ML Density estimation for anomaly searches

Project B3b

CRC Annual Meeting 2024

from arXiv:2303.07364, arXiv:2312.03067, arXiv:2309.13111

Luigi Favaro

the B3b group M. Krämer, T. Plehn, A. Mück, M. Hein, T. Modak, T. Finke

March 12, 2024

UNIVERSITÄT HEIDELBERG
ZUKUNFT SEIT 1386

Funded by
DFG Deutsche Forschungsgemeinschaft
German Research Foundation

Table of Contents

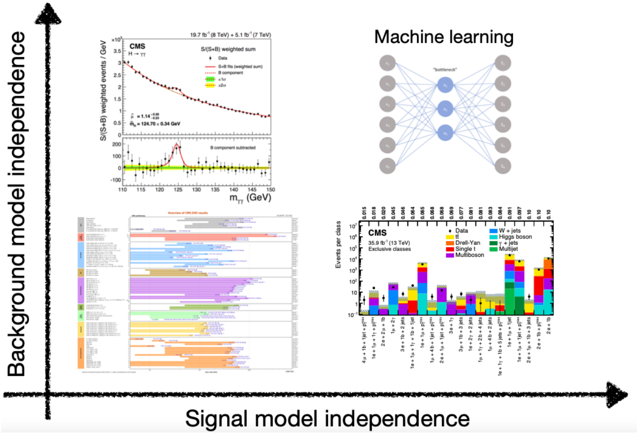
Anomaly searches

- We are still looking for BSM physics;
- No clear anomalies in the near future:
  - direct searches are not feasible;
  - reduce model assumption in favor of agnostic methods;
- No loss in sensitivity

Are we fully exploring our data?

Project B3b

1 Introduction



*M. Krämer, CRC annual meeting 2023

Density estimation

1 Introduction

- Model agnostic $\rightarrow$ no signal involved;
- Estimating density of high-dimensional spaces:
  - estimate background density (e.g. QCD jets);
  - likelihood-ratio in signal and control regions;

- Model agnostic $\rightarrow$ no signal involved;
- Estimating density of high-dimensional spaces:
  - estimate background density (e.g. QCD jets);
  - likelihood-ratio in signal and control regions;
- New architectures is not all you need!

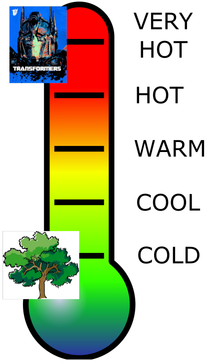


Table of Contents

Framing the problem

2 Learning the language of QCD

*based on "Learning the Language of QCD jets with transformers", Finke T. et al., arXiv:2303:07364

Treating jets as sentences:

- discretize jets in p_T, $\Delta\eta$, $\Delta\phi$;
- model the density auto-regressively;
- perform inference and sampling;

Framing the problem

2 Learning the language of QCD

*based on "Learning the Language of QCD jets with transformers", Finke T. et al., arXiv:2303:07364

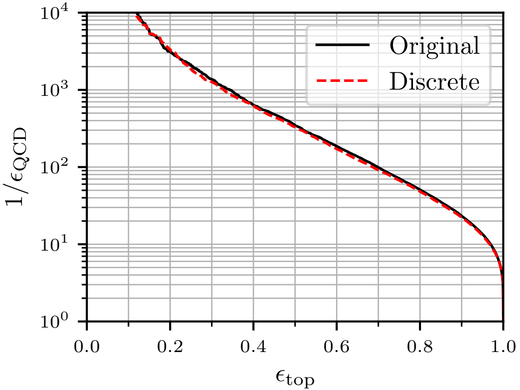
Jets have continuous features, words are not:

- discretize jets in p_T, $\Delta\eta$, $\Delta\phi$;
- model the density auto-regressively;
- perform inference and sampling;

*based on "Learning the Language of QCD jets with transformers", Finke T. et al., arXiv:2303:07364

Jets have continuous features, words are not:

- discretize jets in p_T, $\Delta\eta$, $\Delta\phi$;
- model the density auto-regressively;
- perform inference and sampling;



A `ParticleNet` classifier shows no information loss from discretization

Training procedure

Training follows closely NLP approaches:

- embed features in continuous vectors;

Training procedure

Training follows closely NLP approaches:

- embed features in continuous vectors;
- perform self-attention;

$$x_i' = A_{ij} v_j, \quad A_{ij} = Softmax\left(\frac{W^Q x_i W^k x_j}{\sqrt{d}}\right) \quad (1)$$

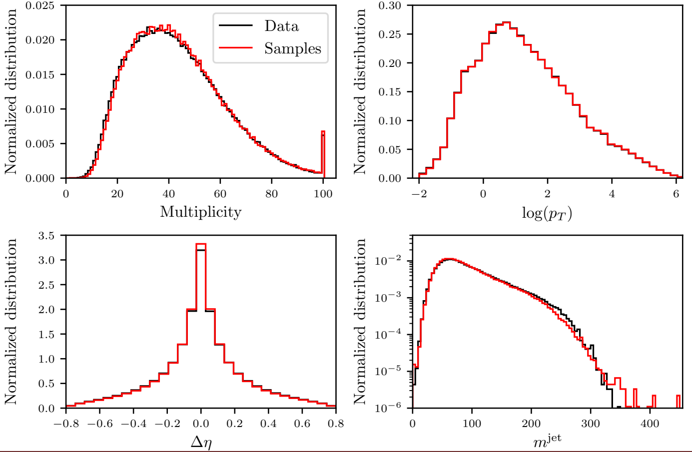
Training follows closely NLP approaches:

- embed features in continuous vectors;
- perform self-attention;
- minimize cross-entropy with SGD.

$$x_i' = A_{ij} v_j, \quad A_{ij} = Softmax(\frac{W^Q x_i W^k x_j}{\sqrt{d}}) \quad (1)$$

$$- - \log p_\theta (x) = - \sum_i \sum_{j<i} \log p_\theta (x_i | x_j) \quad (2)$$

Samples from the network:



Evaluating generative networks

2 Learning the language of QCD

- Classifiers are the best tools we have to test generative networks;
  - see also arXiv:2305.16774
- The output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \qquad \longrightarrow \qquad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

Evaluating generative networks

- Classifiers are the best tools we have to test generative networks;
  - see also arXiv:2305.16774
- The output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \quad \longrightarrow \quad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

- Optimal observable for a two hypothesis test according to the Neyman-Pearson lemma

Evaluating generative networks

- Classifiers are the best tools we have to test generative networks;
  - see also arXiv:2305.16774
- The output approximates the quantity:

$$C(x) = \frac{p_{true}(x)}{p_{true}(x) + p_{model}(x)} \qquad \longrightarrow \qquad \frac{p_{true}}{p_{model}}(x) = \frac{C(x)}{1 - C(x)}$$

- Optimal observable for a two hypothesis test according to the Neyman-Pearson lemma
- Proper training is essential: architecture, over-fitting, calibration, . . .

Evaluating generative networks

- Evaluation based on threshold-free quantities;
  - ROC curves;
  - corresponding AUC;

- Evaluation based on threshold-free quantities;
    - ROC curves;
    - corresponding AUC;
- Classifier builds an approx. $w = p_{true}/p_{model}$;

Evaluating generative networks

2 Learning the language of QCD

- Evaluation based on threshold-free quantities;
  - ROC curves;
  - corresponding AUC;
- Classifier builds an approx. $w = p_{true}/p_{model}$;
- **AUC=0.62**

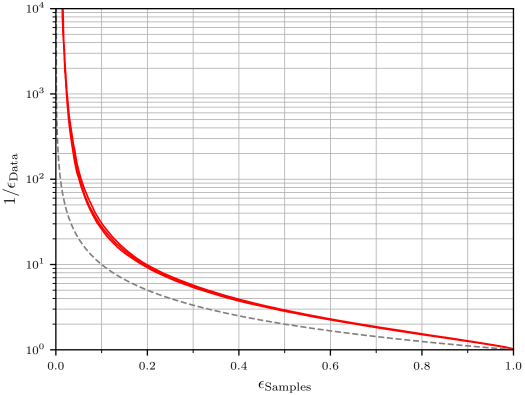


Table of Contents

Dark showers

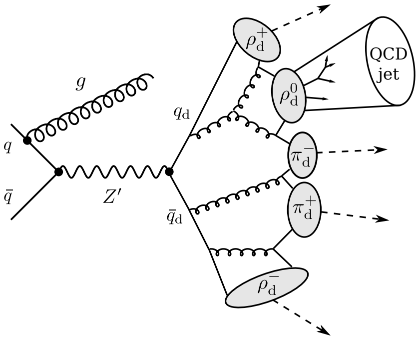
3 Representing dark showers

Create a representation space sensitive to dark jets

Benchmark signal: semi-visible jets

- $Z' = 2$TeV dark sects mediator;
- q_d dark quarks charged under $SU(3)_d$;
- $m_{q_d} = 500$MeV;
- $\Lambda = m_{\pi_d} = m_{\rho_d} = 5$GeV;

QCD-like showers with fraction of invisible particles



Self-supervision

- Neural Networks are not invariant to physical symmetries in data
- Typically solved through “pre-processing”
- Self-supervision: during training we use pseudo-labels, not truth labels

Key aspects of representations:

- invariance to certain transformations of the jet/event
- discriminative power

In CLR we construct a mapping to a new representation space

*based on JetCLR, arXiv:210804253 and "Semivisible-jets, energy-based models and self-supervision", arXiv:2312:03067

Contrastive Learning for anomaly detection:

- positive pairs: $\{(x_i, x_i')\}$ where is an augmented version of x_i;
- anomalous pairs: $\{(x_i, x_i^*)\}$ where x_i^* is motivated by BSM;

Augmentation: any transformation (e.g. rotation) of the original jet *based on JetCLR, arXiv:210804253 and "Semivisible-jets, energy-based models and self-supervision", arXiv:2312:03067

Contrastive Learning for anomaly detection:

- positive pairs: $\{(x_i, x_i')\}$ where is an augmented version of x_i;
- anomalous pairs: $\{(x_i, x_i^*)\}$ where x_i^* is motivated by BSM;

Augmentation: any transformation (e.g. rotation) of the original jet

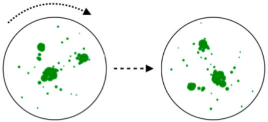
Train a Transformer-encoder network to map the data to a new repr. space, $z : \mathcal{Z} \to \mathcal{R}$

$$\mathcal{L} = s(z_i, z_i^*) - s(z_i, z_i') \qquad s(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i||z_j|} \tag{3}$$

Augmentations

3 Representing dark showers

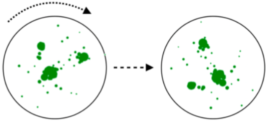
rotations in $[0, 2\pi]$:



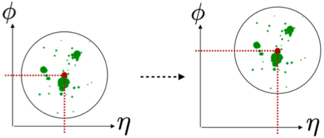
Augmentations

3 Representing dark showers

rotations in $[0, 2\pi]$:



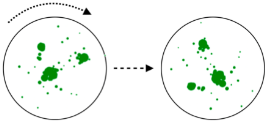
translations in $[\eta, \phi]$:



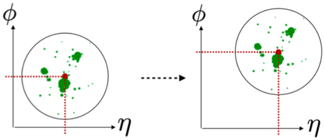
Augmentations

3 Representing dark showers

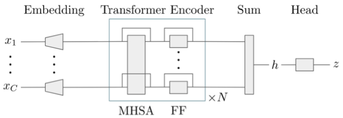
rotations in $[0, 2\pi]$:



translations in $[\eta, \phi]$:

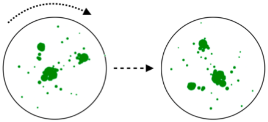


permutation invariance:

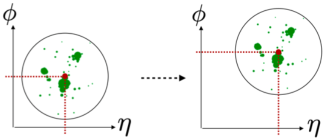


Augmentations

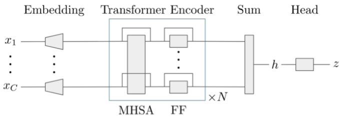
rotations in $[0, 2\pi]$:



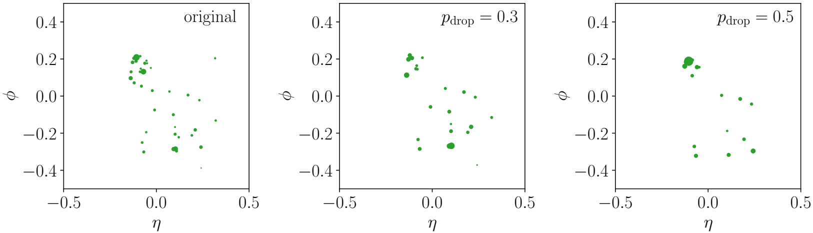
translations in $[\eta, \phi]$:



permutation invariance:



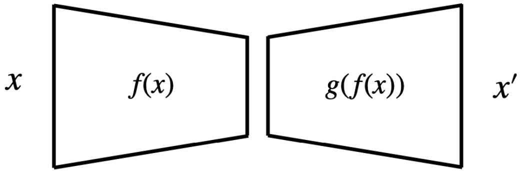
Applying p_{drop} to a QCD jet:



Anomaly scores

- (N)AutoEncoder based anomaly score: $MSE(x, D(E(x)))$

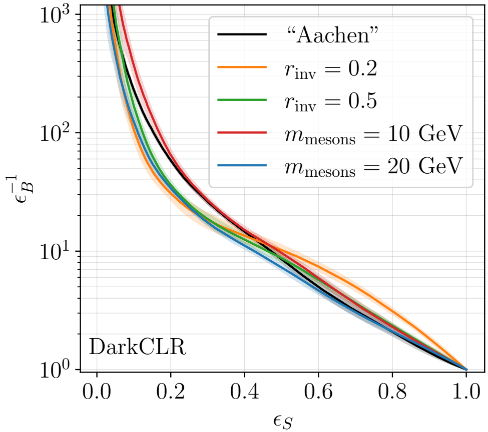
$$p_\theta(x) = \frac{e^{-E_\theta(x)}}{\Omega} \qquad E_\theta(x) = MSE(x, D(E(x))) \tag{4}$$



The corresponding anomaly score will be (approx) invariant to the augmentations

Robustness of darkCLR

3 Representing dark showers



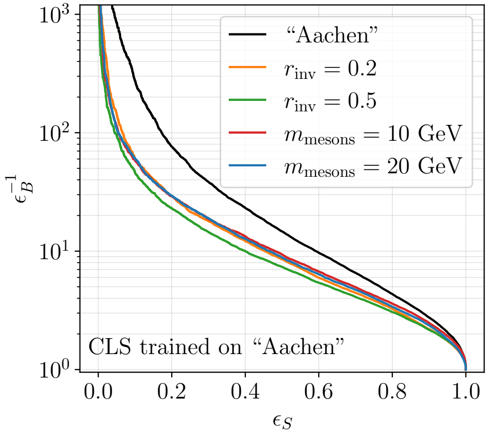


Table of Contents

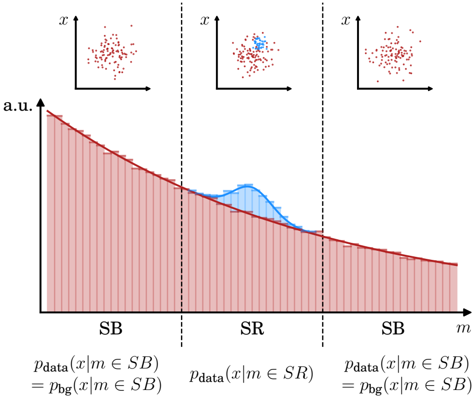
4 Resonant anomaly detection

Resonant AD

4 Resonant anomaly detection

*based on "Back To The Roots: tree-based algorithms for weakly supervised anomaly detection", Finke T. et al., arXiv:2309.13111

- divide feature in signal and control region;
- get a background template in SR;
- train a classifier between datasets with noisy labels;
- w_{noisy} is still optimal:
  - monotonically increasing function of w_{true};



*taken from CATHODE

BDTs for bump hunts

Dataset:

- signal: $W' \to XY$ with $X/Y \to qq$
- $m_{W'} = 3.5\text{TeV}$, $m_X = 0.5\text{TeV}$, $m_Y = 0.1\text{TeV}$;

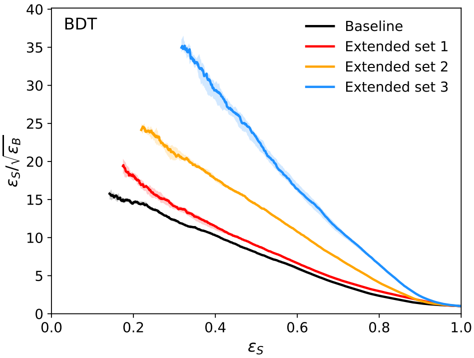
Feature selection is an issue:

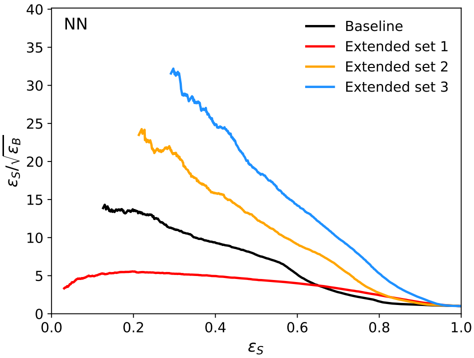
- start from 4 features;
- extended sets up to 56 features
- **"Extended 1" has uninformative features!**

Name	# features	Features
Baseline	4	$\{m_{J_1},\ \Delta m_J,\ \tau_{21}^{\beta=1,J_1},\ \tau_{21}^{\beta=1,J_2}\}$
Extended 1	10	$\{m_{J_1},\ \Delta m_J,\ \tau_{N,N-1}^{\beta=1,J_1},\ \tau_{N,N-1}^{\beta=1,J_2}\}$ for $2 \leq N \leq 5$
Extended 2	12	$\{m_{J_1},\ \Delta m_J,\ \tau_N^{\beta=1,J_1},\ \tau_N^{\beta=1,J_2}\}$ for $N \leq 5$
Extended 3	56	$\{m_{J_1},\ \Delta m_J,\ \tau_N^{\beta,J_1},\ \tau_N^{\beta,J_2}\}$ for $N \leq 9$ and $\beta \in \{0.5, 1, 2\}$

BDTs for bump hunts

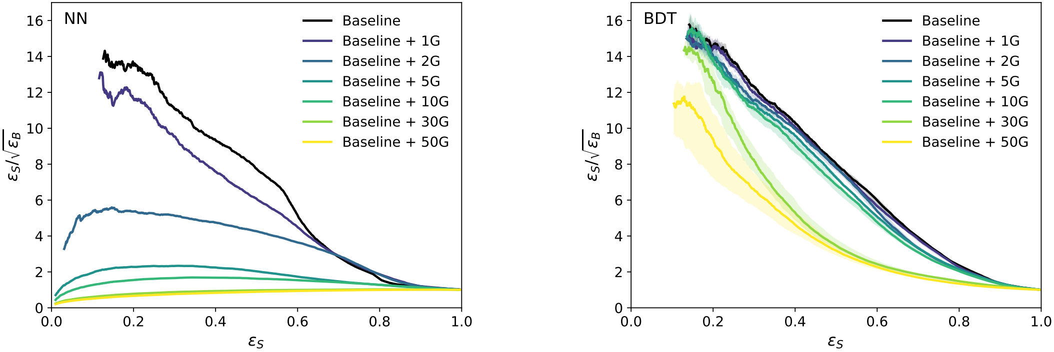
4 Resonant anomaly detection





BDTs for bump hunts

4 Resonant anomaly detection



Conclusions

- Reliable density estimates are essential for future anomaly searches;

Conclusions

- Reliable density estimates are essential for future anomaly searches;
- Transformers are powerful tool for:
  - process jets at constituents level;
  - density estimates and representation learning

Conclusions

5 Conclusions

- Reliable density estimates are essential for future anomaly searches;
- Transformers are powerful tool for:
  - process jets at constituents level;
  - density estimates and representation learning
- Resonant searches seem to prefer BDTs;
- No "winner-take-all", different problems require different solutions;

Conclusions

5 Conclusions

- Reliable density estimates are essential for future anomaly searches;
- Transformers are powerful tool for:
  - process jets at constituents level;
  - density estimates and representation learning
- Resonant searches seem to prefer BDTs;
- No "winner-take-all", different problems require different solutions;

Outlook:

- extending models to more complex signatures;
- include uncertainties;
- **still a lot of work to do!**

Conclusions

- Reliable density estimates are essential for future anomaly searches;
- Transformers are powerful tool for:
  - process jets at constituents level;
  - density estimates and representation learning
- Resonant searches seem to prefer BDTs;
- No "winner-take-all", different problems require different solutions;

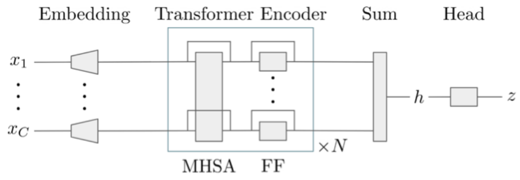
Outlook:

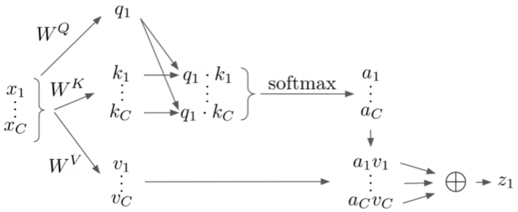
- extending models to more complex signatures;
- include uncertainties;
- **still a lot of work to do!**

Thanks for your attention!

Backup

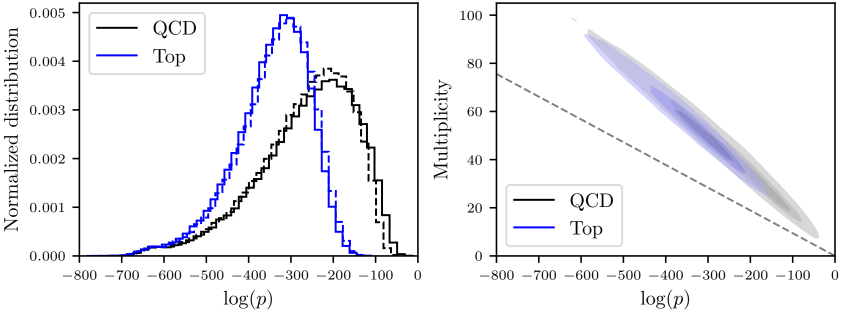
Transformer Encoder

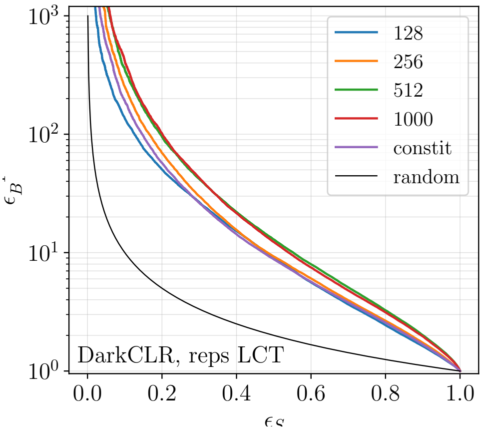


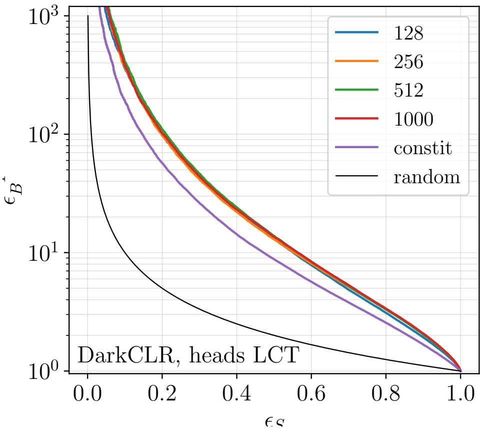


Density estimates

6 Backup




DarkCLR, reps LCT
128
256
512
1000
constit
random
ϵ_B^{-1}
ϵ_S


DarkCLR, heads LCT
128
256
512
1000
constit
random
ϵ_B^{-1}
ϵ_S