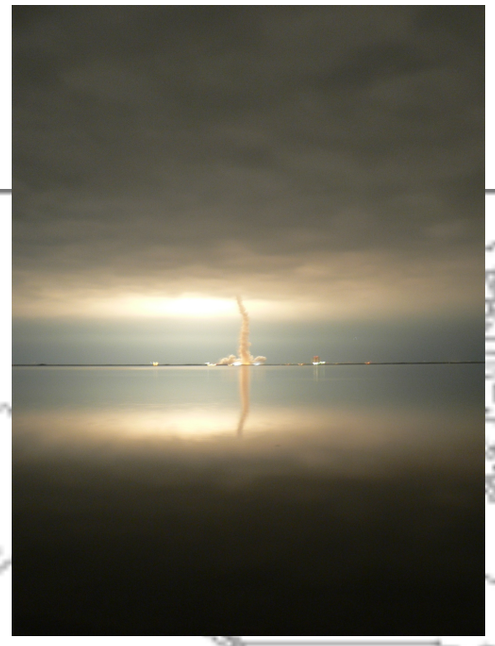
NitheCS MiniSchool: Introduction to Big Data and Machine Learning for Particle Physics

Deepak Kar

School of Physics, University of Witwatersrand and
ATLAS Collaboration, CERN, Geneva, Switzerland

August, 2021

Who am I?



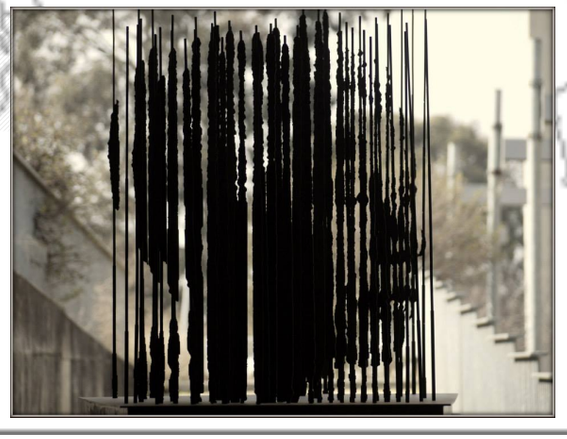
Gainesville, FL, USA
2003-2008

Glasgow, UK
2012-2014

Dresden, Germany
2009-2011

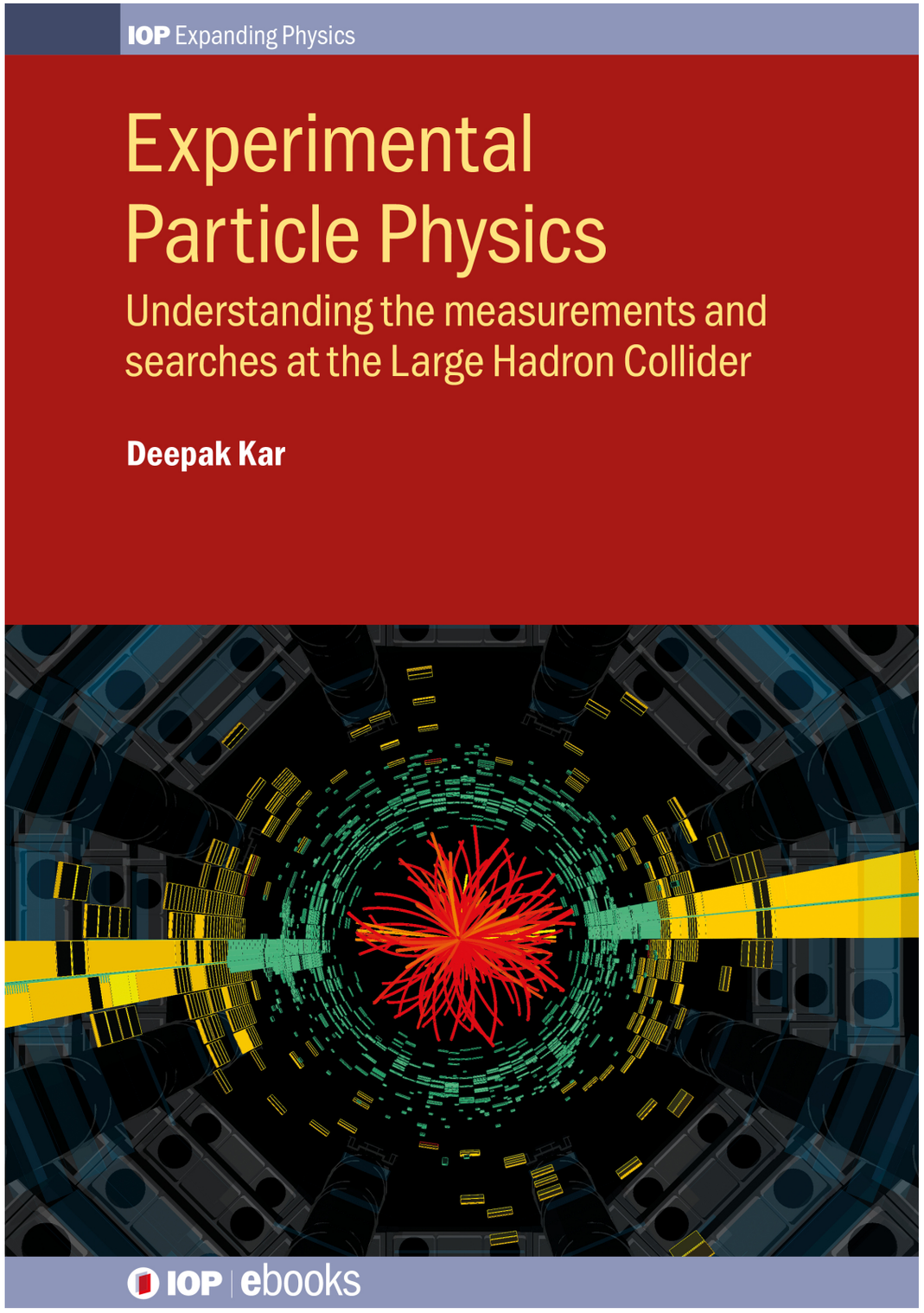
Calcutta, India till *2003*

Geneva, Switzerland
2011-2012



Johannesburg, SA
2015 -

Resource



Link

Prologue

- I am a practising physicist, so I apply machine learning problems to gain insights into the data from Large Hadron Collider at CERN.

- Most (all?) of you are (hopefully) enthusiastic about the techniques we use, but agonistic about the physics part.

- So we will meet somewhere in the middle…

Open questions

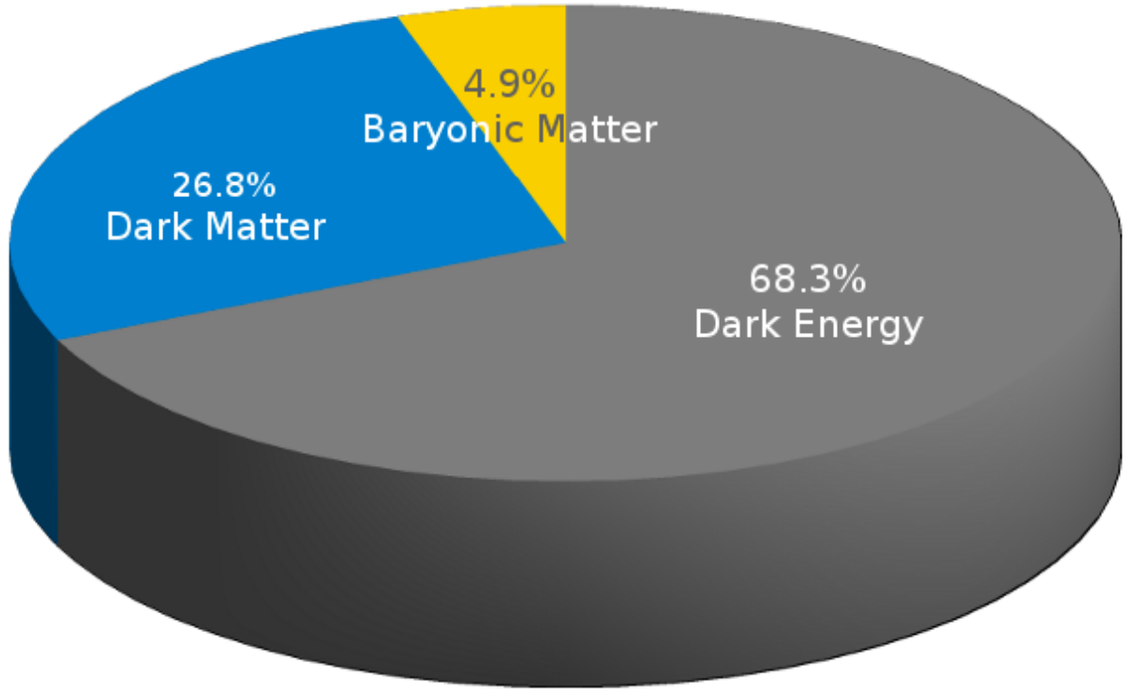
Mass values

- Why particles of these particular mass values?
- Why are there three generations of fermions, with a pattern in masses?
- Higgs mass fine tuning, hierarchy problem?
- Why are neutrinos so light?

Dark matter and dark energy

- Origin of dark matter and dark energy?

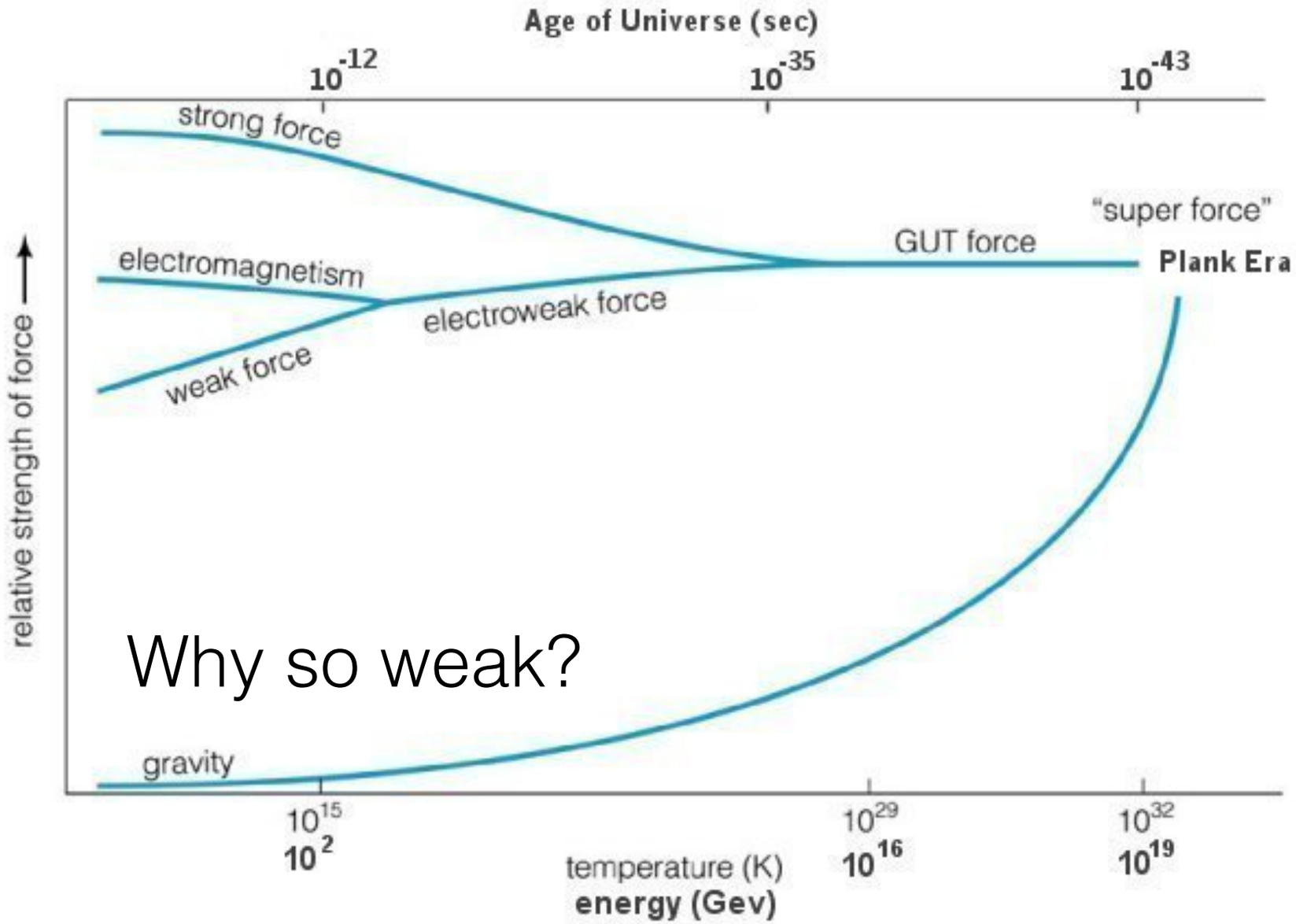
- How to find them?

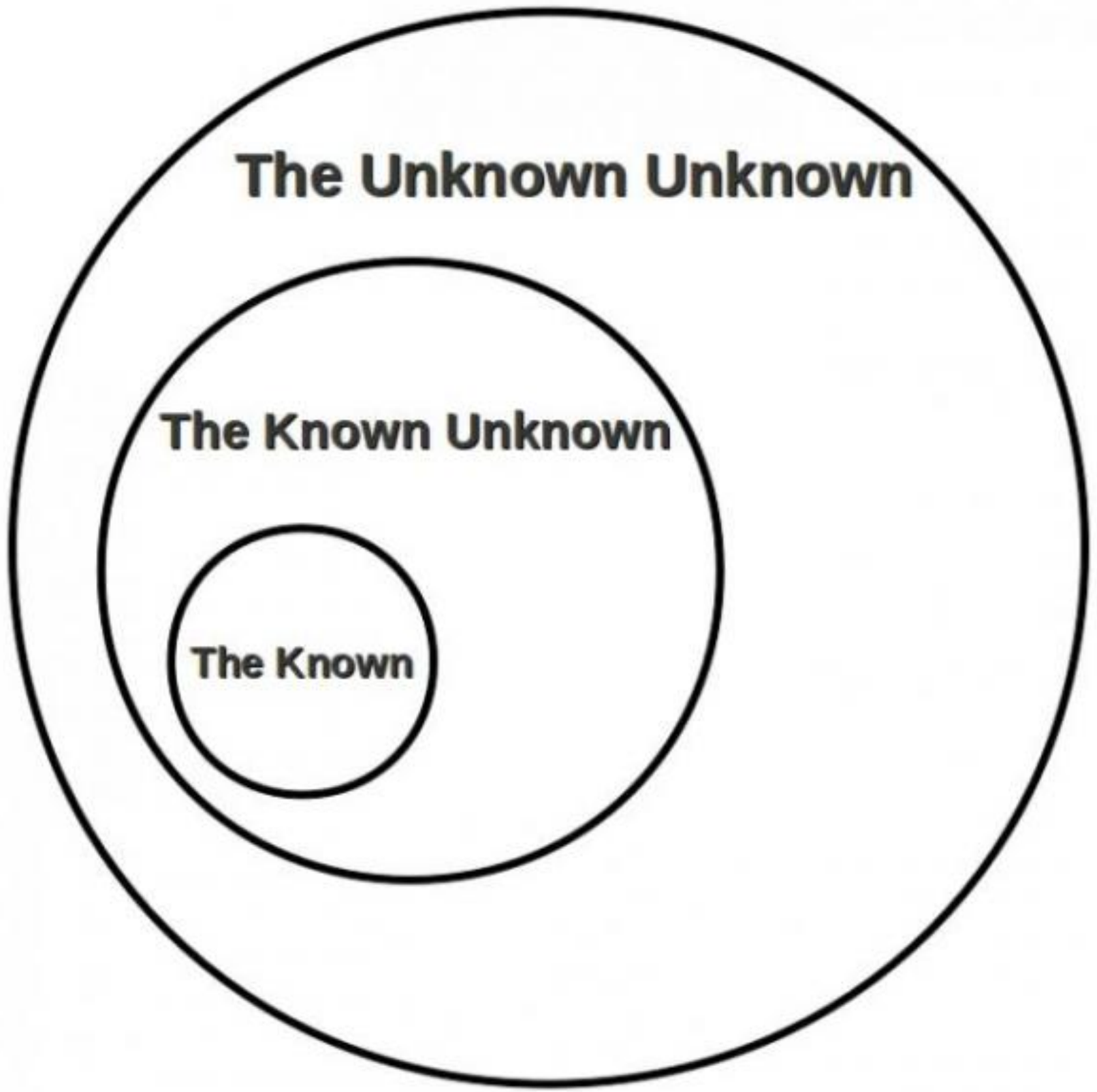


CP violation

- Matter-antimatter asymmetry?
- Violation of Baryon number conservation
- Not enough CP violating processes in SM

Unification of forces?

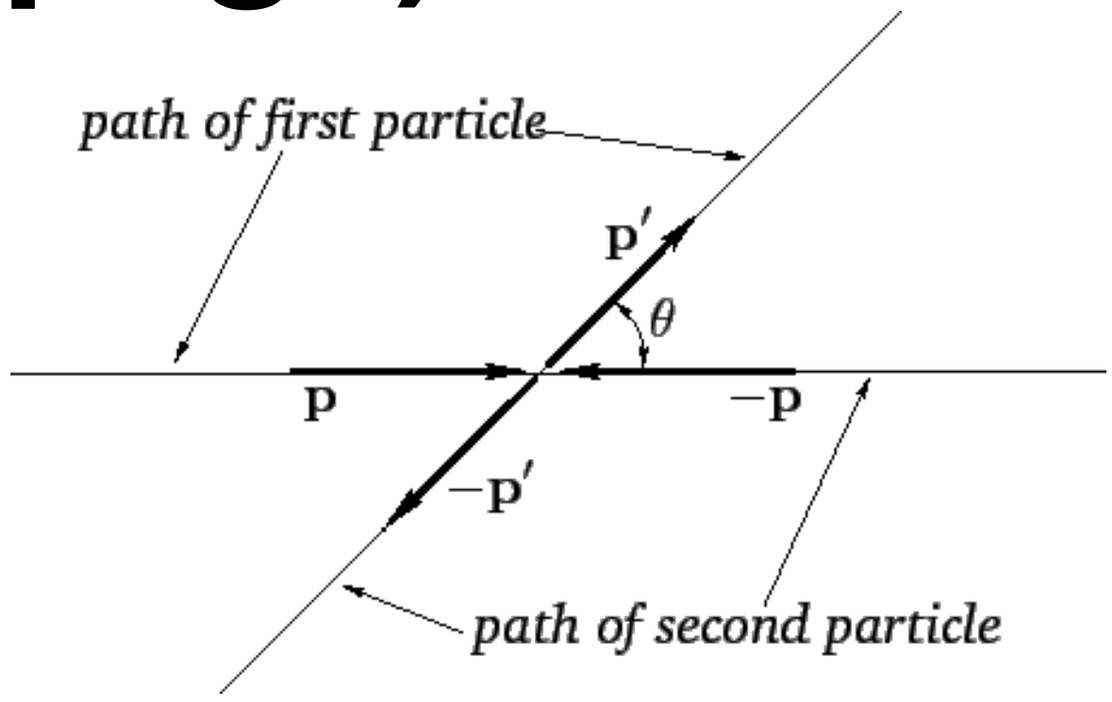



The Unknown Unknown
The Known Unknown
The Known

Relativistic kinematics (in one page)

Frames:

1. Center-of-mass frame
2. Detector frame



We use four momentum: (E, p_x, p_y, p_z)

$$(P_0, \vec{P}) = (E, \vec{p}) = m(dx_0/d\tau, dx_1/d\tau, dx_2/d\tau, dx_3/d\tau)$$

Rest energy — Three momentum

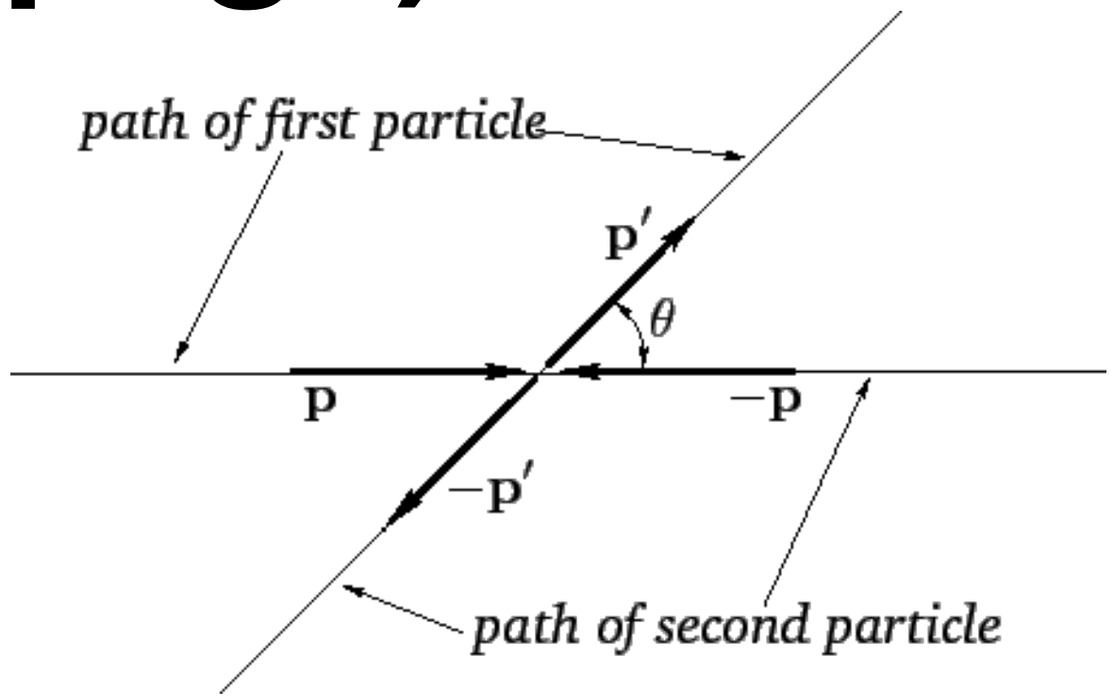
Scalar product with itself is invariant!

Invariant mass: $P \cdot P = E^2 - p^2$

Relativistic kinematics (in one page)

Frames:

1. Center-of-mass frame
2. Detector frame



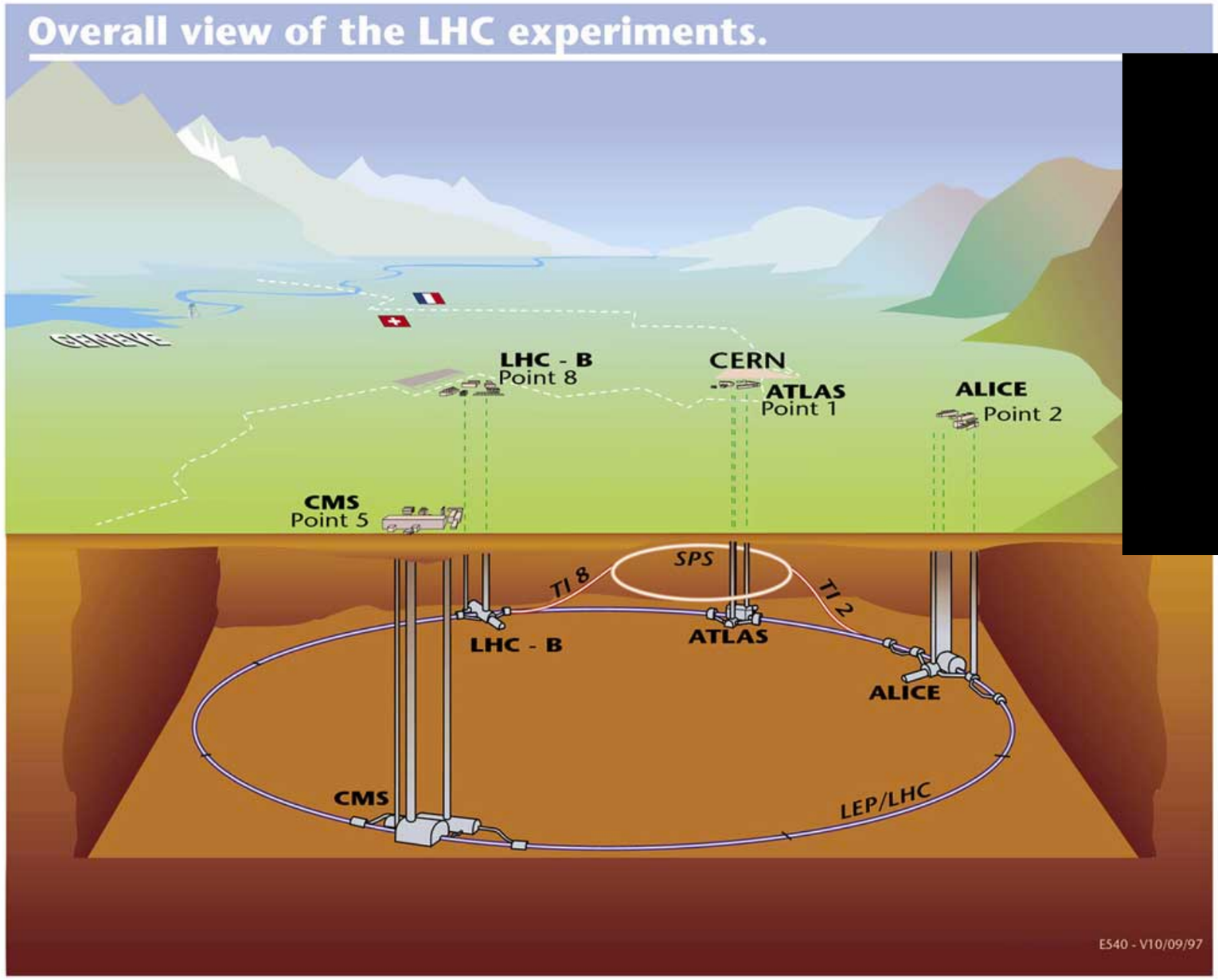
We use four momentum: (E, p_x, p_y, p_z)

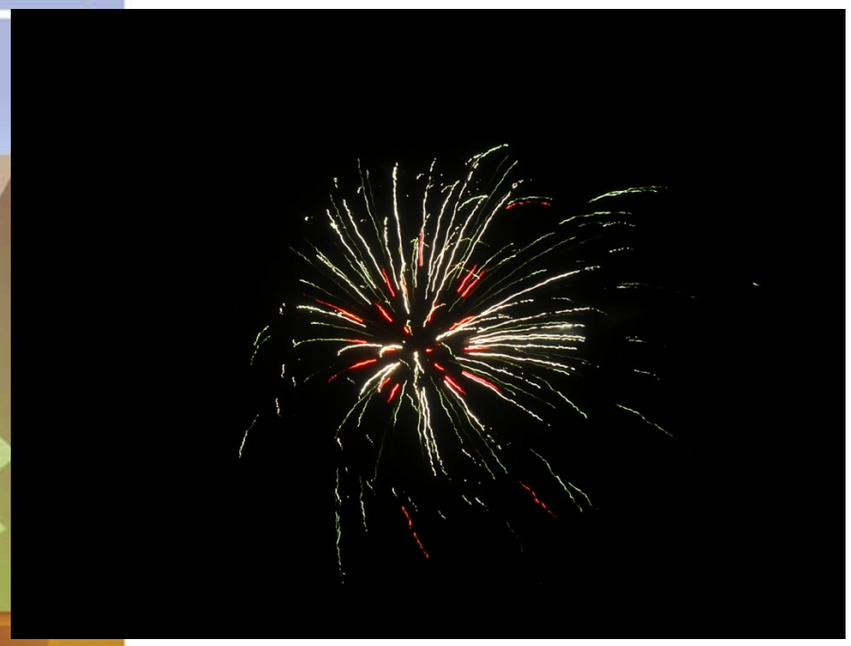
$$(P_0, \vec{P}) = (E, \vec{p}) = m(dx_0/d\tau, dx_1/d\tau, dx_2/d\tau, dx_3/d\tau)$$

Rest energy Three momentum

For a particle decaying to multiple particles, then the outgoing daughter energies and momenta can be added to find the invariant mass. As this is the same in all frames, including the centre-of-mass frame, then the particle can be identified in any frame from its decay product energies and momenta.

LHC and what we do there?

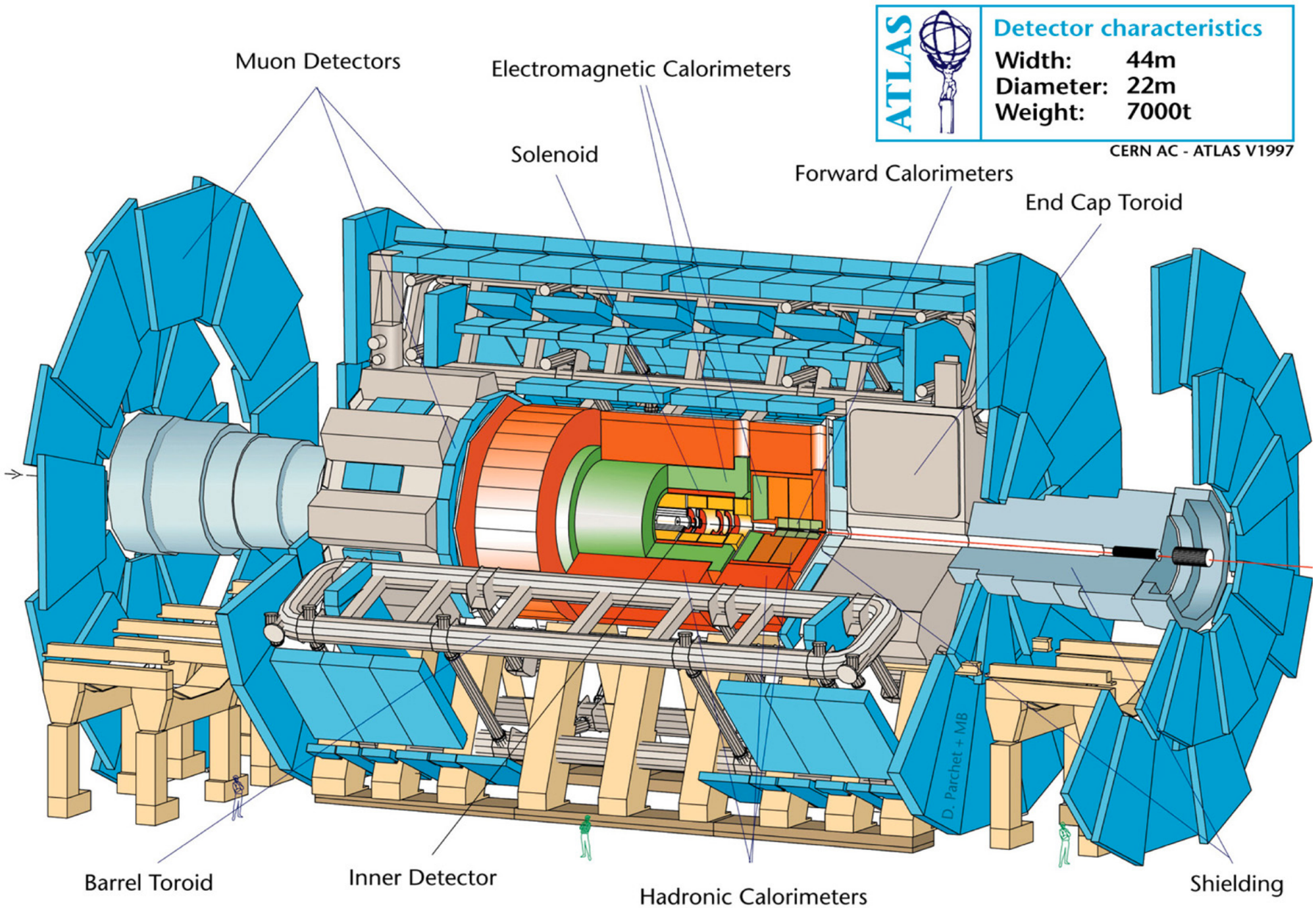




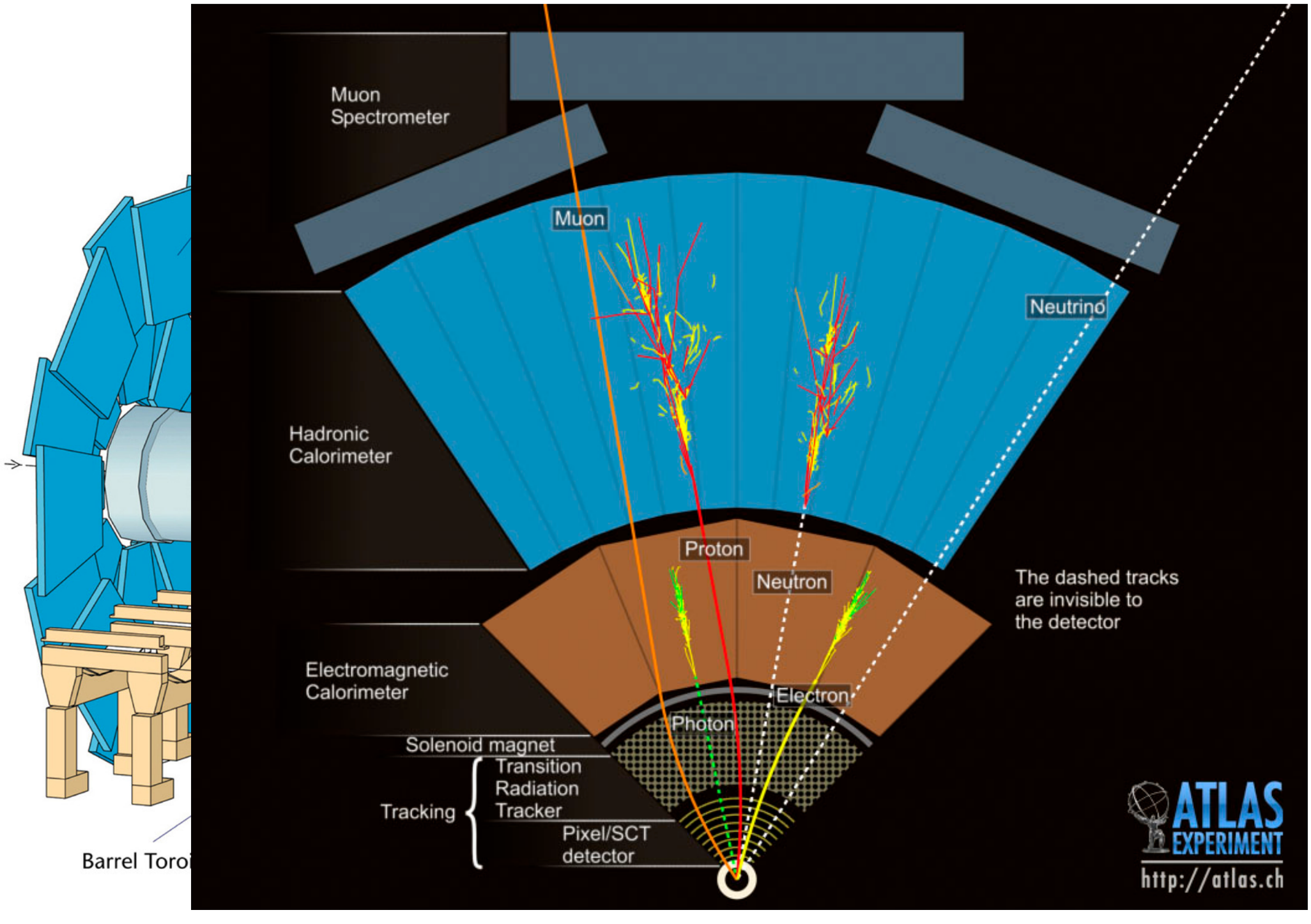
Proton-proton collisions!

40 million of them in one second!

Particle Detection



Particle Detection

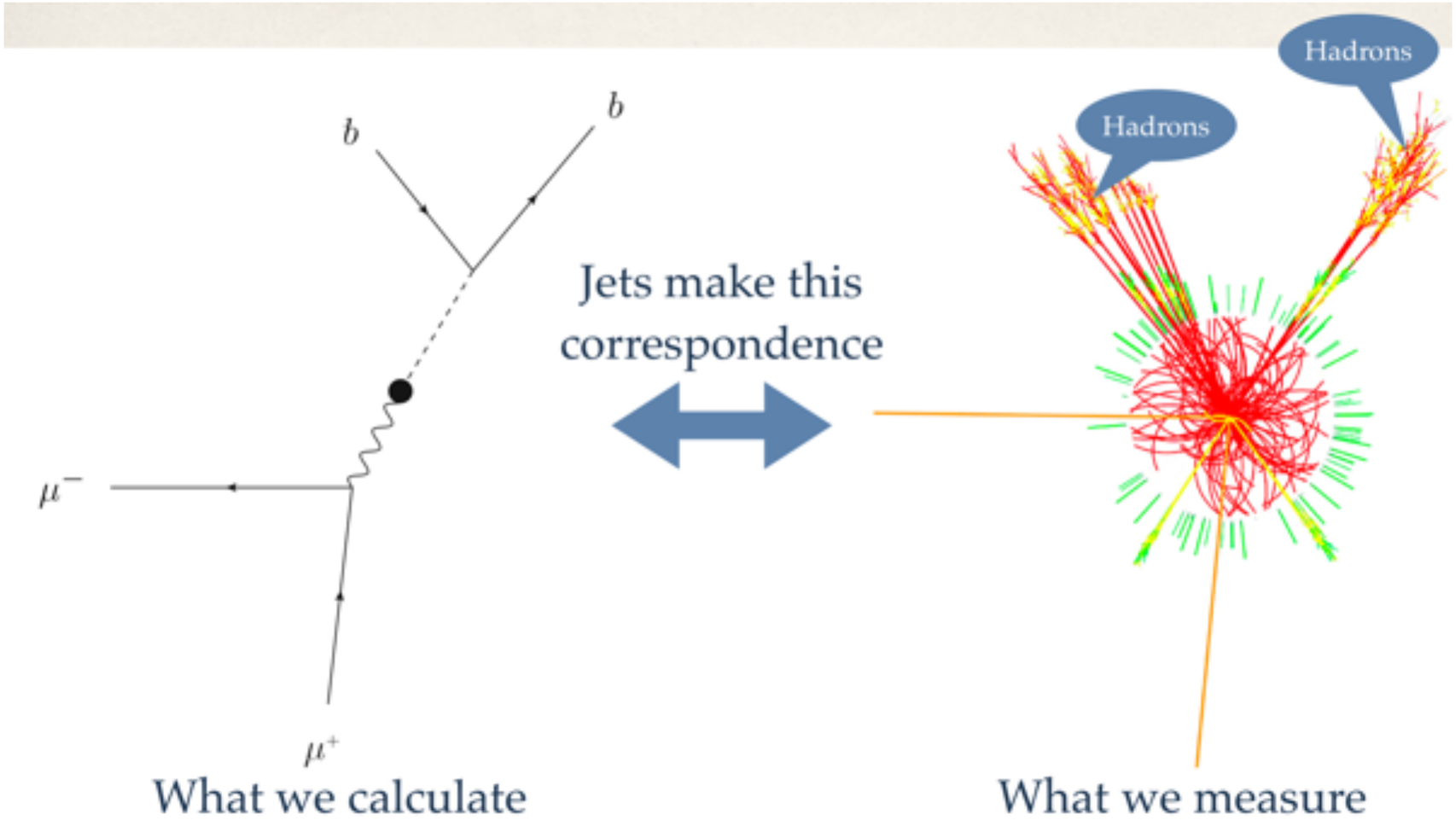


- Detectors measure particle trajectories, energy deposits, charge etc in form of electrical signals.

- From that, we need to “reconstruct” elementary particles like electrons, muons, photons, pions etc.

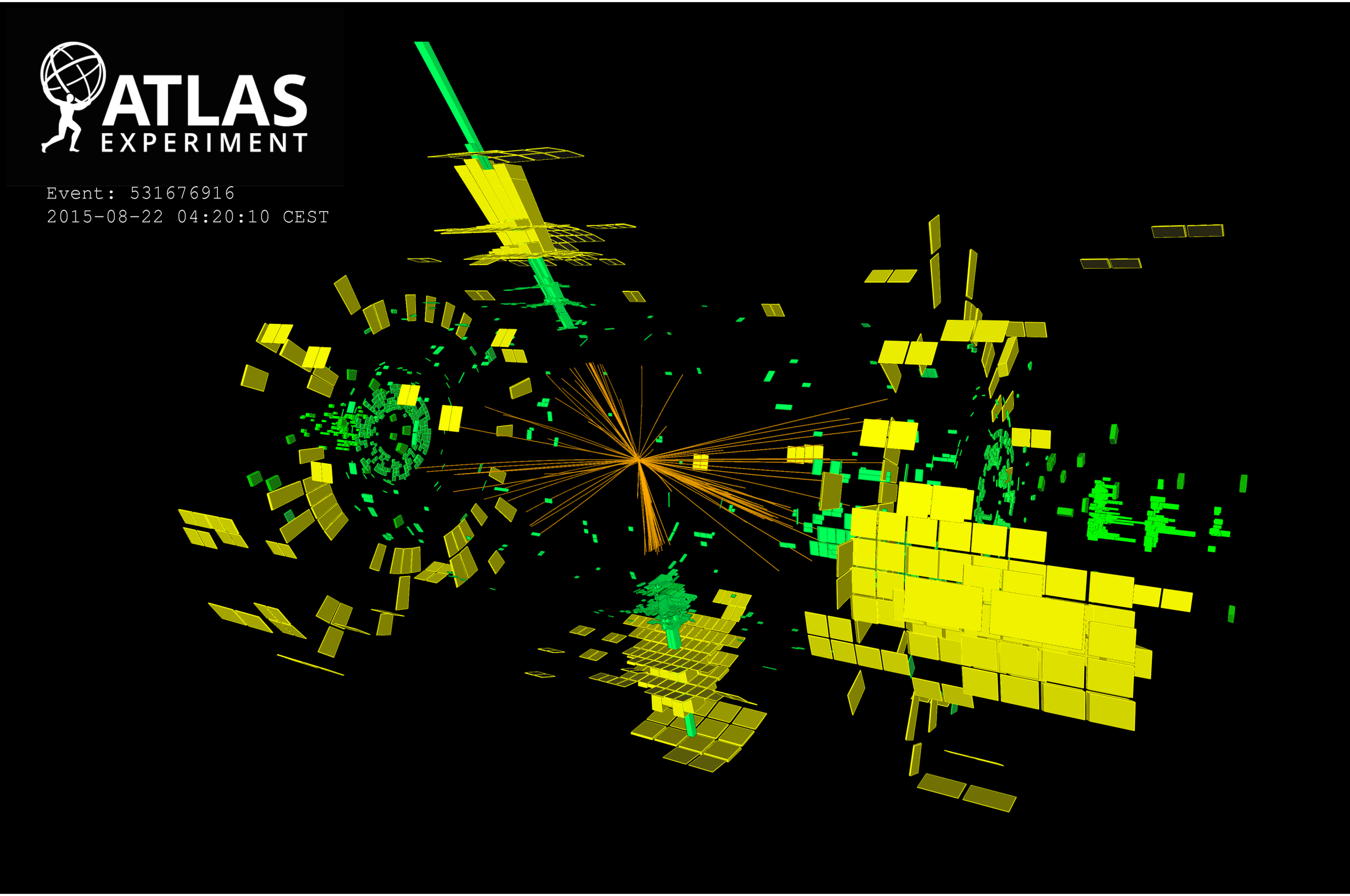
- Also trigger to identify the interesting events to store, hundreds of events/sec from ~100 millions. Filter events to reduce data rates to manageable level. But nature is kind..

Jets



Jets are defined by how it is formed (algorithm), and the (cone) size/radius.

Jets


ATLAS
EXPERIMENT
Event: 531676916
2015-08-22 04:20:10 CEST

What you see is not what you get!

- We dont get what is coming out of the collisions.
- Finite lifetime of particles, decays before reaching the detector.
- Detectors have finite resolution, less than perfect response and efficiency.

Process		Cross-section (in pb)	
Total Inelastic		8×10^{10}	
Process	Cross-section (in pb)	Process	Cross-section (in pb)
W^+, W^-	$1.2 \times 10^4, 8.8 \times 10^3$	Wt	94
Z	2×10^3	H	43
$t\bar{t}$	820	WZ	40
Single $t, \bar{t}$	136, 80	ZZ	15
W^+W^-	142	HH	0.03

- Cross-section: how often a particular process occurs.

- Measured as an effective area the target particle presents to projectile particles.

Barn = 10^{-28} m^2

Unit		Prefix	
Barn (b)		1	
Unit	Prefix	Unit	Prefix
Mili (mb)	10^{-3}	Pico (pb)	10^{-12}
Micro (μb)	10^{-6}	Femto (fb)	10^{-15}
Nano (nb)	10^{-9}	Atto (ab)	10^{-18}

Some numbers

- Design frequency of LHC: 40 MHz = 25 ns bunch spacing.

- 1 bunch contains ~1.15 x 10^{11} protons, 2808 bunches per beam.

- 1 inverse fb = ~ 10^{12} pp collisions at LHC

- ATLAS records ~300 Hz, i.e 300 events per second.

- Size of event ~1.5 MB, so 1 fb^{-1} means ~500 TB of data.

Coordinates

- Only transverse plane (x-y) is relevant
- Coordinates encoding both energy and position information
- Lorentz boost invariant!

Pseudo-rapidity

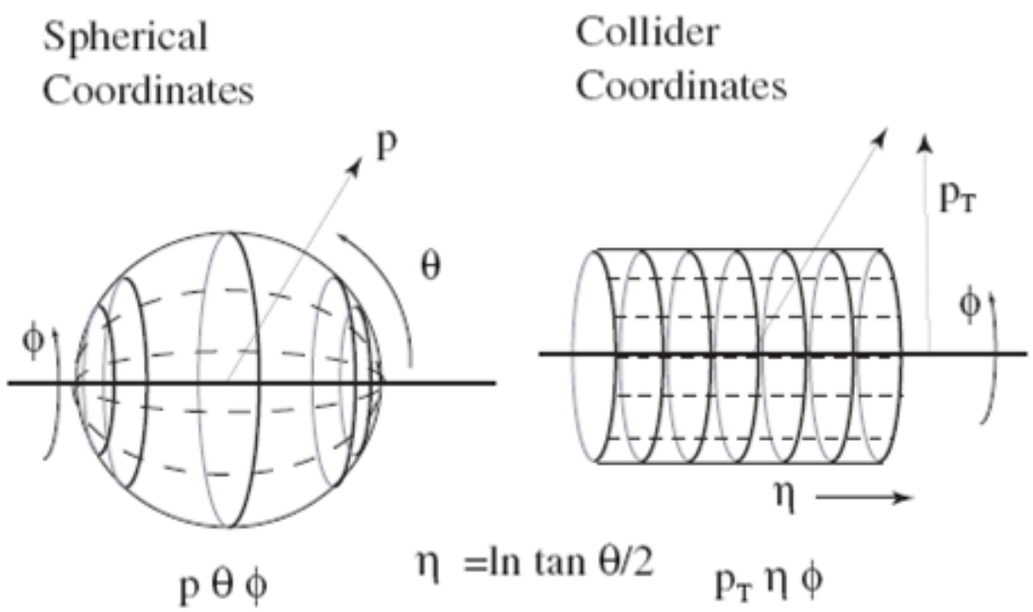
Rapidity: $y = \frac{1}{2}\ln(\frac{E+p_z}{E-p_z})$

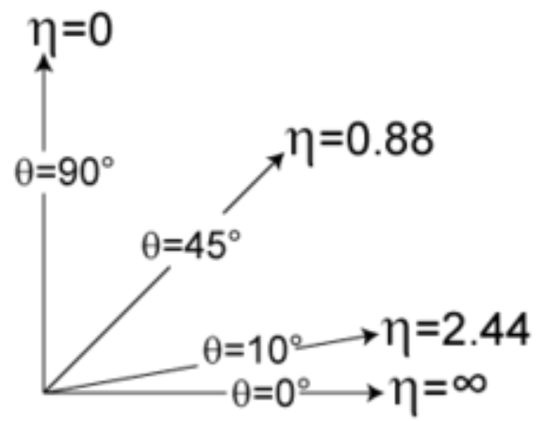
Cant measure p_z!

Difference of rapidity is invariant under Lorentz boost (along z-axis).
But hard to measure, since only transverse momentum is measured.

Pseudo-rapidity: $\eta = -\ln\tan\theta/2$

Identical for massless particles





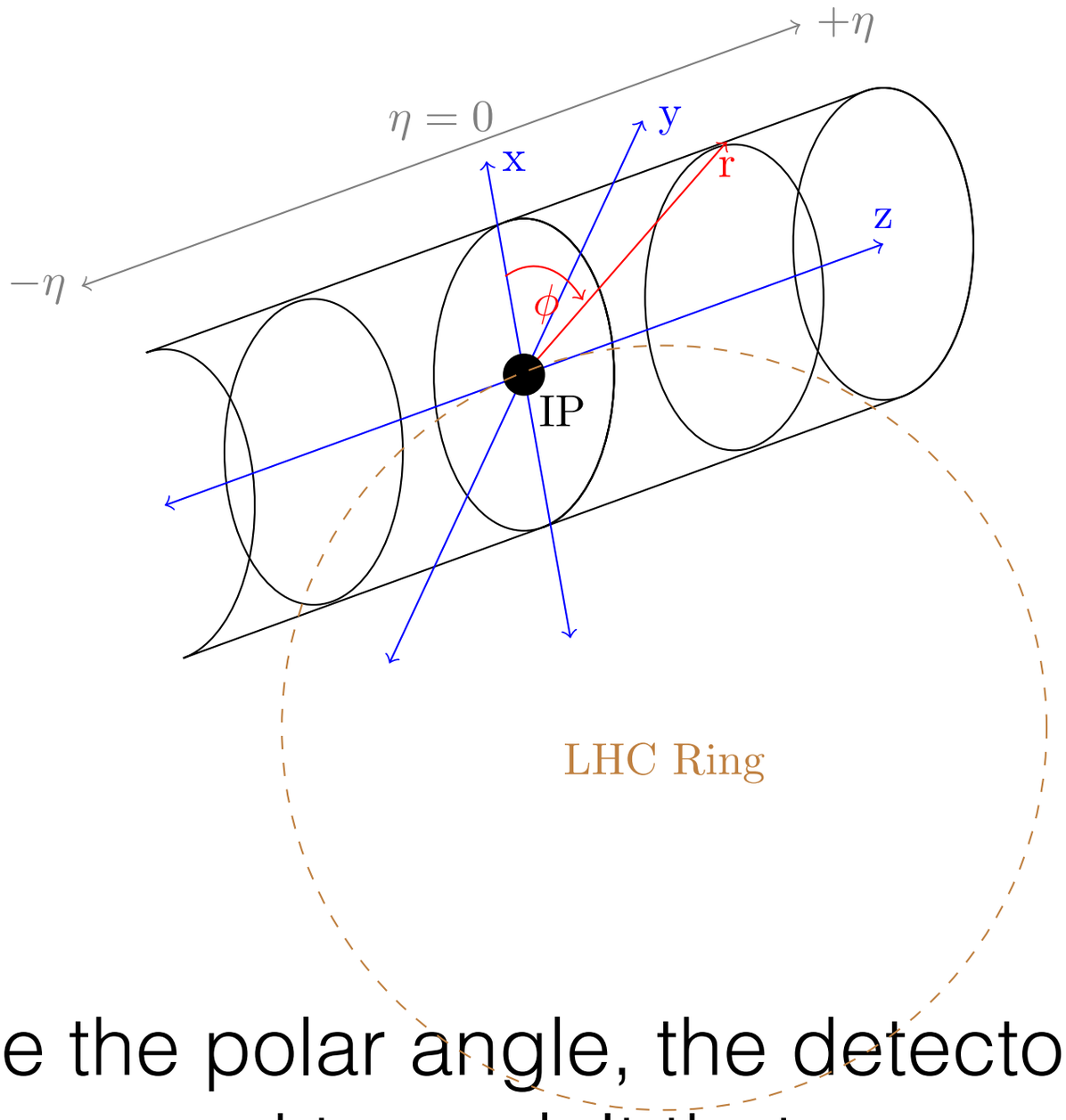
Position!

Distance:

Transverse Momentum: $p_{\mathrm{T}} = \sqrt{p_x^2 + p_y^2}$

$\Delta R = \sqrt{\Delta\eta^2 + \Delta\phi^2}$

Coordinates



We dont use the polar angle, the detector is cylindrical, so need to exploit that symmetry.

One last bit…

- When we say data, we mean data collected from the machine.

- However, you cant judge if there is a difference from expectation (i.e new physics!) unless you know what the expectation is…

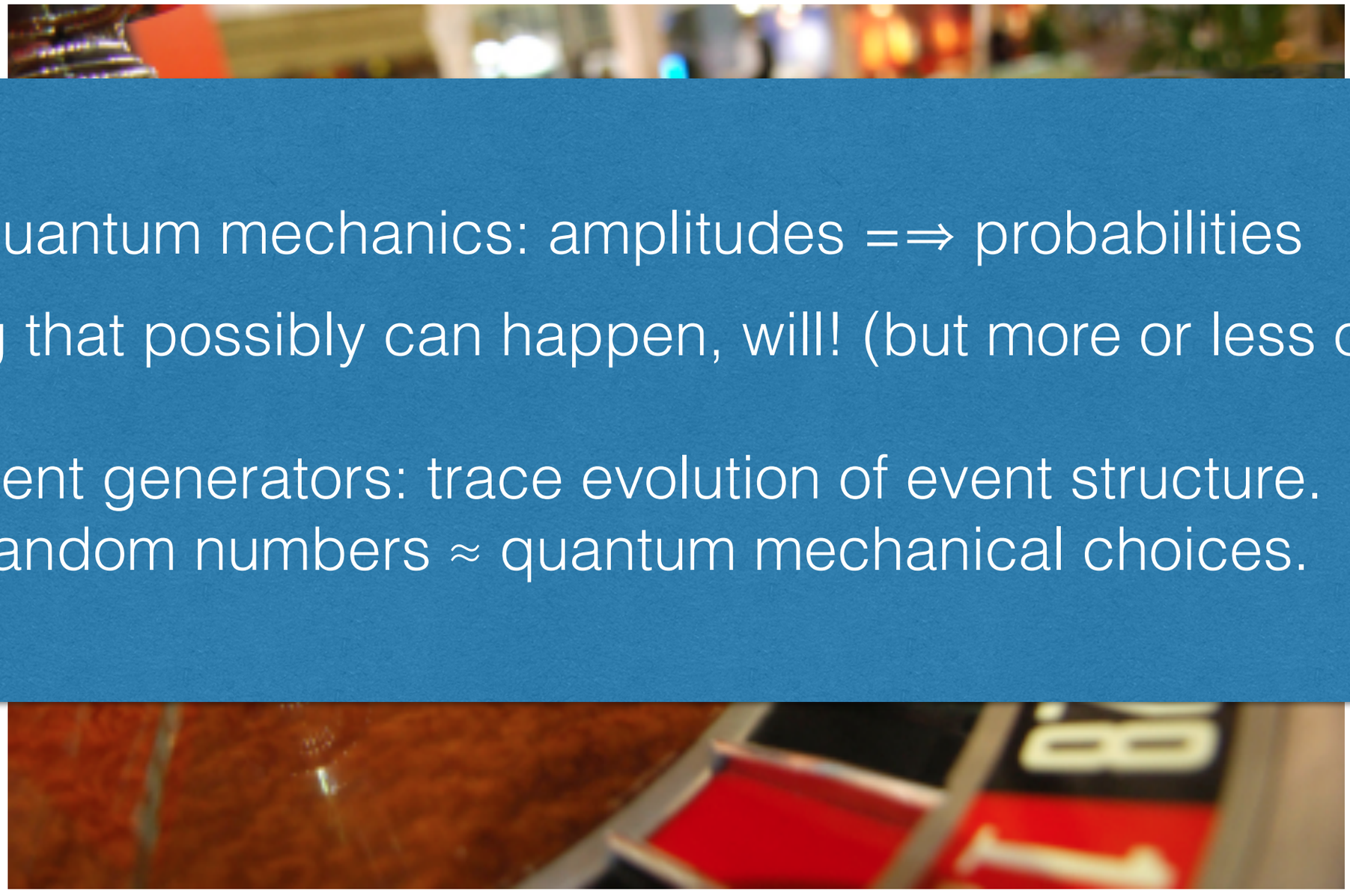
- (also for calibration, corrections, etc)

Monte Carlo?!

It is impossible to calculate and integrate the matrix elements for a large numbers of particles

Monte Carlo?!

It is impossible to calculate and integrate the matrix elements for a large numbers of particles



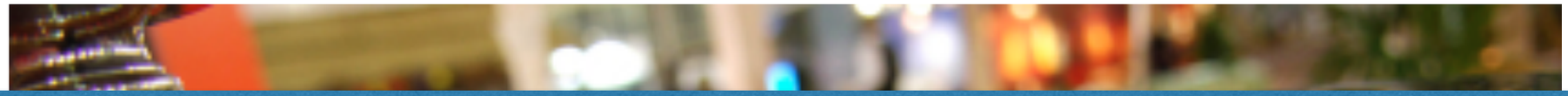
Quantum mechanics: amplitudes ⟹ probabilities

Anything that possibly can happen, will! (but more or less often)

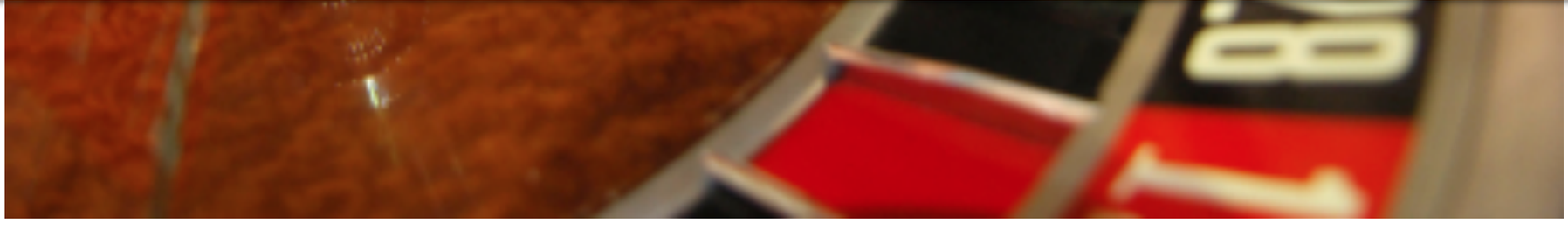
Event generators: trace evolution of event structure.
Random numbers ≈ quantum mechanical choices.

Monte Carlo?!

It is impossible to calculate and integrate the matrix elements for a large numbers of particles



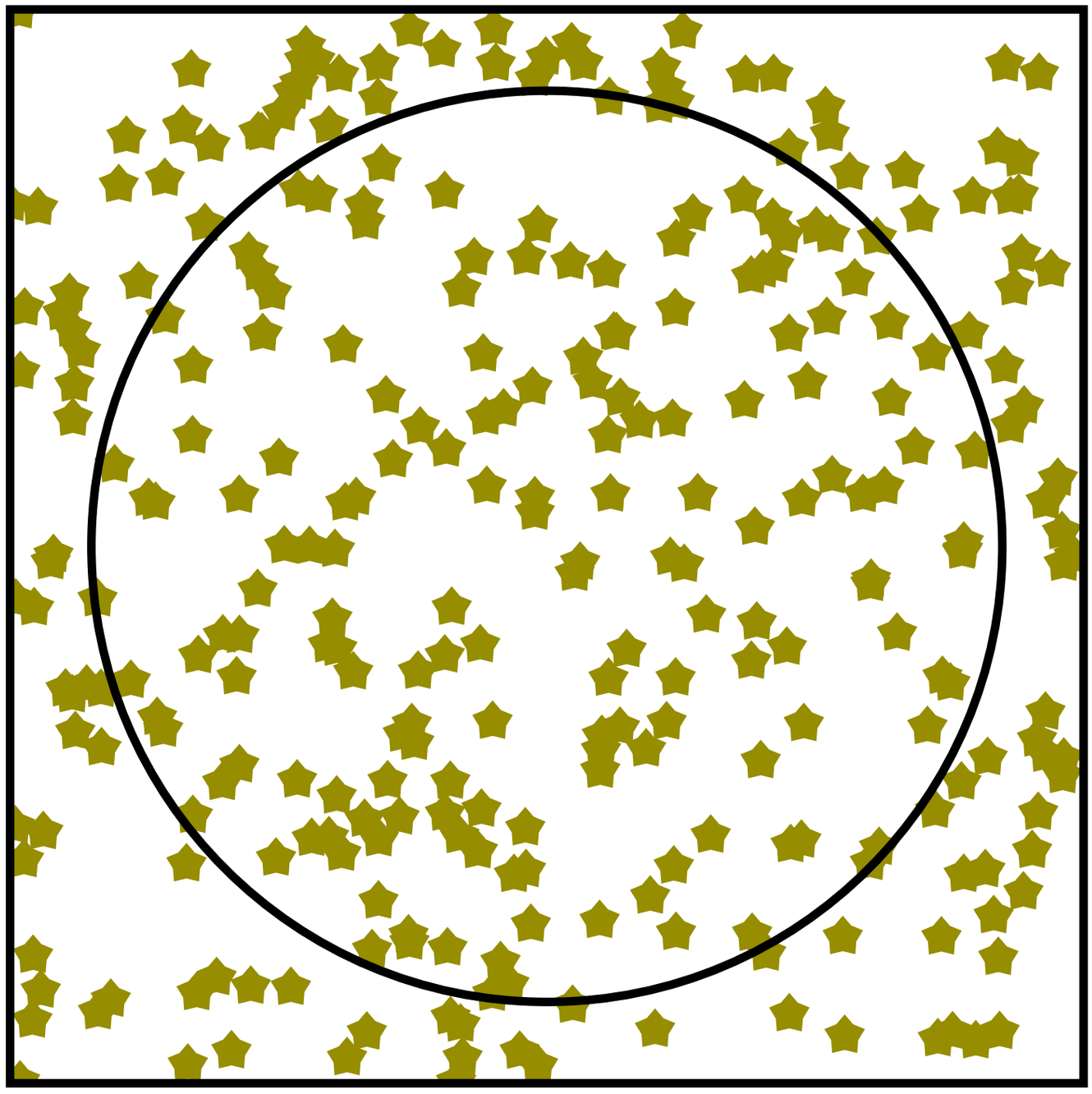
an event with n particles involves $\mathcal{O}(10n)$ random choices,
(flavour, mass, momentum, spin, production vertex, lifetime, ...)
LHC: ~ 100 charged and ~ 200 neutral ($+$ intermediate stages)
$\Longrightarrow$ several thousand choices
(of $\mathcal{O}(100)$ different kinds)

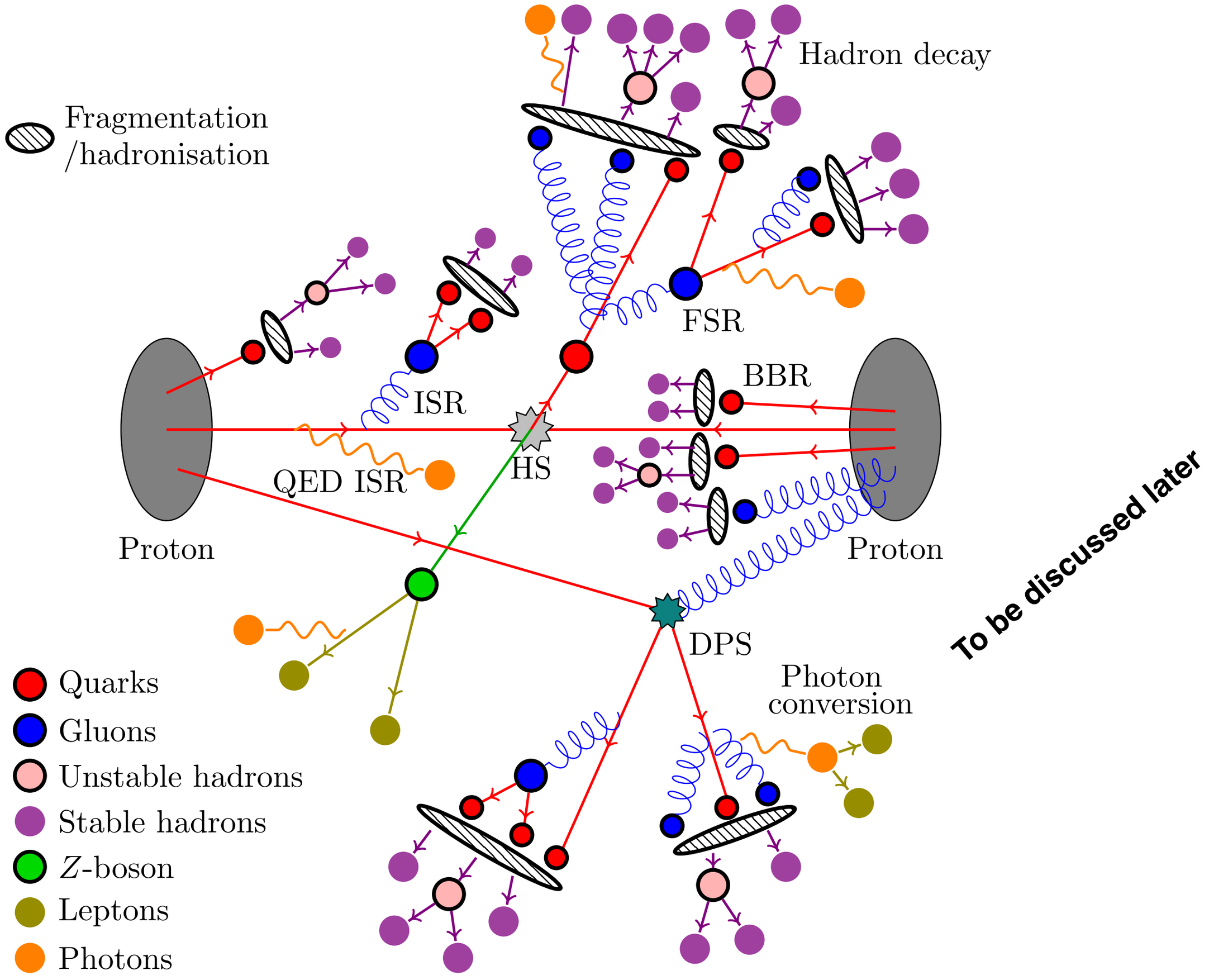


Monte Carlo Method

Find the area of a circle?

Monte Carlo Method




Fragmentation
/hadronisation
Hadron decay
FSR
BBR
ISR
QED ISR
HS
Proton
Proton
DPS
Photon
conversion
Quarks
Gluons
Unstable hadrons
Stable hadrons
Z-boson
Leptons
Photons
To be discussed later

So, where do we use (or try to use) ML?

Analysis is the procedure by which we extract physics information from data (with a lot of help from simulation). Measurements are designed to verify standard model predictions, while the searches are designed to look for beyond the standard model signatures.

Observables

- So we have charged tracks, jets, electrons, muons, photons, and MET.
- From these, we need to form observables which will tell us about either what process occurred during collisions, or whether our simulation programmes can describe the event topology correctly.

Analysis

- A particular physics signal has a specific final state
- Other SM processes can produce same or similar final states (background)
- Apply cuts on observables to retain most of the signal, reject most of the background
- Object and event selection!

Object and Event Selection

- Event: GRL, trigger, final state configuration, topology, MET, H_T
- Object: acceptance, quality criteria
- Decided by signal to background discrimination power, and by right balance!

Significance: $S/\sqrt{N} = S/\sqrt{S+B} \approx S/\sqrt{B}$

Types of backgrounds

- Irreducible: same final state. SM ZZ for H to ZZ.

- Reducible: not the same final state, resulting from misreconstructed processes or misidentified objects. W(lnu)+jets for Z(ll)+jets.

- Combinatorial: random combination of objects looking like the signal. All hadronic ttbar.

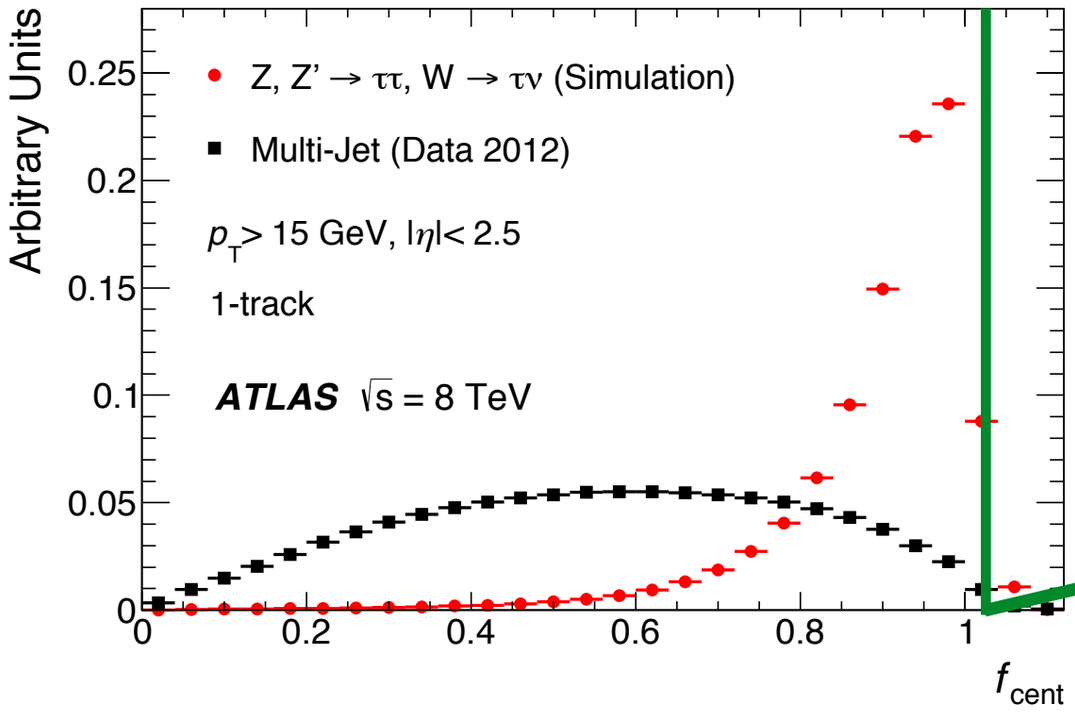
Estimate the backgrounds

Data and/or simulation driven

- Anti-selection/inversion of cuts
- Side-bands/shape extraction by fit
- ABCD method
- …

Signal-Background Discrimination/ROC

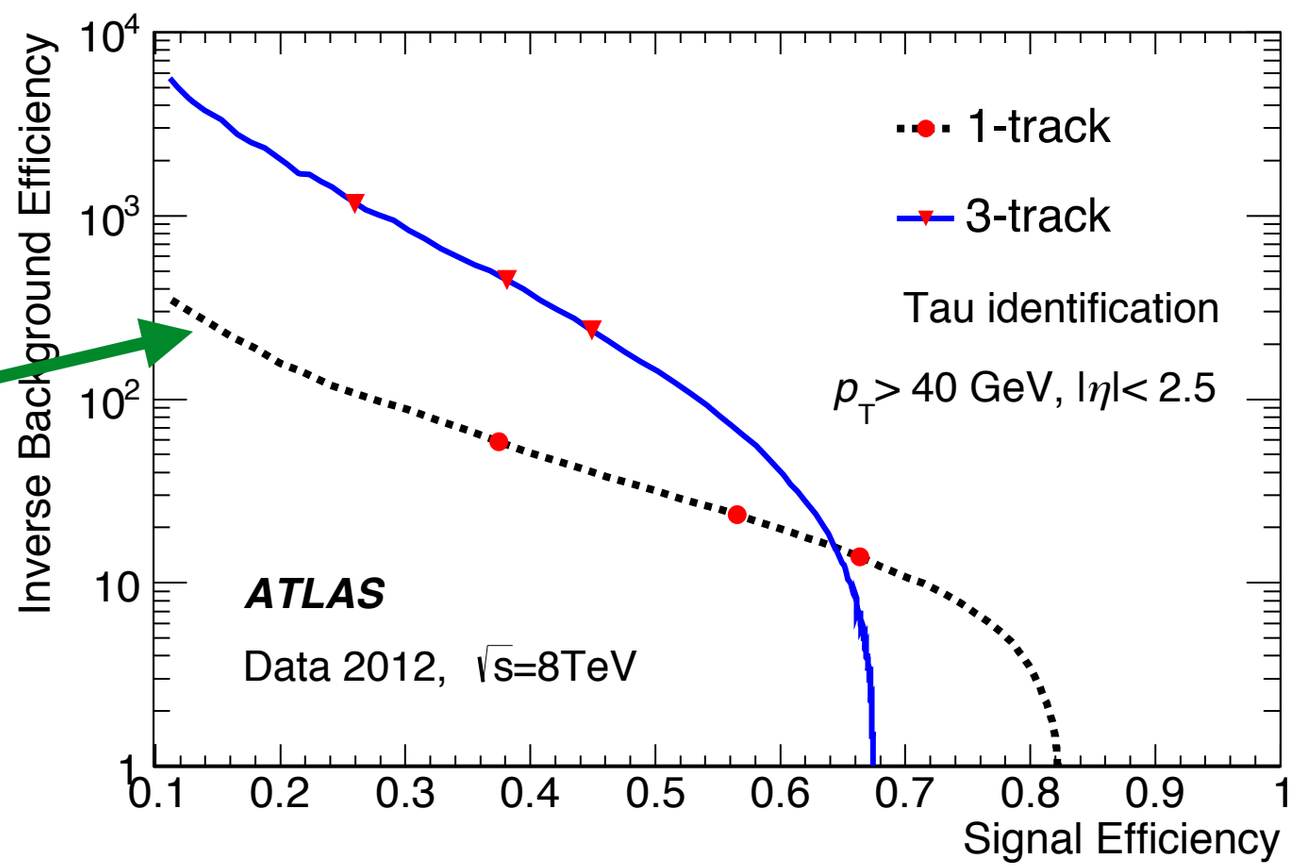
From MC



Scan over the full range

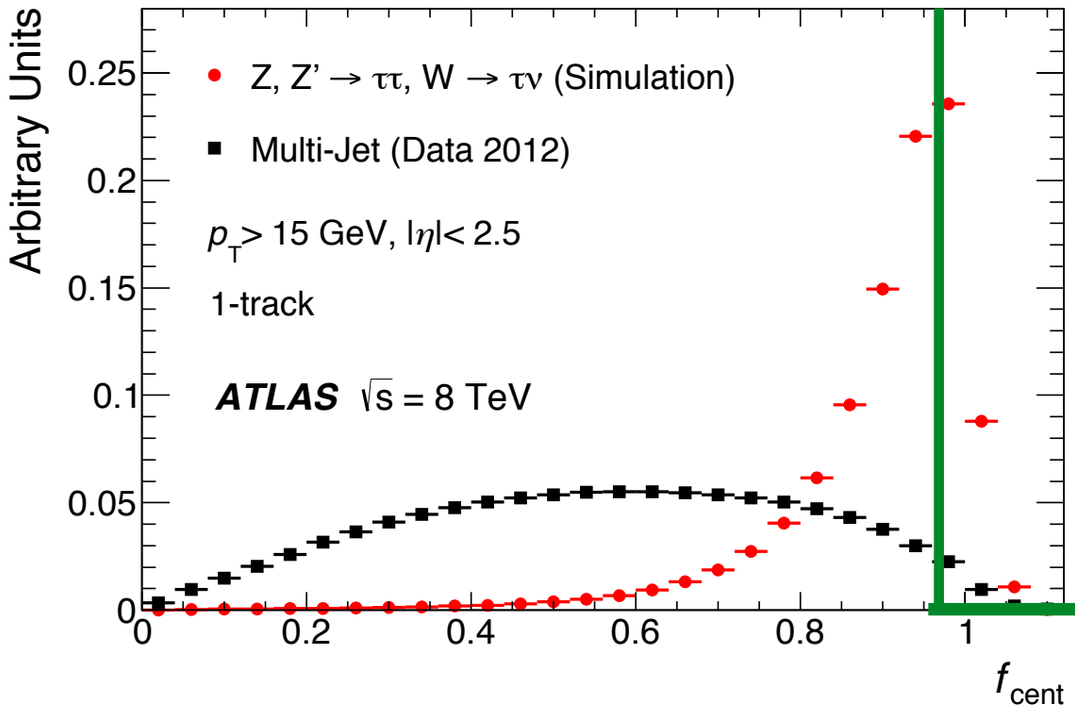
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

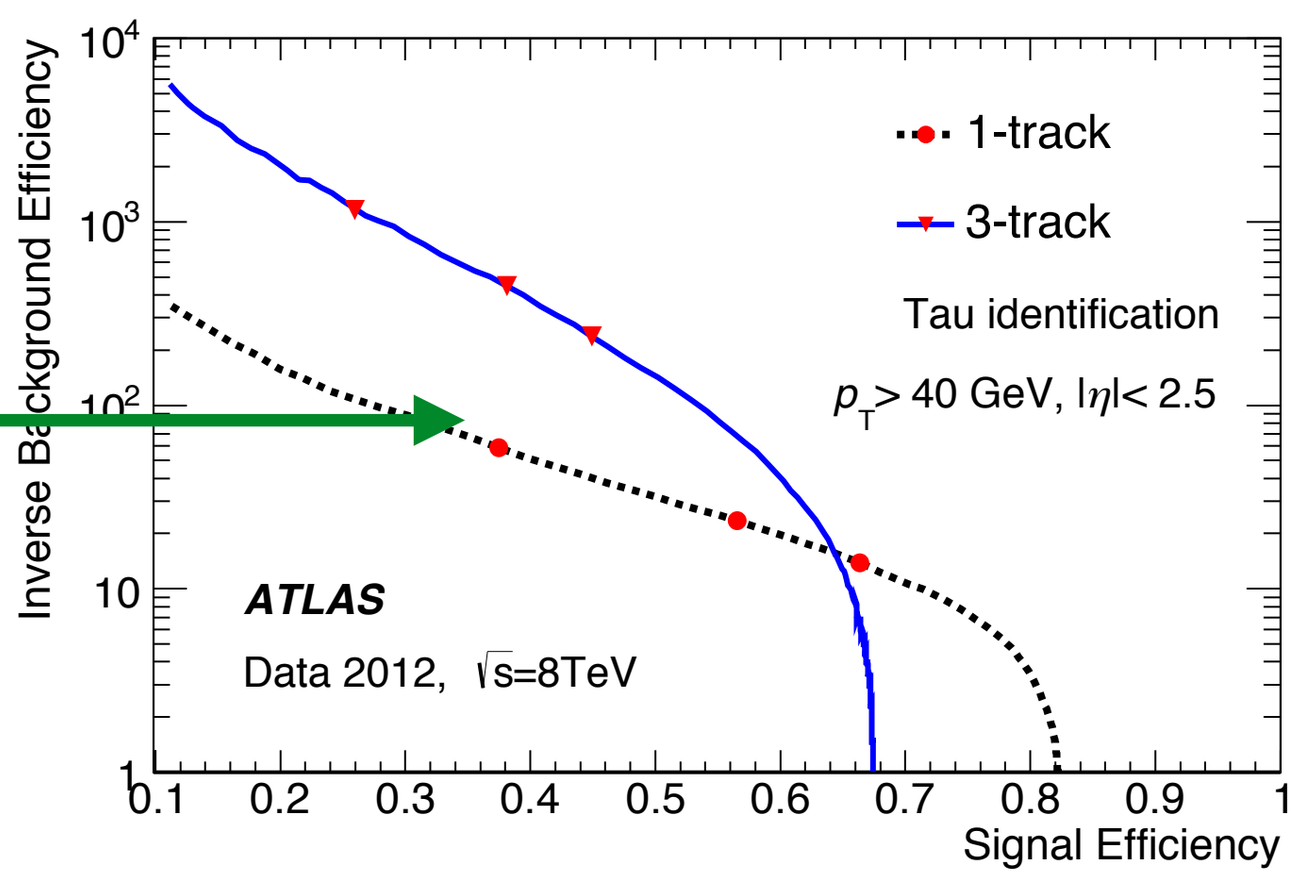
From MC



Scan over the full range

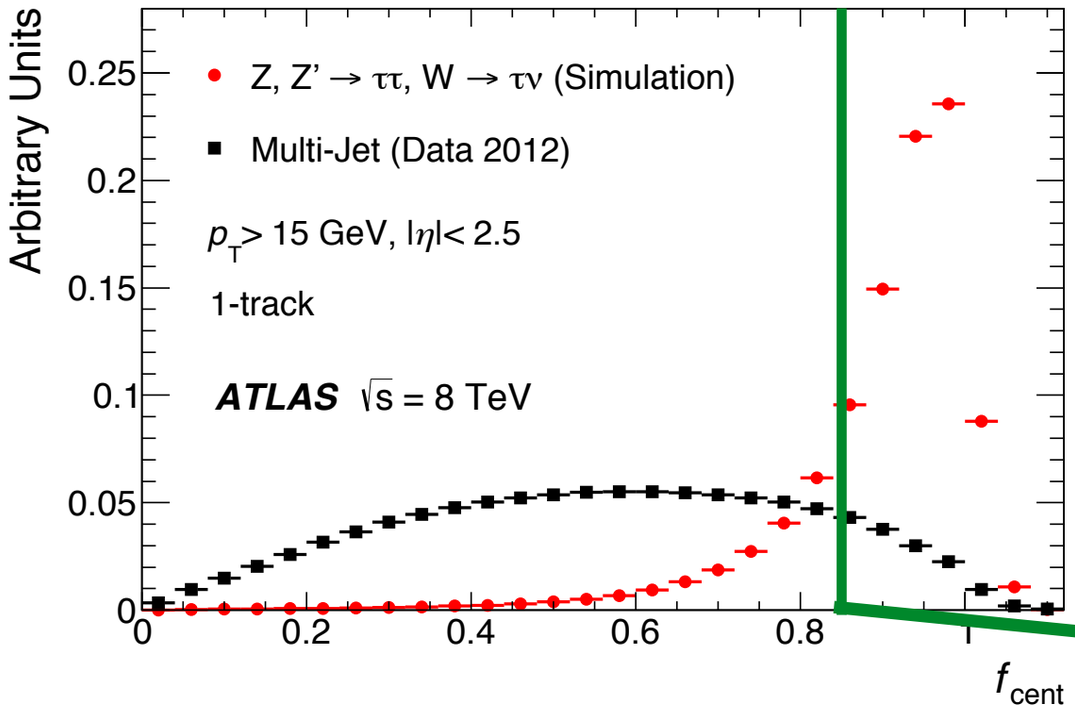
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

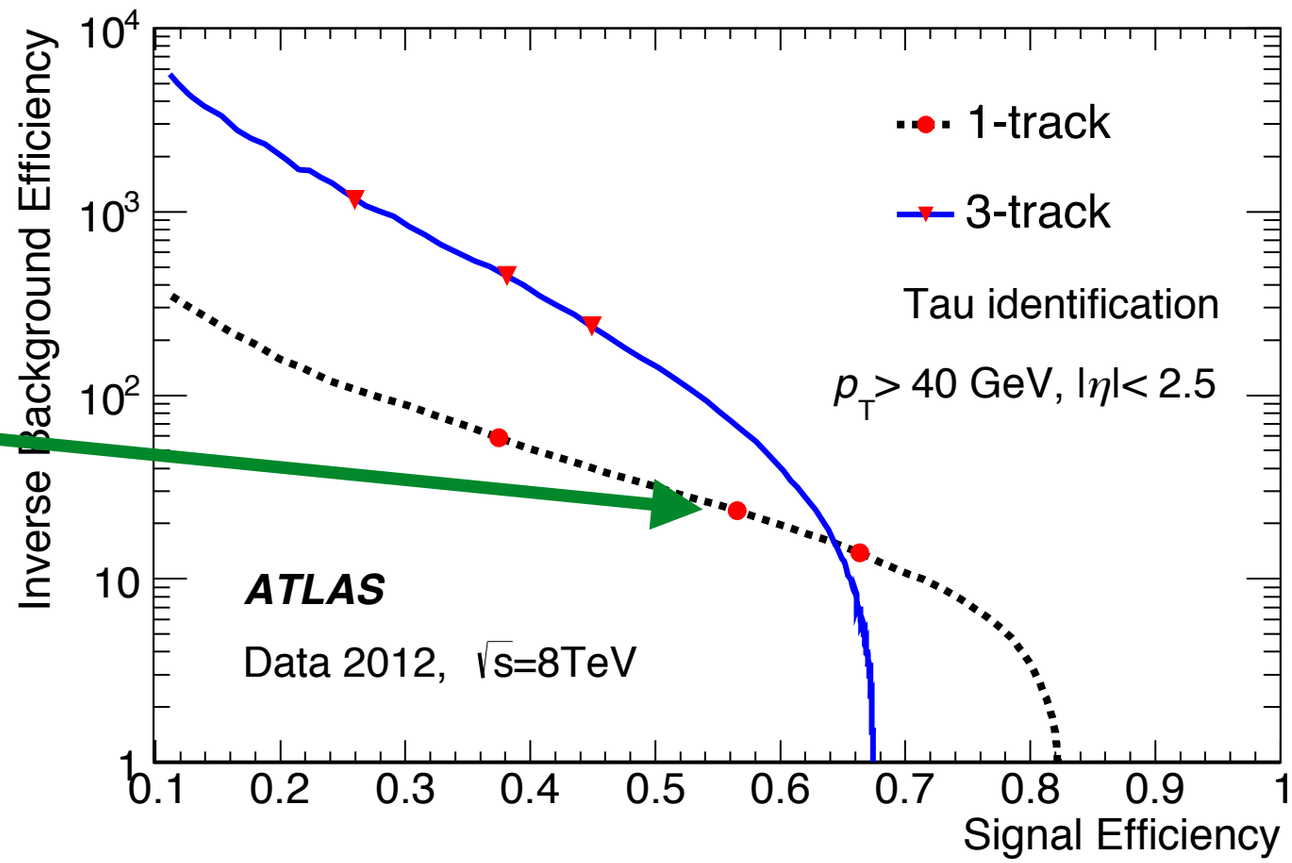
From MC



Scan over the full range

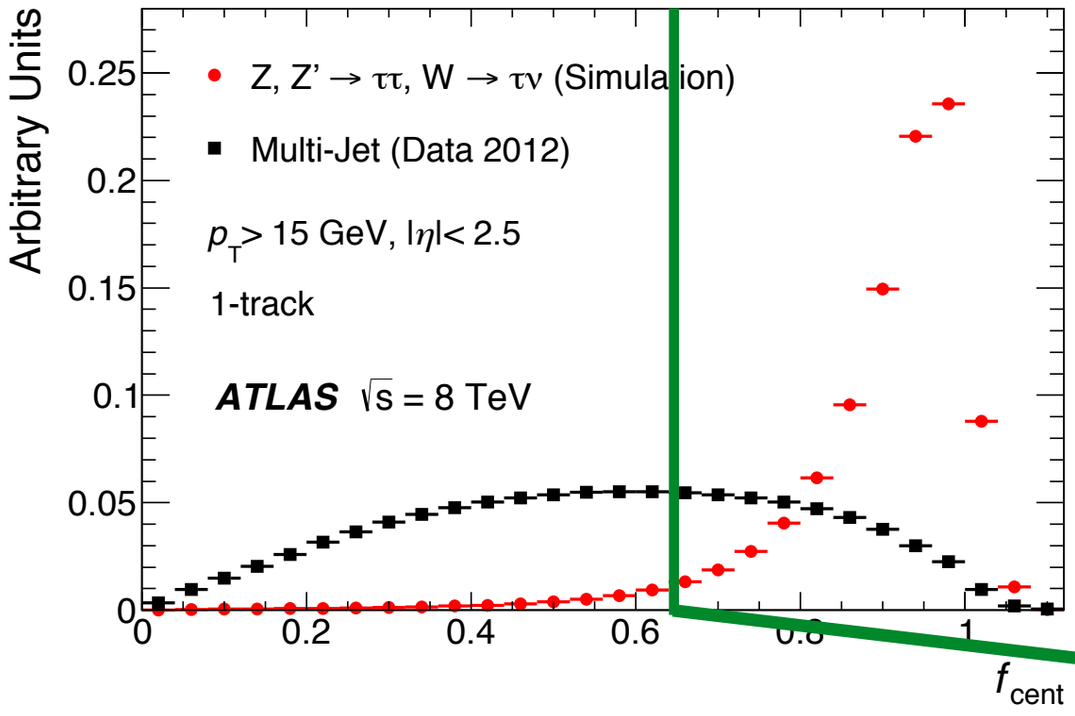
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

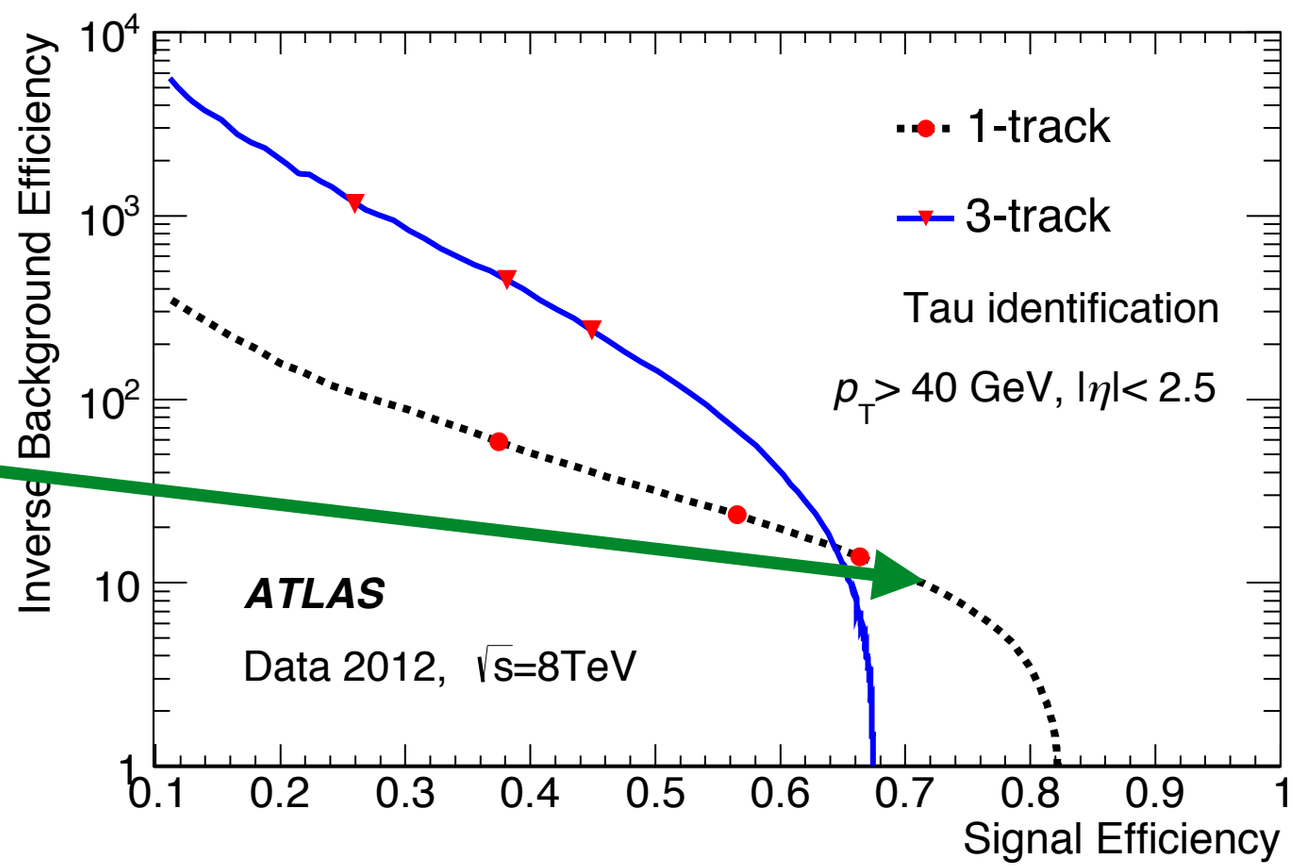
From MC



Scan over the full range

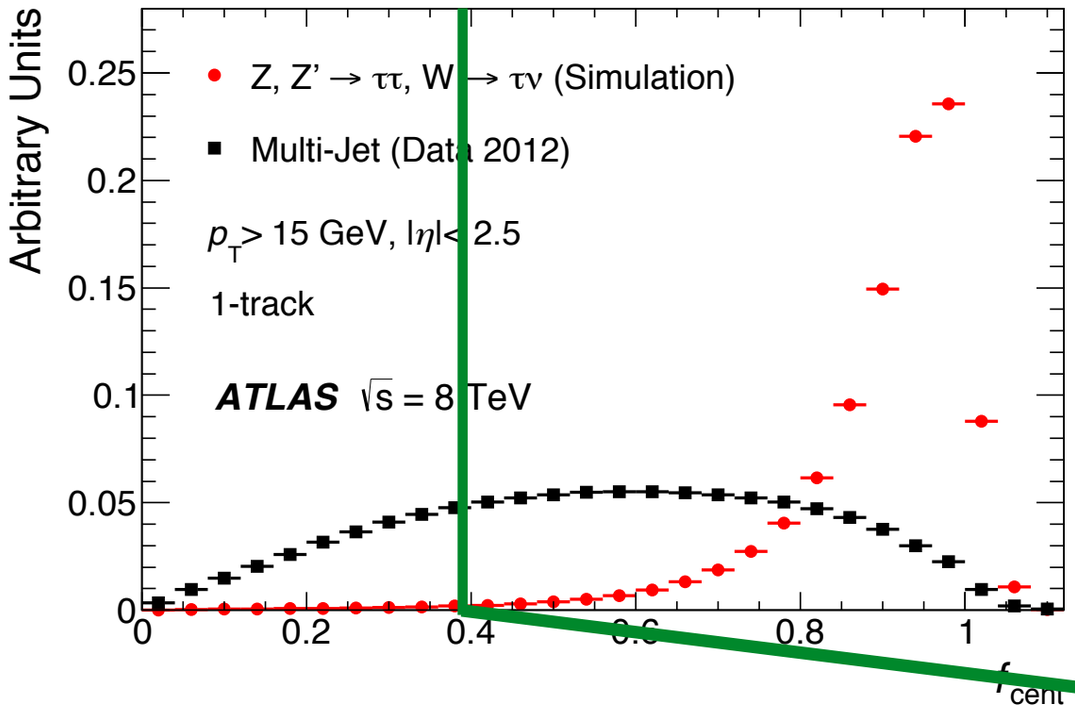
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

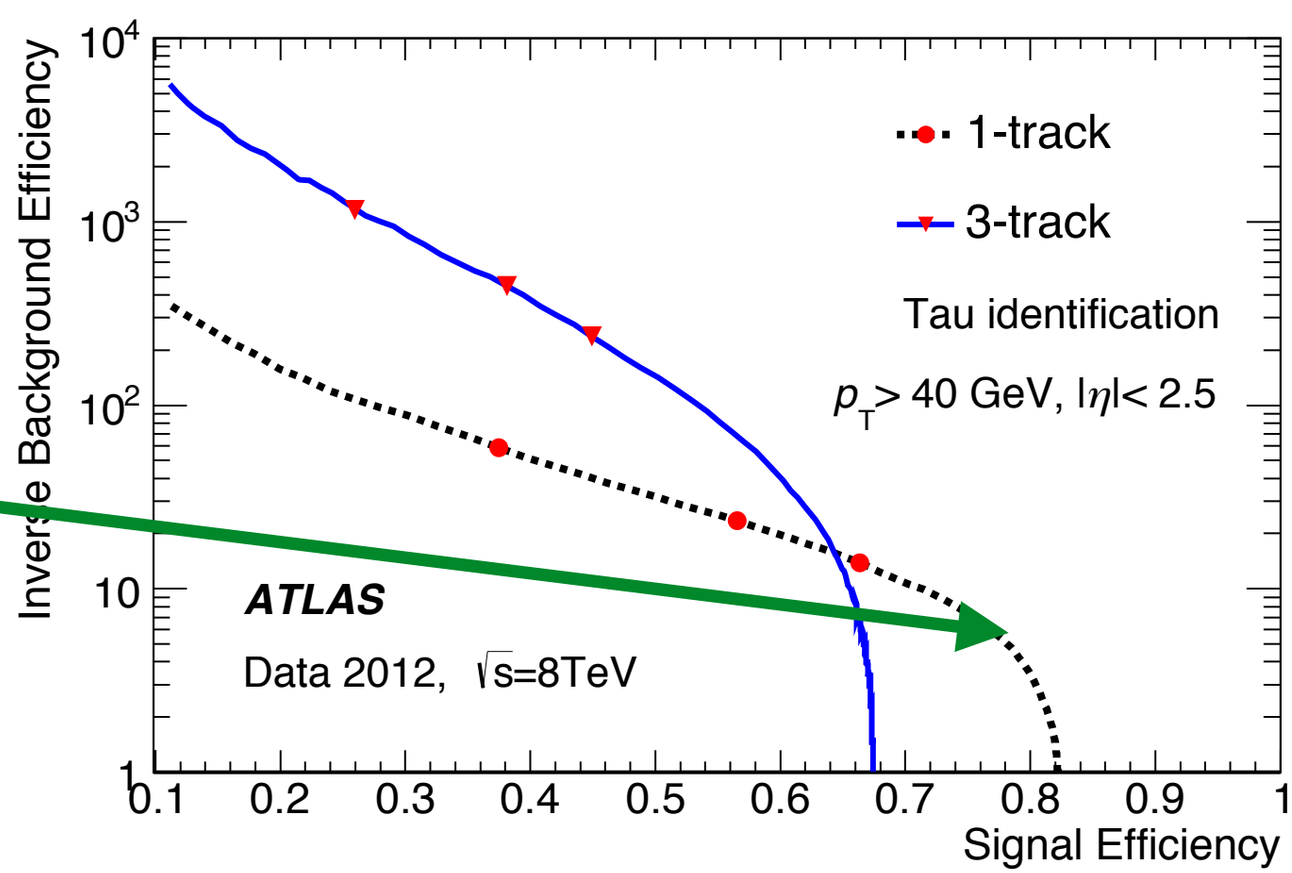
From MC



Scan over the full range

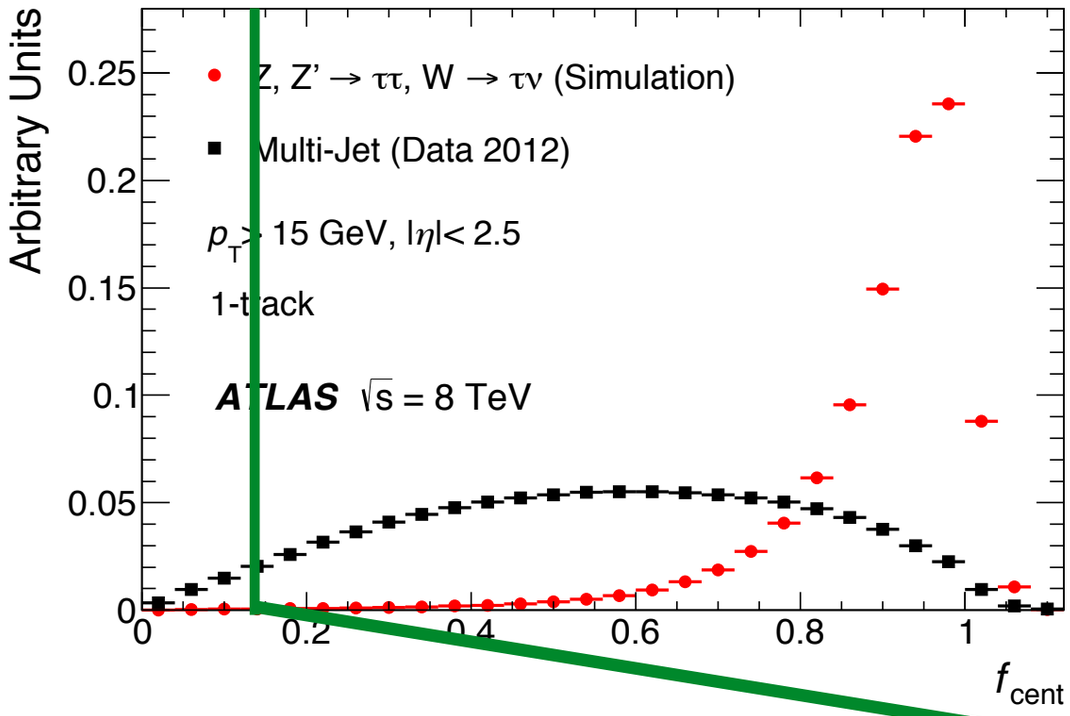
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

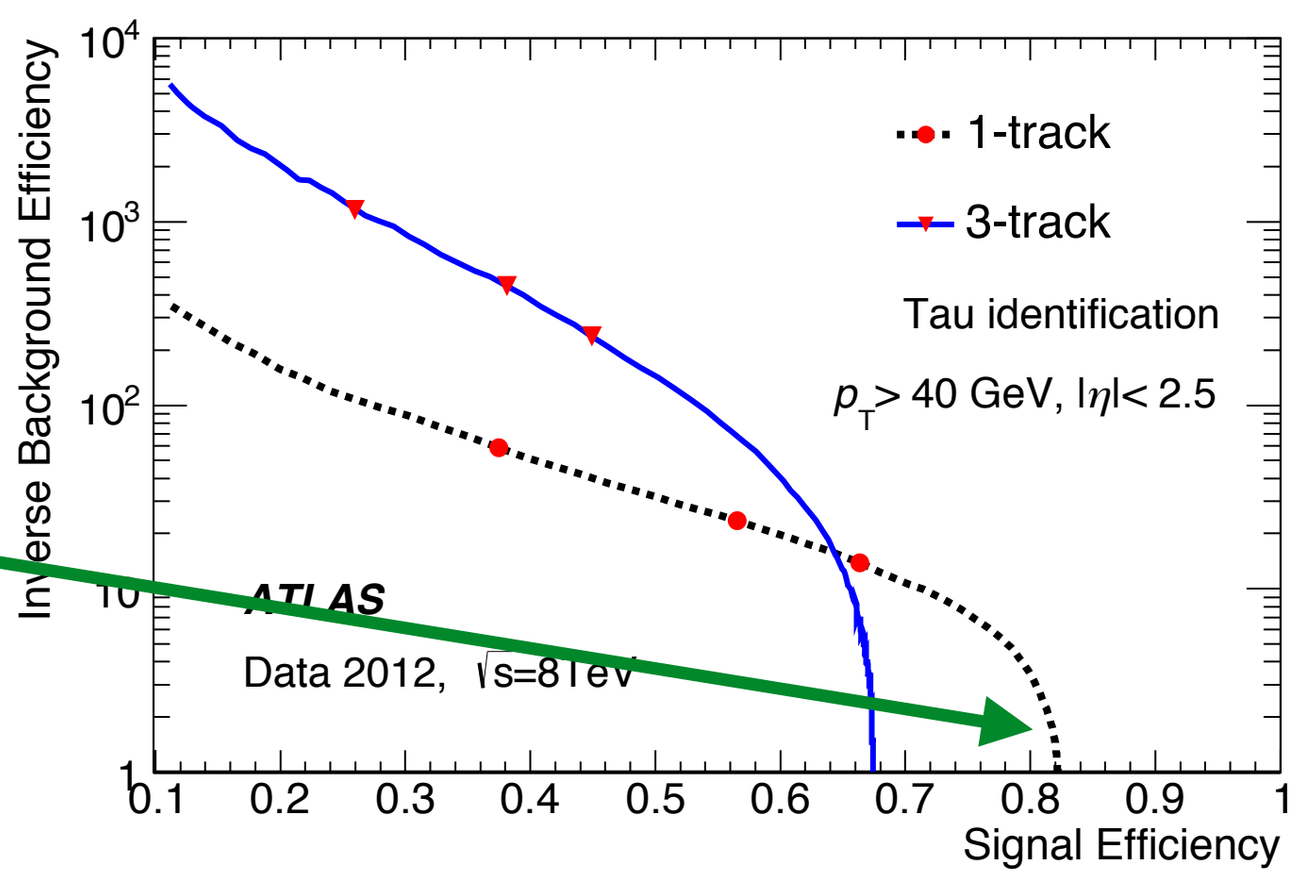
From MC



Scan over the full range

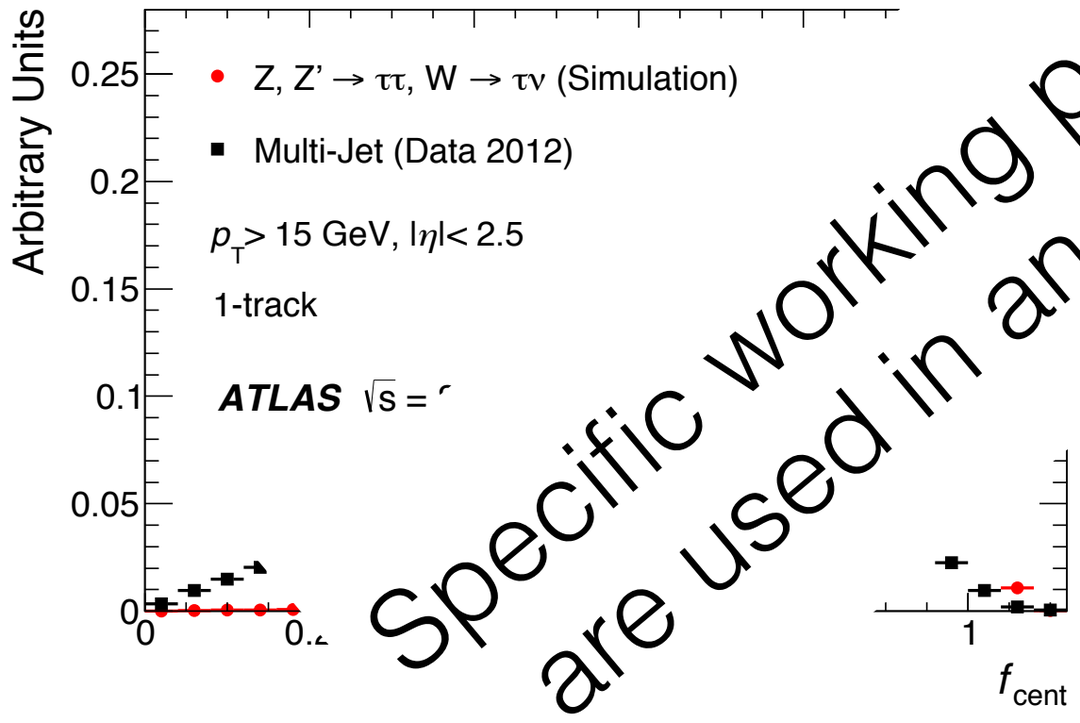
Signal efficiency: fraction of signal events right of the line

Inverse background efficiency: inverse fraction of background events right of the line



Signal-Background Discrimination/ROC

From MC

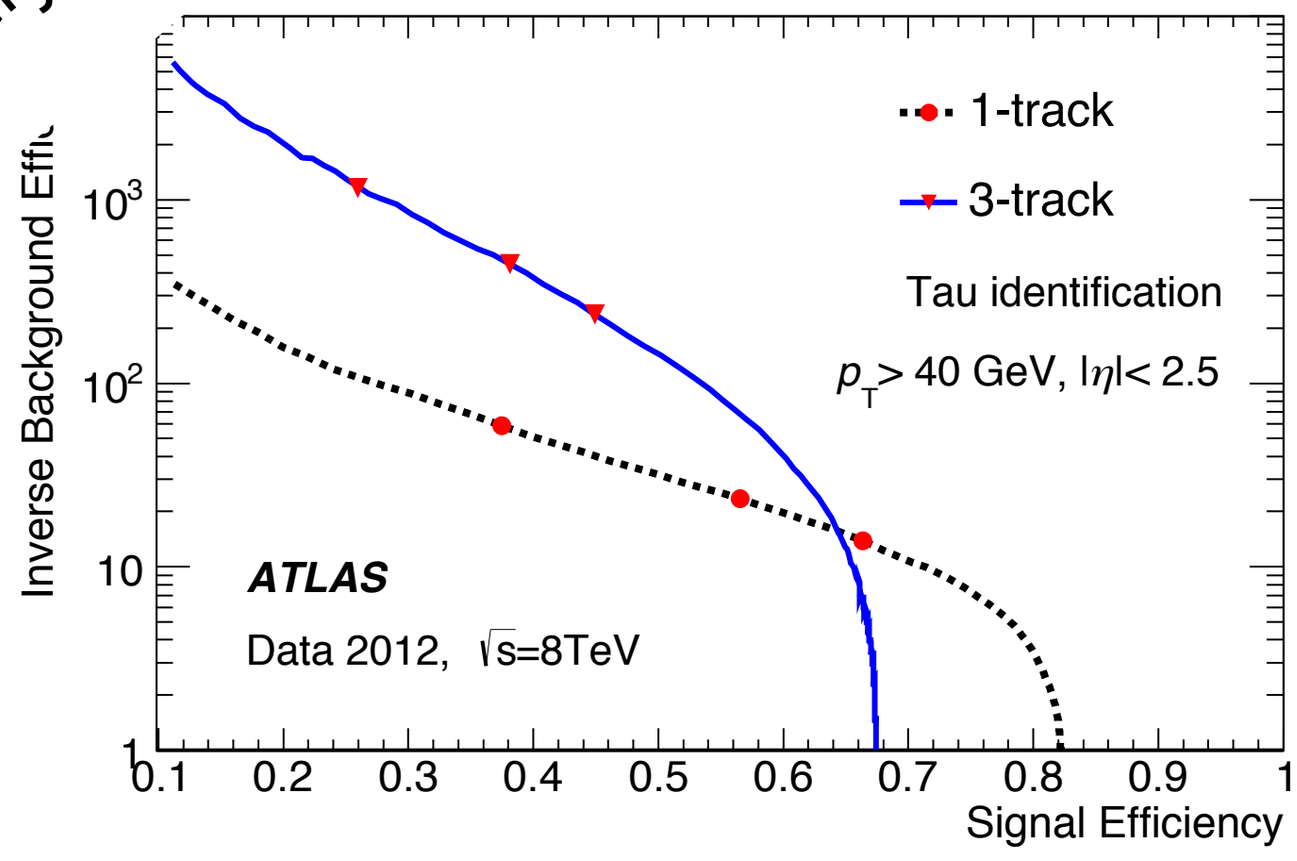


Specific working points (WP) are used in analysis

Scan over the full range

iency: fraction of signal events
right of the line

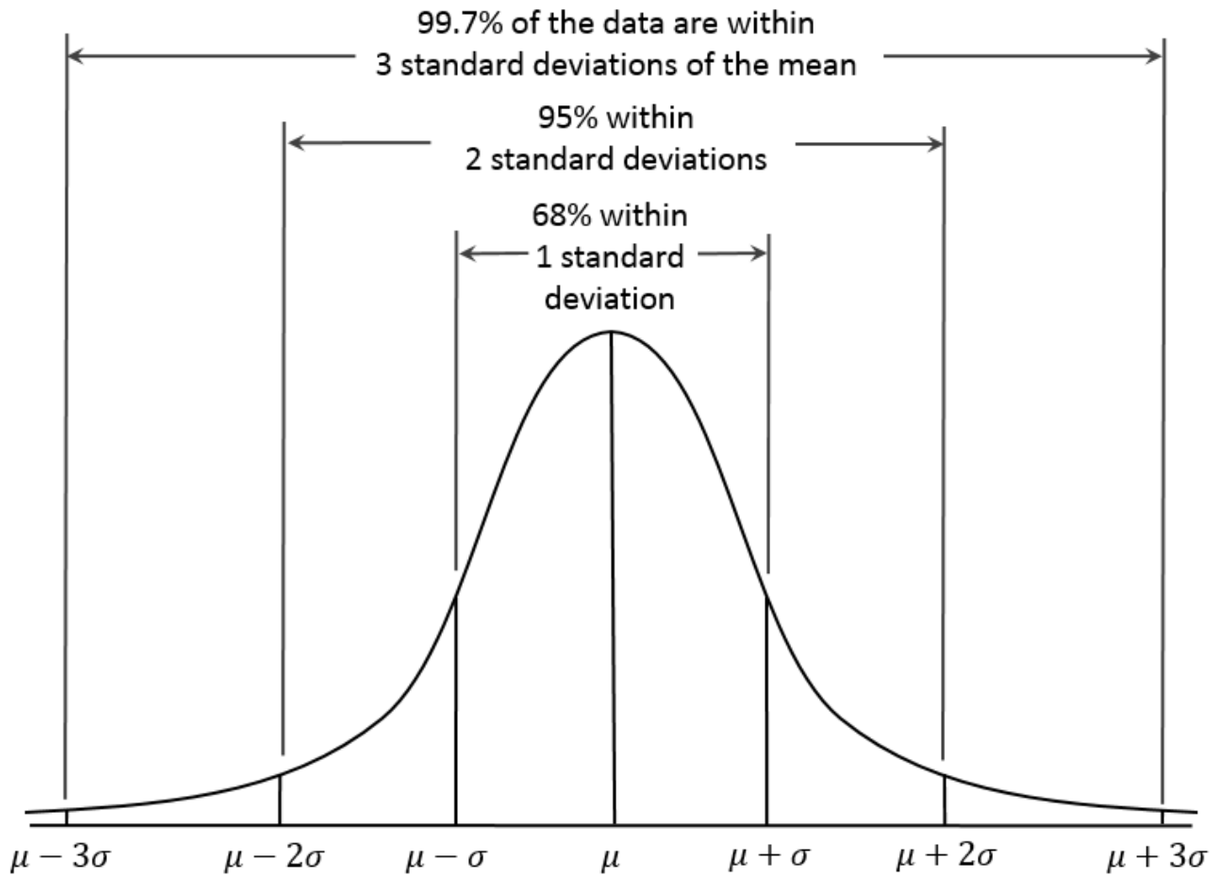
background efficiency:
fraction of background events
right of the line



Uncertainties

- Statistical: random fluctuations, improves with more data. Goes as 1/sqrt(N) (assumed Gaussian)

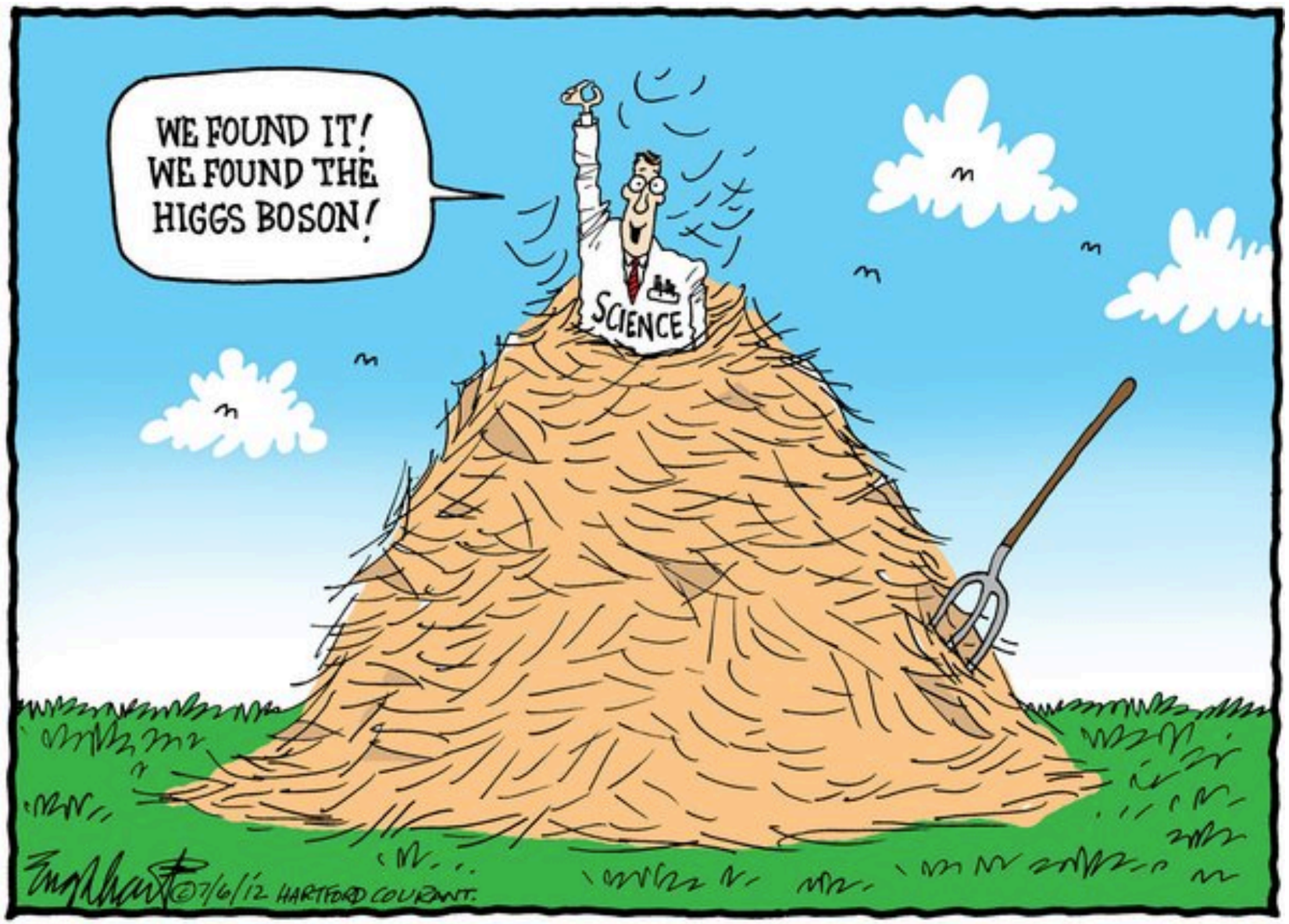
- Systematic: bias in the measurement, challenging to estimate well.



Results presented as:

$$m_W = 80370 \pm 7(stat.) \pm 11(exp.syst.) \pm 14(mod.syst.)MeV = 80370 \pm 19MeV$$

Finally …


WE FOUND IT!
WE FOUND THE
HIGGS BOSON!
SCIENCE
©7/6/12 HARTFORD COURANT.

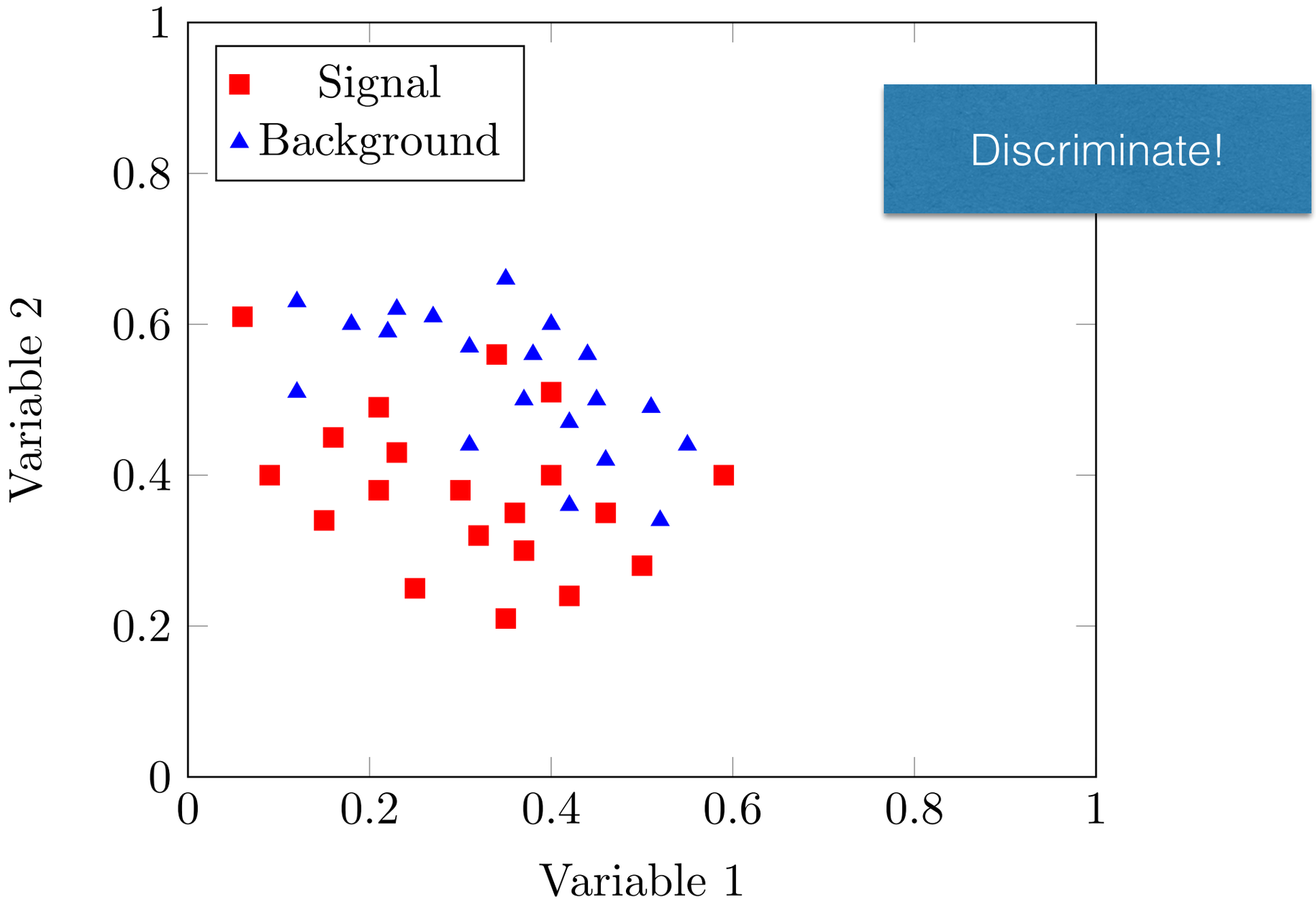
So, where do we use (or try to use) ML?

- Object reconstruction and identification
- Event generation
- Calorimeter simulation
- Triggering
- Jet tagging and jet images
- Signal to background discrimination
- Anomaly detection

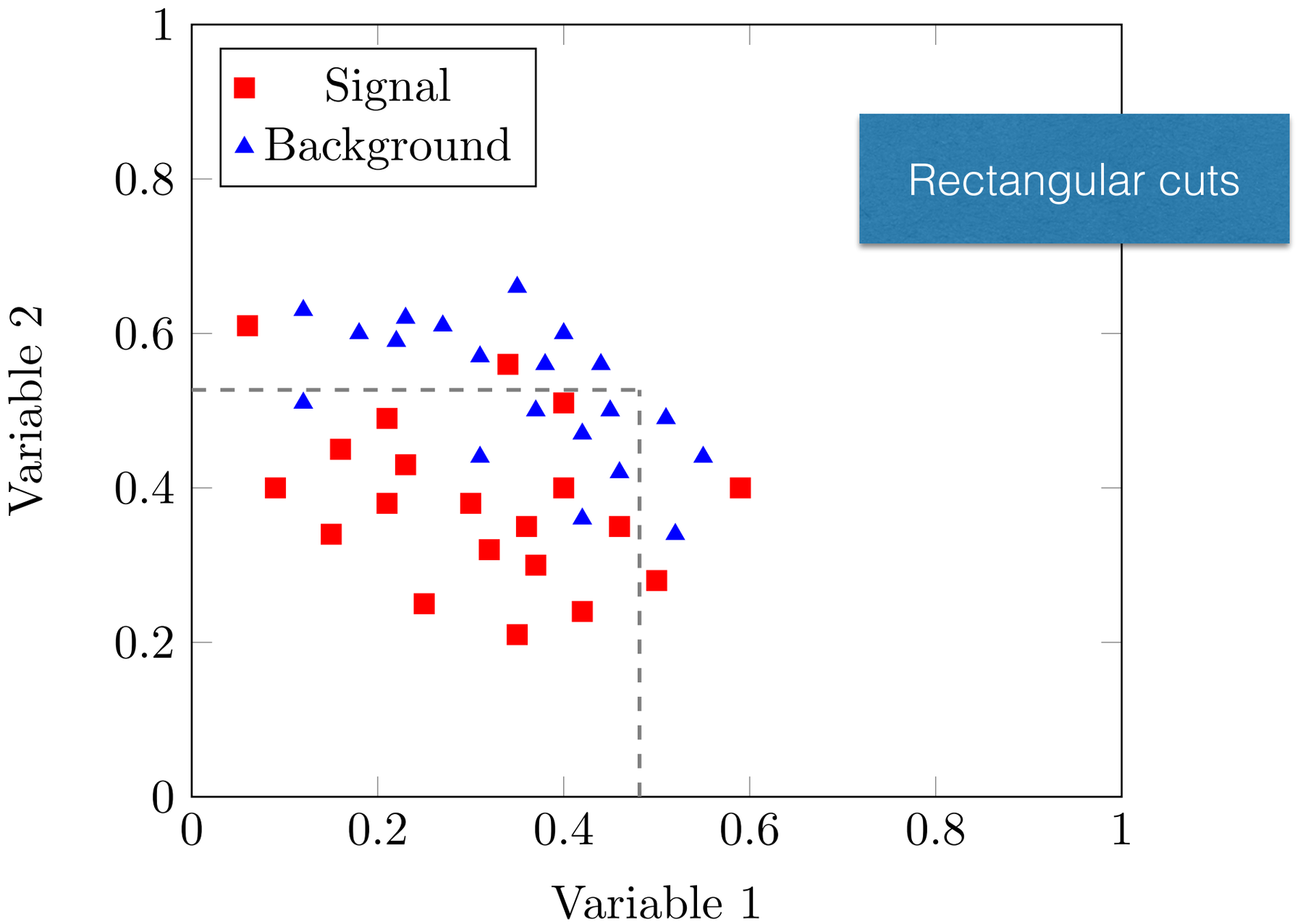
So, where do we use (or try to use) ML?

- **Object reconstruction and identification**
- Event generation
- **Calorimeter simulation**
- **Triggering**
- **Jet tagging and jet images**
- **Signal to background discrimination**
- Anomaly detection

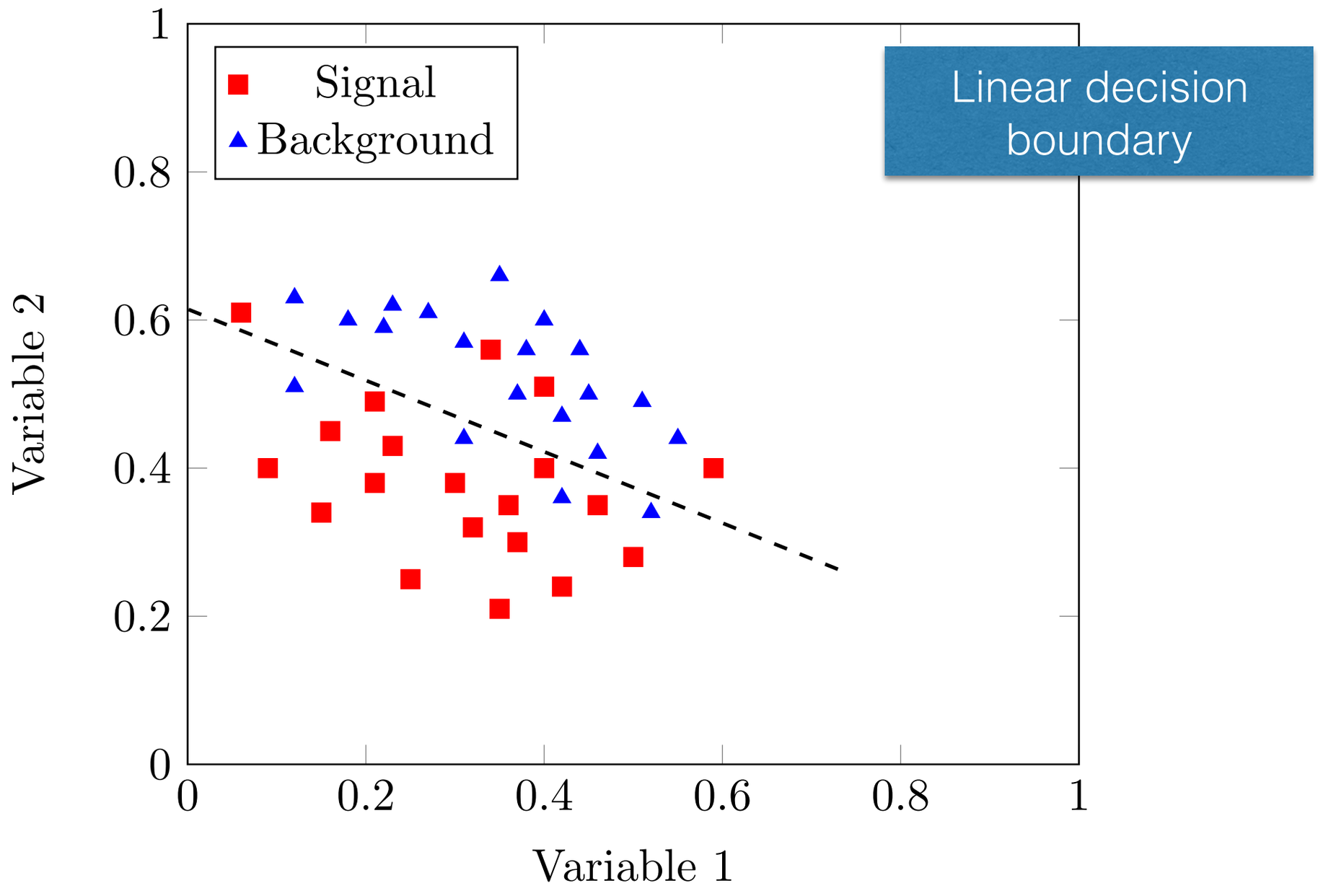
Precursor: Multivariate analysis



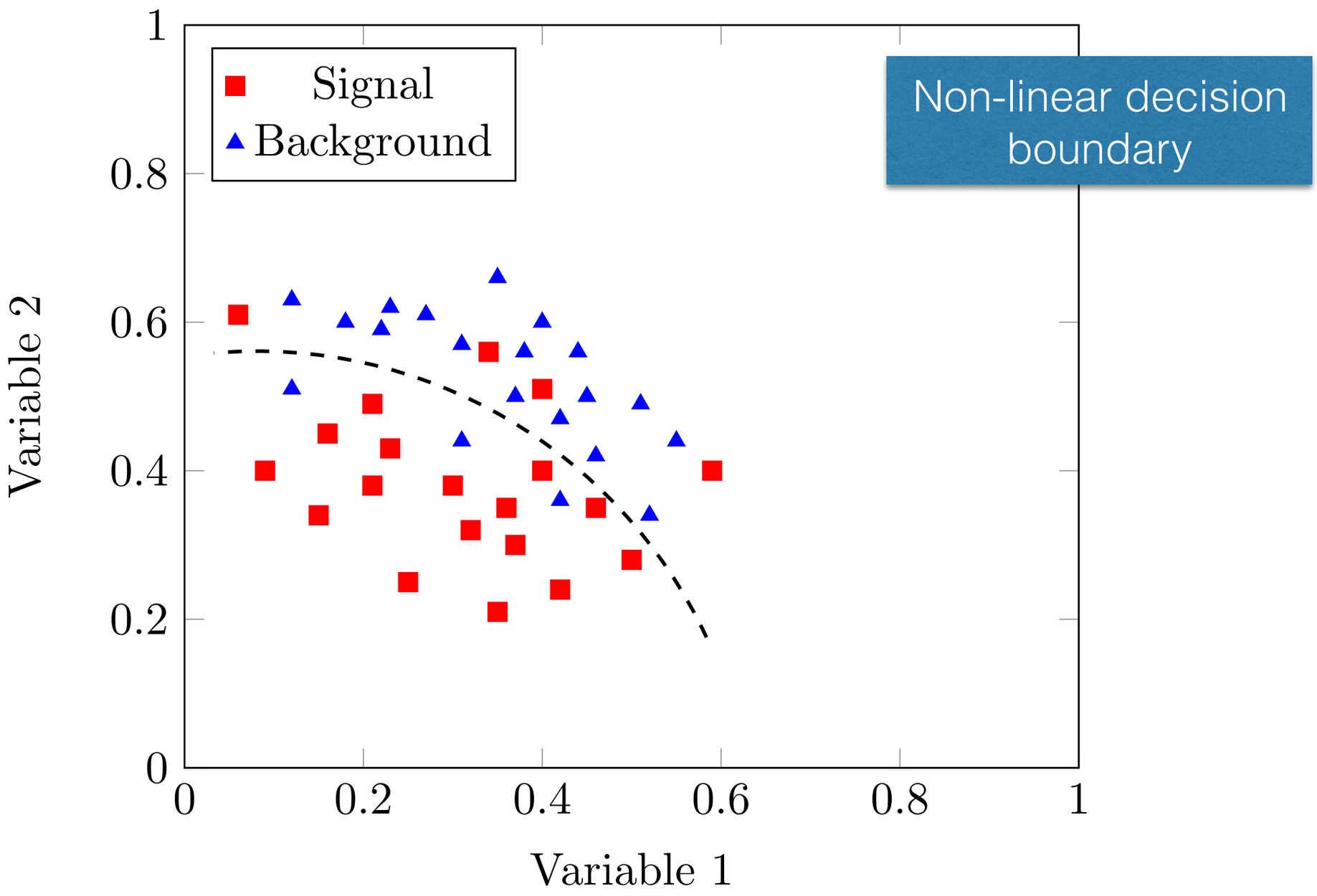
Precursor: Multivariate analysis



Precursor: Multivariate analysis



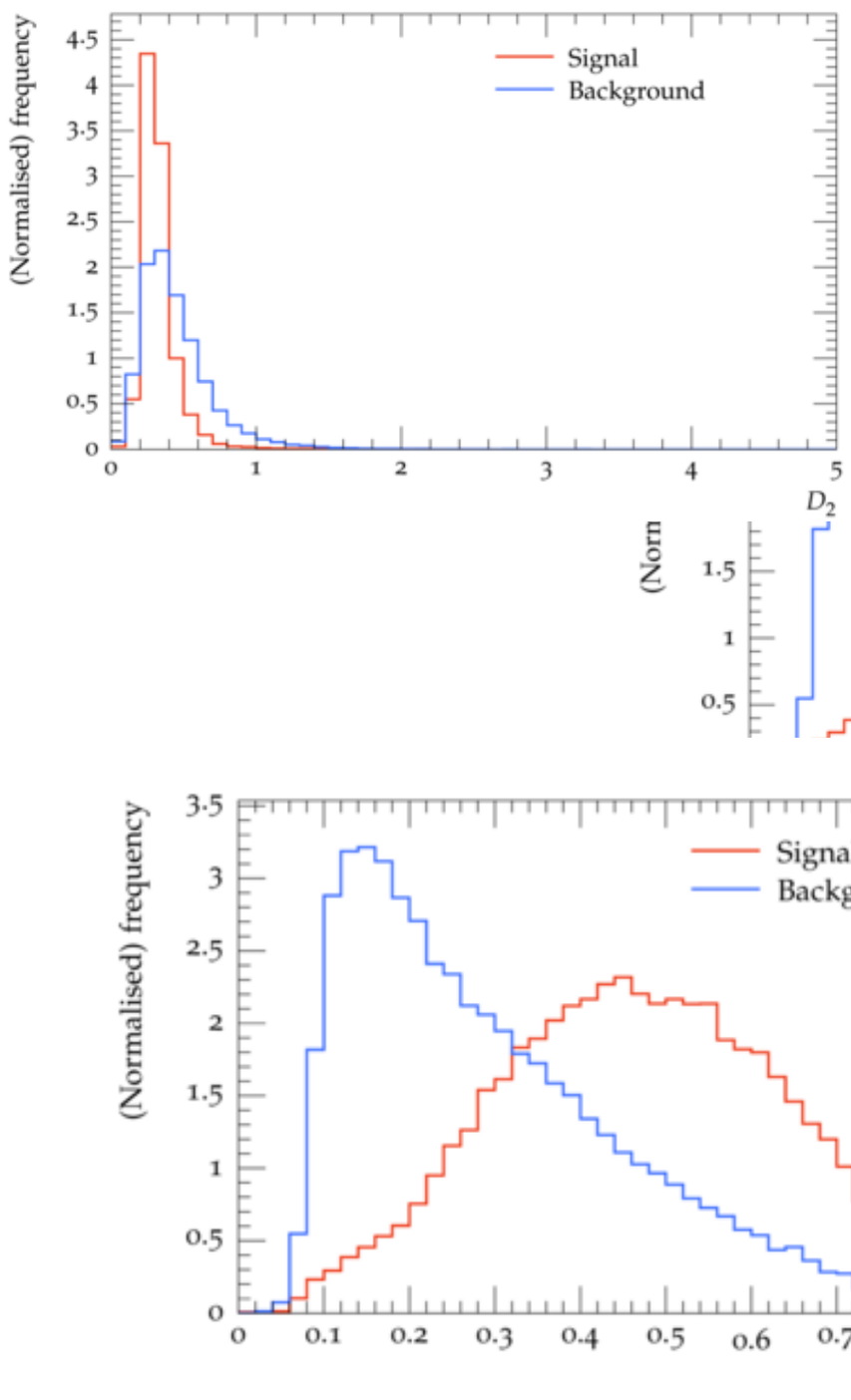
Precursor: Multivariate analysis

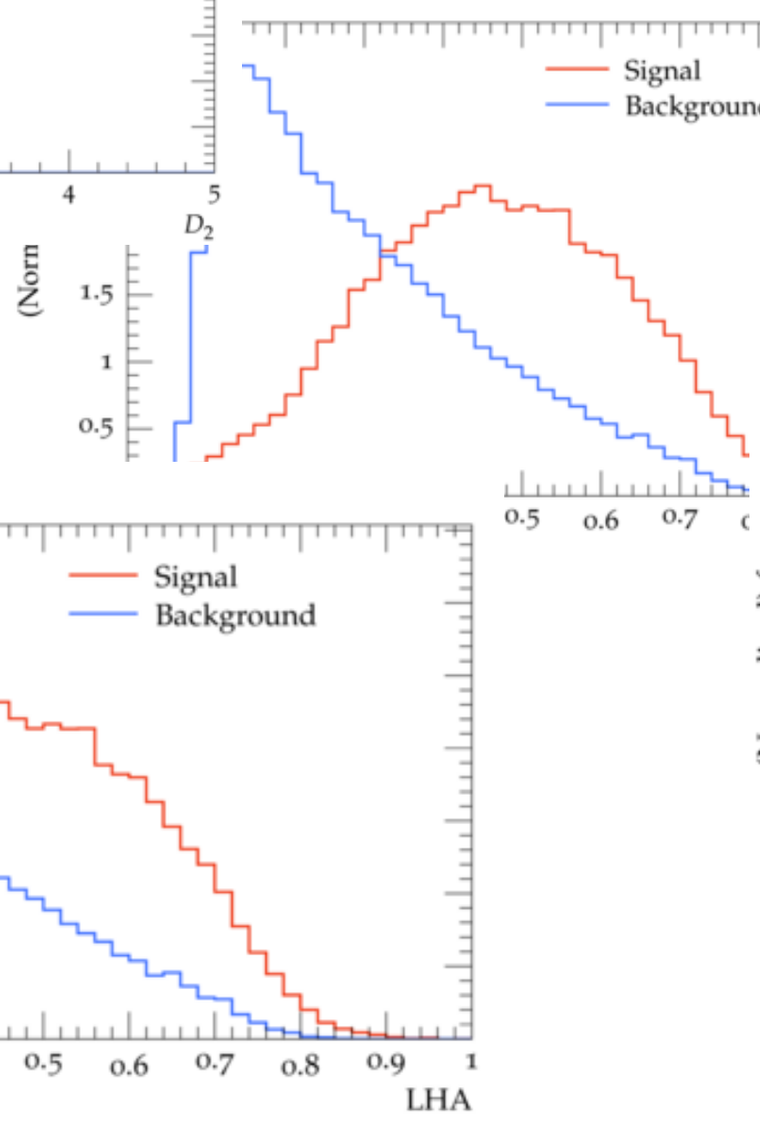


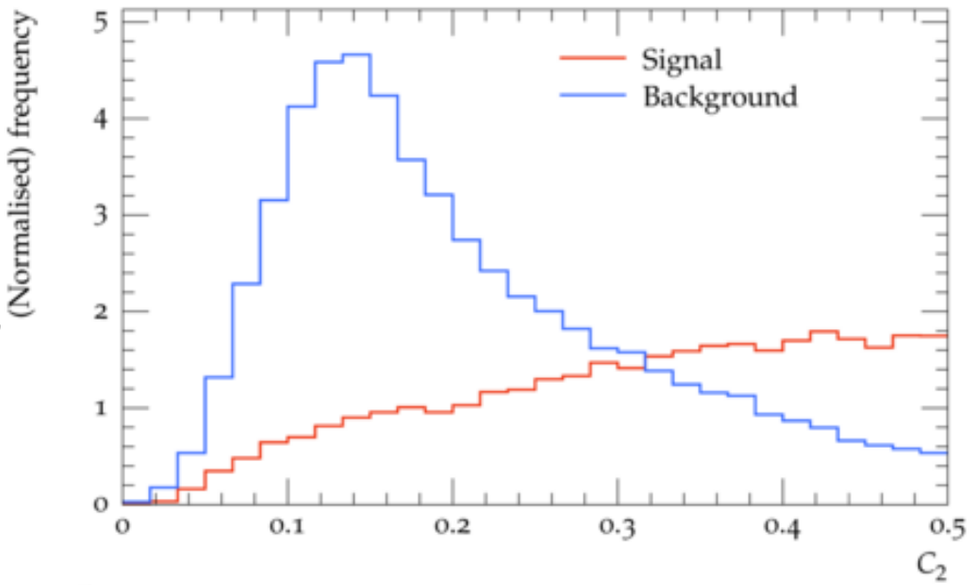
Challenge:

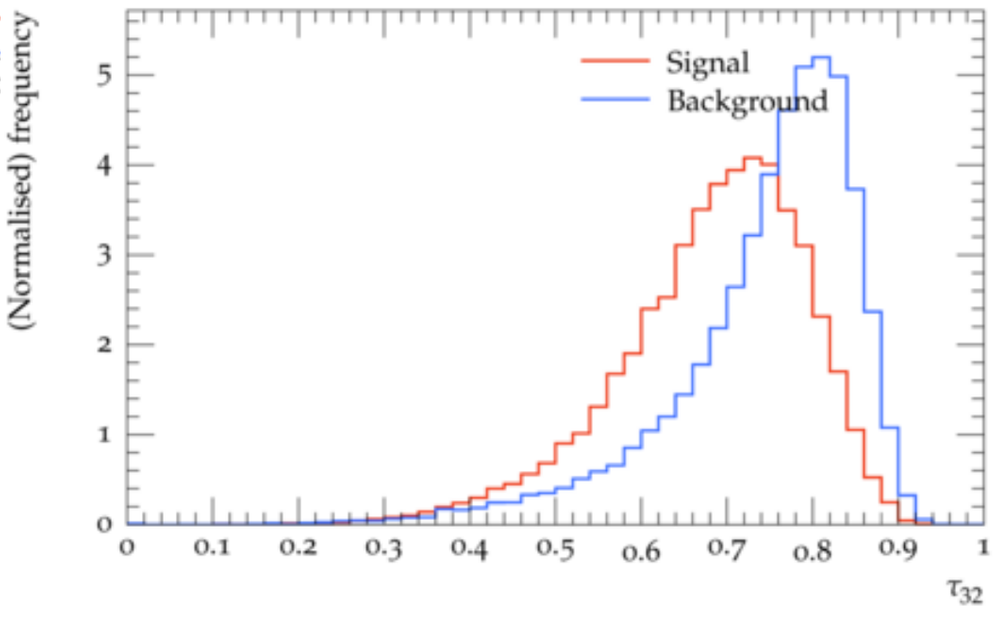
Find the optimal decision boundary in N-dimension!

Signal vs Background









MVA

- N variables used in classification: feature variables
- Correlation reduces dimensionality
- N dimensional constant surface —> mapping to a single discriminating variable
- Actual cut on the variable, as before!

Preprocessing

- Combine or transform the variables to bring out physical features —> called feature extraction

- Example: W-boson transverse mass, scaling by sqrt(s),

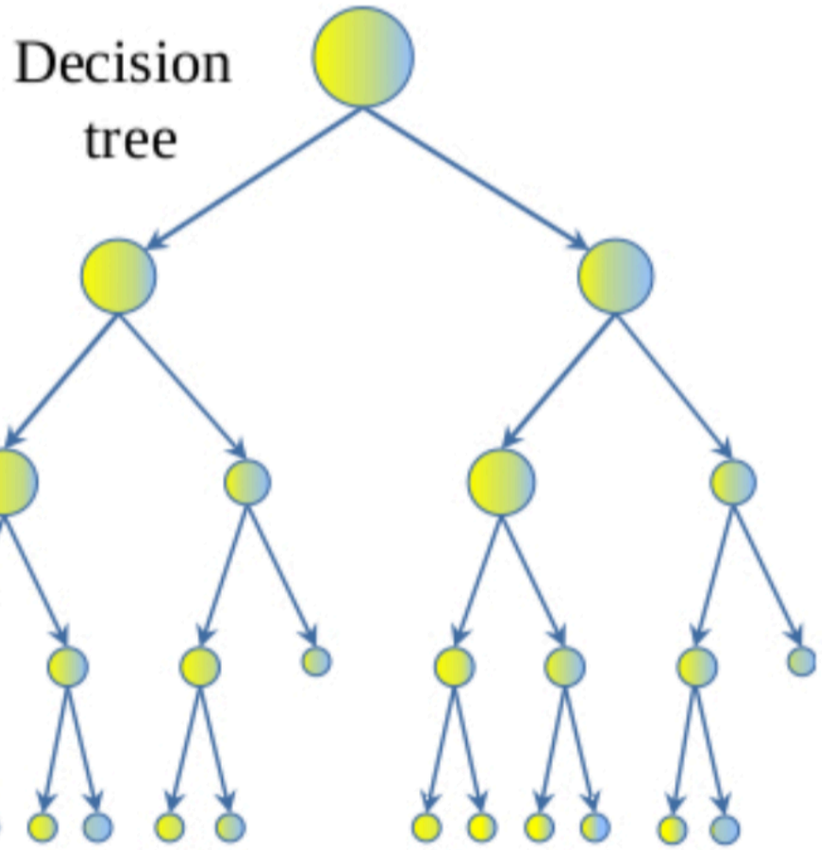
Machine Learning

- Algorithmically find this decision boundary based on *data* (without explicitly programmed)

- Learning: represent the data by an approximate functional form (whether or not such a form exists is immaterial) between input variables x and output variables y

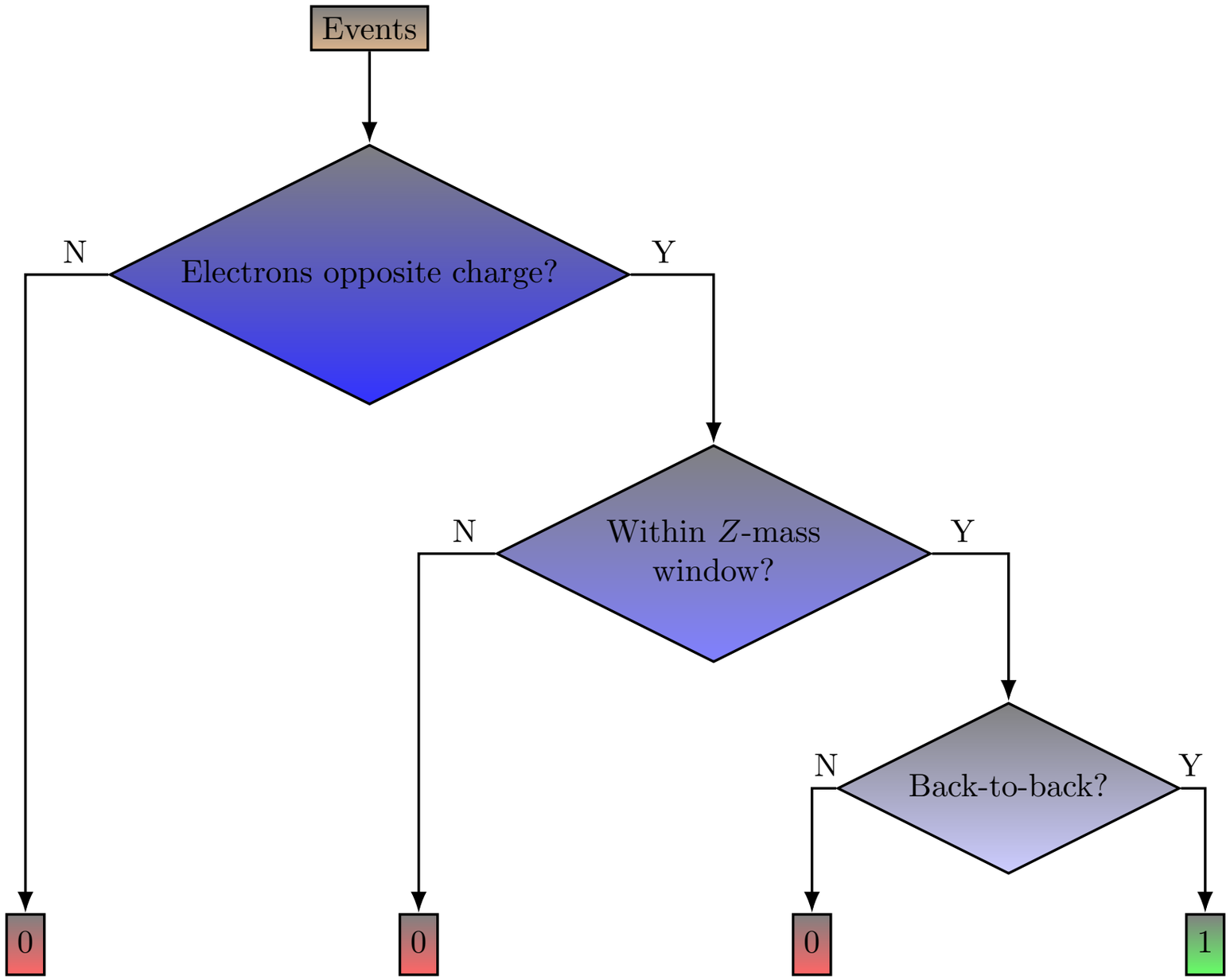
- Use that predict behaviour of future similar datasets from x' to y'.

Decision Trees



- For classification problems
- Sequence of criteria
- Arrive at a decision minimising contamination

Decision Trees



Growing a Tree

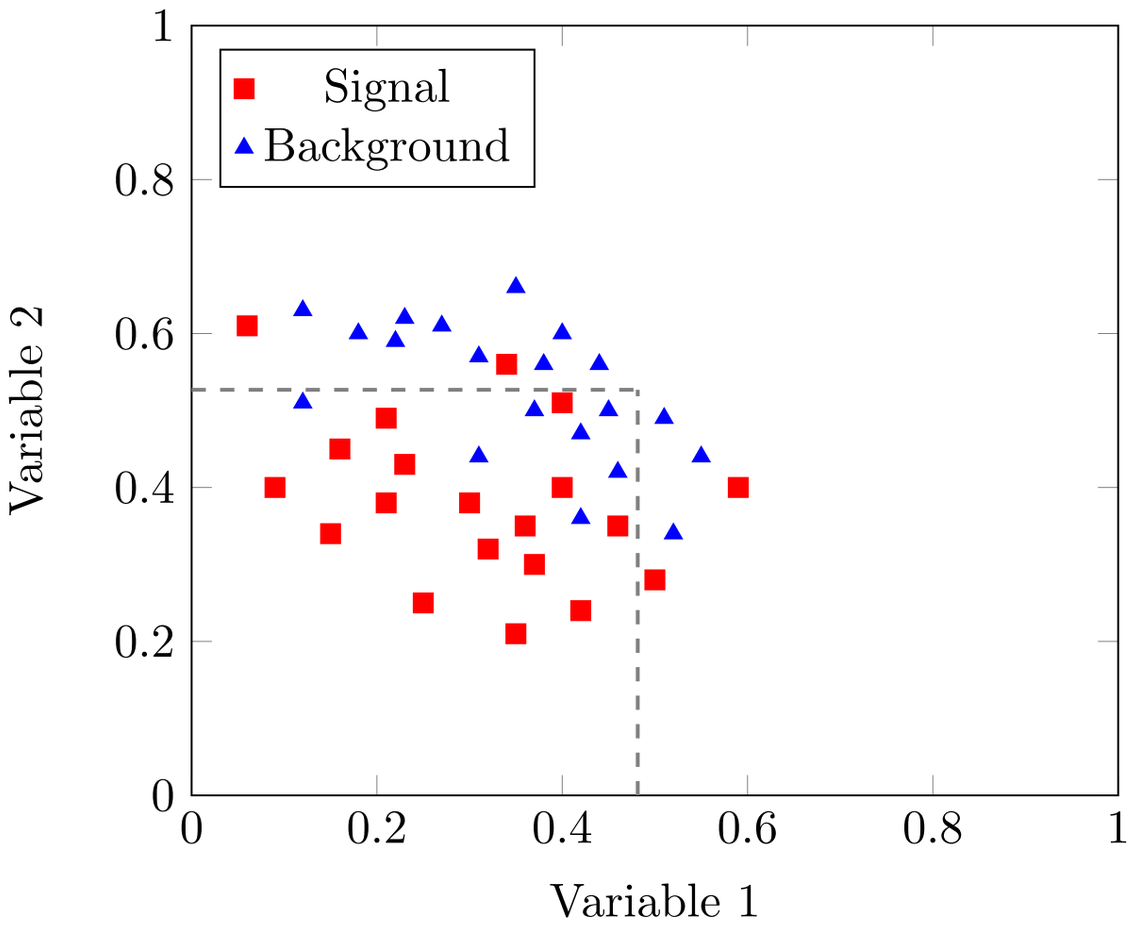
Growing a tree involves deciding which features to chose, and what conditions to use for branching, along with knowing when to stop.

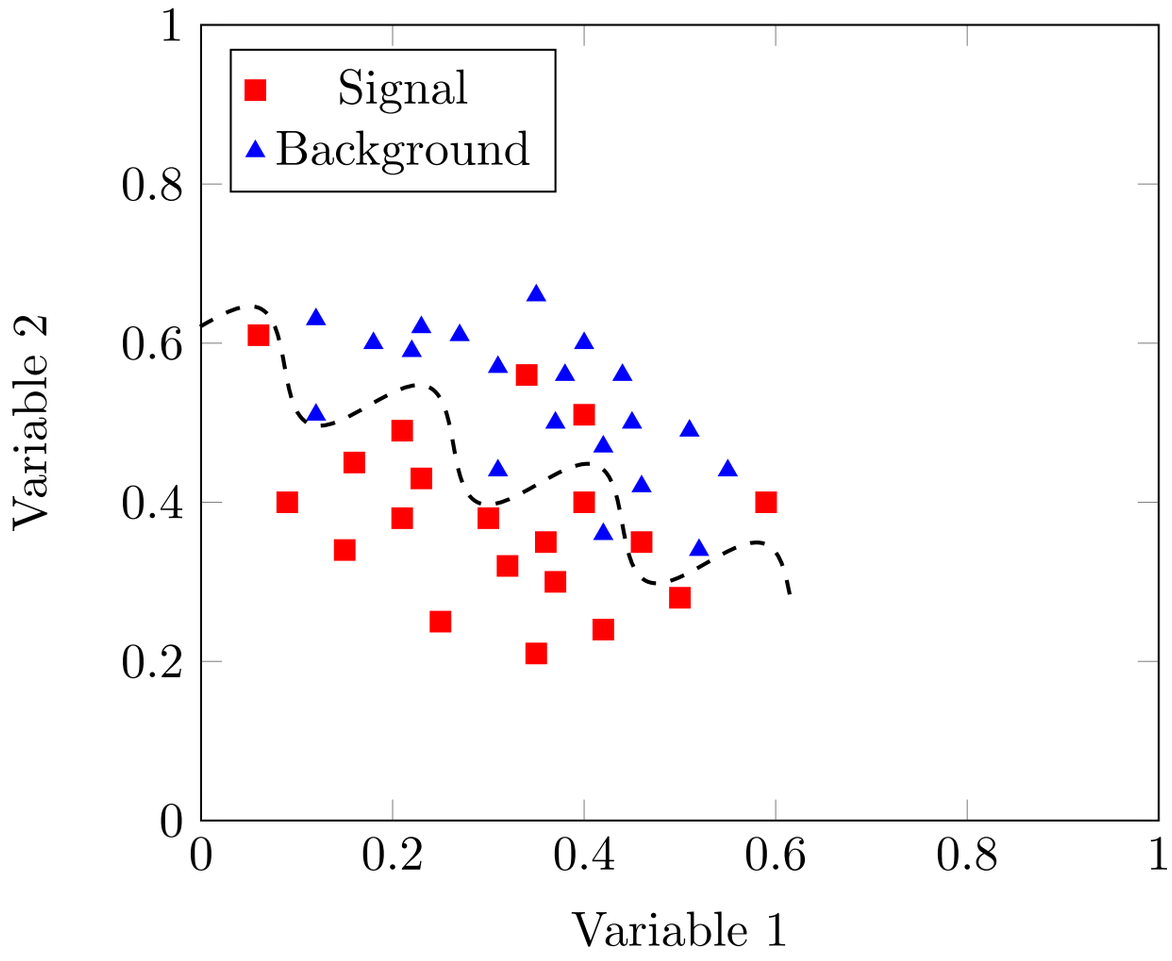
- Every node corresponds to a cut
- Train by trying different orders, to see if classification performance is improved

Training Example

- Cost function at each node, G=p(1-p), p is the fraction of correctly categorised inputs in that node
- Sum of cost function for each node characterises the performance of that tree
- Pick the tree with lowest cost function!
- Higher up nodes are more important

Underfitting and Overfitting





Deep Thinking
vs
Deep Learning

Neural Networks

- *Forward propagation*: pass x values via the activation function, compare with actual y values

- Value of w which minimises the loss function will give the desired functional form

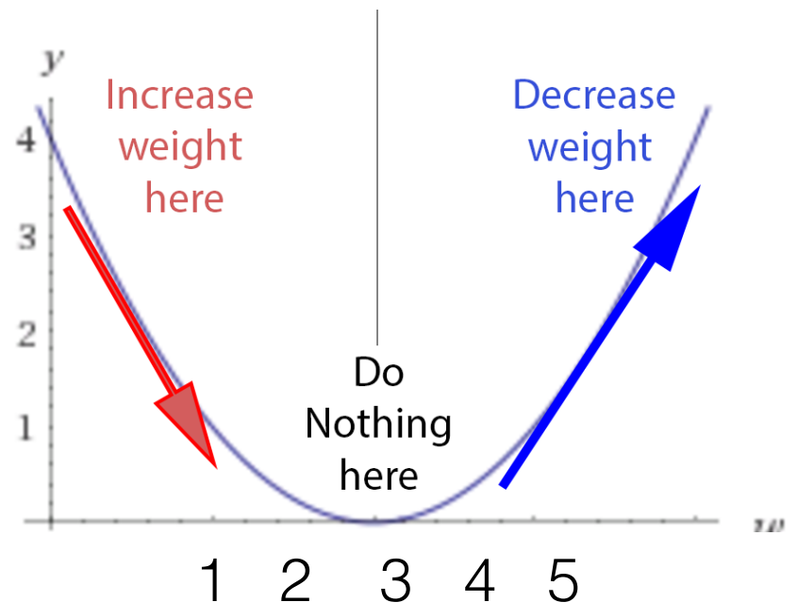
- Use derivative!

- Learning involves rerunning the activation function with different weights (*backpropagation*)

Loss function

w=2

Input	Output	Target	Squared difference
0	0	0	0
1	2	3	1
2	4	6	4
3	6	9	9
4	8	12	16
Sum			30



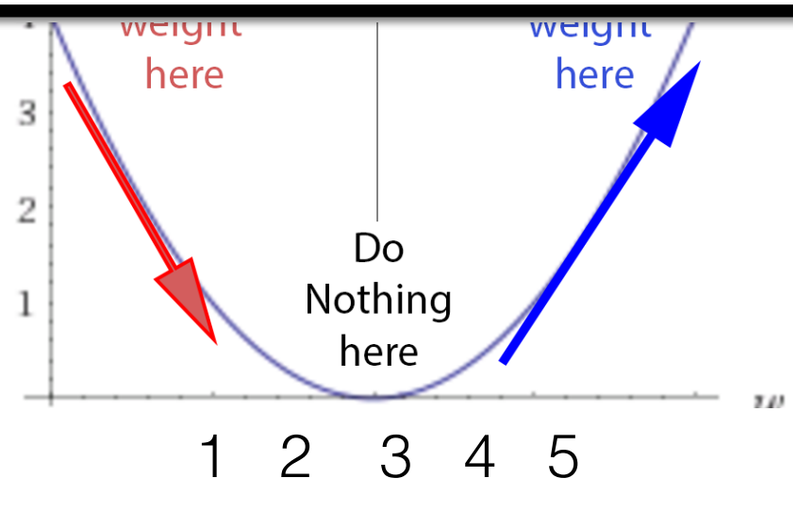
Neural Networks

- *Forward propagation*: pass x values via the activation function.

Loss function

As the derivative of a function at a certain point gives the rate at which this function is changing its values at that point, a derivative of the loss function indicates what modification of the weight will help. For example, a change of $\delta w = 0.01$ will result in loss function of $(0.99)^2 + (1.98)^2 + (2.97)^2 + (3.96)^2 = 29.403$, corresponding to a rate of $\delta y / \delta w = -0.597/0.01 \approx 60$ (where $30 - 29.403 = 0.597$). Since the derivative is negative, it means the error decreases if we increase the weights, which is of course true in this case. If we decreased the weight by the same value, then the loss function will be the 30.603, but the derivative will be same amount but positive, meaning the error increases if we increase the weights.

- Learning involves rerunning the activation function with different weights (*backpropagation*)



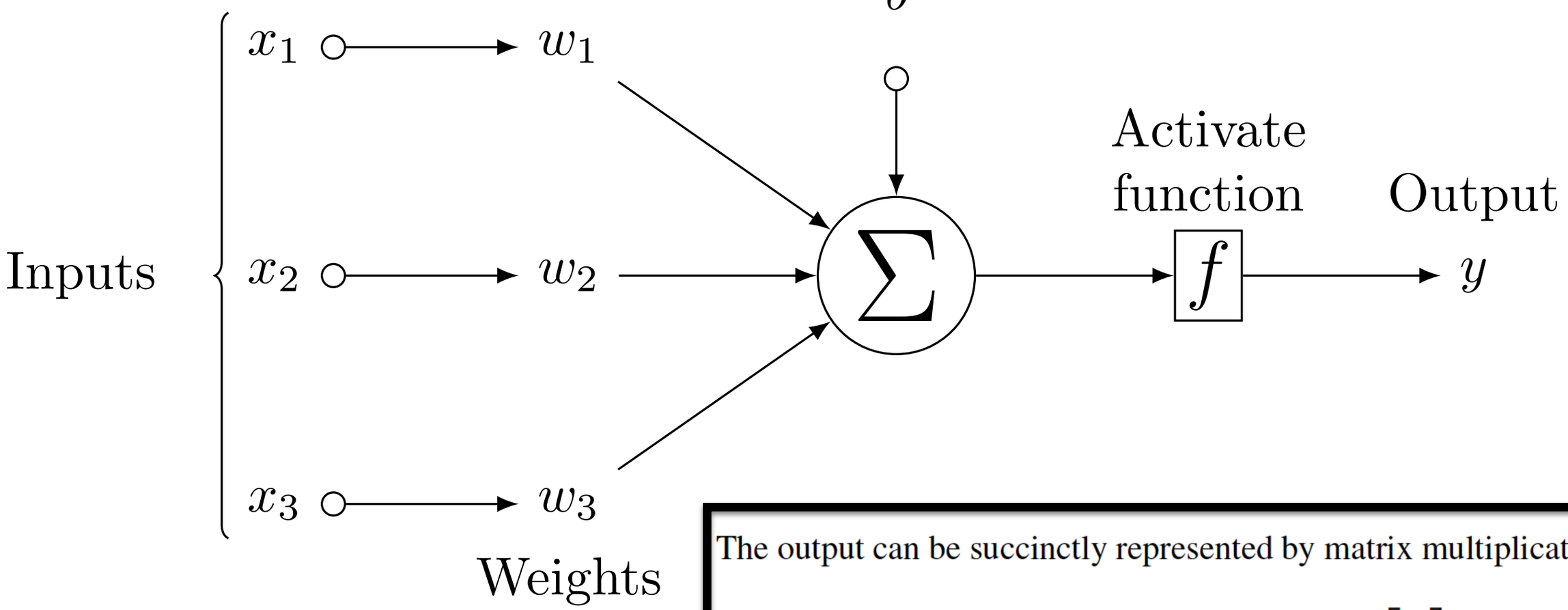
Neural Networks

While for this case, the rate of change of loss function clearly indicates what change (increase 0.5 times) is needed to reach a zero value of loss function corresponding to $w = 3$, such large change in the loss function is usually avoided to not encounter numerical instability (a derivative is only local at the point where we are calculating the derivative). The weights are updated by introducing a constant learning rate parameter, L: $w_{new} = w_{current} - L\frac{\delta J(w)}{\delta w}$, where $J(w)$ is the loss function. The learning rate determines how much we are adjusting the weights of our network with respect to the loss gradient, while approaching a minimum. If this rate is too large, the optimiser may keep overshooting a minimum and never converge. If it is too small, the network may take a very long time to converge or may select a local minima. Learning rate is an example of a *hyperparameter*, which are the parameters set in the algorithm (as opposed to the weights) independent of the training process.

Neural Networks

neurons

Bias :intercept in linear function



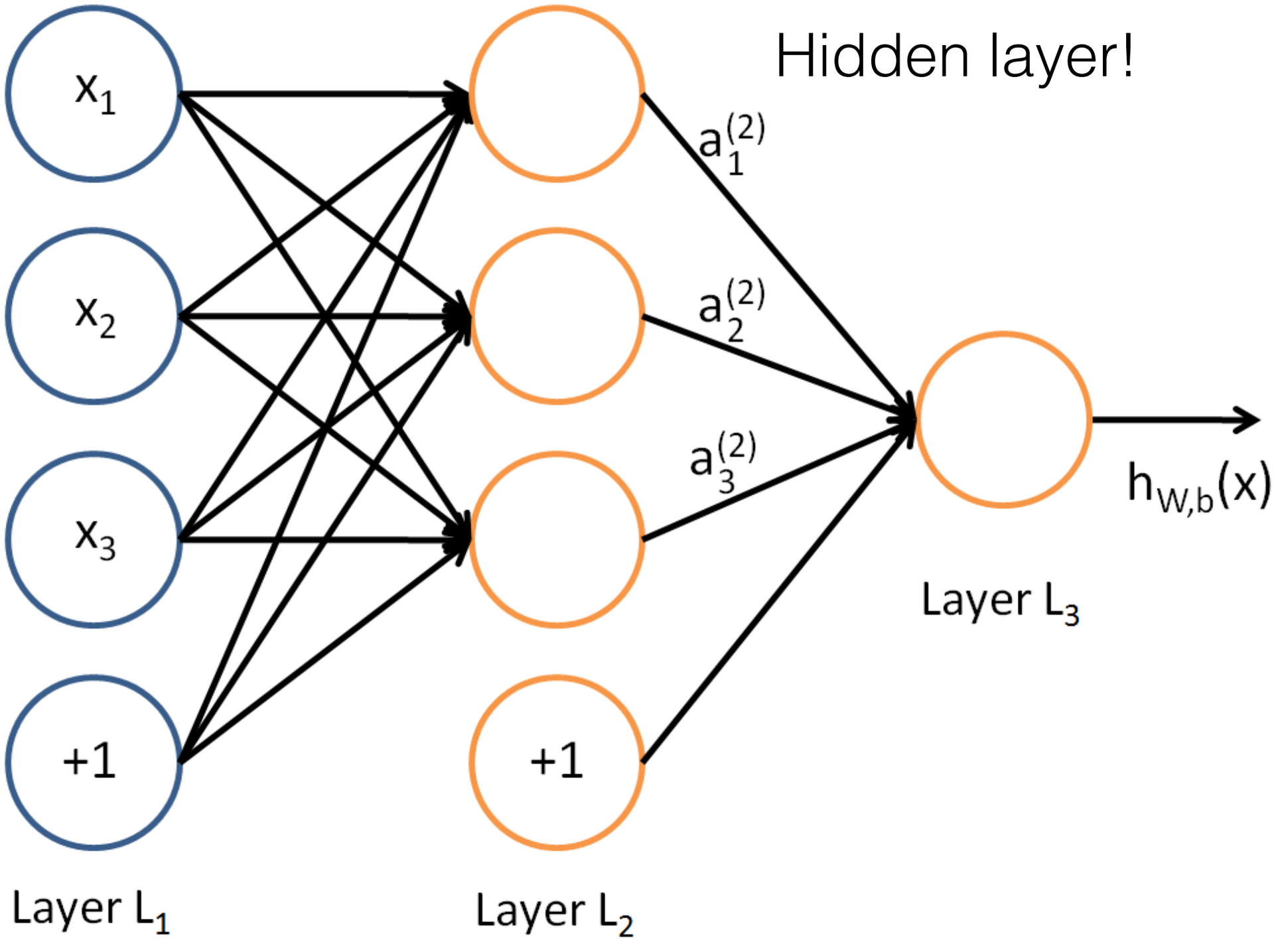
The output can be succinctly represented by matrix multiplication:

$$\begin{bmatrix} w_1 & w_2 & w_3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + b$$

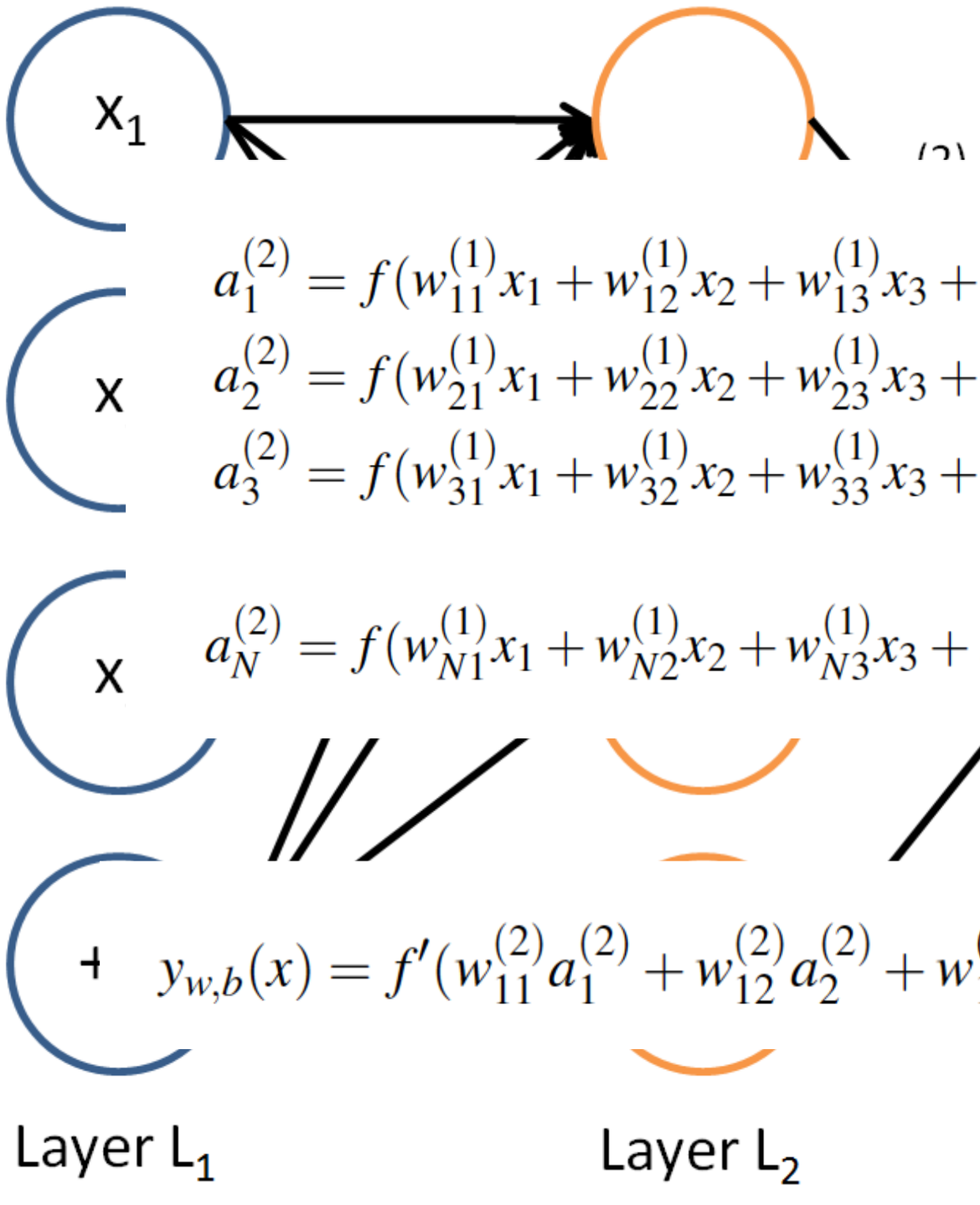
Neural Networks

A more complicated relationship between the input and output may not be represented by a simple linear mapping, rather by a series of functions, each represented by an additional layer in the neural network. These are termed hidden layers (because its values are not observed in the training set). One hidden layer is usually sufficient for the large majority of problems.

Neural Networks

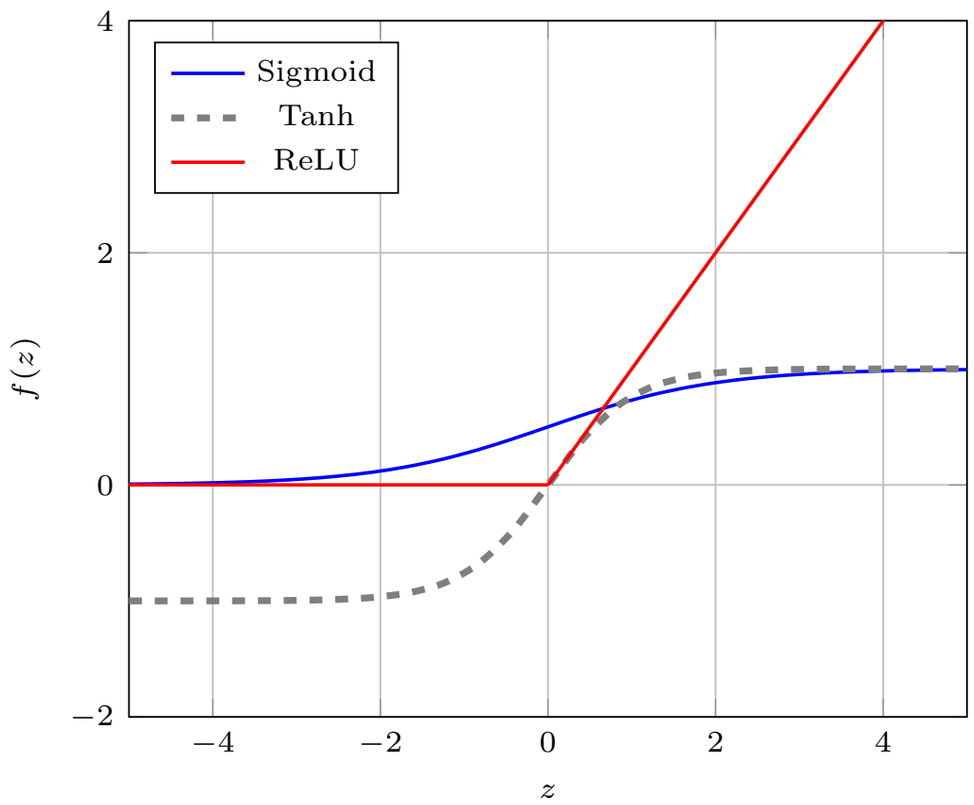


Neural Networks



Commonly used Activation Function

- Sigmoid, where $f(z) = \frac{1}{1-\exp(-z)}$, with output range between 0 to 1.
- Hyperbolic tangent (tanh), where $f(z) = \tanh(z) = \frac{\exp(z)-\exp(-z)}{\exp(z)+\exp(z)}$, with output range between -1 to 1.
- Rectified Linear Unit (ReLU), where $f(z) = max(0, z)$. While it is not bounded, it piece-wise linear and saturates at exactly 0 whenever the input $z < 0$.



Creating a NN therefore means coming up with values for the number of layers of each type and the number of nodes in each of these layers, along with the choice of activation functions.

Types of NN

- ANN: as shown before
- CNN: image processing
- GAN: combination of two, competing against one another

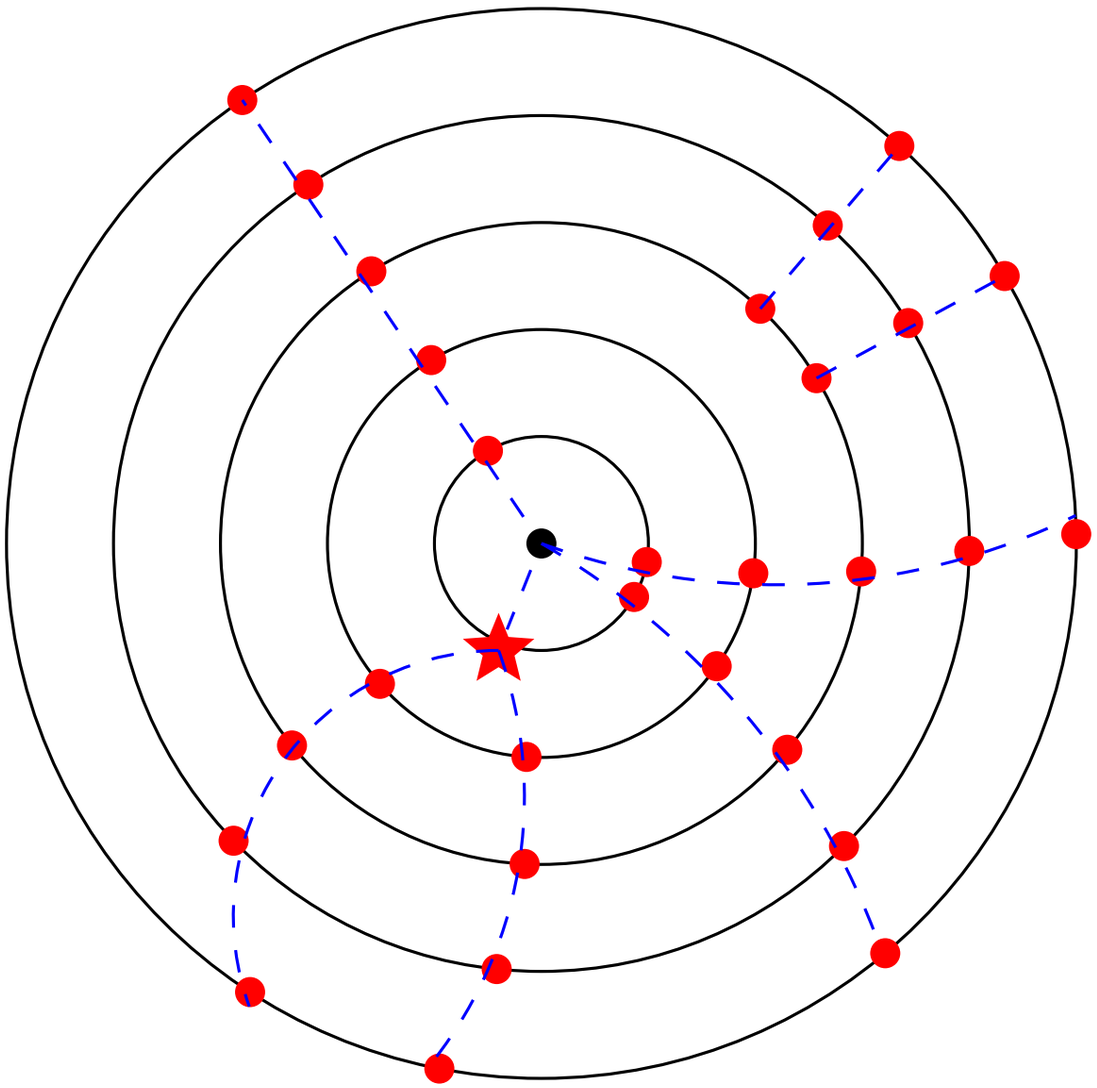
Types of Learning

- Supervised learning: labeled dataset
- Unsupervised learning: not labeled, extract patterns in data
- Semi/weakly supervised learning: middle ground, both labeled and unlabelled. GAN!
- Reinforcement learning: trial and error at each step, but successes are incentivised.

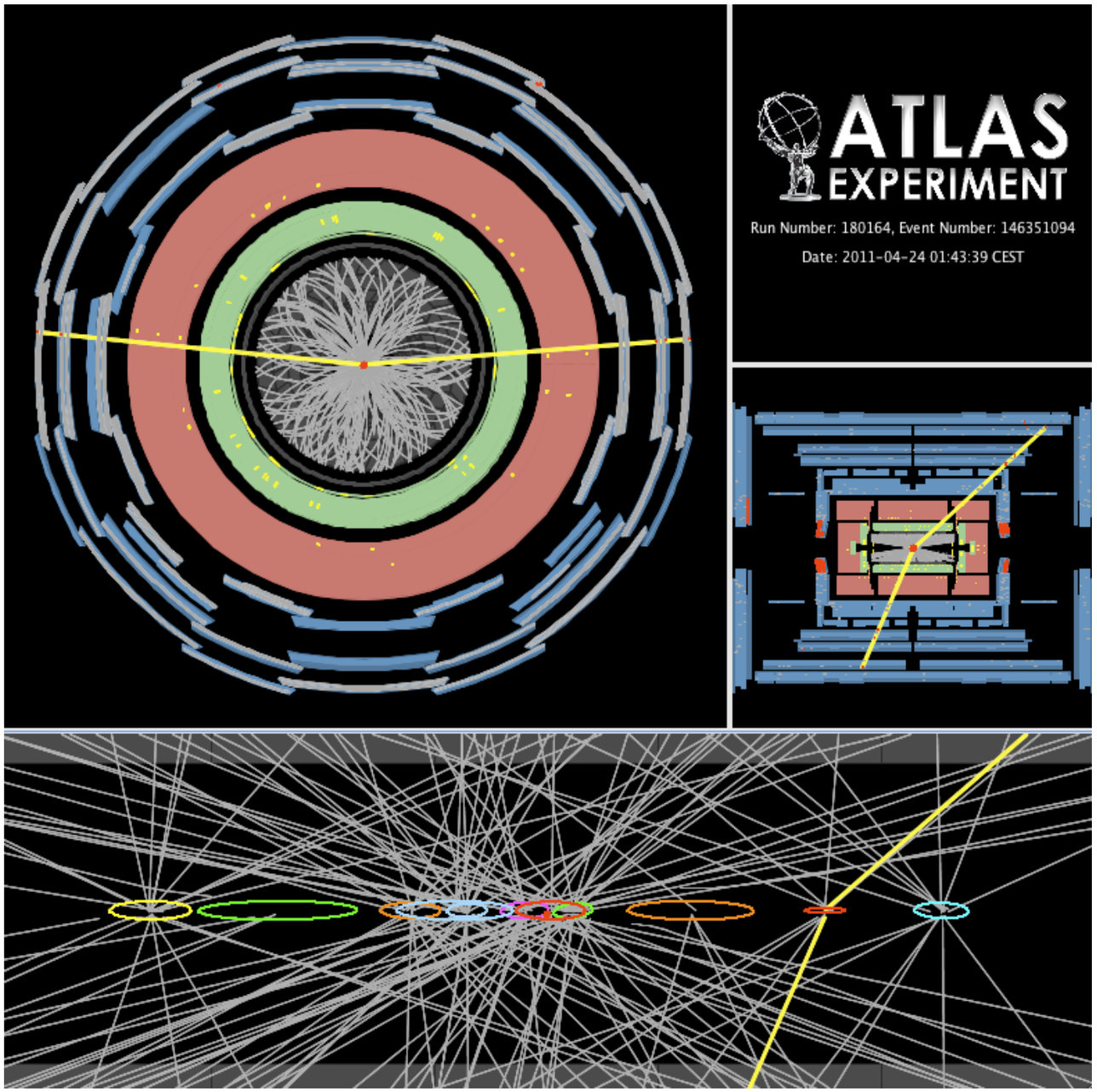
Let's look at some examples

Track Reconstruction

Pattern Recognition

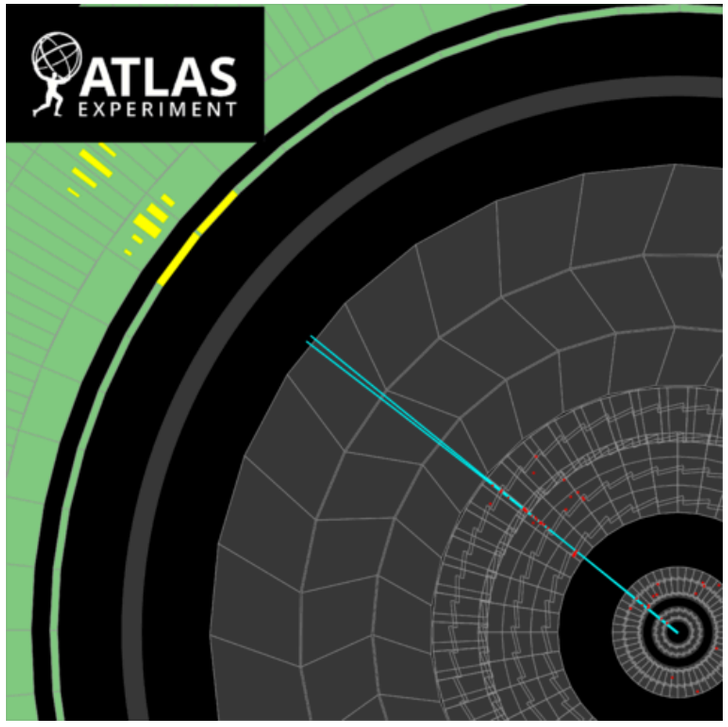


Photon conversion/Hadronic interactions



Binary classification

Electron Reconstruction

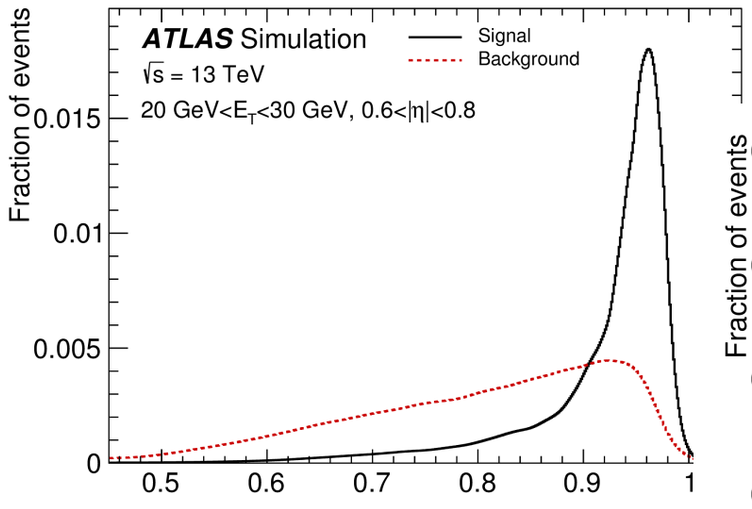


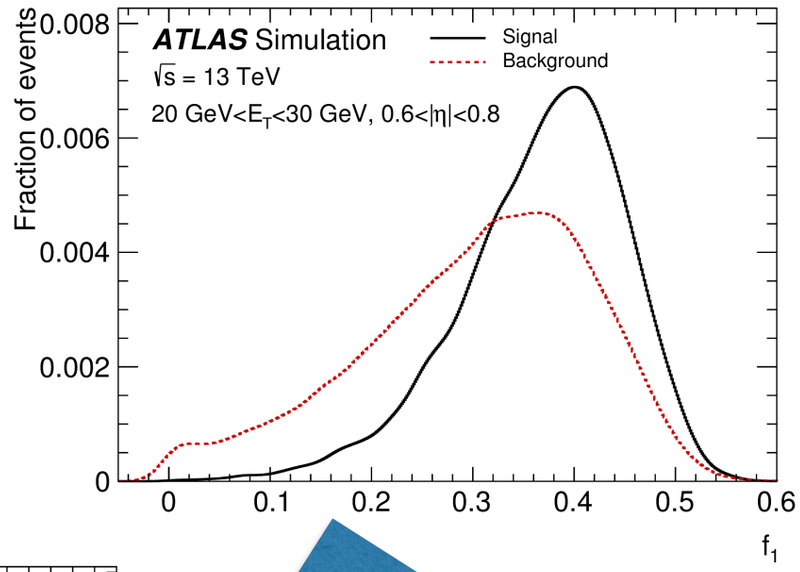
Pattern of energy deposits in different layers of calorimeters, isolation, and matching track and cluster

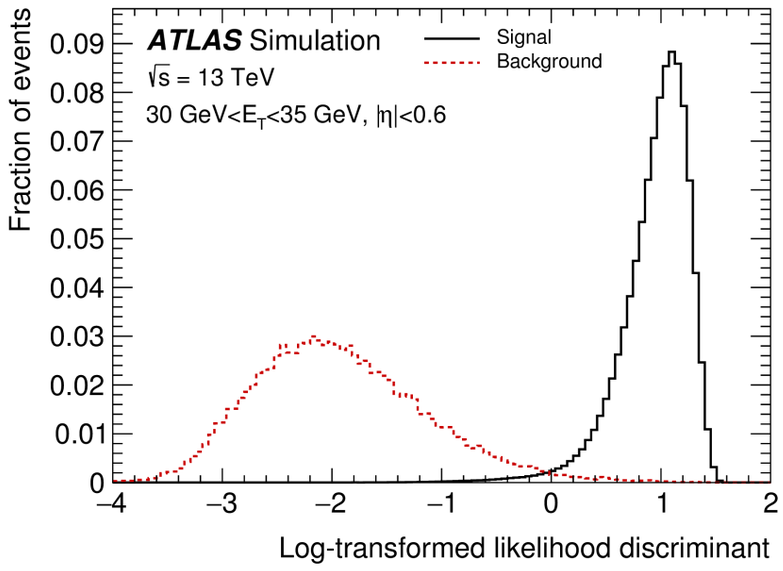
Type	Description	Name	Rejects			Usage
			LF	γ	HF	
Hadronic leakage	Ratio of E_{T} in the first layer of the hadronic calorimeter to E_{T} of the EM cluster (used over the range $\lvert\eta\rvert < 0.8$ or $\lvert\eta\rvert > 1.37$)	R_{had1}	x	x		LH
	Ratio of E_{T} in the hadronic calorimeter to E_{T} of the EM cluster (used over the range $0.8 < \lvert\eta\rvert < 1.37$)	R_{had}	x	x		LH
Third layer of EM calorimeter	Ratio of the energy in the third layer to the total energy in the EM calorimeter. This variable is only used for $E_{\mathrm{T}} < 80\ GeV$, due to inefficiencies at high E_{T}, and is also removed from the LH for $\lvert\eta\rvert > 2.37$, where it is poorly modelled by the simulation.	f_3	x			LH
Second layer of EM calorimeter	Lateral shower width, $\sqrt{(\Sigma E_i \eta_i^2)/(\Sigma E_i) - ((\Sigma E_i \eta_i)/(\Sigma E_i))^2}$, where E_i is the energy and η_i is the pseudorapidity of cell i and the sum is calculated within a window of 3×5 cells	$w_{\eta 2}$	x	x		LH
	Ratio of the energy in 3×3 cells over the energy in 3×7 cells centred at the electron cluster position	R_ϕ	x	x		LH
	Ratio of the energy in 3×7 cells over the energy in 7×7 cells centred at the electron cluster position	R_η	x	x	x	LH
First layer of EM calorimeter	Shower width, $\sqrt{(\Sigma E_i (i - i_{\mathrm{max}})^2)/(\Sigma E_i)}$, where i runs over all strips in a window of $\Delta\eta \times \Delta\phi \approx 0.0625 \times 0.2$, corresponding typically to 20 strips in η, and i_{max} is the index of the highest-energy strip, used for $E_{\mathrm{T}} > 150$ GeV only	w_{stot}	x	x	x	C
	Ratio of the energy difference between the maximum energy deposit and the energy deposit in a secondary maximum in the cluster to the sum of these energies	E_{ratio}	x	x		LH
	Ratio of the energy in the first layer to the total energy in the EM calorimeter	f_1	x			LH
Track conditions	Number of hits in the innermost pixel layer	n_{Blayer}		x		C
	Number of hits in the pixel detector	n_{Pixel}		x		C
	Total number of hits in the pixel and SCT detectors	n_{Si}		x		C
	Transverse impact parameter relative to the beam-line	d_0		x	x	LH
	Significance of transverse impact parameter defined as the ratio of d_0 to its uncertainty	$\lvert d_0/\sigma(d_0)\rvert$		x	x	LH
	Momentum lost by the track between the perigee and the last measurement point divided by the momentum at perigee	$\Delta p/p$	x			LH
TRT	Likelihood probability based on transition radiation in the TRT	eProbabilityHT	x			LH
Track–cluster matching	$\Delta\eta$ between the cluster position in the first layer and the extrapolated track	$\Delta\eta_1$	x	x		LH
	$\Delta\phi$ between the cluster position in the second layer of the EM calorimeter and the momentum-rescaled track, extrapolated from the perigee, times the charge q	$\Delta\phi_{\mathrm{res}}$	x	x		LH
	Ratio of the cluster energy to the track momentum, used for $E_{\mathrm{T}} > 150$ GeV only	E/p	x	x		C

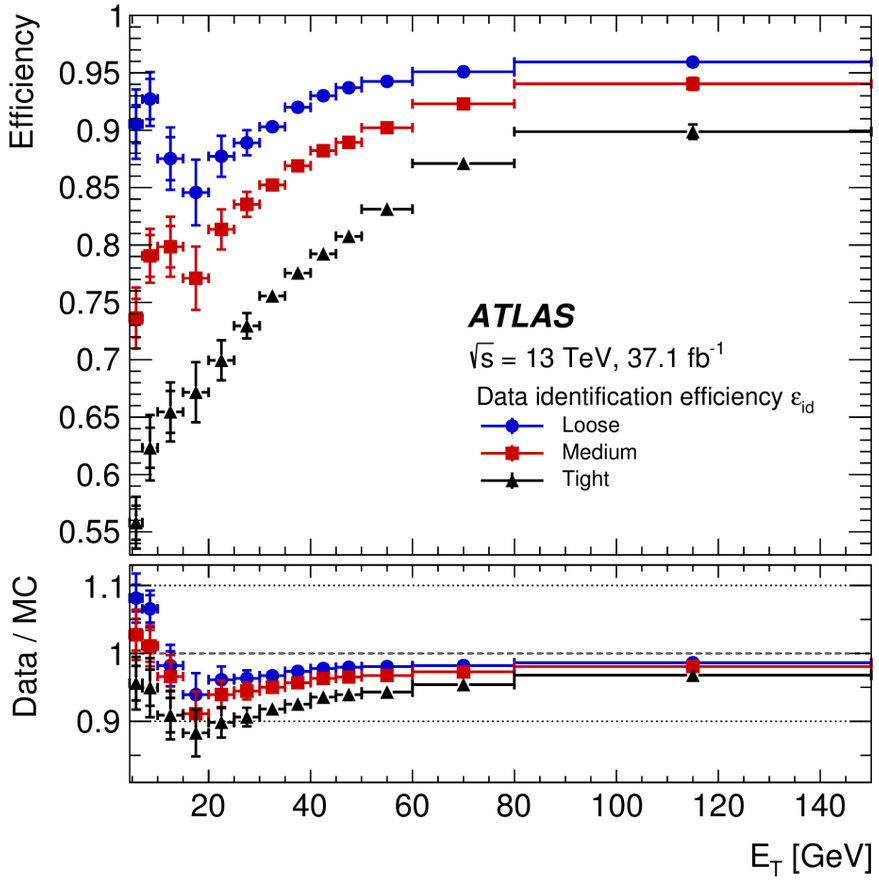
bea

Combine the observables!







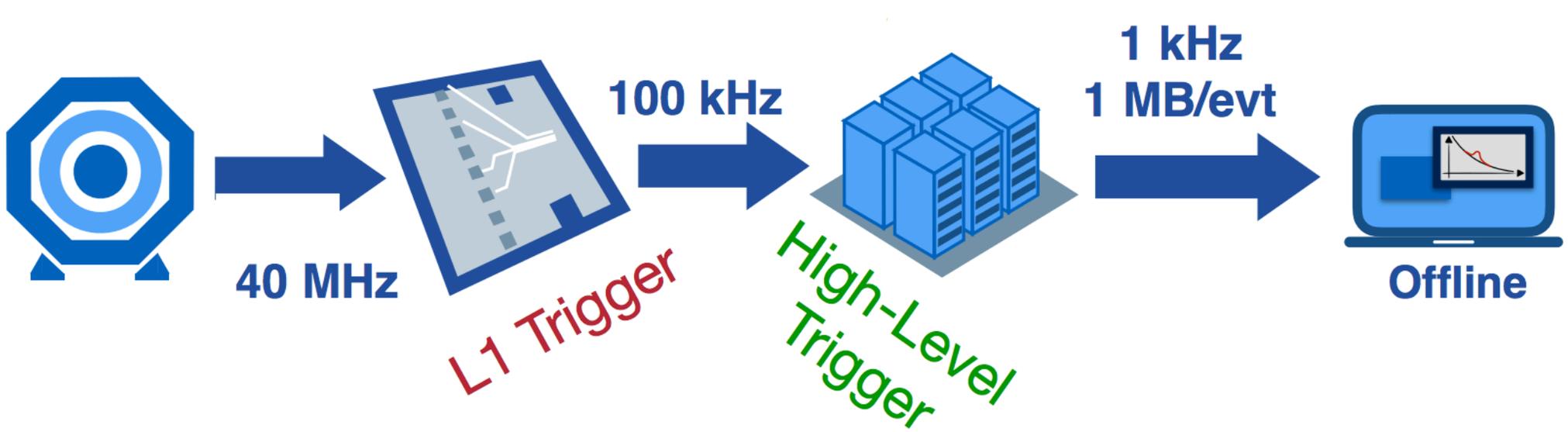


Slide from Jennifer Ngadiuba

Triggering

A typical trigger system

Triggering typically performed in multiple stages @ ATLAS and CMS



Absorbs 100s TB/s

Trigger decision to be made in O(μs)

Latencies require all-FPGA design

Computing farm for detailed analysis of the full event

Latency O(100 ms)

Slide from Jennifer Ngadiuba

What are FPGAs?

Latencies at L1 trigger require all-FPGA design

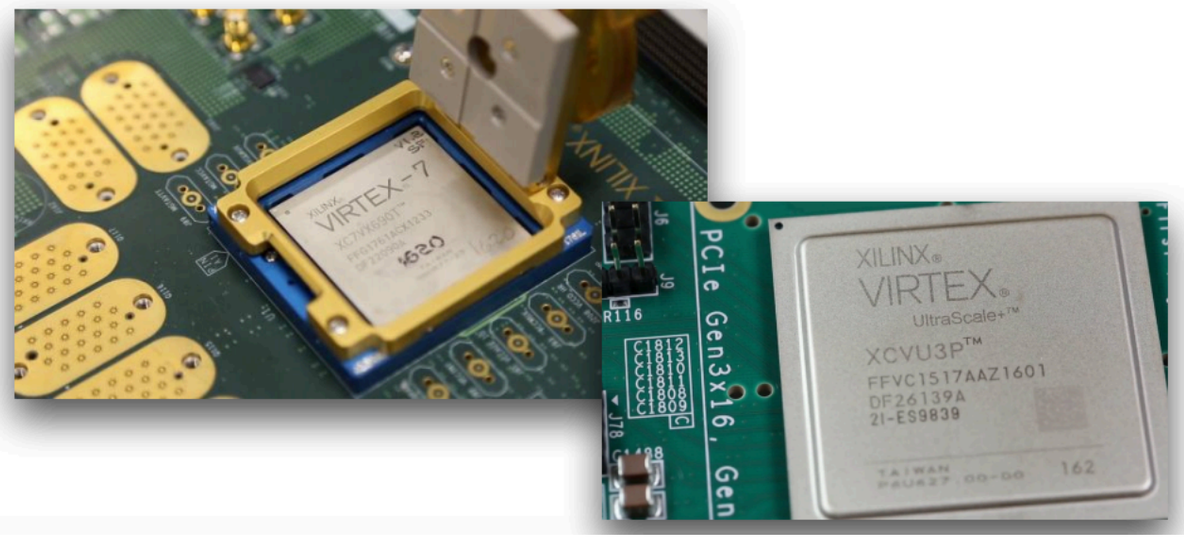
Field Programmable Gate Arrays are reprogrammable integrated circuits

Contain array of **logic cells** embedded with **DSPs**, **BRAMs**, etc.

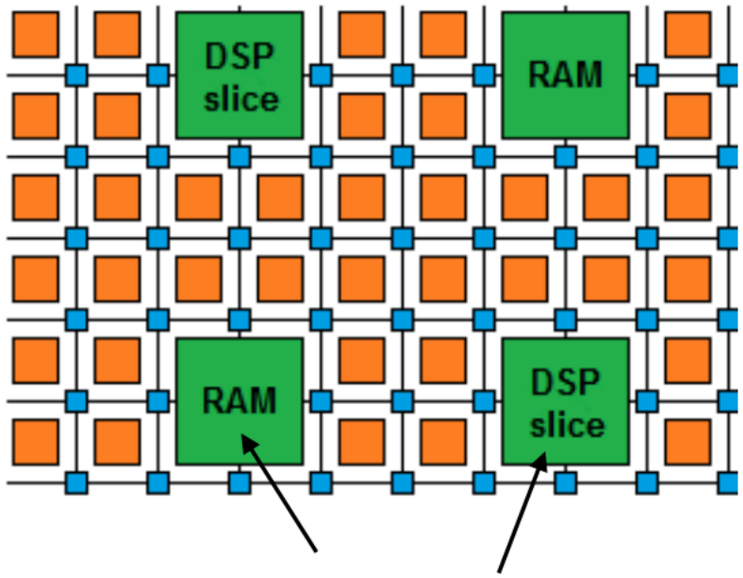
High speed input/output to handle the large bandwith

Support **highly parallel** algorithm implementations

Low power (relative to CPU/GPU)



FPGA diagram



Digital Signal Processors (DSPs): logic units used for multiplications

Random-access memories (RAMs): embedded memory elements

Flip-flops (FF) and look up tables (LUTs) for additions

Slide from Jennifer Ngadiuba

The rise of specialized hardware for ML

GPUs excell at parallel processing

Good for complex NN training of huge amount of data!

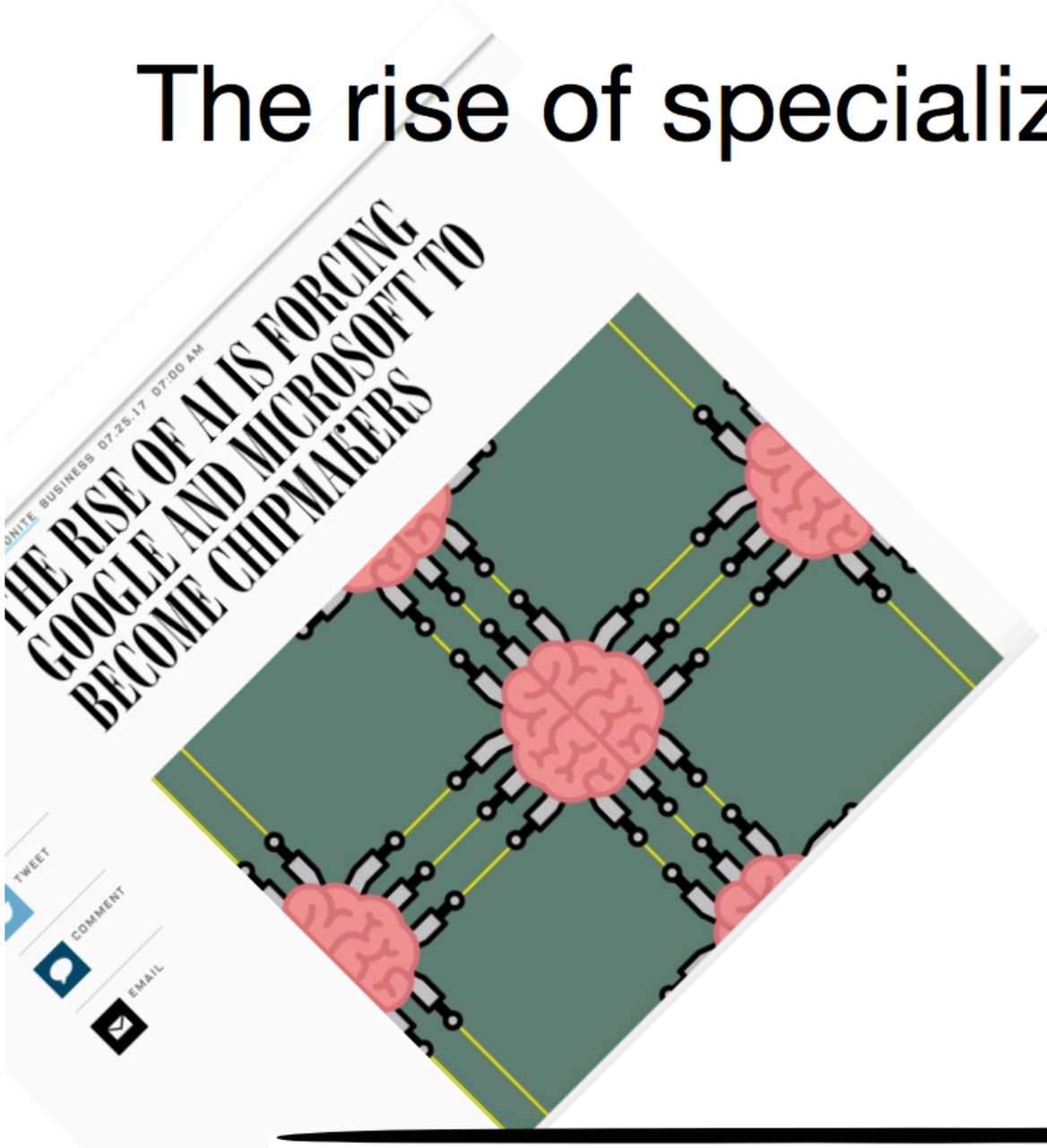
Notoriously power-hungry

Sub-optimal for fast and simple NN inference

Optimize resources utilization for less intensive tasks

Slide from Jennifer Ngadiuba

The rise of specialized hardware for ML



GPUs excell at parallel processing

Good for complex NN training of huge amount of data!

Notoriously power-hungry

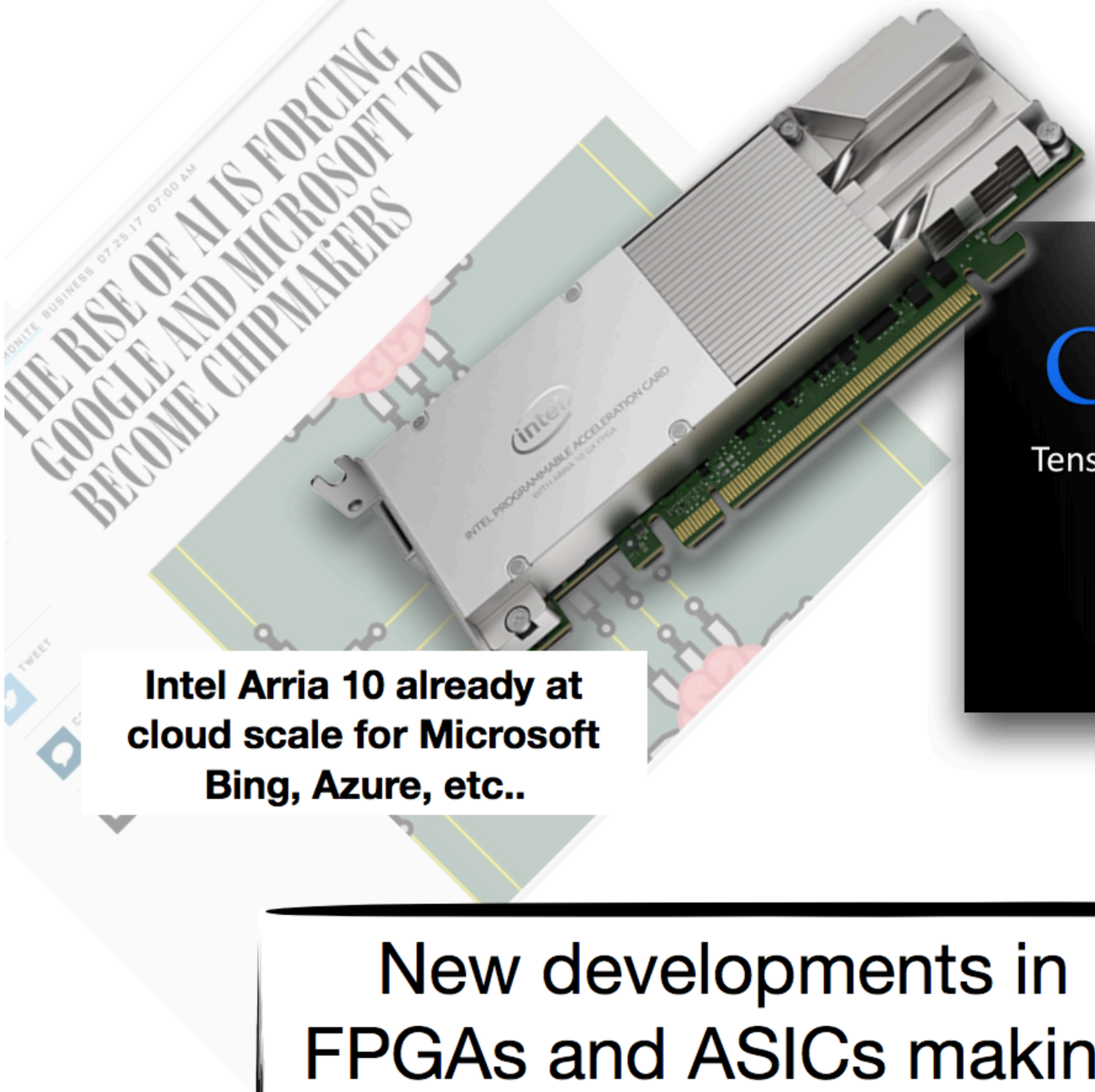
Sub-optimal for fast and simple NN inference

Optimize resources utilization for less intensive tasks

New developments in FPGAs and ASICs making #RealTimeAI possible!

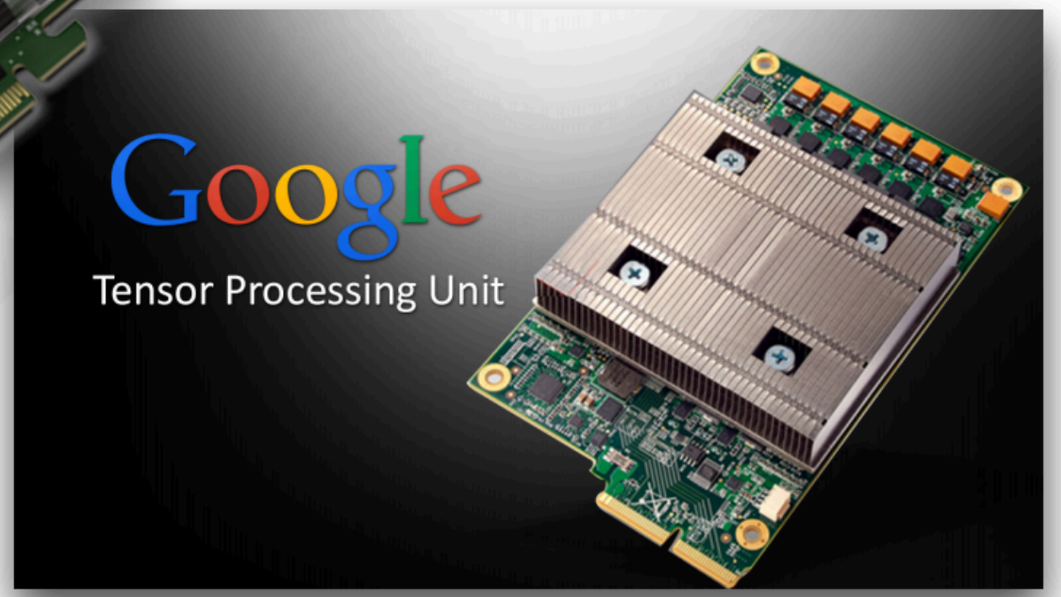
Slide from Jennifer Ngadiuba

The rise of specialized hardware for ML



Intel Arria 10 already at cloud scale for Microsoft Bing, Azure, etc..

Custom AI hardware for Google on the cloud



New developments in FPGAs and ASICs making #RealTimeAI possible!

Support **highly parallel algorithm** implementations with guaranteed latency

Low power (relative to CPU/GPU)

High speed I/O to handle the large bandwith

Slide from Jennifer Ngadiuba

Efficient NN design for FPGAs

FPGAs provide huge flexibility

Performance depends on how well you take advantage of this

Constraints:

Input bandwidth

FPGA resources

Latency

We have three handles:

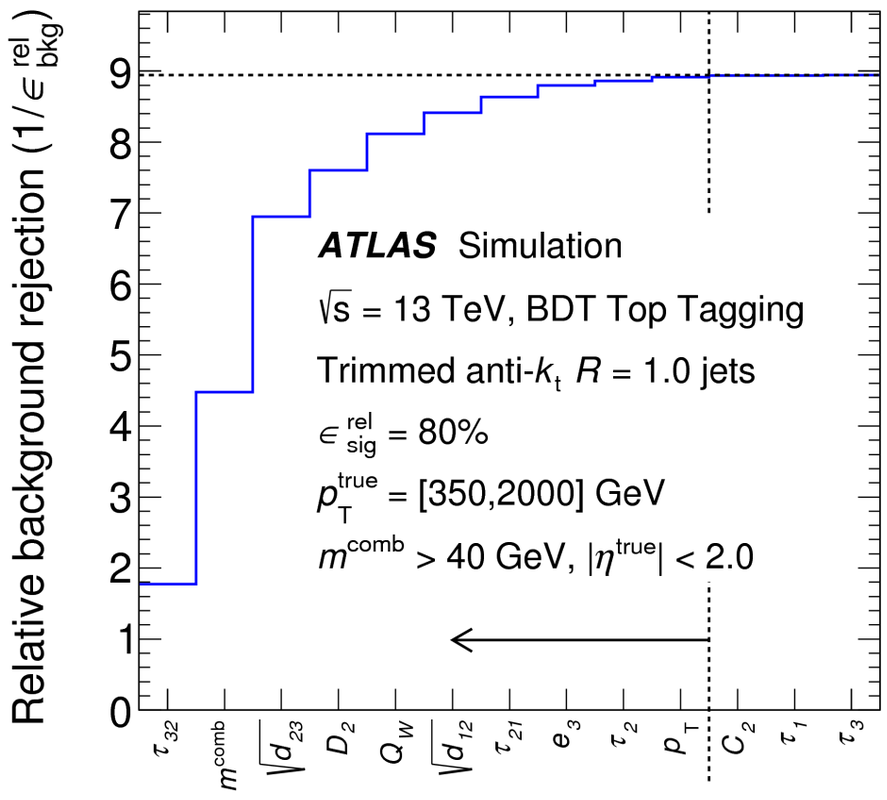
- **compression:** reduce number of synapses or neurons
- **quantization:** reduces the precision of the calculations (inputs, weights, biases)
- **parallelization**: tune how much to parallelize to make the inference faster/slower versus FPGA resources

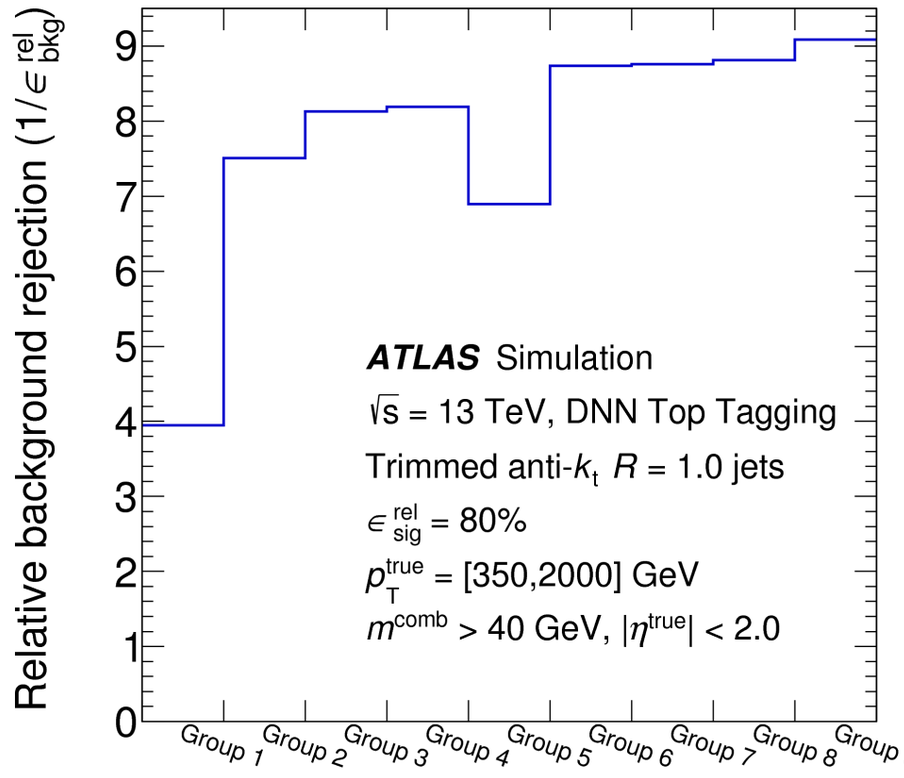
ML4Jet

- Classifying jets based on origin has been a testbed for ML applications!

- Mostly uses supervised learning using simulation

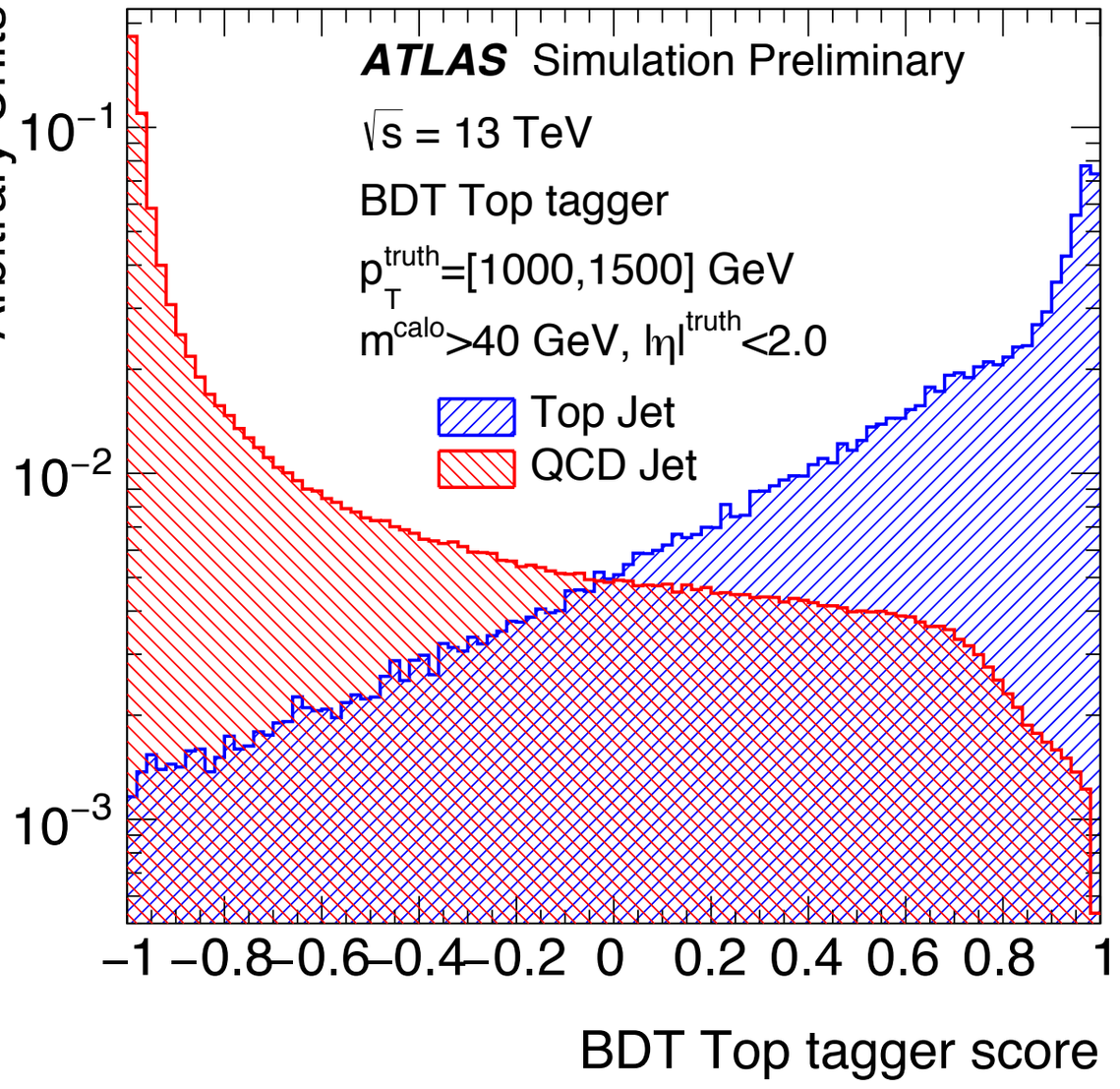
Jet Tagging

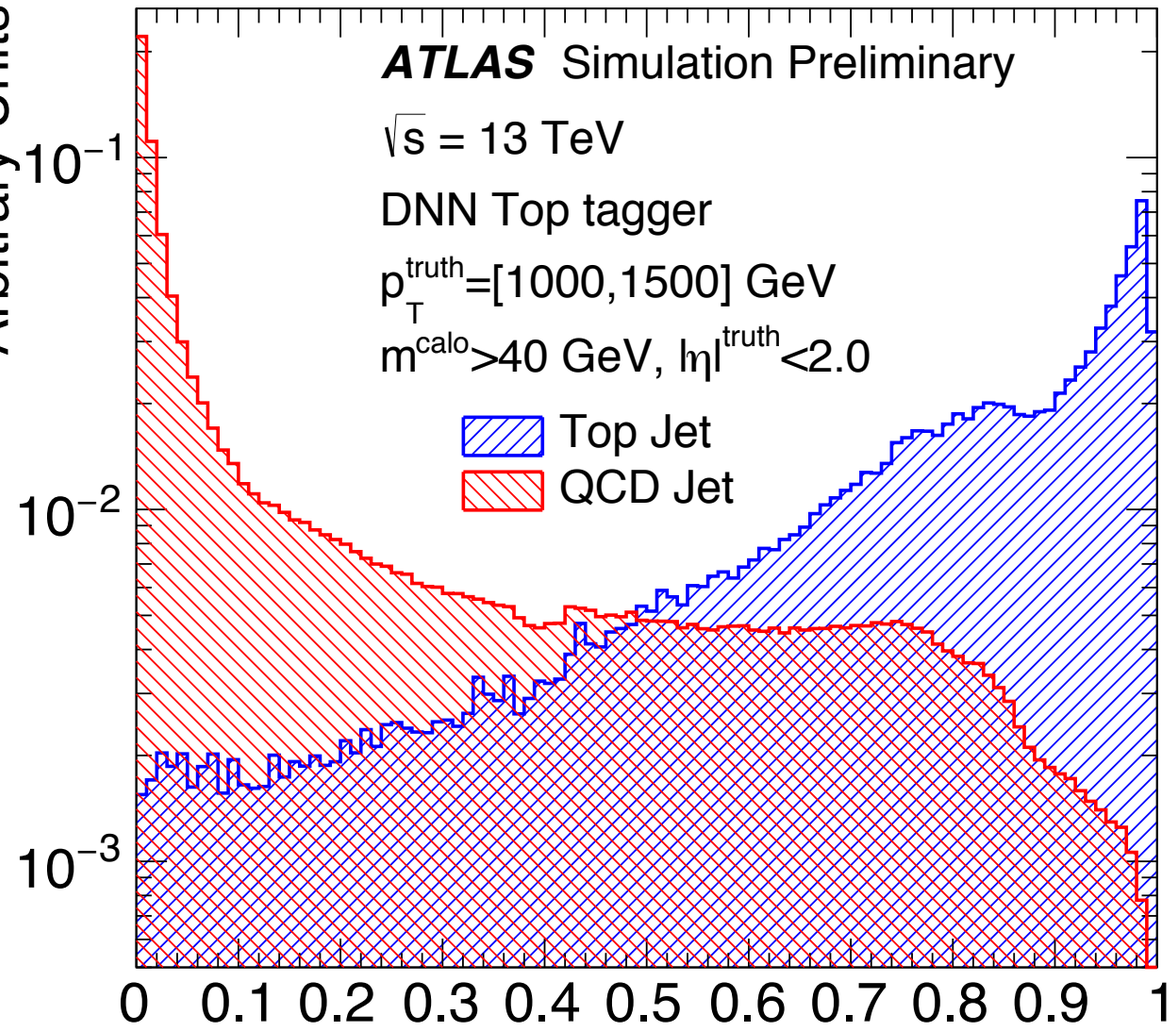




	W Boson Tagging											Top Quark Tagging										
	DNN Test Groups									Chosen Inputs		DNN Test Groups									Chosen Inputs	
Observable	1	2	3	4	5	6	7	8	9	BDT	DNN	1	2	3	4	5	6	7	8	9	BDT	DNN
m^{comb}	○	○		○	○	○	○	○	○	○	○		○	○	○		○	○	○	○	○	○
p_T	○	○			○	○		○	○	○	○			○	○			○	○	○	○	○
e_3	○	○				○			○						○			○		○	○	○
C_2			○	○	○		○	○	○		○	○	○	○		○	○		○	○		○
D_2			○	○	○		○	○	○	○	○	○	○	○		○	○		○	○		○
τ_1	○	○				○			○	○					○			○		○		○
τ_2	○	○				○			○						○			○		○	○	○
τ_3															○			○		○		○
τ_{21}			○	○	○		○	○	○	○	○	○	○	○		○	○		○	○	○	○
τ_{32}												○	○	○		○	○		○	○	○	○
R_2^{FW}			○	○	○	○	○	○	○	○	○											
$\mathcal{P}$			○	○	○	○	○	○	○	○	○											
a_3			○	○	○	○	○	○	○	○	○											
A			○	○	○	○	○	○	○	○	○											
z_{cut}			○	○	○		○	○	○		○											
$\sqrt{d_{12}}$		○				○	○	○	○	○	○					○	○	○	○	○	○	○
$\sqrt{d_{23}}$																○	○	○	○	○	○	○
$KtDR$		○				○	○	○	○	○	○											
Q_w																○	○	○	○	○	○	

Jet Tagging

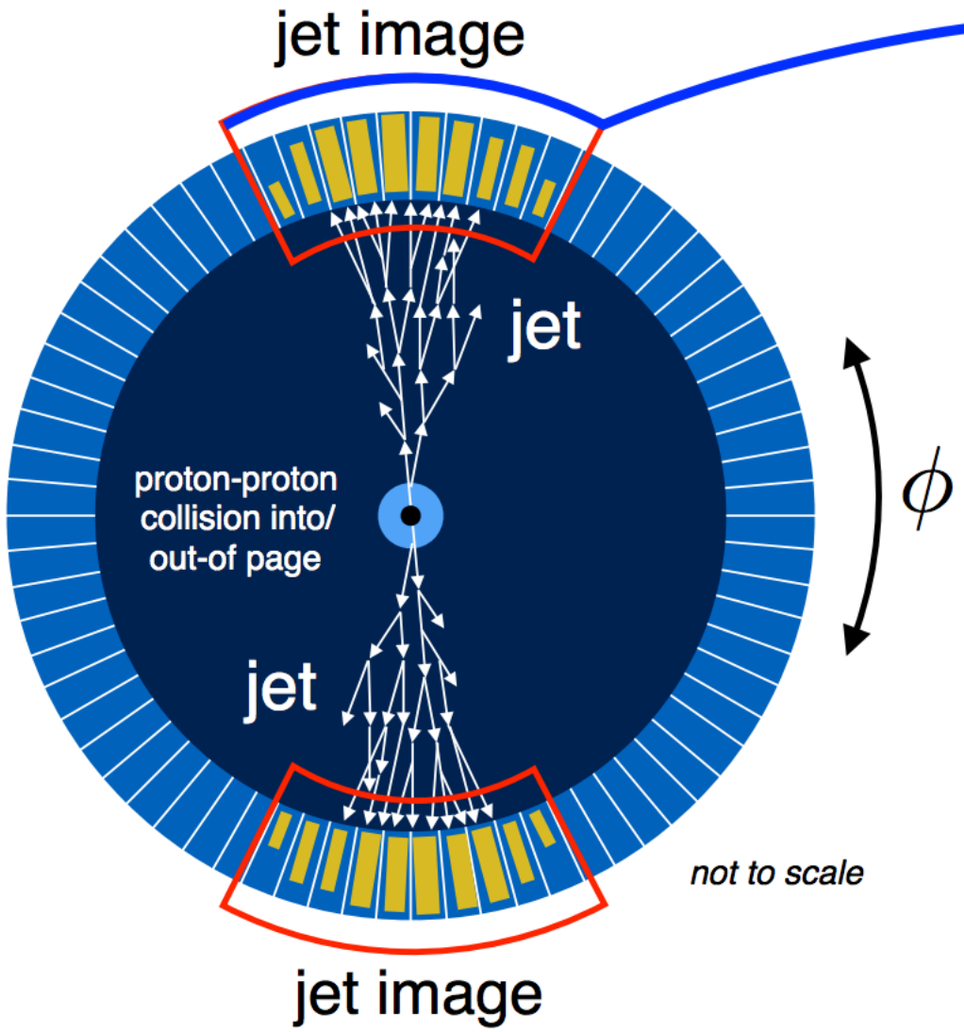

ATLAS Simulation Preliminary
√s = 13 TeV
BDT Top tagger
p_T^truth=[1000,1500] GeV
m^calo>40 GeV, |η|^truth<2.0
Top Jet
QCD Jet
Arbitrary Units
10^−1
10^−2
10^−3
−1 −0.8 −0.6 −0.4 −0.2 0 0.2 0.4 0.6 0.8 1
BDT Top tagger score

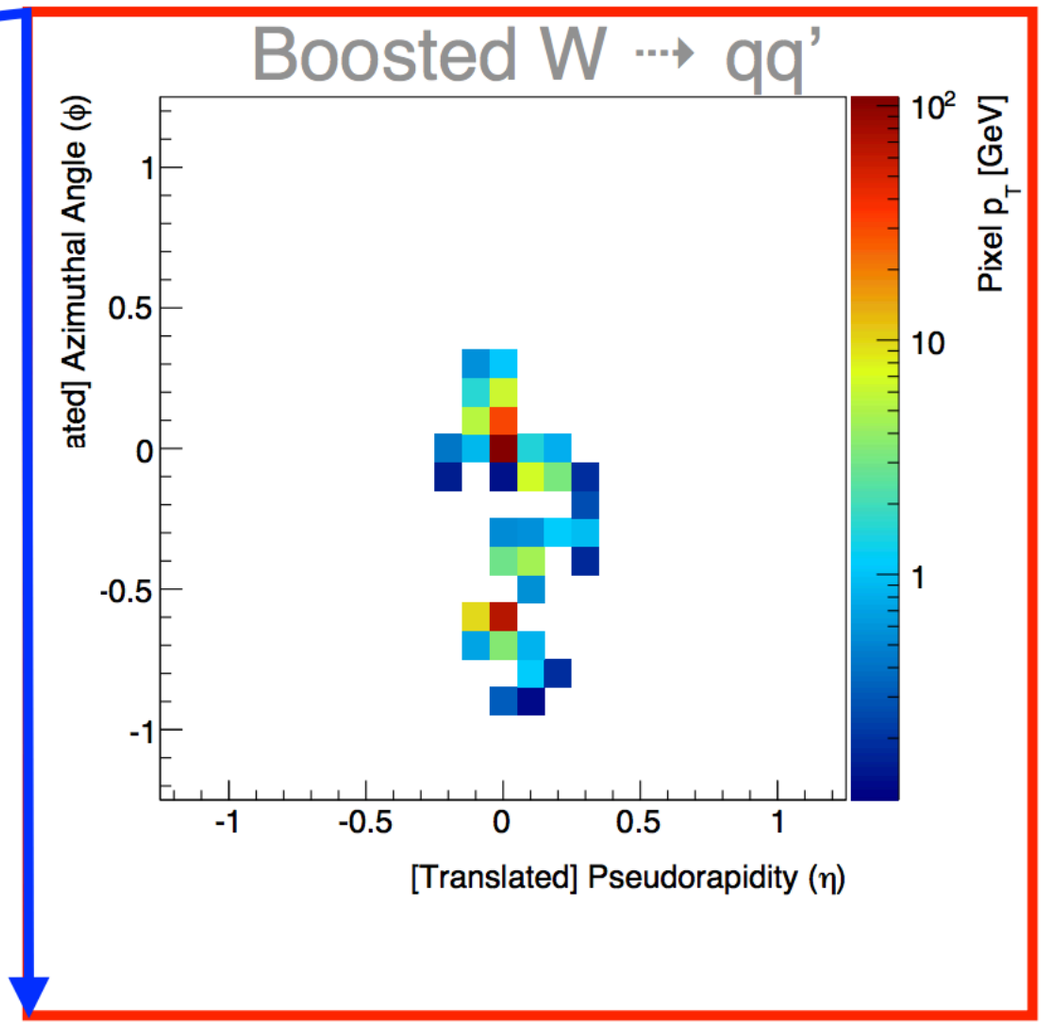

ATLAS Simulation Preliminary
√s = 13 TeV
DNN Top tagger
p_T^truth=[1000,1500] GeV
m^calo>40 GeV, |η|^truth<2.0
Top Jet
QCD Jet
Arbitrary Units
10^−1
10^−2
10^−3
0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1
DNN Top tagger score

Slide from Ben Nachman

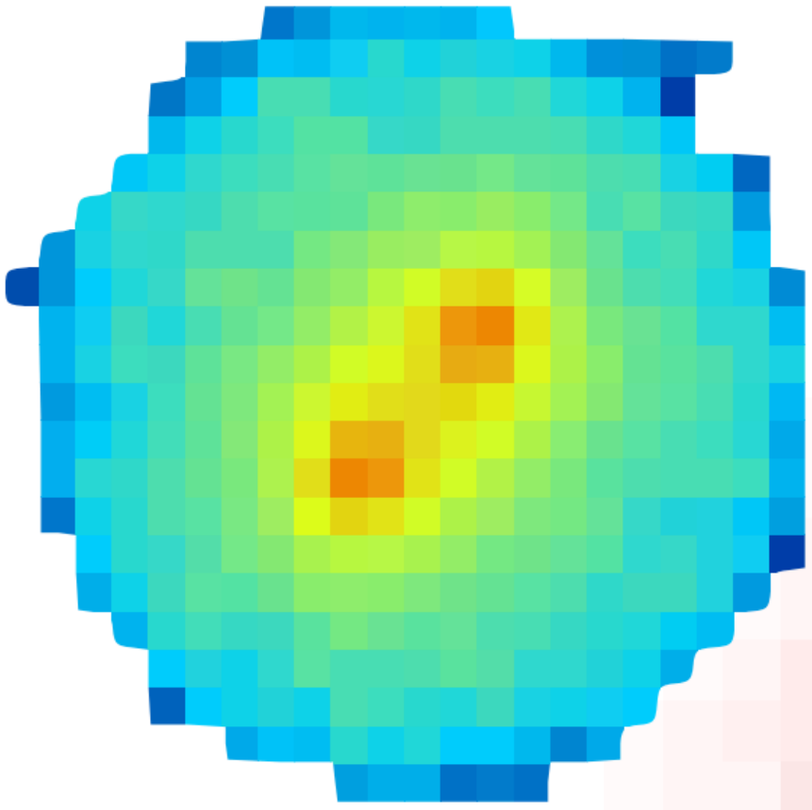
Jets as Images

Jet Image: *A two-dimensional fixed representation of the radiation pattern inside a jet*





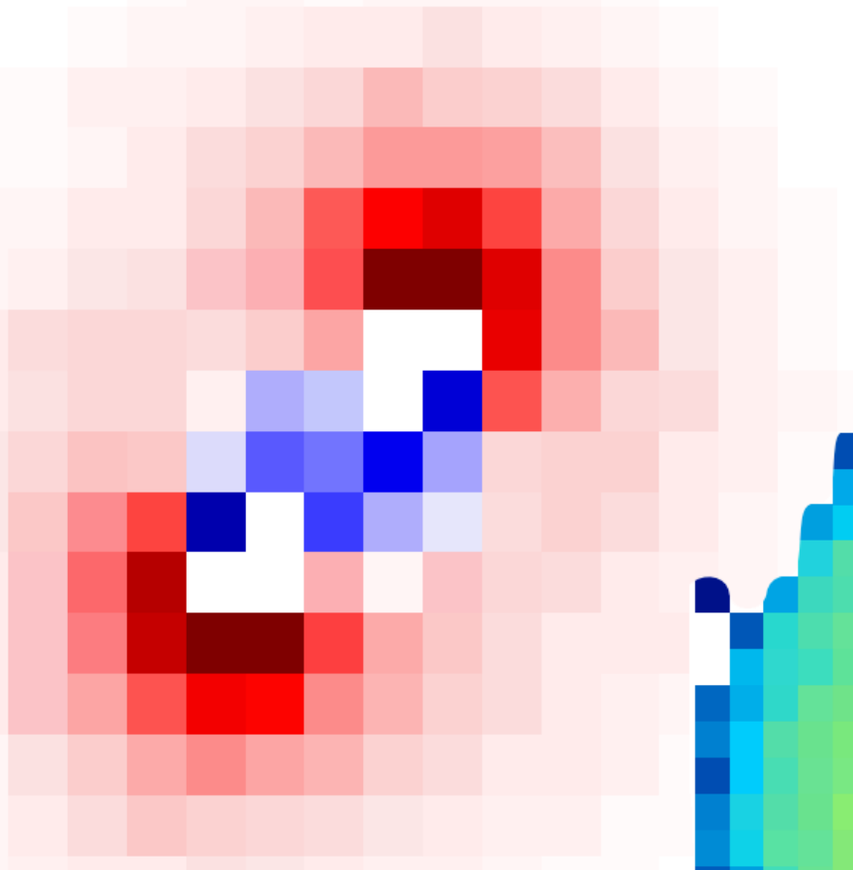
Why images?



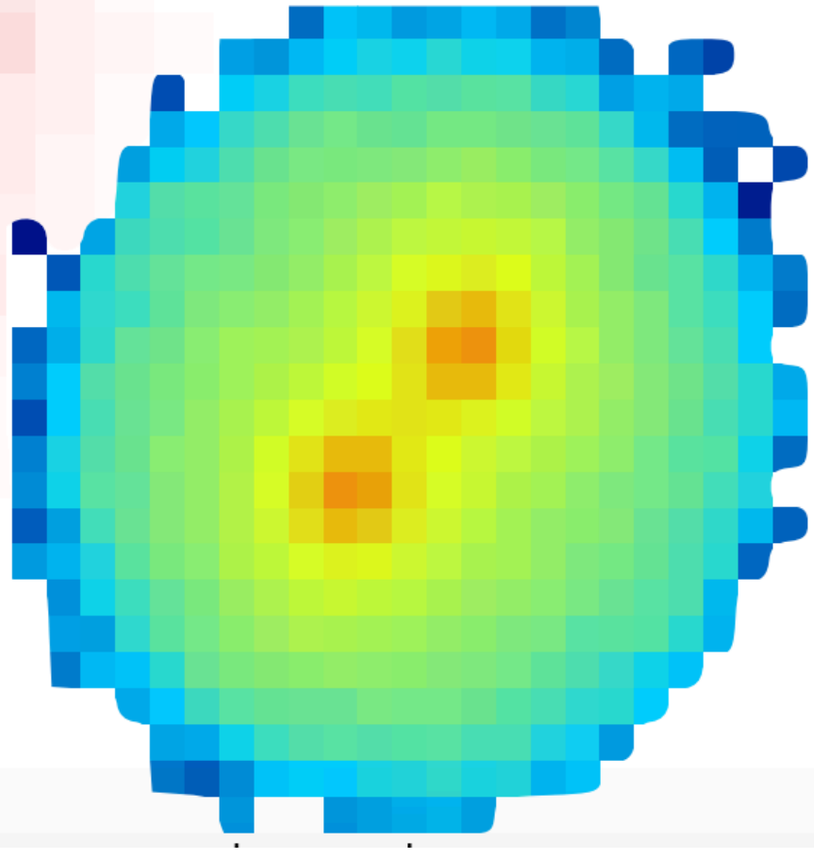
singlet ⇢ qq’

Can directly visualize physics

and we can benefit from the extensive image processing literature



octet ⇢ qq’



there is information encoded in the physical distance between pixels

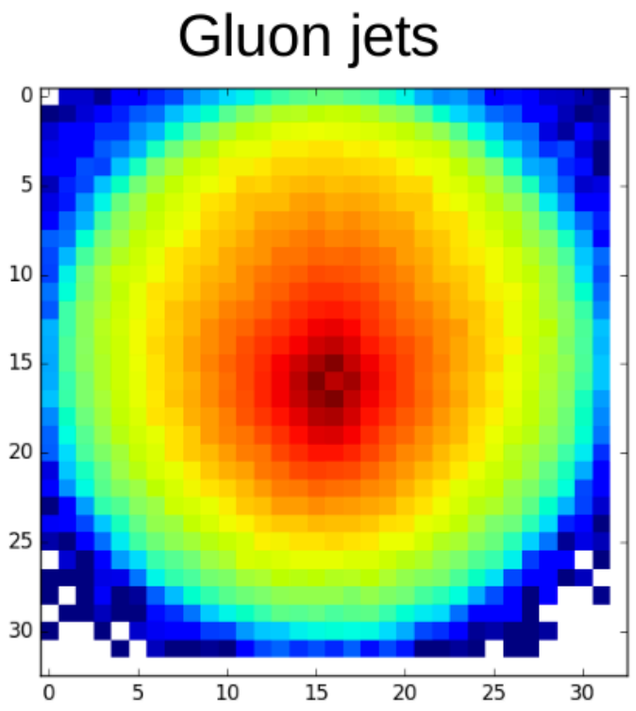
(will mention other fixed representations later)

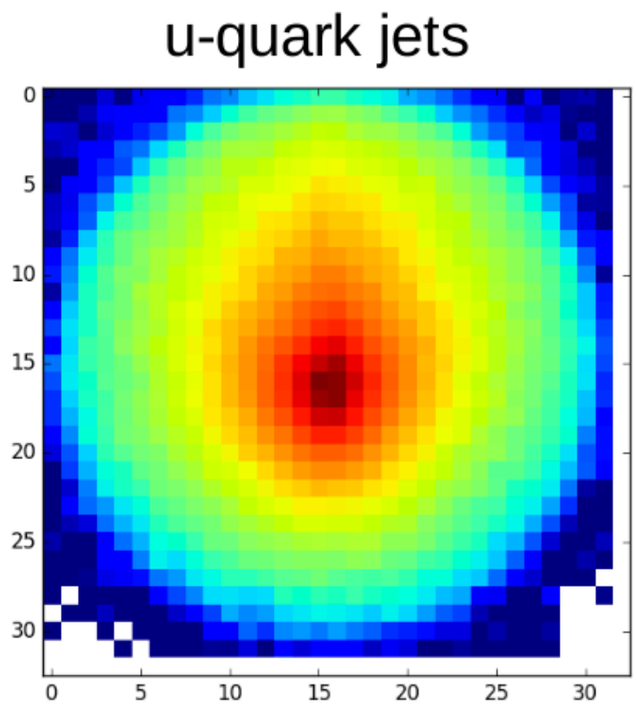
Jets as Images

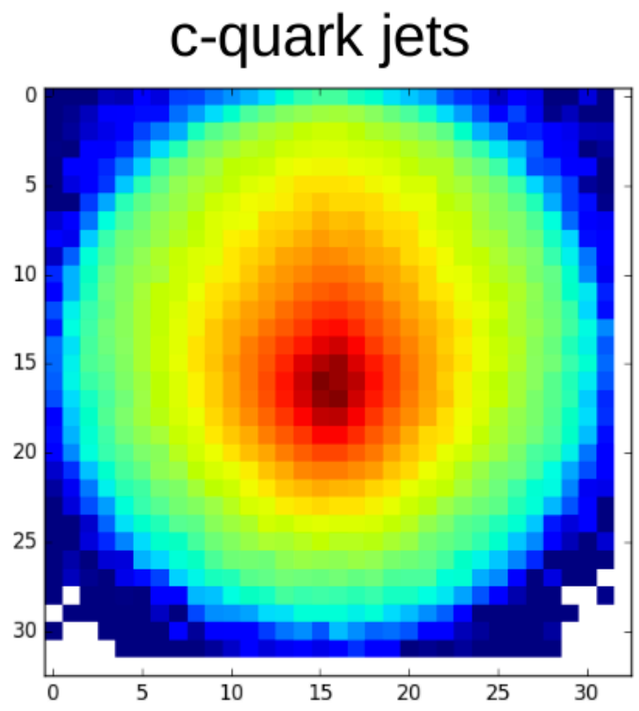
The energy deposit in calorimeter cells, which are the inputs to jet forming can be represented as pixels in a two-dimensional y–ϕ plane, treating the jet as an image. This allows using CNN, as employed in image recognition and classification problems, to be used in jet identification. Since the actual detector geometry is not perfectly regular, some preprocessing is required to represent the jet as an image, and standardise the representation.

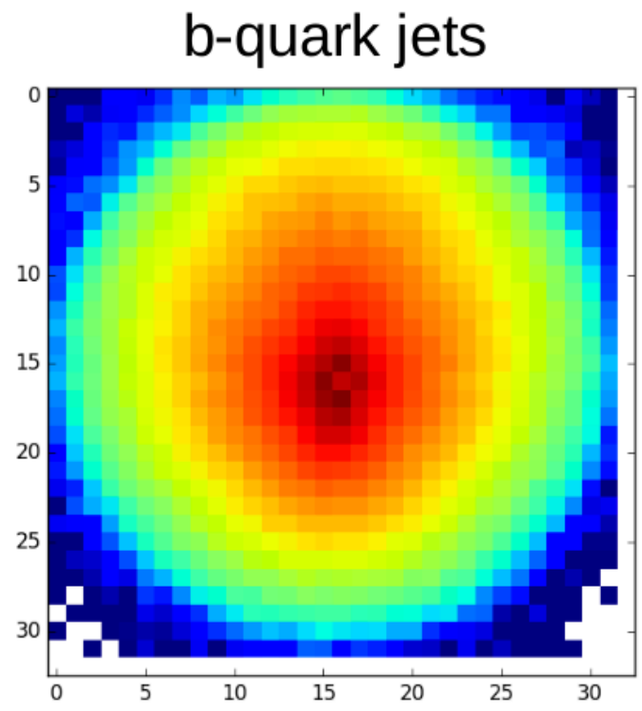
1. Translation: the leading p_{T} subjet (i.e. with the highest pixel intensity) is placed in the centre of the image.
2. Rotation: the subleading p_{T} subjet is aligned along the vertical axis of the jet-image.
3. Parity flip: the images are flipped over the vertical axis such that the right side of the image has a higher energy than the left.
4. Normalisation: each pixel in the image is multiplied by a normalisation factor such that the sum of the squared pixel information is equal to unity. This is required so that the algorithm is not misled by large shifts in intensity between two training images.

Jets as images









Use image processes toolkit!

Event Generation

- Multistep process
- Slowest is detector simulation, as interactions of particles with layers of detectors is a complex process

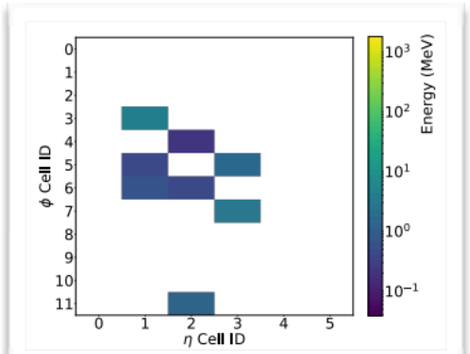
Slide from Michela Paganini

Shower Images

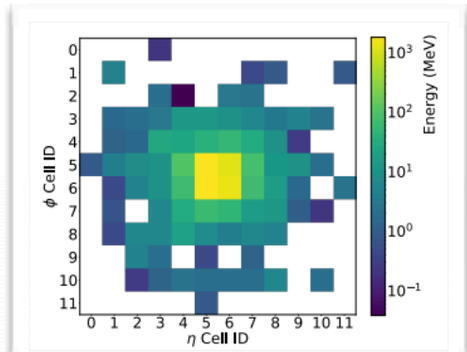
Yale

- Energy depositions in each layer as a **2D image**, similar to jet image

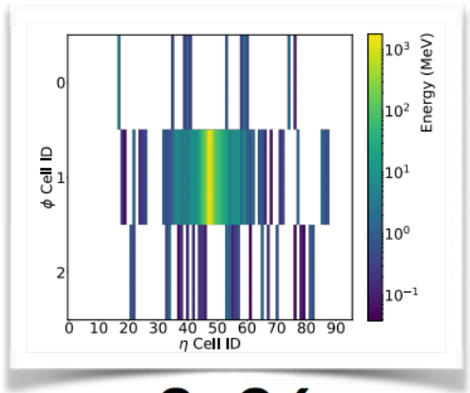
Layer	z segmentation [mm]	η segmentation [mm]	ϕ segmentation [mm]
0	90	5	160
1	347	40	40
2	43	80	40



12x6

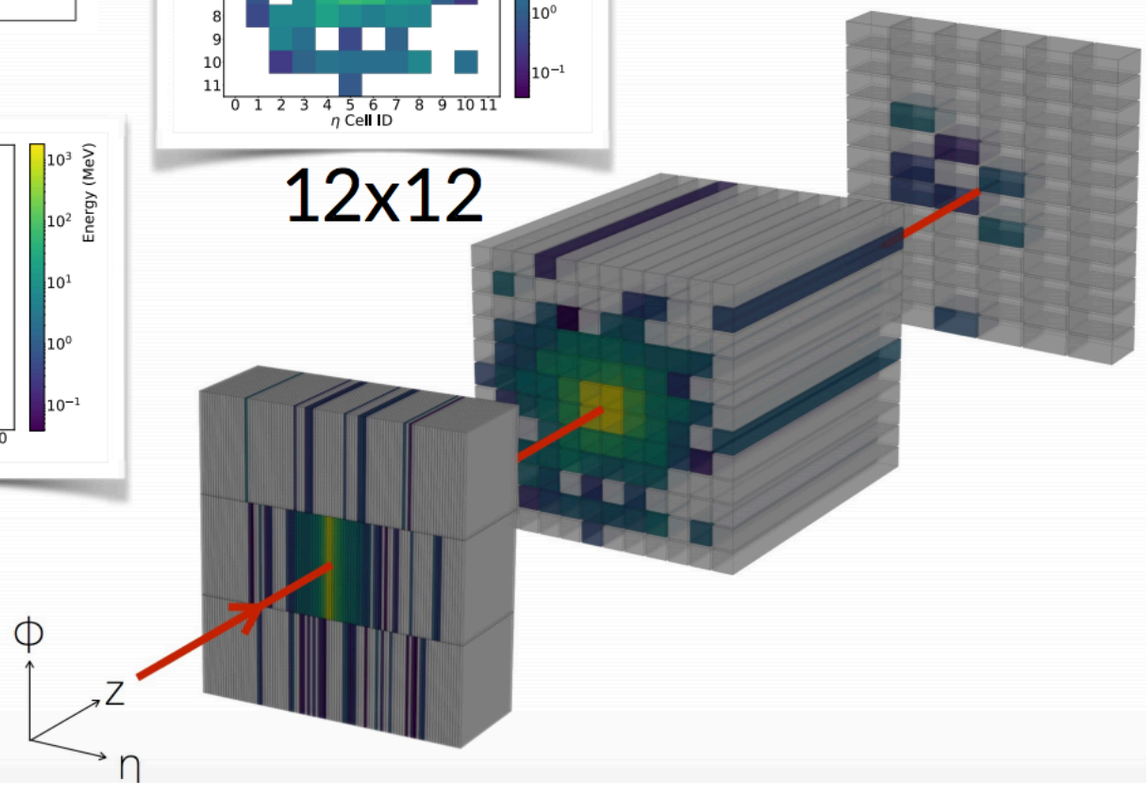


12x12



3x96

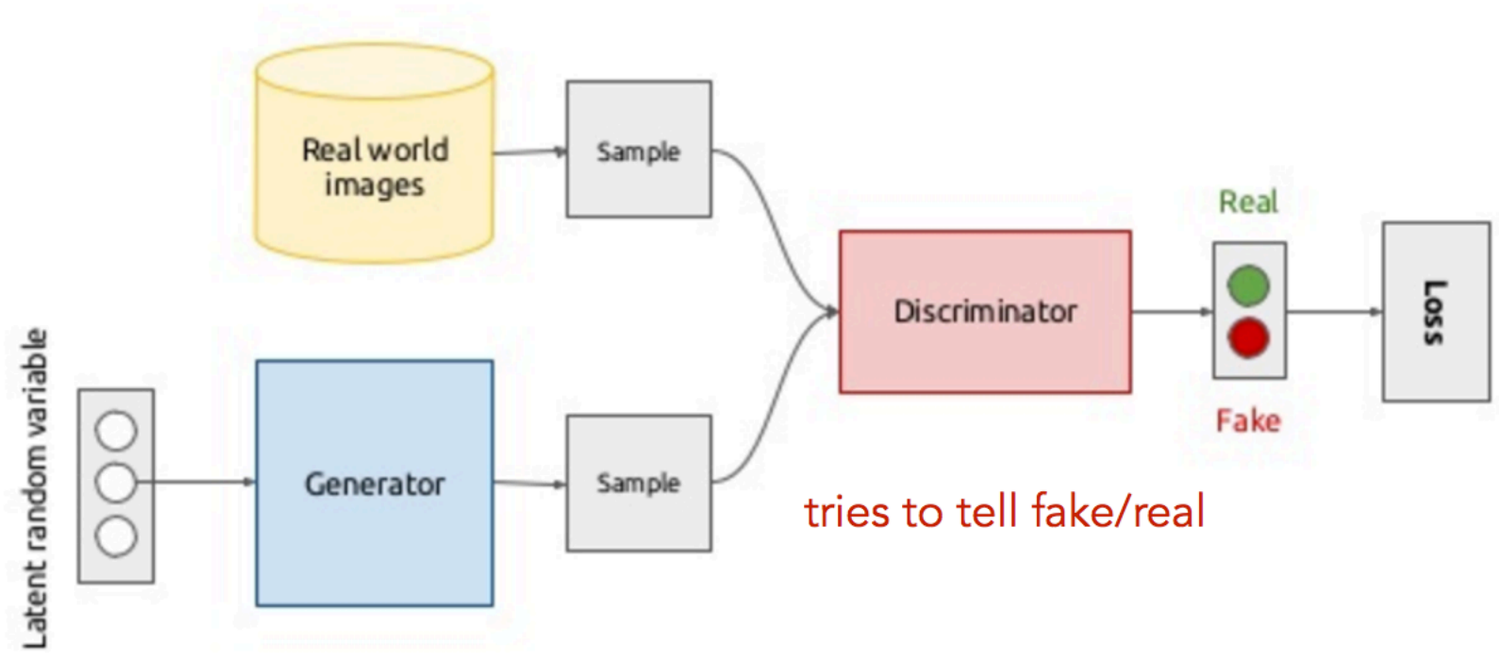
- <u>Goal</u>: generate this fixed representation



Slide from Michela Paganini

Use GAN: CaloGan

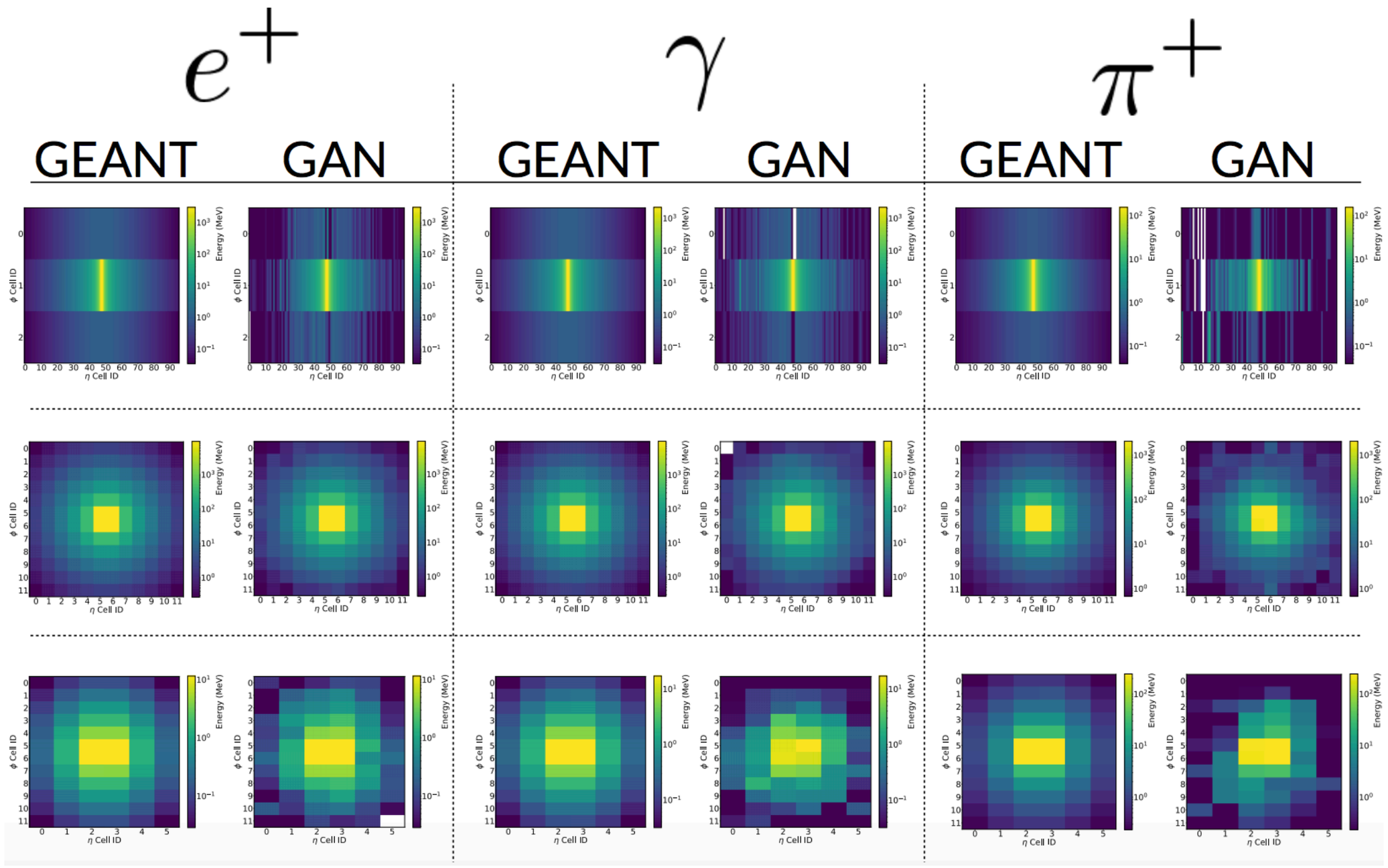
Turn generative modeling into a
two player, non-cooperative game.



tries to tell fake/real

tries to produce real looking samples

Slide from Michela Paganini



Slide from Michela Paganini

Can obtain an evaluation time speed up of **100,000x** on GPU!

Generation Method	Hardware	Batch Size	milliseconds/shower
GEANT4	CPU	N/A	1772
CALOGAN	CPU	1	13.1
		10	5.11
		128	2.19
		1024	2.03
	GPU	1	14.5
		4	3.68
		128	0.021
		512	0.014
		1024	0.012

Unique problems we face!

- Supervised learning makes us all too dependent on simulation, which we know is not perfect!

- Unsupervised learning on data is useful for anomaly detection, but harder to measure performance…

- No robust way of assessing systematic uncertainties.

The eventual goal is not only to have these ML methods give a better result than the previous methods, but also to extract physics information from what the algorithms are learning. So rather than focussing solely on what we teach the machine, what the machine teaches us is getting more attention. In this context, we can refer to what a lot of particle physicists thought at the beginning of the ML era, that *deep thinking triumphs deep learning*. Now the paradigm has shifted to *deep thinking and deep learning together*!