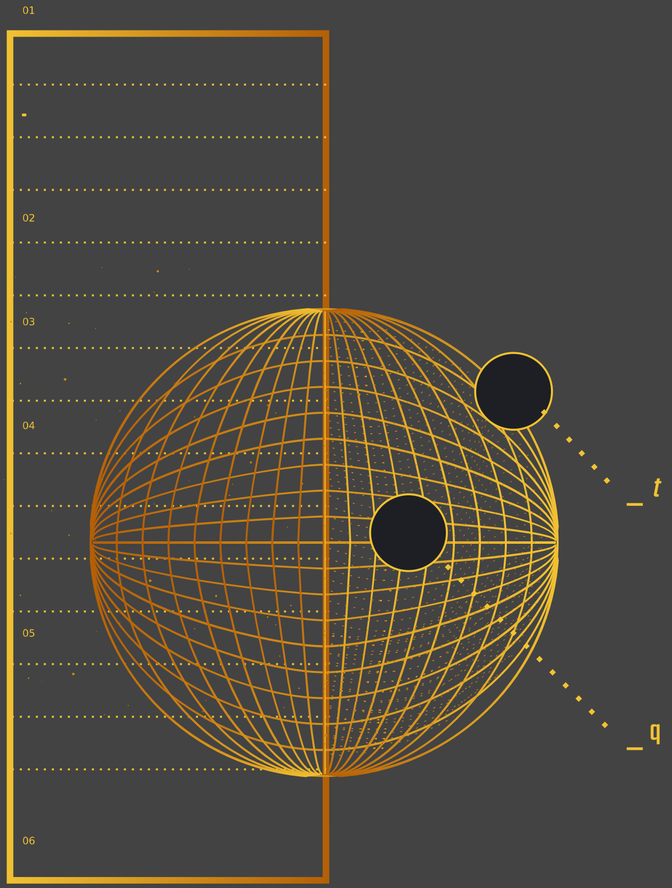


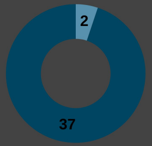
BIP

The lightweight invariant representation

Christoph Ortner
UBC University

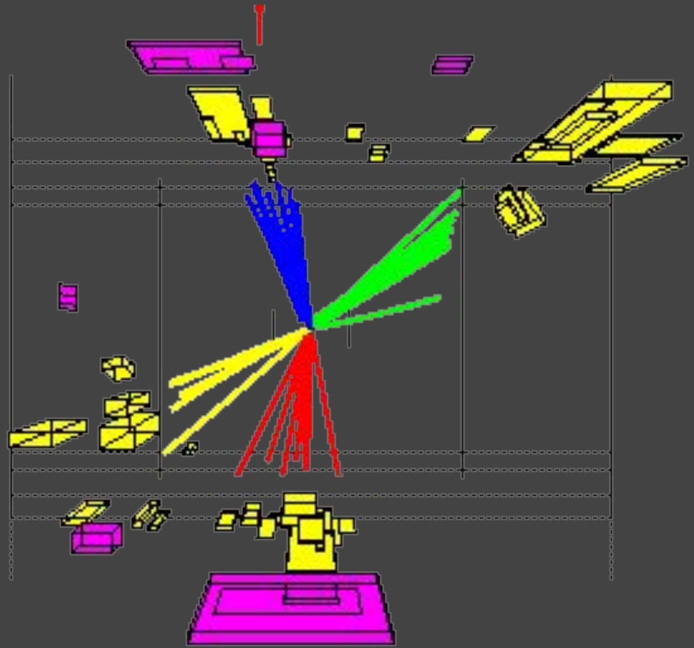
Ilyes Batatia
Cambridge University

Jose M Munoz
EIA University

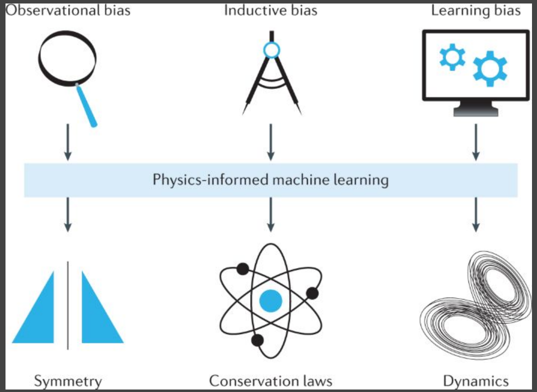


Straightforward…

But still hard



Karniadakis , et al

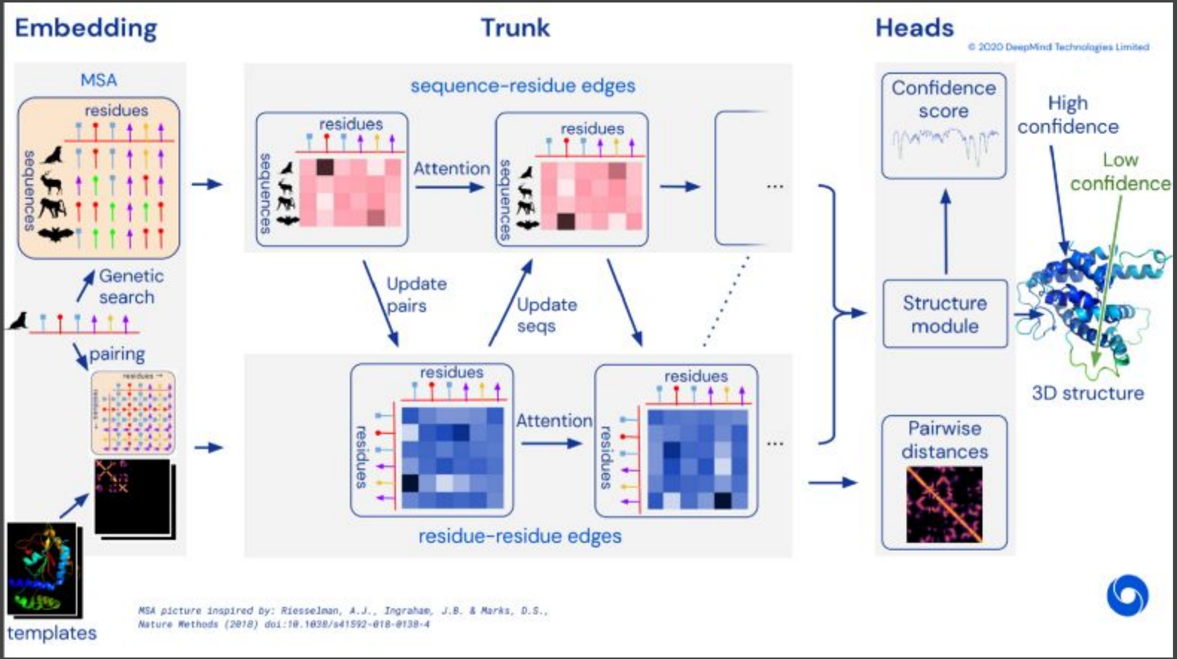


Why Invariance?

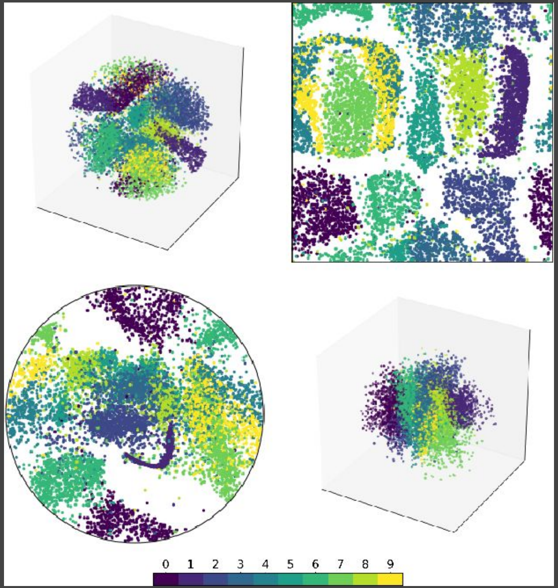
Why not just let the tagger/algorithm learn the invariance?

arXiv:2207.08272v2

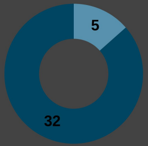
A couple of examples...



OpenAI, Alfa-Fold



Jiang, et al.



A couple of examples…

arXiv:2207.08272v2

PELICAN: Permutation Equivariant and Lorentz Invariant or Covariant Aggregator Network for Particle Physics

Alexander Bogatskiy
Center for Computational Mathematics
Flatiron Institute, New York, NY, U.S.A.
abogatskiy@flatironintitute.org

Timothy Hoffman
Department of Physics, University of Chicago
Chicago, IL, U.S.A.
hoffmant@uchicago.edu

David W. Miller
Department of Physics, University of Chicago
Enrico Fermi Institute
Chicago, IL, U.S.A.
David.W.Miller@uchicago.edu

Jan T. Offermann
Department of Physics, University of Chicago
Enrico Fermi Institute
Chicago, IL, U.S.A.
jano@uchicago.edu

Equivariant Energy Flow Networks for jet tagging

Matthew J. Dolan[1,*] and Ayodele Ore[1,†]

[1] *ARC Centre of Excellence for Dark Matter Particle Physics, School of Physics, The University of Melbourne, Victoria 3010, Australia*

An Efficient Lorentz Equivariant Graph Neural Network for Jet Tagging

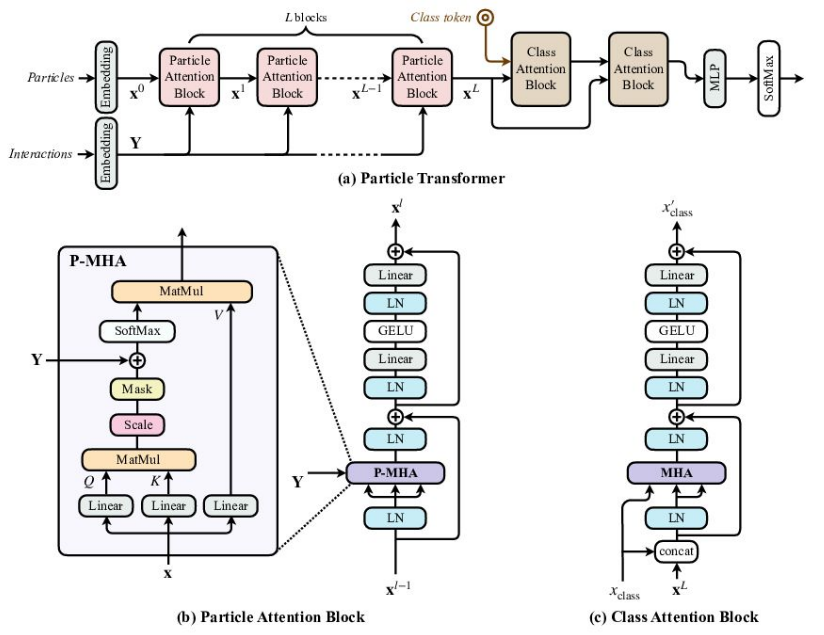
Shiqi Gong[a,e,1] Qi Meng[b] Jue Zhang[b] Huilin Qu[c] Congqiao Li[d] Sitian Qian[d] Weitao Du[a] Zhi-Ming Ma[a] Tie-Yan Liu[b]

Semi-Equivariant GNN Architectures for Jet Tagging

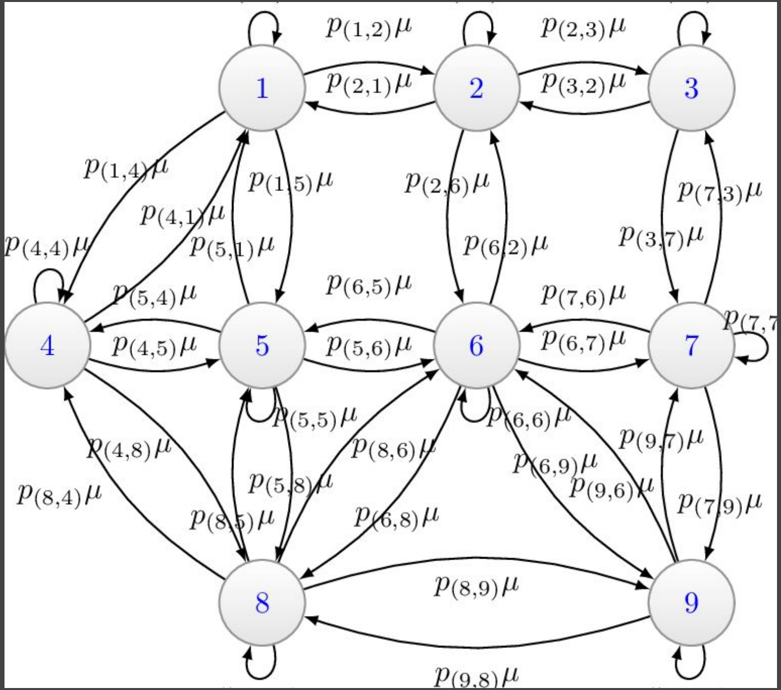
Daniel Murnane[1], Savannah Thais[2], Jason Wong[3]

Many more…

End even builds on...

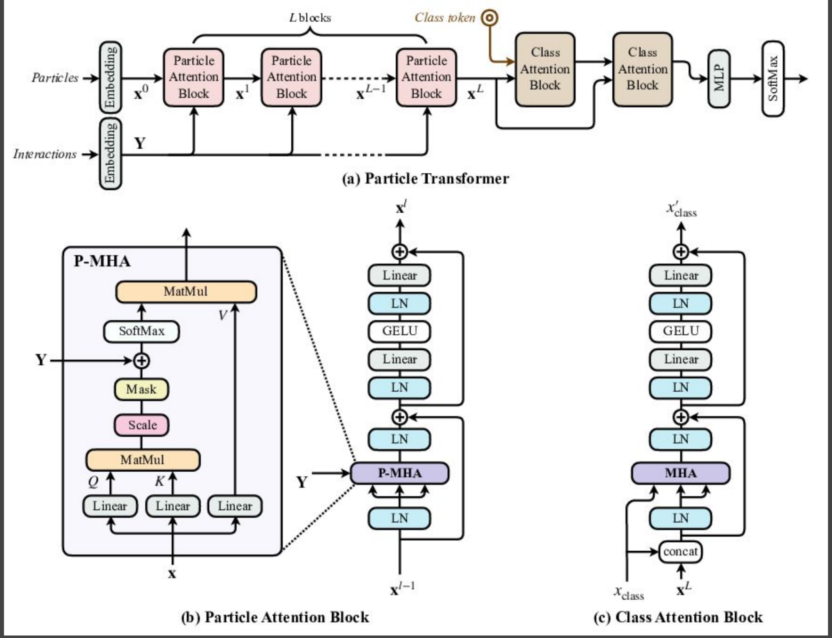


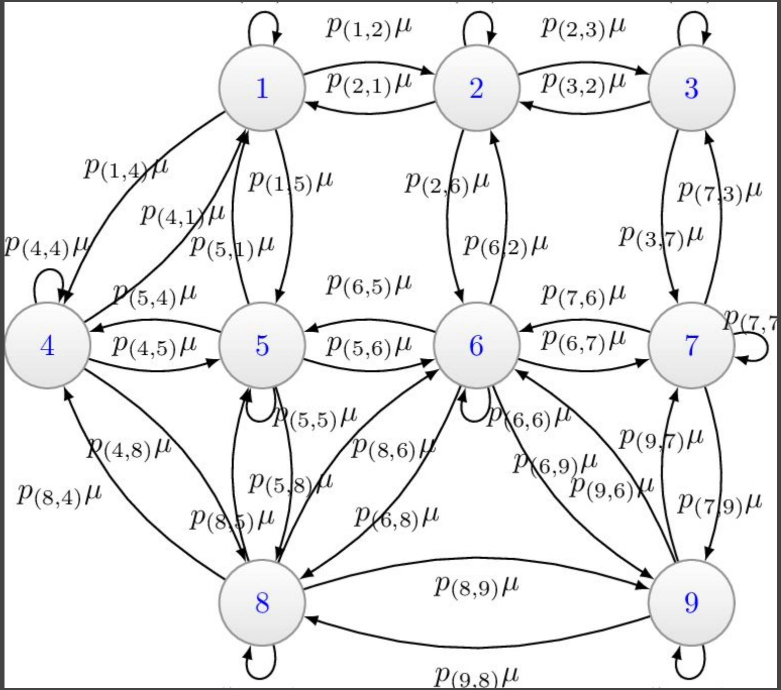
Qu, et al.



arXiv:2207.08272v2

End even builds on...

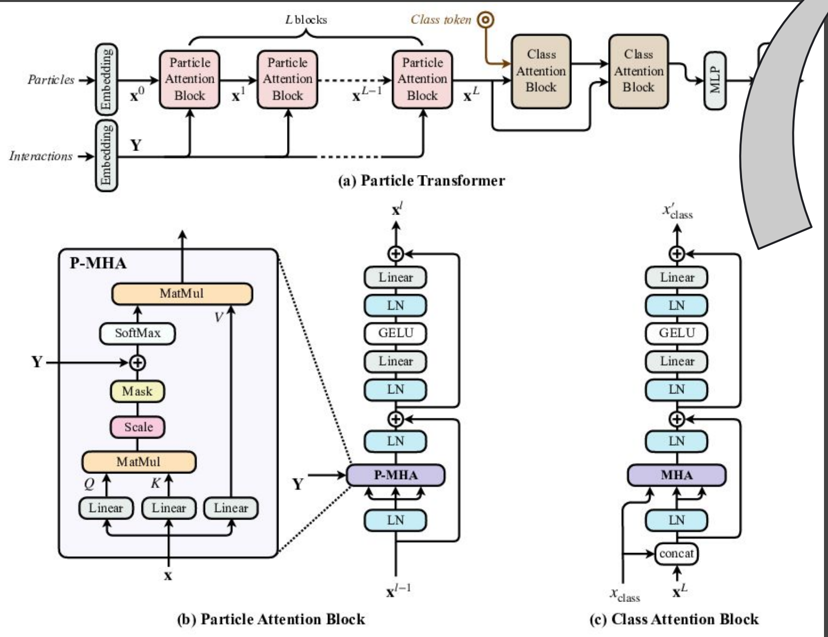




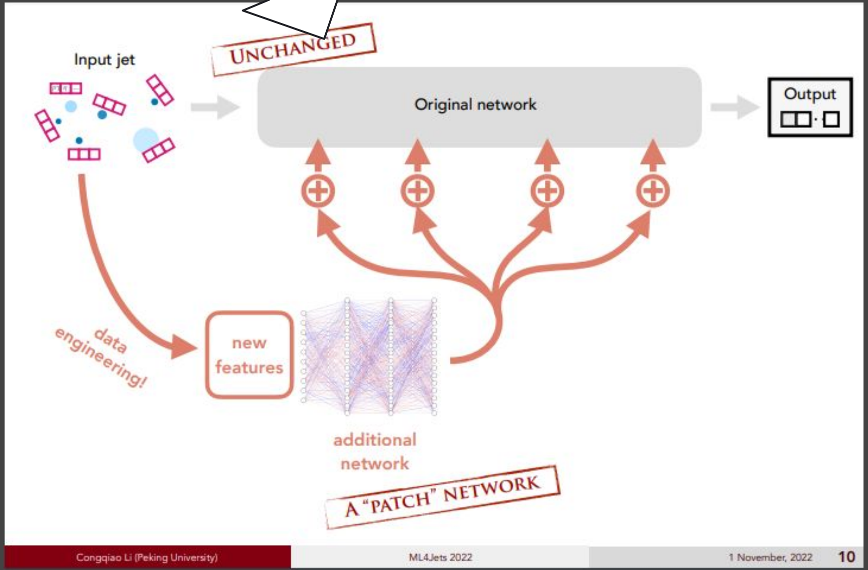
Qu, et al.

arXiv:2207.08272v2

End even builds on...



Qu, et al.



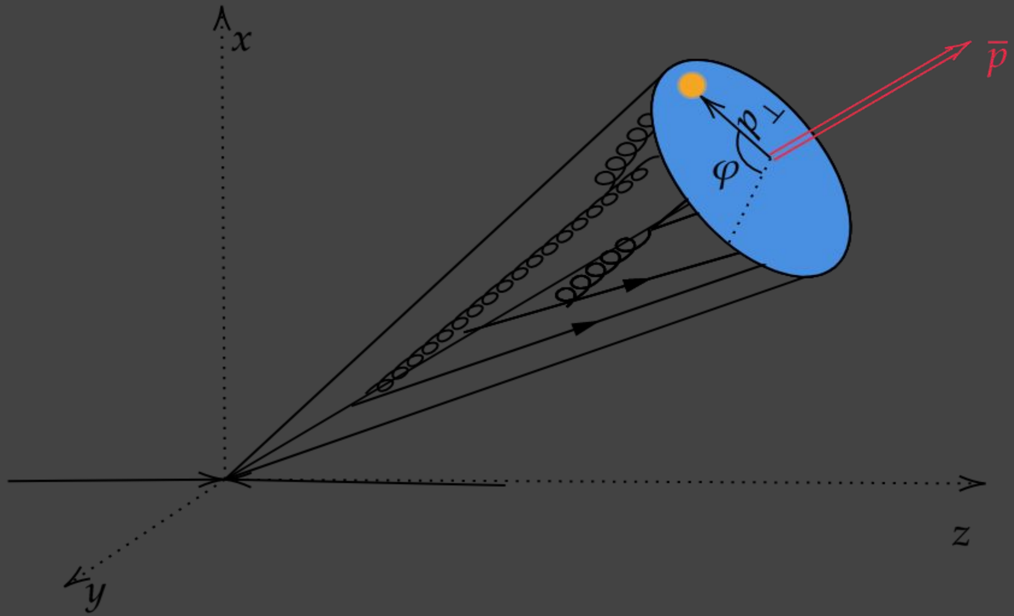
Congqiao Li

arXiv:2207.08272v2

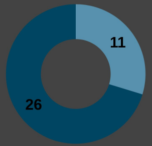
Building Invariant Polynomials

Transform, and then feed…

1. Perform a geometric transform



arXiv:2207.08272v2



Transform, and then feed...

$$A_{nlk} = \sum_{i=1}^{N} Q_n(p_{\perp,i}, E_{\perp,i}, \xi_i) e^{il\varphi_i} e^{-\lambda k y_i}$$

$$Q_n(E_{\perp,i}, p_{\perp,i}) = B_n(\tilde{p}_{\perp,i}) \log(1 + E_{\perp,i})$$

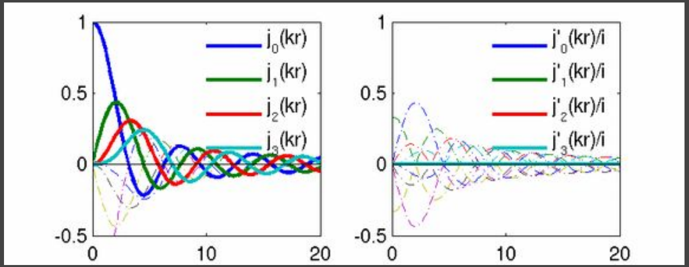
arXiv:2207.08272v2

Transform, and then feed...

2. Compute a one particle basis

$$A_{nlk} = \sum_{i=1}^{N} Q_n(p_{\perp,i}, E_{\perp,i}, \xi_i) e^{il\varphi_i} e^{-\lambda k y_i}$$

$$Q_n(E_{\perp,i}, p_{\perp,i}) = B_n(\tilde{p}_{\perp,i}) \log(1 + E_{\perp,i})$$

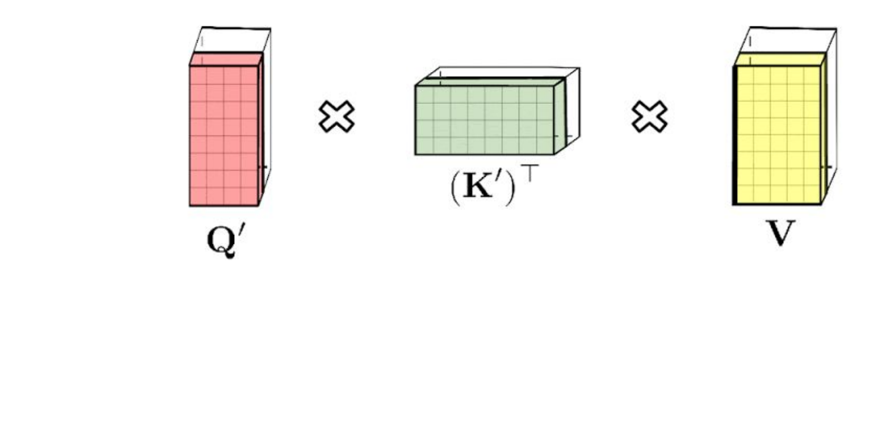


Transform, and then feed...

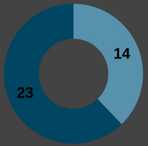
3. Symmetrize the basis

$$A_{\boldsymbol{nlk}} = \prod_{t=1}^{\nu} A_{n_t l_t k_t}$$

$$(n_1 l_1 k_1, \ \ldots, \ n_\nu l_\nu k_\nu)$$



arXiv:2207.08272v2

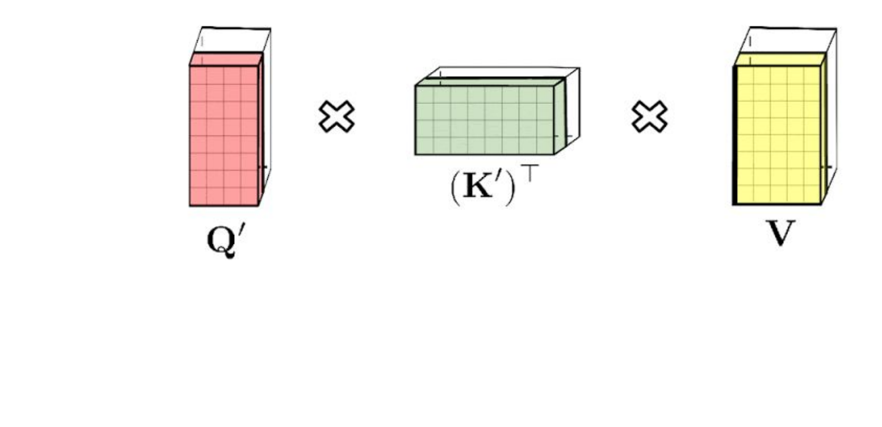


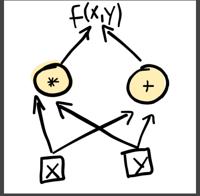
Transform, and then feed...

3. Symmetrize the basis

$$A_{\boldsymbol{nlk}} = \prod_{t=1}^{\nu} A_{n_t l_t k_t}$$

$$(n_1 l_1 k_1, \; ..., \; n_\nu l_\nu k_\nu)$$





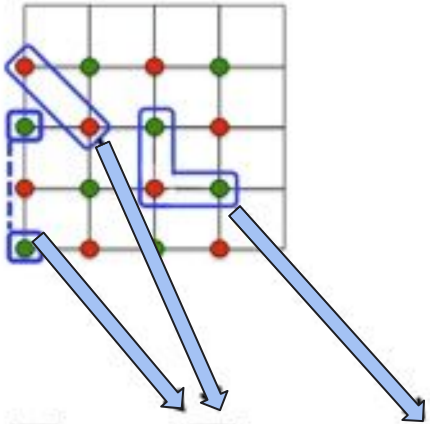
arXiv:2207.08272v2

Transform, and then feed...

3. Symmetrize the basis

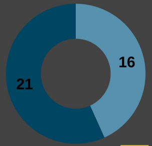
$$\boldsymbol{A_{nlk}} = \prod_{t=1}^{\nu} A_{n_t l_t k_t}$$

$$(n_1 l_1 k_1,\ \ldots,\ n_\nu l_\nu k_\nu)$$



$$\kappa(\sigma) = J_0 + \sum_i J_i\sigma_i + \sum_{i,j} J_{i,j}\sigma_i\sigma_j + \sum_{i,j,k} J_{i,j,k}\sigma_i\sigma_j\sigma_k + \cdots$$

Kadkhodaei, et al.

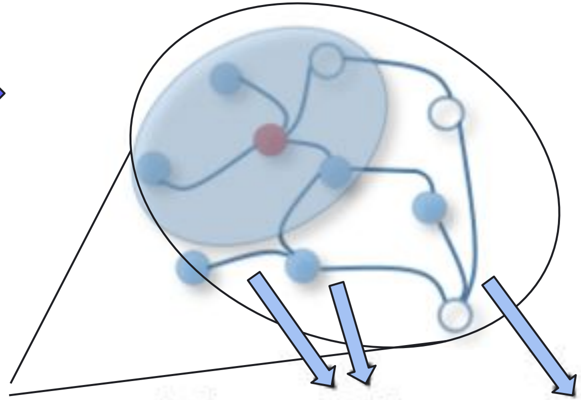


Transform, and then feed...

3. Symmetrize the basis

$$\boldsymbol{A_{nlk}} = \prod_{t=1}^{\nu} A_{n_t l_t k_t}$$

$$(n_1 l_1 k_1, \ \ldots, \ n_\nu l_\nu k_\nu)$$



$$\kappa(\sigma) = J_0 + \sum_i J_i \sigma_i + \sum_{i,j} J_{i,j}\sigma_i\sigma_j + \sum_{i,j,k} J_{i,j,k}\sigma_i\sigma_j\sigma_k + \cdots$$

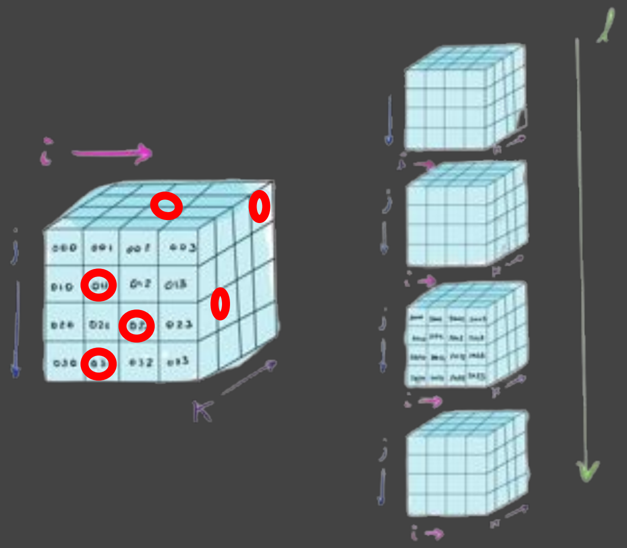
Transform, and then feed...

3. Go and select invariant indices

$$\boldsymbol{A_{nlk}} = \prod_{t=1}^{\nu} \sum_{i=1}^{N} Q_n(p_{\perp,i}, E_{\perp,i}, \xi_i) e^{\mathrm{i}l\varphi_i} e^{-\lambda k y_i}$$

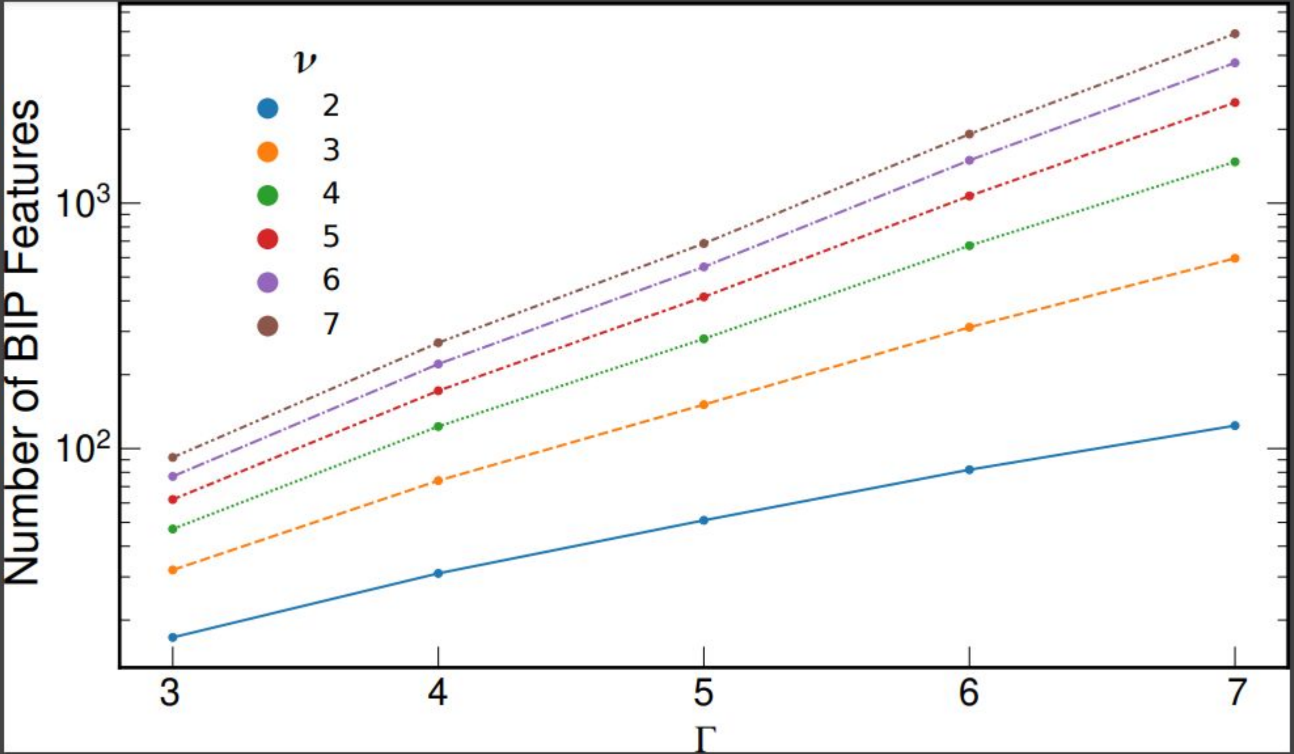
$$\sum_t l_t = \sum_t k_t = 0$$

$$\sum_{t=1}^{\nu} |l_t| + |k_t| + n_t \leq \Gamma$$



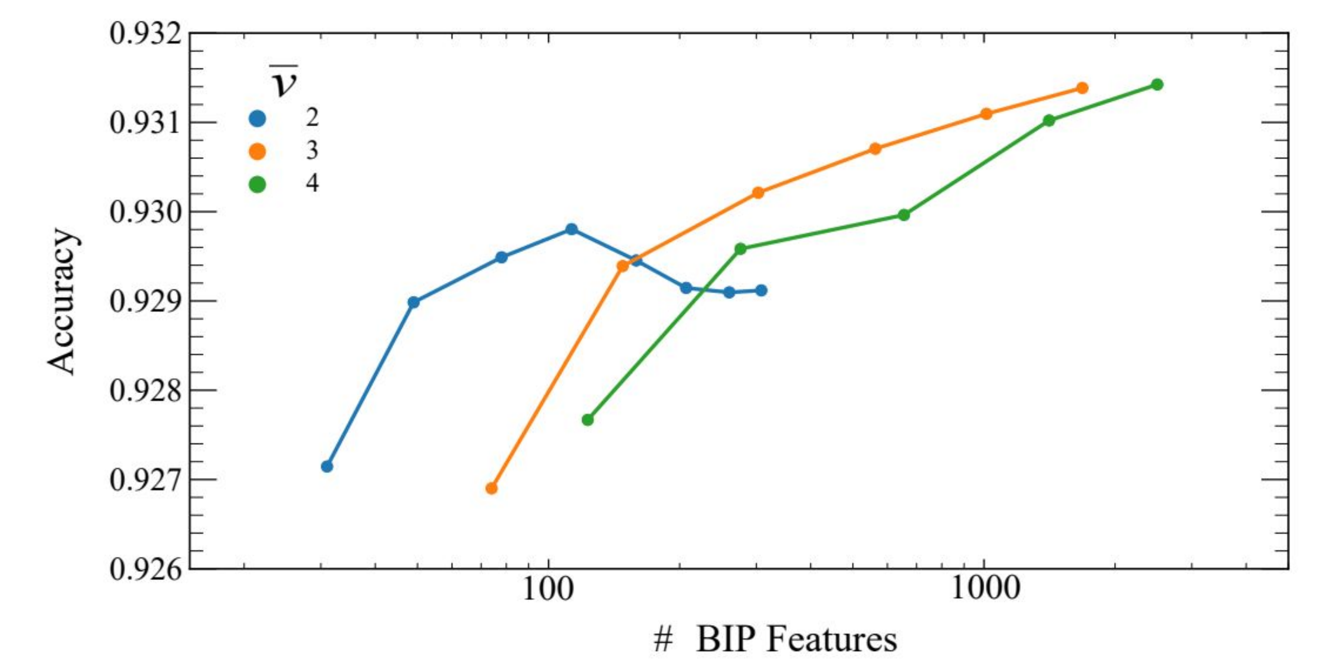
arXiv:2207.08272v2

Gives versatility… at some price

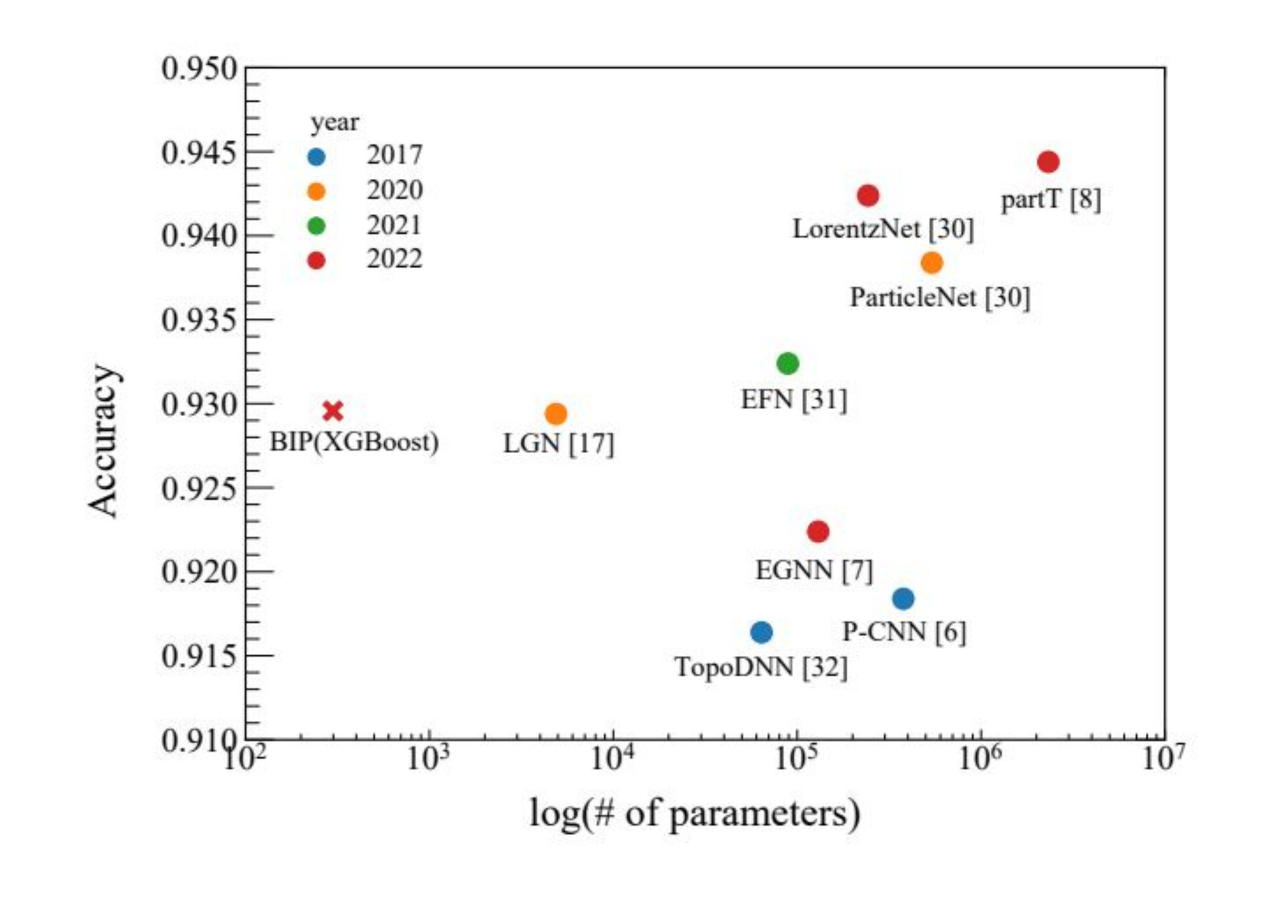


arXiv:2207.08272v2

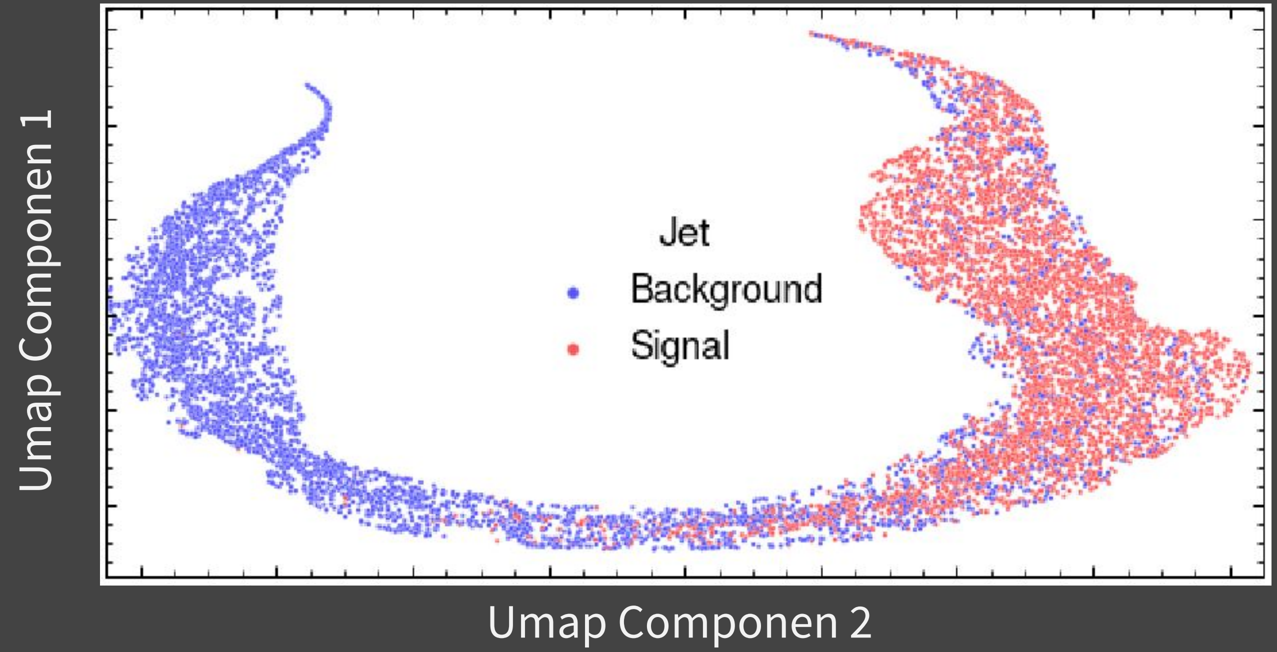
Gives versatility... at some price



arXiv:2207.08272v2

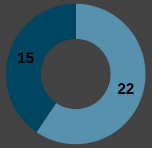


arXiv:2207.08272v2



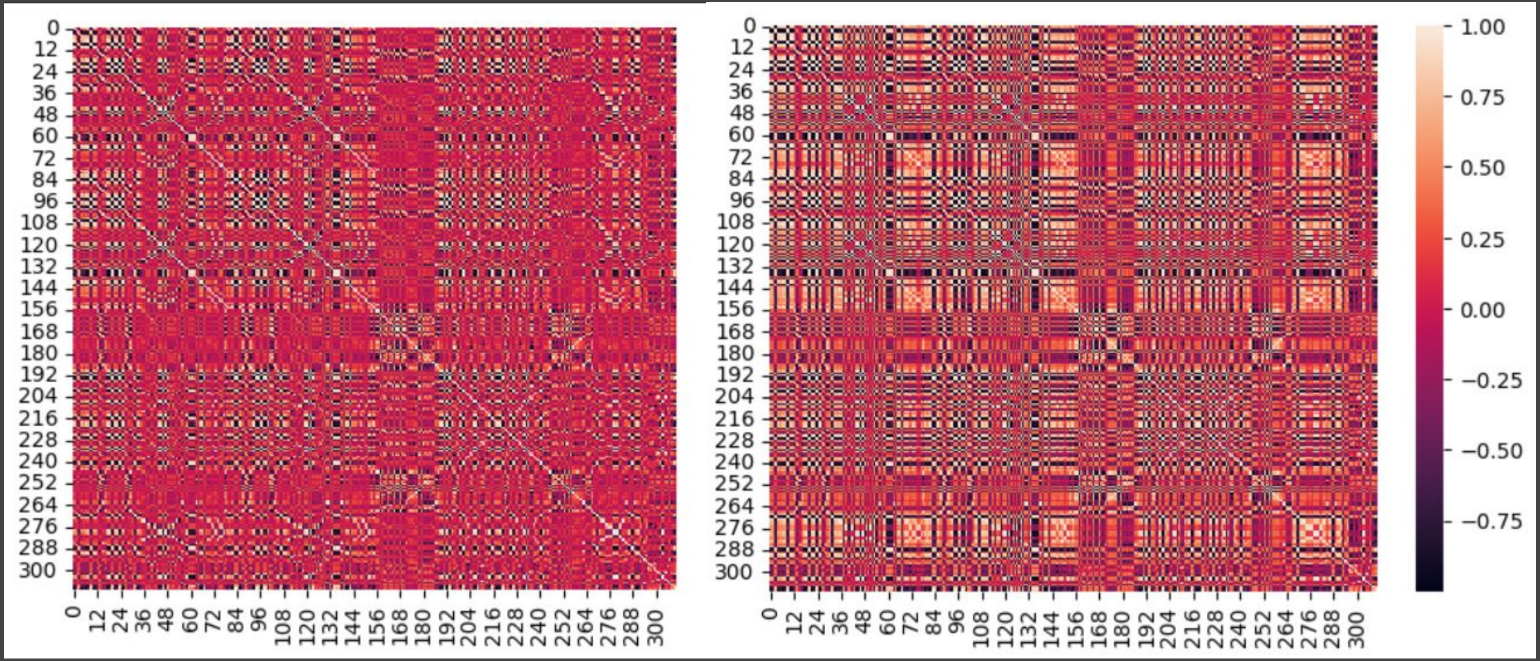
arXiv:2207.08272v2

Learning Lower in lower dimensions

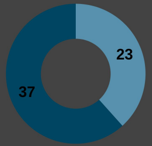


Why does it works?

Background | **Signal**

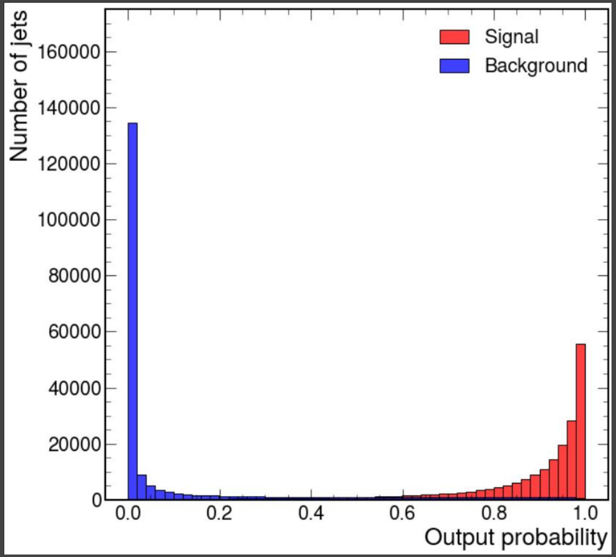


arXiv:2207.08272v2

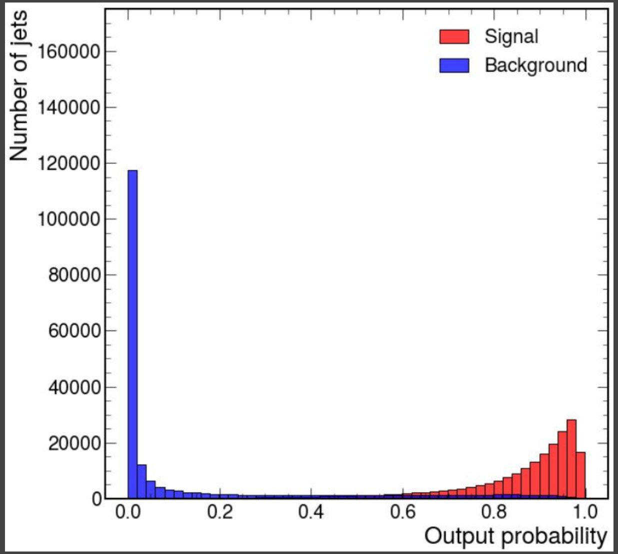


Is the frame telling something?

Jet axis



Beam Axis

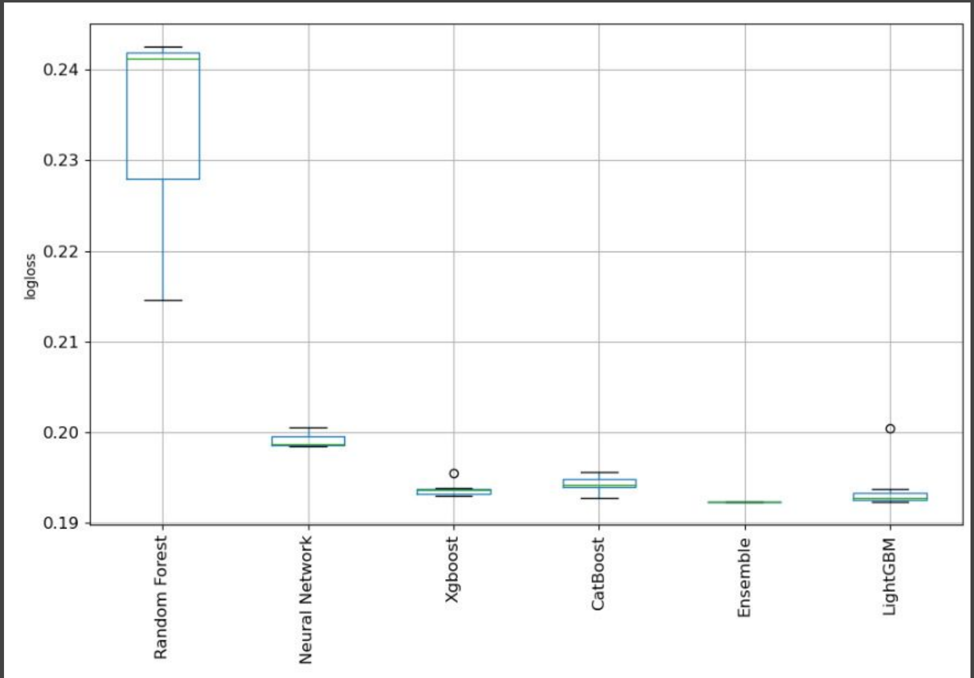


arXiv:2207.08272v2

Was

that interpretability?

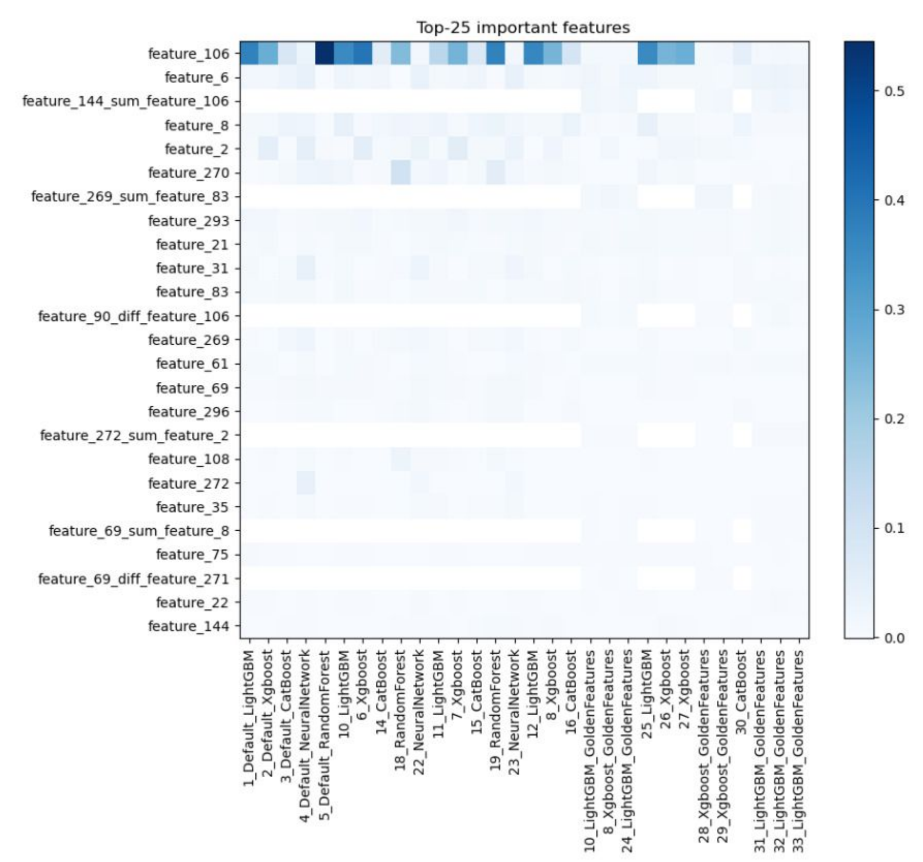
You can really fit a large amount of models...

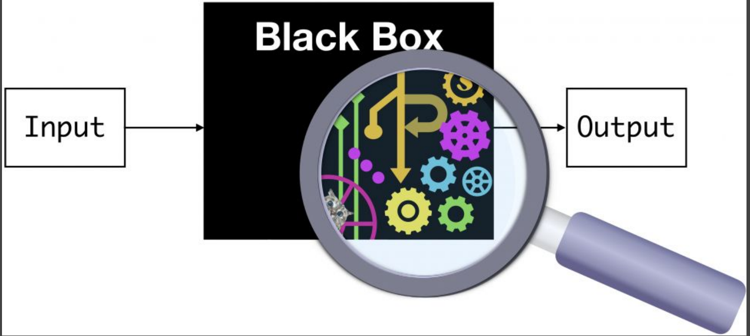


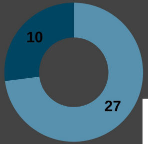
All of them trained using the same basis size.
312 Features in the BIP Basis

arXiv:2207.08272v2

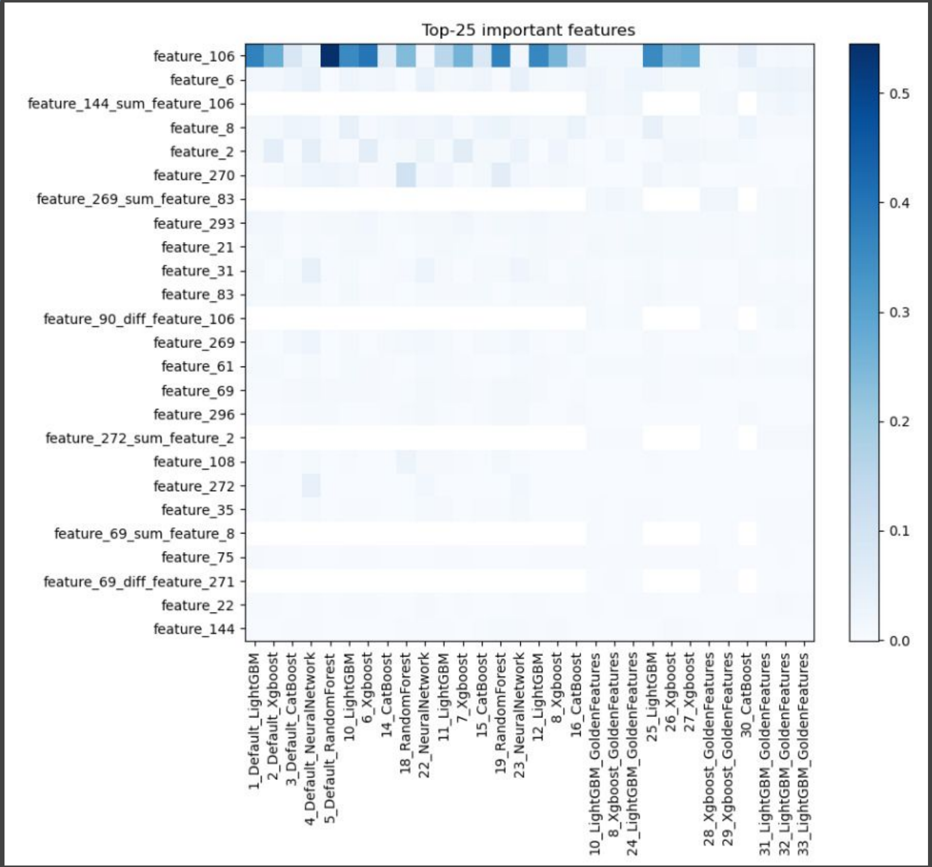
Permutation importance, or shapley values

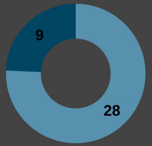






Permutation importance, or shapley values

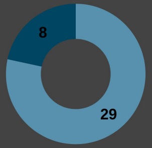




Background

Signal

arXiv:2207.08272v2



Extraction of symbolic relations

Most important feature

$$A_{\boldsymbol{nlk}} = \prod_{t=1}^{\nu} \sum_{i=1}^{N} Q_n(p_{\perp,i}, E_{\perp,i}, \xi_i) e^{il\varphi_i} e^{-\lambda k y_i}$$

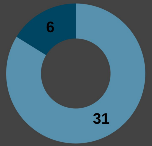
1 6 6

$$\boldsymbol{A_{166}} = A_{100} \cdot A_{100} \cdot A_{000}$$

Extraction of symbolic relations

Most important feature

$$A_{166} = \left[\sum_{i=1}^{N} B_1(\tilde{p}_{\perp,i}) \log(1+E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right] \cdot \left[\sum_{i=1}^{N} B_1(\tilde{p}_{\perp,i}) \log(1+E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right] \cdot \left[\sum_{i=1}^{N} B_0(\tilde{p}_{\perp,i}) \log(1+E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right]$$



Extraction of symbolic relations

Most important feature

Importance of edge information!

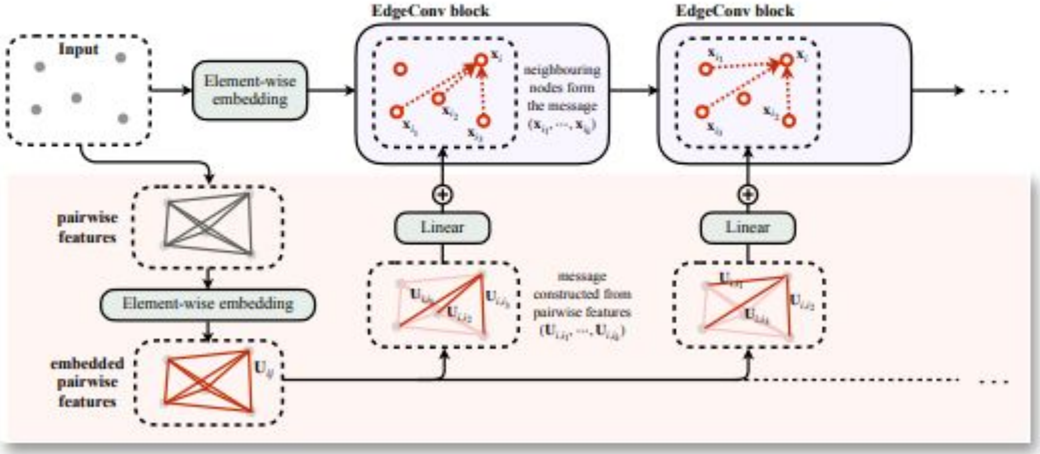
$$\boldsymbol{A}_{166} = \left[\sum_{i=1}^{N} B_1(\tilde{p}_{\perp,i}) \log(1 + E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right] \cdot \left[\sum_{i=1}^{N} B_1(\tilde{p}_{\perp,i}) \log(1 + E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right] \cdot \left[\sum_{i=1}^{N} B_0(\tilde{p}_{\perp,i}) \log(1 + E_{\perp,i}) e^{i0\varphi_i} e^{-\lambda 0 y_i}\right]$$

$$\boldsymbol{A}_{166} = \sum_{i=1}^{N} \log\left(\frac{p_{\perp,i}}{\sum_i p_{\perp,i}} + \delta\right) \cdot \log(1 + E_{\perp,i})$$

arXiv:2207.08272v2

Experiments on ParticleNet and LorentzNet

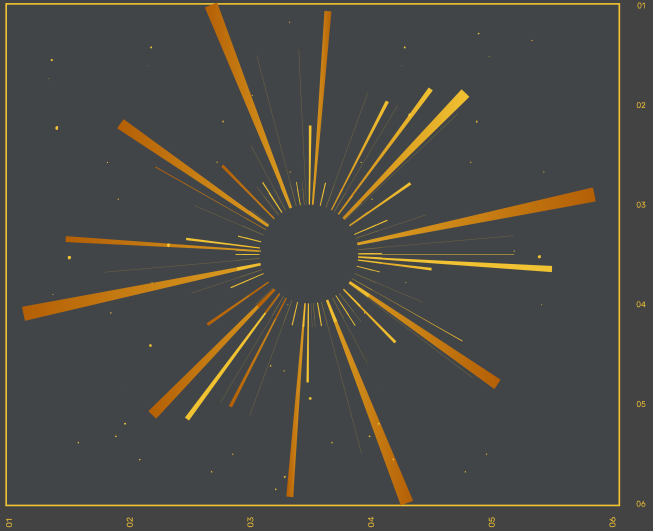
- Two baseline networks studied ***ParticleNet*** & ***LorentzNet$_{base}$***:
  - how to combine pairwise features + additional patch network to the baseline network?
- ***ParticleNet***: integrate pairwise features into the network according to the intrinsic k-NN pairs



Outlooks!

arXiv:2207.08272v2

Conclusions



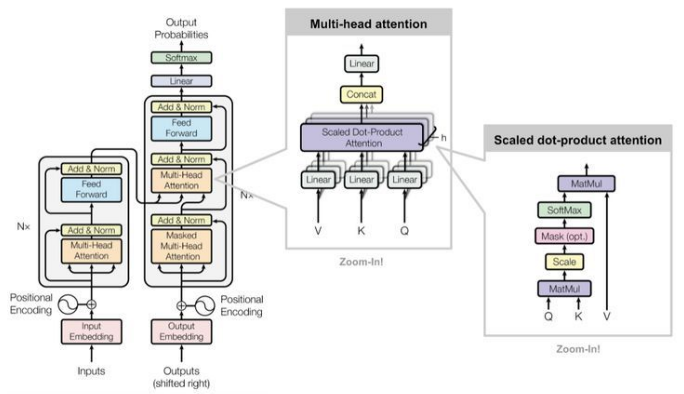
- Should we rethink **which invariances** to include?
- Feasibility of **highly efficiency** in jet descriptions.
- Extraction of **symbolic expressions** out of the features.
- **Classifier independent** invariant representations

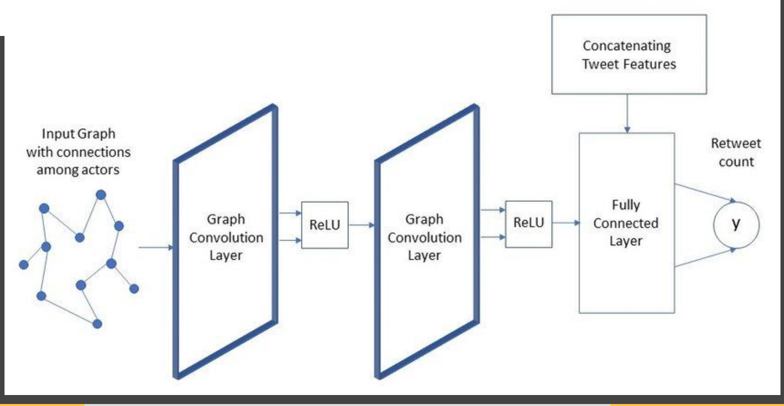
Outlooks

- Explore addition of ***high level variables***.
- Explore **automatic subgroup invariant selection**.
- Automatic **symbolic extraction**.
- **Different processes**?
- **Extrapolation capabilities** of the multiple classifiers.

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

How deep could one go?





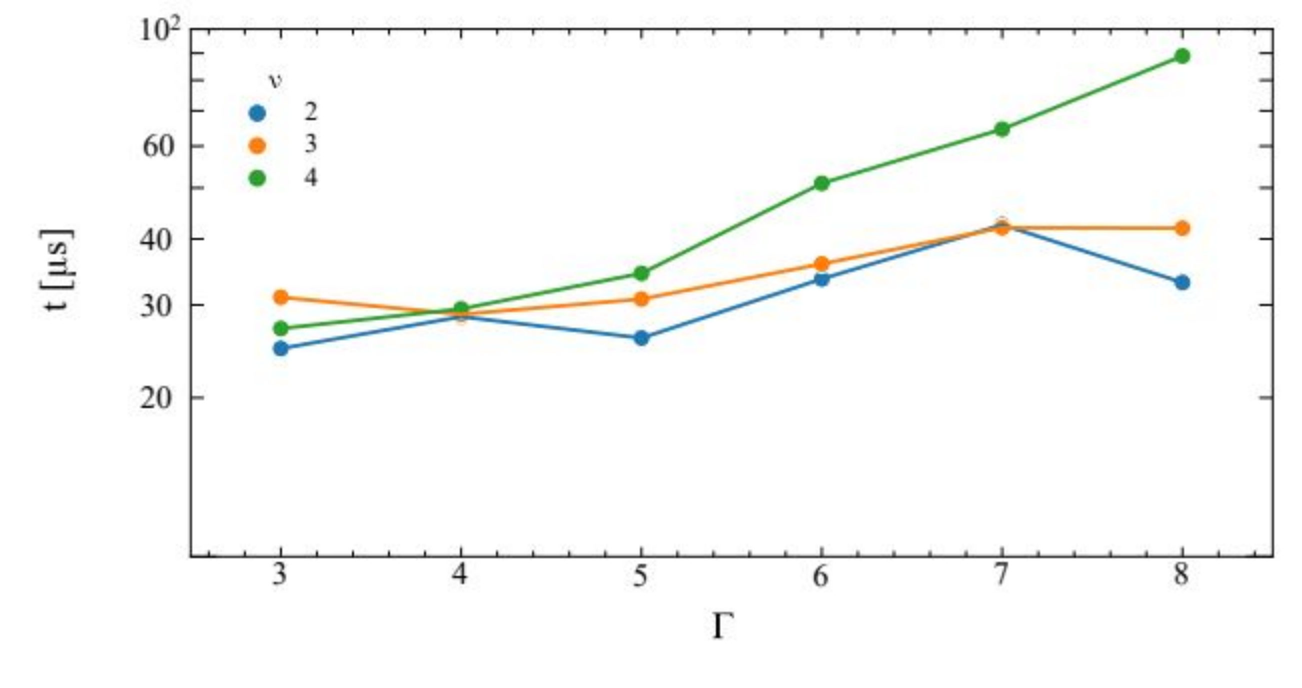
arXiv:2207.08272v2

THANKS!

Ask me Anything

003-1040559 1250 003-77156.8 1760 0009-14563.7 73273

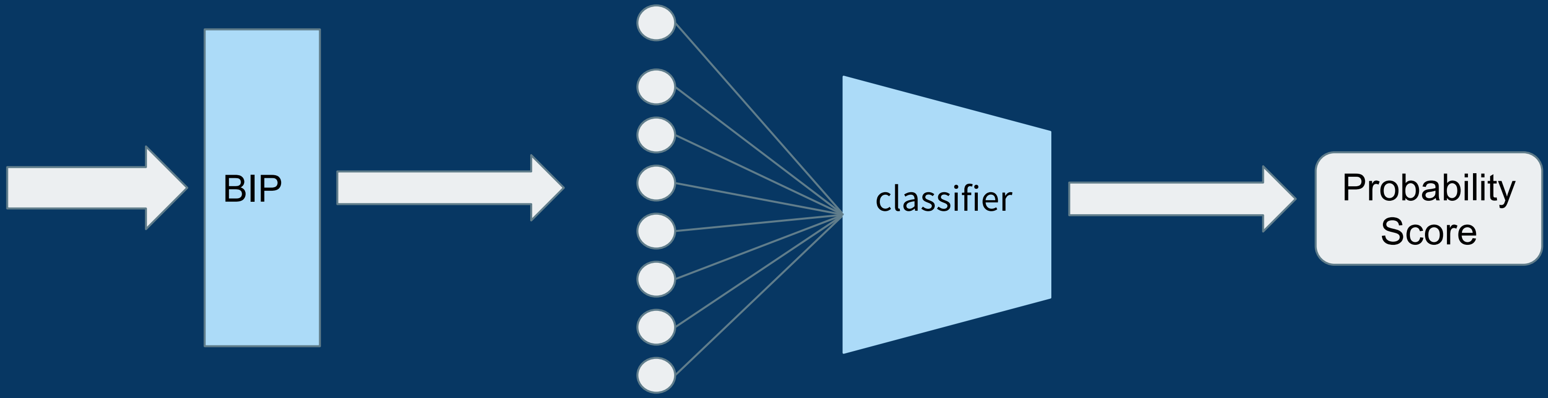
Backup



Play with the model

Paper

$$f\left(\{E_i, \boldsymbol{p}_i, \xi_i\}_i\right) = \sum_{\boldsymbol{nlk}} w_{\boldsymbol{nlk}} \boldsymbol{A}_{\boldsymbol{nlk}},$$



A many body expansion

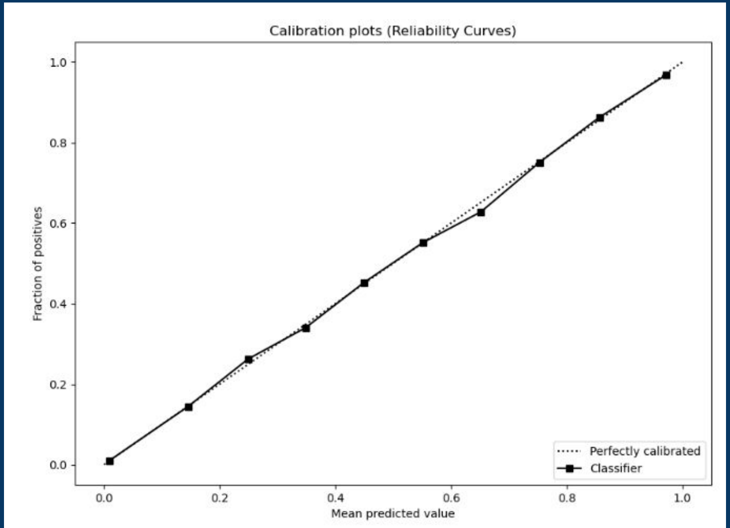
$$f(\{x_i\}_i) = f_0 + \sum_i f_1(x_i) + \sum_{i_1,i_2} f_2(x_{i_1}, x_{i_2}) + \cdots + \sum_{i_1,\ldots,i_{\bar{\nu}}} f_{\bar{\nu}}(x_{i_1}, \ldots, x_{i_{\bar{\nu}}}).$$

Or a density

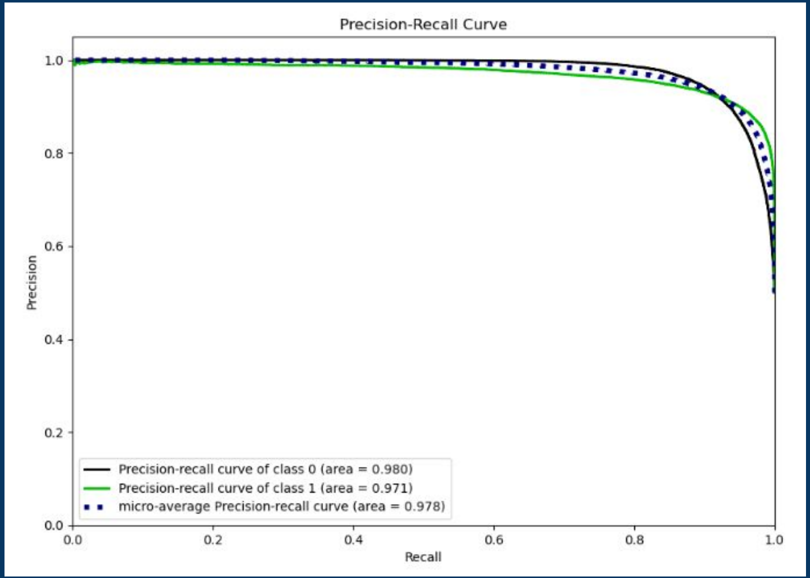
$$\rho(x) = \sum_{i=1}^{N} \delta(x - x_i)$$

$$A_v = \langle Q_n(p_\perp, E_\perp, \xi) e^{il\varphi} e^{-ky} \,|\, \rho\rangle$$

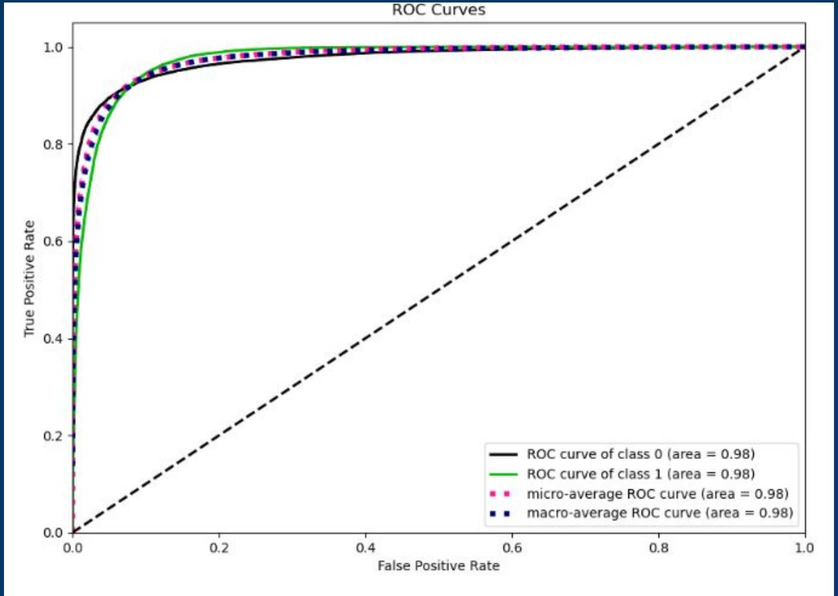
$$\langle \phi_{v_1} \otimes \cdots \otimes \phi_{v_\nu} \,|\, \rho \otimes \cdots \otimes \rho \rangle = \prod_t \langle \phi_{v_t} \,|\, \rho\rangle = \prod_t A_{v_t} = \boldsymbol{A}_{\boldsymbol{v}},$$

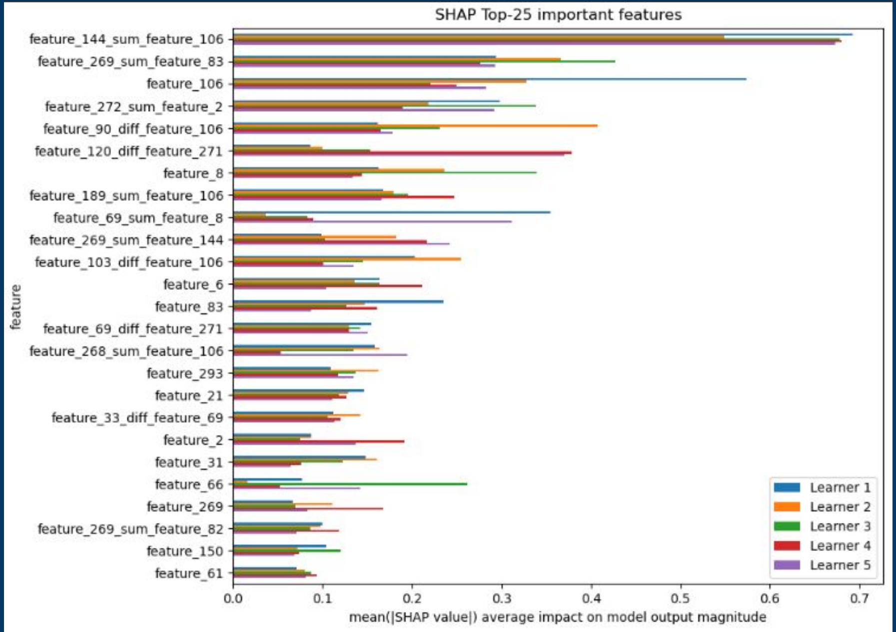


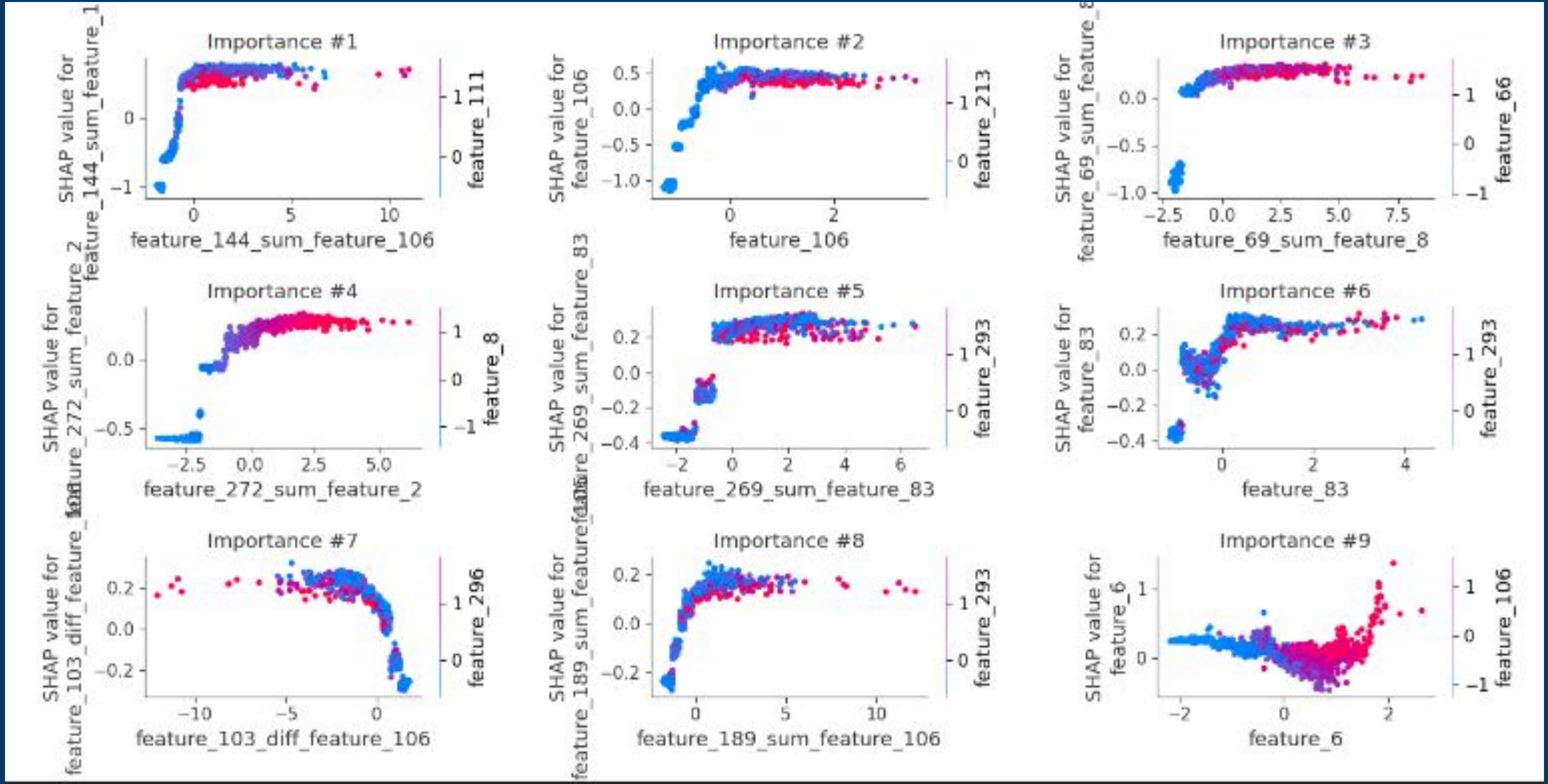
Calibration plots (Reliability Curves)
Fraction of positives
Mean predicted value
0.0
0.2
0.4
0.6
0.8
1.0
Perfectly calibrated
Classifier



Precision-Recall Curve
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0
Precision-recall curve of class 0 (area = 0.980)
Precision-recall curve of class 1 (area = 0.971)
micro-average Precision-recall curve (area = 0.978)


ROC Curves
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ROC curve of class 0 (area = 0.98)
ROC curve of class 1 (area = 0.98)
micro-average ROC curve (area = 0.98)
macro-average ROC curve (area = 0.98)


SHAP Top-25 important features
feature
feature_144_sum_feature_106
feature_269_sum_feature_83
feature_106
feature_272_sum_feature_2
feature_90_diff_feature_106
feature_120_diff_feature_271
feature_8
feature_189_sum_feature_106
feature_69_sum_feature_8
feature_269_sum_feature_144
feature_103_diff_feature_106
feature_6
feature_83
feature_69_diff_feature_271
feature_268_sum_feature_106
feature_293
feature_21
feature_33_diff_feature_69
feature_2
feature_31
feature_66
feature_269
feature_269_sum_feature_82
feature_150
feature_61
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
mean(|SHAP value|) average impact on model output magnitude
Learner 1
Learner 2
Learner 3
Learner 4
Learner 5


Importance #1
SHAP value for feature_144_sum_feature_106
feature_144_sum_feature_106
feature_111
Importance #2
SHAP value for feature_106
feature_106
feature_213
Importance #3
SHAP value for feature_69_sum_feature_8
feature_69_sum_feature_8
feature_66
Importance #4
SHAP value for feature_272_sum_feature_2
feature_272_sum_feature_2
feature_8
Importance #5
SHAP value for feature_269_sum_feature_83
feature_269_sum_feature_83
feature_293
Importance #6
SHAP value for feature_83
feature_83
feature_293
Importance #7
SHAP value for feature_103_diff_feature_106
feature_103_diff_feature_106
feature_296
Importance #8
SHAP value for feature_189_sum_feature_106
feature_189_sum_feature_106
feature_293
Importance #9
SHAP value for feature_6
feature_6
feature_106