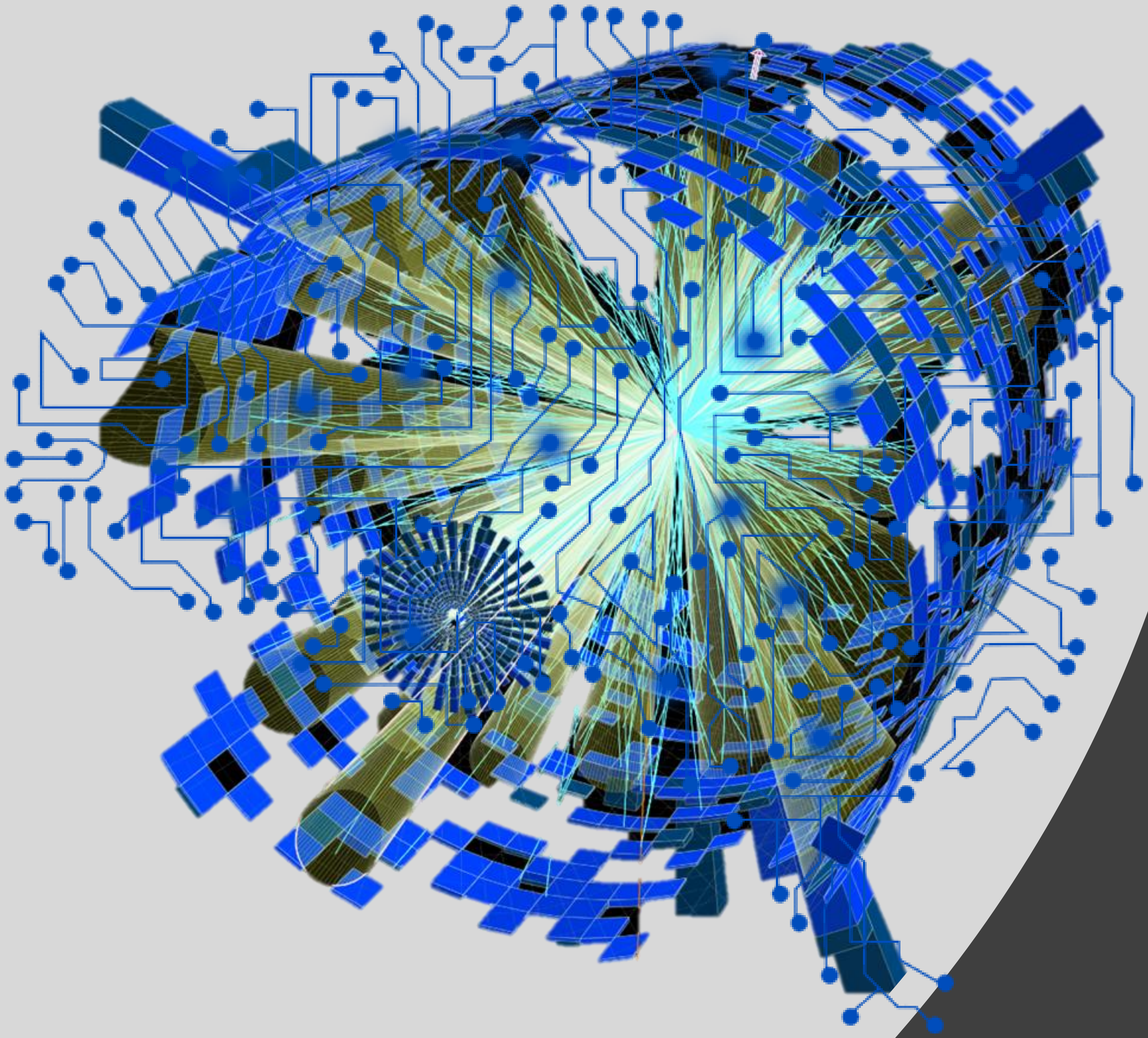


Simulating Reality & Searching for the Unknown

Tobias Golling,

University of Geneva

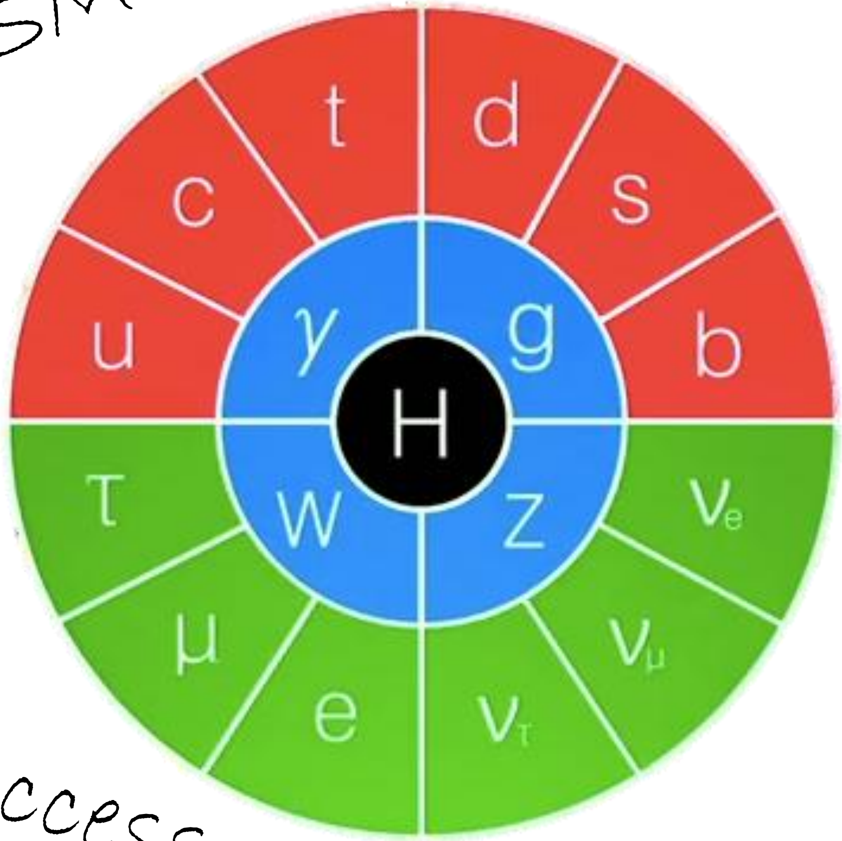
Outline

- Establish the goal: maximize LHC's sensitivity to new physics
- The need for accurate and fast background modeling
- Extend LHC's physics portfolio to model-agnostic searches
- How machine learning can help to overcome the challenges
  - Automate
  - Reduce complexity

[Disclaimer: highly biased & personal coverage]

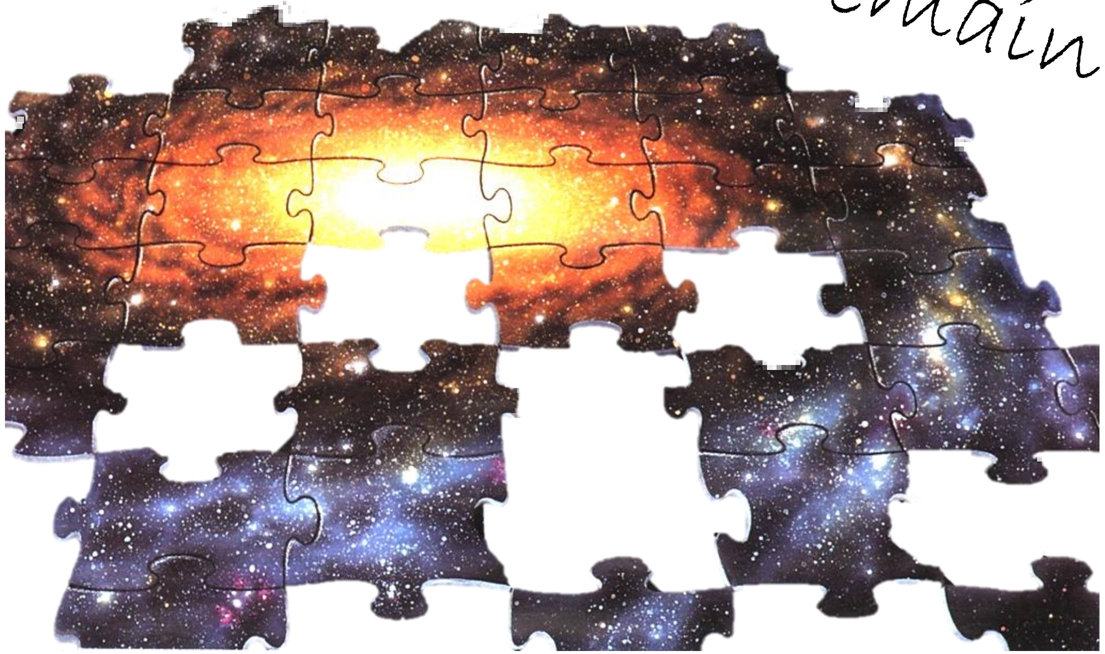
The current situation

The SM is complete



Success story!

Open mysteries remain

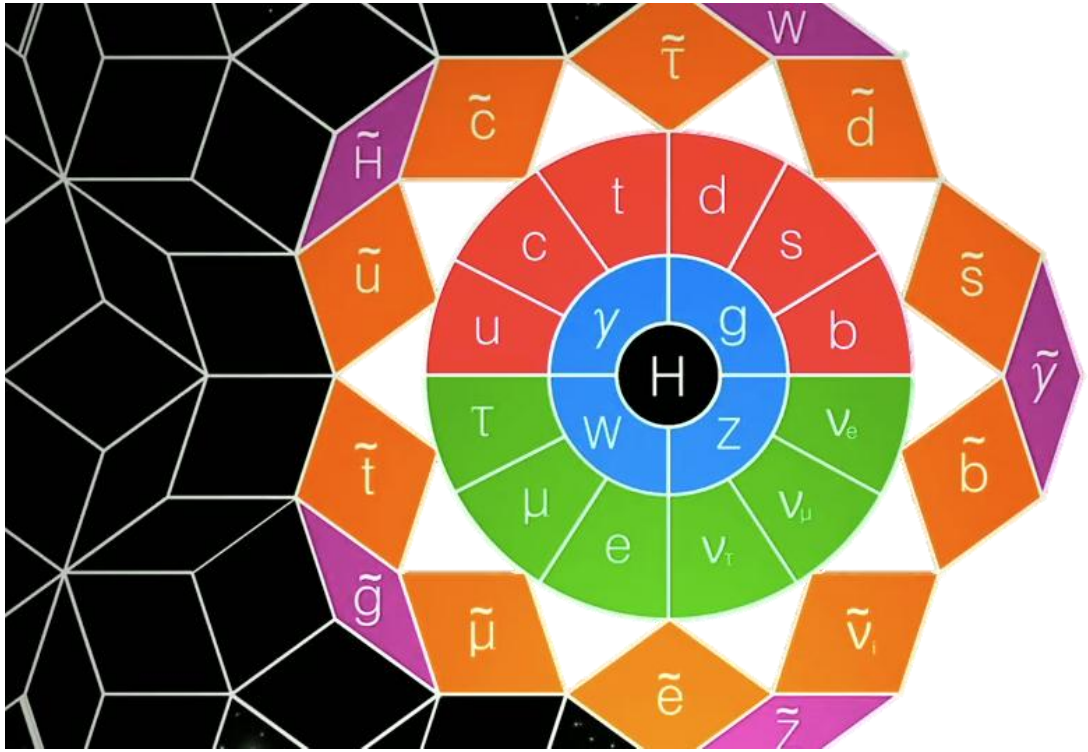


Dark matter, dark energy, quantum gravity,...

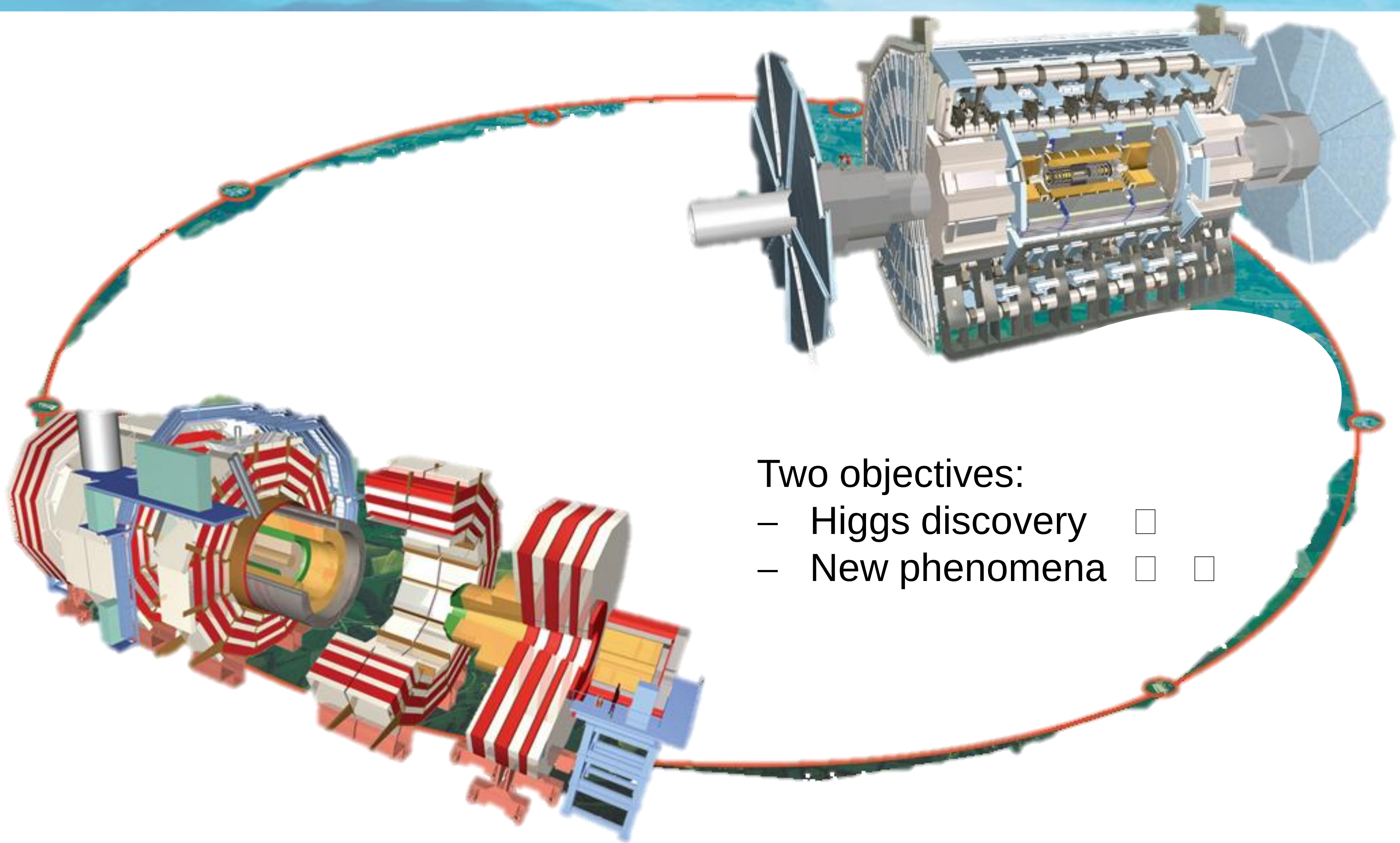
The theory guidance

- Hypothesize extensions of the SM
  – Addressing SM shortcomings
  – Leading to *testable* predictions

- Plethora of Beyond-the-SM extensions…



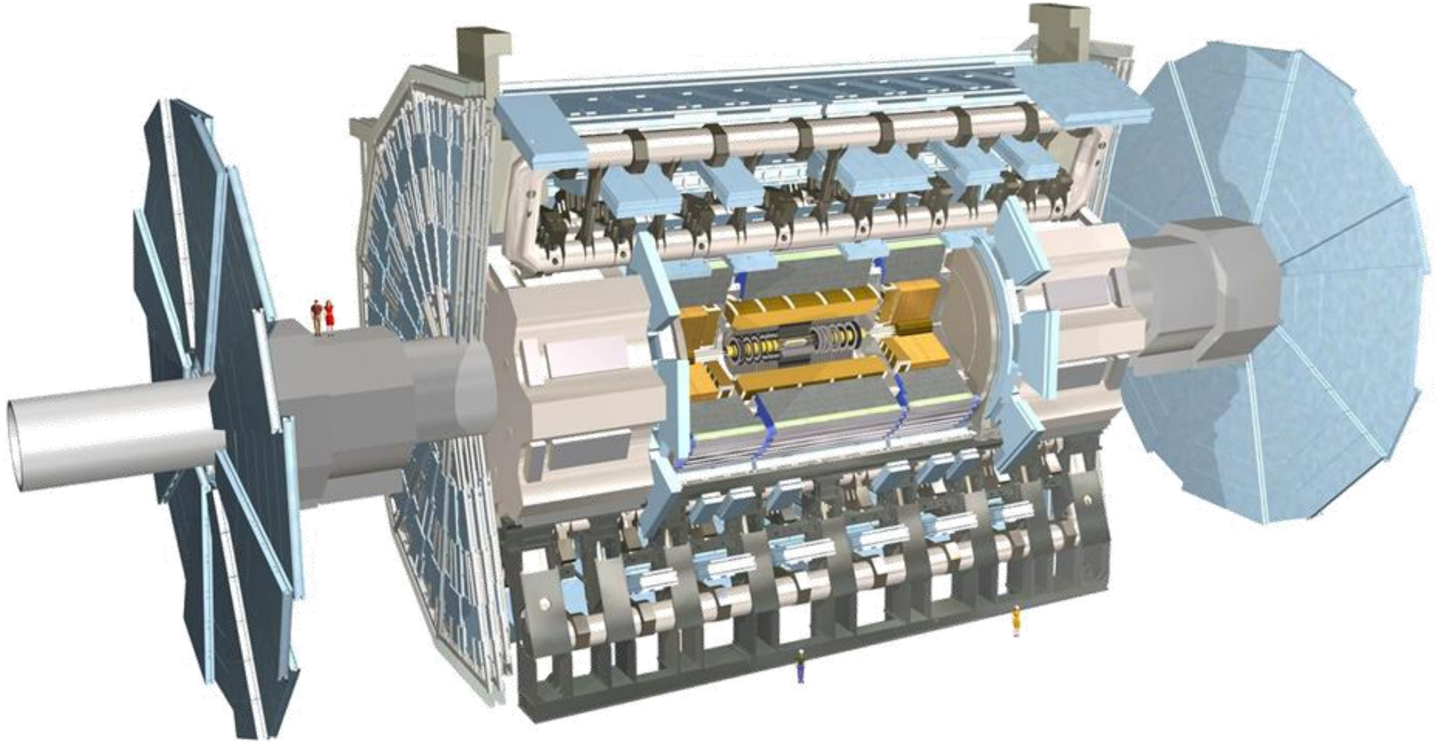
The Large Hadron Collider (LHC)



Two objectives:

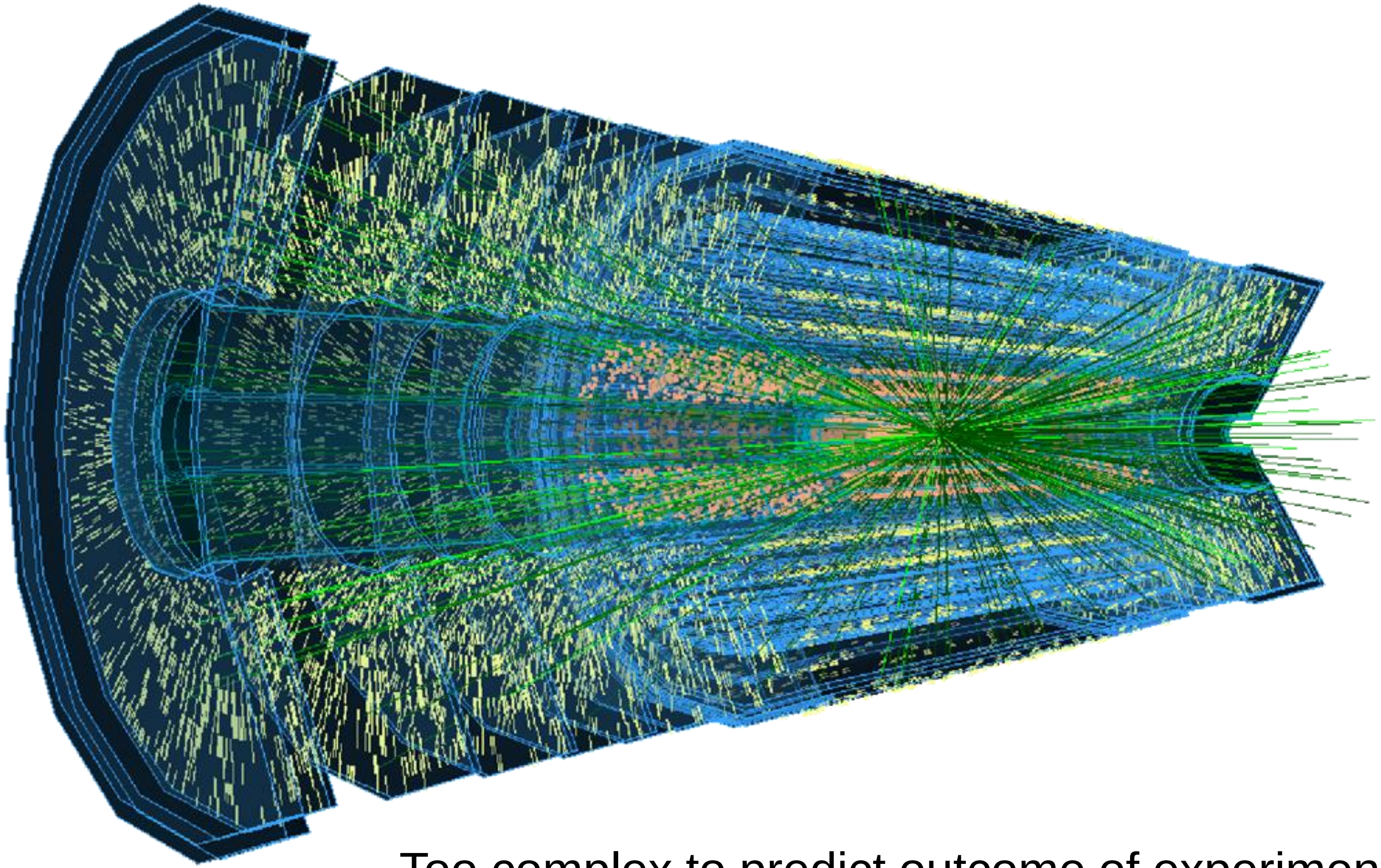
- Higgs discovery □
- New phenomena □ □

The ATLAS detector



- 40 MHz collision rate – online filter to record ~1kHz
- Thousands of particles per collision
- 100M readout channels, ~1% occupancy
- Trillions of collisions in data & simulation – hundreds of petabytes

The need for synthetic data

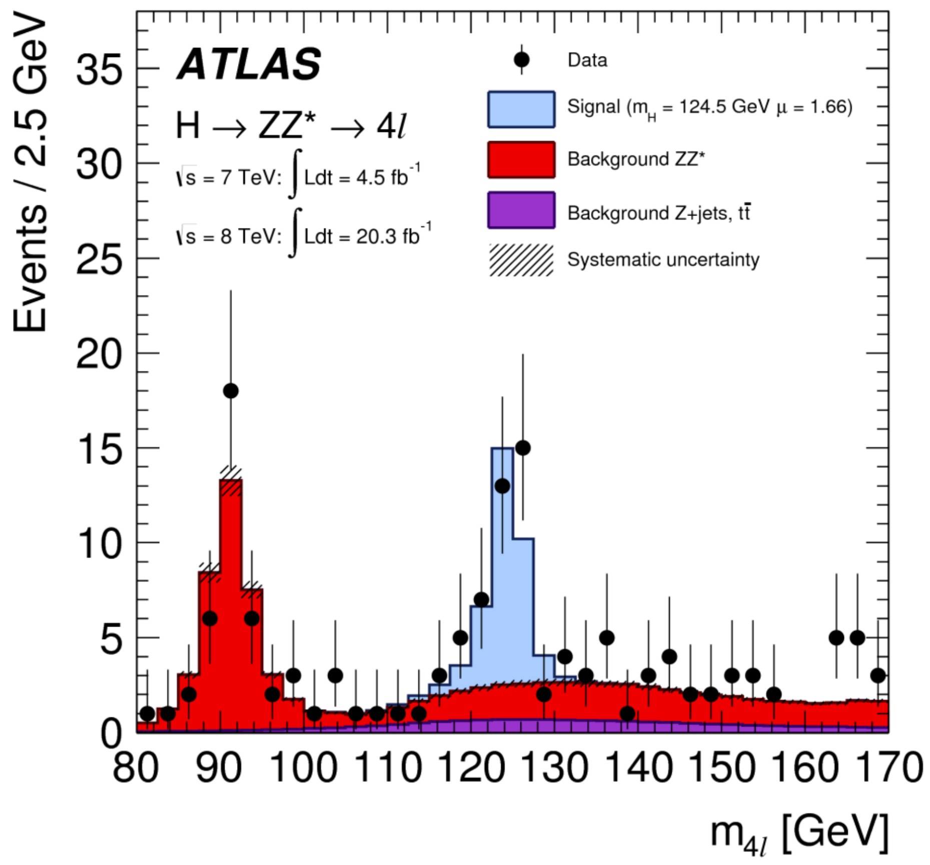


Too complex to predict outcome of experiment from first principles
→ **Monte Carlo simulation**

The method of hypothesis testing

- Example: Higgs boson discovery:
  - – H_0: no Higgs
  - – H_1: null+Higgs

- Our standard inference approach:
  - Reduce input data $O(10^6)$ to $O(1)$ human-engineered feature
  - *Far from ideal*



[Phys. Rev. D. 90 (2014) 052004]

Toolbox: what is ML good for?

Search for something *rare* in a *deluge of data:*

1. We *know* the signal (i.e. label) – **supervised ML**
2. We *do not know* the signal (no labels) – **unsupervised ML / anomaly detection**
    i. Partial/noisy labels - **weakly-/semi-supervised ML**
3. High-fidelity and *high-speed* modeling – **generative ML**

- Use *Deep Neural Networks* to make the best out of the data we have

Deep Neural Network

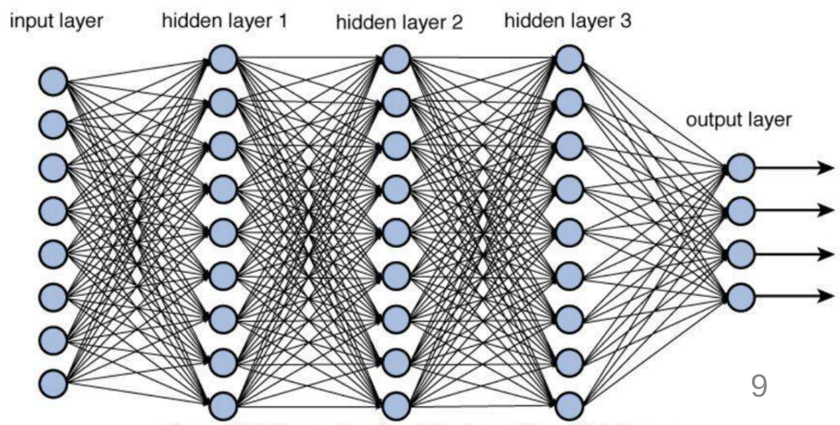
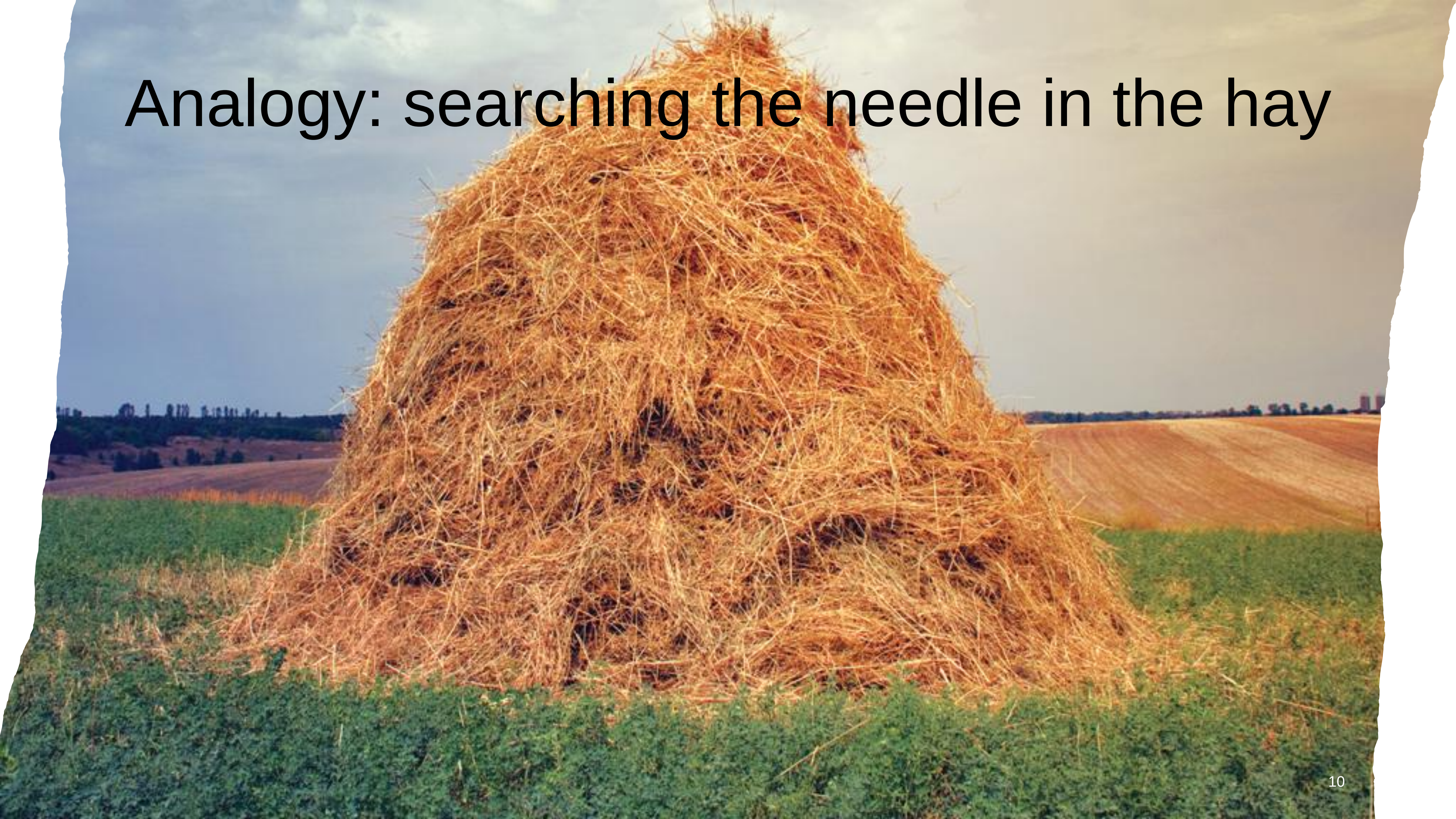


Figure 12.2 Deep network architecture with multiple layers.

Analogy: searching the needle in the hay

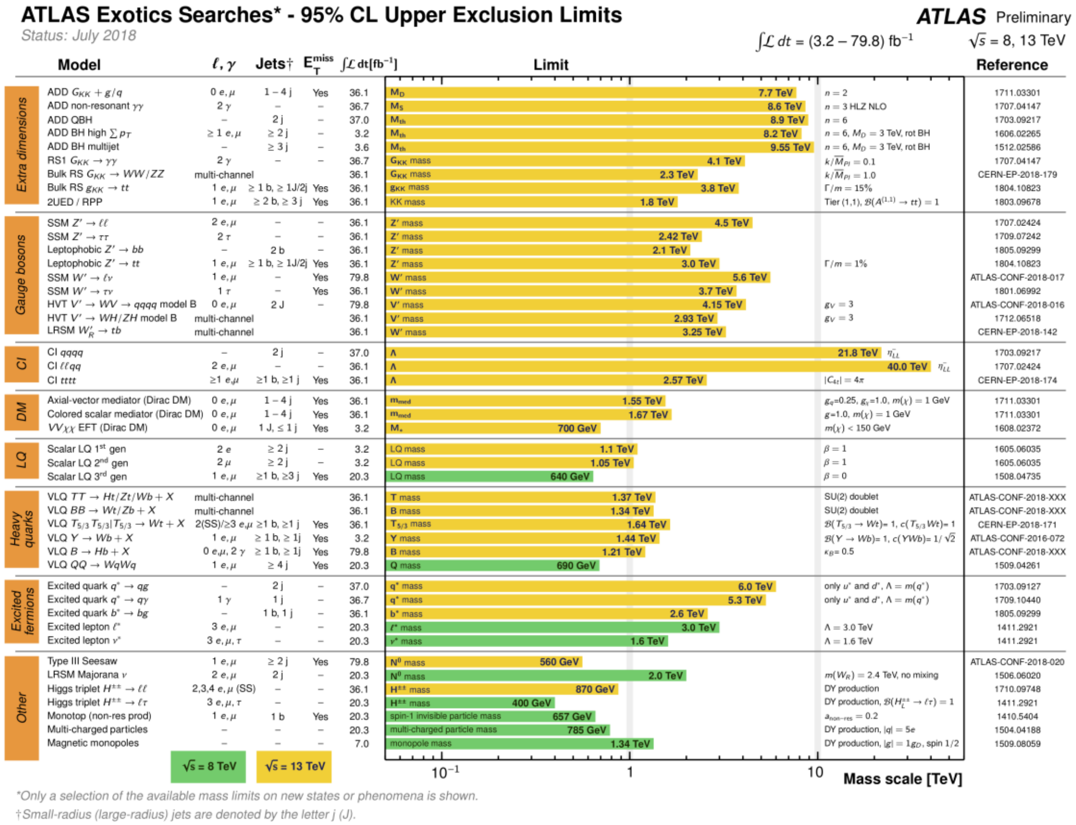


1. Searching for the **known**

- Take theory guidance at face value
  - We **know** how a needle & hay *look like*
- **Supervised** approach to fully exploit this knowledge

The *blemish*:

No sign of physics Beyond the SM



2. Searching for the **un**known

- **Discard** theory guidance
  - Don’t know **what** we’re looking for in the hay
- **Un**supervised approach to search for **structure** in the data

- Anomaly detection
  - **Outlier** *easy:* Not a needle but maybe a shiny object…
  - **Inlier/over-density** *much harder but closer to reality*: a tiny bit of *special* hay in a humongous haystack

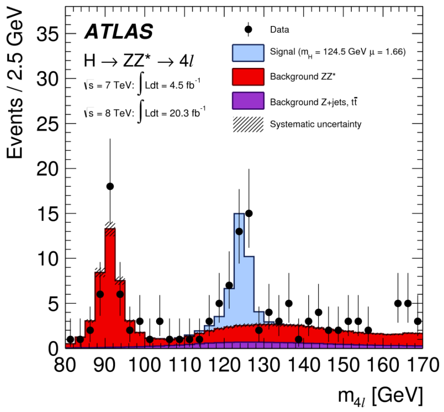
Assumptions

- **Anomalies are rare** – otherwise we would have seen them already
  - No issues of *overlapping anomalies*

- **Anomalies are localized** – most prominent are resonances
  - Can define signal region (SR) with enhanced anomalous events
  - Control region (CR) depleted in anomalies

- **The data is smooth** – BG features vary slowly between SR & CR
  - Can use CR data to estimate BG in SR

- Only interested in statistical statement of group anomaly
  - Not trying to identify individual outliers



Analogy: searching for anomalies in the desert

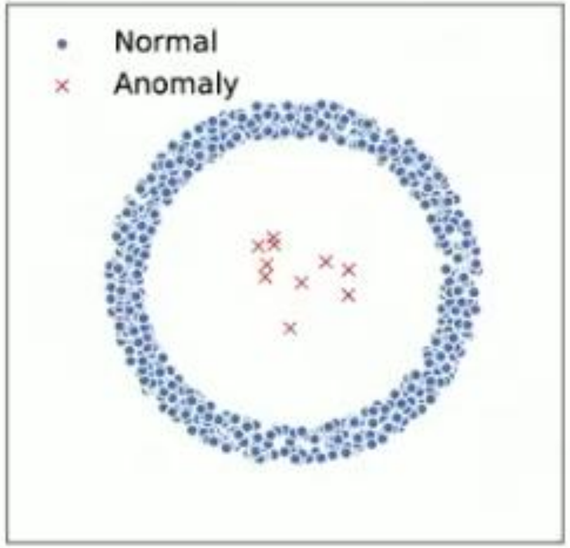
- Grain of sand ≙ LHC data collision
- What is an **outlier**
- What is an **inlier / over-density**

Example of an outlier

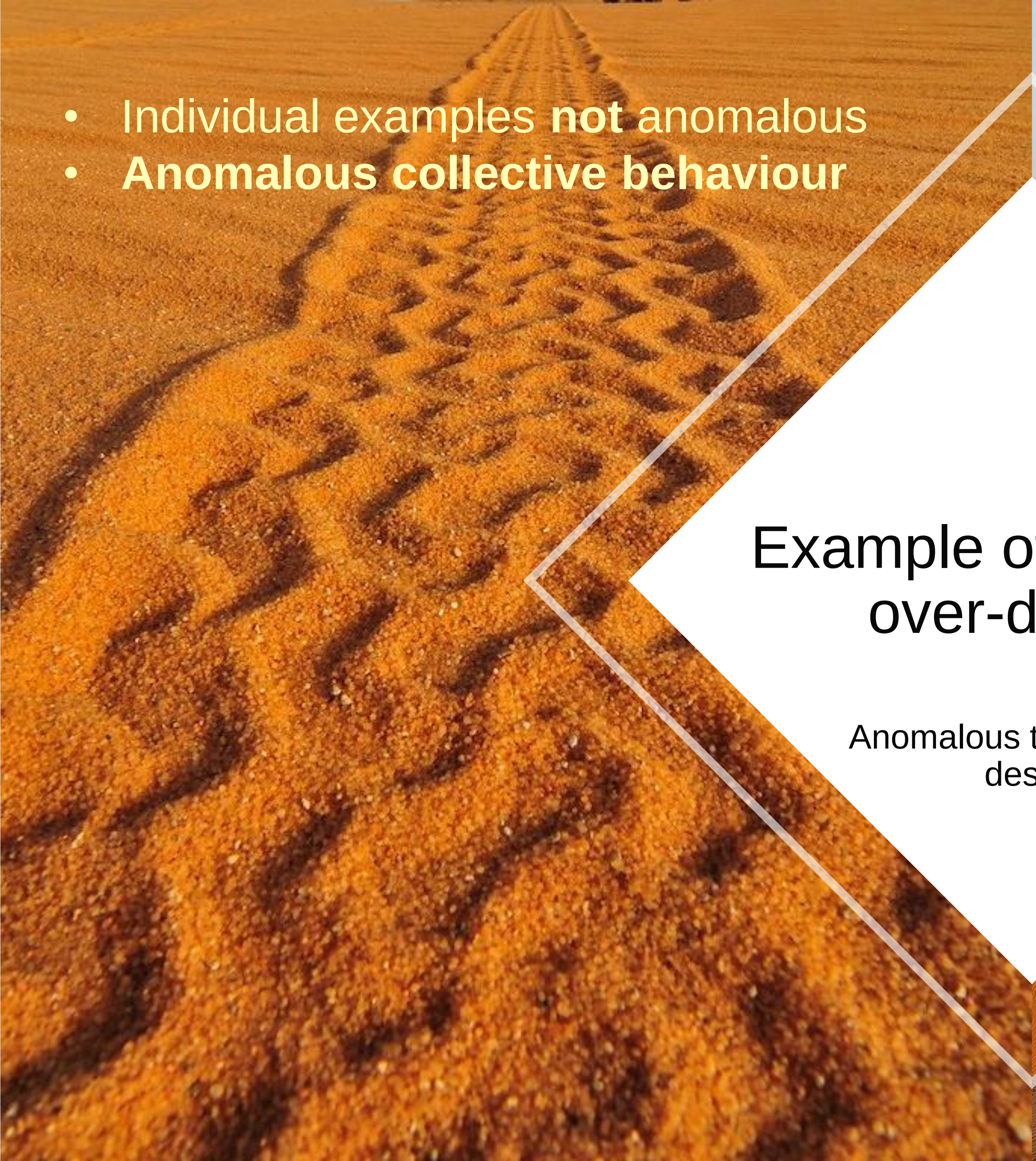
- *Anomalous monolith* in the desert

- Imagine each data point is a
  - *photo* of a grain of sand
  - equivalent *grain of monolith*

- *Grain of sand* easily separable from *grain of monolith*



[https://www.vox.com/culture/22062796/monoliths-utah-california-romania]



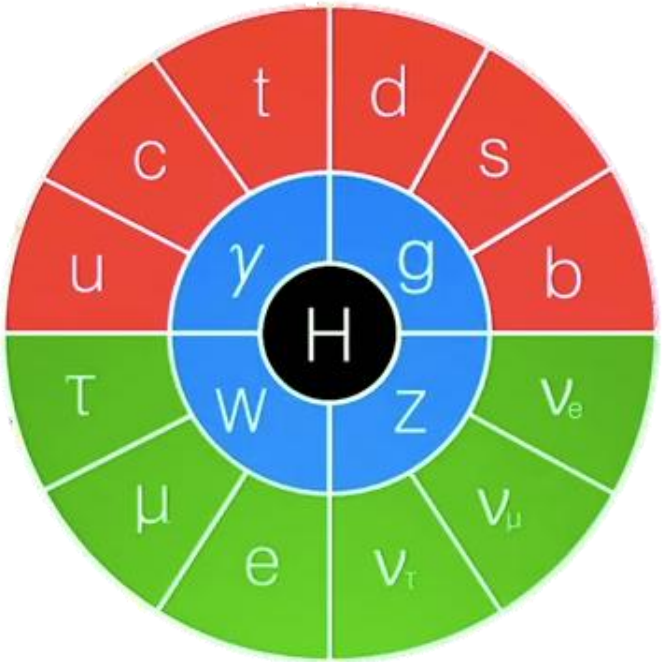
- Individual examples **not** anomalous
- **Anomalous collective behaviour**

Example of an inlier / over-density

Anomalous tracks in the desert

Need to know your **normal** events before you can look for **anomalous** events

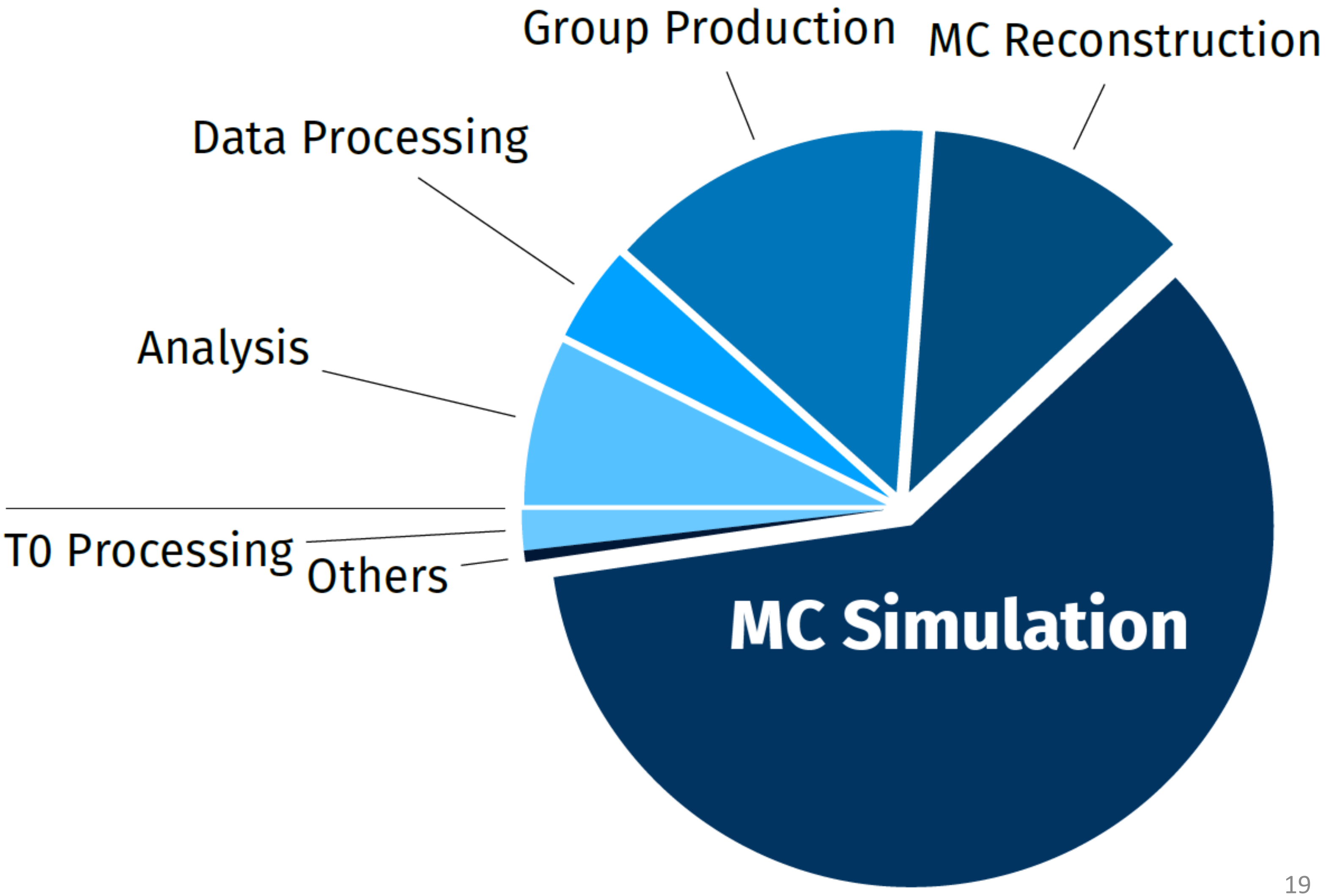
- Model of the desert



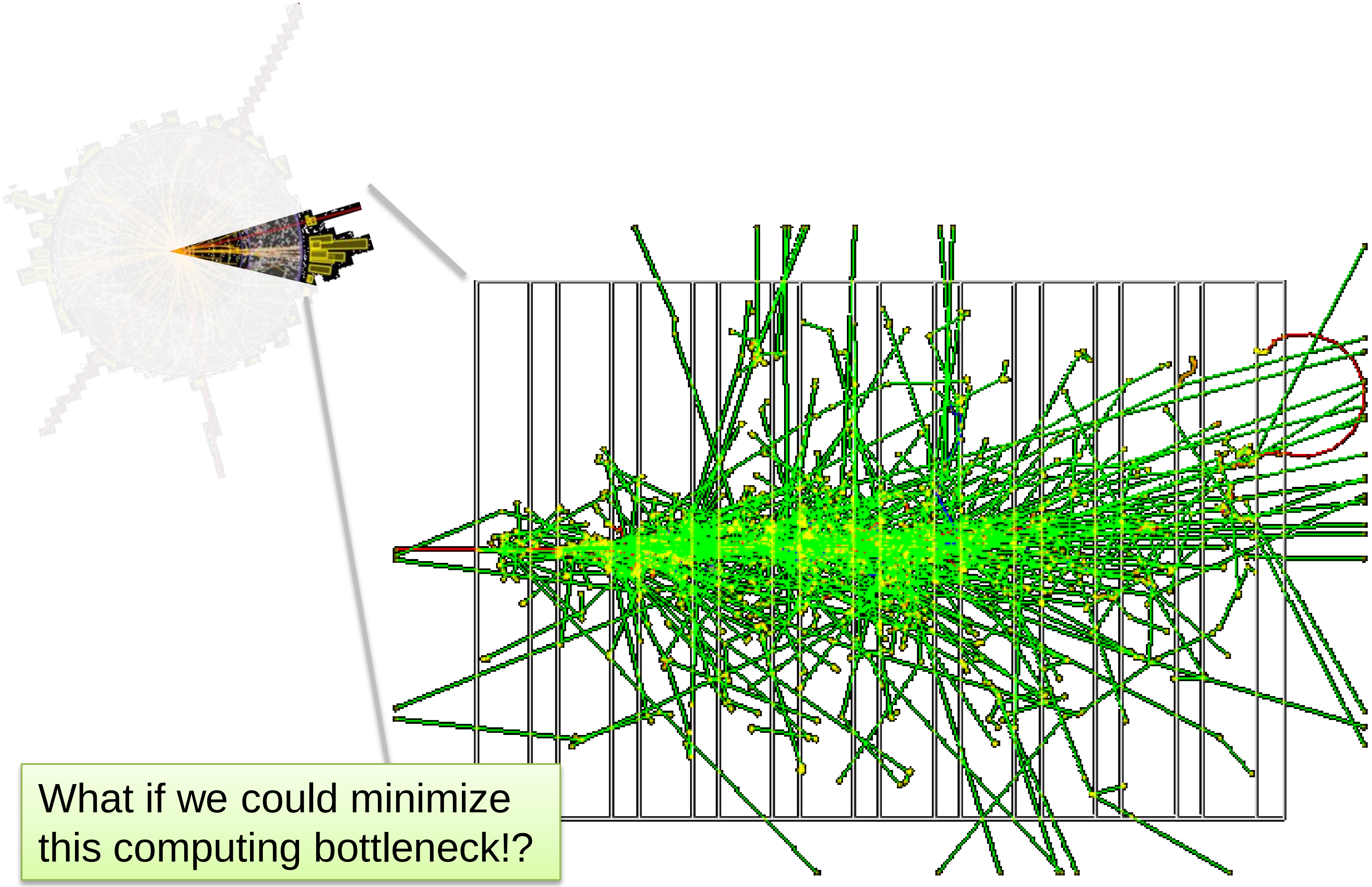
- Model of our SM events

Forward Monte Carlo modeling

Computing bottleneck:
Monte Carlo simulation



One particle entering the calorimeter…

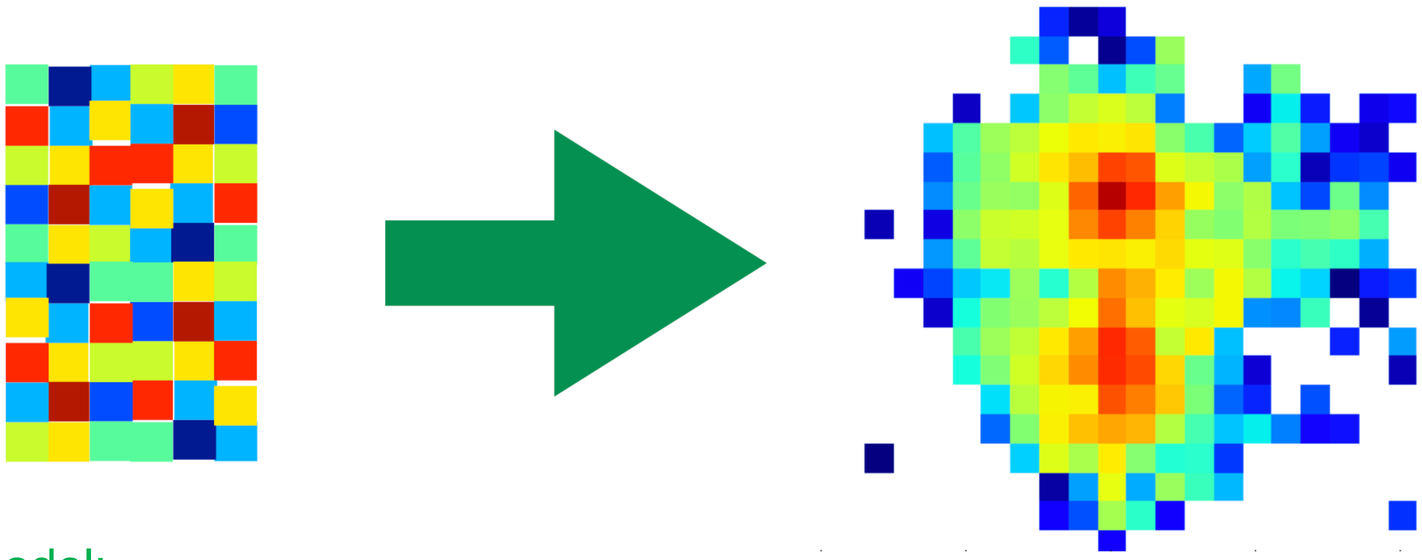


- **Geant4:** simulate at **microscopic level** interaction of particles with matter
- Bottleneck: calorimeter simulation - **O(10 min) per 1 event**
- ⇒ Need **trillions** of simulated events

What if we could minimize this computing bottleneck!?

Toolbox: generative modeling

Build a generator* which maps
random numbers to structure



$$p_{\text{model}} \approx p_{\text{data}}$$

*Deep generative NN model:

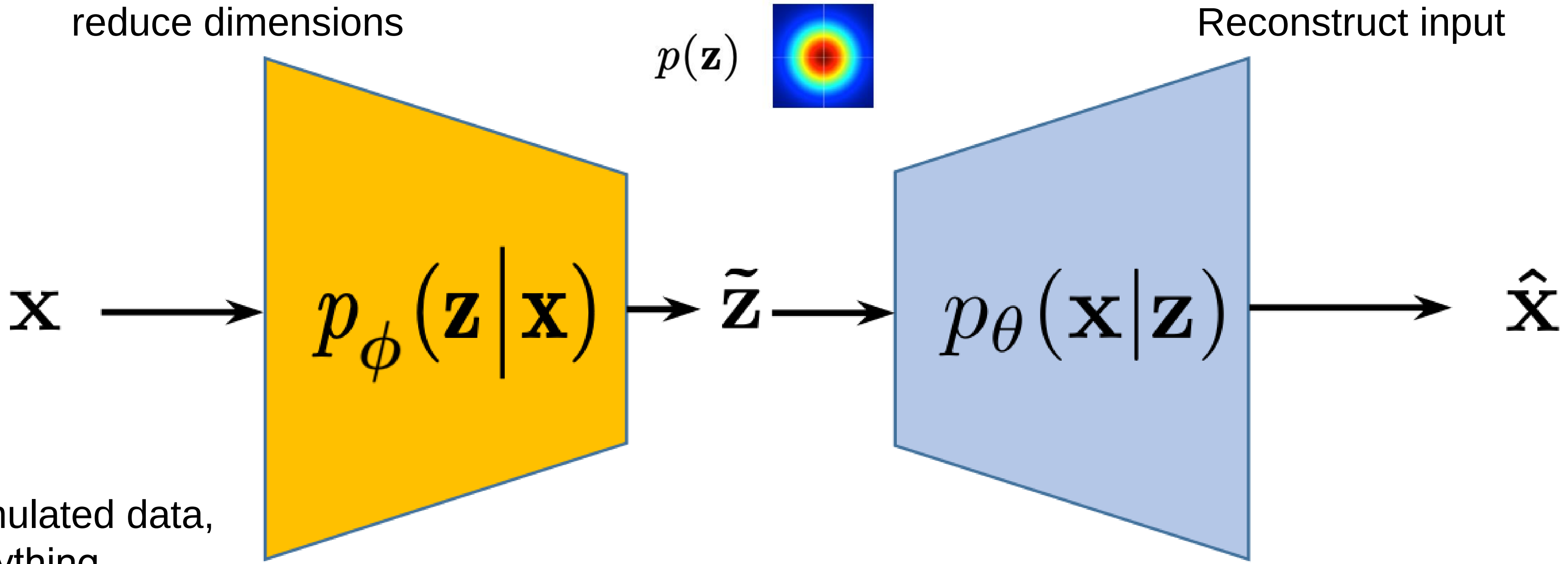
- Generative Adversarial Network (GANs)
- Normalizing Flows (NFs)
- **Variational Autoencoders (VAEs)**

Toolbox: Variational Autoencoder (VAE)

<u>Latent space (with given prior)</u>:
easy to sample from

<u>Probabilistic **encoder**</u>:
reduce dimensions

<u>Probabilistic **decoder**</u>:
Reconstruct input

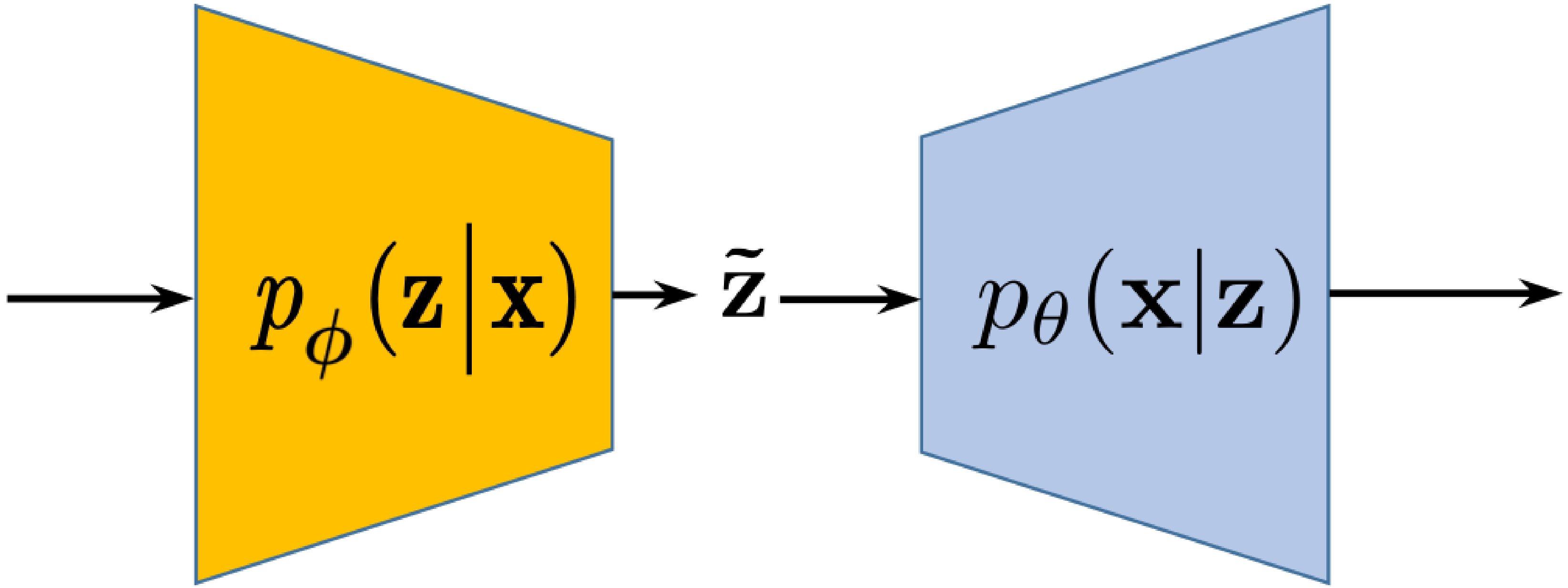


<u>Input x</u>:
Raw data, simulated data,
features,…anything

1. <u>Reconstruction mode</u>
2. <u>Generation mode</u>

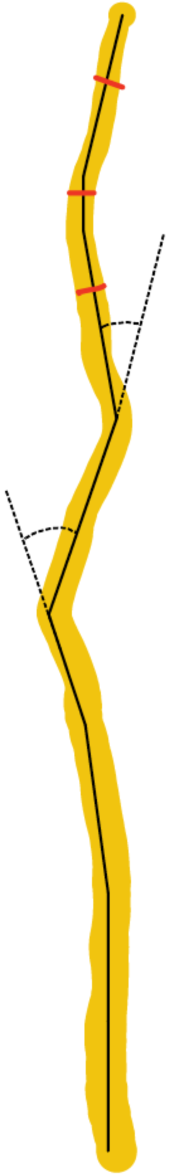
<u>Information bottleneck</u>:
maximize encoded information

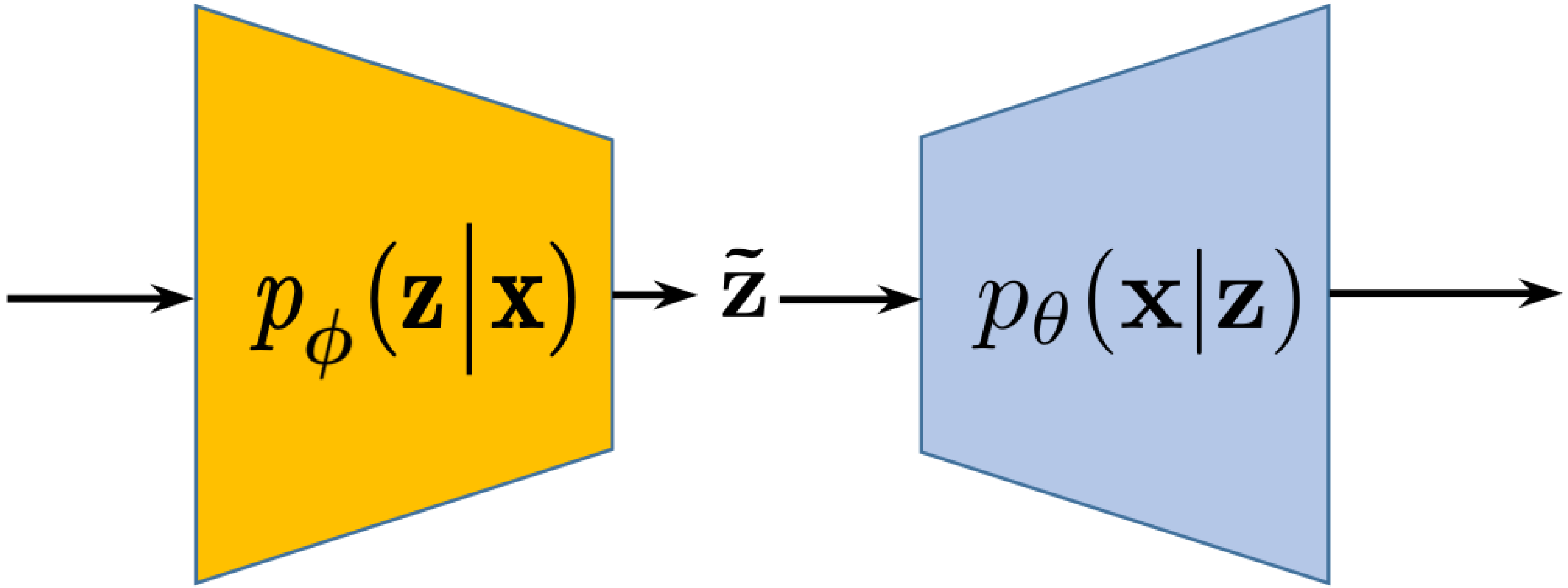
[Data volume reduction]

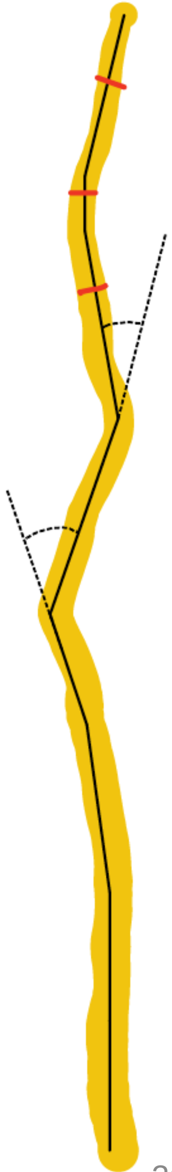


- Lossy compression with auto encoders
- Only maintain key features in data
- Example application in PP: trigger
  - reduce bandwidth to increase event rate

Reconstruction mode



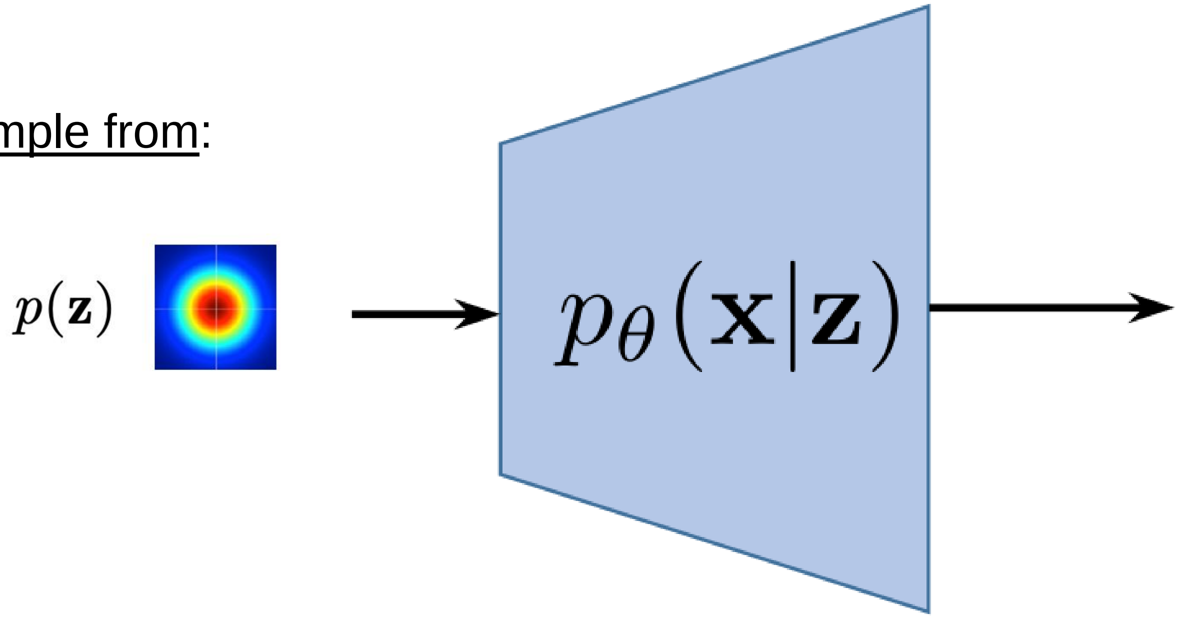


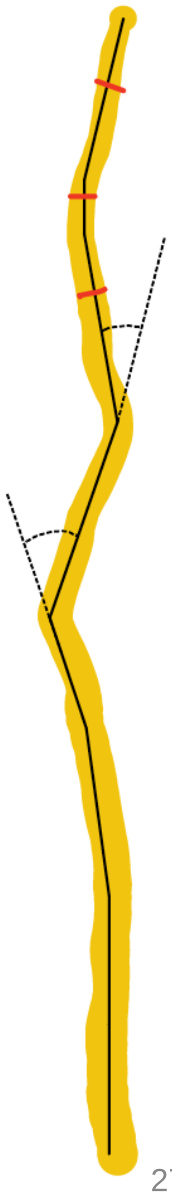


- Train on hay
- Apply to data: poor reconstruction for non-hay = anomaly

Generation mode

Sample from:





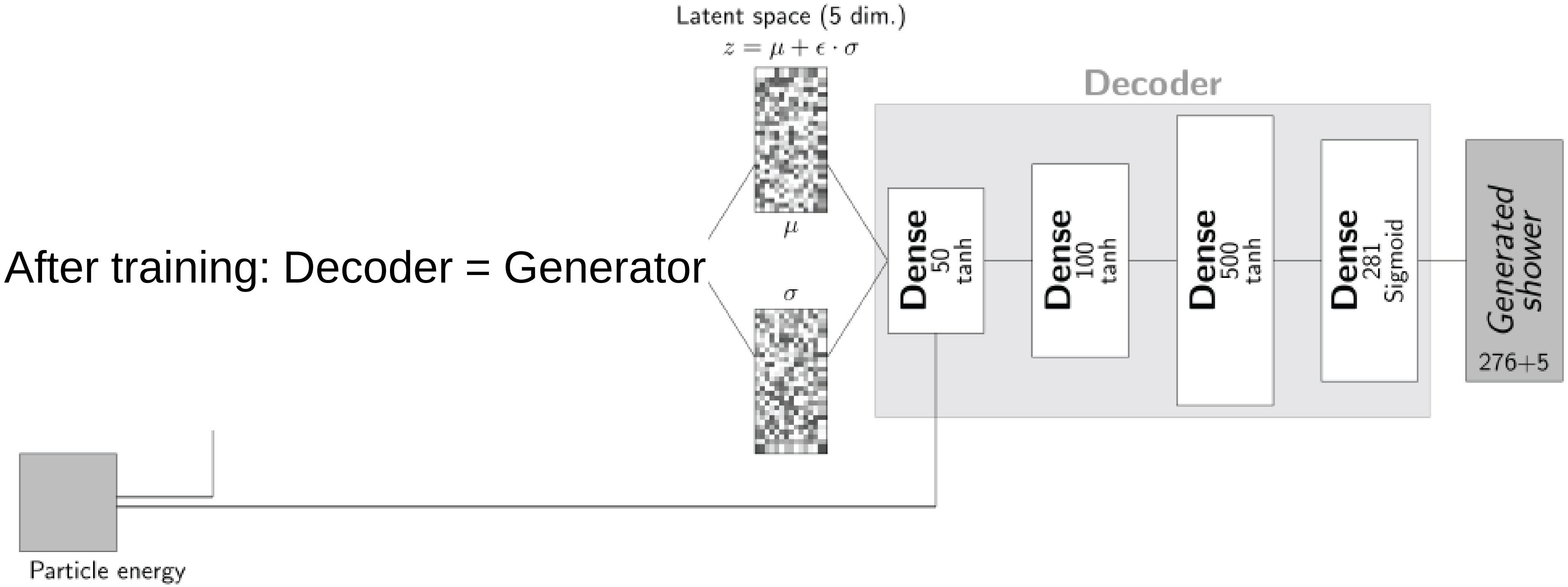
- Train on hay in reco mode
- **Rapidly** sample hay from a normal distribution

These faces do not exist !

[Karras et al., 2018]

VAE architecture

After training: Decoder = Generator



[ATL-SOFT-PUB-2018-001, ATL-PLOT-SIM-2019-007]

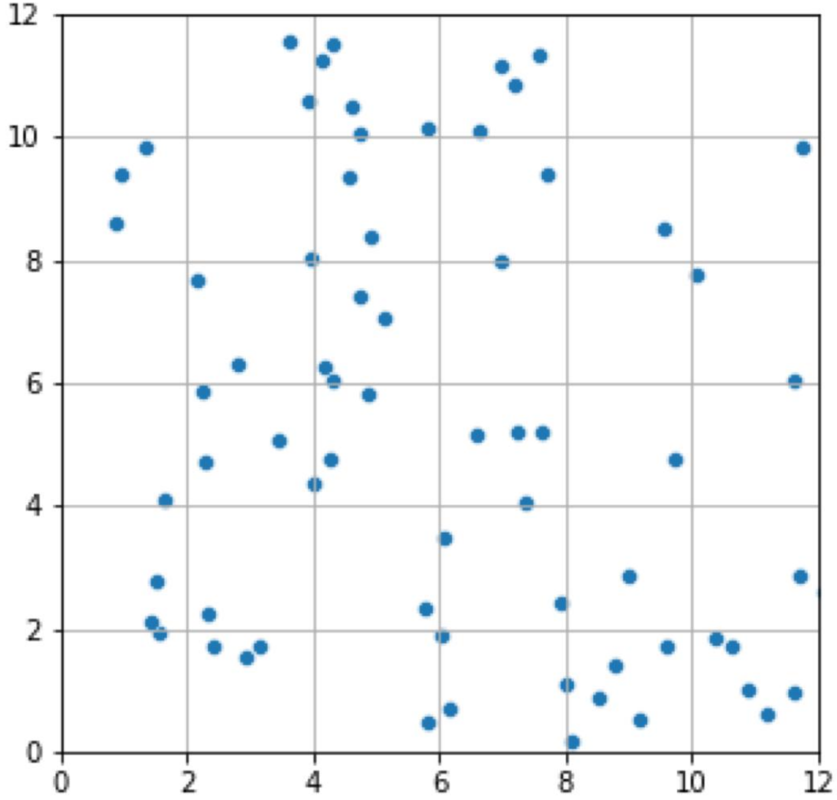
Validation: *marginals*

Generative modeling assessment

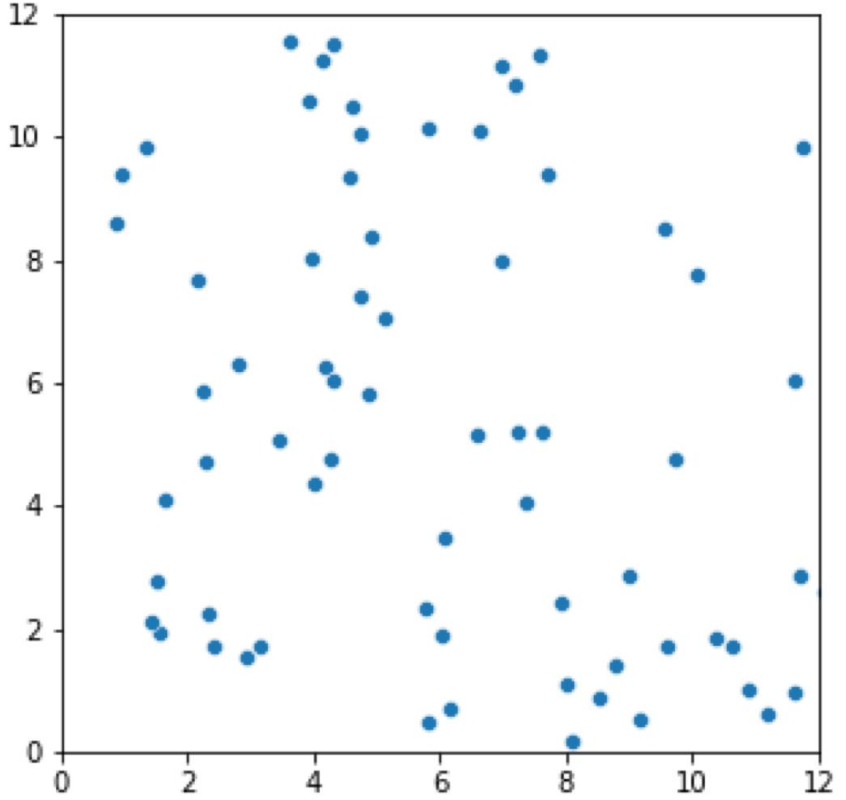
- Promising results but bottlenecks exist:
  - **Slow** development cycle
  - **Expensive & inflexible** training data (Geant4)
  - **Non-portable solution** highly dependent on detector geometry*
- Objectives:
  - Faster R&D
  - Decouple modeling from detector geometry → **point cloud format**

* A Common Tracking Software (ACTS) – portable tracking solution

Geant4 point cloud exists already



Current: mapping to fixed cells (**sparse**)
Intensity = sum of energy in each cell



Geant4 raw output: point cloud

The world of point-cloud data sets

[source]

Sweet spot ?

A SIMULATION TOOLKIT

- Existing public point cloud data sets
  - Not a good proxy for physics data
  - Improvements don't *generalize*

- Costly and expertise-requiring Geant4 simulation
  - Hard to scale complexity, change geometry, detector,…

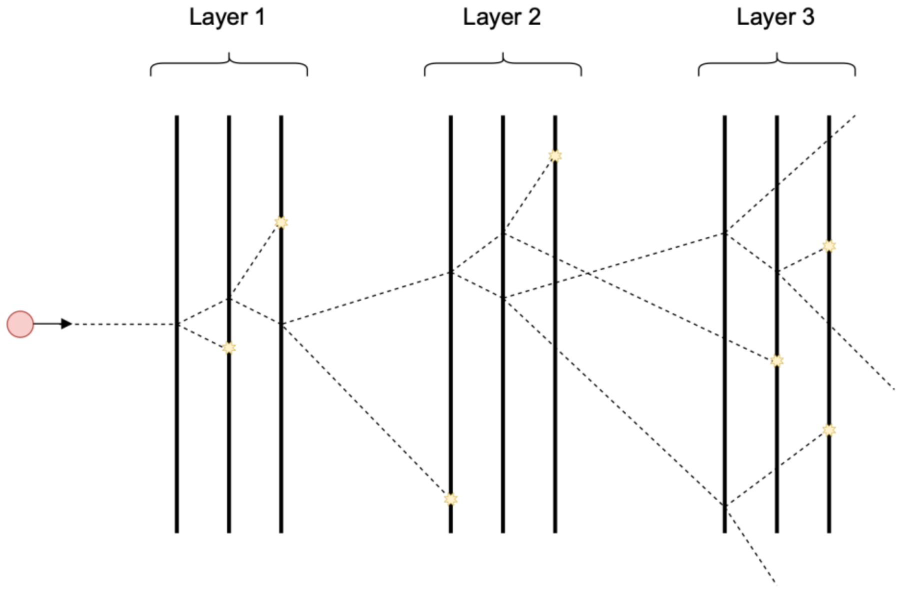
The  for generative modeling

- SUPA*
  - – Flexible & configurable proxy data sets

- Diagnostics tool to develop new generative surrogate simulators

- Point-cloud format promotes *GNN-based* generative models

Simplified
- particle propagation,
- scattering &
- shower development



*SUrrogate PArticle propagation simulator

[https://arxiv.org/abs/2202.05012]

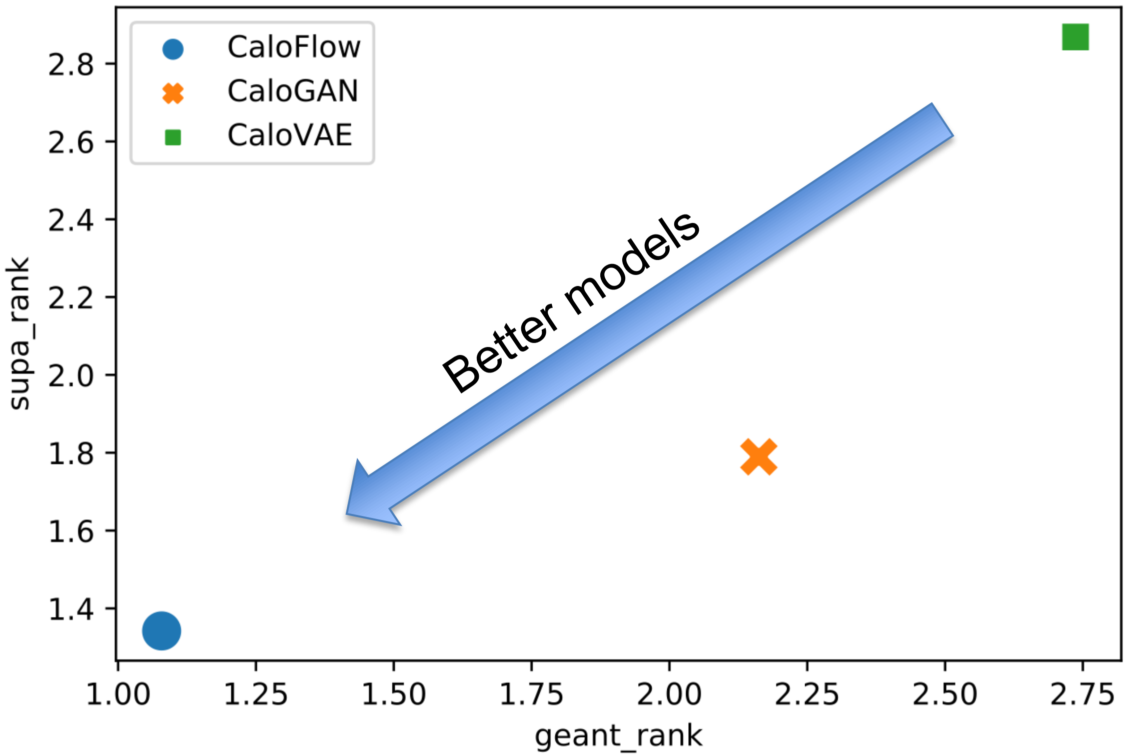
SUPA is *realistic enough*

- Improvements on SUPA translate to Geant4

Model design **on SUPA**:

- – Vary data complexity
- – Optimize model
- – Validation metrics

SUPA tracks improvements of model on Geant4



[https://arxiv.org/abs/2202.05012]

Modeling vs. learning

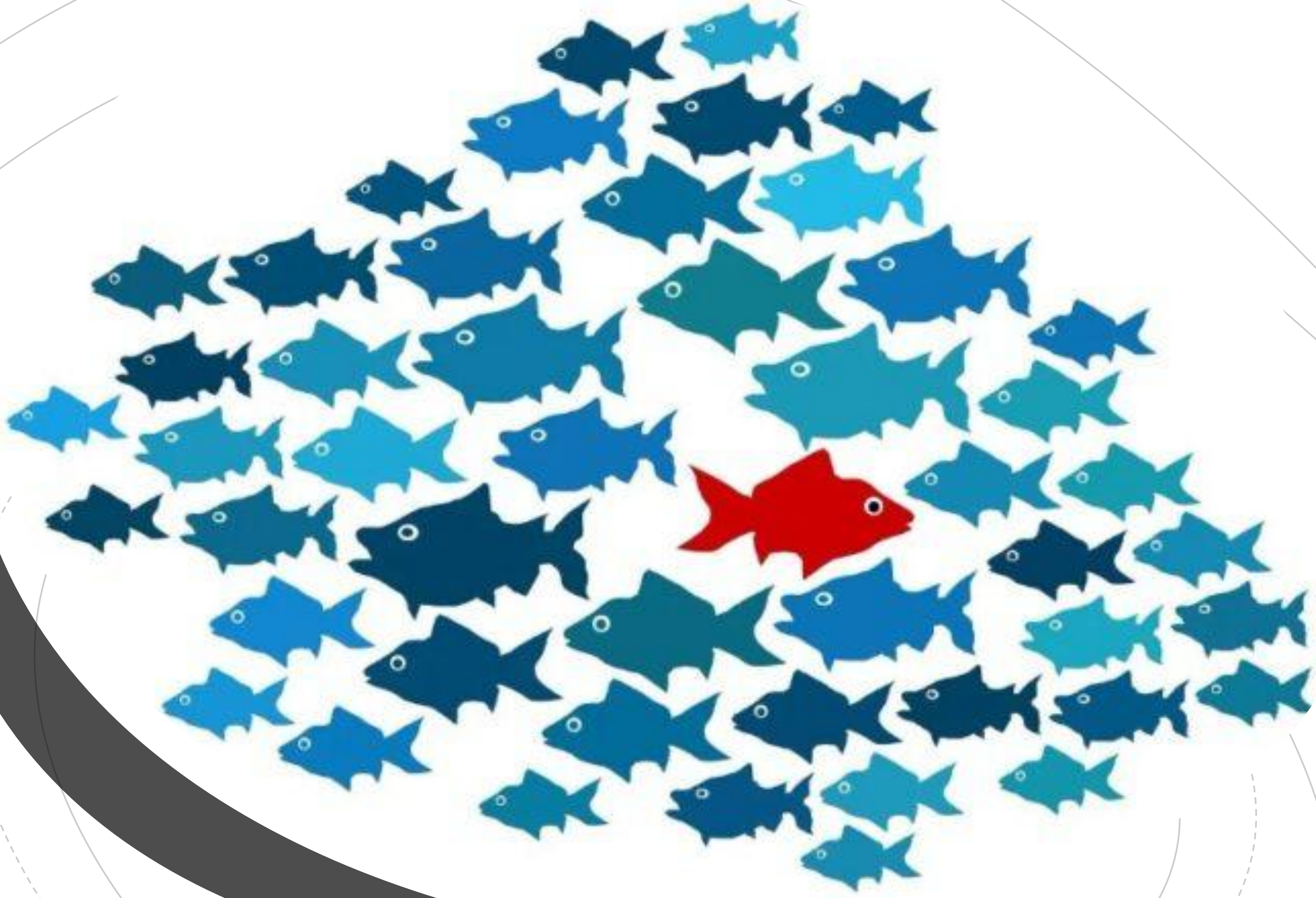
The world of modeling

- The Standard Model of particle physics
- High-fidelity Monte Carlo simulation
- Fast & accurate surrogate models

The world of learning

- Learning from **lots** of LHC data

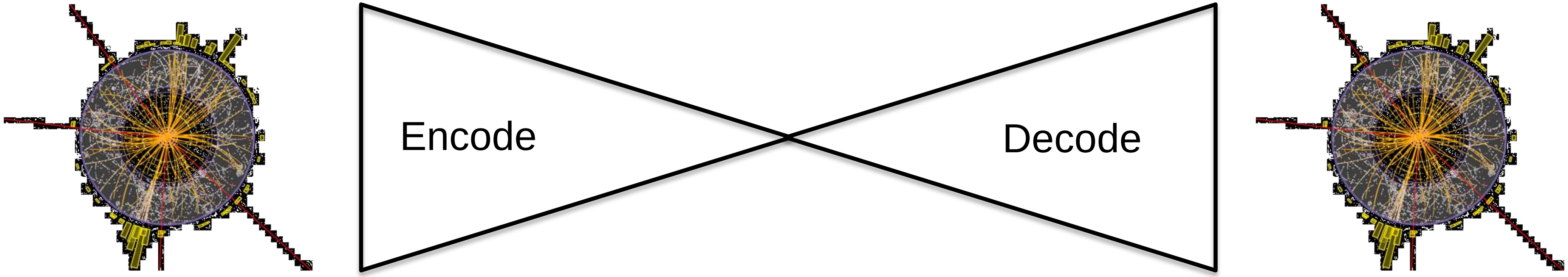
The best of both worlds?



Outlier detection

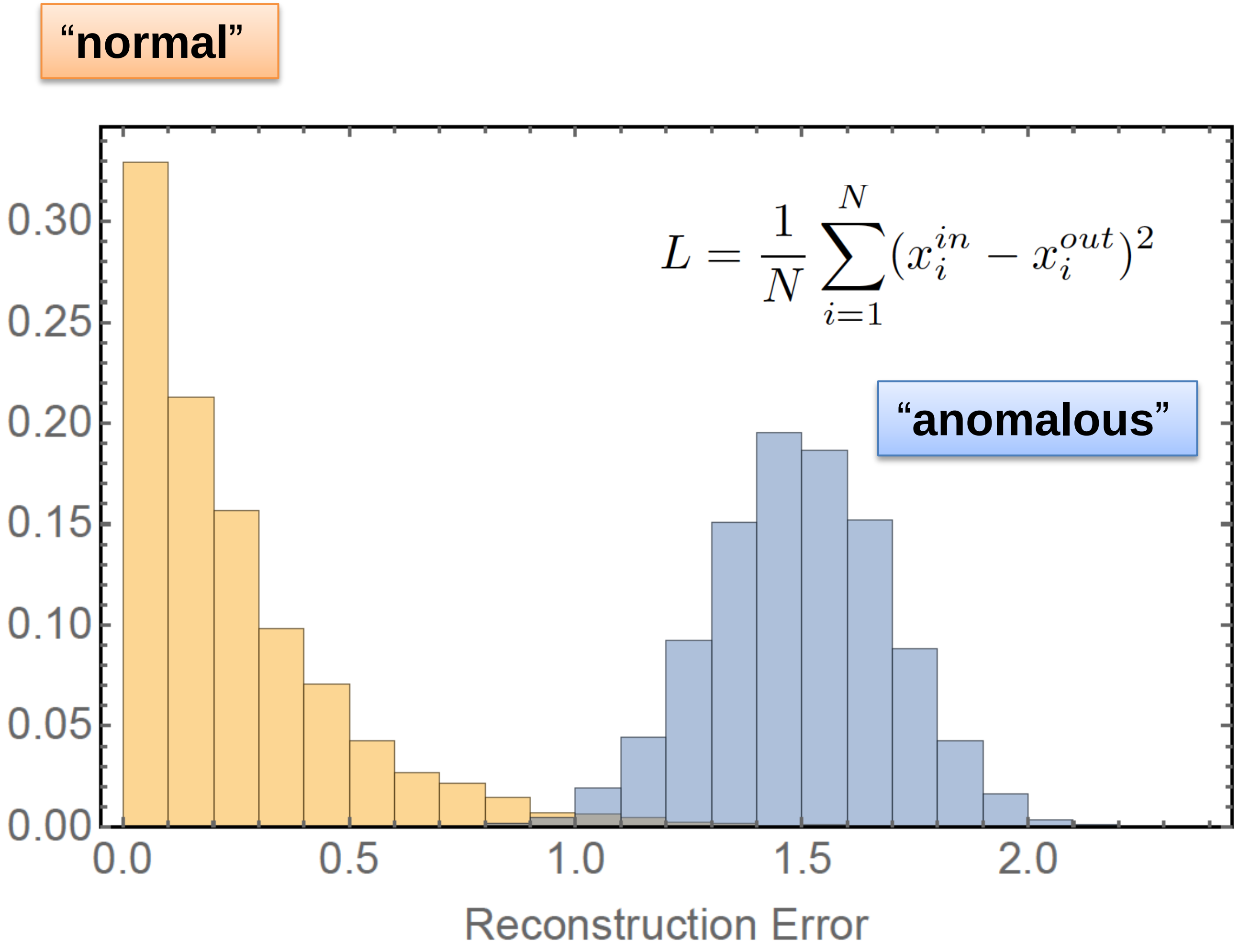
VAE in reconstruction mode: search for anomalous boosted objects

Encode and decode “normal” objects / events



Compare original and reconstructed image

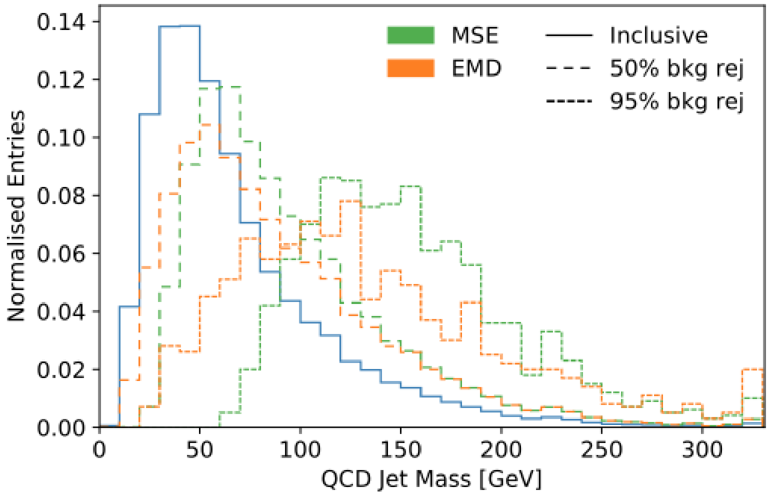
Anomalous jets



[1709.01087, 1808.08979, 1808.08992, 1905.12651, 2007.01850]

Challenge:

- Tool picks up mainly on *dominant* difference, i.e. the mass of the anomalous jet

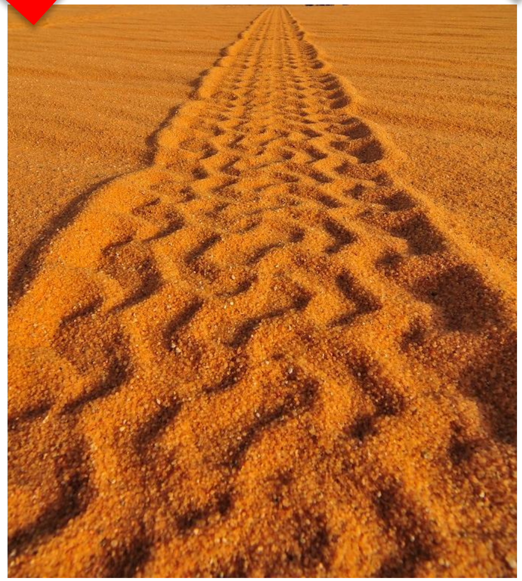


[https://ml4physicalsciences.github.io/2020/files/NeurIPS_ML4PS_2020_56.pdf]

The problem with outlier detection

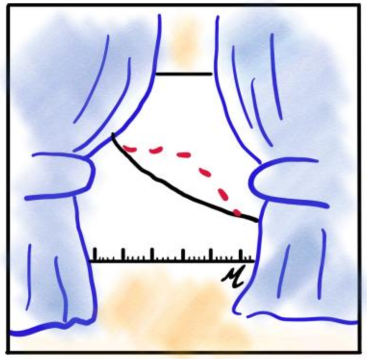
- Rarely *true* outliers in our data

- We look for an excess = over-density



Constructing **U**nobserved **R**egions by **T**ransforming **A**djacent **IN**terval**s**

All windows need ***CURTAINs***



Data driven method for constructing

background templates with **arbitrary variables**

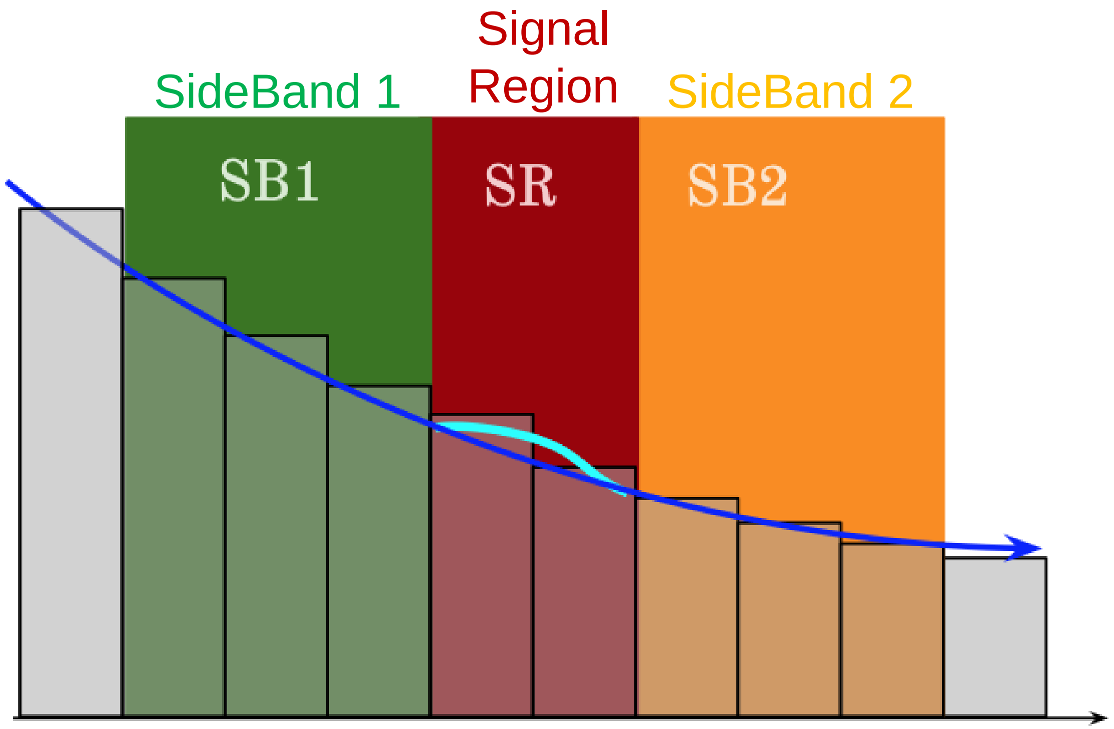
[https://arxiv.org/pdf/2203.09470.pdf]

Bump hunt

Focus on resonant signal = **bump**

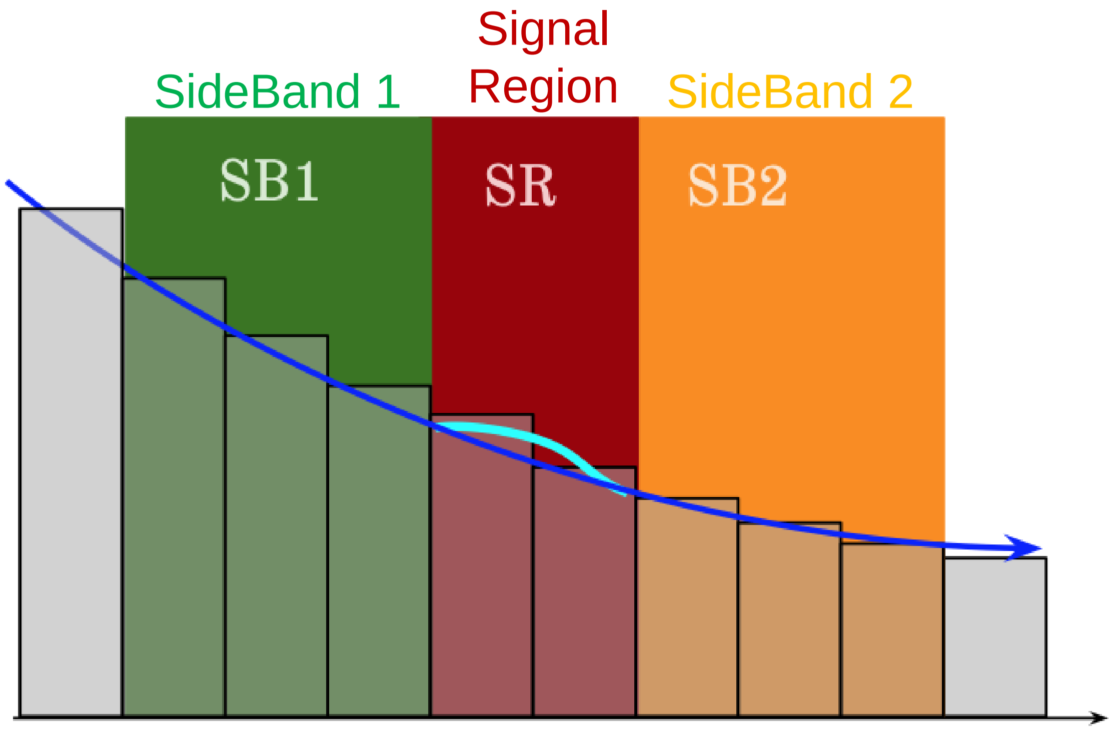
Method:

1. Split spectrum into sliding SBs
2. Fit the distribution in SBs
3. Interpolate into the SR
4. Look for an excess



Extended bump hunt

- Looking for tiny signal
- Increase sensitivity to new physics
  - ⇒ **use additional observables**
- Observables often strongly correlated to the mass
- **Interpolate** to find BG template in SR



CURTAINs approach

1. **Transform** data from **the SBs** into the **SR**

2. Transformed side bands = background template

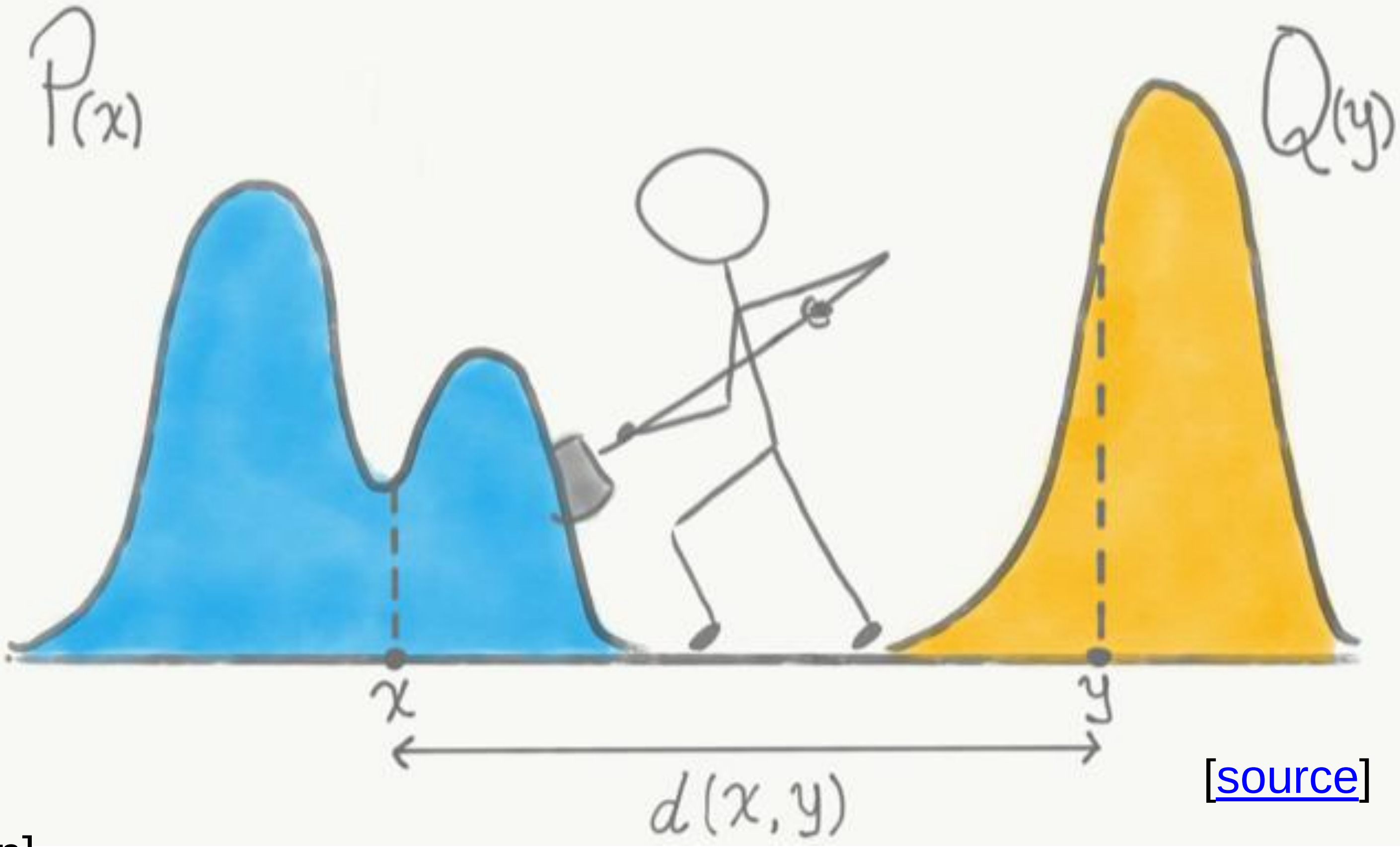
3. Train a classifier to separate background from *signal*

Toolbox: optimal transport

- Transforming **P** into **Q** while minimizing a cost

- Cost based on **distance d** between data points

Originally about transporting dirt…



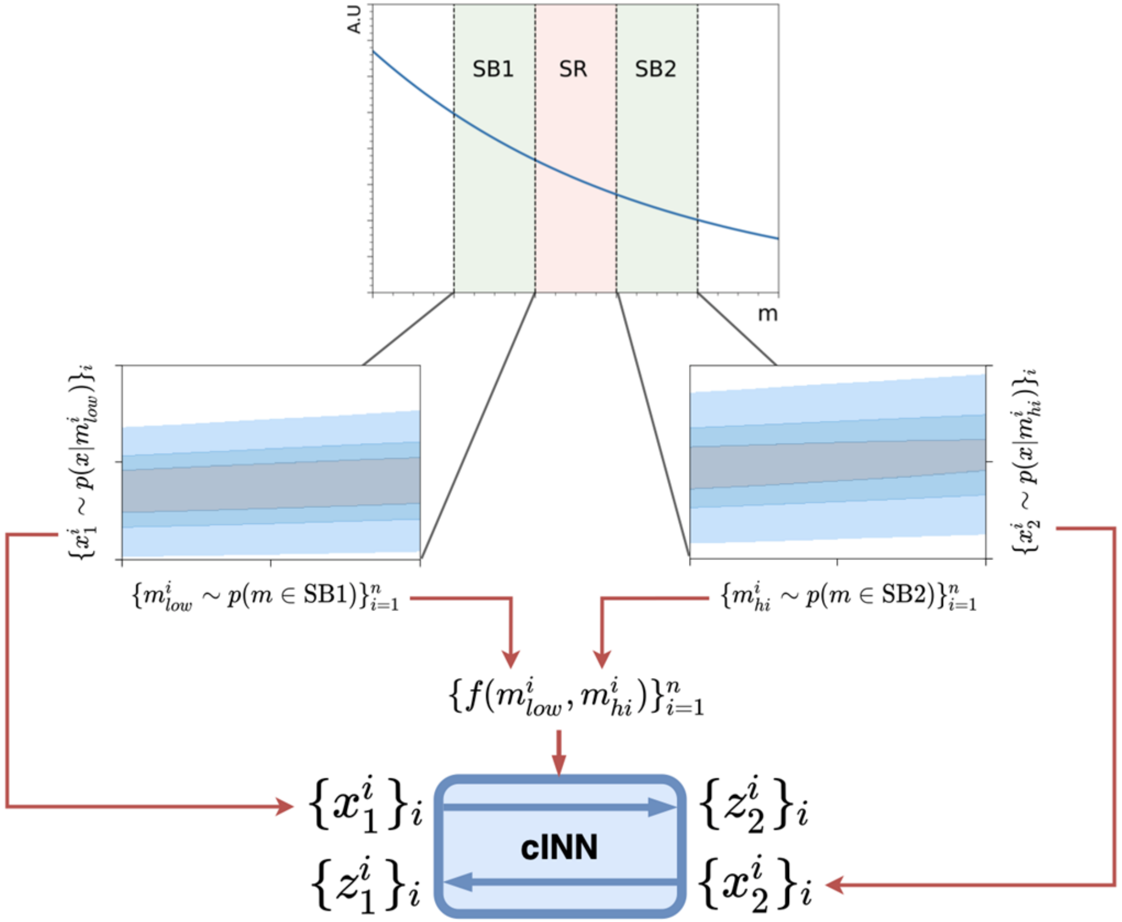
[source]

[Approximate Wasserstein distance with Sinkhorn]

Training "SB-to-SR" transformation

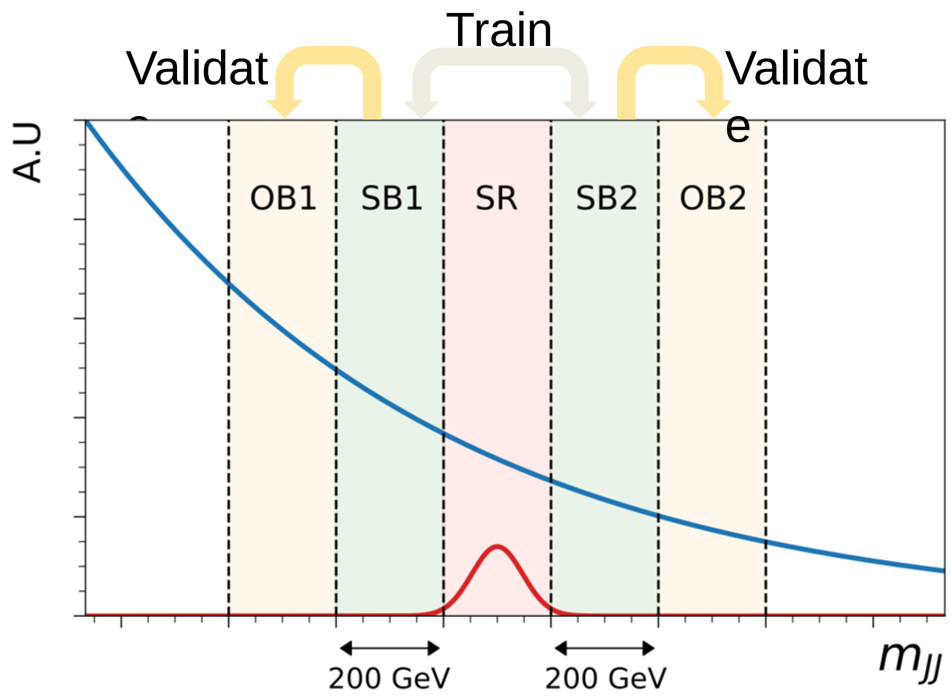
- Use a **conditional invertible** neural network (cINN)

- Map from SB1 to SB2 and vice versa



CURTAINs validation

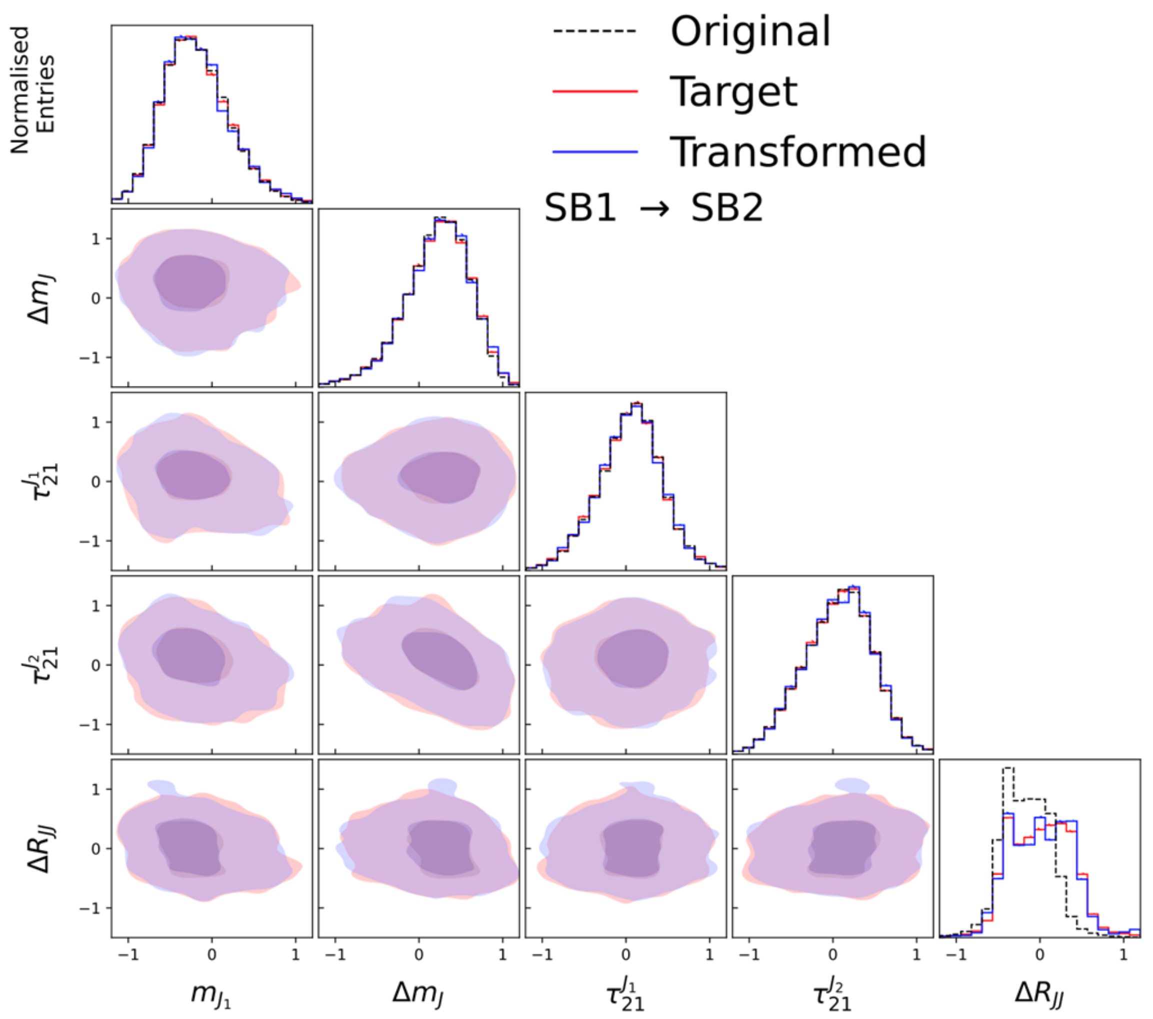
- Fix sidebands
- Define OuterBand (OB) validation regions
- Train CURTAINs transformer
- Validate on OBs



Training data

SB1: [3200, 3400] GeV
SB2: [3600, 3800] GeV

- Training on the LHC Olympics R&D dijet dataset*
  - Based on jet substructure & ΔR_{jj}
- SB1 → SB2
  - as good for SB2 → SB1, OBs, SR



*[https://doi.org/10.5281/zenodo.4536377]

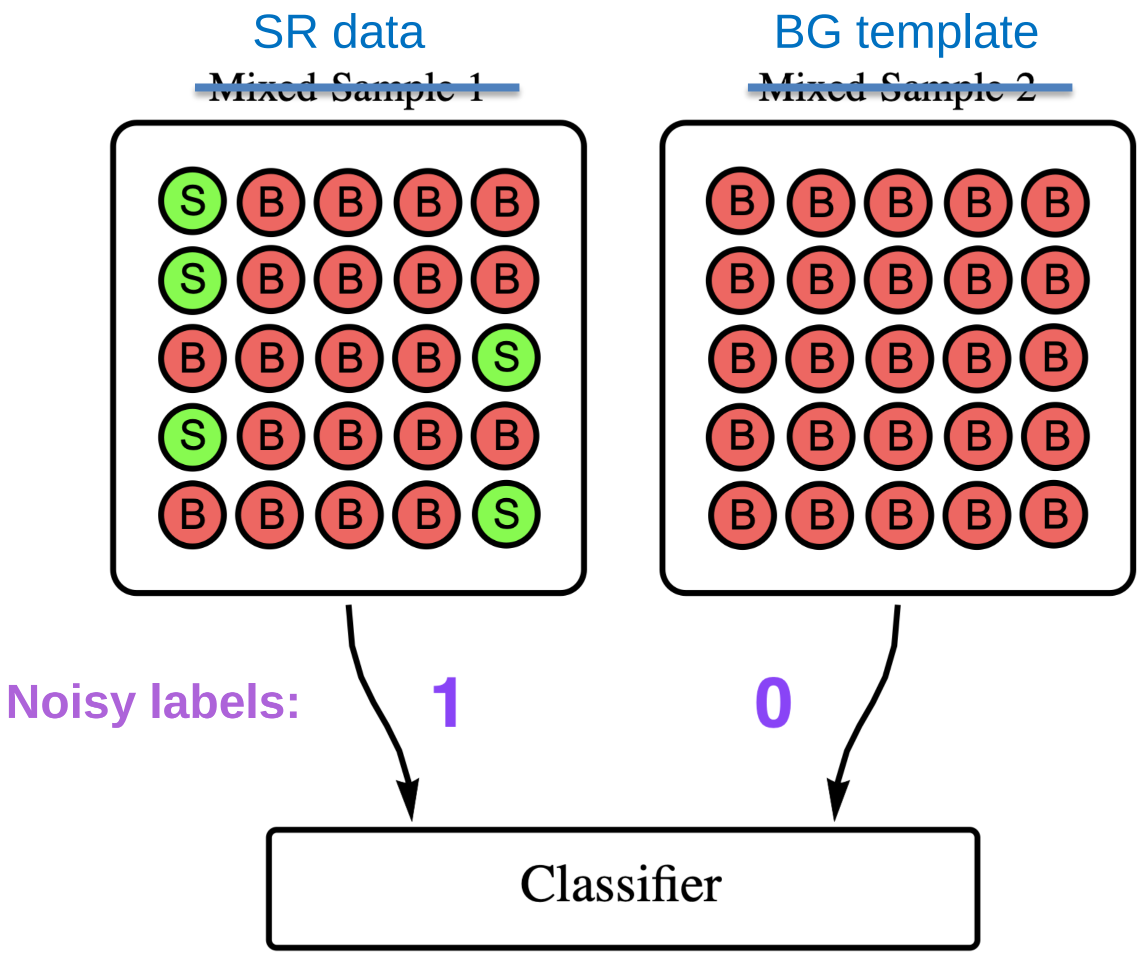
CURTAINs so far

✓**Transform** data from **the SBs** into the **SR**

✓Transformed side bands = background template

❑Train a classifier to separate background from *signal*

Classification without labeling (CWoLa)

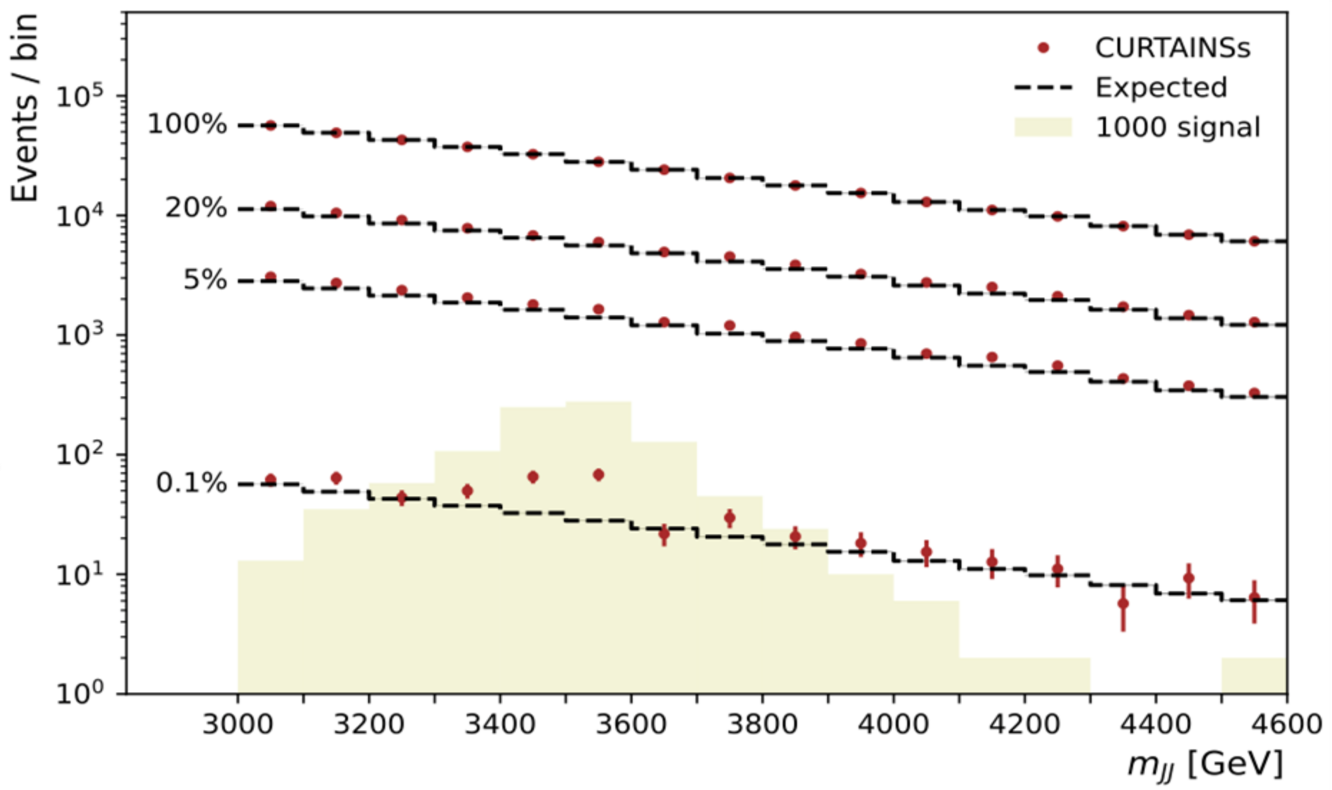


- Use noisy labels
- Shown to be optimal classifier
- Apply to data-only
- CWoLa for CURTAINs

[1702.00414, 1708.02949]

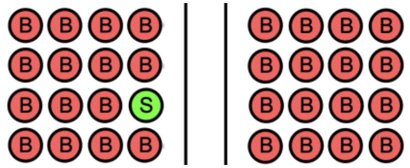
CURTAINs in action

- True BG (Expected)
- Predicted BG from CURTAINs
- Add signal

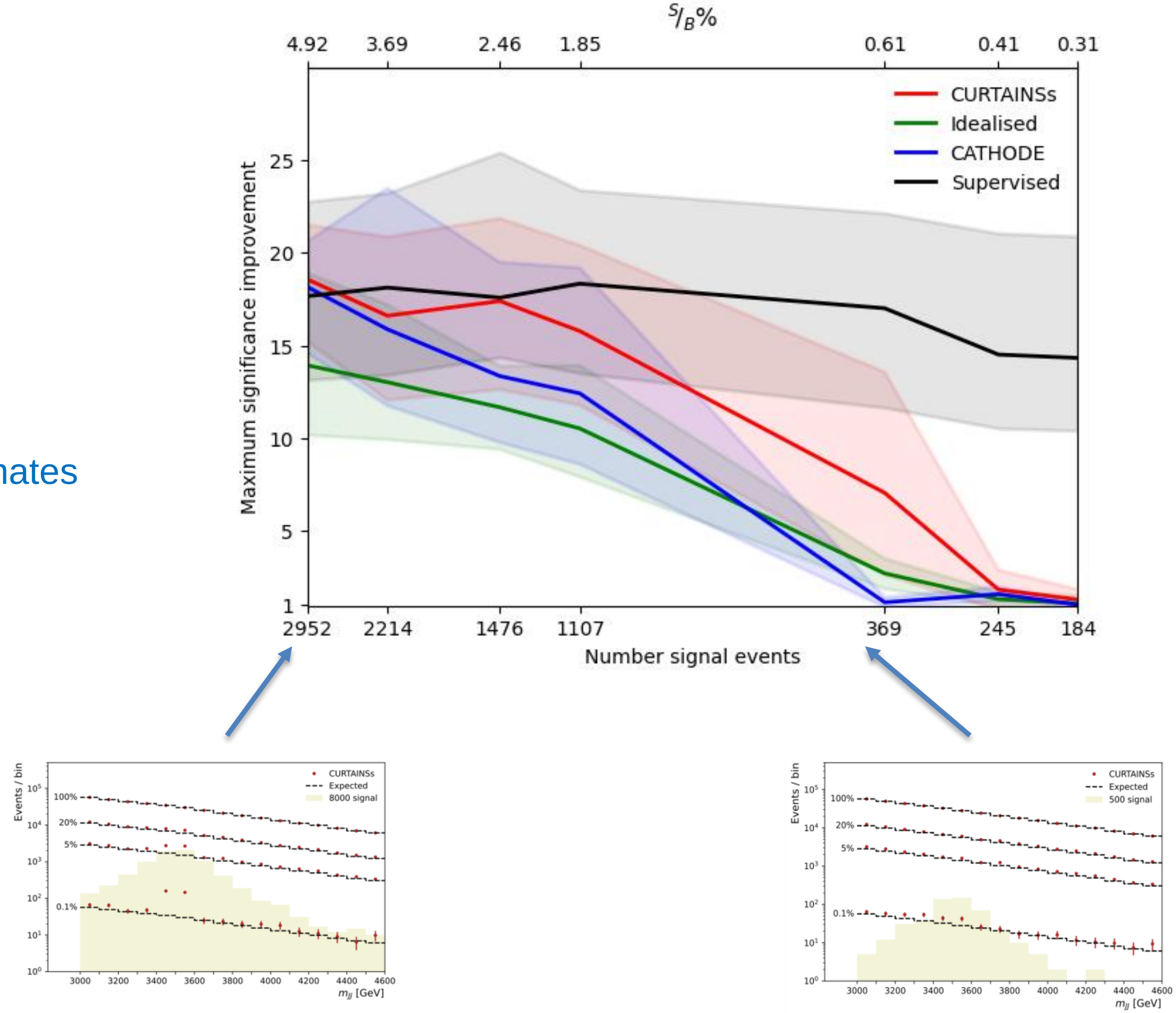


Apply cut on CWoLa classifier

CURTAINs performance



- CURTAINs
- Idealised: assume perfect BG template
- CATHODE
  - *Competition* generating BG template from density estimates
- Supervised



[CURTAINs > Idealised due to *oversampling*]

Summary

- *Extend* LHC's physics portfolio to anomaly detection

- Key: robust background estimate
  - Data-derived: CURTAINs
  - MC modeling: speed & accuracy with generative models
  - Work in progress: **combine modeling & learning**

- Promote automation & reduce complexity