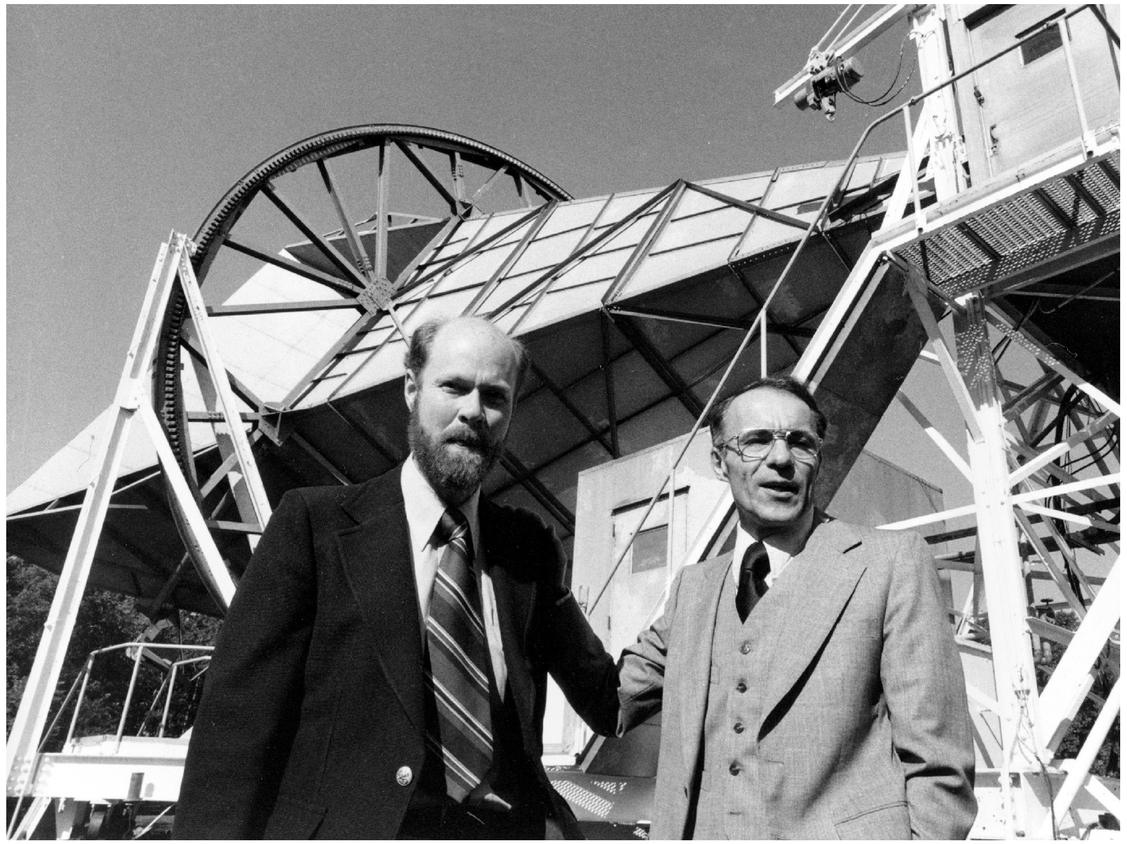
Intro to Anomaly Detection in Particle Physics

Melissa Quinnan (UC San Diego)

Machine Learning for Fundamental Physics 2026
June 3, 2026

melissa.quinnan@cern.ch

Unexpected Discoveries

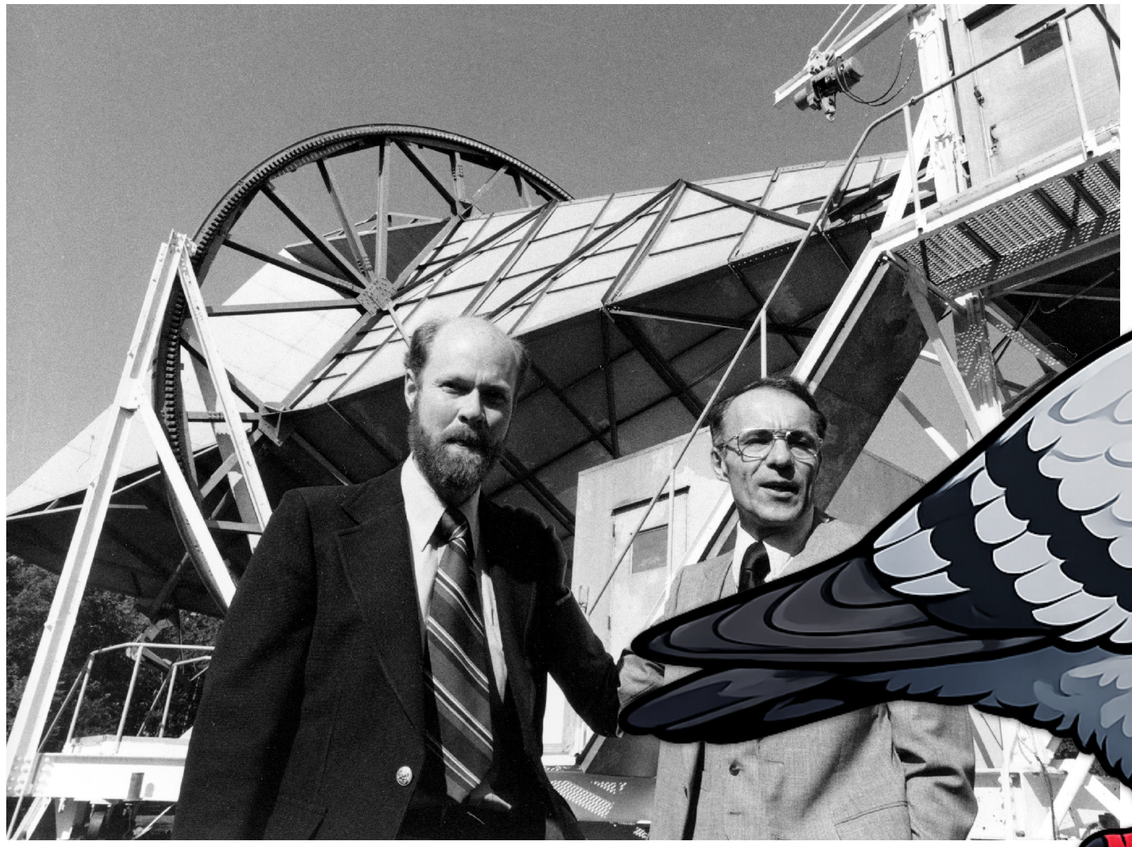


Robert Wilson Arno Penzias

Horn Antenna

- Want to detect radio waves from satellites → as little noise as possible
- →Low, steady, irreducible noise that was evenly spread over the sky, and present day and night

Unexpected Discoveries



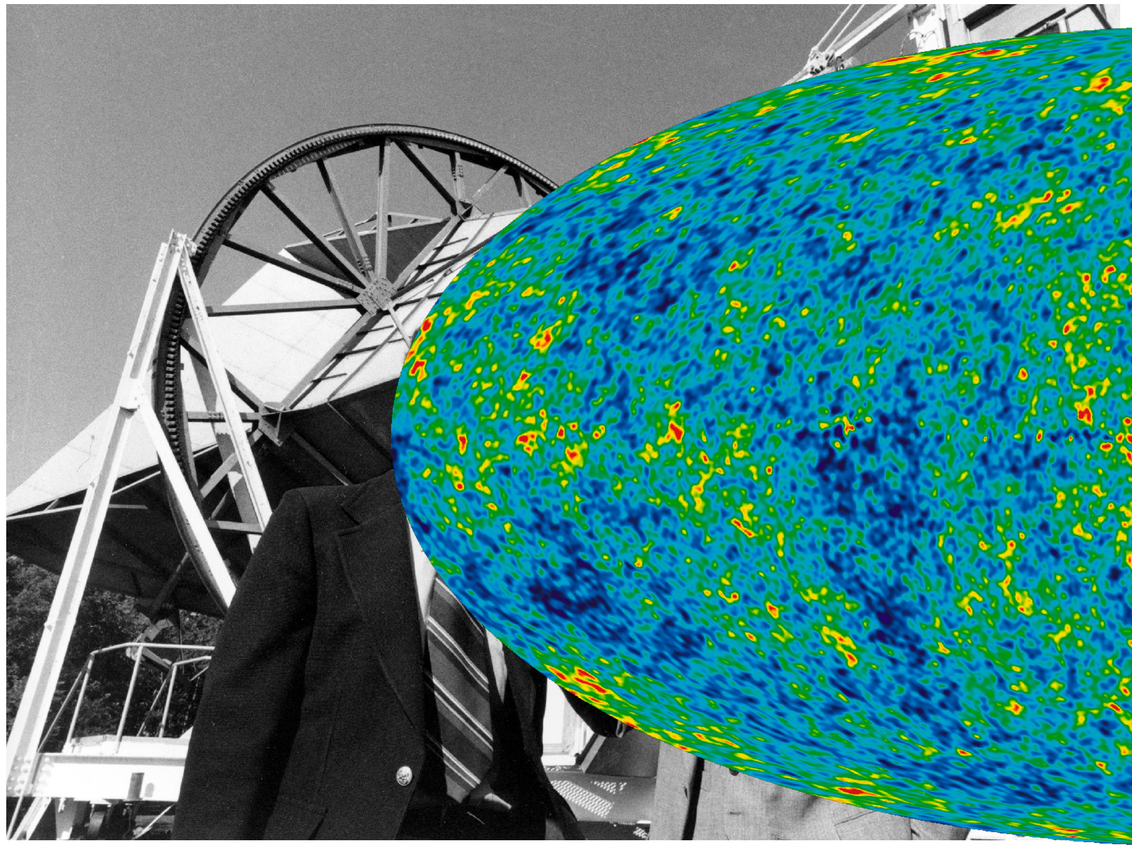
Robert Wilson Arno Penzias

Horn Antenna

detect radio waves from
ellites → as little noise as possible

→Low, steady, irreducible noise that
evenly spread over the sky, and
present day and night

Unexpected Discoveries



Robert Wilson **Arno Penzias**

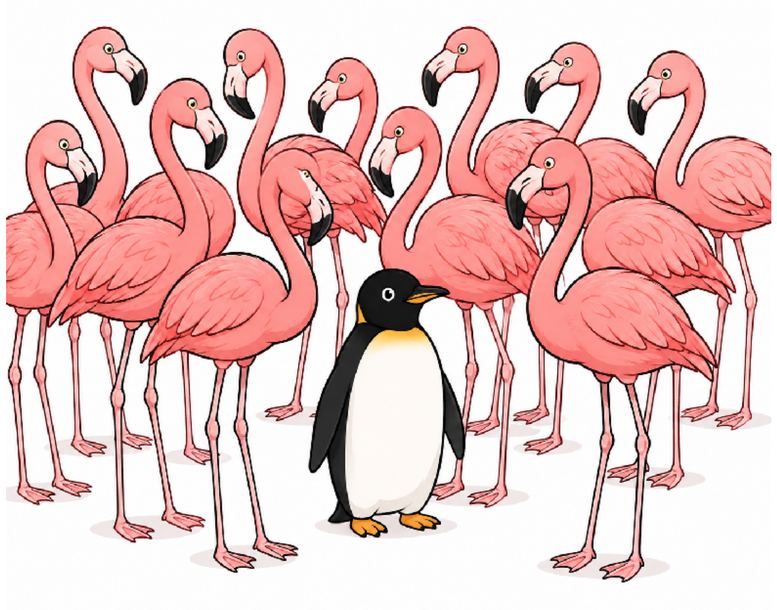
es from
oise as possible
y, irreducible noise that
was evenly spread over the sky, and
present day and night

- New physics may be hiding where we least expect it!
  - Or you fix defects in your detector!

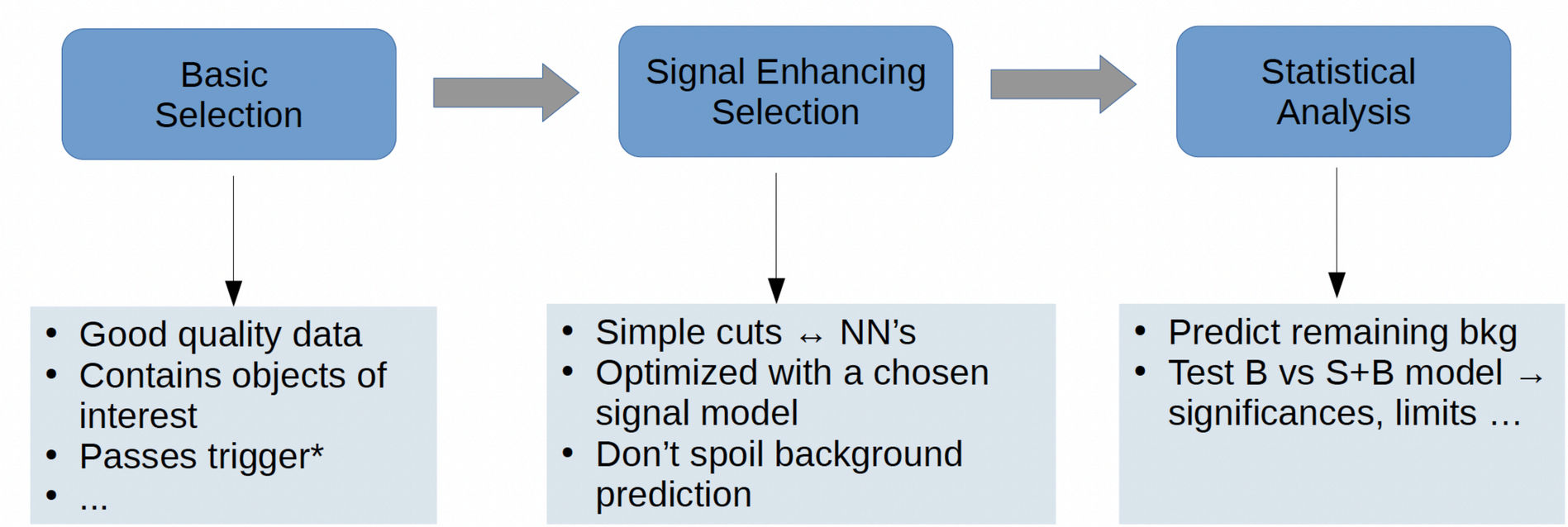
Introduction

- **What is anomaly detection?**
  - "Finding something interesting without specifying exactly what you are looking for"
- **Why would you want to do it?**
  - Many possible signals in your data→ Cannot search for them all one by one
  - Science is full of many **unexpected discoveries!**

For much more detail see Oz's slides from last year!

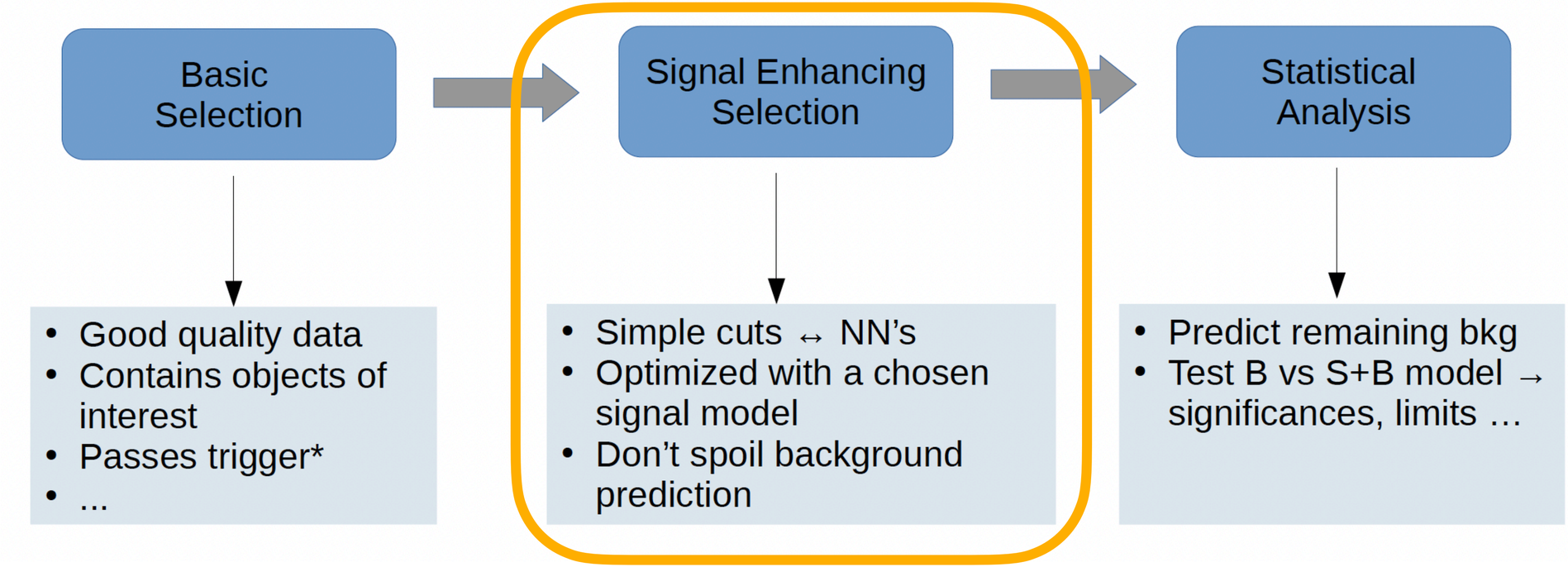


HEP Data Analysis



HEP Data Analysis

*Read about the Neyman-Pearson lemma if you are unfamiliar!



Optimal signal vs. background discriminant is the Likelihood Ratio*:

*Supervised classifiers **directly** approximate the Likelihood Ratio*

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

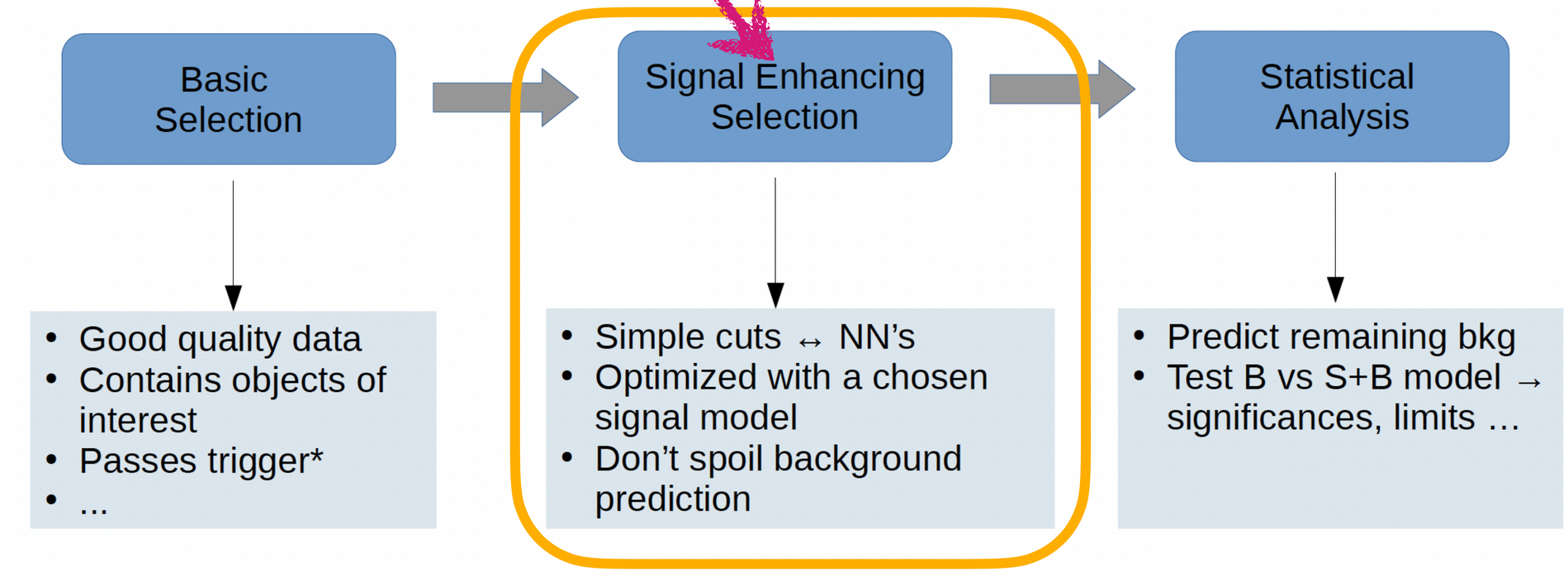
P_s ← Prob. distribution of signal

P_B ← Prob. distribution of background

HEP Data Analysis

Can we do this without specifying a signal model?

*Read about the Neyman-Pearson lemma if you are unfamiliar!



Optimal signal vs. background discriminant is the Likelihood Ratio*:

*Unsupervised classifiers **indirectly** approximate the Likelihood Ratio*

$$L_{S/B}(X) = \frac{P_S}{P_B}$$

Unknown!

~~Prob. distribution of signal~~

Prob. distribution of background

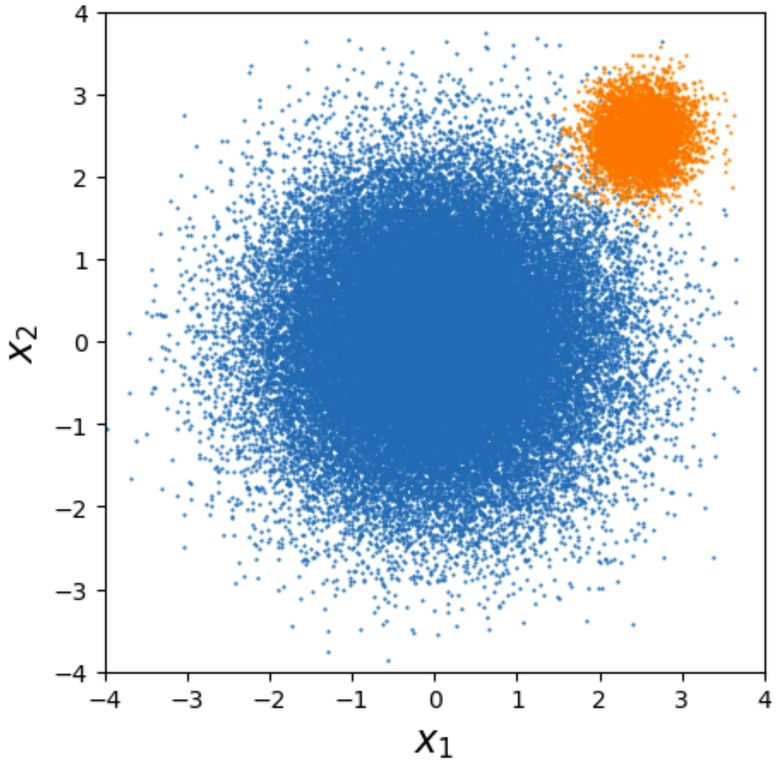
Anomaly Detection

- Two strategies to approximate $L_{S/B}$ without P_s:

    1. **Outlier Detection:** Learn P_B, take anomaly score as $1/P_B$
    2. **Data-driven likelihood ratio:** Leverage localization of signal to learn $L_{S/B}$ from data

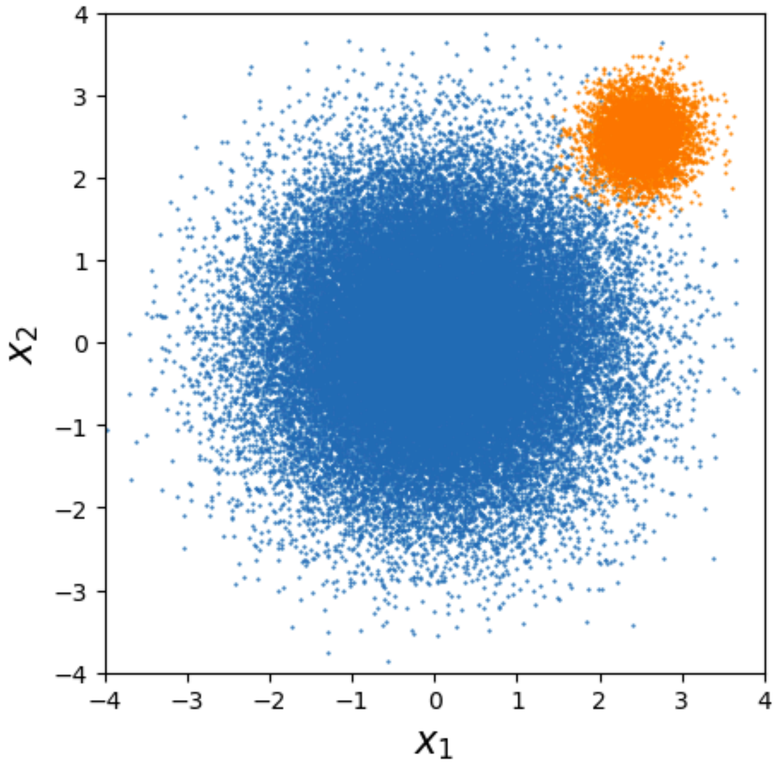
Outlier Detection

- Objective: **Learn** $P_B(X)$, take anomaly score as $1/P_B$
  - Low P_B → **anomalous!**
- Typically: have lots of background events but **don't know** P_B **explicitly**...
- What can we do?



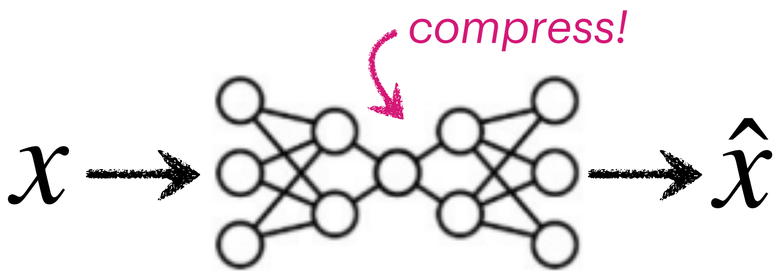
Outlier Detection

- Objective: **Learn** $P_B(X)$, take anomaly score as $1/P_B$
  - Low P_B → **anomalous!**
- Typically: have lots of background events but **don't know** P_B **explicitly**...
- What can we do?
  - Simple tools to estimate bkg pdf (KDE, GP, ...) ← For complex high dim. data hard to explicitly model P_B :(
  - Generative models to estimate P_B (VAE's, Normalizing flows, some types of diffusion/flow matching models...) ← This morning
  - **Autoencoders** to learn a *proxy* of P_B ← Let's talk about it!



Autoencoders

- Learn to compress/reconstruct data to/from a low dimensional representation (**latent space**)



$$\text{Loss} = ||x - \hat{x}||^2$$

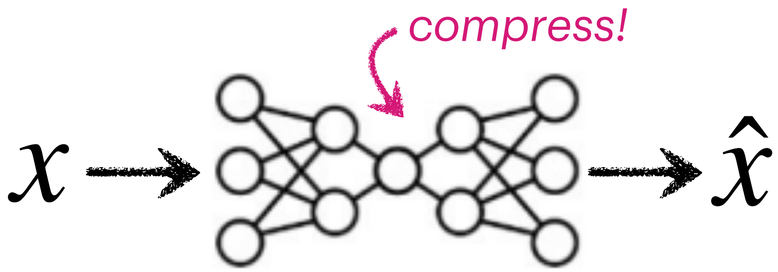
Reconstruction loss

Training dataset

1808.08979
1808.08992

Autoencoders

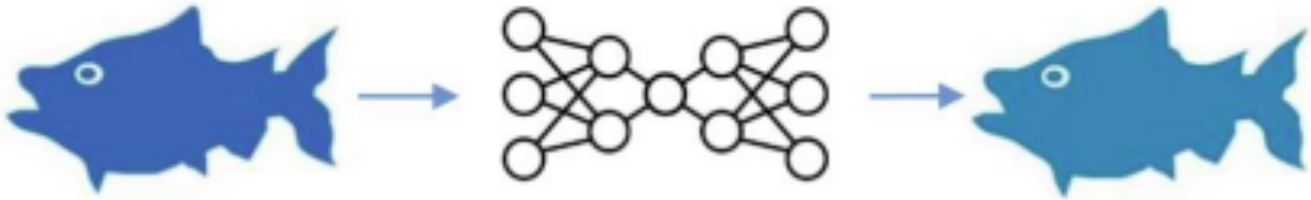
- Learn to compress/reconstruct data to/from a low dimensional representation (**latent space**)



$$\text{Loss} = ||x - \hat{x}||^2$$

Reconstruction loss

Training dataset



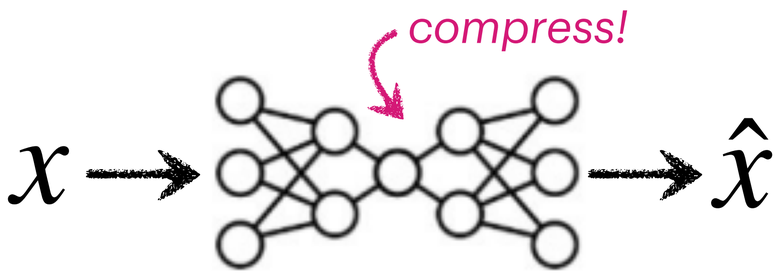
← Good reco, low loss

1808.08979
1808.08992

Autoencoders

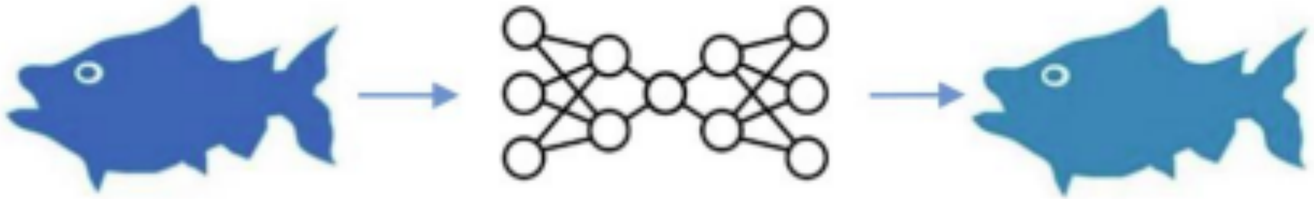
- Learn to compress/reconstruct data to/from a low dimensional representation (**latent space**)

Training dataset

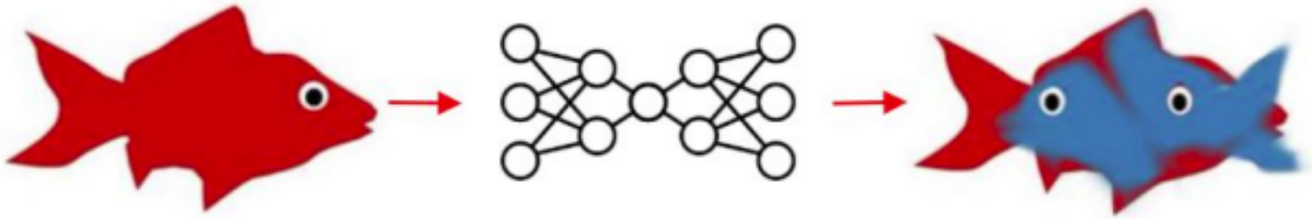


$$\text{Loss} = ||x - \hat{x}||^2$$

Reconstruction loss



Good reco, low loss

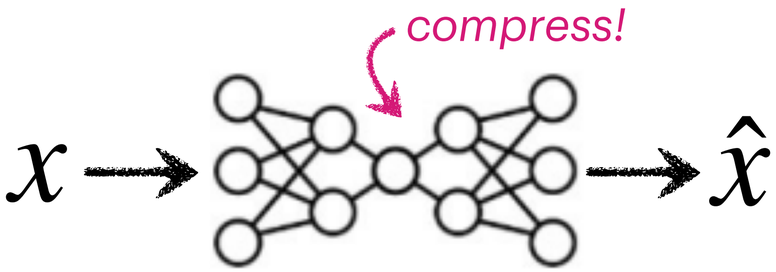


Bad reco, high loss

1808.08979
1808.08992

Autoencoders

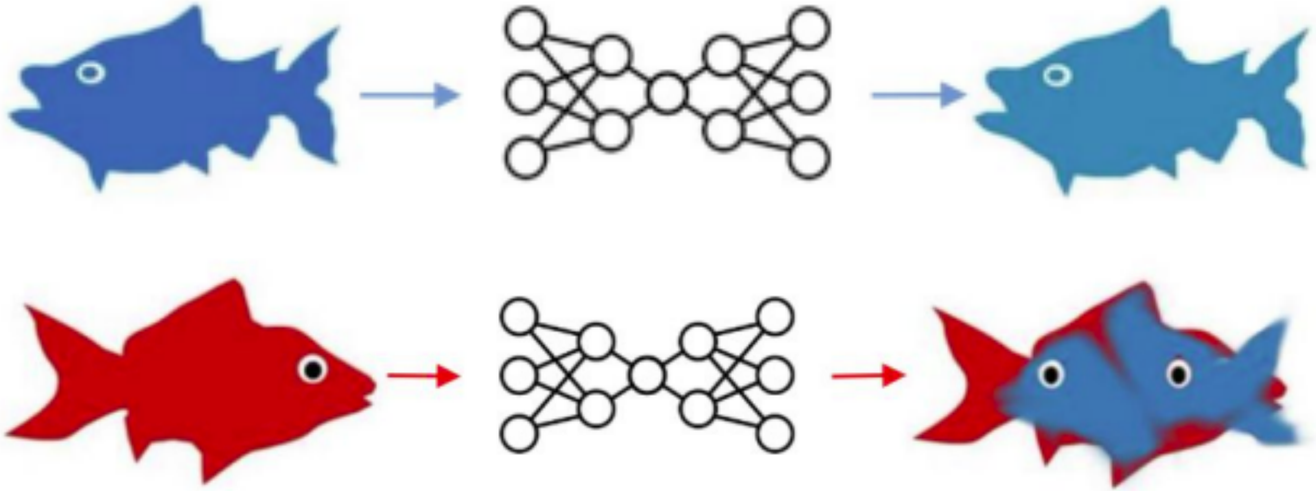
- Learn to compress/reconstruct data to/from a low dimensional representation (**latent space**)



$$\text{Loss} = ||x - \hat{x}||^2$$

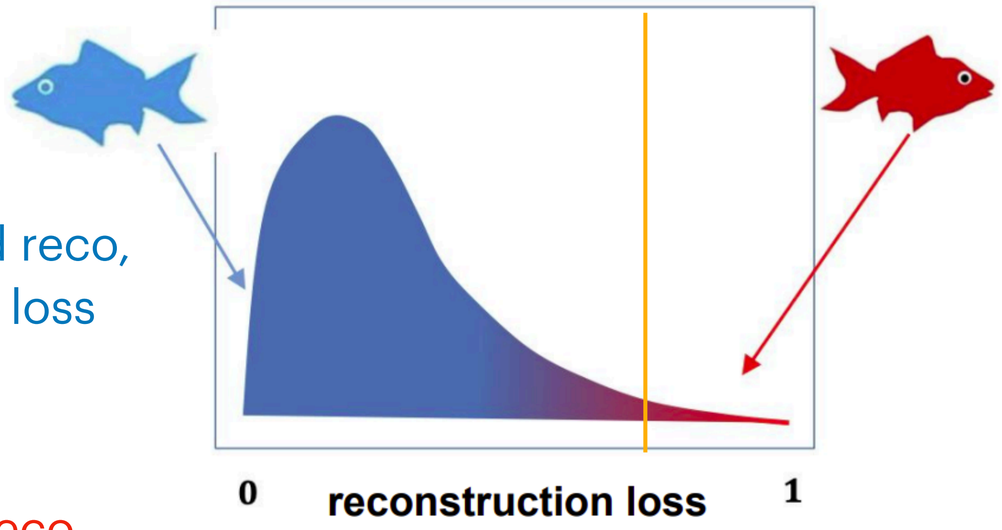
Reconstruction loss

Training dataset



← Good reco, low loss

← Bad reco, high loss

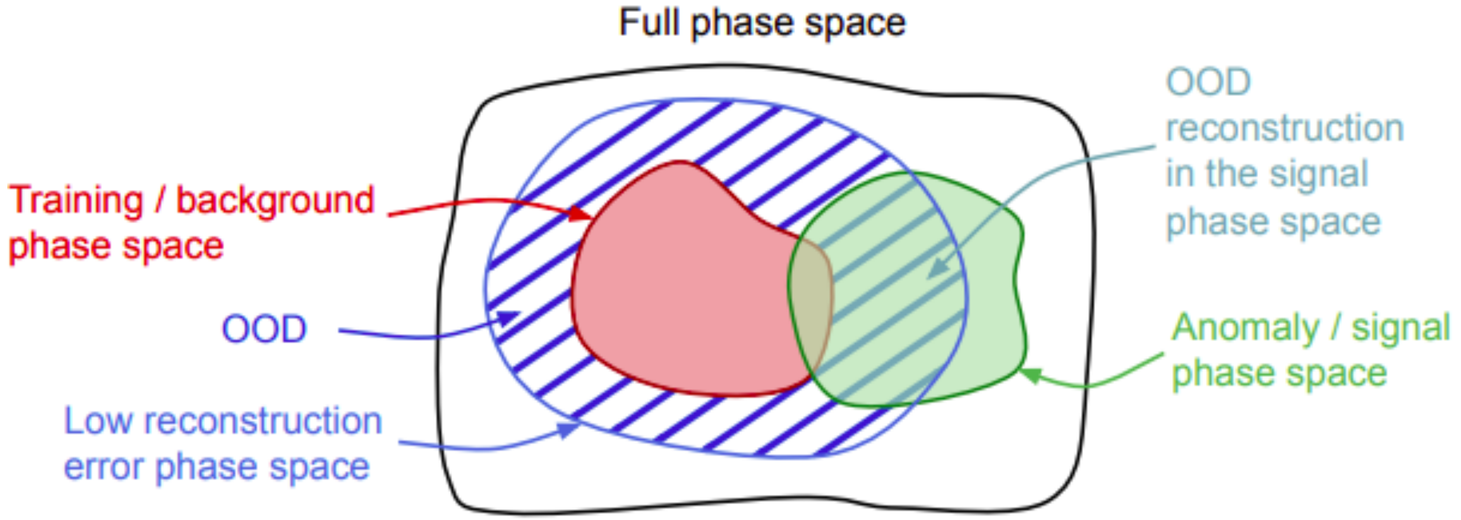


1808.08979
1808.08992

Autoencoder Challenges →Do not directly model $P_B(X)$:

Problem 1: They are biased

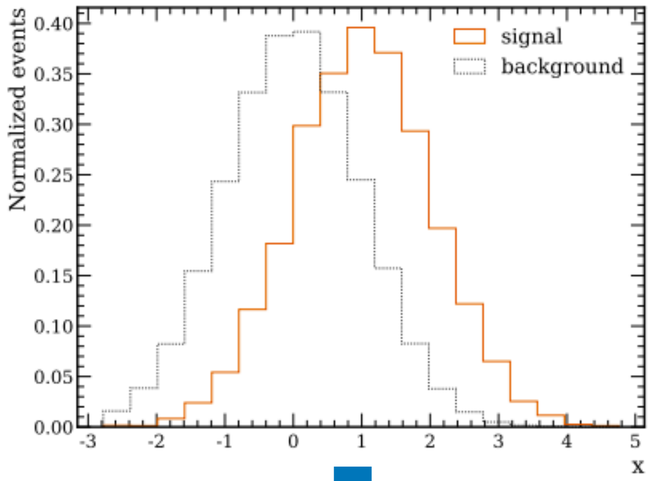
- Complexity bias→ more 'complex' data harder to compress, seen as more anomalous
- Over-generalization→ AE can reconstruct things well even outside training phase space because no penalty to do this



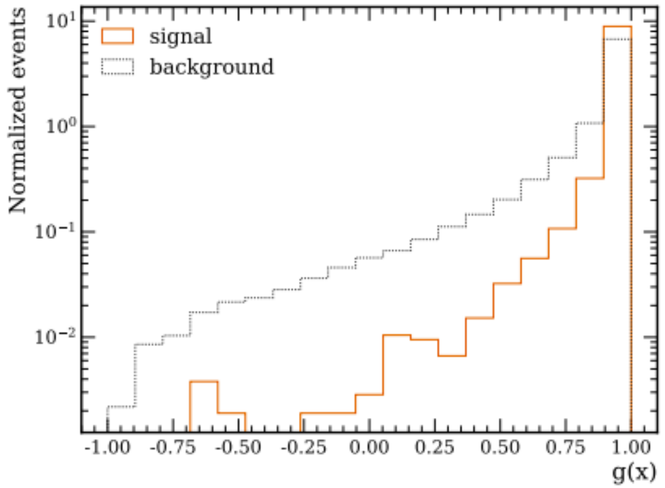
Normalized autoencoders and generative models can help here

Autoencoder Challenges

$y = f(x)$



$y = tanh(x + 2)$



Problem 2: Coordinate invariance

- Probability densities (like $P_B(X)$) not invariant under coordinate transformations
  - 'Anomalous' is a coordinate system dependent statement!
- Data representation is an inductive bias for anomalies

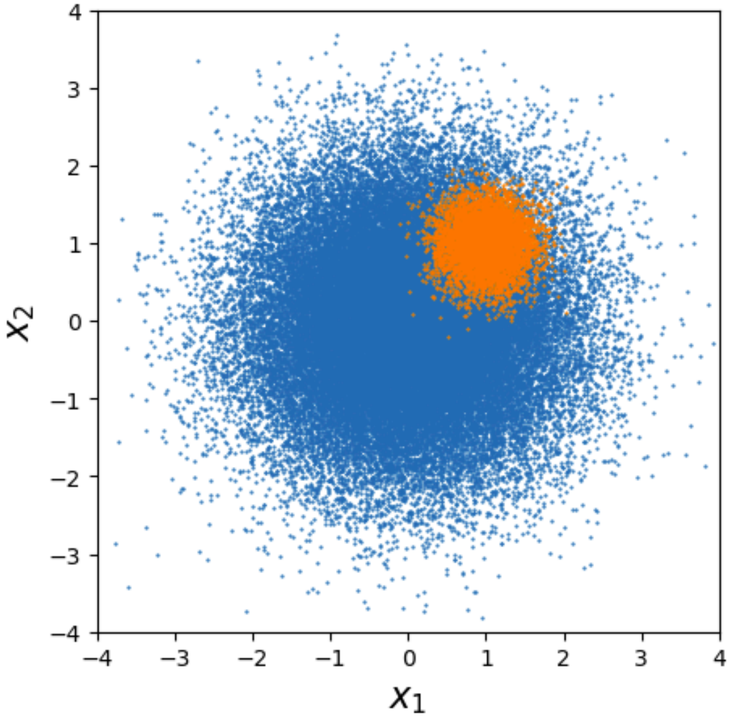
Unavoidable limitation of using only $P_B(X)$...

Probability densities are not invariant:

$$p_y(y) = p_x(f^{-1}(y)) \left| \frac{d}{dy} f^{-1}(y) \right|$$

Data-Driven Likelihood Ratio

- Objective: Learn approximation of $L_{S/B}$
- Can find in-distribution signals:
  - Often signals are **within** the background distribution rather than full outliers
  - What makes them anomalous is a **cluster of similar events**
  - →Cannot be found with outlier detection methods!
- Likelihood ratio is coordinate invariant
- Outlier methods have an upper bound on sensitivity- they never learn P_S

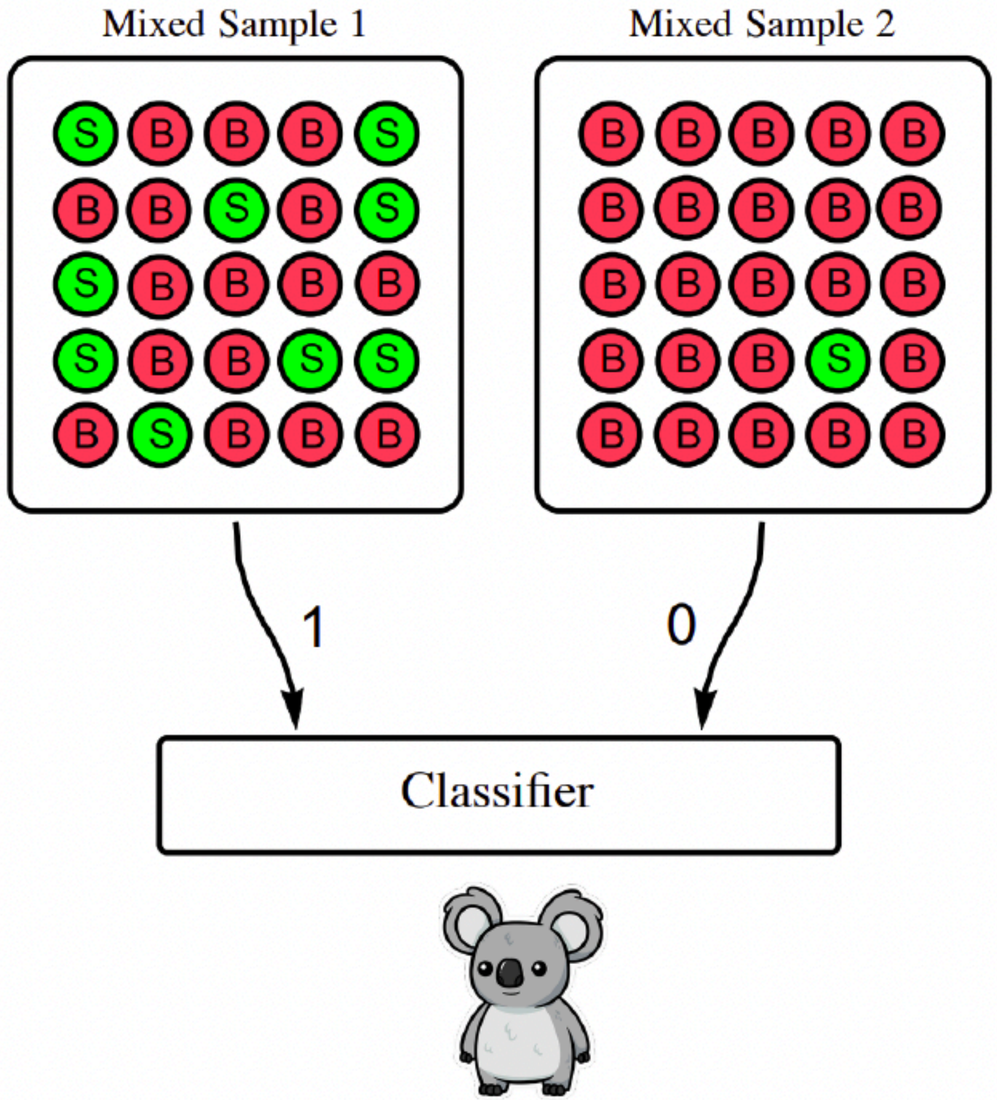


→But how do you learn $L_{S/B}$ from unlabeled data?

Learning the Likelihood Ratio

1708.02949

- Given two samples of mixed signal and background:
  - The optimal classifier for distinguishing these mixed samples is also $L_{S/B}$*!
  - →**Training a classifier with these mixed samples will mimic a supervised classifier!**
  - What matters is that the signal to background fraction is **different** in both samples



(*Assuming the background has the same underlying distribution)

*Proof in backup

Weak Supervision

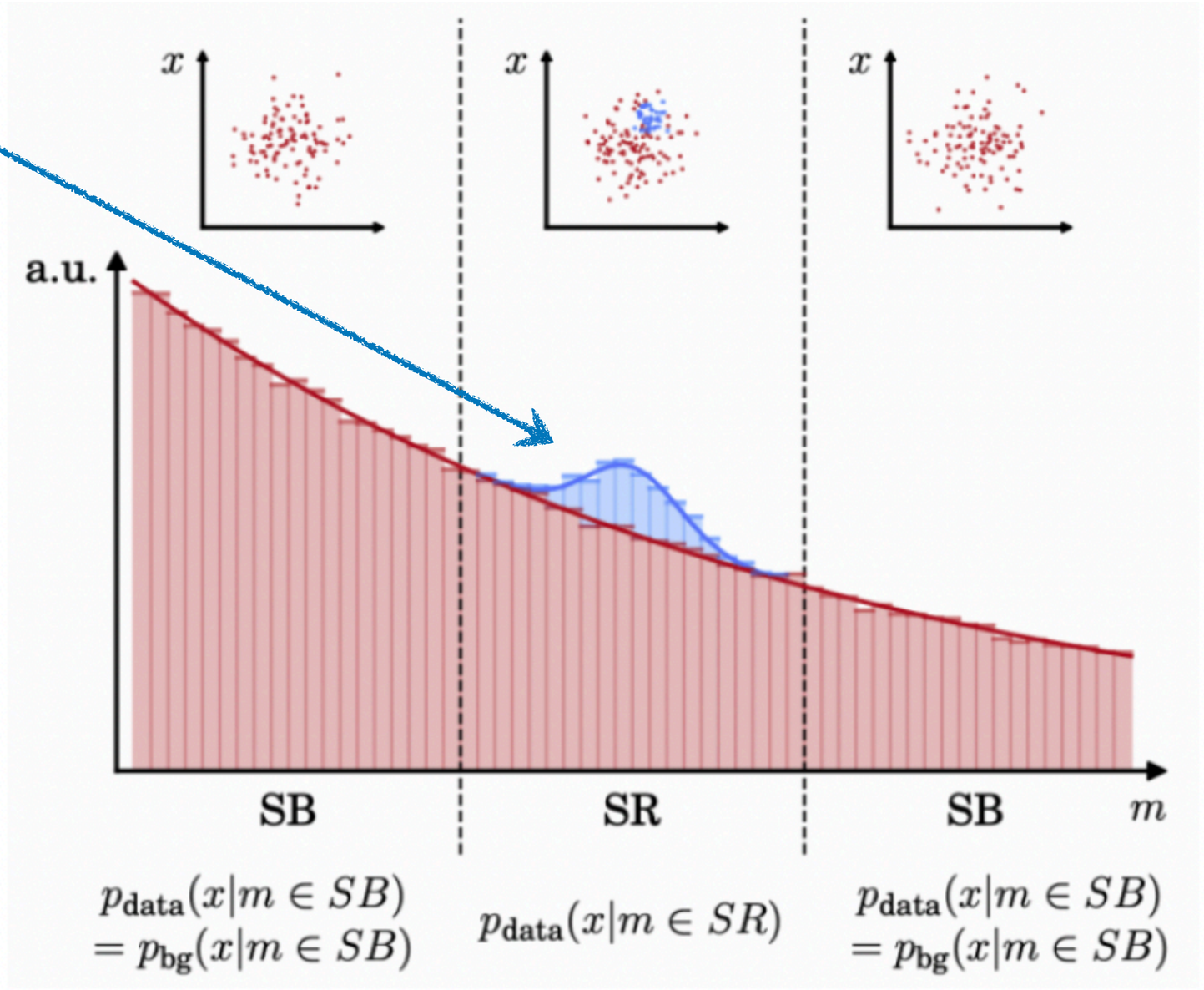
- This method of training between mixed samples is called **weak supervision** (or Classification Without Labels, CWoLa)
- In practice, convergence to full supervision depends on:
  - How large the signal fraction is
  - How many training samples you have
  - How ‘distinctive’ the signal is compared to the background
- Good performance can be achieved with realistic ~1% signal fractions!

1708.02949

Weak Supervision + Bump Hunt

1902.02634

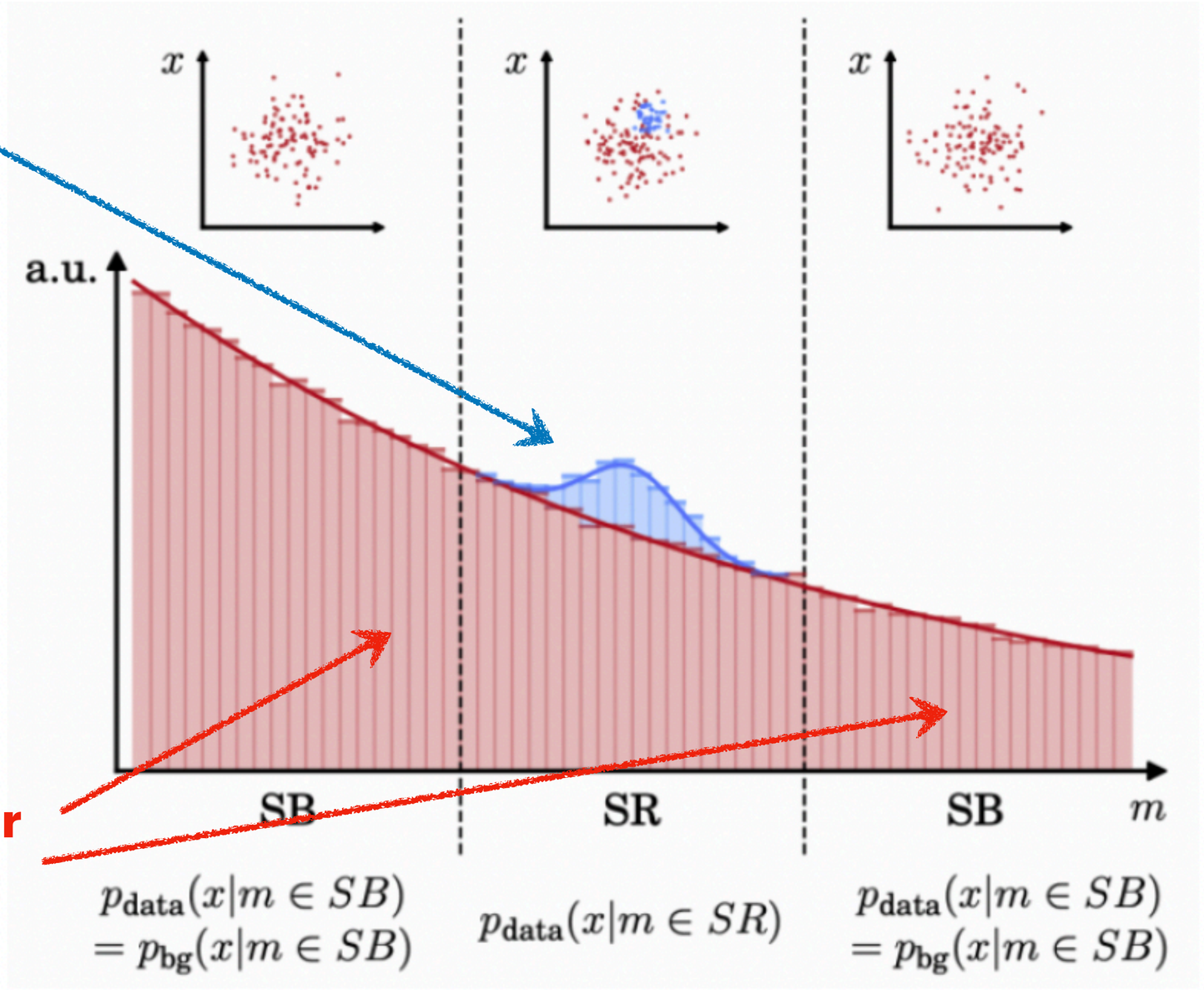
Signal narrow resonance in localized region of mass



Weak Supervision + Bump Hunt

1902.02634

Signal narrow resonance in localized region of mass

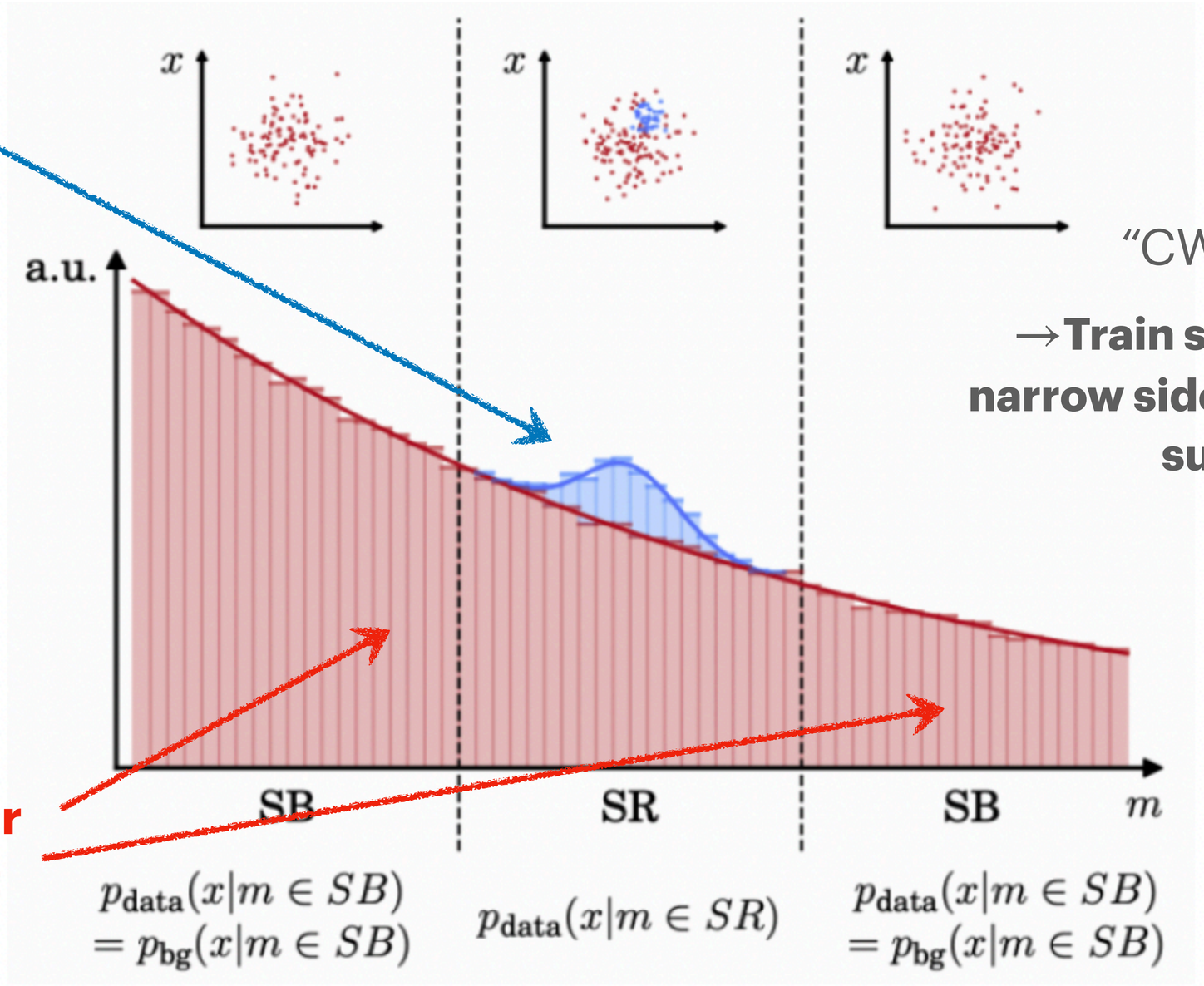


Sidebands similar background but minimal signal

Weak Supervision + Bump Hunt

1902.02634

Signal narrow resonance in localized region of mass



Sidebands similar background but minimal signal

"CWoLa Hunting"

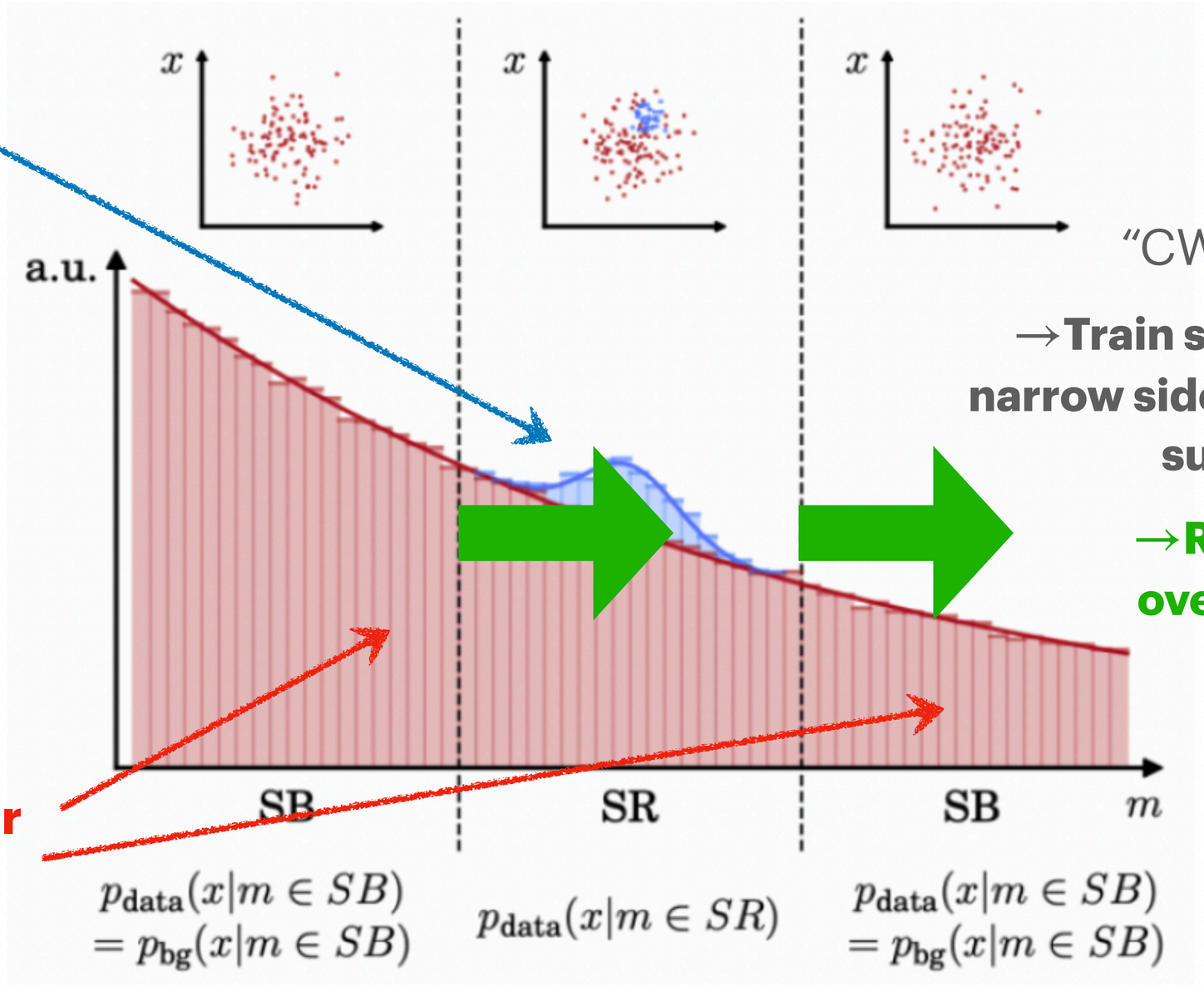
→**Train signal window vs. narrow sidebands using weak supervision**

*be careful about correlations with mass!

Weak Supervision + Bump Hunt

1902.02634

Signal narrow resonance in localized region of mass



"CWoLa Hunting"

→Train signal window vs. narrow sidebands using weak supervision

→Repeat, scanning over different mass windows

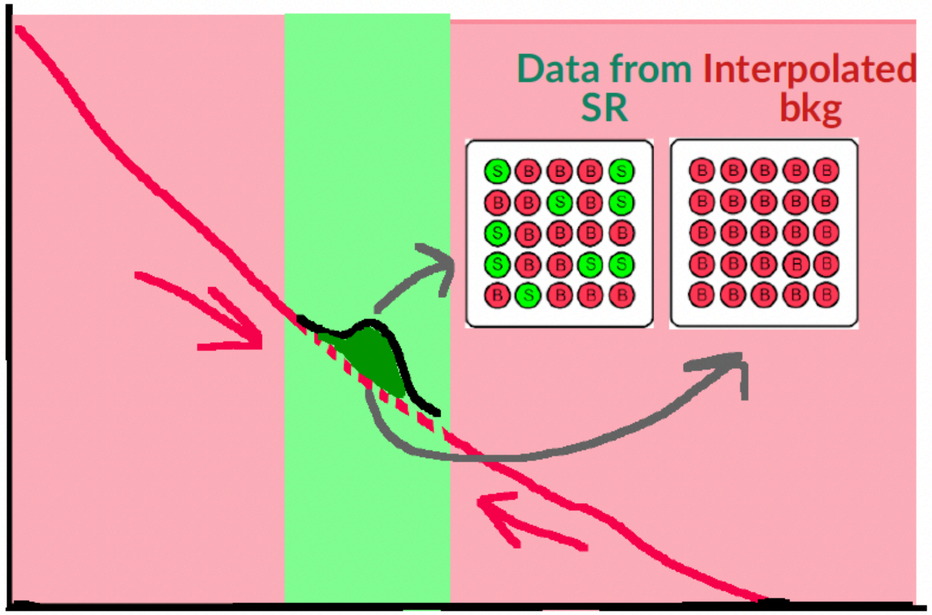
Sidebands similar background but minimal signal

*be careful about correlations with mass!

Related Methods

CATHODE
Interpolates bkg events into SR using **generative model**
Use gen. model. To construct bkg sample

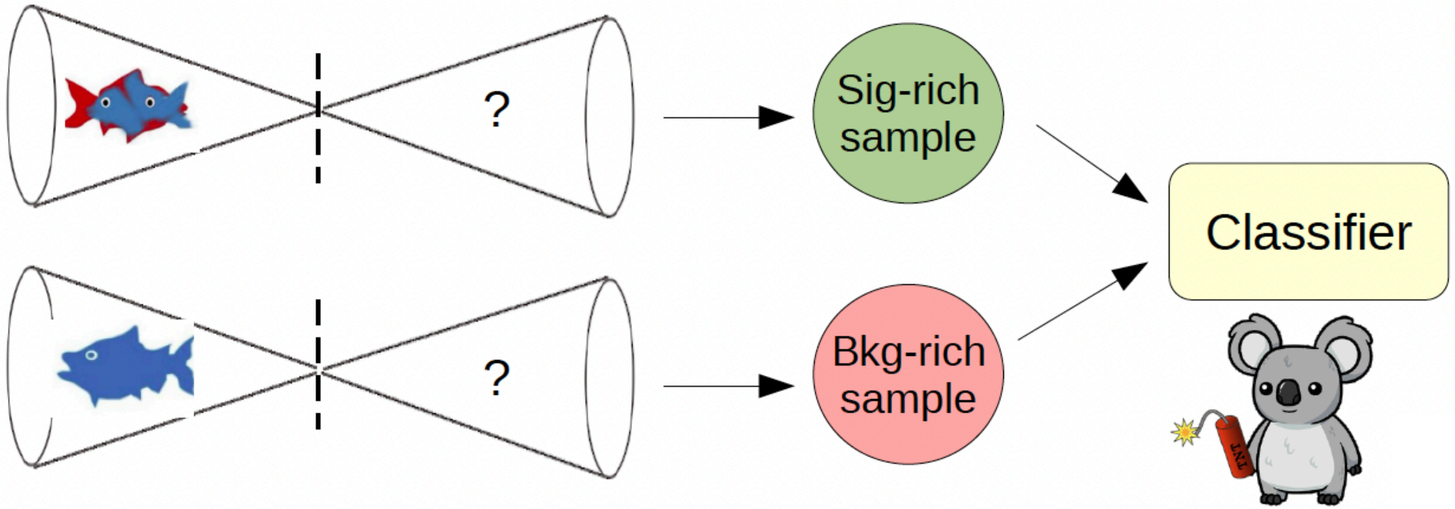
2109.00546



Other variants with different interpolation methods (~similar performance)
CURTAINS, SALAD, FETA, …

Tag N' Train
purifies samples by first tagging with AE

[**OA** & Suarez 2002.12376]



Weak Supervision Challenges

Problem 1: Weak supervision training is noisy

- At low signal fraction, works better with high level features → less model independent
- Ensembles of BDT's seem better than NN's!

Weak Supervision Challenges

Problem 1: Weak supervision training is noisy

- At low signal fraction, works better with high level features → less model independent
- Ensembles of BDT's seem better than NN's!

Problem 2: Not easy to create mixed samples

- Method assumes that the background distribution is identical in both samples
- Hard to apply this beyond bump hunts

Weak Supervision Challenges

Problem 1: Weak supervision training is noisy

- At low signal fraction, works better with high level features → less model independent
- Ensembles of BDT's seem better than NN's!

Problem 2: Not easy to create mixed samples

- Method assumes that the background distribution is identical in both samples
- Hard to apply this beyond bump hunts

Problem 3: Statistical interpretation is difficult

- Performance varying with signal strength makes limit setting/statistical interpretation painful

What should you use?

- You know the answer

What should you use?

- You know the answer → ***Pick the model that best suits your task***
- Anomaly detection is an underspecified problem → no single ‘optimal’ solution
  - Method that works brilliantly for one class of signal may be completely blind to another!
  - **Weak supervision** gives best sensitivity
  - But **outlier detection** needed if:
    - Anomalies are single instances, not clusters
    - Can’t find suitable mixed samples in data
    - Only see one event at a time (ie trigger)

See also:

- semi-supervised AD
  - (e.g. PAWS)
- exhaustive AD
  - (e.g. MuSiC)

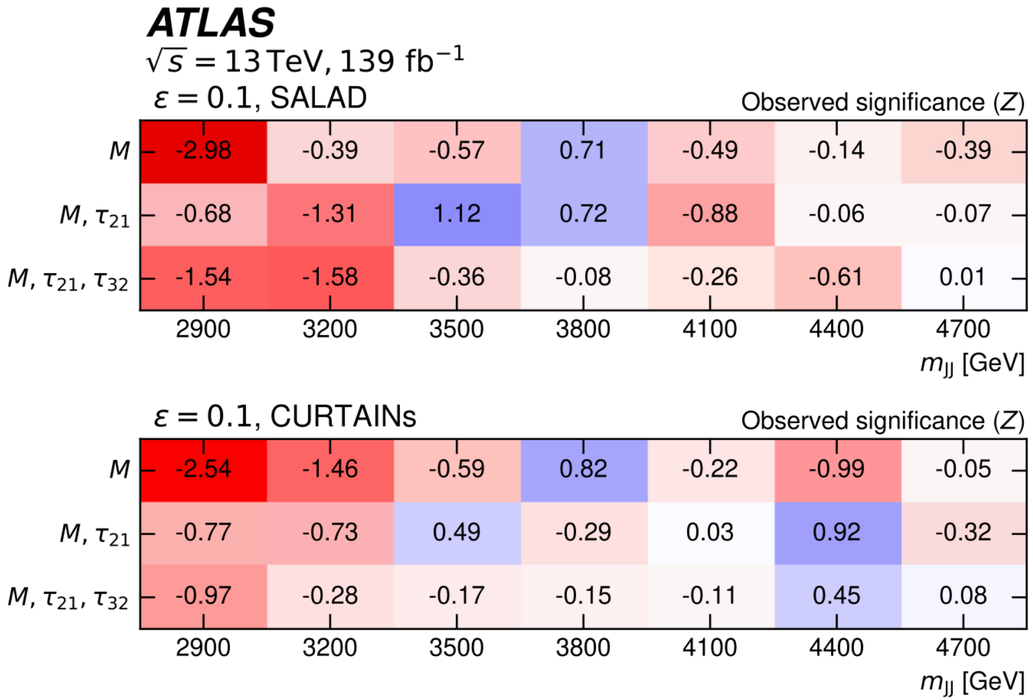
Propaganda
(Biased examples)

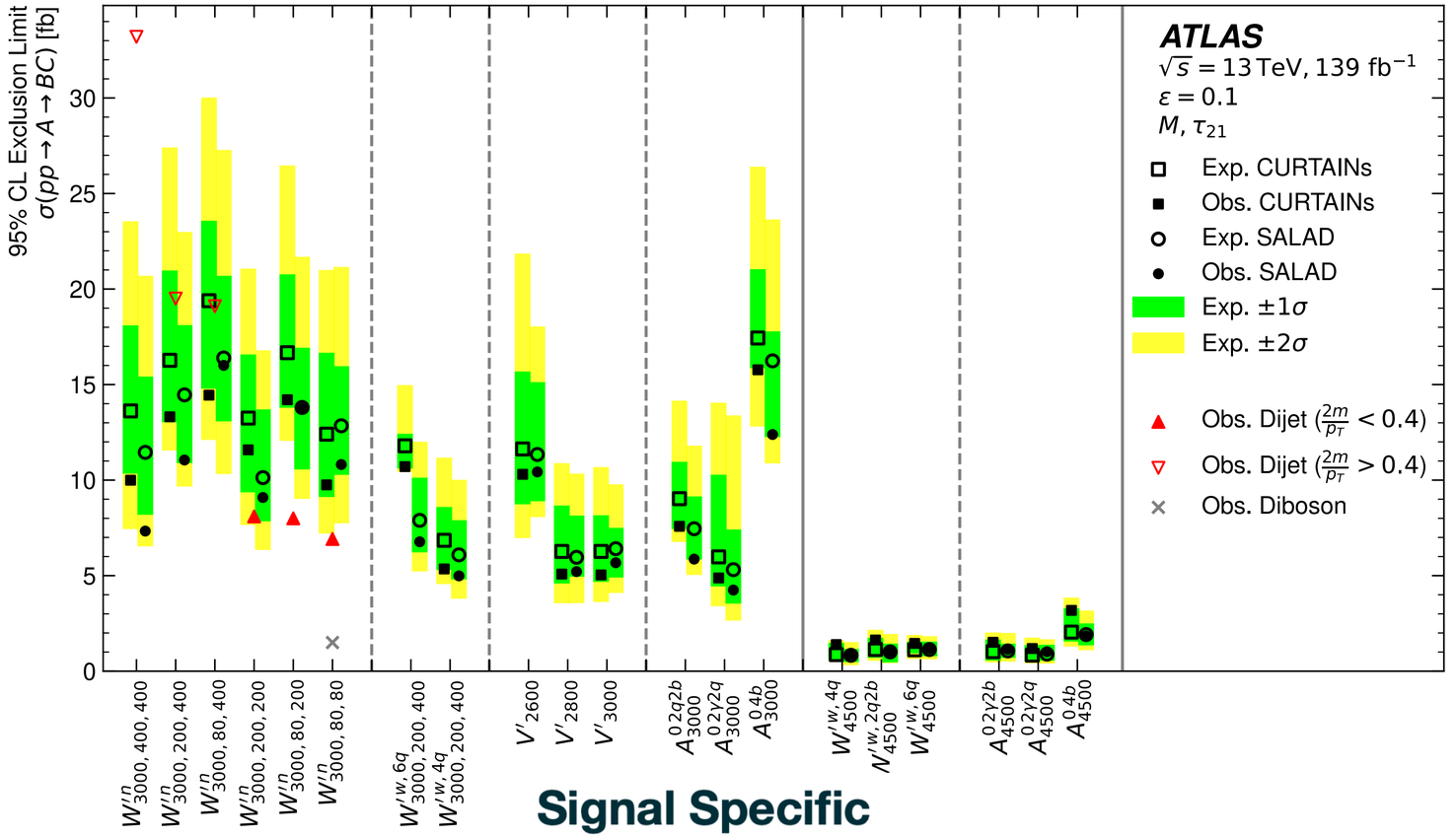
ATLAS Di-jet AD

[Most recent result]

[Previous result]

ATLAS EXPERIMENT





Signal Agnostic

- Observed significance of: 2 methods, 2 efficiencies, 7 m_{JJ} regions, 3 feature sets
- Largest significance is 1.24σ (1.26σ) & local deficit of –2.98σ (–2.54σ) for SALAD (CURTAINs)

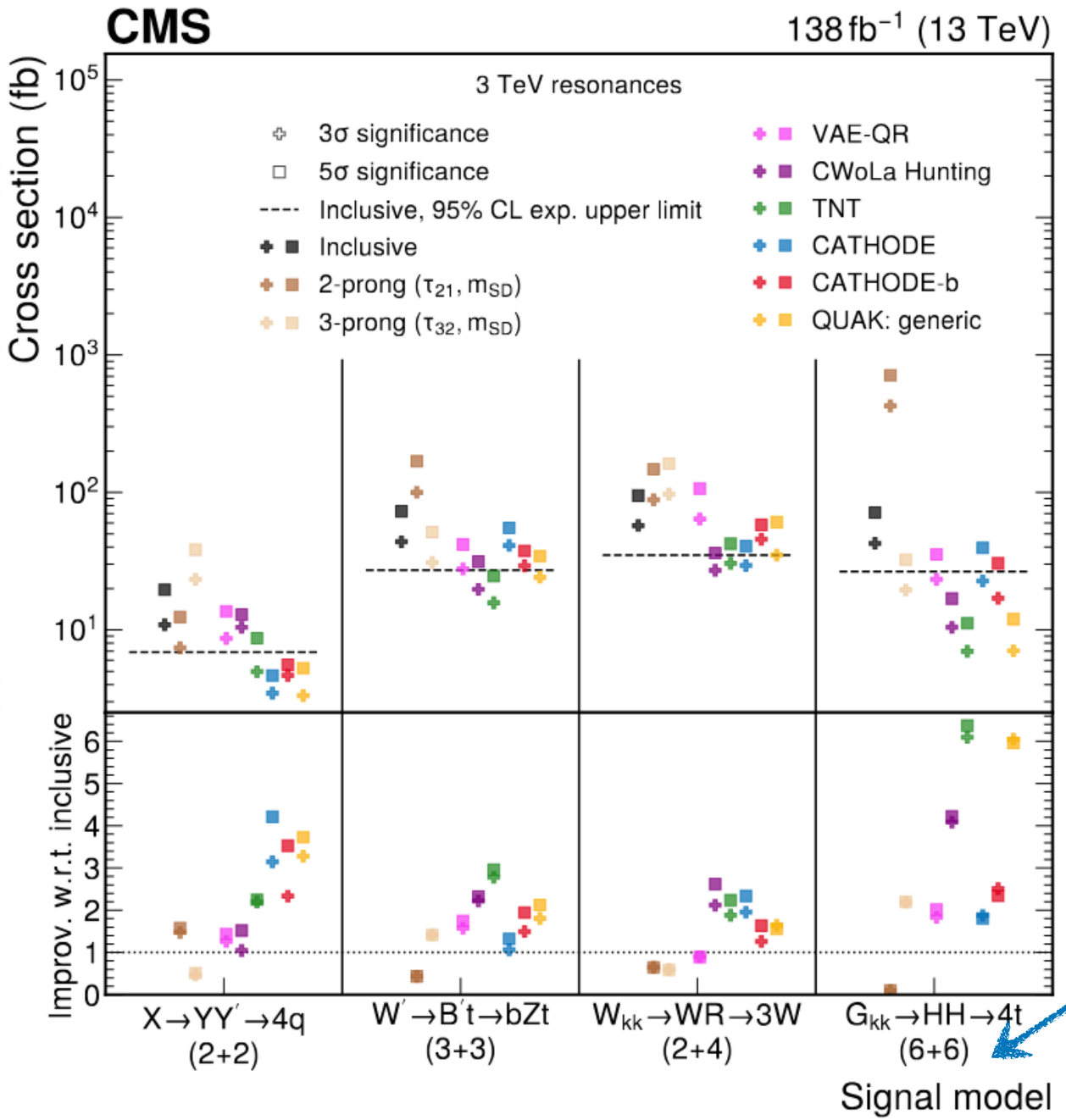
Signal Specific

- Set limits @ 95% CLs to 20 investigated signal models
- Analysis has a broad performance on many models
- Similar performance for SALAD and CURTAINs
- Different feature sets have different sensitivity (more not always better) - scan over feature sets is one of the strengths of this analysis

CMS Di-Jet AD

- CMS employed AD in recent search for dijet resonances
  - Anomaly tag substructure of the jets
- Compared multiple different anomaly methods
  - "What xsec do I need for 3/5σ of signal?"
  - Up to **7x gain** in discovery sensitivity!
- **Lesson : No one universal, 'best' method**

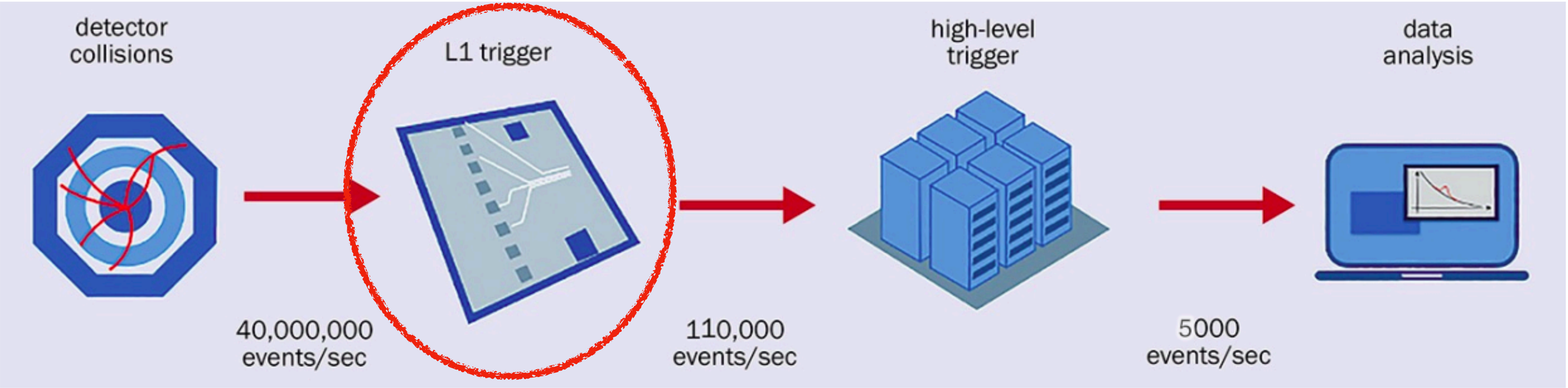
Also interesting: paper on AD for data quality monitoring in CMS



Results

Very diff. signals!

CMS/ATLAS AD Triggers



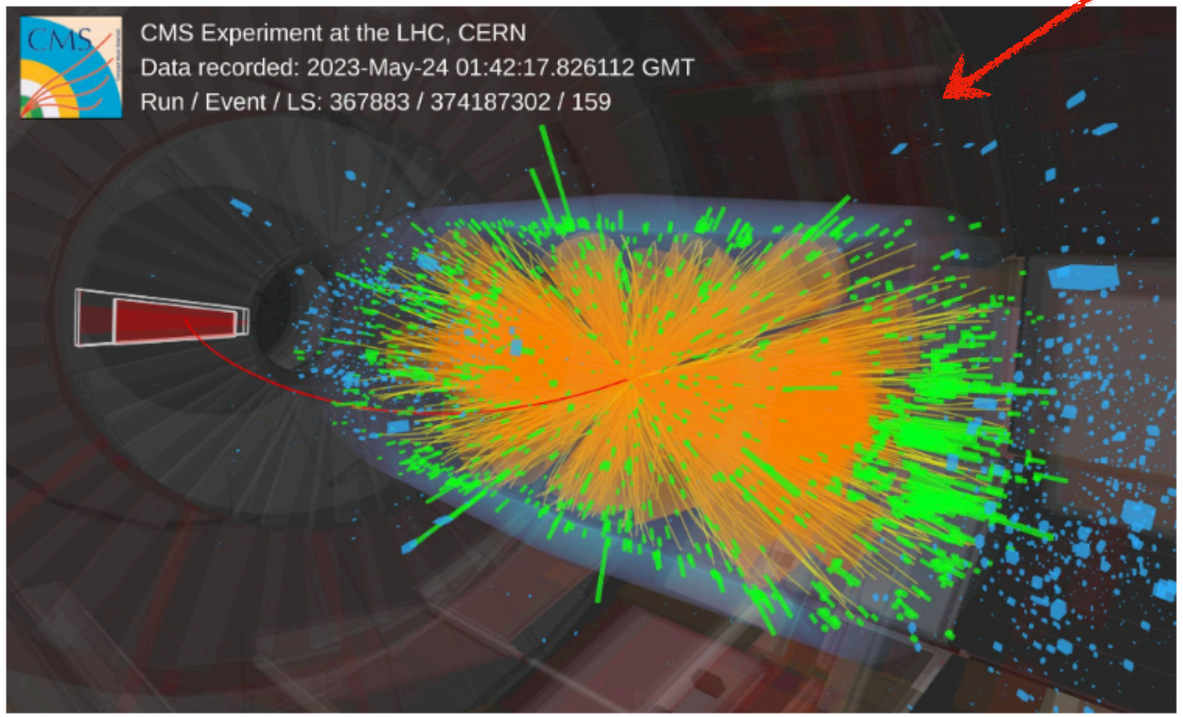
Discard >99% of events at L1 trigger

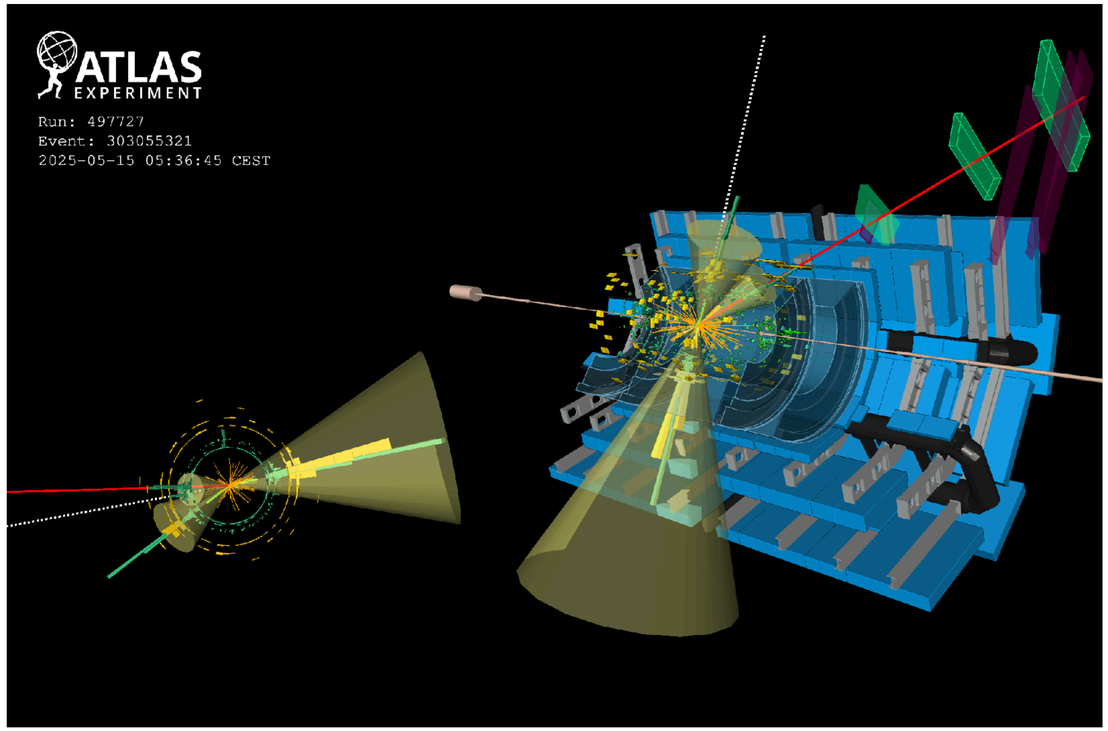
→ could be missing signals!

→ Autoencoder-based anomaly detection in FPGA based triggers!

CMS/ATLAS AD Triggers

👀 First analysis of AD data+dedicated paper on CMS AD triggers coming soon!





AXOL1TL

[AXOL1TL algo]
[AXOL1TL 2024 results]
[AXOL1TL 2025 results]

CICADA

[CICADA algo]
[CICADA 2025 results]

[GELATO algo]
[GELATO 2025 results]

Neutrinos are cool too!

- Talks on AD in neutrino experiments in Neutrino Physics and Machine Learning (NPML 2025) conference
- Paper on Real-time Anomaly Detection for Liquid Argon Time Projection Chambers

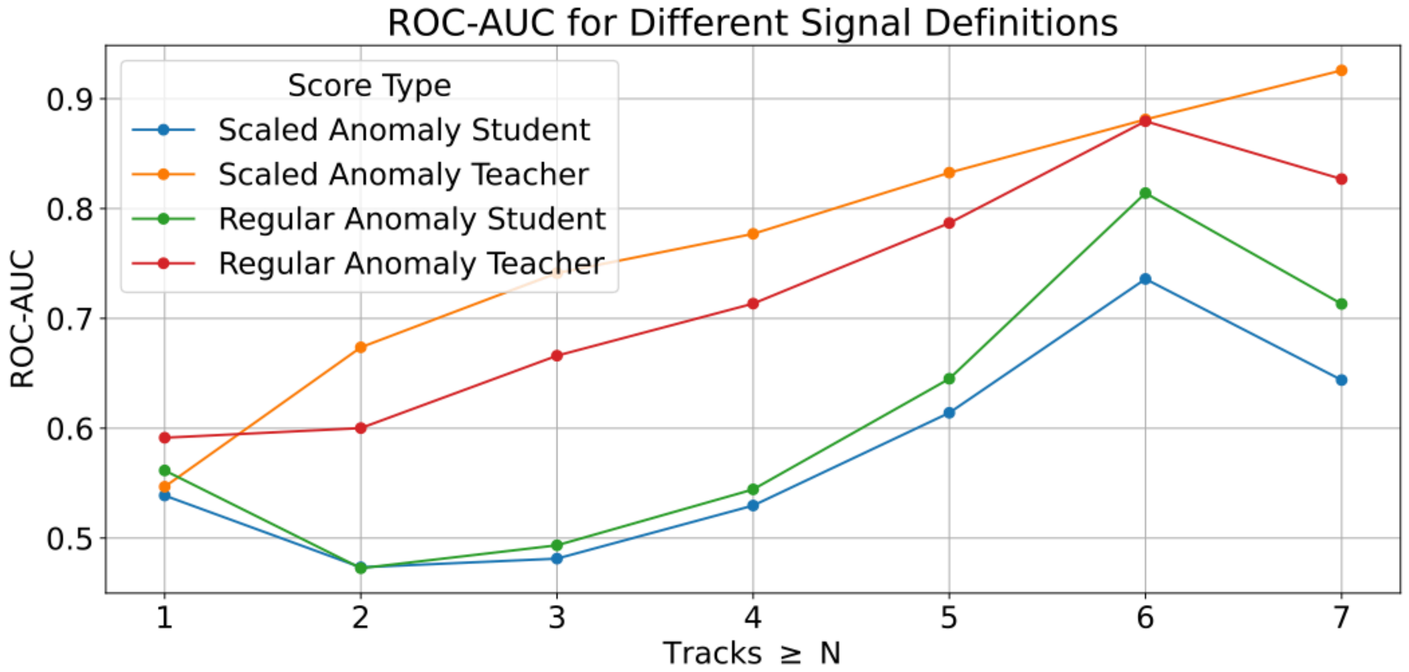
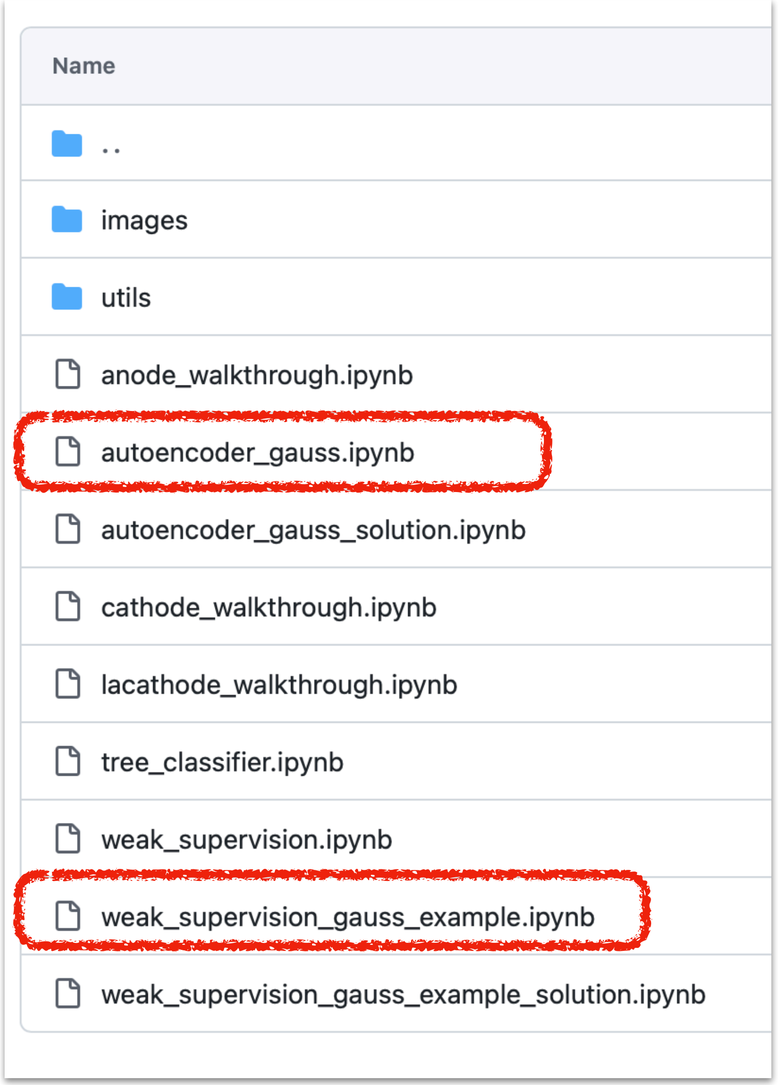


Figure 16: ROC-AUC values for different signal definitions with models trained on 864×64-sized input segments. The signal in each entry was defined as having n or more tracks.

Tutorial Time!

Tutorial

- This is ~~based on~~ the same as Oz Amram's 2025 AD tutorial
  - Credits to Manuel Sommerhalder for building the repo!
- **Set up instructions:**
  - Tutorial day 3 set-up instructions
  - Anomaly detection tutorial instructions
- **'demos' directory includes much more material than we have time for** (Ex. Full CATHODE demos + additional variants)
  - We will focus on Gaussian data for simplicity to illustrate the main ideas
    - Start with **'autoencoder_gauss'**
    - Then do **'weak_supervision_gauss'** if you get to it
    - Explore the other demos if you have time/interest!



Tutorial Setup

	Login Node	Shared GPU Node	Exclusive CPU Node	Exclusive GPU Node	Configurable Job
Perlmutter	start	start	start	start	start
Resources	Use a login node shared with other users, outside the batch queues.	Use a single GPU on a node within a job allocation using defaults.	Use your own node within a job allocation using defaults.		Use multiple compute nodes with specialized settings.
Use Cases	Visualization and analytics that are not memory intensive and can run on just a few cores.	Work that fits on a single GPU, and uses at most a quarter of a GPU node's CPU cores and host memory.	Visualization, analytics, machine learning that is compute or memory intensive but can be done on a single node.		Multi-node analytics jobs, jobs in reservations, custom project charging, and more.

Request a "**Configurable Job**"

Jupyter Hub:
https://jupyter.nersc.gov

Server Options

Account ("_g" suffix will be added as needed):
ntrain6

Constraint:
gpu

QOS:
jupyter_shared

cpus-per-task (GPU node has 128 cpus, CPU node has 256 cpus):
32

gpus-per-task (node has 4 GPUs):
1

nodes (maximum of 4 for jupyter QOS):
1

ntasks-per-node:
1

Reservation:
ml4fp2026_day3

time (time limit in minutes):
360

Start

Tutorial Setup

Update the repo and setup environment:

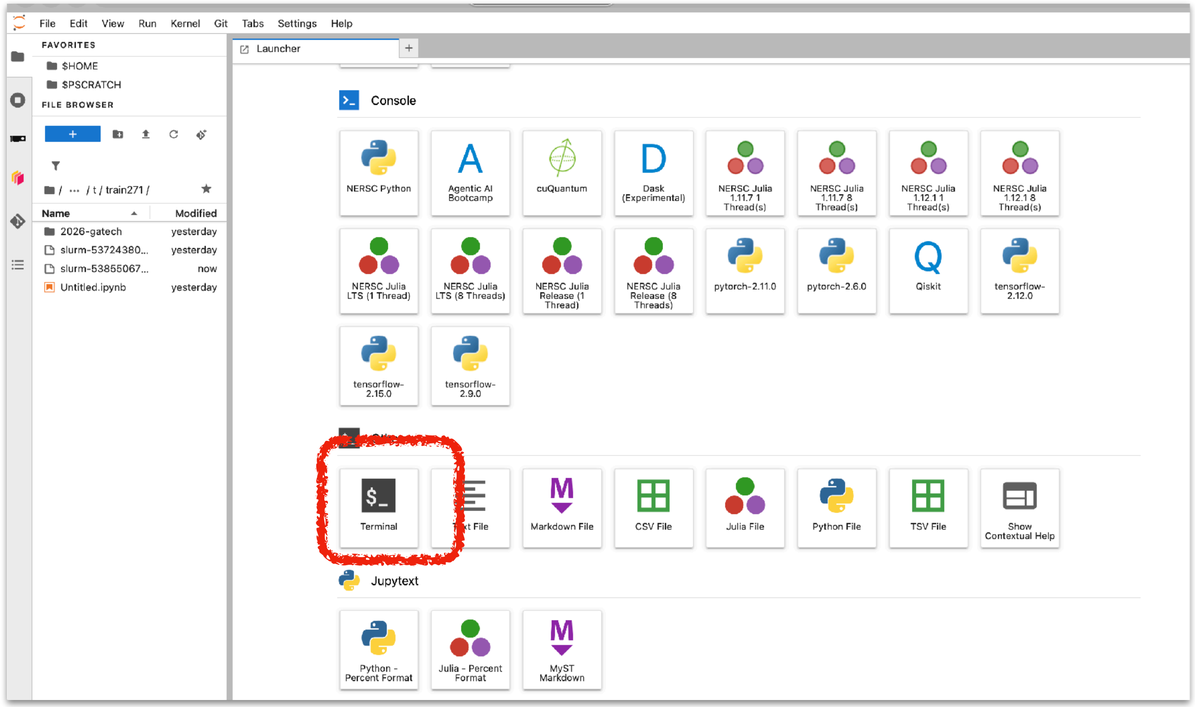
1. Go to the terminal after joining the jupyterhub

2. Go to the repo:
```
cd ~/2026-gatech
```

3. Update the repo:
```
git add -u
git commit -m 'past tutorials'
git pull
```

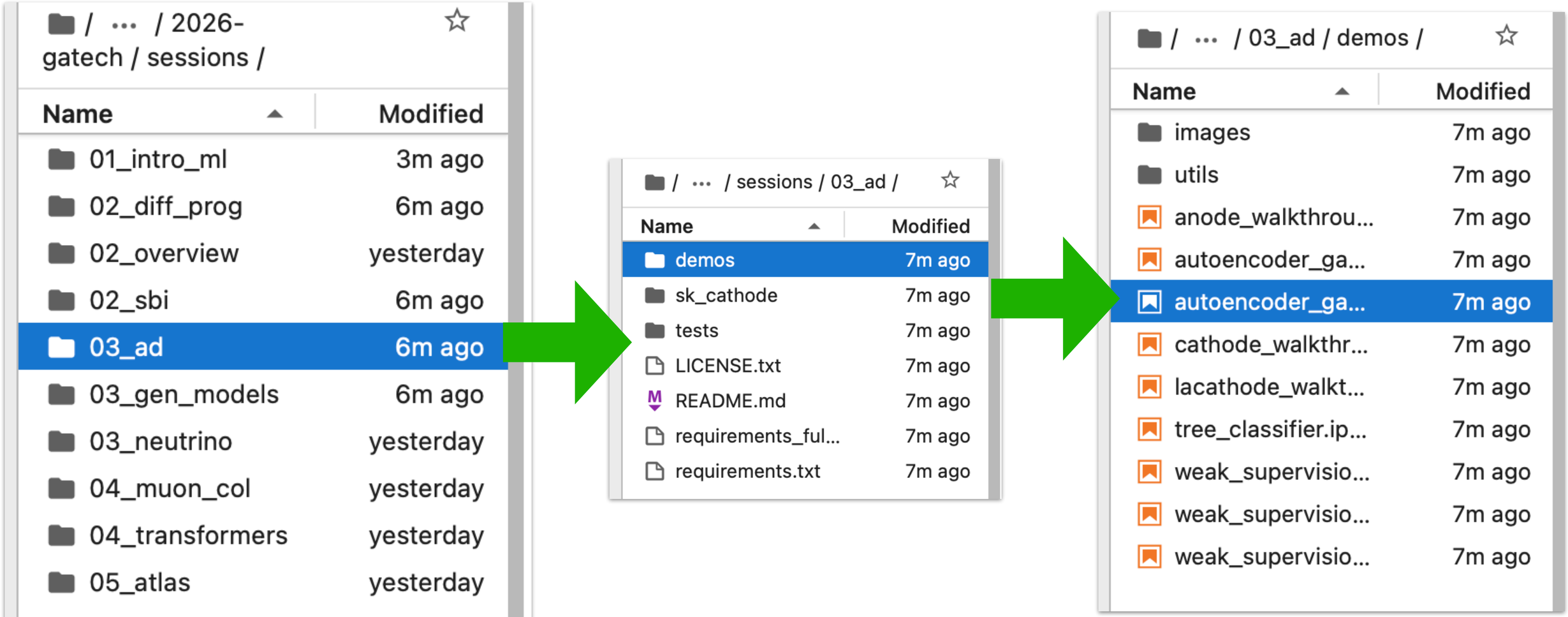
4. Run `cd ~/2026-gatech/sessions/03_ad/`

5. Reload the Jupyter page

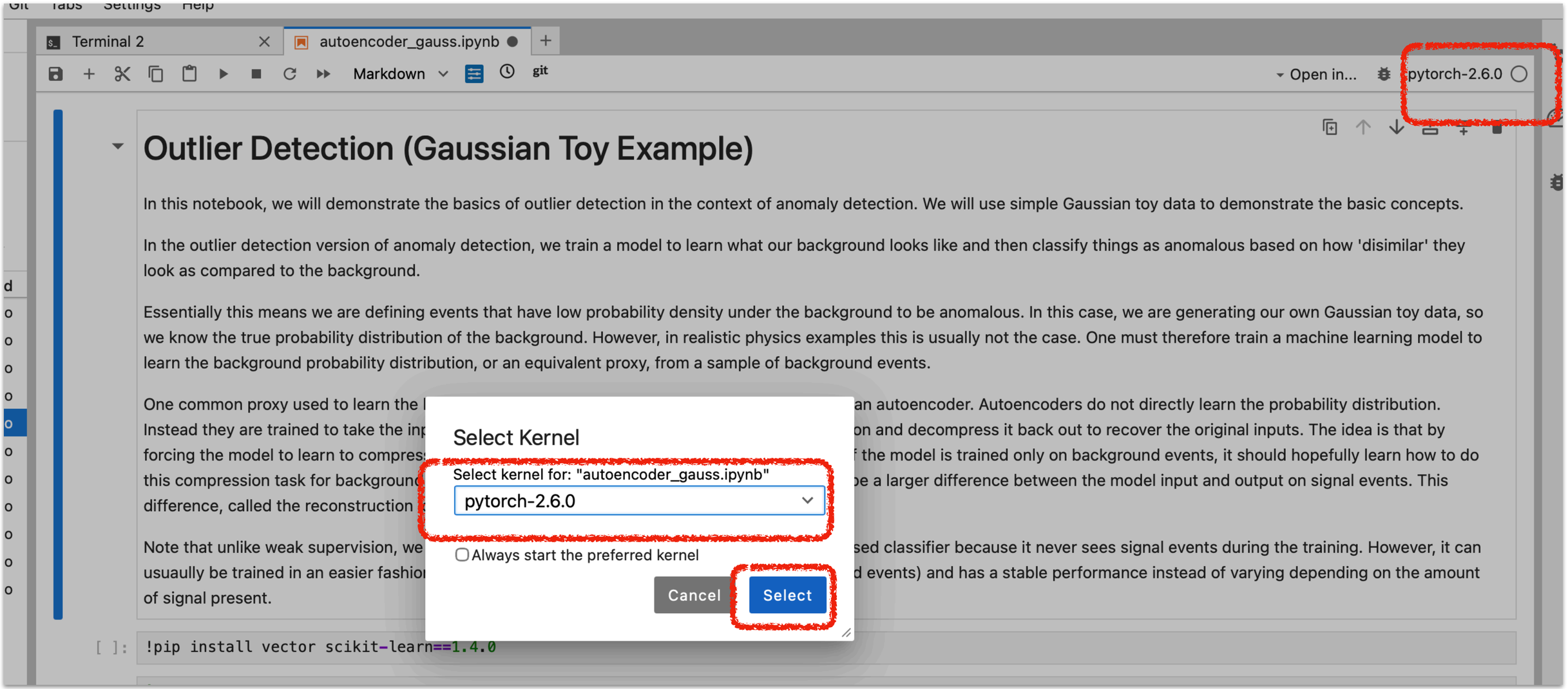


*Some optional ipynbs need additional dependencies- see readme

Tutorial Setup



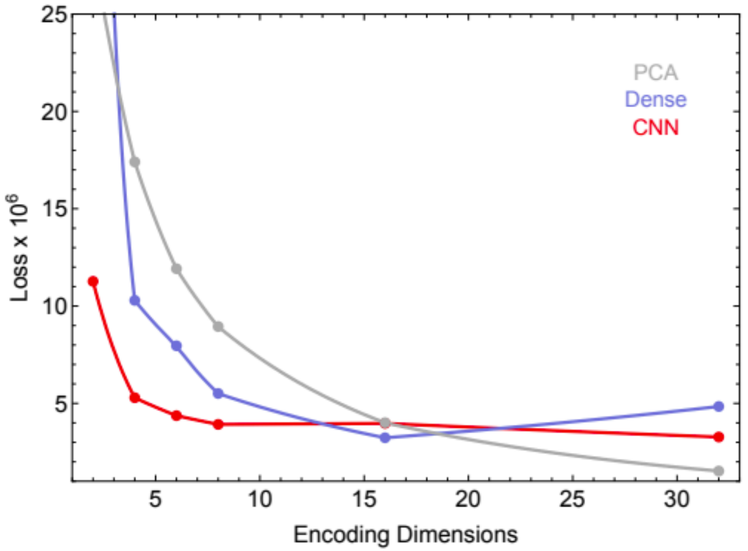
Tutorial Setup



Backup

Autoencoder Practicalities

- Training loss is (typically) MSE between input & output
- Size of compressed (latent) dimension is an important hyperp
  - No exact method to pick it
  - Often look for ‘elbow’ in loss vs. dim distribution
- Can train directly from data!
  - Performance resilient to small amount of signal presence
- Can use variational autoencoder (VAE)
  - Same idea but force latent space to be Gaussian
  - Doesn’t seem to be a huge performance gain



1808.08979
1808.08992

Short Proof

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

Two mixed samples (M_1, M_2) with signal fractions (f_1, f_2)

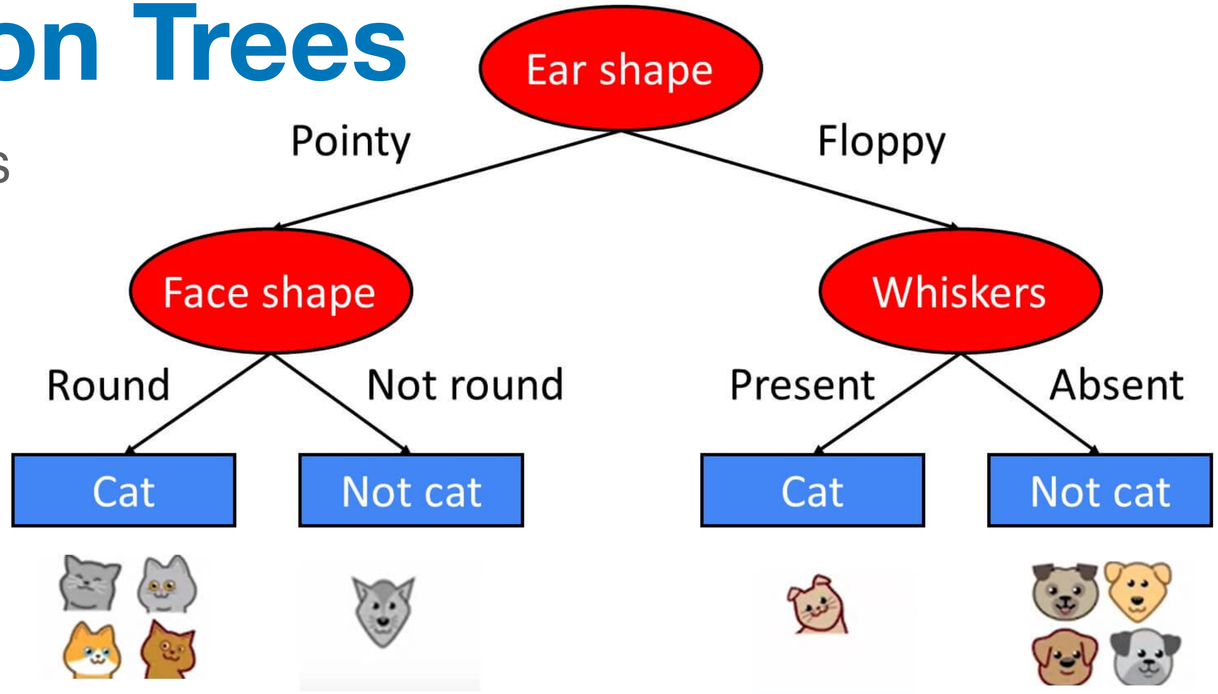
$$L_{M1/M2}(X) = \frac{P_{M1}(X)}{P_{M2}(X)} = \frac{f_1 P_s(X) + (1-f_1)P_b(X)}{f_2 P_s(X) + (1-f_2)P_b(X)} = \frac{f_1 L_{S/B} + (1-f_1)}{f_2 L_{S/B} + (1-f_2)}$$

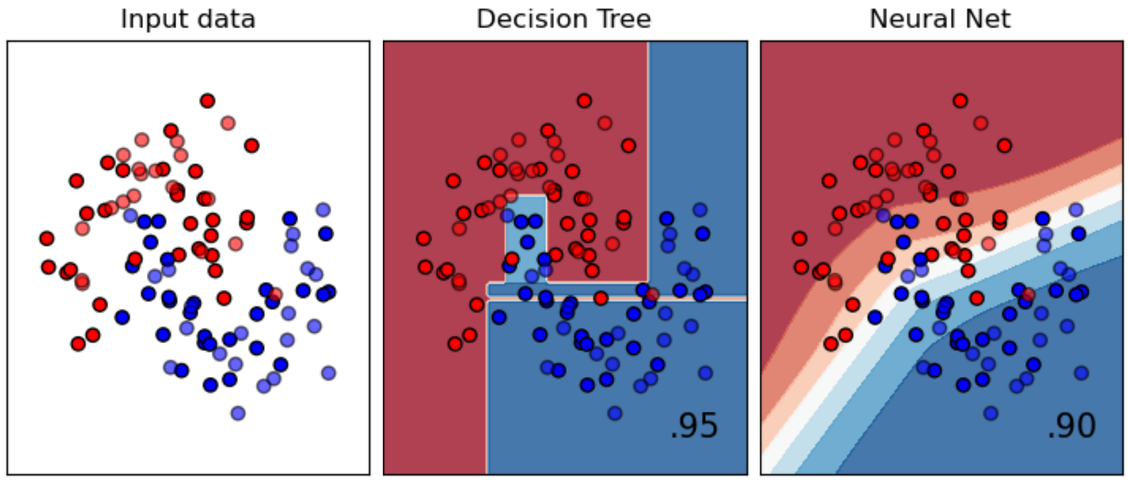
If $f_2 \to 0$ (ie one sample is 'background pure') then simplifies

$$L_{M1/M2}(X) = \frac{f_1 P_s(X) + (1-f_1)P_b(X)}{P_b(X)} = f_1 + (1-f_1)L_{S/B}$$

Sidenote: Boosted Decision Trees

- ~Flowchart of binary cuts on multiple variables
- Boosting: combine many trees:
  - When a tree misclassifies events, the next tree is trained with those misclassified events given higher weight
  - The final score is a weighted vote across all trees.
- Interpretable, easy to train
- Excellent results for appropriate problems!





→Pick the model that suits your task

BDT tutorial scikit-learn.org

Statistical interpretation is difficult

- To set a limit you need to know your signal efficiency
  - Your signal efficiency depends on how well the classifier learned the signal
  - How well the classifier learned depends on how much signal is present
    - But how much signal is present is what you're trying to measure!