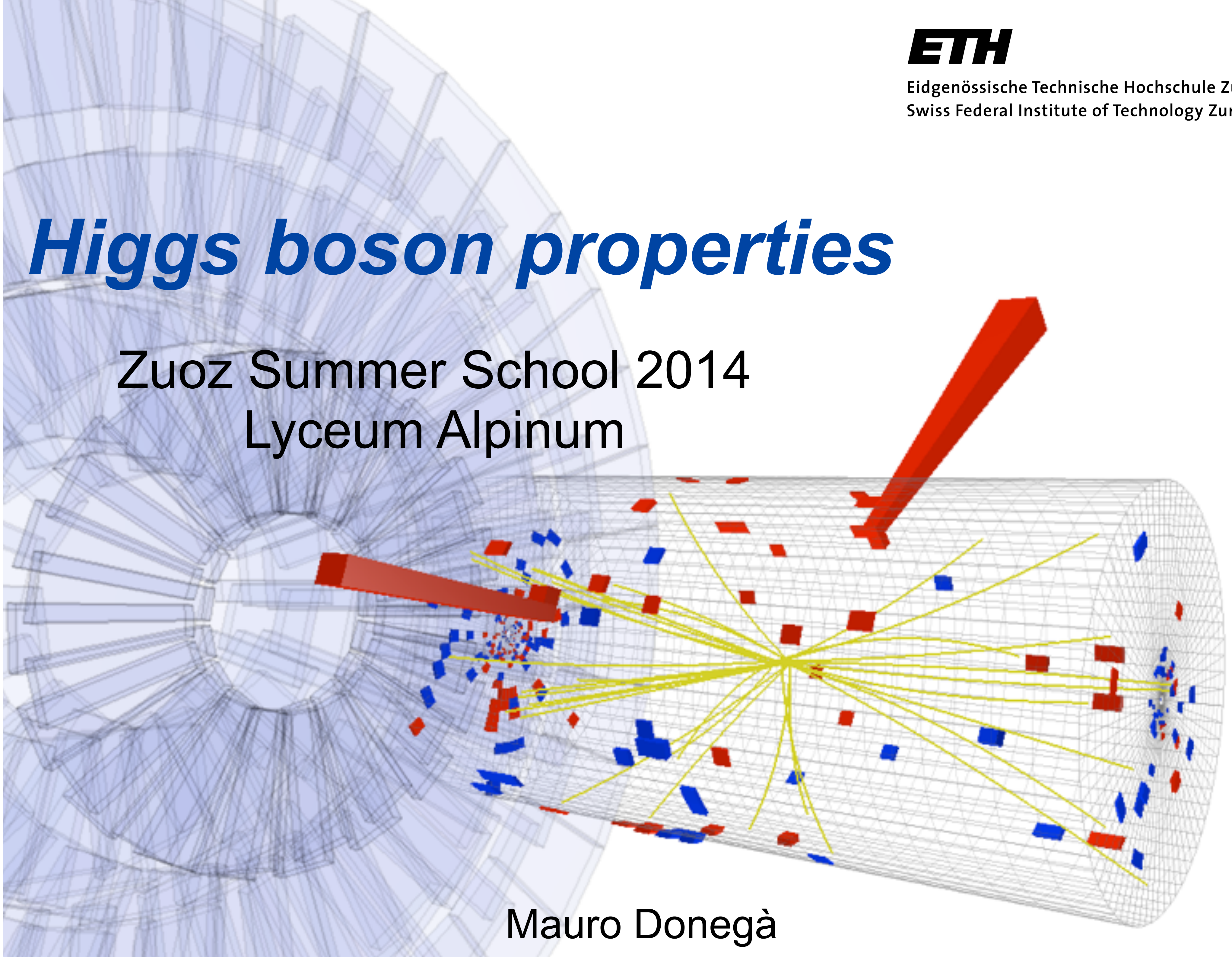
ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Higgs boson properties

Zuoz Summer School 2014
Lyceum Alpinum



Mauro Donegà

Lecture 1

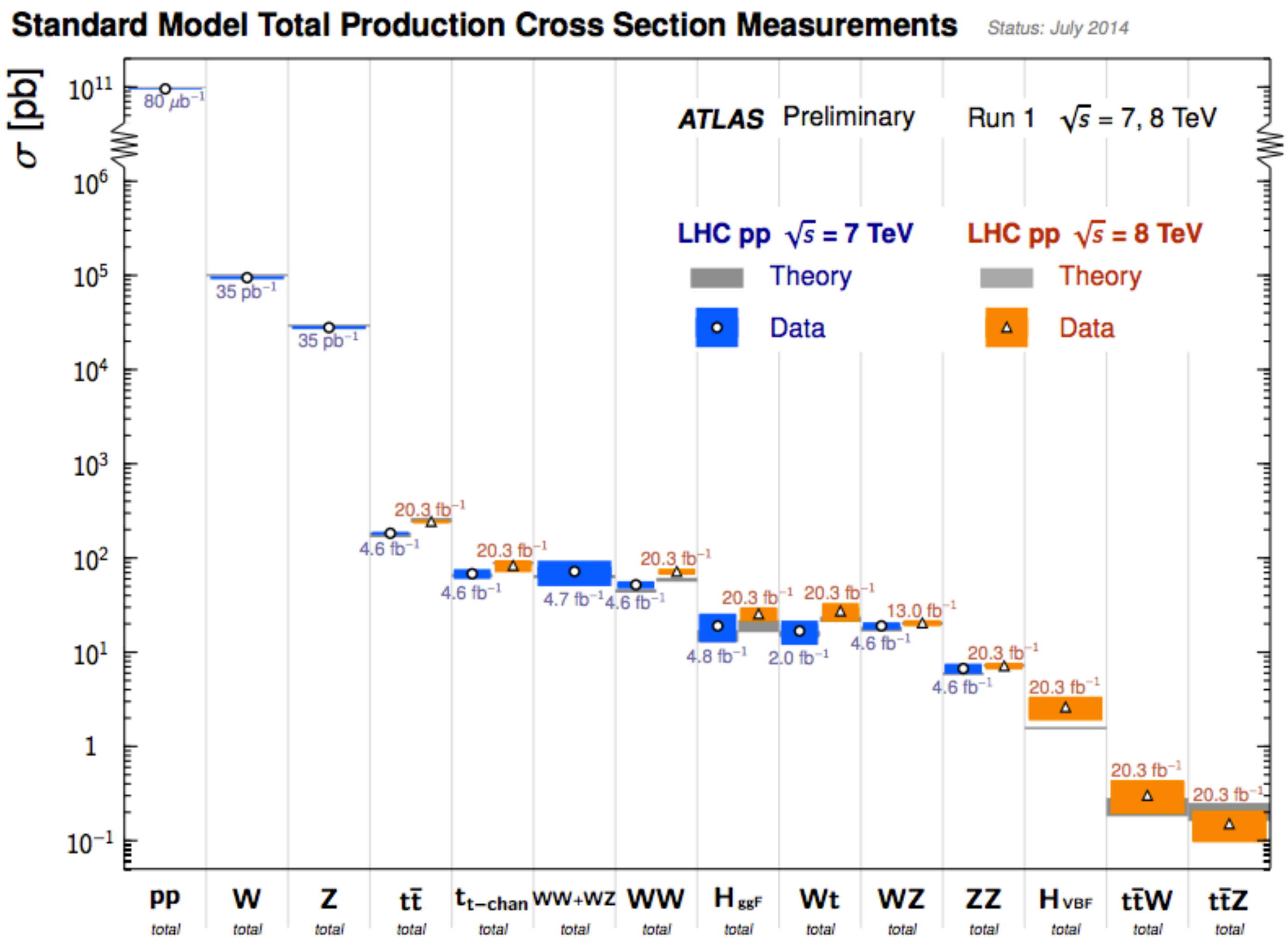
Detectors
BDT
Statistics
Dissect one analysis
Main decay channels
top/Higgs
Coupling measurements
Differential measurements
Mass measurements
Width measurements
Spin structure

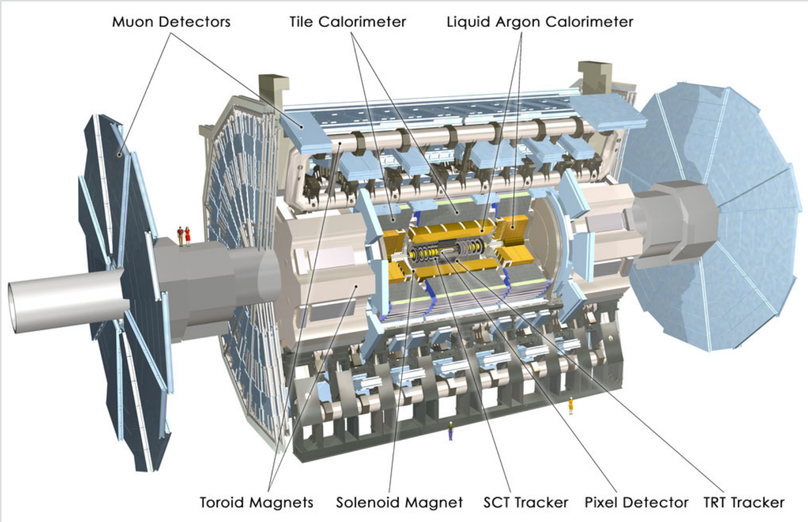
Not covered: searches and a lot more…

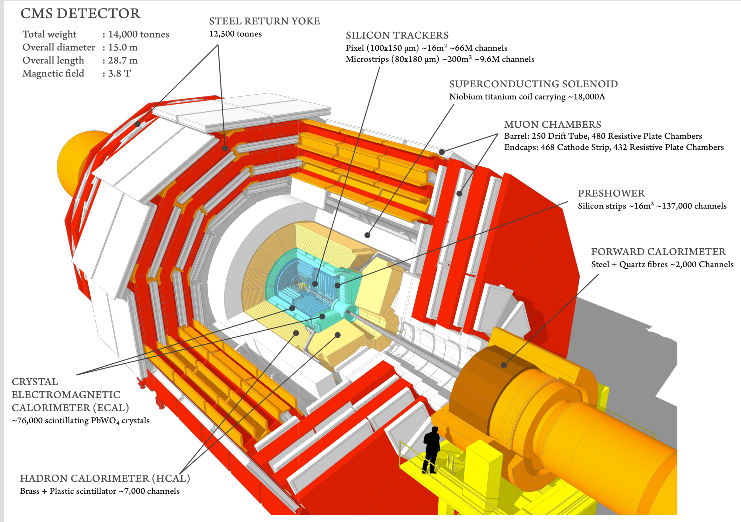
Toolbox

Theoretical tools

SM processes are now calculated at very high level of precision
"Next-(Next)-(Next)-…to revolution" of the past ~decade




Muon Detectors
Tile Calorimeter
Liquid Argon Calorimeter
Toroid Magnets
Solenoid Magnet
SCT Tracker
Pixel Detector
TRT Tracker


CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm) ~16m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 468 Cathode Strip, 432 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL
ELECTROMAGNETIC
CALORIMETER (ECAL)
~76,000 scintillating $PbWO_4$ crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

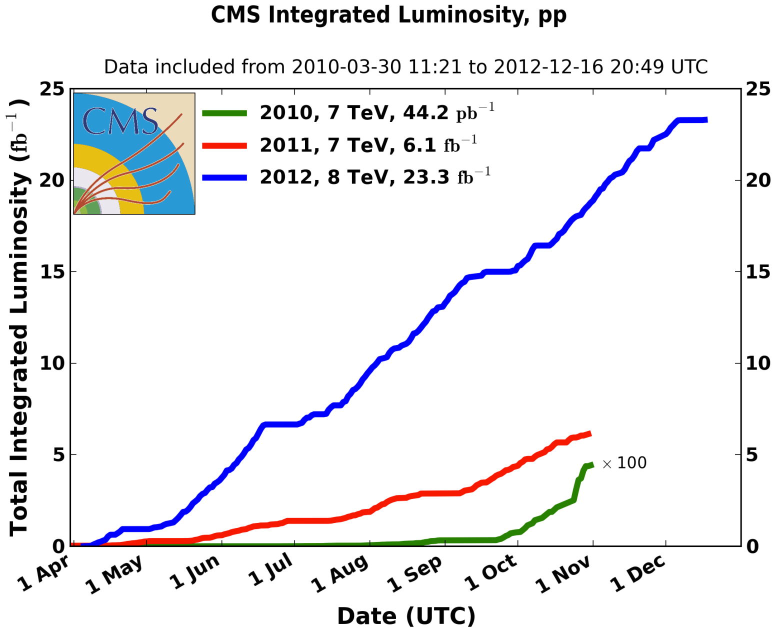
Detectors

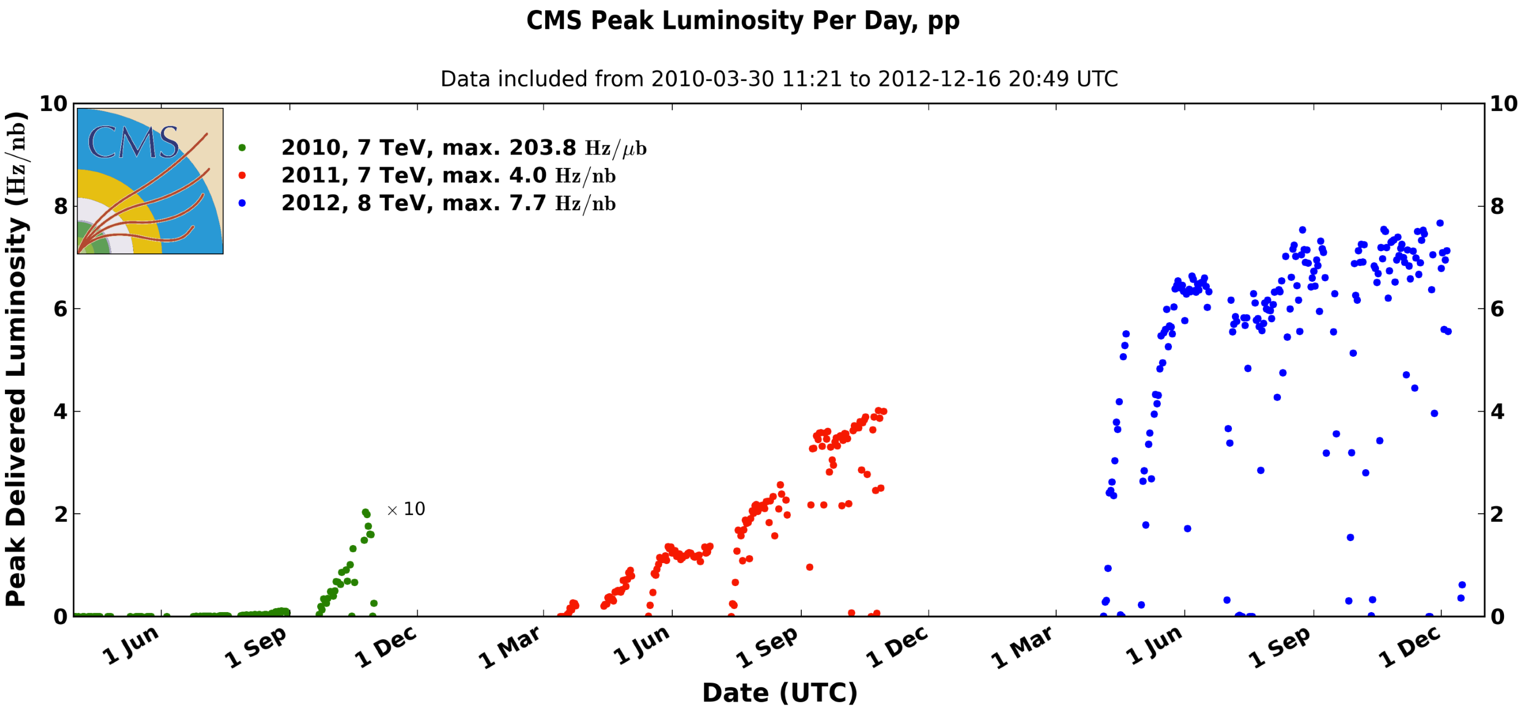
LHC Run 1

~30 fb^{-1} delivered to experiments

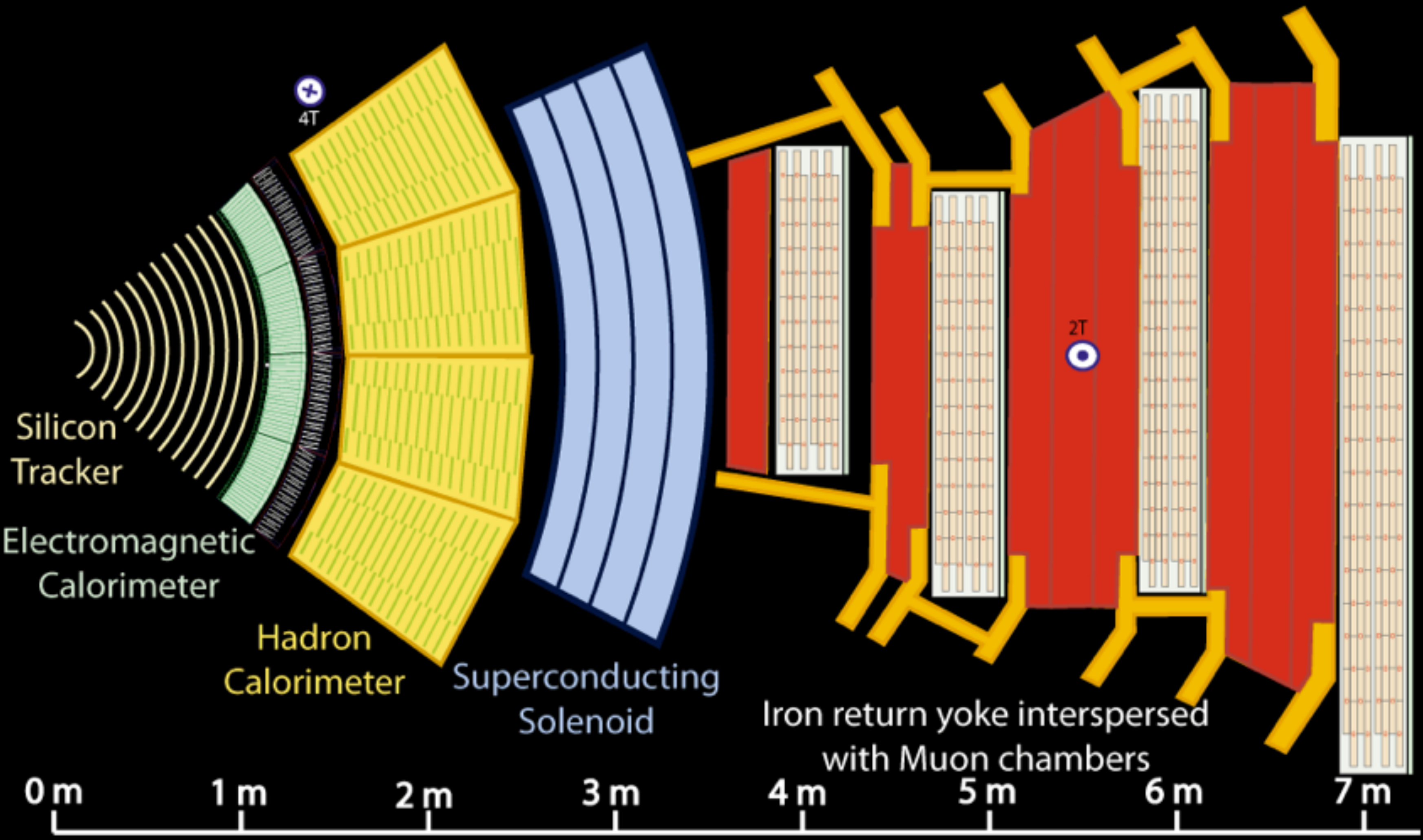
7 TeV: ~44 pb^{-1} in 2010,
~6 fb^{-1} in 2011
8 TeV: ~23 fb^{-1} in 2012

Peak luminosity > 7 Hz/nb
#evts = L σ(pp) ~ o(10^9) event/sec

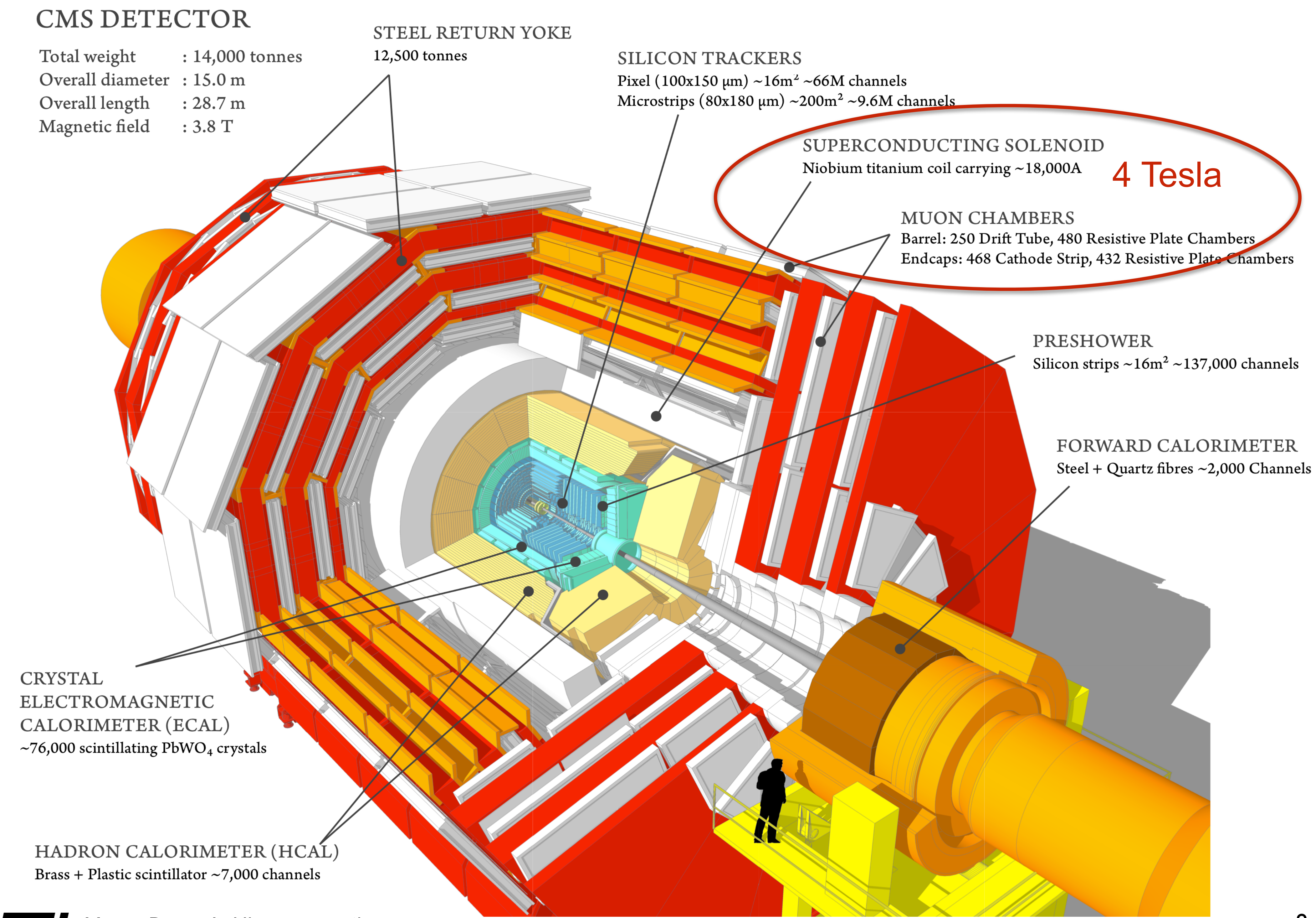




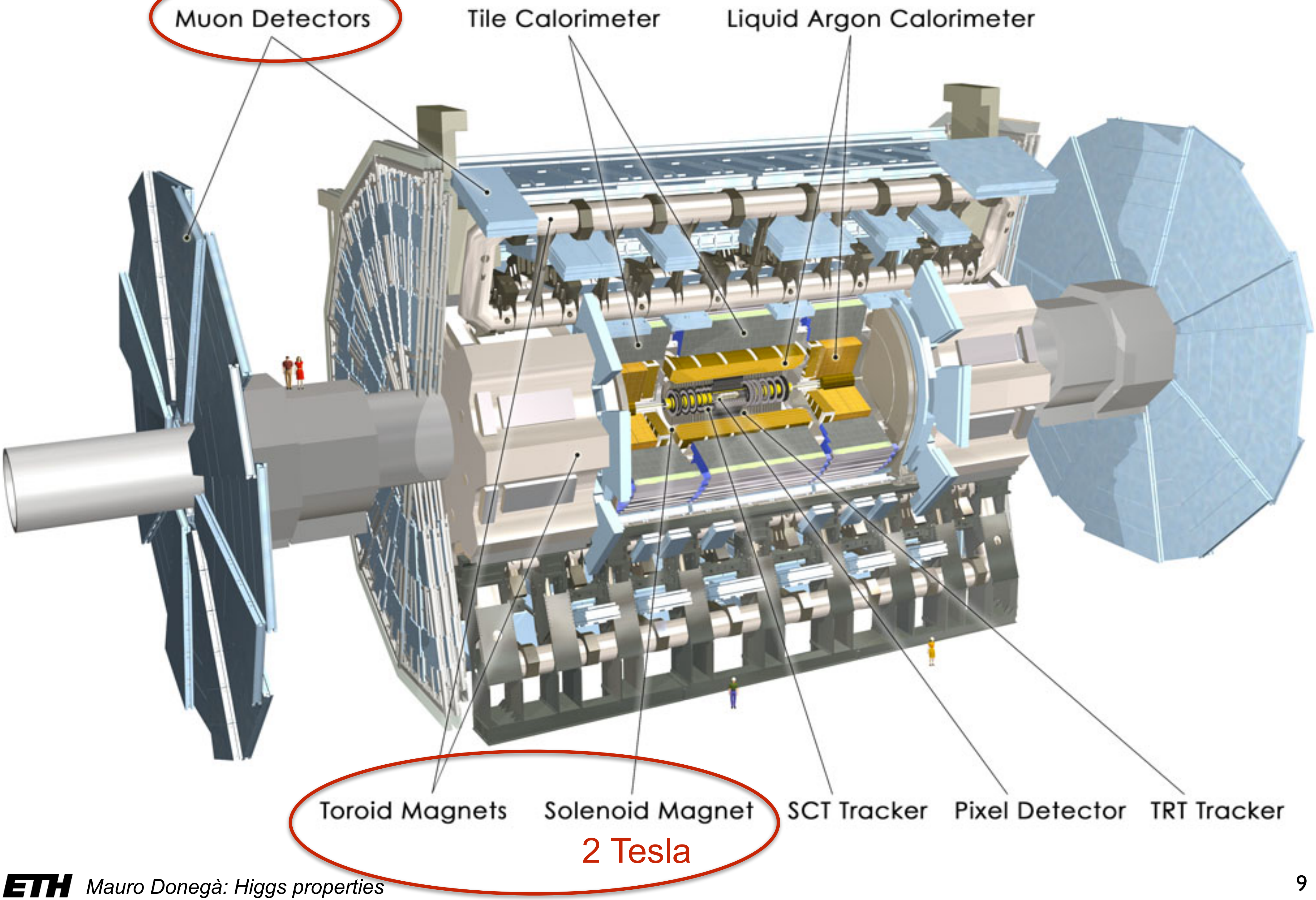
HEP collider detector



Different experiments choose different technologies


CMS DETECTOR
Total weight : 14,000 tonnes
Overall diameter : 15.0 m
Overall length : 28.7 m
Magnetic field : 3.8 T
STEEL RETURN YOKE
12,500 tonnes
SILICON TRACKERS
Pixel (100x150 μm) ~16m² ~66M channels
Microstrips (80x180 μm) ~200m² ~9.6M channels
SUPERCONDUCTING SOLENOID
Niobium titanium coil carrying ~18,000A
4 Tesla
MUON CHAMBERS
Barrel: 250 Drift Tube, 480 Resistive Plate Chambers
Endcaps: 468 Cathode Strip, 432 Resistive Plate Chambers
PRESHOWER
Silicon strips ~16m² ~137,000 channels
FORWARD CALORIMETER
Steel + Quartz fibres ~2,000 Channels
CRYSTAL ELECTROMAGNETIC CALORIMETER (ECAL)
~76,000 scintillating PbWO4 crystals
HADRON CALORIMETER (HCAL)
Brass + Plastic scintillator ~7,000 channels

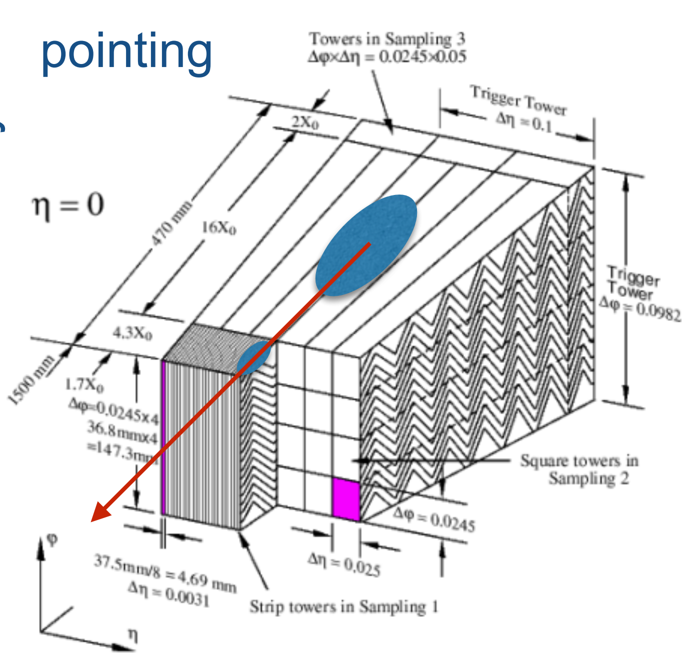
ATLAS


Muon Detectors
Tile Calorimeter
Liquid Argon Calorimeter
Toroid Magnets
Solenoid Magnet
2 Tesla
SCT Tracker
Pixel Detector
TRT Tracker

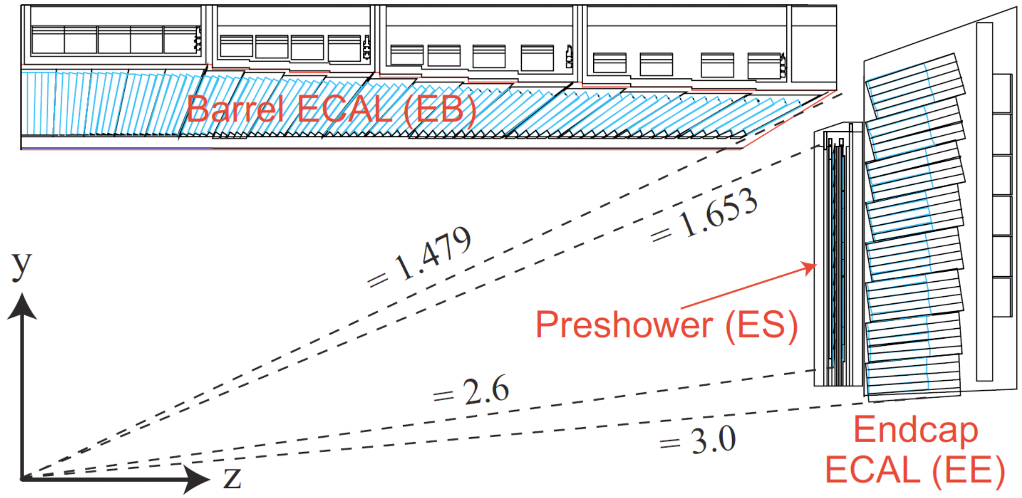
Trackers and e.m. calorimeters

ATLAS LAr accordeon
Ouside the Solenoid/Cryostat

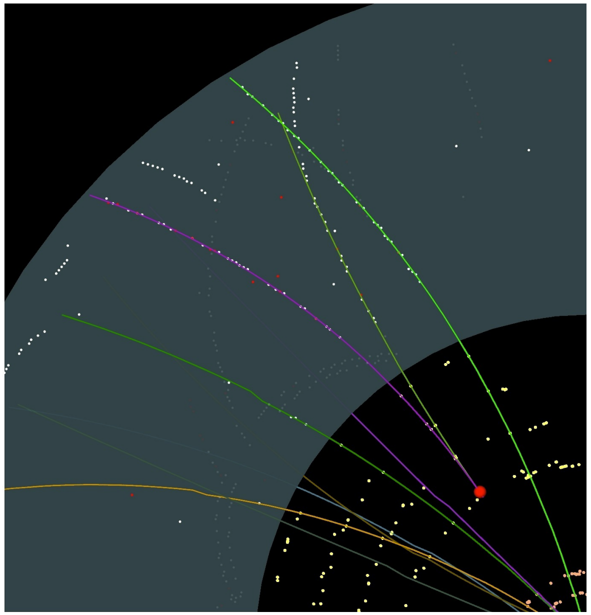
pointing



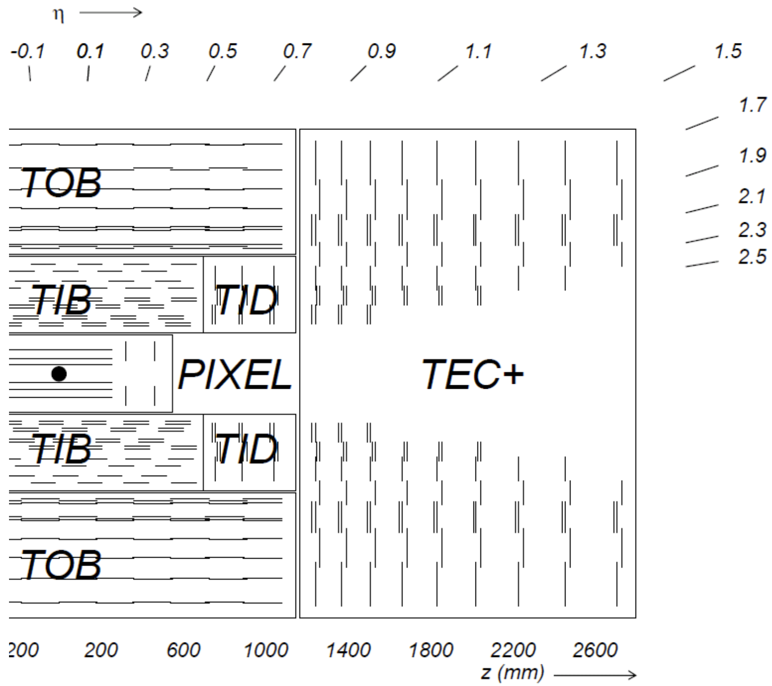
CMS crystals $PbWO_4$
Inside the Solenoid



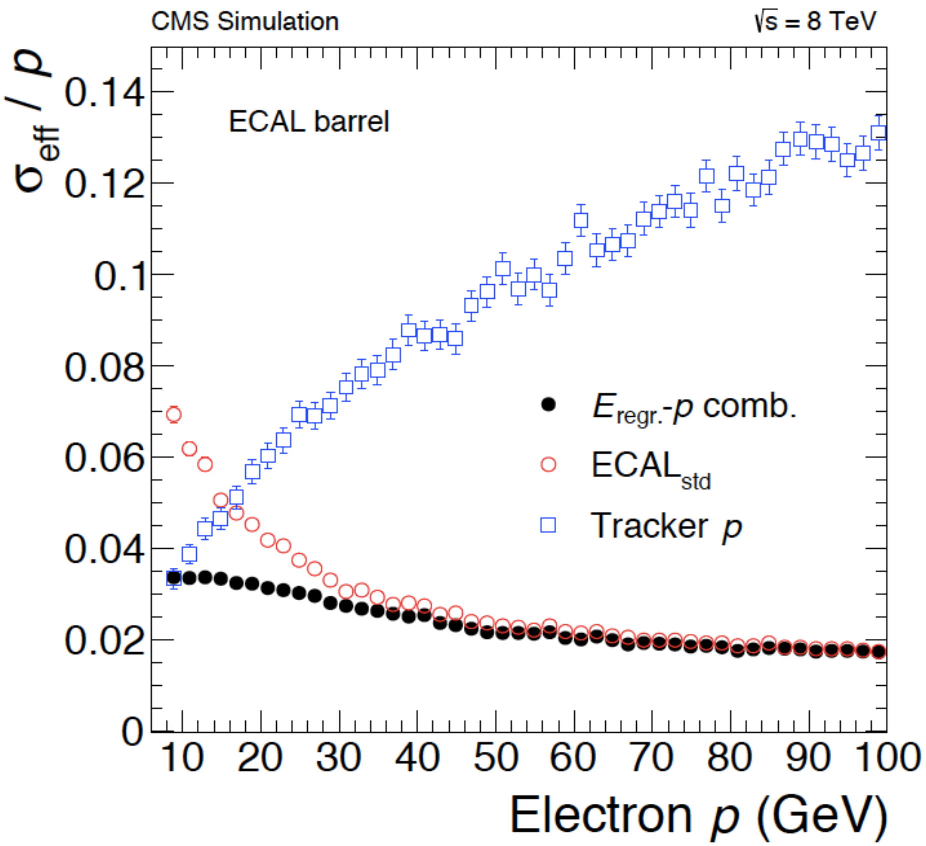
TRT e/hadron separation

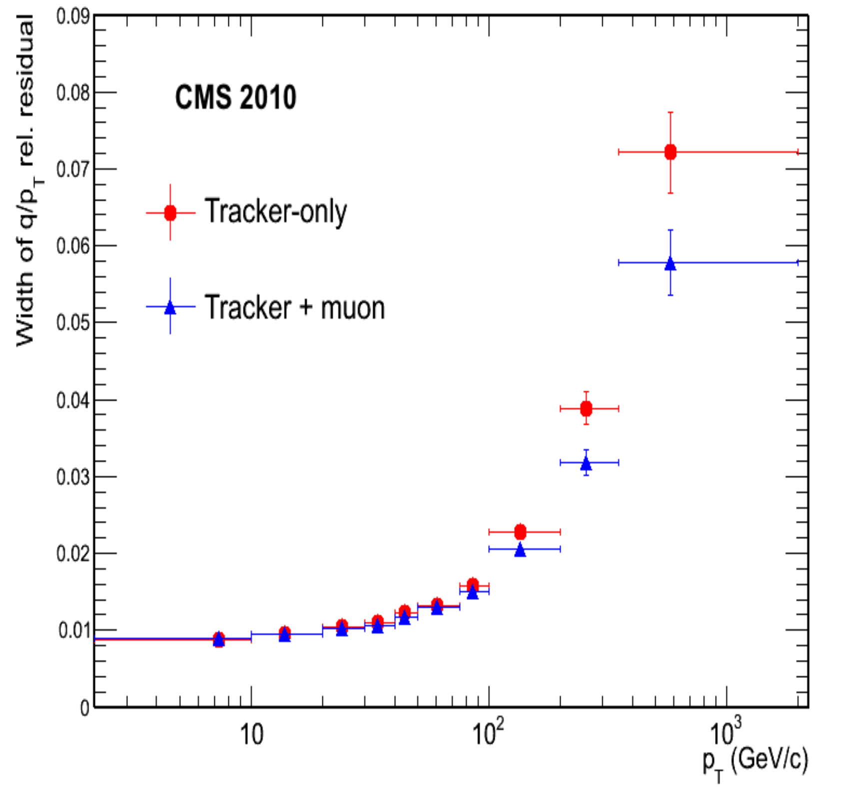


All Silicon



electrons: tracker / calorimeter
muons: tracker / mu-spectrometer

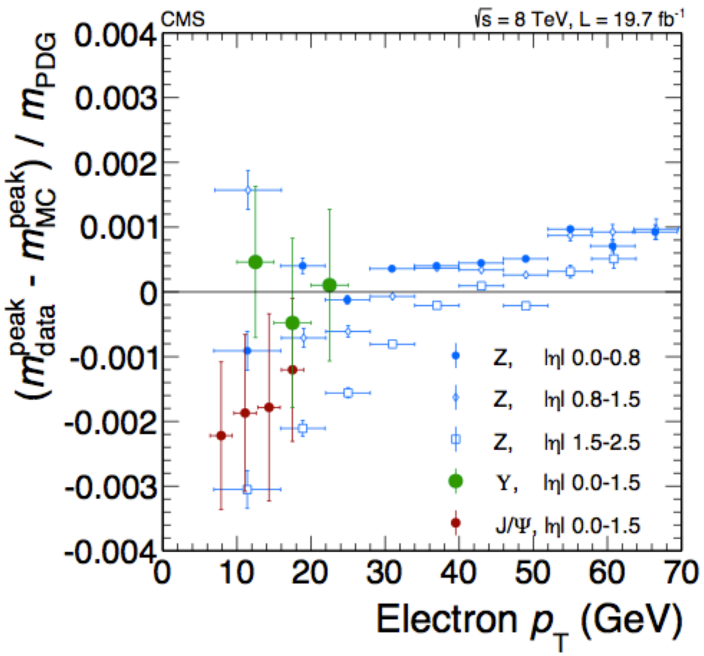


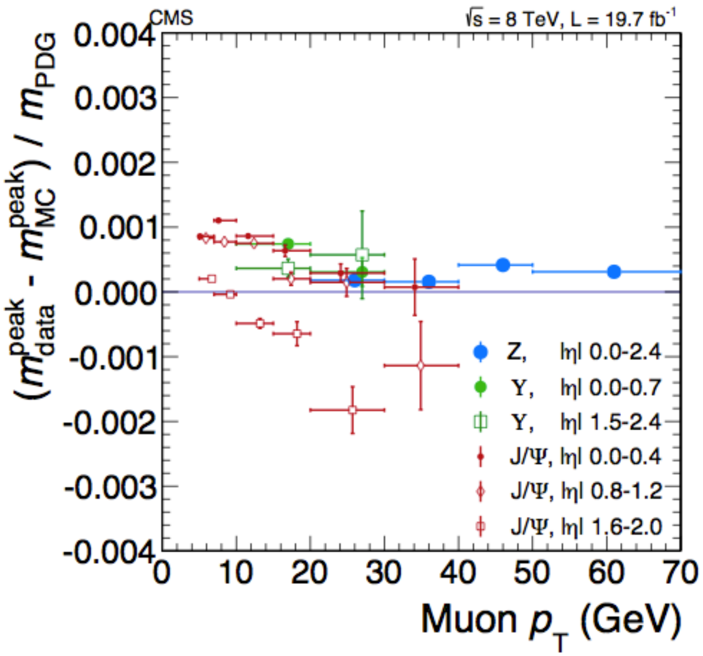


electron

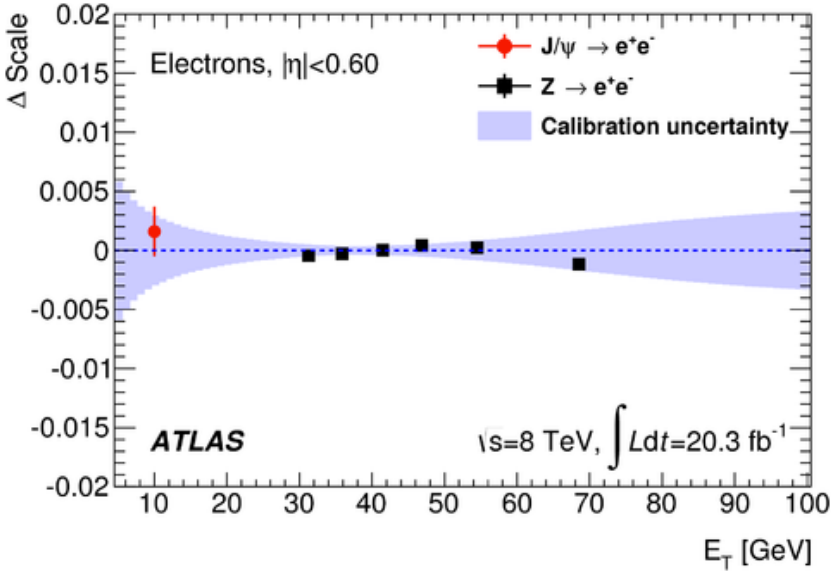
muons

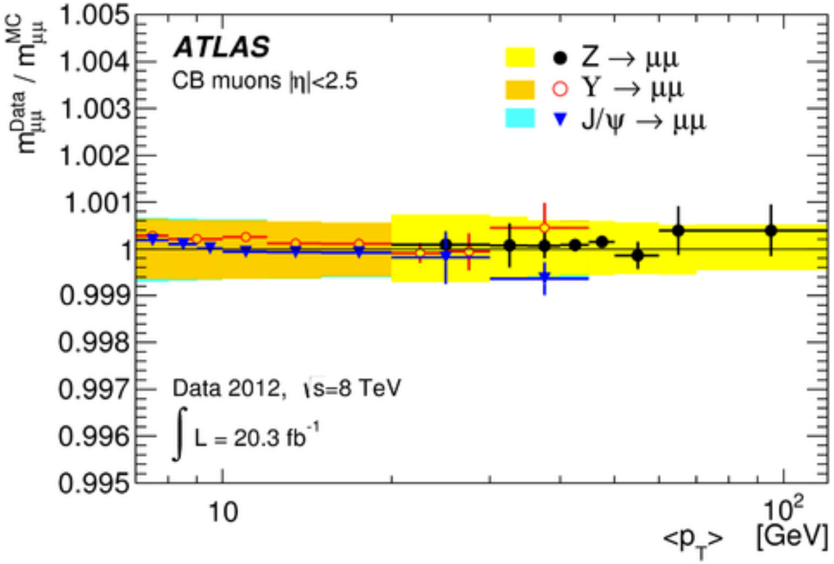
CMS



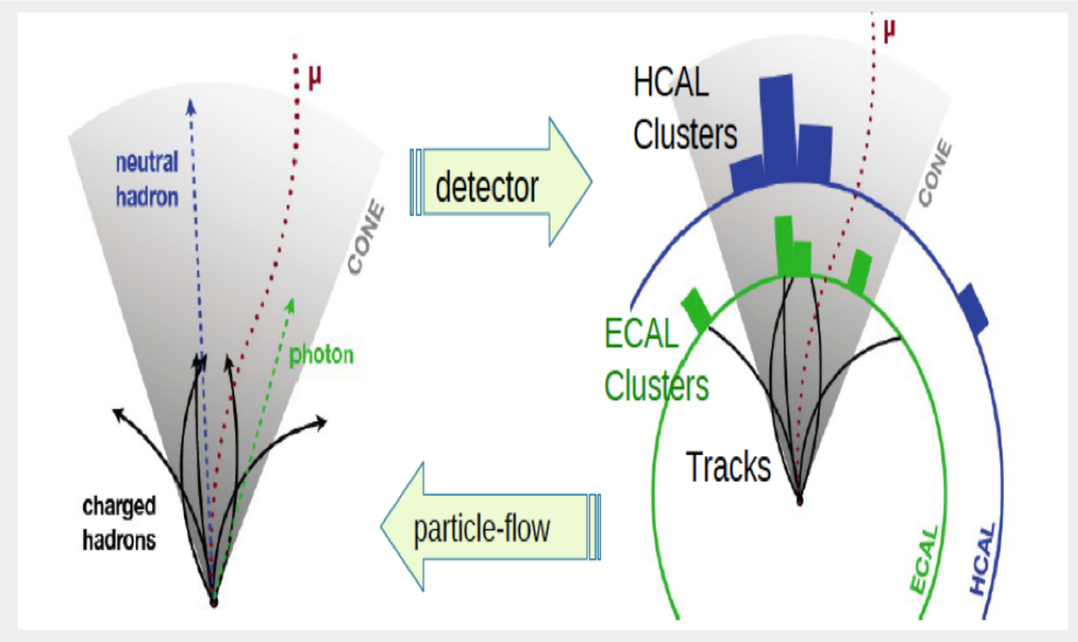


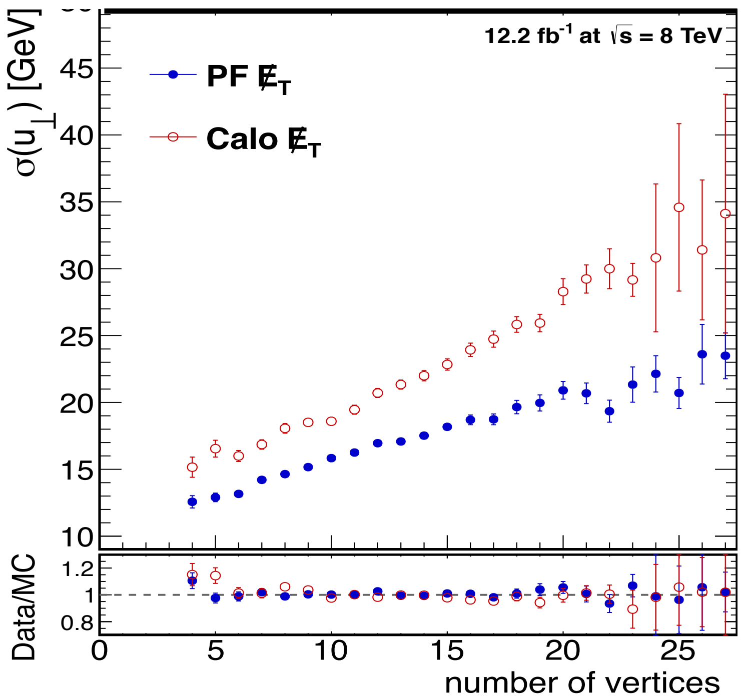
ATLAS





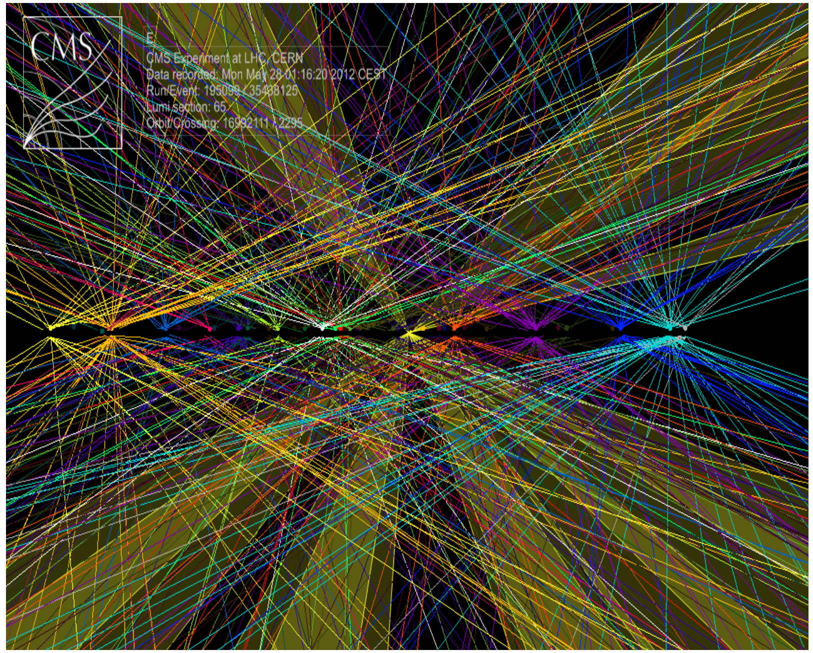
ETmiss - CMS Particle Flow





Take full advantage of the high granularity of the tracker and ECAL and of the 4T magnetic field.
Reconstruct each single object “as at generator level”.
Main impact on MET

One word about Pile up

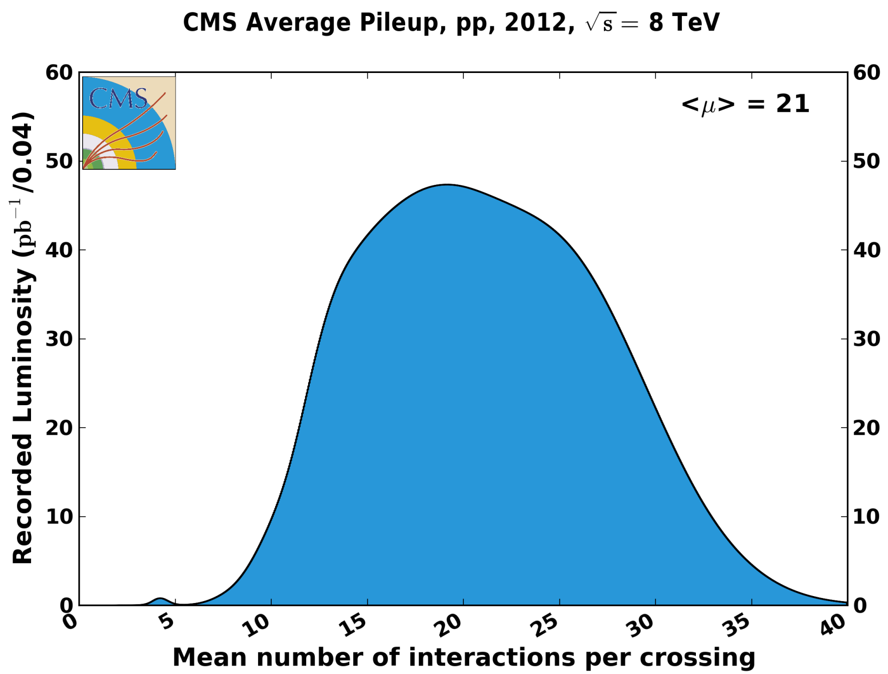


multiple pp interactions in one bunch crossing

It has a strong influence on:
- objects identification (isolation)
- energy reconstruction
- reconstruction time

All analysis have set up specific tools to mitigate the loss of performance

One of the biggest experimental challenges in 2015

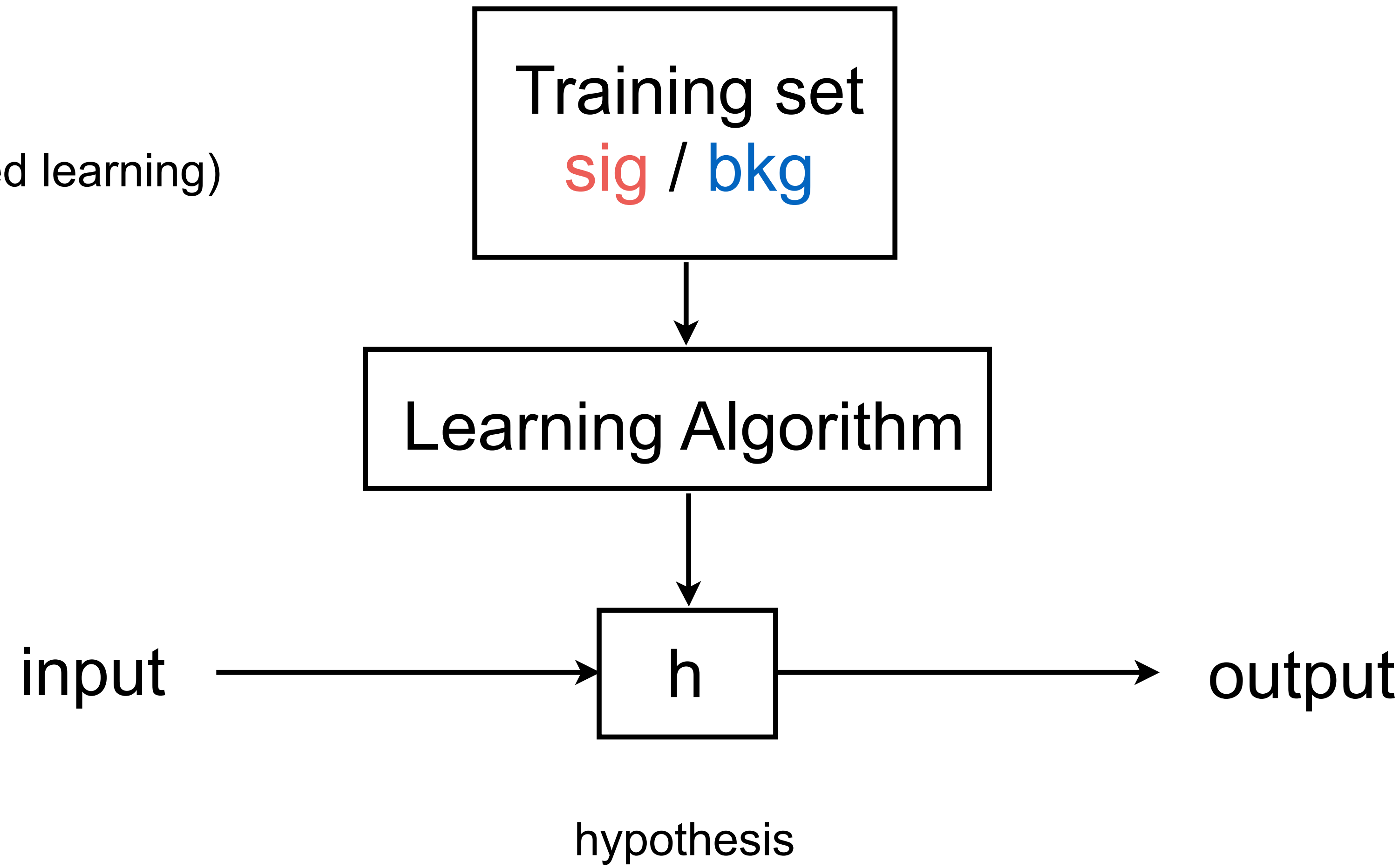


BDT

Multivariate Analysis: Learning algorithm

Two classes of problems:
classification (e.g. separate sig/bkg: output 1 for sign 0 for bkg)
regression (e.g. energy corrections: output will be a weigh such that (output x E_{rec})/E_{gen} =1)

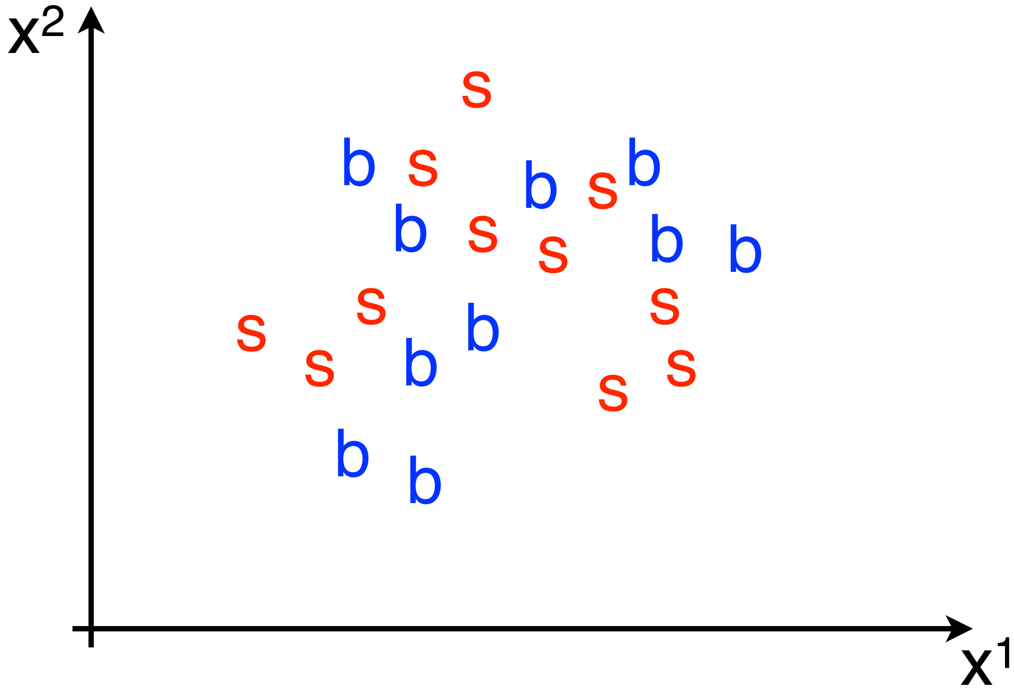
(Supervised learning)



Decision trees: classification

Ex: Training sample $\in \Re^2$

	classes
(x^1_1, x^2_1)	
...	
(x^1_i, x^2_i)	b
...	s
(x^1_n, x^2_n)	



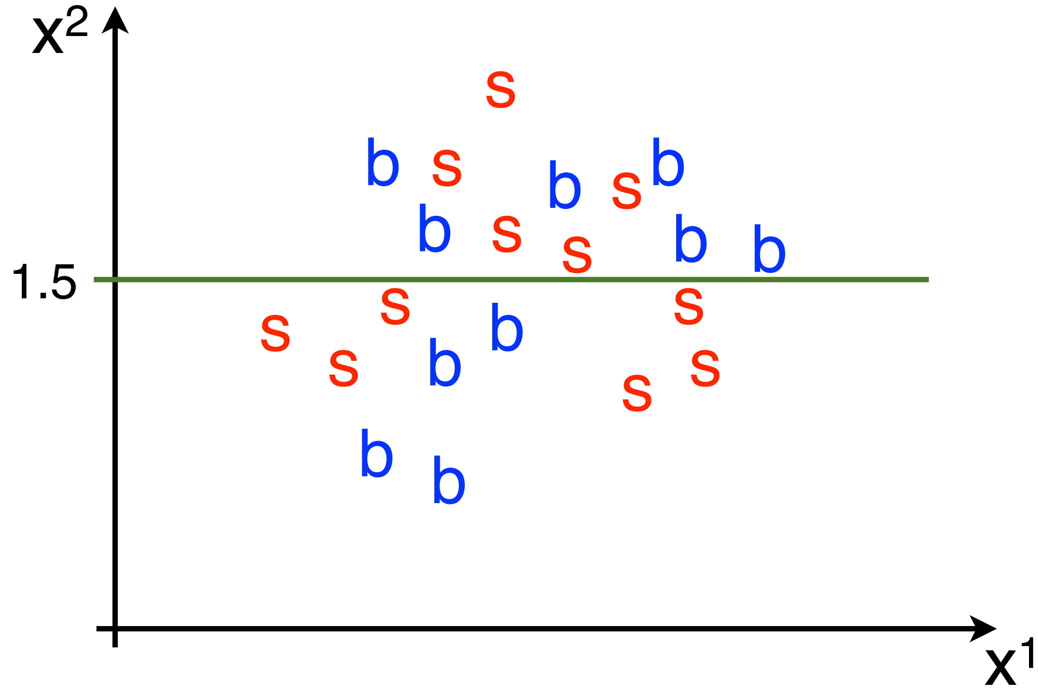
In this case it's difficult to have a good separation with a single linear cut.
Introduce non linearities

For the DT the idea is to separate the classes using placing several simple cuts
(i.e. binary splits of the data $x^i <$ value or $x^i >$ value)

Decision trees: classification

Training sample $\in \Re^2$

	classes
(x^1_1, x^2_1)	
...	
(x^1_i, x^2_i)	b
...	s
(x^1_n, x^2_n)	



Where to put the cuts ?
Strategy is to minimize the misclassification at each step

You choose the variable that provides the greatest increase in the separation measure (e.g., Gini index) in the two daughter nodes relative to the parent. (The same variable may be used at several nodes or ignored)

Define a metric for the separation:
the "Gini index" Gini = P(1-P) Where P =purity:

$$P = \frac{\sum_{\text{signal}} w_i}{\sum_{\text{signal}} w_i + \sum_{\text{background}} w_i}$$

This is maximum for P = 0.5 (no separation / random guess) and zero for P = 0 or 1.
(having purity of 0 or 1 is the same, you always have max separation)

Decision trees: classification

Training sample $\in \Re^2$

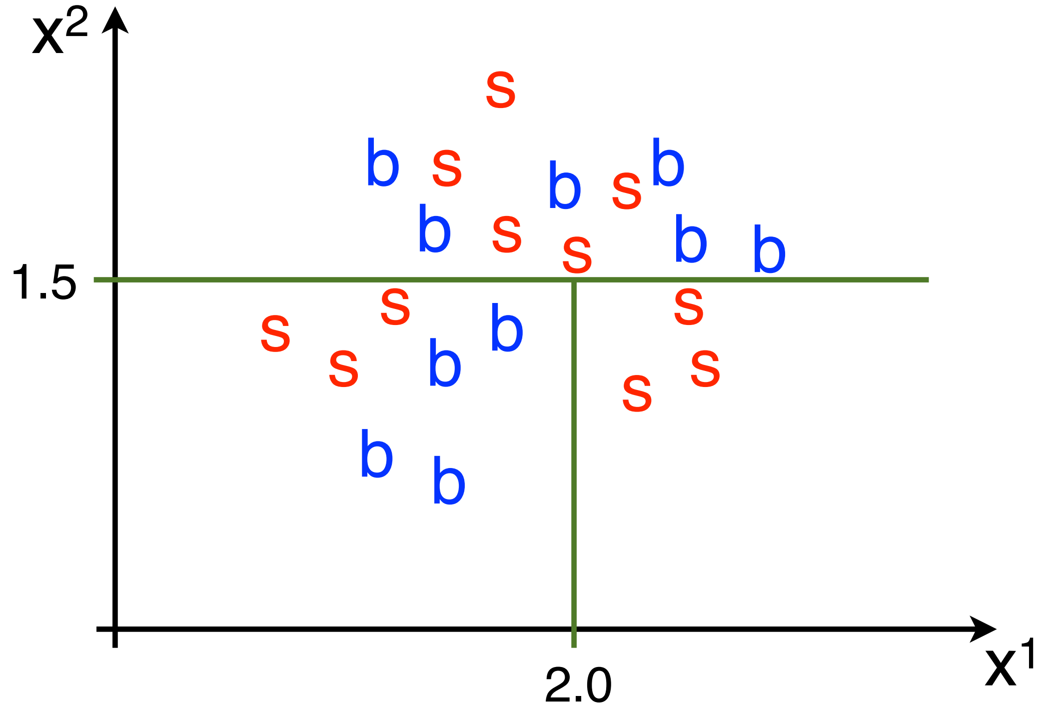
$(x^1{}_1, x^2{}_1)$ classes

...

$(x^1{}_i, x^2{}_i)$ b

...

$(x^1{}_n, x^2{}_n)$ s



Strategy is to minimize the misclassification at each leaf

Decision trees: classification

Training sample $\in \Re^2$

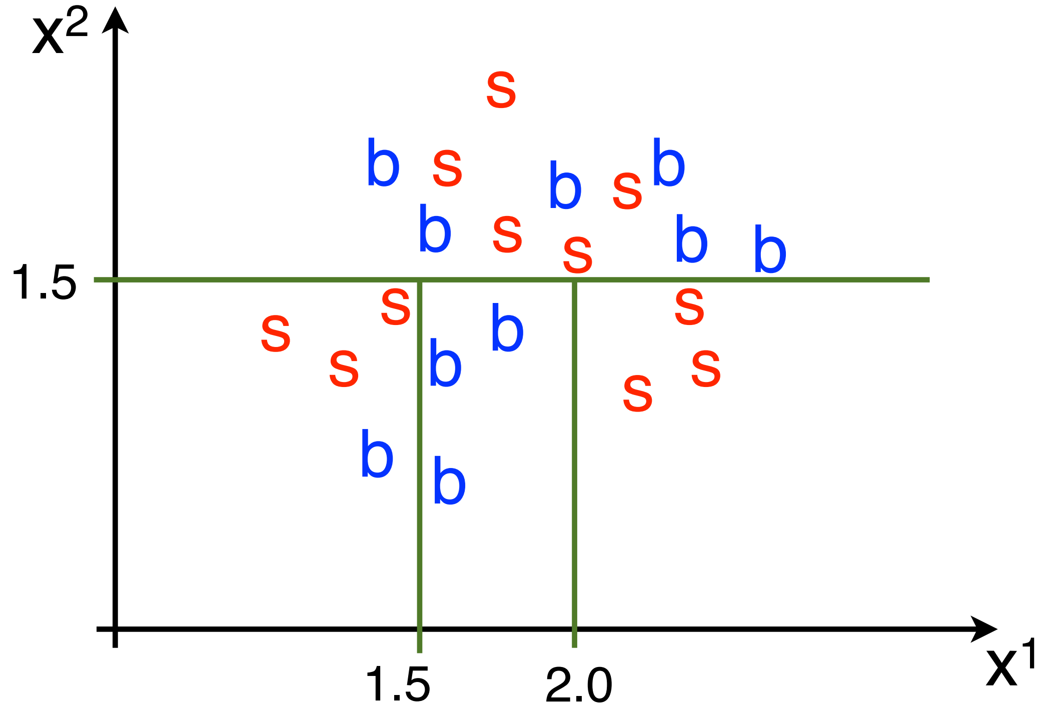
$(x^1{}_1, x^2{}_1)$ classes

...

$(x^1{}_i, x^2{}_i)$ b

...

$(x^1{}_n, x^2{}_n)$ s

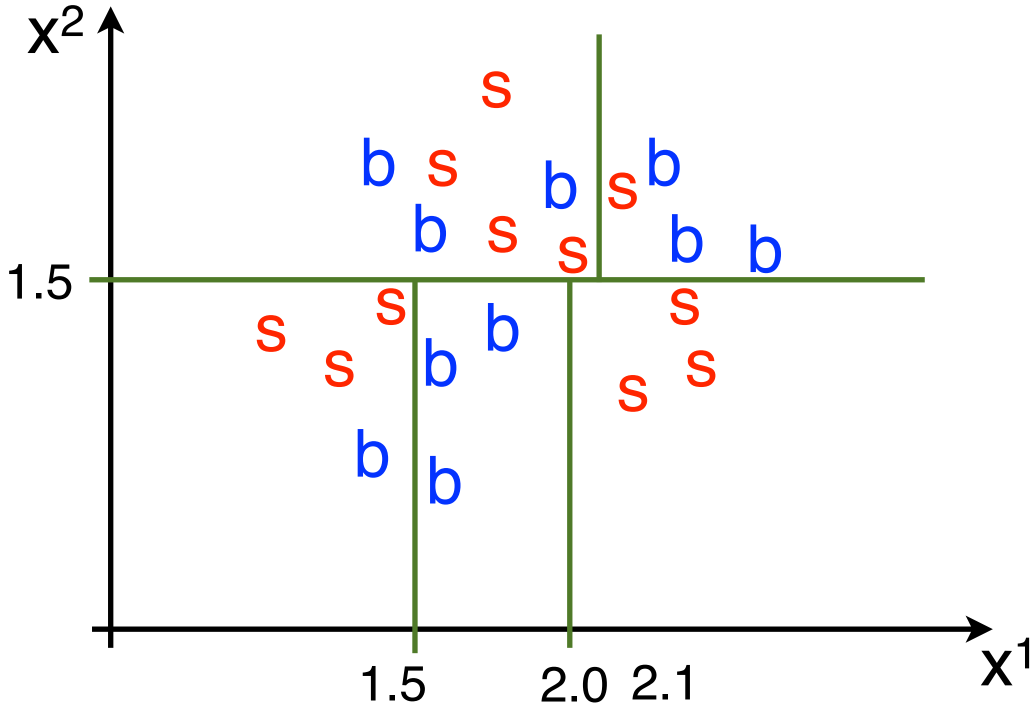


Strategy is to minimize the misclassification at each leaf

Decision trees: classification

Training sample $\in \Re^2$

$(x^1{}_1, x^2{}_1)$	classes
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	

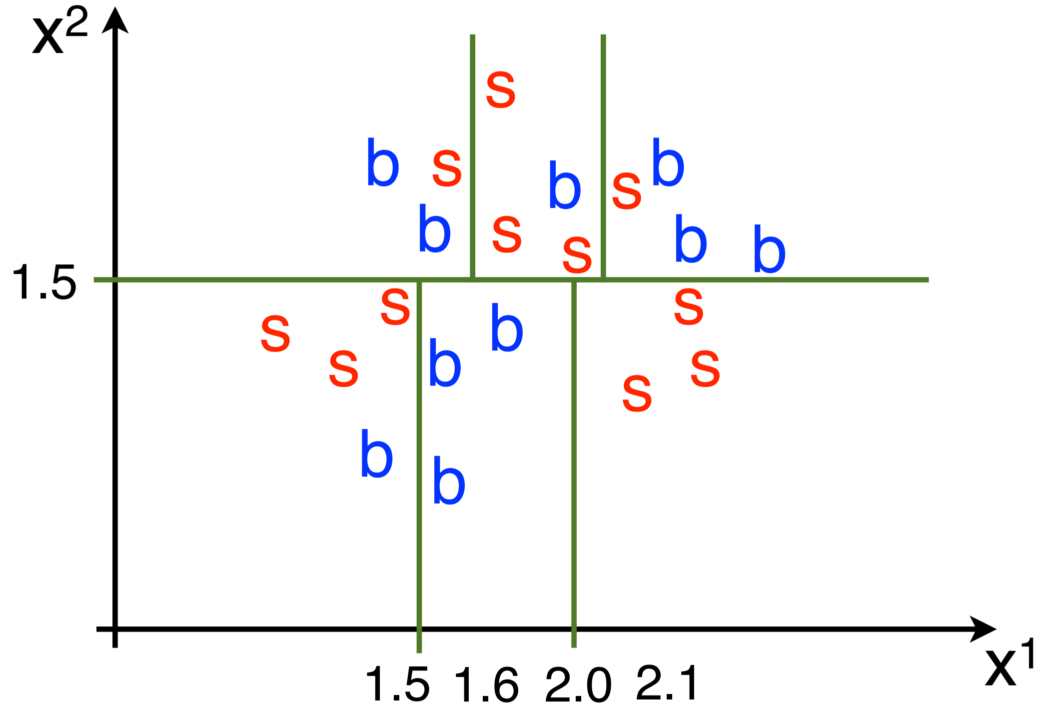


Strategy is to minimize the misclassification at each leaf

Decision trees: classification

Training sample $\in \Re^2$

$(x^1{}_1, x^2{}_1)$	classes
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	



Strategy is to minimize the misclassification at each leaf

Decision trees: classification

Training sample $\in \Re^2$

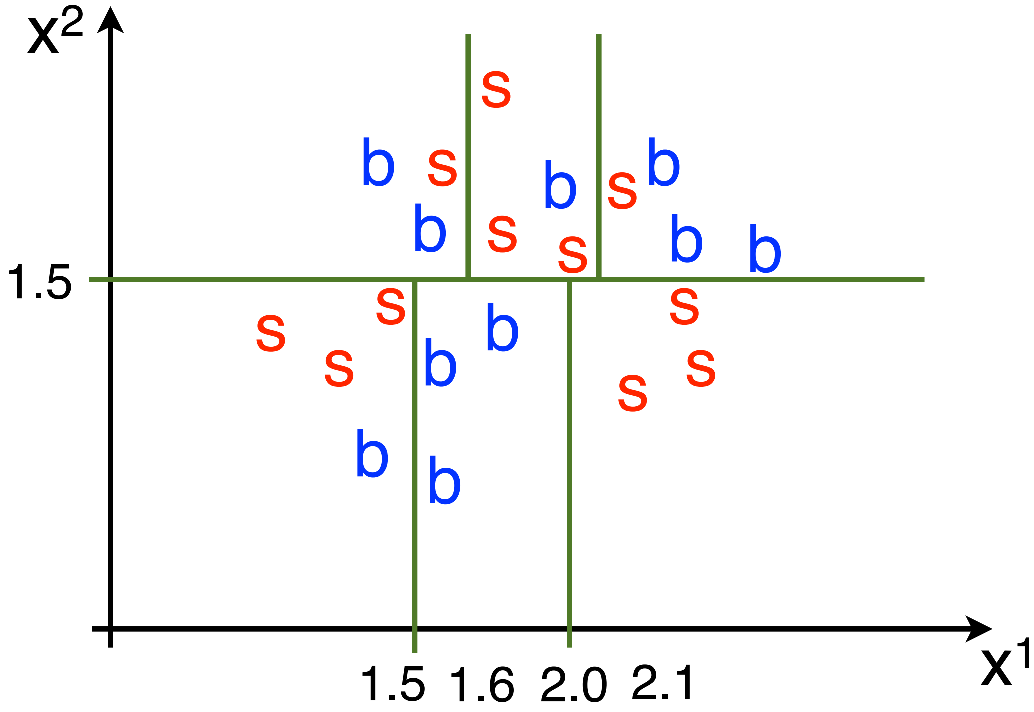
$(x^1{}_1, x^2{}_1)$ classes

...

$(x^1{}_i, x^2{}_i)$ b

...

$(x^1{}_n, x^2{}_n)$ s



Repeat until every region contains a “minimum” number of points.

Strategy is to minimize the misclassification at each leaf

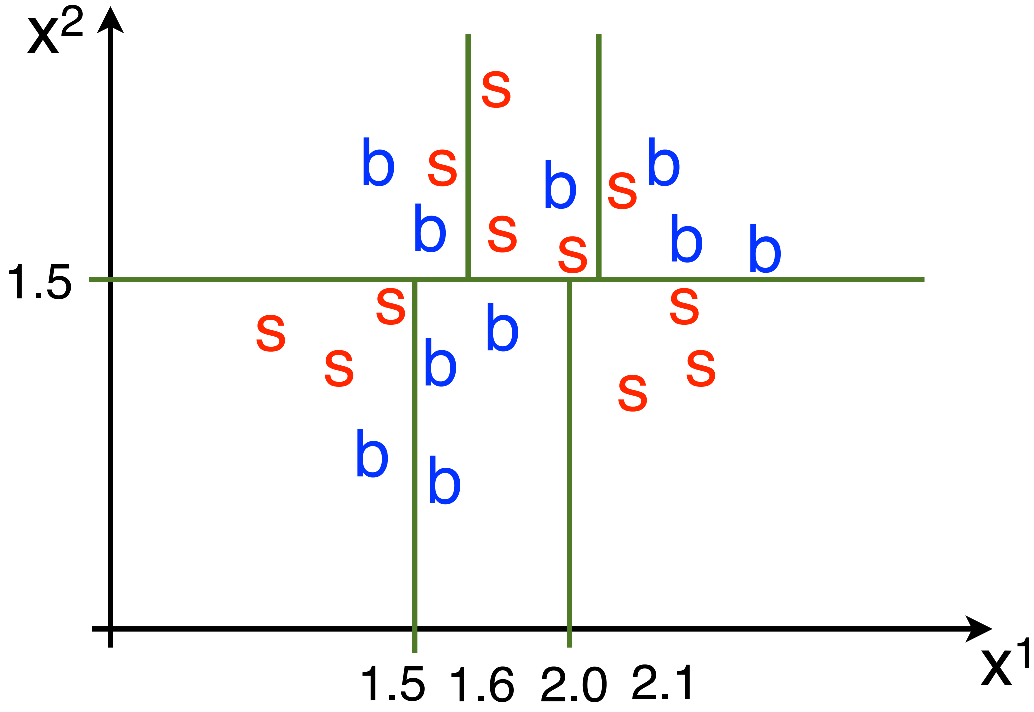
Decision trees: classification

Training sample $\in \Re^2$

	classes
$(x^1{}_1, x^2{}_1)$	
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	

Build a binary tree:

$\boxed{x^2 > 1.5}$



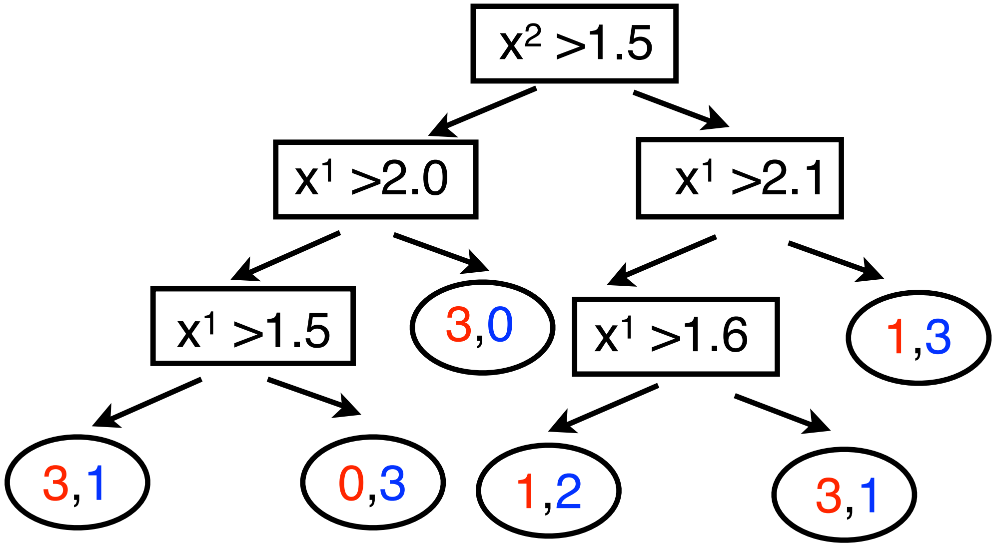
Strategy is to minimize the misclassification at each leaf

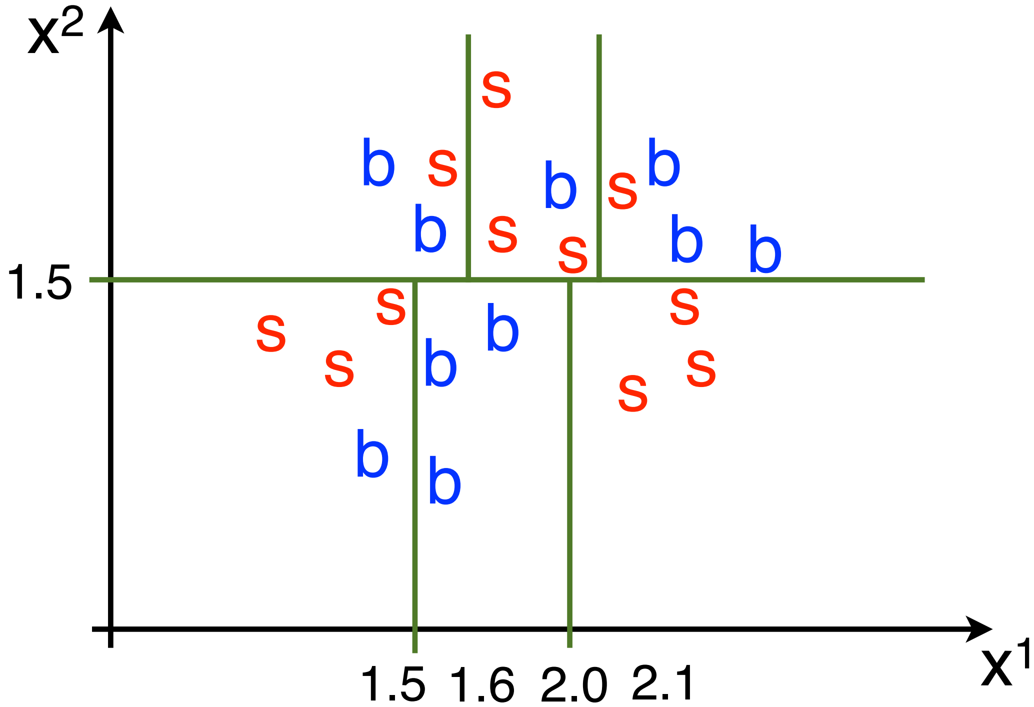
Decision trees: classification

Training sample $\in \Re^2$

$(x^1{}_1, x^2{}_1)$	classes
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	

Build a binary tree:



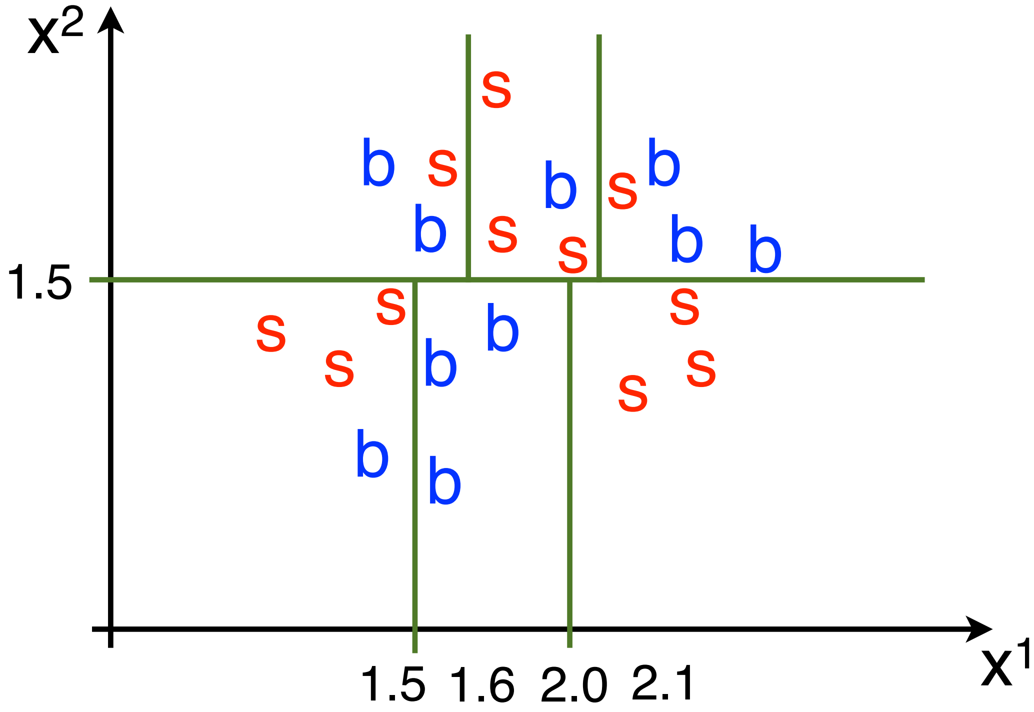


Strategy is to minimize the misclassification at each leaf

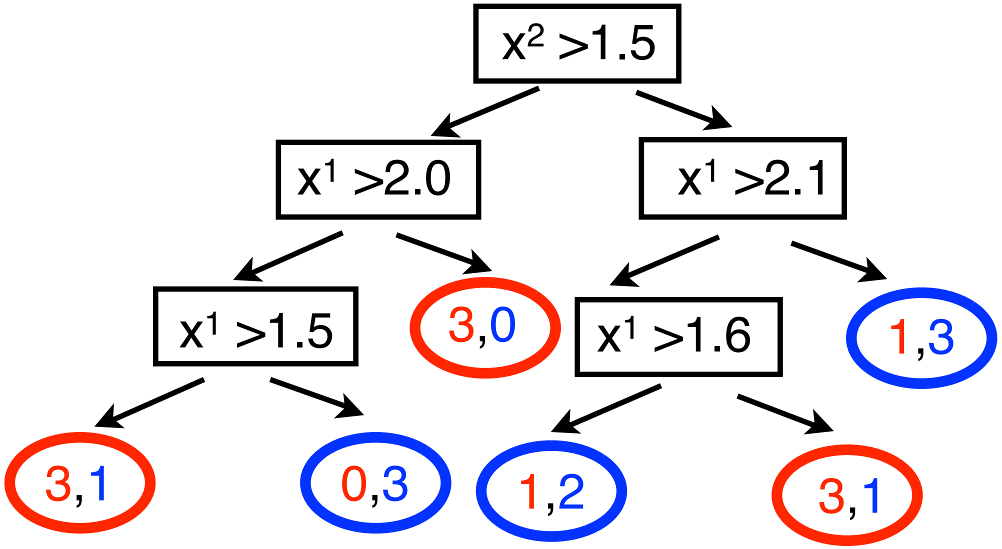
Decision trees: classification

Training sample $\in \Re^2$

	classes
$(x^1{}_1, x^2{}_1)$	
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	



Build a binary tree:



Strategy is to minimize the misclassification at each leaf

Now you have to choose how to classify the leaves: Majority vote

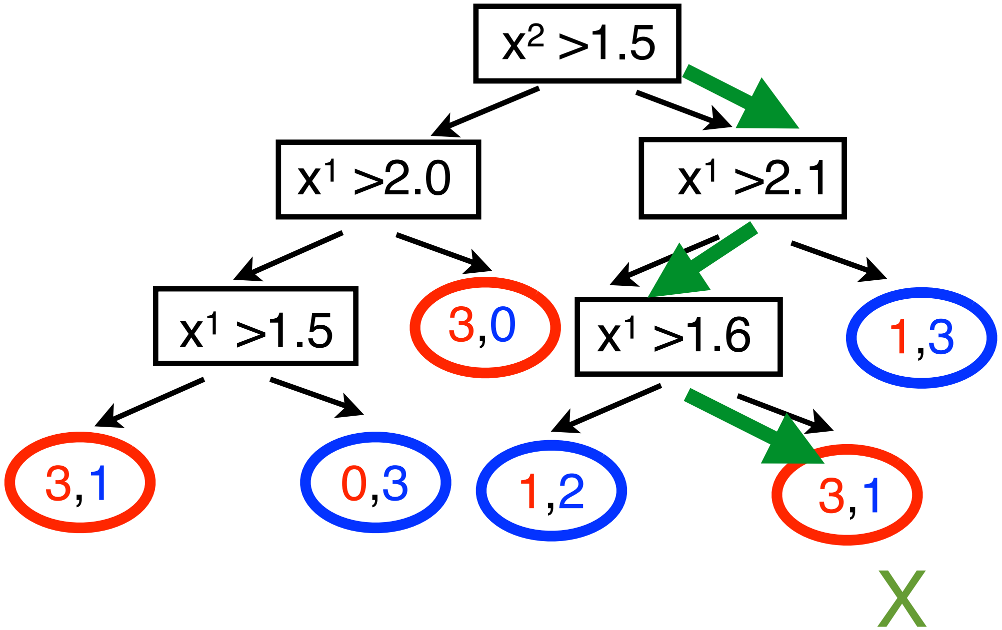
It's like writing a function piece wise constant over the plane

Decision trees: classification

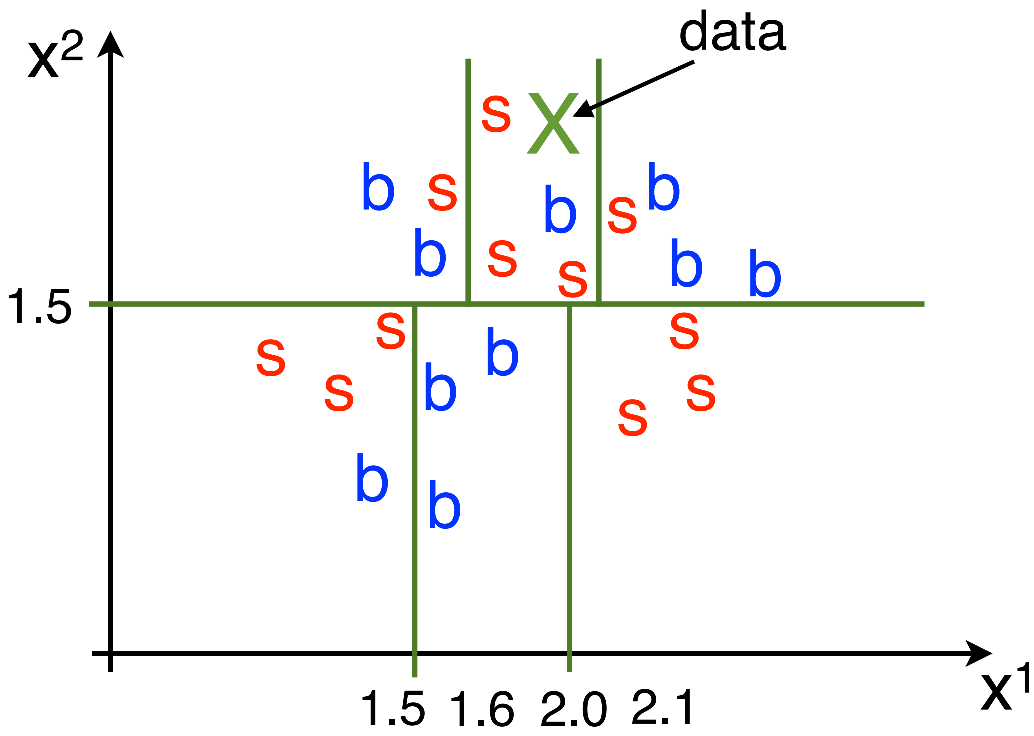
Training sample $\in \Re^2$

	classes
$(x^1{}_1, x^2{}_1)$	
...	
$(x^1{}_i, x^2{}_i)$	b
...	s
$(x^1{}_n, x^2{}_n)$	

Build a binary tree:



is classified as s



Strategy is to minimize the misclassification at each leaf

Decision trees: regression

Training sample $\in \Re$

(x_1,y_1)

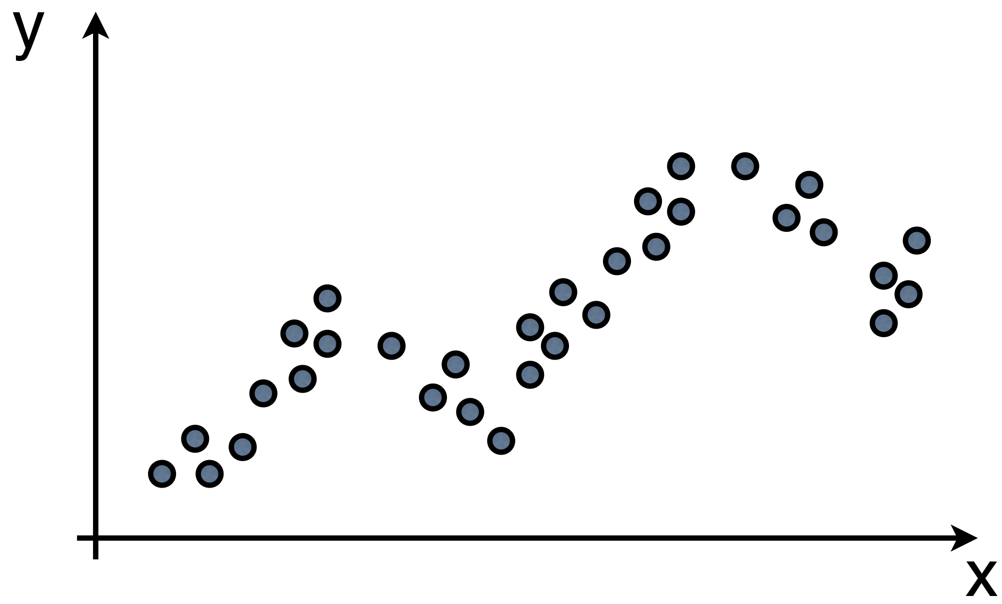
...

(x_i,y_i)

...

(x_n,y_n)

continuous target



Decision trees: regression

Training sample $\in\Re$ (trivial…)

(x_1,y_1)

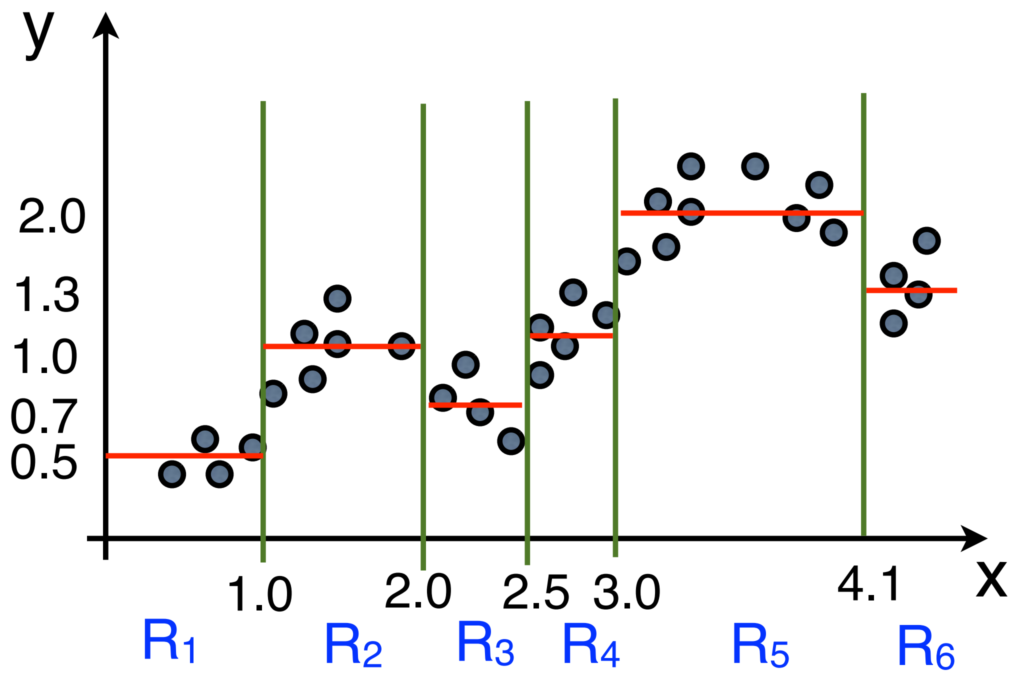
...

(x_i,y_i)

...

(x_n,y_n)

continuous target



Strategy is to minimize the error at each leaf

Repeat until every region contains a “minimum” number of points

Average of the points in each region

i.e. given x predict y

Decision trees: classification

Training sample $\in \Re$

(x_1,y_1)

...

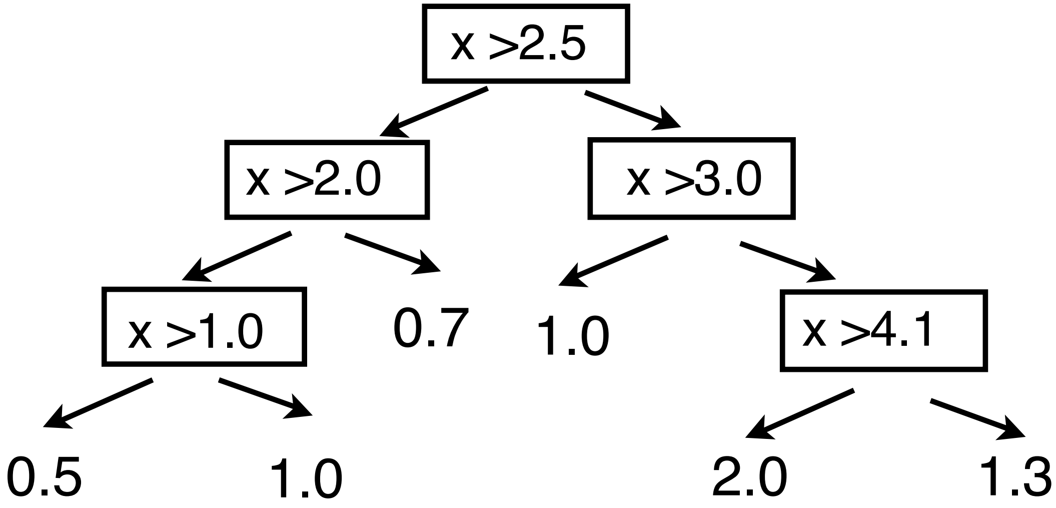
(x_i,y_i)

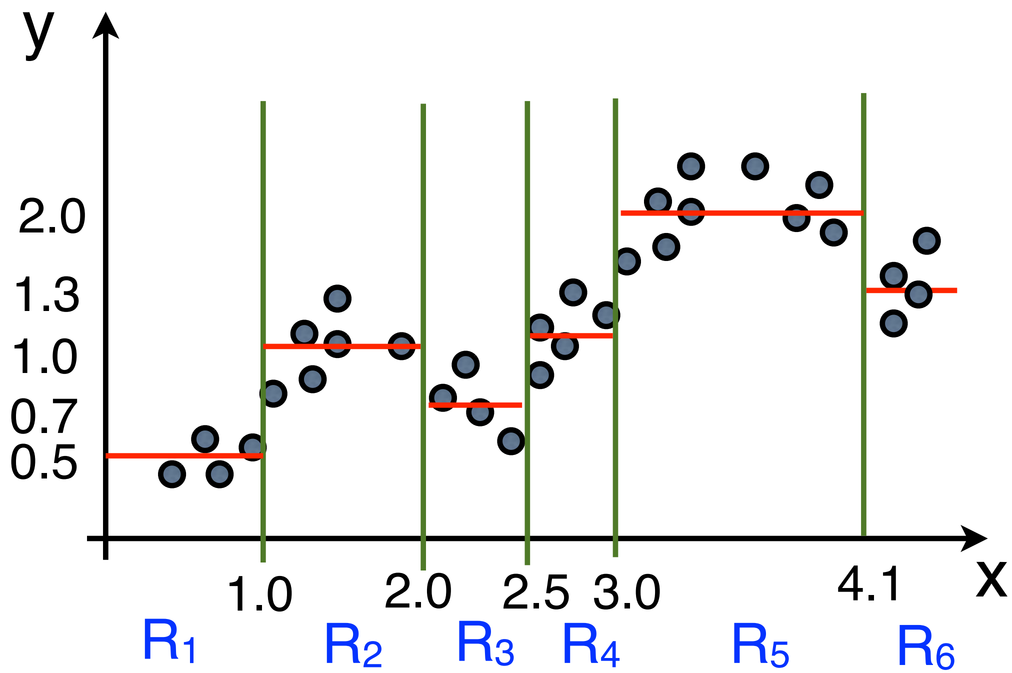
...

(x_n,y_n)

continuous target

Build a binary tree





Strategy is to minimize the error at each leaf

Average of the points in each region

i.e. given x predict y

Decision trees: classification

Training sample $\in \Re$

(x_1,y_1)

...

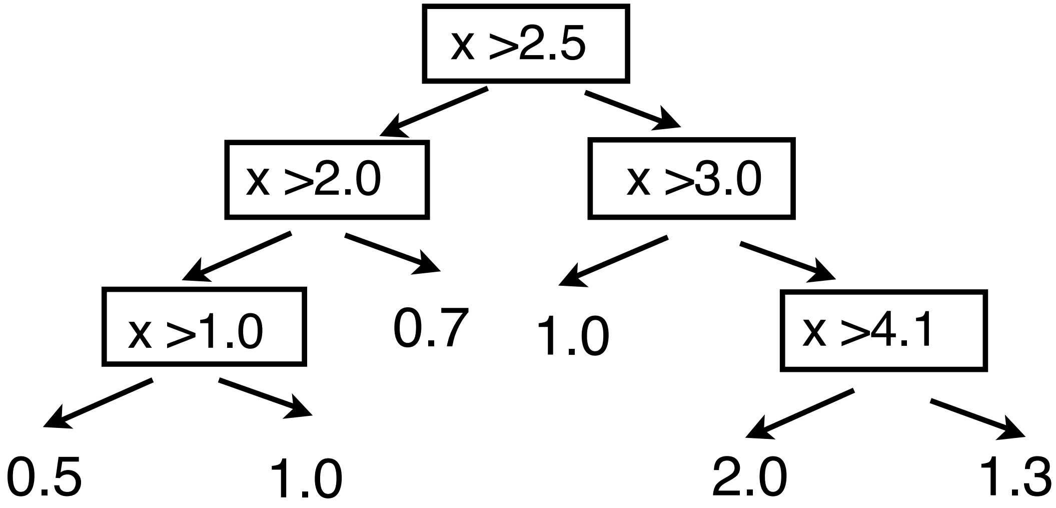
(x_i,y_i)

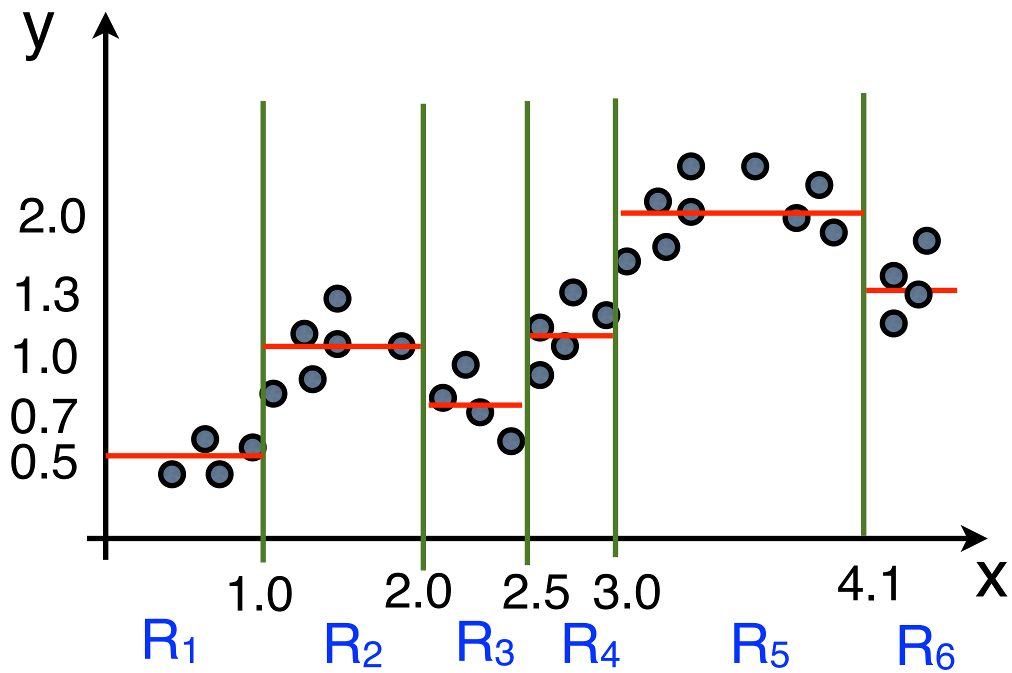
...

(x_n,y_n)

continuous target

Build a binary tree





Strategy is to minimize the error at each leaf

Average of the points in each region

It's like writing a function piece wise constant in $\Re$

Comments

The variables and the order are chosen on the base of separation.
So if you change the training sample you might get different trees.

Whatever variable is the most discriminating it will influence the rest of the tree

Decision trees tend to be very sensitive to statistical fluctuations of the training sample.
Decision trees are too unstable to be used safely.

Several aggregation techniques have been developed to improve the performance of the DT. (aggregating copies of the same tree)
The most commonly used is BOOSTING = BDT.

These techniques can be applied to classification and regression (and to any kind of classifier not only DT).

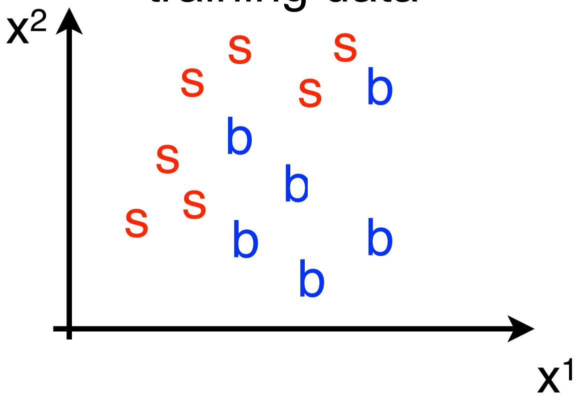
Boosting

Sequentially training a model learning from the errors of the previous ones.

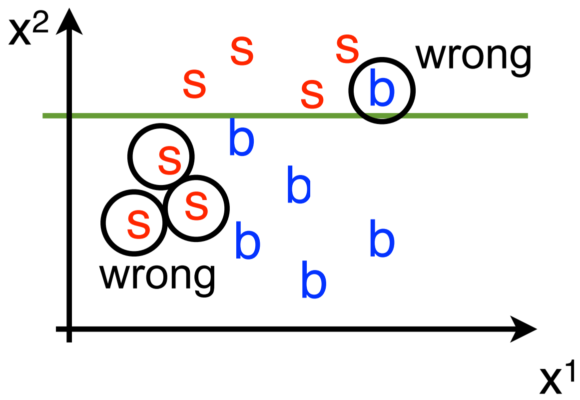
The idea is to create modifications that give smaller error rates than those of the preceding classifiers.

(for graphical reasons I use 2 variables and a single level DT, i.e. one cut)

training data



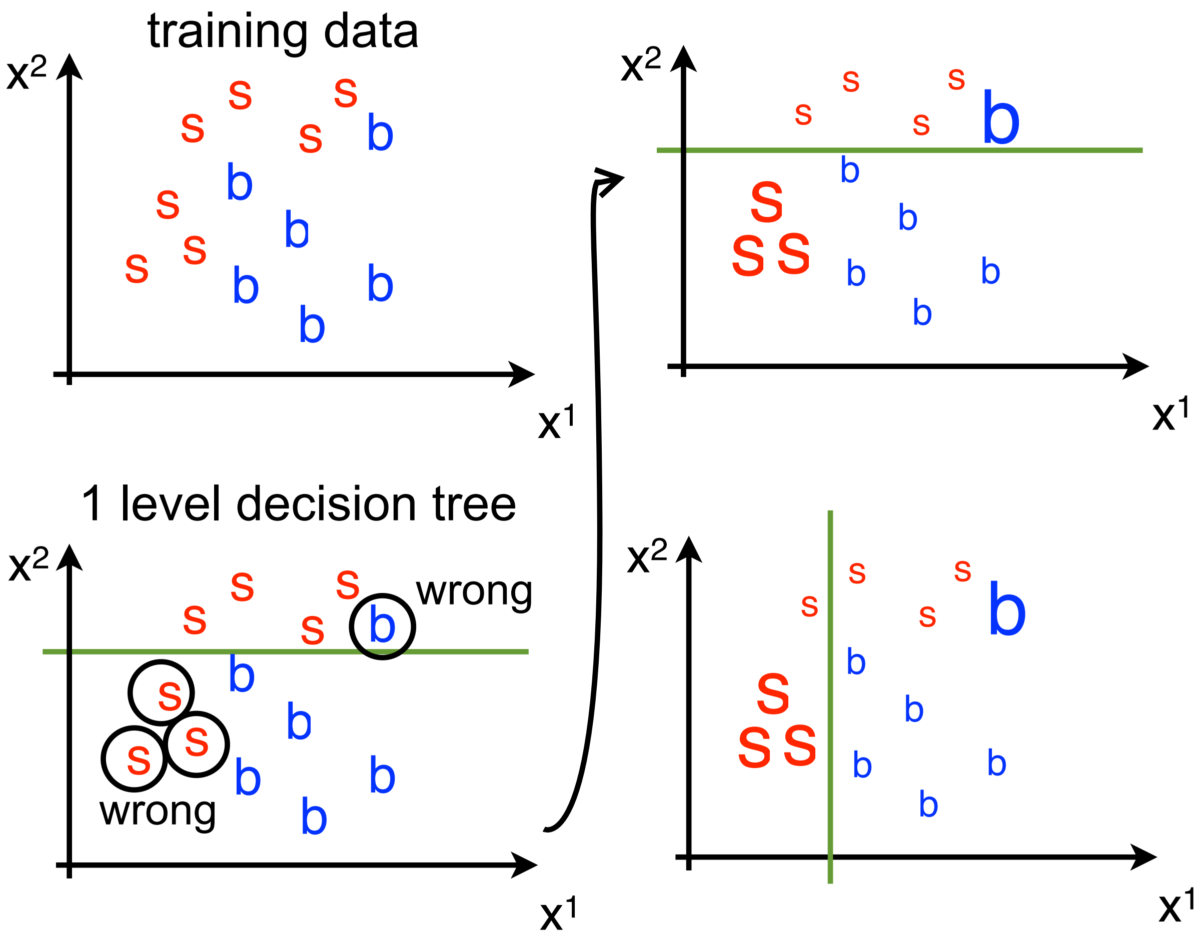
1 level decision tree



Focus on the 4 wrong ones

Boosting

Sequentially training a model learning from the errors of the previous ones. The idea is to create modifications that give smaller error rates than those of the preceding classifiers.



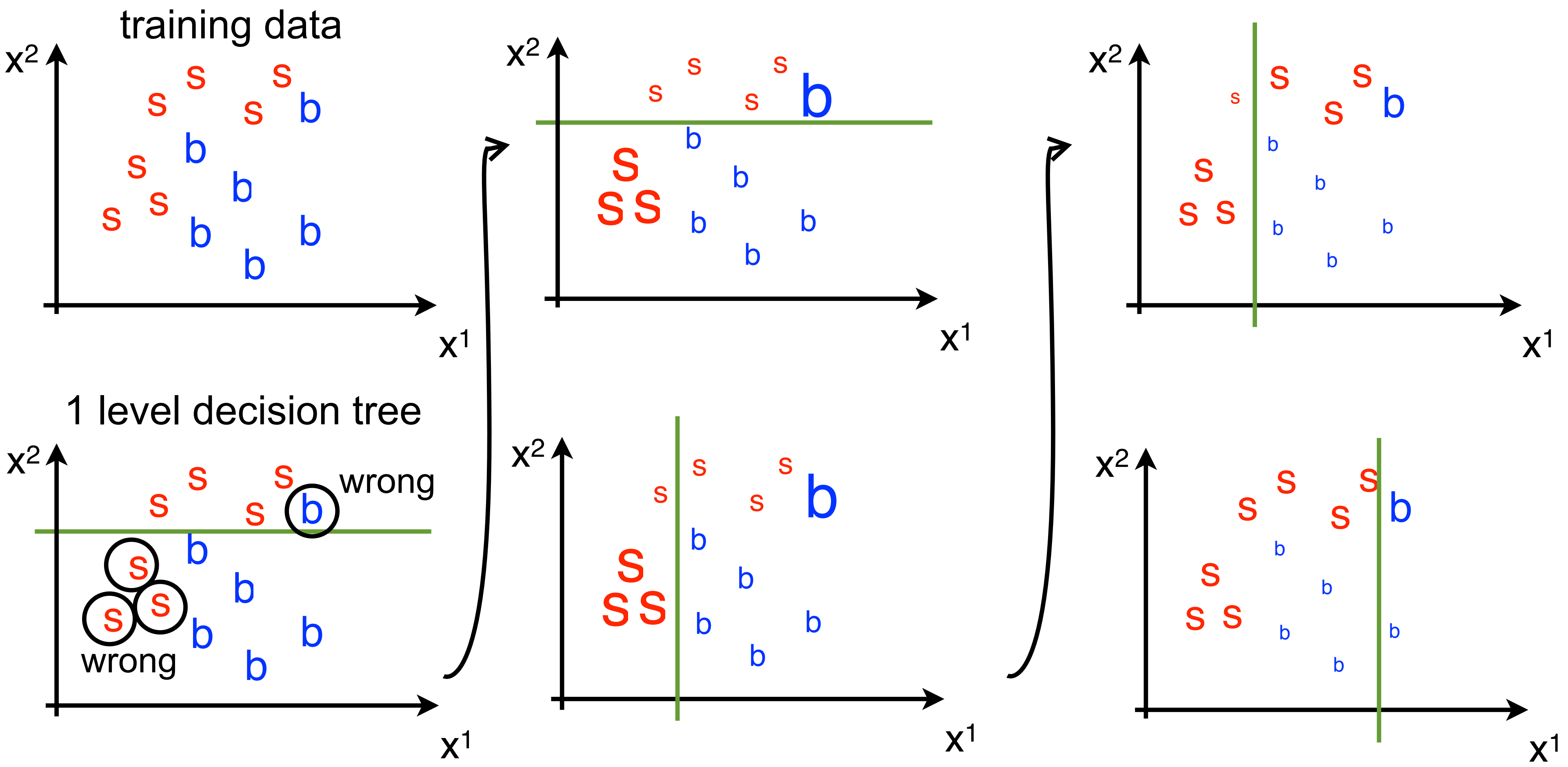
Right classification:
decrease weight

Wrong classification:
increase weight

it is more important to make this three “S” right than the other wrong

Boosting

Sequentially training a model learning from the errors of the previous ones. The idea is to create modifications that give smaller error rates than those of the preceding classifiers.

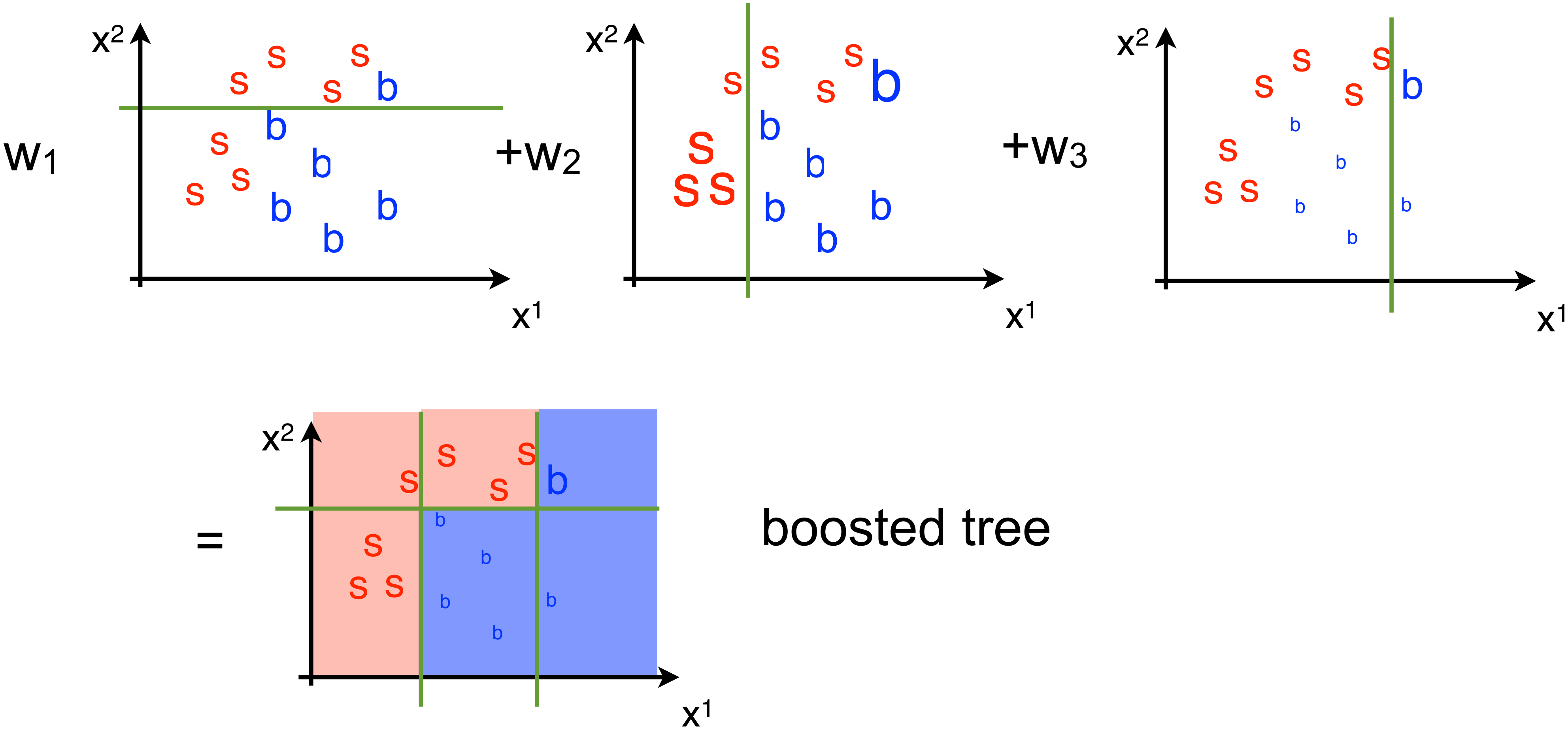


Boosting

In practice:

assign numerical values to the two classes: b = +1 s =-1

assign a weight w_i to each of the trees and sum them



Boosting: how to assign the weights

Adaptive Boost: Adaboost (one of many algorithms)

For i = 1.. N_{boost}

{

c(i) = train $(\vec{x}, \vec{y}, \vec{w})$

$\hat{\vec{y}}$ = predict (c(i), $\vec{x}$)

Compute the vector of errors

$$e = \vec{w} * (\vec{y} == \hat{\vec{y}})$$

Set $\alpha_i = \frac{1}{2} \log \left(\frac{1-e}{e} \right)$

$$\vec{w} = \vec{w} e^{-\alpha_i (\vec{y}_i \cdot \hat{\vec{y}}_i)}$$

$$\vec{w} = \vec{w} / \sum (\vec{w})$$

}

final classifier/tree = $\sum_i \alpha_i \, predict(c(i), x)$

c(i) = classifier/tree (i)
$\vec{x}$ = vector of variables in
$\vec{y}$ = vector of class/target out
$\vec{w}$ = vector of weights

initially set all weights to 1, then evolve them

e = scalar error = vector of weights * vector of 0s and 1s
correct/wrong

α at the step i

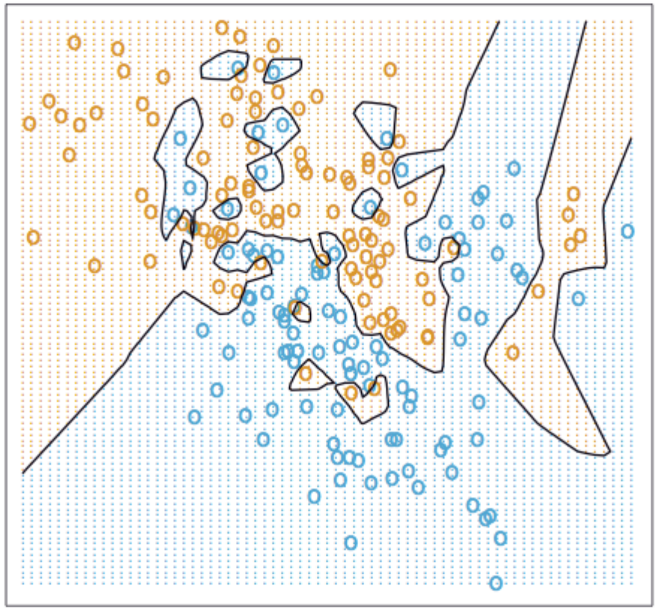
(true,predict)

correct (s,s) or (b,b) ⇒ "+ sign" down-weighted
wrong (s,b) or (b,s) ⇒ "- sign" up-weighted

normalize by the sum of all weights

Overtraining:

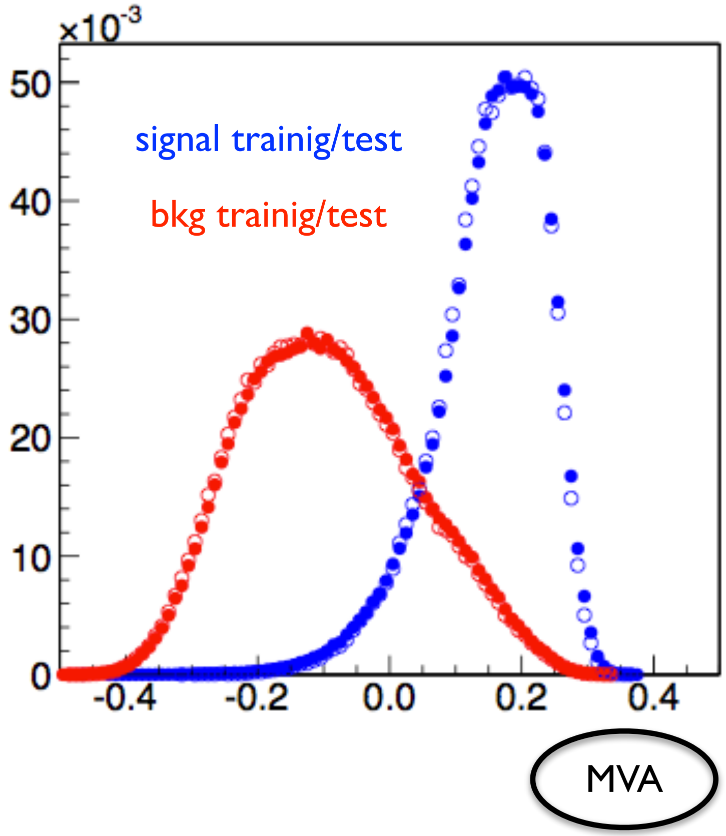
it is easy to control using tuning the number of events in the final leaves



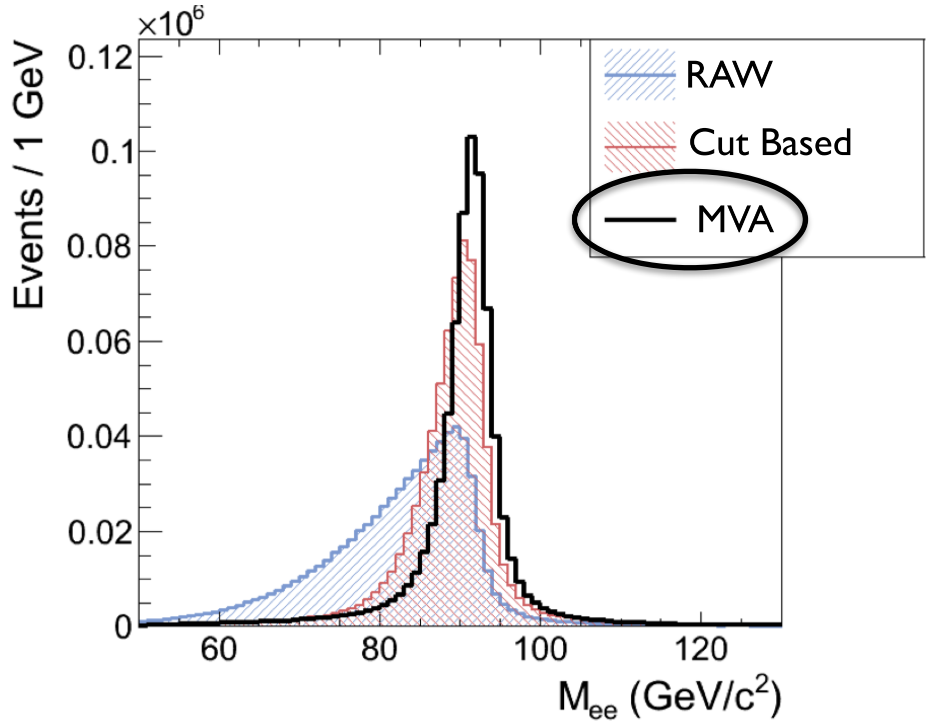
Correlated variables:

adding several variables will not degrade the performance of the BDT because the less discriminating will be automatically de-weighted (Gini index)

Classification



Regression



Statistics tools

Two main classes of tools used:

- parameters estimation: maximum likelihood fits
- hypothesis testing

$$L = \prod_{i=1}^{N_{evt}} \mathcal{L}_i$$

L(data|parameters)

Hypothesis testing

Formulate an hypothesis, test the data against the hypothesis then accept or reject.
An hypothesis is a statement that can be proved experimentally:
eg: the data are not described by the background only model

Null hypothesis H_0 is defined to be the hypothesis under consideration (in searches this is the background only hp). A statement on H_0 (often) involves an Alternative hypothesis H_1 (in searches it's the signal + background hp).

To quantify the agreement between the observed data and a given hypothesis one construct a function of the measured variables (**x**) and the given hypothesis H

test statistics = q(**x**|H)

The test statistics will be distributed differently depending on the data and the HP.

To build the test statistics distribution P(q(**x**|H)) typically we generate pseudo-data **x** (toy MC).

Excess of events

The test statistics chosen for the LHC is based on a profile likelihood ratio.
To quantify an excess of events we use:

background

nuisances describing the systematic uncertainties

$$q_0 = -2\ln\frac{\mathcal{L}(\text{data}\,|\,b,\hat{\theta}_0)}{\mathcal{L}(\text{data}\,|\,\hat{\mu}\cdot s+b,\hat{\theta})}$$

profile likelihood
nuisances are “profiled” (fit on data)

It is a function of $\hat{\mu}$

signal strength modifier

signal expected from SM

$\hat{\theta}_0$ maximizes the likelihood at the numerator (bkg only hp)

$\hat{\mu}$ and $\hat{\theta}$ maximizes the likelihood at the denominator (sig+bkg hp)

Define local *p*-value as: $p_0 = \mathrm{P}\left(q_0 \geq q_0^{\text{data}}\,\middle|\,b\right)$

and we transform it into a local significance *z* on the one-sided tail Gaussian

$$p_0 = \int_z^{+\infty}\frac{1}{\sqrt{2\pi}}\exp(-x^2/2)\,dx$$

Signal model parameters

Take any parameter "a" that has an influence on the signal model and define the test statistics:

$$q(a) = -2\,\Delta \ln \mathcal{L} = -2 \ln \frac{\mathcal{L}(\text{data}\,|\,s(a)+b,\,\hat{\theta}_a)}{\mathcal{L}(\text{data}\,|\,s(\hat{a})+b,\,\hat{\theta})}$$

$\hat{a}$ is the best fit for the parameter "a" and the nuisance are profiled as before

The 68% and 95% CL intervals are defined by $q(a_i) = 1.00$ $q(a_i) = 3.84$

and the two dimensional contours by $q(a_i, a_j) = 2.30$ $q(a_i, a_j) = 6.99$

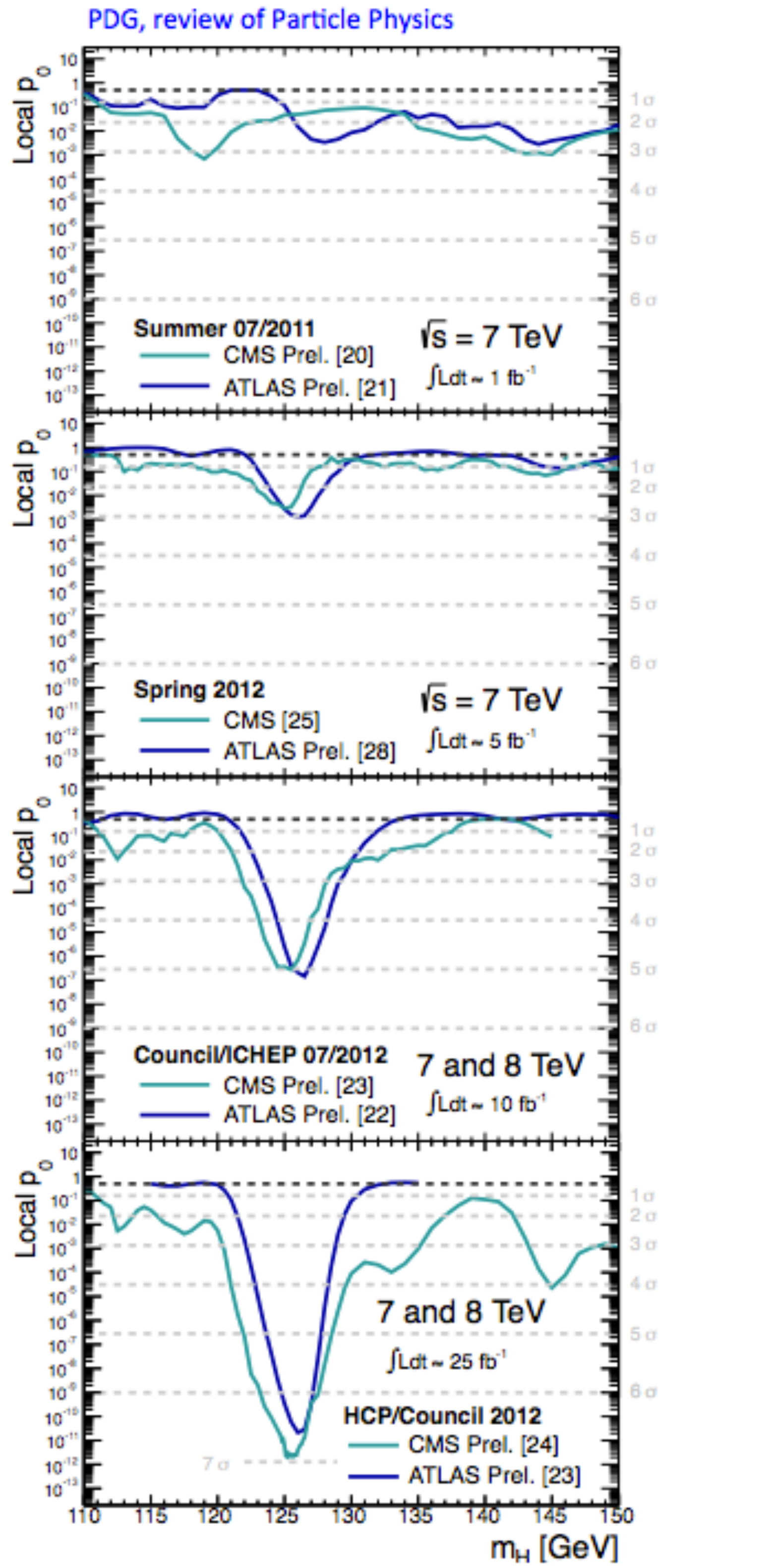
excitement (?)

big excitement

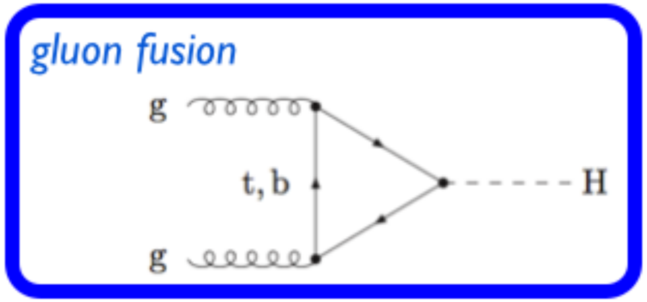
discovery

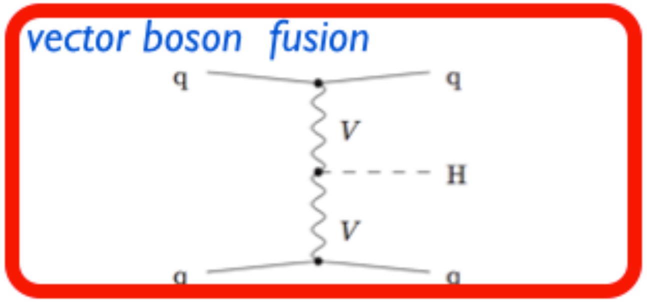
today

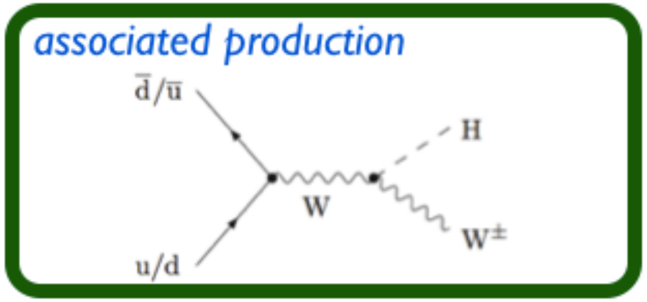
Higgs

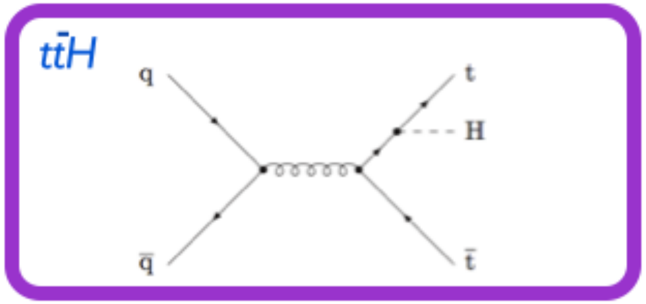


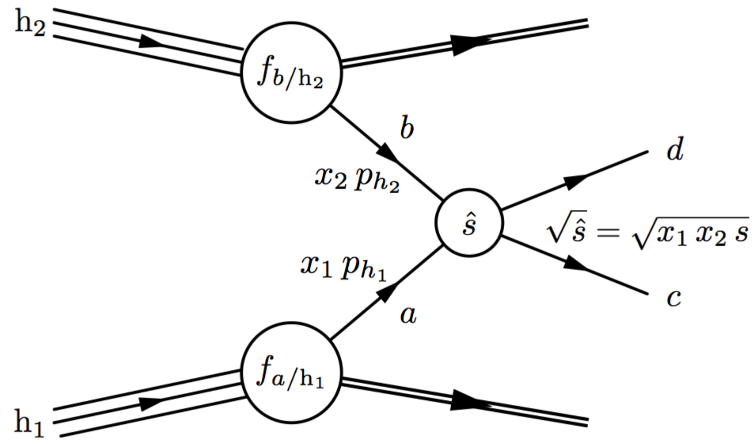
Production modes



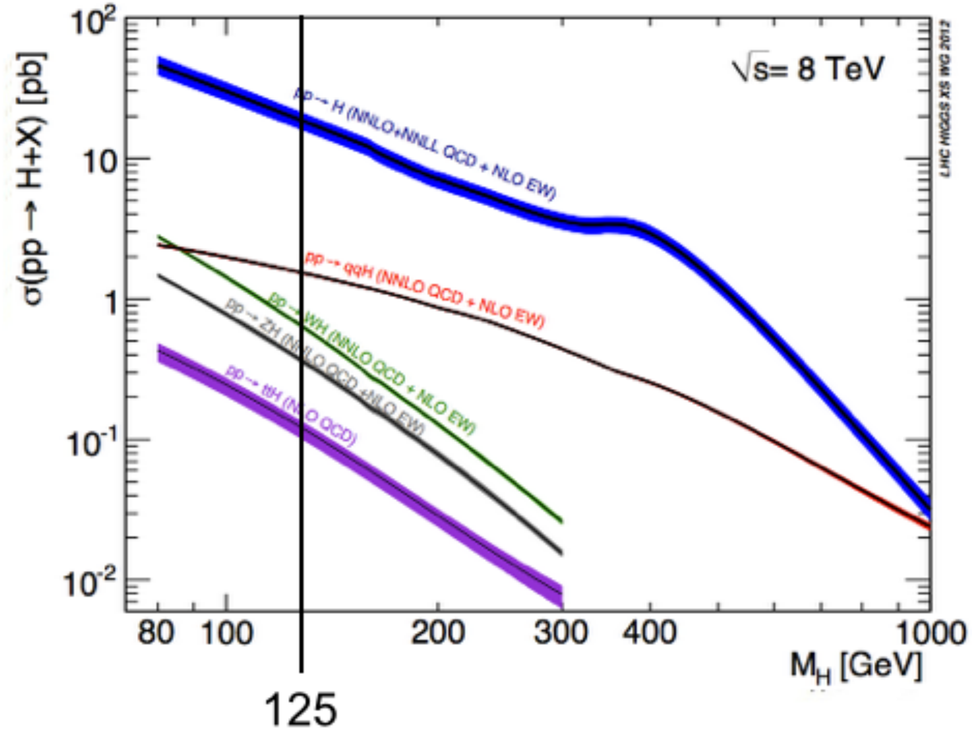




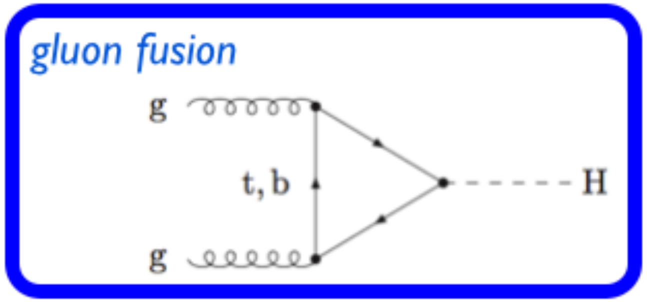


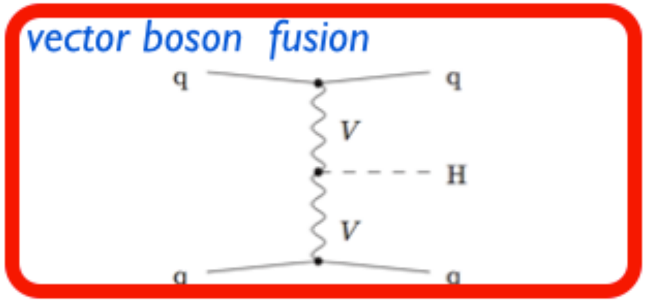


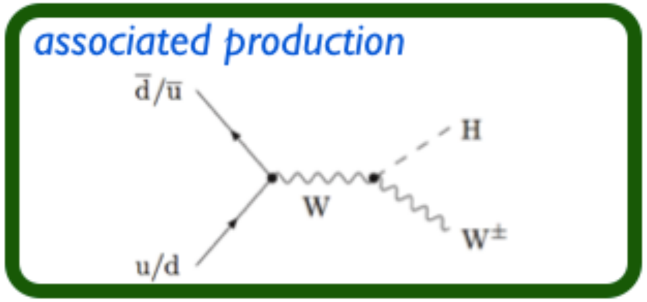
$$d\sigma(h_1 h_2 \to cd) = \int_0^1 dx_1 dx_2 \sum_{a,b} f_{a/h_1}(x_1, \mu_F^2) f_{b/h_2}(x_2, \mu_F^2) d\hat{\sigma}^{(ab\to cd)}(Q^2, \mu_F^2)$$

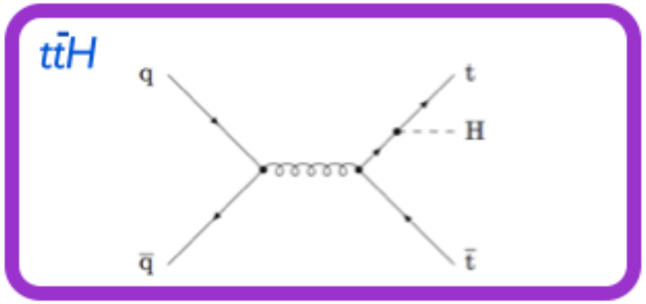


Production modes









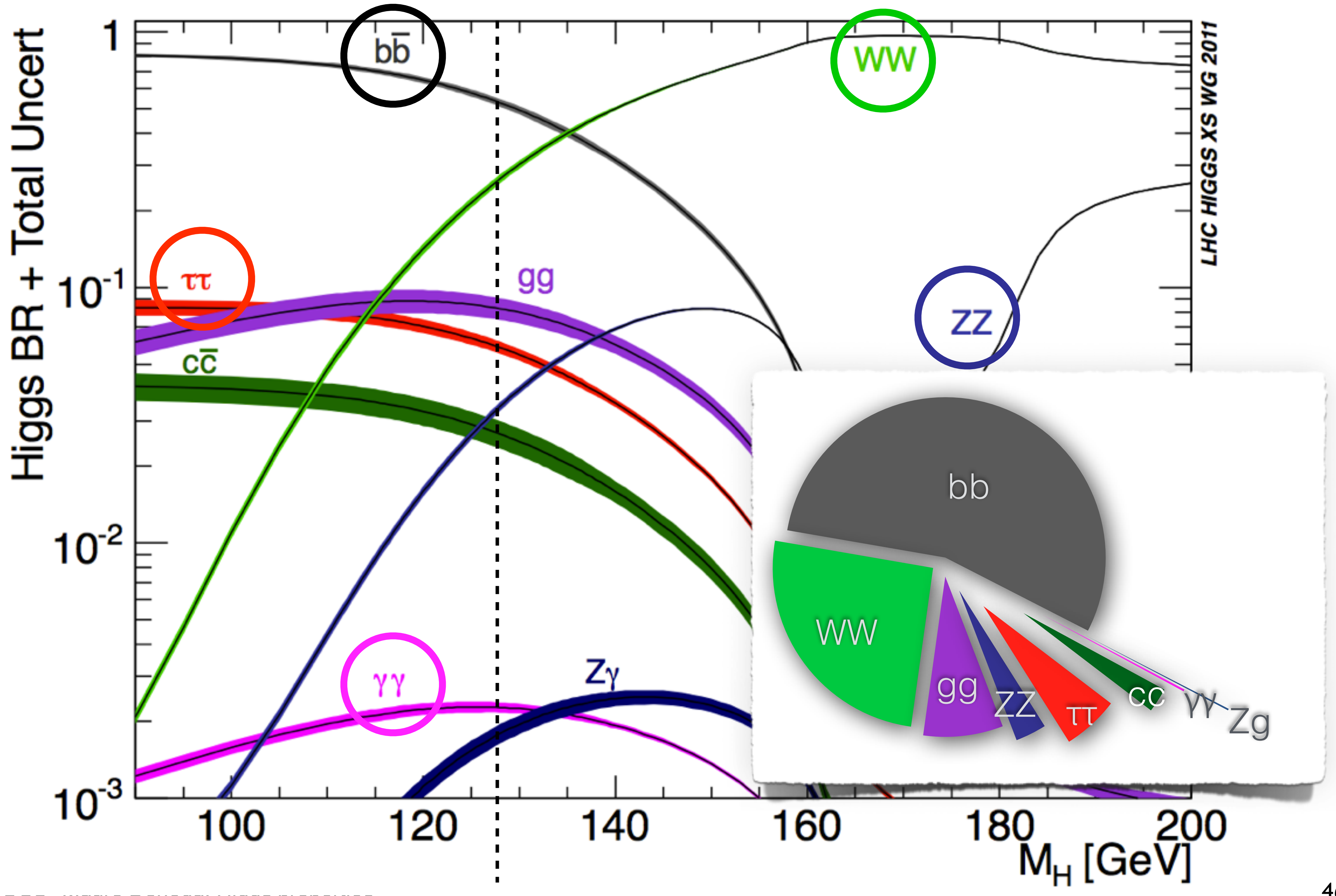
ggF:
- largest cross section
- no extra jet activity

VBF:
- harder pT spectrum
- two high eta jets (large rapidity gap no colorflow)

VH:
- tag on the presence of the W/Z
- pT spectrum similar to VBF

ttH:
- busy environment influence the isolation
- tag on the tops (high pT leptons, b-jets, #jets)

Decay modes



Higgs BR + Total Uncert
1
10^{-1}
10^{-2}
10^{-3}
100
120
140
160
180
200
M_H [GeV]
LHC HIGGS XS WG 2011
$b\bar{b}$
WW
$\tau\tau$
gg
$c\bar{c}$
ZZ
$\gamma\gamma$
$Z\gamma$
bb
WW
gg
ZZ
ττ
cc
γγ
Zg

Dissect one analysis: $H \rightarrow \gamma\gamma$

Naive analysis sequence:

1- Choose the signature your looking for

2- Setup a trigger such that your detector will record it

3- Identify the backgrounds sources in your data sample

4- Build a way to discriminate signal / background

5- Estimate the signal component / the backgrounds left in your signal region

6- Assess the significance of your signal:
- a- set limits / significance of a signal - HP testing
- b- HP testing on the properties of the signal ($0^+/2^+$)
- c- measure the properties of your signal

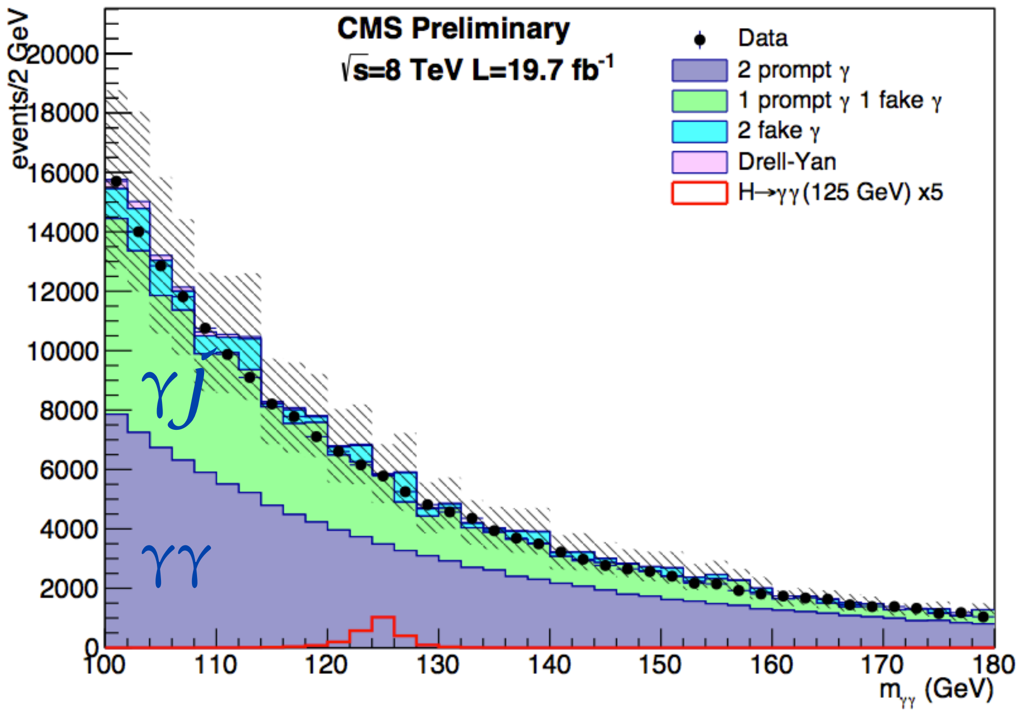
Dissect one analysis: H→$\gamma\gamma$

Narrow resonance on a large steeply falling background

$$m_{\gamma\gamma} = \sqrt{2E_1E_2(1-\cos\alpha)}$$

Analysis steps:

- select high pT isolated $\gamma\gamma$
- get the correct vertex
- get the best energy resolutions (see mass)
- photon Identification (gamma/jet)
- events classification
- model the background
- extract the signal
- measure properties



H→γγ diphoton vertex

Diphoton vertex: no ionisation from the two photons in the tracker.
Use transverse quantities to train a BDT classifier to select the right vertex

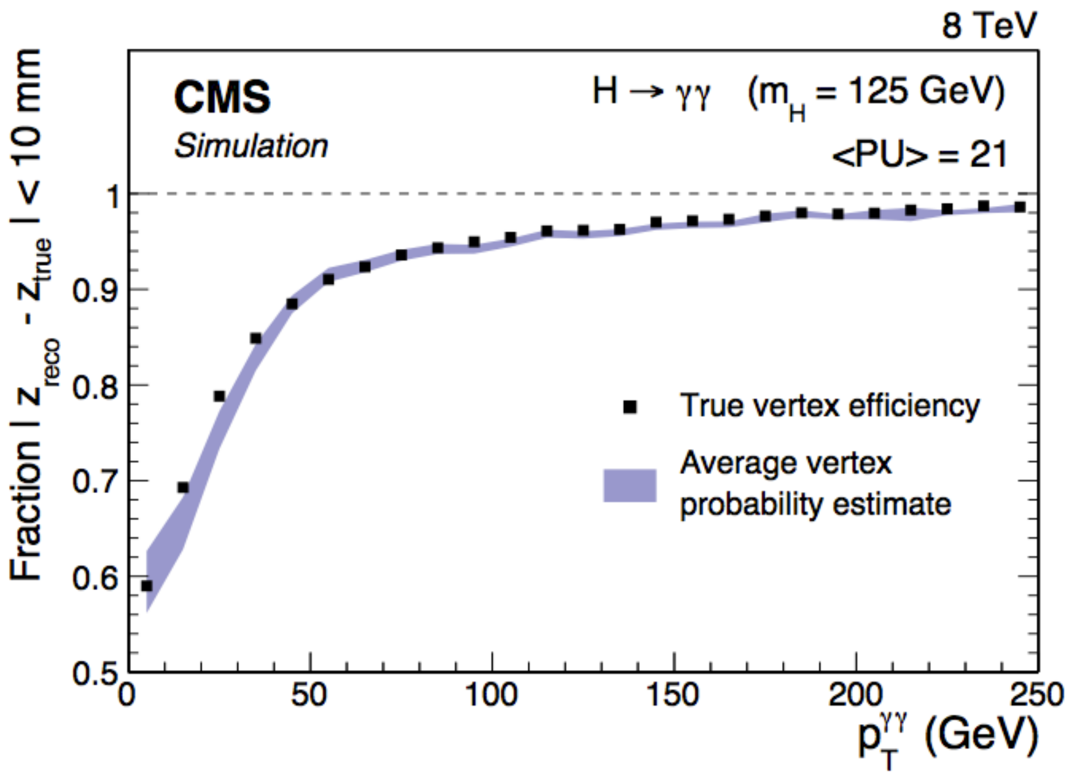
$$m_{\gamma\gamma} = \sqrt{2E_1E_2(1-\cos\alpha)}$$

$$\sum \vec{p}_\mathrm{T}^{\,2}$$

$$-\sum(\vec{p}_\mathrm{T} \cdot \frac{\vec{p}_\mathrm{T}^{\,\gamma\gamma}}{|\vec{p}_\mathrm{T}^{\,\gamma\gamma}|}), \text{ and}$$

$$(|\sum \vec{p}_\mathrm{T}| - |\vec{p}_\mathrm{T}^{\,\gamma\gamma}|)/(|\sum \vec{p}_\mathrm{T}| + |\vec{p}_\mathrm{T}^{\,\gamma\gamma}|).$$

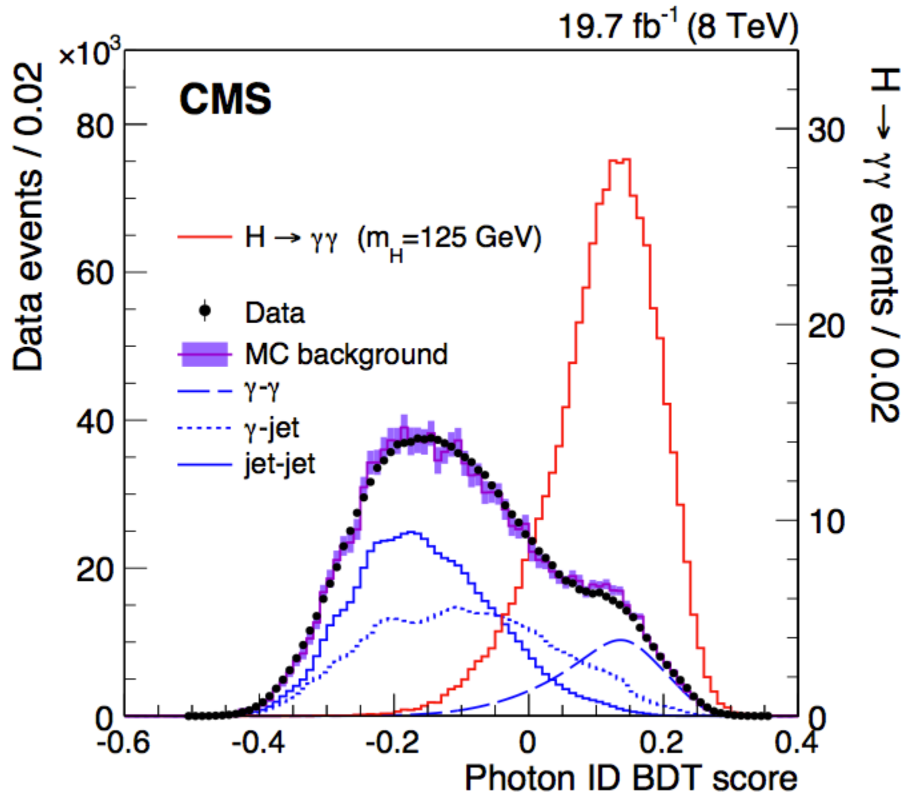
If you get the vertex close to <1cm to the true one, the effect of the wrong vertex is subdominant w.r.t. to the energy resolution on the mass resolution



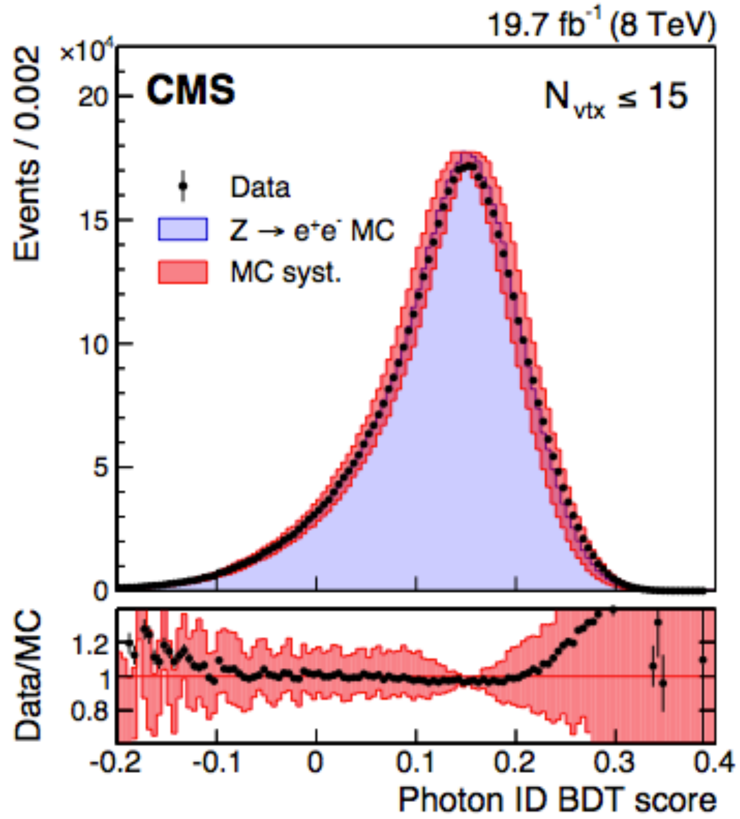
H→γγ photon identification

A jet where the pT fluctuates to a single neutral hadron can fake a photon

Photon identification using a BDT:
(use shower shapes, isolation, rho, eta, E)



Validation on Z→ee events



Hairy problem: MVA systematics

$H \rightarrow \gamma\gamma$ event classification

Select events in a region $100 < m_{\gamma\gamma} < 180$

$pT(\gamma_1) > m_{\gamma\gamma}/3$; $pT(\gamma_1) > m_{\gamma\gamma}/4$ (don't want to feed any mass information to the classifier !)

photonID > -0.2 (99% efficient , remove 1/4 of the bkg)

Start by selecting the events tagging specific production mechanisms:

ttH lepton tag	1 cat *	At least 1 b-tagged jet +1 lepton
VH tight lepton	1 cat	2 same flavour leptons consistent with Z OR 1 lepton and MET consistent with W
VH loose lepton	1 cat	One lepton
VBF dijet tag	3 (2) cats	2 jets. Categorised with combined dijet-diphoton BDT
VH MET tag	1 cat	MET > 70 GeV
ttH multijet tag	1 cat *	At least 1 b-tagged jet + 4 more jets
VH dijet tag	1 cat	Jet pair consistent with W or Z
Untagged	5 (4) cats	Remainder classified with diphoton BDT

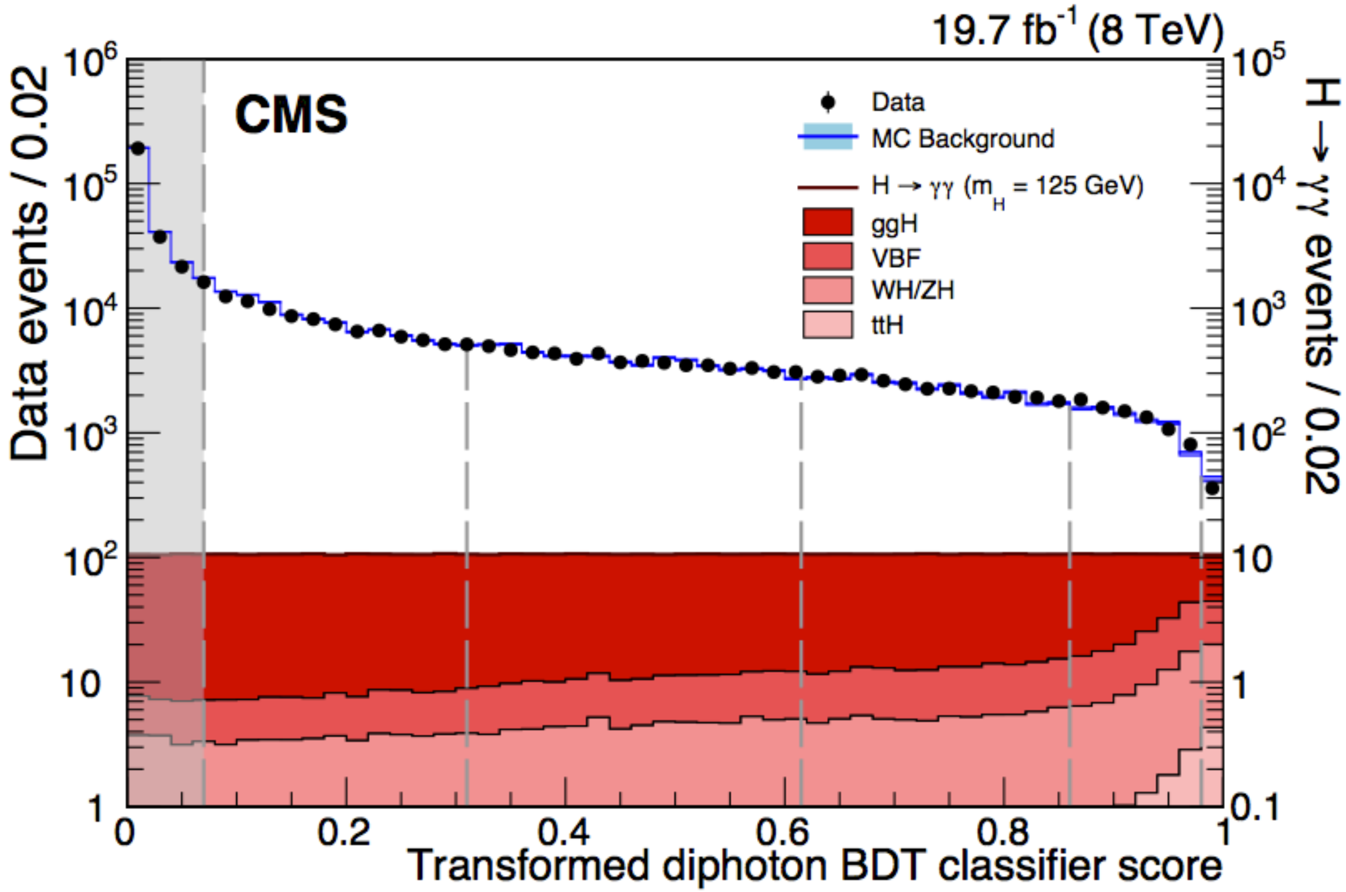
H→γγ event classification

Built a BDT classifier to give a high score to events with:

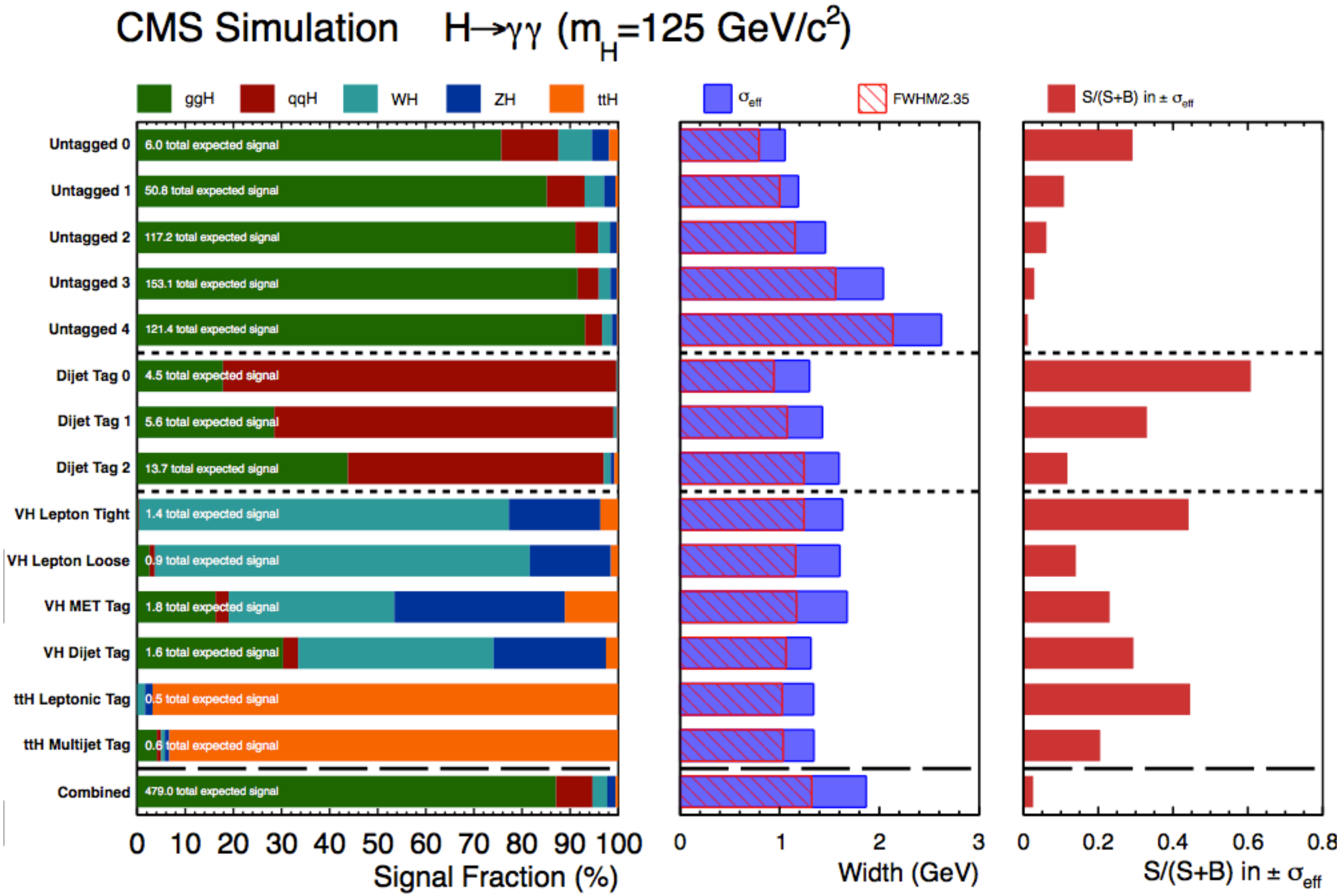
good $m_{\gamma\gamma}$ resolution

high probability to be a signal (kinematics, photonID, etc…)

be mass INDEPENDENT (should not look for events based on their mass)

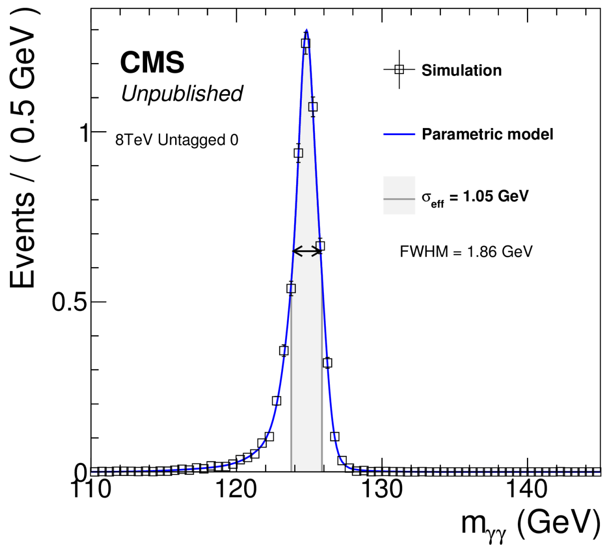


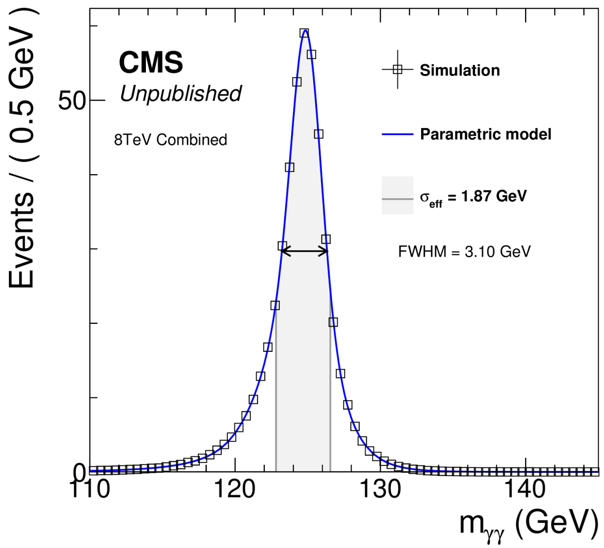
H→γγ signal composition

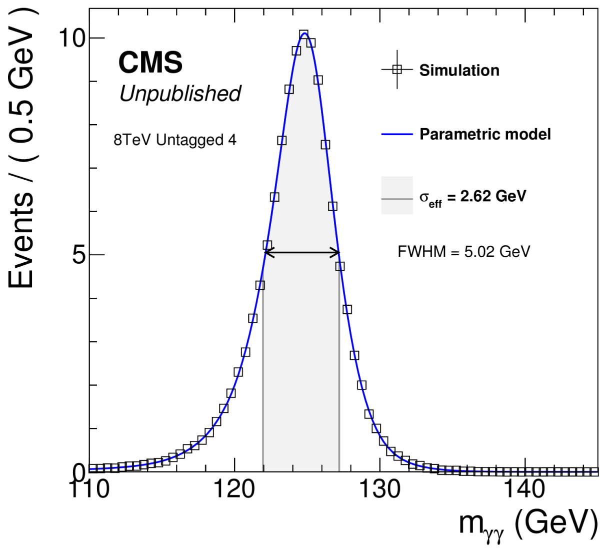


H→γγ signal/background model

For each category produce a signal model taking into account the proportion of different production mechanisms right/wrong vertex assignments (model = sum of gaussians)







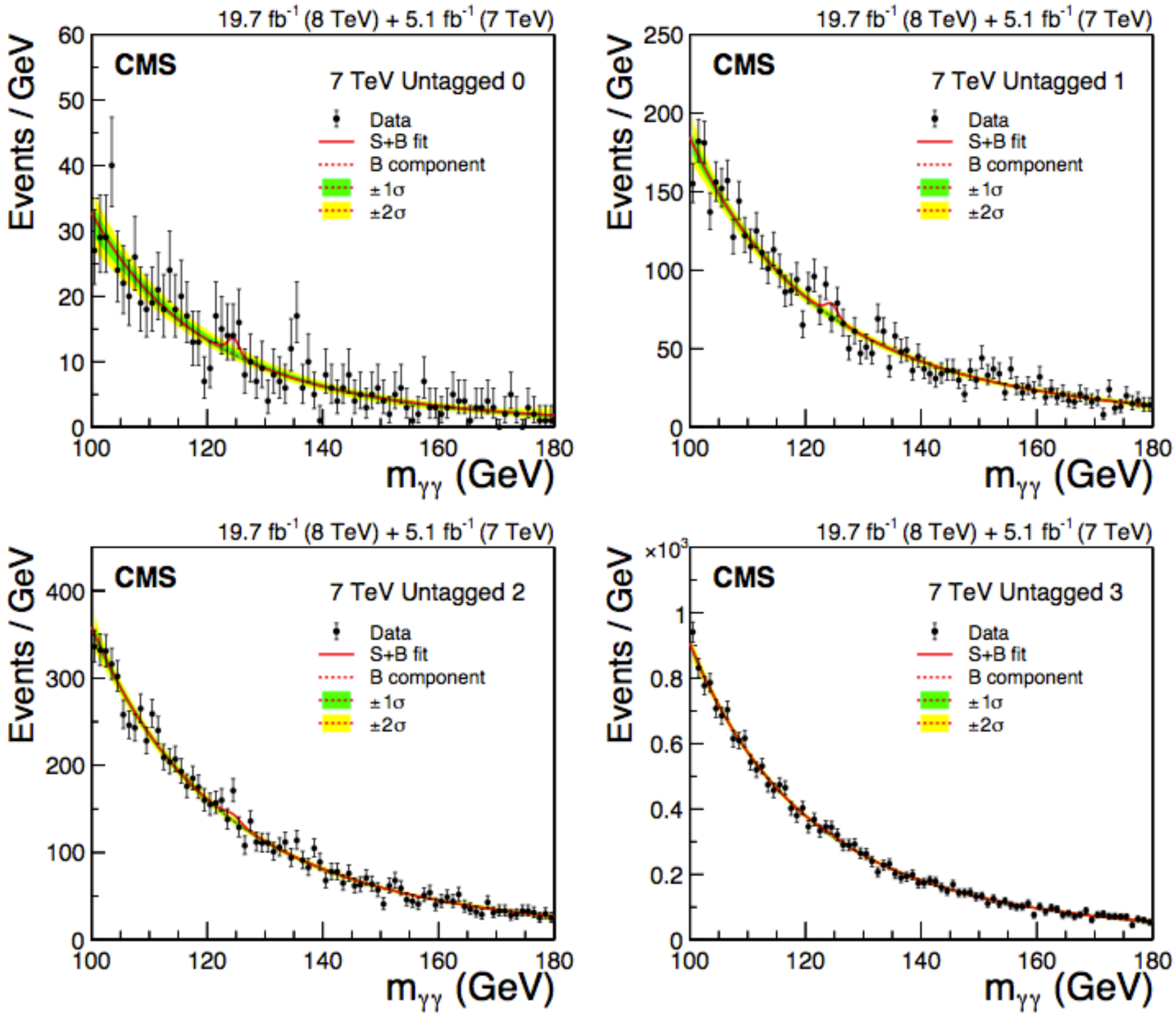
Background

CMS: discrete profiling method;
the systematics uncertainty on the bkg goes into the statistical error

ATLAS: gets the functional forms fitting on MC, then throws toys and look for one function that fit them all
Systematics uncertainty as the maximum bias the largest absolute signal component fitted anywhere in [110-150] GeV with the background samples above

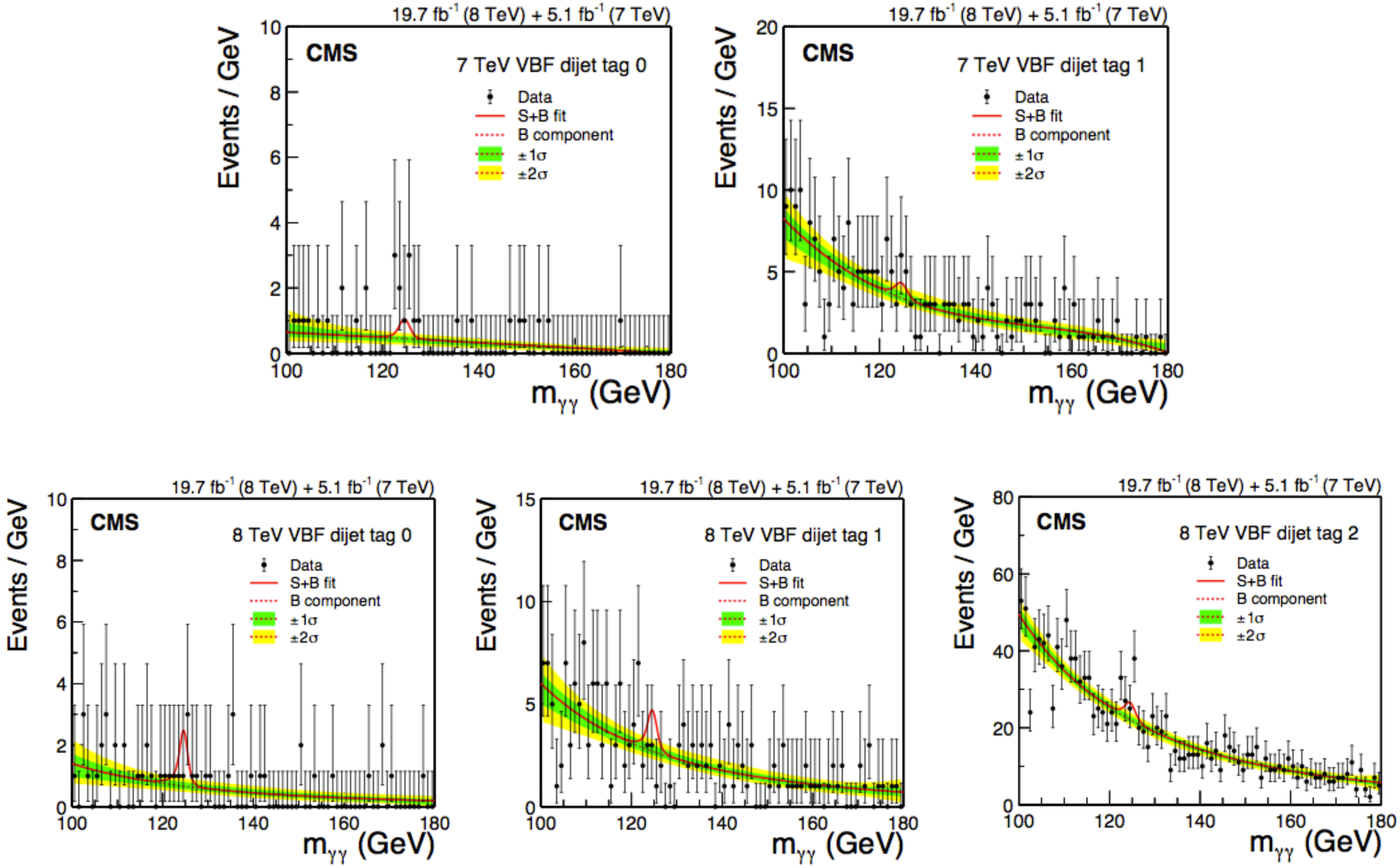
H→γγ fits

Fit the signal on all categories **simultaneously**

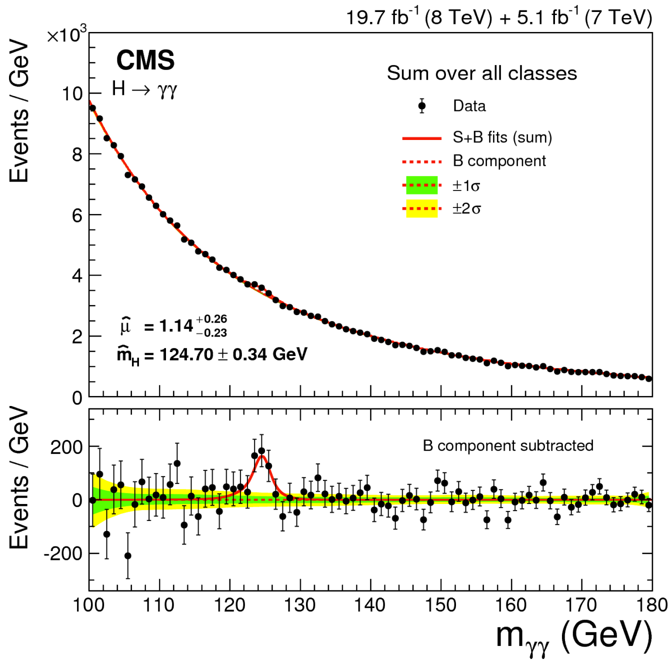


H→γγ fits

Fit the signal on all categories **simultaneously**

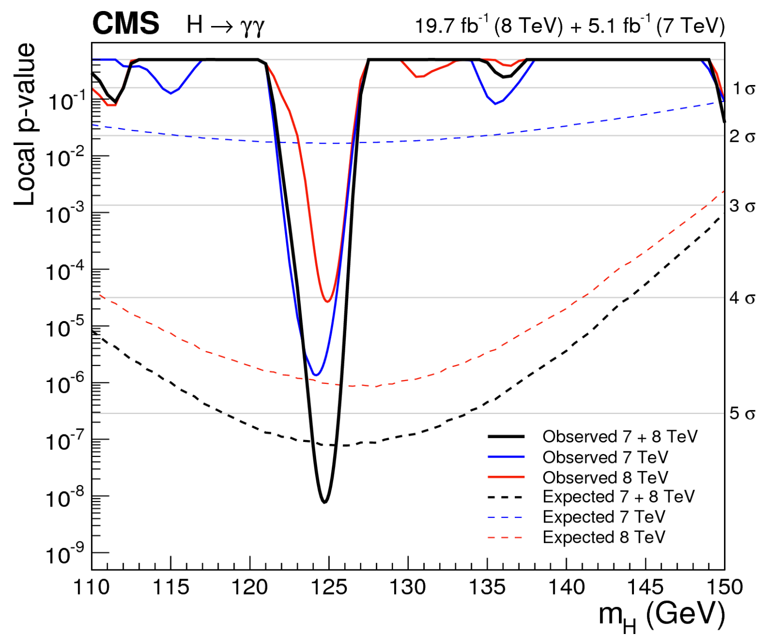


H→γγ results



mH IS NOT mγγ

At each test mass compute what you need



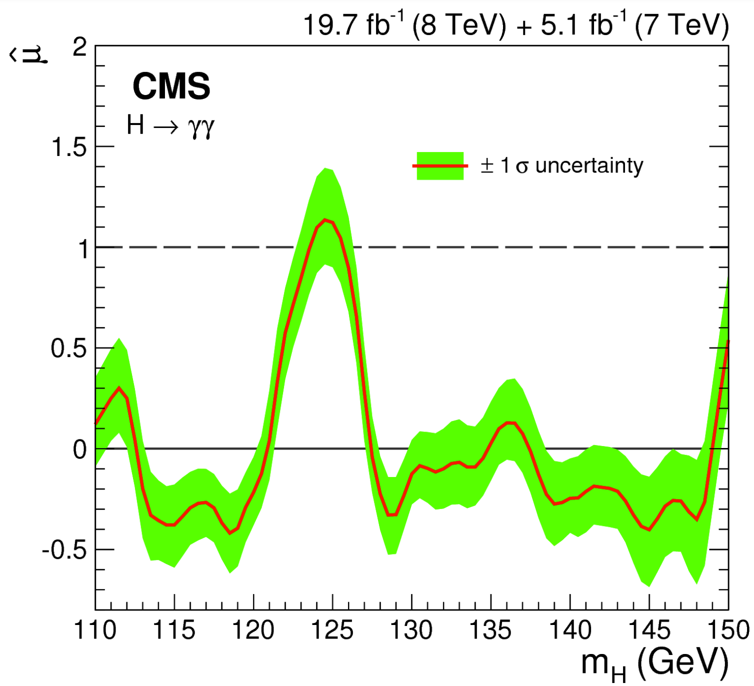
Significance @ 124.7 GeV

Expected 5.2 σ

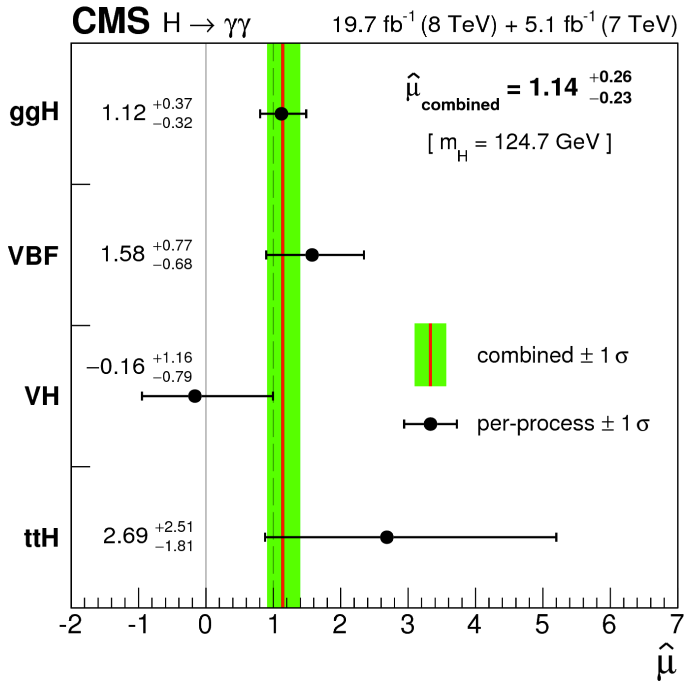
Observed 5.7 σ

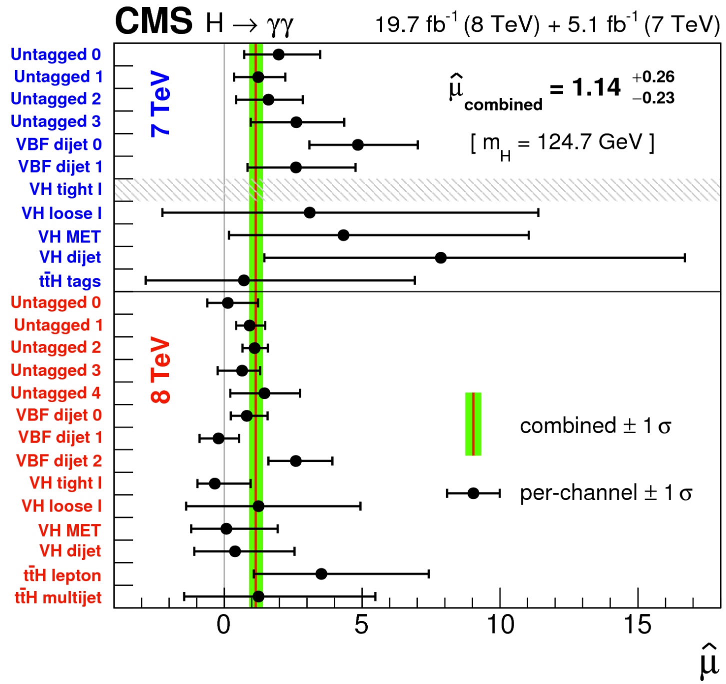
$$\hat{\mu} = 1.14^{+0.26}_{-0.23}$$

$$1.14 \pm 0.21\,(\text{stat})\ ^{+0.09}_{-0.05}\,(\text{syst})\ ^{+0.13}_{-0.09}\,(\text{theo})$$



H→γγ results





Source of uncertainty	Uncertainty in $\hat{\mu}$
Production cross sect. and branching frac.	0.11
Shower shape modelling	0.06
Energy scale and resolution	0.02
Other	0.04
All syst. uncert. in the signal model	0.13
Statistical	0.21
Total	0.25