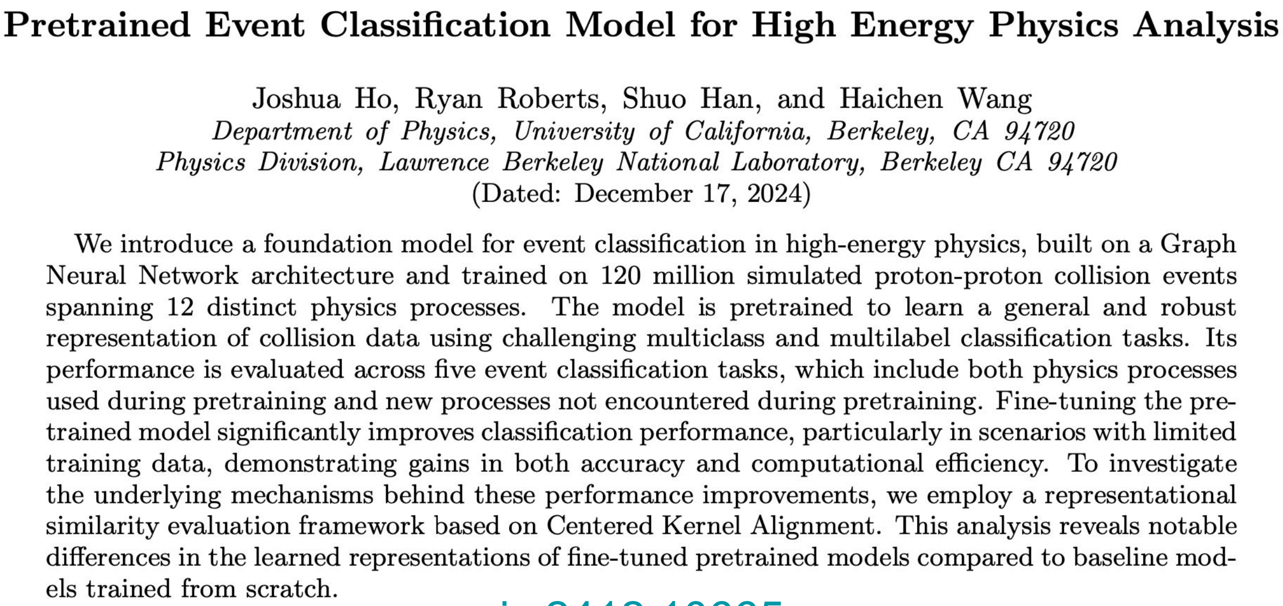
Pretrained Event Classification Model for HEP analysis

Shuo Han
LBNL & UCB

Jan 11, 2025, USTC



Pretrained Event Classification Model for High Energy Physics Analysis

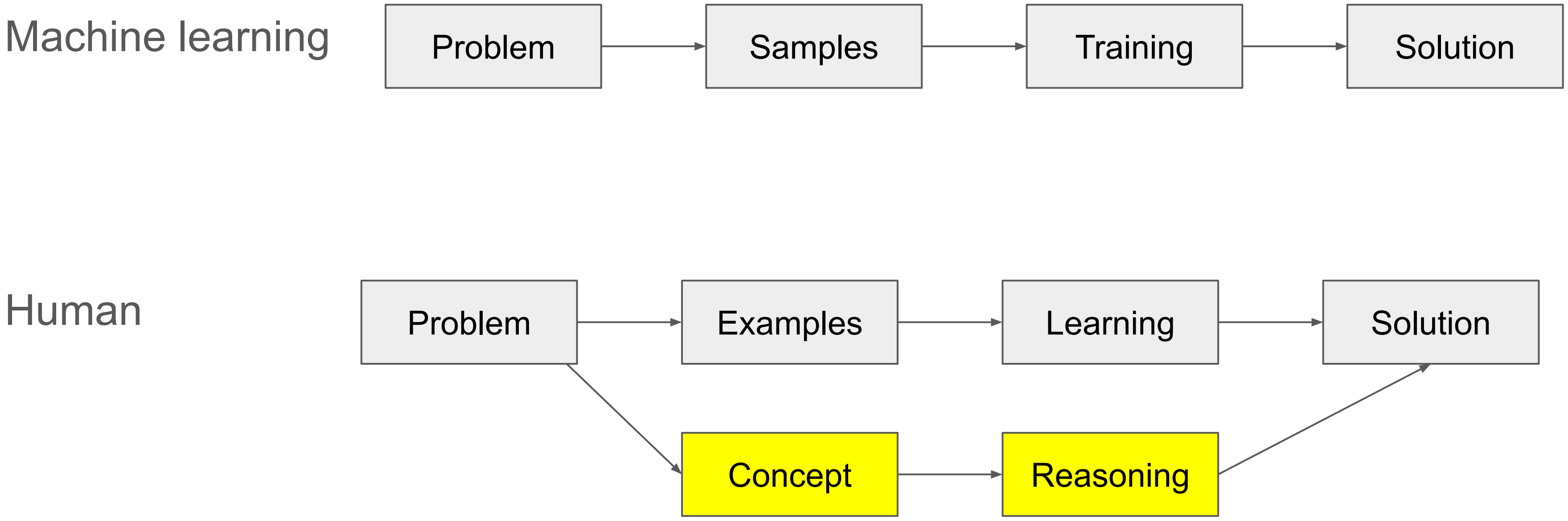
Joshua Ho, Ryan Roberts, Shuo Han, and Haichen Wang
Department of Physics, University of California, Berkeley, CA 94720
Physics Division, Lawrence Berkeley National Laboratory, Berkeley CA 94720

(Dated: December 17, 2024)

We introduce a foundation model for event classification in high-energy physics, built on a Graph Neural Network architecture and trained on 120 million simulated proton-proton collision events spanning 12 distinct physics processes. The model is pretrained to learn a general and robust representation of collision data using challenging multiclass and multilabel classification tasks. Its performance is evaluated across five event classification tasks, which include both physics processes used during pretraining and new processes not encountered during pretraining. Fine-tuning the pretrained model significantly improves classification performance, particularly in scenarios with limited training data, demonstrating gains in both accuracy and computational efficiency. To investigate the underlying mechanisms behind these performance improvements, we employ a representational similarity evaluation framework based on Centered Kernel Alignment. This analysis reveals notable differences in the learned representations of fine-tuned pretrained models compared to baseline models trained from scratch.

arxiv:2412.10665

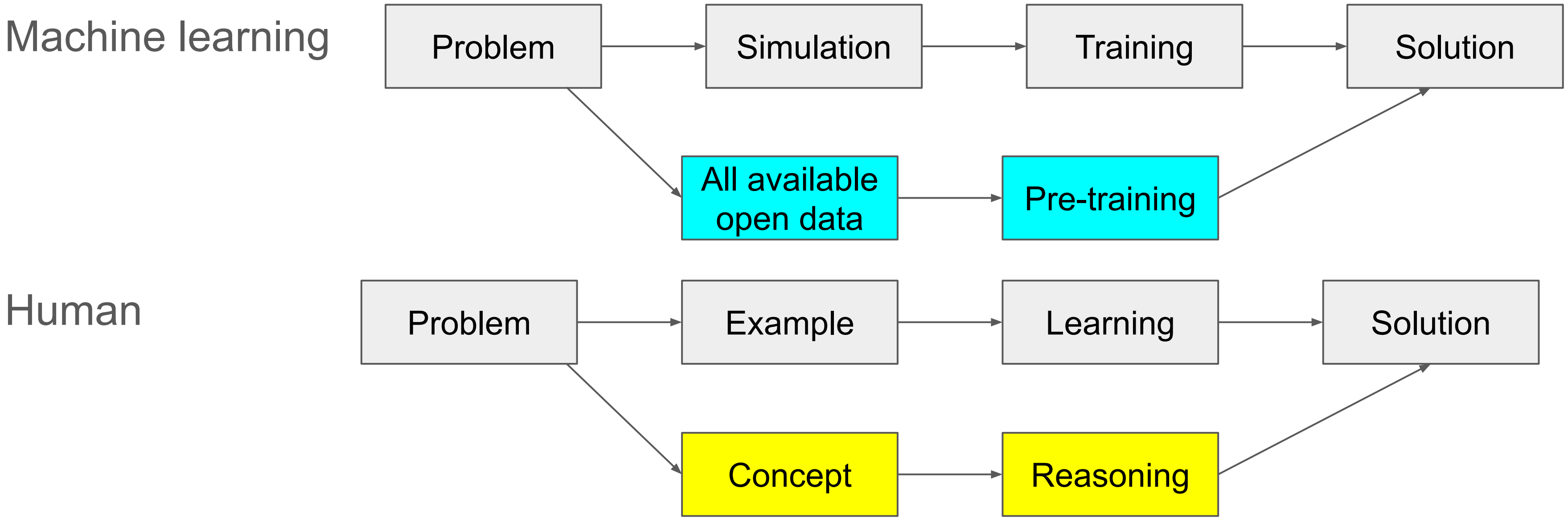
Human workflow and typical ML workflow



Human workflow has 2 advantages:

- Lower computing time
- Working with limited data

ML workflow with pre-training



(Although it’s still far from general AI) Pre-training can help AI to reduce GPU time, and work with limited data

Introduction

- In HEP analysis, each experiment carries out hundreds of measurements, most of which require **many iterations of training neural network models**.
- Our goal: a single model with pre-training that can be used for a wide range of tasks
  - Better overall performance
  - Lower training time
  - Capability with limited statistics
- This development could contribute to a foundational model for HEP analysis in future

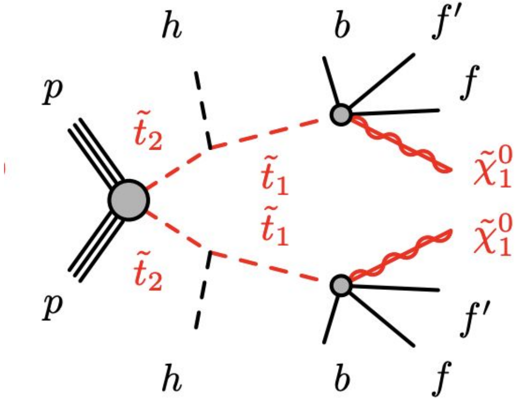
Outline

- We start with a simple task: binary classification for Higgs/Top processes
- How the pretrained model was built and its performance
- Model interpretability → model similarity
- GPU resources cost comparison between model frameworks

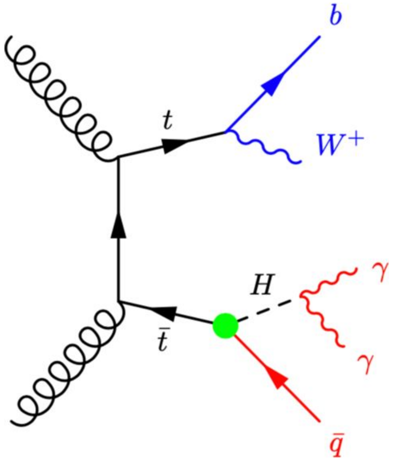
Training Setup: Pretraining Model Data

Pretrained Model Training Data:

- Higgs processes: ggF/VBF/VH/ttH/tH, and BSM CP-odd, FCNC, STOP
- Top processes: single top, ttbar, ttt, tttt, ttW, ttyy
- Statistics: ~120M total (~10M per class)



stop



FCNC

Binary classification Tasks:

- ttH CP Even vs CP Odd (H → γγ)
- FCNC vs tHjb (H → inclusive)
- stop vs ttH (H → inclusive)
- WH vs ZH (H → inclusive)
- ttW vs ttt

Training Setup: Inputs

Graph Neural Networks (GNNs) are a natural choice because of the point-cloud-like structure of our data, the choice of GNNs is just a proof of concept though

Input Node Features:

- Reconstructed objects: particle 4-vectors
- Particle Labels: type, b-tagging, lepton charge

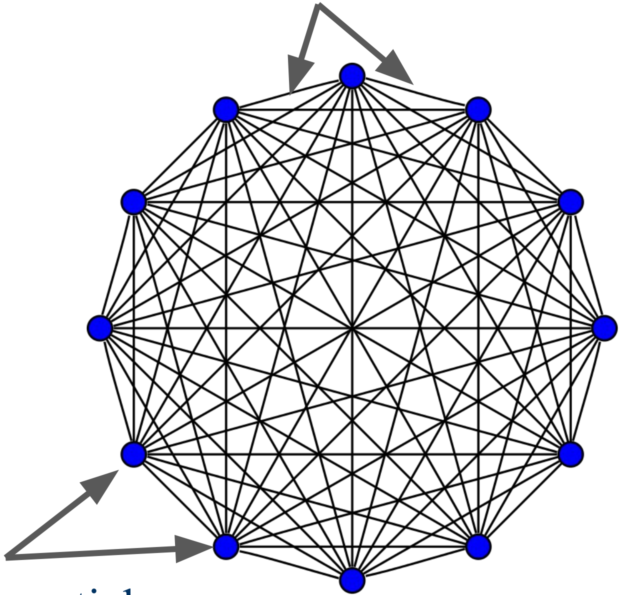
Input Edge Features:

- Angular and Translational Separation

Input Global Features:

- Number of particles in each graph

edges = relationship between particles



nodes = particle

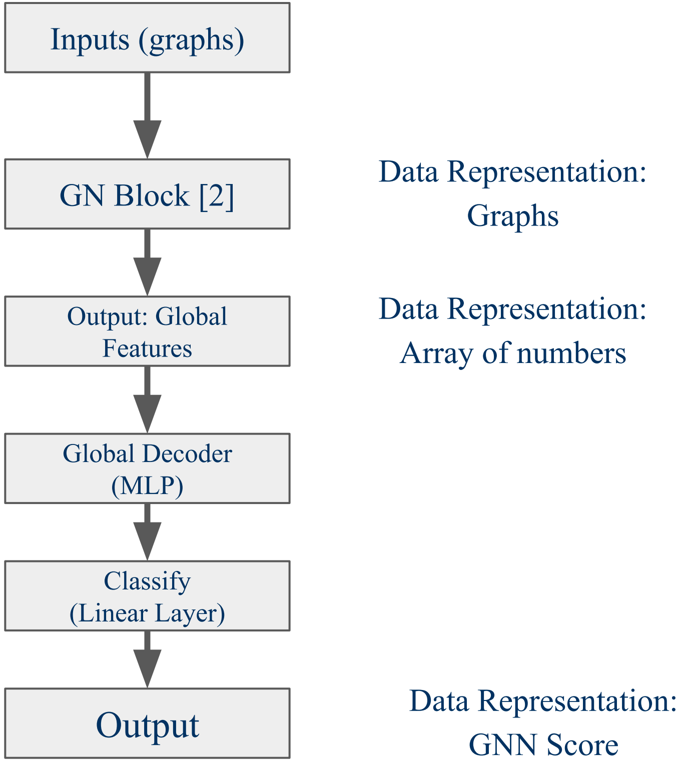
Training Setup: Baseline Model Architecture

Baseline Model:

A standard GNN trained for binary classification for one of our example analysis tasks.

Same model as ATLAS 4-tops observation paper [1]

Pytorch and Deep Graph Library (DGL)



[1] The ATLAS Collaboration: Eur. Phys. J. C 83, 496 (2023)

[2] arXiv:1806.01261

Training Setup: Pretrained Model Architecture

Pretraining Model:

Trained on **large and diverse** dataset, with different training goals

Multi-Class Classification:

Separate the data by processes

Predictions:

- P(ttH)
- P(ggF)
- P(WH)
- … etc

Multi-Label Classification:

Separate the data by phase spaces

Prediction:

- Exists: higgs_exists, top1_exists, …
- Pt: higgs_pt, top1_pt, …
- η: higgs_eta, top1_eta, …
- ***ϕ***: higgs_phi, top1_phi, …

Training Setup: Fine-tuning Model Architecture

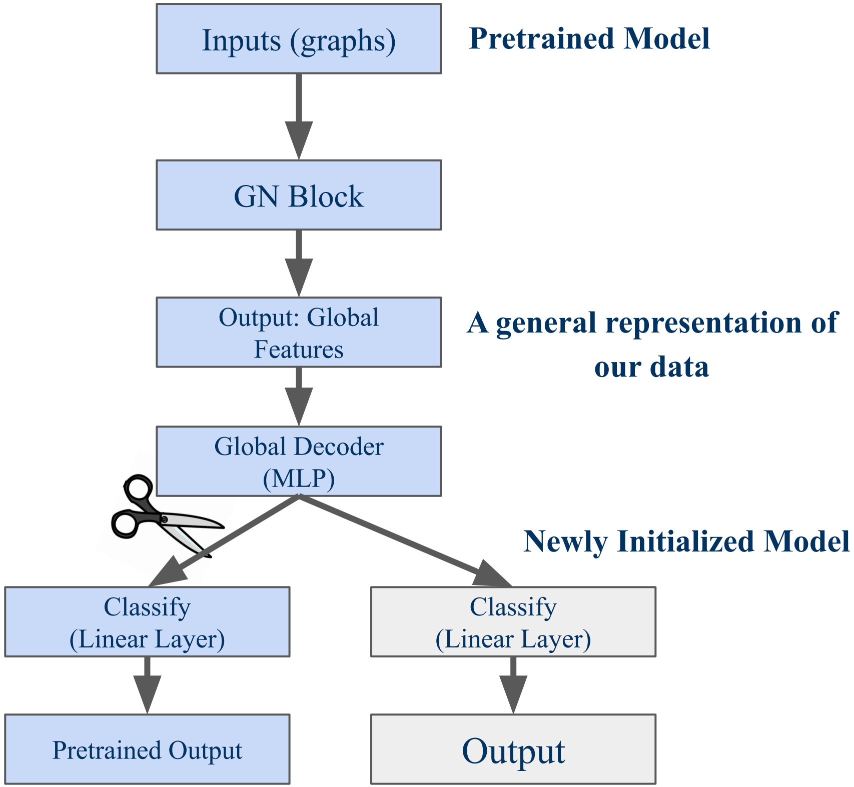
Fine-tuned Model:

- obtain the pre-trained model first
- Keep using weights of the pretrained model, but define a newly initialized MLP

Adjustment of learning rate

- Keep updating the pretrained model with a lower learning rate of 10^{-5}
- Train the newly initialized model at a regular learning rate 10^{-4}
- Learning rates decay every epoch

NOT transferred learning because the pretraining is still trainable (the transferred learning setup is WIP)



Results (Overall Performance)

Utilizing full statistics:

- 120M Pretraining
- 20M Analysis

Immediate Performance:

- Initial boost in performance
- Seen in all analysis tasks

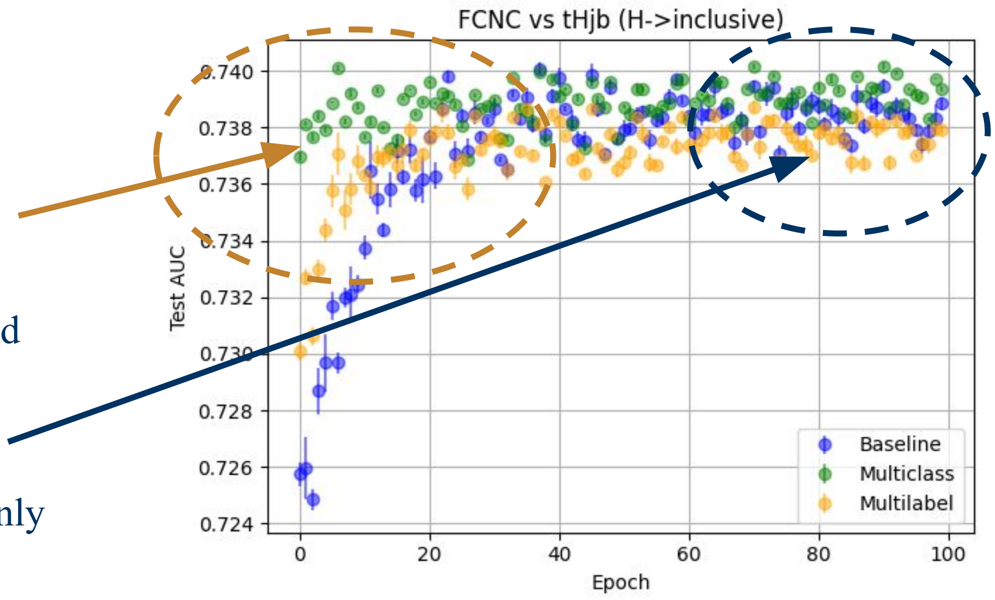
Ultimate Performance:

- The ultimate performance is slightly increased
- Seen only in some of the analysis tasks

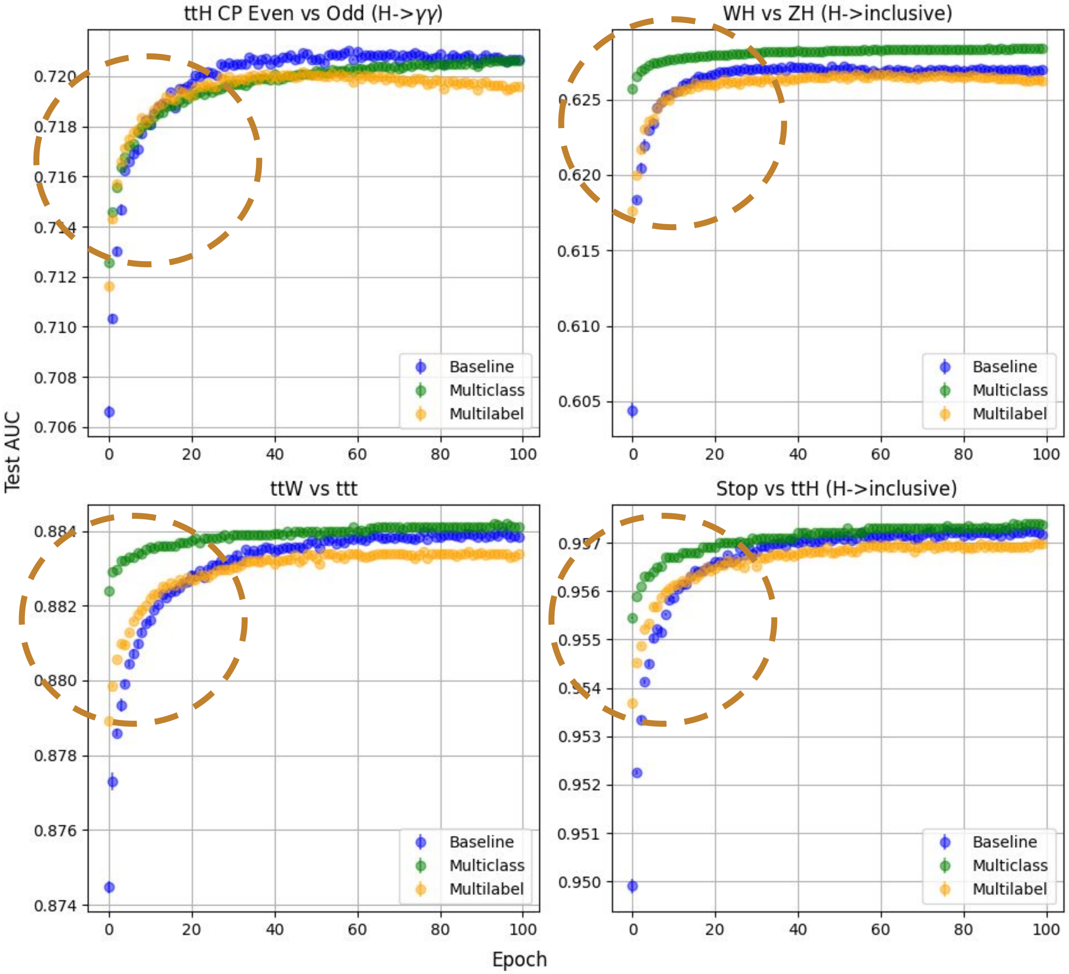
Usage:

- Computing power is expensive and we can only afford a few epochs

Performance for Example Analysis Tasks



Performance For Example Analysis Tasks



Results (when Limited Stat.)

Limited Statistics:

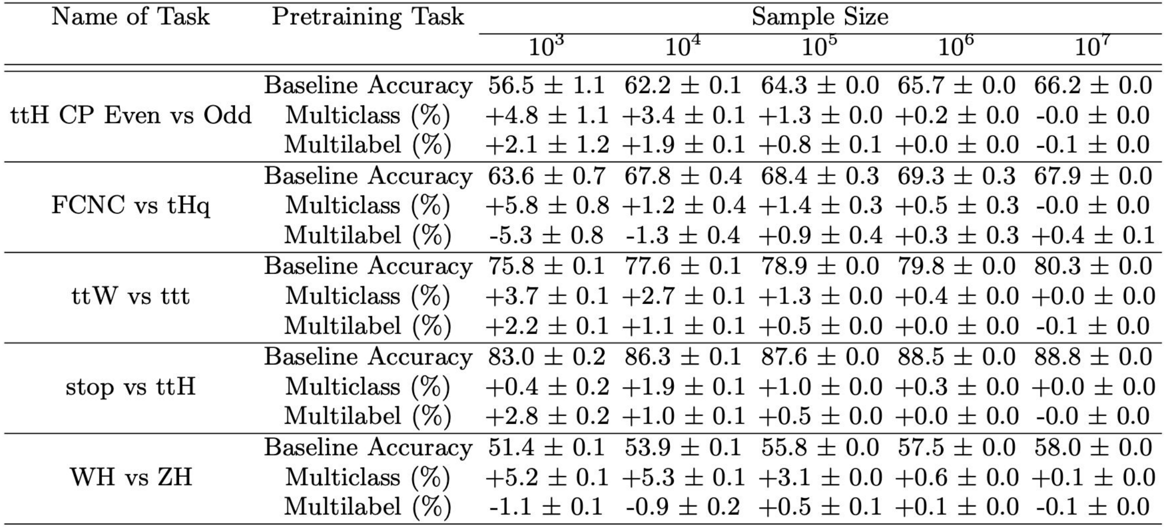
- Significant increase in performance (up to 15% improvement in AUC, and 5% in accuracy)

Large Statistics:

- Small increase in performance (0-2% increase in AUC, and 0-0.5% in accuracy)

Usage:

- Training on real collider data
- Categorization in small phase spaces, or signal region selection is very tight
- Simulation is expensive



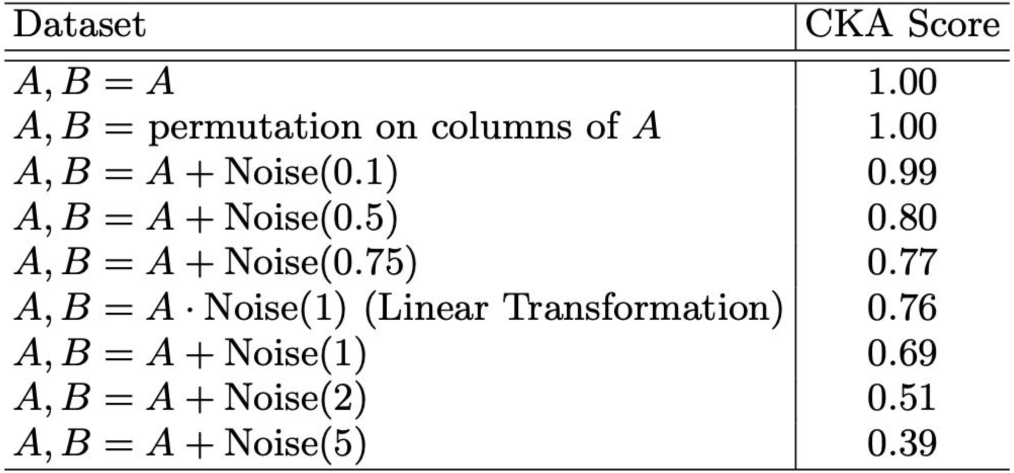
Name of Task	Pretraining Task	Sample Size				
		10^3	10^4	10^5	10^6	10^7
ttH CP Even vs Odd	Baseline Accuracy	56.5 ± 1.1	62.2 ± 0.1	64.3 ± 0.0	65.7 ± 0.0	66.2 ± 0.0
	Multiclass (%)	+4.8 ± 1.1	+3.4 ± 0.1	+1.3 ± 0.0	+0.2 ± 0.0	-0.0 ± 0.0
	Multilabel (%)	+2.1 ± 1.2	+1.9 ± 0.1	+0.8 ± 0.1	+0.0 ± 0.0	-0.1 ± 0.0
FCNC vs tHq	Baseline Accuracy	63.6 ± 0.7	67.8 ± 0.4	68.4 ± 0.3	69.3 ± 0.3	67.9 ± 0.0
	Multiclass (%)	+5.8 ± 0.8	+1.2 ± 0.4	+1.4 ± 0.3	+0.5 ± 0.3	-0.0 ± 0.0
	Multilabel (%)	-5.3 ± 0.8	-1.3 ± 0.4	+0.9 ± 0.4	+0.3 ± 0.3	+0.4 ± 0.1
ttW vs ttt	Baseline Accuracy	75.8 ± 0.1	77.6 ± 0.1	78.9 ± 0.0	79.8 ± 0.0	80.3 ± 0.0
	Multiclass (%)	+3.7 ± 0.1	+2.7 ± 0.1	+1.3 ± 0.0	+0.4 ± 0.0	+0.0 ± 0.0
	Multilabel (%)	+2.2 ± 0.1	+1.1 ± 0.1	+0.5 ± 0.0	+0.0 ± 0.0	-0.1 ± 0.0
stop vs ttH	Baseline Accuracy	83.0 ± 0.2	86.3 ± 0.1	87.6 ± 0.0	88.5 ± 0.0	88.8 ± 0.0
	Multiclass (%)	+0.4 ± 0.2	+1.9 ± 0.1	+1.0 ± 0.0	+0.3 ± 0.0	+0.0 ± 0.0
	Multilabel (%)	+2.8 ± 0.2	+1.0 ± 0.1	+0.5 ± 0.0	+0.0 ± 0.0	-0.0 ± 0.0
WH vs ZH	Baseline Accuracy	51.4 ± 0.1	53.9 ± 0.1	55.8 ± 0.0	57.5 ± 0.0	58.0 ± 0.0
	Multiclass (%)	+5.2 ± 0.1	+5.3 ± 0.1	+3.1 ± 0.0	+0.6 ± 0.0	+0.1 ± 0.0
	Multilabel (%)	-1.1 ± 0.1	-0.9 ± 0.2	+0.5 ± 0.1	+0.1 ± 0.0	-0.1 ± 0.0

Similarity calculation

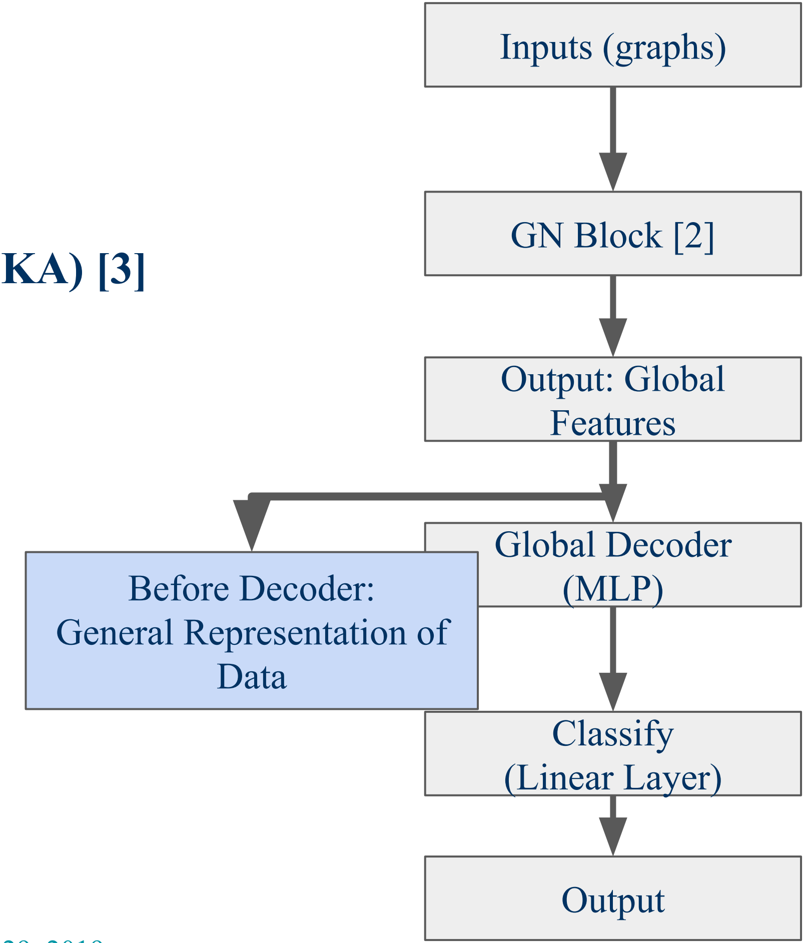
How to tell that the pretraining works?

- Using similarity between different models

Similarity indicator: Centered Kernel Alignment (CKA) [3]



Dataset	CKA Score
$A, B = A$	1.00
$A, B =$ permutation on columns of A	1.00
$A, B = A + \text{Noise}(0.1)$	0.99
$A, B = A + \text{Noise}(0.5)$	0.80
$A, B = A + \text{Noise}(0.75)$	0.77
$A, B = A \cdot \text{Noise}(1)$ (Linear Transformation)	0.76
$A, B = A + \text{Noise}(1)$	0.69
$A, B = A + \text{Noise}(2)$	0.51
$A, B = A + \text{Noise}(5)$	0.39

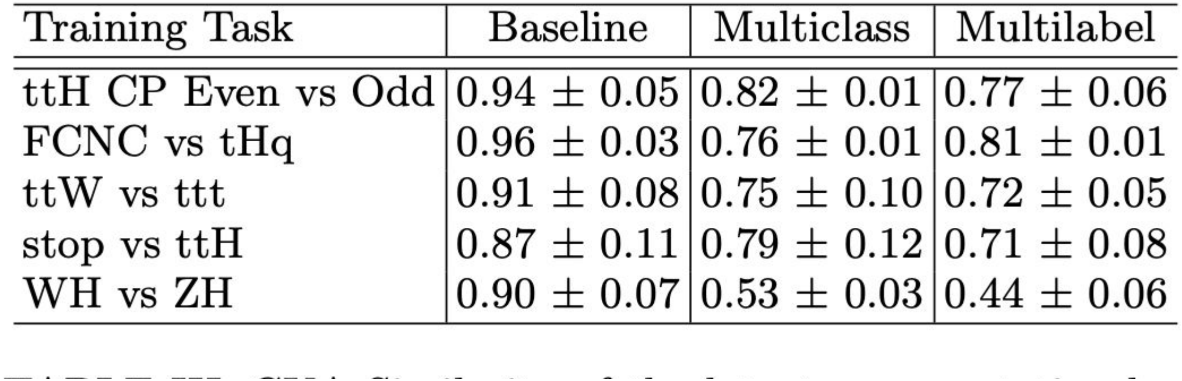


[3] S. Kornblith, et. al. In International Conference on Machine Learning, p. 3519–3529, 2019.

Similarity between models

Compare each model with a well-trained benchmark baseline model:

- Similarity between (benchmark) baseline and pre-trained models are <80%.
- Pre-trained models have slightly better performance to solve binary classification problems
- In summary, pre-trained models are utilizing different representations of collision events

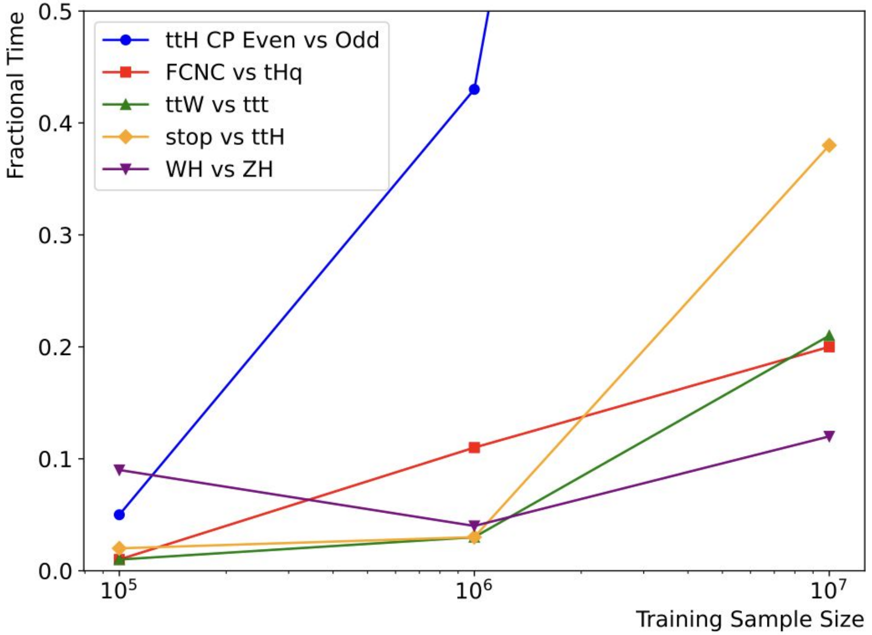


Training Task	Baseline	Multiclass	Multilabel
ttH CP Even vs Odd	0.94 ± 0.05	0.82 ± 0.01	0.77 ± 0.06
FCNC vs tHq	0.96 ± 0.03	0.76 ± 0.01	0.81 ± 0.01
ttW vs ttt	0.91 ± 0.08	0.75 ± 0.10	0.72 ± 0.05
stop vs ttH	0.87 ± 0.11	0.79 ± 0.12	0.71 ± 0.08
WH vs ZH	0.90 ± 0.07	0.53 ± 0.03	0.44 ± 0.06

Compare each model with a well-trained benchmark baseline model

Resources Used For Training

GPU hours to get achieve performance of 99% of the baseline ultimate performance



With full statistic (10^7)

- Multiclass Pretraining: 46 GPU hours
- Multilabel Pretraining: 60 GPU hours
- Baseline: 2.9 GPU hours in ave
- With pretraining: 1.1 GPU hours in ave

Conclusions

- Pretrained models **has better performance** with **limited statistics** or **limited epochs**
- Pretrained models **converge faster, which** leads to a **decrease in GPU resources**
- We can calculate model similarity to gain insight on the information what the pre-trained models have learned

Thank you.

Thank you!