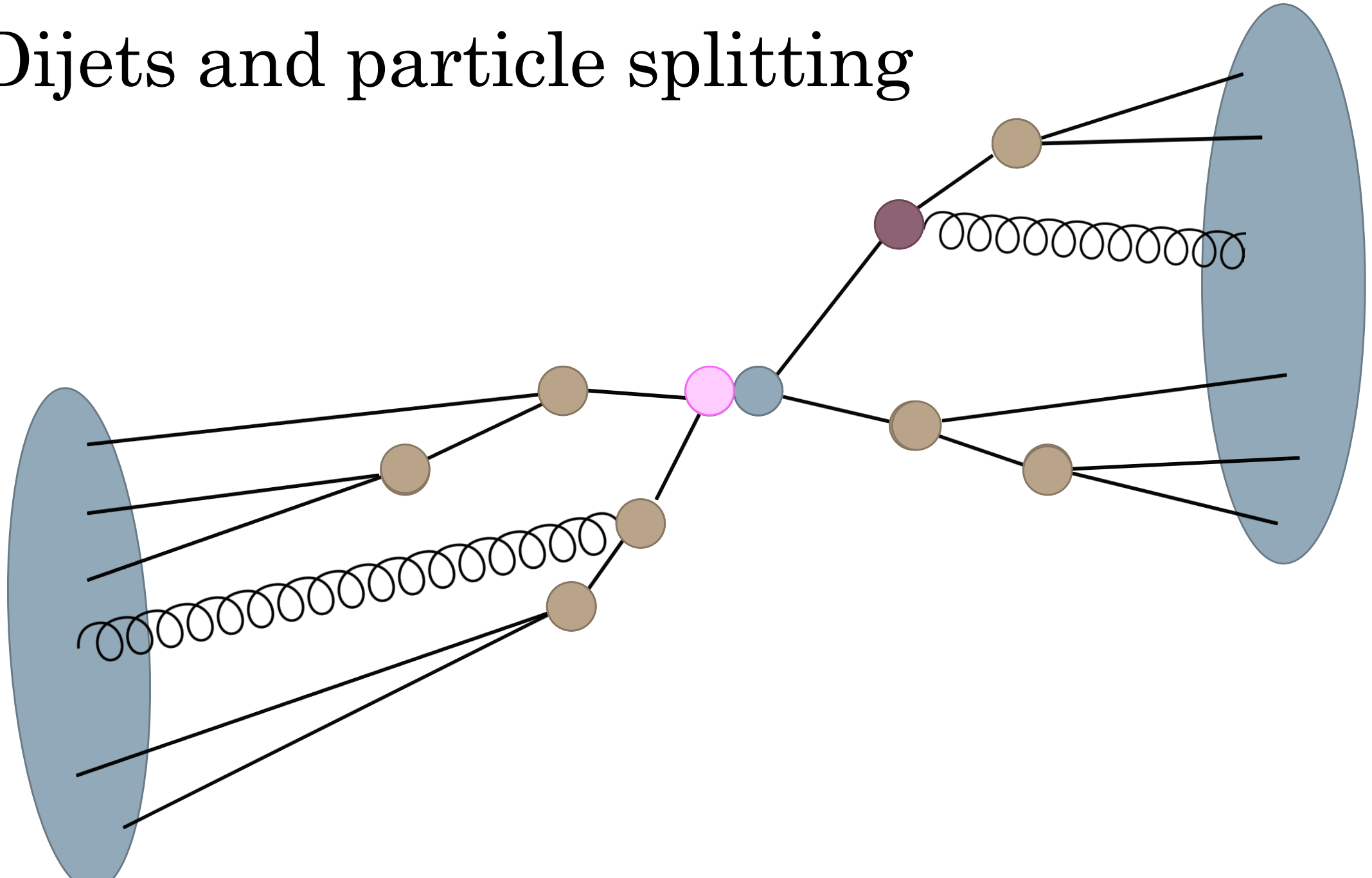
# Parton Shower: From QCD to Monte Carlo event generators

Íris Silva

Sérgio Carrôlo

# Dijets and particle splitting



# Pythia

- Pythia is a tool for the generation of high-energy collisions;
- It can generate pp, pp, $e^+e^-$ and $\mu^+\mu^-$ collisions;
- Initial- and final-state algorithms are based on $k_\perp$-ordered evolution ;
- In our project we use pythia to simulate pp collisions.

# Pythia – exemple of an event

-------- 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 | 5020.000 | 5020.000 |
| 1 | 2212 | (p+) | -12 | 0 | 0 | 130 | 0 | 0 | 0 | 0.000 | 0.000 | 2510.000 | 2510.000 | 0.938 |
| 2 | 2212 | (p+) | -12 | 0 | 0 | 131 | 0 | 0 | 0 | 0.000 | 0.000 | -2510.000 | 2510.000 | 0.938 |
| 3 | 21 | (g) | -21 | 12 | 12 | 5 | 6 | 102 | 101 | 0.000 | 0.000 | 220.970 | 220.970 | 0.000 |
| 4 | 21 | (g) | -21 | 9 | 9 | 5 | 6 | 104 | 103 | 0.000 | 0.000 | -93.065 | 93.065 | 0.000 |
| 5 | 21 | (g) | -23 | 3 | 4 | 10 | 11 | 104 | 101 | 122.068 | -24.854 | 141.735 | 188.698 | 0.000 |
| 6 | 21 | (g) | -23 | 3 | 4 | 7 | 8 | 102 | 103 | -122.068 | 24.854 | -13.830 | 125.337 | 0.000 |
| 7 | 21 | (g) | -51 | 6 | 0 | 19 | 20 | 102 | 105 | -63.359 | 24.142 | -28.528 | 73.560 | 0.000 |
| 8 | 21 | (g) | -51 | 6 | 0 | 13 | 14 | 105 | 103 | -58.708 | 0.712 | 7.309 | 59.166 | 0.000 |
| 9 | 21 | (g) | -53 | 15 | 15 | 4 | 4 | 104 | 103 | 0.000 | -0.000 | -100.454 | 100.454 | 0.000 |
| 10 | 21 | (g) | -51 | 5 | 0 | 16 | 17 | 104 | 106 | 76.698 | -19.606 | 88.346 | 118.626 | 0.000 |
| 11 | 21 | (g) | -51 | 5 | 0 | 31 | 32 | 106 | 101 | 45.370 | -5.247 | 54.172 | 70.856 | 0.000 |
| 12 | 21 | (g) | -53 | 21 | 21 | 3 | 3 | 102 | 101 | 0.000 | 0.000 | 221.754 | 221.754 | 0.000 |
| 13 | 21 | (g) | -51 | 8 | 0 | 24 | 24 | 105 | 107 | -52.074 | 4.273 | 6.313 | 52.629 | 0.000 |
| 14 | 21 | (g) | -51 | 8 | 0 | 22 | 23 | 107 | 103 | -6.634 | -3.561 | 0.003 | 7.530 | 0.000 |
| 15 | 21 | (g) | -53 | 18 | 18 | 9 | 9 | 104 | 103 | 0.000 | -0.000 | -101.447 | 101.447 | 0.000 |
| 16 | 21 | (g) | -51 | 10 | 0 | 40 | 41 | 108 | 106 | 47.210 | -14.543 | 51.907 | 71.656 | 0.000 |
| 17 | 21 | (g) | -51 | 10 | 0 | 46 | 47 | 104 | 108 | 29.487 | -5.064 | 36.338 | 47.071 | 0.000 |
| 18 | 21 | (g) | -53 | 48 | 48 | 15 | 15 | 104 | 103 | 0.000 | -0.000 | -101.548 | 101.548 | 0.000 |
| 19 | 21 | (g) | -51 | 7 | 0 | 25 | 26 | 109 | 105 | -62.787 | 25.463 | -28.139 | 73.364 | 0.000 |
| 20 | 21 | (g) | -51 | 7 | 0 | 73 | 74 | 102 | 109 | -0.573 | -1.321 | 1.480 | 2.065 | 0.000 |
| 21 | 21 | (g) | -53 | 33 | 33 | 12 | 12 | 102 | 101 | 0.000 | 0.000 | 223.623 | 223.623 | 0.000 |
| 22 | 21 | (g) | -51 | 14 | 0 | 49 | 50 | 110 | 103 | -0.705 | -1.401 | 0.334 | 1.603 | 0.000 |
| 23 | 21 | (g) | -51 | 14 | 0 | 36 | 36 | 107 | 110 | -14.614 | -1.448 | 0.723 | 14.704 | 0.000 |
| 24 | 21 | (g) | -52 | 13 | 13 | 27 | 27 | 105 | 107 | -43.389 | 3.560 | 5.260 | 43.852 | 0.000 |
| 25 | 21 | (g) | -51 | 19 | 0 | 75 | 75 | 109 | 111 | -4.407 | 2.740 | -2.866 | 5.928 | 0.000 |
| 26 | 21 | (g) | -51 | 19 | 0 | 28 | 29 | 111 | 105 | -59.091 | 22.781 | -25.187 | 68.155 | 0.000 |
| 27 | 21 | (g) | -52 | 24 | 24 | 30 | 30 | 105 | 107 | -42.678 | 3.502 | 5.174 | 43.133 | 0.000 |
| 28 | 21 | (g) | -51 | 26 | 0 | 97 | 98 | 111 | 112 | -33.899 | 13.955 | -15.803 | 39.920 | 0.000 |
| 29 | 21 | (g) | -51 | 26 | 0 | 37 | 38 | 112 | 105 | -25.643 | 8.863 | -9.329 | 28.691 | 0.000 |
| 30 | 21 | (g) | -52 | 27 | 27 | 34 | 35 | 105 | 107 | -42.227 | 3.465 | 5.119 | 42.677 | 0.000 |

# Project and its applications

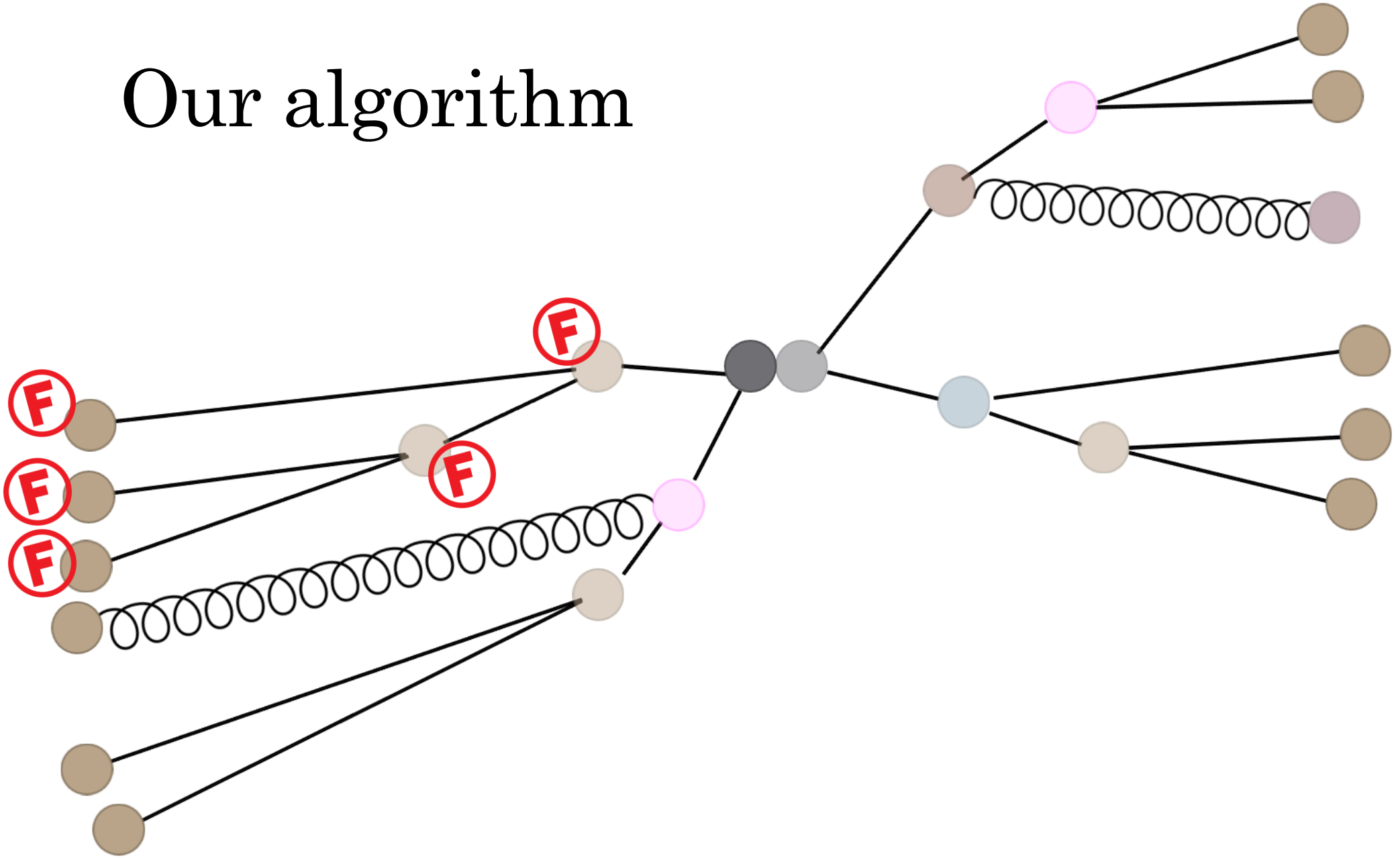
- Our main goal is to be able to visualize an entire collision from the beggining using the data from a pythia event.
- Its main applications:

1. For education purposes
2. Analyzing jet reconstruction theories' consistency
3. Better understanding of how today's theories reconstruct jets and what it looks like

# The class that makes everything possible

```
class part{
        public:
                part();
                part(int);
                part(int,int,int);
                ~part();
                int &index();
                int indexd(int);
                part *ret_daughter(int);
                void add_daughter(int);
                part operator=(part);
                void check_stuff();
                bool &finalstate(){return isfinal;}
                part *ret_mother();
        private:
                int ind;
                bool isfinal;
                part *d1;
                part *d2;
                part *mother;
};
```

# Our algorithm



# Where we're at