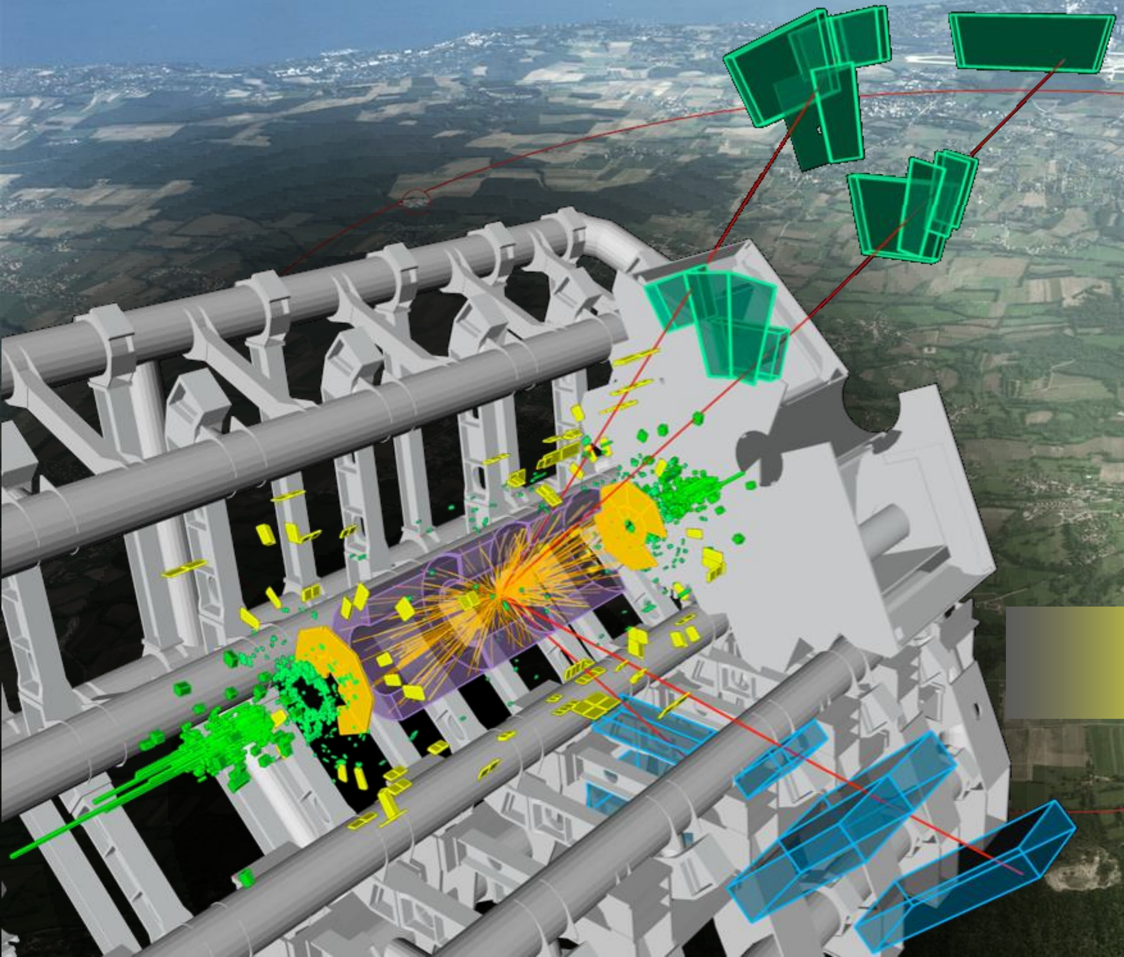
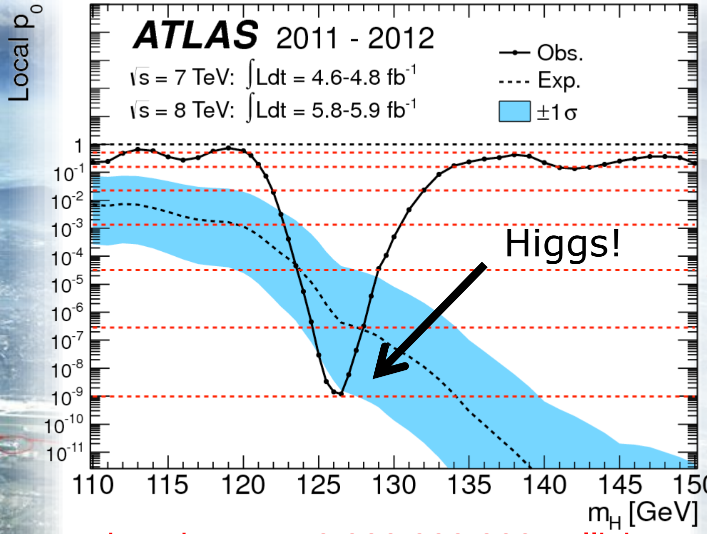
Discovering the Higgs – finding the needle in the haystack

W. Verkerke (Nikhef)

How do you find the Higgs boson?



Statistical analysis of all collisions

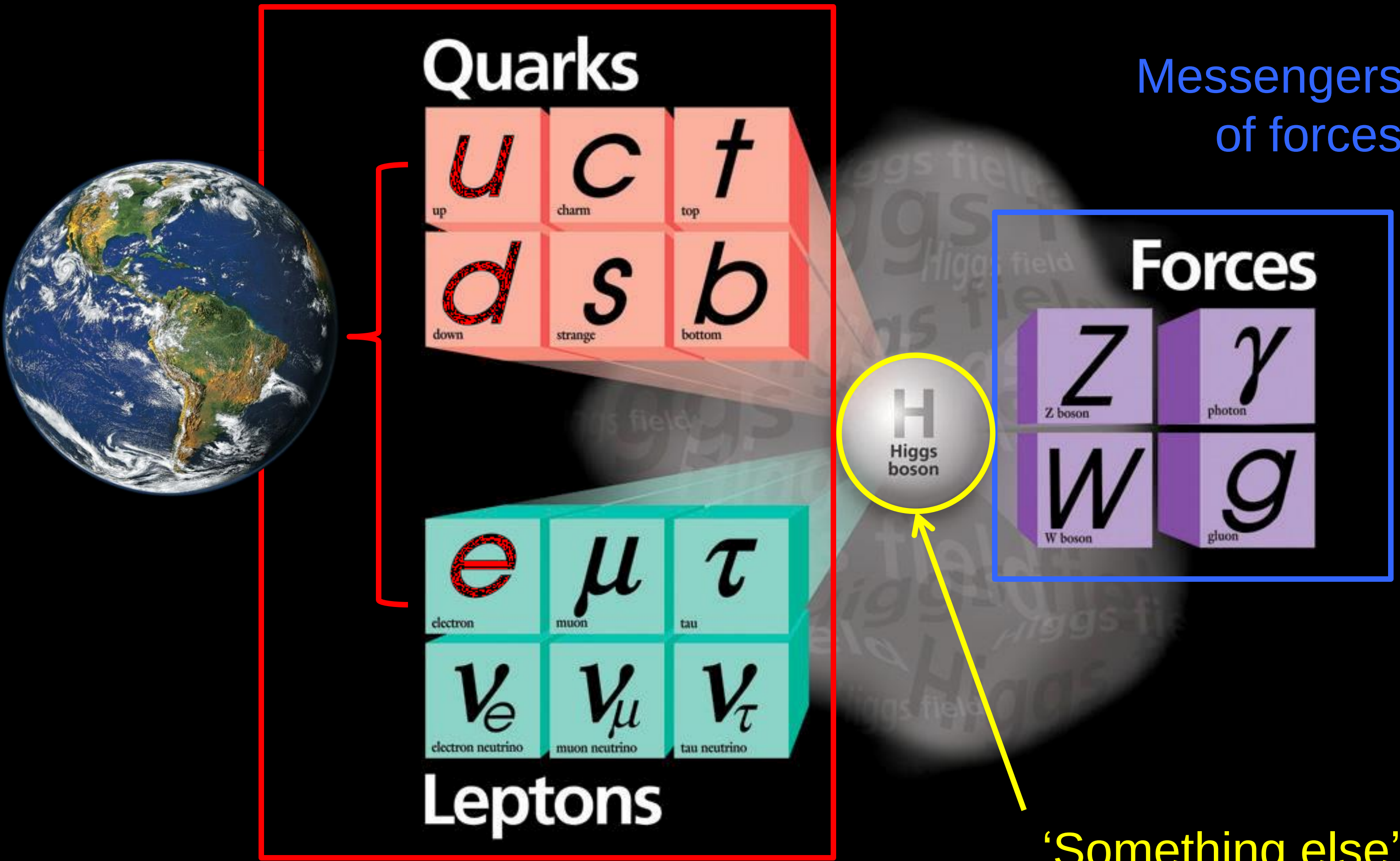


- Higgs in 1 on 10.000.000.000 collisions

- 5.000.000 Gb data
- 2.000.000.000.000 collisions

Particle physics: Elementary particles & Forces

Building blocks of matter

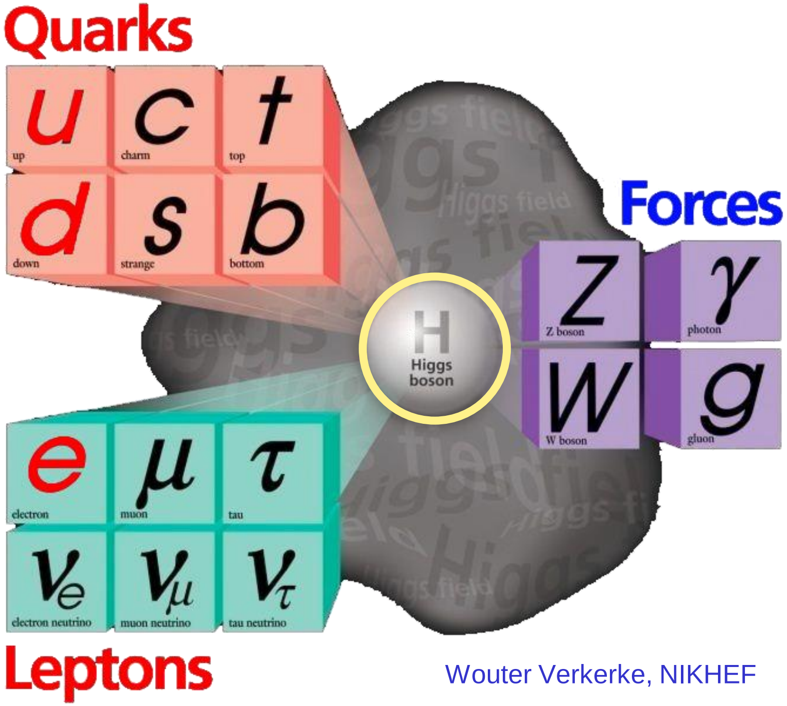


Messengers of forces

‘Something else’

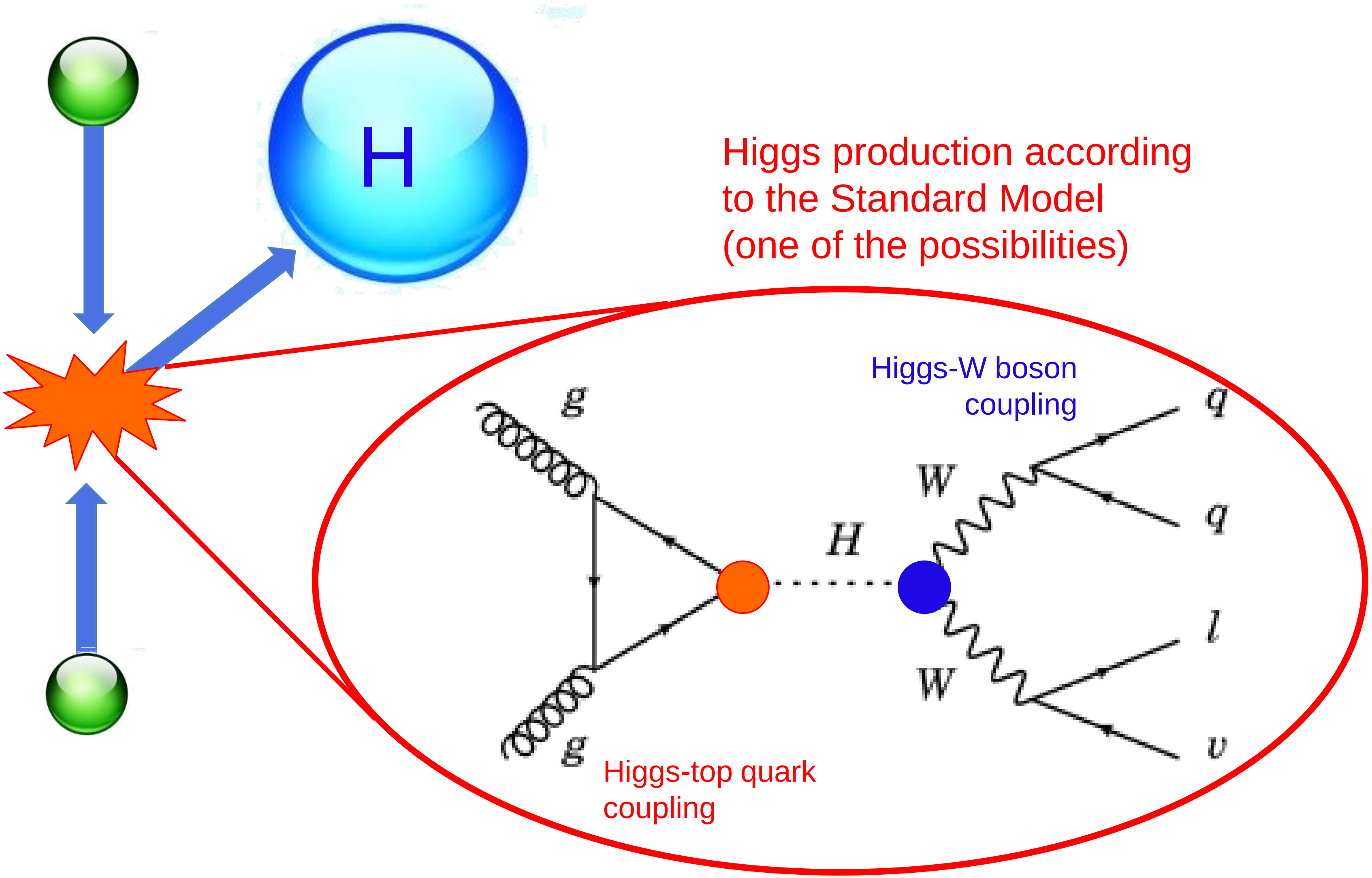
What do we know about the Higgs boson?

- Key ingredient of the Standard Model
- Special role: origin of mass of elementary particles
  - Space filled invisible with omni-present Higgs field.
    Mass of elementary particles is consequence of interaction of particles with this field
  - Large particle mass → strong coupling to Higgs field
    small particle mass → weak coupling to Higgs field
- Peter Higgs:
  field → particle
  - Particle manifestation of the Higgs field, with same properties as field
  - If you have access to Higgs particles you can directly measure coupling strength to other particles



Making a Higgs boson - theory

- Theory: if Higgs boson exists you can make it in high-energy particle collisions



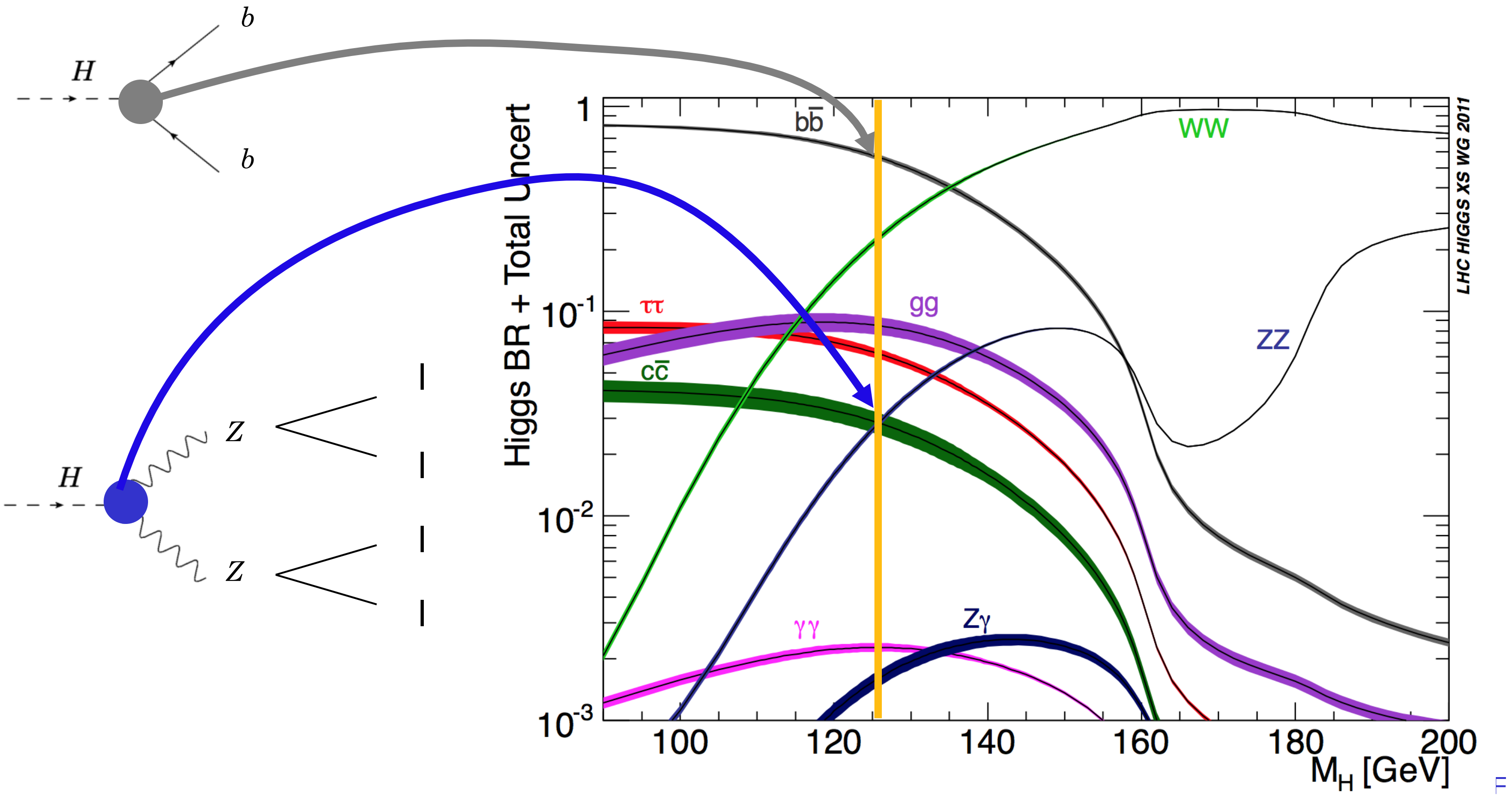
Higgs production according to the Standard Model (one of the possibilities)

Higgs-W boson coupling

Higgs-top quark coupling

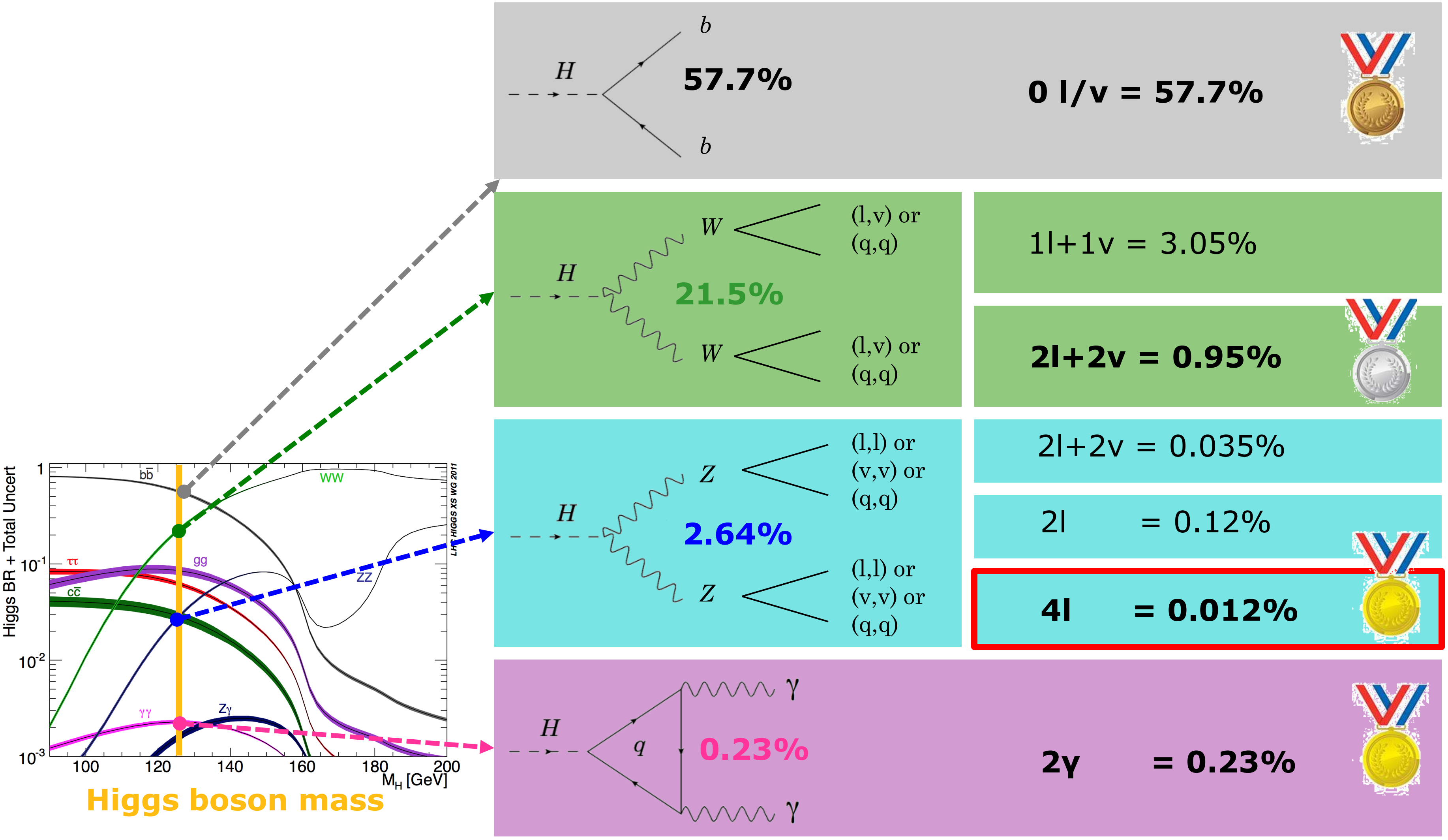
Other types of Higgs decays

- So far showed one decay (H → WW), but many other types of decays can happen.
  - Relative rate of ocurrence (and most promising channel) depend on mass of Higgs boson (which was a priori unknown, but we now know is 125 GeV)



What does a Higgs boson look like, and how often?

Experimental feasibility rank



Higgs boson mass

57.7%

0 l/v = 57.7%

21.5%

1l+1v = 3.05%

2l+2v = 0.95%

2.64%

2l+2v = 0.035%

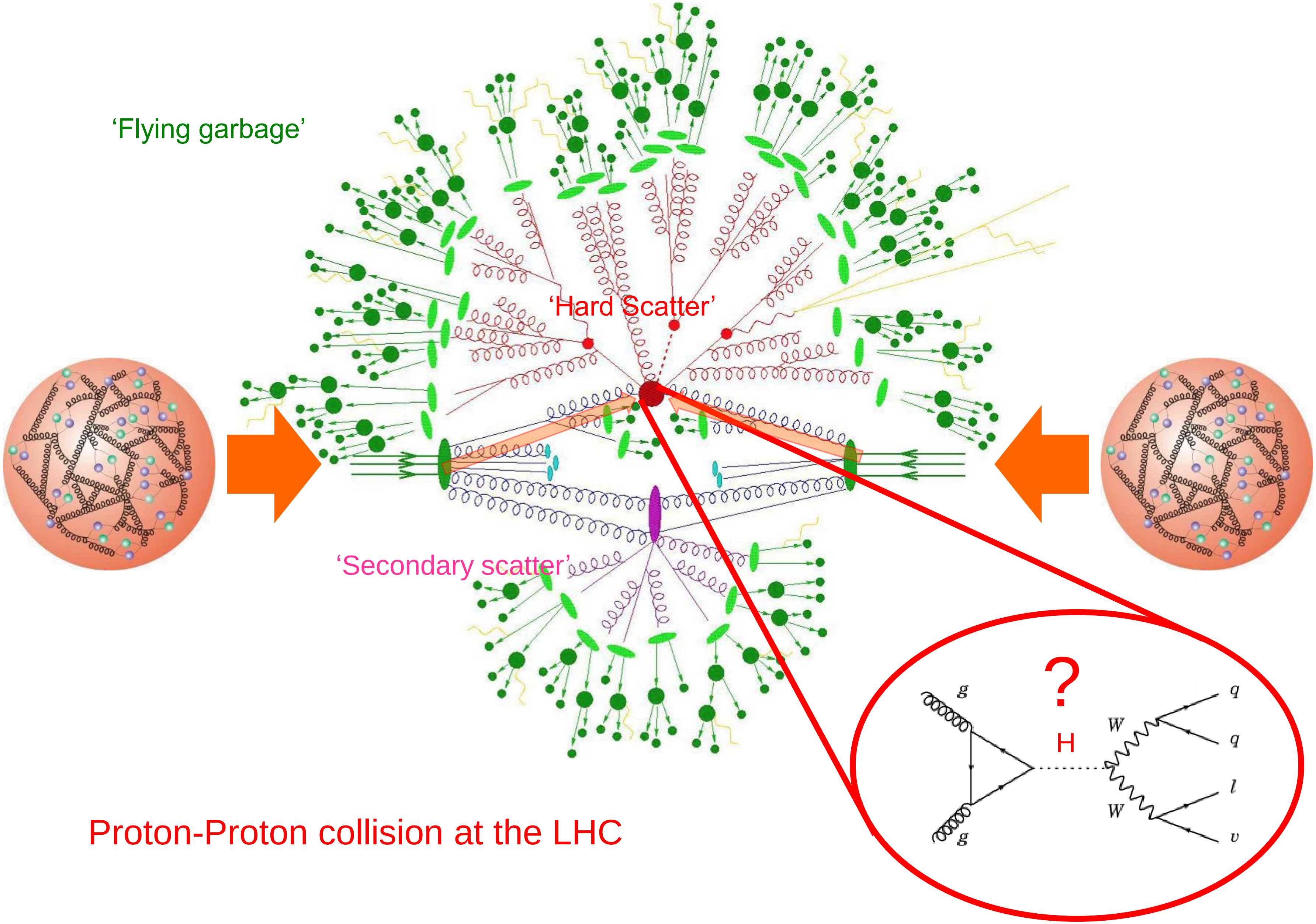
2l = 0.12%

4l = 0.012%

0.23%

2γ = 0.23%

A more accurate picture of what happens



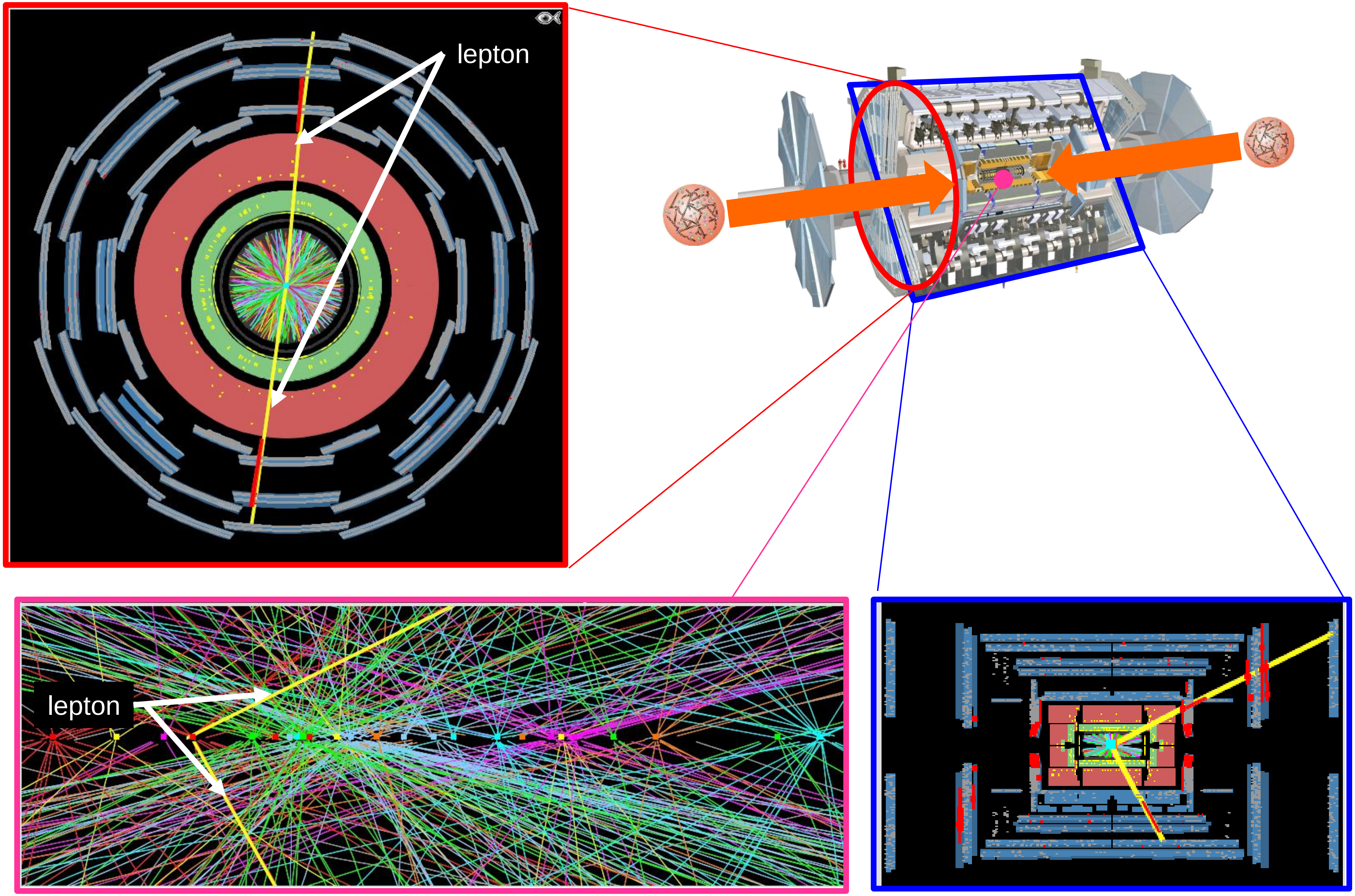
‘Flying garbage’

‘Hard Scatter’

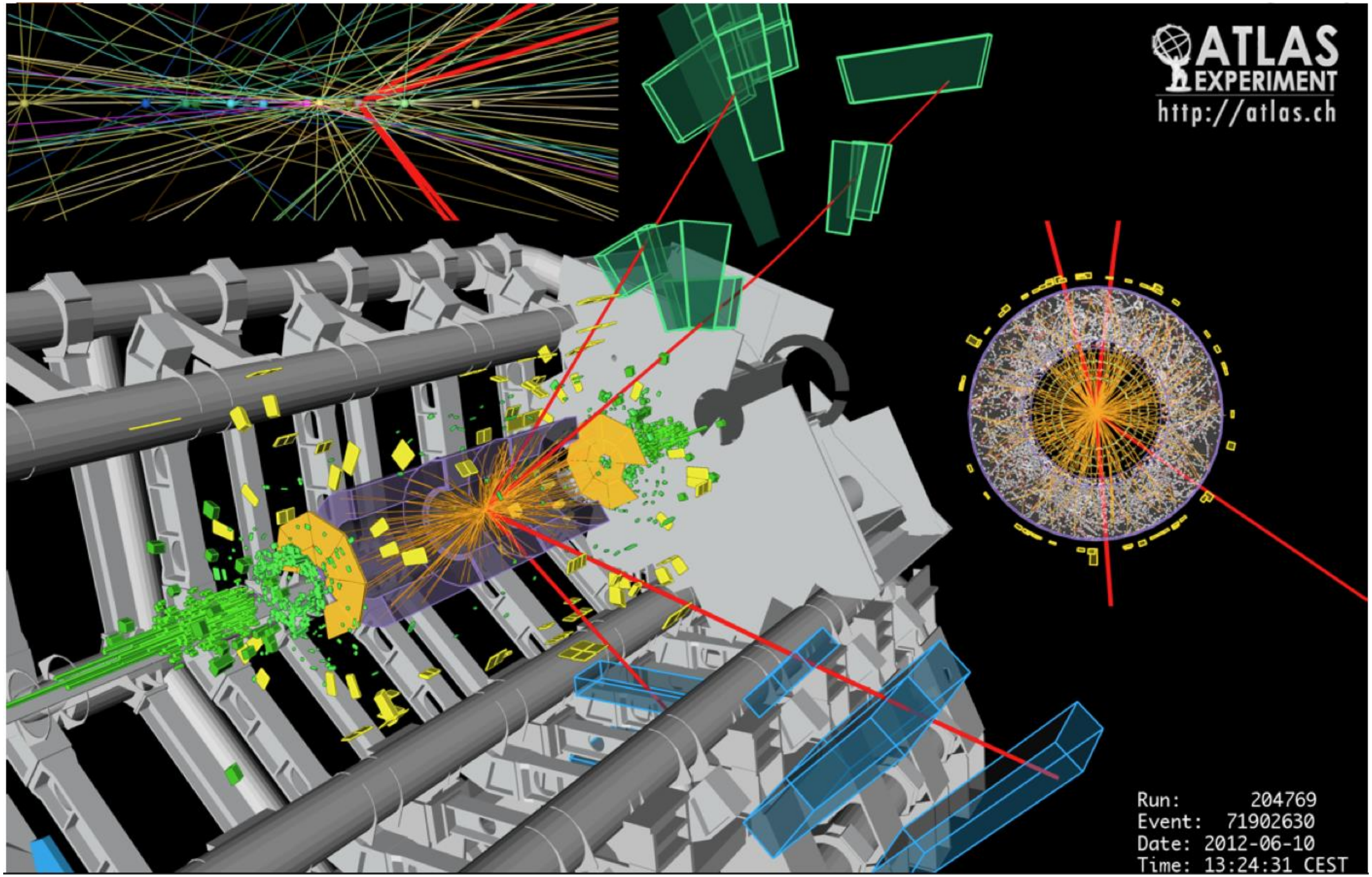
‘Secondary scatter’

Proton-Proton collision at the LHC

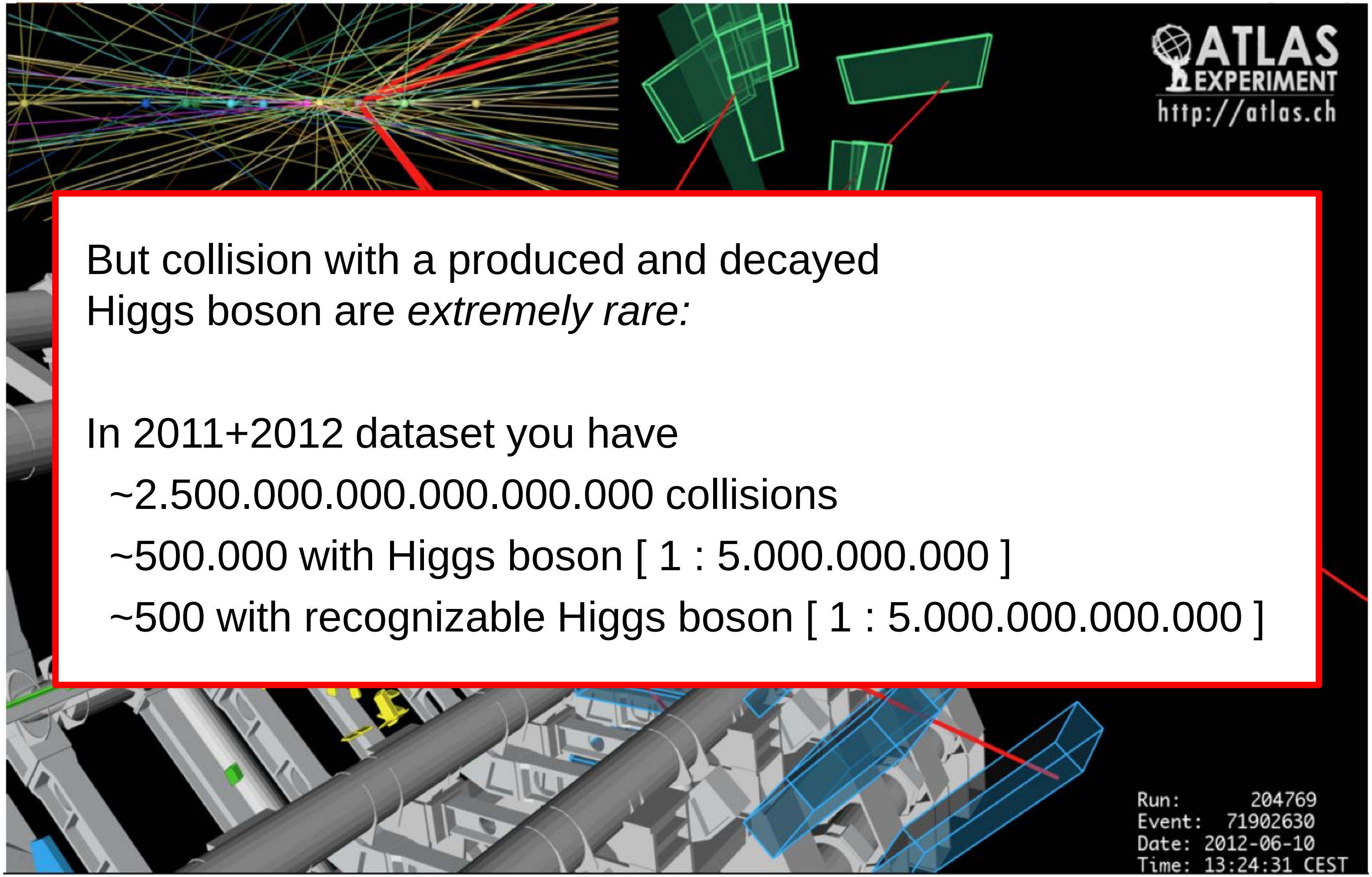
A typical proton-proton collision



Find the Higgs – need something stands out, e.g. 4 leptons



Find the Higgs – need something stands out, e.g. 4 leptons



But collision with a produced and decayed Higgs boson are *extremely rare:*

In 2011+2012 dataset you have

~2.500.000.000.000.000.000 collisions

~500.000 with Higgs boson [1 : 5.000.000.000]

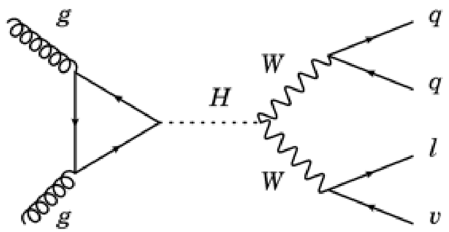
~500 with recognizable Higgs boson [1 : 5.000.000.000.000]

Online selection and trigger

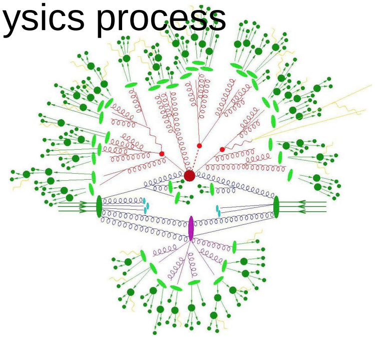
- You have already seen in previous lectures that a large part of the pre-selection of collision events is performed in real-time (‘the trigger’)
  - Reduces 40 MHz LHC collision rate to ~600Hz of selected events
  - Still leaves you with a few billion events written to disk/tape
- Goal: find the O(100) collision with a Higgs decay in a collection of a few billion events
- Open questions
  - How do you know what events with Higgs collisions look like?
  - Can you ever be sure that any given selected collision really contained a Higgs decay (since you can only see its decay products)?
  - How do you formulate evidence of the existence of a Higgs particle, if you can never really prove what happened ‘inside’ a collision?

How do you know what events with a Higgs looks like?

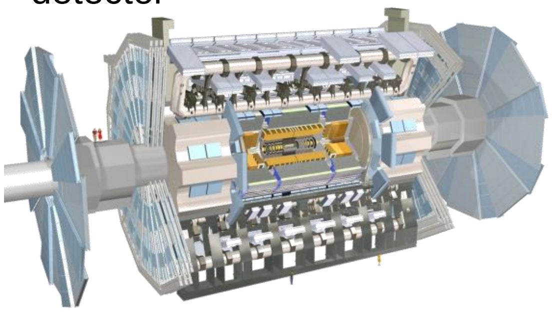
Simulation of high-energy physics process



Simulation of ‘soft physics’ physics process

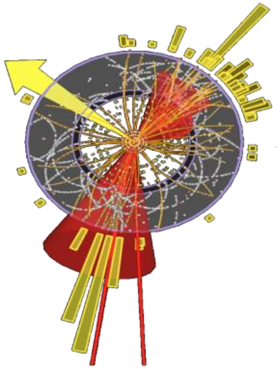


Simulation of ATLAS detector

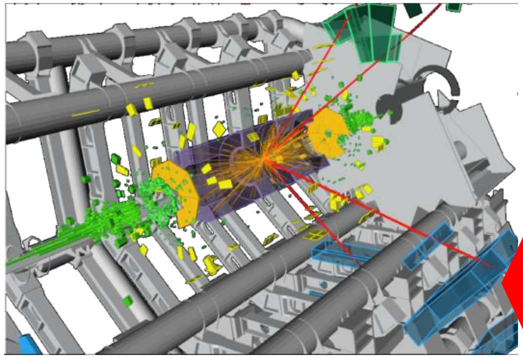


LHC data

Reconstruction of ATLAS detector

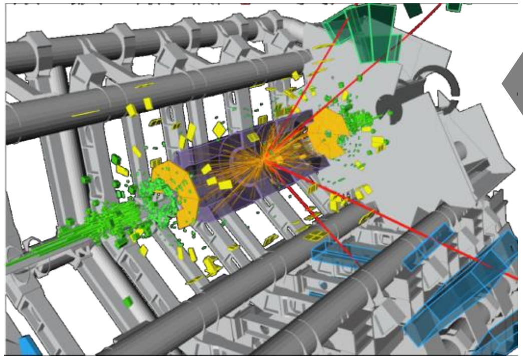


Simulated LHC event with H→Z→llll decay

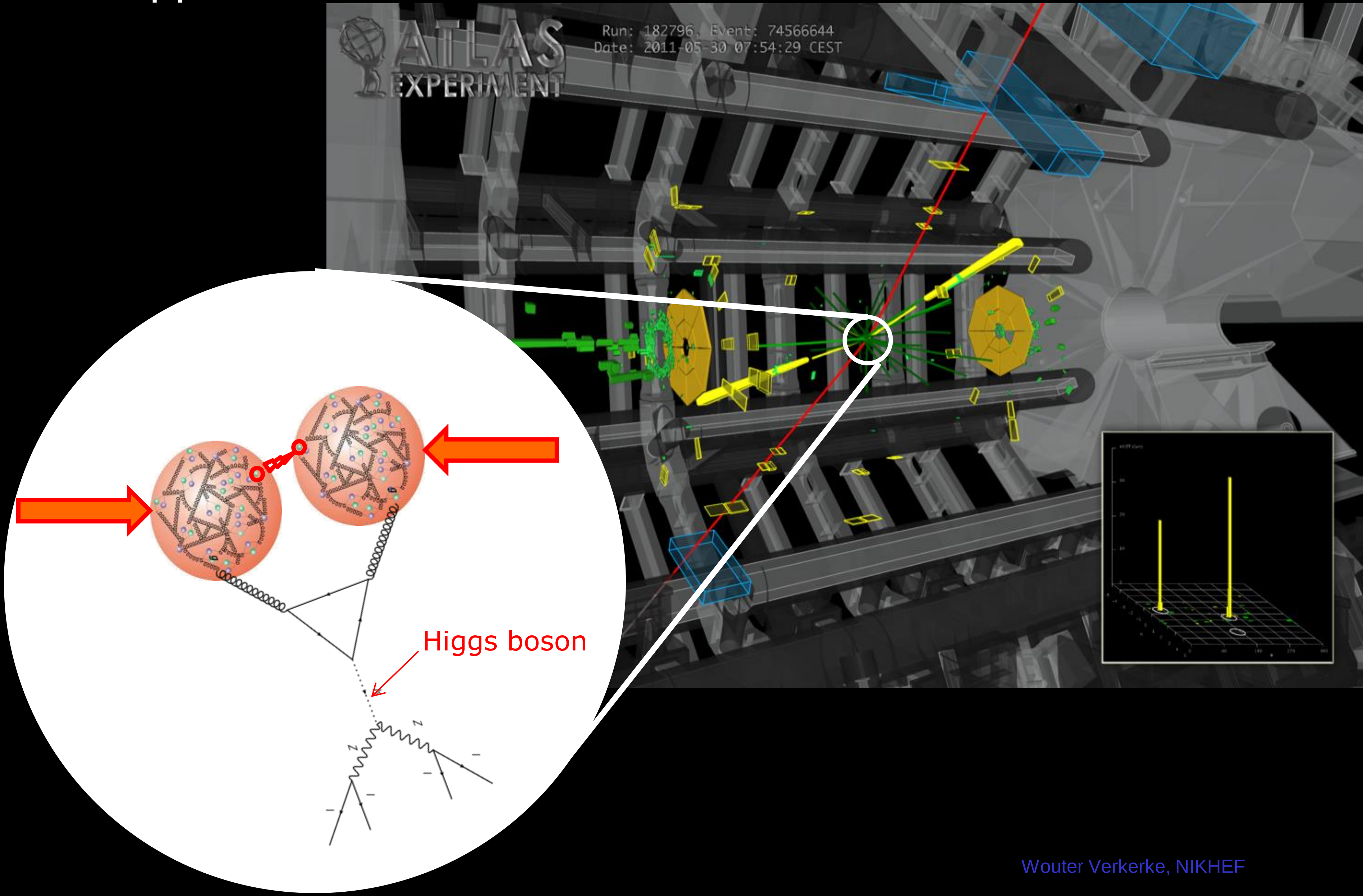


Observed LHC event with llll (4 leptons)

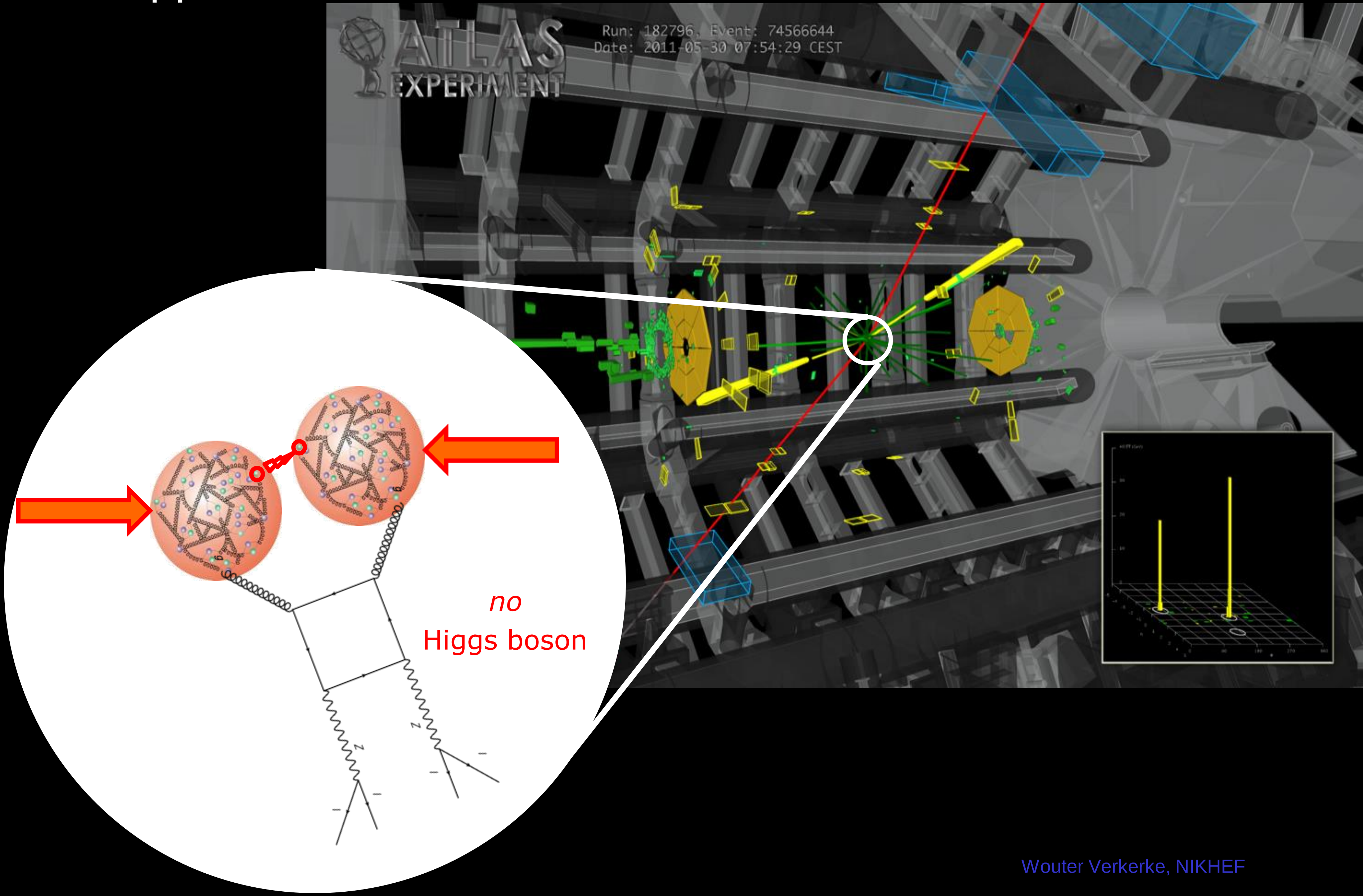
But is it H→ZZ→llll or [somethinge else] → llll?



Quantum mechanics – you are never sure what happened…



Quantum mechanics – you are never sure what happened…



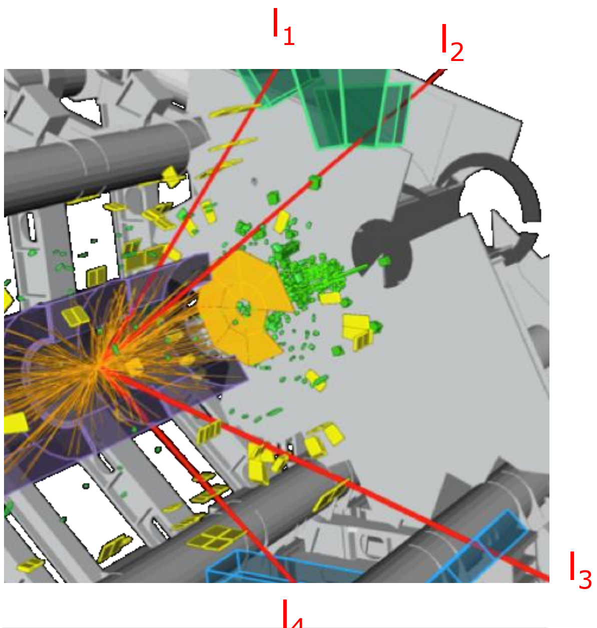
no
Higgs boson

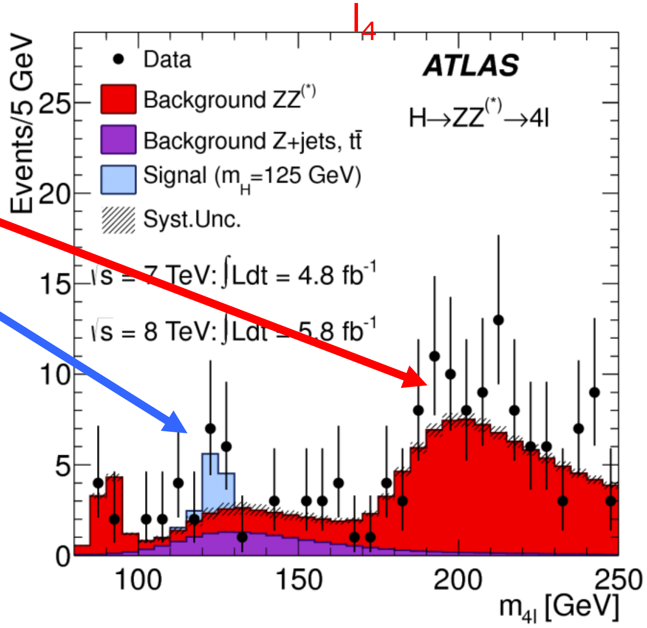
But properties of leptons will still tell you something…

- Higgs: 4 leptons originate from decay of a *single particle*
- Background – leptons originate from decay of *unrelated particles*
- The 4-lepton invariant mass will tell…

$$m_{4l} \equiv \sqrt{\left(\sum_{1,2,3,4} E_i\right)^2 - \left(\sum_{1,2,3,4} \vec{p}_i\right)\cdot\left(\sum_{1,2,3,4} \vec{p}_i\right)}$$

  - No Higgs: m_{4l} = ~random
  - Higgs: m_{4l} = Higgs boson mass
- Look for peak in m(4l), but don't a priori know where!
- Still – no *single* event provides conclusive evidence!





Statistical formulation of evidence

- When a single observation can – for fundamental quantum-mechanical reasons – not be conclusive, but can still make a probabilistic statement ('statistics')

- Start of with a simple analogy using dice
  We have a dice. Q: is it a regular dice, or a fake one?

Regular dice

Fake dice

- Quantum aspect: we can't see the dice, we can only ask someone to roll it for us (repeatedly) and report the outcome

Statistical formulation of evidence

- How can we ‘discover’ that the dice is fake?
- Start with formulation of two competing theories
  - Hypothesis 1 – Regular dice ‘no Higgs’
  - Hypothesis 2 – Fake dice (always 6) ‘Higgs’
- Perform an experiment – result: score ‘6’
- What can we say about nature of dice?
  - Prob(score 6|fake)=100%→ Thus dice is fake?
  - But prob(score 6 |normal) = 1:6 →
    Probability of ‘accidental’ score 6 with regular dice fairly large
- No clear conclusion → need more data

Statistical formulation of evidence

- Repeat experiment twice – result: 3 x score ‘6’
  - Prob(3x score 6|fake) = 100%
  - Kans(3x score 6|normal) = 1:(6x6x6) = 1:216

- Becoming more convinced that dice might be fake, but not absolutely sure.
- Q: How sure do you want to be?
- A: Depend on prior credibility of theory you’re testing.
  - If you’re aiming to discovery existence of Martians, bar is very high as theory is a priori very incredible
  - If you’re aiming to discovery a new particle that theory clearly predicts, bar might be lower
- Repeat experiment again twice – result: 5 x score ‘6’
  - Prob(5x score 6|fake) = 100%
  - Kans(5x score 6|normal) = 1:(6x6x6x6x6) = 1:7776

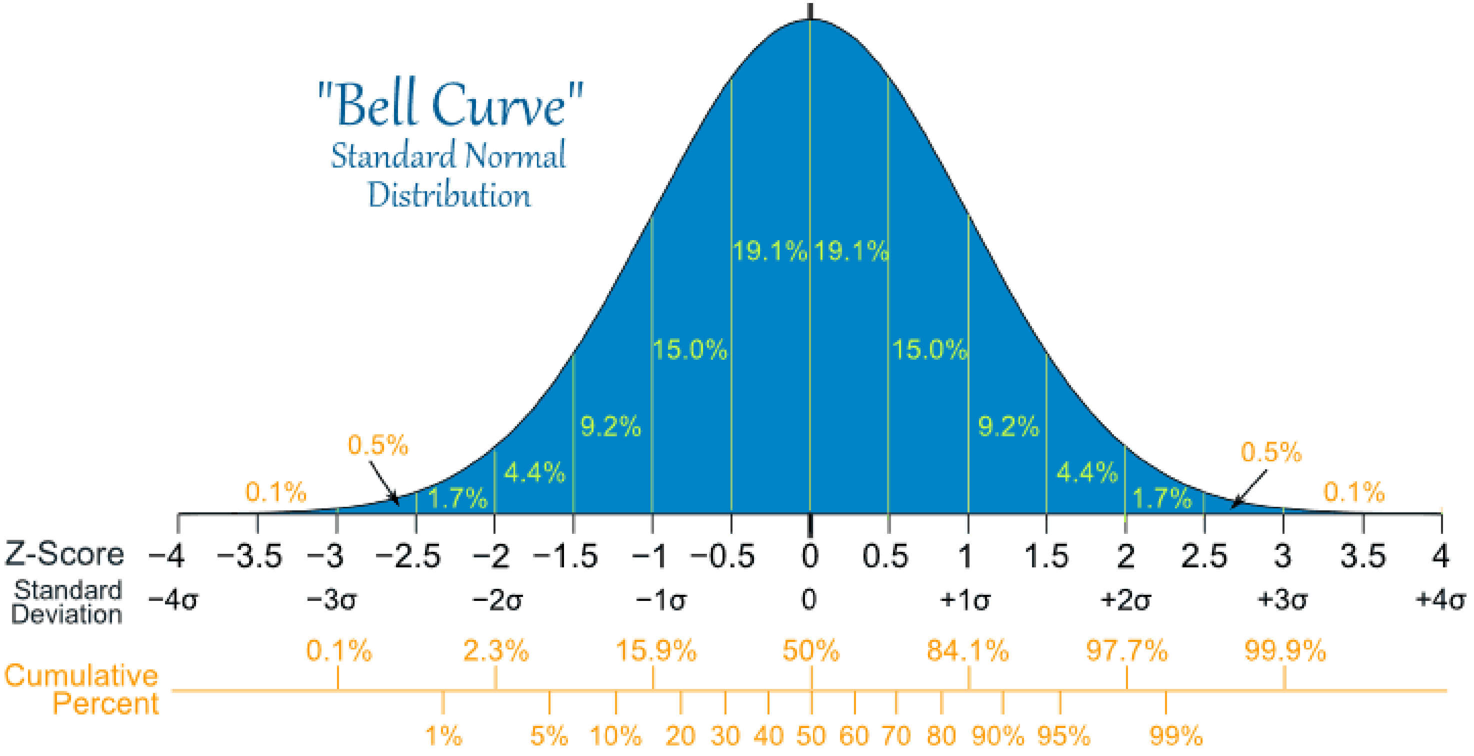
Statistical formulation of evidence

- Usual standard in particle physics is known as ‘5 sigma’
  - Defined as probability of a unit Gaussian distribution to deviate by >5, which has a probility of 2.8x10^{-7}
  - In other words: probability that your ‘background-only’ hypothesis results in observed signal must be less than ~1:3.5 million
- Using the ‘5 sigma’ standard you would accept only 9 consecutive dice rolls with score 6 as evidence for a fake dice

- *Nomenclature – The probability obtain your result under the ‘null’ (background hypothesis) is called the ‘p-value’*

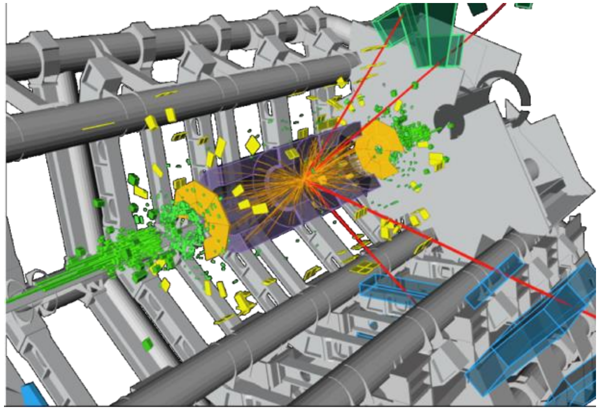
Statistical formulation of evidence

- Usual standard in particle physics is known as ‘5 sigma’
  - Gaussian sigmas ‘Z-score’ are simply another way to conventiently express small probabilities
  - Relates probabilities to the ‘normal (Gaussian) distribution’
  - For example a ‘3 sigma’ excess is an excess where the p-value is 0.001 (since only 0.001 of the Gaussian curve is beyond 3 sigma)

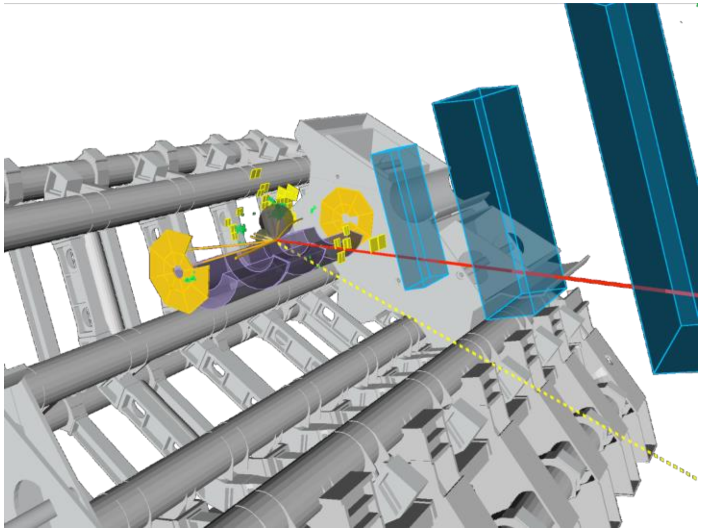


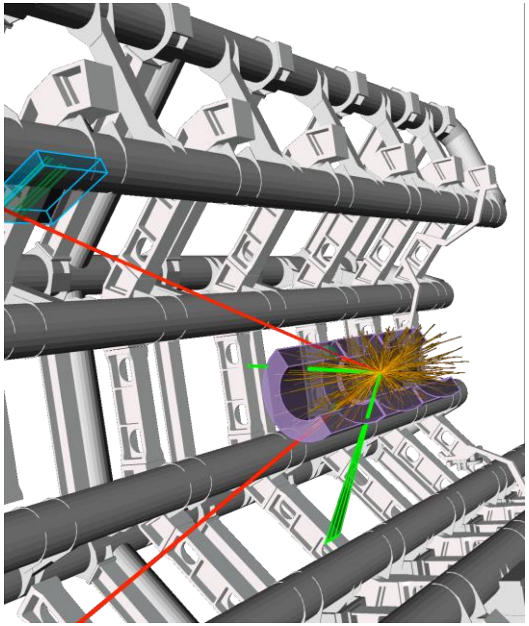
From dice to LHC collisions

- Dice provide easy example for calculating odds, but how do these calculations apply to LHC collisions
- Each dice has six possible outcomes: score 1 … score 6
- What are the possibly outcomes of LHC collisions? Number of possibilities is almost infinite… How do we deal with this?



Need
(automated)
event classification...



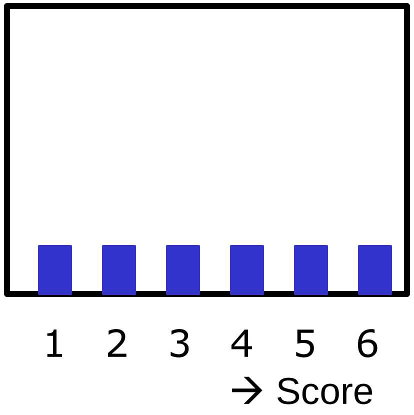


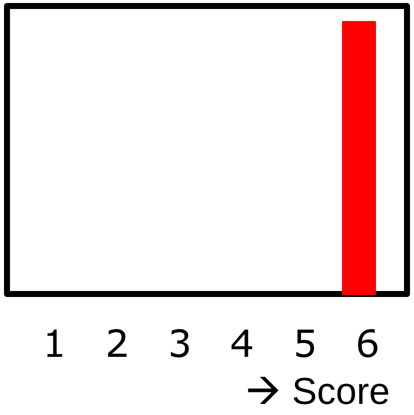
Event classification

- For simple Higgs decay signatures can do event classification ‘by hand’ (using our physics knowledge)
  - Classify events 2 types: ‘signal-like’ (selected) and ‘background-like’ (discarded)
- For example: decay H→ZZ→llll can be selected by requiring events to have four lepton tracks what appear to originate from Z decays.
  - Signal-like events: all events with 4 leptons with certain criteria
  - Background-like events: all other events
- Reduces properties of each LHC collision event to a single Boolean
- But we analyze all LHC collision events:
  output of full analysis is *count of selected signal-like events*
  → Output of full analysis is characterized by a integer number
- Compare observed number of selected events with expectation of selected event count for Higgs hypothesis and no-Higgs hypothesis

Statistical evidence from dice counting

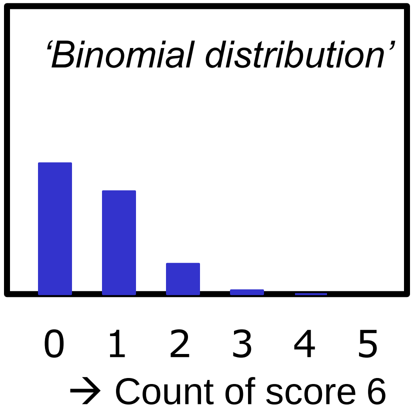
- Ilustration of event counting with dice. First consider rolling a single dice.



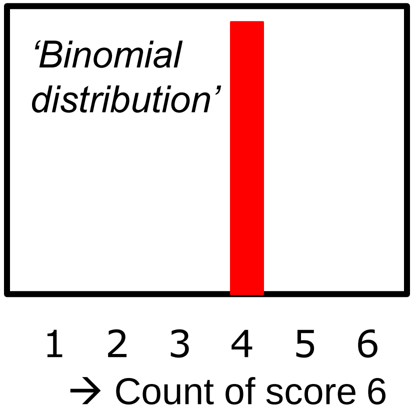


- *Now rolls dice 4 times (=1 expt), count number of sixes in each expt*

<Nsix>=4/6



<Nsix>=4



Statistical evidence from dice counting

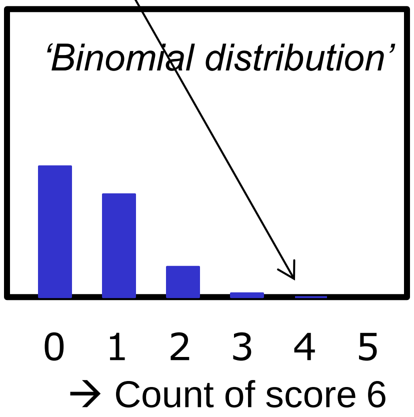
Suppose we observe Nsix=4

Can now trivially obtain probabilities from score distributions:

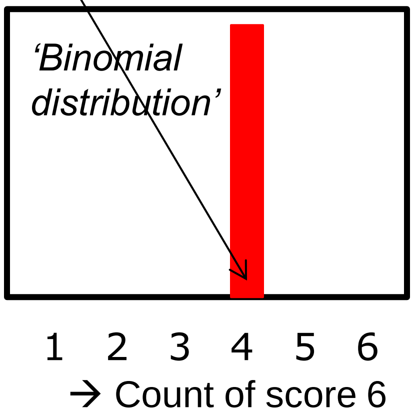
P(Nsix=4|regular) = 0.00077 P(Nsix=4|fake)=1

- *Now rolls dice 4 times (=1 expt), count number of sixes in each expt*

<Nsix>=4/6



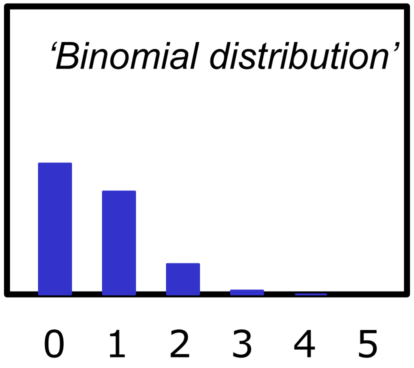
<Nsix>=4



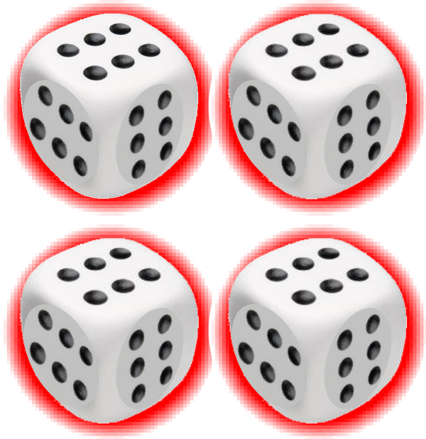
From dice counting to LHC event counting

- *Each experiments rolls dice 4 times, count number of number of sixes*

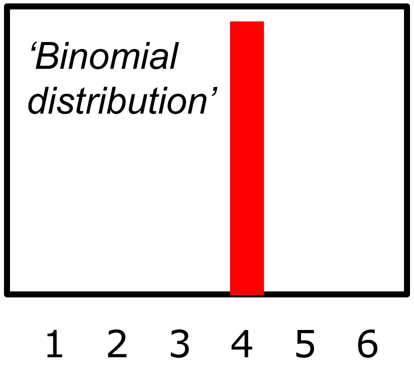
<Nsix>=4/6



→ Count of score 6



<Nsix>=4

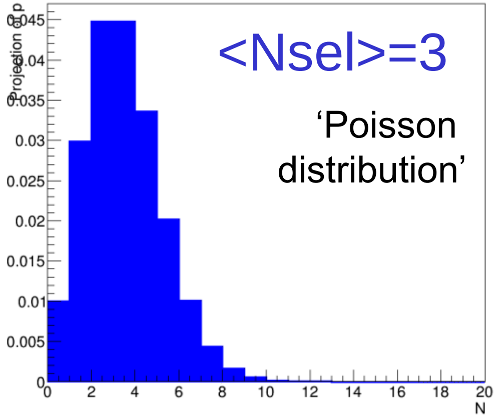


→ Count of score 6

- *Each expt collects 2 years of LHC data, count # of four-lepton events*

PREDICTION 1:

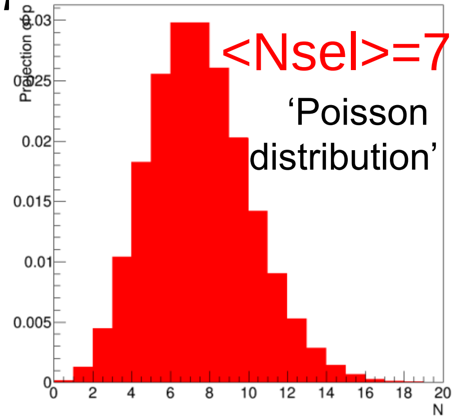
Number of LHC events with 4 leptons for theory with no Higgs boson



→ Observed #selected events

PREDICTION 2:

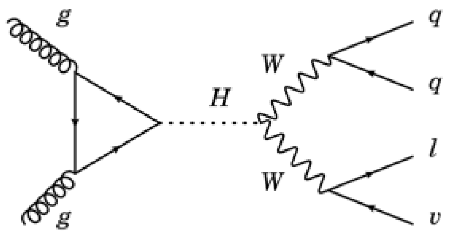
Number of LHC events with 4 leptons for theory with Higgs boson



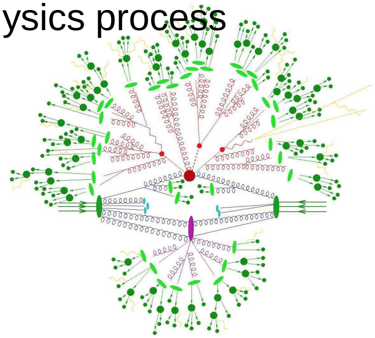
→ Observed #selected events

Calculating the *expected* outcome of an experiment

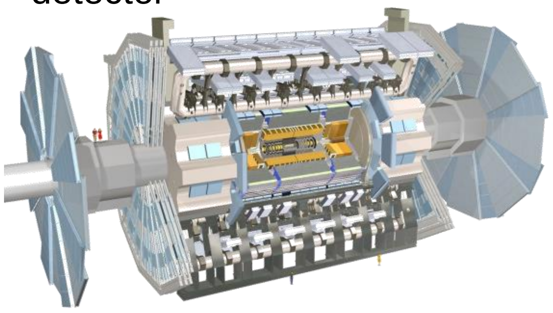
Simulation of high-energy physics process



Simulation of ‘soft physics’ physics process

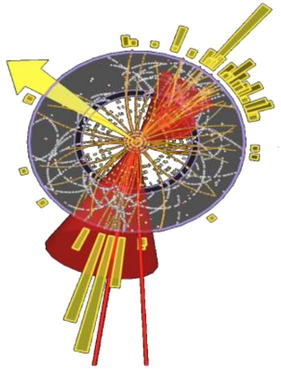


Simulation of ATLAS detector



LHC data

Reconstruction of ATLAS detector



Analysis Event selection

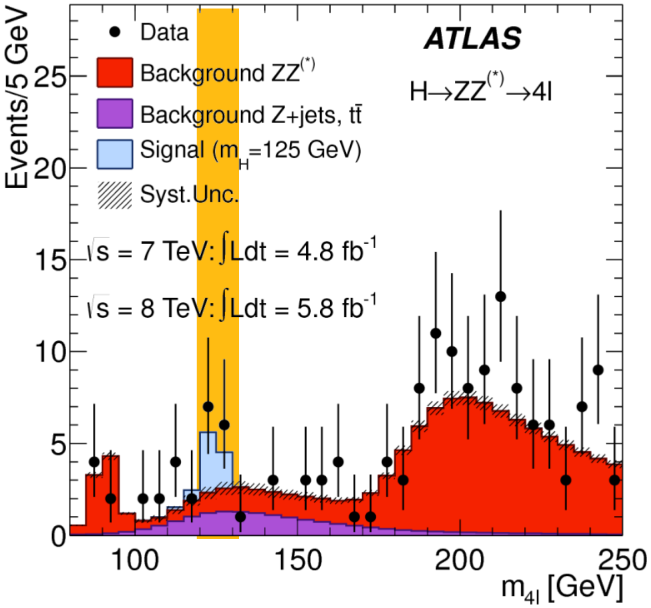
<Nsel>=3 if Higgs doesn’t exist

<Nsel>=7 if Higgs does exist

Observed <Nsel>=11

Event counting for Higgs – example with ATLAS H→ZZ→lll signal

- Now apply calculation of probabilities of event counting to a realistic example: ATLAS H→ZZ→llll sample
- Count events in yellow band



N(observed) = 13

Expectation – no Higgs

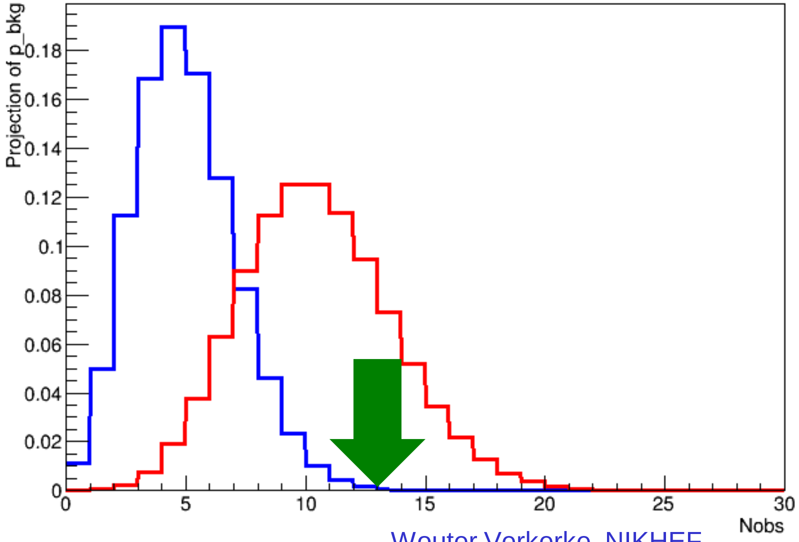
Poisson distribution with <N>=4.5

➔ prob(N≥13) = 0.08%

Expectation – SM Higgs

Poisson distribution with <N>=10

➔prob(N≥13) = 21%



Wouter Verkerke, NIKHEF

Doing better than counting – multivariate analysis & machine learning

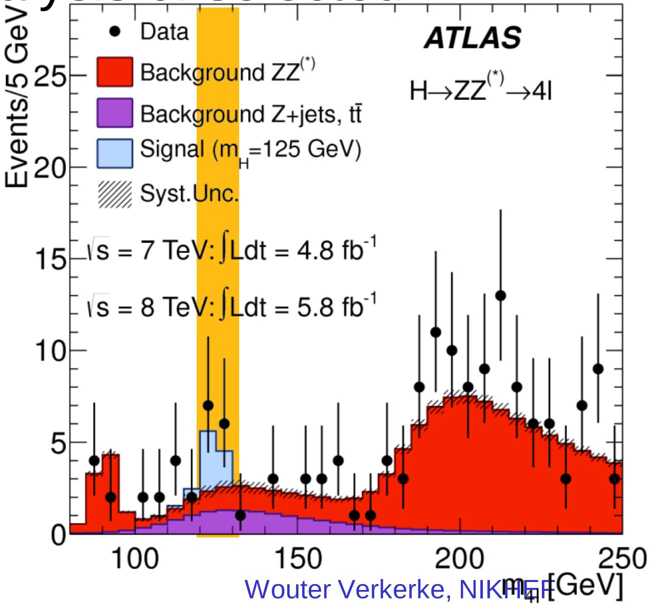
- Example of H→ZZ→llll was chosen because it lends itself well to a simple counting analysis, because signal is quite clean and selection criteria are relatively simple, but doesn't give enough statistical evidence to claim a discovery.

- How can we do better?

1. Design 'better' event selection (using more information that 'simple count' of leptons)

2. Exploit more information in statistical analysis of selected events

3. Look for Higgs in additional channels that are more challenging



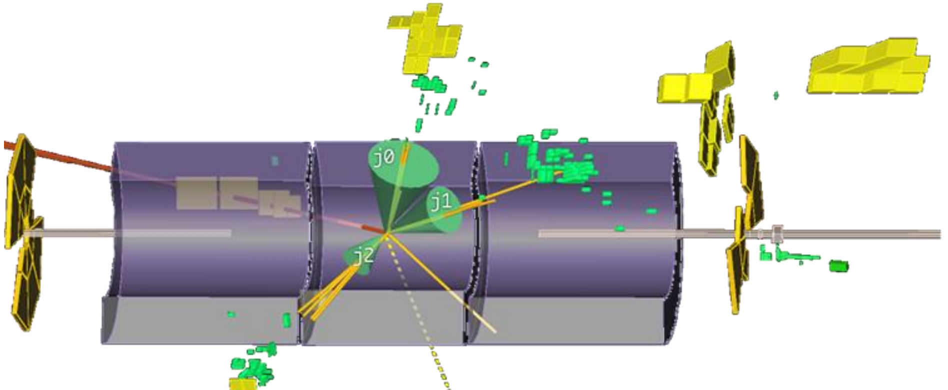
Machine learning example – Boosted decision trees

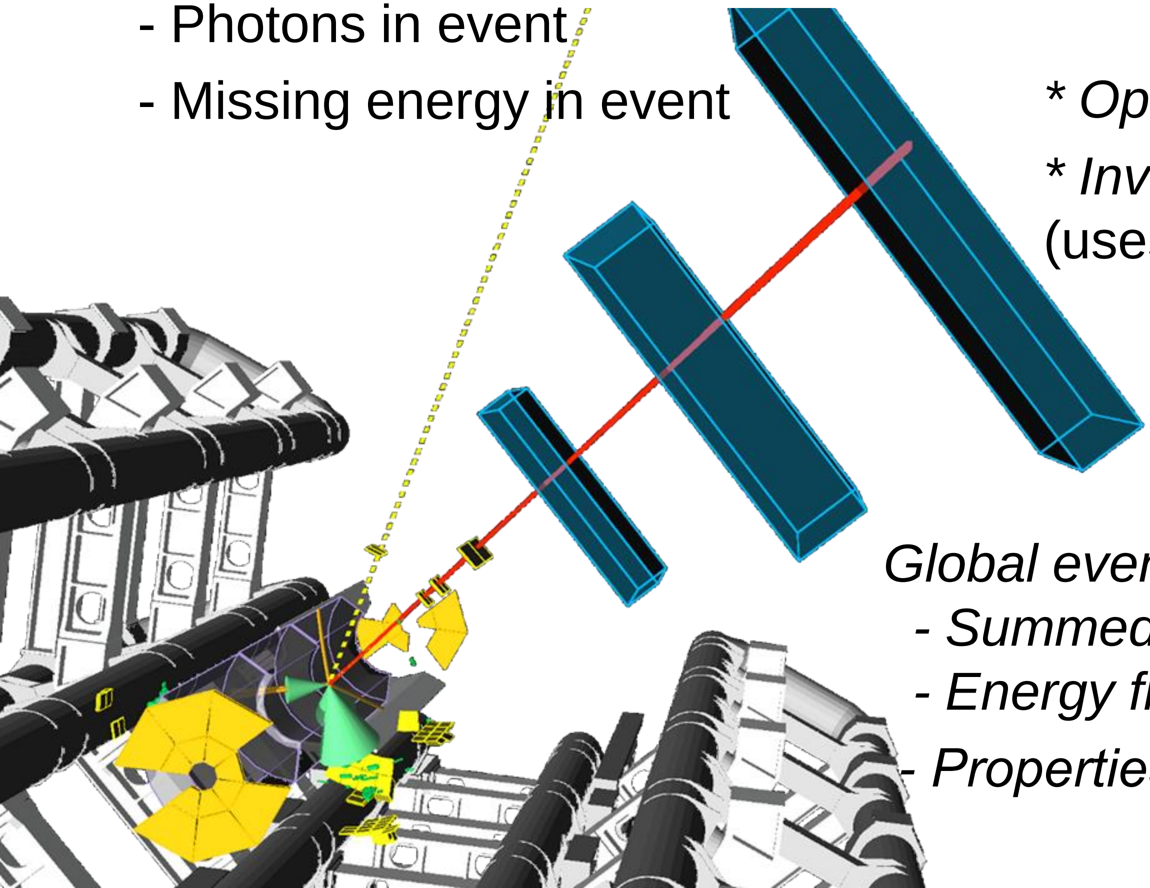
- Instead of a phycisist using his time and knowledge to design a clever event selection → Feed information about properties of signal and background algorithm to a ‘machine learning’ algorithm that can design the ‘best’ selection for you
  - Can (in principle) give better results, as much more information can be considered and can be used
  - But careful supervision and validation is needed – machine learning treats all provided properties of signal and background events as ‘exact’, whereas in reality simulation of certain event properties may be quite uncertain
- Popular technique at LHC is technique of ‘boosted decision trees’
- Decision tree = flow chart of binary selection cuts
  - Conceptualy similar to ‘manual’ 4-lepton selection illustrated for H→ZZ selection
  - But now let learner automatically decide on what observable event property~ best discriminates between signal and background

Event properties that can be used in machine learning

** Momentum (p_T) and direction of*

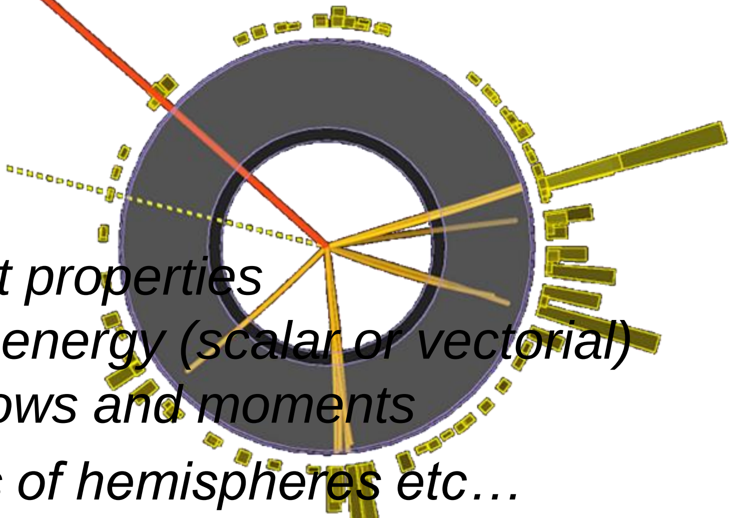
- Electrons, Muons, Taus in event
- Jets in event
- Flavor-tagged jets in event
- Photons in event
- Missing energy in event





** Open angles between objects*

** Invariant masses of objects*
(uses opening angles and momenta)



Global event properties

- Summed energy (scalar or vectorial)

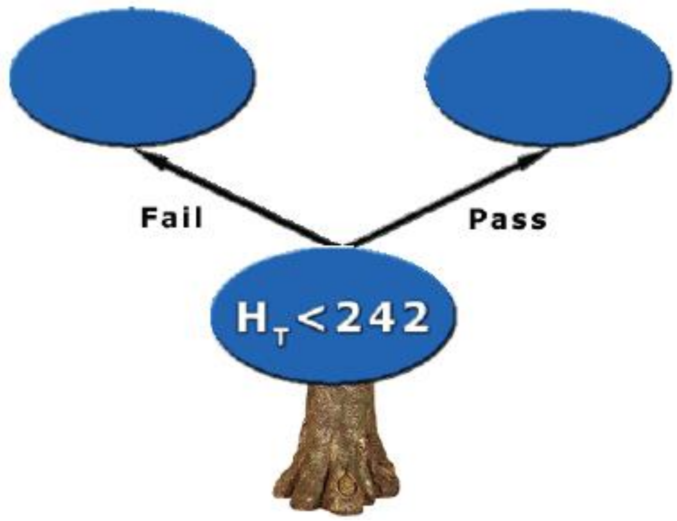
- Energy flows and moments

- Properties of hemispheres etc…

Wouter Verkerke, NIKHEF

Building a tree – splitting the data

- Essential operation :
  splitting the data in 2 groups using a single cut, e.g. H_T<242

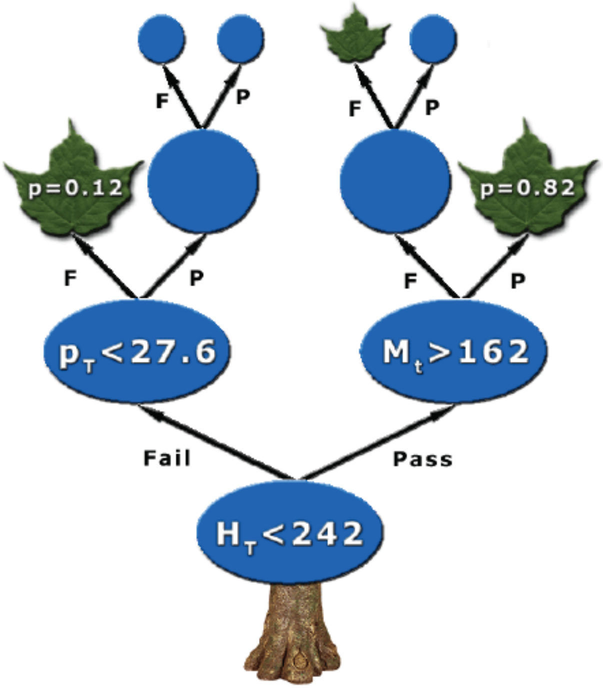


- Goal: find 'best cut' as quantified through best separation of signal and background (requires some metric to quantify this)

- Procedure:
  1) Find cut value with best separation for *each* observable
  2) Apply **only** cut on observable that results in best separation

Building a tree – recursive splitting

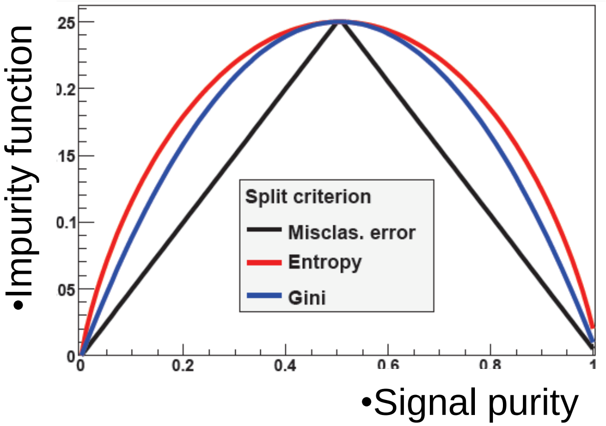
- Repeat splitting procedure on sub-samples of previous split



- Output of decision tree:
  - ‘signal’ or ‘background’ (0/1) or
  - probability based on *expected purity* of leaf (s/s+b)

Machine learning with Decision Trees

- Goal: Minimize ‘Impurity Function’ of leaves
  - Impurity function *i(t)* quantifies (im)purity of a sample, but is not uniquely defined
  - Simplest option: i(t) = misclassification rate



- For a proposed split *s* on a node *t*, decrease of impurity is

$$\Delta i(s,t) = i(t) - p_L \cdot i(t_L) - p_R \cdot i(t_R)$$

•Impurity of sample before split

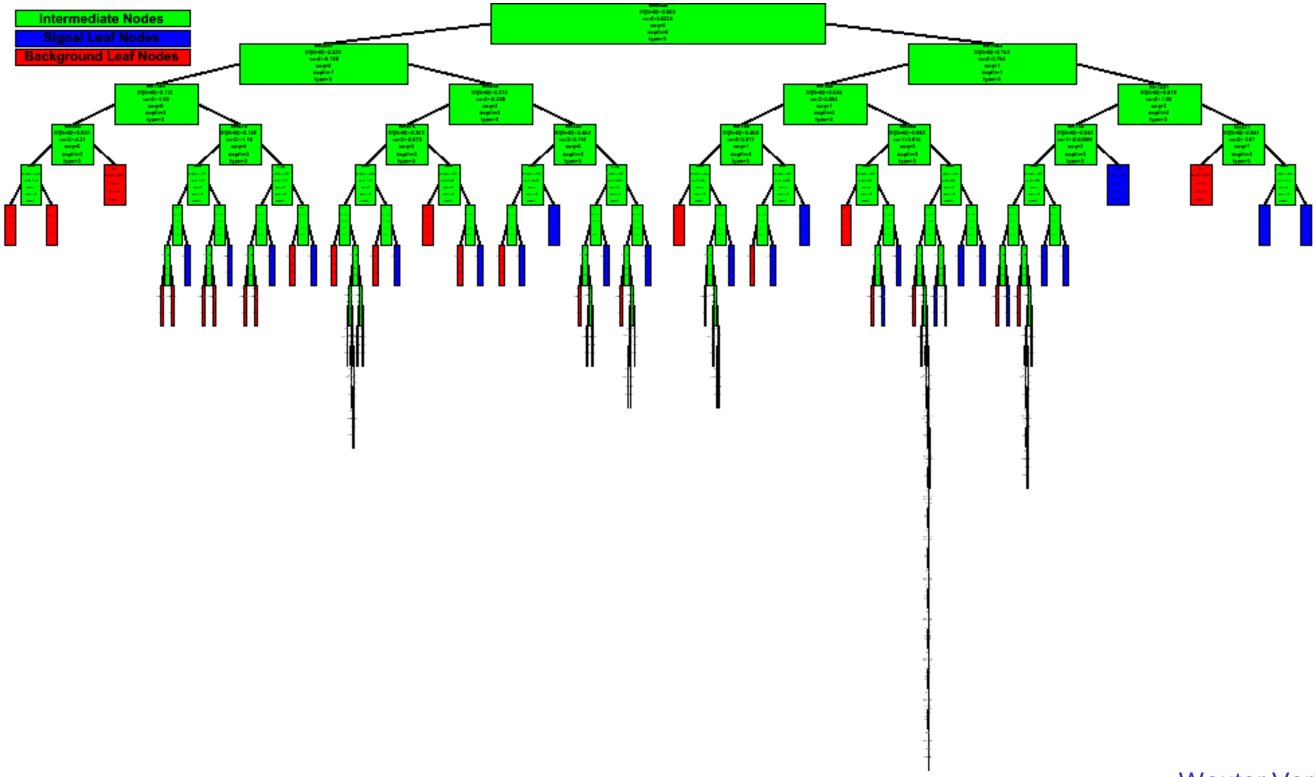
•Impurity of ‘left’ sample

•Impurity of ‘right’ sample

- Take split that results in largest Δi

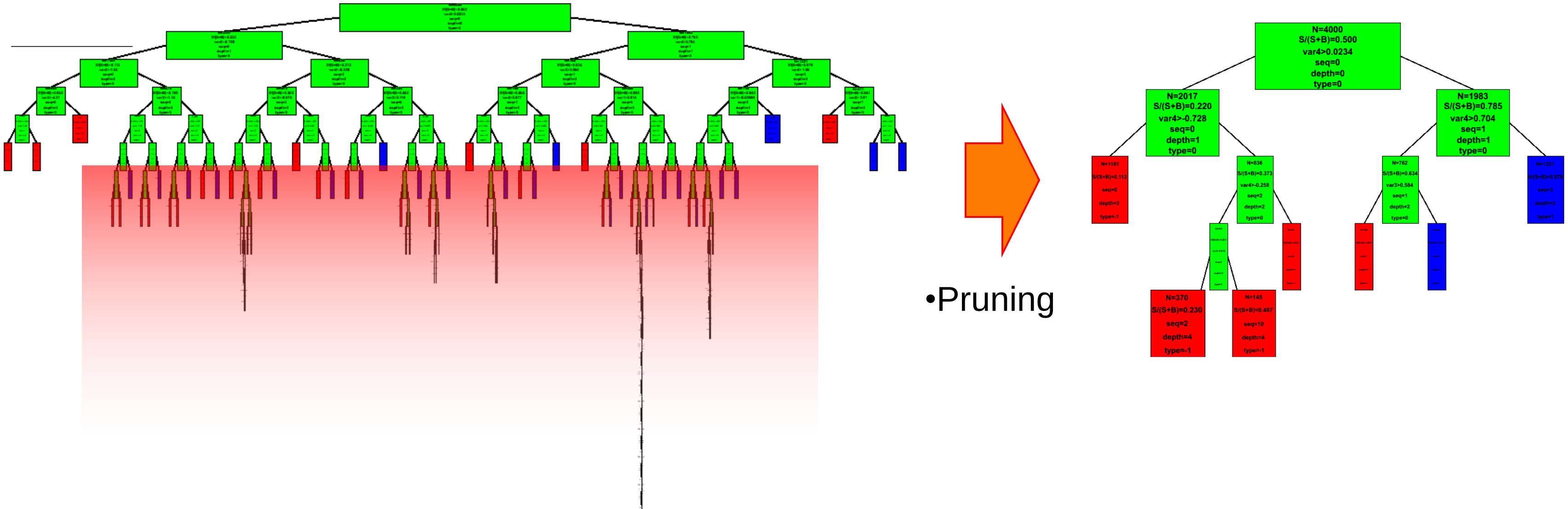
Machine learning with Decision Trees

- Stop splitting when
  - not enough improvement (introduce a cutoff Δi)
  - not enough statistics in sample, or node is pure (signal or background)
- Example decision tree from learning process



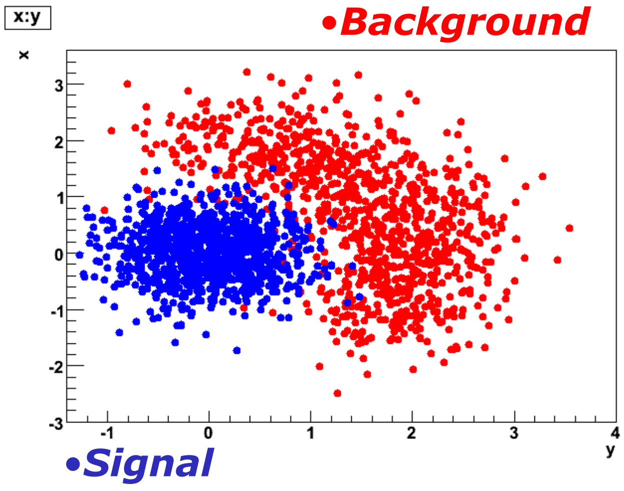
Machine learning with Decision Trees

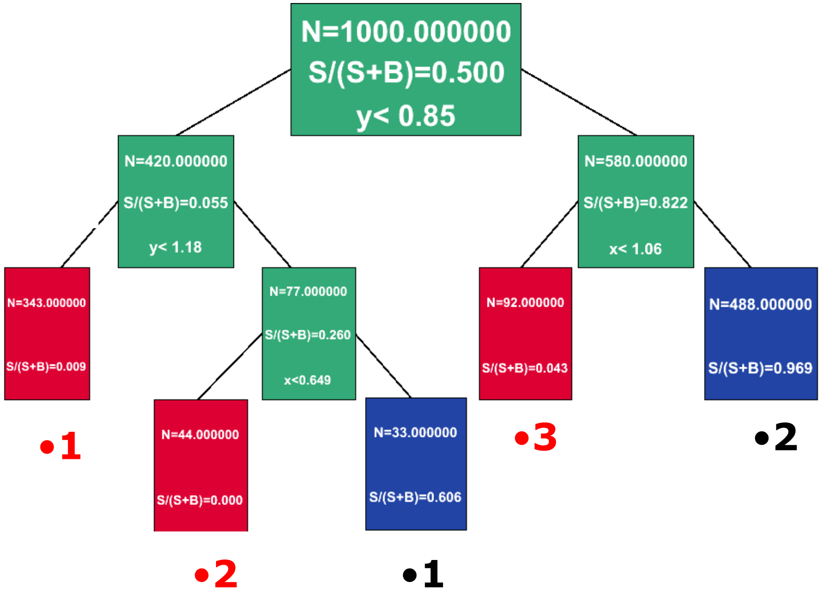
- Given that training happens on finite samples of simulated signal and background events, splitting decisions are based on 'empirical impurity' rather than true 'impurity' → risk of overtraining exists



- Can mitigate effects of overtraining by 'pruning' tree *a posteriori*
  - Expected error pruning (prune weak splits that are consistent with original leaf within statistical error of training sample)
  - Cost/Complexity pruning (generally strategy to trade tree complexity against performance)

Concrete example of a trained Decision Tree





N=1000.000000
S/(S+B)=0.500
y< 0.85

N=420.000000
S/(S+B)=0.055
y< 1.18

N=580.000000
S/(S+B)=0.822
x< 1.06

N=343.000000
S/(S+B)=0.009

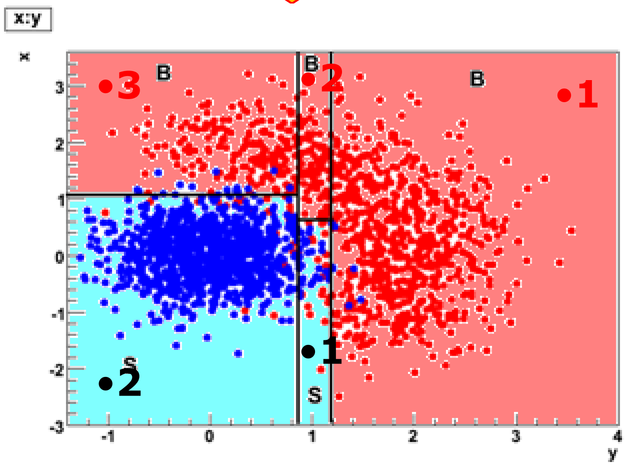
N=77.000000
S/(S+B)=0.260
x<0.649

N=92.000000
S/(S+B)=0.043

N=488.000000
S/(S+B)=0.969

N=44.000000
S/(S+B)=0.000

N=33.000000
S/(S+B)=0.606



Boosted Decision trees

- Decision trees largely used with ‘boosting strategy’
- Boosting = strategy to combine multiple weaker classifiers into a single strong classifier

- First provable boosting algorithm by Schapire (1990)
  - Train classifier *T1* on N events
  - Train *T2* on new N-sample, half of which misclassified by *T1*
  - Build *T3* on events where *T1* and *T2* disagree
  - Boosted classifier: MajorityVote(*T1,T2,T3*)

- **Most used: AdaBoost** = Adaptive Boosting (Freund & Shapire ‘96)
  - Learning procedure adjusts to training data to classify it better
  - Many variations on the same theme for actual implementation

AdaBoost

- Schematic view of *iterative* algorithm
  - Train Decision Tree on (weighted) signal and background training samples
  - Calculate misclassification rate for Tree K (initial tree has k=1)

$$\epsilon_k = \frac{\sum_{i=1}^{N} w_i^k \times \text{isMisclassified}_k(i)}{\sum_{i=1}^{N} w_i^k}$$

•“Weighted average of isMisclassified over all training events”

  - Calculate weight of tree K in ‘forest decision’ $\alpha_k = \beta \times \ln((1-\epsilon_k)/\epsilon_k)$
  - Increase weight of misclassified events in Sample(k) to create Sample(k+1)

$$w_i^k \rightarrow w_i^{k+1} = w_i^k \times e^{\alpha_k}$$

- Boosted classifier is result is performance-weighted ‘forest’

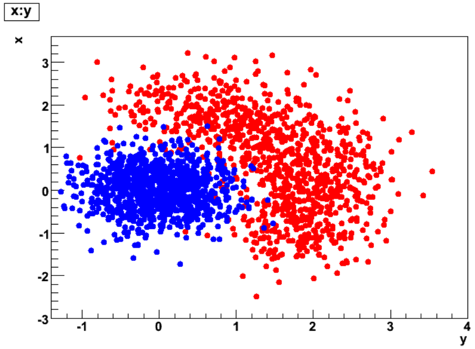
$$T(i) = \sum_{k=1}^{N_{\text{tree}}} \alpha_k T_k(i)$$

•“Weighted average of Trees by their performance”

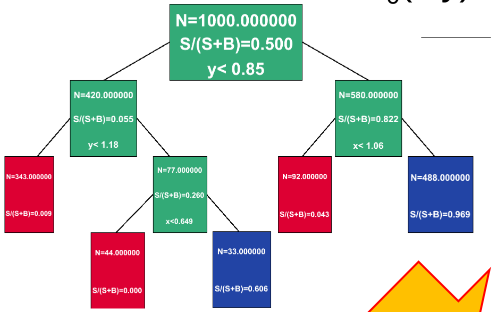
AdaBoost by example

- So-so classifier (Error rate = 40%) $\alpha = \ln \frac{1-0.4}{0.4} = 0.4$
  - Misclassified events get their weight multiplied by **exp(0.4)=1.5**
  - Next tree will have to work a bit harder on these events
- Good classifier (Error rate = 5%) $\alpha = \ln \frac{1-0.05}{0.05} = 2.9$
  - Misclassified events get their weight multiplied by **exp(2.9)=19** (!!)
  - Being failed by a good classifier means a big penalty: must be a difficult case
  - Next tree will have to pay much more attention to this event and try to get it right
- Note that boosting usually results in (strong) overtraining
  - Since with misclassification rate will ultimately go to zero

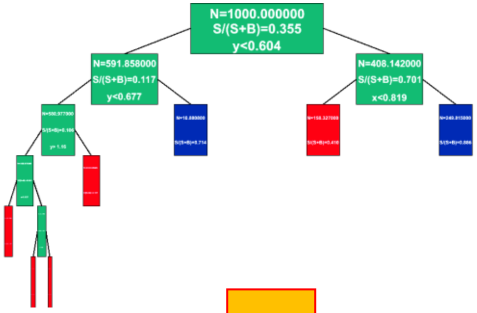
Example of Boosting



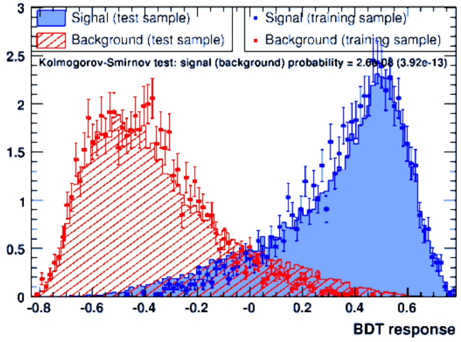
- $T_0(x,y)$

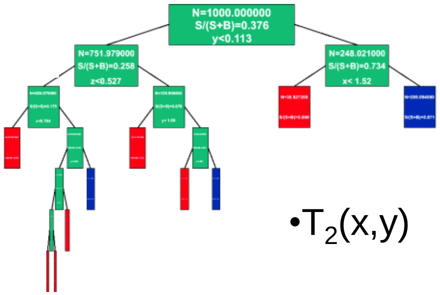


- $T_1(x,y)$

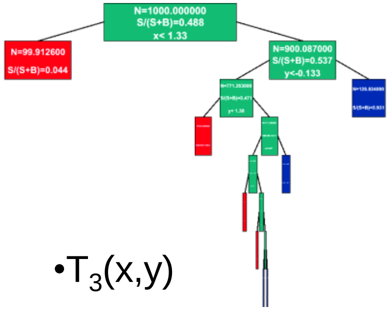


$$B(x,y)=\sum_{i=0}^{4}\alpha_i T_i(x,y)$$

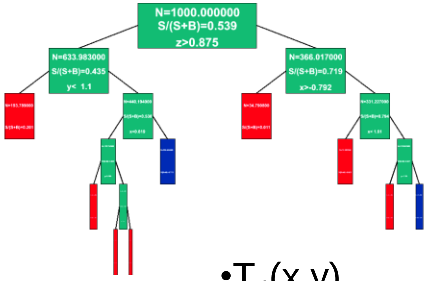




- $T_2(x,y)$



- $T_3(x,y)$



- $T_4(x,y)$

•What is *T*MVA

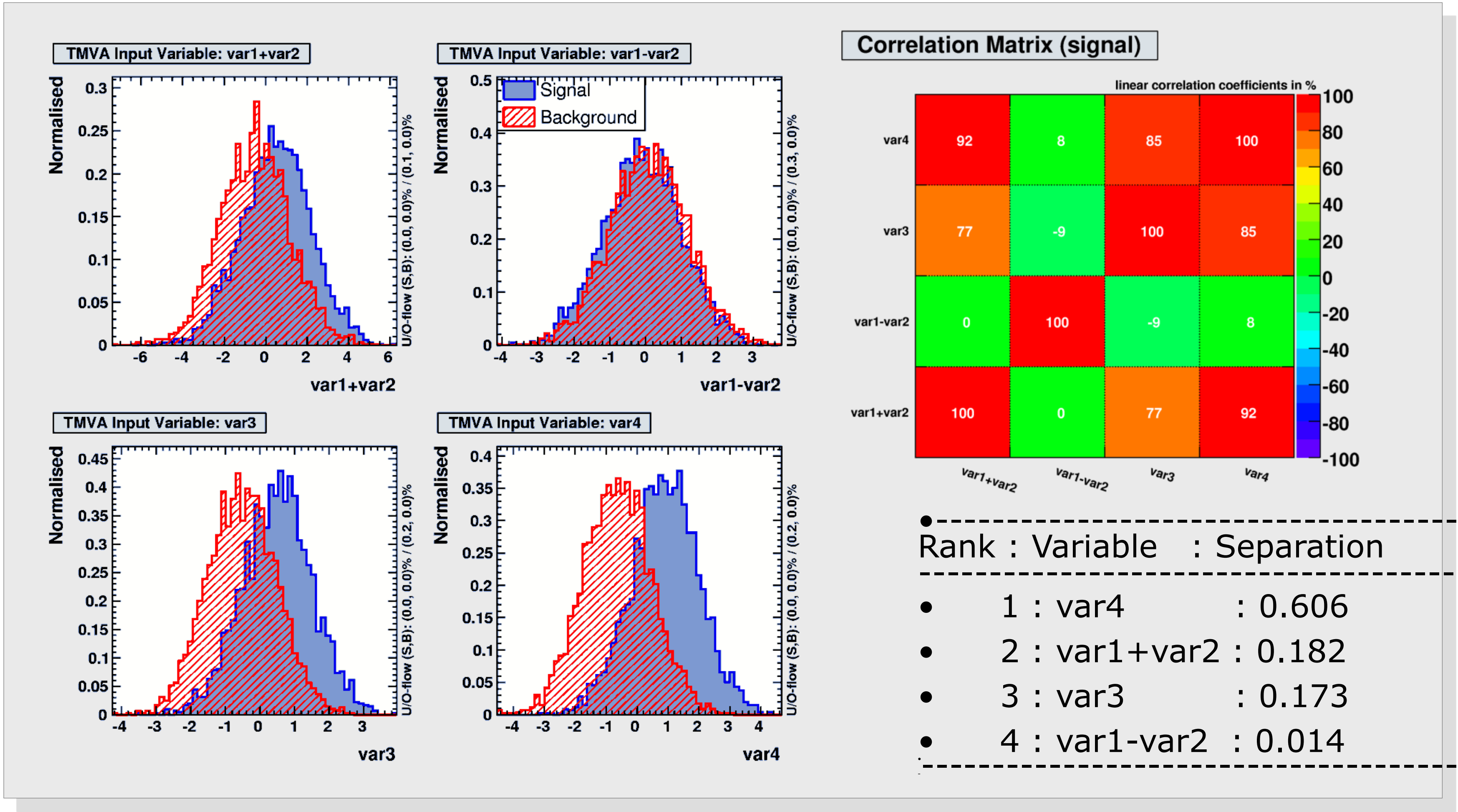
- ROOT: is the analysis framework used by most (HEP)-physicists
- Idea: rather than just implementing new MVA techniques and making them available in ROOT (*i.e.*, like TMultiLayerPercetron does):
  - Have one common platform / interface for all MVA classifiers
  - Have common data pre-processing capabilities
  - Train and test all classifiers on same data sample and evaluate consistently
  - Provide common analysis (ROOT scripts) and application framework
  - Provide access with and without ROOT, through macros, C++ executables or python

*T*MVA Content

- Currently implemented classifiers
  - Rectangular cut optimisation
  - Projective and multidimensional likelihood estimator
  - k-Nearest Neighbor algorithm
  - Fisher and H-Matrix discriminants
  - Function discriminant
  - Artificial neural networks (3 *multilayer perceptron* impls)
  - **Boosted/bagged decision trees**
  - RuleFit
  - Support Vector Machine
- Currently implemented data preprocessing stages:
  - Decorrelation
  - Principal Value Decomposition
  - Transformation to uniform and Gaussian distributions

•A Toy Example (idealized)

- Use data set with 4 linearly correlated Gaussian distributed variables:

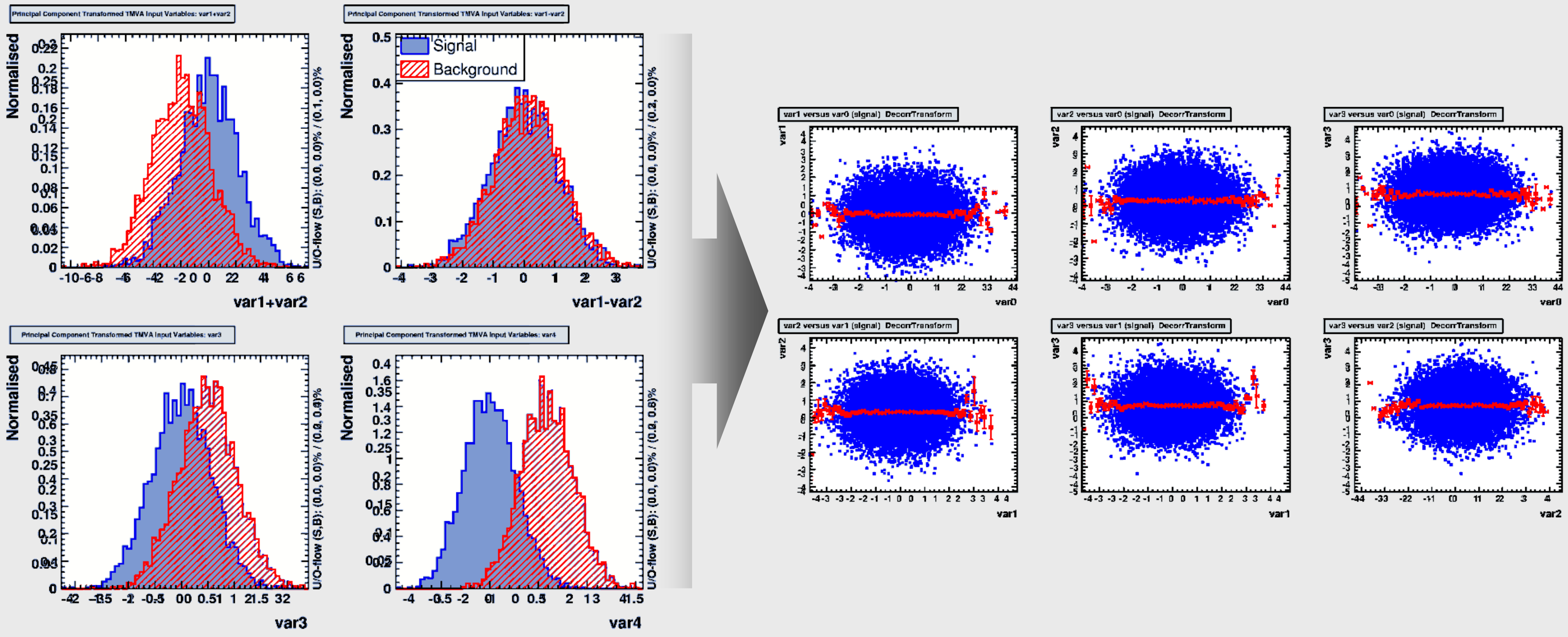


	var1+var2	var1-var2	var3	var4
var4	92	8	85	100
var3	77	-9	100	85
var1-var2	0	100	-9	8
var1+var2	100	0	77	92

```
•---------------------------------------------
Rank : Variable    : Separation
---------------------------------------------
•       1 : var4          : 0.606
•       2 : var1+var2 : 0.182
•       3 : var3          : 0.173
•       4 : var1-var2  : 0.014
---------------------------------------------
```

•Preprocessing the Input Variables

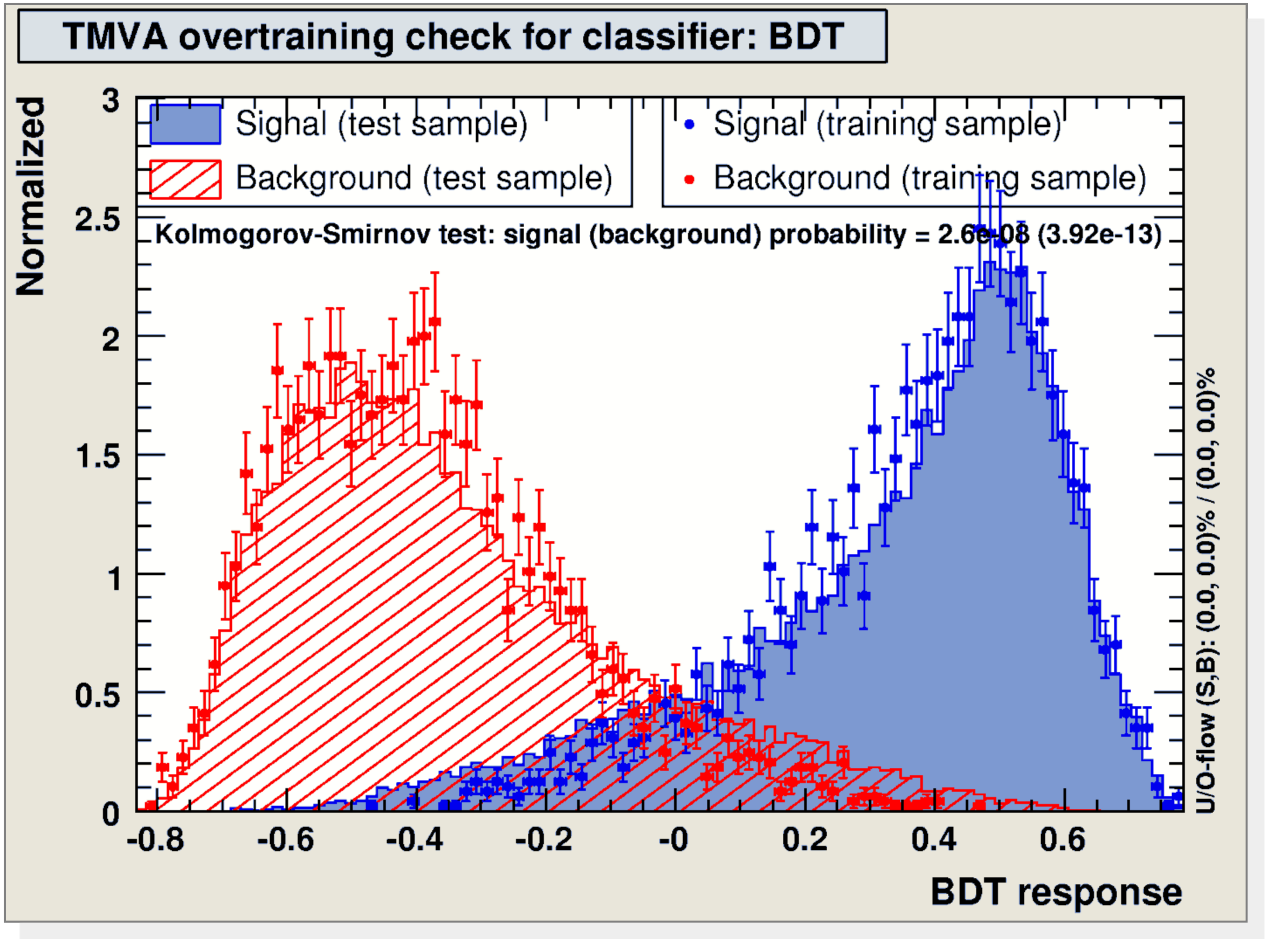
- Decorrelation of variables before training is useful for *this* example



- Note that in cases with non-Gaussian distributions and/or nonlinear correlations decorrelation may do more harm than any good

Evaluating the Classifier Training (II)

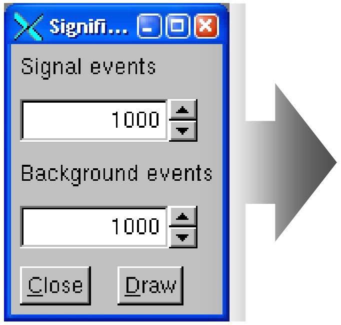
- Check for overtraining: classifier output for test *and* training samples …



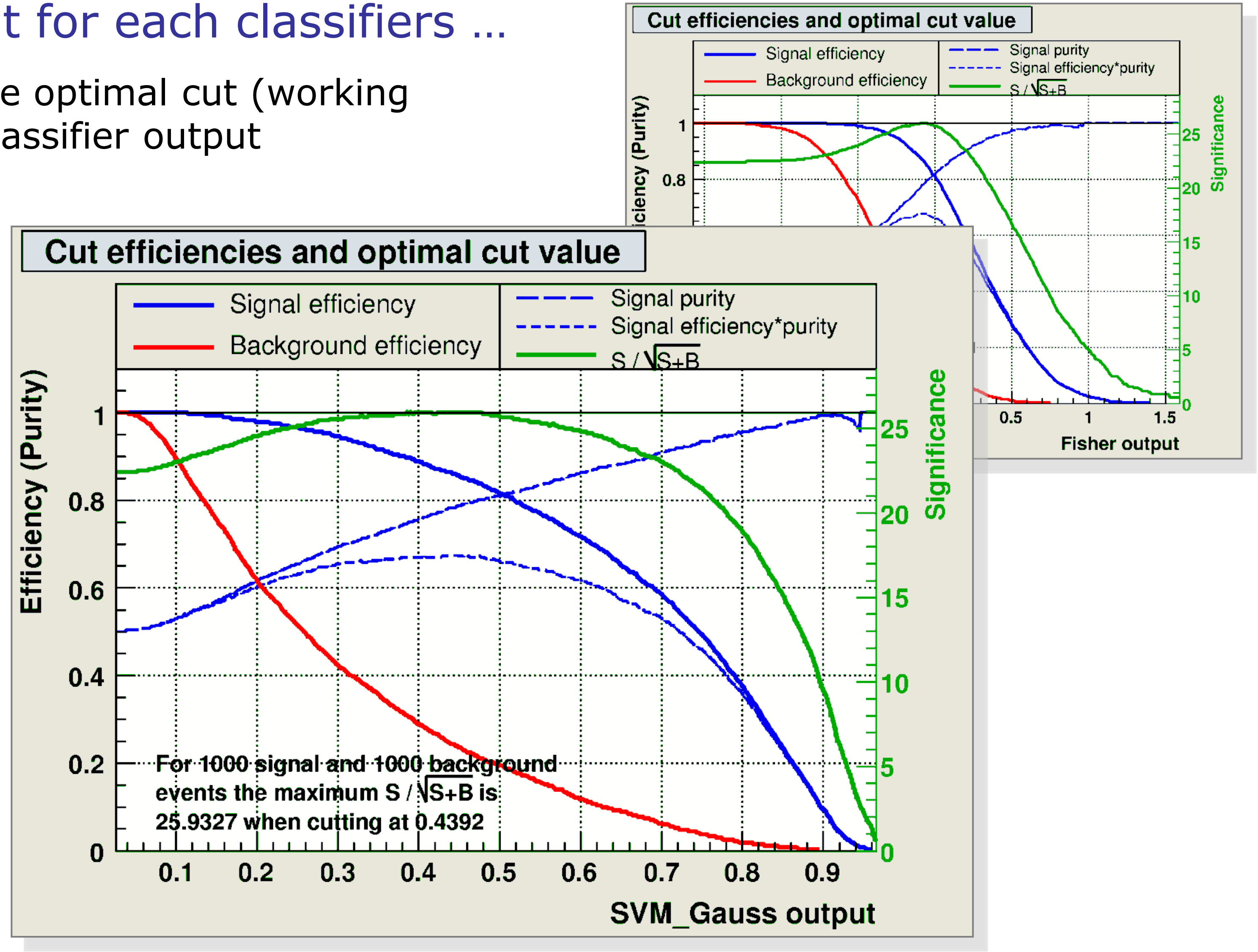
Evaluating the Classifier Training (V)

- Optimal cut for each classifiers …

  Determine the optimal cut (working point) on a classifier output

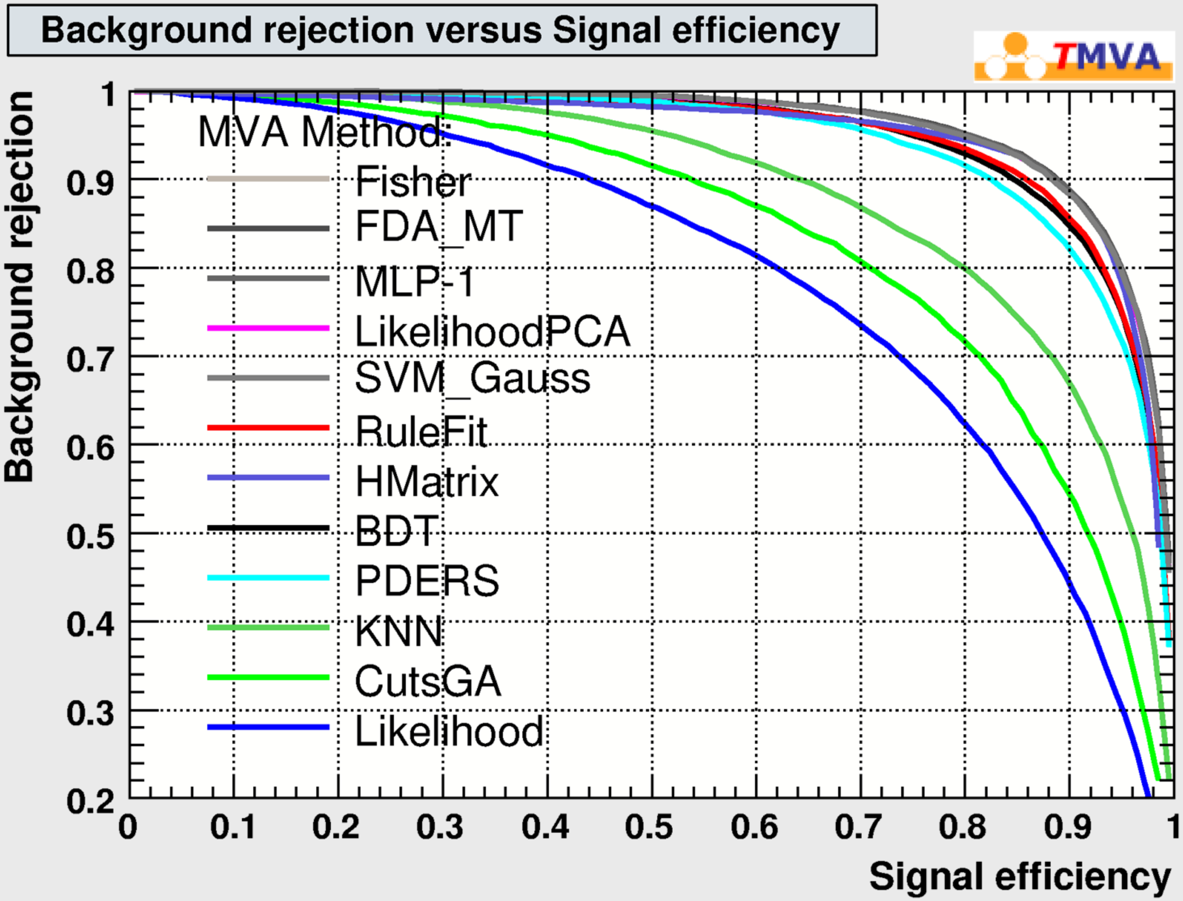

Signal events: 1000

Background events: 1000


For 1000 signal and 1000 background events the maximum S / $\sqrt{S+B}$ is 25.9327 when cutting at 0.4392

•Receiver Operating Characteristics (ROC) Curve

- Smooth background rejection versus signal efficiency curve: (from cut on classifier output)



- "Specificity" (probability to predict B if true B)

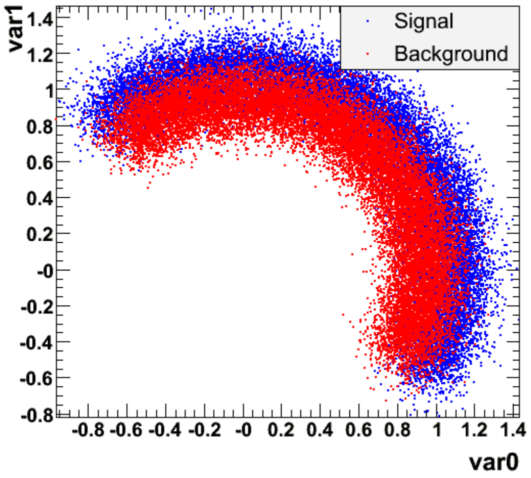
- "Sensitivity" (probability to predict S if true S)

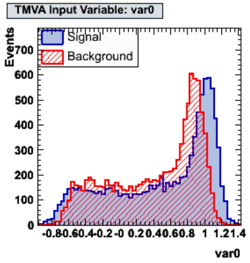
•Example: Circular Correlation

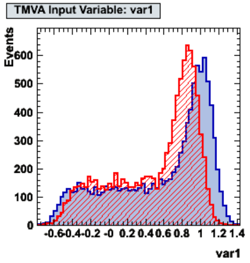
- Illustrate the behavior of linear and nonlinear classifiers

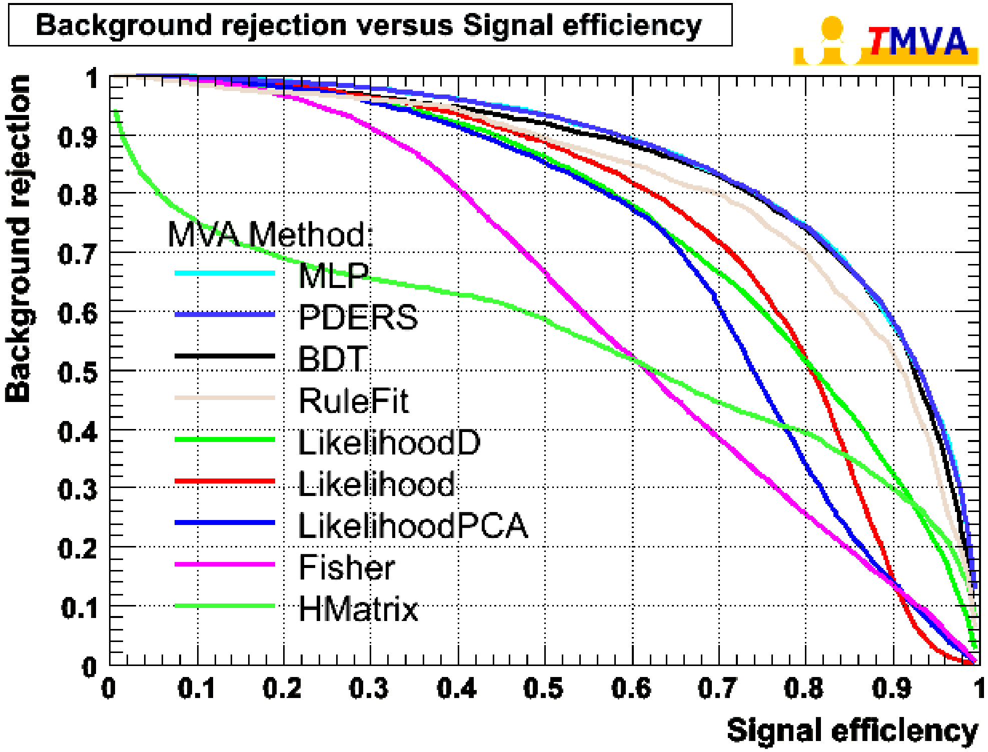
•Circular correlations

•(same for signal and background)

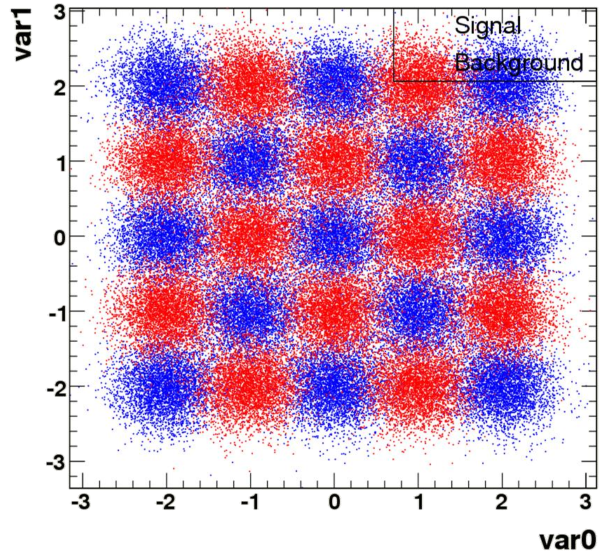


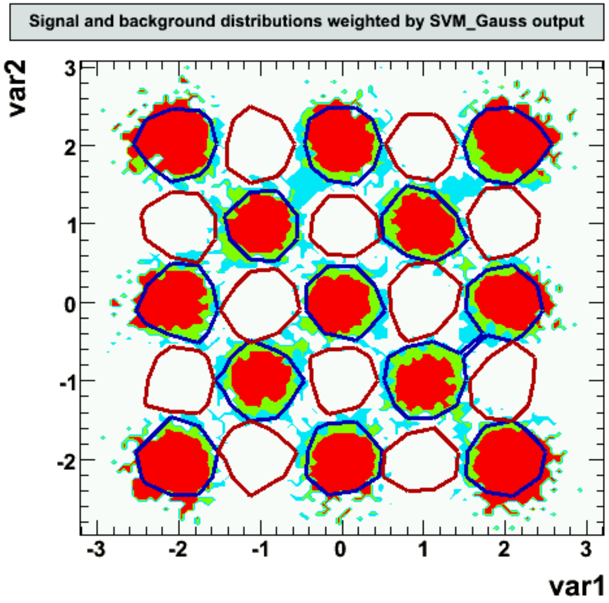






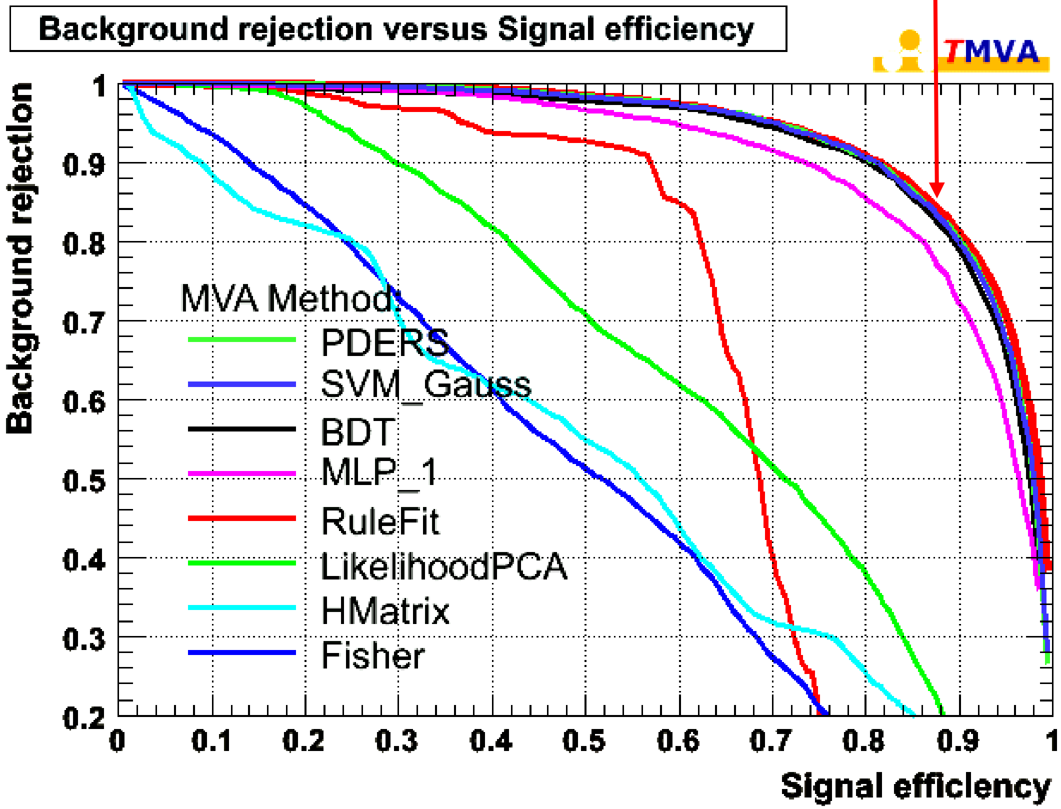
•The “Schachbrett” Toy





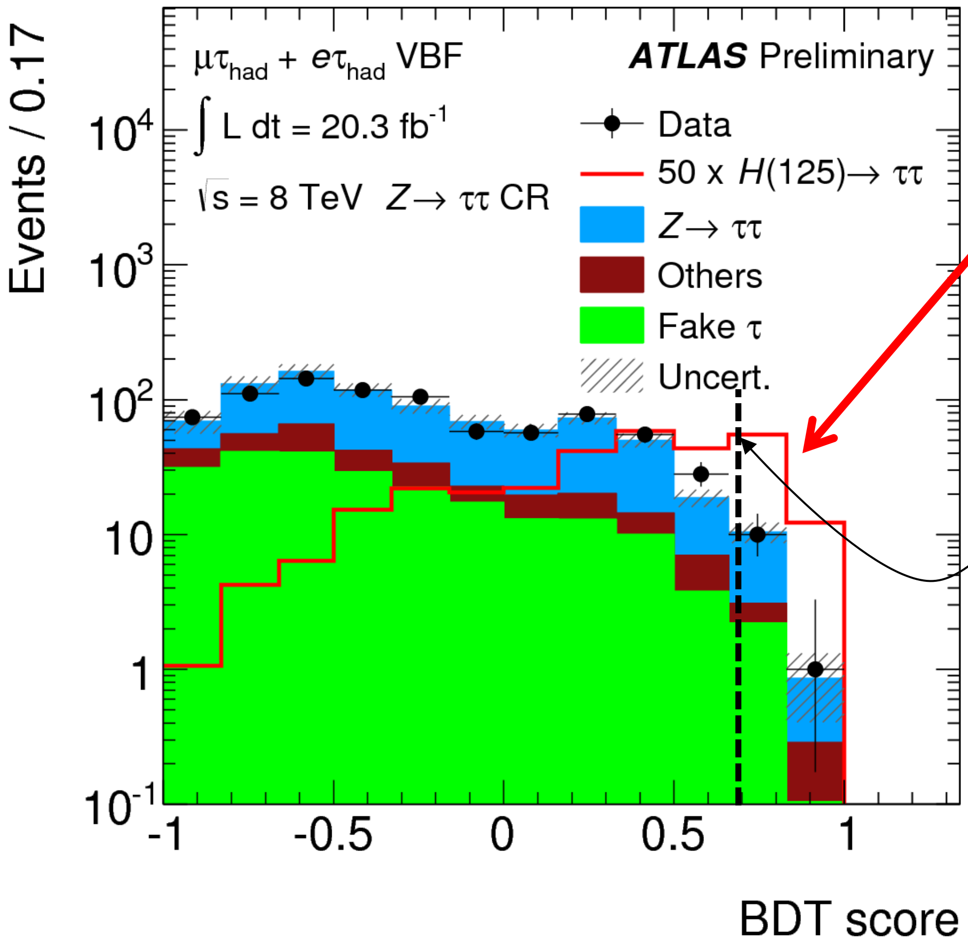
- Performance achieved without parameter tuning: PDERS and BDT best “out of the box” classifiers
- After specific tuning, also SVM und MLP perform well

- Theoretical maxir



Example of BDT use in Higgs

- Distribution of BDT score in search for for H→ττ events



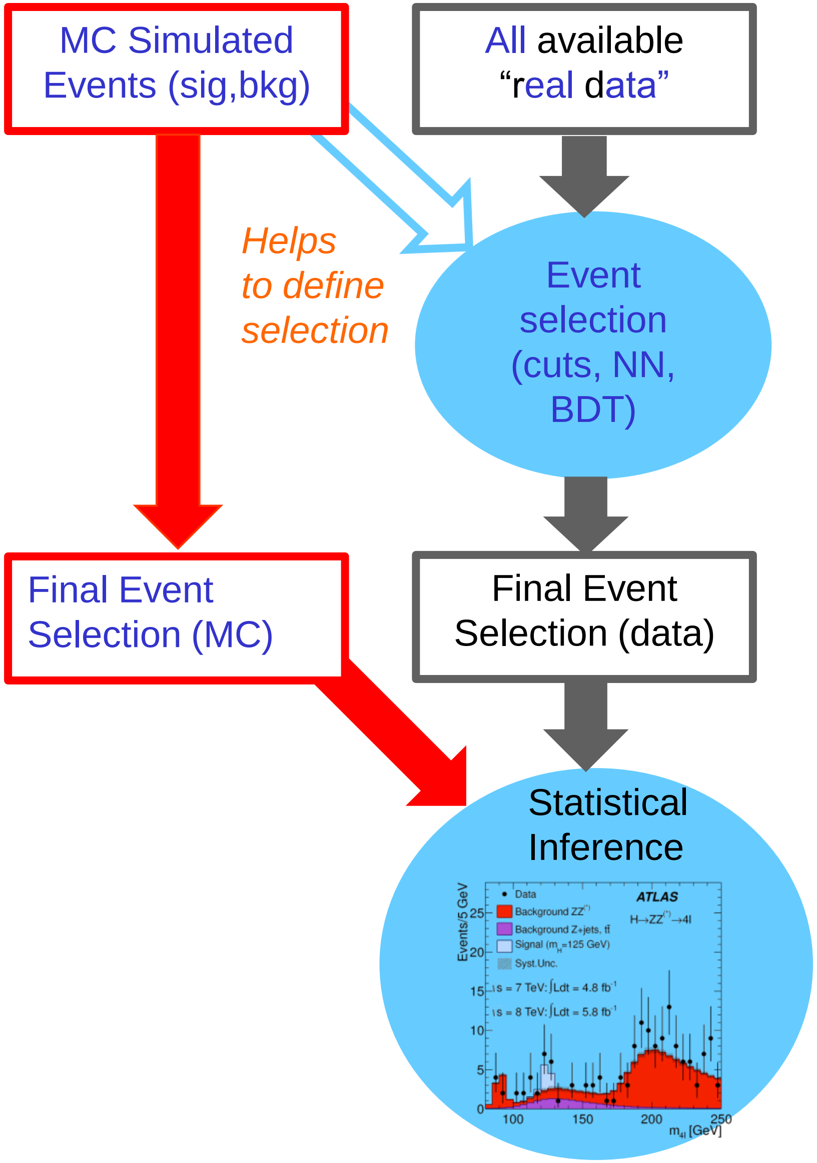
Signal expectation x 50

Simplest analysis strategy is to select events based on BDT score (e.g. BDT>0.7) and then perform a counting analysis,

but clearly throwing away some information (and signal events)

← Bkg-like events Signal-like events →

Outline of analysis procedure so far



MC Simulated Events (sig,bkg)

All available “real data”

Helps to define selection

Event selection (cuts, NN, BDT)

Final Event Selection (MC)

Final Event Selection (data)

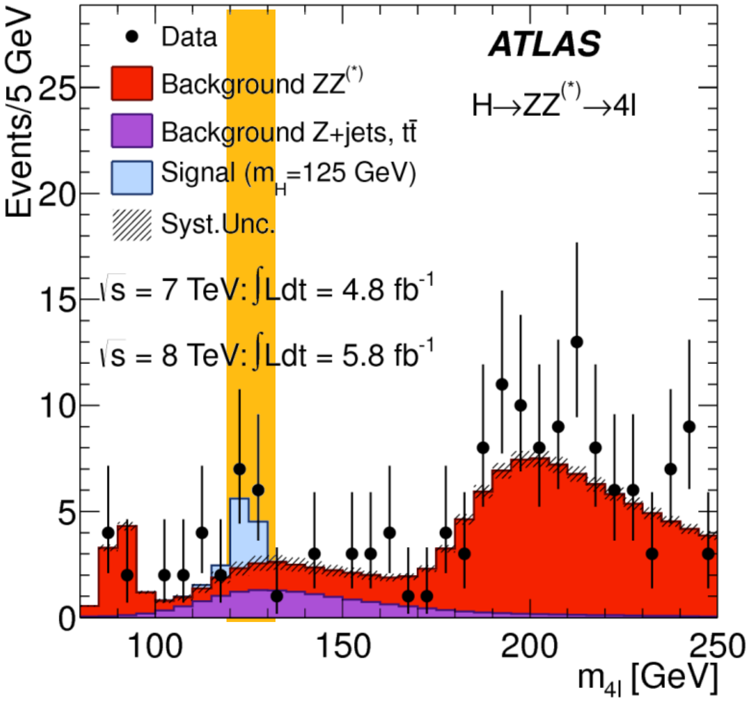
Statistical Inference

‘Hand-designed selection’
‘Machine-learned selection’
(with careful supervision)

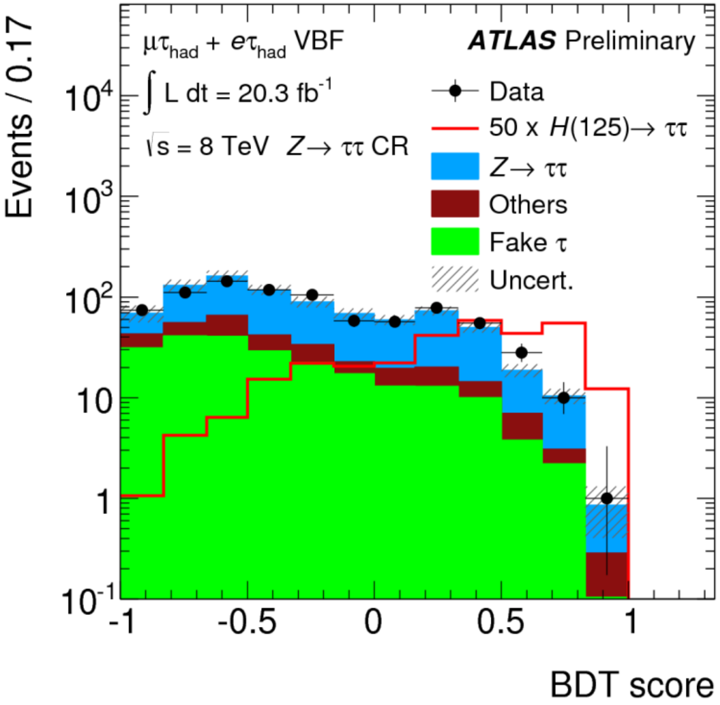
Event counting (so far)

Beyond event counting – exploiting all information

- Both H→ττ and H→ZZ searches illustrated that more discriminating information is available in each event that is used in the selection



Events in center of window have higher probability to be signal than at edge
→ ignored in counting analysis



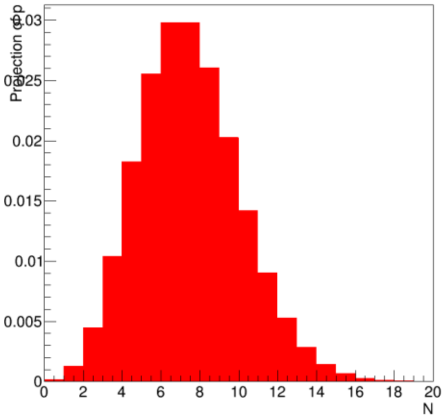
Hypothetical cut on BDT score (e.g. BDT>0.7) throws away some signal events

➔ Can we use all of this information to increase our sensitivity?

Beyond counting analysis – building *likelihood* models

- Can we include all such extra information in the statistical inference (i.e. calculation of the p-value)
- Example of probabilistic interpretation of results with dice and event counting were ‘light’ on mathematical detail.
- If we work out the math we see that include additional information is mathematically straightforward (although formulas for practical cases can become very complex)

Probability distribution for counting experiment



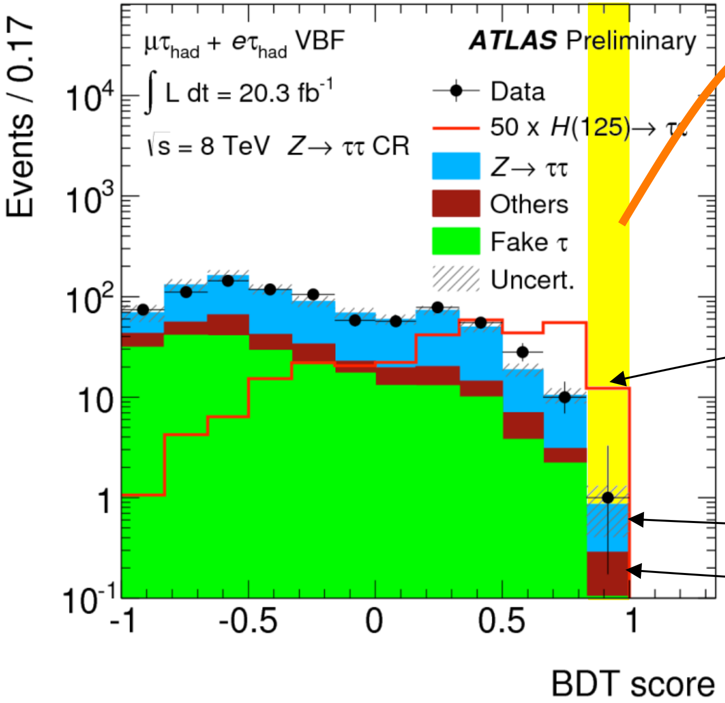
Observed event count

$$P(N|\lambda) = \frac{\lambda^N e^{-\lambda}}{N!}$$

Expected (average) count

Beyond counting analysis – building *likelihood* models

- How do we build a probability model for a histogram?
- Note that every bin is in effect a counting experiment



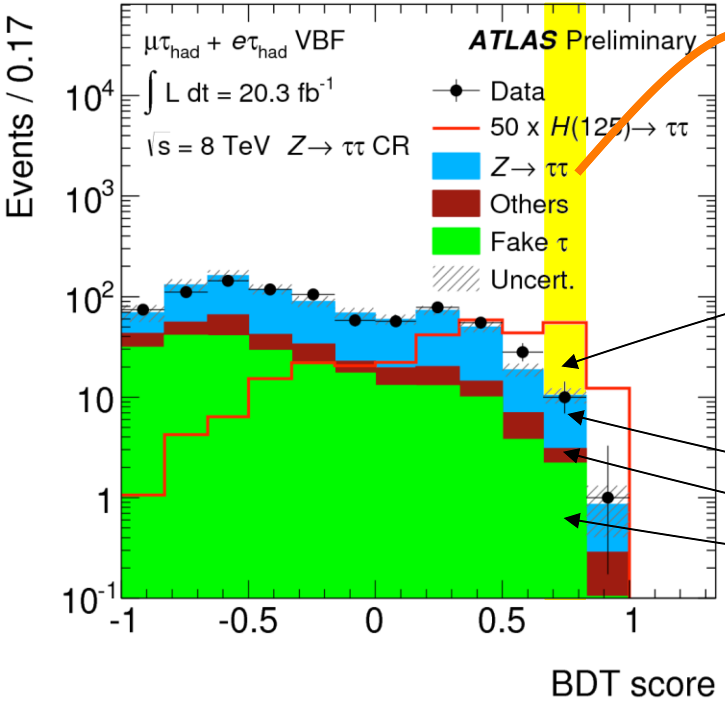
$$P(N\,|\,\lambda) = \frac{\lambda^N e^{-\lambda}}{N!}$$

$$= Poisson(N^i\,|\,\mu\times s^i + b_1^i + b_2^i + b_3^i)$$

Expected event rate is sum of expected signal and background rates

Beyond counting analysis – building *likelihood* models

- How do we build a probability model for a histogram?
- Note that every bin is in effect a counting experiment



$$P(N|\mu) = \frac{\mu^{N} e^{-\mu}}{N!}$$

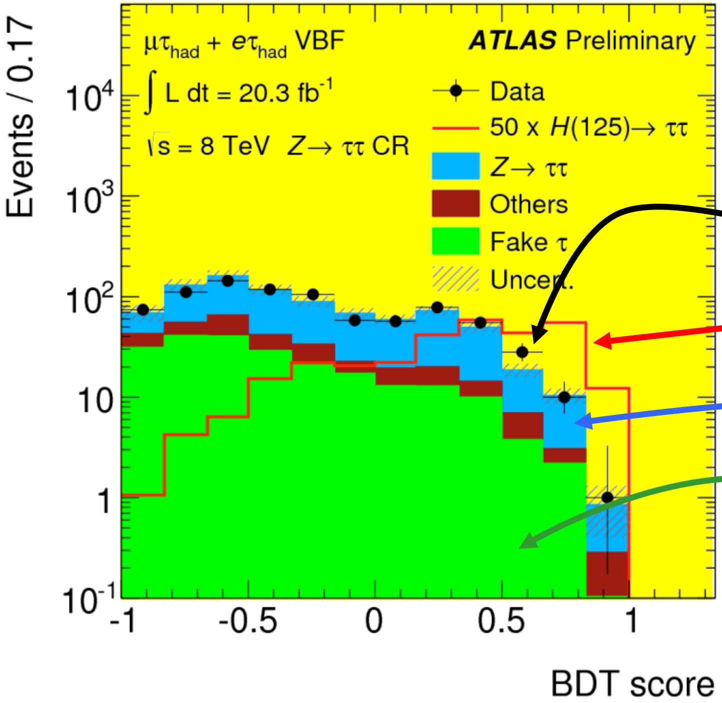
$$= Poisson(N^{i} | \mu \times s^{i} + b_{1}^{i} + b_{2}^{i} + b_{3}^{i})$$

Expected event rate is sum of expected signal and background rates

Beyond counting analysis – building *likelihood* models

- How do we build a probability model for a histogram?
- Note that every bin is in effect a counting experiment

$$P(\vec{N}\,|\,\vec{\lambda}) = Poisson(N_1\,|\,\lambda_1)\cdot Poisson(N_2\,|\,\lambda_2)....Poisson(N_n\,|\,\lambda_n)$$



$$P(\vec{N}\,|\,\mu,\vec{b}_1,\vec{b}_2,...) =$$

$$\prod_i Poisson(N^i\,|\,\mu\cdot s^i + b_1^i + b_2^i + ...)$$

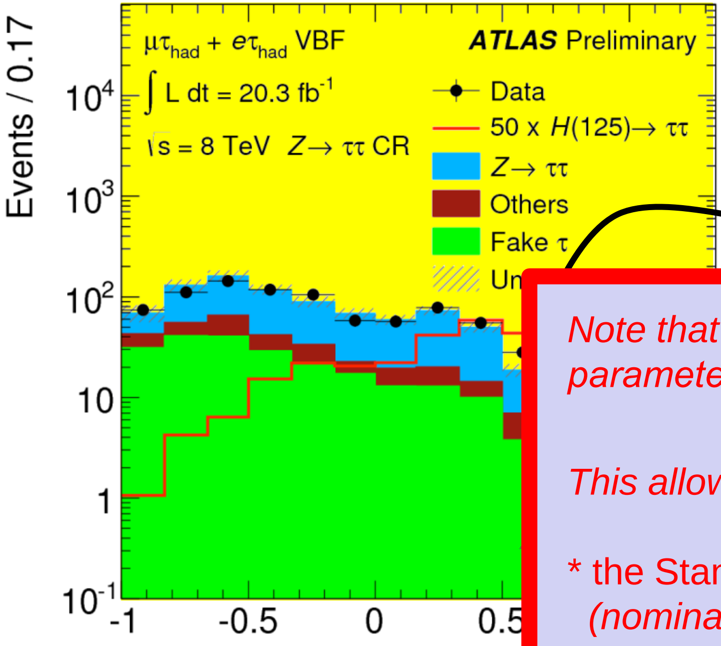
- Note: the function P(N|μ) is called the *Likelihood function*

Beyond counting analysis – building *likelihood* models

- How do we build a probability model for a histogram?

- Note that every bin is in effect a counting experimen t

$$P(\vec{N}|\vec{\lambda}) = Poisson(N_1|\lambda_1)\cdot Poisson(N_2|\lambda_2)....Poisson(N_n|\lambda_n)$$



$$P(\vec{N}|\mu,\vec{b}_1,\vec{b}_2,\ldots) = \prod_i Poisson(N^i|\mu\cdot s^i + b_1^i + b_2^i + \ldots)$$

Note that signal rate in function was cleverly written as a global scale parameter μ times the nominal prediction for each bin

This allows us to use the ***same*** *likelihood function to describe*

* the Standard Higgs model
(nominal signal + background → choose μ=1)

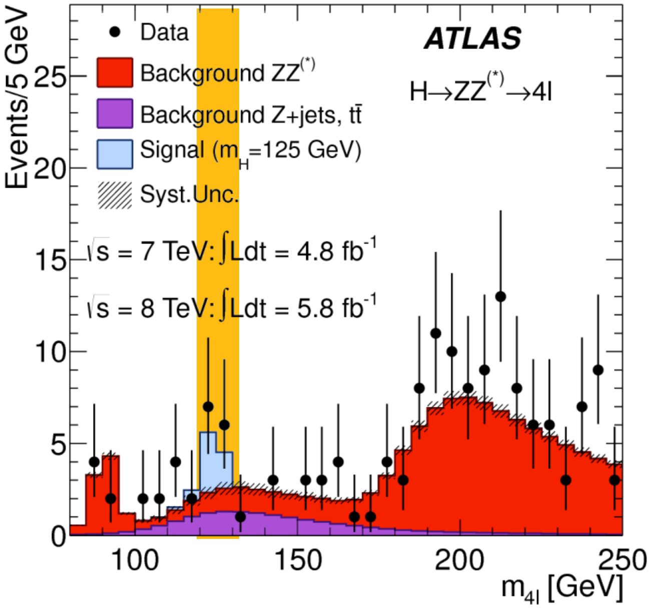
** The no-Higgs model*
(background only → choose μ=0)

** non-standard Higgs models*
(choose μ<1 or μ>1 to have less or more Higgs produced)

- Note: the function

Event counting for Higgs – example with ATLAS H→ZZ→lll signal

- Now apply calculation of probabilities of event counting to a realistic example: ATLAS H→ZZ→llll sample
- Count events in yellow band



N(observed) = 13

Expectation – no Higgs

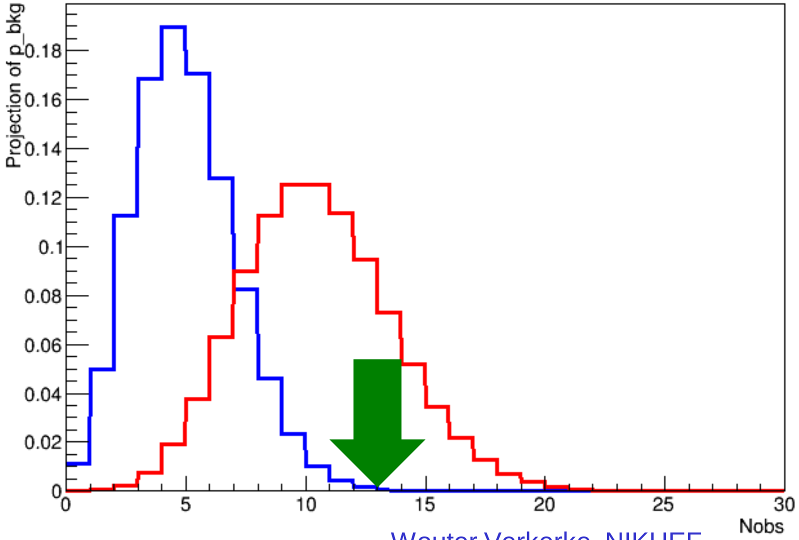
Poisson distribution with <N>=4.5

➔ prob(N≥13) = 0.08%

Expectation – SM Higgs

Poisson distribution with <N>=10

➔prob(N≥13) = 21%

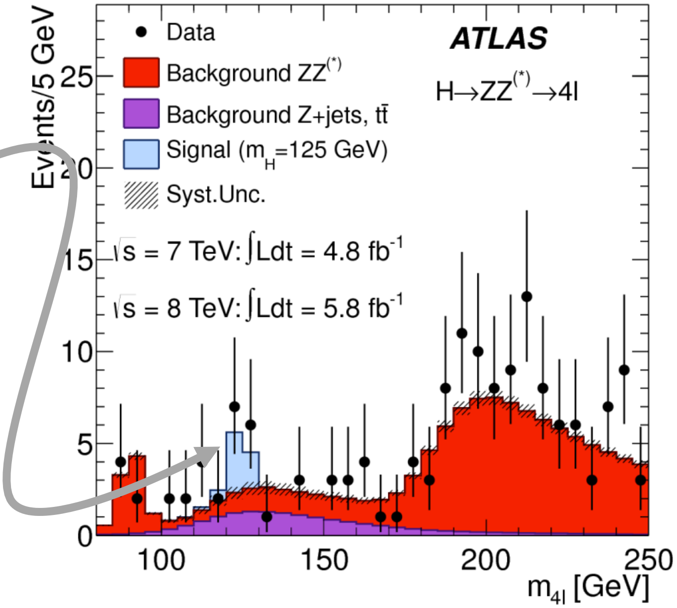


Wouter Verkerke, NIKHEF

Calculating the p-value for distributions

- For distributions replace N(observed) by a Likelihood ratio

$$\lambda_0(\vec{N}_{obs}) = \frac{L(\vec{N} \mid \mu = 0)}{L(\vec{N} \mid \mu = \hat{\mu})}$$



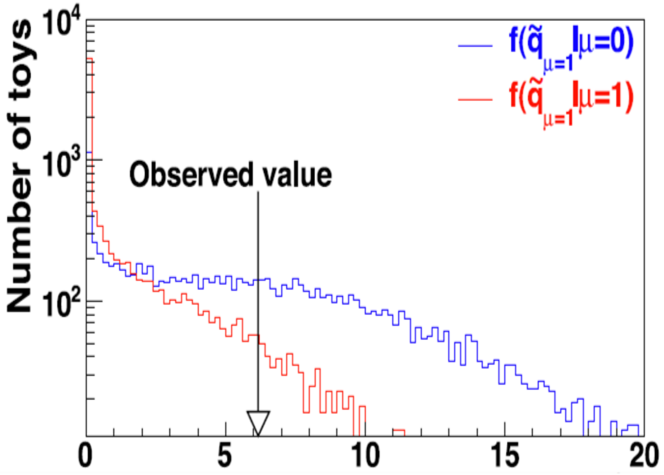
Numeric example:

$\log\lambda_0$(observed) = 6.8

Expectation – no Higgs

Asymptotically a 2log(χ^2) distribution

prob(q>…) = $p_{\chi 2}$(2*6.8,1)=0.02% (‘3.5 σ’)

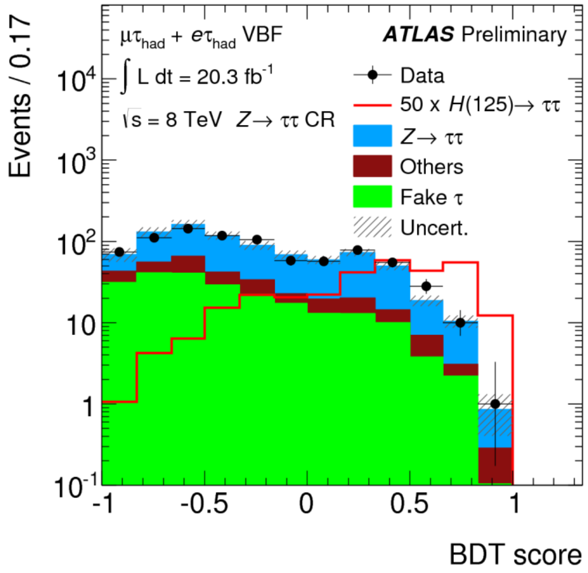


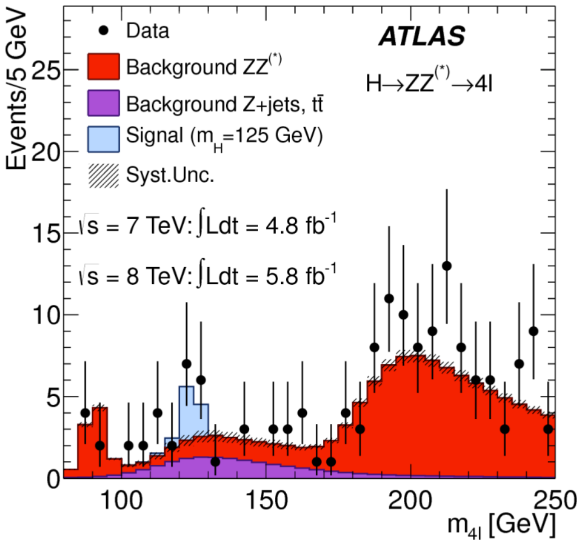
Expectation – SM Higgs

Asymptotically a non-central Chi-squared distribution

Putting even more information in

- So far I showed Likelihood functions that correspond to 1-dimensional distributions



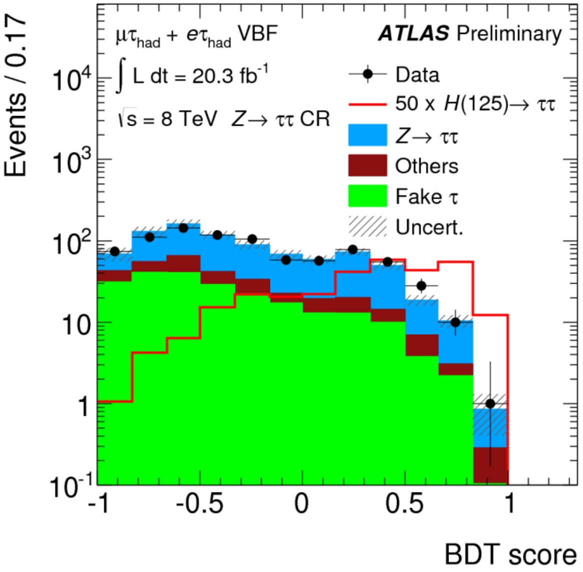


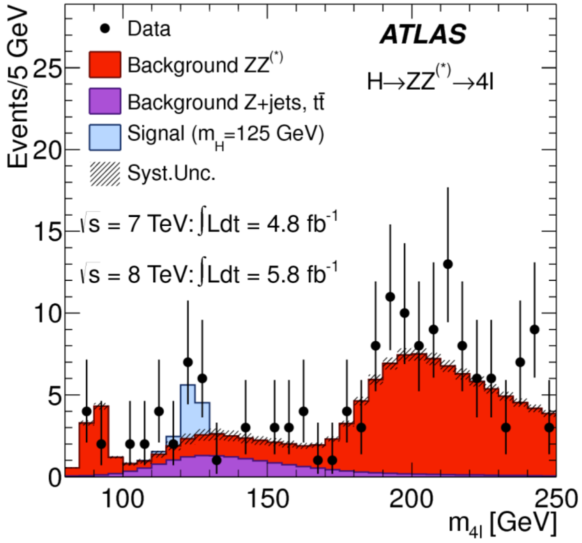
$$P(\vec{N} \mid \mu, \vec{b}_1, \vec{b}_2, \ldots) = \prod_i Poisson(N^i \mid \mu \cdot s^i + b_1^i + b_2^i + \ldots)$$

- But you can build much more complex models that look at many distributions simultaneously. Difficult to visualize, but also not needed, p-value (and discovery claim) only relies on you being able to calculate ratio of two likelihoods functions

Example combining information of H→ZZ and H→ττ

- If you have a Likelihood function for your H→ZZ analysis and a Likelihood function of your H→ττ analysis, you can combined both channels in a ‘joint likelihood’ by simply multiplying them (as we did for the bins within a channel)

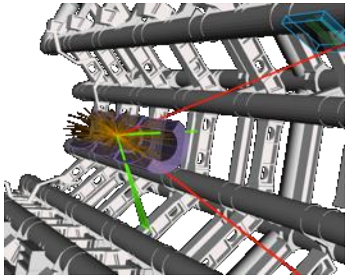




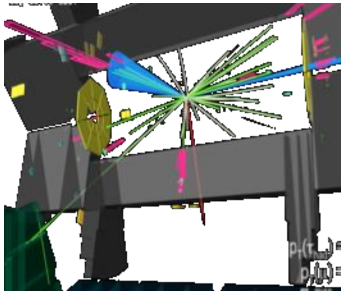
$$L(\vec{N}_{ZZ},\vec{N}_{\tau\tau}\mid\mu,\ldots)=L(\vec{N}_{ZZ}\mid\mu,\ldots)\cdot L(\vec{N}_{\tau\tau}\mid\mu,\ldots)$$

Higgs **discovery** strategy – add everything together

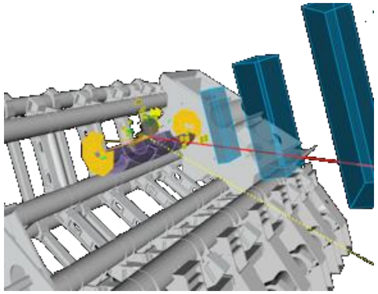
H→ZZ→llll



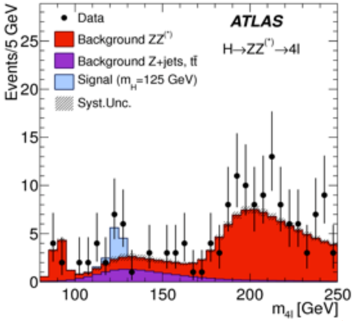
H→ττ

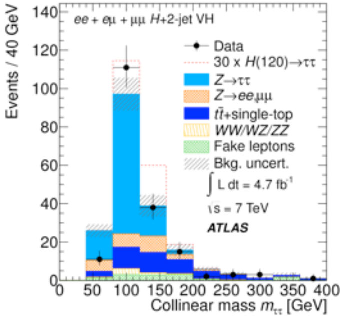


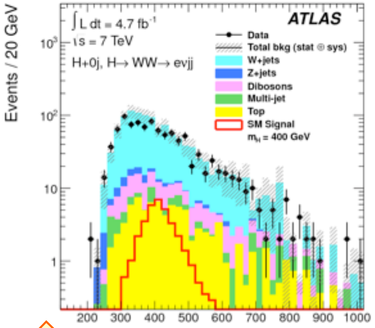
H→WW→μνjj



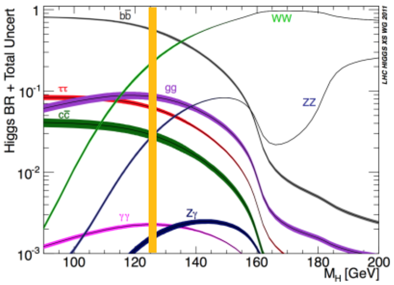
+…







Assume SM rates



Joint likelihood model for all observation channels

$$L(m,\vec{q}) = L_{H\to WW}(m_{WW},\vec{q})\cdot L_{H\to gg}(m_{gg},\vec{q})\cdot L_{H\to ZZ}(m_{ZZ},\vec{q})\cdot\square$$

Further complexity - dealing with systematic uncertainties

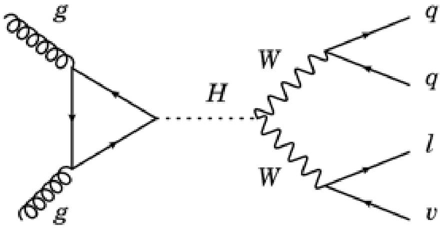
- So far all action was geared to improve sensitivity of analysis
  - Use Machine Learning to optimize event selection using many observables
  - Use complex Likelihood models to exploit additional information inside selected events, and to combine multiple channels together

- But we have so far ignored an important scientific aspect
  - Not all our knowledge about signal and background precise
  - Yet, both ML event selection, and Likelihood models so far treat information provided as ‘the exact truth’

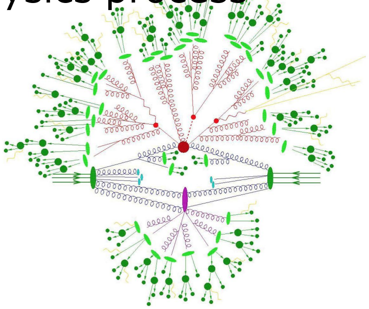
- Main scientific challenge – incorporate effect of ‘systematic uncertainties’ into the analysis (and probability calculations)

Understanding signal and background

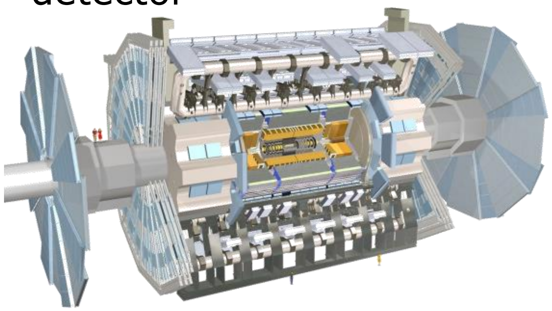
Simulation of high-energy physics process



Simulation of 'soft physics' physics process

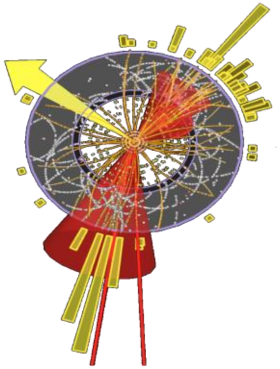


Simulation of ATLAS detector

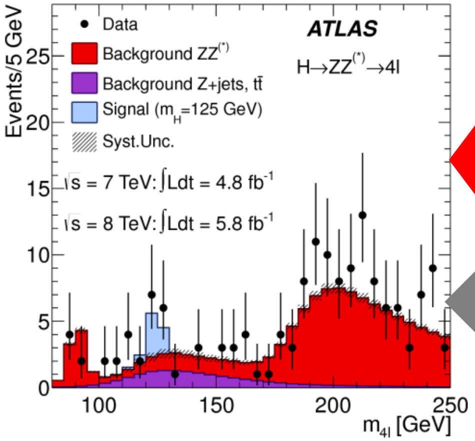


LHC data

Reconstruction of ATLAS detector

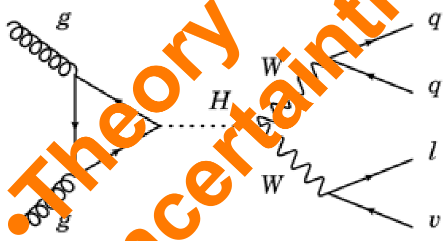


Analysis Event selection



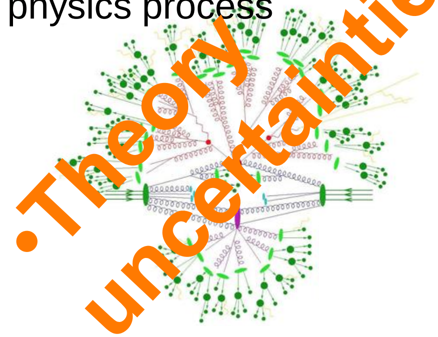
The simulation workflow and origin of uncertainties

Simulation of high-energy physics process



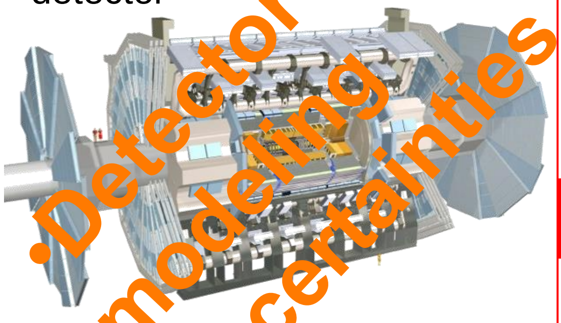
•Theory uncertainties

Simulation of ‘soft physics’ physics process



•Theory uncertainties

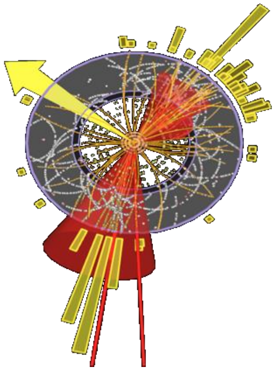
Simulation of ATLAS detector



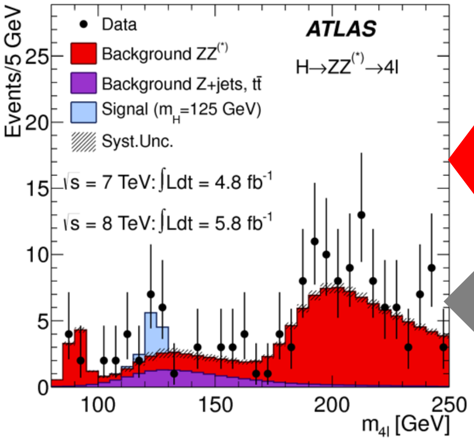
•Detector modeling uncertainties

LHC data

Reconstruction of ATLAS detector



Analysis Event selection



Wouter Verkerke, NIKHEF

Typical systematic uncertainties in HEP

- Detector-simulation related
  - “The Jet Energy scale uncertainty is 5%”
  - “The b-tagging efficiency uncertainty is 20% for jets with p_T<40”
- Physics/Theory related
  - The top cross-section uncertainty is 8%
  - “Vary the factorization scale by a factor 0.5 and 2.0 and consider the difference the systematic uncertainty”
  - “Evaluate the effect of using Herwig and Pythia and consider the difference the systematic uncertainty”
- MC simulation statistical uncertainty
  - Effect of (bin-by-bin) statistical uncertainties in MC samples

How do we take uncertainties into account

- In (Machine Learned) event selections
  - Essentially very difficult.
  - Main strategy – only use ‘safe’ observables in selection process (those with little uncertainty on them), and make selection not to tight (so that a small shift in e.g. a calibration does not change the fraction of selected signal by much)
- In Likelihood-based calculation of p-values
  - In principle straightforward!
  - Likelihood models can have parameters that can be weakly that represent the known systematic uncertainties on various quantities

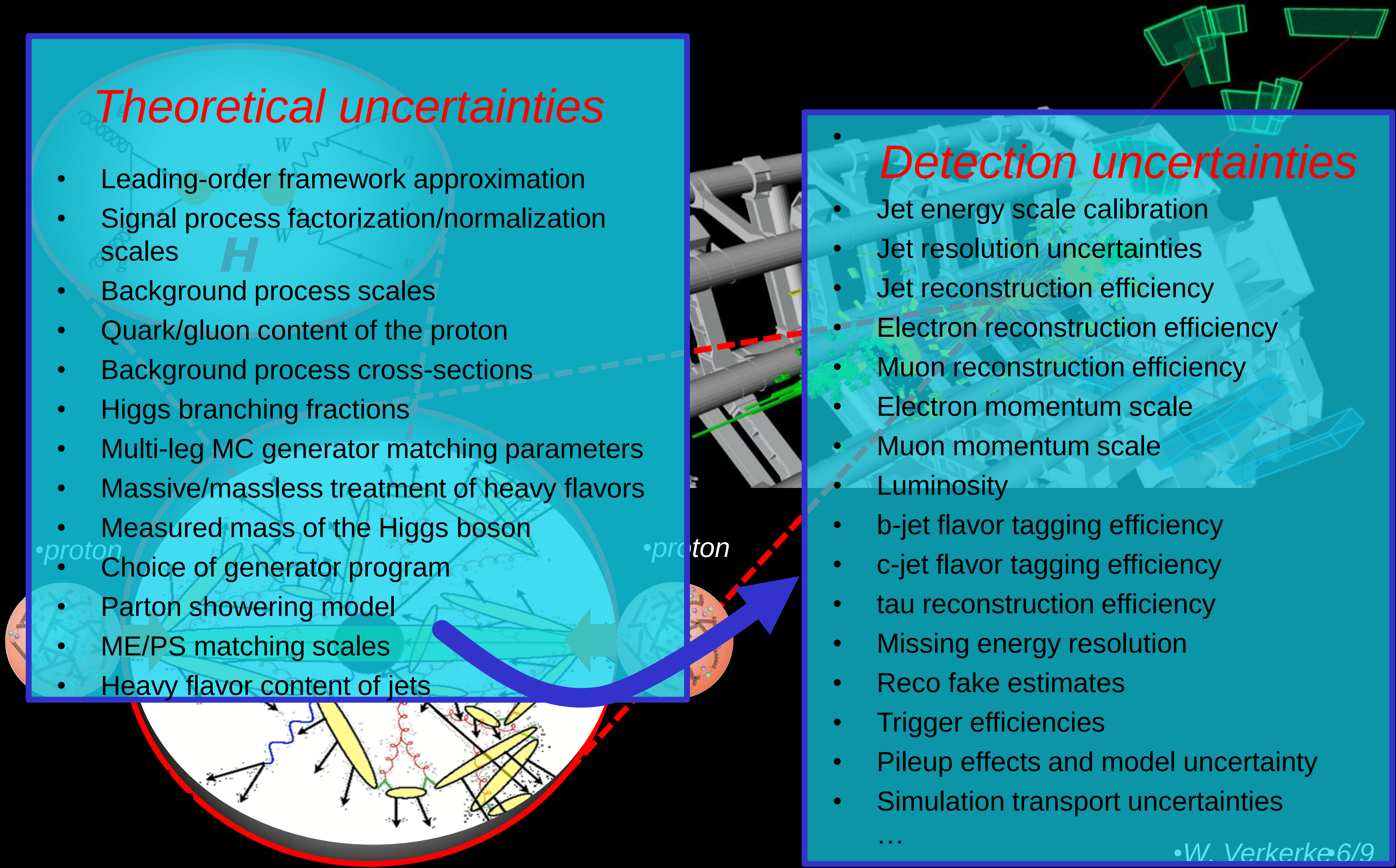
Likelihood model for counting experiment with exactly known background

$$L(m) = Poisson(N_{SR} \,|\, m \times s + b)$$

Likelihood model for counting experiment with 8% uncertainty on background

$$L(\mu, b) = Poisson(N_{SR} \,|\, \mu \cdot s + b) \cdot Gauss(\tilde{b} \,|\, b, 0.08)$$

We have many systematic uncertainties!



Theoretical uncertainties

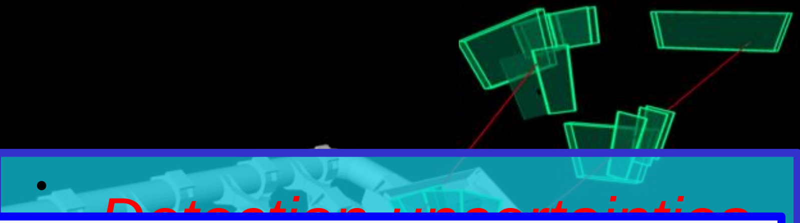
- Leading-order framework approximation
- Signal process factorization/normalization scales
- Background process scales
- Quark/gluon content of the proton
- Background process cross-sections
- Higgs branching fractions
- Multi-leg MC generator matching parameters
- Massive/massless treatment of heavy flavors
- Measured mass of the Higgs boson
- Choice of generator program
- Parton showering model
- ME/PS matching scales
- Heavy flavor content of jets

Detection uncertainties

- Jet energy scale calibration
- Jet resolution uncertainties
- Jet reconstruction efficiency
- Electron reconstruction efficiency
- Muon reconstruction efficiency
- Electron momentum scale
- Muon momentum scale
- Luminosity
- b-jet flavor tagging efficiency
- c-jet flavor tagging efficiency
- tau reconstruction efficiency
- Missing energy resolution
- Reco fake estimates
- Trigger efficiencies
- Pileup effects and model uncertainty
- Simulation transport uncertainties

…

We have many systematic uncertainties!



Theoretical uncertainties

Detection uncertainties

Mathematical form of Likelihood model will get very complex

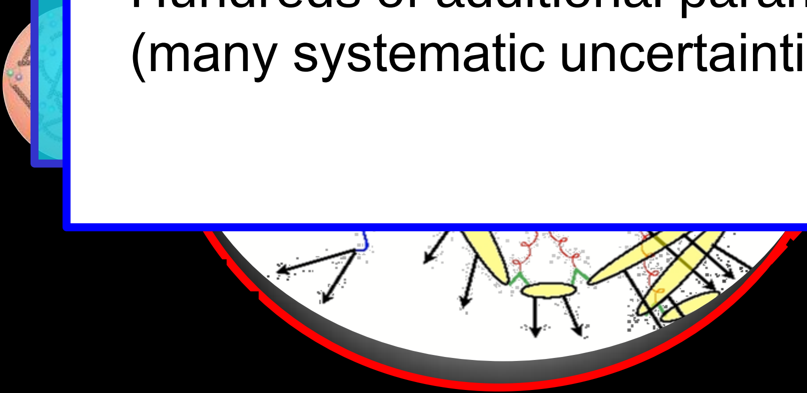
Likelihood model for counting experiment with exactly known background

$$L(\mu) = Poisson(N_{SR} \mid \mu \cdot s + b)$$

Likelihood model for counting experiment with 8% uncertainty on background

$$L(\mu,b) = Poisson(N_{SR} \mid \mu \cdot s + b) \cdot Gauss(\tilde{b} \mid b, 0.08)$$

Hundreds of additional parameters modeling systematic uncertainties (many systematic uncertainties require >1 parameter)…

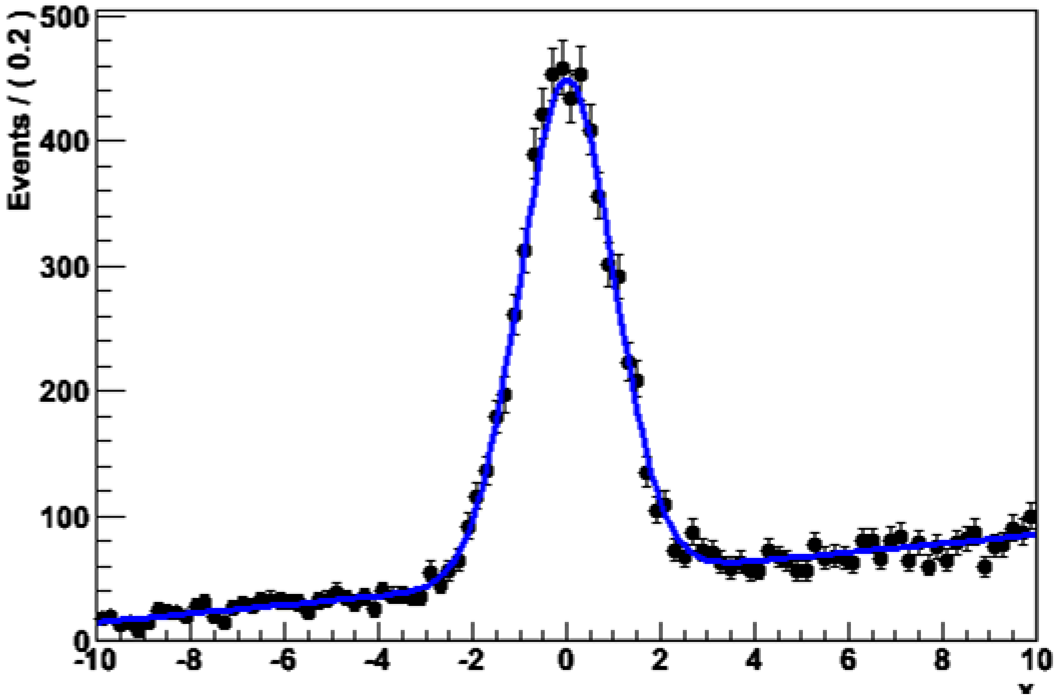


- Pileup effects and model uncertainty
- Simulation transport uncertainties

…

RooFit – Focus: coding a probability density function

- Focus on one practical aspect of many data analysis in HEP:
  How do you formulate your p.d.f. in ROOT
  - For ‘simple’ problems (gauss, polynomial) this is easy



  - But if you want to do unbinned ML fits, use non-trivial functions, or work with multidimensional functions you quickly find that you need some tools to help you

RooFit – a toolkit to formulate probability models in C++

- Key concept: represent individual elements of a mathematical model by separate C++ objects

Mathematical concept		RooFit class
variable	x, p	`RooRealVar`
function	$f(\vec{x})$	`RooAbsReal`
PDF	$F(\vec{x};\vec{p},\vec{q})$	`RooAbsPdf`
space point	$\vec{x}$	`RooArgSet`
integral	$\int_{x_{\min}}^{x_{\max}} f(x)dx$	`RooRealIntegral`
list of space points	$\vec{x}_k$	`RooAbsData`

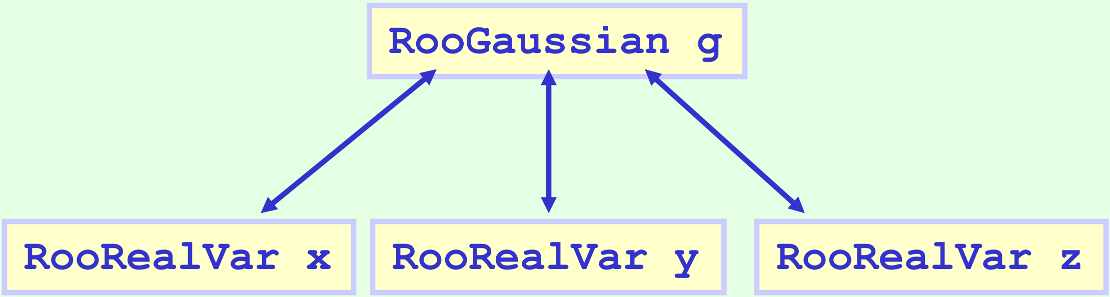
RooFit core design philosophy

- Build likelihood function out of many small software objects, rather than a monolithical `double L(double * params)` function

Math

$\mathrm{Gauss}(x,\mu,\sigma)$

RooFit diagram



RooFit code

```
RooRealVar x(“x”,”x”,-10,10) ;
RooRealVar m(“m”,”y”,0,-10,10) ;
RooRealVar s(“s”,”z”,3,0.1,10) ;
RooGaussian g(“g”,”g”,x,m,s) ;
```

RooFit core design philosophy

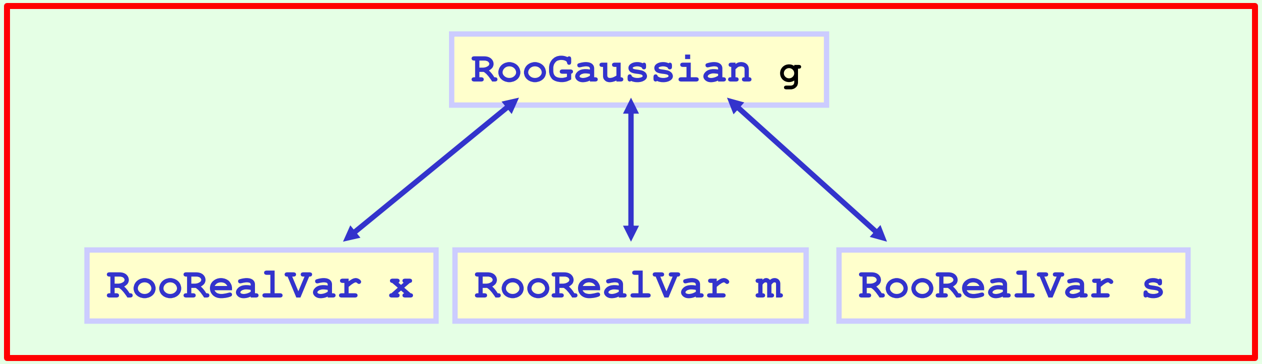
- Build likelihood function out of many small software objects, rather than a monolithical `double L(double * params)` function

Math

Gauss(x,μ,σ)

RooWorkspace (keeps all parts together)

RooFit diagram



RooFit code

```
RooRealVar x(“x”,”x”,-10,10) ;
RooRealVar m(“m”,”y”,0,-10,10) ;
RooRealVar s(“s”,”z”,3,0.1,10) ;
RooGaussian g(“g”,”g”,x,m,s) ;
RooWorkspace w(“w”) ;
w.import(g) ;
```

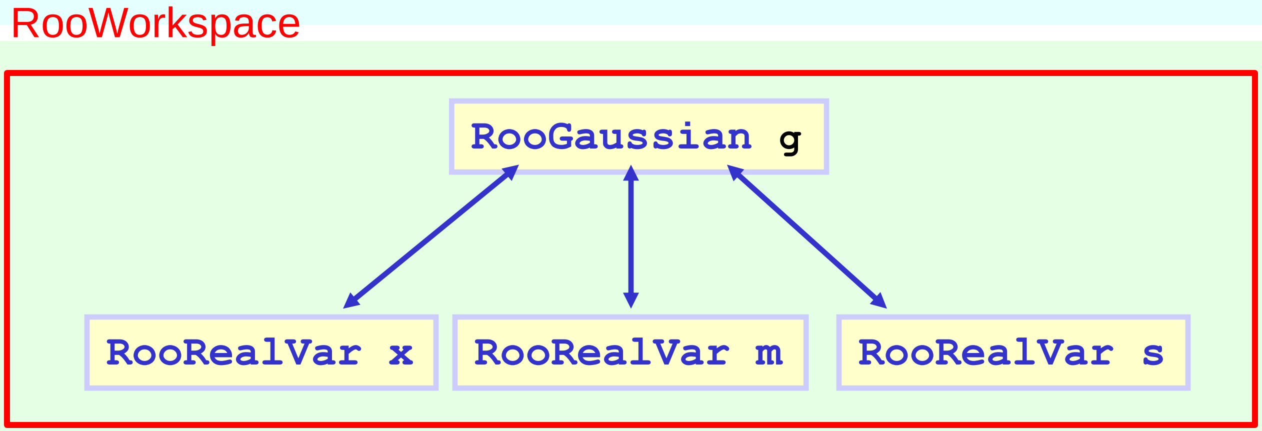
RooFit core design philosophy - Workspace

- Alternatively, a simple math-like ‘factory language’ can quickly populates a workspace with the same objects

Math

Gauss(x,μ,σ)

RooFit diagram



RooFit code

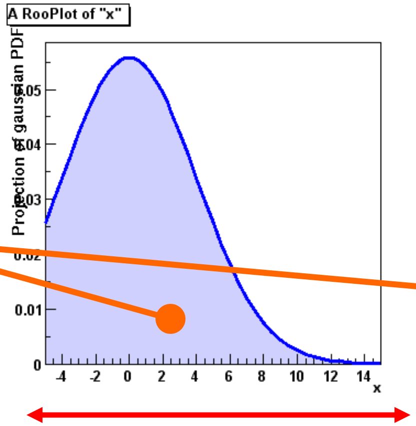
```
RooWorkspace w(“w”) ;
w.factory(“Gaussian::g(x[-10,10],m[0],s[5])”) ;
```

RooFit implements *normalized* probability models

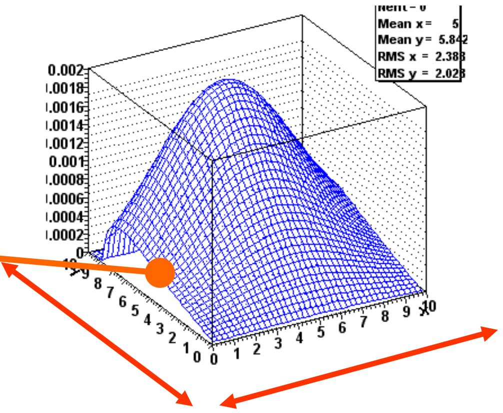
- Normalized probability (density) models are the basis of all fundamental statistical techniques
  - Defining feature:

$$\int f(\vec{x},\vec{p})\,d\vec{x} \equiv 1,$$

$$f(\vec{x},\vec{p}) \geq 0$$



$$\int F(x)dx \equiv 1$$

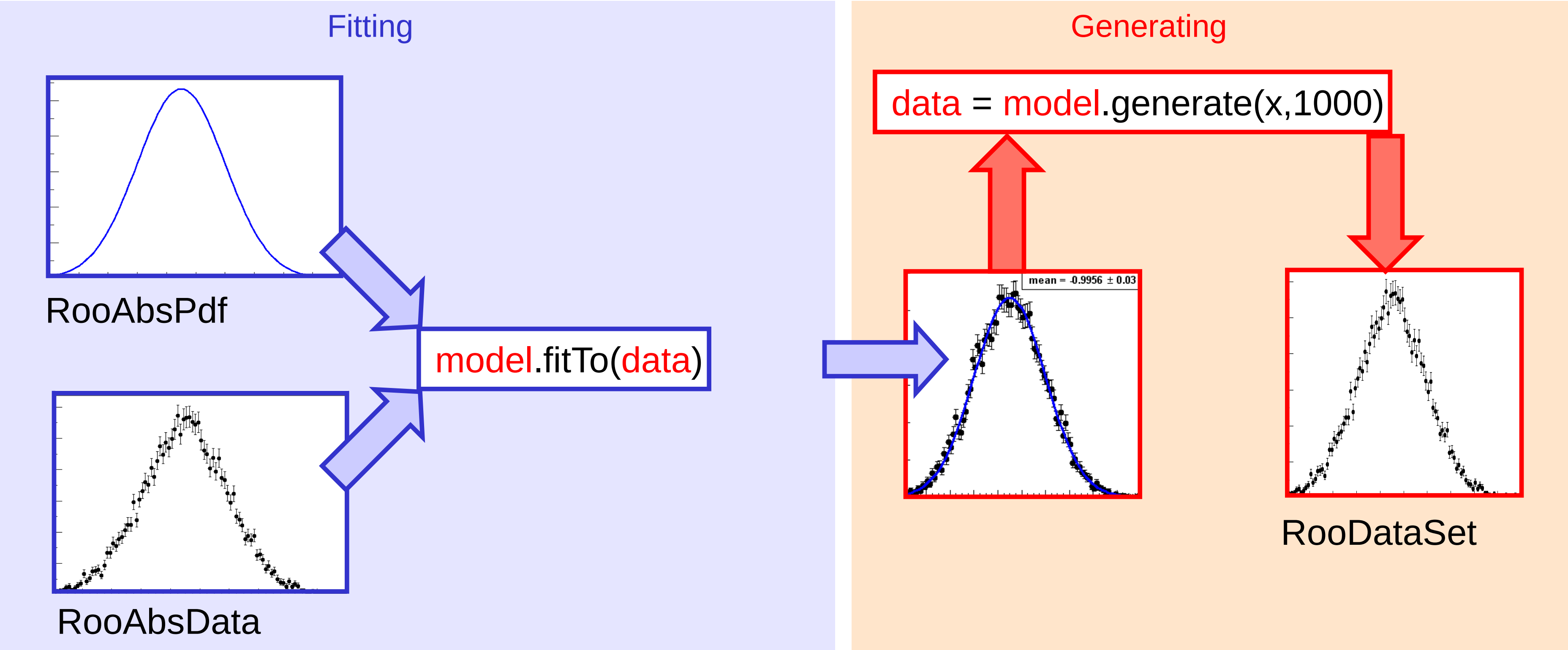


$$\int F(x,y)dxdy \equiv 1$$

- Normalization guarantee introduces extra complication in calculation, but has important advantages
  - Directly usable in fundamental statistical techniques
  - Easier construction of complex models (will shows this in moment)
- RooFit provides built-in support for normalization, taking away down-side for users, leaving upside
  - Default normalization strategy relies on numeric techniques, but user can specify known (partial) analytical integrals in pdf classes.

Abstract interfaces make fitting and toy generation easy

- Can make fits of models to data, and generate simulated data from toys with one-line comments, regardless of model complexity



Fitting

RooAbsPdf

RooAbsData

```
model.fitTo(data)
```

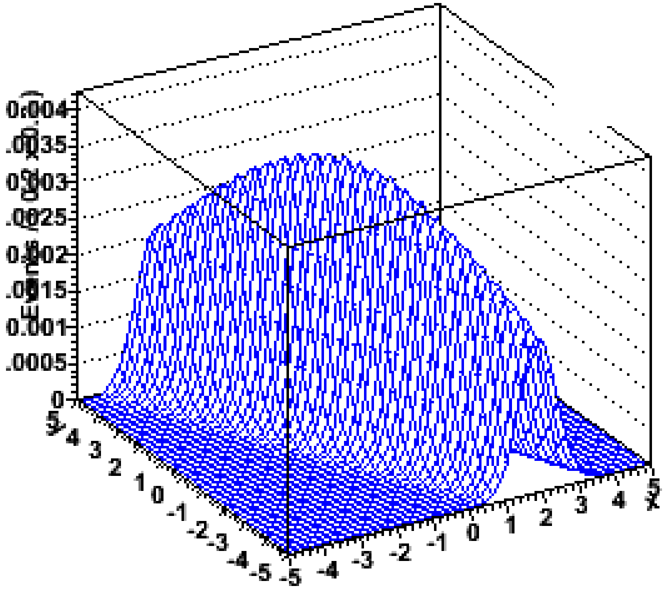
Generating

```
data = model.generate(x,1000)
```

RooDataSet

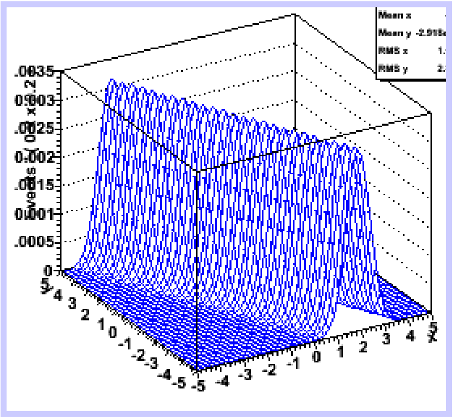
The power of *conditional* probability modeling

- Take following model f(x,y):
  what is the analytical form?

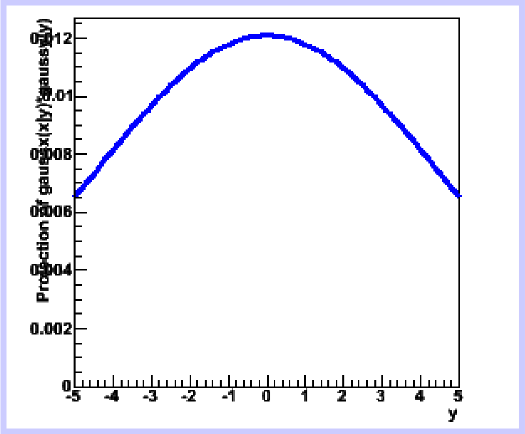


- Trivially constructed with (conditional) probability density functions!

Gauss f(x|a*y+b,1)



Gauss g(y,0,3)

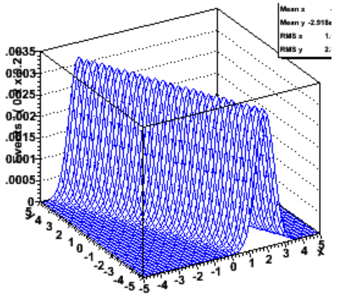


F(x,y) = f(x|y)*g(y)

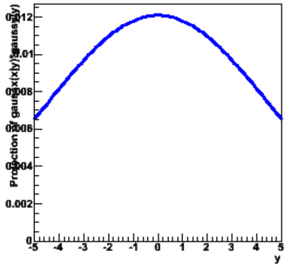
Coding a conditional product model in RooFit

- Construct each ingredient with a single line of code

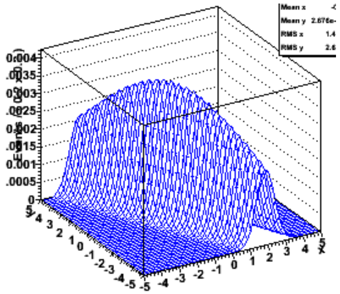
Gauss f(x,a*y+b,1)



Gauss g(y,0,3)



F(x,y) = f(x|y)*g(y)

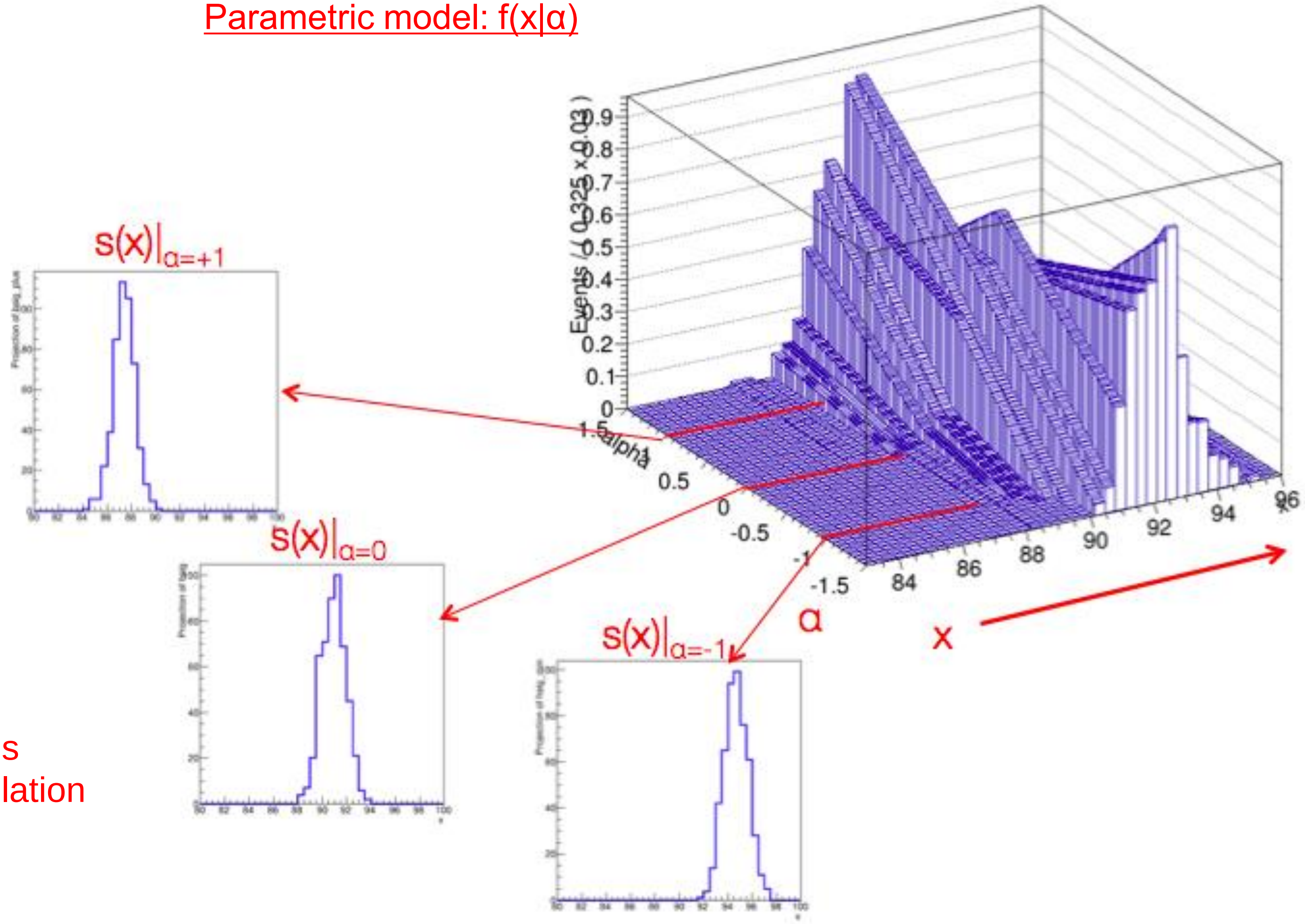


```
RooRealVar x(“x”,”x”,-10,10) ;
RooRealVar y(“y”,”y”,-10,10) ;
RooRealVar a(“a”,”a”,0) ;
RooRealVar b(“b”,”b”,-1.5) ;

RooFormulaVar m(“a*y+b”,a,y,b) ;
RooGaussian f(“f”,”f”,x,m,C(1)) ;

RooGaussian g(“g”,”g”,y,C(0),C(3)) ;

RooProdPdf F(“F”,”F”,g,Conditional(f,y)) ;
```

Note that code doesn’t care if input expression is variable or function!

Advanced modeling building – template morphing

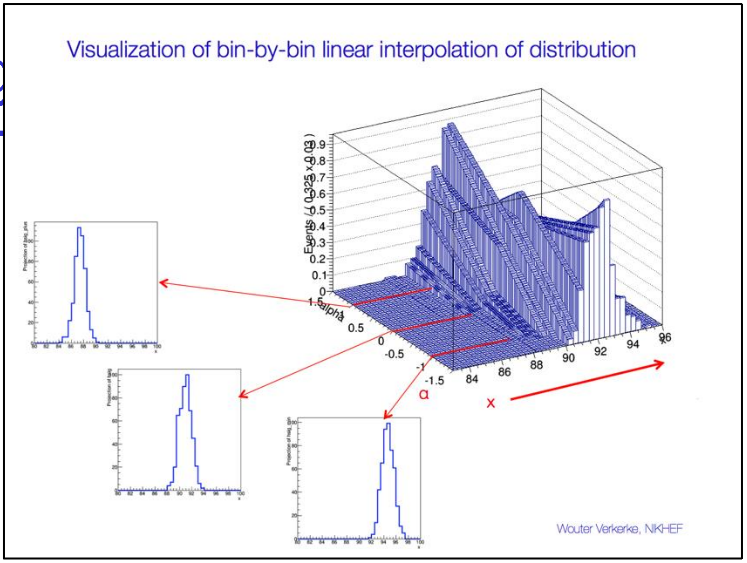
- At LHC shapes are often derived from histograms, instead of relying on analytical shapes . Construct parametric from histograms using ‘template morphing’ techniques



Parametric model: f(x|α)

Input histograms from simulation

Code example – template morphing


Visualization of bin-by-bin linear interpolation of distribution

- Example of template morphing systematic in a binned likelihood

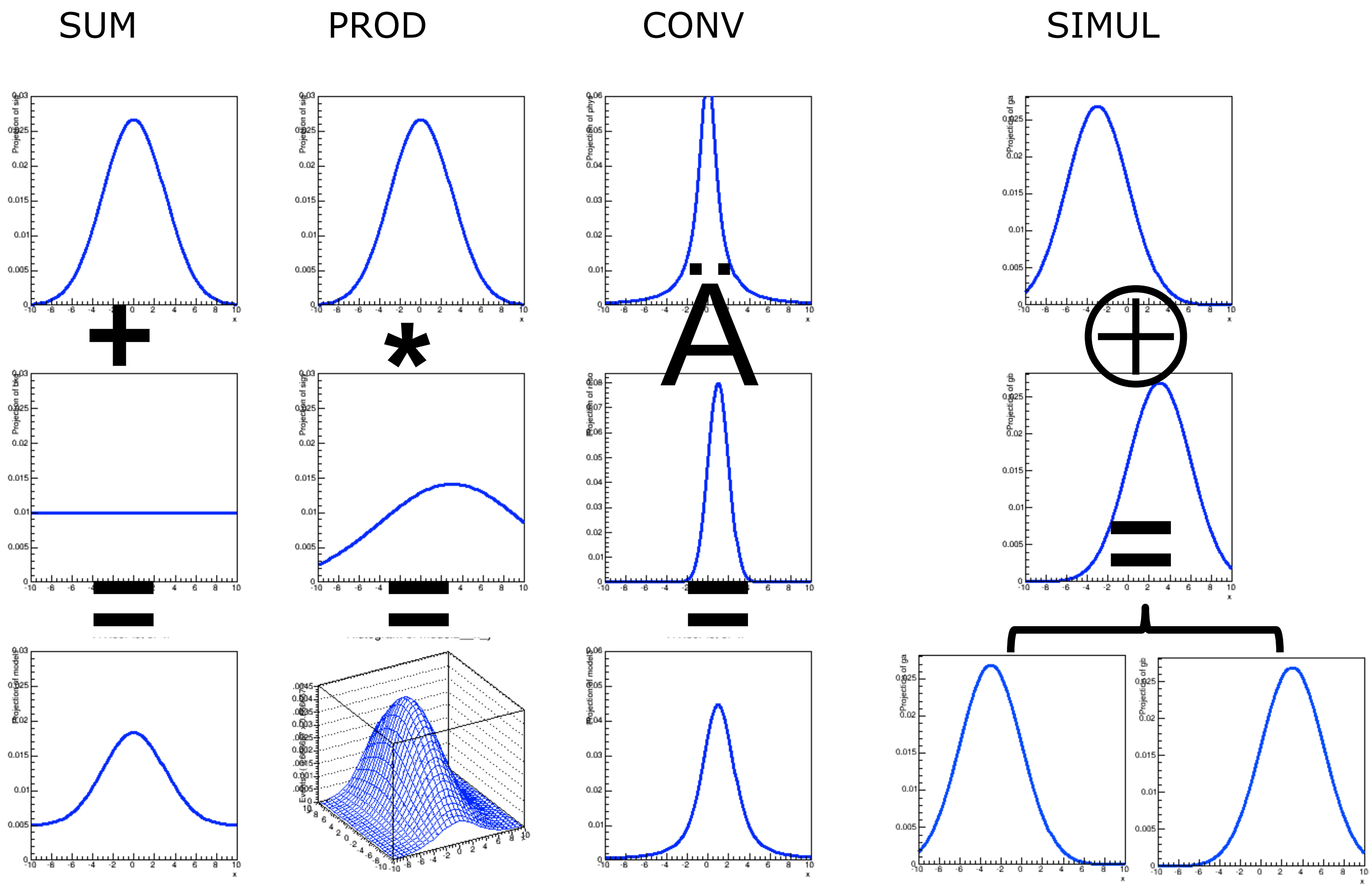
$$s_i(\alpha,\ldots) = \begin{cases} s_i^0 + \alpha \cdot (s_i^+ - s_i^0) & \forall\ \alpha > 0 \\ s_i^0 + \alpha \cdot (s_i^0 - s_i^-) & \forall\ \alpha < 0 \end{cases}$$

$$L(\vec{N} \mid \alpha, \vec{s}^-, \vec{s}^0, \vec{s}^+) = \prod_{bins} P(N_i \mid \underbrace{s_i(\alpha, s_i^-, s_i^0, s_i^+)}_{}) \cdot \underbrace{G(0 \mid \alpha, 1)}_{}$$

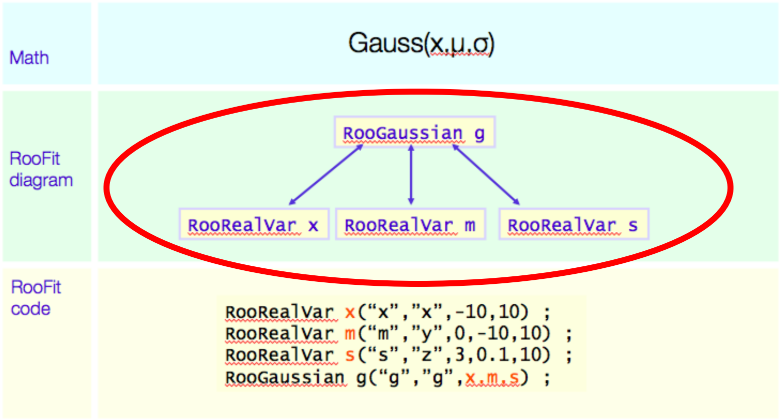
```
// Construct template models from histograms
w.factory("HistFunc::s_0(x[80,100],hs_0)") ;
w.factory("HistFunc::s_p(x,hs_p)") ;
w.factory("HistFunc::s_m(x,hs_m)") ;

// Construct morphing model
w.factory("PiecewiseInterpolation::sig(s_0,s_,m,s_p,alpha[-5,5])") ;

// Construct full model
w.factory("PROD::model(ASUM(sig,bkg,f[0,1]),Gaussian(0,alpha,1))") ;
```

From simple to realistic models: composition techniques

- Realistic models with signal and bkg, and with control regions built from basic shapes using *addition*, *product, convolution, simultaneous* operator classes

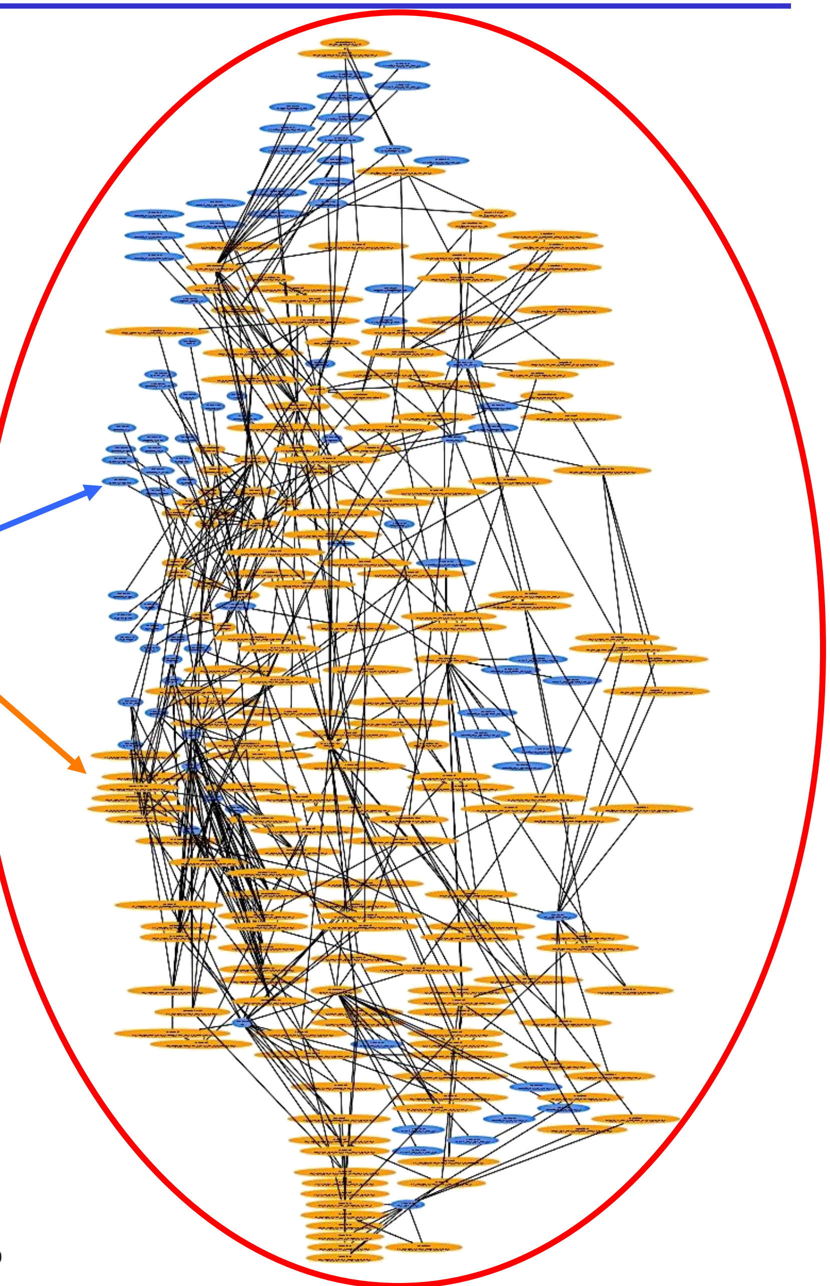


Graphical example of realistic complex models



Math	Gauss(x,μ,σ)
RooFit diagram	RooGaussian g RooRealVar x RooRealVar m RooRealVar s

RooFit code	`RooRealVar x("x","x",-10,10) ;`
	`RooRealVar m("m","y",0,-10,10) ;`
	`RooRealVar s("s","z",3,0.1,10) ;`
	`RooGaussian g("g","g",x,m,s) ;`

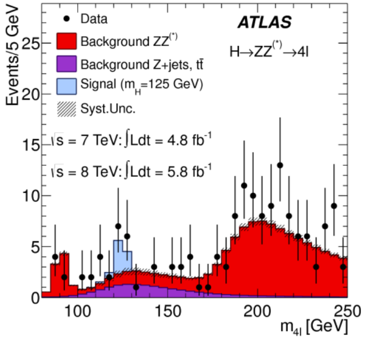


variables

function objects

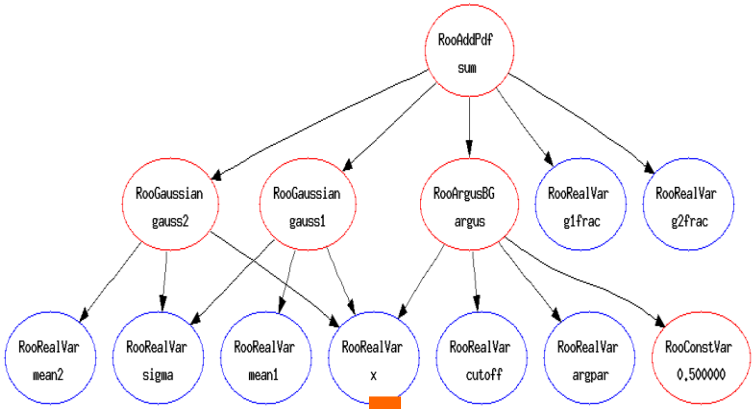
Expression graphs are autogenerated using

```
pdf->graphVizTree("file.dot")
```



Abstracting model building from model use - 2

- Must be able to *practically* separate model building code from statistical analysis code.
- Solution: you can *persist* RooFit models of arbitrary complexity in ‘workspace’ containers
- The workspace concept has revolutionized the way people share and combine analyses!

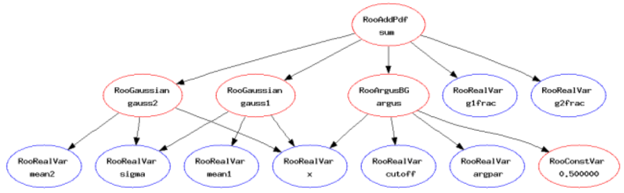


Realizes complete and practical factorization of process of building and using likelihood functions!

```
RooWorkspace w(“w”) ;
w.import(sum) ;
w.writeToFile(“model.root”) ;
```

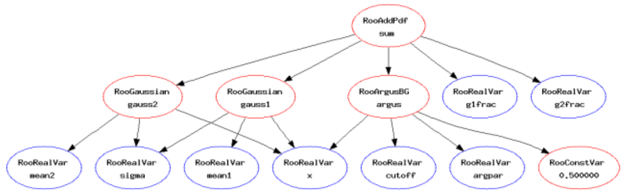
model.root

RooWorkspace



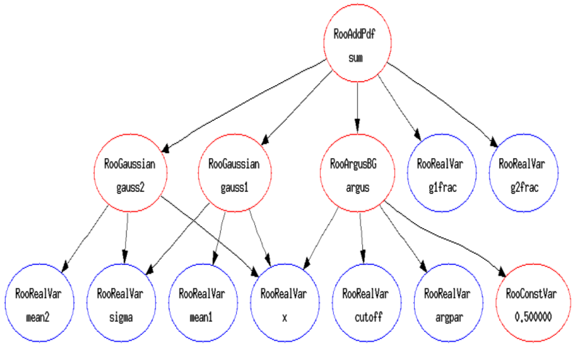
Using a workspace file given to you…

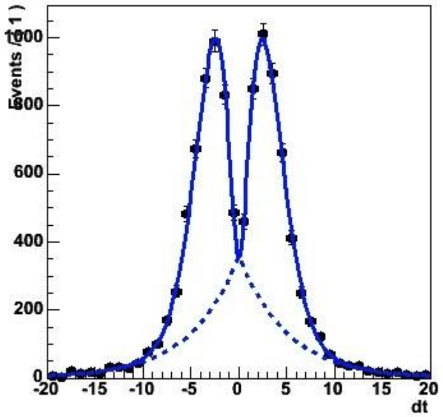
RooWorkspace



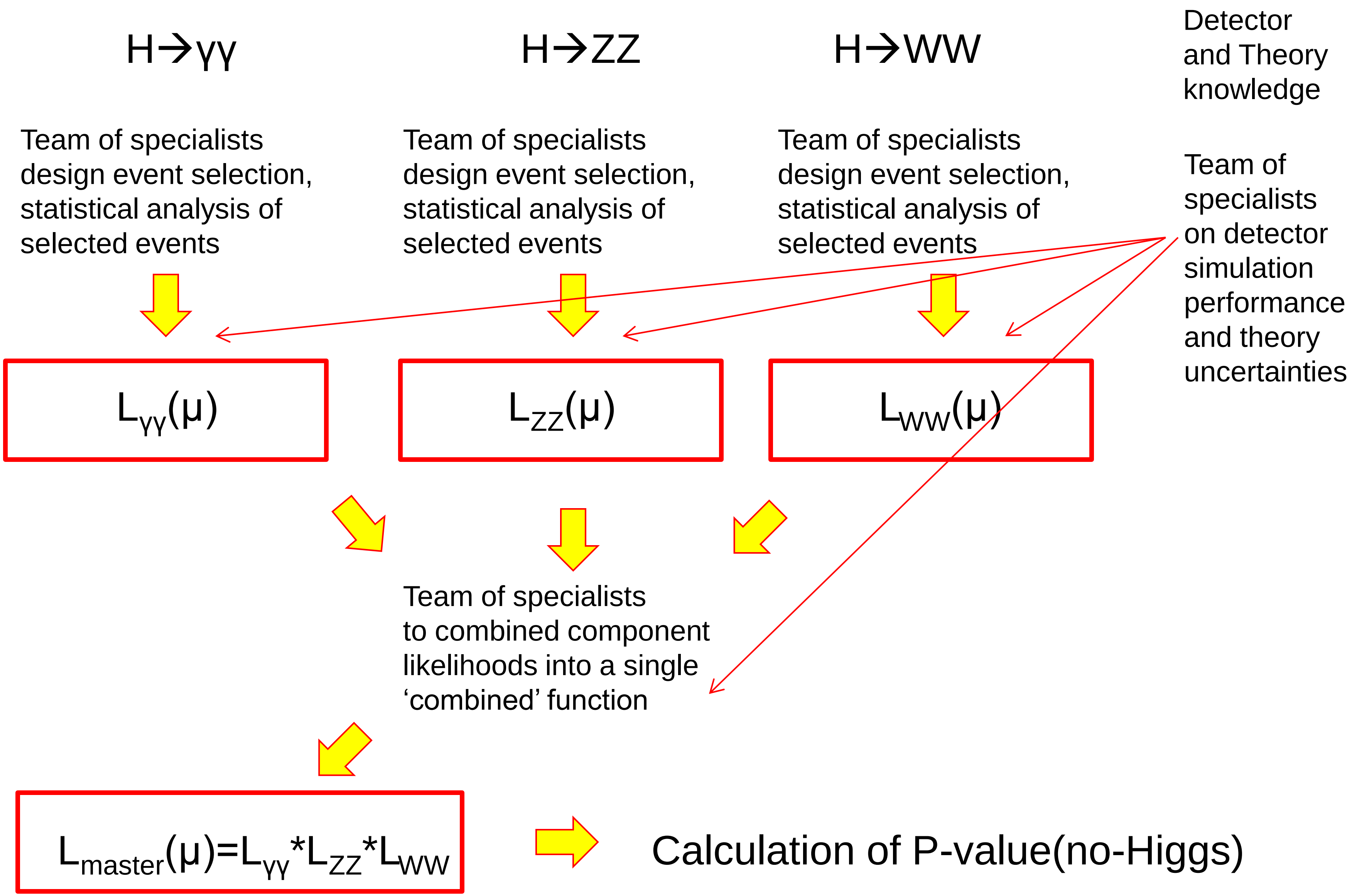
```
// Resurrect model and data
TFile f("model.root") ;
RooWorkspace* w = f.Get("w") ;
RooAbsPdf* model = w->pdf("sum") ;
RooAbsData* data = w->data("xxx") ;
```

```
// Use model and data
model->fitTo(*data) ;

RooPlot* frame =
         w->var("dt")->frame() ;
data->plotOn(frame) ;
model->plotOn(frame) ;
```





The Higgs discovery workflow



$H \rightarrow \gamma\gamma$

$H \rightarrow ZZ$

$H \rightarrow WW$

Detector and Theory knowledge

Team of specialists design event selection, statistical analysis of selected events

Team of specialists design event selection, statistical analysis of selected events

Team of specialists design event selection, statistical analysis of selected events

Team of specialists on detector simulation performance and theory uncertainties

$L_{\gamma\gamma}(\mu)$

$L_{ZZ}(\mu)$

$L_{WW}(\mu)$

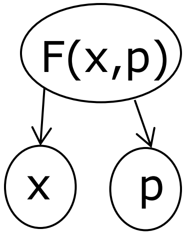
Team of specialists to combined component likelihoods into a single 'combined' function

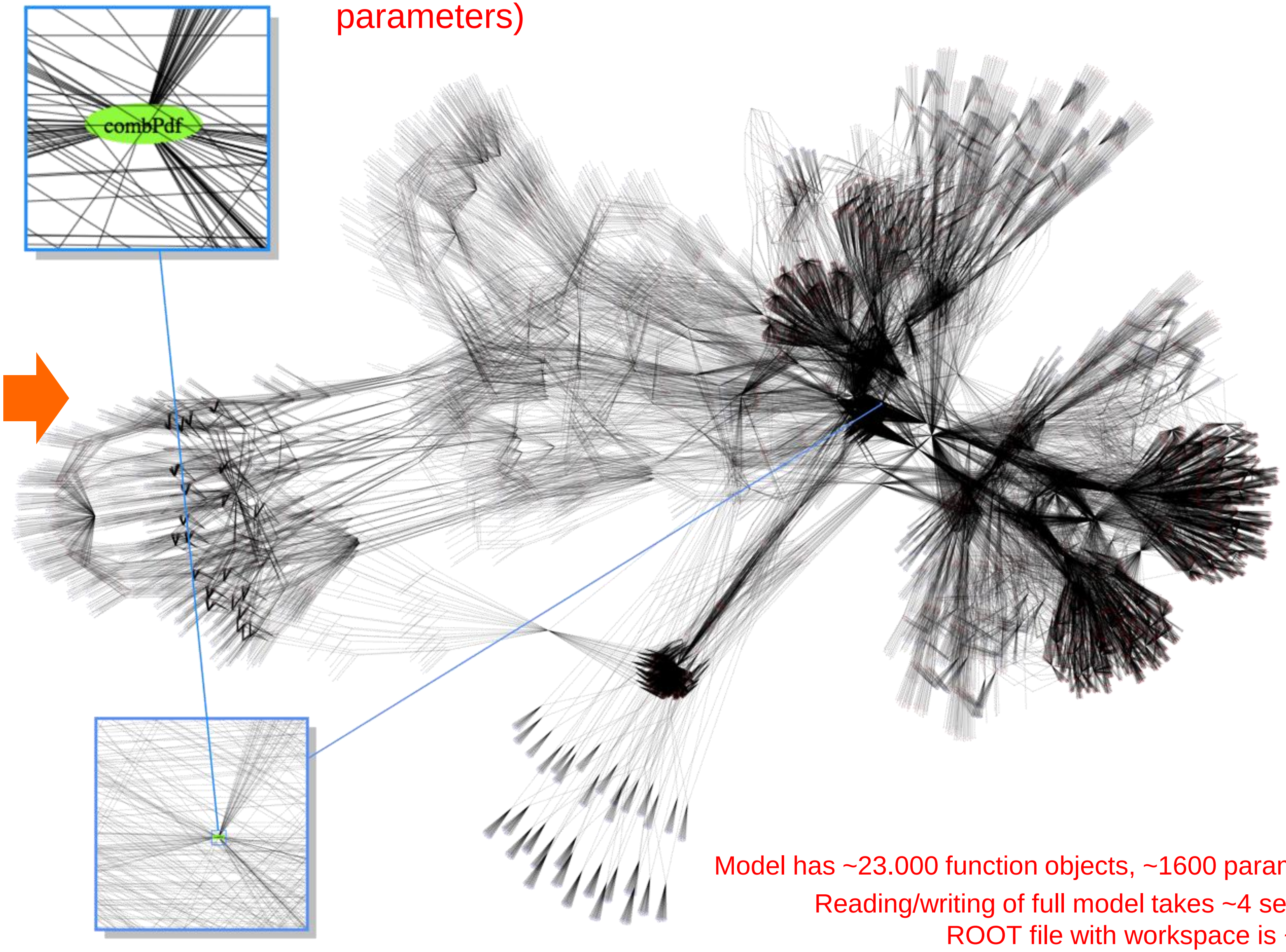
$L_{master}(\mu)=L_{\gamma\gamma}*L_{ZZ}*L_{WW}$

Calculation of P-value(no-Higgs)

The full ATLAS Higgs combination in a single workspace…

Atlas Higgs combination model (23.000 functions, 1600 parameters)

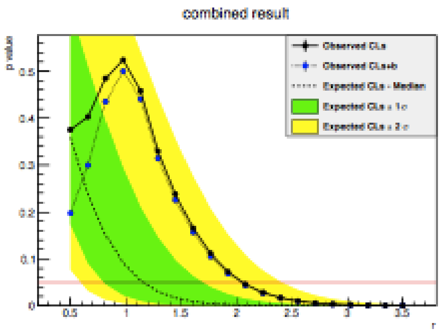




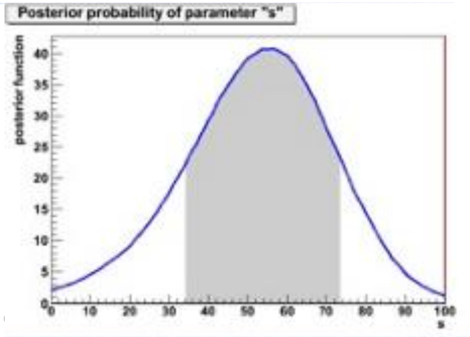
Model has ~23.000 function objects, ~1600 parameters
Reading/writing of full model takes ~4 seconds
ROOT file with workspace is ~6 Mb

RooStats – Statistical analysis of RooFit models

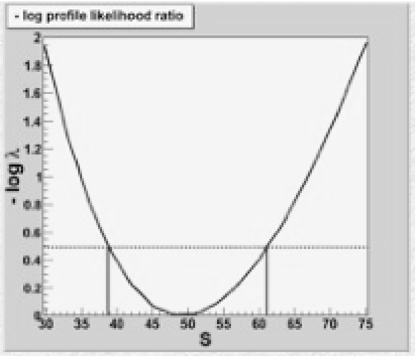
- With RooFits one has (almost) limitless possibility to construct probability density models
  - With the workspaces one also has the ability to deliver such models to statistical tools that are completely decoupled from the model construction code.
    Will now focus on the design of those statistical tools



- The RooStats projected was started in 2007 as a joint venture between ATLAS, CMS, the ROOT team and myself.
  Goal: to deliver a series of tools that can calculate intervals and perform hypothesis tests using a variety of statistical techniques



  - Frequentist methods (confidence intervals, hypothesis testin
  - Bayesian methods (credible intervals, odd-ratios)
  - Likelihood-based methods



Confidence intervals: $[\theta_-, \theta_+]$,or $\theta<X$ at 95% C.L.
Hypothesis testing: → $p(data|\theta=0) = 1.10^{-7}$

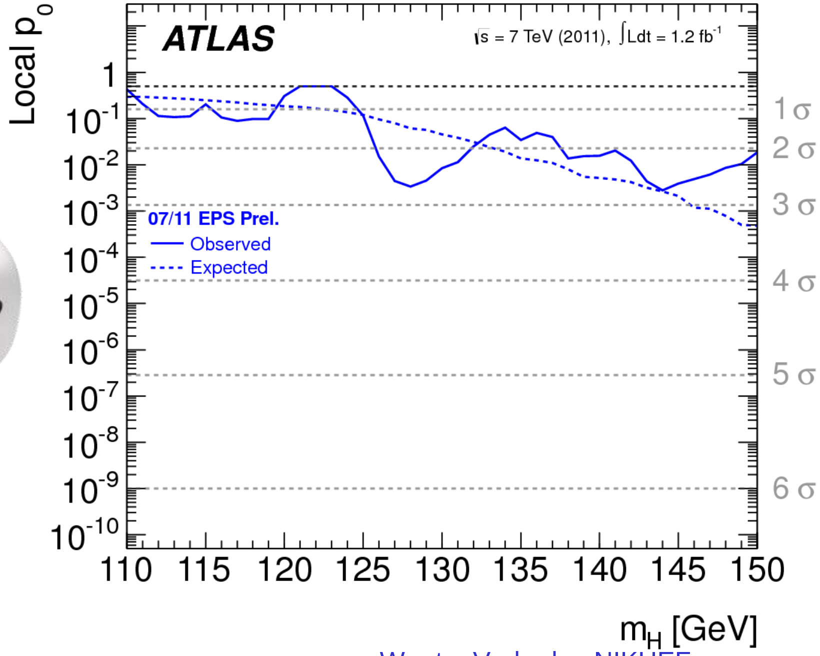
W

The result – evolution over time – July 2011

- Full analysis and combination chain in place since 2011.
- Since mass of Higgs boson was not a priori known and gives that properties of Higgs depends strongly on it, p-value (input to discovery declaration) calculated for a range of assumed Higgs masses (110-150 GeV)

‘p-value’

Juli 2011

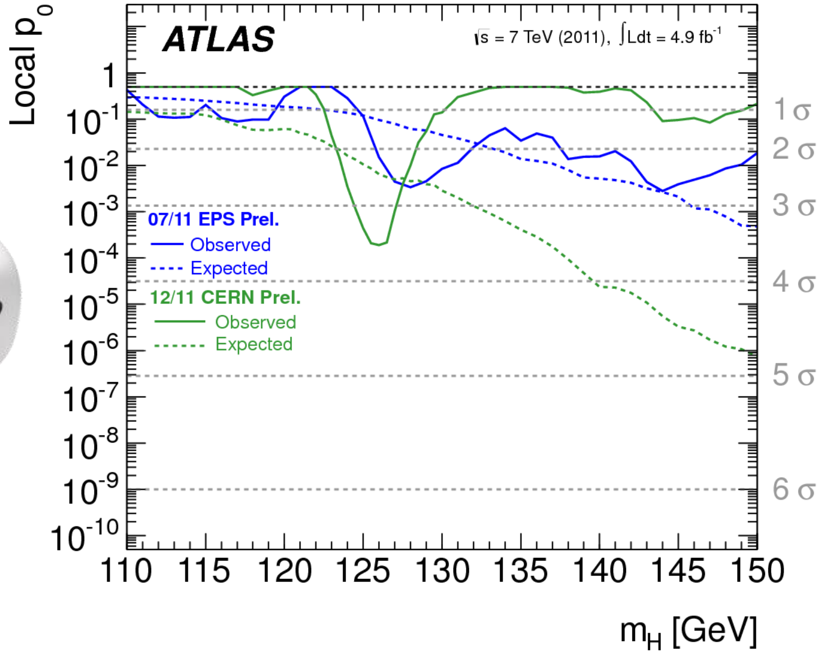


The result – evolution over time – December 2011

- Full analysis and combination chain in place since 2011.
- Since mass of Higgs boson was not a priori known
  and gives that properties of Higgs depends strongly on it,
  p-value (input to discovery declaration) calculated for a range
  of assumed Higgs masses (110-150 GeV)

'p-value'

December 2011

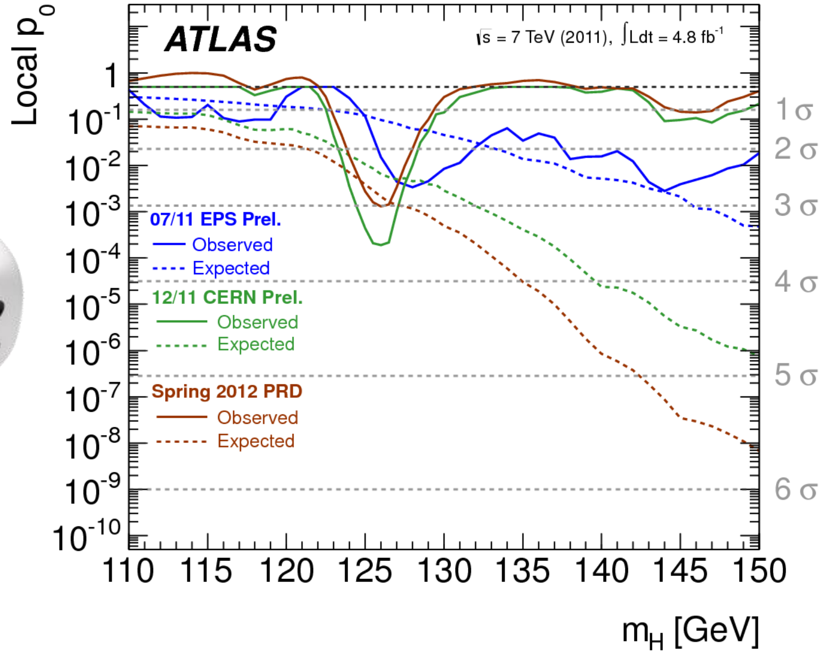


The result – evolution over time – April 2012

- Full analysis and combination chain in place since 2011.
- Since mass of Higgs boson was not a priori known and gives that properties of Higgs depends strongly on it, p-value (input to discovery declaration) calculated for a range of assumed Higgs masses (110-150 GeV)

‘p-value’

April 2012

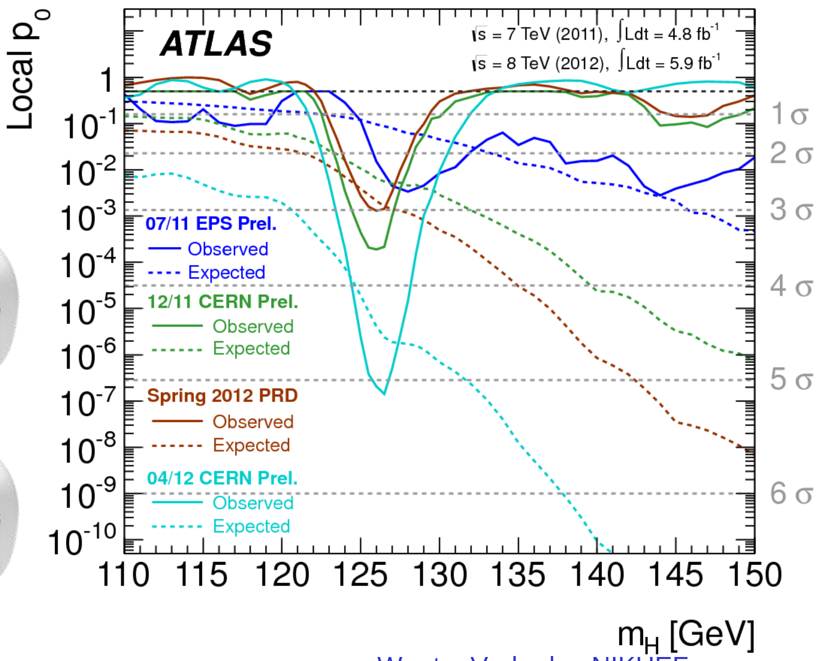


The result – evolution over time – June 2012

- Full analysis and combination chain in place since 2011.
- Since mass of Higgs boson was not a priori known and gives that properties of Higgs depends strongly on it, p-value (input to discovery declaration) calculated for a range of assumed Higgs masses (110-150 GeV)

'p-value'

Juni 2012

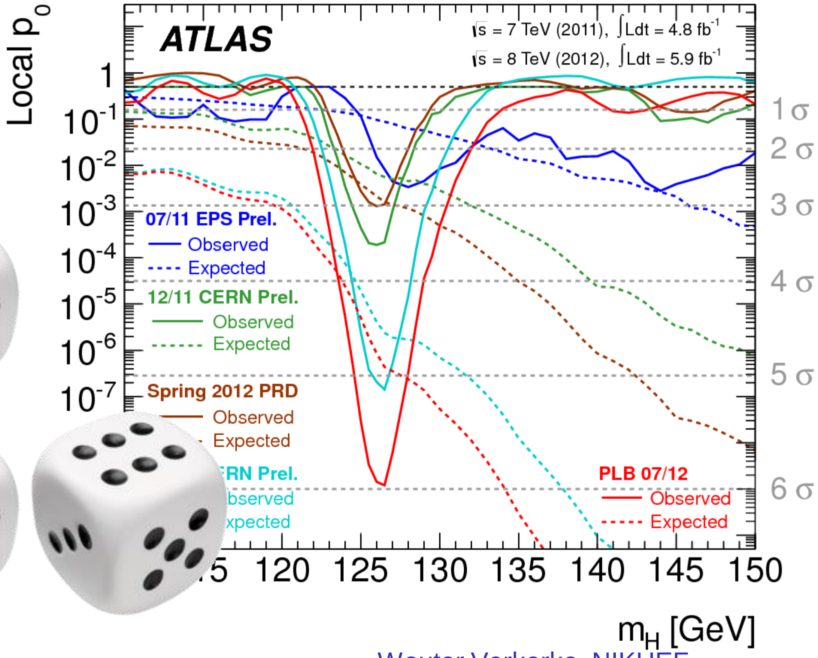


The result – evolution over time – July 2012

- Full analysis and combination chain in place since 2011.
- Since mass of Higgs boson was not a priori known and gives that properties of Higgs depends strongly on it, p-value (input to discovery declaration) calculated for a range of assumed Higgs masses (110-150 GeV)

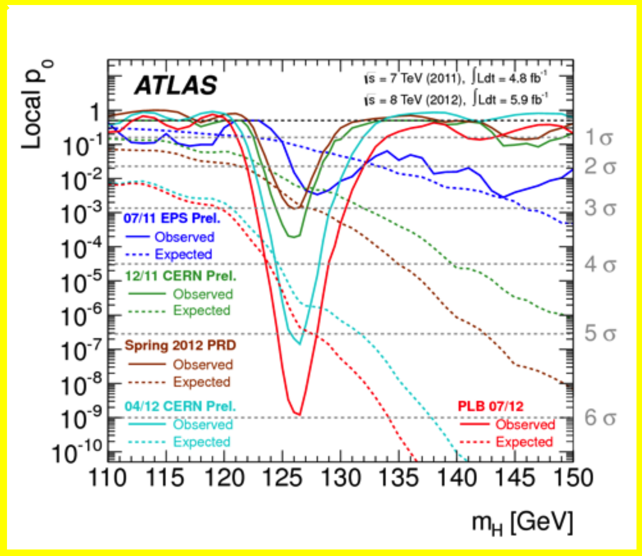
'p-value'

Juli 2012



July 4 – Declaration of discovery

After July 4 – A small party



Summary

- The discovery of the Higgs has been one of the most complex data analysis challenges performed in particle physics
  - No single Higgs decay signature was sufficiently powerful to result in a discovery
  - Due to the unknown Higgs mass it wasn’t even known in advance where to look best
- Enormous effort to isolate LHC collision events with Higgs-like signature in many decay channels in parallel
  - Event selection process often helped with machine-learned criteria (e.g. boosted decision trees) [Tools: TMVA, Neurobayes]
  - Likelihood models built describing selected that maximize statistical power by taking into account properties of selected events, _and_ take into account known uncertainties on hundreds of aspects of detector and theory simulation [Tools: RooFit, RooStats, Histfactory]
- Joint likelihood model across all channels combines information into single most powerful test
  - Convincing evidence obtained first on July 2012 dataset, when it was calculated that the odds of the observed signal arising as a statistical fluctuation (‘no Higgs hypothesis’) was less than 1 in ~3.500.000 (‘5 sigma’)