Unsupervised Anomaly Detection with CATHODE

14th Annual Meeting of the Helmholtz Alliance "Physics at the Terascale"

Gregor Kasieczka, Tobias Quadfasel, Manuel Sommerhalder
(Institute of Experimental Physics, University of Hamburg)

23.11.2021

CLUSTER OF EXCELLENCE
QUANTUM UNIVERSE

» Anomaly Search

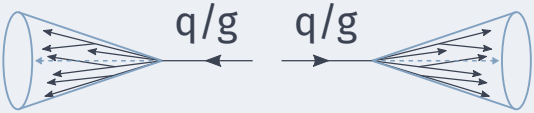
Majority of searches for new physics rely heavily on both signal and SM background models

Impossible to cover all models/phase space regions with a dedicated search
→Need **model-independent** searches

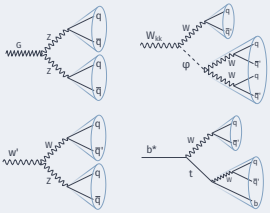
Test scenario:

→Dijet anomaly search **X→YZ**, Y & Z decaying hadronically
→Look for **resonant** new physics with **anomalous jet substructure**

Ultimately want to tell this:

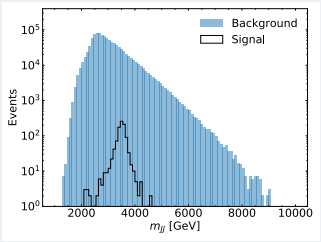


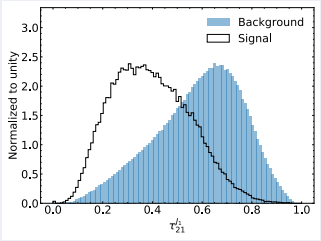
from any of these:



Benchmark Dataset

- Benchmark: LHC Olympics 2020 challenge R&D dataset (*arxiv:2101.08320*)
- Background: 1M simulated QCD multijet events
- Signal: 100k W'→YZ events where Y→qq and Z→qq
- $m_{W'} = 3.5\,\text{TeV}$, $m_Y = 500\,\text{GeV}$, $m_Z = 100\,\text{GeV}$
- Input: 4 variables
  - Lower jet mass m_{j1}
  - mass difference $\Delta m_{1,2}$
  - Jet subjettiness ratios $\tau_{21,j1}$ and $\tau_{21,j2}$





arXiv:2001.04990

Given distributions of signal $p_S(\boldsymbol{x})$ and background $p_B(\boldsymbol{x})$ in some set of variables $\boldsymbol{x}$,
Neyman-Pearson-Lemma:
→best test based on likelihood ratio

Problem: Signal is buried under large amount of background
→We can't estimate $p_S(\boldsymbol{x})$ directly

The best we can do: Estimate $p_{S+B}(\boldsymbol{x}|\mathrm{SR})$ in "Signal Region" and $p_B(\boldsymbol{x}|\mathrm{SB})$ from region without signal ("Sidebands")
→Conditional variable containing resonance: m_{jj}

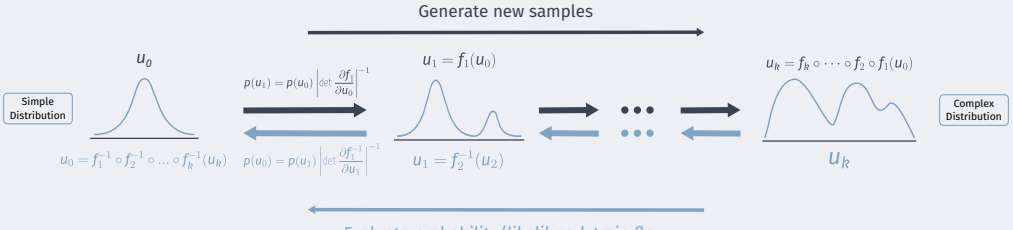
We need to take LR in SR:
→Interpolate $p_B(\boldsymbol{x}|\mathrm{SB})$ into SR
→Construct estimate of LR: $\frac{p_{S+B}(\boldsymbol{x}|SR)}{p_B(\boldsymbol{x}|SR)}$

Get estimate of LR using:
→**classification**
→**density estimation**
(Normalizing Flows)

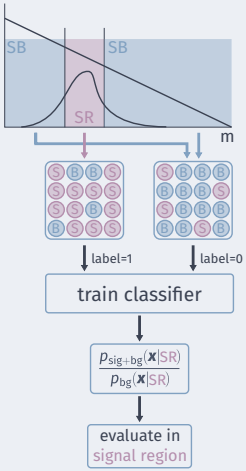
» Normalizing Flows

- Flows based on random variable transformation
- $f : U \to X;\ p(x) = p(u) \left| \frac{df(u)}{du} \right|^{-1}$
- Learn invertible mapping f from latent variables $\boldsymbol{u}$ to data $\boldsymbol{x}$
- Flow: stack many invertible transformations f_i: $f = f_k \circ \ldots \circ f_2 \circ f_1$

$$p(\boldsymbol{x}) = p(f^{-1}(\boldsymbol{x})) \prod_i \left| \det \left(\frac{\partial f_i^{-1}}{\partial \boldsymbol{x}} \right) \right| = p(\boldsymbol{u}) \prod_i \left| \det \left(\frac{\partial f_i}{\partial \boldsymbol{u}} \right) \right|^{-1}$$



Classification Without Labels (CWoLa)

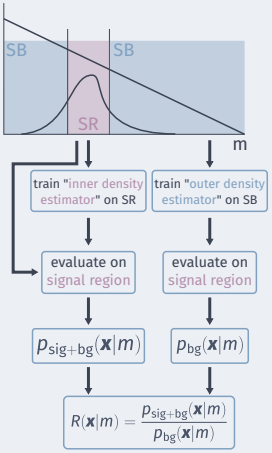


\+ Simple classification task

\+ Basic DNN architecture

\− Highly dependent on correlations between $\boldsymbol{x}$ and m
→Variables $\boldsymbol{x}$ need to be hand-picked

Anomaly Detection with Density Estimation (ANODE)



\+ Direct estimation of conditional densities

\+ Easy to interpolate p_{bg}

\+ Robust against correlations between $\boldsymbol{x}$ and m
→Arbitrary choice of $\boldsymbol{x}$

\− Computationally intense

\− Estimation of signal contribution difficult

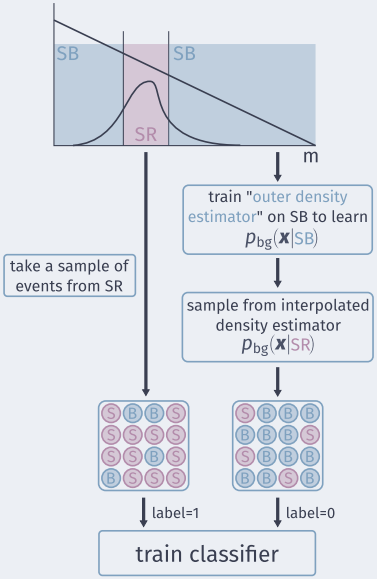
DOI:10.1007/JHEP10(2017)174

DOI:10.1103/PhysRevD.101.075042

CATHODE

Classifying Anomalies THrough Outer Density Estimation (CATHODE)

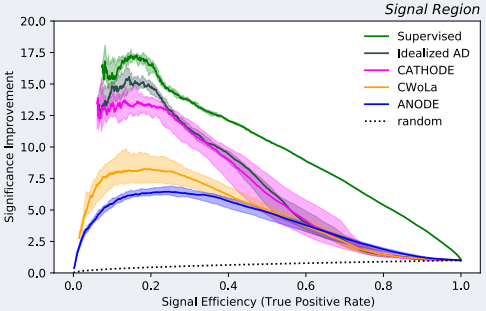
+ Only one density estimator needed
+ Due to interpolation: robust against correlations between $\boldsymbol{x}$ and m →Arbitrary choice of $\boldsymbol{x}$
+ Final tagger based on simple classification task
+ No density estimator for signal contribution needed
− Computationally intense



arxiv:2109.00546

- Most important performance measure: **significance improvement characteristic (SIC)**
- ”Supervised” training using signal/background labels →overall upper performance limit
- ”Idealized AD”: distinguish actual sample vs. background-only sample from signal region →upper limit for unsupervised anomaly search
- CATHODE shows highest SIC amongst non-idealized anomaly taggers
- Performance reaches idealized AD limit
- Significance improvement about factor 14

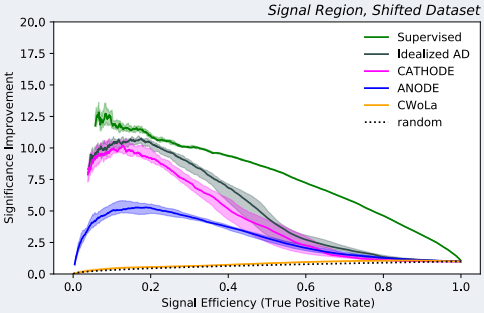
$$\text{SIC} = \frac{\left.\frac{S}{\sqrt{B}}\right|_{\text{cut}}}{\left.\frac{S}{\sqrt{B}}\right|_{\text{no cut}}} = \frac{\text{TPR}}{\sqrt{\text{FPR}}}$$



arxiv:2109.00546

Robustness Against Correlations

- Study impact of correlations between $\boldsymbol{x}$ and m_{jj}
- Introduce artificial correlations
- Add 10% of corresponding m_{jj} value to m_{j1} and $\Delta m_{1,2}$
- All methods suffer performance →"Smearing" of variables
- CWoLa performance completely breaks down
- CATHODE retains good performance, similar to idealized AD



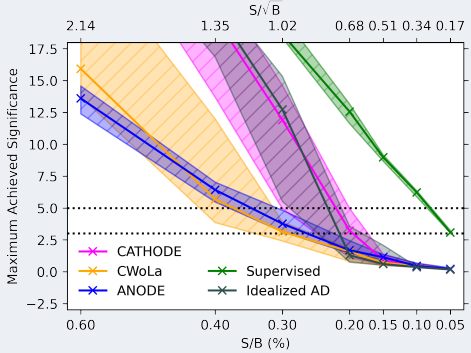
arxiv:2109.00546

» Summary

- Investigation of dijet resonances with anomalous jet substructure using normalizing flow–based density estimation & classification
- Introduction of new method: CATHODE that combines the advantages of purely density estimation–based (ANODE) and classification-based (CWoLa) approaches
- CATHODE outperforms all other non-idealized anomaly detectors
- Performance similar to idealized anomaly detector
- Robust against correlations between features $\boldsymbol{x}$ and conditional variable m_{jj}
- Future studies
  - Other datasets/topologies
  - Study sensitivity for different types of anomalies (e.g. very broad resonances)
  - Studies using more (low-level) features

BACKUP

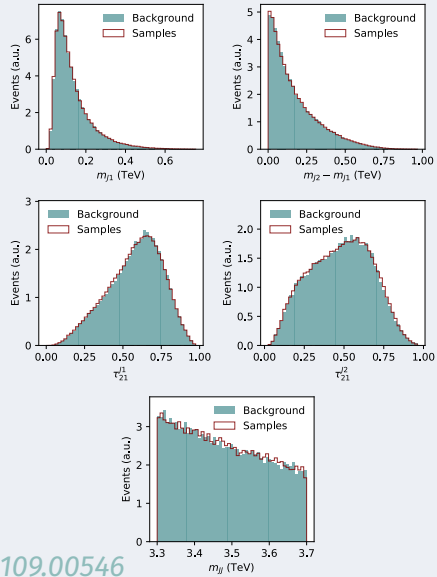
- Comparison for different amounts of signal injected
- CATHODE outperforms other AD methods significantly down to a S/B as low as 0.3%
- CATHODE achieves similar performance as idealized anomaly detector
- Below 0.2% S/B: even idealized AD cannot raise significance above 3σ
  →Too limited number of data points in the signal region



arxiv:2109.00546

Training & Sampling

- Flow trained for 100 epochs
- Model ensembling: pick 10 epochs with lowest validation loss
- Draw m_{jj} values from a KDE in signal region
- Use these values as conditional and sample from density estimator
  →Interpolation into SR
- We can oversample to produce more samples than we have in data
- Background densities are modelled well by flow

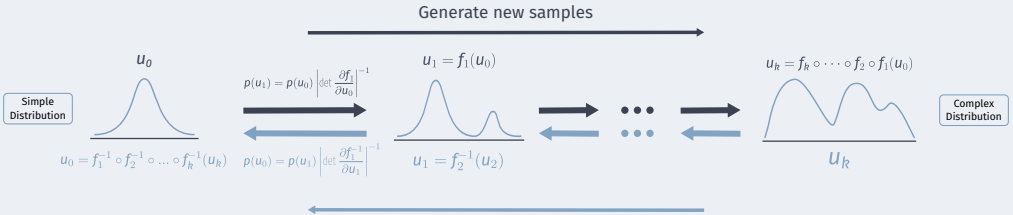


arxiv:2109.00546

» Normalizing Flows

- Flows based on random variable transformation
- $f : U \to X;\ p(x) = p(u) \left| \frac{df(u)}{du} \right|^{-1}$
- Multivariate case: scale with Jacobian determinant
- Flow: repeat (invertible) transformation to get complex distributions

$$p(\boldsymbol{x}) = p(f^{-1}(\boldsymbol{x})) \prod_i \left| \det \left(\frac{\partial f_i^{-1}}{\partial \boldsymbol{x}} \right) \right| = p(\boldsymbol{u}) \prod_i \left| \det \left(\frac{\partial f_i}{\partial \boldsymbol{u}} \right) \right|^{-1}$$

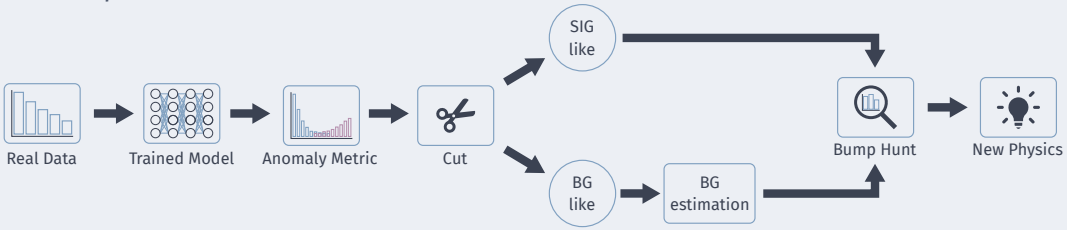


CASE (**C**MS **A**nomaly **SE**arch):
Explore different techniques, all involving **ML models trained directly on data!**

Currently investigated methods:

- (Variational) autoencoders
- Weak supervision (CWoLa hunting, Tag N' Train)
- Quasi-anomalous knowledge (QUAK)
- Density estimation & classification (CATHODE)

General procedure:



» Autoregressive Flows

Autoregressive property:

$p(x) = \prod_i p(x_i|x_{1:i-1})$

Conditional densities depend on trainable parameters:

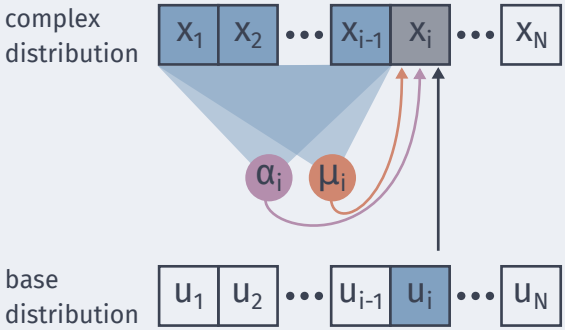
$p(x_i|x_{1:i-1}) = \mathcal{N}(x_i|\mu_i, (\exp \alpha_i)^2)$

$\mu_i = f_{\mu_i}(x_{1:i-1})$

$\alpha_i = f_{\alpha_i}(x_{1:i-1})$

→Earlier variables must not depend on later variables
→Solution: stack transformations into a normalizing flow, change ordering of the x_i after each transformation

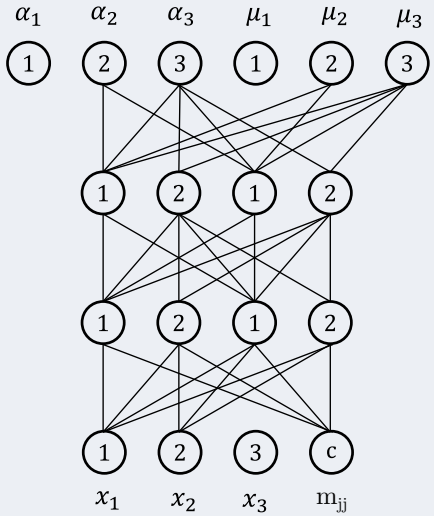
$x_i = f(u_i) = u_i \cdot \exp(\alpha_i) + \mu_i$



Autoregressive property →Jacobian is upper triangular

$$\left|\det\left(\frac{\partial f}{\partial \boldsymbol{u}}\right)\right| = \exp\left(\sum_i \alpha_i\right)$$

- DNN architecture to implement a single f_i in autoregressive flows (1502.03509)
- Compute α and μ in one forward pass
- Outputs α_j and μ_j only connected to inputs $\{x_1, ..., x_{j-1}\}$ →autoregressive property
- No connection dropped for conditional input m_{jj}



- Stack MADE networks to build "Masked Autoregressive Flow" (MAF)
- Learn tranformation $\boldsymbol{u} = f^{-1}(\boldsymbol{x})$ from input features $\boldsymbol{x}$ to $\boldsymbol{u} \sim \mathcal{N}(0, \mathbb{I})$
- Compute $p(\boldsymbol{x})$ with normalizing flow from $p(\boldsymbol{u})$
- Minimize NLL loss $\mathcal{L} = -log(p(\boldsymbol{x}))$
- Architecture used for both density estimators