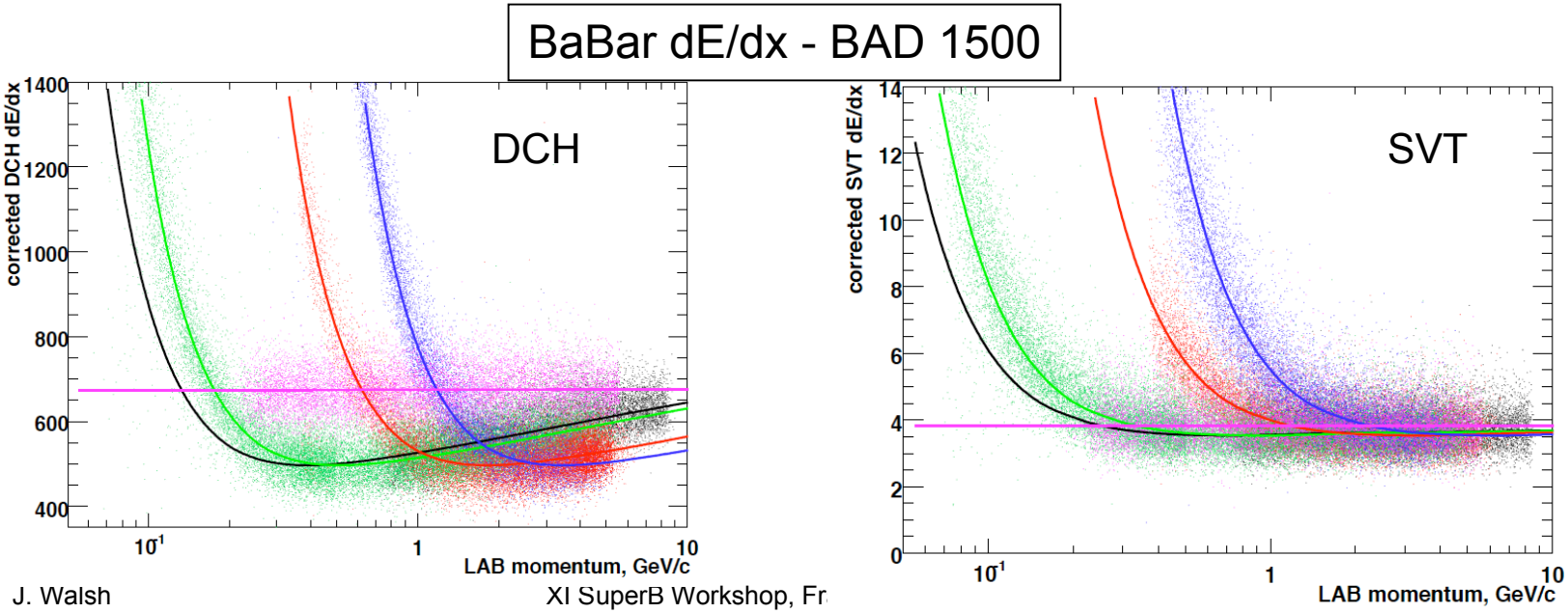
SVT dE/dx in FastSim

John Walsh
INFN, Pisa

XI SuperB General Meeting
Frascati, December 2009

SVT dE/dx

- Clearly, DCH and DIRC are most important systems for particle ID
- However, SVT dE/dx measurements can be important, especially for low-p_t tracks (e.g. soft pions from D* decays)
- SVT dE/dx measurements have been used in BaBar PID algorithms
- It should be in FastSim



SuperB SVT dE/dx

- We expect SVT dE/dx performance in SuperB to be similar to that of BaBar, perhaps slightly better
  - Layer 0: depending on the technology used, we may have some dE/dx information (no for hyrbrid pixels, yes for striplets)
  - Layers 1-5 will have new readout electronics. We're not yet able to predict the dE/dx performance for the proposed electronics:
    - BaBar: ATOM chip: Time Over Threshold (logarithmic) determination of charge
    - SuperB: FSSR2 chip: only 3 bits of analog charge readout, although the 8 available values for deposited charge can be set independently
  - Use BaBar performance as a reasonable baseline for Fastsim.
- BaBar dE/dx resolution: ~16% for MIPs (at track level)

Fastsim Implementation

- Follow work done for DCH by Matteo Rama
- For each **hit** (actually, track/detector intersection) calculate dE/dx value
  1. calc. average expected value from Bethe-Bloch
  2. calc. uncertainty on average (see following slide)
  3. use random Gaussian generator to calculate actual value of hit dE/dx
  4. if dE/dx < threshold, go to 3. (more on this in a bit)
- Note: Gaussian distribution is not quite right, but good enough and it makes life simpler

Uncertainty on hit dE/dx

- Parameterize, as for DCH, as:

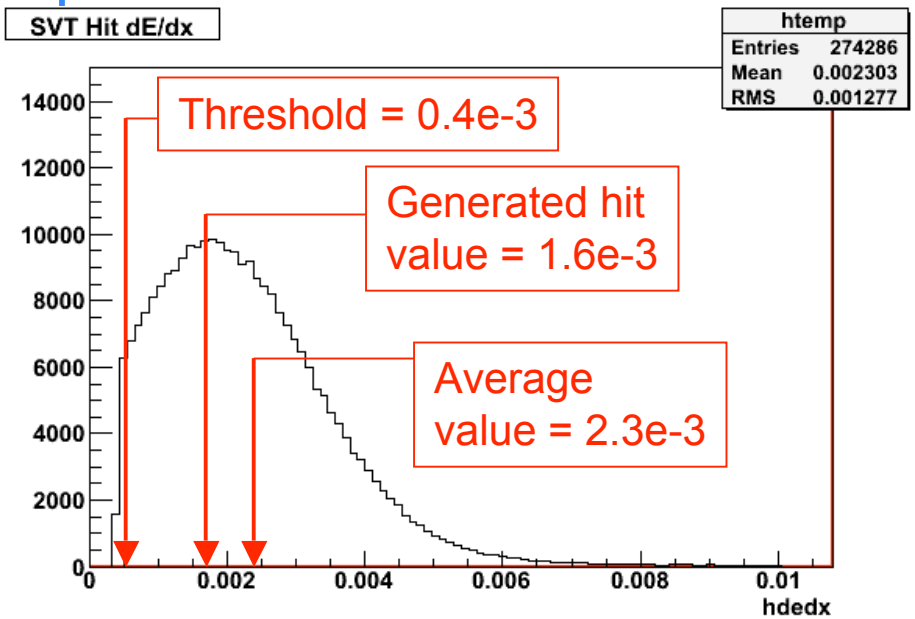
$$\sigma_{dE/dx} = p_1 \left[\frac{\langle dE/dx \rangle}{1.622 \times 10^{-3}} \right]^{p2} dx^{p3}$$

dE/dx of MIP

- We then set:
  - p_2 = 1 and p_3 = -0.5 (BAD 1500)
  - adjust p_1 to give desired resolution

Threshold effect

- Given our method, we will sometimes generate a negative value of hit dE/dx
- However, in real life, any hit with deposited charge below some threshold will not be registered as a hit.



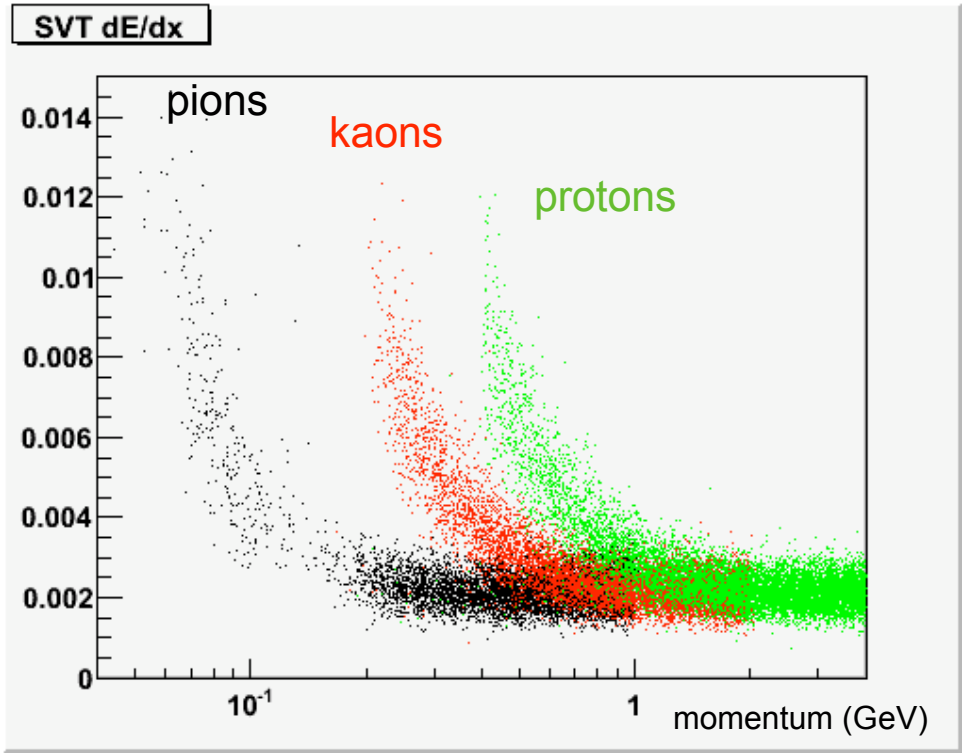
- Generated dE/dx values are required to be above some threshold that is specified in configuration.
- Typical threshold=MIP/4
- This results in <dE/dx> being higher than Bethe-Bloch value due to truncation

Track dE/dx

- The dE/dx of a track is simply the truncated mean of the values of hit dE/dx for that track
- The truncation fraction is specified as a configuration parameter -- default currently = 0.
- The averaging algorithm does not take into account correlations between the ϕ- and z-sides
  - this leads to an underestimate of the track dE/dx uncertainty
- The code can easily accommodate any averaging algorithm that we want to write

Tuning SVT dE/dx

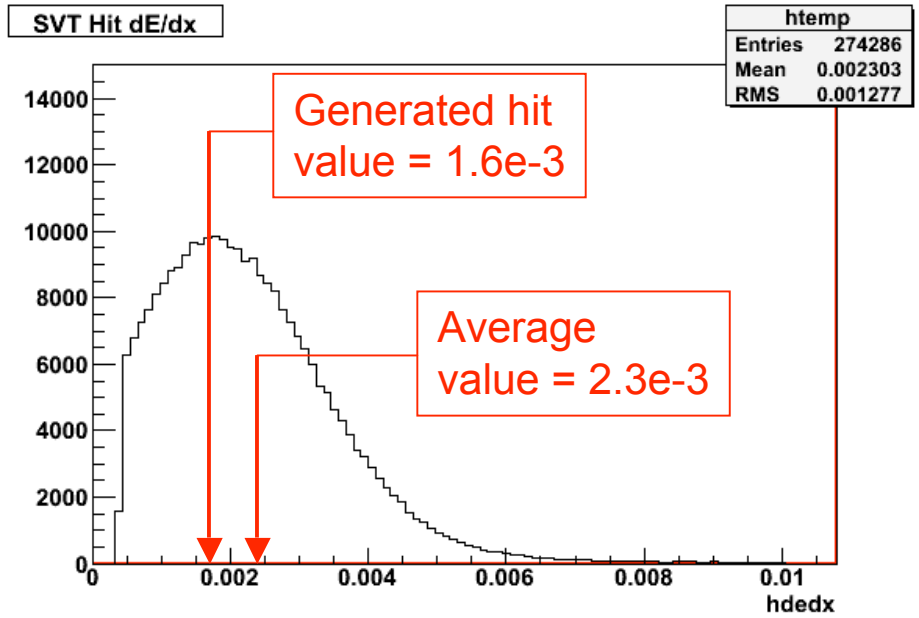
- Only crude tuning of dE/dx to BaBar performance has been done
  - probably good enough, though, given our currently level of knowledge about the SuperB detector
- Procedure:
  - adjust hit uncertainty parameter p_1 to give desired track dE/dx resolution of 16% for minimum ionizing pions



Making SVT dE/dx useful for PID selectors

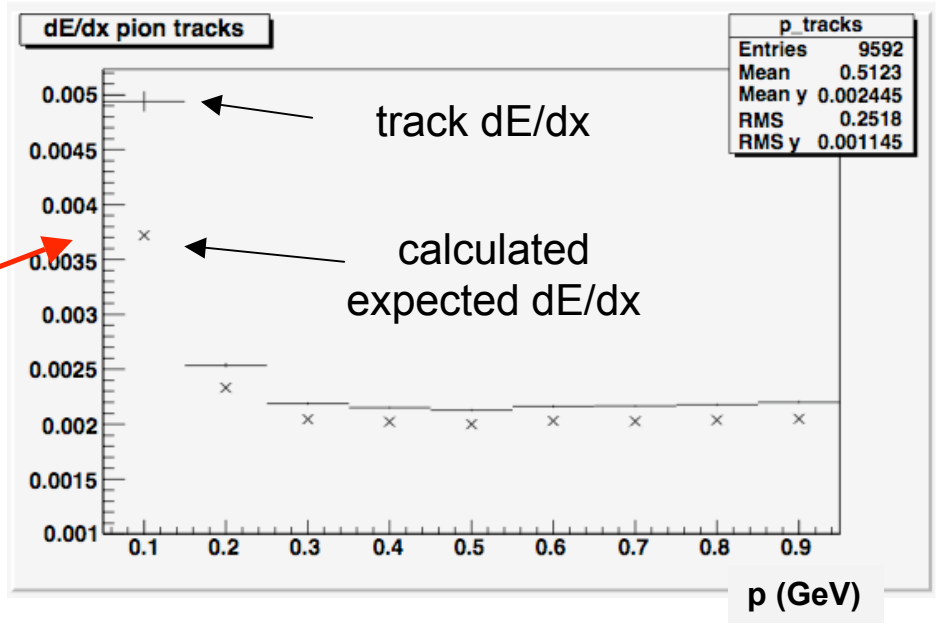
- Generating SVT dE/dx values is only the first step in using it for PID
- PID selectors typically need three pieces of information for a given track:
  1) the value of dE/dx
  2) the expected value of dE/dx
  3) the error on dE/dx
- I have described 1) and 3), but need to talk about 2).

First implementation used generated hit dE/dx value for expected track value, but we already saw that this doesn't work.



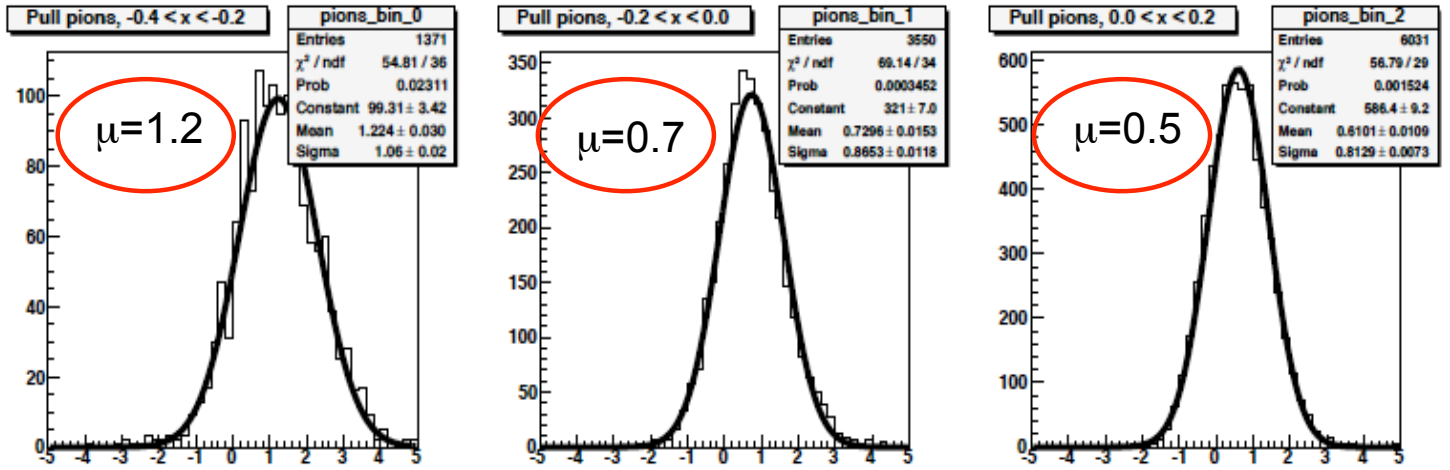
Getting expected dE/dx

- Also seen at the track level, where track dE/dx and expected value don't agree
- Need something better



dE/dx Pulls

$x = \log_{10}(\beta\gamma)$

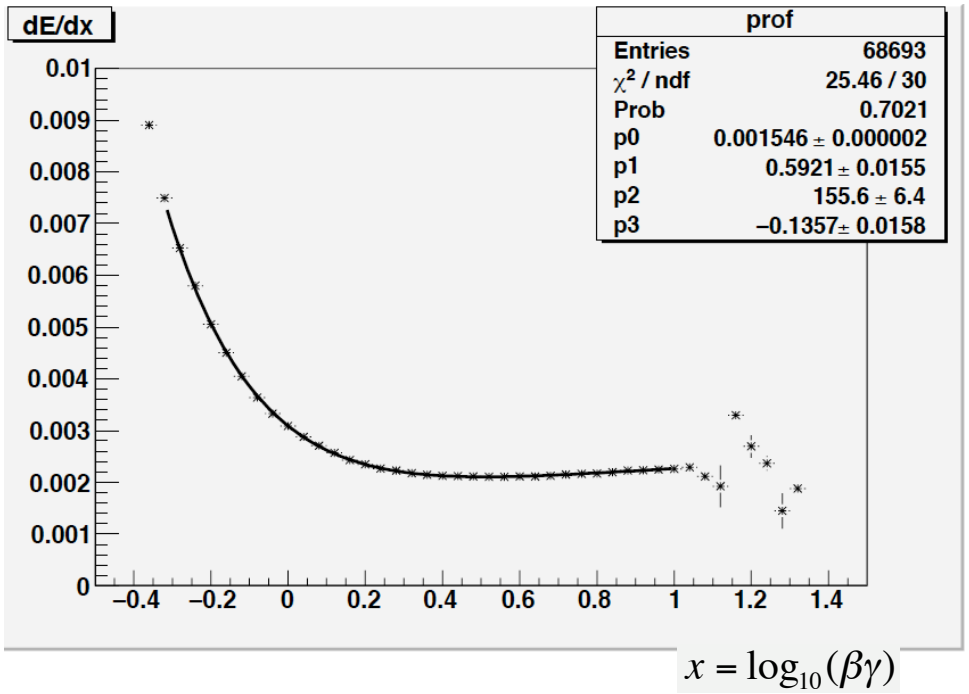


Calibrating SVT dE/dx

- Need to calibrate dE/dx, i.e., parameterize expected dE/dx by fitting track dE/dx directly
- Use BAD1500 as a guide, but use simpler 4-parameter fit function:

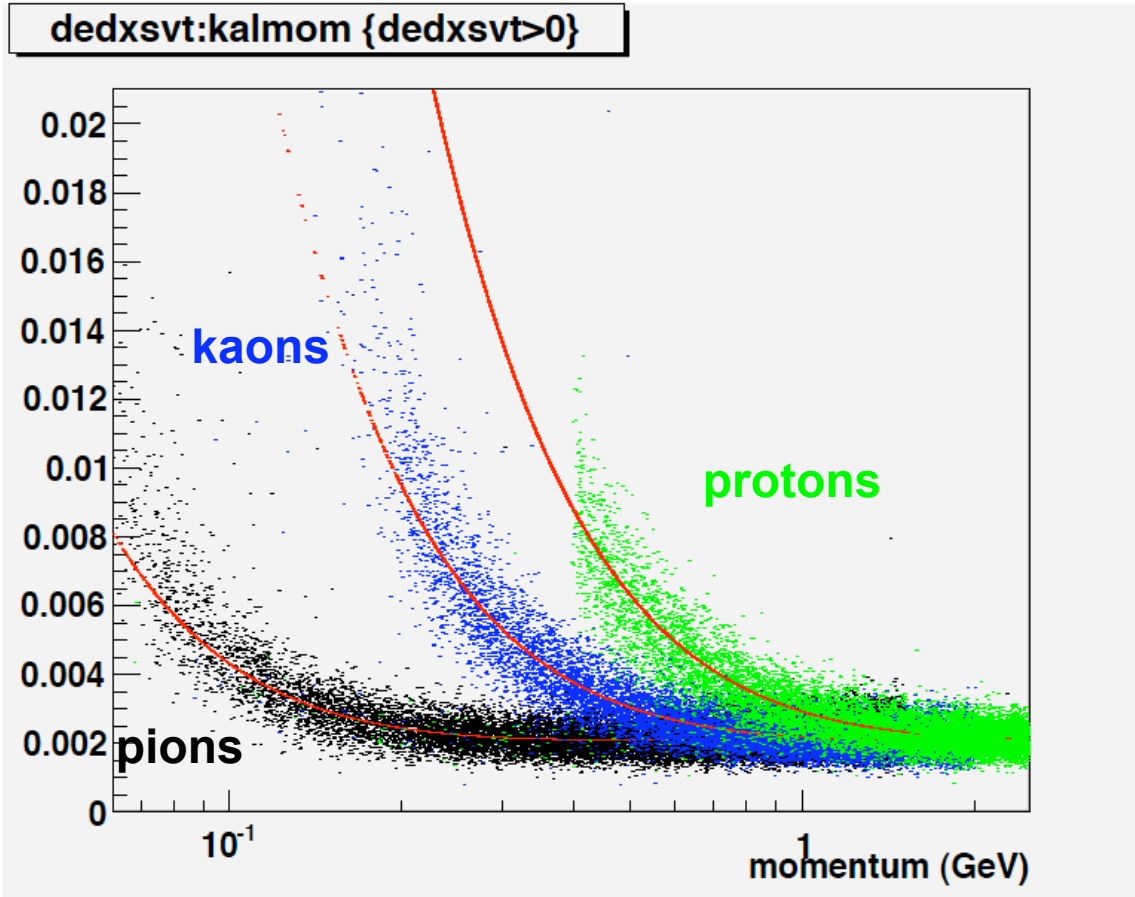
$$\left.\frac{dE}{dx}\right|_{\text{exp}} = p_0(1 + p_1 x + p_3 x^2)(1 + p_2^{-x})$$

- For x>2 (in BaBar, only electrons), expected dE/dx is constant (0.00243)

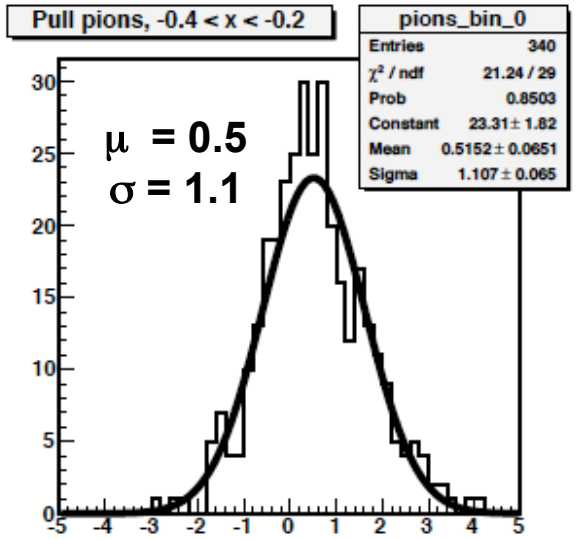


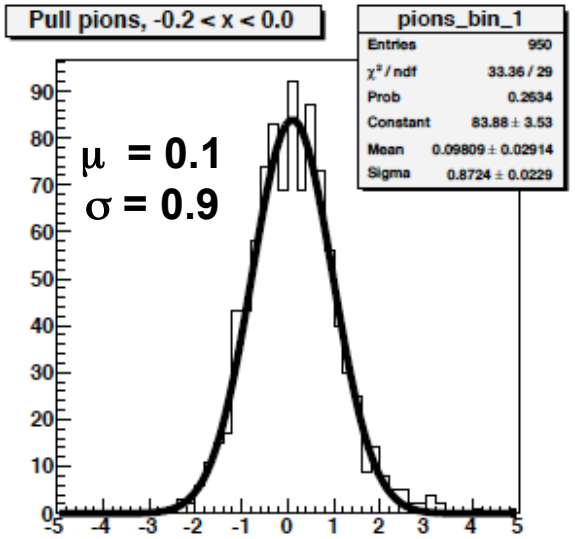
Calibrated dE/dx

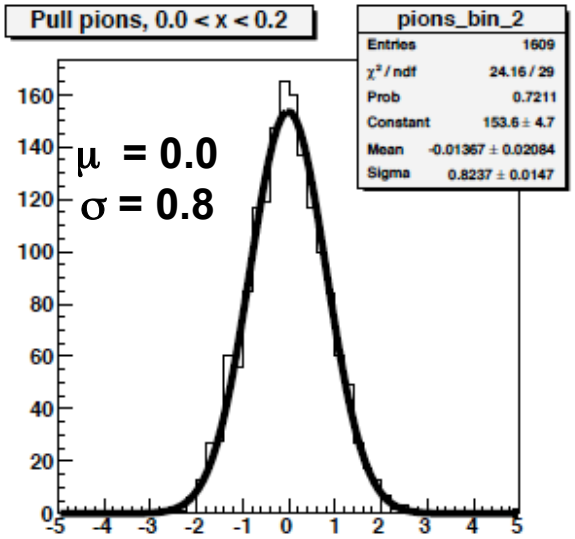
- Results look qualitatively pretty good

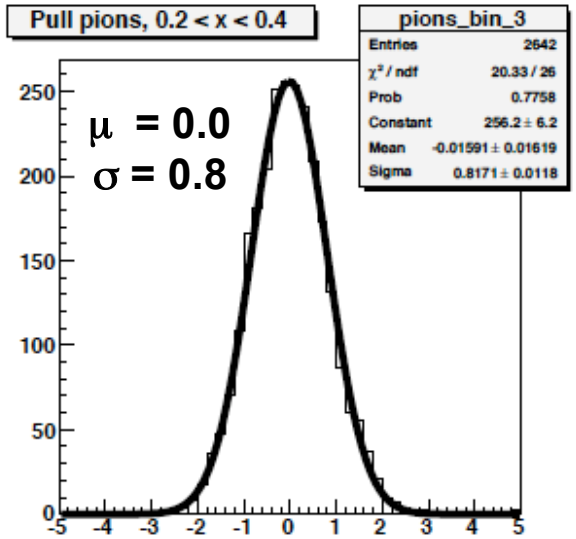


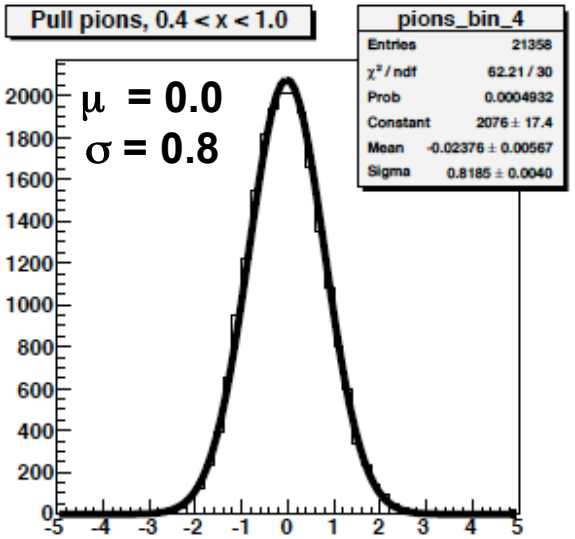
dE/dx Pulls - pions



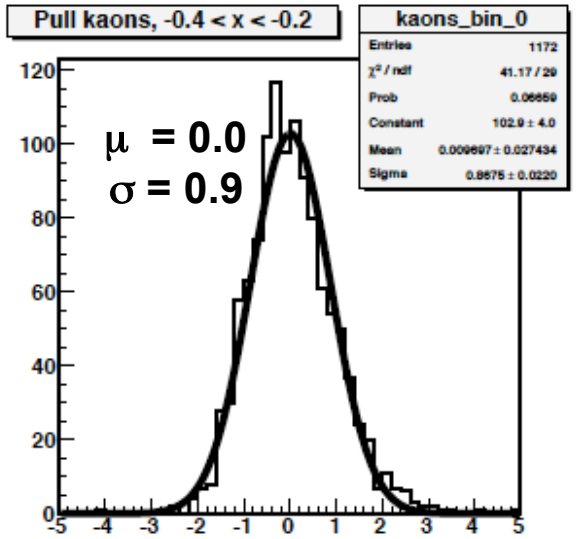


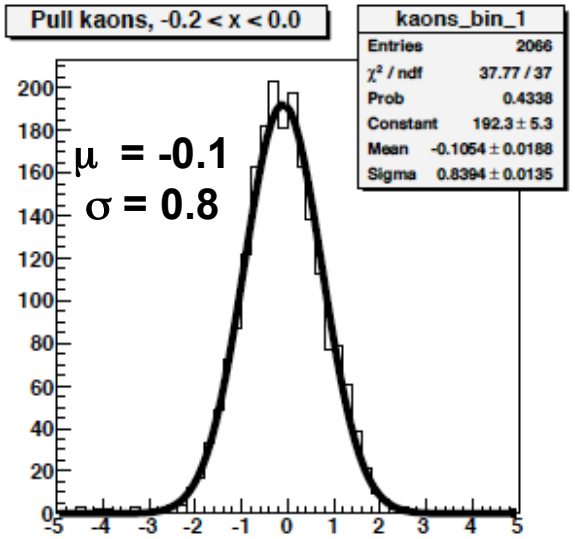


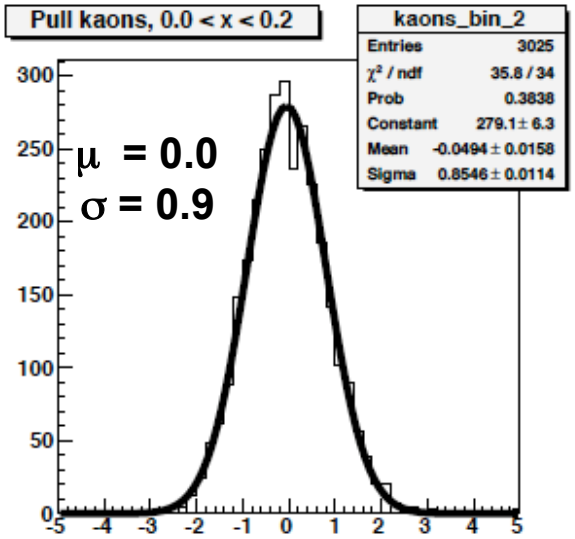


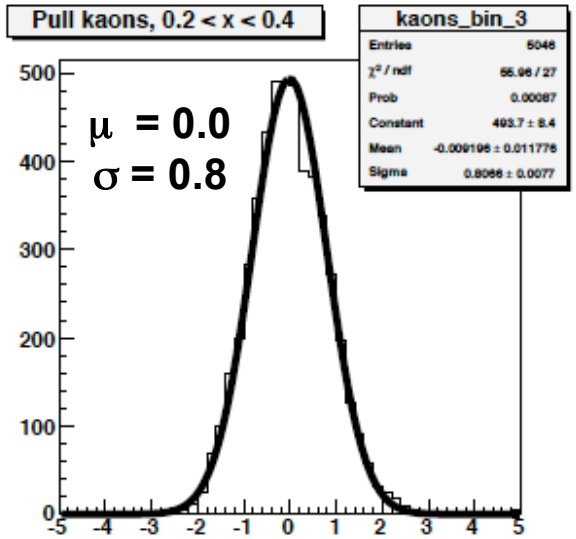


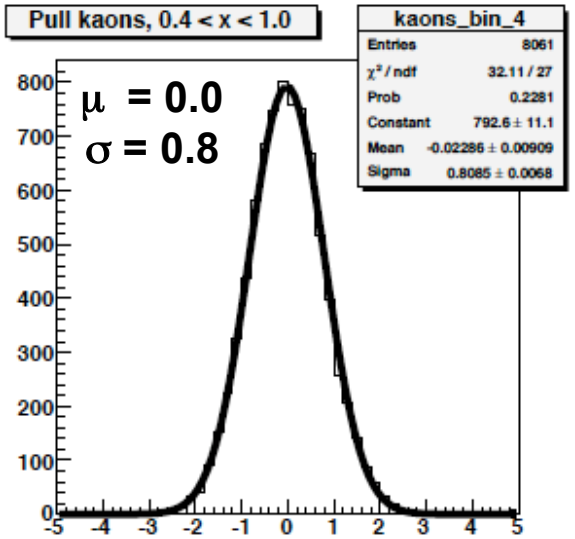
dE/dx Pulls - kaons



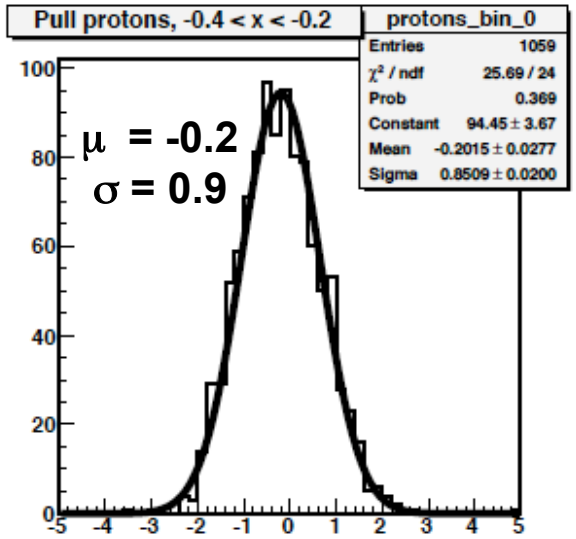


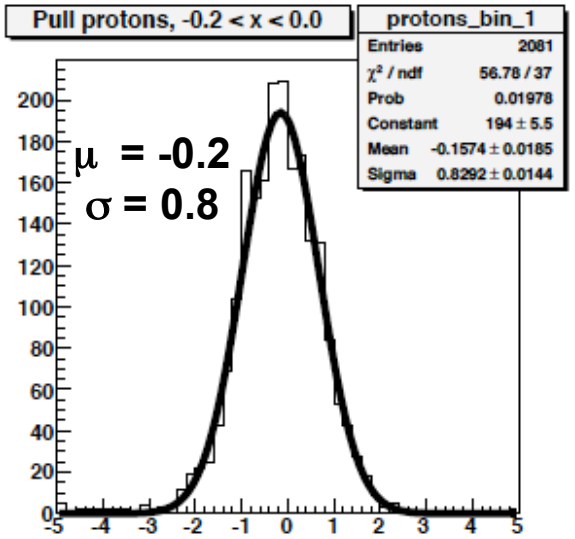


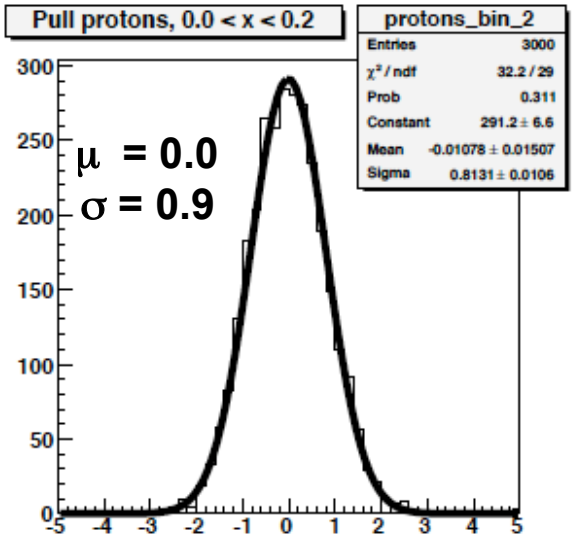


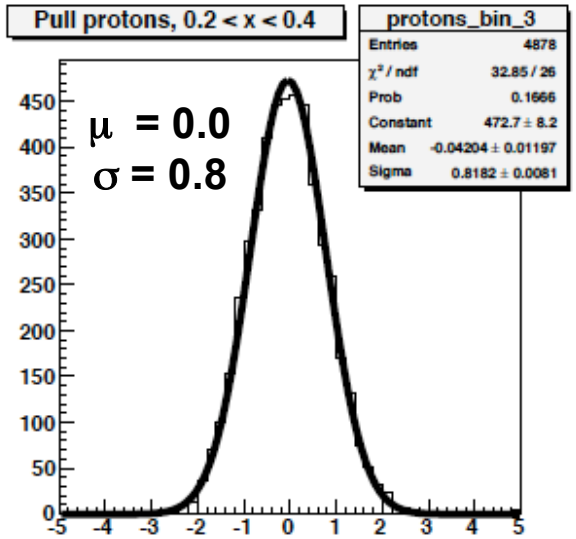


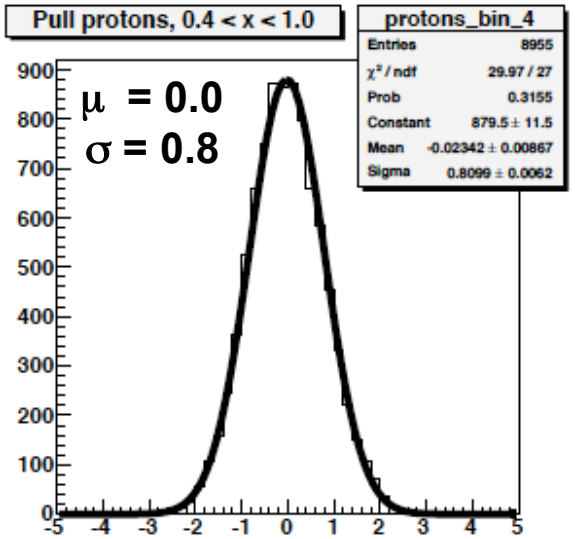
dE/dx Pulls - protons









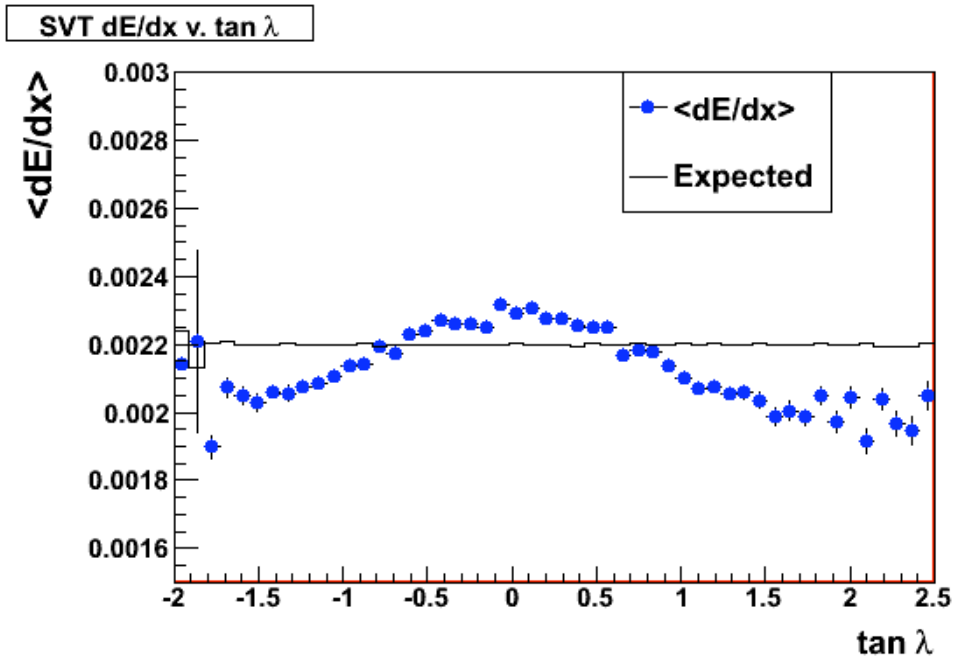


General comments on pulls:

- Most means are close to zero, largest deviation is 0.5.
- Most sigma's are around 0.8, meaning error is underestimated.

Angular dependence

- Calibration was done globally, no attempt to break down by ϕ and/or θ
- However, there is an important θ-dependence (many thanks to Leonid Burmistrov for looking at this):

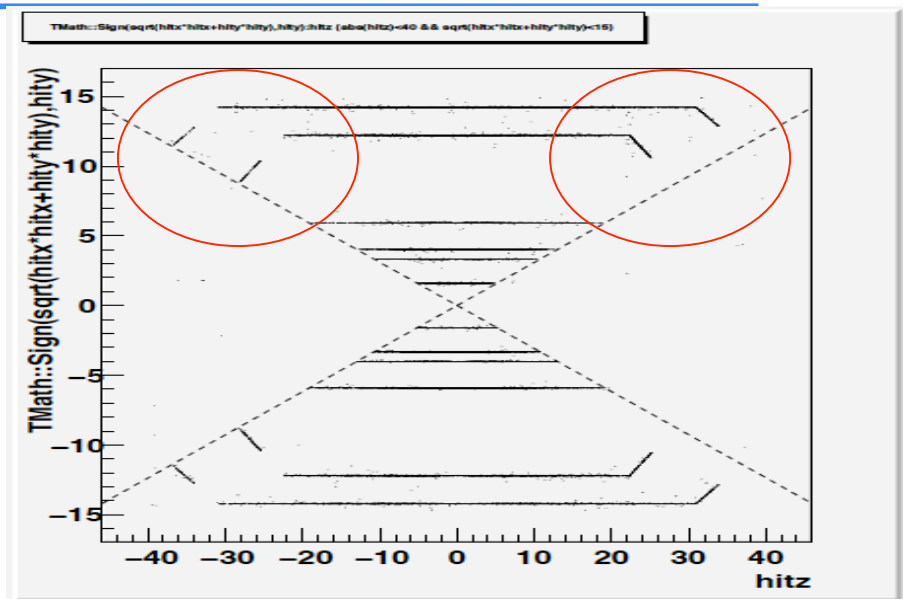


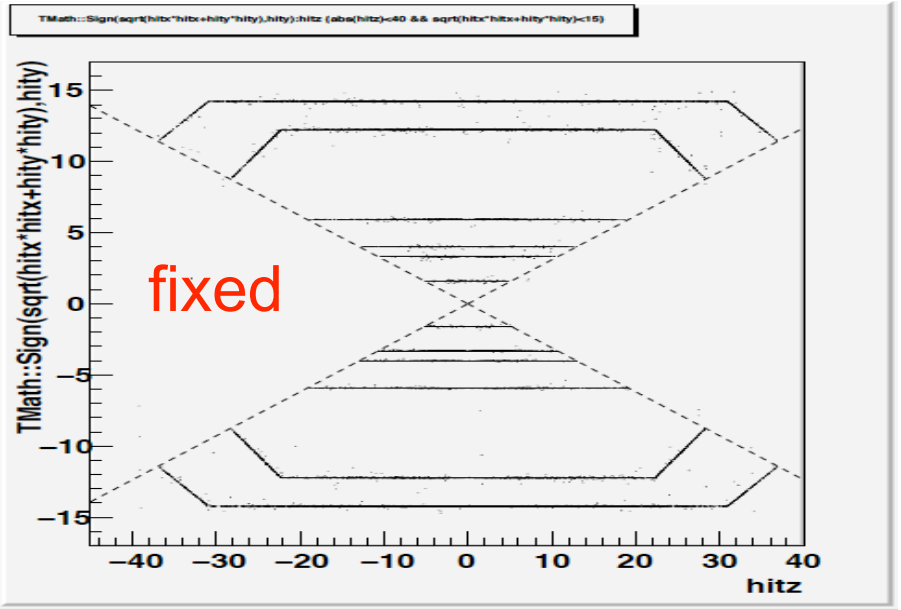
pions with p>500 MeV (MIPs)

- Need to take this variation into account in calibration

Miscellaneous SVT work

- In context of detector studies, new SVT configurations developed:
  - L0 striplets (Nicola Neri)
  - 4- and 5-layer geometries
- Currently private code, but could be committed
- Working on validation scripts, macros. Should be able to commit something soon.
- Bug fix in nominal SuperB geometry:





Summary

- SVT dE/dx has been implemented in Fastsim
- For SuperB, performance similar to BaBar is assumed: 16% resolution for MIPs
- Performed global calibration fitting a 4-parameter empirical “Bethe-Bloch” function to <dE/dx> of tracks as a function of $x = \log_{10}(\beta\gamma)$
- Calibration works on average but does not capture θ-dependence of <dE/dx>
- Need design for storing many dE/dx parameters to allow for (e.g.) θ-dependent calibration.
  - Not sure that the config files are the place for such parameters

Backup slides

Double-sided silicon devices

- Charge deposited by a track passing through the silicon is measured twice -- on the ϕ-side and on the z-side.
- Two sources of fluctuations, then:
  1. actual charge deposited fluctuates according to the usual Landau distribution -- **common to ϕ- and z-sides**
  2. additional fluctuations originating in the readout -- potentially **different for the ϕ- and z-sides**.

Different from DCH

- The code has been designed to allow for both types, although only the first type has actually been implemented
  - ϕ- and z-side dE/dx have the same value, for now.
  - adding the additional side-dependent smearing is straightforward

Fastsim code (I)

- Started with Matteo's design, but made some modifications to accommodate two SVT views for a single charge deposit
- `PacTrkdEdxMeas` - new class that generates the hit dE/dx values for both DCH and SVT.
- `PacTrkHitMeas` - now optionally takes `PacTrkdEdxMeas*` argument in c'tor. Non-zero for DCH and Silicon strips
- `PacTrkHitViewSvt` - will handle ϕ- and z-side independent smearing (when implemented)
- `PacMicroAdapter` - adds SVT dE/dx info to PidQual object
- `BtaPidQual::dEdXSvt()` (and related functions) provides SVT dE/dx to PID algorithms
- `Si_Measures_baseline.xml` - contains dE/dx parameters for SVT

Fastsim code (II)

- Added parameters to configuration, implies changes to:
  - PacTrk/Si_Measures_baseline.xml
  - PacTrk/Si_BaBar_Measures.xml
  - PacDetector/PacMeasurementFactory.cc
  - PacEnv/EdmlParser.cc
- Average dE/dx function calculated in PacTrk/PacTrkdEdxMeas::getExpectedTrackdEdx(momentum,mass)
  - this class has new parameters (5 of them) as data members
  - PacTrk/PacTruncMean::getExpectedTrkDedxSvt(PdtPid::PidType pidType) supplies expected dE/dx to higher level code
- For example: PacMC/PacMicroAdapter.cc uses PacTruncMean to get all DCH and SVT dE/dx information and pass it into BtaPidQual object
- PacPid selectors should access all SVT dE/dx info from the PidQual object. These are the relevant functions:
  - dEdXSvt() - returns track dE/dx value
  - errdEdXSvt() - returns error on dE/dx value
  - nSamplesDeDxSvt() - number of samples used to calculate track average
  - dEdx{Ele,Mu,Pi,K,P}Svt() - expected track dE/dx for particle hypothesis
- To use, you need the trunk of PacTrk, PacDetector, PacEnv and PacMC on top of patched version of V0.1.1.