



HAL
open science

Can Federated Learning Save the Planet?

Xinchi Qiu, Titouan Parcollet, Daniel J Beutel, Taner Topal, Akhil Mathur,
Nicholas D Lane

► **To cite this version:**

Xinchi Qiu, Titouan Parcollet, Daniel J Beutel, Taner Topal, Akhil Mathur, et al.. Can Federated Learning Save the Planet?. NeurIPS - Tackling Climate Change with Machine Learning, Dec 2020, Vancouver, Canada. hal-02967192v2

HAL Id: hal-02967192

<https://hal.science/hal-02967192v2>

Submitted on 10 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Can Federated Learning Save The Planet ?

Xinchi Qiu¹, Titouan Parcollet^{2,1}, Daniel J. Beutel⁵, Taner Topal⁵,
Akhil Mathur³, Nicholas D. Lane^{1,4}

¹Department of Computer Science and Technology, University of Cambridge, United-Kingdom

²Avignon Université, France

³University College London, United Kingdom

⁴Samsung AI, United-Kingdom

⁵Adap, Germany

Abstract

Despite impressive results, deep learning-based technologies also raise severe privacy and environmental concerns induced by the training procedure often conducted in data centers. In response, alternatives to centralized training such as Federated Learning (FL) have emerged. Perhaps unexpectedly, FL in particular is starting to be deployed at a global scale by companies that must adhere to new legal demands and policies originating from governments and the civil society for privacy protection. However, the potential environmental impact related to FL remains unclear and unexplored. This paper offers the first-ever systematic study of the carbon footprint of FL. First, we propose a rigorous model to quantify the carbon footprint, hence facilitating the investigation of the relationship between FL design and carbon emissions. Then, we compare the carbon footprint of FL to traditional centralized learning. Finally, we highlight and connect the reported results to the future challenges and trends in FL to reduce its environmental impact, including algorithms efficiency, hardware capabilities, and stronger industry transparency.

1 Introduction

Atmospheric concentrations of carbon dioxide, methane, and nitrous oxide are at unprecedented levels not seen in the last 800,000 years (1), and are extremely likely to have been the dominant cause of the observed global warming since the mid-20th century (2; 3). Unfortunately, deep learning (DL) algorithms keep growing in complexity, and numerous “state-of-the-art” models continue to emerge, each requiring a substantial amount of computational energy, resulting in clear environmental costs (4). However, such a trend is not going to end soon as (5) shows that the amount of compute used by machine learning training has grown by more than 300,000× from 2012 to 2018. This is an observation that forces us to seriously consider the carbon footprint of deep learning methods.

The data centers that enable DL research and commercial operations are not often accompanied by visual signs of pollution, but they are still responsible for significant carbon footprint. Each year they use 200 TWh energy, which accounts for 0.3% of global CO₂ emissions. In comparison, the entire information and communications technology ecosystem accounts for only 2% (6).

Fortunately, alternatives to data center based DL (and other forms of machine learning) are emerging. The most prominent of these to date is *Federated Learning (FL)* proposed by (7). Under FL, the training of models primarily occurs across a large number of typically user owned and controlled personal devices, such as smartphones. Devices collaboratively learn a shared prediction model but do so without uploading to a data center any of the locally stored raw sensitive data. While FL is still a maturing technology, it is already being used by millions of users on a daily basis; for example,

Google uses FL to train models for: predictive keyboard, device setting recommendation, and hot keyword personalization on phones (8).

Whilst the carbon footprint for centralized learning has been studied in many previous works (9; 10; 11; 12), the energy consumption and carbon footprint related to FL remain unexplored. To this extent, this paper proposes to overcome this issue by giving a first look into the carbon analysis of FL. SOTA results in deep learning are usually determined by standard accuracy metrics, such as the accuracy of a given model, while energy efficiency or privacy concerns are often overlooked. Whilst accuracy remains crucial, we hope to encourage researchers to also focus on other metrics that are in line with the increasing interest of the civil society for global warming. By quantifying carbon emissions for FL, we encourage the integration of the released CO₂ as a crucial metric to the FL deployment.

In this paper, we firstly provide a *first-of-its-kind* quantitative CO₂ emissions estimation method for FL (Section 2). Then a carbon sensitivity analysis is conducted with this method on real FL hardware with the CIFAR10 dataset (Section 3). Furthermore, we provide a comprehensive analysis and discussion of the results to highlight the challenges and future research directions to develop an environmentally-friendly federated learning.

2 Quantifying CO₂ emissions

Two major steps can be followed to quantify the environmental cost of training deep learning models either in data centers or on edge. First, we perform an analysis of the energy required by the method (Section 2.1), mostly accounting for the total amount of energy consumed by the hardware. Then, the latter amount is converted to CO₂ emissions based on geographical locations (Section 2.2).

2.1 Total Energy Consumption

First, we need to consider the energy consumption coming from GPU and CPU, which can be measured by sampling GPU and CPU power consumption at training time (13). Alternatively, we can use the official hardware power specification or TDP, assuming a full GPU utilization. Such a use case is not realistic as a GPU is rarely used at 100% of its capacity. In the context of FL, not all clients are equipped with a GPU, and this part can thus be removed from the equation if necessary. To this extent, we propose to consider $e_{clients}$ as the power of a single client combining both GPU and CPU measurements. Then, we can connect these measurements to the total training time.

However, estimating the wall clock time of FL can be challenging. Unlike centralized distributed training, FL runs following communication rounds. During each communication round, certain devices (or clients) are chosen for training. In addition, FL might suffer from system heterogeneity as different edge devices might not offer the same computational power (14). To simplify this highly scenario dependent assumption, we propose to fix the time needed for each round, corresponding to a common FL setup (14). As a matter of fact, such a distribution of clients is extremely difficult to estimate as sales figures for these devices are not publicly released by the industry. Then, the total time needed to train the model also depends on the communication efficiency between the clients and the server. It is worth mentioning that such communications also have an impact on the final carbon footprint. Therefore, let s be the size of model parameters in GB and $5s$ be the energy needed in KWh to transfer the parameters from the clients to the server (15). The latter estimation is highly dependent on the infrastructure efficiency and is only used as an indicator. Finally, the total energy consumed for n clients chosen in each communication round with wall clock time t per round is:

$$n(te_{clients} + 5s) \tag{1}$$

It is important to note that other hardware components may also be responsible for energy consumption, such as RAM or HDD. According to (16), one may expect a variation of around 10% while considering these parameters. However, they are also highly dependent on the infrastructure considered and the device distribution that is unfortunately unavailable.

The particular case of cooling in centralized training. Cooling in data centers accounts for up to 40% of the total energy consumed (17). While this parameter does not exist for FL as the heat is distributed across the set of clients, it is crucial to consider it when estimating the cost of centralized training. Such estimation is highly dependent on the data center efficiency, so we propose to use the Power Usage Effectiveness (PUE) ratio. According to the *2019 Data Centre Industry Survey Results* (18), the world average PUE for the year 2019 is 1.67. As expected, observed PUE strongly vary depending on the considered company. For instance, *Google* declares a PUE ratio of 1.06 (19)

compared to 1.2 and 1.125 for *Amazon* (20) and *Microsoft* (21) respectively. Therefore, Eq. 1 is extended to centralized training as:

$$PUE(te_{clients}), \quad (2)$$

with $n = 1$ and $s = 0$ in the context of centralized training, and t stands for the total training time. In addition, the cost of transferring the model parameters from the RAM to the VRAM is negligible.

2.2 Converting to CO₂ emissions

We assume that all data centers and edge devices are connected to their local grid directly linked to their physical location, and we use the electricity-specific CO₂ emission factors obtained from (22) for estimation. The estimation methodology provided takes into accounts both transmission and distribution emission factors (*i.e.* energy lost when transmitting and distributing electricity) and the efficiency of heat plants. As expected, countries relying on carbon-efficient productions are able to lower their corresponding emission factor (*e.g.* France, Canada). Therefore, the total amount of CO₂ emitted in **kg** for FL and centralized training are obtained from Eq. 1 and Eq. 2 as:

$$rc_{rate}n(t * e_{clients} + 5s) \quad \text{and} \quad c_{rate}PUE(te_{clients}), \quad (3)$$

with c_{rate} the emission factor and r the total number of training rounds needed during the FL procedure. Carbon emissions may be compensated by carbon offsetting or with the purchases of Renewable Energy Credits (RECs). Even though a lot of companies are devoting to the carbon offsetting scheme, this approach still contributes to a net increase in the absolute rate of global emission growth in the atmosphere (23). Therefore, following preliminary works on data-centres CO₂ emission estimations (10; 11), we ignore this practice to only consider the real amount of CO₂ emitted during the training of the DL models.

3 Experiments on FL carbon footprint

We provide an estimation of the carbon footprint of a realistic FL setup for the CIFAR10 classification task, and provide analysis of these results. The entire FL pipeline is implemented within the Flower toolkit¹ (24), with FedAVG (25) and the PyTorch definition of ResNet-18. The exact hardware configuration for our experiments can be found in (Appendix A.1). Also, more estimations are given in Appendix with the FashionMNIST (Appendix A.3) and ImageNet (Appendix A.4) dataset.

We conduct our estimations based on the image classification tasks of CIFAR10. As we would like to only estimate the carbon footprint, we are not interested in achieving the best performance possible. The estimations are computed once 60% testing accuracy is reached. However, more details on the performances and training time are given in Appendix A.2. Both FL and centralized training rely on an identically implemented ResNet-18 architecture with plain SGD and momentum to alleviate any variations. Also, the experiment are based on the standard training and test sets (26).

As CIFAR10 does not offer any natural partitioning, we propose to simulate both IID and non-IID scenarios. In the former case, all the clients have the equal number of samples evenly distributed across all the classes (*e.g.* with 10 total clients, each of them has 500 samples per class). For the non-IID setup, we first distribute evenly 50% of the data across the clients, then we only distribute samples from a subset of the classes (*e.g.* 1 – 5 and 6 – 10) to half of the clients, while the remaining half will get samples from the other subset. We also propose to vary the number of local epochs done on each client to better highlight the contribution of the local computations to the total emissions. The total number of clients is fixed to 10, while 5 of them are randomly selected in each FL round.

Before discussing the estimated emissions, it is worth noting that solely 2 epochs were required for centralized training to reach 60% of testing accuracy with CIFAR10 corresponding to 48 and 84 seconds of training time for Tesla V100 and K80 respectively. Conversely, the fastest FL setup (*i.e.* IID and 1 epoch per round) took 822 seconds to achieve the same level of performance with 16 FL rounds. As expected, and due to the much lower compute capabilities of FL devices, the training time is much longer than centralized training. More details on accuracies and training time can be found in Appendix A.2.

¹Code is not anonymized, but available.

Country/CO ₂ (g) CIFAR10	V100	K80	V100	K80	FL (IID)		FL (non-IID)	
	<i>PUE</i> = 1.67		<i>PUE</i> = 1.11		1 epoch	5 epochs	1 epoch	5 epochs
USA	3.1	6.5	2.1	4.3	6.3	17.5	29.3	21.5
China	5.5	11.5	3.7	7.7	11.1	31.3	52.2	38.3
France	0.4	0.9	0.3	0.6	0.9	2.5	4.2	3.1

Table 1: CO₂ emissions (expressed in grams, *i.e.* **lower is better**) for centralized training and FL on CIFAR10. The number of epoch reported on the FL column relates to the number of local epoch done per client. “IID” and “non-IID” terms are employed to distinguish between clients that have an evenly distributed set of samples containing all the classes (IID) and clients that have more samples of certain classes (non-IID).

CO₂ emissions reported on Table 1 are interesting in many aspects. First, it is clear that FL is more polluting than V100-equipped centralized training (*e.g.* up to three times more with a *PUE* = 1.11, and 2.5 with a *PUE* = 1.67) on the CIFAR benchmark. Furthermore, this assumption does not hold anymore once we consider less efficient GPUs such as Tesla K80. Indeed, in this scenario and with a *PUE* = 1.67, FL becomes equivalently polluting or slightly less than centralized training. Second, if we put in perspective both the training time and emissions reported, it is worth underlining that despite being much slower to train, FL remains highly competitive in term of CO₂ emissions. Indeed, the latter finding demonstrates the potential of FL to become greener than centralized learning by decreasing the training time.

Therefore, based on our FL CO₂ footprint estimation method, we also propose a formalization that integrates carbon emission into the common neural networks optimization process to lower the final *Carbon Cost* as can be found in Appendix A.5.

3.1 Discussion

CO₂ emissions induced by FL highly depends on different factors. First, we show that a basic FL setup relying on FedAVG clearly emits more carbon compared to modern GPUs and centralized training. The latter finding does not hold with older and cheaper GPUs such as Tesla K80. Hence hardware efficiency is one of the most important factors when estimating the total carbon footprint. More precisely, we considered NVIDIA Tegra X2 as our client hardware. While such chips could realistically be embedded in numerous devices, including smartphones, tablets, game consoles, and others, they are certainly not an exact estimate of what the industry uses for FL. Therefore, and to facilitate environmental impact estimations of large scale FL deployment, the industry must increase its transparency with respect to their devices distribution over the market.

Of course, both FL and centralized learning benefit from more efficient hardware. However, and as explained in our estimation methodology, FL will always have an advantage due to the cooling needs of data centers. In fact, even though GPUs or even TPUs are getting more efficient in terms of computational power delivered by the amount of energy consumed, the need for a strong and energy-consuming cooling remains – thus the FL advantage only grows. Unlike centralized learning, FL always benefits from a net decrease in CO₂ emission each time the hardware is improved.

Furthermore, our results show that realistic training conditions for FL (*i.e.* non-IID data) are largely responsible for longer training times and high level of CO₂ emissions. While it is well known that the simpler form of FL (*e.g.* FedAVG) struggle with non-IID partitioned data in terms of accuracy (14; 27), we presented another motivation to pursue the research trying to address this issue: this could lead to a significant decrease in carbon emissions. Future algorithms on non-IID data could easily replicate our estimation methodology to highlight their efficacy.

Finally, FL depends on hyper-parameters, such as the number of clients selected, the number of local epochs, and others. These variables, when tuned, are usually decided to suit hardware resources available or by grid search optimization. These parameters have potential to be included in the optimization process at training time to lower the overall carbon emission. Novel algorithms should carefully be designed to minimize the carbon emission by jointly maximizing the accuracy and minimizing the released CO₂.

4 Conclusion

Climate change is real, and DL plays an increasing role in this tragedy. Fortunately, a number of recent studies have begun to detail the environmental costs of their novel deep learning methods, sometimes even integrating CO₂ emissions as an objective to be minimized. Following this important trend, this paper takes a first look into the carbon footprint of an increasingly deployed training strategy known as federated learning. In particular, this work introduces a generalized methodology to systematically compute the carbon footprint of any federated learning setups. Finally, novel research directions are highlighted to make FL a greener alternative compared to centralized training.

References

- [1] IPCC, “Climate change 2014 synthesis report,” 2014.
- [2] R. K. Pachauri, L. Gomez-Echeverri, and K. Riahi, “Synthesis report: summary for policy makers,” 2014.
- [3] T. J. Crowley, “Causes of climate change over the past 1000 years,” *Science*, vol. 289, no. 5477, pp. 270–277, 2000.
- [4] K. Hao, “Training a single ai model can emit as much carbon as five cars in their lifetimes,” *MIT Technology Review*, 2019.
- [5] D. Amodei and D. Hernandez, “Ai and compute,” *Heruntergeladen von <https://blog.openai.com/aiand-compute>*, 2018.
- [6] Nature, “How to stop data centres from gobbling up the world’s electricity,” <https://www.nature.com/articles/d41586-018-06610-y>, Sep 2018.
- [7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-Efficient Learning of Deep Networks from Decentralized Data,” ser. *Proceedings of Machine Learning Research*, A. Singh and J. Zhu, Eds., vol. 54. Fort Lauderdale, FL, USA: PMLR, 20–22 Apr 2017, pp. 1273–1282. [Online]. Available: <http://proceedings.mlr.press/v54/mcmahan17a.html>
- [8] B. McMahan and D. Ramage, “Federated learning: Collaborative machine learning without centralized training data,” *Google Research Blog*, vol. 3, 2017.
- [9] L. F. W. Anthony, B. Kanding, and R. Selvan, “Carbontracker: Tracking and predicting the carbon footprint of training deep learning models,” *arXiv preprint arXiv:2007.03051*, 2020.
- [10] A. Lacoste, A. Luccioni, V. Schmidt, and T. Dandres, “Quantifying the carbon emissions of machine learning,” *arXiv preprint arXiv:1910.09700*, 2019.
- [11] P. Henderson, J. Hu, J. Romoff, E. Brunskill, D. Jurafsky, and J. Pineau, “Towards the systematic reporting of the energy and carbon footprints of machine learning,” *arXiv preprint arXiv:2002.05651*, 2020.
- [12] A. Uchchukwu, K. Li, Y. Shen *et al.*, “Energy consumption in cloud computing data centers,” *International Journal of Cloud Computing and Services Science (IJ-CLOSER)*, vol. 3, no. 3, pp. 145–162, 2014.
- [13] E. Strubell, A. Ganesh, and A. McCallum, “Energy and policy considerations for deep learning in nlp,” *arXiv preprint arXiv:1906.02243*, 2019.
- [14] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” *arXiv preprint arXiv:1812.06127*, 2018.
- [15] D. Costenaro and A. Duer, “The megawatts behind your megabytes: going from data-center to desktop,” *Proceedings of the 2012 ACEEE Summer Study on Energy Efficiency in Buildings, ACEEE, Washington*, pp. 13–65, 2012.
- [16] M. Hodak, M. Gorkovenko, and A. Dholakia, “Towards power efficiency in deep learning on data center hardware,” in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1814–1820.

- [17] A. Capozzoli and G. Primiceri, "Cooling systems in data centers: state of art and emerging technologies," *Energy Procedia*, vol. 83, pp. 484–493, 2015.
- [18] UptimeInstitute, "2019 data center industry survey results," <https://uptimeinstitute.com/2019-data-center-industry-survey-results>, 2019.
- [19] Google, "Efficiency-data centres," <https://www.google.co.uk/about/datacenters/efficiency/>, 2020.
- [20] AWS, "Aws and sustainability," <https://aws.amazon.com/about-aws/sustainability/>, 2020.
- [21] Microsoft, "Datacenter fact sheet - microsoft download center," http://download.microsoft.com/download/8/2/9/8297f7c7-ae81-4e99-b1db-d65a01f7a8ef/microsoft_cloud_infrastructure_datacenter_and_network_fact_sheet.pdf, 2015.
- [22] M. Hodak, M. Gorkovenko, and A. Dholakia, "Towards power efficiency in deep learning on data center hardware," in *2019 IEEE International Conference on Big Data (Big Data)*. IEEE, 2019, pp. 1814–1820.
- [23] K. Anderson, "The inconvenient truth of carbon offsets," *Nature*, vol. 484, no. 7392, pp. 7–7, 2012.
- [24] D. J. Beutel, T. Topal, A. Mathur, X. Qiu, T. Parcollet, and N. D. Lane, "Flower: A friendly federated learning research framework," 2020.
- [25] X. Li, K. Huang, W. Yang, S. Wang, and Z. Zhang, "On the convergence of fedavg on non-iid data," *arXiv preprint arXiv:1907.02189*, 2019.
- [26] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.
- [27] J. Qian, X. Fafoutis, and L. K. Hansen, "Towards federated learning: Robustness analytics to data heterogeneity," *arXiv preprint arXiv:2002.05038*, 2020.
- [28] E. Bisong, "Google colaboy," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Springer, 2019, pp. 59–64.
- [29] K. Lepak, G. Talbot, S. White, N. Beck, S. Naffziger *et al.*, "The next generation amd enterprise server product architecture," *IEEE hot chips*, vol. 29, 2017.
- [30] R. Smith, "Nvidia announces jetson tx2: Parker comes to nvidia's embedded system kit," mar 2017. [Online]. Available: <https://www.anandtech.com/show/11185/nvidia-announces-jetson-tx2-parker>
- [31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

A Appendix

A.1 Hardware configuration for the experiments

In addition to the carbon model (Section 2), our results are influenced by the configuration of the hardware and systems of data center and federated learning respectively.

Centralized training. NVIDIA Tesla V100 and K80 are used as reference graphics card in our experiments. The former GPU proposes a competitive performance / TDP ratio, while the latter GPU is often deployed in collaborative environments such as *Google Colaboratory* (28). It is worth noting that V100s and K80 have theoretical maximum TDP of 250W and 300W respectively. We also consider an AMD EPYC processor of 64 cores and a TDP of 200W (29). Hence, the estimated CPU energy for one physical core and two threads per GPU is of 3W.

Federated learning. We propose to use a uniform set of NVIDIA Tegra X2 devices to compose our FL devices (30). Indeed, such chips are embedded in various IoT devices including cars, smartphones, video game consoles and others, and can be viewed as a realistic pool of FL clients. NVIDIA Tegra X2 have a reported TDP comprised between 7.5W and 15W. Across our different runs, we observed an average of 10W and we kept this measurement as a basis. Finally clients are assumed to be located in the same geographical region.

A.2 Detailed results on CIFAR10 experiments

This section enriches the results described in Section 3 with the context on the training time and number of rounds needed to compute the reported CO₂ estimate. All the results are given in Table 2.

Setup CIFAR10	Hardware	Rounds or Epochs		Time (s)	Max Acc. (%)
		50%	60%		
centralized Training	V100	1	2	48	71.0
centralized Training	K80	1	2	84	71.0
FL (IID, 1 epoch)	Tegra X2	5	16	51.4	64.3
FL (IID, 5 epoch)	Tegra X2	4	9	257	65.0
FL (non-IID, 1 epoch)	Tegra X2	20	75	51.4	61.0
FL (non-IID, 5 epoch)	Tegra X2	5	11	257	63.5

Table 2: Run time results for FL and centralized training on the CIFAR10. “IID” and “non-IID” terms are employed to distinguish between clients that have an evenly distributed set of samples containing all the classes (IID) and clients that have more samples of certain classes (non-IID). The “Time” column corresponds to the run time per epoch for centralized training or run time per communication round for FL. The “Max Acc.” column reports the maximum testing accuracy obtained. Finally, the “Rounds or Epochs” column gives the number of rounds or epochs needed to reach the indicated level of testing accuracy.

It is worth noting that in this setup, FL is slower and offers worst performance compared to centralized training on CIFAR10. Since each client has 1/10 of the total dataset, each *local* epoch can be seen as 1/10 of a *global* epoch. The accuracy difference could be explained by the simple FedAVG strategy employed for FL that could lead to a loss of information through the weight averaging performed at each communication round, or by a shift in the running statistics contained in the batch-normalisation layers of the ResNet-18 model. Indeed, FedAVG averages the latter statistics certainly leading to a small shift at every round. However, better performance may also be obtained with FL as shown in Figure 1 by simply considering better setups (*i.e* number of clients, number of local epochs ...).

Carbon emission estimation example. Eq. 3 can be applied to Table 2 to compute the CO₂ emitted to reach 50% of accuracy with the *FL (IID, 5 epoch)* setup as:

$$16 \times 0.9746 \times 5 \times \left(\frac{51.4}{3600} \times 10 \right) = 11.13g, \tag{4}$$

with 16 the number of rounds, 0.9746 the energy conversion rate of China, 5 the number of selected clients, 51.4 the time in seconds needed to complete one round and 10 the power consumed by the Nvidia TX2.

A.3 Carbon footprint estimation with FashionMNIST

The CO₂ estimation methodology detailed in Section 2 and applied to CIFAR10 in Section 3 is extended in this section to the FashionMNIST dataset for further context.

FashionMNIST consists in 60,000 training images of size 28 × 28 distributed across 10 classes, and 10,000 test samples. Since there is no natural user partitioning of this dataset, we follow the same IID and non-IID protocol than for the CIFAR10 experiments. However, for the sake of diversity, the model architecture considered for this experiment is reduced to a simple CNN with two convolutional layers with a kernel size of 5, followed with 2 fully connected layer composed by 512 hidden neurons and a final layer of size 10. In this setup, each client only performs 1 local epoch per round.

Setup	Hardware	Rounds or Epochs		Time (s)	Max Acc. (%)
		85%	90%		
FashionMNIST					
centralized Training	V100	2	5	5	92.0
centralized Training	K80	2	5	13.5	92.0
FL (IID, 1 epoch)	Tegra X2	8	26	5.6	92.0
FL (non-IID, 1 epoch)	Tegra X2	30	50	5.6	91.0

Table 3: Run time results for FL and centralized training on the FashionMNIST dataset. “IID” and “non-IID” terms are employed to distinguish between clients that have an evenly distributed set of samples containing all the classes (IID) and clients that have more samples of certain classes (non-IID). The “Time” column corresponds to the run time per epoch for centralized training or run time per communication round for FL. The “Max Acc.” column reports the maximum testing accuracy obtained. Finally, the “Rounds or Epochs” column gives the number of rounds or epochs needed to reach the indicated level of testing accuracy.

Table 3 reports the run time observed with the different setups. As expected, FL remains slower compared to centralized training. However, certainly due to the simplicity of the task and the neural network architecture, both FL and centralized learning achieve the same level of maximum accuracy (92%).

Country/CO ₂ (g)	V100		K80		FL	FL
	PUE = 1.67		PUE = 1.11		IID	non-IID
USA	1.6	5.2	1.1	3.5	1.1	2.1
China	2.9	9.2	1.9	6.2	2.0	3.8
France	0.2	0.8	0.2	0.5	0.2	0.3

Table 4: CO₂ emissions (expressed in grams, *i.e.* **lower is better**) for centralized training and FL on FashionMNIST. Emissions are calculated once the top-1 accuracy on the test set reaches 90%. The number of epoch reported on the FL column relates to the number of local epoch done per client. “IID” and “non-IID” terms are employed to distinguish between clients that have an evenly distributed set of samples containing all the classes (IID) and clients that have more samples of certain classes (non-IID).

Interestingly, with such a simple task, FL is almost as efficient as centralized training while considering Tesla V100. Hence, FL is even greener than a data center solution relying on Tesla K80. In summary, and has already shown in Section 3, the comparison between FL and centralized training highly depends on: 1. the efficiency of the considered data center (*i.e.* PUE and GPU efficiency). 2. The partitioning of the FL dataset (*i.e.* IID vs non-IID). 3. The FL setup.

A.4 Carbon footprint estimation with ImageNet

The CO₂ estimation methodology is also extended to the ImageNet dataset. To the best of our knowledge, this is the first time that ImageNet-scale experiments are performed using FL. ImageNet benchmarks follow the ILSVRC-2012 partitioning with 1.2M pictures for training and 50K images for testing (31). The model is trained with plain SGD and momentum as well.

Benchmarks conducted with ImageNet solely rely on the IID partitioning, with total of 40 clients. Therefore, each of the 40 clients has an even number of samples per class. Then, 10 clients are randomly picked in every FL round to perform 3 epochs of local training.

Table 5 reports the run time observed with both centralized learning and FL, and Table 6 reports the CO₂ emission observed. Centralized training reached 50% of top-1 accuracy in 5 epochs and 5.5

Setup ImageNet	Hardware	Rounds or Epochs 50%	Time (s)
centralized Training	V100	5	3,840
FL (IID, 3 epochs)	Tegra X2	25	3,840

Table 5: Run time results for FL and centralized learning on ImageNet dataset. “IID” terms means that clients have an evenly distributed set of samples containing all the classes. The “Time” column corresponds to the run time per epoch for centralized training or run time per communication round for FL, and the “Rounds or Epochs” column gives the number of rounds or epochs needed to reach the indicated level of testing accuracy.

Country/CO ₂ (g) ImageNet	V100 PUE = 1.67	V100 PUE = 1.11	FL (IID) 3 epochs
USA	1,230	820	1,460
China	2,290	1,500	2,600
France	180	120	210

Table 6: CO₂ emissions (expressed in grams, *i.e.* lower is better) for centralized training and FL ImageNet. Emissions are calculated once the top-1 accuracy on the test set reaches 50%. The number of epoch reported on the FL column relates to the number of local epoch done per client. “IID” terms means that clients have an evenly distributed set of samples containing all the classes.

hours compared to 25 rounds and 26.7 hours for FL. As expected, and due to much lower compute capabilities of FL devices, the training time is much longer than centralized training.

We can see from Table 5 that the same as for CIFAR10 dataset, FL is more polluting than V100-equipped centralized training, but the gap is much smaller. It is only 1.1 and 1.7 times more with PUE equals to 1.67 and 1.11 respectively.

A.5 Joint optimization to reduce FL carbon footprint

Based on our FL CO₂ footprint method, we propose a formalization that integrates carbon emissions into the common neural network optimization process to lower the final *Carbon Cost*.

As demonstrated in the Section 3, the outcome of the comparison between FL and centralized training highly depends on: 1. the efficiency of the considered data center (*i.e.* PUE and GPU efficiency). 2. The partitioning of the FL dataset (*i.e.* IID vs non-IID). 3. An optimal FL setup (*i.e.* number of local epochs, clients ...).

Unfortunately, the first two points can not be easily tweaked in realistic scenarios as they depend on the physical environment related to a specific task. Therefore, this section proposes to formalize the third identified lever as a joint optimization problem and validate empirically the interest of different FL setup to reduce the total amount of released CO₂.

Minimising CO₂. To achieve a proper reduction in carbon emissions, the latter goal must be defined as an objective:

$$\min_{r,n,t} r c_{rate} n (t * e_{clients} + 5s) = \min_{r,n,t} F(r, n, t). \quad (5)$$

Then, we must define the second objective relating to the performance of the trained model:

$$\max_{w \in \mathbb{R}^d} \frac{1}{|N|} \sum_{i \in dataset} t_i \mathbb{1}\{f(x_i) = t_i\} = \min_{w \in \mathbb{R}^d} \frac{|N|}{\sum_{i \in dataset} t_i \mathbb{1}\{f(x_i) = t_i\}}, \quad (6)$$

$$= \min_{w \in \mathbb{R}^d} \frac{1}{G(w)}, \quad (7)$$

with w corresponding to the model trainable parameters, $|N|$ the size of the dataset, t_i the ground truth for the sample i , and $f(x_i)$ the posterior probabilities obtained for this sample. Note that Eq. 6 is turned into a minimisation problem to avoid a more complex *min-max* optimization problem. Finally, both objectives are combined into a single problem as:

$$\min_{r,n,t} \frac{F(r, n, t)}{G(w)} = \min_{r,n,t} Carbon Cost. \quad (8)$$

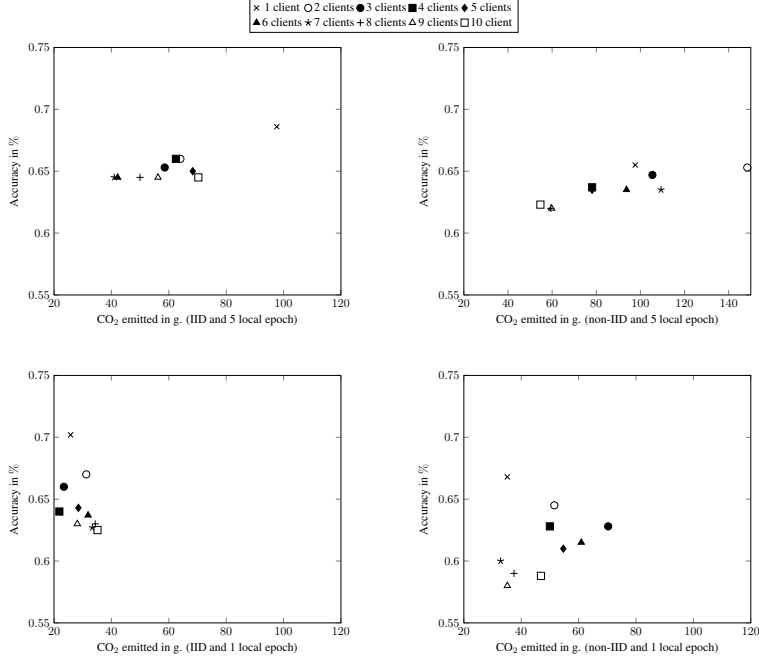


Figure 1: Scatter plot illustrating the relation between the best observed test accuracy on the CIFAR10 dataset (y-axis) and the amount of emitted CO₂ in grams (x-axis) with respect to the number of randomly selected clients. “IID” and “non-IID” correspond to the corpus partitioning strategy that is employed as described in Section 3. It is nearly impossible to pick the right FL setup to maximise the performance and minimise the CO₂ emissions without an appropriate algorithm. However, the latter solution allows important CO₂ savings. Please note that the x-axis scale of the top-right plot is different due to a key increase in CO₂ emissions.

It is worth noting that Eq. 8 is optimized with respect to r, n, t but not w , because $G(w)$ and $F(r, n, t)$ are dynamically dependent on each other (*i.e.* it tends to emit more when the performance improves). Indeed, the former is a function of neural parameters trained via gradient descent while the latter is a function of hyper-parameters often manually tuned. However, building a bridge between both would ensure a nice blend between the accuracy of the model and the environmental impact of the training procedure.

Empirical validation. To motivate further research on novel FL algorithms that could dynamically change the design of an experiment, we propose to visualize the value of Eq. 8 by varying three parameters on the CIFAR10 image classification task. In the definition of $F(r, n, t)$, r (number of rounds) mostly depends on n (number of selected clients) and t . Moreover, t is a variable depending on multiple factors including the computation and networking capabilities of the client, the number of local epoch, and the size of the local dataset. To properly analyse the variation of the *Carbon Cost*, we propose to vary n from 1 to 10, the number of local epoch between 1 and 5 and the type of partitioning of the local dataset (*i.e.* IID or non-IID). Indeed, all the other variables are directly related to physical or task-specific constraints that are commonly fixed for a certain experimental protocol. All the FL models are then trained for 500 rounds and the carbon emission estimations are computed on the best test accuracy observed (the detailed results are reported in Table 7). Finally, the *Carbon Cost* is plotted on Figure 1 with the amount of emitted CO₂ (*i.e.* Eq. 5) and the best accuracy (*i.e.* Eq. 6) on the x and y axes respectively.

Interestingly, it is nearly impossible to find any clear winning FL setup from Figure 1. As an example, a single client obtains the best *Carbon Cost* ($25.8/0.70 = 36.7$) with one local epoch, but also becomes the worst possible solution with 5 local epochs ($97.6/0.68 = 142.3$). As expected, certain tendencies are clearly visible with this graph. First, on the specific case of CIFAR10, an increasing number of local epoch leads to an average increase of the produced CO₂. Then, a non-IID partitioning is responsible for stronger variations in the observed *Carbon Cost* compared to IID. The latter finding suggests that further efforts should be put into developing FL methods robust to heterogeneous datasets.

Figure 1 clearly demonstrates the importance of optimizing the *Carbon Cost* (Eq. 8) rather than solely training our models with respect to the training accuracy. Indeed, estimating precisely the CO₂ emissions of a specific FL setup without a prior training of the model is impossible. Therefore, the development of novel FL algorithms able to dynamically change the number of selected clients or local epochs is of crucial interest to lower the final environmental cost of our deep learning models.

Clients/round	Test Acc 60%			Acc	Stable Acc		
	Rounds	CO ₂ (g)	Carbon Cost		Rounds	CO ₂ (g)	Carbon Cost
IID, 5 local epochs							
1	14	5.47	9.11	68.6%	250	97.64	142.33
2	14	10.94	18.23	66.0%	82	64.05	97.05
3	9	10.55	17.58	65.3%	50	58.58	89.72
4	9	14.06	23.43	66.0%	40	62.49	94.68
5	9	17.58	29.29	65.0%	35	68.35	105.15
6	8	18.75	31.25	64.5%	18	42.18	65.40
7	8	21.87	36.45	64.5%	15	41.01	63.58
8	7	21.87	36.45	64.5%	16	49.99	77.51
9	8	28.12	46.87	64.5%	16	56.24	87.20
10	8	31.25	52.08	64.5%	16	70.30	109.00
IID, 1 local epochs							
1	28	2.19	3.65	70.2%	330	25.78	36.72
2	24	3.75	6.25	67.0%	200	31.25	46.63
3	19	4.45	7.42	66.2%	100	23.43	35.40
4	16	5.00	8.33	64.0%	70	21.87	34.17
5	16	6.25	10.42	64.3%	73	28.51	44.34
6	16	7.50	12.50	63.7%	68	31.87	50.03
7	17	9.30	15.49	62.7%	61	33.35	53.20
8	16	10.00	16.66	63.0%	55	34.37	54.56
9	14	9.84	16.40	63.0%	40	28.12	44.64
10	17	13.28	22.13	62.5%	45	35.15	56.24
non-IID, 5 local epochs							
1	43	16.79	27.99	65.5%	250	97.64	149.07
2	16	12.50	20.83	65.3%	190	148.41	227.28
3	15	17.58	29.29	64.7%	90	105.45	162.99
4	12	18.75	31.25	63.7%	50	78.11	122.63
5	11	21.48	35.80	63.5%	40	78.11	123.01
6	12	28.12	46.87	63.5%	40	93.74	147.62
7	10	27.34	45.57	63.5%	40	109.36	172.22
8	11	34.37	57.28	62.0%	19	59.37	95.75
9	10	35.15	58.58	62.0%	17	59.76	96.38
10	9	35.15	58.58	62.3%	14	54.68	87.77
non-IID, 1 local epochs							
1	250	19.53	32.55	66.8%	450	35.15	52.62
2	135	21.09	35.15	64.5%	330	51.55	79.93
3	90	21.09	35.15	62.8%	300	70.30	111.95
4	75	23.43	39.06	62.8%	160	49.99	79.61
5	75	29.29	48.82	61.0%	140	54.68	89.64
6	75	35.15	58.58	61.5%	130	60.93	99.07
7	60	32.81	54.68	60.0%	60	32.81	54.68
8	NA	NA	NA	59.0%	60	37.49	63.55
9	NA	NA	NA	58.0%	50	35.15	60.61
10	NA	NA	NA	58.8%	60	46.87	79.70

Table 7: Details of the results obtained on CIFAR10 with multiple FL setups. “IID” and “non-IID” terms are employed to distinguish between clients that have an evenly distributed set of samples containing all the classes (IID) and clients that have more samples of certain classes (non-IID). The “Max Acc.” column reports the maximum testing accuracy obtained. The “Rounds” column gives the number of rounds needed to reach the indicated level of testing accuracy. Finally, “Carbon Cost” numbers are obtained by applying Eq. 8 (lower is better).