Group 2 : Phasespace Generator

Physics Team Project Report

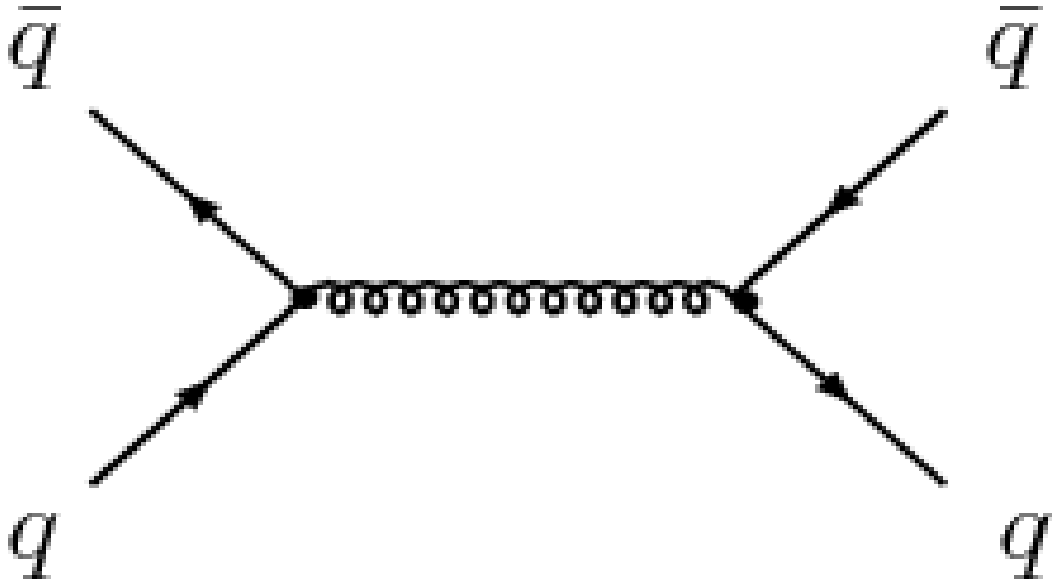
Svende Braun
Johann Brehmer
Manuel Geisler
Dominik Mitzel

Heidelberg, 15.03.16

Introduction

- tool which translates a vector of random numbers into external momenta for given process

- Process: u ubar → u ubar in LO QCD



Conditions

- Beam: pp collisions at 13 TeV
  - No dependencies on external libraries
  - Analytical parametrization of PDF by K. Armour, M. Bowen, S. Ellis, M. Strassler 2004
- Shared repository on gitHub allowing for parallel code development
- Flexibility:
  - Variable IR cut
  - Variable number of generated events

Our process...

- For 2x2 scattering @LHC, 4DOF per event
  - Z-component of initial momenta(x1,x2)
  - Spatial orientation of final state momenta (Mandelstam t, phi)
  - If we have x1,x2,t,phi → all momenta are fixed!

- Start from 4 random numbers z1,z2,z3,z4 in [0,1]

```
x1 = exp((1.-z1) * log(xmin));

x2 = exp((1.-z2) * log(xmin));

s = x1*x2*13TeV;
```

```
t = -s + z3*s;

phi3 = 2.*M_PI*z4;
```

Variable transformation

$$\sigma = \int \mathrm{d}x_1 \int \mathrm{d}x_2 \int \mathrm{d}t \int \frac{\mathrm{d}\phi}{2\pi} \quad f(x_1)\, f(x_2) \quad \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}t}(x_1 x_2 s, t)$$

$$= \int_0^1 \mathrm{d}^4\mathbf{z} \quad \left|\det \frac{\partial(x_1, x_2, \phi, t)}{\partial \mathbf{z}}(\mathbf{z})\right| \quad f(x_1(\mathbf{z}))\, f(x_2(\mathbf{z})) \quad \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}t}(\mathbf{z})$$

$$\simeq \sum_i \underbrace{\frac{1}{N} \quad \left|\det \frac{\partial(x_1, x_2, \phi, t)}{\partial \mathbf{z}}(\mathbf{z}_i)\right| \quad f(x_1(\mathbf{z}_i))\, f(x_2(\mathbf{z}_i)) \quad \frac{\mathrm{d}\hat{\sigma}}{\mathrm{d}t}(\mathbf{z}_i)}_{=w_i} \cdot \Delta z_1 \Delta z_2 \Delta z_3 \Delta z_4$$

- W is global event weight
- Kinematics are fixed

Where:

```
double Generator::CalculateDSigmaDt(double s, double t, double u) {
 const double alphaS = 0.1;
  if (t != 0. && s != 0. && u != 0.)
    return 4.*M_PI*alphaS*alphaS/(9.*s*s) * (t*t + u*u)/(s*s);
  Return 0.;}
```

Thanks Peskin-Schröder ;)

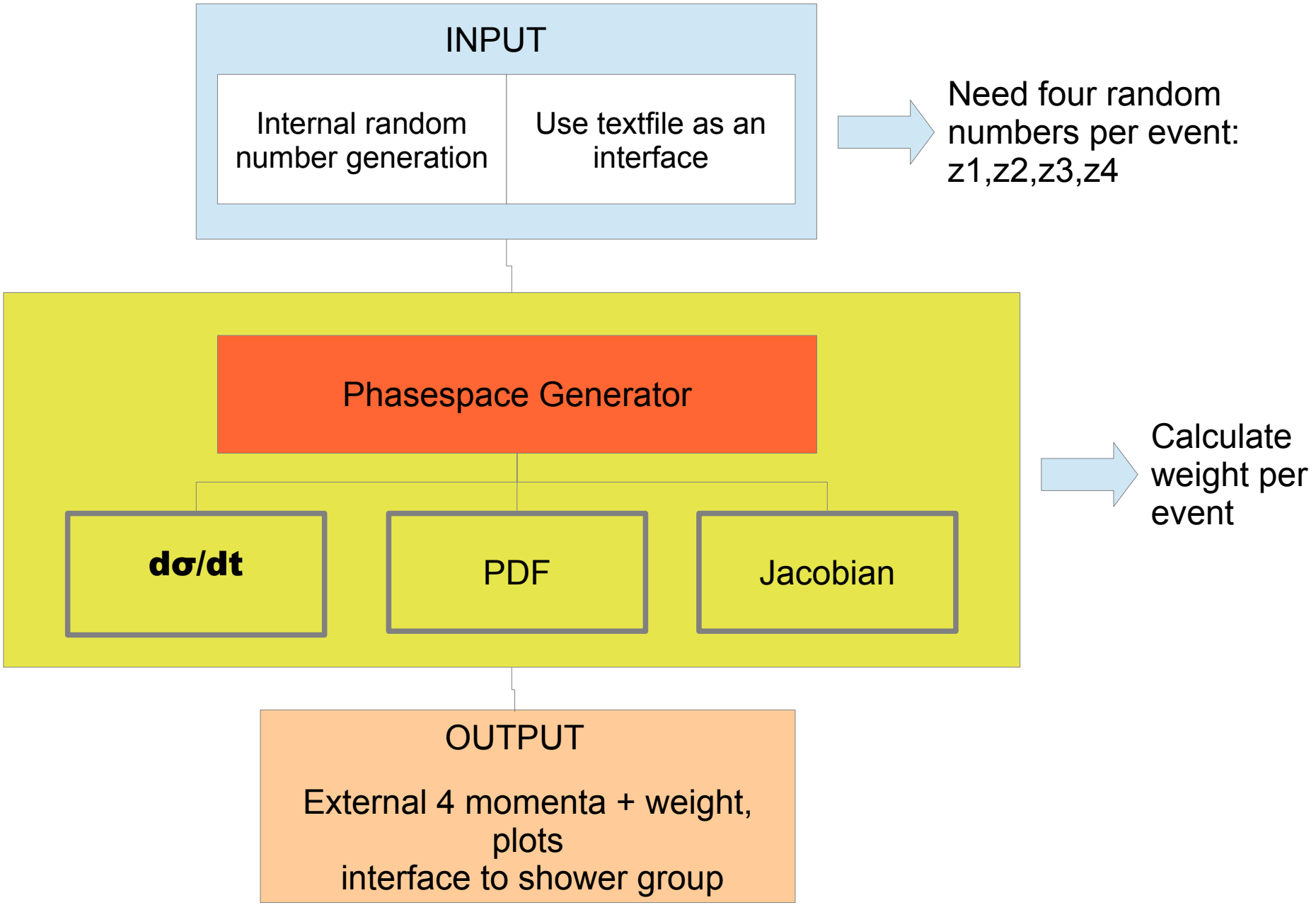
```
double Generator::GetPDFValue(int pdgid, double x, double q)
{

....
  return 1./n * pow(x,-b) * pow((1-x),a);}
```

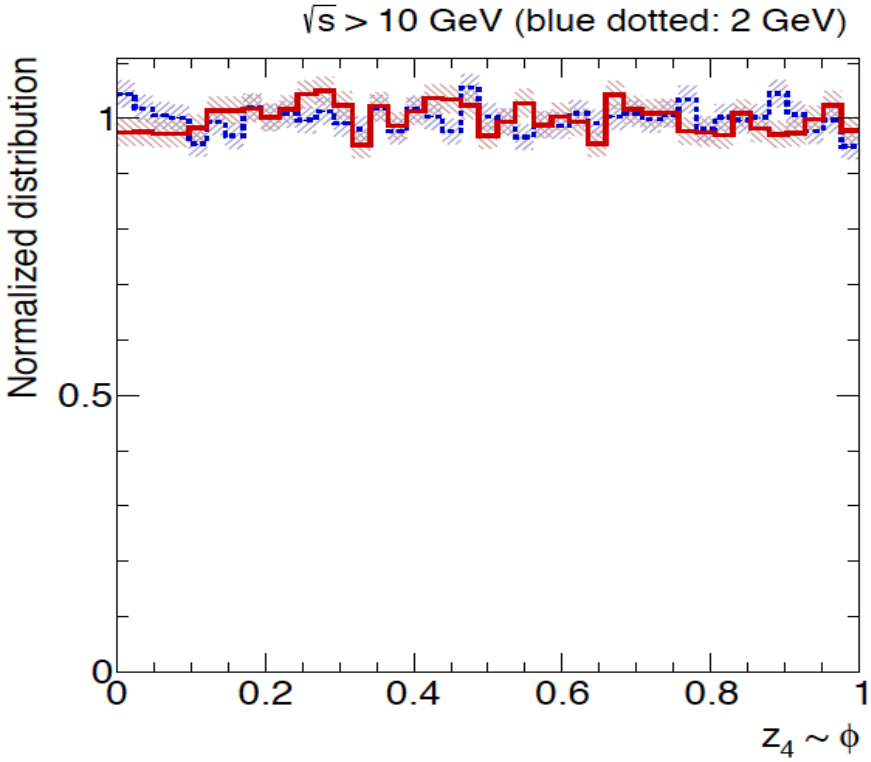
Analytical parametrisation

With appropriate coefficients

Schematic

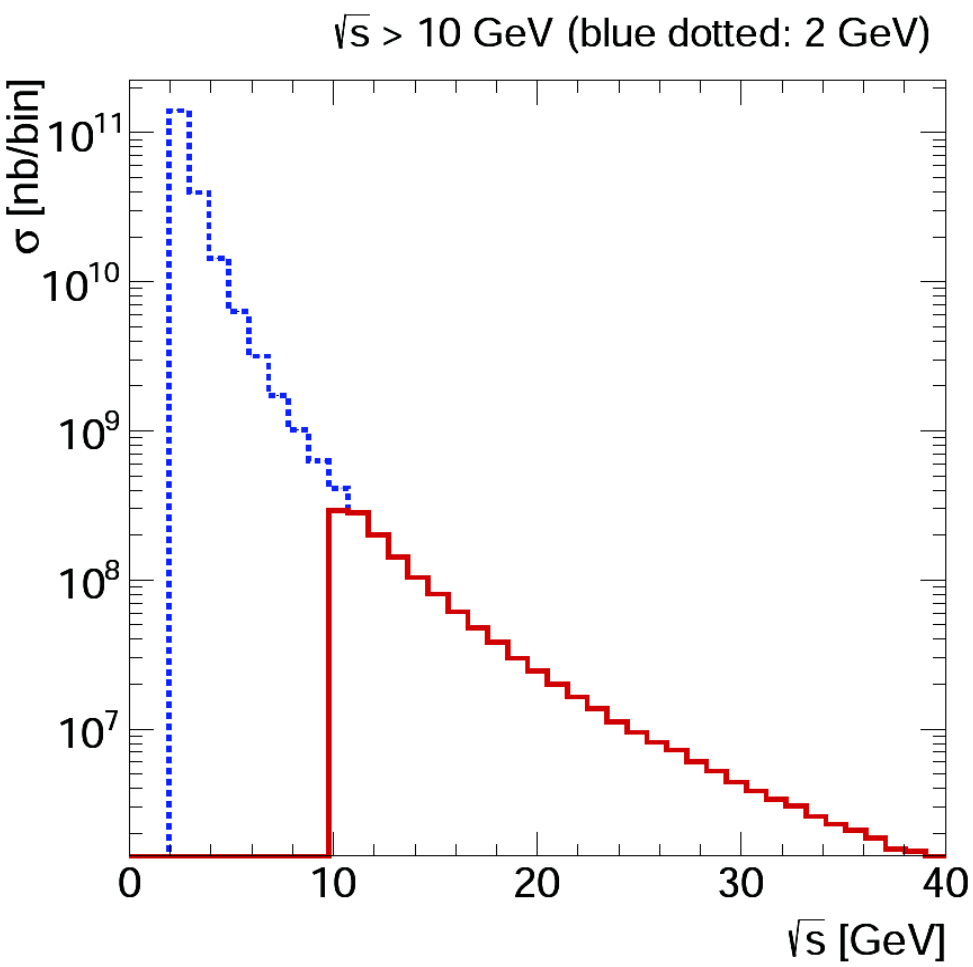

INPUT
Internal random number generation
Use textfile as an interface
Need four random numbers per event: z1,z2,z3,z4
Phasespace Generator
dσ/dt
PDF
Jacobian
Calculate weight per event
OUTPUT
External 4 momenta + weight, plots
interface to shower group

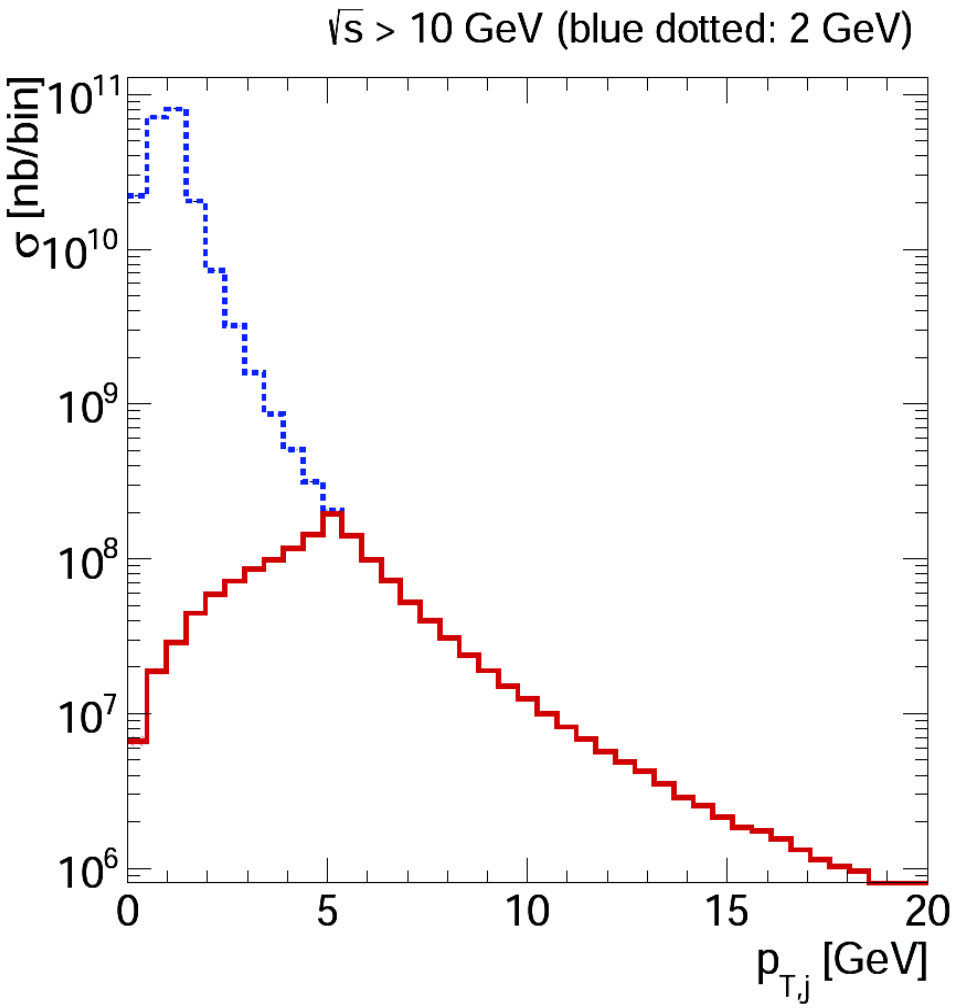
Plots



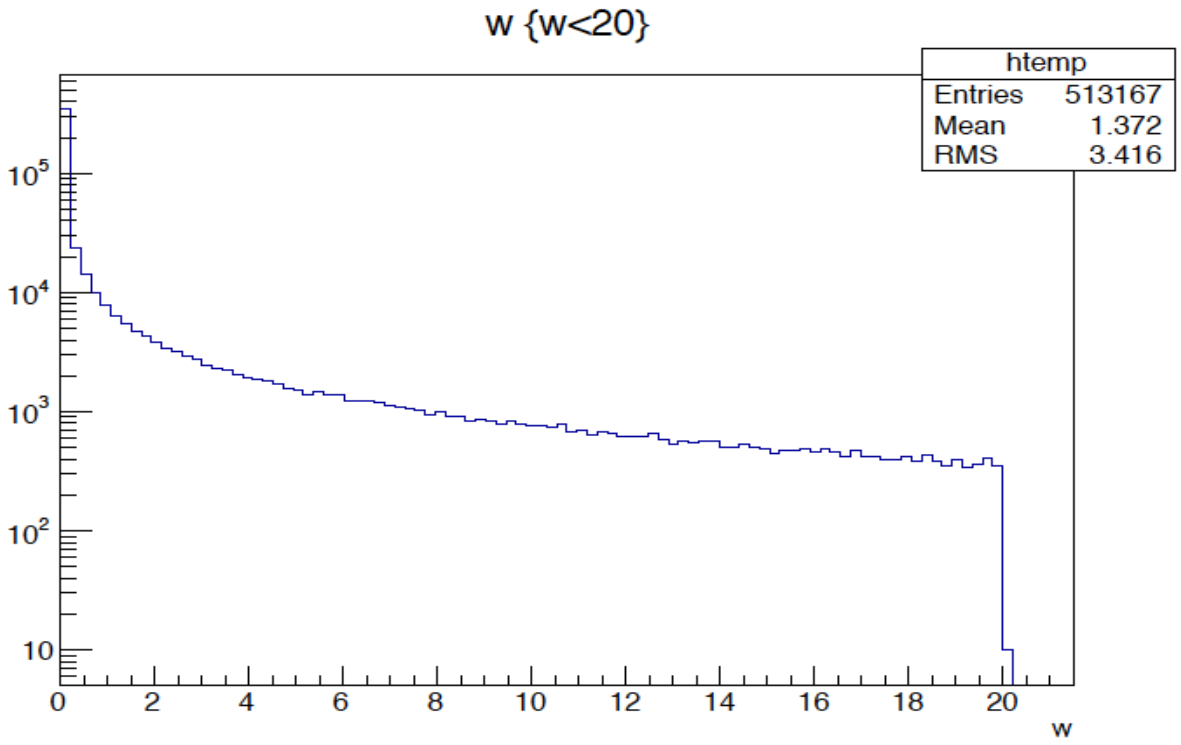
Consistency check: process invariant under phi transformations

Plots


√s > 10 GeV (blue dotted: 2 GeV)
σ [nb/bin]
10^11
10^10
10^9
10^8
10^7
0
10
20
30
40
√s [GeV]


√s > 10 GeV (blue dotted: 2 GeV)
σ [nb/bin]
10^11
10^10
10^9
10^8
10^7
10^6
0
5
10
15
20
p_T,j [GeV]

Plots

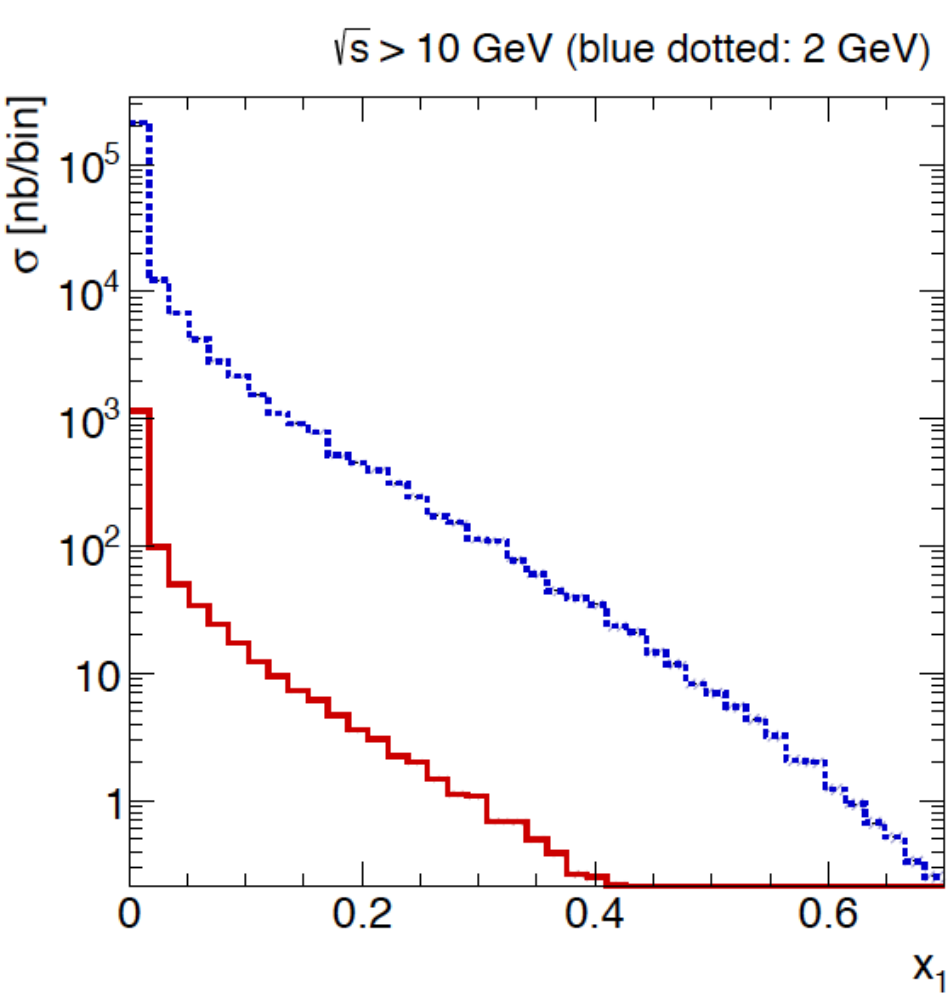


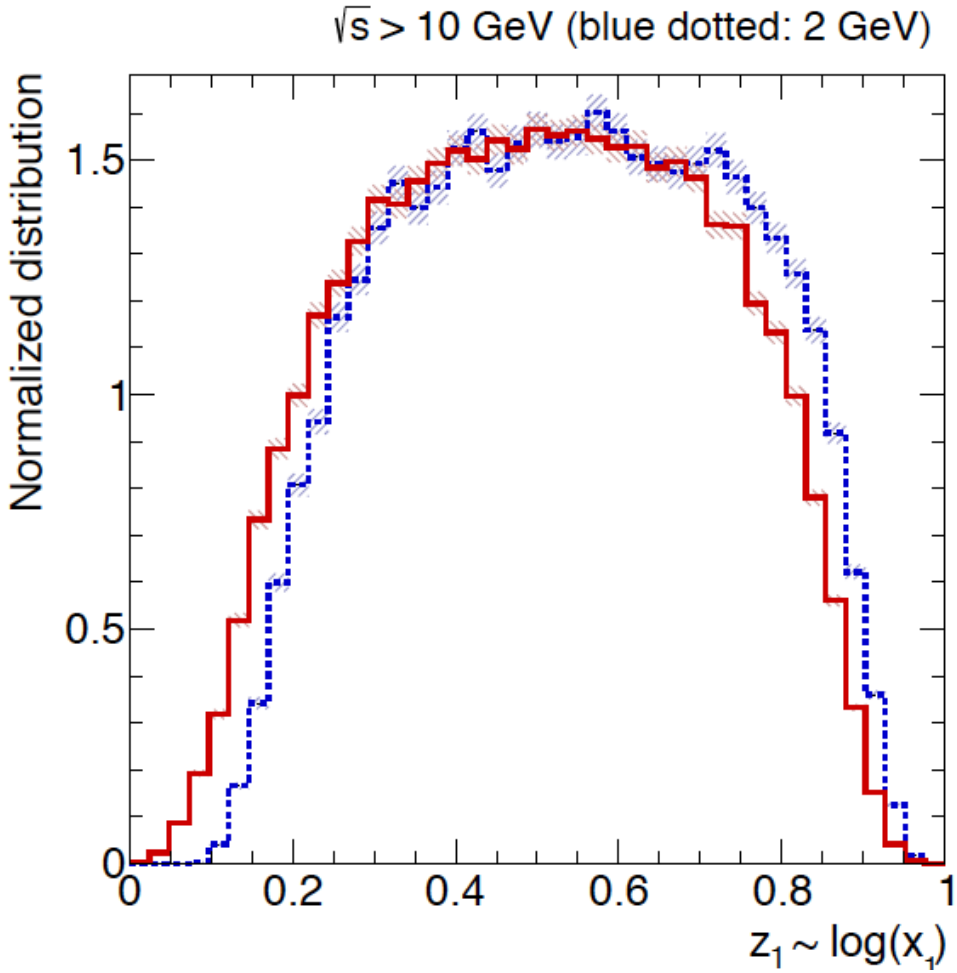
- weights not flat → not the ideal case for MC integration to get total xsec
- Better transformation could improve performance

Conclusions

- We have written a modular framework for a phase space generator of LO QCD processes
- Currently implemented for u ubar → u ubar in s-channel interactions
- Easily extendable to include further processes, more sophisticated PDFs or flatter transformations
  → especially important for inclusion of resonances
- PSG works and produces reasonable results (cross-checked with MadGraph)
- Can easily be chained with the other Physics Team projects

Appendix


√s > 10 GeV (blue dotted: 2 GeV)
σ [nb/bin]
10^5
10^4
10^3
10^2
10
1
0
0.2
0.4
0.6
x_1


√s > 10 GeV (blue dotted: 2 GeV)
Normalized distribution
1.5
1
0.5
0
0
0.2
0.4
0.6
0.8
1
z_1 ~ log(x_1)