LUND UNIVERSITY

Monte Carlo School
Dresden
6 January 2006

PYTHIA 8 — The First Release

Torbjörn Sjöstrand

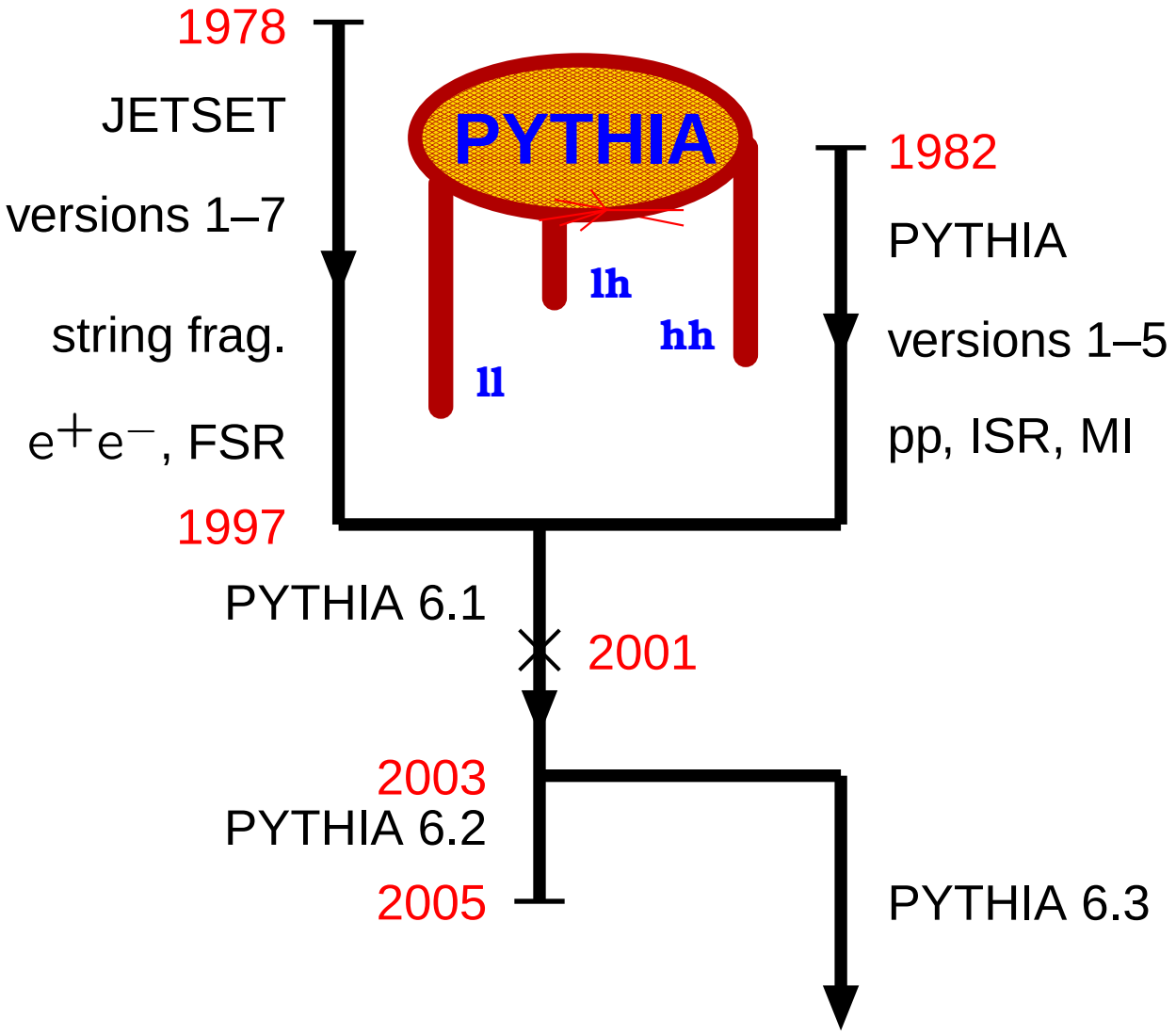
Department of Theoretical Physics, Lund University

PYTHIA 8.040 released on 20 July 2005

- What is in it?
- How to use it?

PYTHIA 8.041: modest updates for this meeting

PYTHIA history



All in Fortran 77

Time axis
not to scale

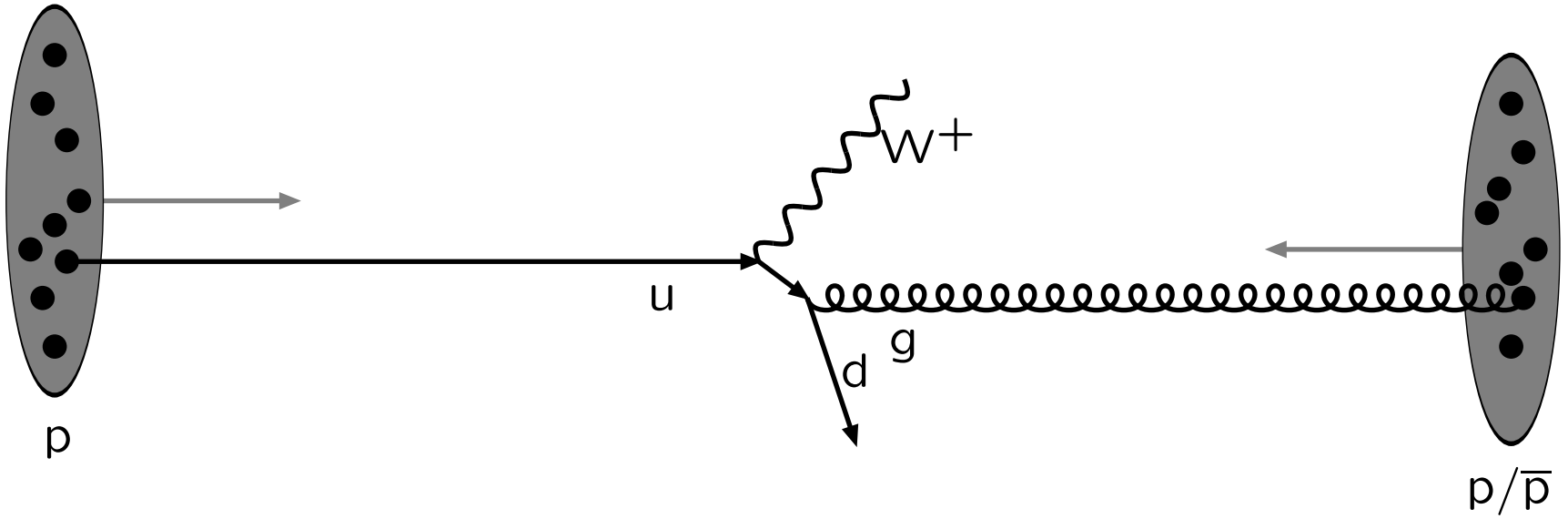
PYTHIA standalone,
but other programs
rely on PYTHIA:

LEPTO
ARIADNE/LDC
RAPGAP/CASCADE
POMPYT
HIJING
SHERPA
EVTGEN
. . .

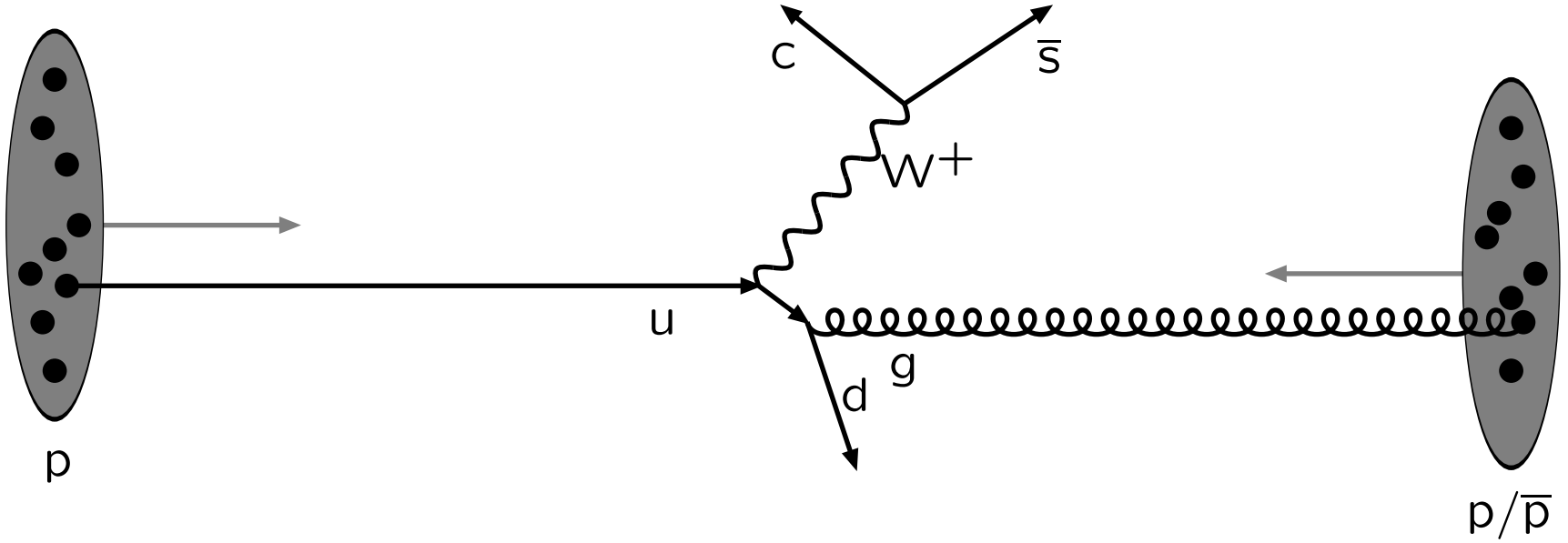
The structure of an event

Warning: schematic only, everything simplified, nothing to scale, . . .

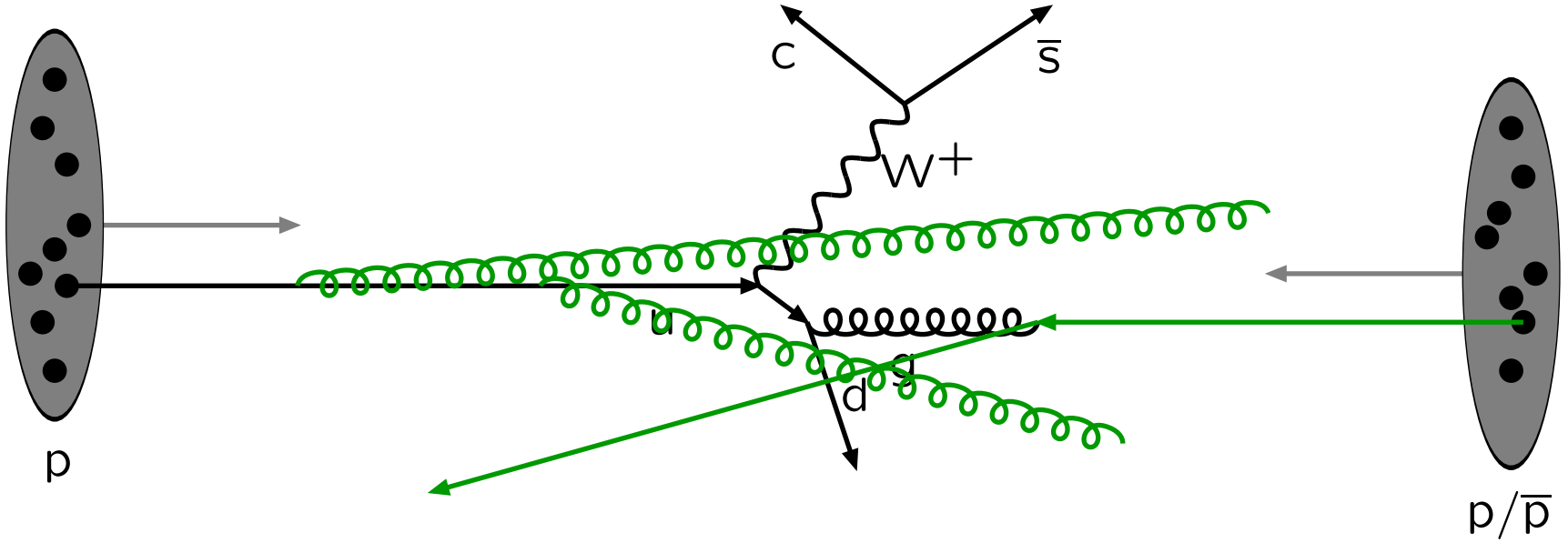
Incoming beams: parton densities



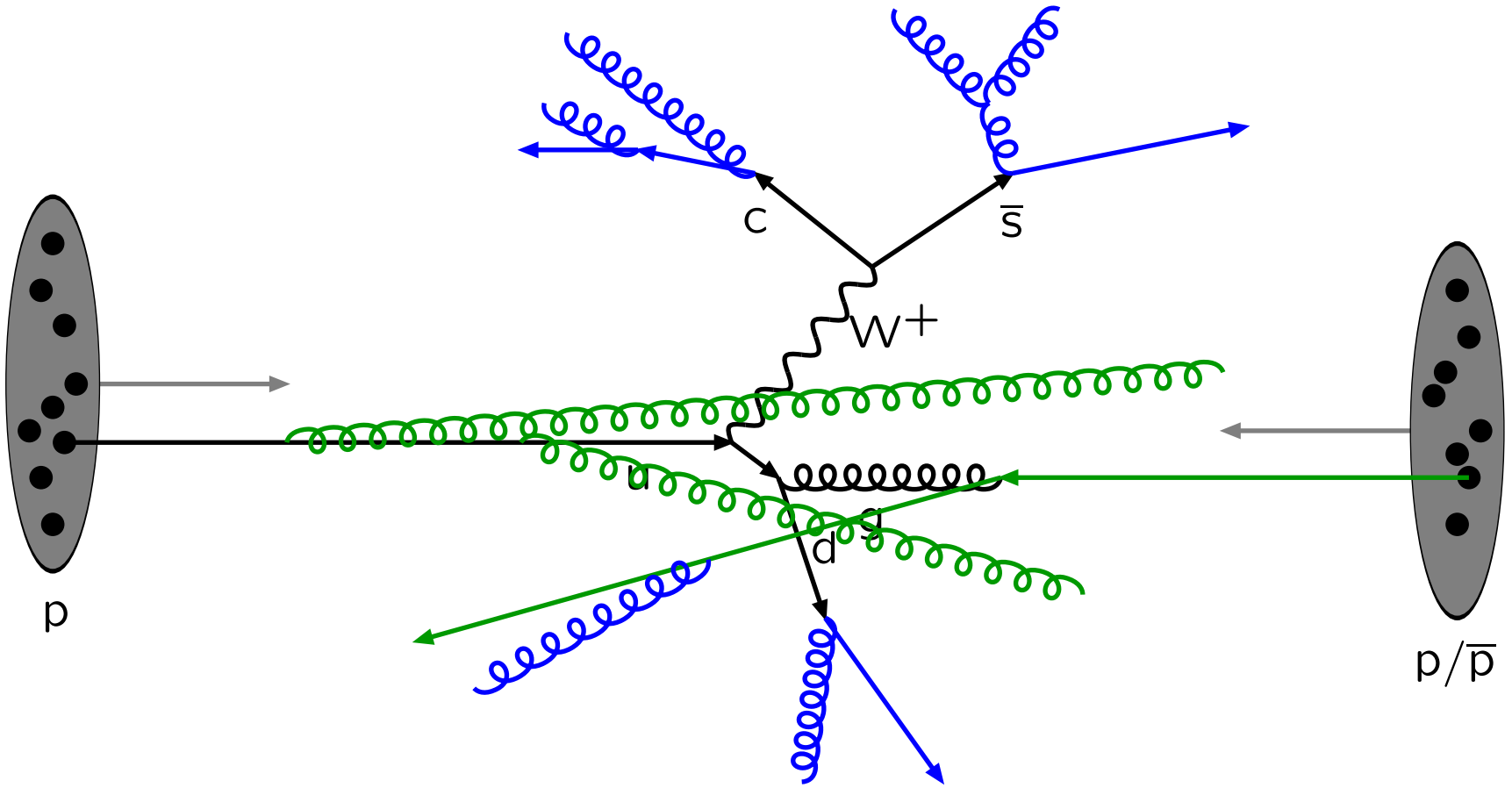
Hard subprocess: described by matrix elements



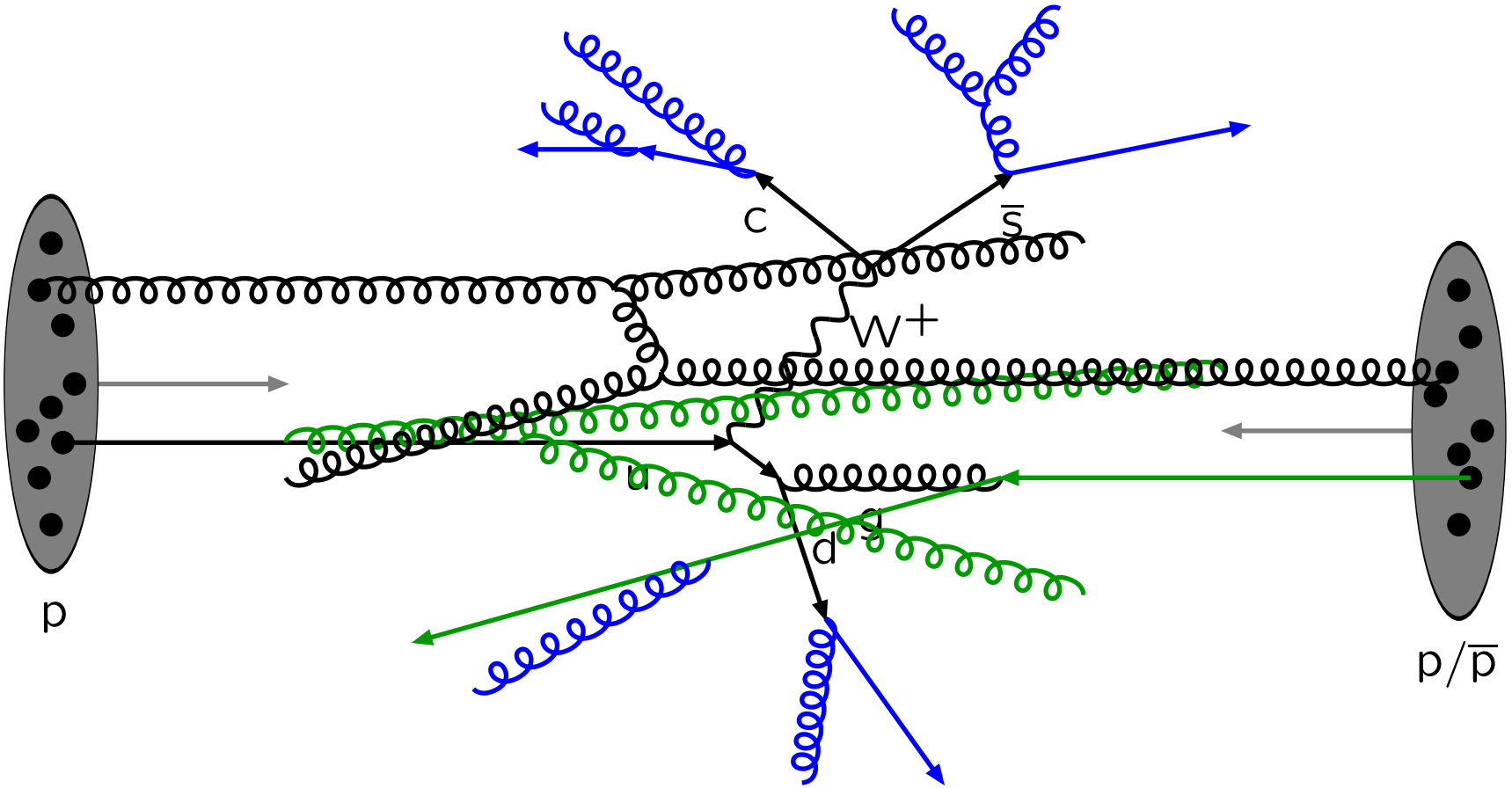
Resonance decays: correlated with hard subprocess



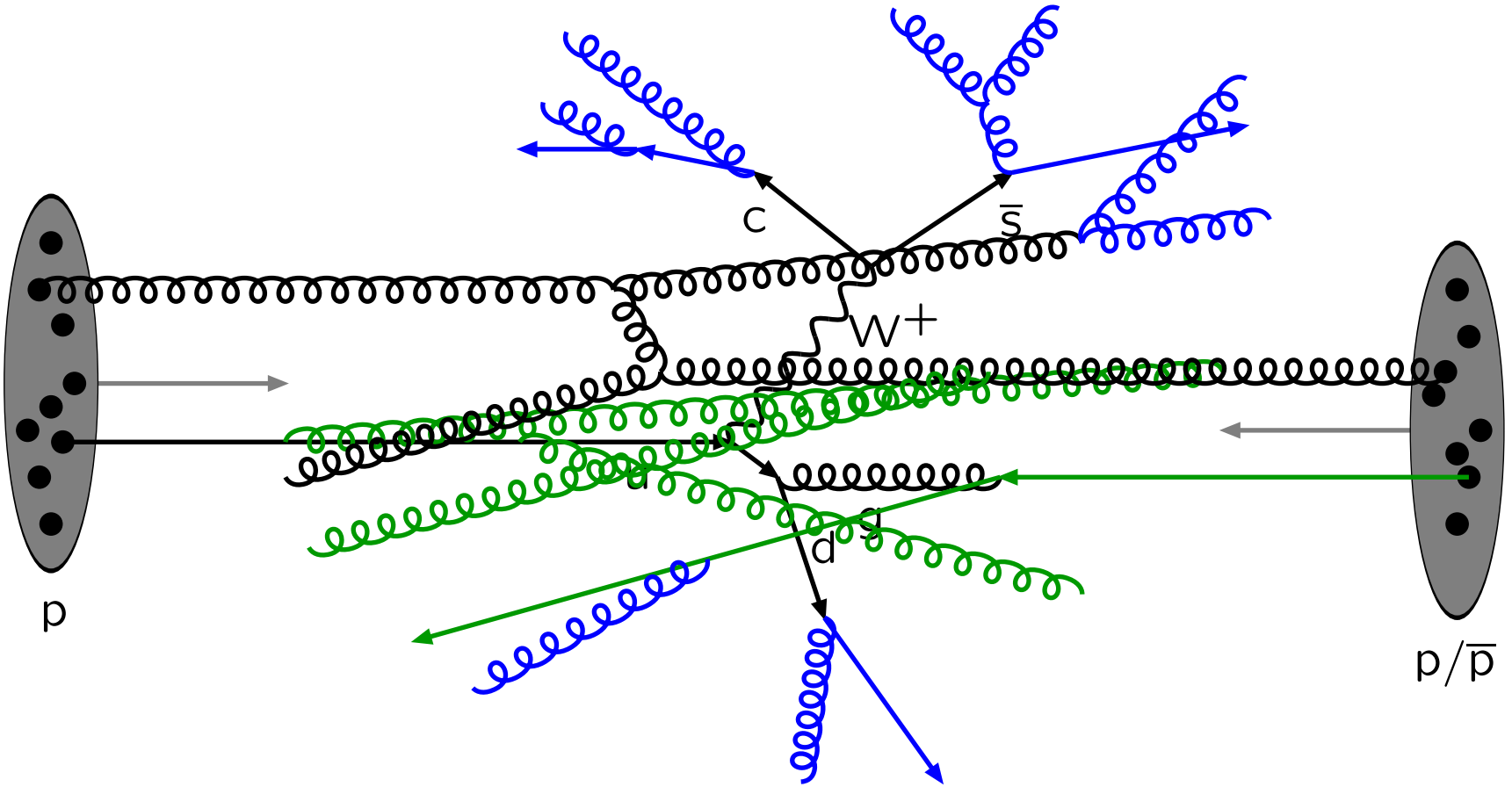
Initial-state radiation: spacelike parton showers



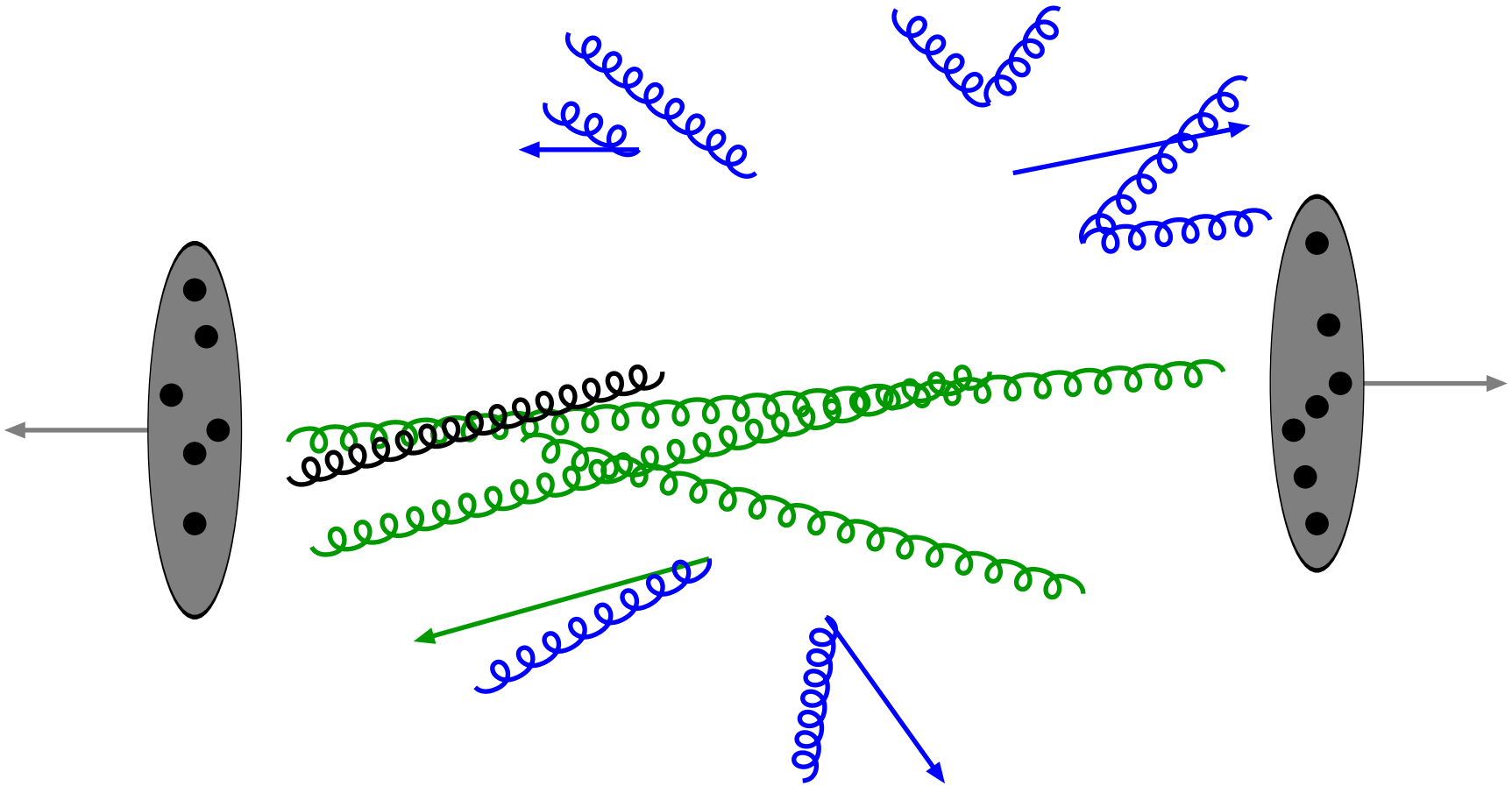
Final-state radiation: timelike parton showers



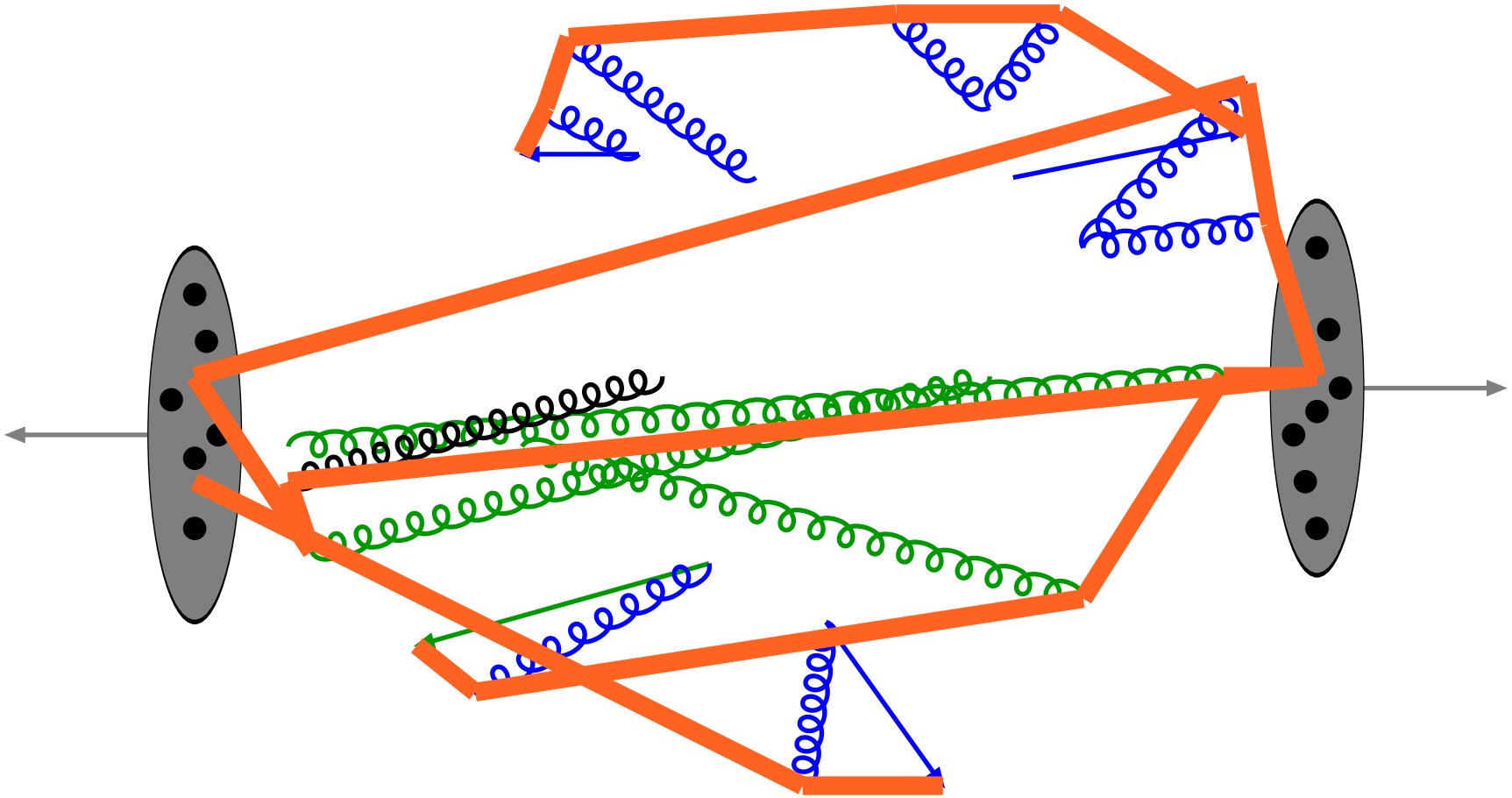
Multiple parton–parton interactions . . .



. . . with its initial- and final-state radiation

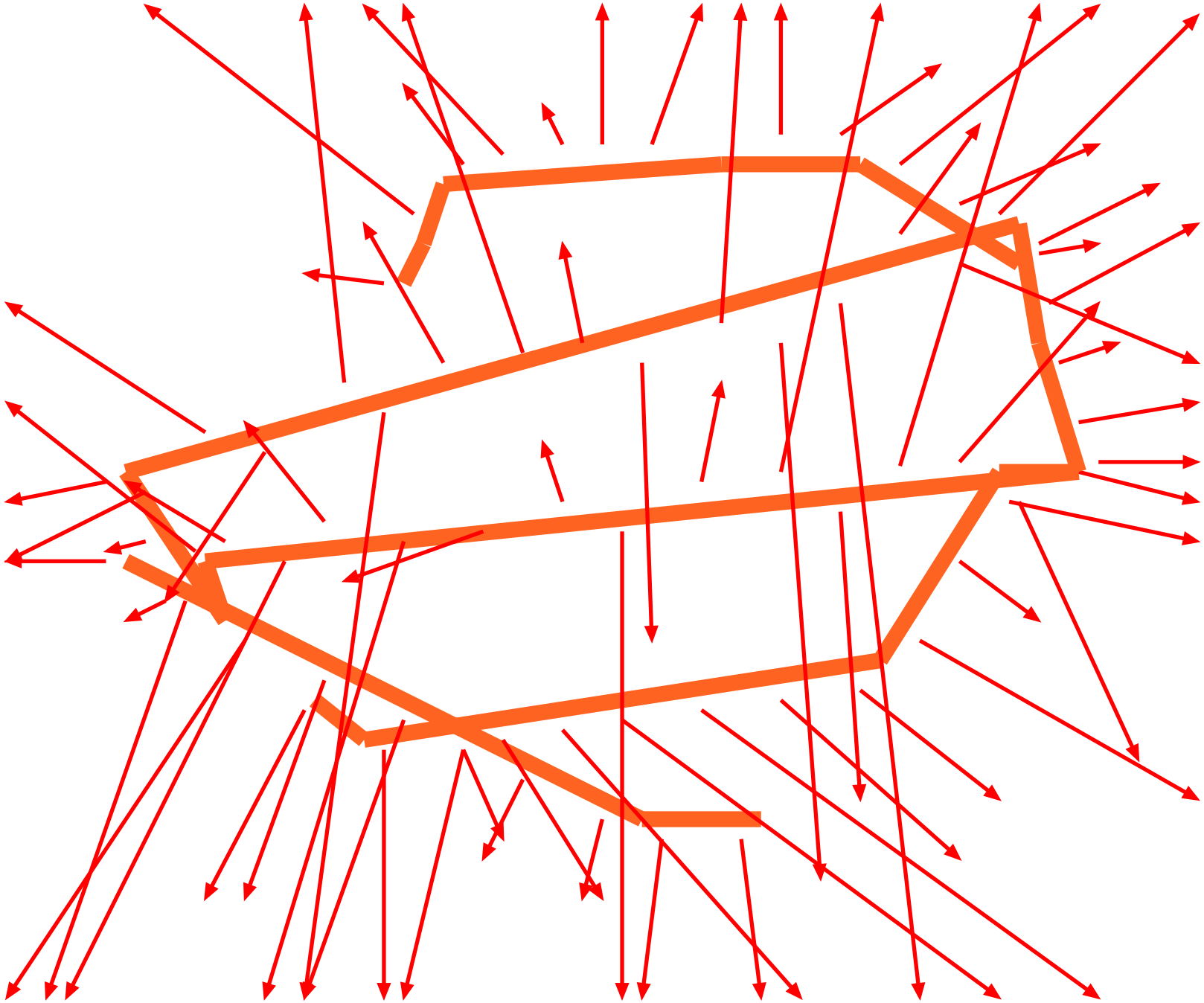


Beam remnants and other outgoing partons

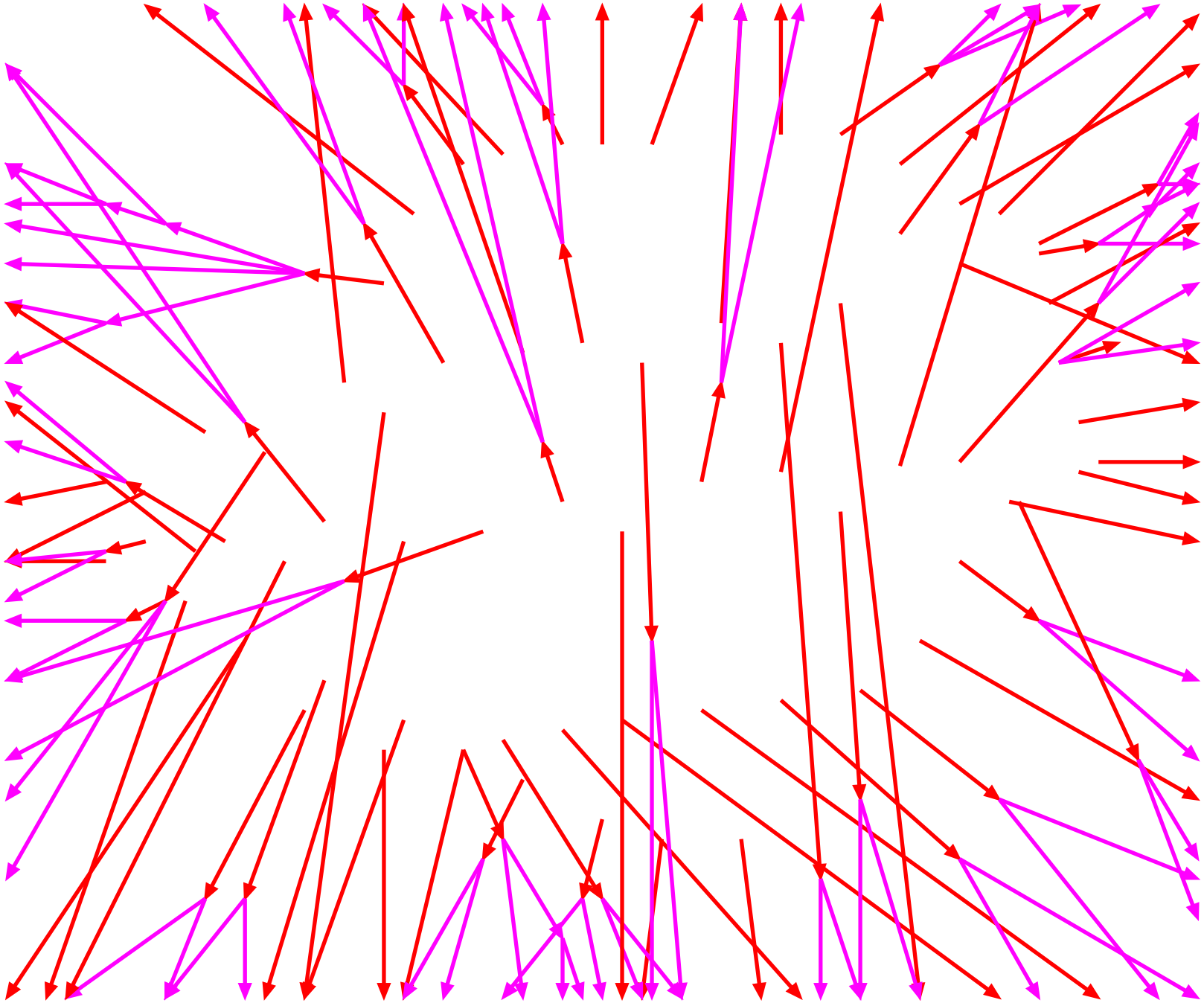


Everything is connected by colour confinement strings

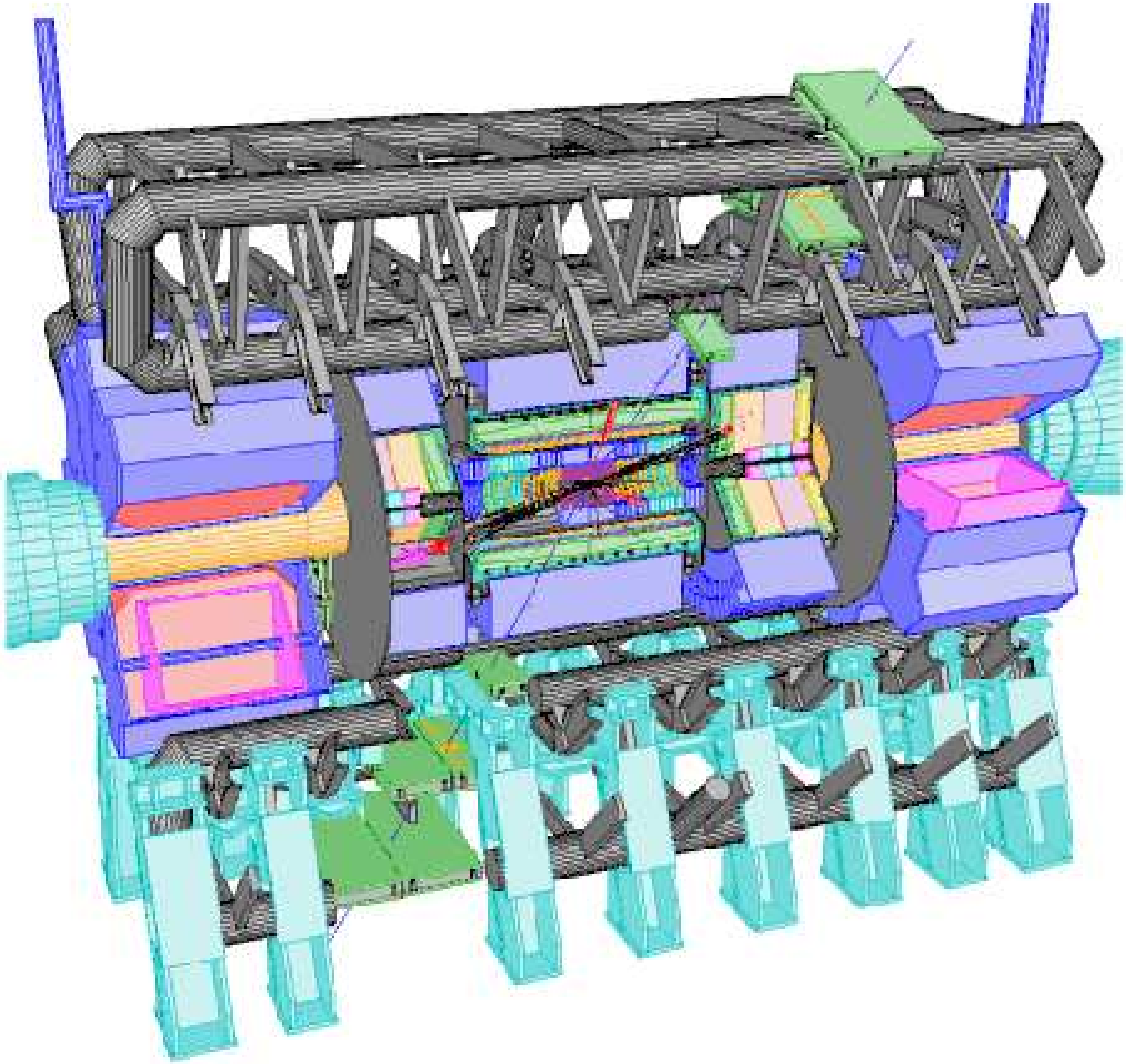
Recall! Not to scale: strings are of hadronic widths



The strings fragment to produce primary hadrons



Many hadrons are unstable and decay further



These are the particles that hit the detector

PYTHIA 6 status

PYTHIA 6 is a general-purpose generator:

- large subprocess library
- virtuality-ordered initial- and final-state showers
- 'original' multiple interactions models
- 'one and only' string fragmentation implementation
- lots of utilities

Currently PYTHIA 6.325:

- 73,000 lines of code (including comments/blanks)
- 450 page manual (35,000 lines input)
- 2,000 lines long update notes
- available on `http://www.thep.lu.se/~torbjorn/Pythia.html`
- together with sample main programs, old code, etc.

PYTHIA 6.400 in preparation:

- final Fortran main version (?)
- updated manual (to be submitted to JHEP)

PYTHIA Process Library

No.	Subprocess
Hard QCD processes:	
11	$f_i f_j \to f_i f_j$
12	$f_i \overline{f}_i \to f_k \overline{f}_k$
13	$f_i \overline{f}_i \to gg$
28	$f_i g \to f_i g$
53	$gg \to f_k \overline{f}_k$
68	$gg \to gg$
Soft QCD processes:	
91	elastic scattering
92	single diffraction (XB)
93	single diffraction (AX)
94	double diffraction
95	low-$p_\perp$ production
Open heavy flavour: (also fourth generation)	
81	$f_i \overline{f}_i \to Q_k \overline{Q}_k$
82	$gg \to Q_k \overline{Q}_k$
83	$q_i f_j \to Q_k f_l$
84	$g\gamma \to Q_k \overline{Q}_k$
85	$\gamma\gamma \to F_k \overline{F}_k$
Closed heavy flavour:	
86	$gg \to J/\psi g$
87	$gg \to \chi_{0c} g$
88	$gg \to \chi_{1c} g$
89	$gg \to \chi_{2c} g$
104	$gg \to \chi_{0c}$
105	$gg \to \chi_{2c}$
106	$gg \to J/\psi\gamma$
107	$g\gamma \to J/\psi g$
108	$\gamma\gamma \to J/\psi\gamma$
W/Z production:	
1	$f_i \overline{f}_i \to \gamma^*/Z^0$
2	$f_i \overline{f}_j \to W^\pm$
22	$f_i \overline{f}_i \to Z^0 Z^0$
23	$f_i \overline{f}_j \to Z^0 W^\pm$
25	$f_i \overline{f}_i \to W^+ W^-$
15	$f_i \overline{f}_i \to g Z^0$
16	$f_i \overline{f}_j \to g W^\pm$
30	$f_i g \to f_i Z^0$
31	$f_i g \to f_k W^\pm$
19	$f_i \overline{f}_i \to \gamma Z^0$
20	$f_i \overline{f}_j \to \gamma W^\pm$
35	$f_i \gamma \to f_i Z^0$

No.	Subprocess
36	$f_i \gamma \to f_k W^\pm$
69	$\gamma\gamma \to W^+ W^-$
70	$\gamma W^\pm \to Z^0 W^\pm$
Prompt photons:	
14	$f_i \overline{f}_i \to g\gamma$
18	$f_i \overline{f}_i \to \gamma\gamma$
29	$f_i g \to f_i \gamma$
114	$gg \to \gamma\gamma$
115	$gg \to g\gamma$
Deeply Inel. Scatt.:	
10	$f_i f_j \to f_k f_l$
99	$\gamma^* q \to q$
Photon-induced:	
33	$f_i \gamma \to f_i g$
34	$f_i \gamma \to f_i \gamma$
54	$g\gamma \to f_k \overline{f}_k$
58	$\gamma\gamma \to f_k \overline{f}_k$
131	$f_i \gamma_T^* \to f_i g$
132	$f_i \gamma_L^* \to f_i g$
133	$f_i \gamma_T^* \to f_i \gamma$
134	$f_i \gamma_L^* \to f_i \gamma$
135	$g\gamma_T^* \to f_i \overline{f}_i$
136	$g\gamma_L^* \to f_i \overline{f}_i$
137	$\gamma_T^* \gamma_T^* \to f_i \overline{f}_i$
138	$\gamma_T^* \gamma_L^* \to f_i \overline{f}_i$
139	$\gamma_L^* \gamma_T^* \to f_i \overline{f}_i$
140	$\gamma_L^* \gamma_L^* \to f_i \overline{f}_i$
80	$q_i \gamma \to q_k \pi^\pm$
Light SM Higgs:	
3	$f_i \overline{f}_i \to h^0$
24	$f_i \overline{f}_i \to Z^0 h^0$
26	$f_i \overline{f}_j \to W^\pm h^0$
32	$f_i g \to f_i h^0$
102	$gg \to h^0$
103	$\gamma\gamma \to h^0$
110	$f_i \overline{f}_i \to \gamma h^0$
111	$f_i \overline{f}_i \to g h^0$
112	$f_i g \to f_i h^0$
113	$gg \to g h^0$
121	$gg \to Q_k \overline{Q}_k h^0$
122	$q_i \overline{q}_i \to Q_k \overline{Q}_k h^0$
123	$f_i f_j \to f_i f_j h^0$
124	$f_i f_j \to f_k f_l h^0$

No.	Subprocess
New gauge bosons:	
141	$f_i \overline{f}_i \to \gamma/Z^0/Z'^0$
142	$f_i \overline{f}_j \to W'^+$
144	$f_i \overline{f}_j \to R$
Heavy SM Higgs:	
5	$Z^0 Z^0 \to h^0$
8	$W^+ W^- \to h^0$
71	$Z_L^0 Z_L^0 \to Z_L^0 Z_L^0$
72	$Z_L^0 Z_L^0 \to W_L^+ W_L^-$
73	$Z_L^0 W_L^\pm \to Z_L^0 W_L^\pm$
76	$W_L^+ W_L^- \to Z_L^0 Z_L^0$
77	$W_L^\pm W_L^\pm \to W_L^\pm W_L^\pm$
BSM Neutral Higgs:	
151	$f_i \overline{f}_i \to H^0$
152	$gg \to H^0$
153	$\gamma\gamma \to H^0$
171	$f_i \overline{f}_i \to Z^0 H^0$
172	$f_i \overline{f}_j \to W^\pm H^0$
173	$f_i f_j \to f_i f_j H^0$
174	$f_i f_j \to f_k f_l H^0$
181	$gg \to Q_k \overline{Q}_k H^0$
182	$q_i \overline{q}_i \to Q_k \overline{Q}_k H^0$
183	$f_i \overline{f}_i \to g H^0$
184	$f_i g \to f_i H^0$
185	$gg \to g H^0$
156	$f_i \overline{f}_i \to A^0$
157	$gg \to A^0$
158	$\gamma\gamma \to A^0$
176	$f_i \overline{f}_i \to Z^0 A^0$
177	$f_i \overline{f}_j \to W^\pm A^0$
178	$f_i f_j \to f_i f_j A^0$
179	$f_i f_j \to f_k f_l A^0$
186	$gg \to Q_k \overline{Q}_k A^0$
187	$q_i \overline{q}_i \to Q_k \overline{Q}_k A^0$
188	$f_i \overline{f}_i \to g A^0$
189	$f_i g \to f_i A^0$
190	$gg \to g A^0$
Charged Higgs:	
143	$f_i \overline{f}_j \to H^+$
161	$f_i g \to f_k H^+$
401	$gg \to \overline{t} b H^+$
402	$q\overline{q} \to \overline{t} b H^+$

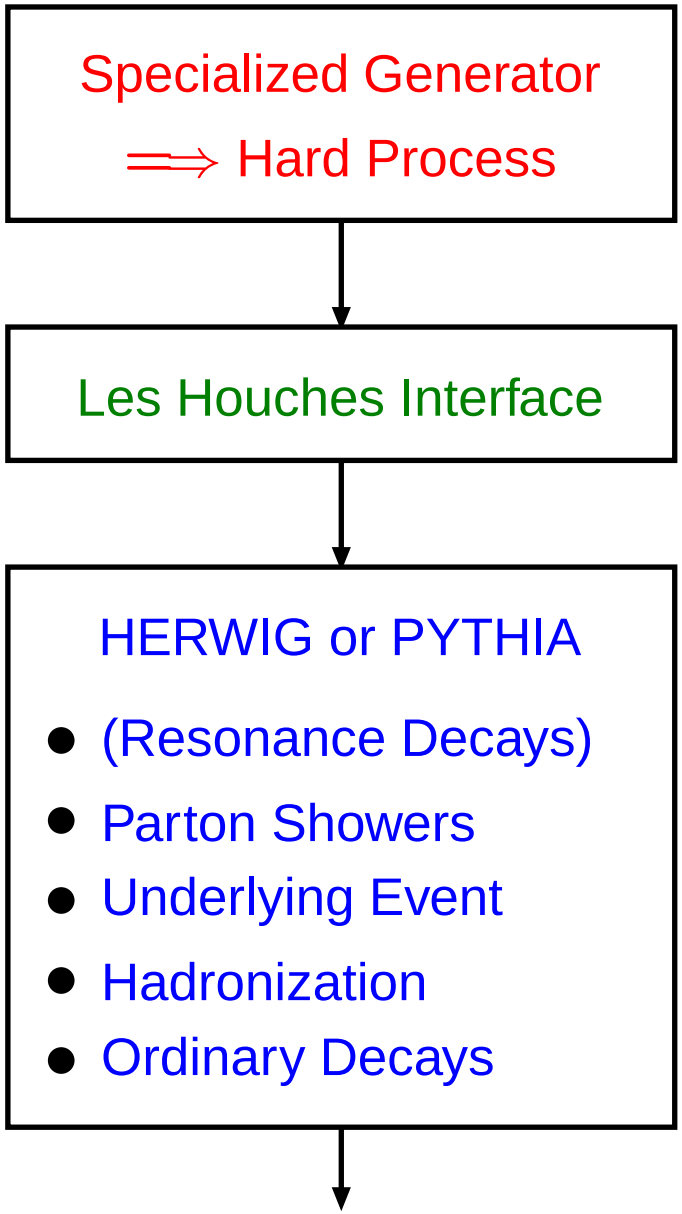
No.	Subprocess
Higgs pairs:	
297	$f_i \overline{f}_j \to H^\pm h^0$
298	$f_i \overline{f}_j \to H^\pm H^0$
299	$f_i \overline{f}_i \to A^0 h^0$
300	$f_i \overline{f}_i \to A^0 H^0$
301	$f_i \overline{f}_i \to H^+ H^-$
Leptoquarks:	
145	$q_i \ell_j \to L_Q$
162	$qg \to \ell L_Q$
163	$gg \to L_Q \overline{L}_Q$
164	$q_i \overline{q}_i \to L_Q \overline{L}_Q$
Technicolor:	
149	$gg \to \eta_{tc}$
191	$f_i \overline{f}_i \to \rho_{tc}^0$
192	$f_i \overline{f}_j \to \rho_{tc}^+$
193	$f_i \overline{f}_i \to \omega_{tc}^0$
194	$f_i \overline{f}_i \to f_k \overline{f}_k$
195	$f_i \overline{f}_j \to f_k \overline{f}_l$
361	$f_i \overline{f}_i \to W_L^+ W_L^-$
362	$f_i \overline{f}_i \to W_L^\pm \pi_{tc}^\mp$
363	$f_i \overline{f}_i \to \pi_{tc}^+ \pi_{tc}^-$
364	$f_i \overline{f}_i \to \gamma \pi_{tc}^0$
365	$f_i \overline{f}_i \to \gamma \pi'^0_{tc}$
366	$f_i \overline{f}_i \to Z^0 \pi_{tc}^0$
367	$f_i \overline{f}_i \to Z^0 \pi'^0_{tc}$
368	$f_i \overline{f}_i \to W^\pm \pi_{tc}^\mp$
370	$f_i \overline{f}_j \to W_L^\pm Z_L^0$
371	$f_i \overline{f}_j \to W_L^\pm \pi_{tc}^0$
372	$f_i \overline{f}_j \to \pi_{tc}^\pm Z_L^0$
373	$f_i \overline{f}_j \to \pi_{tc}^\pm \pi_{tc}^0$
374	$f_i \overline{f}_j \to \gamma \pi_{tc}^\pm$
375	$f_i \overline{f}_j \to Z^0 \pi_{tc}^\pm$
376	$f_i \overline{f}_j \to W^\pm \pi_{tc}^0$
377	$f_i \overline{f}_j \to W^\pm \pi'^0_{tc}$
381	$q_i q_j \to q_i q_j$
382	$q_i \overline{q}_i \to q_k \overline{q}_k$
383	$q_i \overline{q}_i \to gg$
384	$f_i g \to f_i g$
385	$gg \to q_k \overline{q}_k$
386	$gg \to gg$
387	$f_i \overline{f}_i \to Q_k \overline{Q}_k$
388	$gg \to Q_k \overline{Q}_k$

No.	Subprocess
Compositeness:	
146	$e\gamma \to e^*$
147	$dg \to d^*$
148	$ug \to u^*$
167	$q_i q_j \to d^* q_k$
168	$q_i q_j \to u^* q_k$
169	$q_i \overline{q}_i \to e^\pm e^{*\mp}$
165	$f_i \overline{f}_i (\to \gamma^*/Z^0) \to f_k \overline{f}_k$
166	$f_i \overline{f}_j (\to W^\pm) \to f_k \overline{f}_l$
Extra Dimensions:	
391	$f\overline{f} \to G^*$
392	$gg \to G^*$
393	$q\overline{q} \to gG^*$
394	$qg \to qG^*$
395	$gg \to gG^*$
Left–right symmetry:	
341	$\ell_i \ell_j \to H_L^{\pm\pm}$
342	$\ell_i \ell_j \to H_R^{\pm\pm}$
343	$\ell_i^\pm \gamma \to H_L^{\pm\pm} e^\mp$
344	$\ell_i^\pm \gamma \to H_R^{\pm\pm} e^\mp$
345	$\ell_i^\pm \gamma \to H_L^{\pm\pm} \mu^\mp$
346	$\ell_i^\pm \gamma \to H_R^{\pm\pm} \mu^\mp$
347	$\ell_i^\pm \gamma \to H_L^{\pm\pm} \tau^\mp$
348	$\ell_i^\pm \gamma \to H_R^{\pm\pm} \tau^\mp$
349	$f_i \overline{f}_i \to H_L^{++} H_L^{--}$
350	$f_i \overline{f}_i \to H_R^{++} H_R^{--}$
351	$f_i f_j \to f_k f_l H_L^{\pm\pm}$
352	$f_i f_j \to f_k f_l H_R^{\pm\pm}$
353	$f_i \overline{f}_i \to Z_R^0$
354	$f_i \overline{f}_j \to W_R^\pm$
SUSY:	
201	$f_i \overline{f}_i \to \tilde{e}_L \tilde{e}_L^*$
202	$f_i \overline{f}_i \to \tilde{e}_R \tilde{e}_R^*$
203	$f_i \overline{f}_i \to \tilde{e}_L \tilde{e}_R^* +$
204	$f_i \overline{f}_i \to \tilde{\mu}_L \tilde{\mu}_L^*$
205	$f_i \overline{f}_i \to \tilde{\mu}_R \tilde{\mu}_R^*$
206	$f_i \overline{f}_i \to \tilde{\mu}_L \tilde{\mu}_R^* +$
207	$f_i \overline{f}_i \to \tilde{\tau}_1 \tilde{\tau}_1^*$
208	$f_i \overline{f}_i \to \tilde{\tau}_2 \tilde{\tau}_2^*$
209	$f_i \overline{f}_i \to \tilde{\tau}_1 \tilde{\tau}_2^* +$

No.	Subprocess
210	$f_i \overline{f}_j \to \tilde{\ell}_L \tilde{\nu}_\ell^* +$
211	$f_i \overline{f}_j \to \tilde{\tau}_1 \tilde{\nu}_\tau^* +$
212	$f_i \overline{f}_j \to \tilde{\tau}_2 \tilde{\nu}_\tau^* +$
213	$f_i \overline{f}_i \to \tilde{\nu}_\ell \tilde{\nu}_\ell^*$
214	$f_i \overline{f}_i \to \tilde{\nu}_\tau \tilde{\nu}_\tau^*$
216	$f_i \overline{f}_i \to \tilde{\chi}_1 \tilde{\chi}_1$
217	$f_i \overline{f}_i \to \tilde{\chi}_2 \tilde{\chi}_2$
218	$f_i \overline{f}_i \to \tilde{\chi}_3 \tilde{\chi}_3$
219	$f_i \overline{f}_i \to \tilde{\chi}_4 \tilde{\chi}_4$
220	$f_i \overline{f}_i \to \tilde{\chi}_1 \tilde{\chi}_2$
221	$f_i \overline{f}_i \to \tilde{\chi}_1 \tilde{\chi}_3$
222	$f_i \overline{f}_i \to \tilde{\chi}_1 \tilde{\chi}_4$
223	$f_i \overline{f}_i \to \tilde{\chi}_2 \tilde{\chi}_3$
224	$f_i \overline{f}_i \to \tilde{\chi}_2 \tilde{\chi}_4$
225	$f_i \overline{f}_i \to \tilde{\chi}_3 \tilde{\chi}_4$
226	$f_i \overline{f}_i \to \tilde{\chi}_1^\pm \tilde{\chi}_1^\mp$
227	$f_i \overline{f}_i \to \tilde{\chi}_2^\pm \tilde{\chi}_2^\mp$
228	$f_i \overline{f}_i \to \tilde{\chi}_1^\pm \tilde{\chi}_2^\mp$
229	$f_i \overline{f}_j \to \tilde{\chi}_1 \tilde{\chi}_1^\pm$
230	$f_i \overline{f}_j \to \tilde{\chi}_2 \tilde{\chi}_1^\pm$
231	$f_i \overline{f}_j \to \tilde{\chi}_3 \tilde{\chi}_1^\pm$
232	$f_i \overline{f}_j \to \tilde{\chi}_4 \tilde{\chi}_1^\pm$
233	$f_i \overline{f}_j \to \tilde{\chi}_1 \tilde{\chi}_2^\pm$
234	$f_i \overline{f}_j \to \tilde{\chi}_2 \tilde{\chi}_2^\pm$
235	$f_i \overline{f}_j \to \tilde{\chi}_3 \tilde{\chi}_2^\pm$
236	$f_i \overline{f}_j \to \tilde{\chi}_4 \tilde{\chi}_2^\pm$
237	$f_i \overline{f}_i \to \tilde{g} \tilde{\chi}_1$
238	$f_i \overline{f}_i \to \tilde{g} \tilde{\chi}_2$
239	$f_i \overline{f}_i \to \tilde{g} \tilde{\chi}_3$
240	$f_i \overline{f}_i \to \tilde{g} \tilde{\chi}_4$
241	$f_i \overline{f}_j \to \tilde{g} \tilde{\chi}_1^\pm$
242	$f_i \overline{f}_j \to \tilde{g} \tilde{\chi}_2^\pm$
243	$f_i \overline{f}_i \to \tilde{g} \tilde{g}$
244	$gg \to \tilde{g} \tilde{g}$
246	$f_i g \to \tilde{q}_{iL} \tilde{\chi}_1$
247	$f_i g \to \tilde{q}_{iR} \tilde{\chi}_1$
248	$f_i g \to \tilde{q}_{iL} \tilde{\chi}_2$
249	$f_i g \to \tilde{q}_{iR} \tilde{\chi}_2$

No.	Subprocess
250	$f_i g \to \tilde{q}_{iL} \tilde{\chi}_3$
251	$f_i g \to \tilde{q}_{iR} \tilde{\chi}_3$
252	$f_i g \to \tilde{q}_{iL} \tilde{\chi}_4$
253	$f_i g \to \tilde{q}_{iR} \tilde{\chi}_4$
254	$f_i g \to \tilde{q}_{jL} \tilde{\chi}_1^\pm$
256	$f_i g \to \tilde{q}_{jL} \tilde{\chi}_2^\pm$
258	$f_i g \to \tilde{q}_{iL} \tilde{g}$
259	$f_i g \to \tilde{q}_{iR} \tilde{g}$
261	$f_i \overline{f}_i \to \tilde{t}_1 \tilde{t}_1^*$
262	$f_i \overline{f}_i \to \tilde{t}_2 \tilde{t}_2^*$
263	$f_i \overline{f}_i \to \tilde{t}_1 \tilde{t}_2^* +$
264	$gg \to \tilde{t}_1 \tilde{t}_1^*$
265	$gg \to \tilde{t}_2 \tilde{t}_2^*$
271	$f_i f_j \to \tilde{q}_{iL} \tilde{q}_{jL}$
272	$f_i f_j \to \tilde{q}_{iR} \tilde{q}_{jR}$
273	$f_i f_j \to \tilde{q}_{iL} \tilde{q}_{jR} +$
274	$f_i \overline{f}_j \to \tilde{q}_{iL} \tilde{q}_{jL}^*$
275	$f_i \overline{f}_j \to \tilde{q}_{iR} \tilde{q}_{jR}^*$
276	$f_i \overline{f}_j \to \tilde{q}_{iL} \tilde{q}_{jR}^* +$
277	$f_i \overline{f}_i \to \tilde{q}_{jL} \tilde{q}_{jL}^*$
278	$f_i \overline{f}_i \to \tilde{q}_{jR} \tilde{q}_{jR}^*$
279	$gg \to \tilde{q}_{iL} \tilde{q}_{iL}^*$
280	$gg \to \tilde{q}_{iR} \tilde{q}_{iR}^*$
281	$bq_i \to \tilde{b}_1 \tilde{q}_{iL}$
282	$bq_i \to \tilde{b}_2 \tilde{q}_{iR}$
283	$bq_i \to \tilde{b}_1 \tilde{q}_{iR} +$
284	$b\overline{q}_i \to \tilde{b}_1 \tilde{q}_{iL}^*$
285	$b\overline{q}_i \to \tilde{b}_2 \tilde{q}_{iR}^*$
286	$b\overline{q}_i \to \tilde{b}_1 \tilde{q}_{iR}^* +$
287	$f_i \overline{f}_i \to \tilde{b}_1 \tilde{b}_1^*$
288	$f_i \overline{f}_i \to \tilde{b}_2 \tilde{b}_2^*$
289	$gg \to \tilde{b}_1 \tilde{b}_1^*$
290	$gg \to \tilde{b}_2 \tilde{b}_2^*$
291	$bb \to \tilde{b}_1 \tilde{b}_1$
292	$bb \to \tilde{b}_2 \tilde{b}_2$
293	$bb \to \tilde{b}_1 \tilde{b}_2$
294	$bg \to \tilde{b}_1 \tilde{g}$
295	$bg \to \tilde{b}_2 \tilde{g}$
296	$b\overline{b} \to \tilde{b}_1 \tilde{b}_2^* +$

The Les Houches Accord

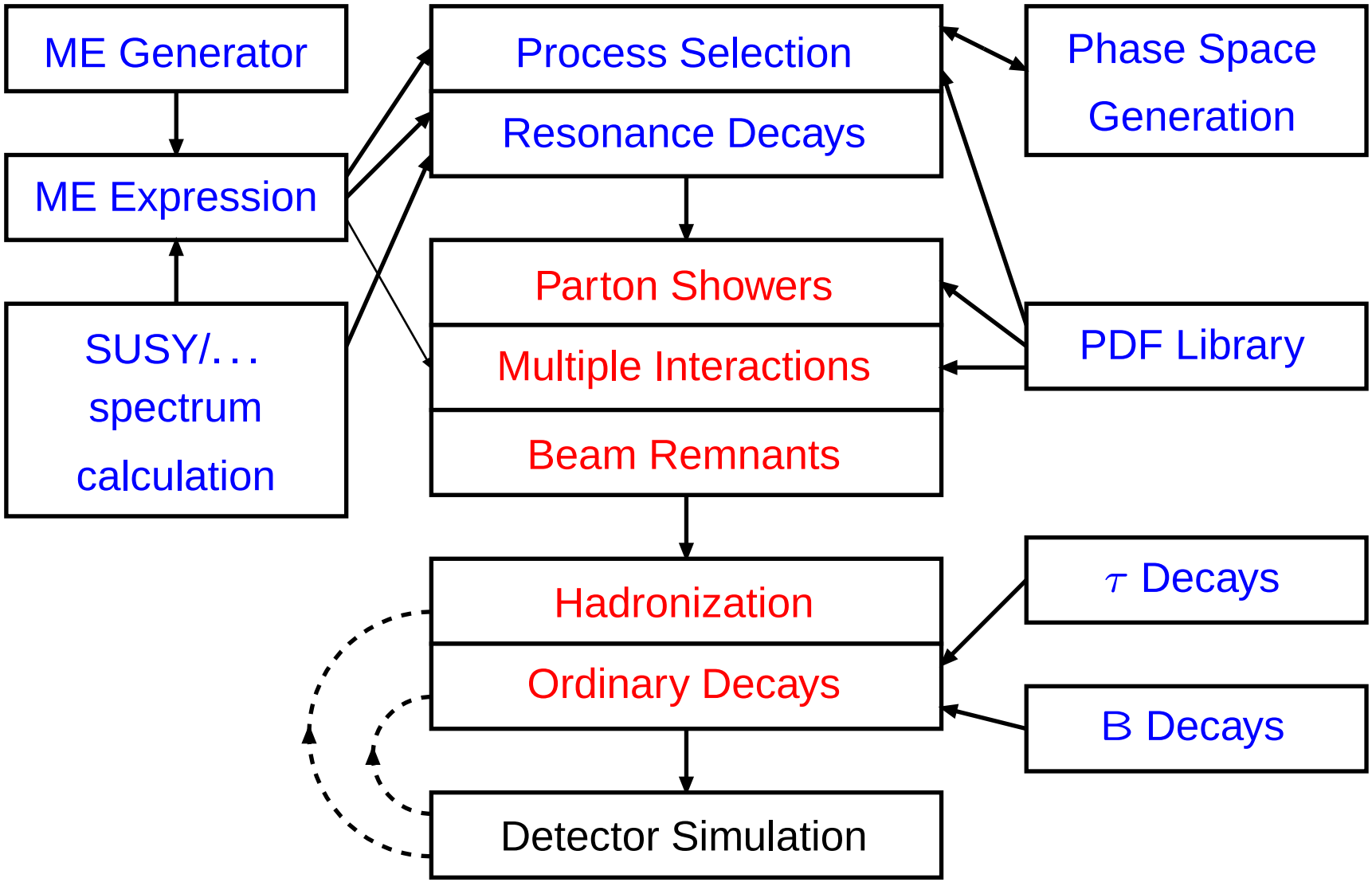


Some Specialized Generators:

- AcerMC: $\mathrm{t\bar{t}b\bar{b}}$, ...
- ALPGEN: $\mathrm{W/Z+} \leq 6\mathrm{j}$, $n\mathrm{W} + m\mathrm{Z} + k\mathrm{H+} \leq 3\mathrm{j}$, ...
- AMEGIC++: generic LO
- CompHEP: generic LO
- GRACE+Bases/Spring: generic LO+ some NLO loops
- GR@PPA: $\mathrm{b\bar{b}b\bar{b}}$
- MadCUP: $\mathrm{W/Z+} \leq 3\mathrm{j}, \mathrm{t\bar{t}b\bar{b}}$
- MadGraph+HELAS: generic LO
- MCFM: NLO $\mathrm{W/Z+} \leq 2\mathrm{j}$, WZ, WH, $\mathrm{H+} \leq 1\mathrm{j}$
- O'Mega+WHIZARD: generic LO
- VECBOS: $\mathrm{W/Z+} \leq 4\mathrm{j}$

Apologies for all unlisted programs

The Bigger Picture



$\Longrightarrow$ need standardized interfaces (LHAPDF, SUSY LHA, . . .)

Transverse-momentum-ordered showers

1) *Define* $\mathrm{p}^2_{\perp\mathrm{evol}} = z(1-z)Q^2 = z(1-z)M^2$ for FSR

$\mathrm{p}^2_{\perp\mathrm{evol}} = (1-z)Q^2 = (1-z)(-M^2)$ for ISR

2) *Evolve* all partons *downwards* in $\mathrm{p}_{\perp\mathrm{evol}}$ from common $p_{\perp\mathrm{max}}$

$$\mathrm{d}\mathcal{P}_a = \frac{\mathrm{d}\mathrm{p}^2_{\perp\mathrm{evol}}}{\mathrm{p}^2_{\perp\mathrm{evol}}} \frac{\alpha_\mathrm{s}(\mathrm{p}^2_{\perp\mathrm{evol}})}{2\pi} P_{a\to bc}(z)\,\mathrm{d}z\,\exp\left(-\int_{\mathrm{p}^2_{\perp\mathrm{evol}}}^{p^2_{\perp\mathrm{max}}} \cdots\right)$$

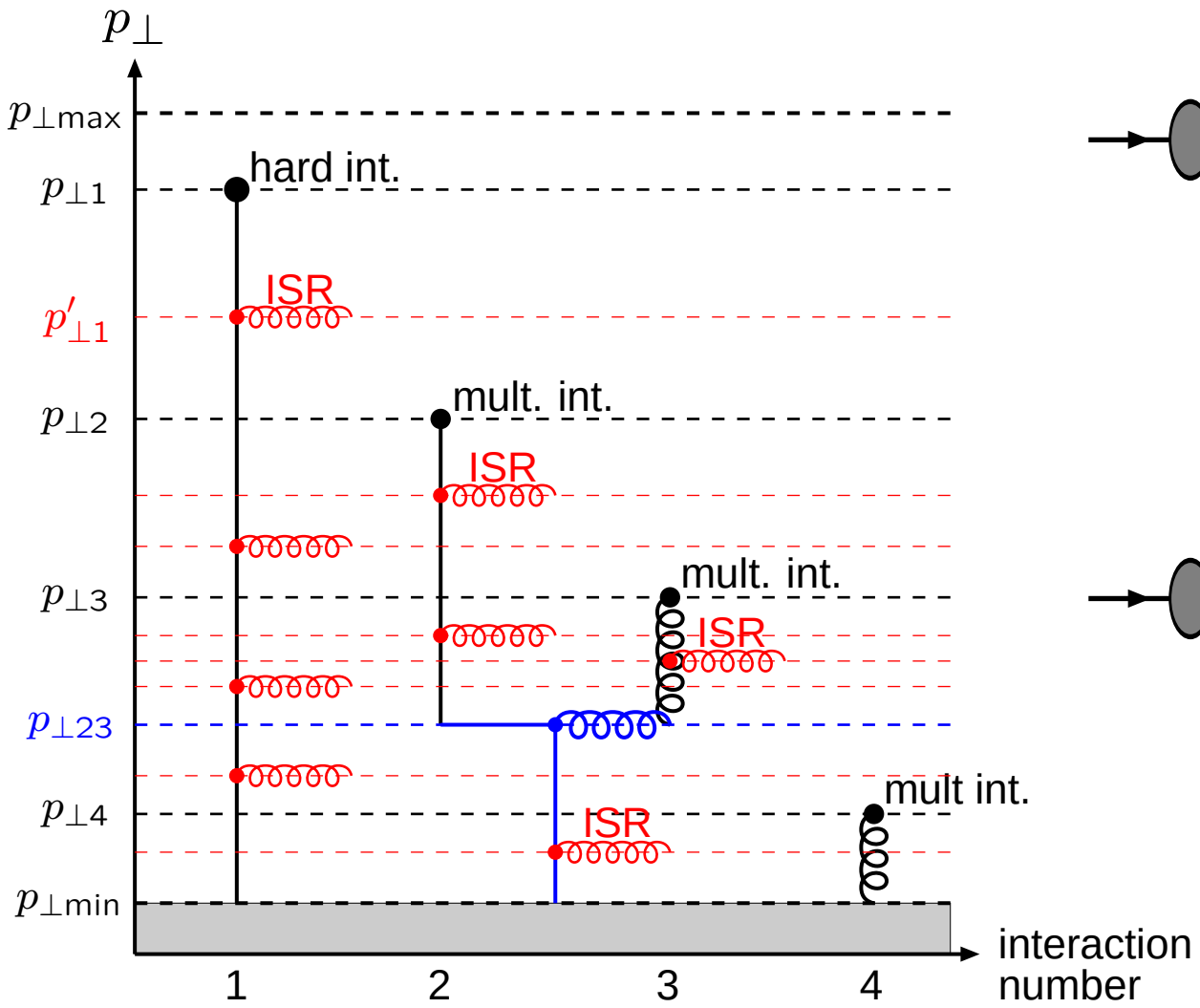
$$\mathrm{d}\mathcal{P}_b = \frac{\mathrm{d}\mathrm{p}^2_{\perp\mathrm{evol}}}{\mathrm{p}^2_{\perp\mathrm{evol}}} \frac{\alpha_\mathrm{s}(\mathrm{p}^2_{\perp\mathrm{evol}})}{2\pi} \frac{x' f_a(x', \mathrm{p}^2_{\perp\mathrm{evol}})}{x f_b(x, \mathrm{p}^2_{\perp\mathrm{evol}})} P_{a\to bc}(z)\,\mathrm{d}z\,\exp\left(-\cdots\right)$$

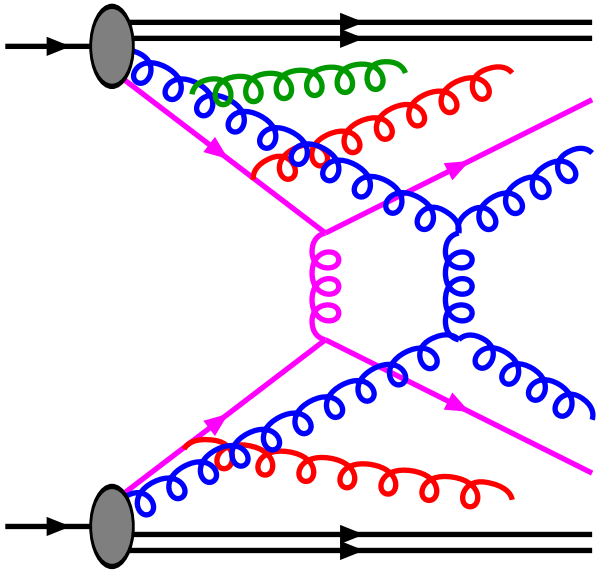
Pick the one with *largest* $\mathrm{p}_{\perp\mathrm{evol}}$ to undergo branching; also gives z.

3) Kinematics: *Derive* $Q^2 = \pm M^2$ by inversion of 1), but then interpret z as *energy fraction* (not lightcone) in "dipole" rest frame, so that *Lorentz invariant* and matched to matrix elements. Assume yet unbranched partons on-shell and shuffle $(E, \mathbf{p})$ inside dipole.

4)*Iterate* $\Rightarrow$ combined sequence $p_{\perp\mathrm{max}} > p_{\perp 1} > p_{\perp 2} > \ldots > p_{\perp\mathrm{min}}$.

Interleaved Multiple Interactions





PYTHIA 8: A fresh start

Problem: PYTHIA 7 stalled, no other manpower

Solution?: take a sabbatical and work “full-time”!

($\Rightarrow$ baseline model, S. Mrenna & P. Skands join later ?)

Tentative schedule:

time	date	processes	final states
0 =	1 Sept. 2004	—	—
1 =	1 Sept. 2005	LHA-style input	incomplete draft
2 =	1 Sept. 2006	a few processes	complete, buggy(?)
3 =	1 Sept. 2007	more processes	stable, debugged

. . . but don’t forget Murphy’s law

Objectives:

- clean up, keep the most recent models
- core program completely standalone, but
- Les Houches Accord style input central
- interfaces to other libraries foreseen

Distribution

Contents of PYTHIA 8.040 distribution:

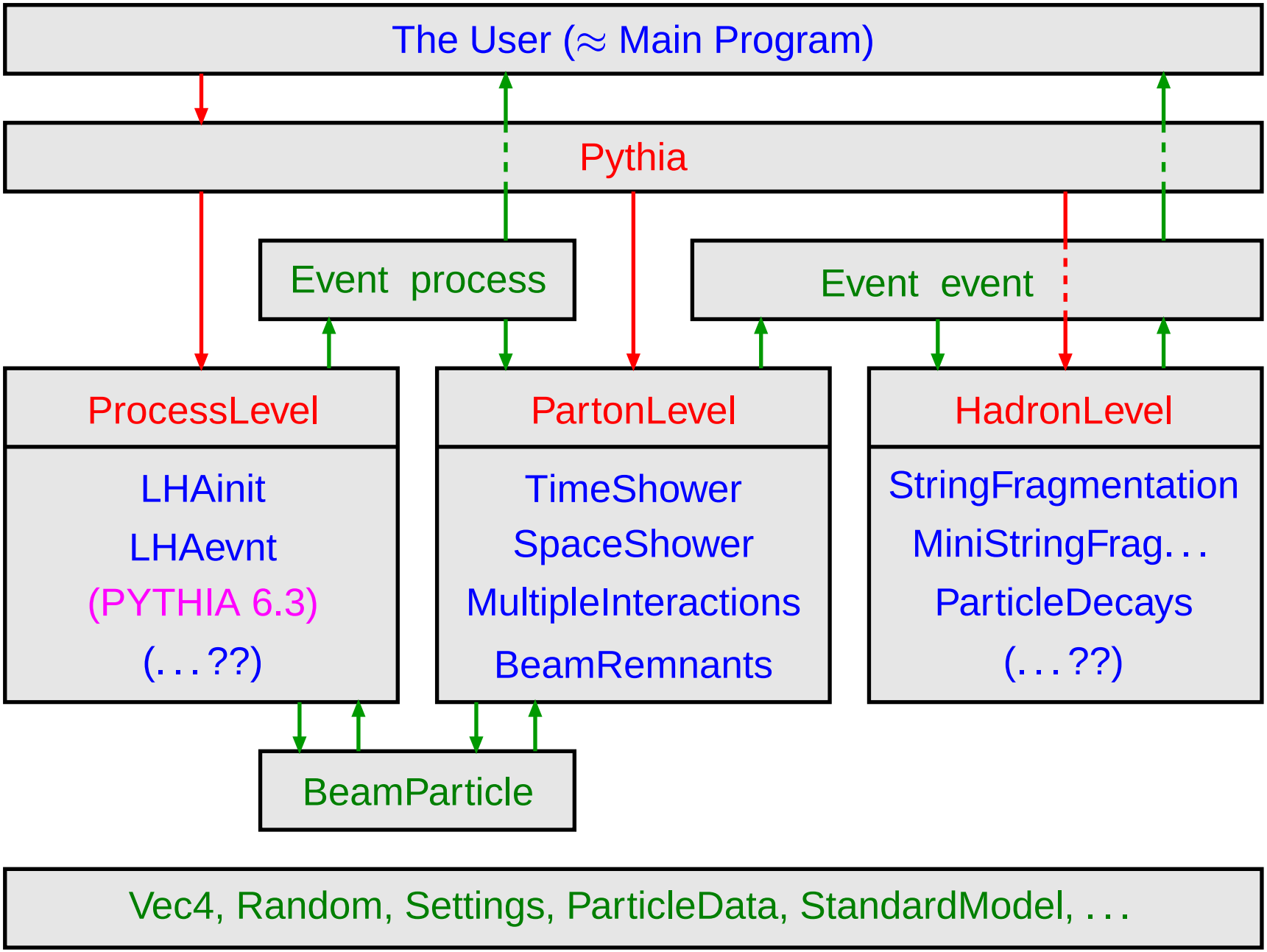
no	Description	size
1	Introduction (.pdf)	20 pp
24	Header files (.h)	3,850 lines
22	Code files (.cc)	14,750 lines
1	PYTHIA 6.3 file (.f)	71,500 lines
25	Documentation files (.man)	4,700 lines
5	Sample main programs (.cc)	870 lines
3	Input to above	1,380 lines
1	Makefile	150 lines
1	pythia8040.tar.gz (all)	1 MB

To get going: download from PYTHIA webpage
$\Rightarrow$ gunzip $\Rightarrow$ tar xvf $\Rightarrow$ make $\Rightarrow$ run test program(s)

Self-contained, but hooks to external programs for

- hard processes, Les Houches Accord style
- parton distribution functions
- decays (of some particles, e.g. τ, B^0, B^+)
- random number generators (shared with other programs)

Current PYTHIA 8 structure



Current PYTHIA 8 status

	Existing classes		Missing classes/topics
Process Level	LHAinit	⋆⋆	Cross section administration
	LHAevnt	⋆⋆	Phase space selection
	(PYTHIA 6.3)	⋆⋆⋆	Process matrix elements
Parton Level	TimeShower	⋆⋆	Parton density libraries
	SpaceShower	⋆⋆	Resonance decays
	MultipleInteractions	⋆⋆	ThePEG input (?)
	BeamRemnants	⋆⋆	MI/ISR/FSR interleaving
Hadron Level	StringFragmentation	⋆⋆	colour flow models
	MiniStringFrag. . .	⋆⋆	ME/PS matching
	ParticleDecays	⋆⋆	Popcorn baryons
—	Event	⋆⋆	updated decay tables
	BeamParticle	⋆⋆	Bose-Einstein
	Vec4, Random	⋆⋆⋆	event analysis routines
	Settings	⋆⋆	. . . and much, much more
	ParticleData	⋆⋆	

Event generation structure

1) Initialization step
- select process(es) to study
- modify physics parameters
- set kinematics constraints
- modify generator settings
- initialize generator
- book histograms

```
#include "Pythia.h"
using namespace Pythia8;
Pythia pythia;
pythia.readString("command");
pythia.readFile("command.file");
pythia.init(idBeamA,idBeamB,eCM);
```

2) Generation loop
- generate one event at a time
- analyze it (or store for later)
- add results to histograms
- print a few events

```
pythia.next();
pythia.process.list();
pythia.event.list();
int id = pythia.event[i].id();
```

3) Finishing step
- print deduced cross-sections
- print/save histograms etc.

```
pythia.statistics();
pythia.settings.listChanged();
```

Initialization and generation commands

Standard in preamble:

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

Initialization by one of different forms:

- `pythia.init( idA, idB, eA, eB)` along $\pm z$ axis
- `pythia.init( idA, idB, eCM)` in c.m. frame
- `pythia.init( machine, eCM)` with `pp`, `pbarp`, `ppbar`, `e+e-`, `e-e+`
- `pythia.init( LHAinit*, LHAevnt*)` for Les Houches Accord

Generation of next event by:

- `pythia.next()`

with no arguments, but value `false` if failed (rare!)

At the end of the generation loop:

- `pythia.statistics()`

provides some summary information

The Settings class

Want to modify event execution by

- Flags: on/off switches, `bool`
- Modes: enumerated options, `int`
- Parameters: continuum of values, `double`

For each such, need to store

- name: of form `location:name`, e.g. `TimeShower:pTmin`
- default value
- current value
- allowed range: maximum/minimum on/off (not for flags).

Info is stored in `.man` files, matched to `.cc/.h` files,
and used to build three maps at instantiation of `Pythia` object.
User modifies by methods, most commonly

- `Settings::readString("location:name = value")`
- `Settings::readFile("filename")` with one command per line

e.g. `TimeShower:pTmin = 1.0` as argument or line in file.

Also useful:

- `Settings::listAll()` : complete list
- `Settings::listChanged()` : only changed ones

Example of latter:

```
--------  Pythia Flag + Mode + Parameter Settings (changes only)  ----------------------

  Kind  Name                                    Now    Default       Min       Max
  bool  Beams:allowJunction                     off         on
double  Beams:primordialKTwidth                2.0000    1.0000    0.0000
  bool  HadronLevel:Decay                       off         on
double  Main:eCM                            1.40e+04 1960.0000   10.0000
   int  Main:numberToList                          1          2         0
   int  Main:timesToShow                          20         50         0
   int  MultipleInteractions:bProfile              3          2         0         3
double  MultipleInteractions:expPow            1.3000    1.0000    0.4000   10.0000
double  MultipleInteractions:pT0Ref            3.5000    3.0000    0.5000   10.0000
double  SpaceShower:pT0Ref                     3.0000    2.5000    0.5000   10.0000
  bool  SpaceShower:samePTasMI                  off         on
  bool  TimeShower:QEDshowerByQ                 off         on

--------  End Pythia Flag + Mode + Parameter Settings  -------------------------------
```

Particle Data

The static `ParticleDataTable` class contains info by PDG `id` code:

- `hasAnti(id)`
- `name(id)`
- `charge3(id), charge(id), colType(id)`
- `m0(id), constituentMass(id), width(id), range(id)`
- `tau0(id)`
- `mayDecay(id)`

plus a vector of `DecayChannel`s with

- `branchingRatio()`
- `modeME()`
- `multiplicity()`
- `product(i)`

User modifies by methods, most commonly

- `ParticleDataTable::readString("...")`
- `ParticleDataTable::readFile("filename")`

with command `Particle:id:property = value`

or `Particle:id:channel:property = value`

List current data by `ParticleDataTable::list()` (here truncated):

```
--------  Pythia Particle Data Table (complete)  ---------------------------------------------------

     id   name            antiname    3*charge colour      m0        width       range        tau0 de
            no branchratio mode        products

      1   d               dbar              -1   1      0.3300      0.0000      0.0000  0.0000e+00

      2   u               ubar               2   1      0.3300      0.0000      0.0000  0.0000e+00

    111   pi0                                0   0      0.1350      0.0000      0.0000  3.0000e-05
            0   0.9880000    0       22        22
            1   0.0120000    2       22        11      -11

    113   rho0                               0   0      0.7685      0.1510      0.4000  0.0000e+00
            0   0.9987390    3      211      -211
            1   0.0007900    0      111        22
            2   0.0003800    0      221        22
            3   0.0000460    0       13       -13
            4   0.0000450    0       11       -11
```

Currently PYTHIA 6.3 data $\Rightarrow$ long overdue for update

Hard-process generation with PYTHIA 6

Currently no hard processes at all in PYTHIA 8
$\Longrightarrow$ use Fortran PYTHIA 6 library, and then
transfer events via (Fortran) LHA interface.
Provide settings with `pythia.readString("...")`
or `pythia.readFile("filename")`,
of form `Pythia6:variable = value`,
where `variable` is anything recognized by `PYGIVE`,
but only ones relevant for hard process are actually used.

To simplify,

- `pythia.readString("...")` and
- `pythia.readFile("filename")`

will take one or many commands that are directed either
to `Settings`, to `ParticleDataTable` or to `Pythia6`.
Thus the specialized methods in `Settings` and `ParticleDataTable`
are superfluous (for the normal user).

The Particle class

Each `Particle` object stores the properties:

- `id()` : particle identity, by PDG codes.
- `status()` : status code. Provides info on where and why a given particle was produced. Negative code = no longer existing particle.
- `mother1(), mother2()` : first and last mother indices.
- `daughter1(), daughter2()` : first and last daughter indices.
- `col(), acol()` : colour and anticolour tags, Les Houches Accord.
- `px(), py(), pz(), e()` : four-momentum components (in GeV).
- `m()` : mass.
- `scale()` : scale at which a parton was produced; model-specific.
- `xProd(), yProd(), zProd(), tProd()` : production vertex (in mm).
- `tau()` : proper lifetime.

Methods above can also be used, with argument, for setting properties.
Many further methods for extraction only, e.g. for rapidity.
Also pointer to `ParticleDataEntry` object; gives e.g. `name()` and `charge()`.

The Event class

Two `Event` objects inside a `Pythia` object:

- `process` : hard subprocess, roughly like Les Houches.
- `event` : complete event history.

An `Event` $\approx$ a `vector<Particle>`

e.g. `pythia.event[i].id()` = identity of i'th particle

index 0 = event-as-a-whole; not really part of history

thus mother/daughter = 0 $\Leftrightarrow$ empty

Specific methods include:

- `size()` : 0 $\leq$ `i` $<$ `event.size()`.
- `list()` : provide event listing.
- `motherList(i), daughterList(i), sisterList()` : a `vector<int>` of mothers, daughters, sisters.
- `iTopCopy(i), iBotCopy(i)` : top or bottom "carbon copy".

Further: info on junctions, . . .

Sample event listings

First with `pythia.process.list()`, truncated to fit:

```
--------  Pythia Event Listing  (hard process)  -------------------------------

    no        id   name         status     mothers   daughters     colours      p_x
     0        90   (system)        -11     0     0     0     0     0     0     0.000
     1      2212   (p+)            -12     0     0     3     0     0     0     0.000
     2      2212   (p+)            -12     0     0     4     0     0     0     0.000
     3        21   (g)             -21     1     0     5     6   101   102     0.000
     4        21   (g)             -21     2     0     5     6   103   101     0.000
     5        -6   (tbar)          -22     3     4     7     8     0   102  -107.572
     6         6   (t)             -22     3     4     9    10   103     0   107.572
     7       -24   (W-)            -22     5     0    11    12     0     0   -71.772
     8        -5   bbar             23     5     0     0     0     0   102   -35.799
     9        24   (W+)            -22     6     0    13    14     0     0   113.539
    10         5   b                23     6     0     0     0   103     0    -5.968
    11        11   e-               23     7     0     0     0     0     0   -38.516
    12       -12   nu_ebar          23     7     0     0     0     0     0   -33.256
    13        -1   dbar             23     9     0     0     0     0   104    24.321
    14         2   u                23     9     0     0     0   104     0    89.218
                                                                  Sum:        -0.000
--------  End Pythia Event Listing  ---------------------------------------------
```

next with `pythia.event.list()`, omissions to fit:

-------- Pythia Event Listing (complete event) ----------

no	id	name	status	mothers		daughters		colours		p_x	p_y	p_z	e	m
0	90	(system)	-11	0	0	0	0	0	0	0.000	0.000	0.000	14000.000	14000.000
1	2212	(p+)	-12	0	0	187	0	0	0	0.000	0.000	7000.000	7000.000	0.938
2	2212	(p+)	-12	0	0	188	0	0	0	0.000	0.000	-7000.000	7000.000	0.938
3	21	(g)	-21	7	0	5	6	101	102	0.000	0.000	53.792	53.792	0.000
4	21	(g)	-21	8	8	5	6	103	101	0.000	0.000	-829.022	829.022	0.000
5	-6	(tbar)	-22	3	4	9	9	0	102	-107.572	-45.614	-345.827	404.638	174.595
6	6	(t)	-22	3	4	10	10	103	0	107.572	45.614	-429.402	478.176	174.969
7	21	(g)	-41	12	12	11	3	105	102	-0.000	-0.000	76.351	76.351	0.000
8	21	(g)	-42	13	0	4	4	103	101	-0.000	0.000	-829.022	829.022	0.000
9	-6	(tbar)	-44	5	5	14	14	0	102	-127.853	-17.612	-332.165	396.829	174.595
10	6	(t)	-44	6	6	15	15	103	0	90.752	68.837	-379.579	433.208	174.969
11	21	(g)	-43	7	0	16	16	105	101	37.101	-51.226	-40.927	75.336	0.000
(skipped)														
63	21	(g)	-31	111	0	65	66	112	111	0.000	0.000	0.070	0.070	0.000
64	-4	(cbar)	-31	112	112	65	66	0	110	0.000	0.000	-926.957	926.957	0.000
65	21	(g)	-33	63	64	113	113	112	110	5.011	-0.788	-104.687	104.810	0.000
66	-4	(cbar)	-33	63	64	114	114	0	111	-5.011	0.788	-822.200	822.217	1.500
(skipped)														
237	2101	(ud_0)	-63	1	0	0	0	0	137	0.240	-0.007	3177.306	3177.306	0.579
238	-1	(dbar)	-63	1	0	0	0	0	124	1.153	-0.432	839.002	839.003	0.330
239	2101	(ud_0)	-63	2	0	0	0	0	142	-1.091	0.128	-2613.733	2613.733	0.579
240	4	(c)	-63	2	0	0	0	142	0	-0.557	1.321	-174.031	174.043	1.500
(skipped)														
241	-24	(W-)	-22	195	0	245	245	0	0	-102.292	-46.372	-349.729	376.307	81.747
242	-5	(bbar)	-23	195	0	243	244	0	102	-39.504	23.812	-8.300	47.111	4.800
243	-5	(bbar)	-51	242	0	248	248	0	144	-26.921	15.510	-8.835	32.656	4.800
244	21	(g)	-51	242	0	246	247	144	102	-12.740	8.184	-0.143	15.143	0.000
245	-24	(W-)	-52	241	241	263	264	0	0	-102.135	-46.255	-349.051	375.619	81.747
(skipped)														
263	11	(e-)	-23	245	0	265	266	0	0	-49.476	20.517	-126.258	137.149	0.001
264	-12	(nu_ebar)	-23	245	0	267	267	0	0	-52.659	-66.772	-222.793	238.470	0.000
265	11	e-	51	263	0	0	0	0	0	-48.966	20.308	-124.957	135.736	0.001
266	22	gamma	51	263	0	0	0	0	0	-0.510	0.210	-1.301	1.413	0.000
267	-12	nu_ebar	52	264	264	0	0	0	0	-52.659	-66.772	-222.793	238.470	0.000
(skipped)														
285	323	K*+	73	247	0	0	0	0	0	-8.774	4.484	-1.202	9.966	0.892
286	533	B*_s0	73	248	0	0	0	0	0	-24.787	14.045	-6.657	29.754	5.416
287	423	D*0	73	240	0	0	0	0	0	-0.604	1.434	-307.590	307.600	2.007
288	223	omega	73	240	0	0	0	0	0	-0.097	-0.243	-316.742	316.743	0.782
289	113	rho0	73	239	0	0	0	0	0	-0.424	-0.021	-525.177	525.178	0.768
290	2212	p+	73	239	0	0	0	0	0	-0.522	0.279	-1638.254	1638.254	0.938
(skipped)														
490	223	omega	73	237	0	0	0	0	0	0.481	-0.049	154.560	154.563	0.782
491	2212	p+	73	237	0	0	0	0	0	-0.269	-0.100	2588.971	2588.972	0.938
									Sum:	-0.000	-0.000	-0.000	14000.000	14000.000

-------- End Pythia Event Listing ----------

Utilities

Four-vectors in class `Vec4`, with overloaded operators.

A small package for one-dimensional histograms:

- Book with `Hist name( title, numberOfBins, xMin, xMax);`
  or `Hist name; name.book(title, numberOfBins, xMin, xMax);`
- Fill with `name.fill( xValue, weight);` with default `weight = 1`
- Print with `cout << name;`
- Overloaded operators for addition, multiplication, ...

Sphericity analysis:

- Instantiate with `Sphericity sph( power, select);`
- Analyze with `sph.analyze( event);`
- Info with `sph.sph(), sph.EigenVector(i), sph.list(), ...`

Cone jet finder a la UA1 (PYCELL):

- Instantiate with `CellJet cellJet(eTjetMin,coneRadius,select,etaMax,`
  `nEta,nPhi,eTseed,smear,resolution,upperCut,threshold)`
- Analyze with `cellJet.analyze( event);`
- Info with `cellJet.size(), cellJet.eT(i), cellJet.list(), ...`

Example of a main program

```
// Test program main06: study pTZ spectrum at the Tevatron.
#include "Pythia.h"
using namespace Pythia8;
int main() {
  // Generator. Process selection. Tevatron initialization. Histogram.
  Pythia pythia;
  pythia.readString("Pythia6:msel = 11");
  pythia.readString("Pythia6:ckin(1) = 80.");
  pythia.readString("PartonLevel:MI = off");
  pythia.readString("Beams:primordialKTwidth = 2.");
  pythia.init( 2212, -2212, 1960.);
  Hist pTZ("dN/dpTZ",100,0.,100.);
  // Begin event loop. Generate event. Skip if error. List first few.
  for (int iEvent = 0; iEvent < 10000; ++iEvent) {
    if (!pythia.next()) continue;
    if (iEvent < 2) pythia.event.list();
    // Loop over particles in event. Find last Z0 copy. Fill its pT.
    int iZ = 0;
    for (int i = 0; i < pythia.event.size(); ++i)
      if (pythia.event[i].id() == 23) iZ = i;
    pTZ.fill( pythia.event[iZ].pT() );
  // End of event loop. Statistics. Histogram. Done.
  }
  pythia.statistics();
  cout << pTZ;
  return 0;
}
```

Sample run

Main program `main07.cc`, with run data in `main07.cmnd`:

```
#include "Pythia.h"

using namespace Pythia8;

int main() {

  // Generator. Shorthand for the event and the (static) Settings.
  Pythia pythia;
  Event& event = pythia.event;
  Settings& settings = pythia.settings;

  // Read in commands from external file.
  pythia.readFile("main07.cmnd");

  // Extract settings to be used in the main program.
  int idBeamA = settings.mode("Main:idBeamA");
  int idBeamB = settings.mode("Main:idBeamB");
  double eCM = settings.parameter("Main:eCM");
  int nEvent = settings.mode("Main:numberOfEvents");
  int nList = settings.mode("Main:numberToList");
  int nShow = settings.mode("Main:timesToShow");
  bool showChangedSettings = settings.flag("Main:showChangedSettings");
  bool showAllSettings = settings.flag("Main:showAllSettings");
```

```
// Initialization for Pythia6 event input.
pythia.init( idBeamA, idBeamB, eCM);

// List changed data.
if (showChangedSettings) settings.listChanged();
if (showAllSettings) settings.listAll();

// Histograms.
double epTol = 1e-6 * eCM;
Hist epCons("deviation from energy-momentum conservation",100,0.,epTol);
Hist nFinal("final particle multiplicity",100,-0.5,799.5);
Hist dnparticledy("dn/dy for particles",100,-10.,10.);

// Begin event loop.
int nPace = max(1,nEvent/nShow);
for (int iEvent = 0; iEvent < nEvent; ++iEvent) {
  if (iEvent%nPace == 0) cout << " Now begin event " << iEvent << "\n";

  // Generate events. Quit if failure.
  if (!pythia.next()) {
    cout << " Event generation aborted prematurely, owing to error!\n";
    break;
  }

  // List first few events, both hard process and complete events.
  if (iEvent < nList) {
    pythia.process.list();
    event.list();
  }
```

```
    // Loop over final particles in the event.
    int nFin = 0;
    Vec4 pSum;
    for (int i = 0; i < event.size(); ++i) if (event[i].remains()) {
      nFin++;
      pSum += event[i].p();
      dnparticledy.fill(event[i].y());
    }

    // Check and print event with too big energy-momentum deviation.
    nFinal.fill(nFin);
    double epDev = abs(pSum.e() - eCM) + abs(pSum.px()) + abs(pSum.py())
      + abs(pSum.pz());
    epCons.fill(epDev);
    if (epDev > epTol) {
      cout << " Warning! Event with epDev = " << scientific
           << setprecision(4) << epDev << " now listed:";
      event.list();
    }

  // End of event loop.
  }

  // Final statistics and histogram output.
  pythia.statistics();
  cout << epCons << nFinal << dnparticledy;

  return 0;
}
```

Sample input cards

```
! This file contains commands to be read in for a Pythia8 run.
! Lines not beginning with a letter are comments.

! 1) Settings that could be used in a main program, if desired.
Main:idBeamA = 2212                ! first beam, p = 2212, pbar = -2212
Main:idBeamB = 2212                ! second beam, p = 2212, pbar = -2212
Main:eCM = 14000.                  ! CM energy of collision
Main:numberOfEvents = 1000         ! number of events to generate
Main:numberToList = 2              ! number of events to print
Main:timesToShow = 20             ! show how far along run is
Main:showChangedSettings = on      ! print changed flags/modes/parameters
Main:showAllSettings = off         ! print all flags/modes/parameters

! 2) Settings for the hard-process generation.
! Based on an interface to the Fortran Pythia6 program.
#Pythia6:msel = 1                   ! QCD production
#Pythia6:ckin(3) = 100.              ! pTmin cut
Pythia6:msel = 6                   ! t tbar production

! 3) Settings for the event generation process in the Pythia8 library.
#PartonLevel:MI = off               ! no multiple interactions
#PartonLevel:ISR = off              ! no initial-state radiation
#PartonLevel:FSRinProcess = off     ! no final-state radiation
PartonLevel:FSRinResonances = off  ! no FSR in resonance decays
#HadronLevel:Hadronize = off        ! no hadronization
SpaceShower:pT0Ref = 2.0            ! dampening of pT -> 0 divergence
MultipleInteractions:pTmin = 3.0   ! lower pT cutoff for interactions
```

LHAinit

Public methods:

`idBeamA(), idBeamB()`: incoming beam particles

`eBeamA(), eBeamB()`: incoming beam energies (GeV)

`pdfGroupBeamA(), pdfGroupBeamB(),`
`pdfSetBeamA(), pdfSetBeamB()`: PDF's

`strategy()`: weighting strategy

`size()`: number of processes, index `i` in range $0 \leq$ `i` $<$ `size`

`idProcess(i)`: integer identifier for each process

`xSec(i)`: σ_{tot} for each process

`xErr(i)`: error on σ_{tot} for each process

`xMax(i)`: $\mathrm{d}\sigma_{\mathrm{max}}$ for each process

Protected methods, to be used by `set`:

`LHAinit`, `~LHAinit`: constructor, destructor

`beamA(id, e, pdfGroup, pdfSet)`, same for `beamB`: set beams

`strategy(choice)`: set weighting strategy

`process(id, xSec, xErr, xMax)`: append process to list

LHAevnt

Public methods:

`idProc()`: identity of current process

`weight()`: event weight

`scale()`: scale Q of parton distributions etc.

`alphaQED(), alphaQCD()`: α_{em}, α_{s} used in event

`size()`: number of particles $+1$, index `i` in range $1 \leq$ `i` $<$ `size`
(keep slot 0 empty, for consistency with Fortran, mothers/daughters)

`id(i)`: PDG identity code for particle `i`

`status(i)`: status code

`mother1(), mother2()`: position of one or two mothers

`col1(), col2()`: colour and anticolour indices

`px(i), py(i), pz(i), e(i), m(i)`: (p_x, p_y, p_z, E, m)

`tau(i)`: invariant lifetime $c\tau$

`spin(i)`: spin (helicity) information

Protected methods, to be used by `set`:

`LHAevnt`, `~LHAevnt`: constructor, destructor

`process(id, weight, scale, alphaQED, alphaQCD)`: info on process

`particle(id, status, mother1, mother2, col1, col2,`
`px, py, pz, e, m, tau, spin)`: info on particle

Sample run with Les Houches input

```
#include "Pythia.h"
using namespace Pythia8;
int main() {

  int nPrint = 2;                                 // Number of events to print.
  Pythia pythia;                                  // Generator.
  pythia.readString("PartonLevel:MI = off");      // No multiple interactions.
  pythia.readString("SpaceShower:pTmin = 1.0"); // Change pTmin cutoff of ISR.
  LHAinitPythia6 lhaInit("ttsample.init");        // Les Houches initialization object.
  LHAevntPythia6 lhaEvnt("ttsample.evnt");        // Les Houches event object.
  pythia.init(&lhaInit, &lhaEvnt);                // Initialize with pointers.
  cout << lhaInit;                                // List initialization information.
  Hist nFinal("final particle multiplicity",100,-0.5,499.5);        // Histogram.

  int iEvent = 0;                                 // Begin event loop
  while (pythia.next()) {                         // Generate event until none left.
    if (iEvent++ < nPrint) {                      // List first few events.
      cout << lhaEvnt;                            // List Les Houches input event.
      pythia.process.list();                      // List Pythia hard-process event.
      pythia.event.list();                        // List Pythia complete event.
    }                                             // End listing.
    int nFin = 0;                                 // Sum up final multiplicity
    for (int i = 0; i < pythia.event.size(); ++i)
      if (pythia.event[i].remains()) nFin++;
    nFinal.fill(nFin);                            // Fill histogram.
  }                                               // End of event loop.

  cout << nFinal;                                 // Print histogram.
  return 0;                                       // Done.
}
```

Outlook

• C++ PYTHIA 8 is coming along •

⋆ Roughly according to three-year plan the first year! ⋆

⋆ On hold during autumn, hope to pick up again early 2006 ⋆

⋆ $\sim$ 1 sub-subversion per working week (backup) ⋆

⋆ Release latest sub-subversion every 2–3 months ⋆

⋆ First production-quality release, 8.100, early 2007 (?)

⋆ Debugged and tuned by LHC startup (??) ⋆

⋆ Overtaking Fortran version usage by 2009 (???) ⋆

• Early feedback is most welcome •

⋆ Now is the time for any major course changes ⋆

⋆ In a year's time the structure will be frozen ⋆

Trying It Out

- PYTHIA 8.041 is new for this meeting, but only minor changes/additions.
- Create a (sub)directory `pythia8041` and go to it.
- Download `pythia8041.tar.gz` into it, either of
  - ⋆ `http://www.thep.lu.se/~torbjorn/Pythia.html`, link “Future”
  - ⋆ CD disk
  - ⋆ USB memory stick
- `gunzip pythia8041.tar.gz`
- `tar xvf pythia8041.tar`
- `make`
- `./a.out > out` to run `main01.cc`
- Edit line `MAIN = main01.cc` in `Makefile` to run other program.
- Edit programs to change run, or create own programs.
- Use `Pythia8040.pdf` for overview and the various `.man` files for details.