Scaling ML for HEP

2024 Chicagoland Computational Traineeship for High Energy Particle Physics

ML is Everywhere in HEP

- Particle-level simulation
  Validation
- Detector-level simulation
  Production
- Tracking, calorimeter clustering
  Validation
- Jet reconstruction, tagging, ID
  Production
- Data analysis/Inverse problems
  Ubiquitous

Why Scale it up?

- Large and/or Composite Generative Adversarial Networks (GANs) → 100s of hours to train
- Millions of measurements, graph embeddings → GPU memory
- Multi-scale "end-to-end" models: raw → particles
- Active learning, anomaly detection, uncertainty quantification

The HEP Advantage

Abundance of **labelled training data** from **high-fidelity simulation**

What are we scaling and potential impact of scaling

Scaling: training and inference (i.e., running your model after it has been trained)

Impact:

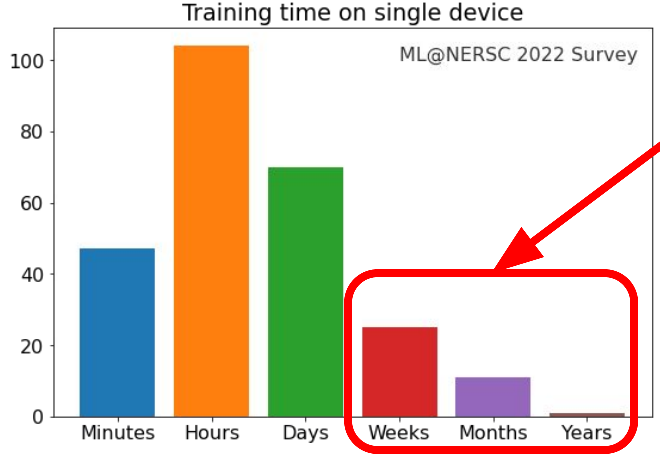
- Faster development cycle of computationally demanding models
- Better models via (Hyper Parameter) Optimization of models
- Enabling the use of computationally intense models (i.e., GANs)
- Faster processing of data with trained models

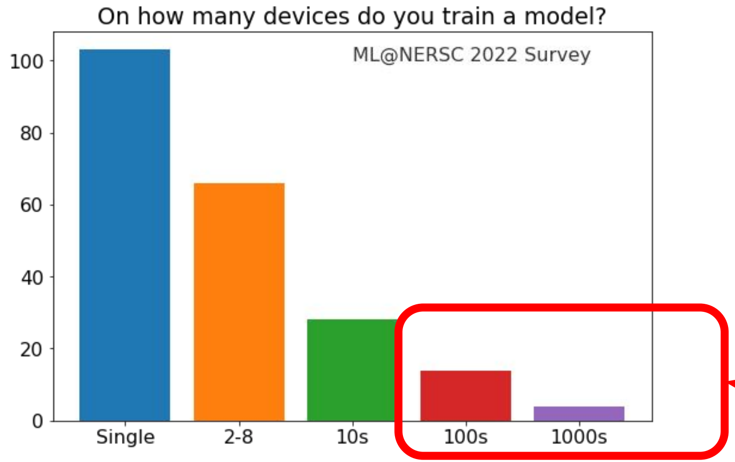
Enable maximum impact of ML on HEP

By mitigating computational limitations

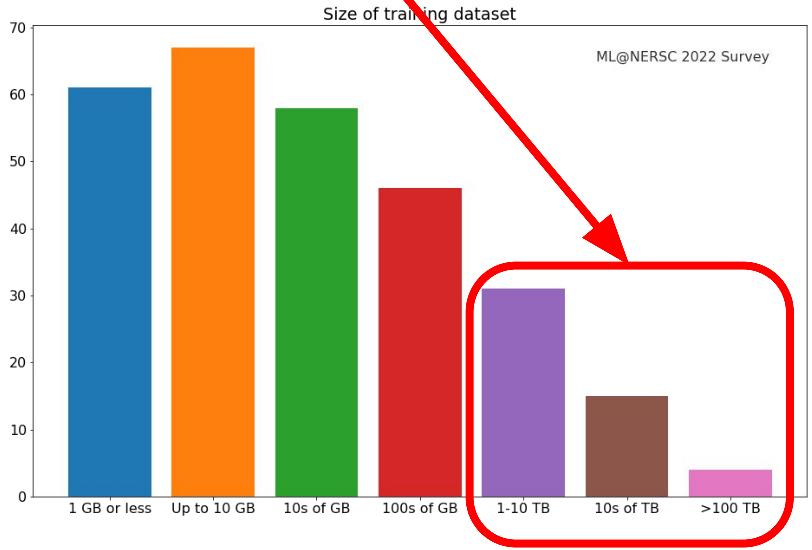
Scaling up Training

ML@NERSC 2022 survey





Large problems

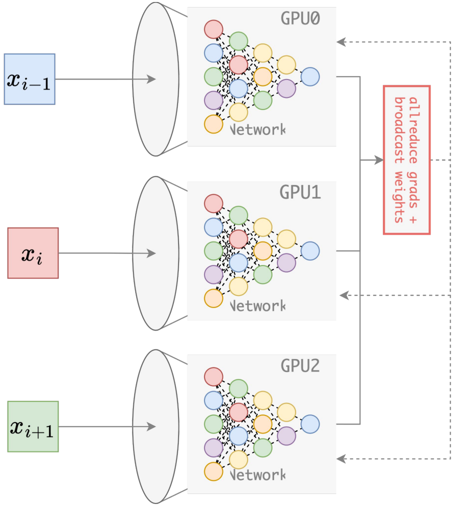


Require large scale training

Distributed training

Goal: Shorten the **development cycle** for large models **weeks→hours**

1. Distributed=across multiple nodes and multiple processors, even across nodes at different locations (i.e., across a network)
    - How you distribute the training load has a significant effect on computational performance
2. Several distributed training solutions exist, e.g.,
    - Horovod
    - LBANN (used to train multiple GANs simultaneously)

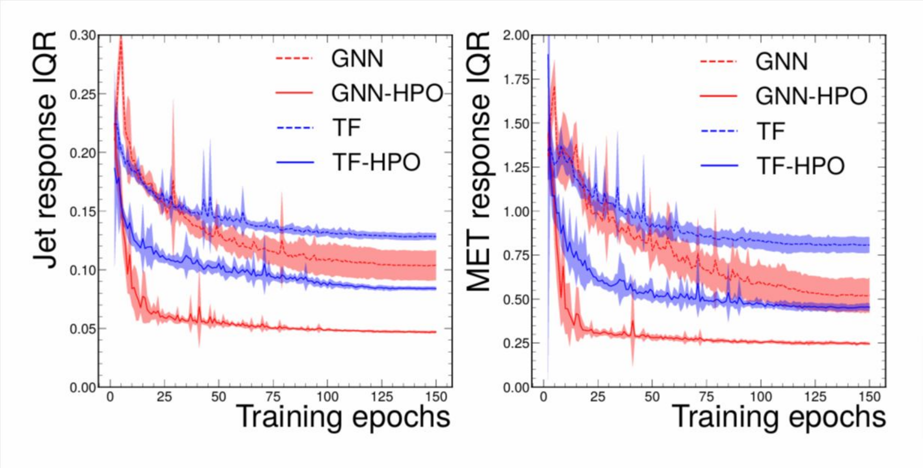


HOROVOD

Model and Hyper Parameter Optimization (HPO)

Why is it useful?

- Model and hyper parameter optimization can have dramatic effects on physics performance (ACAT talk on particle flow HPO)



Starting point: GNN and/or Transformer based models (particle flow, flavour tagging, tracking, jet reco...) and optimize

Both industry tools (e.g., RayTune) and HEP workflow-based approaches (PanDA/iDDS) exist

Weights & Biases

SIGOPT

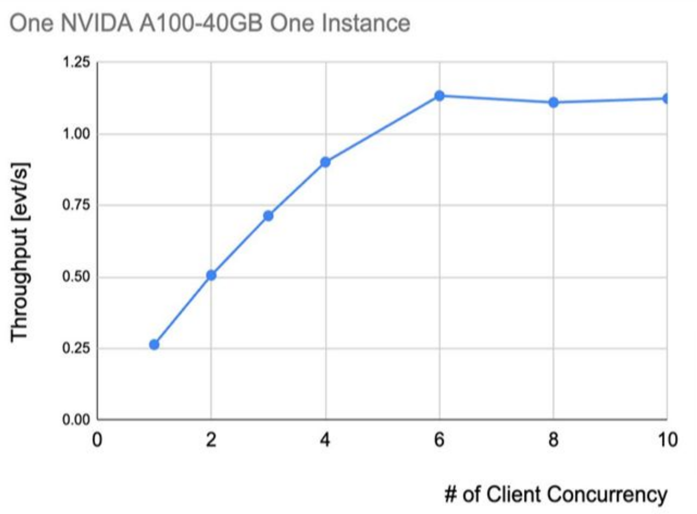
RAY tune

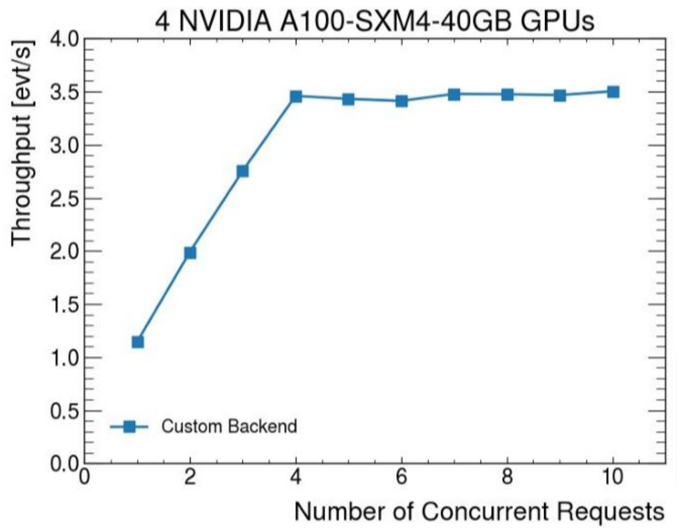
DeepHyper

Inference as a Service (IaaS)

Why is it useful?

- Simple interface that handles complex interaction between nodes, availability of different resources, even across network
- Fully utilize GPU
- Scale out to multiple GPU and nodes
- Example of existing IaaS efforts: Exa.TrkX and [ACTS as a Service]()





High Performance Computers (HPCs)

Opportunity: Enable/encourage HEP ML practitioners to **think big**

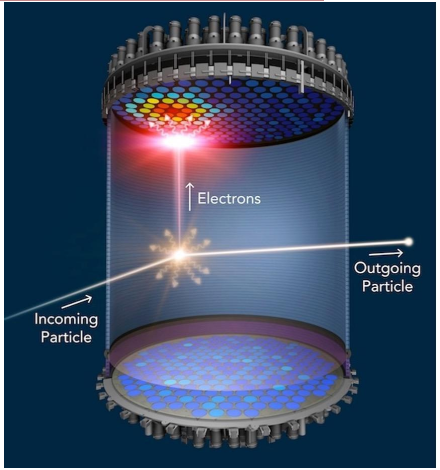
- HPCs are large systems with GPU and CPU
  - Examples: Aurora at Argonne, Frontier at Oak Ridge, NERSC at Berkeley
- HPCs can offer immense scales: we are in the exa-era of computational power
  - Aurora has 21K CPUs and 64K GPUs
- Most are optimized for ML workflows

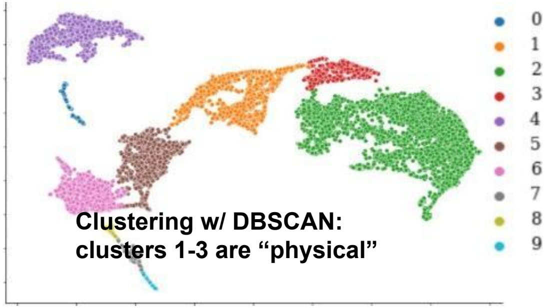
Compute intensive ML models examples

- **Simulation**
  - FastCaloGAN -> a lot of human intervention to make the GANs converge. LBANN has multi-generator, multi-discriminator framework that is only possible with scaling. Cosmological simulations, DES adversarial domain adaptation
- **Reconstruction**
  - Particle/Jet ID, e.g., flavor tagging
  - Tracking
- **Analysis**
  - Simulation-based inference, unfolding (inverse problem)
  - LSST image processing
- **Resource constrained models**
  - Size of model vs performance
    - E.g., quantization reduces model size and improves inference time but increases training time

Scaling Up Anomaly Detection in Lux-Zeppelin

- World's most sensitive Dark Matter search, as of last year 🎉
- 1,000-live-days run planned, expected **5 billion events** or **5 PB of raw data**, overwhelmingly dominated by background
- Anomaly detection has been attempted in LZ with some success, on a subset of the data (detector & simulated).
- Identified 2 types of anomalies: "unphysical" detector events and problems with the reconstruction algorithm, Anomalies becoming rare, **down to 1%**. .
- **Next step**: apply variational autoencoders to the **full dataset** (at the waveform level), to reach **10^9 sensitivity**.
- Challenge: **train VAE on the entire 5PB dataset**, to tackle unknown and/or unmodeled backgrounds.





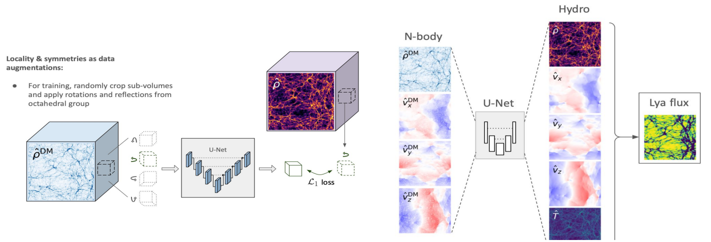
Scaling Up Cosmology Simulations

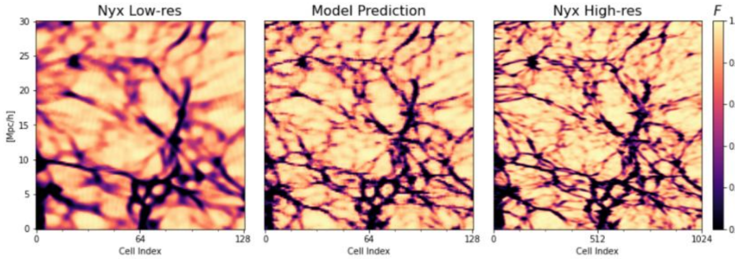
Solving cosmology inverse problems using full-physics simulations costs **10-100s millions of hours** on HPC systems.

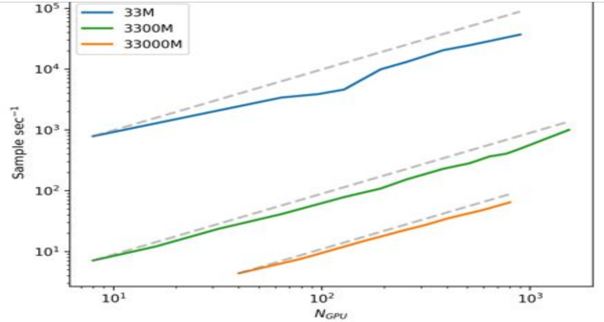
Instead, use partial/full-physics simulations to train

- 3-D U-Net CNN to map N-body simulations to hydro fields or to higher-resolution simulations
- Convolutional VAE to generate jointly
  - accurate hydrodynamical fields
  - reasonable variance estimates
- U-Net generative models to improve the accuracy of low-resolution simulations and use for covariance estimation
- These models trained and run on 4-GPU node

New approach using sharding/replication scales up to **thousands of GPUs**, allowing use of **transformer**-based models

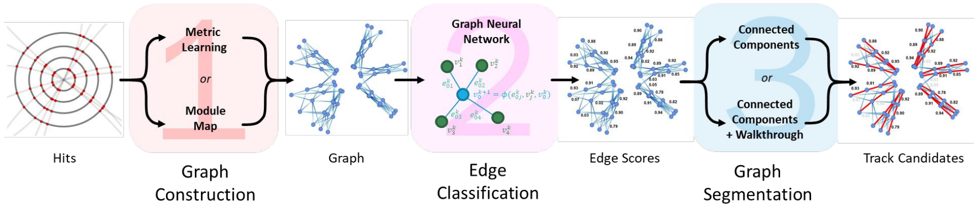


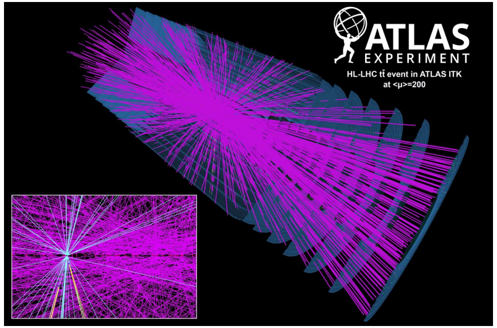




Test run of a transformer-based model (3.3 billion parameters) on Polaris with up to 1696 GPUs

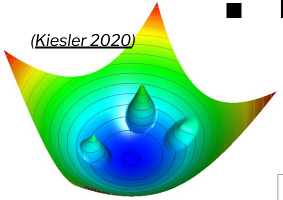
Scaling Up Particle Tracking



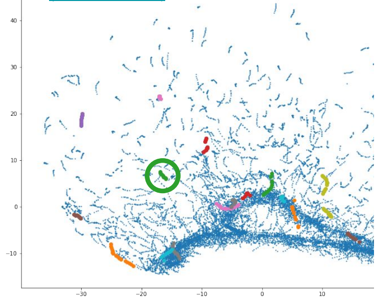


Tracking most compute-intensive reconstruction algorithm for ATLAS, CMS, DUNE
Graph Networks deliver competitive performance across multiple detectors

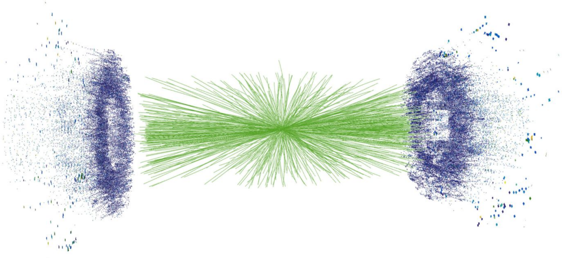
- Resource-intensive **hybrid GPU pipeline**
- **Several weeks** required to train the full pipeline
- **Memory** limited, during **training** *and* **inference**
  - Distributed training needed to maximize physics performance



(Kiesler 2020)



(Murnane -2023)

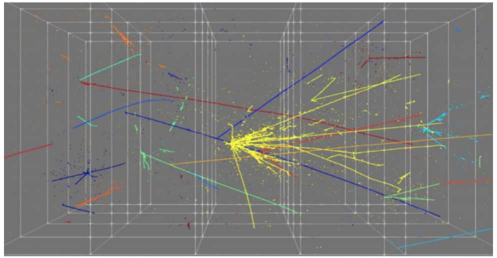


Several R&D projects focused on
end-to-end rawdata→particle reconstruction

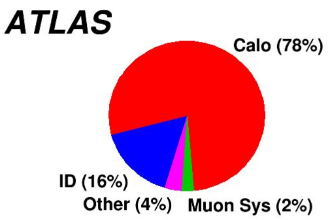
- multiscale hierarchical GNNs
- object condensation networks

Current **compute and memory-limited**

- Would need to **scale up** resources **10-100x**



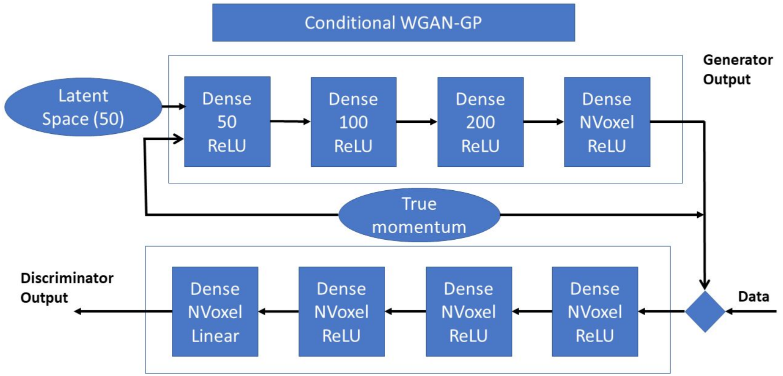
Scaling Up Calorimeter Simulation

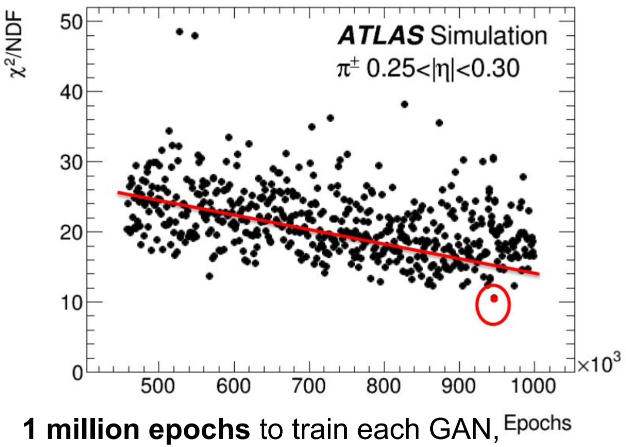


Calorimeter simulation in dominates G4 CPU usage

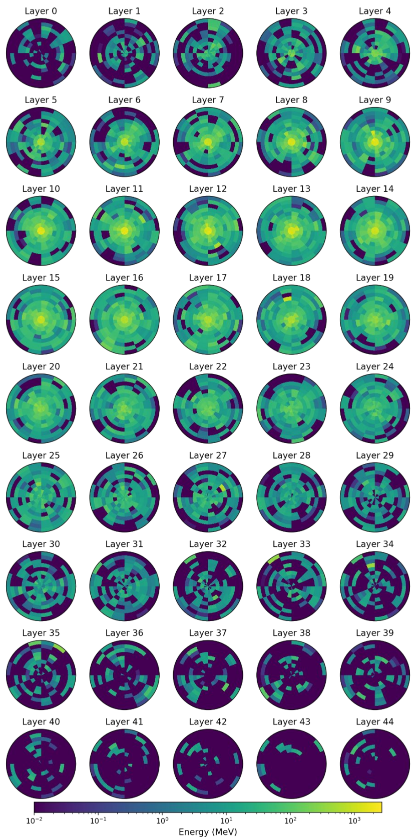
FastCaloGAN

- First large-scale DNN to run in production in ATLAS
- A combination of **300 WGAN** trained to simulate the response of pions, electrons, and photon in an [energy, η] bin
- **~100 GPU days total to train**





1 million epochs to train each GAN, looks like it could have used **10x more!**



Conclusion

- ML models are getting larger and tackling more complex problems
- Developing and optimizing models is becoming computationally expensive
  - Development cycle should not be stunted by computing
- HPCs offer an opportunity to scale training and inference
- Facilitating scaling of ML can have impact on our science

Thanks to

Paolo Calafiura

Wahid Bhimji,

Steve Farrell,

Zarija Lukic,

Maria Elena Monzani

for their contributions. All mistakes are mine!