Towards a Methodology and Framework for AI Sustainability Metrics

ABSTRACT

Recently, we are witnessing truly groundbreaking achievements using AI models, such as the much talked about generative large language models, the broader area of foundation models, and the wide range of applications with a tremendous potential to accelerate scientific discovery, and enhance productivity. AI models and their use are growing at a super-linear pace leading to staggering environmental footprint implications. Inference jobs are measured by the trillions, and model parameters by the billions. This scaling up comes with a tremendous environmental cost, associated with every aspect of models' life cycle: data preparation, pre-training, and post deployment re-training, inference, and, the embodied emission of the systems used to support the execution. There is an urgent need for the community to come together and conduct a meaningful conversation about the environmental cost of AI. To do that, we ought to develop an agreed upon set of metrics, methodology, and framework to quantify AI's sustainability cost in a holistic and complete fashion. In this paper, we propose unified AI Sustainability metrics that can help foster a sustainability mind-set and enable analysis, and strategy setting. To do that, we build on and extend the data center sustainability metrics, defined in [**?**], by introducing (for the first time to our knowledge) the concept of *embodied product emission (EPC)* to capture the creation cost of software assets, such as software platforms, models, and data-sets. We then use this new concept to expand the job sustainability cost metrics (*JCS* and *ASC*) offered in [**?**] to factor in the context of the execution of jobs, such as a platform as-a-service, or a model serving inference jobs. The result is applicable to any data center job, not just for AI, and contributes towards accuracy and completeness. We then show how to apply this approach to AI, with a particular focus on the entire life cycle, including all phases of the life cycle, as well as the provenance of models, where one model is used (distilled) to create another one. We demonstrate how the metric can be used to inform a more meaningful debate about AI strategies and cost.

KEYWORDS

sustainable computing, green AI, sustainable AI, foundation models

ACM Reference Format:

. 2023. Towards a Methodology and Framework for AI Sustainability Metrics. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, **??** pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 BACKGROUND AND MOTIVATION

Artificial Intelligence (AI) is one of the fastest growing technology domains, involving academic research, businesses, and users. The enormous investment in AI led to groundbreaking applications in a diverse set of areas. AI is used for accelerating the discovery of drugs (e.g., [**?**], [**?**]), driving efficiencies at work (e.g., [**?**]), discovering new materials towards renewable storage (e.g., [**?**]), and more. At the same time, we are also in the midst of a heated debate about the potential of AI to harm (e.g., [**?**]). Risks frequently associated with AI include, fake news, biases, job losses, and, the enormous environmental cost.

The amount of compute used to train deep learning models have increased 300, 000× in six years [**?**]. Data has increased significantly, reaching exabyte scale [**?**]. The data size increase has led to a super-linear growth trend in model size [**?**]. For example, *GPT*3 based language translation tasks have increased in size 1000× ([**?**]). In contrast, systems' memory capacity only grew moderately, which has motivated a variety of scale-out infrastructure solutions (e.g., [**?** **?**]), involving thousands of AI accelerators and other specialized systems. There is no doubt that these trends come with a dire environmental cost (stemming both from embodied and operational costs). Indeed, multiple researchers and practitioners have raised the alarm on the environmental cost of AI, and offered a calls-to-action ([**?** **?** **?** **?**]).

Meanwhile, a recent development in the field of AI, is foundation models (FMs) coming to the front stage (e.g., [**?**]). FMs are trained on very broad data-sets using self supervision at scale. One of the interesting characteristics of foundation models is that through transfer learning [**?**] they can be adapted (e.g., fine-tuned, or distilled) to a wide range of downstream tasks. In fact, the majority of state-of- the-art NLP models are now adapted from one of a few foundation models, such as BERT [**?**], RoBERTa [**?**], BART [**?**], T5 [**?**], BLOOM [**?**], LLaMA [**?**]. Foundation models are not new, but the scale, scope, and emergent capabilities of foundation models in the last few years have exceeded everyone's imagination. For example, GPT-3 has 175 billion parameters (in comparison with the 'modest' 1.5 billion parameters of GPT-2), and it can be adapted via natural language prompts to perform a range of tasks despite not being trained explicitly to do many of those tasks [**?**]. Proponents of foundation models claim that the enormous amount of upfront carbon cost to train a broad model is balanced in the 'big picture' by the low cost of re-using it via fine-tuning or distillation for a particular task (e.g., [**?**]). Alas, with no unified metric that analyses the entire provenance chain of models, data-sets, and their associated cost, it is very hard to substantiate or refute this claim.

In this position paper, we apply broad knowledge of sustainability principles, protocols and standards, including the GHG Protocol ([1]), and the accompanying guidance document for cloud services ([2], chapter 4), and product Life Cycle Assessment principles (LCA) [3], to the domain of AI, in order to develop metric that can be meaningfully used to drive sustainability mindset and examine trade-offs across the life cycle of AI models, and across the

'supply-chain' of models and data-sets. As in [**?**], and also as in the Software Carbon Intensity (CSI) specification offered by the Green Software Foundation [**?**], we focus on a unit of work submitted on behalf of an end-user, i.e., inference jobs in the case of AI, as the primary object of interest. The metric is meant to capture the actual amortized sustainability cost of a job, taking into account operational overheads and associated embodied cost.

We leverage and expand the metrics defined in [**?**], concerning the sustainability cost of jobs. Specifically, We define and motivate a new term *Embodied Product Cost* for the sustainability cost of software, such as the development and testing of platforms delivered as an always running service (such as AWS's Lambda [**?**]), and, in the domain of AI, the preparing of data-sets, and training of models. We then expand the definition of job sustainability cost metrics (*JSC* and *ASC*), to also factor in the associated embodied product cost. The expanded definition can contribute towards accuracy and completeness of job sustainability metrics. We then show how to apply this expanded definition to the area of AI. To summarize, the contribution of our paper is as follows:

(1) We define a new metric *Embodied Product Cost*, that aims at expressing the 'embodied' carbon of software assets, i.e., the carbon cost of 'manufacturing' a software asset, such as the development and testing of an on-line platform, or the pre-deployment training of an AI model.
(2) We expand the definitions of Job Sustainability Cost (*JSC*), and Amortized Sustainability cost (*ASC*), defined in [**?**] to factor-in the operational cost of the software asset used as the context of the execution of the job, as well as, the 'embodied' costs of these software assets. The expanded definitions can be used generally for any data center job, not just AI, and contribute towards accuracy and completeness.
(3) We specialize and apply these new metrics to the case of AI. We show how our approach can promote a sustainability mind-set, and in particular can be used to analytically prove or refute the claim that foundations model re-use for downstream tasks is advantageous to the environment, relative to the construction of smaller more specific models, from scratch.
(4) We analyse the new opportunities for sustainability based research across the life-cycle and provenance chain of models.

1.1 Related Work

Sustainable computing has gained significant attention since a decade ago, owing to the escalating demand for computing power in Cloud computing and hyper-scale data centers, which has resulted in a significant surge in energy consumption at the data center and energy proportional computing [5]. To address this concern, there have been continuous efforts aimed at reducing power/energy consumption across different levels, ranging from chip/HW components [**? ? ?**] to data center scale [**? ? ? ?**], including cooling techniques (Power Usage Effectiveness, PUE) and/or total cost ownership (TCO) [**? ? ? ?**].

Multiple recent works take a holistic view of the problem area of sustainability in computing. Gupta *et. al.*([**?**]) posits that attention must be given to the embodied emission of systems, which is becoming the dominant factor in the carbon cost of computing environments. [**?**] argues that we need to replace static metric such as PUE and grid emission factors (CI) with dynamic, time series based, metric to enable de-carbonization of data centers by co-optimization. We agree with these assertions and believe they are complementary to the approach presented here. The most relevant to this paper is the work by Gandhi *et. al.* [**?**] that argues that we should focus on the unit of work performed (namely, *a job*), and, include in its amortized sustainability cost also a 'slice' of the embodied emission cost of the systems used for the execution and other operational overheads such as for cooling. This approach agrees with the software Carbon Intensity (SCI) specification [**?**], put forward by the Green Software Foundations [**?**]. We agree that we should aim at amortizing all these aspects into the carbon sustainability cost of jobs. Toward that goal, we posit that we are neglecting to consider additional aspects associated with the *software product(s)* used as an execution context for the job. Examples of software products are a software platform, deployed as an always-running service, required for the execution of jobs (such as a serverless job running on the AWS Lambda platform service [**?**]), and, in the domain of AI, a model, that is used to serve inference requests initiated by end users. In both these cases there is a significant overhead associated with (1) the on-going maintenance of the product such as service operations, or, model continuous re-training, and, (2) the up-front construction of the product (including, development and testing, or, data preparation and training). We propose new metrics, that build on the previous ones, but include these two neglected factors.

In the area of AI, an astounding amount of progress has been accomplished on the systems side, to design energy efficient accelerators specifically for AI workloads, optimized for metrics multiplication, and incorporating low precision, and low voltage (e.g., [4**?**]). Beyond the lens of system design, data scientists spend most of their attention fiercely competing over accuracy as a primary goal, and time-to-value as a secondary goals (e.g., [**?**]). The papers [**?**] and [**?**], were among the first to call attention to the issue of the environmental cost of AI, focusing in particularly, on the training phase of the life cycle. Michel *et. al.* [**?**], offers a single metric that can be used to compare the energy efficiency of different models with respect to their complexity (e.g., based on number of classes), and their accuracy. The work [**?**], asserts the difficulty in assessing the $gC0_2e$ of AI models, which necessitate considering the energy mix in the location of the training, and the hardware architecture used. By considering these two aspect, they correct previous estimates, such as in [**?**] by as much as 88×. They also join the call to define standards and norms for Sustainable AI. Lastly, Wu et. al [**?**], is first to suggest we need a *holistic* approach to Sustainable AI that considers *all stages of the life cycle*, and also including the embodied emission of systems used. Some of the interesting findings in this paper are: (1) that the often completely overlooked data preparation phase, consumes in some cases an equal amount of energy as the training phase, and, (2) that the ratio between energy spent in different phases of the model life cycle *varies across different use cases*. These two key observations contributed to motivating this position paper. First, we focus on two specific software assets: data-sets, and models. By defining the concept of 'embodied emission of software' we capture the gCO_2e cost of pre-preparation of data set, and

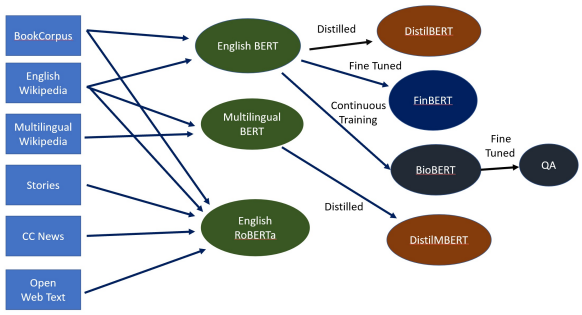

Figure 1: The 'Supply Chain' of data-sets and models

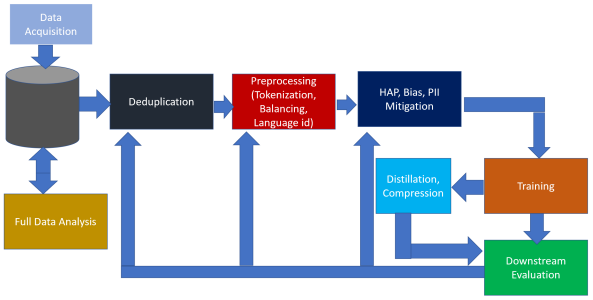

Figure 2: A typical preprocessing pipeline

pre-training of models. Further we factor-in these costs, as well as the, mostly overlooked, cost of maintaining a model with frequent re-training, in amortizing the gCO_2e cost per inference transaction, which is the unit of work performed on behalf on an end user. We postulate that this approach will allow us to analytically compare different strategies, e.g., the size of the model used, and whether it was distilled based on a previous model, or trained from scratch.

1.2 The life cycle of an AI model

To fully understand the real environmental impact, and to be able to develop the right approach to assess the impact, we must closely examine the model life cycle, end-to-end, including: data collection, model exploration and experimentation, model training, model distillation, and fine tuning, deployment, and then re-training, and inference cost.

A pre-processing pipeline, needed to curate data for training a model, could consist of the following steps: Data acquisition: via crawling (for NLP), or running simulations (materials or physics); De-duplicating: to ensure there is one copy of a document from multiple sources; Selecting documents of certain languages of interest; Splitting the documents into sentences for training; Identifying Hate Abuse Profanity and PII information one may want to filter; Forming the data files for training based on a given format. For example, NLP datasets such as Wikipedia, Stories, OpenWebText, BookCorpus, CC News are constructed via web crawling. Each of them could potentially be processed through all the above steps before it is ready for being used for training. Fig. 2 shows such a pipeline. The data-sets are used to train multiple models in various combinations. BERT, for example, was trained using English Wikipedia and BookCorpus. Whereas Multilingual BERT (mBERT) was trained on multilingual Wikipedia over 100+ languages [**?**]. RoBERTa on English data-sets spanning Wikipedia, Stories, OpenWebText, BoockCorpus as well as CC-News.

After a model is trained, the model may be distilled to bring it to a form factor which fits certain latency or space budget. For example DistilBERT [**?**]and DistilMBERT are based on BERT and mBERT respectively. After the base model is deployed, if often needs to be continuously trained to maintain accuracy. A model trained before 2019 would not have known about COVID. The frequency of re-training varies. For example, [**?**] reported re-training in weekly, or daily frequency for two different use cases.

2 TOWARDS METRICS AND METHODOLOGY FOR SUSTAINABLE AI

As we stated above, our approach includes defining a new concept, which we term *Embodied Product Cost* (Section 2.2), then using it to expend the metric definition from [**?**] (Section 2.2), and lastly, applying the result to the domain of AI (2.3).

2.1 Overview of Data Center Sustainability Metric

In this section, we give a brief overview of two of the relevant metrics defined in [**?**]. The reader is referred to [**?**] for a complete description. All metrics defined, employ the unit of "carbon dioxide equivalent" or $gCO2e$ [1].

The **Job Sustainability Cost (JSC)** is aimed at quantifying the carbon footprint associated with running a job. A *job* is any unit of work, that an application performs relevant for scaling (i.e., work jobs require more resources). The Job Sustainability Cost (JSC) is calculated as the sum of energy used by the job's share of all systems participating in executing the job, and also including a 'tax' for the cooling and power loss overheads. Energy is converted to carbon based on the mix of energy sources and their associated carbon. For example, consider a job j that consumes $1kJ$ energy executing on a host, and an additional $0.08kJ$ due to cooling and power distribution losses. If the energy source mix is 80% coal and 20% solar, then, using carbon-intensity values (and converting kJ to kWh), we have $JSC(j) = 1.08 \times (0.8 \times 820 + 0.2 \times 48) \div 3600 \approx 0.2gCO2e$.

The **Amortized Sustainability Costs (ASC)** is aimed at further including an additional 'tax' for the life cycle cost of the systems used in the computation. It is calculated by adding the job's share (or tax) of the systems' embodied emission and other costs of all systems participating in the computation to the previously defined JSC. For the job j from the example above, assume that it runs exclusively for 5 minutes on a system S with an expected lifetime of 3 years, and the embodied cost of S is $10,000$ $gCO2e$. Let $ec(j)$ be the 'tax' for embodied carbon. Then, $ec(j) = 10,000 \times \frac{5}{3 \times 365 \times 24 \times 60} = 0.031$ And, $ASC(j) = JSC(j) + ec(j) \approx 0.231gCO2e$.

The paper defines other useful metrics, such as **Job Quality per Cost Rate (JQCR)** which we will not get into here due to space limits.

[1]"gCO2e" stands for CO2 equivalent emissions, accounting for carbon dioxide and all the other greenhouse gases, such as methane and nitrous oxide

2.2 Proposed Extension

We agree with [**?**] that an end-to-end data center sustainability metrics must ultimately focus on the cost of a unit of work executed, and that all overheads to the extent possible, must be added in.

However, one overhead which is not accounted for in the definitions in Section 2.1, is that of a job *execution context*. Most jobs today, in cloud or on-premise environments, execute in a context of a continuously running shared software platform. For example, a Serverless job on AWS executes in the context of a platform termed Lambda [**?**]. A container in IBM Cloud, is executed in a context of a platform termed IBM Kubernetes Service (IKS) [**?**].There are significant overheads associated with the operations and life cycle of these platforms. For example, in IKS, there are a number of servers dedicated to managing the service in every location where the service is deployed. These management servers are always running, independently of the number of jobs executing. They are used to run management software to, e.g., monitor the health of the systems, and to meter usage for billing purposes. In storage and data services, such as IBM's Cloud Object Storage (COS) [**?**], processes wake up periodically, unprompted by any user initiated action, to perform cleanup, and tier management. Other shared services may be used across services (i.e., logging service), and the proportionate share of their use must be taken into account as well (see, [**?**]). In addition, we ought to not forget about the cost of continuous deployment (CD), and testing of the platform.

We use the term *software product* (in short, *product*) to denote any software asset, such as, a software platform that is used as-a-services to support the execution of jobs, or, an AI model used in serving inference requests, or, a data-sets which are prepared and then used for the training of AI models, etc. Each software product is associated with a life cycle, including development, deployment, and use.

Next, we define a new metric to capture the 'embodied' carbon of software products, according to the same principles that are used for calculating the embodied carbon of systems. For systems, activities include material extraction, transportation, manufacturing processes, etc, factoring-in the *entire supply chain* leading to the end product. For software, activities may vary based on the type of software product. For a platform delivered as a service they will include development and testing; for a data-set it includes data preparation and de-duplication; and, for an AI model it includes training. Each such activity consists of humans working on systems that consume energy from certain power grid, with a particular energy mix.

The **Embodied Product Cost (EPC)** is the upfront development cost of any given software product (up to the point of its deployment and use). It is calculated as the carbon footprint associated with all activities needed to create the product. We can for example refer to an activity of a developer, or, a tester, or a data-scientist, a day, as a job j. The Embodied Product Cost EPC, is the summation over all $ASC(j)$ of all the jobs over the course of the creation of the product (a period that can easily take a couple of years).

Next, we propose to expand the definition of JSC to factor-in the platform's operational cost. To avoid confusion, we denote the expanded definition as JSC^e.

The **Expanded Job Sustainability Cost** (denoted JSC^e) is calculated as the sum of energy (converted to carbon) of all systems participating in the computation, and a 'tax' for cooling and power loses, and a 'tax' for the platform overhead if a platform is used as the execution context. For example, assume that the job j is a container that runs in the context of a platform such as IBM's IKS [**?**]. This platform, in a particular location such as Dallas, uses 3 servers dedicated to management, and their associated carbon is 50 $gCO2e$ for every minute of operation. Let's assume that the job j executes for 5 minutes, and that while it is executing, there are concurrently 9 other jobs, of roughly equal size, executing on IKS in Dallas. Then, our job is 'taxed' $\frac{50}{10} \times 5$ for its share of the platform overhead. This number is added to $JSC(j)$ to derive $JSC^e(j)$. In addition, we also need to add the cost of service maintenance, and continuous development and testing. We leave this as an exercise to the reader.

Lastly, we expand accordingly the definition of the Amortized Sustainability Cost (ASC) of jobs. We claim that in addition to the embodied emission of systems participating in the execution of a job, we also have to add the embodied emission of any software product that serves in the execution.

The **Expanded Amortized Sustainability Cost**, denoted, ASC^e, is calculated as the sum of JSC^e, and, the job's share (or tax) of the systems' embodied emission (for all systems participating in the computation), and, the Embodied Product Cost (EPC) of the platform supporting the computation. As an example, if j is a container running on IKS, in Dallas, and it runs for 5 days, concurrently with 9 other equally sized jobs, and lets us assume that the expected life time of the IKS service in Dallas is 6 years, then it is taxed $EPC(IKS) \times \frac{5}{10 \times 365 \times 6}$, which is added to $ASC(j)$ to derive $ASC^e(j)$.

With the introduction of a new metric EPC to capture the embodied cost of software products, and with the expanded definition of JCS and ASC, we are now ready to turn our attention to the unique and fascinating life of AI models. We show how we apply these three metrics to meaningfully calculate the sustainability cost of AI inference jobs.

2.3 Metrics for Sustainable AI

We are now ready to apply the concepts and definitions from Section 2.2 towards metrics for Sustainable AI.

There are two 'products' that are of key relevance for AI. The first is a data-set. Refer to Section 1.2 for activities used to create a data-set. Once a data-set is ready, it can be used to train multiple different models. For a data-set d, we calculate the Embodied Product Cost $EPC(d)$ as the sum of carbon footprint of all activities involved in preparing the data-set. If we refer to each such activity as a single job then $EPC(d) = \sum_j ASC^e(j)$.

The second 'product' that is relevant to AI, is an AI model. A model is prepared (i.e., developed, or 'manufactured') via a process of experimentation, and training (see Section 1.2) leveraging one or more data-sets. A given model can also be used to develop another, sometimes task-specific, model, in a process called *distillation* or *fine-tuning*.

For a model m, $EPC(m)$ is calculated as the sum of the carbon footprint associated with the development ('manufacturing') of a model, recursively. Formally, $EPC(m) = ASC^e(j) + w_1 \times EPC(m') + w_2 \times \sum EPC(d_i)$, where, j is defined as the 'job' of preparing m based on either another model m' or multiple data-sets $d_1, d_2, ...$, and, w_1 is a tax weight to factor-in the re-use of a different model m'. It is 0 if there was no model that was re-used, or proportioned, based on its share of model re-use, and finally, w_2 is the tax weight, if the model was created based on data-sets $d_1, d_2, ...$ It is 0 if there was no data-set that was used, or proportioned, based on its share of re-use. As an example, consider the RoBERTa model ([**?**]). It was pre-trained based on a set of datasets (Wikipedia, CC-NEWS, Stories, OpenWebText, BookCorpus), using 1024 $32GB$ NVIDIA $V100$ GPUs for approximately one day. Assuming that the GPUs were working at full capacity, the maximum power consumption is $300j/s$. Thus, the energy consumed for pre-training is $1024 \times 300 \times 360 \times 24 = 737kwh$. We still have to add cooling/power-loss overheads, as well as, the 'tax' for the embodied system carbon footprint of the GPUs, as well as the fraction of embodied product carbon of the datasets used $EPC(\text{CC} - \text{NEWS} + \text{BookCorpus} + \text{Wikipedia} + \text{OpenWebText} + \text{Stories})$ (and then convert to carbon based on the grid energy mix). Yet as another example, DistilBERT ([**?**]), was created based on BERT via distillation. It has about half the total number of parameters of BERT base and retains 95% of BERT's performances on the language understanding benchmark GLUE. To create DistilBERT based on BERT, the team ([**?**]) used eight 16GB V100 GPUs for approximately three and a half days. Again assuming GPU's were used to their max capacity ($250j/s$) the energy for distillation turns out to be $14.4kwh$, and we need to add a tax for the fraction of re-use of BERT based on its EPC, and the other components as explained above.

Once a model is 'ready' it is deployed, and used to serve inference jobs. We can say that a model is *deployed* when it is available to be used to serve inference jobs. We can refer to the life cycle phase when it is serving inference jobs as *the operational phase.*

In addition to serving inference jobs, the model must be kept accurate, thus, it is being continuously re-trained. The frequency of re-training, and the cost of it, varies across models, and use cases. For example, in [**?**], the frequency reported for two different use-cases is daily, and weekly.

Let's assume that at time interval t, a model m was used to serve n inference jobs, and the carbon cost of re-training at that interval was cf_{rt} (note, there may have been multiple re-training 'jobs' at time interval T, in which case we take their sum). Then, the carbon cost of re-training cf_{rt}, must be split equally (or in proportion to the size, for non-equal jobs), across all jobs executing at time interval t. I.e., cf_{rt}/n is added.

As an example, inference jobs based on DistilBERT take $\approx$ 410 seconds to complete on CPU (Intel Xeon E5-2690 v3 Haswell @2.9GHz), which translate to $\approx 0.015Kwh$ of energy. This is about 60% of the cost of inference with the original BERT. This demonstrates the benefits of tolerating the additional upfront cost associated with distillation, for downstream efficiency gains. To calculate JCA^e we need to also include the cost of re-training which is use case specific. Adding in the cost of model re-training will encourage data scientists to practice sustainability mindset, examining, for example, the needed frequency.

Finally, lets now discuss how we calculate $ASC^e(j)$, where j is an inference job performed against a model m. Here, we leverage our Embodied Product Cost metric ($EPC(m)$), in order to account for the carbon associated with the construction of a model, as well as the embodied system cost. Thus, $ASC^e(j)$ is calculated adding in both. For a model m, if the lifetime expectancy of the model is LT, for example, 3 years, and the execution time of an inference job is t, and there are n parallel jobs executing at any given time, then $ASC^e(j) = ASC(j) + \frac{EPC(m) \times t}{LT \times n}$, where $ASC(j)$ is defined as expected, adding to $JSC^e(j)$ the embodied system emission tax, and related costs.

Again, this metric exemplifies the usefulness for fostering a sustainability mindset. A data-scientist will be encouraged to ask questions, such as 'what is the right allocation of energy to train a model based on the expected usage, and expected life time'.

3 OPPORTUNITIES

A fundamental requirement would be to establish a reliable system to collect, store and report data concerning the power/energy utilization across the complete lifecycle of the AI model. Such a challenge poses significant difficulties as certain facilities may lack the capacity for monitoring or measuring. Furthermore, even in facilities equipped with monitoring capabilities, the methodologies employed for measuring power/energy could vary widely, such as how to allocate shared resources for model training. Consequently, this issue can be bifurcated into two principal aspects: standardization and reliable information tracking.

First and foremost, it is important to standardize the procedures employed for measuring, collecting, and reporting power/energy/carbon consumption data pertaining to the AI model. This measure would enable consistent and comparable reporting metrics. This paper takes a step towards that goal, more work is needed on the system energy consumption, in particular in lieu of sharing. Additionally, it is important to ensure accuracy and integrity of the records. Models and data sets should be published with a data sheet that reports based on the metrics. Yet another area is use case based optimization. The first advantage of the proposed approach is mind-set shifts in the AI model design process. Historically, AI models have been developed with an emphasis on performance. However, upon considering our AI lifetime observability matrics, system designers would begin to prioritize both performance and energy efficiency in their AI model architectures, as evidenced by factors such as neural network depth, width, input resolution, and parameters [**?**]. This would facilitate the development of AI models that are not only high-performing but also energy-efficient, thereby reducing the need for unnecessary training practices. In particular, it would be necessary to develop methods to compare different life cycle strategies in the context of use cases, i.e., re-use (and what) vs. start from scratch.

Furthermore, this approach would allow for the identification of inefficiencies within the various phases of AI, which could be targeted for optimization. Through the implementation of scheduling techniques and power knob configurations (such as power capping and dynamic voltage and frequency scaling), we would be able to enhance the utilization of each resource while simultaneously reducing energy consumption.