Normalizing Energy Consumption for Hardware-Independent Evaluation

Constance Douwes, Romain Serizel

2024 IEEE INTERNATIONAL WORKSHOP ON MACHINE LEARNING FOR SIGNAL PROCESSING, SEPT. 22–25, 2024, LONDON, UK

NORMALIZING ENERGY CONSUMPTION FOR HARDWARE-INDEPENDENT EVALUATION

Constance Douwes, Romain Serizel

Université de Lorraine, CNRS, Inria, Loria, Nancy, France

ABSTRACT

The increasing use of machine learning (ML) models in signal processing has raised concerns about their environmental impact, particularly during resource-intensive training phases. In this study, we present a novel methodology for normalizing energy consumption across different hardware platforms to facilitate fair and consistent comparisons. We evaluate different normalization strategies by measuring the energy used to train different ML architectures on different GPUs, focusing on audio tagging tasks. Our approach shows that the number of reference points, the type of regression and the inclusion of computational metrics significantly influences the normalization process. We find that the appropriate selection of two reference points provides robust normalization, while incorporating the number of floating-point operations and parameters improves the accuracy of energy consumption predictions. By supporting more accurate energy consumption evaluation, our methodology promotes the development of environmentally sustainable ML practices.

***Index Terms*—** Machine learning, energy consumption, normalization, GPU, FLOPs, signal processing.

1. INTRODUCTION

In the field of signal processing, deep learning (DL) has seen a widespread adoption with applications in various domains, such as music generation [1], speech recognition [2] and sound event detection (SED) [3]. However, its growing popularity has drawn attention to a major problem : the significant amount of compute associated with learning and running deep learning models [4, 5, 6]. Notably, in the field of natural language processing (NLP), Strubell et al. [7] have highlighted the colossal environmental and financial impact of training such large language models. This research contributes to an emerging field known as Green AI [8], which seeks to measure and mitigate the energy consumption of AI technologies, with new tools for estimating both energy consumption and carbon emissions [9, 10, 11].

Even though models used in audio processing are smaller than those used in NLP, they still present similar problems [12, 13], and efforts have been made balance energy efficiency with performance. For example, Douwes et al. [14] performed a detailed analysis of generative models for audio synthesis, exploring the trade-offs between energy consumption and audio quality. In speech recognition, Parcollet and Ravanelli [13] showed that small performance improvements often have extremely high energy costs. Similarly, Serizel et al. [15] investigated the relationships between energy consumption, training time, GPU type and performance when training SED systems. In addition, Ronchini et al. [16] balanced performance and energy based on an energy-weighted metric [17] to evaluate SED systems across different hardware types. This issue is particularly critical in the context of cross-hardware comparison, where a consistent and comparable measurement of energy consumption is required. While it is common practice in research to test a system's performance and compare it with others, comparing energy consumption presents significant challenges due to the heavy dependence on the hardware used. Even if systems perform similarly, they may use different amounts of energy depending on the hardware they are trained on. Therefore, it is imperative to develop methods for normalizing energy metrics, ensuring independence from hardware and isolating the influence of the model's energy consumption. For example, from 2022, the DCASE Challenge Task 4 [17] requests participants to report their energy consumption along with a reference consumption in order to normalize energy consumption. However, no experimental analysis of this normalization strategy has been conducted to prove its effectiveness.

A potential alternative approach to assessing energy efficiency is to use computational metrics, such as the number of floating-point operations (FLOPs), as proxies. This method has been explored by Asperti et al. [18], where they estimated the energy consumed by convolutional networks during inference on GPU using a slightly modified computation of FLOPs. However, their study was limited to CNN architectures, which raises questions about the applicability of this metric to other neural network architectures. In particular, Ronchini et al. [16] showed that the relationship between FLOPs and energy consumption is not straightforward without prior knowledge of the network. Another approach to energy estimation is to predict per-layer energy consumption given layer characteristics, as done by Getzner et al. [19].

The total consumption of the network is then the sum of all the individual predicted energy contributions of the layers. Again, detailed knowledge of the architecture is required to accurately estimate the total consumption. Although these methods give promising results, they are heavily dependent on specific equipment and do not address training. Yet, as ML audio researchers, the majority of our energy consumption lies in the training phase and should not be neglected.

In this paper, we focus on developing a methodology to compare the energy consumption of the training processes independently of the hardware used, with minimal information on the neural network architecture. As an initial study, we measure the energy consumed during the training of four neural network architectures (MLP, CNN, RNN, CRNN) designed for audio tagging tasks on four different GPUs. We first study the influence of the number of reference point to normalize the energy consumption. We show that by strategically choosing those points, a strong linear regression fit can be achieved. We then explore different regression models to determine which one best fits the energy consumption on two hardware. We consider linear, polynomial and support vector regression (SVR). Our results show that the most appropriate regression model varies depending on the specific hardware pair. Finally, we integrate computational metrics such as the number of FLOPs and parameters to the linear regression. This hybrid approach provides robust predictions that adapts to diverse hardware configurations.

2. EXPERIMENTAL SETUP

Our goal is to normalize the energy consumption for training neural network models on different hardware. As a proof of concept, we work on an audio tagging task, and record the energy for training different types and sizes of architectures.[1]

Task description. Audio tagging consists in assigning tags (one or many) to an audio signal without any additional temporal information. In this experiment, we use the DESED dataset [20], a well-known resource in the sound event detection community. This dataset consists of 10-second audio clips containing sound from domestic environments. We focus on the subset of real recordings, which provide a reliable 10-class annotation. We convert these recordings into mel-spectrogram representations using 128 bands, with an FFT size of 2048 and a hop size of 256. We use the first 64 frames as input, which approximately corresponds to taking the first 1 second of the audio signal. This drastically changes the performance of the model but also reduces the complexity of the system and allows for lighter 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.

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 combine both CNN and RNN. 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.

Training. Our experiments differ from the traditional search for accuracy measures. Instead, we focus our analysis on the energy consumption associated with training neural networks. To do this, we trained all models for 10 epochs on four different GPU types. The specifications for each GPU are summarized in Table 2 and are referred to throughout the paper by their respective names: RTX, GTX, T4 and A40. We also include the Thermal Design Power (TDP) of each GPU, which indicates the maximum heat generated at peak load, providing an insight into the power requirements of each unit. We chose a common batch size of 8, use the cross-entropy function as the criterion, and set the learning rate at 10^{-3} with an ADAM optimiser [21]. We deliberately omit any validation steps in the training routine in order to attribute the energy measurements only to the training process.

Hardware	TDP
NVIDIA GeForce RTX 2080 Ti (11 GiB)	250 W
NVIDIA GeForce GTX 1080 Ti (11 GiB)	250 W
NVIDIA Tesla T4 (15GiB)	70 W
NVIDIA A40 (45 GiB)	300 W

Table 2: Specifications of the GPUs used in the study, including each unit's Thermal Design Power (TDP).

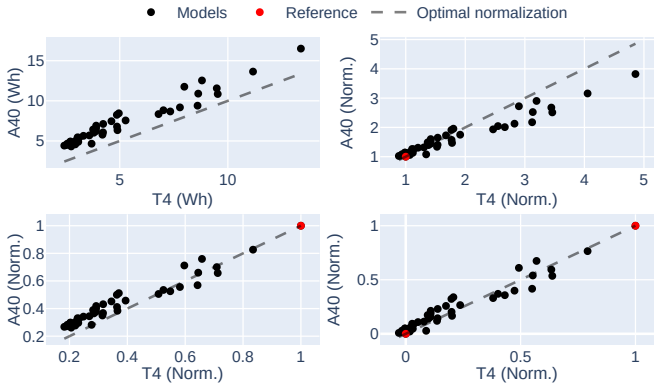


Fig. 1: Normalization of energy consumption using different reference points for the T4-A40 GPU pair.

Energy and computational costs. We monitor the energy consumption for training using CodeCarbon [9], which provides detailed energy consumption for the three components of the computing system: GPU, CPU and RAM. We focus exclusively on the GPU consumption, as this is the primary energy drain in our experiments. We also compute the number of FLOPs using the deepspeed profiler [22] to quantify the number of forward pass operations accurately. We also report the number of parameters of each configuration.

3. RESULTS

We start with a study of the normalization strategy proposed by Ronchini et al. [16] and extend to a quantitative analysis of the influence of the number of reference points chosen for the regression used as normalization, followed by a study of the influence of the type of regression, and the inclusion of computational metrics to the regression.

3.1. Influence of the number of reference points

We explore here the normalization strategy from Ronchini et al. [16], which involves selecting a reference model and recording its energy consumption on each GPU. This approach assumes that the ratio of energy consumption between any given model and the reference model should remain consistent across different GPU types. Figures 1 and 2 show the results of this normalization experiment for T4-A40 and T4-RTX GPU pairs respectively. The top left plots show the

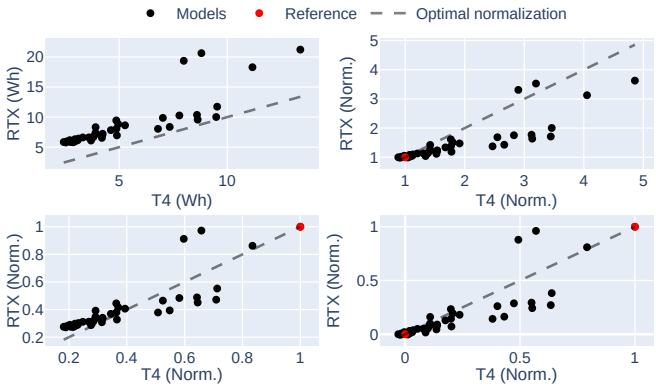
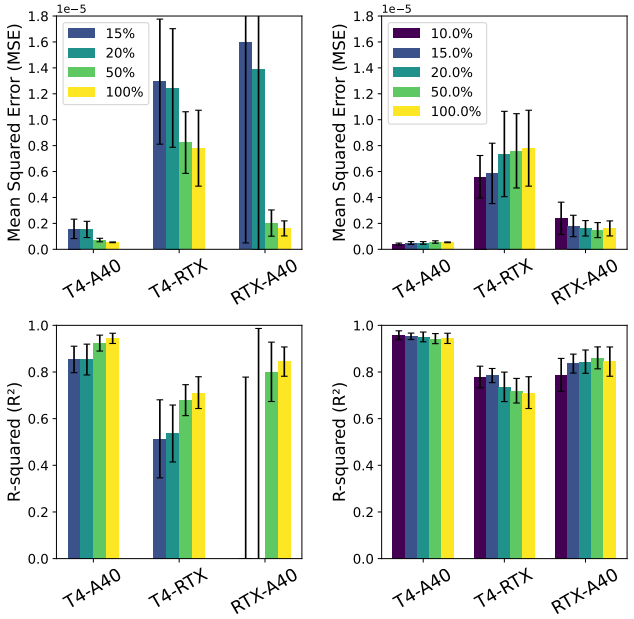


Fig. 2: Normalization of energy consumption using different reference points for the T4-RTX GPU pair.

energy consumption without any normalization. On the top right the graphs show the normalization using a low energy model as the reference, and on the bottom left a high energy model as the reference. The last graphs at the bottom right present our dual reference approach, where both a low and a high energy model are used as reference points. The closer the results are to the optimum line (in dashed), the more accurate the normalization is. We can see that the normalization is significantly affected by the choice of the reference model. Using a low-energy model as the reference effectively normalizes the energy consumption of other low-energy models but reduces normalization accuracy for high-energy models, and vice versa. Our strategy, which uses dual reference points, achieves a closer approximation to the ideal linear relationship between T4-A40. However, for the T4-RTX pair, while the dual-reference approach corrects the inclination of the normalization for low and high models, the linear regression still appears non-optimal. Note that this dual approach is a two-point regression.

To quantify the impact of increasing the number of reference points, we divide our dataset of 43 models into two subsets: 80% (34 models) for training and 20% (9 models) for testing. This division is randomly generated five times to perform cross-validation. At each iteration, we select a range from 10%, 15%, 20%, 50%, and 100% of the train data to be used as reference points, corresponding to 2, 4, 6, 16, and 32 models respectively. We explore two methods of subsampling reference models: a random selection of models and a min-max strategy, where we select half of the models with the lowest and half with the highest energy consumption, 10% corresponding to our dual reference point approach. We assess the quality of the predictions by computing the coefficient of determination (R^2) and the mean squared error (MSE).

Results are presented in Figure 3, where the left bar graph shows outcomes from random sampling (3a) and the right from min-max sampling (3b). We did not include the results of random sampling at 10% due to its highly imprecise and



(a) Random sampling (b) Min-max sampling

Fig. 3: Impact of reference point selection on the linear regression between GPU pairs using different data sampling strategies.

Fig. 4: Illustration of different regression types to model the energy consumption of the RTX-GTX hardware pair.

poor normalization performance. Across all pairs, we see that increasing the percentage of the train set used as reference points for random sampling improves the R^2 and significantly reduces the MSE. For the T4-RTX pair, lower percentages such as 15% and 20% even result in negative R^2 values (not represented for more clarity) highlighting the ineffectiveness of random sampling at smaller sample sizes. Furthermore, while random sampling generally requires the entire train set to achieve accurate regression, a careful selection of just two points can be a viable strategy. Specifically, choosing two models at the extremes notably improves the results for the T4-RTX pair, outperforming the outcomes obtained using the full train set.

Limitations. The overall precision of the regression remains moderate for the T4-RTX pair. This suggests that the relationship between the energy consumption of the different devices may not be strictly linear and that further experiments are needed to explore alternative models that might better capture this relationship.

3.2. Influence of the regression type

In this section we explore different types of regression to model energy consumption across different hardware pairs. We compare linear, polynomial (degree 2) and support vector regression using 100% of the training set. For the SVR, we set the regularisation parameter $C = 0.1$ and the loss tolerance $\epsilon = 0.0001$ after a preliminary grid search. An illustrative example of these regressions applied to a realisation of the train-test split is given in Figure 4 for the RTX-GTX pair. We observe that increasing the complexity of the regression model tends to improve the fit to the training data points. Quantitative results of these observations are presented in Figure 5. For the T4-A40 pair, linear regression provides the best fit, achieving the highest R^2 and lowest MSE, indicating strong predictive accuracy. In contrast, polynomial regression outperforms linear regression for all other pairs, although with less confidence. SVR appears to be effective for the RTX-GTX pair but performs poorly for other pairings.

Limitations. The performance of each regression model varies significantly depending on the hardware, making unified normalization across different types of hardware difficult. A preference for the linear regression emerges due to its computational efficiency and its generally robust performance and reliability in the majority of pairs. However, the variability observed in its performance across different hardware pairs suggests that focusing solely on energy might not fully capture the underlying dynamics of the energy consumption.

3.3. Influence of computational metrics

In this section we look at the effect of the incorporation of computational factors into linear regression models. Rather than examining the relationship between energy consumption and a 2D space, where each dimension represents the

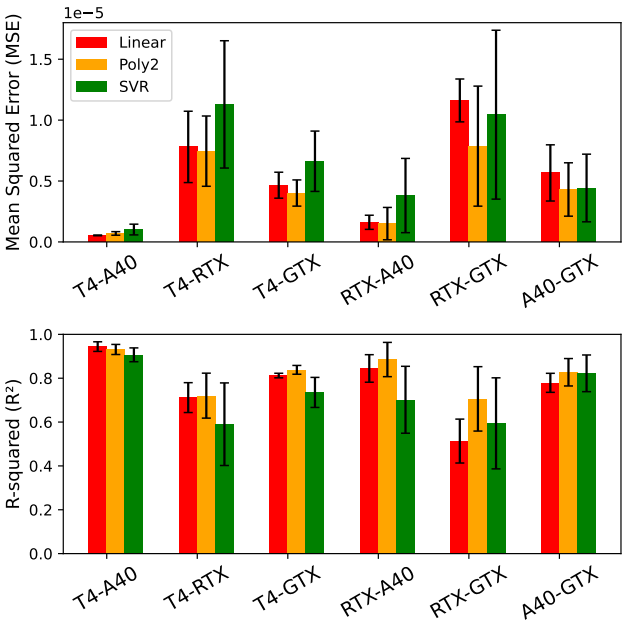


Fig. 5: Comparison of regression types among linear, polynomial and support vector regression for normalizing energy consumption across different hardware.

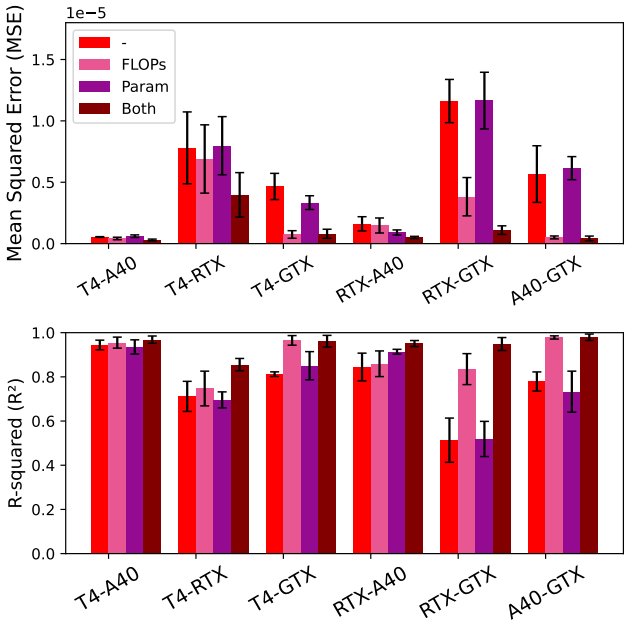


Fig. 6: Comparison of linear regression models integrating additional computational metrics—FLOPs, number of parameters, and a combination of both—across various hardware pairs.

energy consumption of one hardware, we propose to extend the regression space by incorporating a third (and a forth) dimension, which is related to the architectural elements of the models. Therefore, we include the number of FLOPs, then the number of parameters, and finally both in the regression. Figure 7 illustrates the linear regression when adding the FLOPs to the energy consumption for the RTX-GTX pair. We observe that the test data are closer to the regression surface compared to the 2D linear regression shown in Figure 4. This improvement suggests that incorporating FLOPs captures some of the nonlinear relationships from the 2D representations. Quantitative results of this analysis are presented in Figure 6. Across all hardware pairs, significant improvements in regression performance are noted when FLOPs are included along with energy. The inclusion of the number of parameters alone generally leads to decreased accuracy. Yet, combining both FLOPs and parameters yields even better results than using FLOPs alone for most hardware pairs.

Limitations. To normalize energy consumption using a three-dimensional approach with FLOPs, at least three reference points are required. When parameters are added, this requirement increases to four reference points, further complicating the normalization process and model selection.

4. CONCLUSION AND FUTURE WORKS

This paper presents several methods for normalizing the energy consumption of machine learning models across various types of hardware. We show that the choice of reference points has a critical impact on the normalization process. For example, the use of dual reference points provides a more balanced and accurate approach suitable for different hardware setups than using only one reference. Furthermore, we find that the integration of computational metrics such as the number of FLOPs and parameters improves the normalization strategy, but also introduces additional complexity.

Future work should address the complexity introduced by the inclusion of additional computational factors. In particular, efforts should focus on optimizing the balance between accuracy and practical applicability in real-world scenarios, ensuring that the normalization strategies developed are both effective and feasible for widespread use. In addition, it is crucial to extend the research to a wider range of machine learning applications, as well as different hardware configurations, neural network architectures and training routines.

5. REFERENCES

[1] Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al.,

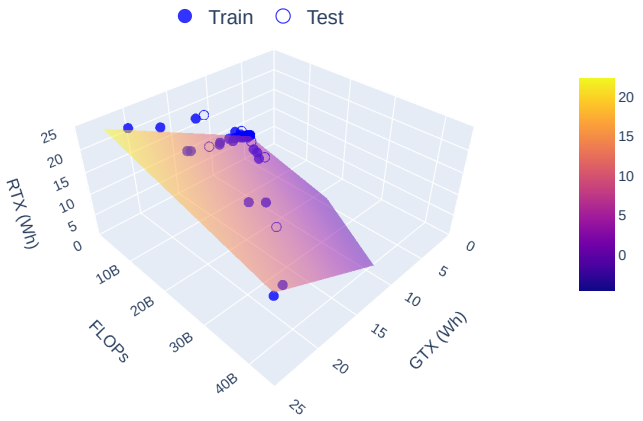


Fig. 7: Illustration of a 3D linear regression model incorporating FLOPs alongside energy consumption data for the RTX-GTX hardware pair.

"Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[2] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28492–28518.

[3] Kang Li, Yan Song, Li-Rong Dai, Ian McLoughlin, Xin Fang, and Lin Liu, "Ast-sed: An effective sound event detection method based on audio spectrogram transformer," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[4] Dario Amodei, Danny Hernandez, Girish Sastry, Jack Clark, Greg Brockman, and Ilya Sutskever, "Ai and compute," 2018.

[5] Jaime Sevilla, Lennart Heim, Anson Ho, Tamay Besiroglu, Marius Hobbhahn, and Pablo Villalobos, "Compute trends across three eras of machine learning," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[6] Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso, "The computational limits of deep learning (2020)," *arXiv preprint arXiv:2007.05558*, 2007.

[7] Emma Strubell, Ananya Ganesh, and Andrew McCallum, "Energy and policy considerations for deep learning in nlp," *arXiv preprint arXiv:1906.02243*, 2019.

[8] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni, "Green ai," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[9] Victor Schmidt, Kamal Goyal, Aditya Joshi, Boris Feld, Liam Conell, Nikolas Laskaris, Doug Blank, Jonathan Wilson, Sorelle Friedler, and Sasha Luccioni, "Codecarbon: estimate and track carbon emissions from machine learning computing (2021)," *DOI: https://doi. org/10.5281/zenodo*, vol. 4658424, 2021.

[10] Lasse F. Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan, "Carbontracker: Tracking and predicting the carbon footprint of training deep learning models," ICML Workshop on Challenges in Deploying and monitoring Machine Learning Systems, July 2020, arXiv:2007.03051.

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

[12] Constance Douwes, Philippe Esling, and Jean-Pierre Briot, "Energy consumption of deep generative audio models," *arXiv preprint arXiv:2107.02621*, 2021.

[13] Titouan Parcollet and Mirco Ravanelli, "The energy and carbon footprint of training end-to-end speech recognizers," 2021.

[14] Constance Douwes, Giovanni Bindi, Antoine Caillon, Philippe Esling, and Jean-Pierre Briot, "Is quality enough*f* 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.

[15] Romain Serizel, Samuele Cornell, and Nicolas 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.

[16] Francesca Ronchini and Romain Serizel, "Performance and energy balance: a comprehensive study of state-of-the-art sound event detection systems," *arXiv preprint arXiv:2310.03455*, 2023.

[17] Francesca Ronchini, Samuele Cornell, Romain Serizel, Nicolas Turpault, Eduardo Fonseca, and Daniel PW Ellis, "Description and analysis of novelties introduced in dcase task 4 2022 on the baseline system," *arXiv preprint arXiv:2210.07856*, 2022.

[18] Andrea Asperti, Davide Evangelista, and Moreno Marzolla, "Dissecting flops along input dimensions for greenai cost estimations," in *International Conference on Machine Learning, Optimization, and Data Science*. Springer, 2021, pp. 86–100.

[19] Johannes Getzner, Bertrand Charpentier, and Stephan Günnemann, "Accuracy is not the only metric that matters: Estimating the energy consumption of deep learning models," *arXiv preprint arXiv:2304.00897*, 2023.

[20] Nicolas Turpault, Romain Serizel, Ankit Parag Shah, and Justin 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.

[21] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[22] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong 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.