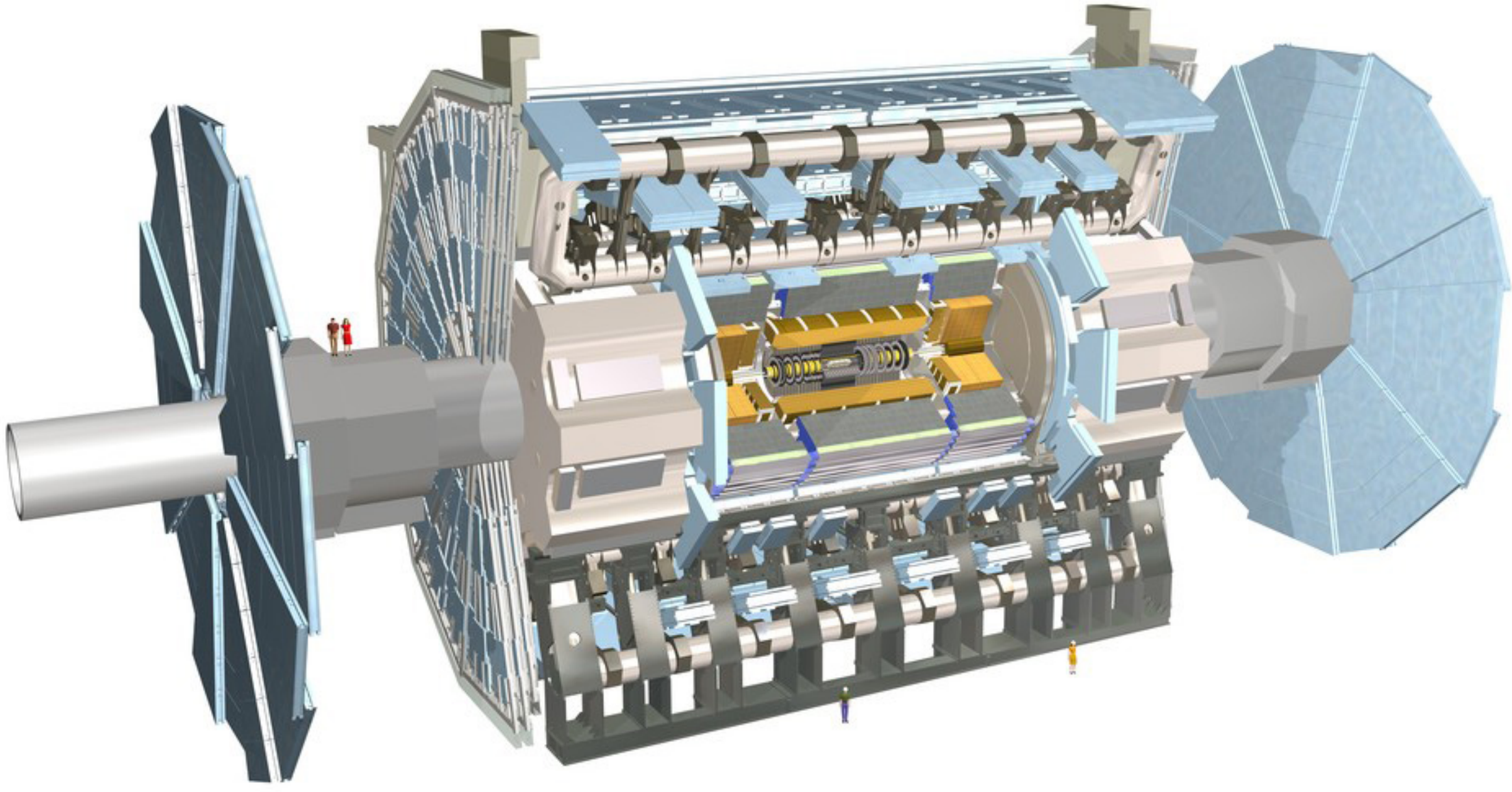
Adversarial neural networks

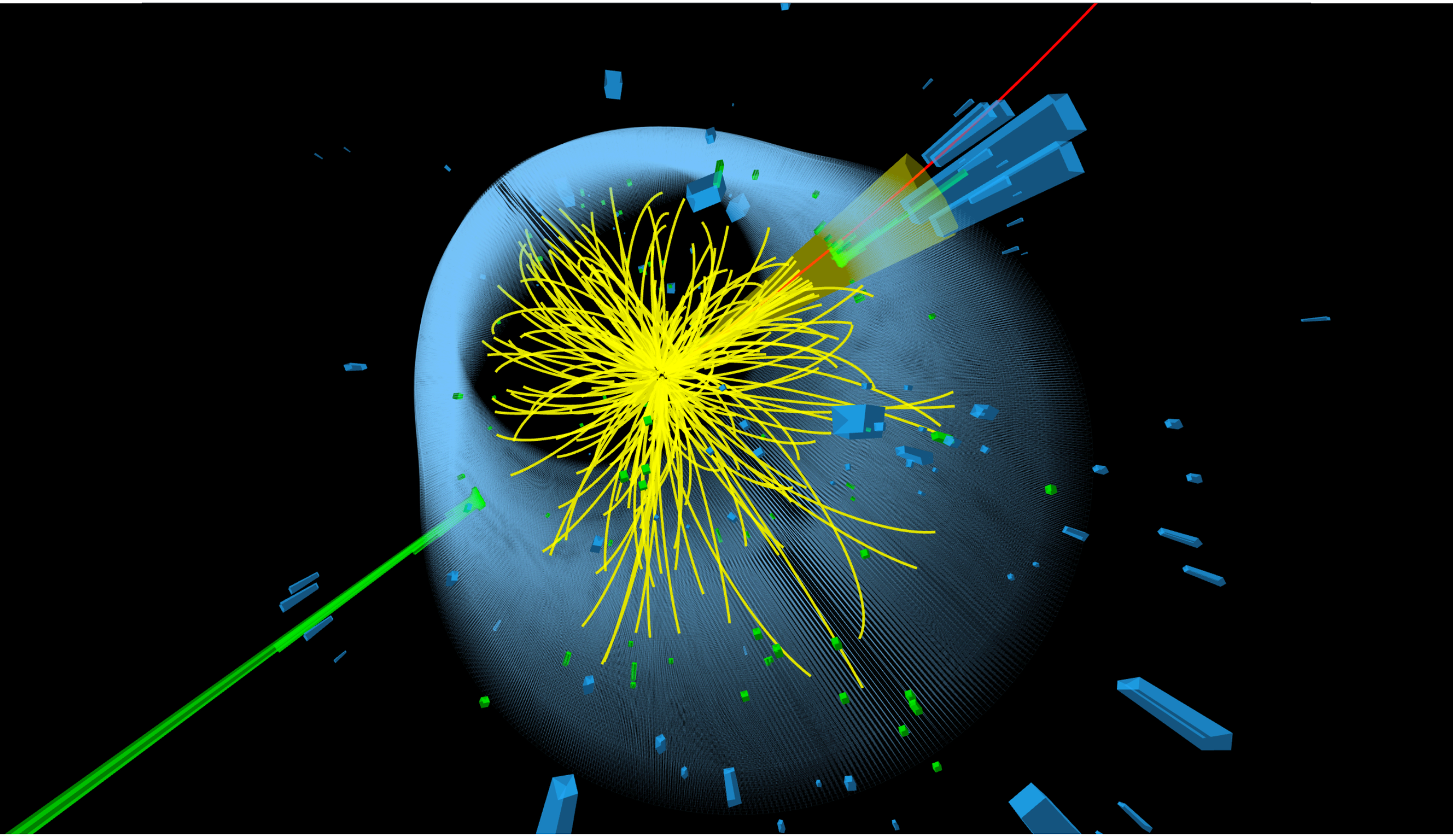
PPE all-group Christmas Meeting / 14 December 2018 🎄

Andreas Søgaard / University of Edinburgh

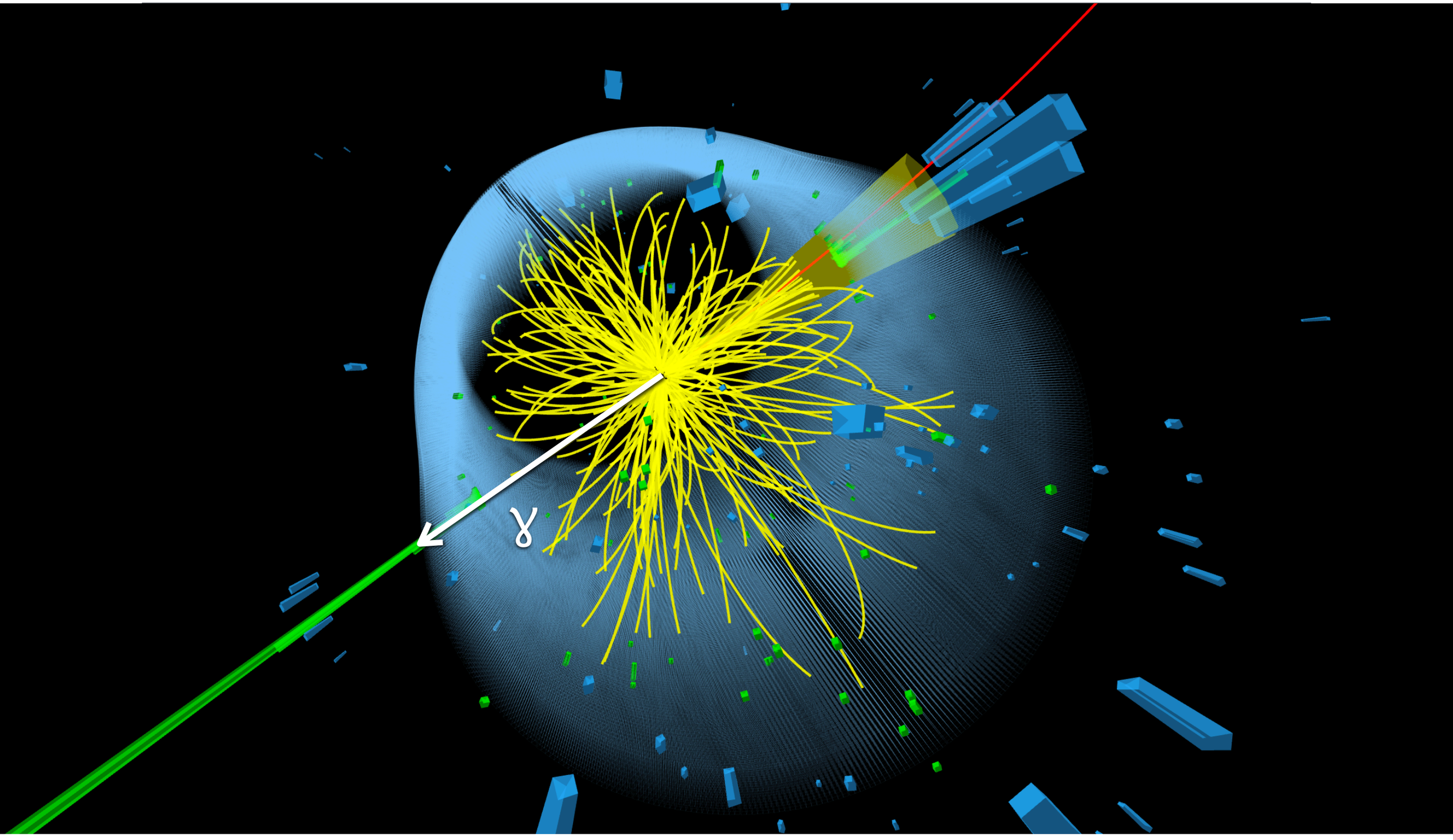
ATLAS Experiment



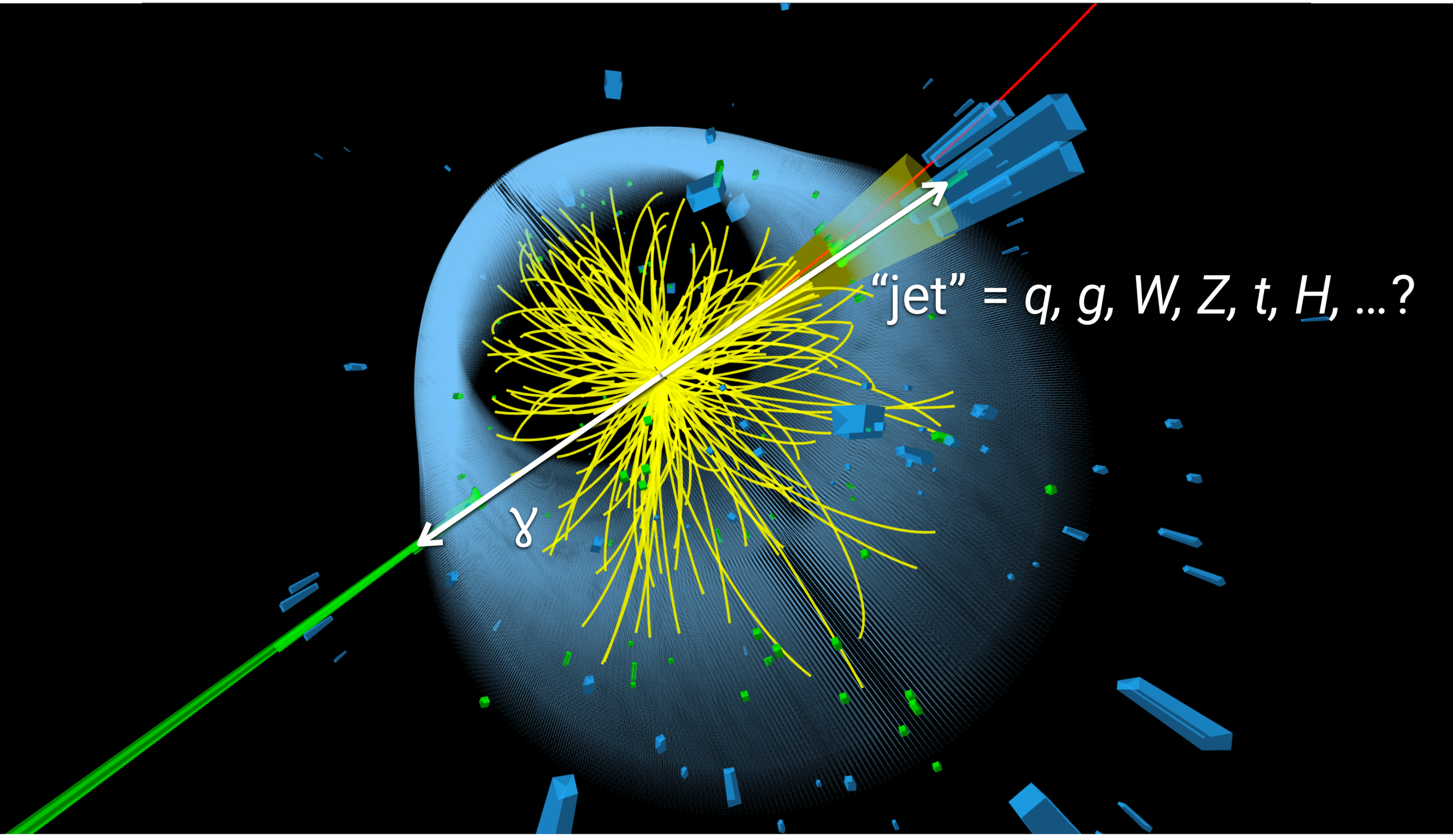
Object identification



Object identification

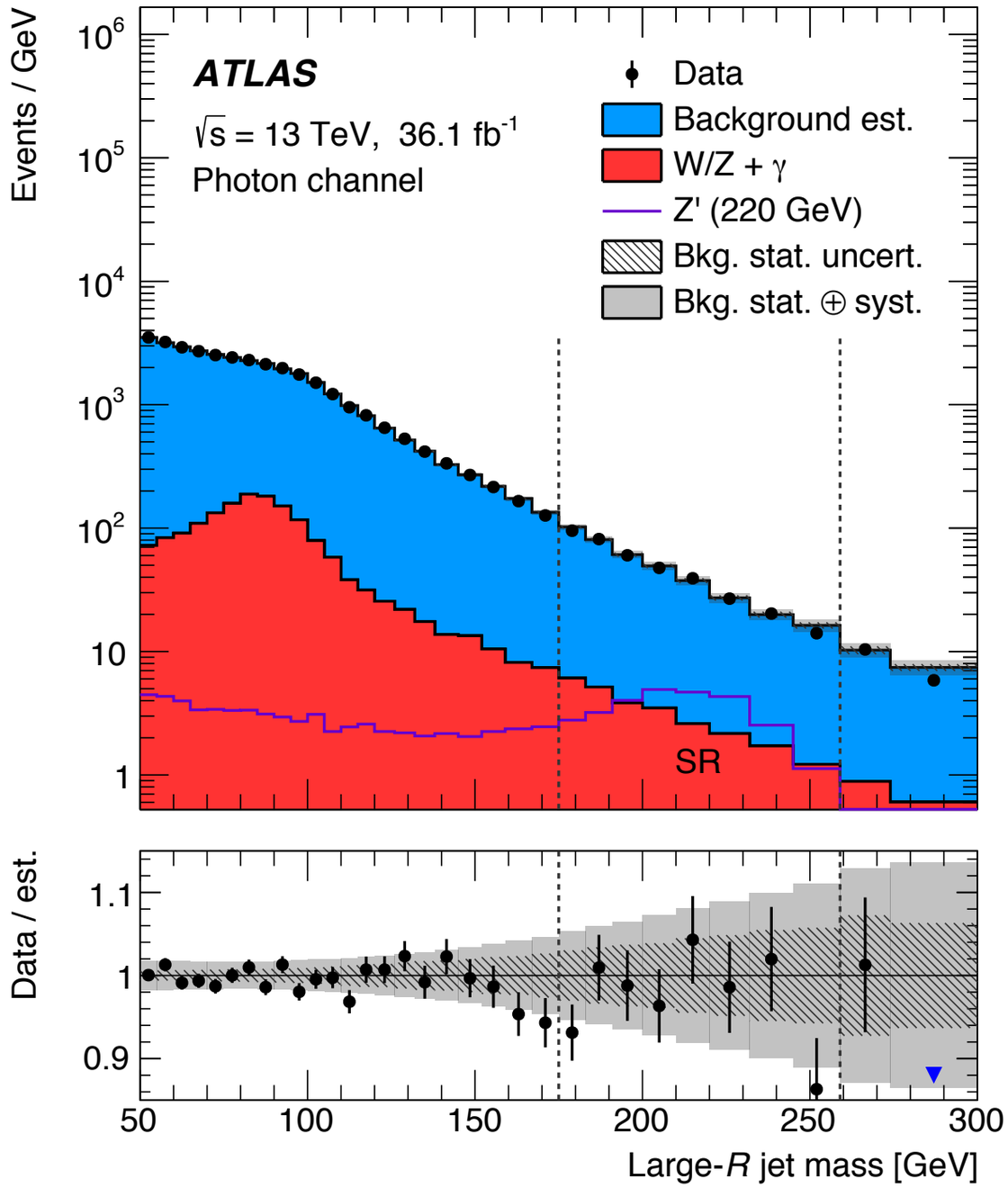


Object identification



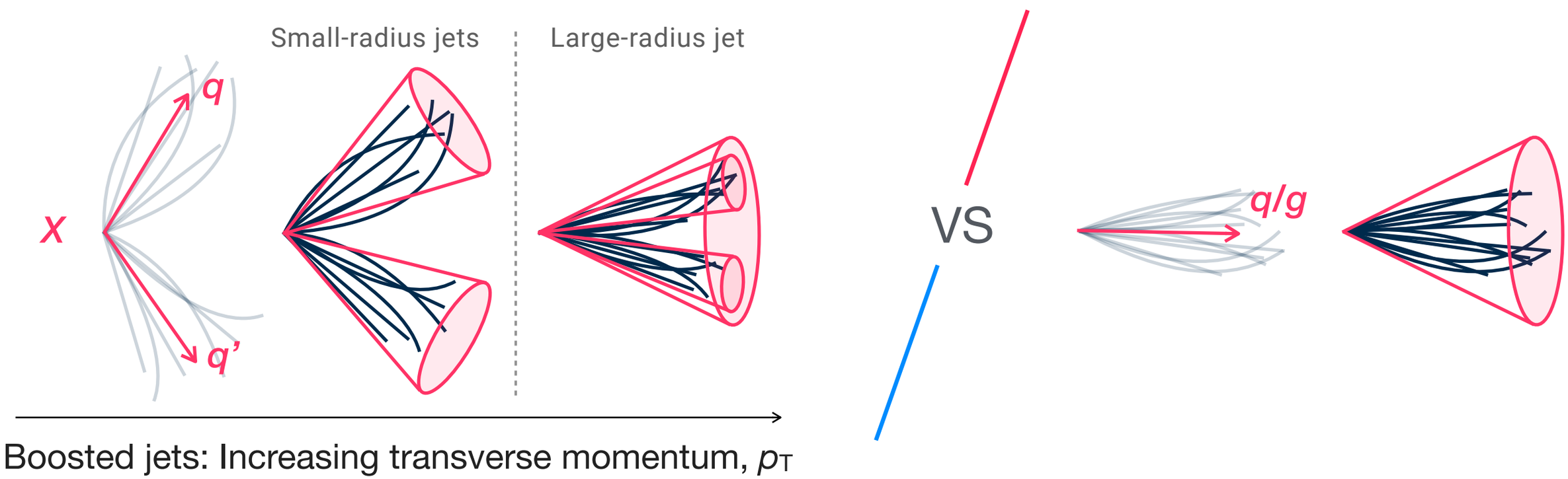
Searches using jets

- New physics can manifest as “jets,” e.g. **Dark Matter mediator particles**
- **Search for “bumps”** in mass spectrum
- **Challenge:** Distinguishing rare, new physics from vast Standard Model background



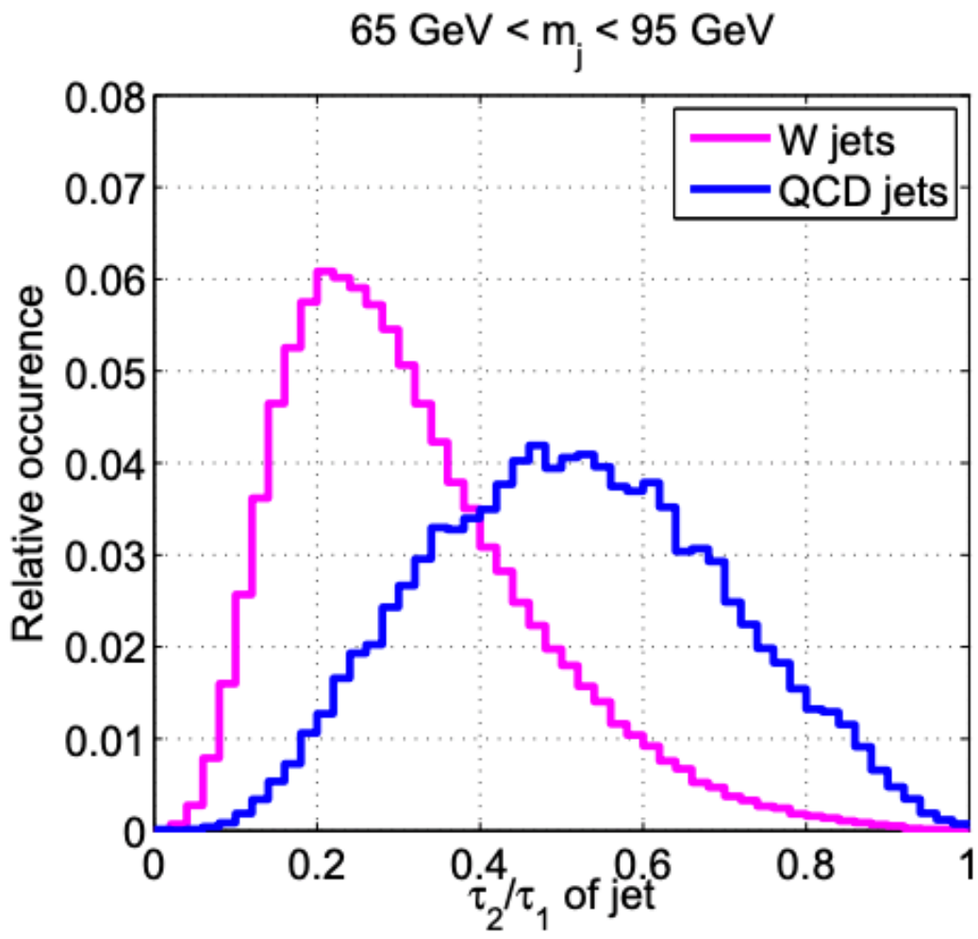
Jet substructure

- Typically:
  - searching for resonant hadronic **two-body decay** ($X \rightarrow qq'$)
  - background from non-resonant, **single-parton emission**
- Use the **substructure** of the jet to perform distinction: more two- (signal) or one- (background) “subjet-like”?



Jet classification

- Substructure observables measure radiation patterns directly
- For instance, *N*-subjettiness measures likelihood of *N*-subjet hypothesis
- Ratio $\tau_{21} = \tau_2 / \tau_1$ is good discriminator for jets from two-body decays, e.g. *W*
- Countless such variables exist, and are non-redundant



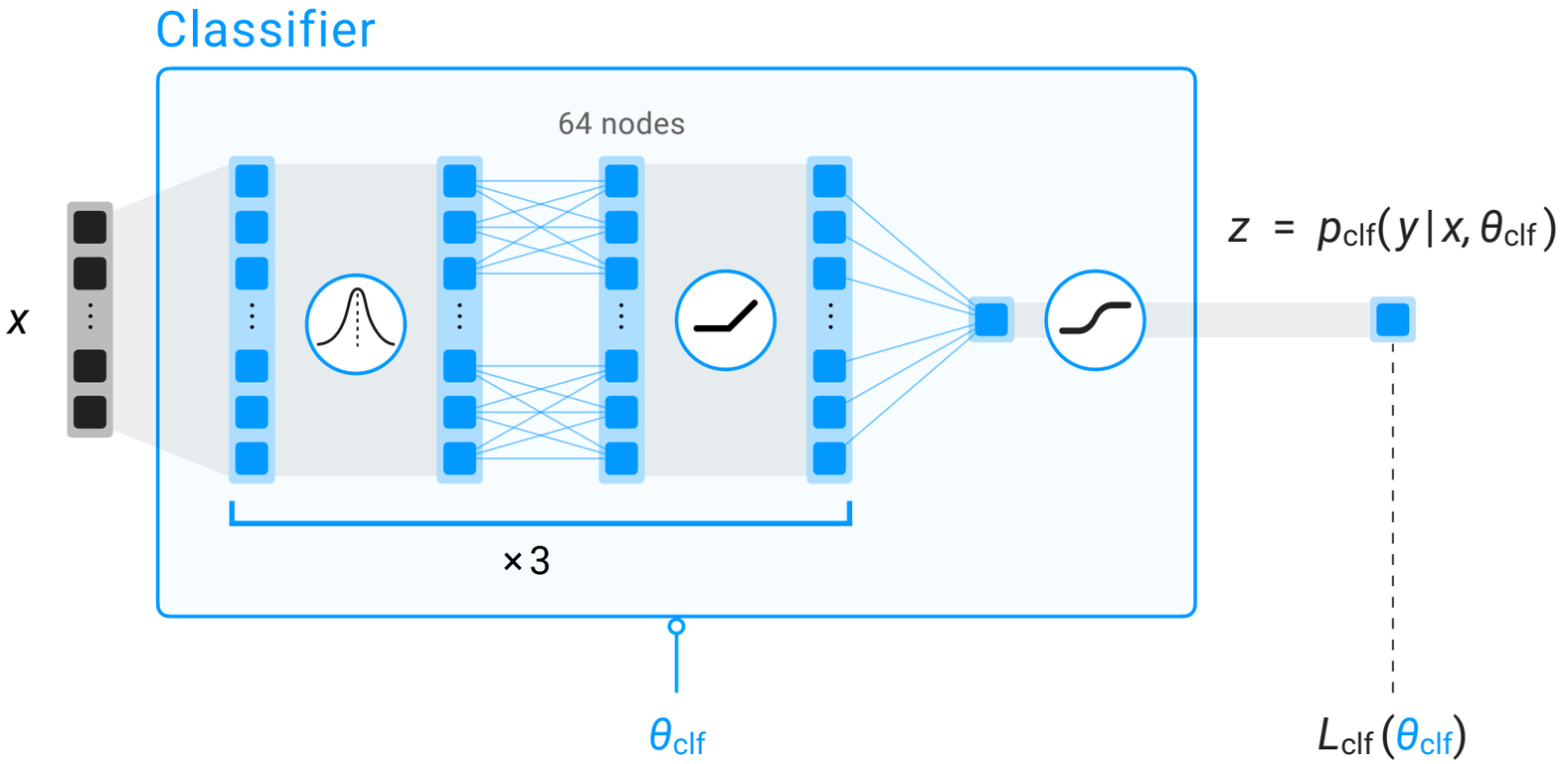
Jet classification

- To improve classification further, use densely-connected **neural network** to combine N_{feat} jet substructure observables ("weak discriminators") to single, powerful **jet classifier**

- ***x*:** Observables

- ***y*:** Jet label
  sig. = 1
  bkg.= 0

- $\boldsymbol{\theta_{\text{clf}}}$**:** Network weights

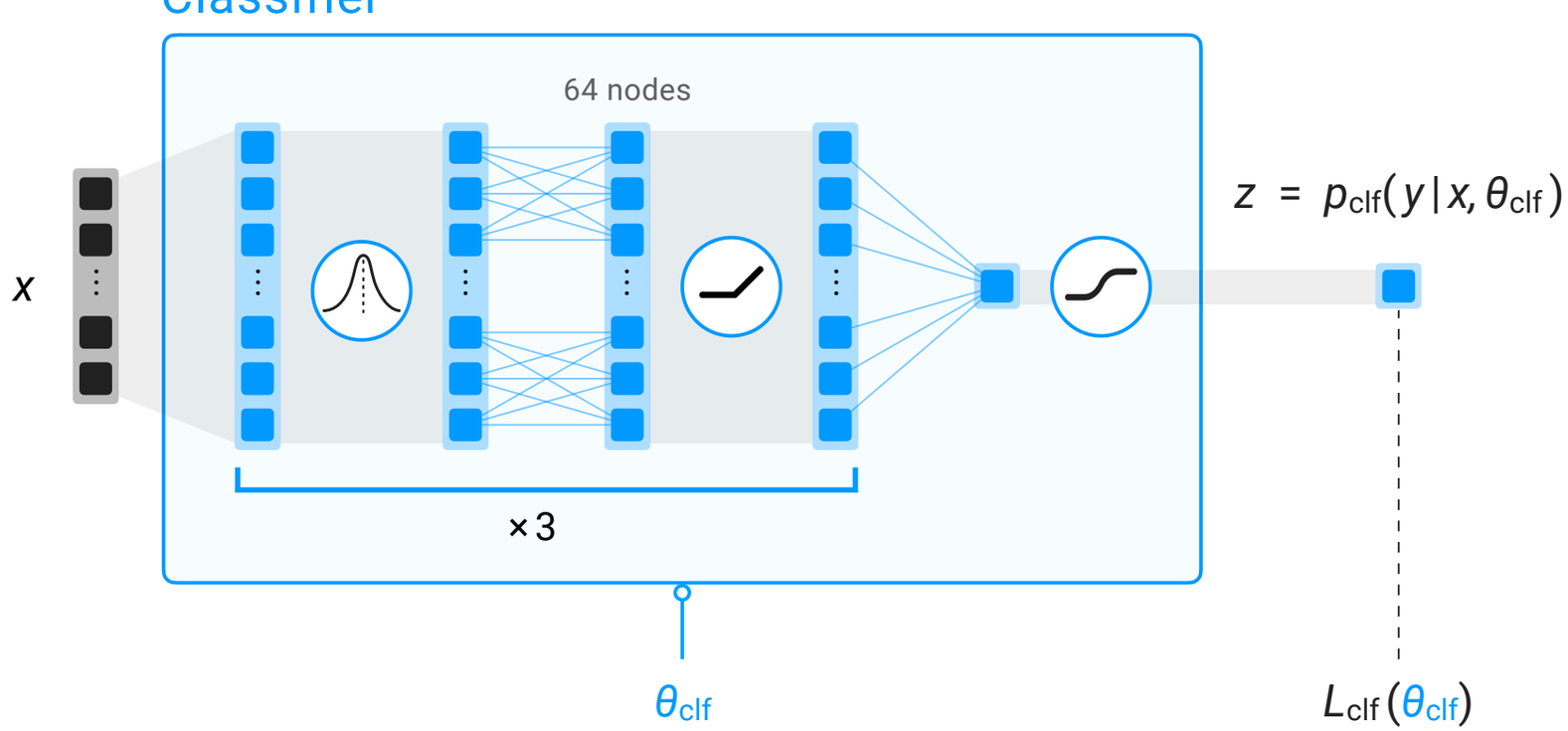


- $\boldsymbol{p_{\text{clf}}(y \mid x, \theta_{\text{clf}})}$**:** NN classifier observable

- $\boldsymbol{L_{\text{clf}}}$**:** Classifier training loss

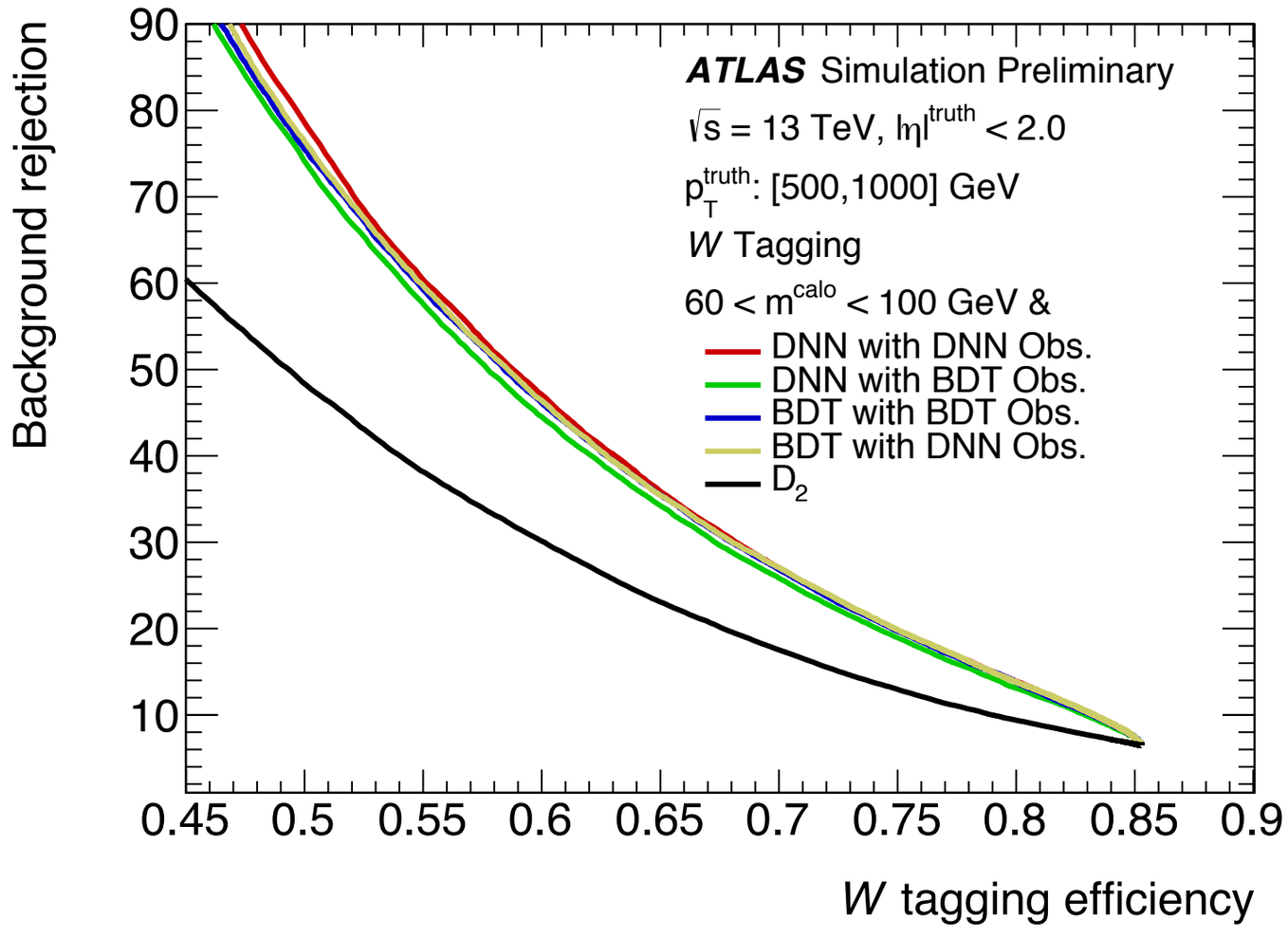
Jet classification

- In practice:
  - Data is regular numpy data array of shape (N_{jets}, N_{feat})
  - Classifier is created with Keras, trained with binary cross-entropy loss (signals jet vs. background jets)
  - Typically using $N_{\text{feat}} \approx 10$ high-level features as input



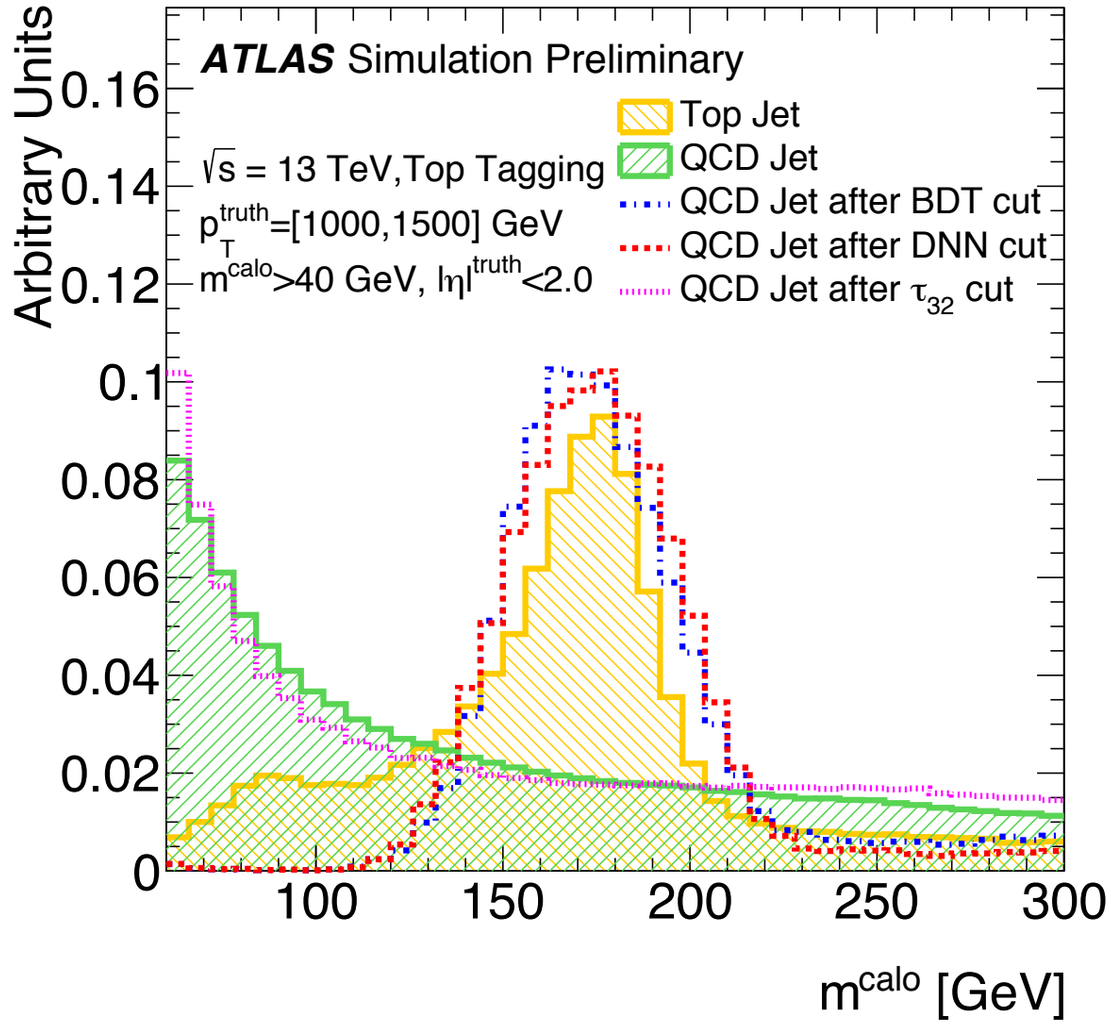
Jet classification

- Compared to individual substructure observables (here: D_2), NN classifiers increase background rejection (*y*-axis) at similar levels of signal efficiency (*x*-axis)



Problem

- *Neural networks are smart…*
- **Task:**
  - Distinguish signal ( ) from background ( )
- **Observation:**
  - Signal has fixed mass, background doesn't
  - Mass is good predictor!

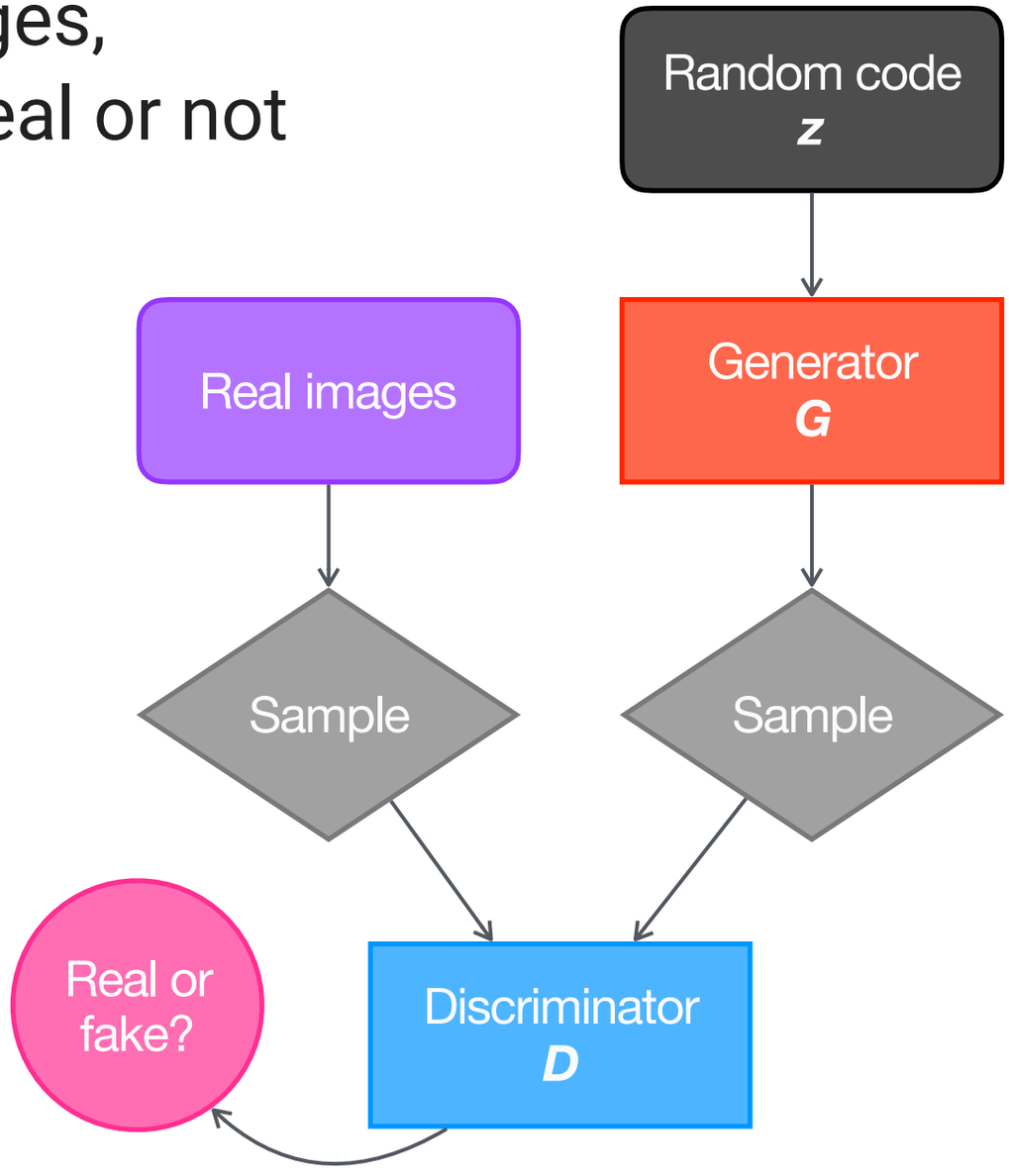


- NN classifier output is highly **correlated with the jet mass**
- Background jets passing the classification ( ) are indistinguishable from signal → **"bump" search impossible!**

Aside: Generative adversarial NNs

- In computer vision, **adversarial neural networks** have been used to generate synthetic images of people
- **Generator** produces synthetic images, **discriminator** tries to guess if it's real or not

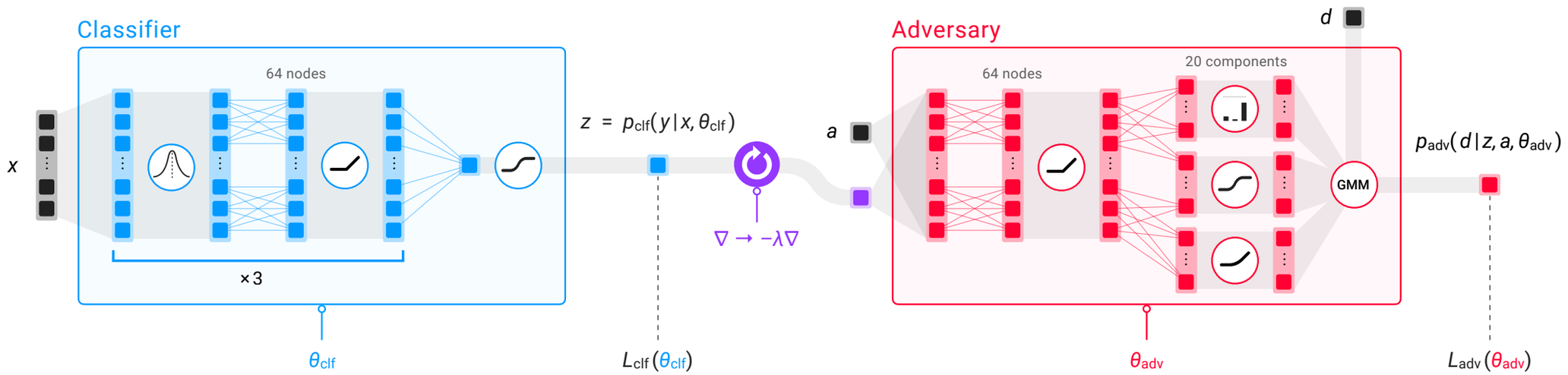
*— These are **not** real people —*



Mass-decorrelation

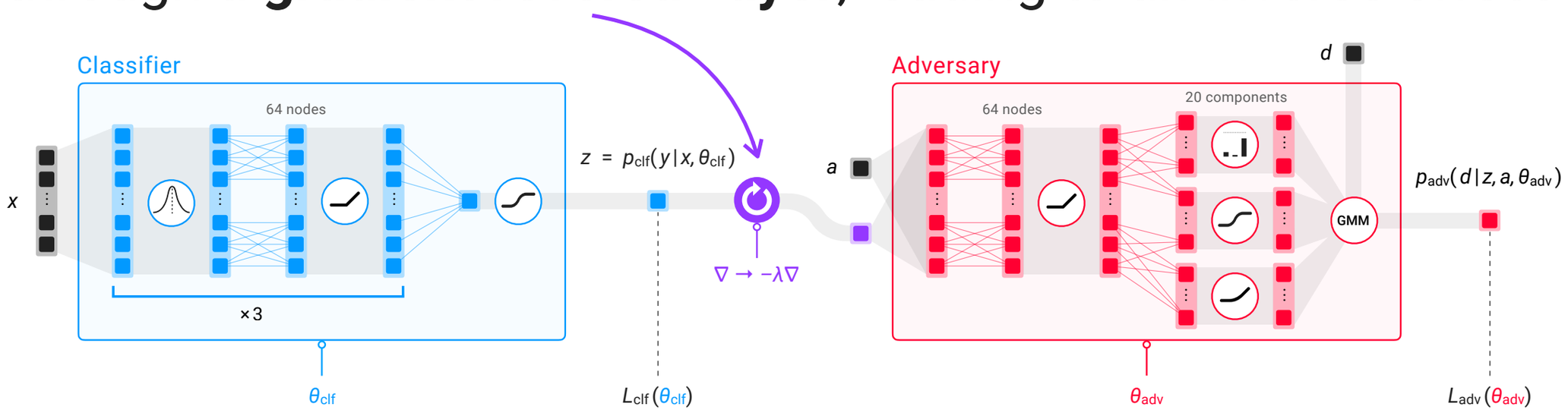
- Idea: Use adversarial architecture to remove correlation of NN jet classifier with jet mass

  1. Pit classifier network against **adversary**
  2. Classifier tries to guess jet label *y* (0 or 1) from inputs *x*
  3. Adversary tries to guess jet mass from classifier output
  4. If possible, the two are correlated and the clf. is penalised



Mass-decorrelation

- Adversary NN parametrises p.d.f. in the jet mass m conditional on classifier output z, i.e. $p_{\text{adv}}(m \mid z)$
- Trained with loss $L_{\text{adv}} = -\log p_{\text{adv}}(m \mid z)$
- If $p_{\text{adv}}(m \mid z)$ is able to do **better than prior,** i.e. $\mathrm{d}N_{\text{jets}}/\mathrm{d}m|_z$, the classifier output carries information about the jet mass
- Gradient minimising L_{adv} is propagated back to the classifier through a **gradient reversal layer**, leading to an inverse effect

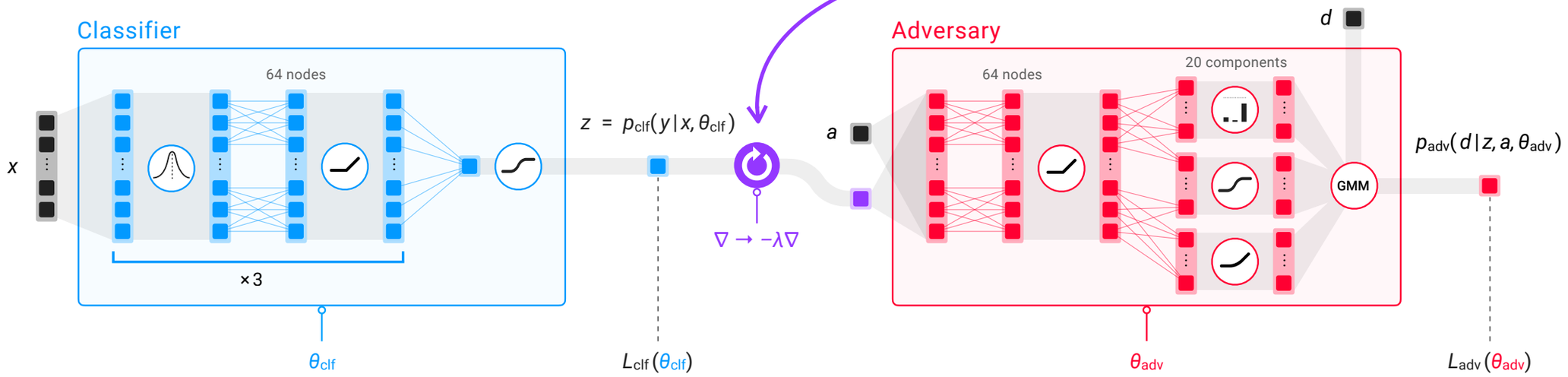


Mass-decorrelation

- Both networks trained simultaneously with loss

$$\min_{\theta_{\text{clf}}} \max_{\theta_{\text{adv}}} L_{\text{clf}}(\theta_{\text{clf}}) - \lambda L_{\text{adv}}(\theta_{\text{clf}}, \theta_{\text{adv}})$$

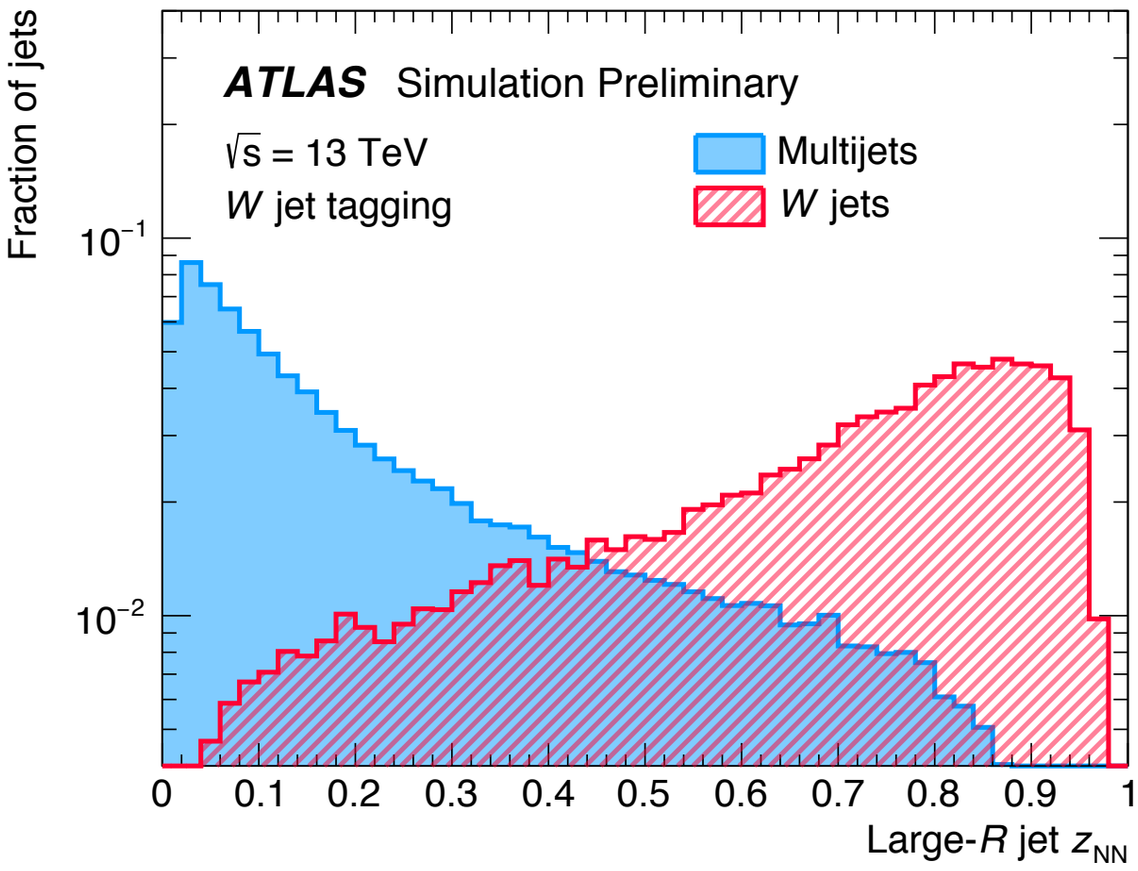
- Adversary tries to improve mass-prediction
- Classifier tries to improve classification *and* make the adversary's job harder
- Trade-off controlled by parameter λ



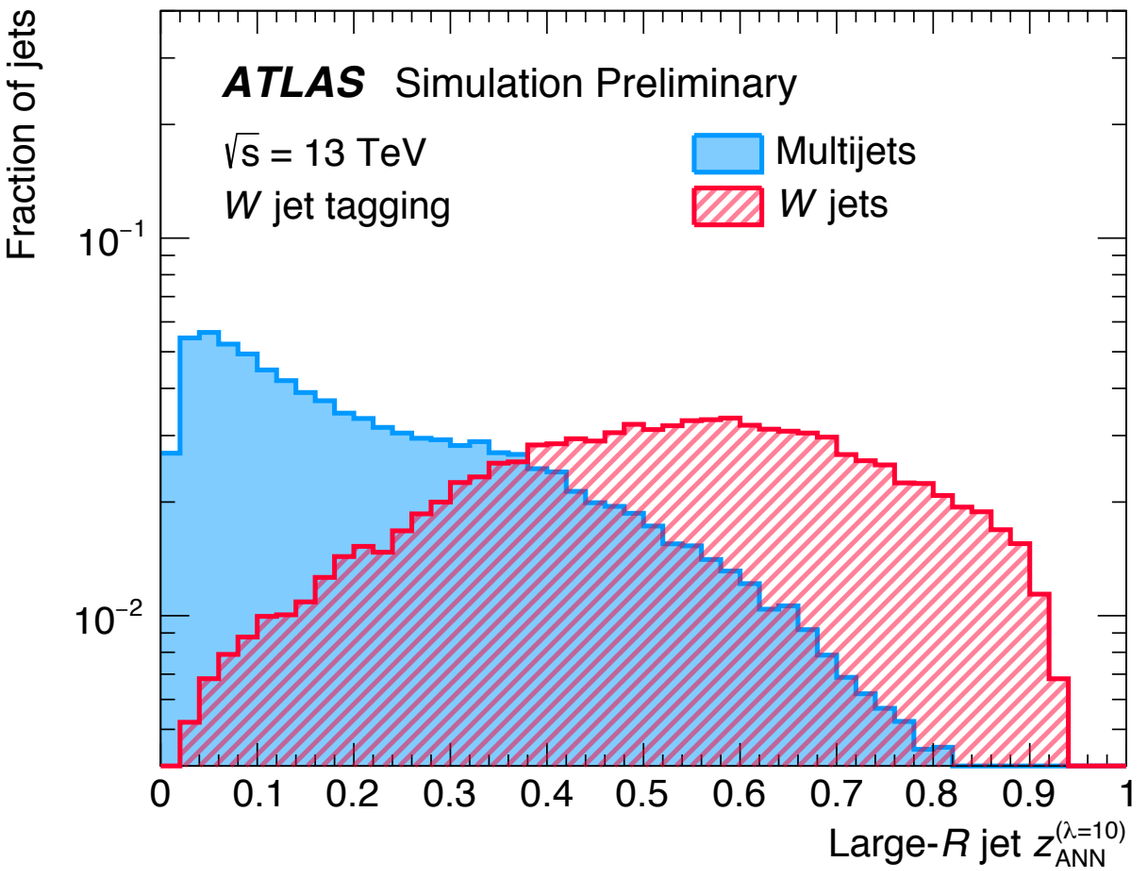
Results

- (A)NN classifier distributions

Stand-alone NN



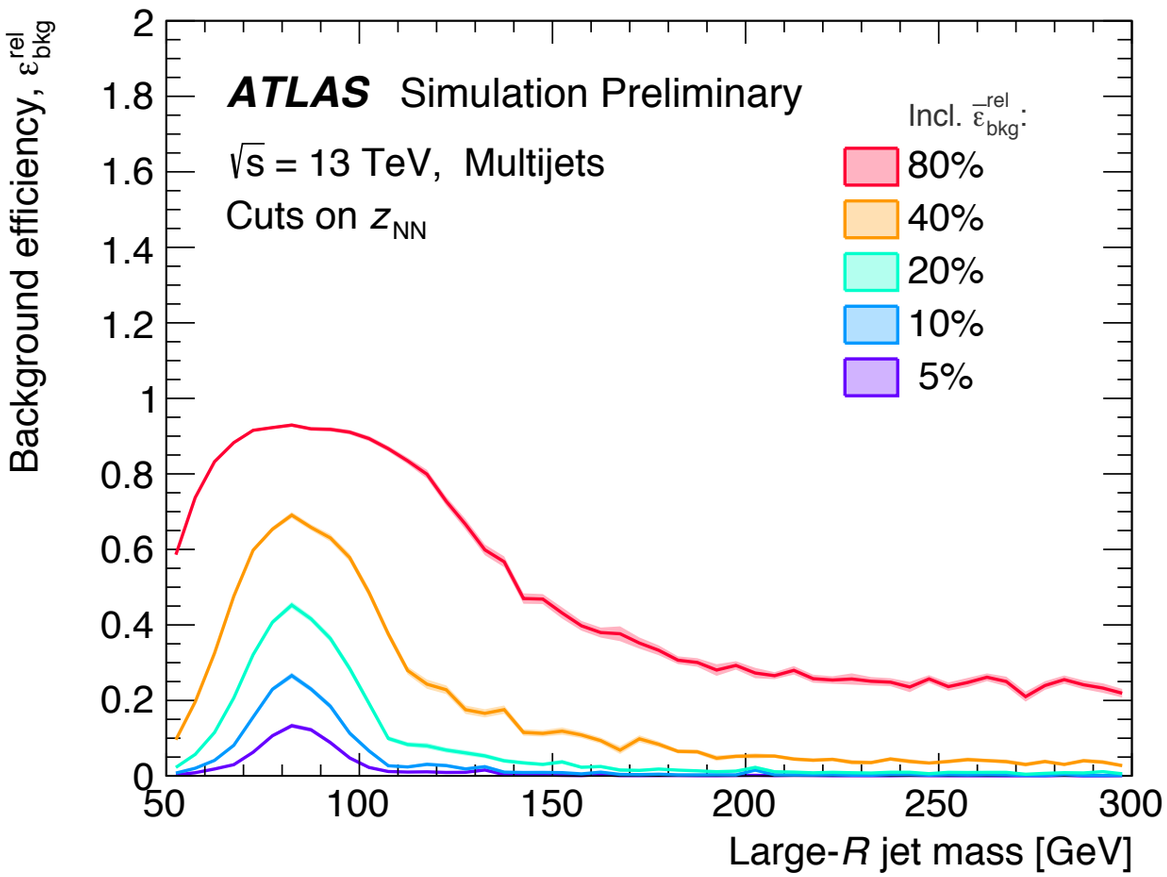
Adversarial NN



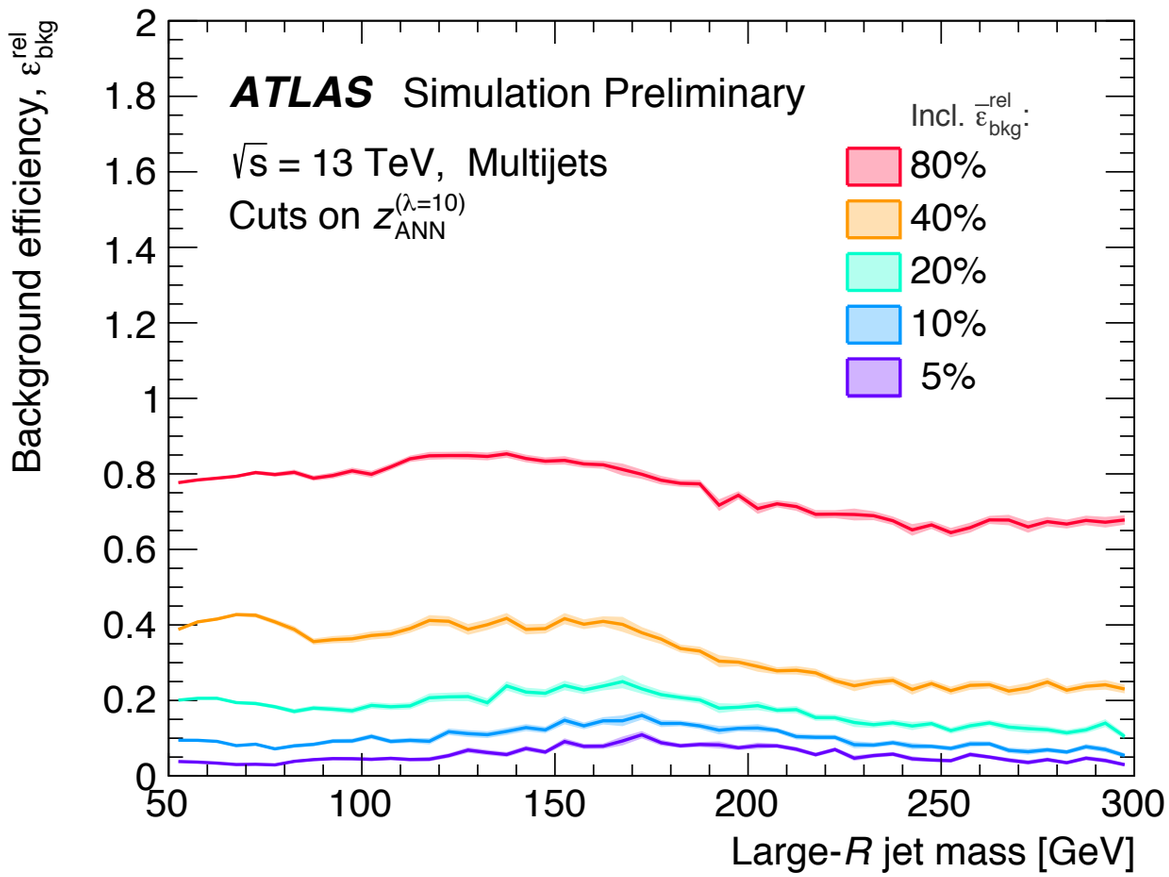
Results

- (A)NN classifier background selection efficiencies

Stand-alone NN



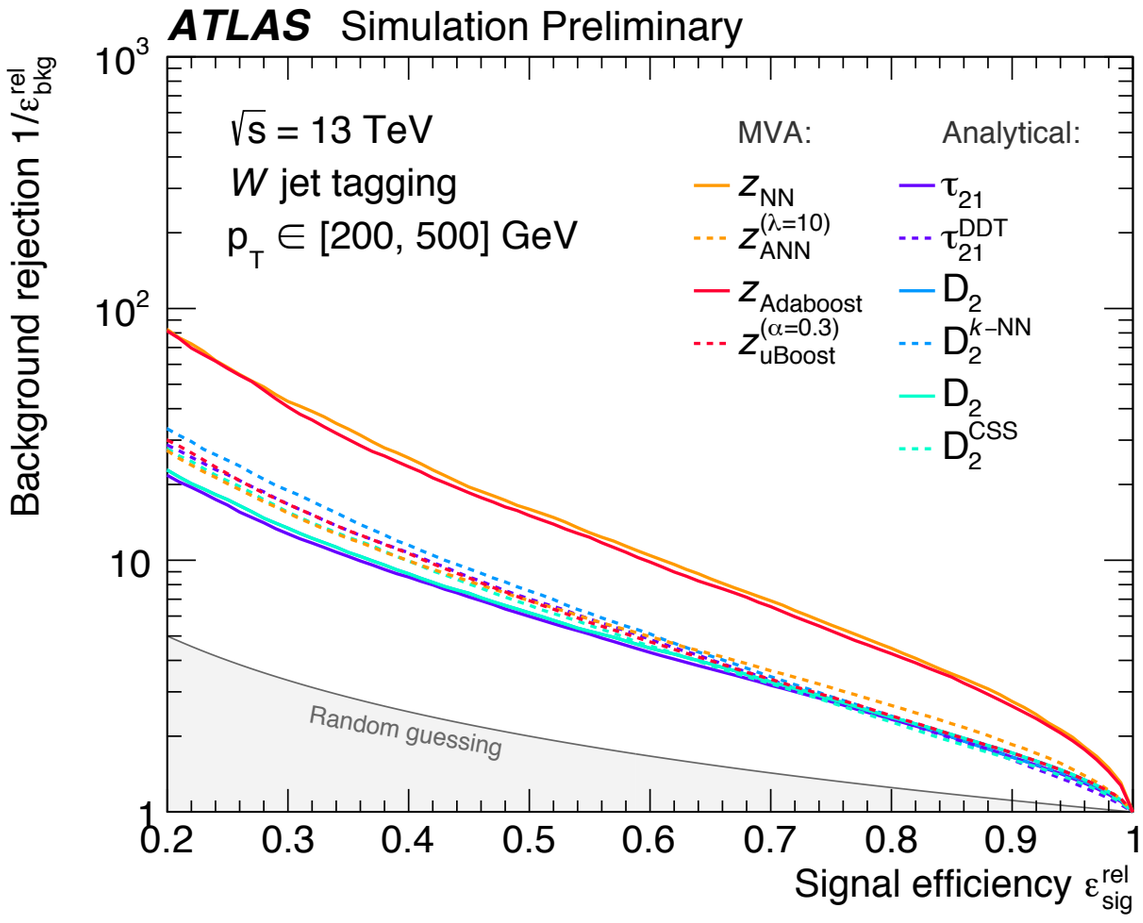
Adversarial NN



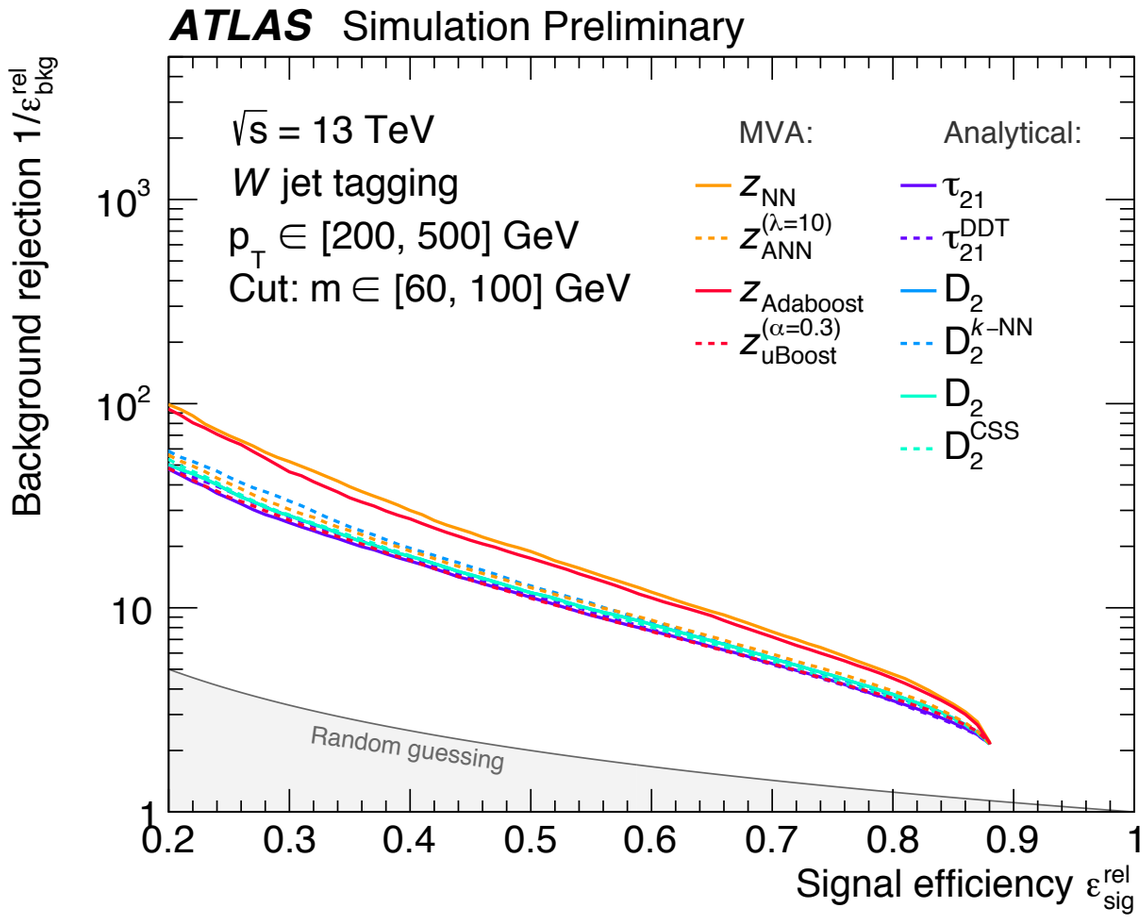
Results

- Receiver-operator characteristic (ROC) curves

Without jet mass-selection



With jet mass-selection



Results

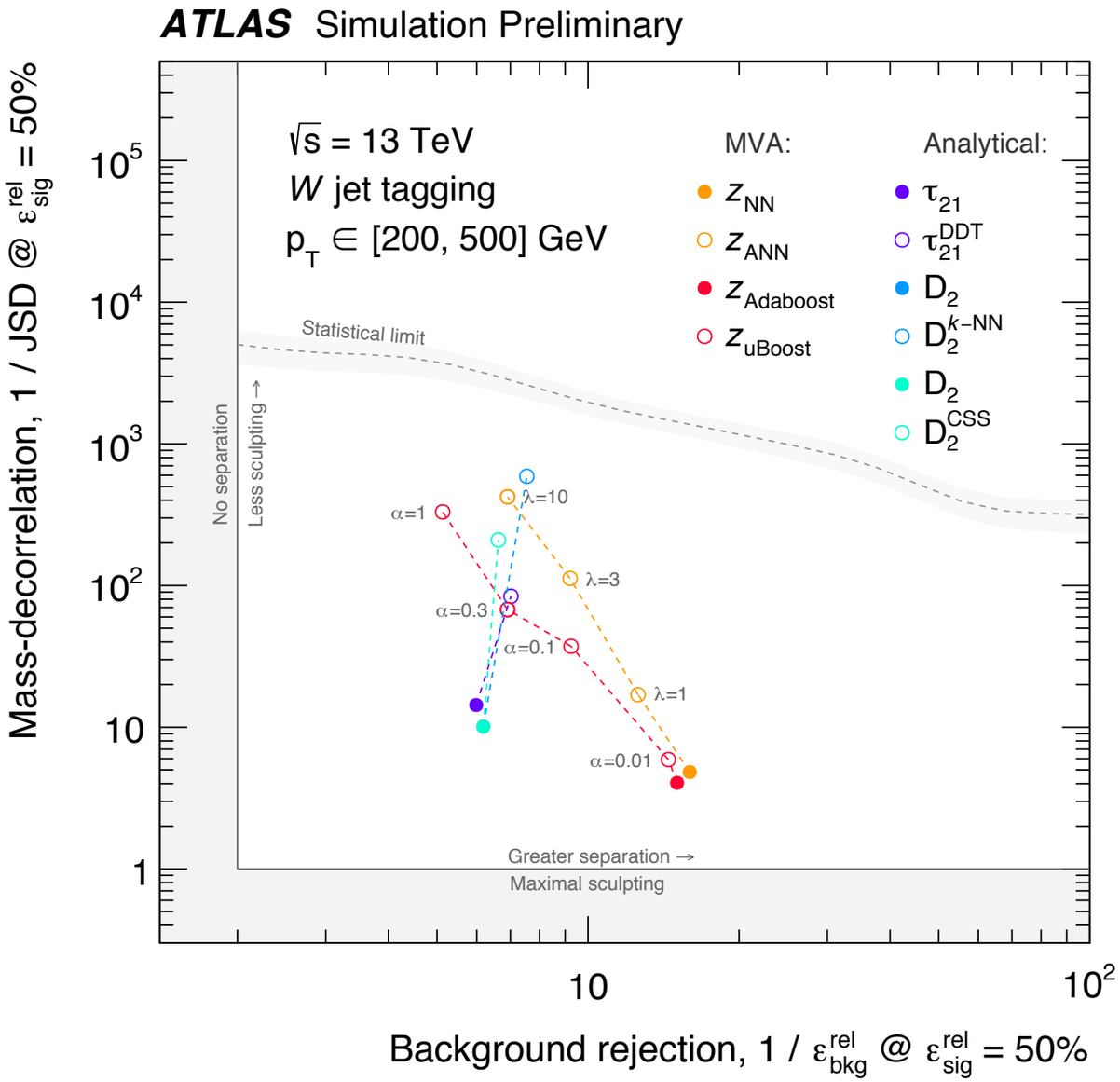
- Combined (classification, mass-decorrelation) performance
- Mass-decorrelation measured using inverse Jensen-Shannon divergence:

$$\mathrm{JSD}(P \parallel Q) = \frac{\mathrm{KL}(P \parallel M) + \mathrm{KL}(Q \parallel M)}{2}$$

where $M = (P + Q)/2$ and Kullback-Leibler div.:

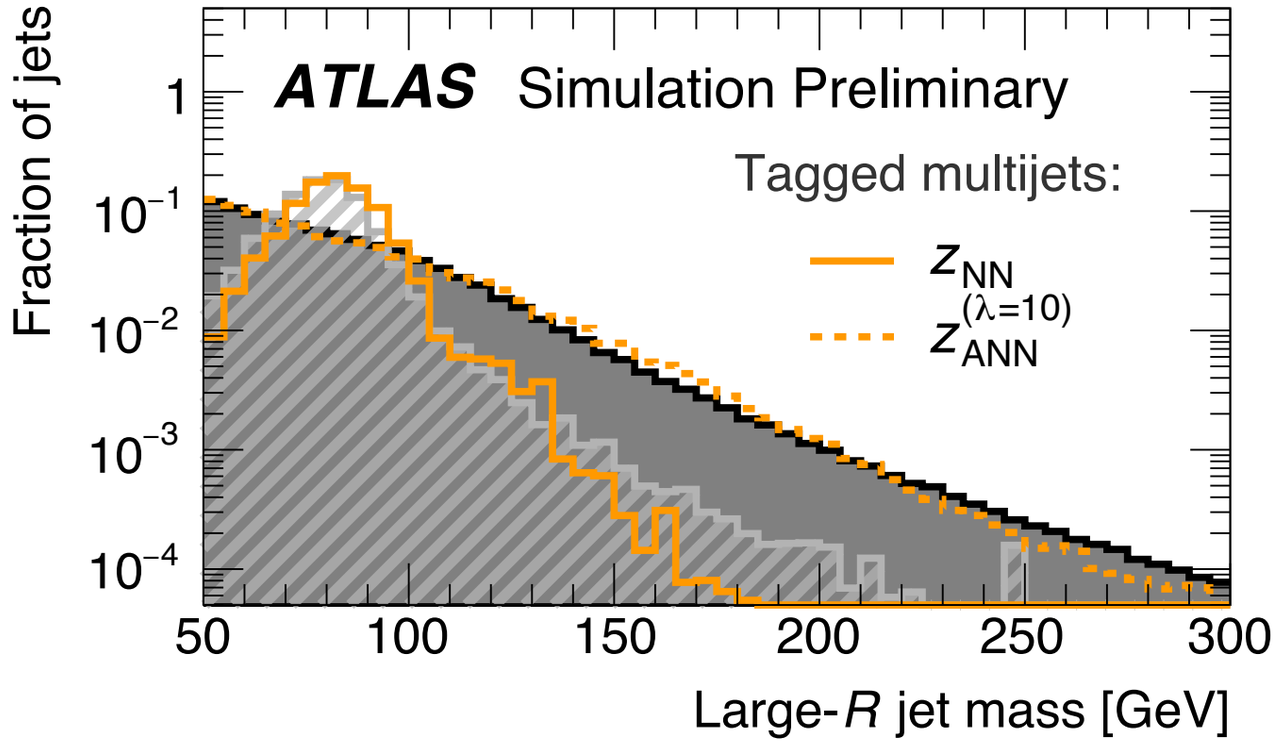
$$\mathrm{KL}(P \parallel Q) = \sum_i P_i \log_2 \frac{Q_i}{P_i}$$

for classification pass- and fail jet mass distributions P_i and Q_i



Mass-decorrelation

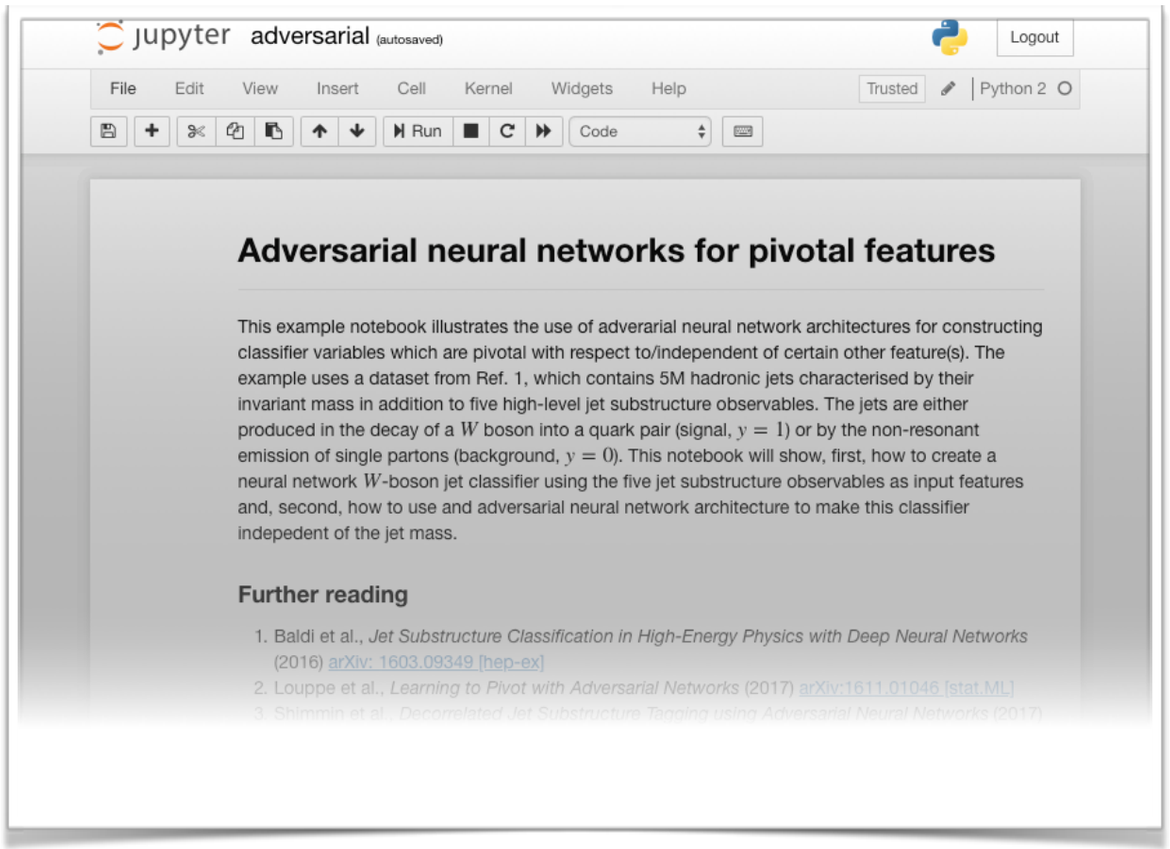
- **Result:**
  - Standard clf. (——) sculpts background, mass-decorrelated one (- - - -) doesn't!



- Balance between objectives (classification and mass-decorrelation) to be determined on a case-by-case basis

Companion notebook

- Prepared notebook **`adversarial.ipynb`** provides a examples of how to train stand-alone classifiers and how to use adversarial architecture to impose mass-decorrelation



- Location: **`http://github.com/asogaard/ep2mlf`**

Thank you