Introduction to Pythia8

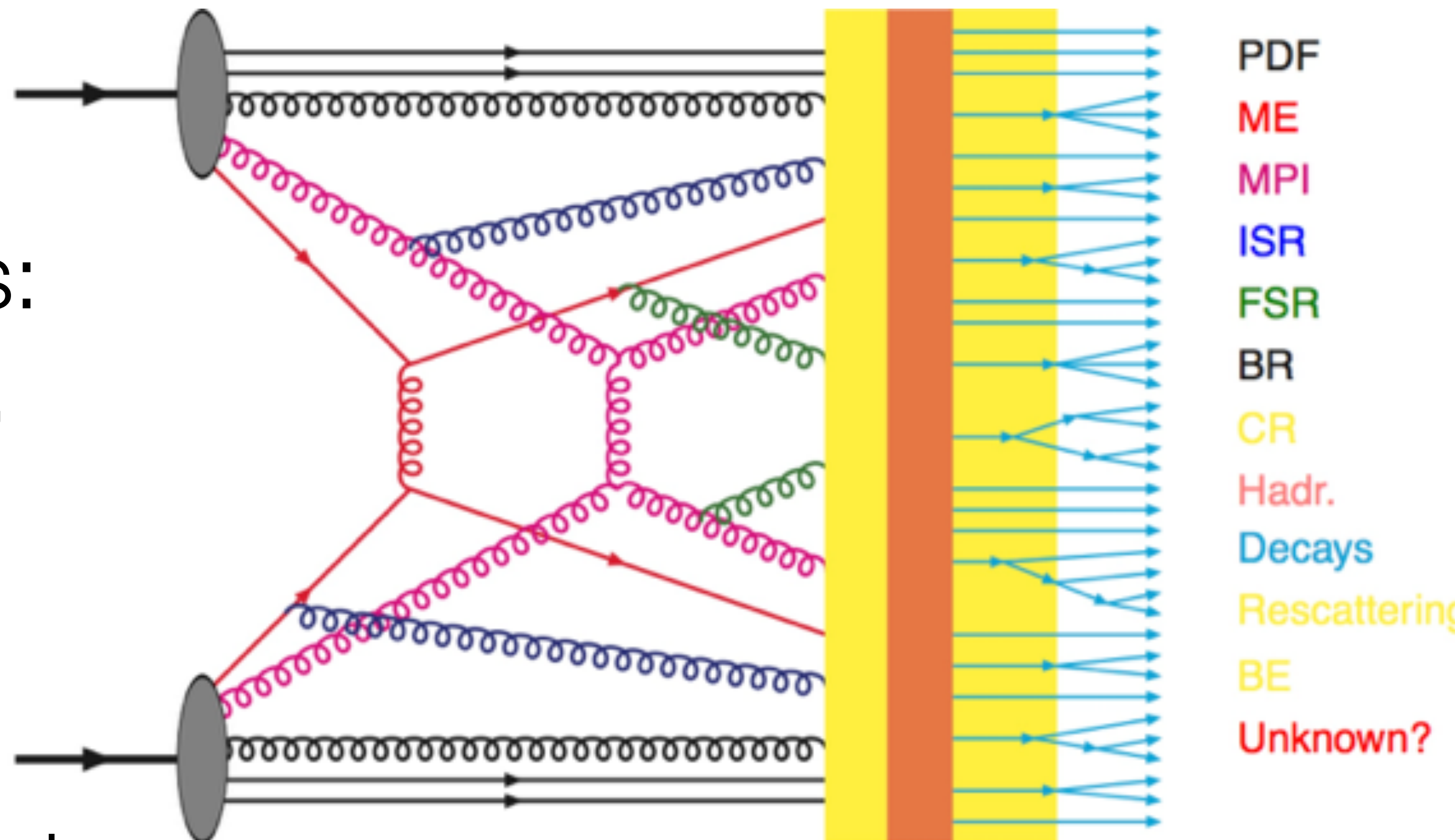
IFT Madrid, 9 Sep 2015

Plan of this talk:

- Purpose of “general-purpose” MC generators
- Program flow
- ISR + FSR
- Hadronisation
- MPI
- (TUT) Hola Mundo! (Simple Z production)
- Using internal BSM models (Z’ production)
- (TUT) Links to external programs I: LHE files
- Aside: Links to external programs II: Semi-internal processes
- Aside: Jet algorithms
- Simplified event analysis (w/o dedicated detector sim.)
- (TUT) Links to external programs III: HEPMC and ROOT
- (TUT) Matching and merging

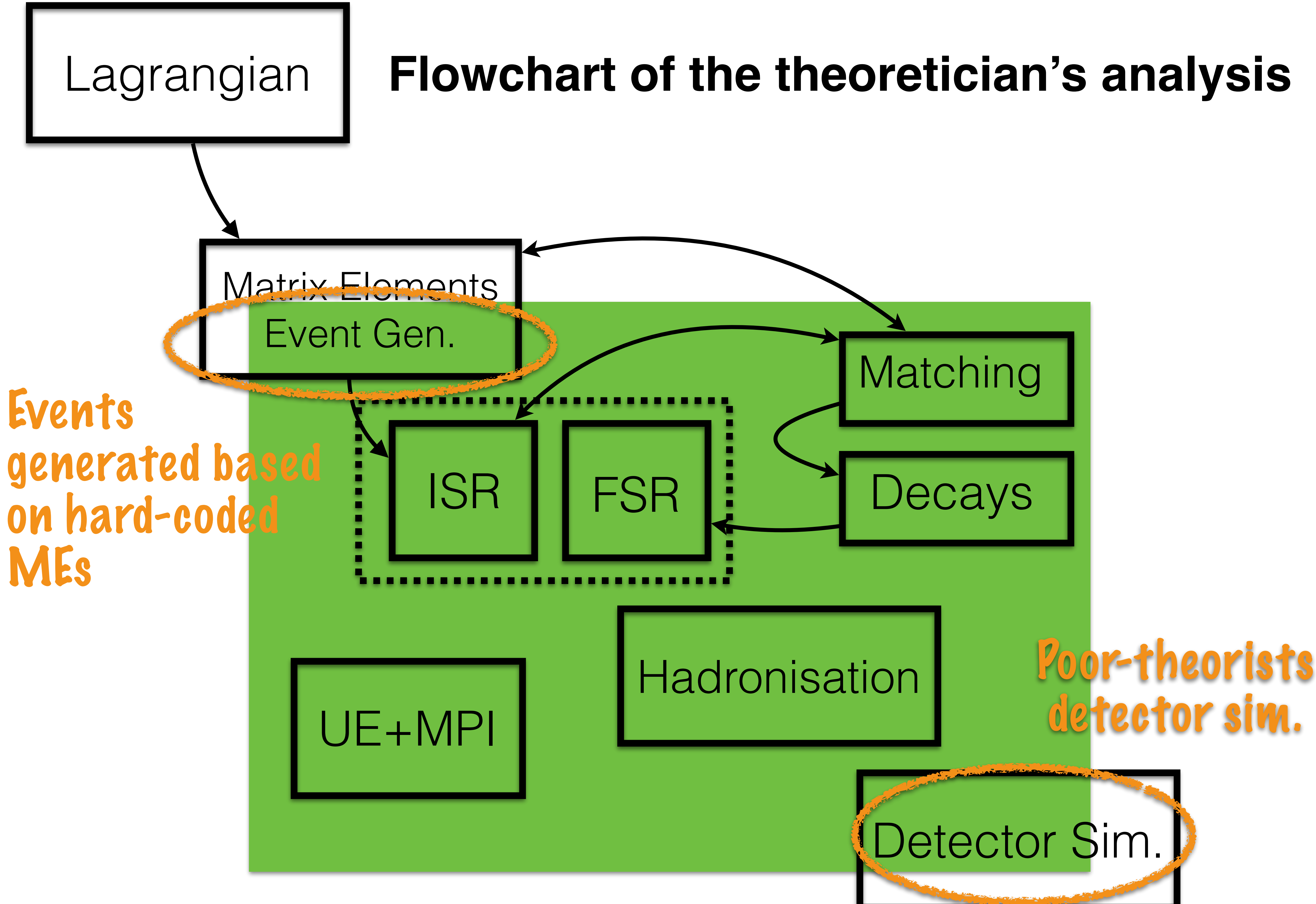
What is Pythia 8?

- General purpose Monte-Carlo generator
- Simulates collision events: hard process, showering, hadronisation, multiple interactions, underlying event ...
- Version 8.2 released in Oct 2014. Current version 8.210



Download and online manual from http://home.thep.lu.se/~torbjorn/Pythia.html

Flowchart of the theoretician's analysis



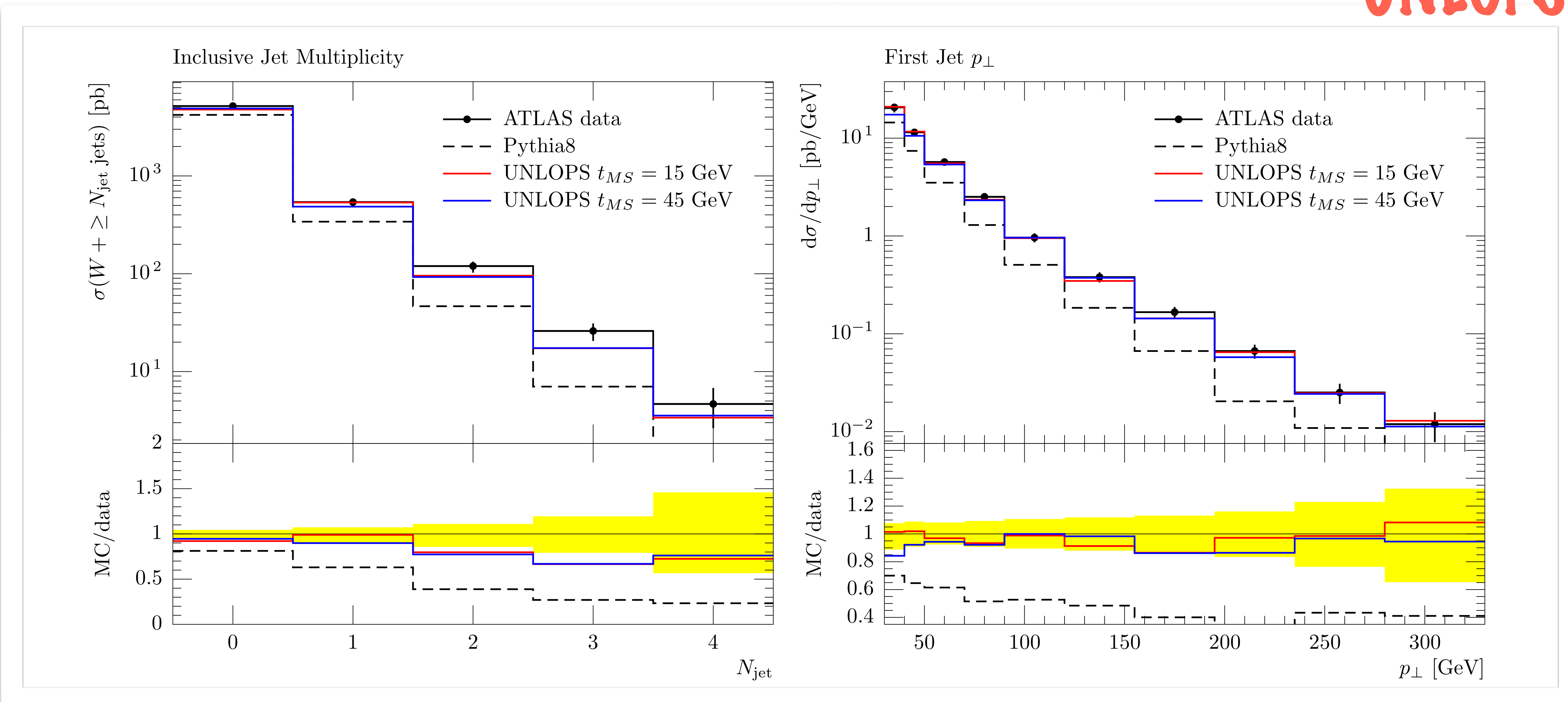
- Pythia 8 has a basic library of BSM processes that can be used for quick studies.
  - BSM Higgses (2HDM)
  - Fourth generation quarks
  - New Gauge Bosons
  - Left-Right symmetric models
  - Leptoquarks
  - Compositeness
  - Hidden Valley
  - Extra Dimensions
  - SUSY

- More exotic processes may be implemented via external programs.

- ✦ Important to model backgrounds correctly to observe BSM signatures.
- ✦ Current state-of-the-art SM calculations use ME generators + PS with matching followed by hadronisation.
  - ➡ Pythia 8 provides various interfaces to external ME generators
  - ➡ LO/NLO matching for processes available

ME + PS matching at NLO!

UNLOPS



Lonnblad and Prestel; arXiv:1211.7278

Also available LO matching via new Unitarised ME+PS merging (UMEPS), MLM and CKKW-L.

Interfaces

- ✦ Interface to LHAPDF or other external PDF libraries.
- ✦ Les Houches Accord (LHA) files for reading events or runtime LHA interface.
- ✦ Semi-internal processes for programs like Madgraph 5.
- ✦ HepMC output for programs like RIVET, Delphes etc.
- ✦ Can be compiled as a plugin to ROOT.
- ✦ Input from generalised SLHA input for any BSM model.

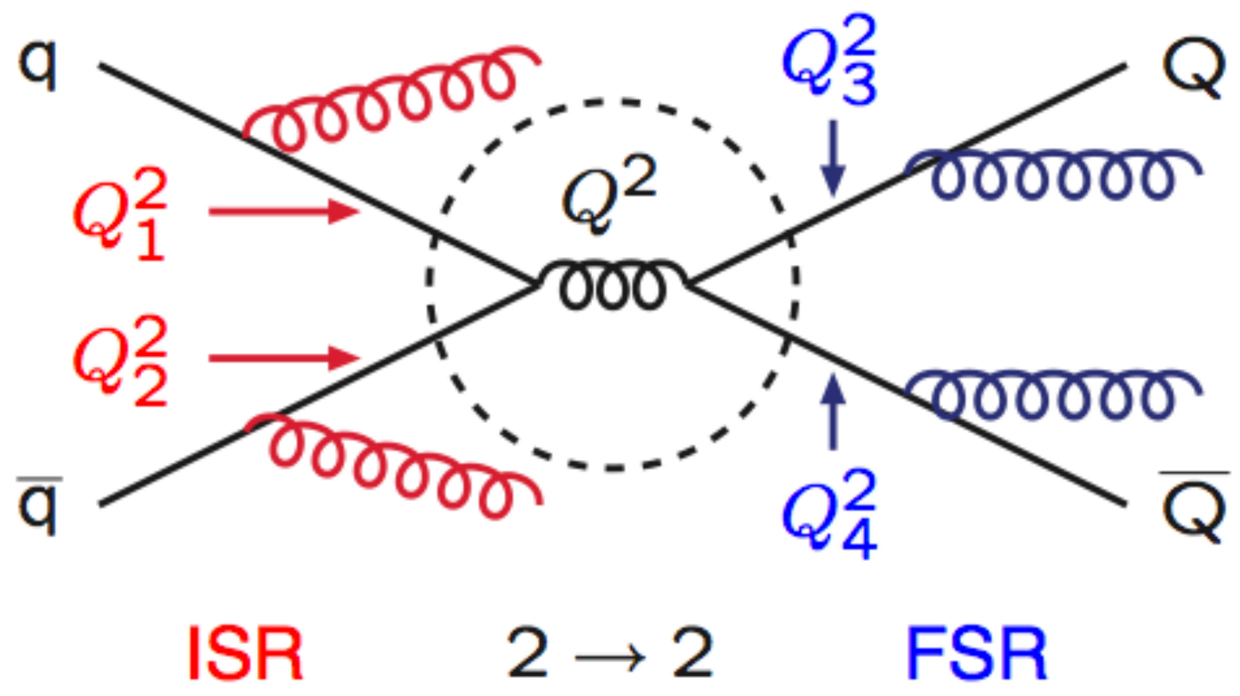
Other major improvements:

- ✦ Improvements to parton showers; possibility to use external PS programs (e.g. Vincia)
- ✦ Improvements to MPI
- ✦ Showering to take into account colour-epsilon topologies, sextets.
- ✦ Hadronisation in presence of coloured exotic particles (R-hadrons [*M. Fairbairn et al.,Phys. Rep. 438 (2007)*], long-lived triplets or octets, ...)
- ✦ Tau polarisation in both production and decay [*P. Ilten, arXiv:1211.6730 [hep-ph]*]

User-friendly, more intuitive; settings can be changed without requiring recompilation.

The Parton-Shower Approach

$$2 \to n \quad = \quad (2 \to 2) \oplus \text{ISR} \oplus \text{FSR}$$



FSR = Final-State Radiation = timelike shower

$Q_i^2 \sim m^2 > 0$ decreasing

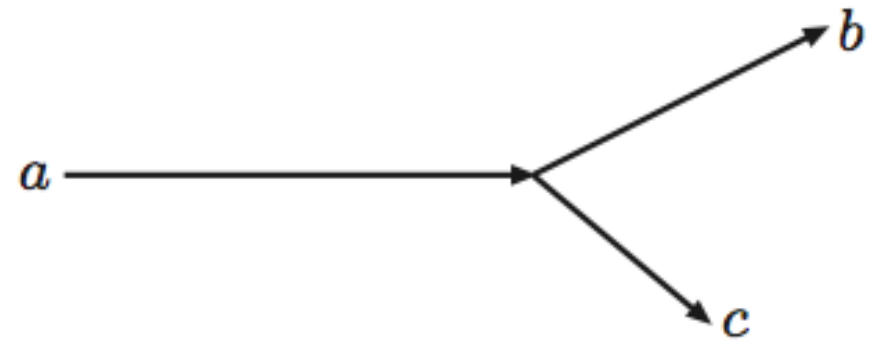
ISR = Initial-State Radiation = spacelike showers

$Q_i^2 \sim -m^2 > 0$ increasing

Why "time" like and "space" like?

Consider four-momentum conservation in a branching $a \to b\,c$

$$\mathbf{p}_{\perp a} = 0 \quad \Rightarrow \quad \mathbf{p}_{\perp c} = -\mathbf{p}_{\perp b}$$
$$p_{+} = E + p_{\mathrm{L}} \quad \Rightarrow \quad p_{+a} = p_{+b} + p_{+c}$$
$$p_{-} = E - p_{\mathrm{L}} \quad \Rightarrow \quad p_{-a} = p_{-b} + p_{-c}$$



Define $p_{+b} = z\,p_{+a}, \quad p_{+c} = (1-z)\,p_{+a}$

Use $p_{+}p_{-} = E^2 - p_{\mathrm{L}}^2 = m^2 + p_{\perp}^2$

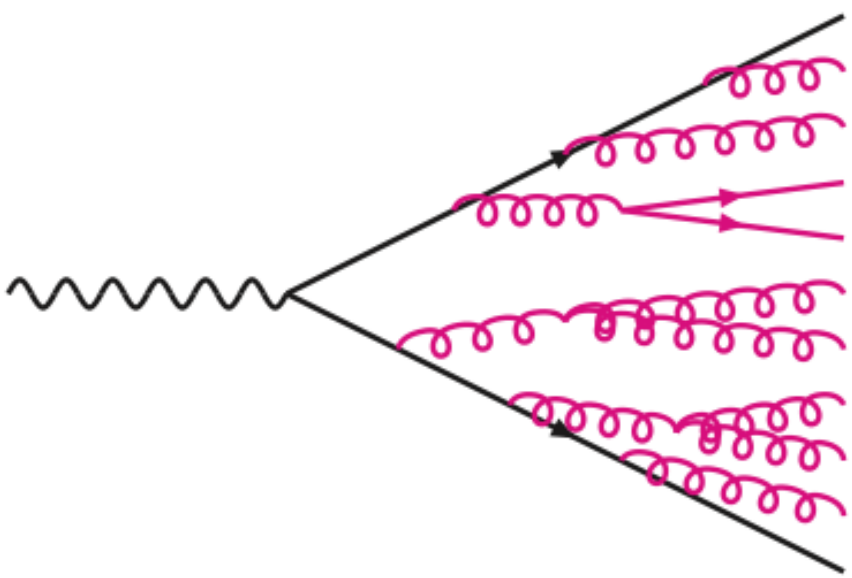
$$\frac{m_a^2 + p_{\perp a}^2}{p_{+a}} = \frac{m_b^2 + p_{\perp b}^2}{z\,p_{+a}} + \frac{m_c^2 + p_{\perp c}^2}{(1-z)\,p_{+a}}$$

$$\Rightarrow \quad m_a^2 = \frac{m_b^2 + p_{\perp}^2}{z} + \frac{m_c^2 + p_{\perp}^2}{1-z} = \frac{m_b^2}{z} + \frac{m_c^2}{1-z} + \frac{p_{\perp}^2}{z(1-z)}$$

Final-state shower: $m_b = m_c = 0 \Rightarrow m_a^2 = \frac{p_{\perp}^2}{z(1-z)} > 0 \Rightarrow$ timelike

Initial-state shower: $m_a = m_c = 0 \Rightarrow m_b^2 = -\frac{p_{\perp}^2}{1-z} < 0 \Rightarrow$ spacelike

Initial & Final State Radiation

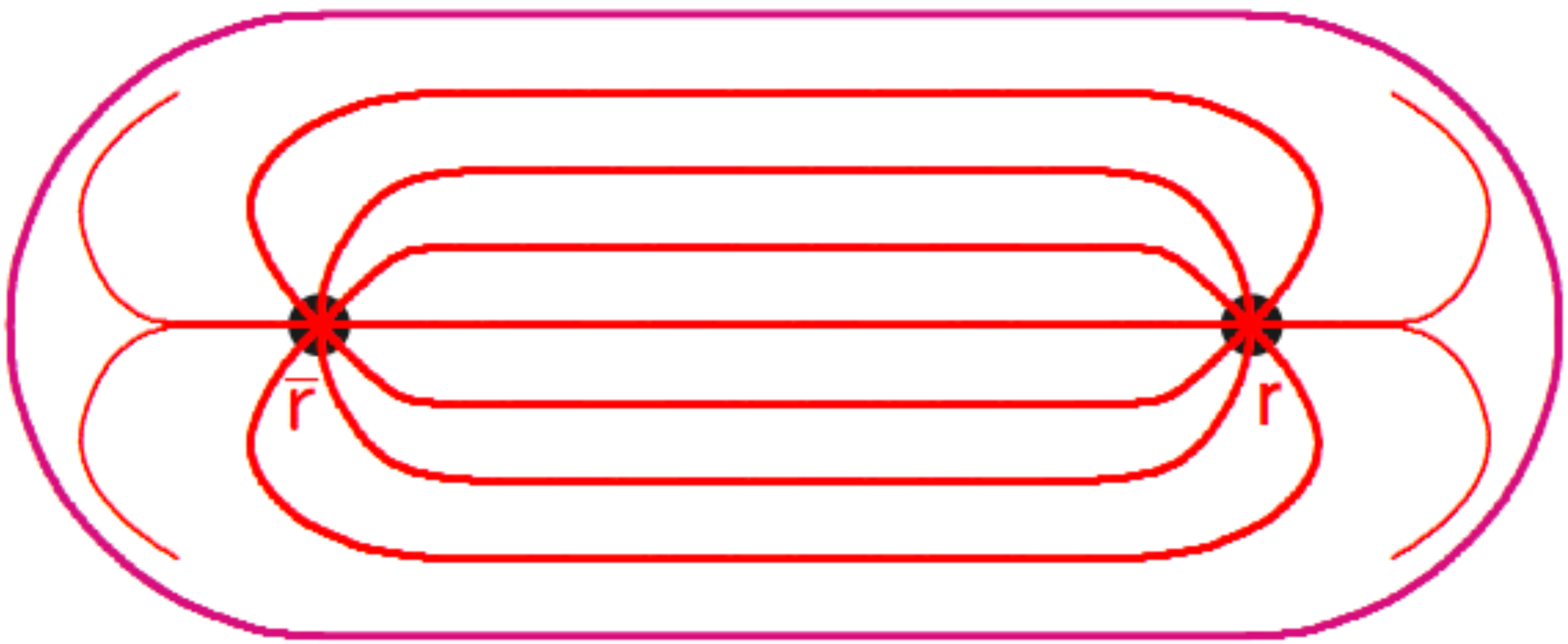


$$P_{\mathrm{q}\to\mathrm{qg}}(z) = C_F \frac{1+z^2}{1-z} \ ,$$
$$P_{\mathrm{g}\to\mathrm{gg}}(z) = C_A \frac{(1-z(1-z))^2}{z(1-z)} \ ,$$
$$P_{\mathrm{g}\to\mathrm{q}\bar{\mathrm{q}}}(z) = T_R \, (z^2 + (1-z)^2) \ ,$$

$$\frac{\mathrm{d}\mathcal{P}_{\mathrm{FSR}}}{\mathrm{d}p_\perp^2} = \frac{1}{p_\perp^2} \int \mathrm{d}z \, \frac{\alpha_{\mathrm{s}}}{2\pi} \, P(z) \ ,$$
$$\frac{\mathrm{d}\mathcal{P}_{\mathrm{ISR}}}{\mathrm{d}p_\perp^2} = \frac{1}{p_\perp^2} \int \mathrm{d}z \, \frac{\alpha_{\mathrm{s}}}{2\pi} \, P(z) \, \frac{f'(x/z, p_\perp^2)}{z f(x, p_\perp^2)} \ ,$$

The QCD potential – 1

In QCD, for large charge separation, field lines are believed to be compressed to tubelike region(s) $\Rightarrow$ **string(s)**



Gives force/potential between a q and a $\overline{\mathrm{q}}$:

$F(r) \approx \mathrm{const} = \kappa \iff V(r) \approx \kappa r$

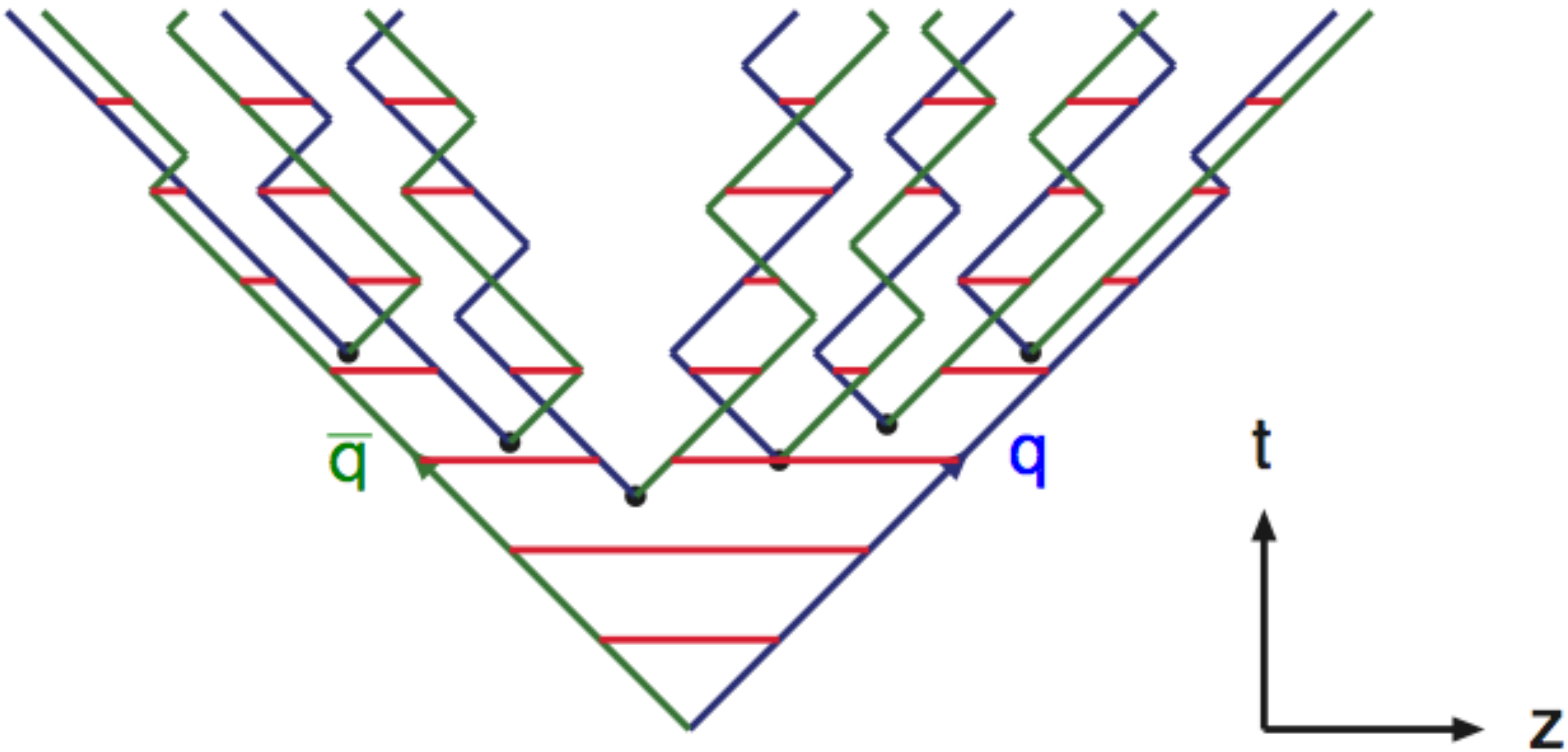
$\kappa \approx 1\ \mathrm{GeV/fm} \approx$ potential energy gain lifting a 16 ton truck.

Flux tube parametrized by center location as a function of time $\Rightarrow$ simple description as a 1+1-dimensional object – a string.

The Lund Model

Combine yo-yo-style string motion with string breakings!

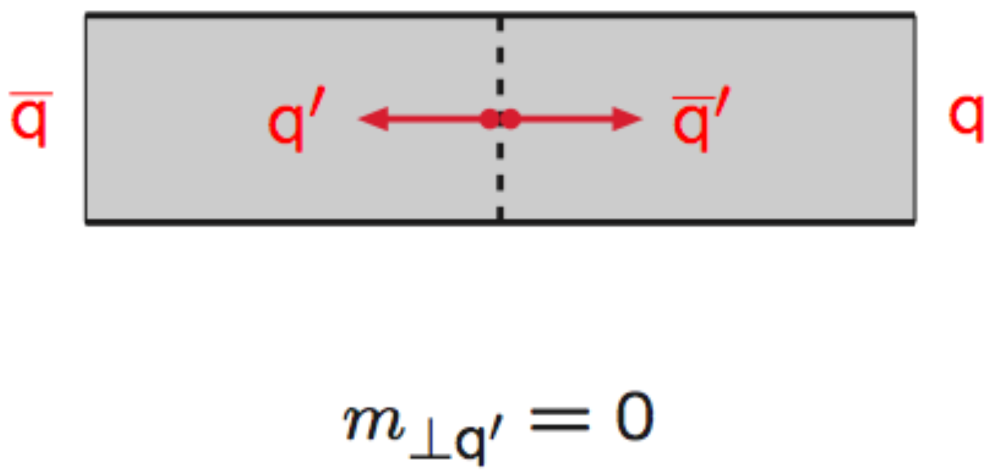
Motion of quarks and antiquarks with intermediate string pieces:

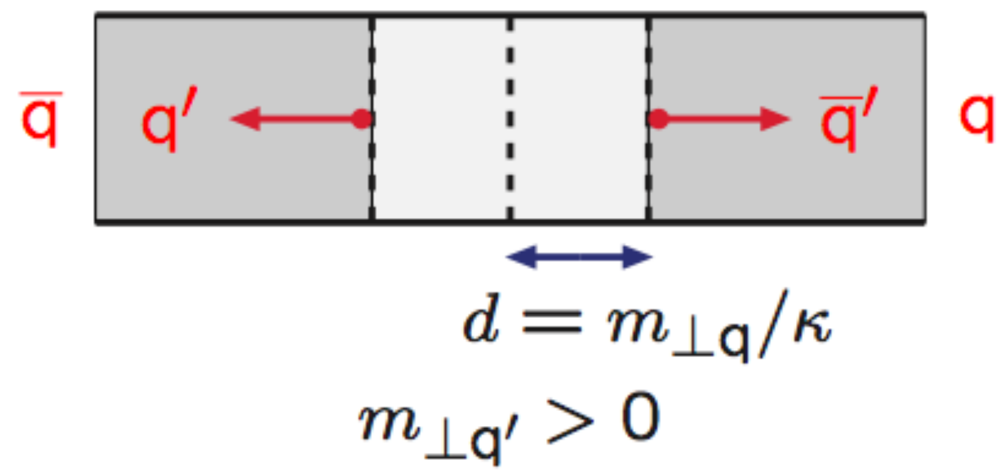


A q from one string break combines with a $\overline{q}$ from an adjacent one.

Gives simple but powerful picture of hadron production.

How does the string break?





String breaking modelled by tunneling:

$$\mathcal{P} \propto \exp\left(-\frac{\pi m_{\perp q}^2}{\kappa}\right) = \exp\left(-\frac{\pi p_{\perp q}^2}{\kappa}\right) \exp\left(-\frac{\pi m_q^2}{\kappa}\right)$$

- Common Gaussian $p_\perp$ spectrum, $\langle p_\perp \rangle \approx 0.4$ GeV.
- Suppression of heavy quarks, $\mathrm{u\overline{u}} : \mathrm{d\overline{d}} : \mathrm{s\overline{s}} : \mathrm{c\overline{c}} \approx 1 : 1 : 0.3 : 10^{-11}$.
- Diquark $\sim$ antiquark $\Rightarrow$ simple model for baryon production.

String model unpredictive in understanding of hadron mass effects $\Rightarrow$ many parameters, 10–20 depending on how you count.

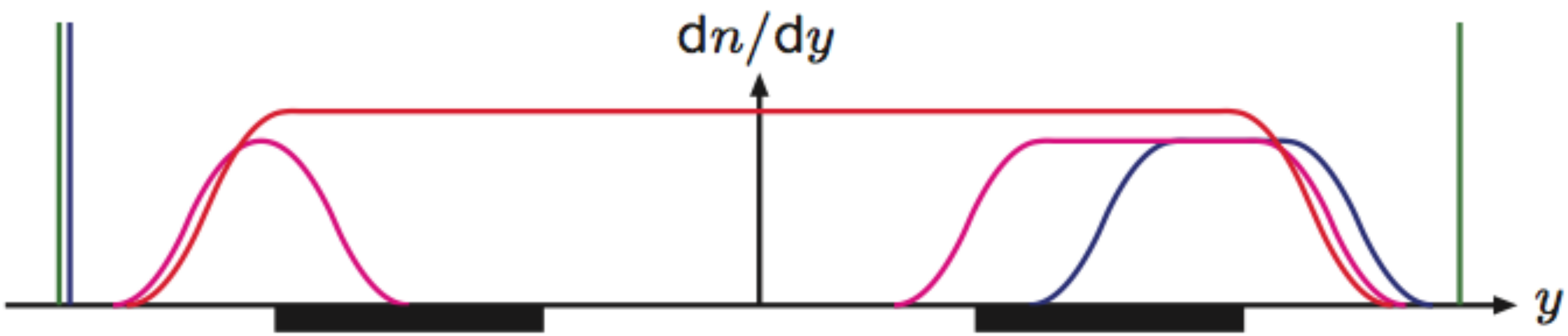
What is minimum bias (MB)?

MB ≈ "all events, with no bias from restricted trigger conditions"

$\sigma_{\mathrm{tot}} =$

$\sigma_{\mathrm{elastic}} + \sigma_{\mathrm{single-diffractive}} + \sigma_{\mathrm{double-diffractive}} + \cdots + \sigma_{\mathrm{non-diffractive}}$

Schematically:

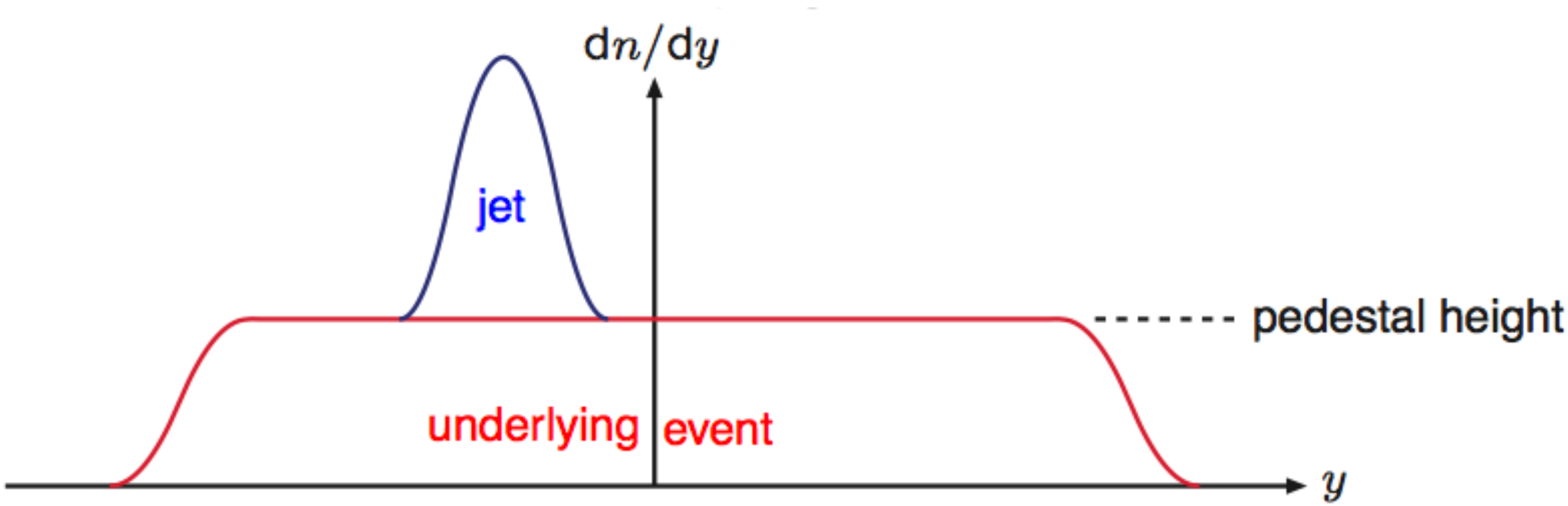


Reality: can only observe events with particles in central detector:

no universally accepted, detector-independent definition

$\sigma_{\mathrm{min-bias}} \approx \sigma_{\mathrm{non-diffractive}} + \sigma_{\mathrm{double-diffractive}} \approx 2/3 \times \sigma_{\mathrm{tot}}$

What is underlying event (UE)?

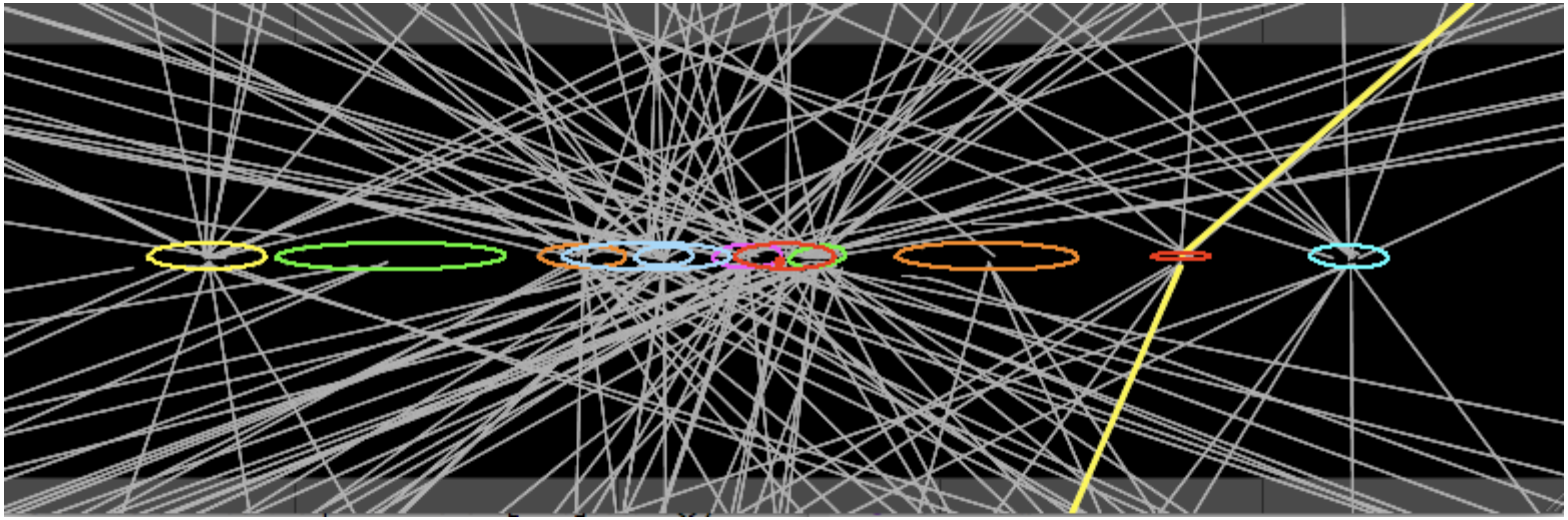


In an event containing a jet pair or another hard process, how much further activity is there, that does not have its origin in the hard process itself, but in other physics processes?

Pedestal effect: the UE contains more activity than a normal MB event does (even discarding diffractive events).

Trigger bias: a jet "trigger" criterion $E_{\perp\mathrm{jet}} > E_{\perp\mathrm{min}}$ is more easily fulfilled in events with upwards-fluctuating UE activity, since the UE $E_{\perp}$ in the jet cone counts towards the $E_{\perp\mathrm{jet}}$. *Not enough!*

What is pileup?



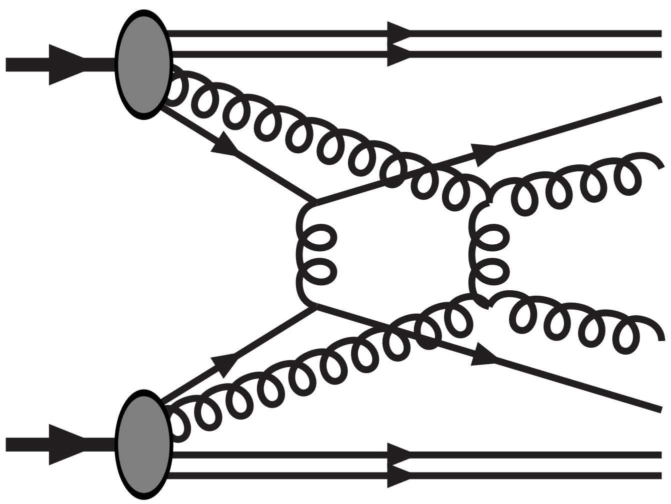
$\langle n \rangle = \overline{\mathcal{L}}\,\sigma$

where $\overline{\mathcal{L}}$ is machine luminosity per bunch crossing, $\overline{\mathcal{L}} \sim n_1 n_2 / A$
and $\sigma \sim \sigma_{\mathrm{tot}} \approx 100$ mb.
Current LHC machine conditions $\Rightarrow \langle n \rangle \sim 10 - 20$.

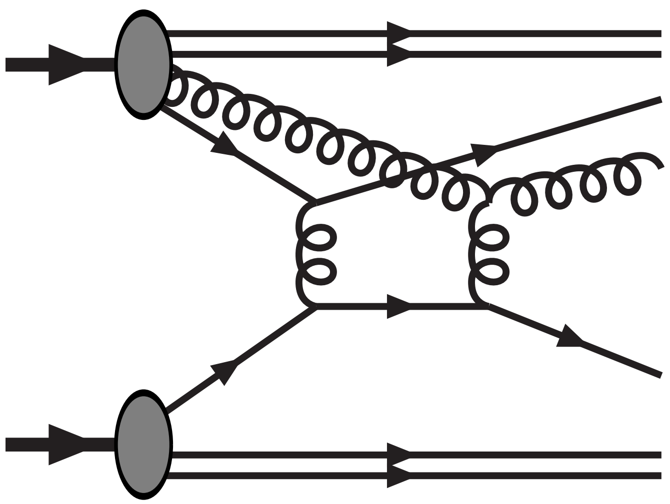
Pileup introduces no new physics, and is thus not further considered here, but can be a nuisance.
However, keep in mind concept of bunches of hadrons leading to multiple collisions.

Rescattering

Often assume that MPI =



. . . but should also include



Same order in α_s, $\sim$ same propagators, but

- one PDF weight less $\Rightarrow$ smaller σ
- one jet less $\Rightarrow$ QCD radiation background $2 \rightarrow 3$ larger than $2 \rightarrow 4$

$\Rightarrow$ will be tough to find direct evidence.

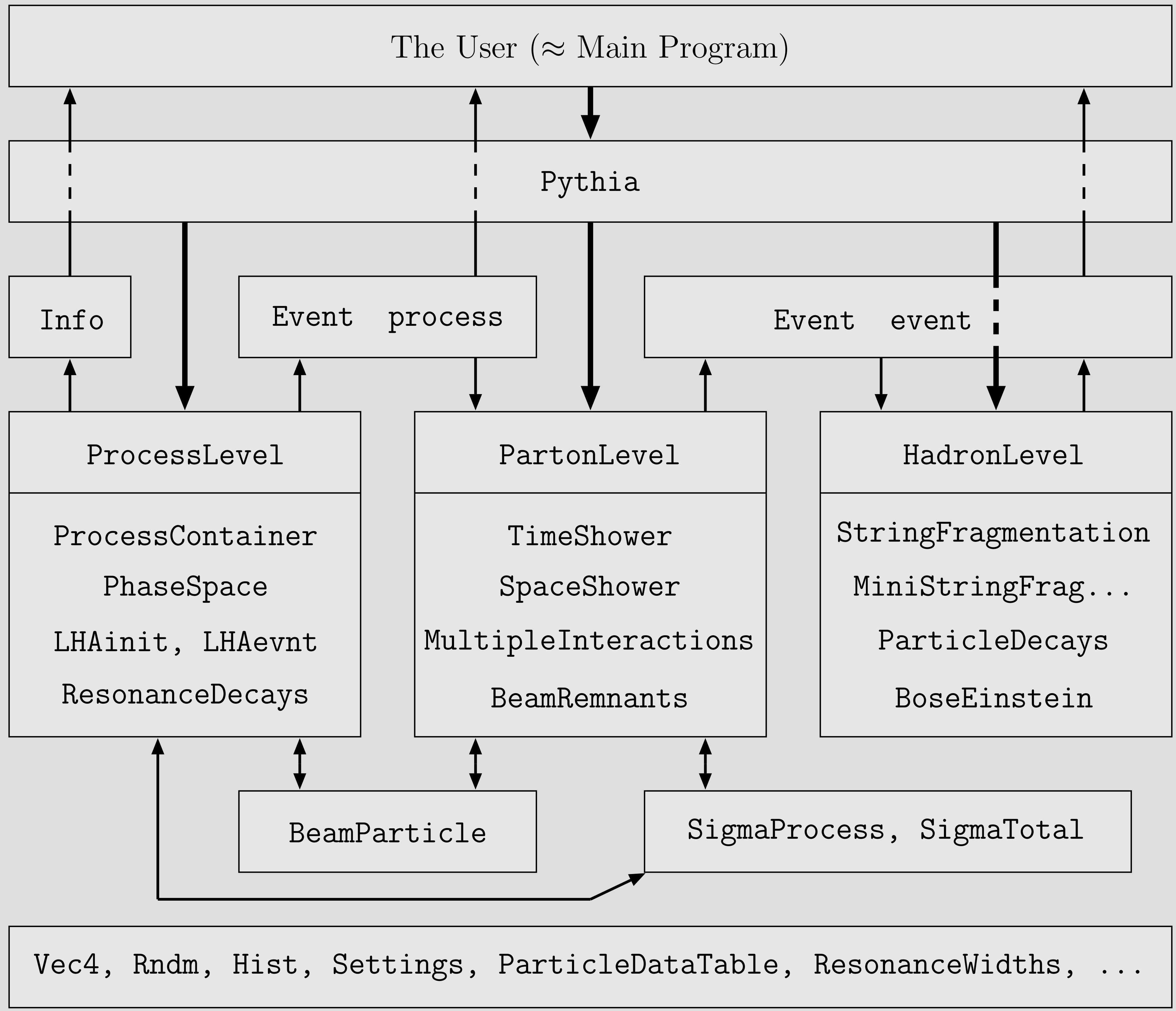
Rescattering grows with number of “previous” scatterings:

	Tevatron		LHC	
	Min Bias	QCD Jets	Min Bias	QCD Jets
Normal scattering	2.81	5.09	5.19	12.19
Single rescatterings	0.41	1.32	1.03	4.10
Double rescatterings	0.01	0.04	0.03	0.15

The anatomy to a Pythia8 program

Inside the BLACK BOX

Inside the BLACK BOX



To Begin:

1. Tell your program to use the Pythia libraries

```
#include "Pythia.h"
using namespace Pythia8;
```

2. Declare a Pythia object and initialise

```
Pythia pythia;
pythia.readString(string);
pythia.readFile(fileName);
```

```
pythia.init(idA, idB, eCM);
```

3. To generate the next event

```
pythia.next();
```

4. All the particles in the event are stored in

```
pythia.event.
```

5. At the end

```
pythia.statistics();
```

The `event.list()` listing provides the main properties of each particles, by column:

- `no`, the index number of the particle (`i` above);
- `id`, the PDG particle identity code (method `id()`);
- `name`, a plaintext rendering of the particle name (method `name()`), within brackets for initial or intermediate particles and without for final-state ones;
- `status`, the reason why a new particle was added to the event record (method `status()`);
- `mothers` and `daughters`, documentation on the event history (methods `mother1()`, `mother2()`, `daughter1()` and `daughter2()`);
- `colours`, the colour flow of the process (methods `col()` and `acol()`);
- `p_x`, `p_y`, `p_z` and `e`, the components of the momentum four-vector (p_x, p_y, p_z, E), in units of GeV with $c = 1$ (methods `px()`, `py()`, `pz()` and `e()`);
- `m`, the mass, in units as above (method `m()`).

1	2212	(p+)	-12	0	0	3	0	0	0	0.000	0.000	7000.000	7000.000	0.938
2	2212	(p+)	-12	0	0	4	0	0	0	0.000	0.000	-7000.000	7000.000	0.938
3	21	(g)	-21	1	0	5	6	503	502	0.000	0.000	122.026	122.026	0.000
4	21	(g)	-21	2	0	5	6	501	503	0.000	0.000	-441.322	441.322	0.000
5	6	(t)	-22	3	4	7	8	501	0	-91.828	87.872	-266.639	342.315	173.000
6	-6	(tbar)	-22	3	4	9	10	0	502	91.828	-87.872	-52.657	221.033	173.000

Example card file

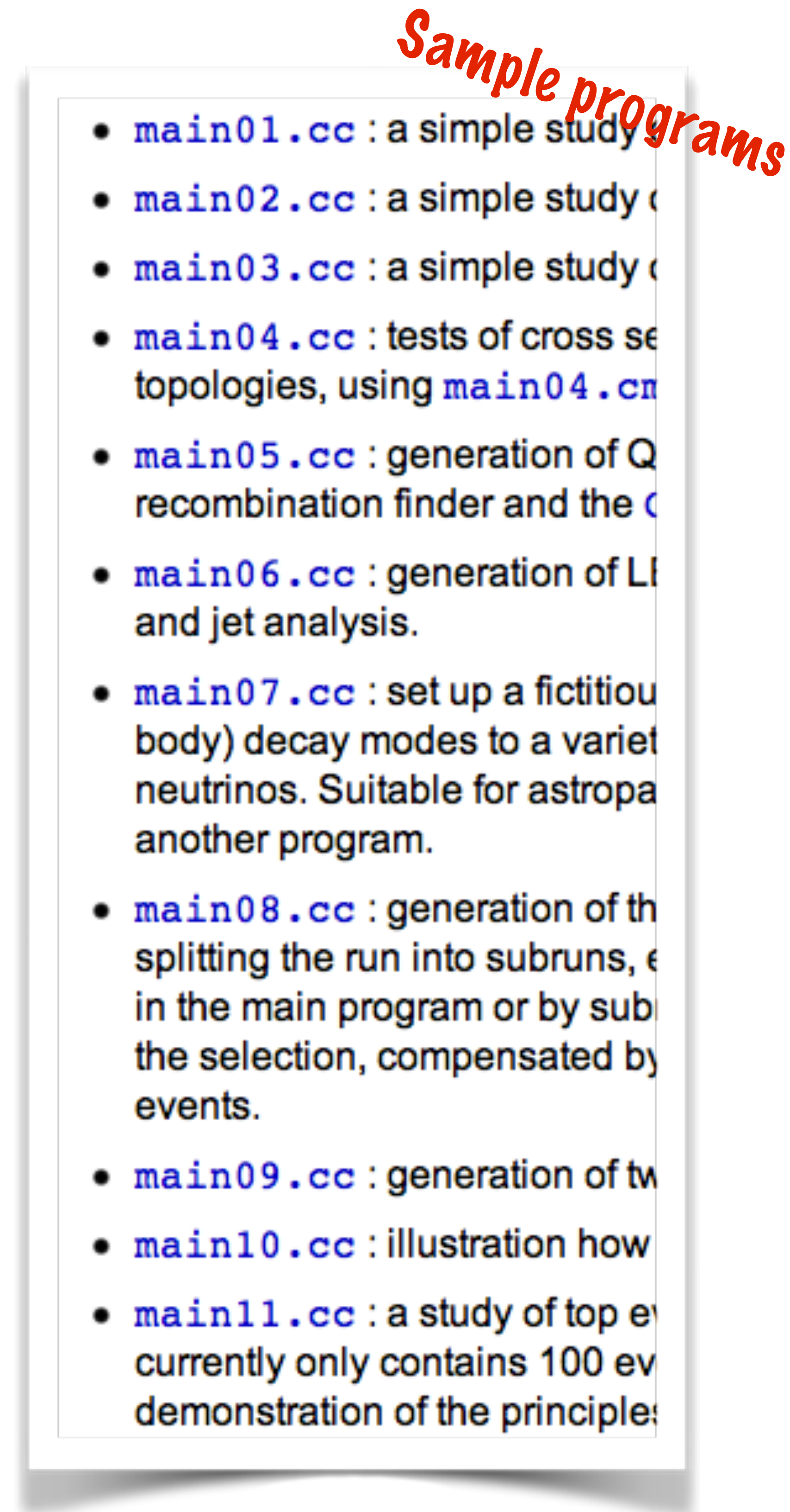
```
! 1) Settings that are used in the main program.
Main:numberOfEvents = 1000          ! number of events to generate
Main:timesAllowErrors = 10          ! abort run after this many flawed events

! 2) Settings related to output in init(), next() and stat().
Init:showChangedSettings = on       ! list changed settings
Init:showChangedParticleData = on   ! list changed particle data
#Init:showChangedResonanceData = on ! also print changed resonance data
Init:showOneParticleData = 25       ! print data for this particular particle
Next:numberCount = 100              ! print message every n events
Next:numberShowInfo = 1             ! print event information n times
Next:numberShowProcess = 1          ! print process record n times
Next:numberShowEvent = 0            ! print event record n times
Stat:showPartonLevel = on           ! more statistics on MPI

! 3) Beam settings.
Beams:idA = 2212                    ! first beam, p = 2212, pbar = -2212
Beams:idB = 2212                    ! second beam, p = 2212, pbar = -2212
Beams:eCM = 14000.                  ! CM energy of collision, LHC

! 4) Settings for hard-process generation internal to Pythia8.
HiggsSM:gg2H = on                   ! Higgs production by gluon-gluon fusion
25:m0 = 125.4                       ! Higgs mass

! 5) Switch off some key components of the simulation, for comparisons.
#PartonLevel:all = off               ! stop after hard process
#PartonLevel:MPI = off               ! no multiparton interactions
#PartonLevel:ISR = off               ! no initial-state radiation
#PartonLevel:FSR = off               ! no final-state radiation
#HadronLevel:all = off               ! stop after parton level
#HadronLevel:Hadronize = off         ! no hadronization
```



How do you set the parameters?

Extra Gauge Bosons

NewGaugeBoson:ffbar2gmZZprime = on
Scattering *f fbar ->Z'^0*. [..]

`mode` **Zprime:gmZmode** (`default = 0`; `minimum = 0`; `maximum = 6`)
Choice of full *gamma^*/Z^0/Z'^0* structure or not in the above process. [...]
`option` **0** : full *gamma^*/Z^0/Z'^0* structure, with interference included.
`option` **1** : only pure *gamma^** contribution.
`option` **2** : only pure *Z^0* contribution.
`option` **3** : only pure *Z'^0* contribution.
`option` **4** : only the *gamma^*/Z^0* contribution, including interference.
`option` **5** : only the *gamma^*/Z'^0* contribution, including interference.
`option` **6** : only the *Z^0/Z'^0* contribution, including interference.

Can be turned on with full interference with gamma/Z

`flag` **Zprime:universality** (`default = on`)
If on then you need only set the first-generation couplings below, and these are automatically also used for the second and third generation. If off, then couplings can be chosen separately for each generation.

`parm` **Zprime:vd** (`default = -0.693`)
vector coupling of *d* quarks.

`parm` **Zprime:ad** (`default = -1.`)
axial coupling of *d* quarks.

`parm` **Zprime:vu** (`default = 0.387`)
vector coupling of *u* quarks.

`parm` **Zprime:au** (`default = 1.`)
axial coupling of *u* quarks.

Versatile assignment of couplings

Improvements to SLHA interface

Was designed for SUSY

```
BLOCK MODSEL  # Model selection
     1     1   sugra
#
BLOCK SMINPUTS  # Standard Model inputs
     1     1.27934000E+02   # alpha_em^-1(M_Z)^MSbar
     2     1.16637000E-05   # G_F [GeV^-2]
     3     1.18000000E-01   # alpha_S(M_Z)^MSbar
     4     9.11876000E+01   # M_Z pole mass
     5     4.25000000E+00   # mb(mb)^MSbar
     6     1.75000000E+02   # mt pole mass
     7     1.77700000E+00   # mtau pole mass
#
BLOCK MINPAR  # Input parameters - minimal models
     1     1.00000000E+02   # m0
     2     2.50000000E+02   # m12
     3     1.00000000E+01   # tanb
     4     1.00000000E+00   # sign(mu)
     5    -1.00000000E+02   # A0
```

Problem with designing a generic interface for all BSM models is how to implement arbitrary blocks particle ID   particle name

```
BLOCK QNUMBERS 7654321 # balleron
      1      0  # 3 times electric charge
      2      2  # number of spin states (2S+1)
      3      8  # colour rep (1: singlet, 3: triplet, 6: sextet, 8: octet)
      4      0  # Particle/Antiparticle distinction (0=own anti)
```

particle name   anti-particle name

```
BLOCK QNUMBERS 8765432 # yup yupbar
      1      2  # 3 times electric charge
      2      2  # number of spin states (2S+1)
      3      3  # colour rep (1: singlet, 3: triplet, 6: sextet, 8: octet)
      4      1  # Particle/Antiparticle distinction (0=own anti)

 BLOCK MASS
 #     ID code  pole mass in GeV
       7654321   800.0  # m(balleron)
       8765432   600.0  # m(yup)

 #           ID      WIDTH in GeV
 DECAY   7654321  2.034369169E+00  # balleron decays
 #     BR                  NDA       ID1       ID2       ID3
      9.900000000E-01     3         6          5         3    # BR( -> t b s )
      1.000000000E-02     3         4          5         3    # BR( -> c b s )
```

Use either a **semi-internal process** (your own derived subclass of a Pythia process) to provide production cross section expressions or **read in LHE file** generated externally.

What if you need extra parameters (blocks)?

➡Pythia provides functionality to retrieve data from arbitrarily named blocks

```
bool slhaPtr->getEntry(string blockName, double& val);
bool slhaPtr->getEntry(string blockName, int indx, double& val);
bool slhaPtr->getEntry(string blockName, int indx, int jndx, double& val);
bool slhaPtr->getEntry(string blockName, int indx, int jndx, int kndx, double& val);
```