Monitoring environmental impact of DCASE systems: Why and how ?

Constance Douwes, Francesca Ronchini , Romain Serizel

September 2023, Tampere, Finland

DCASE2023 WORKSHOP

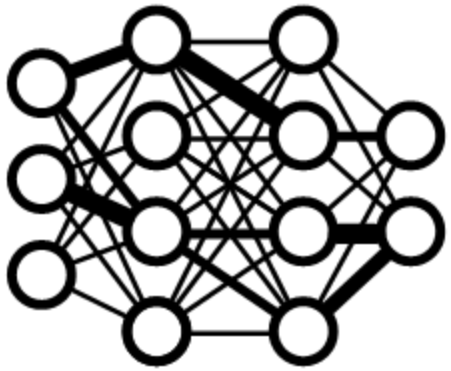
Introduction

Romain Serizel

Monitoring environmental impact: Why?

- What is the footprint of our systems?
- What is the cost of performance improvement?

Monitoring environmental impact: How?



- Which metrics?
- Are they reliable?
- How to relate with performance?

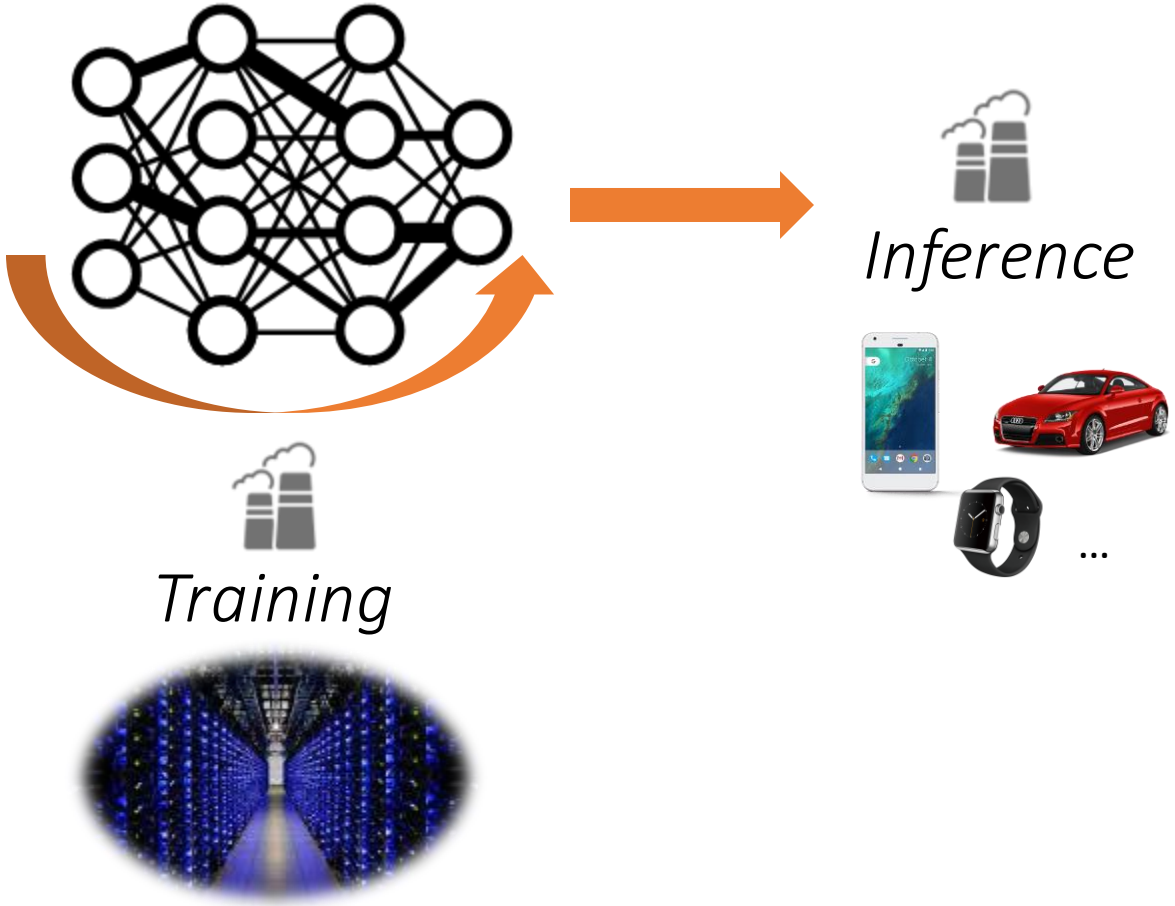
Outline

Part 1	Monitoring our footprint • Introduction to metrics and librairies • Estimating footprint and performance	(Constance Douwes)
Interlude	How can we implement this at community level?	(You)
Part 2	Case study on DCASE task 4 • Setup and metrics • Systems comparison	(Francesca Ronchini)
Part 3	Hands on tutorial	(Constance & Francesca)

Part 1 : Monitoring our footprint

Constance Douwes

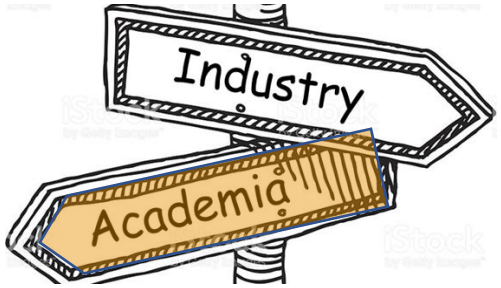
What is the footprint of our systems ?



What is the footprint of our systems ?

Inference

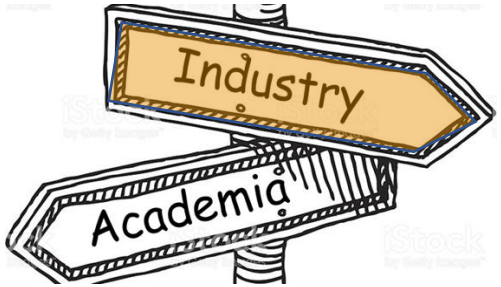
Training



What is the footprint of our systems ?

Training

Inference



Our goal

Understand and control the carbon footprint of our systems

- Comparative study of different metrics
- Benchmark of toolboxes and packages
- Trade-off between footprint and accuracy

- Study of the full life-cycle of our systems
- Hardware production

Comparative study of metrics

Comparative study of metrics

Runtime

- Straightforward method in every developing environment
- Highly dependent of the model's implementation
- Number & performance of GPU

Comparative study of metrics

Runtime

Number of parameters

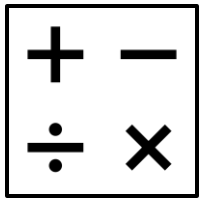
- Correlated with computational complexity
- Support from most DL libraries
- Different operations costs

Comparative study of metrics

Runtime

Number of parameters

Number of operations



- Hardware independent
- No trivial computation
- Closer to the energy footprint

Comparative study of metrics

Runtime

Number of parameters

Number of operations

Energy consumption

- Good indicator of the footprint
- Other jobs running
- Target a particular device

Comparative study of metrics

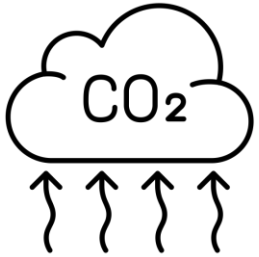
Runtime

Number of parameters

Number of operations

Energy consumption

Carbon emissions



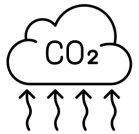
- Direct link with energy consumption
- Real carbon footprint impact
- Depends on local electricity infrastructure

Comparative study of metrics

Number of operations

Energy consumption

Carbon emissions



Number of operations

Sum of any mathematical operations (such as +, -, *, /) across all layers.

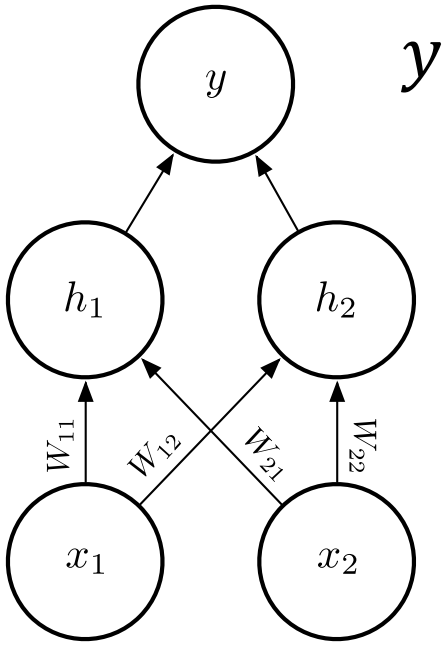
$$y = \underbrace{\mathrm{a} \cdot \mathrm{x} + b}$$

2 Floating-point Operations (2 Flops)

1 Multiply-Accumulate Operation (1 MAC)

Number of operations

Sum of any mathematical operations (such as +, -, *, /) across all layers.

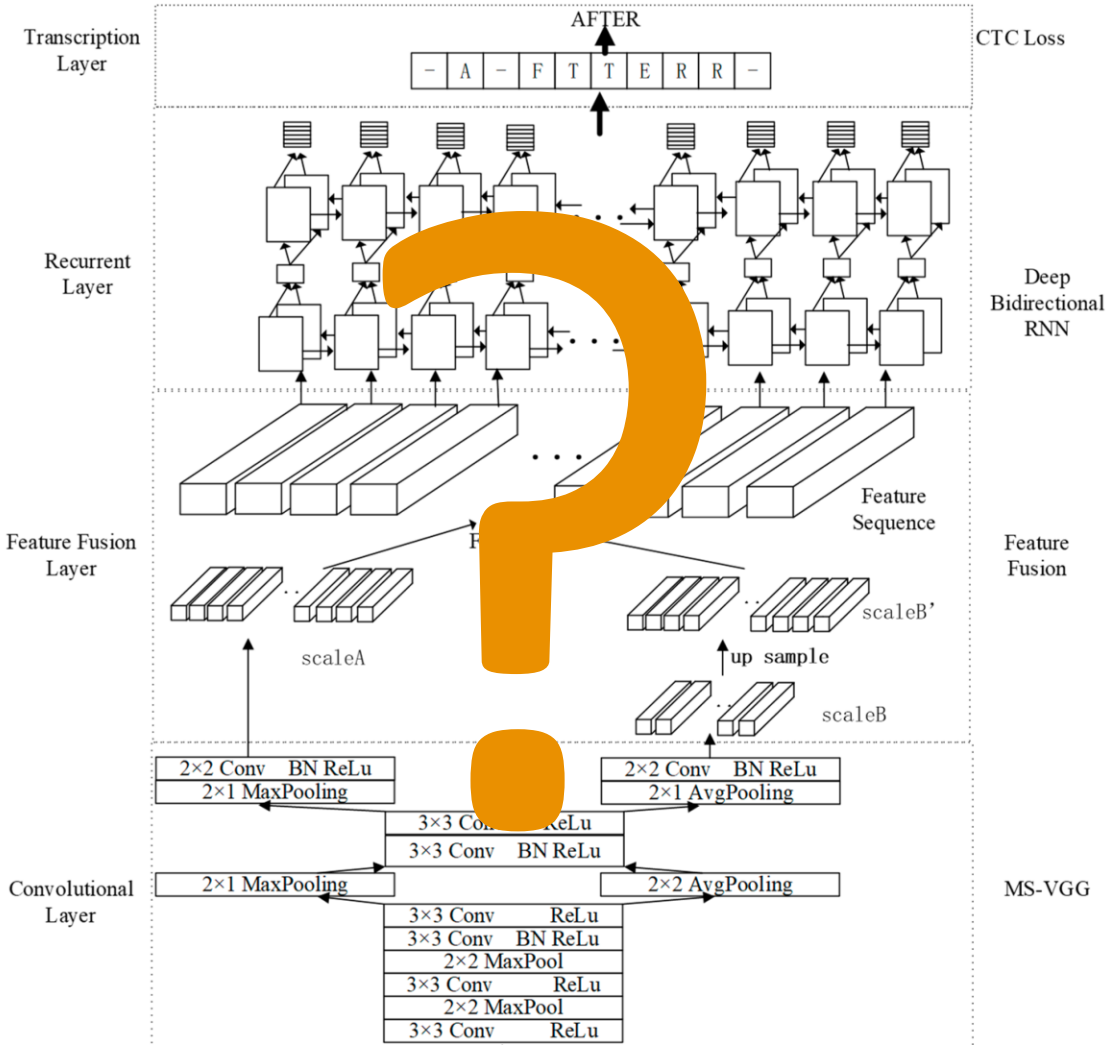


$$y = \underbrace{x_1 \cdot W_{11} + x_1 \cdot W_{12} + x_2 \cdot W_{21} + x_2 \cdot W_{22} + b}$$

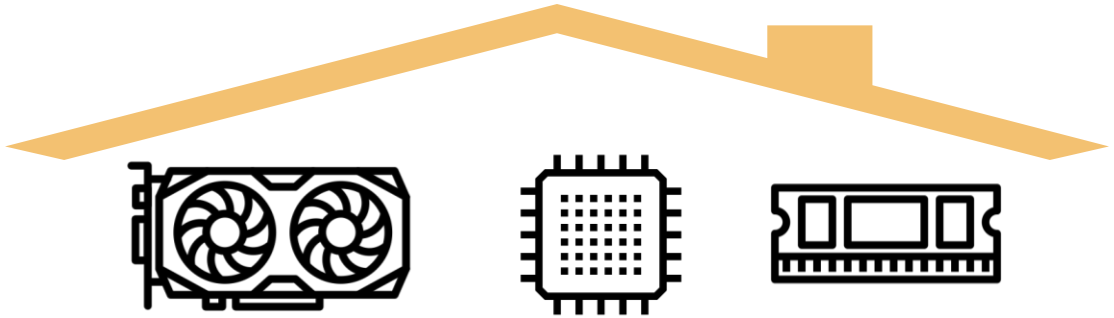
8 Flops

4 MACs

Number of operations



Energy consumption



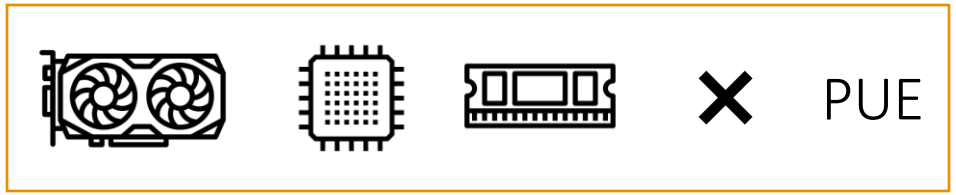
GPU + CPU + RAM

× PUE

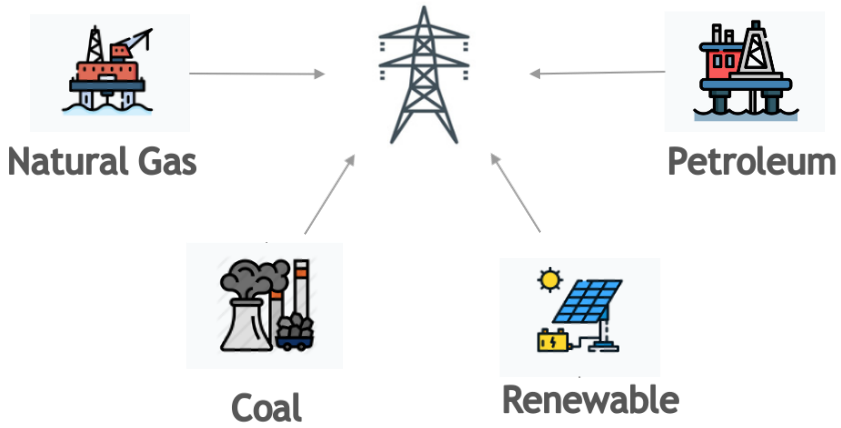
Power Usage Effectiveness
(Mainly Cooling)

Energy efficiency : Using less energy to accomplish the same task

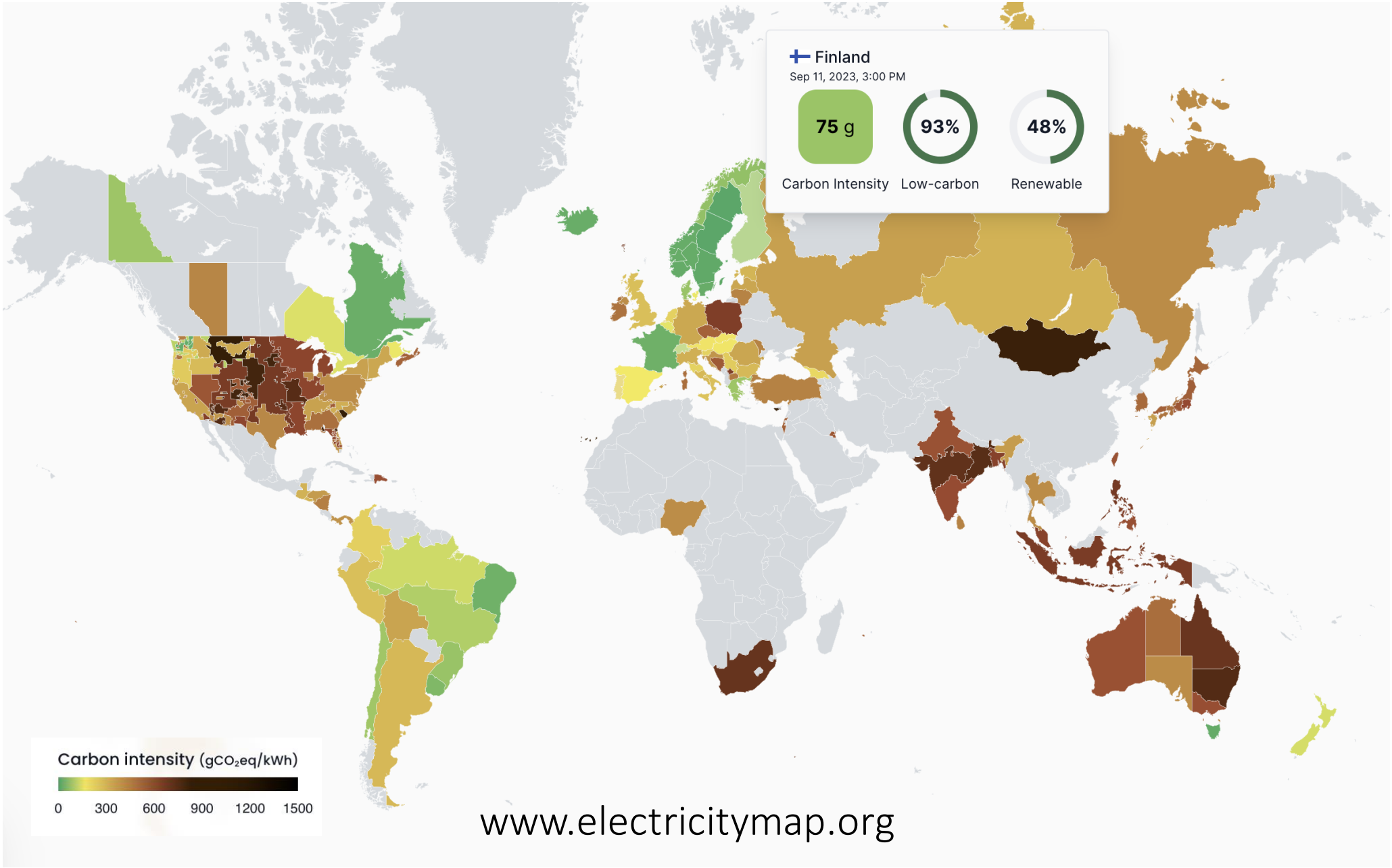
Carbon emissions



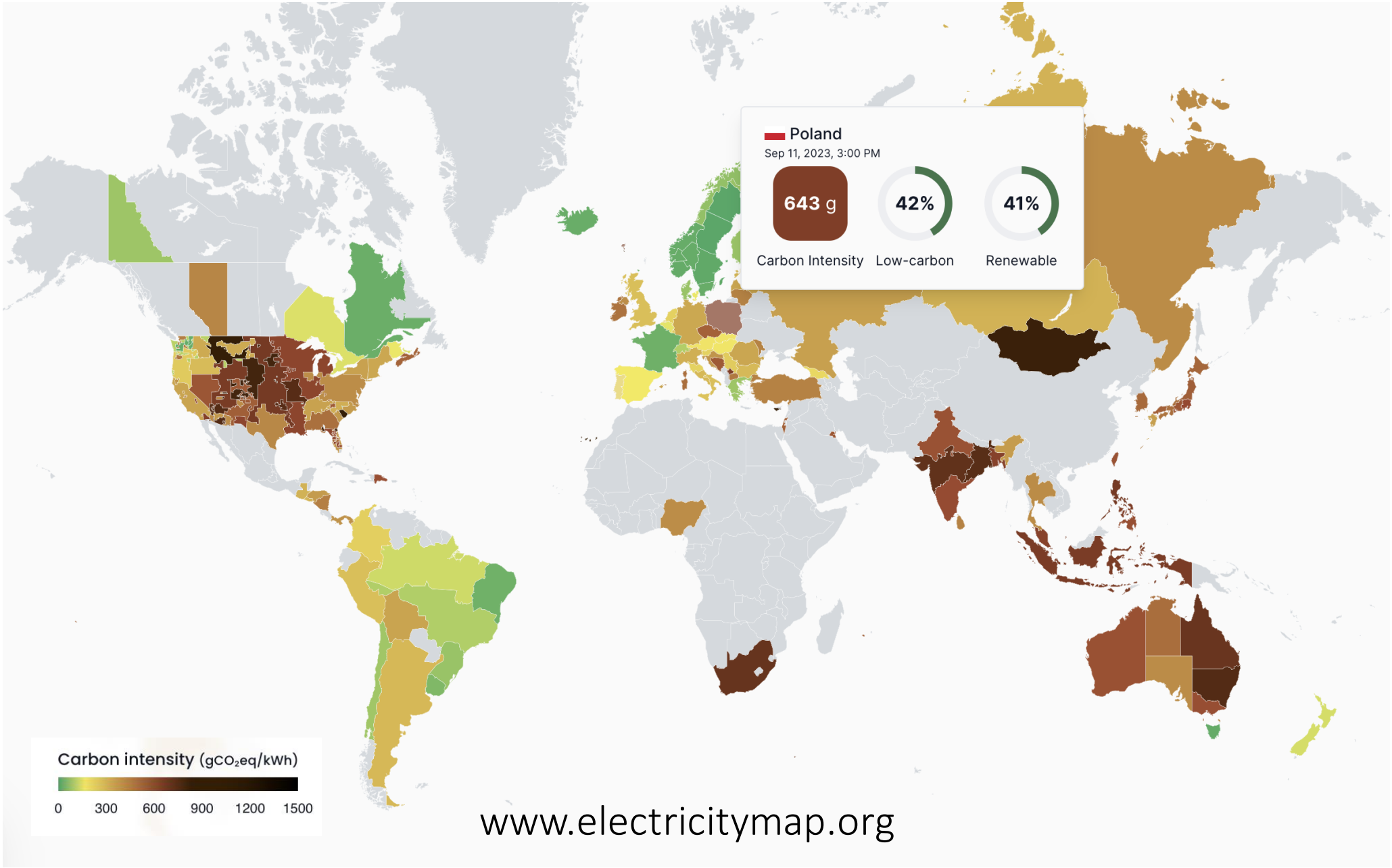
Energy consumption



Leads to unfair comparisons



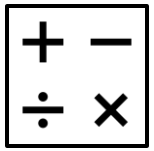
www.electricitymap.org



www.electricitymap.org

Toolboxes

Toolboxes



Lyken17/**pytorch-OpCounter**

Count the MACs / FLOPs of your PyTorch model.

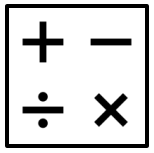
```
from thop import profile

input = torch.randn(1,28,28)

macs, params = profile(model,inputs=(input))
```

```
MACS: 205.563M PARAMS: 112.394K
```

User-friendly implementation

Only MACs computed

Toolboxes



deepspeed

```
import torch
from deepspeed.profiling.flops_profiler import get_model_profile

flops, macs, params = get_model_profile(model=model,
                                        input_shape=(1, 28, 28))
```

```
params per gpu:                                 112.39 k
fwd MACs per GPU:                               182.95 MMACs
fwd flops per GPU:                              389.48 M
fwd latency:                                    124.8 ms
fwd FLOPS per GPU = fwd flops/fwd latency:      3.12 GFLOPS
```

Lots of outputs

Compatibility problems with torch versions

Toolboxes

PyJoules

```
from pyJoules.energy_meter import measure_energy

@measure_energy
def evaluate():
    # Instructions to be evaluated.

evaluate()
```

```
begin timestamp :          1694544016.3633583;
tag :                      evaluate;
duration :                 0.1718430519104004;
nvidia_gpu_0 :             1908
```

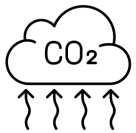
✓ Fast implementation

✗ Supports only Intel RAPL and Nvidia NVML hardware

Toolboxes

lfwa/**carbontracker**

Track and predict the energy consumption and carbon footprint of training deep learning models.



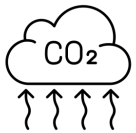

```python
from carbontracker.tracker import CarbonTracker

max_epochs = 200
tracker = CarbonTracker(epochs=max_epochs)

# Training loop.
for epoch in range(max_epochs):
    tracker.epoch_start()

    # your training loop
    pass

    tracker.epoch_end()

tracker.stop()
```

- ✓ Predict the energy consumed by the whole training from the training of one epoch
- ✗ Supports only Intel RAPL and Nvidia NVML hardware

```
Actual consumption for 1 epoch(s):  Predicted consumption for 1000 epoch(s):
        Time:    0:00:10                     Time:    2:52:22
        Energy:  0.000038 kWh                Energy:  0.038168 kWh
        CO2eq:   0.003130 g                  CO2eq:   4.096665 g
        This is equivalent to:               This is equivalent to:
        0.000026 km travelled by car         0.034025 km travelled by car
```

Toolboxes

CODE CARBON

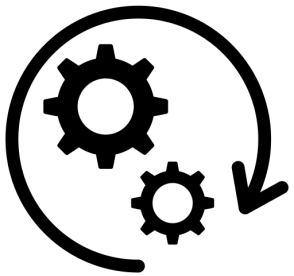



```python
from codecarbon import EmissionsTracker

tracker = EmissionsTracker()

tracker.start()

# Your training loop

emissions = tracker.stop()
```

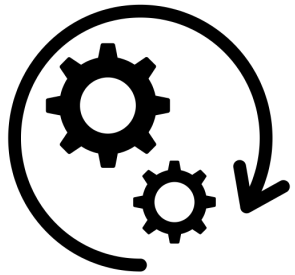
```
Energy consumed for RAM :        0.000000 kWh.
RAM Power :                      4.754392147064209 W
Energy consumed for all GPUs :   0.000001 kWh.
Total GPU Power :                18.936634403737294 W
Energy consumed for all CPUs :   0.000002 kWh.
Total CPU Power :                42.5 W
0.000003 kWh of electricity used since the beginning.
```

Works with most hardware

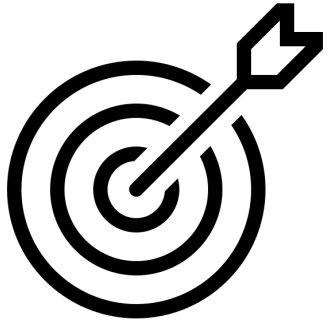
Outputs the emissions at the end of training only



Efficiency



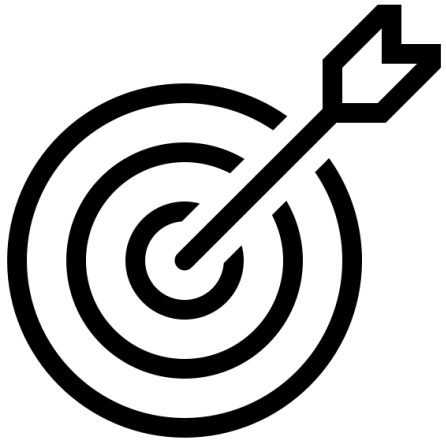
Efficiency



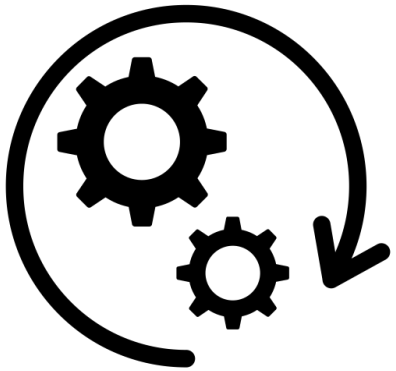
Accuracy

How?

Efficiency

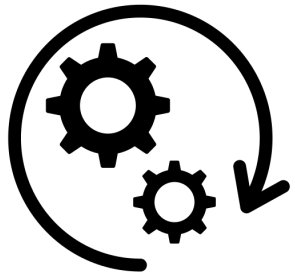


Accuracy

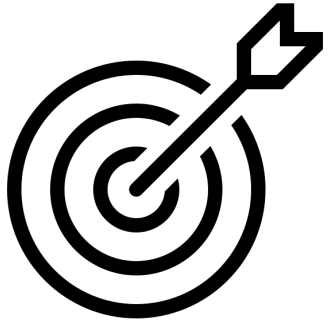


Efficiency

Accuracy



Efficiency



Accuracy

conflicting objectives

Ad-hoc approach

Score = Accuracy ✕ Efficiency

Ad-hoc approach

$$\text{Score} = \text{Accuracy} \times \frac{\text{Energy baseline}}{\text{Energy submission}}$$

Ad-hoc approach

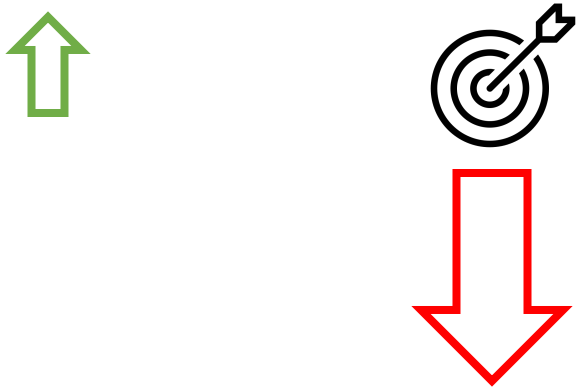
$$\text{Score} = \text{Accuracy} \times \underbrace{\frac{\text{Energy baseline}}{2 \times \text{Energy baseline}}}_{= 0.5}$$

Ad-hoc approach

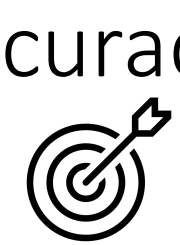
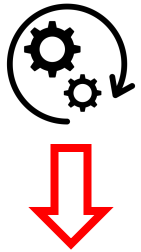
$$\text{Score} = \text{Accuracy} \times \underbrace{\frac{\text{Energy baseline}}{0.5 \times \text{Energy baseline}}}_{=2}$$

Ad-hoc approach

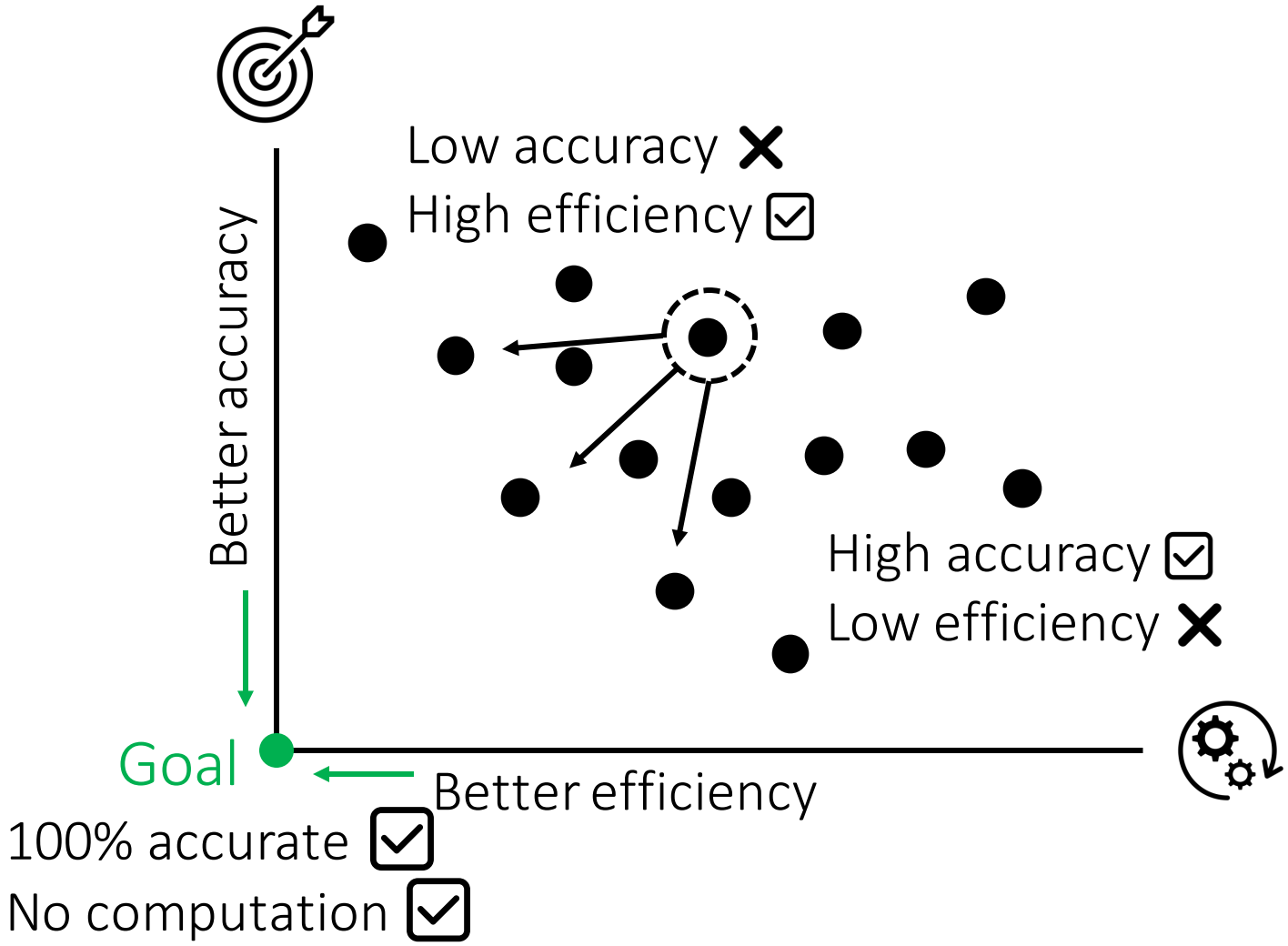
Score = Accuracy × Efficiency



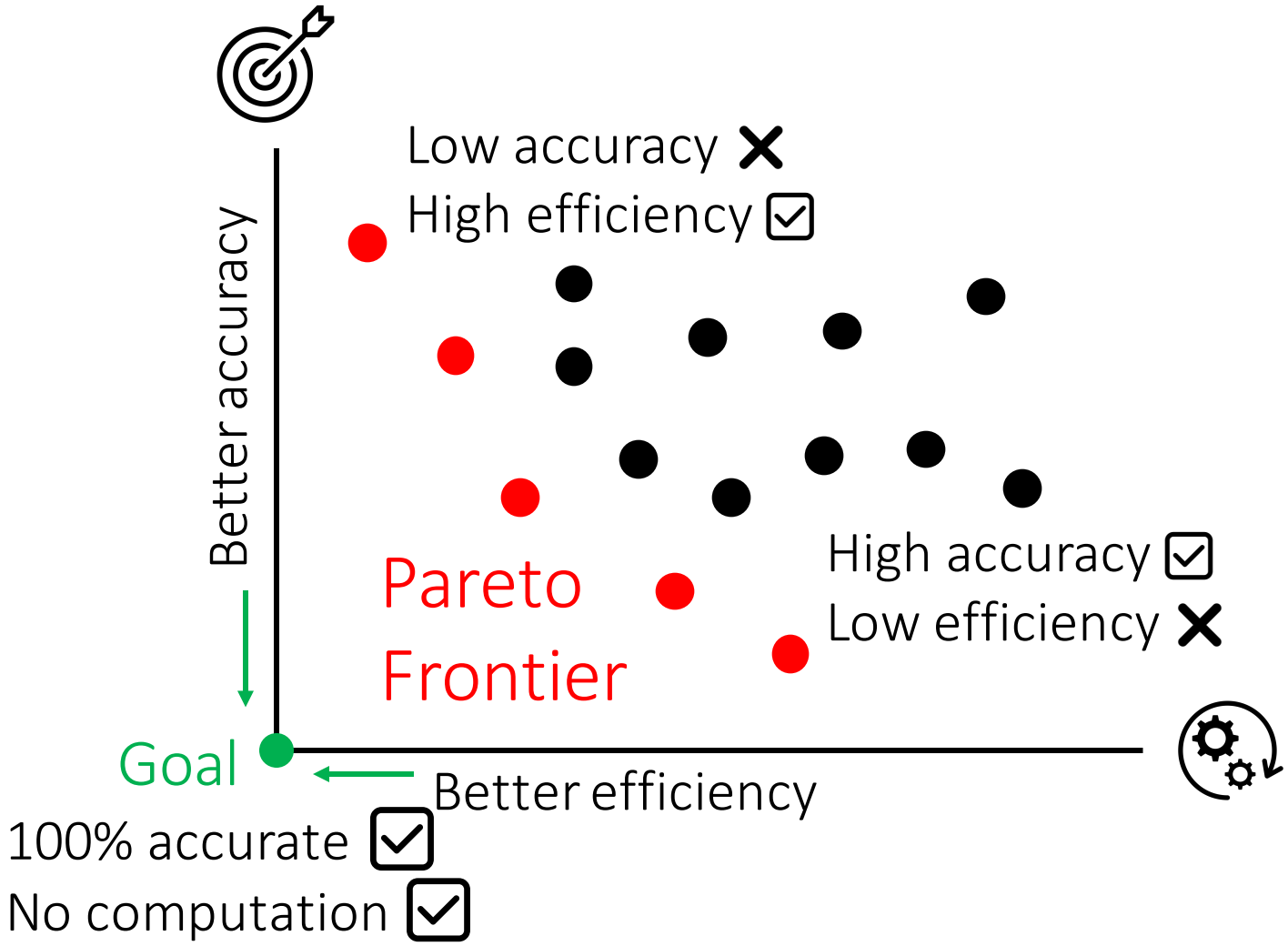
Ad-hoc approach

Score = Accuracy  Efficiency 

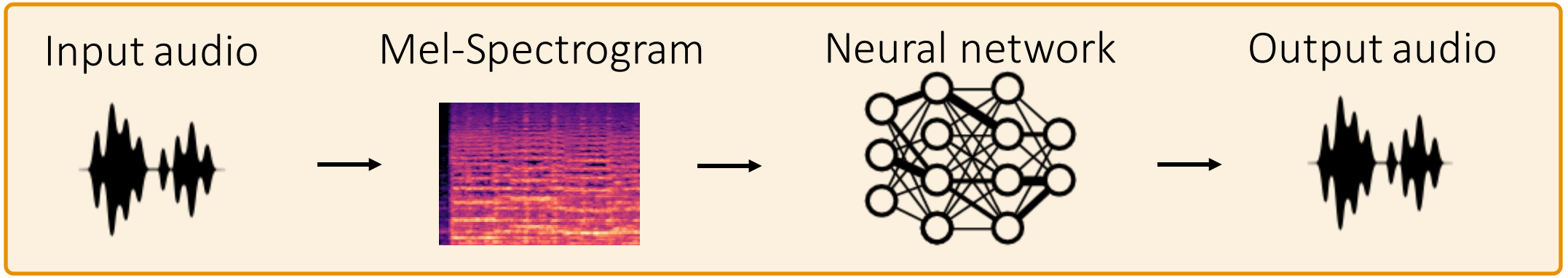
Pareto efficiency



Pareto efficiency



Neural audio synthesis



LJSpeech[1]
13,100 audio clips at
22kHz, 16-bit
~ 24 Hours

[1]https://keithito.com/LJ-Speech-Dataset/

6 Models
- 2x GANs

(MelGAN, Hifi-GAN)
- 2x Normalizing Flows

(WaveFlow, WaveGlow)
- 2x Diffusion Models

(Wavegrad, Diffwave)

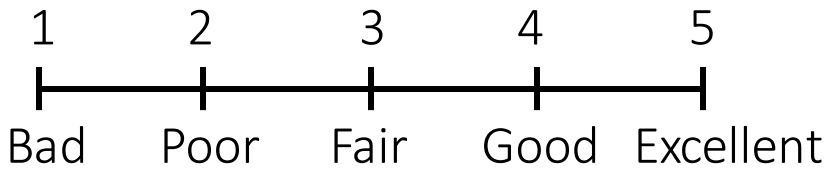
3 configurations
Small + Medium + Large

18 Models
5 days (120Hours)
NVIDIA RTX A5000 GPU
~ 30kWh

Neural audio synthesis

Mean Opinion Score (MOS)

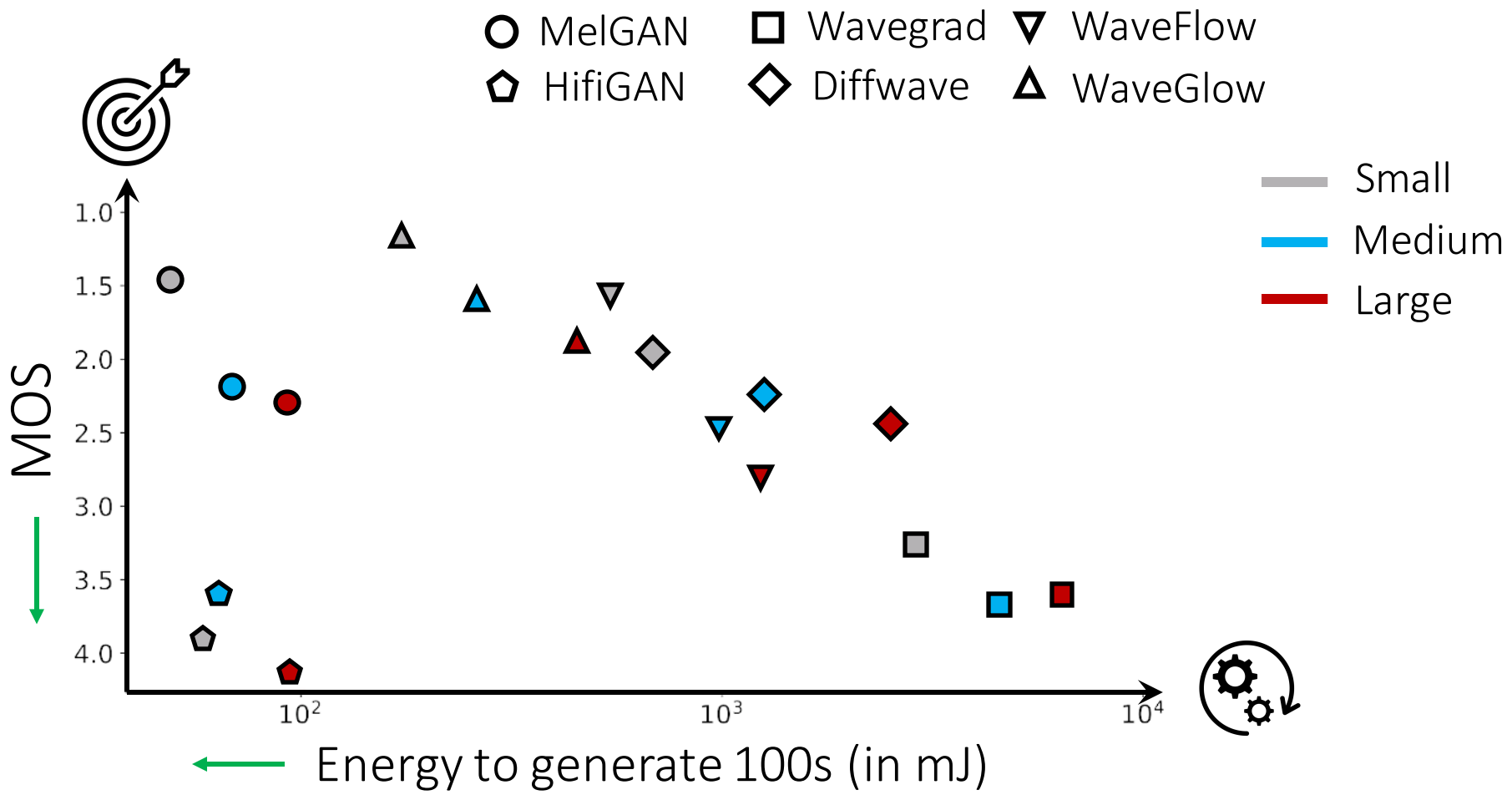
Online survey
40 participants



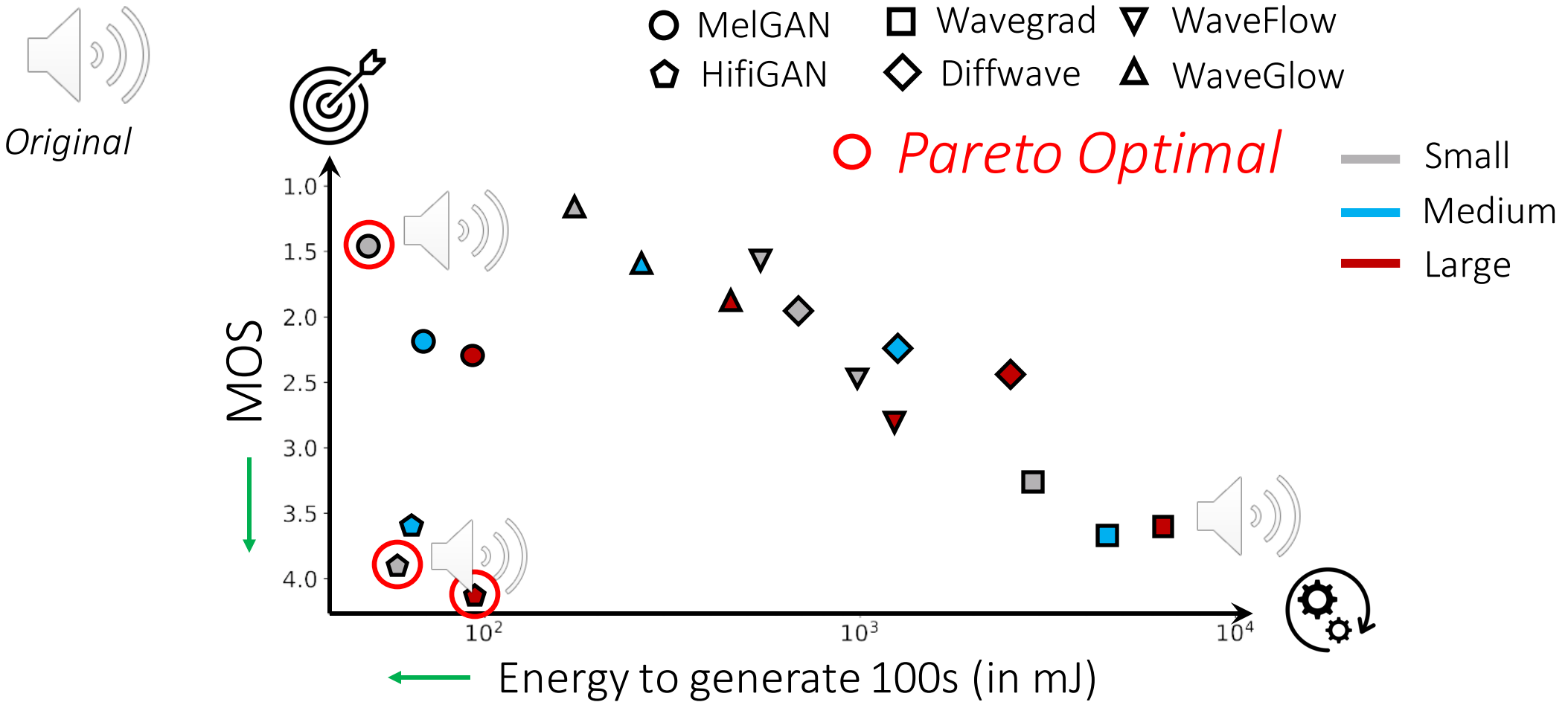
Inference footprint

- Floating Points Operations
  `deepspeed`
- Energy to generate 100s on a GPU
  `pyJoules`

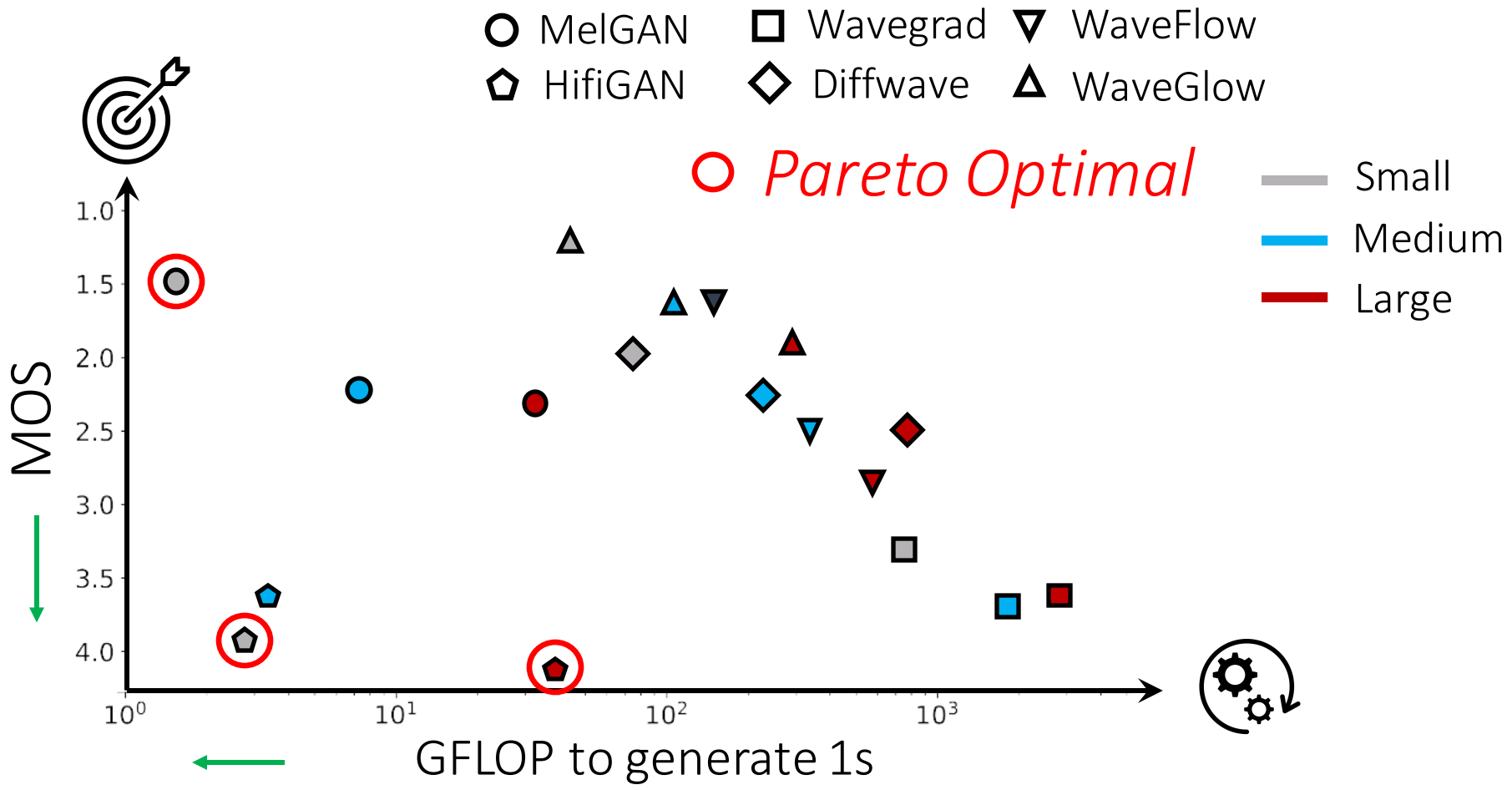
Neural audio synthesis



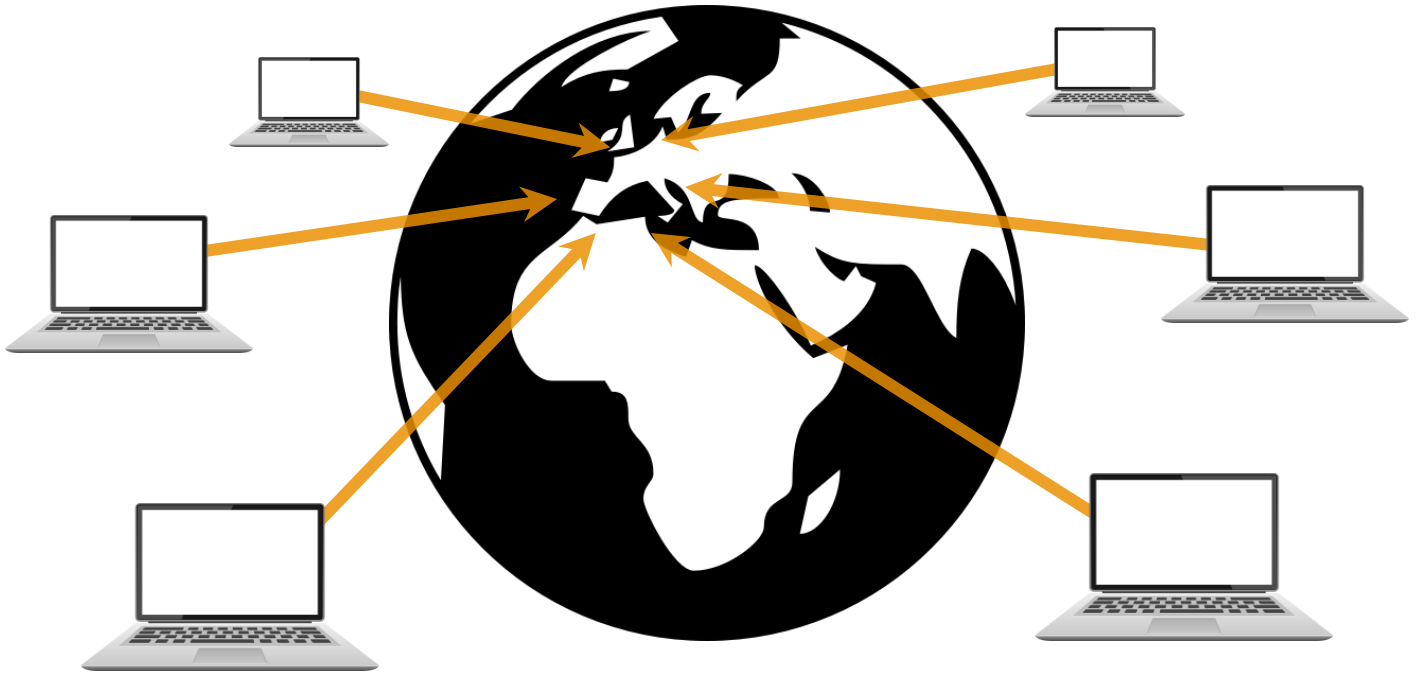
Neural audio synthesis



Neural audio synthesis

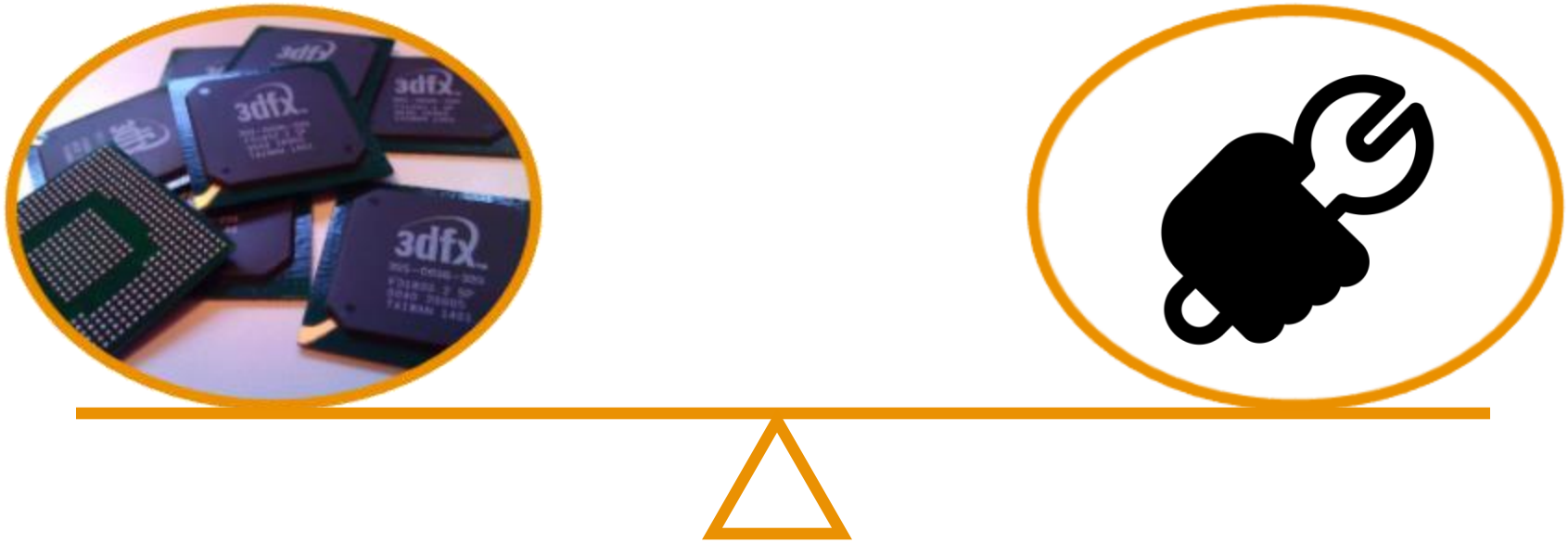


And now ?



Easy way to compare energy consumption across sites?

And now ?



Which aspect impact the energy consumption most?

Open Discussion

- Easy way to compare energy consumption across sites ?
- Which aspect impact the energy consumption most ?
- What is a worthy improvement ?

Part 2: Case study on DCASE task 4

Francesca Ronchini

DCASE Challenge Task 4

Sound Event Detection in Domestic Environments

DCASE Challenge Task 4

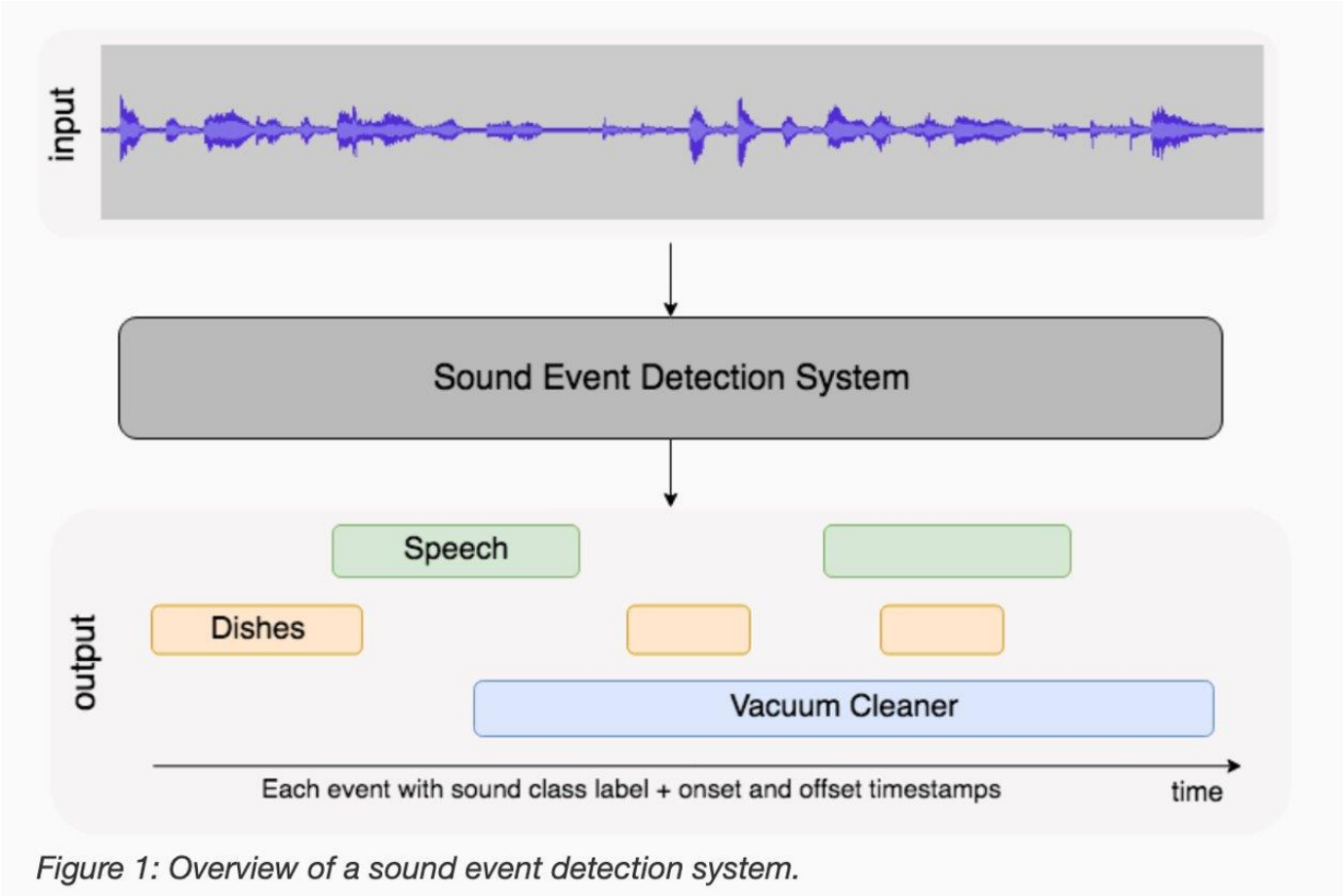
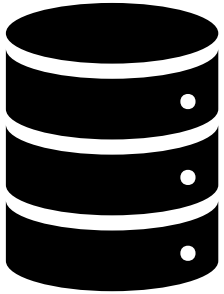


Figure 1: Overview of a sound event detection system.

DCASE Challenge Task 4



Heterogeneous dataset

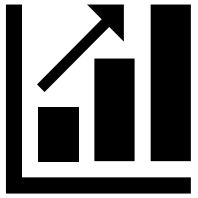
DESED dataset: weak labeled and unlabeled data

Synthetic strongly labeled data added to the dataset

Recorded strongly labeled data added to the dataset

Non-target events added to the synthetic strong labeled dataset

Motivations for environmental impact metrics



Continuous growth of models in terms of parameter complexity, often incorporating ensemble techniques.

Addressing the carbon emissions and environmental implications associated with data-driven SED systems.

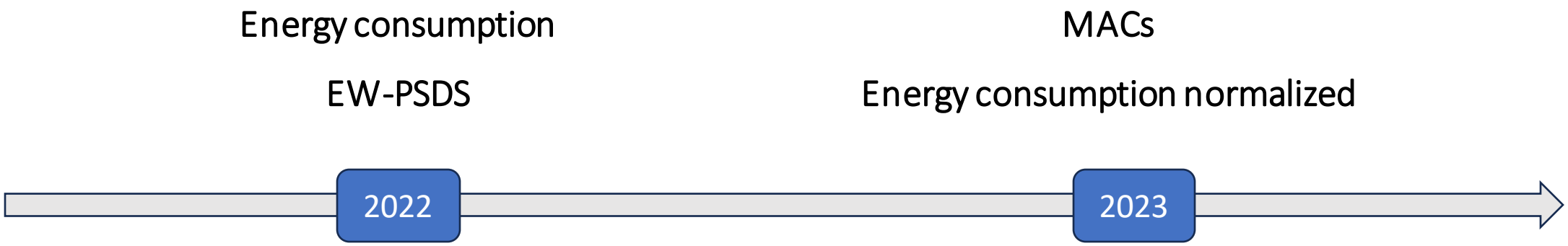
Motivations for environmental impact metrics

What is the carbon footprint of SED systems?

How can we measure the carbon footprint of SED systems?

DCASE Challenge Task 4: environmental impact

How can we measure the carbon footprint of SED systems?



DCASE Challenge Task 4: environmental impact

$$\text{EW} - \text{PSDS} = \text{PSDS} \times \frac{\text{kWh}_{\text{baseline}}}{\text{kWh}_{\text{submission}}}$$

Polyphonic Sound Event Detection Score[1]

Bilen, Çağdaş, et al. "A framework for the robust evaluation of sound event detection." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

DCASE Challenge Task 4: environmental impact

$$\text{EW} - \text{PSDS} = \text{PSDS} \times \frac{\text{kWh}_{\text{baseline}}}{\text{kWh}_{\text{submission}}}$$

✗ It is not hardware independent

✗ Does not guarantee fairness comparison between SED systems

! Could be biased by the energy consumption.

! Could lead to a high discrepancy between systems.

Bilen, Çağdaş, et al. "A framework for the robust evaluation of sound event detection." ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2020.

DCASE Challenge Task 4: environmental impact

To overcome the previous limitations, from 2023 :

- Multiply–Accumulate operations (MACs) for 10 seconds of audio prediction as additional metric.

- Energy consumption normalized relative to the baseline.

A case study : task 4

Analysis setup

101 submissions for 2022

123 submission for 2023

Filtering process

60 submissions for 2022

64 submission for 2023

Relation between energy consumption and SED metrics

2022 and 2023 submissions comparison

	System complexity ↓			Energy train (kWh) ↓			Energy test (kWh) ↓		
	25%	Median	75%	25%	Median	75%	25%	Median	75%
2022 Entries	2200000	6676303	18903660	1.815	3.699	17.291	0.010	0.026	0.046
2023 Entries	4804956	14662273	97176570	1.615	4.295	13.975	0.019	0.035	0.283

- Quartiles and median: chosen as measures of central tendency due to the presence of significant data variability.
- Inclination towards increased system complexity and energy consumption during both training and testing.

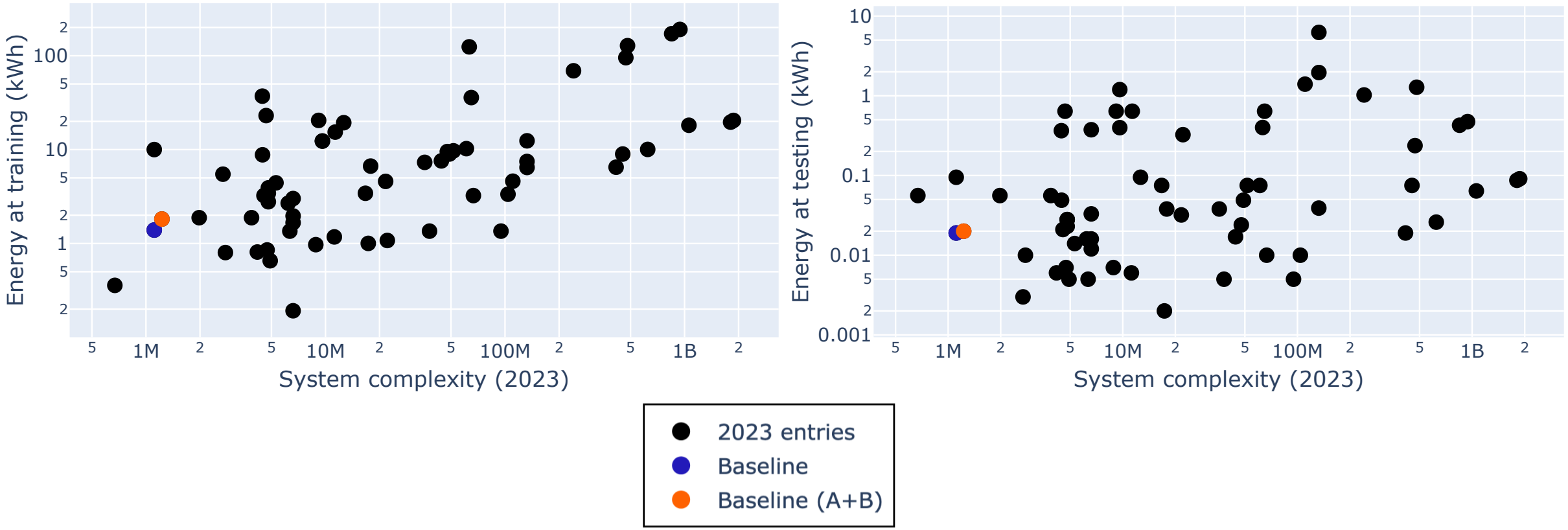
2022 and 2023 submissions comparison

The energy consumption data for this general analysis is not normalized

Not all 2022 submissions provided energy consumption values

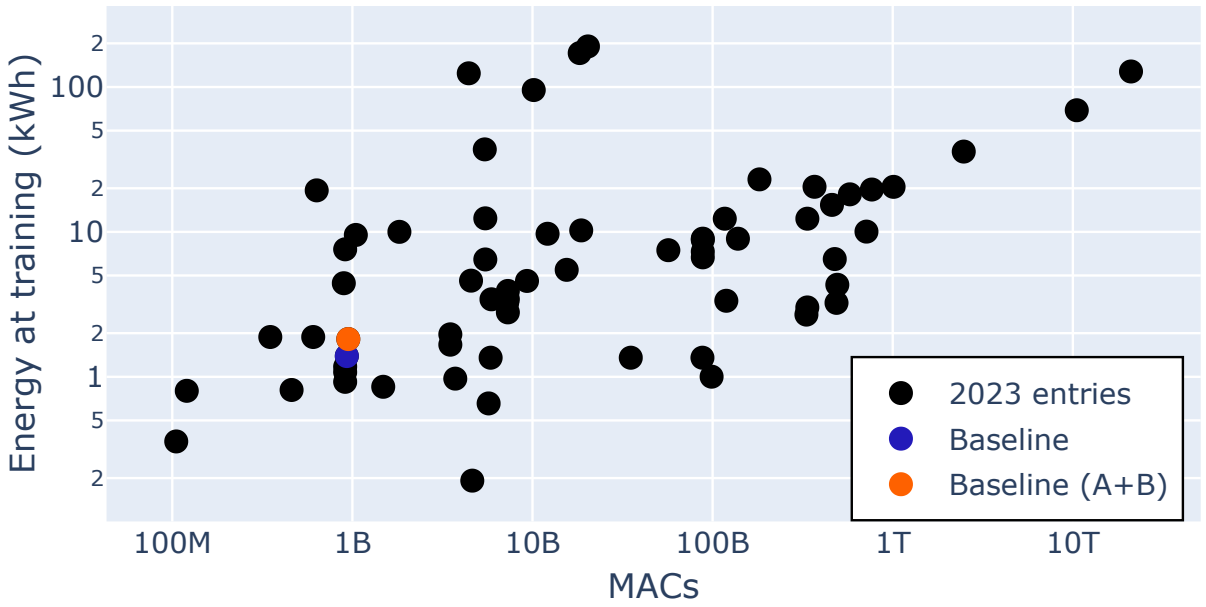
The rest of the study is going to be focused only on **DCASE 2023 submissions**

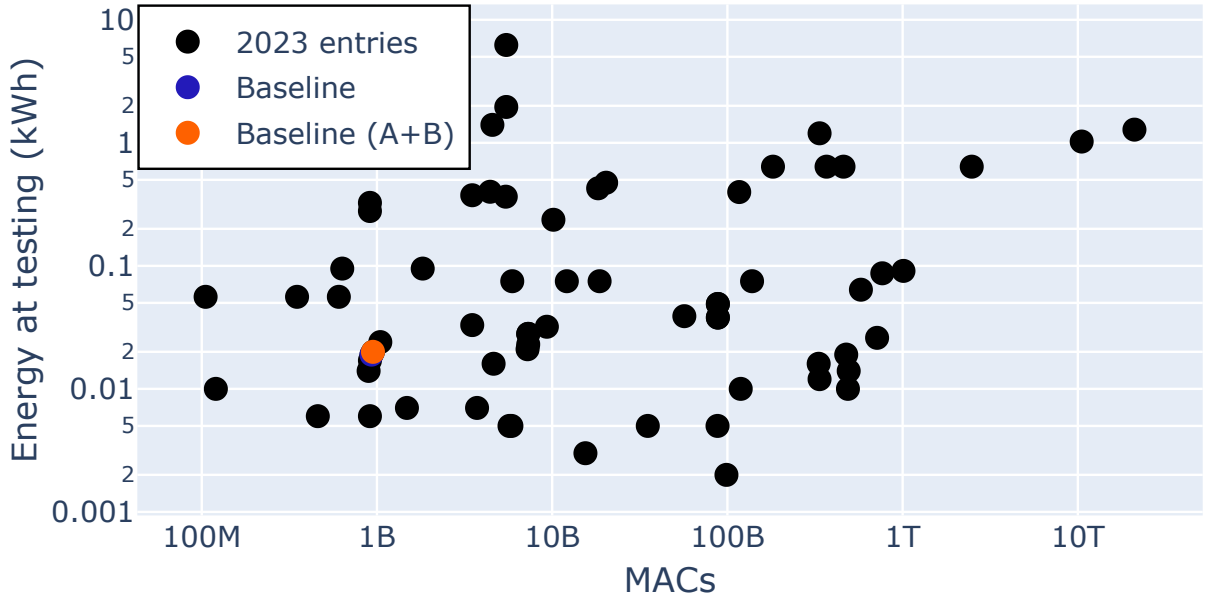
System complexity and energy consumption relation



No straightforward relation between system complexity and energy consumed at training and testing for 2023 entries.

MACs and energy consumption relation





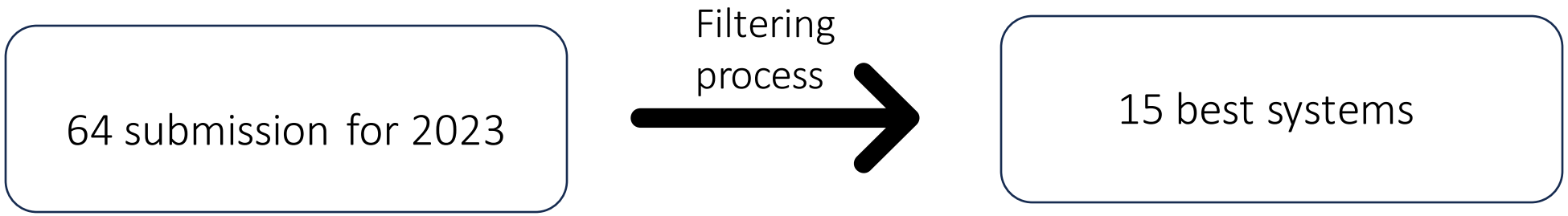
- MACs correlate slightly more with energy at training than system complexity.

Relation between MACs and energy consumption (2023)

! These three different metrics independently are insufficient to provide a comprehensive understanding of the system’s footprint.

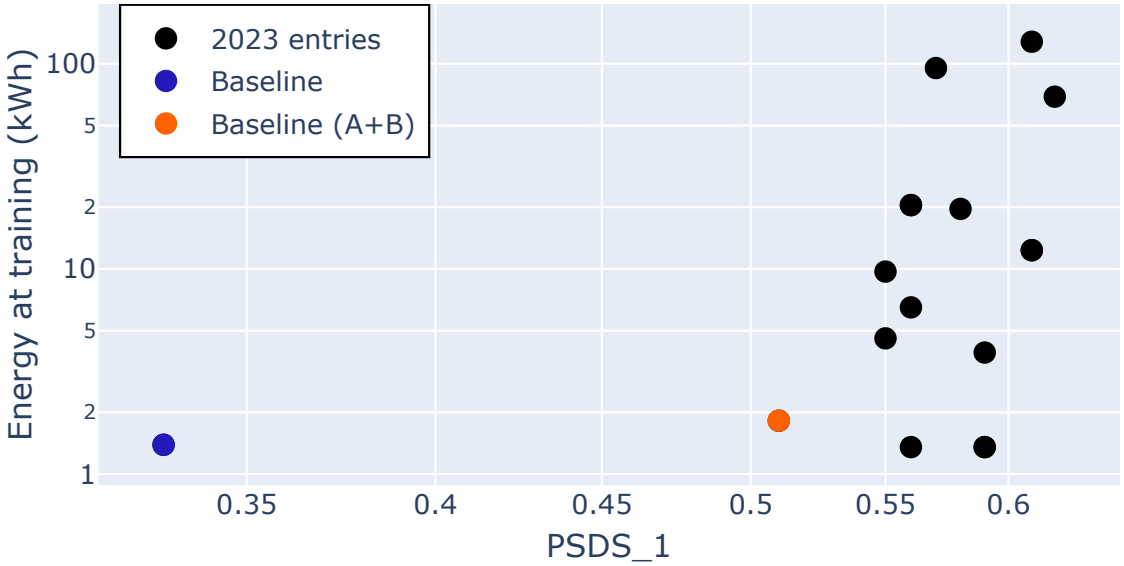
- The findings show that using just one measure is not enough to accurately assessing the extent of a system's impact.

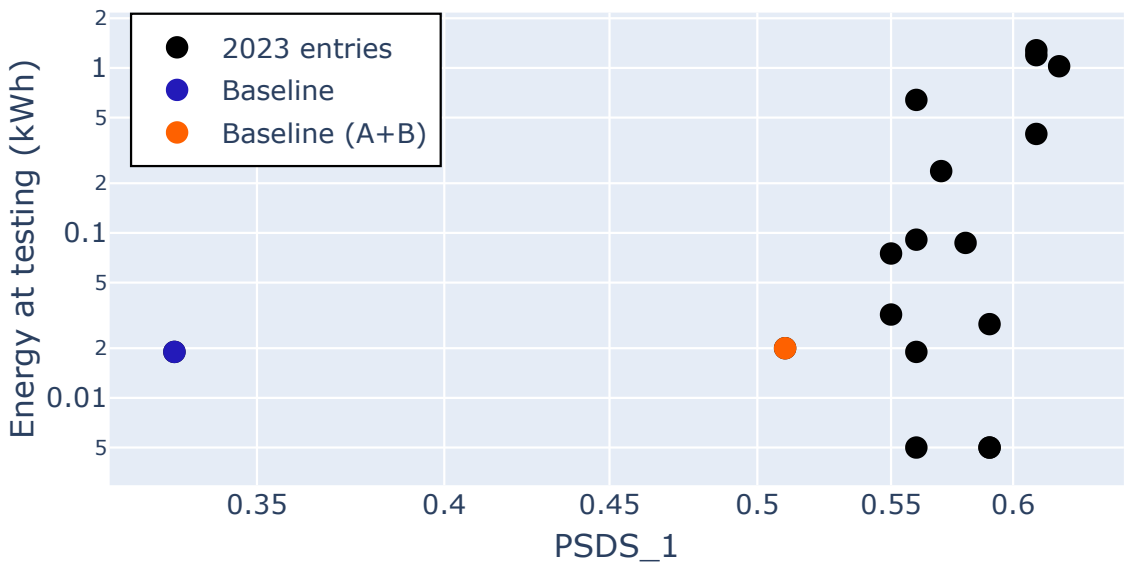
Relation between performance and energy consumption



- Metric considered: PSDS_1.
- Same conclusions hold true for PSDS_2.

Relation between performance and energy consumption

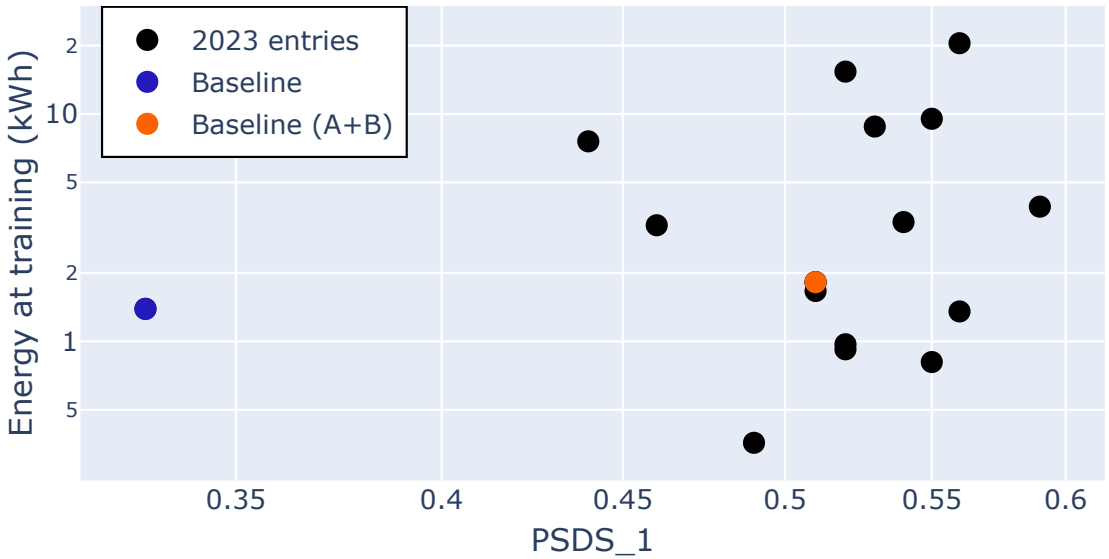




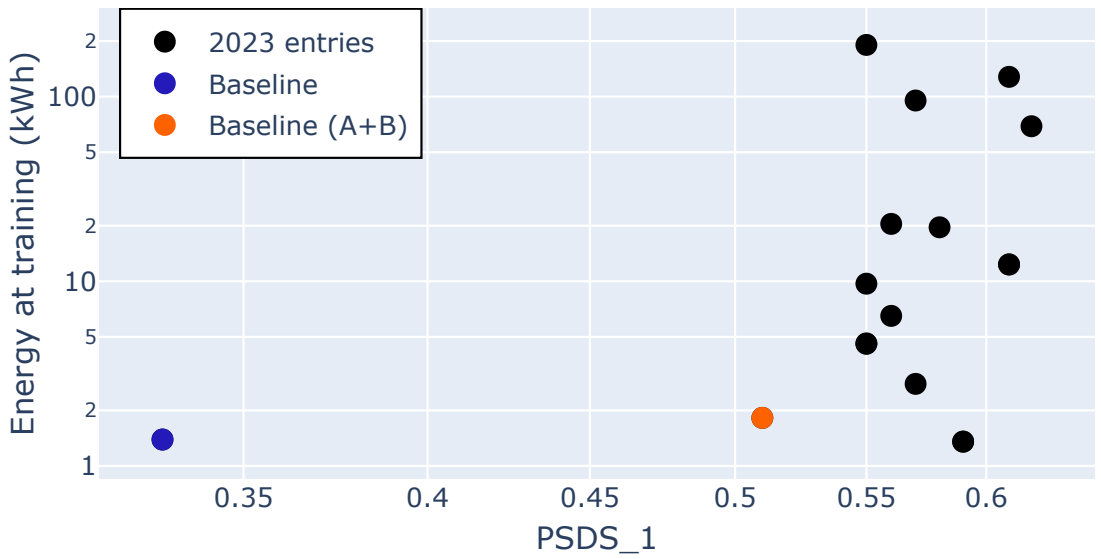
- Some systems manage to outperform the baseline results while consuming less energy.
- The top-performing systems are not the systems that consume the most energy.

Comparison between ensemble/no-ensemble systems

No-ensemble



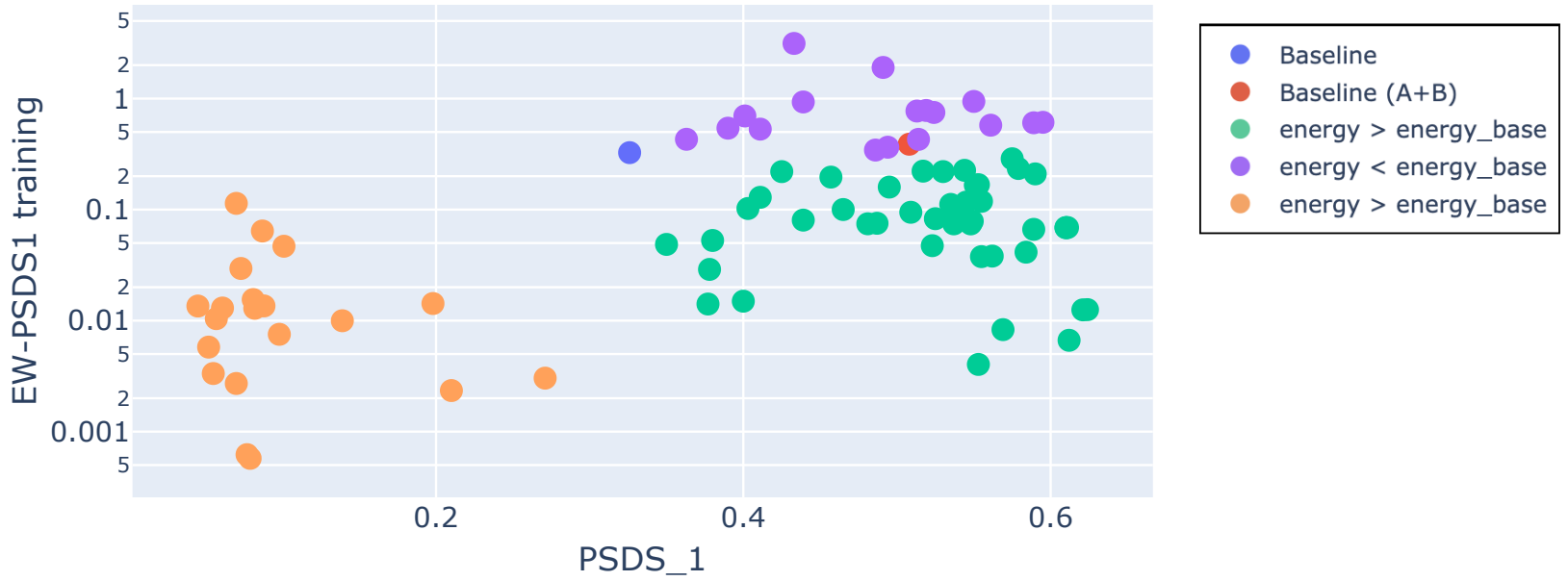
Ensemble



- A single system can provide a lighter alternative to reach good performance anyway.

- An ensemble is useful at combining systems that alone are not so good in achieving decent performance.

Relation between EW-PSDS and PSDS_1



Top-right corner systems:

✓ able to reach high performance

✓ not underestimating the environmental impact they are going to have.

Unfortunately, they are still not so many.

Thresholding based on energy consumption metrics

How much do performances degrade when a footprint cap is set?

	System complexity		MACs		Energy train (kWh)		System complexity		MACs		Energy train (kWh)	
	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1
All	1B	0.59	492 B	0.59	23.00	0.59	1B	0.62	21 T	0.62	190.00	0.62
25th	5 M	0.55	912 M	0.55	0.99	0.55	25 M	0.61	8 B	0.58	4.59	0.60
Median	6 M	0.59	4 B	0.55	2.33	0.56	67 M	0.61	72 B	0.60	9.34	0.60
75th	11 M	0.59	34 B	0.59	6.00	0.59	443 M	0.62	485 B	0.61	20.25	0.61

No-ensemble

Ensemble

Thresholding based on energy consumption metrics

	System complexity		MACs		Energy train (kWh)		System complexity		MACs		Energy train (kWh)	
	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1	Max	PSDS_1
All	1B	0.59	492 B	0.59	23.00	0.59	1B	0.62	21 T	0.62	190.00	0.62
25th	5 M	0.55	912 M	0.55	0.99	0.55	25 M	0.61	8 B	0.58	4.59	0.60
Median	6 M	0.59	4 B	0.55	2.33	0.56	67 M	0.61	72 B	0.60	9.34	0.60
75th	11 M	0.59	34 B	0.59	6.00	0.59	443 M	0.62	485 B	0.61	20.25	0.61

- PSDS performance remains rather stable regardless of the threshold cap.
- Complexity, MACs and energy consumption are substantially decreased.

! We are spending a large amount of energy and computation to increase the performance only marginally.

To conclude

- Relying on a single metric is insufficient for accurately measuring a system's footprint.
- Systems consuming the most energy (or having the most MACs) do not necessarily outperform less computationally expressive systems.
- Pressing need for metric(s) capable of taking into account various factors to accurately estimate the energy consumption.
- Taking into account the task-wise performance of the systems.

Conclusions

Romain Serizel

Take home

- Many (complementary) metrics
  - A single one is not sufficient
- Many potential shortcomings when comparing systems
  - Across site
  - Hardware
  - Configuration

⇒ Need for standardize procedures

- Combining footprint/performance metric is not obvious
  - Balance between the criterion
  - Fit actual application needs

- How can we can this attractive at community level?
  - Wrap-up from the discussion (hopefuly)

Part 3: Hands-on

Constance Douwes & Francesca Ronchini

Hands-on

- **Step 1:** Copy the link https://colab.research.google.com/drive/1JxhtPFHZ3Cbe9LfOUBQ3peqsGlhN4r6q?usp=sharing
- **Step 2:** Connect to your Google account
- **Step 3:** Create a copy (File -> Save a Copy in Drive)
- **Step 4:** Set the execution on the GPU (Runtime -> Change runtime type -> T4 GPU)
- **Step 5:** Walk through the hands-on