estimating the carbon footprint of intermediate experimentation and evaluation processes run within the scope of the BigScience workshop and how they compare to that of training the final BLOOM model.

5.2 Carbon Footprint of the BigScience Workshop

The training of the 176B parameter BLOOM model represents only part of the experiments that were run on the Jean Jay computing cluster as part of the BigScience workshop. In fact, if we consider the totality of experiments run by members of the BigScience project, they add up to a total of 3.46 million GPU hours (2.2 million hours of which used V100 GPUs and 1.24 million hours

2. Source: Meta (2021)
3. By contacting the authors of the OPT paper, we were able to establish that they did not consider datacenter PUE in their carbon footprint estimation; however, we do not have the necessary information to accurately do so ourselves.

used A100 GPUs), which represents an electrical consumption of 1,163,032 kWh of electricity and approximately 66.29 tonnes of CO_2eq emitted via dynamic power consumption. We break down this total into its different components, including the final BLOOM training, in Table 5, below.

Process	**Energy consumed (kWh)**	**CO_2emissions (tonnes of CO_2eq)**	**Percentage of total emissions**
176B BLOOM Model	433,196	24.69	37.24%
104B Model	266,522	15.19	22.92%
1B Model	158,972	9.06	13.68%
13B Model	87,210	4.97	7.49%
Other Models	64,257	3.66	5.53%
Miscellaneous Processes	57,961	3.30	4.98%
6B Model	51,686	2.95	4.45%
Model Evaluation	43,172	2.46	3.71%
Total	**1,163,088**	**66.29**	**100.00%**

Table 5: Breakdown of dynamic energy consumption and CO_2 emissions of different parts of the BigScience project

It is interesting to note that experimenting with intermediate models (such as the 104B, 13B and 1B models) add up to a total of 35.8 tonnes of CO_2eq, which is more than the training of the final model. This is slightly higher than the estimate made by the authors of the OPT paper, who stated that the total carbon footprint of their model is roughly 2 times higher due to experimentation, baselines and ablations (Zhang et al., 2022). However, training these models allowed us to converge on the architecture and hyperparameters of the final BLOOM model, and many of these intermediate models were also shared with the community (e.g. BLOOM 1B and BLOOM 3B). Other processes that contributed to the overall carbon emissions of the workshop included model evaluation, which accounted for 2.46 tonnes of CO_2eq, as well as miscellaneous processes such as benchmarking, data processing and tokenization (3.3 tonnes of CO_2eq). While these processes are not part of the training of the model itself, we believe that it is important to estimate and report them as part of the research and development process – we touch upon this point further in Section 5.3

If we take into account the embodied emissions of these processes, given that we used a total of 3.46 million GPU hours, this amounts to a total 35.9 tonnes of CO_2eq. Adding the embodied emissions and the idle consumption of equipment (according to the percentage described in Section 4.3, this accounts for 73.32 further tonnes of CO_2eq, bringing up the total tally up to 123.82 tonnes of CO_2eq. Furthermore, given that these additional experiments also produced several LLMs that were shared with the community and deployed, they also continue to generate carbon emissions during their deployment and usage, which we are unable to account for but keep in mind as a further addition to our estimation.

5.3 Future Work

We hope that the present article shed some light on the different sources of carbon that contribute towards an ML model's total carbon footprint and how they compare. There are, however, many

unanswered questions that we are lacking the data to pursue, some of which we will enumerate in the current section.

Gathering more precise figures regarding embodied emissions. We used the closest available figures to compute the embodied emissions of manufacturing the GPUs used for training BLOOM. However, we were unable to get the figures for the exact hardware we are using, which makes the numbers we report an estimate. More transparency is needed regarding the environmental impacts of manufacturing computing equipment given the large quantities of chemicals and minerals required (Stephens and Didden, 2013; Crawford, 2021), the significant quantities of ultra-pure water and energy needed to manufacture it (United Nations Environment Programme, 2013), as well as the complex and carbon-intensive supply chains and transportation involved in shipping them around the world (Berkhout and Hertin, 2004).

Running additional studies on model inference and deployment. The results we report in Section 4.4 barely scratch the surface of the complexity involved in deploying, scaling and maintaining ML models in practice and in real-time. We recognize this complexity and are pursuing more empirical studies to test different hardware setups and configurations and how they impact energy consumption and carbon emissions: for instance, comparing optimal conditions of model inference (i.e. when enough queries are received to fill every batch) and how that impacts the carbon cost per-query, as well as investigating different LLM prompting and usage strategies and approaches that can be used to evaluate them.

Advocating for increased transparency and granularity in carbon reporting. While some papers introducing ML models have begun reporting CO_2 emissions, which we applaud, we also believe that disaggregating this single figure into aspects such as energy consumption, carbon intensity, and PUE is needed to allow for more meaningful comparisons between models. Furthermore, tallying and reporting the carbon emissions attributable to research and development, as well as evaluation and benchmarking, is useful to contextualize the relative contribution of the final model training towards that number.

Considering the broader impacts of ML. In the existing research, the environmental impacts of information and communications technologies are classified into 3 categories: computing-related impacts due to the manufacturing of hardware and devices as well as electricity consumption; indirect impacts of deploying the models, and system-level impacts on other domains (Kaack et al., 2022). In the current article, and all those we discussed in Section 2, the focus is put solely on the direct impacts of ML models. However, it can be useful to also consider their indirect impacts on industries such as transportation, agriculture or urban planning, given increased reliance on ML technologies in these sectors. We also do not discuss ML's impact on changing consumer behaviors, for instance more usage of devices such as smart speakers or connected devices to carry out tasks which were previously done by hand. While these impacts are harder to quantify precisely, we believe that they are nonetheless worth including in the broader environmental impacts of ML.

Acknowledgements

We would like to thank the following people for their help and guidance in writing this paper: Christopher Akiki, Clement Delangue, Priya Donti, Udit Gupta, Lynn Kaack, Remi Lacroix, Pierre-Francois Lavallée, Teven Le Scao, Nicolas Patry, David Rolnick, Thomas Wang, and the other

members of the BigScience workshop. This work was granted access to the HPC resources of IDRIS under the allocation 2022-A0101012475 made by GENCI.

References

Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models, 2020.

Aurora Energy Research. France - energy and carbon. https://www.statista.com/statistics/1190067/carbon-intensity-outlook-of-france/, 2020.

Nesrine Bannour, Sahar Ghannay, Aurélie Névéol, and Anne-Laure Ligozat. Evaluating the carbon footprint of NLP methods: a survey and analysis of existing tools. In *EMNLP, Workshop SustaiNLP*, 2021.

Frans Berkhout and Julia Hertin. De-materialising and re-materialising: digital technologies and the environment. *Futures*, 36(8):903–920, 2004.

Gemma A Brady, Nikil Kapur, Jonathan L Summers, and Harvey M Thompson. A case study and critical assessment in calculating power usage effectiveness for a data centre. *Energy Conversion and Management*, 76:155–161, 2013.

Copenhagen Centre on Energy Efficiency. Greenhouse gas emissions in the ICT sector: Trends and methodologies. https://c2e2.unepdtu.org/wp-content/uploads/sites/3/2020/03/greenhouse-gas-emissions-in-the-ict-sector.pdf, 2020.

Kate Crawford. *The atlas of AI: Power, politics, and the planetary costs of artificial intelligence*. Yale University Press, 2021.

Benjamin Davy. Building an AWS EC2 carbon emissions dataset. https://medium.com/teads-engineering/building-an-aws-ec2-carbon-emissions-dataset-3f0fd76c98ac, 2021.

Google. Carbon free energy for google cloud regions. https://cloud.google.com/sustainability/region-carbon, 2022.

Abhishek Gupta, Camylle Lanteigne, and Sara Kingsley. SECure: A Social and Environmental Certificate for AI Systems. *arXiv preprint arXiv:2006.06217*, 2020.

Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. Chasing Carbon: The Elusive Environmental Footprint of Computing. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 854–867. IEEE, 2021.

HPE. HPE ProLiant DL345 Gen10 Plus server– Data Sheet. https://www.hpe.com/psnow/doc/a50005151enw, 2021.

International Telecommunication Union. Greenhouse gas emissions trajectories for the information and communication technology sector compatible with the UNFCCC Paris agreement: L. 1470. http://handle.itu.int/11.1002/1000/14084, 2020.

Lynn H Kaack, Priya L Donti, Emma Strubell, George Kamiya, Felix Creutzig, and David Rolnick. Aligning artificial intelligence with climate change mitigation. *Nature Climate Change*, pages 1–10, 2022.

Walter Klöpffer. Life cycle assessment. *Environmental Science and Pollution Research*, 4(4): 223–228, 1997.

Mascha Kurpicz, Anne-Cécile Orgerie, Anita Sobe, and Pascal Felber. Energy-proportional profiling and accounting in heterogeneous virtualized environments. *Sustainable Computing: Informatics and Systems*, 18:175–185, 2018.

Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. BLOOM: A 176B-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

George Leopold. AWS to Offer NVIDIA's T4 GPUs for AI Inferencing. `www.hpcwire.com/2019/03/19/aws-upgrades-its-gpu-backed-ai-inference-platform/`, 2019.

Anne-Laure Ligozat, Julien Lefèvre, Aurélie Bugeau, and Jacques Combaz. Unraveling the hidden environmental impacts of AI solutions for environment. *arXiv preprint arXiv:2110.11822*, 2021.

Jens Malmodin and Dag Lundén. The energy and carbon footprint of the global ICT and E&M sectors 2010–2015. *Sustainability*, 10(9):3027, 2018.

Valérie Masson-Delmotte, Panmao Zhai, Hans-Otto Pörtner, Debra Roberts, Jim Skea, Priyadarshi R Shukla, Anna Pirani, Wilfran Moufouma-Okia, Clotilde Péan, Roz Pidcock, et al. Global warming of 1.5 C. *An IPCC Special Report on the impacts of global warming of*, 1(5), 2018.

Meta. Meta 2021 sustainability report. `https://sustainability.fb.com/2021-sustainability-report/`, 2021.

Rakshit Naidu, Harshita Diddee, Ajinkya Mulay, Aleti Vardhan, Krithika Ramesh, and Ahmed Zamzam. Towards quantifying the carbon emissions of differentially private machine learning. *arXiv preprint arXiv:2107.06946*, 2021.

NVIDIA. Nvidia A100 tensor Core GPU datasheet. `https://www.nvidia.com/content/dam/en-zz/Solutions/Data-Center/a100/pdf/nvidia-a100-datasheet-nvidia-us-2188504-web.pdf`, 2022.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. The carbon footprint of machine learning training will plateau, then shrink, 2022. URL `https://arxiv.org/abs/2204.05149`.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. Scaling language models: Methods, analysis & insights from training Gopher. *arXiv preprint arXiv:2112.11446*, 2021.

Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni. Codecarbon: Estimate and track carbon emissions from machine learning computing, 2021.

Matthew Skiles, Euijin Yang, Orad Reshef, Diego Robalino Muñoz, Diana Cintron, Mary Laura Lind, Alexander Rush, Patricia Perez Calleja, Robert Nerenberg, Andrea Armani, Kasey M. Faust, and Manish Kumar. Conference demographics and footprint changed by virtual platforms. *Nature Sustainability*, 2398-9629, 2021.

Andie Stephens and Mark Didden. The development of ICT Sector Guidance: rationale, development and outcomes. In *ICT4S 2013: Proceedings of the First International Conference on Information and Communication Technologies for Sustainability, ETH Zurich*, pages 8–11, 2013.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in NLP. *arXiv preprint arXiv:1906.02243*, 2019.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*, 2020.

United Nations Environment Programme. GEO-5 for Business impacts of a changing environment on the corporate sector. *UNON*, 2013.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, James Huang, Charles Bai, et al. Sustainable AI: Environmental Implications, Challenges and Opportunities. *arXiv preprint arXiv:2111.00364*, 2021.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.