Anomalies in Hadronic Resonances using Autoencoders

Irina Ene

PH290E Seminar — May 4, 2022

Introduction

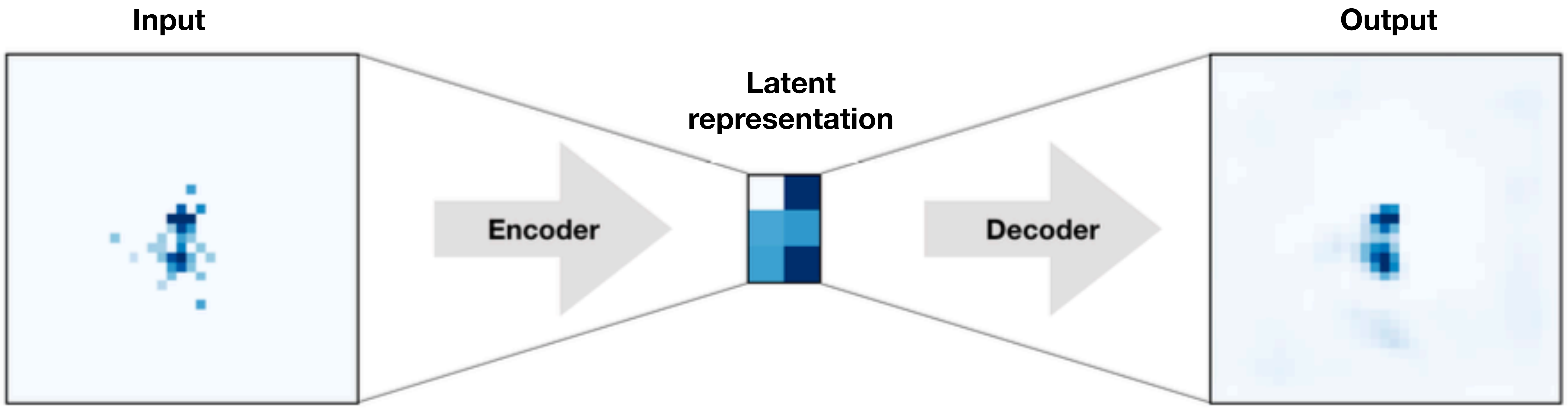
Motivation

- Searches for BSM physics at LHC are theory motivated (e.g. supersymmetry, Z’ etc.) and have so far not found definitive evidence
    - Need more **open-ended ways** to search for new physics at LHC
- One possible way to achieve this is through **anomaly detection using autoencoders**
    - Autoencoder learns what “normal” events looks like and how to reconstruct them
    - Autoencoder will fail to reconstruct “anomalous” events (because it hasn’t seen them before / they look different from the “normal” events)

Introduction

Autoencoders

arXiv 1808.08992

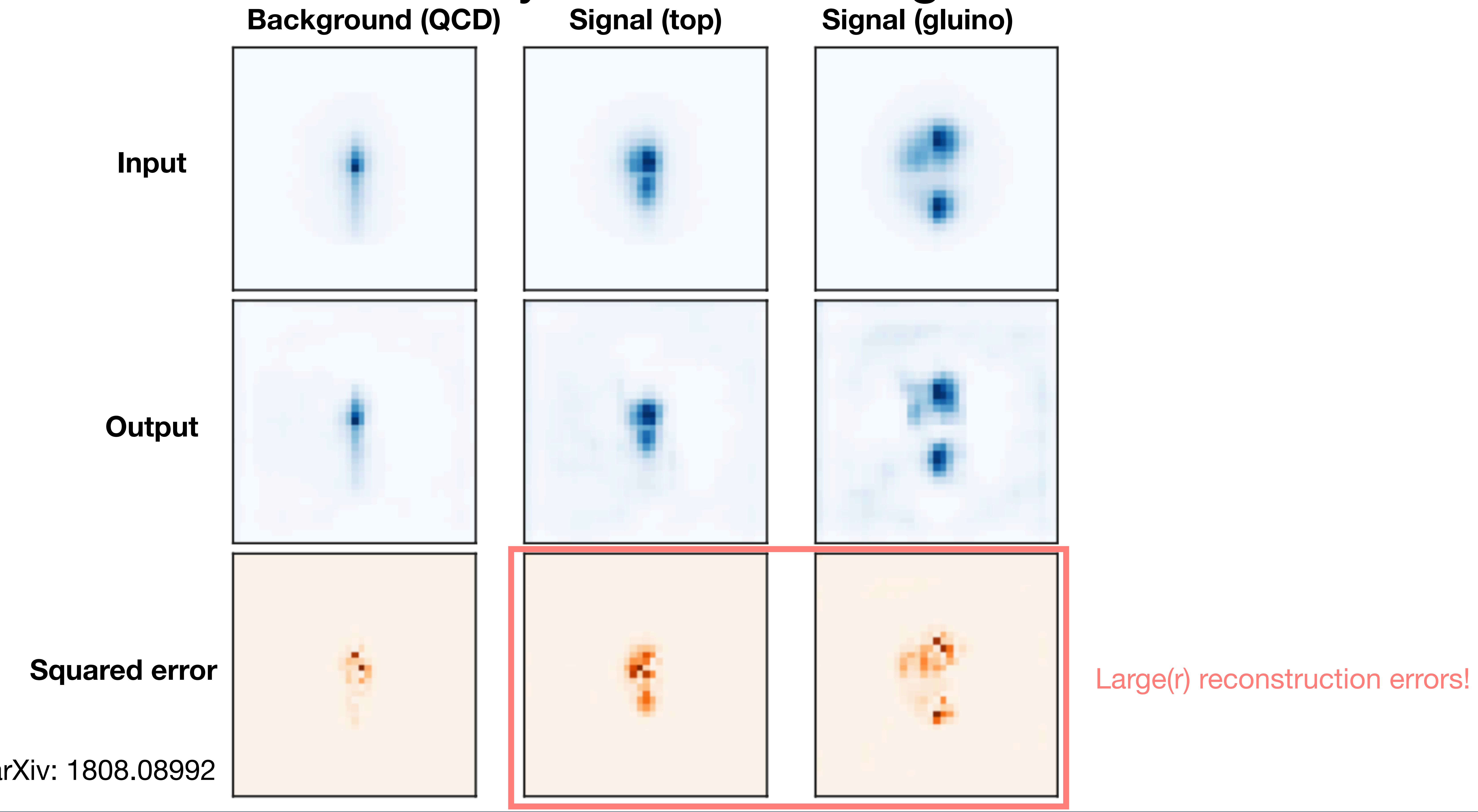


- Algorithm that maps **input** to a **latent representation** and then **back to itself**
- Relevant metric is "**reconstruction error**" — difference between input and output
  - e.g. sum/average of pixel-wise squared difference for images
- Train autoencoder to minimize the reconstruction error for the inputs it was trained on

Introduction

Autoencoders

- **Anomalies** should be **poorly reconstructed** by an autoencoder that is (optimally) trained on a sufficiently different background



arXiv: 1808.08992

Introduction

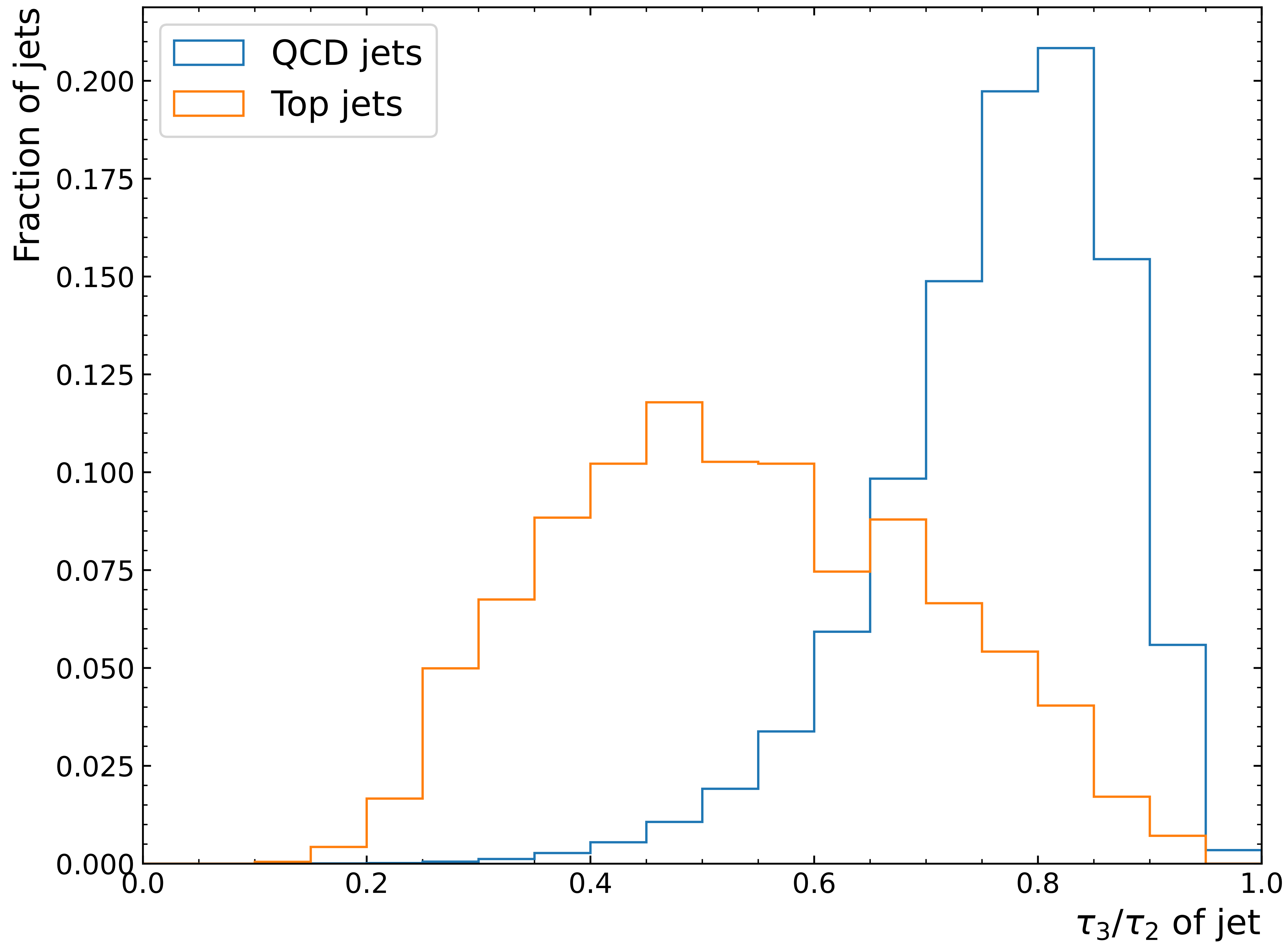
Project details

- Karol produced samples — simulated using the Delphes framework
  - **background** sample: QCD dijet events
  - **signal** sample: $t\bar{t}$ events
  - Jets are in boosted regime (decay products merged — “fat” jet)
- **Project goal**: Implement and train an autoencoder on the QCD jets and evaluate the performance at reconstructing top jets
  - Possible architectures:
    - deep neural network (on jet moments or jet images)
    - convolutional neural network (on jet images)
    - graph neural network (on jet constituents)
    - something else?

Autoencoders

Baseline for comparison

- Baseline: use **N-subjettiness**
  - Identify sub-structure of highly-boosted jets
  - $\tau_1 \ldots \tau_5$ saved as jet members
- For top jet, construct the **ratio** $\tau_{32} = \tau_3 / \tau_2$ which was found to be effective for 3-prong objects
- Vary cut on τ_{32} to adjust signal efficiency versus background rejection (shown later)

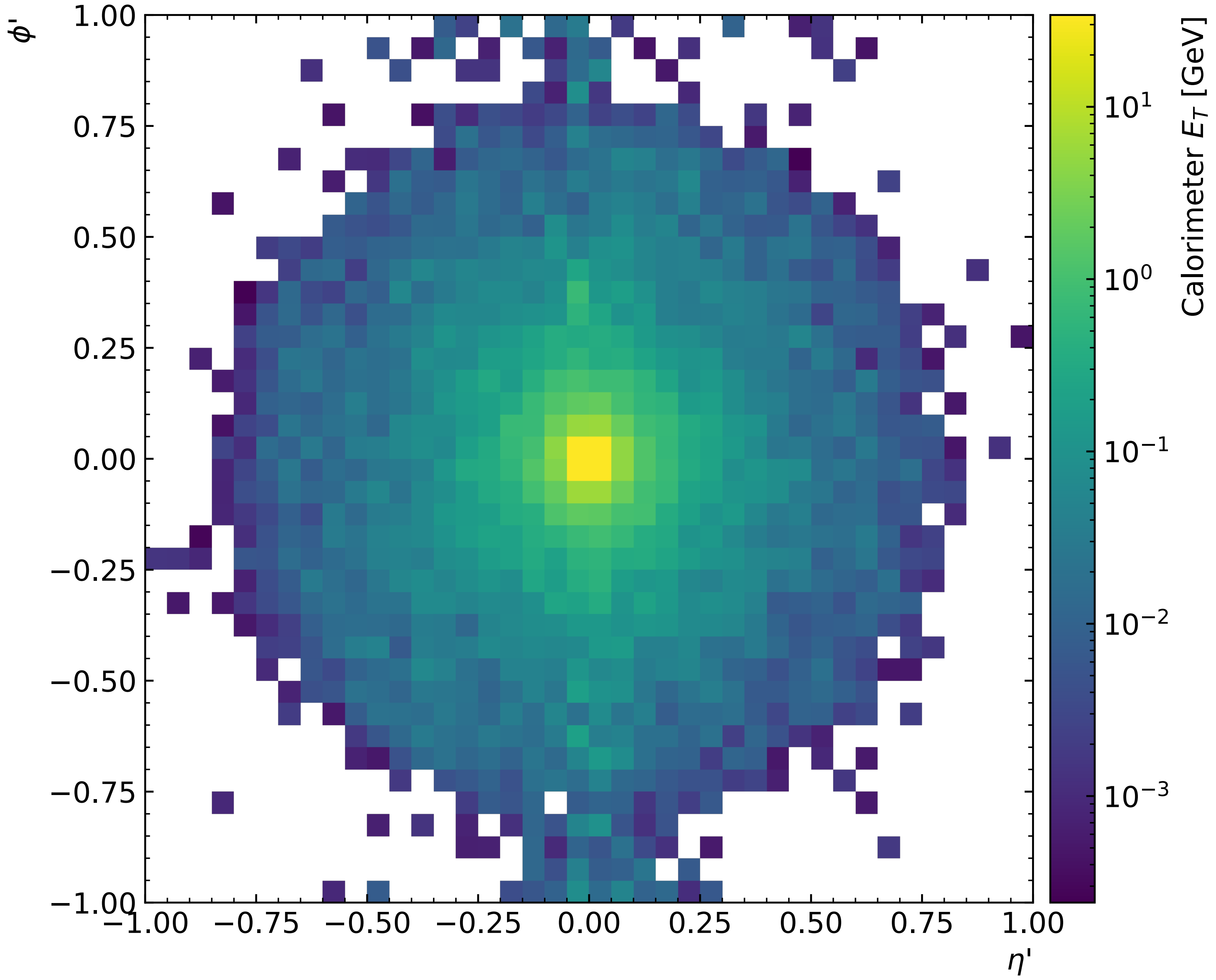


Autoencoder inputs

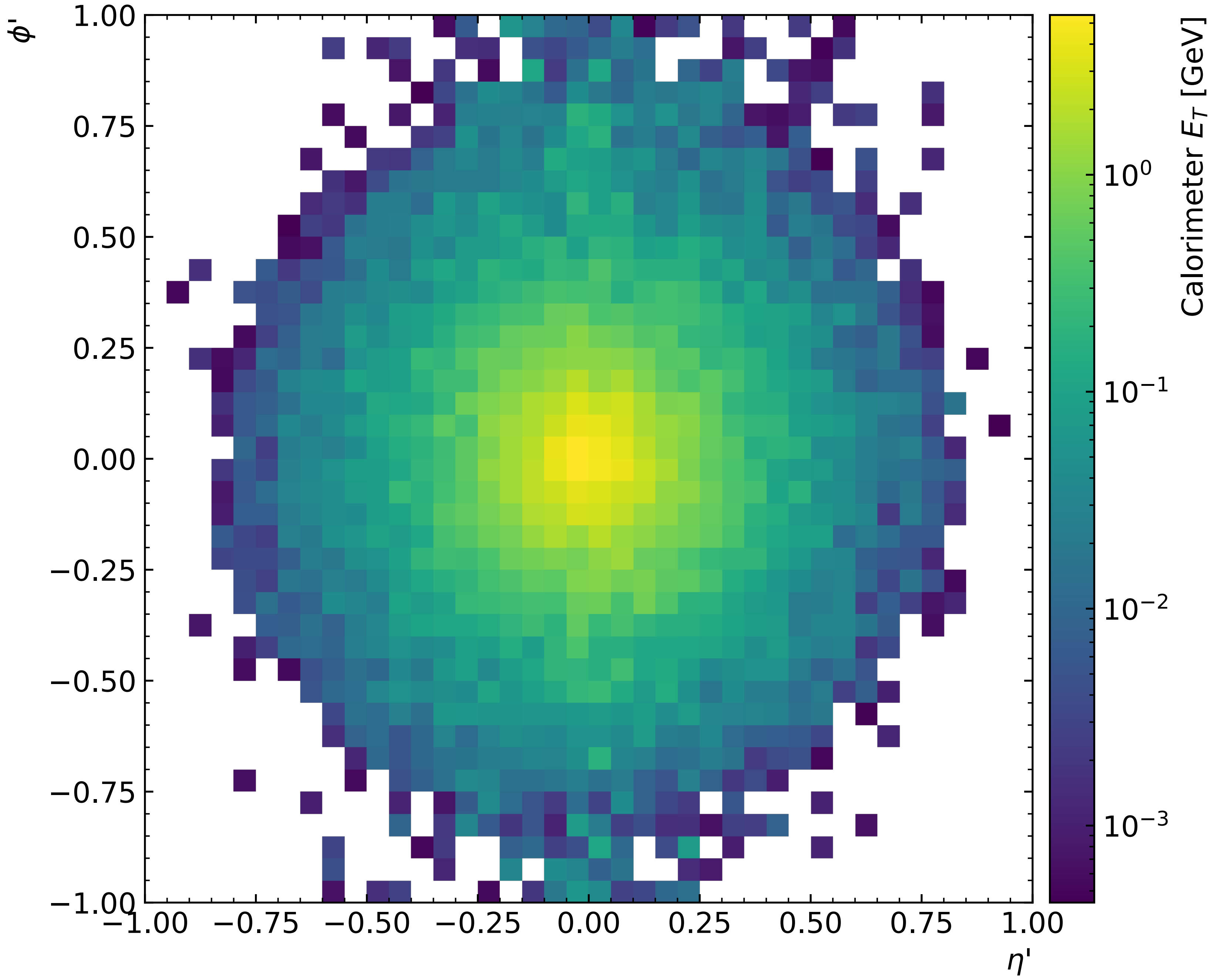
Jet images

- Choice: train autoencoder on **jet images** (from calorimeter E_{T})
  - Following Karol's suggestion for pre-selection: $p_{\mathrm{T}} > 500$ GeV, leading jet only
  - ~230k QCD jets, ~2.3k top jets
- (Raw) jet images already look different between background and signal

Background: (average) QCD jet



Signal: (average) top jet

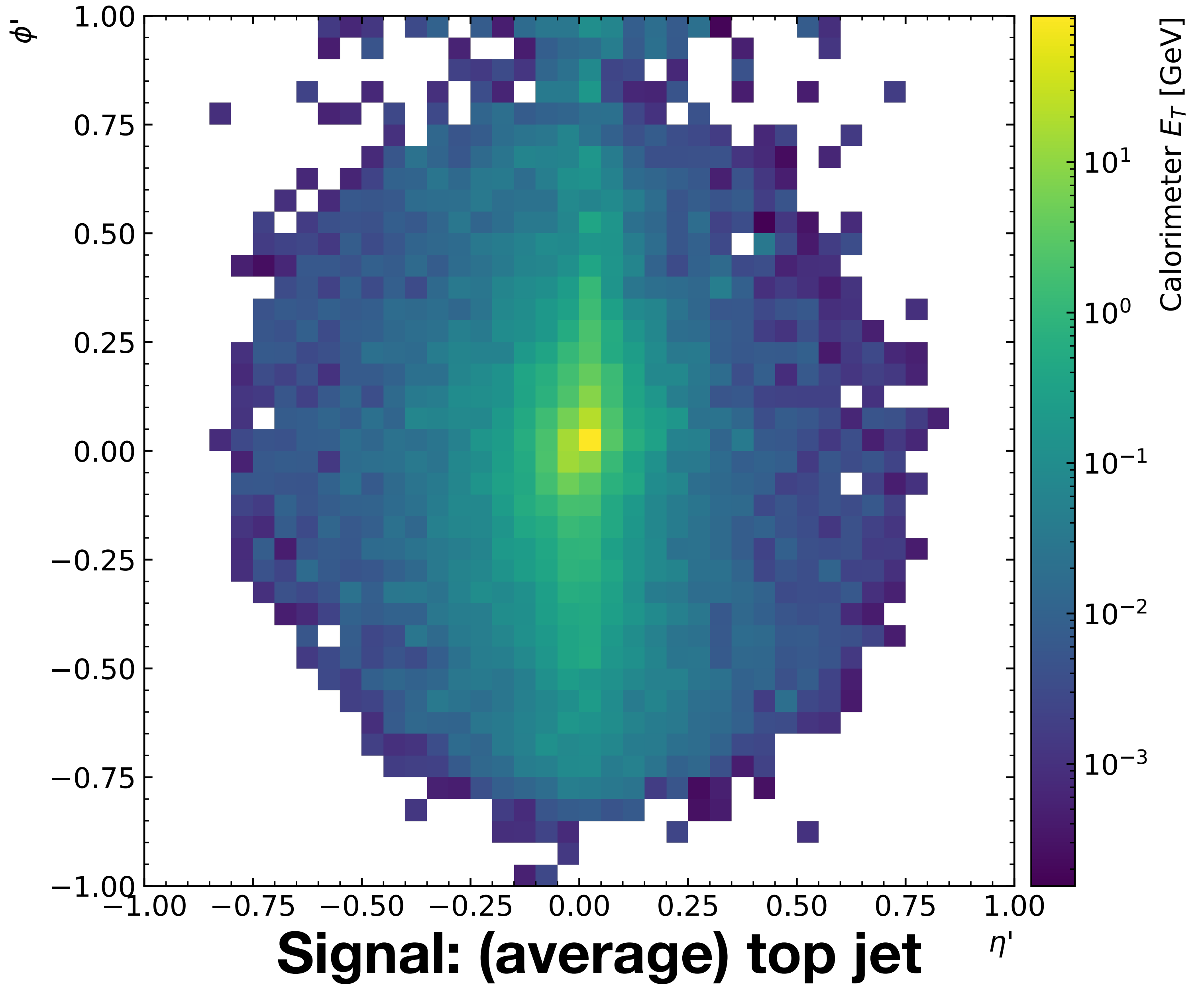


Autoencoder inputs

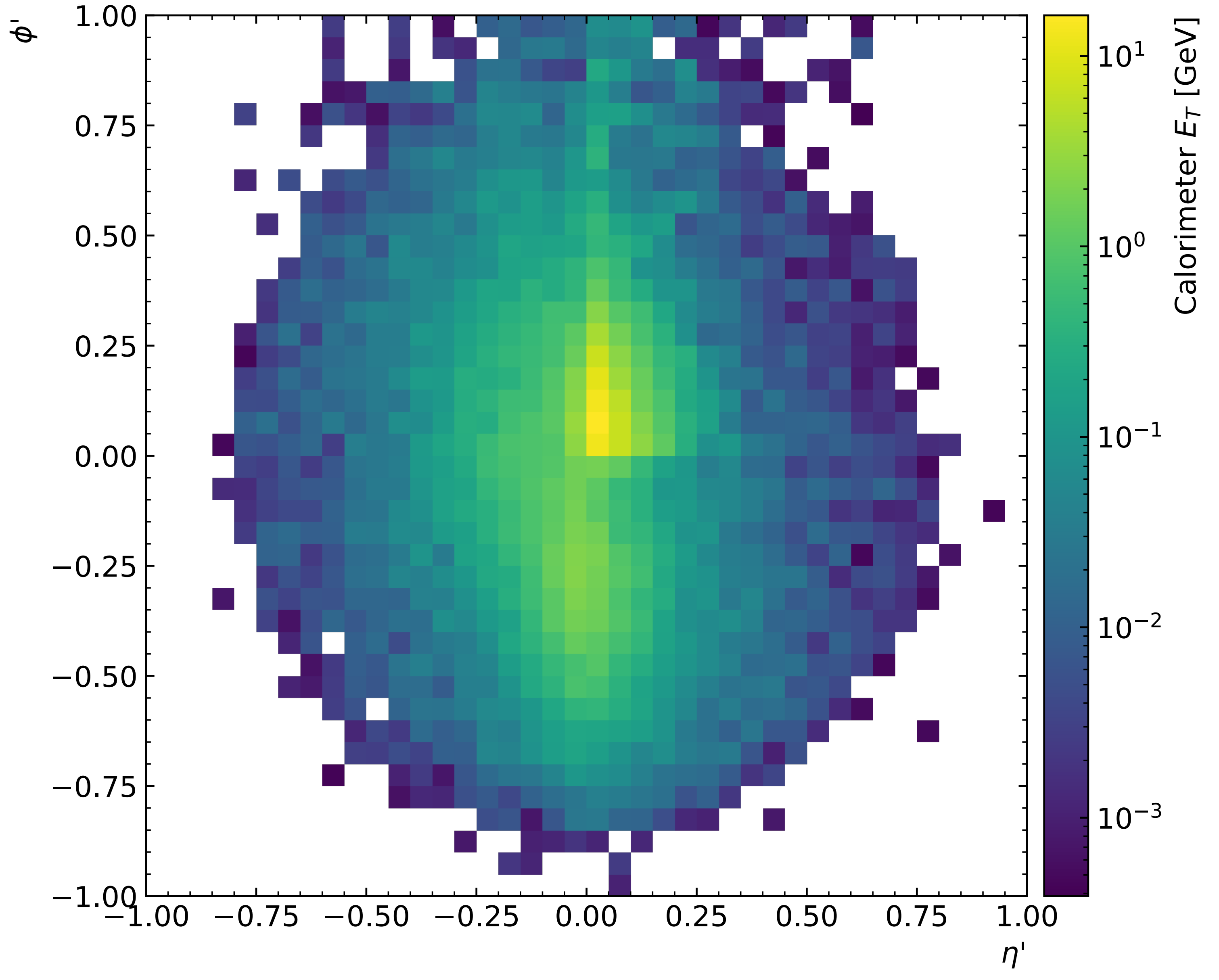
Jet images — pre-processing

- Several references (1808.08992, 1808.08979, 1511.05190, 1803.00107) adopt a **pre-processing step** (→ model should efficiently learn discriminating features between signal and background):
  - center (image centroid at origin)
  - rotate (image's principal axis vertical)
  - flip (U/D + L/R such that max. intensity in upper right quadrant)
  - pixelate and normalize (divide by total p_{T})
- After pre-processing, ideally more readily apparent the 3-prong substructure of top jets vs dipole-like for QCD jets

Background: (average) QCD jet



Signal: (average) top jet



Autoencoder architecture

- Choice: **convolutional neural network** (trained on jet images)
- Use same architecture as reference paper from Karol
- Input: 40x40 jet image
- Encoder: Conv2D(128, relu), MaxPooling2D(2x2), Conv2D(128, relu), MaxPooling2D(2x2), Conv2D(128, relu), Dense(32, relu)
- Latent representation layer: Dense(encoding dimension) → see next slide
- Decoder: Dense(32, relu), Dense(12800, relu), Conv2D(128, relu), UpSampling2D(2x2), Conv2D(128, relu), UpSampling2D(2x2), Conv2D(1), Activation(softmax)
- Output: reshape to 40x40 image

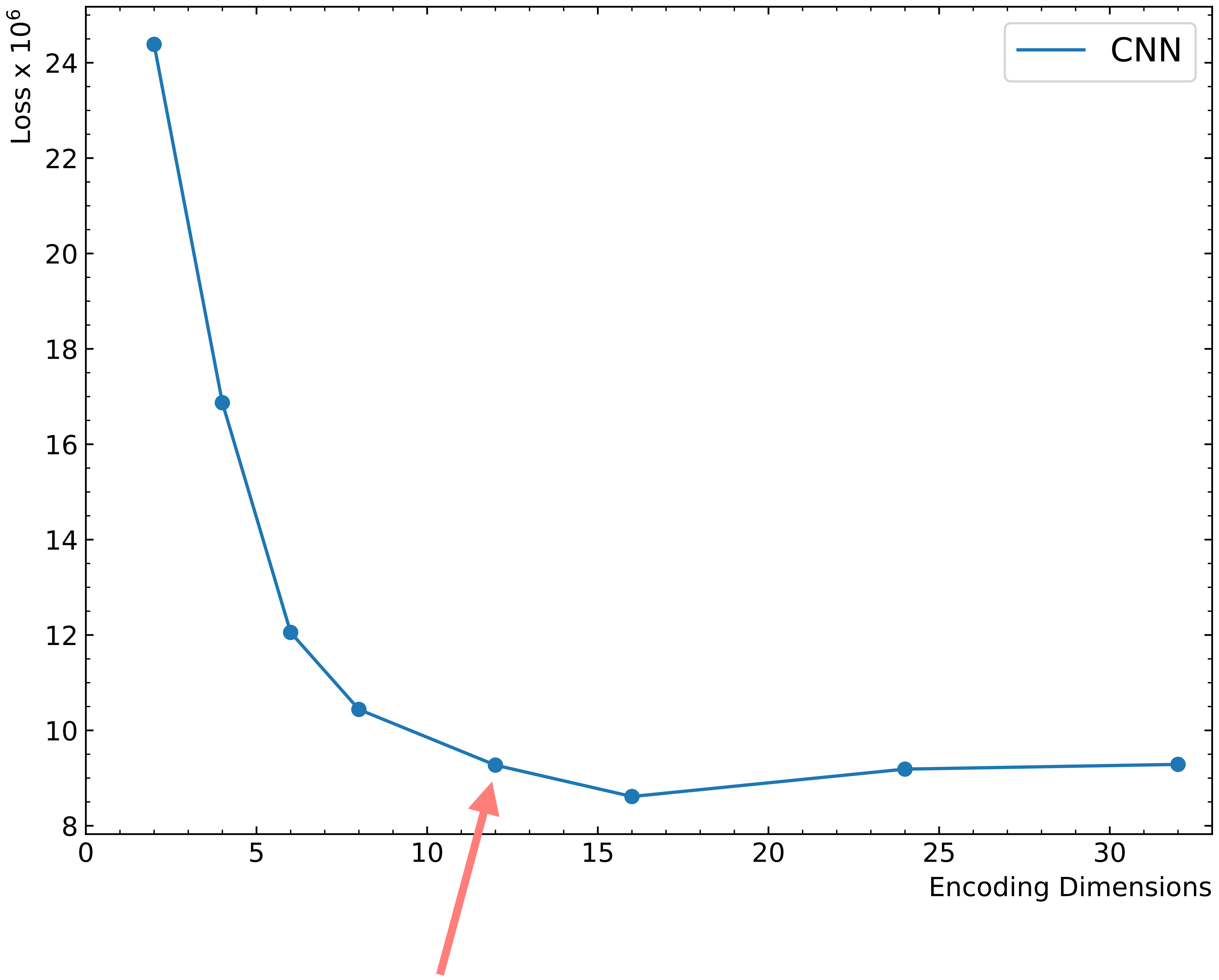
Autoencoder architecture

Finding the encoding dimension

- Vary the encoding dimension
- For each choice, run 5 independent trainings on 100k QCD jets
- Compute average loss across the 5 trainings
- Loss sharply decreases as more important features are learned by the latent representation and eventually starts to flatten out
- Choose **encoding dimension = 12** (~elbow of the curve)

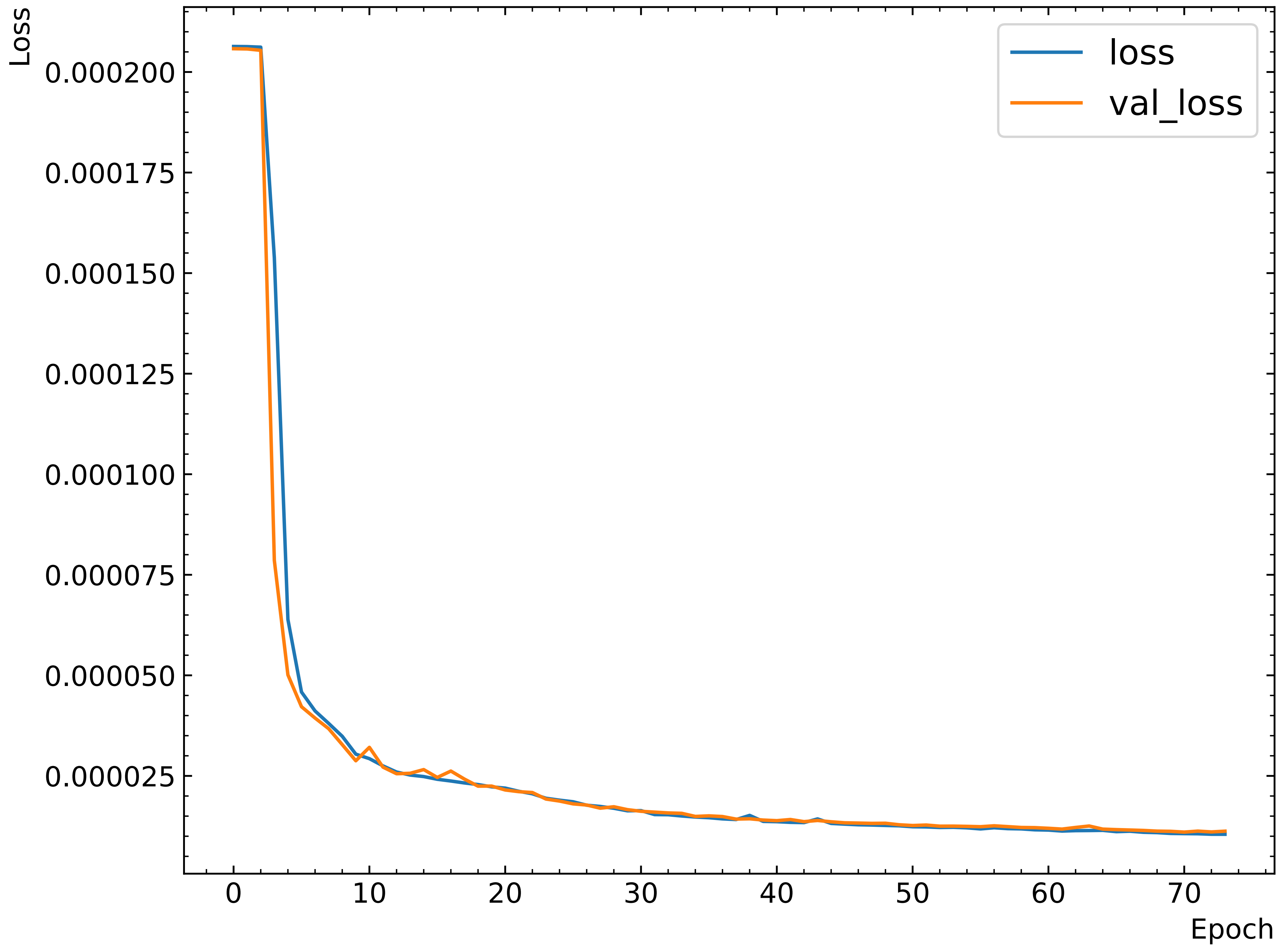
Dimension low → autoencoder can't learn all relevant features (fast to run)

Dimension high → approach trivial representation (slower to run)



Autoencoder training

- Using Tensorflow 2.7.0 and Keras
- 100k QCD jets for training, 25k QCD jets for validation
- Adam optimizer with default settings
- Batch size = 1024
- Loss = mean squared error
- Early stopping criterion: threshold = 0, patience = 3 (e.g. stop if validation loss hasn't decreased in the past 3 epochs)



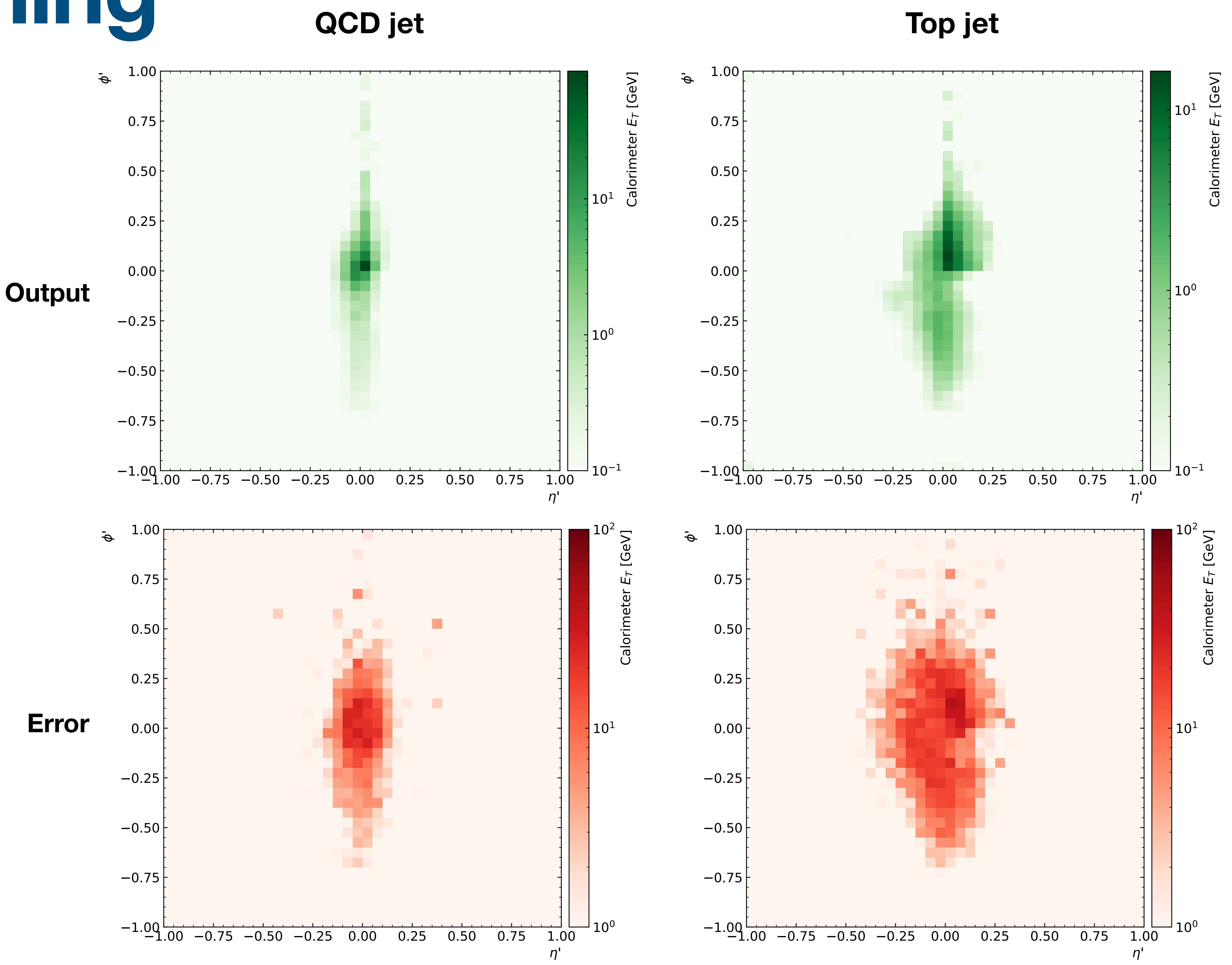
Autoencoder training

Performance evaluation

- Evaluate on a different set of 100k QCD jets and the set of top jets

- Output images for QCD and top jets look similar to the input images

- But, we can see that reconstructed top jet images have **larger (squared) errors** compared to QCD jets



Autoencoder training

Performance evaluation

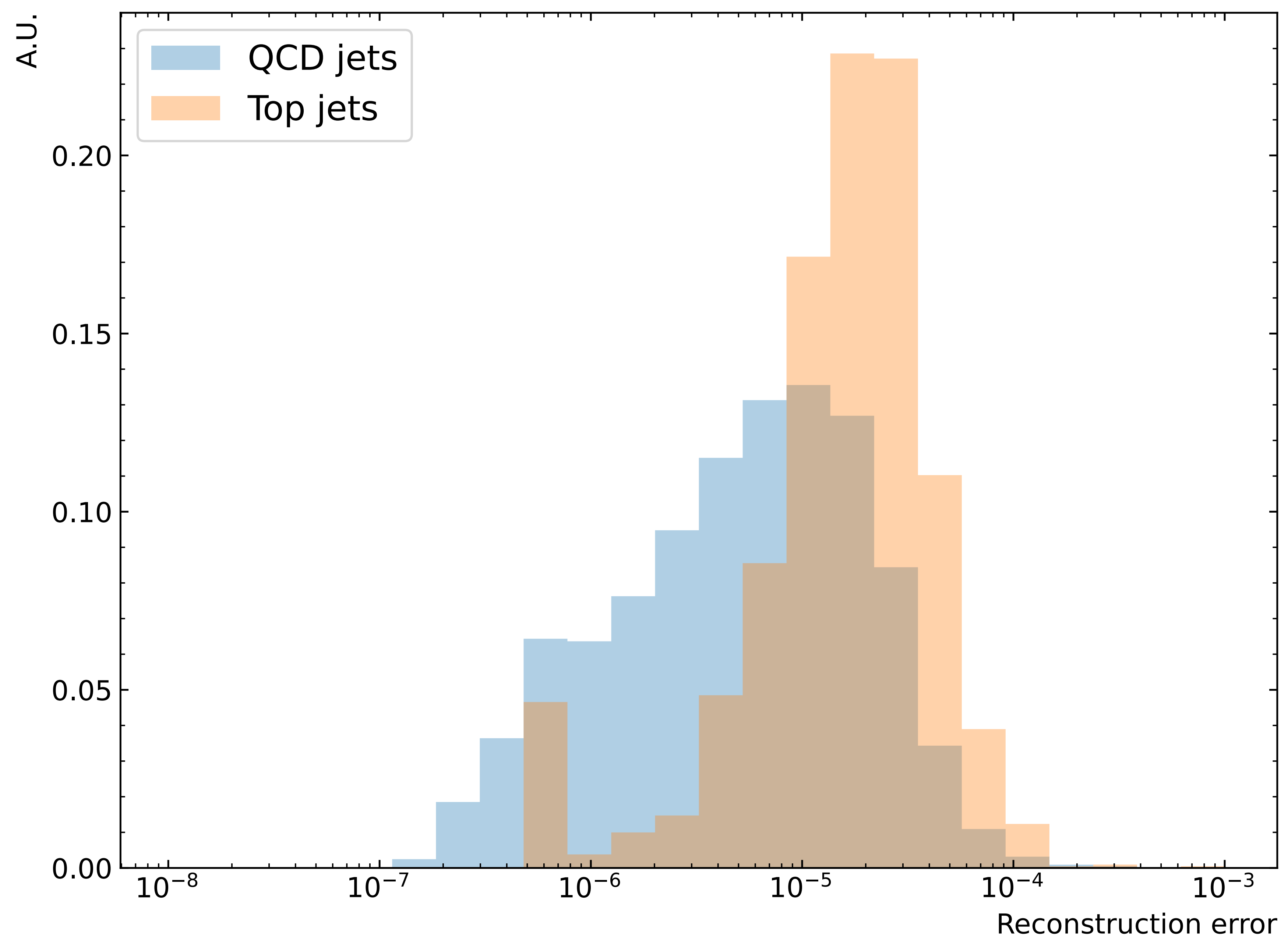
- Evaluate on a different set of 100k QCD jets and the set of top jets

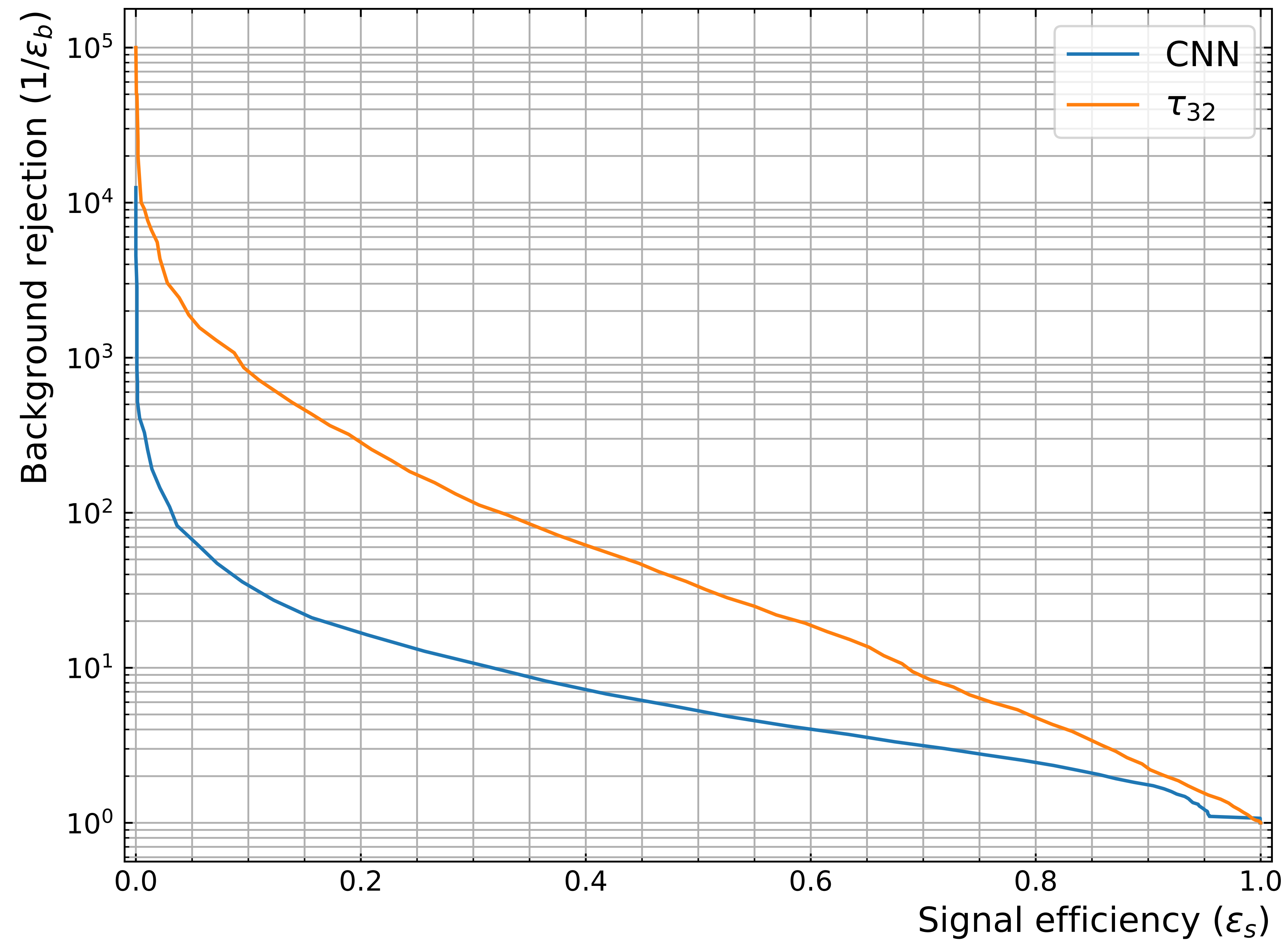
- Reconstruction error = average pixel-wise squared difference between input and output images

- Top jets have **slightly higher reconstruction error** than QCD jets

  - 10% signal efficiency with factor of ~35 background rejection

- Unfortunately, doesn't outperform a cut on τ_{32} variable

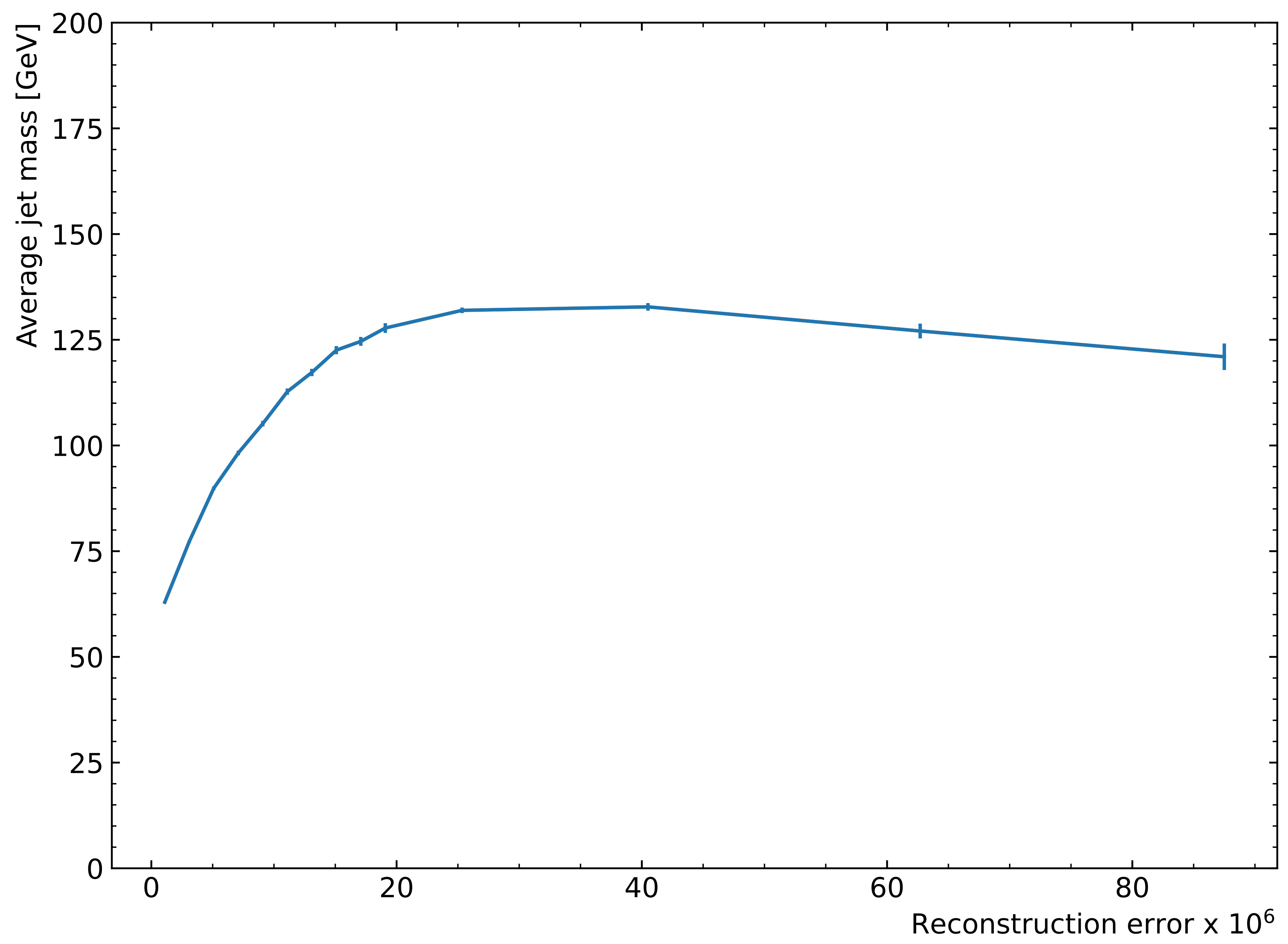


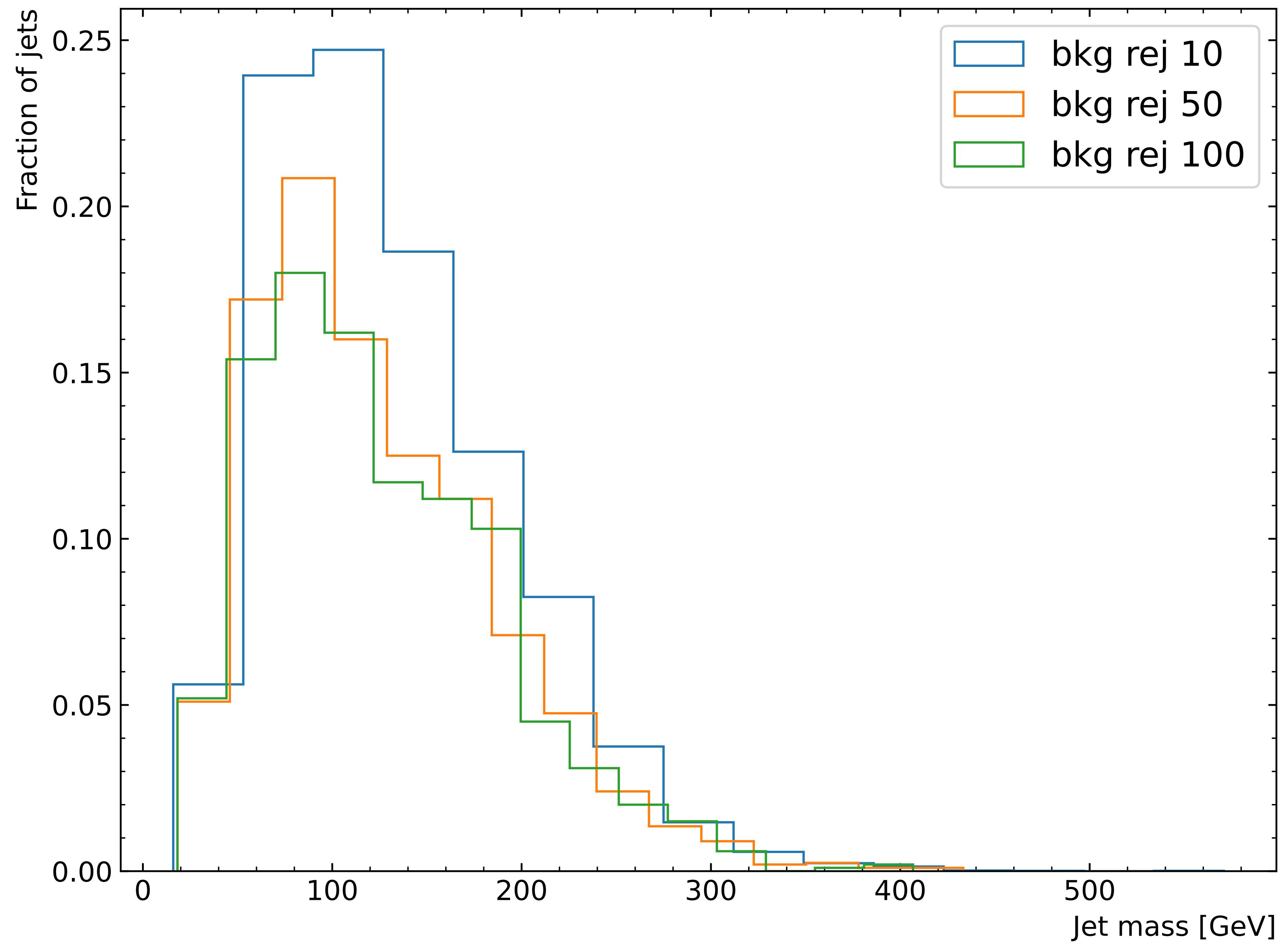


Autoencoder performance

Why CNN?

- QCD background estimates are data-driven; side band in jet mass
  - Avoid artificially shaping the mass distribution
- Advantage of CNN autoencoder is that it shows **little correlation between the reconstruction error and the jet mass**
- Jet mass distribution has ~same shape for increasingly tighter cuts on reconstruction error (e.g. bkg rej 50 and 100)





Conclusion

- Goal of the project was to train an autoencoder to learn what QCD jets “look like” and then evaluate the performance of the autoencoder at reconstructing top jets that it hadn’t seen before

- For the autoencoder architecture, used a convolutional neural network trained on jet images with latent representation of dimension 12

- CNN autoencoder reconstruction error is indeed larger for top jets

- CNN autoencoder still under-performs a cut on N-subjettiness variable ratio $\tau_{32} \rightarrow$ more optimization work need

- Advantage of CNN autoencoder is that its reconstruction error decorrelates with the jet mass $\rightarrow$ CNN can be used to reduce QCD background and look for a bump in jet mass (from new physics)

References

- Github repo with code: https://github.com/irinaene/ph290e-autoencoders
- Resources:

  - **Searching for New Physics with Deep Autoencoders** arXiv:1808.08992 [**hep-ph**]
  - **QCD or What?** arXiv:1808.08979 [**hep-ph**]
  - **Jet-Images — Deep Learning Edition** arXiv:1511.05190 [**hep-ph**]
  - **Pulling Out All the Tops with Computer Vision and Deep Learning** arXiv:1803.00107 [**hep-ph**]
  - **Deep-learning Top Taggers or The End of QCD?** arXiv:1701.08784 [**hep-ph**]
  - **Jet-Images: Computer Vision Inspired Techniques for Jet Tagging** arXiv:1407.5675 [**hep-ph**]

Backup

Jet Images

Effect of pre-processing

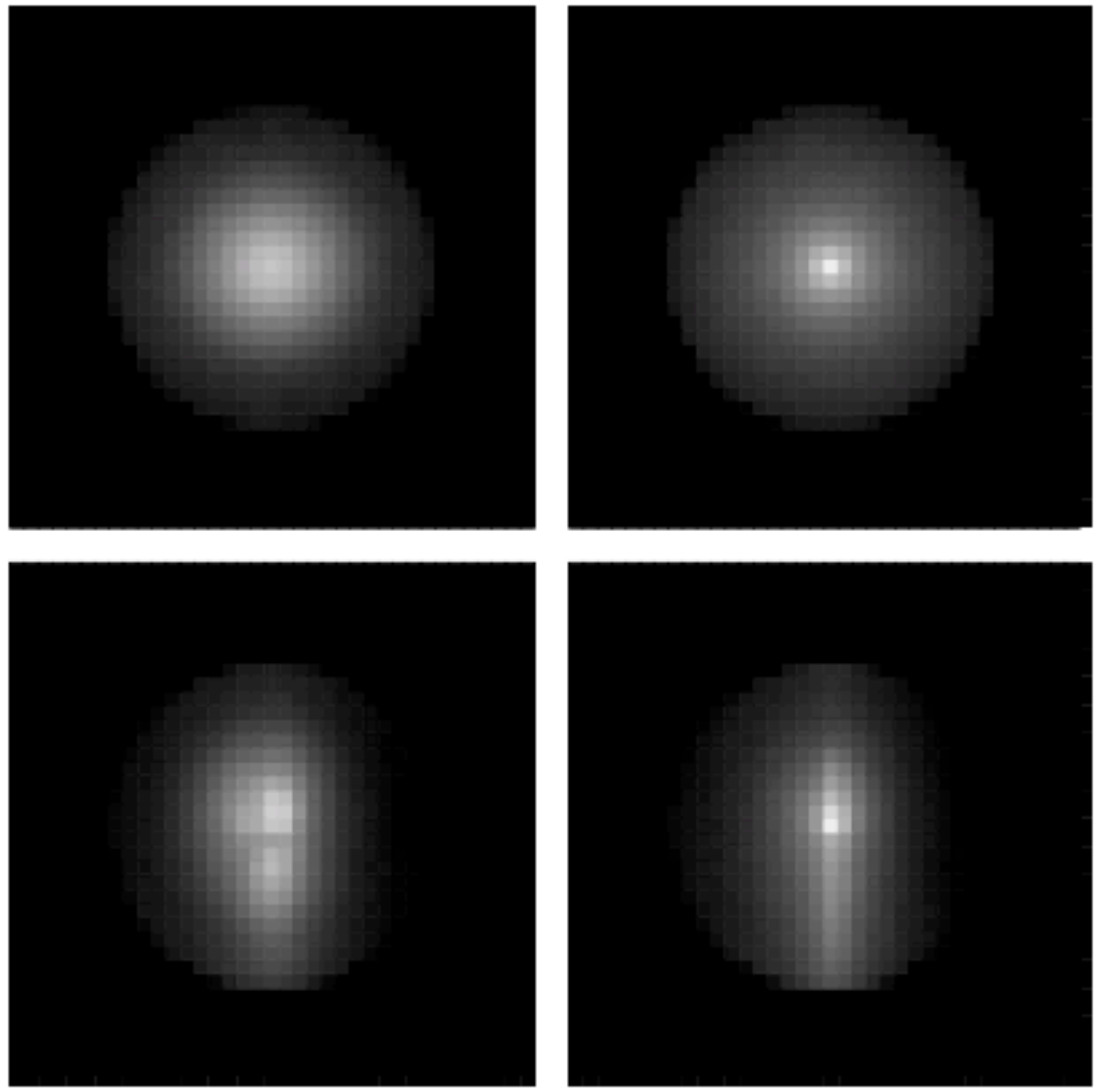


Figure 2: The average of 100k jet images drawn from the CMS sample (37 × 37 pixels spanning $\Delta\eta = \Delta\phi = 3.2$). The grayscale intensity corresponds to the total p_T in each pixel. Upper: no preprocessing besides centering. Lower: with full preprocessing. Left: top jets. Right: QCD jets

arXiv:1803.00107

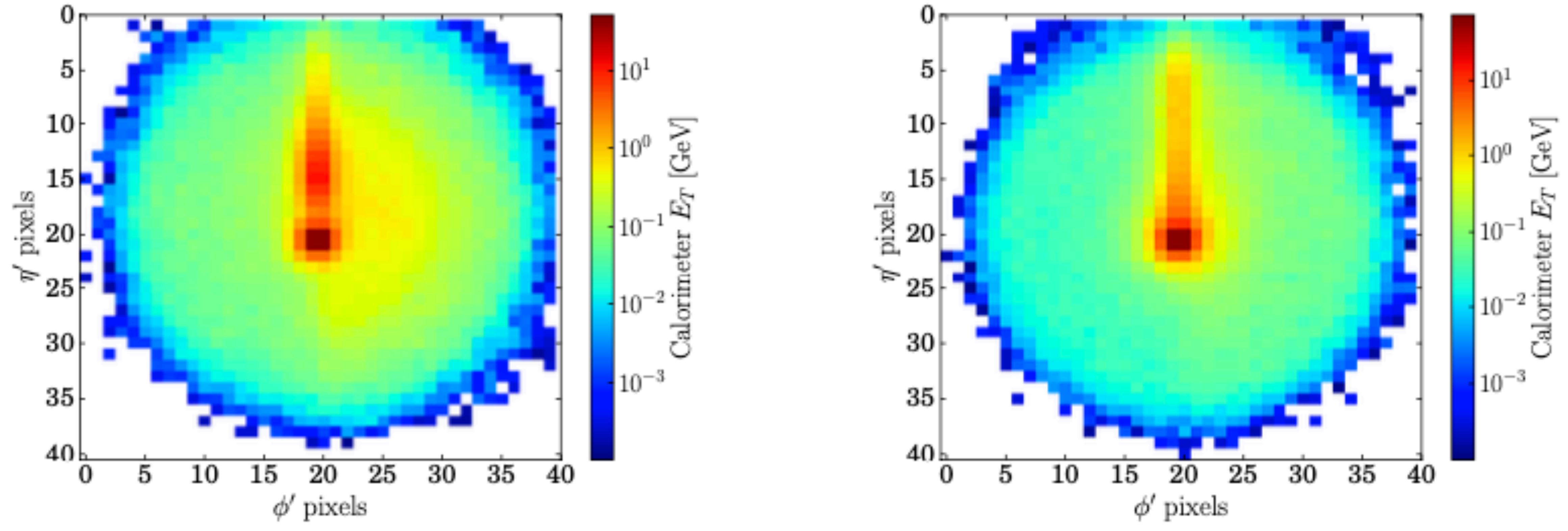


Figure 1. Jet image after pre-processing for the signal (left) and background (right). Each picture is averaged over 10,000 actual images.

arXiv:1701.08784