# Towards Decentralized and Sustainable Foundation Model Training with the Edge

LEYANG XUE, The University of Edinburgh, United Kingdom
MEGHANA MADHYASTHA, Johns Hopkins University, United States
RANDAL BURNS, Johns Hopkins University, United States
MYUNGJIN LEE, Cisco Research, United States
MAHESH K. MARINA, The University of Edinburgh, United Kingdom

Foundation models are at the forefront of AI research, appealing for their ability to learn from vast datasets and cater to diverse tasks. Yet, their significant computational demands raise issues of environmental impact and the risk of centralized control in their development. We put forward a vision towards decentralized and sustainable foundation model training that leverages the collective compute of sparingly used connected edge AI devices. We present the rationale behind our vision, particularly in support of its sustainability benefit. We further outline a set of challenges that need to be addressed to turn this vision into reality.

CCS Concepts: • **Computer systems organization → Distributed architectures**; • **Social and professional topics → Sustainability**; • **Computing methodologies → Artificial intelligence**.

## 1 Introduction

"Foundation models" [16], which are models trained at scale on broad data that can then be adapted to a wide range of downstream tasks, are at the heart of the current AI revolution. Such foundation models leverage the power of generative AI and make AI a general-purpose technology, heralding it into the industrial age [27]. They span domains as diverse as natural language processing (NLP) [65], computer vision (CV) [72], software development [34], networks [49, 54], biology [44] and more.

The immense promise that foundation models offer is in large measure owed to their scale; they exhibit an "emergent" behavior with model size and show a sharp rise in accuracy as they are scaled up beyond a point and trained with large amounts of data [45]. This has resulted in the compute demand for their training skyrocketing in recent years, needing 1000s of AI accelerators (GPUs, TPUs, etc.) [75]. Such high compute resource requirements come with a significant economic cost (e.g., [78]) and environmental cost (e.g., [38]), affordable to only a handful of global entities mostly those who run the cloud. This increasing centralization is a big impediment to the collective development of AI to benefit all [14, 17, 85]. Not only that, there is an enormous environmental cost that is rising rapidly, again rooted in the compute-intensive nature of foundation model development [58, 93]. Unsurprisingly then, AI has become one of the big four contributors to global ICT-related carbon footprint [47].

In this paper, we put forward a vision centered on the "edge" as a distributed computing platform to drive future foundation model development in a decentralized and sustainable manner. The key idea is to harness the spare compute across an amorphous collection of connected edge AI devices for foundation model training. There are billions of such mostly idle edge devices across the globe that offer ample opportunity to this end. The approach we advocate builds on the pioneering works in the realm of volunteer computing [59], aimed at harnessing the idle computing power from personal devices for distributed computing tasks. SETI@home [6] is a notable example that leveraged a million volunteered computers in the search for extraterrestrial intelligence. Differently from these early works, our focus is on enabling decentralized and sustainable foundation model development via the edge at a lower overall carbon footprint, while maintaining accuracy as with cloud-based training.

The decentralized aspect of edge-based foundation model training is obvious. We make a case that it also enhances sustainability overall by presenting our rationale through a three-step argument, as outlined below and elaborated in §4.2:

- Edge devices are designed with energy efficiency in mind and increasingly feature AI accelerators to support ML tasks, including training. By comparing the energy consumption of cloud- and edge-based training through a series of experiments, we show that it can be several times more energy efficient to train with edge devices.
- Edge devices are mostly idle (75% of the time in the case of smartphones) but have a high embodied carbon footprint. The latter is unrelated to their operational use and instead linked to manufacturing, supply chain, and recycling. So, utilizing them better helps amortize their high embodied carbon footprint.
- The above enables the opportunity to offload compute from the cloud to the edge, thereby helping reduce the cloud-side carbon footprint. This is because the baseline (embodied + operational) carbon footprint of edge devices would always be incurred simply by individuals owning them. Their better use can yield net gains in carbon footprint due to their relatively better energy efficiency and ample idle time. Our analysis shows that a 4–8× net carbon footprint reduction is possible by replacing the overall carbon footprint of a cloud GPU device with a small addition to the operational carbon footprint of a set of edge devices providing equivalent compute.

There have been some efforts in line with our vision aimed at decentralized ML training with volunteered devices [25, 73, 74] but they do not account for all the unique characteristics of the edge environment (device heterogeneity and dynamism) and crucially overlook the sustainability dimension altogether. Significant challenges remain to be addressed to bring our vision to fruition, including: (i) *Distributed training methods for edge*: Scalable and

Authors' Contact Information: Leyang Xue, leyang.xue@ed.ac.uk, The University of Edinburgh, United Kingdom; Meghana Madhyastha, mmadhya1@jhu.edu, Johns Hopkins University, United States; Randal Burns, randal@cs.jhu.edu, Johns Hopkins University, United States; Myungjin Lee, myungjle@cisco.com, Cisco Research, United States; Mahesh K. Marina, mahesh@ed.ac.uk, The University of Edinburgh, United Kingdom.

energy-efficient training requires edge-aware workload partitioning and communication minimization, as well as accounting for the carbon footprint; (ii) *Training task orchestration at the edge*: Coordinating training tasks across dynamic edge environments demands fault tolerance and low-overhead, carbon-aware scheduling; (iii) *Holistic, reliable and efficient energy consumption monitoring*: Measuring carbon impact at the edge requires accurate, low-overhead energy monitoring across all hardware components. (iv) *User incentives, security and privacy*: These are key concerns that require maintaining a seamless user experience when their devices are leveraged for training, as well as efficient ways to achieve robustness and ensure training process isolation on user devices.

## 2 Background

### 2.1 Foundation Model Development Process

Foundation models are characterized by three key traits: training on massive datasets, large model sizes, and broad generalization across tasks [16]. Large language models (LLMs), such as GPT-4, are a prominent example. Similar models exist for other modalities (e.g., Stable Diffusion [72] for vision) and for multimodal data (e.g., DeepSeek-VL2 [94], QWen2.5-VL [13]). Their development typically involves two phases (Fig. 1): pre-training and fine-tuning.

**Pre-training.** In the pre-training phase, foundation models are trained on vast datasets (e.g., 45TB in GPT-3) [18] typically through self-supervised learning to develop a broad understanding of the world, capturing general patterns, structures, and knowledge from the data. Pre-training requires all model parameters to participate as well as multiple levels of precision [87]. As the size of the dataset and model is often beyond a single node's capability, sharding of pre-training computation is required to scale it out across multiple computation and communication resources [104]. Pre-training is not necessarily a one-time operation; recently, continual pre-training [46, 81] has emerged to enable model pre-training over multiple rounds on domain-specific data.

**Fine-tuning.** Pre-training is followed by fine-tuning to specialize in specific tasks. Fine-tuning can be a lighter training that tunes models to specific benchmarks (e.g., 300GB data in FLAN [91]) or capability of chat [42, 84]. It can also be in the form of few-shot learning trained on only a few representative samples [18]. Fine-tuning not only has smaller data samples but also needs less computation when model parameters are partially fixed or quantized [24]. Post-training processes such as Reinforcement Learning from Human Feedback (RLHF) [80] also involve fine-tuning foundation models, actively with user/developer feedback. RLHF is either a continual process that runs along with model inference or the model needs to be retrained in order to adapt to new data [99].

Among the above two phases, pre-training is the most computationally heavy part followed by fine-tuning with RLHF.

### 2.2 Components of Carbon Emission

Carbon emissions for a computing device can be categorized into two main types: (i) *embodied carbon emissions* encompassing all the carbon released throughout the life cycle of the device before it starts to be used or after its disposal, including manufacturing, transportation, and installation of materials and products [1, 28, 67]; (ii) *operational carbon emissions* arising due to electricity consumption during operation of the device for computation and communication (data movement between processors/accelerators and memory, over network), while accounting for power usage efficiency of the computing infrastructure and carbon intensity of the energy source powering the infrastructure [1, 7, 67]. $CO_2$ is one of seven different greenhouse gases (GHGs) that cause global warming, but it is the most common one. So, the emissions due to different GHGs are expressed using $CO_2$ equivalent ($CO_2e$) as the common unit.

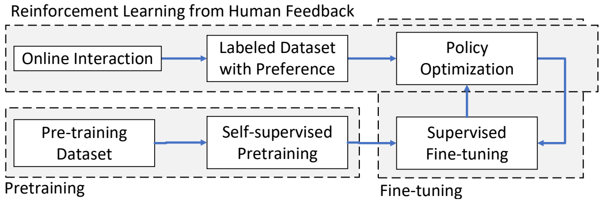


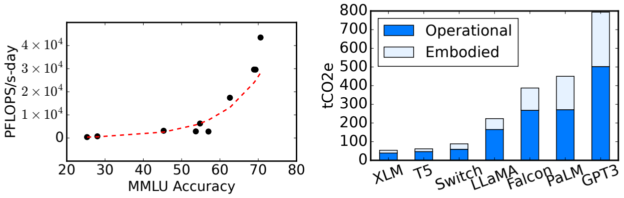
Fig. 1. Foundation model development process.

Carbon accounting is used to track the emissions from organizations, sectors, etc. This is being done for machine learning (ML) workloads too. To account for "operational" carbon in the ML context, tools like MLCO$_2$ [53] and CarbonExplorer [1] combine the power consumed (e.g., in $kW$), workload duration (e.g., in hours ($h$)) and carbon intensity of the local electricity grid (e.g., in $kgCO_2e/kWh$) to estimate the operational carbon footprint (e.g., $kW \cdot h \cdot kgCO_2e/kWh$). In contrast, the "embodied" carbon is harder to track as it involves several different aspects. Encouraged by the GHG protocol [35], holistic reporting of carbon emissions for computing products is happening (e.g., Apple [9, 10]).

## 3 Motivation

Compute requirements for model training in the deep learning era have exploded by 10-100 fold, increasing exponentially [75]. Considering the representative case of LLMs, the scaling laws [45] suggest that achieving linear growth in accuracy requires exponential growth in dataset size, model size and number of training iterations, that in turn require an exponential growth in compute demand (in terms of number and time on GPUs/TPUs). The aforementioned three ways of scaling have a multiplicative effect on the compute resource needed. Fig. 2a illustrates this relationship between compute demand (in terms of petaflop-per-second needed to complete training in one day (PFLOP/s-day) [18, 22]) and post-training model accuracy on the MMLU dataset [39] for a range of models (shown in Fig. 2b). Historically, models underwent size scaling (from XLM to GPT3), which has been shown to be essential to align with human performance [103]. At present, models additionally leverage data scaling, given that LLMs in particular and foundation models generally are usually under-trained with insufficient data [40]. Moreover, recent requirements on cross-domain [99], instruction-handling [42, 84] and multi-modal capability [26, 31] of the models further contribute to the growth in data and training length.

This rapidly growing compute demand for large model training to advance the state-of-the-art has two undesirable implications:



(a) Compute demand (b) Carbon footprint

Fig. 2. High costs of foundation model development.

**1. Centralization.** As model training has high compute demand, requiring tens of thousands of GPUs or other types of accelerators [37], it can be incredibly expensive. For instance, each training run of GPT-3 required at least $5 million worth of GPUs [78]. The overall cost of training such models is significantly more as many training runs are needed as they are developed and tuned. To make the situation even worse, essential techniques like neural architecture search might require the training of tens of thousands of neural networks [105]. As a result, the computationally intensive nature of foundation model development and the associated prohibitively high economic costs naturally favor few resource-rich organizations, leading to increased centralization of the ecosystem [14, 17, 85].

**2. Unsustainable.** The data center infrastructure to train LLMs and foundation models consumes an enormous amount of power with a potentially large carbon footprint. For instance, training ChatGPT has a 10 Gigawatt-hour (GWh) energy consumption, equivalent to the yearly consumption of 1000 US households [58]. To drive home this point, in Fig. 2b, we report the estimated carbon footprint in tons of $CO_2e$ ($tCO_2e$) for a range of foundation models to achieve the increasing levels of accuracy on the MMLU dataset, as in Fig. 2a. For this result, we use the carbon emission data from the papers of the models [22, 65, 69, 70, 84] where available; Otherwise, we use LLMCarbon [30]. This shows that the accuracy advancement comes at the expense of exponential growth in carbon footprint. Both the embodied carbon footprint (linked to renewing GPUs every 3-4 years [88]) as well as the operational carbon footprint of training in a data center [66, 68] contribute to this trend. Crucially, this path of advancement in foundation model is unsustainable [93].

## 4 Edge Centric Foundation Model Training

### 4.1 Vision

We posit that the edge can offer an effective alternative for decentralized and sustainable foundation model development. The key idea is to harness the spare (unused) compute across an amorphous collection of edge devices for foundation model training. Crucially, we seek to lower the overall carbon footprint associated with foundation model training via the edge while maintaining the accuracy as with cloud based training. Just as the notion of edge computing itself is broad, there are a wide variety of edge devices [90]. For our context, we focus on edge devices that are network-connected and equipped with AI accelerators. Examples include smartphones and laptops with GPUs and ML accelerators (e.g., Google Edge TPU, Apple M3/A16, Samsung NPU). There exist billions of such suitable edge devices. Just considering smartphones alone, there are currently over 6.5 billion of them [67]. Moreover, they are mostly idle; typically usage ranges from 3-6 hours a day [12]. With the right incentives, the massive scale and global scope of such edge devices offers plentiful opportunity to select a subset of them with collective compute capability matching that in the cloud to engage in the process of foundation model training.

The approach we advocate to leverage idle periods of edge devices is inspired by prior efforts like SETI@home [6], Computing While Charging [11] and more generally, volunteer computing [59]. The general idea behind these works is that performance-insensitive yet computationally heavy tasks can be naturally distributed among personal devices to reduce the cost. The BOINC framework [4] realized this concept, tapping into approximately one million computers volunteered by 600,000 individuals to power fundamental scientific research, especially search for extraterrestrial intelligence [5, 48]. Subsequently, Arslan et al. [11] explored the potential for exploiting unused compute in mobile devices during charging. Our work builds on these above pioneering works with the vision of turning the edge into a seamless distributed computing platform, motivated by foundation model training as a compelling and timely use case.

### 4.2 Rationale

**Energy-Efficient Edge Devices with AI Capabilities.** Energy efficiency is a principal consideration when designing edge device hardware due to the following factors: (i) *Power constraints:* Edge devices typically are battery-powered, so energy-efficient design is necessary to extend the device's operational lifetime between charges [8]. (ii) *Thermal constraints:* Unlike in cloud data centers, these devices typically lack sophisticated cooling systems. Thus, they must be designed to minimize heat production and operate effectively within their thermal budgets [77]. (iii) *Task specialization:* Modern edge devices are increasingly well-suited for machine learning workloads, as they feature specialized computation units for matrix multiplication, low-precision operations, etc. Hardware heterogeneity is a hallmark of edge devices, both across different devices and within a single device (e.g., the ARM Big.LITTLE architecture). These designs maximize both task performance and power efficiency [50, 51].

To demonstrate the energy efficiency and suitability of edge devices for ML training, we conduct experiments with three different devices: (1) Smartphone with Snapdragon 888 SoC; (2) Apple MacBook Pro laptop with M2 Pro chip; (3) Cloud GPU represented by NVDIA A5000. We consider models from the OPT series [101]. First, we pick a small OPT-125m model that can be trained using any of the above three devices. Table 1 shows the result of training this model on the MMLU dataset [39] for 100 steps with a batch size of 16 and sequence length 512. We observe that, although edge devices fare worse by 2-10x in terms of training time compared to the cloud GPU case, they have 1.5-7.5x lower energy consumption due to 15-20x lower power consumption.

Next, we choose a larger OPT-1.3B whose training can still be done with a *single* cloud GPU but requires multiple edge class devices. Specifically, assuming a homogeneous set of edge devices, this model needs 4 laptops and 15 smartphones to hold all parameters and training states. For distributed edge training, we use the state of the art (SOTA) DT-FM method [98] that employs a combination of data and pipeline parallelism. For this experiment, we

| Device | Power | Time | Energy |
|---|---|---|---|
| Smartphone (Snapdragon 888) | 10W | 3510s | 9.75Wh |
| Laptop (M2 Pro) | 15W | 480s | 2Wh |
| Cloud GPU (NVIDIA A5000) | 220W | 250s | 15.28Wh |

Table 1. Energy efficiency of ML training with single device considering OPT-125m model.

| Device | Energy |
|---|---|
| Cloud GPU (NVIDIA A5000) | 152Wh |
| 4 Laptops (M2 Pro) | 27Wh |
| 15 Smartphones (Snapdragon 888) | 98Wh |

Table 2. Energy efficiency of distributed ML training across edge devices with DT-FM and OPT-1.3B model.

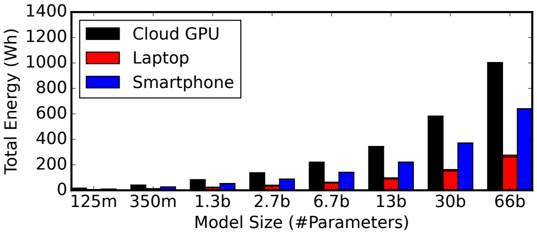


Fig. 3. Training energy consumption comparison across cloud and edge settings for different OPT model sizes with an idealized distributed training method.

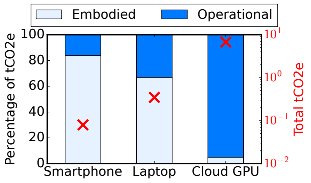


Fig. 4. Single device carbon footprint comparison: percentage breakdown between embodied and operational (left y-axis) and total absolute carbon footprint (right y-axis).

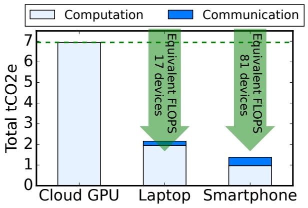


Fig. 5. Carbon emission over 3 years. Total carbon footprint on edge devices is lower when computation capacity is matched with cloud.

keep the dataset and training hyperparameters the same as in the last experiment. Assuming homogeneous and symmetric network bandwidth of 10MB/s for each edge device and 0.5W peak power for their WiFi communication modules [82], results in Table 2 show that distributed training across edge devices still offers 1.5-5x better energy efficiency compared to training with a cloud GPU, even after accounting for communication related energy consumption.

Building on the above two experiments where two specific model sizes are considered, we now study generalization across different model sizes. To this end, for fair comparison between cloud and edge settings, we consider an *idealized*[1] training method in order to factor out the differences arising due to specific distributed training methods employed in the cloud (e.g., [61]) and those designed for the edge (e.g., [98]). Fig. 3 shows the energy consumption comparison for distributed training with cloud and edge devices as a function of varying model size, again considering OPT series of models. The idealized distributed training method as outlined above is used throughout the paper. Note that depending on the model size, multiple devices are required for training in both cloud and edge settings. From the results in Fig. 3, we observe that the energy efficiency of edge devices relative to cloud GPU devices contributes to lower overall training related energy consumption with edge devices. This is particularly pronounced with the laptop case. In general, these results highlight the potential for lowering training related energy consumption with edge devices by 1.5-4x compared to the cloud case across a range of model sizes.

**Low Utilization and High Embodied Carbon Footprint.** As already noted above, edge devices often stay idle. For smartphones, this is at least 75% of the time [11, 12, 79]. At the same time, these devices possess powerful compute capabilities to efficiently perform AI tasks including training. On the flip side, edge devices have a rather high embodied carbon footprint, as highlighted in Fig. 4 that compares the carbon footprint of mobile and laptop devices against a data center GPU device (H100) over a 3-year period. For the mobile device, we use the average reported data from Apple's official product environmental report of iPhone 15 [10]. For the laptop case, we use the data from a similar report for MacBook Pro for its 3-year lifetime [9]. We use NVIDIA H100 GPU as a representative data center GPU device and estimate its carbon footprint following the methodology from $MLCO_2$ [53] and using the carbon intensity of the electricity grid averaged across North America and Europe. The embodied carbon per GPU is estimated as one eighth of the server footprint [67], given a typical GPU server has 8 GPUs.

We make two key observations from this comparison: (1) carbon footprint for edge devices is dominated by embodied carbon footprint, over 80% for mobile devices (see left y-axis), in line with the observation in previous studies [67]. Low utilization of edge devices contributes to amplifying the proportion of their embodied carbon footprint. On the other hand, operational carbon footprint is significant for data center GPU devices. (2) Data center GPU devices have significantly higher absolute amount of carbon footprint for the compute capability they offer (see right y-axis). For example, compared to the laptop case, data center GPU has at least an order of magnitude higher carbon footprint for 5x more compute capability (H100 has 267 TFLOPS versus 53 TFLOPS of M2-Ultra for FP16 computation). Considering the cloud data center infrastructure as a whole, e.g., further including footprint for CPUs and server cooling, make this comparison worse for the data center case.

**Cloud to Edge Computation Offloading Opportunity.** We further observe that the carbon footprint associated with personal devices at the edge (as shown in Fig. 4) would always be incurred simply by their ownership and baseline use. This includes the rather high embodied carbon footprint plus the operational carbon linked to the typical use of an edge device. Better utilizing them (with a corresponding marginal increase in their operational carbon footprint) can help amortize their high embodied carbon footprint. Crucially, doing so has a bigger payoff in enabling the computational tasks such as foundation model training to be shifted/offloaded from the cloud to the edge, thereby reducing the corresponding carbon footprint (both embodied and operational) on the cloud side.

[1] The idealized training method models training as a series of operators in a directed acyclic graph (e.g., [104]), with a controller distributing the computation of operations across devices. The communication load includes the model size and all intermediate results. The controller can aggregate gradients locally during forward and backward propagation without additional communication, as devices transmit output from each operator back to the controller without peer-to-peer broadcasting. In addition with this ideal method, the gradient for each parameter is transmitted only once and the intermediate result in each layer is transmitted from devices only once so that the total data transmitted is model size + (intermediate size * number of layers) for each batch.

Fig. 5 highlights the aforementioned offloading opportunity, reusing the data from Fig. 4. Considering a 3-year replacement cycle across the board, the total carbon footprint for the data center GPU (H100) is estimated to be 7 $tCO_2e$ following the methodology from MLCO$_2$ [53] and using average carbon intensity for Europe and North America for each of the years in the period 2021-23 [20]. To obtain the same compute capability in terms of FLOPS, we estimate needing 15 (M2 Pro) laptops or 69 (Snapdragon 888) smartphones assuming an additional 8 hour daily usage per device while charging [8, 11, 67]. Increase in operational energy use and corresponding carbon footprint for both types of edge devices is shown in Fig. 5. We observe that the computation from a data center GPU device can be fully offloaded to smartphone (laptop) devices with equivalent compute, resulting in a net reduction of 8x (4x) in total carbon footprint.

Fig. 5 also shows the increased communication related operational carbon footprint when edge devices participate in foundation model training 8 hours daily while they are being charged. Here we consider training the OPT-1.3B. We estimate this communication related carbon footprint using the WiFi communication related carbon emissions from prior work [82] for single devices and multiply that with the equivalent number of edge devices (15 laptops or 69 smartphones to match the cloud GPU's compute capability). We only consider a single cloud GPU in isolation for this analysis, and thus the communication overhead in that case is zero. We observe that even after accounting communication related carbon footprint, offloading to smartphone (laptop) type edge devices can still result in net reduction of 6x (3.5x) in total carbon footprint.

## 4.3 Related Work

Foundation model training is computationally intensive, requiring a large number of accelerators [37] and so is expensive. To cut the costs, recent works use spot instances (e.g., [43, 83, 98]) or multiple different clouds [96] but these do not lower the carbon footprint associated with cloud computing facilities.

In line with our vision in §4.1, some recent works pursued decentralized training with volunteered devices [25, 73, 74]. However, these works do not fully consider the unique characteristics of the edge environment that include device heterogeneity (in terms of compute and communication) as well as dynamism (in terms of participation and failures). Some of these works are also not suitable for large model training (e.g., [25]). Crucially, all these prior efforts aim solely at training efficiency and overlook the sustainability dimension altogether. Focusing only on optimizing training performance without regard to sustainability may result in an undesirably high carbon footprint. For example, devices with a higher embodied carbon footprint can be excessively used to maximize the training speed. As another example, devices in regions powered by the grid with high carbon intensity can end up being used, causing a higher operational carbon footprint as suggested by the analysis in [67].

Federated Learning (FL) [57] is a well-known distributed learning approach designed with edge devices in mind. However, the scenario targeted by FL is different from our aim to leverage edge devices for foundation model training in terms of the following three aspects: (i) *Model*: FL is akin to data parallelism [55] and typically assumes that the model can fit within the memory of a single device [21]. In contrast, foundation models including LLMs exceed the memory capacity of individual devices and so need to be sharded across devices. (ii) *Data*: FL is designed to ensure the privacy of data held on individual devices [57], whereas foundation models (e.g., [84]) are typically trained on large publicly available datasets, so our setting is more flexible with respect to data movement. (iii) *Algorithm*: Non-IID data is a defining characteristic of the setting that FL targets [57] which necessitates the use of specialized optimizers and client selection for higher model accuracy [29, 57, 71]. Our foundation model training setting does not present such a requirement. Moreover, even for training models that can fit in a single edge device, a recent analysis [67] highlighted energy efficiency concerns with the training method underlying FL.

Recently, [82] proposes repurposing *discarded* phones to create a compute cluster they call "Junkyard Computer". Similarly, accelerators from discarded phones can be put together as a server [76, 100]. The motivation is to effectively extend the lifetime of such discarded phones and amortize their high manufacturing related embodied carbon footprint. At a broad level, we share the same motivation. However, our means to that end are different and complementary via better utilizing edge devices that are operational and without modification. Furthermore, unlike [82], we have a particular focus on challenges associated with large ML model training with edge devices. For this purpose, operational devices have higher energy efficiency and better capability for training (e.g., with cutting-edge edge device accelerators).

# 5 Challenges and Potential Solutions

Decentralized training of foundation models using edge devices promises a democratized and sustainable path to AI. However, realizing this vision requires overcoming several key challenges. Unlike centralized cloud infrastructures, edge environments are inherently heterogeneous, intermittently available, and constrained in compute, memory, and energy capacity. To make edge-based training viable and competitive with cloud-based alternatives, it must achieve comparable training throughput and time-to-convergence, and maintain model fidelity using the same architectures, optimizers, and hyperparameters.

**Distributed training methods for the edge.** Training foundation models at the edge must account for heterogeneous compute, variable connectivity, and dynamic device participation, all while minimizing carbon footprint. Although edge devices collectively offer ample compute, communication overhead grows rapidly with scale and can potentially dominate energy usage. Existing compression techniques [24, 33, 87, 89, 95] reduce communication but are typically limited to fine-tuning due to accuracy concerns. While energy costs of wide-area data transmission are relatively lower—around 0.001 kWh/GB [2, 36]—compared to local computation (at least 0.02 kWh/GB), the overhead still compounds with inefficient communication patterns and should not be overlooked. The key challenge is to design distributed training methods that optimize both speed and carbon footprint by flattening communication-related energy costs as the system scales. A potential solution is to use hybrid forms of parallelism strategies (e.g., data + tensor parallelism) in conjunction with dynamic workload scheduling that maximizes parallelism and adapts to diverse characteristics across devices including bandwidth availability.

**Training task orchestration at the edge.** Orchestrating training across edge devices must balance fault tolerance, training speed, and carbon footprint for effective operation within a highly dynamic and resource-constrained environment. A unique aspect of edge devices is that they are susceptible to thermal throttling. This means sustained compute loads cause an increase in device temperature and trigger hardware-imposed slowdowns, which in turn increase latency and reduce energy efficiency.

On the fault tolerance front, traditional methods (e.g., checkpointing [43, 56, 86], replication [83], and recomputation [73, 97]) pose trade-offs between carbon footprint and recovery latency, with replication increasing carbon costs and recomputation risking slowdowns. Achieving seamless fault tolerance with minimal overhead requires identifying Pareto-optimal strategies suited for dense, communication-bound edge training workloads. Furthermore, sustainable orchestration must incorporate thermal- and carbon-aware device selection and remain lightweight compared to existing frameworks [15, 52, 60].

In addition, efficient management of millions of edge devices and connections using a limited number of server-side resources requires highly scalable and low-overhead coordination mechanisms. A promising direction is to leverage historical device activity and energy profiles to guide carbon-efficient scheduling, while enabling lightweight, decentralized orchestration with built-in support for preemptible execution and fast state recovery via proactive partial replication or reactive live migration. Techniques such as hierarchical orchestration architecture, event-driven communication, sparse state synchronization, and pub-sub messaging systems can further minimize coordination overhead.

**Holistic, reliable and efficient energy consumption monitoring.** Accurate energy monitoring is essential for environmentally sustainable training on edge devices, as it enables quantifying the operational carbon footprint, identifying inefficiencies, and informing carbon-aware scheduling and incentives. Reliable measurements are essential to evaluate and optimize the environmental impact of training workloads, particularly in heterogeneous and resource-constrained edge environments. Existing methods are often GPU- or cloud-focused [1, 19, 23, 30, 53] and fail to account holistically for other system components like memory, storage, and networking. Moreover, software-based tools offer only coarse-grained measurements [32, 64], missing the sub-millisecond energy dynamics of ML operations and risking misattribution of energy use [7]. The key challenge is to create cross-platform solutions that remain lightweight, accurate, and holistic across diverse edge devices. A promising direction is to develop accurate, component-level energy models that can infer fine-grained consumption patterns and use coarse-grained measurements for periodic calibration.

**Security and privacy.** Security and privacy are critical to enable decentralized and sustainable distributed training, as trust in the platform underpins long-term, collaborative model development. Attacks on data or model parameters can waste compute resources and increase carbon cost through retraining or recovery. While current solutions like confidential computing [41, 63] or encrypted training [92, 102] focus on local and computation-rich settings, scalable and low-overhead protections are needed for distributed, resource-constrained environments. In addition, these methods do not prevent data poisoning attacks such as adversarial model updates. Although the use of public datasets and open-source models can mitigate some privacy concerns, the training platform must still operate under zero-trust assumptions, as it may run alongside other user applications and thus be vulnerable to attacks. A potential solution is to incorporate lightweight, decentralized anomaly detection and attestation protocols that verify input data integrity and model behavior in real-time, enabling secure training without incurring the overhead of heavy cryptographic methods.

**User incentives.** To build a sustainable large-scale training platform leveraging user devices, incentivizing participation must go beyond economic rewards to include environmental awareness. Current systems offer service credits (e.g., NetMind [62], Aioz [3]), but few encourage behaviors like charging during cleaner energy hours or using efficient chargers. Since training adds compute and memory load, it can degrade user experience and charging efficiency, necessitating careful coordination. The challenge lies in enabling preemptible, seamless training with fault-tolerant recovery, informed by user activity and energy conditions. It would be more beneficial to reward users not only for contributing compute but also for aligning their participation with low-carbon and high-efficiency energy windows (e.g., more hours during days with solar power), using lightweight client-side monitors to track availability, charging state, and responsiveness.

## Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions that greatly improved this paper. The authors would like to also specially thank Jon Crowcroft for his review and helpful feedback on the paper.

## 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 *ASPLOS (2)*. ACM, 118–132.

[2] Tersiteab Adem, Andrew McCrabb, Vidushi Goyal, and Valeria Bertacco. 2024. Evergreen: Comprehensive Carbon Model for Performance-Emission Tradeoffs. In *IISWC*. IEEE, 132–143.

[3] Aioz. 2025. DePIN for Web3, Empowering a Fast, Secure and Decentralized Future. https://aioz.network/.

[4] David P. Anderson. 2020. BOINC: A Platform for Volunteer Computing. *J. Grid Comput.* 18, 1 (2020), 99–122.

[5] David P. Anderson, Carl Christensen, and Bruce Allen. 2006. Grid resource management – Designing a runtime system for volunteer computing. In *SC*. ACM Press, 126.

[6] David P. Anderson, Jeff Cobb, Eric Korpela, Matt Lebofsky, and Dan Werthimer. 2002. SETI@home: An Experiment in Public-Resource Computing. *Commun. ACM* 45, 11 (2002), 56–61.

[7] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. Carbontracker: Tracking and Predicting the Carbon Footprint of Training Deep Learning Models. arXiv:2007.03051

[8] Apple. [n. d.]. Apple unveils M2 Pro and M2 Max: next-generation chips for next-level workflows. https://www.apple.com/uk/newsroom/2023/01/apple-unveils-m2-pro-and-m2-max-next-generation-chips-for-next-level-workflows/.

[9] Apple. 2023. Product Environmental Report 16-inch MacBook Pro. https://www.apple.com/environment/pdf/products/notebooks/16-inch_MacBook_Pro_PER_Oct2023.pdf.

[10] Apple. 2023. Product Environmental Report iPhone 15 Pro and iPhone 15 Pro Max. https://www.apple.com/environment/pdf/products/iphone/iPhone_15_Pr

o_and_iPhone_15_Pro_Max_Sept2023.pdf.

[11] Mustafa Y. Arslan, Indrajeet Singh, Shailendra Singh, Harsha V. Madhyastha, Karthikeyan Sundaresan, and Srikanth V. Krishnamurthy. 2012. Computing while charging: building a distributed computing infrastructure using smartphones. In *CoNEXT*. ACM, 193–204.

[12] Backlinko. [n. d.]. Smartphone Usage Statistics. https://backlinko.com/smartphone-usage-statistics.

[13] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. 2025. Qwen2.5-VL Technical Report. arXiv:2502.13923

[14] Brian R. Bartoldson, Bhavya Kailkhura, and Davis W. Blalock. 2023. Compute-Efficient Deep Learning: Algorithmic Trends and Opportunities. *J. Mach. Learn. Res.* 24 (2023), 122:1–122:77.

[15] Giovanni Bartolomeo, Mehdi Yosofie, Simon Bäurle, Oliver Haluszczynski, Nitinder Mohan, and Jörg Ott. 2023. Oakestra: A Lightweight Hierarchical Orchestration Framework for Edge Computing. In *ATC*. USENIX Association.

[16] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ B. Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri S. Chatterji, Annie S. Chen, Kathleen Creel, Jared Quincy Davis, Dorottya Demszky, Chris Donahue, Moussa Doumbouya, Esin Durmus, Stefano Ermon, John Etchemendy, Kawin Ethayarajh, Li Fei-Fei, Chelsea Finn, Trevor Gale, Lauren Gillespie, Karan Goel, Noah D. Goodman, Shelby Grossman, Neel Guha, Tatsunori Hashimoto, Peter Henderson, John Hewitt, Daniel E. Ho, Jenny Hong, Kyle Hsu, Jing Huang, Thomas Icard, Saahil Jain, Dan Jurafsky, Pratyusha Kalluri, Siddharth Karamcheti, Geoff Keeling, Fereshte Khani, Omar Khattab, Pang Wei Koh, Mark S. Krass, Ranjay Krishna, Rohith Kuditipudi, and et al. 2021. On the Opportunities and Risks of Foundation Models. arXiv:2108.07258

[17] Nicholas J Borge. 2022. *Deep pockets: The economics of deep learning and the emergence of new AI platforms.* Ph. D. Dissertation. Massachusetts Institute of Technology.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models are Few-Shot Learners. In *NeurIPS*.

[19] Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2021. IrEne: Interpretable Energy Prediction for Transformers. In *ACL/IJCNLP (1)*. Association for Computational Linguistics, 2145–2157.

[20] Carbon Footprint. [n. d.]. COUNTRY SPECIFIC ELECTRICITY GRID GREENHOUSE GAS EMISSION FACTORS. https://www.carbonfootprint.com/.

[21] Cheng-Wei Ching, Xin Chen, Taehwan Kim, Bo Ji, Qingyang Wang, Dilma Da Silva, and Liting Hu. 2024. Totoro: A Scalable Federated Learning Engine for the Edge. In *EuroSys*. ACM, 182–199.

[22] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. PaLM: Scaling Language Modeling with Pathways. *J. Mach. Learn. Res.* 24 (2023), 240:1–240:113.

[23] Shaiful Alam Chowdhury, Stephanie Borle, Stephen Romansky, and Abram Hindle. 2019. GreenScaler: training software energy models with automatic test generation. *Empir. Softw. Eng.* 24, 4 (2019), 1649–1692.

[24] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In *NeurIPS*.

[25] Michael Diskin, Alexey Bukhtiyarov, Max Ryabinin, Lucile Saulnier, Quentin Lhoest, Anton Sinitsin, Dmitry Popov, Dmitry V. Pyrkin, Maxim Kashirin, Alexander Borzunov, Albert Villanova del Moral, Denis Mazur, Ilia Kobelev, Yacine Jernite, Thomas Wolf, and Gennady Pekhimenko. 2021. Distributed Deep Learning In Open Collaborations. In *NeurIPS*. 7879–7897.

[26] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. 2023. PaLM-E: An Embodied Multimodal Language Model. In *ICML (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 8469–8488.

[27] The Economist. [n. d.]. Artificial intelligence enters its industrial age. https://www.economist.com/foundational-AI-pod.

[28] Tamar Eilam, Pedro D. Bello-Maldonado, Bishwaranjan Bhattacharjee, Carlos H. A. Costa, Eun Kyung Lee, and Asser N. Tantawi. 2023. Towards a Methodology and Framework for AI Sustainability Metrics. In *HotCarbon*. ACM, 13:1–13:7.

[29] Sannara Ek, François Portet, Philippe Lalanda, and Germán Vega. 2021. A Federated Learning Aggregation Algorithm for Pervasive Computing: Evaluation and Comparison. In *PerCom*. IEEE, 1–10.

[30] Ahmad Faiz, Sotaro Kaneda, Ruhan Wang, Rita Osi, Prateek Sharma, Fan Chen, and Lei Jiang. 2024. LLMCarbon: Modeling the end-to-end Carbon Footprint of Large Language Models. In *ICLR*. OpenReview.net.

[31] Nanyi Fei, Zhiwu Lu, Yizhao Gao, Guoxing Yang, Yuqi Huo, Jingyuan Wen, Haoyu Lu, Ruihua Song, Xin Gao, Tao Xiang, et al. 2022. Towards artificial general intelligence via a multimodal foundation model. *Nature Communications* 13, 1 (2022), 3094.

[32] Firefox. 2025. Firefox Source Tree Documentation. https://firefox-source-docs.mozilla.org/performance/tools_power_rapl.html.

[33] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers. arXiv:2210.17323

[34] GitHub. [n. d.]. GitHub Copilot · Your AI pair programmer. https://github.com/features/copilot.

[35] Greenhouse Gas Protocol. [n. d.]. The GHG Protocol for Project Accounting. https://ghgprotocol.org/project-protocol.

[36] Viktor Urban Gsteiger, Pin Hong (Daniel) Long, Yiran (Jerry) Sun, Parshan Javanrood, and Mohammad Shahrad. 2024. Caribou: Fine-Grained Geospatial Shifting of Serverless Applications for Sustainability. In *SOSP*. ACM, 403–420.

[37] Matt Hamblen. [n. d.]. Update: ChatGPT runs 10K Nvidia training GPUs with potential for thousands more. https://www.fierceelectronics.com/sensors/chatgpt-runs-10k-nvidia-training-gpus-potential-thousands-more.

[38] Yuelin Han, Zhifeng Wu, Pengfei Li, Adam Wierman, and Shaolei Ren. 2024. The Unpaid Toll: Quantifying the Public Health Impact of AI. arXiv:2303.08774

[39] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring Massive Multitask Language Understanding. In *ICLR*. OpenReview.net.

[40] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and Laurent Sifre. 2022. Training Compute-Optimal Large Language Models. arXiv:2203.15556

[41] Intel. [n. d.]. Intel Confidential Computing Solutions. https://www.intel.com/content/www/us/en/security/confidential-computing.html.

[42] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of Experts. arXiv:2401.04088

[43] Harry Jiang, Xiaoxi Zhang, and Carlee Joe-Wong. 2022. DOLL: Distributed OnLine Learning Using Preemptible Cloud Instances. *SIGMETRICS Perform. Evaluation Rev.* 50, 2 (2022), 21–23.

[44] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature* 596, 7873 (2021), 583–589.

[45] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling Laws for Neural Language Models. arXiv:2001.08361

[46] Zixuan Ke, Yijia Shao, Haowei Lin, Tatsuya Konishi, Gyuhak Kim, and Bing Liu. 2023. Continual Pre-training of Language Models. In *ICLR*. OpenReview.net.

[47] Bran Knowles. 2021. ACM TechBrief: Computing and climate change.

[48] Eric Korpela, Dan Werthimer, David Anderson, Jeff Cobb, and Matt Lebofsky. 2001. SETI@home – Massively Distributed Computing for SETI. *Computing in Science & Engineering* 3, 1 (2001), 78–83.

[49] Manikanta Kotaru. 2023. Adapting Foundation Models for Operator Data Analytics. In *Proceedings of the 22nd ACM Workshop on Hot Topics in Networks, HotNets 2023, Cambridge, MA, USA, November 28-29, 2023*. ACM, 172–179. https://doi.org/10.1145/3626111.3628191

[50] Gokul Krishnan, A. Alper Goksoy, Sumit K. Mandal, Zhenyu Wang, Chaitali Chakrabarti, Jae-sun Seo, Ümit Y. Ogras, and Yu Cao. 2022. Big-Little Chiplets for In-Memory Acceleration of DNNs: A Scalable Heterogeneous Architecture. In *ICCAD*. ACM, 8:1–8:9.

[51] Gokul Krishnan, Sumit K. Mandal, Chaitali Chakrabarti, Jae-sun Seo, Ümit Y. Ogras, and Yu Cao. 2020. Interconnect-Aware Area and Energy Optimization for In-Memory Acceleration of DNNs. *IEEE Des. Test* 37, 6 (2020), 79–87.

[52] KubeEdge. 2025. Kubernetes Native Edge Computing Framework. https://kubeedge.io/.

[53] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. 2019. Quantifying the Carbon Emissions of Machine Learning. arXiv:1910.09700

[54] Franck Le, Mudhakar Srivatsa, Raghu K. Ganti, and Vyas Sekar. 2022. Rethinking data-driven networking with foundation models: challenges and opportunities. In *HotNets*. ACM, 188–197.

[55] Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. 2014. Scaling Distributed Machine Learning with the Parameter Server. In *OSDI*. USENIX Association, 583–598.

[56] Weijian Liu, Mingzhen Li, Guangming Tan, and Weile Jia. 2025. Mario: Near Zero-cost Activation Checkpointing in Pipeline Parallelism. In *PPoPP*. ACM, 197–211.

[57] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *AISTATS (Proceedings of Machine Learning Research, Vol. 54)*. PMLR, 1273–1282.

[58] Sarah McQuate. 2023. Q&A: UW researcher discusses just how much energy ChatGPT uses. https://www.washington.edu/news/2023/07/27/how-much-energy-does-chatgpt-use/.

[59] Tessema M. Mengistu and Dunren Che. 2019. Survey and Taxonomy of Volunteer Computing. *ACM Comput. Surv.* 52, 3 (2019), 59:1–59:35.

[60] MLCommons. [n. d.]. MLCube. https://github.com/mlcommons/mlcube.

[61] Deepak Narayanan, Mohammad Shoeybi, Jared Casper, Patrick LeGresley, Mostofa Patwary, Vijay Korthikanti, Dmitri Vainbrand, Prethvi Kashinkunti, Julie Bernauer, Bryan Catanzaro, Amar Phanishayee, and Matei Zaharia. 2021. Efficient large-scale language model training on GPU clusters using Megatron-LM. In *SC*. ACM, 58.

[62] NetMind. 2025. AI powerhouse for the future. https://netmind.ai/home.

[63] NVIDIA. [n. d.]. NVIDIA Confidential Computing Secure data and AI models in use. https://www.nvidia.com/en-gb/data-center/solutions/confidential-computing.

[64] NVIDIA. 2025. NVML API Reference. http://docs.nvidia.com/deploy/nvml-api/index.html.

[65] OpenAI. 2023. GPT-4 Technical Report. arXiv:2303.08774

[66] Pratyush Patel, Theo Gregersen, and Thomas E. Anderson. 2023. An Agile Pathway Towards Carbon-aware Clouds. In *HotCarbon*. ACM, 10:1–10:8.

[67] David Patterson, Jeffrey M. Gilbert, Marco Gruteser, Efren Robles, Krishna Sekar, Yong Wei, and Tenghui Zhu. 2024. Energy and Emissions of Machine Learning on Smartphones vs. the Cloud. *Commun. ACM* 67, 2 (2024), 86–97.

[68] David A. Patterson, Joseph Gonzalez, Urs Hölzle, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. 2022. The Carbon Footprint of Machine Learning Training Will Plateau, Then Shrink. *Computer* 55, 7 (2022), 18–28.

[69] Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling Language Models: Methods, Analysis & Insights from Training Gopher. arXiv:2112.11446

[70] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *J. Mach. Learn. Res.* 21 (2020), 140:1–140:67.

[71] Sashank J. Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečný, Sanjiv Kumar, and Hugh Brendan McMahan. 2021. Adaptive Federated Optimization. In *ICLR*. OpenReview.net.

[72] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-Resolution Image Synthesis with Latent Diffusion Models. In *CVPR*. IEEE, 10674–10685.

[73] Max Ryabinin, Tim Dettmers, Michael Diskin, and Alexander Borzunov. 2023. SWARM Parallelism: Training Large Models Can Be Surprisingly Communication-Efficient. In *ICML (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 29416–29440.

[74] Max Ryabinin and Anton Gusev. 2020. Towards Crowdsourced Training of Large Neural Networks using Decentralized Mixture-of-Experts. In *NeurIPS*.

[75] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos. 2022. Compute Trends Across Three Eras of Machine Learning. In *IJCNN*. IEEE, 1–8.

[76] Mohammad Shahrad and David Wentzlaff. 2017. Towards Deploying Decommissioned Mobile Devices as Cheap Energy-Efficient Compute Nodes. In *HotCloud*. USENIX Association.

[77] Yakun Sophia Shao, Jason Clemons, Rangharajan Venkatesan, Brian Zimmer, Matthew Fojtik, Nan Jiang, Ben Keller, Alicia Klinefelter, Nathaniel Ross Pinckney, Priyanka Raina, Stephen G. Tell, Yanqing Zhang, William J. Dally, Joel S. Emer, C. Thomas Gray, Brucek Khailany, and Stephen W. Keckler. 2019. Simba: Scaling Deep-Learning Inference with Multi-Chip-Module-Based Architecture. In *MICRO*. ACM, 14–27.

[78] Craig S. Smith. 2023. What Large Models Cost You – There Is No Free AI Lunch. https://www.forbes.com/sites/craigsmith/2023/09/08/what-large-models-cost-you--there-is-no-free-ai-lunch/?sh=2b6d10724af7.

[79] Vijay Srinivasan, Saeed Moghaddam, Abhishek Mukherji, Kiran K. Rachuri, Chenren Xu, and Emmanuel Munguia Tapia. 2014. MobileMiner: mining your frequent patterns on your phone. In *UbiComp*. ACM, 389–400.

[80] Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize from human feedback. arXiv:2009.01325 https://arxiv.org/abs/2009.01325

[81] Yu Sun, Shuohuan Wang, Yu-Kun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2020. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. In *AAAI*. AAAI Press, 8968–8975.

[82] Jennifer Switzer, Gabriel Marcano, Ryan Kastner, and Pat Pannuto. 2023. Junkyard Computing: Repurposing Discarded Smartphones to Minimize Carbon. In *ASPLOS (2)*. ACM, 400–412.

[83] John Thorpe, Pengzhan Zhao, Jonathan Eyolfson, Yifan Qiao, Zhihao Jia, Minjia Zhang, Ravi Netravali, and Guoqing Harry Xu. 2023. Bamboo: Making Preemptible Instances Resilient for Affordable Training of Large DNNs. In *NSDI*. USENIX Association, 497–513.

[84] Hugo Touvron, Louis Martin, Kevin Stone, and et al. 2023. Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv:2307.09288

[85] UK Government. 2024. AI Foundation Models: initial review. https://www.gov.uk/cma-cases/ai-foundation-models-initial-review.

[86] Marcel Wagenländer, Guo Li, Bo Zhao, Luo Mai, and Peter R. Pietzuch. 2024. Tenplex: Dynamic Parallelism for Deep Learning using Parallelizable Tensor Collections. In *SOSP*. ACM, 195–210.

[87] Guanhua Wang, Heyang Qin, Sam Ade Jacobs, Connor Holmes, Samyam Rajbhandari, Olatunji Ruwase, Feng Yan, Lei Yang, and Yuxiong He. 2023. ZeRO++: Extremely Efficient Collective Communication for Giant Model Training. arXiv:2306.10209

[88] Jaylen Wang, Udit Gupta, and Akshitha Sriraman. 2023. Peeling Back the Carbon Curtain: Carbon Optimization Challenges in Cloud Computing. In *HotCarbon*. ACM, 8:1–8:7.

[89] Jue Wang, Binhang Yuan, Luka Rimanic, Yongjun He, Tri Dao, Beidi Chen, Christopher Ré, and Ce Zhang. 2022. Fine-tuning Language Models over Slow Networks using Activation Compression with Guarantees. arXiv:2206.01299

[90] Xiaofei Wang, Yiwen Han, Victor C. M. Leung, Dusit Niyato, Xueqiang Yan, and Xu Chen. 2020. Convergence of Edge Computing and Deep Learning: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* 22, 2 (2020), 869–904.

[91] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned Language Models are Zero-Shot Learners. In *ICLR*. OpenReview.net.

[92] Alexander Wood, Kayvan Najarian, and Delaram Kahrobaei. 2021. Homomorphic Encryption for Machine Learning in Medicine and Bioinformatics. *ACM Comput. Surv.* 53, 4 (2021), 70:1–70:35.

[93] Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga Behram, Jinshi Huang, Charles Bai,

Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin S. Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim M. Hazelwood. 2022. Sustainable AI: Environmental Implications, Challenges and Opportunities. In *MLSys*. mlsys.org.

[94] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, Zhenda Xie, Yu Wu, Kai Hu, Jiawei Wang, Yaofeng Sun, Yukun Li, Yishi Piao, Kang Guan, Aixin Liu, Xin Xie, Yuxiang You, Kai Dong, Xingkai Yu, Haowei Zhang, Liang Zhao, Yisong Wang, and Chong Ruan. 2024. DeepSeek-VL2: Mixture-of-Experts Vision-Language Models for Advanced Multimodal Understanding. arXiv:2412.10302

[95] Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and Efficient Post-Training Quantization for Large Language Models. In *ICML (Proceedings of Machine Learning Research, Vol. 202)*. PMLR, 38087–38099.

[96] Zongheng Yang, Zhanghao Wu, Michael Luo, Wei-Lin Chiang, Romil Bhardwaj, Woosuk Kwon, Siyuan Zhuang, Frank Sifei Luan, Gautam Mittal, Scott Shenker, and Ion Stoica. 2023. SkyPilot: An Intercloud Broker for Sky Computing. In *20th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2023, Boston, MA, April 17-19, 2023*, Mahesh Balakrishnan and Manya Ghobadi (Eds.). USENIX Association, 437–455. https://www.usenix.org/conference/nsdi23/presentation/yang-zongheng

[97] Shengyuan Ye, Liekang Zeng, Xiaowen Chu, Guoliang Xing, and Xu Chen. 2024. Asteroid: Resource-Efficient Hybrid Pipeline Parallelism for Collaborative DNN Training on Heterogeneous Edge Devices. In *MobiCom*. ACM, 312–326.

[98] Binhang Yuan, Yongjun He, Jared Davis, Tianyi Zhang, Tri Dao, Beidi Chen, Percy Liang, Christopher Ré, and Ce Zhang. 2022. Decentralized Training of Foundation Models in Heterogeneous Environments. In *NeurIPS*.

[99] Han Zhang, Yu Lei, Lin Gui, Min Yang, Yulan He, Hui Wang, and Ruifeng Xu. 2024. CPPO: Continual Learning for Reinforcement Learning with Human Feedback. In *ICLR*. https://openreview.net/forum?id=86zAUE80pP

[100] Li Zhang, Zhe Fu, Boqing Shi, Xiang Li, Rujin Lai, Chenyang Yang, Ao Zhou, Xiao Ma, Shangguang Wang, and Mengwei Xu. 2024. More is Different: Prototyping and Analyzing a New Form of Edge Server with Massive Mobile SoCs. In *USENIX ATC*. USENIX Association, 285–302.

[101] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open Pre-trained Transformer Language Models. arXiv:2205.01068

[102] Yancheng Zhang, Mengxin Zheng, Yuzhang Shang, Xun Chen, and Qian Lou. 2024. HEPrune: Fast Private Training of Deep Neural Networks With Encrypted Data Pruning. In *NeurIPS*.

[103] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. arXiv:2303.18223

[104] Lianmin Zheng, Zhuohan Li, Hao Zhang, Yonghao Zhuang, Zhifeng Chen, Yanping Huang, Yida Wang, Yuanzhong Xu, Danyang Zhuo, Eric P. Xing, Joseph E. Gonzalez, and Ion Stoica. 2022. Alpa: Automating Inter- and Intra-Operator Parallelism for Distributed Deep Learning. In *OSDI*. USENIX Association, 559–578.

[105] Barret Zoph and Quoc V. Le. 2017. Neural Architecture Search with Reinforcement Learning. In *ICLR*. OpenReview.net.