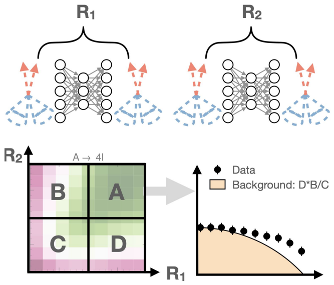
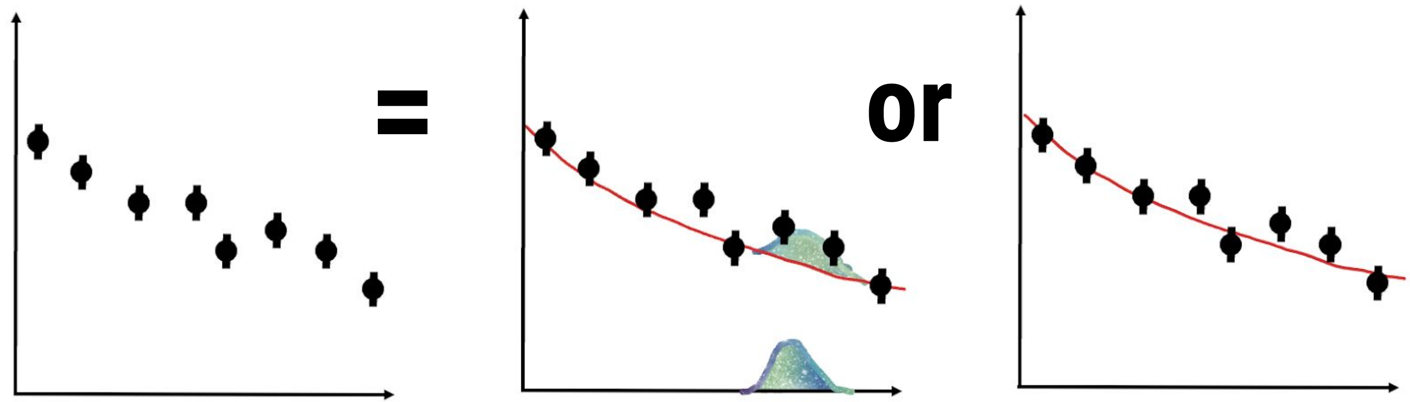
Online-compatible Unsupervised Non-resonant Anomaly Detection

Vinicius M. Mikuni, Benjamin Nachman, David Shih

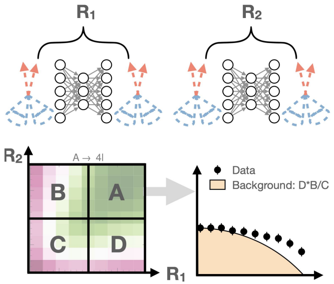
Anomaly detection



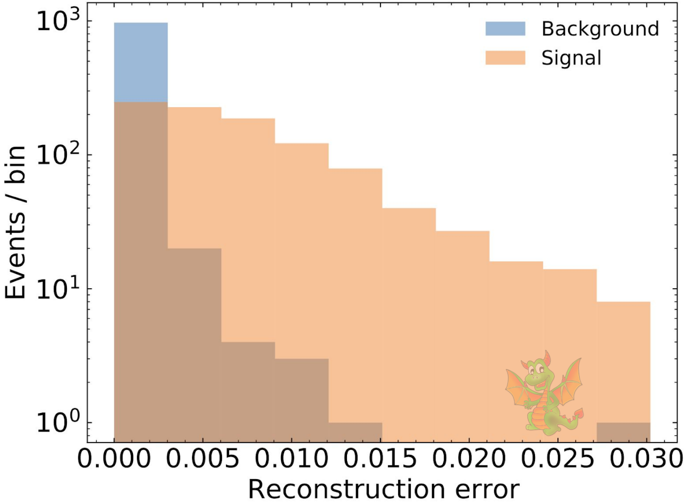
- How to look for **new physics processes** without knowing how they should look like?
- New physics should be rare: **Anomaly detection** 
- Even if you are able to identify “anomalies”, how to **interpret the observation**?
- A good method of anomaly detection requires:
  - A method that **identifies** particle collisions that seem to be **anomalous**
  - Able to provide context: how should **false positives** look like?



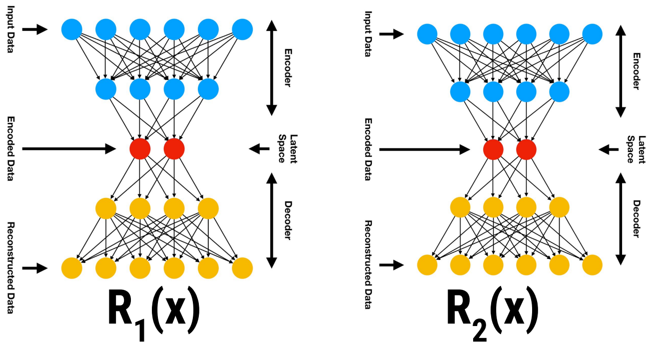
Decorrelated autoencoders



- Anomaly detection based on **autoencoders**: algorithm learns how to **compress** and **decompress** the data using background events
- Events that are **poorly decompressed** are often **rare** and point to anomalous events

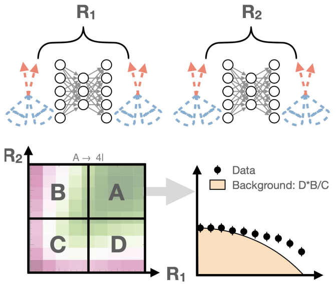


- Train **multiple autoencoders** such that their reconstruction is **independent** for the background



$$L[f_1, f_2, g_1, g_2] = \sum_i R_1(x_i)^2 + \sum_i R_2(x_i)^2 + \lambda \, \mathrm{DisCo}^2[R_1(X), R_2(X)]$$

Anomaly detection performance

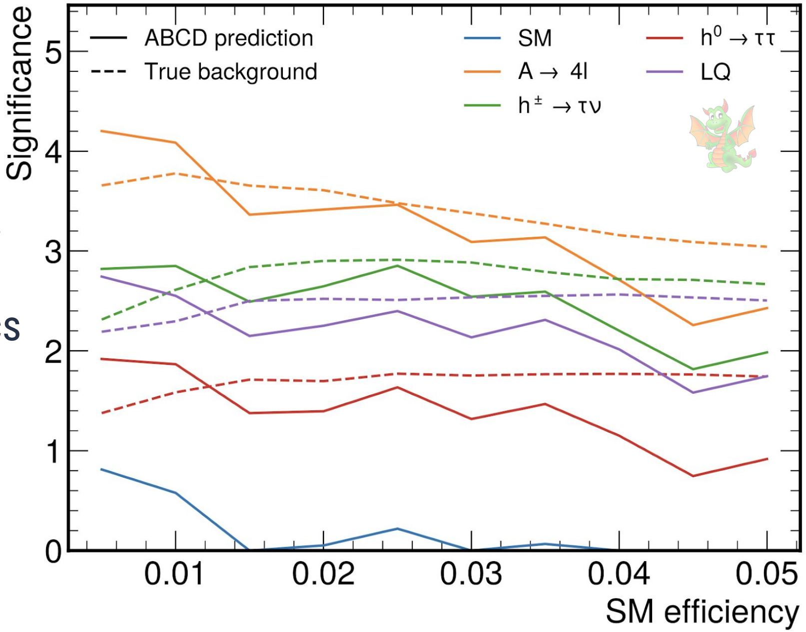


Use the independent reconstructions to estimate the **number of false positives**

- **Significance**: how often your observation is compatible with the **no new physics hypothesis**

No anomalies

Other colors: datasets with **0.1%** anomalies and **99.9%** standard physics processes

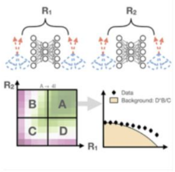


Editors' Suggestion

Online-compatible unsupervised nonresonant anomaly detection

Vinicius Mikuni, Benjamin Nachman, and David Shih

Phys. Rev. D **105**, 055006 (2022) – Published 8 March 2022

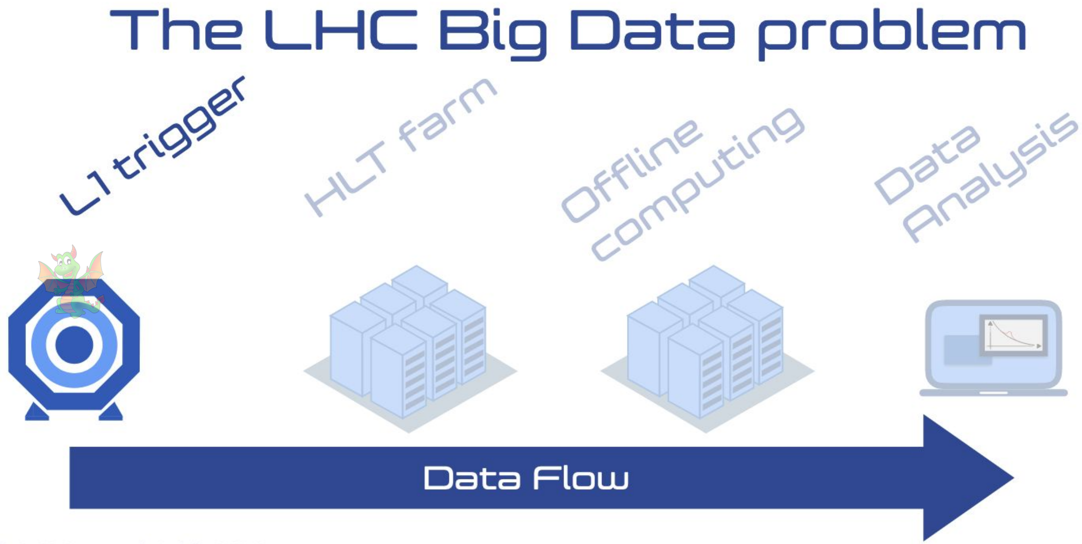


The authors of this paper employ two (or more) autoencoders to provide a complete strategy for unsupervised non-resonant anomaly detection. Both signal extraction and data-driven background estimation can be determined with decorrelated autoencoders. The method shows strong performance on test datasets and has the advantage of being online-compatible.

Show Abstract +

Online compatibility

Slides from [Maurizio Pierini]

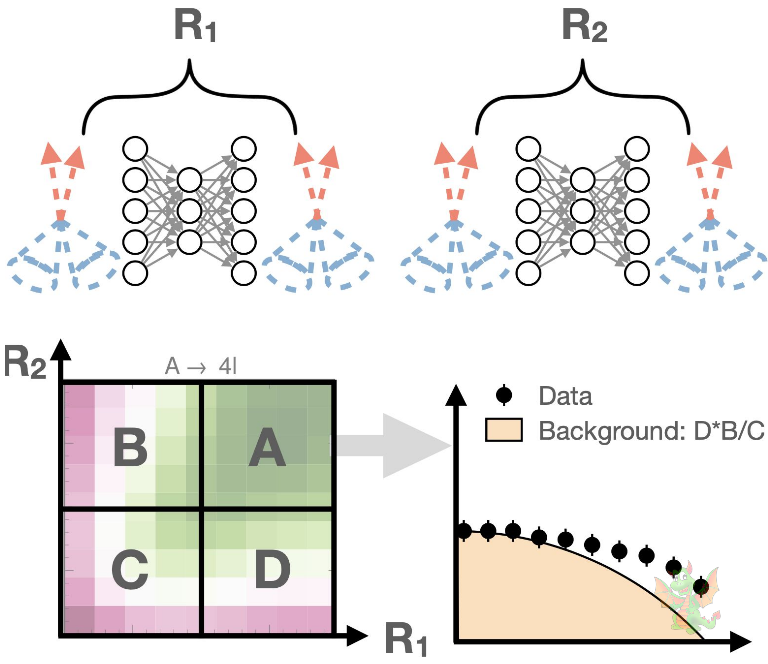


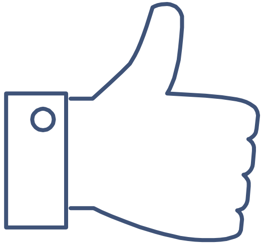
- 40 MHz in / 100 KHz out
- ~ 500 KB / event
- Processing time: ~10 µs
- Based on coarse local reconstructions
- FPGAs / Hardware implemented

- Tune the autoencoder thresholds to **save all events in the signal enriched region**
- **Prescale** the other 3 regions to determine the b**ackground composition**
- Train events using **simulation** or **data** directly
  - Use data from a previous run or independent trigger

Conclusions

- In this work we proposed an **online-compatible Unsupervised Non-resonant Anomaly detection** method
- We use **autoencoders** as **anomaly detectors** and enforce the **decorrelation** between reconstruction losses using the **DisCo** loss
- Background estimation using the ABCD method
  - **Non-closure** for samples containing **new physics** events
  - **Significances** up to **4** for initial signal **contaminations** of **0.1%**
- **Online compatibility:** Signal enriched region saved together with prescaled sidebands for background estimation
- Available on [Phys. Rev. D 105, 055006]
- Scripts to run the model available on [github]





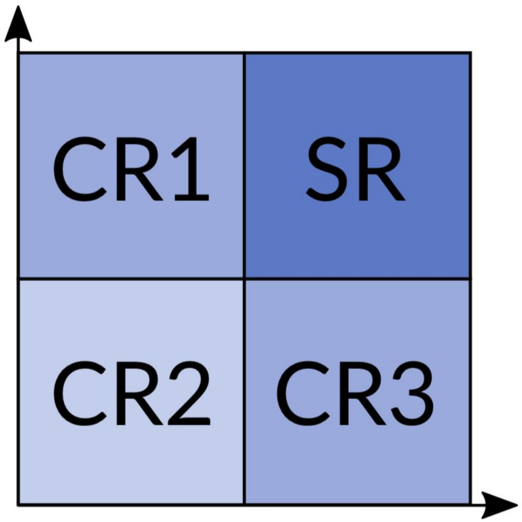
THANKS!

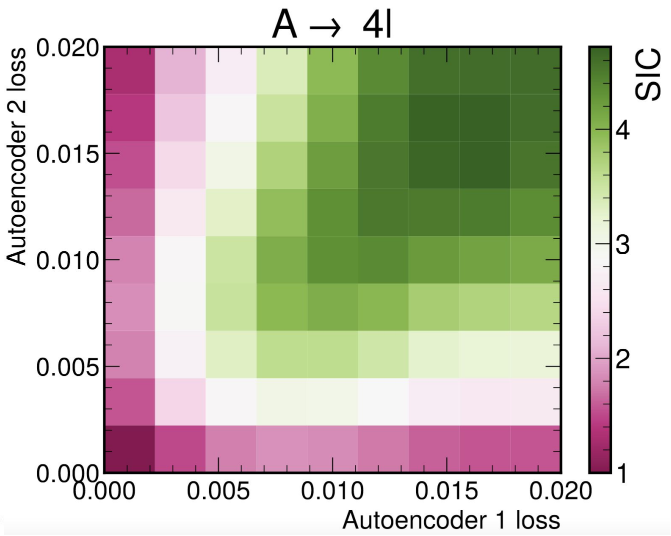
Any questions?

Autoencoders

ABCD method is a popular choice of data-driven background estimation

- Requires 2 **background-independent** distributions
- **Both** distributions should provide **signal sensitivity** to avoid contamination
- Background in the signal-enriched region is described by the other background-dominated regions

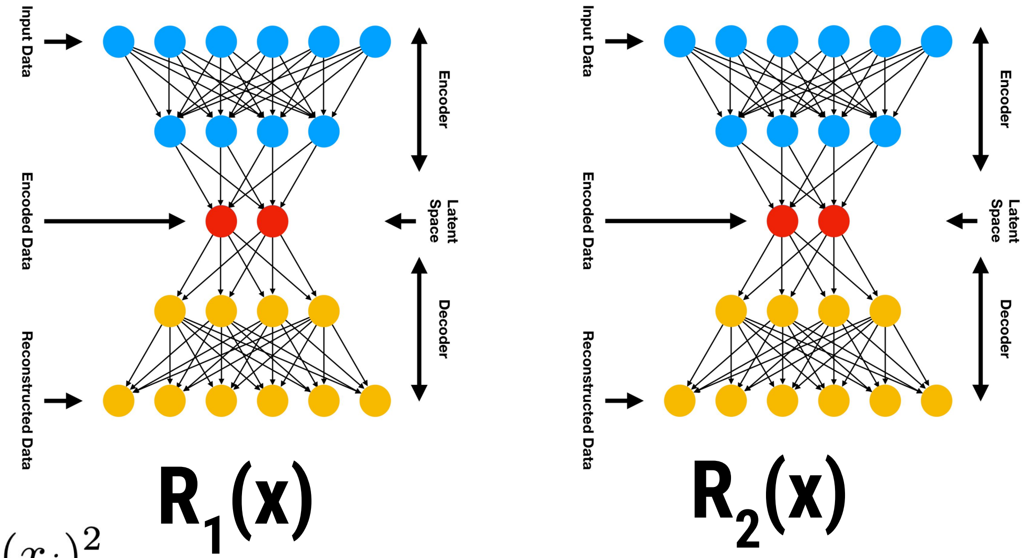




SR=CR1*CR3/CR2

Decorrelated autoencoders

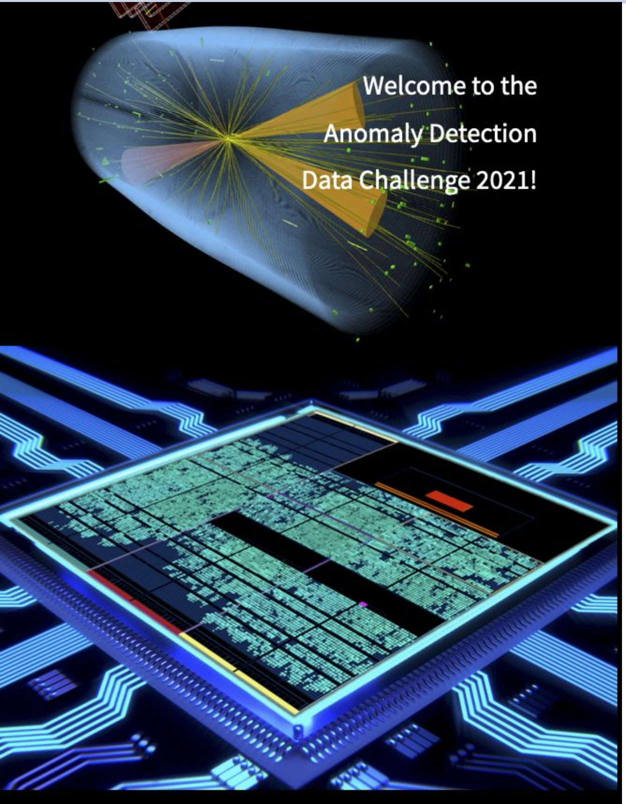
- Use the **reconstruction loss** of each autoencoder to define thresholds for the ABCD method
- Enforce the decorrelation between loss functions using the distance correlation (**DisCo**[1]) loss



$$L[f_1, f_2, g_1, g_2] = \sum_i R_1(x_i)^2 + \sum_i R_2(x_i)^2 + \lambda\,\mathrm{DisCo}^2[R_1(X), R_2(X)]$$

1: G. Kasieczka, B. Nachman, M. Schwartz, and D. Shih, Phys. Rev. D 103, 035021

ADC2021 dataset

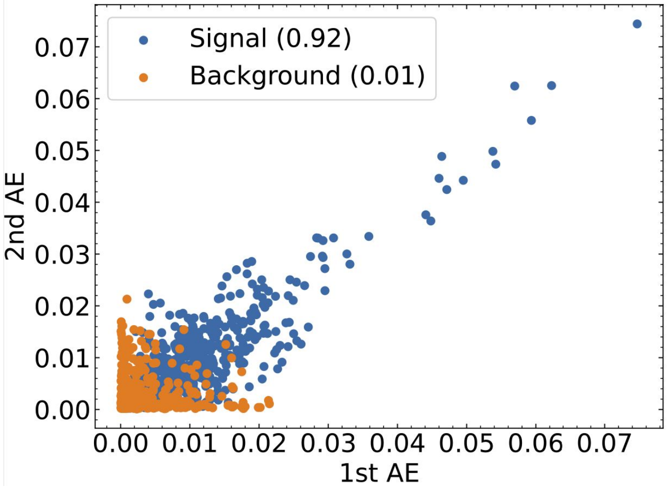


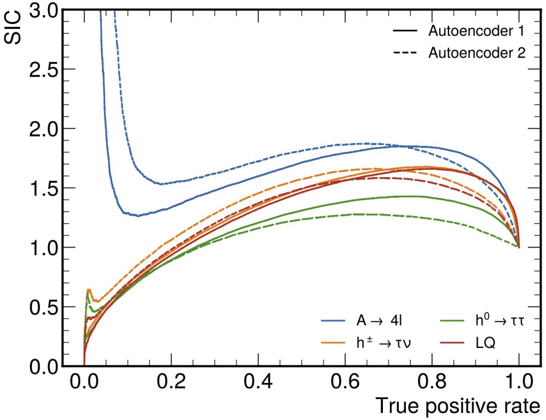
- Test the idea in a realistic setting: **anomaly detection at trigger level**
- **Goal:** Create algorithms that can trigger anomalous events that would otherwise be thrown away
- Dataset consists of a cocktail of **Standard Model processes** passing a **single lepton trigger**
- Momenta of leading **4 leptons** and **10 jets** are saved and used as inputs to the autoencoder
- **No invariant mass information used**
- **Train on background events** and **evaluate** over different new **physics scenarios** to test the performance

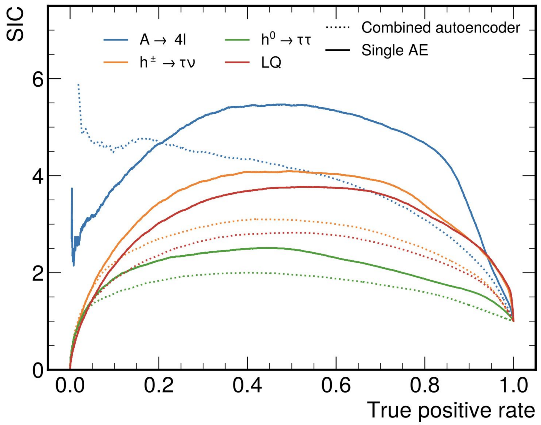
ADC2021 dataset

New physics benchmarks

- Neutral scalar boson (A), 50 GeV ⟶ 4 l
- Leptoquark (LQ), 80 GeV ⟶ b τ
- Scalar boson (h^0), 60 GeV ⟶ τ τ
- Charged scalar boson(h^+), 60 GeV ⟶ τ ν





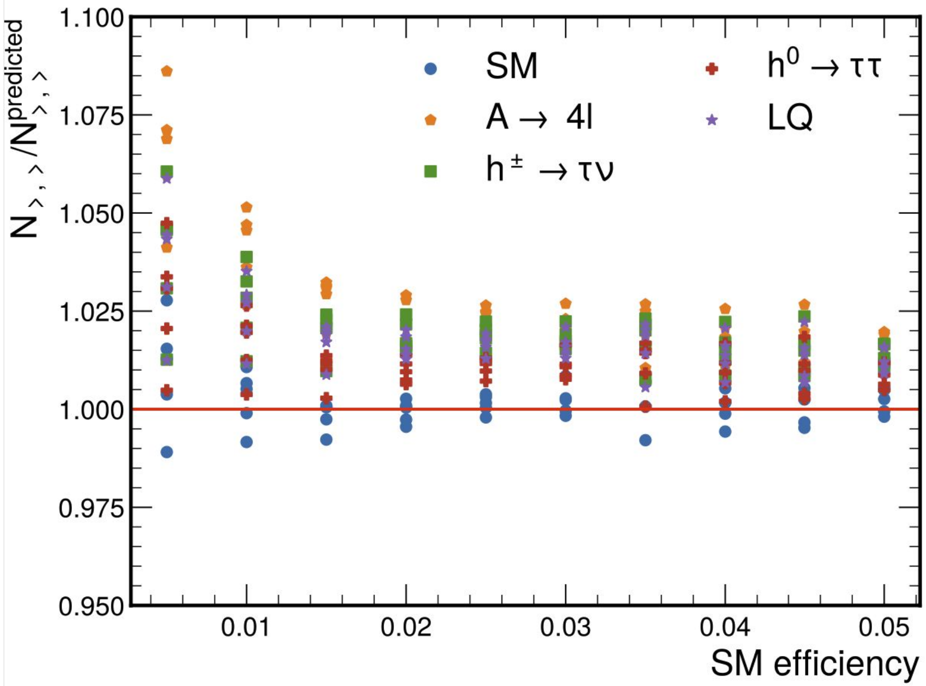


SIC = Significance improvement characteristic: **tpr/sqrt(fpr)**

ADC2021 dataset

Calculate the **background** in the signal enriched region using the **ABCD method**

- **Non-closure test:** compare **real number** of events with **predicted background**
- Different threshold choices resulting in different results
- Nevertheless, samples with **new physics scenarios** consistently having **more events** than predicted



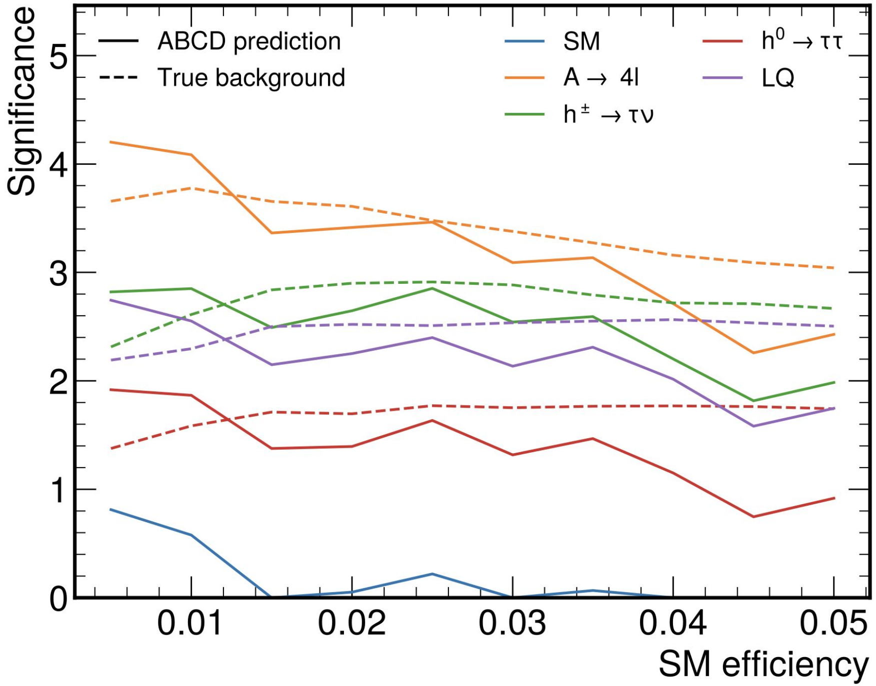
Spread in the y-axis represents the results when different selection thresholds are used

0.1% initial signal contamination

ADC2021 dataset

Quantify the difference in terms of **signal significance**

- **Less** than **1 sigma** for sample **without NP** and **1-4 for different NP scenarios**
- **Signal contamination** in the sidebands can lead to incorrect significances: Corrections to background prediction for limit setting



Significance = (N-B)/sqrt(N), if N>B and 0 otherwise

Online compatibility

Autoencoders on FPGAs for real-time, unsupervised new physics detection at 40 MHz at the Large Hadron Collider

VIII. CONCLUSIONS

We discussed how to extend new physics detection strategies at the LHC with autoencoders deployed in the L1T infrastructure of the experiments. In particular, we show how one could deploy a deep neural network (DNN) or convolutional neural network (CNN) AE on a field-programmable gate array (FPGA) using the `hls4ml` library, within a $\mathcal{O}(1)\mu$s latency and with small resource utilization once the model is quantized and pruned. We show that one can retain accuracy by compressing the model at training time. Moreover, we discuss different

- Our model uses only fully connected layers: **demonstrated** to satisfy trigger budget constraints when running on FPGAs after pruning and compression
- **First complete** online compatible anomaly detection protocol to be proposed

Distance correlation

Decorrelation function

- Given the output space of 2 neural networks F and G, the distance covariance is defined as

$$\mathrm{dCov}^2[f,g] = \Big\langle |f-f'| \times |g-g'| \Big\rangle + \Big\langle |f-f'| \Big\rangle \times \Big\langle |g-g'| \Big\rangle - 2\Big\langle |f-f'| \times |g-g''| \Big\rangle$$

- Where f and f' are sampled from F and g, g', and g" are sampled from G
- The correlation distance is then defined as

$$\mathrm{dCorr}^2[f,g] = \frac{\mathrm{dCov}^2[f,g]}{\mathrm{dCov}[f,f]\,\mathrm{dCov}[g,g]}\,.$$