Fast Simulation: EMC Status and Plan

Chih-hsiang Cheng, Caltech

SuperB workshop VII, Elba, Italy, May 31-June 3, 2008

Abstract

- SuperB will reuse the BaBar EMC Barrel (cylinder).
- Forward endcap will be using new crystals, but the overall geometry (cone) will be the same (at least very similar).
- There may be a backward calorimeter (possibly a disk).
- PacTrk provides true simulated tracks to (and through) the EMC.
- Detector response will be parameterized/sampled from a shower library, which can be created from BaBar and GEANT4 full simulation, based on incident particle's species, momentum, and entry angle, etc.
- FastSim creates reconstructed objects, and then builds BtaCandidates.

Data representation

- Original idea was to compile a library of statistical properties (energy, #crystals, various moments, etc.) and sample from it. However, the distributions and (especially) the correlations are not trivial. => Hard to sample correctly. Plus, the numbers need to be stored may exceed the number of crystals.
- New idea: create a simplified EmcCluster class to store crystal indices and energy deposits. All the moments and such can be calculated easily (code available in BaBar). We can even merge clusters.
- Detector geometry is segmented. <<= configuration file.

What needs to be done in PacTrk

- Add cone and disk geometries.
- Create simple EmcCluster class.
- Create line trajectory for neutral particles (PacSimulate).
- Define interaction for showering (PacSimulate).
- Create a reconstruction class (~PacReconstructTrk) to return an EmcCluster based on a PacSimTrack, looking up from a library.
- ... ...
- Some of the above may be in a separate package.

What needs to be done in PravdaMC

- Add a class analogous to PmcOpPacReco that uses PacReconstructXX to create clusters and calculate parameters for BtaCandidates.
- Expand PmcOpPacReco to allow cluster creation too.
- Expand PmcReconstruct (that uses PmcOpXXX) to make BtaCandidate lists for neutrals too.
- ... ...

Status of coding

- No much has happened yet.
- Most time spent on how to create a shower library (more later).
- Now I believe I have sufficient understanding of PacTrk to implement much of what is said.

How to make a shower library

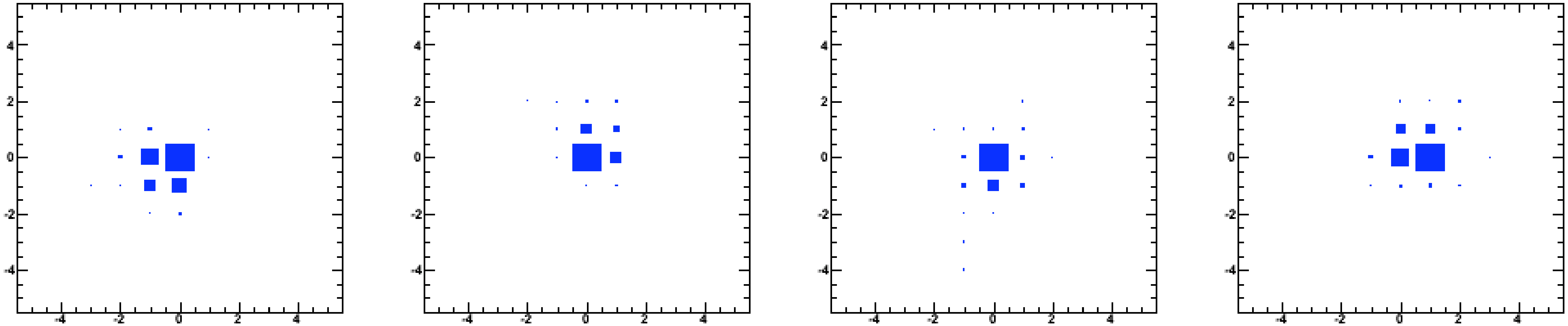
- One needs: true particle type, momentum and angle at the entry, the intersection with the detector, and the associated cluster.
- Using BaBar software, it is easy to get those truths for neutrals but virtually impossible for charged particles because only GTrack at the origin of the particle is stored.
- Solution: generate single particles right at the front face of the detector, so all properties are well controlled.
  - May need small tweaking of BaBar code, but should be simple.
- First generate an ntuple, where for each particle lists of crystals' coordinates (relative to the one that got hit) and deposited energies.

What goes to the library

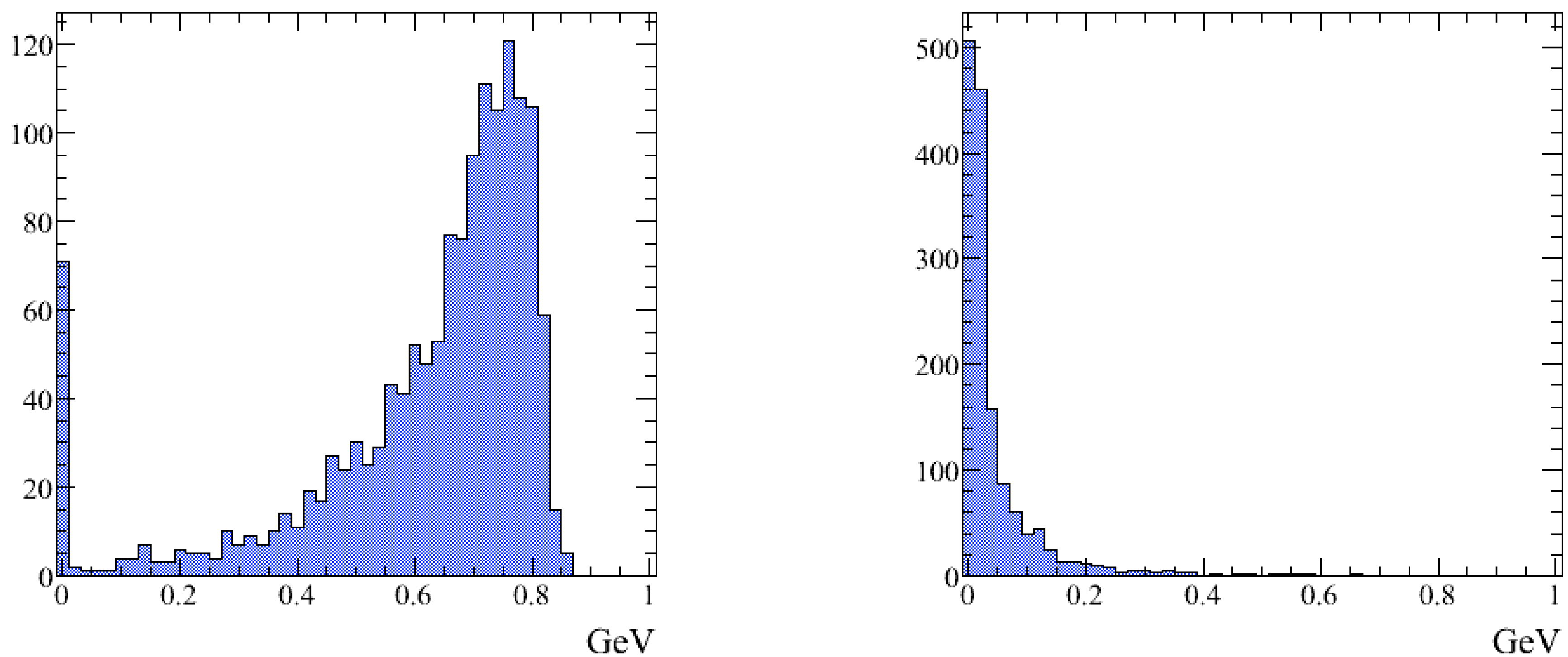
- We can store a 3D histogram (x, y, energy) for each species in each bin of momentum, entry angle, etc.
- When generating, we pick up the corresponding 3D histogram and for each crystal (x,y) we sample the energy from the 1D histogram.

1GeV photon example

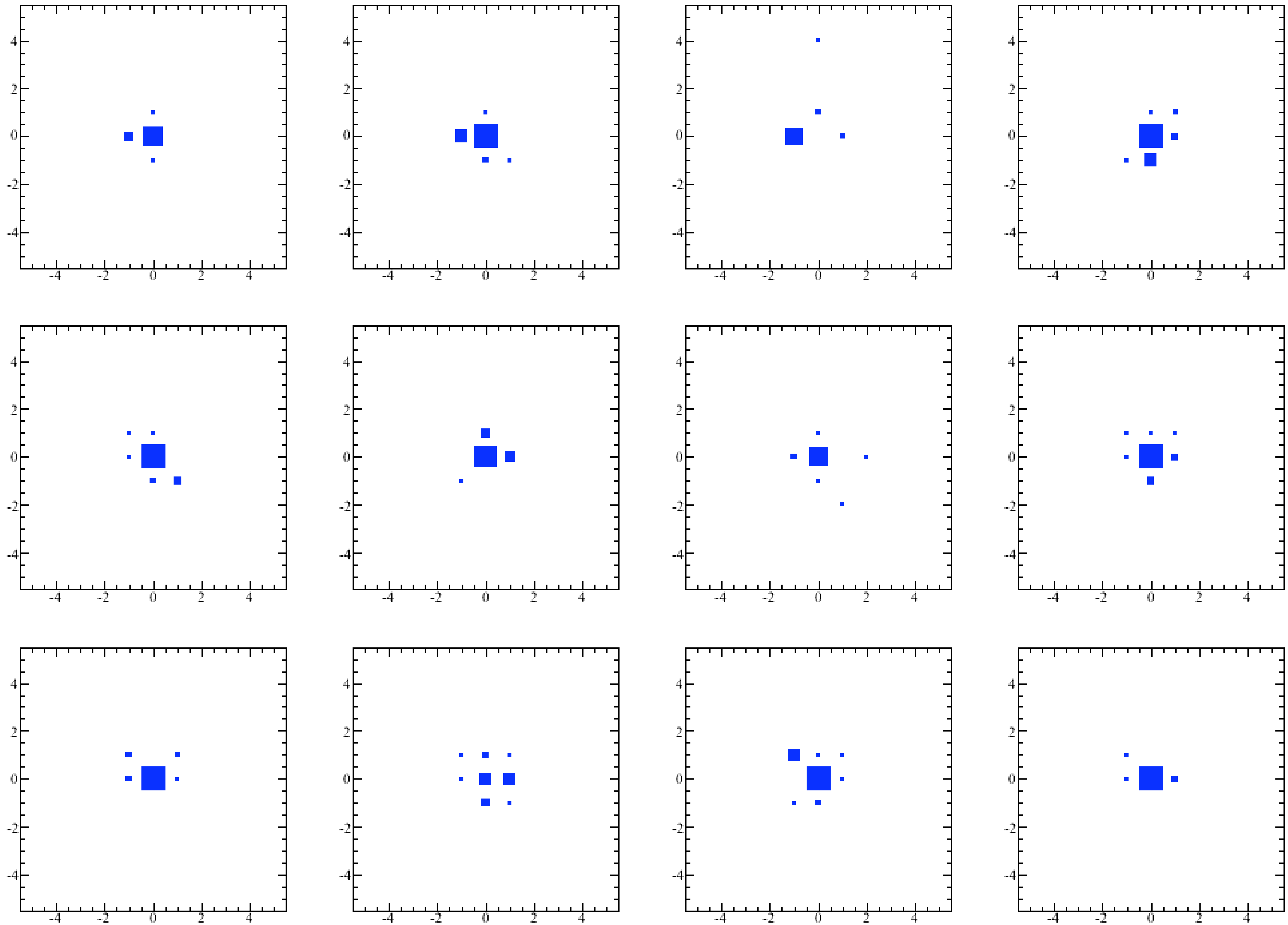
- Samples of shower shape



- Energy profiles of central crystal and one next to it.

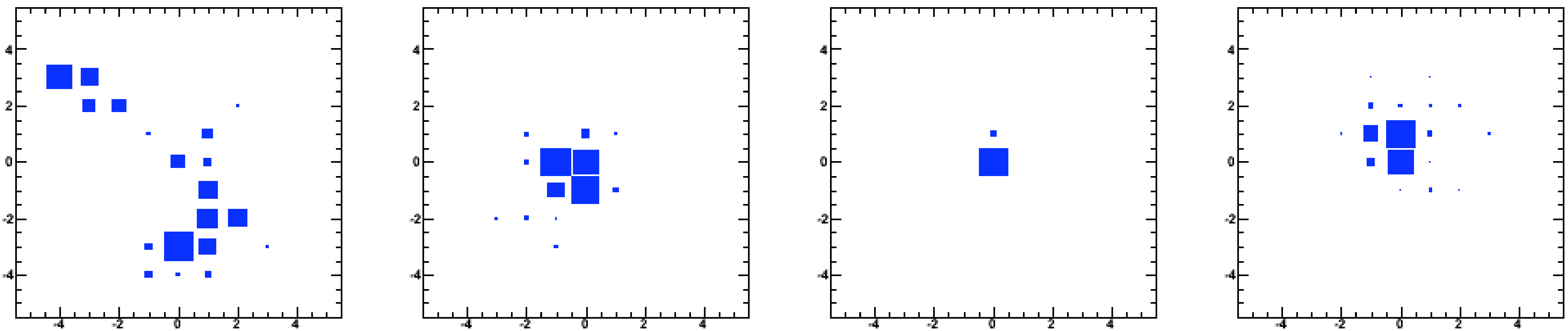


Generated showers, look more or less OK

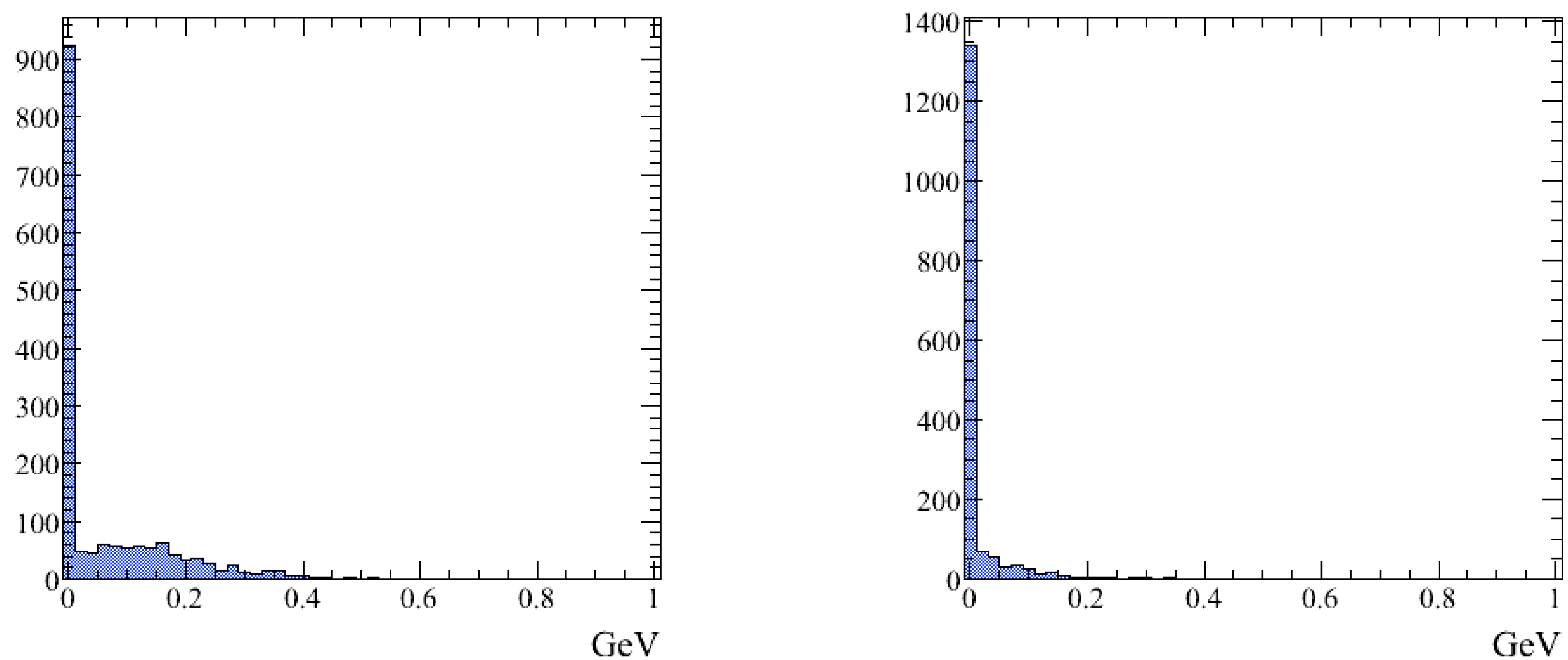


1GeV K_L example

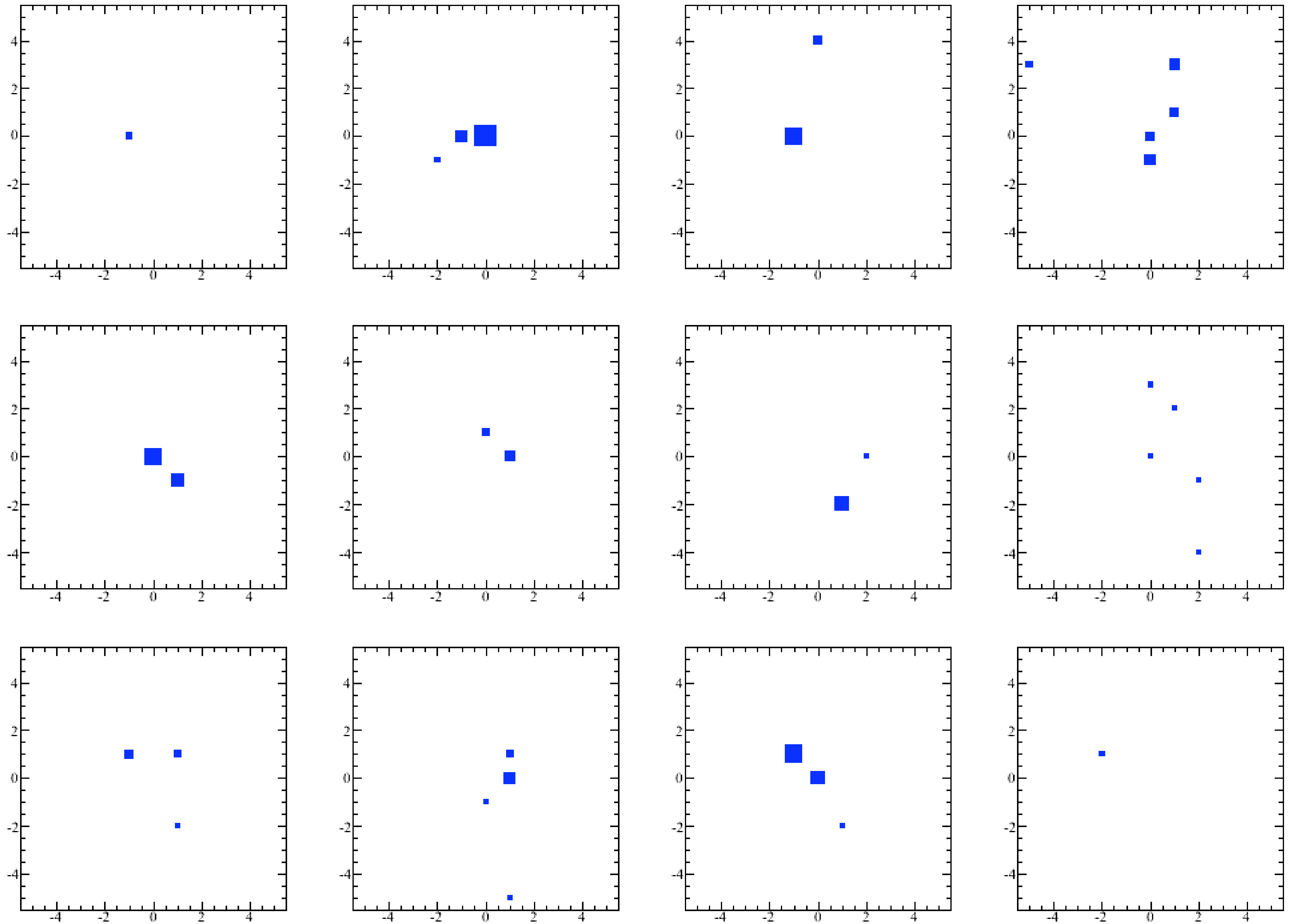
- Samples of shower shape



- Energy profiles of central crystal and one next to it.



Generated shows look BAD



Can we store actual samples of showers

- Store many actual samples of showers for each species and each bin of momentum, entry angle, etc., and sample from them.
- How large would the library be?
  - Say, 15 crystals/shower (15 floats + 15 x 2 integers = 15x(32 +8x2) bits= 720 bits) ; 100 p-bins; 5 charged particles X 2 + n + n-bar + γ + KL = 14 species; 10 angle bins; 1000 samples/ bin. ==> Total= 1.26 GB.
  - Can combine species; reduce p-bins for not e or γ; angle and p are highly correlated, total bins can be reduced.
- Only a limited number of samples. Can we somehow “fluctuate” the shower? (Do we have to?)
- Speed of accessing the library?

Conclusions

- Better understanding of PacTrk and more concrete ideas of implementation.
- Understand the limitation of BaBar software in terms of creating a shower library. Believe we have a solution.