SML
Scaling up ML Applications

Paolo Calafiura
for HEP-CCE

LBL AHM
July 22, 2024

One Slide Summary

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?

- Correlations, Design of experiments
- Large and/or Composite GANs → 100s of hours to train
- Millions of measurements, graph embeddings → GPU memory
- Model ensembles, collectively billions of parameters
- Active learning, anomaly detection, uncertainty quantification

The HEP ML Advantage

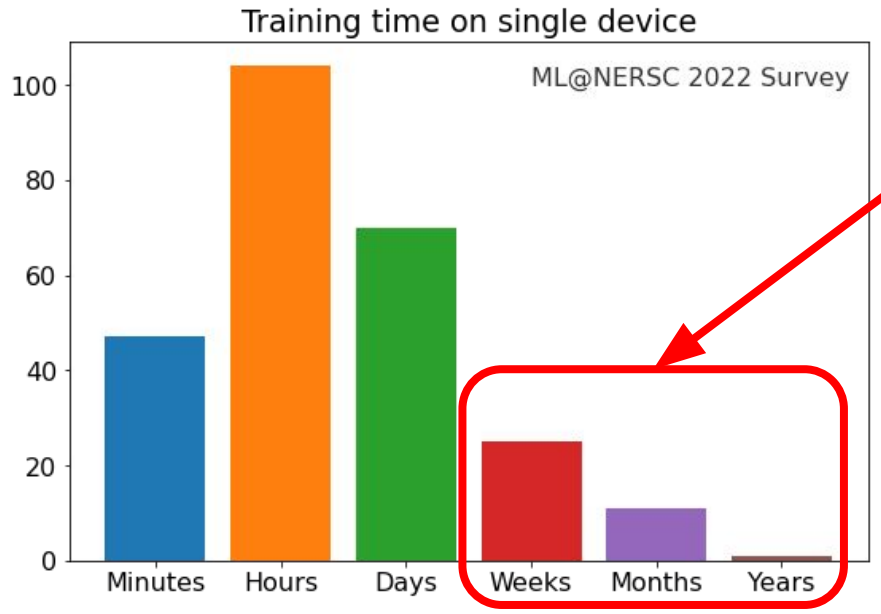
Abundance of **labelled training data** from **high-fidelity MC simulation**

HEP-CCE Goals

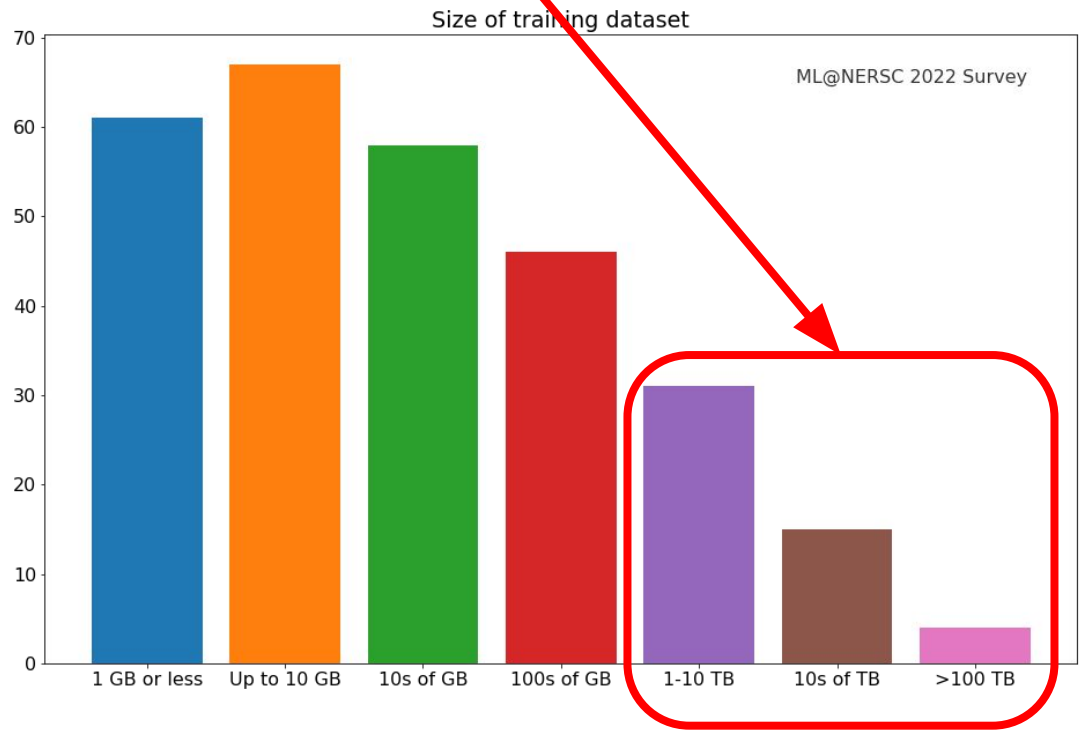
- Study **performance of ML pipelines** on heterogeneous resources
- Shorten the **development cycle** for large models **weeks→hours**
- **Democratize access** to HPC ML resources
  - Enable/encourage HEP ML practitioners to **think big**

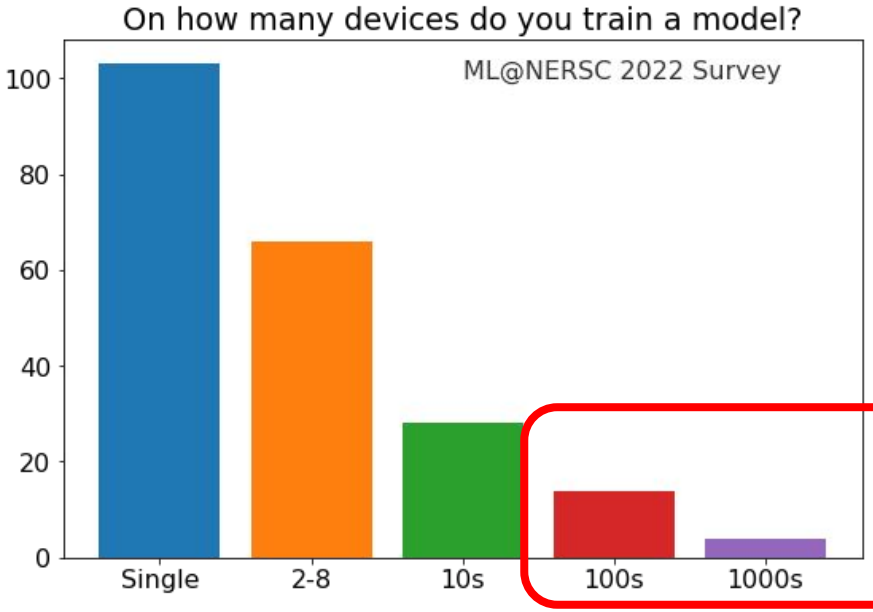
Scaling up Training

ML@NERSC 2022 survey



Large problems

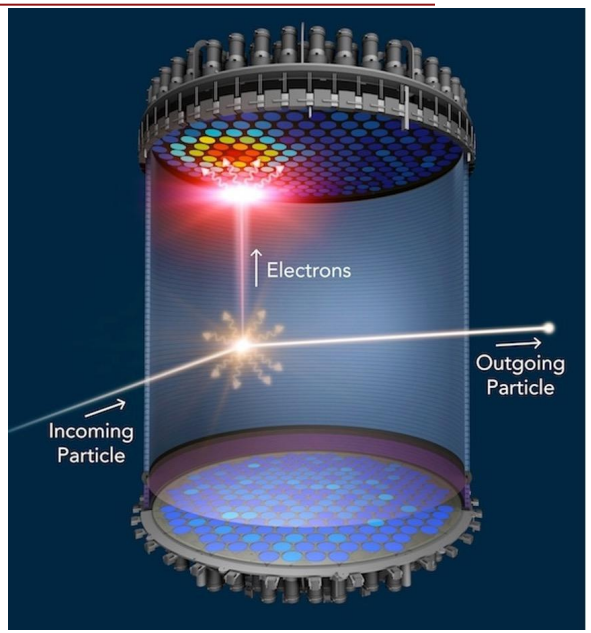


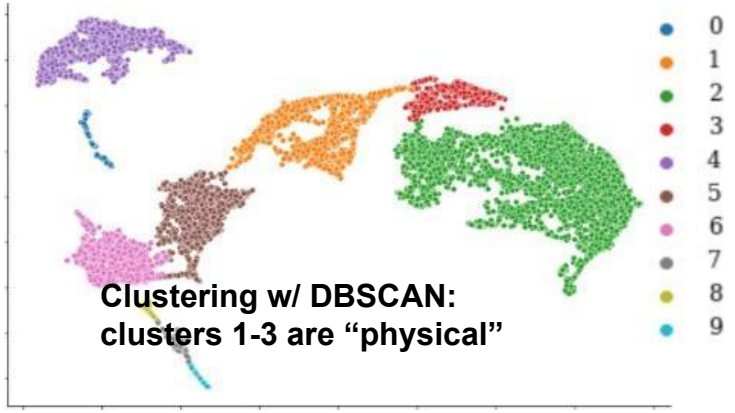


Require large scale training

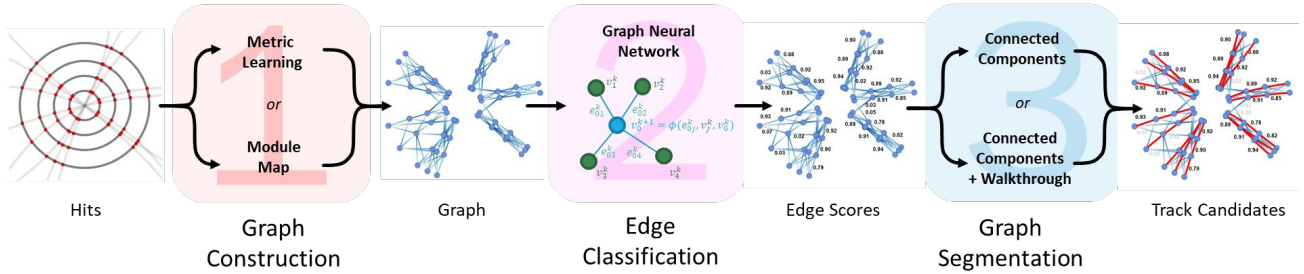
Scaling Up Anomaly Detection in Lux-Zeppelin

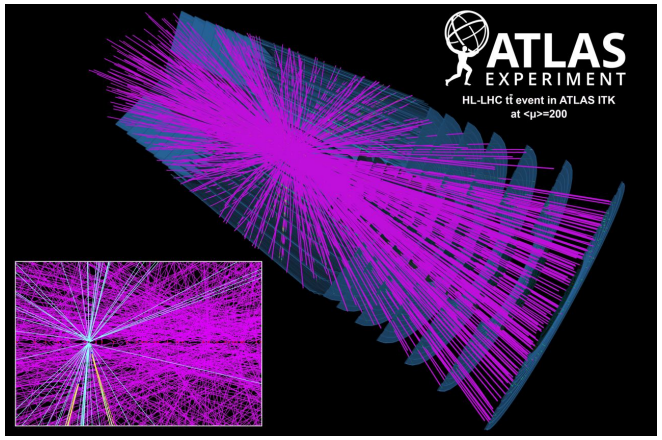
- World's most sensitive Dark Matter search, as of last year 🎉
- 1,000-live-days run planned, expected **5 billion events** or **5 PB of raw data**, overwhelmingly dominated by background
- Anomaly detection has been attempted in LZ with some success, on a subset of the data (detector & simulated).
- 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 Particle Tracking

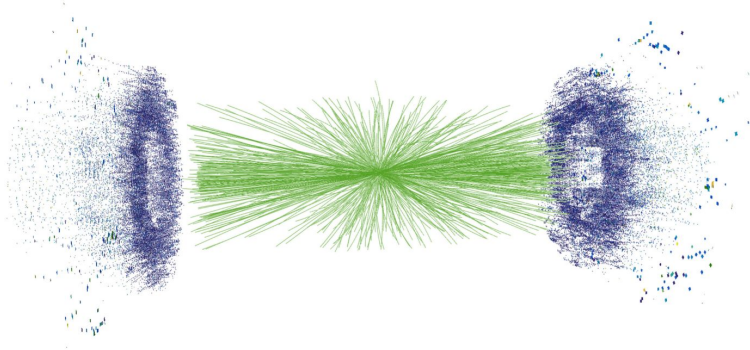




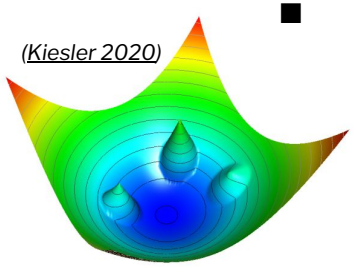
Tracking most compute-intensive reconstruction algorithm for ATLAS, CMS, DUNE

Graph Networks deliver competitive performance across multiple detectors

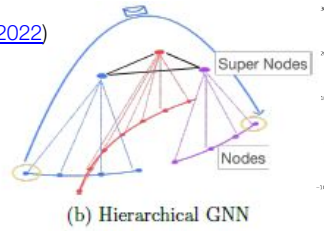
- Resource-intensive **hybrid GPU pipeline**, good PAW testbed
- **Several weeks** required to train the full pipeline
- **Memory** limited, during **training** *and* **inference**
  - Distributed training needed to maximize physics performance



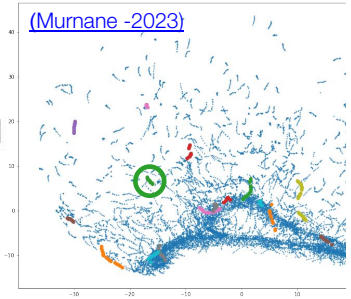
(*Kiesler 2020*)



(Liu - 2022)



(Murnane -2023)



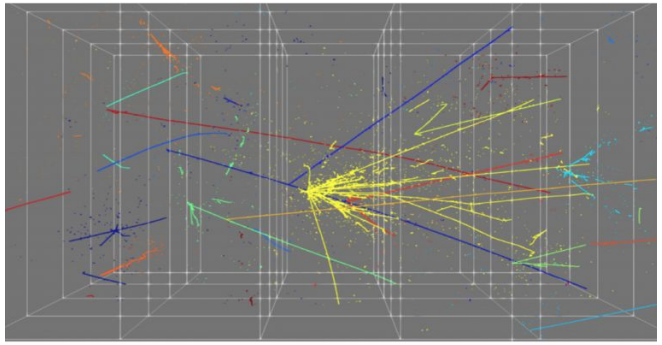
Several R&D projects focused on

end-to-end rawdata→particle reconstruction

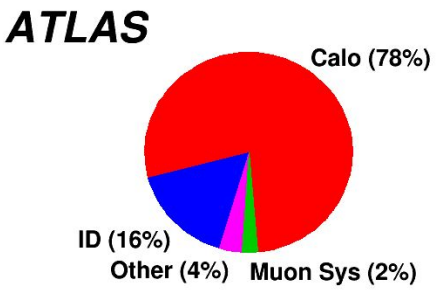
- multiscale hierarchical GNNs
- object condensation networks

Current c**ompute and memory-limited**

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



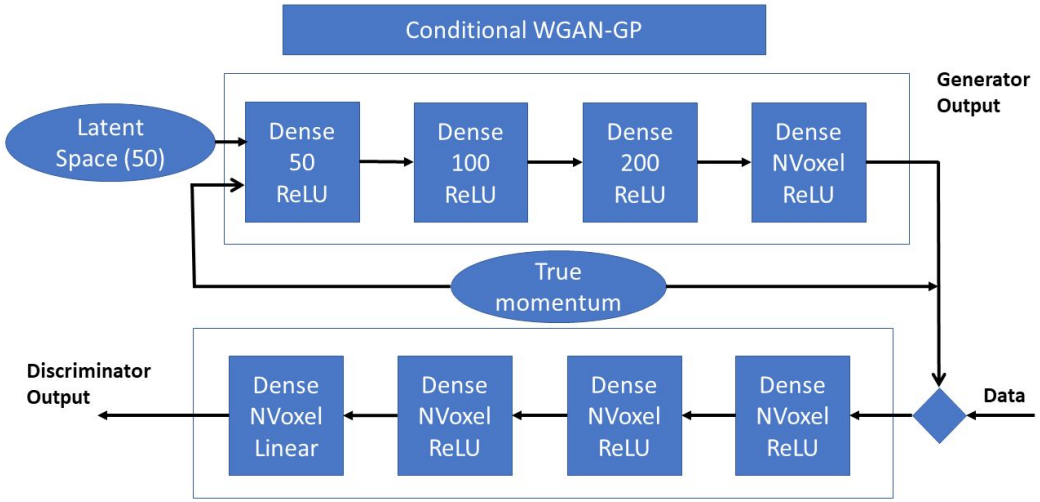
Scaling Up Calorimeter Simulation

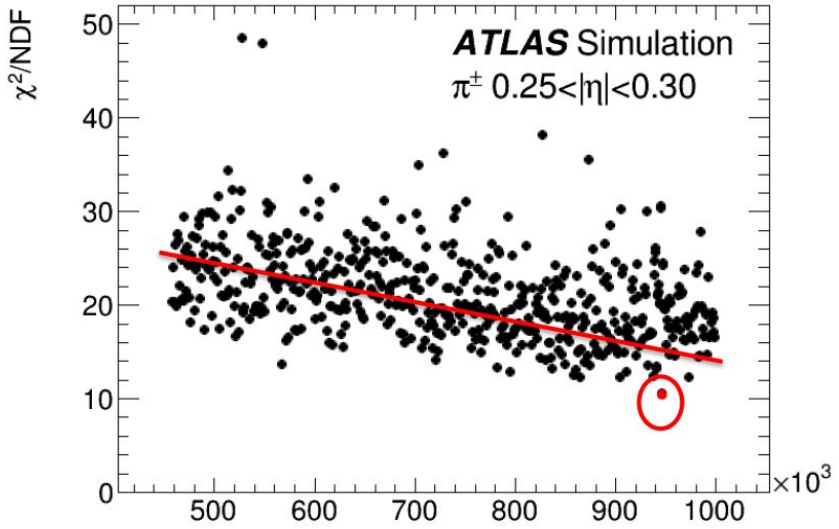


Calorimeter simulation in dominates G4 CPU usage

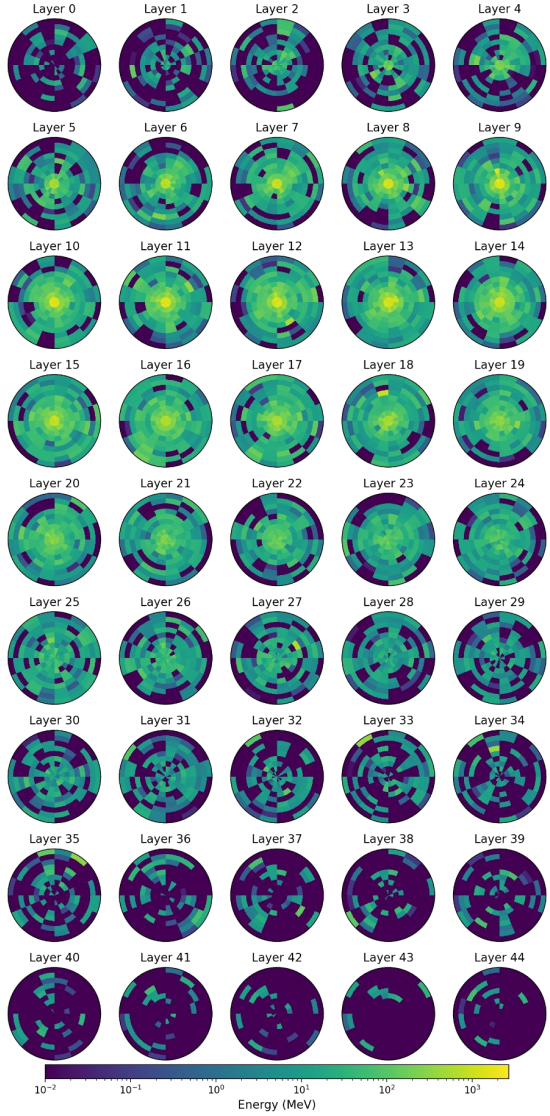
FastCaloGAN

- 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!**



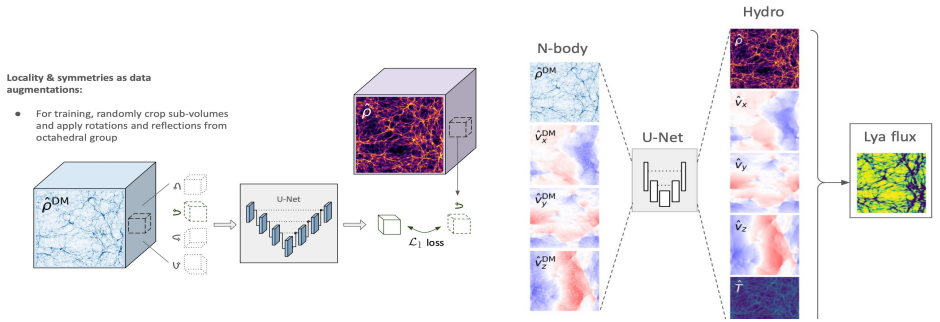
Scaling Up Cosmology Simulations

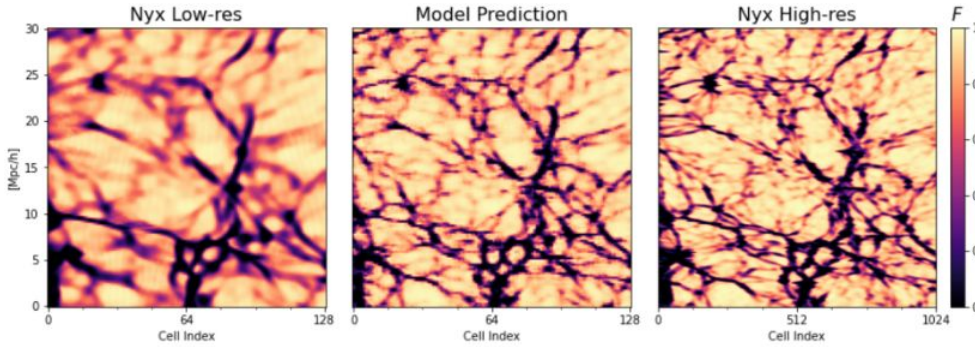
Solving cosmology inverse problems using full-physics simulations costs **10-100s millions of hours** on HPC systems.

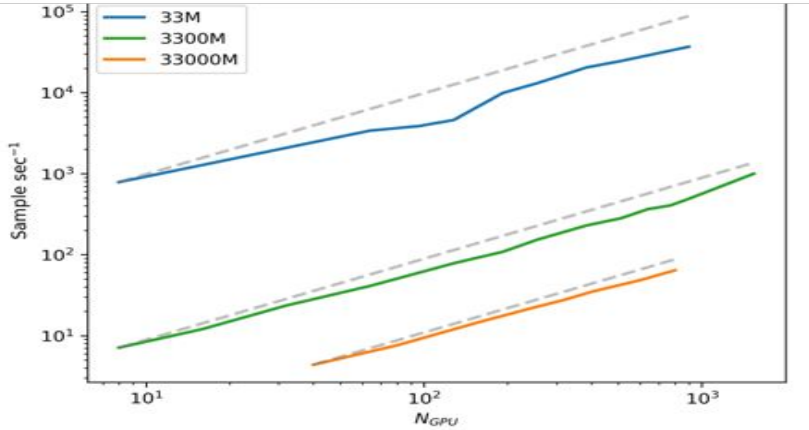
Instead, use partial/full-physics simulations to train

- 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

SML Year 1 Milestones

1. **Identify target ML models in collaboration with experiments**
2. *Port, train, and run at least two target models on two different HPC systems*
3. *Compare two data parallel training solutions for at least one target model*
4. *Compare two hyperparameter optimization tools on at least one target model*
5. Setting up a prototype Inference as-a-service platform on at least one DOE HPC system

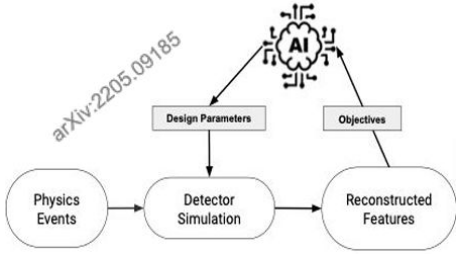
Email list: CCE-SML@anl.gov

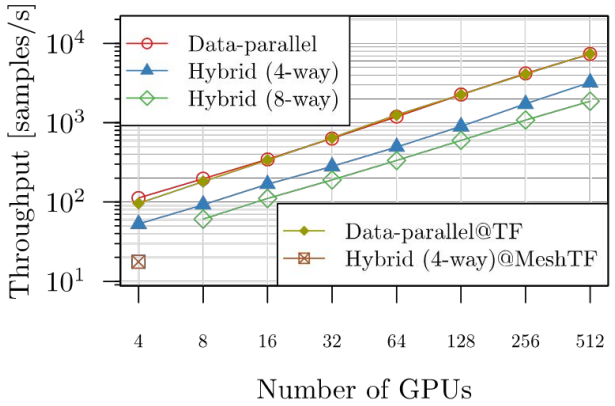
First meeting was in January, eight meetings so far

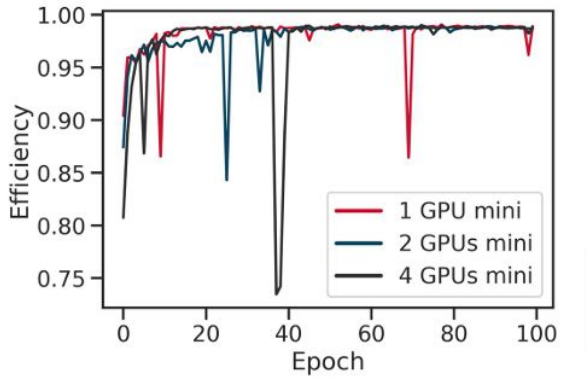
Goal is to meet every other week, everyone welcome

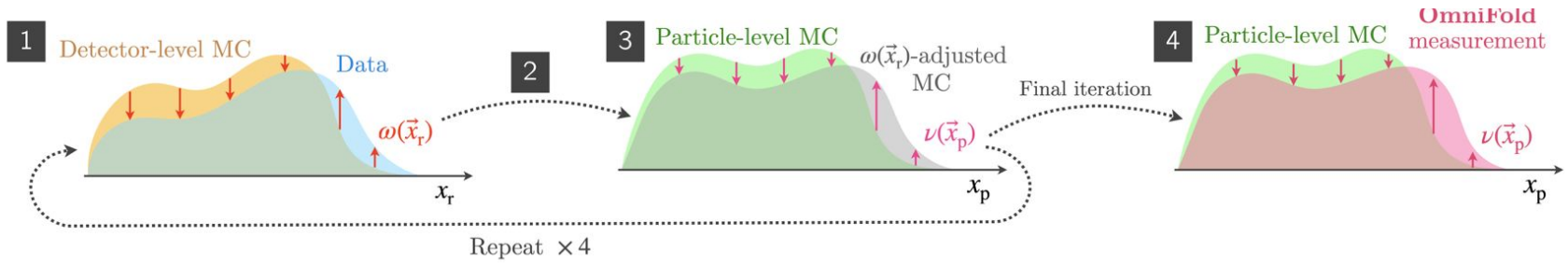
Highlights from SML bi-weekly meetings

- Wen Guan: HPO and detector optimization using iDDS (slides)
- Peter Nugent: Introduction to LBANN (slides)
- Alina Lazar: Scaling up Distributed GNN Training (slides)
- Ben Nachman: Unbinned Unfolding with Omnifold (slides)
- Aishik, Jay, and Rafael:
  Neural Simulation Based Inference on HPCs (notes)









Model Selection Criteria

Maturity of model

Should be (near) production ready

Representative

Should be a model that is commonly used and addresses needs that are common among experiments and will likely be needed in the long term

Impactful scaling

Scaling the model or performing an HPO will improve the model significantly

Person power

Should be a model with names attached to it. People who want to work on scaling that particular model

2 different scaling tasks:
Inference as a service (IaaS)
And
Training

ML models under consideration

Categories:

- **Simulation (training and maybe IaaS)**
  - 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**
  - Flavor tagging (scaling focused on training):
  - Tracking (training and IaaS)
  - DUNE reconstruction (training and IaaS)
- **Analysis (training and IaaS)**
  - **Simulation-based inference** (**upcoming talk**)
  - LSST image processing
- **Resource constrained (FPGA/ASIC) model (training and maybe Iaas)**
  - Size of model vs performance
  - Quantization slows down training
  - Smart pixels (6-layer CNN) takes 3 days (tracking related)

SML Year 1 Milestones

1. Identify target ML models in collaboration with experiments
2. *Port, train, and run at least two target models on two different HPC systems*
3. *Compare two data parallel training solutions for at least one target model*
4. *Compare two hyperparameter optimization tools on at least one target model*
5. **Setting up a prototype Inference as-a-service platform on at least one DOE HPC system**

Email list: CCE-SML@anl.gov

First meeting was in January, eight meetings so far

Goal is to meet every other week, everyone welcome

Year 1: Inference as a Service (IaaS)

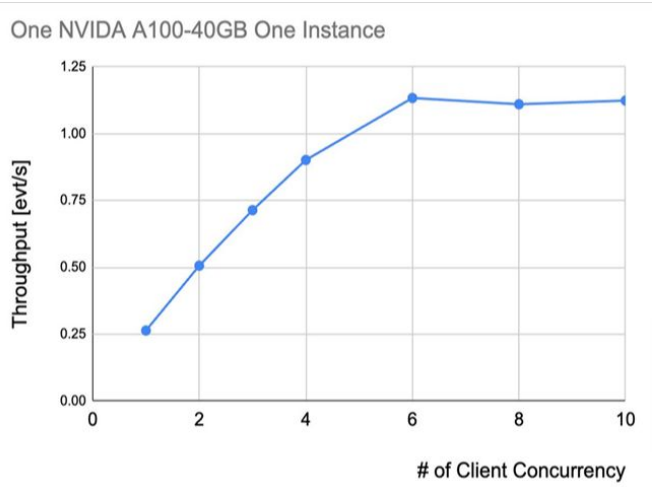
Why is it useful?

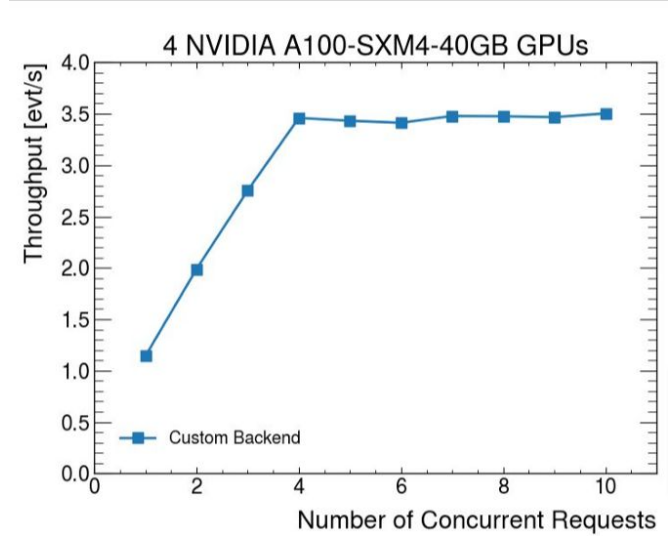
- Clean interface to abstract host/device communication, even across WAN (study performance impact, usability, security on DOE HPC systems)
- Fully utilize GPU
- Scale out to multiple GPU and nodes

What is SML role?

- Build on existing IaaS efforts (including Exa.TrkX and ACTS as a Service)
- Deploy IaaS on DOE HPCs (Perlmutter as a start)

Collaborate with **PAW** on optimization&portability?

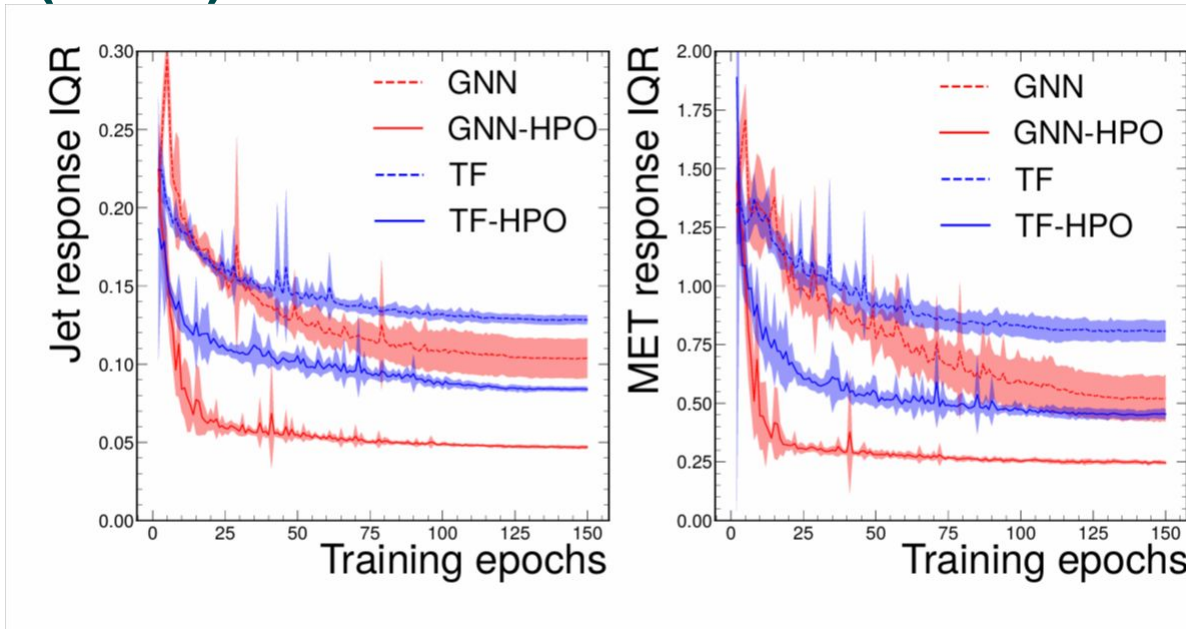




Year 1½: Hyper Parameter Optimization (HPO)

Why is it useful?

HPO can have dramatic effects on physics performance (ACAT talk on particle flow HPO)



Start with GNN and or Transformer based models (particle flow, flavour tagging, tracking, jet reco...)

Compare industry tools like RayTune to HEP workflow-based approaches (PanDA/iDDS)

Longer-term: SML Roadmap

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

1. Select a handful of large-scale “testbed” models representative of HEP ML architectures across the participating experiments.
   - Port and measure baseline single-node performance on DOE HPC systems
2. Select distributed training solutions using the testbed models to define requirements
3. Study and optimize scaling of computing performance on DOE HPC systems (compare, where relevant, to experiment-owned resources)
4. Asses impact (and cost) of HPO with different algorithms

Collaboration opportunities

1. Optimize data delivery to distributed training workflows
2. Study performance and scalability of hybrid ML pipelines on HPC systems
3. Understand scalability and system+workflow implications of IaaS.

Roadmap, continued

Goal: Democratize access to HPC ML resources,
Enable/encourage HEP ML practitioners to **think big**

Distributed training is considered “hard” by most ML practitioners.

1. Help experiments define a **distributed training (*and inference)* HEP ML platform**,
   - deployable across ASCR facilities and experiment resources.
2. Develop a **distributed training and optimization curriculum for HEP practitioners**,
   - Leveraging existing ASCR and HEP ML material and expertise.
   - Based on CCE testbed models, training and optimization tools.
3. Provide fellowships and mentorship opportunities to students and early career researchers.

The FAAST Opportunity

DOE is preparing for a multi-billion dollar AI initiative. Big picture is to scale up Scientific ML beyond Exascale. Four pillars:

Data	Platforms	Models	Applications
Create vast repositories of curated scientific data to train, test, and validate next gen ML	*Next gen HPC, networks, and algorithms(?)*	*learn to speak the languages of physics, chemistry, and biology,*	*uniquely tailored models into strategic and critical application spaces that would otherwise be underinvested.*
HEP already has “vast” amounts of open and proprietary data	Looking forward to using them...	HEP is starting to look into foundation models. Research opportunities e.g. in graph foundation models	Sure, we got some juicy ones. See e.g. the SBI talk

Let’s discuss tomorrow afternoon what can we do to help HEP prepare for this unique opportunity.

Thanks to

Walter Hopkins (SML tech lead)
Wahid Bhimji,
Steve Farrell,
Alina Lazar,
Zarija Lukic,
Maria Elena Monzani
Aishik Ghosh
Rafael Coelho Lopes de Sa
Wen Guan
Ben Nachman
Peter Nugent

for their contributions. All mistakes are mine!

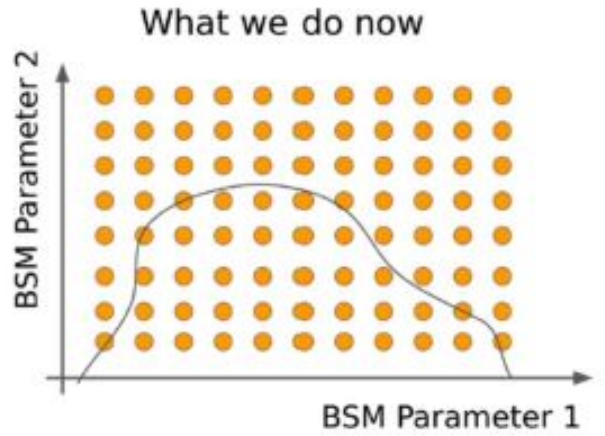
Backup

Scaling up HEP ML workflows

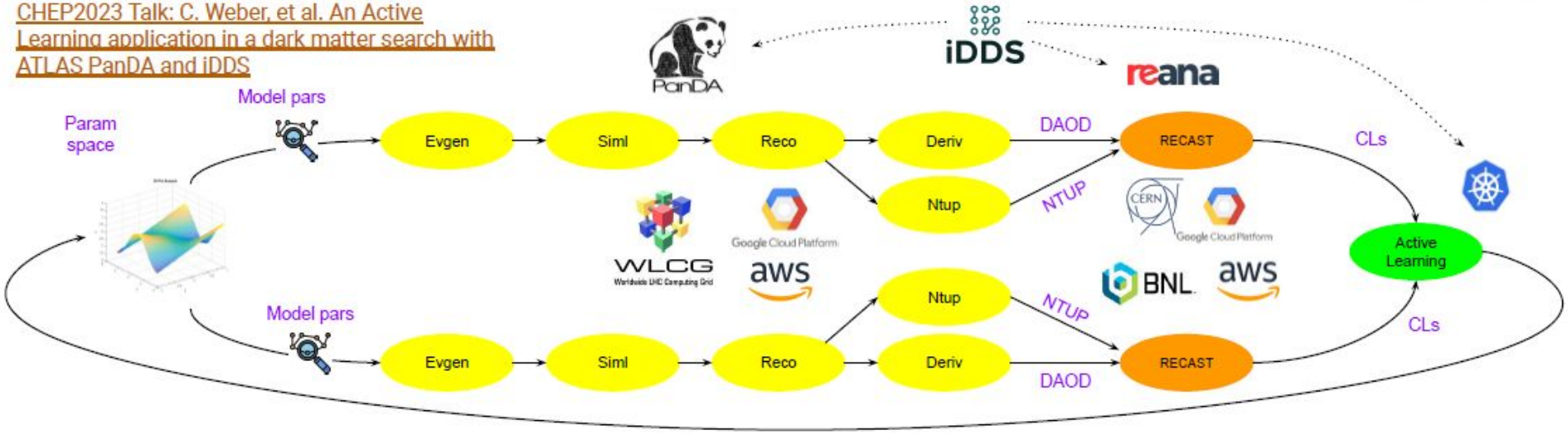
HEP Advantage → Challenge

Massive amounts of high accuracy labelled training data available

- **JIT data delivery** → Connection to Storage Optimization area,
  - relevant experience in ATLAS & CMS.
- Model-directed simulation "in the loop" → **active learning**
  - Study started during Phase 1 in collaboration with ATLAS ([CHEP23 talk](#))



CHEP2023 Talk: C. Weber, et al. An Active Learning application in a dark matter search with ATLAS PanDA and iDDS



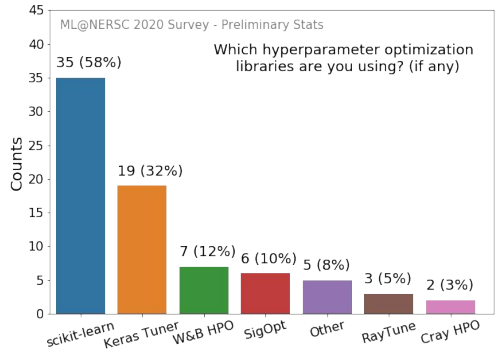
Scaling up Hyperparameter Optimization (HPO)

Grid searches for the optimal model architecture and configuration are at best an expensive exercise.

- Prohibitively expensive with more than a couple of free hyperparameters

Dozens of algorithms, implemented by standard libraries.

- Many scale well to multi-GPUs and higher dimensional searches
  - **Great match for HPC resources**



Not as popular in HEP ML community as one may expect

- 4 papers out of ~1300 in the HEPML LivingReview

Can CCE facilitate adoption?

Weights & Biases

SIGOPT

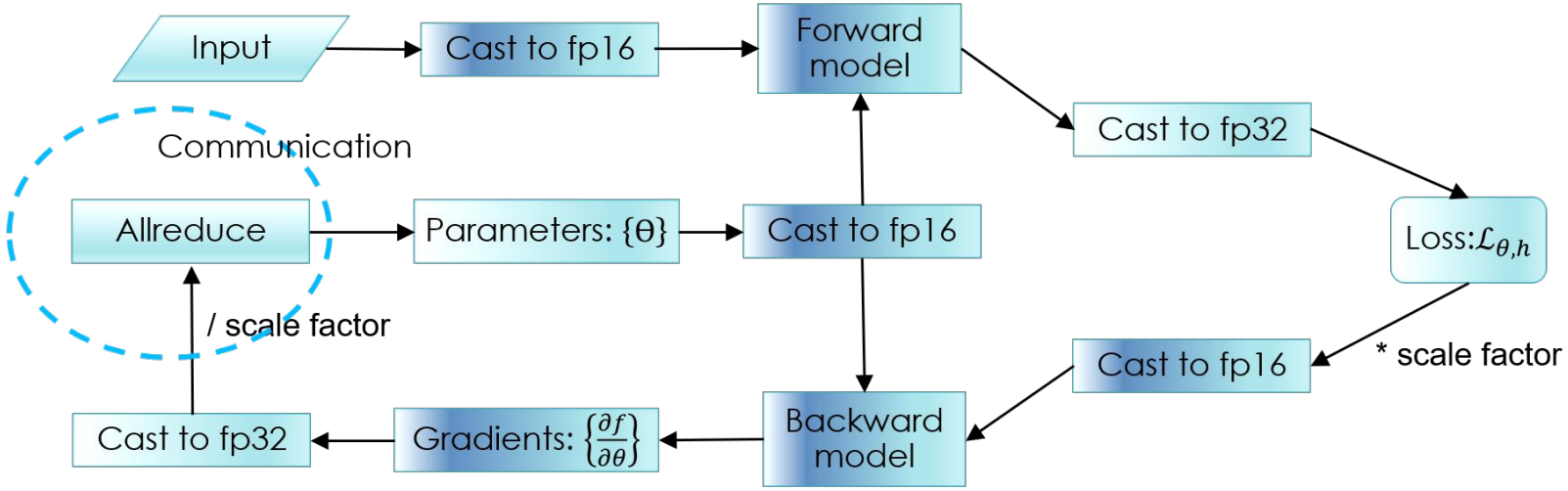
RAY tune

DeepHyper

Methods for Scaling up Training

Data Parallel Training

- Model agnostic. Well supported by Tensorflow, PyTorch, etc.
- Embarrassingly parallel
- Large model → memory limited

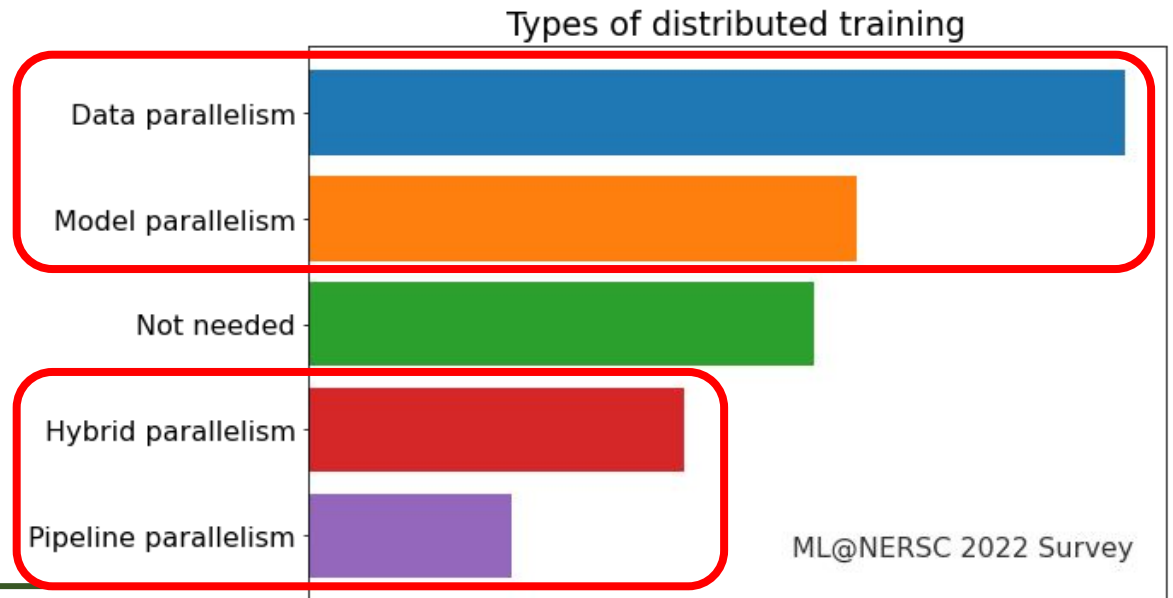


(source: 2012.01839)

Model Parallel Training

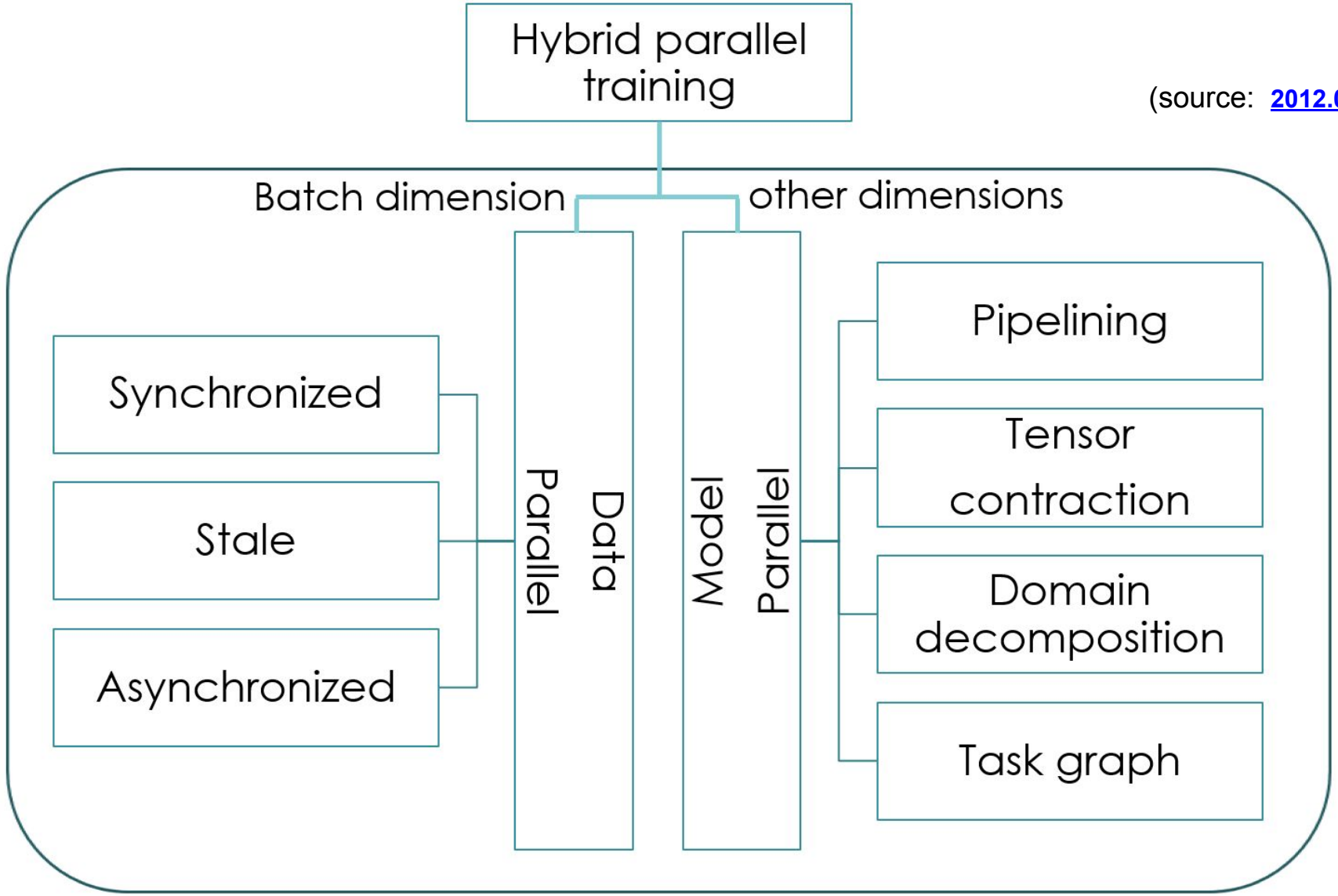
- Model specific, no general framework support
- Hard to parallelize (dependencies)
- Increasingly necessary for large models
  - Also for certain high-dimensional inputs
    - HEP: Graph partitioning
- Frequently combined with Data Parallel

→ Hybrid Parallel Training



Hybrid Parallel Training

(source: 2012.01839)



Training vs Optimization

A Neural Network f is characterized by two arrays of parameters

θ – the ***weights and biases***, and

h – the ***hyperparameters*** (e.g., # of hidden layers)

$$Y = f_{|\theta,h}(X)$$

$$L_{\theta,h} := F(f(X))_{|\theta,h}$$

By *definition*, the NN *loss function* L is

differentiable in θ and *non-differentiable* in h

following [2012.01839]

The optimal θ^* for a given dataset are determined during **training**

- (mostly) using *stochastic gradient descent* (SGD)

The optimal h^* are determined during **hyper-parameter optimization (HPO)**

- (typically) using *Bayesian optimization* or *evolutionary algorithms*