Resonant Anomaly Detection in the Presence of Irrelevant Features

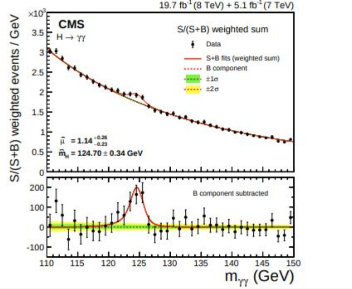
Yik Chuen San, Cornell University

Cornell University

ML4Jets 2023

Based on 2310.13057, with Marat Freytsis & Maxim Perelstein

Fancy Bump Hunting



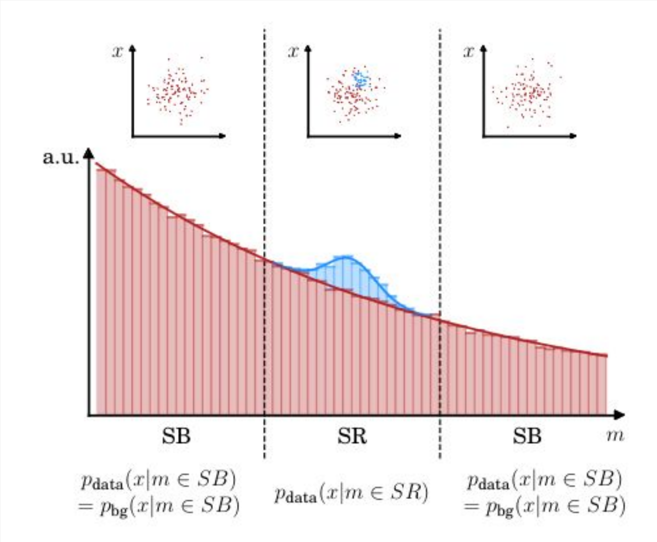
- So much more information than just invariant mass…
- Bump hunting in higher dimensional space using ML:

This talk

  - **CWoLa hunting** [Collins, Howe, Nachman 1902.02634]
  - **ANODE** [Nachman, Shih 2001.04990]
  - CATHODE [Hallin et. al. 2109.00546]
  - ……
- Assumption: $p(\vec{x}, m|\text{sig}) = 0 \text{ for } m \notin SR$

(+ Extra assumptions)

- Data-driven!



[Figure taken from 2109.00546]

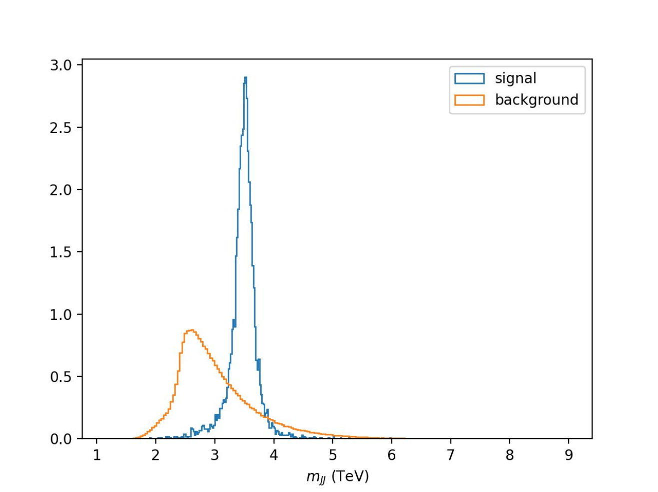
A Problem with Fancy Bump Hunting

- They all work very well, until they don't
  - Literature: Hand-crafted useful features (i.e. observables)
    - Assumes some extended knowledge about signal models
  - Realistically: include *irrelevant* features
    - Quite drastic degradation (see later slides)

Dataset Used* - LHCO R&D

- Background: QCD dijets
  Signal: W’ -> X (-> qq) Y (-> qq)
- Useful features
  - m_{J1} : mass of lighter jets
  - Δm_J : absolute difference of masses of two jets
  - τ_{21}^{J1}
  - τ_{21}^{J2}

  $\left.\begin{matrix}\tau_{21}^{J1} \\ \tau_{21}^{J2}\end{matrix}\right\}$ N-subjettiness ratios for two jets
- Irrelevant features
  - Independently drawn gaussian variables



*Accessed here

CWoLa Hunting [Collins, Howe, Nachman 1902.02634]

- *If* extra features are independent of m in the background, *then* an optimal* test statistic is

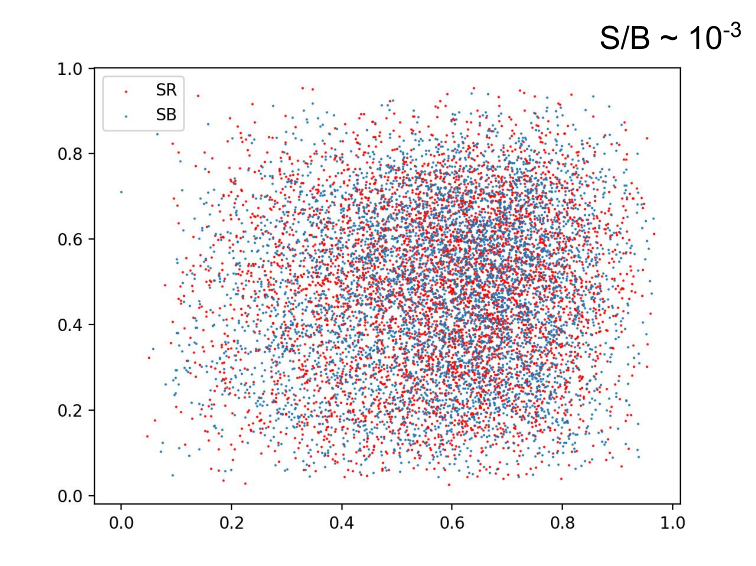
$$R_{\text{cwola}} = \frac{p(\vec{x}|m \in \text{SR})}{p(\vec{x}|m \in \text{SB})}$$

- Just train a classifier on SR vs SB, easy! 🙂

*In the Neyman-Pearson sense

One challenge: Overfitting

- What the classifier sees (in 2D)



- Imagine this in higher dimension with many irrelevant directions!

Preventing Overfitting on Irrelevant Features

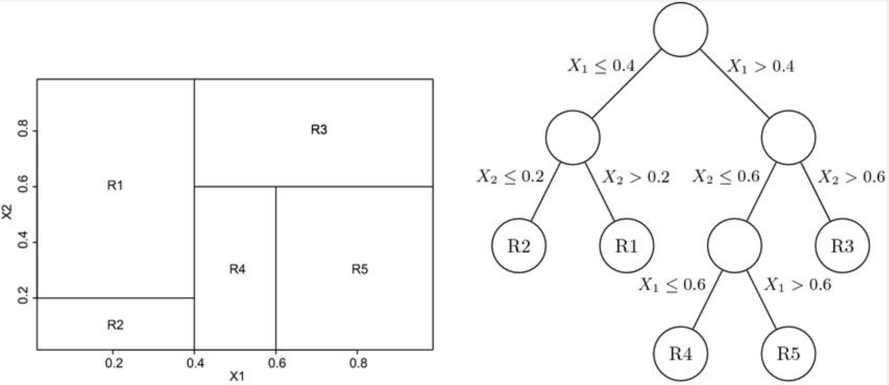
- Choice of algorithm
  - Trees: internal feature selection
- Limit model complexity, add regularization
  - cross-validation -> can be expensive!

-> xgboost: tree-based, fast, performan

Other choices also exist

TABLE 10.1. *Some characteristics of different learning methods. Key:* ▲ *= good,* ◆ *=fair, and* ▼ *=poor.*

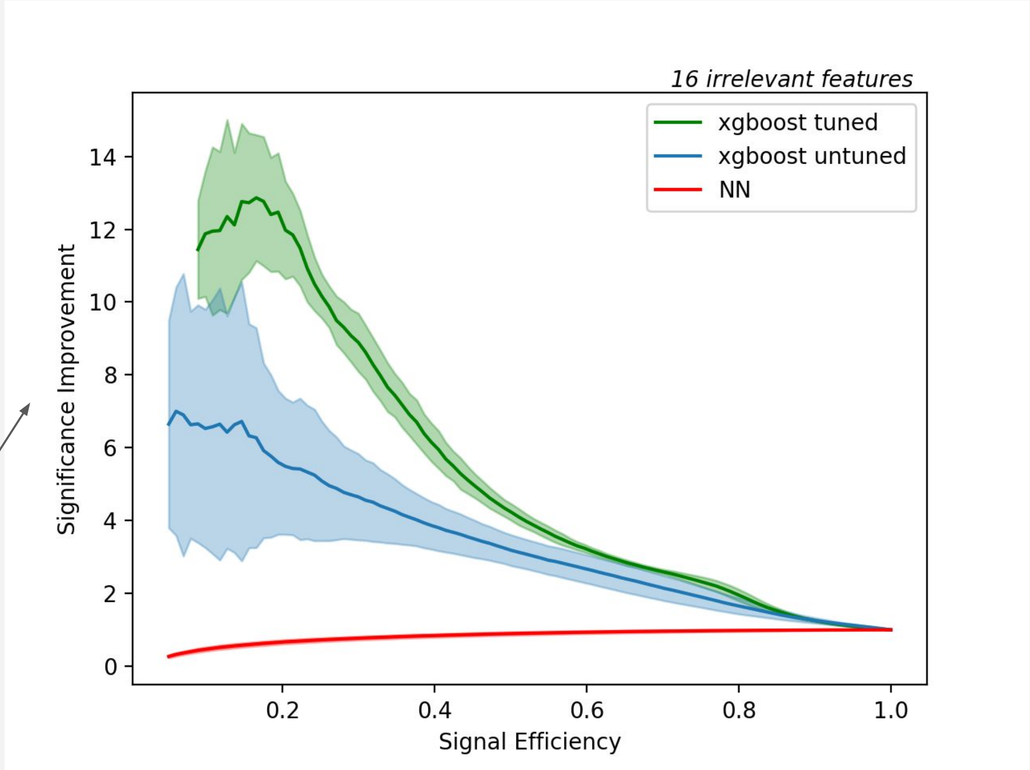
Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲



Taken from *The Elements of Statistical Learning* by Hastie, Tibshirani and Friedman

CWoLa Hunting with NNs vs Trees

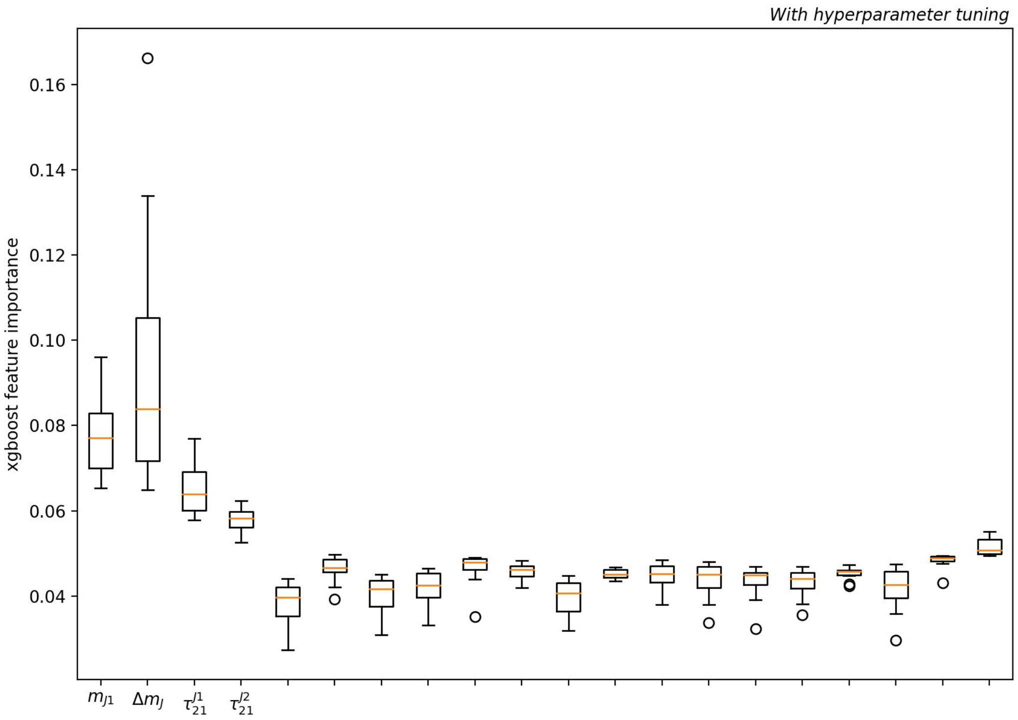
$\epsilon_S/\sqrt{\epsilon_B}$



CWoLa prefers trees… 🤔

NN Generated by code at
https://github.com/HEPML-AnomalyDetection/CATHODE

Bonus: Feature Importance



Another Issue with CWoLa Hunting

If extra features are independent of m in the background,

then an optimal* test statistic is

$$R_{\text{cwola}} = \frac{p(\vec{x}|m \in \text{SR})}{p(\vec{x}|m \in \text{SB})}$$

Density-estimation Based Method

- Another optimal statistic:

$$R = \frac{p(\vec{x}|m)}{p(\vec{x}|m, \mathrm{bkgd})}$$

No extra assumption needed

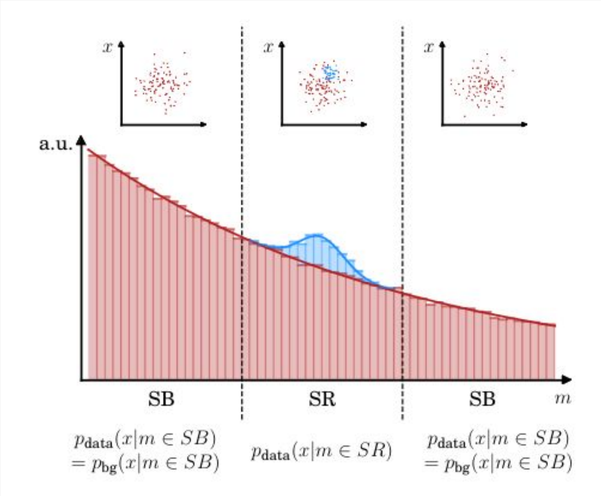
This is problem

ANODE [Nachman, Shih 2001.04990]

$$R = \frac{p(\vec{x}|m)}{p(\vec{x}|m, \mathrm{bkgd})}$$

Estimate from data, “easy”

Estimate by interpolation, recall
$p(\vec{x}, m|\mathrm{sig}) = 0$ for $m \notin SR$



Interpolation: Manual vs Auto

- “The interpolation is done automatically by the neural conditional density estimator” [2001.04990]
    - Black-box: a blessing and a curse
- Manual interpolation, a simple baseline:

$$p(\vec{x}|m, \mathrm{bkgd}) \approx p(\vec{x}|m_L) + \frac{p(\vec{x}|m_R) - p(\vec{x}|m_L)}{m_R - m_L}(m - m_L)$$

    - Simple, quick to evaluate
    - Linear in estimated densities

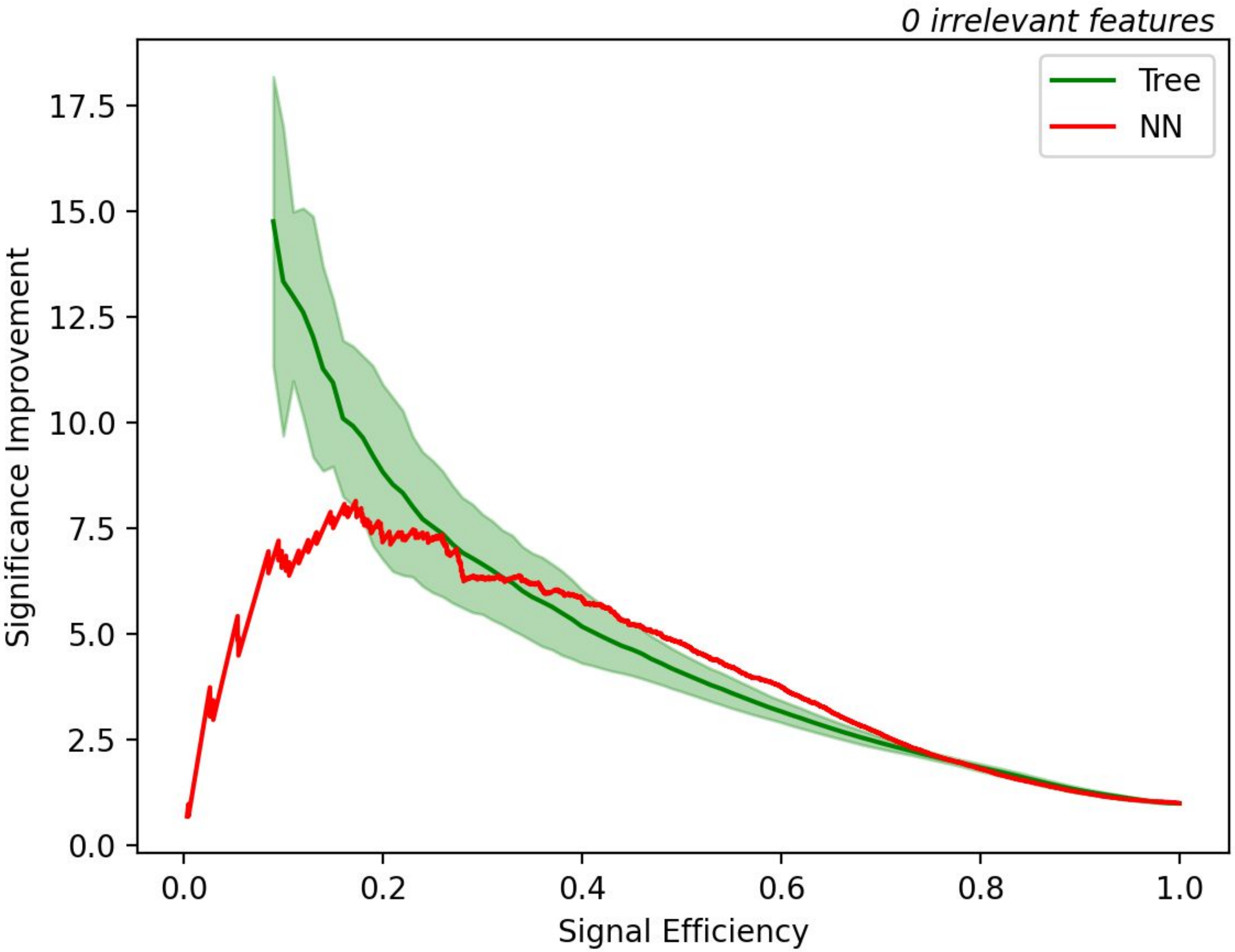
$$R = \frac{p(\vec{x}|m)}{p(\vec{x}|m, \mathrm{bkgd})}$$

"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

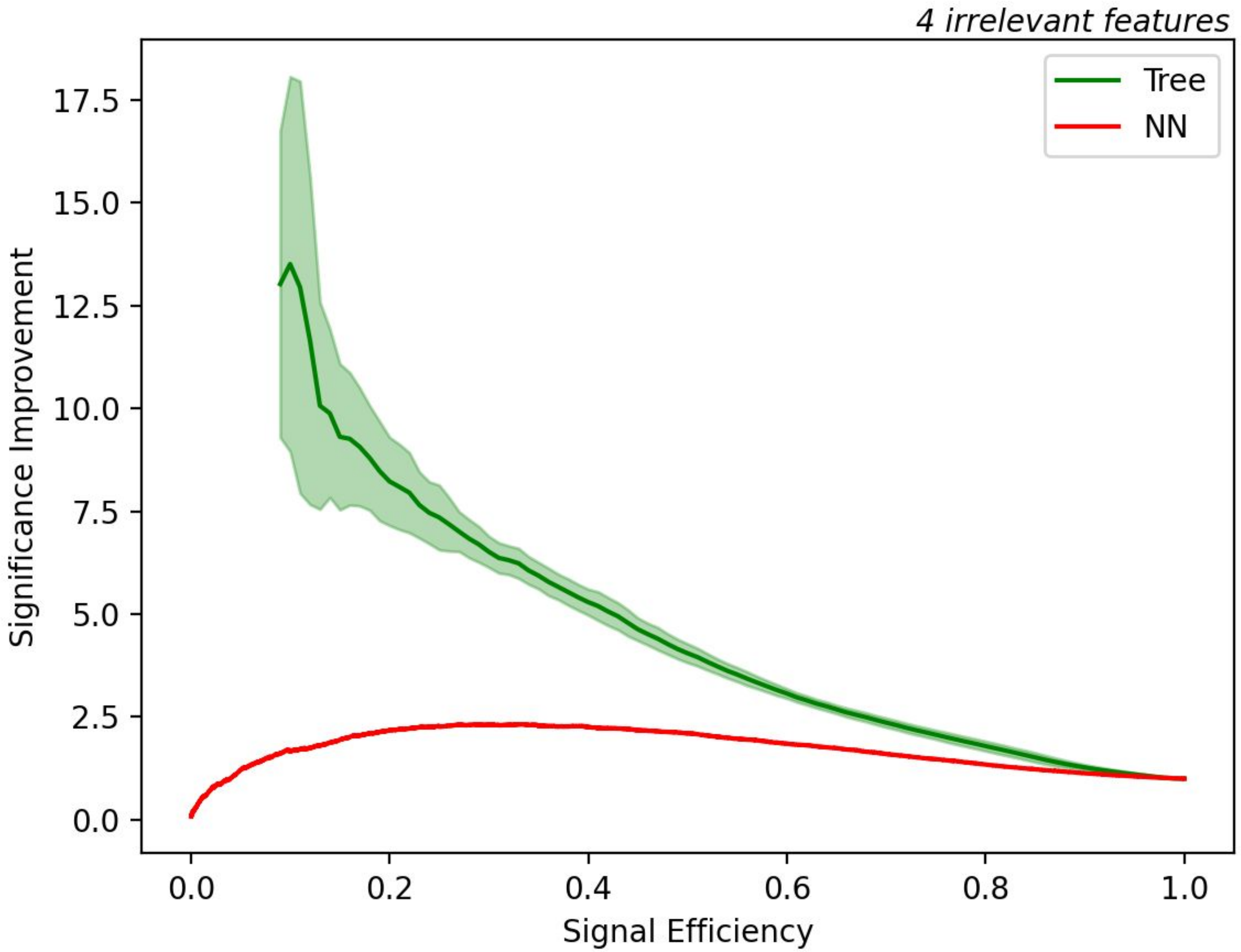
Density Estimation With Trees

- A boosting-inspired tree-based density estimator [Ma and Awaya, 2101.11083]
  - Conceptually similar to normalizing flow
  - Transformations built from leaf-wise constant functions
  - Fast and performant

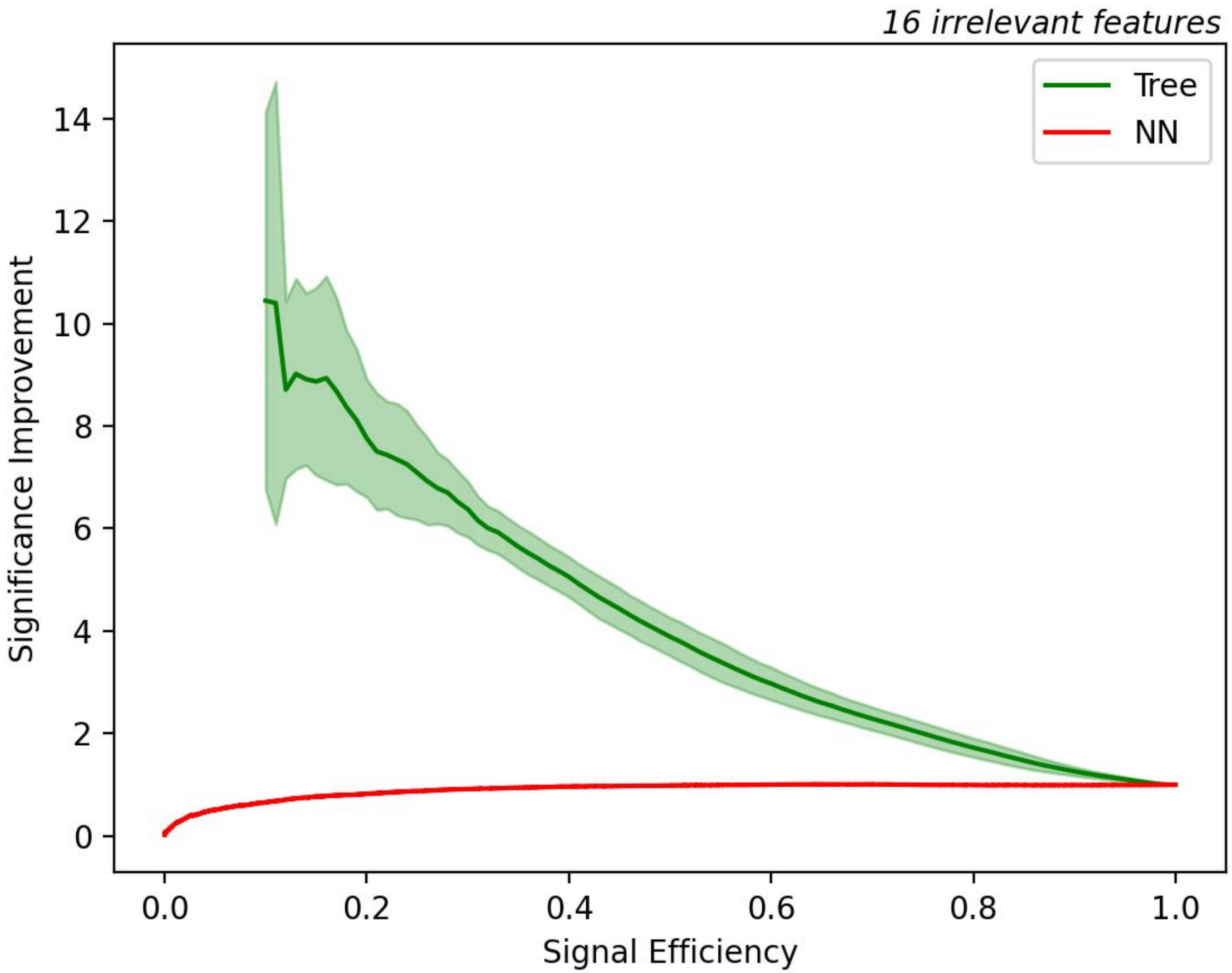
- Why?
  - Why not?
  - *If* a feature is $\perp\!\!\!\perp$ all other features, tend not to cut in such directions



NN Generated by code at https://github.com/HEPML-AnomalyDetection/CATHODE



NN Generated by code at https://github.com/HEPML-AnomalyDetection/CATHODE



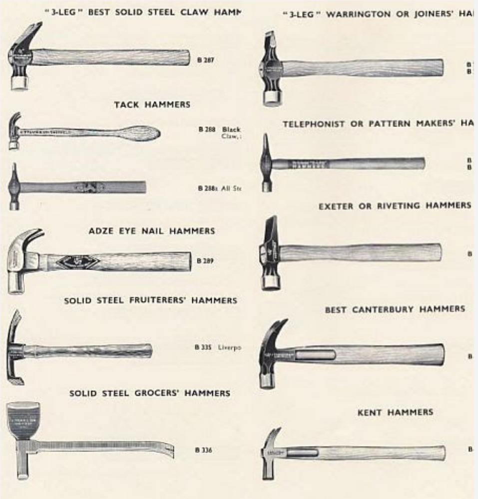
NN Generated by code at https://github.com/HEPML-AnomalyDetection/CATHODE

More possibilities…

- More realistic model of irrelevant features (see previous talk by Marie Hein)
- Fancier interpolation schemes
- More state-of-the-art method, e.g. CATHODE

Conclusion

- Deep learning is *not* all you (should) need
    - Important to understand types and properties of data under analysis
    - Tree-based models can still be powerful in terms of performance, speed and robustness
    - Quality of background interpolation remains an important issue



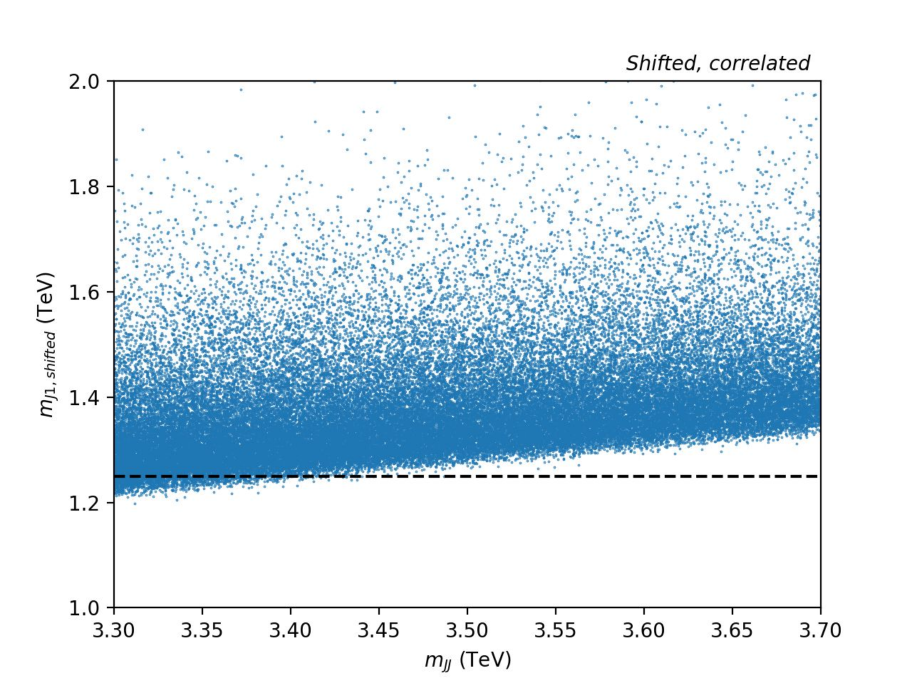
Thank you for coming to my **ML4Jets 2023** talk!

Back-up: xgboost hyperparameter optimizations

- Metric: tpr at fixed fpr=0.001
- 10 fold cross validations
- Bayesian optimization
  - scan hyperparameter space via gaussian process regression

n_estimators	max_depth	eta	alpha	lambda	subsample
292	9	6.2×10^{-3}	50	74	0.75

Back-up: Correlated Features in ANODE



$$m_{J_1} \to m_{J_1} + \log(m_{JJ})$$

$$\Delta m_J \to \Delta m_J + \log(m_{JJ})$$

$$p(\vec{x}|m, \text{bkgd}) \not\approx p(\vec{x}|m_L) + \frac{p(\vec{x}|m_R) - p(\vec{x}|m_L)}{m_R - m_L}(m - m_L)$$

Back-up: A simple decorrelation scheme

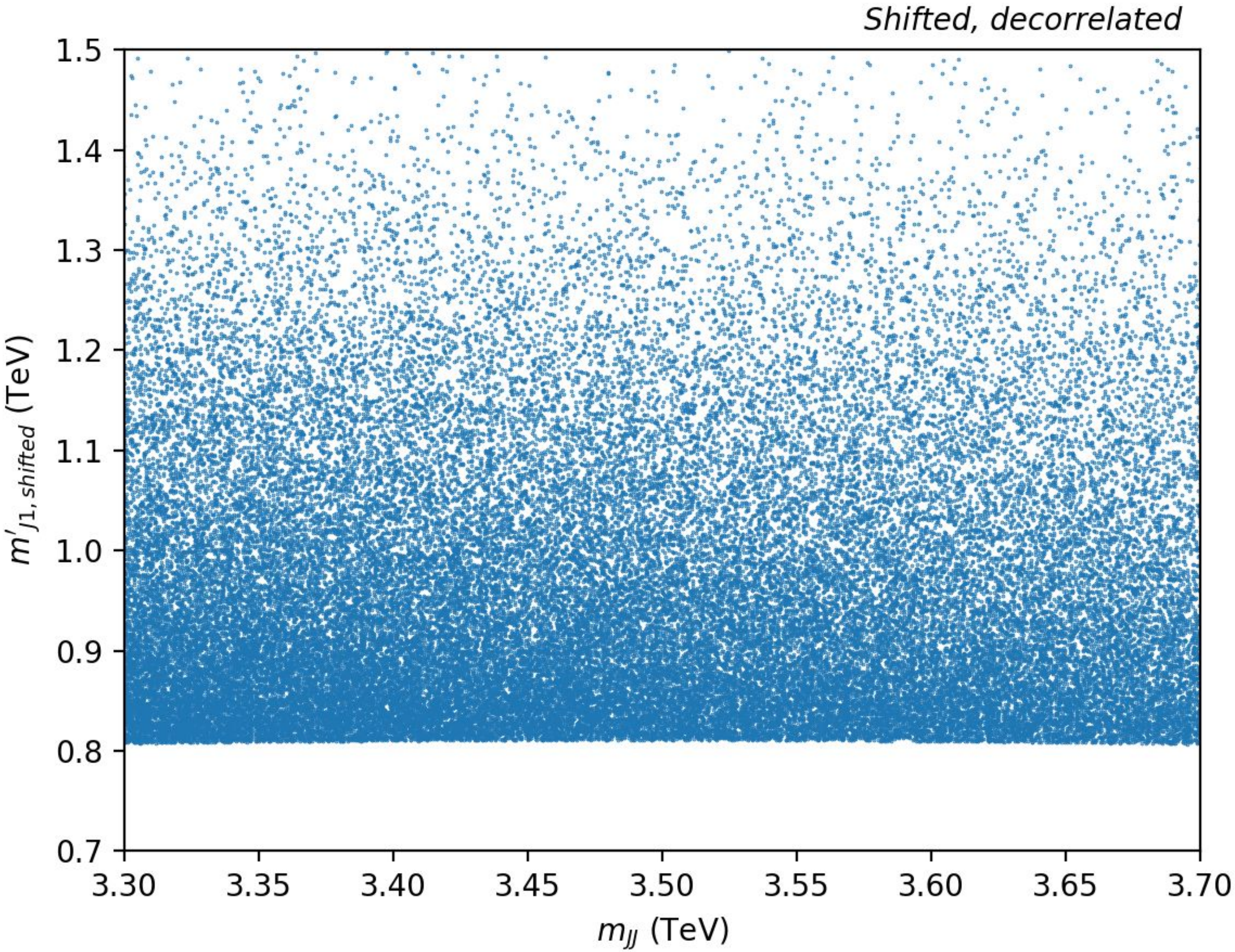
Ad-hoc-ness warning

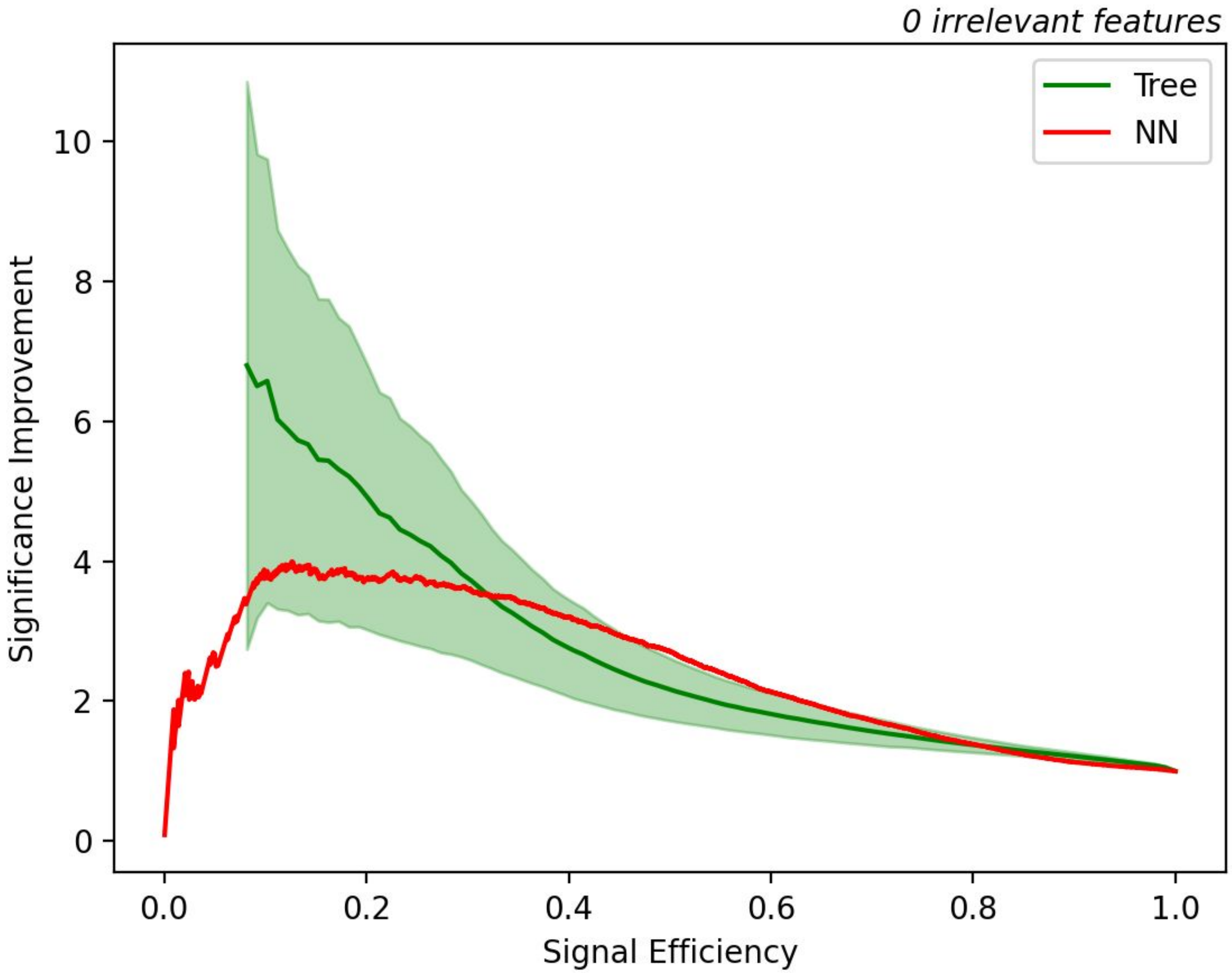
$$\left.\begin{array}{l} m \to m \\ x \to f(x, m) \end{array}\right\} \quad \text{such that } \mathrm{dCorr}(x, m) \text{ is minimized}$$

Needs to be invertible!

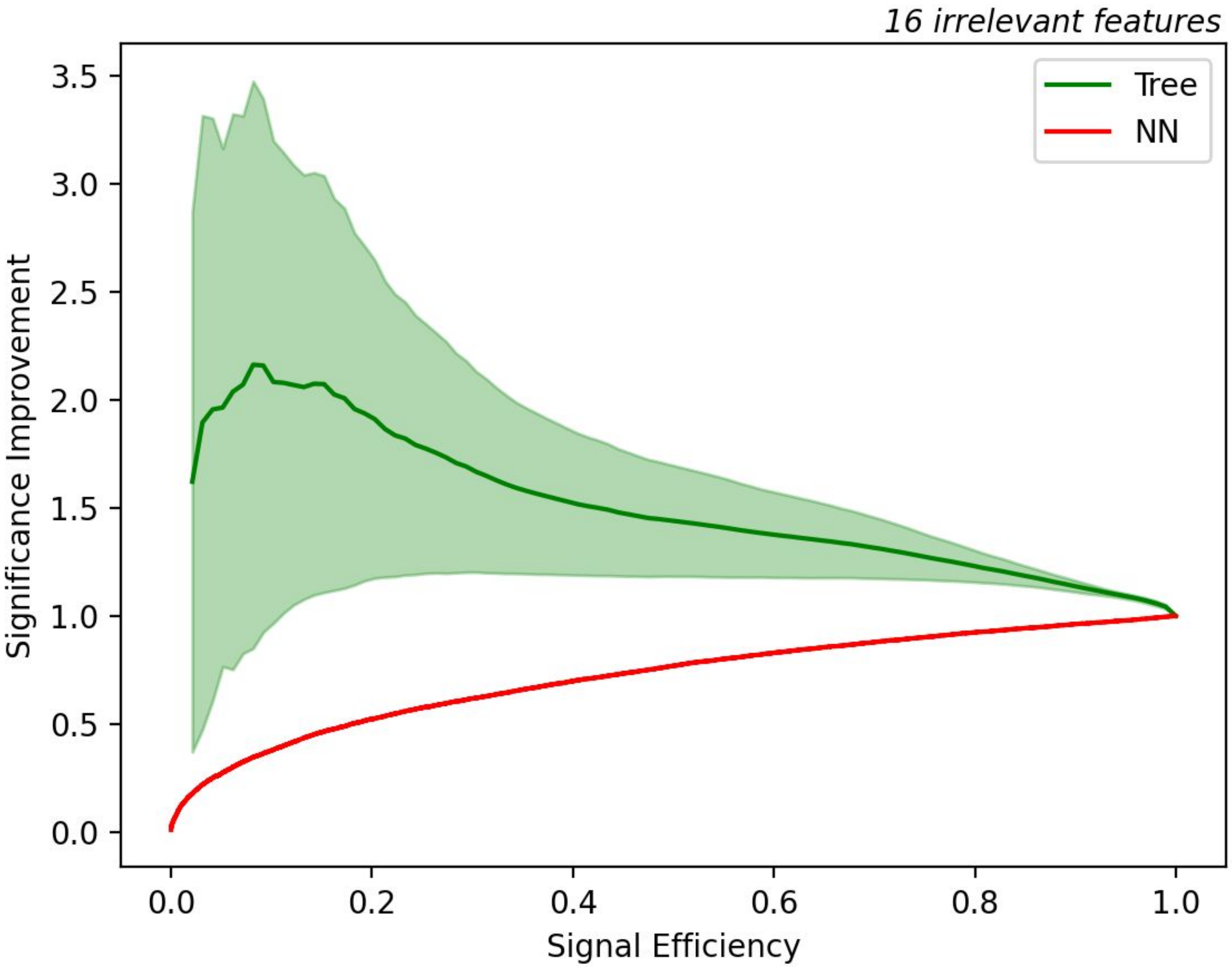
One simple choice: $f(x, m) = a_0(m) + a_1(m)x$

$\alpha m + \beta m^2$ $\qquad$ $1 + \gamma m + \delta m^2$


Shifted, decorrelated
1.5
1.4
1.3
1.2
1.1
1.0
0.9
0.8
0.7
$m'_{J1,shifted}$ (TeV)
3.30
3.35
3.40
3.45
3.50
3.55
3.60
3.65
3.70
m_{JJ} (TeV)

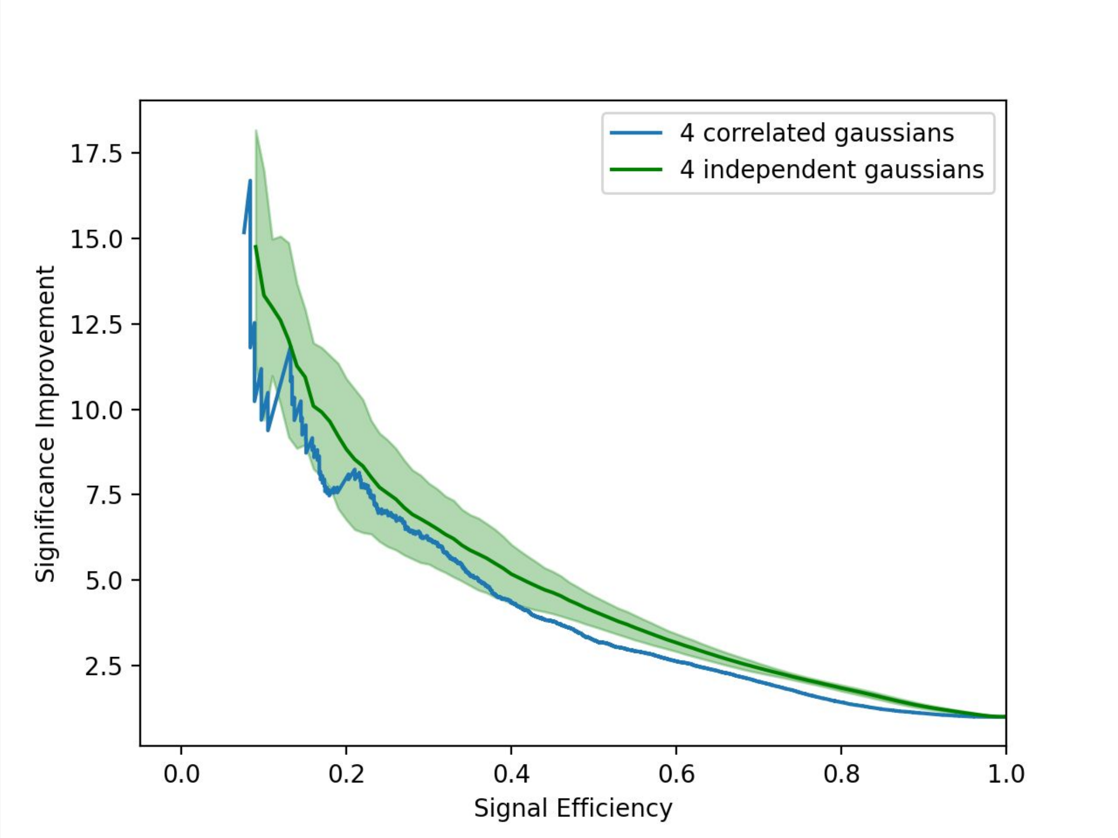


NN Generated by code at https://github.com/HEPML-AnomalyDetection/CATHODE



NN Generated by code at https://github.com/HEPML-AnomalyDetection/CATHODE

Back-up: Correlated gaussian noises for ANODE



Back-up: Copula

- Density estimation quality can often be improved by separating task of estimating marginals and task of estimating dependence structure
- Sklar's theorem [Sklar, 1959]:

$$p(x_1, \ldots, x_n) = c(F_1(x_1), \ldots, F_n(x_n))p(x_1) \cdots p(x_n)$$

- If x_n is independent of all others: c is independent of x_n
    - Trees can benefit from this

Back-up: Precise definition of irrelevance

- Set of all features: $\{x_1, \ldots, x_n\}$
- x_i is called irrelevant iff

$$p(Y|S_i, x_i) = p(Y|S_i), \quad \forall S_i \subset \{x_1, \ldots, \hat{x}_i, \ldots, x_n\}$$

  - Irrelevancy is *not* intrinsic!
  - Does *not* imply x_i is independent of relevant ones
  - Question: How likely is an irrelevant feature dependent on relevant one?