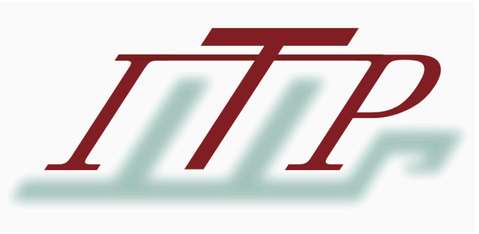
A Normalized Autoencoder for LHC Triggers

Favaro Luigi

UNIVERSITÄT
HEIDELBERG
ZUKUNFT
SEIT 1386



ML4Jets 2022 - Rutgers University

in collaboration with:
Barry Dillon, Michael Krämer, Tilman Plehn, Peter Sorrenson

arXiv:2206.14225

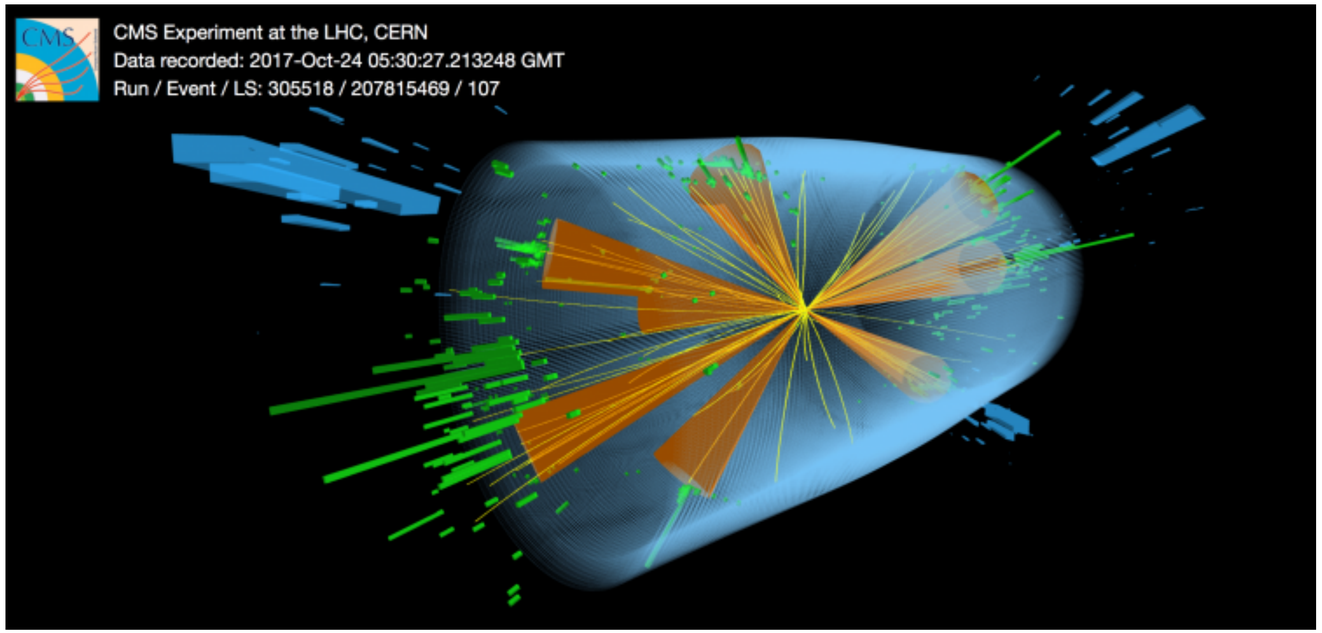
HEP challenges

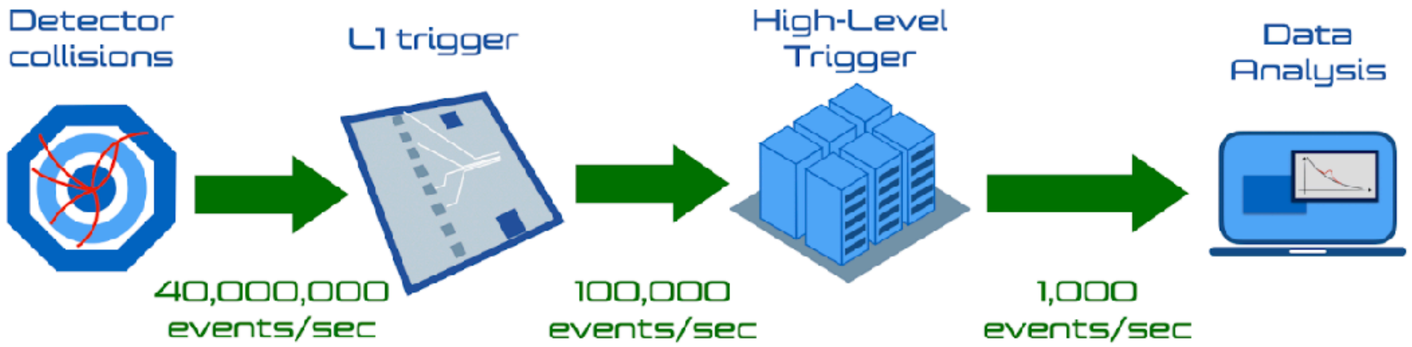
LHC is still looking for BSM physics:

New physics may appear
as unexpected jet events

↓

Jet tagging





Are we discarding interesting events?

↓

Improving triggers/analysis

A Machine Learning approach

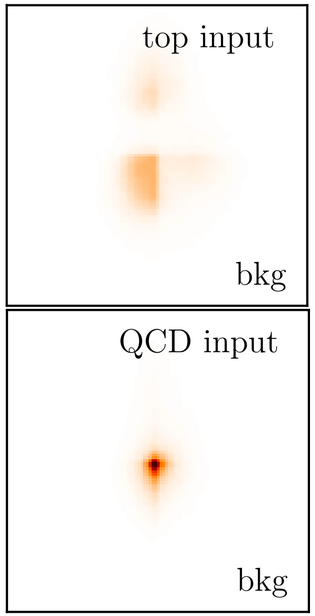
Both problems drew attention of ML methods:

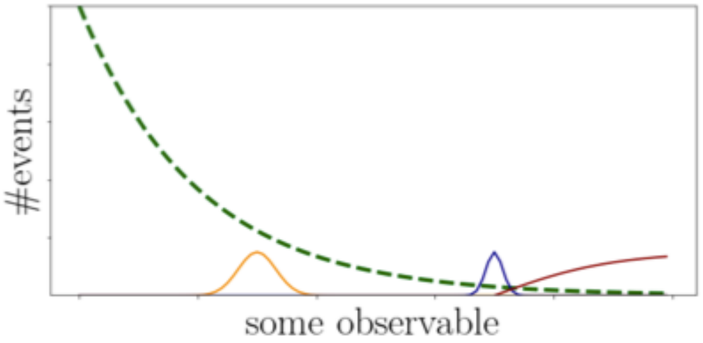
Jets classification:
Train the network on jets representation
(images, graphs, hlf)

Anomaly detection:
unsupervised and semi-supervised methods
(AE, VAE, DVAE, CWOLA, …)

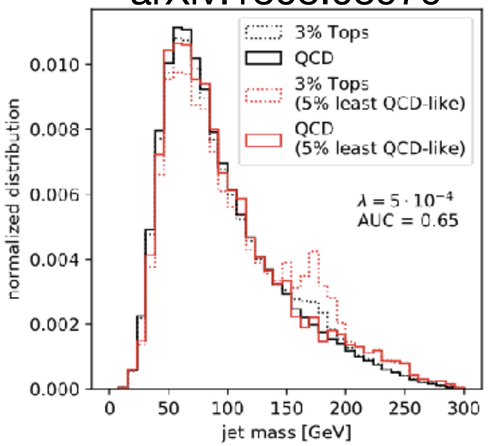
Triggering:
CMS@40MHz challenge:

arXiv:2107.02157

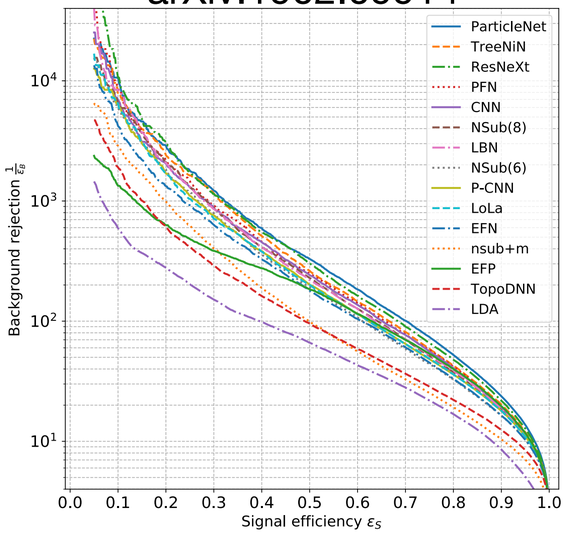




arXiv:1808.08979



arXiv:1902.09914

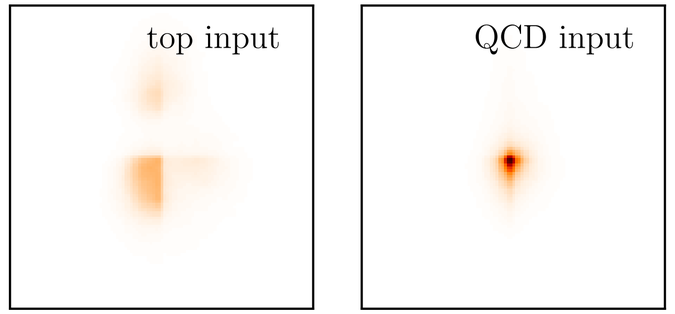


Building jet images

We build our representation starting from jet constituents:

- collect a major fraction of constituents;
- apply preprocessing $\longrightarrow$ introduce symmetries
- pixelize the data in the (η, ϕ) plane;

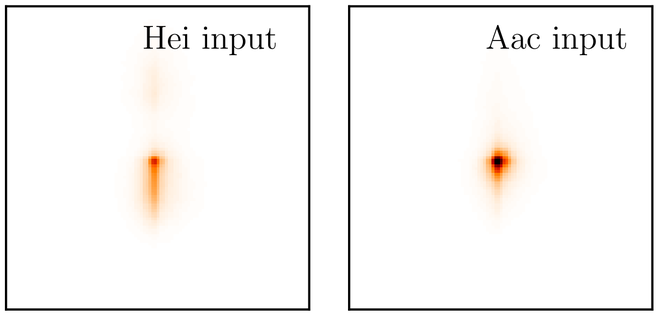
benchmark test



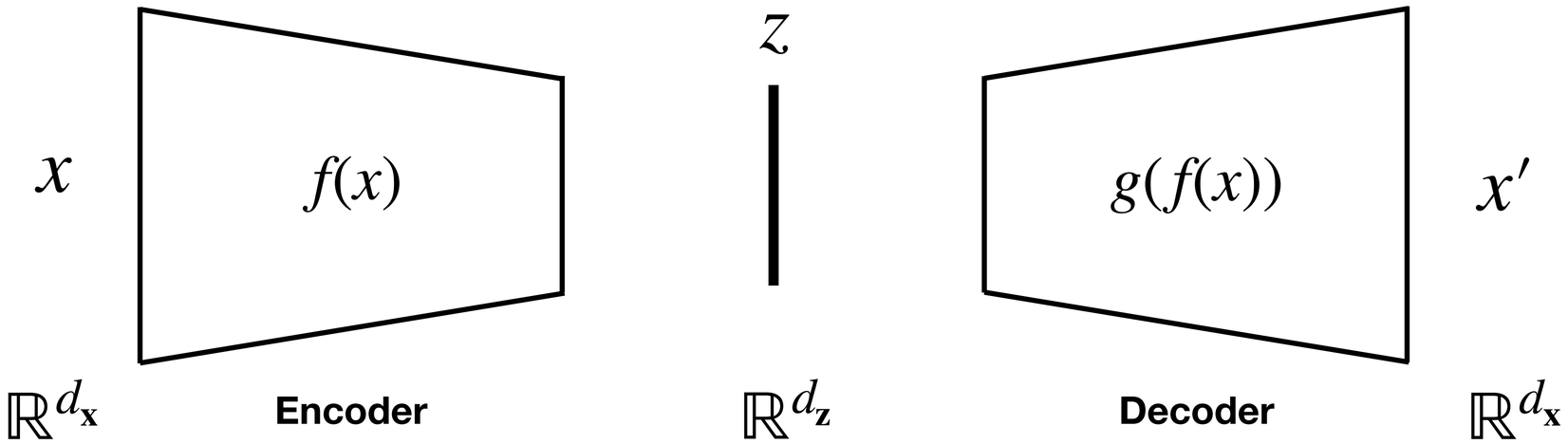
New BSM signals may have a QCD-like structure:

- DM hidden valley scenario:
  - invisible dark components (Aachen dataset)
  - modified QCD structure (Heidelberg dataset)
- tagging after introducing an implicit bias: $p_T \rightarrow p_T^n$

BSM signal



Autoencoders for AD



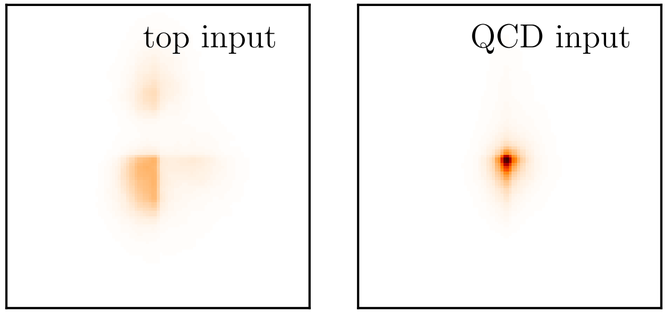
Building an AE:

- define an encoder - decoder network;
- encode features in a low-dimensional latent space;
- use the reconstruction error as anomaly score, MSE(x, x');

⚠ MSE is not a fail-proof anomaly score

Autoencoders for AD

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



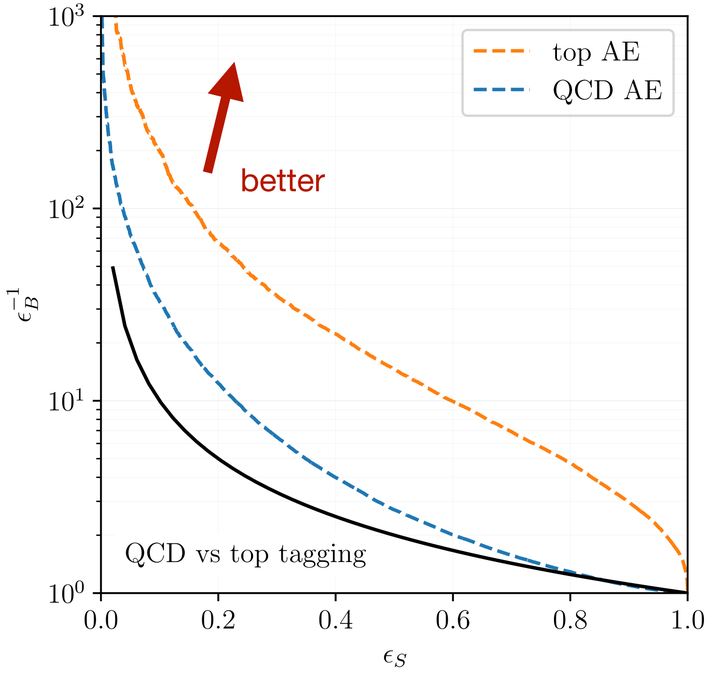
Robustness test: inverse training

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

Example: QCD tagging

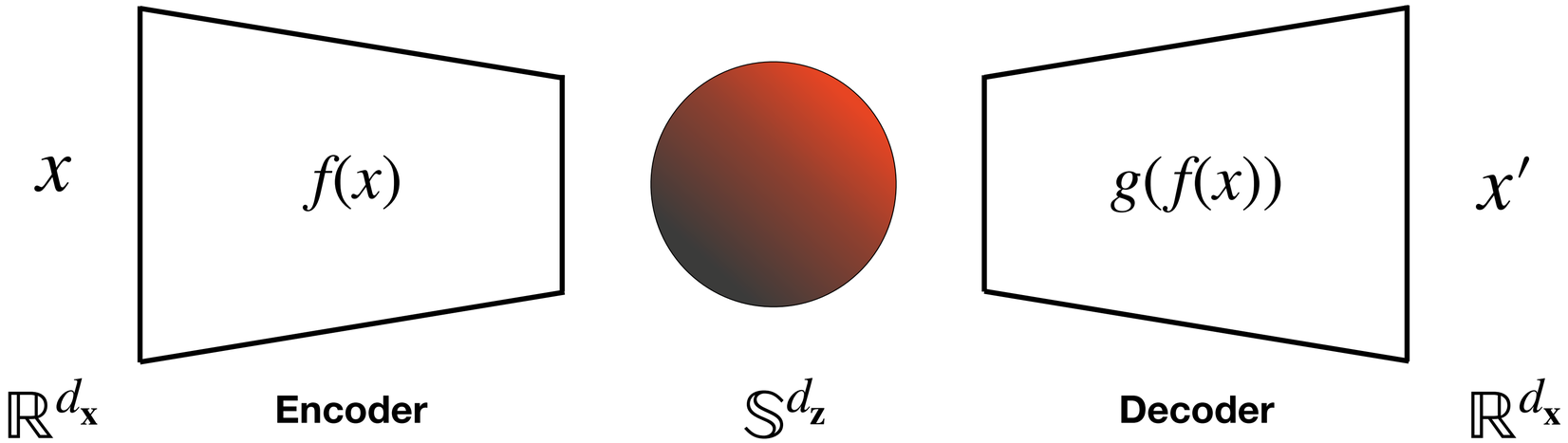
- use top jets as background…
- … and tag QCD jets;
- usual AE-like approach don’t solve the problem;

arXiv:2104.08291



Normalized Auto-Encoders

arXiv:2105.05735



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

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

If we consider the reconstruction error as energy function, we can train by minimizing the negative log-likelihood of the probability distribution:

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

Ω high-dimensional space → intractable integral

Training a NAE

Consider the gradients of the loss function:

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

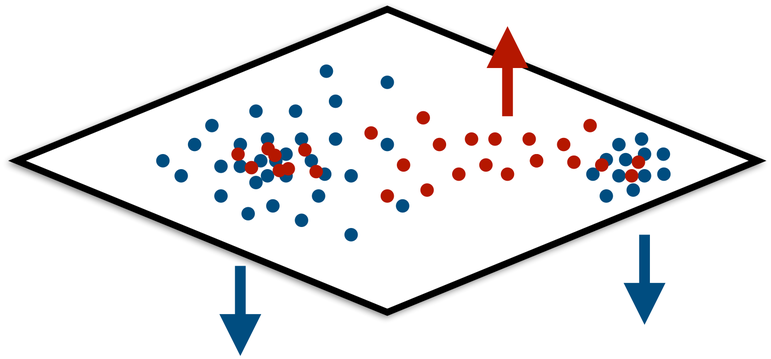
Minimizes the usual AE reconstruction error;

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

Sampling from the model*

arXiv:2105.05735

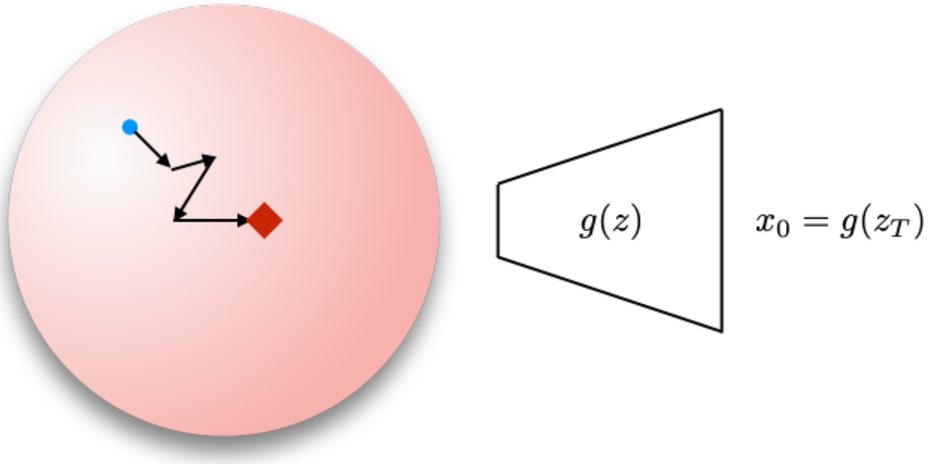
*choices made to reduce the training time

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

Sampling from the model*

arXiv:2105.05735

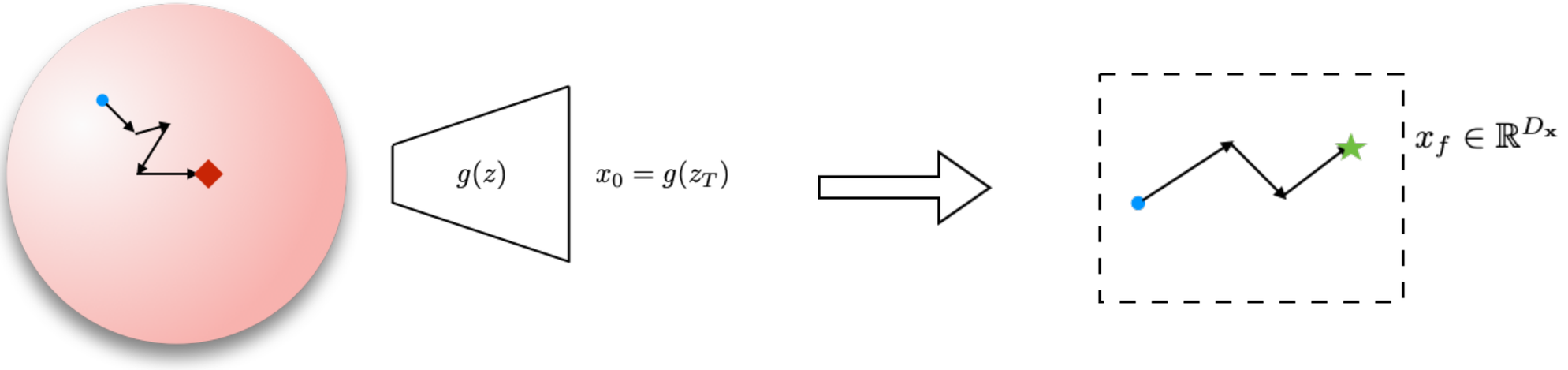
*choices made to reduce the training time

- 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:

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

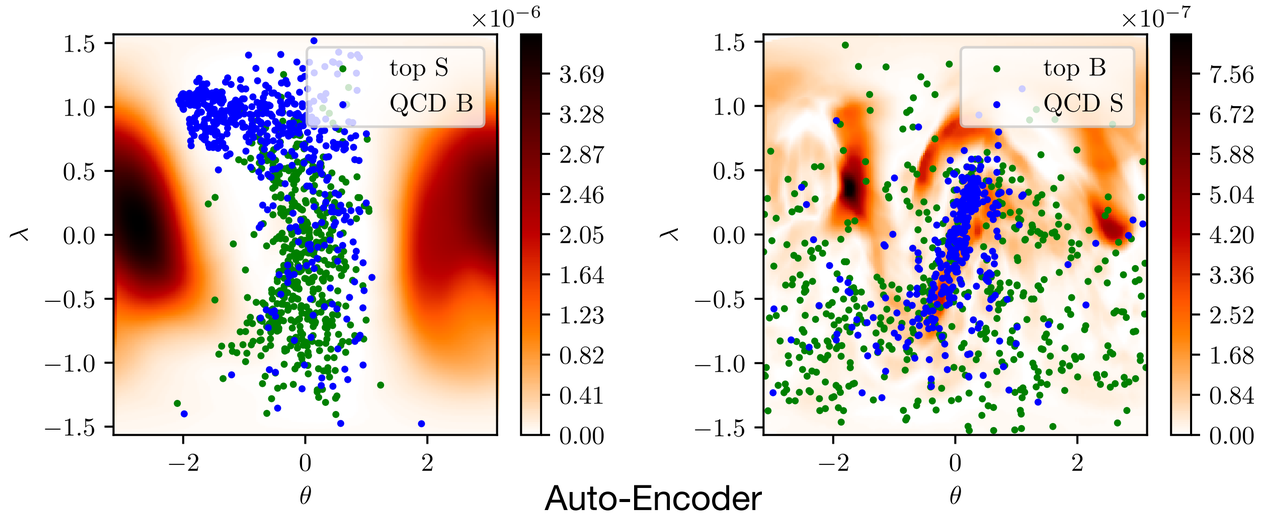


Let’s have a look at some results

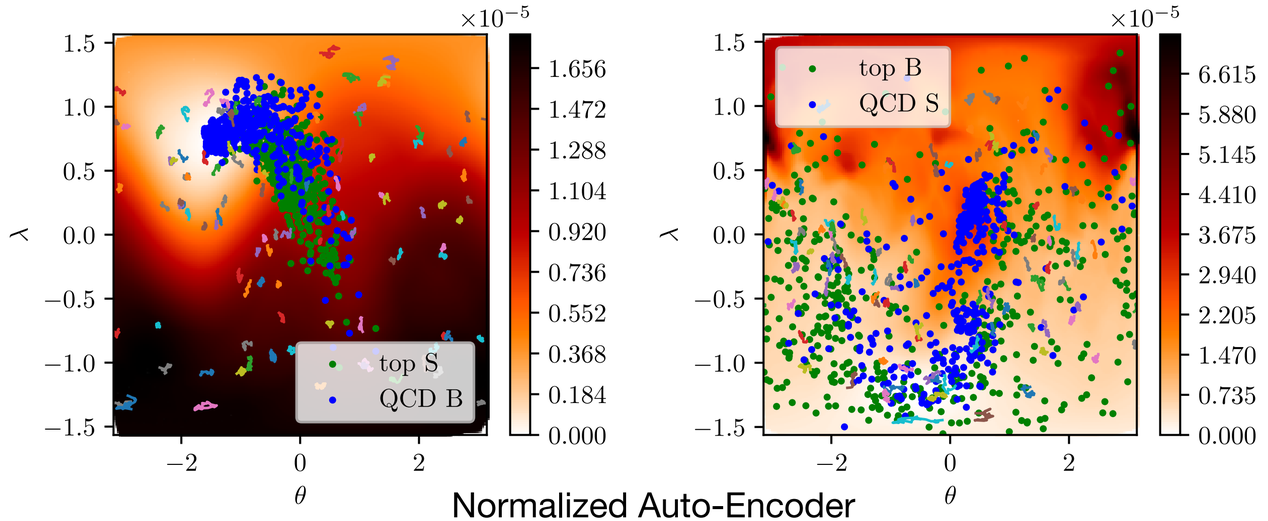
Results: decoder manifold

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;



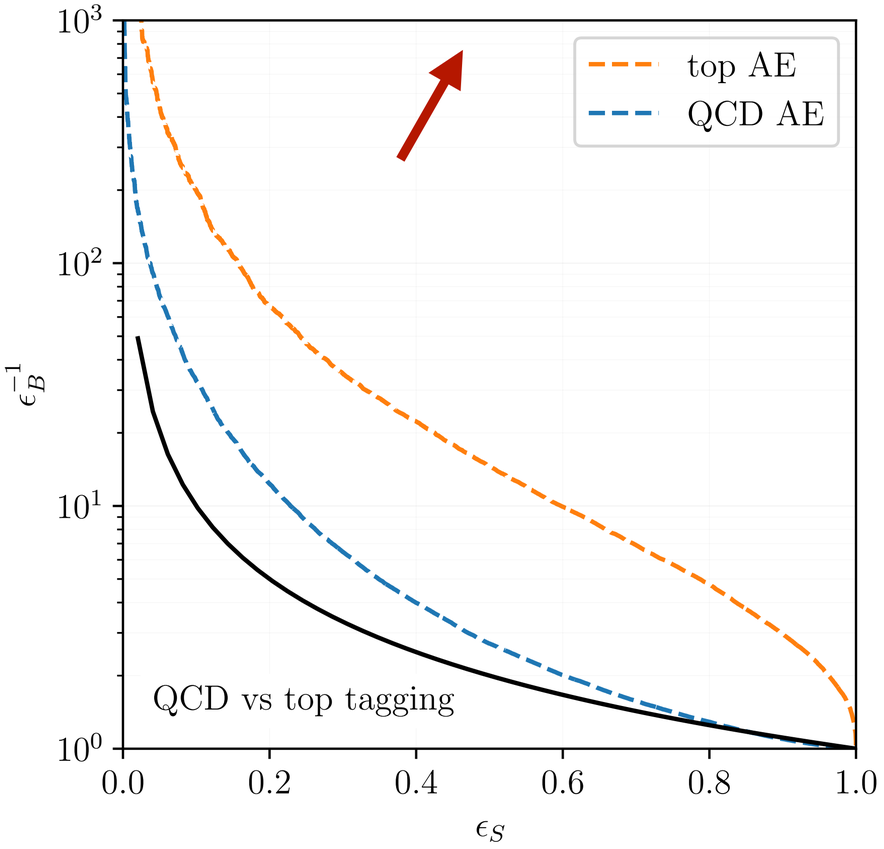
Auto-Encoder



Normalized Auto-Encoder

Results: QCD vs top tagging

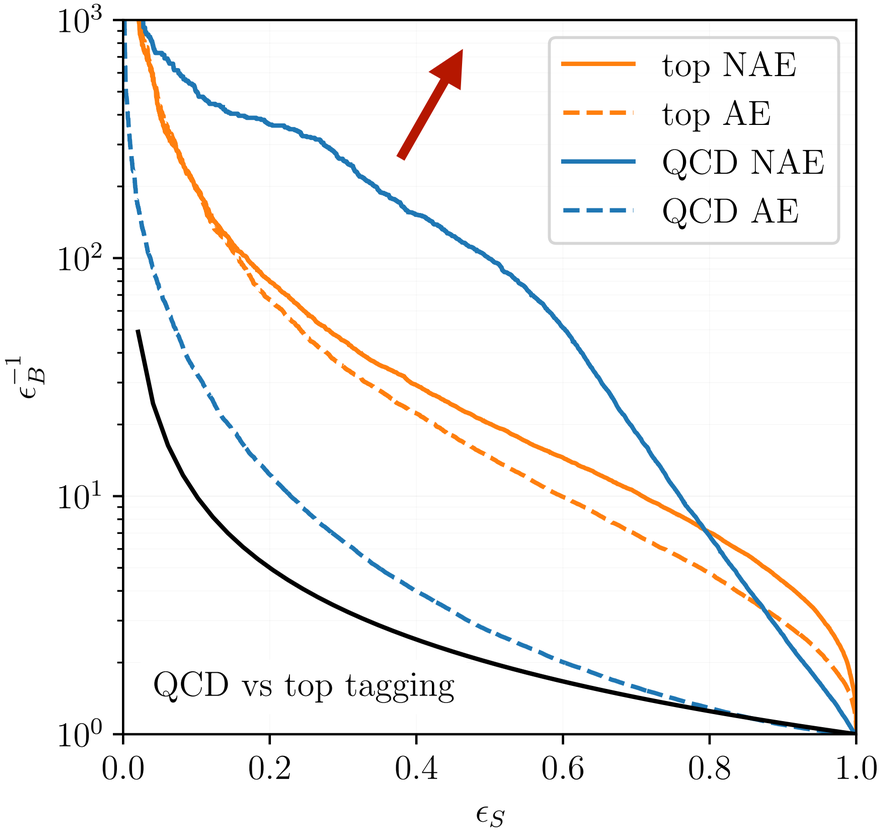
- AE trained on jet images fails at tagging QCD jets;
- an AE is able to interpolate the simpler QCD features;



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		



NAE on events (preliminary)

training on SM cocktail events:

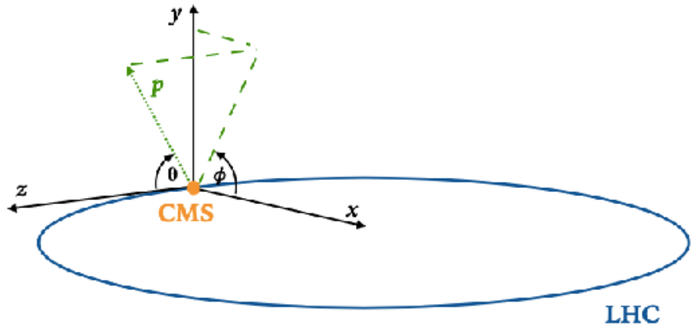
$W \to l\nu$ (59.2%)
$Z \to ll$ (6.7%)
$t\bar{t}$ production (0.3%)
QCD multijet (33.8 %)

look for various BSM signals:

$A \to 4l$
$LQ \to b\nu$
$h_0 \to \tau\tau$
$h_+ \to \tau\nu$

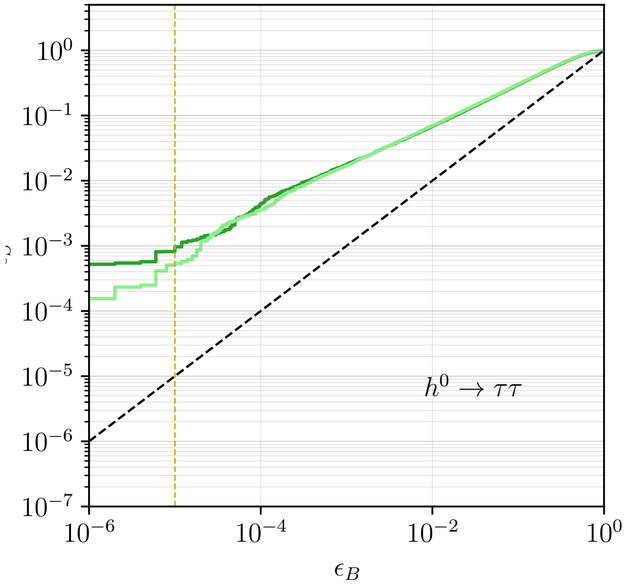
The events are represented in the typical L1 format: (19, 3) entries

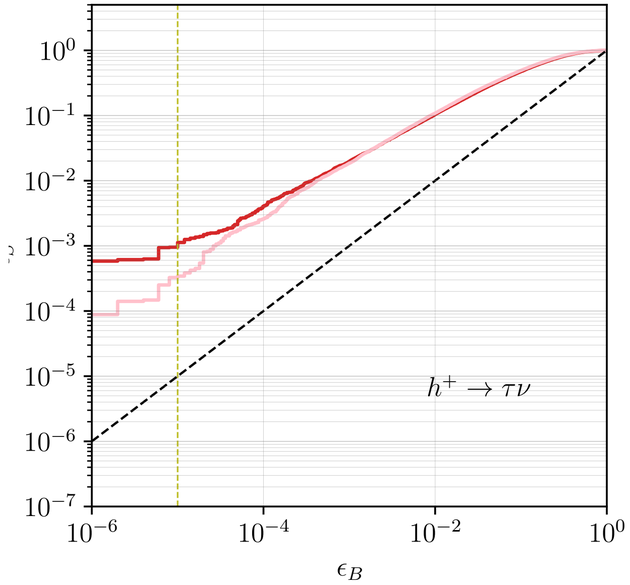
- 19 particles: MET, 4 electrons, 4 muons, and 10 jets
- 3 observables: p_T, η, ϕ
- lepton cut $p_T > 23$ GeV

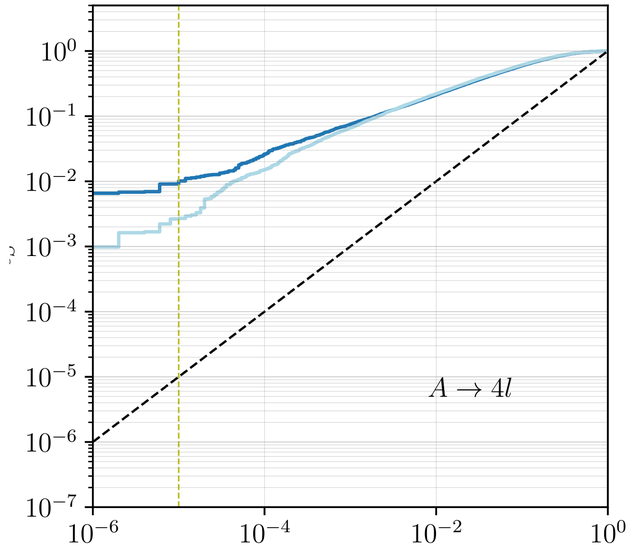


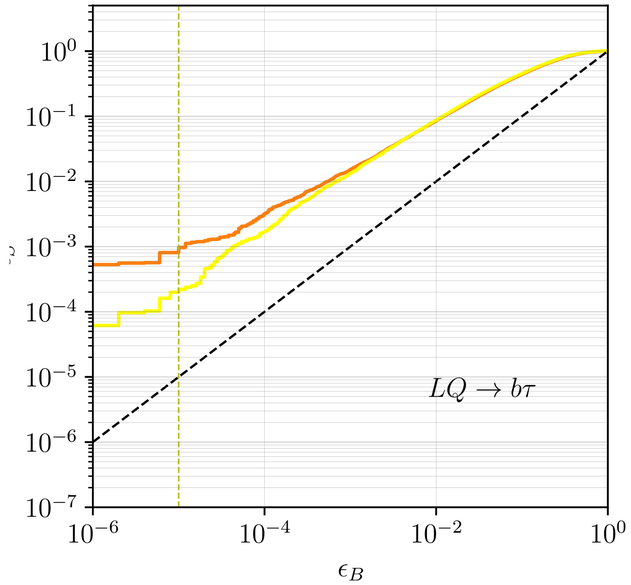
Tagging performances

- focusing on $\epsilon_B \sim 10^{-5}$ (output data rate available);



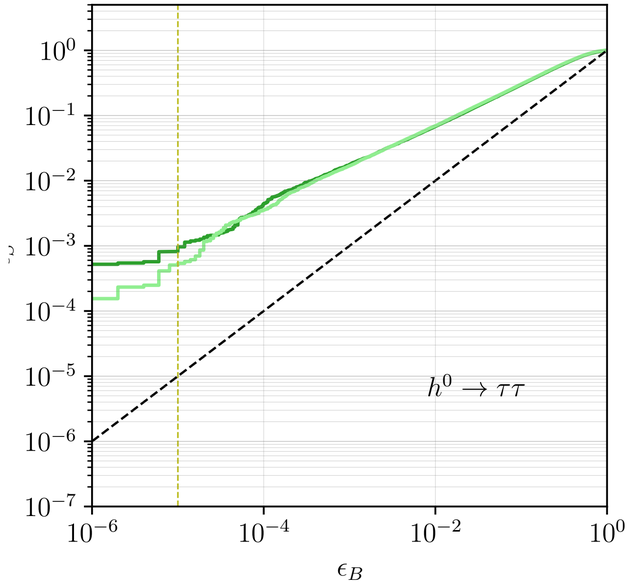


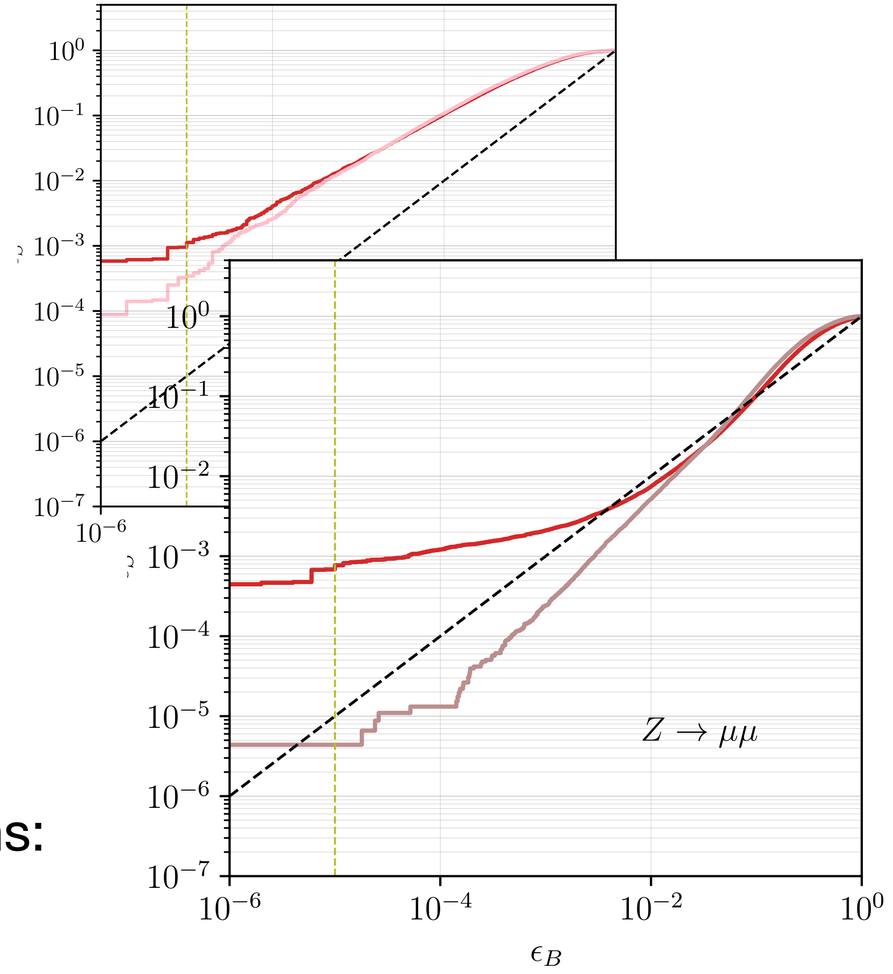


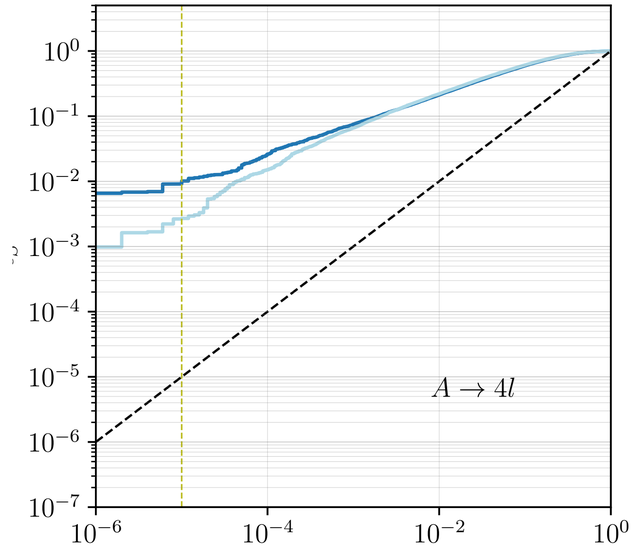


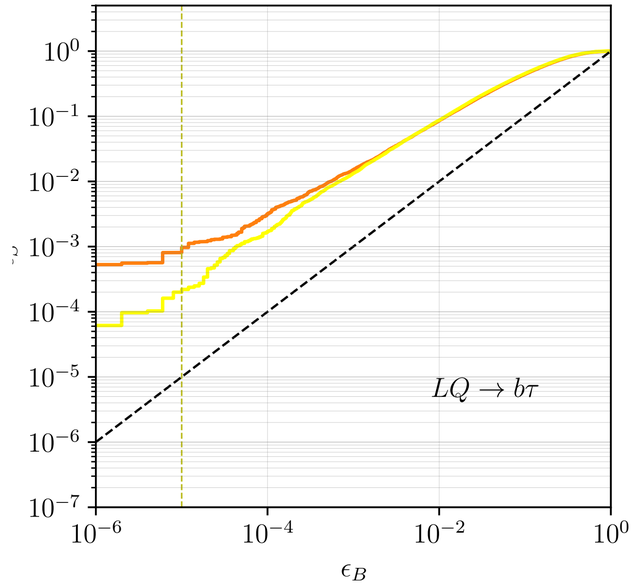
Tagging performances

- focusing on $\epsilon_B \sim 10^{-5}$ (output data rate available);









light Z decaying into two muons:

$$Z' \to \mu\mu$$

Conclusions

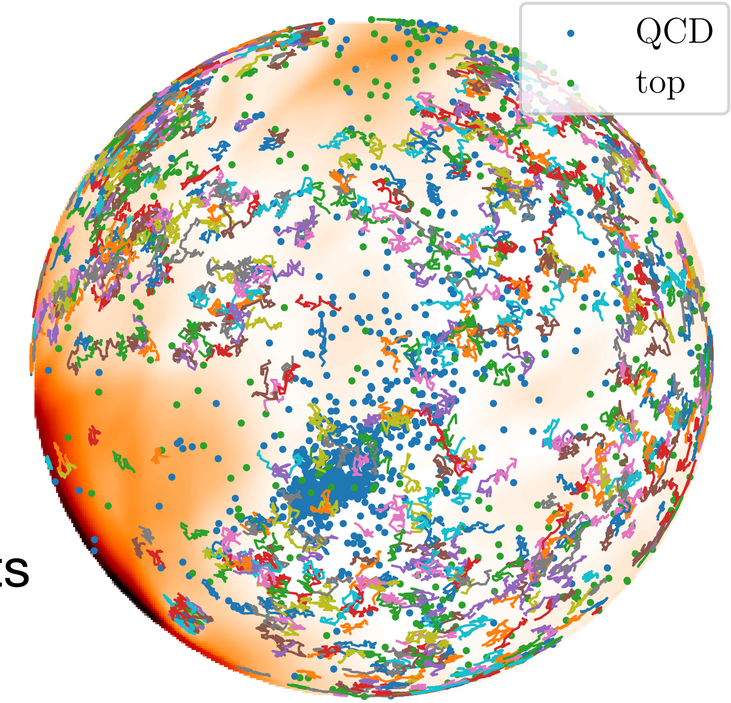
Normalized Auto-Encoders allow for:

- an energy-based description of an Auto-Encoder
- penalization of regions not covered by the training distribution:

  → no complexity bias
- the code will be available on GitHub:

  luigifvr/normalized-autoencoders

Example results: tagging QCD vs top and NAE on events



Next steps:

- Detailed study and comparison between AE and NAE on events
- study implementation of NAE on FPGA
- paper on arXiv soon…

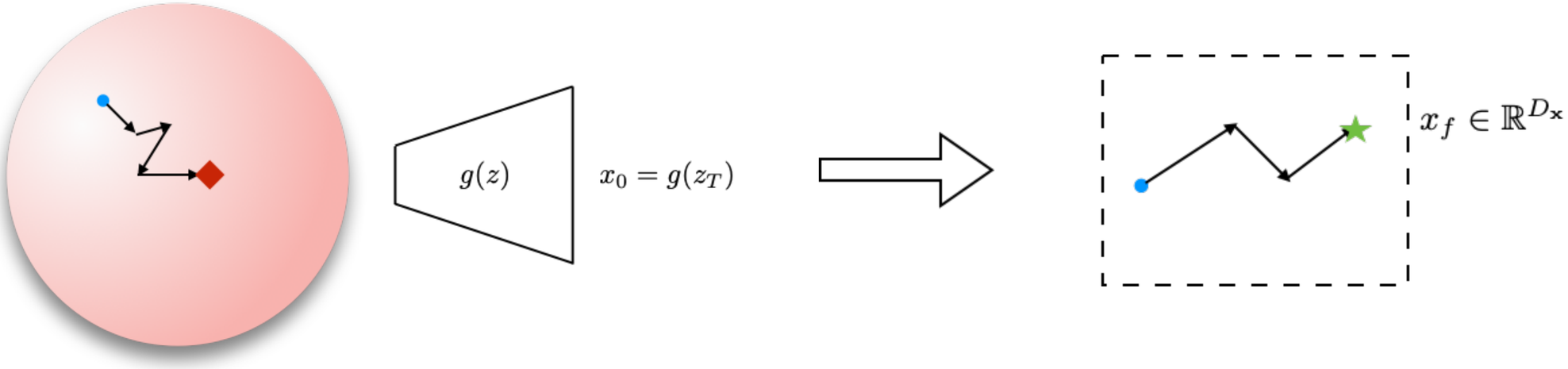
Thanks for you attention!

Sampling from the model*

The second chain is performed in the input space using the distribution $p_\theta(x)$:

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

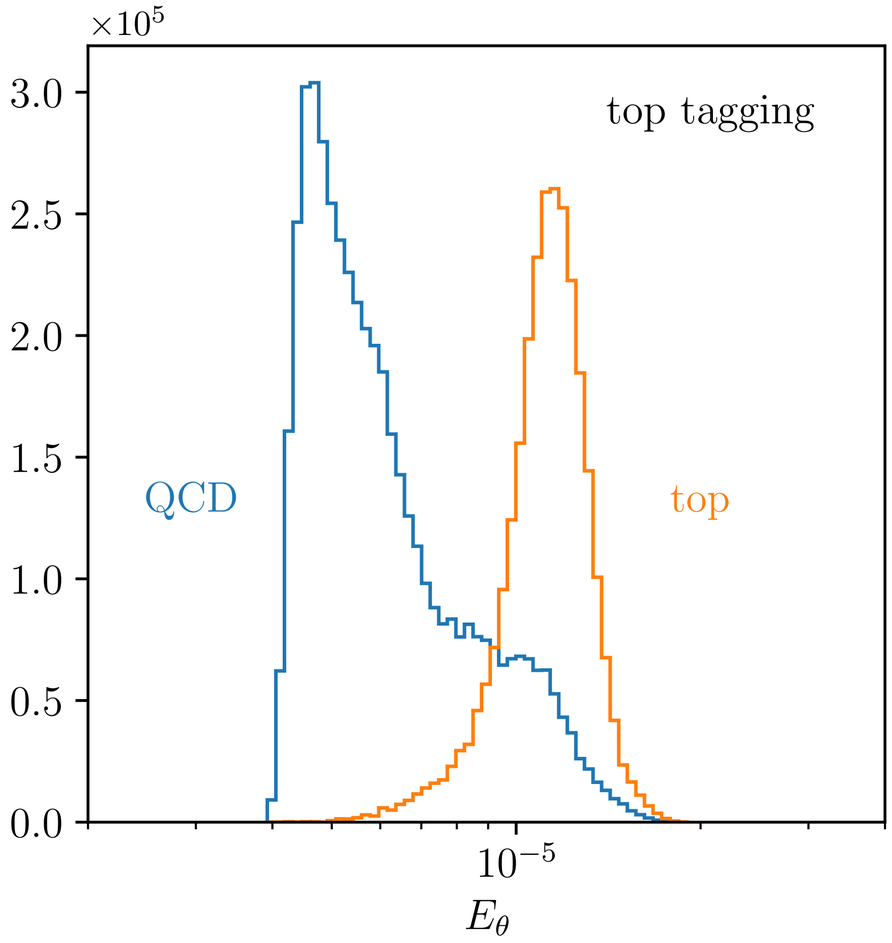
* both chains run for a small number of steps $\mathcal{O}(100)$, and they are constrained into low energy regions by taking $\lambda > \sigma$

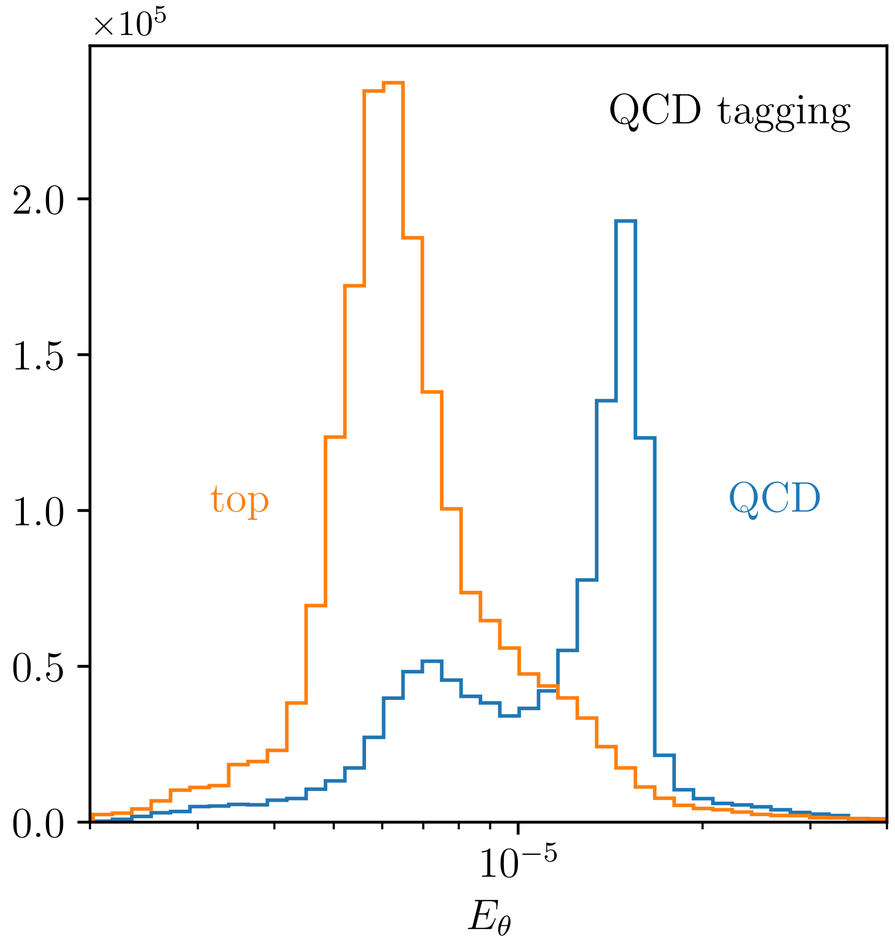


Let's have a look at some results!

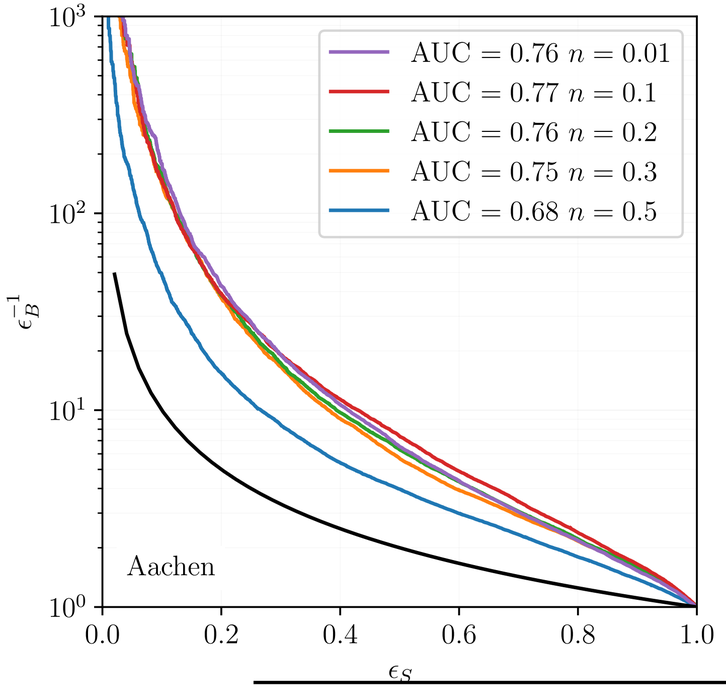
Backup

MSE histograms for QCD and top tagging

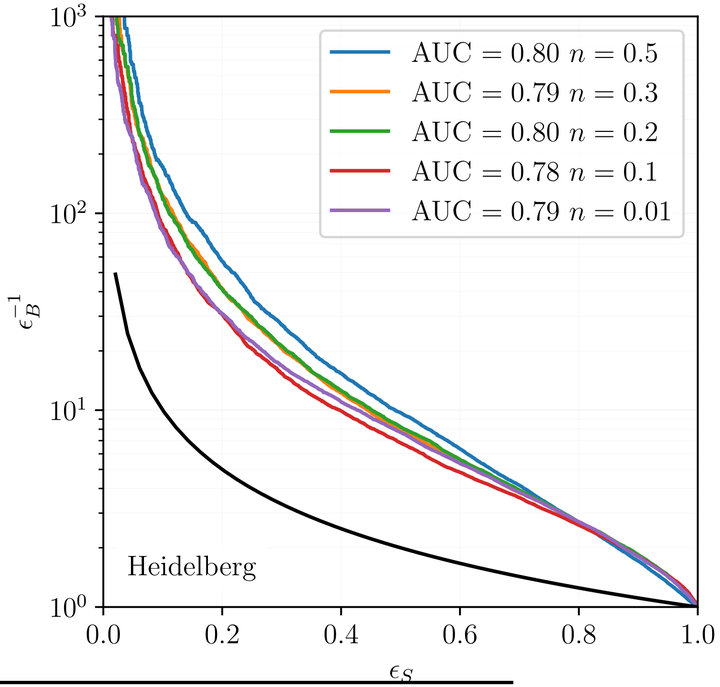




Backup

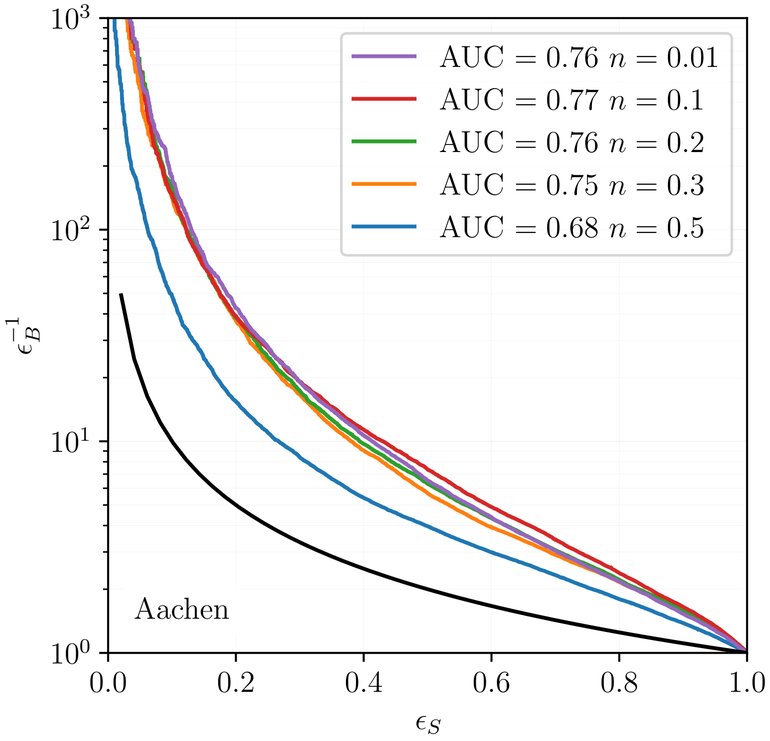


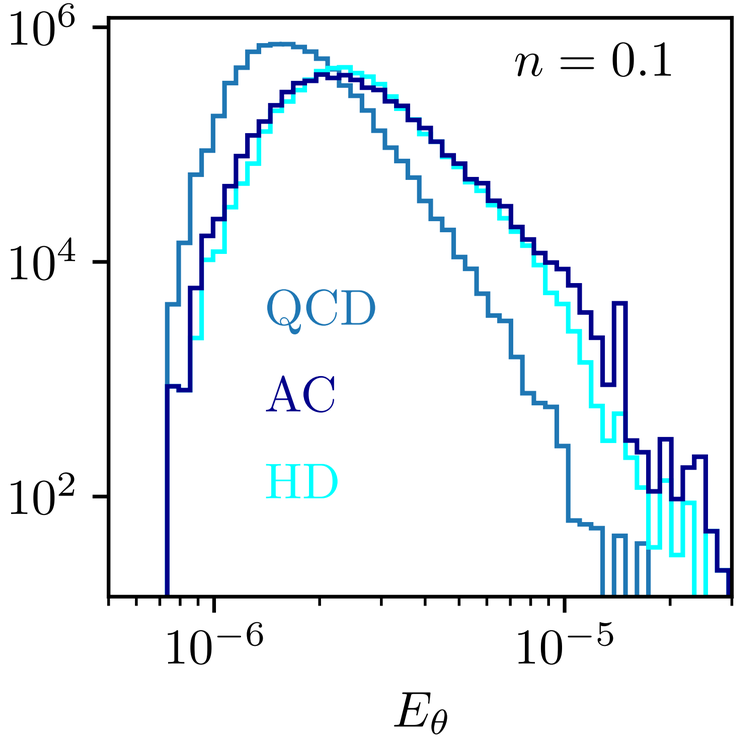
small dependence
on the implicit bias!



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)

Backup


AUC = 0.76 $n = 0.01$
AUC = 0.77 $n = 0.1$
AUC = 0.76 $n = 0.2$
AUC = 0.75 $n = 0.3$
AUC = 0.68 $n = 0.5$
Aachen
ϵ_B^{-1}
ϵ_S
10^0
10^1
10^2
10^3
0.0
0.2
0.4
0.6
0.8
1.0


$n = 0.1$
QCD
AC
HD
10^2
10^4
10^6
10^{-6}
10^{-5}
E_θ