Explainable AI for Interpretability of Deep Neural Networks

Mark Neubauer

University of Illinois at Urbana-Champaign

Artificial Intelligence and the Uncertainty Challenge in Fundamental Physics Workshop

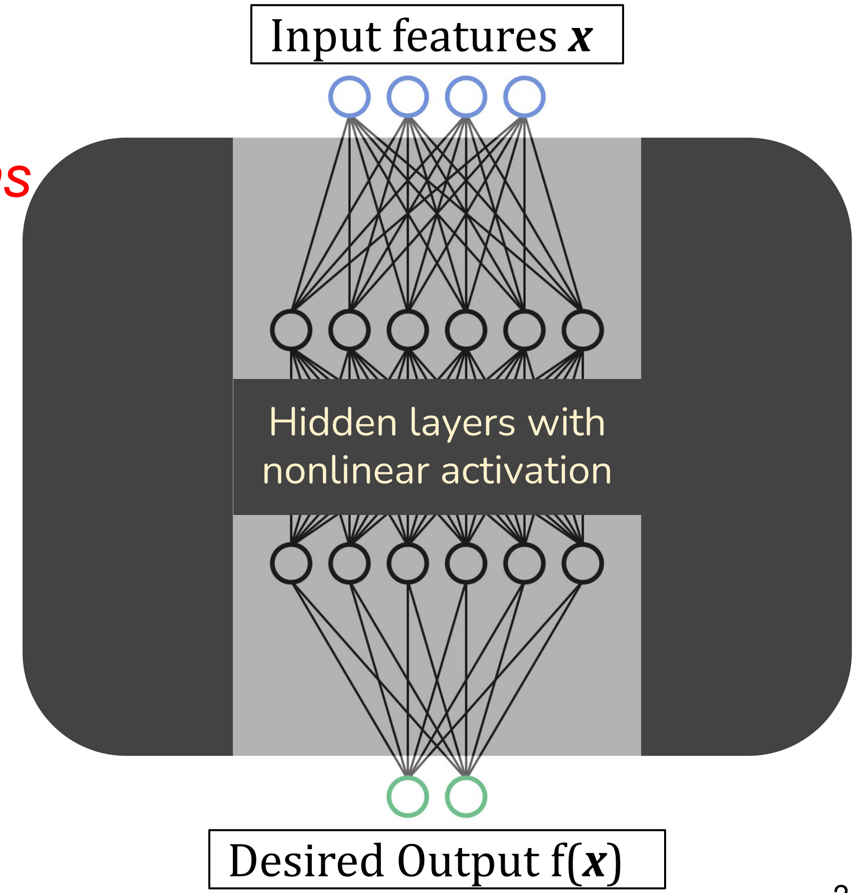
27 Nov - 1 Dec 2023

SCAI, Paris and Institut Pascal Paris-Saclay

Neural Networks are Black Boxes

- **Deep Learning** (DL) **Models** have a *large number of parameters* and *nonlinear intermediate representations*
  - ChatGPT has **1.5B** parameters
  - GPT-3(4) have **175B** (**1.5T**) parameters
- It is difficult to understand the exact WHYs (and HOWs) of Deep Neural Networks (DNNs)
  - How much does this really matter?

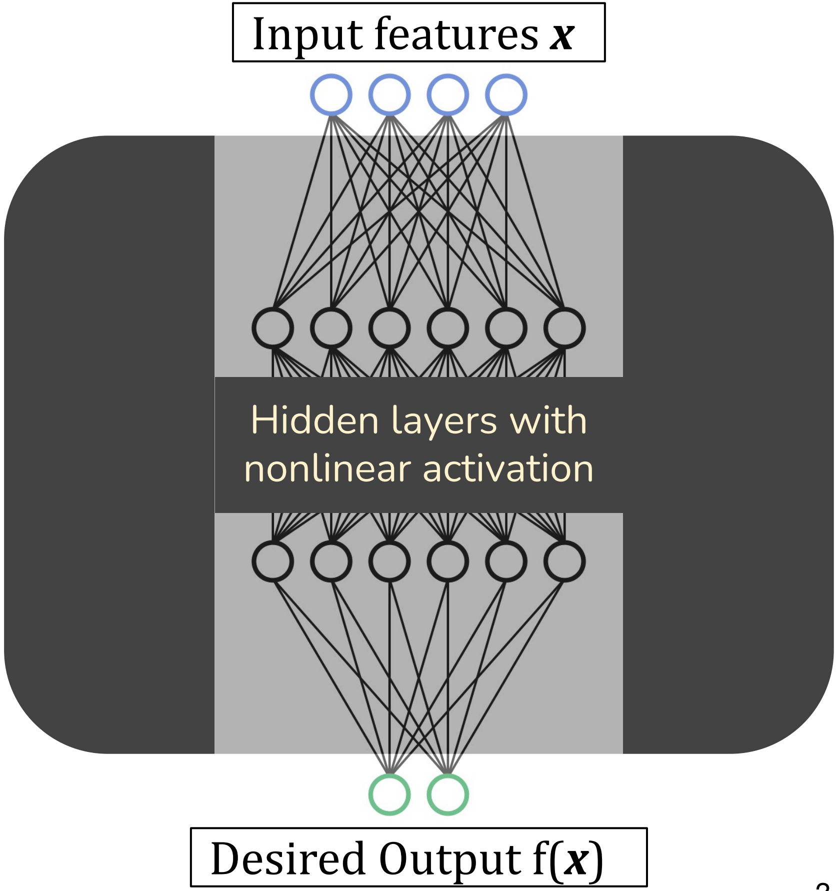
E.g. I don't know every detail about how my car works. Nor ATLAS software & detector. *Should I worry?*



Explainable Artificial Intelligence

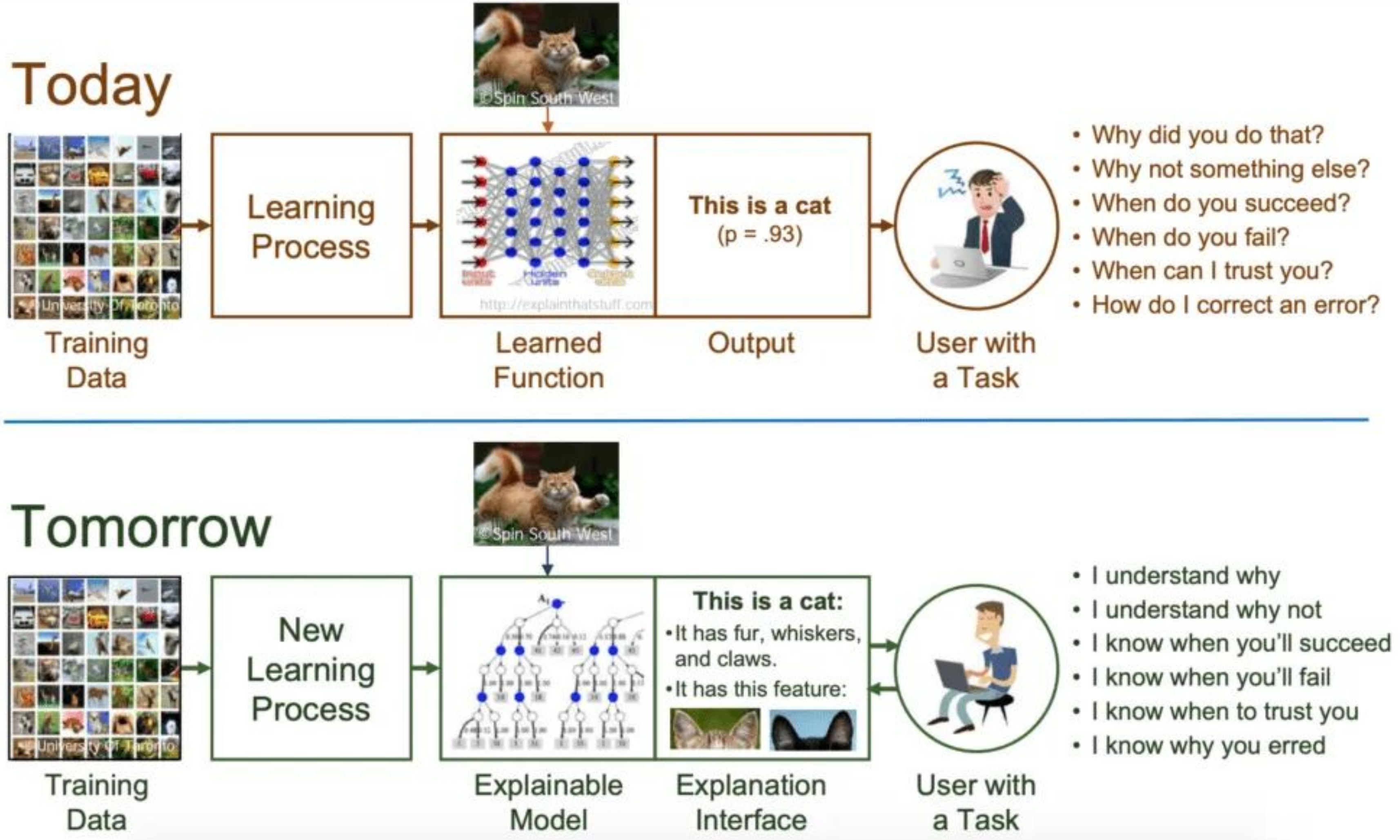
- ***Explainable AI*** (**XAI**): a set of processes and methods that allows human users to comprehend results created by AI algorithms
- With XAI, we can create AI models that are more robust against noise and adversarial samples, fair to biases in data populations, and trustworthy in terms of predictions

See talk Julian's talk in this session for a more thorough overview of XAI



Why should I care about XAI?

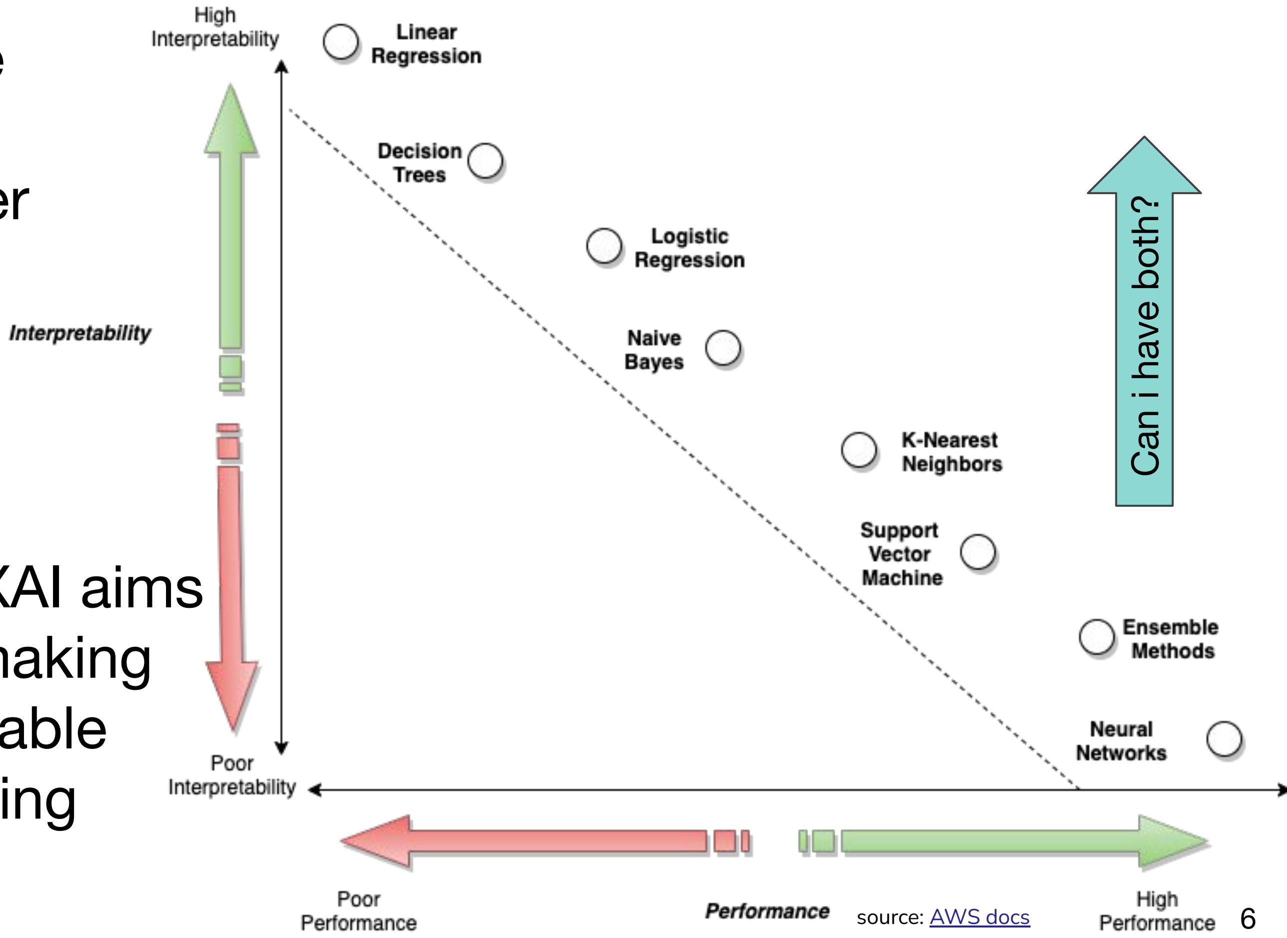
- If you want to assess or improve a DL model impacting your life
    - **Generalizibility**, **robustness**, **biases**, **trustworthiness**, **sustainability**, …
    - If things you care about (e.g. safety, health, $, scientific credibility) depend on items like ones above, you should care. Basically everyone.
- XAI methods and importance vary greatly across field of application
    - ***Methods***: No single method works for all AI applications
    - ***Importance***: Big difference between models that lives depend upon (e.g. medicine, health) vs. curiousity about how some RL game engine works
- A challenge is that XAI is hard to define and even harder to evaluate
    - No universal definition of what it means for an AI model to be explainable nor well-defined metrics to evaluate "goodness" of AI model explanations



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

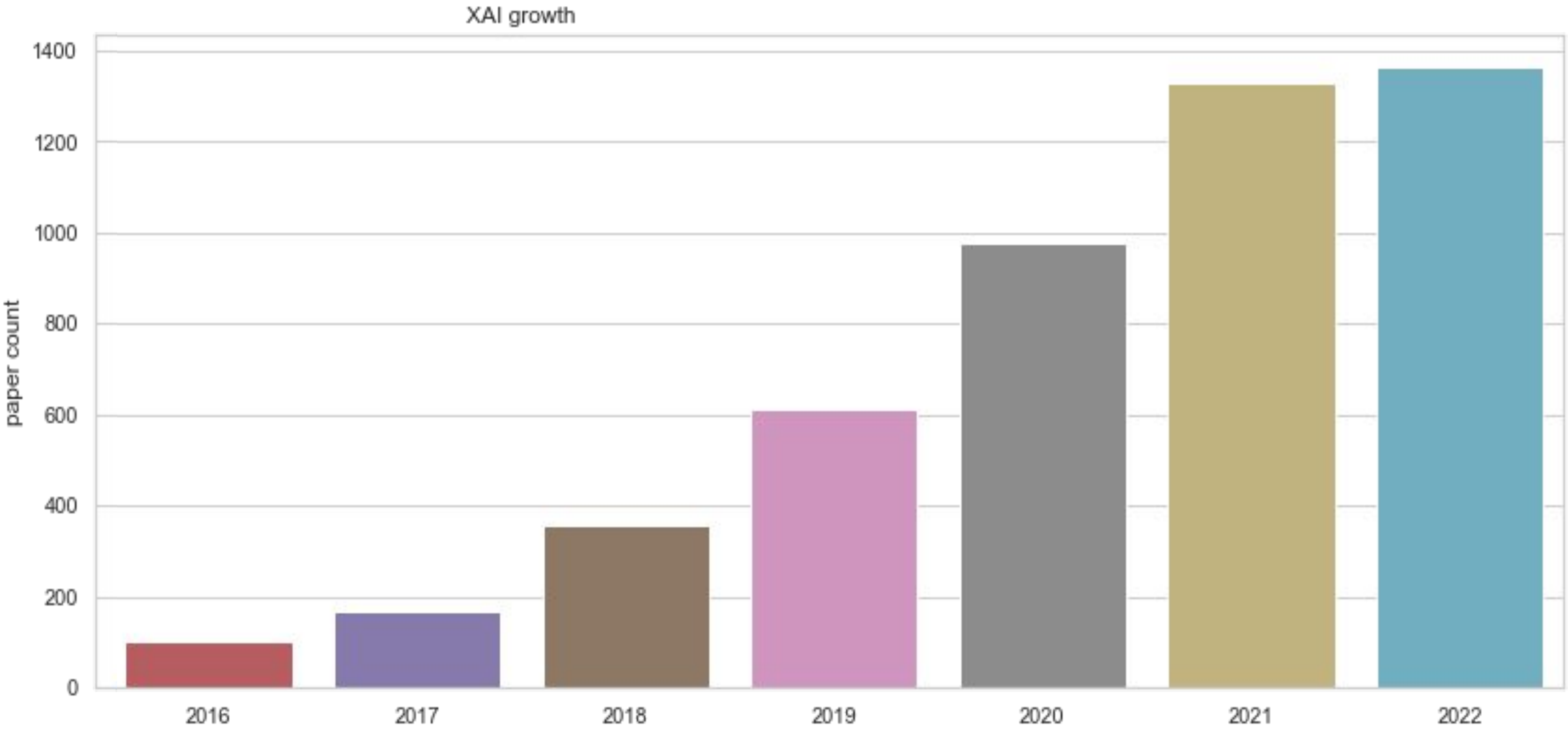
Explainabiltiy or Performance?

- Typically, explainable models are simpler and often have poorer performance
- High performing models are often too large to interpret
- Modern research in XAI aims to bridge this gap, making models more explainable while not compromising performance



source: AWS docs

XAI is an Active Area of Research



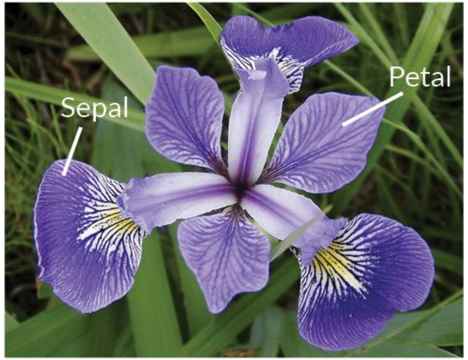
source: [This nice repo]

XAI Toolbox

- Simpler AI methods like linear regression or decision trees are generally highly explainable
- Other techniques include:
  - Performance deviation based methods
  - Local linear surrogate models
  - Feature importance attribution
  - Principal Component Analysis (for neural embeddings)
  - White box models
  - Occlusion test with **ΔAUC score**
  - **Shapely Additive Explanations** (SHAP)
  - **Layerwise Relevance Propogation** (LRP)
  - *And many more…*
    - *E.g. See the **Mean Absolute Differential Relevance** (MAD) Score and **Neural Activation Patterns** diagrams developed as part of our work*

XAI Warm-Up

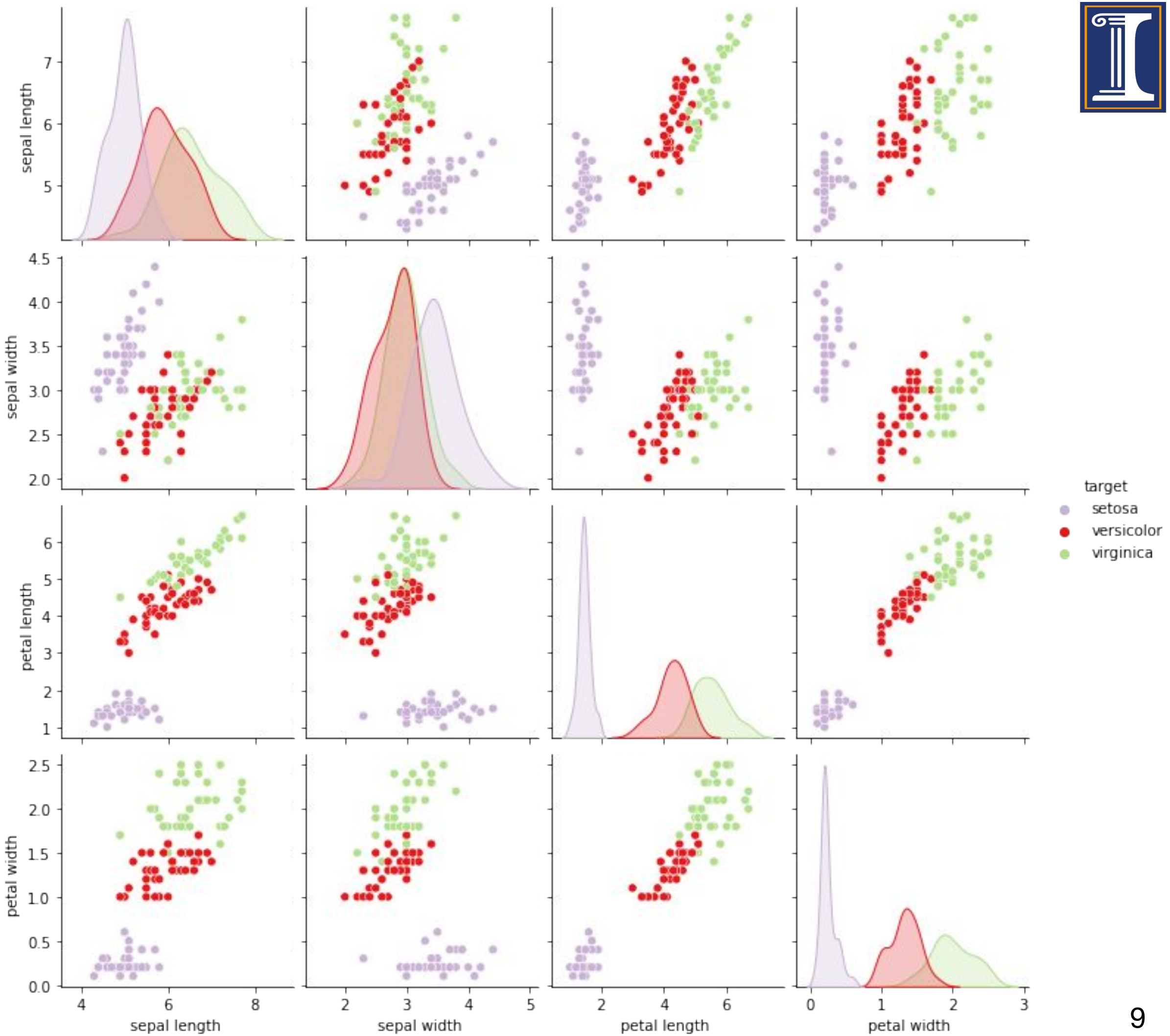
(with the IRIS dataset)



Iris Versicolor

Iris Setosa

Iris Virginica



Training an Explainable Model - I

Logistic Regression

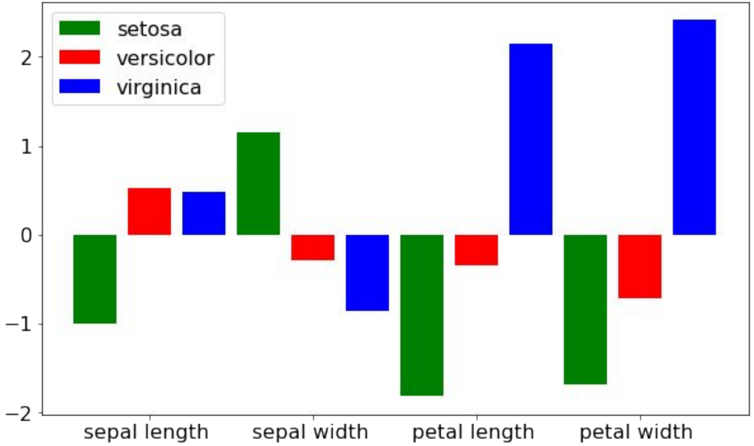
- Predicting class probabilities with a linear model to predict the logit for each class

$$f_k(\vec{x}) = \beta_{k,0} + \sum_i \beta_{k,i} x_i$$

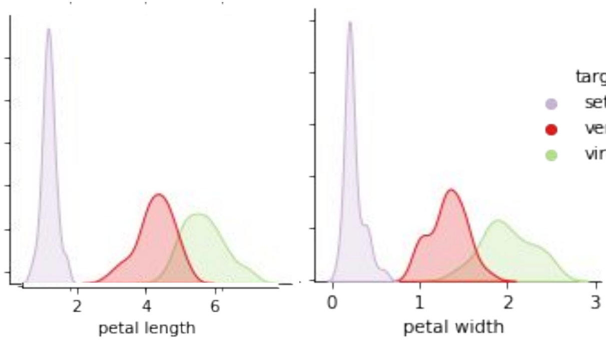
$$p_k(\vec{x}) = \frac{\exp f_k(\vec{x})}{\sum_j \exp f_j(\vec{x})}$$

- The dataset is standardized to get rid of "scale effects"
- The β coefficients tell you which feature is important for which class

virginica has larger petal length and widths



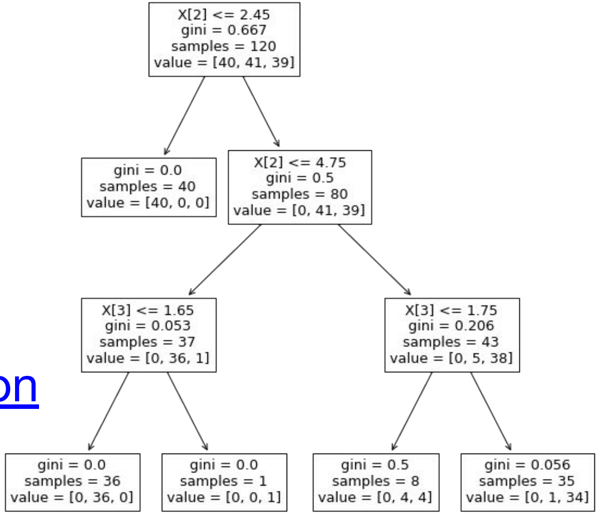
setosa has smaller petal length and widths

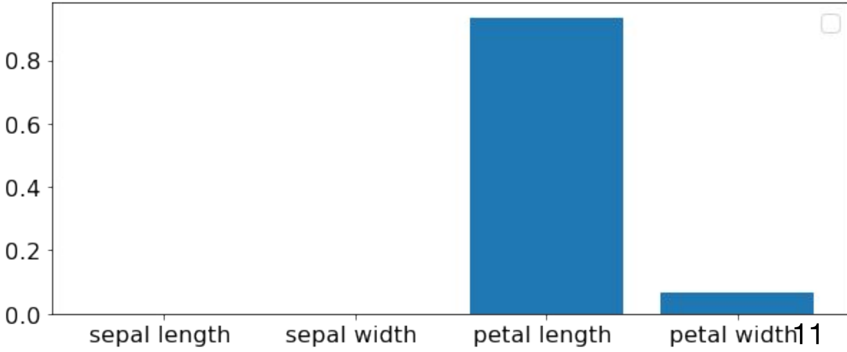


Training an Explainable Model - II

Decision Tree

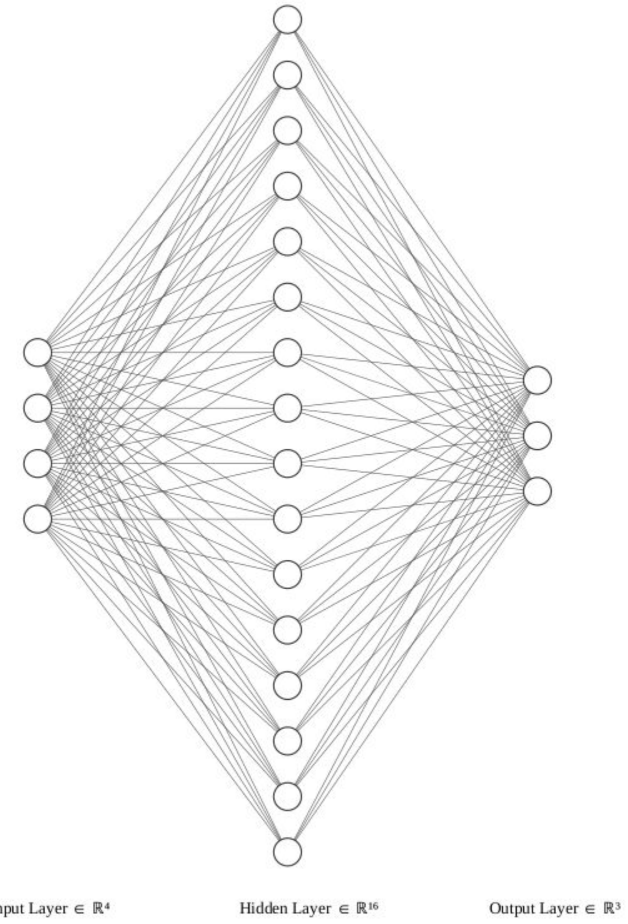
- A decision tree splits the dataset according to some threshold on the features
- No standardization is typically needed
- For smaller trees, the decision diagram can be visualized
- Also gives a “feature importance” list based on [permutation feature importance]()
  - Decrease in a model score when a single feature value is randomly shuffled
  - Breaks the relationship between the feature and the target → the drop in model score is indicative of how much the model depends on the feature





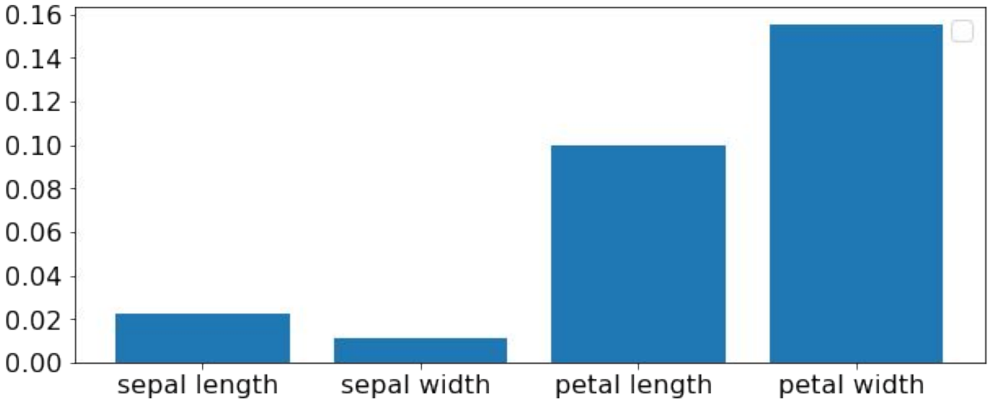
A Neural Network Based Model

- Using a simple, fully connected multi-layer perceptron (MLP)
  - 4 inputs (standardized)
  - 1 hidden layer with 16 nodes
  - ReLU activation
  - 3 outputs with a softmax layer to predict class probabilities
  - 131 trainable parameters
- Trained with Adam optimizer with a learning rate 0.01
- Optimized by minimizing Cross-entropy loss



Explaining the Neural Network - I

Performance Deviation Metric



- One quick way to check how much impact each feature has is to quantify the model's performance (e.g. prediction accuracy) degradation
- Each input is replaced by its population mean as a mask value
- Change in accuracy is assigned as the importance of the corresponding feature

Two major limitations:

- Does not offer local (i.e. sample-wise) explanation
- Large changes in predictions [e.g. (0.8, 0.1, 0.1) → (0.6, 0.2, 0.2)] may be ignored when class assignments are identical

Explaining the Neural Network - II

Shapely Additive Explanations *(SHAP)*

- A game theoretic approach to explain the contribution of each feature [1705.07874]

$x \rightarrow z \in \{0,1\}^M$ Binary representation of input based on if a feature is considered

True model output

$$\boxed{f(x) \approx g(z) = \phi_0 + \sum_i \phi_i(x) z_i}$$

Surrogate model output

Model average

Feature contributions

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S) \right]$$

Explaining the Neural Network - II

Shapely Additive Explanations (SHAP)

$$f(x) \approx g(z) = \phi_0 + \sum_i \phi_i(x) z_i$$

The entire procedure is very expensive, requires iterating over 2^M subsets for each sample. Simplifying techniques have been introduced by the authors to make it work faster

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}(x_{S \cup \{i\}}) - f_S(x_S) \right]$$

Consider each subset of the input features that don't include feature *i*

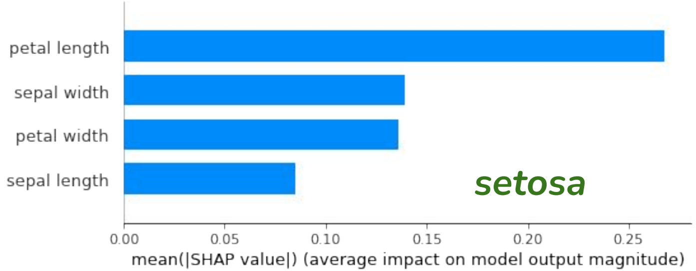
Functional value after the feature is included

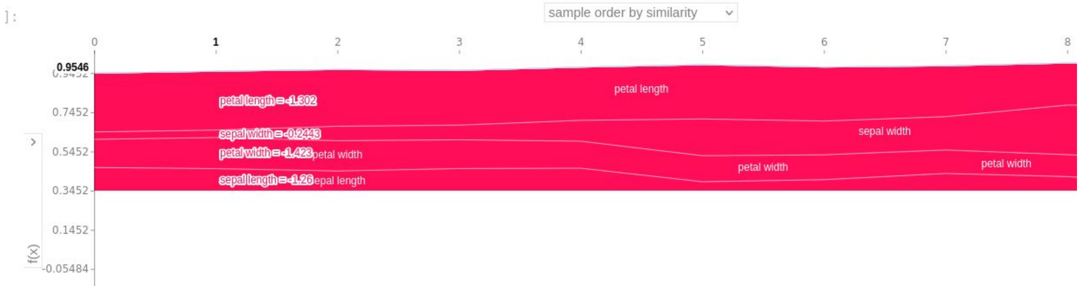
Functional value w/o the feature

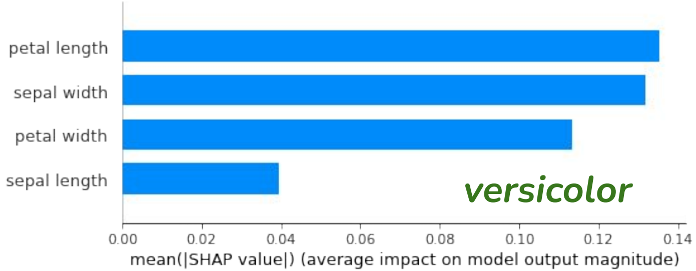
SHAP library: https://github.com/slundberg/shap

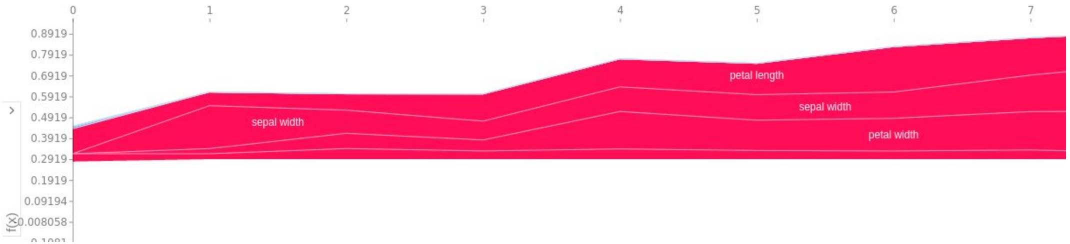
Explaining the Neural Network - II

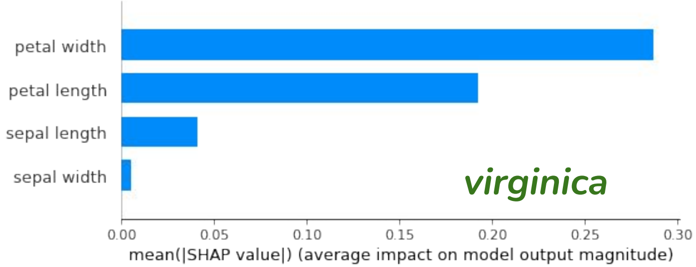
Shapely Additive Explanations (SHAP)









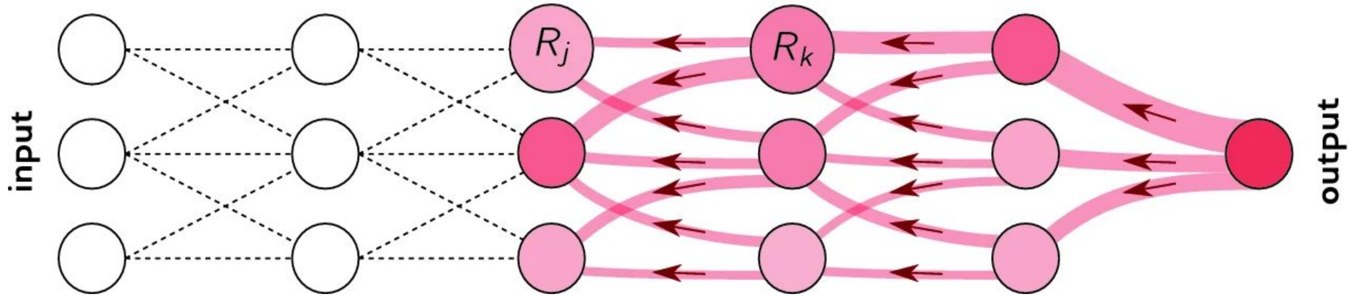


Explaining the Neural Network - III

Layerwise Relevance Propogation *(LRP)*

- Backpropagate the output of a NN according by linearly redistributing it to the nodes in the previous layers - eventually assigning relevance scores to each of the inputs

https://doi.org/10.1007/978-3-030-28954-6_10



$$r_j^{(n)} = \sum_k \frac{a_j^{(n)} w_{jk:n}}{\sum_m a_m^{(n)} w_{mk:n}} r_k^{(n+1)}$$

Explaining the Neural Network - III

Layerwise Relevance Propogation (LRP)

$$r_j^{(n)} = \sum_k \frac{a_j^{(n)} w_{jk:n}}{\sum_m a_m^{(n)} w_{mk:n}} r_k^{(n+1)}$$

Contribution of *j*-th node in layer *n* towards *k*-th node in layer *n* + 1

Relevance of node *j* in layer *n*

Summing over all nodes in layer *n+1*

Activation of m-th node in layer *n*

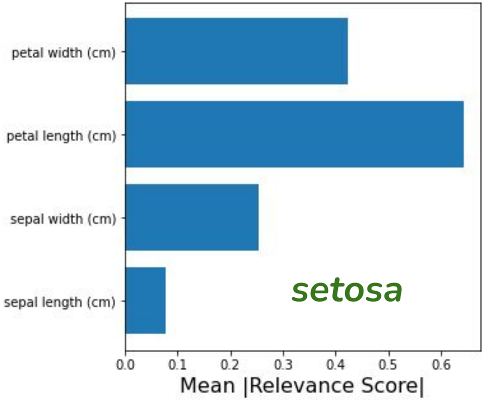
Weight of *m*-th node in layer *n* towards *k*-th node in layer *n* + 1

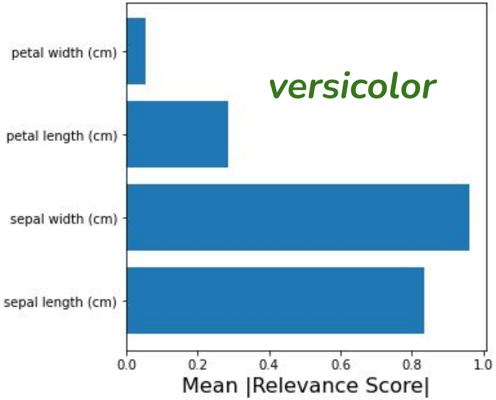
Relevance of node *k* in layer *n+1*

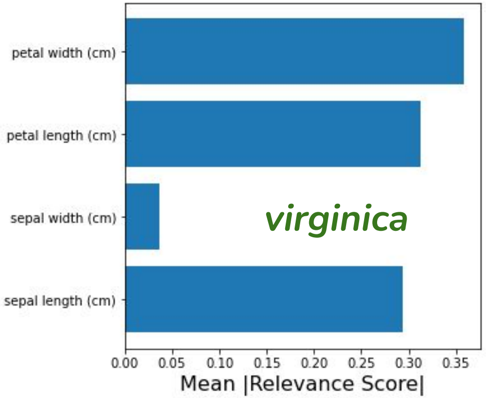
Total contribution of all nodes in layer *n* towards *k*-th node in layer *n* + 1

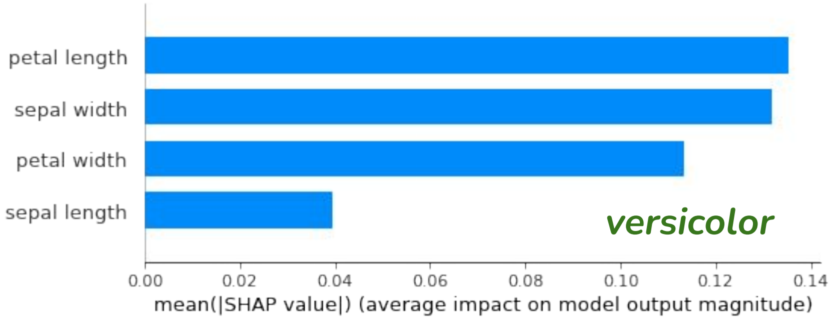
Explaining the Neural Network - III

Layerwise Relevance Propogation (LRP)









Difference between relevance and SHAP scores - why ?

Understanding the LRP Results

- *Why SHAP can be quite different from LRP?*

SHAP calculates "Deviation" from mean-behavior

It represents the impact of including the true value of a feature compared to the mean or an *uninformative* value

LRP score includes mean-behavior relevance!!

$$f(\vec{x}) = \sum_i r(x_i) \approx f(\vec{x}_{\backslash k}) + \frac{\partial f}{\partial x_k}(x_k - \bar{x}_k)$$

$$\vec{x}_{\backslash k} = \vec{x}\backslash\{x_k\} \bigcup \{\mathbf{E}(X_k)\}$$

Modified input where the *k*-th feature is replaced by its mean value

Differential Relevance Score

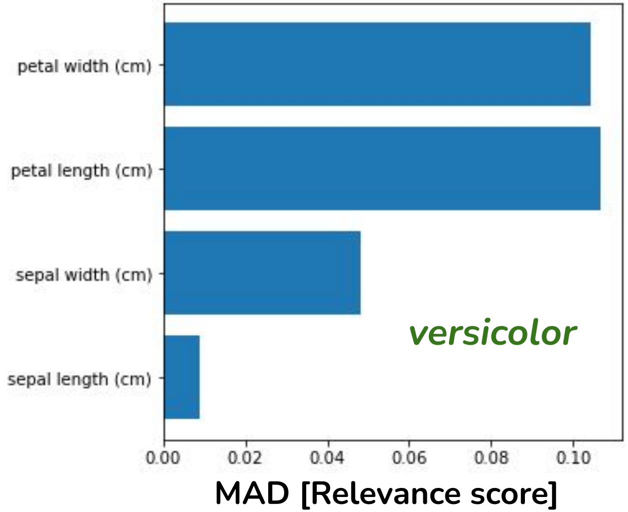
2210.04371

$$f(\vec{x}) = \sum_i r(x_i) \approx f(\vec{x}_{\setminus k}) + \frac{\partial f}{\partial x_k}(x_k - \bar{x}_k)$$

Differential relevance

$$f(\vec{x}_{\setminus k}) = \sum_{i \neq k} r(x_i) + r(\bar{x}_k)$$

Mean-behavior relevance



- When features are uncorrelated (or weakly correlated), calculate mean-behavior relevance by simply replacing all features by their mean value and then calculating their relevances
- Differential relevance is more exact, determined by calculating the deviation in model's output when a particular feature is replaced by its mean value

MAD relevance:
Mean Absolute Differential Relevance

Has a stronger resemblance with the SHAP scores since this takes the "deviations" into account

(Actually, diff. Rel. is one of the leading terms that contribute to SHAP score)

Now let's look at some more challenging data!

Results from arxiv: 2210.04371
Published in 2023 *Mach. Learn.: Sci. Technol.* 4 035003
Git repo: https://github.com/FAIR4HEP/xAI4toptagger/

Ayush Khot

Avik Roy

Mark Neubauer

This work was supported by the FAIR Data program of the DOE ASCR under contract number DE-SC0021258 and by DOE OHEP, under contract number DE-SC0023365

U.S. DEPARTMENT OF ENERGY | Office of Science

FAIR4HEP

A Detailed Study of Interpretability of Deep Neural Network based Top Taggers

Ayush Khot, Mark S. Neubauer, and Avik Roy[1]

Department of Physics & National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign

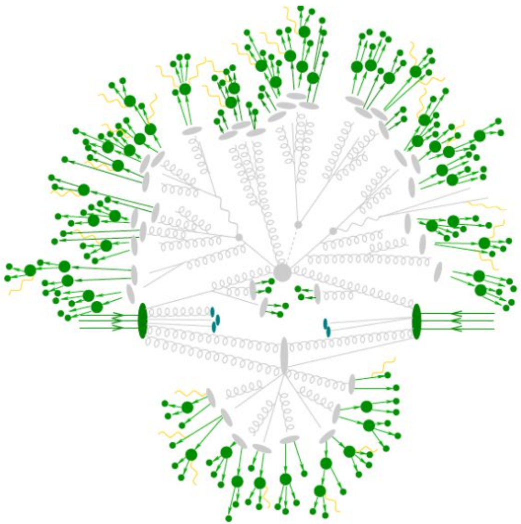
E-mail: akhot2@illinois.edu, msn@illinois.edu, avroy@illinois.edu

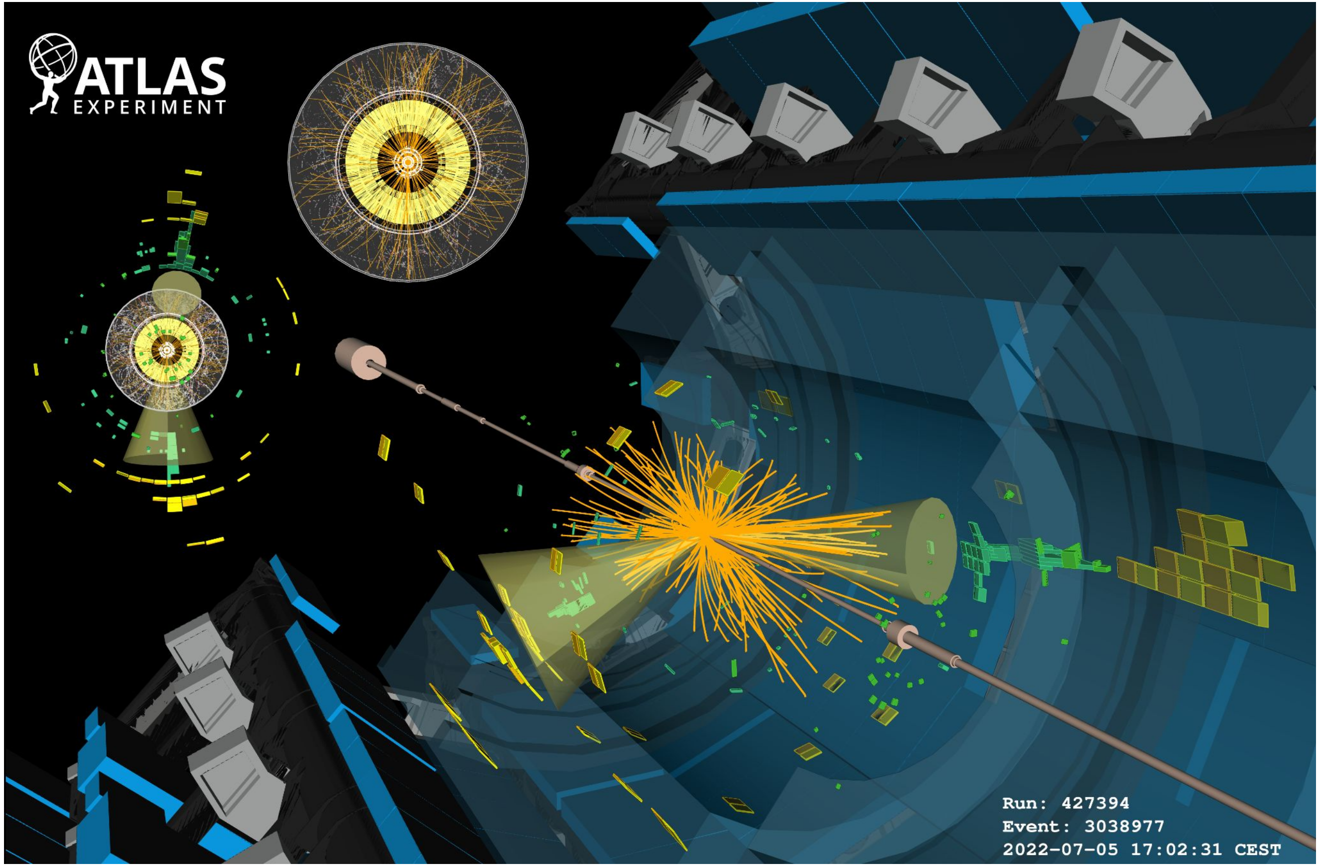
Abstract: Recent developments in the methods of explainable AI (XAI) allow researchers to explore the inner workings of deep neural networks (DNNs), revealing crucial information about input-output relationships and realizing how data connects with machine learning models. In this paper we explore interpretability of DNN models designed to identify jets coming from top quark decay in high energy proton-proton collisions at the Large Hadron Collider (LHC). We review a subset of existing top tagger models and explore different quantitative methods to identify which features play the most important roles in identifying the top jets. We also investigate how and why feature importance varies across different XAI metrics, how correlations among features impact their explainability, and how latent space representations encode information as well as correlate with physically meaningful quantities. Our studies uncover some major pitfalls of existing XAI methods and illustrate how they can be overcome to obtain consistent and meaningful interpretation of these models. We additionally illustrate the activity of hidden layers as Neural Activation Pattern (NAP) diagrams and demonstrate how they can be used to understand how DNNs relay information across the layers and how this understanding can help to make such models significantly simpler by allowing effective model reoptimization and hyperparameter tuning. These studies not only facilitate a methodological approach to interpreting models but also unveil new insights about what these models learn. Incorporating these observations into augmented model design, we propose the Particle Flow Interaction Network (PFIN) model and demonstrate how interpretability-inspired model augmentation can improve top tagging performance.

Jets at the Large Hadron Collider

Colliding protons at the LHC produce collections of particles called “**jets**”

- Observed as clusters of energy and tracks





Jet Tagging: Classification in HEP

- Jets can emerge from many processes, and we want to identify the “type” of the process that gives us the jet
- Classic example from HEP: QCD and top jet classification
- Includes information about momenta (p_x, p_y, p_z) and energy (e) of up to 200 particles that make up the jet
- The total energy (momentum) of the jet is obtained by a scalar (vector) summation of the particle-level

Transverse momentum

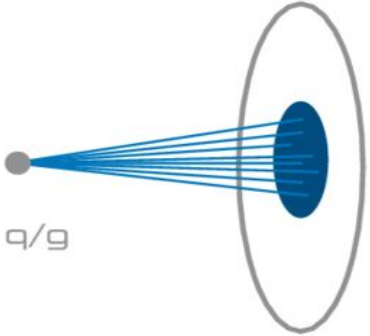
$$p_t = \sqrt{p_x^2 + p_y^2}$$

Azimuthal angle

$$\phi = tan^{-1}\left(\frac{p_y}{p_x}\right)$$

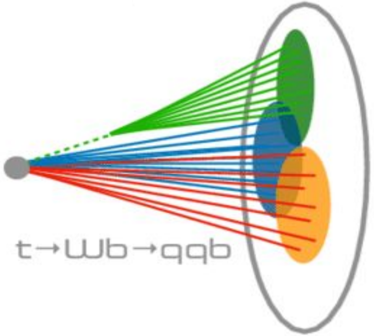
pseudorapidity

$$\eta = \frac{1}{2}\ln\left(\frac{e+p_z}{e-p_z}\right)$$



Jets from light quarks/gluons: QCD jets

Simulated dataset with 2M jets available at: zenodo: 2603256



Jets from top quarks: top jets

TopoDNN

- Simplest DNN architecture, implemented with an MLP with multiple hidden layers
- Uses p_T, η, ϕ of top 30 (p_T ordered) jet constituents - zero padding for missing entries
- Data is pre-processed to
  - align the highest p_T constituent along (0,0) in η-ϕ
  - align the second highest p_T constituent along the negative ϕ axis
  - scale the p_T values by 1/1700

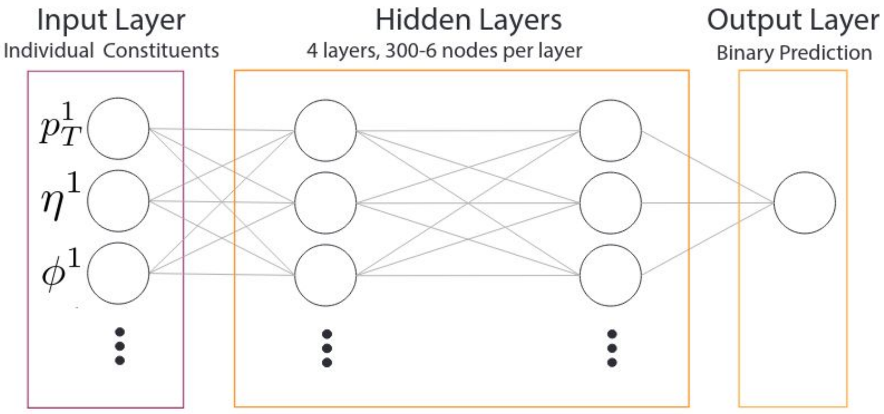


Image from 1704.02124

Baseline Architecture	
N_{in}	90 (= 3 x 30)
N_{out}	1
Hidden Layers	(300, 102, 12, 6)
Accuracy	91.6%
ROC-AUC	0.971

Explainability Methods Explored

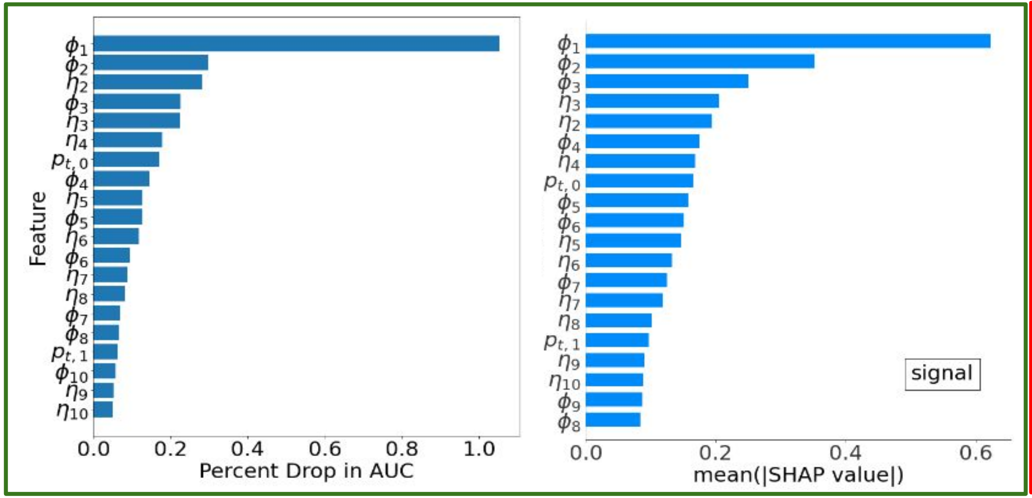
- ***Occlusion test with ΔAUC score***
  - Feature ranking based on replacing certain features with their mean values and calculating the change in model's ROC-AUC score
- ***SHAP scores***
  - Use the model-agnostic Kernel SHAP approach to identify the weighted marginal contribution of each feature
- ***Layer-wise Relevance Propagation***
  - Back propagates the score from the final output layer to original inputs using a linear redistribution
- ***Neural Activation Patterns***
  - **Relative Neural Activity** (RNA) at each node and visualises information pathways along with model's sparsity

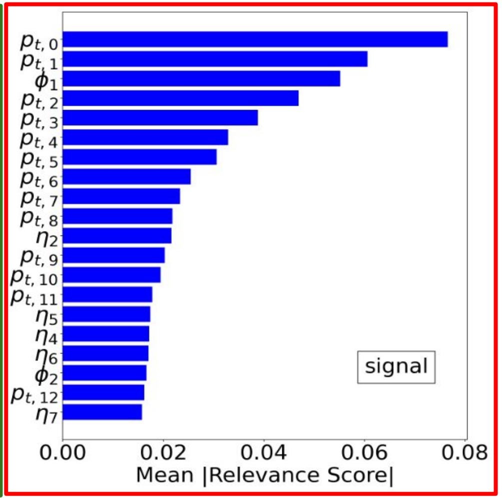
	ΔAUC	SHAP	LRP	RNA/NAP
Scalability in input dimension	✗	✗	✓	✓
Local explanation	✗	✓	✓	✗
Global explanation	✓	✓	✓	✓
Requires Forward Propagation	✓	✓	✓	✓
Requires Backward Propagation	✗	✗	✓	✗
Susceptible to spurious correlations	✓	✓	✓	✗
Addresses Model Complexity	✗	✗	✗	✓
Requires Retraining	✗	✗	✗	✗

New!

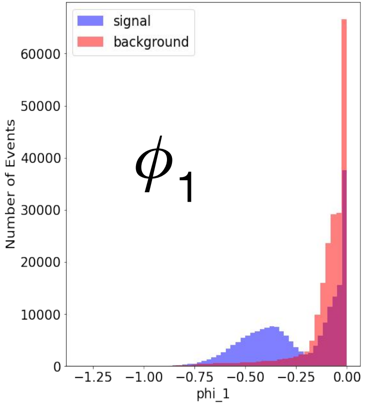
$$\boxed{\mathrm{RNA}(j,k;S) = \frac{\sum_{i=1}^{N} a_{j,k}(s_i)}{\max_j \sum_{i=1}^{N} a_{j,k}(s_i)}}$$

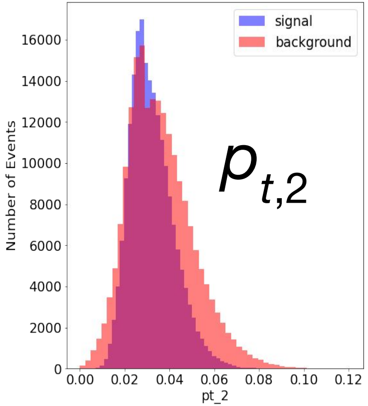
Feature Importance in TopoDNN





Why are results from LRP so different and assign large scores to non-expressive features?





Differential Relevance Score

2210.04371

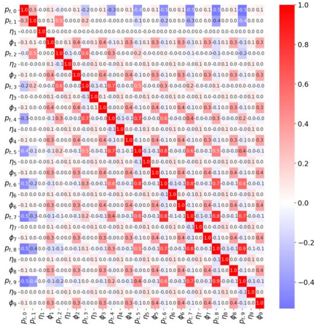
$$f(\vec{x}) = \sum_i r(x_i) \approx f(\vec{x}_{\setminus k}) + \frac{\partial f}{\partial x_k}(x_k - \bar{x}_k)$$

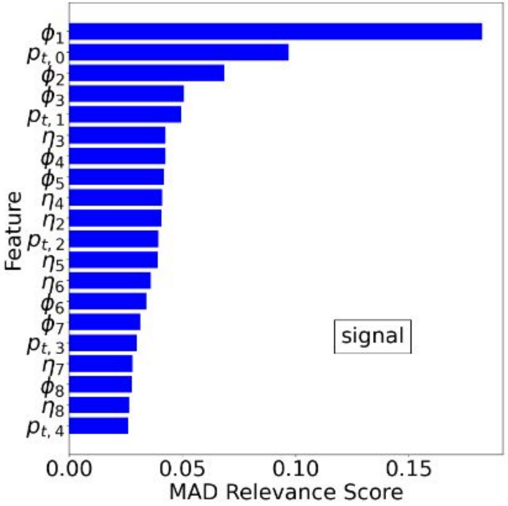
Differential relevance

$$f(\vec{x}_{\setminus k}) = \sum_{i \neq k} r(x_i) + r(\bar{x}_k)$$

Mean-behavior relevance

Feature correlation for tops





- When features are uncorrelated (or weakly correlated), calculate mean-behavior relevance by simply replacing all features by their mean value and then calculating their relevances
- Differential relevance is more exact, determined by calculating the deviation in model's output when a particular feature is replaced by its mean value

MAD relevance:
Mean Absolute Differential Relevance

Has a stronger resemblance with the SHAP scores since this takes the "deviations" into account

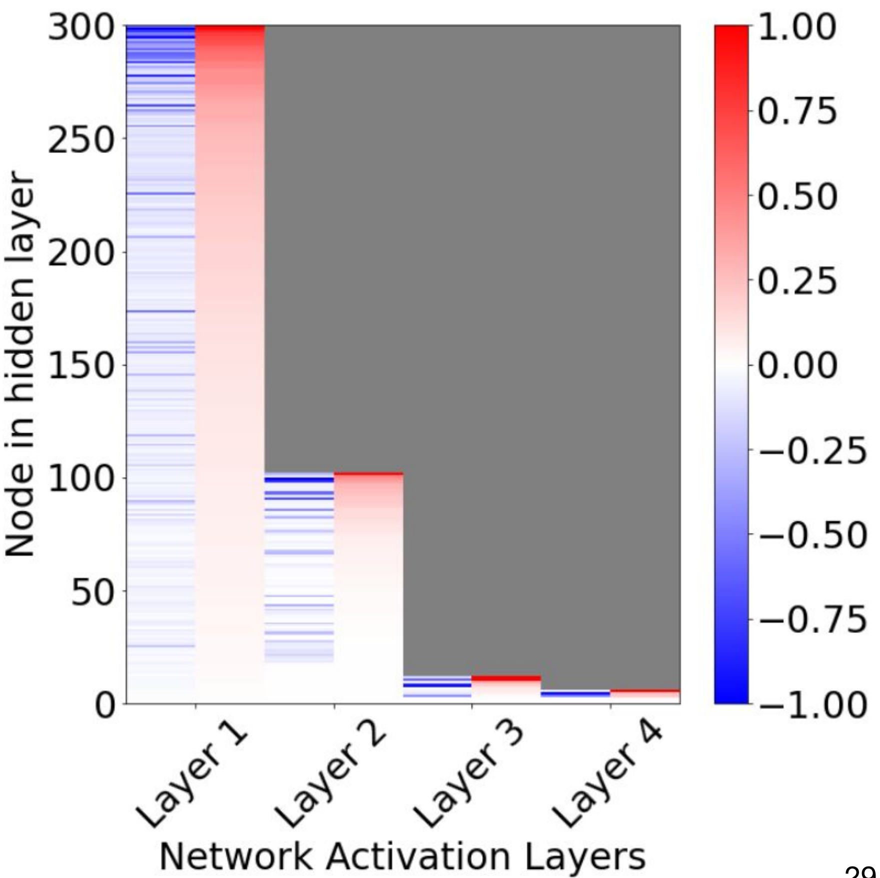
(Actually, diff. Rel. is one of the leading terms that contribute to SHAP score)

Neuron Activation Pattern (NAPs)

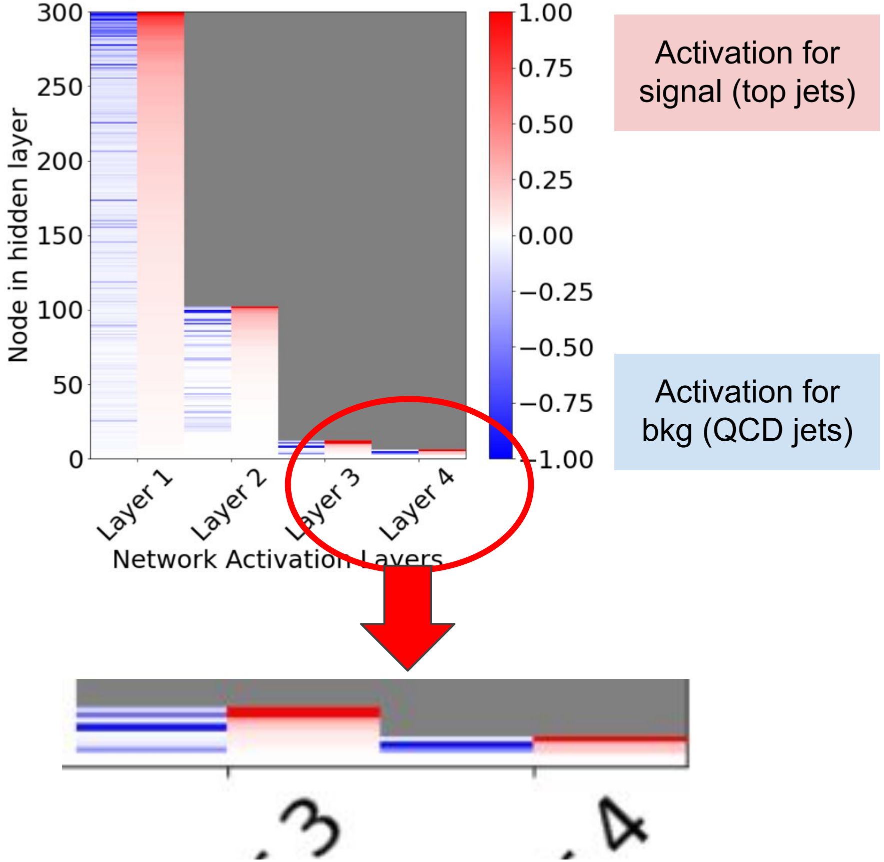
- Feature importance metrics don't reveal information about model's inner workings
- Want to detect internal disentanglements, context-aware neural pathways, hyperparameter reoptimization
- Define a **Relative Neural Activity** (RNA) score for different nodes within a layer

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

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



NAP Diagram for TopoDNN



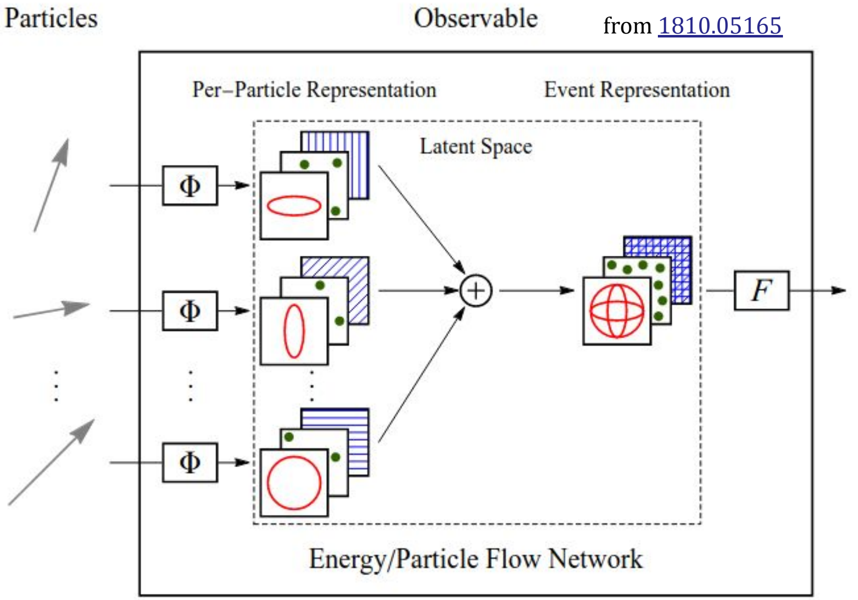
- RNA scores of QCD jets mapped as negative numbers for simultaneous visualization
- Observations
  - The model is very sparse
  - The information pathways for jet classes are disentangled by layer 3, layer 4 is kind of redundant
- **Retrained the model with (120,40,6) hidden nodes, got same performance**

Particle Flow Network (PFN)

- Deep-set architecture, designed to be invariant under permutation of constituents

$$\mathrm{PFN} = F\left(\sum_{i=0}^{N-1} \Phi\left(p_i\right)\right)$$

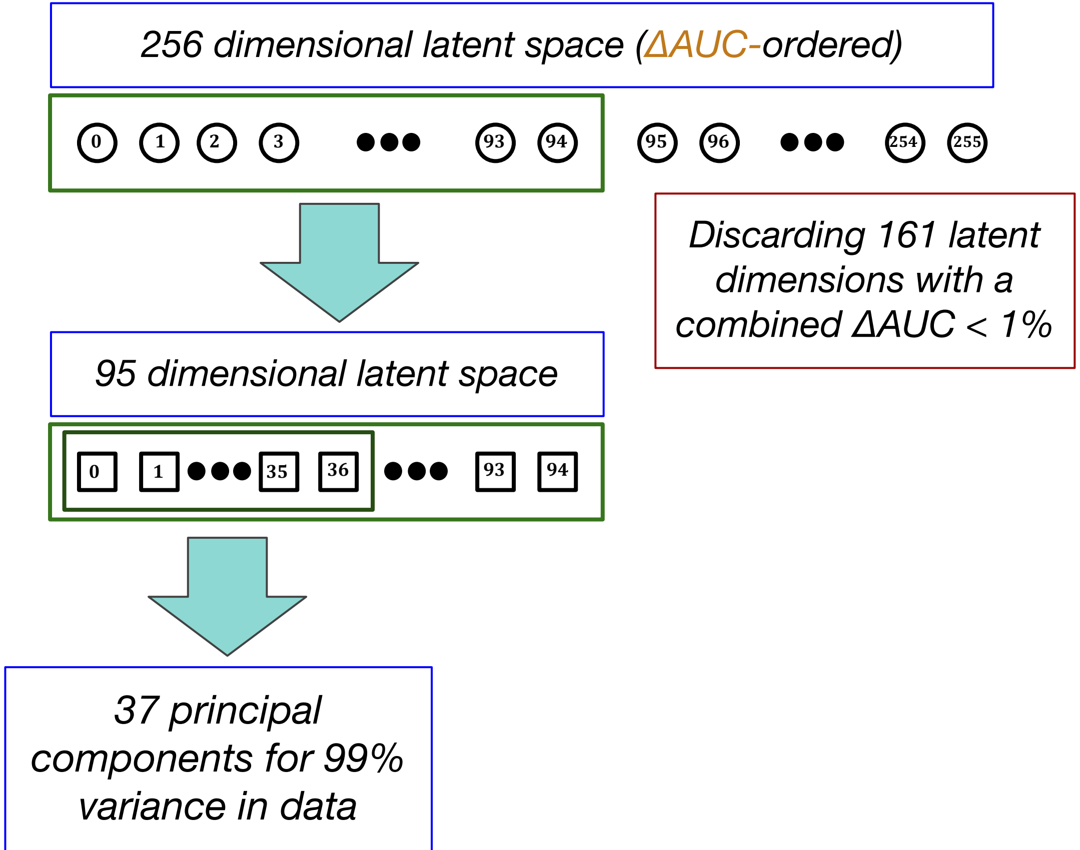
- Use MLPs to approximate the non-linear functions Φ and F
- Obtain and analyze ***latent space representation*** for jet-level observables



Baseline Architecture and Performance	
N_{in}, N_{out}	Φ: 3,256 *F:* 256,2
Layers	Φ: (3,100,100,256) F: (256,100,100,100,2)
Accuracy, AUC	92.8%, 0.980

Disentangling Information from Encoded Correlations

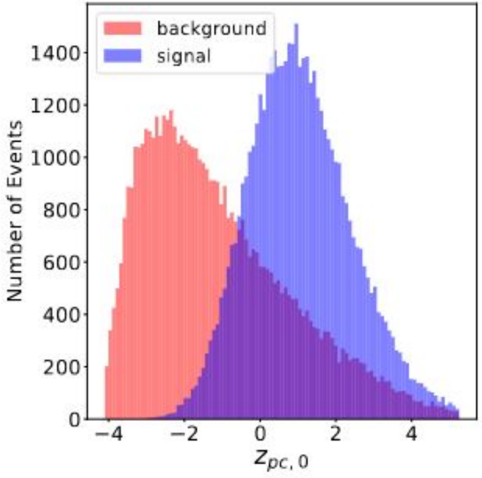
- Choose a latent subspace of highly ranked variables (ΔAUC < 1%)
- Perform **Principal Component Analysis** (PCA) on this latent subspace
- Select top principal components to account for up to 99% of the variance in latent data



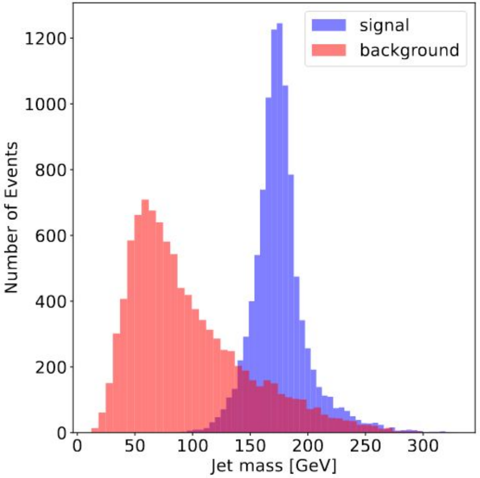
PFN Latent Space

$$z_k = \sum_i \Phi_k(p_i)$$

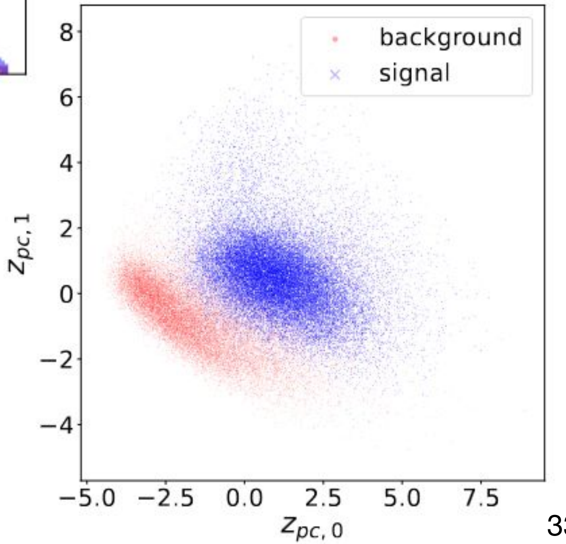
- Principal component learns to somewhat mimic jet mass distribution
- Also shows large correlations with some other physical variables



Jet class information is encoded in the correlation structure of the latent spaces

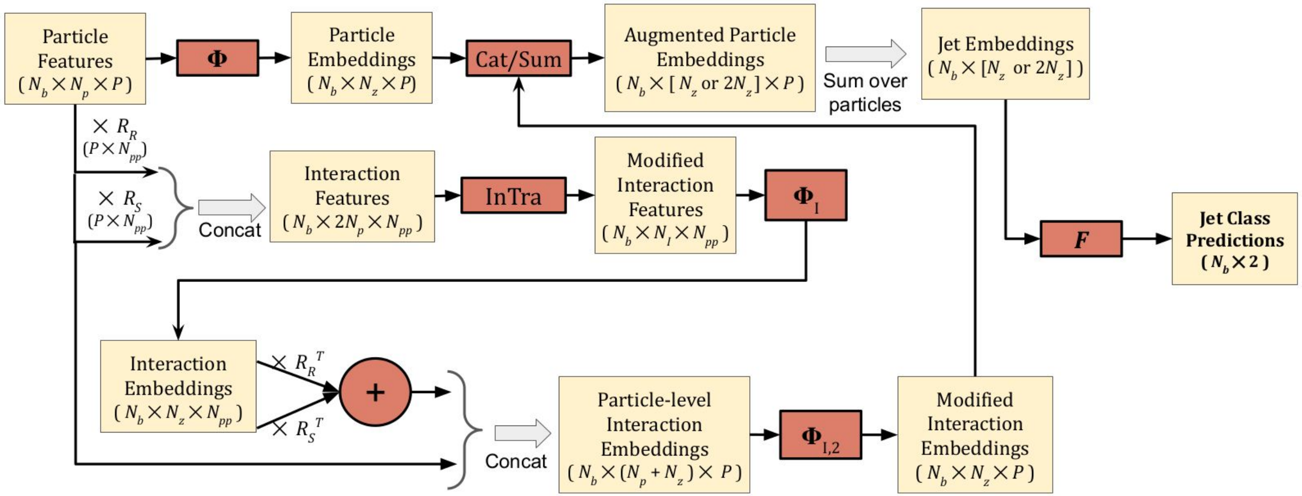


The network has learned to somewhat mimic jet mass distribution



XAI-Inspires a New Model: *Particle Flow Interaction Network*

- What did we learn from the XAI studies of TopoDNN and PFN?
  - PFN is limited by not considering inter-particle interactions is considered *room for improvement!*
  - Latent space for PFN is sparse – *scope for model simplification*



- Augment the PFN model with a Graph-net called **Interaction Network**
- This network models the *pairwise particle interaction* in the latent space

Particle Flow Interaction Network (PFIN)

Model Hyperparameters	
Number of constituents, N_p	60
Nodes in Φ Network	(100,100,64)
Nodes in Φ_I Network	(128,128,64)
Nodes in $\Phi_{I,2}$ Network	(128,128,64)
Nodes in F Network	(64,100,100)
Latent space dimension	64 (s), 128 (c)
Number of Parameters	97k (s), 101k (c)
Performance Metrics	
ROC-AUC	0.9839 (s), 0.9838 (c)
Accuracy	0.937 (s), 0.937 (c)
Background Rejection Rate ($1/\epsilon_B$)	1041 (s), 1030 (c)

Number of constituents reduced to 60 (from 200) since only the most energetic constituents show up as important features

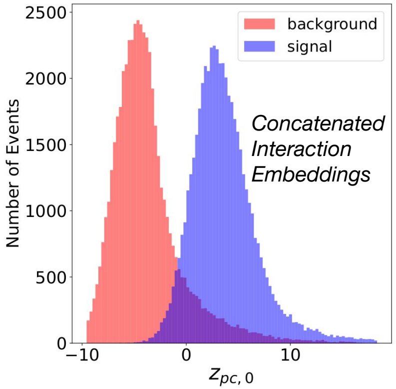
Network architectures are inspired by the NAP diagrams for the PFN model

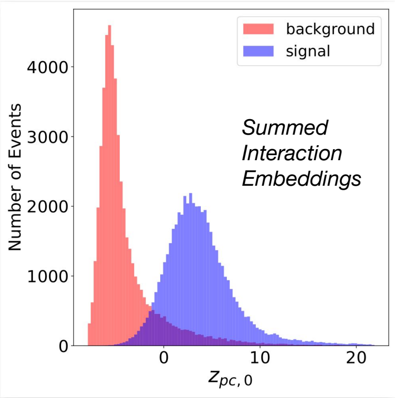
Outperforms both PFN and the IN models, comparable with ResNext and ParticleNet with a much smaller number of parameters and faster convergence

PFIN's Latent Space

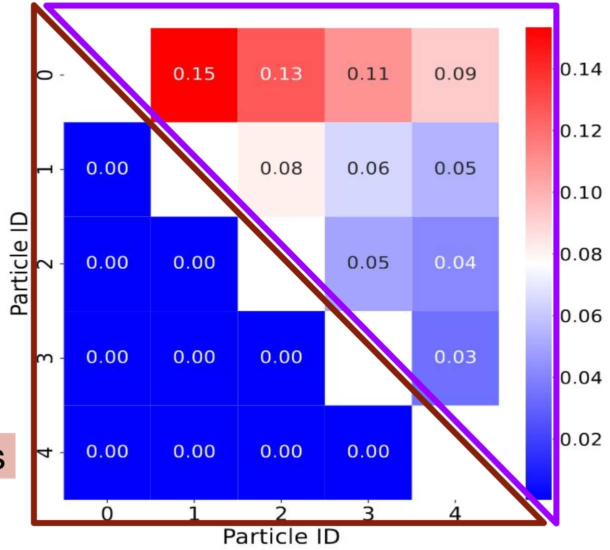
- PFIN latent space shows a much stronger correlation with the jet mass and the subjettiness variables

Latent space learns to mimic physical observables





Average impact on classification probability



- We can investigate the importance of pairwise particle interactions using MAD relevance of probability scores
- Inter-particle interactions play a significant role in top jet identification compared to QCD jets

Lessons Learned and Outlook

- Just like models themselves, one size does not fit all for model interpretation
- Model explanations can tricky and unreliable, especially when models
    - have highly correlated inputs
    - concurrently treat categorical and continuous features
    - have inputs that span over multiple orders
- RNA scores and NAP diagrams reveal important insight into model's desired complexity, can we use them for in-situ model optimization?
- Latent spaces are interesting- can they mimic physical features in more general settings (e.g. in multi-class classification)?
- Interpreting more complex models like graph nets, transformers etc. may require even better techniques
- ***Applying XAI methods can lead to better understanding and better networks!***