Toward a Life Cycle Assessment for the Carbon Footprint of Data

Gabriel Mersy
gmersy@uchicago.edu
The University of Chicago
Chicago, Illinois, USA

Sanjay Krishnan
skr@uchicago.edu
The University of Chicago
Chicago, Illinois, USA

ABSTRACT

The growing data economy features a complex ecosystem of organizations, individuals, and devices. With digital data exchange between entities becoming ubiquitous in modern society, there is a need for carbon cost estimates that span the entire life cycle of data. We argue that accounting at the granularity of an application, process, or request can be augmented by a scheme that associates carbon annotations with data. Such a scheme would preserve continuity between interacting entities in the data economy to ensure that carbon costs are accounted for across the entire value chain of data. In short, the contributions of this paper are (1) a vision toward tracking the life cycle carbon emissions of data, as well as (2) several techniques for reducing these life cycle costs. In particular, we define the distinction between *embodied* carbon from data collection, transfer, and storage, and *operational* carbon from data use.

CCS CONCEPTS

• **Information systems → Information lifecycle management**.

KEYWORDS

data, life cycle assessment, sustainability, approximation

ACM Reference Format:
Gabriel Mersy and Sanjay Krishnan. 2023. Toward a Life Cycle Assessment for the Carbon Footprint of Data. In *2nd Workshop on Sustainable Computer Systems (HotCarbon '23), July 9, 2023, Boston, MA, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3604930.3605724

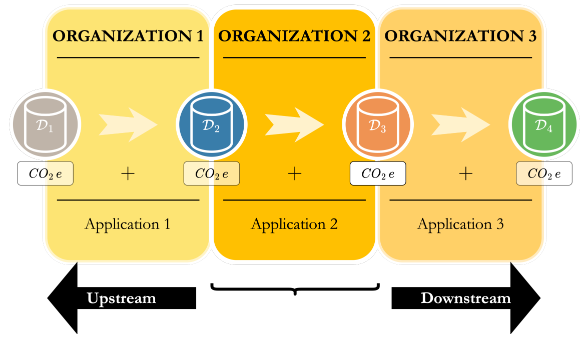

Figure 1: Data are sold from one organization to another, creating a value chain. $\mathcal{D}_i$ is purchased data that links two organizations, and each small arrow represents an application run on $\mathcal{D}_{i-1}$ to produce $\mathcal{D}_i$. Without emission estimates for the linking data, accounting is effectively siloed.

1 INTRODUCTION

As the amount of digital data continues to increase [72], a holistic assessment of computing's environmental impact must account for the costs arising from these data. While the end of Dennard scaling and the slowing of Moore's law present a resource challenge for processing these data streams at scale [24, 66], there are also costs from collecting, transferring, and storing data [79]. By viewing data as a good [43] that is manufactured, transported, and stored in memory for later use, it becomes plausible that a cradle-to-grave assessment of carbon footprint is warranted.

Except in highly specialized cases of analyzing simulated data, data are usually collected outside of the data center. For example, an e-commerce website expends energy on customer machines to track mouse events via Javascript. Similarly, air quality sensors in a building might transfer their readings to the cloud, but the sensors themselves are likely connected to the building's electricity source. In aggregate, estimates place the fraction of carbon emissions from "user" devices and networking at around two-thirds of ICT's total emissions [30]. As edge computing increases in popularity and viability, data centers will start to offload services to such devices [7, 51, 58, 61]. This trend illustrates why carbon accounting across edge devices, network devices, and data centers is necessary [6, 8, 47, 86]. The heterogeneity of device types, power sources, and geographical differences in carbon intensity make carbon accounting an acute technical challenge [34, 79].

This paper focuses on tracking the carbon implications of the data economy. Data are often exchanged among individuals and organizations, which suggests the need for the attribution of emissions at a finer granularity than that of a device or organization (e.g., such as [6]). We argue that when data are exchanged at a data market [16], the cumulative carbon cost of the data must be made aware to the buyer (in addition to the seller). In order to enable end-to-end value chain emissions for organizations [34, 62], *the cost of data must be derived from its entire value chain, spanning applications in organizations both upstream and downstream from the reporting organization (Scope 3)* as is shown in Figure 1. Similarly, in order for users to understand the carbon costs of the data that they supply, these costs must also reflect the carbon expended after the data are shared with other entities.

Akin to data provenance [14, 17, 32, 44], we introduce *carbon provenance* as an automated life cycle assessment for the carbon footprint of data [37]. Inspired by advancements in analyzing and

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
HotCarbon '23, July 9, 2023, Boston, MA, USA
© 2023 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 979-8-4007-0242-6/23/07…$15.00
https://doi.org/10.1145/3604930.3605724

optimizing the carbon footprint of hardware [33, 34, 75], we *argue that there is a similar notion of embodied carbon in the context of data*. Just as a device incurs a carbon cost due to manufacturing, data incur a carbon cost at the sensor. Just as a device is housed in a storage facility and transported to market, data are stored in memory and transferred over a network. We denote the category of emissions that are separated from any use of the data as *embodied*. On the other hand, just as a device accumulates carbon as it is used, a task that makes use of the data sets in motion the accumulation of *operational* carbon.

Carbon provenance would annotate data with carbon emission metadata, providing information to organizations and individuals alike about the environmental impact of their data. We believe that this information could enable a paradigm shift where data become a primary target for carbon optimization, addressing emissions in the embodied and operational categories. For example, if sensor data are collected routinely at a high sample rate and transferred to the cloud just to sit unused on disk, the embodied annotation exceeding the operational annotation would indicate that there are carbon costs that can be optimized in that category moving forward. In this situation, there are a variety of optimizations that can be carried out. First, the sampling rate of the sensor could be adjusted to collect fewer data (collection). Second, an approximation of the signal could be sent to the cloud instead of the original signal to save bits (transfer). Third, the data could be aged to reduce the storage requirements over time and gradually make room for other data on the device (storage). To illustrate the importance of a life cycle assessment for data, we estimate the magnitude of example embodied costs for both data collection and communication.

Data Collection Costs In Figure 2, we display the estimated embodied carbon emitted to collect 24 short webcam videos under the assumption that power is either drawn from California (CAISO) or the Midwest (MISO). An hourly average carbon intensity from December 1st, 2022 was computed for each video [3, 10, 59, 60]. Processor and DRAM power were estimated via Intel RAPL [48] while capturing frames and encoding with MJPEG [69] at 30 FPS.

The results show that the carbon cost of data collection alone can be significant. For example, assuming that this specific personal computer is connected to MISO on December 1st, 2022, a grid that is similar to the current United States average for carbon intensity, the most optimistic estimate would place the carbon emissions for MJPEG compression of a 26 second video at 74 mg CO_2 equivalent, and the most pessimistic estimate would place the value at 119 mg. *In this setting, between 3394 and 5460 26-second videos taken during that day would produce the same amount of carbon emissions as driving an average gasoline passenger vehicle one mile (1.61 km) [4].* Data collection costs will vary substantially according to both carbon intensity and factors that affect the energy consumption of sensing including the type of sensor (i.e., data modality), type of device (e.g., smartphone), sampling rate, codec, and hardware accelerator offloading.

Data Communication Costs In Figure 3, the carbon intensity along the internet backbone path of a traceroute [52] execution is displayed. In our approach, we calculate the mean carbon intensity [68] over the core nodes, and multiply by an electricity intensity (kWh/GB) estimate [9]. Packets are routed through locations that have variable carbon intensity. *Using the emission intensity estimate of 1.51 g CO_2 e/GB for this particular transfer, it would take 268 GB to produce the emissions equivalent to driving an average gasoline passenger vehicle one mile (1.61 km) [4]. To be more concrete, if a 1 GB video is uploaded to the data center and the communication path is fixed, it would take 267 complete views of the video by users in that region to produce the same amount of network emissions.*

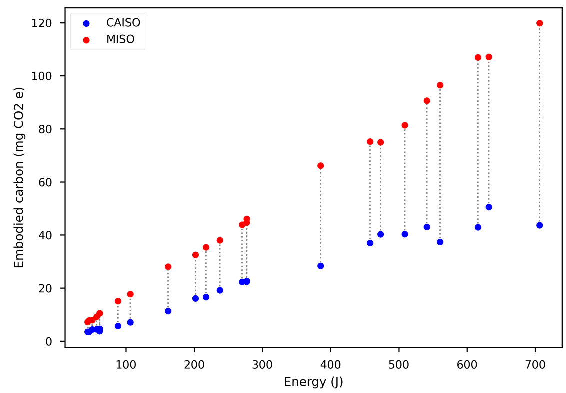

Figure 2: Estimated Energy vs. estimated embodied carbon in milligrams on MISO (red) and CAISO (blue) grids for 24 laptop webcam videos ranging from 3 to 26 seconds.

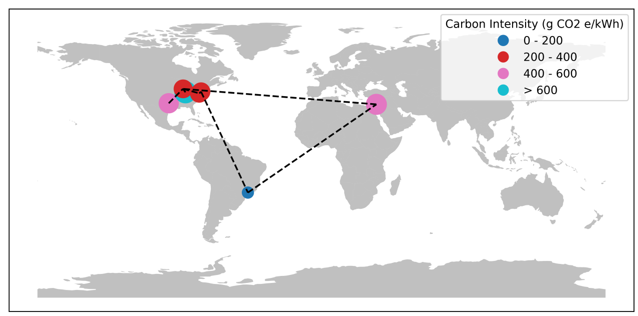

Figure 3: Estimated path carbon intensity on May 16th, 2023 at 11:00 AM CST (size and color of the circle) between the Midwest and a data center for a social media site. We obtain an emission intensity estimate of 1.51 g CO_2 e/GB.

The paper can be summarized by two key points:

(1) **Carbon Provenance**. The life cycle carbon costs of data are often hidden from decision makers, especially costs that are distributed over multiple entities and costs that originate outside of the data center. So, we need a way to track these costs in an end-to-end fashion.

(2) **Carbon-responsive Data**. There are many unexplored carbon reduction opportunities in the data life cycle that can complement existing approaches.

Fittingly, in Section 2, we discuss the estimation of carbon footprint and the annotation machinery that is propagated alongside the data. Equipped with carbon footprint information, in Section 3, we describe the promise of carbon-responsive data as a framework for

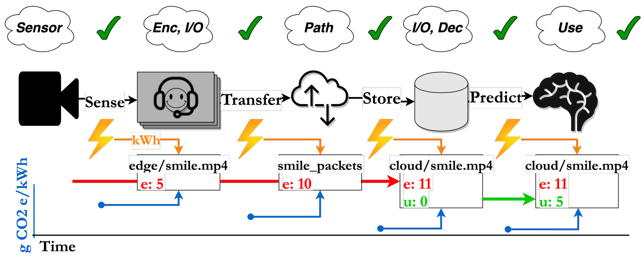

Figure 4: An example of a video life cycle in an edge architecture.

controlling emissions aside from workload shifting and designing carbon-efficient data centers [1, 35, 47, 49, 50, 64, 81, 84].

2 CARBON ACROSS THE DATA LIFE CYCLE

A *data item* is the basic component that defines the granularity of carbon accounting. For example, we can equivalently define a data item as a single value in a spreadsheet, an entire spreadsheet, or a collection of spreadsheets. Each data item is associated with two annotations $\langle e, u\rangle$ that track the embodied e and operational u categories. We show a simple example of a video life cycle in Figure 4 with the two categories clearly delineated. Note that the collection of frames consumes carbon during sensing, encoding (enc), NIC I/O, transfer between the edge and the cloud, storage device I/O, decoding (dec), and as the frames are used for an activity recognition task. If accounting only begins at the data center, costs will be underestimated, although the extent of underestimation will vary. Carbon provenance is simply metadata counters that track the carbon footprint associated with a data item across its life cycle.

Many devices are connected to either electric grids or batteries [79]. In the case of an electric grid, carbon intensity is derived from the electricity generation sources at a given time [18]. There are also rechargeable and disposable batteries. A rechargeable battery has a carbon intensity that is a function of the electricity carbon intensity over the charging period [75]. However, in the case of a disposable battery, carbon intensity could be defined according to the embodied carbon from manufacturing the battery, since the energy capacity is usually already known.

2.1 Carbon Headers

We envision a world where every meaningful data transfer between entities (e.g., through an API) is annotated with a carbon footprint. This approach is similar to [6], but we specifically define the semantics of cost aggregation *over multiple entities* and across the *full life cycle of data*. This includes costs that originate outside of the data center. Some might argue that such an approach is wasteful and might actually increase the overall carbon footprint of a system. This concern is misplaced because common data transfer protocols such as HTTP already contain significant header information (such as `User-Agent` and `X-Forwarded-For`) that is employed for responsive design and security purposes. What if HTTP requests also contained a carbon header that allowed for a retrieving server to optimize its decision based on carbon?

For the sake of simplicity, we model data exchange between two parties as `GET` and `POST` operations similar to a RESTful API. Every message is annotated with four carbon headers:

```
X-Message-Carbon-Embodied: <float>
X-Message-Carbon-Operational: <float>
X-Message-Unique-Identifier: <string>
X-Message-Carbon-Estimation-Method: <string>
```

The first header tracks all of the carbon attributed to this data item from previous collection, transfer, and storage operations. The second header tracks the carbon from previous use of the data item. The third header specifies a unique identifier that enables the aggregation of costs across entities. The fourth header indicates the accounting standard (i.e., method) that was used to estimate those numbers to preserve consistency.

New annotations for embodied and operational carbon are created at the receiving entity that track the entity's *local* carbon footprint for the data item. The cost of the network transfer between the sending and receiving entities is divided amongst the two entities. The total global footprint is then the sum of the local costs across the entities that share a data item.

As with all such headers, they are designed for application-level optimizations. They are not designed to be inherently secure or trustworthy. For example, a web-browser that observes an anomalously high carbon footprint from a web service might route future requests through a different CDN. Or a company streaming data from a data broker may decide to stop using certain data if the carbon costs are too high.

2.2 Open-Source API for Carbon Estimation

The deployment of the principle above requires reasonably accurate data item carbon estimates for a single entity (e.g., organization). We are developing an open-source toolkit that allows application developers to estimate these values on devices located outside of a data center. We highlight the initial steps that can realize such a vision below. Essentially, this is a baseline method that would be placed in a carbon header.

2.2.1 On-Device Estimates. Existing energy consumption estimates are often based on low-level power models derived from hardware performance counters or external sensor measurements from the power source [41, 45, 54, 65]. While these estimates are often accurate for the hardware, it is difficult to disaggregate the energy estimate by process. On the other hand, existing models for process-level energy estimation often do not make use of the accurate estimates provided by low-level mechanisms or external sensors [25, 56, 57], and when they do, these models are often not comprehensive (e.g., omit DRAM or accelerators) [40]. Other process-level approaches do not give a meaningful estimate of energy, since the metrics are derived exclusively from operating system statistics such as CPU usage or wakeup frequency [55].

Broadly, we seek to address these limitations by *combining accurate energy estimates for device hardware with process-level statistics.* In particular, we divide the process-level energy estimation task into two phases: (1) hardware statistics gathering and (2) estimation. Statistics are measured at a sampling interval of Δ seconds and used to estimate the energy consumption of the process over that interval. An open question is how to best distribute energy from a process over all of the involved data items. For example, a directed acyclic graph that defines a workflow consists of many tasks that produce data item outputs from data item inputs [29]. Specifically, energy can be distributed over the inputs alone, the outputs alone, or both the inputs and outputs. Machine learning inference is another concrete example, where energy can be attributed to the model or to the predictions themselves. We believe that a cost model developed from a trace of pertinent system calls could be a promising direction [71].

Hardware Statistics Gathering Phase In this phase, statistics are collected about both energy/power and resource usage. Example metrics include CPU usage, percent of RAM or swap occupied, and I/O device throughput over the past sampling interval (e.g., in MB/s) [25]. A major challenge is to ensure that the chosen metrics are available at both the global level and process level. Hardware statistics gathering can always be run in an online fashion to predict energy consumption in real-time for a given data item, but certain types of estimation models (e.g., machine learning) may require that statistics are first gathered in an offline fashion prior to model deployment.

Estimation Phase There are at least two plausible methods to estimate energy consumption for device hardware. First, if readings from an external sensor (e.g., watt meter) at the power source are available for the particular device type [41], machine learning can map global statistics for each hardware category to energy consumption [8]. *These training statistics could be collected by a third party or the hardware specifications themselves could be encoded as features to build a cross-device model.* Second, one could also train a model on low-level energy estimates that are already included internally such as processor and DRAM; these estimates would be combined with I/O device models that rely on a combination of power data from manufacturer datasheets and throughput.

Machine Learning Machine learning has been used to accurately predict data center power from CPU usage and hardware characteristics [63]. The relationship between CPU usage and power is often non-linear [28]. The problem in the process-level context is to construct a model $f(\text{GlobalStats}) = \text{GlobalEnergy}$ that produces a reasonable approximation of another function $g(\text{ProcessStats}) = \text{ProcessEnergy}$ where *process energy cannot be directly measured.* A model is first trained to predict global device energy consumption from global hardware statistics. The model is tested using a signed *carbon-sensitive* loss function such as $\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} \hat{e}_t - e_t$ where $\hat{e}_t$ is the model prediction for the global energy estimate, e_t is the true global energy estimate, and T is the number of samples. A positive value indicates that the model may overestimate the true global energy on average, while a negative value indicates that the model may underestimate the true global energy on average. Once the model is tested, process-level statistics are gathered to predict energy consumption.

Process-level I/O Device Models Input/output devices are less likely to have embedded energy counters, so without data from an external power sensor, it is unlikely that we can train a machine learning model in this setting. Instead, a model that combines operating system device counter statistics with power data from a manufacturer datasheet or a third-party experiment may be a reasonable alternative. Existing models for I/O devices often assume that the power of a particular I/O state (e.g., disk read) is constant under varying utilization [25, 56]. These models are inconsistent with recent improvements in the energy proportionality (and asymmetry) of certain device types such as low latency SSDs [12, 36]. Earlier work has found that SSD, HDD, and network interface card power is often positively correlated with throughput within the dynamic range between idle and maximum power [5, 70, 85]. We propose that device throughput statistics (e.g., MB written or MB sent) be sampled at a certain frequency at the process level and combined with power data in a model, such that each model is sensitive to the energy proportionality of the device.

Controlling Energy Overhead Adaptive sampling is a technique that can effectively conserve energy and system resources in low-power wireless sensor networks [42, 46, 82]. Adaptation of sample rate is important in the context of general data collection because the sample rate controls the end-to-end energy expenditure at the source. The system designer can adapt the sampling rate at each device according to energy or carbon emission constraints. Beyond controlling the overhead of carbon provenance, adaptive sampling can also serve as a powerful approximation method to reduce the energy expenditure of data generation (Section 3).

2.2.2 Data Production and Transportation. Accounting for the energy expended during the data life cycle begins at the sensor, as a digital signal is sampled. An energy estimate per data item is calculated from available sensor power information or an existing profile from a power meter. Once the data item is in memory, annotations $\langle e, u \rangle$ are created.

During a network transfer between devices controlled by a single entity, two possible types of operations can occur. First, a *migration* operation transfers the annotations alongside the item, since the item will only continue to be stored at the destination. Second, a *replication* operation creates a new pair of annotations that will be transferred to the destination. A data item must be assigned a unique identifier within the organization to link these local replicas. Broad estimates of internet transfer electricity intensity in kWh/GB have been developed, but these estimates are often restricted to the internet backbone [9]. Succinctly, network device reporting (or modeling) is required to account for emissions from intra-entity data transfer [86].

3 CARBON-RESPONSIVE DATA

In this section, we introduce *carbon-responsive data*, the idea that in certain use cases, modulating the *error* of a data item can reduce carbon emissions. With information on the carbon emissions of a data item, we can begin to address questions about emission reduction throughout the entire life cycle of data, including those oriented toward *constructing approximate representations of data* to conserve resources. Is there an effective way to conserve storage resources when data are becoming less valuable over time without requiring deletion? Can we reduce the retrieval carbon cost of a popular social media photo by storing multiple lossy versions that are each transmitted under a specific range of carbon intensities? Or suppose that we are using Amazon Mechanical Turk to label training data for a machine learning task, can we intelligently

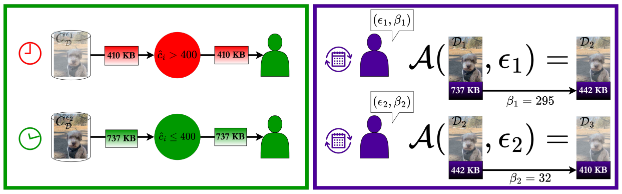

Figure 5: Adaptive compression (left) and lossy aging (right).

approve labelers based on their regional carbon intensity? All of these questions introduce interesting trade-offs that can only be addressed if we begin to account for carbon. *The embodied and operational categories are particularly important, because if embodied costs far exceed operational costs, this may indicate that the item is wasting resources as unused "dark data."*

3.1 Carbon-adaptive Approximation

Approximation, the notion of trading error for performance [19, 39, 67, 78, 80], is a technique that can reduce energy consumption in certain use cases that can tolerate error. In the context of data, many of these approximation techniques trade information loss [23], query error [20], or model accuracy [13] for cost savings. Specifically, *manipulating error tolerances in these workloads could reduce energy consumption, and this can be accomplished without explicitly requiring that the workload be shifted across time or space.* Additionally, since some network devices are energy proportional with respect to utilization [86], we describe how lossy compression could reduce the number of packets communicated during times of high carbon intensity. To manipulate error, we require an error policy $\mathcal{E}(\text{CarbonIntensity}) = \text{ErrorParameter}_1, \ldots, \text{ErrorParameter}_p$ that defines how the parameters that control the error will be set according to carbon intensity. An error policy would likely be defined by the user. This carbon-quality trade-off can be viewed as a carbon-aware data market, where error is traded for a reduction in the carbon cost of an entity's data item.

3.1.1 Carbon-adaptive Data Science. As the demand for machine learning grows, there is a need to reduce carbon costs both from training and inference [13, 15, 38, 58, 73, 74, 83]. However, there are also optimization opportunities present in other types of workloads, such as an aggregate query. For example, suppose that at the start of every hour during a given day, we observe a stream of equally-spaced sensor readings from a particulate matter 2.5 sensor. We would like to estimate the average reading over the past hour and immediately send the result to a weather application. Instead of shifting the query across time or space, we can apply statistical sampling to control the energy expenditure of the query by tolerating some error [19]. Using the current carbon intensity, a sample size is specified by an error policy, and then the query is executed over a random sample of readings. Another observation is that to save additional carbon at the air quality sensor, the sampling rate could be set according to carbon intensity, so fewer readings are generated in the first place [42].

3.1.2 Carbon-adaptive Compression. Suppose that a user frequently downloads the same batch of photos from cloud storage to their mobile device. This scenario is *communication-expensive*, since the primary energy expenditure is from packet transfer and decompression on the user's device. Additionally, we have little temporal flexibility available in this instance, since it is conceivable that the user would like to view their photos immediately. While the data center operator may optimize the storage location according to retrieval frequency [31], there is a network energy cost that is higher for frequently-retrieved photos. A predicted carbon intensity path $\hat{C} = \hat{c}_1 \rightarrow \hat{c}_2 \rightarrow \ldots \rightarrow \hat{c}_k$ is constructed where $\hat{c}_i$ is the carbon intensity at node i. Using the predicted path, the mean carbon intensity for the nodes $\bar{c}$ is computed.

Now, the question becomes: how do we reduce emissions given an estimate of the spatial carbon intensity conditions between the cloud and user device? A first idea would be to choose a more carbon-efficient route for the photos [86]. While this approach propagates packets along low carbon intensity routes to save carbon, under the assumption of at least weak energy proportionality, we describe a comparable strategy called *multiresolution compression* that adapts the *encoding* of the data that is transmitted according to estimated route carbon intensity.

Multiresolution compression can adapt the file size for frequently-retrieved items according to carbon intensity. This technique is only applied in certain lossy use cases. The core idea is to construct a single encoding that combines sub-encodings that have different errors. The approach is similar to adaptive streaming in that multiple decodings are produced [77]. A multiresolution compression algorithm produces a combined encoding $C_{\mathcal{D}}$ that is decomposed into sub-encodings $C_{\mathcal{D}}^{\epsilon_1}, \ldots, C_{\mathcal{D}}^{\epsilon_l}$ such that:

$$C_{\mathcal{D}} = C_{\mathcal{D}}^{\epsilon_1} \bigoplus C_{\mathcal{D}}^{\epsilon_2} \bigoplus \cdots \bigoplus C_{\mathcal{D}}^{\epsilon_l} \tag{1}$$

where each sub-encoding has a distinct error $\epsilon_1, \epsilon_2, \ldots, \epsilon_l$ and $\bigoplus$ denotes a combination operation [11]. To obtain a storage savings, the sub-encodings should decrease in size as the corresponding error increases.

In the trivial case, the combination operation reduces to a concatenation of the l sub-encodings. For example, we could compress a frequently-accessed image at $l = 2$ JPEG qualities to produce two sub-encodings of different sizes. When the carbon intensity along the communication path is high, the smaller sub-encoding can be transmitted, as is shown in Figure 5.

To further formalize this scheme, we can define an example error policy $\mathcal{E}$ for multiresolution compression. We start by creating a total of l carbon intensity buckets (one for each sub-encoding): $[\bar{c}_1^-, \bar{c}_1^+), [\bar{c}_2^-, \bar{c}_2^+), \ldots, [\bar{c}_l^-, \bar{c}_l^+)$ where $\bar{c}_i^-$ and $\bar{c}_i^+$ denote the minimum and maximum carbon intensity values of the bucket (i.e., the bucket boundaries). In practice, it is likely that $\bar{c}_i^- = \bar{c}_{i-1}^+$ so that the buckets cover the entire range of possible carbon intensity values. The error policy then maps the mean carbon intensity $\bar{c}$ to a specific sub-encoding $C_{\mathcal{D}}^{\epsilon_i}$ like so: $\mathcal{E}(\bar{c}) = C_{\mathcal{D}}^{\epsilon_i}$. In particular, if the predicted mean carbon intensity $\bar{c}$ falls in bucket i, the corresponding sub-encoding $C_{\mathcal{D}}^{\epsilon_i}$ is transmitted.

There is an extra storage cost in multiresolution compression, roughly upper bounded by the size of concatenating $l - 1$ sub-encodings, but this cost could potentially be justified by a reduction in retrieval latency and communication carbon emissions. There is also an extra energy cost due to the additional work that is performed during encoding and storage writes. If these costs can be minimized, we believe that this compression technique can improve the communication carbon efficiency of certain frequently-accessed items. Although multiresolution provides a routing-agnostic framework for reducing network emissions in frequently-transmitted data items, there is another category of data items that could also benefit from carbon optimization—those that are progressively becoming less valuable to the owner over time.

3.2 Lossy Data Aging

In the below discussion, we pivot to exploring the relationship between data item value and carbon costs from both operational energy and hardware manufacturing. In doing so, we describe a technique that recursively approximates data at regular time intervals to reduce storage and network resource requirements.

As the number of bits in storage and in transit increases over time, new approaches beyond scaling carbon-intensive infrastructure to match the growth in data are required. One such approach is *disposal by design*, where policies are defined to either discard items that are no longer useful or to reduce the quality of items that are becoming less useful [21, 22, 53]. A data item occupies space in a storage device that may have a high manufacturing carbon cost [76]. Similarly, retrieval requests transmit a data item over networks that must provision sufficient physical infrastructure to support the volume of data in transit [9, 86]. We argue that *disposal by design can optimize storage and network resources, preventing future manufacturing emissions that would arise from increasing resource capacity to match the growth in the volume of ingested data.*

A data item that is no longer needed can be deleted by a lossless disposal by design policy. There are also situations where lossy disposal by design can reduce the quality of the item. We can make such a reduction in quality concrete via the *data wrinkle*, an object capturing a lossy bit reduction operation on a data item. A data item can accumulate data wrinkles as it ages, progressively reducing the number of bits allocated for that item. *Note the caveat that a data wrinkle is only applied to an item that can tolerate information loss.* For example, financial reporting information often needs to be retained in a lossless form, while a photo can tolerate error in resolution and pixel intensity. Departing from a static view of the error-space trade-off, *data wrinkles dynamically introduce error across time in an incremental, recursive fashion to progressively reduce storage size. Intuitively, a data wrinkle spans the grey area between retention and deletion.*

While the idea of adding error over time may at first seem to only apply to a limited number of use cases, we argue that privacy, and more specifically differential privacy, are motivating reasons for why this approach could be considered more broadly applicable [27]. Mechanisms in differential privacy add noise from a distribution such as Laplace [26] to provide a formal guarantee on individual privacy. The key observation is that these mechanisms are structurally similar to approximation (e.g., compression with uniform quantization). In differential privacy, error is seen as an beneficial quantity. To improve the generality of a data wrinkle, we pose an open question similar to the authors of [2]: are there ways in which approximation error can improve individual privacy while simultaneously improving communication and storage carbon efficiency?

3.2.1 The Data Wrinkle. Consider a data item $\mathcal{D}$ such as a spreadsheet table or a photo. Define an (ϵ, β)-*data wrinkle* with respect to $\mathcal{D}$ where error ϵ is traded for a strictly-positive bit reduction $\beta > 0$ via an approximation operation. The specific approximation operation is kept abstract, but examples include lossy compression, a text summary, or cropping a photo. The definition of a data wrinkle is not overly restrictive in that approximation can be applied to only part of the data item.

3.2.2 Lossy Aging Over Time. Consider multiple data wrinkles that are applied over time. Let $\mathcal{D}_t$ denote the current state of a data item $\mathcal{D}$ at time step t. A user captures a photo $\mathcal{D}_0$ on their device and transfers the photo to cloud storage. In the conventional case, an image codec such as JPEG [78] would compress the user's image once after collection (a single data wrinkle), and the compressed image would be represented as b bits for any subsequent network or storage operation until deletion. In its most basic form, JPEG forces high frequency discrete cosine transform coefficients to zero by applying a quantization matrix that introduces error. *Defining the quantization matrix once is the standard approach, making an assumption that the user's tolerance for error never changes as the photo grows older. This is precisely the assumption that we challenge in the formulation of a data wrinkle, where multiple lossy operations could be applied over time.* For example, photo quality may become less important to a user over time.

At each year t, the user is given the option to further reduce the quality of the image by ϵ_t to save β_t bits in paid storage. In the running example, the quality of the image could be further reduced by applying a new quantization matrix to the remaining non-zero coefficients. Each year t that the user error tolerance increases, a data wrinkle $\mathcal{D}_t = \mathcal{A}(\mathcal{D}_{t-1}, \epsilon_t)$ is applied to save β_t bits where $\mathcal{A}$ is an approximation operation. Otherwise, the data item remains unchanged $\mathcal{D}_t = \mathcal{D}_{t-1}$. A space savings of $b - \sum_{i=1}^{t} \beta_i$ bits is achieved where b denotes the size in bits of the original compressed photo and $\beta_i = 0$ if $\mathcal{D}_i = \mathcal{D}_{i-1}$. This process is illustrated in Figure 5. While there is some carbon overhead from recursive approximation when compared to a single approximation step, we believe that in the long run, data wrinkles could serve as a viable alternative to deletion for certain use cases when applied over longer time intervals (e.g., years).

4 CONCLUSION

We presented a vision for *carbon provenance*, a life cycle assessment for the carbon footprint of data. We described techniques to perform carbon accounting for a data item both within and across entities, as well as the technical challenges of realizing our vision. Carbon-responsive data was introduced as a framework to reduce carbon emissions on a per-data item basis. Although this life cycle assessment is currently restricted to hardware energy use, there are open questions concerning the apportionment of hardware embodied carbon among data items.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work is supported by a grant from TATRC/SRI.

REFERENCES

[1] Bilge Acun, Benjamin Lee, Fiodar Kazhamiaka, Kiwan Maeng, Udit Gupta, Manoj Chakkaravarthy, David Brooks, and Carole-Jean Wu. 2023. Carbon Explorer: A Holistic Framework for Designing Carbon Aware Datacenters. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 118–132. https://doi.org/10.1145/3575693.3575754

[2] Naman Agarwal, Ananda Theertha Suresh, Felix X. Yu, Sanjiv Kumar, and Brendan McMahan. 2018. cpSGD: Communication-efficient and differentially-private distributed SGD. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (Eds.). 7575–7586. https://proceedings.neurips.cc/paper/2018/hash/21ce689121e39821d07d04faab328370-Abstract.html

[3] Environmental Protection Agency. 2023. Data Explorer. https://www.epa.gov/egrid/data-explorer.

[4] United States Environmental Protection Agency. 2022. Greenhouse Gas Emissions from a Typical Passenger Vehicle. https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle.

[5] Miriam Allalouf, Yuriy Arbitman, Michael Factor, Ronen I. Kat, Kalman Meth, and Dalit Naor. 2009. Storage Modeling for Power Estimation. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference* (Haifa, Israel) *(SYSTOR '09)*. Association for Computing Machinery, New York, NY, USA, Article 3, 10 pages. https://doi.org/10.1145/1534530.1534535

[6] Vaastav Anand, Zhiqiang Xie, Matheus Stolet, Roberta De Viti, Thomas Davidson, Reyhaneh Karimipour, Safya Alzayat, and Jonathan Mace. 2022. The Odd One Out: Energy is not like Other Metrics. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. https://www.microsoft.com/en-us/research/publication/the-odd-one-out-energy-is-not-like-other-metrics/

[7] Ganesh Ananthanarayanan, Victor Bahl, Landon Cox, Alex Crown, Shadi Nogbahi, and Yuanchao Shu. 2019. Video Analytics - Killer App for Edge Computing. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. Association for Computing Machinery, New York, NY, USA, 695–696. https://doi.org/10.1145/3307334.3328589

[8] Thomas Anderson, Adam Belay, Mosharaf Chowdhury, Asaf Cidon, and Irene Zhang. 2022. Treehouse: A Case For Carbon-Aware Datacenter Software. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. 9 pages.

[9] Joshua Aslan, Kieren Mayers, Jonathan G Koomey, and Chris France. 2018. Electricity intensity of internet data transmission: Untangling the estimates. *Journal of industrial ecology* 22, 4 (2018), 785–798.

[10] Johannes Bachmaier, Mathias Effenberger, and Andreas Gronauer. 2010. Greenhouse gas balance and resource demand of biogas plants in agriculture. *Engineering in Life Sciences* 10, 6 (2010), 560–569. https://doi.org/10.1002/elsc.201000073

[11] Bruno Barbarioli, Gabriel Mersy, Stavros Sintos, and Sanjay Krishnan. 2023. Hierarchical Residual Encoding for Multiresolution Time Series Compression. *Proc. ACM Manag. Data* 1, 1, Article 99 (may 2023), 26 pages. https://doi.org/10.1145/3588953

[12] Luiz André Barroso and Urs Hölzle. 2007. The Case for Energy-Proportional Computing. *Computer* 40, 12 (2007), 33–37. https://doi.org/10.1109/MC.2007.443

[13] Davis W. Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John V. Guttag. 2020. What is the State of Neural Network Pruning?. In *Proceedings of Machine Learning and Systems 2020, MLSys 2020, Austin, TX, USA, March 2-4, 2020*, Inderjit S. Dhillon, Dimitris S. Papailiopoulos, and Vivienne Sze (Eds.). mlsys.org. https://proceedings.mlsys.org/book/296.pdf

[14] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. 2001. Why and Where: A Characterization of Data Provenance. In *Database Theory — ICDT 2001 (Lecture Notes in Computer Science)*, Jan Van den Bussche and Victor Vianu (Eds.). Springer-Verlag GmbH, 316–330. https://doi.org/10.1007/3-540-44503-X_20

[15] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. 2019. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791* (2019).

[16] Raul Castro Fernandez. 2022. Protecting Data Markets from Strategic Buyers. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) *(SIGMOD '22)*. Association for Computing Machinery, New York, NY, USA, 1755–1769. https://doi.org/10.1145/3514221.3517855

[17] James Cheney, Laura Chiticariu, Wang-Chiew Tan, et al. 2009. Provenance in databases: Why, how, and where. *Foundations and Trends® in Databases* 1, 4 (2009), 379–474.

[18] Andrew A. Chien, Chaojie Zhang, Liuzixuan Lin, and Varsha Rao. 2022. Beyond PUE: Flexible Datacenters Empowering the Cloud to Decarbonize. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. 7 pages.

[19] Graham Cormode, Minos Garofalakis, Peter J Haas, Chris Jermaine, et al. 2011. Synopses for massive data: Samples, histograms, wavelets, sketches. *Foundations and Trends® in Databases* 4, 1–3 (2011), 1–294.

[20] Graham Cormode and S. Muthukrishnan. 2005. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithms* 55, 1 (2005), 58–75. https://doi.org/10.1016/j.jalgor.2003.12.001

[21] Susan B. Davidson, Shay Gershtein, Tova Milo, and Slava Novgorodov. 2022. Disposal by Design. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (2022). http://sites.computer.org/debull/A22mar/p10.pdf

[22] Susan B. Davidson, Shay Gershtein, Tova Milo, Slava Novgorodov, and May Shoshan. 2022. PHOcus: Efficiently Archiving Photos. *Proc. VLDB Endow.* 15, 12 (sep 2022), 3630–3633. https://doi.org/10.14778/3554821.3554861

[23] Sheng Di and Franck Cappello. 2016. Fast Error-Bounded Lossy HPC Data Compression with SZ. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 730–739. https://doi.org/10.1109/IPDPS.2016.11

[24] Mark Dietrich and Andrew Chien. 2022. Navigating Dennard, Carbon and Moore: Scenarios for the Future of NSF Advanced Computational Infrastructure. In *Practice and Experience in Advanced Research Computing* (Boston, MA, USA) *(PEARC '22)*. Association for Computing Machinery, New York, NY, USA, Article 50, 6 pages. https://doi.org/10.1145/3491418.3535156

[25] Thanh Do, Suhib Rawshdeh, and Weisong Shi. 2009. ptop: A process-level power profiling tool. In *SOSP Workshop on Power Aware Computing and Systems (HotPower'09)*.

[26] Cynthia Dwork. 2006. Differential privacy. In *Automata, Languages and Programming: 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II 33*. Springer, 1–12.

[27] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[28] Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre Barroso. 2007. Power Provisioning for a Warehouse-Sized Computer. *SIGARCH Comput. Archit. News* 35, 2 (jun 2007), 13–23. https://doi.org/10.1145/1273440.1250665

[29] The Apache Software Foundation. 2023. DAGs. https://airflow.apache.org/docs/apache-airflow/stable/core-concepts/dags.html.

[30] Charlotte Freitag, Mike Berners-Lee, Kelly Widdicks, Bran Knowles, Gordon S. Blair, and Adrian Friday. 2021. The real climate and transformative impact of ICT: A critique of estimates, trends, and regulations. *Patterns* 2, 9 (2021), 100340. https://doi.org/10.1016/j.patter.2021.100340

[31] Google. 2022. Object storage for companies of all sizes. https://cloud.google.com/storage.

[32] Todd J. Green, Grigoris Karvounarakis, and Val Tannen. 2007. Provenance Semirings. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Beijing, China) *(PODS '07)*. Association for Computing Machinery, New York, NY, USA, 31–40. https://doi.org/10.1145/1265530.1265535

[33] Udit Gupta, Mariam Elgamal, Gage Hills, Gu-Yeon Wei, Hsien-Hsin S. Lee, David Brooks, and Carole-Jean Wu. 2022. ACT: Designing Sustainable Computer Systems with an Architectural Carbon Modeling Tool. In *Proceedings of the 49th Annual International Symposium on Computer Architecture* (New York, New York) *(ISCA '22)*. Association for Computing Machinery, New York, NY, USA, 784–799. https://doi.org/10.1145/3470496.3527408

[34] Udit Gupta, Young Geun Kim, Sylvia Lee, Jordan Tse, Hsien-Hsin S. Lee, Gu-Yeon Wei, David Brooks, and Carole-Jean Wu. 2022. Chasing Carbon: The Elusive Environmental Footprint of Computing. *IEEE Micro* 42, 4 (2022), 37–47. https://doi.org/10.1109/MM.2022.3163226

[35] Walid A. Hanafy, Qianlin Liang, Noman Bashir, David E. Irwin, and Prashant J. Shenoy. 2023. CarbonScaler: Leveraging Cloud Workload Elasticity for Optimizing Carbon-Efficiency. *ArXiv* abs/2302.08681 (2023).

[36] Bryan Harris and Nihat Altiparmak. 2020. Ultra-Low Latency SSDs' Impact on Overall Energy Efficiency. In *USENIX Workshop on Hot Topics in Storage and File Systems*.

[37] Stefanie Hellweg and Llorenç Milà i Canals. 2014. Emerging approaches, challenges and opportunities in life cycle assessment. *Science* 344, 6188 (2014), 1109–1113. https://doi.org/10.1126/science.1248361

[38] Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2022. Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning. *J. Mach. Learn. Res.* 21, 1, Article 248 (jun 2022), 43 pages.

[39] Henry Hoffmann. 2015. JouleGuard: Energy Guarantees for Approximate Applications. In *Proceedings of the 25th Symposium on Operating Systems Principles* (Monterey, California) *(SOSP '15)*. Association for Computing Machinery, New York, NY, USA, 198–214. https://doi.org/10.1145/2815400.2815403

[40] hubblo org. 2023. Scaphandre. https://github.com/hubblo-org/scaphandre.

[41] Connor Imes, Lars Bergstrom, and Henry Hoffmann. 2016. A Portable Interface for Runtime Energy Monitoring. In *Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering* (Seattle, WA, USA) *(FSE 2016)*. Association for Computing Machinery, New York, NY, USA, 968–974. https://doi.org/10.1145/2950290.2983956

[42] Ankur Jain and Edward Y. Chang. 2004. Adaptive Sampling for Sensor Networks. In *Proceeedings of the 1st International Workshop on Data Management for Sensor Networks: In Conjunction with VLDB 2004* (Toronto, Canada) *(DMSN '04)*. Association for Computing Machinery, New York, NY, USA, 10–16. https://doi.org/10.1145/1052199.1052202

[43] Charles I. Jones and Christopher Tonetti. 2020. Nonrivalry and the Economics of Data. *American Economic Review* 110, 9 (September 2020), 2819–58. https://doi.org/10.1257/aer.20191330

[44] Grigoris Karvounarakis, Zachary G. Ives, and Val Tannen. 2010. Querying Data Provenance. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data* (Indianapolis, Indiana, USA) *(SIGMOD '10)*. Association for Computing Machinery, New York, NY, USA, 951–962. https://doi.org/10.1145/1807167.1807269

[45] Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen, and Zhonghong Ou. 2018. RAPL in Action: Experiences in Using RAPL for Power Measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3, 2, Article 9 (mar 2018), 26 pages. https://doi.org/10.1145/3177754

[46] Johnsen Kho, Alex Rogers, and Nicholas R Jennings. 2009. Decentralized control of adaptive sampling in wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)* 5, 3 (2009), 1–35.

[47] Young Geun Kim, Udit Gupta, Andrew McCrabb, Yonglak Son, Valeria Bertacco, David Brooks, and Carole-Jean Wu. 2023. GreenScale: Carbon-Aware Systems for Edge Computing. arXiv:2304.00404 [cs.DC]

[48] Patrick Konsor. 2019. Intel Power Gadget. https://www.intel.com/content/www/us/en/developer/articles/tool/power-gadget.html.

[49] Liuzixuan Lin, Victor M. Zavala, and Andrew A. Chien. 2021. Evaluating Coupling Models for Cloud Datacenters and Power Grids. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (Virtual Event, Italy) *(e-Energy '21)*. Association for Computing Machinery, New York, NY, USA, 171–184. https://doi.org/10.1145/3447555.3464868

[50] Julia Lindberg, Yasmine Abdennadher, Jiaqi Chen, Bernard C. Lesieutre, and Line Roald. 2021. A Guide to Reducing Carbon Emissions through Data Center Geographical Load Shifting. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems* (Virtual Event, Italy) *(e-Energy '21)*. Association for Computing Machinery, New York, NY, USA, 430–436. https://doi.org/10.1145/3447555.3466582

[51] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. 2019. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11006–11015.

[52] Zhuoqing Morley Mao, Jennifer Rexford, Jia Wang, and Randy H. Katz. 2003. Towards an Accurate AS-Level Traceroute Tool. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (Karlsruhe, Germany) *(SIGCOMM '03)*. Association for Computing Machinery, New York, NY, USA, 365–378. https://doi.org/10.1145/863955.863996

[53] Tova Milo. 2019. Getting Rid of Data. *J. Data and Information Quality* 12, 1, Article 1 (nov 2019), 7 pages. https://doi.org/10.1145/3326920

[54] mlco2. 2023. CodeCarbon. https://github.com/mlco2/codecarbon.

[55] Nicholas Nethercote. 2015. What does the OS X Activity Monitor's "Energy Impact" actually measure? https://blog.mozilla.org/nnethercote/2015/08/26/what-does-the-os-x-activity-monitors-energy-impact-actually-measure/.

[56] Adel Noureddine, Aurelien Bourdon, Romain Rouvoy, and Lionel Seinturier. 2012. Runtime Monitoring of Software Energy Hotspots. In *Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering* (Essen, Germany) *(ASE '12)*. Association for Computing Machinery, New York, NY, USA, 160–169. https://doi.org/10.1145/2351676.2351699

[57] Adel Noureddine, Romain Rouvoy, and Lionel Seinturier. 2013. A review of energy measurement approaches. *ACM SIGOPS Operating Systems Review* 47, 3 (2013), 42–49.

[58] Sébastien Ollivier, Sheng Li, Yue Tang, Stephen Cahoon, Ryan Caginalp, Chayanika Chaudhuri, Peipei Zhou, Xulong Tang, Jingtong Hu, and Alex K. Jones. 2023. Sustainable AI Processing at the Edge. *IEEE Micro* 43, 1 (2023), 19–28. https://doi.org/10.1109/MM.2022.3220399

[59] California Independent System Operator. 2023. Supply and Renewables. https://www.caiso.com/TodaysOutlook/Pages/supply.aspx.

[60] Midcontinent Independent System Operator. 2023. MISO Market Data. https://www.misoenergy.org/markets-and-operations/real-time--market-data/market-reports/.

[61] John Paparrizos, Chunwei Liu, Bruno Barbarioli, John Hwang, Ikraduya Edian, Aaron J. Elmore, Michael J. Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices. In *Conference on Innovative Data Systems Research*.

[62] Greenhouse Gas Protocol. 2023. Corporate Value Chain (Scope 3) Accounting and Reporting Standard. https://ghgprotocol.org/sites/default/files/standards/Corporate-Value-Chain-Accounting-Reporing-Standard_041613_2.pdf.

[63] Ana Radovanovic, Bokan Chen, Saurav Talukdar, Binz Roy, Alexandre Duarte, and Mahya Shahbazi. 2022. Power Modeling for Effective Datacenter Planning and Compute Management. *IEEE Transactions on Smart Grid* 13, 2 (2022), 1611–1621. https://doi.org/10.1109/TSG.2021.3125275

[64] Ana Radovanovic, Ross Koningstein, Ian Schneider, Bokan Chen, Alexandre Duarte, Binz Roy, Diyue Xiao, Maya Haridasan, Patrick Hung, Nick Care, Saurav Talukdar, Eric Mullen, Kendal Smith, Mariellen Cottman, and Walfredo Cirne. 2022. Carbon-Aware Computing for Datacenters. *IEEE Transactions on Power Systems* (2022), 1–1. https://doi.org/10.1109/TPWRS.2022.3173250

[65] Efraim Rotem, Alon Naveh, Avinash Ananthakrishnan, Eliezer Weissmann, and Doron Rajwan. 2012. Power-Management Architecture of the Intel Microarchitecture Code-Named Sandy Bridge. *IEEE Micro* 32, 2 (2012), 20–27. https://doi.org/10.1109/MM.2012.12

[66] Zechao Shang, Xi Liang, Dixin Tang, Cong Ding, Aaron J. Elmore, Sanjay Krishnan, and Michael J. Franklin. 2020. CrocodileDB: Efficient Database Execution through Intelligent Deferment. In *Conference on Innovative Data Systems Research*.

[67] Stelios Sidiroglou-Douskos, Sasa Misailovic, Henry Hoffmann, and Martin Rinard. 2011. Managing Performance vs. Accuracy Trade-Offs with Loop Perforation. In *Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering* (Szeged, Hungary) *(ESEC/FSE '11)*. Association for Computing Machinery, New York, NY, USA, 124–134. https://doi.org/10.1145/2025113.2025133

[68] CO2 Signal. 2023. Use electricity when its carbon footprint is lowest. https://www.co2signal.com/.

[69] Brian Smith. 1994. Fast Software Processing of Motion JPEG Video. In *Proceedings of the Second ACM International Conference on Multimedia* (San Francisco, California, USA) *(MULTIMEDIA '94)*. Association for Computing Machinery, New York, NY, USA, 77–88. https://doi.org/10.1145/192593.192628

[70] Ripduman Sohan, Andrew Rice, W. Moore Andrew, and Kieran Mansley. 2010. Characterizing 10 Gbps network interface energy consumption. In *IEEE Local Computer Network Conference*. 268–271. https://doi.org/10.1109/LCN.2010.5735719

[71] Diomidis Spinellis. 1994. Trace: A Tool for Logging Operating System Call Transactions. *SIGOPS Oper. Syst. Rev.* 28, 4 (oct 1994), 56–63. https://doi.org/10.1145/191525.191540

[72] Statista. 2022. Volume of data/information created, captured, copied, and consumed worldwide from 2010 to 2020, with forecasts from 2021 to 2025. https://www.statista.com/statistics/871513/worldwide-data-created/.

[73] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 3645–3650. https://doi.org/10.18653/v1/P19-1355

[74] Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and Policy Considerations for Modern Deep Learning Research. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 13693–13696. https://ojs.aaai.org/index.php/AAAI/article/view/7123

[75] Jennifer Switzer, Gabriel Marcano, Ryan Kastner, and Pat Pannuto. 2023. Junkyard Computing: Repurposing Discarded Smartphones to Minimize Carbon. In *Proceedings of the 28th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2* (Vancouver, BC, Canada) *(ASPLOS 2023)*. Association for Computing Machinery, New York, NY, USA, 400–412. https://doi.org/10.1145/3575693.3575710

[76] Swamit Tannu and Prashant J. Nair. 2022. The Dirty Secret of SSDs: Embodied Carbon. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. 7 pages.

[77] Laura Toni, Ramon Aparicio-Pardo, Karine Pires, Gwendal Simon, Alberto Blanc, and Pascal Frossard. 2015. Optimal selection of adaptive streaming representations. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 11, 2s (2015), 1–26.

[78] Gregory K Wallace. 1992. The JPEG still picture compression standard. *IEEE transactions on consumer electronics* 38, 1 (1992), xviii–xxxiv.

[79] Wenpeng Wang, Victor Ariel, Leal Sobral, Md Fazlay Rabbi Masum Billah, Nurani Saoda, Nabeel Nasir, and Bradford Campbell. 2022. Low Power but High Energy: The Looming Costs of Billions of Smart Devices. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. 7 pages.

[80] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology* 13, 7 (2003), 560–576.

[81] Philipp Wiesner, Ilja Behnke, Dominik Scheinert, Kordian Gontarska, and Lauritz Thamsen. 2021. Let's Wait Awhile: How Temporal Workload Shifting Can Reduce Carbon Emissions in the Cloud. In *Proceedings of the 22nd International Middleware Conference* (Québec city, Canada) *(Middleware '21)*. Association for Computing Machinery, New York, NY, USA, 260–272. https://doi.org/10.1145/3464298.3493399

[82] Rebecca Willett, Aline Martin, and Robert Nowak. 2004. Backcasting: adaptive sampling for sensor networks. In *Proceedings of the 3rd international symposium on Information processing in sensor networks*. 124–133.

[83] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems* 4 (2022), 795–813.

[84] Fan Yang and Andrew A. Chien. 2016. ZCCloud: Exploring Wasted Green Power for High-Performance Computing. In *2016 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 1051–1060. https://doi.org/10.1109/IPDPS.2016.96

[85] Balgeun Yoo, Youjip Won, Seokhei Cho, Sooyong Kang, Jongmoo Choi, and Sungroh Yoon. 2011. SSD Characterization: From Energy Consumption's Perspective. In *3rd Workshop on Hot Topics in Storage and File Systems (HotStorage 11)*. USENIX Association, Portland, OR. https://www.usenix.org/conference/hotstorage11/ssd-characterization-energy-consumptions-perspective

[86] Noa Zilberman, Eve M. Schooler, Uri Cummings, Rajit Manohar, Dawn Nafus, Robert Soulé, and Rick Taylor. 2022. Toward Carbon-Aware Networking. In *1st Workshop on Sustainable Computer Systems Design and Implementation (HotCarbon '22)*. 7 pages.