Anomalies and self-supervised learning

Luigi Favaro

IML Machine Learning Working Group - 14/02/2023

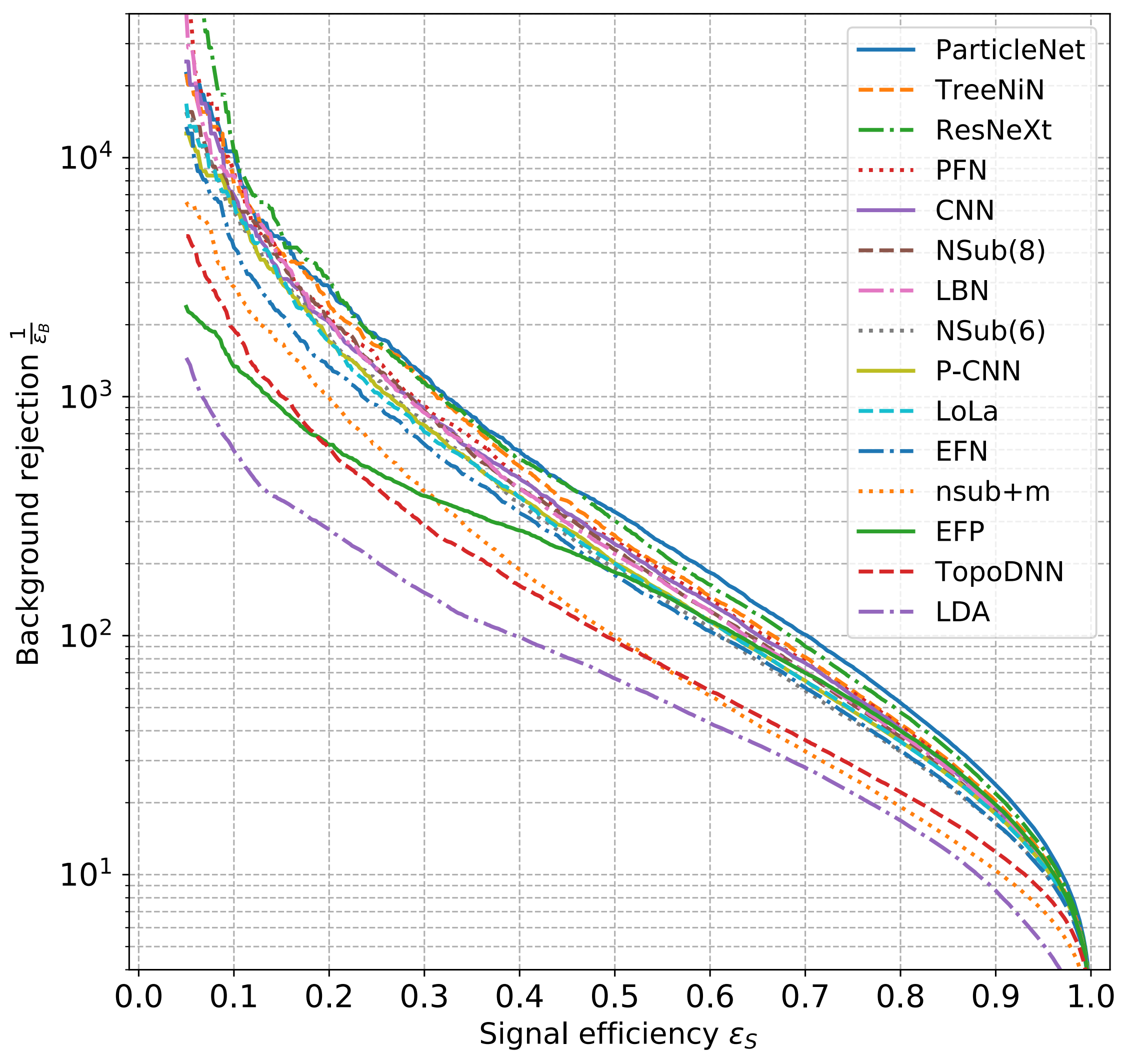
Universität Heidelberg
Zukunft. Seit 1386.

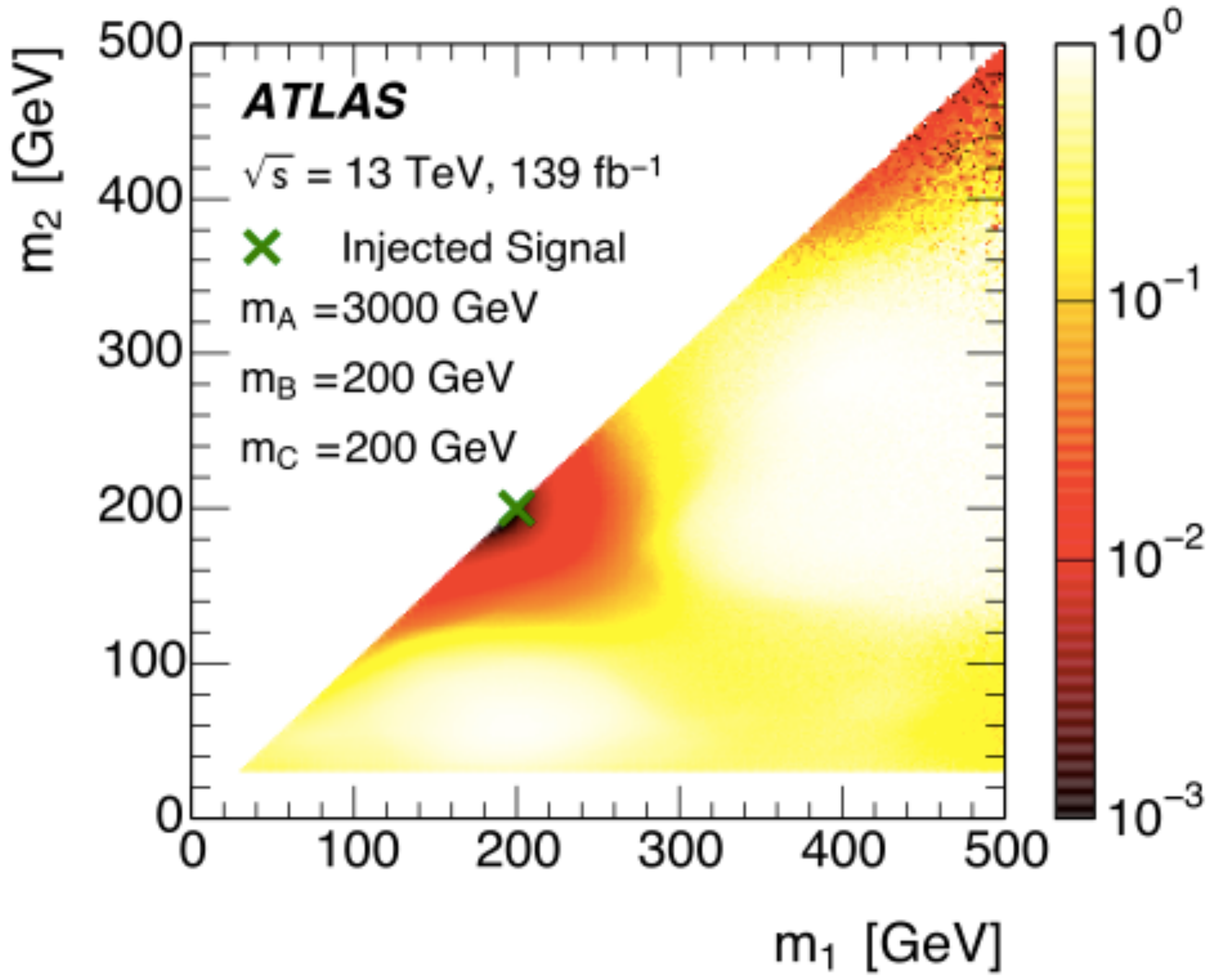
Model-agnostic searches & ML

- Are we leaving stones unturned? Can we answer this question only via direct searches?
- **Anomaly searches**: define background from the data and find “anomalous” events

a known problem in Machine Learning (or not?)

- looking for group anomalies
- robust anomaly detection tool
- level of agnosticism
- performing analysis (bump hunt, ABCD, …)



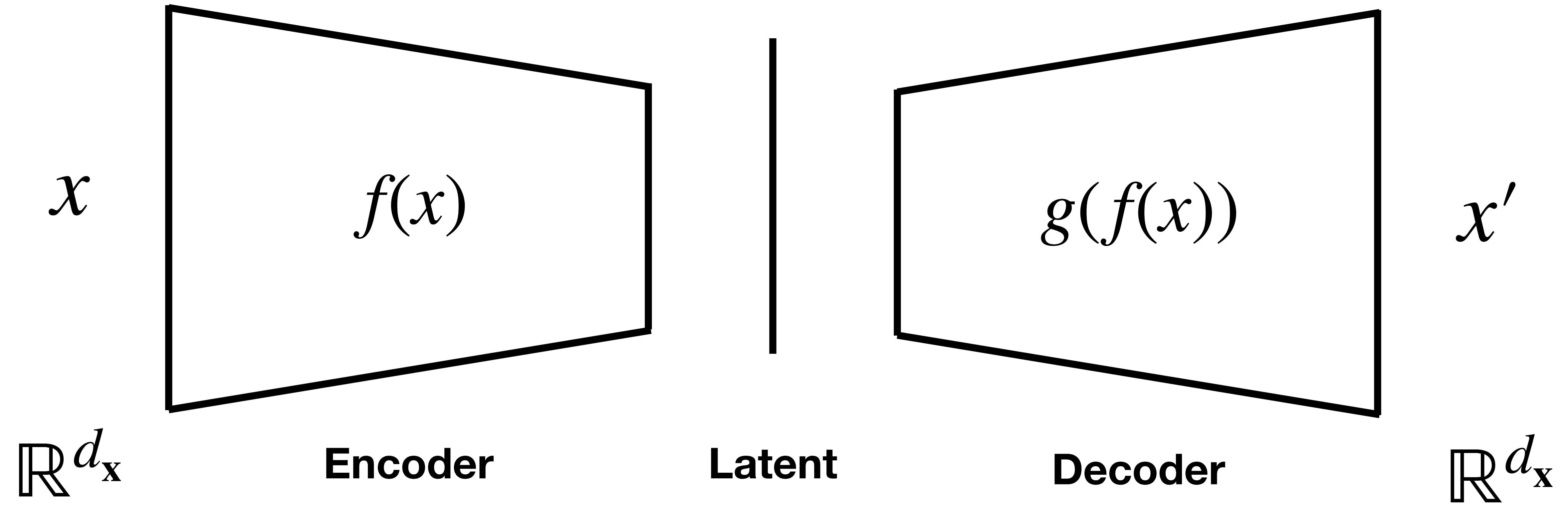


Already many interesting challenges/applications of ML techniques

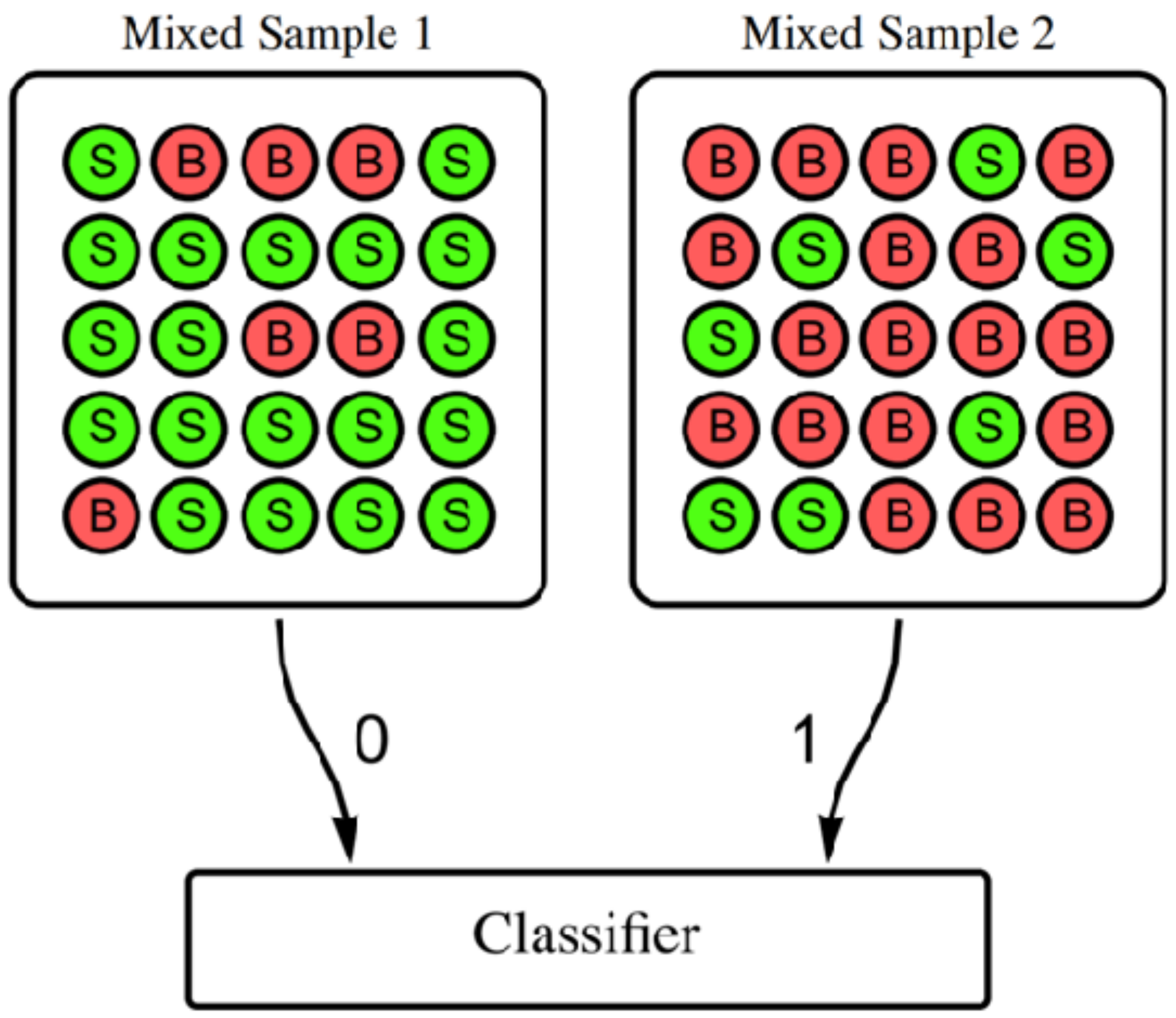
Model-agnostic searches & ML

Two big families:

Autoencoders (AE)



Classification without labels (CWOLA)



Autoencoders for Jet tagging

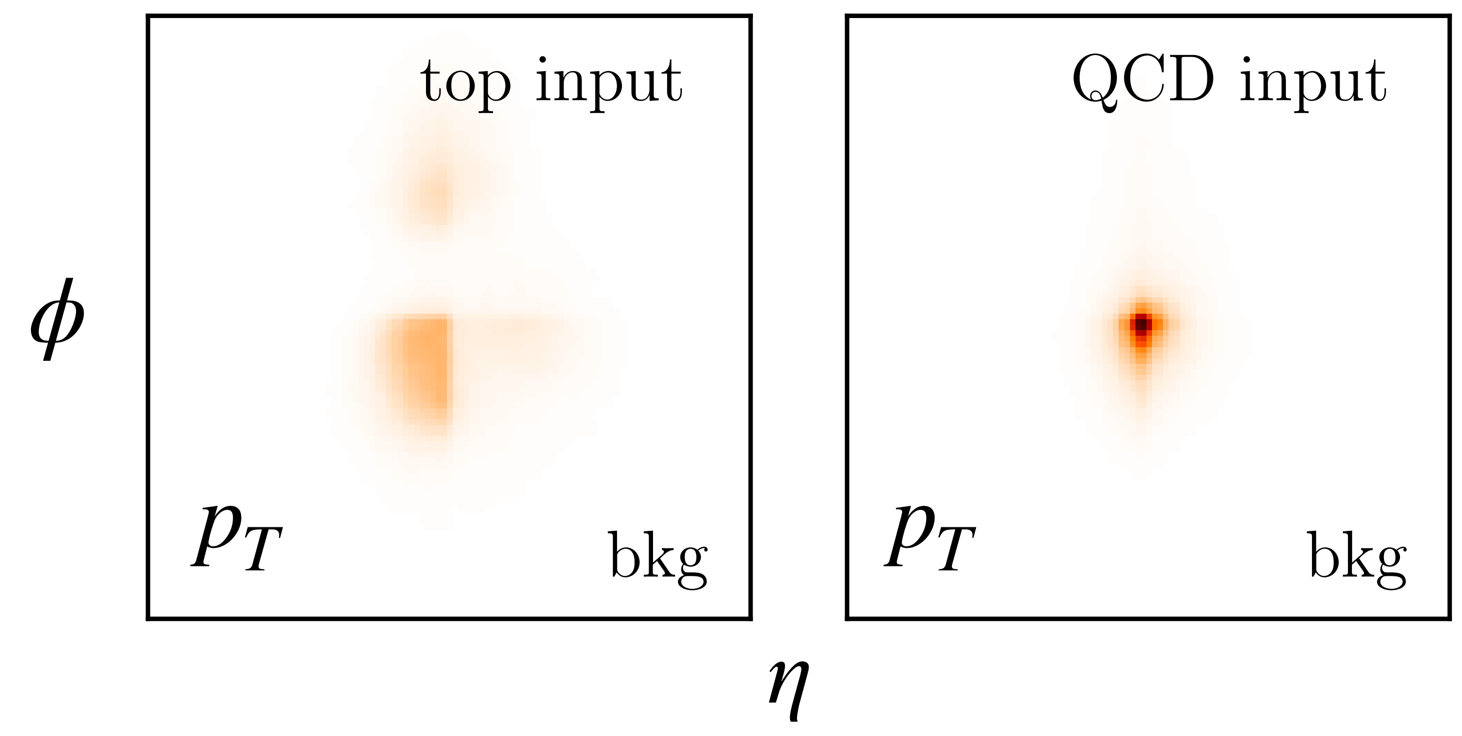
- Auto-Encoders can easily tag complex signals;
- the opposite is not generally true → ‘complexity bias’

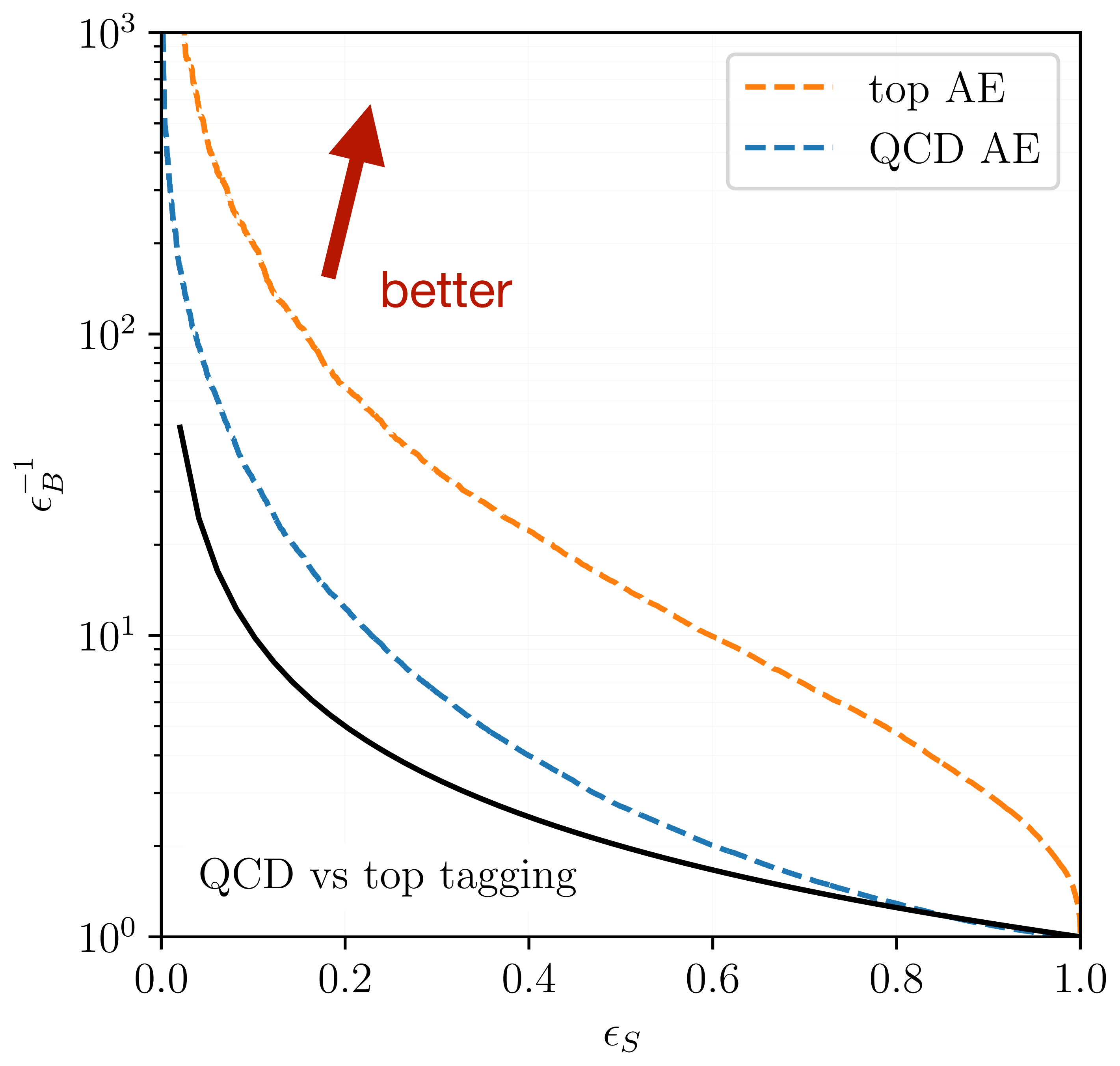
Robustness test: inverse training

- take a background and a signal signature
- train an AE on the direct and inverse task

Example: QCD tagging

Scores not invariant to data preprocessing

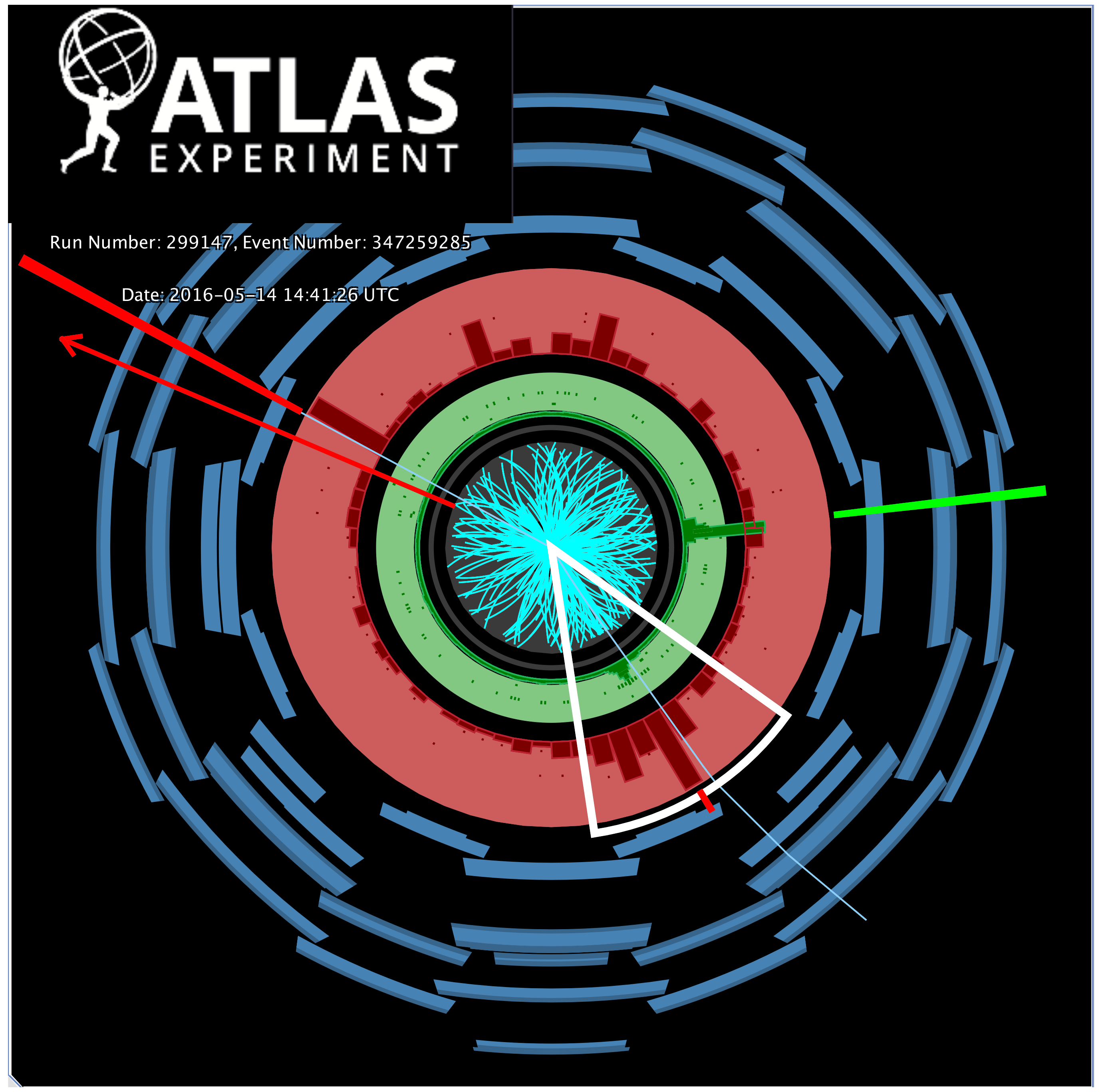




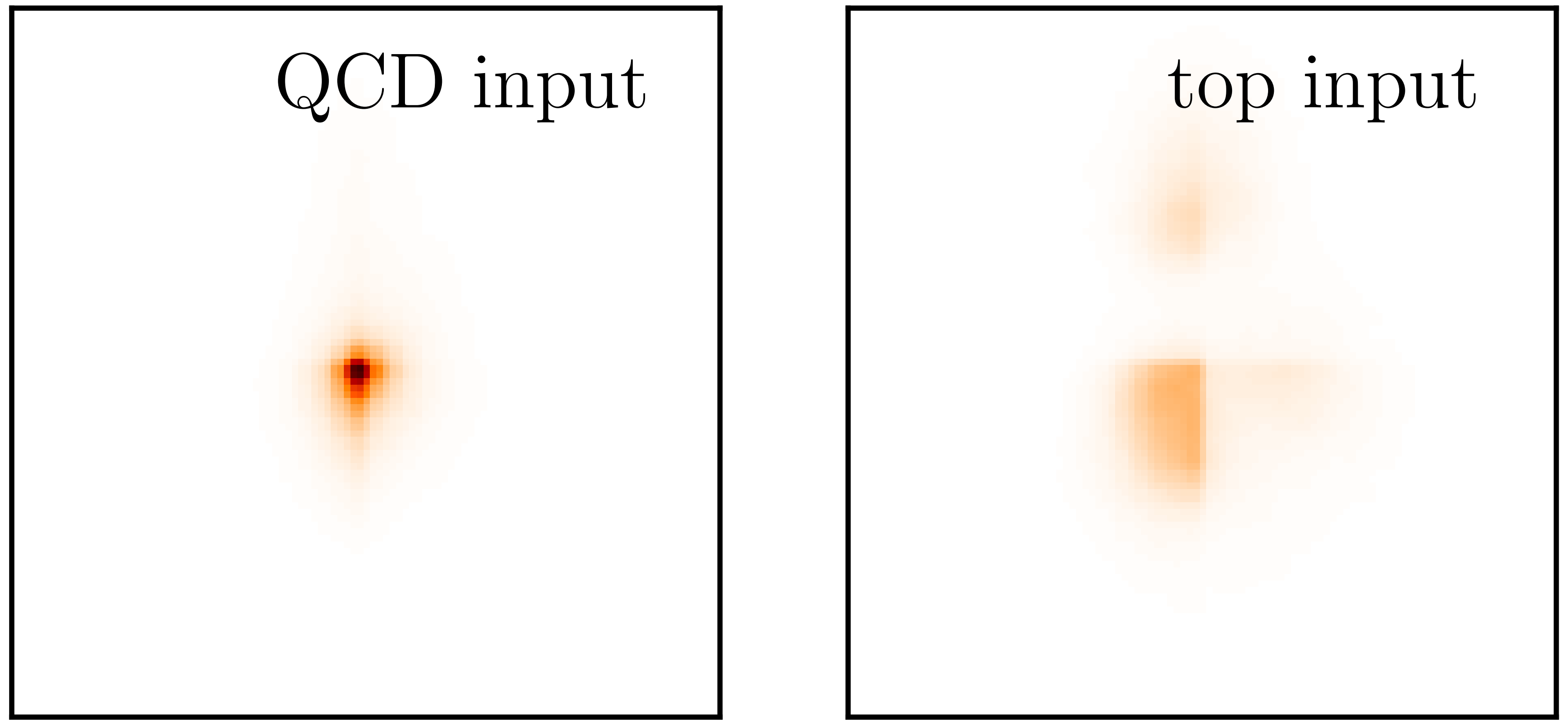
Finding the right observables

How to choose the best representation?

Examples:



Reconstructed objects



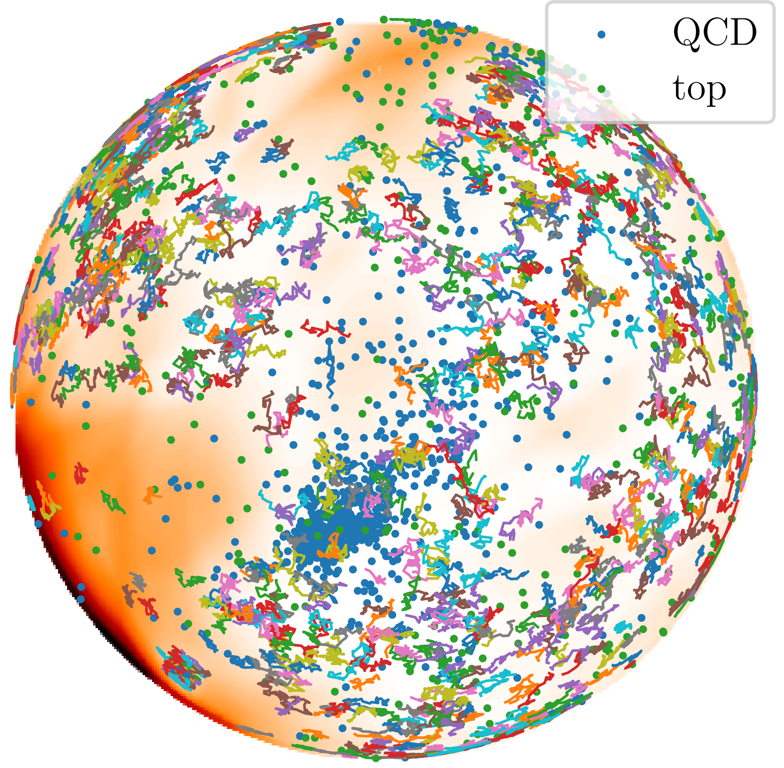
Jet constituents

A common solution?

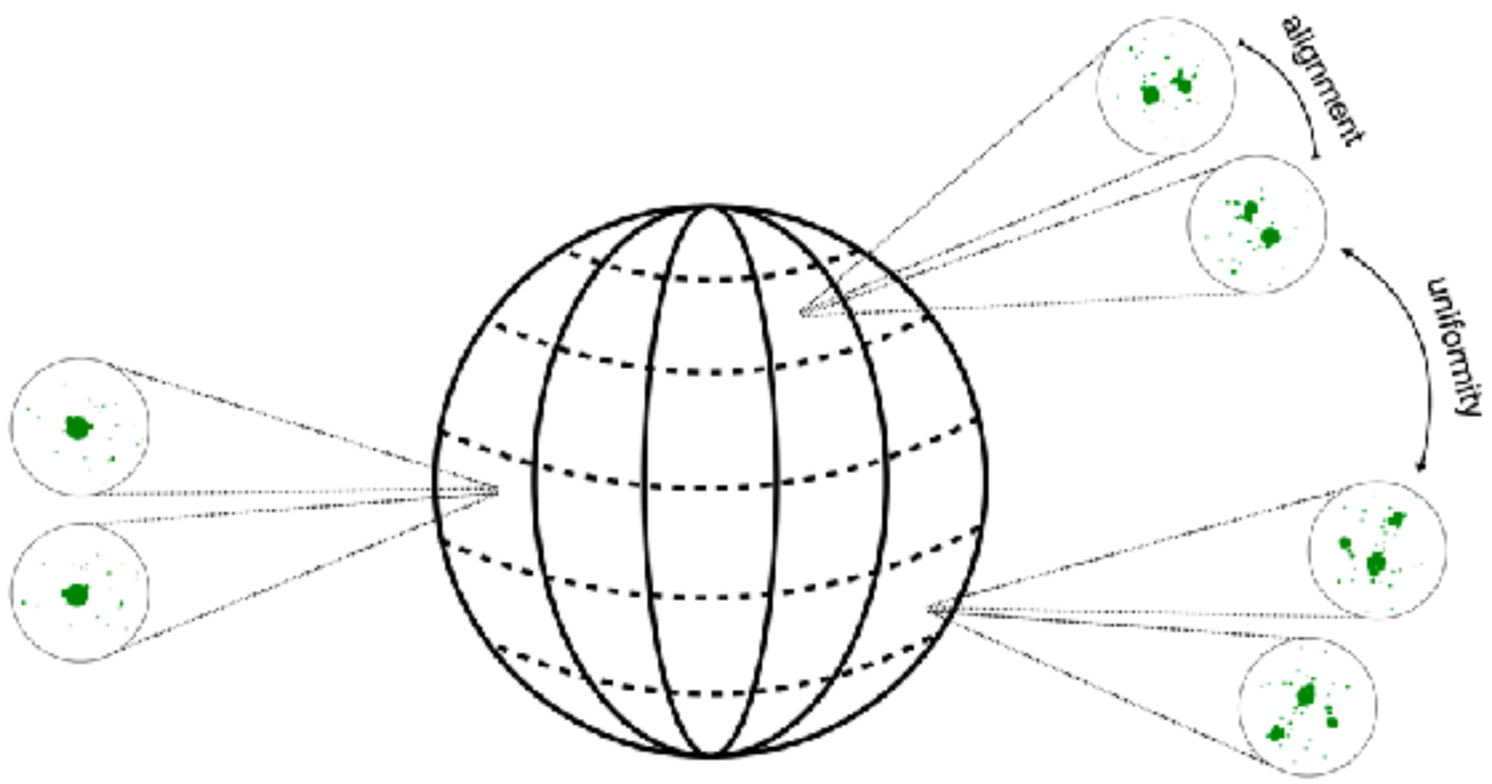
A common approach to these problems is **Contrastive Learning** (CLR):

- phrase the objective loss as a contrastive loss with
  - **positive samples**
  - **negative samples**
- shape a non-degenerate energy landscape

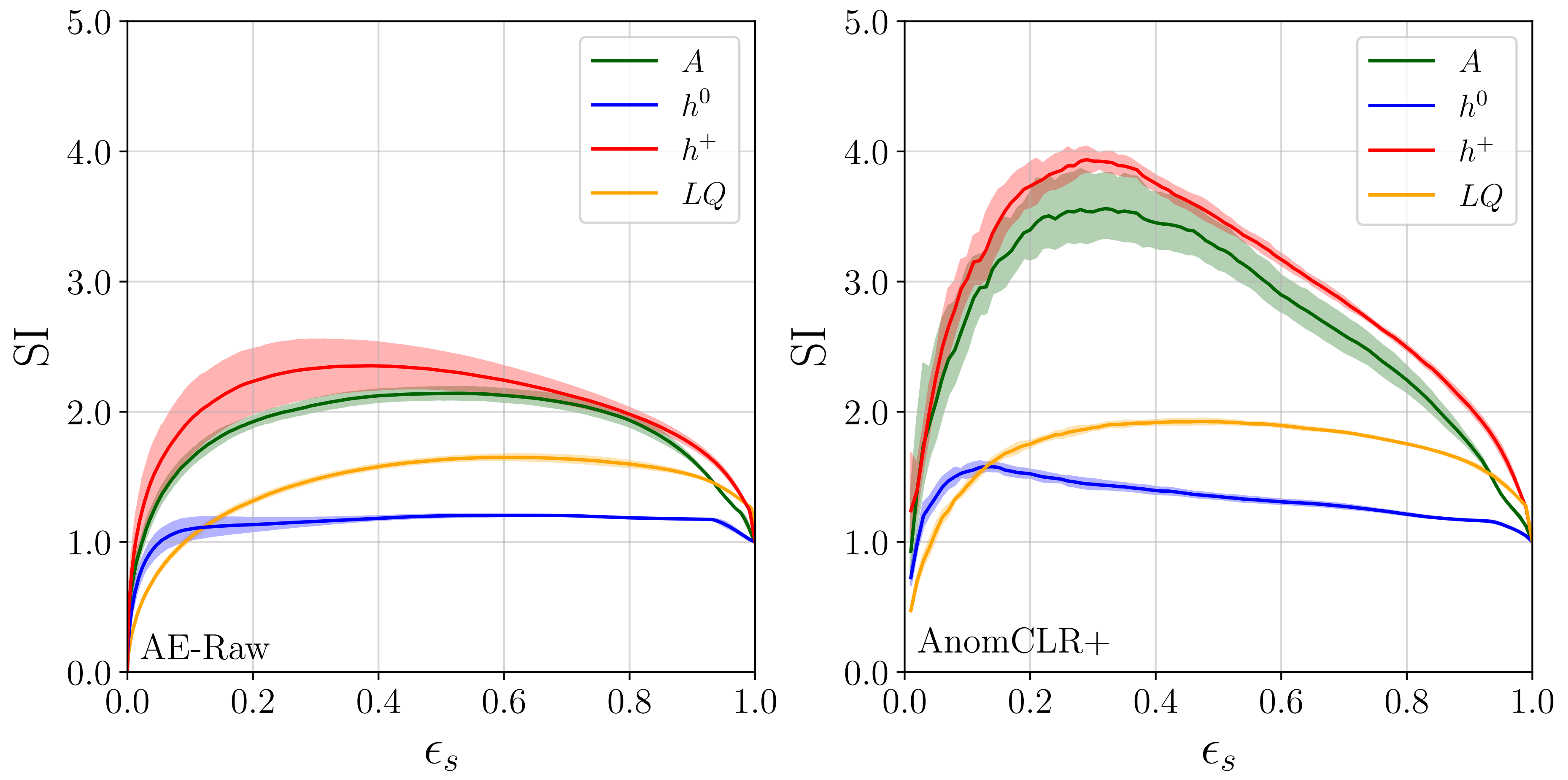
Normalized Auto-Encoders



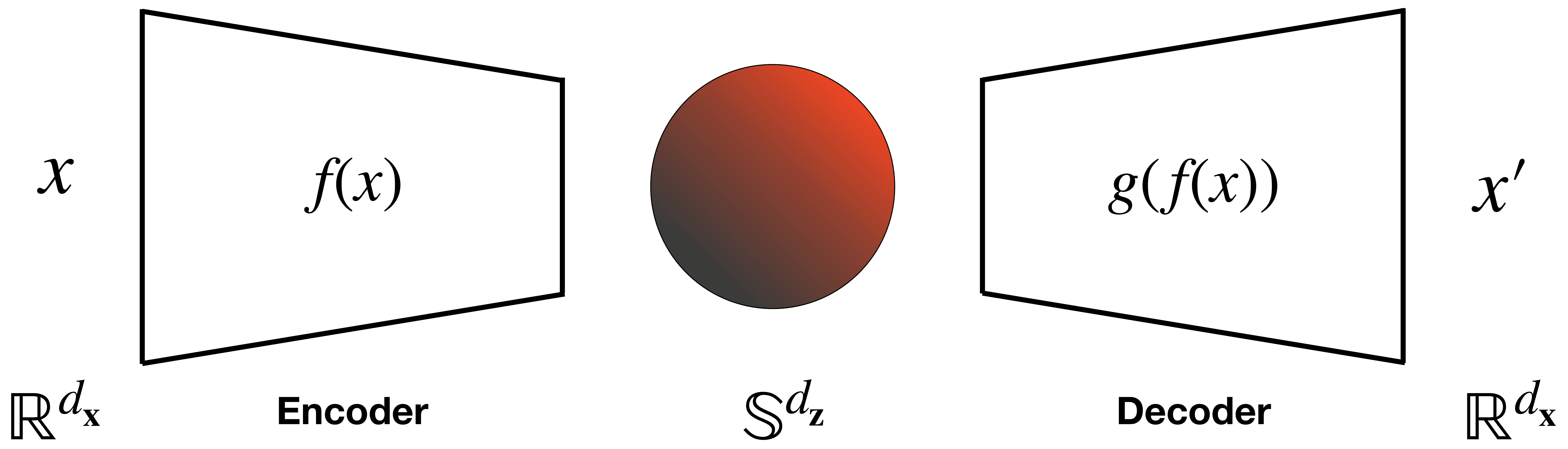
JetCLR



AnomalyCLR



Normalized Auto-Encoders



Building a NAE:

- define two neural networks like an usual Auto-Encoder;
- encode features in a low-dimensional latent space;
- set the latent space to a spherical hyper-surface $\mathbb{S}^{d_{\mathbf{z}}}$;
- use the reconstruction error as anomaly score, MSE(x, x').

Training a NAE

We need to explore the anomaly score space during training → **looking for a normalized distribution**

Define a Boltzmann probability distribution and use the MSE as energy function: $$p_\theta(x) = \frac{e^{-E_\theta(x)}}{\Omega}$$

$$\Omega = \int_x e^{-E_\theta(x)}\, dx \qquad\qquad E_\theta(x, x') = \|x - x'\|_2$$

we can train by minimizing the **negative log-likelihood** of the probability distribution:

$$\mathscr{L} = -\log p_\theta(x) = E_\theta(x) - \log \Omega$$

[Autoencoding under normalization constraints, Yoon S. et al. arXiv:2105.05735]
[A Normalized Autoencoder for LHC triggers, Dillon B. et al. arXiv:2206.14225]

Training a NAE

Consider the gradients of the loss function:

$$\nabla_\theta \mathscr{L} = \nabla_\theta E_\theta(x) - \nabla_\theta \log \Omega$$

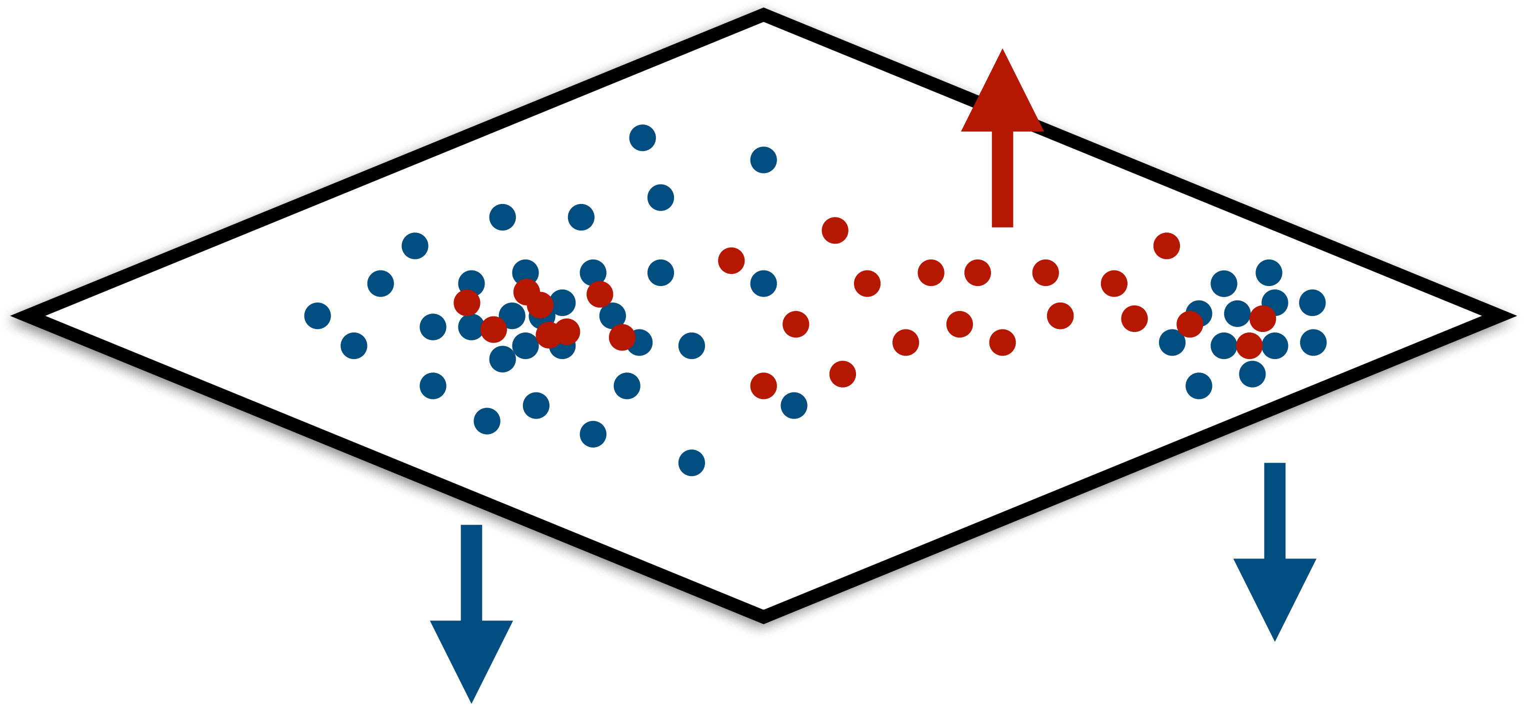
Minimizes the usual AE reconstruction error;

Can be rewritten as: $\nabla_\theta E_\theta(x)\,, \quad x \sim p_\theta(x)$

Rewriting the gradient of the loss function:

$$\nabla_\theta \mathscr{L} = \mathbb{E}\left[\,\nabla E_\theta(\mathbf{x})\,\right]_{x\sim p_{data}} - \mathbb{E}\left[\,\nabla E_\theta(\mathbf{x})\,\right]_{x\sim p_\theta}$$

- positive energy: gradient descent step
- negative energy: gradient ascent step



✓ **at equilibrium:** $p_\theta(x) = p_{data}(x)$

Normalization

Everything is really general…

… but why does this work?

Ω ***high-dimensional space*** $\rightarrow$ ***approx. high dimensional integral***

Input space is high dimensional $\rightarrow$ ***sampling from*** p_θ***?***

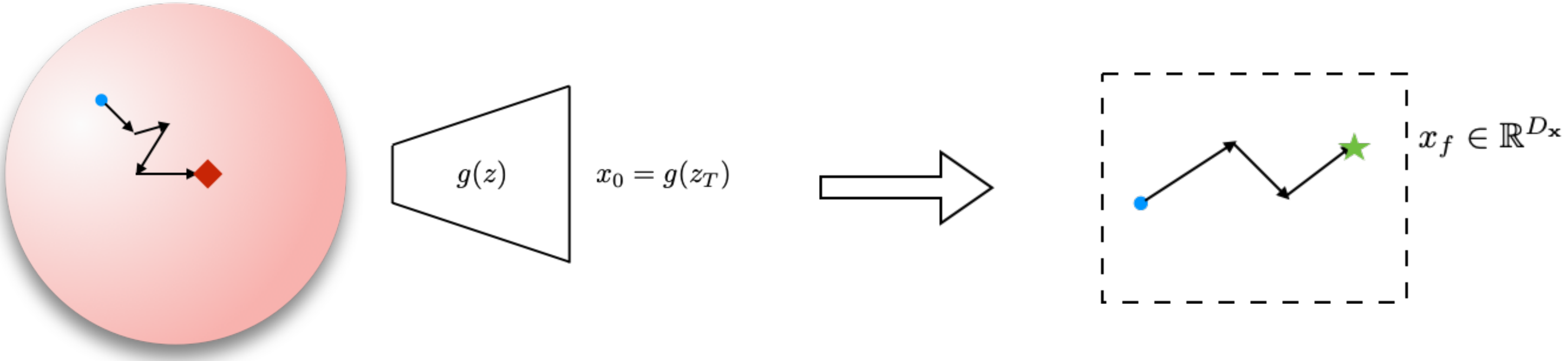
Sampling from p_θ

Sampling is done via two Langevin Markov chains:

- latent space: using the energy $H_\theta = E_\theta(g(z), f(g(z)))$;
- input space: through the distribution $p_\theta(x)$.

$$x_{t+1} = x_t + \lambda_t \nabla_x \log p_\theta(x) + \sigma_t \epsilon \qquad\qquad \epsilon \sim \mathcal{N}(0,1)$$

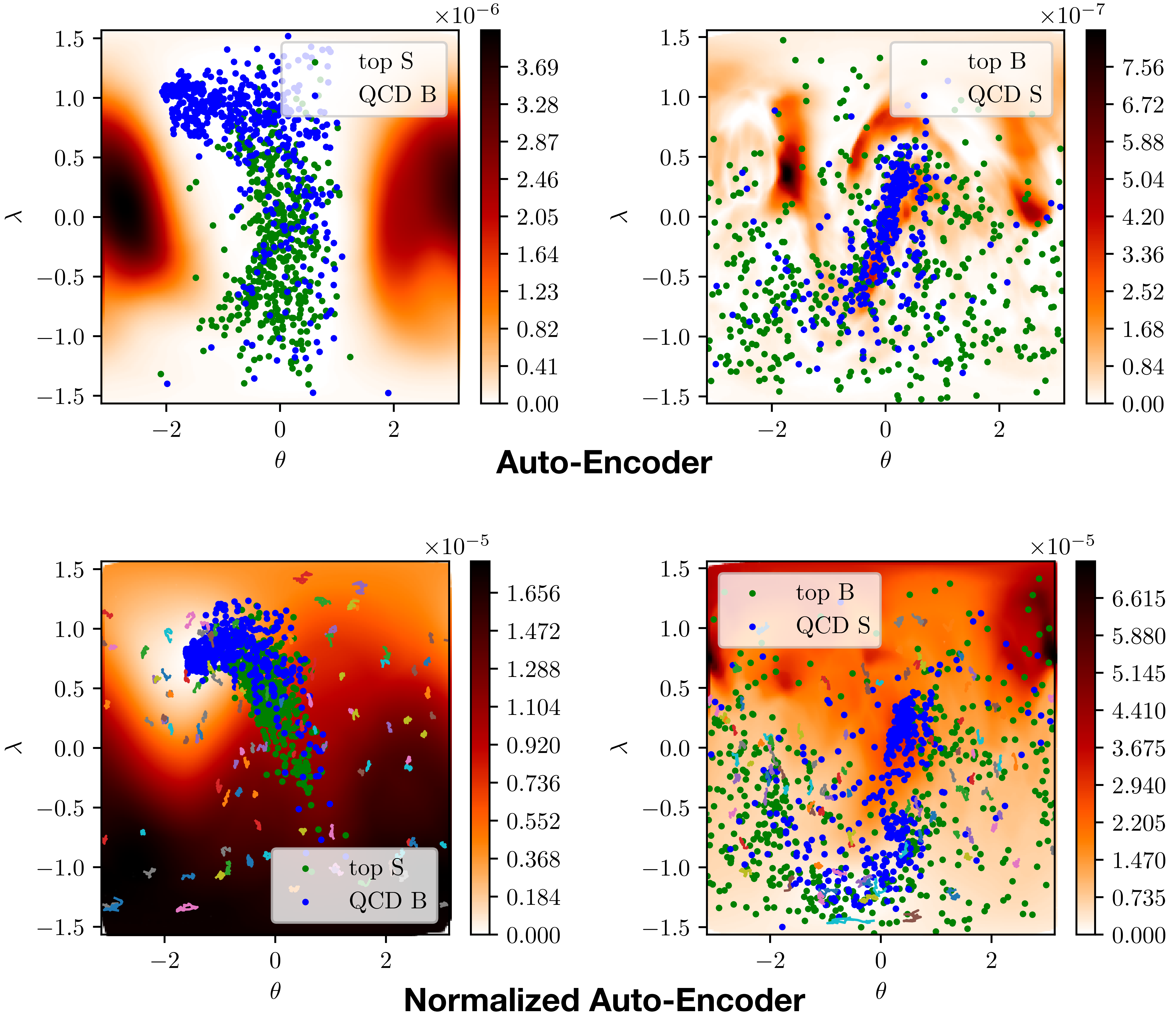
* small number of steps $\mathcal{O}(100)$, constrained into low energy regions by taking $\lambda > \sigma$



Results: decoder manifold

We can study what happens during training:

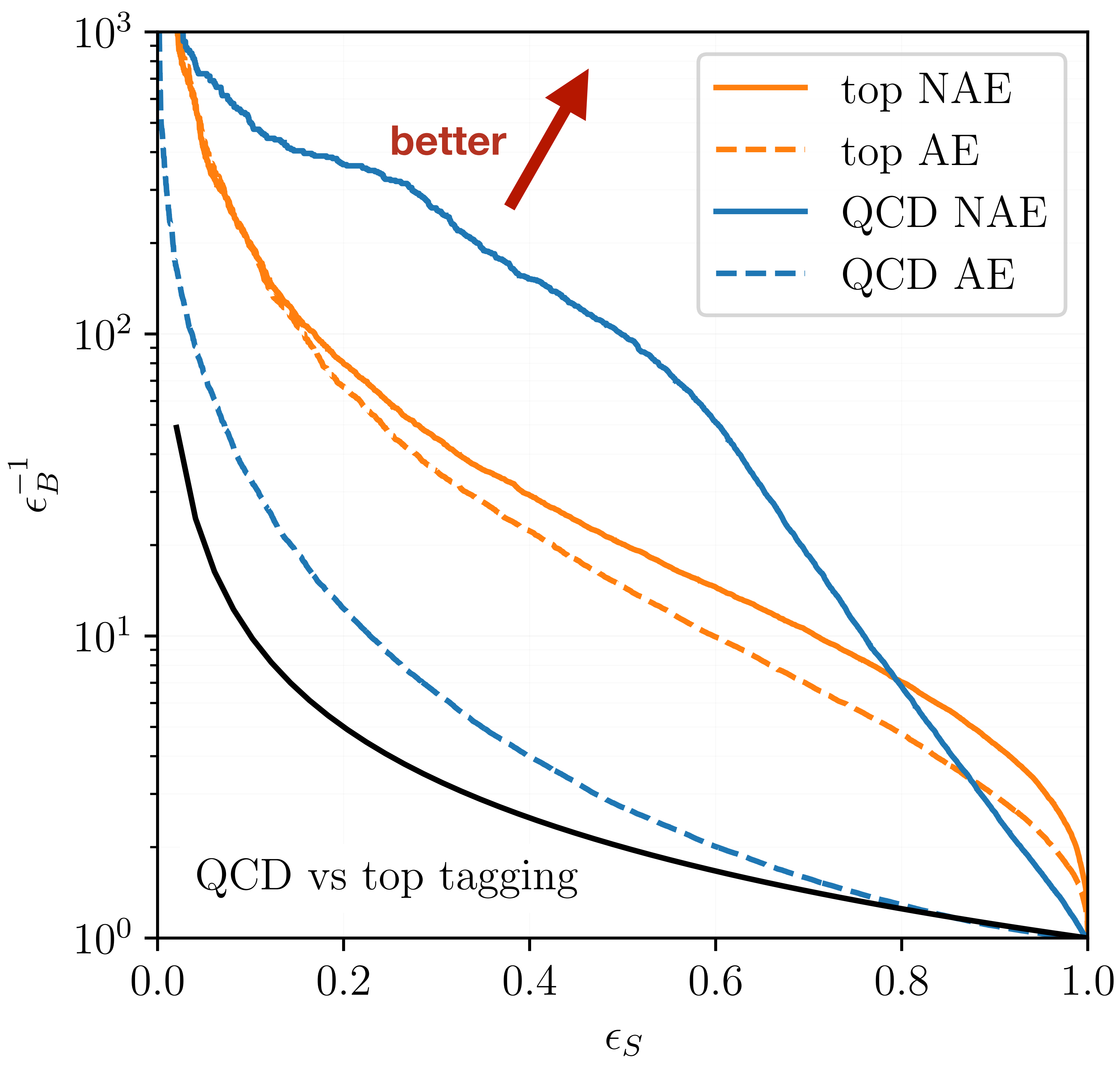
- 2D projection of the latent space;
- decoder manifold for tops is more complex
- inducing an underlying metric via $\log \Omega$;
- after training both QCD and top jets are mapped in high reconstruction regions of the decoder manifold;



Results: QCD vs top tagging

- AE trained on jet images fails at tagging QCD jets;
- an AE is able to interpolate the simpler QCD features;
- NAE explicitly penalizes well-reconstructed regions not in the training dataset;
- nice performance on both tasks, symmetric training.

Signal	NAE AUC	NAE $\epsilon_B^{-1}(\epsilon_S = 0.2)$	AE [1] AUC	DVAE [6] AUC
top (AE)	0.875	68	0.89	0.87
top (NAE)	0.91	80		
QCD (AE)	0.579	12	–	0.75
QCD (NAE)	0.89	350		



Self-supervision

- Neural Networks are not invariant to physical symmetries in data
- Typically solved through "pre-processing"

Our goal: control the training to ensure we learn physical quantities

What the **representations** should have: invariance to certain transformations of the jets/events

- CLR: map raw data to a new representation/observables
- Self-supervision: during training we use pseudo-labels, not truth labels

JetCLR

Dataset: mixture of top and QCD jets

Contrastive Learning paradigm:

- 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 $i \neq j$

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

Train a Transformer-encoder network to map the data to a new repr. space, $f : \mathscr{I} \rightarrow \mathscr{R}$

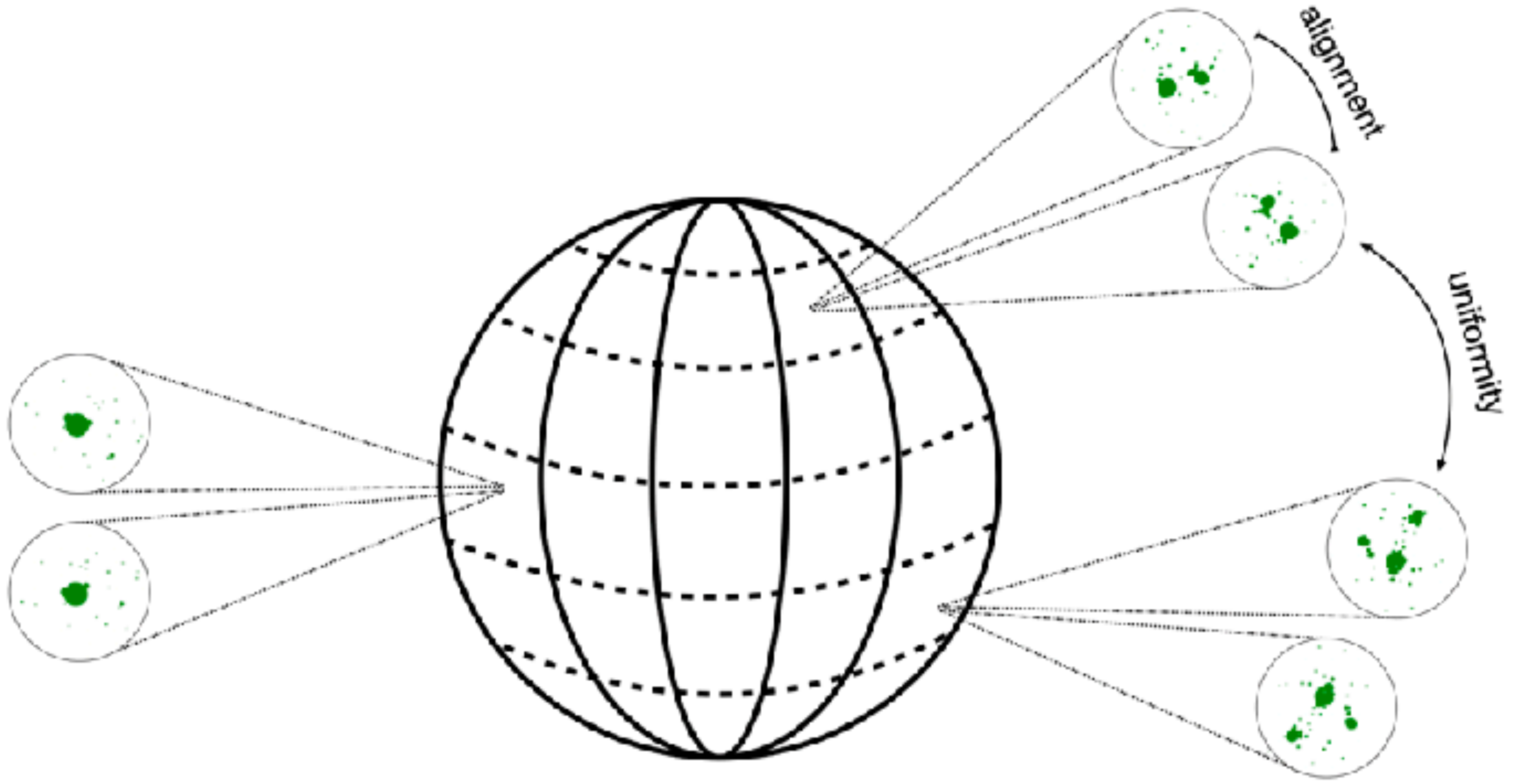
Loss function:

$$\mathscr{L} = -\log \frac{exp(s(z_i, z_j')/\tau)}{\sum_{x \in batch} \mathbb{I}_{i \neq j}[exp(s(z_i, z_j)/\tau) + exp(s(z_i, z_j')/\tau)]}$$

[Symmetries, safety, and self-supervision, Dillon B. et al. arXiv:2108.04253]

Invariances

$$\mathscr{L} = -\log \frac{exp(s(z_i, z_i')/\tau)}{\sum_{x \in batch} \mathbb{I}_{i \neq j}[exp(s(z_i, z_j)/\tau) + exp(s(z_i, z_j')/\tau)]}$$



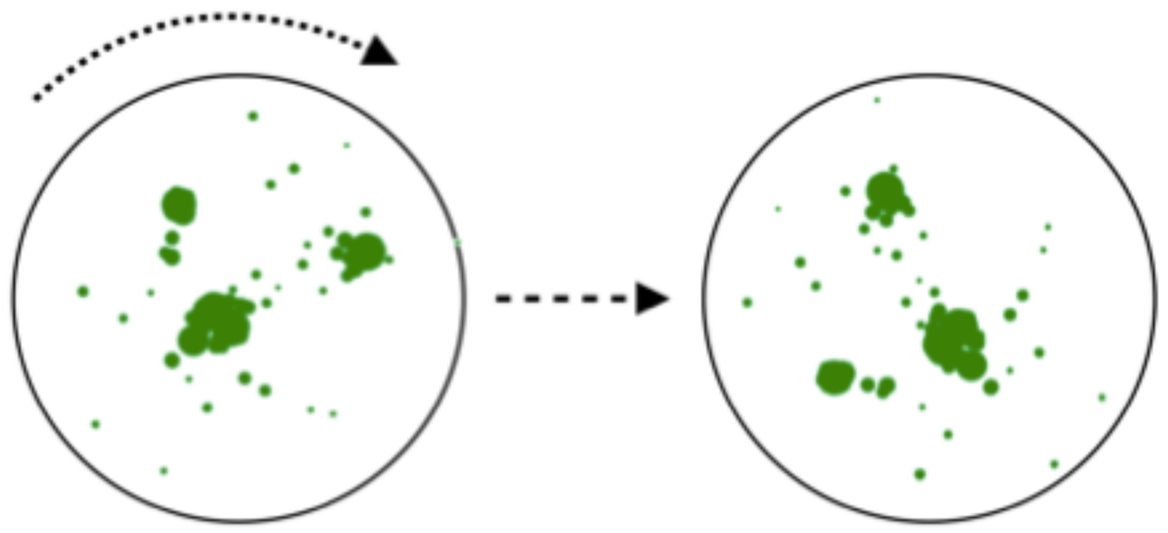
Similarity measure:

$$s(z_i, z_j) = \frac{z_i \cdot z_j}{|z_i||z_j|}, \qquad z_i = f(x_i)$$

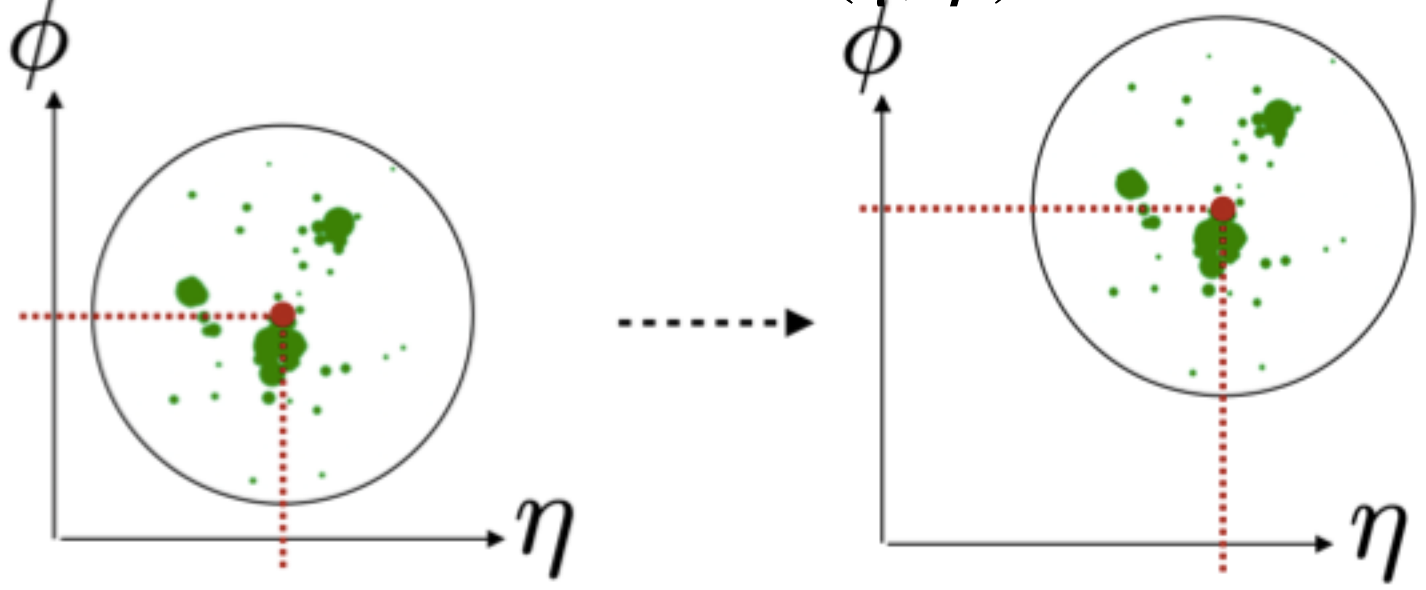
Applied augmentations:

- rotations
- translations
- collinear splittings
- low p_T smearing

rotation in $[0,2\pi]$



translation in (η, ϕ)



random split of constituents

$$p_{T,a} + p_{T,b} = p_T$$

$$\eta_a = \eta_b = \eta$$

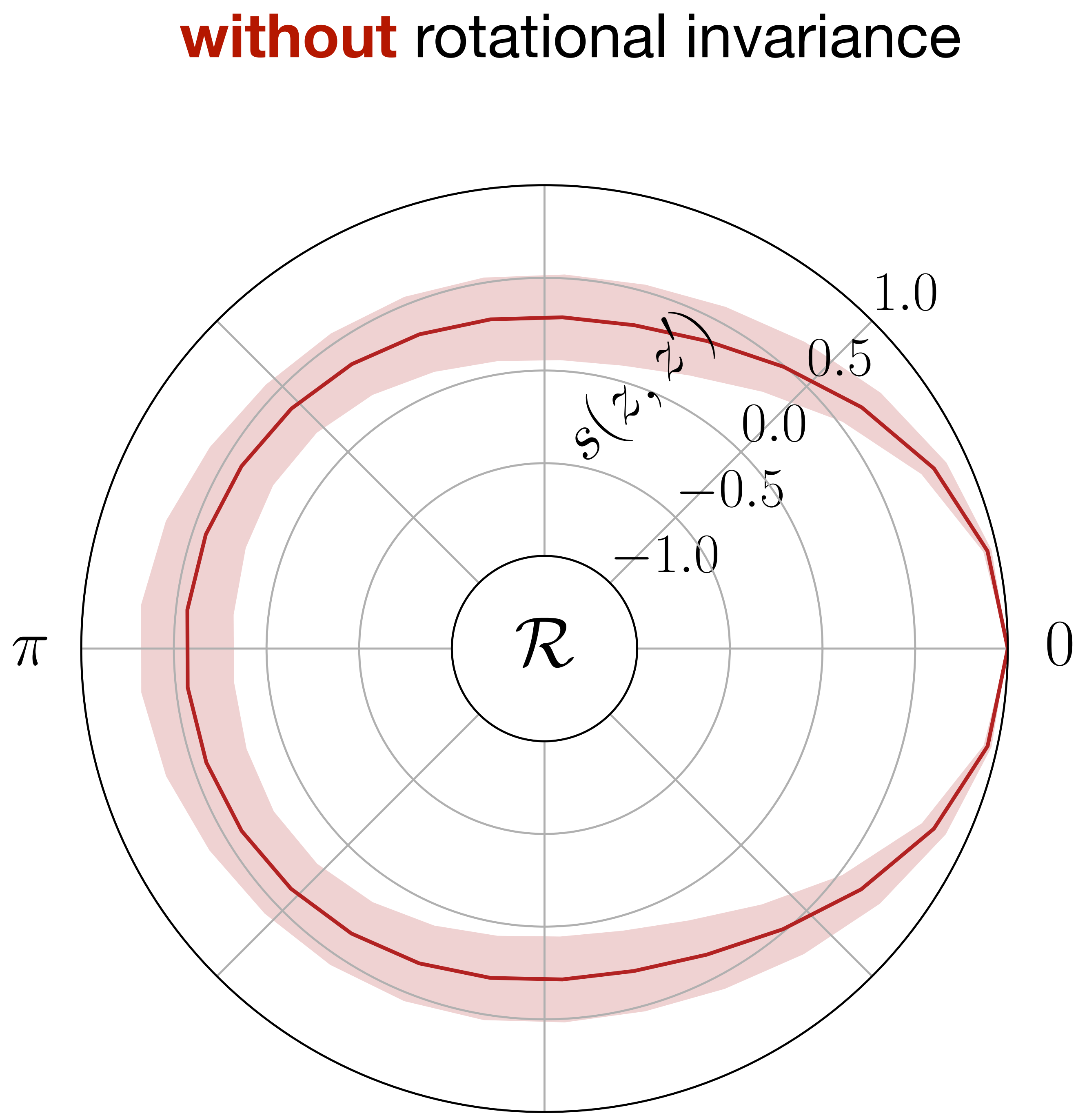
$$\phi_a = \phi_b = \phi$$

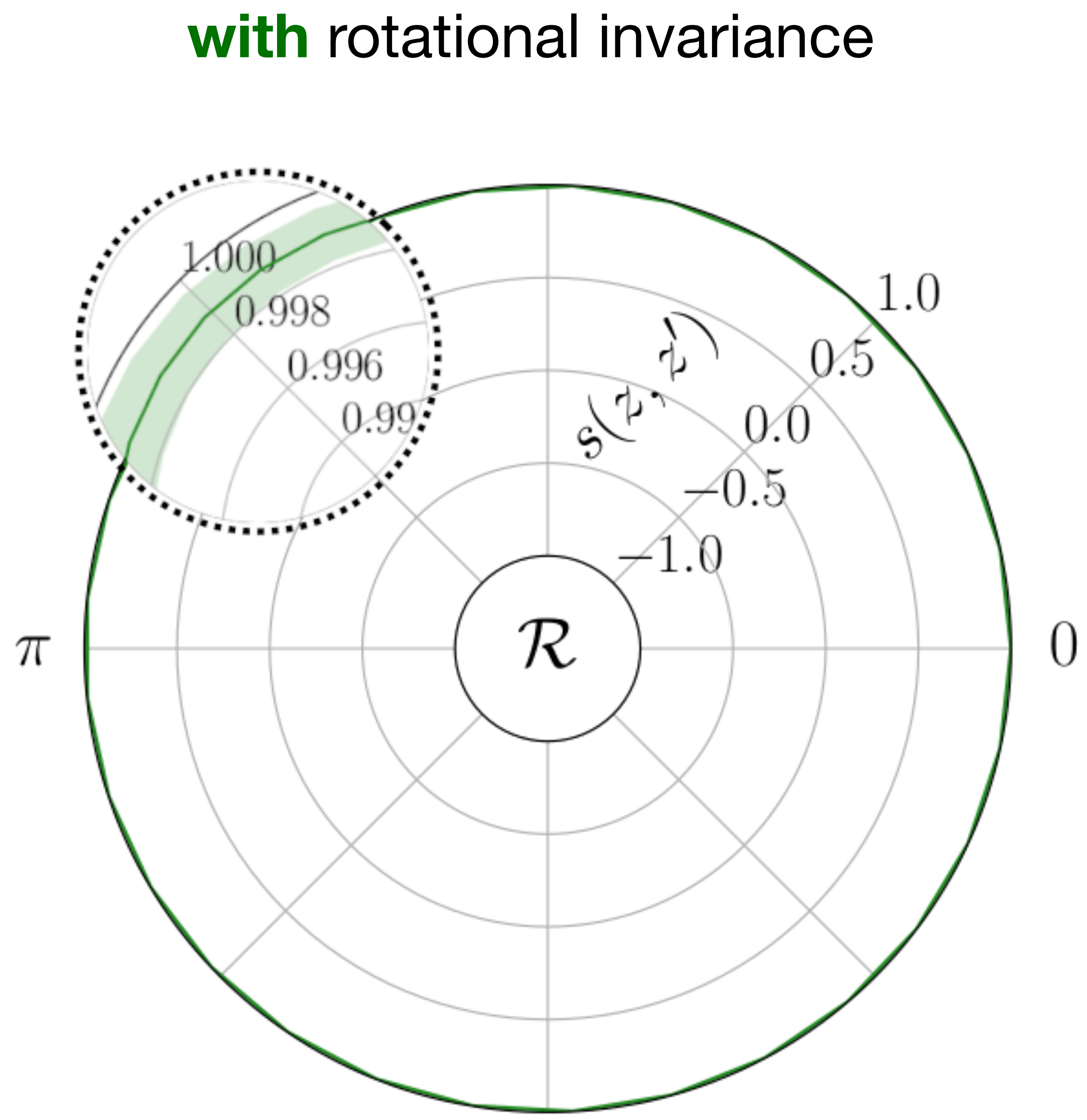
re-sampling of (η, ϕ)

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

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

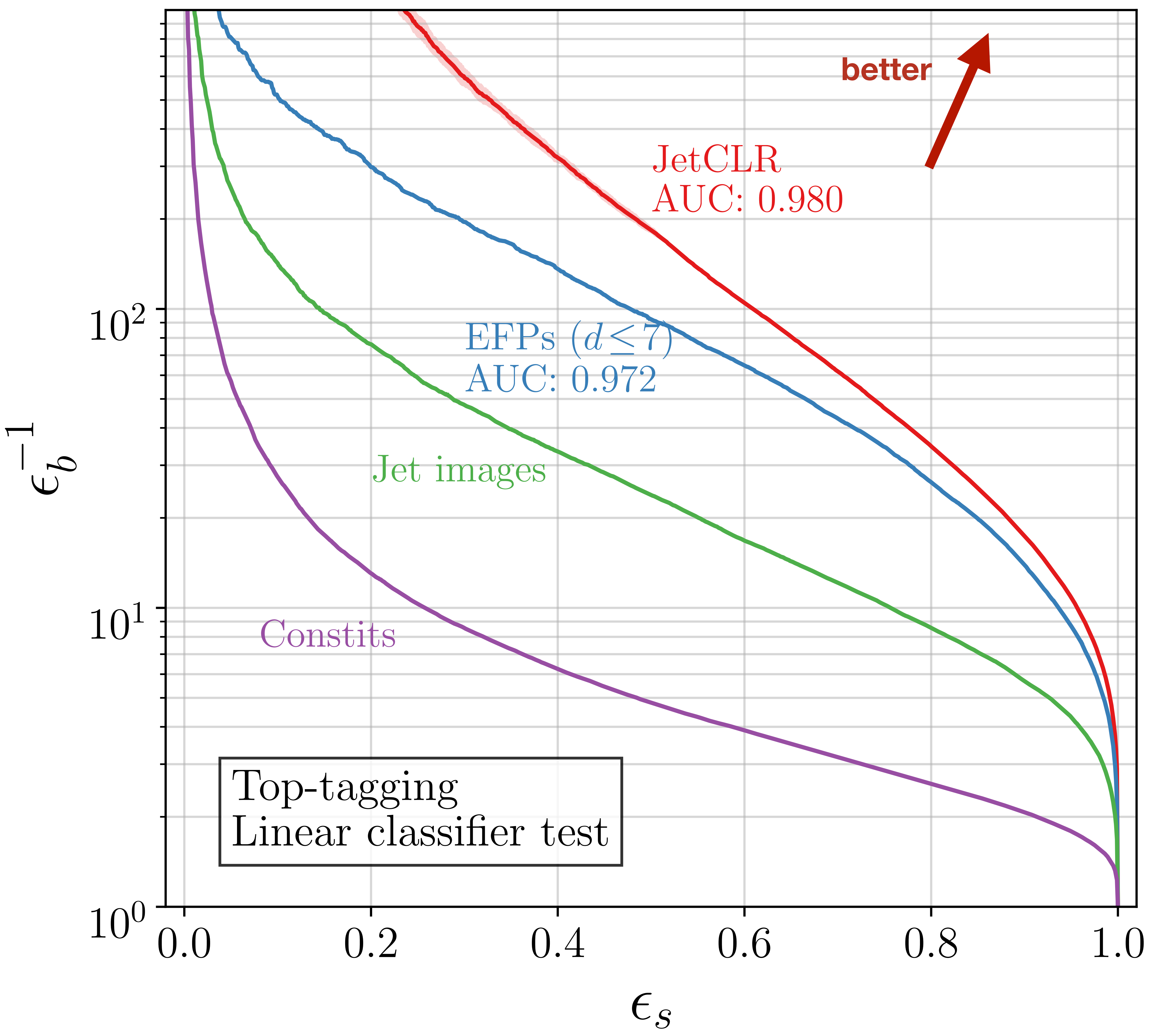
Are we learning invariances?





The network $f(\mathbf{x})$ is approximately rotationally invariant

Linear Classifier test



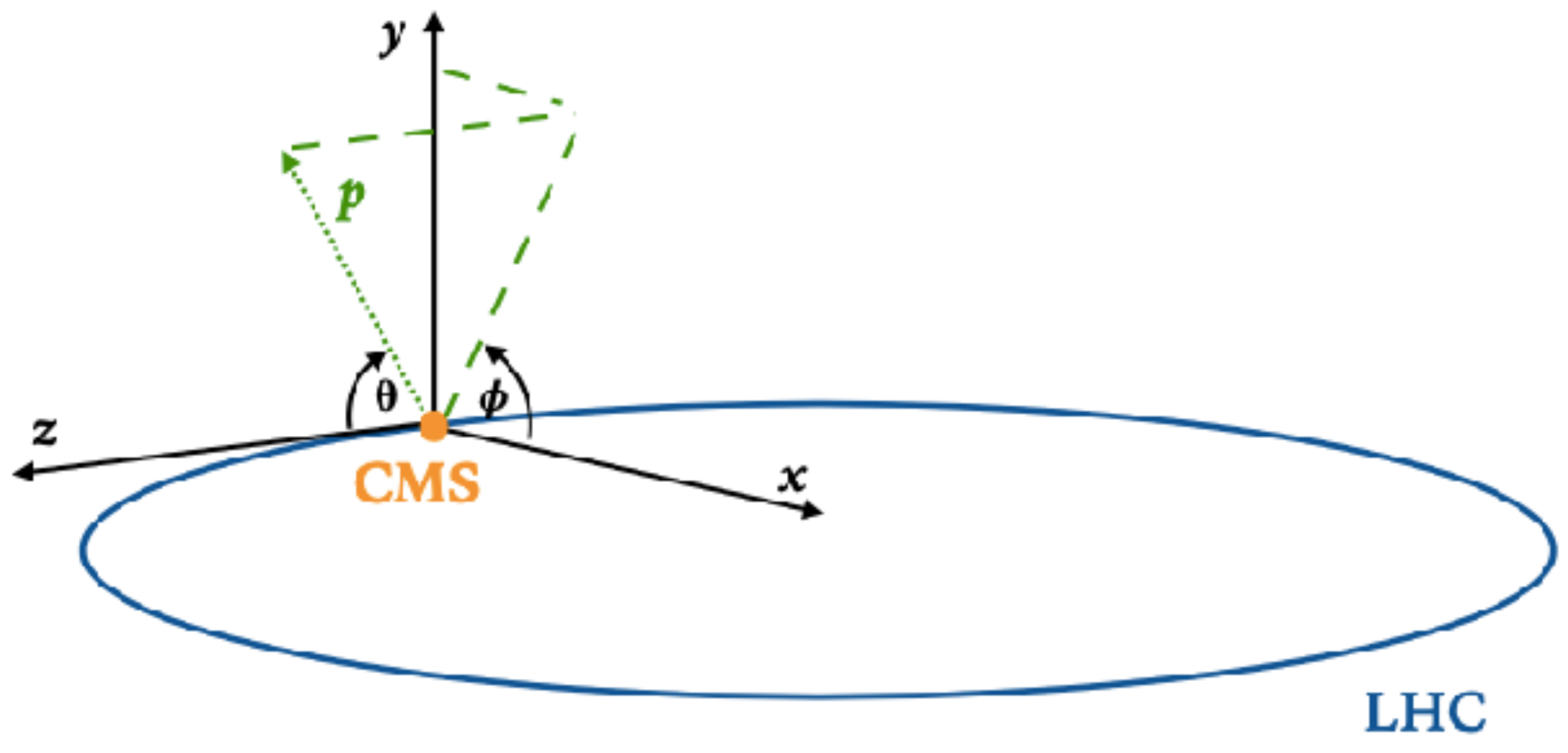
AnomalyCLR on events

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 in format: (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

[Anomalies, representations, and self-supervision, Dillon B. et al. arXiv:2301.04660]

Enhancing discriminative features

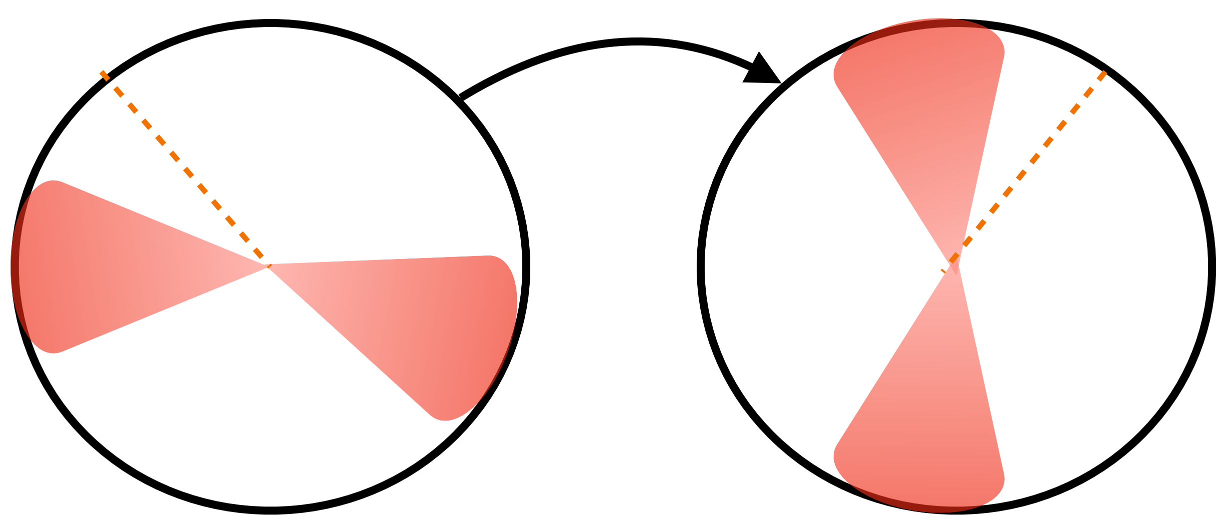
unsupervised training $\longrightarrow$ no signals available during training

Representations may not be sensitive to BSM features:

- physical augmentations: alignment between positive pairs
- anomalous augmentations: discriminative power of possible BSM features

Physical augmentations:

- azimuthal rotations
- η, ϕ smearing
- energy smearing



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

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

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

Anom. augmentations are motivated by non-SM features $\longrightarrow$ **model-agnosticism is preserved**

Anomalous augmentations

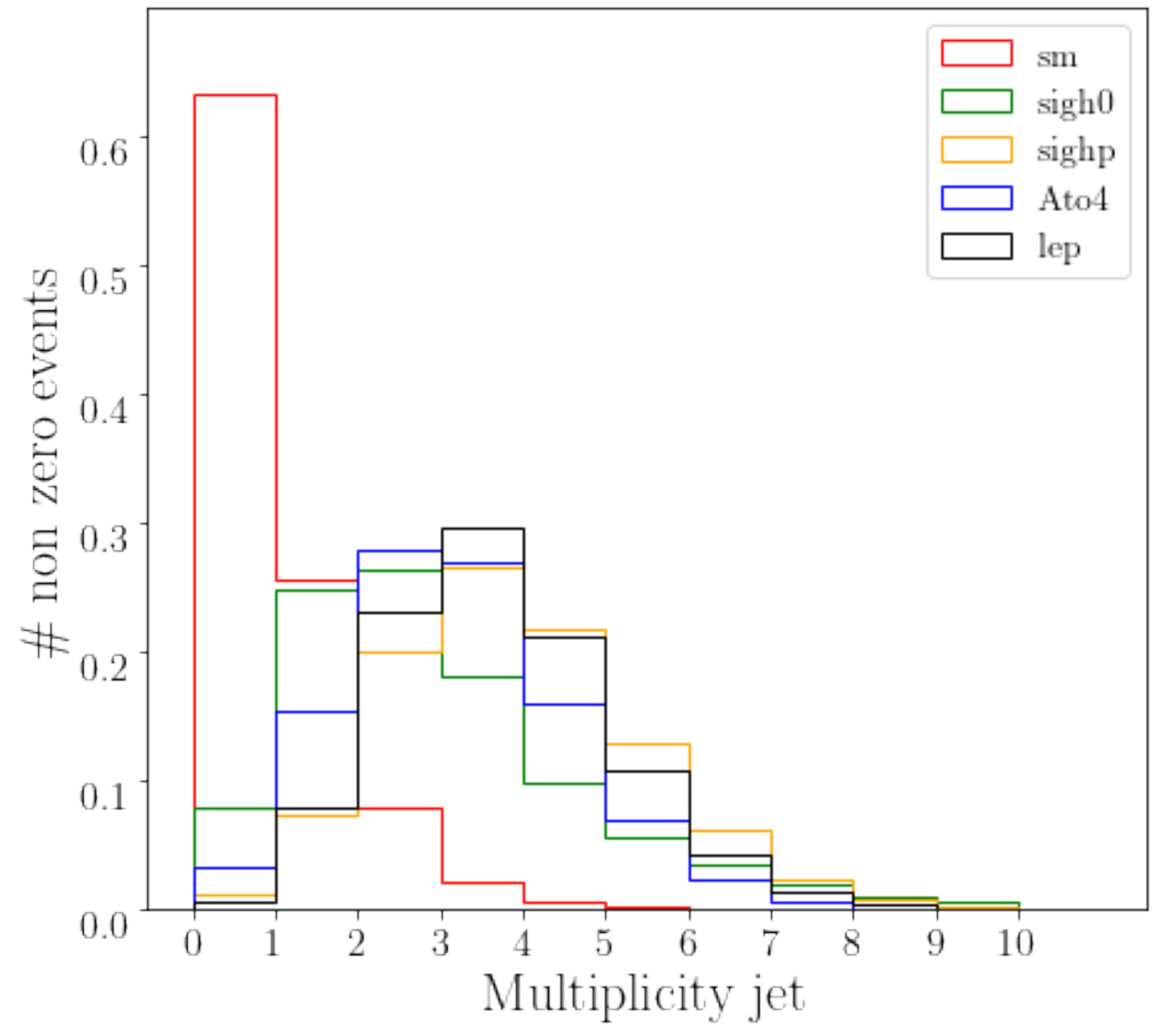
Loss function:

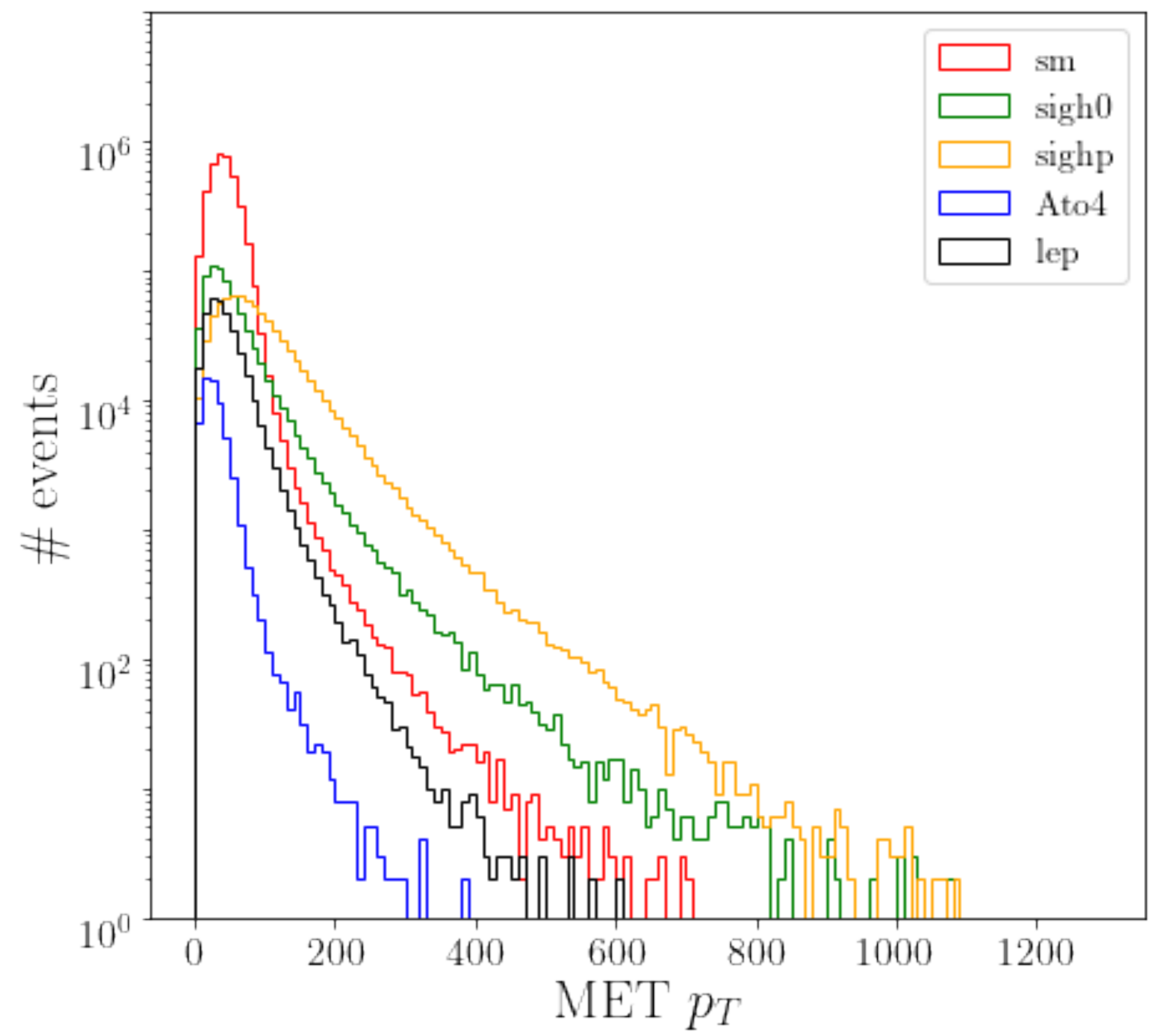
$$\mathscr{L}_{AnomCLR+} = -\log e^{[s(z_i,z_i')-s(z_i,z_i^*)]/\tau} = \frac{s(z_i,z_i^*)-s(z_i,z_i)}{\tau}$$

Anomalous augmentations:

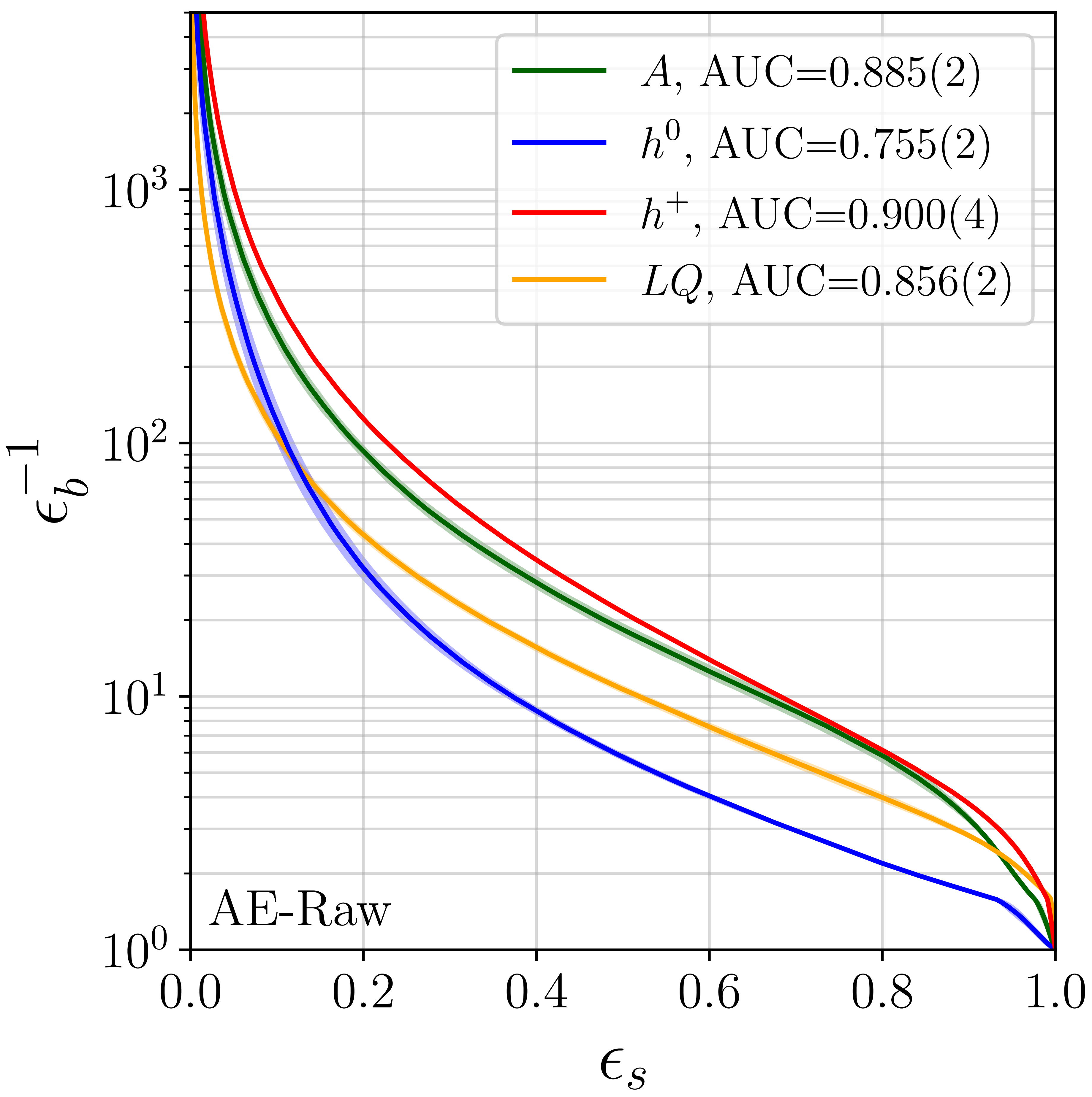
- multiplicity shifts:
  - add a random number of particles, update MET
  - split existing particles, keeping total p_T and MET fixed
- p_T and MET shifts

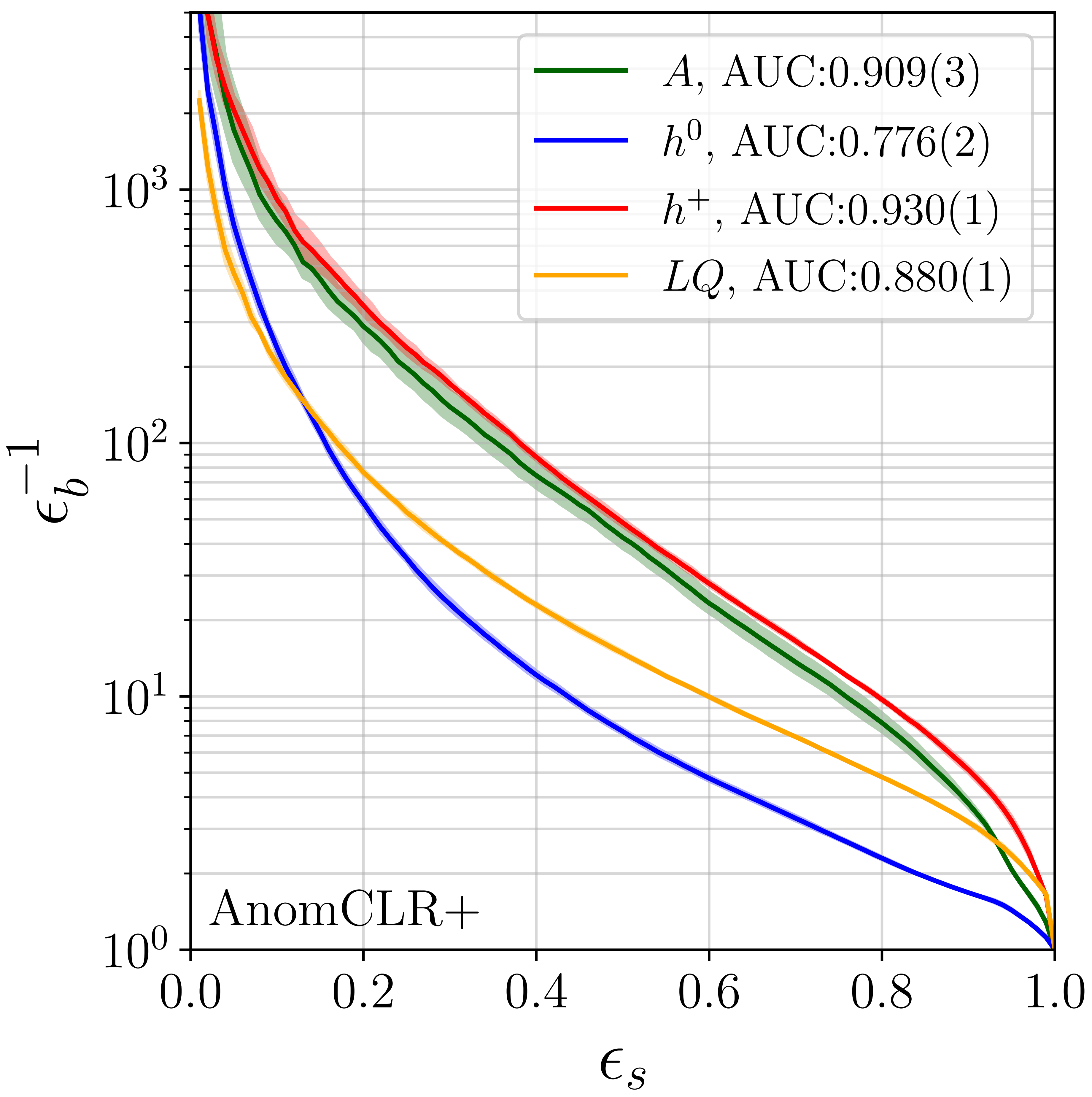
Each augmentation increase sensitivity to BSM-like features





Results: improved sensitivity





Conclusions/Outlook

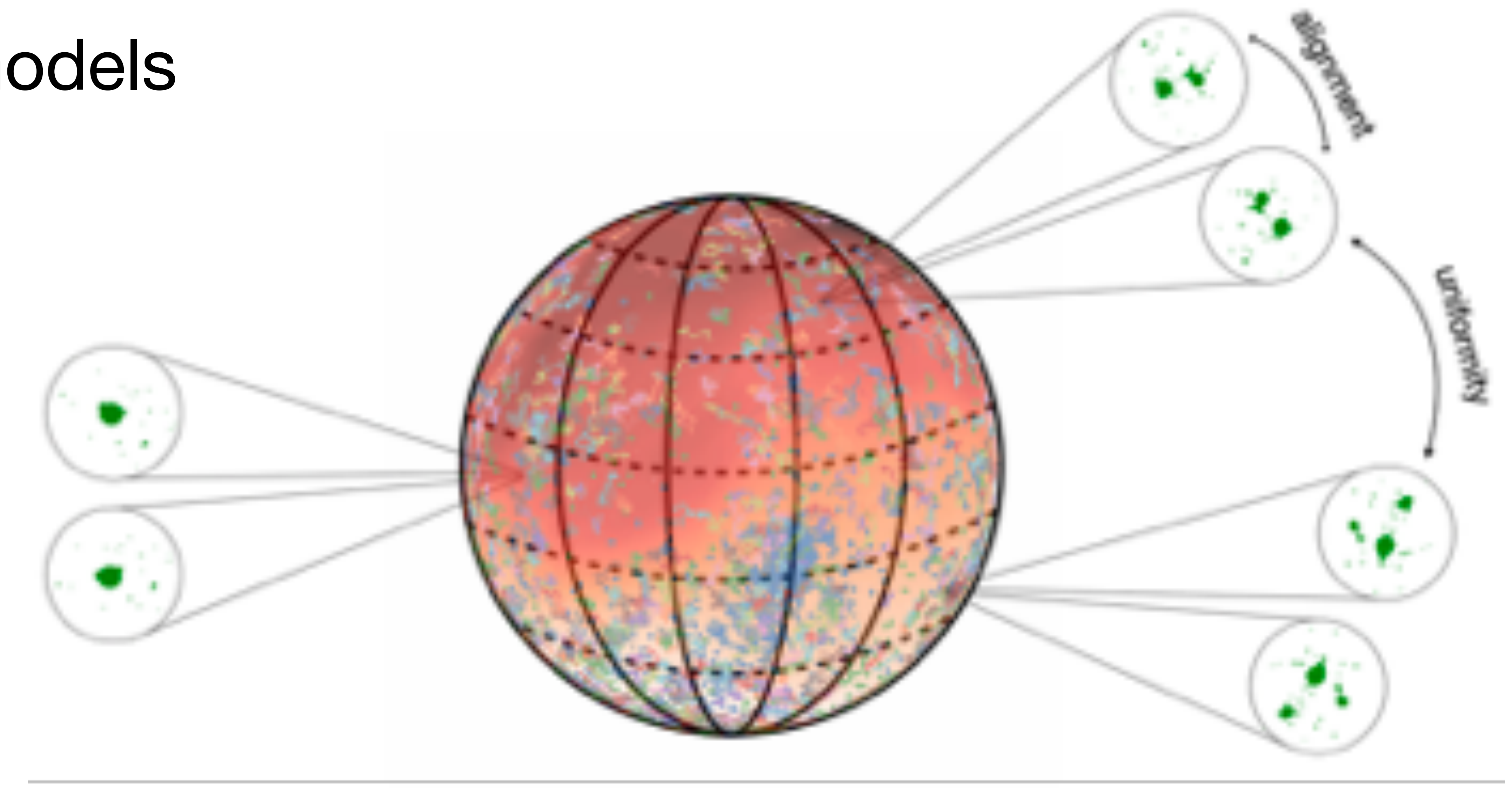
Auto-Encoders can be used for robust OOD detection
energy-based models are versatile tools used to learn prob. distributions

$\longrightarrow$ **Normalized Auto-Encoders (NAE)**

Self-supervision is a powerful tool to build representations

JetCLR and **AnomalyCLR** $\longrightarrow$ invariances, and high discriminative power

They can be paraphrased as **Contrastive Learning** (CLR) models



Next steps

- Combine them for improved discriminative power
- NAE for trigger applications
- Contrastive learning for semi visible jets
- …what about non-contrastive learning techniques?

Thanks for your attention!

Backup

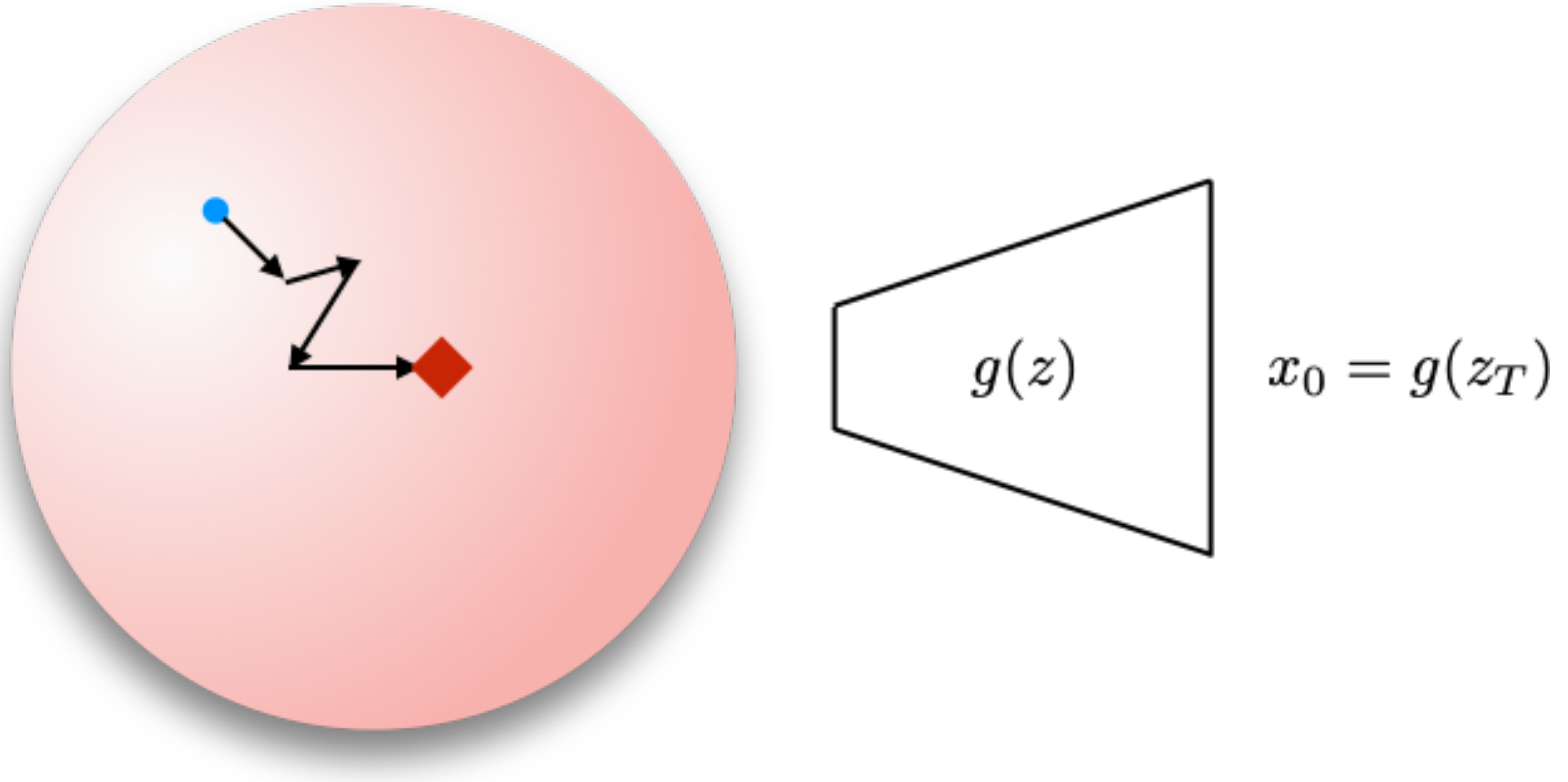
Sampling from the model

- Sampling is done via Metropolis-Adjusted Langevin* (MALA) Markov chains;
- given the dimensionality of the input space the initialization of the MCMC do matter:

On-Manifold Initialization → use latent space information

Latent space chains are defined by On-Manifold distribution and On-Manifold energy:

$$z_{t+1} = z_t + \lambda_t \nabla_z \log q_\theta(z) + \sigma_t \epsilon \qquad \epsilon \sim \mathcal{N}(0,1)$$



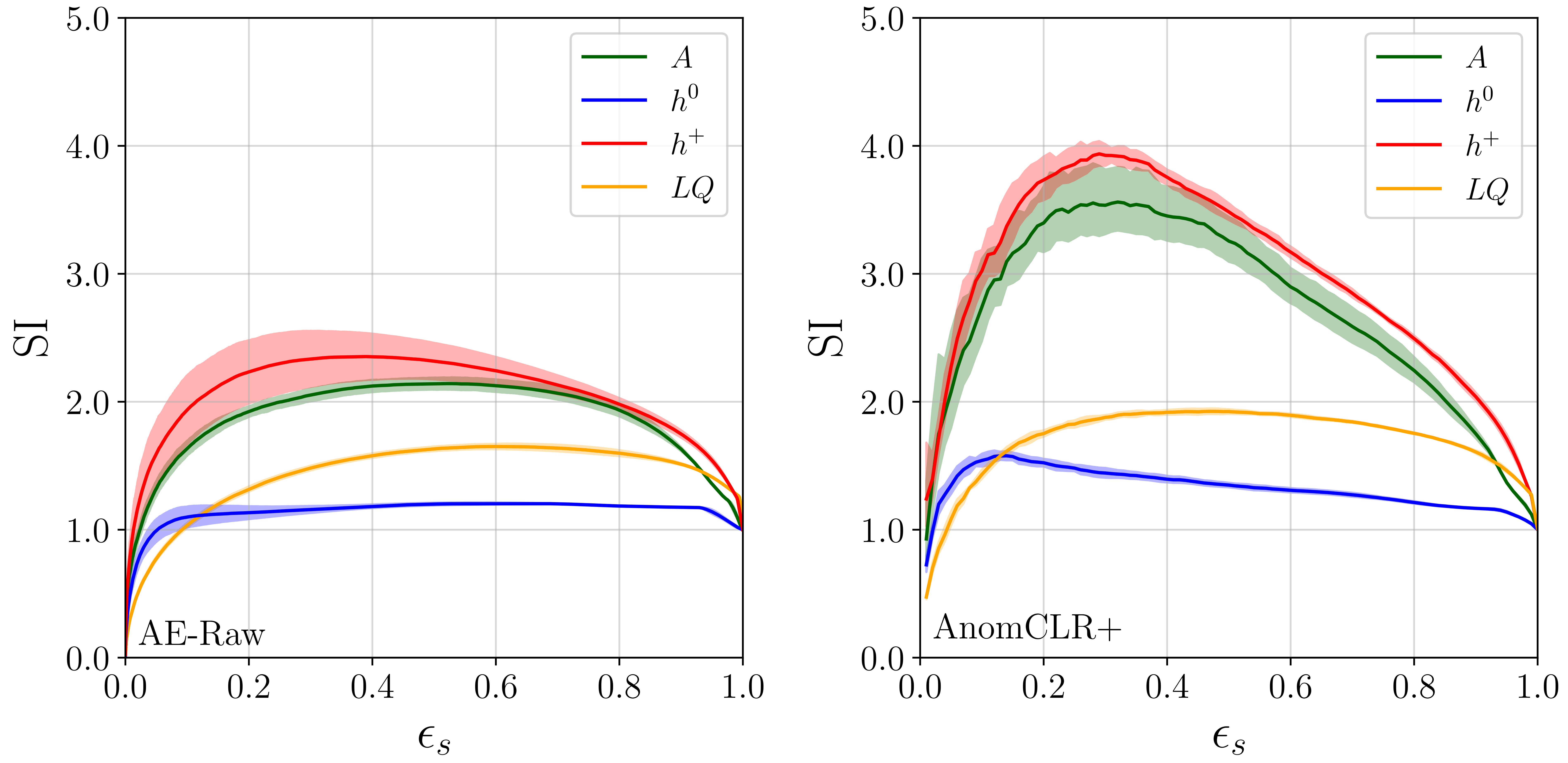
On-manifold distribution: $$q_\theta(z) = \frac{e^{H_\theta(z)}}{\Psi}$$

On-manifold energy: $$H_\theta(z) = E_\theta(g(z))$$

JetCLR performance

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

Results: SIC CURVES



Effect of anomalous augmentations