
SUSTAINABLE AI: ENVIRONMENTAL IMPLICATIONS, CHALLENGES AND OPPORTUNITIES

Carole-Jean Wu¹ Ramya Raghavendra¹ Udit Gupta^{1,2} Bilge Acun¹ Newsha Ardalani¹ Kiwan Maeng¹
Gloria Chang¹ Fiona Aga Behram¹ James Huang¹ Charles Bai¹ Michael Gschwind¹ Anurag Gupta¹
Myle Ott¹ Anastasia Melnikov¹ Salvatore Candido¹ David Brooks^{1,2} Geeta Chauhan¹ Benjamin Lee^{1,3}
Hsien-Hsin S. Lee¹ Bugra Akyildiz¹ Max Balandat¹ Joe Spisak¹ Ravi Jain¹ Mike Rabbat¹ Kim Hazelwood¹

ABSTRACT

This paper explores the environmental impact of the super-linear growth trends for AI from a holistic perspective, spanning *Data*, *Algorithms*, and *System Hardware*. We characterize the carbon footprint of AI computing by examining the model development cycle across industry-scale machine learning use cases and, at the same time, considering the life cycle of system hardware. Taking a step further, we capture the operational and manufacturing carbon footprint of AI computing and present an end-to-end analysis for *what* and *how* hardware-software design and at-scale optimization can help reduce the overall carbon footprint of AI. Based on the industry experience and lessons learned, we share the key challenges and chart out important development directions across the many dimensions of AI. We hope the key messages and insights presented in this paper can inspire the community to advance the field of AI in an environmentally-responsible manner.

1 INTRODUCTION

Artificial Intelligence (AI) is one of the fastest growing domains spanning research and product development. The significant investment in AI has also stimulated novel applications of AI in domains such as science, medicine, finance, and education. AI-powered robots are deployed by the farming industry to grow plants using less resources (Sheikh, 2020). AI is applied to tackle the protein structure prediction challenge, which can lead to revolutionary advances for biological sciences (Jumper et al., 2021). AI is also used to discover new electrocatalysts for efficient and scalable ways to store and use renewable energy (Zitnick et al., 2020) while also being applied to predict renewable energy availability in advance to improve energy utilization (Elkin & Witherspoon, 2019). All of these investments in research, development, and deployment have led to a super-linear growth in AI data, models, and infrastructure capacity.

This work explores the environmental impact of AI from a *holistic* perspective. More specifically, we present the challenges and opportunities to designing sustainable AI computing across the key phases of the machine learning (ML) development process — *Data*, *Experimentation*, *Training*, and *Inference* — for a variety of AI use cases at Meta.

¹Meta AI ²Harvard University ³University of Pennsylvania.
Correspondence to: Carole-Jean Wu <carolejeanwu@fb.com>, Ramya Raghavendra <r.rramya@fb.com>.

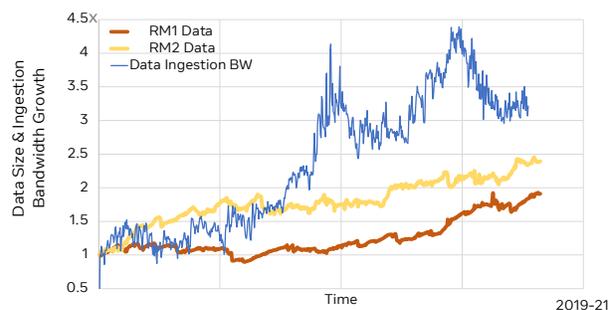


Figure 1. The amount of data for recommendation use cases has roughly doubled, leading to 3.2 times increase in the data ingestion bandwidth demand (2019-21). The ingestion bandwidth here refers to the I/O bandwidth requirement of the data ingestion stage.

The solution space spans across our fleet of datacenters and on-device computing. Given particular use cases, we consider the impact of AI *data*, *algorithms*, and *system hardware*. Finally, we consider emissions across the life cycle of hardware systems, from manufacturing to operational use.

AI Data Growth. In the past decade, we have seen an exponential increase in AI training data and model capacity. Figure 1 illustrates that the amount of training data at Meta for two recommendation use cases — one of the fastest growing areas of ML usage at Meta (Gupta et al., 2020). Data has increased by 2.4 \times and 1.9 \times in the last two years, reaching exabyte scale. The data size increase has led to a 3.2 \times increase in the data ingestion bandwidth demand. Given this increase, data storage and the ingestion pipeline

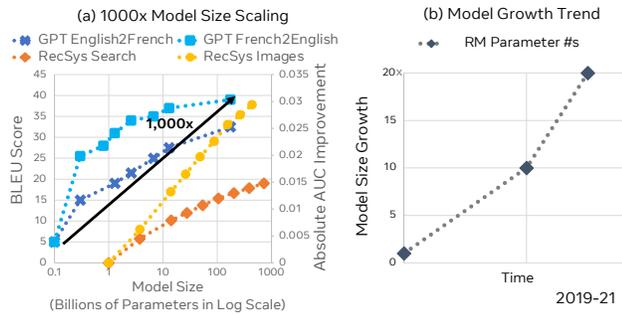


Figure 2. (a) The $1000\times$ model size growth has led to higher model accuracy for various ML tasks, e.g. BLEU score improvement from 5 to 40 for GPT3. Similar model size increase is evident for RecSys-Search and RecSys-Image — ML tasks that implement recommendation models as the backend. (b) To learn valuable information from the data, model capacities have also increased to achieve higher model quality. Meta’s recommendation and ranking model sizes have increased by 20 times (2019-21).

accounts for a significant portion of the infrastructure and power capacity compared to ML training and end-to-end machine learning life cycles (Zhao et al., 2021).

AI Model Growth. The ever-increasing data volume has driven a super-linear trend in model size growth. Figure 2(a) depicts the $1000\times$ model size increase for GPT3-based language translation tasks (Brown et al., 2020; Hernandez & Brown, 2020a) whereas for Baidu’s search engine, the model of $1000\times$ larger in size improves accuracy in AUC by 0.030. Despite small, the accuracy improvement leads to significantly higher-quality search outcomes because deep learning techniques can consider the rich context in search queries effectively. Similarly, Figure 2(b) illustrates that between 2019 and 2021, the size of recommendation models at Meta has increased by $20\times$ into the terabyte scale (Lui et al., 2021). Despite the large model size growth, the memory capacity of GPU-based accelerators, e.g. 32GB (NVIDIA V100, 2018) to 80GB (NVIDIA A100, 2021), has increased by $< 2\times$ every 2 years. The resource requirements for strong AI scaling clearly outpaces that of system hardware.

AI Infrastructure Growth. The strong performance scaling demand for training motivates a variety of *scale-out* solutions (Mudigere et al., 2021; Rajbhandari et al., 2021) by leveraging parallelism at scale with a massive collection of accelerators. The explosive growth in AI use cases at Meta has driven $2.9\times$ increase in AI training infrastructure capacity over the 1.5 years. In addition, we observe trillions of inference per day across Meta’s data centers—more than doubling in the past 3 years. The increase in inference demands has also led to an $2.5\times$ increase in infrastructure capacity. Last but not least, the carbon footprint of AI goes beyond its *operational* energy use. The *embodied* carbon footprint of systems is becoming a growing factor for AI’s overall environmental impact (Gupta et al., 2021, Section 3).

The Elephant in the Room. Despite the positive societal benefits (Tomaev et al., 2020), the endless pursuit of achieving higher model quality has led to the exponential scaling of AI with significant energy and environmental footprint implications. Although recent work shows the carbon footprint of training one large ML model, such as *Meena* (Patterson et al., 2021), is equivalent to 242,231 miles driven by an average passenger vehicle (EPA, 2021), this is only one aspect; to fully understand the real environmental impact we must consider the AI ecosystem *holistically* going forward — beyond looking at model training alone and by accounting for both *operational* and *embodied carbon footprint* of AI. We must look at the ML pipeline end-to-end: data collection, model exploration and experimentation, model training, model optimization and run-time inference. The *frequency of training* and *scale* of each stage of the ML development cycle matter. From the system’s perspective, the life cycle of ML software and system hardware, including manufacturing and operational use, must also be considered.

Optimizing across ML pipelines and systems life cycles end-to-end is a complex and challenging task. While training large, sparsely-activated neural networks improves model scalability, achieving higher accuracy at lower operational energy footprint, it can incur higher embodied carbon footprint from the increase in the system resource requirement. Shifting model training and inference to data centers with carbon-free energy can reduce emissions; however, this approach may not scale to a broad set of use cases. Infrastructure for carbon-free energy is limited by factors such as geography and available materials (e.g. rare metals), and takes significant economic resources and time to build. In addition, as on-device learning becomes more ubiquitously adopted to improve data privacy, we can see more computation being shifted away from data centers to the edge, where access to renewable energy is limited.

A Holistic Approach. This paper is the first to take a holistic approach to characterize the environmental footprint of AI computing from *experimentation* and *training* to *inference*. We quantify the carbon footprint of AI by examining the model development cycle across industry-scale machine learning use cases at Meta (Section 2). We illustrate more than $800\times$ operational carbon footprint reduction achieved through iterative hardware-software co-design for a Transformer-based universal language model. Taking a step further, we present an end-to-end analysis for both operational *and* embodied carbon footprint for AI training and inference (Section 3). Based on the industry experience and lessons learned, we chart out opportunities and development directions across the key dimensions of AI — data, algorithm, systems, metrics, standards, and best practices (Section 4). We hope the key messages (Section 5) and the insights in this paper can inspire the community to advance the field of AI in an environmentally-responsible manner.

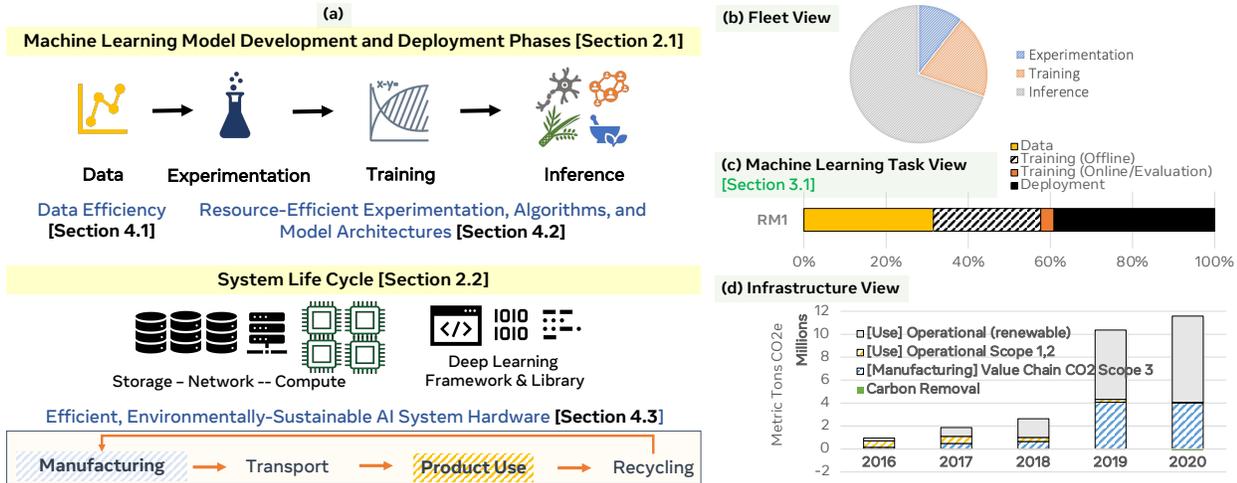


Figure 3. (a) Model Development Phases over AI System Hardware Life Cycle. (b) At Meta, we observe a rough power capacity breakdown of **10:20:70** for AI infrastructures devoted to the three key phases — **Experimentation**, **Training**, and **Inference**; (c) Considering the primary stages of the machine learning pipeline end-to-end, the energy footprint of RM1 is roughly **31:29:40** over **Data**, **Experimentation/Training**, and **Inference**; (d) Despite the investment to neutralize the operational footprint with carbon-free energy, the overall data center electricity use continues to grow, demanding over 7.17 million MWh in 2020 (Meta, 2021).

2 MODEL DEVELOPMENT PHASES AND AI SYSTEM HARDWARE LIFE CYCLE

Figure 3(a) depicts the major development phases for ML — **Data Processing**, **Experimentation**, **Training**, and **Inference** (Section 2.1) — over the life cycle of AI system hardware (Section 2.2). Driven by distinct objectives of AI research and advanced product development, infrastructure is designed and built specifically to maximize data storage and ingestion efficiency for the phase of **Data Processing**, developer efficiency for the phase of **Experimentation**, training throughput efficiency for the phase of **Training**, and tail-latency bounded throughput efficiency for **Inference**.

2.1 Machine Learning Model Development Cycle

ML researchers extract features from data during the **Data Processing** phase and apply weights to individual features based on feature importance to the model optimization objective. During **Experimentation**, the researchers design, implement and evaluate the quality of proposed algorithms, model architectures, modeling techniques, and/or training methods for determining model parameters. This model exploration process is computationally-intensive, where a large collection of diverse ML ideas are explored simultaneously at-scale. Thus, during this phase, we observe diverse system resource requirements from the large pool of training experiments. Within Meta’s ML research cluster, 50% (p50) of ML training experiments take up to 1.5 GPU days while 99% (p99) of training experiments complete within 24 GPU days. There are a number of large-scale, trillion parameter models which require over 500 GPU days.

Once a ML solution is determined as promising, it moves into **Training** where the ML solution is evaluated using ex-

tensive production data — data that is *more recent*, is *larger in quantity*, and contains *richer features*. The process often requires additional hyper-parameter tuning. The majority of data ingestion at Meta is from model training. This has motivated a redesign of the warehouse-scale database and data ingestion pipeline specific for ML. Figure 3(c) illustrates that the power footprint of data ingestion is already higher than that of model training for certain large-scale ML tasks.

Depending on the ML task requirement, the models can be trained/re-trained at different frequencies. For example, models supporting Meta’s *Search* service were trained at an hourly cadence whereas the *Language Translation* models were trained weekly. A p50 production model training workflow takes 2.96 GPU days while a training workflow at p99 can take up to 125 GPU days. Finally, for **Inference**, the best-performing model is deployed for production ML use cases, producing trillions of daily predictions to serve billions of users on the Meta platform. The total compute cycles for inference predictions are expected to exceed the corresponding training cycles for the deployed model.

2.2 Machine Learning System Life Cycle

Life Cycle Analysis (LCA) is a commonly-used methodology to assess the carbon emissions over a product’s life cycle. There are four major phases: *manufacturing*, *transport*, *product use*, and *recycling*. In this work, we consider the overall carbon footprint of AI by including *manufacturing* — carbon emissions from building infrastructures specifically for AI (i.e., *embodied carbon footprint*) and *product use* — carbon emissions from the use of AI (i.e., *operational carbon footprint*).

While quantifying the exact breakdown between operational

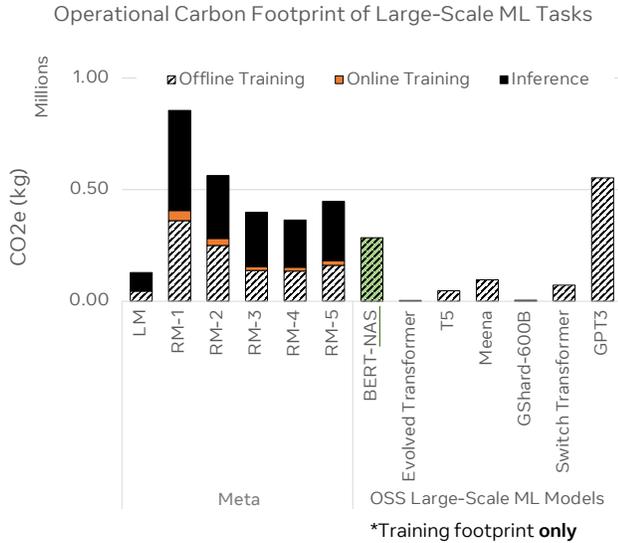


Figure 4. The carbon footprint of the LM model is dominated by Inference whereas, for RM1 – RM5, the carbon footprint of Training versus Inference is roughly equal. The average carbon footprint for ML training tasks at Meta is 1.8 times larger than that of Meena used in modern conversational agents and 0.3 times of GPT-3’s carbon footprint. Carbon footprint for inference tasks is included for models that are used in production. Note: the operational carbon footprint of AI does not correlate with the number of model parameters. The open-source large-scale ML tasks (OSS) are based on the vanilla model architectures and may not be reflective of production use cases.

and embodied carbon footprint is a complex process, we estimate the significance of embodied carbon emissions using Meta’s Greenhouse Gas (GHG) emission statistics. *More than 50% of Meta’s emissions owe to its value chain – Scope 3 of Meta’s GHG emission.* As a result, a significant embodied carbon cost is paid upfront for every system component brought into Meta’s fleet of data centers, where AI is the biggest growth driver.

3 AI COMPUTING’S CARBON FOOTPRINT

3.1 Carbon Footprint Analysis for Industry-Scale ML Training and Deployment

Figure 4 illustrates the operational carbon emissions for model training and inference across the ML tasks. We analyze six representative machine learning models in production at Meta¹. LM refers to Meta’s Transformer-based Universal Language Model for text translation (Conneau et al., 2020). RM1 – RM5 represent five unique deep learning recommendation and ranking models for various Meta products (Naumov et al., 2019; Gupta et al., 2020).

¹In total, the six models account for a vast majority of compute resources for the overall inference predictions at Meta, serving billions of users world wide.

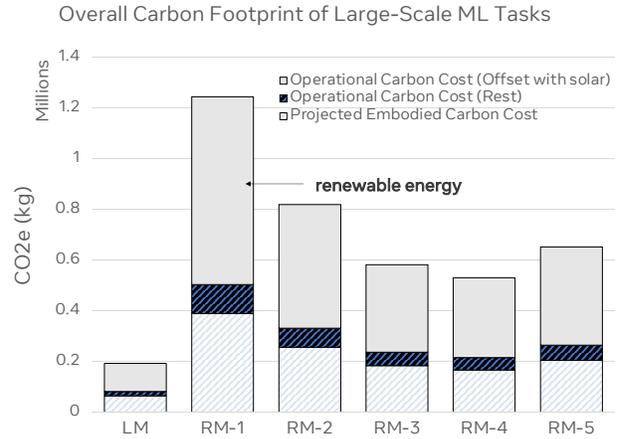


Figure 5. When considering the overall life cycle of ML models and systems in this analysis, manufacturing carbon cost is roughly 50% of the (location-based) operational carbon footprint of large-scale ML tasks (Figure 4). Taking into account carbon-free energy, such as solar, the operational energy consumption can be significantly reduced, leaving the manufacturing carbon cost as the dominating source of AI’s carbon footprint.

We compare the carbon footprint of Meta’s production ML models with seven large-scale, open-source (OSS) models: BERT-NAS, T5, Meena, GShard-600B, Switch Transformer, and GPT-3. Note, we present the operational carbon footprint of the OSS model training from (Strubell et al., 2019; Patterson et al., 2021). The operational carbon footprint results can vary based on the exact AI systems used and the carbon intensity of the energy mixture. Models with more parameters do not necessarily result in higher carbon emissions (Artetxe et al., 2021). Training the Switch Transformer model equipped with 1.5 trillion parameters (Fedus et al., 2021) produces significantly less carbon emission than that of GPT-3 (750 billion parameters) (Brown et al., 2020). This illustrates the carbon footprint advantage of operationally-efficient model architectures.

Both Training and Inference can contribute significantly to the overall carbon footprint of machine learning tasks at Meta. The exact breakdown between the two phases varies across ML use cases.

The overall operational carbon footprint is categorized into *offline training*, *online training*, and *inference*. Offline training encompasses both experimentation and training models with historical data. Online training is particularly relevant to recommendation models where parameters are continuously updated based on recent data. The inference footprint represents the emission from serving production traffic. The online training and inference emissions are considered over the period of offline training. For recommendation use cases, we find the carbon footprint is split evenly between training and inference. On the other hand, the carbon footprint of LM is dominated by the inference phase, using much higher

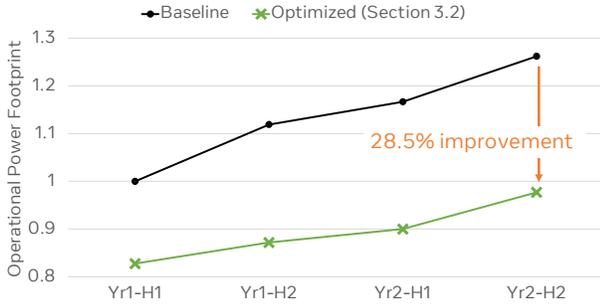


Figure 6. The iterative optimization process has led to 28.5% operational energy footprint reduction over the two-year time period (Section 3.2). Despite the significant operational power footprint reduction, we continue to see the overall electricity demand for AI to increase over time — an example of *Jevon’s Paradox*, where efficiency improvement stimulates additional novel AI use cases.

inference resources (65%) as compared to training (35%).

Operational Carbon Footprint: Across the life cycle of the six Meta models in Figure 4, the average carbon footprint is $1.8\times$ higher than that of the open-source Meena model (Adiwardana & Luong, 2020) and one-third of GPT-3’s training footprint. To quantify the emissions of Meta’s models, we measure the total energy consumption, assume location-based carbon intensities for energy mixes, and use Power Usage Effectiveness (PUE) of 1.1. In addition to model- and hardware-level optimizations, Meta’s renewable energy procurement programs mitigates these emissions.

Embodied Carbon Footprint: To quantify the embodied carbon footprint of AI hardware, we use LCA (Section 2.2). We assume GPU-based AI training systems have similar embodied footprint as the production footprint of Apple’s 28-core CPU with dual AMD Radeon GPUs (Apple, 2019)². For CPU-only systems we assume half the embodied emissions. Based on the characterization of model training and inference at Meta, we assume an average utilization of 30-60% over the 3- to 5-year lifetime for servers. The embodied carbon footprint is estimated as follows:

$$CO_2^{embodied} = \sum_i \frac{\text{Time}}{\text{Lifetime}} CO_2^{embodied}(AI_{System})(i) \quad (1)$$

Figure 5 presents the carbon footprint estimation for the large scale ML tasks at Meta, spanning both operational and embodied carbon footprint. Based on the assumptions of location-based renewable energy availability, the split between the embodied and (location-based) operational carbon footprint is roughly 30% / 70% for the large scale ML tasks. Taking into account carbon-free energy, such as solar, the operational carbon footprint can be significantly reduced,

²Ideally, the embodied carbon analysis should be based on the LCA for the AI system hardware. However, LCA for GPUs and domain-specific accelerators is limited — an important future direction to enabling environmentally-sustainable AI systems.

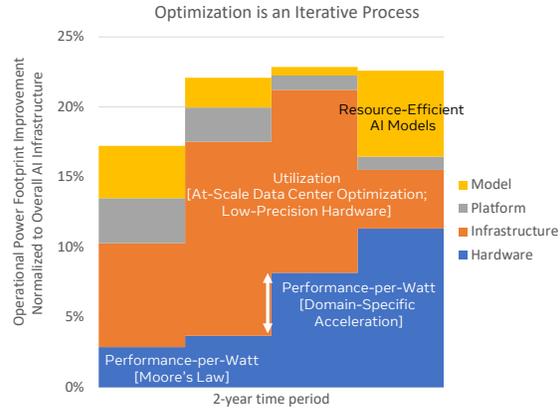


Figure 7. Optimization is an iterative process — we have achieved an average of 20% operational energy footprint reduction every 6 months across the machine learning hardware-software stack.

leaving the manufacturing carbon cost as the dominating source of AI’s carbon footprint. The operational carbon footprint (rest) portion of the bars represent the carbon intensity of renewable energy which has been further mitigated by Meta’s carbon removal programs.

3.2 Carbon Footprint Optimization from Hardware-Software Co-Design

Optimization is an iterative process — we reduce the power footprint across the machine learning hardware-software stack by 20% every 6 months. But at the same time, AI infrastructure continued to scale out. The net effect, with Jevon’s Paradox, is a 28.5% operational power footprint reduction over two years (Figure 6).

Optimization across AI Model Development and System Stack over Time: Figure 7 shows the operational power footprint reduction across Meta’s AI fleet over two years. The improvement come from four areas of optimizations: *model* (e.g., designing resource-efficient models), *platform* (e.g., PyTorch’s support for quantization), *infrastructure* (e.g., data center optimization and low-precision hardware), and *hardware* (e.g., domain-specific acceleration). Each bar illustrates the operational power reduction across Meta’s AI fleet over 6-month period from each of the optimization areas. The optimizations in aggregate provide, on average, a 20% reduction in operational power consumption every six months. The compounded benefits highlight the need for cross-stack optimizations.

Optimizing the Carbon Footprint of LM: Figure 8 shows a deep dive into a specific, yet important, machine learning task at Meta: language translation using a Transformer-based architecture (LM). LM is designed based on the state-of-the-art cross-lingual understanding through self-supervision. Figure 8 analyzes the power footprint improvements over a collection of optimization steps for LM: *platform-level caching, GPU acceleration, low precision*

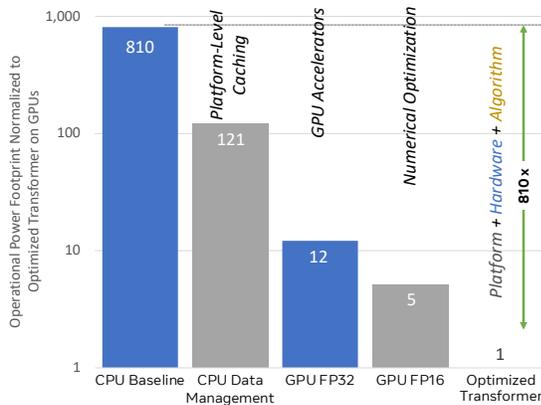


Figure 8. For the cross-lingual ML task (LM), the operational energy footprint can be significantly reduced by more than $800\times$ using *platform-level caching*, *GPUs*, *low precision data format*, and *additional algorithmic optimization*.

format on accelerator, and *model optimization*. In aggregate the optimizations reduce the infrastructure resources required to serve LM at scale by over $800\times$. We outline the optimization benefits from each area below.

Platform-Level Caching. Starting with a CPU server baseline, application-level caching improves power efficiency by $6.7\times$. These improvements are a result of pre-computing and caching frequently accessed embeddings for language translation tasks. Using DRAM and Flash storage devices as caches, these pre-computed embeddings can be shared across applications and use cases.

GPU acceleration. In addition to caching, deploying LM across GPU-based specialized AI hardware unlocks an additional $10.1\times$ energy efficiency improvement.

Algorithmic optimization. Finally, algorithmic optimizations provide an additional $12\times$ energy efficiency reduction. Halving precision, from 32-bit to 16-bit operations, provides a $2.4\times$ energy efficiency improvement on GPUs. Another $5\times$ energy efficiency gain can be achieved by using custom operators to schedule encoding steps within a single kernel of the Transformer module, such as (NVIDIA, 2021b).

Optimizing the Carbon Footprint of RMs: A major infrastructure challenge faced by deep RM training and deployment (RM1 – RM5) is the fast-rising memory capacity and bandwidth demands. There are two primary sub-nets in a RM: compute-intensive, dense fully-connected network and the memory capacity and bandwidth-intensive, sparse embedding-based network. For a number of important recommendation and ranking use cases, embedding operations dominate model execution time (Gupta et al., 2020; Ke et al., 2020; Wilkening et al., 2021; Sethi et al., 2022).

To tackle the significant memory capacity and bandwidth requirement, we deploy model quantization for RMs (Deng

et al., 2021). Quantization offers two primary efficiency benefits: the low-precision data representation reduces the amount of computation requirement and, at the same time, lowers the overall memory capacity need. By converting 32-bit floating-point numerical representation to 16-bit, we can reduce the overall RM2 model size by 15%. This has led to 20.7% reduction in memory bandwidth consumption. Furthermore, the memory capacity reduction enabled by quantization unblocks novel systems with lower on-chip memory. For example, for RM1, quantization has enabled RM deployment on highly power-efficient systems with smaller on-chip memory, leading to an end-to-end inference latency improvement of 2.5 times.

3.3 Machine Learning Infrastructures at Scale

ML Accelerators: GPUs are the de-facto training accelerators at Meta, contributing to significant power capacity investment in the context of Meta’s data centers. However, GPUs can be severely under-utilized during both the ML Experimentation and Training phases (Wesolowski et al., 2021). To amortize the upfront embodied carbon cost of every accelerator deployed into Meta’s datacenters, maximizing accelerator utilization is a must.

At-Scale Efficiency Optimization: Servers in Meta data center fleets are customized for internal workloads only — machine learning tasks (Hazelwood et al., 2018) and others (Sriraman et al., 2019; Sriraman & Dhanotia, 2020). Compared to public cloud providers, this puts Meta at a unique position for at-scale resource management design and optimization. First, Meta customizes server SKUs — compute, memcached, storage tiers and ML accelerators — to maximize performance and power efficiency.

Furthermore, the large-scale deployment of servers of different types provides an opportunity to build performance measurement and optimization tools to ensure high utilization of the underlying infrastructure. For data center fleets in different geographical regions where the actual server utilization exhibits a diurnal pattern, Auto-Scaling frees the over-provisioned capacity during off-peak hours, by up to 25% of the web tier’s machines (Tang et al., 2020). By doing so, it provides opportunistic server capacity for others to use, including offline ML training. Furthermore, static power consumption plays a non-trivial role in the context of the overall data center electricity footprint. This motivates more effective processor idle state management.

Finally, large data center operators have invested in carbon free energy sources to neutralize its operational carbon footprint. Reaching net zero emissions entails matching every unit of energy consumed by data centers with 100% renewable energy purchased. Remaining emissions are often offset with various sustainability programs, further reducing the operational carbon footprint of AI computing.

3.4 Going Beyond Efficiency Optimization

Despite the opportunities for optimizing energy efficiency and reducing environmental footprint at scale, there are many reasons why we must care about scaling AI in a more environmentally-sustainable manner. AI growth is multiplicative beyond current industrial use cases. Although domain-specific architectures improve the operational energy footprint of AI model training by more than 90% (Patterson et al., 2021), these architectures require more system resources, leading to larger embodied carbon footprints.

While shifting model training and inference to data centers with carbon-free energy sources can reduce emissions, the solution may not scale to all AI use cases. Infrastructure for carbon free energy is limited by rare metals and materials, and takes significant economic resources and time to build. Furthermore, the carbon footprint of federated learning and optimization use cases at the edge is estimated to be similar to that of training a *Transformer_{Big}* model (Figure 11). As on-device learning becomes more ubiquitous to improve data privacy, we expect to see more computation being shifted away from data centers to the edge, where access to renewable energy may be limited. The edge-cloud space for AI opens interesting design opportunities (Section 4.3).

The growth of AI in all dimensions outpaces the efficiency improvement at-scale. Embodied carbon cost is becoming the dominating source of AI’s overall carbon footprint. To curb the rising carbon footprint of AI at-scale, *we must look beyond efficiency optimization and complement efficiency and utilization optimization with efforts to tackle the remaining embodied carbon footprint of AI systems.*

4 A SUSTAINABILITY MINDSET FOR AI

To tackle the environmental implications of AI’s exponential growth, the first key step requires ML practitioners and researchers to develop and adopt an *sustainability mindset*. The solution space is wide open—while there are significant efforts looking at *AI system and infrastructure efficiency optimization, the AI data, experimentation, and training algorithm efficiency space* (Sections 4.1 and 4.2) beyond system design and optimization (Section 4.3) is less well explored. We cannot optimize what cannot be measured—telemetry to track the carbon footprint of AI technologies must be adopted by the community (Section 4.4). We identify a number of important directions to scale AI in a *sustainable* manner and to minimize the environmental impact of AI for the next decades.

The field of AI is currently primarily driven by research that seeks to maximize model accuracy — *progress* is often used synonymously with improved prediction quality. This has led to a decade of AI achieved through the use of massive computational power while disregarding resources or environmental footprint. To develop AI technologies re-

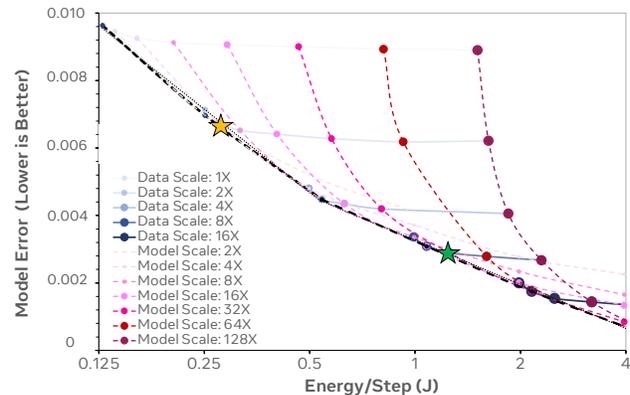


Figure 9. Model quality of recommendation use cases improves as we scale up the amount of data and/or the number of model parameters (e.g., embedding dimension), leading to higher energy and carbon footprint. Maximizing model accuracy for the specific recommendation use case comes with significant energy cost — Roughly 4× energy saving can be achieved with only 0.004 model quality degradation (green vs. yellow stars).

sponsibly, *we must achieve competitive model accuracy at a fixed or even reduced computational and environmental cost.* Despite the recent calls-to-action (Strubell et al., 2019; Lacoste et al., 2019; Henderson et al., 2020; Bender et al., 2021; Patterson et al., 2021), the overall community remains under-invested in research that aims at deeply understanding and minimizing the cost of AI. To bend the exponential growth curve of AI and its environmental footprint, we must build a future, where efficiency is an evaluation criterion for publishing ML research on computationally-intensive models beyond accuracy-related measures.

4.1 Data Utilization Efficiency

Data Scaling and Sampling: *No data is like more data* — data scaling is the de-facto approach to increase model quality, where the primary factor for accuracy improvement is driven by the size and quality of training data, instead of algorithmic optimization. However, data scaling has significant environmental footprint implications. To keep the model training time manageable, overall system resources must be scaled with the increase in the data set size, resulting in larger embodied carbon footprint and operational carbon footprint from the data storage and ingestion pipeline and model training. Alternatively, if training system resources are kept fixed, data scaling increases training time, resulting in a larger operational energy footprint.

Figure 9 depicts energy footprint reduction potential when data and model scaling is performed in conjunction. The x-axis represents the energy required per training step whereas the y-axis represents model error. The **blue** solid lines capture model size scaling (through embedding hash scaling) while the training data set size is kept fixed. The **red** dashed

lines capture data scaling while the model size is kept fixed. The dashed black line captures the performance scaling trend as we scale data and model sizes in tandem. This represents the energy-optimal scaling approach. Scaling data sizes or model sizes independently deviates from the energy-optimal trend. We highlight two energy-optimal settings along the Pareto-frontier curve. The yellow star uses the scaling setting of *Data scaling* $2\times$ and *Model scaling* $2\times$ whereas the green star adopts the setting of *Data scaling* $8\times$ and *Model scaling* $16\times$. The yellow star consumes roughly $4\times$ lower energy as compared to the green star with only 0.004 model quality degradation. Overall model quality performance has a (diminishing) power-law relationship with the corresponding energy consumption and the power of the power law is extremely small (0.002-0.004). This means achieving higher model quality through model-data scaling for recommendation use cases incurs significant energy cost.

When designed well, however, data scaling, sampling and selection strategies can improve the competitive analysis for ML algorithms, reducing the environmental footprint of the process. For instance, Sachdeva et al. (2021) demonstrated that intelligent data sampling with merely 10% of data sub-samples can effectively preserve the relative ranking performance of different recommendation algorithms. This ranking performance is achieved with an average of 5.8 times execution time speedup, leading to significant operating carbon footprint reduction.

Data Perishability: Understanding key characteristics of data is fundamental to efficient data utilization for AI applications. *Not all data is created equal* and data collected over time loses its predictive value gradually. Understanding the rate at which data loses its predictive value has strong implications on the resulting carbon footprint. For example, natural language data sets can lose half of their predictive value in the time period of less than 7 years (the half-life time of data) (Valavi et al., 2020). The exact half-life period is a function of context. If we were able to predict the half-life time of data, we can devise effective sampling strategies to subset data at different rates based on its half-life. By doing so, the resource requirement for the data storage and ingestion pipeline can be significantly reduced — lower training time (operational carbon footprint) as well as storage needs (embodied carbon footprint).

4.2 Efficiency in Experimentation and Training

The experimentation and training phases are closely coupled (Section 2). There is a natural trade-off between the investment in experimentation and the subsequent training cost (Section 3). **Neural architecture search (NAS) and hyperparameter optimization (HPO)** are techniques that automate the design space exploration. Despite their capability to discover higher-performing neural networks, NAS and HPO can be extremely resource-intensive, involving

training many models, especially when using simple approaches. Strubell et al. (2019) show that grid-search NAS can incur over $3000\times$ environmental footprint overhead. Utilizing much more sample-efficient NAS and HPO methods (Turner et al., 2021; Ren et al., 2021) can translate directly into carbon footprint improvement. In addition to reducing the number of training experiments, one can also reduce the training time of each experiment. By detecting and *stopping under-performing training workflows early*, unnecessary training cycles can be eliminated.

Multi-objective optimization explores the Pareto frontier of efficient model quality and system resource trade-offs. If used early in the model exploration process, it enables more informed decisions about *which* model to train fully and deploy given certain infrastructure capacity. Beyond model accuracy and timing performance (Joglekar et al., 2020; Song et al., 2020; Tan & Le, 2020; Eriksson et al., 2021), energy and carbon footprint can be directly incorporated into the cost function as optimization objectives (Yang et al., 2021) to enable discovery of environmentally-friendly models. Furthermore, when training is decoupled from NAS, sub-networks tailoring to specialized system hardware can be selected *without additional training* (Stamoulis et al., 2019; Cai et al., 2020a; Chen et al., 2021; Mellor et al., 2021). Such approaches can significantly reduce the overall training time, however, at the expense of increased embodied carbon footprint.

Developing **resource-efficient model architectures** fundamentally reduce the overall system capacity need of ML tasks. From the systems perspective, accelerator memory is scarce. However, DNNs, such as neural recommendation models, require significantly higher memory capacity and bandwidth (Ke et al., 2020; Acun et al., 2021). This motivates researchers to develop memory-efficient model architectures. For example, the Tensor-Train compression technique (TT-Rec) achieves more than $100\times$ memory capacity reduction with negligible training time and accuracy trade-off (Yin et al., 2021). Similarly, the design space trade-off between memory capacity requirement, training time, and model accuracy is also explored in Deep Hash Embedding (DHE) (Kang et al., 2021). While training time increases lead to higher operational carbon footprint, in the case of TT-Rec and DHE, the memory-efficient model architectures require significantly lower memory capacity while better utilizing the computational capability of training accelerators, resulting in lower embodied carbon footprint.

Developing **efficient training algorithms** is a long-time objective of research in optimization and numerical methods (Nemirovskij & Yudin, 1983). Evaluations of optimization methods should account for *all* experimentation efforts required to tune optimizer hyperparameters, not just the method performance after tuning (Choi et al., 2019;

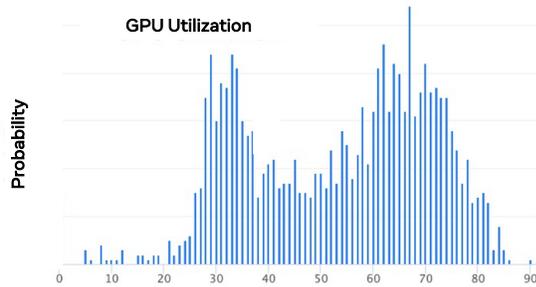


Figure 10. A vast majority of model experimentation (over tens of thousands of training workflows) utilizes GPUs at only 30-50%, leaving room for utilization and efficiency improvements.

Sivaprasad et al., 2020). In addition, significant research has gone into algorithmic approaches to efficiently scale training (Goyal et al., 2017; Ott et al., 2018) by reducing communication cost via compression (Alistarh et al., 2017; Vogels et al., 2019), pipelining (Huang et al., 2019), and sharding (Rajbhandari et al., 2020; Rasley et al., 2020). The advances have enabled efficient scaling to larger models and larger datasets. We expect efficient training methods to continue as an important domain. While this paper has focused on supervised learning relying labeled data, algorithmic efficiency extends to other learning paradigms including self-supervised and semi-supervised learning. We discuss promising efficiency scaling opportunities for self- and semi-supervised learning in Appendix B.

4.3 Efficient and Environmentally-Sustainable AI Infrastructure and System Hardware

To amortize the embodied carbon footprint, model developers and system architects must *maximize the utilization of accelerator and system resources* when in use and *prolong the lifetime of AI infrastructures*. Existing practices such as the move to domain-specific architectures at cloud scale (Jouppi et al., 2017; Hamilton, 2018; Azure, 2019) reduce AI computing’s footprint by consolidating computing resources at scale and by operating the shared infrastructures more environmentally-friendly with carbon free energy³.

Accelerator Virtualization and Multi-Tenancy Support:

Figure 10 illustrates the utilization of GPU accelerators in Meta’s research training infrastructure. A significant portion of machine learning model experimentation utilizes GPUs at only 30-50%, leaving significant room for improvements to efficiency and overall utilization. Virtualization and workload consolidation technologies can help maximize accelerator utilization (NVIDIA, 2021a). Google’s TPUs have also recently started supporting virtualization (Spiridonov, 2021). Multi-tenancy for AI accelerators is gaining traction as an effective way to improve resource utilization,

³We discuss additional directions for enabling environmentally-sustainable systems in Appendix A: implications of general-purpose processors, accelerators, reconfigurable hardware for AI; infrastructure disaggregation; fault tolerant, resilient AI systems.

thereby amortizing the upfront embodied carbon footprint of customized system hardware for AI at the expense of potential operational carbon footprint increase (Gschwind et al., 2017; Jeon et al., 2019; Yu & Chowdhury, 2019; Ghodrati et al., 2020; Kao & Krishna, 2021).

Environmental Sustainability as a Key AI System Design Principle:

Today, servers are designed to optimize performance and power efficiency. However, system design with a focus on operational energy efficiency optimization does not always produce the most environmentally-sustainable solution (Jain & Wullert, 2002; Chang et al., 2010; Gupta et al., 2021). With the rising embodied carbon cost and the exponential demand growth of AI, system designers and architects must re-think fundamental system hardware design principles to minimize computing’s footprint end-to-end, considering the entire hardware and ML model development life cycle. In addition to the respective performance, power, and cost profiles, the environmental footprint characteristics of processors over the generations of CMOS technologies, DDRx and HBM memory technologies, SSD/NAND-flash/HDD storage technologies can be orders-of-magnitude different (Bardon et al., 2020). Thus, designing AI systems with the least environmental impact requires explicit consideration of environmental footprint characteristics at the design time.

Carbon-Efficient Scheduling for AI Computing At-Scale:

As the electricity consumption of hyperscale data centers continues to rise, data center operators have devoted significant investment to neutralize operational carbon footprint. By operating large-scale computing infrastructures with carbon free energy, technology companies are taking an important step to address the environmental implications of computing. *More can be done however.*

As the renewable energy proportion in the electricity grid increases, fluctuations in energy generation will increase due to the intermittent nature of renewable energy sources (i.e. wind, solar). Elastic carbon-aware workload scheduling techniques can be used in and across datacenters to predict and exploit the intermittent energy generation patterns (Radovanovic et al., 2021). However such scheduling algorithms might require server over-provisioning to allow for flexibility of shifting workloads to times when carbon-free energy is available. Furthermore, any additional server capacity comes with manufacturing carbon cost which needs to be incorporated into the design space. Alternatively, energy storage (e.g. batteries, pumped hydro, flywheels, molten salt) can be used to store renewable energy during peak generation times for use during low generation times. There is an interesting design space to achieve carbon-efficient AI computing (Acun et al., 2022).

On-Device Learning is becoming more ubiquitous to enable model personalization (Bonawitz et al., 2019; Wang

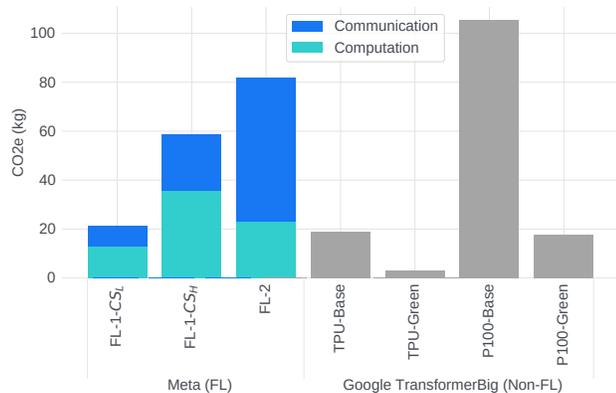


Figure 11. Federated learning can result in a non-negligible amount of carbon emissions, comparable to the carbon footprint of training *TransformerBig*.

et al., 2019b; Cai et al., 2020b) while improving data privacy (Hard et al., 2018; Yang et al., 2018; Ramaswamy et al., 2019; Huba et al., 2022), yet its impact in terms of carbon emission is often overlooked. Especially, federated learning (FL), which trains local models on client devices and periodically aggregates the model parameters for a global model without collecting raw user data, is becoming widely adopted in industry for user privacy (Hard et al., 2018; Yang et al., 2018; Ramaswamy et al., 2019; Huba et al., 2022). The FL process can emit non-negligible carbon at the edge due to both computation and wireless communication.

To estimate the carbon emission of FL, we analyzed data of representative federated learning application use cases at Meta, such as (Huba et al., 2022). First, energy consumption of FL is estimated based on the time spent on FL computation, the number of participants on each model aggregation round (i.e. cohort sizes), the number of rounds to model convergence, the model sizes, and a carbon intensity of 0.429 kilogram CO₂e per KWh (representative of the average energy mix of the United States).

Computation energy is estimated as the product of device power and the computation time of each client. For device power, we used 3W (Wang et al., 2019a).

Communication energy is modeled as the sum of energy spent by the client device and energy spent by the communication infrastructure, including the Wi-Fi access point, edge router, core router, broadband network gateway, and Ethernet switch (Vishwanath et al., 2015). Appendix C provides further methodology detail.

Model training on client edge devices is inherently less energy-efficient because of the high wireless communication overheads, sub-optimal training data distribution in individual client devices, large degree of system heterogeneity among client edge devices, and highly-fragmented edge device architectures that make system-level optimization significantly more challenging (Wu et al., 2019). Figure 11

compares the carbon footprint of two FL applications (FL-1; FL-2) at Meta with that of training *TransformerBig* (Patterson et al., 2021) on a centralized server. FL-1-*CS_L* and FL-1-*CS_H* represent two different configurations of FL-1: FL-1-*CS_L* corresponds to a smaller cohort size whereas FL-1-*CS_H* a larger cohort size. FL-1-*CS_H* converges roughly 2× faster but results in higher energy consumption, as large cohort training is known to be less efficient (Charles et al., 2021). P100-Base represents the carbon footprint of *TransformerBig* training on P100 GPU whereas TPU-Base is *TransformerBig* training on TPU. P100-Green and TPU-Green consider renewable energy at the cloud. The figure shows that FL can result in a non-negligible amount of carbon emissions comparable to that of training a much larger model on a centralized setup, i.e., *TransformerBig*. Also, the computation and communication energy consumption ratios differ between the use cases. Communication energy dominates for FL-2 use cases while computation energy dominates for FL-1.

It is important to reduce AI’s environmental footprint at the edge. With the ever-increasing demand for on-device use cases over billions of client devices, such as teaching AI to understand the physical environment from first-person perception (Grauman et al., 2021) or personalizing AI tasks, the carbon footprint for on-device AI can add up to a dire amount quickly. Also, renewable energy is far more limited for client devices compared to data centers. Optimizing the overall energy efficiency of FL and on-device AI is an important first step (Kang et al., 2017; Stamoulis et al., 2018; Kim & Wu, 2020; 2021). Reducing embodied carbon cost for edge devices is also important, as manufacturing carbon cost accounts for 74% of the total footprint (Gupta et al., 2021) of client devices. However, amortizing the embodied carbon footprint is particularly challenging due to significant under-utilization (Gao et al., 2015), motivating new environmentally-sustainable system design.

4.4 Carbon Footprint Telemetry for AI

While the open source community has started building tools to enable automatic measurement of AI training’s environmental footprint (Lacoste et al., 2019; Henderson et al., 2020; Schmidt et al., 2021) and the ML research community requiring a broader impact statement for the submitted research manuscript, more can be done in order to incorporate efficiency and sustainability into the design process. Enabling carbon accounting methodologies and telemetry that is easy to adopt is an important step to quantify the significance of our progress in developing AI technologies in an environmentally-responsible manner. While assessing the novelty and quality of ML solutions, it is crucial to consider sustainability metrics including *energy consumption* and *carbon footprint* along with measures of *model quality* and *system performance*. Note, while embodied carbon footprints for AI hardware may not be readily available,

describing hardware platforms, the number of machines, total runtime used to produce results presented in a research manuscript is an important first step.

Metrics for AI Model and System Life Cycles: Standard carbon footprint accounting methods for AI’s overall carbon footprint are at a nascent stage. We need simple, easy-to-adopt metrics to make fair and useful comparisons between AI innovations. Many different aspects must be accounted for, including the life cycles of both AI models (*Data, Experimentation, Training, Deployment*) and system hardware (*Manufacturing and Use*).

In addition to incorporating an efficiency measure as part of leader boards for various ML tasks, data (Kiela et al., 2021), models⁴, training algorithms (Hernandez & Brown, 2020b), environmental impact must also be considered and adopted by AI system hardware developers, using standard AI system performance benchmarks, such as MLPerf (Mattson et al., 2020; Reddi et al., 2021). Moreover, an algorithm efficiency benchmark is under development. The MLPerf benchmark standards can advance the field of AI in an environmentally-competitive manner.

5 KEY TAKEAWAYS

The Growth of AI: Deep learning has witnessed an exponential growth in training data, model parameters, and system resources over the recent years. The amount of data for AI has grown by $2.4\times$, leading to $3.2\times$ increase in the data ingestion bandwidth demand. Recommendation model sizes have increased by $20\times$ between 2019 and 2021. The explosive growth in AI use cases has driven $2.9\times$ and $2.5\times$ capacity increases for AI training and inference at Meta over the recent 18 months, respectively. The environmental footprint of AI is staggering (Figure 4, Figure 5).

A Holistic Approach: To ensure an environmentally-sustainable growth of AI, we must consider the AI ecosystem holistically going forward. We must look at the machine learning pipelines end-to-end — data collection, model exploration and experimentation, model training, optimization and run-time inference (Section 2). The frequency of training and scale of each stage of the ML pipeline must be considered to understand salient bottlenecks to sustainable AI. From the system’s perspective, the life cycle of model development and system hardware, including *manufacturing* and *operational use*, must also be accounted for.

Efficiency Optimization: Optimization across the axes of algorithms, platforms, infrastructures, hardware can significantly reduce the operational carbon footprint for the universal translation model by $810\times$. Along with other efficiency optimization at-scale, this has translated into 25.8%

⁴Papers with code: <https://paperswithcode.com/ota/image-classification-on-imagenet>

operational energy footprint reduction over the two-year period. *More must be done to bend the environmental impact from the exponential growth of AI* (Figure 6).

An Sustainability Mindset for AI: Optimization beyond efficiency across the software and hardware stack at scale is crucial to enabling future sustainable AI systems. To develop AI technologies responsibly, we must achieve competitive model accuracy at a fixed or even reduced computational and environmental cost. We chart out potentially high-impact research and development directions across the *data, algorithms and model, experimentation and system hardware, and telemetry* dimensions for AI at datacenters and at the edge (Section 4).

We must take a deliberate approach when developing AI research and technologies, considering the environmental impact of innovations and taking a responsible approach to technology development (Wu et al., 2021). That is, we need AI to be green and environmentally-sustainable.

6 CONCLUSION

This paper is the first effort to explore the environmental impact of super-linear trends for AI growth from a holistic perspective, spanning *data, algorithms, and system hardware*. We characterize the carbon footprint of AI computing by examining the model development cycle across industry-scale ML use cases at Meta while considering the life cycle of system hardware. Furthermore, we capture the operational and manufacturing carbon footprint of AI computing and present an end-to-end analysis for *what* and *how* hardware-software design and at-scale optimization can help reduce the overall carbon footprint of AI. We share key challenges and chart out important directions across all dimensions of AI—*data, algorithms, systems, metrics, standards, and best experimentation practices*. Advancing the field of machine intelligence must not in turn make climate change worse. We must develop AI technologies with a deeper understanding of the societal and environmental implications.

ACKNOWLEDGEMENTS

We would like to thank the anonymous reviewers for their valuable feedback. We also thank Nikhil Gupta, Lei Tian, Weiyi Zheng, Manisha Jain, Adnan Aziz, and Adam Lerer for their discussions around ML efficiency; Adina Williams, Emily Dinan, Mona Diab, Ashkan Yousefpour, Mark Zhou, Niket Agarwal, Jongsoo Park, Michael Anderson, Xiaodong Wang for the valuable discussions and insights on AI and environmental responsibility; John Nguyen, Kunal Bhalla, Haiyu Lu, Kaikai Wang, Jesik Min, and Dzmityr Huba for their help in the federated learning and mobile AI measurement; Yatharth Saraf, Hagay Lupesco, Jigar Desai, Ram Valliyappan, Rajesh Mosur, Ananth Sankarnarayanan, Eytan Bakshy, and Joelle Pineau for their support without which this work would not have been possible.

REFERENCES

- Acun, B., Murphy, M., Wang, X., Nie, J., Wu, C., and Hazelwood, K. Understanding training efficiency of deep learning recommendation models at scale. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, 2021.
- Acun, B., Lee, B., Maeng, K., Chakkaravarthy, M., Gupta, U., Brooks, D., and Wu, C.-J. A holistic approach for designing carbon aware datacenters, 2022.
- Adiwardana, D. and Luong, T. Towards a conversational agent that can chat about... anything, 2020. URL <https://ai.googleblog.com/2020/01/towards-conversational-agent-that-can.html>.
- Alistarh, D., Grubic, D., Li, J., Tomioka, R., and Vojnovic, M. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 30, 2017.
- Apple. Product environmental report Mac Pro, 2019. URL https://www.apple.com/environment/pdf/products/desktops/Mac_Pro_PER_Dec2019.pdf.
- Artetxe, M., Bhosale, S., Goyal, N., Mihaylov, T., Ott, M., Shleifer, S., Lin, X. V., Du, J., Iyer, S., Pasunuru, R., Anantharaman, G., Li, X., Chen, S., Akin, H., Baines, M., Martin, L., Zhou, X., Koura, P. S., O'Horo, B., Wang, J., Zettlemoyer, L., Diab, M., Kozareva, Z., and Stoyanov, V. Efficient large scale language modeling with mixtures of experts, 2021.
- Assran, M., Caron, M., Misra, I., Bojanowski, P., Joulin, A., Ballas, N., and Rabbat, M. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *arXiv preprint arXiv:2104.13963*, 2021.
- Azure. New Azure HPC and partner offerings at Supercomputing 19, 2019. URL <https://azure.microsoft.com/en-us/blog/new-azure-hpc-and-partner-offerings-at-supercomputing-19/>.
- Bardon, M., Wuytens, P., Ragnarsson, L.-, Mirabelli, G., Jang, D., Willems, G., Mallik, A., Spessot, A., Ryckaert, J., and Parvais, B. DTCO including sustainability: Power-performance-area-cost-environmental score (PPACE) analysis for logic technologies. In *Proceedings of the IEEE International Electron Devices Meeting*, 2020.
- Bender, E. M., Gebru, T., McMillan-Major, A., and Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency*, 2021.
- Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., Arora, S., von Arx, S., Bernstein, M. S., Bohg, J., Bosselut, A., Brunskill, E., et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.
- Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., Kiddon, C., Konečný, J., Mazzocchi, S., McMahan, H. B., et al. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*, 2019.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
- Cai, H., Gan, C., Wang, T., Zhang, Z., and Han, S. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2020a.
- Cai, H., Gan, C., Zhu, L., and Han, S. Tinytl: Reduce memory, not parameters for efficient on-device learning. *arXiv preprint arXiv:2007.11622*, 2020b.
- Chang, J., Meza, J., Ranganathan, P., Bash, C., and Shah, A. Green server design: Beyond operational energy to sustainability. In *Proceedings of the International Conference on Power Aware Computing and Systems*, 2010.
- Charles, Z., Garrett, Z., Huo, Z., Shmulyian, S., and Smith, V. On large-cohort training for federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.
- Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the International conference on machine learning*, pp. 1597–1607, 2020.
- Chen, W., Gong, X., and Wang, Z. Neural architecture search on imagenet in four gpu hours: A theoretically inspired perspective. *arXiv preprint arXiv:2102.11535*, 2021.
- Chen, Y.-H., Emer, J., and Sze, V. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture*, 2016.

- Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., and Dahl, G. E. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*, 2020.
- Deng, Z., Park, J., Tang, P. T. P., Liu, H., Yang, J., Yuen, H., Huang, J., Khudia, D., Wei, X., Wen, E., Choudhary, D., Krishnamoorthi, R., Wu, C.-J., Nadathur, S., Kim, C., Naumov, M., Naghshineh, S., and Smelyanskiy, M. Low-precision hardware architectures meet recommendation model inference at scale. *IEEE Micro*, 41(5):93–100, 2021.
- Dery, L. M., Michel, P., Talwalkar, A., and Neubig, G. Should we be pre-training? an argument for end-task aware training as an alternative. *arXiv preprint arXiv:2109.07437*, 2021.
- Dixit, H. D., Pendharkar, S., Beadon, M., Mason, C., Chakravarthy, T., Muthiah, B., and Sankar, S. Silent data corruptions at scale. *arXiv preprint arXiv:2102.11245*, 2021.
- Eisenman, A., Matam, K. K., Ingram, S., Mudigere, D., Krishnamoorthi, R., Nair, K., Smelyanskiy, M., and Annavaram, M. Check-n-run: A checkpointing system for training deep learning recommendation models. *arXiv preprint arXiv:2010.08679*, 2021.
- Elkin, C. and Witherspoon, S. Machine learning can boost the value of wind energy, 2019. URL <https://deepmind.com/blog/article/machine-learning-can-boost-value-wind-energy>.
- EPA. United states environmental protection agency greenhouse gas equivalencies calculator, 2021.
- Eriksson, D., Chuang, P. I., Daulton, S., Xia, P., Shrivastava, A., Babu, A., Zhao, S., Aly, A., Venkatesh, G., and Balandat, M. Latency-aware neural architecture search with multi-objective bayesian optimization. *CoRR*, abs/2106.11890, 2021.
- Meta. 2020 sustainability report, 2021.
- Fedus, W., Zoph, B., and Shazeer, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *CoRR*, abs/2101.03961, 2021.
- Gao, C., Gutierrez, A., Rajan, M., Dreslinski, R. G., Mudge, T., and Wu, C.-J. A study of mobile device utilization. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, 2015.
- Ghodrati, S., Ahn, B. H., Kyung Kim, J., Kinzer, S., Yatham, B. R., Alla, N., Sharma, H., Alian, M., Ebrahimi, E., Kim, N. S., Young, C., and Esmaeilzadeh, H. Planaria: Dynamic architecture fission for spatial multi-tenant acceleration of deep neural networks. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, 2020.
- Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.
- Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., Martin, M., Nagarajan, T., Radosavovic, I., Ramakrishnan, S. K., Ryan, F., Sharma, J., Wray, M., Xu, M., Xu, E. Z., Zhao, C., Bansal, S., Batra, D., Cartillier, V., Crane, S., Do, T., Doulaty, M., Erapalli, A., Feichtenhofer, C., Fragomeni, A., Fu, Q., Fuegen, C., Gebreselasie, A., Gonzalez, C., Hillis, J., Huang, X., Huang, Y., Jia, W., Khoo, W., Kolar, J., Kottur, S., Kumar, A., Landini, F., Li, C., Li, Y., Li, Z., Mangalam, K., Modhugu, R., Munro, J., Murrell, T., Nishiyasu, T., Price, W., Puentes, P. R., Ramazanov, M., Sari, L., Somasundaram, K., Southerland, A., Sugano, Y., Tao, R., Vo, M., Wang, Y., Wu, X., Yagi, T., Zhu, Y., Arbelaez, P., Crandall, D., Damen, D., Farinella, G. M., Ghanem, B., Ithapu, V. K., Jawahar, C. V., Joo, H., Kitani, K., Li, H., Newcombe, R., Oliva, A., Park, H. S., Rehg, J. M., Sato, Y., Shi, J., Shou, M. Z., Torralba, A., Torresani, L., Yan, M., and Malik, J. Ego4d: Around the world in 3,000 hours of egocentric video. *arXiv:2110.07058*, 2021.
- Gschwind, M., Kaldewey, T., and Tam, D. Optimizing the efficiency of deep learning through accelerator virtualization. *IBM Journal of Research and Development*, 61(4-5), 2017.
- Gupta, U., Wu, C.-J., Wang, X., Naumov, M., Reagen, B., Brooks, D., Cottel, B., Hazelwood, K., Hempstead, M., Jia, B., Lee, H.-H. S., Malevich, A., Mudigere, D., Smelyanskiy, M., Xiong, L., and Zhang, X. The architectural implications of facebook’s dnn-based personalized recommendation. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, 2020.
- Gupta, U., Kim, Y., Lee, S., Tse, J., Lee, H. S., Wei, G., Brooks, D., and Wu, C. Chasing carbon: The elusive environmental footprint of computing. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, 2021.
- Hamilton, J. AWS Inferentia Machine Learning Processor, 2018. URL <https://perspectives.mvdirona>.

- com/2018/11/aws-inferentia-machine-learning-processor/.
- Hard, A., Rao, K., Mathews, R., Ramaswamy, S., Beaufays, F., Augenstein, S., Eichner, H., Kiddon, C., and Ramage, D. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.
- Hazelwood, K., Bird, S., Brooks, D., Chintala, S., Diril, U., Dzhulgakov, D., Fawzy, M., Jia, B., Jia, Y., Kalro, A., Law, J., Lee, K., Lu, J., Noordhuis, P., Smelyanskiy, M., Xiong, L., and Wang, X. Applied machine learning at facebook: A datacenter infrastructure perspective. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, 2018.
- Henderson, P., Hu, J., Romoff, J., Brunskill, E., Jurafsky, D., and Pineau, J. Towards the systematic reporting of the energy and carbon footprints of machine learning. *CoRR*, abs/2002.05651, 2020.
- Hernandez, D. and Brown, T. B. Measuring the algorithmic efficiency of neural networks. *arXiv preprint arXiv:2005.04305*, 2020a.
- Hernandez, D. and Brown, T. B. Measuring the algorithmic efficiency of neural networks. *arXiv preprint arXiv:2005.04305*, 2020b.
- Hochschild, P. H., Turner, P., Mogul, J. C., Govindaraju, R., Ranganathan, P., Culler, D. E., and Vahdat, A. Cores that don't count. In *Proceedings of the Workshop on Hot Topics in Operating Systems*, 2021.
- Huang, Y., Cheng, Y., Bapna, A., Firat, O., Chen, D., Chen, M., Lee, H., Ngiam, J., Le, Q. V., Wu, Y., et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Proceedings of the Advances in neural information processing systems*, volume 32, 2019.
- Huba, D., Nguyen, J., Malik, K., Zhu, R., Rabbat, M., Yousefpour, A., Wu, C.-J., Zhan, H., Ustinov, P., Srinivas, H., Wang, K., Shoumikhin, A., Min, J., and Malek, M. Papaya: Practical, private, and scalable federated learning. In *Proceedings of the Conference on Machine Learning and Systems*, 2022.
- Jain, R. and Wullert, J. Challenges: Environmental design for pervasive computing systems. In *Proceedings of the International Conference on Mobile Computing and Networking*, 2002.
- Jeon, M., Venkataraman, S., Phanishayee, A., Qian, u., Xiao, W., and Yang, F. Analysis of large-scale multi-tenant gpu clusters for dnn training workloads. In *Proceedings of the USENIX Annual Technical Conference*, 2019.
- Joglekar, M. R., Li, C., Chen, M., Xu, T., Wang, X., Adams, J. K., Khaitan, P., Liu, J., and Le, Q. V. Neural input search for large scale recommendation models. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2020.
- Jouppi, N. P., Young, C., Patil, N., Patterson, D., Agrawal, G., Bajwa, R., Bates, S., Bhatia, S., Boden, N., Borchers, A., Boyle, R., Cantin, P.-l., Chao, C., Clark, C., Coriell, J., Daley, M., Dau, M., Dean, J., Gelb, B., Ghaemmaghami, T. V., Gottipati, R., Gulland, W., Hagmann, R., Ho, C. R., Hogberg, D., Hu, J., Hundt, R., Hurt, D., Ibarz, J., Jaffey, A., Jaworski, A., Kaplan, A., Khaitan, H., Killebrew, D., Koch, A., Kumar, N., Lacy, S., Laudon, J., Law, J., Le, D., Leary, C., Liu, Z., Lucke, K., Lundin, A., MacKean, G., Maggiore, A., Mahony, M., Miller, K., Nagarajan, N., Narayanaswami, R., Ni, R., Nix, K., Norrie, T., Omernick, M., Penukonda, N., Phelps, A., Ross, J., Ross, M., Salek, A., Samadiani, E., Severn, C., Sizikov, G., Snelham, M., Souter, J., Steinberg, D., Swing, A., Tan, M., Thorson, G., Tian, B., Toma, H., Tuttle, E., Vasudevan, V., Walter, R., Wang, W., Wilcox, E., and Yoon, D. H. In-datacenter performance analysis of a tensor processing unit. In *Proceedings of the ACM/IEEE International Symposium on Computer Architecture*, 2017.
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstern, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 2021.
- Kang, W.-C., Cheng, D. Z., Yao, T., Yi, X., Chen, T., Hong, L., and Chi, E. H. Learning to embed categorical features without embedding tables for recommendation. *arXiv preprint arXiv:2010.10784*, 2021.
- Kang, Y., Hauswald, J., Gao, C., Rovinski, A., Mudge, T., Mars, J., and Tang, L. Neurosurgeon: Collaborative intelligence between the cloud and mobile edge. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2017.
- Kao, S.-C. and Krishna, T. Domain-specific genetic algorithm for multi-tenant dnnaccelerator scheduling. *arXiv preprint arXiv:2104.13997*, 2021.
- Ke, L., Gupta, U., Cho, B. Y., Brooks, D., Chandra, V., Diril, U., Firoozshahian, A., Hazelwood, K., Jia, B., Lee, H.-H. S., Li, M., Maher, B., Mudigere, D., Naumov,

- M., Schatz, M., Smelyanskiy, M., Wang, X., Reagen, B., Wu, C.-J., Hempstead, M., and Zhang, X. *RecNMP: Accelerating Personalized Recommendation with near-Memory Processing*. 2020.
- Kiela, D., Bartolo, M., Nie, Y., Kaushik, D., Geiger, A., Wu, Z., Vidgen, B., Prasad, G., Singh, A., Ringshia, P., Ma, Z., Thrush, T., Riedel, S., Waseem, Z., Stenertorp, P., Jia, R., Bansal, M., Potts, C., and Williams, A. Dynabench: Rethinking benchmarking in NLP. *arXiv preprint arXiv:2104.14337*, 2021.
- Kim, Y. G. and Wu, C.-J. Autoscale: Energy efficiency optimization for stochastic edge inference using reinforcement learning. In *Proceedings of the IEEE/ACM International Symposium on Microarchitecture*, 2020.
- Kim, Y. G. and Wu, C.-J. Autofl: Enabling heterogeneity-aware energy efficient federated learning. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, 2021.
- Lacoste, A., Luccioni, A., Schmidt, V., and Dandres, T. Quantifying the carbon emissions of machine learning. *Workshop on Tackling Climate Change with Machine Learning at NeurIPS 2019*, 2019.
- Lui, M., Yetim, Y., Özkan, , Zhao, Z., Tsai, S.-Y., Wu, C.-J., and Hempstead, M. Understanding capacity-driven scale-out neural recommendation inference. In *Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, 2021.
- Maeng, K., Bharuka, S., Gao, I., Jeffrey, M. C., Saraph, V., Su, B.-Y., Trippel, C., Yang, J., Rabbat, M., Lucia, B., and Wu, C.-J. Cpr: Understanding and improving failure tolerant training for deep learning recommendation with partial recovery. In *Proceedings of the Conference on Machine Learning and Systems*, 2021.
- Mattson, P., Reddi, V. J., Cheng, C., Coleman, C., Diamos, G., Kanter, D., Micikevicius, P., Patterson, D., Schmuelling, G., Tang, H., Wei, G.-Y., and Wu, C.-J. Mlperf: An industry standard benchmark suite for machine learning performance. *IEEE Micro*, 40(2):8–16, 2020.
- Mellor, J., Turner, J., Storkey, A., and Crowley, E. J. Neural architecture search without training. *arXiv preprint arXiv:2006.04647*, 2021.
- Mudigere, D., Hao, Y., Huang, J., Tulloch, A., Sridharan, S., Liu, X., Ozdal, M., Nie, J., Park, J., Luo, L., Yang, J. A., Gao, L., Ivchenko, D., Basant, A., Hu, Y., Yang, J., Ardestani, E. K., Wang, X., Komuravelli, R., Chu, C., Yilmaz, S., Li, H., Qian, J., Feng, Z., Ma, Y., Yang, J., Wen, E., Li, H., Yang, L., Sun, C., Zhao, W., Melts, D., Dhulipala, K., Kishore, K. R., Graf, T., Eisenman, A., Matam, K. K., Gangidi, A., Chen, G. J., Krishnan, M., Nayak, A., Nair, K., Muthiah, B., khorashadi, M., Bhattacharya, P., Lapukhov, P., Naumov, M., Qiao, L., Smelyanskiy, M., Jia, B., and Rao, V. Software-hardware co-design for fast and scalable training of deep learning recommendation models. *arXiv preprint arXiv:2104.05158*, 2021.
- Naumov, M., Mudigere, D., Shi, H.-J. M., Huang, J., Sundaraman, N., Park, J., Wang, X., Gupta, U., Wu, C.-J., Azzolini, A. G., Dzhulgakov, D., Mallevech, A., Cherniavskii, I., Lu, Y., Krishnamoorthi, R., Yu, A., Kondratenko, V., Pereira, S., Chen, X., Chen, W., Rao, V., Jia, B., Xiong, L., and Smelyanskiy, M. Deep learning recommendation model for personalization and recommendation systems. *arXiv preprint arXiv:1906.00091*, 2019.
- Nemirovskij, A. S. and Yudin, D. B. Problem complexity and method efficiency in optimization. 1983.
- NVIDIA. GPUs for Virtualization. 2021a. URL <https://www.nvidia.com/en-us/data-center/graphics-cards-for-virtualization/>.
- NVIDIA. Faster Transformer, 2021b. URL <https://github.com/NVIDIA/FasterTransformer>.
- Ott, M., Edunov, S., Grangier, D., and Auli, M. Scaling neural machine translation. *arXiv preprint arXiv:1806.00187*, 2018.
- Patterson, D., Gonzalez, J., Le, Q., Liang, C., Munguia, L.-M., Rothchild, D., So, D., Texier, M., and Dean, J. Carbon emissions and large neural network training. *arXiv preprint arXiv:2104.10350*, 2021.
- Putnam, A., Caulfield, A. M., Chung, E. S., Chiou, D., Constantinides, K., Demme, J., Esmailzadeh, H., Fowers, J., Gopal, G. P., Gray, J., Haselman, M., Hauck, S., Heil, S., Hormati, A., Kim, J.-Y., Lanka, S., Larus, J., Peterson, E., Pope, S., Smith, A., Thong, J., Xiao, P. Y., and Burger, D. A reconfigurable fabric for accelerating large-scale datacenter services. *IEEE Micro*, 2015.
- Qiu, X., Parcollet, T., Fernandez-Marques, J., de Gusmao, P. P. B., Beutel, D. J., Topal, T., Mathur, A., and Lane, N. D. A first look into the carbon footprint of federated learning. *arXiv preprint arXiv:2102.07627*, 2021.
- Radovanovic, A., Koningstein, R., Schneider, I., Chen, B., Duarte, A., Roy, B., Xiao, D., Haridasan, M., Hung, P., Care, N., et al. Carbon-aware computing for datacenters. *arXiv preprint arXiv:2106.11750*, 2021.
- Rajbhandari, S., Rasley, J., Ruwase, O., and He, Y. Zero: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020.

- Rajbhandari, S., Ruwase, O., Rasley, J., Smith, S., and He, Y. Zero-infinity: Breaking the gpu memory wall for extreme scale deep learning. *arXiv preprint arXiv:2104.07857*, 2021.
- Ramaswamy, S., Mathews, R., Rao, K., and Beaufays, F. Federated learning for emoji prediction in a mobile keyboard. *arXiv preprint arXiv:1906.04329*, 2019.
- Rasley, J., Rajbhandari, S., Ruwase, O., and He, Y. Deep-speed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.
- Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., and Wu, C.-J. The vision behind mlperf: Understanding ai inference performance. *IEEE Micro*, 41(3):10–18, 2021.
- Ren, P., Xiao, Y., Chang, X., Huang, P.-y., Li, Z., Chen, X., and Wang, X. A comprehensive survey of neural architecture search: Challenges and solutions. *ACM Comput. Surv.*, 54(4), 2021.
- Sachdeva, N., Wu, C.-J., and McAuley, J. Svp-cf: Selection via proxy for collaborative filtering data. *arXiv preprint arXiv:2107.04984*, 2021.
- Schmidt, V., Goyal, K., Joshi, A., Feld, B., Conell, L., Laskaris, N., Blank, D., Wilson, J., Friedler, S., and Lucioni, S. CodeCarbon: Estimate and Track Carbon Emissions from Machine Learning Computing. 2021. doi: 10.5281/zenodo.4658424. URL <https://github.com/mlco2/codecarbon>.
- Sethi, G., Acun, B., Agarwal, N., Kozyrakis, C., Trippel, C., and Wu, C.-J. Recshard: Statistical feature-based memory optimization for industry-scale neural recommendation. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2022.
- Sheikh, K. A Growing Presence on the Farm: Robots, February 2020. URL <https://www.nytimes.com/2020/02/13/science/farm-agriculture-robots.html>.
- Sivaprasad, P. T., Mai, F., Vogels, T., Jaggi, M., and Fleuret, F. Optimizer benchmarking needs to account for hyperparameter tuning. In *Proceedings of the International Conference on Machine Learning*, 2020.
- Song, Q., Cheng, D., Zhou, H., Yang, J., Tian, Y., and Hu, X. Towards automated neural interaction discovery for click-through rate prediction. *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*, 2020.
- Speed Index. Speedtest Global Index, January 2022. URL <https://www.speedtest.net/global-index>.
- Spiridonov, A. New Cloud TPU VMs make training your ML models on TPUs easier than ever, 2021. URL <https://cloud.google.com/blog/products/compute/introducing-cloud-tpu-vm>.
- Sriraman, A. and Dhanotia, A. Accelerometer: Understanding acceleration opportunities for data center overheads at hyperscale. In *Proceedings of the International Conference on Architectural Support for Programming Languages and Operating Systems*, 2020.
- Sriraman, A., Dhanotia, A., and Wenisch, T. F. Softsku: Optimizing server architectures for microservice diversity @scale. In *Proceedings of the 46th International Symposium on Computer Architecture*. Association for Computing Machinery, 2019.
- Stamoulis, D., Chin, T.-W. R., Prakash, A. K., Fang, H., Sajja, S., Bogner, M., and Marculescu, D. Designing adaptive neural networks for energy-constrained image classification. In *Proceedings of the International Conference on Computer-Aided Design*, 2018.
- Stamoulis, D., Ding, R., Wang, D., Lymberopoulos, D., Priyantha, B., Liu, J., and Marculescu, D. Single-path nas: Designing hardware-efficient convnets in less than 4 hours. *arXiv preprint arXiv:1904.02877*, 2019.
- Strubell, E., Ganesh, A., and McCallum, A. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019.
- Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2020.
- Tang, C., Yu, K., Veeraraghavan, K., Kaldor, J., Michelson, S., Kooburat, T., Anbudurai, A., Clark, M., Gogia, K., Cheng, L., Christensen, B., Gartrell, A., Khutorenko, M., Kulkarni, S., Pawlowski, M., Pelkonen, T., Rodrigues, A., Tibrewal, R., Venkatesan, V., and Zhang, P. Twine: A unified cluster management system for shared infrastructure. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation*, 2020.
- Tomaev, N., Cornebise, J., Hutter, F., Mohamed, S., Picciariello, A., Connelly, B., Belgrave, D., Ezer, D., van der Haert, F. C., Mugisha, F., Abila, G., Arai, H., Almiraat, H., Proskurnia, J., Snyder, K., Otake-Matsuura, M., Othman, M., Glasmachers, T., Wever, W. D., Teh, Y., Khan, M. E., Winne, R. D., Schaul, T., and Clopath, C. Ai for social good: unlocking the opportunity for positive impact. *Nature Communications*, 11, 2020.

- TPCDB. The Power Consumption Database, January 2022. URL <http://www.tpcdb.com/>.
- Turner, R., Eriksson, D., McCourt, M., Kiili, J., Laaksonen, E., Xu, Z., and Guyon, I. Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020. *CoRR*, abs/2104.10201, 2021.
- Valavi, E., Hestness, J., Ardalani, N., and Iansiti, M. *Time and the Value of Data*. Working papers. Harvard Business School, 2020. URL <https://books.google.com/books?id=IFbmzQEACAAJ>.
- Vishwanath, A., Jalali, F., Hinton, K., Alpcan, T., Ayre, R. W., and Tucker, R. S. Energy consumption comparison of interactive cloud-based and local applications. *IEEE Journal on selected areas in communications*, 33(4):616–626, 2015.
- Vogels, T., Karinireddy, S. P., and Jaggi, M. Powersgd: Practical low-rank gradient compression for distributed optimization. In *Proceedings of the Advances In Neural Information Processing Systems*, volume 32, 2019.
- Wang, H., Kim, B., Xie, J., and Han, Z. How is energy consumed in smartphone deep learning apps? executing locally vs. remotely. In *Proceedings of the IEEE Global Communications Conference*, 2019a.
- Wang, K., Mathews, R., Kiddon, C., Eichner, H., Beaufays, F., and Ramage, D. Federated evaluation of on-device personalization. *arXiv preprint arXiv:1910.10252*, 2019b.
- Wesolowski, L., Acun, B., Andrei, V., Aziz, A., Dankel, G., Gregg, C., Meng, X., Meurillon, C., Sheahan, D., Tian, L., Yang, J., Yu, P., and Hazelwood, K. Datacenter-scale analysis and optimization of gpu machine learning workloads. *IEEE Micro*, 41(5), 2021.
- Wilkening, M., Gupta, U., Hsia, S., Trippel, C., Wu, C.-J., Brooks, D., and Wei, G.-Y. *RecSSD: Near Data Processing for Solid State Drive Based Recommendation Inference*. 2021.
- Wu, C.-J., Brooks, D., Chen, K., Chen, D., Choudhury, S., Dukhan, M., Hazelwood, K., Isaac, E., Jia, Y., Jia, B., Leyvand, T., Lu, H., Lu, Y., Qiao, L., Reagen, B., Spisak, J., Sun, F., Tulloch, A., Vajda, P., Wang, X., Wang, Y., Wasti, B., Wu, Y., Xian, R., Yoo, S., and Zhang, P. Machine learning at facebook: Understanding inference at the edge. In *Proceedings of the IEEE International Symposium on High Performance Computer Architecture*, 2019.
- Wu, C.-J., Manne, S., Ranganathan, P., Bird, S., and Greenstein, S. Socio-technological challenges and opportunities: Paths forward. *arXiv preprint arXiv:2108.06738*, 2021.
- Yang, T., Andrew, G., Eichner, H., Sun, H., Li, W., Kong, N., Ramage, D., and Beaufays, F. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.
- Yang, T.-J., Liao, Y.-L., and Sze, V. Netadaptv2: Efficient neural architecture search with fast super-network training and architecture optimization, 2021.
- Yin, C., Acun, B., Liu, X., and Wu, C.-J. TT-Rec: Tensor train compression for deep learning recommendation models. In *Proceedings of the Conference on Machine Learning and Systems*, 2021.
- Yu, P. and Chowdhury, M. Salus: Fine-grained gpu sharing primitives for deep learning applications. *arXiv preprint arXiv:1902.04610*, 2019.
- Zhao, M., Agarwal, N., Basant, A., Gedik, B., Pan, S., Ozdal, M., Komuravelli, R., Pan, J., Bao, T., Lu, H., Narayanan, S., Langman, J., Wilfong, K., Rastogi, H., Wu, C., Kozyrakis, C., and Pol, P. Understanding and co-designing the data ingestion pipeline for industry-scale recsys training. *CoRR*, abs/2108.09373, 2021.
- Zitnick, C. L., Chanussot, L., Das, A., Goyal, S., Heras-Domingo, J., Ho, C., Hu, W., Lavril, T., Palizhati, A., Riviere, M., Shuaibi, M., Sriram, A., Tran, K., Wood, B., Yoon, J., Parikh, D., and Ulissi, Z. An introduction to electrocatalyst design using machine learning for renewable energy storage. *arXiv preprint arXiv:2010.09435*, 2020.

APPENDIX

A EFFICIENT, SUSTAINABLE AI SYSTEMS

General-Purpose Processors, General-Purpose Accelerators, Reconfigurable Systems, and ASICs for AI: There is a wide variety of system hardware choices for AI from general-purpose processors (CPUs), general-purpose accelerators (GPUs or TPUs), field-programmable gate arrays (FPGAs) (Putnam et al., 2015), to application-specific integrated circuit (ASIC), such as Eyeriss (Chen et al., 2016). The exact system deployment choice can be multifaceted — the cadence of ML algorithm and model architecture evolution, the diversity of ML use cases and the respective system resource requirements, and the maturity of the software stack. While ML accelerator deployment brings a step-function improvement in *operational energy efficiency*, it may not necessarily reduce the carbon footprint of AI computing overall. This is because of the upfront embodied carbon footprint associated with the different system hardware choices. From the environmental sustainability perspective, the optimal point depends on the compounding factor of operational efficiency improvement over generations of ML algorithms/models, deployment lifetime and embodied carbon footprint of the system hardware. Thus, to design for environmental sustainability, one must strike a careful balance between *efficiency* and *flexibility* and, at the same time, consider environmental impact as a key design dimension for next-generation AI systems.

Disaggregating Machine Learning Pipeline Stages: As depicted in Figure 3, the overall training throughput efficiency for large-scale ML models depends on the throughput performance of both *data ingestion and pre-processing* and *model training*. Disaggregating the data ingestion and pre-processing stage of the machine learning pipeline from model training is the de-facto approach for industry-scale machine learning model training. This allows training accelerator, network and storage I/O bandwidth utilization to scale independently, thereby increasing the overall model training throughput by 56% (Zhao et al., 2021). Disaggregation with well-designed check-pointing support (Maeng et al., 2021; Eisenman et al., 2021) improves training fault tolerance as well. By doing so, failure on nodes that are responsible for data ingestion and pre-processing can be recovered efficiently without requiring re-runs of the entire training experiment. From a sustainability perspective, disaggregating the data storage and ingestion stage from model training maximizes infrastructure efficiency by *using less system resources to achieve higher training throughput*, resulting in lower embodied carbon footprint. By increasing fault tolerance, the operational carbon footprint is reduced at the same time.

Fault-Tolerant AI Systems and Hardware: One way to

amortize the rising embodied carbon cost of AI infrastructures is to extend hardware lifetime. However, hardware ages — depending on the wear-out characteristics, increasingly more errors can surface over time and result in *silent data corruption*, leading to erroneous computation, model accuracy degradation, non-deterministic ML execution, or fatal system failure. In a large fleet of processors, silent data corruption can occur frequently enough to have disruptive impact on service productivity (Dixit et al., 2021; Hochschild et al., 2021). Decommissioning an AI system entirely because of hardware faults is expensive from the perspective of resource and environmental footprints. System architects can design differential reliability levels for micro architectural components on an AI system depending on the ML model execution characteristics. Alternatively, algorithmic fault tolerance can be built into deep learning programming frameworks to provide a code execution path that is cognizant of hardware wear-out characteristics.

B EFFICIENCY AND SELF-SUPERVISED LEARNING

Self-supervised learning (SSL) have received much attention in the research community in recent years. SSL methods train deep neural networks without using explicit supervision in the form of human-annotated labels for each training sample. Having humans annotate data is a time-consuming, expensive, and typically noisy process. SSL methods are typically used to train *foundation models* — models that can readily be fine-tuned using a small amount of labeled data on a down-stream task (Bommasani et al., 2021). SSL methods have been extremely successful for pre-training large language models, becoming the de-facto standard, and they have also attracted great interest in computer vision.

When comparing supervised and self-supervised methods, there is a glaring trade-off between having labels and the amount of computational overhead involved in pre-training. For example, Chen et al. (2020) report achieving 69.3% top-1 validation accuracy with a ResNet-50 model after SSL pre-training for 1000 epochs on the ImageNet dataset and using the linear evaluation protocol, freezing the pre-trained feature extractor, and fine-tuning a linear classifier on top for 60 epochs using the full ImageNet dataset with all labels. In contrast, the same model typically achieves at least 76.1% top-1 accuracy after 90 epochs of fully-supervised training. Thus, in this example, using labels and supervised training is worth a roughly 10× reduction in training effort, measured in terms of number of passes over the dataset.

Recent work suggests that incorporating even a small amount of labeled data can significantly bridge this gap. Assran et al. (2021) describe an approach called *Predicting view Assignments With Support samples* (PAWS) for semi-supervised pre-training inspired by SSL. With access

to labels for just 10% of the training images in ImageNet, a ResNet-50 achieves 75.5% top-1 accuracy after just 200 epochs of PAWS pre-training. Running on 64 V100 GPUs, this takes roughly 16 hours. Similar observations have recently been made for language model pre-training as well (Dery et al., 2021).

Self-supervised pre-training potentially has advantages in that a single foundation model can be trained (expensive) but then fine-tuned (inexpensive), amortizing the up front cost across many tasks (Bommasani et al., 2021). Substantial additional research is needed to better understand the cost-benefit trade-offs for this paradigm.

C CARBON EMISSION ESTIMATES FOR FEDERATED LEARNING

Carbon emissions of FL are based on the energy consumption of computation and communication. *Computation energy* is estimated as the product of device power and the computation time of each client. For device power, we used 3W (Wang et al., 2019a). *Communication energy* is modeled as the sum of energy spent by the client device and energy spent by the communication infrastructure, including the Wi-Fi access point, edge router, core router, broadband network gateway, and Ethernet switch (Vishwanath et al., 2015).

Communication power of client devices was estimated based on the vendor-provided power models of Android phones (`power.profile.xml`) over 52,000 Android devices. We then estimated the time taken for model parameter upload/download based on the model size of the FL use cases and the expected bandwidth available (Speed Index, 2022), resulting in the client device communication energy.

Energy consumed at the Wi-Fi access point was modeled as a linear model, where energy consumption linearly increases with the transmitted data size (Vishwanath et al., 2015). We used the measured maximum and idle power consumption data of 20 common Wi-Fi devices (TPCDB, 2022) to create the power model. Note, we exclude the idle power of the Wi-Fi hardware unlike prior work (Qiu et al., 2021). For the energy per bit transmission for the rest of the communication infrastructure, we directly used measurements from prior work (Vishwanath et al., 2015).