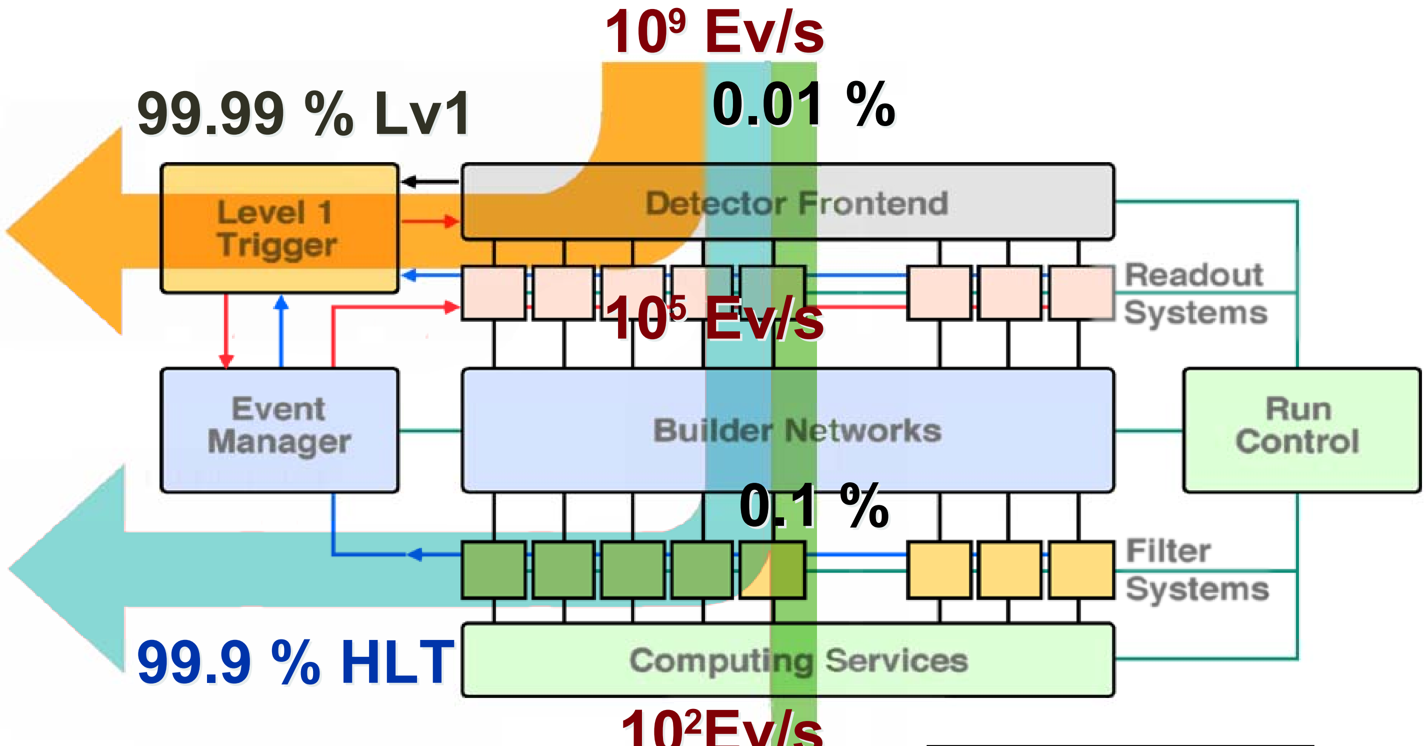
Metody eksperymentalne w fizyce wysokich energii

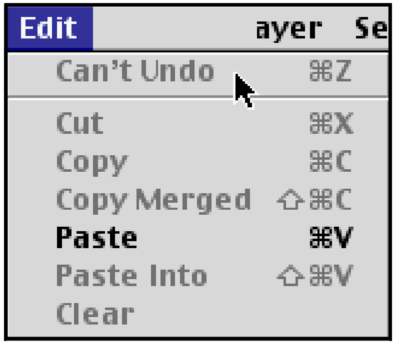
prof. dr hab. A.F.Żarnecki
Zakład Cząstek i Oddziaływań Fundamentalnych IFD

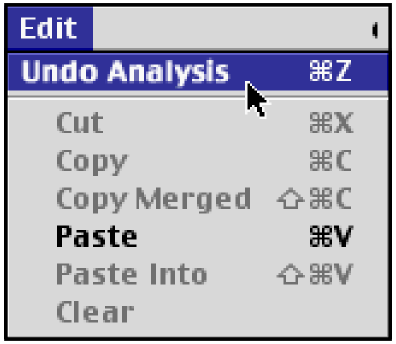
Wykład XII

- Rekonstrukcja przypadków

A parting thought







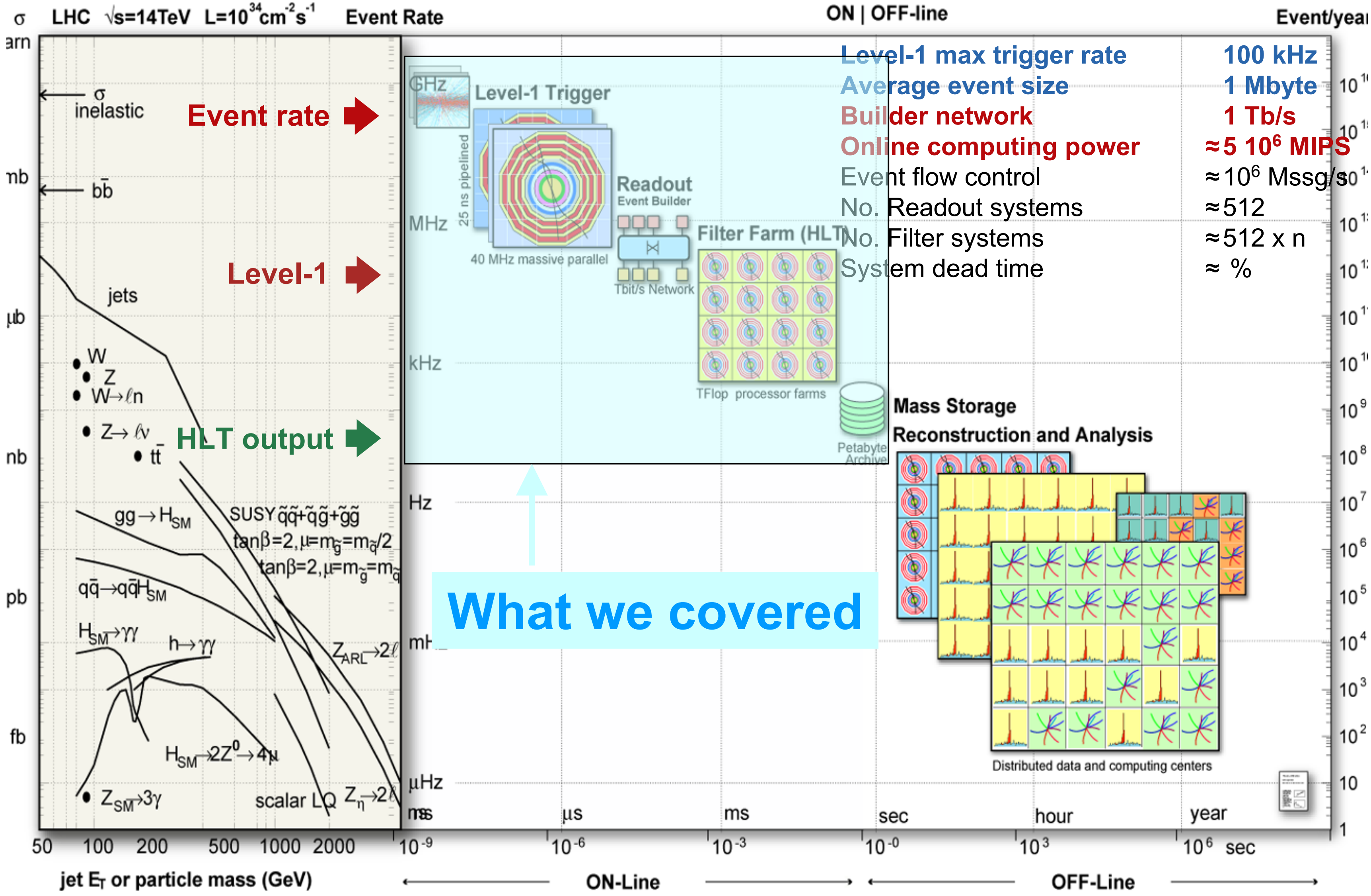
With respect to offline analysis:

Same hardware (Filter Subfarms)

Same software

But different situations

Online Physics Selection: summary



Level-1 max trigger rate	100 kHz
Average event size	1 Mbyte
Builder network	1 Tb/s
Online computing power	≈5 10^6 MIPS
Event flow control	≈10^6 Mssg/s
No. Readout systems	≈512
No. Filter systems	≈512 x n
System dead time	≈ %

What we covered

Dane odczytywane z detektora:

- "zapalone" cele kalorymetru
  - ID celi + sygnał
  - + ew. informacja o profilu podłużnym lub poprzecznym w przypadku wielokrotnych odczytów

- "hity" w detektorach śladowych
  - ID/pozycja drutu/paska/piksla
  - + ew. sygnał (~dE/dx)
  - + ew. czas (komory dryfowe, TPC)

Co potrzebujemy do analizy fizycznej:

- zidentyfikowane fotony: energia, kierunek emisji (=> wektor pędu)
- zidentyfikowane leptony (elektrony/miony/taony): energia, pęd, znak
- zrekonstuowane jety: energia, pęd, masa niezm., zapach (!)
- zrekonstuowany brakujący pęd poprzeczny, energia, masa...

Kolejne kroki rekonstrukcji

Krok I: "normalizacja" na poziomie poszczególnych detektorów

kalorymetry

ID celi => pozycja w detektorze, z uwzględnieniem pozycjonowania

sygnał => depozyt energii w GeV, z uwzględnieniem kalibracji, tła

detektory śladowe

ID elementu, ew. czas dryfu

=> pozycja w detektorze, z uwzgl. pozycjonowania

Krok II: rekonstukcja podstawowych "elementów"

wciąż na poziomie poszczególnych detektorów

kalorymetry

"klastry"/"wyspy" energii (pojedyncze cząstki ?)

detektory śladowe

"klastry"/punkty, ew. elementy/fragmenty toru (punkt + kierunek)

Krok III: rekonstrukcja obiektów na na poziomie globalnym

- rekonstrukcja torów
- rekonstrukcja wierzchołka pierwotnego i w. wtórnych
- rekonstrukcja jetów

Krok IV: identyfikacja obiektów (na poziomie globalnym)

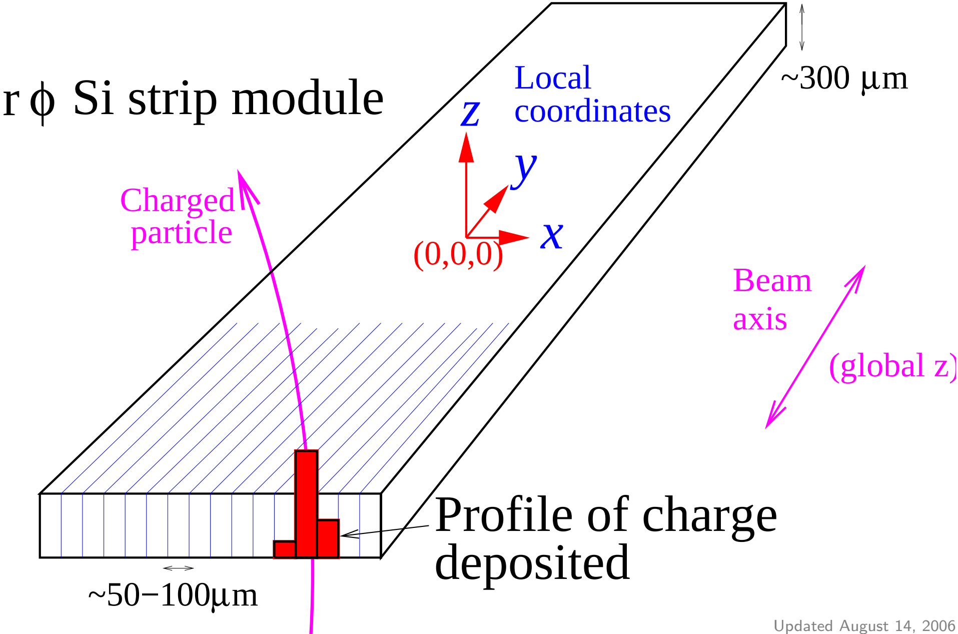
- łączenie tor-klaster (i ew. korekta energii: PFA)
- identyfikacja leptonów
- znaczenie zapachów jetów (ciężkie kwarki)

Tracking: basic idea

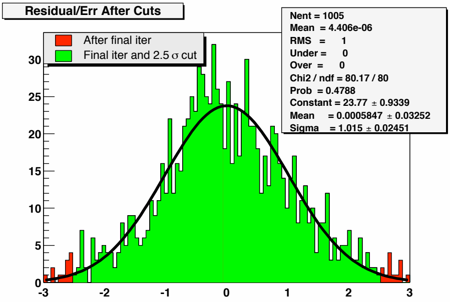
When we talk about “tracking,” we want to do the following:

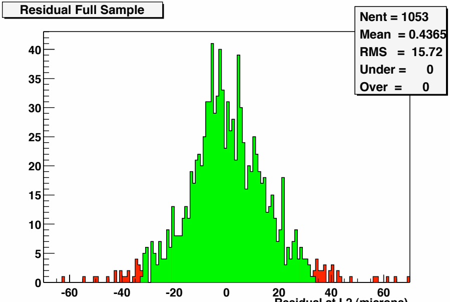
- Measure the true path of the charged particle, which let’s us know...
- The momentum (3-momentum) if we know the magnetic field
- The sign of the charge of the particle
- With other constraints or assumptions, the “origin” in space of the particle
- Without some other detector though, we can’t measure the mass independently just with a tracker

2D Point ($r\phi$ or rz): $\bar{x} = (\sum i * q_i)/\sum q_i$



Typical Resolution of 50 μm Strips (CDF)





Cluster Width	Resolution
1	12 μm
2	9 μm
3	14 μm
4+	22 μm

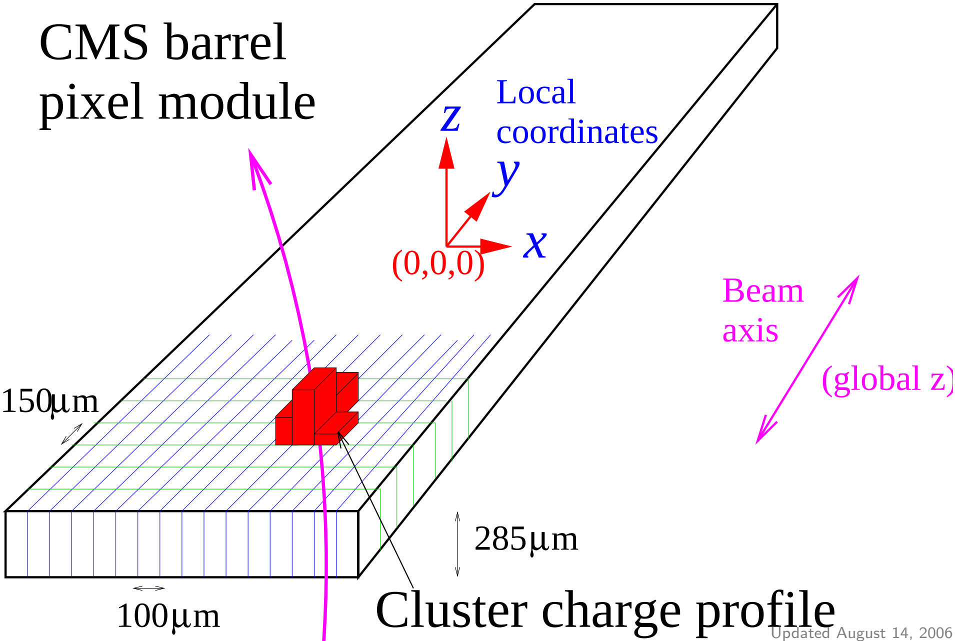
- *rz* strips are either shallow stereo (2°), or 90° stereo but larger pitch
- Intrinsic resolution of larger pitch stereo usually factor of 2+ worse than $r\phi$

Resolution of Si Strip Detector

- The resolution of a Si hit depends on the number of strips in the cluster (2-strip most precise)
- Charge drifts with Lorentz force: $q(\vec{\boldsymbol{E}} + \vec{\boldsymbol{v}} \times \vec{\boldsymbol{B}})$
- Thin material: Landau distribution of charge [2]
- Track impact angle & position makes a difference [3]
- Delta rays (hard knock e^-) can bias charge distribution
- Dead channels, noise, V_{dep}, temperature all could affect this too
- Radiation damage changes
- Pileup from previous event
- Multiple particles passing through same strips

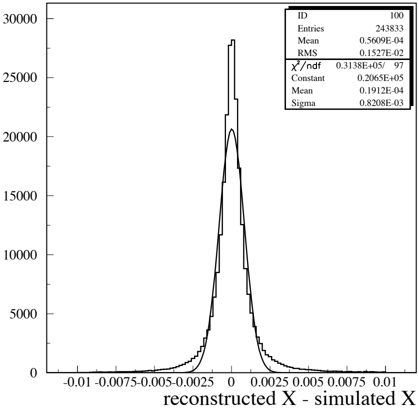
3D Space Point from Pixel Detector

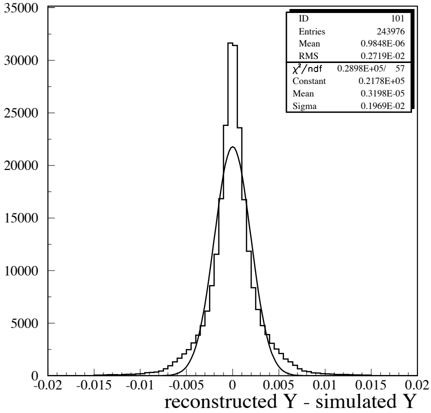
CMS barrel pixel module



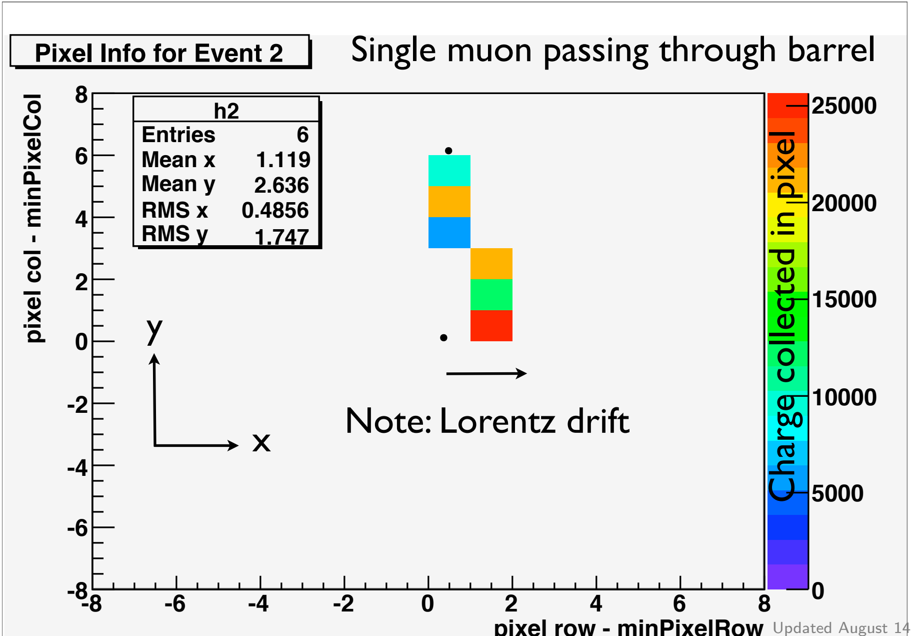
Cluster charge profile

Typical Pixel Resolution (CMS): 8 – 20 $\mu\mathrm{m}$





Pixel Stub! Impact angle (θ) from shape



Pattern Recognition & Track Fitting

Typically, pattern recognition algorithms are either "inside-out" or "outside-in."

- You have to start with some idea of the path of the particle to bootstrap your algorithm: a track seed
- Then you take this candidate, this seed, can try to find compatible hits in other layers
- Continue this process until you've met some criteria for what a "good track" should have
- Once you've got your hits for your track, try to do a good job of fitting your pseudohelix
- Psuedohelix because there is energy loss and multiple scattering
- There are also spurious hits from detector noise and low momentum, unreconstructable tracks. These will mess up your "true" helix

Alignment is Crucial!

- Before you have a hope of matching hits from one layer to the next to make a helix, your detector elements' positions must be known to the level of your intrinsic resolution
- But how can you align your detector without tracks?
- This is the subject of part of Nick Hadley's talks on calibrations
- But it is always an iterative procedure, bootstrapped by a very good optical survey during construction and installation
- Here's an example (from ATLAS) of what needs to be done...

Alignment Strategy

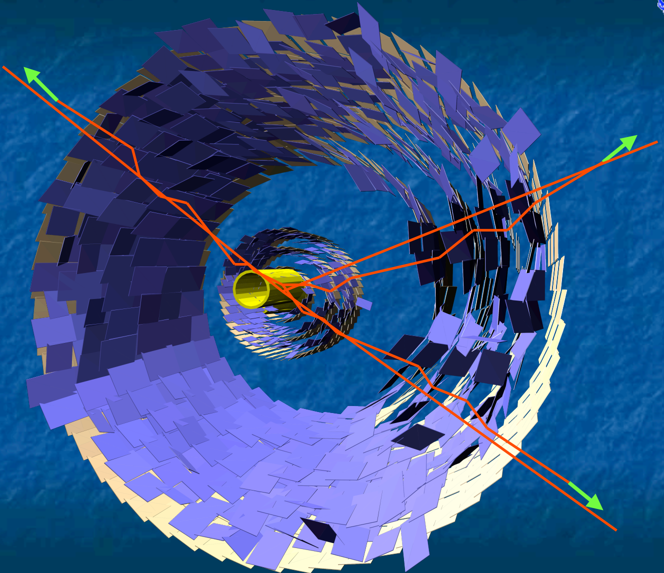
- **Applies to tracking detectors including muon chambers.**
  - Then use tracks to align calorimeters as trackers measure position better (usually) than calorimeters

- **Typically 3 step process**
  1. Measure element (e.g. wire, pixel) position during construction of subdetector using coordinate measuring machines and similar devices.
  2. Measure relative position of subdetectors after assembly using surveying techniques such as lasers.
     - Only works for detectors you can see.
  3. Track based alignment

Alignment Concept & Typical Numbers

	Muon	**Tracker**	
		Strip	Pixel
Assembly:	O(mm)	0.1-0.5mm	50-100 μm
+			
Hardware Alignment:	~<100μm	<100μm	50-100 μm (no HA foreseen)
+			
Track Based Alignment	~100μm (perhaps below)	~10μm	~5 μm
Remarks:	Hardware Alignment will provide the operational alignment level. Track based alignment will be a cross check and eventually a completion	Hardware Alignment will insure pattern recognition. Track Based Alignment must provide the final alignment	Only Track based Alignment. Nothing else!

Global $\chi 2$ fit – the idea

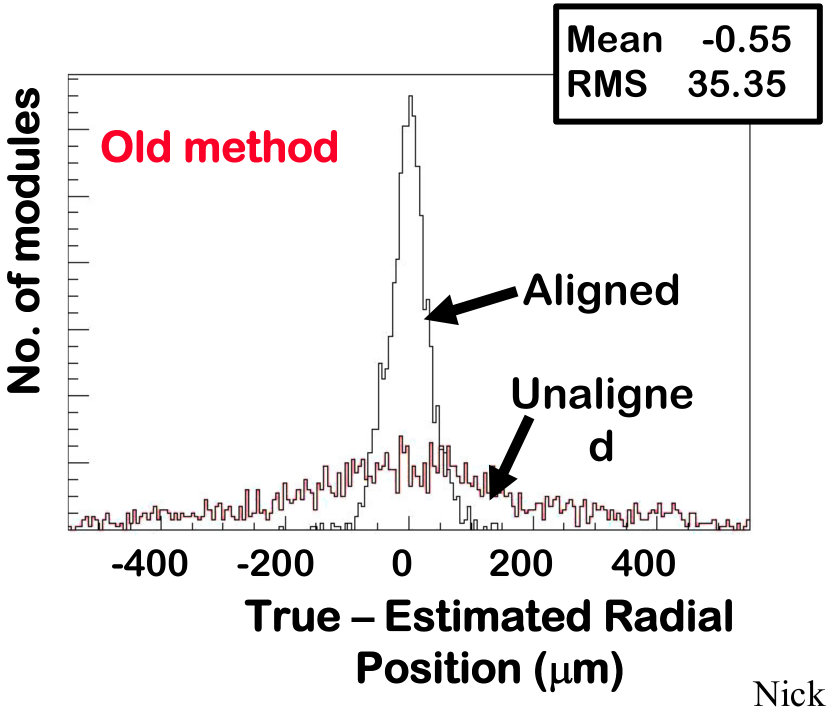


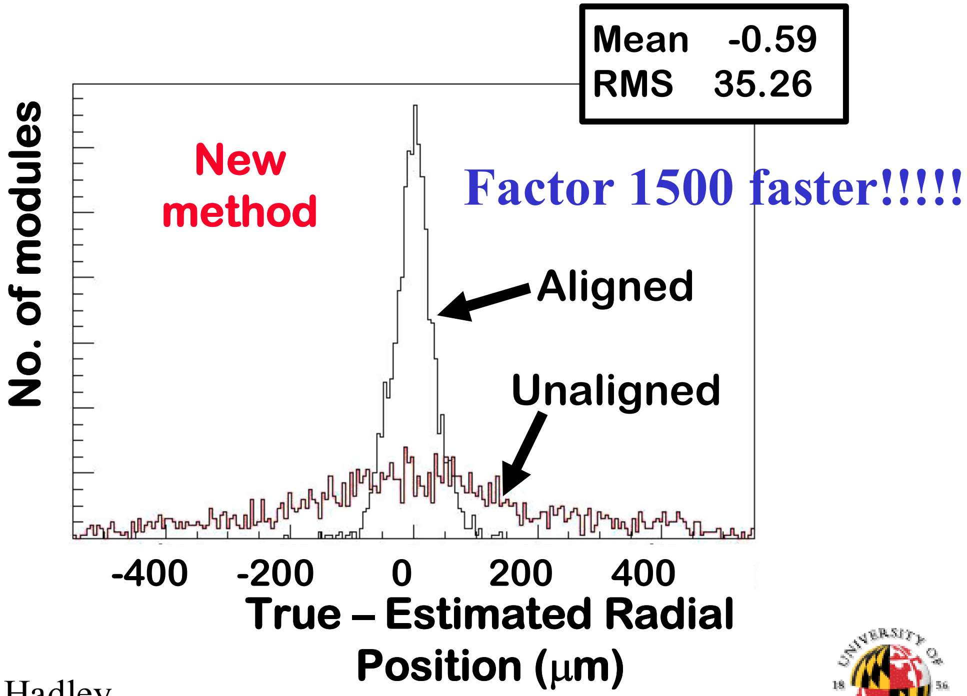
CMS implementation of Millepede II Algorithm (Millepede see www.desy.de/~blobel)

Original Millepede method solves matrix eqn. A x = B, by inverting huge matrix A.
This can only be done for < 12000 alignment parameters.

New Millepede method instead minimises |A x – B|.
Is expected to work for our 100000 alignment parameters.

Both successfully aligned ~12% of Tracker Modules using 2 million $Z \rightarrow \mu+ \mu-$ events.
Results identical, but new method 1500 times faster !

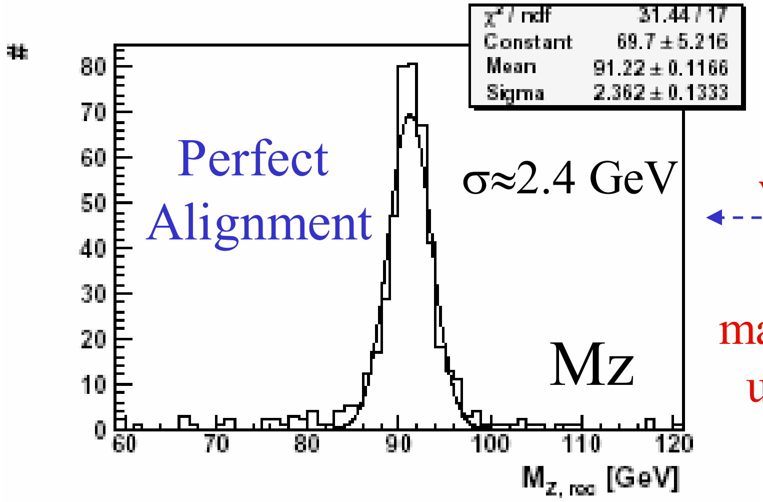




Factor 1500 faster!!!!!

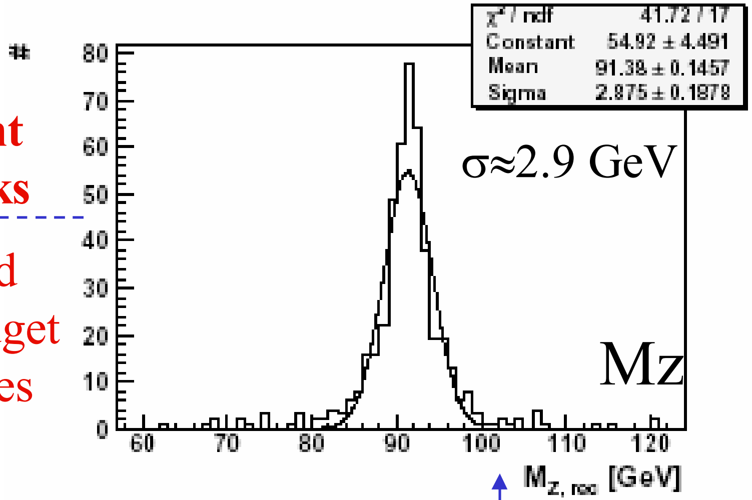
Mis-Alignment: Impact on Physics (important for Z′, LED)

⇒ **Use $Z \to \mu\mu$ to illustrate the impact of mis-alignment on physics**



Alignment with tracks

B field and material budget uncertainties



First Data Taking

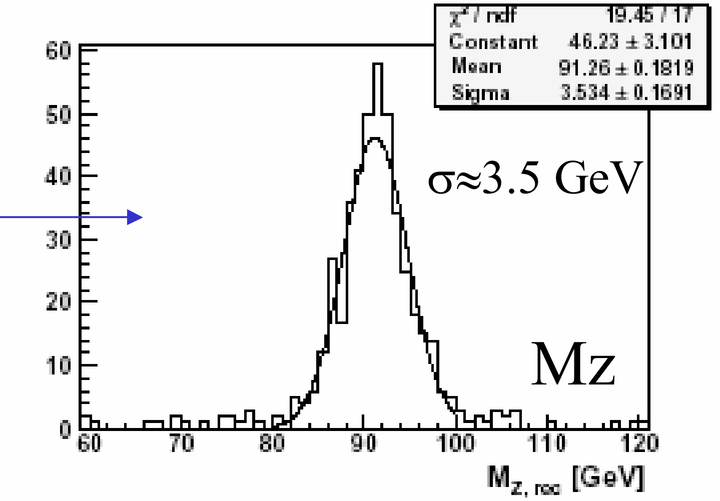
<1fb^{-1}

Laser Alignment

⊗

Mechanical Constraints

⇒≈100μm alignment uncertainties



Long(er) Term:

≈>1fb^{-1}

First results of Alignment with tracks

⇒≈20μm alignment uncertainties

Helix Parameters

We can decompose the momentum of a track in spherical coordinates [1]:

$$\begin{aligned} p_x &= p \cos\phi \sin\theta \\ p_y &= p \sin\phi \sin\theta \\ p_z &= p \cos\theta \end{aligned}$$

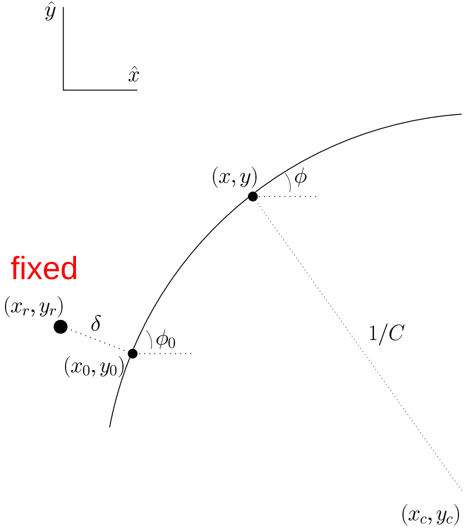
Different experiments choose different ranges for the angles, it's important that you figure out what they are using:

$$\phi \in [-\pi, \pi] \qquad \theta \in [0, \pi]$$

There must also be some "reference point" in space to uniquely define our helix: (x_r, y_r, z_r)

Helix Parameters in $x - y$ Plane

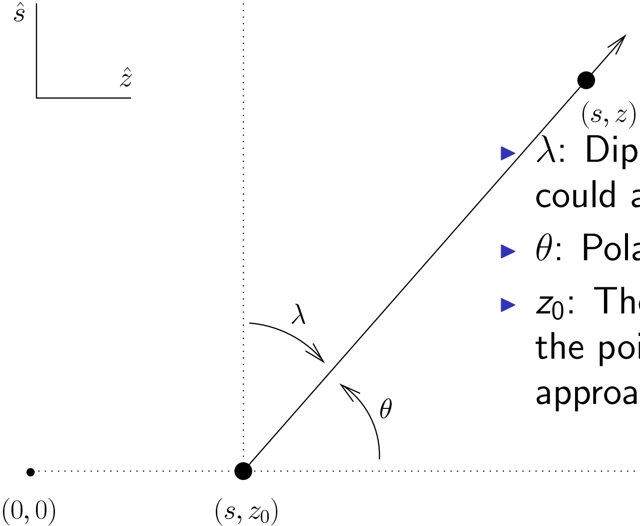
This parameterization is more closely related to things we actually measure with our trackers



- C: Curvature of the track. Signed with charge.
- ϕ_0: Azimuthal angle of the momentum at the point of closest approach
- δ: Distance of closest approach. (Also signed, but differently.)

3 parameters in x-y

$z = z_0 + s \tan \lambda$



- λ: Dip angle of track, or could also use
- θ: Polar angle of track
- z_0: The z of the track at the point of closest approach in $x - y$

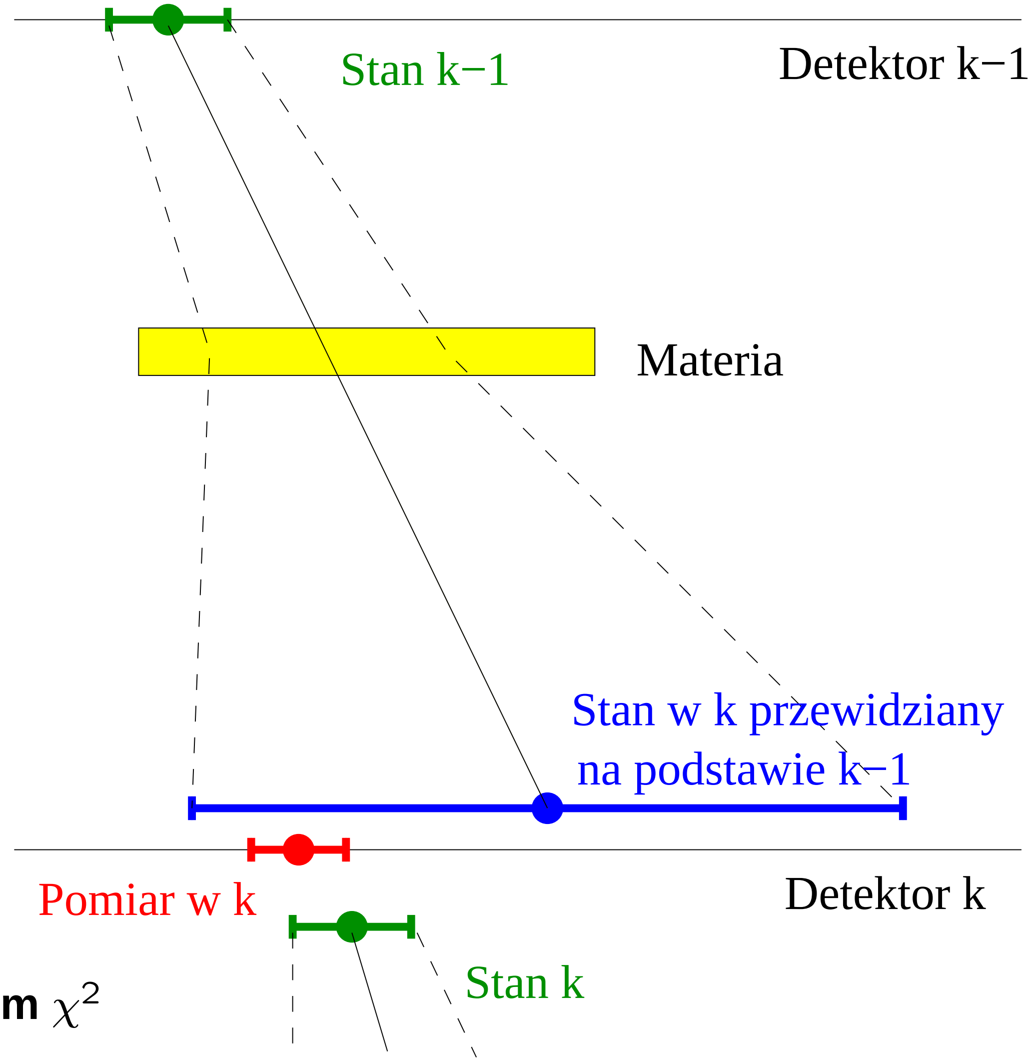
+ 2 parameters in r-z

Kalman Filter

- The Kalman Filter [4] is an iterative procedure
- You start with a seed track. For example, a pair of hits in the pixel detector that line up within 5σ of the primary interaction region
- Then you add points on successive layers, taking into account projected error from current hypothesis track and multiple scattering
- Finally, if this track passes some requirements for a minimum number of hits, you refit it (smooth) again with the filter using a better starting point

Filtr Kalmana - sposób na dopasowywanie torów

- **Algorytm oblicza parametry toru używając punktów "pasujących" do toru:**
- **Punkty są dodawane zaczynając od najbardziej zewnętrznych**
- **Stan początkowy jest brany z zewnętrznych detektorów śladowych**
- **Stan obliczony dla detektora k-1 jest ekstrapolowany do detektora k z uwzględnieniem wpływu materii.**
- **Punkty w znalezione w detektorze k są porównywane z przewidywaniem i są dodawane jeżeli nie wprowadzaja dużego wzrostu χ^2 przy dopasowaniu toru - obliczany jest stan k**
- **Jezeli więcej niż jeden punkt "pasuje" brane są pod uwagę wszystkie możliwości i ostatecznie wybierany jest tor z największą liczbą punktów i najmniejszym χ^2**



Tracking Resolution

The resolution of the various helix parameters depends on a number of things

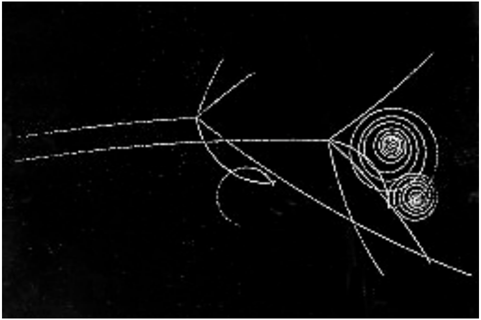
- The number of hits used from various subdetectors (hopefully, more is better)
- The momentum of the particle. Higher momentum particles deflect less from multiple scattering
- The polar angle of the track (η)
- Quality of alignment
- Presence of other tracks!
- Detector noise

High Multiplicity Strategies

- If your inner layers are being swamped, use an outside-in algorithm instead of an inside-out algorithm
- Try to boost efficiency for higher p_T tracks by reconstructing them first, and then remove these hits from consideration
- Upgrade your detector

Vertexing: Basic Idea

The basic idea of vertexing is to figure out where the particles came from. We can associate tracks to particle decays and interactions this way.



Vertexing Strategy

- A vertex is a point where more than one particle comes from
- If there is more than one track coming from the same place, then the helixes should cross each other, right?
- Look for places where helixes cross
- Caveat: The track parameters $\vec{\boldsymbol{p}}_i = (C, \phi, d_0, \tan\lambda, z_0)$ are different in different parts of the helix

Vertexing Algorithm

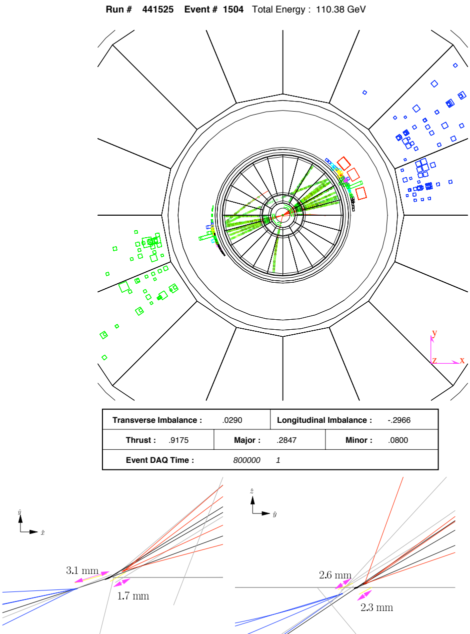
- If helixes cross, that means the track parameters must be the same at some point
- In 2D, this is as simple as looking for crossing circles
- But you can take full advantage of full track parameterization and covariance matrix to look for vertexes
- Most methods are built on some kind of χ^2 of track parameters

$$\chi^2 = \sum_{i}^{N_{\mathrm{trk}}} (\vec{\boldsymbol{x}} - \vec{\boldsymbol{g}}(\vec{\boldsymbol{p}}_i))^{\mathrm{T}} \mathbf{J}^{\mathrm{T}} \mathbf{M}_i^{-1} \mathbf{J} (\vec{\boldsymbol{x}} - \vec{\boldsymbol{g}}(\vec{\boldsymbol{p}}_i))$$

Vertex Algorithm Continued

- Again, we have a problem of pattern recognition
- The χ^2 can be defined for any collection of tracks
- Vertex algorithms are usually iterative however, pruning or down-weighting tracks which make large contributions to the χ^2 (outliers)
- Clumps, or clusters of tracks are what is searched for in this way

B-Tagging: Basic Idea

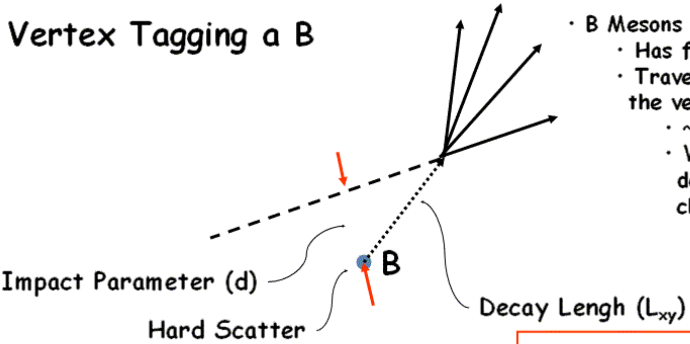


Transverse Imbalance :	.0290	**Longitudinal Imbalance :**	-.2966		
Thrust :	.9175	**Major :**	.2847	**Minor :**	.0800
Event DAQ Time :	*800000*	*1*			

- B hadrons have lifetimes and decay lengths distinct from other species
- Decay length is measurable in a given event by finding a vertex ("secondary") and taking the distance to the "primary" vertex
- Can look for B hadrons by finding these vertexes which are compatible with known properties of B hadrons

Example B-Tag (D0)

Vertex Tagging a B



- B Mesons
  - Has finite life time
  - Travels some distance from the vertex before decaying
    - ~ 1mm
    - With charm cascade decay, about 4.2 charged tracks

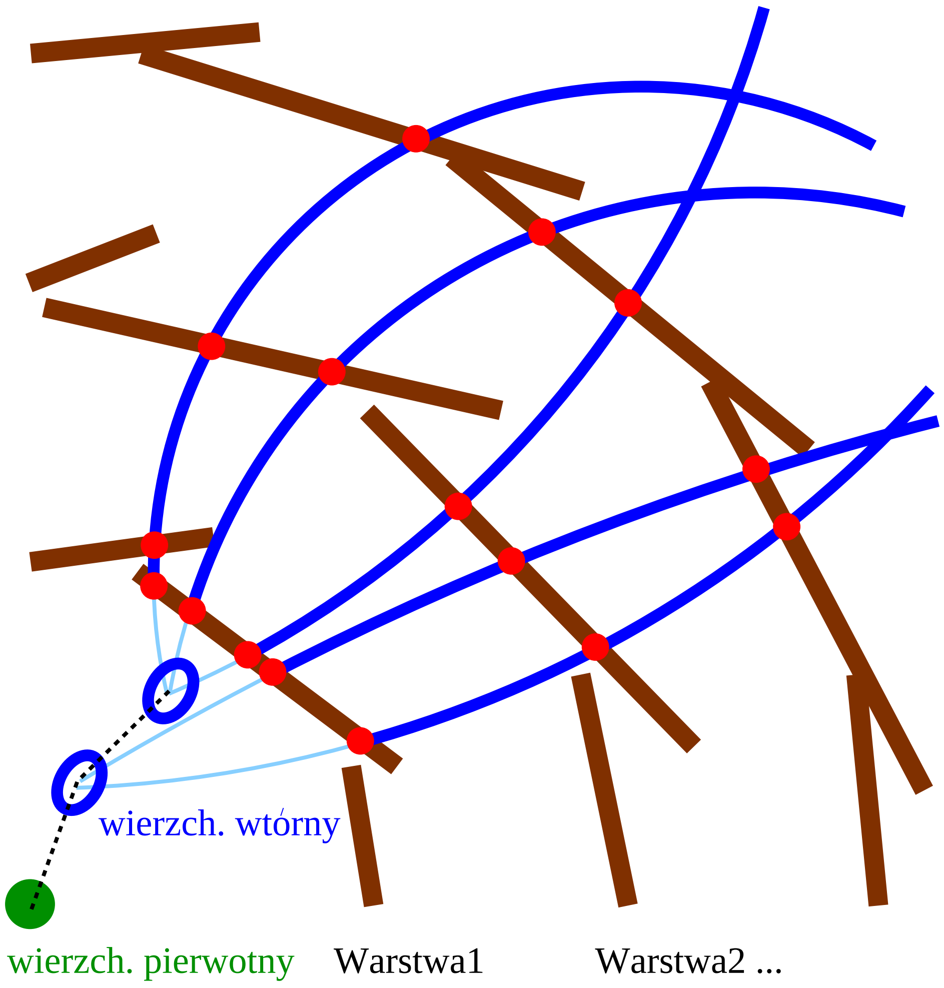
Several algorithms under active development

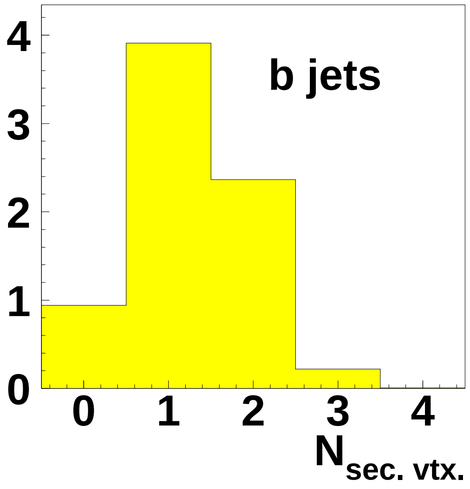
→ Impact Parameter Resolution $d/\sigma(d)$

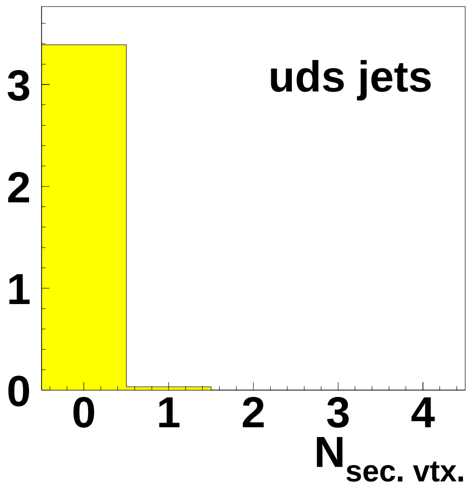
→ Decay Length Resolution $L_{xy}/\sigma(L_{xy})$

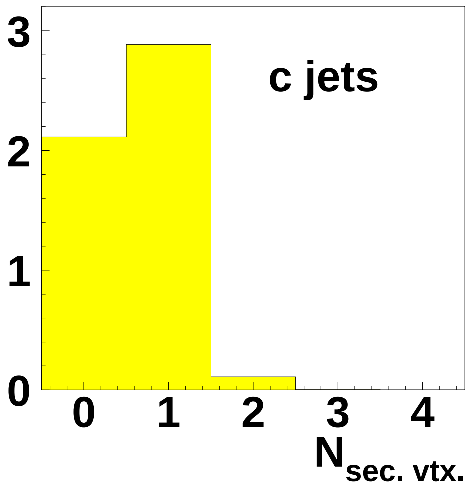
“Jet tagging” - 1

- **Dla każdego dżetu wierzchołki wtórne są wyznaczane za pomocą ZVTOP**

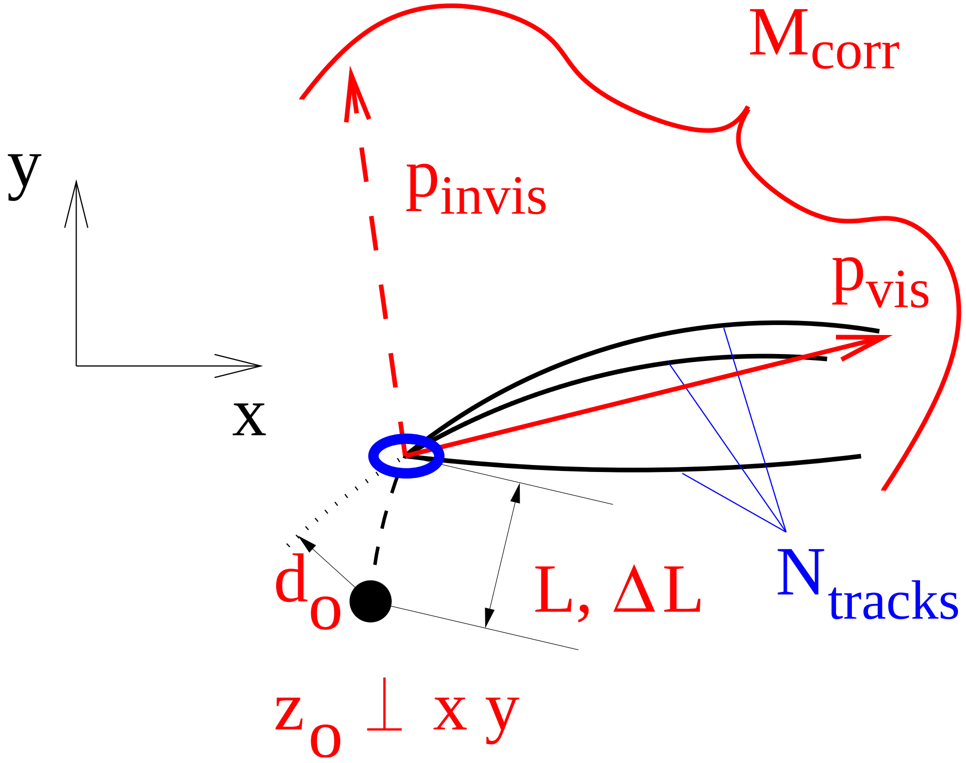


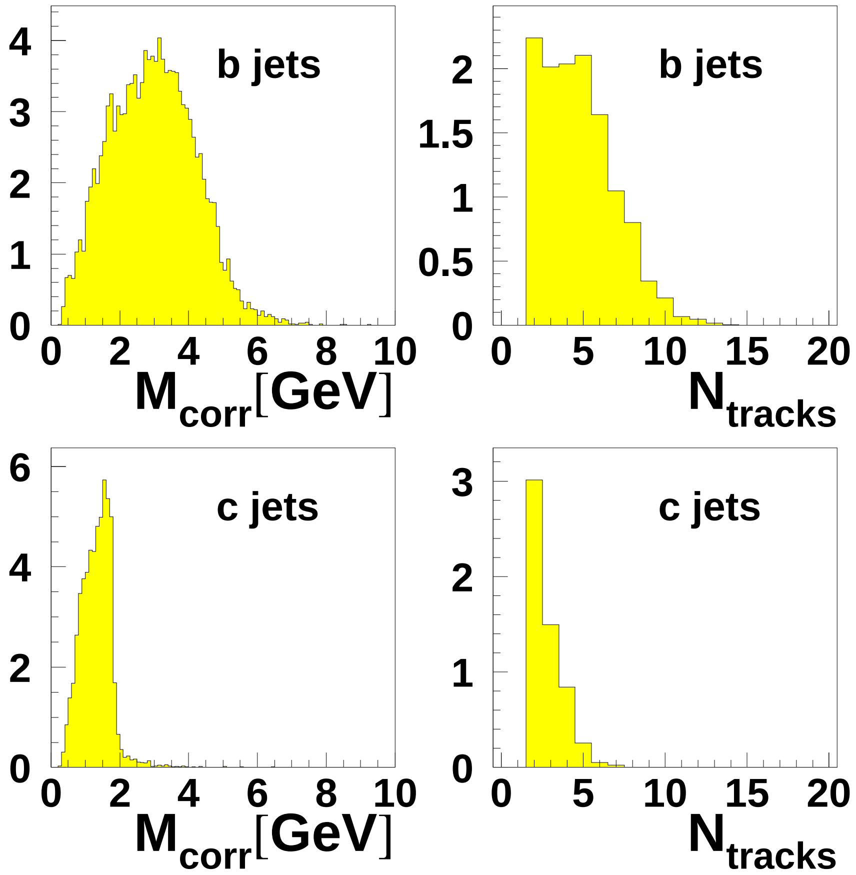






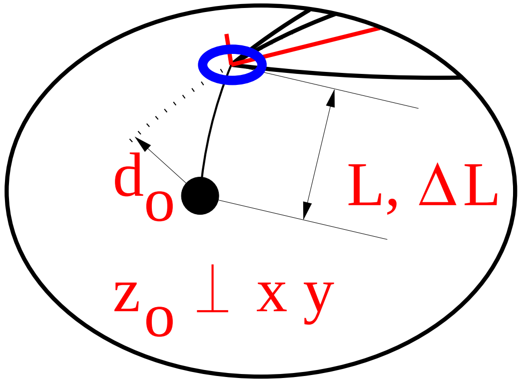
“Jet tagging” - 2

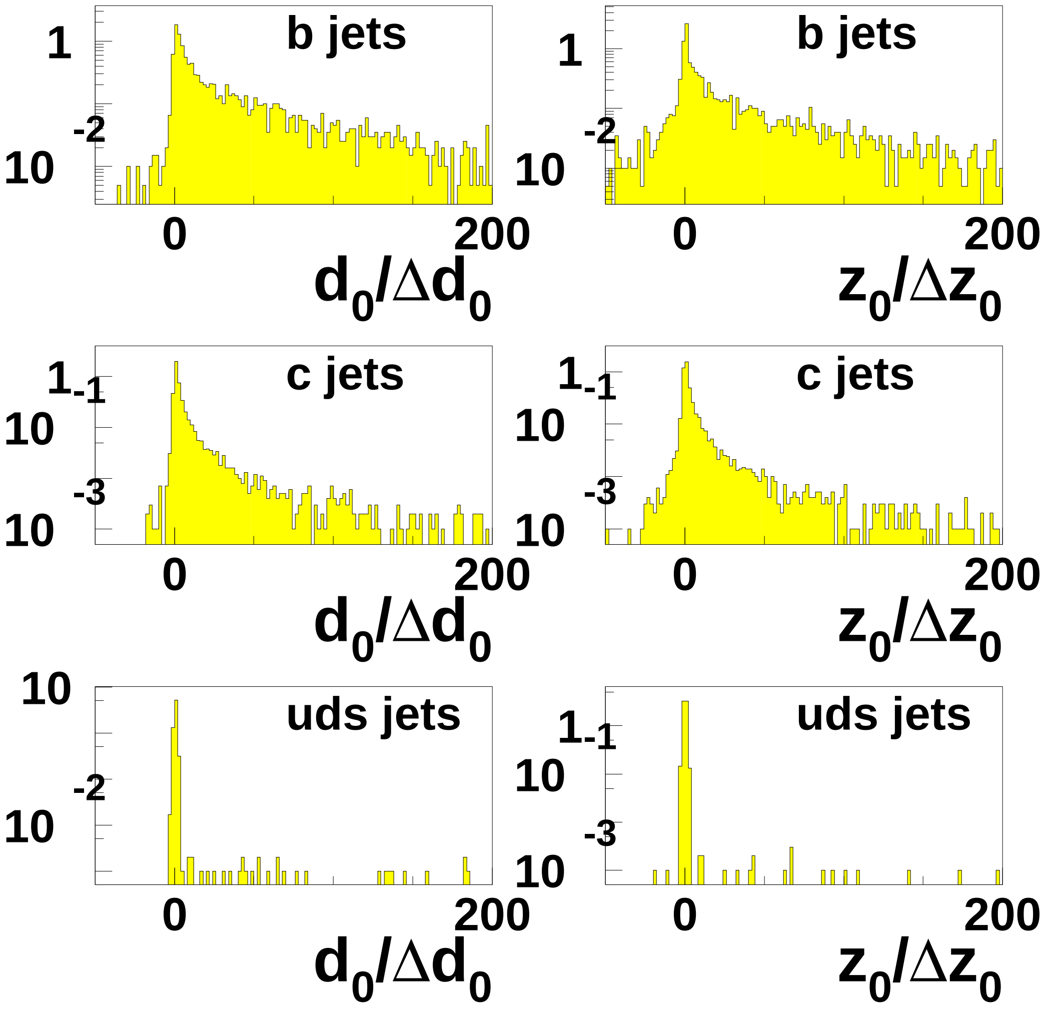


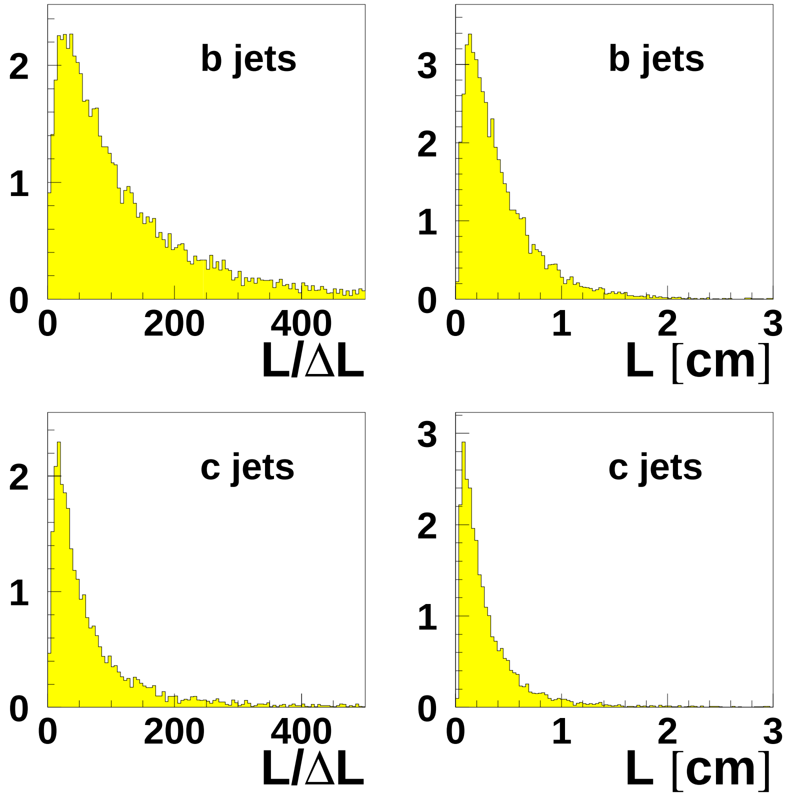


- Dla każdego dżetu sieć neuronowa (na podstawie powyższych wielkości) zwraca **b-tag** i **c-tag** $\in (0,1)$ - wielkości mówiące na ile jet jest podobny do dżetu **“b”** lub **“c”**

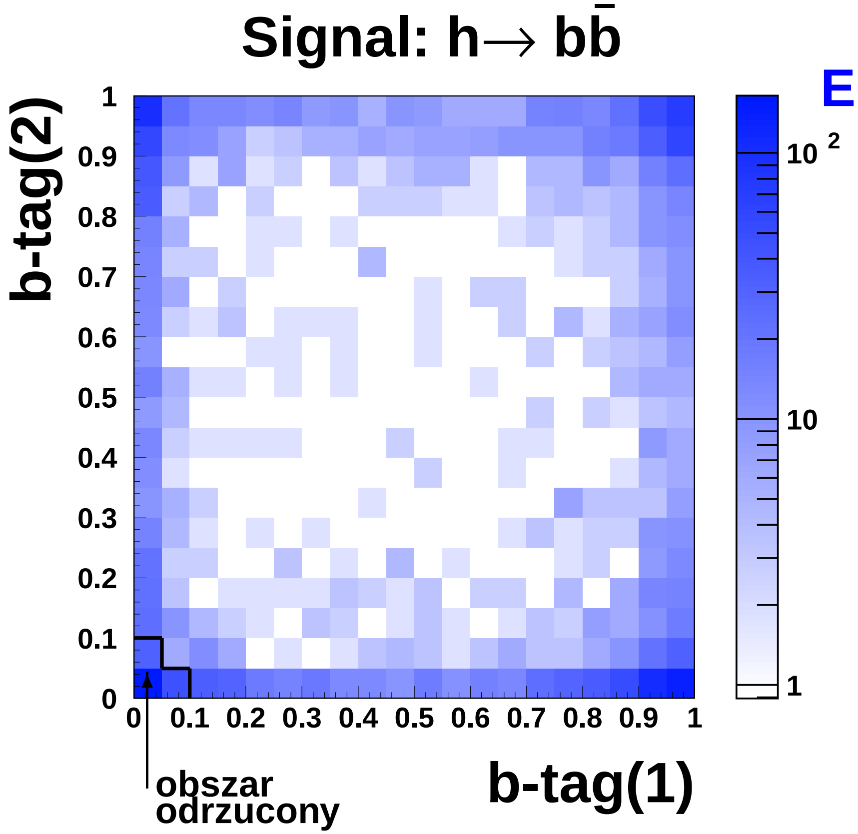
“Jet tagging” - 3





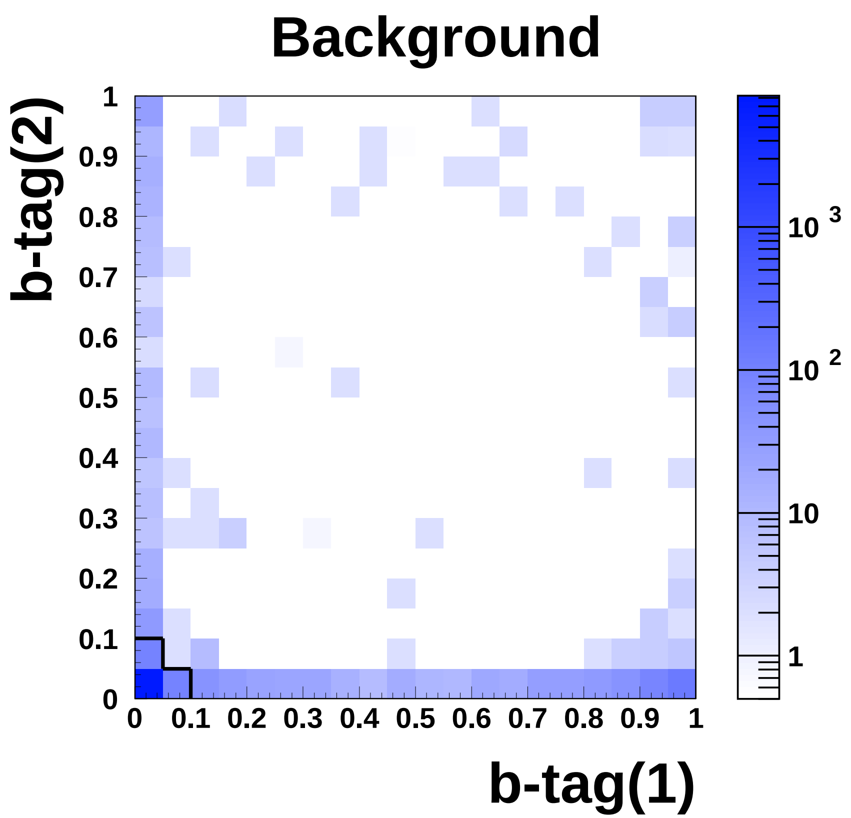


b-tag dla h $\rightarrow$ jet(1) jet(2)


Signal: $h \rightarrow b\bar{b}$

E(1) $>$ E(2)

500 fb^{-1}


Background

Cięcia optymalizujące $\Delta\Gamma/\Gamma$

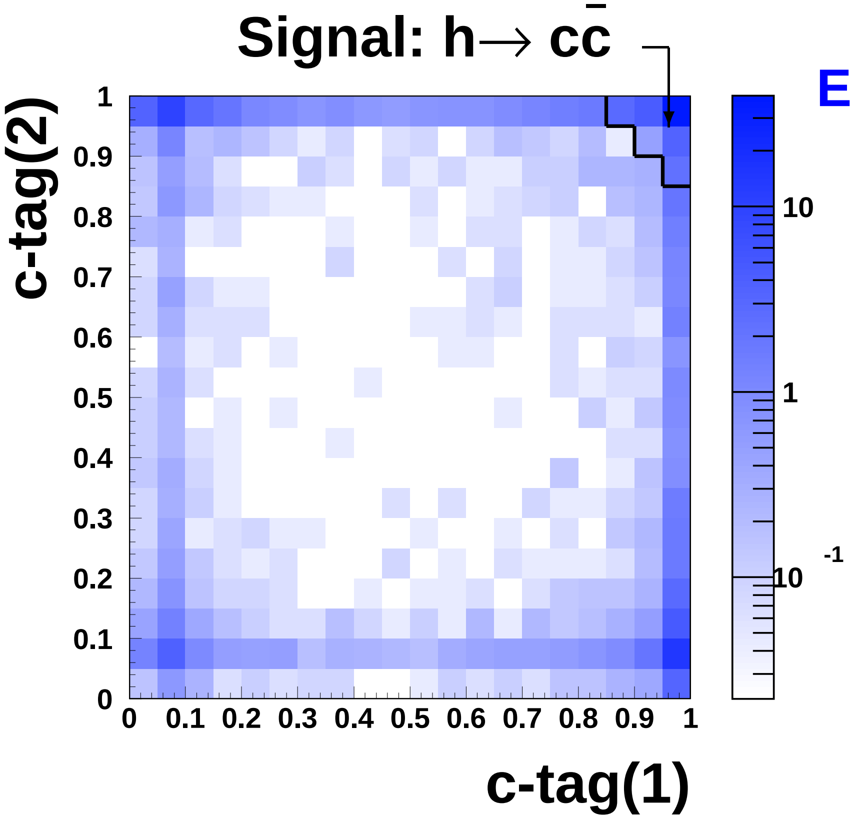
Sygnał:	**2414 przypadków**
Tło:	**785 przypadków**

Precyzja pomiaru:

$$\frac{\Delta\Gamma(h \rightarrow b\bar{b})}{\Gamma(h \rightarrow b\bar{b})} = 2.3\%$$

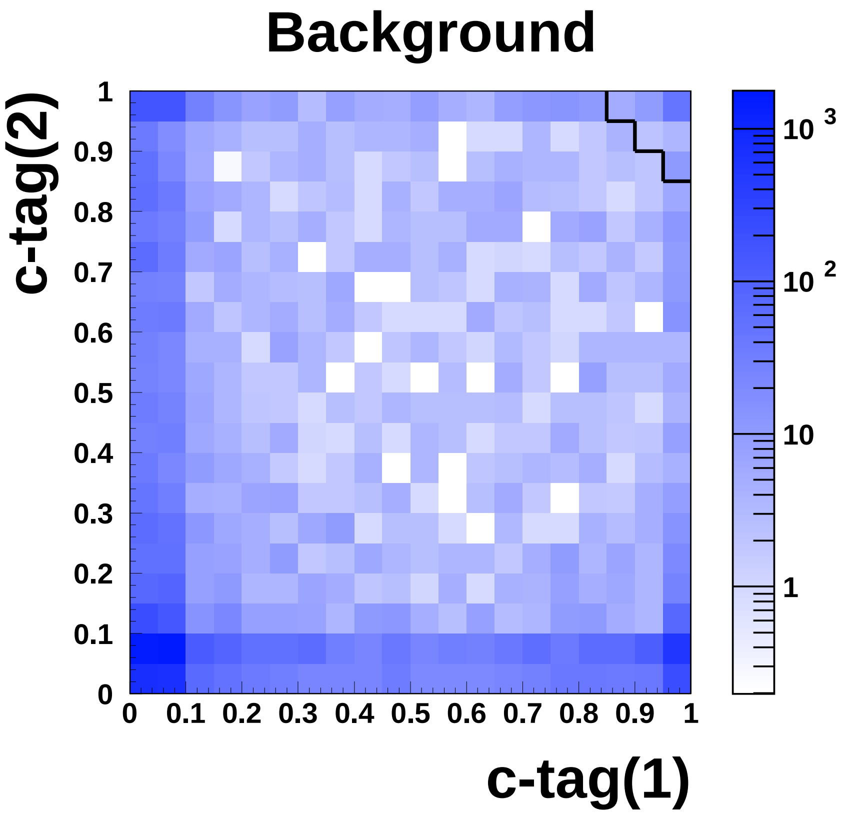
Przykładowe rozkłady dla :
warstwy: 1 - 5
grubość warstwy $50\mu m$
rozdzielczość warstwy $2\mu m$

c-tag dla h $\longrightarrow$ jet(1) jet(2)



E(1) $>$ E(2)

500 fb^{-1}



Cięcia optymalizujące $\Delta\Gamma/\Gamma$

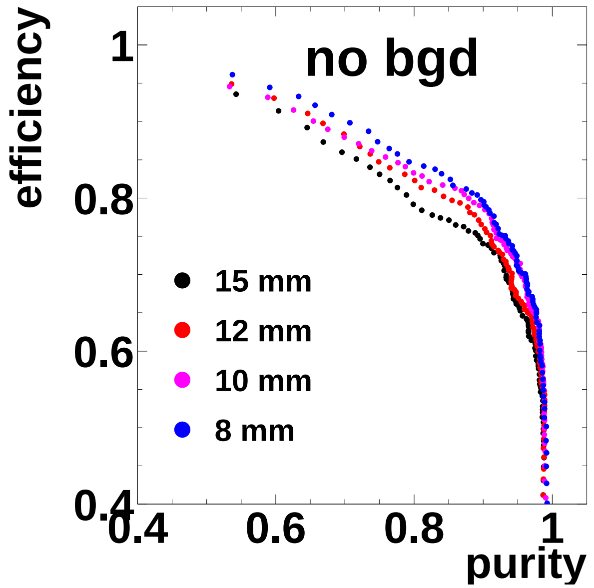
Sygnał:	**42 przypadków**
Tło:	**79 przypadków**

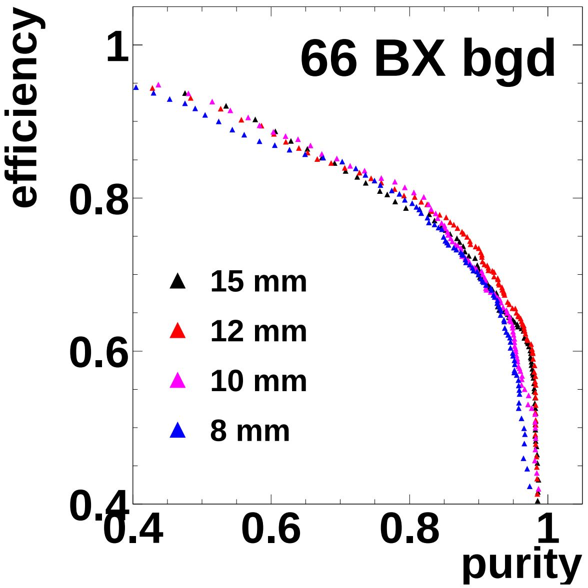
Precyzja pomiaru:

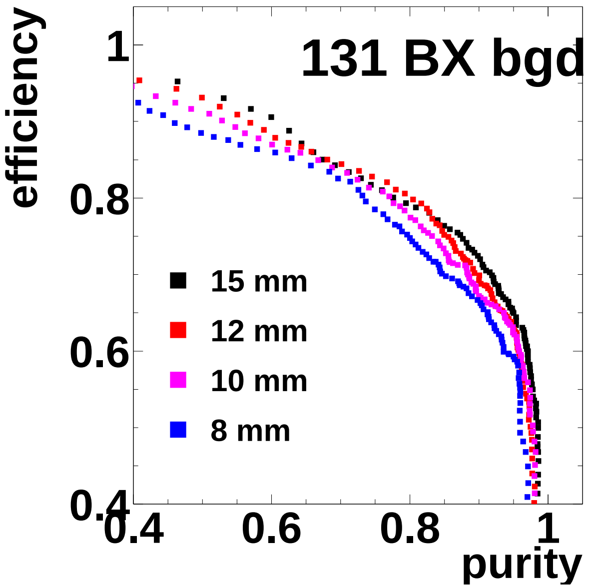
$$\frac{\Delta\Gamma(h\rightarrow c\bar{c})}{\Gamma(h\rightarrow c\bar{c})} = 26\%$$

Przykładowe rozkłady dla :
warstwy: 1 - 5
grubość warstwy $50\mu m$
rozdzielczość warstwy $2\mu m$

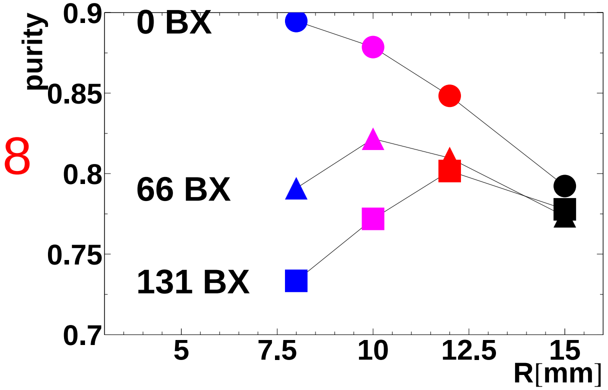
Wpływ tła na identyfikację kwarków b

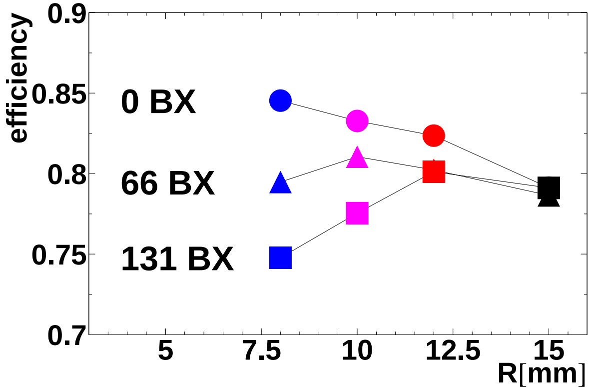






Efficiency 0.8





Purity 0.8

Sampling calorimeters: layer weights

For sampling calorimeters, the signal deposited in each active layer has to be multiplied by an adequate factor to get back the “true” energy

Factors are determined by test beam and/or simulation

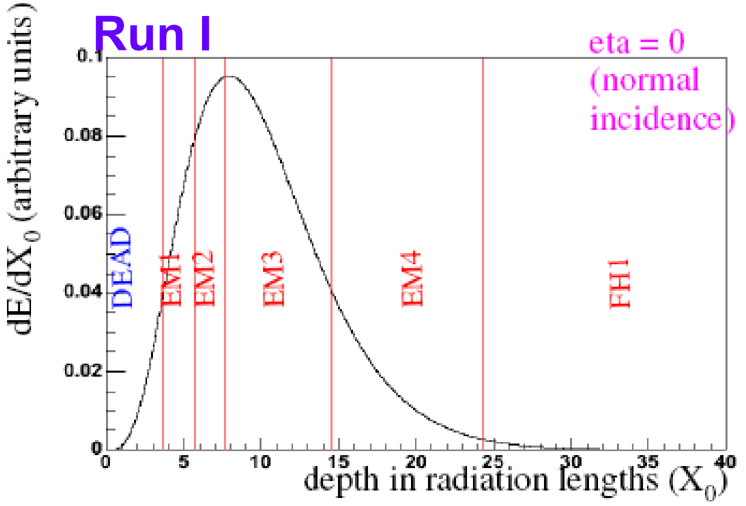
D0 added a solenoide in front of the calorimeter for the Tevatron Run II

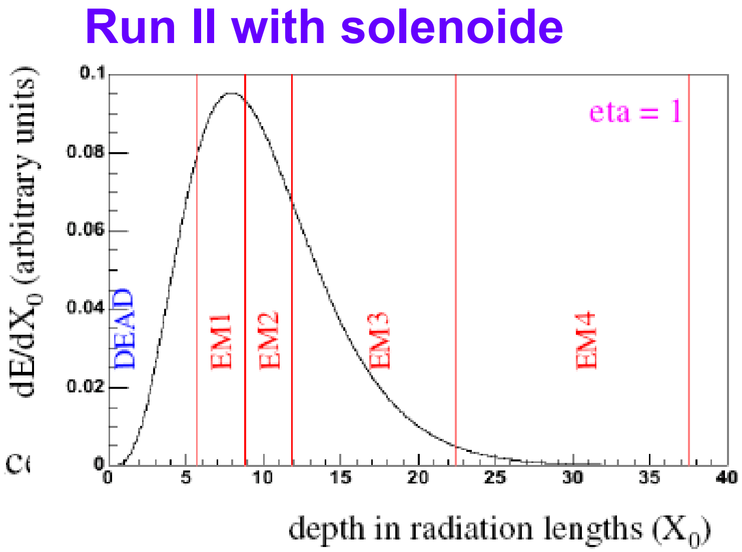
➔increase of dead material

➔➔ the layer weights are increased for the early layers to take the losses into account

According to the angle of the incident particle, the amount of dead material varies

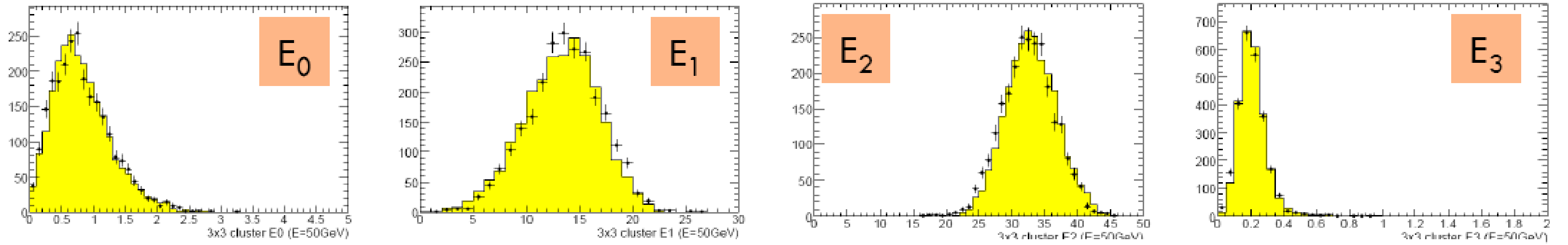
➔ layer weights vary


Run I

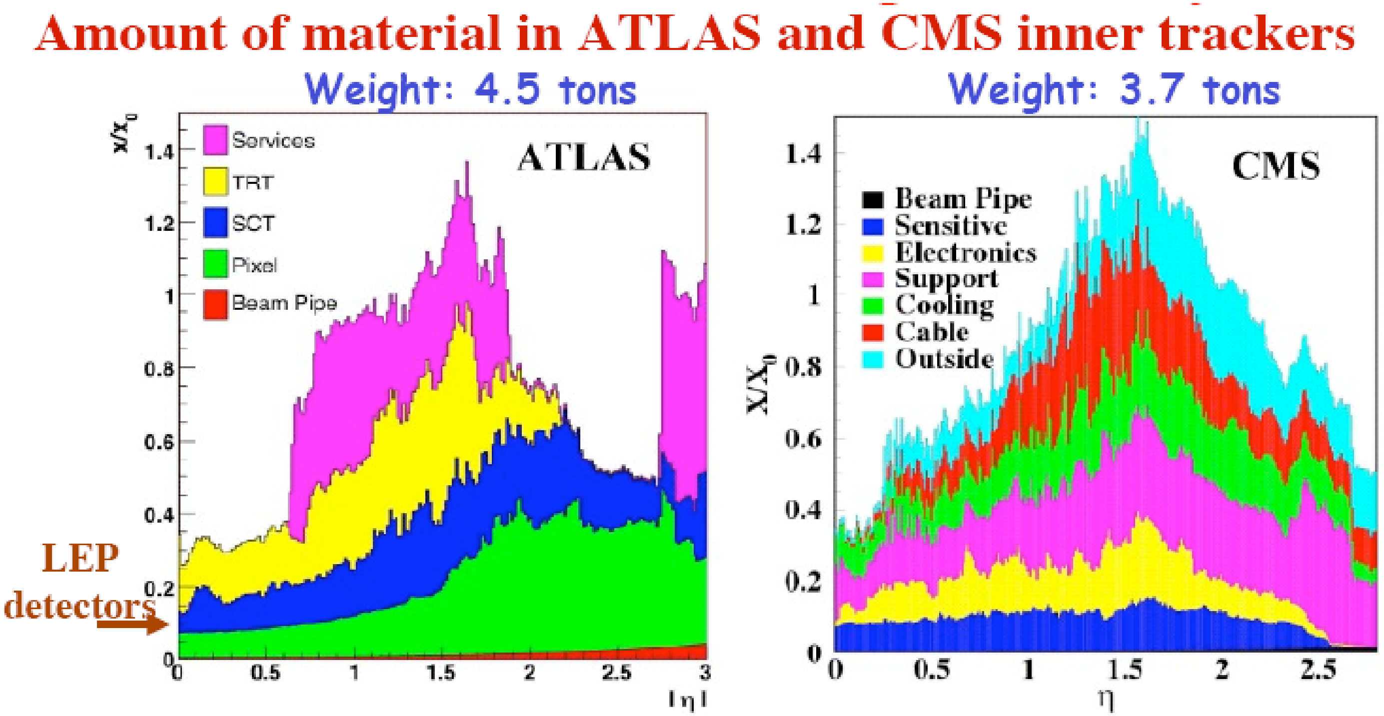

Run II with solenoide

Sampling weights: Data/MC

These sampling weights are determined from test-beam data and simulation



Modeling the amount of dead material in front of the calorimeter is a crucial issue!

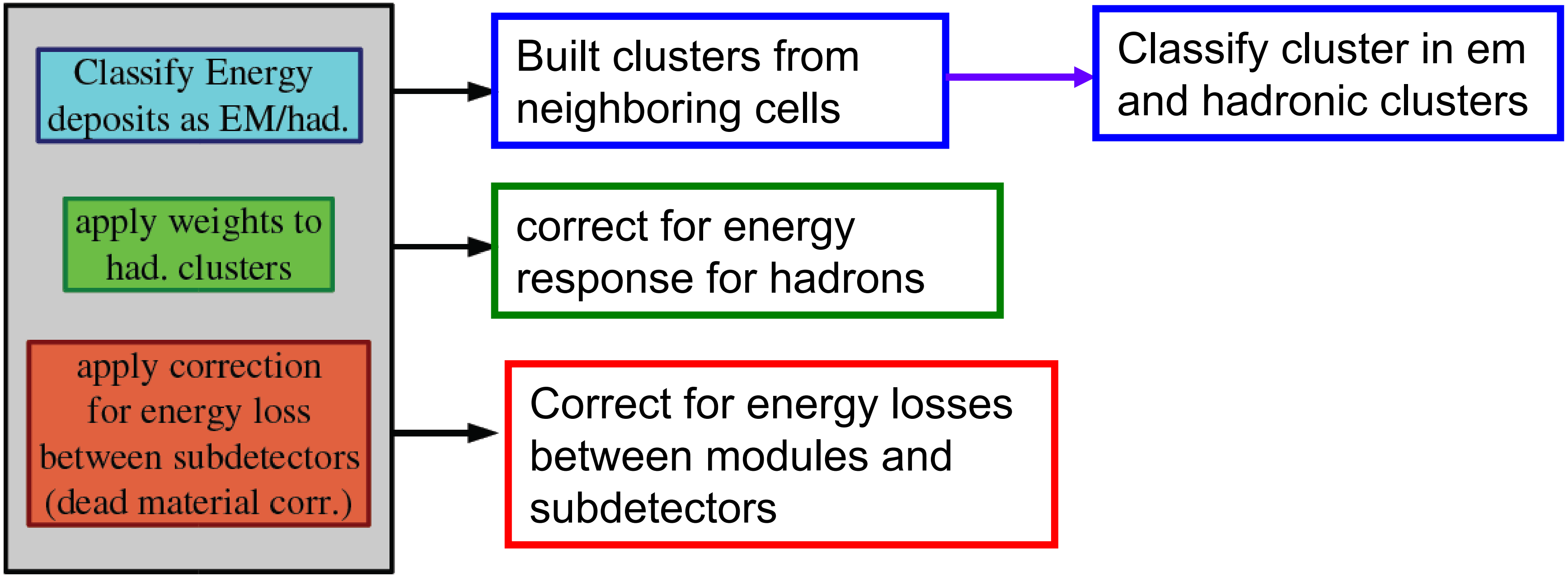

Amount of material in ATLAS and CMS inner trackers

Weight: 4.5 tons

Weight: 3.7 tons

Software compensation

Originaly implemented in H1, now also used by Atlas



Classify Energy deposits as EM/had.

apply weights to had. clusters

apply correction for energy loss between subdetectors (dead material corr.)

Built clusters from neighboring cells

Classify cluster in em and hadronic clusters

correct for energy response for hadrons

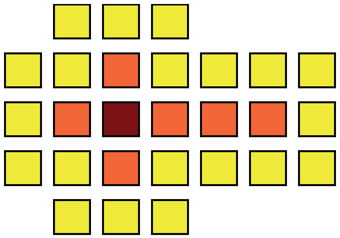
Correct for energy losses between modules and subdetectors

Clusters are reconstructed using “topological algorithm:

- **seeds: highest energy > 4σ**
- **neighbors: energy > 2σ**

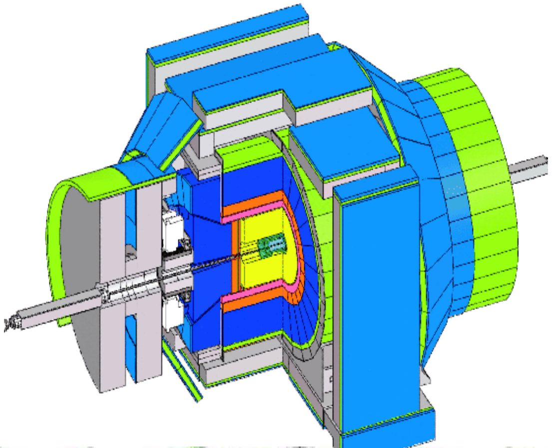
Need to take care of overlapping clusters

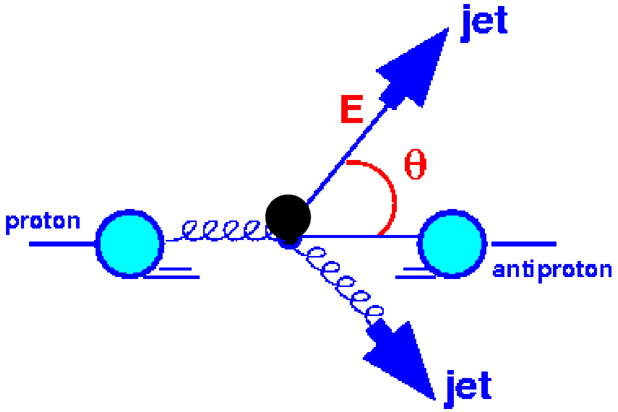
Clusters can span over subdetectors!

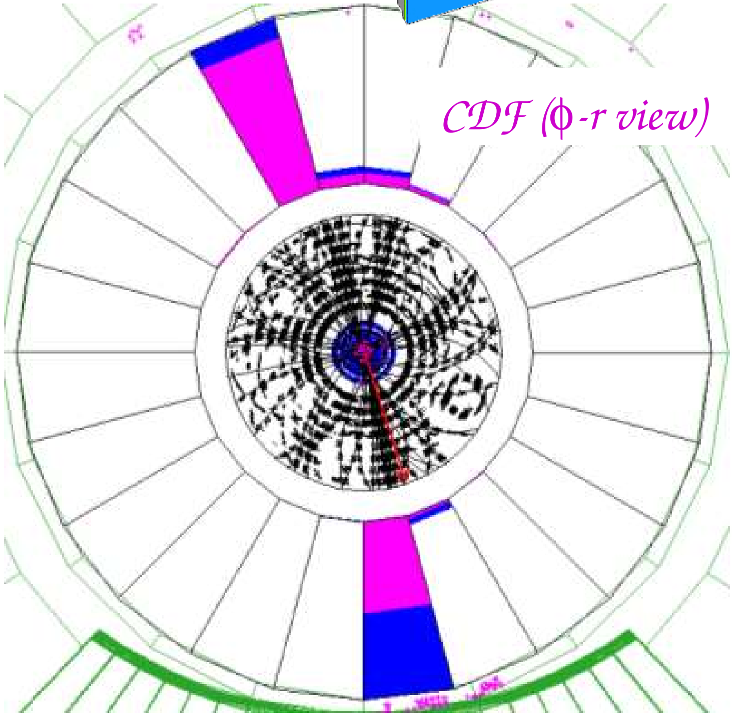


What do Jets Look Like ?

(the highest P_T jet event from CDF)

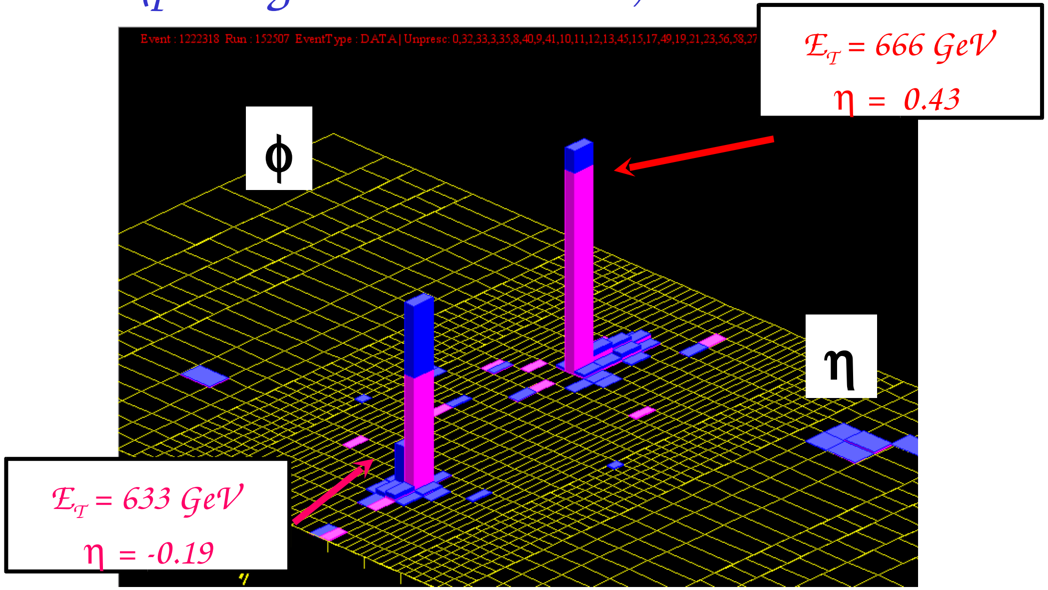






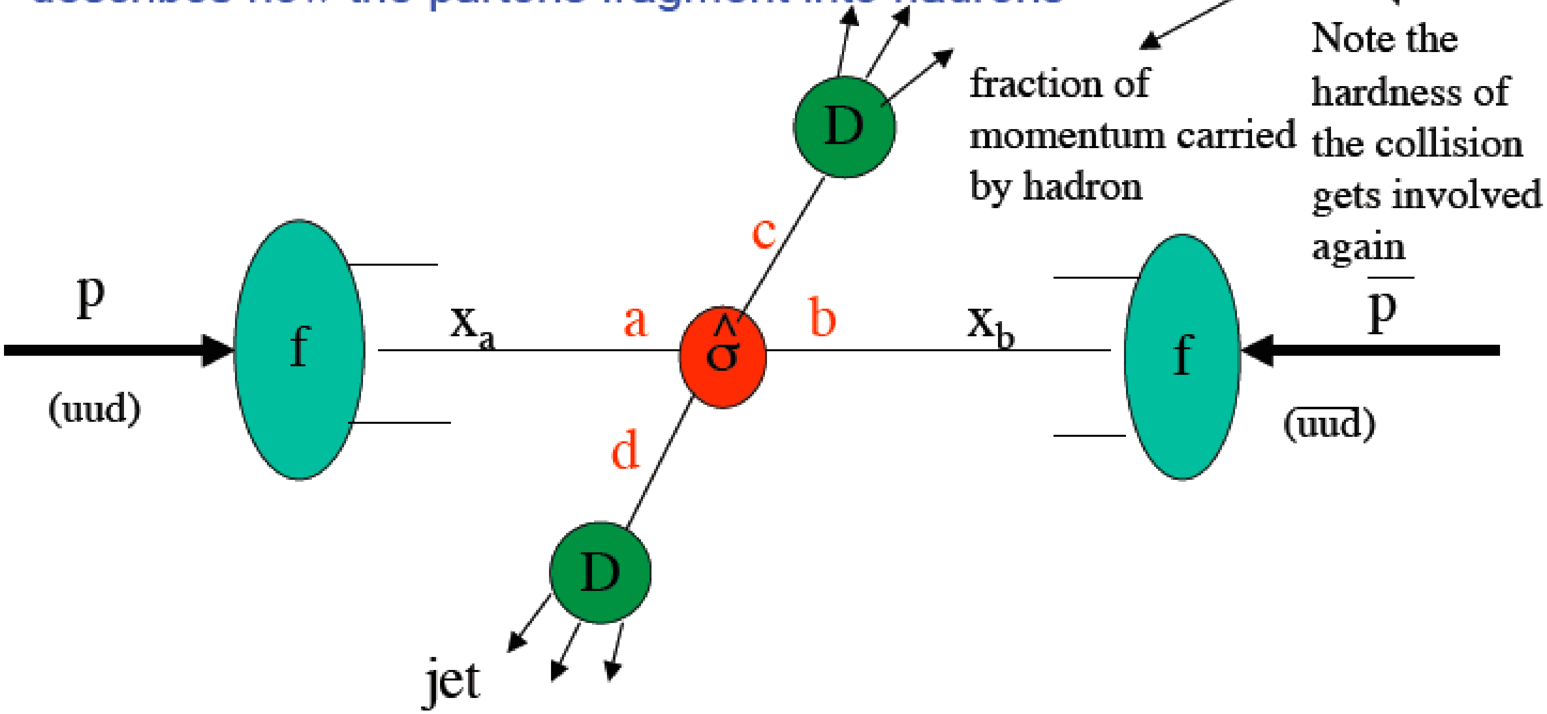
Dijet Mass = 1.36 TeV

(probing distance ~10^{-19} m)



2->2 hard scattering

- c and d are the outgoing partons, but problem: they carry color; we don’t observe the quarks or gluons themselves (don’t see naked color) but instead the partons fragment into jets of hadrons (pions, kaons, and protons)
- it’s the jets of hadrons that we can observe experimentally; $D(z,Q^2)$ describes how the partons fragment into hadrons

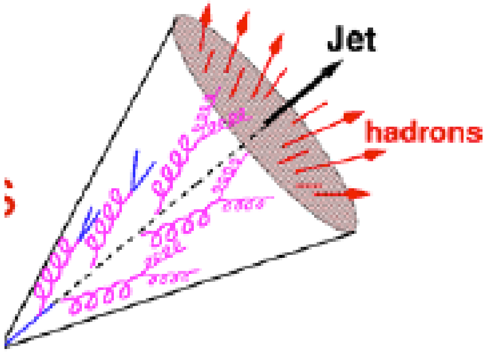


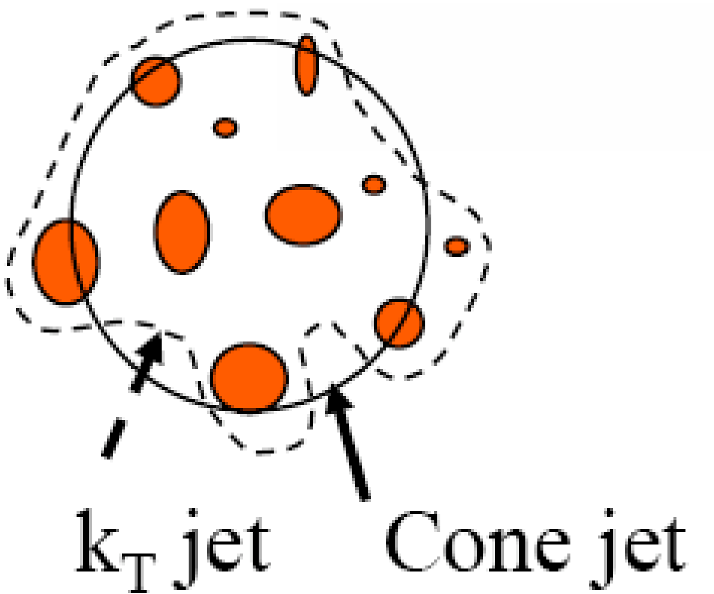
fraction of momentum carried by hadron

Note the hardness of the collision gets involved again

What is a jet?

- Jets are the experimental signatures of quarks and gluons
- Jets manifest themselves as localized clusters of energy
- It is the role of the jet algorithm to identify and measure the properties of a jet
- A jet algorithm can either measure
  - closeness in momentum space: k_T algorithm
    - most often used at LEP and HERA
  - closeness in coordinate space: cone algorithm
    - most often used at the Tevatron
  - at the LHC, hopefully both will be equally used
- Can apply these jet algorithms to calorimeter towers or particles or partons…and would like to get a similar answers, as much as possible

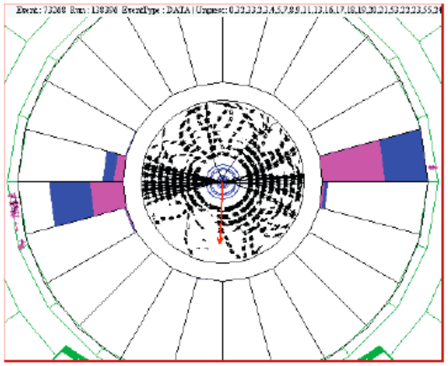


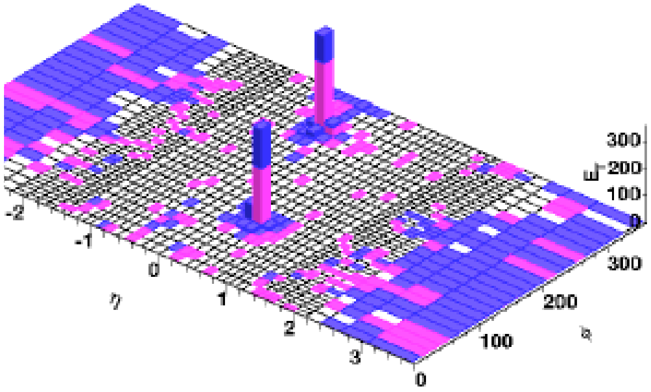


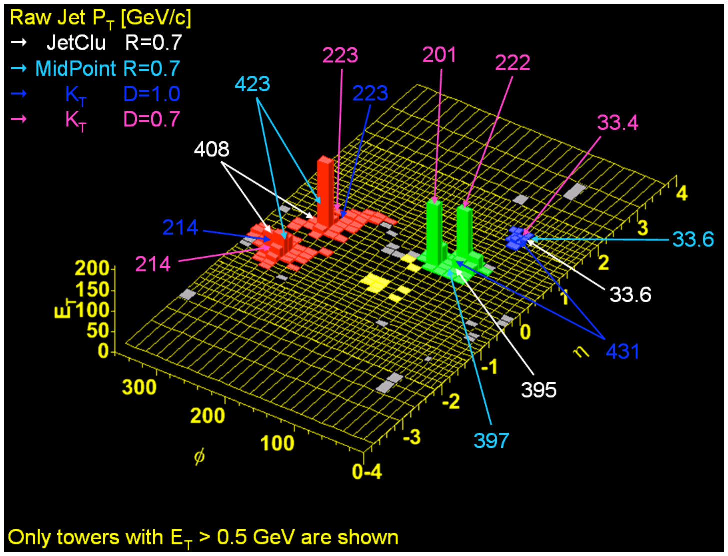
…a cone jet on a bad hair day

Jet algorithms

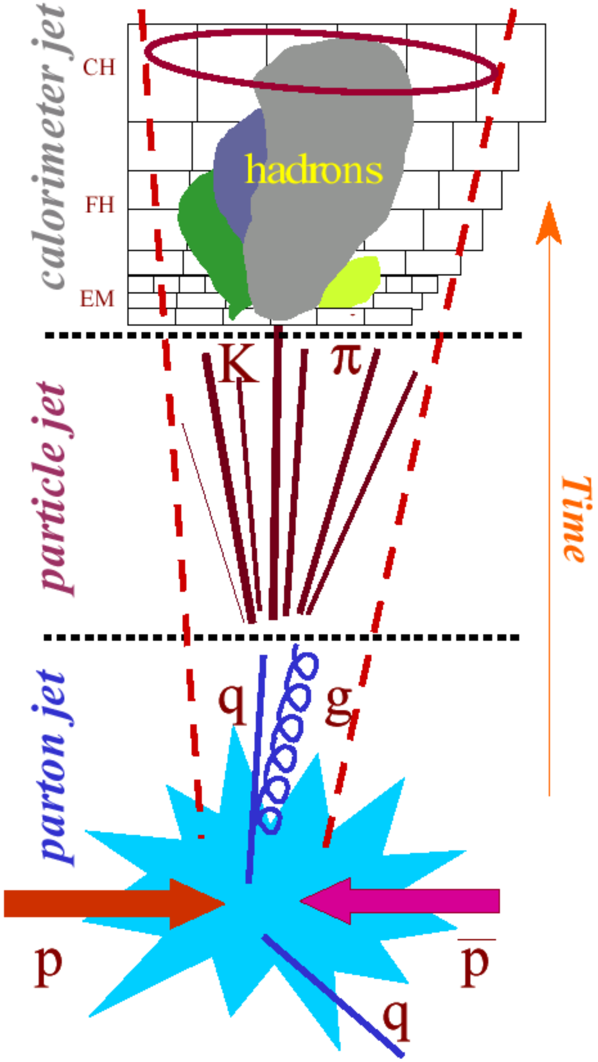
- For some events, the jet structure is very clear and there's little ambiguity about the assignment of towers/particles to the jet
- But for other events, there is ambiguity and the jet algorithm must make decisions that impact precision measurements
- If comparison is to hadron-level Monte Carlo, then hope is that the Monte Carlo will reproduce all of the physics present in the data and influence of jet algorithms can be understood
    - more difficulty when comparing to parton level calculations







Jet Finding



- Calorimeter jet (cone)
  - ◆ jet is a collection of energy deposits with a given cone *R*: $R = \sqrt{\Delta\varphi^2 + \Delta\eta^2}$
  - ◆ cone direction maximizes the total E_T of the jet
  - ◆ various clustering algorithms

➜ correct for finite energy resolution

➜ subtract underlying event

➜ add out of cone energy

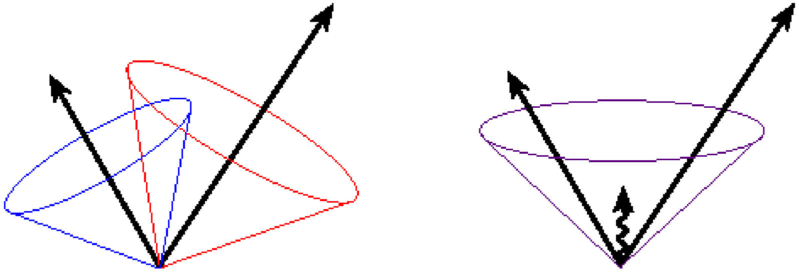
- Particle jet
  - ◆ a spread of particles running roughly in the same direction as the parton after hadronization

Jet Algorithms: Cone

- **Cone algorithms**
  - **draw a cone of fixed size around a seed**
  - **compute jet axis**
  - **draw a new cone around the new jet axis and recalculate axis and new E_T**
  - **iterate until stable**

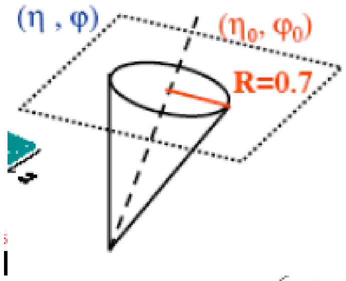
→In addition:

- **add additional midpoint seeds between pairs of close jets**
- **split/merge after stable proto-jets found**



Midpoint cone algorithm

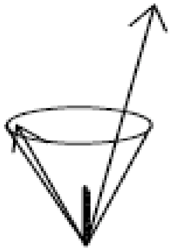
- Generate p_T ordered list of towers (or particles/partons)
- Find proto-jets around seed towers (typically 1 GeV) with p_T>threshold (typically 100 MeV)
  - include tower k in cone if

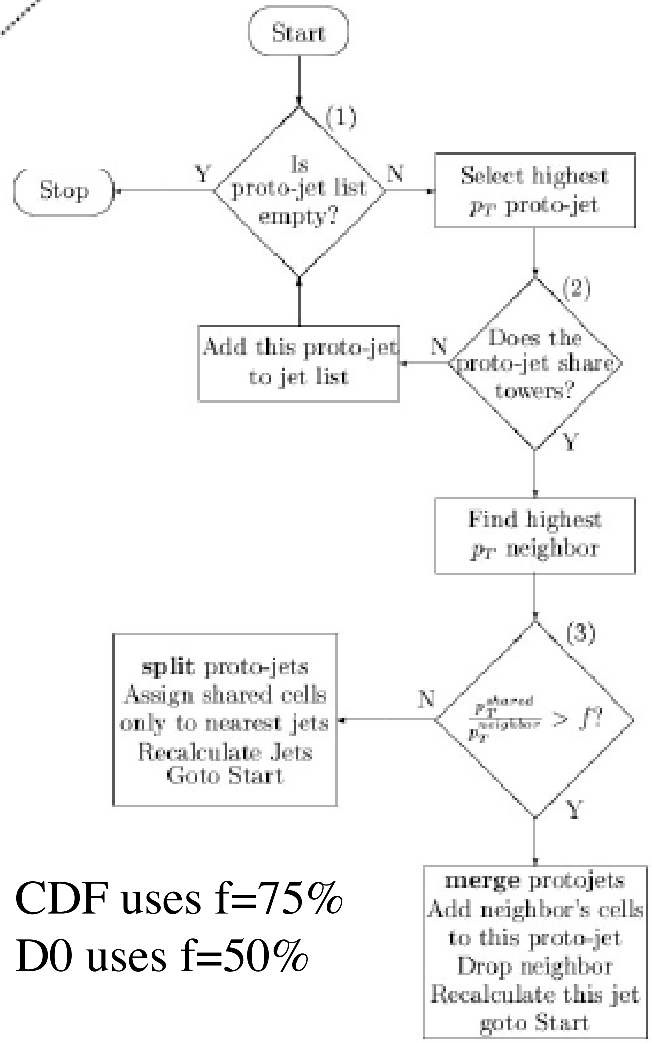


$$k \subset C \text{ iff } \sqrt{(y_k - y_C)^2 + (\phi_k - \phi_C)^2} \le R_{cone},$$

$$p_C = (E_C, \vec{p}_C) = \sum_{k \subset C} (E_k, \vec{p_k}), \; \overline{y}_C \equiv \frac{1}{2} \ln \frac{E_C + p_{z,C}}{E_C - p_{z,C}}, \; \overline{\phi}_C \equiv \tan^{-1} \frac{p_{y,C}}{p_{x,C}}$$

  - iterate if $(\overline{y}_C,\overline{\phi}_C) \neq (y_C,\phi_C)$
  - NB: use of seeds creates IR-sensitivity
- Generate midpoint list from proto-jets
  - using midpoints as seed positions reduces IR-sensitivity
- Find proto-jets around midpoints
- Go to splitting/merging stage
  - real jets have spatial extent and can overlap; have to decide whether to merge the jets or to split them
- Calculate kinematics (p_T,y,ϕ) from final stable cones





CDF uses f=75%
D0 uses f=50%

Jet Algorithms: k_T

For each object and pair of objects:

$$d_{ii} = k_{T,i}^2$$

$$d_{ij} = \underbrace{\min(k_{T,i}^2, k_{T,j}^2)}_{\text{Soft}} \frac{\Delta R_{ij}^2}{\boxed{D^2}}$$

Collinear
(if $\Delta R << 1$)

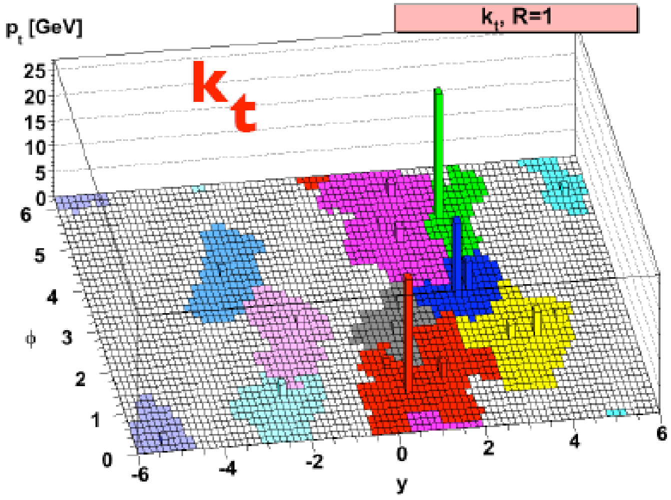
Resolution parameter (D=1)

order all d_{ii} and d_{ij}:

If **$d_{min}=d_{ij}$**
$\Rightarrow$ merge particles

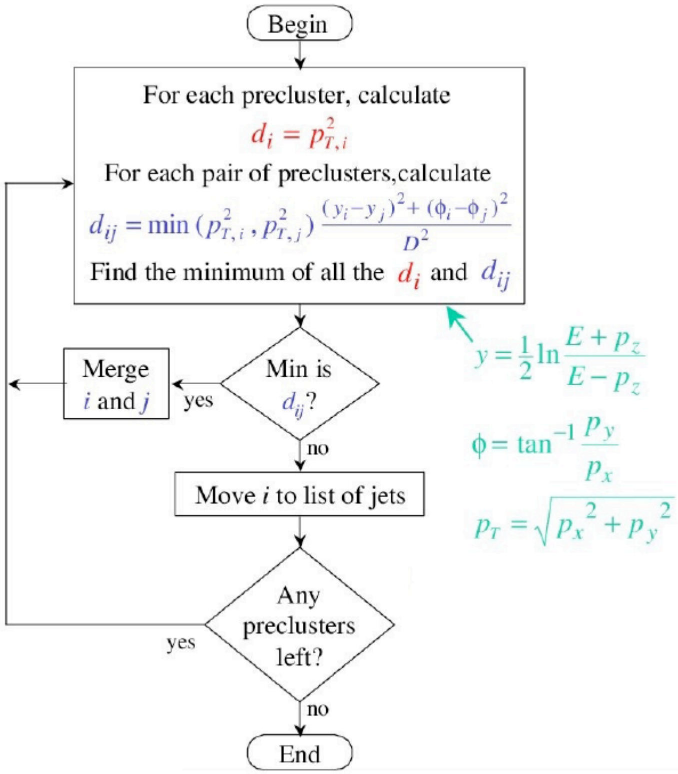
If **$d_{min}=d_{ii}$**
$\Rightarrow$ jet

- **theoretically favored, no split-merge**
- **to reduce computation time, start with 0.2 x 0.2 pre-clusters**



k_T algorithm

- The k_T jet algorithm successively merges pairs of partons, particles or calorimeter towers in order of increasing relative transverse momentum
- The algorithm typically contains a parameter D that controls the termination of the merging and characterizes the approximate size of the resulting jets
- Since the k_T algorithm fundamentally merges nearby particles,there is a correspondence of jets reconstructed in a calorimeter to jets reconstructed from individual hadrons, leptons, and photons
- As the jet does not have a fixed area, the underlying event subtraction is more problematic



Merging scheme:

4-vector addition

(E-scheme)

$p_x = p_{x,i} + p_{x,j}$

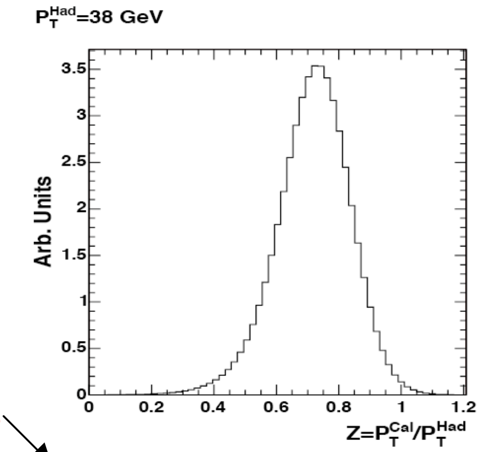
$p_y = p_{y,i} + p_{y,j}$

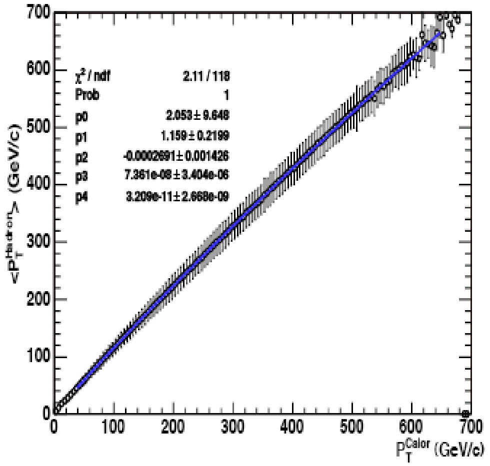
$p_z = p_{z,i} + p_{z,j}$

$E = E_i + E_j$

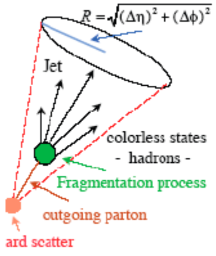
Jet Corrections

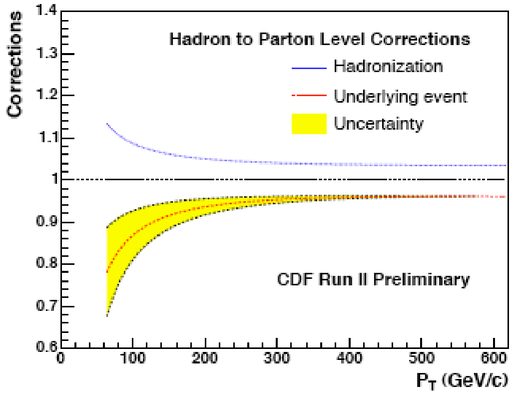
- Need to correct from calorimeter to hadron level (different response of calorimeter to EM and HAD energy)
  - and for resolution effects
- And from hadron to parton level for other observables (such as comparisons to parton level cross sections)
  - underlying event and out-of-cone
    - can correct data to parton level or theory to hadron level…or both and be specific about what the corrections are
  - note that loss due to hadronization is basically constant at 1 GeV/c for all jet p_T values at the Tevatron (for a cone of radius 0.7)
    - for a cone radius of 0.4, the two effects cancel to within a few percent
  - interesting to check over the jet range at the LHC

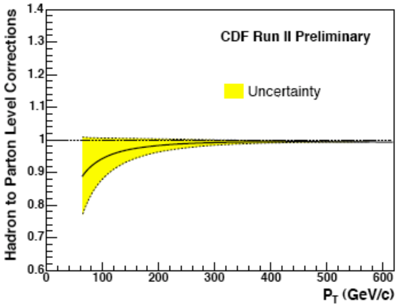




partons in cone give rise to hadrons outside the cone

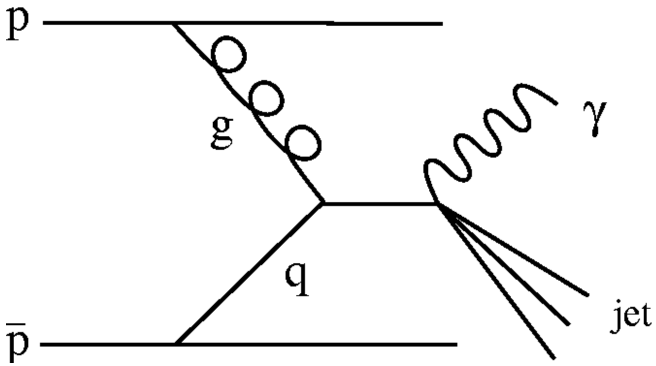






for cone of 0.7, UE correction wins

Response correction



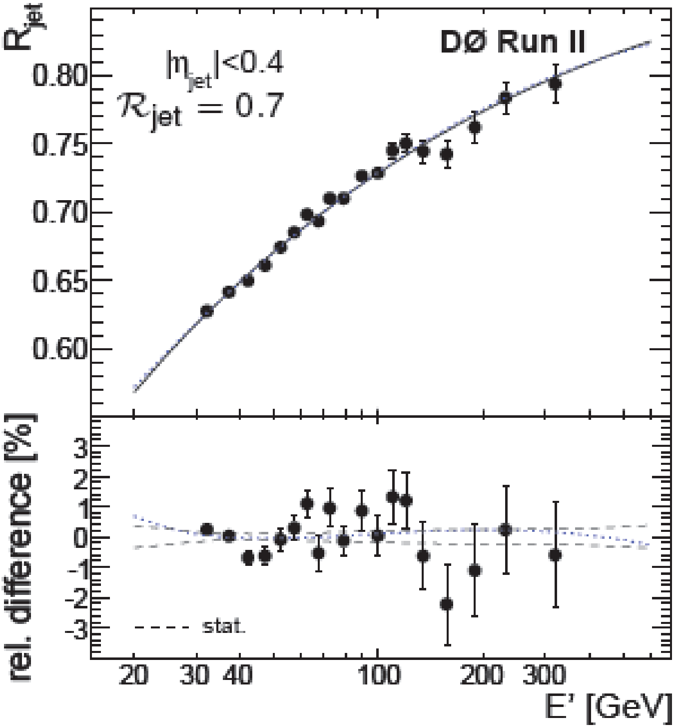
One possibility:

use γ+jet events

-γ well measured and calibrated

- missing transverse energy projection:

$$R_{\mathrm{had}} = 1 + \frac{\vec{E}_{\mathrm{T}} \cdot \vec{p}_{\mathrm{T},\gamma}}{\vec{p}^{\,2}_{\mathrm{T},\gamma}}$$

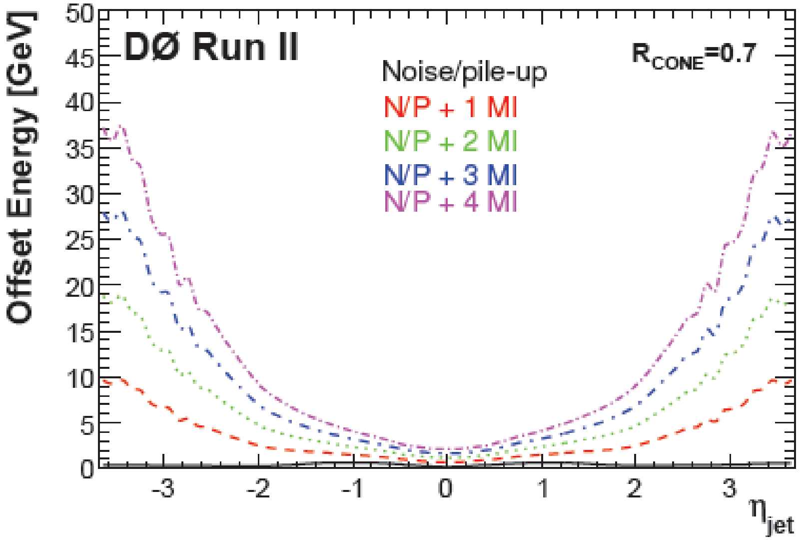


The 25% correction to the response is the largest of all energy scale contributions

Offset corrections

Subtract all energy inside the jet cone not related to the hard interaction:

- electronics and uranium noise
- multiple $p\bar{p}$ interactions in the same bunch crossing
- left-overs from previous bunch crossings (pile-up)



Noise/pile-up

Estimated using zero-bias data; data triggered on the presence of bunch crossings and vetoing any hard interactions

Multiple interactions

Measured in minimum-bias data; data triggered using the luminosity monitors to signal potential inelastic scatters. Contribution from additional interactions determined from:

$$\mathrm{MI}(N_{PV}, L) = \mathrm{MinBias}(N_{PV}, L) - \mathrm{MinBias}(N_{PV} = 1, L)$$

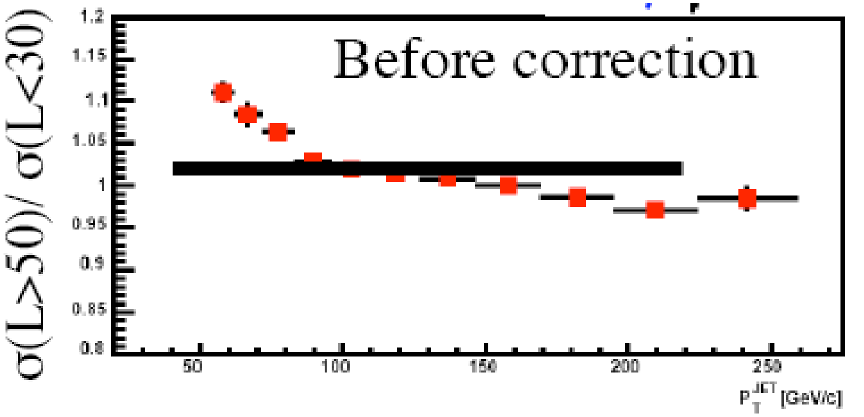
JES for k_T algorithm

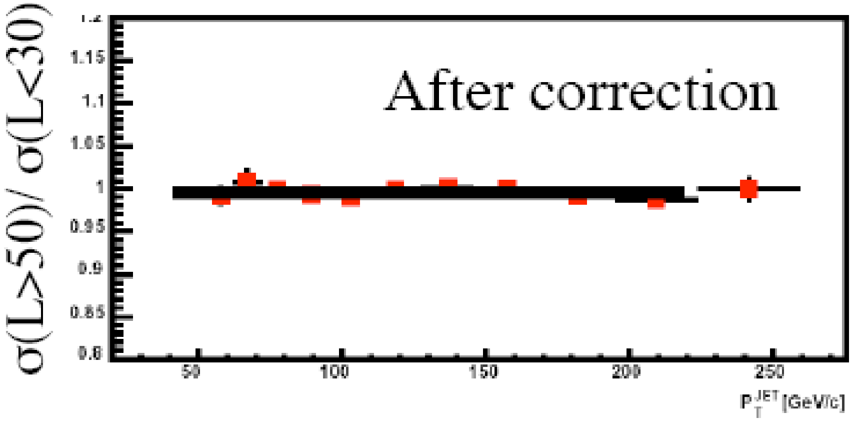
Problem of k_T algorithm in hadron colliders: Multiple interactions

•k_T algorithm is attracted to energy deposits and picks up the energy from MI, UE etc<.

• k_T algorithm has now been used by CDF to measure a cross-section for Run2:

• emiprical energy correction factor used using the fact that the cross-section is luminosity independent





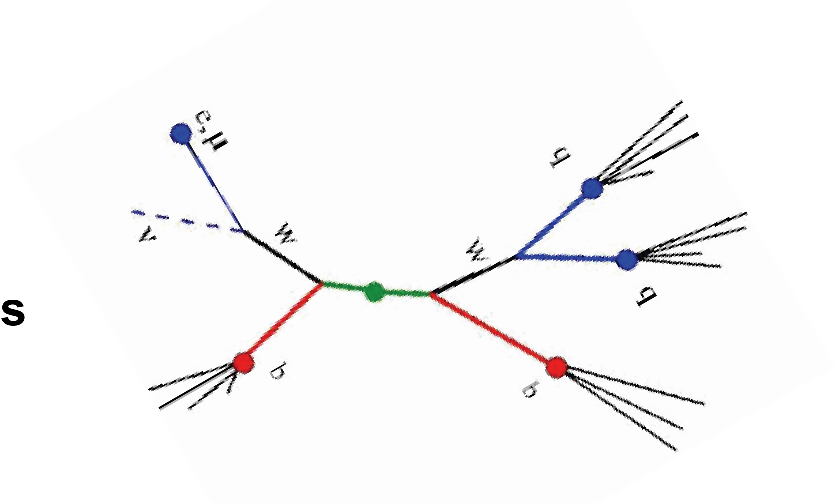
Jet calibration from top events

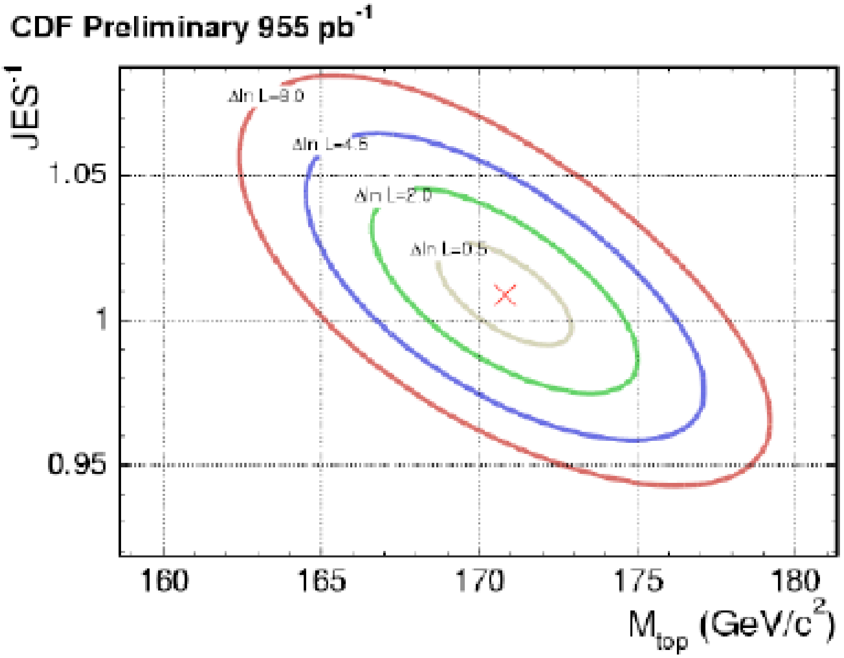
Use the W→ jj decay in top events to calibrate the jet energy scale

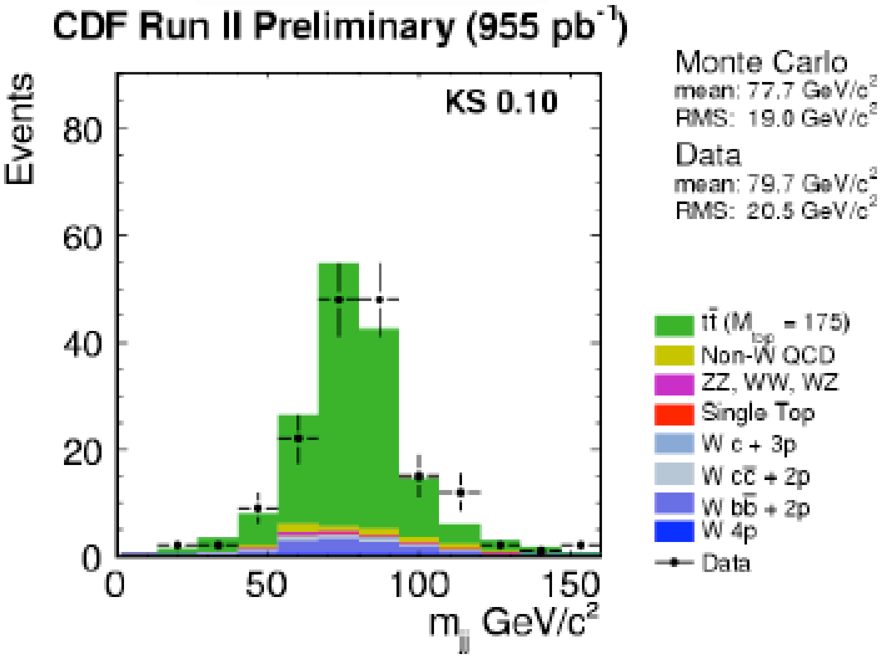
Fit the JES in data an MC

Constraints JES to 2% with 166 evnets

➔ At LHC: 45 000 top events/month!





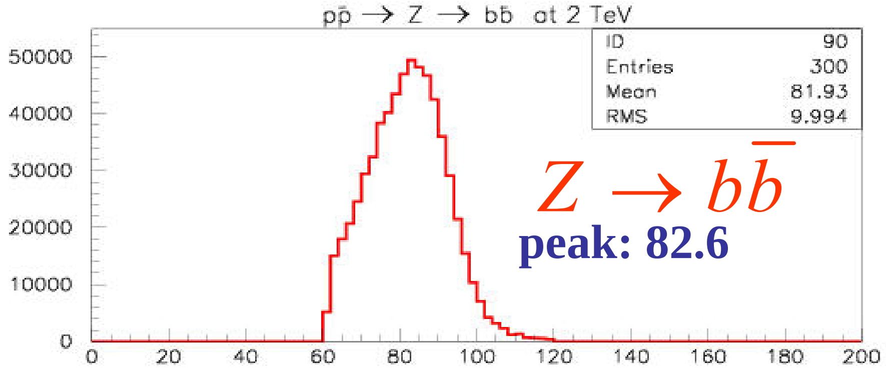


b-jet calibration

- **naïve reconstruction of Z-mass shows a lower mass for selected b-jets than light quark jets.**

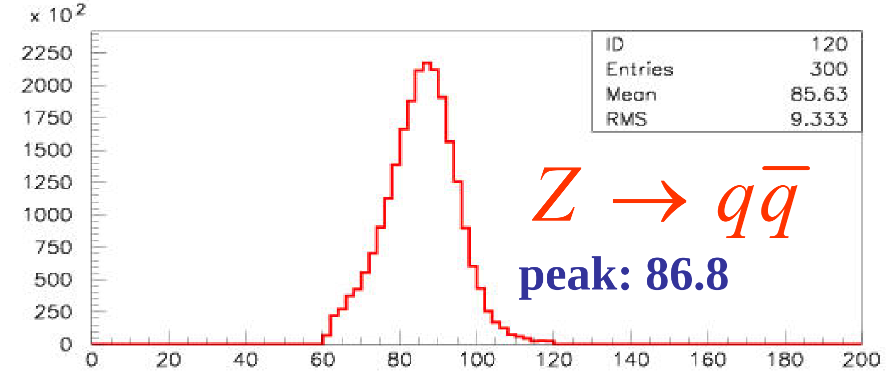
⇒ energy losses from semi-leptonic b decays (ν, μ)

⇒ wider b-jets (due to the large b-mass)



$Z \to b\bar{b}$

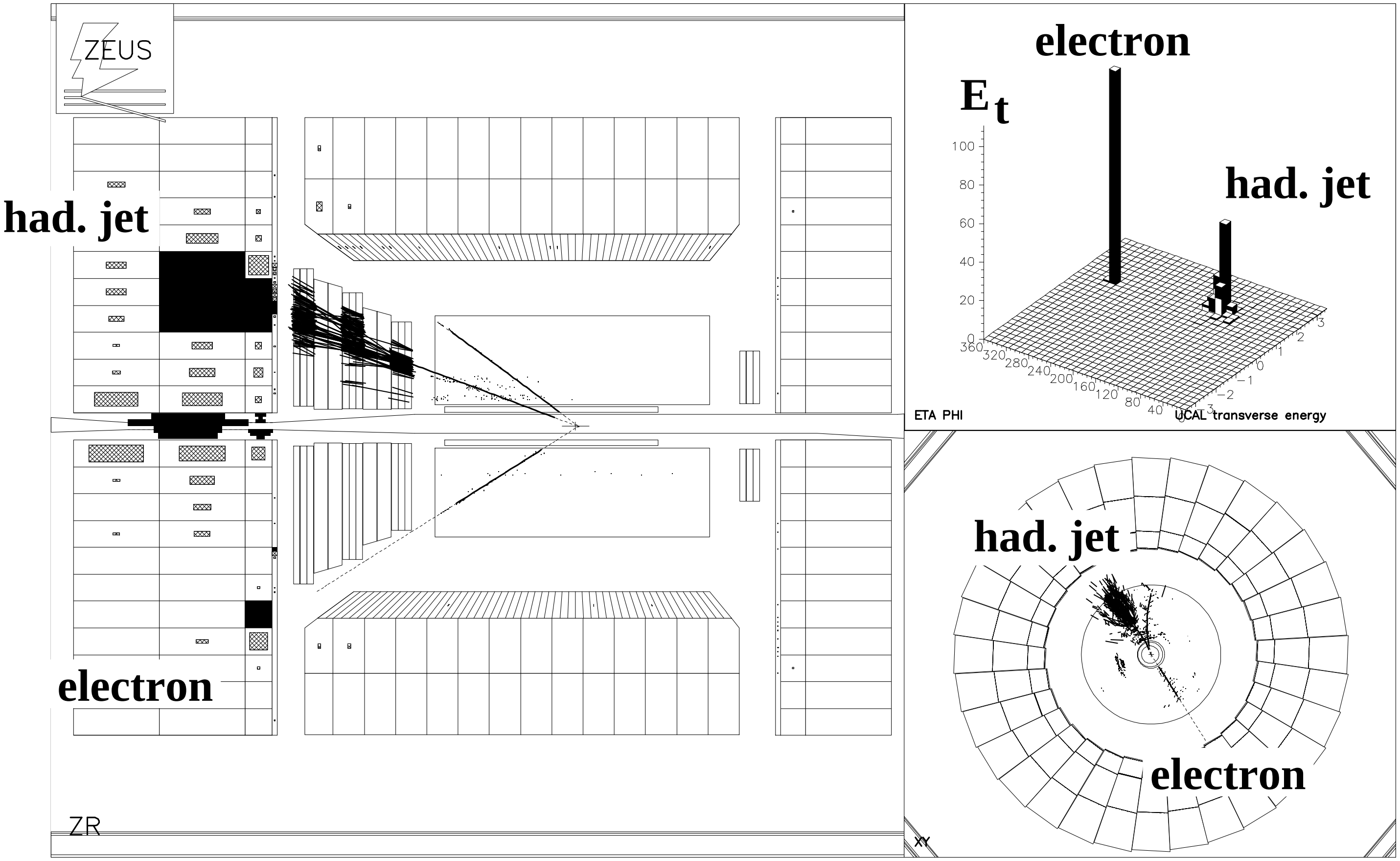
peak: 82.6



$Z \to q\bar{q}$

peak: 86.8

Main features of a scattered electron

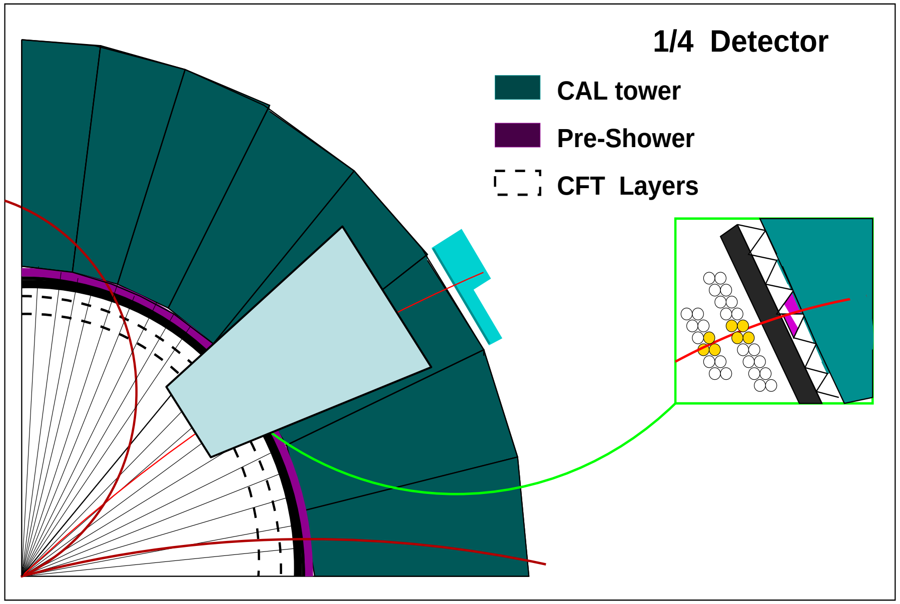


- **Calorimeter:** isolated, narrow EMC cluster
- **CTD:** track that matches EMC cluster (position & momentum)

EM reconstruction

- **basically: look for narrow isolated clusters with high EM fraction**
  - **cone r=0.2**
  - **$E_{em}/E_{had} > 0.9$**
  - **isolation:**

$$\frac{E_{total}^{cone0.4} - E_{EM}^{cone0.2}}{E_{EM}^{cone0.2}} < 0.2$$

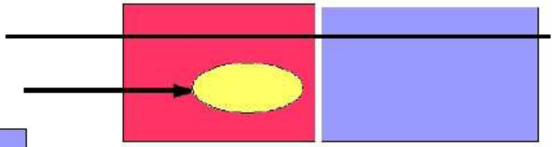


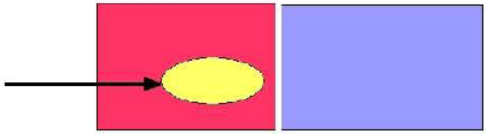
- **track match for electrons (spatial and E/p), none for γ:**
- **matching in preshower detectors**

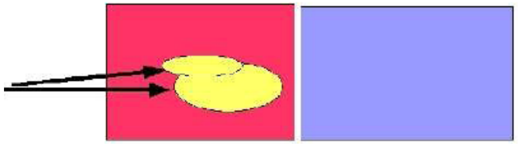
EM identification

Main backgrounds:

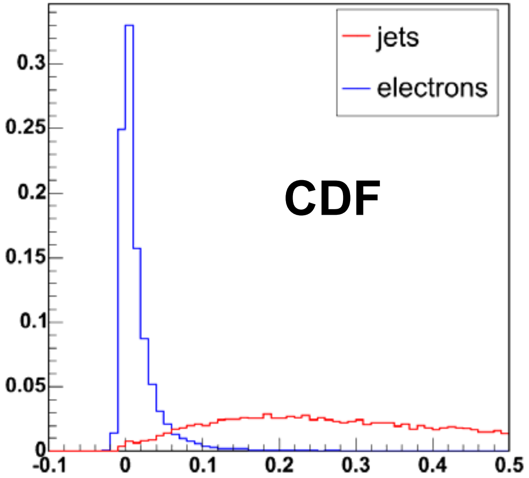
- overlay over a π^0 and a non-interacting $\pi^{+/-}$
- early showering π+/-
- photon conversions



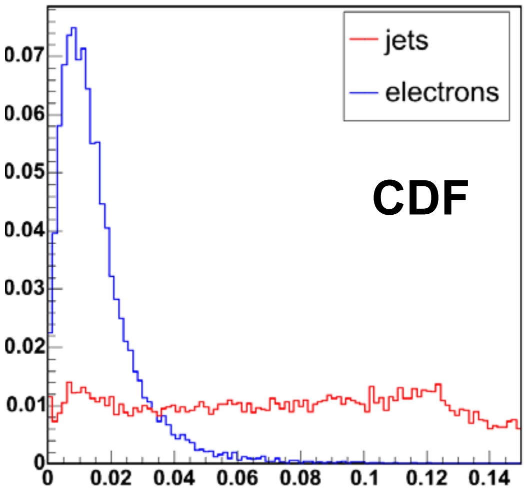




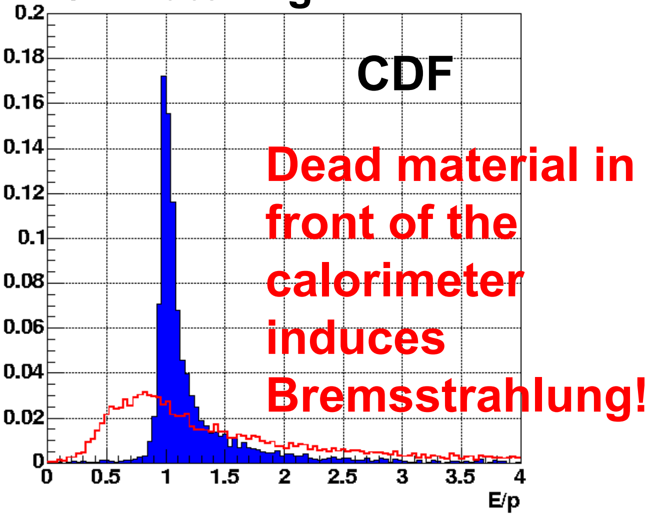
Calorimeter Isolation



Had/em Ratio



E/P matching



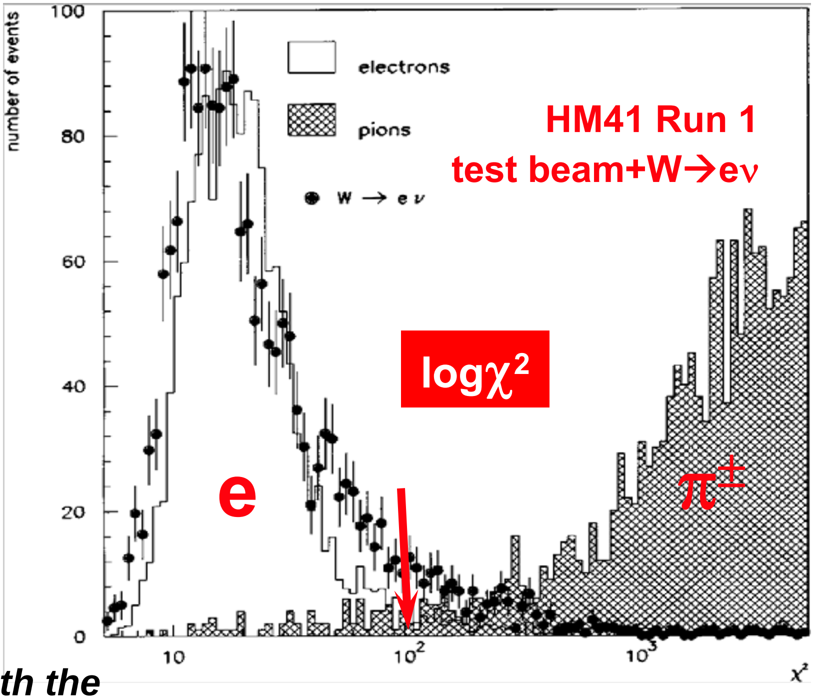
Dead material in front of the calorimeter induces Bremsstrahlung!

Using shower shapes

H-matrix

- use longitudinal and transverse shower shapes to take into account correlations between energy in cells
- measure compatibility of EM cluster with an electron shower → χ^2
- discriminate against hadronic (π) decays that pass EM fraction and isolation cuts

- tuned on MC in $\Delta\eta$ bins of 0.1, $|\eta| < 3.2$ for different energies
- HMx8 / HMx9 / Hmx41
  - energy fractions in each floor (PS), EM1, EM2, EM3, EM4
  - $\Delta\eta$, $\Delta\varphi$ in EM3 → grid (6,6)
  - $\log(E_{tot})$
  - Z/σ_z vertex



For a description of the H-matrix: "*Top Quark Search with the D0 1992-1993 Data Sample*" Phys. Rev. D52 4877 (1995)

How does EM work? (II)

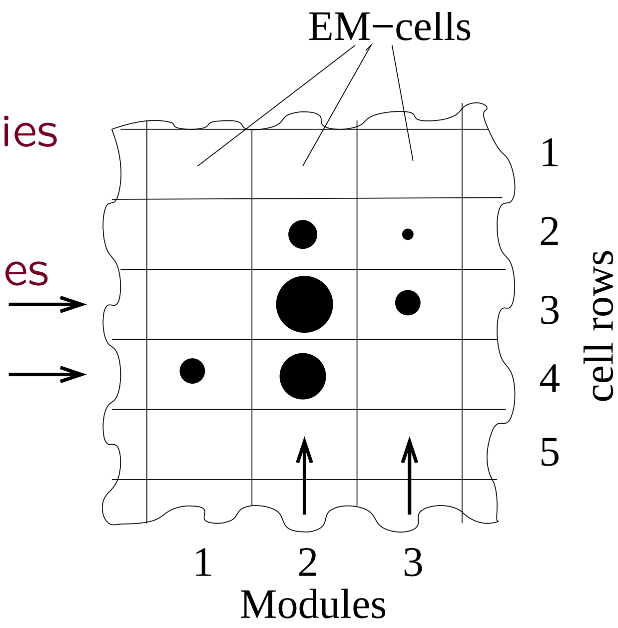
- **Decision variables:**

  1 Energy fraction outside cell rows with highest energies (width of shower along module boundary)

  2 Energy fraction outside modules with highest energies (width of shower transvers to module boundary)

  3 Energy fraction in hadronic section of CAL (depth of shower)

  4 Energy inside η-ϕ cone around candidate (isolation, standard radius 0.8; ϕ in rad)



  5 Difference between θ from track and island

  6 Difference between ϕ from track and island

  7 Difference between track momentum and CAL energy

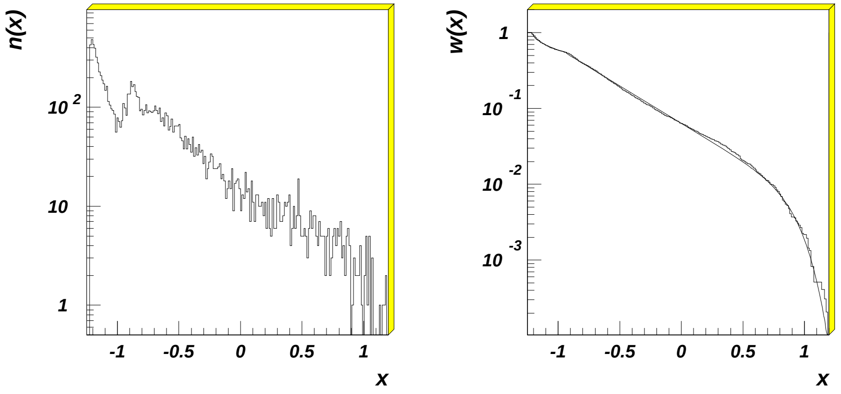
} set to 1 for trackless candidates

How does EM work? (III)

- For each of the 7 variable values x a probability is calculated

Determination of probability distribution for a decision variable:

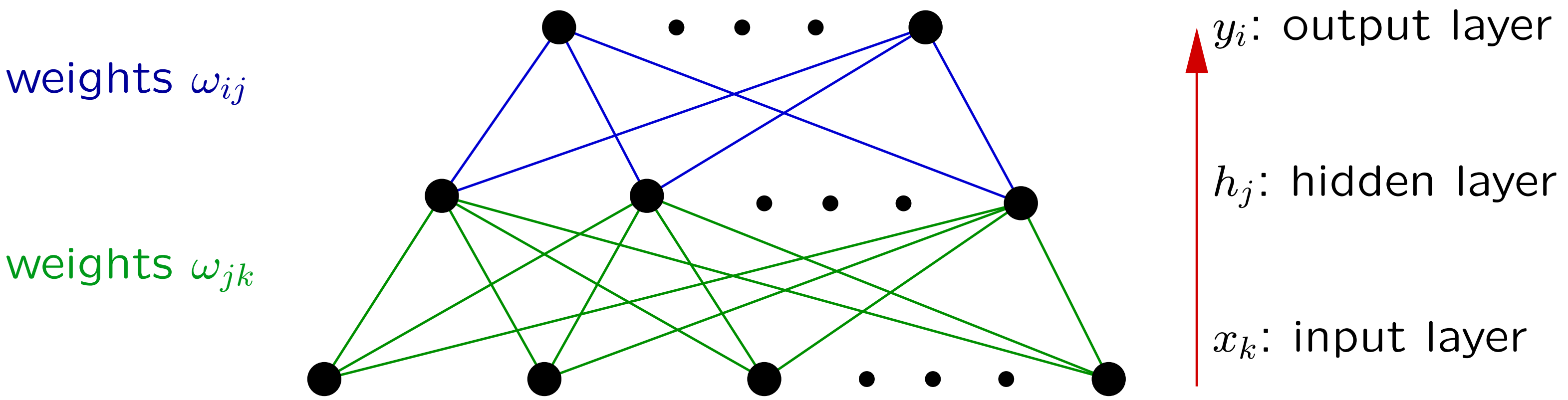
- Take variable distribution for true electrons (separately for F/B/RCAL)
- Calculate $w(x) = \frac{\int_x^\infty n(x')dx'}{\int_{-\infty}^\infty n(x')dx'}$
- Parameterize $w(x)$
- $w(x)$ is probability that a true electron has a variable value $> x$



- **Grand Probability** is product of all 7 individual probabilities
- **Selection Probability** is calculated from Grand probability taking p_t of candidate and $1/Q^2$ obtained with candidate into account
  $\Rightarrow$ **best candidate** is the one with the **highest Selection Probability**
- **Grand Probability** is used to **cut on output value of EM** (0.001 inside CTD acceptance, 0.01 outside CTD acceptance)

What is a neural network?

Feed-Forward Neural Network

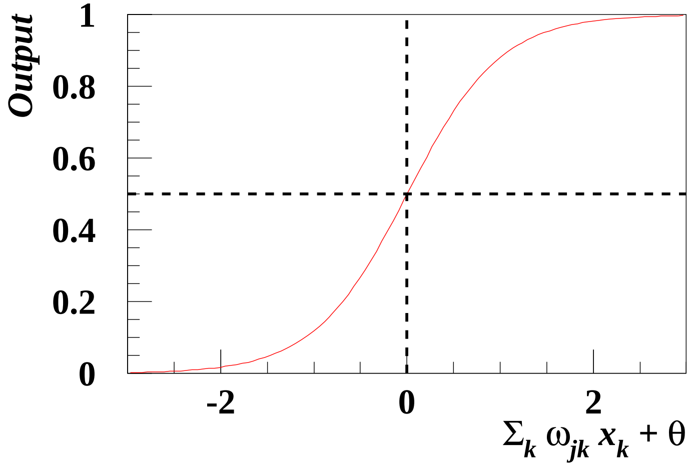


+ a threshold θ_j for each neuron

Output of a neuron:

$$O_j = g(\sum_k \omega_{jk}\ x_k + \theta_j)$$

with $g(x) = \dfrac{1}{1+e^{-2x}}$



Training: adjust weights
→ NN_{out} el. ∼ 1 and had. ∼ 0