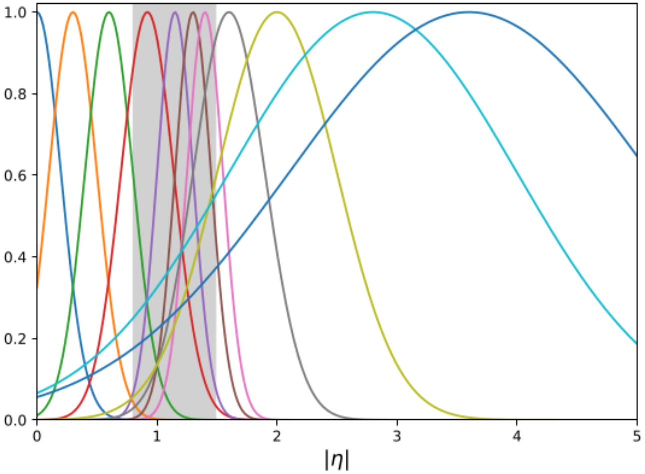
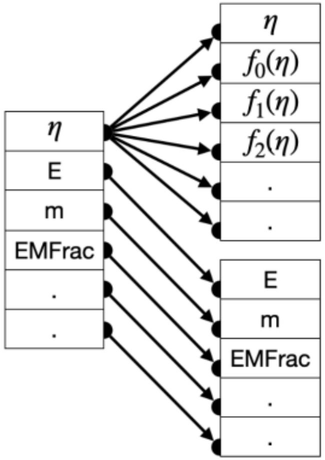
ML4Jets

Jeremy Couthures

Eta Annotation



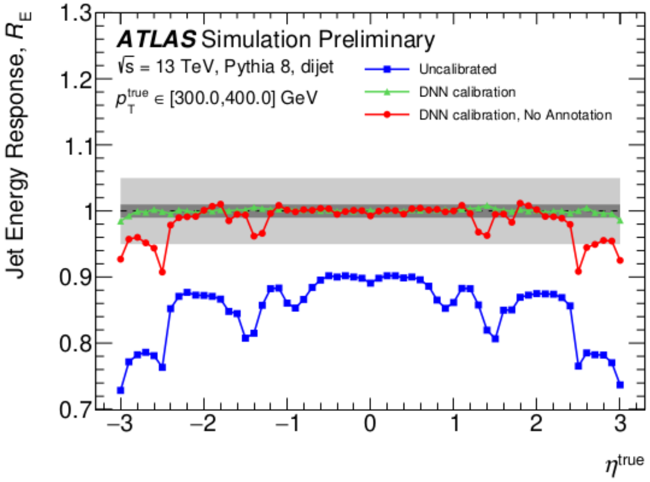
Ref. available soon



Ref. available soon

Complex dependence on η

- With sharp changes from bin-to-bin due to detector geometry/instrumentation
- Difficult for DNN to adapt to this
- Annotation strategy
  - → Add 12 features that are functions of η
  - → Encoding distance to different η regions
- Clear improvement:



Ref. available soon

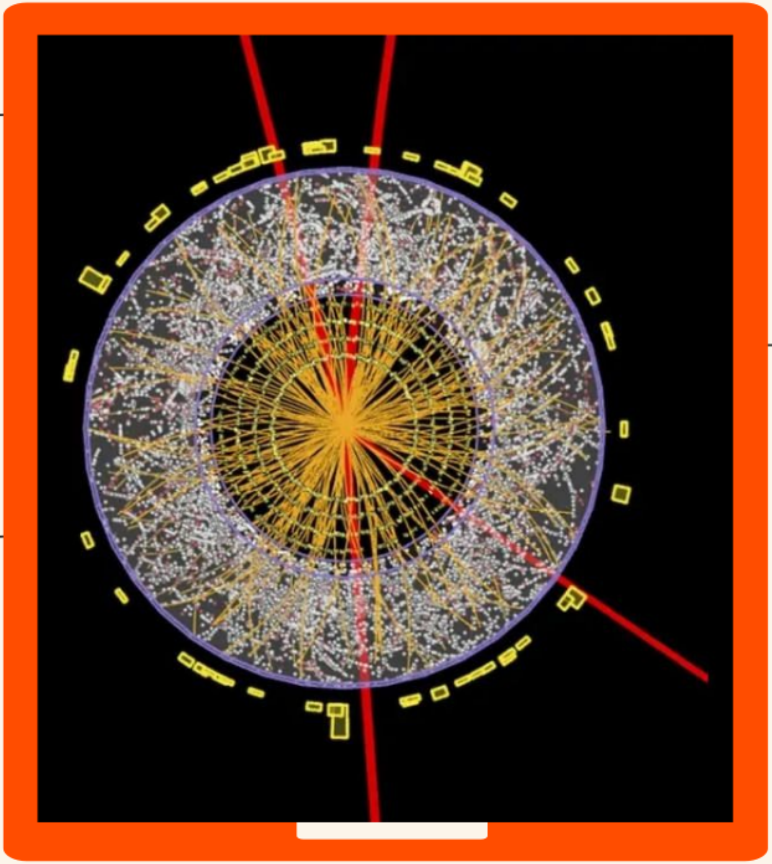
Physical Inductive Biases

Data Structure

Relational structure, ordering, feature selection, pre-processing, etc

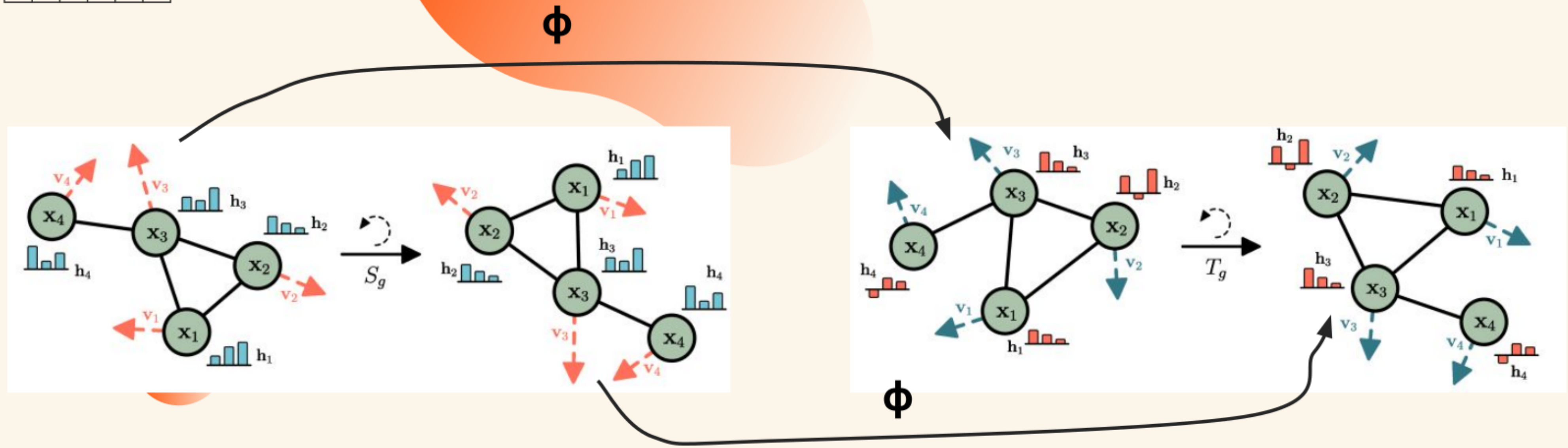
Model Constraints

Restricting model weights, learned function, propagated information, etc



Task Formulation

Physics informed neural networks, incorporating conservation laws or equations through loss function design, etc



Symmetries are (potentially) **powerful physical inductive biases**

Image from [arxiv:2102.09844]

Evaluating **Equivariance**

Tagging	Accuracy	AUC	Parameters	Ant Factor
ResNeXt	0.936	0.984	1.46M	4.28
ParticleNet	0.938	0.985	498k	13.4
PFN	0.932	0.982	82k	67.8
EFP	0.932	0.980	**1k**	**5000**
LGN	0.929	0.964	4.5k	617
VecNet.1	0.935	0.984	633k	9.87
VecNet.2	0.931	0.981	15k	350
PELICAN	**0.943**	**0.987**	45k	171
LorentzNet	0.942	0.9868	220k	35

Tracking	N Hidden	AUC	Parameters	Ant Factor
EuclidNet	8	**0.9913**	**967**	**11887**
InteractionNet	8	0.9849	1432	4625
EuclidNet	16	0.9932	2580	**5700**
InteractionNet	16	0.9932	4392	3348
EuclidNet	32	0.9941	4448	3811
InteractionNet	32	**0.9978**	**6448**	**7049**

Accuracy

- Jet tagging: highest accuracy model is equivariant, but not all equivariant models perform well
- Tracking: for small models equivariant models have highest accuracy, but performance plateaus as models grow
- Overall, relationship between equivariance and accuracy is unclear (confounding factors remain)

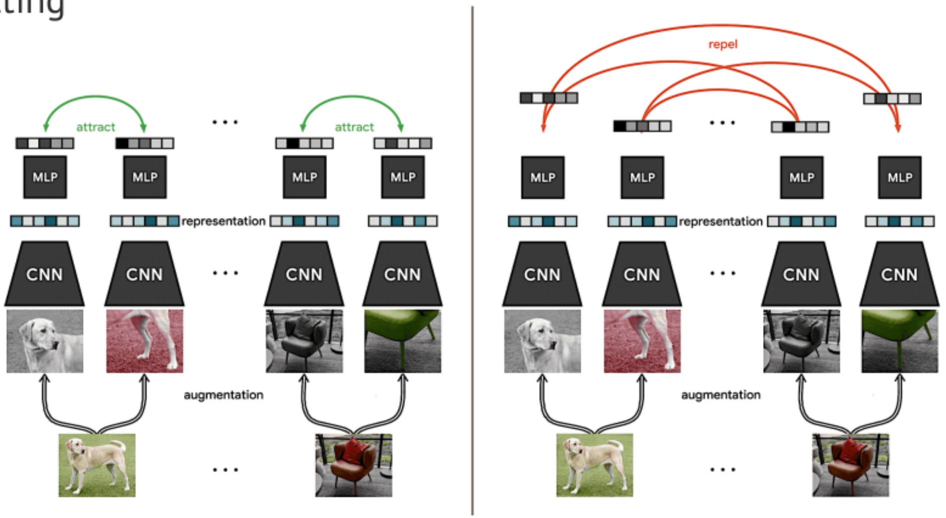
Model Efficiency

- Jet tagging: regression model with physics inputs is most efficient. Semi-equivariant model is also efficient.
- Tracking: relationship changes with model size
- Overall, equivariance does not seem to contribute directly to model efficiency

Ant factor = 10^5/[(1-AUC)*N_p]

SELF-SUPERVISED LEARNING

- Self-Supervised Learning (SSL): Can we rely less on (labelled) simulation to achieve more?
  - learn generic representations for all kinds of downstream tasks: tagging, reconstruction, anomaly detection, …
- **Contrastive learning (SimCLR)**
  - maximize similarity between positive pairs and minimize that between negative pairs
  - positive pairs derived from the same sample, but with different augmentations (cropping, color distortion, …)
- But what is the most effective augmentation for jets?
  - JetCLR [2108.04253]: rotation, translation, soft/collinear splitting
  - re-simulation as an augmentation
    - re-simulate w/ varied shower/hadronization models/configs
      - talk by J. Krupa @ BOOST 2023
    - re-simulate w/ varied detector configurations
      - talk by E. Dreyer on Monday
    - potentially learning a more robust representation
    - but not applicable to real data…



Learning Symmetries

Lax pairs

$$\frac{\mathrm{d}}{\mathrm{dt}} L = [L, M],$$

$$F_1 = 2\lambda,$$

$$F_2 = 2\lambda^2 + 4H,$$

$$F_3 = 2\lambda^3 + 12\lambda H,$$

$$F_4 = 2\lambda^4 + 24\lambda H + 4H^2.$$

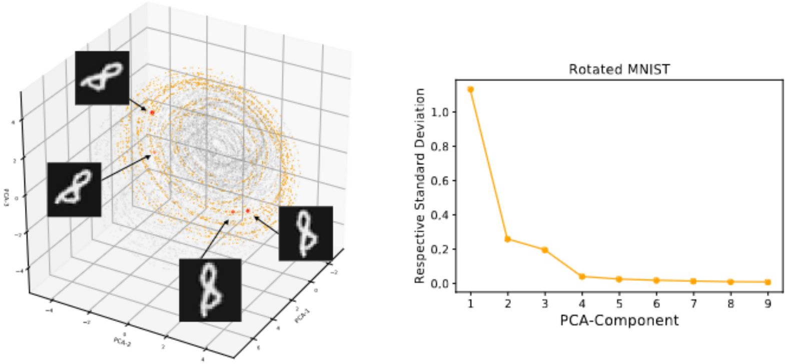


Figure 9: **Left:** Pointcloud of first three PCA components of our rotated MNIST dataset. Highlighted in orange are the orbits of multiple digits eight. Gray points correspond to the other digits present in this dataset. **Right:** The standard deviation on the generators identified from this pointcloud for the digit eight.

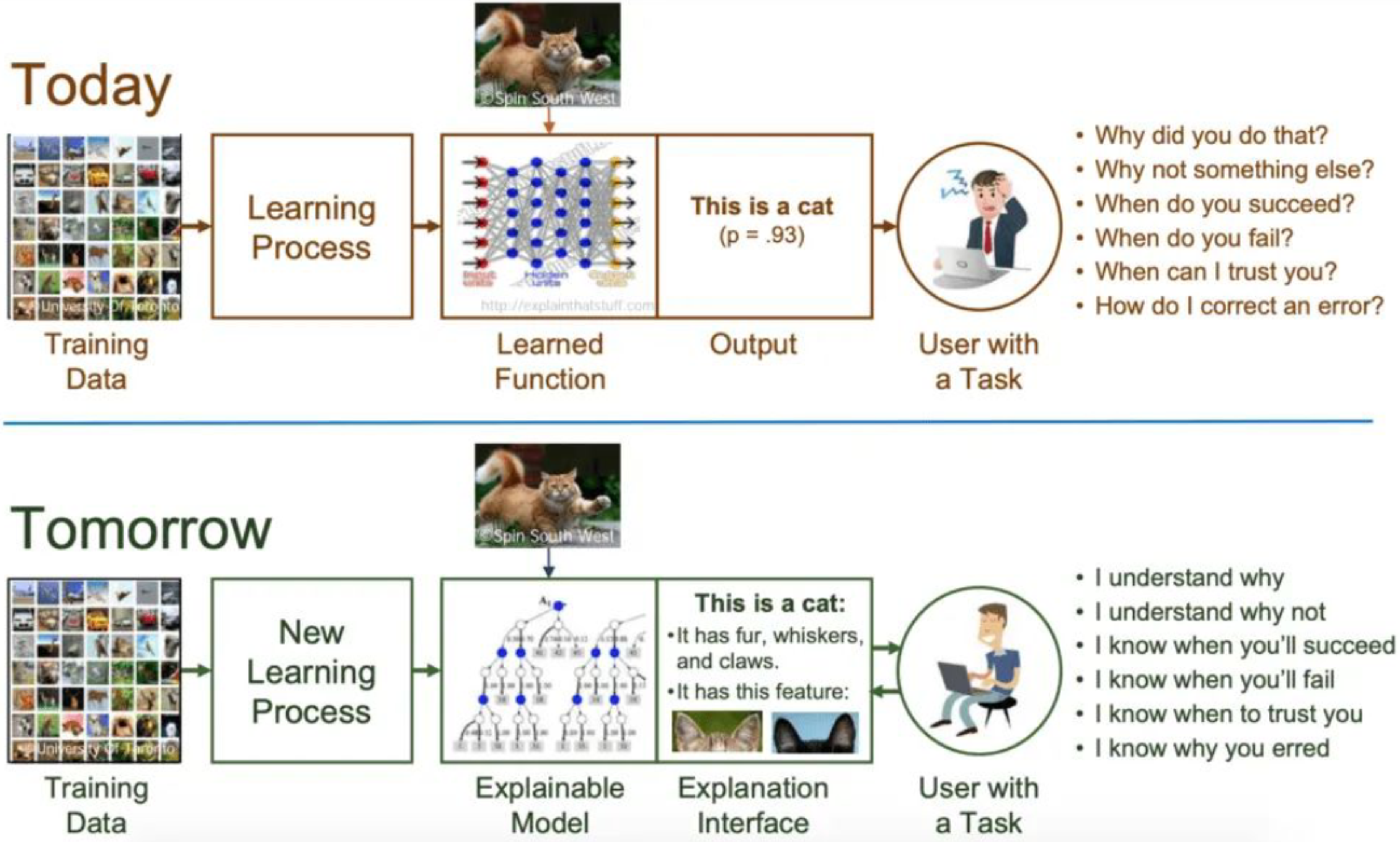
AI and Uncertainty challenge in fundamental physics

Definitions

Explanation

"An explanation is a presentation of (aspects of) the reasoning, functioning and/or behavior of a machine learning model in human-understandable terms" [Nau+23]

"The **belief** (by the trustor) in the ability (of the trustee) to achieve **something**"



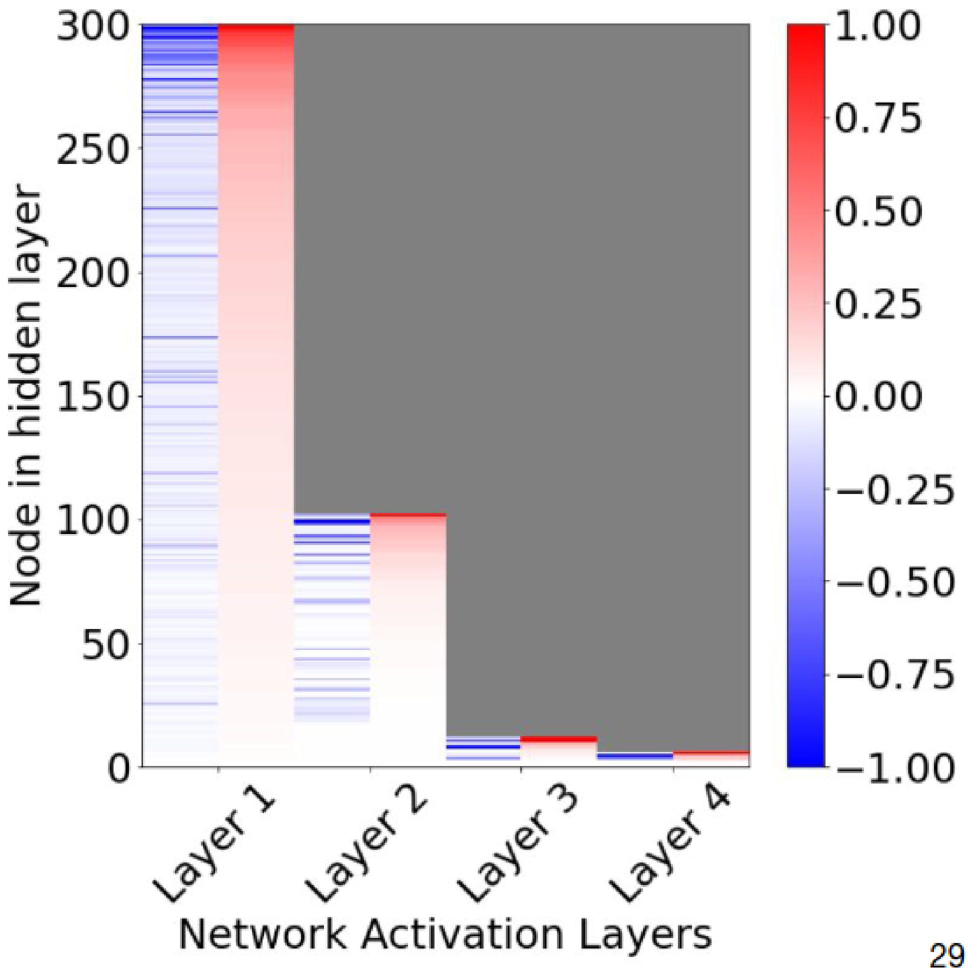
source: https://www.darpa.mil/attachments/XAIProgramUpdate.pdf

Neuron Activation Pattern (NAPs)

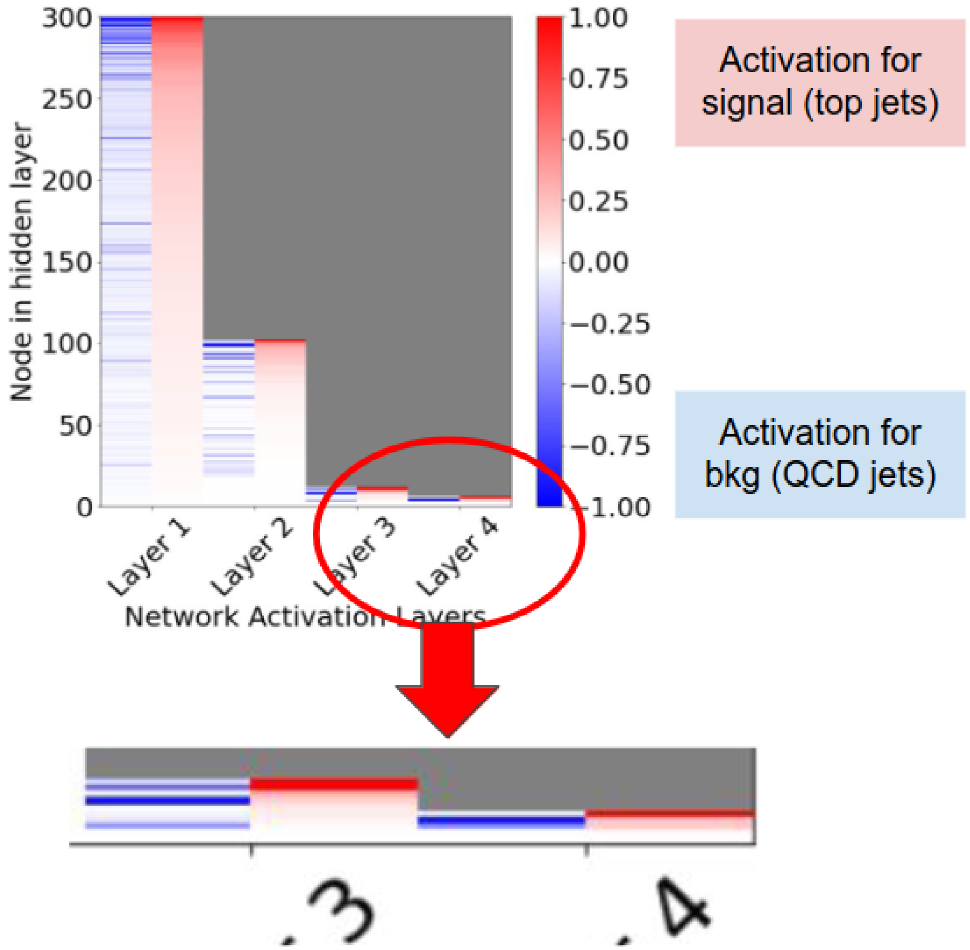
- Feature importance metrics don't reveal information about model's inner workings
- Want to detect internal disentanglements, context-aware neural pathways, hyperparameter reoptimization
- Define a **Relative Neural Activity** (RNA) score for different nodes within a layer

$$\mathrm{RNA}(j,k;\mathcal{S}) = \frac{\sum_{i=1}^{N} a_{j,k}(s_i)}{\max_j \sum_{i=1}^{N} a_{j,k}(s_i)}$$

- *j*, *k* are the node and layer numbers
- *S* is the representative dataset over which the RNA scores are evaluated

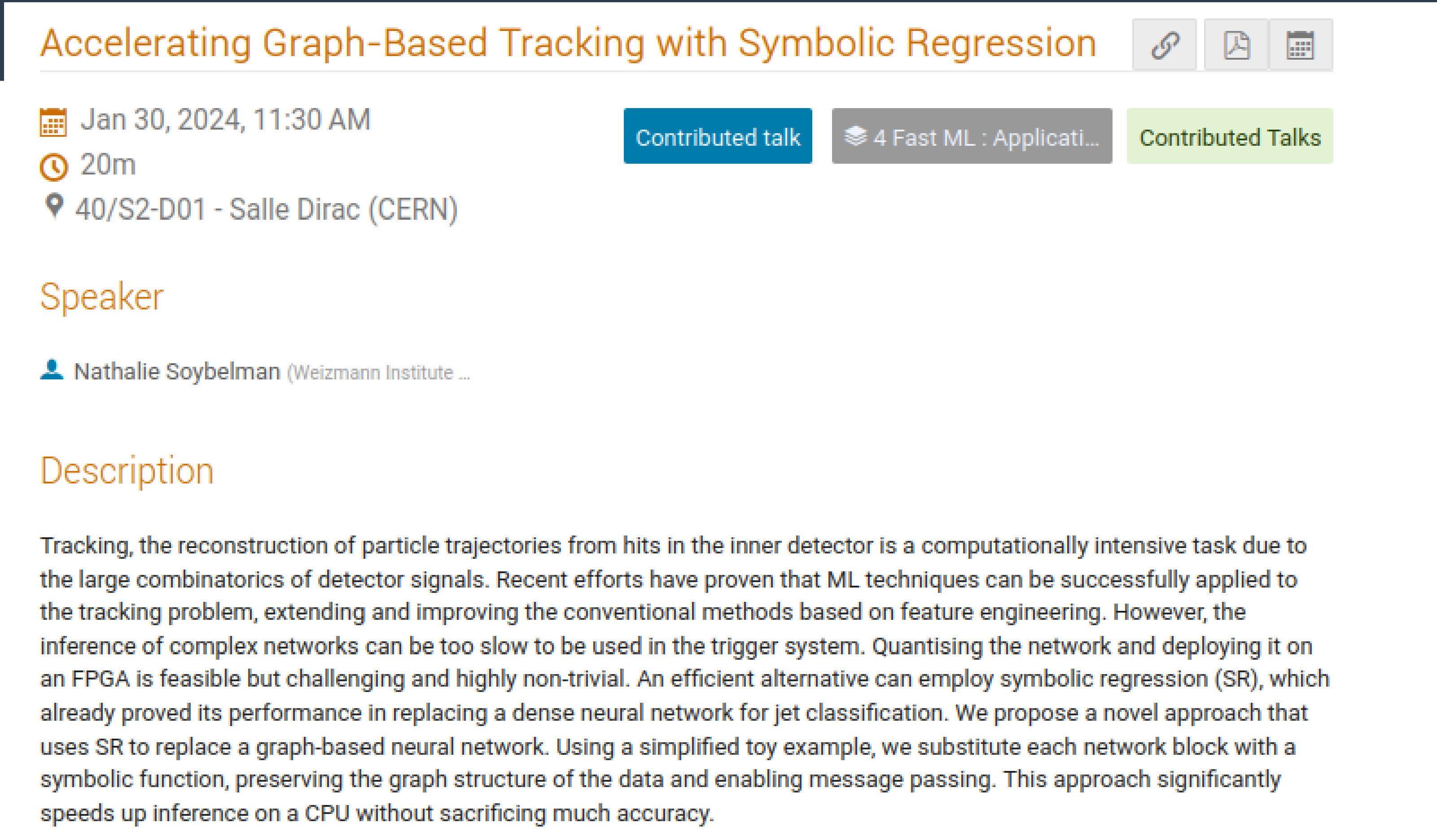


NAP Diagram for TopoDNN



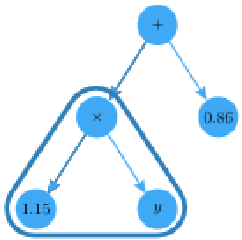
- RNA scores of QCD jets mapped as negative numbers for simultaneous visualization
- Observations
  - The model is very sparse
  - The information pathways for jet classes are disentangled by layer 3, layer 4 is kind of redundant
- **Retrained the model with (120,40,6) hidden nodes, got same performance**

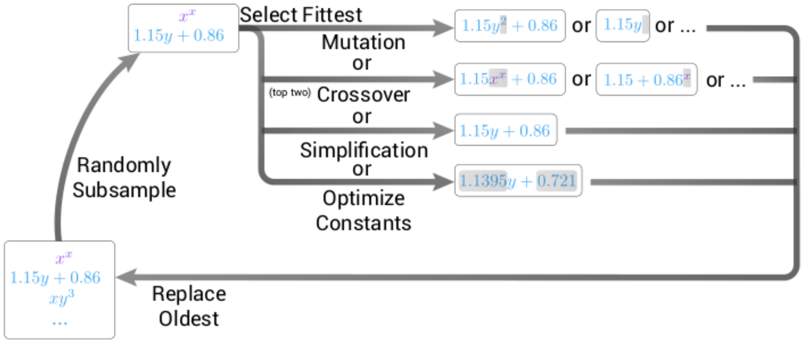
Next IML (next week)


Accelerating Graph-Based Tracking with Symbolic Regression
Jan 30, 2024, 11:30 AM
20m
40/S2-D01 - Salle Dirac (CERN)
Contributed talk
4 Fast ML : Applicati...
Contributed Talks
Speaker
Nathalie Soybelman (Weizmann Institute ...
Description
Tracking, the reconstruction of particle trajectories from hits in the inner detector is a computationally intensive task due to the large combinatorics of detector signals. Recent efforts have proven that ML techniques can be successfully applied to the tracking problem, extending and improving the conventional methods based on feature engineering. However, the inference of complex networks can be too slow to be used in the trigger system. Quantising the network and deploying it on an FPGA is feasible but challenging and highly non-trivial. An efficient alternative can employ symbolic regression (SR), which already proved its performance in replacing a dense neural network for jet classification. We propose a novel approach that uses SR to replace a graph-based neural network. Using a simplified toy example, we substitute each network block with a symbolic function, preserving the graph structure of the data and enabling message passing. This approach significantly speeds up inference on a CPU without sacrificing much accuracy.

Symbolic regression

$1.15y + 0.86$





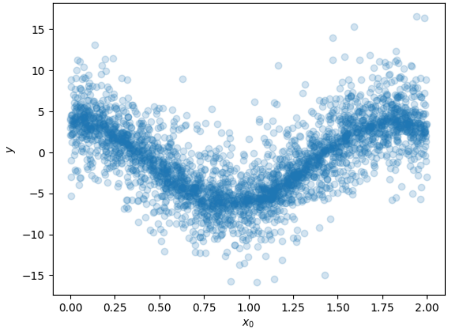
Genetic Algorithm

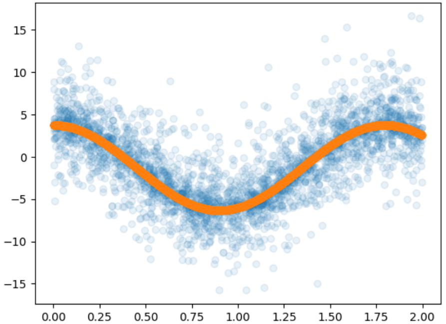
$$\sigma \sim U(0.1, 5.0)$$
$$\epsilon \sim N(0, \sigma^2)$$
$$y = 5\ \cos(3.5 x_0) - 1.3 + \epsilon.$$

Truth

$$5.0337477 \cos(3.496164 x_0) - 1.29099218487498$$

Learned





Neural Nets + Symbolic Regression

https://github.com/MilesCranmer/PySR