Self-supervised searches for new physics

Barry Dillon

Universität Heidelberg
Zukunft. Seit 1386.

KIAS - AI and Quantum Information Applications in Fundamental Physics
Feb 2023

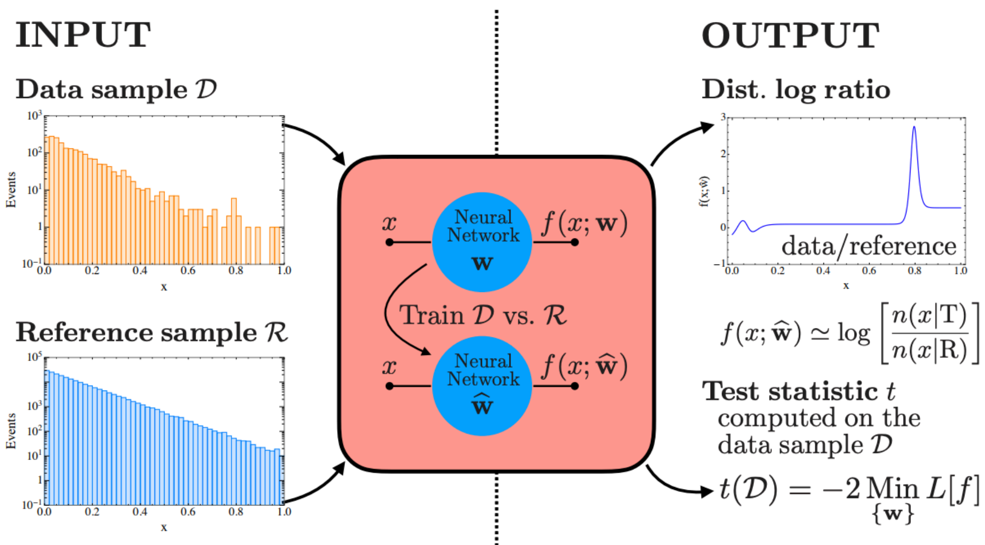
Overview

- Anomaly detection
- Self-supervision
- Self-supervision and AutoEncoders
- Conclusions

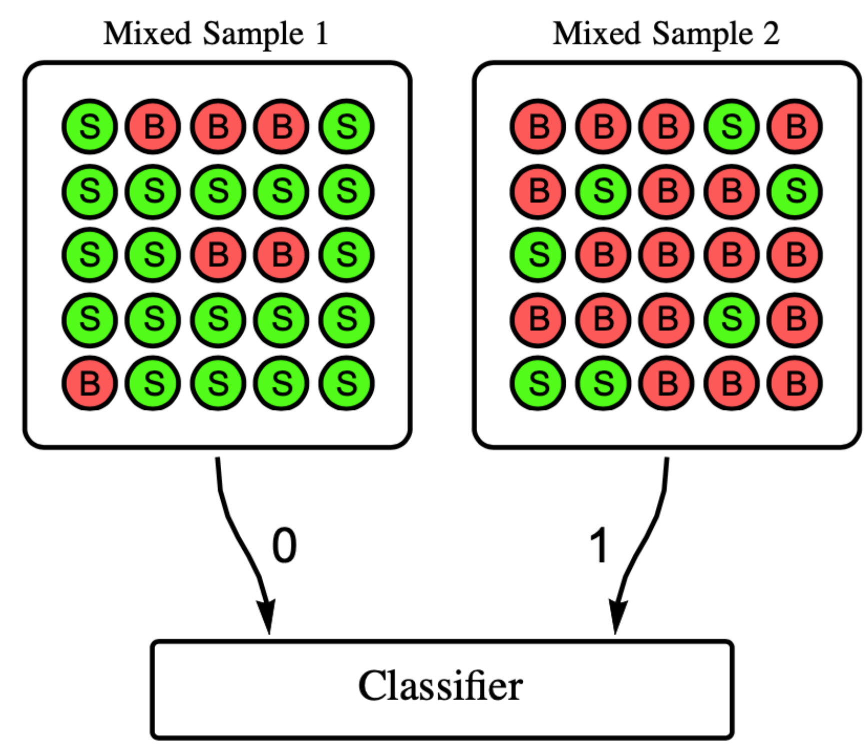
Anomaly-detection

ML-based anomaly detection

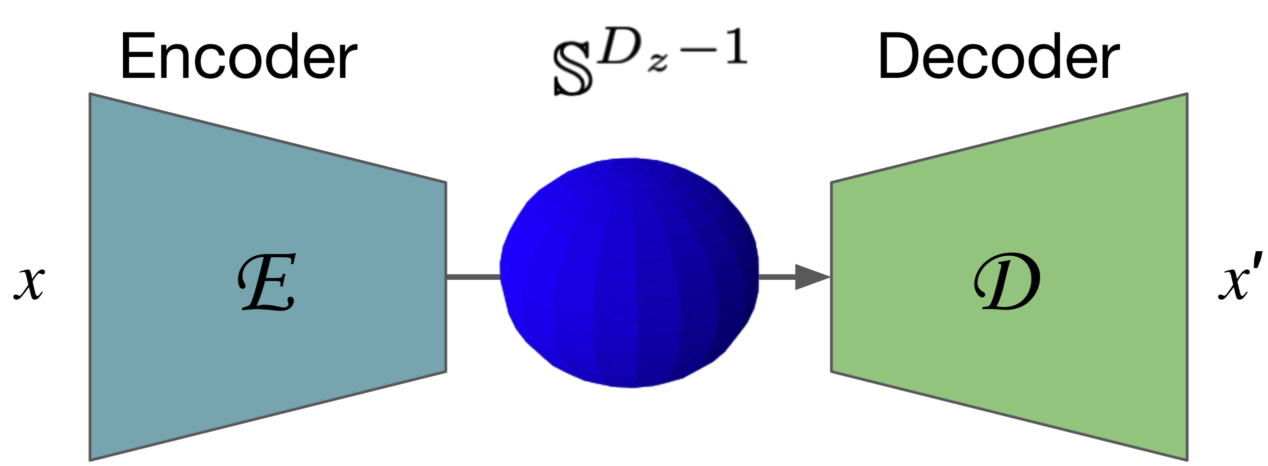
1 - Simulation vs experiment



2 - Classification Without Labels

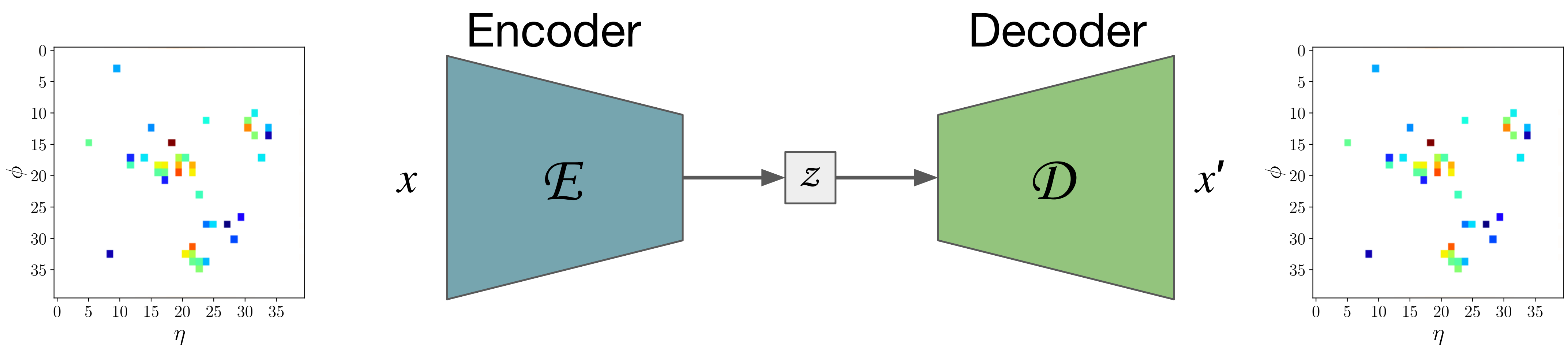


3 - AutoEncoders



AutoEncoder networks

['QCD or What?' Heimel et al]
['Searching for new physics with deep AEs' Farina et al]



- Trained to reconstruct the data they are trained on
- Optimised on background-only/dominant data
- Unsupervised ⟶ model-agnostic, no labels
- Reconstruction loss: $\mathcal{L} = ||x - x'||^2$
- More anomalous ⇒ data the network has seen least ⇒ larger reconstruction loss
- AEs give us an observable to measure OOD-ness

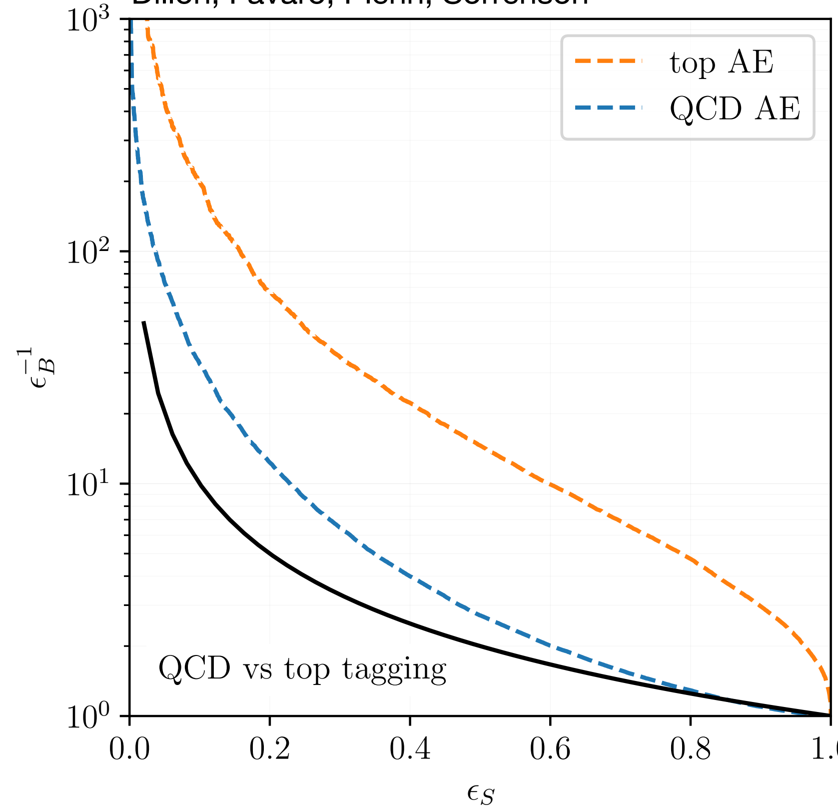
AutoEncoder networks - the problems

They don’t robustly identify anomalous jets.

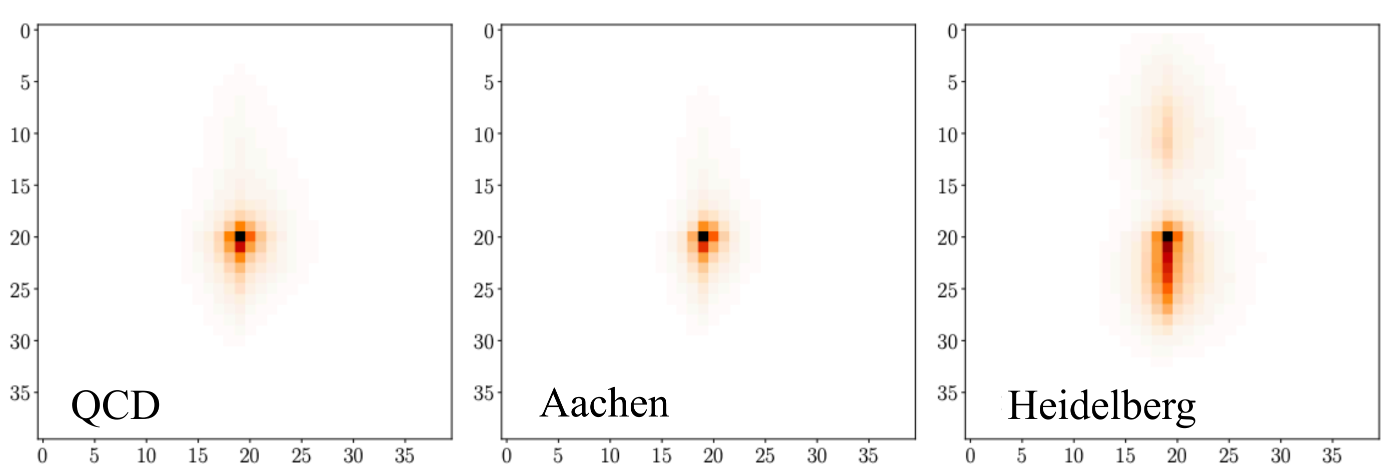
They do robustly identify complex jets.

e.g anomalous top/QCD jets

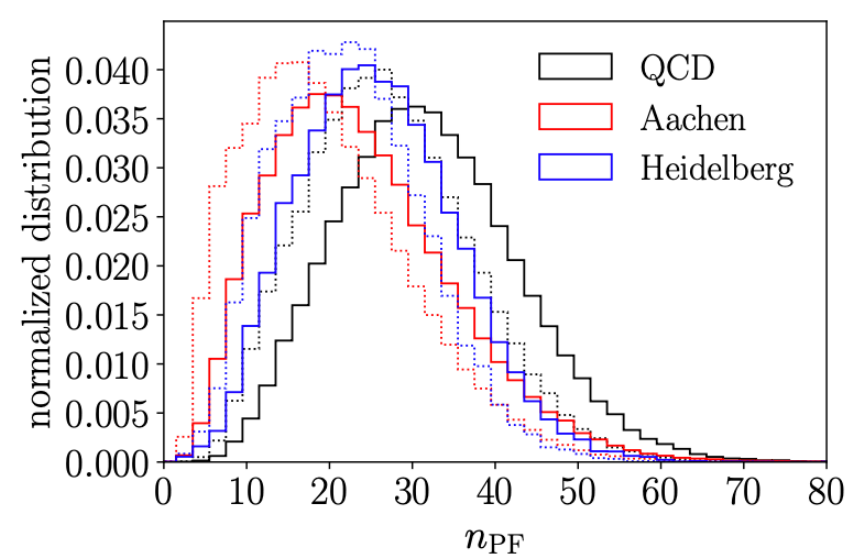
‘A normalised autoencoder for LHC triggers’
Dillon, Favaro, Plehn, Sorrenson



e.g. semi-visible jets - very little substructure [Pythia’s HiddenValley model]



‘What’s anomalous in LHC jets?’
Buss, Dillon, Finke, Krämer, …

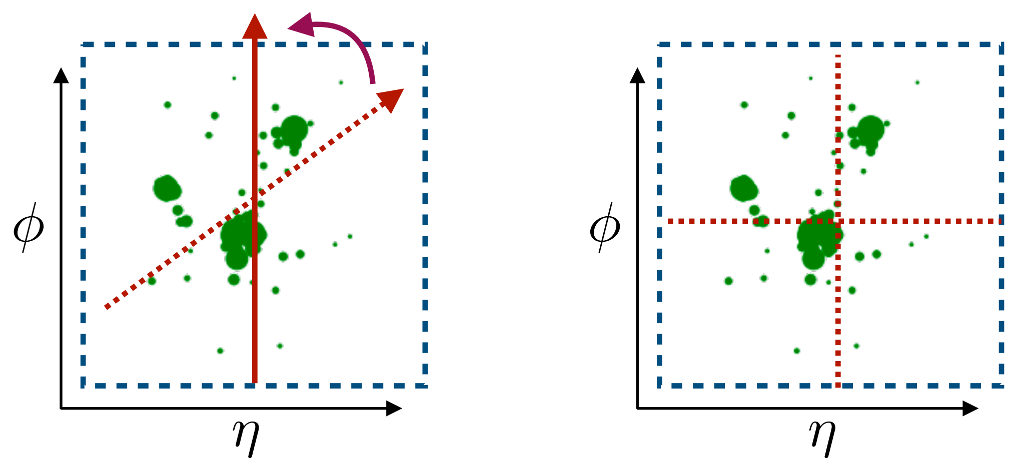


AutoEncoder networks - the problems

Not invariant to symmetries in jet physics.

AE can't reconstruct something the latent space is invariant to…

Preprocessing is necessary, but approximate.

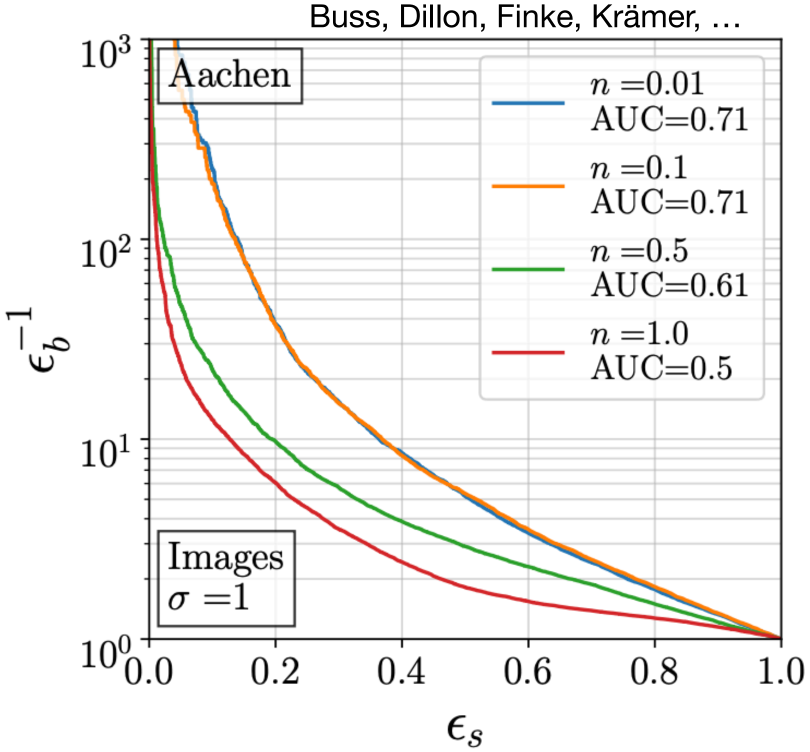


Very sensitive to the choice of representation.

e.g. under re-mapping of p_T's, $\quad p_T \rightarrow p_T^n$

the results vary a lot.

'What's anomalous in LHC jets?'
Buss, Dillon, Finke, Krämer, …



Density-based anomaly detection

Reconstruction is a very vague way to define anomalous (OOD-ness)

More accurately: anomalous events/jets are in low density regions of the feature space

Machine-learned density estimation:

1 - some parameterisation of the density $p_{\text{data}}(\vec{x})$

2 - a scheme to minimise $-\log p_{\text{data}}(\vec{x})$ wrt to the parameters

Also works well in high-dimensions! → Normalised AutoEncoder

['A normalised autoencoder for LHC triggers' Dillon et al]

So, how do we define the representation (i.e. feature space) of the data???

Self-supervision

Self-supervised learning

- Extract useful information from unlabelled data
- Model creates its own supervision by creating tasks from the data
- Allows the model to create rich representations from data for downstream tasks

Supervised	Unsupervised	Self-supervised
uses 'truth labels' from simulation	no labels at all are used	uses 'pseudo-labels' derived from the data

→ reframe the definition of observables as an self-supervised optimisation task

What do we want from the observables?

- Invariance to symmetries
- Discriminative power

JetCLR : contrastive learning of jet representations

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

Dataset: mixture of QCD and top jets, again

From the dataset of jets $\{x_i\}$ we define:

'pseudo labels'

- positive pairs: $\{(x_i, x'_i)\}$ where x'_i is an augmented version of x_i
- negative pairs: $\{(x_i, x_j)\} \cup \{(x_i, x'_j)\}$ for $i \neq j$

Optimise a network to map $f(x_i) = z_i, \quad f : \mathcal{J} \to \mathcal{R}$, optimising for:

- alignment: positive pairs are close together in $\mathcal{R}$

  $\longrightarrow$ forces invariance to augmentations

- uniformity: negative pairs are far apart in $\mathcal{R}$

  $\longrightarrow$ forces discriminative power in representation space

JetCLR : contrastive learning of jet representations

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

Similarity measure in $\mathcal{R}$: $s(z_i, z_j) = \dfrac{z_i \cdot z_j}{|z_i||z_j|}$ ⟶ defined on a unit hypersphere

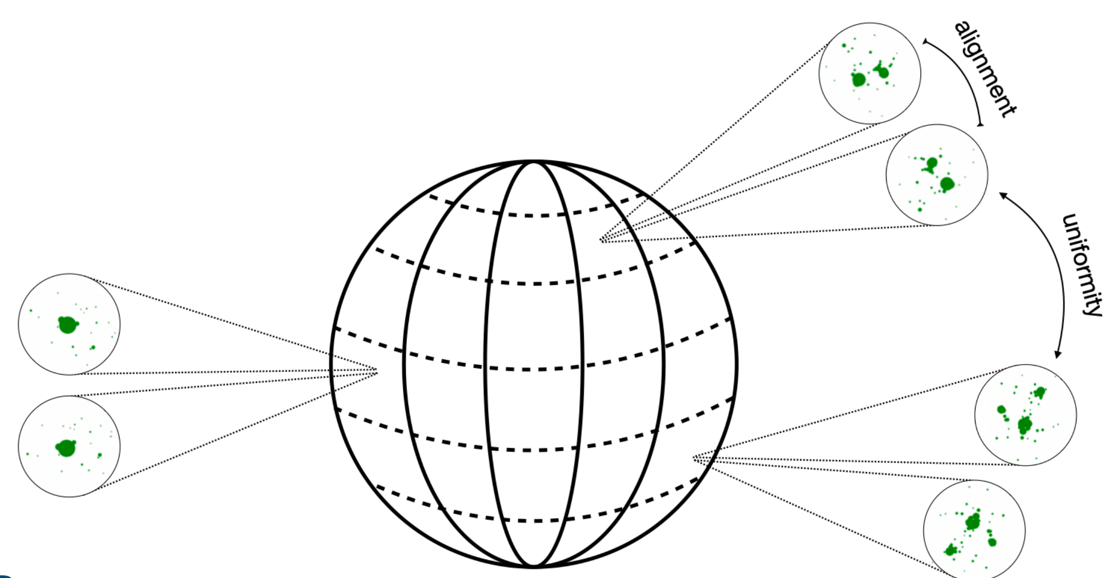
important to constrain uniformity

Contrastive loss: $$\mathcal{L}_i = -\log \frac{\exp\left(s(z_i, z_i')/\tau\right)}{\sum_{x \in \text{batch}} \mathbb{I}_{i \neq j} \left(\exp\left(s(z_i, z_j')/\tau\right) + \exp\left(s(z_i, z_j')/\tau\right)\right)}$$

Numerator: positive pairs & alignment

Denominator: negative pairs & uniformity

Can be completely data-driven, with augmentations applied to experimental data.



JetCLR : contrastive learning of jet representations

‘Symmetries, safety, and self-supervision’, Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

Optimising the network:

1. Sample a batch of jets
2. Create an augmented batch of jets
3. Forward-pass through the network
4. Compute loss and update weights

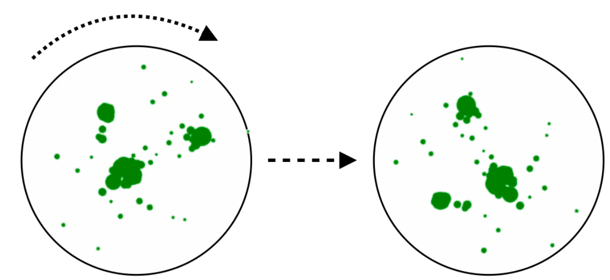
JetCLR : contrastive learning of jet representations

‘Symmetries, safety, and self-supervision’, Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

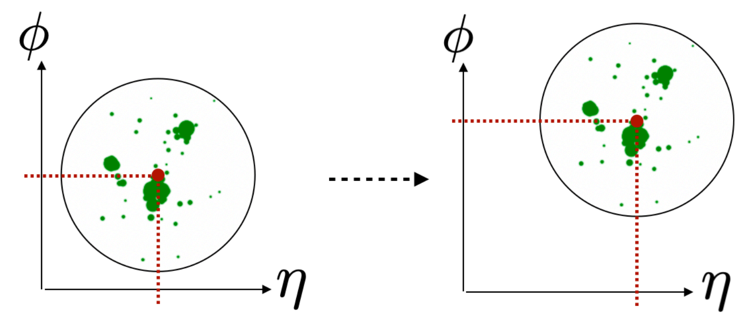
Optimising the network:

1. Sample a batch of jets
2. Create an augmented batch of jets
3. Forward-pass through the network
4. Compute loss and update weights

2.A Rotations



2.B Translations



2.C Collinear splittings

$$p_{T,a} + p_{T,b} = p_T \qquad \eta_a = \eta_b = \eta$$
$$\phi_a = \phi_b = \phi$$

2.D IR smearing

$$\eta' \sim \mathcal{N}\left(\eta, \frac{\Lambda_{\text{soft}}}{p_T}\right)$$
$$\phi' \sim \mathcal{N}\left(\phi, \frac{\Lambda_{\text{soft}}}{p_T}\right)$$

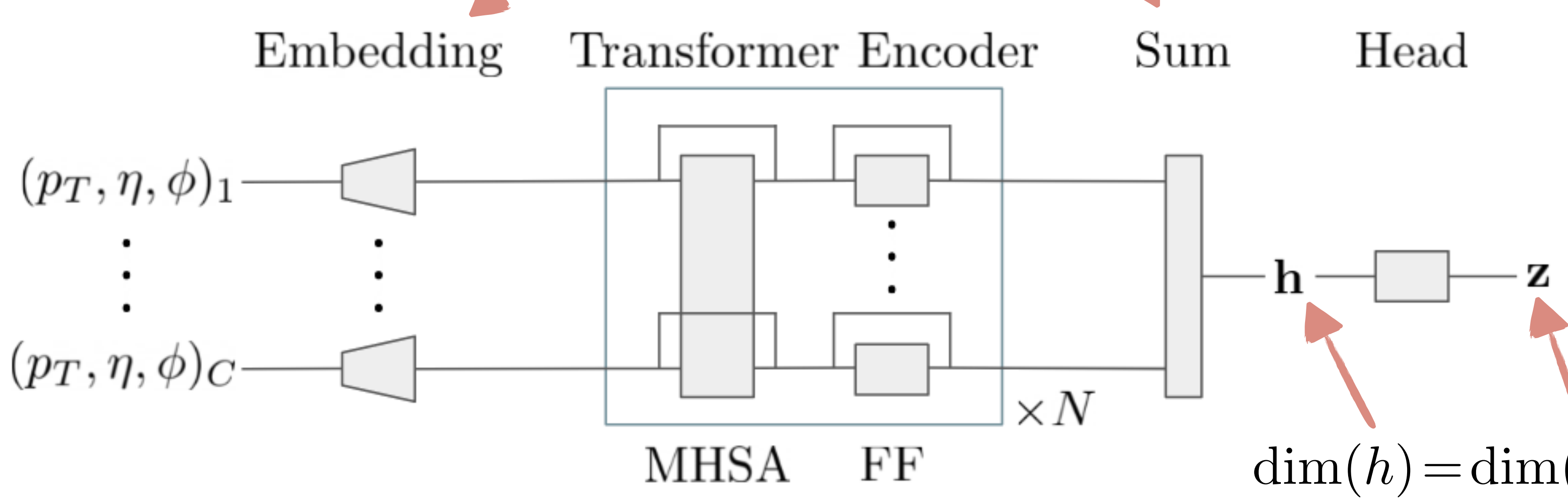
JetCLR : contrastive learning of jet representations

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

Optimising the network:

1. Sample a batch of jets
2. Create an augmented batch of jets
3. Forward-pass through the network
4. Compute loss and update weights

Permutation-invariant deep-sets style architecture



$\dim(h) = \dim(z) = 1000$

JetCLR : representation power

‘Symmetries, safety, and self-supervision’, Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

Measure performance of the JetCLR representations using a Linear Classifier Test

Compare performance with three other widely-used representations

1. 4-vector inputs

   80D rep, no invariances whatsoever

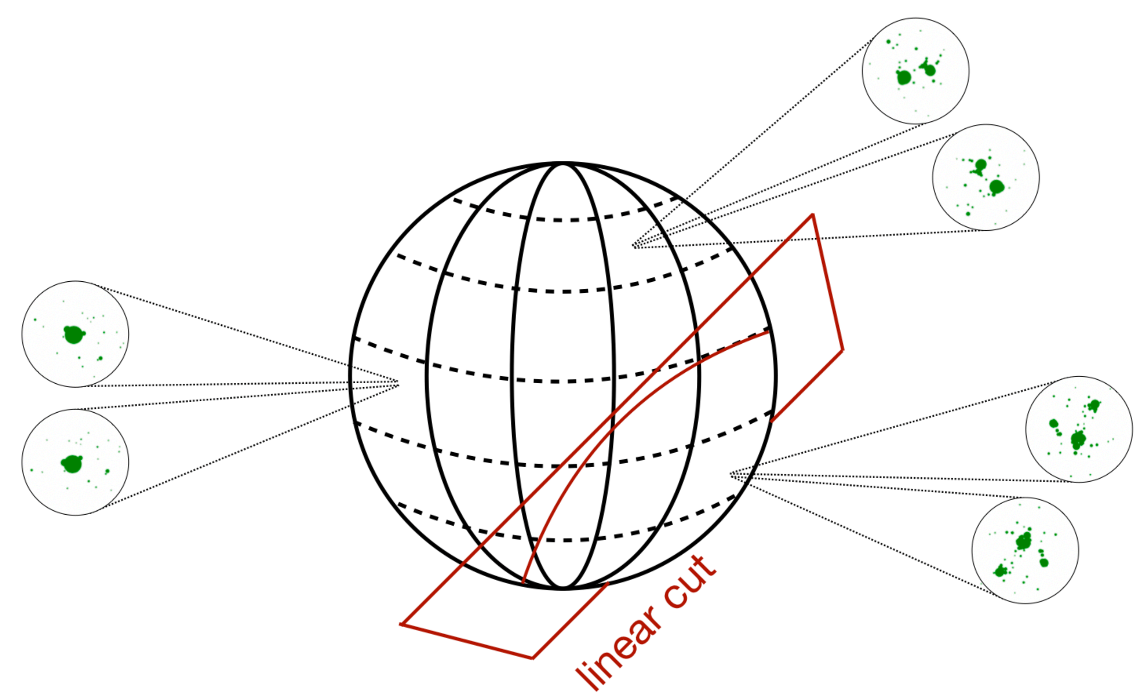
2. Jet images

   1600D rep, approx invariance to rotations & translations, IRC safe

3. Energy Flow Polynomials

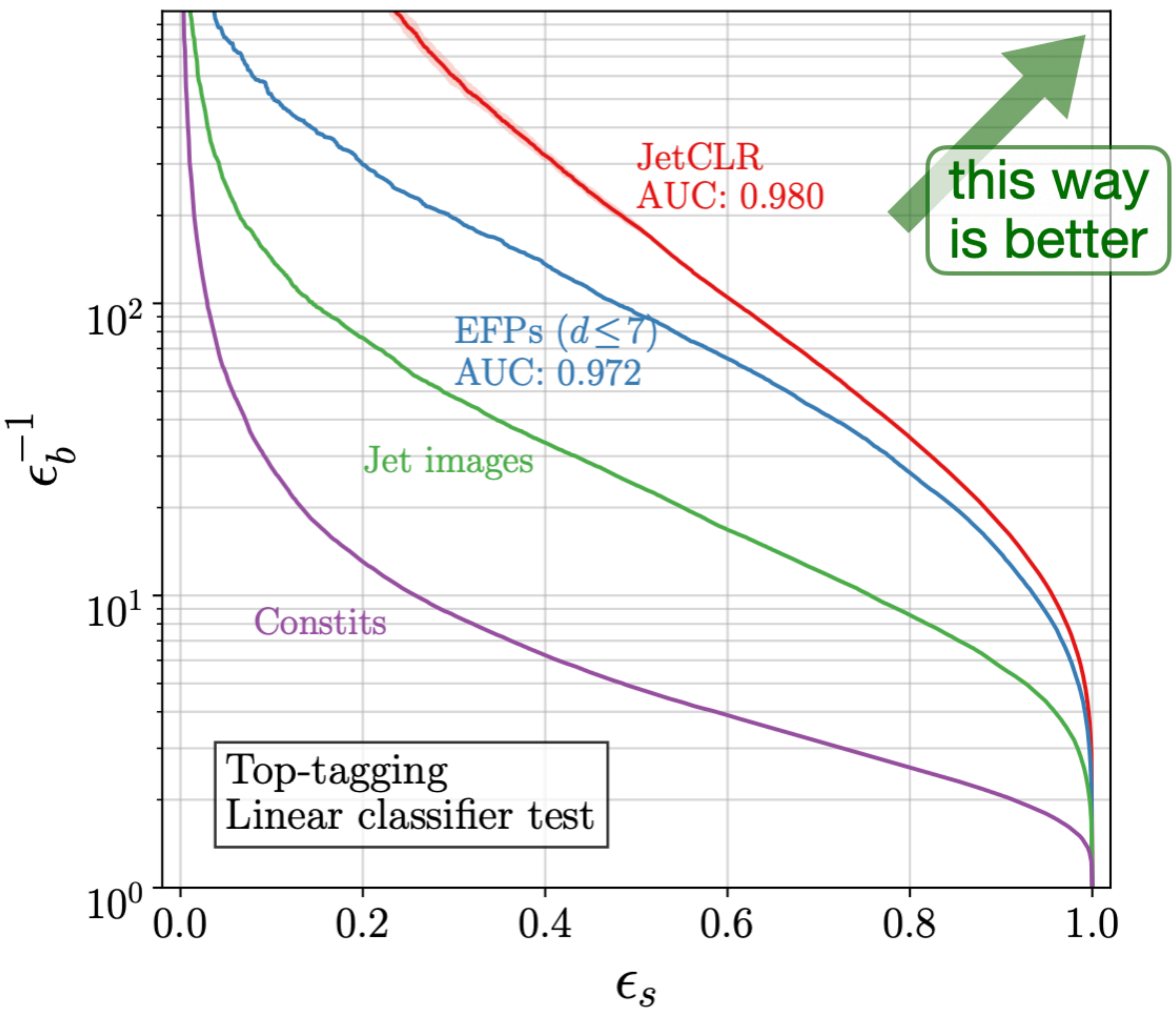
   1000D rep, exact invariance to rotations & translations, and IRC safe

   ‘Energy Flow Polynomials’, Thaler et al



JetCLR : representation power

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel



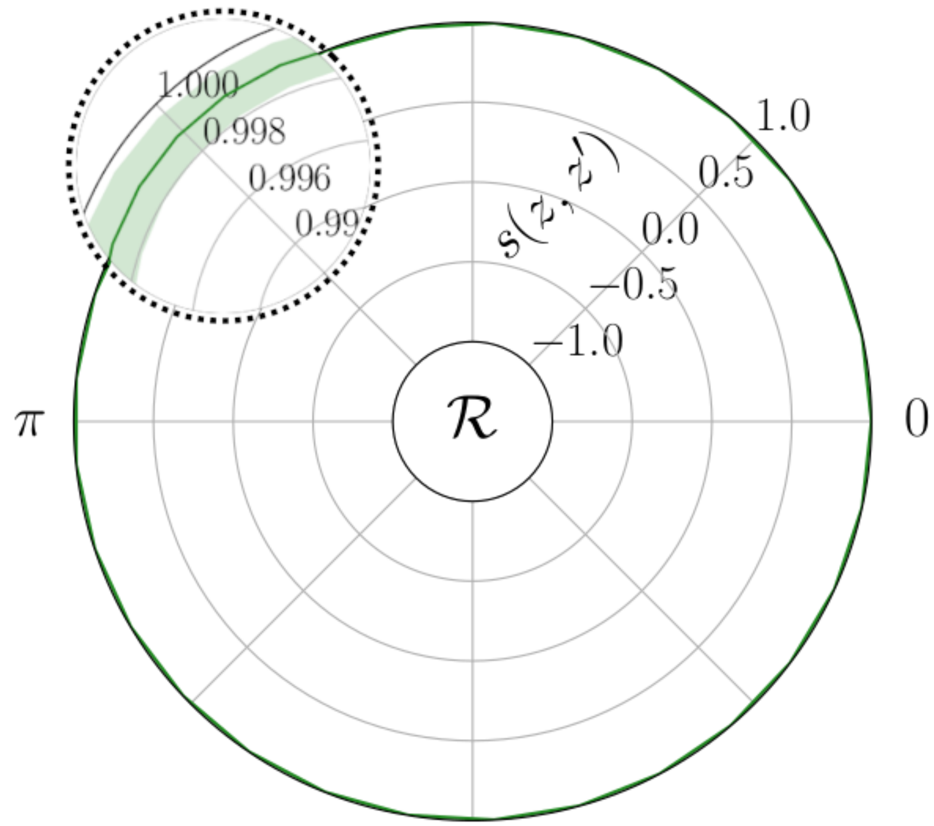
Augmentation	$\epsilon_b^{-1}(\epsilon_s = 0.5)$	AUC
none	15	0.905
translations	19	0.916
rotations	21	0.930
soft+collinear	89	0.970
all combined (default)	181	0.980

Results are very insensitive to S/B as well, implies that JetCLR learns some very general features of jets.

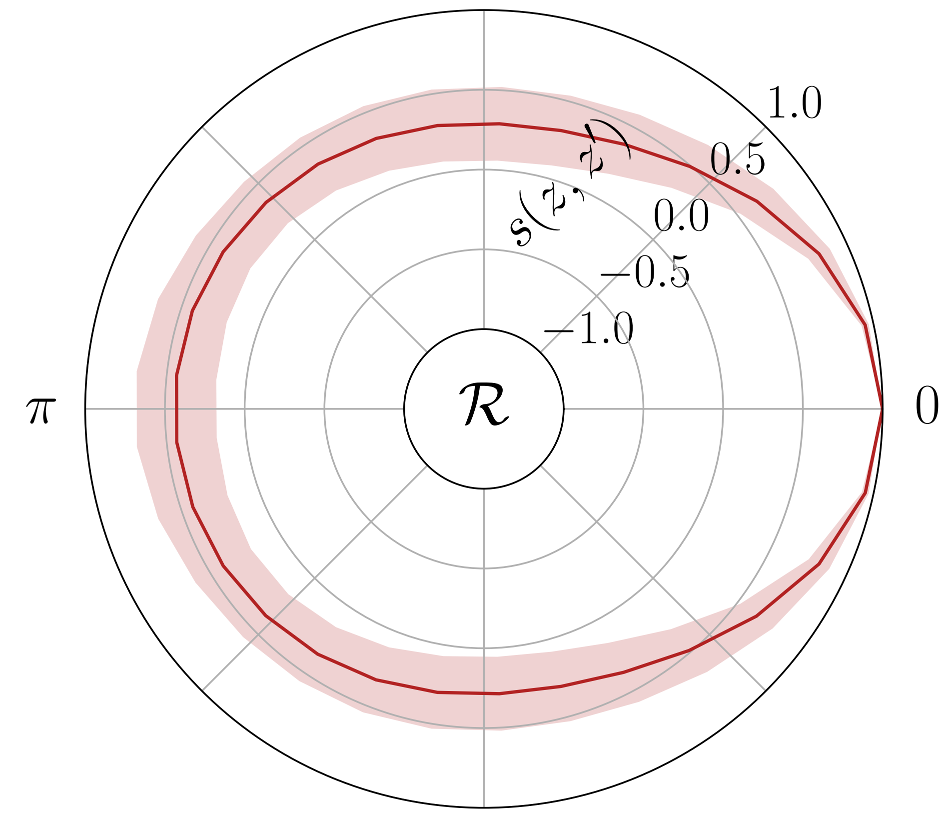
JetCLR : invariance to rotations

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

with rotation invariance



without rotation invariance

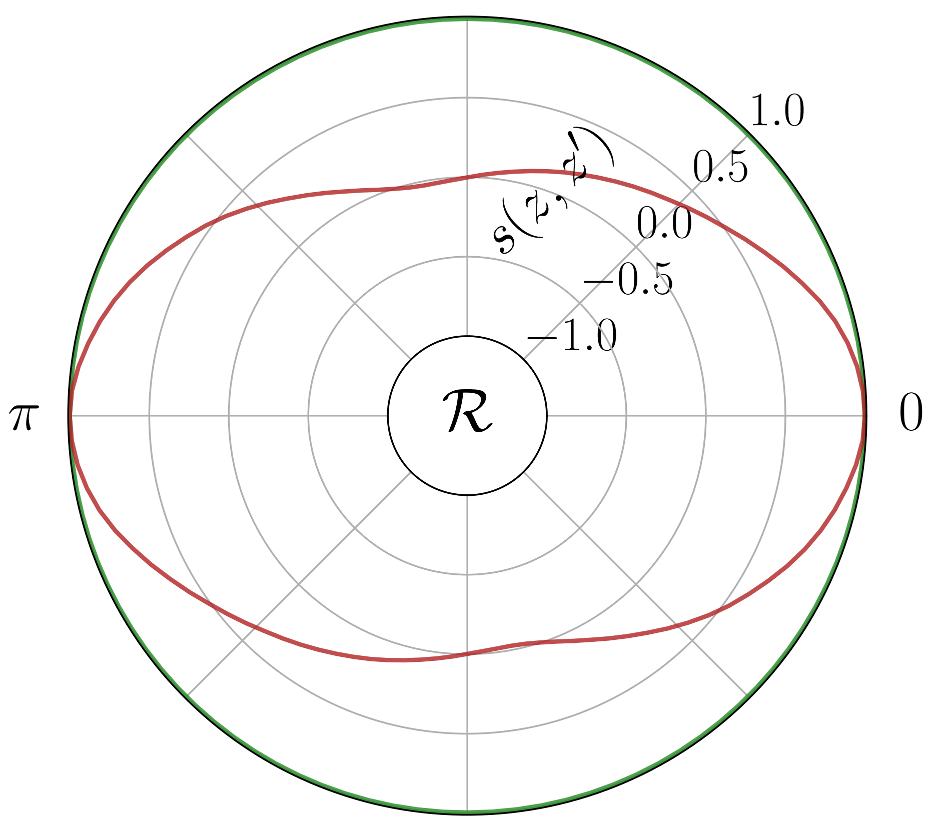


$$s(z,z') = \frac{z \cdot z'}{|z||z'|}, \quad z = f\,(x)\,, \;\; z' = f\,(R(\theta)x)$$

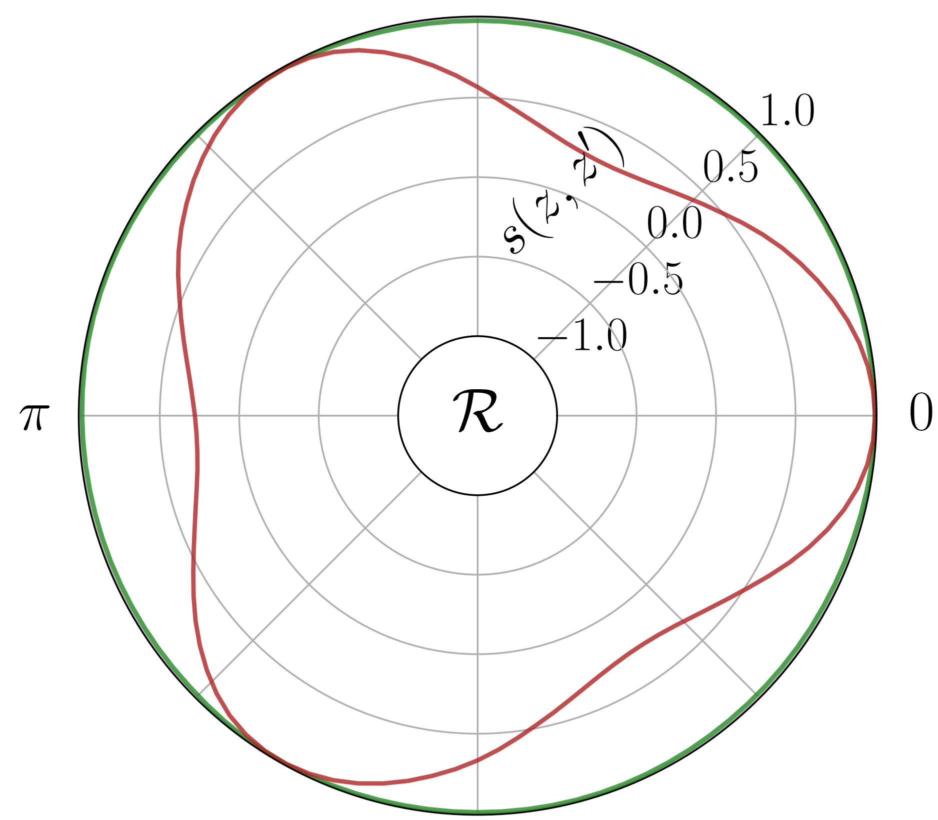
JetCLR : invariance to rotations

'Symmetries, safety, and self-supervision', Dillon, Kasieczka, Olischläger, Plehn, Sorrenson, Vogel

two constituent jet



three constituent jet



$$s(z, z') = \frac{z \cdot z'}{|z||z'|}, \quad z = f\,(x)\,,\;\; z' = f\,(R(\theta)x)$$

Self-supervision & AutoEncoders

AnomalyCLR : contrastive learning for anomalies

'Anomalies, representations, and self-supervision', Dillon, Favaro, Fieden, Modak, Plehn

What if the dataset only contains background?

$$\mathcal{L}_{\text{CLR}} = -\log \frac{\exp(s(z_i, z_i'))}{\sum_{x \in \text{batch}} \mathbb{1}_{i \neq j} \left(\exp(s(z_i, z_j)) + \exp(s(z_i, z_j')) \right)}$$

no guarantee to learn features sensitive to new physics…

Solution??

$$\mathcal{L}_{\text{AnomCLR}} = -\log \frac{\exp(s(z_i, z_i') - s(z_i, z_i^*))}{\sum_{x \in \text{batch}} \mathbb{1}_{i \neq j} \left(\exp(s(z_i, z_j)) + \exp(s(z_i, z_j')) \right)}$$

$$\mathcal{L}^{+}_{\text{AnomCLR}} = s(z_i, z_i^*) - s(z_i, z_i')$$

$z^* \longrightarrow$ anomaly-augmented collider data

Again, can be completely data-driven, with augmentations applied to experimental data.

AnomalyCLR : contrastive learning for anomalies

'Anomalies, representations, and self-supervision', Dillon, Favaro, Fieden, Modak, Plehn

Dataset: mixture of SM events

$W \to l\nu$ (59.2%)

$Z \to ll$ (6.7%)

$t\bar{t}$ production (0.3%)

QCD multijet (33.8 %)

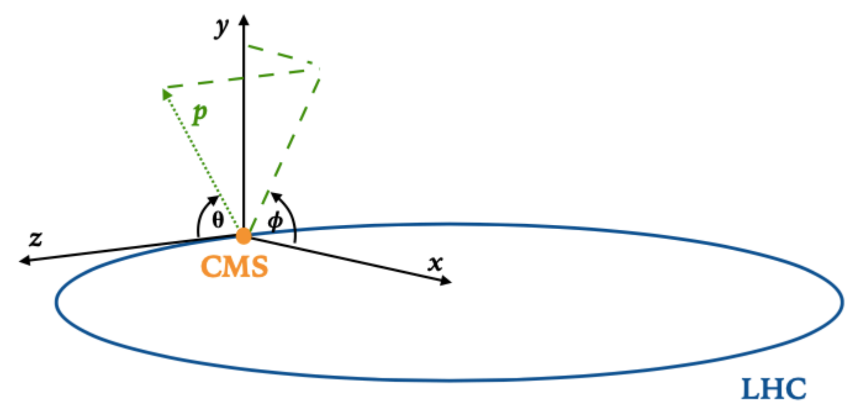
BSM benchmarks

$A \to 4l$

$LQ \to b\nu$

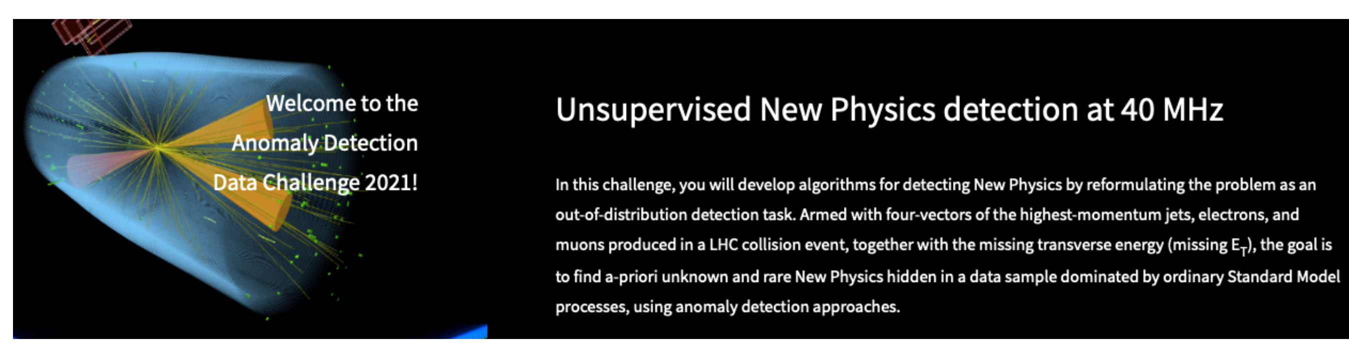
$h_0 \to \tau\tau$

$h_+ \to \tau\nu$



The events are represented as (19, 3) entries

- 19 particles: MET, 4 electrons, 4 muons, and 10 jets
- 3 observables: p_T, η, ϕ
- $|\eta| < [3,\ 2.1,\ 4]$ for $e,\ \mu,\ j$ respectively

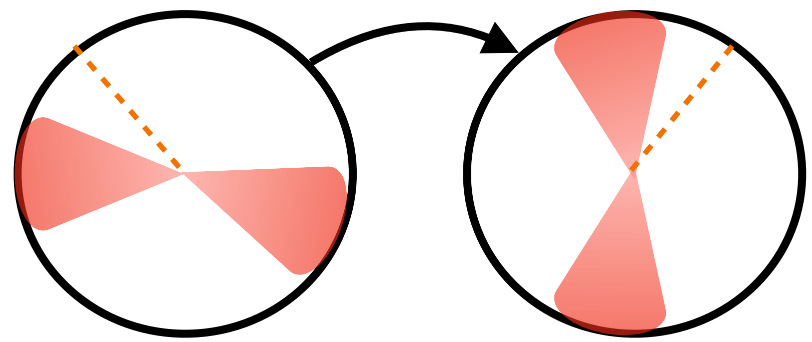


AnomalyCLR : contrastive learning for anomalies

‘Anomalies, representations, and self-supervision’, Dillon, Favaro, Fieden, Modak, Plehn

Physical augmentations:

- azimuthal rotations
- η, ϕ smearing
- energy smearing



$$\eta' \sim \mathcal{N}\Big(\eta, \sigma(p_T)\Big)$$

$$\phi' \sim \mathcal{N}\Big(\phi, \sigma(p_T)\Big)$$

$$p_T \sim \mathcal{N}(p_T, f(p_T)), \qquad f(p_T) = \sqrt{0.052 p_T^2 + 1.502 p_T^2}$$

Anomalous augmentations:

- multiplicity shifts:
  - add a random number of particles, update MET
  - split existing particles, keeping total p_T and MET fixed
- p_T and MET shifts

AnomalyCLR : contrastive learning for anomalies

‘Anomalies, representations, and self-supervision’, Dillon, Favaro, Fieden, Modak, Plehn

Transformer details

- 4 transformer encoder layers
- Model dimension: 200
- Data: add a one-hot-encoded particle ID to inputs

AutoEncoder details

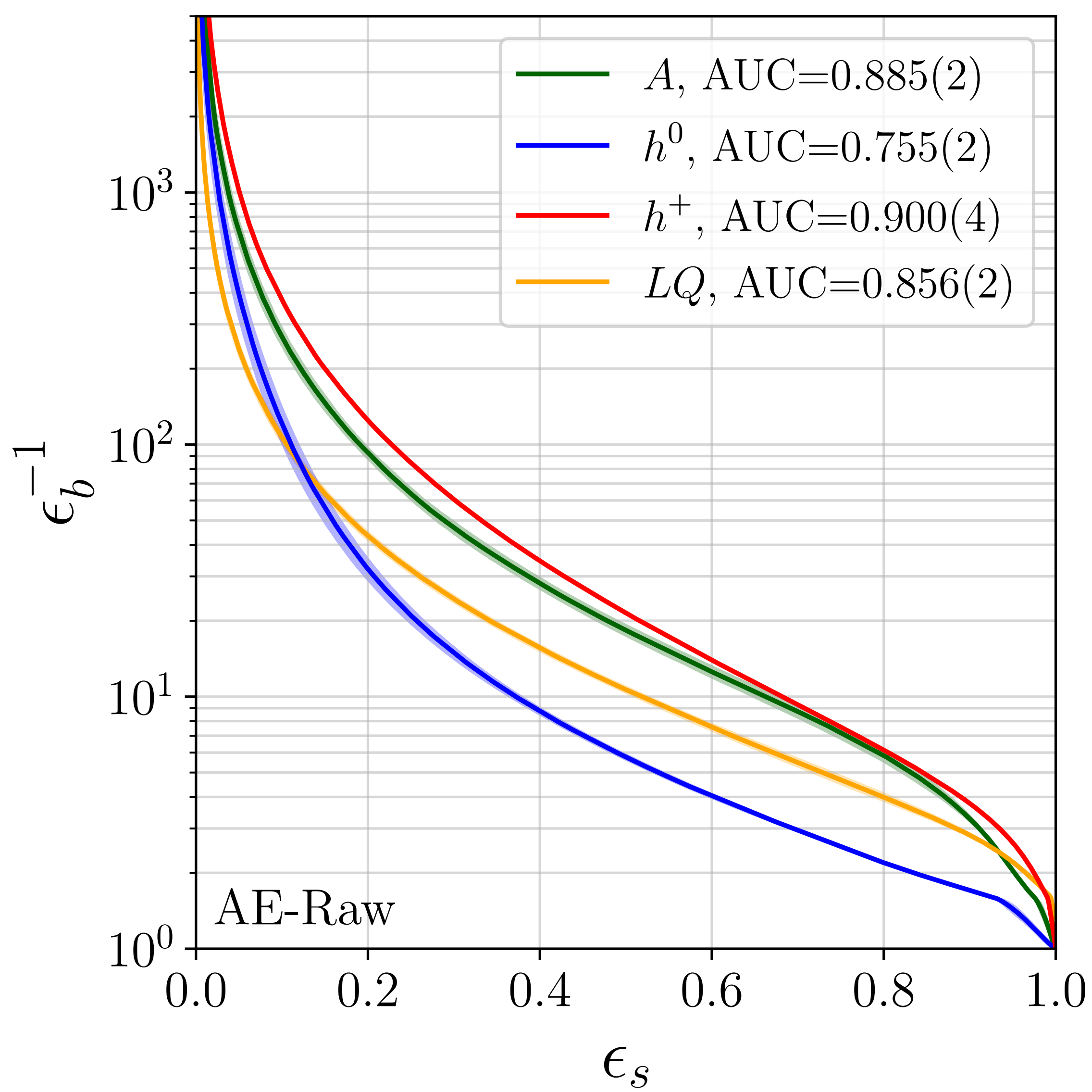
- 5 hidden layers - 256, 128, 64,32,16
- Latent space dimension: 5

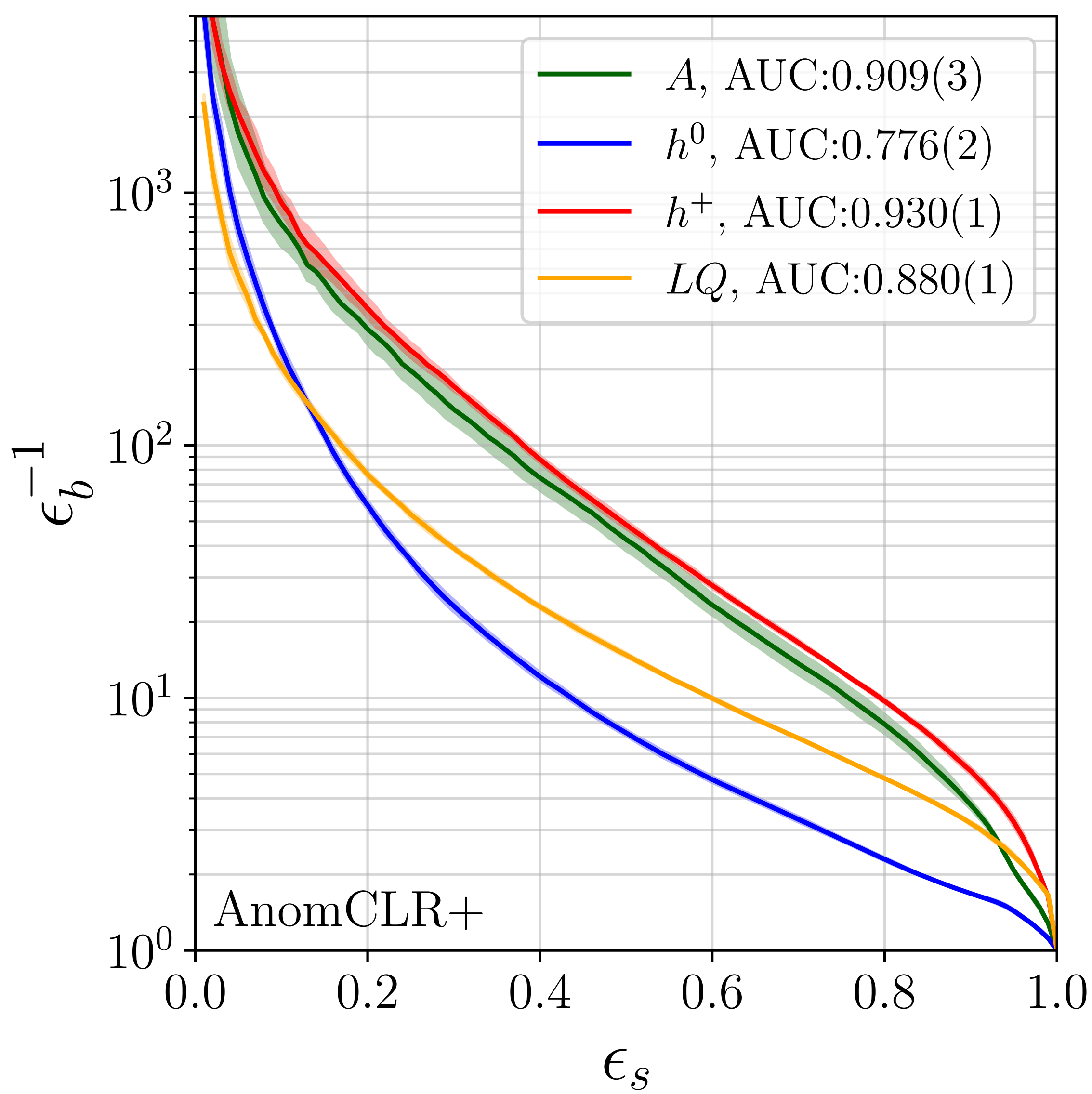
Raw data preprocessing

- Minor preprocessing to make numbers O(1)
- p_T’s divided by average value of the dataset
- η and ϕ values are re-scaled to be between -1 and +1

AnomalyCLR : contrastive learning for anomalies

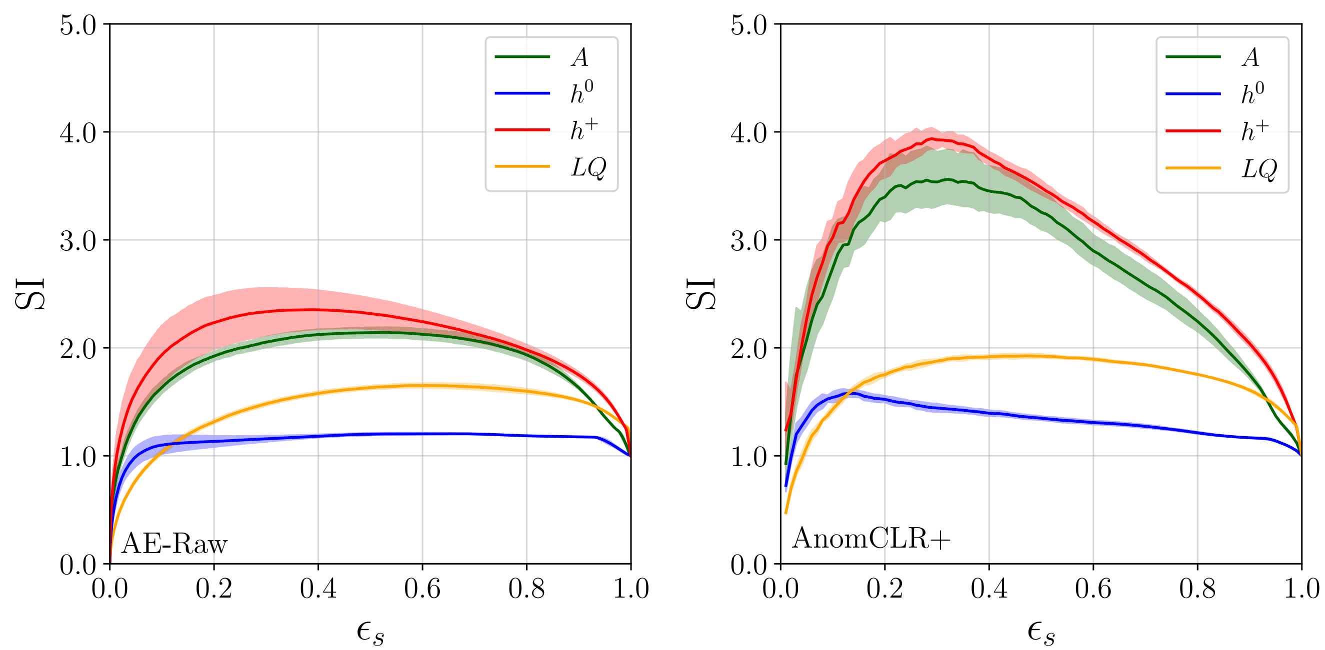
‘Anomalies, representations, and self-supervision’, Dillon, Favaro, Fieden, Modak, Plehn

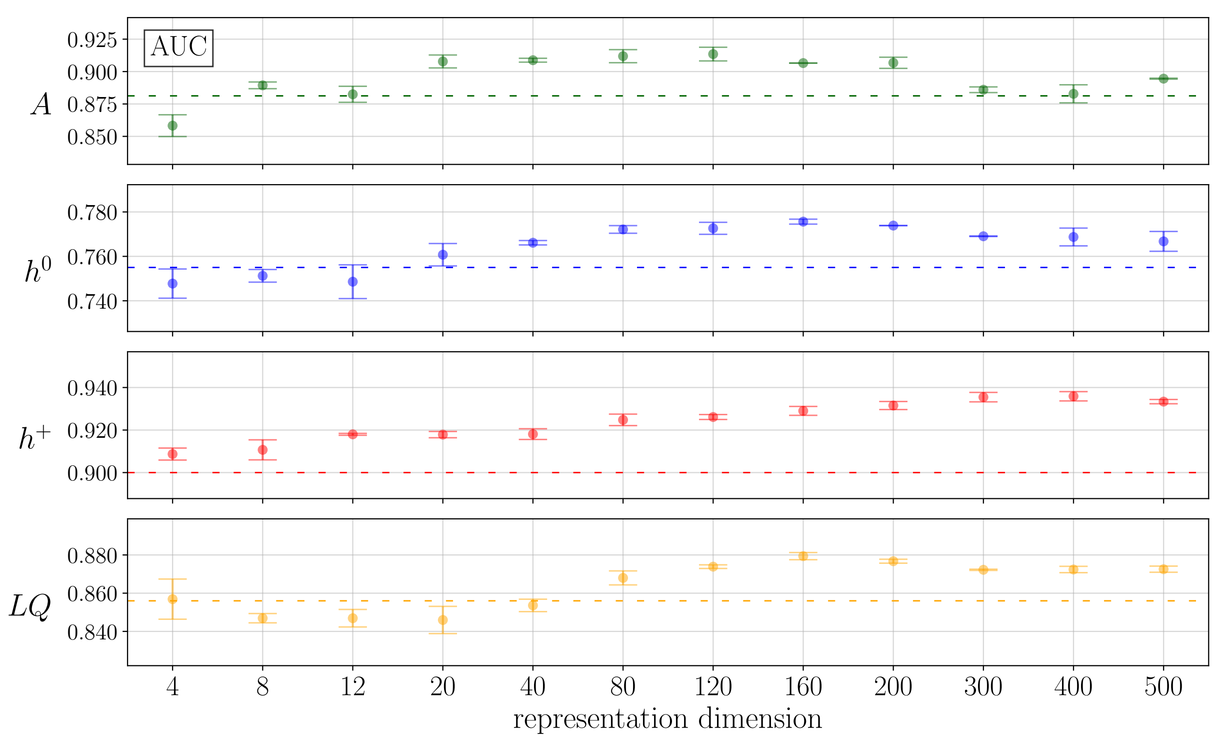


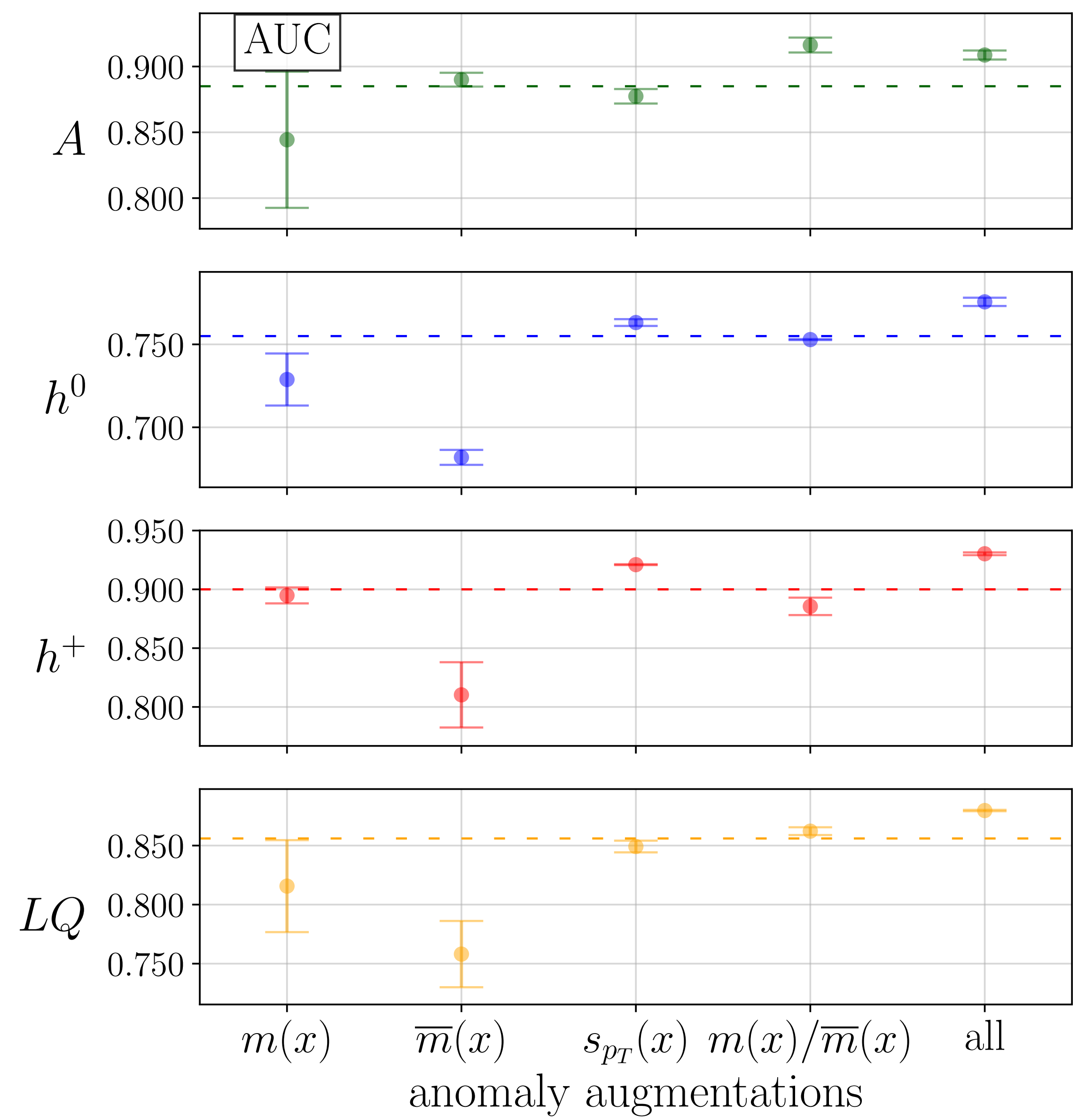


AnomalyCLR : contrastive learning for anomalies

‘Anomalies, representations, and self-supervision’, Dillon, Favaro, Fieden, Modak, Plehn







AnomalyCLR on jets

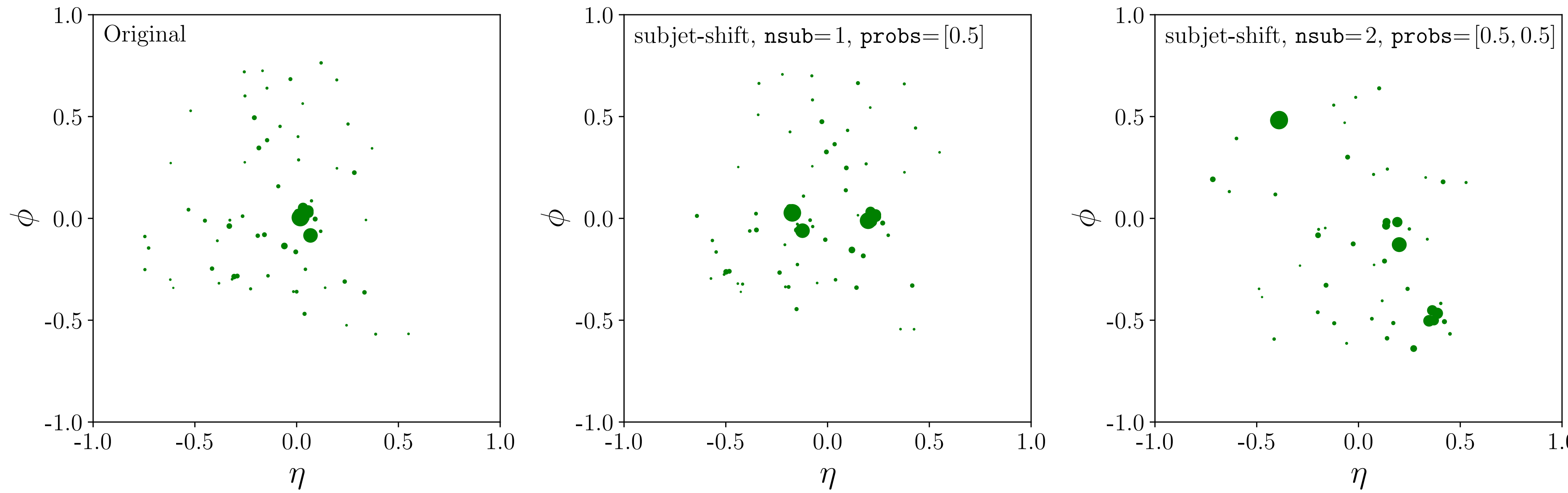
Work in progress - Dillon, Favaro, Fieden, Modak, Plehn

Preliminary - results to come

Exact same procedure as before, except different augmentations, for example…

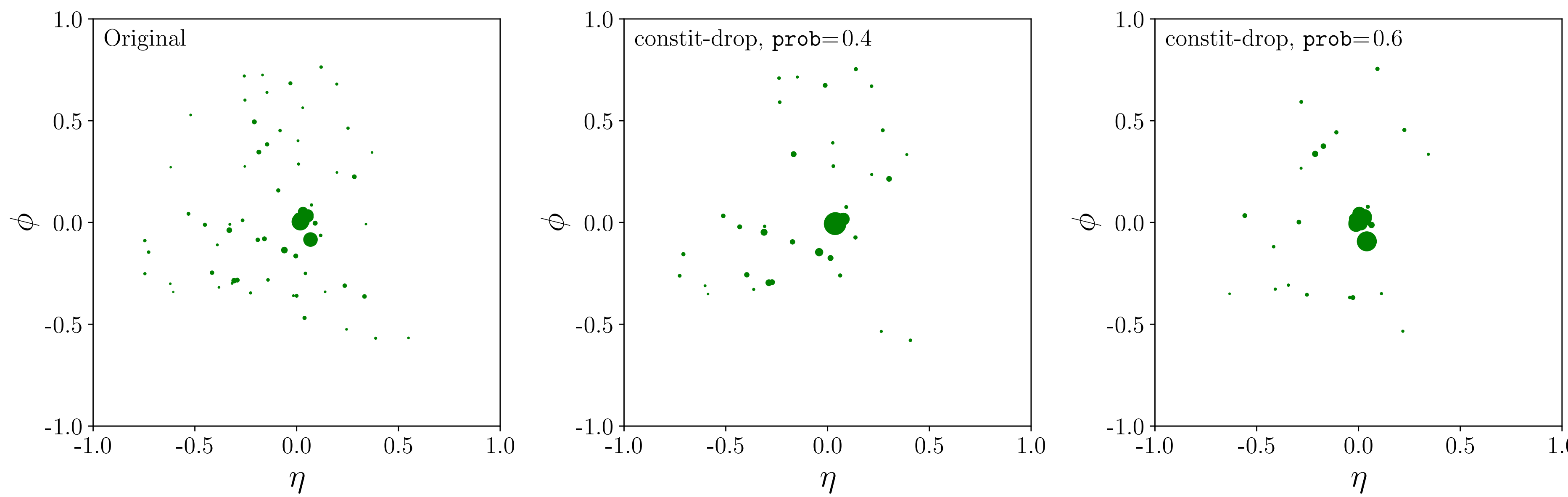
sub-jet shifts

Create subjets within a jet by randomly selecting constituents and shifting them by a random amount.
- heavy decays in jet



constituent drop

Create low multiplicity jets by randomly removing constituents and re-scaling the p_T's
- semi-visible jets



[https://github.com/bmdillon/AnomalyCLR-jets]

Conclusions

Conclusions

- While supervised learning works extremely well on low-level raw data, the same is not true for anomaly detection
- AE-based observables and CWoLa methods both have their disadvantages:
  - AE results depend on data representations
  - CWoLa results degrade with more observables / model-agnosticism
- Self-supervision: extracting features from unlabelled data through pseudo-tasks
  - Allows us to build highly expressive physical representations
  - Can be used for anomaly detection tasks
  - Demonstrated this on event level data (CMS ADC2020)
- Further work:
  - Self-supervision for anomalous jet tagging