Project B3b:
Anomaly searches in jet physics

Luigi Favaro

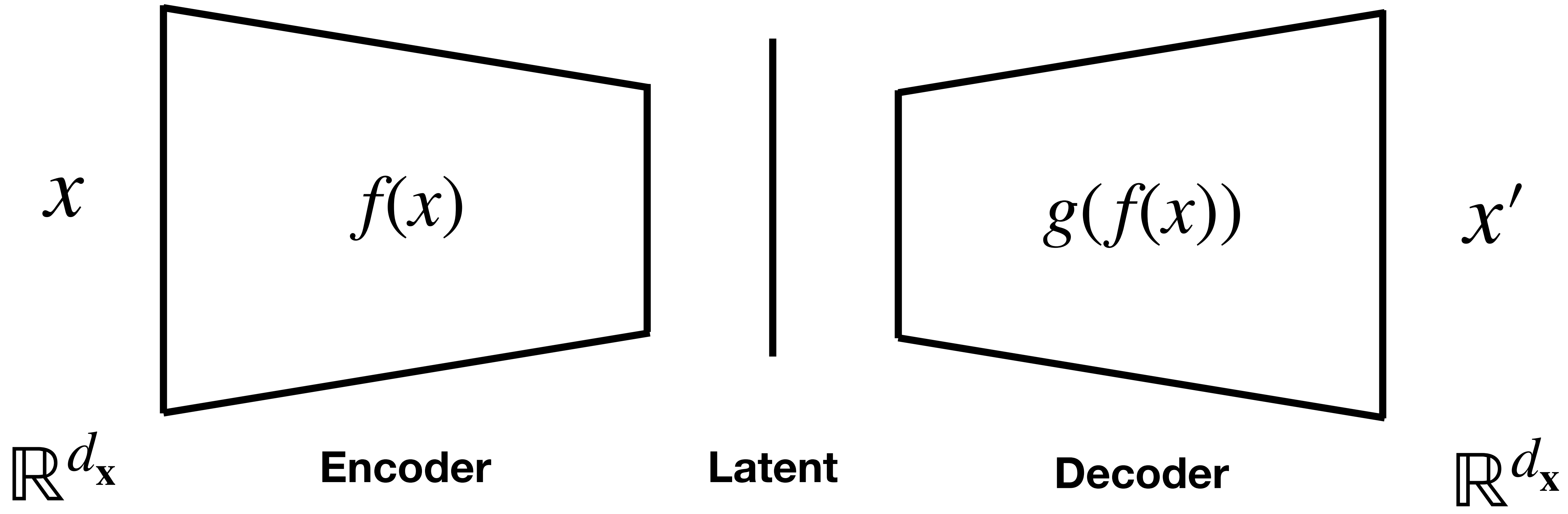
CRC annual meeting - Aachen 01/03/2023
Based on “A Normalized AutoEncoder for LHC triggers” - Dillon, Favaro, Krämer, Plehn, Sorrenson

UNIVERSITÄT HEIDELBERG
Zukunft. Seit 1386.

Model-agnostic searches & ML

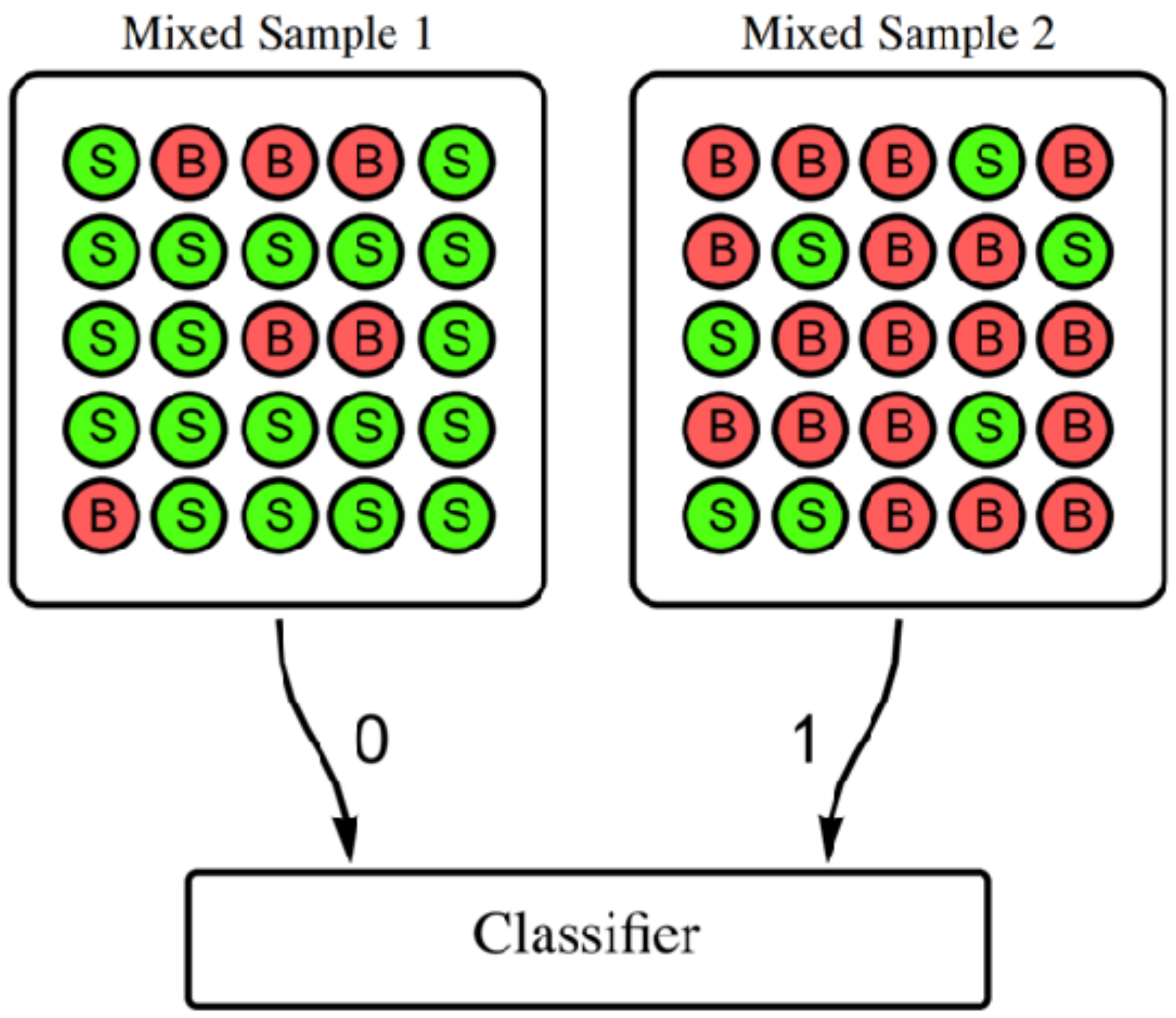
Two big families:

Autoencoders (AE)



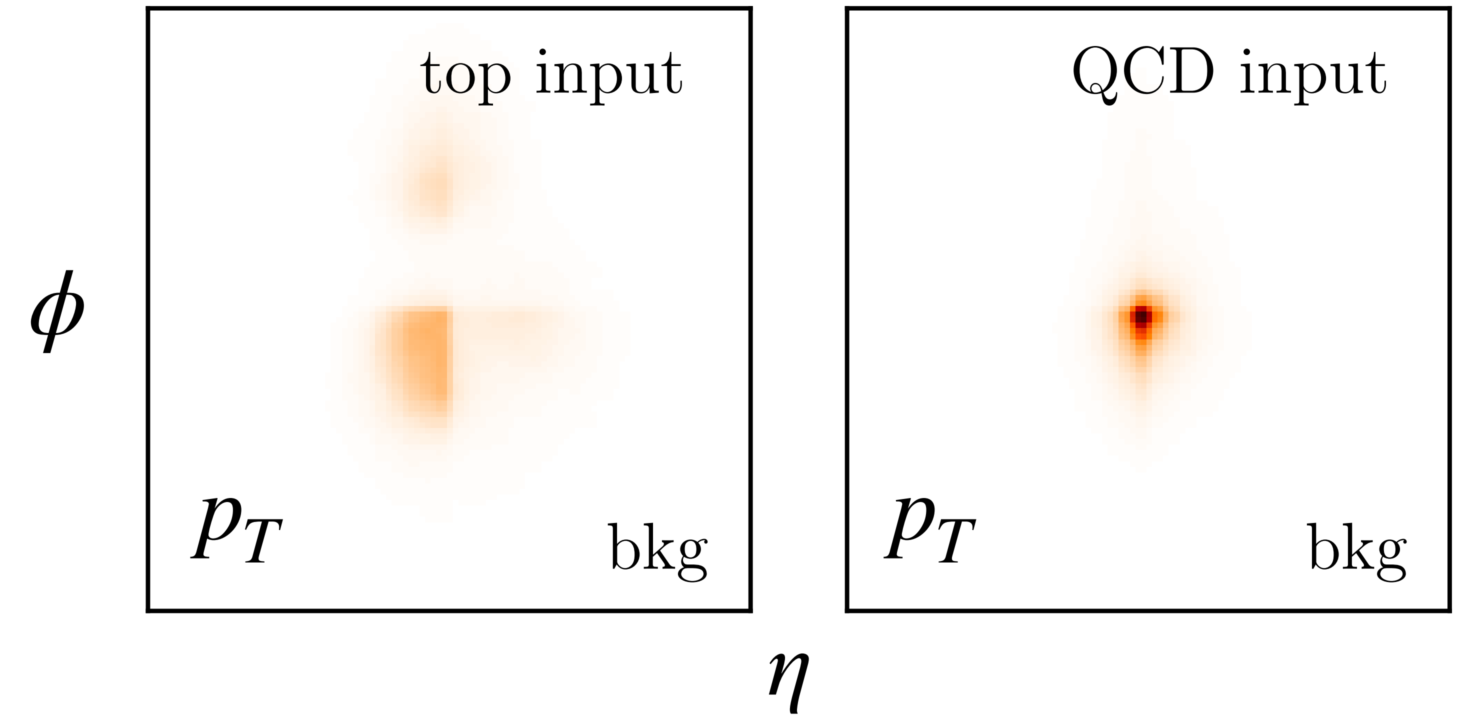
Anomaly score: MSE (x, x')

Classification without labels (CWOLA)



Defining a new observable

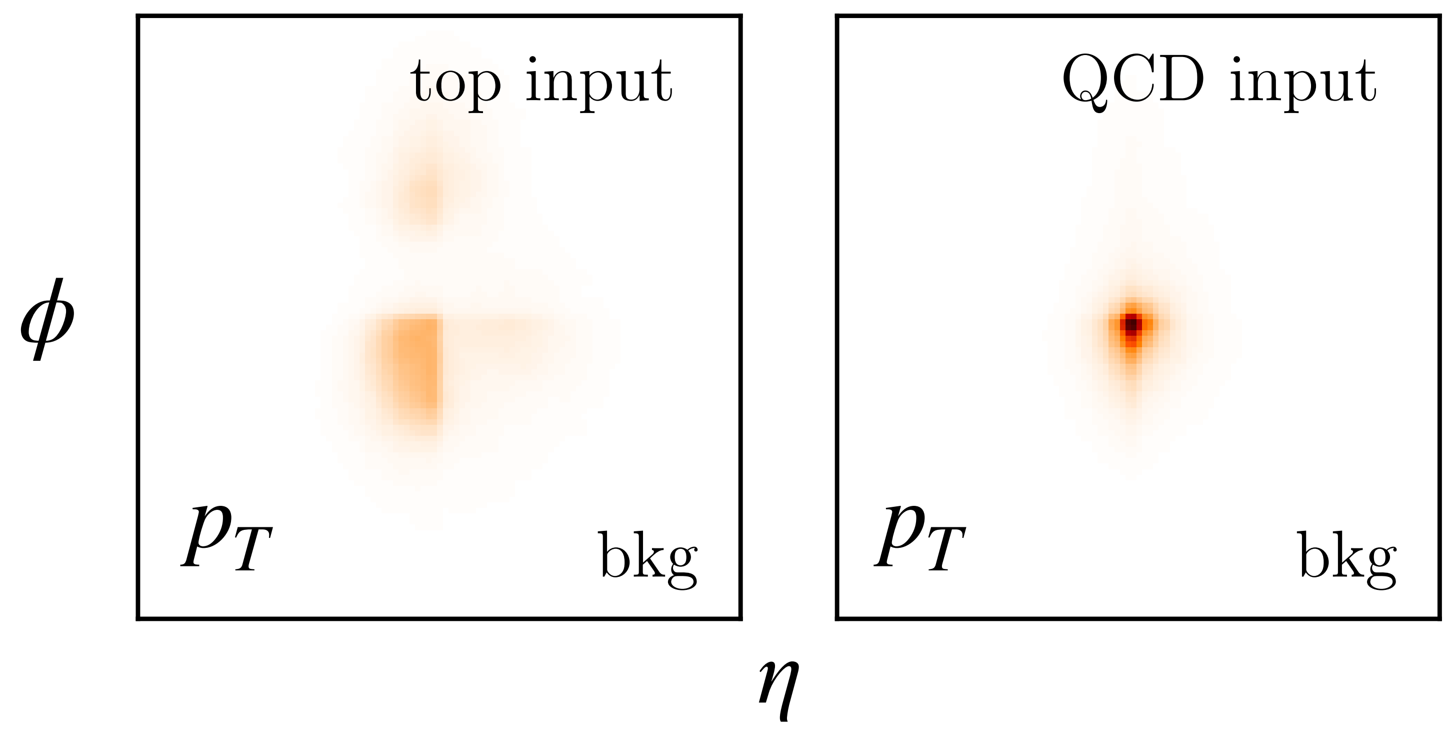
Machine Learning oriented observable for jet tagging: $p_\theta(x)$



Defining a new observable

Machine Learning oriented observable for jet tagging: $p_\theta(x)$

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



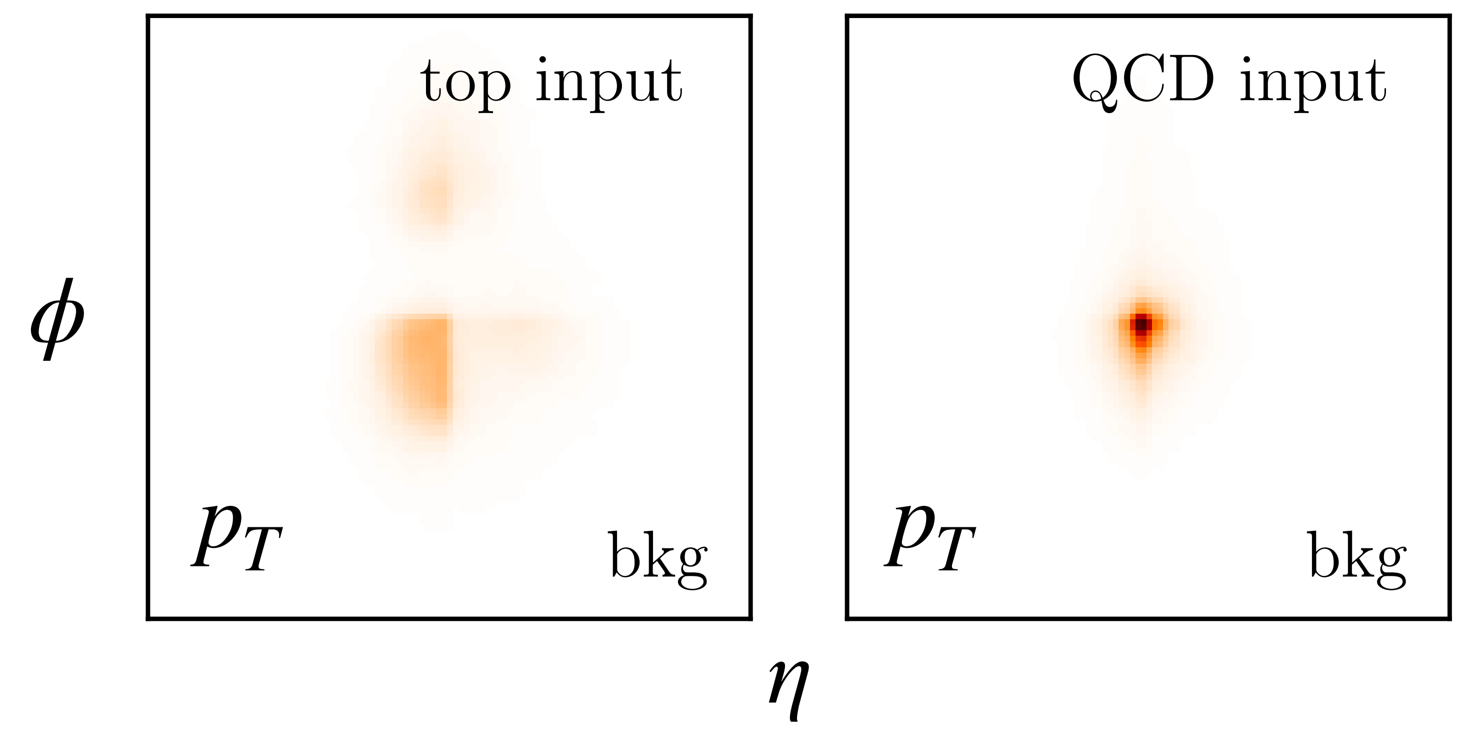
Defining a new observable

Machine Learning oriented observable for jet tagging: $p_\theta(x)$

- 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



Defining a new observable

Machine Learning oriented observable for jet tagging: $p_\theta(x)$

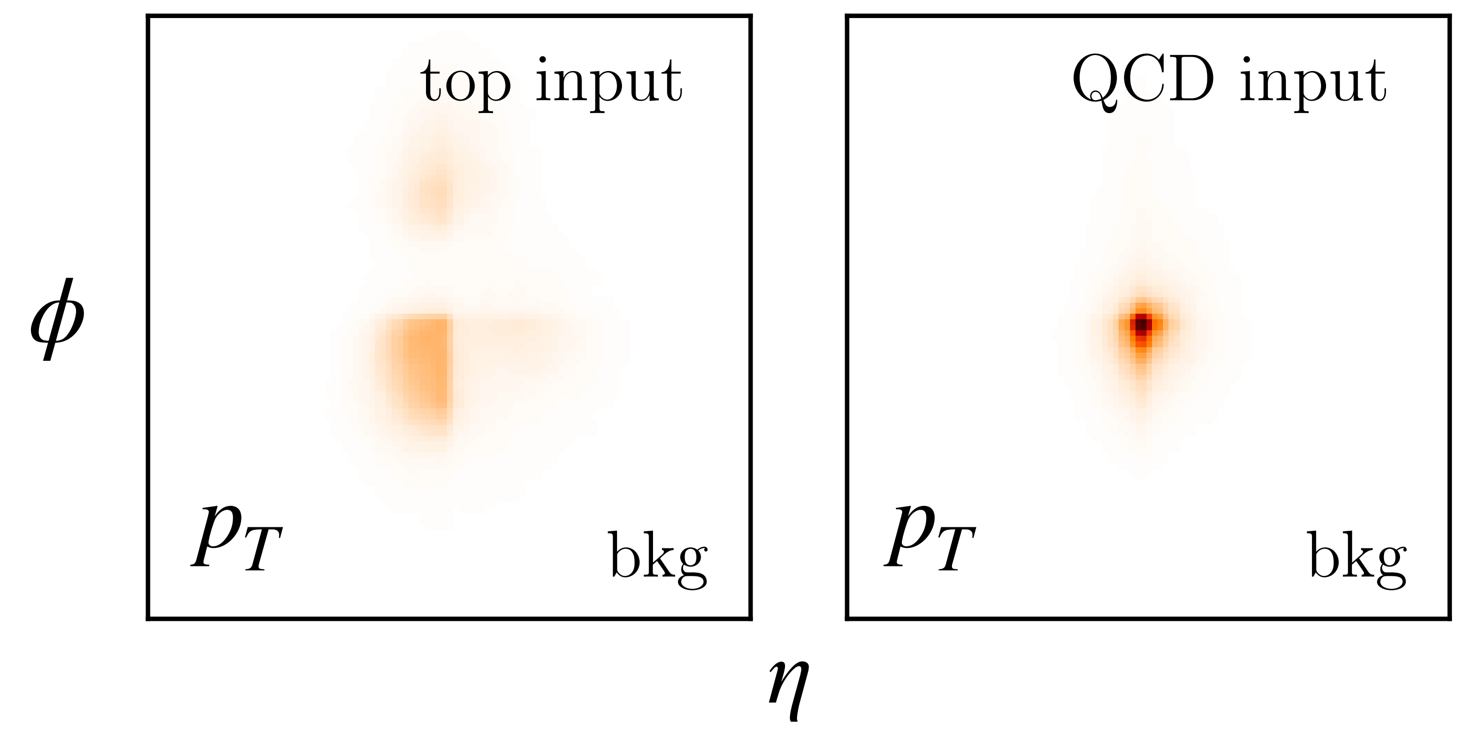
- Auto-Encoders can easily tag complex signals;
- the opposite is not generally true → 'complexity bias'

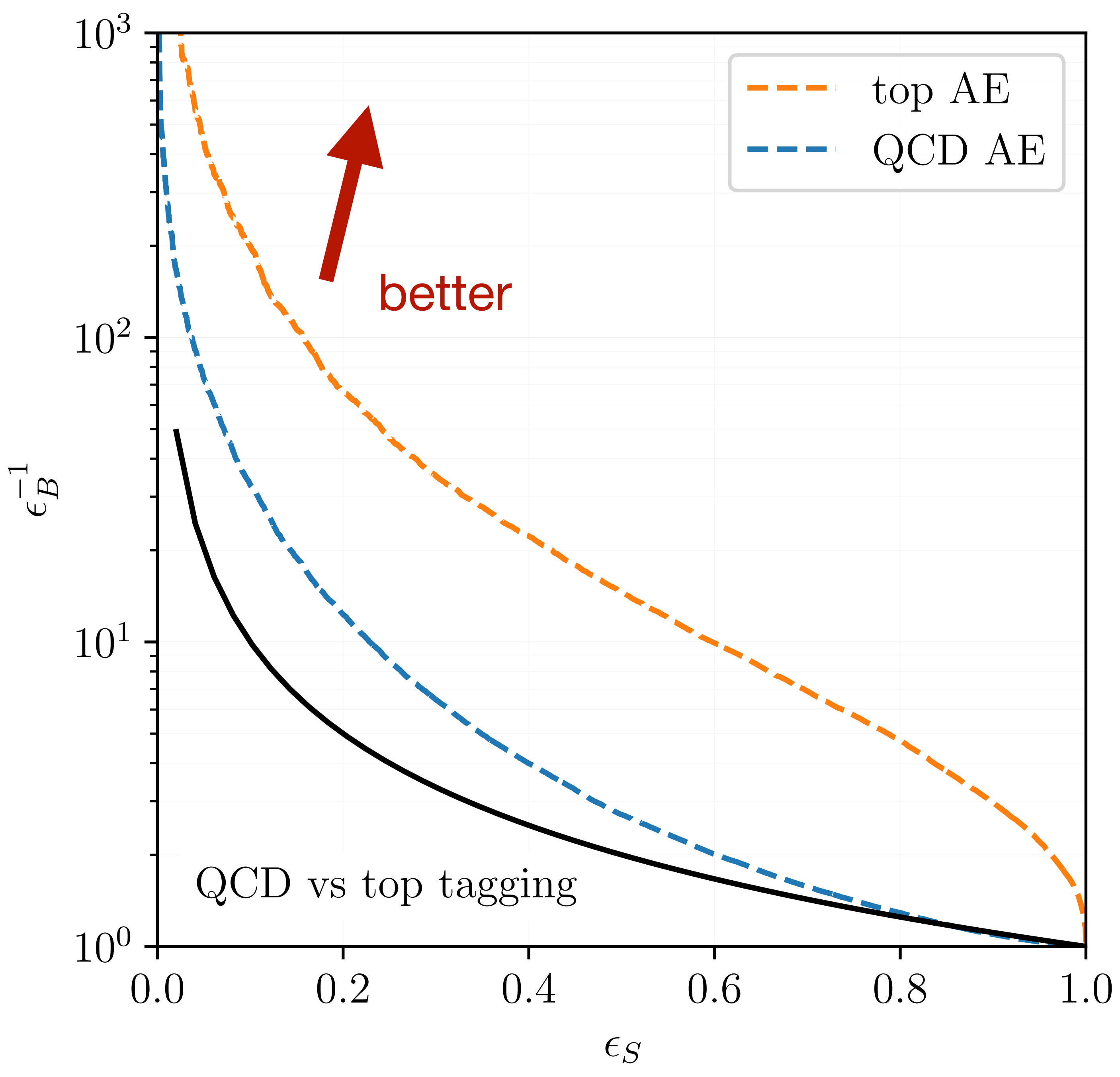
Robustness test: inverse training

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

Example: QCD tagging

Expected phase space regions with only QCD jets, see e.g. τ_3/τ_2





Contrastive Learning approach

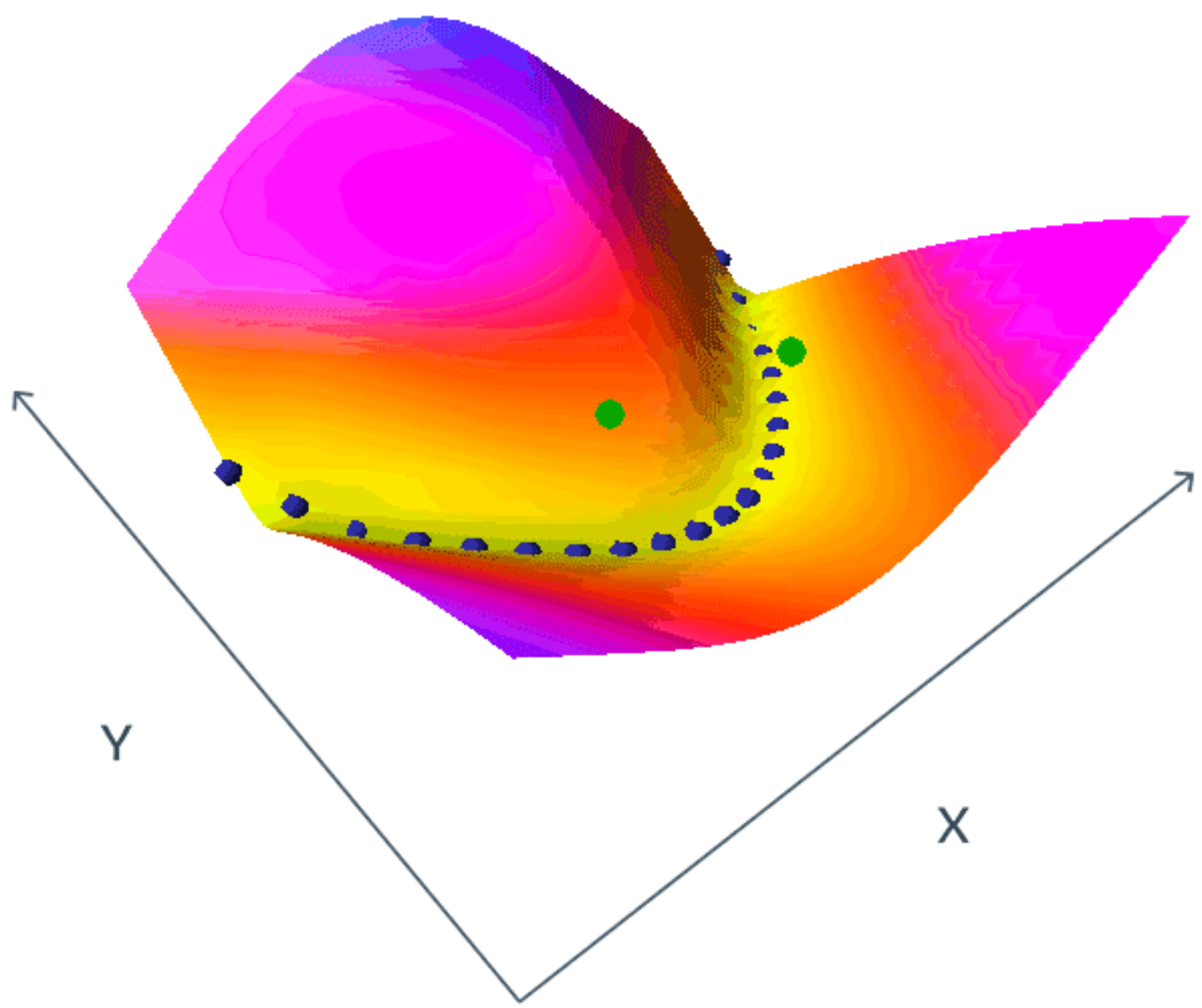
An interesting approach to the problem is **Contrastive Learning** (CLR):

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

Contrastive Learning approach

An interesting approach to the problem is **Contrastive Learning** (CLR):

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

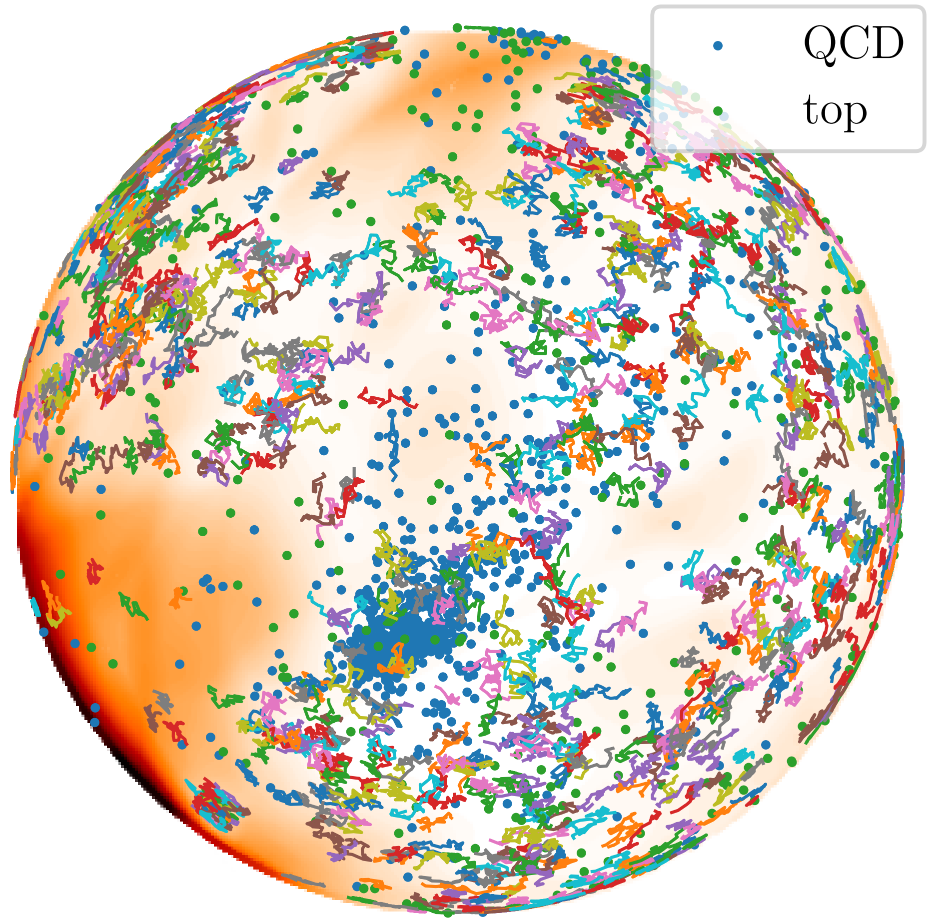


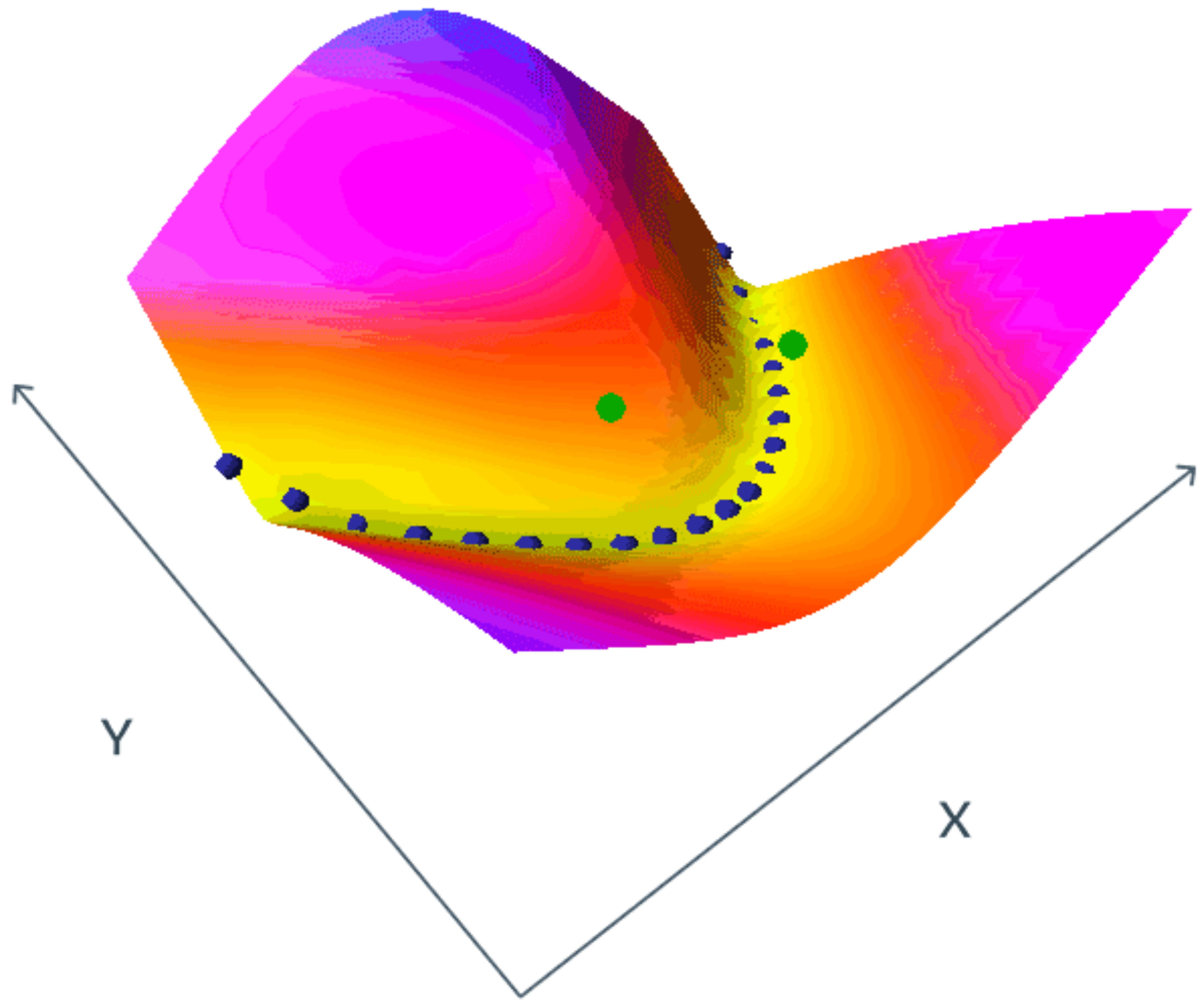
Contrastive Learning approach

An interesting approach to the problem is **Contrastive Learning** (CLR):

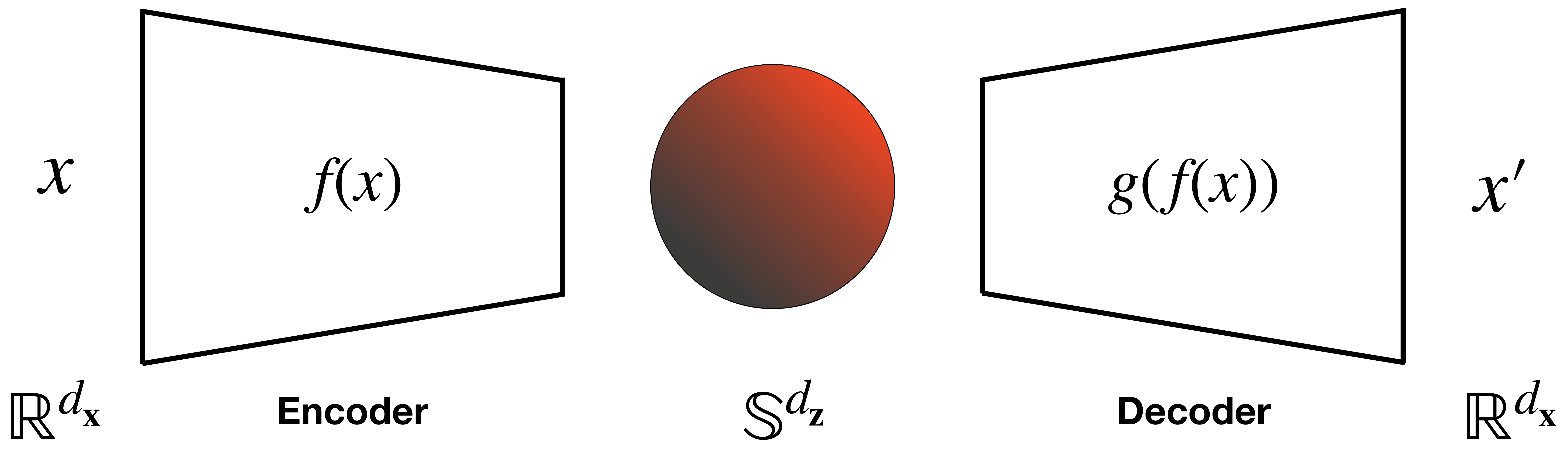
- phrase the objective loss as a contrastive loss with
  - **positive samples**
  - **negative samples**
- shape a non-degenerate energy landscape

Normalized Auto-Encoders





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**

[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

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: $p_\theta(x) = \dfrac{e^{-E_\theta(x)}}{\Omega}$

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

[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

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: $p_\theta(x) = \dfrac{e^{-E_\theta(x)}}{\Omega}$

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

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:

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

Training a NAE

Consider the gradients of the loss function:

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

Minimizes the usual AE reconstruction error;

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

Training a NAE

Consider the gradients of the loss function:

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

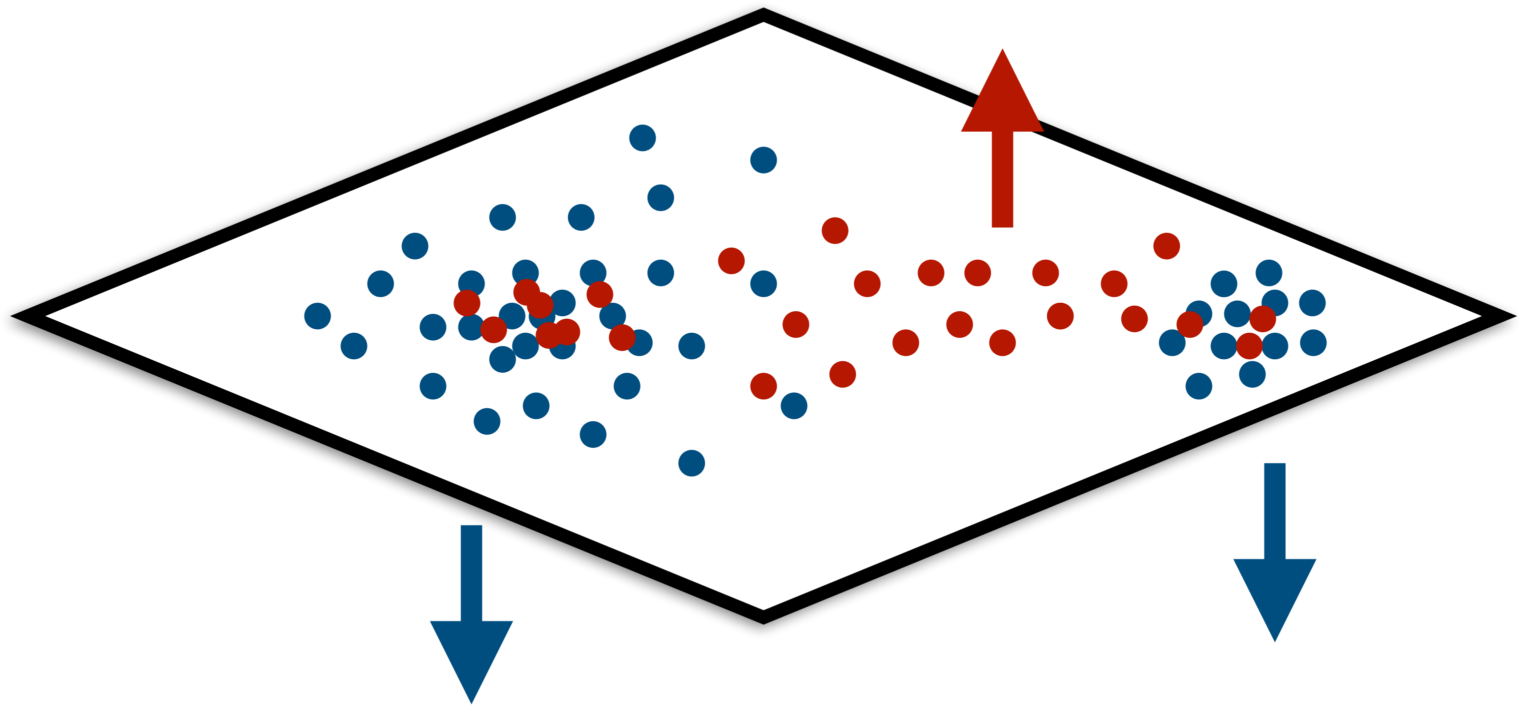
Minimizes the usual AE reconstruction error;

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



Training a NAE

Consider the gradients of the loss function:

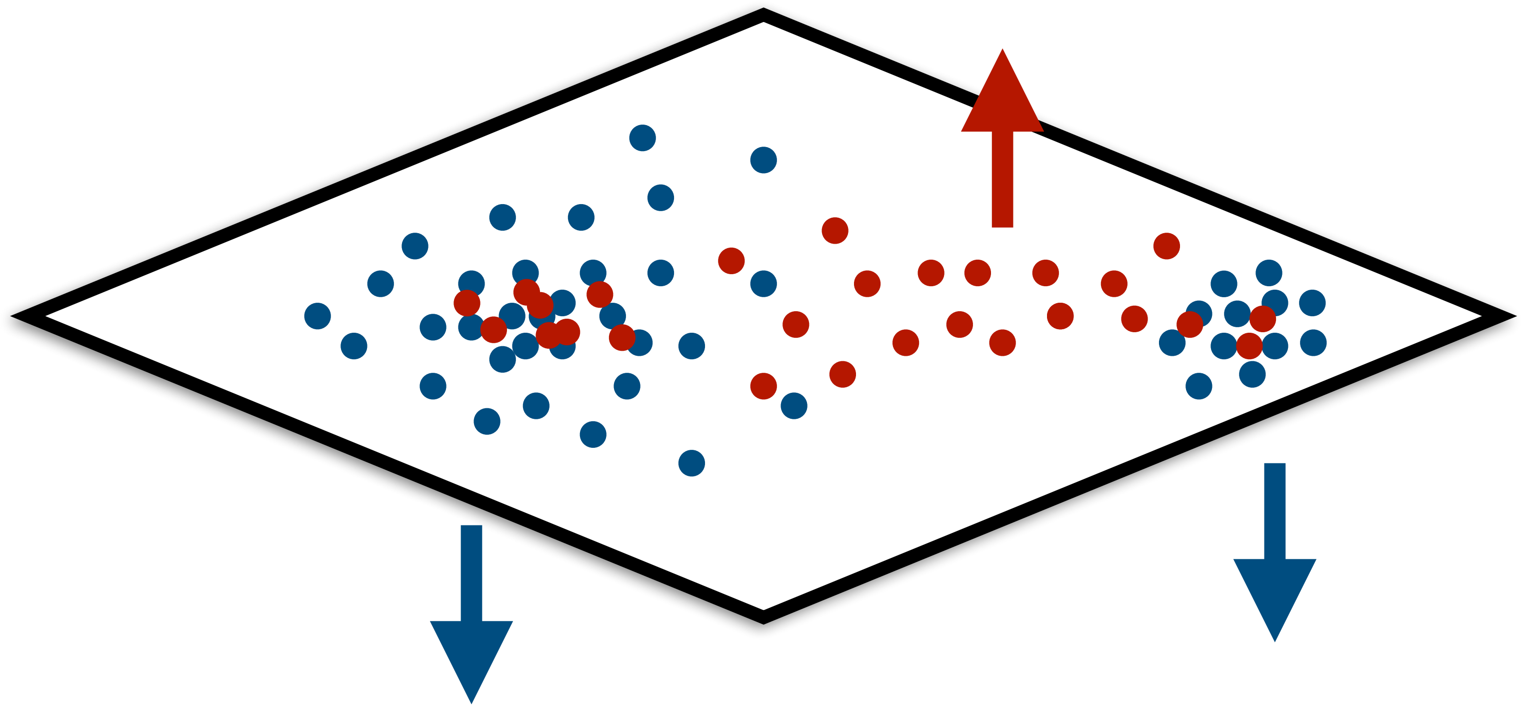
$$\nabla_\theta \mathscr{L} = \nabla_\theta E_\theta(x) + \nabla_\theta \log \Omega$$

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?

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***?***

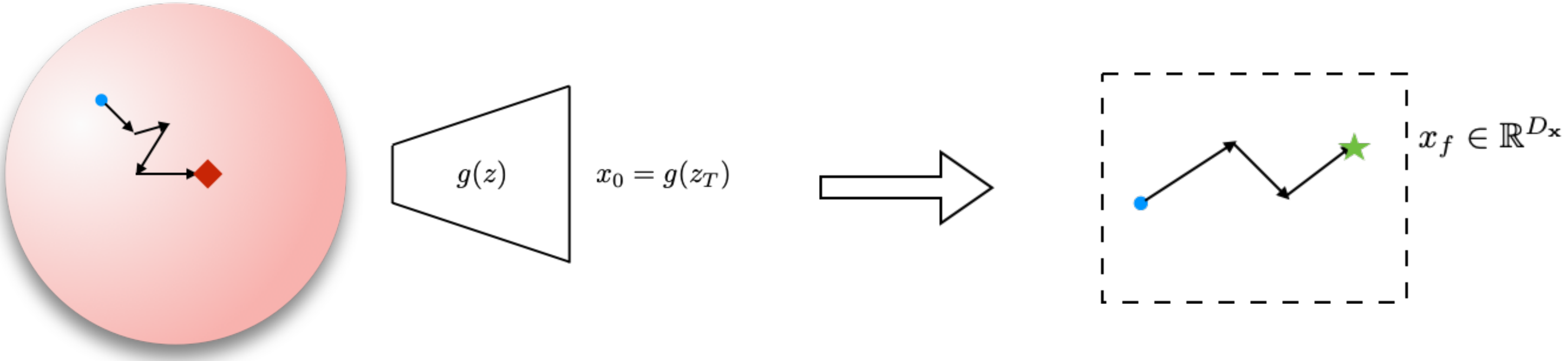
Sampling from p_θ

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 $p_\theta(x)$.

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

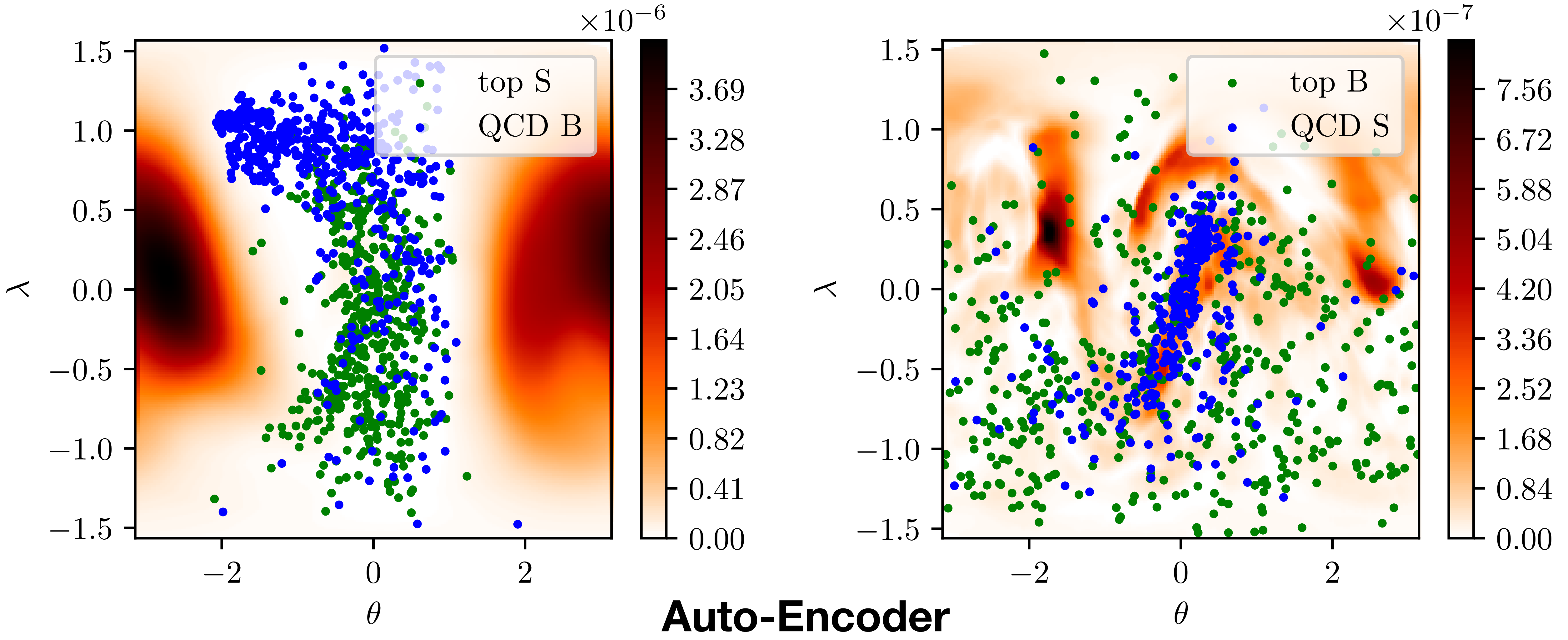
* small number of steps $\mathcal{O}(100)$, constrained into low energy regions by taking $\lambda > \sigma$



Results: decoder manifold

We can study what happens during training:

- 2D projection of the latent space;
- decoder manifold for tops is more complex

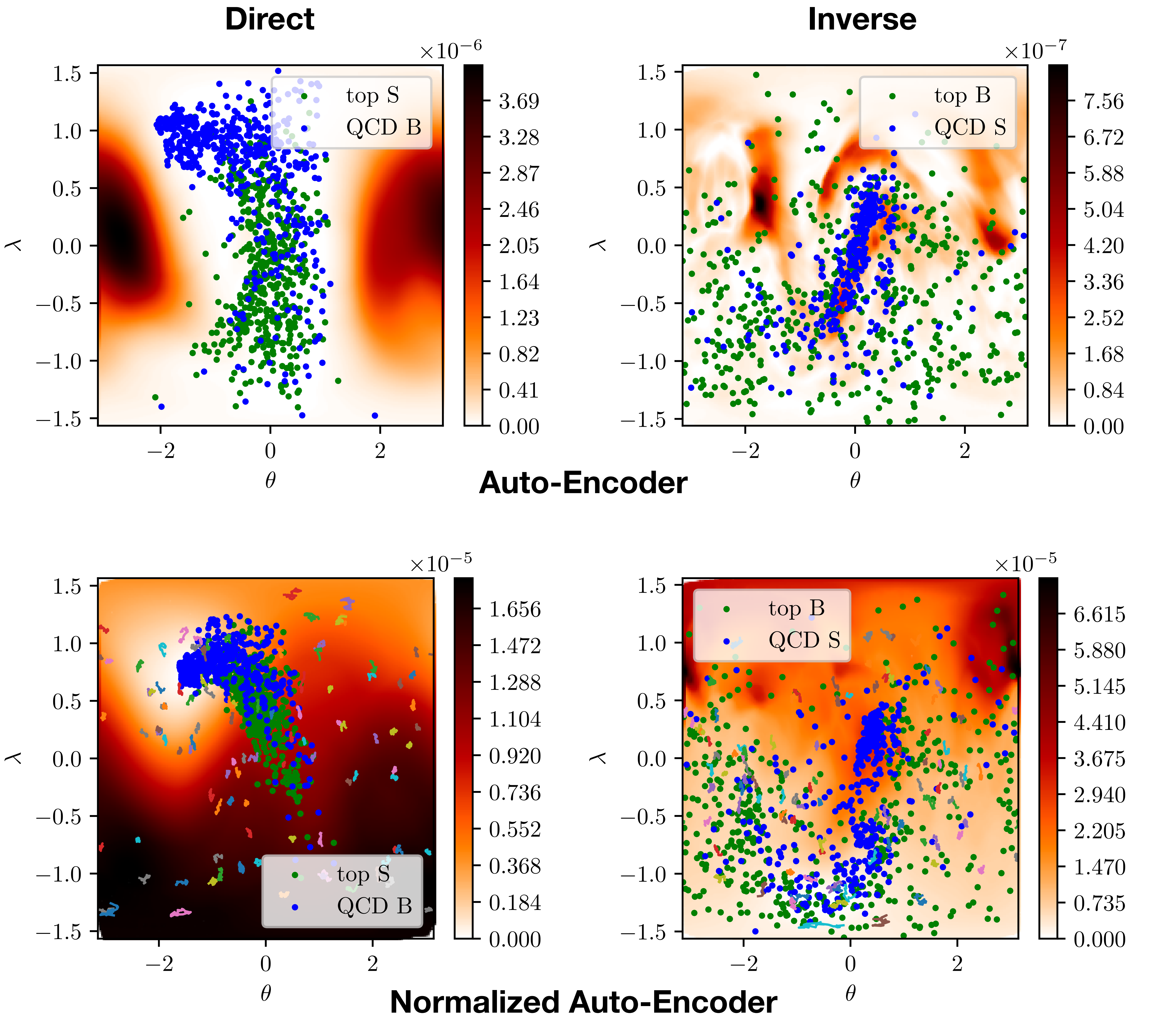


Auto-Encoder

Results: decoder manifold

We can study what happens during training:

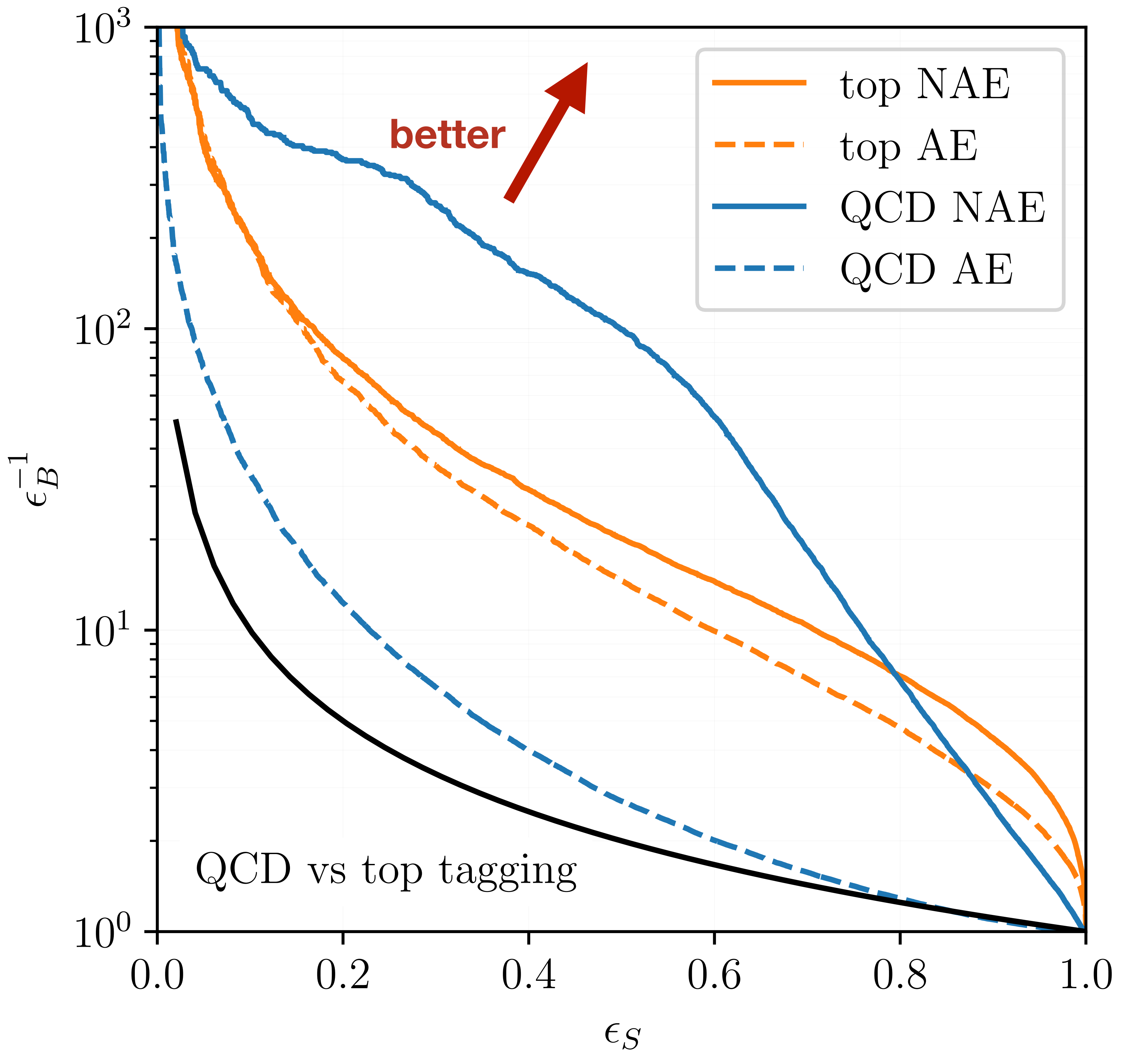
- 2D projection of the latent space;
- decoder manifold for tops is more complex
- inducing an underlying metric via $\log \Omega$;
- after training both QCD and top jets are mapped in high reconstruction regions of the decoder manifold;



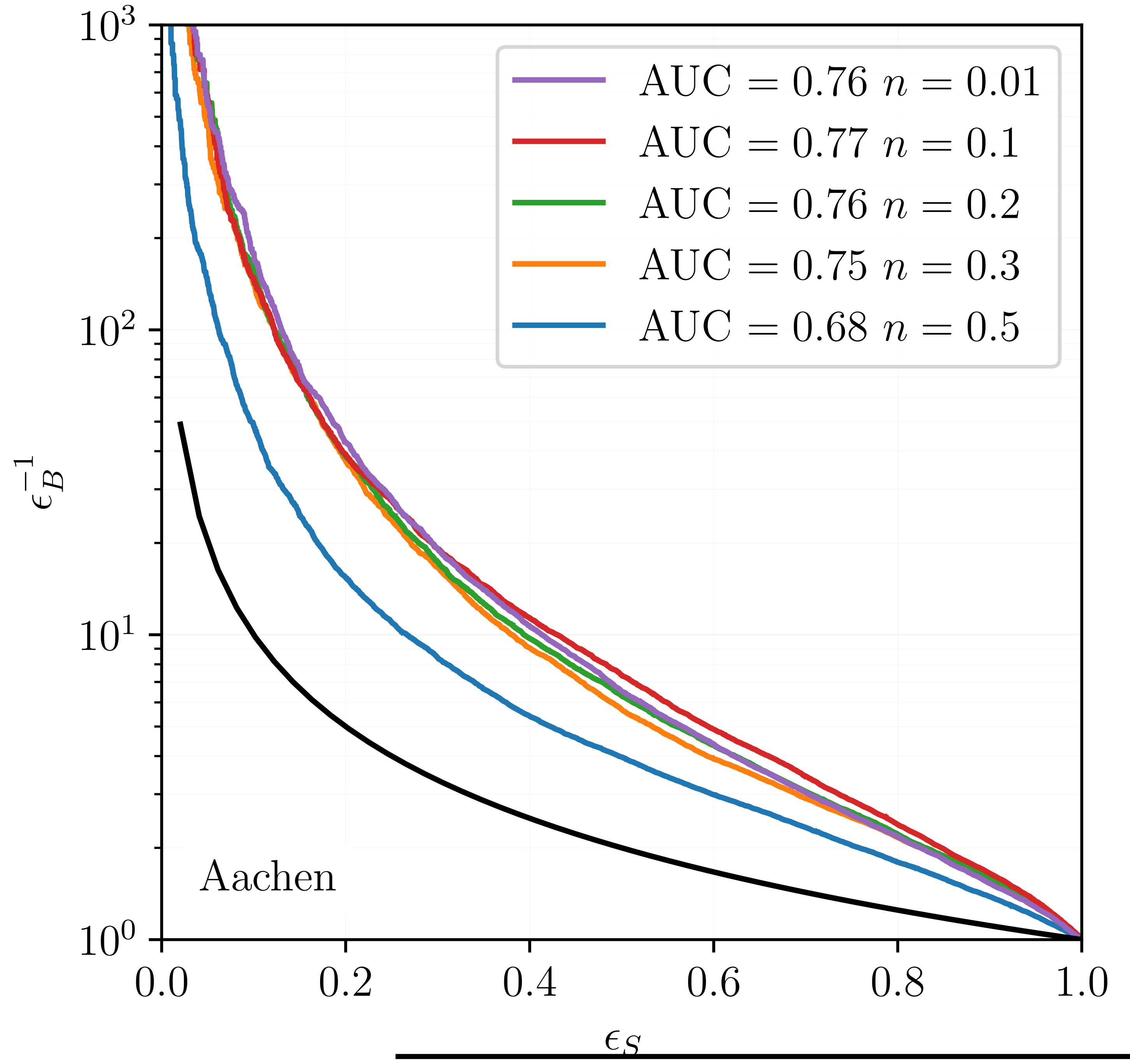
Results: QCD vs top tagging

- 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.

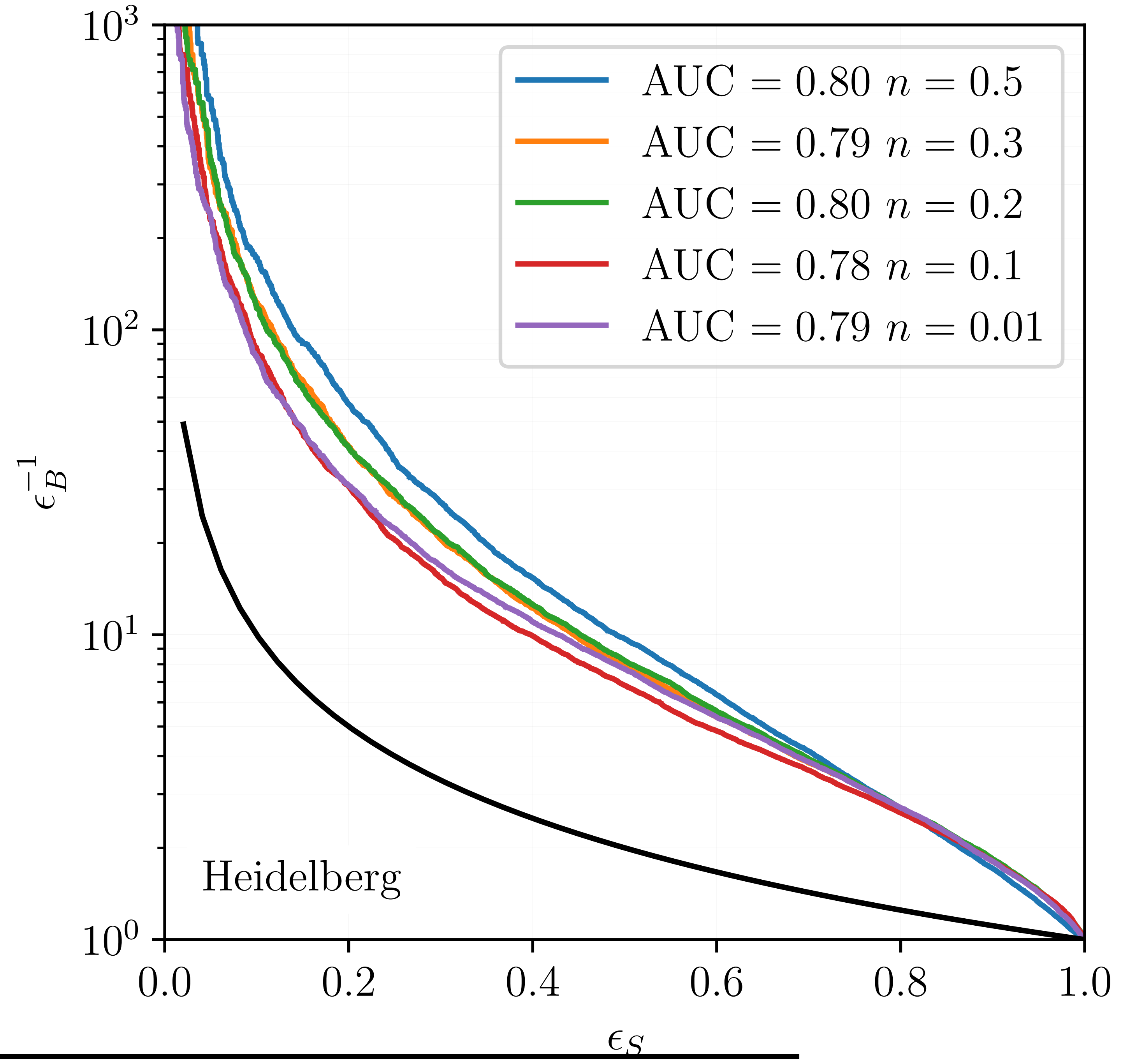
Signal	NAE		AE [1]	DVAE [6]
	AUC	$\epsilon_B^{-1}(\epsilon_S = 0.2)$	AUC	AUC
top (AE)	0.875	68	0.89	0.87
top (NAE)	0.91	80		
QCD (AE)	0.579	12	–	0.75
QCD (NAE)	0.89	350		



Results: BSM signals



small dependence on the implicit bias!



Data		n 0.5	0.3	0.2	0.1	0.01
Heidelberg	AUC	0.795 (5)	0.796 (5)	0.789 (8)	0.78 (1)	0.790 (5)
	$\epsilon_B^{-1}(\epsilon_S = 0.2)$	62 (3)	42 (5)	42 (4)	28 (4)	30 (1)
Aachen	AUC	0.68 (1)	0.746 (5)	0.75 (1)	0.767 (5)	0.755 (5)
	$\epsilon_B^{-1}(\epsilon_S = 0.2)$	15 (1)	38 (3)	33 (7)	41 (2)	41 (1)

Looking forward: Self-Supervision

AutoEncoders not invariant to data preprocessing

[What's anomalous in LHC jets?, Buss, Dillon, Finke, Krämer et al. arXiv:2301.04660]

Looking forward: Self-Supervision

AutoEncoders not invariant to data preprocessing

[What’s anomalous in LHC jets?, Buss, Dillon, Finke, Krämer et al. arXiv:2301.04660]

Create a **representation space** highly discriminative towards anomalous features during training

Looking forward: Self-Supervision

AutoEncoders not invariant to data preprocessing

[What’s anomalous in LHC jets?, Buss, Dillon, Finke, Krämer et al. arXiv:2301.04660]

Create a **representation space** highly discriminative towards anomalous features during training

A **Contrastive Learning** approach to data representation: Self-supervision

- start from raw data (e.g. constituents)
- use pseudo-labels derived from data
- define observables as an optimization task
- invariant to symmetries
- highly discriminative

Looking forward: Self-Supervision

AutoEncoders not invariant to data preprocessing

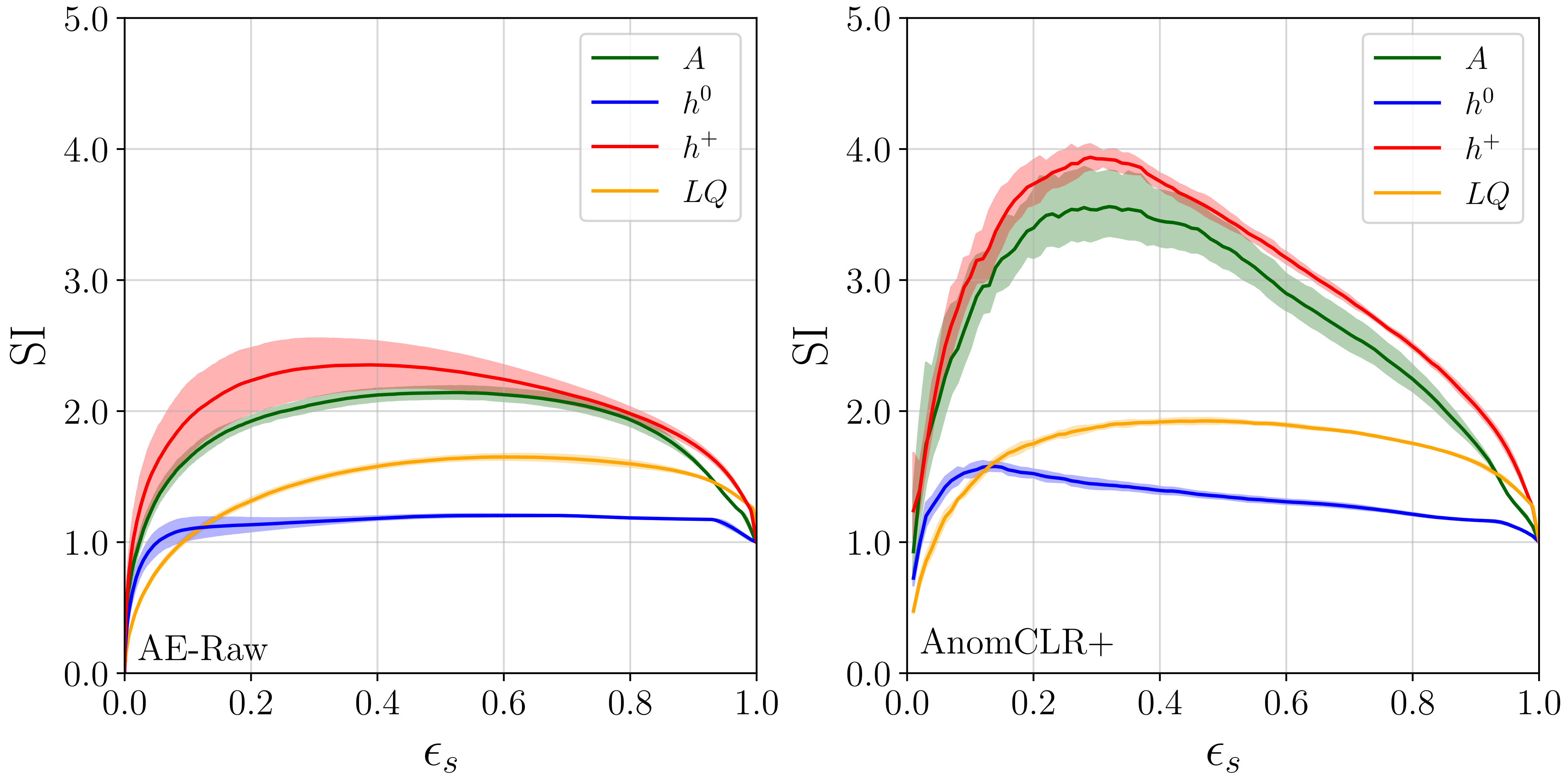
[What's anomalous in LHC jets?, Buss, Dillon, Finke, Krämer et al. arXiv:2301.04660]

Create a **representation space** highly discriminative towards anomalous features during training

A **Contrastive Learning** approach to data representation: Self-supervision

- start from raw data (e.g. constituents)
- use pseudo-labels derived from data
- define observables as an optimization task
- invariant to symmetries
- highly discriminative

AnomalyCLR



First application to reconstructed objects:

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

Outlook

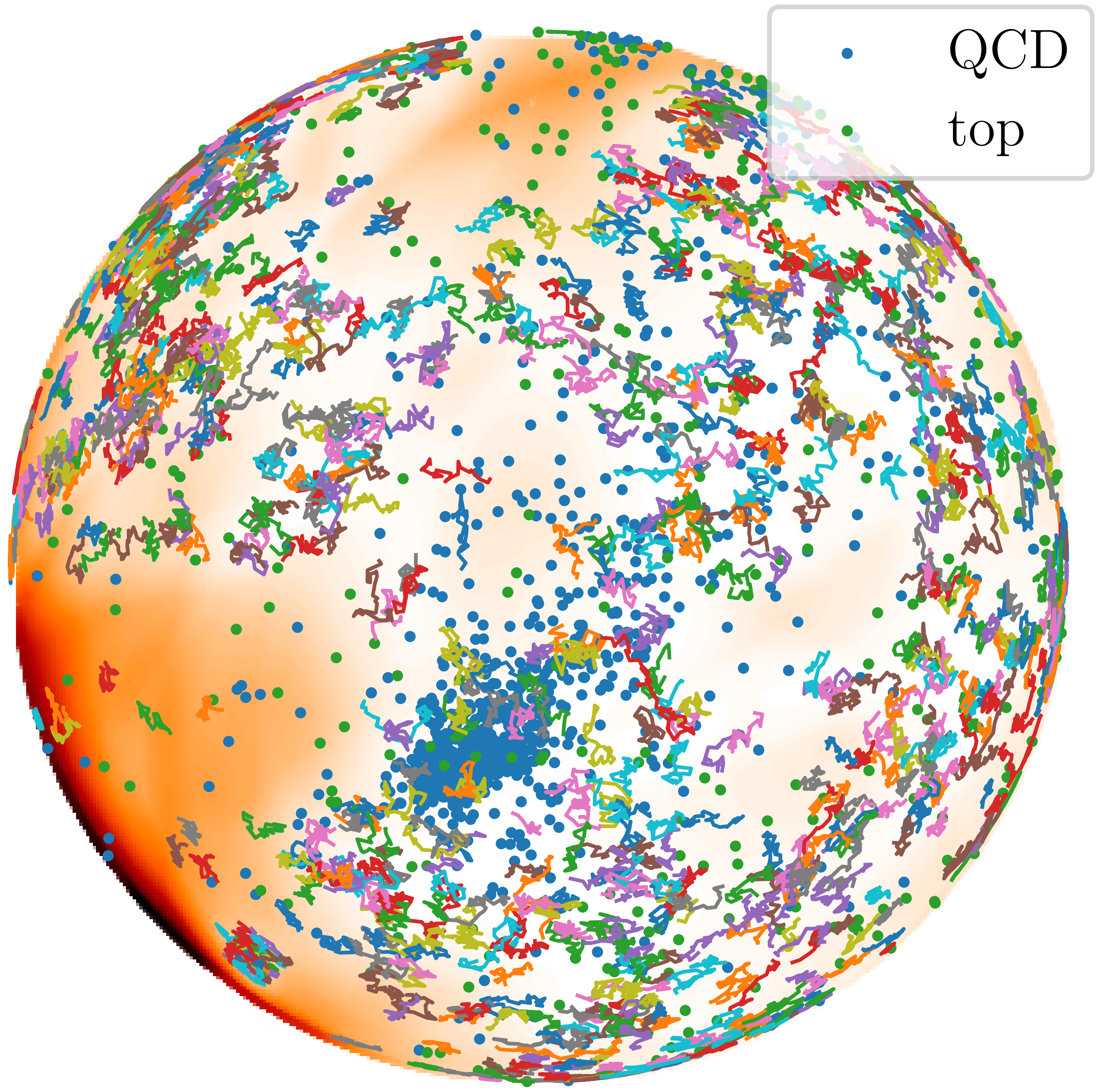
GitHub: heidelberg-hepml/normalized-autoencoders

Normalized Auto-Encoders allow for:

- an energy-based description of an Auto-Encoder
- robust auto-encoder for anomaly detection

  → no complexity bias
- reconstruction error and log-likelihood are directly related:

  → it could be used as a density estimation tool
- the dependence on an implicit bias is greatly reduced
- Example results: QCD vs top and BSM jet tagging



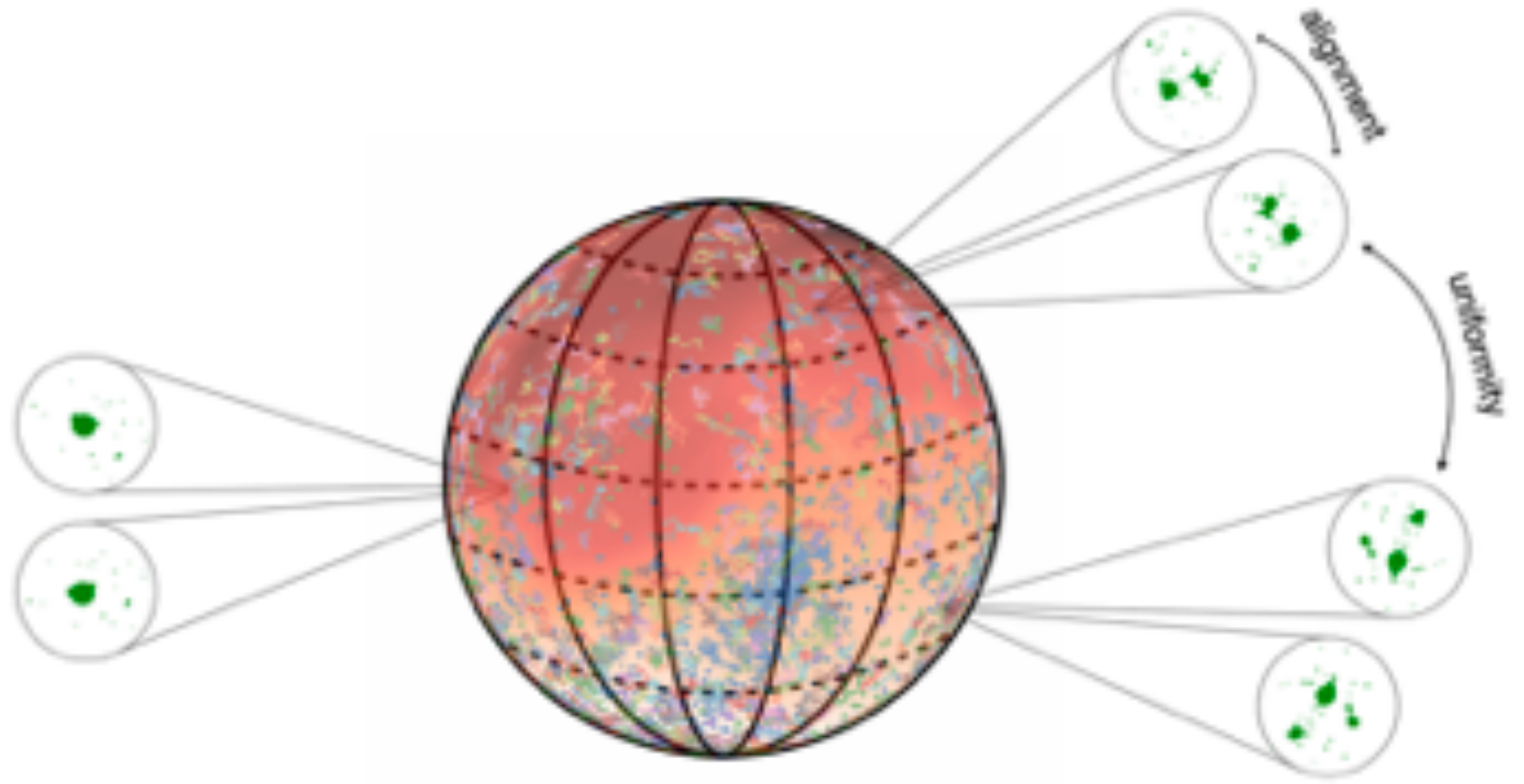
Future directions:

- Benchmarking NAEs for trigger applications
- Introduce self-supervision paradigm in NAE training

Outlook

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
- Recently demonstrated at events level (ADC 2021)



Future directions:

- Self-supervision for semi-visible jet tagging
- Robust estimation of $p(x)$ in the representation space

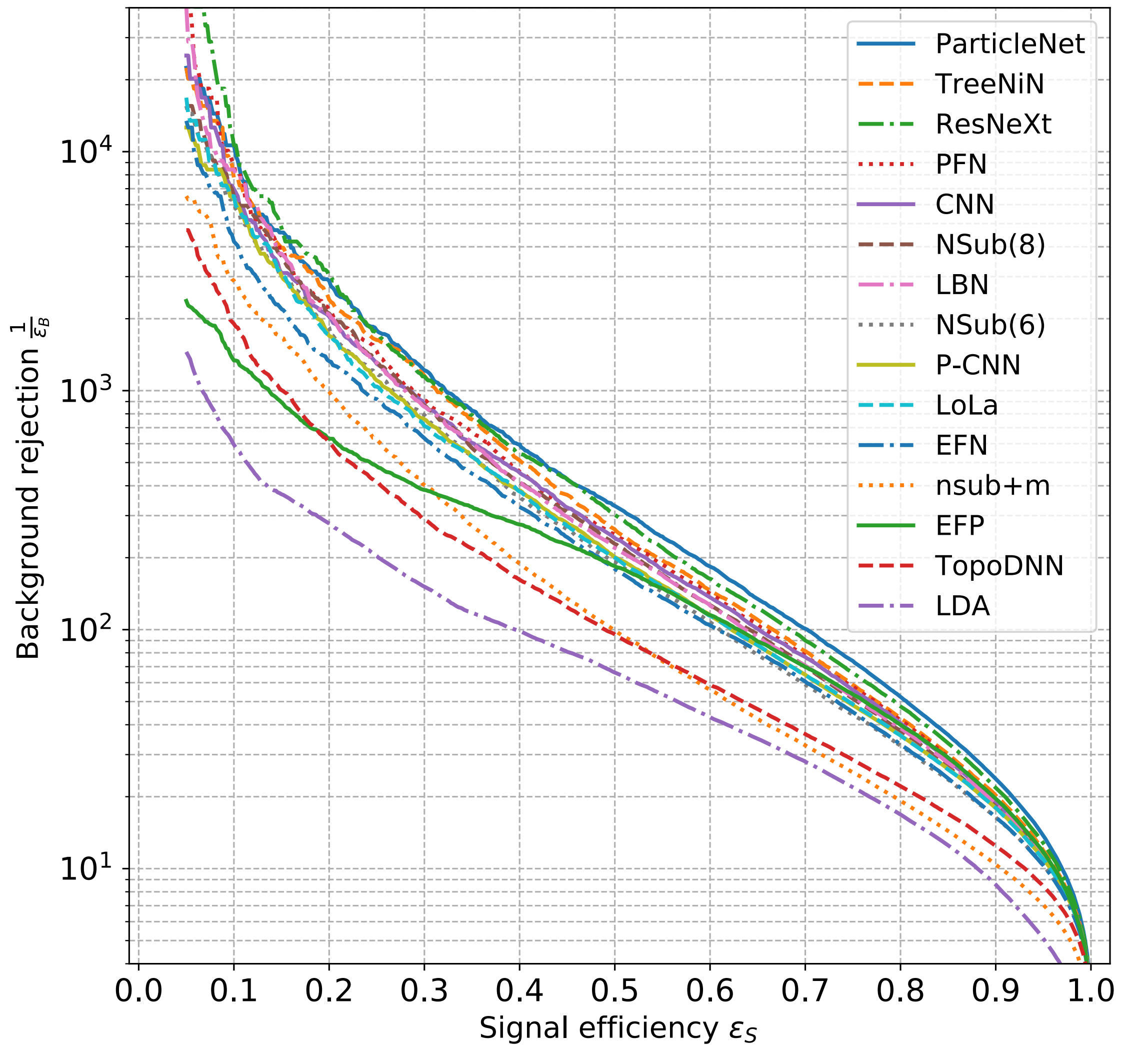
Thanks for your attention!

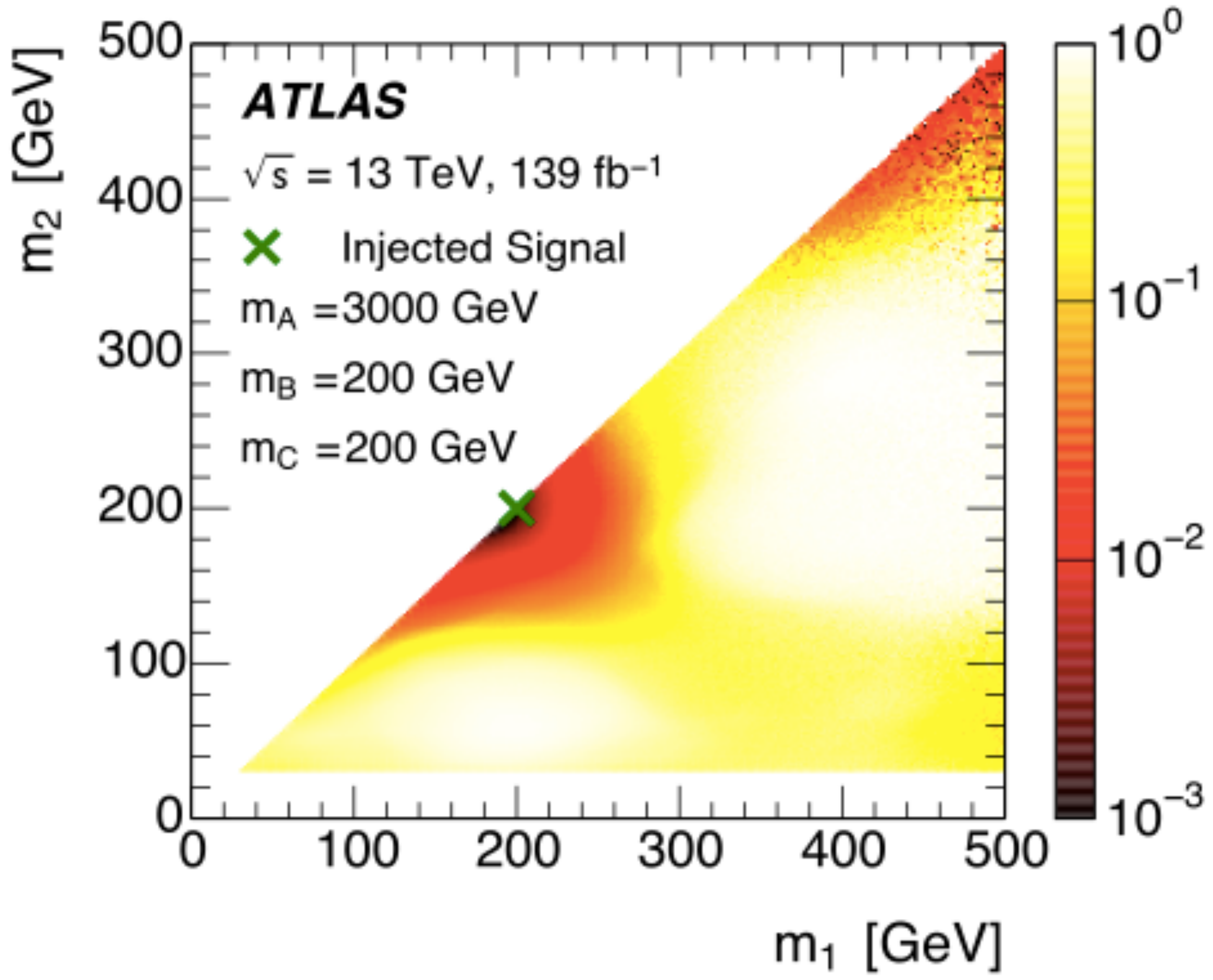
Model-agnostic searches & ML

- 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, …)





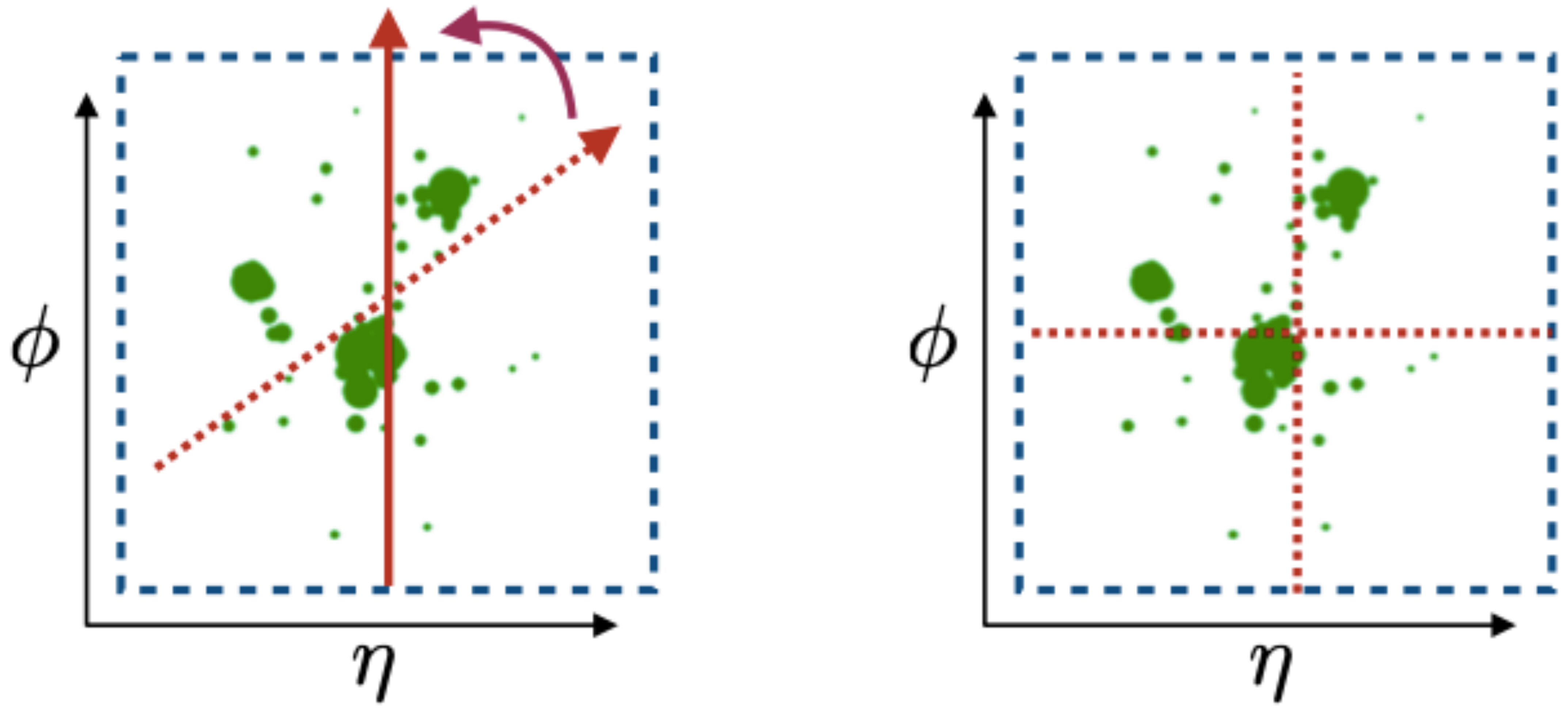
Already many interesting challenges/applications of ML techniques

Images for Jet tagging

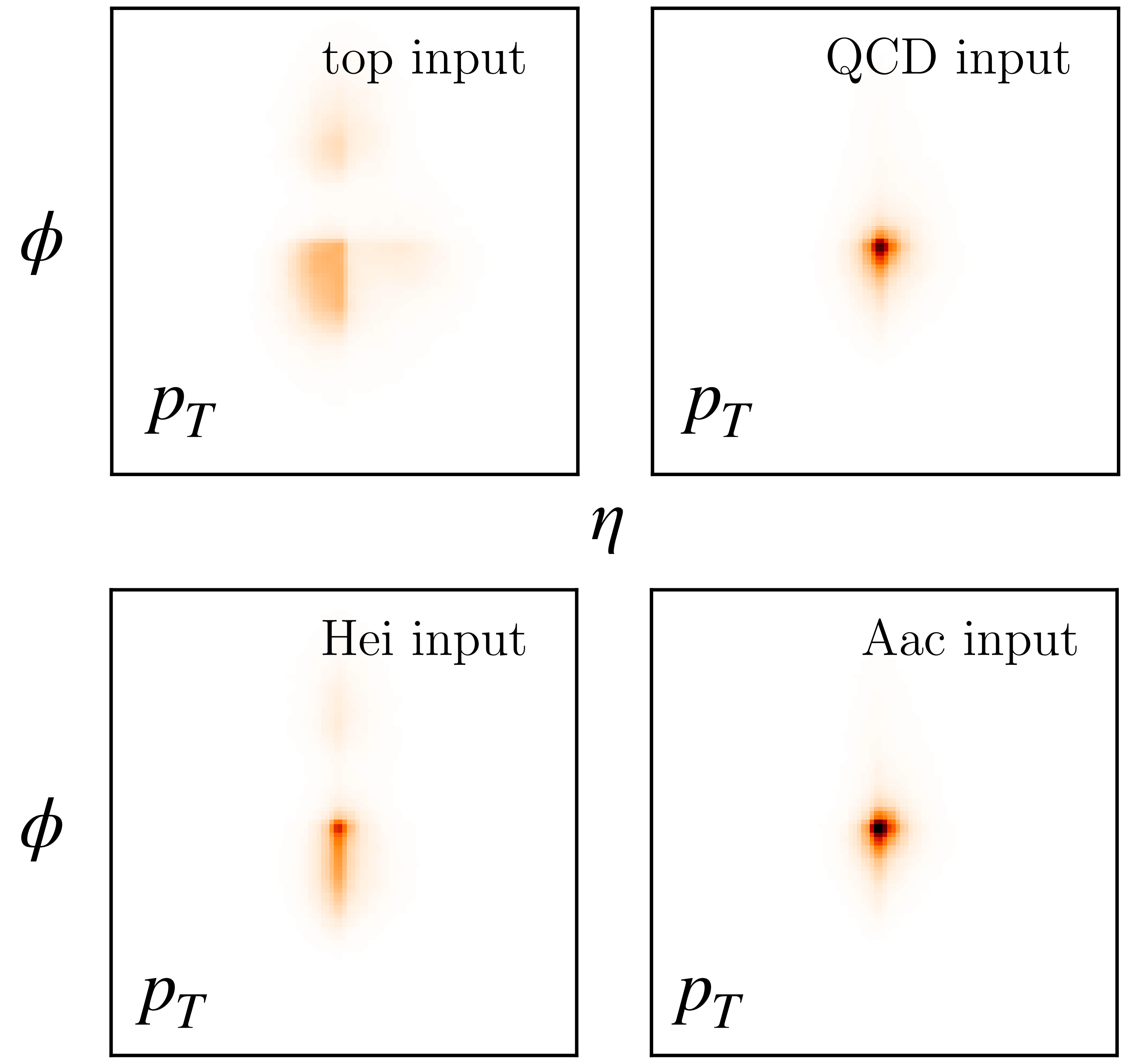
Anomaly detection $\longrightarrow$ as few as possible assumptions.

Preprocessing used to include known symmetries:

- center in (η, ϕ)
- rotate the principle axis
- normalize pixels



Average image:



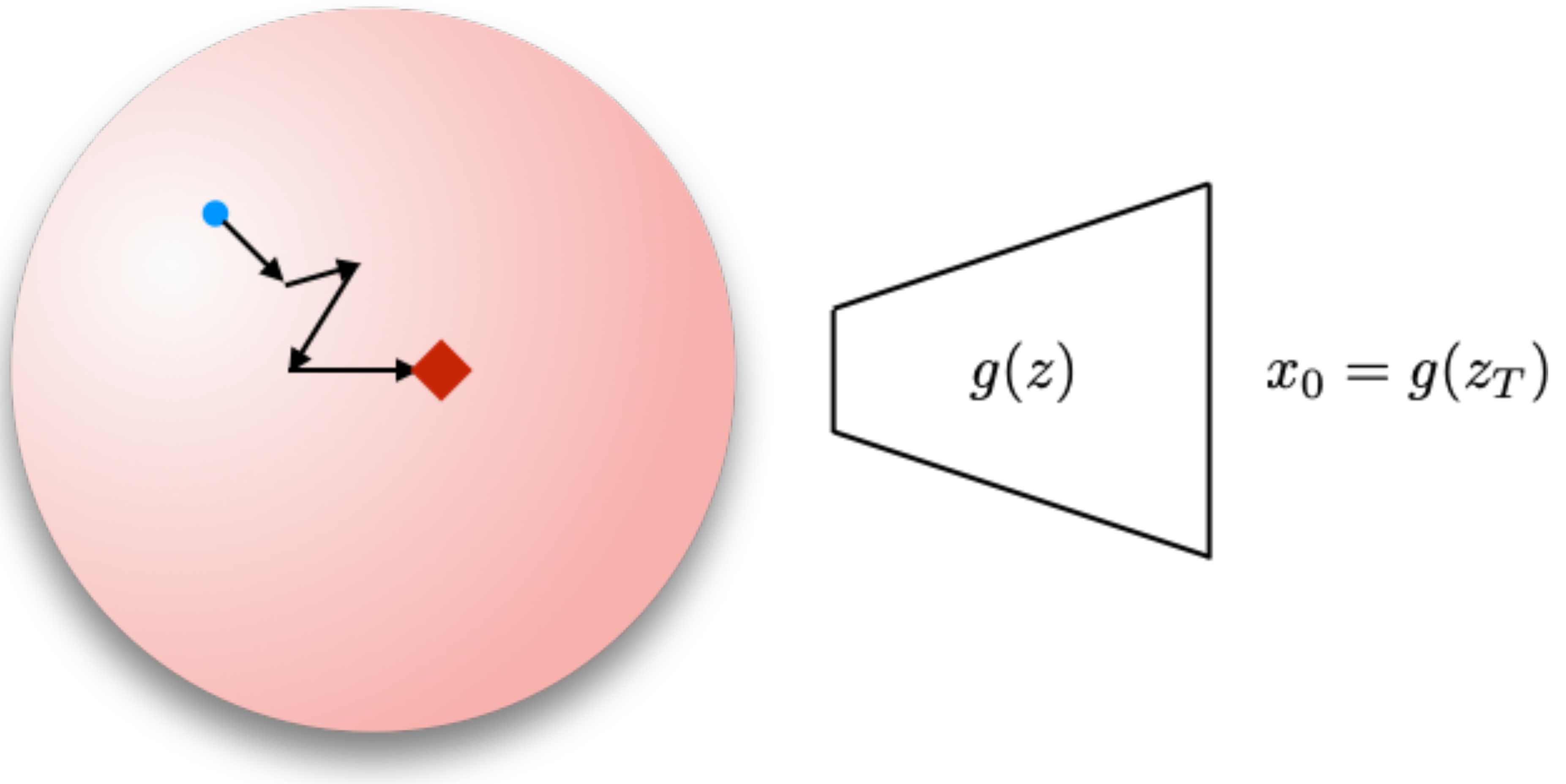
Sampling from the model

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

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

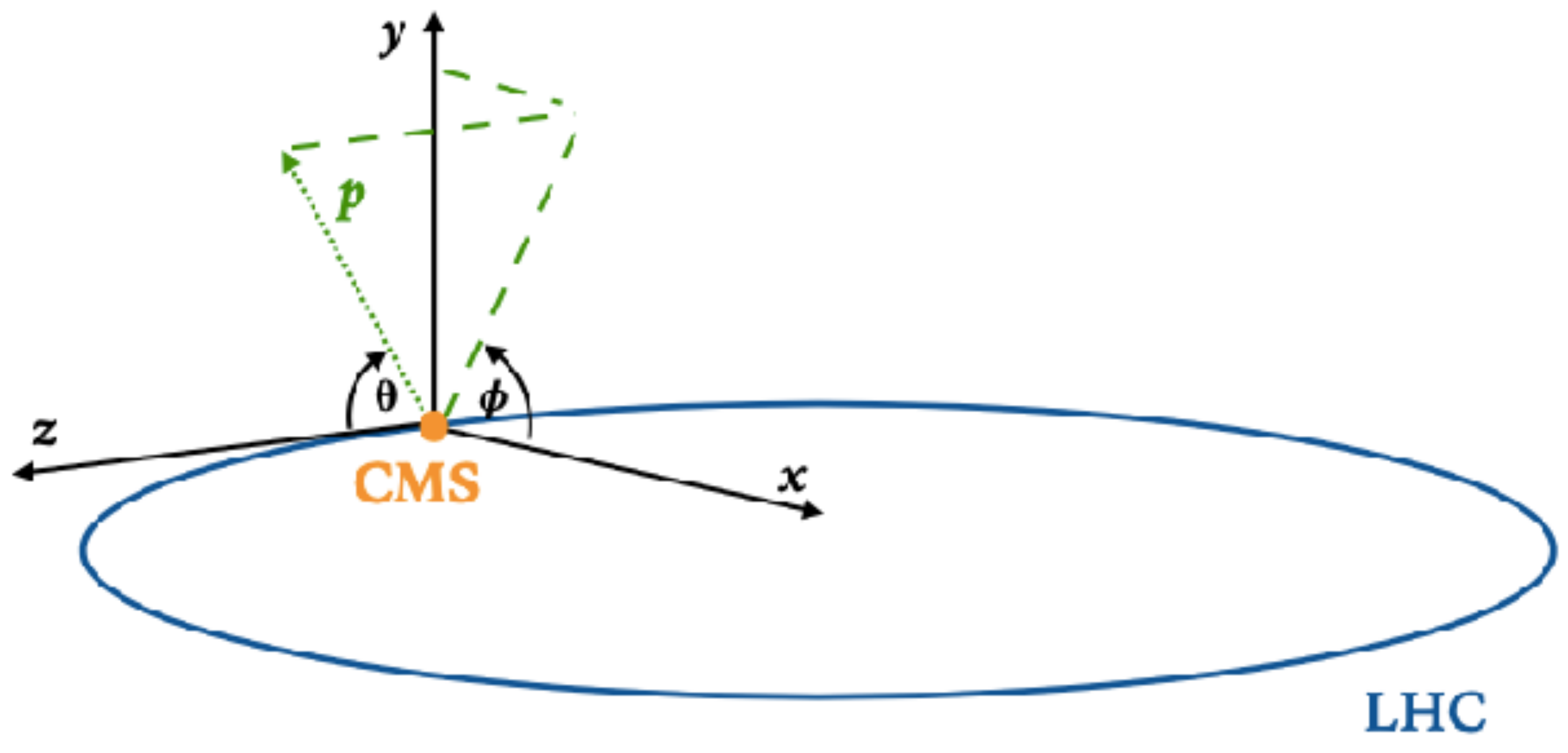
AnomalyCLR on events

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

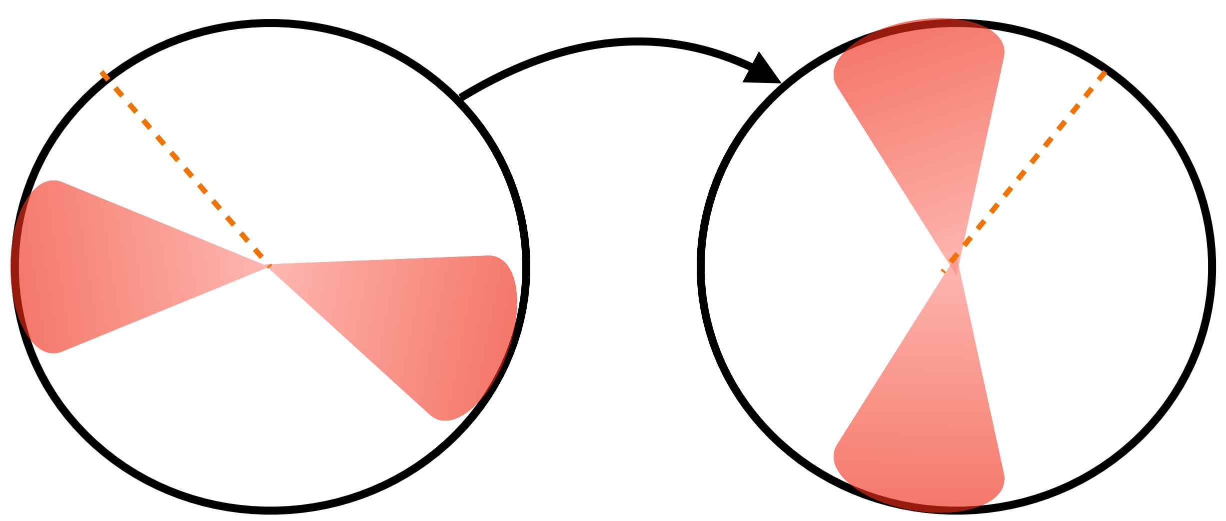
unsupervised training $\longrightarrow$ no signals available during training

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



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

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

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

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

Anomalous augmentations

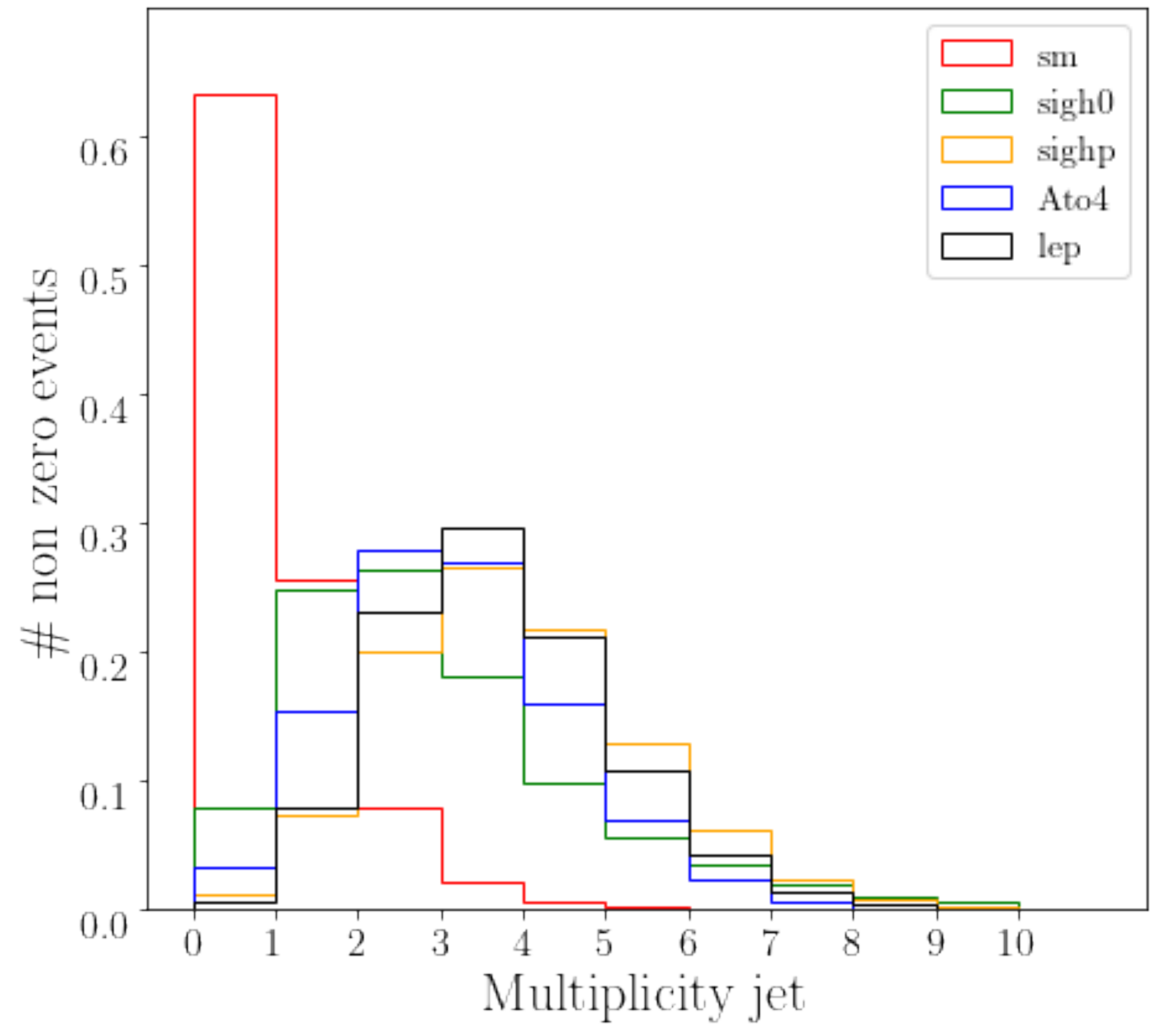
Loss function:

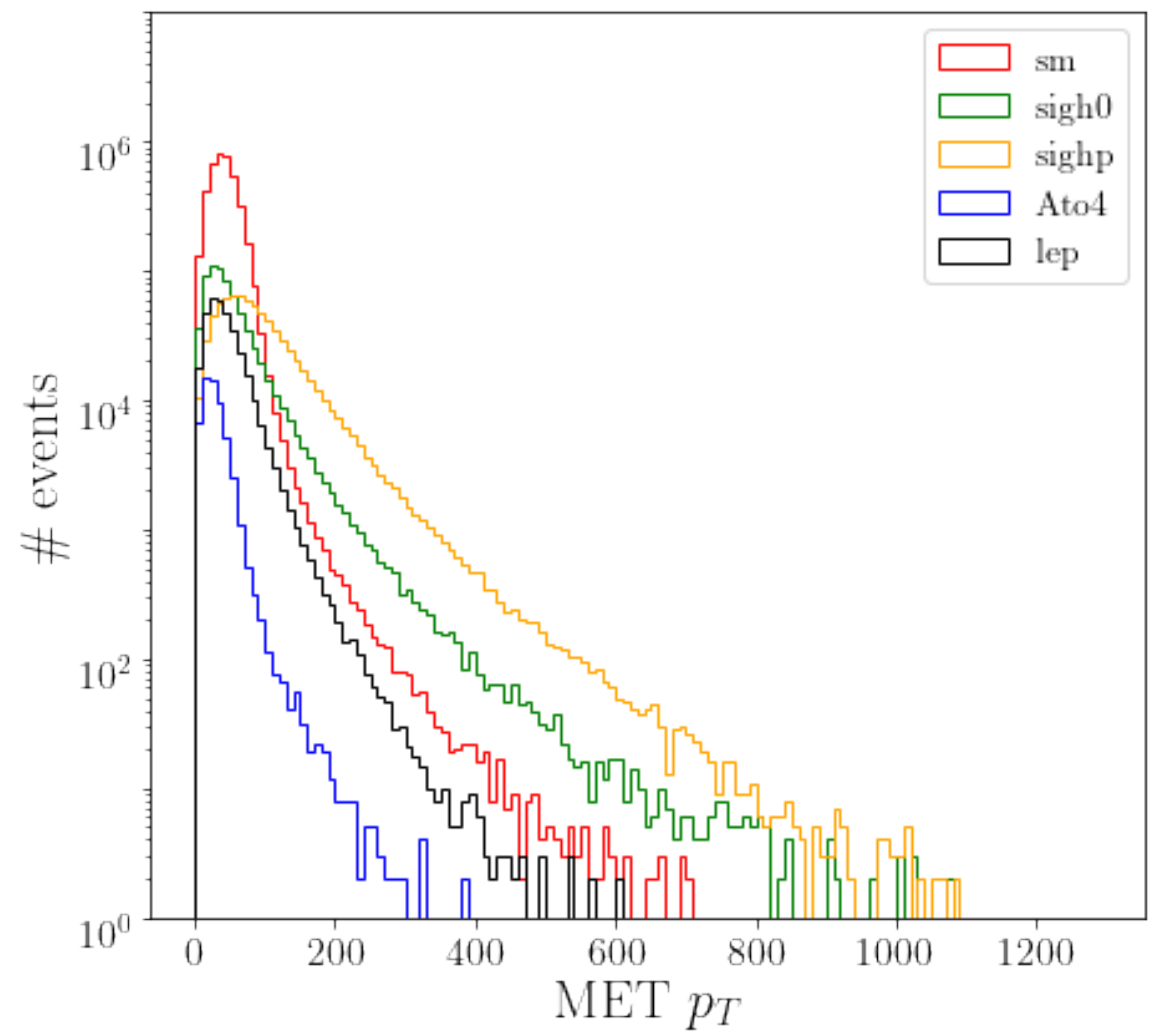
$$\mathcal{L}_{AnomCLR+} = -\log e^{[s(z_i, z_i') - s(z_i, z_i^*)]/\tau} = \frac{s(z_i, z_i^*) - s(z_i, z_i)}{\tau}$$

Anomalous augmentations:

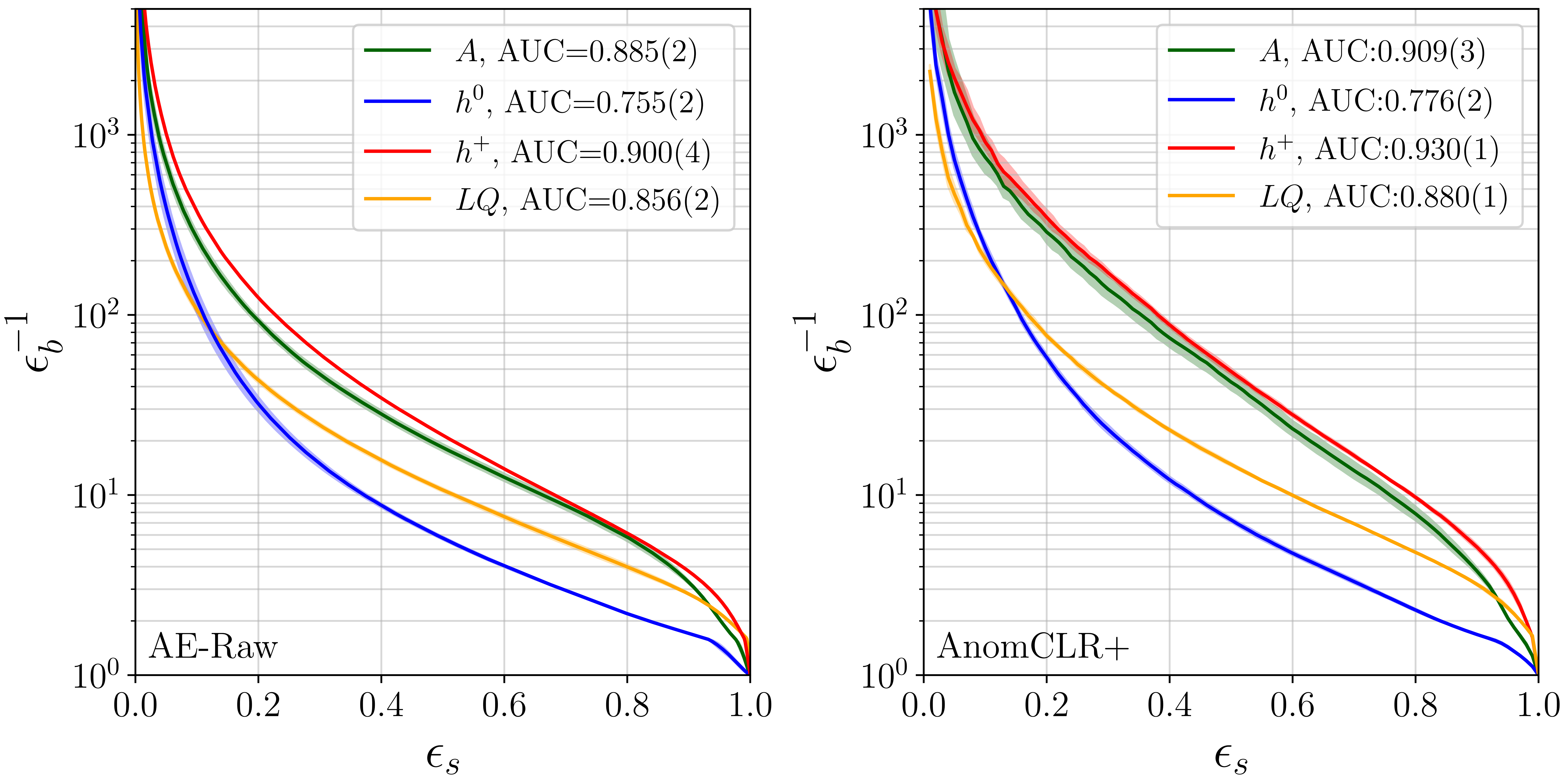
- multiplicity shifts:
  - add a random number of particles, update MET
  - split existing particles, keeping total p_T and MET fixed
- p_T and MET shifts

Each augmentation increase sensitivity to BSM-like features

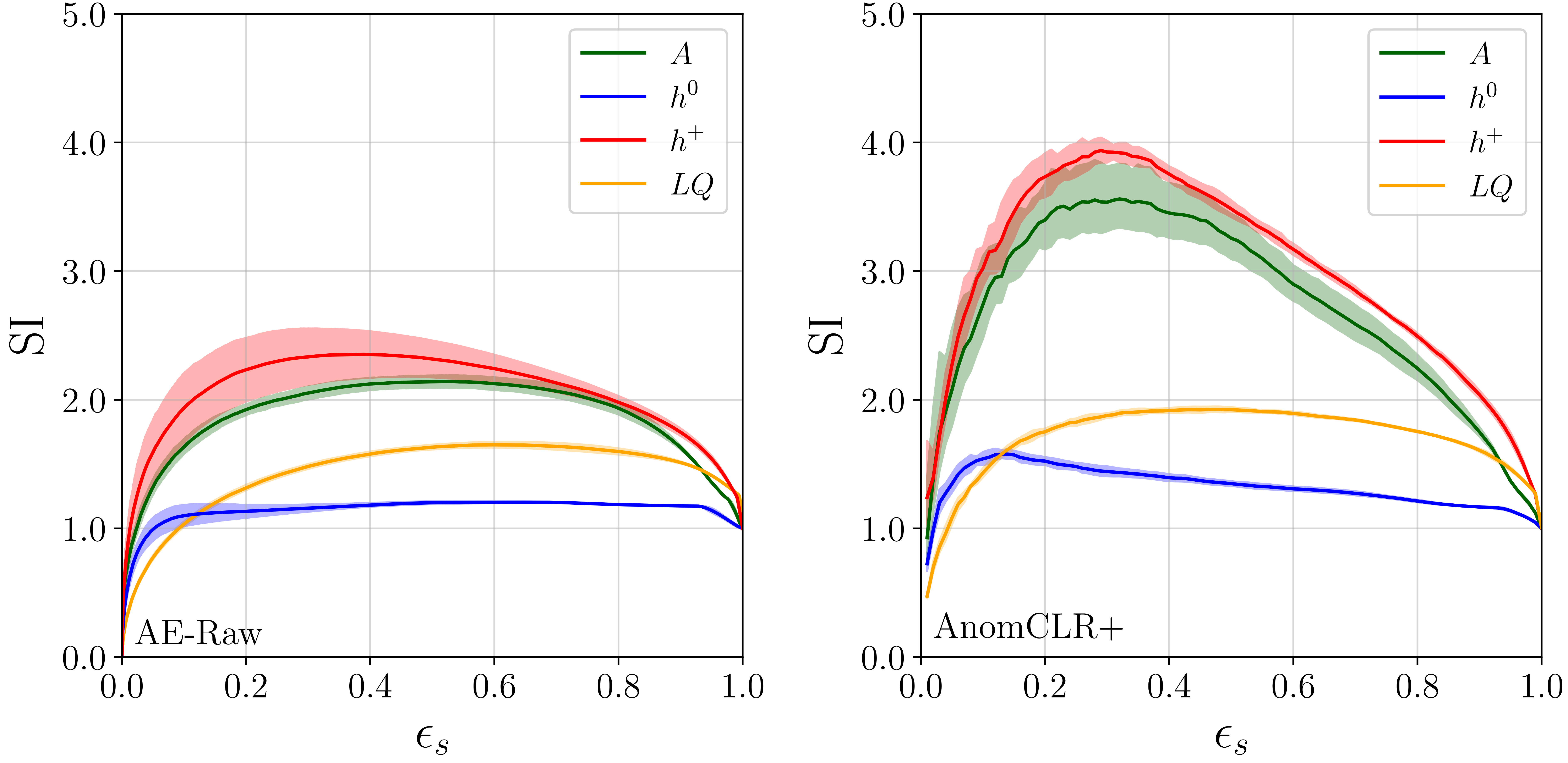




Results: improved sensitivity



Results: SIC CURVES



Effect of anomalous augmentations