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

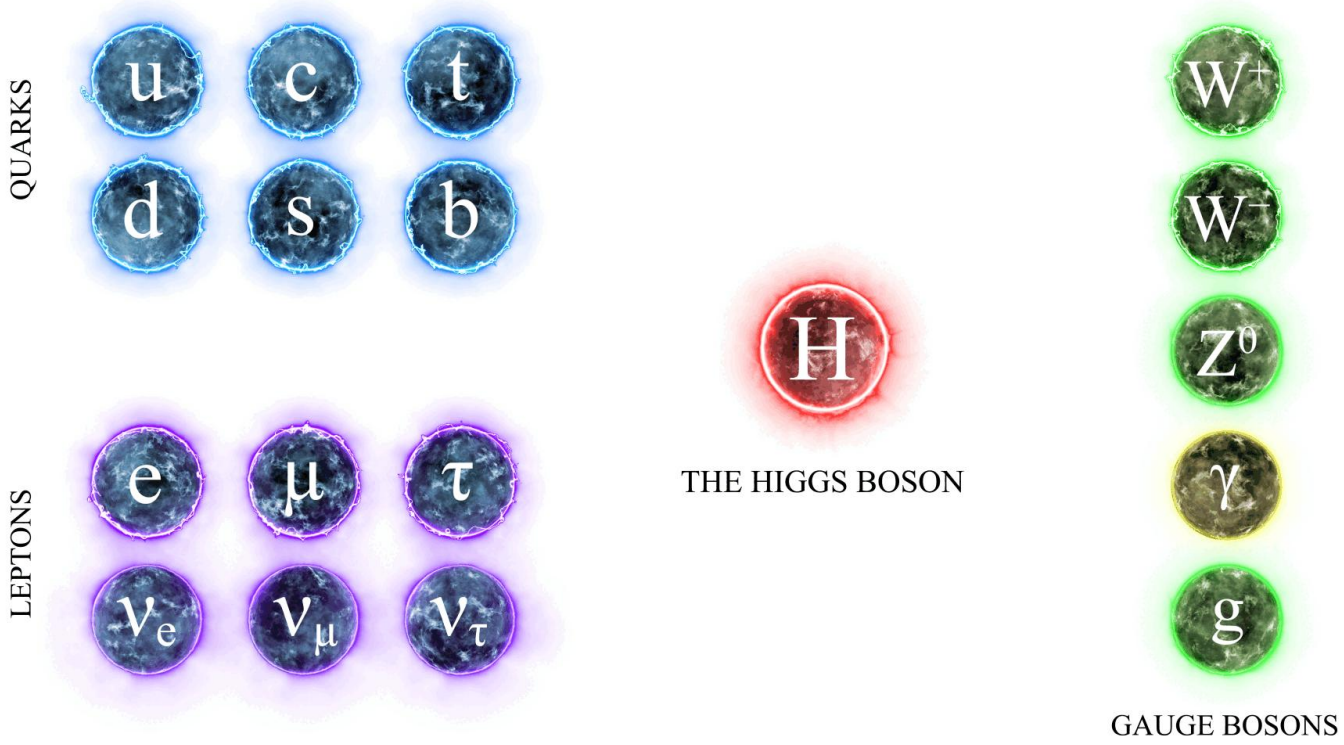
Towards Sustainable AI

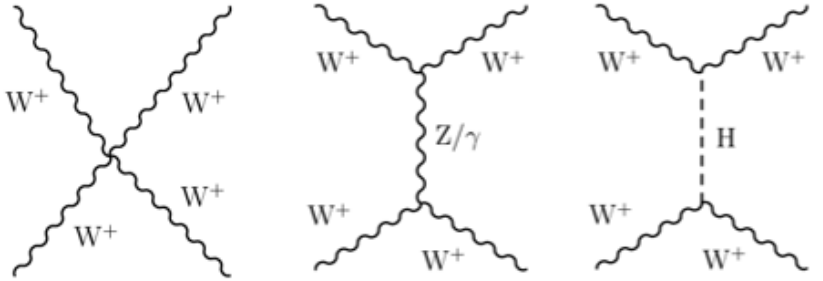
My mini CO_2 project

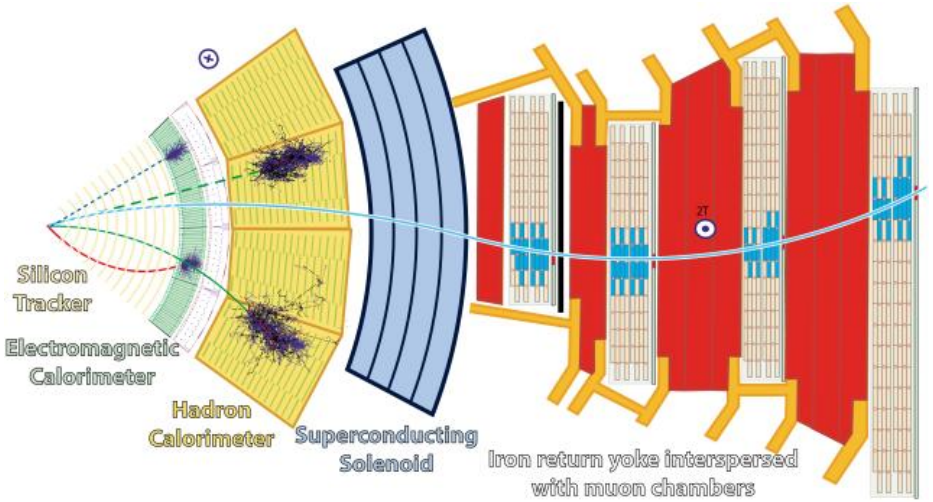
Komal Tauqeer

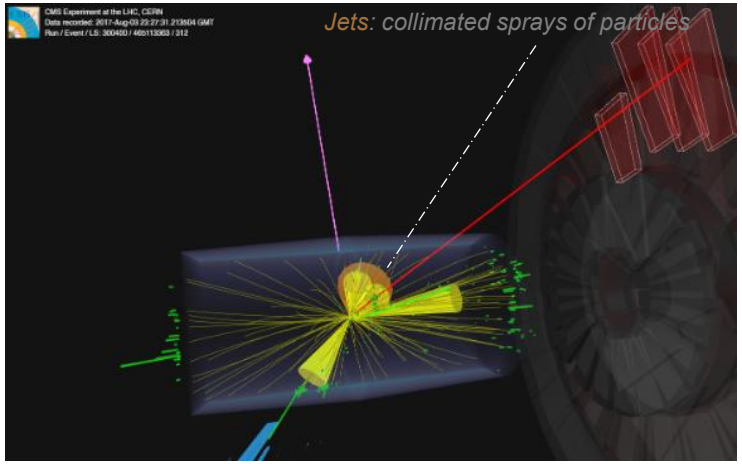
23.09.2026

My research background







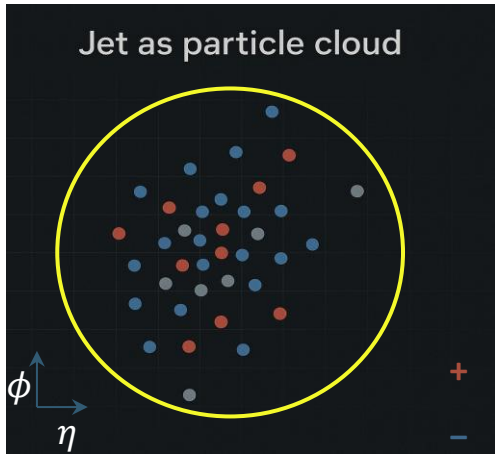


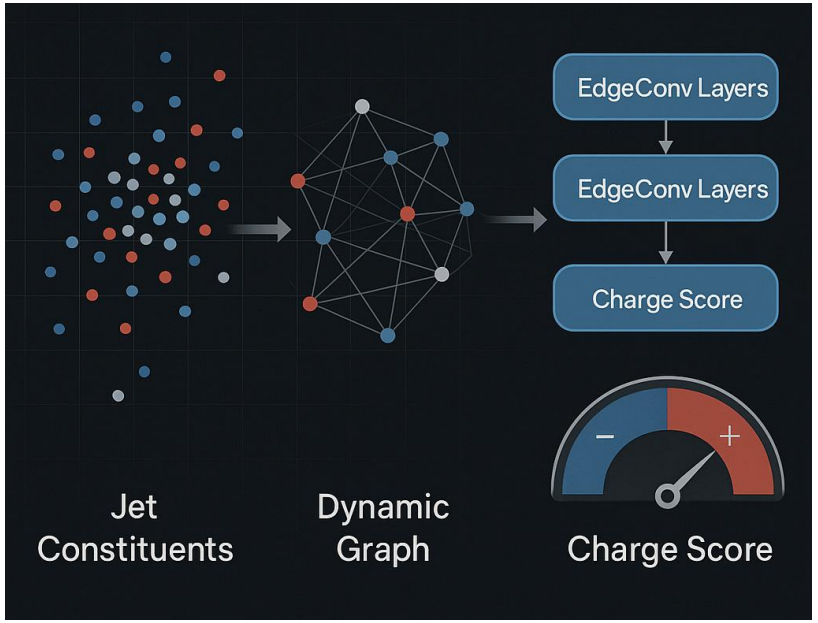
- Understanding the tiniest particles of the universe
- Searching for rare processes with the CMS experiment at the Large Hadron Collider

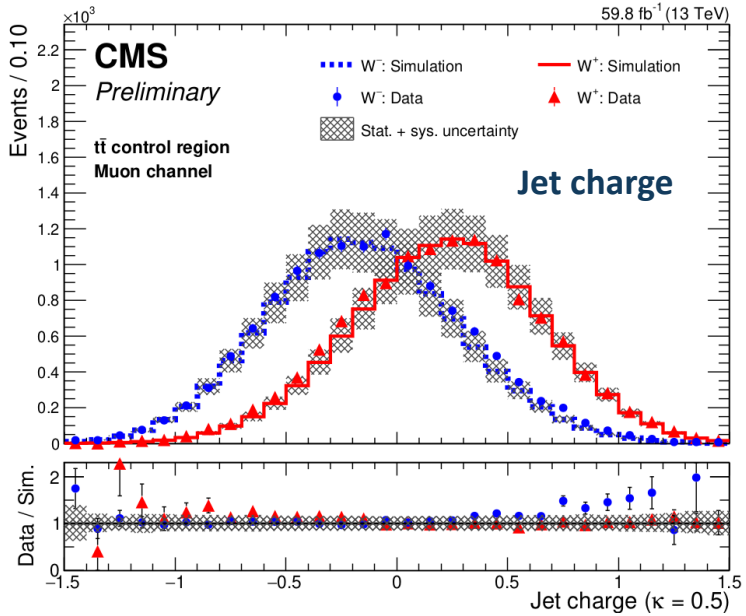
Jet charge tagger for the CMS experiment

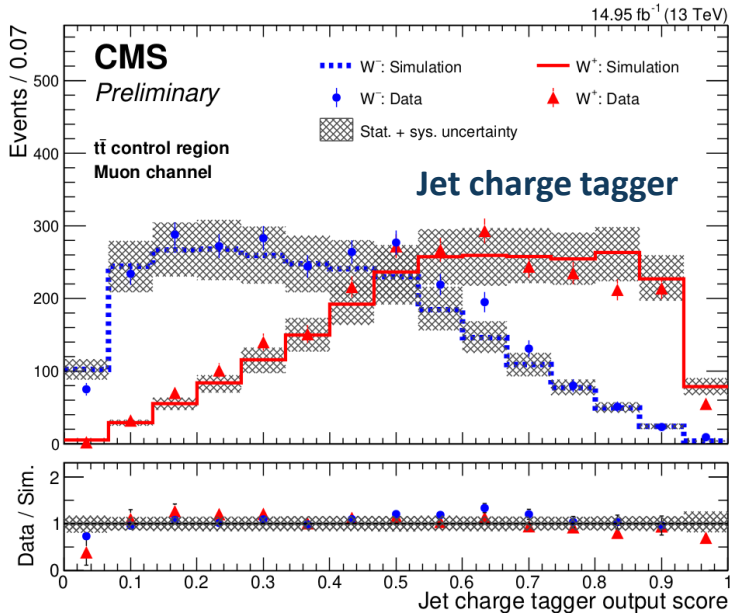
https://cds.cern.ch/record/2904357

https://publikationen.bibliothek.kit.edu/1000185309





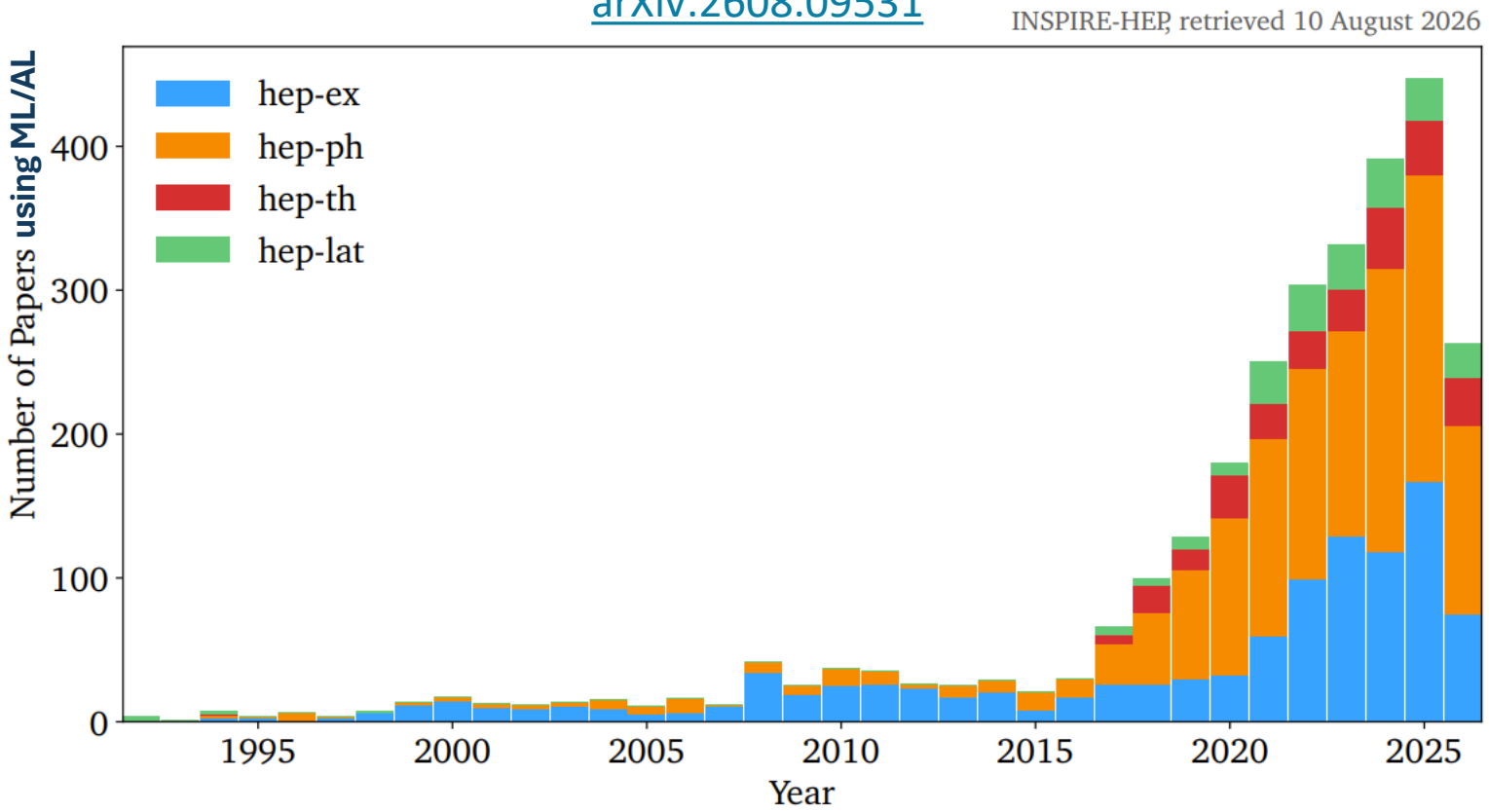




- **Ph.D. Thesis**: A search for **same-sign WW vector boson scattering (VBS)** based on an innovative charge tagging algorithm
- Developed a **Dynamic Graph Convolutional Neural Network (DGCNN)** for tagging charge of jets (collimated sprays of particles)

PEARLS & High-Energy Physics

arXiv:2608.09531



- ML/AI adoption in HEP is accelerating
- Model complexity is rising → energy & emissions

PEARLS

Energy-aware AI research

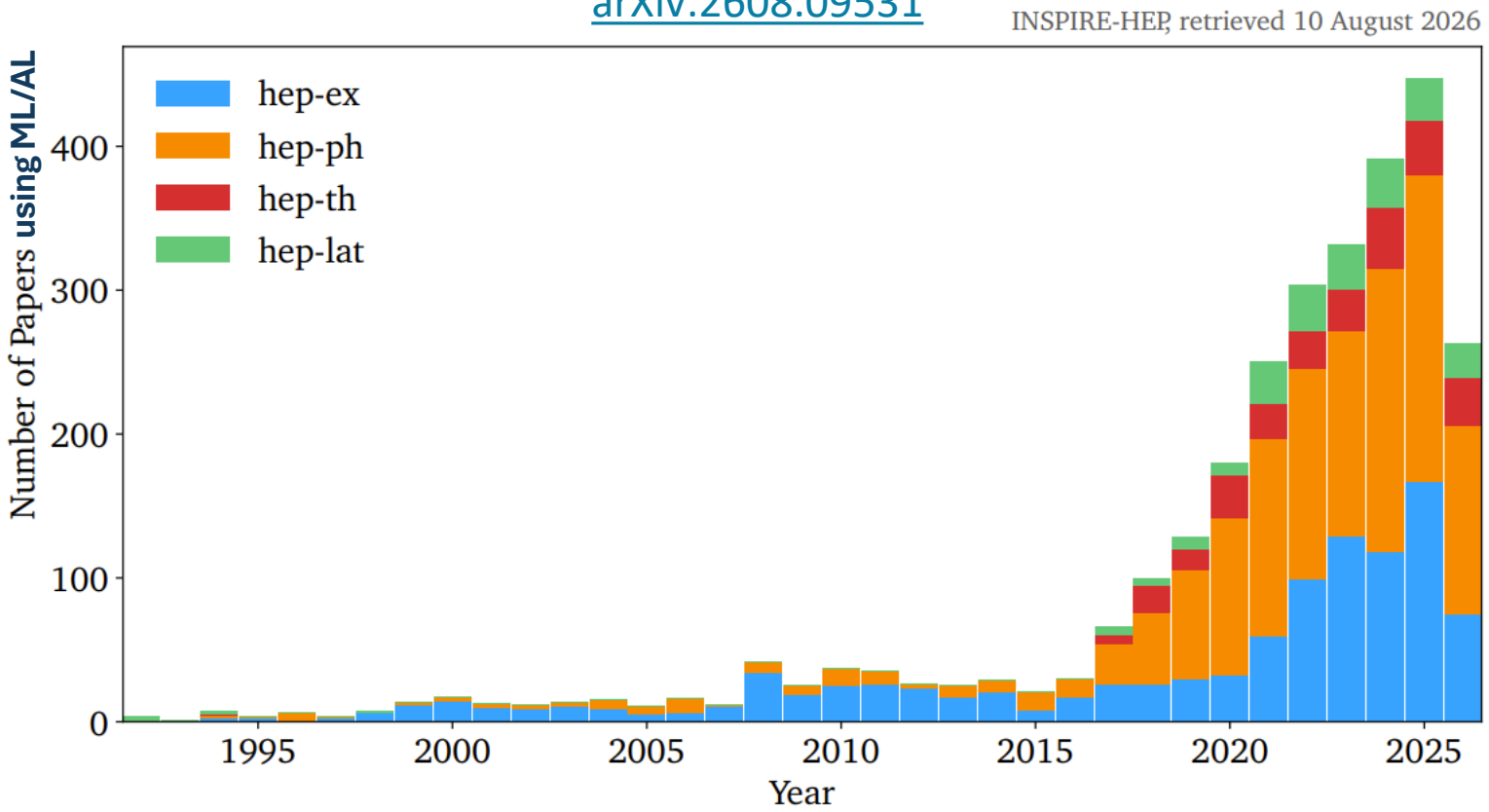
Goals:

- Measure energy consumption and carbon emissions as first-class metrics
- Optimize energy efficiency without sacrificing physics performance
- Turn measurements into actionable low-carbon AI guidance

PEARLS & High-Energy Physics

YOU CANNOT REDUCE WHAT YOU DO NOT MEASURE!

arXiv:2608.09531



- ML/AI adoption in HEP is accelerating
- Model complexity is rising → energy & emissions

PEARLS

Energy-aware AI research

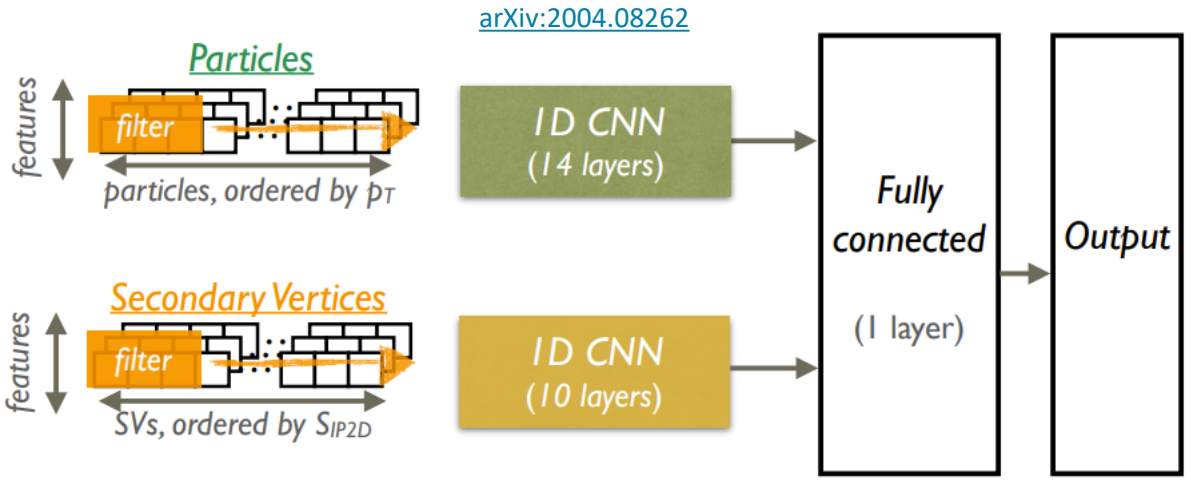
Goals:

- Measure energy consumption and carbon emissions as first-class metrics
- Optimize energy efficiency without sacrificing physics performance
- Turn measurements into actionable low-carbon AI guidance

A demonstration

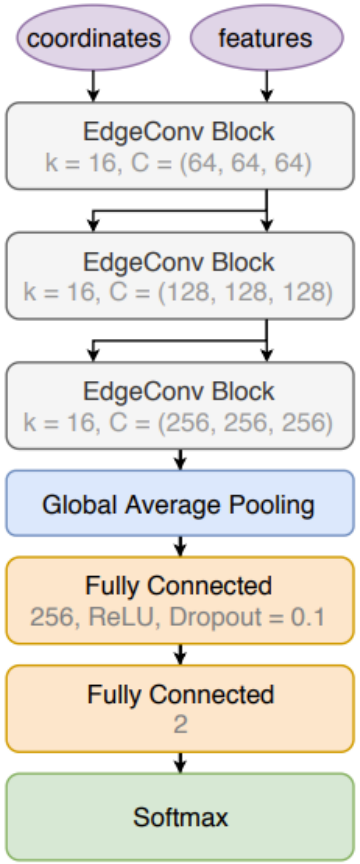
Tracking energy consumption and carbon emissions of the two famous CMS jet tagging models

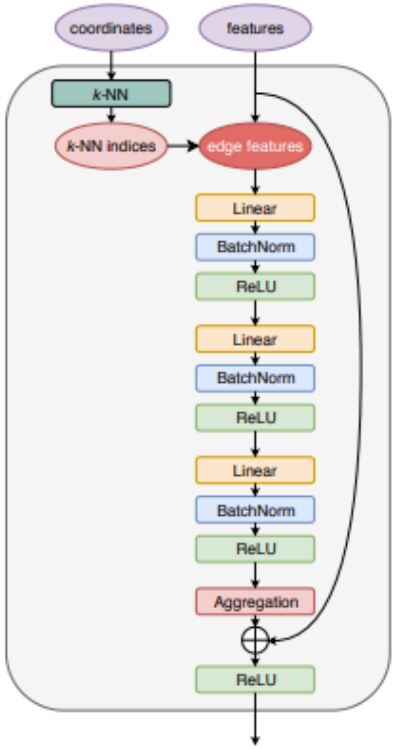
DeepAK8



arXiv:2004.08262

ParticleNet





arXiv:1902.08570

Demonstration setup

Task: Classification between Top-quark and QCD jets

Train DeepAK8 and ParticleNet in similar environment and make predictions, while tracking energy consumption and carbon emissions

Dataset: Benchmark top tagging dataset [arXiv:1707.08966] (training, validation, and test)

Hardware setup: Single GPU on lxplus-gpu

Software framework: weaver-benchmark (implemented in PyTorch)

Model architecture comparison:

Model	Parameters	Computational complexity
DeepAK8	349k	0.012 GMac
ParticleNet	577k	0.441 GMac

Training events: ~ 1.2 Million

Hyperparameters: batch-size, start learning rate, epochs = (164, $5e^{-3}$, 5)

Carbon tracking: with CodeCarbon

CodeCarbon

Track CO_2 from code running on hardware you control

- A lightweight, easy to use Python library
- Open source, free & community driven

pip install codecarbon

Methodology

Carbon dioxide emissions (CO_2e) can then be calculated as C * E , with

C = Carbon intensity* of the electricity consumed for computation: quantified as g of CO_2 emitted per kilowatt-hour of electricity.

E = Energy consumed by the computational infrastructure: quantified as kilowatt-hours.

* For this demonstration the global carbon intensity of Switzerland was used

How it works

A scheduler starts before training or inference, logs the energy consumption of the CPU, GPU, and RAM every 15 seconds by default, and stops when the training or inference process ends.

Energy and CO_2 emissions comparison of training workload

Metric	Value
Training ROOT files	13
Unique training events	1,211,000
Batch size	164
Batches per epoch	7,385
Number of epochs	5
Total batches	**36,925**
Nominal event presentations	**6,055,000**

DeepAK8

0.288 kWh Total energy

CO_2 emissions	**10.03 g CO_2e**
Recorded duration	**~41.2 minutes**
Energy per batch	**7.79×10^{-6} kWh**
Emissions per batch	**0.272 mg CO_2e**

ParticleNet

3.263 kWh Total energy

CO_2 emissions	**113.70 g CO_2e**
Recorded duration	**~476.6 minutes**
Energy per batch	**8.84×10^{-5} kWh**
Emissions per batch	**3.08 mg CO_2e**

Energy and CO_2 emissions comparison of training workload

Metric	Value
Training ROOT files	13
Unique training events	1,211,000
Batch size	164
Batches per epoch	7,385
Number of epochs	5
Total batches	**36,925**
Nominal event presentations	**6,055,000**

ParticleNet recorded energy consumption and carbon emissions were approximately **11.34 times** that of DeepAK8

DeepAK8

0.288 kWh Total energy

CO_2 emissions	**10.03 g CO_2e**
Recorded duration	**~41.2 minutes**
Energy per batch	**7.79×10^{-6} kWh**
Emissions per batch	**0.272 mg CO_2e**

ParticleNet

3.263 kWh Total energy

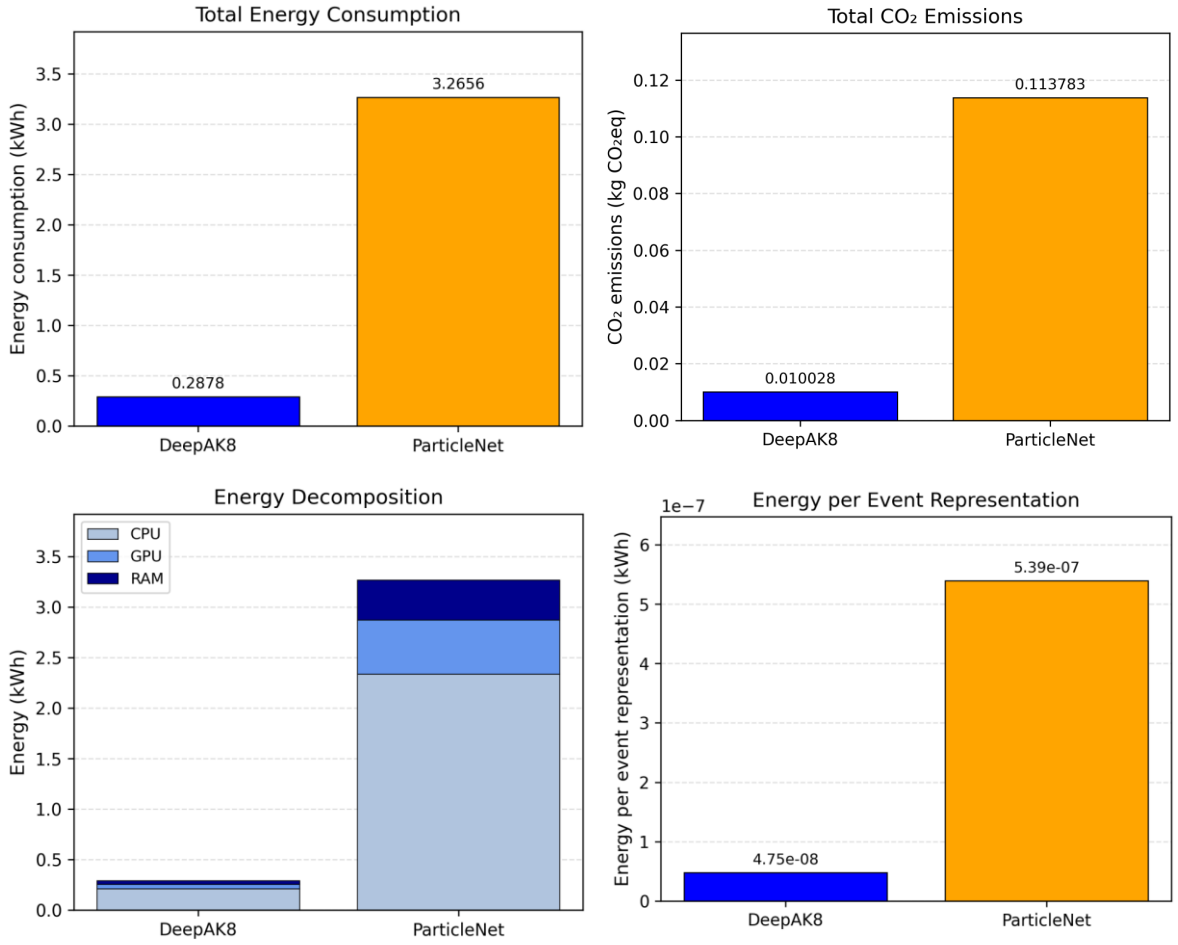
CO_2 emissions	**113.70 g CO_2e**
Recorded duration	**~476.6 minutes**
Energy per batch	**8.84×10^{-5} kWh**
Emissions per batch	**3.08 mg CO_2e**

Normalized comparison

Metric	DeepAK8	ParticleNet
Total energy	0.2878 kWh	3.2631 kWh
Total CO_2e	10.03 g	113.70 g
Energy per epoch	0.0576 kWh	0.6526 kWh
CO_2e per epoch	2.01 g	22.74 g
Energy per batch	0.00779 Wh	0.08838 Wh
CO_2e per batch	0.272 mg	3.079 mg
Energy per event presentation*	0.0475 mWh	0.539 mWh
CO_2e per event presentation	0.00166 mg	0.01878 mg

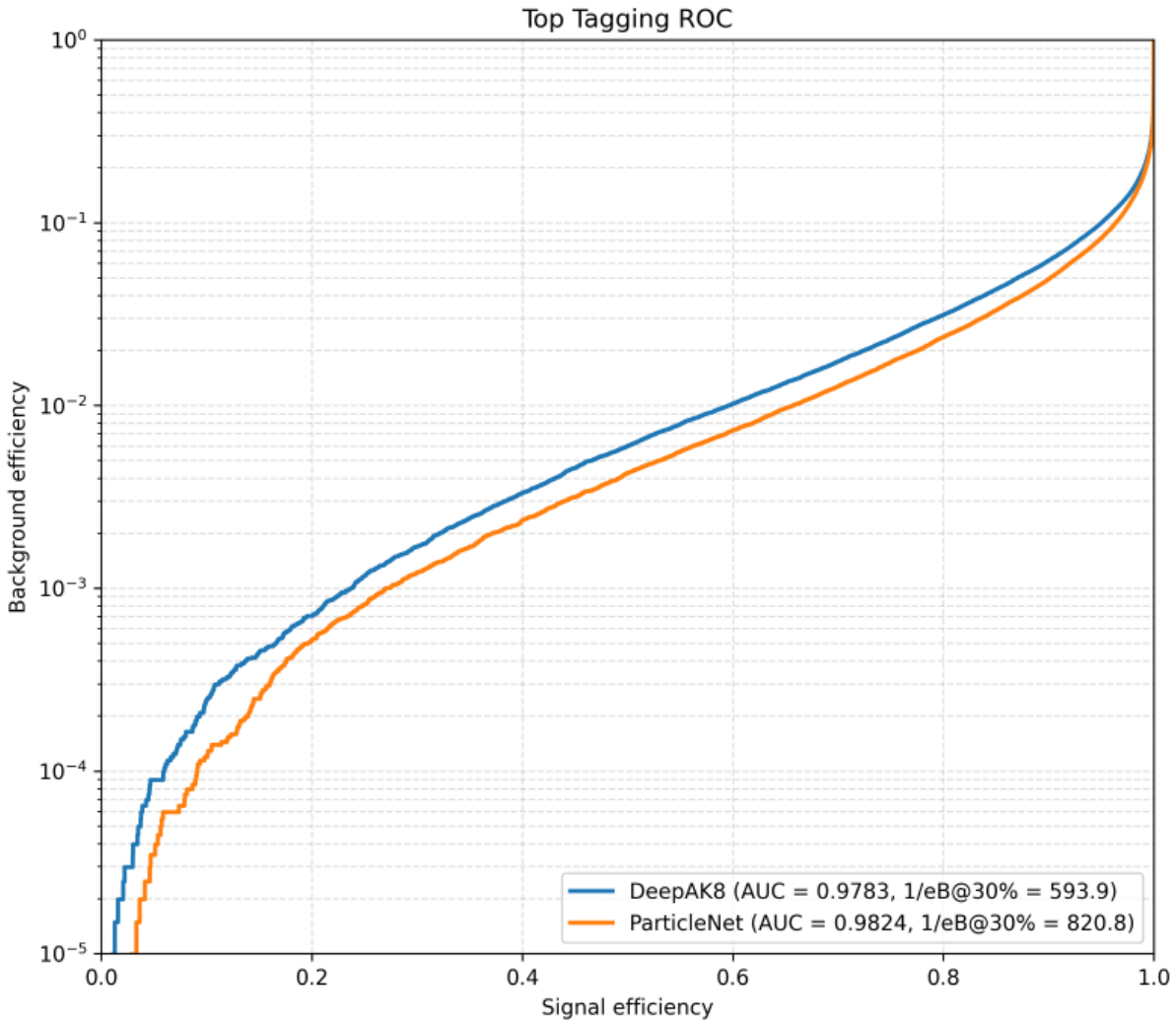
$$^{*}E_{\text{event}} = \frac{E_{\text{total}}}{N_{\text{events}} \times E_{\text{epochs}}}$$

🌱 Carbon Footprint Dashboard



Physics performance comparison

Test sample: 404,000 events per model; 201,914 background and 202,086 signal



At a signal efficiency of 30%:

- DeepAK8 background efficiency: approximately 0.168%.
- ParticleNet background efficiency: approximately 0.122%.
- Increase in background rejection **~ 38.2%**

Inference cost

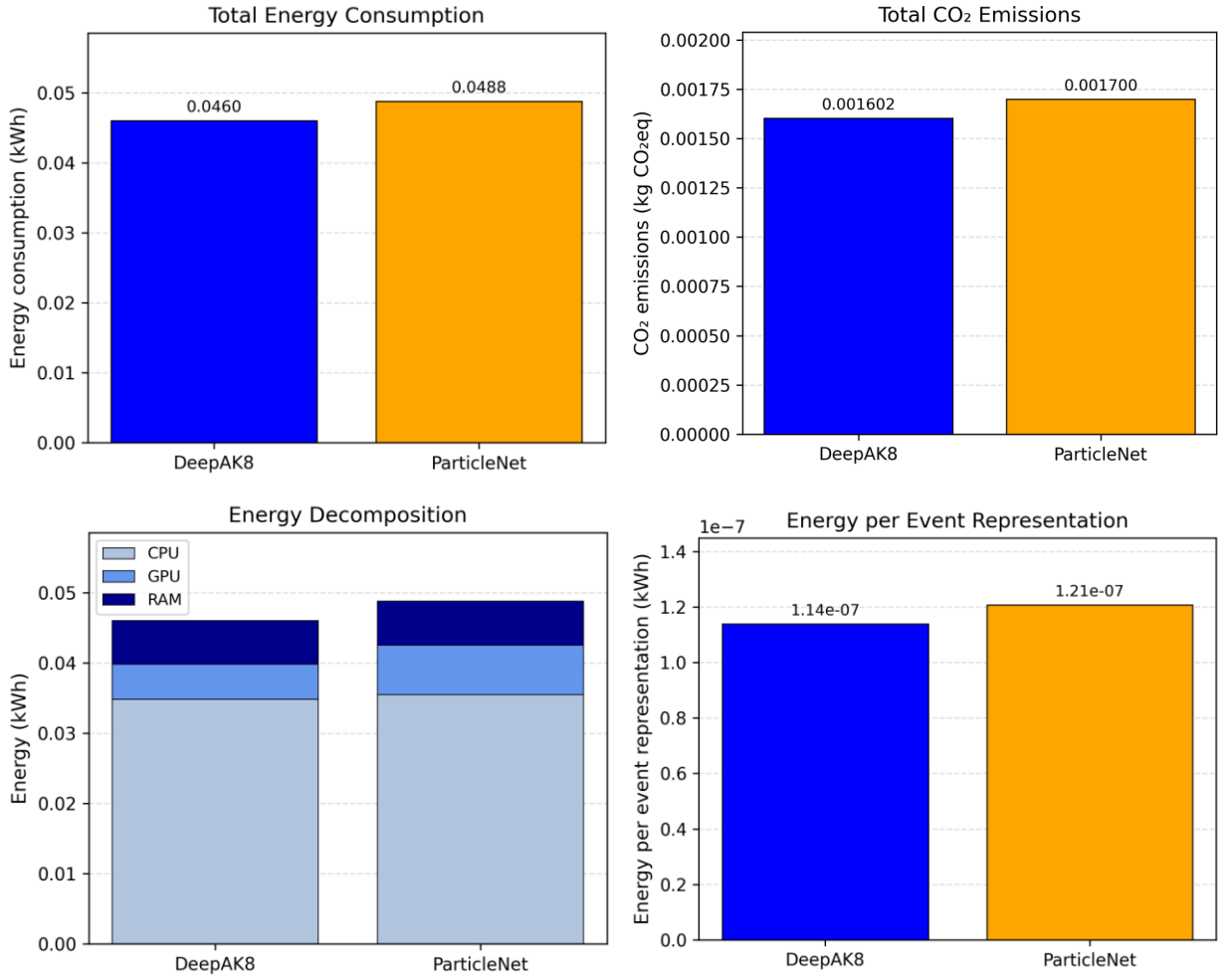
Test sample: 404,000 events per model; 201,914 background and 202,086 signal

Overall comparison:

Metric	DeepAK8	ParticleNet
Runtime	446.05 s (7.43 min)	453.25 s (7.55 min)
Total energy	0.04599 kWh	0.04878 kWh
Total emissions	1.602 g CO_2e	1.700 g CO_2e
Average power	371.17 W	387.44 W
Runtime per event	1.104 ms	1.122 ms
Energy per event	0.1138 mWh	0.1207 mWh
Emissions per event	0.003966 mg	0.004207 mg

ParticleNet was **1.6%** slower, consumed **6%** more energy, and generated **6%** more estimated emissions than DeepAK8

🌱 Carbon Footprint Dashboard



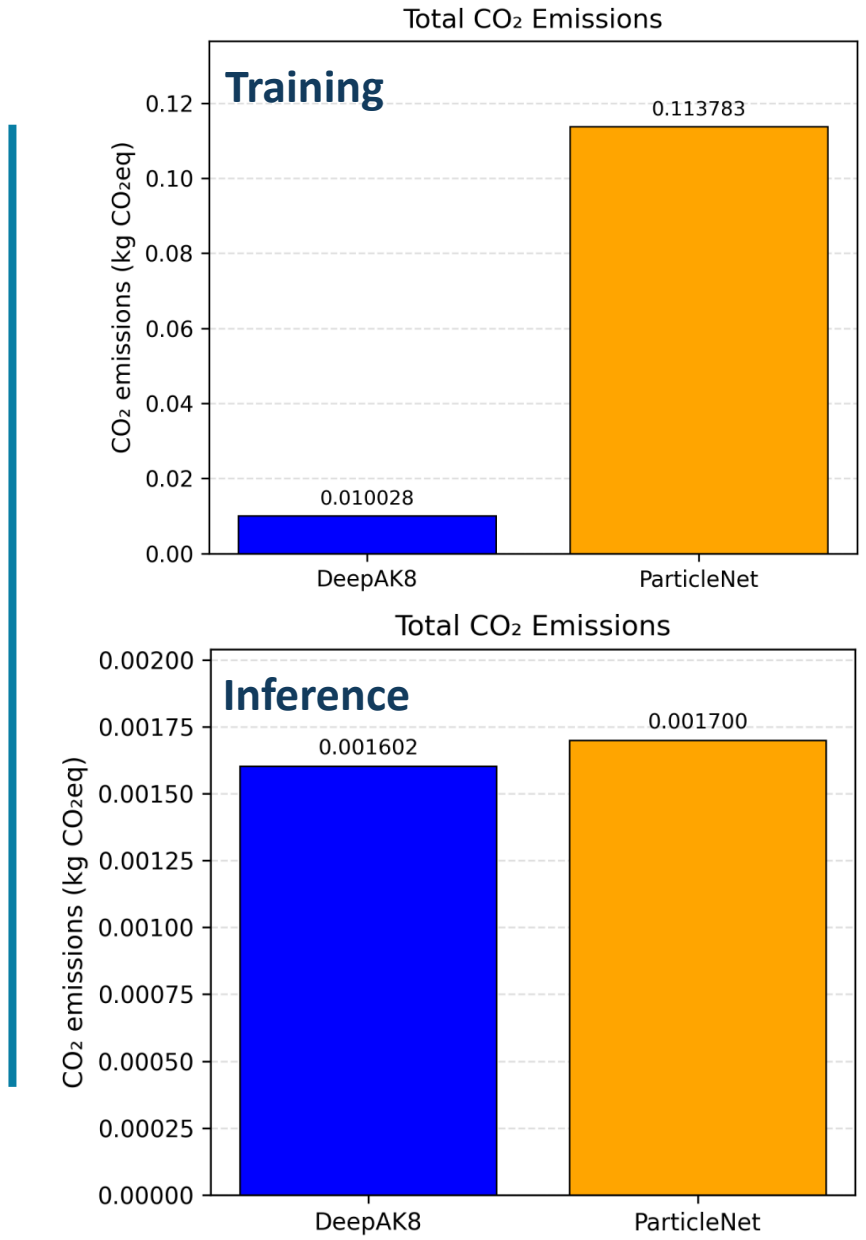
Summary

Performance: ParticleNet achieved better tagging performance and a higher background rejection.

Training: ParticleNet recorded energy consumption and carbon emissions were approximately **11.34 times** that of DeepAK8.

Inference efficiency: ParticleNet consumed approximately **~ 6%** more energy and produced **~ 6%** more CO_2e emissions than DeepAK8.

100M-event extrapolation: ParticleNet produces an estimated **24 g** additional CO_2e per **100 million events** under the benchmark conditions.



Shortcomings/limitations

Limited number of measurements: Results are based on individual training and inference runs; repeated runs can lead to varied results.

Hardware dependence: Energy consumption and emissions depend on the specific CPU, GPU (Tesla T4), and computing environment.

Extrapolation uncertainty: 100-million-event estimates assume linear scaling.

Carbon-intensity uncertainty: CERN's grid carbon intensity may differ from the Swiss average used by CodeCarbon, additional factors related to the data center not included.

CodeCarbon limitations: Machine-level tracking may not capture all power consumption accurately; distributed GPU utilization is not reliably measured.

Model efficiency: Measured energy and emissions reflect the complete execution environment, not the intrinsic efficiency of the models alone.

Conclusion & Outlook

Demonstrated a baseline for Green AI by reproducing CMS top-tagging benchmark

Used CodeCarbon while recognizing limitations

Future research direction: Develop methods to assess and reduce the carbon footprint of AI models deployed in particle physics.

Postdoctoral perspective: Explore how sustainability-aware strategies can be integrated into the development and deployment

