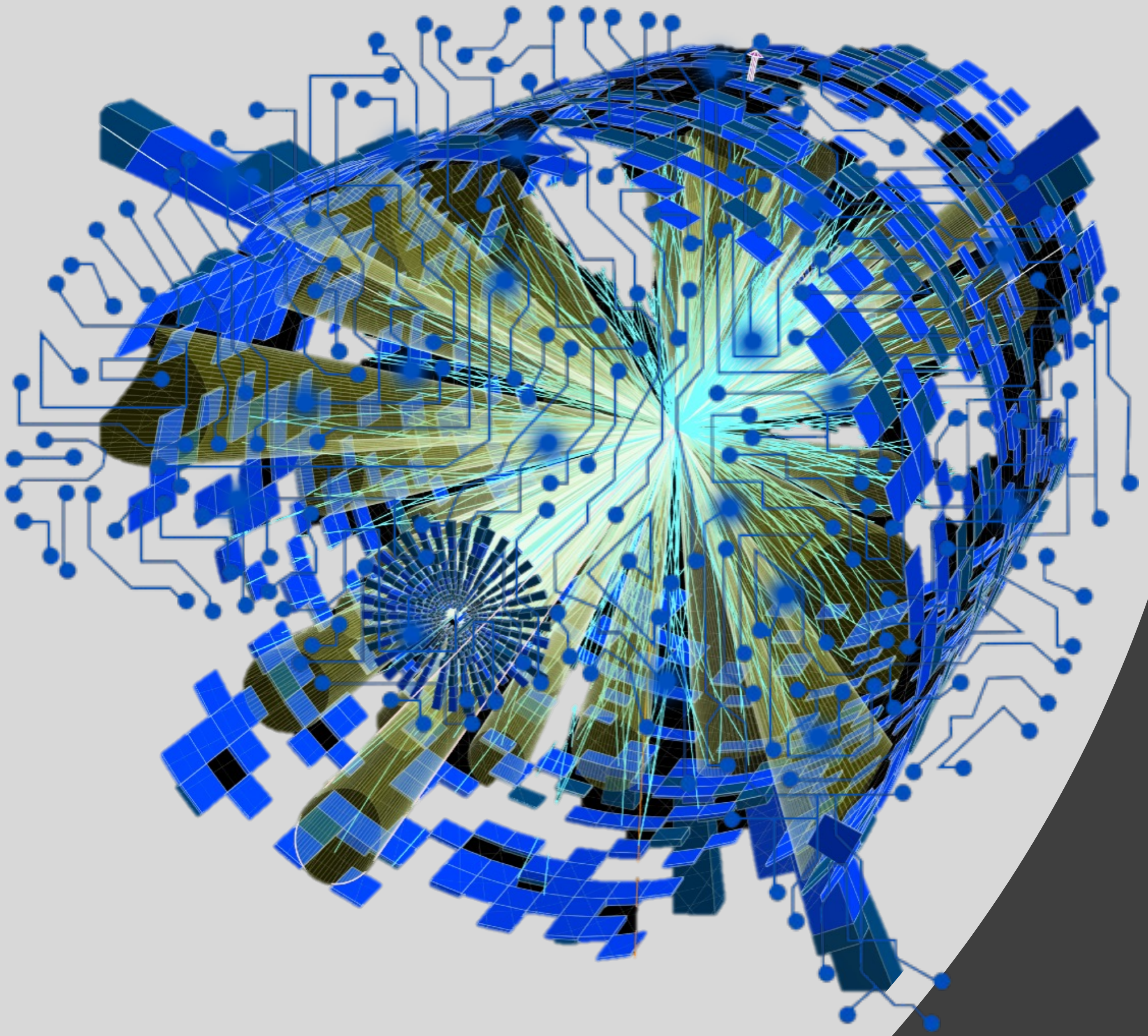


Simulating Reality & Searching for the Unknown

And some things in between

Tobias Golling,
University of Geneva

Disclaimer

- I am an ATLAS member
- Examples I will show are highly biased
  - Personal preference
  - ATLAS bias (please read ATLAS = CMS)
- The main messages are ~independent of these biases

Approaching from both sides

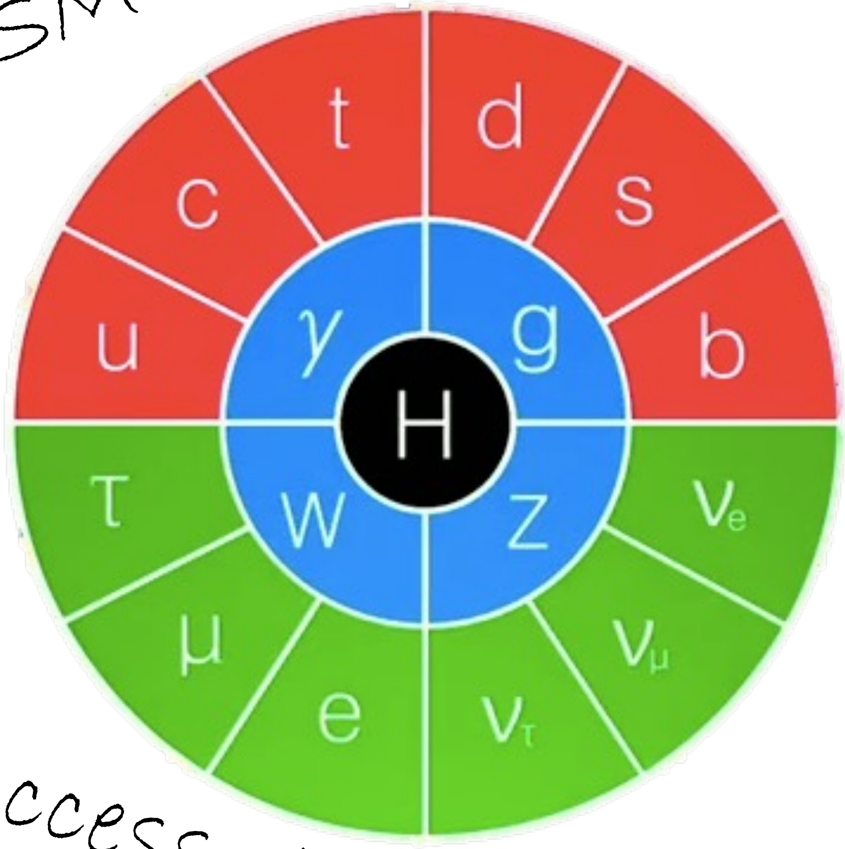
- The HEP challenges
- The Machine Learning (ML) *buffet*
  - (FF, CNN, RNN, GNN, DeepSets, transformers, VAE, GAN, NF,…)
- A lego-game of *mix, match, augment,…*
  - Lots of fun R&D: exploit strengths vs. weaknesses
- A spin-off question: more generic solutions?

Outline

- Establish the goal: maximize LHC's sensitivity to new physics
- The supervised approach
- Extend LHC's physics portfolio to model-agnostic searches
- The need for accurate and fast background modeling
- Machine learning strengths
  - Better
  - Automate
  - Reduce complexity

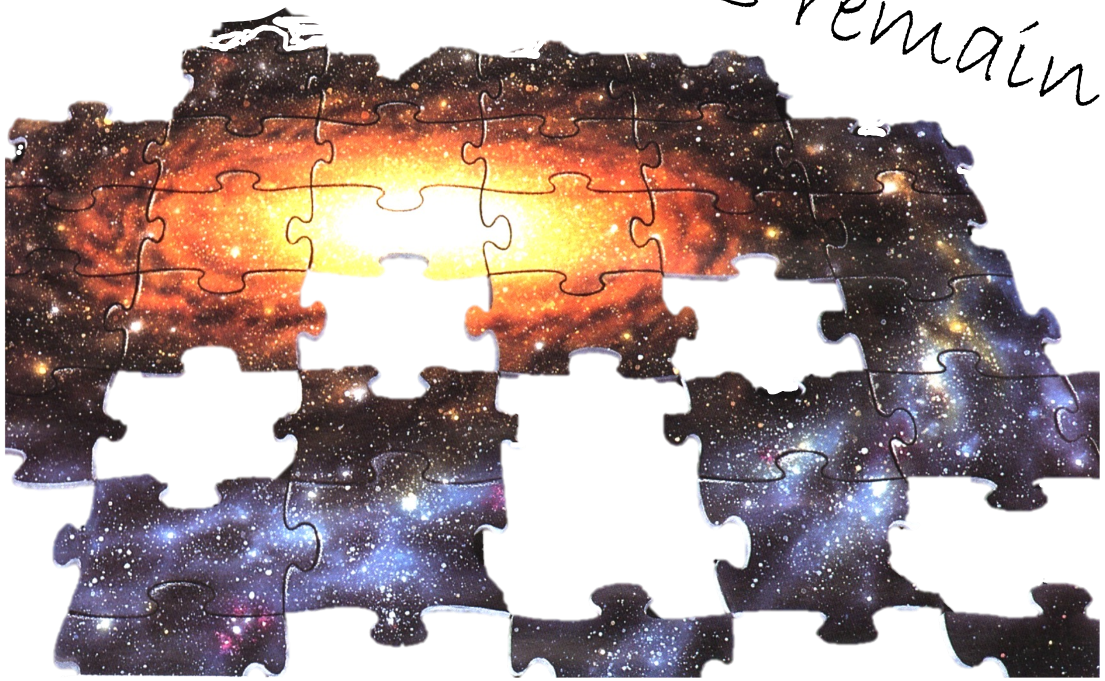
The current situation

The SM is complete



Success story!

Open mysteries remain

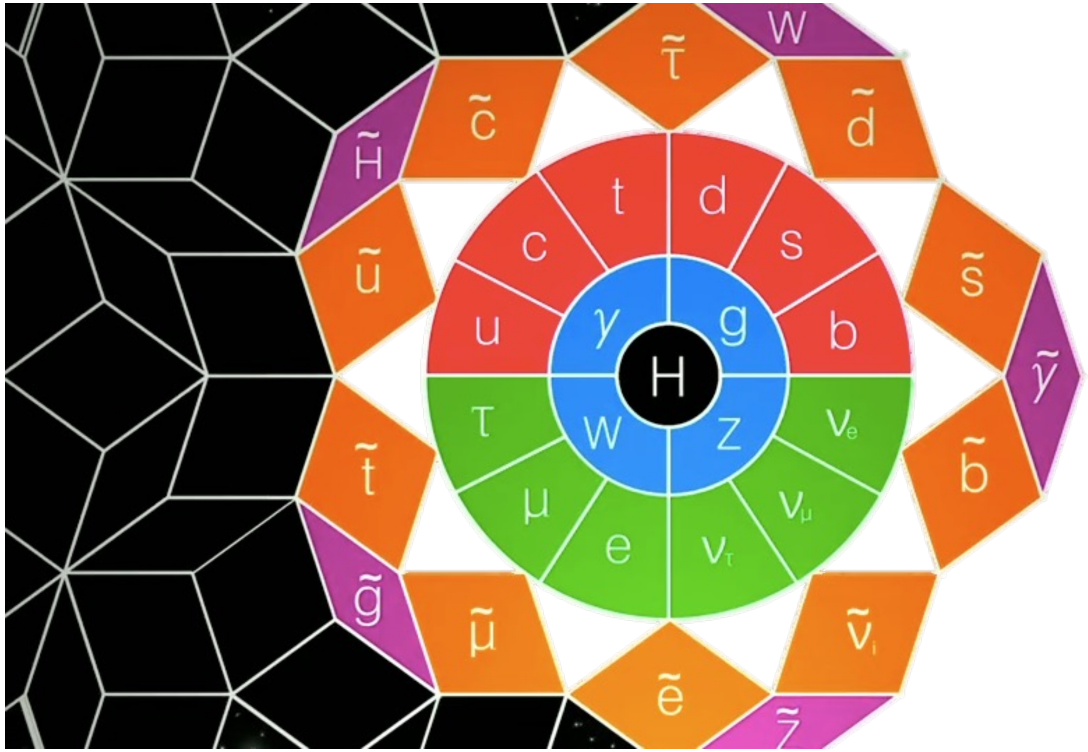


Dark matter, dark energy, quantum gravity,...

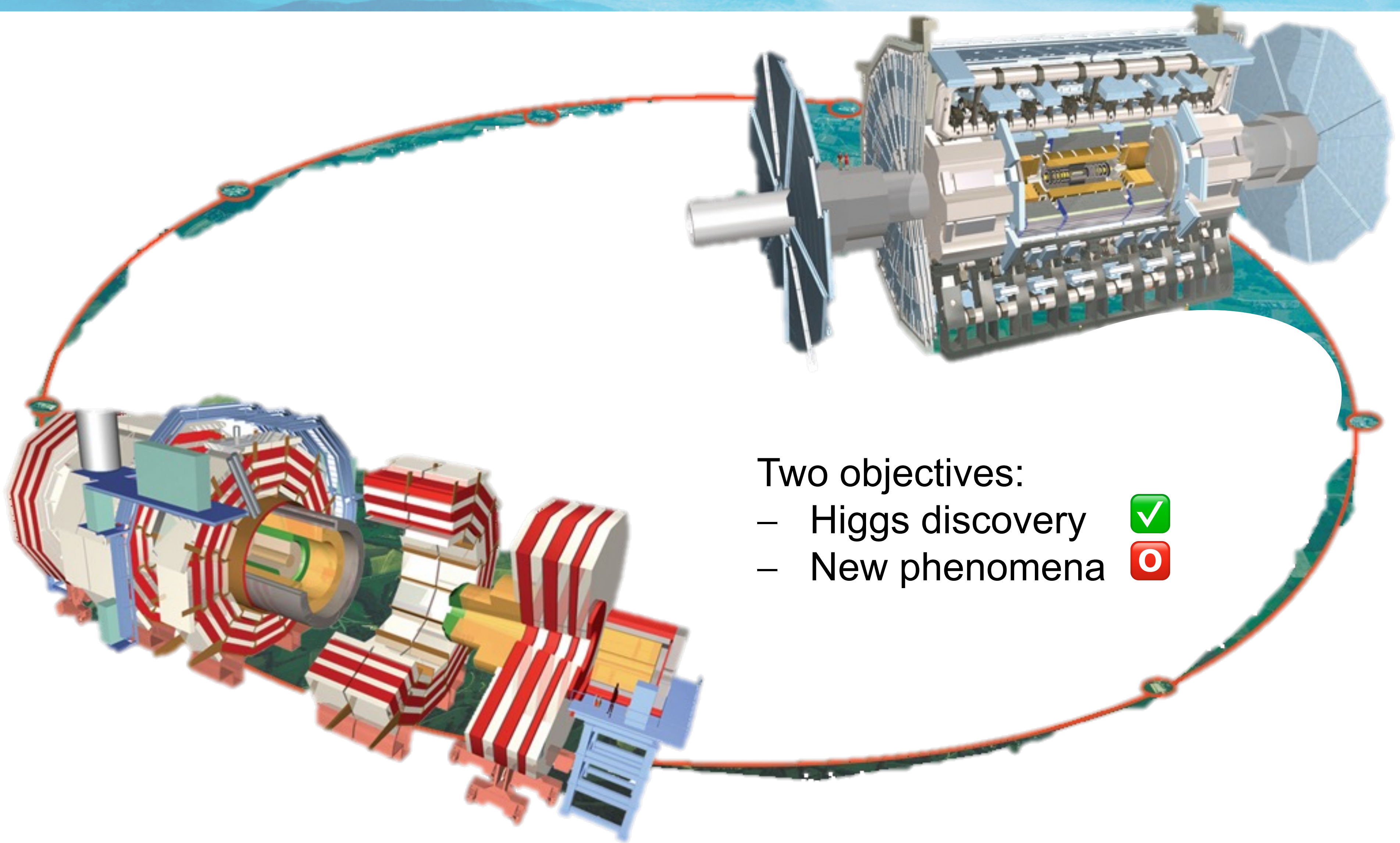
The theory guidance

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

- Plethora of Beyond-the-SM extensions…

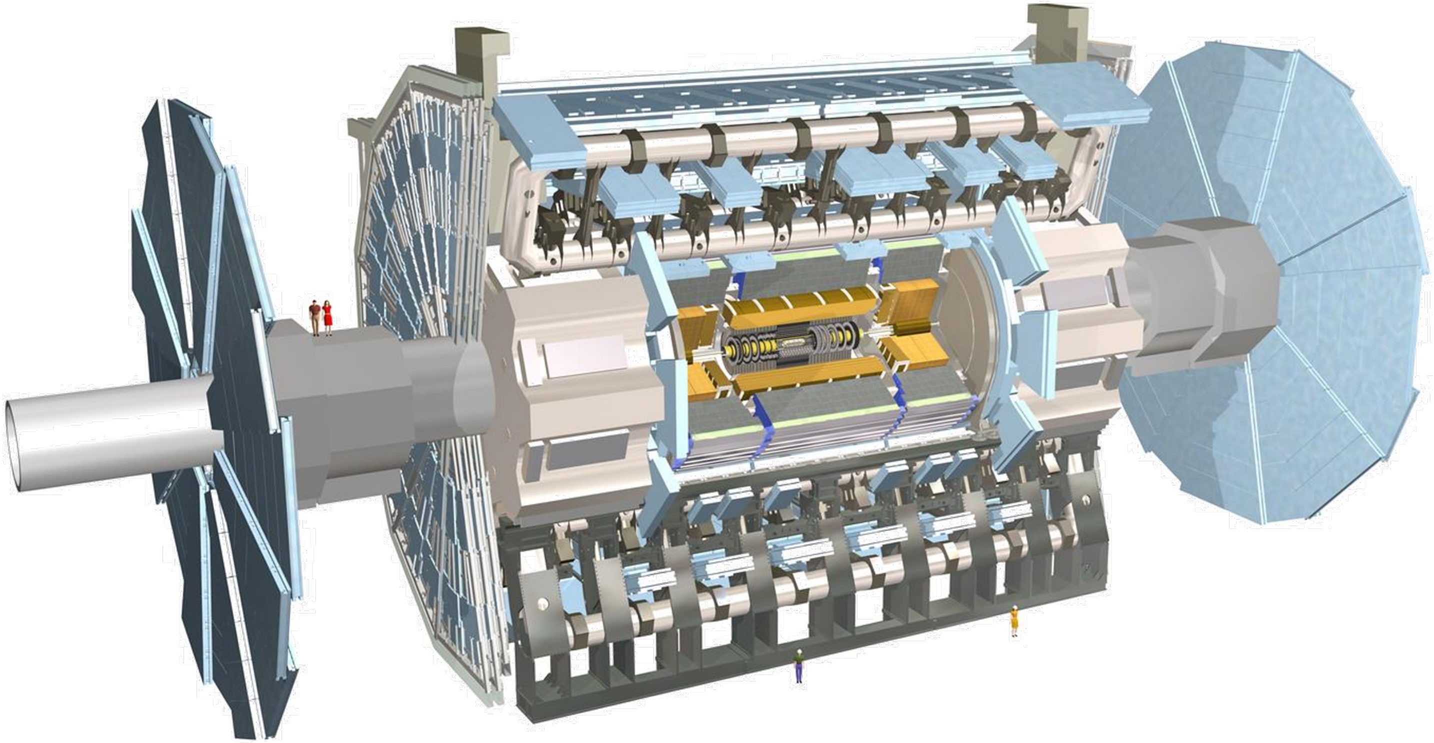


The Large Hadron Collider (LHC)



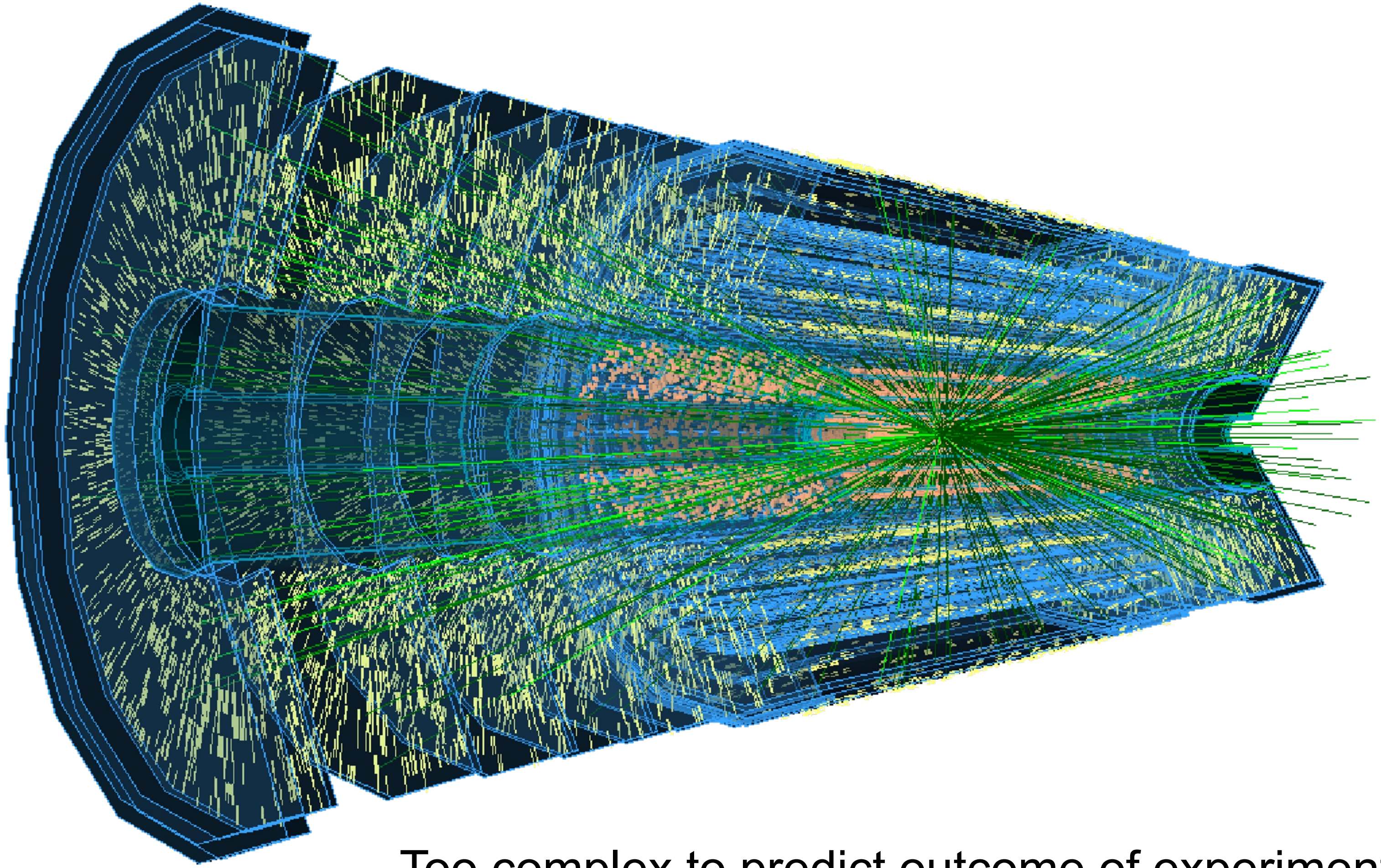
Two objectives:
– Higgs discovery ✅
– New phenomena ⭕

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

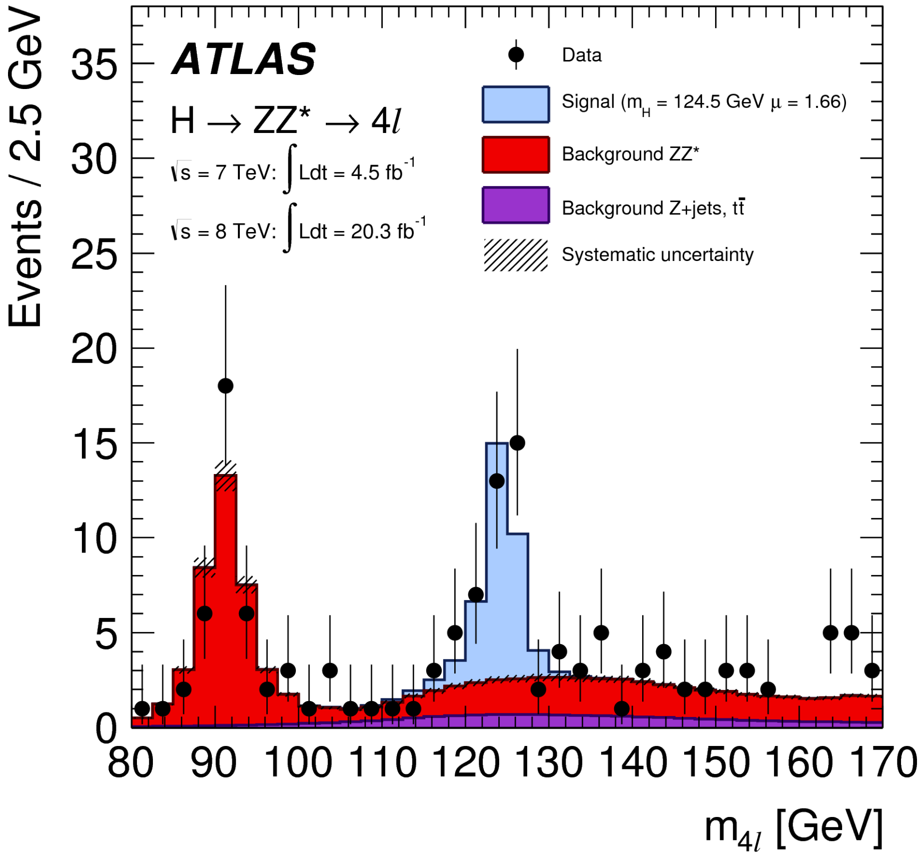


Too complex to predict outcome of experiment from first principles
→ **Monte Carlo simulation**

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

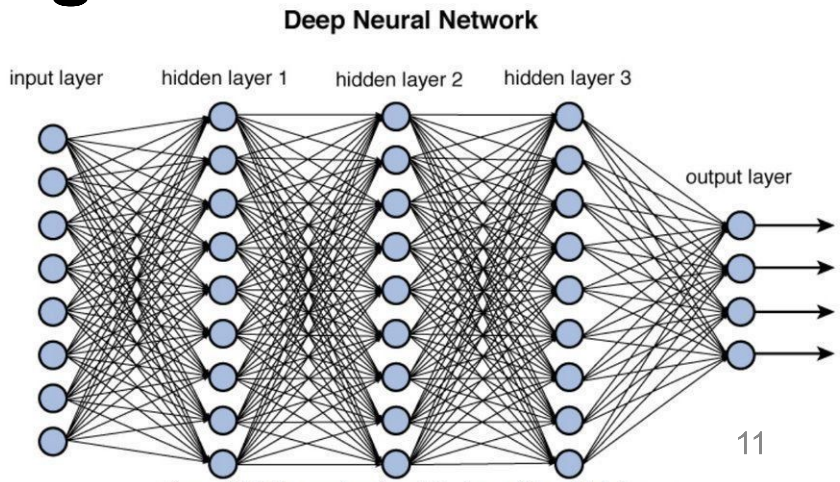


Figure 12.2 Deep network architecture with multiple layers.

Analogy: searching the needle in the hay

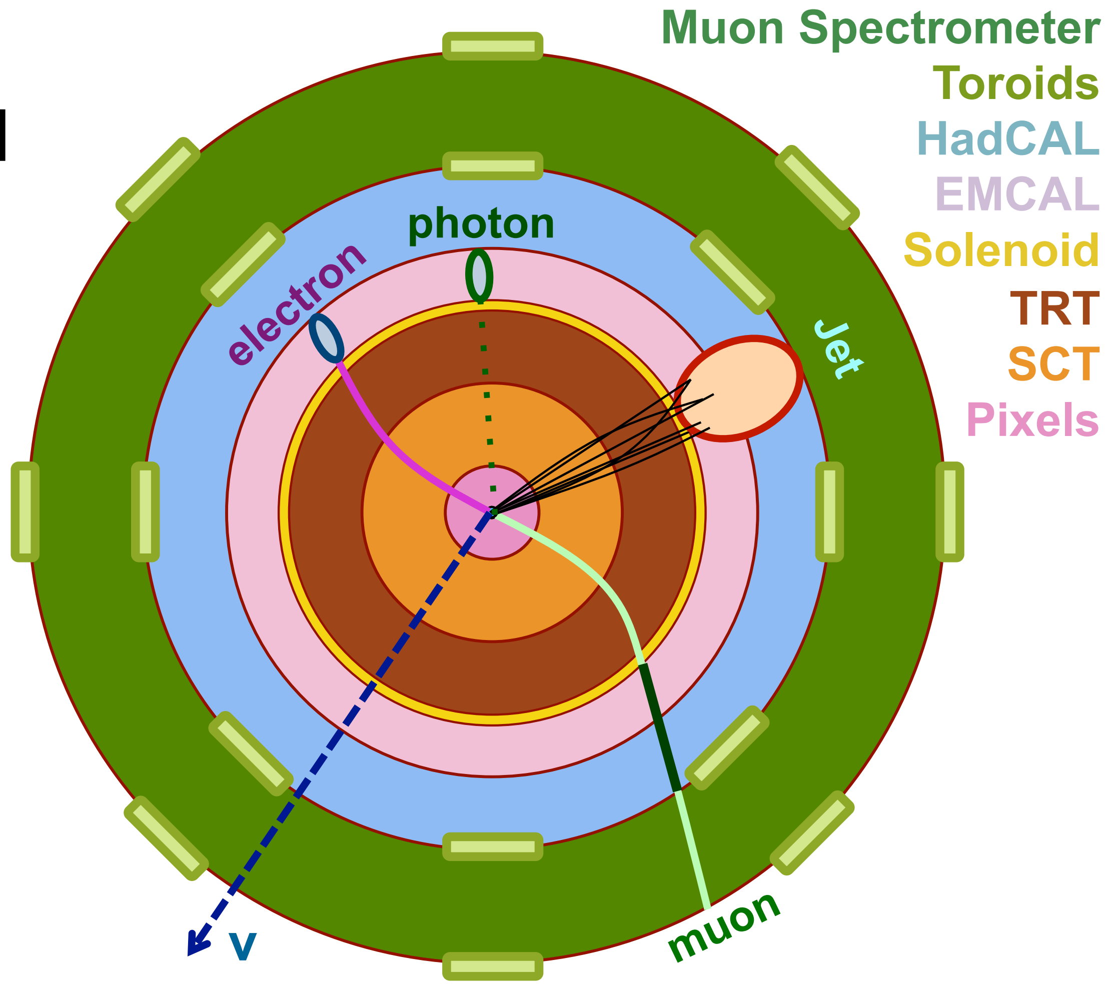
1. Searching for the **known**

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

Break problem down into physics objects

Classify particles based on **labeled** synthetic data (**supervised**)

- Large statistics
- Multi-classification
- Maximum impact
- Excellent modeling



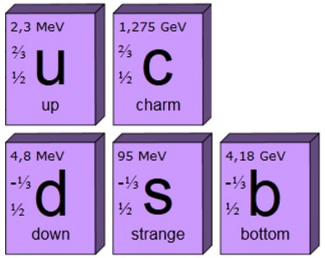
Example: flavor tagging

- Domain in particle physics with longstanding and very active history of ML usage

- Successful exploration of:
  - Data representations
  - Learning algorithms

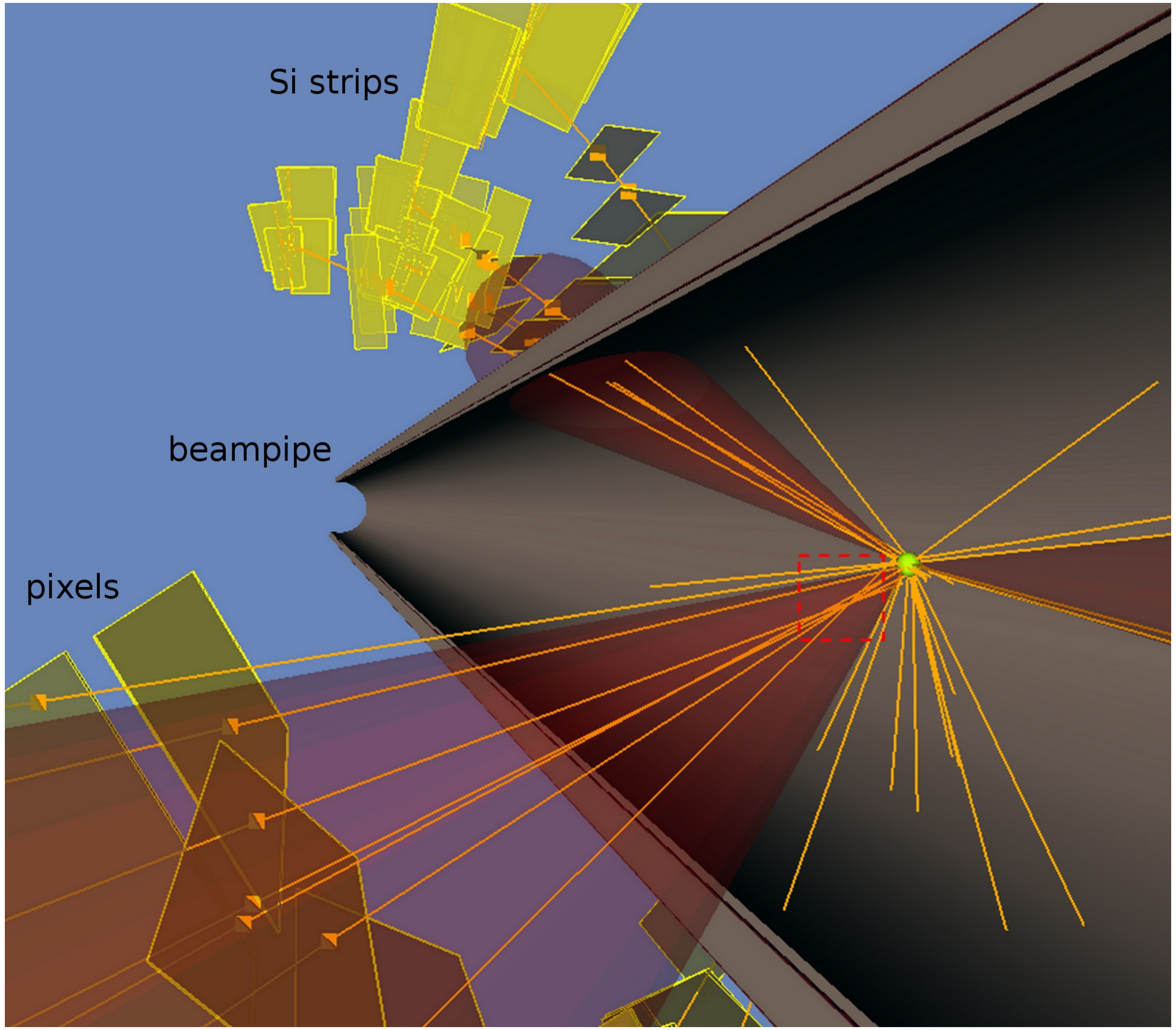
B-tag mini-lecture

- Quark hadronizes to collimated bunch of hadrons = *jet*
- They come in flavors
  - *c*-jet
  - *b*-jet
  - light-jet



- Interesting physics: *b*, *c*
- Task: identify jet flavor
- Train on truth-labelled simulation data

Visualizing a *jet* in a collision

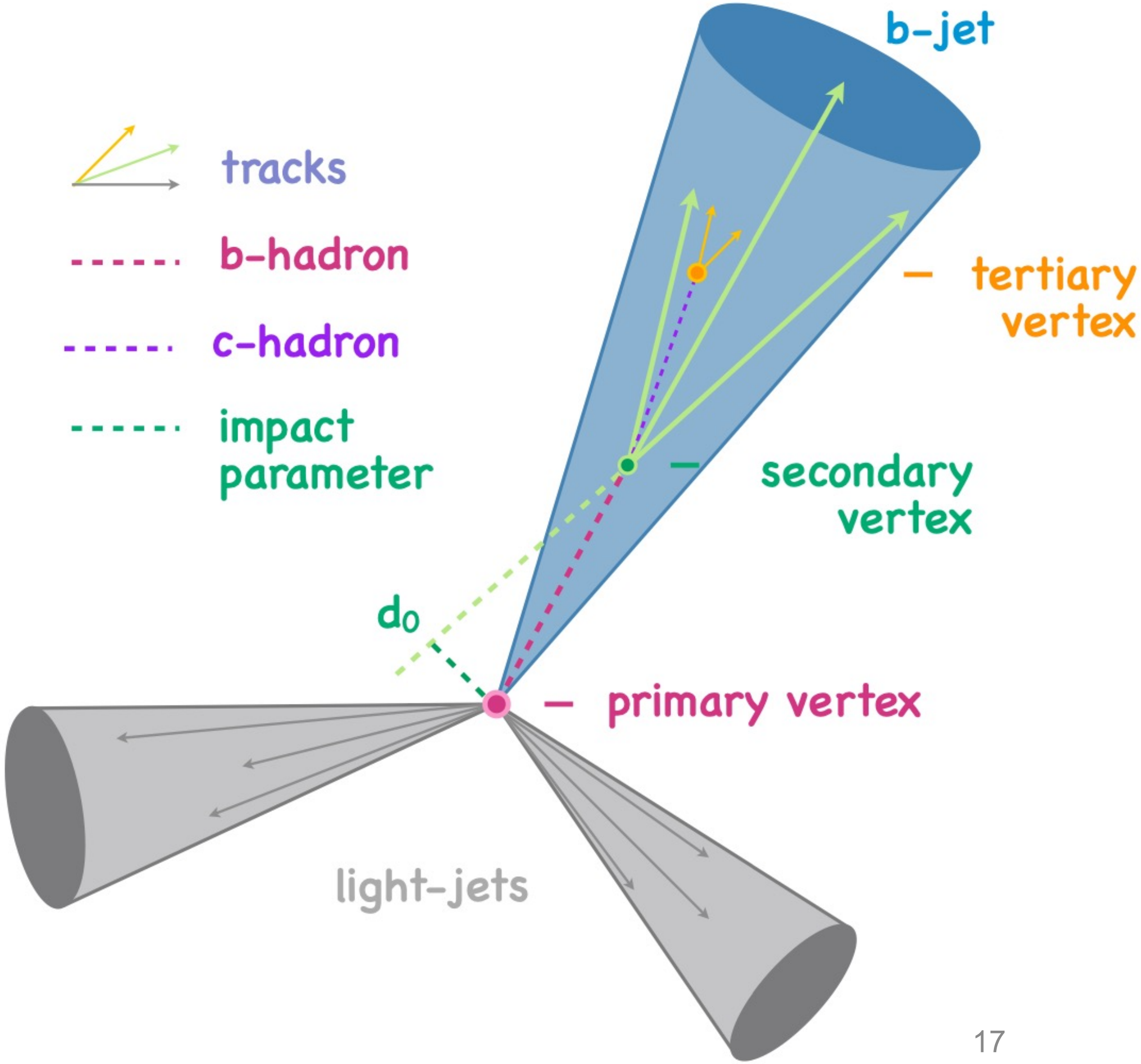


[ATLAS experiment]

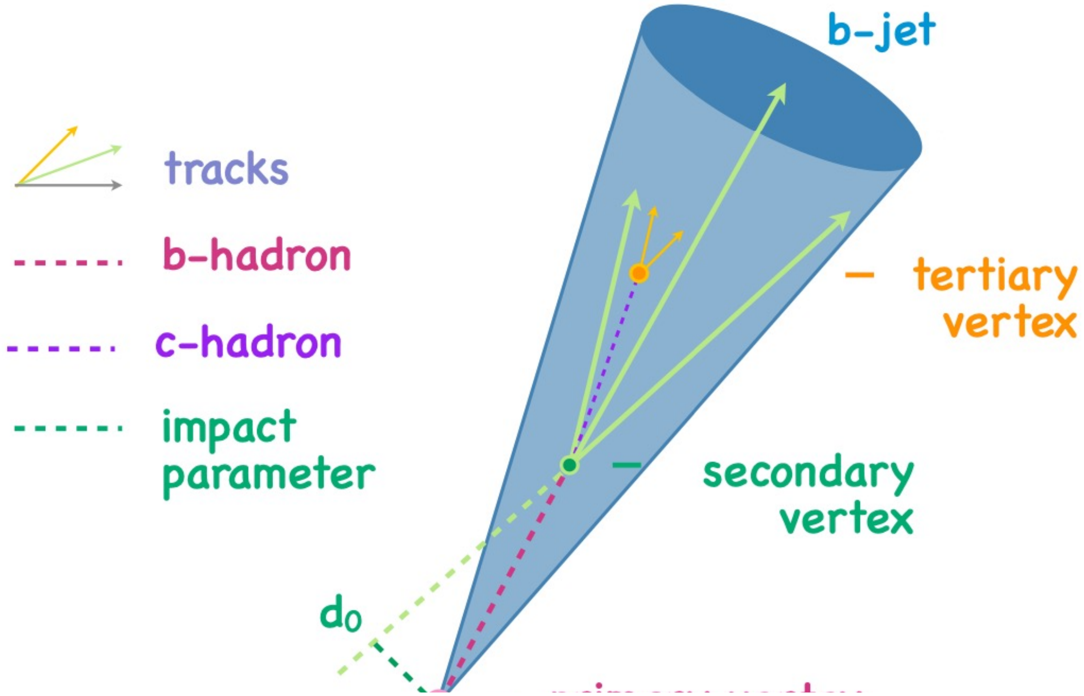
B and C hadron features

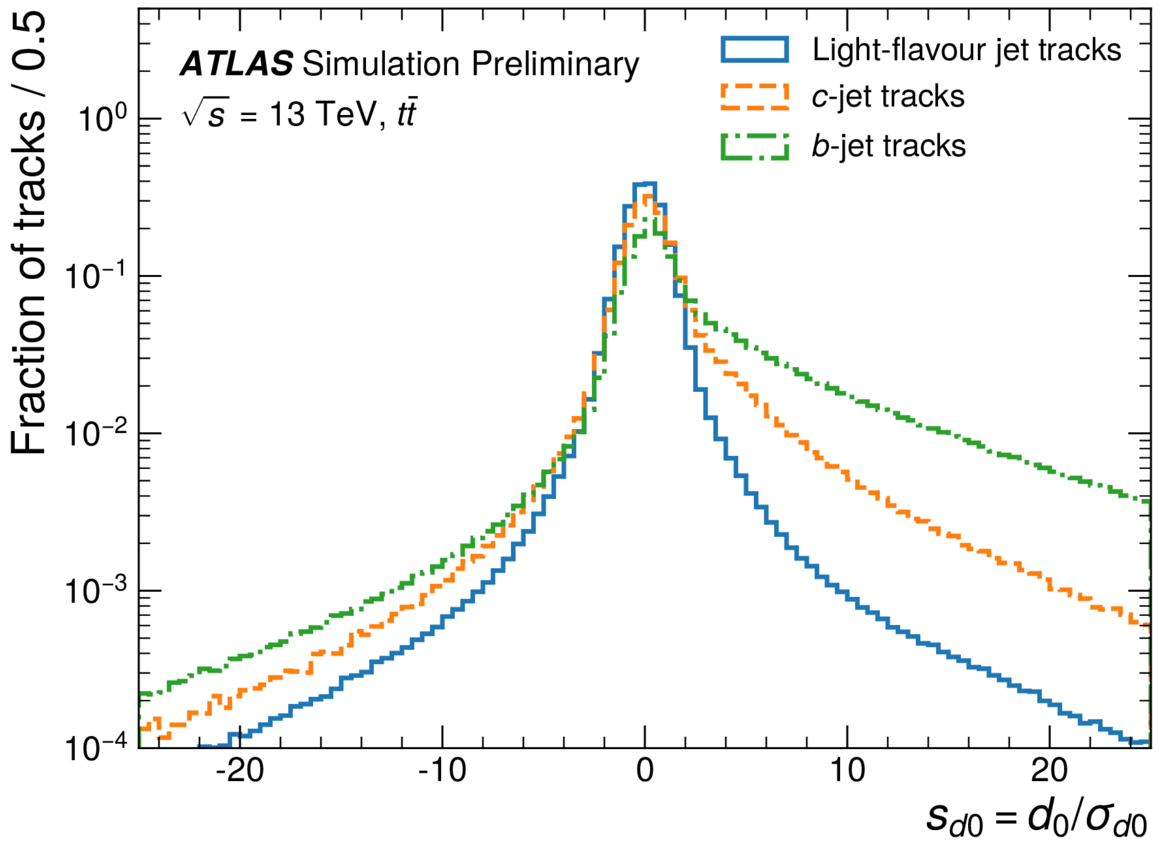
- Long lifetime
- High mass
- High decay product multiplicity
- B hadron often decays to c-hadron

- What we measure in the detector
  – Reconstruct tracks (from hits)
  – Extrapolate tracks to vertices



Track feature: signed IP significance





Interpret this now as probability density functions p_b, p_c, p_l

https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-PHYS-PUB-2020-014/

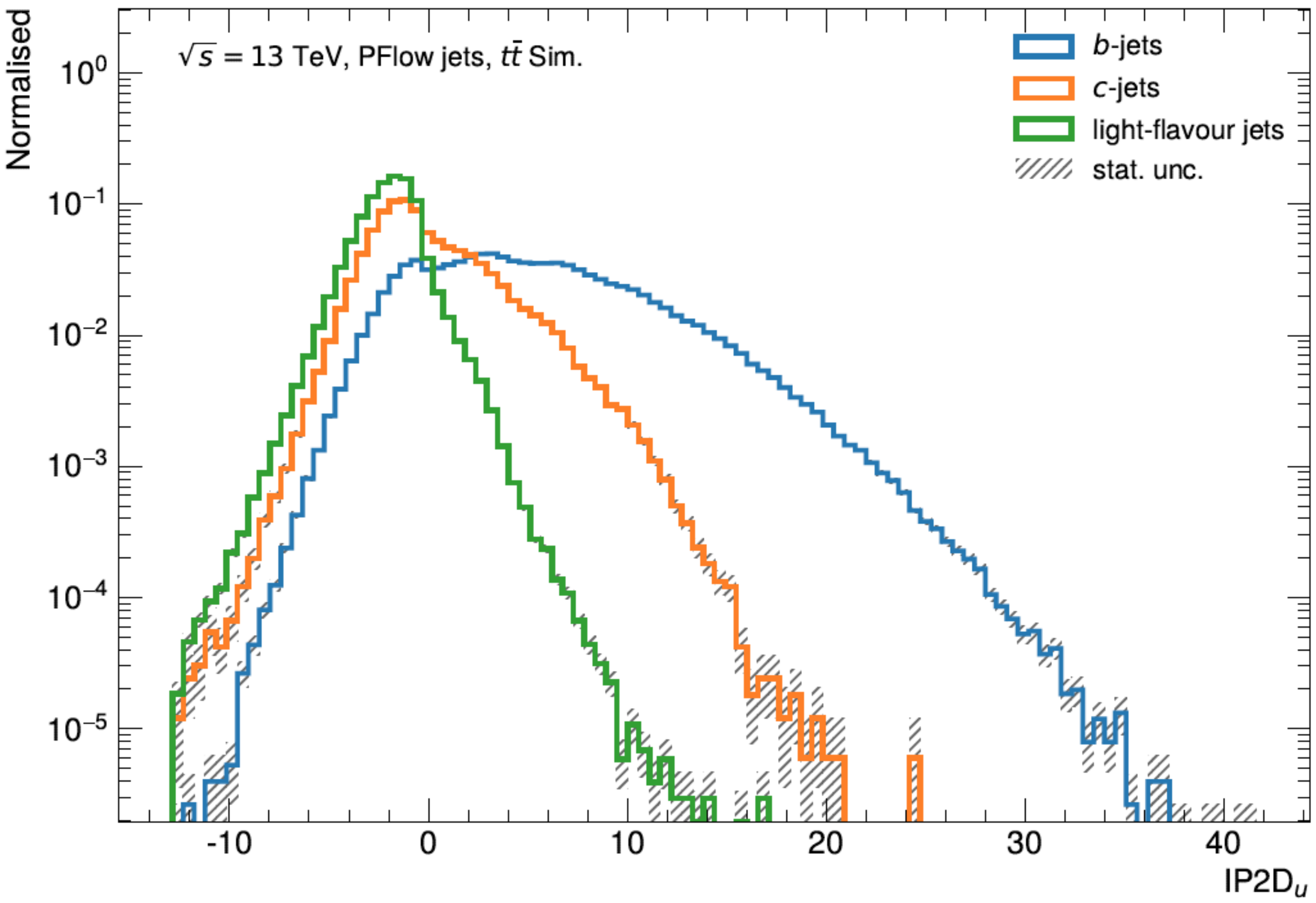
Hand-designed jet feature: IP2D

- Neyman–Pearson lemma:
  - Log-likelihood-ratio (LLR) test has highest power to distinguish competing hypotheses

$$\mathrm{IPxD}_{l,c,cl} = \sum_{i \in \mathrm{tracks}} \log\left(\frac{p^i_{b,b,c}}{p^i_{l,c,l}}\right)$$

Can we really just sum probabilities?

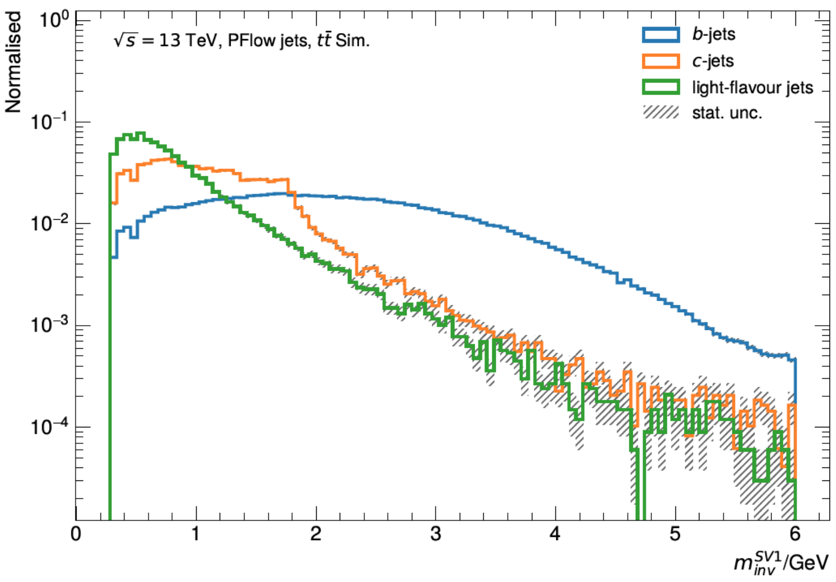
Assumption
independent and identically distributed (i.i.d.) !!!

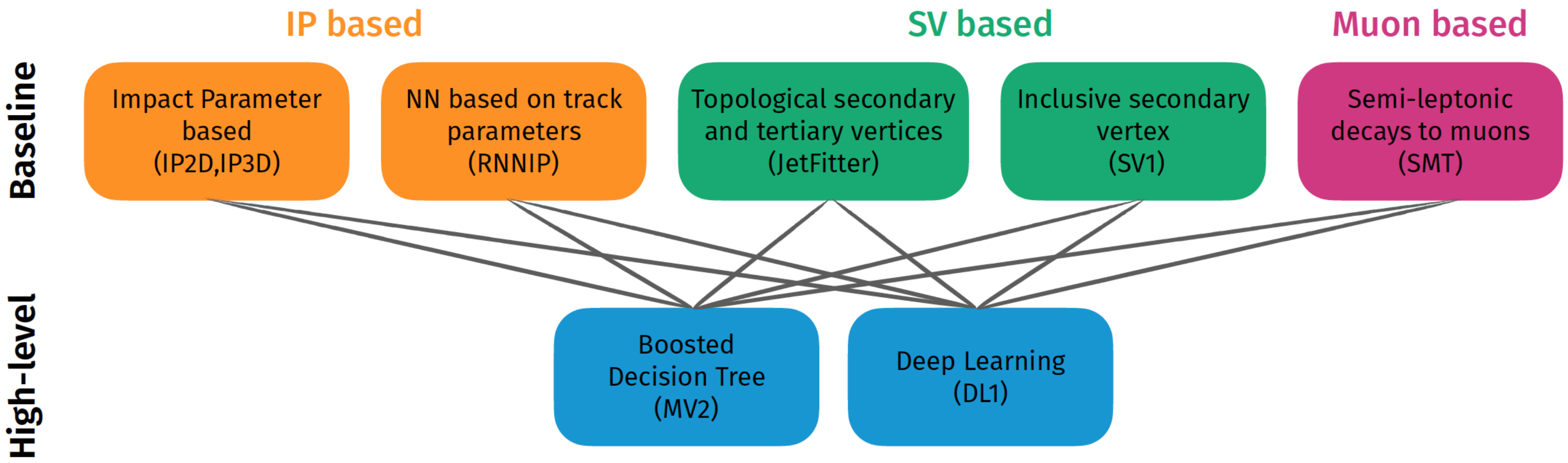


https://cds.cern.ch/record/2765038?ln=en

Putting it all together

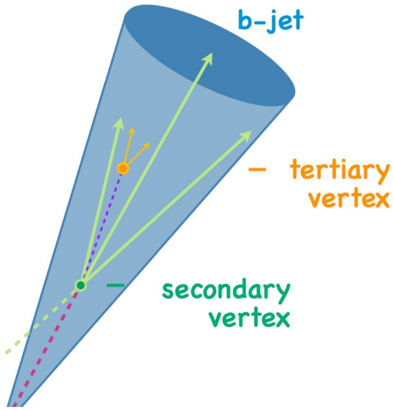
Secondary vertex (SV) reconstruction & many other *feature extractors*





https://cds.cern.ch/record/2765038?ln=en

Limitations of feedforward NNs



- FF NNs need a **fixed-size** number of **ordered** inputs

- The flavor-tagging input space consists of
  - *Hit reconstruction:* **variable** number of measured 3D space points
  - *Track reconstruction:* combine points to **variable** number of tracks per jet
  - *Vertex finding*: extrapolate tracks to **variable** number of vertices per jet

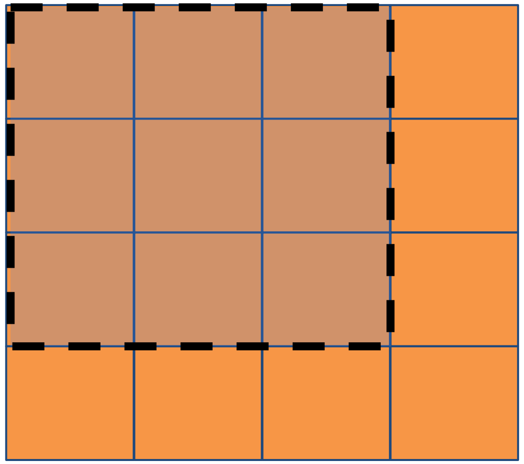
- Ad-hoc workaround:
  - **Fixed-size**: zero-pad/truncate variable-size
  - **Ordered**: leading N tracks
- NOT ideal – why?

The kind of inputs: structured data

Flat inputs

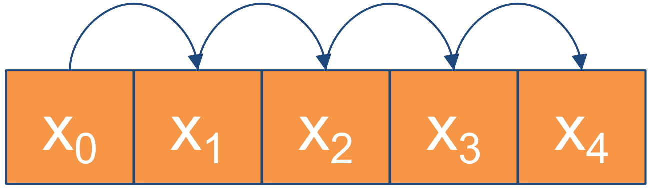
- Inputs are independent
- Each block is a **different** variable
- Fixed-size input
- **Fully connected layers**

Image-like inputs



- Inputs have regular spatial separation
- Each block is the **same** “variable”
- **Convolutional networks**

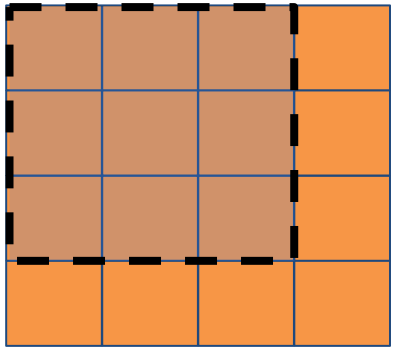
Time series



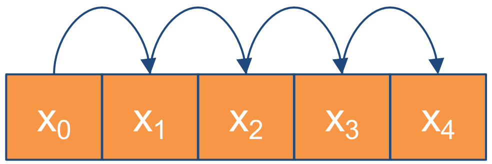
- Inputs come in a sequence
- Each block is the **same** “variable”
- Logical order with dependence on what comes before/after
- **Recurrent networks**

But what about unordered data?

Arbitrary order

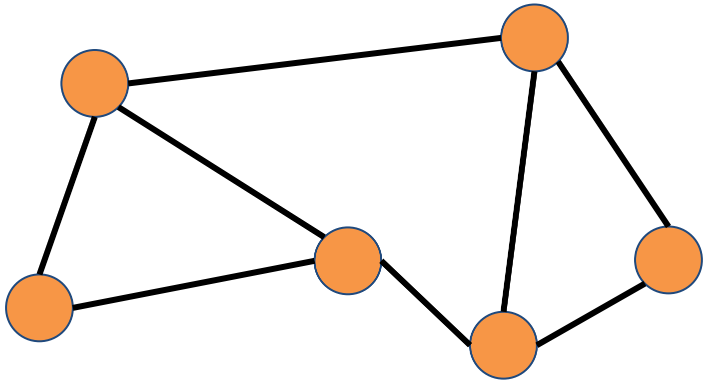


Forced structure/order



Forced order

Graph Networks

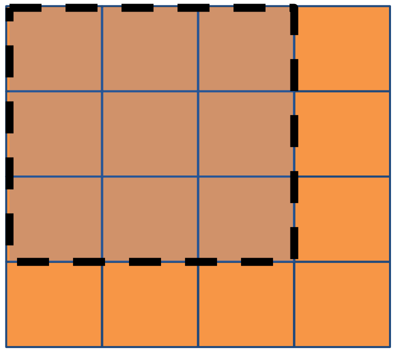


- Operate on nodes and edges
- Update nodes & edges based on connections
- **Permutation invariant: no order enforced**
- **Variable-size input**

Tracks for flavor-tagging

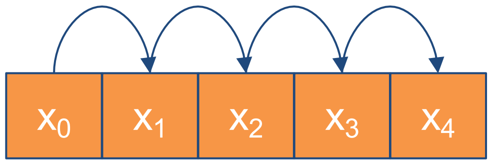
Independent inputs? **No!**

~~Fully connected layers~~



Regular spatial separation? **No!**

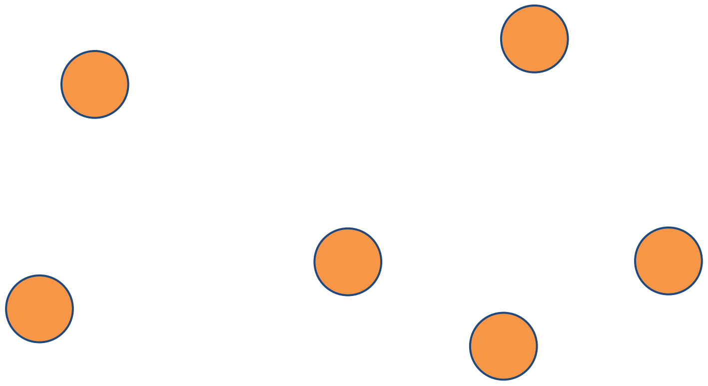
~~Convolutional networks~~



Ordered data? **No!**

~~Recurrent networks~~

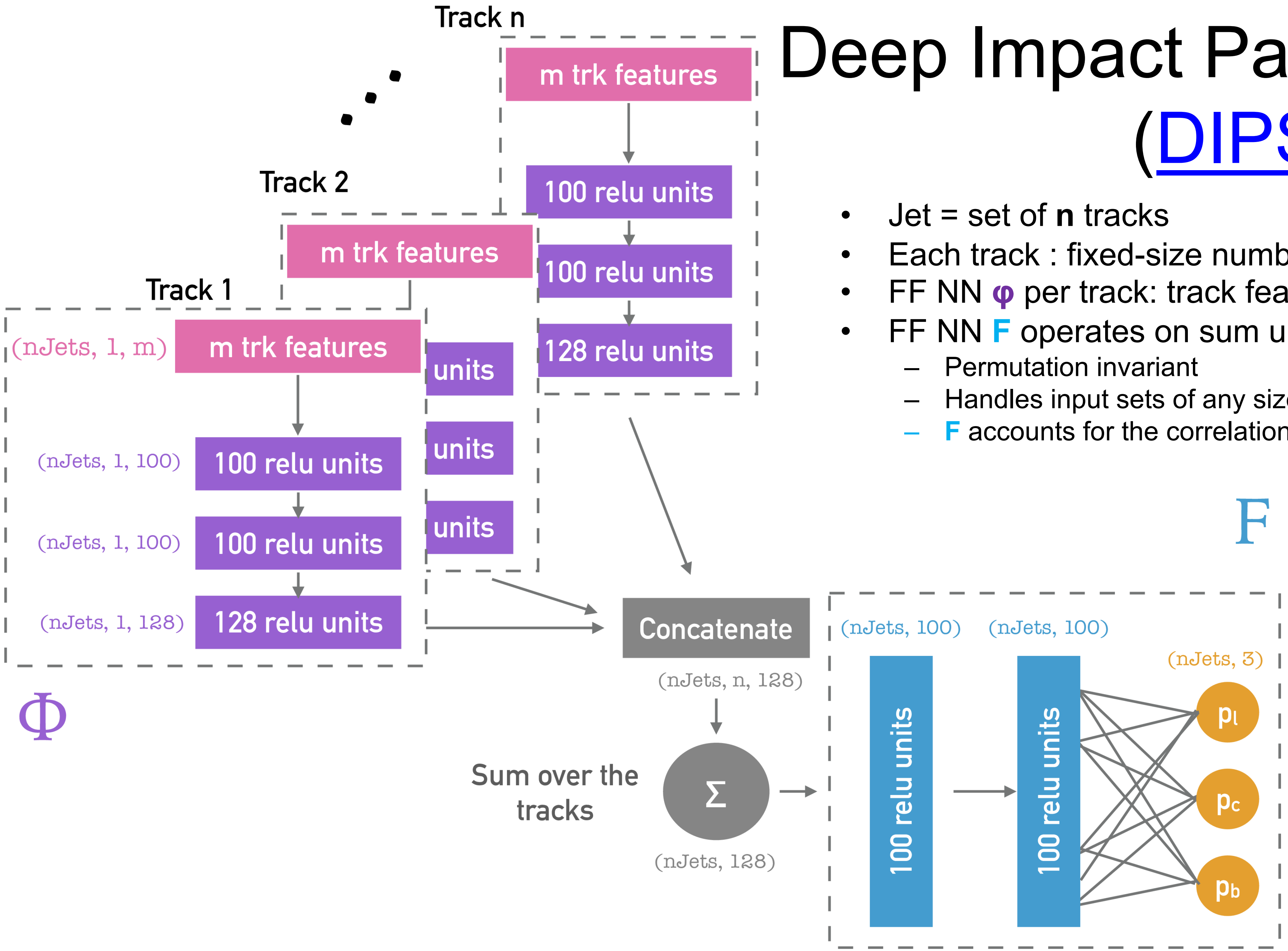
Deep Sets: nodes without edges



- Any input size
- Output invariant to order of inputs
- Same operation **φ** to each node
- Apply pooling **ρ** to output

Deep Impact Parameter Sets (DIPS)

- Jet = set of **n** tracks
- Each track : fixed-size number of features
- FF NN **φ** per track: track features → latent space
- FF NN **F** operates on sum up all tracks
  - Permutation invariant
  - Handles input sets of any size (any number **n** of tracks)
  - **F** accounts for the correlations between the tracks



Probability for jet to be b, c or light:

$$O(\{p_1, \ldots, p_n\}) = F\left(\sum_{i=1}^{n} \Phi(p_i)\right)$$

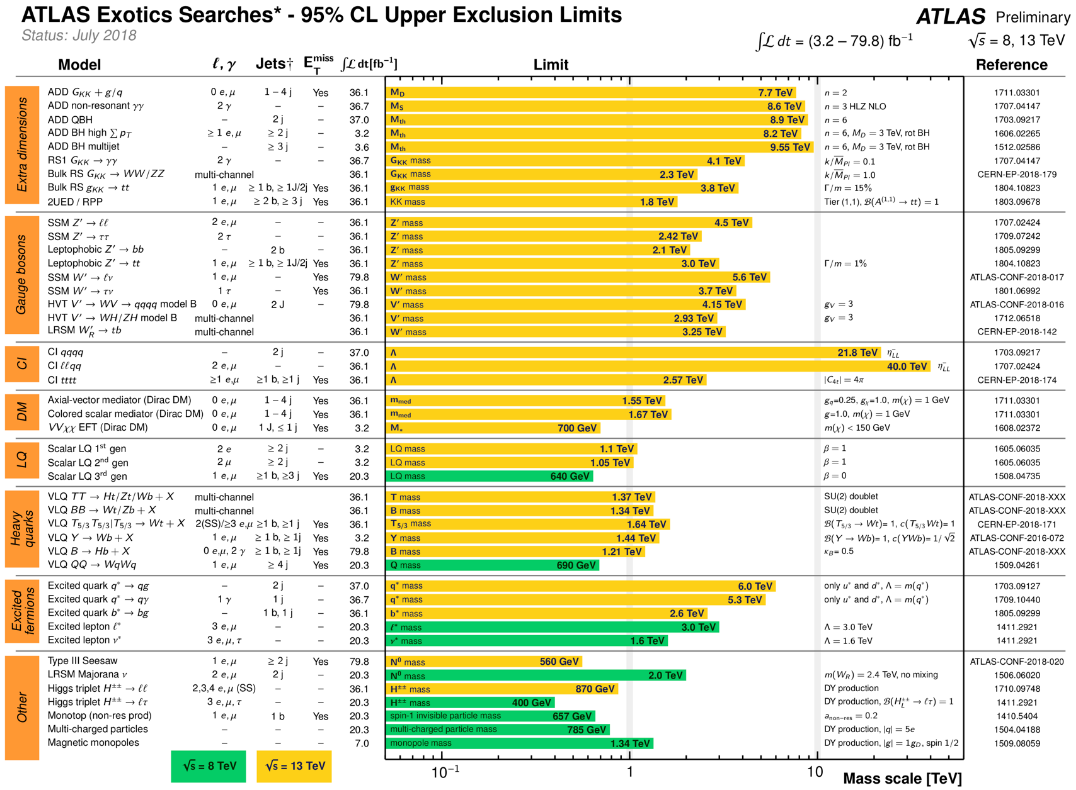
p_i = track features for track i

Supervised++

- Substantial improvements for all physics objects
  - Boosted jet tagging, taus, e/gamma,… also regression
  - Flexible multi-classification
- Apply same idea at event level for signal vs. background for given signal hypothesis
  - Inputs: high-level variables OR 4-vectors of objects
- I spare you long list of examples…

The *blemish*:

No sign of physics Beyond the SM



*Only a selection of the available mass limits on new states or phenomena is shown.
†Small-radius (large-radius) jets are denoted by the letter j (J).

- BSM physics not around the corner
- Current slow-growth era of the LHC: energy & luminosity
- Turning the crank?
  - Negligible increase in sensitivity for most of the search program
  - Signatures of new physics could be hiding in plain sight
  - **Hypothesis: we just have not looked in the right place yet**

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

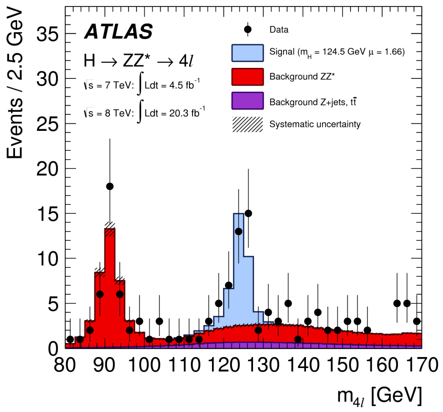
Assumptions

- **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

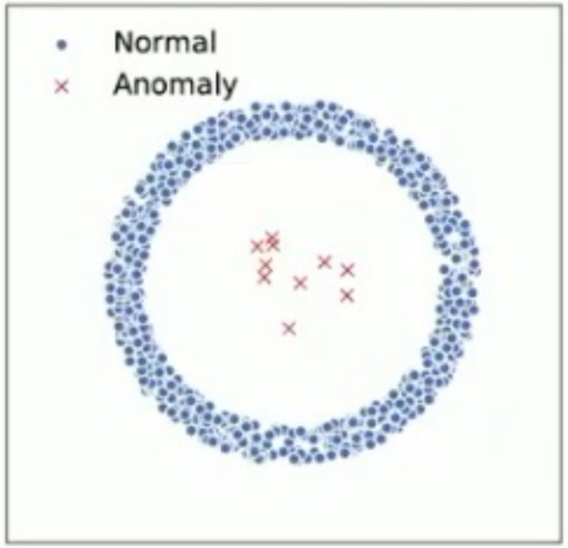
- Grain of sand ≙ LHC data collision
- What is an **outlier**
- What is an **inlier / over-density**

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]

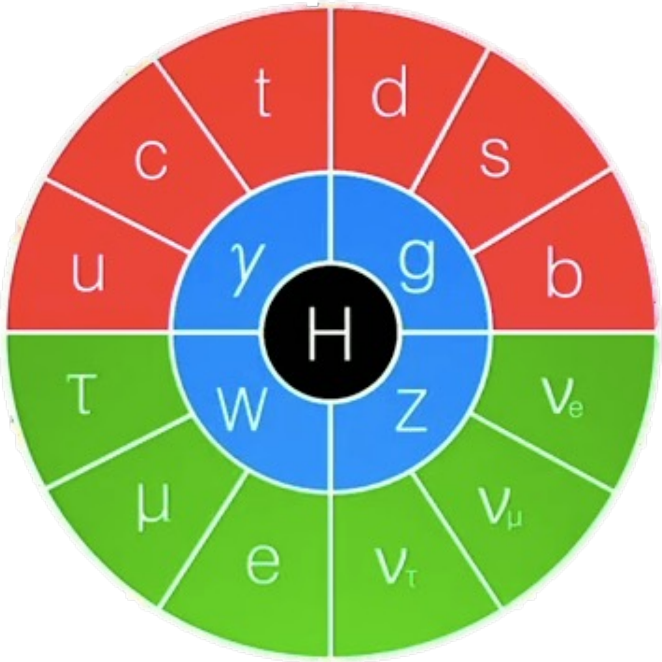
- Individual examples **not** anomalous
- **Anomalous collective behaviour**

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

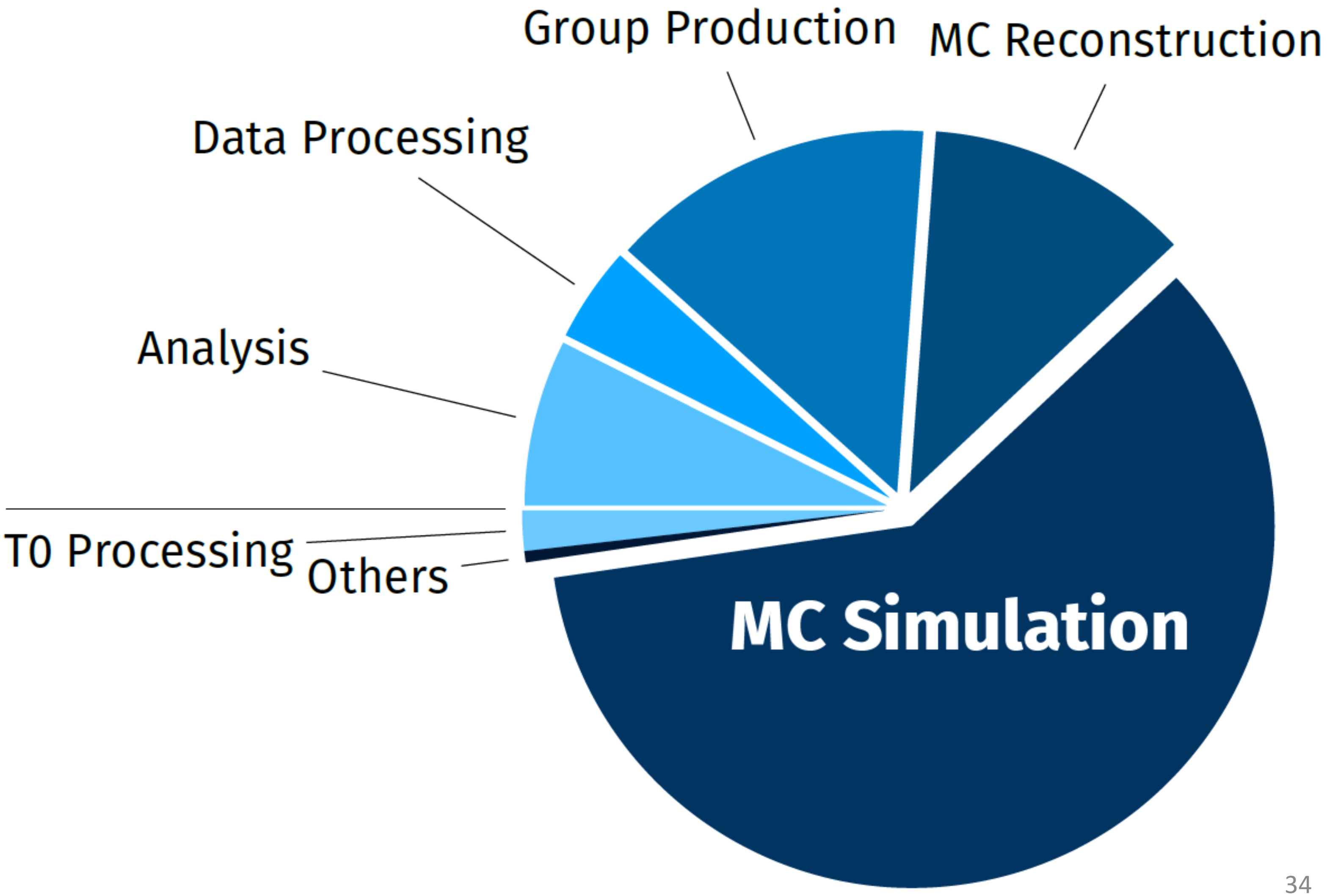


- Model of our SM events

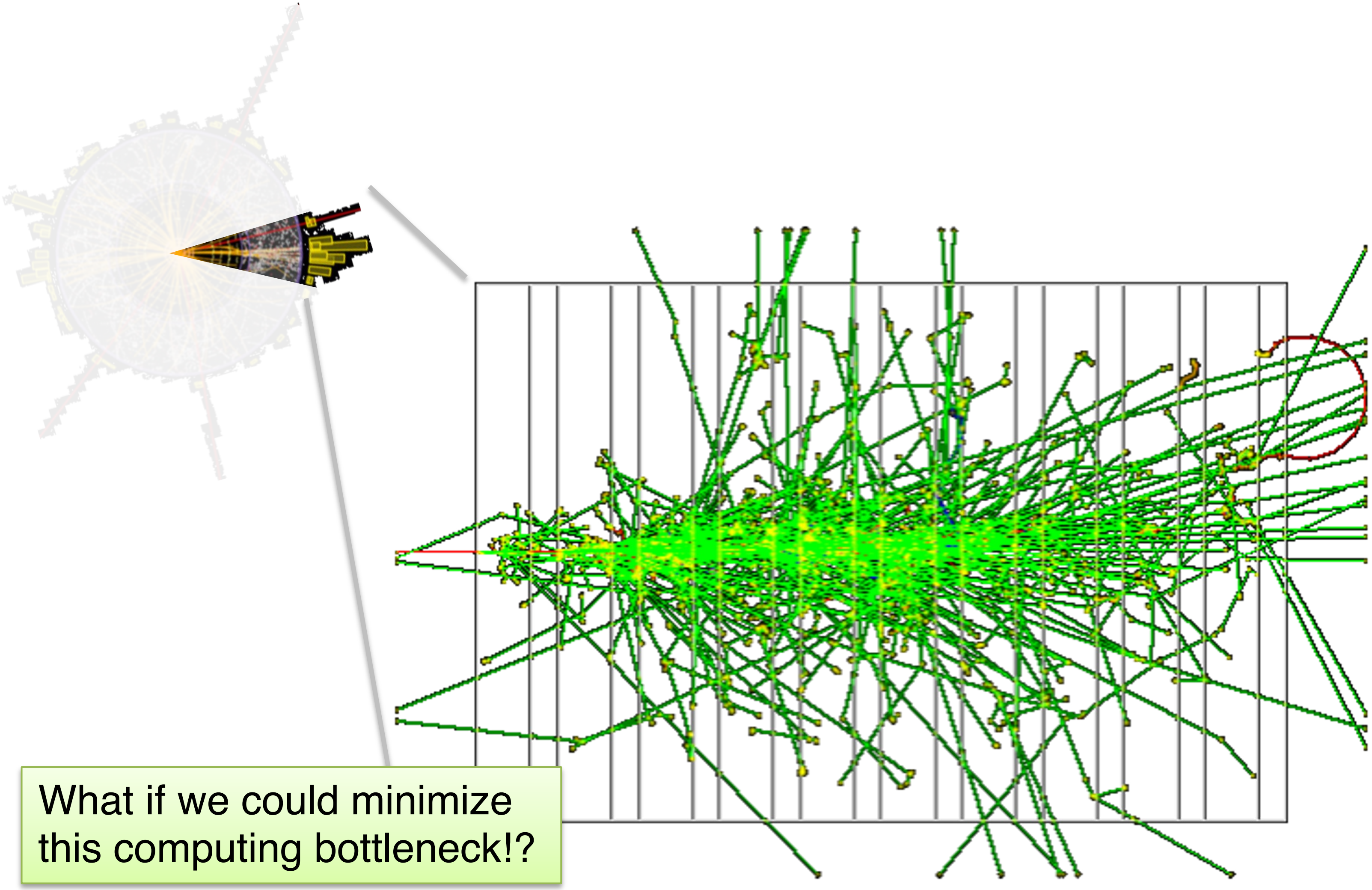
Forward Monte Carlo modeling

Computing bottleneck:
Monte Carlo simulation

Why?



One particle entering the calorimeter…

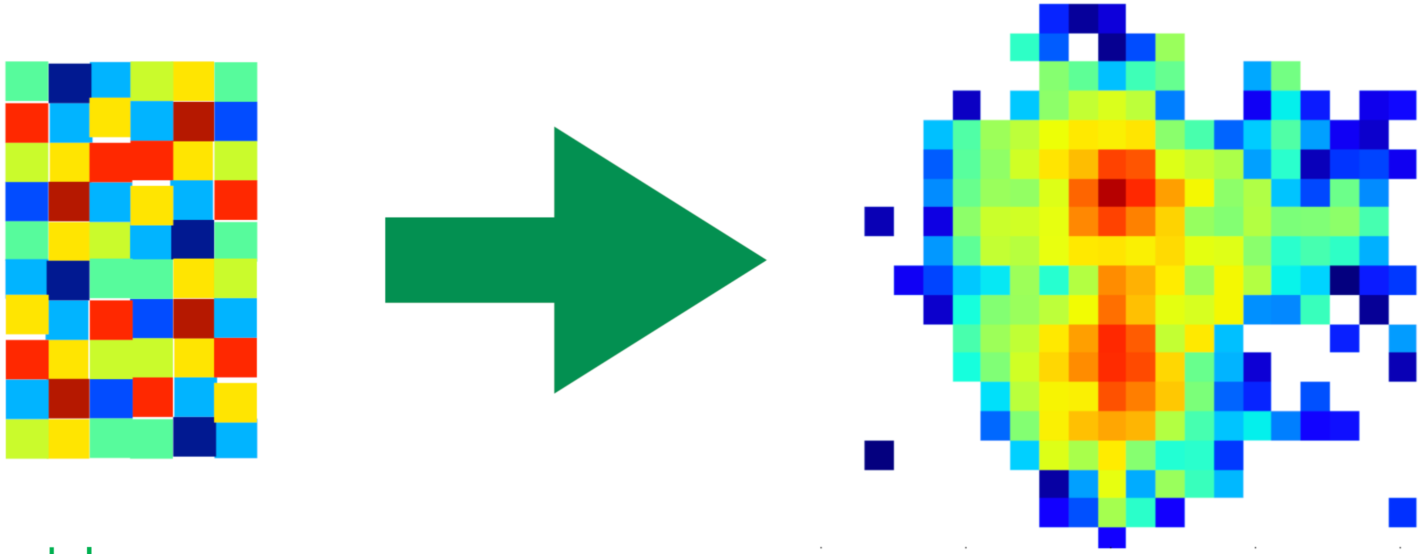


What if we could minimize this computing bottleneck!?

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

Toolbox: generative modeling

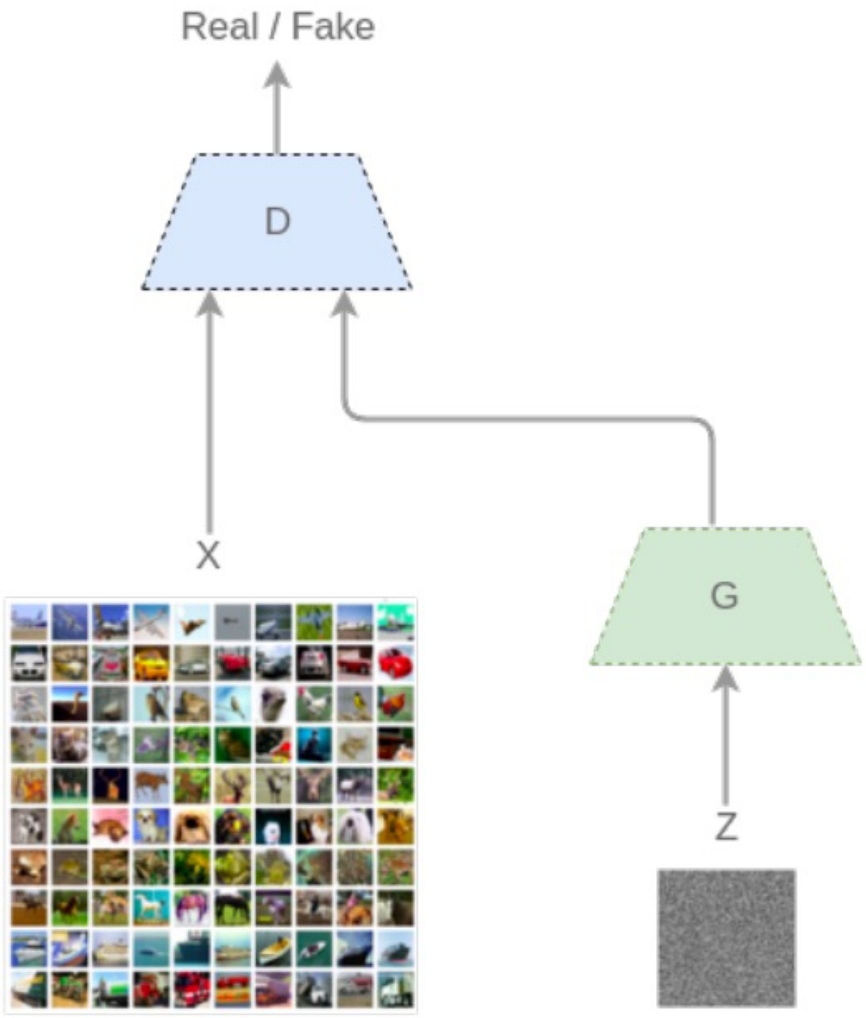
Build a generator* which maps random numbers to structure



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

*Deep generative NN model:
- Generative Adversarial Network (GANs)
- Normalizing Flows (NFs)
- **Variational Autoencoders (VAEs)**

Toolbox: GAN

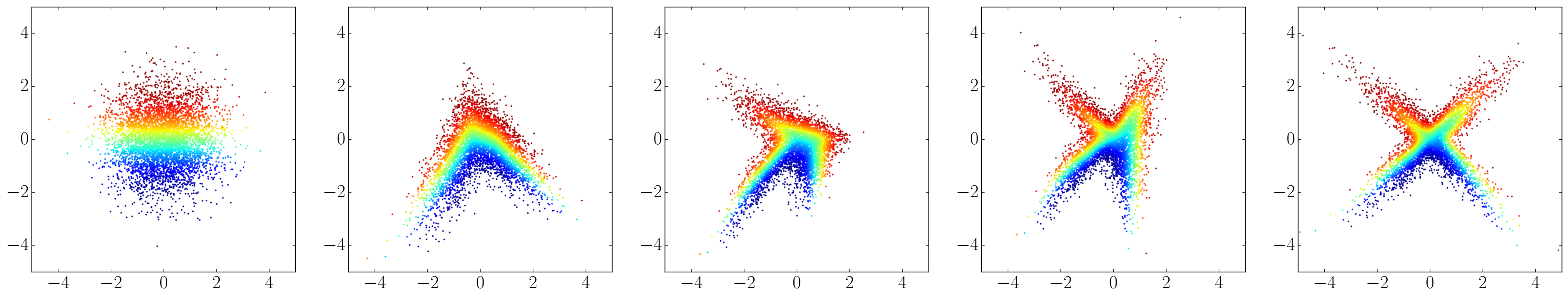


- Generative Adversarial Network

- Two-network game
  - Generator **G** maps noise to structure
  - Discriminator **D** tries to classify images as real or fake
  - When **D** is maximally confused, **G** will be a good generator

Toolbox: normalizing flows (NFs)

- Series of simple invertible transformations to map simple (Gaussian) distribution p(z) to complex data distribution $p_\theta(x)$
  - *Variable transformation*: z → x $\quad p_\theta(x) = p(z)|\det(J_x^{f_\theta})|$
  - Function f_θ parameterized by NN
  - Matching target p_θ (x) by maximizing likelihood $\quad \max_\theta \mathbb{E}_{p_D(x)}[p_\theta(x)]$



[NF cont'd]

- Applications:
  - Importance sampling for Monte Carlo generation by learning weights to model cross-sections
  - Calibration of *synthetic* data to *real* data
  - Calibration of *fast* simulation to *full* simulation

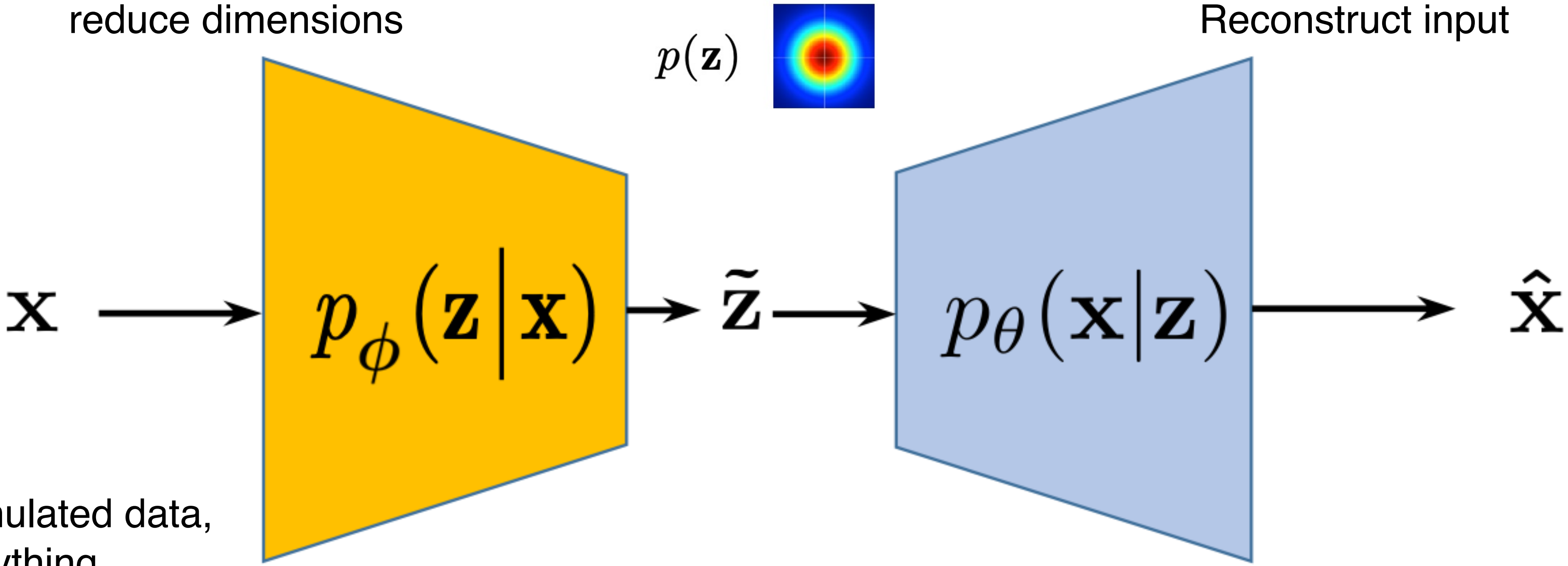
- Limitation:
  - Dimension preserving
    - Can overcome this…

Toolbox: Variational Autoencoder (VAE)

Probabilistic **encoder**:
reduce dimensions

Latent space (with given prior):
easy to sample from

Probabilistic **decoder**:
Reconstruct input

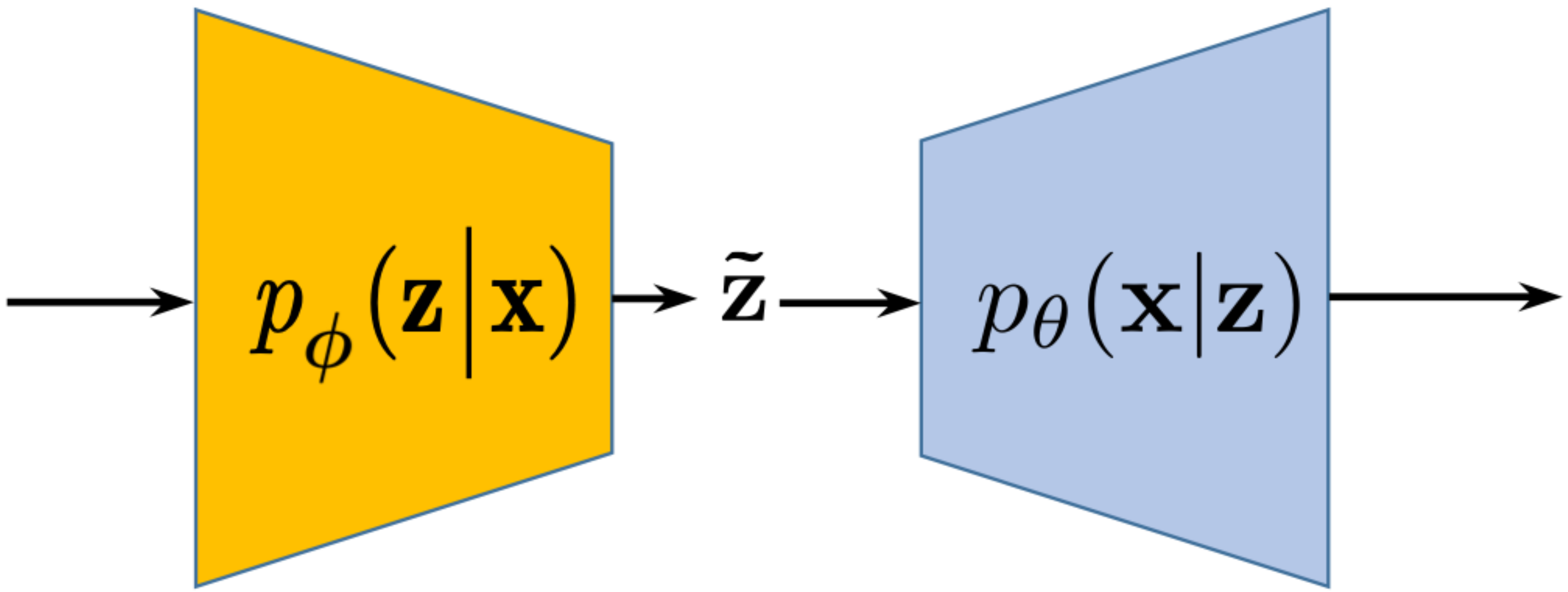


Input x:
Raw data, simulated data, features,…anything

Information bottleneck:
maximize encoded information

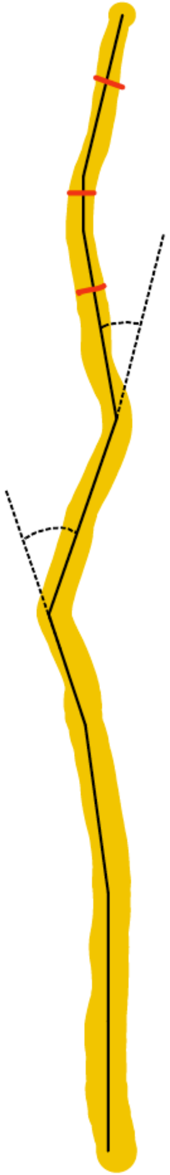
1. Reconstruction mode
2. Generation mode

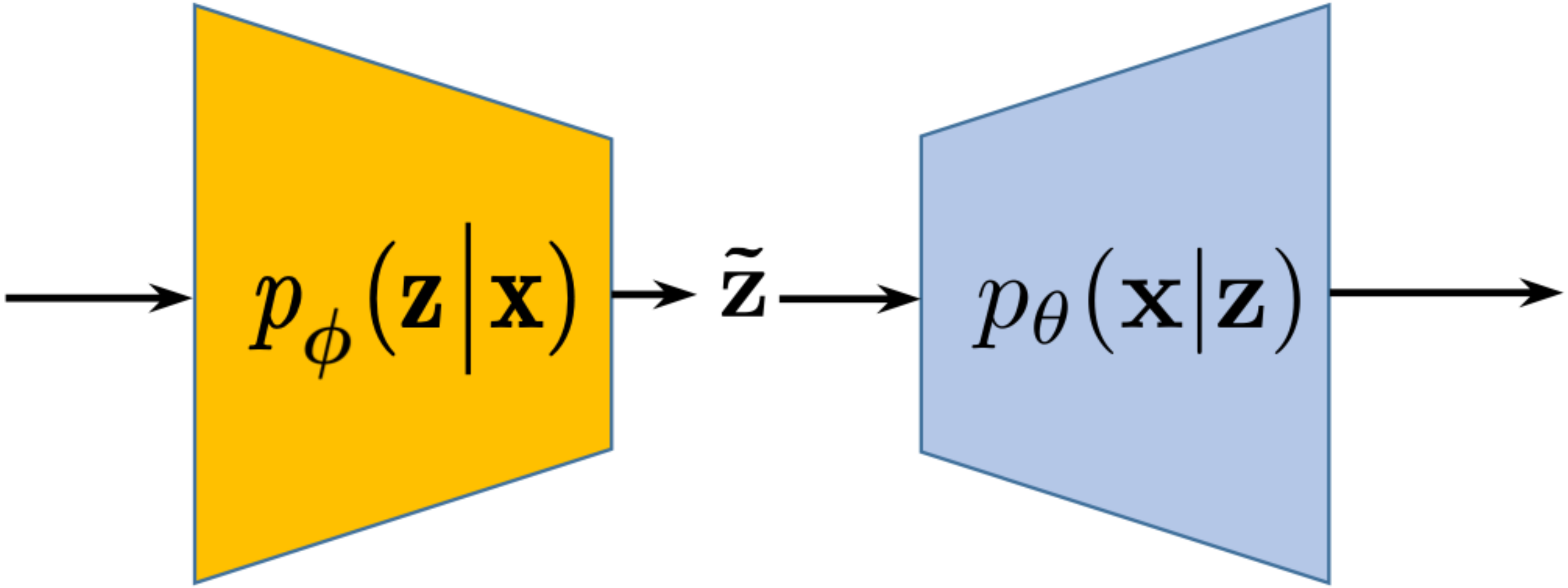
[Data volume reduction]

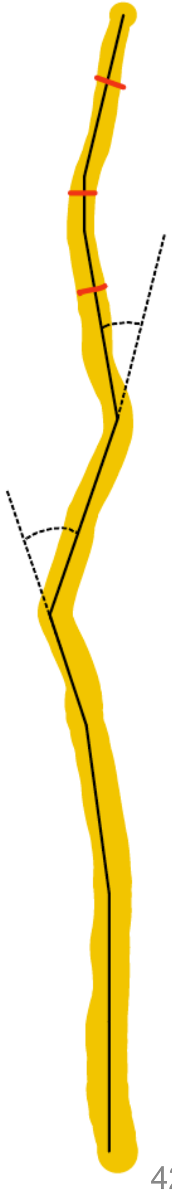


- Lossy compression with auto encoders
- Only maintain key features in data
- Example: reduce bandwidth to increase event rate

Reconstruction mode



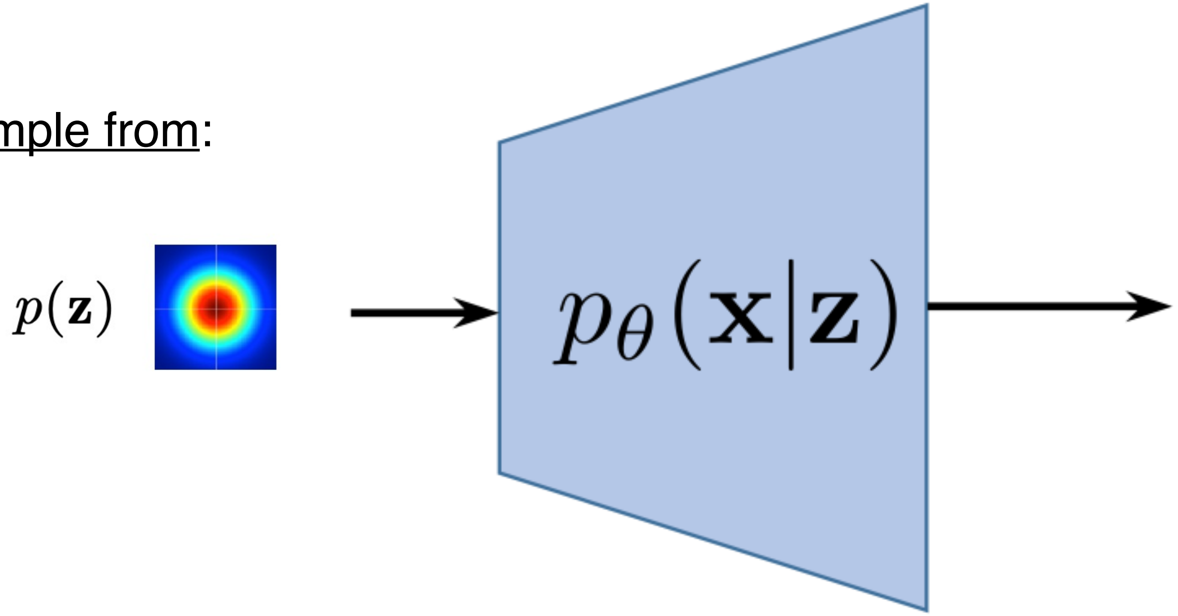


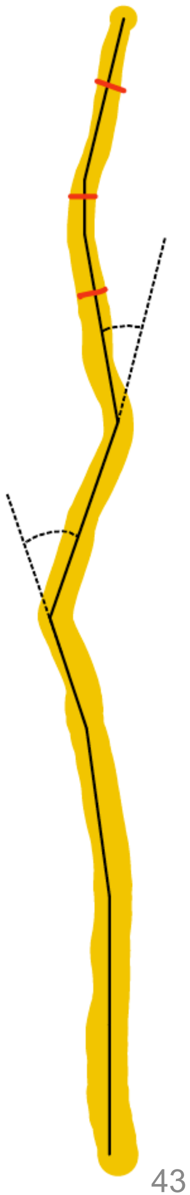


- Train on hay
- Apply to data: poor reconstruction for non-hay = anomaly

Generation mode

Sample from:

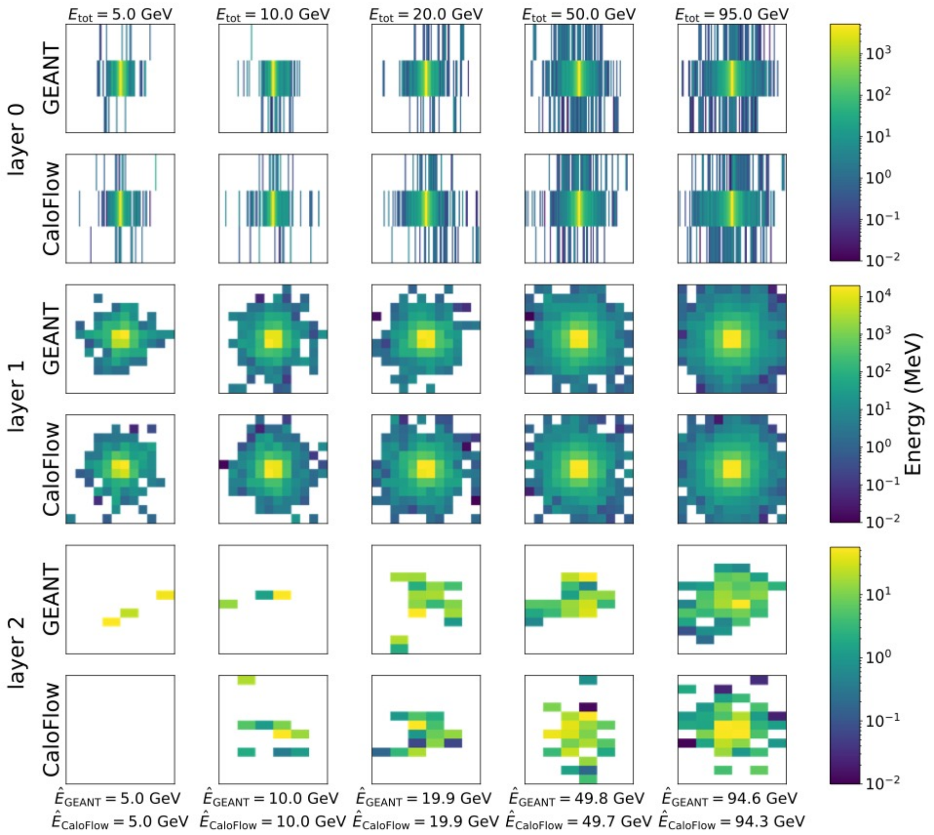




- Train on hay in reco mode
- **Rapidly** sample hay from a normal distribution

Can generate all sorts of things

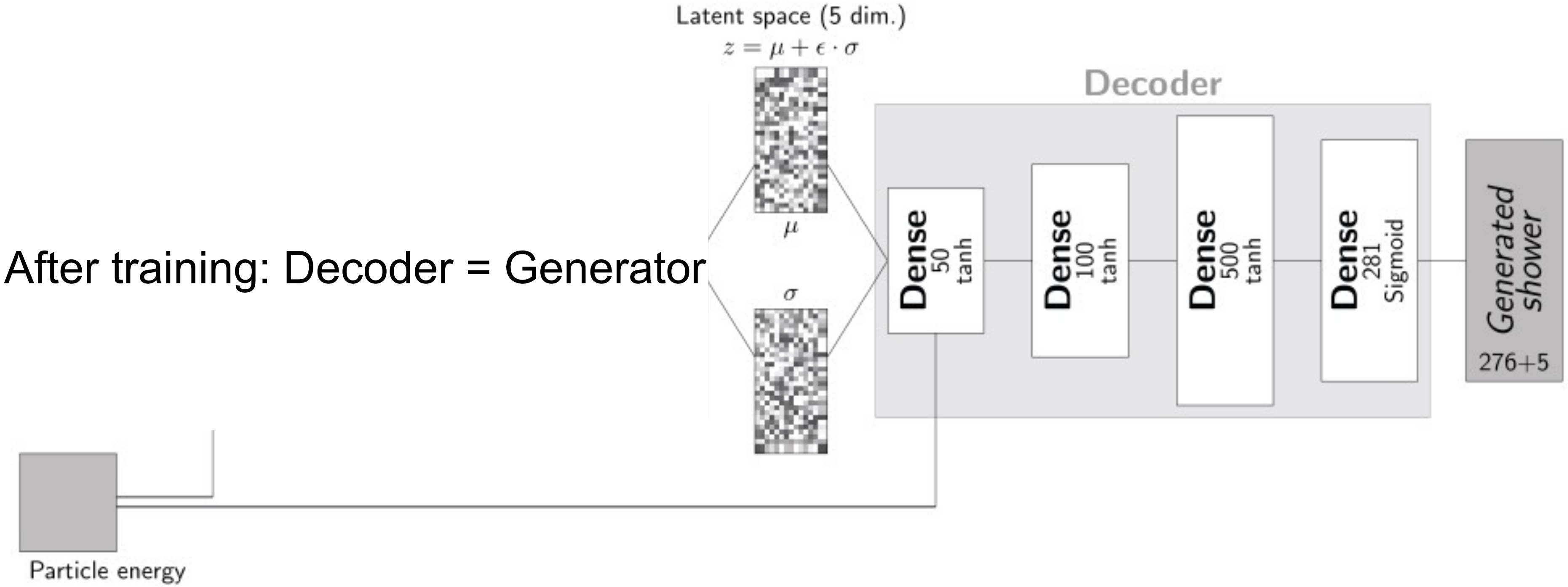
[Karras et al., 2018]



[CaloFlow]

VAE architecture

After training: Decoder = Generator



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

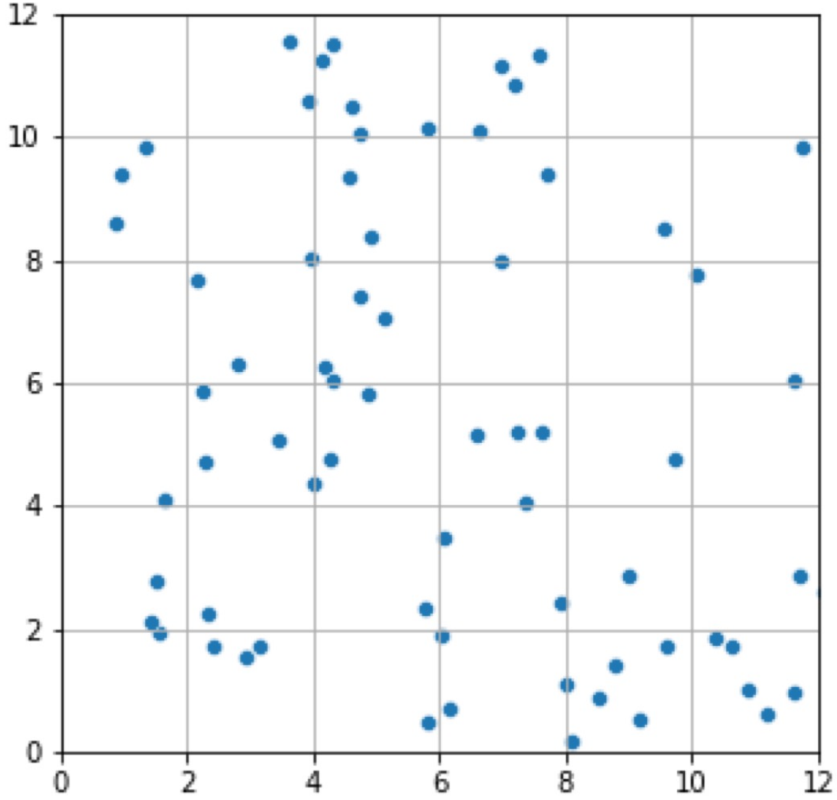
Validation: *marginals*

Generative modeling assessment

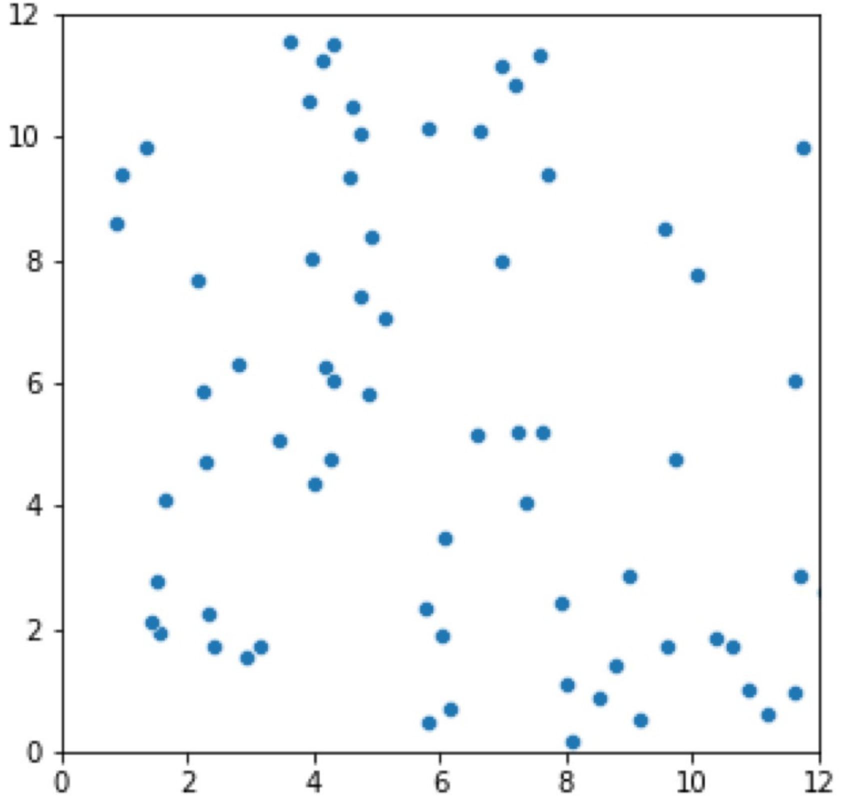
- Promising results but bottlenecks exist:
  - **Slow** development cycle
  - **Expensive & inflexible** training data (Geant4)
  - **Non-portable solution** highly dependent on detector geometry*
- Objectives:
  - Faster R&D
  - Decouple modeling from detector geometry → **point cloud format**

* 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

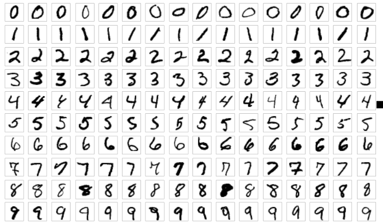
[source]

Sweet
spot
?

GEANT4
A SIMULATION TOOLKIT

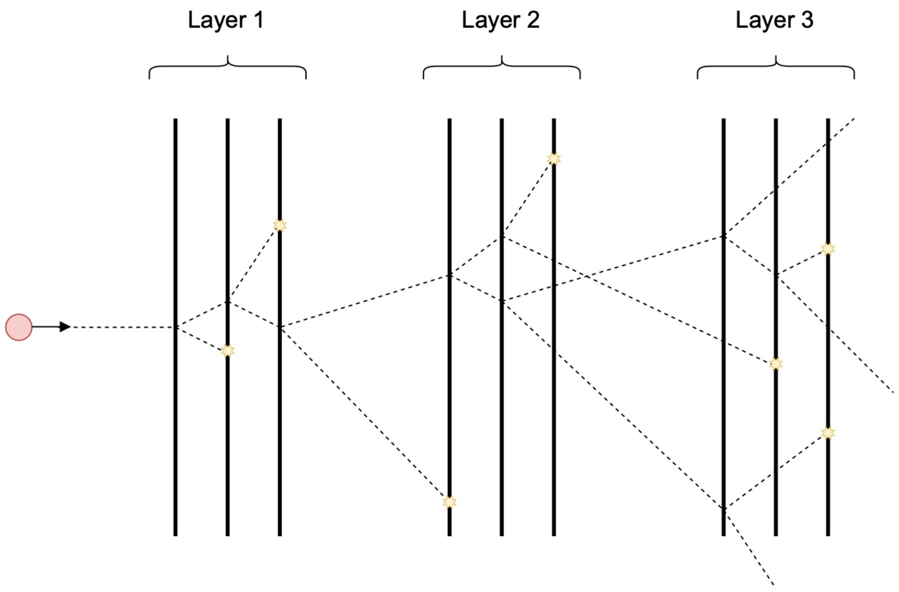
- Existing public point cloud data sets
  - Not a good proxy for physics data
  - Improvements don't *generalize*

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

The *MNIST*  for generative modeling?

- Can we design flexible & configurable proxy data sets?
  - – Diagnostics tool to develop new generative surrogate simulators
  - – Point-cloud format promotes *GNN-based* generative models

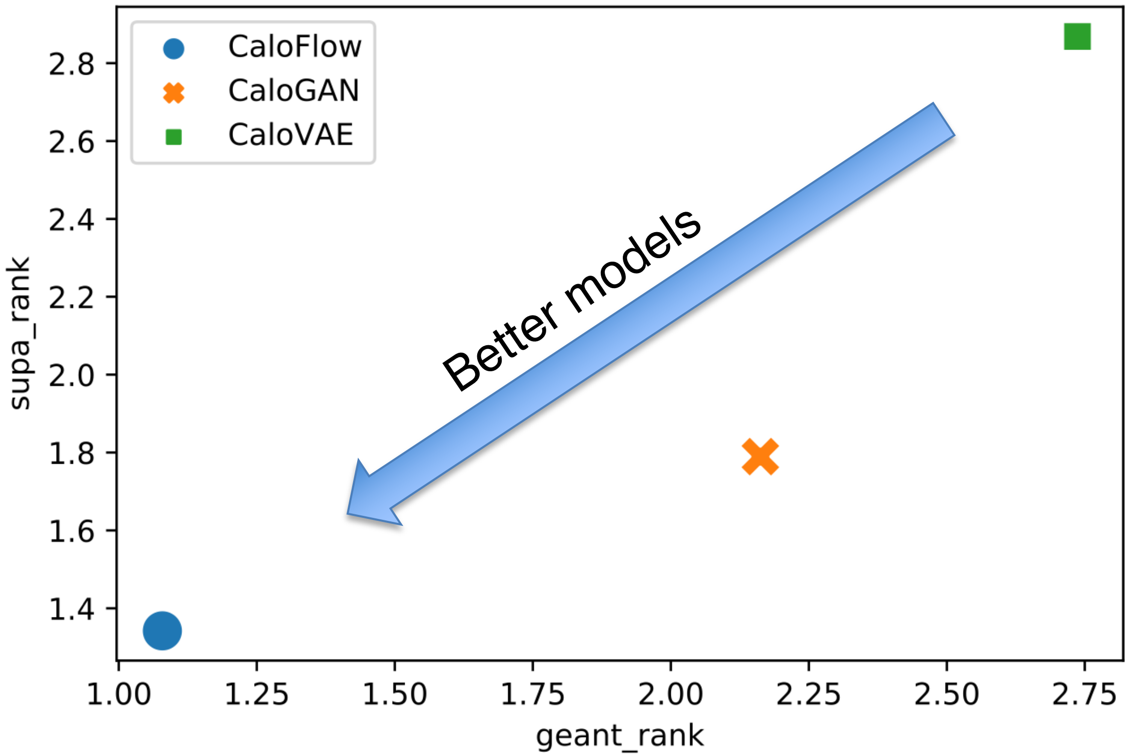
Simplified
- particle propagation,
- scattering &
- shower development



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

Need a simple model which is *realistic enough*

Show that proxy model tracks performance of Geant4 model

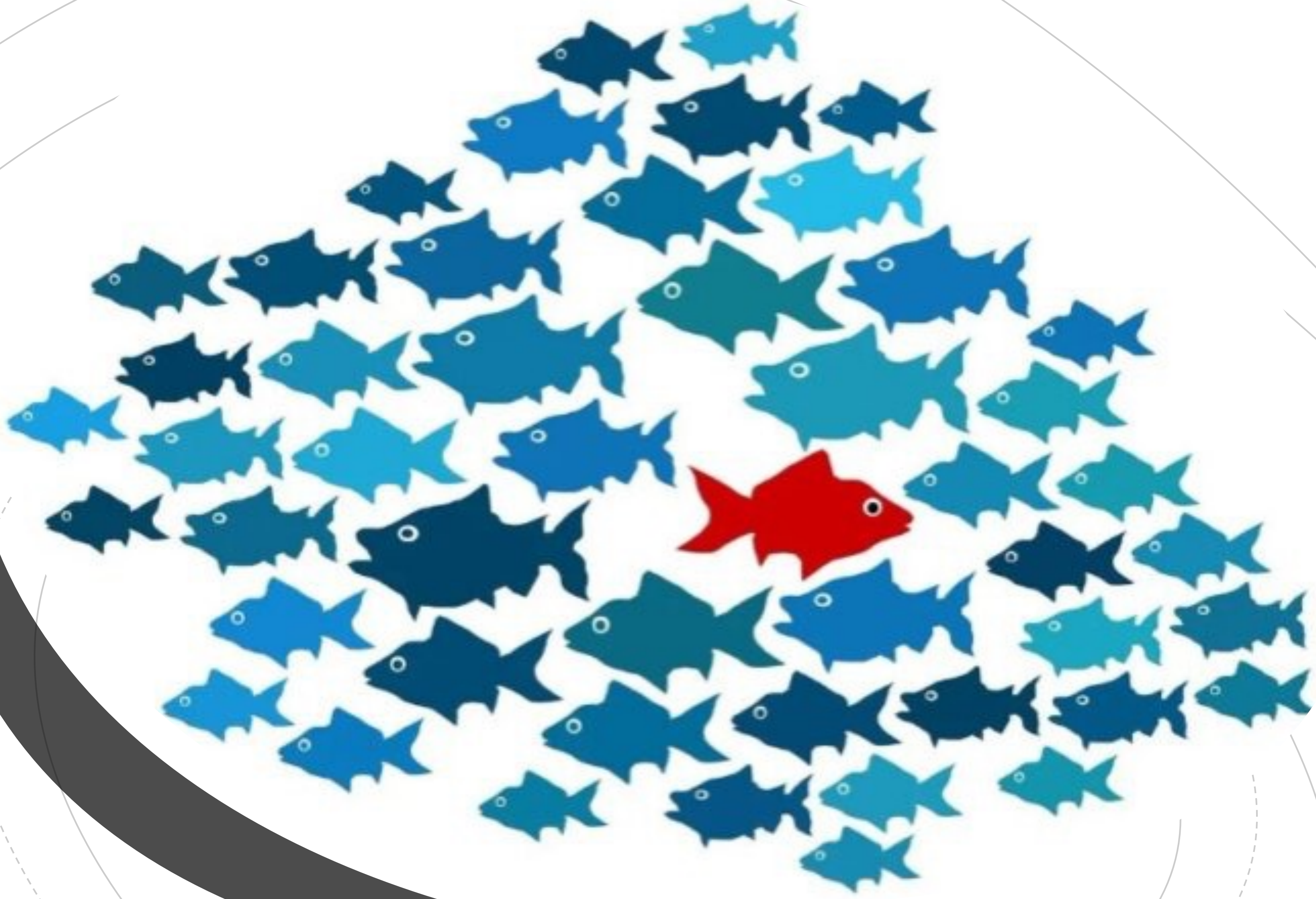


SUPA [SUrrogate PArticle propagation simulator]

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

Do model design on proxy data set:

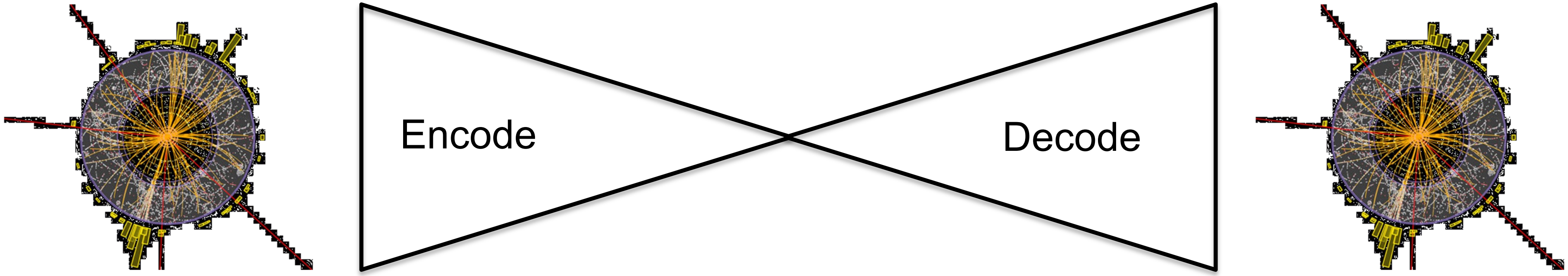
- Vary data complexity
- Optimize model
- Validation metrics



Outlier detection

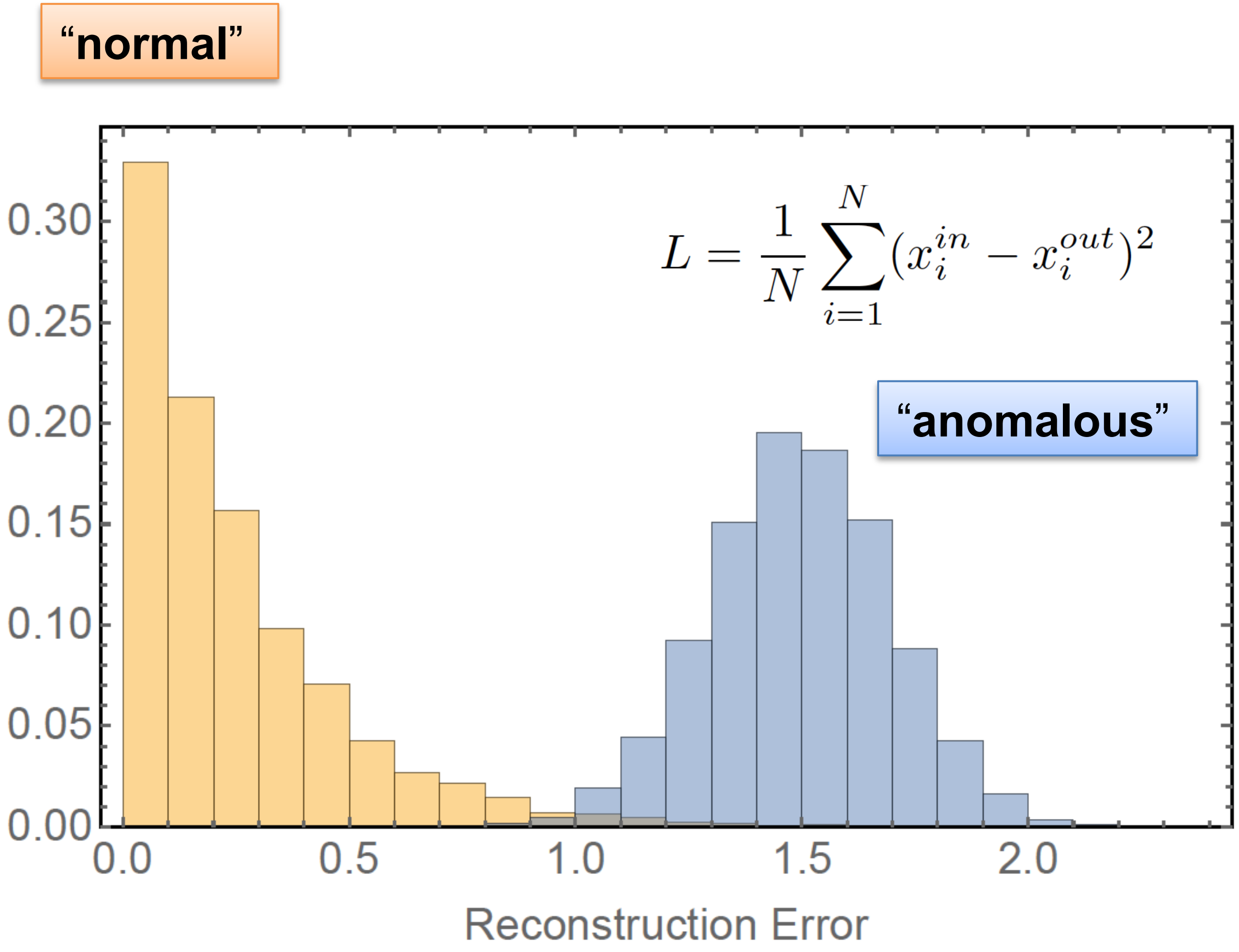
VAE in reconstruction mode: search for anomalous boosted objects

Encode and decode “normal” objects / events



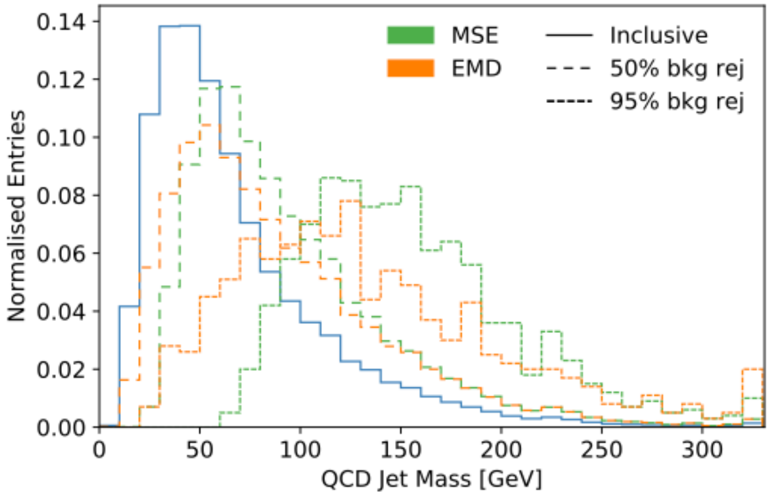
Compare original and reconstructed image

Anomalous jets



Challenge:

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



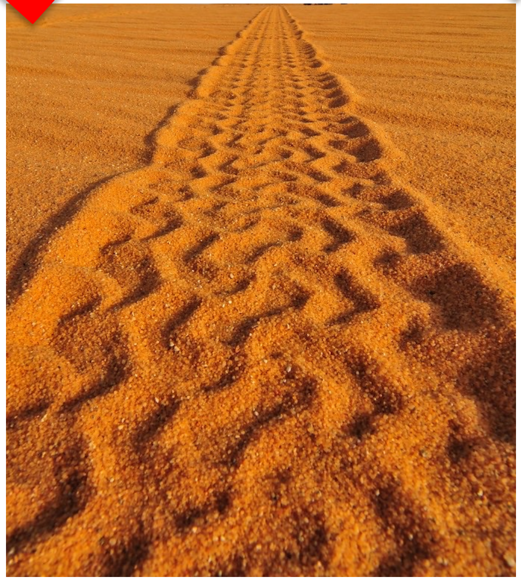
[https://ml4physicalsciences.github.io/2020/files/NeurIPS_ML4PS_2020_56.pdf]

[1709.01087, 1808.08979, 1808.08992, 1905.12651, 2007.01850]

The problem with outlier detection

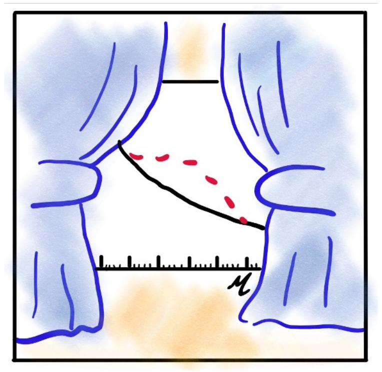
- Rarely *true* outliers in our data

- We look for an excess = over-density



Constructing Unobserved Regions by Transforming Adjacent INtervals

*All windows need **CURTAINs***



Data driven method for constructing

background templates with **arbitrary variables**

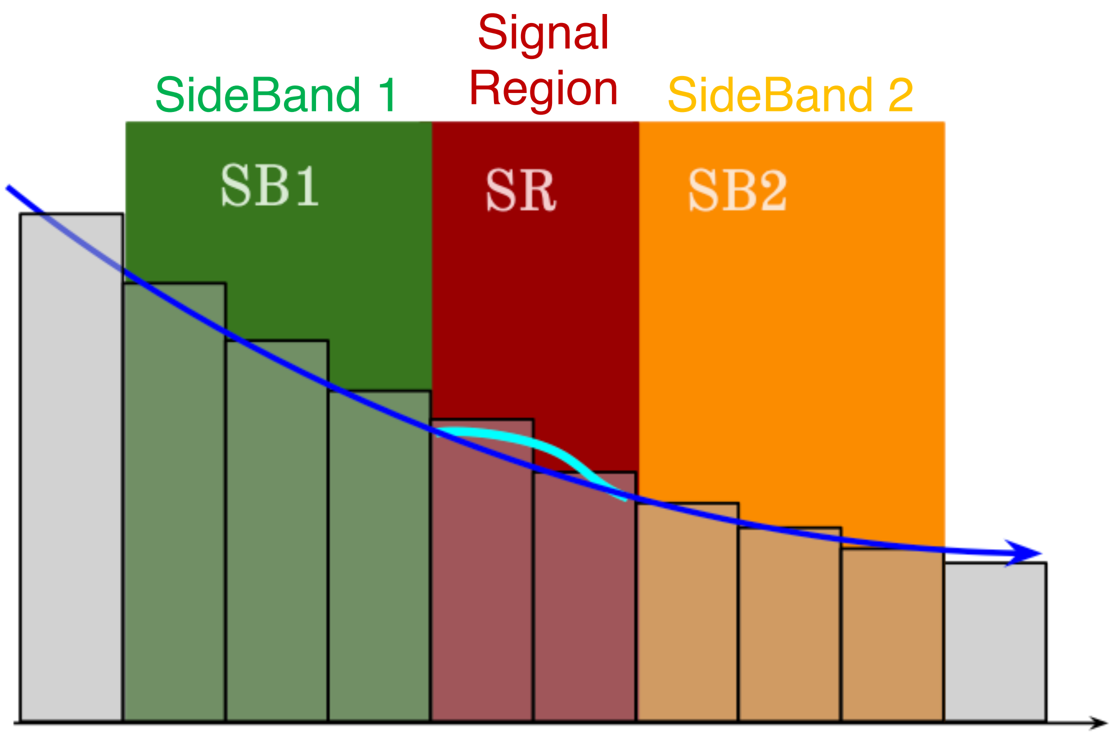
[https://arxiv.org/pdf/2203.09470.pdf]

Bump hunt

Focus on resonant signal = **bump**

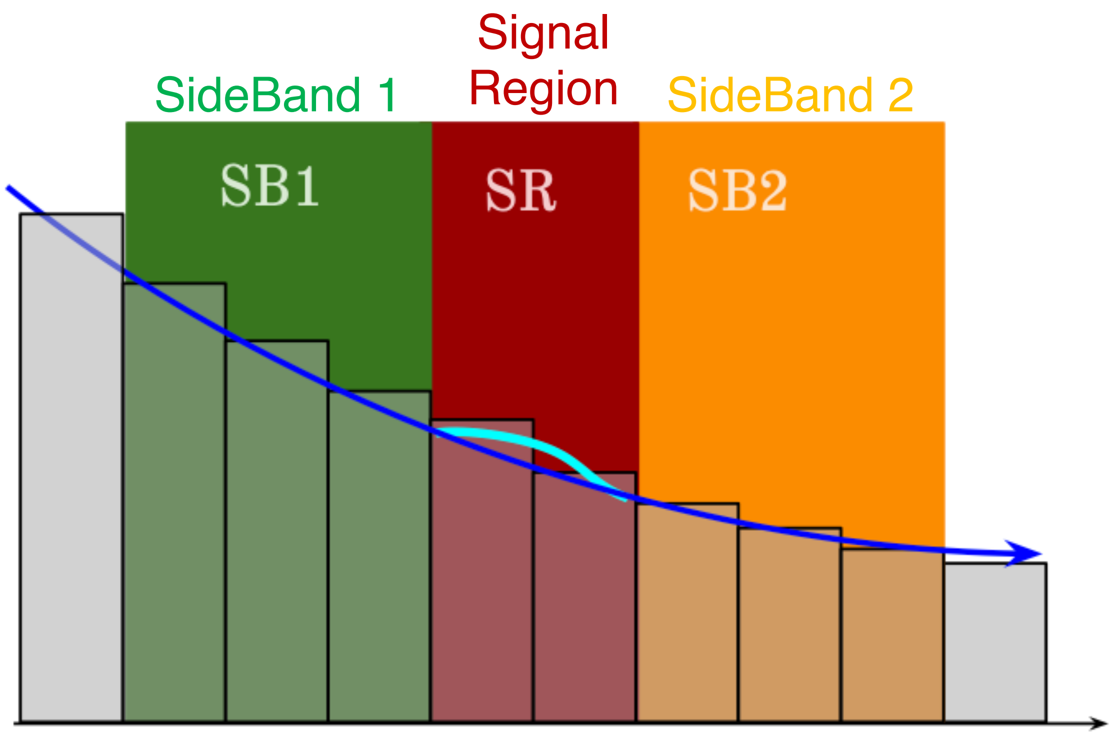
Method:

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

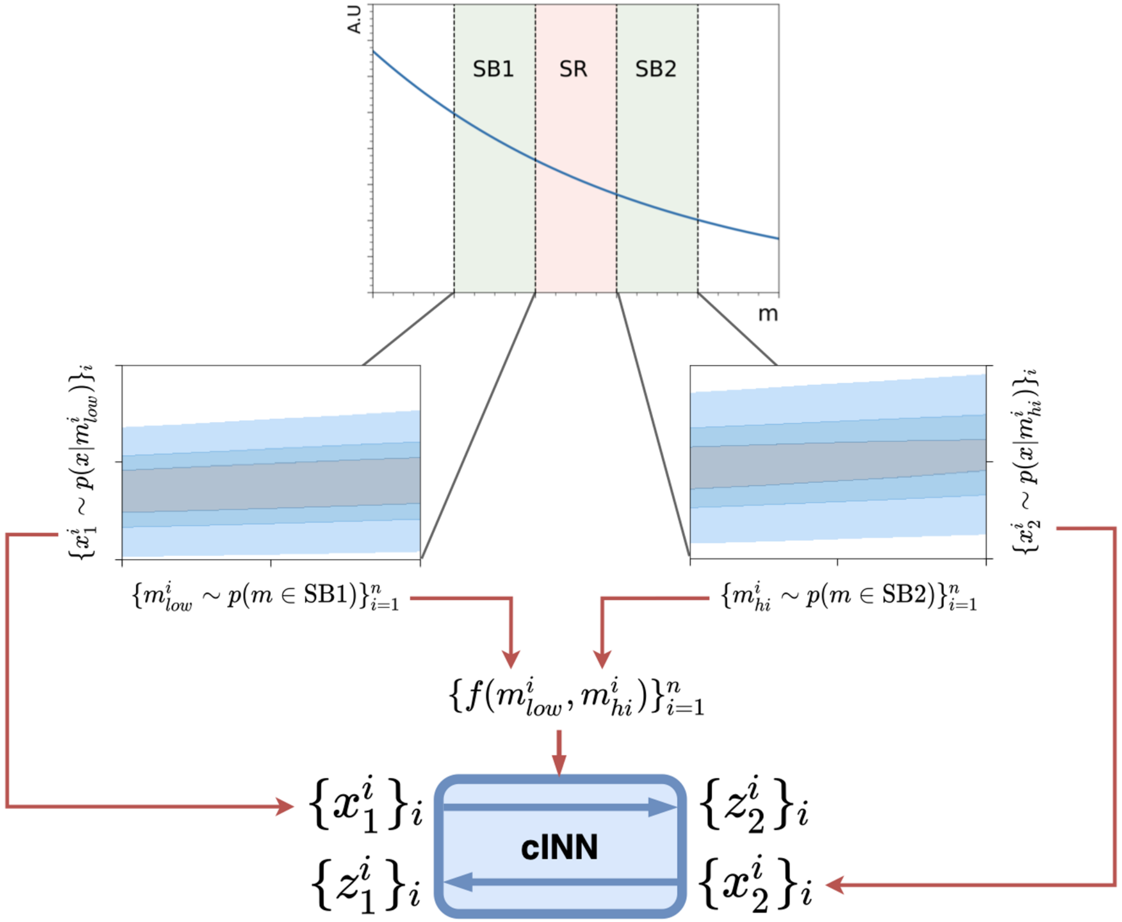
Originally about transporting dirt…

[source]

[Approximate Wasserstein distance with Sinkhorn]

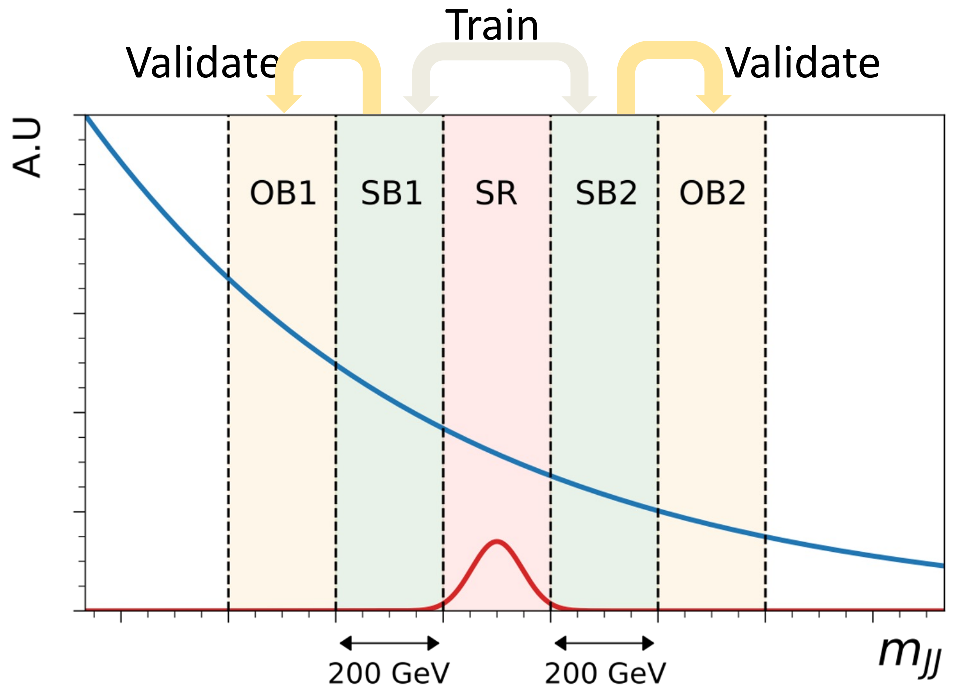
Training “SB-to-SR” transformation

- 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

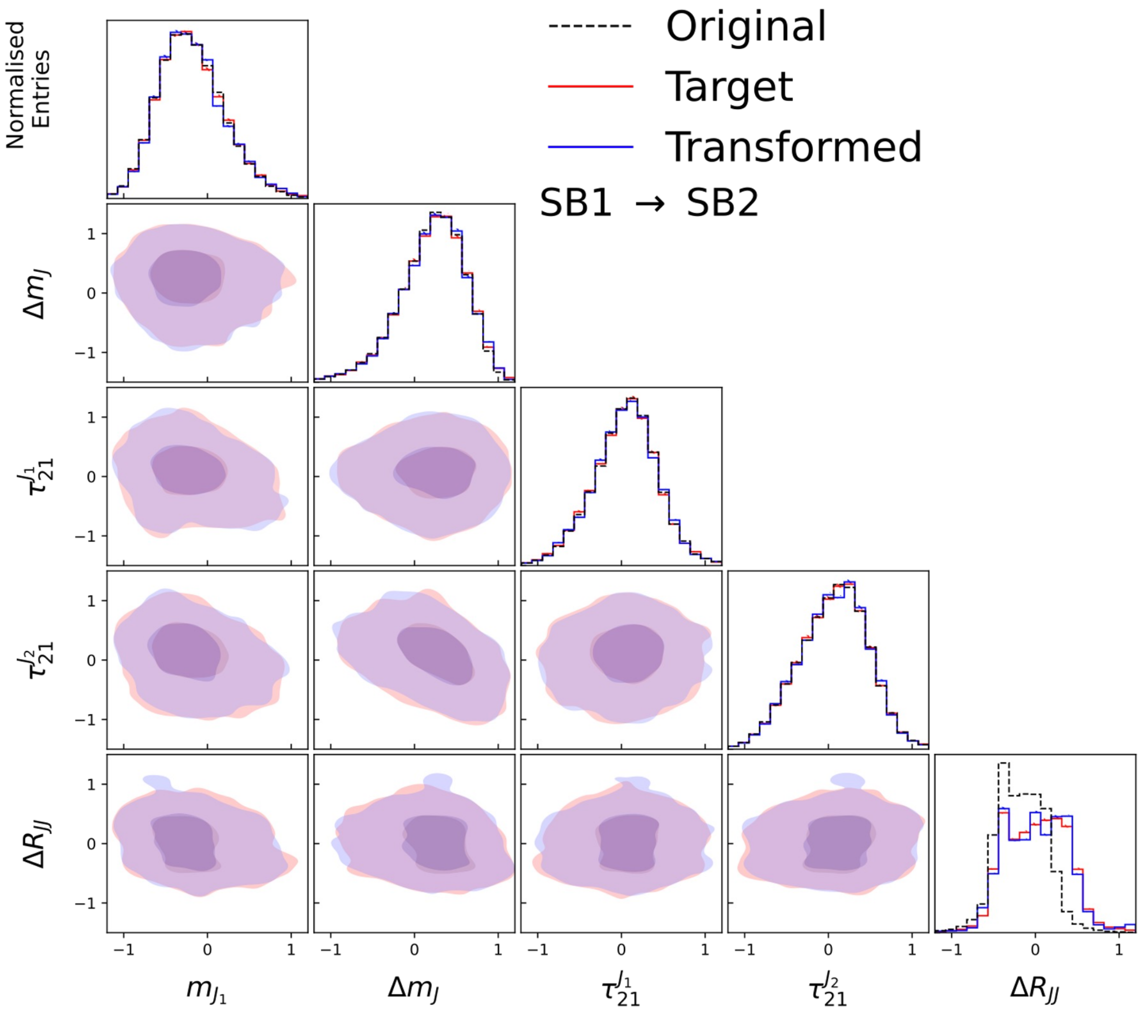


Training data

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

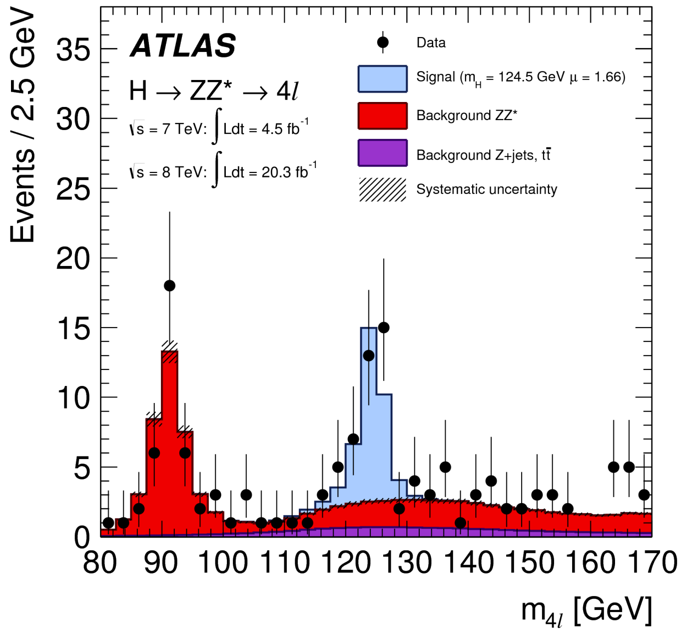
❑ Train a classifier to separate background from *signal*

A word on *labels*

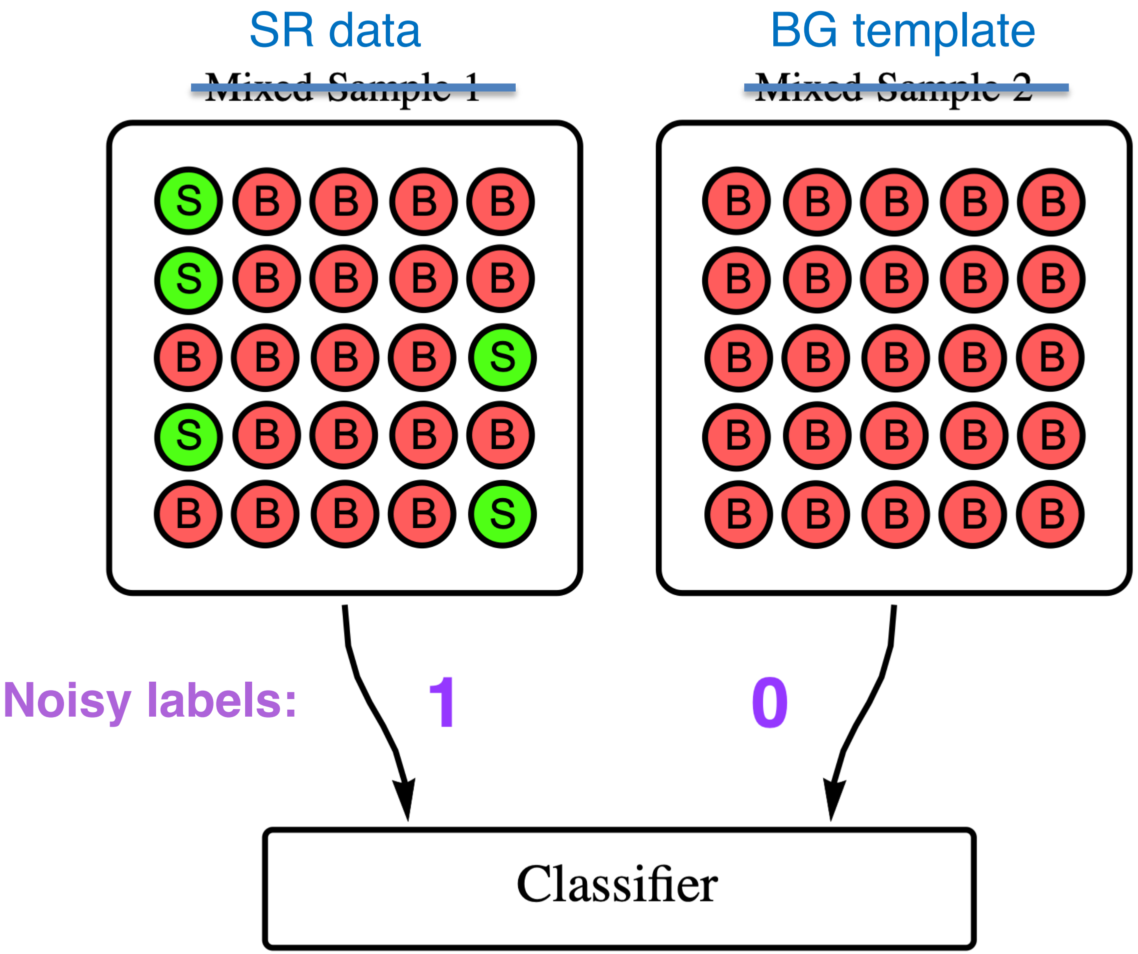
- Supervised labels are *inconsistent* with our view of the data

- No notion of *event label*

- Only *probability* to be signal or background



Classification without labeling (CWoLa)

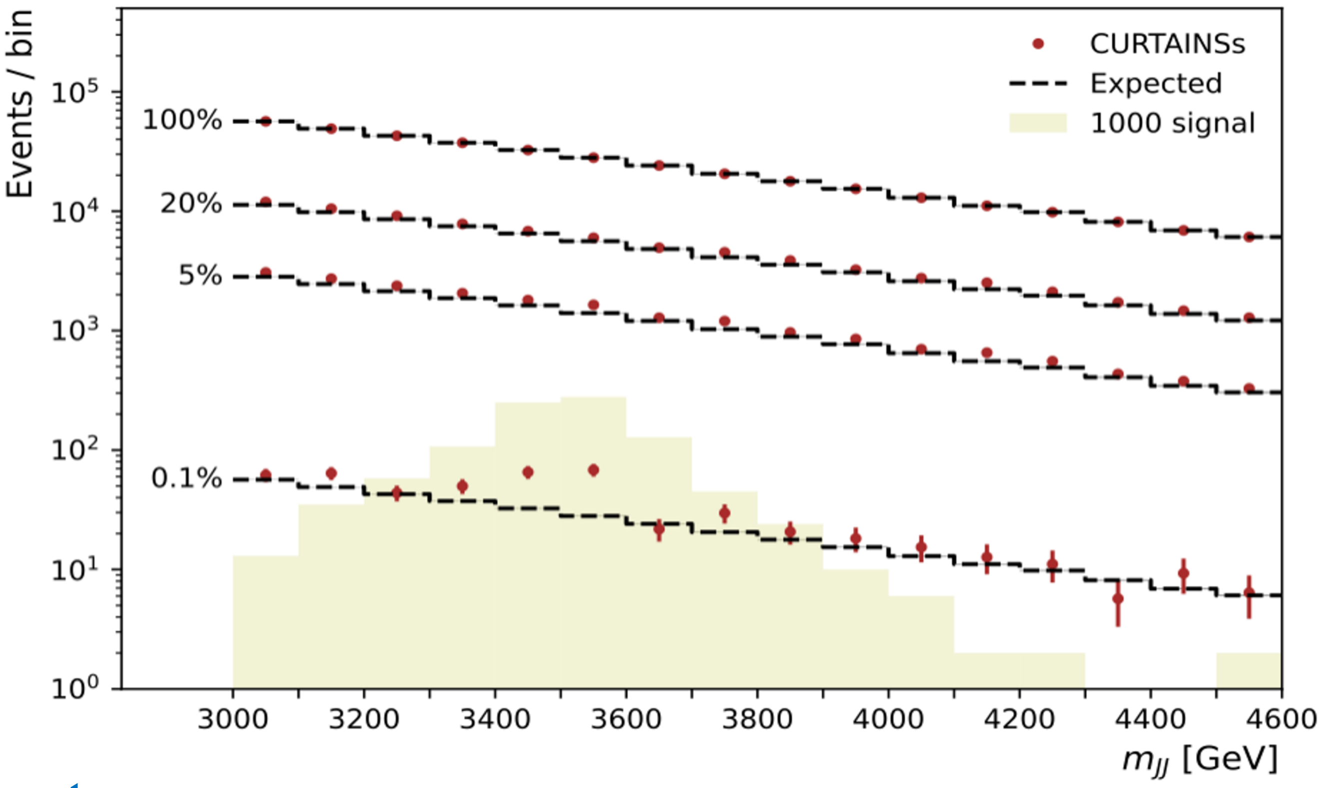


- Use noisy labels
- Shown to be optimal classifier
- Apply to data-only
- CWoLa for CURTAINs

[1702.00414, 1708.02949]

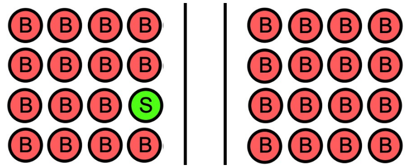
CURTAINs in action

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



Apply cut on CWoLa classifier

CURTAINs performance

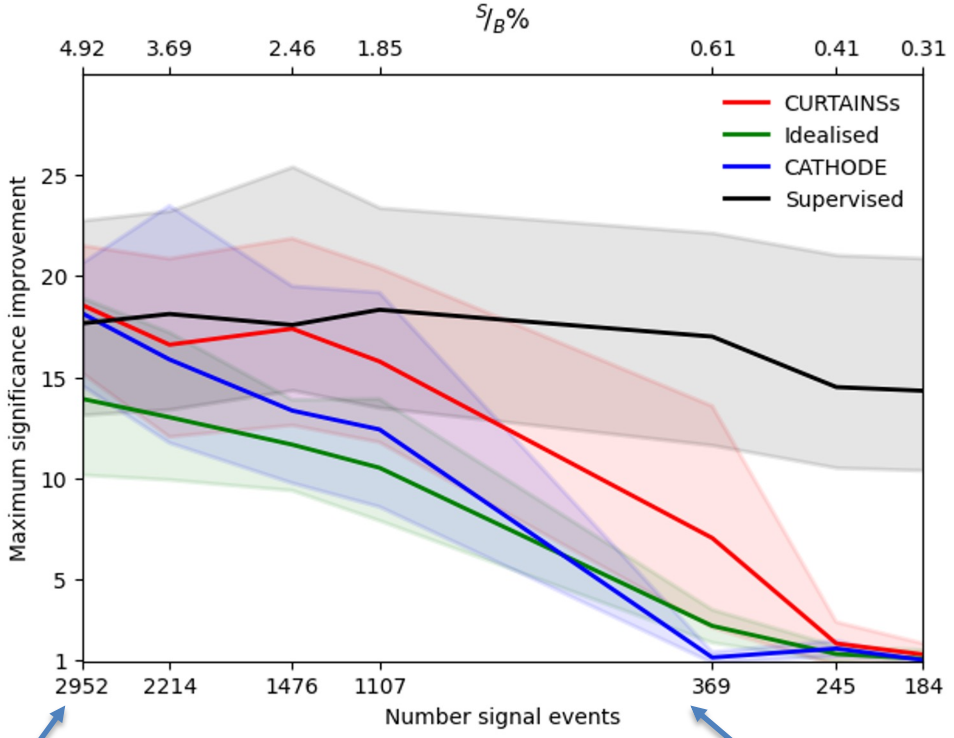


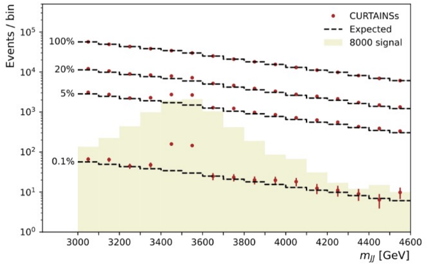
- CURTAINs

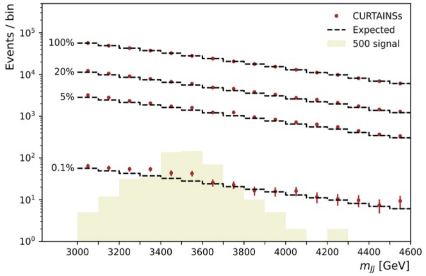
- Idealised: assume perfect BG template

- CATHODE
  - *Competition*: 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

Outlook: modeling vs. learning

The world of modeling

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

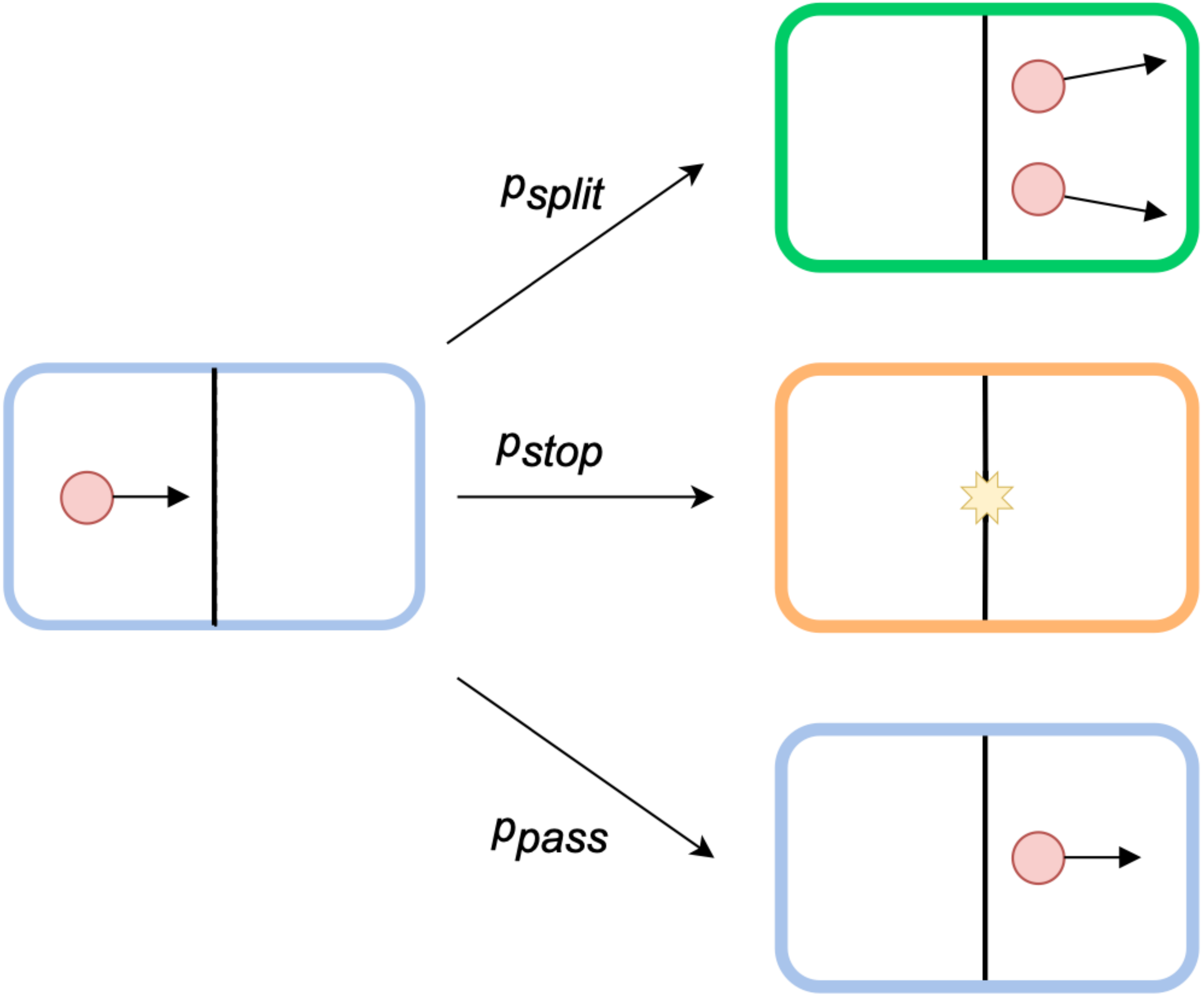
The world of learning

- Learning from **lots** of LHC data

The best of both worlds?

Backup

SUPA propagation model



Distance measure

How to estimate transformations of **distributions** over features?

We don’t have pairs, instead we want to shift one distribution to another

Optimal transport -Distance over batch, matching samples to closest neighbours

Map from SB1 to SB2 and vice versa, shuffling pairs every epoch