Benchmarking the environmental footprint of BERT models

Frederic Sadrieh
Hasso Plattner Institute
Prof.-Dr.-Helmert-Straße 2-3
14482 Potsdam
frederic.sadrieh@student.hpi.de

Abstract

Scaling large language models (LLMs) has led to remarkable gains in accuracy, but it also significantly increases resource consumption and CO_2e emissions. While performance improvements are well documented, the environmental footprint of these models is frequently under-reported. In this paper, we provide a comprehensive analysis of CO_2e emissions across various encoder-only architectures, with a particular focus on English and German BERT models. We investigate how architectural modifications, pretraining strategies, and data adjustments influence model efficiency. Our analysis reveals that, over time, environmental impact diminishes even as performance continues to improve. Using BERTchen we highlight the challenges in accurately reporting emissions, particularly without insider knowledge. Finally, we discuss the importance of incorporating CO_2e emission metrics in model evaluation and propose guidelines to standardize their reporting in future research.

1 Introduction

Large language models (LLMs) have experienced unprecedented scaling since the release of Kaplan et al. (2020), while scaling leads to significant performance gains, it requires more computational power and thus emits more CO_2e. The performance improvements have been thoroughly examined, but the environmental impact remains under-reported. This paper aims to bridge that gap by estimating the CO_2e emissions of well-known English and German encoder-only models.

Early versions of BERT models (Devlin et al., 2019; Liu et al., 2019) exhibited loss plateaus during pretraining, resulting in inefficient training processes and high energy consumption. For instance, the pretraining of RoBERTa produced CO_2e emissions comparable to those generated by an average human in one year (see section 4). Over time, more efficient methods, such as replacing the GELU activation function (Hendrycks and Gimpel, 2016) with GeGLU (Shazeer, 2020), have not only reduced training times (Geiping and Goldstein, 2023; Portes et al., 2023) but have also paved the way for scaling BERT models more sustainably (Warner et al., 2024).

Motivated by these observations, our research addresses the following questions: How does the carbon footprint of encoder-only models evolve over time, and what are the best practices for accurately reporting these emissions? By answering these questions, we aim to provide a framework that enables researchers to compare model efficiency and encourages the development of environmentally sustainable models.

Our main contributions include a comprehensive overview of CO_2e emissions from encoder-only model pretraining. We provide empirical evidence showing that environmental impact decreases over time even as performance improves. Finally, we propose guidelines to standardize emissions reporting in future research, thereby setting a foundation for sustainable practices in model development.

2 Related Work

Environmental sustainability Chen et al. describe the environmental sustainability of AI. The goal is to reduce the carbon footprint of AI training. Chen et al. (2023) divides environmental sustainability into computational efficiency, the cost of training the model, and data efficiency, the cost of collecting labeled data. We will focus on computational efficiency because labeled data has become less relevant due to self-supervised pretraining of models like BERT (Devlin et al., 2019). Chen et al. show potential options to increase model efficiency like quantization, but do not focus on making operations more efficient like using FlashAttention (Dao et al., 2022).

Strubell et al. (2020) compare the CO_2e emissions from the training of early Transformer models with the emissions from everyday items. For example, they find that BERT pretraining has the same emissions as a flight from New York to San Francisco (Strubell et al., 2020). Strubell et al. solve the problem of estimating the training emissions after the fact, by retraining all models on the same hardware. Using the measured power consumption and the same assumed energy mix for all models, they make emission estimates (Strubell et al., 2020). While this allows for a good comparison between models, it is not able to accurately measure the real emissions caused by training. An example is the estimate of the Evolved Transformer in Strubell et al. (2020), which is 88 times larger than that made by Google, which trained the model and has access to internal data (Patterson et al., 2021).

Patterson et al. (2021) and Patterson et al. (2022) calculate the carbon footprint of model training, by its hardware and training time. Patterson et al. (2022) shows how to reduce the emissions through intelligent hardware choices. They do not focus on potential software choices to reduce the environmental impact.

Henderson et al. (2020) and Lacoste et al. (2019) provide trackers that calculate CO_2e emissions. These trackers make it much easier to get emissions estimates during development, but only track the power consumption of the processors, while estimating the PUE of the datacenter and the current energy mix Henderson et al. (2020); Lacoste et al. (2019). Thus, the values are still estimates and the real PUE and carbon intensity should be verified. In addition, it does not estimate the emissions of past models.

To the best of our knowledge, there is no research comparing the CO_2e emissions for BERT-style models.

BERT One of the first encoder-only Transformers (Vaswani et al., 2017) was BERT (Devlin et al., 2019). It was released in 2019 as a model for text encodings (Devlin et al., 2019). A loss plateau can be observed in BERT pretraining (Nagatsuka et al., 2021). The model activations and thus the loss remain unchanged in the plateaus (Ainsworth and Shin, 2020), therefore the author chose to pretrain for 4 days (Devlin et al., 2019). The longer pretraining leads to better performance, but also to higher CO_2e emissions, which are not reported in the paper (Devlin et al., 2019).

RoBERTa (Liu et al., 2019) improves the performance of BERT and requires only one day of pretraining, but uses 1024 V100 GPUs instead of 4 Cloud TPUs. Again, they do not report their emissions and use resources are not feasible for many researchers (Liu et al., 2019).

Several papers try to reduce the resource usage and thus the emissions (Geiping and Goldstein, 2023; Portes et al., 2023). Geiping and Goldstein want to pretrain crammed BERT in 13 exaFLOPs, which allows for 24 hours of pretraining on a RTX A6000 GPU. Omitting the Next Sentence Prediction (NSP) objective, various architectural changes, and different hyperparameters, lead to better results compared to a pretrained BERT within the same computational budget and comparable performance to the original BERT (Geiping and Goldstein, 2023).

Portes et al. train MosaicBERT in even less time, 1.13 hours and on 8 A100 GPUs. By using state-of-the-art methods like FlashAttention (Dao et al., 2022), ALiBi (Press et al., 2022) and bfloat16 (Wang and Kanwar, 2019) they are able to match the performance of BERT(Portes et al., 2023). The plateau problem is solved by using the GeGLU (Shazeer, 2020) activation function[1].

Recently, ModernBERT (Warner et al., 2024) has been released and scales BERT. Warner et al. use similar methods to MosaicBERT, but train for much longer. Therefore, they are able to create the current best BERT-style model (Warner et al., 2024).

All of these papers do not report their CO_2e emissions, and only some of them report their FLOPs (Geiping and Goldstein, 2023). All of them report their training time and resources used, which allows us to make educated guesses about energy consumption and emissions.

German BERT We would like to place an additional focus on the German BERT models in order to highlight the emissions over time on a second example. When attempting transfer, one can pretrain on a multilingual corpus (Devlin et al., 2019) or pretrain on a German corpus ((Chan et al., 2019), (Chan et al., 2020)). Most papers focus on the second approach and publish monolingual models (Chan et al., 2020). The disadvantage of this approach is that training requires roughly the same amount of computation for each language. For Ger-

[1]To find out why the ReLU-like activation function used in BERT causes these plateaus, see Ainsworth and Shin (2020)

man, the German BERT (Chan et al., 2019) and the better GBERT (Chan et al., 2020) both use the classic BERT architecture, so they inherit the same inefficiencies.

After the release of better English BERT models, new German models are released using the same architecture and pretraining recipes. An example of this is GottBERT (Scheible et al., 2020), which is the German version of RoBERTa.

BERTchen (Sadrieh, 2024) improves the MosaicBERT architecture in ways that were later used by Warner et al. (2024), such as using FlashAttention2 (Dao, 2023) or a longer sequence length. It also systematically studies data quality in low computational settings (Sadrieh, 2024), where most other models (like (Portes et al., 2023)) still use the classic C4 dataset (Raffel et al., 2020) as used in (Devlin et al., 2019). It provides us with the currently best German BERT model (Sadrieh, 2024) and a case study on how to report CO_2e emissions from training.

Again, none of the papers reported CO_2e emissions from training, underscoring the underreporting of such important metrics.

3 Methods

3.1 Overview of pretraining improvements

To limit the resource consumption of pretraining, one must perform fewer floating-point operations, either by limiting the number of steps or by increasing the efficiency of each step. We will discuss both approaches in detail below. Note that both approaches should be considered and are symbiotic.

Reducing the amount of steps An example of a step reduction is dropping the NSP target and thus removing the [CLS] token, as done in Sadrieh (2024). This results in more data being processed in each batch, and thus fewer steps per record. Using an in-domain tokenizer increases the information gain of each token (also done in Sadrieh (2024)), so fewer tokens need to be processed. Careful selection of data points, as in CuluraX (Nguyen et al., 2023), omits unhelpful tokens, again decreasing the number of tokens. Finally, making smarter architectural choices, such as using the GeGLU activation function (Shazeer, 2020) to avoid loss plateaus, can reduce the steps to convergence.

Increasing the step efficiency Step efficiency can be increased by either better hardware or more efficient hardware utilization. Two examples of better utilization are FlashAttention (Dao et al., 2022), which reduces the computational complexity of the attention calculation, and bfloat16 (Wang and Kanwar, 2019), which reduces floating point precision. Both allow for faster computation speeds and better utilization of hardware capabilities. Step efficiency is often measured in tokens per second, which (Geiping and Goldstein, 2023) finds to be the most important metric for efficient pretraining.

3.2 Reporting CO_2e emissions

Calculating the emissions For all BERT-style models presented in the related work, the authors did not publish their energy consumption or CO_2e emissions. To calculate the CO_2e emissions, the energy consumption of the training and the carbon intensity of the energy are needed.

In large data centers, compute power is only a fraction of total power consumption (Dayarathna et al., 2016). The impact of pretraining on data center power consumption is difficult to measure. Therefore, the Power Usage Effectiveness (PUE) (Malone and Belady, 2006) is used to estimate total power consumption. The PUE is given by the total data center power consumption divided by the data center IT power consumption (Malone and Belady, 2006) and is used to scale the observed power consumption of compute resources (Patterson et al., 2022).

Measuring the carbon intensity of energy requires knowledge of the type of energy the datacenter receives (Lacoste et al., 2019). Since most data centers do not have their own power plant, the energy mix can vary drastically not only by location but also by time (Lacoste et al., 2019). There is also the question of opportunity cost: Would the CO_2e intensive power plants be shut down if the data center's energy consumption were lower due to more efficient training? If so, then efficient training would lead to less energy consumption and less carbon intense energy. Although, this idea seems far-fetched in one day RoBERTa used 11.640 MWh (see Table 1) and all of Germany 352.929 MWh [2], this means that RoBERTa would have used 3.29% of all the energy used in Germany.

Since neither of these metrics were reported in the papers, we have to make educated assumptions.

[2] See the annual energy consumption of 2018 https://www.destatis.de/EN/Themes/Society-Environment/Environment/Material-Energy-Flows/Tables/electricity-consumption-households.html (accessed 19.12.2024)

Following Patterson et al. (2022), we compute the energy consumption of model training by hours to train · number of processors · average power per processor · PUE. To get the CO_2e emissions, we multiply the energy consumption by the carbon intensity (tCO_2e per MWh) (Patterson et al., 2022). After Patterson et al. we take a PUE of 1.58 and a carbon intensity of 0.429 tCO_2e per MWh. We use the power consumption as stated in the resource specifications. Note that this assumes efficient use of the GPUs, which we found possible in our own logging, but not always the case. Since the papers did not report the power consumption of their resources, we cannot make a better estimation.

Performance per emission In addition to reporting the raw energy consumption and CO_2e emissions of the models, we detail how much the emissions were worth in terms of downstream performance. For the English models, we use the widely used GLUE benchmark, as all models have been evaluated against it. We acknowledge that GLUE has been criticized for being saturated (Warner et al., 2024) and there are known problems with individual tasks, but do not find it feasible to finetune all models on a better benchmark. The GLUE results are taken from each of the papers individually[3]. There are replication studies that include different fine-tuning approaches or other changes that lead to better results after the fact (see Portes et al. (2023) for better results for BERT), which we exclude.

For the German models, we take the results from Table 5 in (Sadrieh, 2024). The authors evaluate on the German SQuAD (Rajpurkar et al., 2016) dataset GermanQuAD (Möller et al., 2021). While it does not test as many tasks, there is a lack of broad German benchmarks that are as widely used as GLUE.

For all models, we divide their score by the power consumed and CO_2e emitted in pretraining to get the average resource cost per score point. It is important to note that, as seen in Kaplan et al. (2020), the scaling laws are smooth power laws, meaning that the first score point is much cheaper than the later ones. This means that we cannot directly compare the average cost of two models with very different performance. If the better performing model also requires more resources, we cannot be sure that it is less efficient at each score point. It could be the performance delta that makes the model more expensive. If the cheaper model is better, it shows a real evolution in efficient pretraining.

4 Experimental Results

CO_2e emissions In Table 1 we report the estimated power consumption and CO_2e emissions for English and German BERT-style models[4]. RoBERTa scales BERT, without many efficient improvements, thus RoBERTa has by far the largest CO_2e emission of all models (see Table 1). It emitted as much CO_2e as an average human in one year (Strubell et al., 2020) and had the energy consumption of 6 German one-person households in 2018[5].

Crammed BERT and MosaicBERT reduce emissions by an order of magnitude compared to BERT. MosaicBERT is even twice as efficient as crammed BERT. ModernBERT reverses this trend and again trains much longer and emits more CO_2e than BERT.

For the German models, we observe a similar phenomenon: emissions decrease over time. There is also an interesting shift between languages: Although GottBERT is the German version of RoBERTa, it uses much less resources. This is due to the fact that the model is trained on more efficient hardware, and fewer training steps, trained for only 100 thousand steps (Scheible et al., 2020) instead of the 1 million steps (Liu et al., 2019). This results in a smaller performance gain compared to the original BERT variant (compare Table 2 and Table 3). The other two German models also required less energy, although the effect is smaller.

The most efficient model, BERTchen, uses only as much power as a coffee maker making 21 coffees[6]. While writing this paper, the laptop probably consumed more power than the final model training,

[3]The exact sources are: BERT Devlin et al. (2019) Tabel 1; RoBERTa Liu et al. (2019) Table 8; crammed BERT Geiping and Goldstein (2023) Table 3 last row; MosaicBERT Portes et al. (2023) Table S1 first row; ModernBERT Warner et al. (2024) Table 1

[4]The sources for the power draw of the processors are: TPU v3 Patterson et al. (2021), TPU v4 Jouppi et al. (2023), Tesla V100 NVIDIA (2018), RTX A6000 NVIDIA (2022b), A100 NVIDIA (2022a), and H100 NVIDIA (2024).

[5]See https://www.destatis.de/EN/Themes/Society-Environment/Environment/Material-Energy-Flows/Tables/electricity-consumption-households.html (accessed 19.12.2024)

[6]See https://www.siliconvalleypower.com/residents/save-energy/appliance-energy-use-chart (accessed 19.12.2024)

	BERT	RoBERTa	Cram. BERT	MosaicBERT	ModernBERT	GBERT	GottBERT	BERTchen
Release date	2019	2019	2023	2023	2024	2020	2020	2024
Compute(h)	96	24	24	1.13	245.6	168	28.8	4
Resource	TPU v3	Tesla V100	RTX A6000	A100 80GB	H100 80GB	TPUs v3	TPU v4	A100-SXM4-40GB
#Resources	4	1024	1	8	8	1	1	1
Processor power(W)	283	300	300	400	700	283	170	400
Power(MWh)	0.172	11.640	0.011	$5.713 \cdot 10^{-3}$	2.173	0.075	$7.736 \cdot 10^{-3}$	$2.528 \cdot 10^{-3}$
CO_2e(t)	0.074	4.994	$4.719 \cdot 10^{-3}$	$2.450 \cdot 10^{-3}$	0.932	0.032	$3.318 \cdot 10^{-3}$	$1.085 \cdot 10^{-3}$

Table 1: Comparison of power consumption and CO_2e emissions for English and German BERT-style models. Processor power is the maximum power consumed by the processor per specification. Power is the total data center power consumed by the pretraining run. Steady improvements in pretraining and model architecture lead to a reduction in CO_2e emissions over time. The trend is reversed by ModernBERT (Warner et al., 2024), which uses the efficiency gains for scaling up rather than down.

the break-even would be between 51–126 hours[7]. Especially interesting is the comparison between RoBERTa and BERTchen, since one can pretrain BERTchen about, 4600 times before it emits the same amount of CO_2e. These CO_2e emissions allow researchers to iterate over many more model architectures. If the authors flew to a conference to present BERTchen (e.g. from Frankfurt to San Francisco), they could pretrain BERTchen 3410 times before reaching the same CO_2e emissions[8].

Performance vs emissions BERT pretraining is relatively inefficient, and while crammed BERT and MosaicBERT do not outperform BERT, they achieve comparable results much more efficiently (see Table 2). Crammed BERT achieves 15.5 times the performance per CO_2e compared to BERT, and MosaicBERT is even twice as efficient as crammed BERT. These improvements show that models will become more efficient over time, drastically reducing the carbon footprint of AI.

RoBERTa requires much more resources than ModernBERT to surpass the performance of BERT. ModernBERT is even better than RoBERTa. Since ModernBERT is based on MosaicBERT, it shows the importance of down-scaling. By iteratively down-scaling the resource consumption, efficient model training methods have been developed that make scaling up efficient.

This practice must be distinguished from developing on small models and then scaling up, as is done in Raffel et al. (2020). Raffel et al. (2020) has been criticized by Tay et al. (2021), which shows that best practices from small models may not be directly transferable to much larger models. In contrast, the parameter counts of all presented models are between 110-150 million[9], making the efficient methods transferable without the challenges presented in Tay et al. (2021).

For the German models, Table 3 lists the performance on the GermanQuAD task and how much energy and CO_2e is required for the scores.The trend is even stronger than for the English models, not only is the performance per model increasing, but the environmental impact is decreasing.

5 Discussion on environmental sustainability

Problems with reporting CO_2e emissions None of the papers have published their CO_2e emissions, and predicting them after the fact is much harder. A prime example of such a estimation comes from Strubell et al. (2020), which predicts that the training of the Evolved Transformer (So et al., 2019) emitted 284 tCO_2e. Patterson et al. refines this estimate to only 3.2 tCO_2e. The extreme difference is partly due to a misunderstanding of the model structure, but also due to different assumed power consumption, PUE, and energy mix. While Strubell et al. (2020) had to make assumptions about the cluster, (Patterson et al., 2021) had access to Google's internal data. It is almost impossible for outsiders to know how much energy was actually consumed during training and on which cluster the model was trained.

There are different PUE and carbon intensity values in the literature. While we use a PUE of

[7] See https://www.siliconvalleypower.com/residents/save-energy/appliance-energy-use-chart (accessed 19.12.2024)

[8] https://co2.myclimate.org/en/flight_calculators puts one roundtrip economy direct flight from FRA to SFO at 3.7 tCO_2e (accessed 13.02.2025)

[9] The exact parameter numbers as listed on the respective Huggingface model cards and papers are: BERT: 110M, RoBERTa: 125M, crammed BERT: 145M, MosaicBERT: 137M, ModernBERT: 150M

	BERT	RoBERTa	Cram. BERT	MosaicBERT	ModernBERT
GLUE score (avg)	79.6	86.4	78.6	79.6	**88.4**
Score per MWh	462.8	7.4	7145.5	**13933.1**	40.7
Score per t CO_2e	1075.7	17.3	16656.1	**32489.8**	94.9

Table 2: Comparison of average GLUE scores and the scores per MWh and ton of CO_2e used in pretraining for English BERT-style models. While crammed BERT and MosaicBERT do not increase the performance of BERT, they make it much more efficient. RoBERTa and ModernBERT increase performance by using more resources. The highest scores are shown in bold.

	GBERT	GottBERT	BERTchen
GermanQuAD F1	55.0	55.1	**95.1**
GermanQuAD EM	72.8	73.1	**91.9**
F1 per MWh	733.1	7127.7	**37618.7**
F1 per t CO_2e	1718.1	16618.4	**87649.8**
EM per MWh	970.9	9444.2	**36352.8**
EM per t CO_2e	2275.6	22019.3	**84700.5**

Table 3: Comparison of GermanQuAD F1 and exact match scores and the scores per MWh and ton of CO_2e used in pretraining for German BERT-style models. Performance increases over time along with sustainability. The highest scores are shown in bold.

1.58 from Patterson et al. (2022), they also mention that some cloud providers have a PUE of 1.1. In addition, Strubell et al. use a carbon intensity of 0.954 pounds CO_2e per KWh (0.433 tons CO_2e per MWh). We use 0.429 tons CO_2e per MWh from Patterson et al. (2022), while Patterson et al. (2021) uses 0.080 tons CO_2e per MWh for the Google datacenter. With the same measured energy consumption, the estimated CO_2e emission can vary by a factor of 8 depending on which PUE and carbon intensity is chosen.

Considering these factors, we can be sure that our results do not reflect the real emissions of the training runs, but should be considered as rough estimates. This raises the question of what authors of new models should do to best report their environmental impact.

Reporting the average measured energy consumption of the runs is relatively trivial: the tools presented in Henderson et al. (2020) and Lacoste et al. (2019) are able to track the energy consumed. While they make predictions about PUE and carbon intensity, it would be beneficial to also report the cluster and time at which the run was trained.

Reporting more accurate CO_2e emissions We report more accurate estimates of CO_2e emissions from BERTchen (Sadrieh, 2024). We contacted the authors and the administrators of the cluster on which the run was trained. The cluster consumed a total of 3.52 GWh and emitted 917 tCO_2e. Of the 3.52 GWh, 2.51 GWh were used by the servers (all as of 13.02.2025). So the PUE is 1.402 and the carbon intensity is 0.261 tCO_2e per MWh. Sadrieh logged an average power consumption of 250W. These updated numbers give us a power consumption of $1.402 \cdot 10^{-3} MWh$ and CO_2e emissions of $3.659 \cdot 10^{-4}$. In Table 1, the CO_2e emissions for BERTchen are 3 times higher. This example underlines two things: It is impossible to make accurate predictions without insider knowledge, and it is important for server administrators to keep these numbers available to researchers.

Should we even report the CO_2e emissions? Besides the problem of estimating emissions after the fact, there is the question of what is the benefit of them. If I want to retrain the model, I will get very different CO_2e emissions depending on my hardware, and one is not able to accurately verify the reported CO_2e emissions. A bad actor could create a supposedly more efficient model by lowering his assumed PUE or carbon intensity. Thus, reporting CO_2e emissions is not done by most research. Many papers only publish their floating-point operations. These can be measured with much fewer assumptions and are hardware independent.

We believe there is a strong case for reporting both. While FLOPs are a good measure of throughput, they do not show real-world impact. To measure the efficiency of a model, the efficiency of the software, the hardware, their interaction, and the environmental impact of the hardware are critical.

6 Conclusion

Our research extends the discourse on environmentally conscious model training. We predict how much CO_2e the pretraining of the most prominent

encoder-only models has emitted and compare it to their performance. We find that the environmental impact of encoder-only models is decreasing, while performance is generally increasing.

These findings contradict the popular narrative of power-hungry machine learning. There is reason to be hopeful about the performance and efficiency gains that machine learning will deliver in the coming years. To achieve these goals, we need accountability and comparability between models. None of the papers have published their environmental impact, and we discuss the challenges of predicting CO_2e emissions after the fact. In order to incentivise future researchers to continuously improve the efficiency of the models, they should report the:

- FLOPs of pretraining
- Energy consumed by the resource in pretraining
- Cluster the model was trained on and time
- Estimated PUE and Carbon Intensity
- Estimated power consumption and CO_2e emissions

7 Future Work

One could extend this work by looking at trends for other languages and other model architectures. We have only focused on high-resource languages and the base variant of the models. It would be interesting to see what the trend is for small decoder-only models.

In addition, it might be of interest to establish the scaling laws of CO_2e emission, similar to Kaplan et al. (2020). We have no evidence that the models presented are optimal in terms of CO_2e emission per performance. An efficient model such as BERTchen can be used to explore how downscaling of CO_2e emissions affects performance.

8 Limitations

As mentioned in section 5, CO_2e emissions can only be considered as rough estimates. The exact emissions are impossible for outsiders to predict after the fact.

We have not included any fine-tuning or inference in the calculation of CO_2e emissions. Although these are generally less costly than pretraining, in practice they are done much more frequently. A model may offset the cost of pretraining through more efficient inference or less fine-tuning. These costs are difficult to estimate because they occur over long periods of time. To our knowledge, no work has attempted to predict CO_2e emissions in these phases for encoder-only models.

In addition, before pretraining the final model, the authors must first try out different approaches. The experimentation phase often emits much more CO_2e than the final pretraining, but is much harder to quantify. Strubell et al. tries to quantify this in an example. These costs are even harder for outsiders to report, as most only report the time for the final model run.

References

Mark Ainsworth and Yeonjong Shin. 2020. Plateau phenomenon in gradient descent training of ReLU networks: Explanation, quantification and avoidance. *Siam Journal On Scientific Computing*, 43:A3438–A3468.

Branden Chan, Timo Möller, Malte Pietsch, and Tanay Soni. 2019. Open Sourcing German BERT Model.

Branden Chan, Stefan Schweter, and Timo Möller. 2020. German's next language model. In *Proceedings of the 28th international conference on computational linguistics*, pages 6788–6796, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhenghua Chen, Min Wu, Alvin Chan, Xiaoli Li, and Yew-Soon Ong. 2023. Survey on AI sustainability: Emerging trends on learning algorithms and research challenges [review article]. *IEEE Computational Intelligence Magazine*, 18(2):60–77.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

Miyuru Dayarathna, Yonggang Wen, and Rui Fan. 2016. Data center energy consumption modeling: a survey. *IEEE Communications Surveys & Tutorials*, 18(1):732–794.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north American chapter of the association for computational linguistics: Human language technologies, volume 1 (long and short papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonas Geiping and Tom Goldstein. 2023. Cramming: Training a language model on a single GPU in one day. In *International conference on machine learning*, pages 11117–11143. PMLR.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43.

Dan Hendrycks and Kevin Gimpel. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415. ArXiv: 1606.08415.

Norm Jouppi, George Kurian, Sheng Li, Peter Ma, Rahul Nagarajan, Lifeng Nai, Nishant Patil, Suvinay Subramanian, Andy Swing, Brian Towles, Clifford Young, Xiang Zhou, Zongwei Zhou, and David A Patterson. 2023. TPU v4: An optically reconfigurable supercomputer for machine learning with hardware support for embeddings. In *Proceedings of the 50th annual international symposium on computer architecture*, Isca '23, New York, NY, USA. Association for Computing Machinery. Number of pages: 14 Place: Orlando, FL, USA tex.articleno: 82.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

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

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Christopher Malone and Christian Belady. 2006. Metrics to characterize data center & IT equipment energy use. In *Proceedings of the digital power forum, richardson, TX*, volume 35.

Timo Möller, Julian Risch, and Malte Pietsch. 2021. GermanQuAD and GermanDPR: Improving non-english question answering and passage retrieval. ArXiv: 2104.12741 [cs.CL].

Koichi Nagatsuka, Clifford Broni-Bediako, and Masayasu Atsumi. 2021. Pre-training a BERT with curriculum learning by increasing block-size of input text. In *Proceedings of the international conference on recent advances in natural language processing (RANLP 2021)*, pages 989–996, Held Online. INCOMA Ltd.

Thuat Nguyen, Chien Van Nguyen, Viet Dac Lai, Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, Ryan A. Rossi, and Thien Huu Nguyen. 2023. CulturaX: a cleaned, enormous, and multilingual dataset for large language models in 167 languages. ArXiv: 2309.09400 [cs.CL].

NVIDIA. 2018. TESLA V100: The Most Advanced Data Center GPU Ever Built.

NVIDIA. 2022a. A100: The Most Powerful Compute Platform for Every Workload.

NVIDIA. 2022b. RTX A6000: Amplified Performance for Professionals.

NVIDIA. 2024. NVIDIA H100 Tensor Core GPU.

David Patterson, Joseph Gonzalez, Urs Hölzle, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean.

2022. The carbon footprint of machine learning training will plateau, then shrink. *Computer*, 55(7):18–28.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon Emissions and Large Neural Network Training. *arXiv preprint*. ArXiv:2104.10350 [cs].

Jacob Portes, Alexander R Trott, Sam Havens, Daniel King, Abhinav Venigalla, Moin Nadeem, Nikhil Sardana, Daya Khudia, and Jonathan Frankle. 2023. Mosaicbert: How to train bert with a lunch money budget. In *Workshop on efficient systems for foundation models@ ICML2023*.

Ofir Press, Noah A. Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. ArXiv: 2108.12409 [cs.CL].

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Frederic Sadrieh. 2024. BERTchen: Training the best and most efficient German BERT model.

Raphael Scheible, Fabian Thomczyk, Patric Tippmann, Victor Jaravine, and Martin Boeker. 2020. GottBERT: a pure german language model. ArXiv: 2012.02110 [cs.CL].

Noam Shazeer. 2020. GLU Variants Improve Transformer. *CoRR*, abs/2002.05202. ArXiv: 2002.05202.

David So, Quoc Le, and Chen Liang. 2019. The evolved transformer. In *International conference on machine learning*, pages 5877–5886. PMLR.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2020. Energy and Policy Considerations for Modern Deep Learning Research. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(09):13693–13696.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. 2021. Scale efficiently: Insights from pre-training and fine-tuning transformers. *arXiv preprint arXiv:2109.10686*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Shibo Wang and Pankaj Kanwar. 2019. BFloat16: The secret to high performance on Cloud TPUs.

Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. 2024. Smarter, Better, Faster, Longer: A Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. *arXiv preprint*. ArXiv:2412.13663 [cs].