A crystal photon radiator code as event generator for GEANT4

O. Dadoun, LAL Orsay

IWLC2010 Geneva, October 2010

dadoun@lal.in2p3.fr

G. Lemeur, F. Touze, F. Poirier & A. Variola (LAL, France)

X. Artru, R. Chehab & M. Chevalier (IPNL, France)

L. Rinolfi (CERN)

V. Strakhovenko (BINP)

Outlook

- Fot : a simulation code for channeling radiation
- G4Fot : Fot associated to Fot simulation
- Results
  - Comparison with Strakhovenko's simulation for different incident e- beam energy
  - Crystal and amorphous gamma yield comparison @ 8GeV
  - Hybrid source results
    - KEKb Case
    - CLIC case
- Conclusion & prospects

Foreword

- CLIC positron production baseline

  Hybrid scheme : association of a thin crystal and an amorphous target

  - F. Poirier's talk

- Hybrid scheme are under study at KEK

  - T. Takahashi's talk

- The crystal output simulations (for both charged particles and γ) have been provided by V. Strakhovenko (BINP) for different incident e- beam energy (using optimal target crystal thickness)

  To simulate a hybrid source we need (at least) :

  1. Vladimir's simulation
  2. Geant4 (Fluka, EGS…)

Fot is an other crystal simulation developed by X. Artru (IPNL) that will be very soon available for the community
(has not been used since more that 10 years)

Fot history : NA33 CERN experiment
[Belkacem et al]

150 GeV γ or e- impinging on 185 μm of crystal germanium oriented on its <110> axis

- With γ beam, it confirmed the strong field QED mechanism of pair creation
- With the e- beam radiation was much more intense than calculated
- An unexpected peak @ (radiated energy)/(incident energy)=0.8

This peak was explained as an effect of transverse energy loss which accompanies channelling radiation
[V. Tikhomirov, X. Artru, V. Baier, V. Katkov and V. Strakhovenko]

Fot simulation

A simulation code for channeling radiation by ultra relativistic e- or e+

While other authors described the channelling radiation in strong field using the uniform field approximation, X. Artru built a MC Fortran simulation code (Fot simulation) using the Baier-Katkov formula for synchrotron radiation in non-uniform field [X. Artru, NIM B48(1990)]

The code has been used to :

1. Present a proposition of positron using channelling (PAC 1989)
2. Simulate the proof of principle experiment at Orsay (1992-93)
3. Reproduce the results of a channelling radiation experiment @ 10 GeV [E.N. Tsyganov et al(1989)]
4. Interpret another CERN experiment [Phys. Lett. B313 (1993)]

Fortran to C++ : Fot++

Important comment :

1. No pair creation generation inside the crystal
2. When the e- is un-channeling : Vt > Vtmax and/or Et > Etmax, the e- is no more simulated (ejected) even if it is still in the crystal

For different reasons we decided to switch Fortran to C++

Due to 1. & 2. Fot++ associated to Geant4 ➔ G4Fot

Fot++ written by G. Lemeur & F. Touze (and S. Berte) and has been adapted for Geant4 uses (Version0 October 2010)

G4Fot : construct your crystal

```
G4VPhysicalVolume* DetectorConstruction::Construct()
{
                                ...
   G4ThreeVector positionCrystal= G4ThreeVector(0,0,current_position);
   G4Box* solidCrystal = new G4Box("Crystal",size/2.,size/2., thickness/2.);
   G4LogicalVolume* logicCrystal =
                 new G4LogicalVolume(solidCrystal, aMaterial,"Crystal");
   new G4PVPlacement(0,positionCrystal,logicCrystal,"Crystal",lWorld,false,0);

   Crystal crysW("W",111);
   RunParameters rp(crysW);
   rp.setZexit(thickness);
                                ...
}
```

G4Fot : construct your crystal

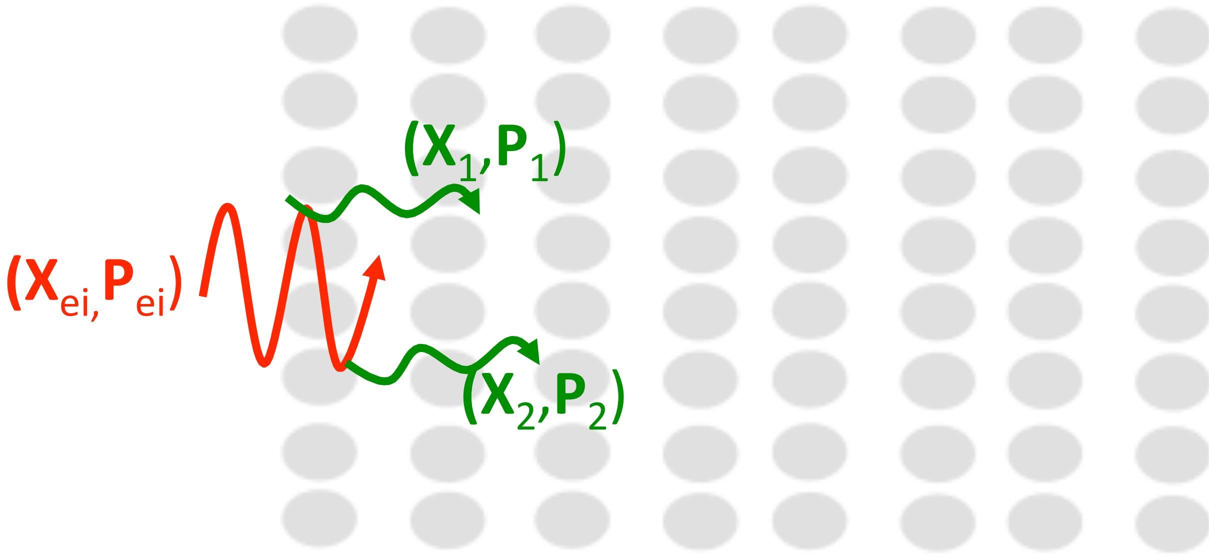
Generation of the photons (and the ejected e-) and their starting Propagation in Geant4 will be done by calling Fot++ in the Primary Generator Action

G4Fot : define your primary generator

```
void PrimaryGeneratorAction ::GeneratePrimaries(G4Event* anEvent)
{

                                        . . .

   G4ParticleDefinition* particle=
                      G4ParticleTable::GetParticleTable()->FindParticle("e-");
   particleGun->SetParticleDefinition(particle);
   particleGun->SetParticleMomentumDirection(G4ThreeVector(px,py,pz));
   particleGun->SetParticleEnergy(energy);
   particleGun->SetParticlePosition(G4ThreeVector(x,y,z));

   Fot f(rp);
   Particle* part = new Particle(charge,x,y,0.,px,py,gamma);
   ParticleInCrystal partCrys = f.makeSingleParticleKumakhov(part)
                                          . . .
```

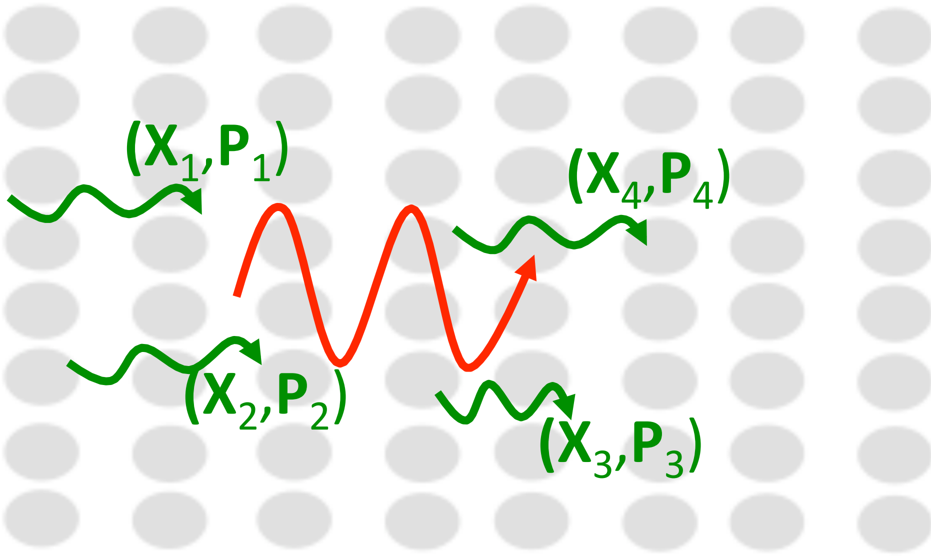
Run->BeamOn(1)

G4Fot : define your primary generator



- Kumakhov radiation
- Coherent Bremsstrahlung
- Incoherent Bremsstrahlung ...

G4Fot : defined your primary generator

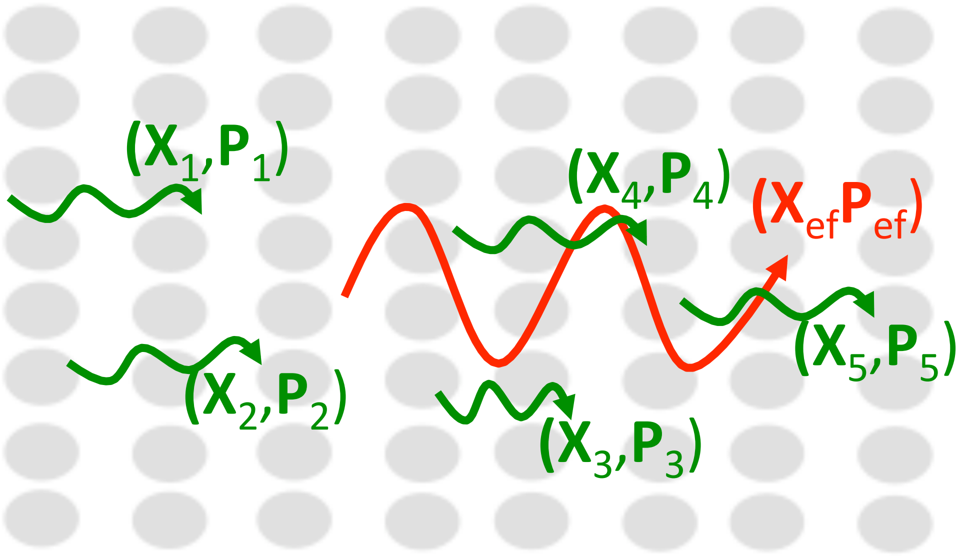


- Kumakhov radiation
- Coherent Bremsstrahlung
- Incoherent Bremsstrahlung

While Vt > Vtmax and/or Et > Etmax

G4Fot : defined your primary generator

A GeV e- beam impinge on a few mm crystal target oriented on its <1 1 1> axis



- FOT++ stop when the e- is un-channeling
- Geant4 takes the γ and e- control
  - Propagations
  - Physics processes

Photon part

```
                                   …
 G4ParticleDefinition* particle =
                 G4ParticleTable::GetParticleTable()->FindParticle("gamma");
particleGun->SetParticleDefinition(particle);
list< struct Photon >::iterator it;
for(it = list_photon_kumakov.begin() ; it != list_photon_kumakov.end() ; it++)
{
     particleGun->SetParticlePosition(position_emission_it);
     particleGun->SetParticleEnergy(GamEnergy_it);
     particleGun->SetParticleMomentumDirection(momentum_emission_it);
     particleGun->GeneratePrimaryVertex(anEvent_it);
}

                                   ….
```

Electron part

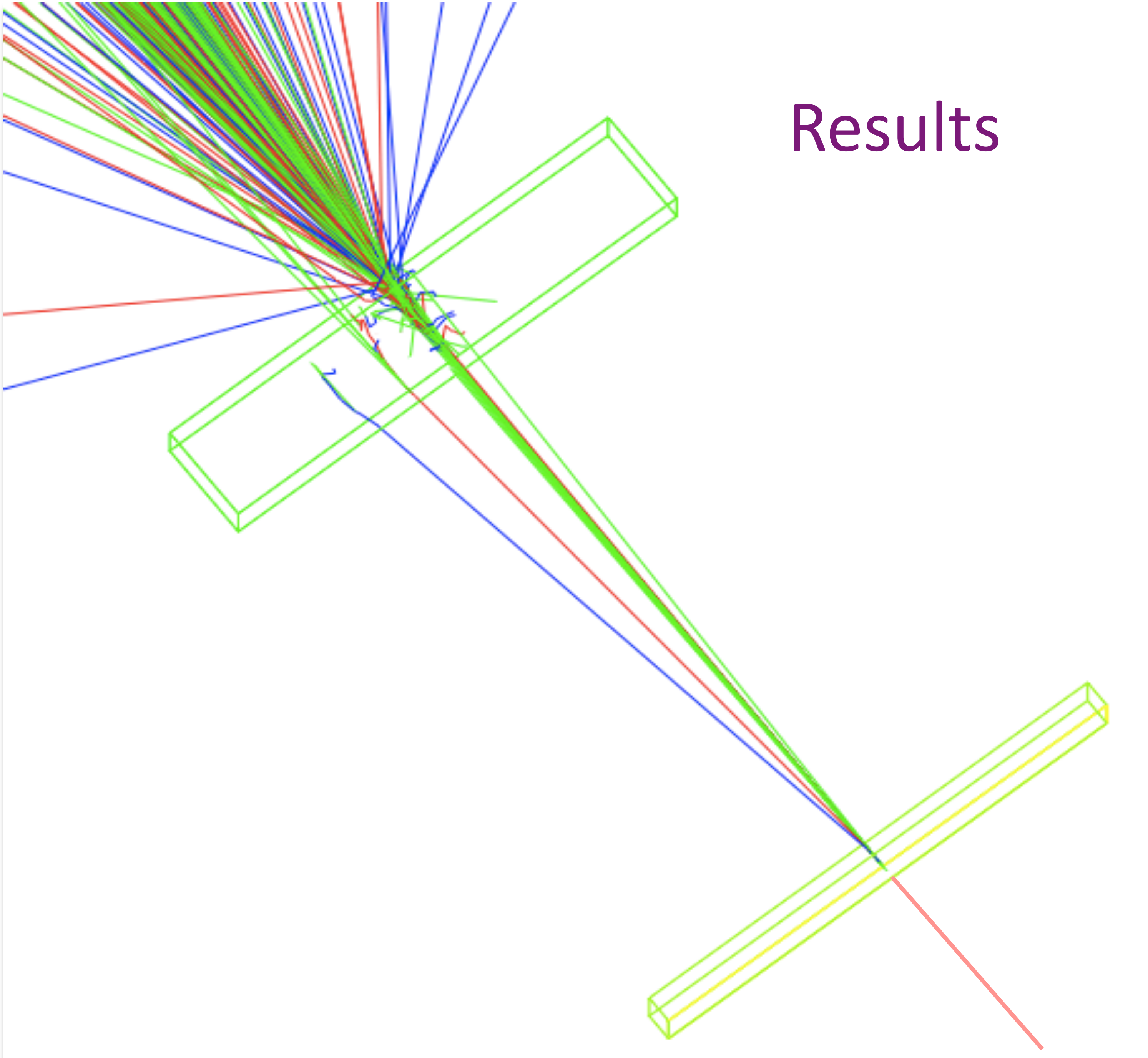
...

```
particle    = G4ParticleTable::GetParticleTable()->FindParticle("e-");
particleGun->SetParticleDefinition(particle);
particleGun->SetParticlePosition(position_emission_e);
particleGun->SetParticleEnergy(Energy_e);
particleGun->SetParticleMomentumDirection(momentum_emission_e);
particleGun->GeneratePrimaryVertex(anEvent);
}
// End of the GeneratePrimaries method
```

That's all folks ....

nothing to modify in your Geant4 code or in the Fot code

Results



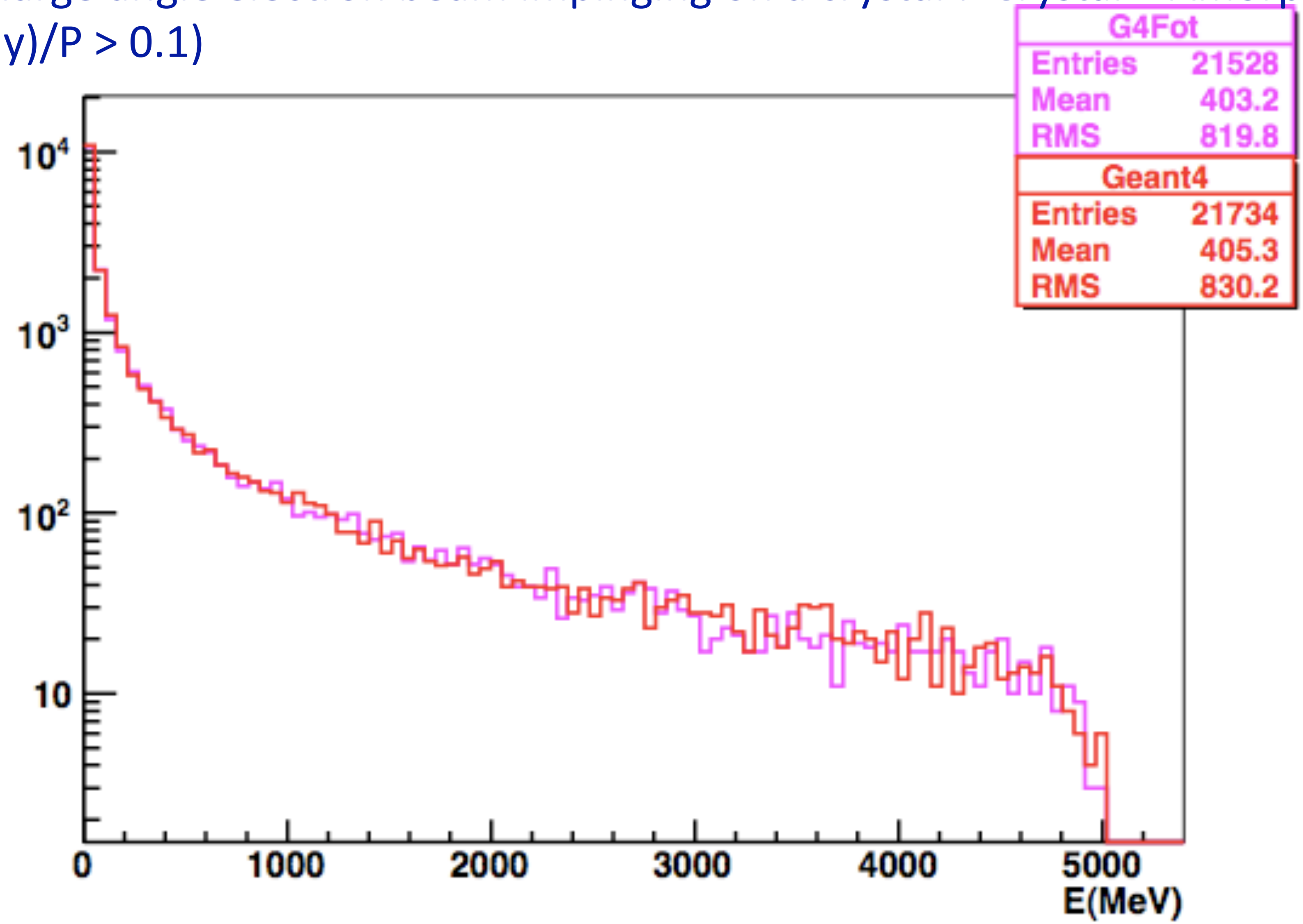
Simulations parameters

- Normal e- incidence **P**(0,0,1)
- 2 MeV energy threshold for both charged particles and gamma
- Crystal tungsten thickness for the different incident electron energy considered

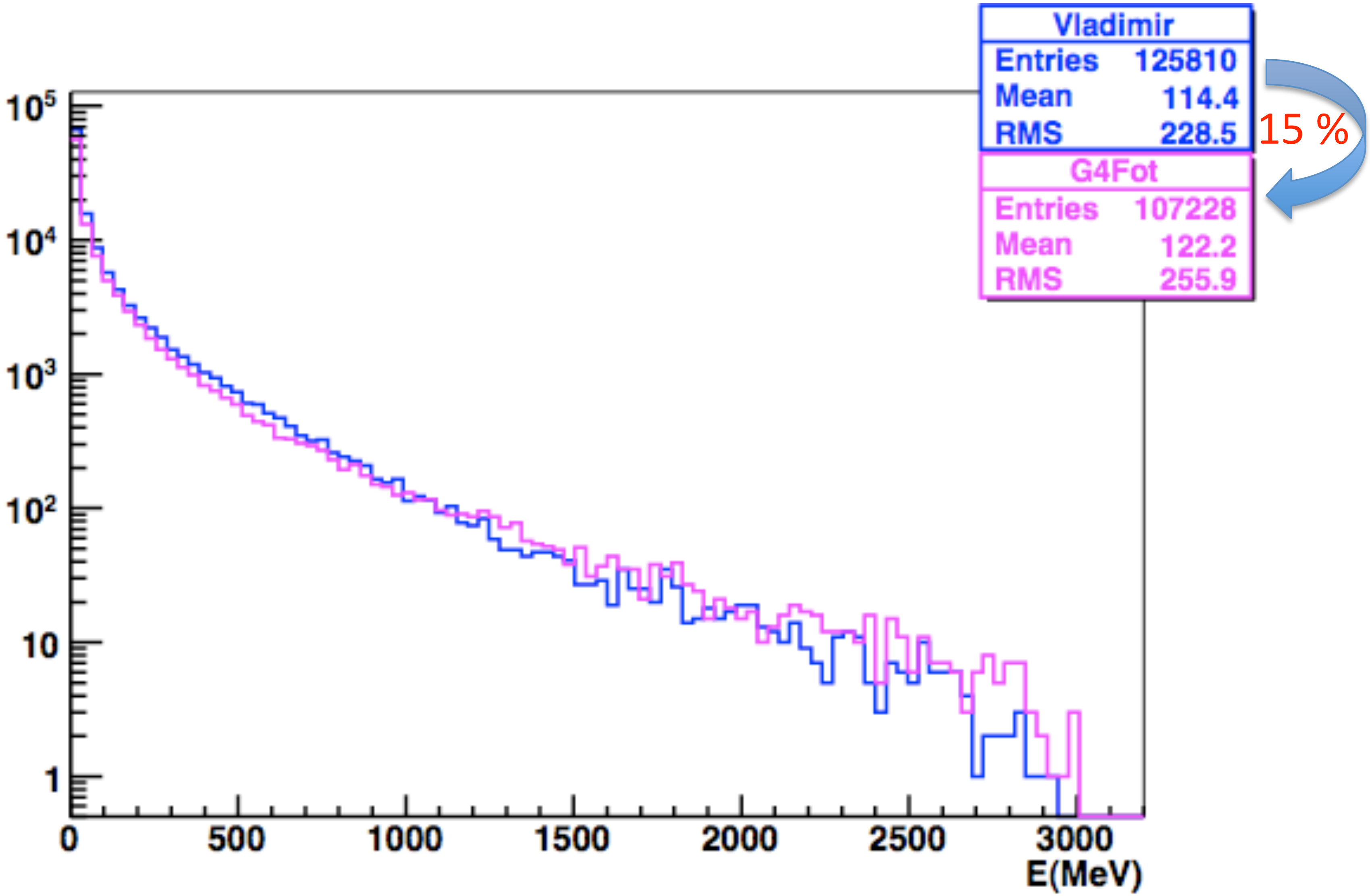
Energy (GeV)	Crystal thickness (mm)	#electrons
10	1.0	5000
5	1.4	6000
3	1.6	8000

First of all (trivial case)

- Use large angle electron beam impinging on a crystal : Crystal ~ Amorphous ($P(x,y)/P > 0.1$)



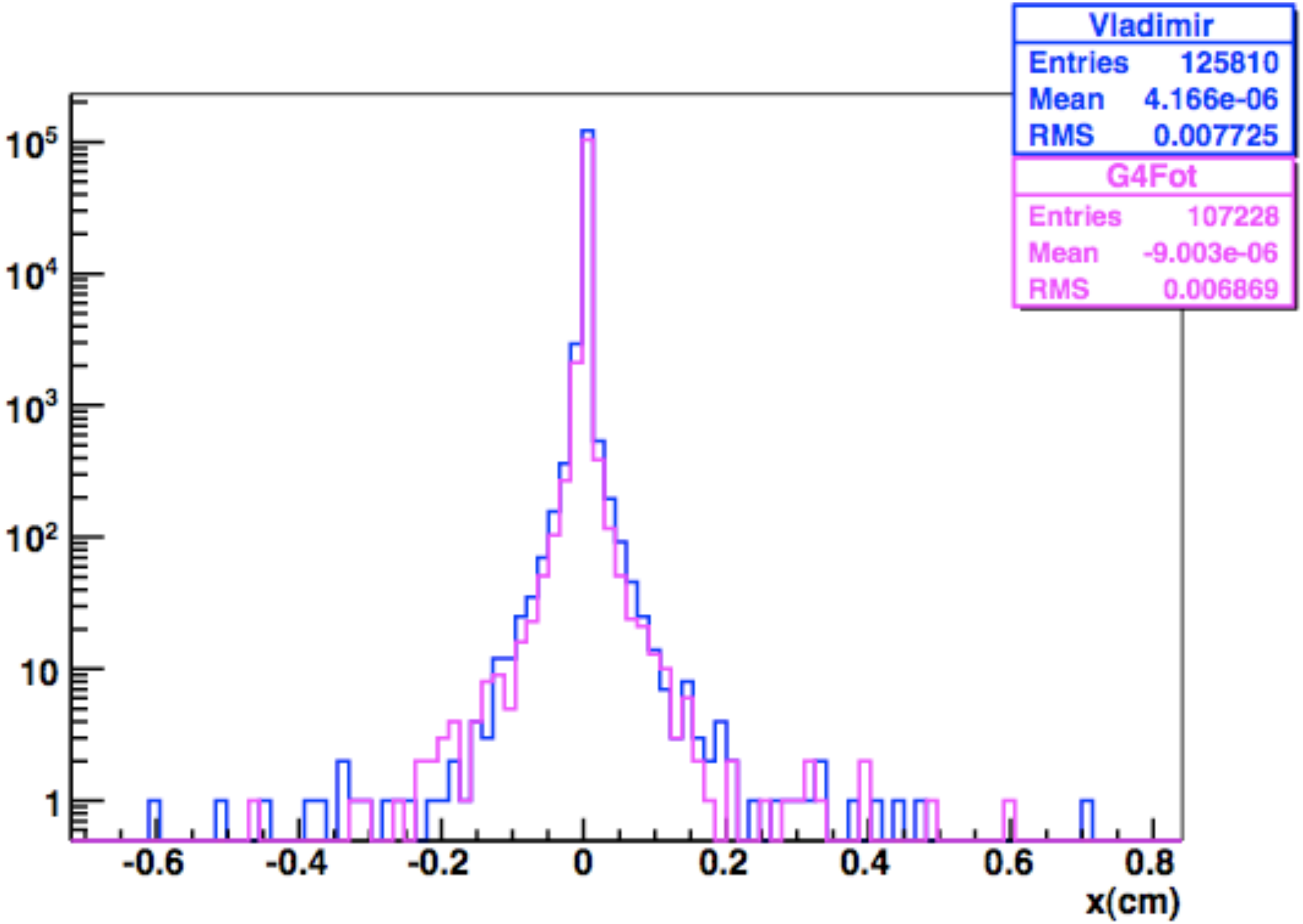
3 GeV incident e- beam (e=1.6 mm)
Photon energy distribution



G4Fot seems to give harder photons than Vladimir's simulation

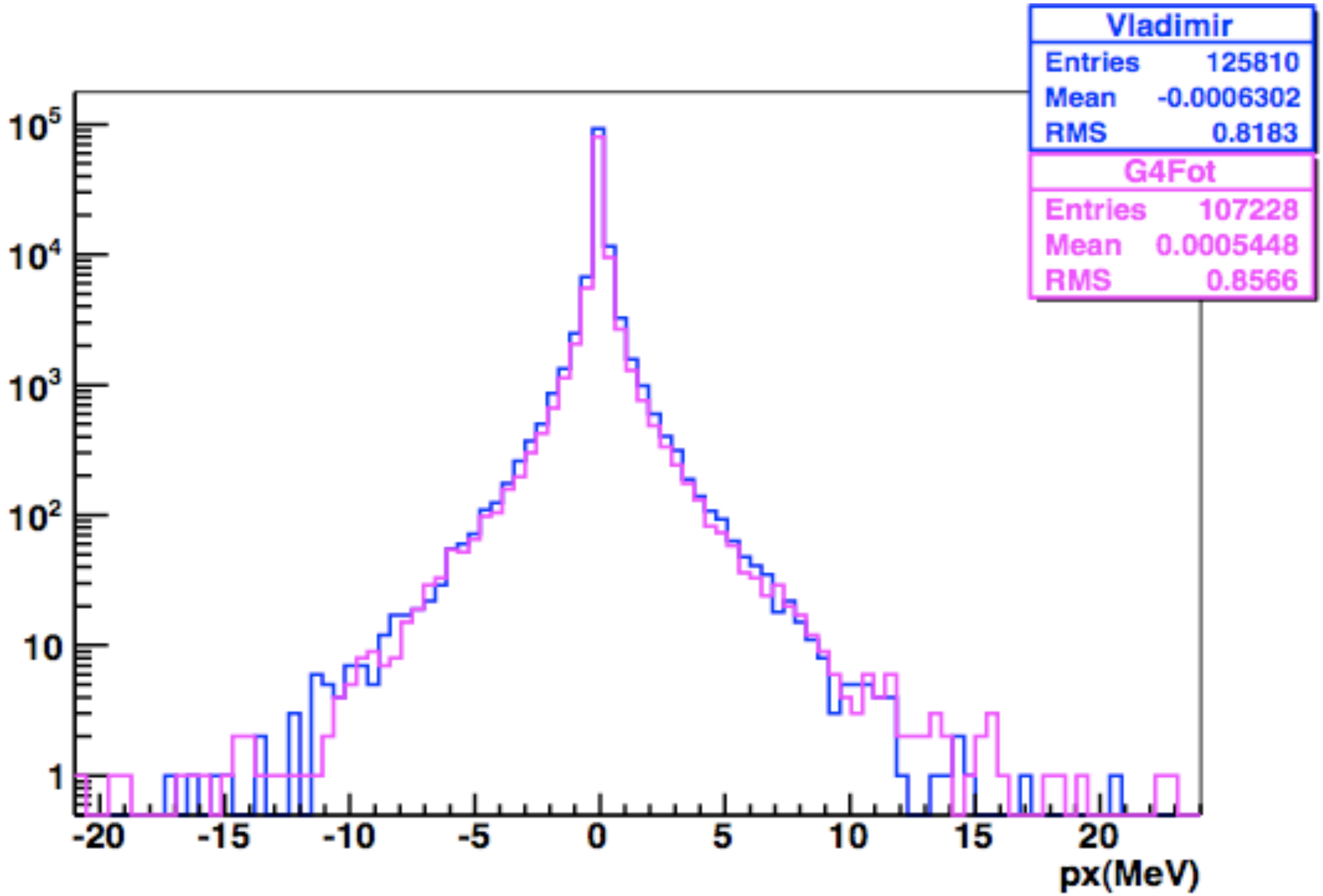
3 GeV incident e- beam (e=1.6 mm)

Photon energy distribution

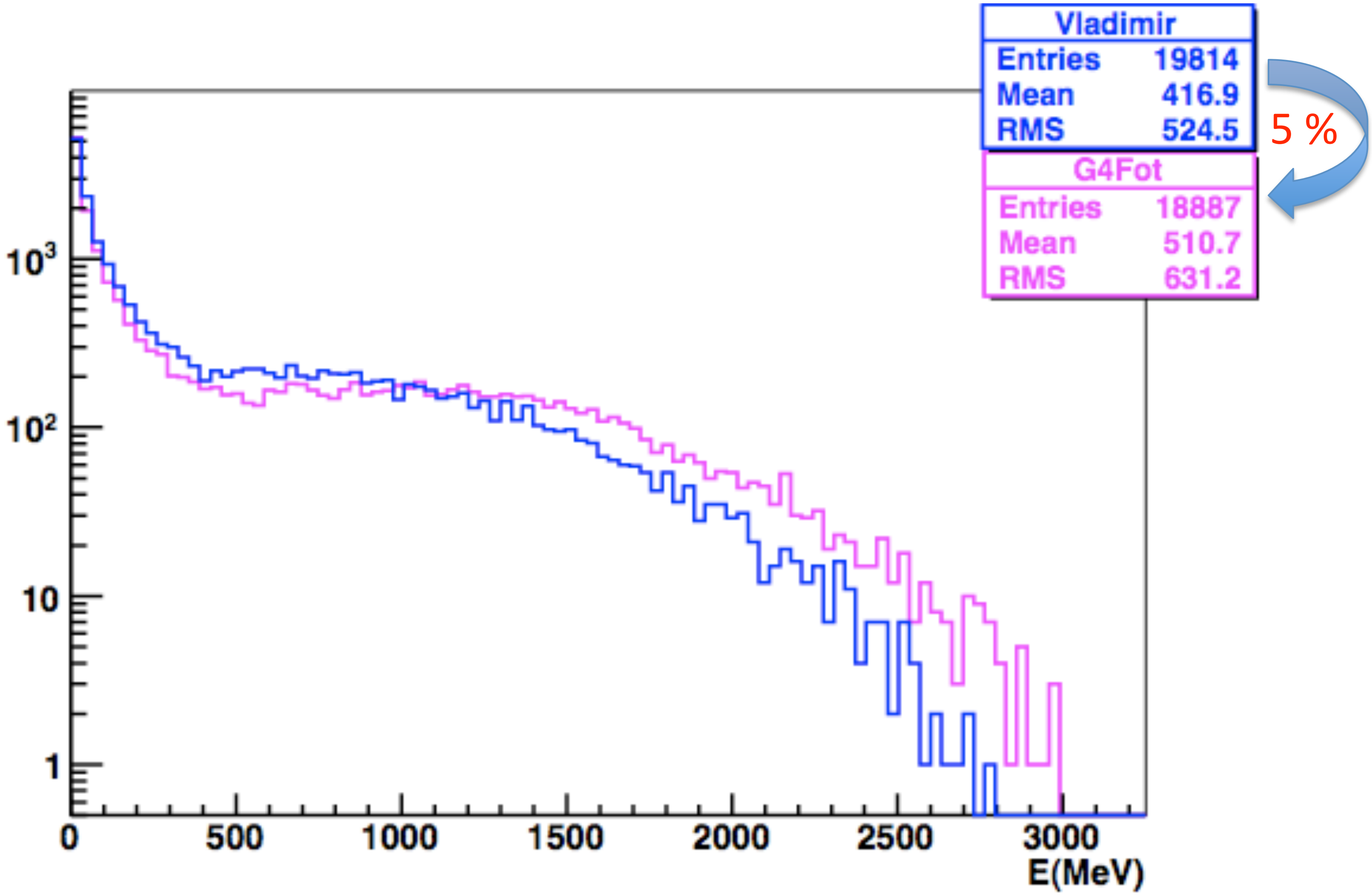


3 GeV incident e- beam (e=1.6 mm)

Photon energy distribution

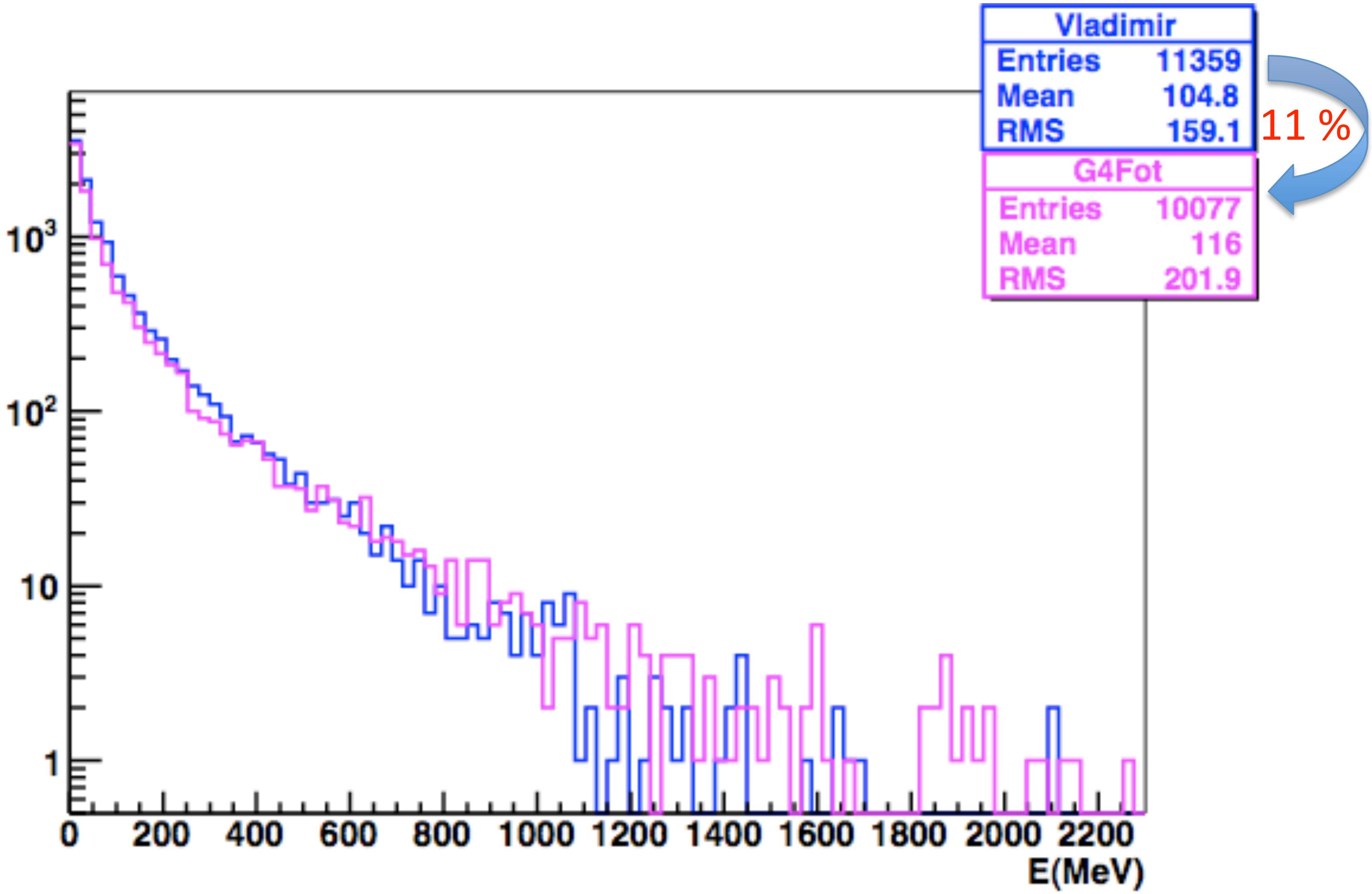


3 GeV incident e- beam (e=1.6 mm)
electron energy distribution



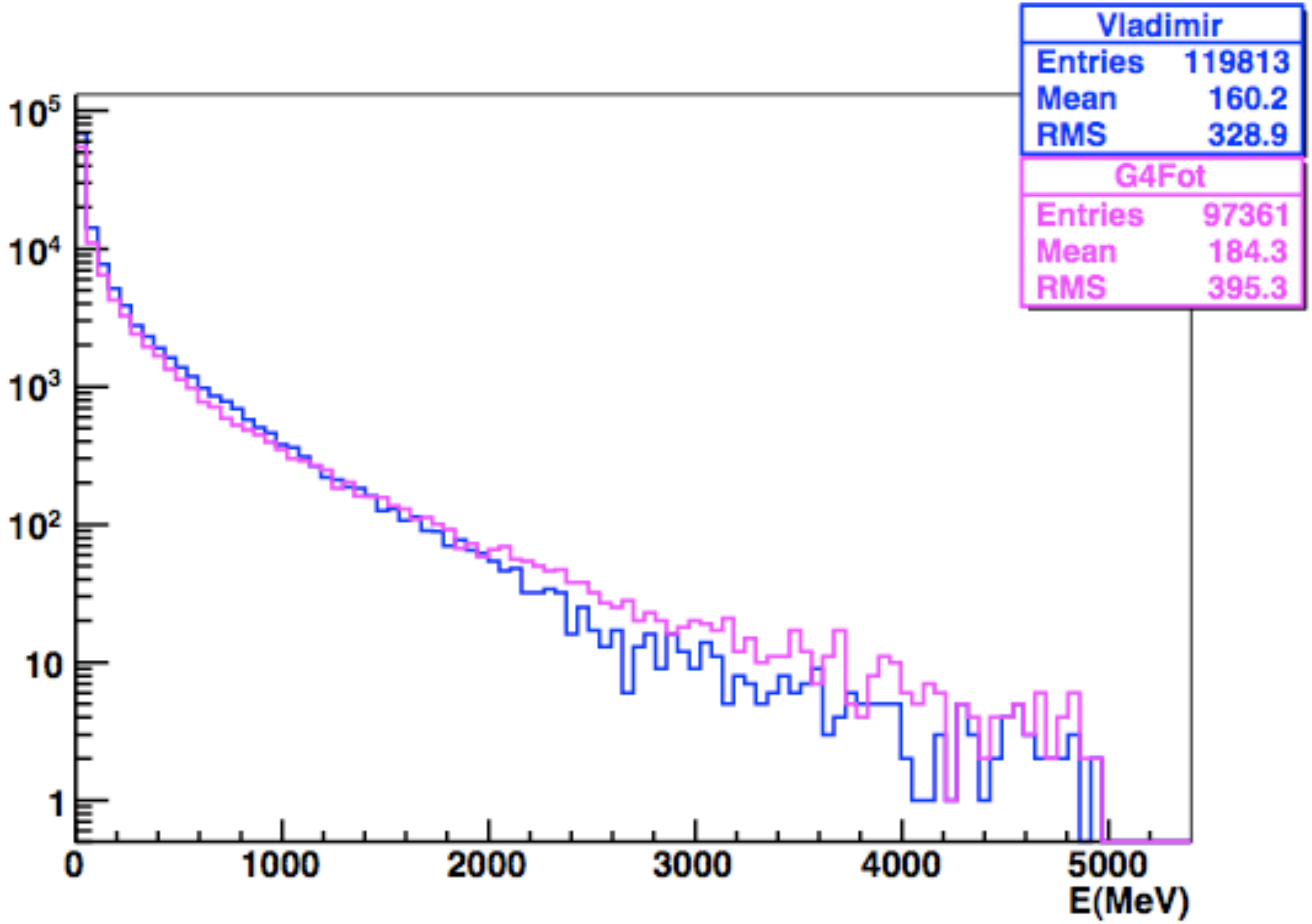
G4Fot seems to give “harder” e- than Vladimir’s simulation

3 GeV incident e- beam (e=1.6 mm)
e+ energy distribution

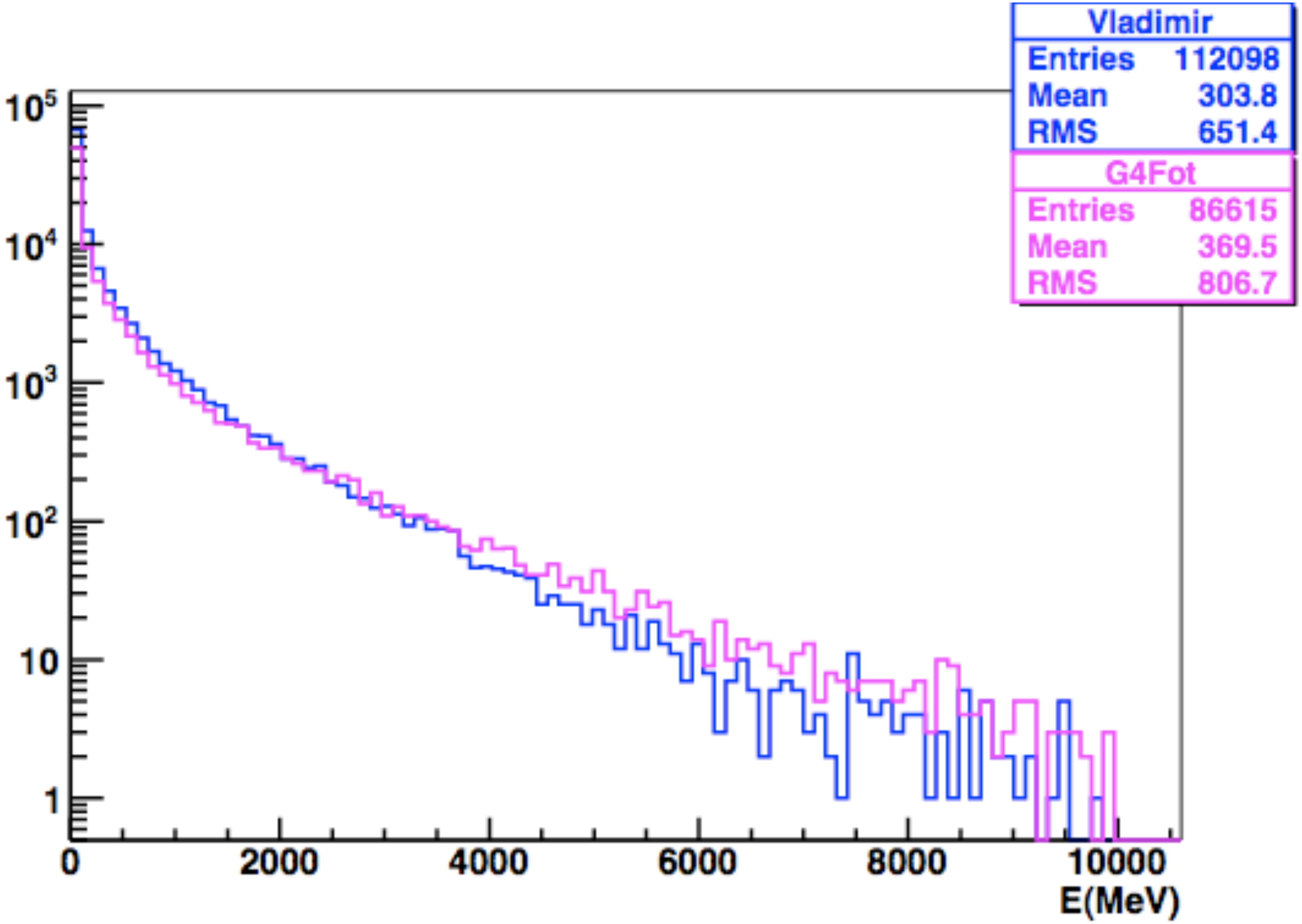


G4Fot seems to give “harder” e+ than Vladimir’s simulation

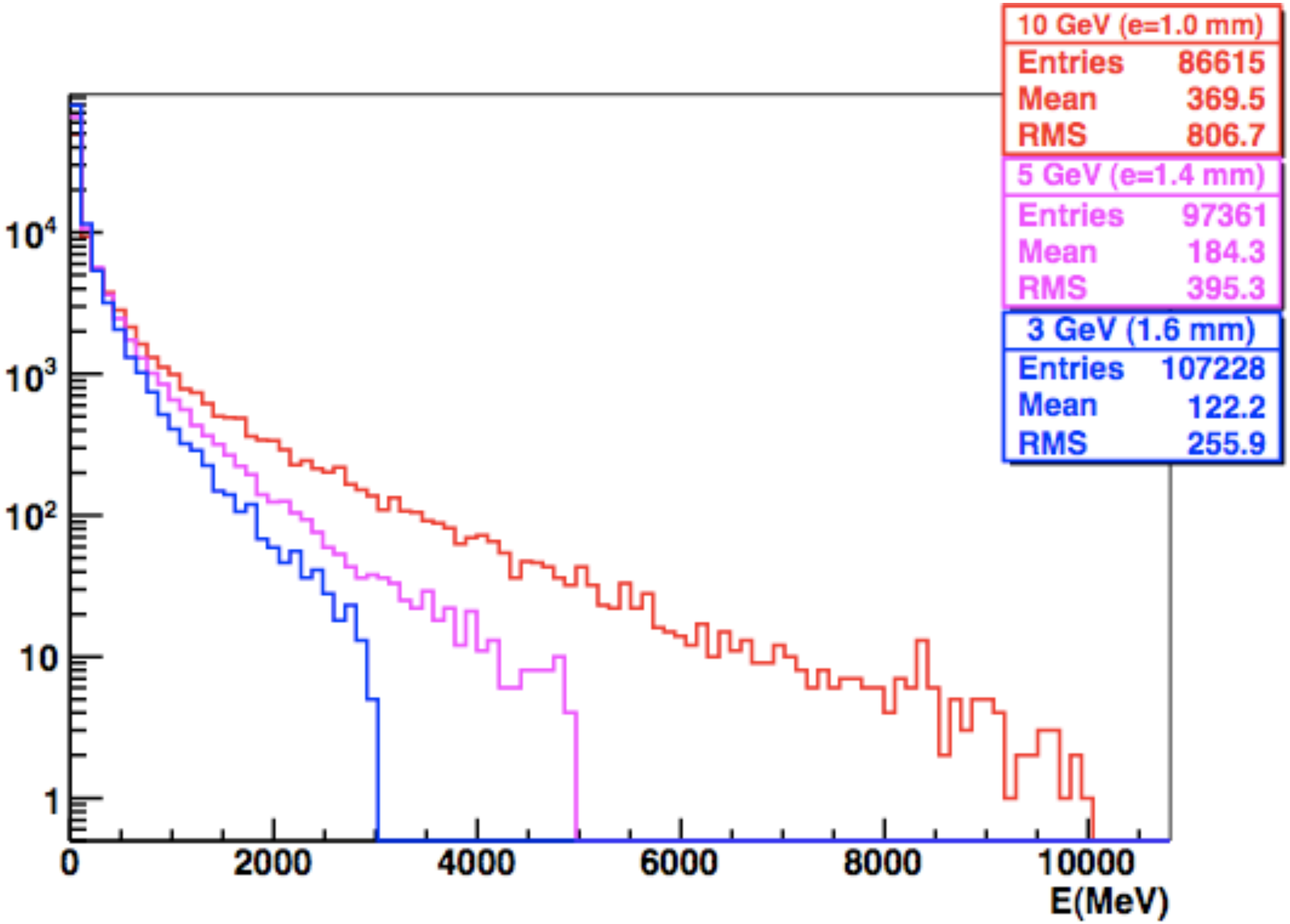
5 GeV incident e- beam (e=1.4 mm)
Photons energy distribution



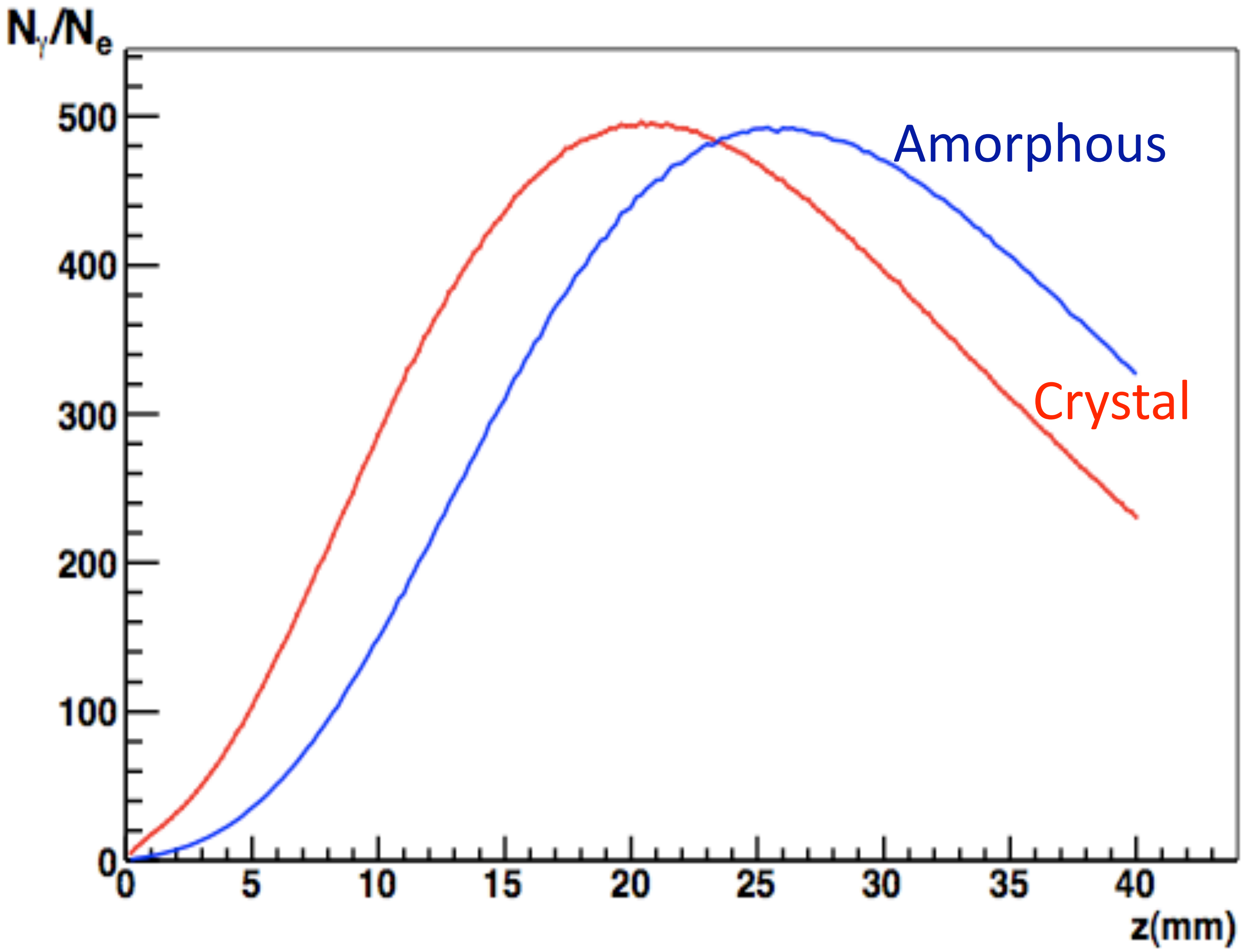
10 GeV incident e- beam (e=1. mm)
Photons energy distribution



Photons energy distribution



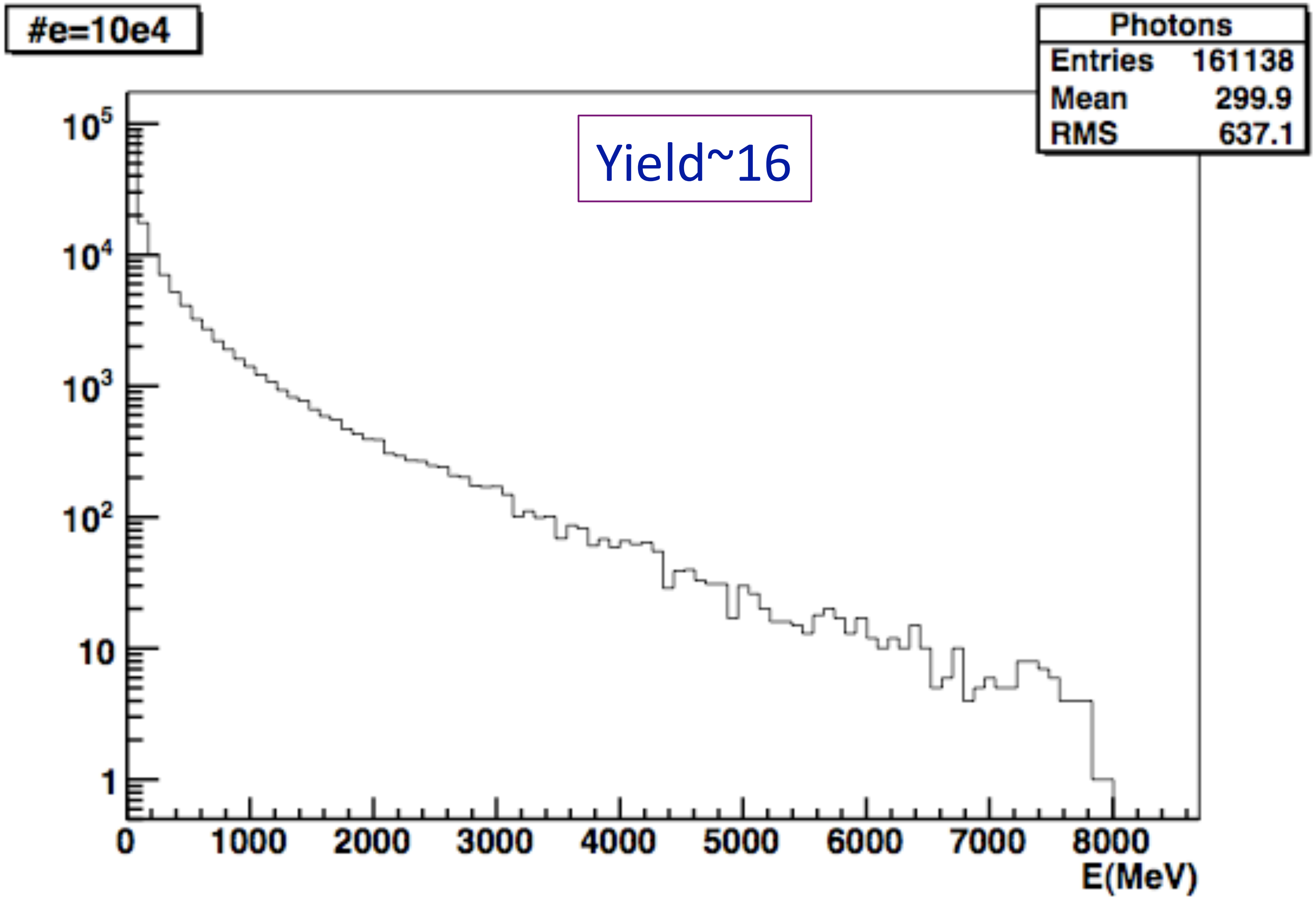
Crystal and amorphous gamma yield comparison @ 8GeV



In the case of the crystal E.M shower growing faster than in the case of the amorphous

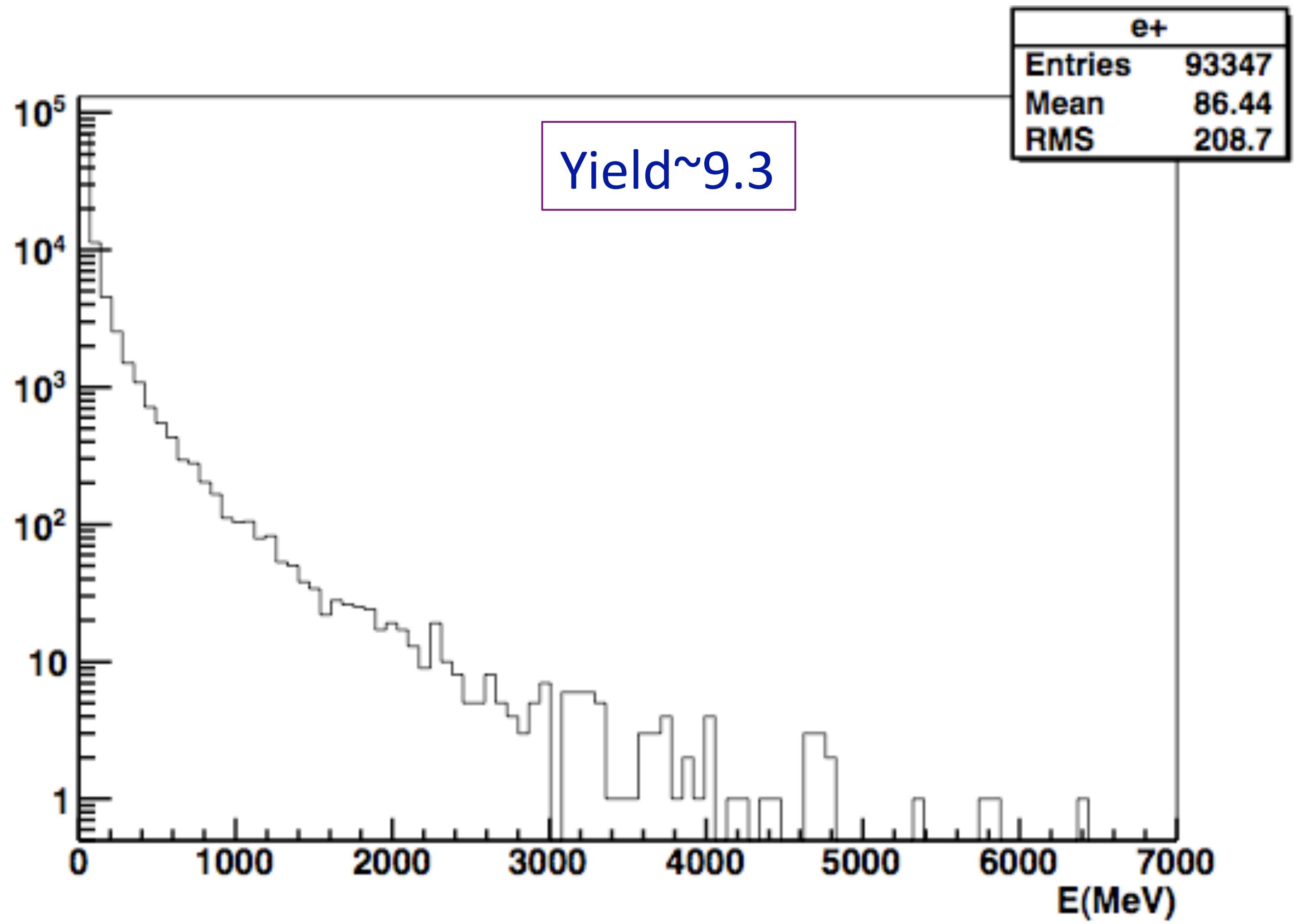
KEK configuration
8 GeV e- : 1mm crystal / 8mm amorphous (3m)

Energy photon distribution after the 1 mm crystal



KEK configuration
8 GeV e- : 1mm crystal / 8mm amorphous (3m)

Energy e+ distribution after the 8 mm amorphous



Hybrid source comparison concerning the e+ yield

- Distance crystal amorphous 2m
- Amorphous thickness 10. mm

Energy (GeV)	X(mm)	Vladimir	G4Fot
10	1.0	15.0	13.1
5	1.4	8.0	7.0
3	1.6	3.4	3.9

Conclusion & prospects

- Without any modification inside your Geant4 code you can use G4FOT (can be used as generator for Fluka or EGS)
- Particle multiplicity is less for G4Fot than for Vladimir's code
- Particles mean energies is "harder" for G4Fot than for Vladimir's code
  - Need to understand those differences
  - Compare G4Fot with an other simulation (S. Dabagov)
- Full STL : trigonometry, random generator …
- Add energy deposition calculation
- A SVN repository will be very soon available for the community
- We have a very very good opportunity to cross check G4Fot with real data from KEKb experiment