Journal of Machine Learning Research 21 (2020) 1-43 Submitted 4/20; Revised 10/20; Published 11/20

Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning

Peter Henderson PHEND@CS.STANFORD.EDU
Stanford University, Stanford, CA, USA

Jieru Hu JIERU@FB.COM
Facebook, Menlo Park, CA, USA

Joshua Romoff JOSHUA.ROMOFF@MAIL.MCGILL.CA
Mila, McGill University, Montreal, QC, Canada

Emma Brunskill EBRUN@CS.STANFORD.EDU
Stanford University, Stanford, CA, USA

Dan Jurafsky JURAFSKY@STANFORD.EDU
Stanford University, Stanford, CA, USA

Joelle Pineau JPINEAU@CS.MCGILL.CA
Facebook AI Research, Mila, McGill University, Montreal, QC, Canada

Editor: David Sontag

Abstract

Accurate reporting of energy and carbon usage is essential for understanding the potential climate impacts of machine learning research. We introduce a framework that makes this easier by providing a simple interface for tracking realtime energy consumption and carbon emissions, as well as generating standardized online appendices. Utilizing this framework, we create a leaderboard for energy efficient reinforcement learning algorithms to incentivize responsible research in this area as an example for other areas of machine learning. Finally, based on case studies using our framework, we propose strategies for mitigation of carbon emissions and reduction of energy consumption. By making accounting easier, we hope to further the sustainable development of machine learning experiments and spur more research into energy efficient algorithms.

Keywords: energy efficiency, green computing, reinforcement learning, deep learning, climate change

1. Introduction

Global climate change is a scientifically well-recognized phenomenon and appears to be accelerated due to greenhouse gas (GHG) emissions such as carbon dioxide or equivalents (CO_{2eq}) (Crowley, 2000; IPCC, 2018). The harmful health and safety impacts of global climate change are projected to "fall disproportionately on the poor and vulnerable" (IPCC, 2018). Energy production remains a large factor in GHG emissions, contributing about $\sim 25\%$ of GHG emissions in 2010 (IPCC, 2018). With the compute and energy demands of many modern machine learning (ML) methods growing exponentially (Amodei and Hernandez, 2018), ML systems have the potential to significantly contribute to carbon emissions. Recent

work has demonstrated these potential impacts through case studies and suggested various mitigating strategies (Strubell et al., 2019; Schwartz et al., 2019).

Systematic and accurate measurements are needed to better estimate the broader energy and carbon footprints of ML—in both research and production settings. Accurate accounting of carbon and energy impacts aligns incentives with energy efficiency (Schwartz et al., 2019), raises awareness, and drives mitigation efforts (Sundar et al., 2018; LaRiviere et al., 2016), among other benefits.[1] Yet, most ML research papers do not regularly report energy or carbon emissions metrics.[2]

We hypothesize that part of the reason that much research does not report energy and carbon metrics is due to the complexities of collecting them. Collecting carbon emission metrics requires understanding emissions from energy grids, recording power outputs from GPUs and CPUs, and navigating among different tools to accomplish these tasks. To reduce this overhead, we present *experiment-impact-tracker*[3]—a lightweight framework for consistent, easy, and more accurate reporting of energy, compute, and carbon impacts of ML systems.

In Section 4, we introduce the design and capabilities of our framework and the issues with accounting we aim to solve with this new framework. Section 5 expands on the challenges of using existing accounting methods and discusses our learnings from analyzing experiments with *experiment-impact-tracker*. For example, in an empirical case study on image classification algorithms, we demonstrate that floating point operations (FPOs), a common measure of efficiency, are often uncorrelated with energy consumption with energy metrics gathered by *experiment-impact-tracker*.

In Section 6, we focus on recommendations for promoting energy-efficient research and mitigation strategies for carbon emissions. Using our framework, we present a *Reinforcement Learning Energy Leaderboard* in Section 6.1.1 to encourage development of energy efficient algorithms. We also present a case study in machine translation to show how regional energy grid differences can result in large variations in CO_{2eq}emissions. Emissions can be reduced by up to 30x just by running experiments in locations powered by more renewable energy sources (Section 6.2). Finally, we suggest systemic and immediate changes based on our findings:

- incentivizing energy-efficient research through leaderboards (Section 6.1)
- running experiments in carbon-friendly regions (Section 6.2)
- reducing overheads for utilizing efficient algorithms and resources (Section 7.1)
- considering energy-performance trade-offs before deploying energy hungry models (Section 7.2)
- selecting efficient test environment especially in RL (Section 7.3)
- ensuring reproducibility to reduce energy consumption from replication difficulties (Section 7.4)
- consistently reporting energy and carbon metrics (Section 7.5)

1. See Section 4.1 for an extended discussion on the importance of accounting.
2. See Section 3 and Appendix B for more information.
3. `https://github.com/Breakend/experiment-impact-tracker`

2. Related Work

Estimating GHG emissions and their downstream consequences is important for setting regulatory standards (U.S. Environment Protection Agency, 2013) and encouraging self-regulation (Byerly et al., 2018). In particular, these estimates are used to set carbon emissions reduction targets and in turn set carbon prices for taxes or emissions trading systems.[4] A large body of work has examined modeling and accounting of carbon emissions[5] at different levels of granularity: at the global scale (IPCC, 2018); using country-specific estimates (Ricke et al., 2018); targeting a particular industrial sector like Information and Communication Technologies, for example, modeled by Malmodin et al. (2013); or even targeting a particular application like bitcoin mining, for example, modeled by Mora et al. (2018).

At the application level, some work has already modeled carbon impacts specifically in computationally intensive settings like bitcoin mining (Krause and Tolaymat, 2018; Stoll et al., 2019; Zade et al., 2019; Mora et al., 2018). Such application-specific efforts are important for prioritizing emissions mitigation strategies: without understanding projected impacts, policy decisions could focus on ineffective regulation. However, with large amounts of heterogeneity and endogeneity in the underlying data, it can be difficult to model all aspects of an application's usage. For example, one study suggested that "bitcoin emissions alone could push global warming above 2 °C" (Mora et al., 2018). But Masanet et al. (2019), Houy (2019), and others, criticized the underlying modeling assumptions which led to such large estimates of carbon emissions. This shows that it is vital that these models provide accurate measurements if they are to be used for informed decision making.

With ML models getting more computationally intensive (Amodei and Hernandez, 2018), we want to better understand how machine learning in research and industry impacts climate change. However, estimating aggregate climate change impacts of ML research and applications would require many assumptions due to a current lack of reporting and accounting. Instead, we aim to emphasize and aid systematic reporting strategies such that accurate field-wide estimates can be conducted in the future.

Some recent work specifically investigates climate impacts of machine learning research. Strubell et al. (2019) demonstrate the issue of carbon and energy impacts of large NLP models by evaluating estimated power usage and carbon emissions for a set of case studies. The authors suggest that: "authors should report training time and sensitivity to hyperparameters", "academic researchers need equitable access to computation resources", and "researchers should prioritize computationally efficient hardware and algorithms". Schwartz et al. (2019) provide similar proposals, suggesting floating point operations (FPOs) as a guiding efficiency metric. Lacoste et al. (2019) recently provided a website for estimating carbon emissions based on GPU type, experiment length, and cloud provider. In Section 5, we discuss how while the estimation methods of these works provide some understanding of carbon and energy impacts,

4. An emissions trading system is a cap on total allowed carbon emissions for a company with permits issued. When a company emits a certain amount of carbon, they trade in a permit, creating a market for emissions permits. This is a market-based approach to incentivize emission reductions. See Ramstein et al. (2019) for a description of such carbon pricing efforts across different countries.

5. See also assorted examinations on carbon accounting, standardized reporting, and policy recommendations (Stechemesser and Guenther, 2012; Dayarathna et al., 2015; IPCC, 2018; Ajani et al., 2013; Bellassen and Stephan, 2015; Andrew and Cortese, 2011; Tang and Demeritt, 2018; Cotter et al., 2011; Tol, 2011; U.S. Environment Protection Agency, 2013; Ricke et al., 2018).

nuances in the estimation methods may make them inaccurate—particularly in experiments which utilize combined CPU and GPU workloads heavily. We build a framework aiming to provide more accurate and easier systematic reporting of carbon and energy footprints. We also provide additional mitigation and reporting strategies—beyond those discussed by these prior works—to emphasize how both companies and research labs can be more carbon and energy efficient.

It is worth noting that prior work has also examined the carbon impacts of research in other fields, focusing mostly on emissions from conference travel (Spinellis and Louridas, 2013; Astudillo and AzariJafari, 2018; Hackel and Sparkman, 2018). We provide a brief discussion on ML-related conference travel in Appendix A, but will focus mainly on accurate accounting of energy and carbon footprints of ML compute.

3. Background

We briefly provide a primer on energy and carbon accounting, which form the basis of our proposed framework for measuring and reporting the ecological footprint of ML research.

3.1 Energy Accounting

Energy accounting is fairly straightforward. The energy consumption of a system can be measured in Joules (J) or Watt-hours (Wh),[6] representing the amount of energy needed to power the system. Life-cycle accounting might also consider the energy required to manufacture components of the system—for example, the production of GPUs or CPUs (Jones et al., 2013). However, we largely ignore life-cycle aspects of energy accounting due to the difficulties in attributing manufacturing impacts on a per-experiment basis. Measuring data-center energy impacts also contain several layers, focusing on hardware-centric and software-centric analyses. Many parts contribute to the power consumption of any computational system. Dayarathna et al. (2015) survey energy consumption components of a data center and their relative consumption: cooling (50%), lighting (3%), power conversion (11%), network hardware (10%), and server/storage (26%).

The server and storage component can further be broken down into contributions from DRAM, CPUs, among other compute components. Accurate accounting for all of these components requires complex modeling and varies depending on workload. In particular, the efficiency of the hardware varies with utilization—often most efficient near maximum utilization—making utilization an important factor in optimization (particularly in large cloud compute systems) Barroso et al. (2018). Since we aim to provide a framework at the per-experiment software level, we only account for aspects of energy consumption which expose interfaces for energy metrics (giving us real-time energy usage and compensating for such workload differences). For the purpose of our work, this is constrained to DRAM, CPUs, and GPUs. To account for all other components, we rely on a power usage effectiveness (PUE) factor (Strubell et al., 2019). This factor rescales the available power metrics by an average projected overhead of other components. With more available software interfaces, more robust modeling can be performed as reviewed by Dayarathna et al. (2015).

6. One Watt is a unit of power—equivalent to one Joule per second.

3.2 Carbon Accounting

Carbon accounting can be all-expansive, so we focus on a narrow definition provided by Stechemesser and Guenther (2012): "carbon accounting at the project scale can be defined as the measuring and non-monetary valuation of carbon and GHG emissions and offsetting from projects, and the monetary assessment of these emissions with offset credits to inform project-owners and investors but also to establish standardized methodologies." Carbon and GHG emissions are typically measured in some form close to units CO_{2eq}. This is the amount of carbon—and other GHG converted to carbon amounts—released into the atmosphere as a result of the project. Carbon offsetting is the amount of carbon emissions saved as a result of the project. For example, a company may purchase renewable energy in excess of the energy required for their project to offset for the carbon emissions they contributed. Since our goal is to inform and assess carbon emissions of machine learning systems, we ignore carbon offsetting. Typical carbon offsetting involves the use of Power Purchase Agreements (PPAs) or other similar agreements which may not reflect the current carbon make-up of the power draw (as they may account for future clean energy).[7] Since carbon effects contribute to feedback loops, cutting emissions now will improve the likelihood of preventing further emissions.[8]. We also do not consider carbon accounting in the financial sense, but do provide metrics on monetary impacts through the social cost of carbon (SC-CO2). The U.S. Environment Protection Agency (2013) uses this metric when developing administrative rules and regulations. According to the EPA, "The SC-CO2 is a measure, in dollars, of the long-term damage done by a ton of carbon dioxide (CO2) emissions in a given year. This dollar figure also represents the value of damages avoided for a small emission reduction (i.e., the benefit of a CO2 reduction)." We rely on the per-country social cost of carbon developed by Ricke et al. (2018), which accounts for different risk profiles of country-level policies and GDP growth in their estimates of SC-CO2.

Carbon emissions from a project can also consider life-cycle emissions (for example, manufacturing of CPUs may emit carbon as part of the process). We do not consider these aspects of emissions. We instead, consider only carbon emissions from energy consumption. A given energy grid powering an experiment will have a carbon intensity: the grams of CO_{2eq} emitted per kWh of energy used. This carbon intensity is determined based on the energy sources supplying the grid. Each energy source has its own carbon intensity accounted for through a full life-cycle analysis (IPCC, 2015). For example, coal power has a median carbon intensity of 820 gCO_{2eq}/ kWh, while hydroelectricity has a mean carbon intensity of 24 gCO_{2eq}/ kWh. The life-cycle emissions of energy source take into account not just emissions from production, but from waste disposal as well. For example, nuclear energy waste disposal has some carbon emissions associated that would be taken into account in a life-cycle carbon intensity metric (IPCC, 2018). Carbon emissions for a compute system can be estimated by understanding the carbon intensity of the local energy grid and the energy consumption of the system. Similar analyses have been done for bitcoin (Krause and Tolaymat, 2018). These analyses, however, attempt to extrapolate impacts of bitcoin

7. See discussion in Appendix C for further information.
8. See, e.g., `https://www.esrl.noaa.gov/gmd/outreach/info_activities/pdfs/TBI_understanding_feedback_loops.pdf`

mining in general, while in this work we attempt to examine machine learning impacts on a per-experiment basis.

3.3 Current State of Reporting in Machine Learning Research

We briefly examine the current state of accounting in the machine learning literature and review commonly reported computational metrics. Here we look at a non-exhaustive list of reported metrics from papers we surveyed and group them into different categories:

- Energy
  - Energy in Joules (Assran et al., 2019)
  - Power consumption in Watts (Canziani et al., 2016)
- Compute
  - PFLOPs-hr (Amodei and Hernandez, 2018), the floating point operations per second needed to run the experiment in one hour
  - Floating Point Operations (FPOs) or Multiply-Additions (Madds), typically reported as the computations required to perform one forward pass through a neural network (Howard et al., 2017; Sandler et al., 2018; Schwartz et al., 2019)
  - The number of parameters defined by a neural network (often reported together with FPOs) (Howard et al., 2017; Sandler et al., 2018)
  - GPU/CPU utilization as a percentage (Assran et al., 2019; Dalton et al., 2019)
  - GPU-hours or CPU-hours, the processor cycles utilized (or in the case of the GPU percentage utilized), times the runtime (Soboczenski et al., 2018)
- Runtime
  - Inference time, the time it takes to run one forward pass through a neural network, (Jeon and Kim, 2018; Qin et al., 2018)
  - Wall clock training time, the total time it takes to train a network (Assran et al., 2019; Dalton et al., 2019).
  - Hardware and time together (e.g., 8 v100 GPUs for 5 days) (Krizhevsky et al., 2012; Ott et al., 2018; Gehring et al., 2017)
- Carbon Emissions
  - US-average carbon emissions (Strubell et al., 2019)

Example 1 *To get a rough estimate of the prevalence of these metrics, we randomly sampled 100 NeurIPS papers from the 2019 proceedings. In addition to the metrics above, we also investigate whether hardware information was reported (important for extrapolating energy and carbon information with partial information). Of these papers, we found 1 measured energy in some way, 45 measured runtime in some way, 46 provided the hardware used, 17 provided some measure of computational complexity (e.g., compute-time, FPOs, parameters), and 0 provided carbon metrics. See Appendix B for more details on methodology.*

Some of these metrics, when combined, can also be used to roughly estimate energy or carbon metrics. For example, the experiment time (h) can be multiplied by the thermal design power (TDP) of the GPUs used (W)[9]. This results in a Watt-hour energy metric. This can then be multiplied by the carbon intensity of the local energy grid to assess the amount of CO_{2eq}emitted. This method of estimation omits CPU usage and assumes a 100% GPU utilization. Alternatively, Amodei and Hernandez (2018) use a utilization factor of 33% for GPUs. Similarly, the PFLOPs-hr metric can by multiplied by TDP (Watts) and divided by the maximum computational throughput of the GPU (in PFLOPs). This once again provides a Watt-hour energy metric. This, however, makes assumptions based on maximum efficiency of a GPU and disregards variations in optimizations made by underlying frameworks (e.g., Tensorflow versus Pytorch; AMD versus NVIDIA drivers).

As we will demonstrate using our framework (see Section 5.2), the assumptions of these estimation methods lead to significant inaccuracies. However, aggregating all necessary accounting information is not straightforward or easy; it requires finding compatible tools, handling nuances on shared machines, among other challenges.

It is worth noting that some metrics focus on the computational requirements of training (which require additional resources to compute gradients and backpropagate, in the case of neural networks) versus the computational requirements of inference. The former is often more energy and carbon intensive in machine learning research, while the later is more intensive in production systems (the cost of training is insignificant when compared to the lifetime costs of running inference millions of times per day, every day). We will remain largely agnostic to this differentiation until some discussions in Sections 6.2 and 7.2.

4. A New Framework for Tracking Machine Learning Impacts

4.1 Motivation

The goal of our *experiment-impact-tracker* framework is to provide an easy to deploy, reproducible, and quickly understood mechanism for all machine learning papers to report carbon impact summaries, along with additional appendices showing detailed energy, carbon, and compute metrics.

Example 2 *A carbon impact summary generated by our framework can be found at the end of this paper in the Carbon Impact Statement section. In brief, the experiments in our paper contributed 8.021 kg of CO_{2eq} to the atmosphere and used 24.344 kWh of electricity, having a USA-specific social cost of carbon of $0.38 ($0.00, $0.95) (Ricke et al., 2018).*

Such statements and informational reporting are important for, among other reasons, awareness, aligning incentives, and enabling accurate cost-benefit analyses.

Awareness: Informational labels and awareness campaigns have been shown to be effective drivers of eco-friendly behaviors (depending on the context) (Banerjee and Solomon, 2003; Sundar et al., 2018; Newell and Siikamäki, 2014; Byerly et al., 2018). Without consistent and accurate accounting, many researchers will simply be unaware of the impacts their models might have and will not pursue mitigating strategies. Consistent reporting also may provide social incentives to reduce carbon impacts in research communities.

9. This is a rough estimate of the maximum operating capacity of a GPU.

Aligning Incentives: While current reporting often focuses solely on performance metrics (accuracy in classification, perplexity in language modeling, average return in reinforcement learning, etc), standardized reporting of energy in addition to these metrics aligns incentives towards energy efficient models in research output (Schwartz et al., 2019). Those who accurately report carbon emissions may have more incentive to reduce their carbon footprint. This may also drive traffic to low-emission regions, spurring construction of more carbon-friendly data centers.[10]

Cost-Benefit Analysis and Meta-Analysis: Cost-benefit analyses can be conducted with accurate energy metrics reporting, but are impossible without it. For example, the estimated generated revenue of a model can be weighed against the cost of electricity. In the case of models suggested by Rolnick et al. (2019), the carbon emissions saved by a model can be weighed against the emissions generated by the model. Consistent reporting also opens the possibility for performing meta-analyses on energy and carbon impacts (Henderson and Brunskill, 2018). Larger extrapolations to field-wide impacts of research conferences can also be assessed with more frequent reporting.

4.2 Design Considerations

We consider five main principles when designing the framework for systematic reporting: usability, interpretability, extensibility, reproducibility, and fault tolerance.

Usability: Perceived ease-of-use can be an important factor in adoption of new technologies and methods (Gefen and Straub, 2000). Since gathering key energy (kWh) and carbon (CO_{2eq}) metrics requires specific knowledge about—and aggregation of—different sources of information, there may be a barrier to the ease-of-use in the current status quo. As a result, a core design consideration in developing tools for these metrics is usability, or ease-of-use. We accomplish this by abstracting away and distilling required knowledge of information sources, keeping amount of required action from the user to a minimum.

Interpretability: Along with ease-of-use, a key factor in adoption is perceived usefulness (Gefen and Straub, 2000). Since we wish for the reporting of carbon and energy metrics to become widespread, we consider perceived usefulness through interpretability. We aim to make reporting tools within the framework useful through simple generation of graphs and web pages from metrics for easy interpretation. We also provide a mechanism to generate a carbon impact statement with the social cost of carbon. This dollar amount represents the projected damage from the experiment's carbon emissions and helps ground results in values that may be more interpretable. As seen in our own statement at the end of this work, we also provide the carbon impact and energy usage directly.

Extensibility: We design the framework in a modular fashion to handle evolving driver support (see Section 5) and new metrics. To improve the accuracy and accessibility of the framework, the ML community can add new metrics, carbon intensity information, and other capabilities easily. For each metric, a central data router stores a description, the function which gathers metric data, and a list of compatibility checks (e.g., the metric can only be gathered on a Linux system). New metrics can be added to this router.[11] Similarly, new

10. See discussion in Section 6.2 on regional carbon emission differences. See discussion by LaRiviere et al. (2016) on how more accurate carbon accounting can result in reduced carbon emissions.
11. See `https://breakend.github.io/experiment-impact-tracker/contributing_new_metric.html`

carbon region and electricity grid information can be added as needed to similar centralized locations.[12]

Reproducibility: Running an algorithm on different sets of hardware has been shown to affect the reproducibility of algorithmic results (Gundersen and Kjensmo, 2018; Sukhoy and Stoytchev, 2019). Our framework aides in automating reproducibility by logging additional metrics like hardware information, Python package versions, etc. These metrics can help future work assess statistically significant differences in model energy requirements by accounting for controlled and random variates (Boquet et al., 2019).

Fault tolerance: Mistakes in software are inevitable—as is discussed in Sidor and Schulman (2017). We try to log all *raw* information so that accounting can be recreated and updated based on new information. We also log the version number of the tool itself, to ensure future comparisons do not mismatch information between versions that may have changed.

4.3 Proposed Framework

The *experiment-impact-tracker* requires a simple code change to automatically gather available metrics and a script to generate online appendices for reporting the data. Currently, on compatible systems, we gather:

- all python packages and version numbers
- CPU and GPU hardware information
- experiment start and end-times
- the version of the *experiment-impact-tracker* framework used
- the energy grid region the experiment is being run in (based on IP address)
- the average carbon intensity in the energy grid region
- CPU- and GPU-package power draw
- per-process utilization of CPUs and GPUs
- GPU performance states
- memory usage
- the realtime CPU frequency (in Hz)
- realtime carbon intensity (only supported in CA right now)
- disk write speed

The code change required for immediate logging of metrics can be seen in Listing 1. In the background, the framework launches a thread which polls system supported tools. For example, the thread polls *psutil* (Rodola, 2016) for measuring CPU utilization. All of these

12. See `https://breakend.github.io/experiment-impact-tracker/contributing_carbon_region.html`.

metrics are logged in parallel with the main machine learning process as described in Figure 1. A script[13] is provided to generate an HTML web page showing graphs and tables for all these metrics, meant to serve as an online appendix for research papers.[14] Results in the generated appendix can be aggregated across multiple experiments to show averages along with standard error as recommended in prior work (Henderson et al., 2018; Colas et al., 2018; Reimers and Gurevych, 2017).

```
from experiment_impact_tracker.compute_tracker import ImpactTracker
tracker = ImpactTracker(<your log directory here>)
tracker.launch_impact_monitor()
```

Listing 1: Simple code addition required to log experiment details via our framework.

4.3.1 Tracking Energy Consumption

Different hardware vendors provide different tooling for tracking energy consumption. Our framework hides these complications from users. We currently use Intel's RAPL tool with the powercap interface (David et al., 2010) or Intel's PowerGadget Tool[15] (depending on availability) to gather CPU/DRAM power draw and Nvidia's *nvidia-smi*[16] for GPU power draw. We use *psutil* for gathering per-process CPU utilization and *nvidia-smi* for per-process GPU utilization. We found that on a shared machine—as when running a job on Slurm—using Intel's RAPL would provide energy metrics for the entire machine (including other jobs running on the worker). If two experiments were launched with Slurm to the same worker, using measurements from RAPL without corrections would double count energy usage from the CPU.

As a result, we assign energy credits on a per-process basis (though we log system-wide information as well). We track the parent process, and any children spawned. Power credits are provided based on relative usage of system resources. If a process uses 25% of the CPU (relative to the entire system's usage), we will credit the process with 25% of the CPU-based power draw. This ensures that any non-experiment-related background processes— software updates, weekly jobs, or multiple experiments on the same machine—will not be taken into account during training.

We calculate total energy as:

$$e_{\text{total}} = \text{PUE} \sum_{p} (\text{p}_{\text{dram}} e_{\text{dram}} + \text{p}_{\text{cpu}} e_{\text{cpu}} + \text{p}_{\text{gpu}} e_{\text{gpu}}), \tag{1}$$

where p_{resource} are the percentages of each system resource used by the attributable processes relative to the total in-use resources and e_{resource} is the energy usage of that resource. This is the per-process equivalent of the method which Strubell et al. (2019) use.

13. https://github.com/Breakend/experiment-impact-tracker/blob/master/scripts/create-compute-appendix
14. Appendices generated by our framework for Figure 7 and Figure 3 are available at: https://breakend.github.io/ClimateChangeFromMachineLearningResearch/measuring_and_mitigating_energy_and_carbon_footprints_in_machine_learning/. Experiments in Figure 5 are available at https://breakend.github.io/RL-Energy-Leaderboard/reinforcement_learning_energy_leaderboard/index.html.
15. https://software.intel.com/content/www/us/en/develop/articles/intel-power-gadget.html
16. https://developer.nvidia.com/nvidia-system-management-interface

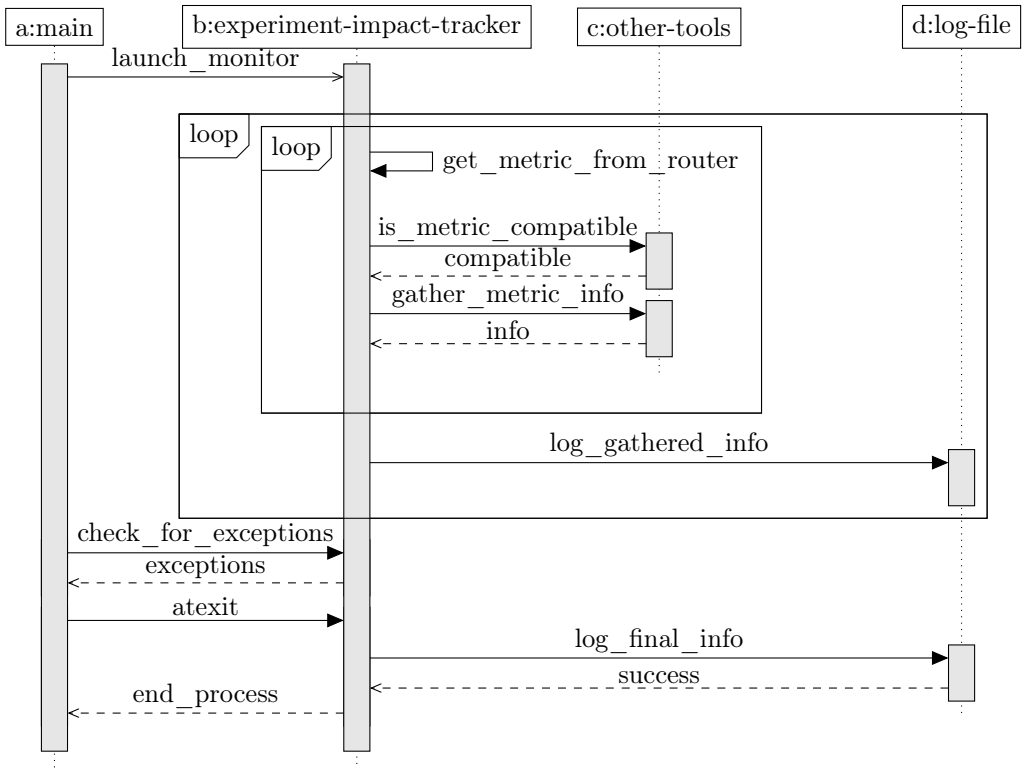

Figure 1: A diagram demonstrating how the released version of the tool works. The main process launches a monitoring thread which iterates over a list of metrics associated with function calls to other tools. For example, if available, we call Intel RAPL to collect CPU power draw or query `caiso.org` to get realtime carbon intensity data for California. Once all the data that is compatible with the current system is gathered, it is logged to a standardized log file and the process repeats. The main thread may check in on this thread for exceptions, but the thread will not interrupt the main process. Once the main thread exits, an *atexit* hook (which is called whenever the main process exits, either successfully or through an exception) gathers the final information (such as the time the experiment ended), logs it, and then ends both the monitor and main process.

We assume the same constant power usage effectiveness (PUE) as Strubell et al. (2019) to be the framework's default PUE. This value compensates for excess energy from cooling or heating the data-center. Users can customize the PUE value when using the framework if needed.

4.3.2 Carbon Accounting

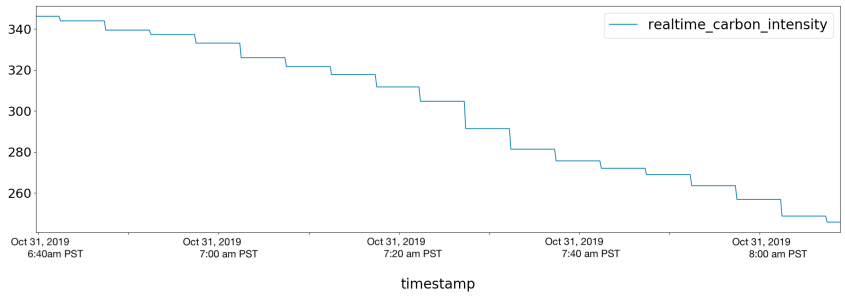

Figure 2: Realtime carbon intensity (gCO_{2eq}/kWh) collected during one experiment using our framework. As the experiment continued, the sun rose in California, and with it the carbon intensity decreased.

For calculating carbon emissions, we use the power estimate from the previous section in kilowatt-hours (kWh) and multiply it by the carbon intensity of the local energy grid (g CO_{2eq}/ kWh). To gather carbon intensity metrics for energy grids, we build on the open-source portions of `https://www.electricitymap.org` and define regions based on map-based geometries, using the smallest bounding region for a given location as the carbon intensity estimate of choice. For example, for an experiment run in San Francisco, if the average carbon intensity is available for both the USA and California, the latter will be used. We estimate the region the experiment is conducted in based on the machine's IP address. Carbon intensities are gathered from the average fallback values provided in the `https://www.electricitymap.org` code where available and supplemented with additional metrics from various governmental or corporate reports. We note that `electricitymap.org` estimates are based on a closed-source system and uses the methodology described by Tranberg et al. (2019). All estimates from `electricitymap.org` are of the regional supply, rather than production (accounting for imports from other regions). Since `https://caiso.com` provides realtime intensities including imports for free, for experiments run in California, we also provide realtime carbon intensity information. We do this by polling `https://caiso.com` for the current intensity of the California energy grid every five minutes. This helps gather even more accurate estimates of carbon emissions to account for daily shifts in supply. For example, experiments run in California during the day time use roughly $\frac{2}{3}$ of night-time experiments. This is because much of California's renewable energy comes from solar plants. Figure 2 is an automatically generated graph showing this phenomenon from an experiment using our framework. We hope that as users find more accurate realtime or average measurements

of regional supply-based carbon intensities, they will add them to the tool for even more accurate measurements in the future.

5. The Importance and Challenges of Accounting: Why a New Framework?

5.1 FPOs Can Be Misleading

Floating Point Operations (FPOs) are the de facto standard for reporting "efficiency" of a deep learning model (Schwartz et al., 2019), and intuitively they should be correlated with energy efficiency—after all, fewer operations should result in faster and more energy efficient processing. However, this is not always the case.

Previously, Jeon and Kim (2018) demonstrated mechanisms for constructing networks with larger FPOs, but lower inference time—discussing the "Trap of FLOPs". Similarly, Qin et al. (2018) show how Depthwise 3x3 Convolutions comprised just 3.06% of an example network's Multiply-Add operations, while utilizing 82.86% of the total training time in the FPO-efficient MobileNet architecture Howard et al. (2017). Underlying optimizations at the firmware, deep learning framework, memory, or even hardware level can change energy efficiency and run-time. This discrepancy has led to Github Issues where users expect efficiency gains from FPO-efficient operations, but do not observe them.[17] This has also been observed by Chen and Gilbert (2018) and Chen et al. (2018).

Example 3 *To investigate this empirically, we repeatedly run inference through pre-trained image classification models and measure FPOs, parameters, energy usage, and experiment length using the* experiment-impact-tracker *framework. As described in Figure 3, we find little correlation between FPOs and energy usage or experiment runtime when comparing across different neural network architectures. However, within an architecture—relying on the same operation types, but with different numbers of operations—FPOs are almost perfectly correlated with energy and runtime efficiency. Thus, while FPOs are useful for measuring relative ordering within architecture classes, they are not adequate on their own to measure energy or even runtime efficiency.*

5.2 Estimates with Partial Information Can Be Inaccurate

The current state of accounting for energy and carbon varies across fields and papers (see Section 3). Few works, if any, report all of the metrics that our framework collects. However, it is possible to extrapolate energy and carbon impacts from some subsets of these metrics. This can give a very rough approximation of the energy used by an experiment in kWh (see Section 3 for background).

Example 4 *We demonstrate how several such estimation methods compare against the more fine-grained accounting methods we describe in Section 4.*[18] *As seen in Figure 4, we find*

17. See for example: https://github.com/tensorflow/tensorflow/issues/12132 and https://github.com/tensorflow/tensorflow/issues/12940
18. We also provide a script to do the rough calculation of energy and carbon footprints based on GPU type, IP address (which is used to retrieve the location of the machine and that region's carbon

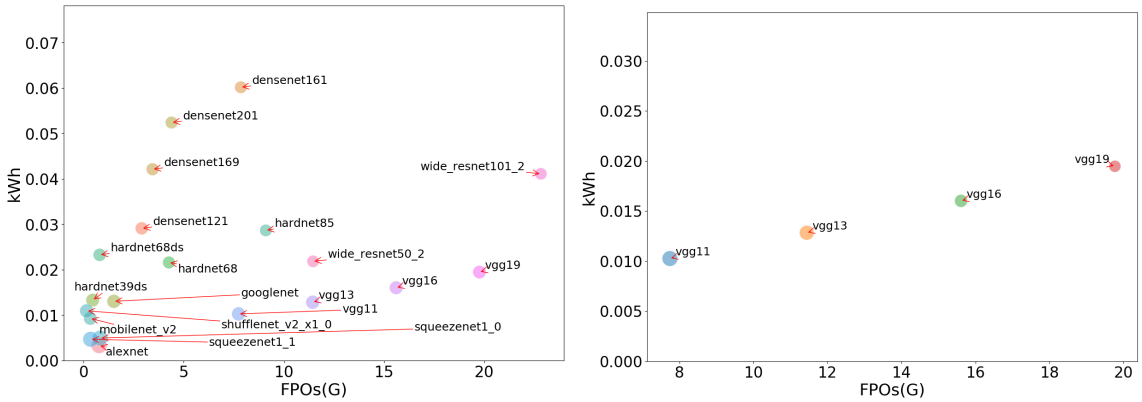

Figure 3: We run 50,000 rounds of inference on a single sampled image through pre-trained image classification models and record kWh, experiment time, FPOs, and number of parameters (repeating 4 times on different random seeds). References for models, code, and expanded experiment details can be found in Appendix D. We run a similar analysis to Canziani et al. (2016) and find (left) that FPOs are not strongly correlated with energy consumption ($R^2 = 0.083$, Pearson 0.289) nor with time ($R^2 = 0.005$, Pearson -0.074) when measured across different architectures. However, within an architecture (right) correlations are much stronger. Only considering different versions of VGG, FPOs are strongly correlated with energy ($R^2 = .999$, Pearson 1.0) and time ($R^2 = .998$, Pearson .999). Comparing parameters against energy yields similar results (see Appendix D for these results and plots against experiment runtime).

significant differences from when we track all data (as through the experiment-impact-tracker *framework) to when we use partial data to extrapolate energy and carbon emissions. Only using GPUs and the experiment time ignores memory or CPU effects; only using the average case US region ignores regional differences. More details for this experiment can be found in Appendix E.*

We also note that the possible estimation differences in Figure 4 do not include possible errors from counting multiple processes at once, as described in Section 4.3.1. Clearly, without detailed accounting, it is easy to severely over- or underestimate carbon or energy emissions by extrapolating from partial information.

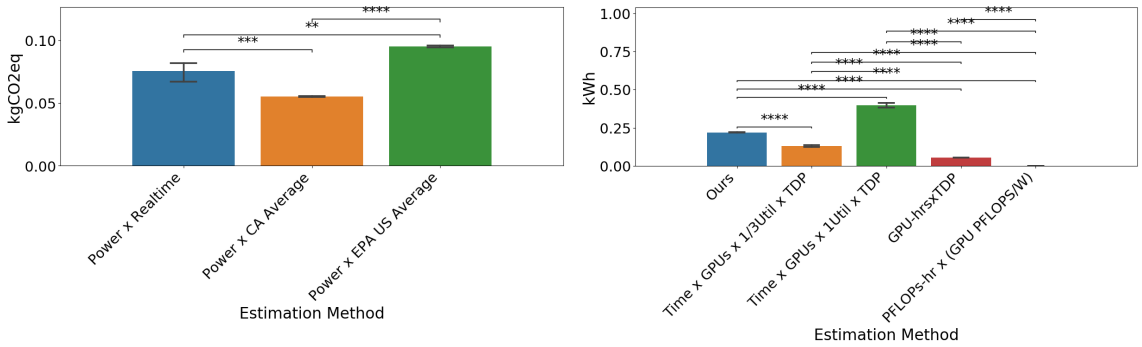

Figure 4: We compare carbon emissions (left) and kWh (right) of our Pong PPO experiment (see Appendix E for more details) by using different estimation methods. By only using country wide or even regional average estimates, carbon emissions may be over or under-estimated (respectively). Similarly, by using partial information to estimate energy usage (right, for more information about the estimation methods see Appendix E), estimates significantly differ from when collecting all data in real time (as in our method). Clearly, without detailed accounting, it is easy to over- or under-estimate carbon or energy emissions in a number of situations. Stars indicate level of significance: * $p < .05$, ** $p < .01$, *** $p < .001$, **** $p < .0001$. Annotation provided via: https://github.com/webermarcolivier/statannot.

6. Encouraging Efficiency and Mitigating Carbon Impacts: Immediate Mitigation Strategies

With *experiment-impact-tracker*, we hope to ease the burden of standardized reporting. We have demonstrated differences in more detailed estimation strategies from the current status quo. In this Section, we examine how accurate reporting can be used to drive immediate mitigating strategies for energy consumption and carbon emissions.

intensity), experiment length, and utilization factor. https://github.com/Breakend/experiment-impact-tracker/blob/master/scripts/get-rough-emissions-estimate

6.1 Energy Efficiency Leaderboards

A body of recent work has emphasized making more computationally efficient models (Wu et al., 2019; Zhou et al., 2020; Reddi et al., 2020; Lu et al., 2018; Coleman et al., 2019; Jiang et al., 2019), yet another line of work has focused on the opposite: building larger models with more parameters to tackle more complex tasks (Amodei and Hernandez, 2018; Sutton, 2019). We suggest leaderboards which utilize carbon emissions and energy metrics to promote an informed balance of performance and efficiency. DawnBench (Wu et al., 2019), MLPerf (Reddi et al., 2020), and HULK (Zhou et al., 2020) have done this in terms of runtime and cost, but by doing the same for energy and carbon emissions directly, baseline implementations can converge to more efficient climate-friendly settings. This can also help spread information about the most energy and climate-friendly combinations of hardware, software, and algorithms such that new work can be built on top of these systems instead of more energy-hungry configurations.[19]

6.1.1 A Deep RL Energy Leaderboard

To demonstrate how energy leaderboards can be used to disseminate information on energy efficiency, we create a Deep RL Energy Leaderboard.[20] The website is generated using the same tool for creating HTML appendices described in Section 4. All information (except for algorithm performance on tasks) comes from the *experiment-impact-tracker* framework. We populate the leaderboard for two common RL benchmarking environments, PongNoFrameskip-v4 and BreakNoFrameskip-v4 (Bellemare et al., 2013; Brockman et al., 2016; Mnih et al., 2013), and four baseline algorithms, PPO (Schulman et al., 2017), A2C (Mnih et al., 2016), A2C with V-Traces (Espeholt et al., 2018; Dalton et al., 2019), and DQN (Mnih et al., 2013). The experimental details and results can also be found in Figure 5. We find that no algorithm is the energy efficiency winner across both environments, though the PPO implementation provided by Hill et al. (2018) attains balance between efficiency and performance when using default settings across algorithms.

Example 5 *To see how such a leaderboard might help save energy, consider a Deep RL class of 235 students.*[21] *For a homework assignment, each student must run an algorithm 5 times on Pong. The class would save 888 kWh of energy by using PPO versus DQN, while achieving similar performance.*[22] *This is roughly the same amount needed to power a US home for one month.*[23]

19. Something to note is that we do not compare carbon efficiency directly—instead focusing on energy specifically. Since running at different times of day and in different regions can affect carbon impacts, these may not have anything to do with the algorithm hardware-software stack and increase the number of confounds when comparing algorithms. While hardware is also immutable to some extent, there may still be information to be gained by finding combinations of efficient low-level optimizations for specific hardware. Hardware can also be held relatively constant by using the same machine for all experimental runs. If comparisons using carbon units are desired, a fixed carbon intensity factor should likely be chosen for approximate comparisons in a given region (rather than using live carbon intensity metrics). See, also, Appendix H.
20. `https://breakend.github.io/RL-Energy-Leaderboard/reinforcement_learning_energy_leaderboard/index.html`
21. See for example, Stanford's CS 234.
22. These rankings may change with different code-bases and hyperparameters.
23. https://www.eia.gov/tools/faqs/faq.php?id=97&t=3

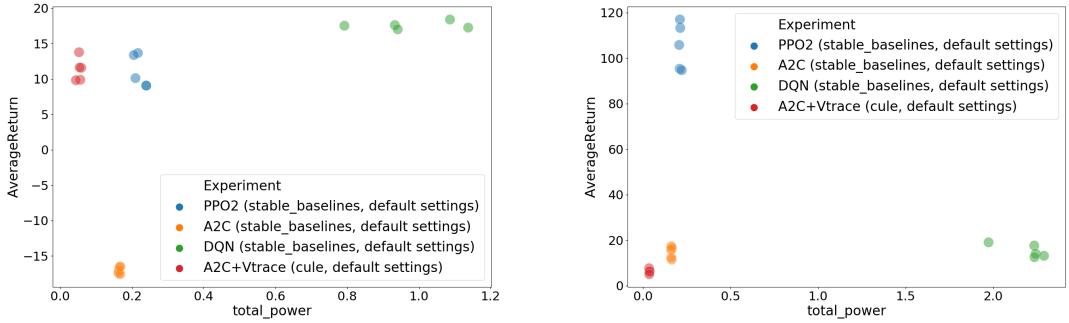

Figure 5: We evaluate A2C, PPO, DQN, and A2C+VTraces on PongNoFrameskip-v4 (left) and BreakoutNoFrameskip-v4 (right), two common evaluation environments included in OpenAI Gym. We train for only 5M timesteps, less than prior work, to encourage energy efficiency and evaluate for 25 episodes every 250k timesteps. We show the Average Return across all evaluations throughout training (giving some measure of both ability and speed of convergence of an algorithm) as compared to the total energy in kWh. Weighted rankings of Average Return per kWh place A2C+Vtrace first on Pong and PPO first on Breakout. Using PPO versus DQN can yield significant energy savings, while retaining performance on both environments (in the 5M samples regime). See Appendix F for more details and results in terms of asymptotic performance.

We, thus, encourage the community to submit more data to the leaderboard to find even more energy efficient algorithms and configurations.

6.2 Running In Carbon-Friendly Regions

We noted in Section 4 that it is important to assess which energy grid experiments are run on due to the large differences in carbon emissions between energy grids. Figure 6 shows CO_{2eq}intensities for an assortment of locations, cloud-provider regions, and energy production methods. We note that an immediate drop in carbon emission can be made by moving all training jobs to carbon-efficient energy grids. In particular, Quebec is the cleanest available cloud region to our knowledge. Running a job in Quebec would result in carbon emission 30x lower than running a job in Estonia (based on 2017 averages).

Example 6 *To demonstrate this in practice, we run inference on two translation models 1000 times and measure energy usage. We extrapolate the amount of emissions and the difference between the two algorithms if run in different energy grids, seen in Figure 7. The absolute difference in emissions between the two models is fairly small (though significant) if run in Quebec (.09 g CO_{2eq}), yet the gap increases as one runs the jobs in less carbon-friendly regions (at 3.04 g CO_{2eq} in Estonia).*

We provide a script with our framework to show all cloud provider region with emission statistics to make this decision-making process easier.[24] We note that Lacoste et al. (2019) provide a website using partial information estimation to extrapolate carbon emissions based on cloud provider region, GPU type, and experiment length in hours. Their tool may also be used for estimating carbon emissions in cloud-based experiments ahead of time. We've also provided a non-exhaustive list of low emissions energy grids that contain cloud regions in Table 1.

For companies that train and deploy large models often, shifting these resources is especially important. ML training is not usually latency bound: companies can run training in cloud regions geographically far away since training models usually does not require round trip communication requirements. Contrary to some opinions,[25] there is not a necessary need to eliminate computation-heavy models entirely, as shifting training resources to low carbon regions will immediately reduce carbon emissions with little impact to production systems. For companies seeking to hit climate change policy targets, promotion of carbon neutral regions and shifting of all machine learning systems to those regions would accelerate reaching targets significantly and reduce the amount of offset purchasing required to meet goals (thus saving resources).[26] It is worth noting that some companies like Google already purchase offsets (Google, 2016), so it may be unclear why shifting resources is necessary. We provide an extended discussion on this in Appendix C. As a matter of total emissions reductions, running compute in carbon-friendly regions prevents emissions now, while offsets may not come into effect for several years. Moreover, continuing offset purchasing at current levels, while shifting resources to green regions would result in a net-negative carbon footprint.

24. See: get-region-emissions-info script and lookup-cloud-region-info script.
25. https://www.theguardian.com/technology/2019/sep/17/tech-climate-change-luddites-data
26. See, for example, Amazon's goal: https://press.aboutamazon.com/news-releases/news-release-details/amazon-co-founds-climate-pledge-setting-goal-meet-paris

Power Grid	Cloud Regions	Carbon Intensity (g CO_{2eq}/ kWh)
Quebec, Canada	ca-central-1 (AWS), canadaeast (Azure), northamerica-northeast1 (GCP)	~ 30
West Norway	norwaywest (Azure)	~ 35
Ontario, Canada	canadacentral (Azure)	~ 45
France	eu-west-3 (AWS), francesouth (Azure), francecentral (Azure)	~ 56
Brazil (Central)	brazilsouth (Azure)	~ 106
Oregon, USA	us-west1 (GCP), us-west-2 (AWS) westus2 (Azure)	~ 127

Table 1: A non-exhaustive list of cloud regions in low carbon intensity energy grids (< 150 gCO_{2eq}/ kWh). All estimates pulled as yearly averages from `https://www.electricitymap.org/map`, except for Quebec which utilizes methodology from `https://piorkowski.ca/rev/2017/06/canadian-electricity-co2-intensities/` and Oregon which uses data from `https://www.eia.gov/electricity/state/oregon/`.

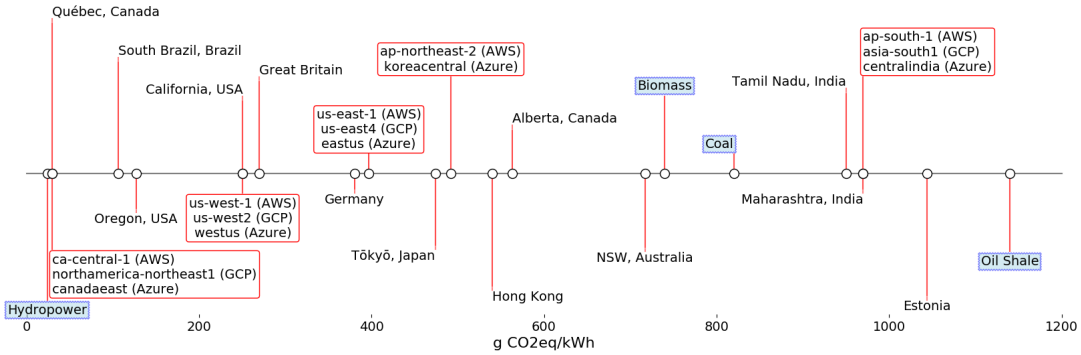

Figure 6: Carbon Intensity (gCO_{2eq}/kWh) of selected energy grid regions is shown from least carbon emissions (left) to most carbon emissions (right). Red/unshaded boxes indicate carbon intensities of cloud provider regions. Blue/shaded boxes indicate carbon intensities of various generation methods. Oil shale is the most carbon emitting method of energy production in the Figure. Estonia is powered mainly by oil shale and thus is close to it in carbon intensity. Similarly, Québec is mostly powered by hydroelectric methods and is close to it in carbon intensity. Cloud provider carbon intensities are based on the regional energy grid in which they are located. Thus, us-west-1, located in California, has the same carbon intensity as the state. See `https://github.com/Breakend/experiment-impact-tracker/` for data sources of regional information. Energy source information from Krey et al. (2014); International Energy Agency (2015).

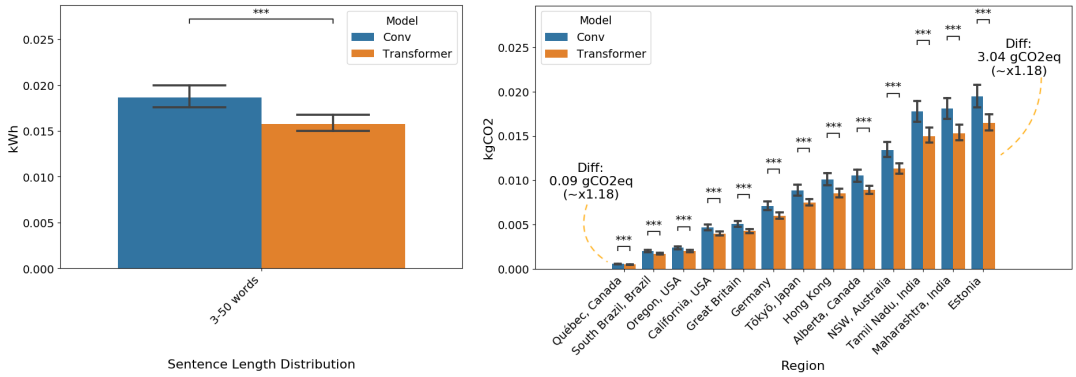

Figure 7: We use pre-trained En-Fr translation models downloaded from PyTorch Hub: a convolutional network (Gehring et al., 2017) and transformer (Ott et al., 2018). We generate 1000 random sequences either between 3-50 words in length using the essential_generators Python package: `https://pypi.org/project/essential-generators/`. We repeat with 20 random seeds. [Left] We show the true difference in energy consumption. [Right] We show estimated $kgCO_{2eq}$released if the experiment had been conducted in a number of increasingly carbon-intensive energy grids. Differences remain significant throughout, but the absolute difference increases as more carbon-intensive regions are assumed.

7. Discussion: Systemic Changes

We demonstrated several use cases for accounting which can drive immediate mitigation strategies. However, the question remains: how can we encourage systemic changes which lead to energy and carbon efficiency in ML systems?

7.1 Green Defaults for Common Platforms and Tools

Energy leaderboards help provide information on energy efficient configurations for the whole stack. However, to truly spread energy efficient configurations, underlying frameworks should by default use the most energy-efficient settings possible. This has been shown to be an effective way to drive pro-environmental behavior (Pichert and Katsikopoulos, 2008). For example, Nvidia apex provides easy mixed-precision computing as an add-on which yields efficiency gains.[27] However, it requires knowing this and using it. Merity (2019) also discusses the current difficulties in using highly efficient components. Making such resources supported as defaults in frequently used frameworks, like PyTorch, would immediately improve the efficiency of all downstream projects. We encourage maintainers of large projects to prioritize and support such changes.

27. `https://github.com/NVIDIA/apex`

7.2 How Much Is Your Performance Gain Worth? Balancing Gains With Cost

While training jobs can easily be shifted to run in clean regions, there are often restrictions for inference-time use of machine learning models which prevent such a move. Many companies are deploying large machine learning models powered by GPUs for everyday services.[28]

Example 7 *Production translation services, can process 100B words per day (Turovsky, 2016): roughly 4.2 million times our experiment in Figure 7. If all translation traffic were in Estonia, 12,768 kgCO_{2eq} (the carbon sequestered by 16.7 acres of forest in one year (Agency, 2008)) would be saved per day by using the more efficient model, yet if all traffic were in Québec, 378 kgCO_{2eq} would be saved (the carbon sequestered by .5 acres of forest in one year (Agency, 2008)). Considering the amounts of required compute, small differences in efficiency can scale to large emissions and energy impacts.*

These services are latency-bound at inference time and thus cannot mitigate carbon emissions by shifting to different regions. Instead, deploying energy-efficient models not only reduces carbon emissions but also benefits the companies by bringing the energy costs down. We encourage companies to consider weighing energy costs (both social and monetary) with the performance gains of a new model before deploying it. In the case of our translation experiment in Figure 7, the pre-trained convolutional model we use is significantly more energy hungry across than the transformer model we use. When deploying a new energy-hungry translation model, we ask companies to consider is the BLEU score improvement really worth the energy cost of deploying it? Are there ways to route to different models to balance this trade-off? For example, suppose an energy-hungry model only improves performance in some subset of the data. Routing to this model only in that subset would maximize performance while minimizing energy footprint.[29].

We note that considering such trade-offs is of increased importance for models aiming to reduce carbon emissions as described by Rolnick et al. (2019). Deploying a large deep learning model for, say, improving the energy efficiency of a building, is not worth it if the energy costs of the model outweigh the gains. We also leave an open question to economists to help assess the welfare benefits of gains on a particular machine learning metric (e.g., how much is BLEU score worth in a translation service). This would allow the social welfare of the metric to be balanced against the social cost of carbon (Ricke et al., 2018) for deployment decisions.

Similarly, it is important to consider other types of cost-benefit analyses. Perhaps the carbon impacts of a long (energy-intensive) training time for a large model is worth it if it reduces the lifetime carbon footprint in production (for example, if the model doesn't require expensive fine-tuning procedures in the future). Understanding the tradeoff between the lifetime deployment costs and training costs is important before moving on to extended training runs. As such, we also encourage reporting of both estimated training and deployment

28. See for example, search which now uses transformer networks at both Microsoft and Google. https://www.blog.google/products/search/search-language-understanding-bert/ and https://azure.microsoft.com/en-us/blog/microsoft-makes-it-easier-to-build-popular-language-representation-model-bert-at-large-scale/

29. Efficient routing of traffic to regions has been considered before by Nguyen et al. (2012) and Berral et al. (2010). It may be worth considering efficient routing of traffic to particular models as well.

energy costs so future adopters have a more comprehensive picture when deciding which model to use.

Central to all of these cost-benefit analyses are accurate accounting. Our tool provides one step in consistent and accurate accounting for such purposes.

7.3 Efficient Testing Environments

In Section 7.1 we discuss the adoption of green default configurations and Section 7.2 discusses cost-benefit analyses for deployments. Another consideration particular to research—especially RL—is the selection of the most efficient testing environments which assess the mechanism under test. For example, if an RL algorithm solves a particularly complex task in an interesting way, like solving a maze environment, is there a way to demonstrate the same phenomenon in a more efficient environment? Several works have developed efficient versions of RL environments which reduce run-times significantly. In particular, Dalton et al. (2019) improve the efficiency of Atari experiments by keeping resources on the GPU (and thus avoiding energy and time overheads from moving memory back and forth). Chevalier-Boisvert et al. (2018) develop a lightweight Grid World environment with efficient runtimes for low-overhead experiments. An important cost-benefit question for researchers is whether the same point can be proven in a more efficient setting.

7.4 Reproducibility

A key aspect to our work is helping to promote reproducibility by aiding in consistent reporting of experimental details. We encourage all researchers to release code and models (when it is socially and ethically responsible to do so), to prevent further carbon emissions. Replicating results is an important, if not required, part of research. If replication resources are not available, then more energy and emissions must be spent to replicate results—in the case of extremely large models, the social cost of carbon may be equivalently large. Thus, we ask researchers to also consider energy and environmental impacts from replication efforts, when weighing model and code release. We note that there may very well be cases where safety makes this trade-off lean in the direction of withholding resources, but this is likely rare in most current research. For production machine learning systems, we encourage developers to release models and codebases internally within a company. This may encourage re-use rather than spending energy resources developing similar products.

7.5 Standardized Reporting

We suggest that all papers include standardized reporting of energy and carbon emissions. We also suggest adding a Carbon Impact Statement at the end of papers (just like ours below) which estimates the carbon emissions of the paper. This can be reported in a dollar amount via the country-specific social cost of carbon Ricke et al. (2018). We provide a script[30] to parse logs from the *experiment-impact-tracker* framework and generate such a statement automatically. We suggest this to spread awareness and bring such considerations

30. `https://github.com/Breakend/experiment-impact-tracker/blob/master/scripts/generate-carbon-impact-statement`

to the forefront. We encourage this statement to include *all* emissions from experimentation to build a more realistic picture of total resources spent.

We also emphasize that research, even when compute intensive, is immensely important for progress. It is unknown what sequence of papers may inspire a breakthrough (Stanley and Lehman, 2015) which would reduce emissions by more than any suggestion here. While emissions should be minimized when possible, we suggest that impact statements be only used for awareness. This is especially true since access to clean energy grids or hardware may be limited for some in the community.

We also suggest that, when developing features which visualize compute intensity for cloud or internal workloads, developers consider providing built-in tools to visualize energy usage and carbon emissions. For example, the Colab Research Environment shows RAM and Disk capacity,[31] but could also show and provide access to these other metrics more easily. Providing similar informational labels (Byerly et al., 2018) within internal tooling could mitigate some energy and carbon impacts within companies.

7.6 Badging

Informational labeling has had a long history of being used in public policy (Banerjee and Solomon, 2003). In the USA, the "Energy Star" label has been used to guide customers to eco-friendly products. More recently, "badges" rewarded by the *Psychological Science* journal were shown to be effective, with a jump from 3% of articles reporting open data to 39% one year later. ACM has introduced similar reproducibility badges.[32] With consistent reporting of carbon and energy metrics, climate friendly research badges can be introduced by conferences to recognize any paper that demonstrates a significant effort to mitigate its impacts. For example, a compute intensive paper, when showing evidence of explicitly running resources in a clean region can be rewarded with such a badge. Another example badge can be awarded to papers that create energy-friendly algorithms with similar performance as the state-of-the-art[33]. The goal of these badges is to draw further attention to efficient versions of state-of-the-art systems and to encourage mitigation efforts while, again, not punishing compute-intensive experiments. Of course this may not apply to conferences such as SysML which often focus on making models more efficient, but rather as a motivational tool for other venues where efficiency may not be in focus.

7.7 Limitations and Opportunities for Extensions

The *experiment-impact-tracker* framework abstracts away many of the previously mentioned difficulties in estimating carbon and energy impacts: it handles routing to appropriate tools for collecting information, aggregates information across tools to handle carbon calculations, finds carbon intensity information automatically, and corrects for multiple processes on one machine. Yet, a few other challenges may be hidden by using the framework which remain difficult to circumvent.

As Khan et al. (2018) discuss, and we encounter ourselves, poor driver support makes tracking energy difficult. Not every chipset supports RAPL, nor does every Linux kernel.

31. https://colab.research.google.com/
32. https://www.acm.org/publications/policies/artifact-review-badging
33. See, for example, Clark et al. (2020) which creates a more efficient version of text encoder pre-training.

Intel also does not provide first party supported python libraries for access to measurements. *nvidia-smi* per-process measurements in docker containers are not supported.[34] A body of work has also looked at improving estimates of energy usage from RAPL by fitting a regression model to real energy usage patterns (Povoa et al., 2019; Kavanagh and Djemame, 2019; Ghosh et al., 2013; Song et al., 2013). The Slurm workload manager provides an energy accounting plugin,[35] but requires administrator access to add. For those without access to Slurm, Intel's RAPL supports access to measurements through three mechanisms, but only one of these (the powercap interface only available on some systems) does not require root access (see more discussion by Khan et al. (2018)). To promote widespread reporting, we avoid any tool which requires administrative access or would not be accessible on most Linux systems. Providing better supported tools for user-level access to power metrics would make it possible to more robustly measure energy usage. Aggregating metrics and handling the intricacies of these downstream tools requires time and knowledge. We try to abstract as much of these challenges away in the *experiment-impact-tracker*, though some driver-related issues require driver developer support. However, these issues make it difficult to support every on-premises or cloud machine. As such, we currently only support instances which have Intel RAPL or PowerGadget capabilities for Mac OS and Linux.

We also note that carbon intensities for machines in cloud data centers may not reflect the regional carbon intensities. Some providers buy clean energy directly for some data centers, changing the realtime energy mix for that particular data center. We were unable to find any information regarding realtime energy mixes in such cases and thus could not account for these scenarios. If providers exposed realtime APIs for such information this would help in generating more accurate estimates. Moreover, customized hardware in cloud provider regions does not always provide energy accounting mechanisms or interfaces. If cloud providers supported libraries for custom hardware, this could be used for more detailed accounting in a wider range of cloud-based compute scenarios.

We further discuss other sources of error and issues arising from these difficulties in Appendix G.

8. Concluding Remarks and Recommendations

We have shown how the *experiment-impact-tracker* and associated tools can help ease the burden of consistent accounting and reporting of energy, compute, and carbon metrics; we encourage contribution to help expand the framework. We hope the Deep RL Energy Leaderboard helps spread information on energy efficient algorithms and encourages research in efficiency. While we focus on compute impacts of machine learning production and research, a plethora of other work considers costs of transportation for conferences (Holden et al., 2017; Spinellis and Louridas, 2013; Bossdorf et al., 2010) and compute hardware manufacturing (Venkatesan, 2015). We encourage researchers and companies to consider these other sources of carbon impacts as well. Finally, we recap several points that we have highlighted in mitigating emissions and supporting consistent accountability.

34. https://github.com/NVIDIA/nvidia-docker/issues/179#issuecomment-242150861
35. https://slurm.schedmd.com/acct_gather_energy_plugins.html

What can machine learning researchers do?

- Run cloud jobs in low carbon regions only (see Section 6.2).
- Report metrics as we do here, make energy-efficient configurations more accessible by reporting these results (see Section 7.5).
- Work on energy-efficient systems, create energy leaderboards (see Section 6).
- Release code and models whenever safe to do so (see Section 7.4).
- Integrate energy efficient configurations as defaults in baseline implementations (see Section 7.1).
- Encourage climate-friendly initiatives at conferences (see Sections 7.6 and 7.5).

What can industry machine learning developers and framework maintainers do?

- Move training jobs to low carbon regions immediately. Make default launch configurations and documentation point to low carbon regions (see Section 6.2).
- Provide more robust tooling for energy tracking and carbon intensities (see Section 7.7).
- Integrate energy efficient operations as default in frameworks (see Section 7.1).
- Release code and models (even just internally in the case of production systems) whenever safe to do so (see Section 7.4).
- Consider energy-based costs versus benefits of deploying new models (see Section 7.2).
- Report model-related energy metrics (see Section 7.5).

We hope that regardless of which tool is used to account for carbon and energy emissions, the insights we provide here will help promote responsible machine learning research and practices.

Carbon Impact Statement

This work contributed 8.021 kg of CO_{2eq} to the atmosphere and used 24.344 kWh of electricity, having a USA-specific social cost of carbon of \$0.38 (\$0.00, \$0.95). Carbon accounting information located at: `https://breakend.github.io/ClimateChangeFromMachineLearningResearch/measuring_and_mitigating_energy_and_carbon_footprints_in_machine_learning/` and `https://breakend.github.io/RL-Energy-Leaderboard/reinforcement_learning_energy_leaderboard/index.html`. The social cost of carbon uses models from (Ricke et al., 2018). This statement and carbon emissions information was generated using *experiment-impact-tracker* described in this paper.

References

US Environmental Protection Agency. Greenhouse gas equivalencies calculator, 2008. URL `https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator`.

Judith I Ajani, Heather Keith, Margaret Blakers, Brendan G Mackey, and Helen P King. Comprehensive carbon stock and flow accounting: a national framework to support climate change mitigation policy. *Ecological Economics*, 89:61–72, 2013.

Dario Amodei and Danny Hernandez. AI and Compute. `https://blog.openai.com/openai-five/`, 2018.

Jane Andrew and Corinne Cortese. Accounting for climate change and the self-regulation of carbon disclosures. In *Accounting Forum*, volume 35, pages 130–138. Taylor & Francis, 2011.

Yehia Arafa, Ammar ElWazir, Abdelrahman ElKanishy, Youssef Aly, Ayatelrahman Elsayed, Abdel-Hameed Badawy, Gopinath Chennupati, Stephan Eidenbenz, and Nandakishore Santhi. Verified instruction-level energy consumption measurement for nvidia gpus. In *Proceedings of the 17th ACM International Conference on Computing Frontiers*, pages 60–70, 2020.

Mahmoud ("Mido") Assran, Joshua Romoff, Nicolas Ballas, Joelle Pineau, and Mike Rabbat. Gossip-based actor-learner architectures for deep reinforcement learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 13299–13309. Curran Associates, Inc., 2019.

Miguel F. Astudillo and Hessam AzariJafari. Estimating the global warming emissions of the LCAXVII conference: connecting flights matter. *The International Journal of Life Cycle Assessment*, 23(7):1512–1516, Jul 2018. ISSN 1614-7502.

Abhijit Banerjee and Barry D Solomon. Eco-labeling for energy efficiency and sustainability: a meta-evaluation of us programs. *Energy policy*, 31(2):109–123, 2003.

L. A. Barroso, U. Hölzle, P. Ranganathan, and M. Martonosi. The datacenter as a computer: Designing warehouse-scale machines. *Synthesis Lectures on Computer Architecture*, 2018.

Valentin Bellassen and Nicolas Stephan. *Accounting for Carbon*. Cambridge University Press, 2015.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The Arcade Learning Environment: An Evaluation Platform for General Agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

Josep Ll. Berral, Íñigo Goiri, Ramón Nou, Ferran Julià, Jordi Guitart, Ricard Gavaldà, and Jordi Torres. Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, e-Energy '10, page 215–224, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781450300421.

Thomas Boquet, Laure Delisle, Denis Kochetkov, Nathan Schucher, Parmida Atighehchian, Boris Oreshkin, and Julien Cornebise. Decovac: Design of experiments with controlled variability components. *arXiv preprint arXiv:1909.09859*, 2019.

Oliver Bossdorf, Madalin Parepa, and Markus Fischer. Climate-neutral ecology conferences: just do it! *Trends in Ecology & Evolution*, 25(2):61, 2010.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

Hilary Byerly, Andrew Balmford, Paul J Ferraro, Courtney Hammond Wagner, Elizabeth Palchak, Stephen Polasky, Taylor H Ricketts, Aaron J Schwartz, and Brendan Fisher. Nudging pro-environmental behavior: evidence and opportunities. *Frontiers in Ecology and the Environment*, 16(3):159–168, 2018.

Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. An analysis of deep neural network models for practical applications. *arXiv preprint arXiv:1605.07678*, 2016.

Ping Chao, Chao-Yang Kao, Yu-Shan Ruan, Chien-Hsiang Huang, and Youn-Long Lin. Hardnet: A low memory traffic network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3552–3561, 2019.

Bo Chen and Jeffrey Gilbert. Introducing the CVPR 2018 on-device visual intelligence challenge. `https://ai.googleblog.com/2018/04/introducing-cvpr-2018-on-device-visual.html`, 2018.

Yu-Hsin Chen, Tien-Ju Yang, Joel Emer, and Vivienne Sze. Understanding the limitations of existing energy-efficient design approaches for deep neural networks. In *Proceedings of the 1st SysML Conference*, 2018.

Maxime Chevalier-Boisvert, Lucas Willems, and Suman Pal. Minimalistic Gridworld Environment for OpenAI Gym. `https://github.com/maximecb/gym-minigrid`, 2018.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. {ELECTRA}: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*, 2020.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Cody Coleman, Daniel Kang, Deepak Narayanan, Luigi Nardi, Tian Zhao, Jian Zhang, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. Analysis of DAWNBench, a Time-to-Accuracy Machine Learning Performance Benchmark. *SIGOPS Oper. Syst. Rev.*, 53(1):14–25, July 2019. ISSN 0163-5980.

Julie Cotter, Muftah Najah, and Shihui Sophie Wang. Standardized reporting of climate change information in australia. *Sustainability accounting, management and policy journal*, 2(2):294–321, 2011.

Thomas J Crowley. Causes of climate change over the past 1000 years. *Science*, 289(5477): 270–277, 2000.

Steven Dalton, Iuri Frosio, and Michael Garland. GPU-Accelerated Atari Emulation for Reinforcement Learning, 2019.

Howard David, Eugene Gorbatov, Ulf R Hanebutte, Rahul Khanna, and Christian Le. RAPL: memory power estimation and capping. In *2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED)*, pages 189–194. IEEE, 2010.

Miyuru Dayarathna, Yonggang Wen, and Rui Fan. Data center energy consumption modeling: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794, 2015.

Spencer Desrochers, Chad Paradis, and Vincent M Weaver. A validation of dram rapl power measurements. In *Proceedings of the Second International Symposium on Memory Systems*, pages 455–470, 2016.

Lasse Espeholt, Hubert Soyer, Remi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, et al. IMPALA: Scalable Distributed Deep-RL with Importance Weighted Actor-Learner Architectures. In *International Conference on Machine Learning*, pages 1406–1415, 2018.

David Gefen and Detmar W Straub. The relative importance of perceived ease of use in is adoption: A study of e-commerce adoption. *Journal of the association for Information Systems*, 1(1):8, 2000.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

Sayan Ghosh, Sunita Chandrasekaran, and Barbara Chapman. Statistical modeling of power/energy of scientific kernels on a multi-gpu system. In *2013 International Green Computing Conference Proceedings*, pages 1–6. IEEE, 2013.

Google. Google's Green PPAs: What, How, and Why. `https://static.googleusercontent.com/media/www.google.com/en//green/pdfs/renewable-energy.pdf`, 2013.

Google. Achieving Our 100% Renewable Energy Purchasing Goal and Going Beyond. `https://static.googleusercontent.com/media/www.google.com/en//green/pdf/achieving-100-renewable-energy-purchasing-goal.pdf`, 2016.

Odd Erik Gundersen and Sigbjørn Kjensmo. State of the art: Reproducibility in artificial intelligence. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Leor Hackel and Gregg Sparkman. Evaluating the climate impact of psychological science: Costs and opportunities. Affective Seminar, 2018. URL `https://osf.io/dg5ap/?show=view`.

Peter Henderson and Emma Brunskill. Distilling information from a flood: A possibility for the use of meta-analysis and systematic review in machine learning research. In *Critiquing and Correcting Trends in Machine Learning Workshop (CRACT) at NeurIPS*, 2018.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Ashley Hill, Antonin Raffin, Maximilian Ernestus, Adam Gleave, Anssi Kanervisto, Rene Traore, Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. Stable baselines. `https://github.com/hill-a/stable-baselines`, 2018.

Matthew H Holden, Nathalie Butt, Alienor Chauvenet, Michaela Plein, Martin Stringer, and Iadine Chadès. Academic conferences urgently need environmental policies. *Nature ecology & evolution*, 2017.

Nicolas Houy. Rational mining limits bitcoin emissions. *Nature Climate Change*, 9(9):655–655, 2019.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*, 2016.

International Energy Agency. *CO2 Emissions from Fuel Combustion*. 2015.

IPCC. *Climate Change 2014: Mitigation of Climate Change: Working Group III Contribution to the IPCC Fifth Assessment Report*. Cambridge University Press, 2015.

IPCC. *Global Warming of 1.5 °C*. 2018.

Yunho Jeon and Junmo Kim. Constructing fast network through deconstruction of convolution. In *Advances in Neural Information Processing Systems*, pages 5951–5961, 2018.

Angela H. Jiang, Daniel L. K. Wong, Giulio Zhou, David G. Andersen, Jeffrey Dean, Gregory R. Ganger, Gauri Joshi, Michael Kaminksy, Michael Kozuch, Zachary C. Lipton, and Padmanabhan Pillai. Accelerating Deep Learning by Focusing on the Biggest Losers. *arXiv e-prints*, art. arXiv:1910.00762, Oct 2019.

Alex K Jones, Liang Liao, William O Collinge, Haifeng Xu, Laura A Schaefer, Amy E Landis, and Melissa M Bilec. Green computing: A life cycle perspective. In *2013 International Green Computing Conference Proceedings*, pages 1–6. IEEE, 2013.

Richard Kavanagh and Karim Djemame. Rapid and accurate energy models through calibration with ipmi and rapl. *Concurrency and Computation: Practice and Experience*, 31(13):e5124, 2019.

Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen, and Zhonghong Ou. RAPL in Action: Experiences in Using RAPL for Power Measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.*, 3(2):9:1–9:26, March 2018. ISSN 2376-3639.

Max J Krause and Thabet Tolaymat. Quantification of energy and carbon costs for mining cryptocurrencies. *Nature Sustainability*, 1(11):711, 2018.

V. Krey, O. Masera, G. Blanford, T. Bruckner, R. Cooke, K. Fisher-Vanden, H. Haberl, E. Hertwich, E. Kriegler, D. Mueller, S. Paltsev, L. Price, S. Schlömer, D. Ürge-Vorsatz, D. van Vuuren, and T. Zwickel. Annex 2 - metrics and methodology. In *Climate Change 2014: Mitigation of Climate Change. IPCC Working Group III Contribution to AR5*. Cambridge University Press, November 2014. URL `http://pure.iiasa.ac.at/id/eprint/11109/`.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

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

Jacob LaRiviere, Gavin Mccormick, and Sho Kawano. How better accounting can more cheaply reduce carbon emissions. *Policy Brief*, 4, 2016.

Yung-Hsiang Lu, Alexander C Berg, and Yiran Chen. Low-power image recognition challenge. *AI Magazine*, 39(2):87–88, 2018.

Jens Malmodin, Pernilla Bergmark, and Dag Lundén. The future carbon footprint of the ict and e&m sectors. *on Information and Communication Technologies*, page 12, 2013.

Eric Masanet, Arman Shehabi, Nuoa Lei, Harald Vranken, Jonathan Koomey, and Jens Malmodin. Implausible projections overestimate near-term bitcoin co2 emissions. *Nature Climate Change*, 9(9):653–654, 2019.

Stephen Merity. Single Headed Attention RNN: Stop Thinking With Your Head. *arXiv preprint arXiv:1911.11423*, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing Atari With Deep Reinforcement Learning. In *NIPS Deep Learning Workshop*. 2013.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

Camilo Mora, Randi L Rollins, Katie Taladay, Michael B Kantar, Mason K Chock, Mio Shimada, and Erik C Franklin. Bitcoin emissions alone could push global warming above 2 °C. *Nature Climate Change*, 8(11):931, 2018.

Richard G Newell and Juha Siikamäki. Nudging energy efficiency behavior: The role of information labels. *Journal of the Association of Environmental and Resource Economists*, 1(4):555–598, 2014.

Kim Khoa Nguyen, Mohamed Cheriet, Mathieu Lemay, Victor Reijs, Andrew Mackarel, and Alin Pastrama. Environmental-aware virtual data center network. *Computer Networks*, 2012.

Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium, 2018. Association for Computational Linguistics.

Daniel Pichert and Konstantinos V. Katsikopoulos. Green defaults: Information presentation and pro-environmental behaviour. *Journal of Environmental Psychology*, 28(1):63 – 73, 2008. ISSN 0272-4944. doi: https://doi.org/10.1016/j.jenvp.2007.09.004. URL `http://www.sciencedirect.com/science/article/pii/S0272494407000758`.

Lucas Venezian Povoa, Cesar Marcondes, and Hermes Senger. Modeling energy consumption based on resource utilization. In *International Conference on Computational Science and Its Applications*, pages 225–240. Springer, 2019.

Zheng Qin, Zhaoning Zhang, Dongsheng Li, Yiming Zhang, and Yuxing Peng. Diagonalwise Refactorization: An Efficient Training Method for Depthwise Convolutions. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

Celine Ramstein, Goran Dominioni, Sanaz Ettehad, Long Lam, Maurice Quant, Jialiang Zhang, Louis Mark, Sam Nierop, Tom Berg, Paige Leuschner, et al. State and trends of carbon pricing 2019, 2019.

Vijay Janapa Reddi, Christine Cheng, David Kanter, Peter Mattson, Guenther Schmuelling, Carole-Jean Wu, Brian Anderson, Maximilien Breughe, Mark Charlebois, William Chou, et al. Mlperf inference benchmark. In *2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)*, pages 446–459. IEEE, 2020.

Nils Reimers and Iryna Gurevych. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *EMNLP*, 2017.

Katharine Ricke, Laurent Drouet, Ken Caldeira, and Massimo Tavoni. Country-level social cost of carbon. *Nature Climate Change*, 2018.

Giampaolo Rodola. Psutil package: a cross-platform library for retrieving information on running processes and system utilization, 2016.

David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. Tackling Climate Change with Machine Learning. *arXiv e-prints*, art. arXiv:1906.05433, Jun 2019.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. Green AI. *arXiv e-prints*, art. arXiv:1907.10597, Jul 2019.

Satyabrata Sen, Neena Imam, and Chung-Hsing Hsu. Quality assessment of gpu power profiling mechanisms. In *2018 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 702–711. IEEE, 2018.

Sam Shead. AI Researchers Left Disappointed As NIPS Sells Out In Under 12 Minutes. *Forbes*, Sep 2018. URL `https://www.forbes.com/sites/samshead/2018/09/05/ai-researchers-left-disappointed-as-nips-sells-out-in-under-12-minutes/#7dda67fc20e9`.

Yoav Shoham, Erik Brynjolfsson, Jack Clark, John Etchemendy, Barbara Grosz, Terah Lyons, James Manyika, Saurabh Mishra, and Juan Carlos Niebles. The ai index 2019 annual report. *AI Index Steering Committee, Human-Centered AI Initiative, Stanford University.*, 2019.

Szymon Sidor and John Schulman. OpenAI Baselines: DQN (Blogpost). 2017.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Frank Soboczenski, Michael D Himes, Molly D O'Beirne, Simone Zorzan, Atilim Gunes Baydin, Adam D Cobb, Yarin Gal, Daniel Angerhausen, Massimo Mascaro, Giada N Arney, et al. Bayesian deep learning for exoplanet atmospheric retrieval. *arXiv preprint arXiv:1811.03390*, 2018.

Susan Solomon, Gian-Kasper Plattner, Reto Knutti, and Pierre Friedlingstein. Irreversible climate change due to carbon dioxide emissions. *Proceedings of the national academy of sciences*, 106(6):1704–1709, 2009.

Shuaiwen Leon Song, Kevin Barker, and Darren Kerbyson. Unified performance and power modeling of scientific workloads. In *Proceedings of the 1st International Workshop on Energy Efficient Supercomputing*, page 4. ACM, 2013.

Diomidis Spinellis and Panos Louridas. The carbon footprint of conference papers. *PloS one*, 8(6):e66508, 2013.

Kenneth O Stanley and Joel Lehman. *Why greatness cannot be planned: The myth of the objective*. Springer, 2015.

Kristin Stechemesser and Edeltraud Guenther. Carbon accounting: a systematic literature review. *Journal of Cleaner Production*, 36:17–38, 2012.

Christian Stoll, Lena Klaaßen, and Ulrich Gallersdörfer. The carbon footprint of bitcoin. *Joule*, 2019.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and Policy Considerations for Deep Learning in NLP. *arXiv preprint arXiv:1906.02243*, 2019.

Vladimir Sukhoy and Alexander Stoytchev. Eliminating the Variability of Cross-Validation Results with LIBLINEAR due to Randomization and Parallelization. 2019.

Shyam Sundar, Ashish Kumar Mishra, and Ram Naresh. Modeling the impact of media awareness programs on mitigation of carbon dioxide emitted from automobiles. *Modeling Earth Systems and Environment*, 4(1):349–357, 2018.

Richard Sutton. The bitter lesson. *Incomplete Ideas (blog), March*, 13, 2019.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, 2015.

Samuel Tang and David Demeritt. Climate change and mandatory carbon reporting: Impacts on business process and performance. *Business Strategy and the Environment*, 27(4): 437–455, 2018.

Richard SJ Tol. The social cost of carbon. *Annu. Rev. Resour. Econ.*, 3(1):419–443, 2011.

Bo Tranberg, Olivier Corradi, Bruno Lajoie, Thomas Gibon, Iain Staffell, and Gorm Bruun Andresen. Real-time carbon accounting method for the european electricity markets. *Energy Strategy Reviews*, 26:100367, 2019.

Barak Turovsky. Ten years of Google Translate. *Google Official Blog*, 2016.

U.S. Environment Protection Agency. Social Cost of Carbon. https://www.epa.gov/sites/production/files/2016-12/documents/social_cost_of_carbon_fact_sheet.pdf, 2013.

Chandramouli Venkatesan. Comparative Carbon Footprint Assessment of the Manufacturing and Use Phases of Two Generations of AMD Accelerated Processing Units, 2015.

David Weisbach and Cass R Sunstein. Climate change and discounting the future: a guide for the perplexed. *Yale L. & Pol'y Rev.*, 27:433, 2008.

Felix Wu, Angela Fan, Alexei Baevski, Yann Dauphin, and Michael Auli. Pay Less Attention with Lightweight and Dynamic Convolutions. In *International Conference on Learning Representations*, 2019.

Michel Zade, Jonas Myklebost, Peter Tzscheutschler, and Ulrich Wagner. Is bitcoin the only problem? a scenario model for the power demand of blockchains. *Frontiers in Energy Research*, 7, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. Hulk: An energy efficiency benchmark platform for responsible natural language processing. *arXiv preprint arXiv:2002.05829*, 2020.

Appendix A. Conference Travel

Prior work has also examined conference travel for various fields as a major source of impact Spinellis and Louridas (2013); Astudillo and AzariJafari (2018); Hackel and Sparkman (2018). For example, Spinellis and Louridas (2013) found that the CO_{2eq}emissions from travel per conference participant was about 801 kg CO_{2eq}, Astudillo and AzariJafari (2018) estimated around 883 kg CO_{2eq} emissions per participant, and Hackel and Sparkman (2018) estimate around 910 kg of CO_{2eq} emissions per participant. Interestingly, these separate papers all align around the same carbon emissions numbers per conference participant. Using this and ML conference participant statistics we can gain some (very) rough insight into the carbon emissions caused by conference travel (not including food purchases, accommodations, and travel within the conference city).

Conference participation has grown particularly popular in ML research, attracting participants from industry and academia. In 2018 the Neural Information Processing Systems (NeurIPS) conference sold out registrations in 12 minutes (Shead, 2018). In 2019, according to the AI Index Report 2019 (Shoham et al., 2019), conferences had the following attendance: CVPR (9,227); IJCAI (3,015); AAAI (3,227); NeurIPS (13,500); IROS (3,509); ICML (6,481); ICLR (2,720); AAMAS (701); ICAPS (283); UAI (334). The larger conferences also showed continued growth: NeurIPS showed a year-over-year growth 41% from 2018 to 2019. Given only these conferences and their attendances in 2019, the lower 801kg CO_{2eq} average emissions estimate per participant (Spinellis and Louridas, 2013), this adds up to roughly 34,440,597 kg CO_{2eq} emitted in 2019 from ML-related conferences (not considering co-location and many other factors).

Appendix B. NeurIPS Sampling on Metric Reporting

We randomly sampled 100 NeurIPS papers from the 2019 proceedings, of these papers we found 1 measured energy in some way, 45 measured runtime in some way, 46 provided the hardware used, 17 provided some measure of computational complexity (e.g., compute-time, FPOs, parameters), and 0 provided carbon metrics. We sampled from the NeurIPS proceedings page: `https://papers.nips.cc/book/advances-in-neural-information-processing-systems-32-2019`. We first automatically check for key words (below) related to energy, compute, and carbon. We then examined the context of the word to classify it as relating to hardware details (e.g., Nvidia Titan X GPU), computational efficiency (e.g., FPOs, MAdds, GPU-hours), runtime (e.g., the experiment ran for 8 hours), energy (e.g., a plot of performance over Joules or Watts), or carbon (e.g., we estimate 10 kg CO_{2eq}were emitted). We also manually validate papers for similar metrics that didn't appear in the keyword search. If a paper

did not contain experiments we removed it and randomly redrew a new paper. In many cases, metrics are only provided for some subset of experiments (or for particular ablation experiments). We nonetheless count these as reporting the metric. Where a neural network diagram or architecture description was provided, we did not consider this to be reporting a compute metric.

compute_terms = ["flop", "fpo", "pflop", "tflops", "tflop", "parameters", "params", "pflops", "flops", "fpos", "gpu-hours", "cpu-hours", "cpu-time", "gpu-time", "multiply-add", "madd"]

hardware_terms = ["nvidia", "intel", "amd", "radeon", "gtx", "titan", "v100", "tpu", "ryzen", "cpu", "gpu"]

time_terms = ["seconds", "second", "hour", "hours", "day", "days", "time", "experiment length", "run-time", "runtime"]

energy_terms = ["watt", "kWh", "joule", "joules", "wh", "kwhs", "watts", "rapl", "energy", "power"]

carbon_terms = ["co2", "carbon", "emissions"]

Appendix C. Carbon Discussion

But cloud providers claim 100% carbon neutrality in my region, why do I need to shift my resources?

While we estimate energy mixes based on regional grids, cloud providers sometimes aim for carbon *neutrality* through a mixture of mechanisms which may change the energy mix being provided to a data center in an otherwise carbon intensive energy grid or otherwise offset unclean energy usage. Data centers draw energy from the local energy grids and as a result the mix of energy they consume largely depends on the composition of the power running in the grids. If the local energy grids are powered by a mix of fuel and renewable energy, a data center will inevitably consume fuel energy as well.

Due to the fact that the consumers do not know the origin of the physical electricity from the utility grid, it is difficult to assign ownership of the renewable energy consumption. The Environmental Protection Agency (EPA) uses renewable energy certificates (RECs) to track the generation and consumption of renewable energy: one REC is issued when one megawatt-hour (MWh) of electricity is generated from a renewable source and delivered to the energy grid.[36] Consumers can then purchase RECs from a renewable energy provider and apply them to their electricity usage. This means consumers can claim they run on renewable energy by purchasing RECs from providers that doesn't actually power the energy grids that they draw electricity from. Although this means that the consumers' realtime carbon footprints will still be decided by the composition of renewable and fuel energy in their local energy grids, more renewable energy can flow onto the grid by purchasing the RECs and future development of renewable sources is supported. Google, to offset its carbon emissions, uses RECs and power purchase agreements (PPAs) with renewable energy providers to ensure that more renewable energy powers the same electricity grids that its data centers are in.[37]

36. `https://www.epa.gov/greenpower/renewable-energy-certificates-recs`
37. We note that this process is likely similar for most cloud providers, but Google is the most open with their methodology, so we are able to gain more insight from the materials they publish. Information described here is mainly put together from Google (2016) and Google (2013).

Google then sells the renewable energy as it becomes available back to the electricity grids and strips away the RECs. Over one year, Google applies equal amounts of RECs to its data centers' total energy consumption. This method helps green energy provider development by creating a long term demand. However, PPAs provide RECs for *future renewables*, not only current energy on the grid which may remain unchanged. As it states: "While the renewable facility output is not being used directly to power a Google data center, the PPA arrangement assures that additional renewable generation sufficient to power the data center came on line in the area."[38]

We can see that even if a cloud provider's data centers are carbon neutral, the actual CO_{2eq}emissions can vary largely and depends on the region and even time of the day (solar energy cannot be generated at night). Since carbon emissions have some long-term or irreversible impacts (Solomon et al., 2009), avoiding carbon emissions now can help down the line—a reason why discount rates are used in calculating impacts (Weisbach and Sunstein, 2008). We suggest that cloud providers release tools for understanding the carbon intensity for each data center region regardless of offset purchasing. While the purchases of PPAs and RECs are valuable for driving towards renewable energy in otherwise dirty regions, for machine learning model training, where the resources can be moved, we believe shifting resources to low intensity regions is more beneficial to long term carbon impacts. Other cloud-based jobs where latency requirements prevent shifting resources will remain to drive PPA/REC purchasing, and consequently renewable energy demand.

Appendix D. ImageNet Experiments

We load pre-trained models available through PyTorch Hub (see `https://pytorch.org/hub`)—namely AlexNet (Krizhevsky et al., 2012), DenseNet (Huang et al., 2017), GoogLeNet (Szegedy et al., 2015), HardNet (Chao et al., 2019), MobileNetv2 (Sandler et al., 2018), ShuffleNet (Zhang et al., 2018), SqueezeNet (Iandola et al., 2016), VGG (Simonyan and Zisserman, 2014), and Wide ResNets (Zagoruyko and Komodakis, 2016). We run 50,000 rounds of inference on a single image through pre-trained image classification models and run similar analysis to Canziani et al. (2016). We repeat experiments on 4 random seeds.

We count flops and parameters using the thop package (for package version numbers see automated logs in the online appendix linked above): `https://github.com/Lyken17/pytorch-OpCounter`

Code for running the experiment is available at:
`https://github.com/Breakend/ClimateChangeFromMachineLearningResearch/blob/master/paper_specific/run_inference.py`

An online appendix showing all per-experiment details can be seen here:
`https://breakend.github.io/ClimateChangeFromMachineLearningResearch/measuring_and_mitigating_energy_and_carbon_footprints_in_machine_learning/`

The plot of FPOs versus runtime can be seen in Figure 8 and plots against number of parameters can be seen in Figure 9. Number of parameters similarly have no strong correlation with energy consumption ($R^2 = 0.002$, Pearson -0.048), nor time ($R^2 = 0.14$,

38. `https://static.googleusercontent.com/media/www.google.com/en/us/green/pdfs/renewable-energy.pdf`

Pearson −.373). We note that our runtime results likely differ from Canziani et al. (2016) due to the architectural differences in the model sets we use.

For parameter plots, see Figure 9, for extended time and energy Figures, see Figure 8.

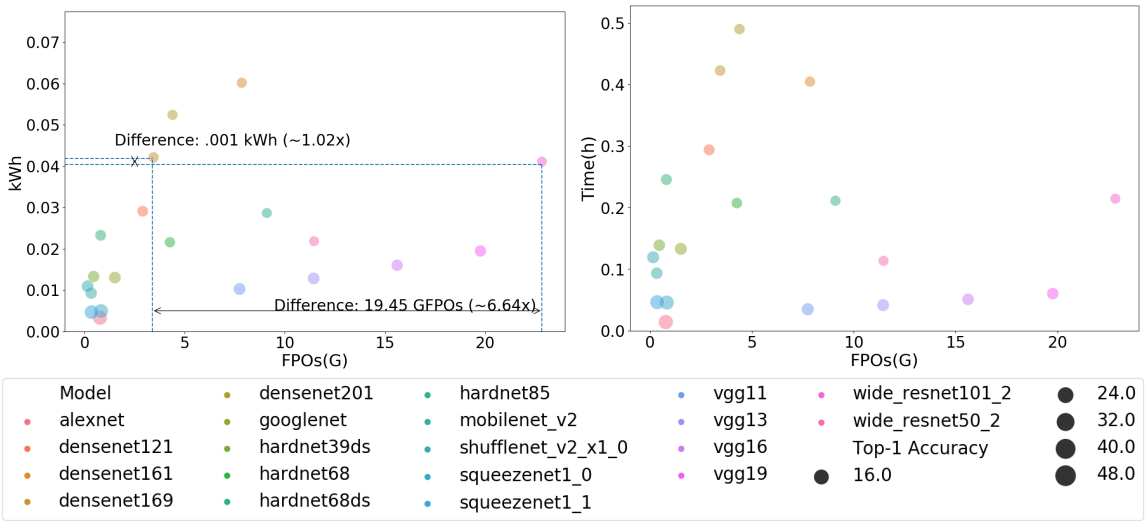

Figure 8: We seek to investigate the connection between FPOs, energy usage, and experiment time, similarly to Canziani et al. (2016). We run 50,000 rounds of inference on a single image through pre-trained image classification models available through PyTorch Hub (see `https://pytorch.org/hub`)—namely (Krizhevsky et al., 2012; Huang et al., 2017; Szegedy et al., 2015; Chao et al., 2019; Sandler et al., 2018; Zhang et al., 2018; Iandola et al., 2016; Simonyan and Zisserman, 2014; Zagoruyko and Komodakis, 2016). We record experiment time and the kWh of energy used to run the experiments and repeat experiments 4 times, averaging results. We find that FPOs are not strongly correlated with energy consumption ($R^2 = 0.083$, Pearson 0.289) nor with time ($R^2 = 0.005$, Pearson −0.074). Number of parameters (plotted in Appendix) similarly have no strong correlation with energy consumption ($R^2 = 0.002$, Pearson −0.048), nor time ($R^2 = 0.14$, Pearson −.373). We note, however, that *within an architecture* correlations are much stronger. For example, only considering different versions of VGG, FPOs are strongly correlated with energy ($R^2 = .999$, Pearson 1.0) and time ($R^2 = .998$, Pearson .999). See Appendix for experiment details, code, and data links. Our runtime results likely differ from Canziani et al. (2016) due to the architectural differences in the model sets we use.

Appendix E. Estimation Methods

We use our PPO Pong experiment (see Appendix F for more details) as the experiment under comparison. For carbon emission estimates, we use three estimation methods: realtime emissions data for California (collected by our framework from `caiso.org`) times the power

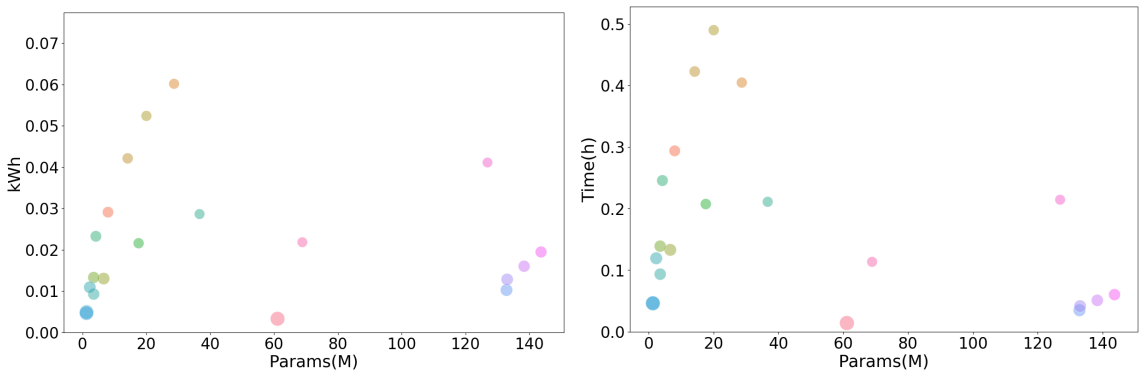

Figure 9: The same experiments as in Figure 3, plotting parameters as the varying factor instead. See Figure 3 for correlation values.

usage at that time integrated over the length of the experiment; multiplying total energy usage recorded by our method by the California average carbon intensity; multiplying total energy usage recorded by our method by the EPA US average carbon intensity (Strubell et al., 2019). For energy estimates, we use: (1) the experiment time multiplied by the number of GPUs, a utilization factor of 1/3 or 1, and the Thermal Design Power (TDP)—which can be thought of as the maximum Watt draw—of the GPU (Amodei and Hernandez, 2018); (2) the measured GPU-hrs of our tool multiplied by the TDP; a rough calculation of PFLOPs-hr (following the methodology of (Amodei and Hernandez, 2018) by the PFLOPs/TDP of the GPU; (3) our tool's accounting method which tracks energy from GPU readings, accounts for CPU time/energy, and measures utilization in realtime.

Appendix F. Reinforcement Learning

We investigate the energy efficiency of four baseline RL algorithms: PPO (Hill et al., 2018; Schulman et al., 2017), A2C (Hill et al., 2018; Mnih et al., 2016), A2C with VTraces (Espeholt et al., 2018; Dalton et al., 2019), and DQN (Hill et al., 2018; Mnih et al., 2016). We evaluate on PongNoFrameskip-v4 (left) and BreakoutNoFrameskip-v4 (right), two common evaluation environments included in OpenAI Gym (Bellemare et al., 2013; Brockman et al., 2016; Mnih et al., 2013).

We train for only 5M timesteps, less than prior work, to encourage energy efficiency (Mnih et al., 2016, 2013). We use default settings from code provided in stable-baselines (Hill et al., 2018) and cule (Dalton et al., 2019), we only modify evaluation code slightly. Modifications can be found here:

- `https://github.com/Breakend/rl-baselines-zoo-1` (for stable-baselines modifications)
- `https://github.com/Breakend/cule` (for cule modifications)

Since we compare both on-policy and off-policy methods, for fairness all evaluation is based on 25 separate rollouts completed every 250k timesteps. This is to ensure parity

across algorithms. We execute these in parallel together as seen in the cule code: `https://github.com/Breakend/cule/blob/master/examples/a2c/test.py`.

While average return across all evaluation episodes (e.g., averaging together the step at 250k timesteps and every evaluation step until 5M timesteps) can be seen in the main text, the asymptotic return (for the final round of evaluation episodes) can be seen Figure 10. Plots comparing experiment runtime to asymptotic and average returns (respectively) can be seen in Figure 11 and Figure 12.

Our online leaderboard can be seen at: `https://breakend.github.io/RL-Energy-Leaderboard/reinforcement_learning_energy_leaderboard/index.html`

We note that while DQN underperforms as compared to PPO here, better hyperparameters may be found such that DQN is the more energy efficient algorithm. Moreover, we only use the 5M samples regime, whereas prior work has used 10M or more samples for training, so DQN results seen here would correspond to earlier points in training in other papers.

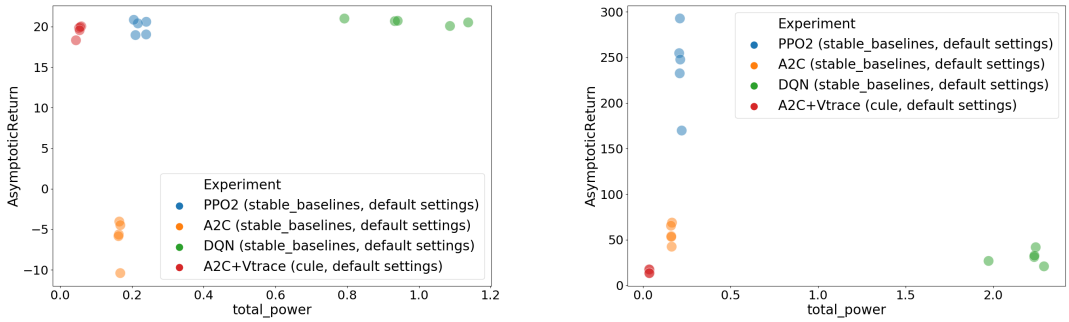

Figure 10: Pong (left) and Breakout (right) asymptotic return.

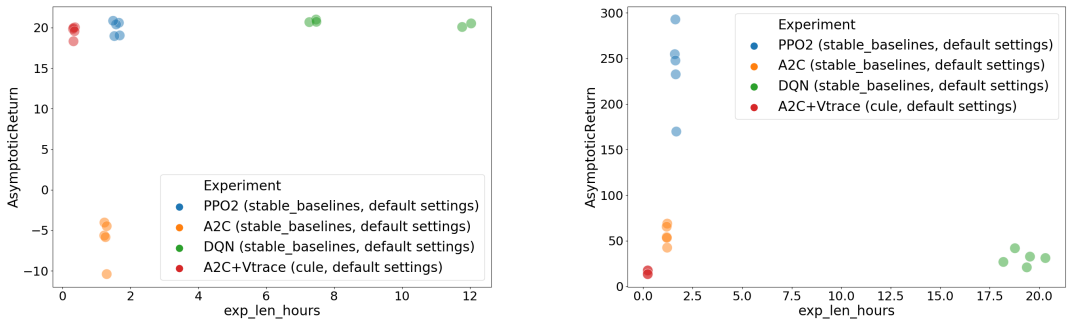

Figure 11: Pong (left) and Breakout (right) as a function of experiment length and asymptotic return.

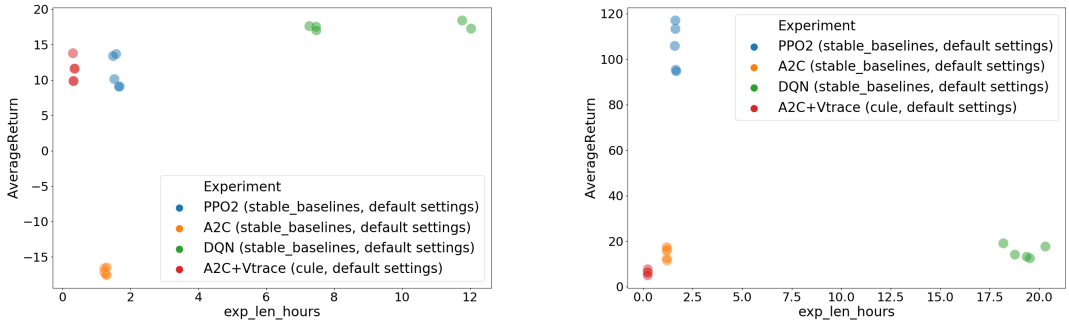

Figure 12: Pong (left) and Breakout (right) as a function of experiment length and average return.

Appendix G. Possible Sources of Error, Limitations, and Overheads

In Sections 5.2 and 5.1, we compared different methods for estimating energy and carbon emissions including extrapolating from FPOs. However, we note that our own framework is not perfect. For transparency, we highlight several such sources here, but we note that utilizing more information—as we do here—is by definition superior to approximations which rely on less accurate assumptions (see Section 5.2).

First, we rely on downstream hardware APIs which themselves have errors. Several works have sought to evaluate the accuracy of RAPL—see for example Desrochers et al. (2016) and Kavanagh and Djemame (2019)—and Nvidia's power profiling tool—see for example, Sen et al. (2018) and Arafa et al. (2020). Errors highly depend on the specific chipset and even the workload, so we refer the reader to these other works for techniques in assessing exact errors. Nvidia's documentation, however, states that the power reading "is accurate to within +/- 5 watts."[39]

Second, we rely on a polling mechanism due to the constraints of these downstream APIs (for GPUs typically only power is provided, rather than an energy counter). In particularly short jobs or highly erratic workloads, the tool may poll at a time that is not representative of the full workload, estimating energy usage from an outlier power sample. Our assumption is that workloads are fairly consistent and long enough that such variability will average out. In the event that comparisons of energy readings across models are needed, we encourage users to report standard errors across several runs (with n appropriate for the experiment setting). Furthermore, because we record many auxiliary data sources (such as CPU frequency), more accurate estimates can further be conducted via mixed effects models to control for sources of variation and noise in energy readings. For an example of how such an analysis would work, see for example Boquet et al. (2019), which compare machine learning algorithms controlling for hyperparameter choice and randomness.

Third, for cloud regions, we do not have access to the exact carbon intensities or PUEs. For example, if a cloud provider has a direct connection to a clean energy power plant for 100% of its energy, we have no way of accessing this information and using it in our tool.

39. https://developer.download.nvidia.com/compute/DCGM/docs/nvidia-smi-367.38.pdf

We encourage companies to report this information per cloud region so that this may be more accurate. In the case of indirect carbon offsetting, we do not consider this to be an inaccuracy—see discussion in Appendix C. Moreover, we rely on IP address information and hand-gathered energy grid information to estimate the energy grid. Either of these may incur errors. Since we report this information and allow users to override grid regions in calculations, these may be corrected by users. We also may not be able to access particular drivers needed on every cloud instance. As such, support may depend on the cloud machine image being used and the drivers available on that image. Generally, if Intel's RAPL is available or PowerGadget can be installed—and nvidia-smi is available—then the system should be compatible.

Regarding overheads to adding a separate process gathering these metrics, the cost should be generally fairly low. There are some startup and shutdown costs associated with adding the tool, so for short-running scripts the absolute percentage of overhead may be he higher. Additionally, if computational capacity of a chipset is maximally used due to the main process, there may be some added cost for thread switching to gather metrics. However, assuming that a core is preserved for the impact tracker there should be minimal overhead. Note, for the sake of reproducibility we also record disk read/write speeds, but this can be turned off if the disk is particularly slow or there is too much disk I/O for the user's liking. While workload overhead can vary depending on the machine and workload, we found that in a small experiment of 200 epochs of regression for a one hidden layer neural network, runtime overheads were less than 1%. For 500 epochs, the overhead was around .5% (indicating that startup/shutdown are the most intensive). This experiment was run on a CPU-only Mac OS machine with a 2.7 GHz Quad-Core Intel Core i7 and 16 GB 2133 MHz LPDDR3.

Supporting every driver and hardware combination is difficult. We note that most of the aforementioned metrics are only supported on Linux systems and we were only able to test hardware combinations available to us. Mac OS support is limited to machines that have Intel's Power Gadget[40] installed and to CPU-only recordings. We hope that future users will help identify missing capabilities and expand the framework for new use-cases and machines. We also note that the tool is limited by driver support in cases that we cannot work around (see Section 7.7).

Finally, we note that we only record CPU, GPU, and DRAM power draw. We do not record disk I/O energy usage, power conversion and voltage regulator overhead. As such, we can expect there to be missing components that contribute to energy that we do not record here. However, we expect that the PUE re-scaling will correct for some of these missing components to some extent.

Appendix H. Comparing Models

We note that it may be tempting to use carbon emissions as a comparative tool: model A is less carbon intensive that model B. However, unless the carbon intensity used for either model is held constant, this comparison cannot be done. In particular, our tool should not be used to compare carbon emissions between models without overriding the carbon intensity used as we sometimes use real-time values. If two models are compared, as in Section 6.1.1, multiple runs on comparable machines should be used. In the event that a robust conclusion

40. https://software.intel.com/content/www/us/en/develop/articles/intel-power-gadget.html

is to be made (e.g., Algorithm A is more energy efficient than Algorithm B), additional metrics regarding workload that we record can be utilized to run a mixed-effects regression analysis. Such an analysis would ensure that there aren't confounding factors jeopardizing the conclusion.