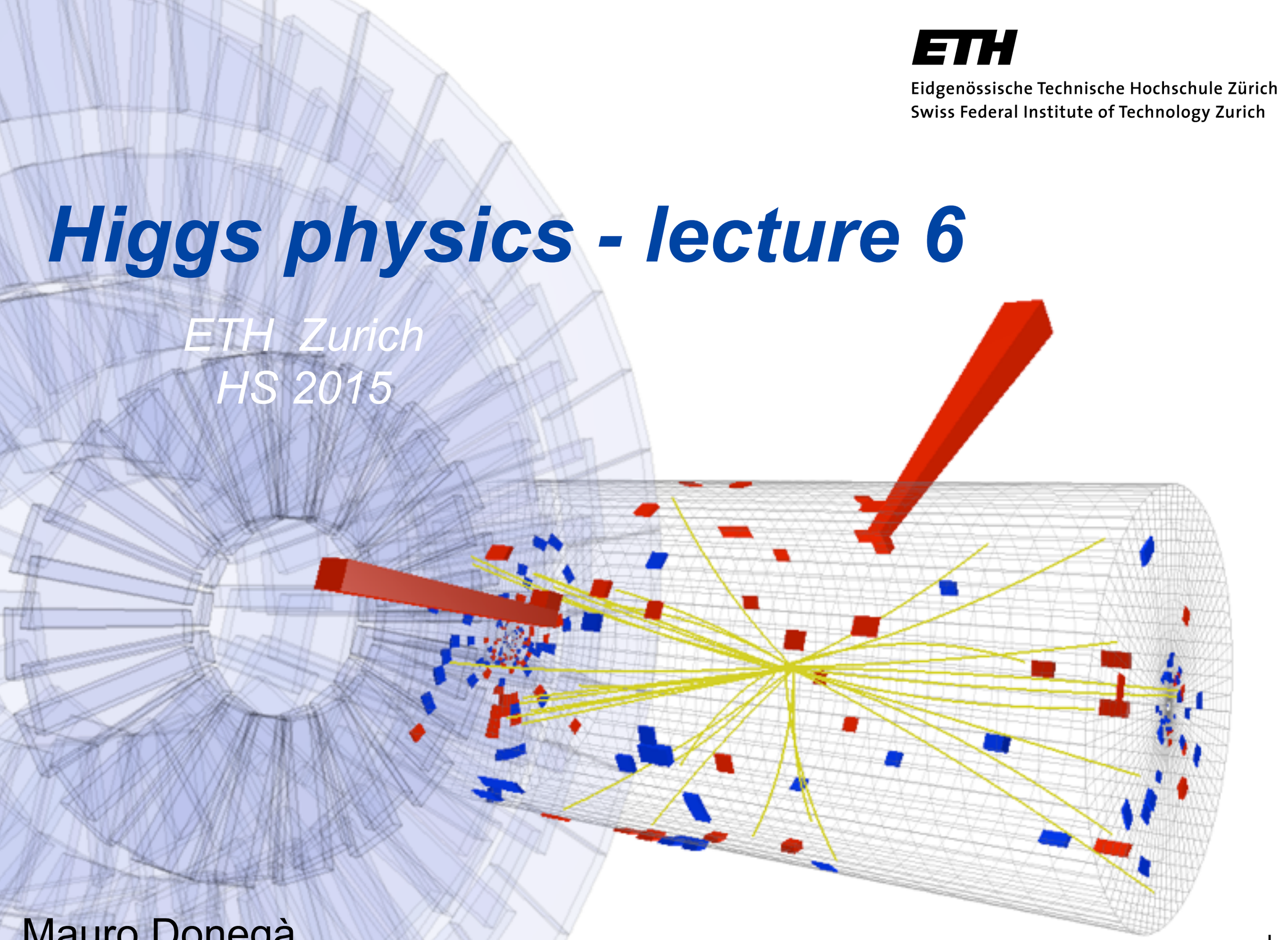
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Higgs physics - lecture 6

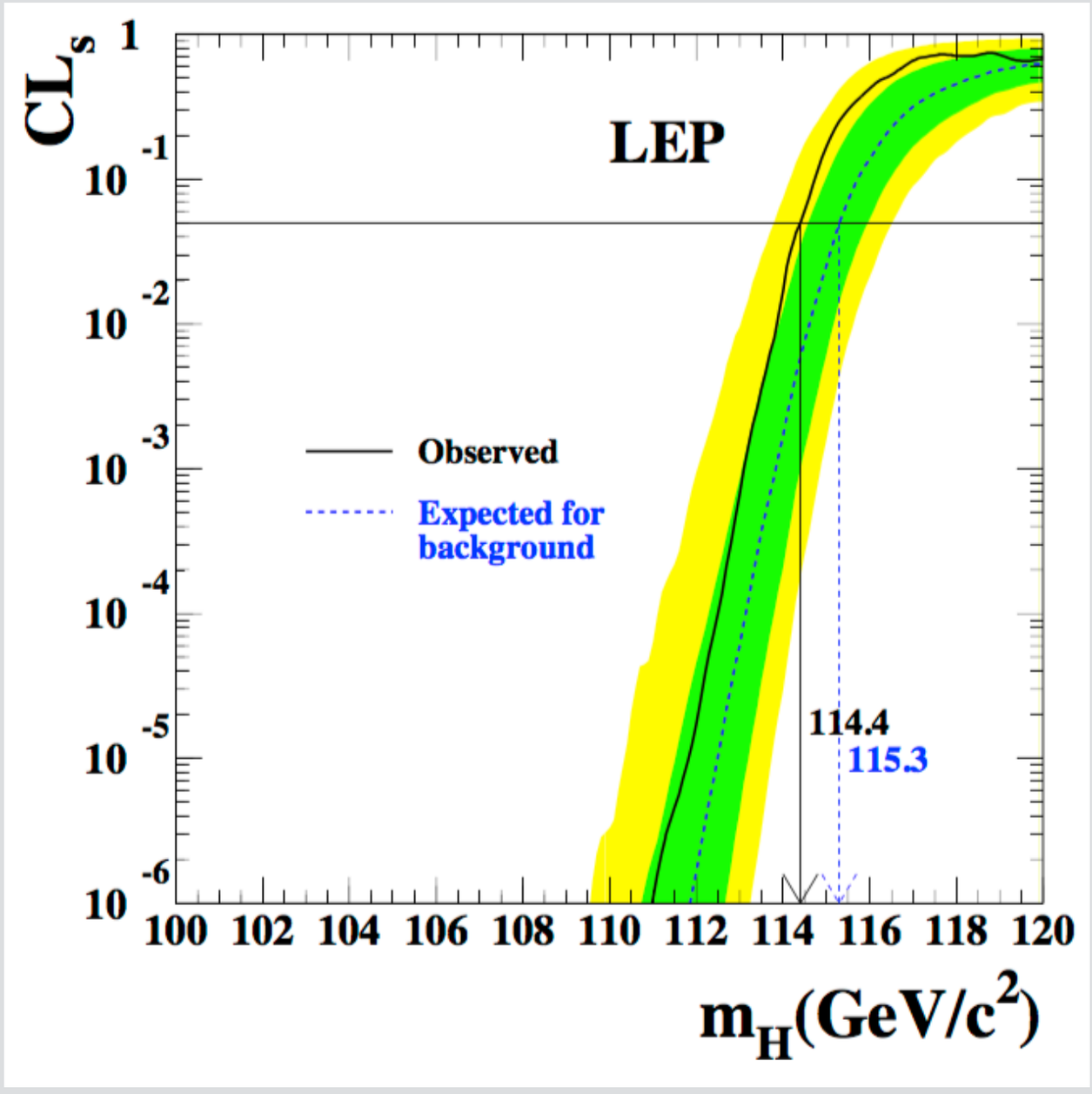
ETH Zurich
HS 2015



Mauro Donegà

Outline

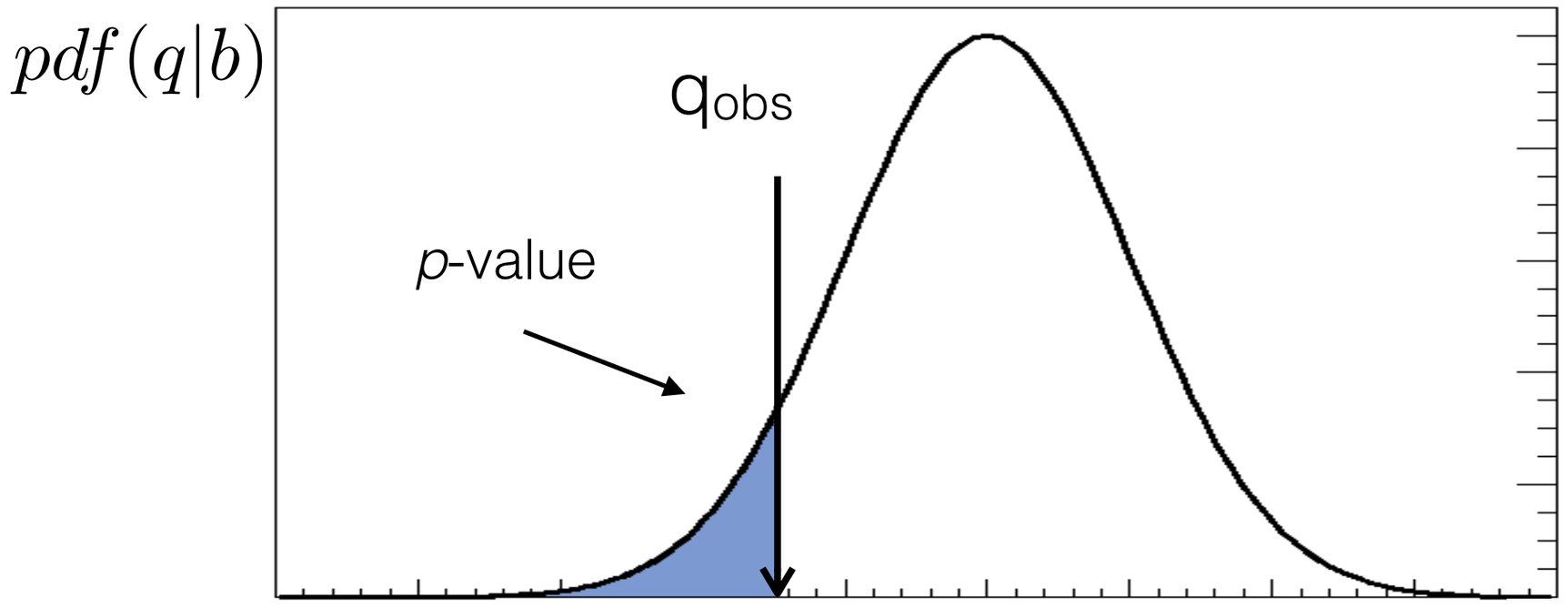
	Lectures	Exercises
1	• Introduction	
	• Accelerators	
	• Detectors	
	• EW constraints	■ W mass
2	• Search at LEP1 / LEP 2	
	• Statistics: likelihood and hypothesis testing	■ CLs
3	• Searches at TeVatron	
	Channels overview	
	Neural Networks	
	Results	■ Hbb TeVatron
4	• LHC	
	Channels overview	
	Dissect one analysis	■ SciKit MVA
5	Higgs combination	■ HZZ/Hgg
6	• Extras	
	Differential distributions	
	Off shell	
	Beyond Standard Model	
	pseudo-observables / EFT	■ Toy CVCF


CL$_s$
1
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
LEP
Observed
Expected for background
114.4
115.3
100 102 104 106 108 110 112 114 116 118 120
m$_H$(GeV/c^2)

CLs @ LEP

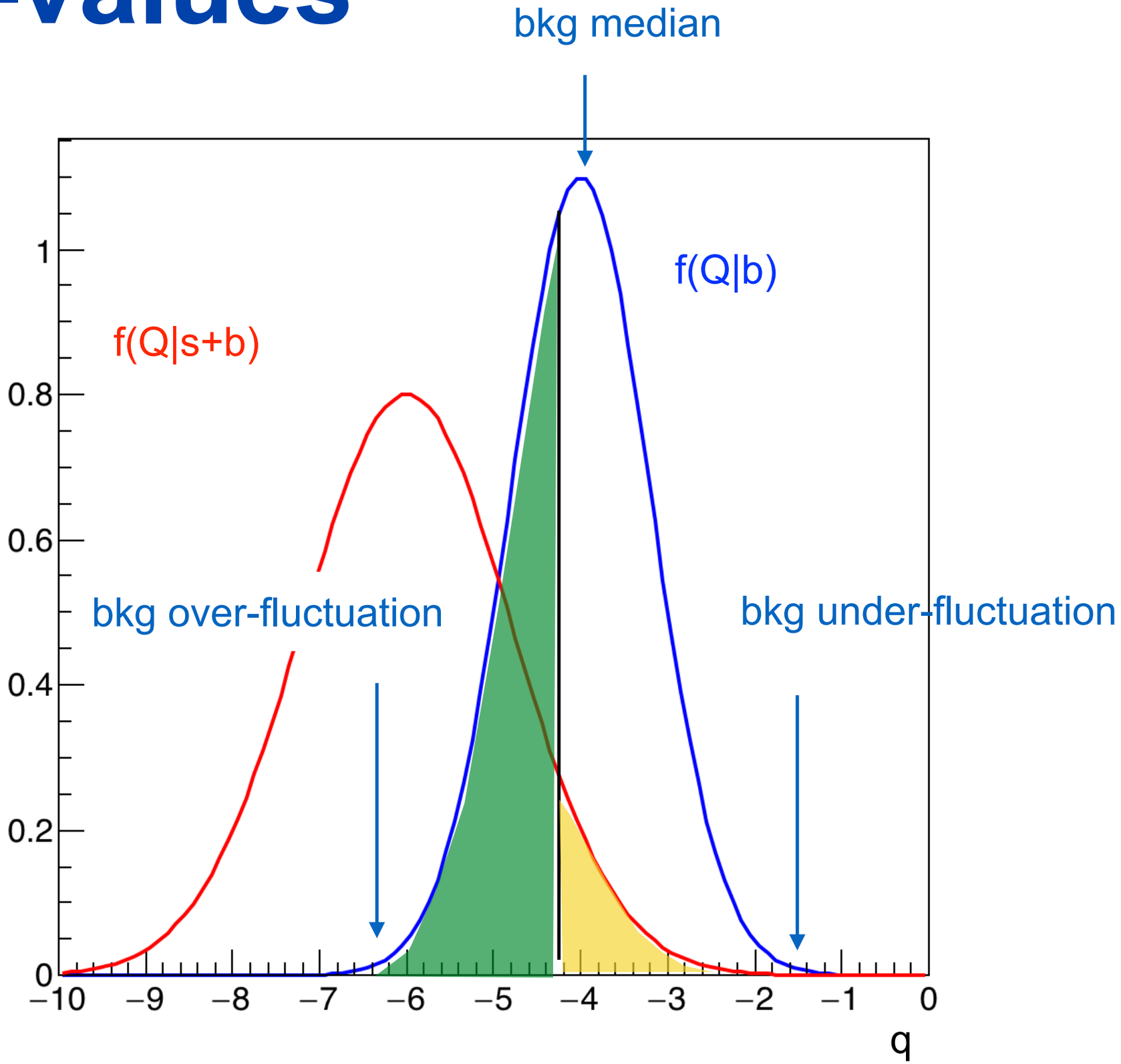
p-values

p-value = *probability, under the assumption of H, to observe data with equal or lesser compatibility with H relative to the data we got*

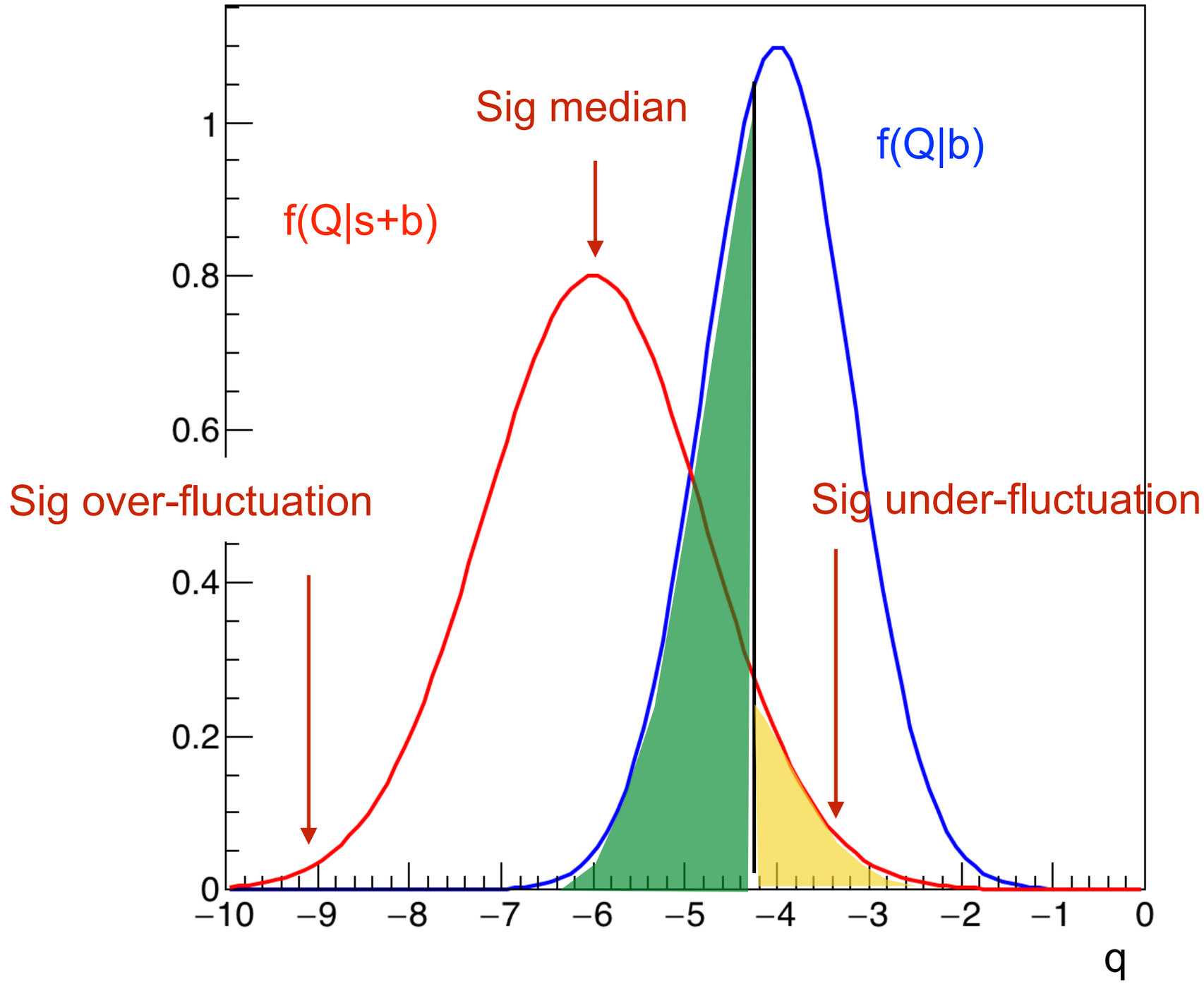


In the HEP folklore we claim (on the background-only hypothesis):
- observation if the p-value < $1.4\ 10^{-3}$ (3σ) `ROOT> 1-TMath::Freq(3)`
- discovery if the p-value < $2.9\ 10^{-7}$ (5σ) `ROOT> 1-TMath::Freq(5)`

more *p*-values



more *p*-values

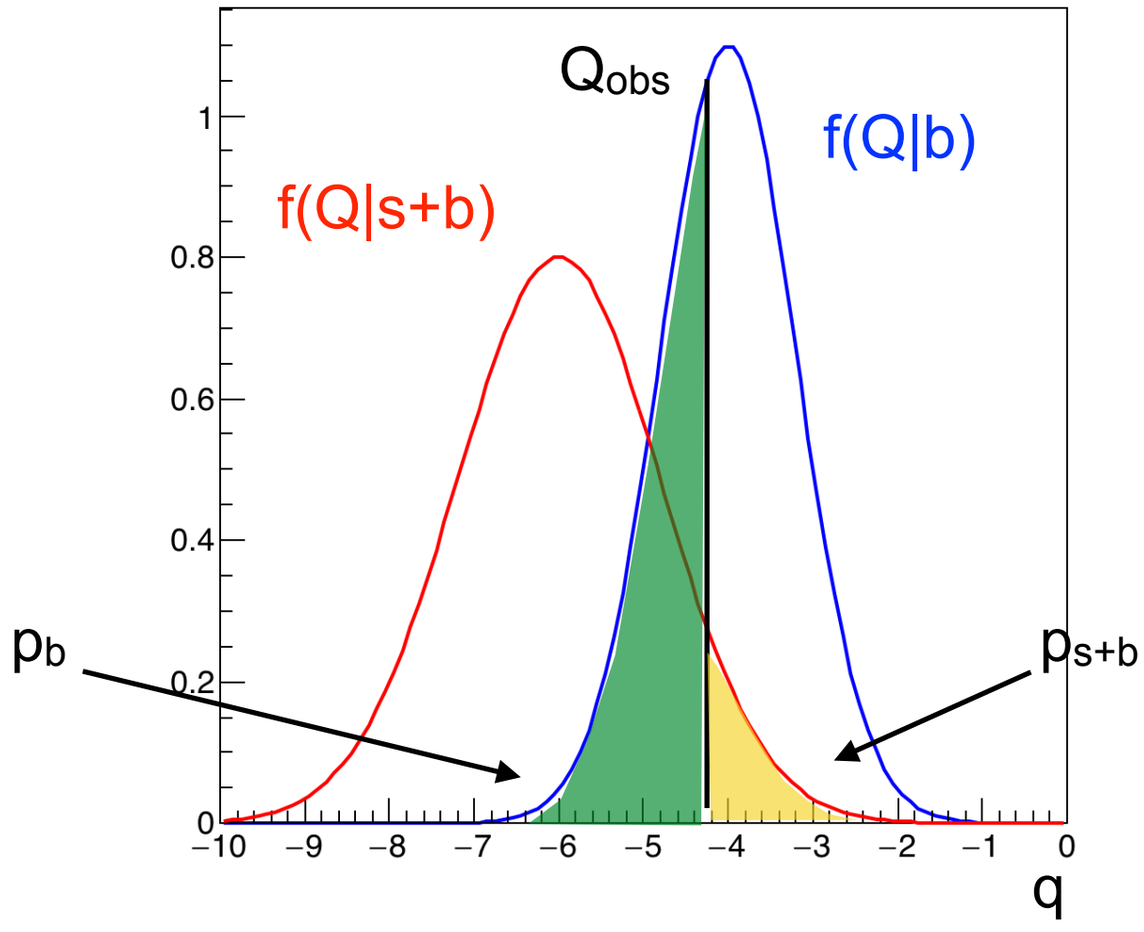


more *p*-values

The p-value is a general way to quantify a deviation.
We usually define :

p_b deviation defined on the background hypothesis

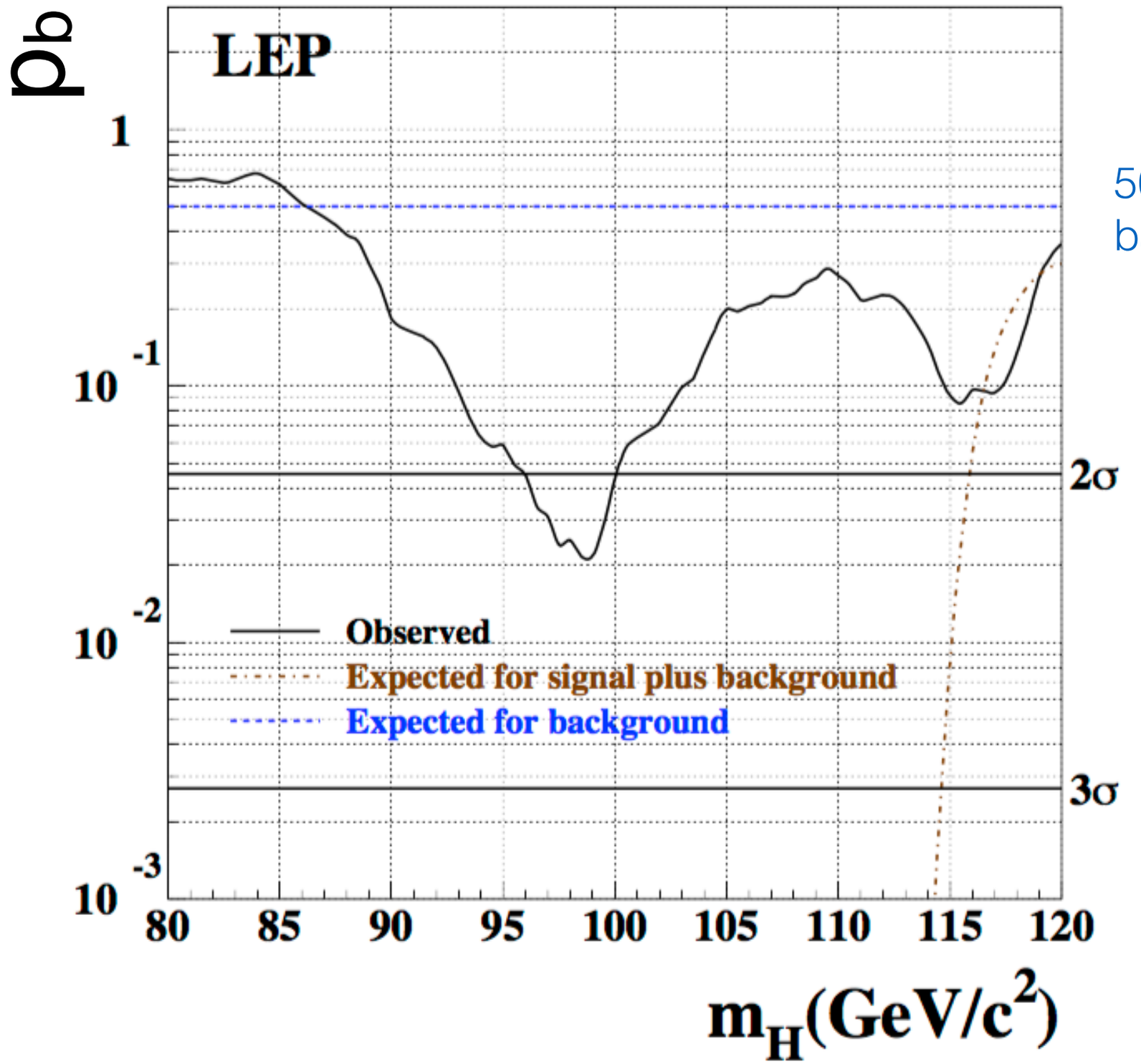
p_{s+b} deviation defined on the signal + background hypothesis



If p_{s+b} <α (computed on f(Q|s+b)) we reject the s+b hp at 1-α CL

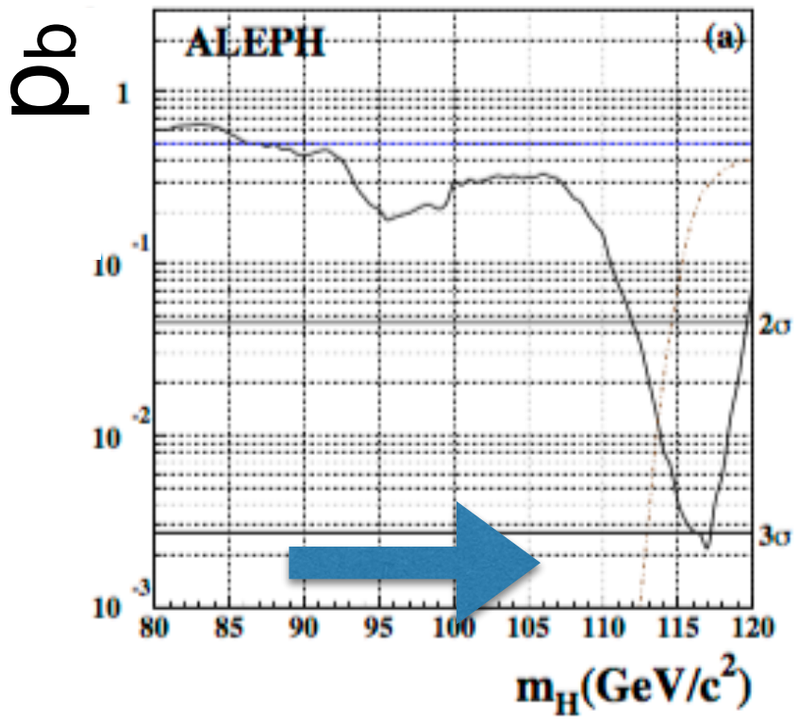
If p_b <α (computed on f(Q|b)) we reject the b hp at 1-α CL

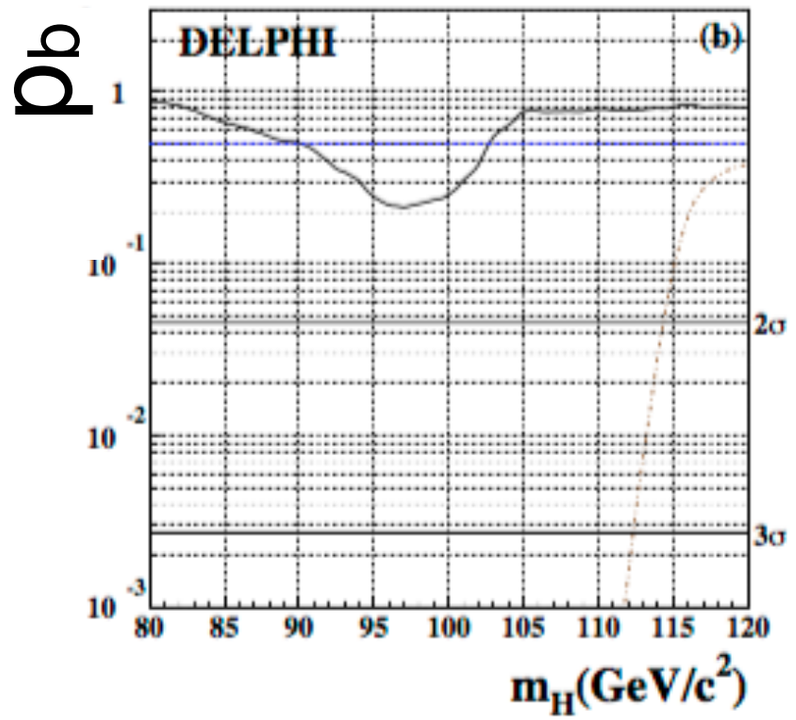
LEP results

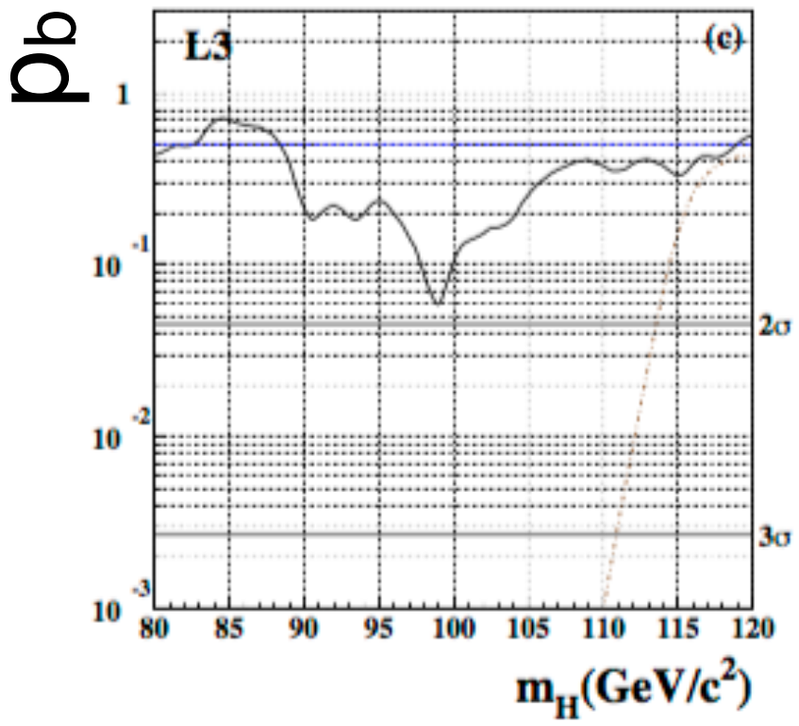


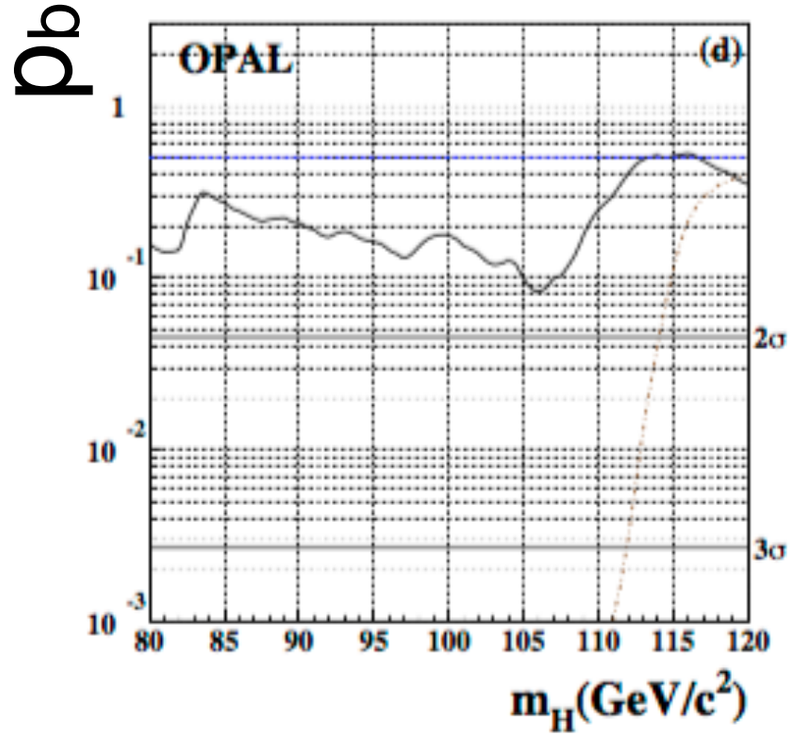
50%
background only

LEP results


ALEPH
(a)
p_b
1
10^{-1}
10^{-2}
10^{-3}
2σ
3σ
80 85 90 95 100 105 110 115 120
$m_H(GeV/c^2)$

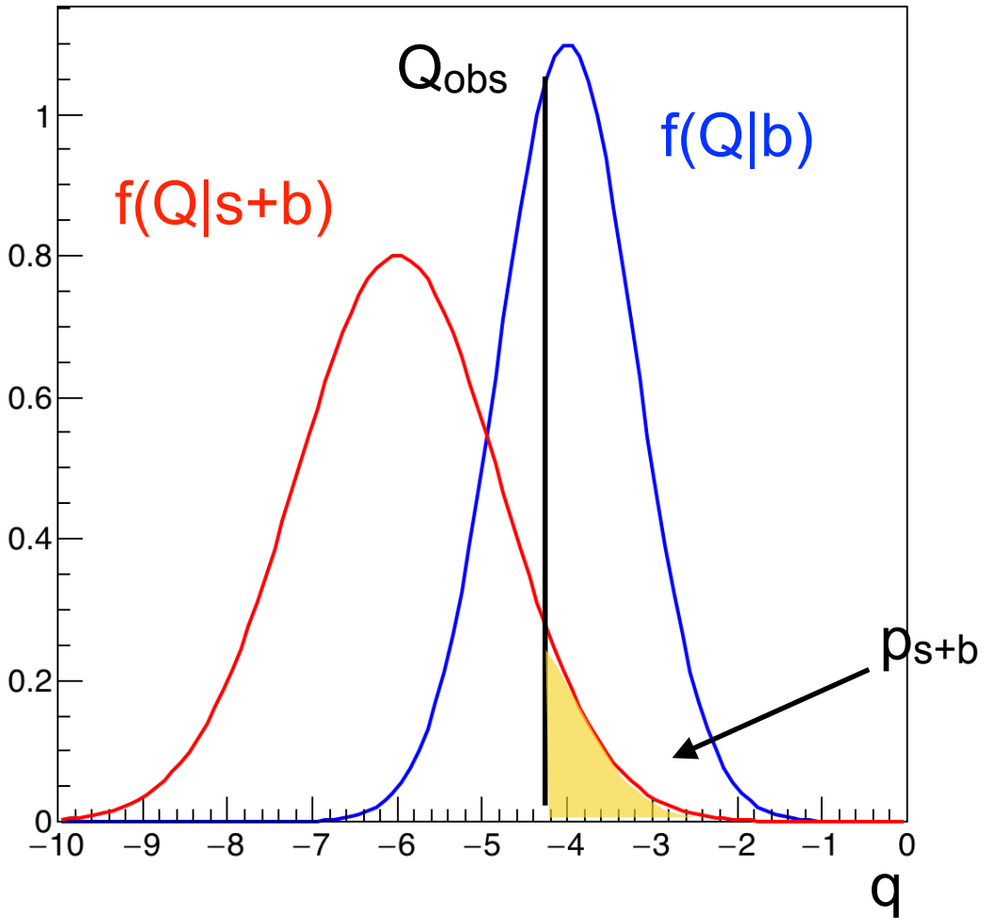

DELPHI
(b)
p_b
1
10^{-1}
10^{-2}
10^{-3}
2σ
3σ
80 85 90 95 100 105 110 115 120
$m_H(GeV/c^2)$


L3
(c)
p_b
1
10^{-1}
10^{-2}
10^{-3}
2σ
3σ
80 85 90 95 100 105 110 115 120
$m_H(GeV/c^2)$

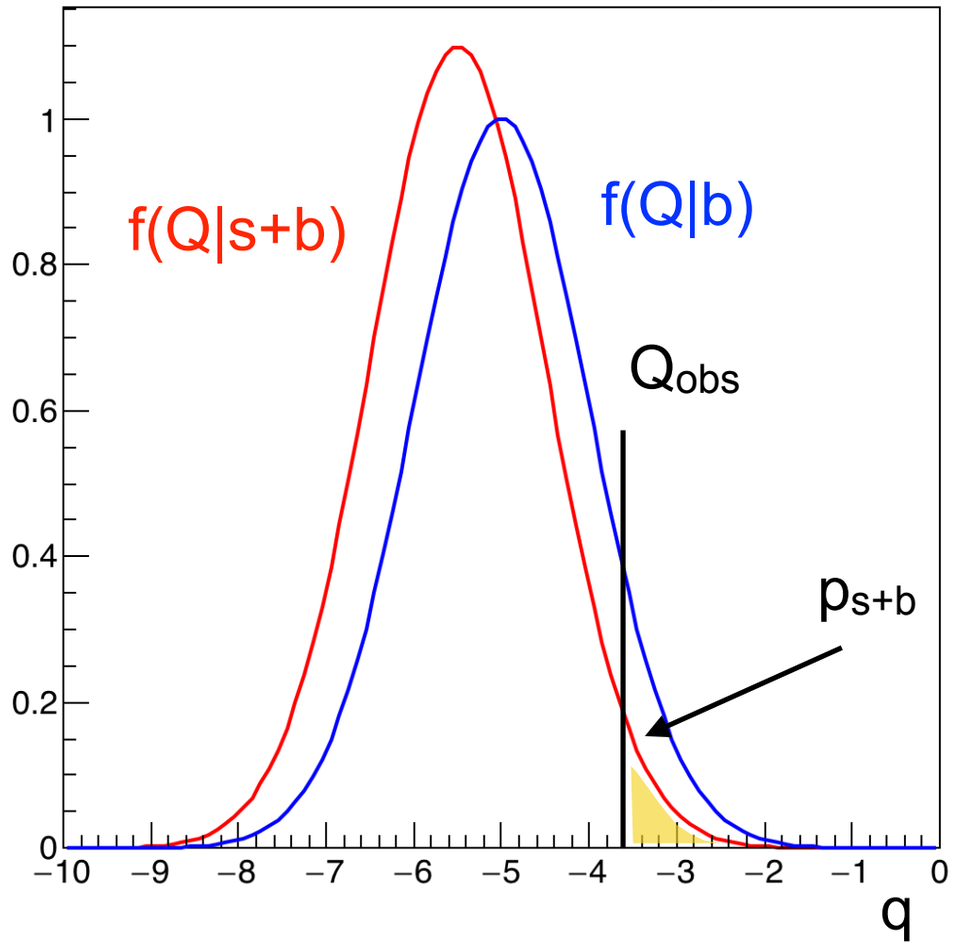

OPAL
(d)
p_b
1
10^{-1}
10^{-2}
10^{-3}
2σ
3σ
80 85 90 95 100 105 110 115 120
$m_H(GeV/c^2)$

The issue about sensitivity

Well separated



NOT well separated f(Q|s+b) ~ f(Q|b)



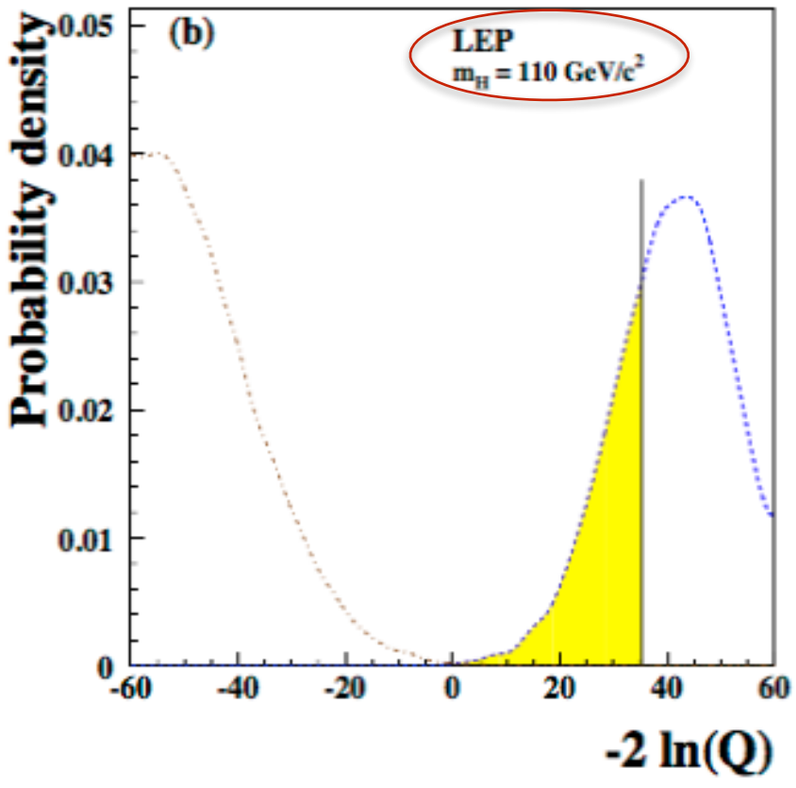
The small separation power means low sensitivity to the signal:
Suppose you look at the LHC (sqrt(s) = 13 TeV) for a resonance at 100 TeV. In this case whatever test statistics you use to discriminate s+b from b will look practically identical for both cases.

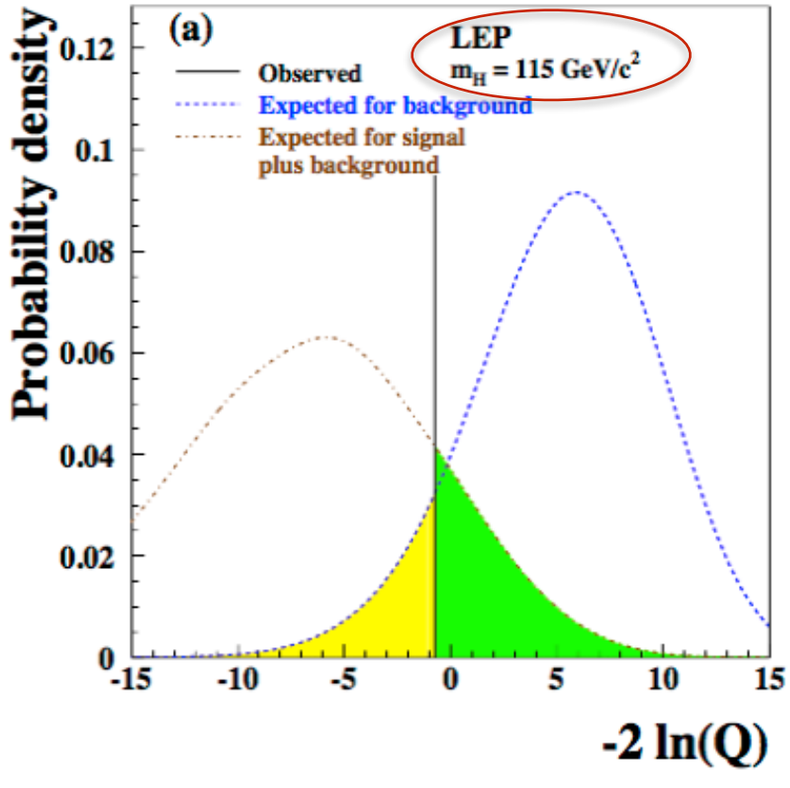
If the number of events fluctuates below the expected bkg,both s+b and b are disfavoured. But, given the low p_{s+b} (e.g. 5%) one can exclude the s+b at 95% CL, which is wrong because you don't have sensitivity !

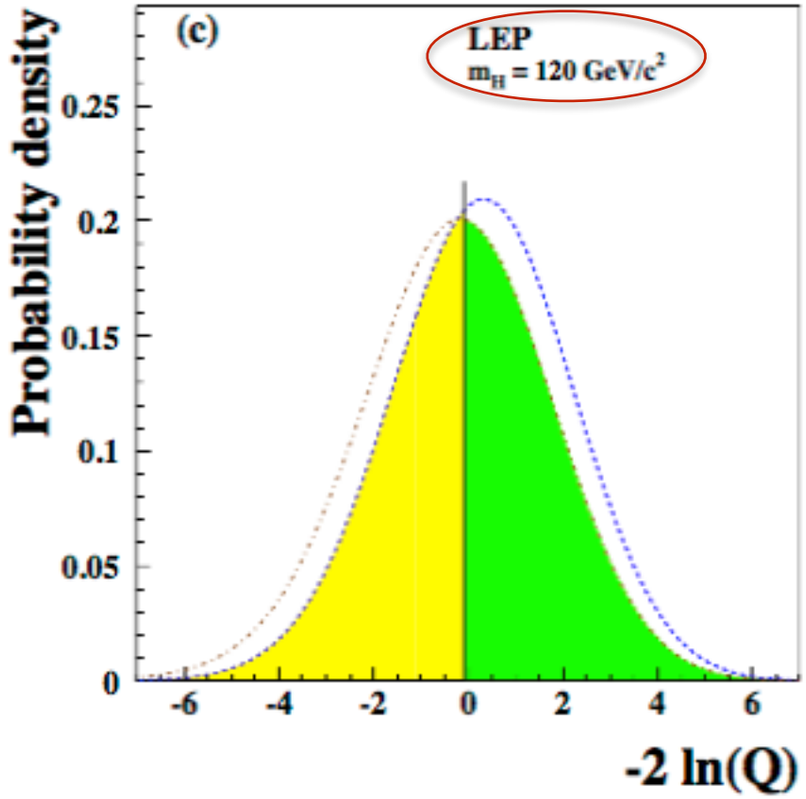
We don't want to exclude a signal that we are not sensitive to !
We have to include some information about p_b

The issue about sensitivity @ LEP

Here we use the same data and build test hypothesis changing the test mass m_H.







The higher the mass the lower the power of the test the lower the sensitivity

CLs method

$$CL_s \equiv CL_{s+b}/CL_b$$

Vocabulary A. Read: "confidence level = p-value"

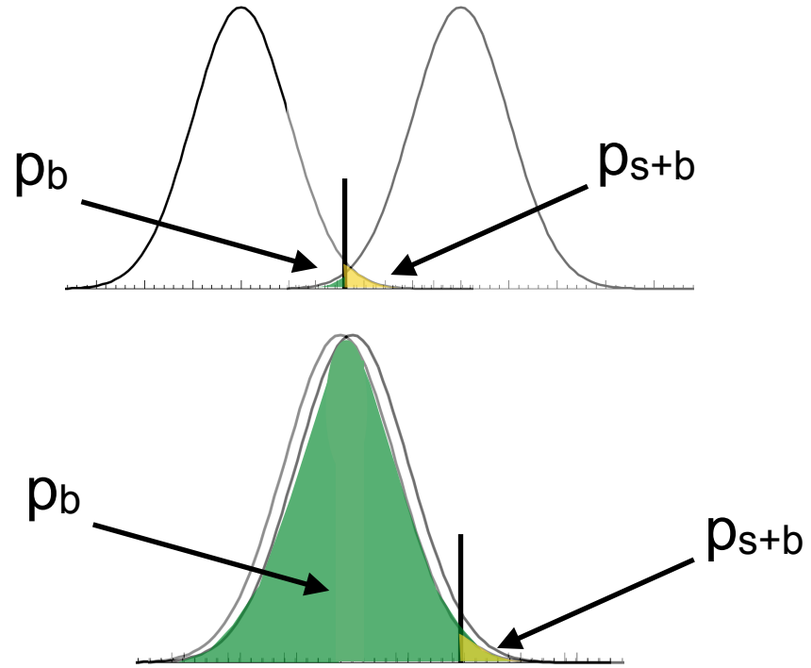
The CLs is NOT a p-value (it's a ratio of p-values) NEVERTHELESS we will say that

a signal will be considered excluded at the confidence level CL if 1-CLs ≤ CL

Consequences:

1) the false exclusion rate is less than the nominal 1-CL

$$CL_s = \frac{CL_{s+b}}{CL_b} = \frac{p_{s+b}}{1-p_b}$$



In case of clear background exclusion

$p_b \rightarrow 0$ then $CL_s \rightarrow p_{s+b}$

In case of no separation

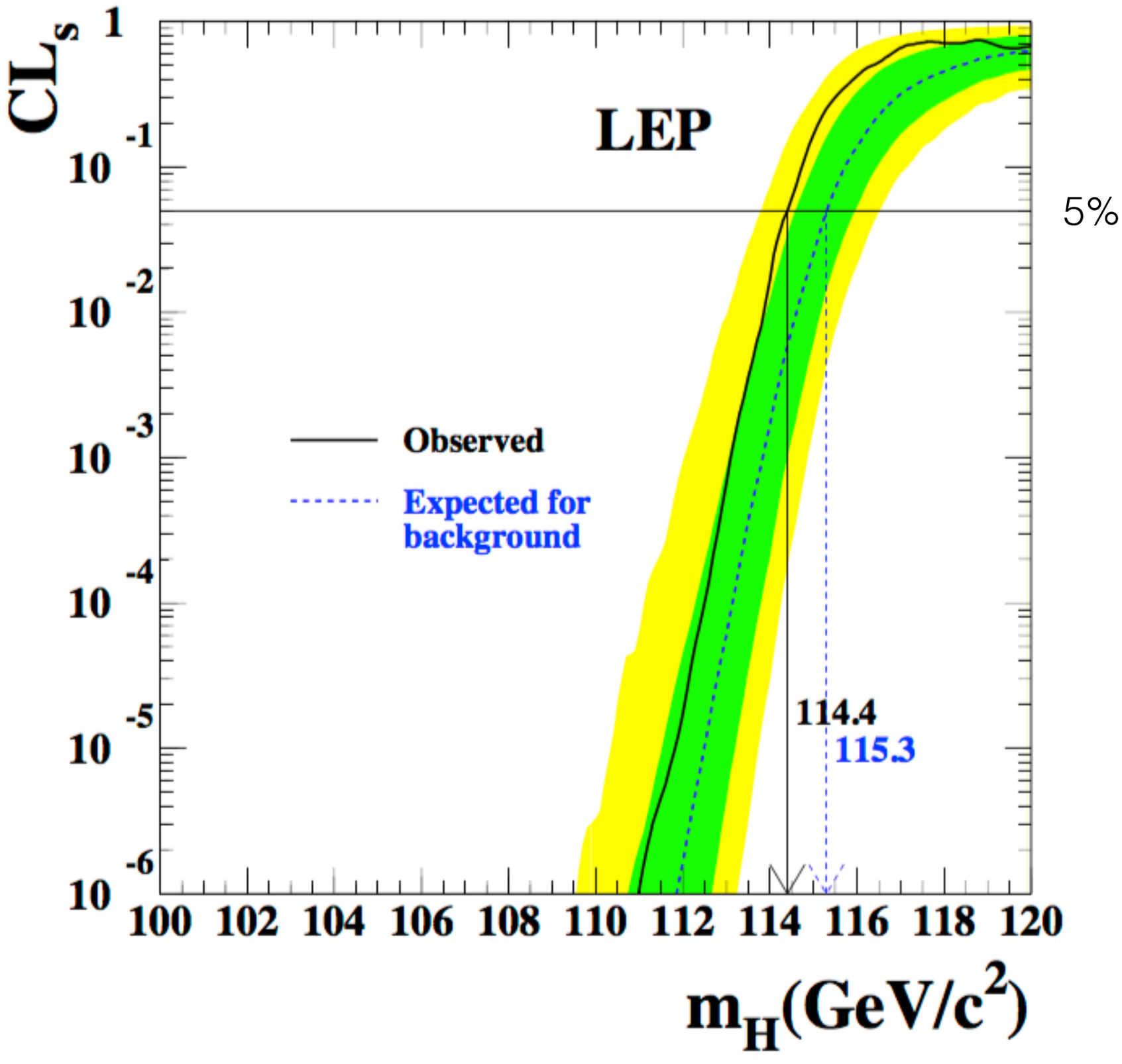
$p_{s+b} \rightarrow 1-p_b$ $CL_s \rightarrow 1$

i.e. the difference between CL and CLs will increase the lower the sig/bkg separation

2) the use of CL_s increases the "coverage" of the analysis

(i.e. at a given CL you exclude a smaller region of the space of parameters)

Final LEP exclusion



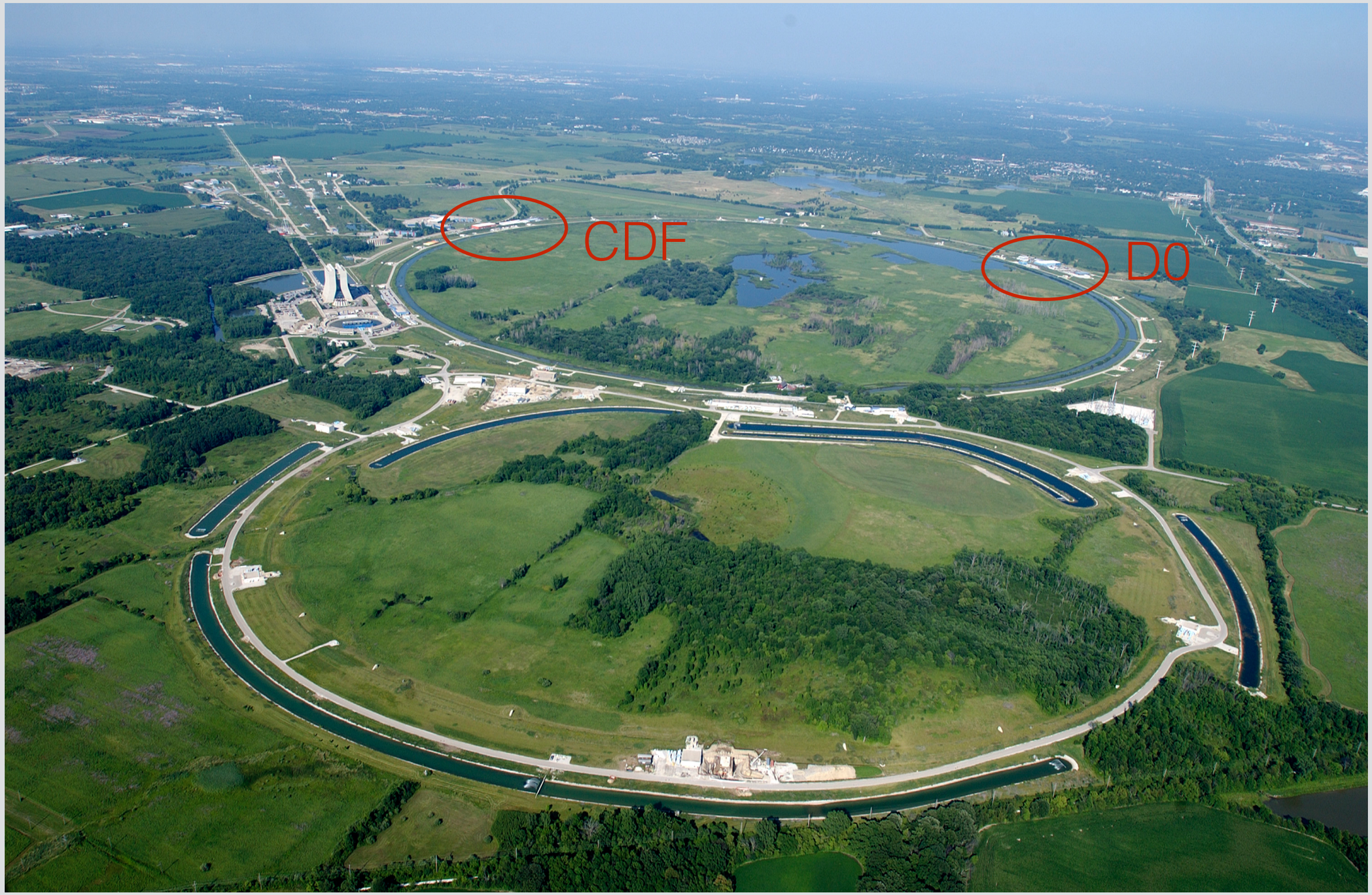
Bibliography

The searches for Higgs Bosons at LEP
M. Kado and C. Tully, Annu. Rev. Nucl. Part. Sci. 2002. 52:65-113

LEP Working group for Higgs Boson Searches: ADLO collaborations
Phys. Lett. B565, 61-75 (2003). arXiv:hep-ex/0306033

Designing and building LEP
K. Hubner, Physics reports 403-404 (2004) 177-188

The Higgs Hunter's Guide
by J.F. Gunion, H. Haber, G. Kane, S.Dawson

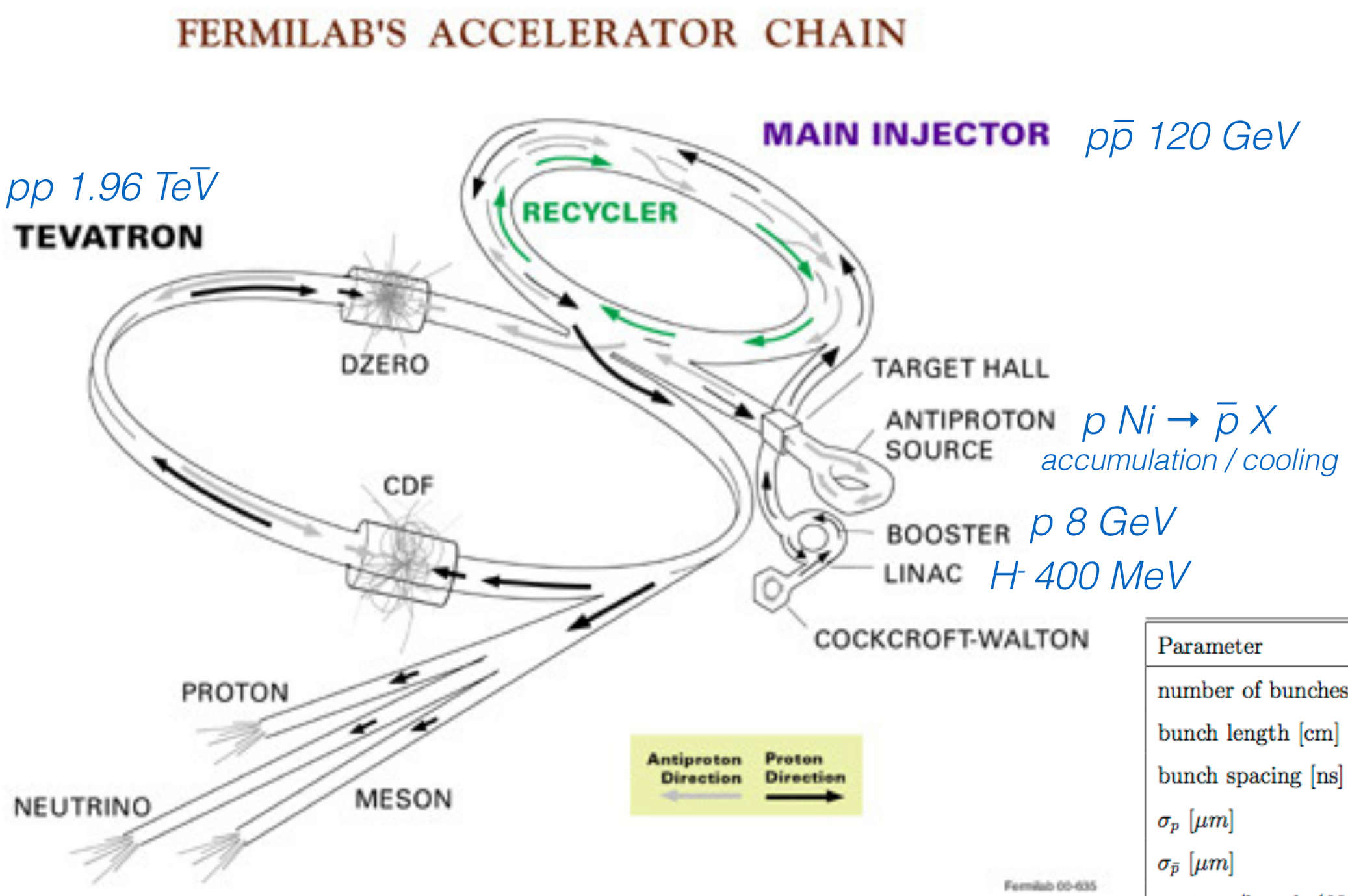


Searches at TeVatron: CDF / D0

ETH *Mauro Donegà: Higgs physics*

TeVatron collider

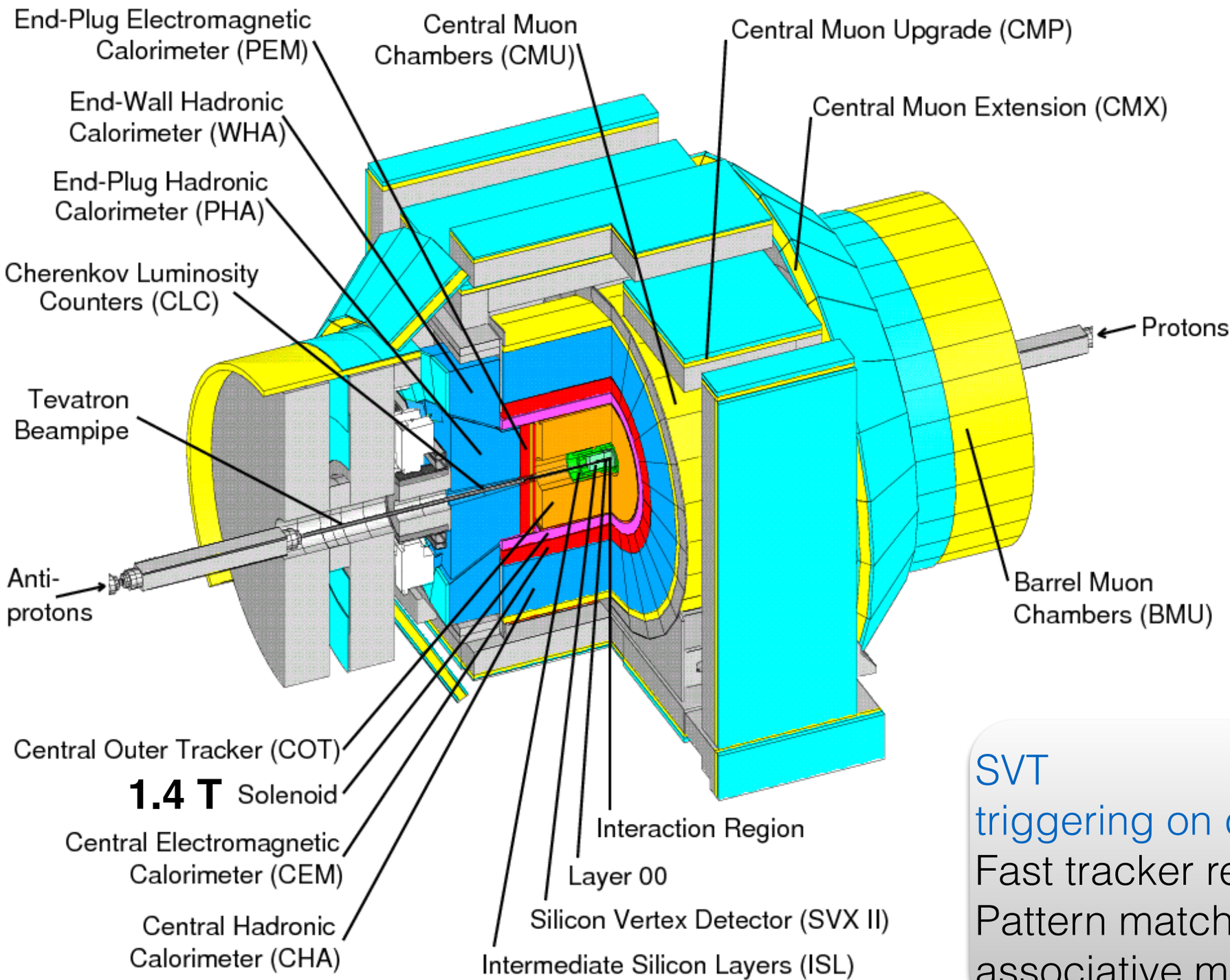
1983 - 2011



Parameter	
number of bunches (N_B)	36
bunch length [cm]	18
bunch spacing [ns]	396
σ_p [μm]	20
$\sigma_{\bar{p}}$ [μm]	20
protons/bunch (N_p)	3.3×10^{11}
anti-protons/bunch (N_p)	3.6×10^{10}
interaction/crossing	5.3
typical luminosity [$cm^{-2}s^{-1}$]	0.9×10^{32}

http://www.fnal.gov/pub/tevatron/

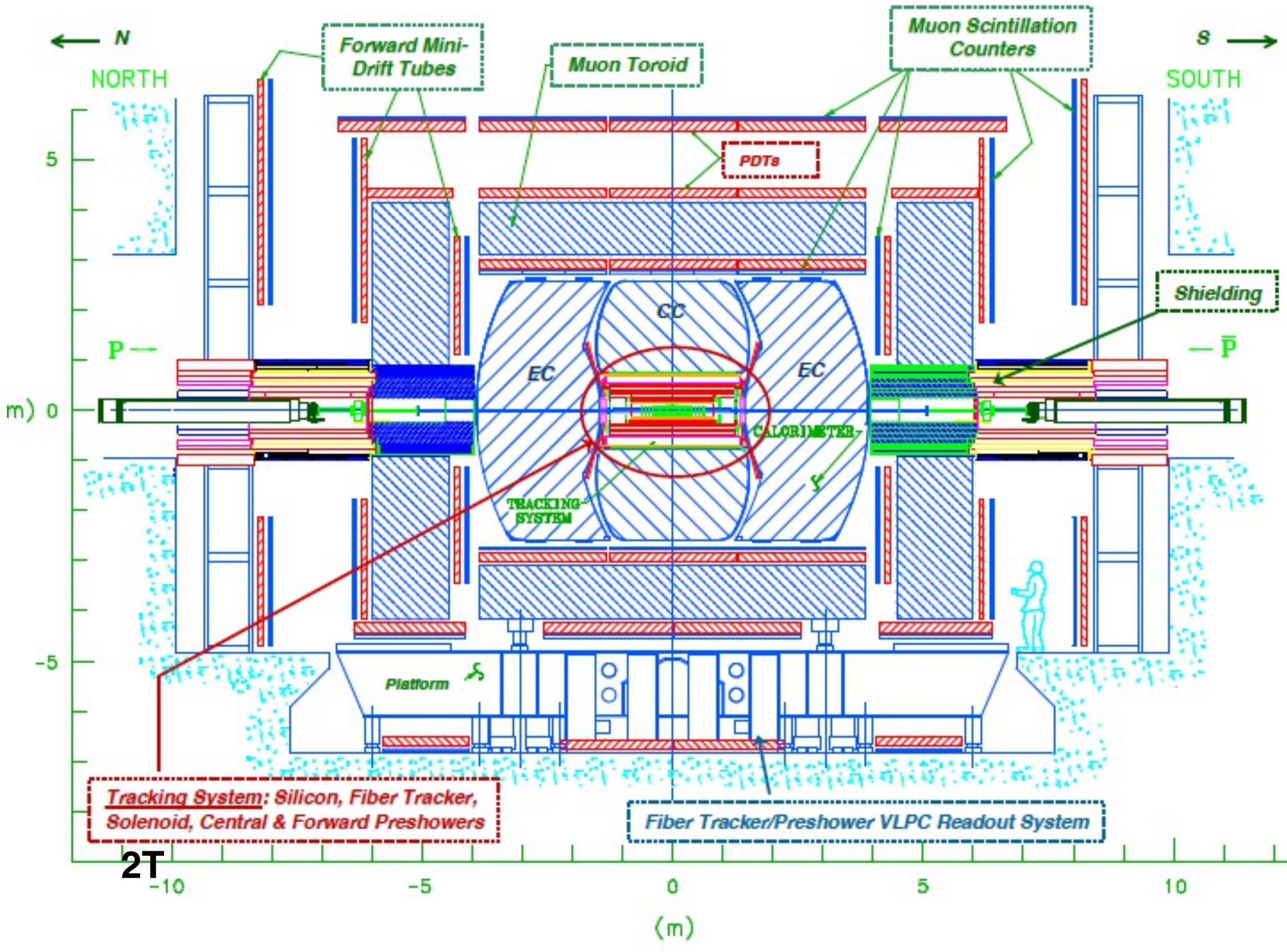
CDF



SVT
triggering on displaced tracks
Fast tracker readout
Pattern matching through associative memories

http://www-cdf.fnal.gov/index.html

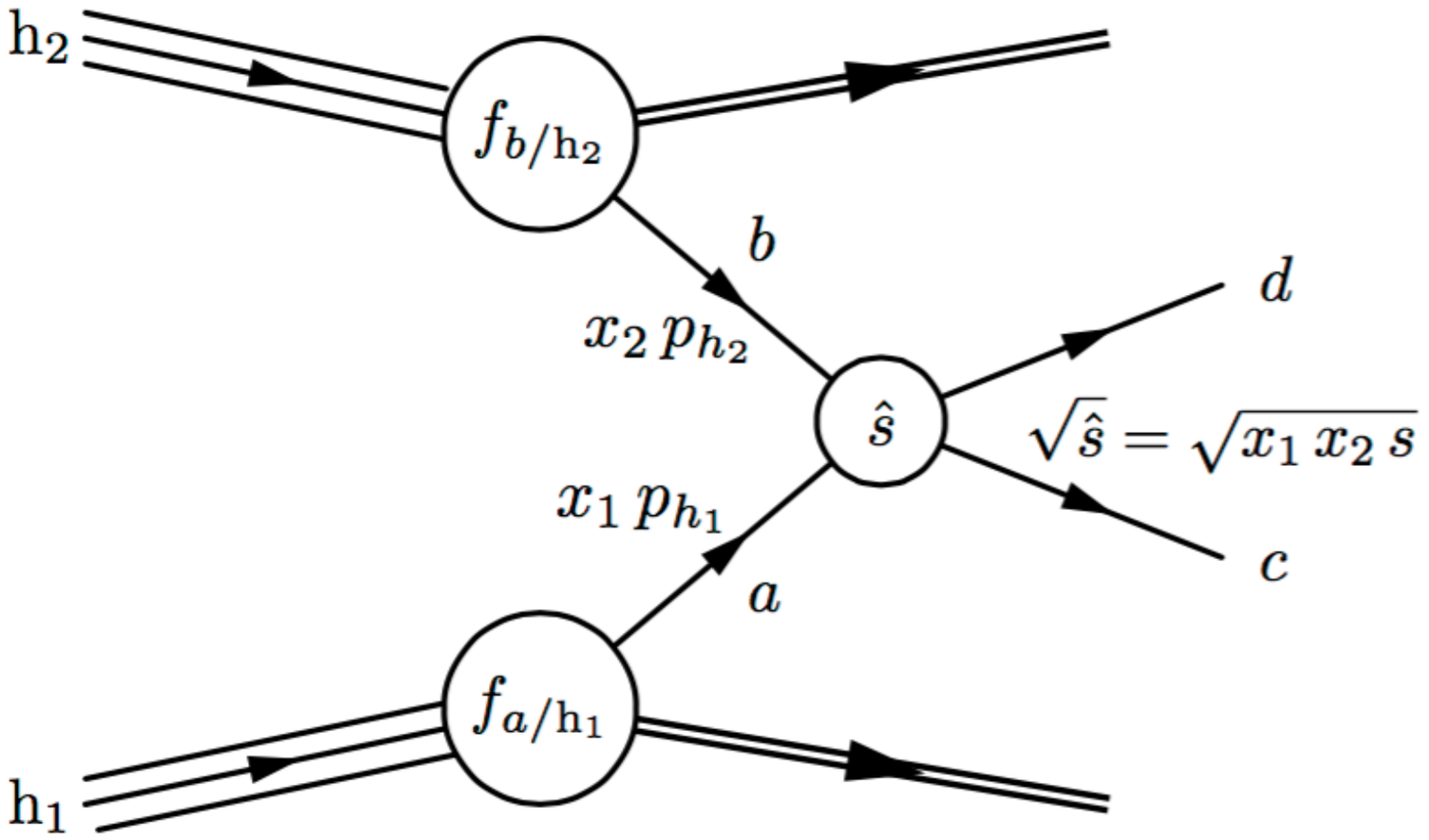
D0


N
NORTH
S
SOUTH
Forward Mini-Drift Tubes
Muon Toroid
Muon Scintillation Counters
PDTs
Shielding
CC
EC
EC
P
$\bar{P}$
CALORIMETER
TRACKING SYSTEM
Platform
Tracking System: Silicon, Fiber Tracker, Solenoid, Central & Forward Preshowers
Fiber Tracker/Preshower VLPC Readout System
2T
5
0
-5
-10
-5
0
5
10
(m)

Hadronic/partonic collision

e^+e^- colliders:
- elementary (at this energies)
- fixed centre-of-mass energy
- clean events, precision physics
- bremsstrahlung-limited energy

Hadron colliders:
- discovery machines
- higher energies
- "no" fixed centre-of-mass energy



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

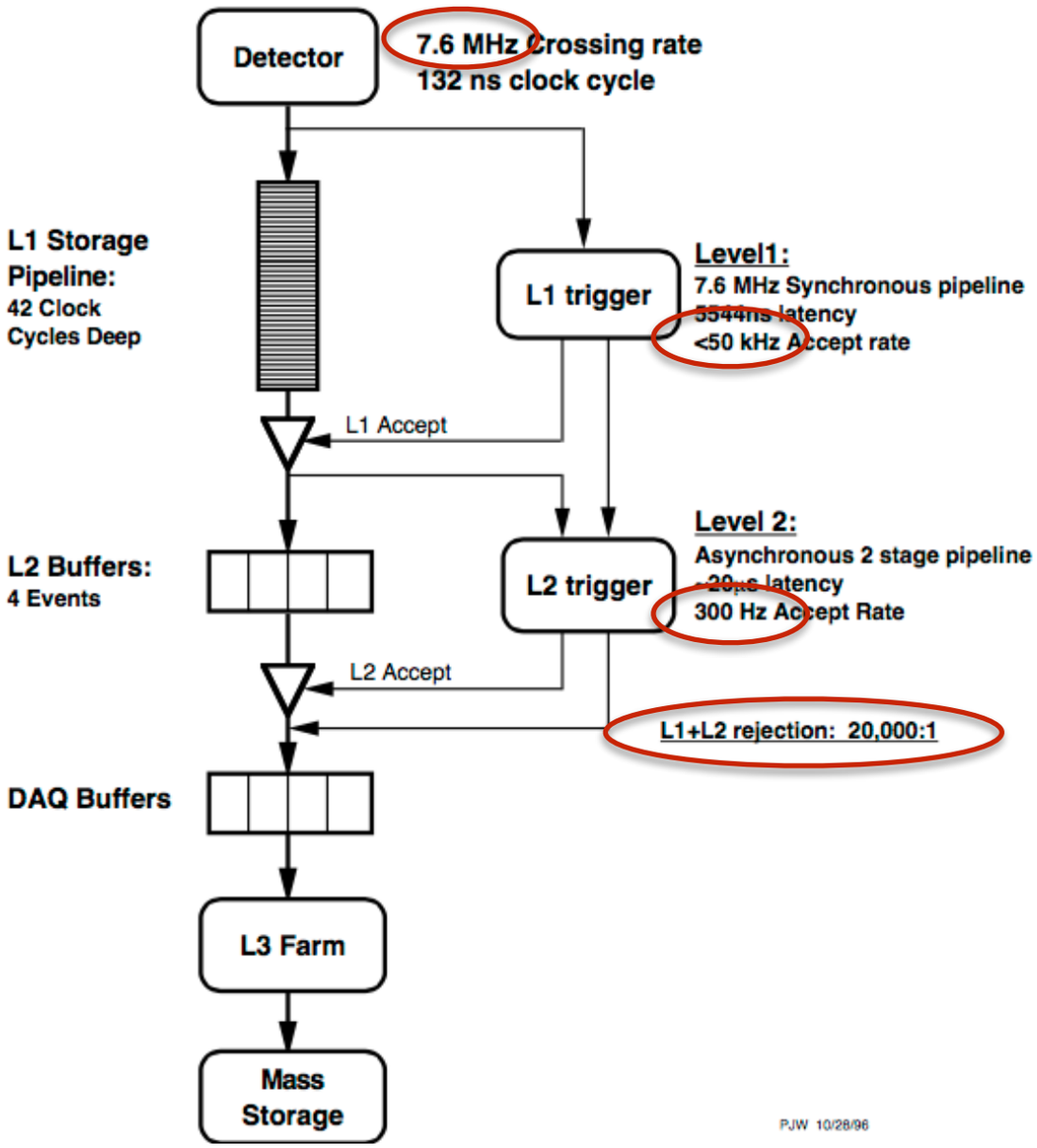
$f_i(x, Q^2)$ = probability of finding in the proton a parton of flavour i (quarks or gluon) carrying a fraction x of the proton momentum with Q being the energy scale of the hard interaction. QCD does not predict the parton content of the proton, the shapes of the PDFs are determined by a fit to data: DIS, Fixed Target, colliders.

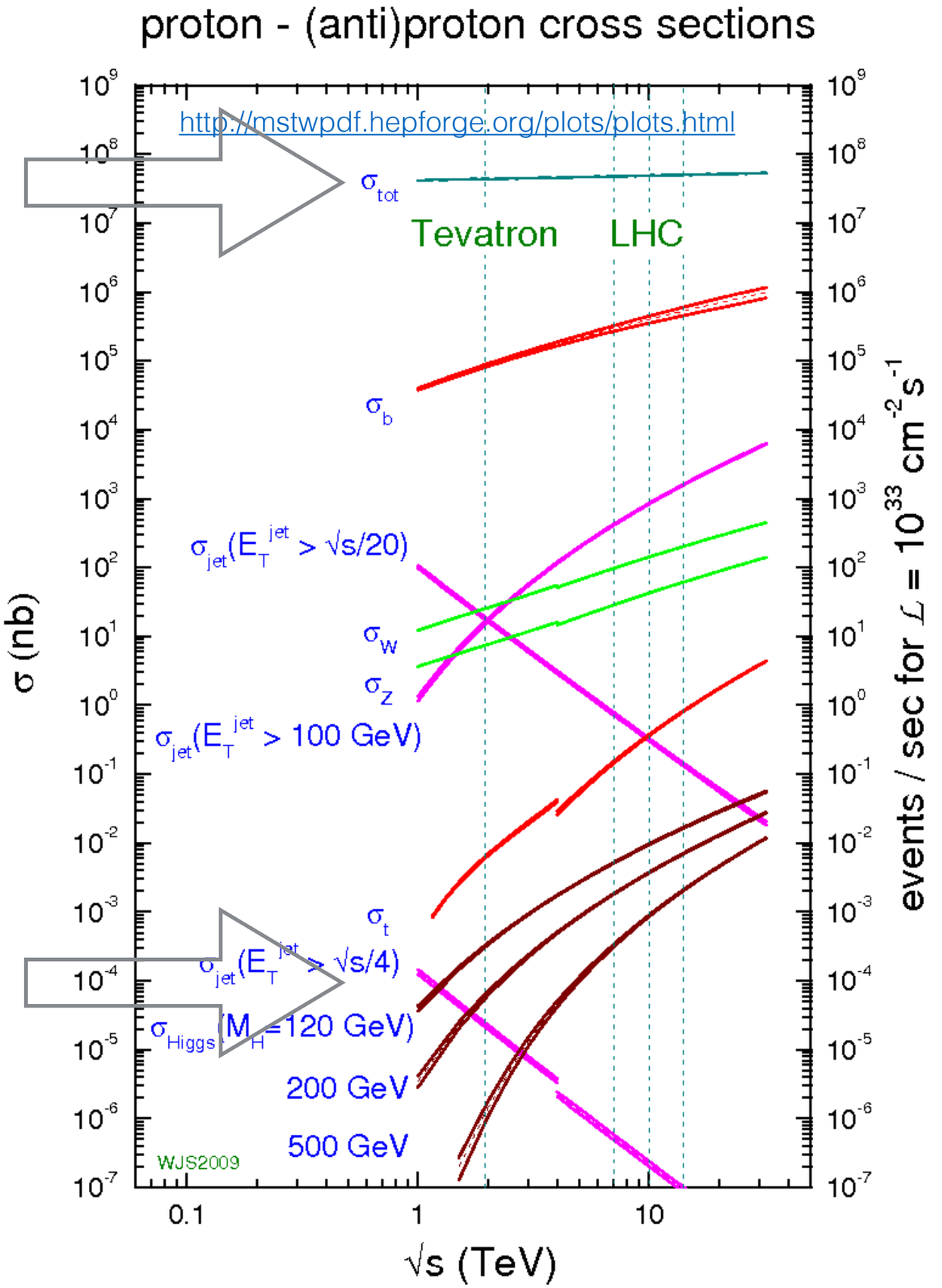
(see next Th. lecture for a quick review of PDF)

Cross sections and rates

Largest cross section from
inelastic low pT pp collisions
average pT < 1 GeV

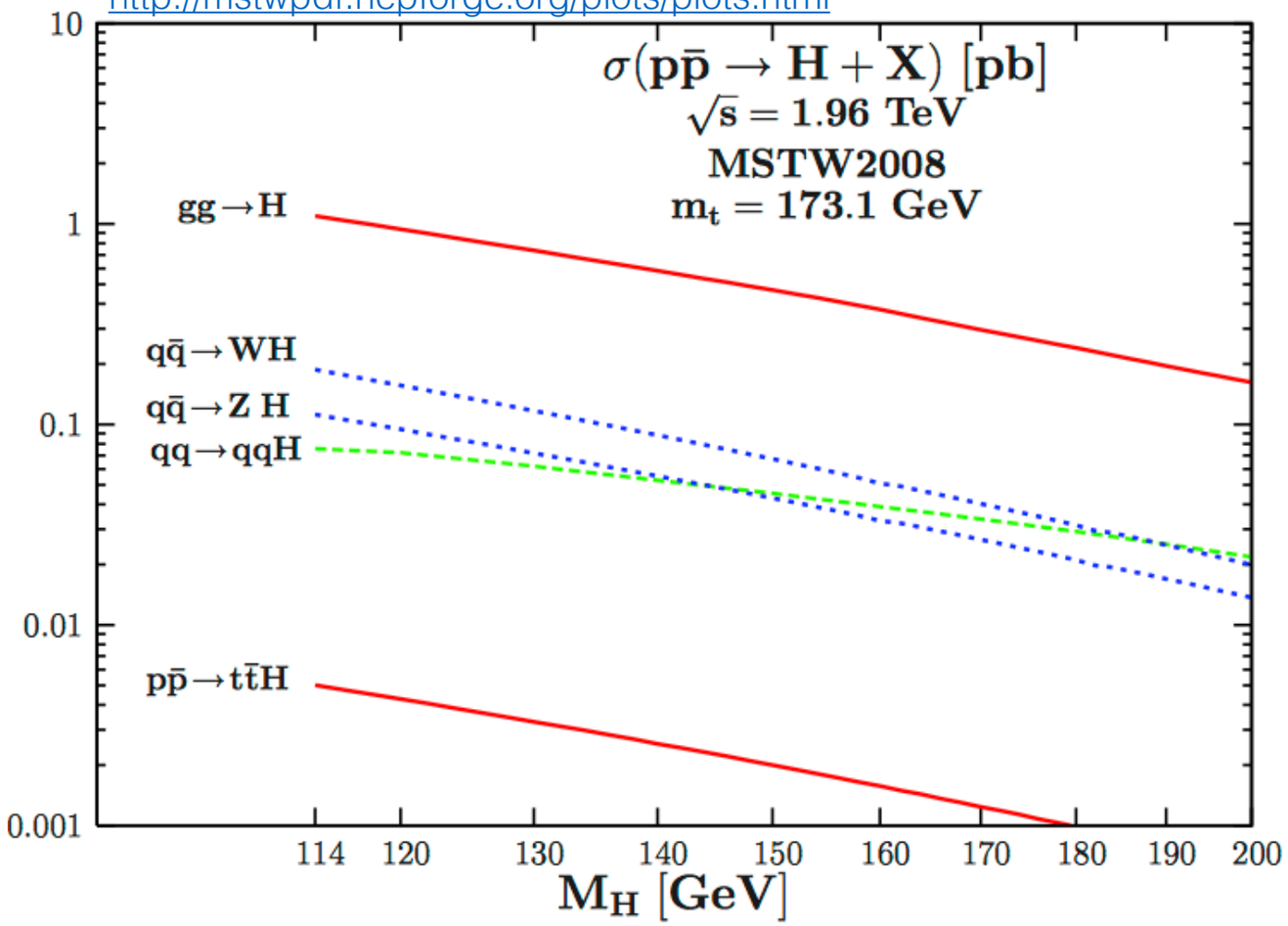
Trigger system:
tight online selection

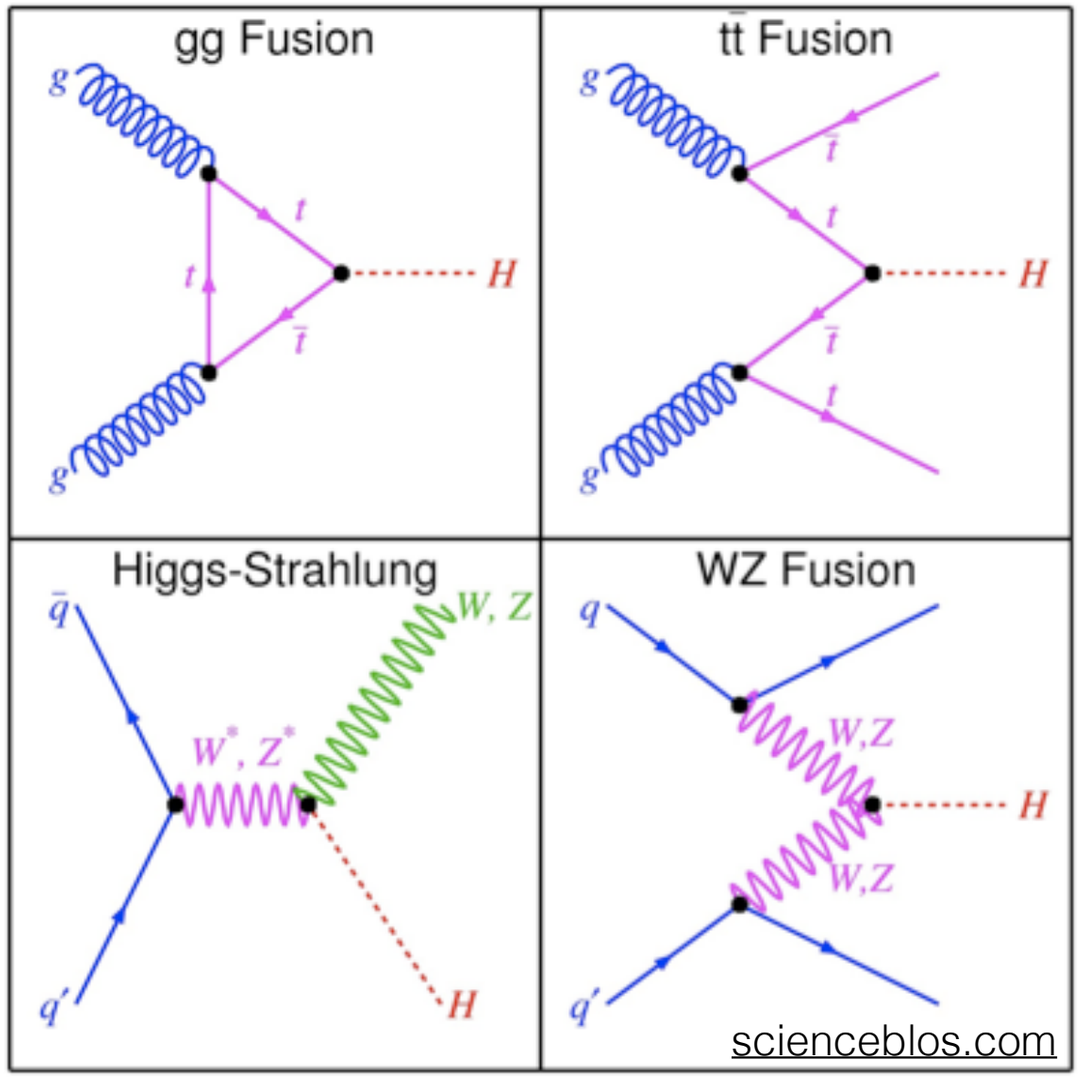




Higgs production and BR

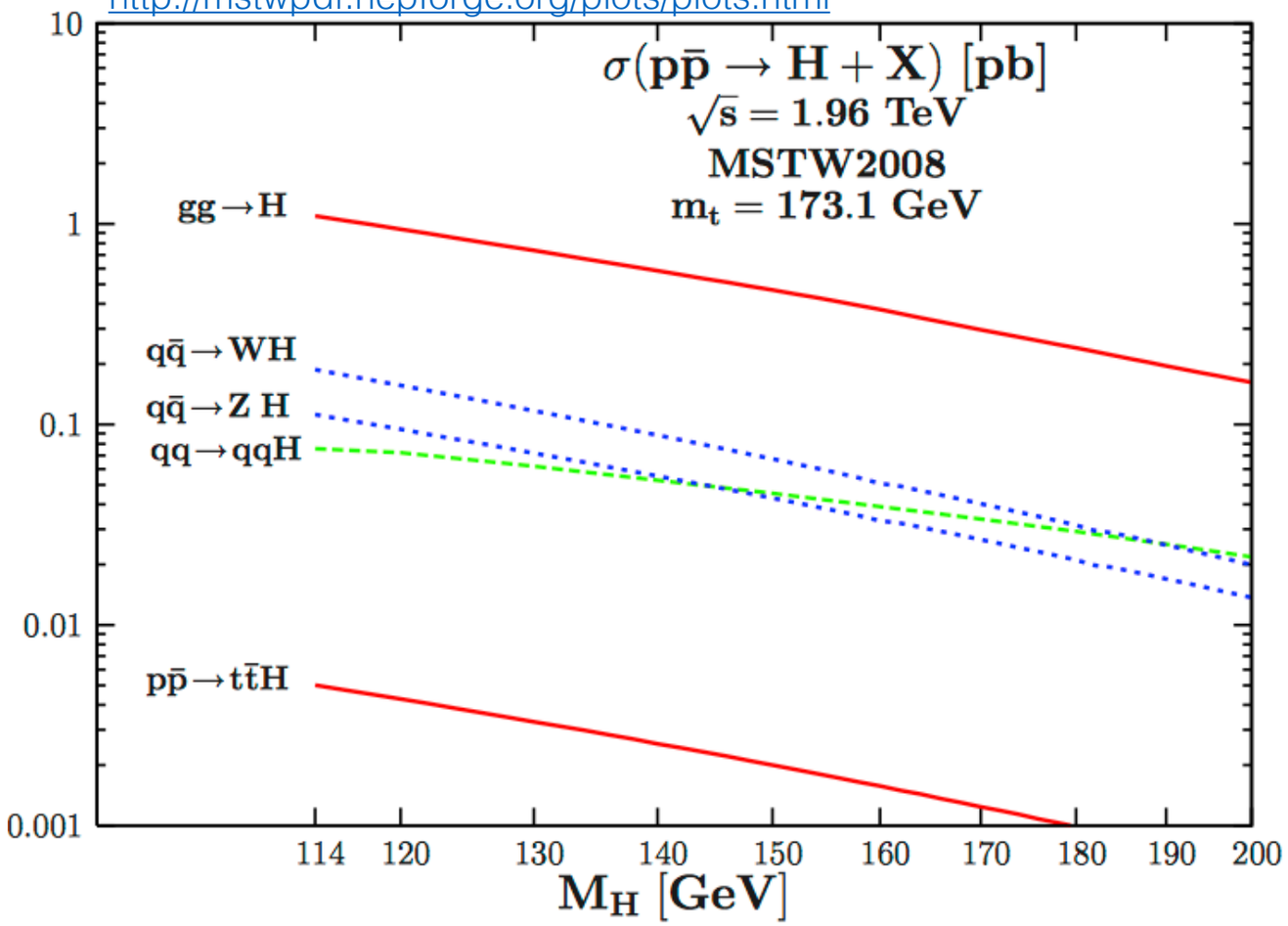
http://mstwpdf.hepforge.org/plots/plots.html

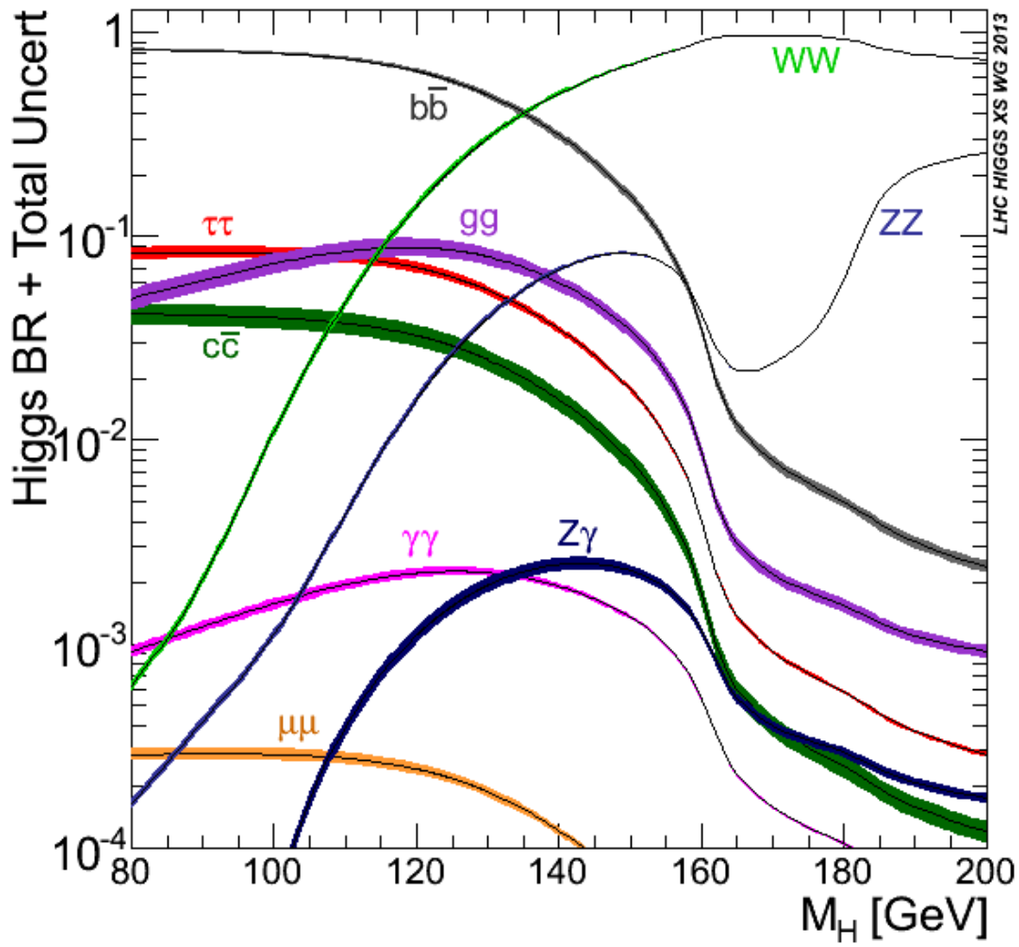


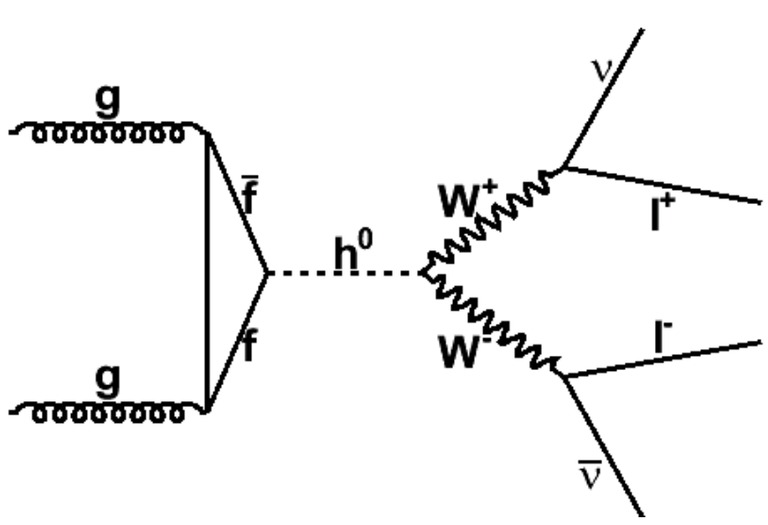


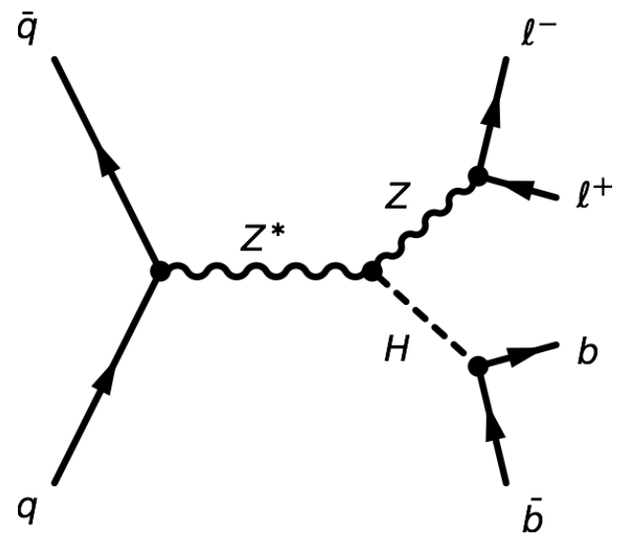
Higgs production and BR

http://mstwpdf.hepforge.org/plots/plots.html









Two analyses @ CDF: H→WW and H→bb

Reminder: basic analysis steps

1- Select the events online:
trigger (typically as loose as you can manage the detector readout rate)

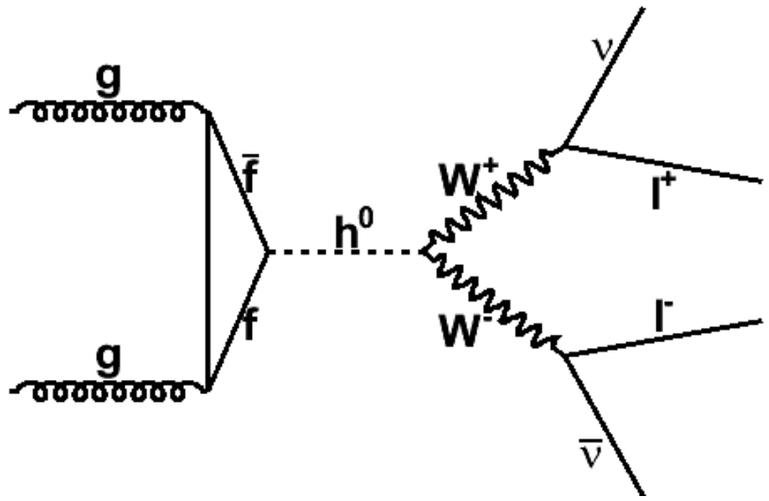
2- Offline event selection:
reduce the dataset to increase the purity of in terms of signal candidates

3- Categorise the events to maximise the analysis sensitivity
(large S/B, better signal resolution…)

4- Signal extraction / Statistical inference
estimate the amount of background in each class (cut and count / fit)
parameters estimation / hypothesis testing

H→WW

CDF L = 9.7/fb



$gg \rightarrow H \rightarrow WW^* \rightarrow l^+l^-\nu\bar{\nu}$,
where $l^\pm$ = e, μ or τ in the final states e^+e^-, $e^\pm\mu^\mp$ and $\mu^+\mu^-$.

Selections:

- Opposite-sign Base Selection
- Low Mll Base Selection
- Same-sign Base Selection (Z/WH →Z/WWW)
- Trilepton Base Selection (Z/WH →Z/WWW)

Selections based on the presence and the transverse momentum of the leptons + requirements on the MET

Channels

- A. 0-jet Analysis
- B. 1-jet Analysis
- C. 2-jet Analysis
- D. Low Mll Analysis
- E. Same-Sign Analysis
- F. Trilepton Analyses

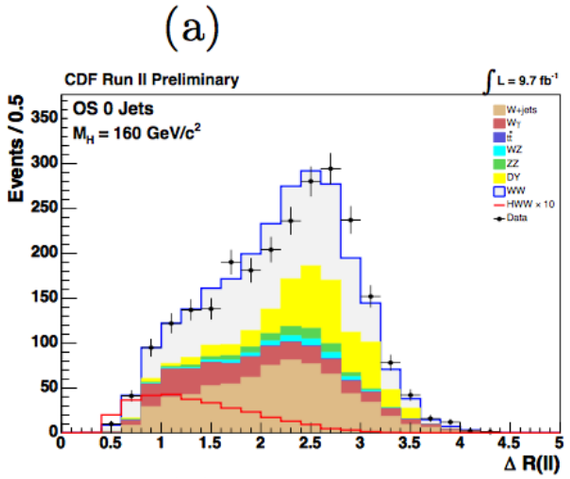
Background:

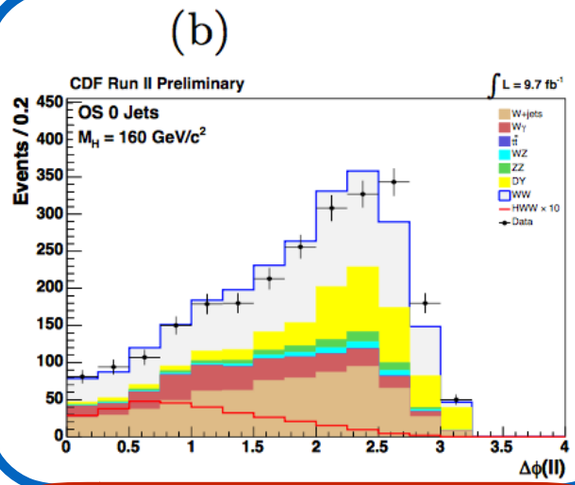
tt,
WZ, ZZ
Z+jets
Z γ

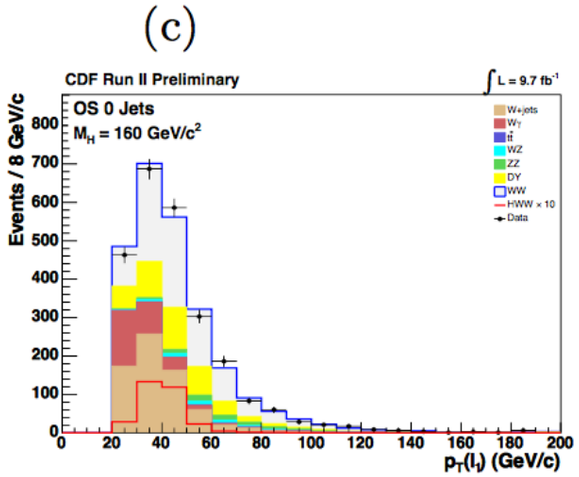
The analysis then needs to separate for each channel Signal from Background.
This is achieved by studying variables differently distributed for signal and background.

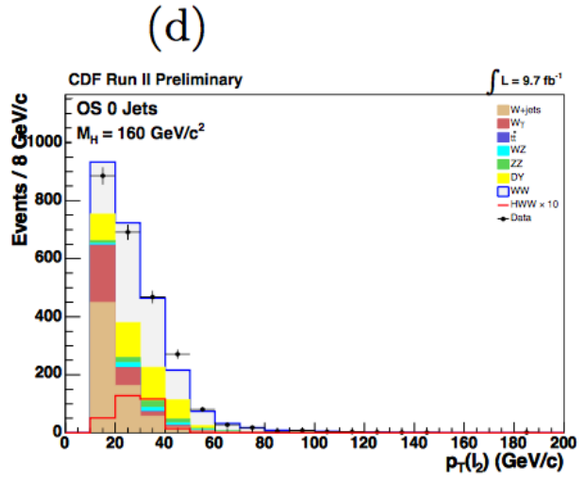
http://www-cdf.fnal.gov/physics/new/hdg/Results_files/results/hwwmenn_120224/HWW_97fb_public_note.pdf

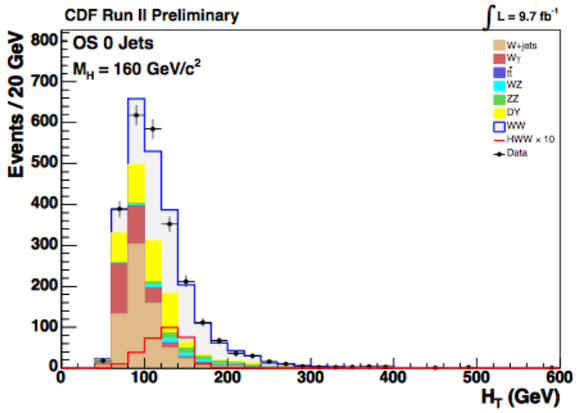
H→WW some discriminating vars

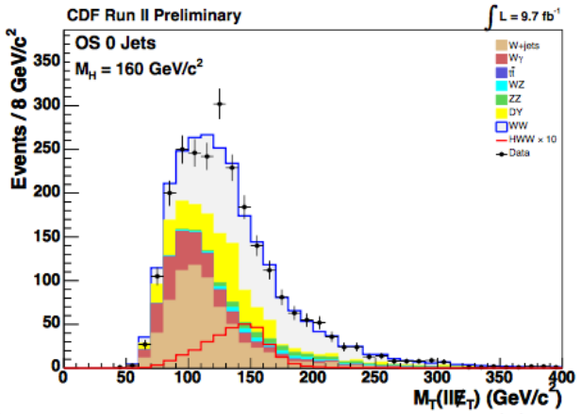


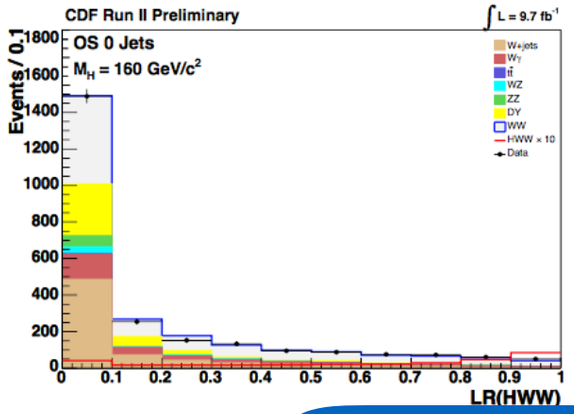


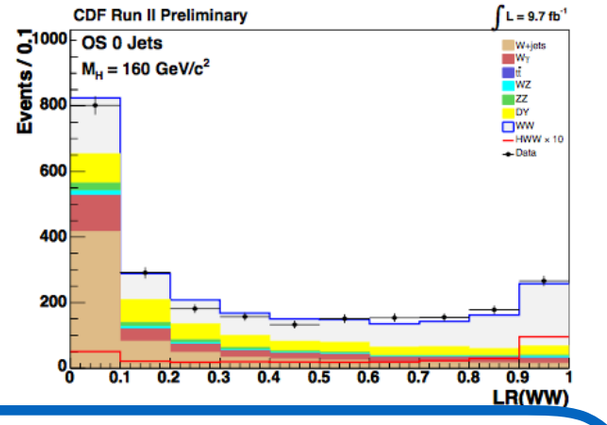






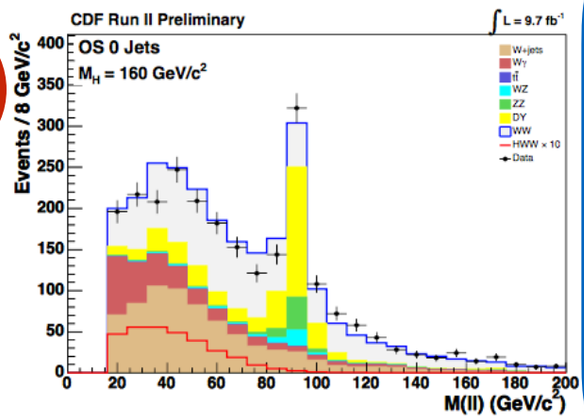


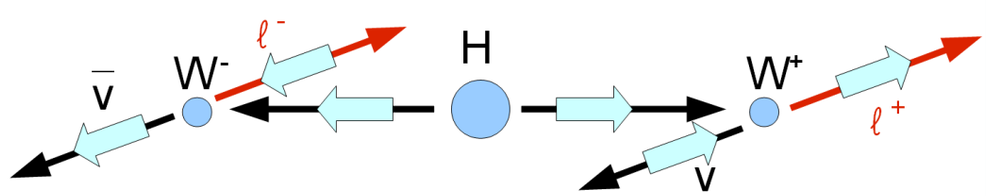




(i)

$$m_{\mathrm{T}}^2 = 2p_{\mathrm{T}}^{\ell\ell} \not{E}_{\mathrm{T}}^{\mathrm{miss}} \left(1 - \cos\Delta\phi(\ell\ell, \vec{\not{E}}_{\mathrm{T}}^{\mathrm{miss}})\right)$$





$\Delta\phi_{l+l-} \rightarrow 0$

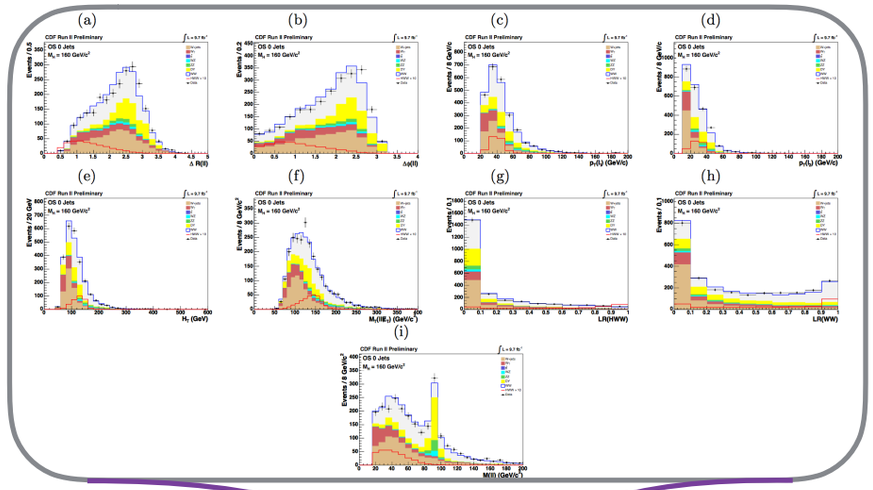
H→WW Neural Net

In the past lecture we've seen the Likelihood Ratio as a way (under some conditions Neyman-Person) to optimally separate H_0 from H_1

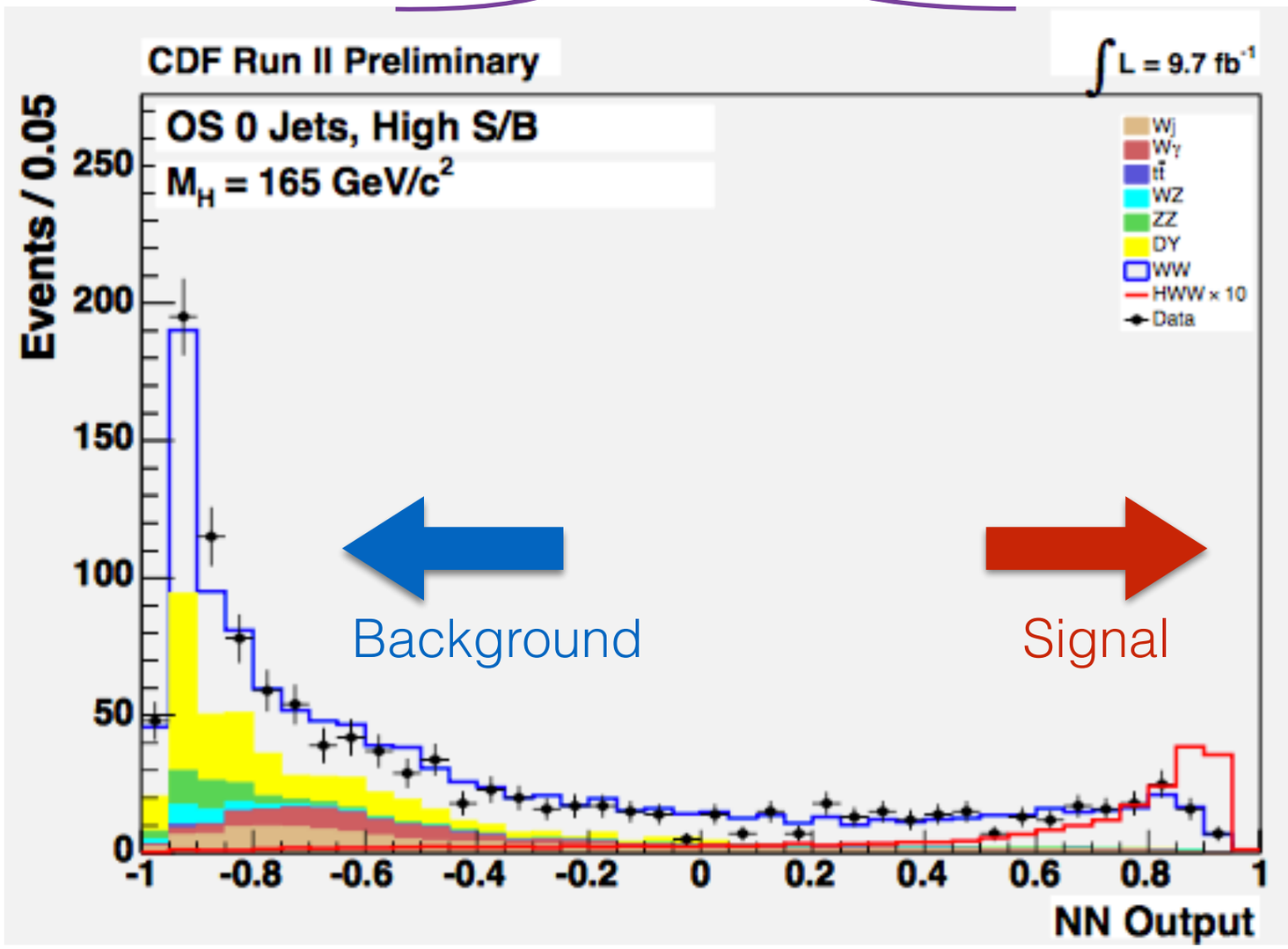
$$\frac{g(\mathbf{t}|H_0)}{g(\mathbf{t}|H_1)} > c.$$

In practice: it is usually difficult to determine c, because one needs to know the complete joint pdfs of x : g(t|H0) and g(t|H1). In general it's challenging to write a pdf with the proper correlations !

In practice: easier to use test statistics based on multivariate methods (MVA).
Here we look at one type of MVA:
Neural Networks



Neural Network



Tom Friedman, Open Black Box (2006)

Neural networks

 A non HEP application https://www.youtube.com/watch?v=VwwB6xcx8Wg

The (very) big picture

Artificial Intelligence — It studies the systems that perceive an environment

Machine Learning

Machine learning is the technology of getting computers to act without being explicitly programmed.

Supervised Learning

"Classification problems"

Unsupervised Learning

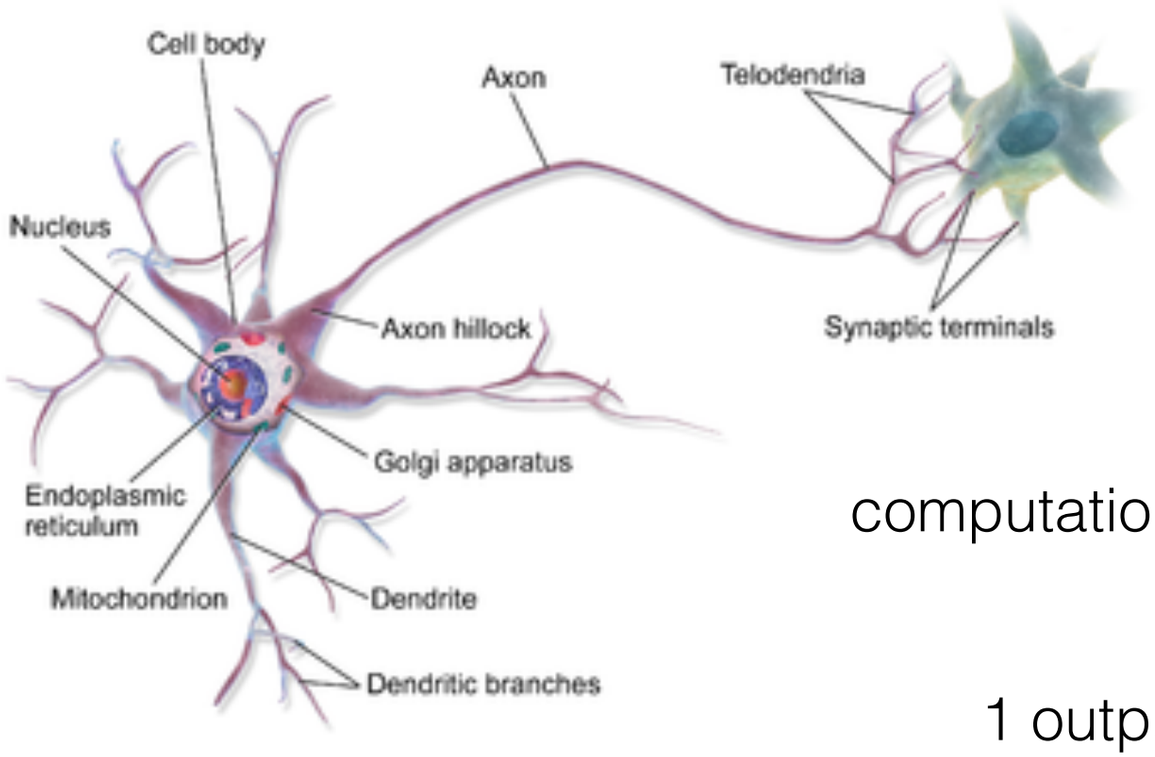
"Clustering"

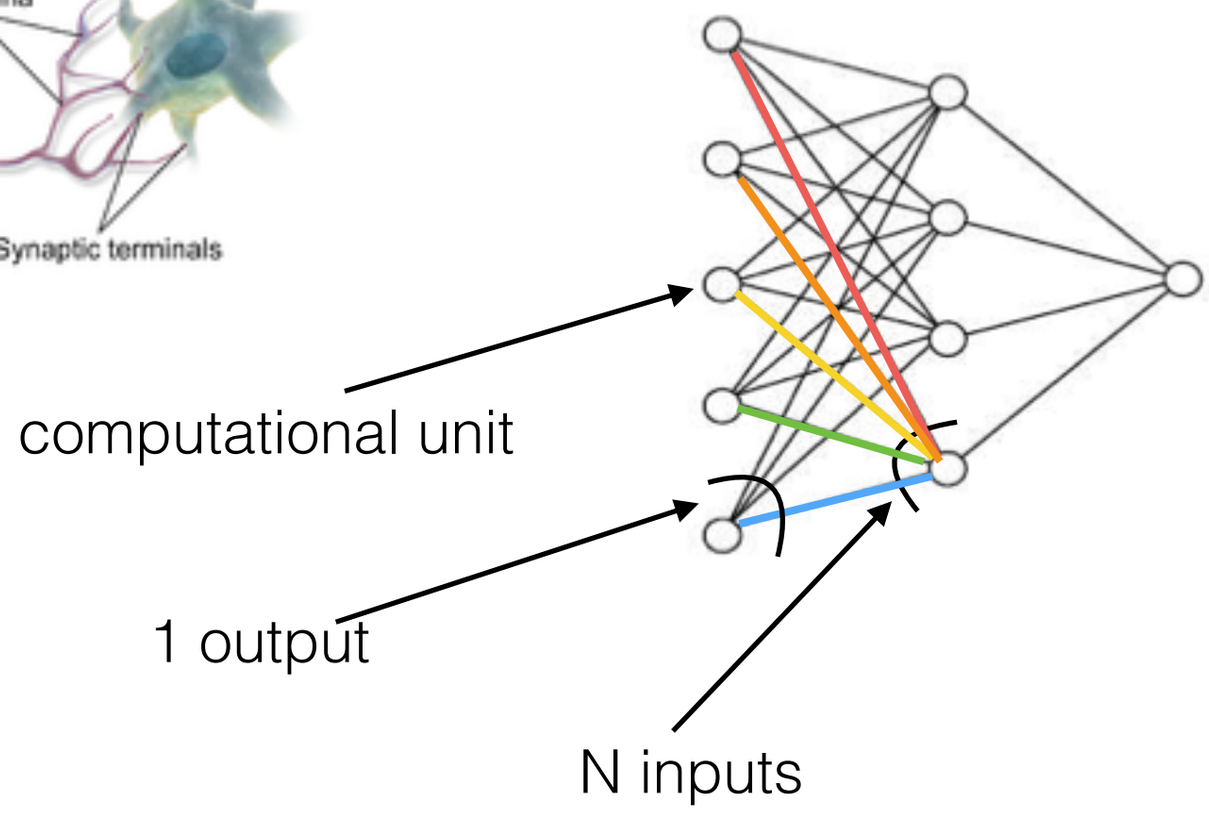
Supervised: you instruct the algorithm using a sample where you give the correct classification
Unsupervised: you let the algorithm find out what are the characteristics of the data and define the classes

Classification: you assign each member of a data samples to a discrete number of categories
Regression: you assign each member of a data samples to a continuous value

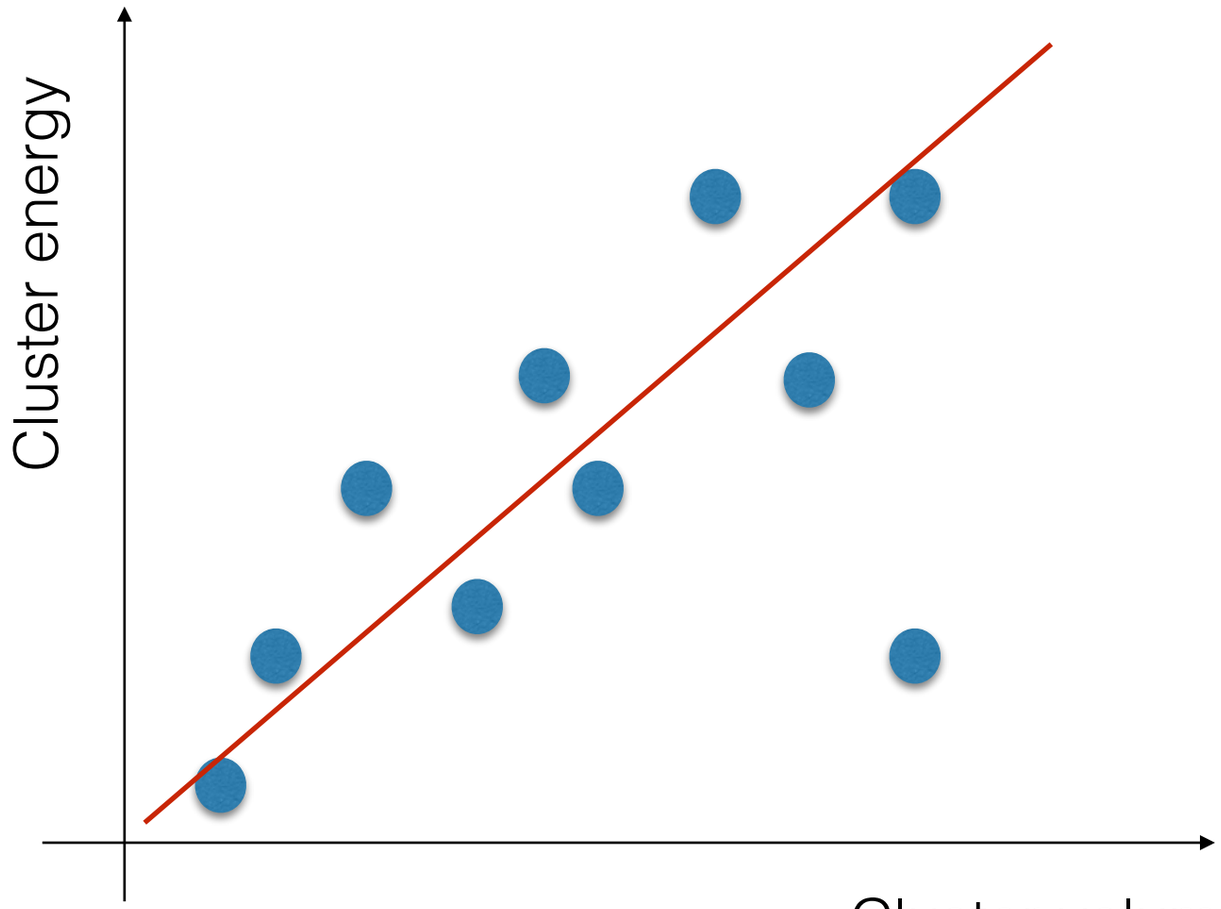
Artificial neural networks

Neural network come from the idea of the ~'70s to model the neurological processes with a complex network of units each of which is capable to perform only a simple operation.





Step by step to NN: Linear Regression



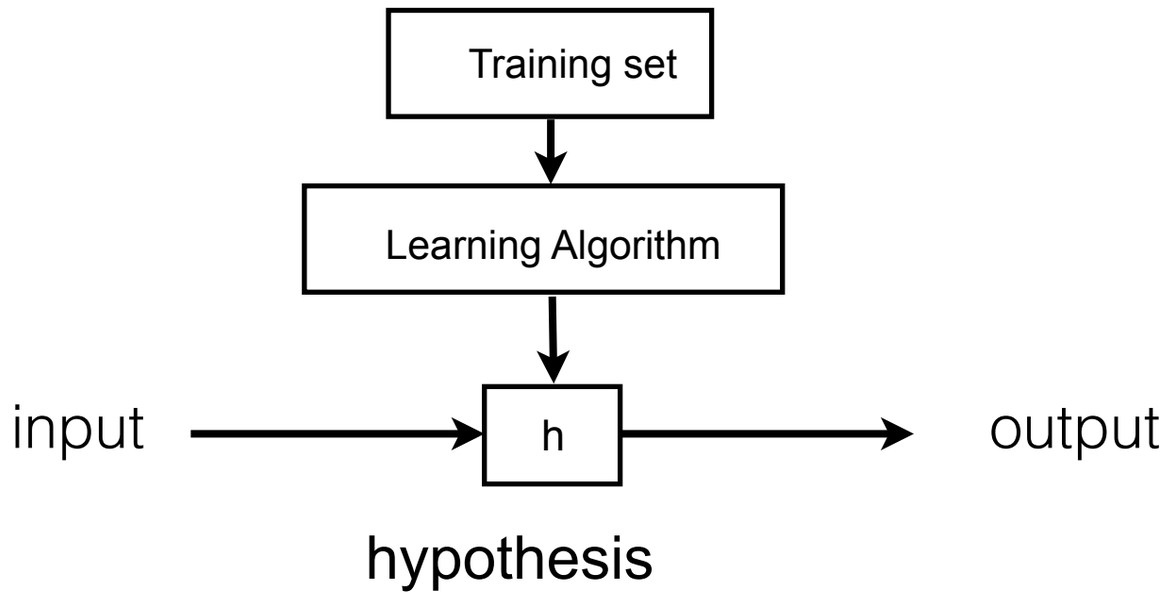
N = 1 = number of input variables
m = 10 = number of training samples

$$h_\theta(x) = \theta_0 + \sum_{j=0}^{N} \theta_j x_j$$

Simple χ^2 minimization:
- in one dimension is a line
- in N dimension is a N-1 hypersurfece

"Two steps" train / apply

Step by step to NN



Linear Regression

$$h_\theta(x) = \theta_0 + \sum_{j=0}^{N} \theta_j x_j$$

$$\min_\theta \frac{1}{2m} \sum_{i=0}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

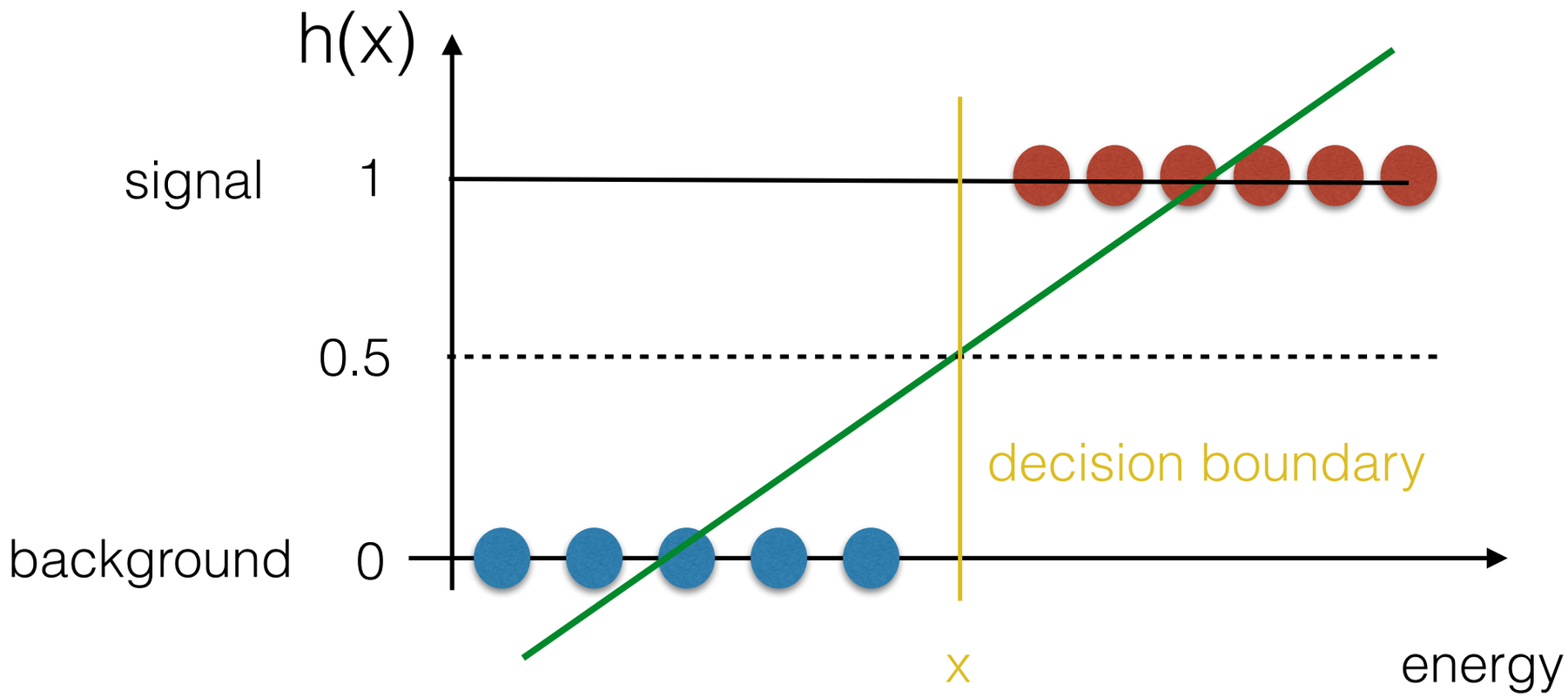
cost function

N = number of input variables
m = number of training samples
x_j = input variable
y_j = output variable
$(x^{(i)},y^{(i)})$= training example

Linear regression for classification

Suppose you want to separate two classes : 

The easiest way to build a classifier



If h(x) > 0.5 predict 1
If h(x) < 0.5 predict 0

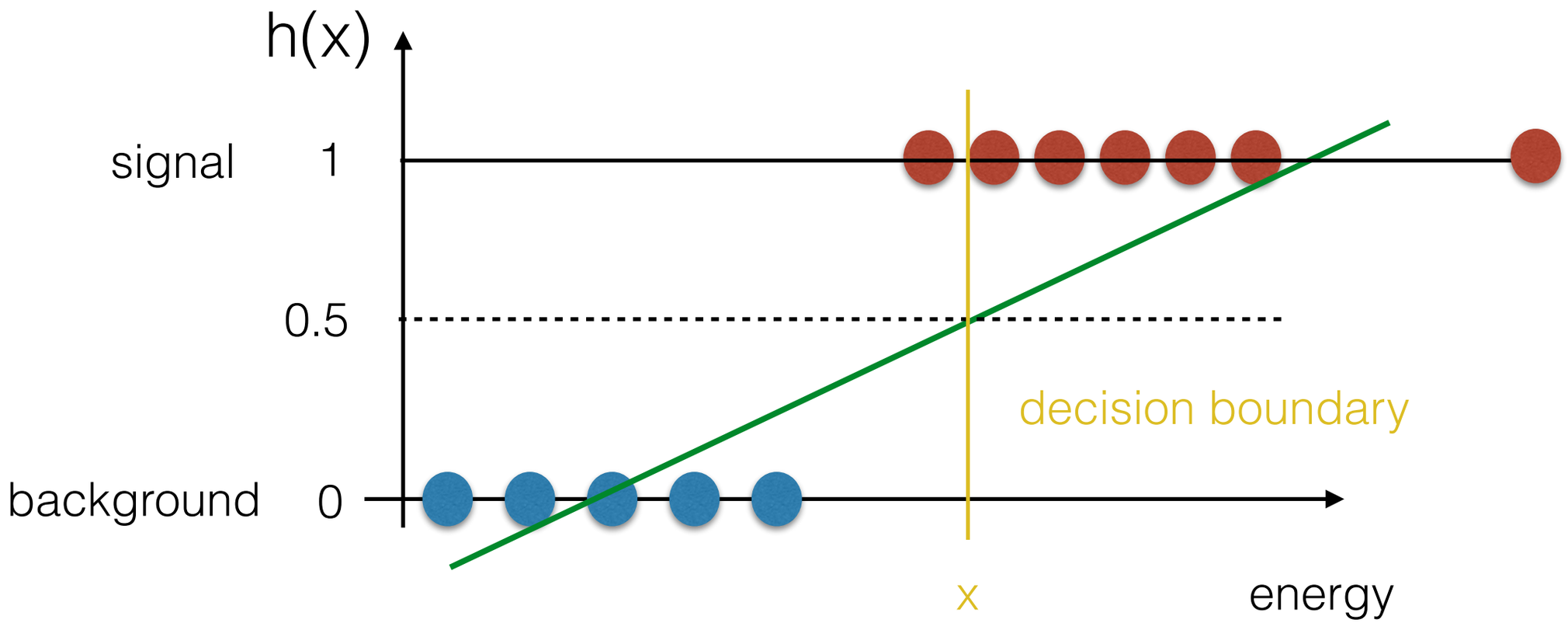
So if I measure energy above/below x
I will flag it as signal/background

It works but it has a few issues:
1 - the prediction goes outside [0,1]

Linear regression for classification

Suppose you want to separate two classes : 

The easiest way to build a classifier



If h(x) > 0.5 predict 1
If h(x) < 0.5 predict 0

So if I measure energy above/below x
I will flag it as signal/background

It works but it has a few issues:
1 - the prediction goes outside [0,1]
2 - adding obvious examples can make things worse

Logistic regression

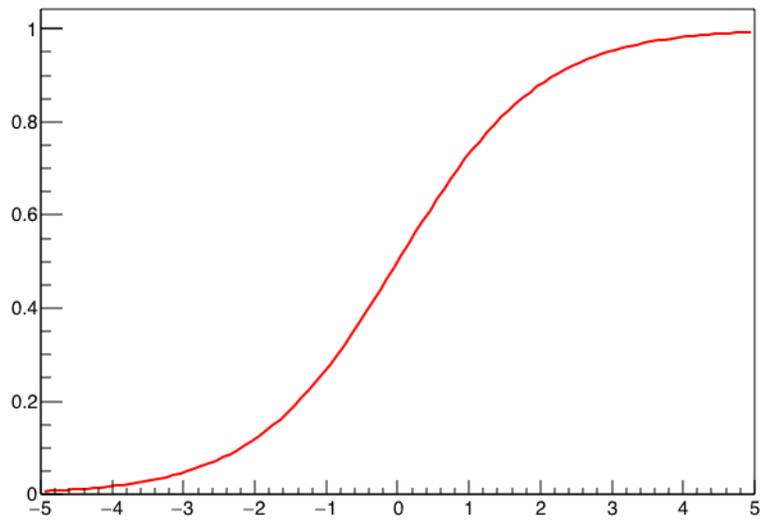
(It's an historical name it is used for classification problems)

We change the model from a linear one to a non-linear model

$$\theta^T x \rightarrow g(\theta^T x)$$

By far the most common transformation is the sigmoid or logistic function

$$g(z) = \frac{1}{1+e^{-z}}$$



The prediction is given as: $g(\theta^T x) > 0.5$ signal

$g(\theta^T x) < 0.5$ background

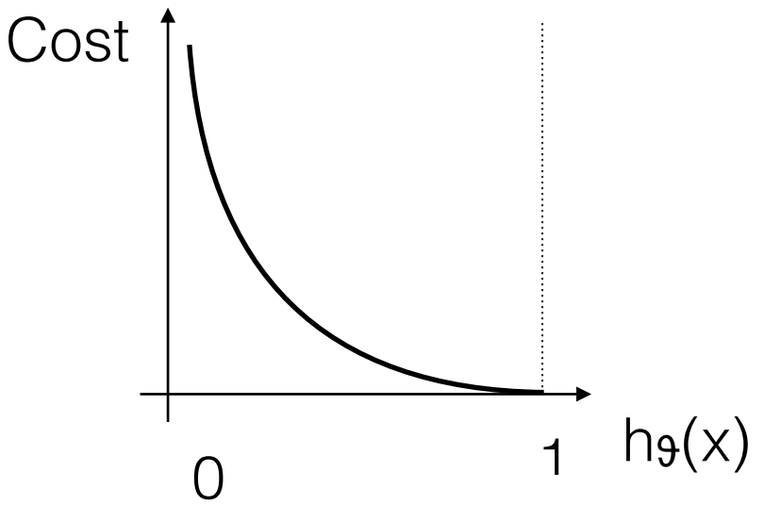
This simple transformation fixes the two problems encountered by the linear regression

Logistic regression: training

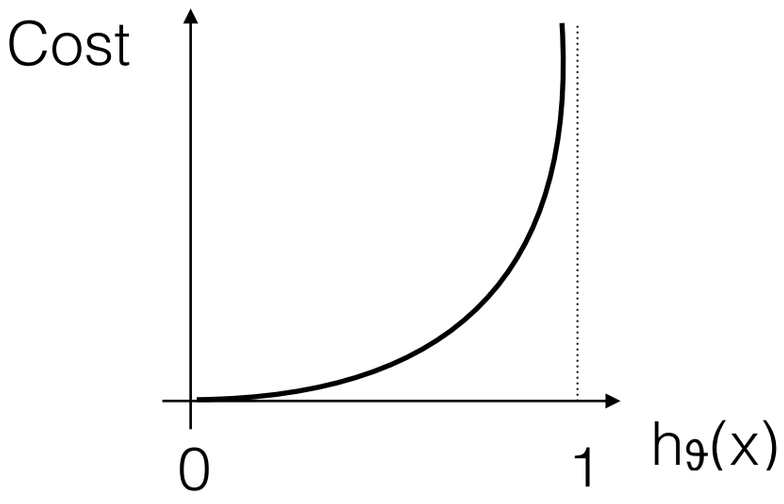
Training means to find the parameters of the model.
To have a convex function we define the cost as

$$\text{Cost}\,(h_\vartheta(x),y) = \begin{cases} -ln(h_\theta(x)) & \text{if y= 1} \\ -ln(1-h_\theta(x)) & \text{if y= 0} \end{cases}$$

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \ln h_\theta(x^{(i)}) + (1-y^{(i)}) \ln(1-h_\theta(x^{(i)}))\right]$$



It the model predicts 1 and the output is 1 the cost is 0
It the model predicts 0 and the output is 1 the cost is ∞

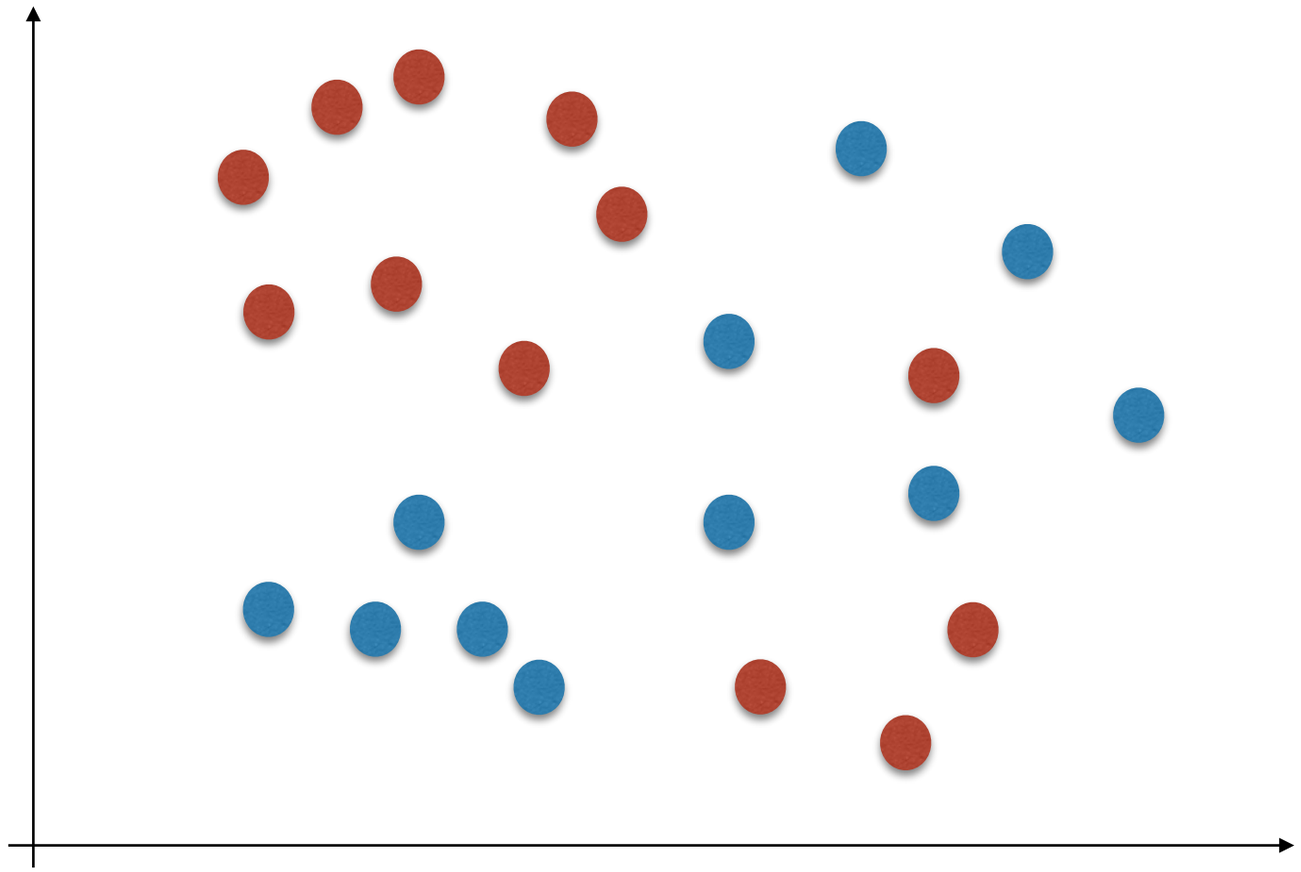


It the model predicts 1 and the output is 0 the cost is ∞
It the model predicts 0 and the output is 0 the cost is 0

The training will consist again in finding the weights ϑ

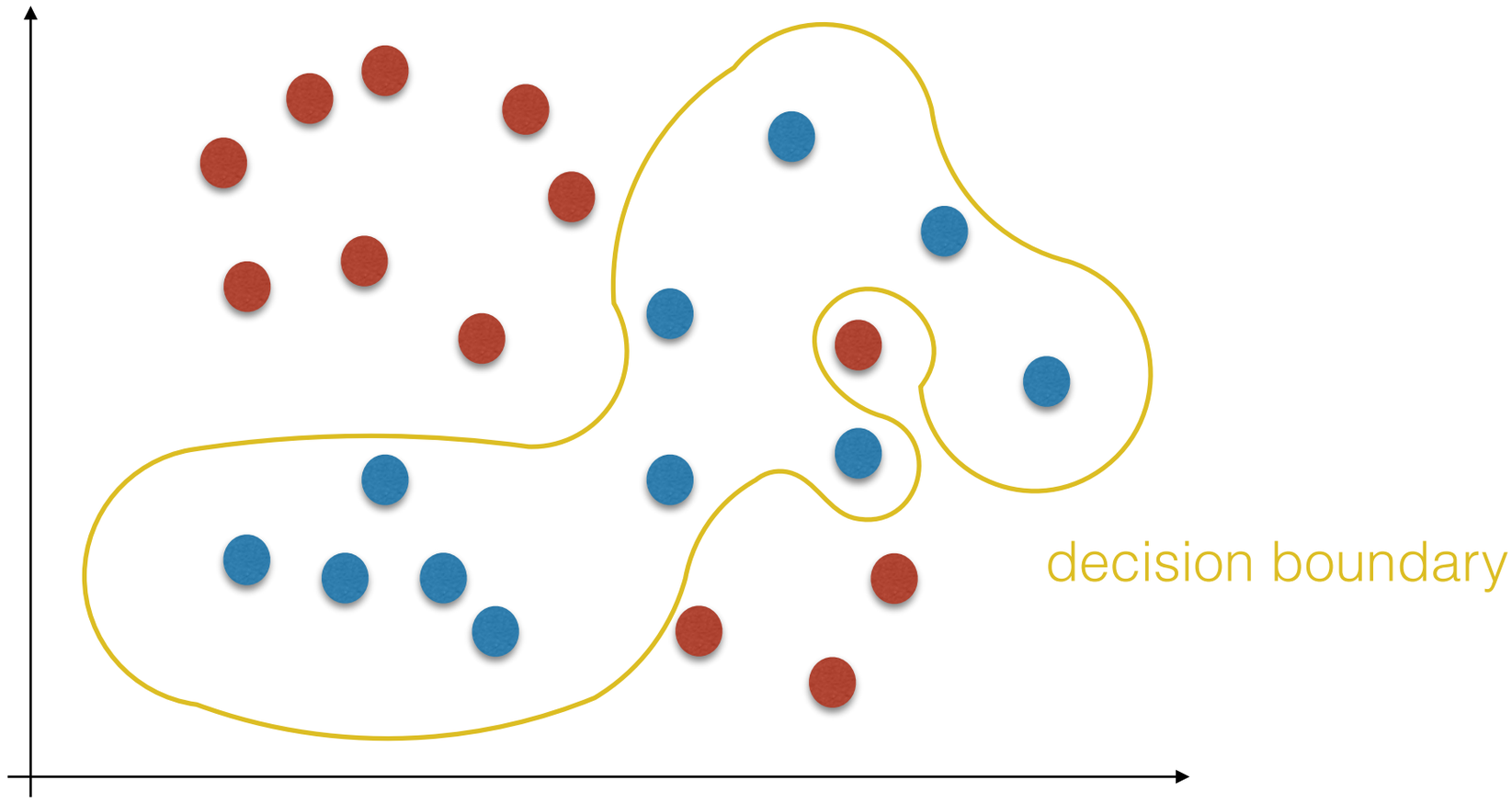
Networks

Suppose now you have a more complicated classification problem:



Networks

Suppose now you have a more “non linear classification” problem:



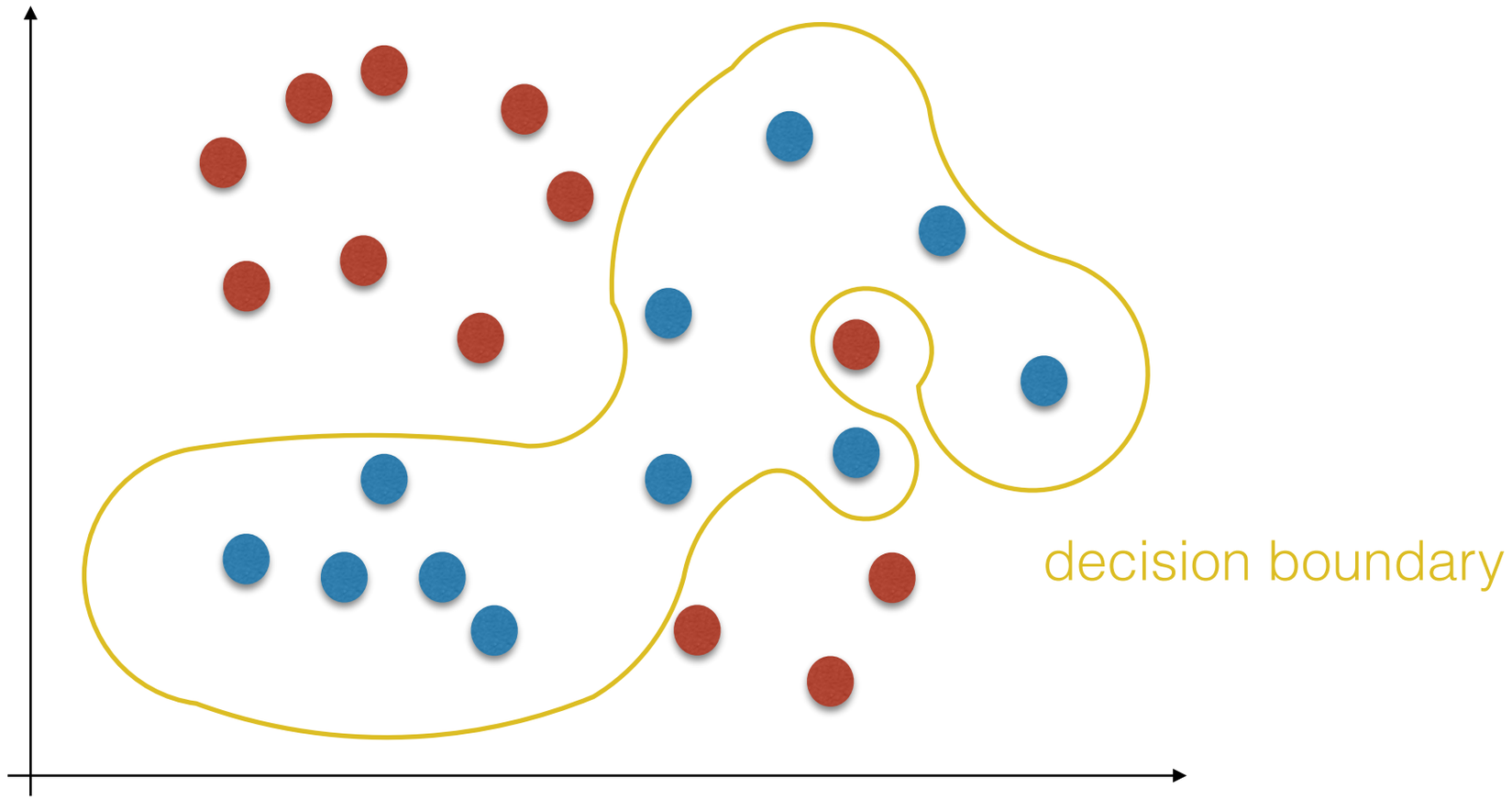
You can use higher order polynomials as model in input to the logistic regression.

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2 + \ldots)$$

The higher the order of the polynomial the higher the more “non-linear” you can make the decision boundary. (In ML slang: the larger the order the lower the bias, the higher the variance) This approach doesn’t scale with large number of input variables

Networks

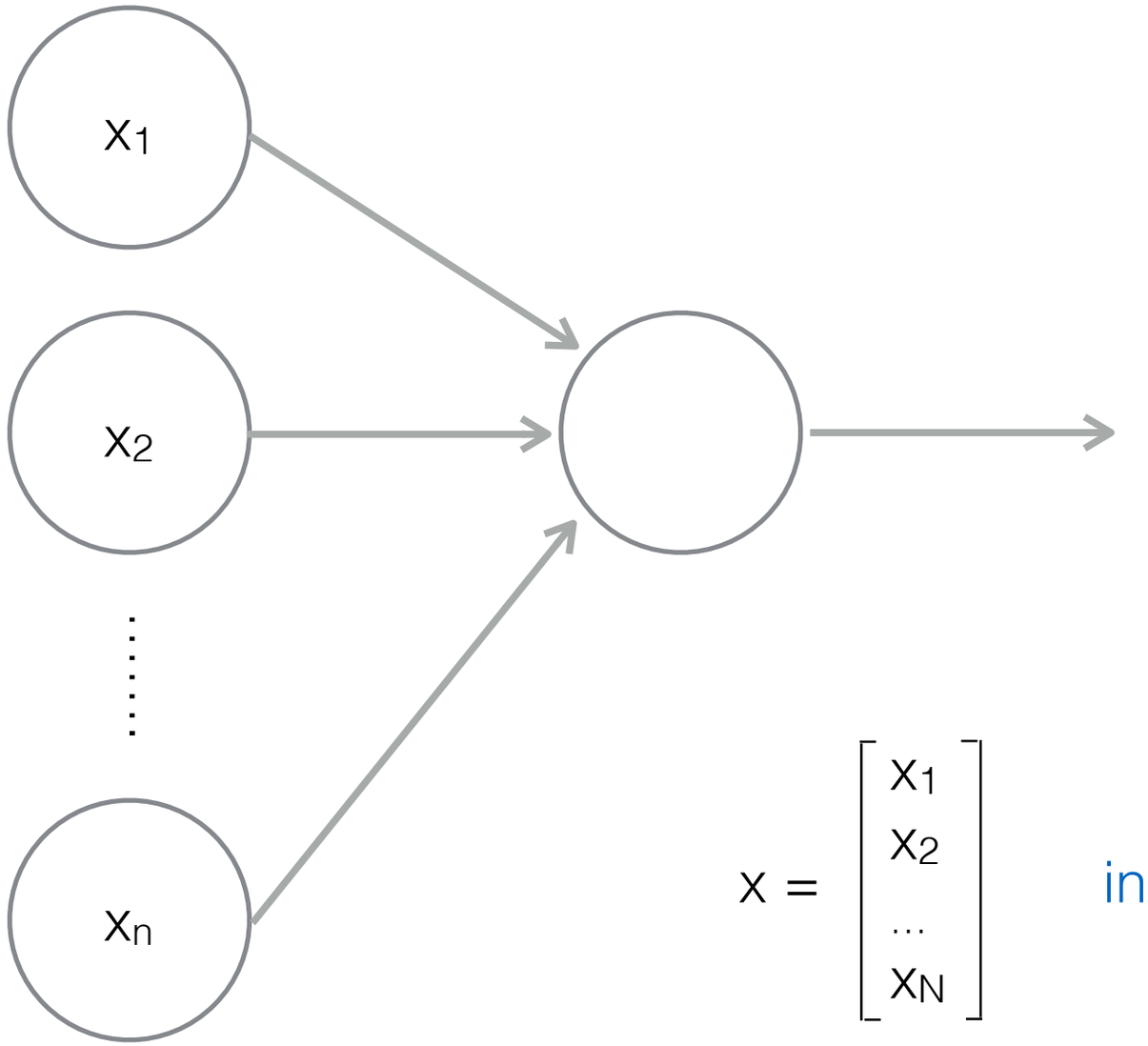
Suppose now you have a more “non linear classification” problem:



Another approach is to connect in a network several “logistic units” each with a very simple model.

$$h_\theta(\vec{x}) = \frac{1}{1 + e^{-\theta^T \vec{x}}}$$

Logistic unit (single layer perceptron)

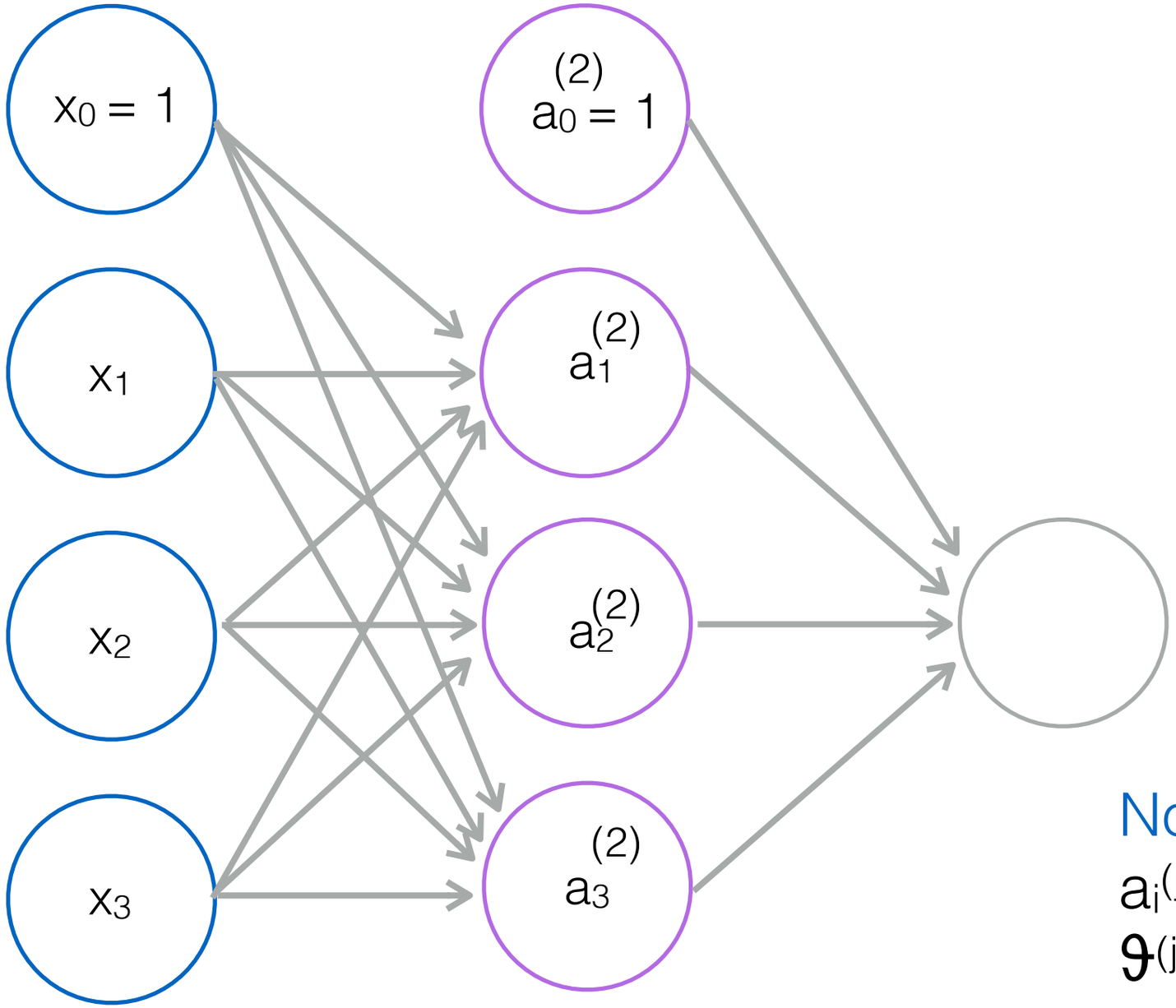


$$h_\theta(\vec{x}) = \frac{1}{1 + e^{-\theta^T \vec{x}}}$$

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_N \end{bmatrix}$$ input variables

$$\vartheta = \begin{bmatrix} \vartheta_1 \\ \vartheta_2 \\ \dots \\ \vartheta_N \end{bmatrix}$$ weights (parameters of the model)

Neural Network: model representation



Notation:

$a_i^{(j)}$ activation of unit i in layer j

$\Theta^{(j)}$ weight matrix for layer j-1

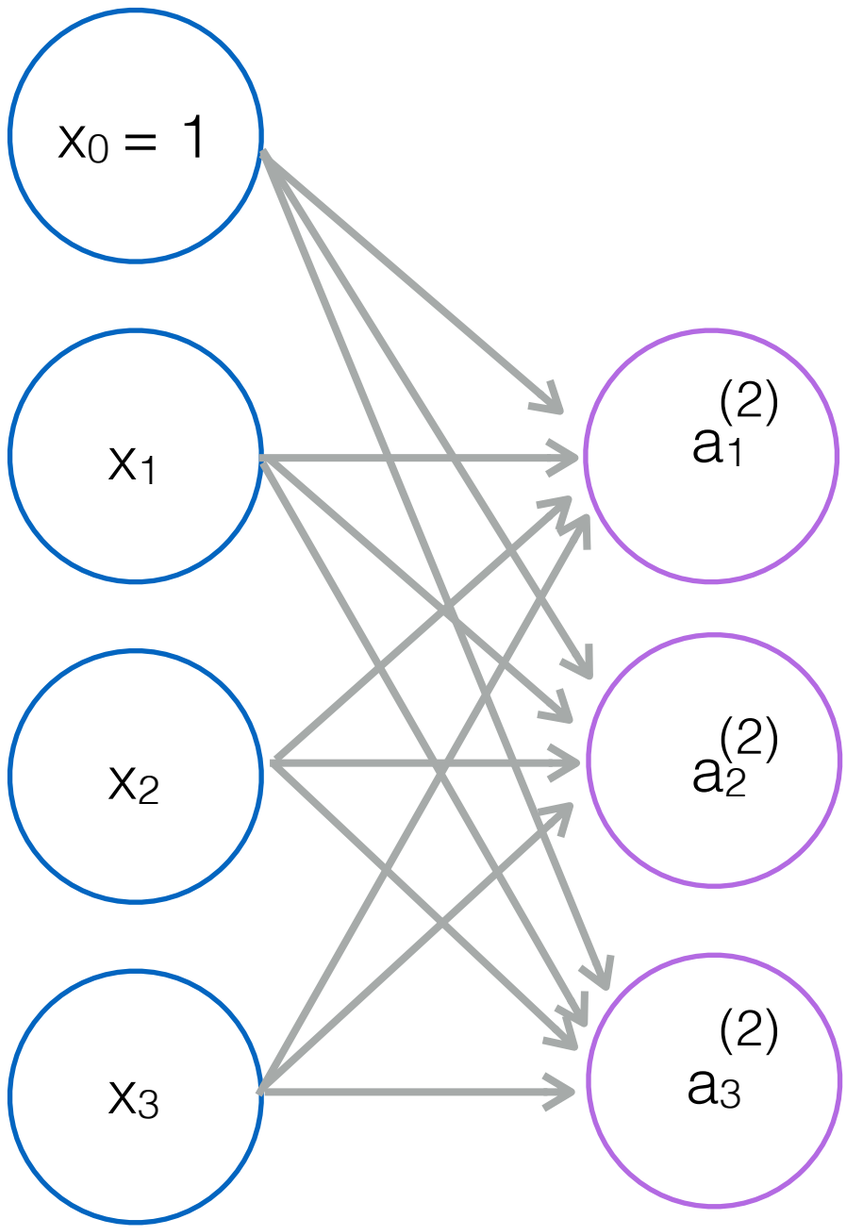
layer 1 — Input layer

layer 2 — hidden layer

layer 3 — output layer

Forward propagation

From the input to the output layer



$$a_1^{(2)} = g(\theta_{10}^{(1)} x_0 + \theta_{11}^{(1)} x_1 + \theta_{12}^{(1)} x_2 + \theta_{13}^{(1)} x_3)$$

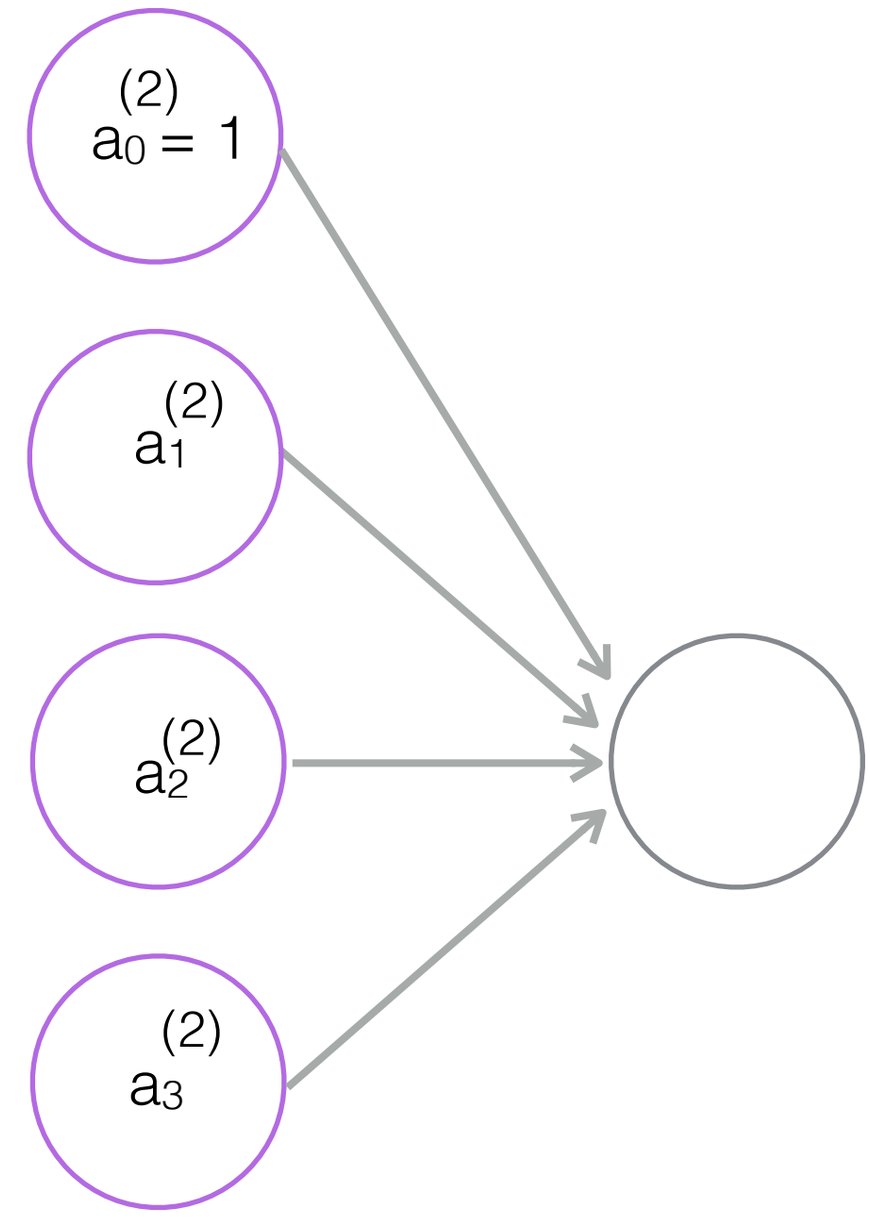
$$a_2^{(2)} = g(\theta_{20}^{(1)} x_0 + \theta_{21}^{(1)} x_1 + \theta_{22}^{(1)} x_2 + \theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\theta_{30}^{(1)} x_0 + \theta_{31}^{(1)} x_1 + \theta_{32}^{(1)} x_2 + \theta_{33}^{(1)} x_3)$$

$$\theta^{(1)} \in \mathbb{R}^{3\times 4}$$

Forward propagation

From the input to the output layer



$$h_\theta(x) = a_1^{(3)} = g(\theta_{10}^{(2)} a_0^{(2)} + \theta_{11}^{(2)} a_1^{(2)} + \theta_{12}^{(2)} x_2^{(2)} + \theta_{13}^{(2)} x_3^{(2)})$$

$$\theta^{(2)} \in \mathbb{R}^{1\times 4}$$

layer 2

hidden layer

layer 3

output layer

NN training

The main point of a neural network is that it will decide by itself how to build the activation units. We will not need to choose them to be linear, quadratic, cubic, etc…

Notation:
training set $\{x^{(i)},y^{(i)}\}$ i = 1,…,m

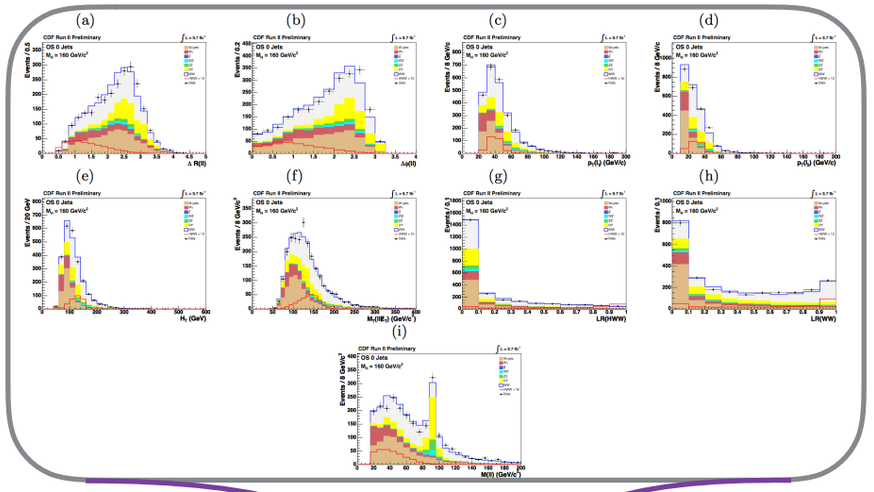
The cost function is the same as for the logistic regression:

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \ln h_\theta(x^{(i)}) + (1-y^{(i)}) \ln(1-h_\theta(x^{(i)}))\right]$$

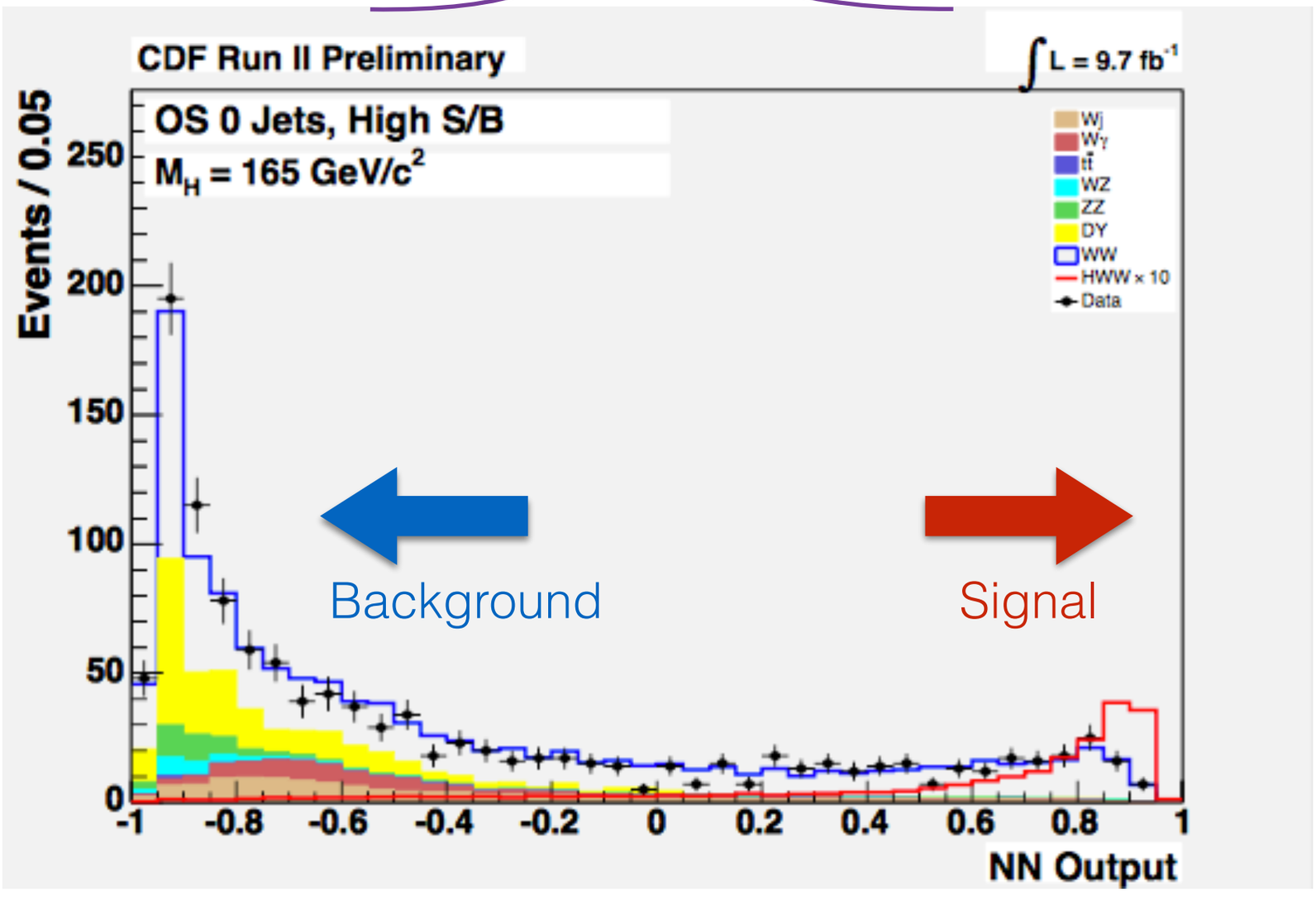
Need to minimize J(ϑ) as a function of the ϑ_{ij} at each layer $\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$

The most common algorithm to compute the derivatives is called "back-propagation" (which in practice tries to minimize the error at each node from the output to the first layer)

H→WW



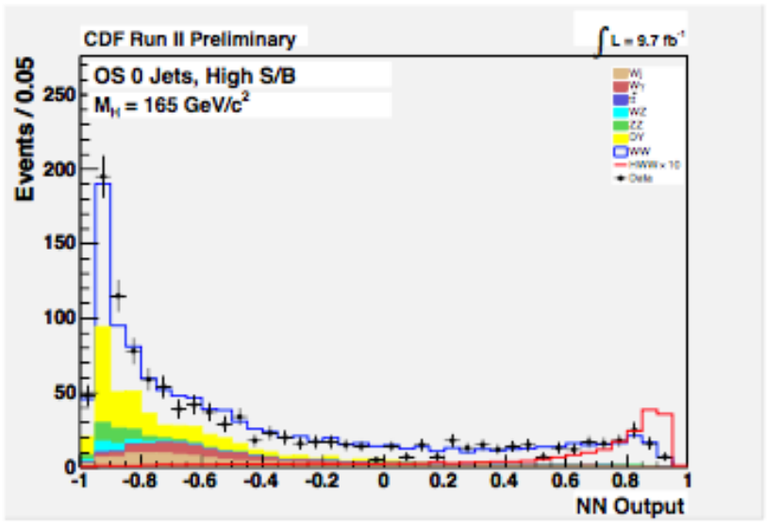
Neural Network

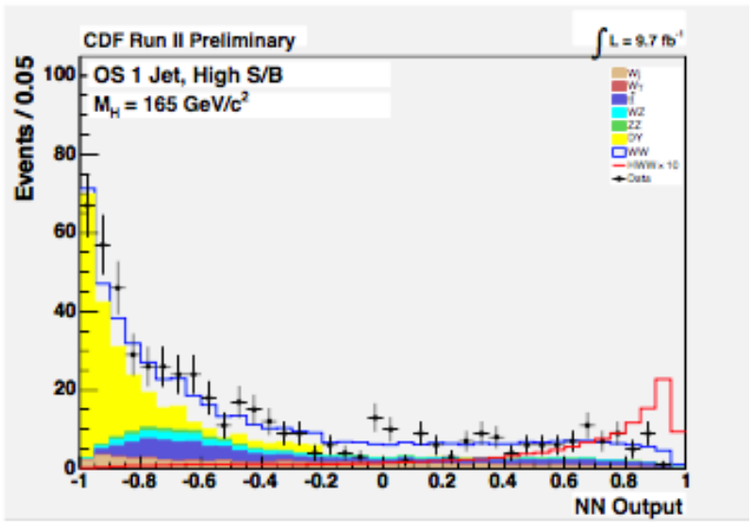


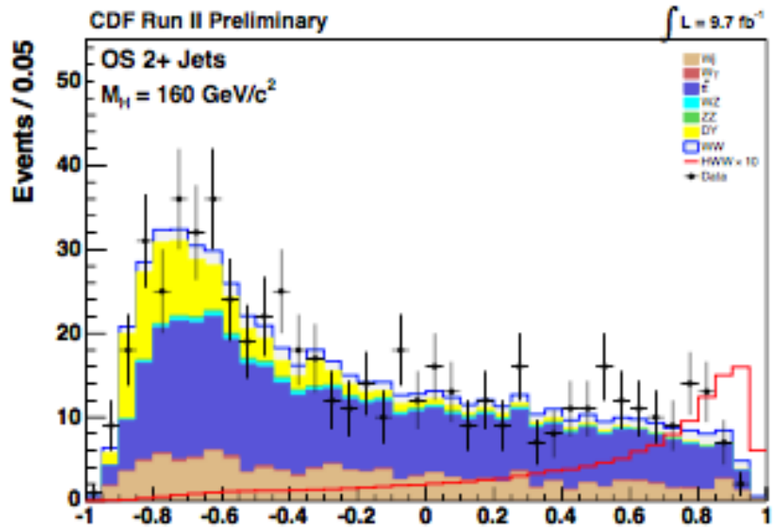
H→WW

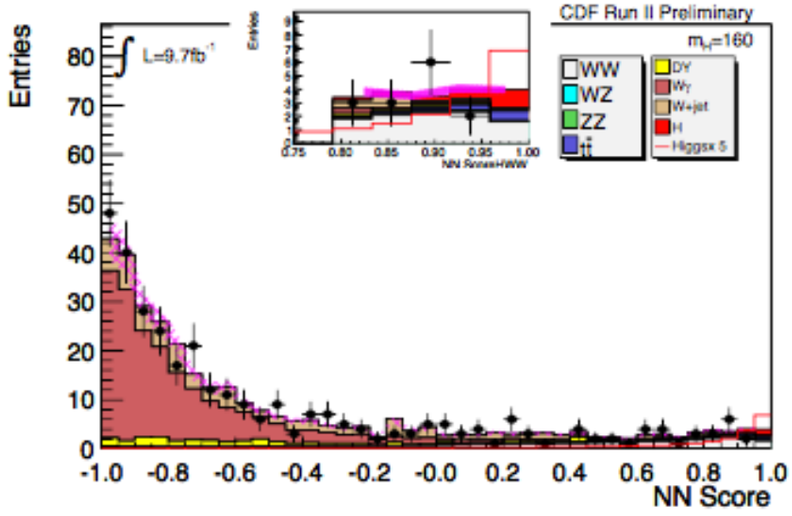
Each channel has a NN trained on a slightly different set of input variables

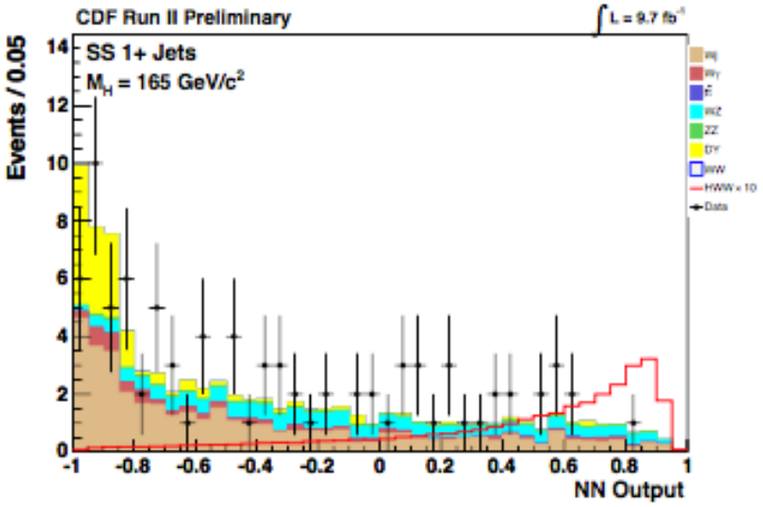
(…a bit cheating classes are also subdivided in sub-classes)

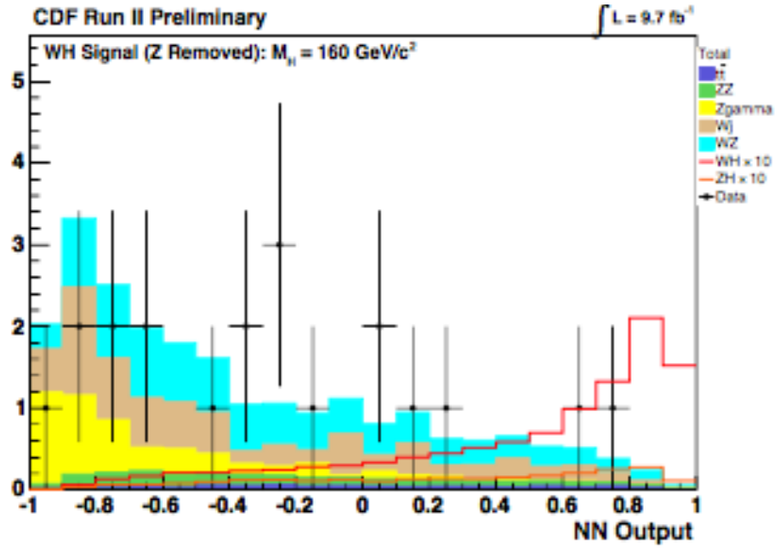






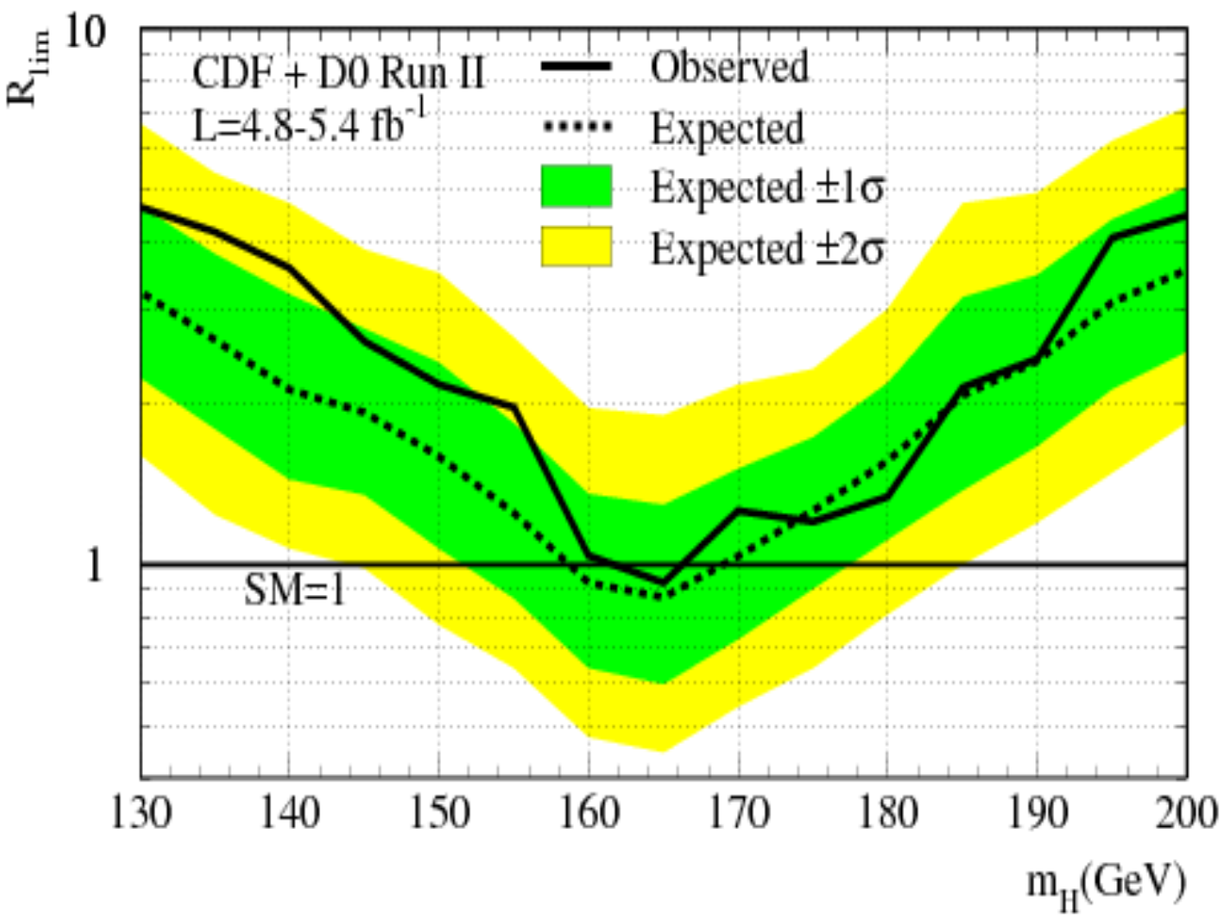




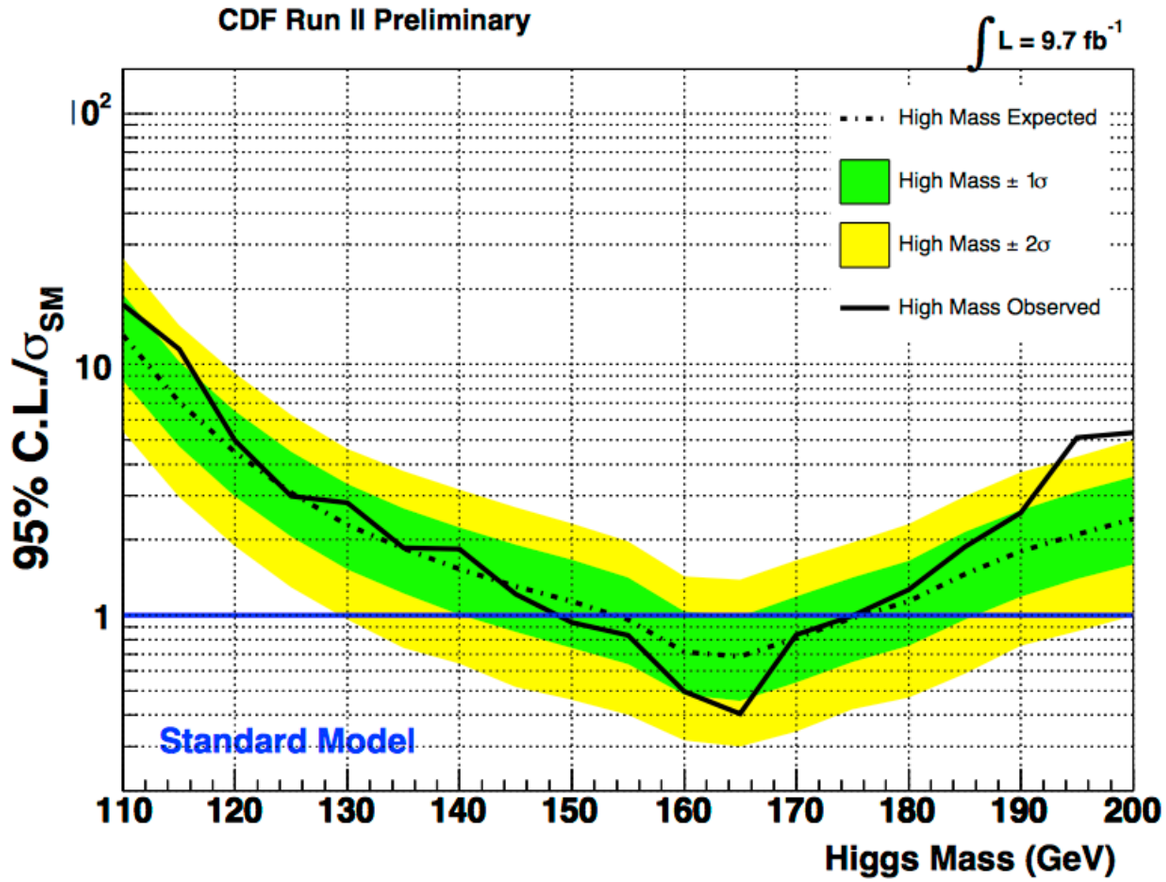


and all the results are combined in the end

H→WW



First exclusion post-LEP !



Final CDF result

http://www-cdf.fnal.gov/physics/new/hdg/Results_files/results/hwwmenn_120224/HWW_97fb_public_note.pdf