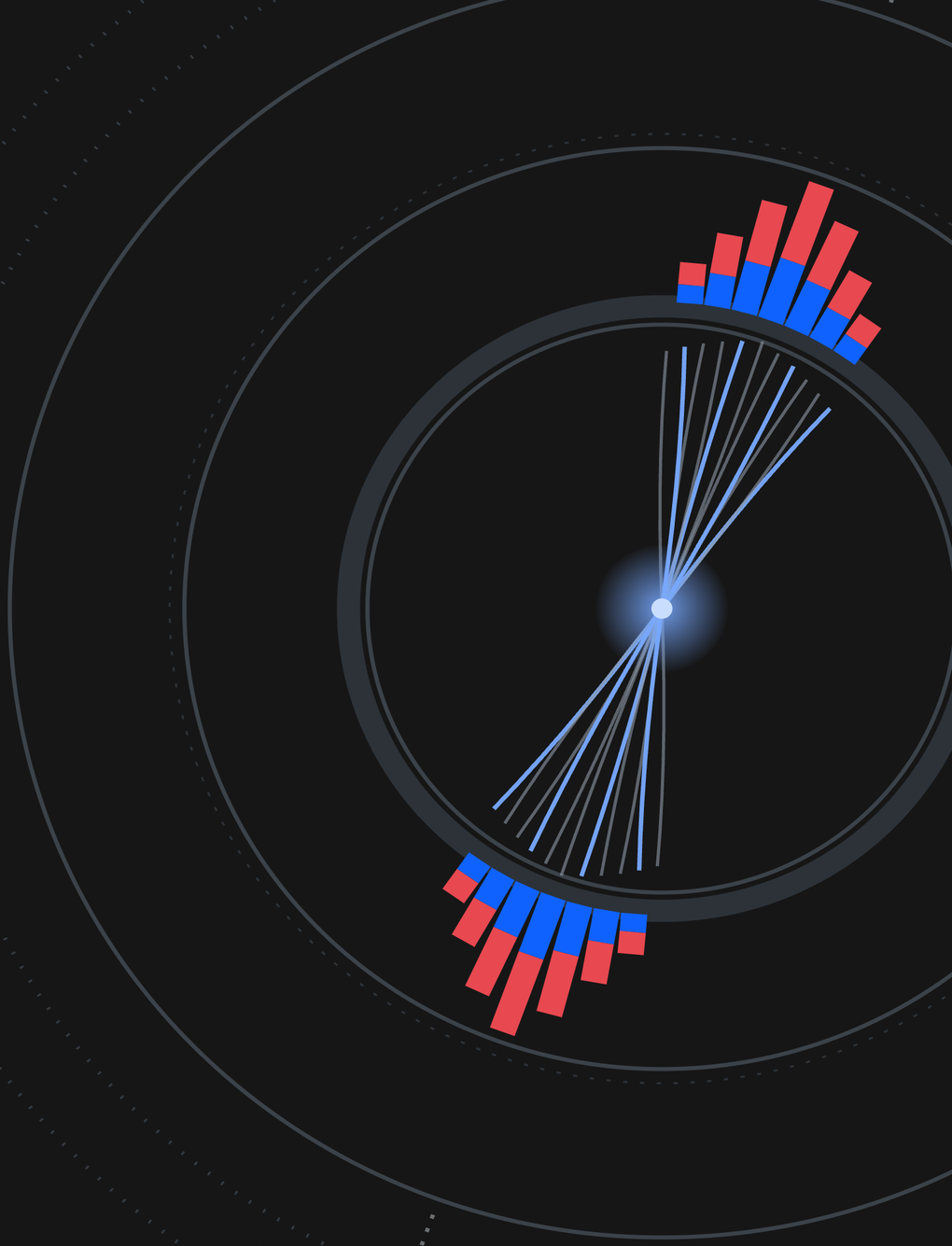
SCHOOL ON MACHINE LEARNING & AI METHODS AT LHC

Anomaly Detection & Unsupervised Learning at the LHC

Jackson Burzynski

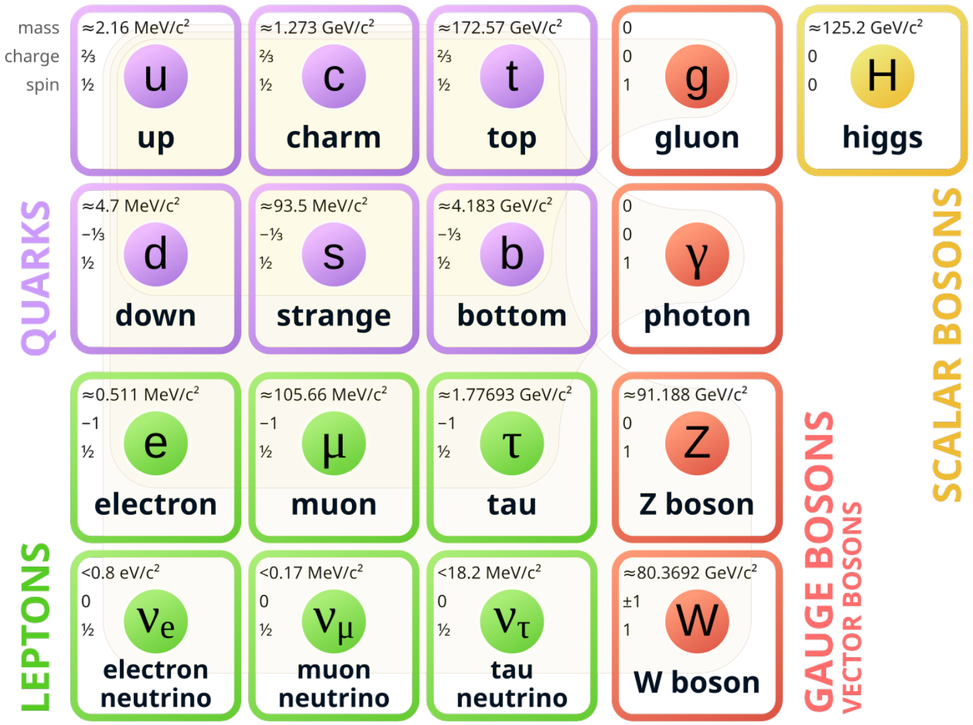
14 July 2026

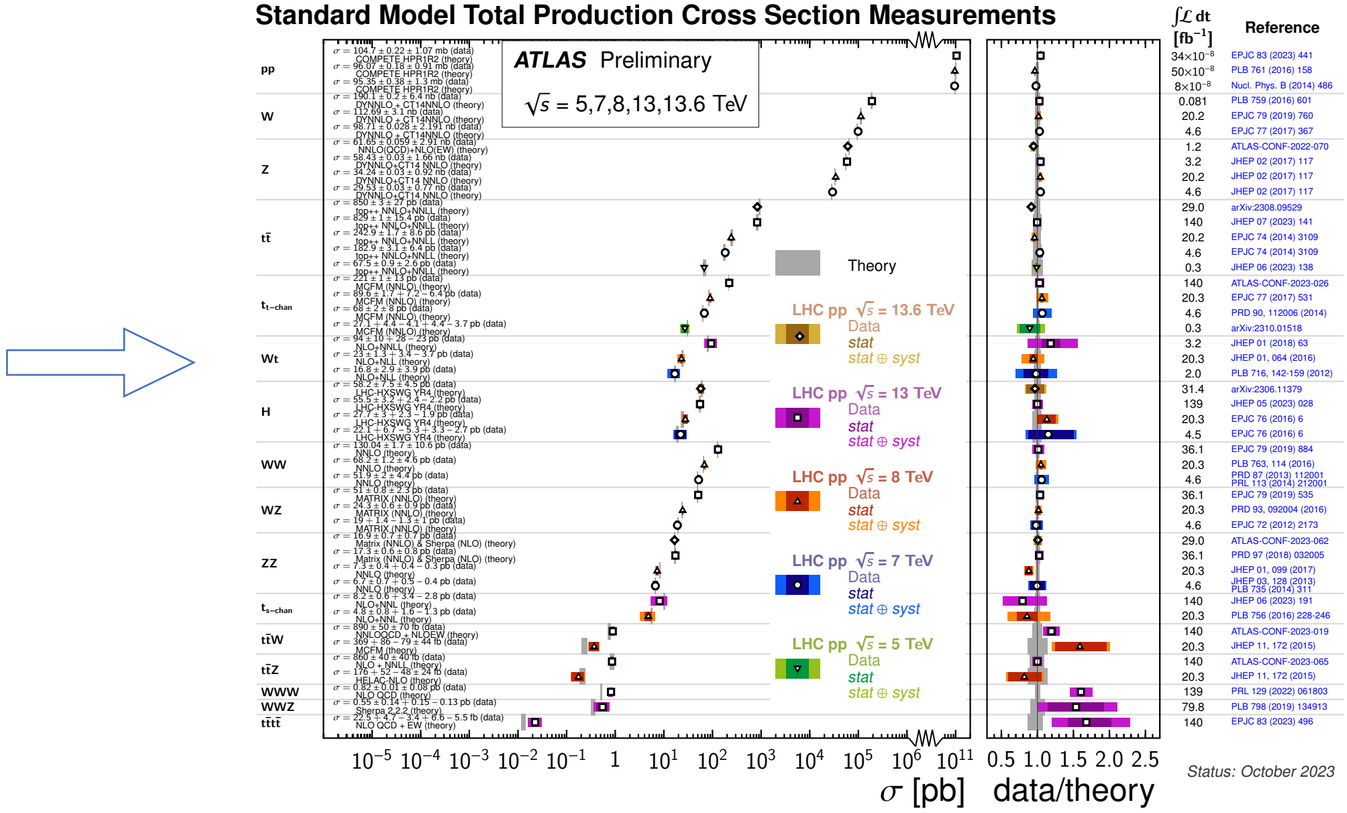


What we know about the universe

The Standard Model is a remarkable theory...

...with extreme predictive power!





MOTIVATION

What we *don't* know about the universe

In spite of the successes of the Standard model, we know it to be incomplete

- Dark matter
- The hierarchy problem
- Matter-antimattery asymmetry
- Neutrino masses

These unexplained phenomena point to the existence of new, ***beyond Standard Model*** **(BSM)** physics

What is dark matter (DM)?

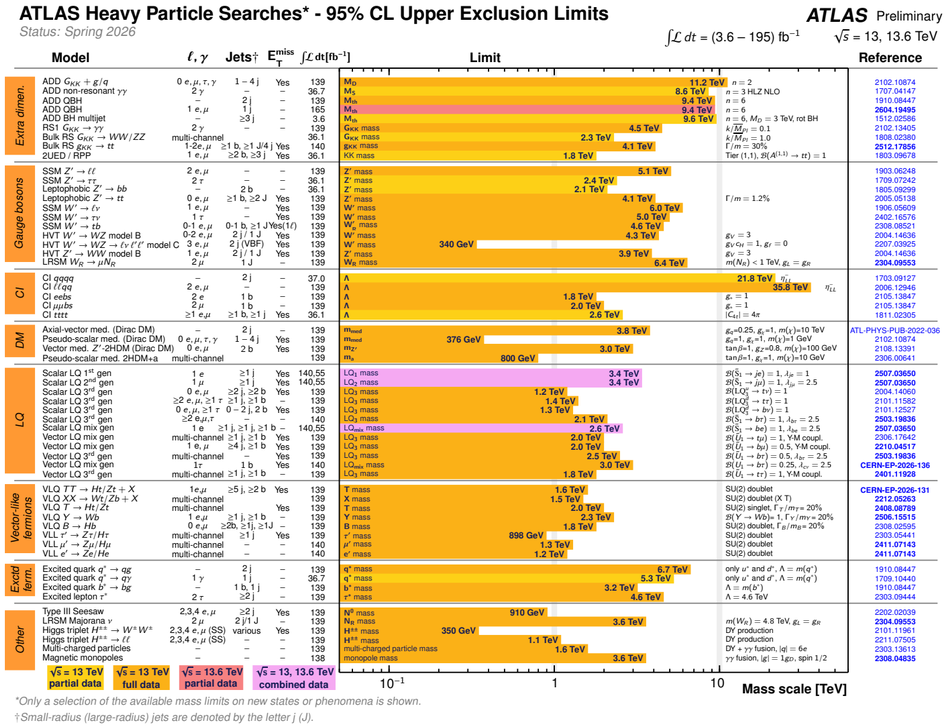
Why is there a hierarchy of scales between the fundamental forces?

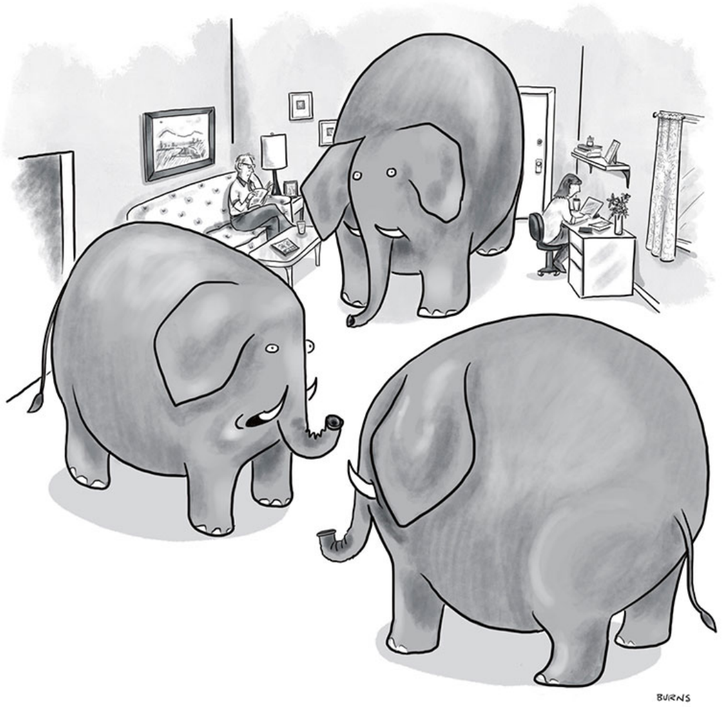
Where is all the antimatter?

Where do neutrinos get their mass?

The elephant in the room

Despite over a decade of searches at the LHC, we have yet to observe any definitive signs of BSM physics





We still have **just as many reasons** to expect new physics as before the LHC turned on…*so why haven't we found it yet?*

How to search for new physics

The “traditional” approach to search for new physics signals at the LHC is as follows:

1. Pick a signal model (e.g. new heavy boson Z'). Simulate the signal and study its detector signature. Understand the SM backgrounds.

2. Design a set of selections which isolate your signal model and remove as much background as possible (the “signal region”)
   - E.g. training a binary classification algorithm to separate signal from background and cutting on the output score

3. Estimate the amount of SM background that will enter your signal region. Look at the data in your signal region and check for consistency with your background prediction. Set exclusion limits on your signal model if no excess observed

PROS

- Maximal sensitivity to *that particular signal model*

CONS

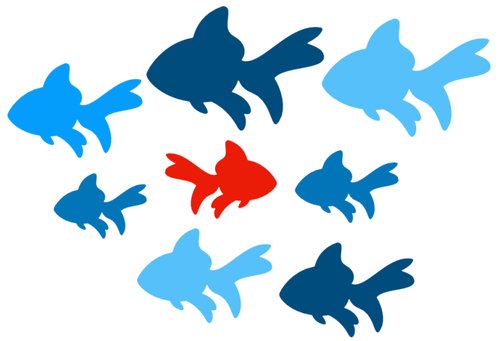
- Sees only the signal it was designed for. Need a dedicated search for each model

An alternative approach

Problem: We don't know what the new physics looks like! There are far more plausible theories than we could ever build dedicated searches for.

An alternative, model-agnostic approach:

1. Learn what "normal" (Standard Model) events look like. Assign a score to each event which quantifies how “anomalous” it looks. Beyond Standard Model signals should have high score.
2. Estimate how many background events will pass the selection on the anomaly score
3. Compare the data to your background prediction. An excess of data indicates something anomalous!



PROS

- One selection is broadly sensitive to many signals at once. Leave no stone unturned.

CONS

- Less sensitive to any one particular signal model than a dedicated search

The likelihood ratio

The optimal classifier is the ***likelihood ratio***

Neyman-Pearson lemma

$$L_{S/B}(X) = \frac{P_s(X)}{P_b(X)}$$

Prob. distribution of signal

Prob. distribution of background

In anomaly detection we do not know P_s. So how can we approximate the likelihood ratio then?

- Outlier Detection : Learn P_b, take anomaly score as $1/P_b$
- Over Densities : Leverage localization of signal to learn $L_{S/B}$ from data

Figures curtesy of A. Badea

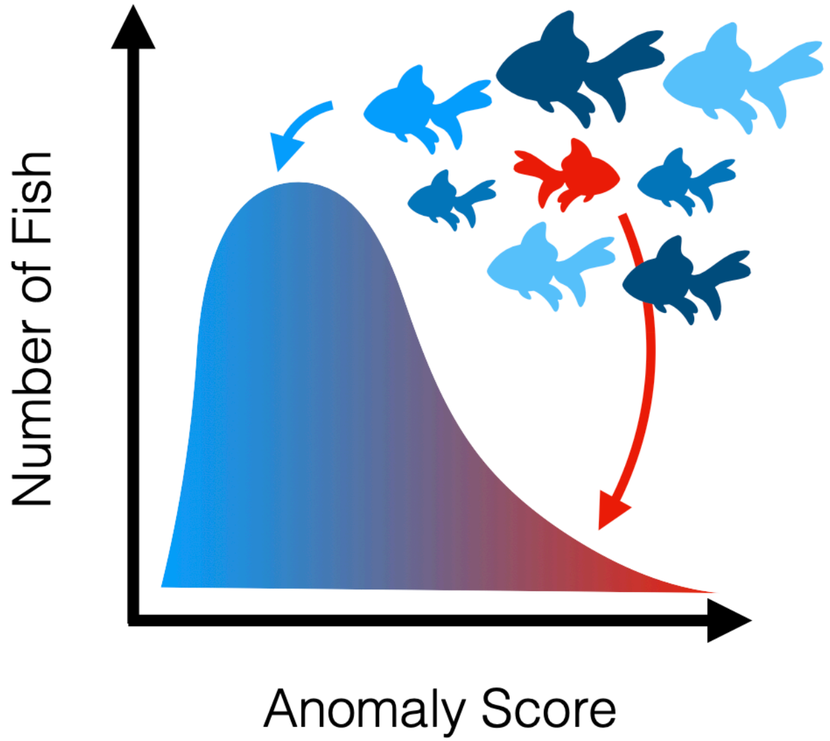
ANOMALY DETECTION

How to quantify “anomalous”

In practice, anomaly detection (AD) generally uses machine learning based methods to quantify anomaly score

ML-based AD searches rely on two tasks:

1. Isolate anomalous events
2. Estimate the background



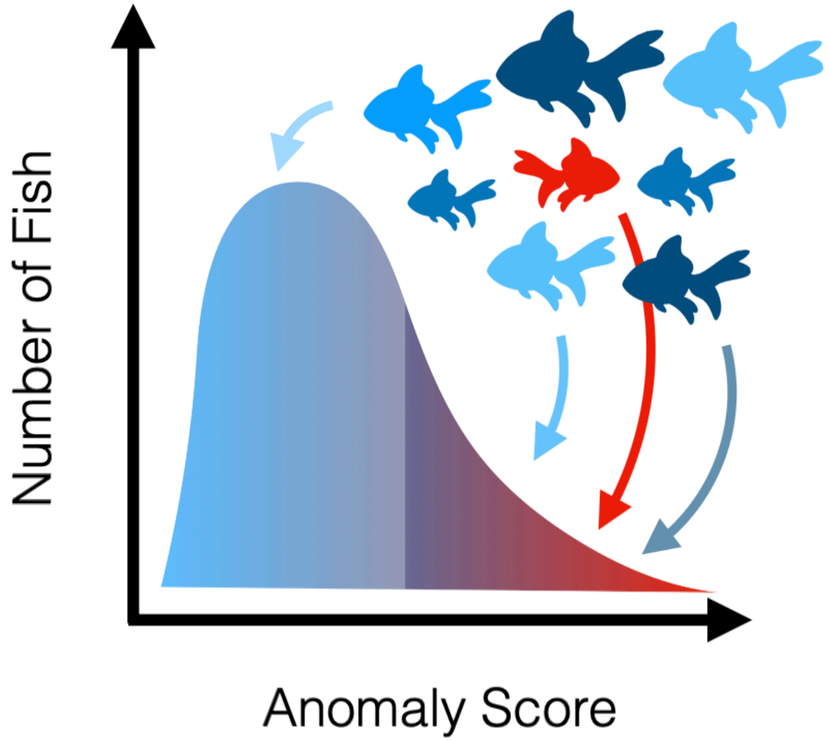
Figures curtesy of A. Badea

How to quantify “anomalous”

In practice, anomaly detection (AD) general uses machine learning based methods to quantify anomaly score

ML-based AD searches rely on two tasks:

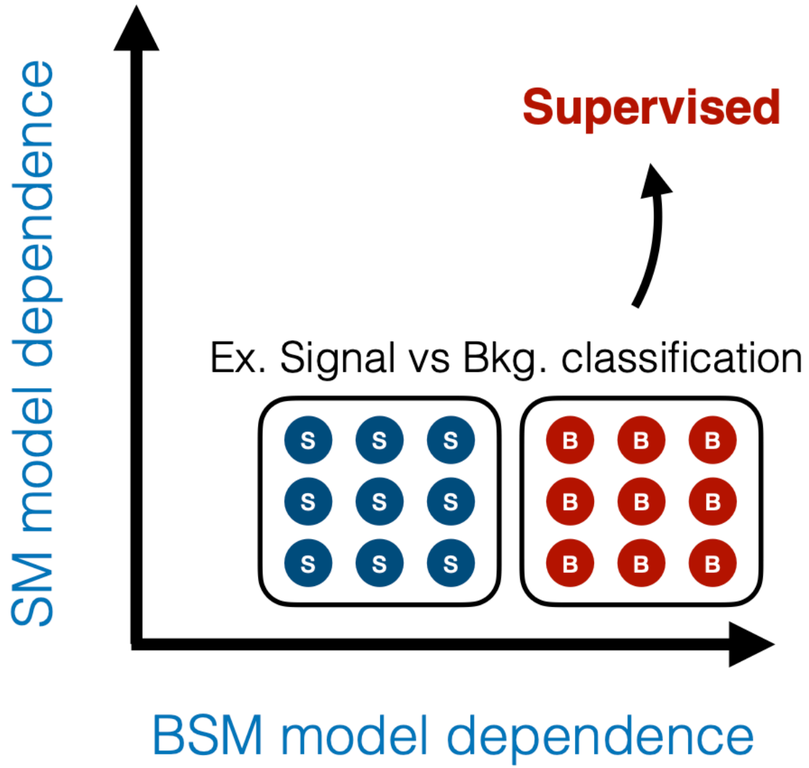
1. Isolate anomalous events
2. Estimate the background



Figures curtesy of A. Badea

The conventional approach — supervised searches

The two axes measure how much a method **depends on a model** — of the signal (horizontal) and of the background (vertical).



Supervised

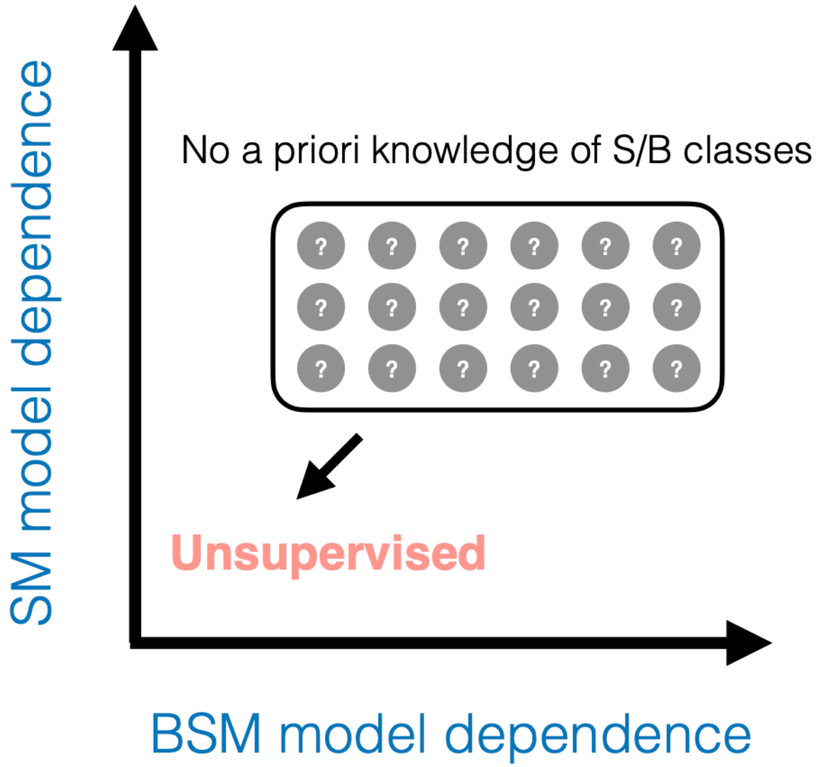
Train on **labelled** signal (S) vs. background (B), both from simulation.

Maximum power — but it sits in the high-model-dependence corner, sensitive only to the signal you assumed.

Figures curtesy of A. Badea

The opposite corner: unsupervised

The two axes measure how much a method **depends on a model** — of the signal (horizontal) and of the background (vertical).



Unsupervised

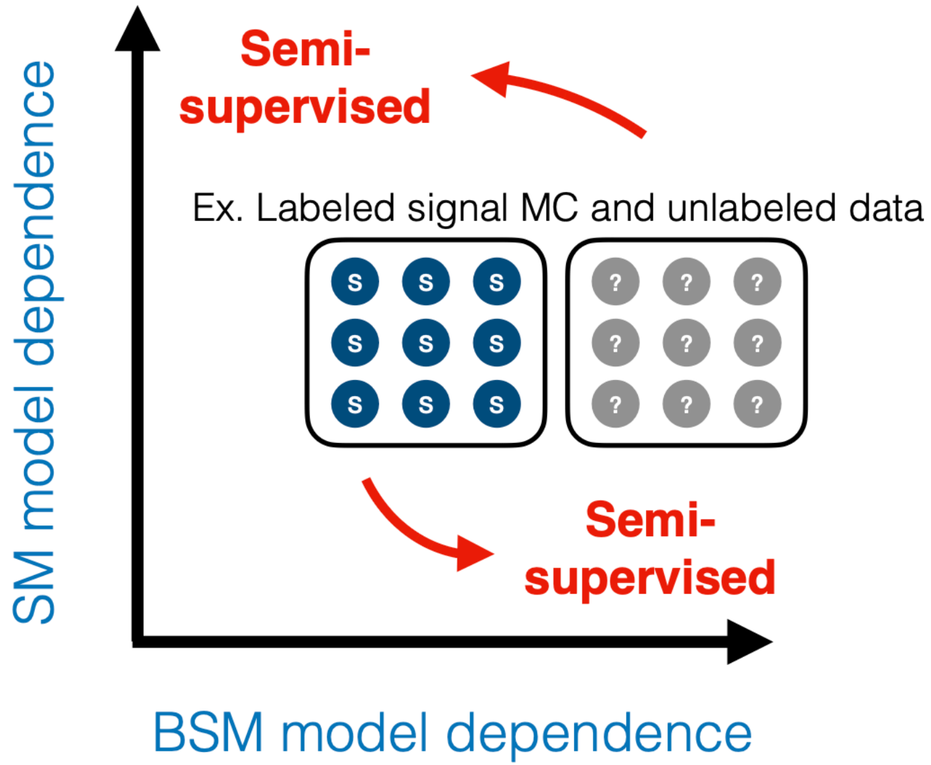
No labels at all. The method sees only unlabelled data and learns its structure — no prior knowledge of what signal or background look like.

Lowest model dependence, broadest reach. Home of **autoencoders, VAEs and normalizing flows** — best at finding **outliers**.

Figures curtesy of A. Badea

In between: semi-supervised

The two axes measure how much a method **depends on a model** — of the signal (horizontal) and of the background (vertical).



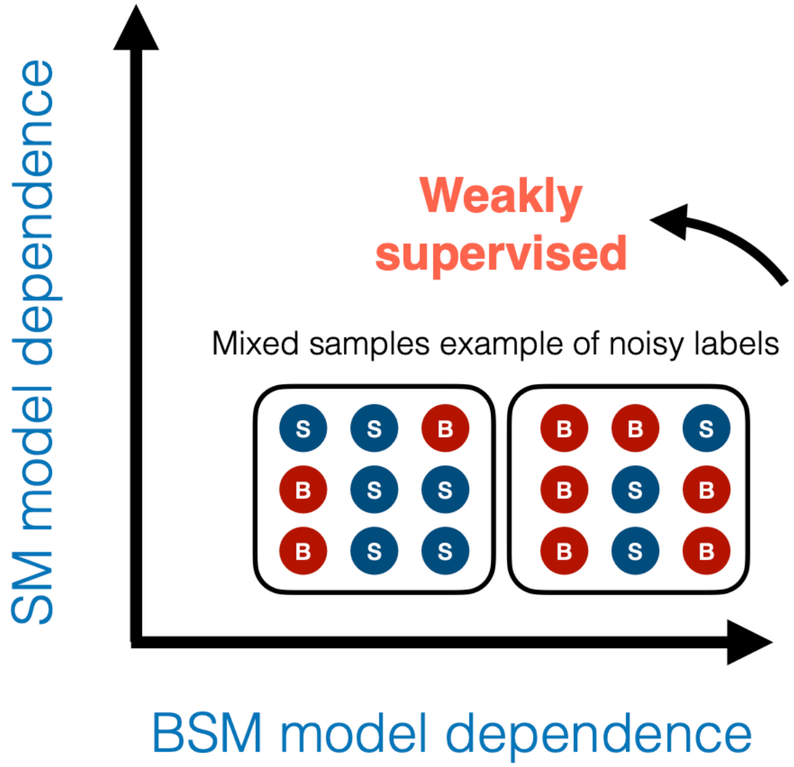
Semi-supervised

Use a sample **known to be background** — typically simulation — alongside unlabelled data. Learn "normal" from the trusted sample, then flag data that departs from it.

Figures curtesy of A. Badea

Weakly supervised

The two axes measure how much a method **depends on a model** — of the signal (horizontal) and of the background (vertical).



Weakly supervised

Noisy / mixed labels. Neither sample is pure — e.g. a signal region vs. a sideband — but they differ in signal content. Best for **over-densities**

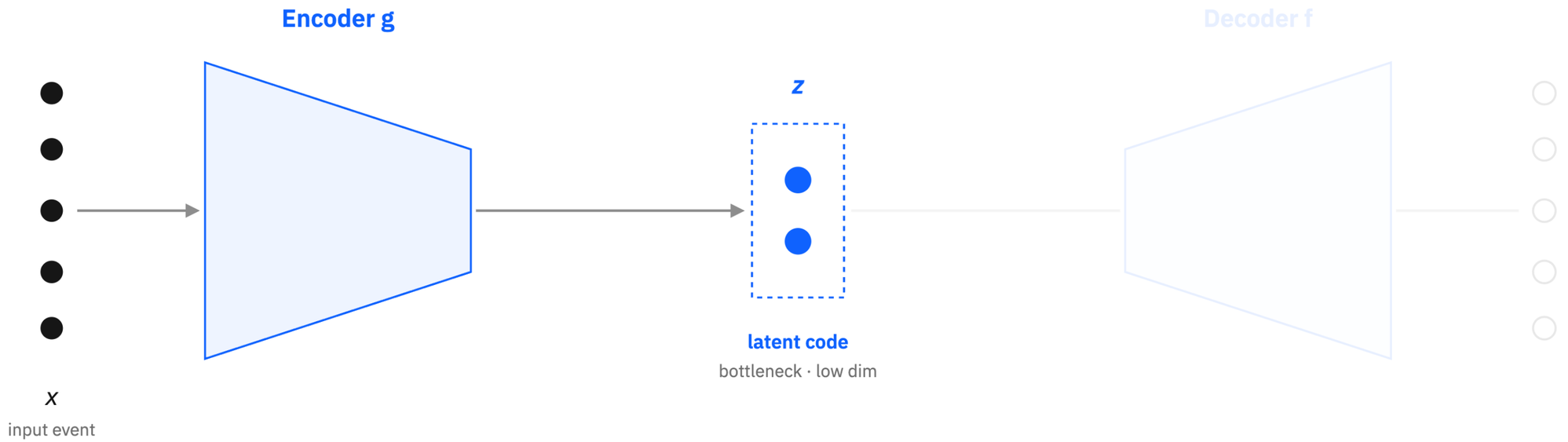
Across the map, **fewer label assumptions → more generality**

01

PART 01

Unsupervised methods

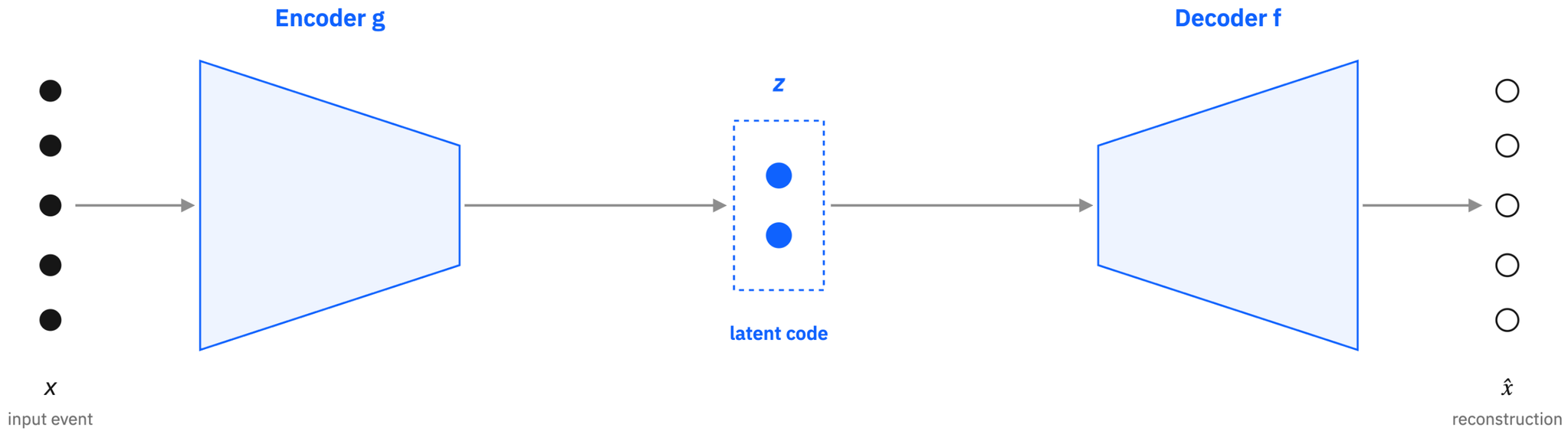
① Encode: compress into a latent code



Encoder $z = g(x)$ squeezes the event into a handful of numbers

The bottleneck is too narrow to memorise the input — so it must capture the **common structure** of the data.

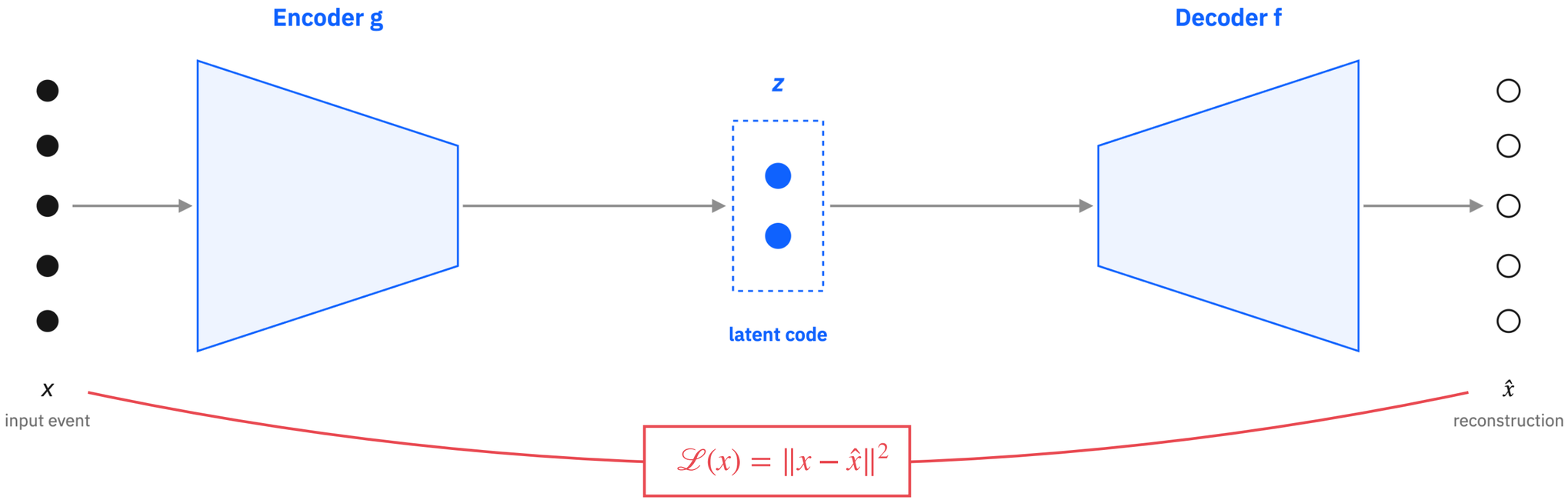
② Reconstruct from the code



Decoder $\hat{x} = f(z)$ rebuilds the event from the code.

Trained only on background, the network learns to reconstruct **typical events — and nothing else**.

③ Score by reconstruction error



The **reconstruction error** is the anomaly score.

Small for typical events the network rebuilds well — **large** for events it never learned to compress

The reconstruction loss

Training adjusts the encoder and decoder weights to make the reconstruction $\hat{x}$ as close as possible to the input x.

The standard measure is the **mean squared error**:

$$\mathscr{L}(x) = \|x - \hat{x}\|^2 = \sum_i (x_i - \hat{x}_i)^2$$

summed over the i = 1 … D features of the event
Training minimises its average over all events

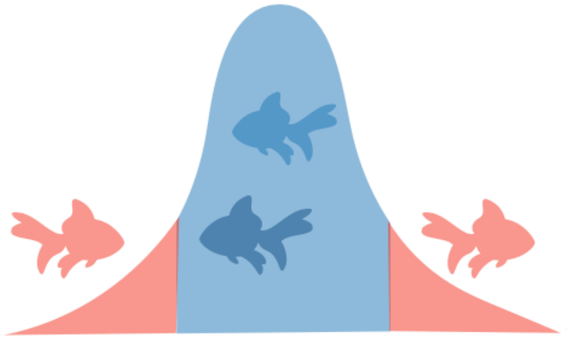
READING THE FORMULA

x — the event, as a vector of measured quantities (masses, momenta, angles).

$\hat{x} = f(g(x))$ — what the autoencoder rebuilds after the bottleneck.

$\mathscr{L}$ small → the event fits the learned “normal"
$\mathscr{L}$ large → the network struggled → **candidate anomaly.**

- The network is trained **only on background-like data** , so it learns to reconstruct background well and nothing else.
- Other choices exist: binary cross-entropy for probabilities, or per-feature weighting when features have different scales.
- The same $\mathscr{L}(x)$, evaluated event-by-event at test time, is the **anomaly score**.

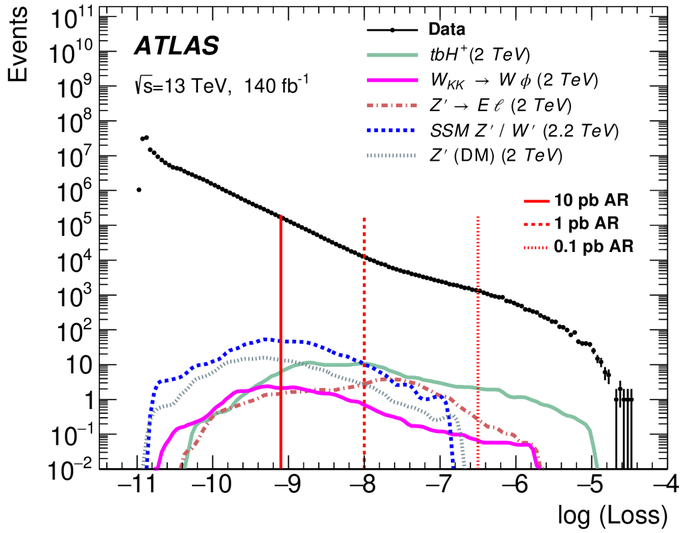


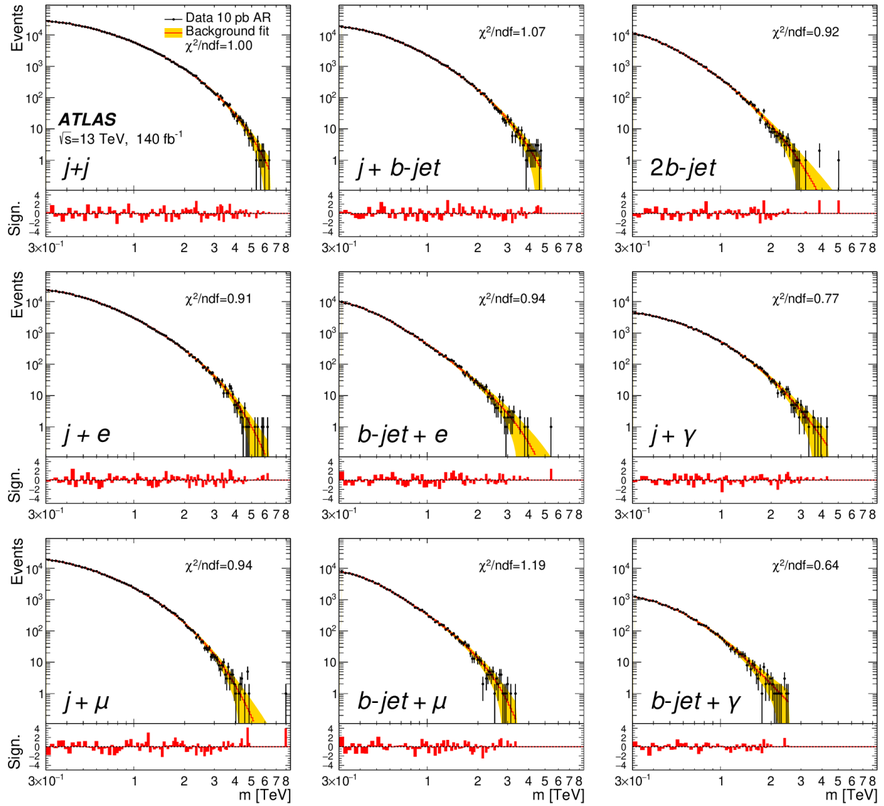
Reconstruction probability

ATLAS | Phys. Rev. Lett. 132 (2024) 081801 · arXiv:2307.01612

A generic two-body resonance search with an autoencoder

- An autoencoder is trained **directly on data** (events with ≥1 electron or muon). The decoder's **reconstruction loss** is the anomaly score.
- Signals concentrate at **high loss**. Define **anomalous-region (AR) thresholds**
- Inside the anomalous region, **nine two-body mass spectra** are built. Each spectrum is fit with a smooth **background function**. A resonance would appear as a localised bump on top.





No significant excess is found. Limits are set on generic Gaussian signals of various widths.

Common pitfalls

⚠ LATENT DIMENSION

Too wide → the network copies the input through. Every event reconstructs perfectly and nothing looks anomalous.

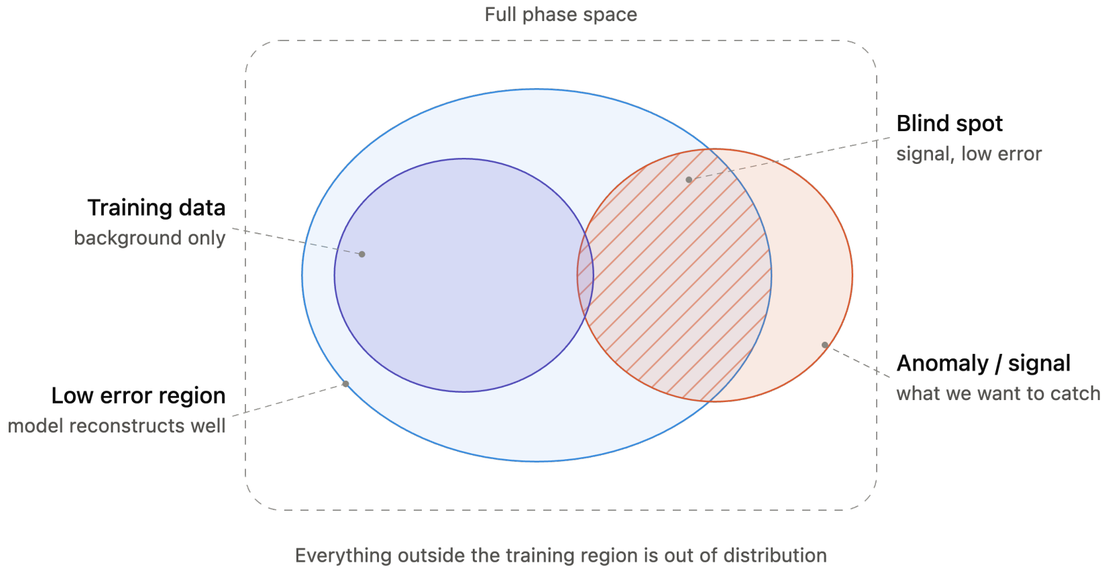
Too narrow → even background reconstructs badly and the loss becomes noise. The sweet spot forces genuine compression of common structure.

⚠ THE COMPLEXITY-BIAS TRAP

More ‘complex’ data (higher intrinsic dim) harder to compress, seen as more anomalous

⚠ OVER GENERALIZATION

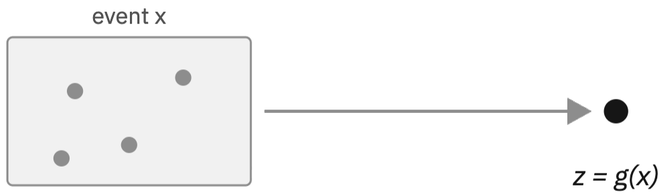
AE can reconstruct things well even outside training phase space because no penalty to do this



From a point to a distribution

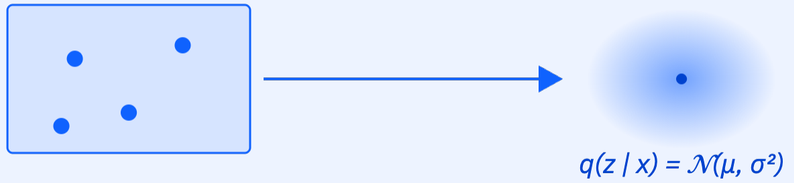
PLAIN AUTOENCODER

z is a single point

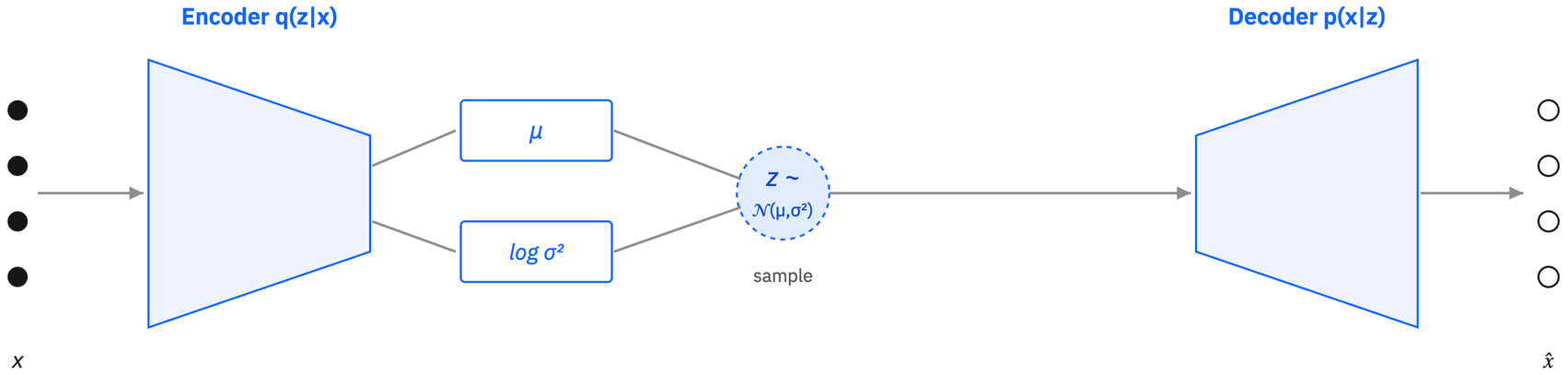


VARIATIONAL AUTOENCODER

z is drawn from a distribution



The encoder outputs a **mean μ and spread σ** . A prior pulls all these Gaussians toward $\mathcal{N}(0,1)$, filling the space smoothly so it is continuous *and* generative.



The training objective: the ELBO

The true likelihood $\log p(x) = \log \int p(x \mid z)\, p(z)\, dz$ is intractable. We maximise the **Evidence Lower BOund (ELBO)** instead:

$$\log p(x) \geq \mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q(z|x)}[\log p(x \mid z)] - KL(q(z \mid x) \,\|\, p(z))$$

① Reconstruction — want high

Expected log-likelihood of rebuilding x from a sampled z.

In practice, the (negative) mean-squared or cross-entropy error, same as the autoencoder loss.

② KL divergence — want small

How far the encoded distribution strays from the prior $\mathcal{N}(0,1)$.

Acts as a regulariser that keeps the latent space continuous and prevents memorisation.

For diagonal Gaussian priors, the KL term is closed-form

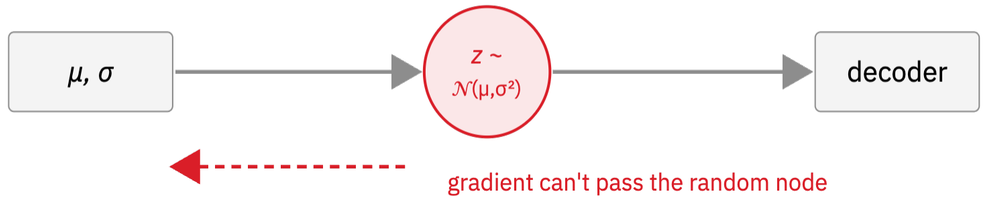
$$KL = -\frac{1}{2} \sum_j (1 + \log \sigma_j^2 - \mu_j^2 - \sigma_j^2)$$

The reparameterization trick

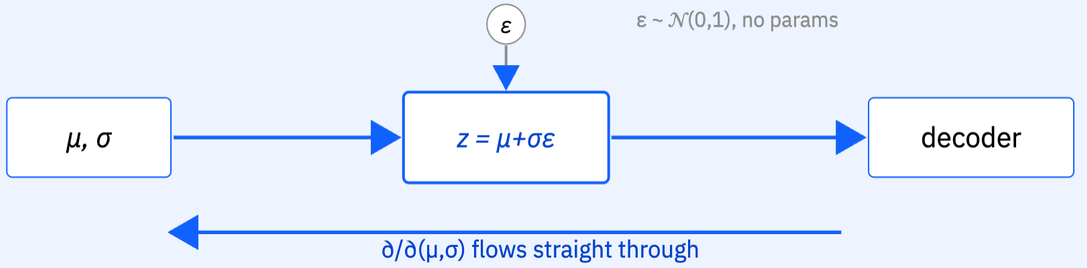
To train the encoder we need gradients through the sampling step — but you **can't back-propagate through a random draw**.

The fix: move the randomness into a parameter-free noise variable.

✗ NAÏVE SAMPLING — gradient blocked



✓ REPARAMETERIZED — gradient flows

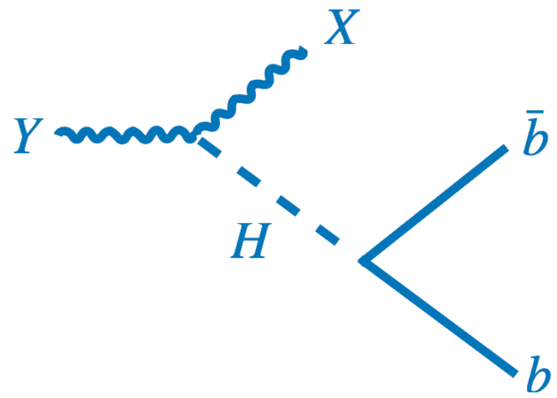


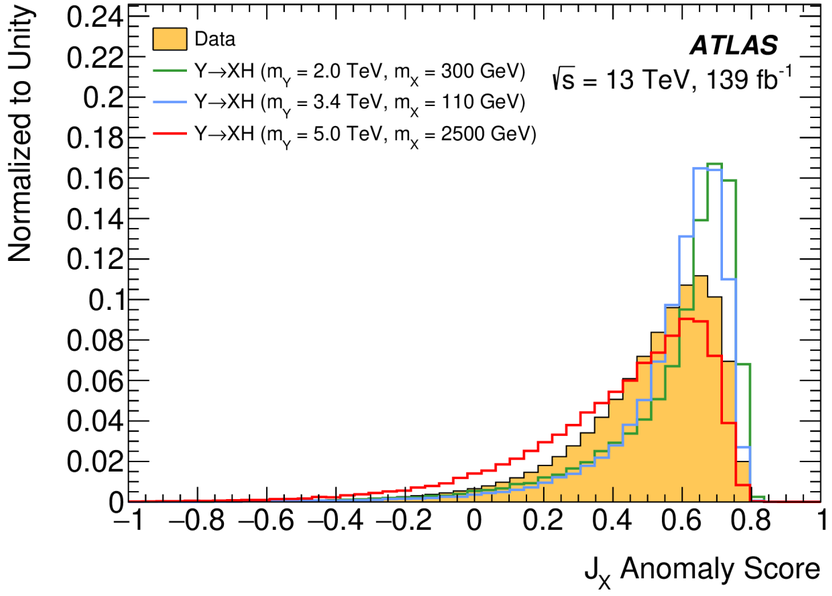
$$z = \mu + \sigma \odot \varepsilon,\ \varepsilon \sim \mathcal{N}(0, I)$$

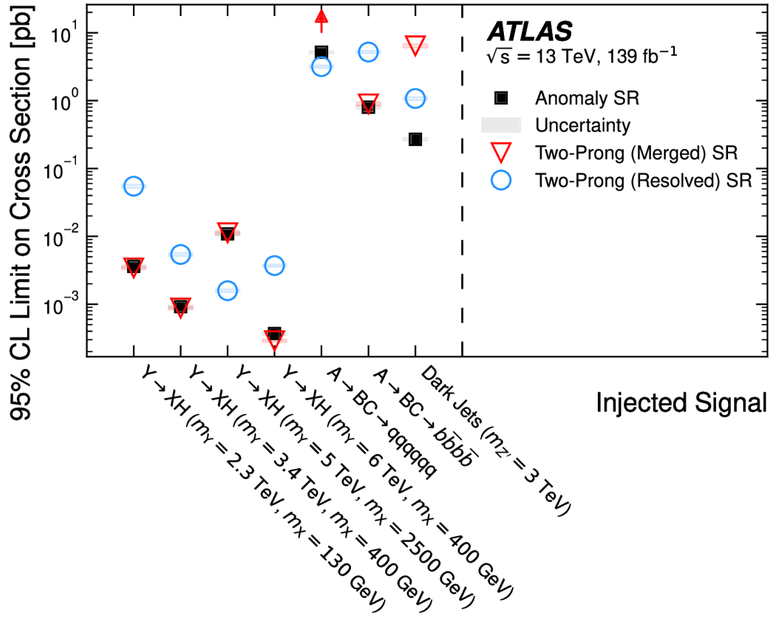
ATLAS | Phys. Rev. D 108 (2023) 052009 · arXiv:2306.03637

A fully-unsupervised jet tagger for Y → XH

- Search for Y → X H, H → $b\bar{b}$, fully hadronic
- Train a fully unsupervised **variational recurrent neural network** (VRNN), a VAE whose latent space is updated at each step of an RNN reading the jet constituents.
- It outputs a jet-level **anomaly score** , built from the VAE reconstruction loss plus its KL term.







Learning the exact likelihood

The methods discussed so far do not attempt to learn the exact likelihood function

- **Autoencoder**: MSE loss encodes no information about the true probability distribution of the background events
- **Variational Autoencoder**: ELBO gives us a lower bound on the log-likelihood, but still approximate

We know from the Neyman-Pearson lemma that the likelihood ratio test is the most powerful, so why not try to learn it?

- This is the idea behind **normalizing flows**

A normalizing flow learns $p(x)$ by transforming a simple distribution into the complicated one with an invertible function.

- Start with a "base" random variable z drawn from something trivial you know exactly: a unit Gaussian, $z \sim N(0,I)$
- Learn an invertible, differentiable function f that maps $z \to x$. Its inverse f^{-1} maps your data x back to z ("normalizing" the data into the Gaussian — hence the name).
- Because f is invertible, every data point x corresponds to exactly one z, and you can move freely in both directions.

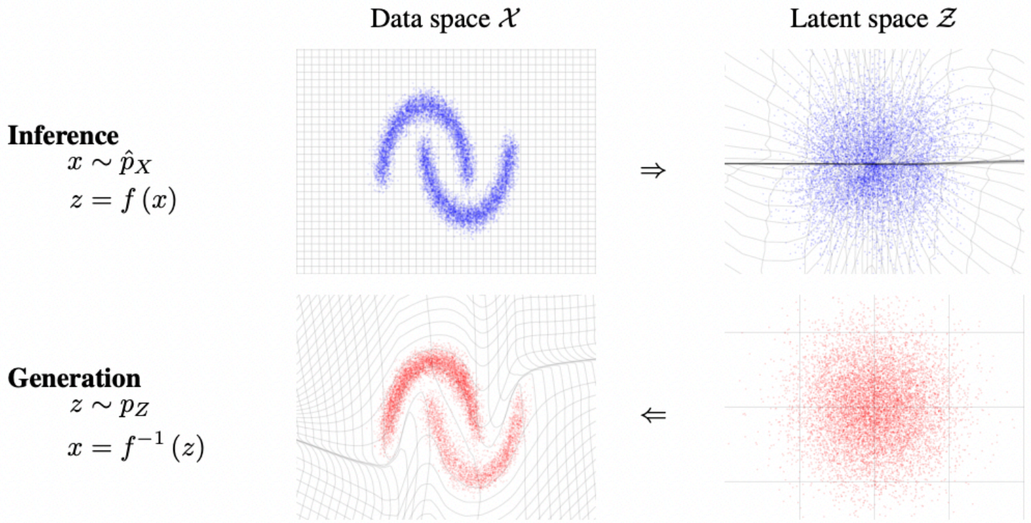
Change of variables

arxiv:1605.08803

Given an observed data variable $x \in X$, a simple prior probability distribution p_z on a latent variable $z \in Z$, and a bijection $f : X \to Z$, the change of variable formula defines a model distribution on X by

$$p_X(x) = p_z(f(x)) \cdot |\det J_f(x)|$$

$$\log p_X(x) = \log p_z(f(x)) + \log |\det J_f(x)|$$



The Jacobian is key component

The determinant measures the local volume change of the transformation. So the density of any data point is:

(how likely its z is under the Gaussian)

× (how much the map locally rescales volume)

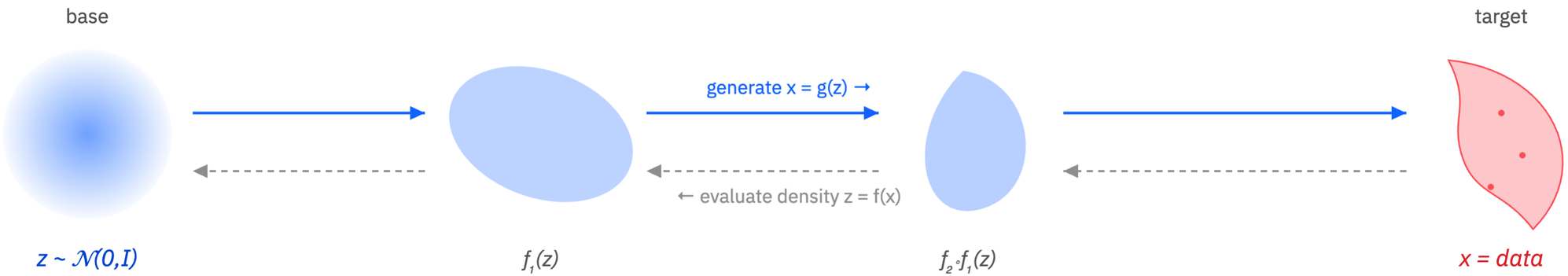
Two requirements

① **Invertible** f — so we can map both ways.

② **Cheap det J** — a general D×D determinant costs $\mathcal{O}(D^3)$, need architectures where it's $\mathcal{O}(D)$.

Stacking invertible transformations

One simple map isn't enough. We **compose many** — the composition is the “flow"



Invertible both ways: flow to the base to score, flow from the base to generate.

Determinants multiply, so the log-Jacobian is a simple sum over layers.

$$\log p_X(x) = \log p_z(f(x)) + \sum_k \log |\det J_k(x)|$$

Coupling layers make it tractable — RealNVP

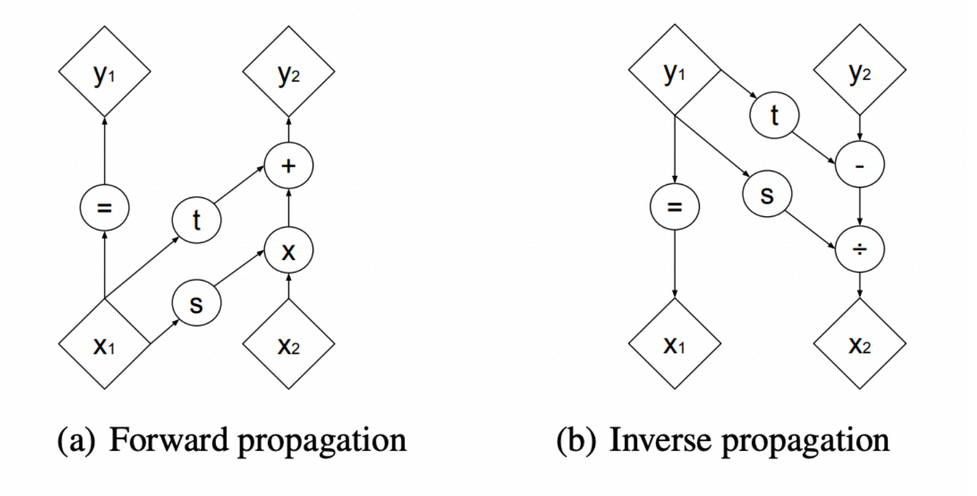
To make the determinant tractable, we design the layers so that each Jacobian is triangular. The determinant is then just the product of the diagonal.

The workhorse is the **coupling layer**:

- Split the input vector into two halves, x_1 and x_2.
- Pass half x_1 through unchanged.
- Transform half x_2 using parameters computed from x_1:

$$y_2 = y_1 \cdot \exp(s(x_1)) + t(x_1)$$

where s and t are ordinary neural networks.



(a) Forward propagation (b) Inverse propagation

Invertible by construction: $x_2 = (y_2 - t(x_1)) \cdot \exp(-s(x_1))$ — no need to invert the networks themselves.

Alternate which half is frozen across layers, so **every dimension** is eventually transformed.

Training, and density as an anomaly score

① Train by maximum likelihood

$$\max \sum_{\text{events}} \log p(x) \iff \min \text{NLL}$$

Fit the flow to a **background sample** — data or SM simulation — so it learns the Standard Model density p(x).

② Score by density

$$s(x) = \log p(x) - \log p_{\max}$$

Mapped to `[0,1]`. Typical events near 0, **rare low-density events near 1**. Cut high to select outliers.

Why flows suit outlier detection

Because the score *is* the likelihood, flows are naturally tuned to **out-of-distribution** events — the sparsely populated tails of phase space — rather than over-densities.

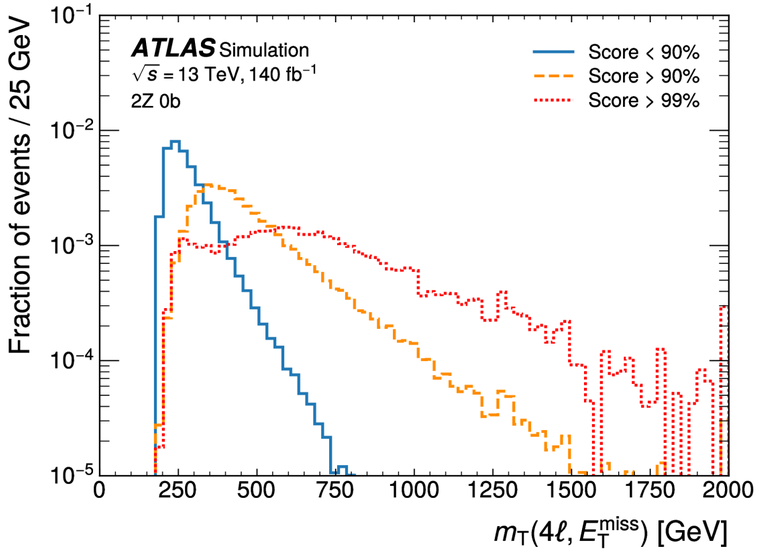
A practical note: like any density model, a flow is only as trustworthy as the sample it was trained on. Careful validation of the background model is essential.

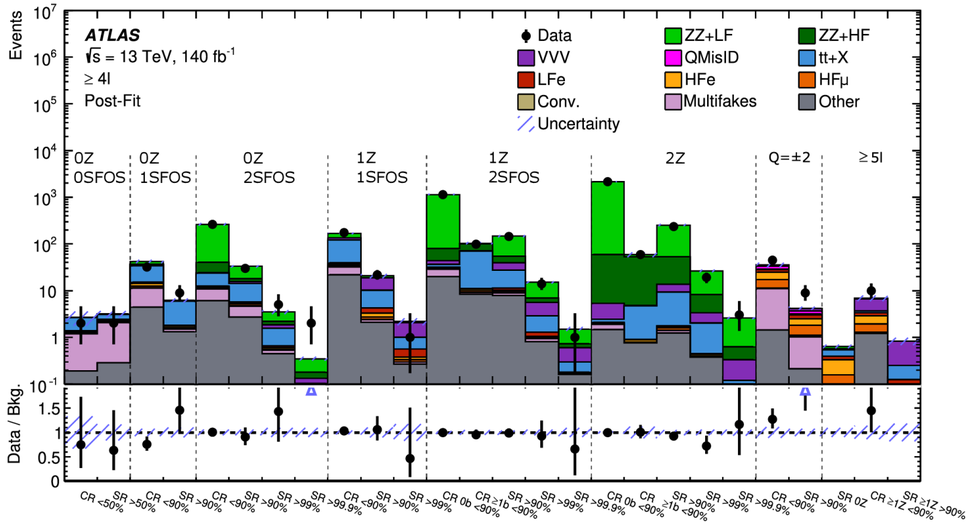
NORMALIZING FLOWS · IN PRACTICE

Outlier detection in multilepton events

- The **first AD search in multilepton final states** (≥ 4 light leptons)
- A **RealNVP flow** is trained on SM background simulation to evaluate its density

  $s(x) = \log p - \log p_{\max}$ is the anomaly score.
- Tightening the score cut (90% → 99% → 99.9% rejection) pushes selected events to **high m_T** — the flow isolates the extreme tails.
- **No excess.** Model-independent limits set across ten discovery regions, plus limits on vector-like leptons and other benchmarks.





02

PART 02

Weakly supervised methods

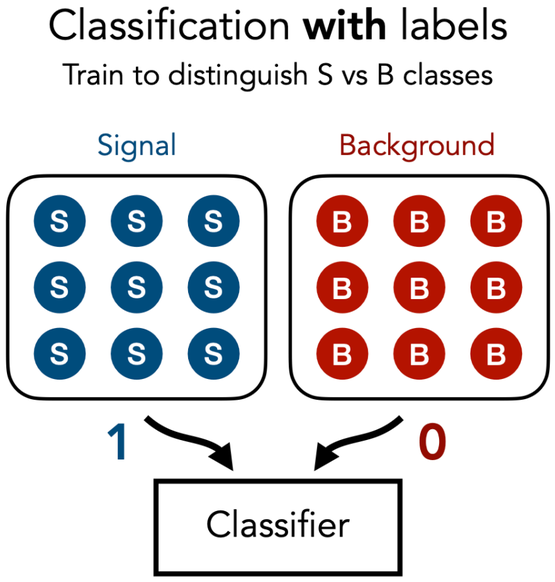
Figures curtesy of A. Badea

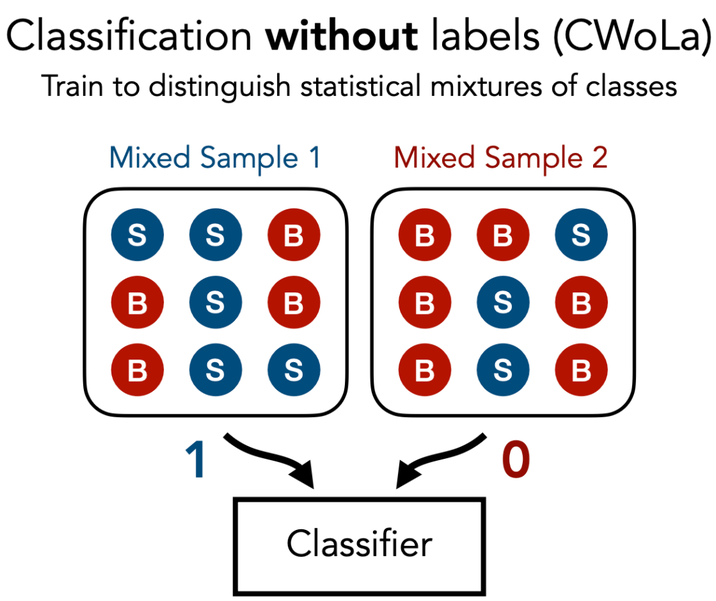
Classification Without Labels (CWoLa)

Train a classifier to separate two *mixed* samples that differ only in their signal fraction yields

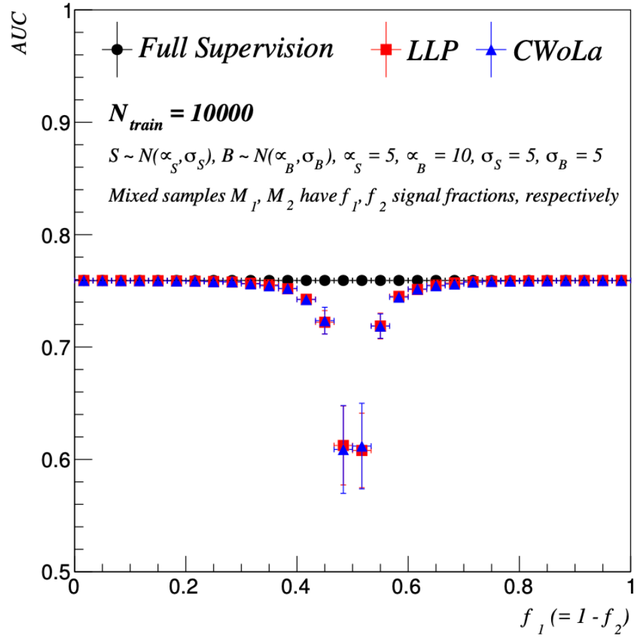
The CWoLa theorem: The optimal **CWoLa classifier** is also the optimal **fully supervised classifier**

- In practice, requires sufficient events and mixed samples with different compositions





arxiv:1708.02949



Figures curtesy of A. Badea

WEAKLY SUPERVISED AD

The bump hunt meets CWoLa

The two samples come **for free from the mass spectrum**:

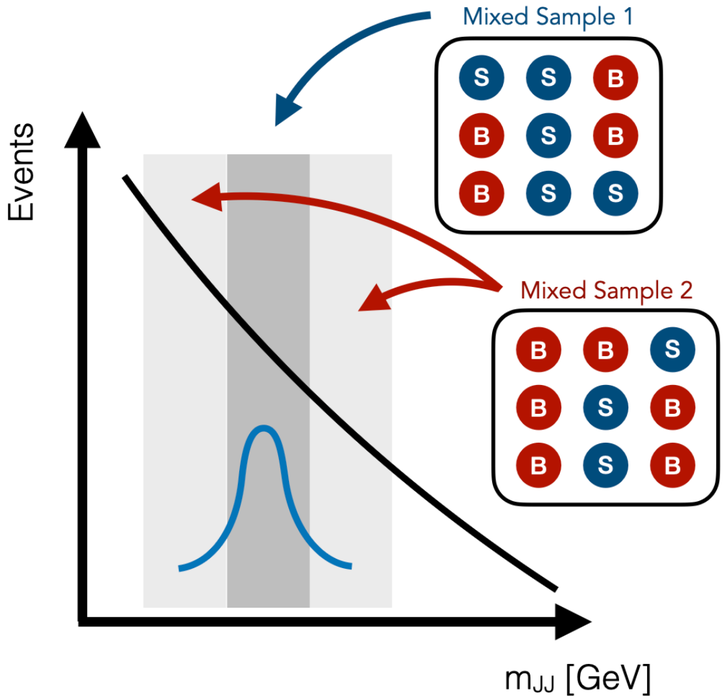
- A signal-region window (enriched if a resonance exists) vs. its sidebands (nearly pure background).

If there is no signal:

- SR and SB look identical → the classifier does nothing → no fake bump.

If there is signal:

- The classifier learns to enrich it → the bump grows in significance.



In practice

CWoLa requires the training of many binary classifiers — (at least) one per signal mass window hypothesis

Care must taken to avoid sculpting the invariant mass distribution with the NN cuts

ATLAS | Phys. Rev. Lett. 125 (2020) 131801 · arXiv:2005.02983

A weakly-supervised dijet resonance search

CWoLa applied to **ATLAS dijet resonance search** for $A \to BC$

- m_1 , m_2 jet masses the only ML features

Signal vs. sideband regions defined in m_{JJ}

- A classifier is trained to separate it from its two neighboring sideband regions

If signal is present, the network learns to tag it and enhances a bump

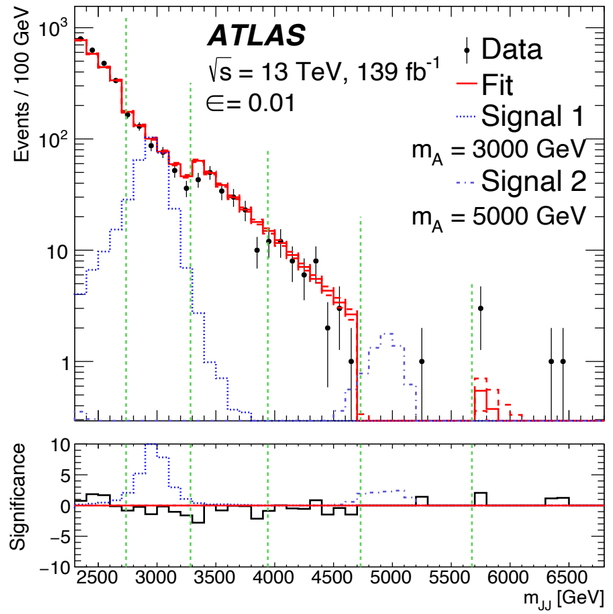
- If absent, the tagging is random and the m_{JJ} spectrum stays smooth.

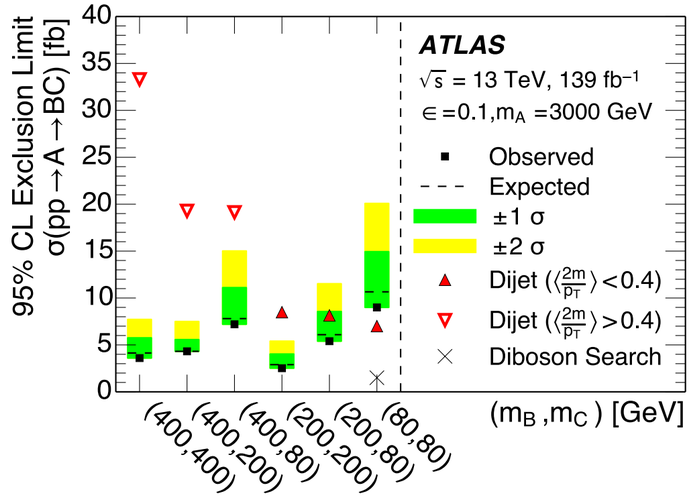
Decorrelation guards against fake bumps

- Jet masses are remapped to uniform via their empirical CDF and sidebands equally weighted

No significant excess found

- Limits up to 10× more sensitive than the inclusive dijet search.





The correlation problem

CWoLa only works if the classifier's features are **nearly independent of the variable** that defines the regions.

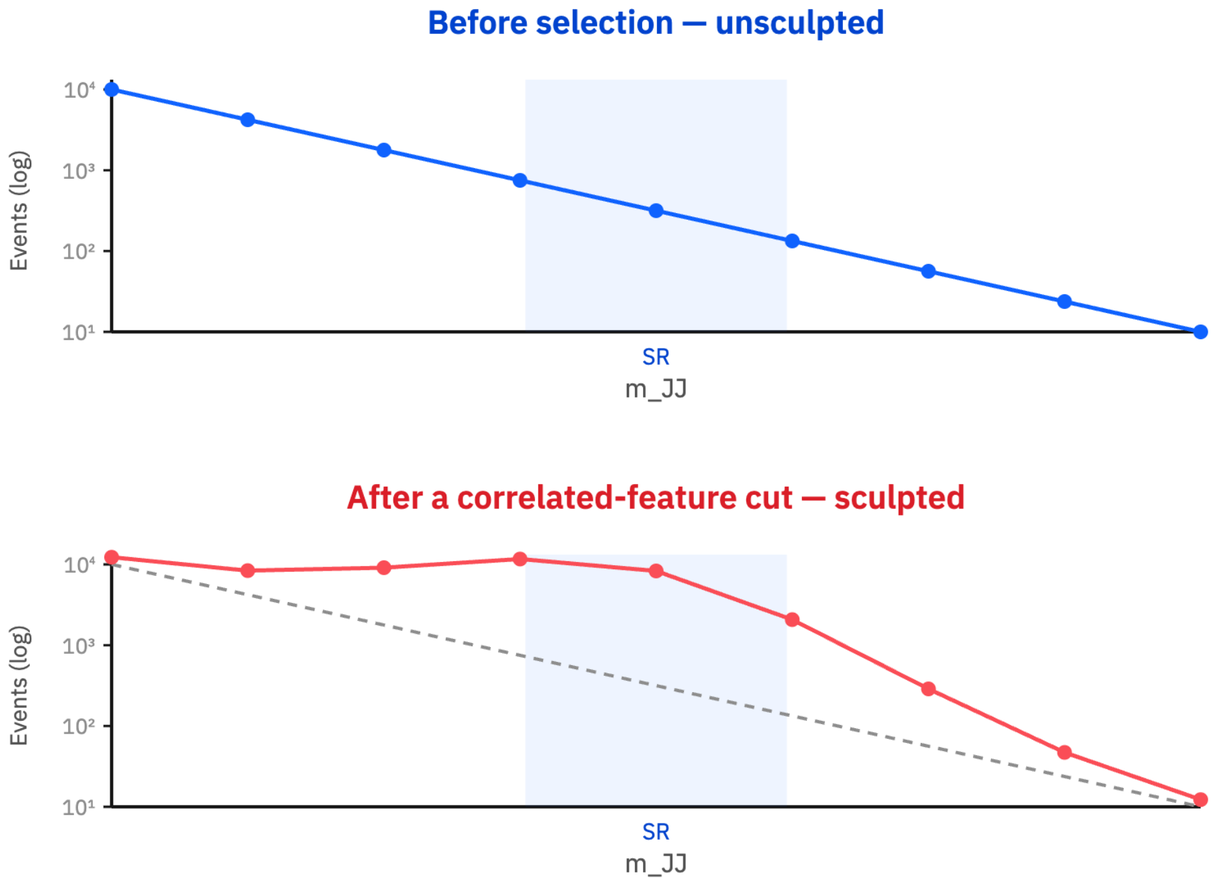
If a feature *is* correlated with mass, the classifier takes the shortcut: it learns **"am I in the SR window?"** instead of **"am I signal-like?"**

Cutting on that classifier then **sculpts a fake bump** out of pure background: **a spurious signal**.

The workaround

Use only decorrelated features.

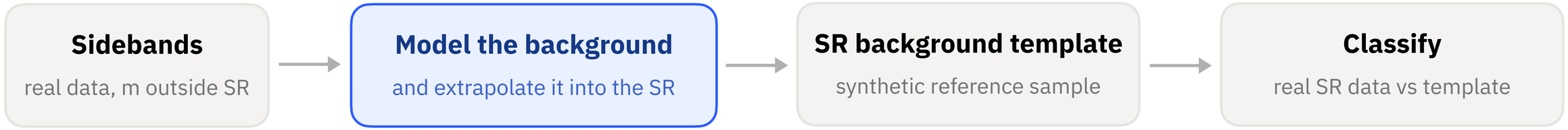
Safe, but it reduces the number of usable features — and therefore the sensitivity — of the search.



Fixing CWoLa's correlation problem: CATHODE, SALAD, CURTAINs

Limitations of CWoLA:

- Needs features independent of m. Need a background estimate in the SR that survives correlated features.
- Does not tell us anything about the background template itself



Three related methods address these limitations with different approaches in the highlighted step:

CATHODE

A **normalizing flow** learns the sideband density, is interpolated to the SR mass, and *sampled* to make the template.

CURTAINs

A flow **morphs** sideband data directly into the SR — a "flows-for-flows" transport. Entirely data-driven.

SALAD

Reweights simulation to match data in the sidebands, then applies the weights in the SR. Uses a simulated prior.

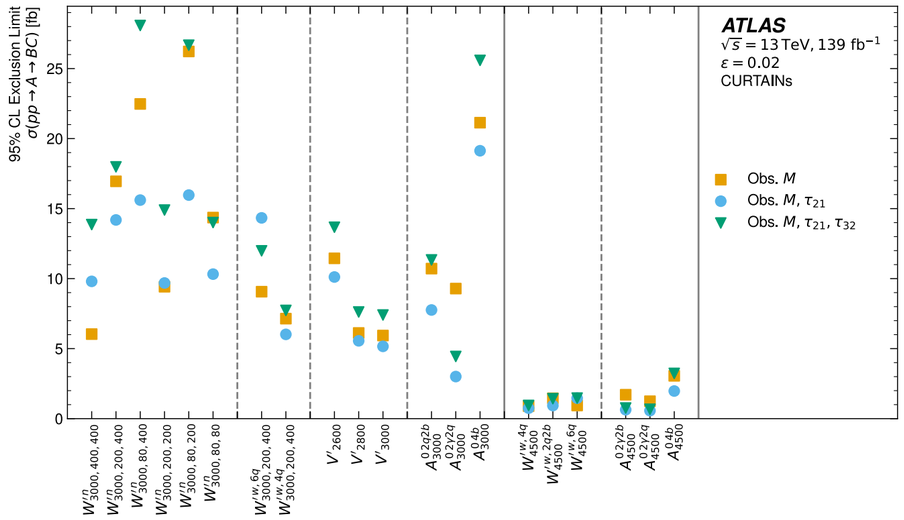
ATLAS | Phys. Rev. D 112 (2025) 072009 · arXiv:2502.09770

A dijet search with a learned background

A weakly-supervised search for a narrow resonance A → BC in the dijet final state.

- Upgrades the previous search by learning the background inside each signal region instead of borrowing the raw sidebands.

1. Scan signal-region windows in the dijet mass m_{JJ}
2. Build a background-only reference sample for the SR with either SALAD or CURTAINs
3. Both correct the m_{JJ} correlation — so the classifier can now add jet substructure (τ_{21}, τ_{32}) on top of the two jet masses.
4. Train the CWoLa classifier to separate SR data from the reference, cut on its output, and look for a bump in m_{JJ}.



The payoff

A learned background removes the correlation problem, unlocking more features and reaching a broader range of signals than the raw-sideband approach.

CMS | Rep. Prog. Phys. 88 (2025) 067802 · arXiv:2412.03747

Which method is right for you?

CMS employed AD in recent search for dijet resonances

- Anomaly tag substructure of the jets

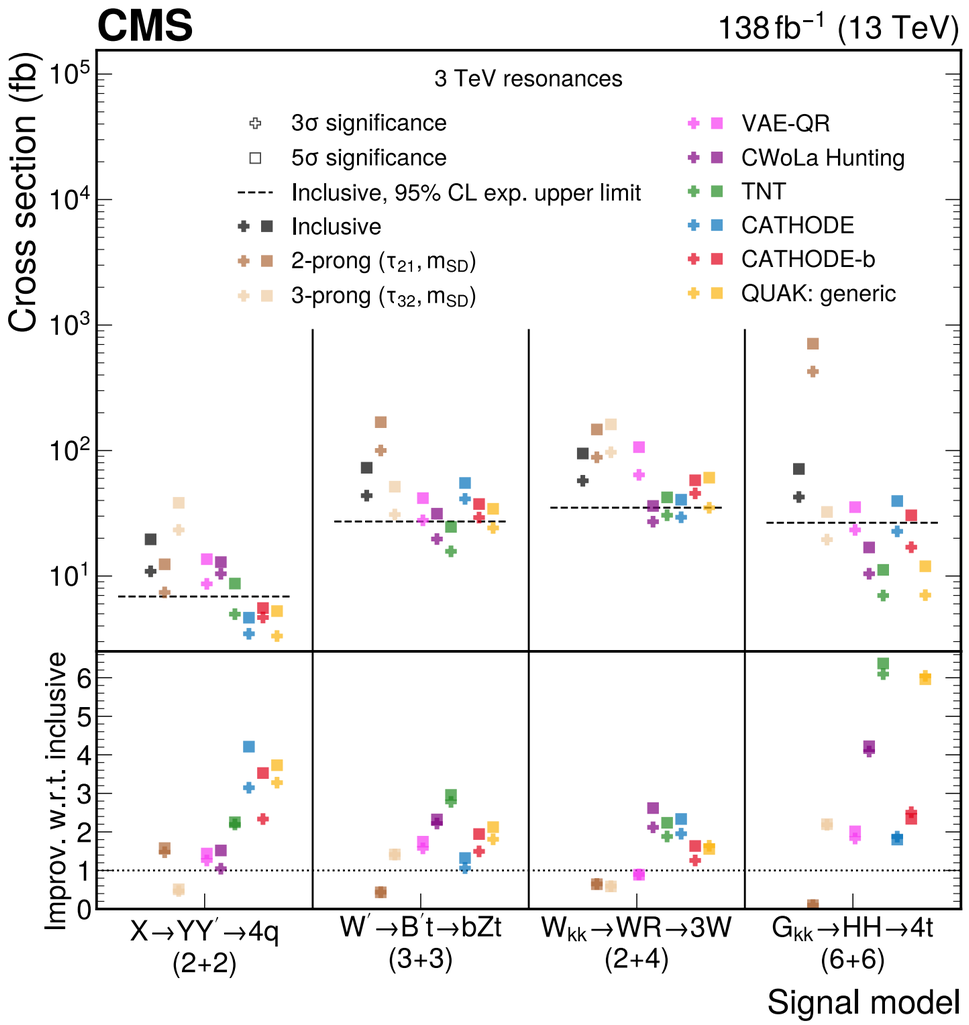
Compared multiple different anomaly methods

- Plot shows “What xsec do I need for $3/5\,\sigma$ of signal?”
- Up to factor of 7 gain in discovery sensitivity!

The Lesson

No one universal, ‘best’ method!

Trying multiple approaches leads to broader sensitivity



Summary

Method	What it learns	Anomaly score	Best at	ATLAS example
Autoencoder	Compressed reconstruction of "normal"	$\|x - \hat{x}\|^2$	Outliers	Two-body resonance search PRL 2024
VAE	A probabilistic latent space	recon · KL · −ELBO	Outliers	Y→XH jet tagger PRD 2023
Normalizing flow	The exact density p(x)	$\log p(x)$	Outliers	Multilepton search EPJC 2026
Weak · CWoLa	The difference between two data samples	classifier output	**Over-densities**	Dijet search PRL 2020

SUMMARY

What to take away

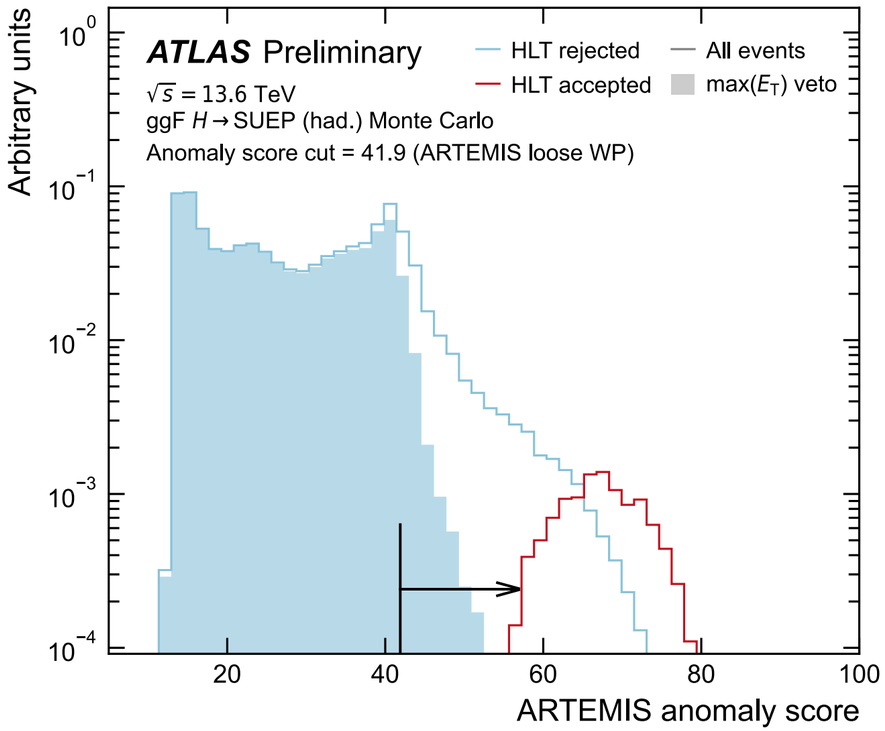
Unsupervised learning is a powerful tool to improve the sensitivity of our search program without sacrificing generality

There is no “one size fits all” solution for anomaly detection! Must try several approaches to maximize sensitivity to BSM physics

The examples given today are only a small subset of how unsupervised learning is being applied to LHC physics

- Anomaly detection in the trigger system to identify unusual events in real time: ATLAS Trigger public plots
- Anomaly detection for data quality monitoring to automate identifying detector issues: ATL-DAPR-PUB-2024-002/

We are currently experiencing an AD revolution — expect many more results to come in the near future!



SUMMARY

References

ATLAS ANALYSES

Two-body resonance search with an autoencoder

Phys. Rev. Lett. 132 (2024) 081801 · arXiv:2307.01612

Y→XH anomaly jet tagger (VRNN / VAE)

Phys. Rev. D 108 (2023) 052009 · arXiv:2306.03637

Multilepton anomaly search (normalizing flow)

Eur. Phys. J. C 86 (2026) 247 · arXiv:2508.19778

Weakly-supervised dijet search (CWoLa)

Phys. Rev. Lett. 125 (2020) 131801 · arXiv:2005.02983

Weakly-supervised dijet search (CWoLa · SALAD · CURTAINs)

Phys. Rev. D 112 (2025) 072009 · arXiv:2502.09770

CMS ANALYSES

Weakly-supervised dijet search (CWoLa · CATHODE · + others)

Rep. Prog. Phys. 88 (2025) 067802 · arXiv:2412.03747

FOUNDATIONAL METHODS

Kingma & Welling — *Auto-Encoding Variational Bayes*

arXiv:1312.6114

Dinh, Sohl-Dickstein & Bengio — *Density estimation using Real NVP*

arXiv:1605.08803

Metodiev, Nachman & Thaler — *Classification Without Labels*

arXiv:1708.02949

Hallin et al. — *CATHODE*

arXiv:2109.00546

Raine et al. — *CURTAINs*

arXiv:2203.09470

Andreassen et al. — *SALAD*

arXiv:2001.05001

THANK YOU

Questions?