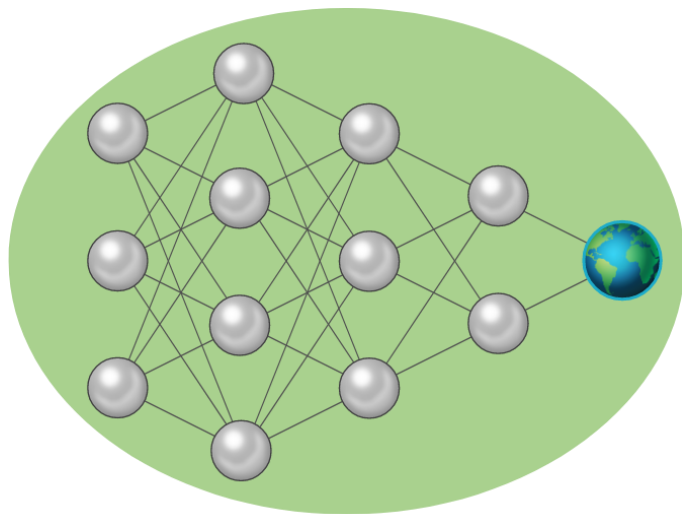

PEARLS-CRAFT



PEARLS

Precision in Energy-aware AI Research
for Low-carbon Solutions

- Configuration Research for AI Footprint Tuning

Rachel Rosten

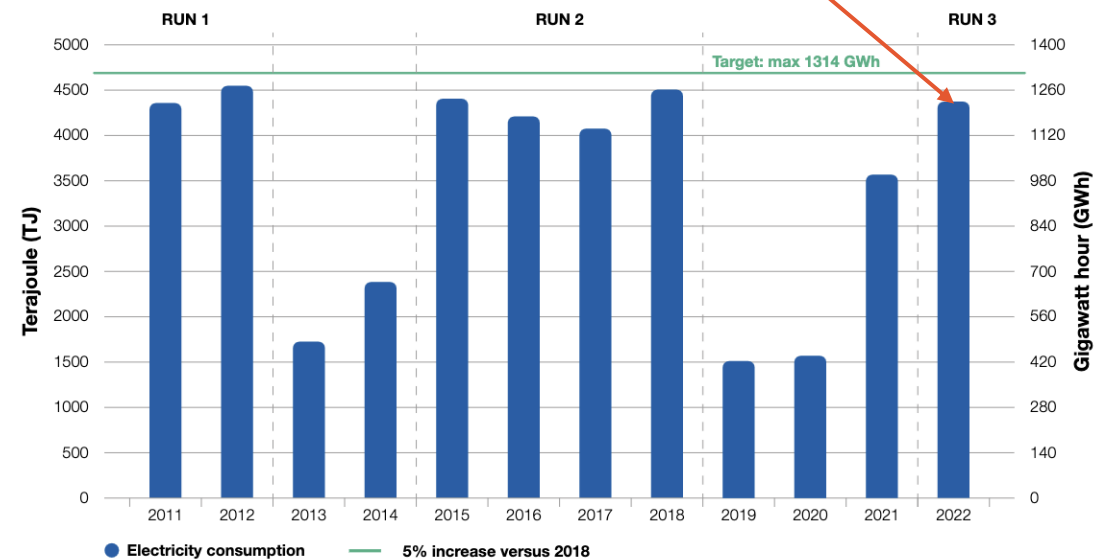
24th September 2026



Does Energy Usage Really Matter for Machine Learning in HEP?

- We all know the answer is yes, but for the sake of argument...
- CERN cites that computing accounts for a minority of its energy usage
- LLMs and Generative AI are the big energy hogs
- In HEP, when training machine learning (ML) algorithms we need to care about:
 - Accuracy
 - Training time
 - Systematics

CERN energy usage, most of this is from the accelerators!



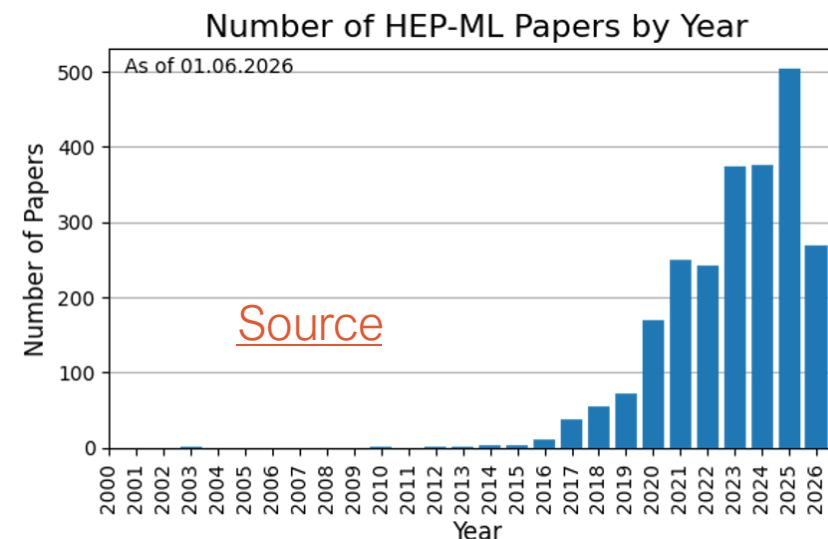
Source: [CERN Energy Report](#)

But...

- The energy usage reported by CERN is not the energy usage of computing
- World-wide LHC Computing Grid:
 - Clusters often in **carbon-hungry countries**
 - No published estimates of worldwide CO2 output of experiments
- ML use is **increasing rapidly** in HEP and algorithm complexity is increasing
- As we all know, the time to start focusing on reducing CO2 footprint is **now**

Table 5. Daily average carbon intensity for two different days. Data taken from [Electricity Maps](#). [Source](#)

	March 5th 2023	March 29th 2023
France	64	137
North Sweden	16	14
South Africa	684	702
South Carolina - USA	432	786



PEARLS-CRAFT Proposal

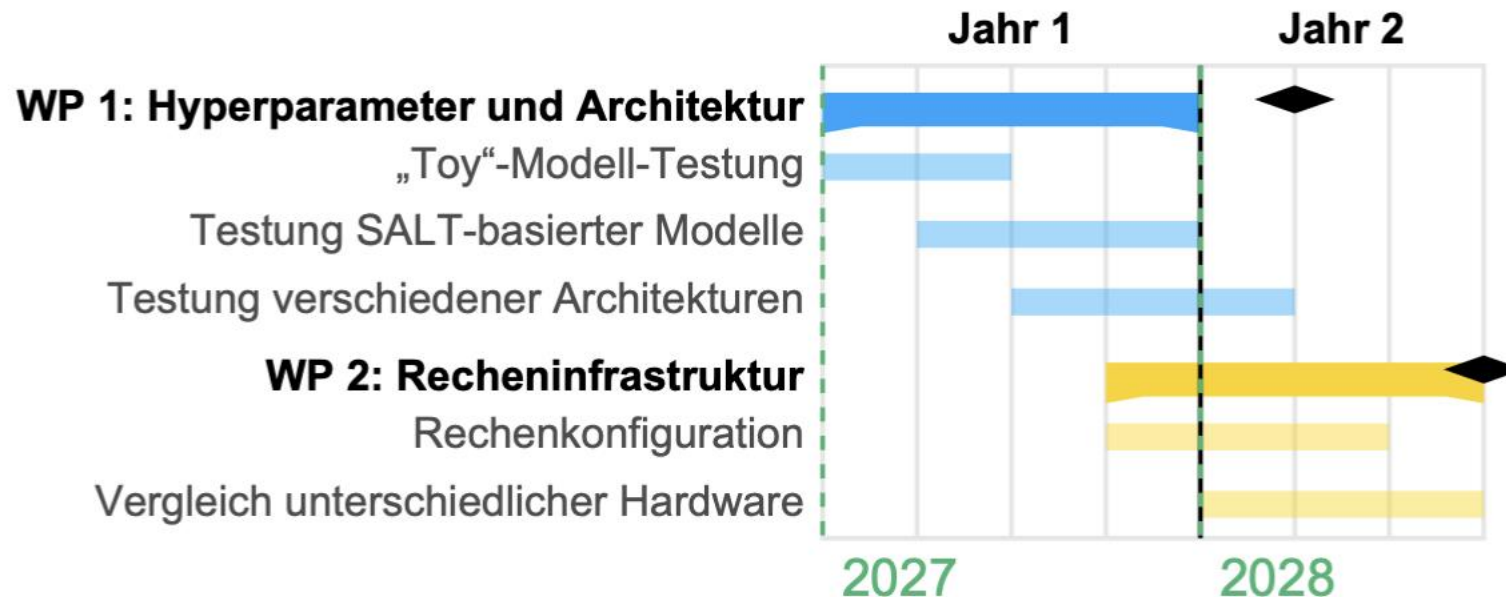
PEARLS – Configuration Research for AI Footprint Tuning



- Proposal for **Hop-On Fellowship** funding submitted to **BMFTR** two weeks ago (massive thanks to Valerie and Evelina)
- PEARLS-CRAFT aims to address two questions:
 1. How does the carbon footprint of training HEP ML algorithms depend on the **hyperparameter tunes**?
 2. How does it depend on the **hardware architecture used** during training?
- We want to be able to use the answers to these questions to develop guidelines for HEP analyzers

PEARLS-CRAFT Strategy

- Research will be separated into two phases (work-packages: WP) based on the two questions
- WP1: Hyperparameter tuning
 - Start with simple toy-algorithms based on common HEP algorithms (e.g. CNNs, GNNs) to develop a baseline understanding of the relative impact of different hyperparameters
 - Apply what is learned to any publicly available (or internal, if retaining ATLAS access) algorithms, such as [SALT](#), to develop guidelines for choosing tunes with a focus on reduced carbon footprint

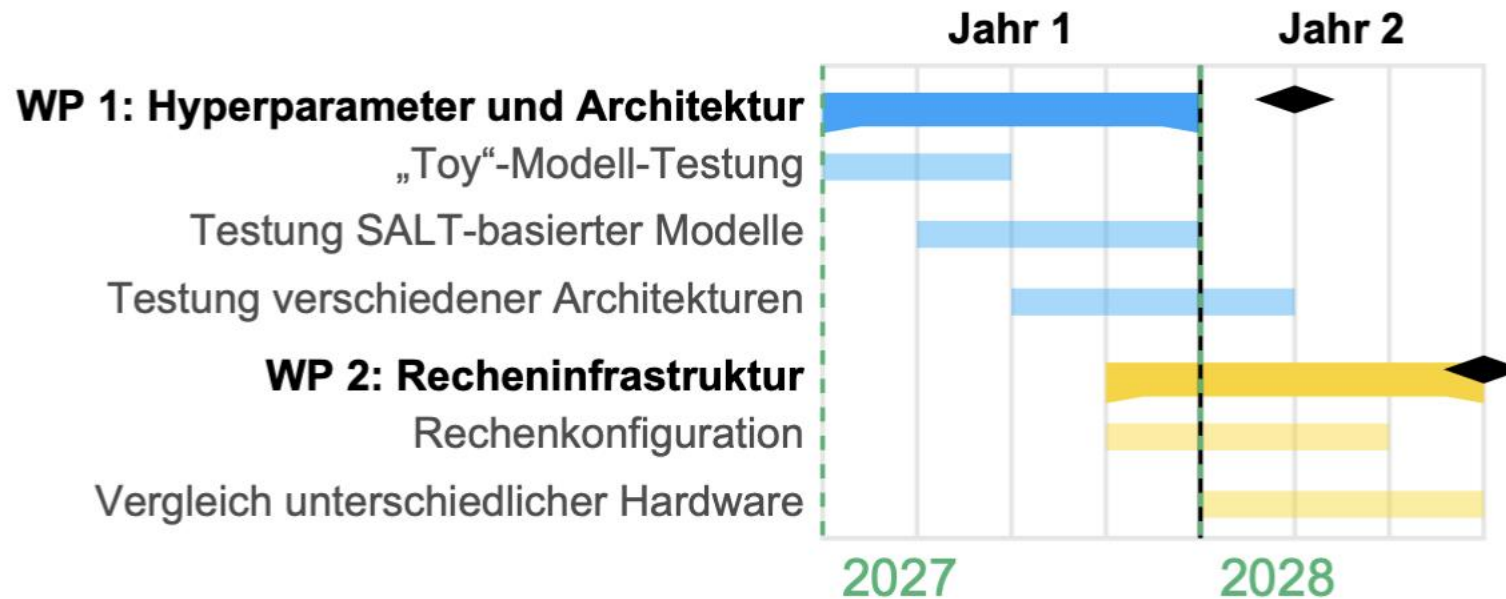


PEARLS-CRAFT Strategy

- Research will be separated into two phases (work-packages: WP) based on the two questions
- WP2: Computing infrastructure, with two approaches
 - Given fixed hardware (GPU models, available memory, etc), how does the job configuration impact emissions
 - How does the same job configuration impact emissions when using different hardware?

Important when running
on local clusters

Important when
considering the grid
Location matters too!



PEARLS-CRAFT Notes

- How to measure the CO₂e during these studies? Will review and select from available packages →
- Which algorithms and data are publicly available such that our results are reproducible to individuals outside of HEP experiments?
- Ideally, we will work in parallel with algorithm developers to share our results and gain feedback on reception

Table 2. Estimation of energy consumption for GPUs.

Green-Algorithms

Energy	uses the model of GPU to search in a list the corresponding TDP. You can load the TDP of the GPU if the model is not listed. If TDP is unknown, it uses an average of 200W, but the paper does not explain the reason for choosing this value.
Usage factor	GPUs usage factor is considered if known by the user. If not, GA considers 100% of usage.

CodeCarbon

Energy	uses <i>pynvml</i> library (only for NVIDIA GPUs). CodeCarbon does not measure consumption of non-NVIDIA GPUs.
Usage factor	not computed. The consumption reported by <i>pynvml</i> represents the consumption of the whole machine, and not only the process.

Eco2AI

Energy	uses <i>pynvml</i> library (only for NVIDIA GPUs). Eco2AI does not measure consumption of non-NVIDIA GPUs.
Usage factor	not computed. The consumption reported by <i>pynvml</i> represents the consumption of the whole machine, and not only the process.

CarbonTracker

Energy	uses <i>pynvml</i> library (only for NVIDIA GPUs). CarbonTracker does not measure consumption of non-NVIDIA GPUs.
Usage factor	not computed. The consumption reported by <i>pynvml</i> represents the consumption of the whole machine, and not only the process.

EIT

Energy	uses <i>nvidia-smi</i> command line (only for NVIDIA GPUs). EIT does not measure consumption of non-NVIDIA GPUs.
Usage factor	uses <i>Popen</i> to open a thread, execute the command <i>nvidia-smi -q -x</i> , get the output in a xml, and parse it to get the usage factor of the GPU.

MLCO2

Energy	uses the model of GPU to search in a list the corresponding TDP. It is not possible to load the TDP of the GPU if the model is not listed. In this case, it is necessary to do a pull request to add the value. It is not possible to choose the quantity of GPUs.
Usage factor	does not use usage factor. The GPU is considered at maximum load and this load is assumed to correspond to the measured process.

Cumulator

Energy	uses the model of GPU to search in a list the corresponding TDP. If TDP is unknown, it uses an average of 250W. It considers just one GPU.
Usage factor	does not use usage factor. The GPU is considered at maximum load and this load is assumed to correspond to the measured process.

Available tools for estimating CO₂
footprint of algorithms

Previous Research: Green Algorithms (2021)

- Green Algorithms is one of several tools developed to estimate CO2e based on run time, hardware configuration, and hardware location
- One study they performed was to estimate the CO2e of a 2017 simulation investigating radiation damage to human DNA which used GEANT4
- Studied dependence of runtime and CO2e on the number of CPU cores used
- Nice example of how reduction in runtime does not always correspond to reduction in CO2 output

Rule of thumb: “if the reduction in running time is lower than the relative increase in the number of cores, distributing the computations will worsen the carbon foot-print.”

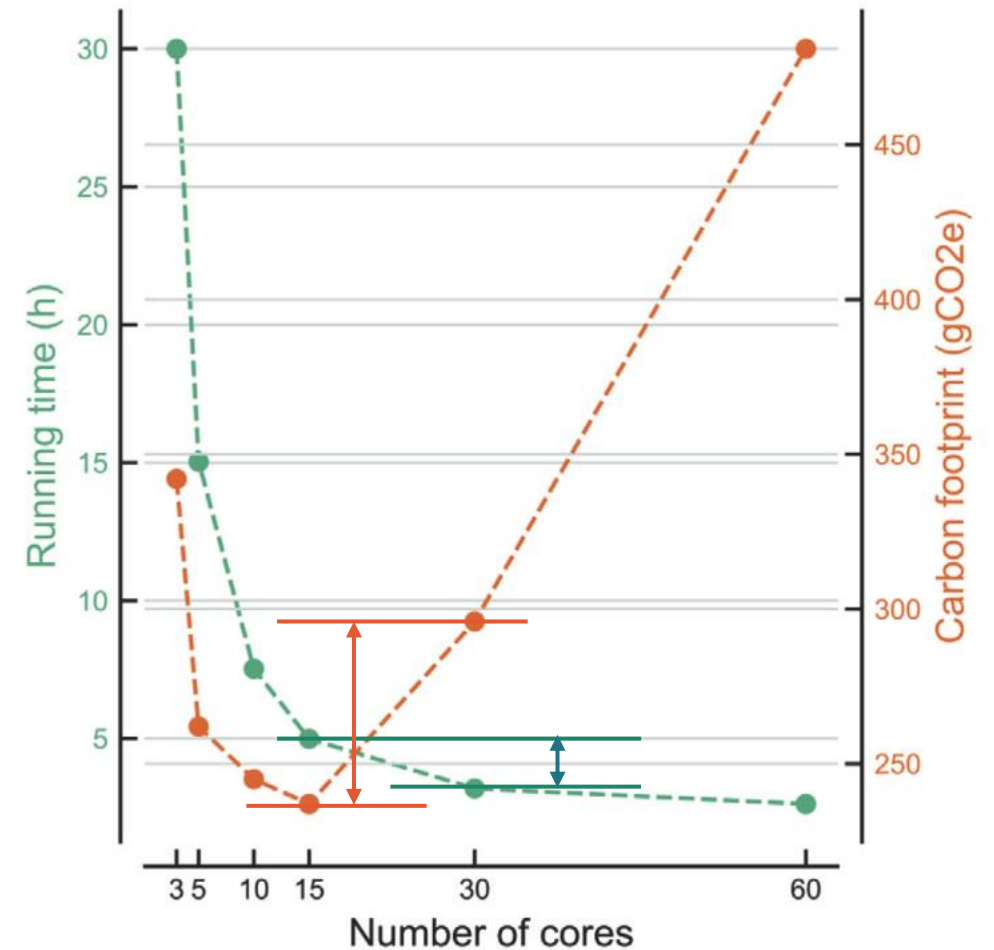
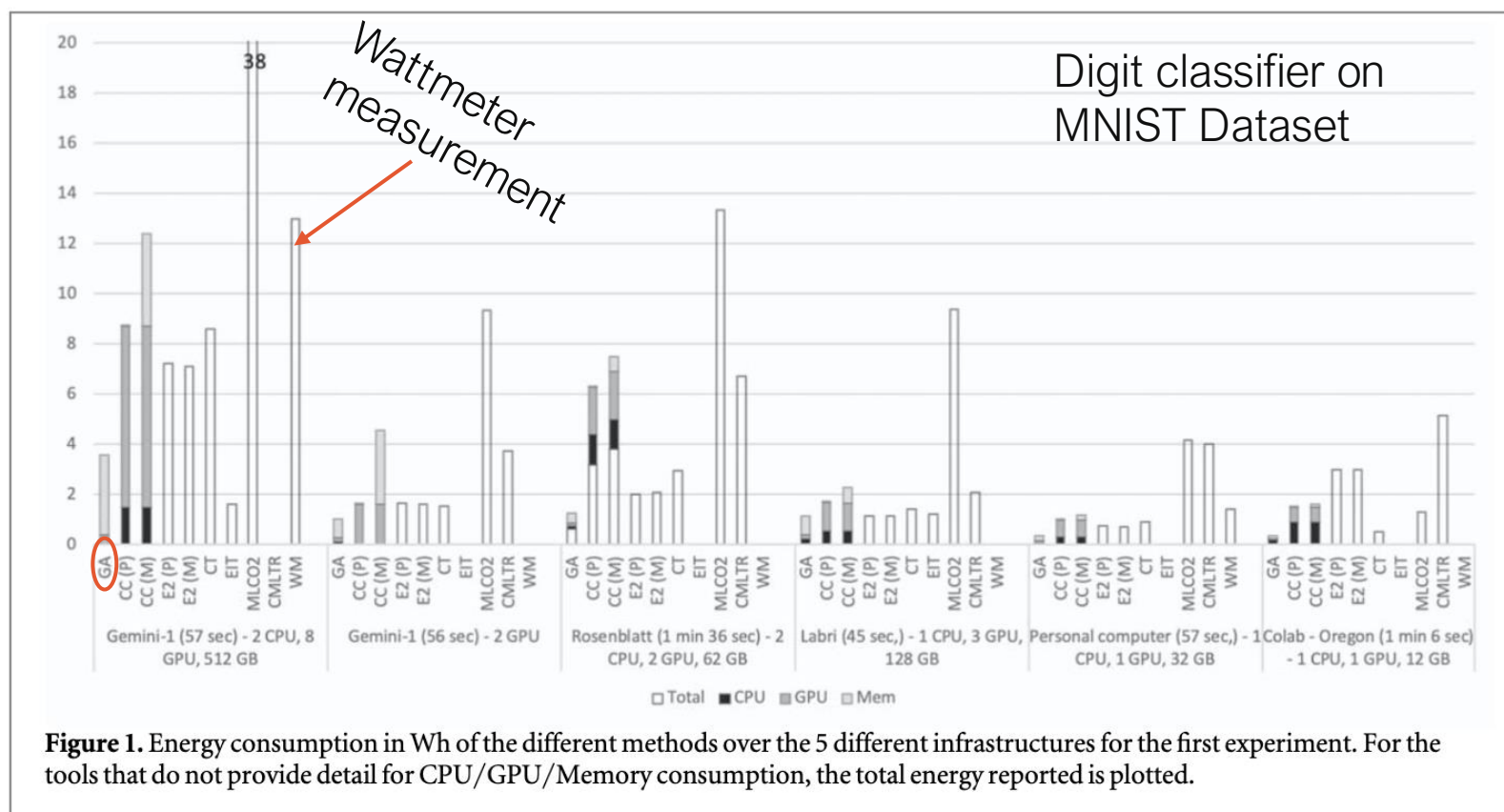


Figure 3. Effect of parallelization using multiple cores on run time and carbon footprint using TestEm12 GEANT4 simulation.

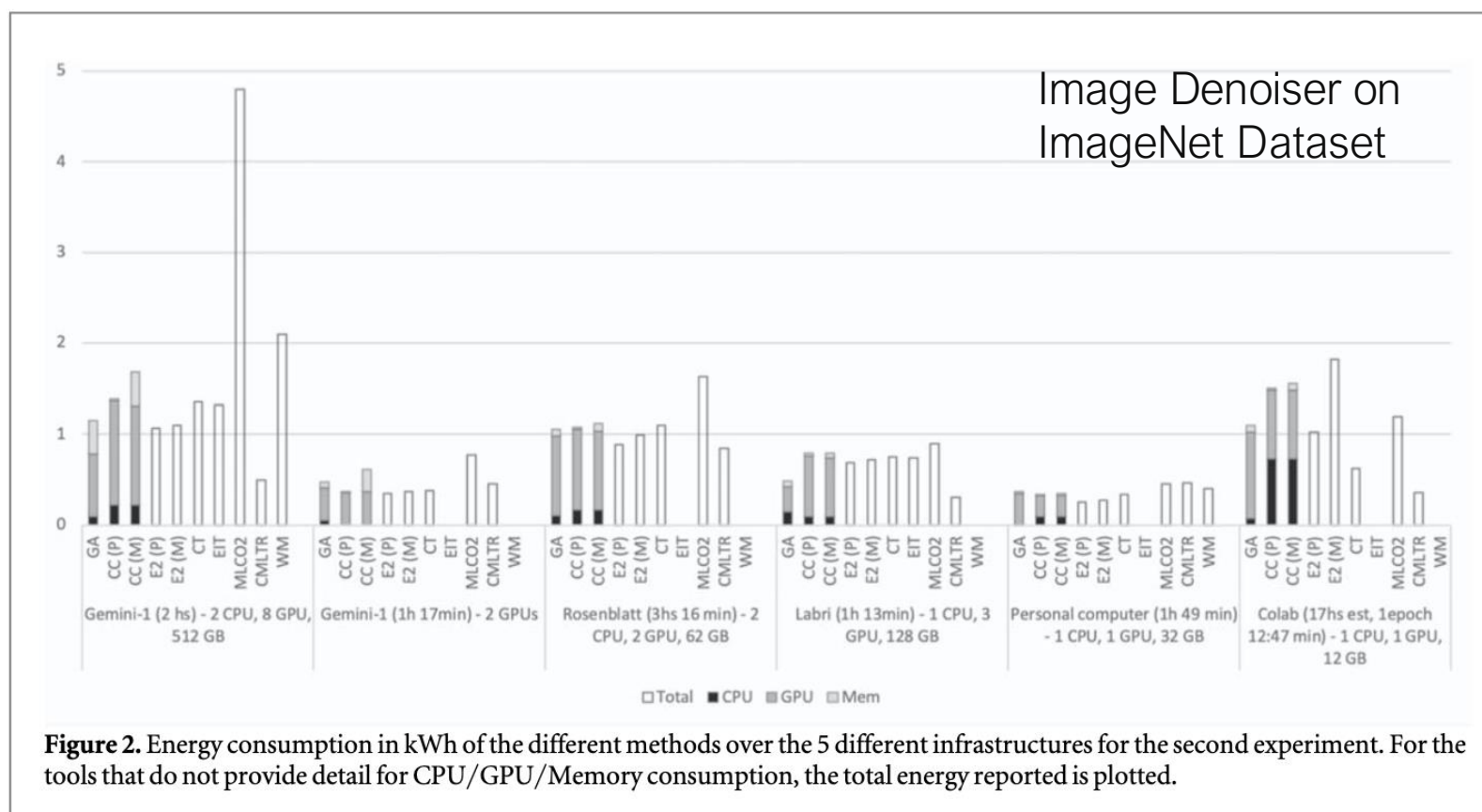
Previous Research: Carbon Footprint and Deep Learning (2023)

- An excellent comparison of various carbon estimations tools and their limitation, as well as an investigation of some of the effects of tuning hyperparameters



Previous Research: Carbon Footprint and Deep Learning (2023)

- Tool reliability will be a concern of ours, will need to choose a tool that reliably measures increases and decreases in CO₂e



Previous Research: Carbon Footprint and Deep Learning (2023)

- Investigated impact of batch size, checkpoints, and epochs
- We can use their results as a guideline/starting point
- Another reason to focus on real HEP algorithms: If balance of run-time and CO2e has poor generalizability, we will still have guidelines for algorithms currently in use

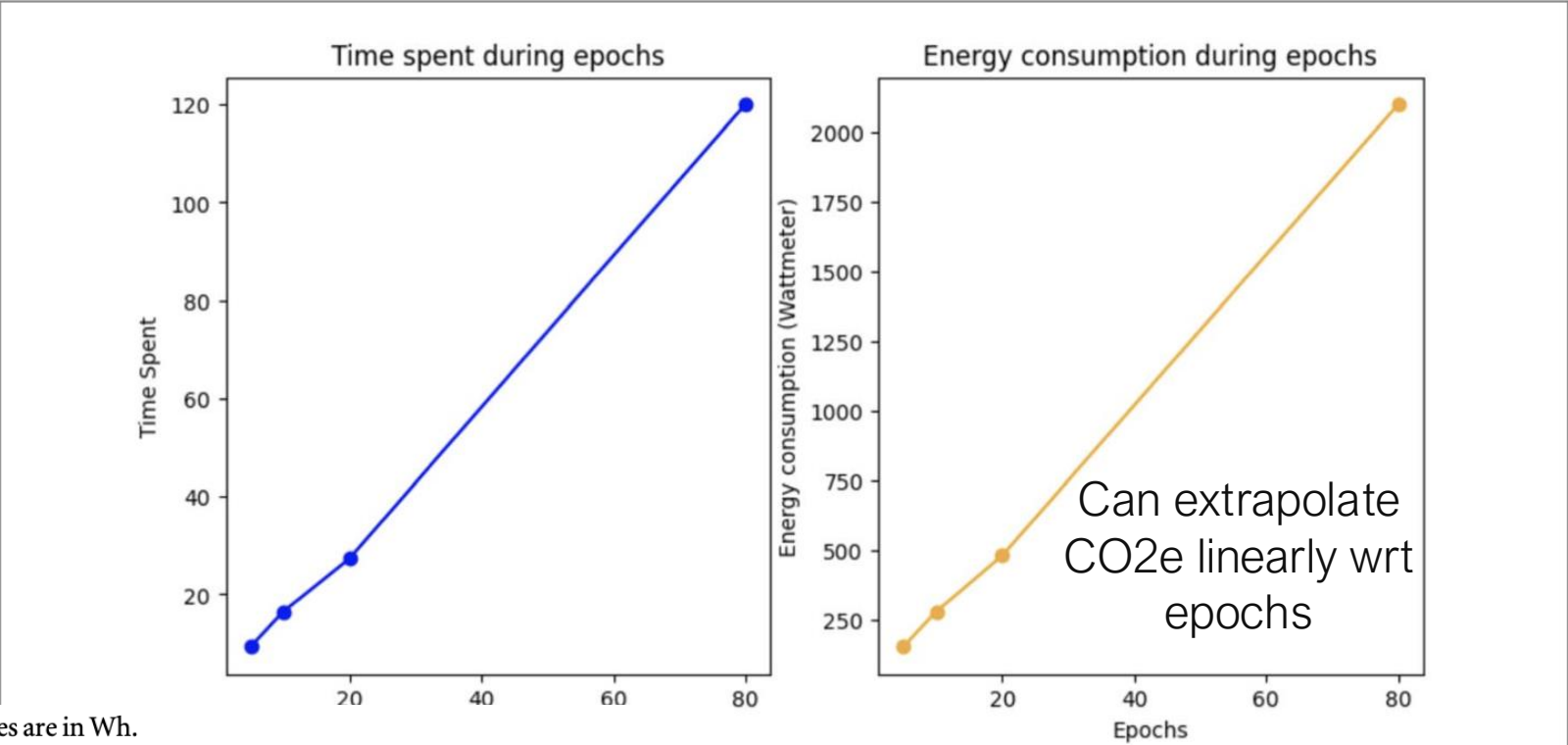


Table 14. Results of experiment 2 with different batch sizes. All consumption values are in Wh.

	Experiment with batch size 32	Experiment with batch size 64	Experiment with batch size 128
Total Energy (CodeCarbon)	252	184	246
CPU (CodeCarbon)	41	29	20
GPU (CodeCarbon)	205	152	224
Memory (CodeCarbon)	6	3	2.3
Total Energy (Wattmeter)	391	280.3	320
Time spent	25:54	16:29	10:30

at number of epochs of Experiment 2. All consumption values are in Wh.

Example Algorithm: SALT

- The [SALT](#) GNN framework was developed for [jet flavor-tagging](#) in ATLAS
- Includes a [public tutorial](#) on open data, so published results would be reproducible to individuals outside ATLAS
- Following its use for flavor-tagging, the SALT framework has been adopted by a number of analyses and projects (n.b. list is from the public documentation, but some examples are ATLAS Internal)
- SALT guidelines offer little in the way of hyperparameter tuning, and no attention is paid to carbon footprint
- Excellent opportunity for PEARLS-CRAFT to make an impact for ongoing HEP work

Current Usage

Salt is currently used for the following projects:

- Jet flavour tagging
- Boosted $(X \rightarrow b\bar{b})$ tagging
- Tau ID
- b-jet energy calibration
- Primary vertexing
- LLP vertexing
- Prompt lepton veto
- multitop analysis
- Pileup rejection using hits
- e/gamma calibration
- Pileup jet rejection with GNJVT

Conclusions and Outlook

- PEARLS-CRAFT is an extension to PEARLS that focuses on the tuning of algorithms and configuration of grid jobs (wrt hardware) for measuring CO₂e so that we can reduce it
- Many questions to ask and answer if the fellowship proposal is successful!
 - Which HEP algorithms do we dedicate time to?
 - Which individuals at CMS and other experiments would make good contacts for not making PEARLS-CRAFT ATLAS-PEARLS-CRAFT?
 - Which CO₂e tools are appropriate to use and how can we report on their reliability
- Lots to look forward to if the fellowship proposal is successful!
 - The ability to answer the question "How much does the hyperparameter tune really matter beyond the accuracy of my algorithm?"
 - The ability to complement ongoing PEARLS work such as pruning
 - A chance to expand on recommendations for grid sites based not only on geographic location to incorporate other parameters