FAIR Principles for ML Models and Model Interpretation

Avik Roy
on behalf of the FAIR4HEP Collaboration

March 14, 2022

FAIR4HEP

Introduction to FAIR Principles

- FAIR principles have been originally proposed to inspire scientific data management for reproducibility and maximal reusability[1]
- Originally proposed for scientific data, these principles can be extended as guidelines to manage and preserve other Digital Objects (DOs) e.g. research software[2], tutorials and notebooks[3], AI and ML models[4]

Findable:	locating DOs in a failsafe fashion
Accessible:	obtaining DOs along with their context, content, and format
Interoperable:	being usable across multiple computing platforms
Reusable:	specifying the context and extent of reusing DOs

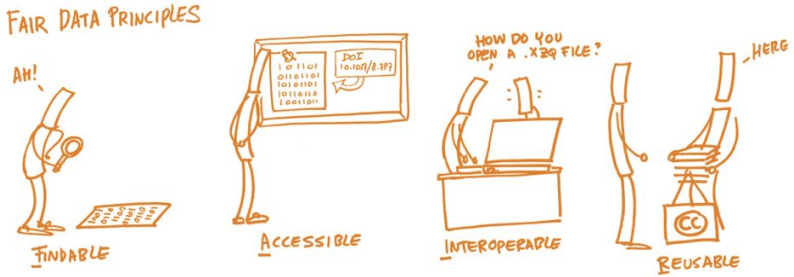
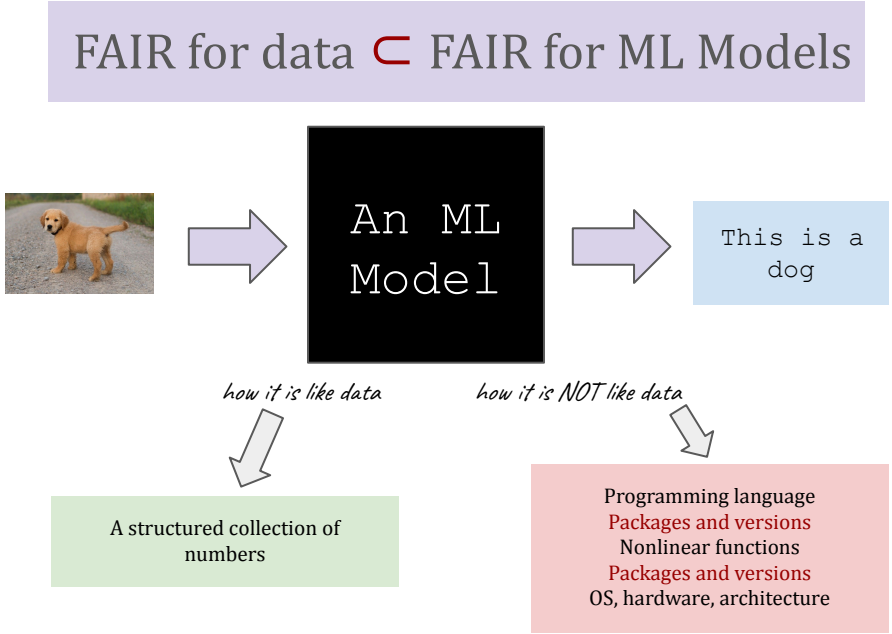


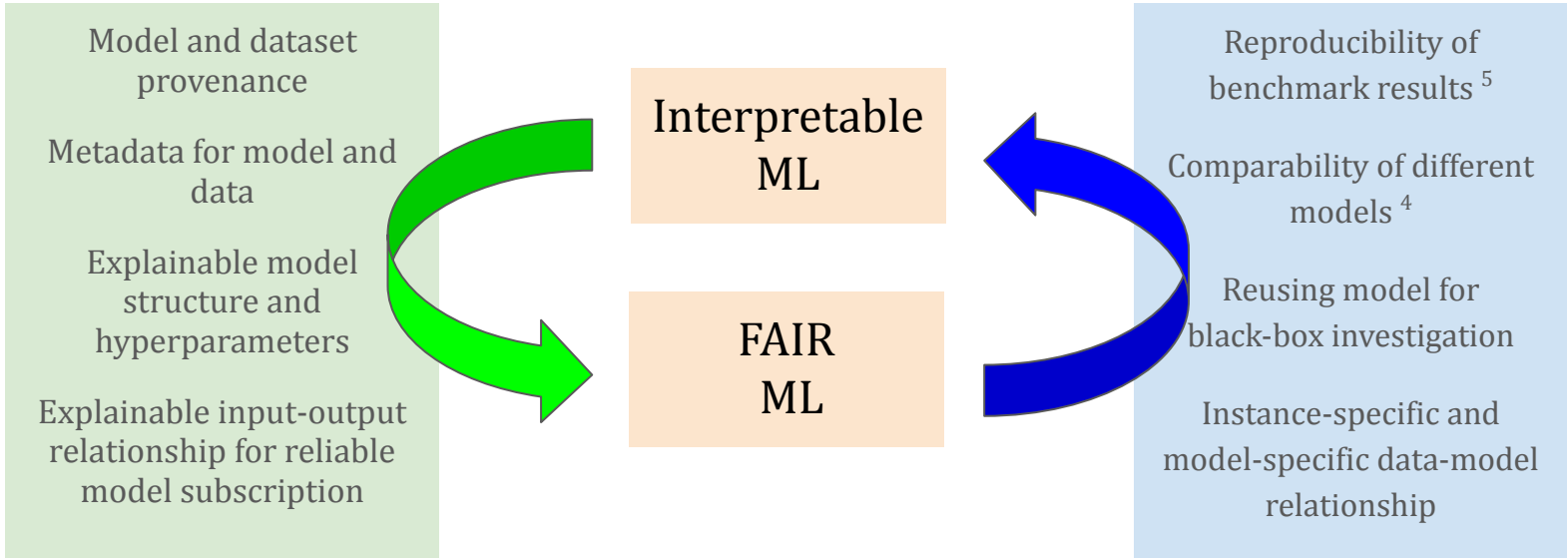
Image: https://book.fosteropenscience.eu/

FAIR Principles for ML Models

- Technical: Specific OS/software/package dependencies, availability of dataset and its provenance
- Analytical:
  - Using model inference for new/curtailed datasets
  - Retraining model
  - Feature Engineering
  - Reoptimizing model hyperparameters



Model Interpretability and FAIR: A two-way bridge



FAIR4HEP: FAIR data and AI for HEP

- Multi-disciplinary, multi-institute team for learning how data-intensive HEP research can benefit from FAIR principles and vice versa
- Develop community standards to implement FAIR principles and tools to implement them
- Develop and share benchmark FAIR data and models
- Explore interplay between data and models to explore interpretability and model robustness

What makes data and AI FAIR for physicists?

How FAIR principles facilitate today's physics research?

Are AI models in HEP robust?

How well do we understand AI models and their relationship with data?

visit us: https://fair4hep.github.io/

A Benchmark AI Model

- Interaction Network Model to distinguish H > bb jets from QCD background
- Input to the model:
  - Particle content of a jet with up to 60 tracks, 30 features per track
  - Up to 5 secondary vertex information, 14 features per secondary vertex
  - Particle-particle and particle-vertex interaction matrices create an interaction network
  - Three MLP as transformation networks:
    - fr : particle interaction
    - fr_{pv} : particle-vertex interaction
    - fo : pre-aggregator

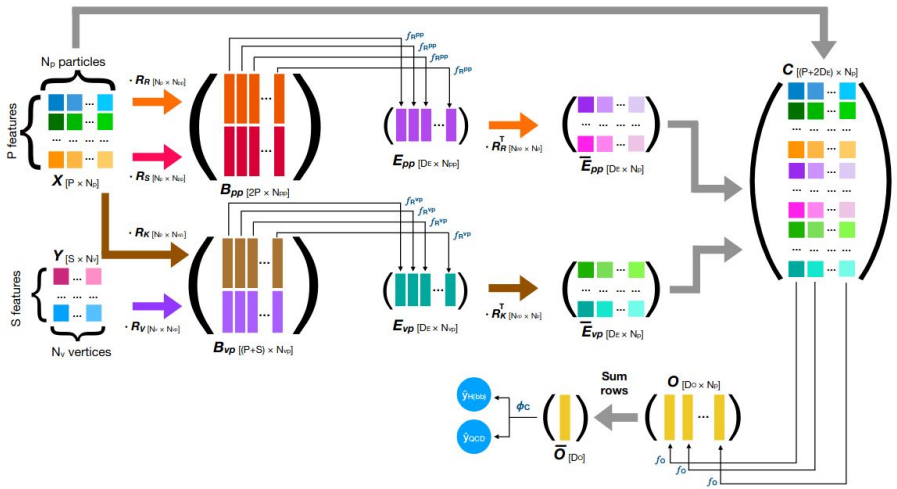
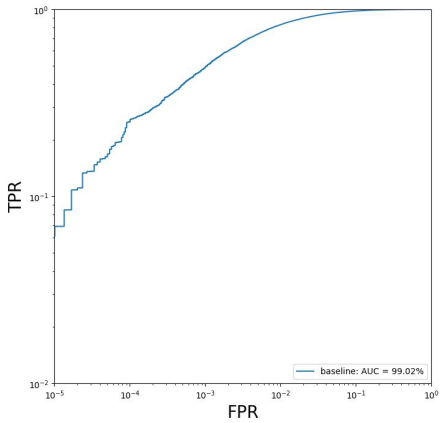


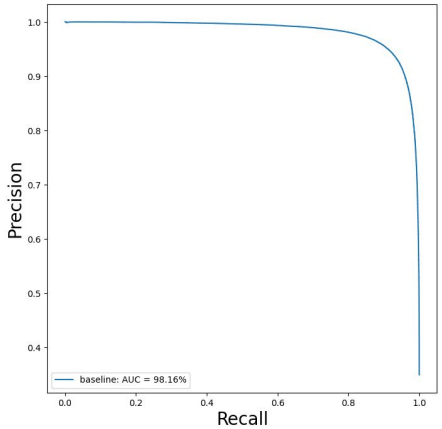
Image: https://doi.org/10.1103/PhysRevD.102.012010

Baseline Model architecture and Performance

- Each MLP has three hidden layers with 60 nodes per hidden layer
- Other hyperparameters:
  - D_e = dimension of particle-particle and particle-vertex interaction internal representations = 20
  - D_o = dimension of pre-aggregator network representation = 24
- Trained with dataset that roughly has a 2:1 distribution for QCD and Hbb jets
  - Validation accuracy of 95% (for a decision threshold of 0.5) with an ROC-AUC of 99.02%



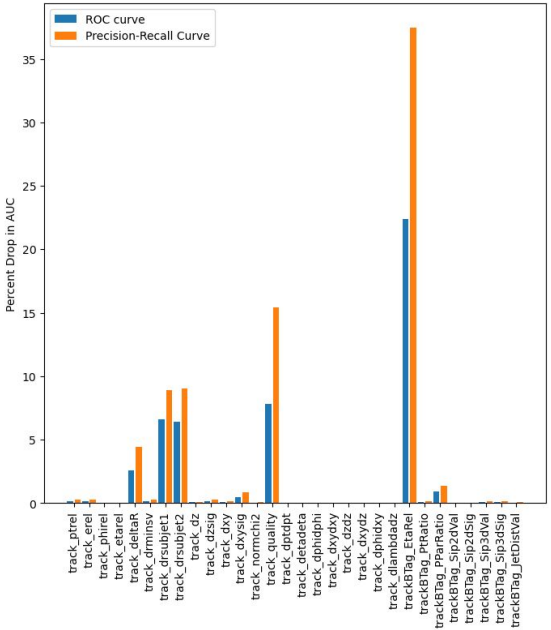
ROC curve

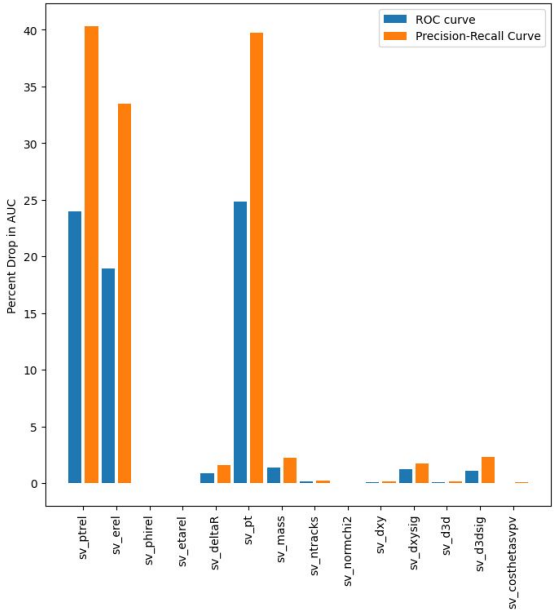


Precision Recall curve

Identifying feature importance

- Which particle and secondary vertex features play the most important roles?
- Identify by masking each feature across all nodes and evaluating the model's performance characteristic AUC for ROC and precision-recall curves

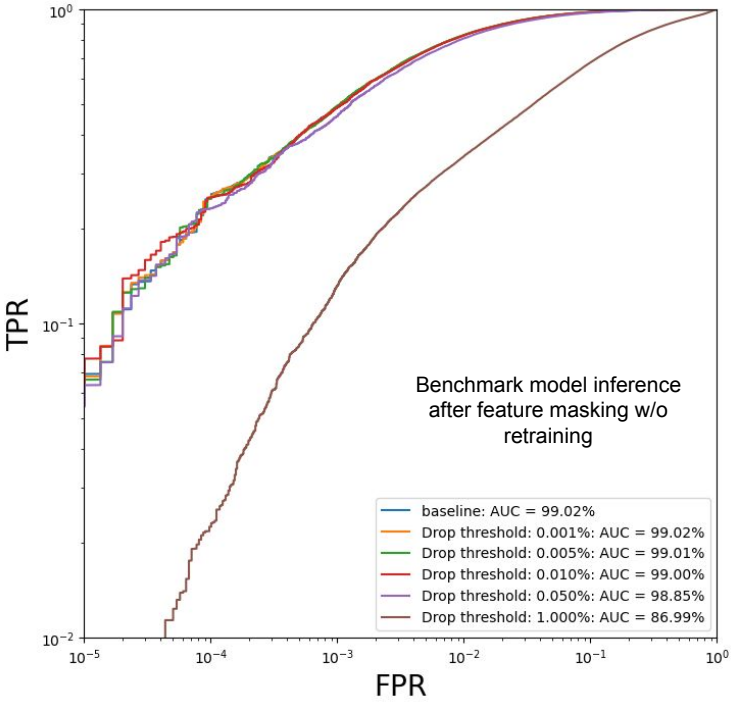




Model Reevaluation with Multiple Features Masked

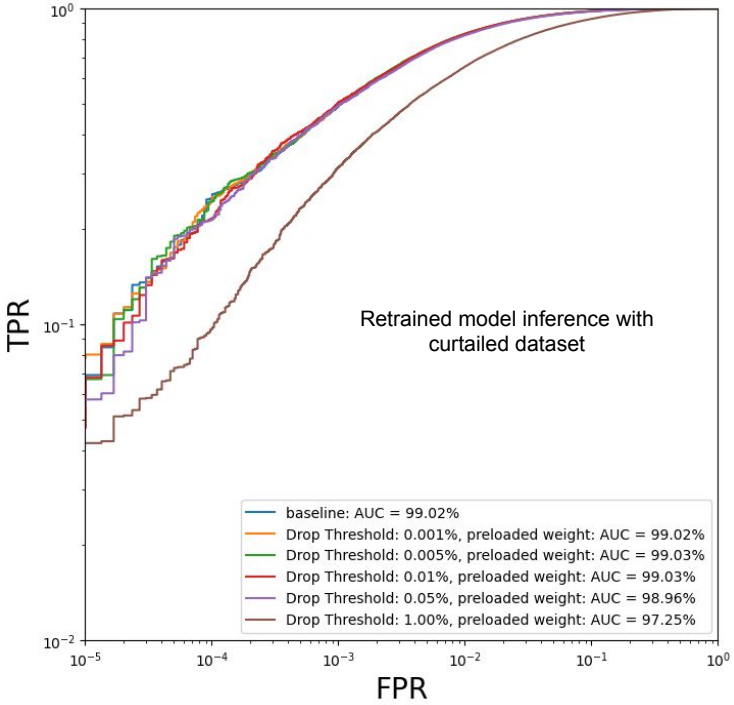
- Based on the ΔAUC measure list of *relatively unimportant* features can be found
- The model's performance was reevaluated by simultaneously dropping multiple features

ΔAUC threshold	# Particle features dropped	# Vertex features dropped
0.001%	8	2
0.005%	9	2
0.01%	11	3
0.05%	14	4
1.00%	25	8



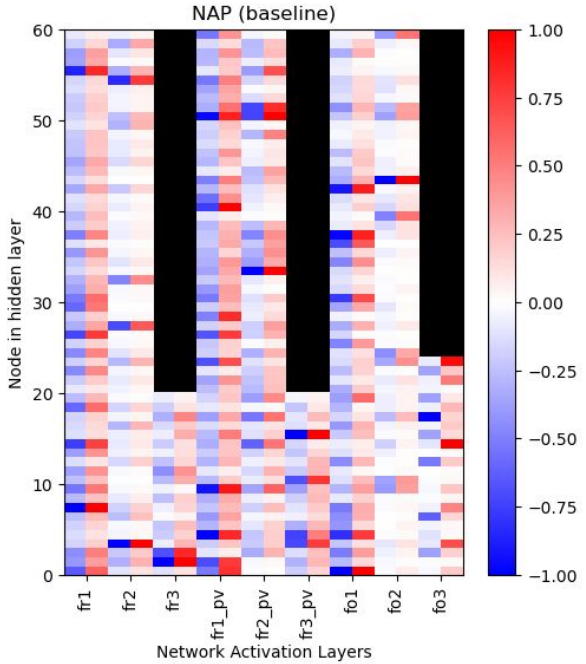
Model Retraining

- To compensate for performance loss, the model was retrained with reduced feature space
- To accelerate model training, relevant weights were preloaded from the baseline models
- Preloading allowed training to be 3x as fast
- Model performance was recovered for all cases, including the large drop case of 1% drop threshold



Neuron Activation Patterns (NAPs)

- Feature importance metrics don't reveal any information about the model's inner workings
- Understanding the model's inner workings help with hyperparameter reoptimization
- To see how the hidden layers respond to input data, we look at the distribution of activations across different nodes within a layer

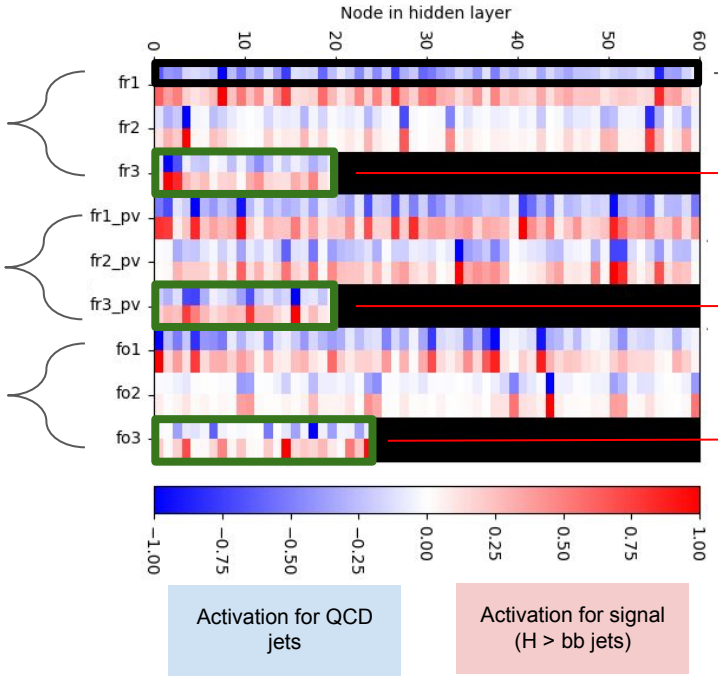


NAP: Taking a Closer Look

Hidden layers for particle interaction transformation network

Hidden layers for particle-vertex interaction transformation network

Hidden layers for pre-aggregator network



For each hidden layer, the distribution is normalized w.r.t largest aggregated neuron activation in the respective layer

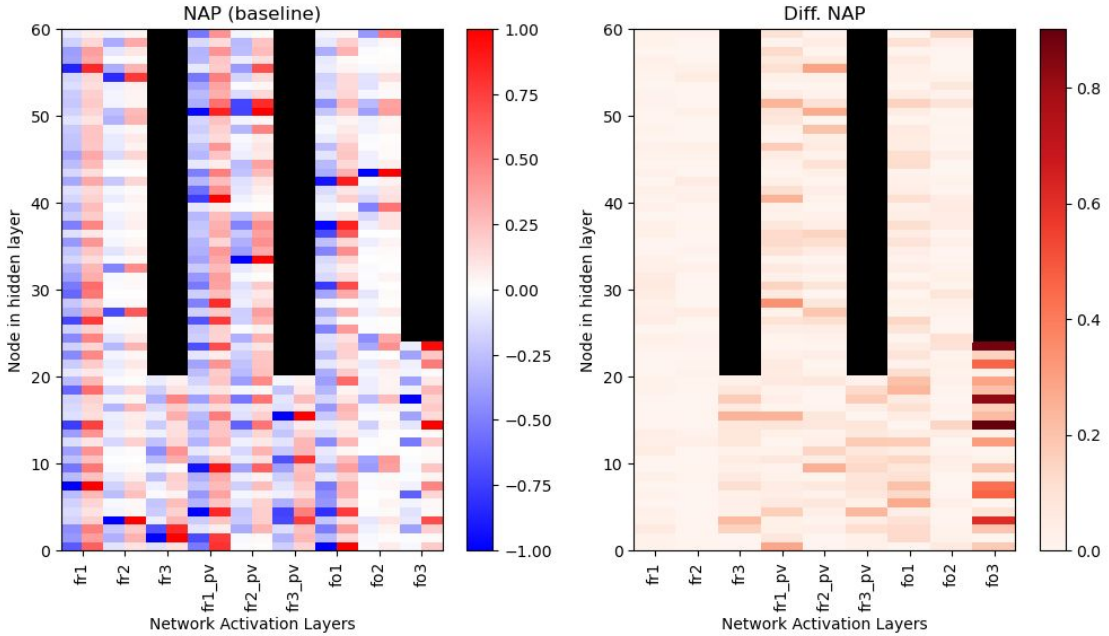
Output layer of the particle interaction transformation network, dimension D_e

Output layer of the particle-vertex interaction transformation network, dimension D_e

Output layer of the pre-aggregator network, dimension D_o

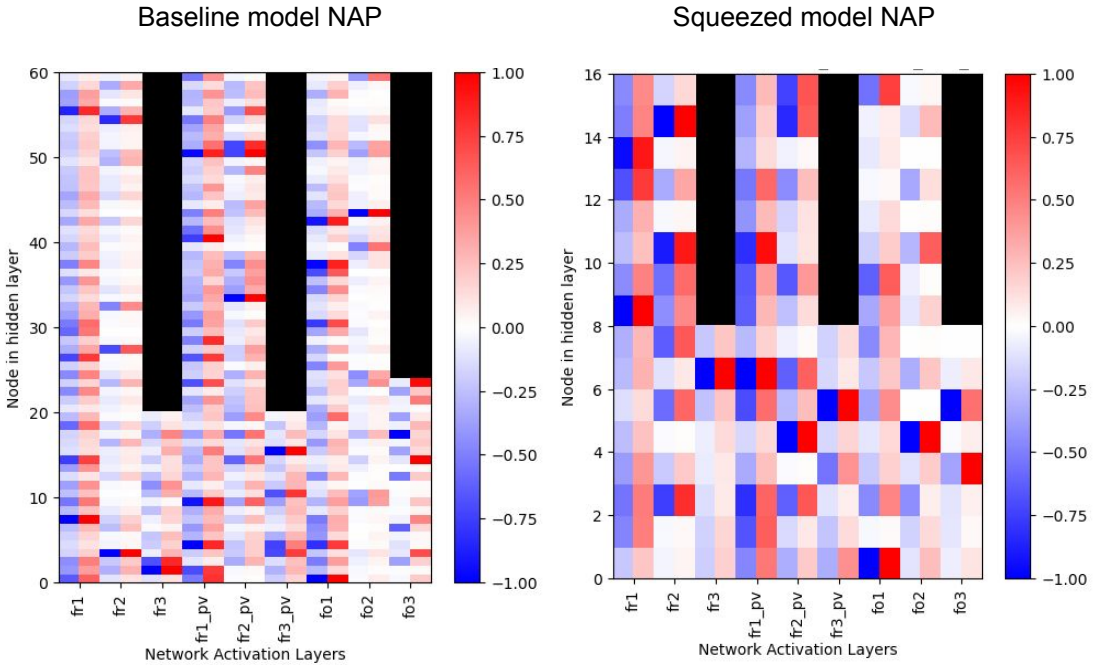
What do NAPs Tell us?

- Activation is rather sparse, largest activations at each layer is are shared by handful of nodes
  - There is scope for model simplification
- Until the very last layer (*fo3*), the activation patterns for signal and background are similar
  - Internal space distributions might not be effective classifiers



Hyperparameter Reoptimization

- NAPs reveal crucial sparsity in model's internal structure
- This can be further illustrated by comparing NAPs for a *squeezed* model:
  - Features associated with a ΔAUC < 0.05% dropped
  - 16 nodes/layer, $D_e = D_o = 8$
  - Gives an ROC-AUC of 98.65%



Summary

- Model interpretability can significantly benefit from FAIR data and AI models
- Model reusability, reliability, provenance, and metadata can benefit from interpretability
- There is a lot more to explore-
  - Instance-specific model explanation
  - Explainable local linear models
- Plan in progress:
  - FAIRification of the Interaction Network Model
  - Demonstrate how FAIR and model interpretability benefit from each other
  - Publish notebooks to demonstrate model interpretability exercises

References

1. Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* **3,** 160018 (2016). https://doi.org/10.1038/sdata.2016.18
2. Lamprecht, Anna-Lena, et al. "Towards FAIR principles for research software." *Data Science* 3.1 (2020): 37-59. https://doi.org/10.3233/DS-190026
   Chue Hong, N.P., Katz, D.S., Barker, M.*et al*. FAIR principles for research software (FAIR4RS principles). Tech. rep., Research Data Alliance (2021). https://doi.org/10.15497/RDA00065
3. Richardson, R. A., et al. "User-friendly Composition of FAIR Workflows in a Notebook Environment." *Proceedings of the 11th on Knowledge Capture Conference*. 2021. https://doi.org/10.1145/3460210.3493546
4. Katz, D. S., Psomopoulos, F. E., and Castro, L. J. "Working towards understanding the role of FAIR for machine learning." *DaMaLOS@ ISWC* (2021): 1-7. https://doi.org/10.4126/FRL01-006429415
5. Samuel, S., Löffler, F., and König-Ries, B. "Machine learning pipelines: provenance, reproducibility and FAIR data principles." Provenance and Annotation of Data and Processes. Springer, Cham, 2020. 226-230. https://doi.org/10.1007/978-3-030-80960-7_17