FROM COMPUTATION TO CONSUMPTION: EXPLORING THE COMPUTE-ENERGY LINK FOR TRAINING AND TESTING NEURAL NETWORKS FOR SED SYSTEMS

Constance Douwes, Romain Serizel

University of Lorraine, CNRS, Inria, Loria, 54000, Nancy, France
constance.douwes@inria.fr, romain.serizel@loria.fr

ABSTRACT

The massive use of machine learning models, particularly neural networks, has raised serious concerns about their environmental impact. Indeed, over the last few years we have seen an explosion in the computing costs associated with training and deploying these systems. It is, therefore, crucial to understand their energy requirements in order to better integrate them into the evaluation of models, which has so far focused mainly on performance. In this paper, we study several neural network architectures that are key components of sound event detection systems, using an audio tagging task as an example. We measure the energy consumption for training and testing small to large architectures and establish complex relationships between the energy consumption, the number of floating-point operations, the number of parameters, and the GPU/memory utilization.

***Index Terms*—** Energy, deep learning, neural networks, FLOPs, parameters, training, inference, sound event detection

1. INTRODUCTION

Deep learning (DL) has become the principal focus of audio processing research, with numerous applications spanning various domains including sound event detection (SED) [1, 2], speech recognition [3, 4] and music generation [5, 6]. As models become increasingly powerful and datasets grow larger, the associated computational costs have exploded [7, 8, 9]. Yet, the true cost of computation often remains obscured, as many computations are carried out on remote infrastructures or data centers. Nevertheless, these energy-intensive processes involved in training and deploying high-performance models have a real environmental footprint linked to their demand for electricity [10, 11]. This raises significant concerns in the current context of climate change and efforts to limit global warming to below 2 degrees [12]. Even though models used in audio processing are smaller than those used in natural language processing, they still present similar problems [13, 14].

The trends described above are driven by an ongoing pursuit of outperforming previous state-of-the-art systems, even by a small margin. Recently, there has been a slight shift towards reporting and quantifying the environmental costs associated with these advances [15, 16]. In the audio processing domain in particular, significant efforts have been made to balance performance and energy in the context of sound event detection [17, 18] or speech recognition [14], and to emphasize the importance of considering quality metrics alongside energy footprint assessments in speech synthesis [13]. All of these studies call for a fair and reliable metric to assess the computational footprint that reflects the energy consumption while being hardware independent to enable accurate comparisons between models. Although work such as Speckhard et al. [19] shows a strong correlation between computational cost and energy consumption during inference for convolution-based models, to our knowledge similar investigations have not been conducted for training or for other architectures. Even if a few hundred experiments are sometimes required to train a model, the cost of the training phase represents only 10% to 20% of the total CO2 emissions of the associated machine learning usage, with the majority occurring during the inference phase [20]. However, as audio processing researchers, the majority of our energy consumption lies in the training phase, and should not be overshadowed.

In this article, we aim to understand the computational factors that impact the energy consumption for the training or testing deep learning models that compose SED systems. This study is conducted in the context of the DCASE challenge task 4, where participants have been required since 2022 [21] to report their energy consumption alongside computational factors such as the number of parameters and the number of operations. Specifically, we seek an indicator that can estimate the energy consumption based on computational measurements. This would allow us to estimate each system's consumption on the same hardware and provide fair comparisons between systems, extending the work of Ronchini et al.[18]. We focus our analysis on well-known architectures such as MLP, RNN, CNN and CRNN. CRNN is specifically the current architecture used in Task 4 of the DCASE Challenge [22]. We compute the number of parameters of the models and the number of floating point operations (FLOPs) as two potential candidate factors for energy consumption estimation. We show that as the number of operations increases, so does the energy consumption across all architectures during both the test and training phases. However, the relative increase in energy consumption varies between architectures and phases. We identify two distinct trends: one for MLP/RNN, and one for CNN/CRNN. Finally, we identify relationship between energy consumption and GPU utilization during both training and testing phases, which could serve as a basis for future research on computational metrics.

In summary, our key contributions are :

- A comparative analysis of prominent architectures (MLP, CNN, RNN, CRNN) and their associated energy consumption.
- The identification of two distinct trends in energy consumption based on architecture type, notably distinguishing between MLP/RNN and CNN/CRNN architectures.
- A relative comparison of power usage between training and test stages.

2. METHODOLOGY

Computing and monitoring the computational and energy costs of the two phases of deep learning systems - training and inference

- is a complex endeavour. We present here our methodology for assessing both, mentioning previous work in these areas.

2.1. Computational cost

Traditional methods rely on metrics such as the size of the model (the number of parameters) and the number of floating-point operations (FLOPs) computed by the model to estimate the computational cost. While computing the number of parameters (or weights) of a model is straightforward, computing the number of operations can be a difficult task, especially for complex architectures, and this number is very sensitive to the size of the input/output. At inference, only forward calculations are performed, so the number of operations is the sum of all operations across all layers. We use the deepspeed profiler [23] to quantify these forward pass operations accurately. In contrast, training is a more complex process involving iterative forward and backward calculations. In particular, the backward pass also computes the gradient with respect to the parameters, the loss and update the weights. However, at the time of writing, no profiler provided the exact number of backward operations, so we derive this number using the ratio 2:1 as an approximation [24]. In total, the number of operations of a training iteration (forward and backward) is three times the number of operations of an inference (forward only).

2.2. Energy consumption

Several Python trackers have emerged to facilitate the computation of energy consumption [25]. In most of the trackers, the total consumption is calculated as the sum of the consumption of each component of the computer: GPU, CPU and RAM. In our study, we focus specifically on analysing the energy consumption of the GPU given by CodeCarbon [26]. Indeed, preliminary experiments have led us to conclude that while GPU power fluctuates, CPU power remains stable. Regarding ram energy, CodeCarbon estimates 3 watts per 8 GB, which also remains constant over time. We made sure that any increases in GPU power with the python trackers were correlated with energy consumption monitored on the system's baseboard management controller (BMC). We also monitor the GPU and memory utilization from Nvidia SMI query every 5 seconds to get the mean uses of the each experiment.

3. EXPERIMENTS

Our objective is to better understand the energy consumption at train and test and to relate it to computational cost of a given model and architecture. To achieve this, we evaluate different types and sizes of architectures for audio tagging systems.[1]

3.1. Task description

Audio tagging involves assigning one or multiple tags to an audio signal without any temporal information. For this experiment, we work on the real part of the DESED dataset [27]. This dataset contains 10-second audio clips recorded in domestic environments. We convert those recordings into mel-spectrogram representations with 128 bands, an FFT size of 2048 and a hop size of 256. We only take the first 64 frames as input, which corresponds to approximately the first 1 second of the audio signal. Although this significantly impacts the performance of the model, it reduce the system's complexity, allowing for more lightweight experiments, as we do not focus on performance but only on energy.

[1] https://github.com/ConstanceDws/toolbox_energy

Model	Num Layers	Hidden Sizes
MLP	1	512, 1024, 2048
	4	1024, 2048, 4096
	6, 10, 16, 32	4096
CNN	1	128, 256, 512, 1024
	2	128, 256, 384, 512, 768, 1024
	6	384, 768
RNN	1	128, 512, 1024, 2048
	4, 6	1024, 2048
	2, 10, 14	2048
CRNN	[1,1], [2,1], [1,2]	[64,64], [256,64], [512, 256]
	[2,2]	[728, 256]
	[1,2], [2,2]	[1024, 256]

Table 1: Summary of all the configurations tested in our experiment. For each number of layer, we tested different hidden sizes. For CRNN, the configurations first indicate the convolutional layers and then the recurrent layers.

3.2. Models

We implement four neural network architectures: multi-layer perceptron (MLP), convolutional neural network (CNN), recurrent neural network (RNN), and convolutional recurrent neural network (CRNN). For the MLP, we implement a series of linear layers followed by ReLU activation functions. For the CNN, we adopt a sequence of Conv2d, ReLU and MaxPool2d layers. For the RNN we use GRU cells and for the CRNN we start with Conv2d, ReLU and MaxPool2d layers followed by a GRU cell. All implementations are completed with a final linear layer and a sigmoid activation function that outputs a probability vector for the 10 classes. For each architecture, we systematically increase the number of layers and adjust the hidden sizes per layer, gradually scaling up to reach the full GPU memory capacity and utilization, resulting in 43 models. We present the summary of all the configurations tested in Table 1. We intentionally chose those configurations to achieve meaningful variations in the number of FLOPs without conducting redundant experiments.

3.3. Training and test

Our experiments diverge from the conventional research of accuracy performance. Instead, we train all models for a single epoch on the same Nvidia Tesla T4 GPU and monitor the energy of the training phase. To focus solely on architectural differences, we use a consistent batch size of 8. Although the choice of criterion, optimizer, and learning rate is crucial for model convergence, it does significantly impact energy measurements. Therefore, we employ the cross-entropy function as the criterion, fix the learning rate at 10^{-3}, and use the ADAM optimizer [28]. We did not include any validation steps in the training routine to isolate the effects of training. Instead, we measure the energy consumption during the test phase separately. The test phase involves running the model (inference) and computing the error. Although inference for such small models can generally be performed on the CPU, we ensure consistency with the training phase measurements by also running the

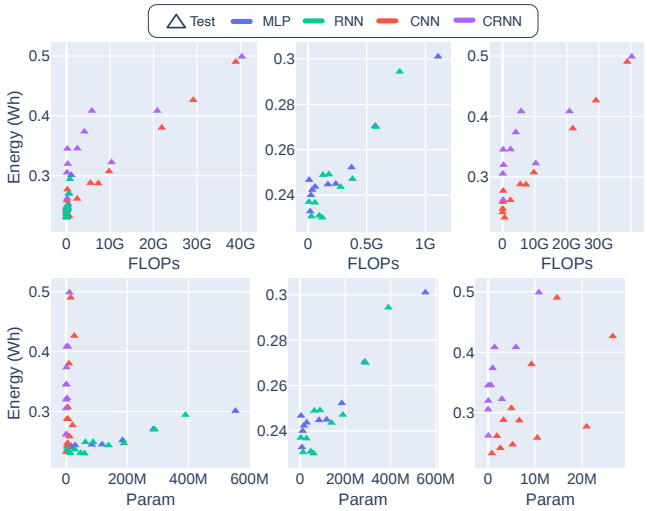

Figure 1: Energy consumption at test for various neural network architectures and configurations, as a function of FLOPs (top) and of parameters (bottom). The three columns show: (1) all architectures together, (2) only MLP/RNN (in blue and green), and (3) only CNN/CRNN (in red and purple).

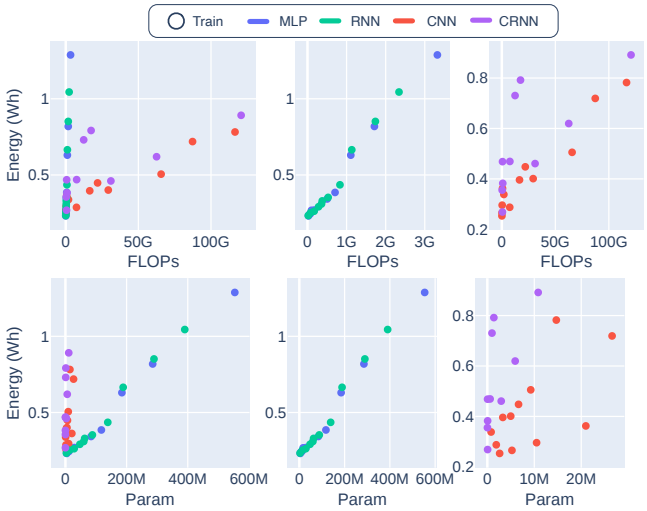

Figure 2: Energy consumption for training various neural network architectures and configurations, as a function of FLOPs (top) and of parameters (bottom). The three columns show: (1) all architectures together, (2) only MLP/RNN (in blue and green), and (3) only CNN/CRNN (in red and purple).

test phase on the same Nvidia T4 GPU for the entire dataset (corresponding to 1 epochs of training).

4. RESULTS

In this section, we explore the relationship between computational metrics and the energy consumption. Our analysis aims to identify trends and discrepancies in energy consumption at train and test across various architectures and configurations.

4.1. Relationship between energy and computational cost at test

First, we examine the energy consumption of the test, as existing research suggests that there is a correlation between FLOPs and energy consumption for convolutional models [19] on CPU. Figure 1 shows the result of this experiment, where the top row presents the GPU energy consumption as a function of FLOPs, and the bottom row the energy consumption as a function of the number of parameters. The first row shows that increasing the number of operations at test leads to an increase in energy consumption for all types of architecture. A closer examination of each architecture type reveals that the relationship between FLOPs and energy consumption exhibits some affine patterns. Examining the number of parameters in the second row, significant disparities emerge between MLP/RNN and CNN/CRNN models: the relationship between the number of parameters and the energy consumption is almost affine for MLP/RNN (and similar to the relationship with FLOPs), but for CNN and CRNN the relationship is more chaotic. This discrepancy is mainly due to the architectural elements composing these networks. Convolutional layers use parameter sharing, which contrasts with fully connected layers where each parameter is unique to its connection. Similarly, in recurrent layers, the connections between units often have unique weights, although some forms of parameter sharing can occur as well. Consequently, MLP and RNN have a higher number of parameters but a lower number of operations relative to CNN. These observations suggest that the number of operations and the number of parameters are not reliable indicators for estimating energy consumption at test, regardless of the model type, as the affine patterns are not consistent across architectures. However, they could be useful within a single architecture scenario comparisons.

4.2. Relationship between energy and computational cost at training

Building on our previous results, we now investigate the energy consumption associated with training. Figure 2 displays the energy consumption for training in function of the two computational metrics arranged as previously described. Regarding the interaction between energy and FLOPs, we observe two distinct trends. For MLP/RNN, the data points follow a steep curve on the left side, while for CNN, the curve smoothly increases and spans the entire plot. The CRNN architecture appears to exhibit characteristics that lie between the two aforementioned trends. In some configurations, the CRNN behaves as a CNN at higher FLOPs and as an RNN at lower FLOPs. A plausible explanation of this two trends could be the higher memory exchanges associated with MLP/RNN compared to CNN architectures that would cause higher energy consumption but do not increase the FLOPs. An important result is the almost affine relationship between FLOPs and energy consumption for MLP and RNN, suggesting that GPUs handle these architectures similarly during training causing close energy consumption for the same FLOPs. However, for CNN and CRNN, FLOPs alone do not provide a conclusive estimate of the energy consumption. Regarding the number of parameters, we conclude consistent results as for the test relationship. As a result, for the training consumption, neither FLOPs nor parameters are good estimators of energy consumption without specific knowledge of the model architecture, and one hypothesis could comes from the difference between the archi-

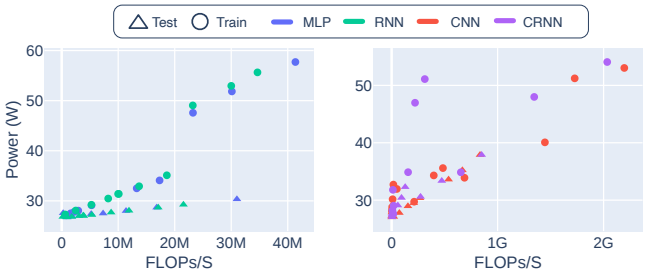

Figure 3: Average power during training (circles) and test (triangles) as a function of FLOPs/S.

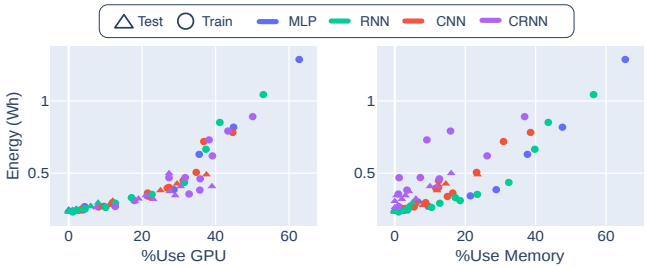

Figure 4: Relationship between the energy consumption and the GPU utilization (left) and memory utilization (right) for training and test.

tectural elements of the network.

4.3. Training and test comparisons

To further investigate the link between energy and computation, we investigate the mean average power at test and train and relate it to the number of floating points operations per seconds. The average is calculated as the energy divided by the length of the experiment. We present the result of this analysis in Figure 3, where the FLOPs/S is computed as the FLOPs divided by the duration of one epoch for training and test. We see that there is a nearly-affine relationship between FLOPs/S and power at test for the MLP/RNN architectures, as indicated by the aligned triangles. However, this affine relationship is less evident for training, as highlighted by a bend around 20M FLOPs/S. An significant result of this analysis is the disparity in average power consumption between MLP/RNN at train and test: circles are positioned higher on the plot, while triangles are lower and there is no overlap between the two sets. In contrast, for CNN and CRNN, triangles and circles occupy similar regions, indicating that MLP and RNN architectures require much more power for training than for testing compared to CNN/CRNN.

4.4. GPU and memory utilization

During our experiments, we also monitored the GPU and memory utilization given by Nvidia SMI. Figure 4 illustrates the relationship between the energy and the GPU and memory utilization during both training and test phases. Notably, a strong correlation exists between GPU use and energy. What is noteworthy is that this correlation remains independent of the phase (train or test) and the architectures. This results in a metric that is highly recommended for estimating the energy consumption of a given model, although it is dependent on the hardware. It would be interesting to find a combination of the number FLOPs and the number of parameters that could reflect the GPU utilization. For memory utilization, the correlation is not as straightforward, but it shows that memory also has an impact on energy consumption, with a higher dependency on the architecture type than GPU utilization.

5. DISCUSSION AND FUTURE WORKS

In this article, we specifically study the audio tagging task, using very simple architectures that are far from current SED models. It would therefore be interesting to explore more advanced models in the field and assess whether similar trends persist. In addition, the training procedure implemented here is one of the most conventional methods of deep learning, but recent advances have introduced much more complex procedures, resulting in higher computational costs and potentially different energy consumption. For example, using techniques such as teacher-student learning (used in the baseline) can lead to higher computational costs and therefore a different energy footprint. It is also important to note that energy consumption throughout our study is measured for a single epoch, and is therefore relative to the dataset. Experiments to determine whether there is a linear relation between data size and energy consumption would be recommended to remove the dependency on the dataset.

Additionally, we focused here on a single hardware (one Nvidia Tesla T4). However, analyzing the differences within a single hardware configuration and exploring the variations between different hardware configurations could provide some additional information on the energy consumption. This approach could also contribute to efforts to normalize hardware energy measurements, such as those proposed by Serizel et al. [17]. Furthermore, our study did not address the performance of the models. It's likely that a CNN and CRNN may have different performances compared to an MLP or an RNN. This concept aligns with Douwes et al. [29], emphasizing the need to explore multi-objective criteria by considering factors such as model performance, energy consumption, and computational efficiency simultaneously.

6. CONCLUSIONS

Our study provides a better understanding of the relationship between computational cost and energy consumption for various neural networks used in SED tasks. We observed that while the number of floating-point operations and the number of parameters influenced energy consumption, these metrics were not consistent predictors across all architectures. We identify distinct trends and discrepancies in energy consumption during both testing and training phases, with notable differences between MLP/RNN and CNN/CRNN models. Finally, we establish correlations between energy consumption and GPU utilization for both training and test phases, that could lay as a foundation for future research on computational indicators. We hope that this study will contribute to the development of green AI practices not only in speech processing but also across other domains.

7. REFERENCES

[1] J. W. Kim, S. W. Son, Y. Song, H. K. Kim, I. H. Song, and J. E. Lim, "Semi-supervsied learning-based sound event de-

tection using freuqency dynamic convolution with large kernel attention for dcase challenge 2023 task 4," *arXiv preprint arXiv:2306.06461*, 2023.

[2] D. Berghi, P. Wu, J. Zhao, W. Wang, and P. J. Jackson, "Fusion of audio and visual embeddings for sound event localization and detection," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 8816–8820.

[3] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12 449–12 460, 2020.

[4] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[5] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[6] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *arXiv preprint arXiv:2111.05011*, 2021.

[7] D. Amodei, D. Hernandez, G. Sastry, J. Clark, G. Brockman, and I. Sutskever, "Ai and compute," 2018.

[8] J. Sevilla, L. Heim, A. Ho, T. Besiroglu, M. Hobbhahn, and P. Villalobos, "Compute trends across three eras of machine learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[9] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning (2020)," *arXiv preprint arXiv:2007.05558*, 2007.

[10] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[11] A. S. Luccioni, S. Viguier, and A.-L. Ligozat, "Estimating the carbon footprint of bloom, a 176b parameter language model," *Journal of Machine Learning Research*, vol. 24, no. 253, pp. 1–15, 2023.

[12] United Nations, "Paris Agreement," 2015.

[13] C. Douwes, G. Bindi, A. Caillon, P. Esling, and J.-P. Briot, "Is quality enough: Integrating energy consumption in a large-scale evaluation of neural audio synthesis models," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[14] T. Parcollet and M. Ravanelli, "The energy and carbon footprint of training end-to-end speech recognizers," 2021.

[15] 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," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 10 039–10 081, 2020.

[16] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[17] R. Serizel, S. Cornell, and N. Turpault, "Performance above all? energy consumption vs. performance, a study on sound event detection with heterogeneous data," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[18] F. Ronchini and R. Serizel, "Performance and energy balance: a comprehensive study of state-of-the-art sound event detection systems," *arXiv preparint arXiv:2310.03455*, 2023.

[19] D. T. Speckhard, K. Misiunas, S. Perel, T. Zhu, S. Carlile, and M. Slaney, "Neural architecture search for energy efficient always-on audio models," *arXiv preprint arXiv:2202.05397*, 2022.

[20] C.-J. Wu, R. Raghavendra, U. Gupta, B. Acun, N. Ardalani, K. Maeng, G. Chang, F. Aga, J. Huang, C. Bai, *et al.*, "Sustainable ai: Environmental implications, challenges and opportunities," *Proceedings of Machine Learning and Systems*, vol. 4, pp. 795–813, 2022.

[21] F. Ronchini, S. Cornell, R. Serizel, N. Turpault, E. Fonseca, and D. P. Ellis, "Description and analysis of novelties introduced in dcase task 4 2022 on the baseline system," *arXiv preprint arXiv:2210.07856*, 2022.

[22] F. Ronchini, J. Ebbers, F. Angulo, D. Perera, S. Essid, and R. Serizel, "DCASE 2023 Task 4a Challenge," https://dcase.community/challenge2023, 2023.

[23] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, "Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 3505–3506.

[24] M. Hobbhahn and J. Sevilla, "What's the backward-forward flop ratio for neural networks?" 2021, accessed: 2024-03-01. [Online]. Available: https://epochai.org/blog/backward-forward-FLOP-ratio

[25] M. Jay, V. Ostapenco, L. Lefèvre, D. Trystram, A.-C. Orgerie, and B. Fichel, "An experimental comparison of software-based power meters: focus on cpu and gpu," in *CCGrid 2023-23rd IEEE/ACM international symposium on cluster, cloud and internet computing*. IEEE, 2023, pp. 1–13.

[26] V. Schmidt, K. Goyal, A. Joshi, B. Feld, L. Conell, N. Laskaris, D. Blank, J. Wilson, S. Friedler, and S. Luccioni, "Codecarbon: estimate and track carbon emissions from machine learning computing (2021)," *DOI: https://doi.org/10.5281/zenodo*, vol. 4658424, 2021.

[27] N. Turpault, R. Serizel, A. P. Shah, and J. Salamon, "Sound event detection in domestic environments with weakly labeled data and soundscape synthesis," in *Workshop on Detection and Classification of Acoustic Scenes and Events*, 2019.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] C. Douwes, P. Esling, and J.-P. Briot, "Energy consumption of deep generative audio models," *arXiv preprint arXiv:2107.02621*, 2021.