Toy Generative Model for Jets

Kyle Cranmer[1], Sebastian Macaluso[1] and Duccio Pappadopulo[2]

1 Center for Cosmology and Particle Physics & Center for Data Science, New York University, USA
2 Bloomberg LP, New York, NY 10022, USA.

1 Introduction

In this notes, we provide a standalone description of a generative model to aid in machine learning (ML) research for jet physics. The motivation is to build a model that has a tractable likelihood, and is as simple and easy to describe as possible but at the same time captures the essential ingredients of parton shower generators in full physics simulations. The aim is for the model to have a python implementation with few software dependencies.

Parton shower generators are software tools that encode a physics model for the simulation of jets that are produced at colliders, e.g. the Large Hadron Collider at CERN. Jets are a collimated spray of energetic charged and neutral particles. Parton showers generate the particle content of a jet, going through a cascade process, starting from an initial unstable particle. In this description, there is a recursive algorithm that produces binary splittings of an unstable parent particle into two children particles, and a stopping rule. Thus, starting from the initial unstable particle, successive splittings are implemented until all the particles are stable (i.e. the stopping rule is satisfied for each of the final particles). We refer to this final particles as the jet constituents.

As a result of this *showering process*, there could be many latent paths that may lead to a specific jet (i.e. the set of constituents). Thus, it is natural and straightforward to represent a jet and the particular showering path that gave rise to it as a binary tree, where the inner nodes represent each of the unstable particles and the leaves represent the jet constituents.

2 Model description

Our model implements a recursive algorithm to generate a binary tree, whose leaves are the jet constituents. Jet constituents in full physics simulations are described in terms of a 4 dimensional vector that specifies their energy E and spatial momentum $\vec{p}$, which determines the direction of motion. We want our model to represent the following features:

- Momentum conservation: the total momentum of the jet (the momentum of the root of the tree) is obtained from adding the momentum of all of its constituents.
- Running of the splitting scale: each splitting is characterized by a scale t that decreases when evolving down the tree from root to leaves. In particular t is the invariant mass squared, $t = m^2$.

We also want our model to lead to a natural analogue of the generalized k_t clustering algorithms for the generated jets. These algorithms are characterized by

- Permutation invariance: the jet momentum should be invariant with respect to the order in which we cluster its constituents.
- Distance measure: the angular separation between two jet constituents is typically used as a distance measure among them. In particular, traditional jet clustering algorithms are based on a measure given by $d_{ij} \propto \Delta R_{ij}^2$ where ΔR_{ij} is the angular separation between two particles.

We build our model as follows. During the generative process, starting from the root of the tree, each parent node is split, generating a left (L) and a right (R) child. At each splitting we sample squared invariant masses for the children, t_L, t_R from a decaying exponential. We require the constraint $\sqrt{t_L} + \sqrt{t_R} < \sqrt{t_P}$, where $\sqrt{t_P}$ is the parent mass. Then we implement a 2-body decay in the parent center-of-mass frame. The children direction is obtained by uniformly sampling a unit vector on the 2-sphere (in the parent center-of-mass frame the children move in opposite directions). Finally, we apply a Lorentz boost to the lab frame, to obtain the 4 dimensional vector $p_\mu = (E, p_x, p_y, p_z)$ that characterizes each node. This prescription ensures *momentum conservation* and *permutation invariance*. We show in Fig. 1 a tree visualization plot of a sample jet generated with our model.

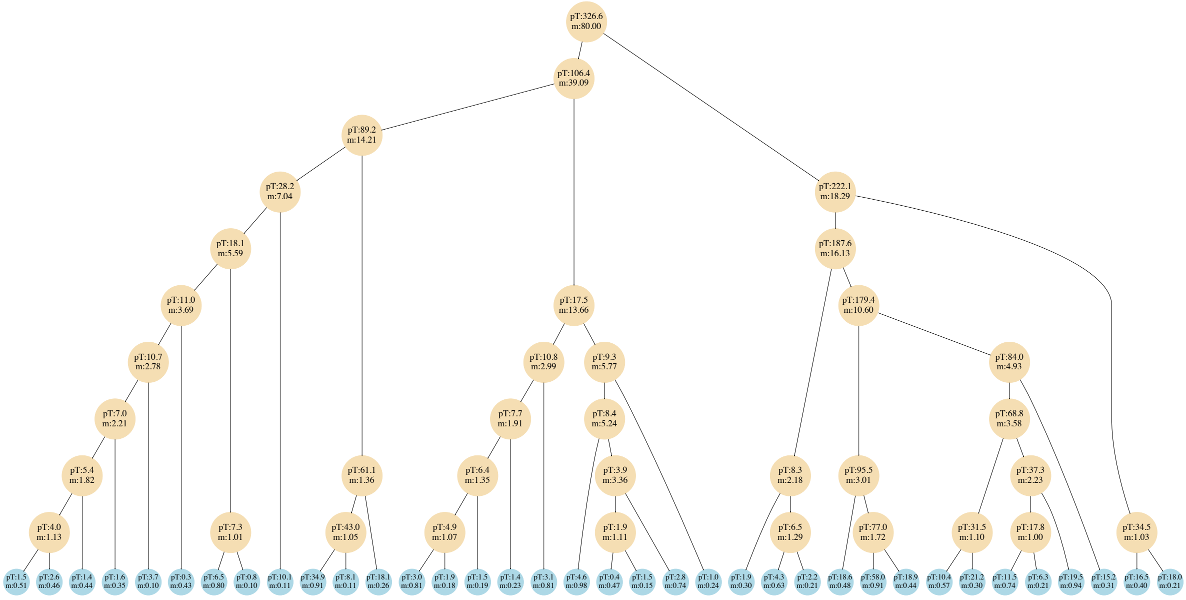

Figure 1: Tree visualization of a sample jet generated with our model that represents a W boson jet, as described in section 2.1.1. We show the values of $p_T = \sqrt{p_x^2 + p_y^2}$ for each node and their mass. The horizontal ordering of the leaves corresponds to the order in which the leaves are accessed when traversing the tree (and is not related to the particle momentum $\vec{p}$).

2.1 Generative process

In this section, we describe the implementation of the generative process, which depends on the following input parameters:

- p_0^μ: 4-momentum of the jet. This will be the input value for the root node of the tree.
- t_0: initial mass squared.
- t_{cut}: cut-off mass squared to stop the showering process.
- λ: decaying rate for the exponential distribution.

Next, we describe the splitting of a node as follows:

1. Draw t_{L} and t_{R} from an exponential distribution as follows,

$$t_L \sim f(t|\lambda, t_{\text{P}}) = \frac{1}{1 - e^{-\lambda}} \frac{\lambda}{t_{\text{P}}} e^{-\frac{\lambda}{t_{\text{P}}} t} \tag{1}$$

$$t_R \sim f(t|\lambda, t_\mathrm{P}, t_L) = \frac{1}{1-e^{-\lambda}} \frac{\lambda}{(\sqrt{t_\mathrm{P}}-\sqrt{t_L})^2} e^{-\frac{\lambda}{(\sqrt{t_\mathrm{P}}-\sqrt{t_L})^2}t} \tag{2}$$

and define $m_L = \sqrt{t_\mathrm{L}}$ and $m_R = \sqrt{t_\mathrm{R}}$. We apply a veto on sampled values where $t_L \geqslant t_\mathrm{P}$ and $t_R \geqslant (\sqrt{t_\mathrm{P}}-\sqrt{t_L})^2$.

2. Compute a 2-body decay in the parent rest frame, where its momentum is $p^\mu_p = p^\mu_\mathrm{L} + p^\mu_\mathrm{R} = (\sqrt{s}, 0, 0, 0)$. From requiring 4-momentum conservation, the children energies are given by

$$\begin{aligned} E_\mathrm{L} &= \frac{\sqrt{s}}{2}\left(1 + \frac{t_\mathrm{L}}{s} - \frac{t_\mathrm{R}}{s}\right) \\ E_\mathrm{R} &= \frac{\sqrt{s}}{2}\left(1 + \frac{t_\mathrm{R}}{s} - \frac{t_\mathrm{L}}{s}\right) \end{aligned} \tag{3}$$

and the magnitude of their 3-momentum by

$$|\vec{p}| = \frac{\sqrt{s}}{2}\bar{\beta} = \frac{\sqrt{s}}{2}\sqrt{1 - \frac{2(t_\mathrm{L}+t_\mathrm{R})}{s} + \frac{(t_\mathrm{L}-t_\mathrm{R})^2}{s^2}} \tag{4}$$

Thus, in the parent rest frame, the left and right child momentum is given by $p^\mu_\mathrm{L} = (E_\mathrm{L}, \vec{p})$ and $p^\mu_\mathrm{R} = (E_\mathrm{R}, -\vec{p})$.

3. Apply a Lorentz boost to each of the children, with $\gamma = \frac{E_p}{\sqrt{t_P}}$ and $\gamma\beta = |\vec{p}_p|/\sqrt{t_P}$.

4. If t_L (t_R) is greater than t_cut repeat the process.

The algorithm is outlined in more detail in Algorithm 1. After running the algorithm, the final binary tree for the jet is obtained.

2.1.1 Heavy resonance vs QCD like jet

To model a jet coming from a heavy resonance X decay, e.g. a W boson jet, we introduce two values for the decaying constant λ_X, λ. This way we model the first splitting (the root node splitting) with λ_X and then use λ for the remaining shower process. We also set $t_\mathrm{root} = m_X^2$.

To model a QCD like jet, we use a single value for λ for the whole process.

2.1.2 Likelihood reconstruction

We reconstruct the parent momentum by adding the children momentum.

$$p^\mu_\mathrm{p} = p^\mu_\mathrm{L} + p^\mu_\mathrm{R} \tag{5}$$

Algorithm 1: Toy Parton Shower Generator

```
function NodeProcessing ($p^{\mu}_{\mathrm{p}}, t_{\mathrm{P}}, t_{\mathrm{cut}}, \lambda, \mathrm{tree}$)
Input : parent momentum $\vec{p}_{\mathrm{p}}$, parent mass squared $t_{\mathrm{p}}$, cut-off mass squared
        $t_{\mathrm{cut}}$, rate for the exponential distribution $\lambda$, binary tree *tree*
    Add parent node to tree.
    if $t_p > t_{cut}$ then
        Sample $t_L$ and $t_R$ from the decaying exponential distribution.
        Sample a unit vector from a uniform distribution over the 2-sphere.
        Compute the 2-body decay of the parent node in the parent rest frame.
        Apply a Lorentz boost to the lab frame to each child.
        NodeProcessing ($p^{\mu}_{\mathrm{p}}, t_{\mathrm{L}}, t_{\mathrm{cut}}, \lambda, \mathrm{tree}$)
        NodeProcessing ($p^{\mu}_{\mathrm{p}}, t_{\mathrm{R}}, t_{\mathrm{cut}}, \lambda, \mathrm{tree}$)
```

Given each 4-momentum vector $p^{\mu} = (E, \vec{p})$, we find the invariant mass squared as $t = E^2 - |\vec{p}|^2$. Then we identify t_L with the child with greater invariant mass and find the likelihood of a parent going to t_L and t_R from Eqs. 1 and 2:

$$\ell = f(t_L|\lambda, t_{\mathrm{P}}) \cdot f(t_R|\lambda, t_{\mathrm{P}}, t_L) \tag{6}$$

For the leaves, we find the probability of having sampled a value of $t < t_{\mathrm{cut}}$.

2.1.3 Upper Bound on trees Likelihood

Below we show how to obtain an upper bound on a jet likelihood. This could be useful to implement the A* search algorithm on Ginkgo jets.

a) **Inner nodes**

$$f(t|\lambda, t_{\mathrm{P}}) = \frac{1}{1 - e^{-\lambda}} \frac{\lambda}{t_{\mathrm{P}}} e^{-\frac{\lambda}{t_{\mathrm{P}}} t} \tag{7}$$

We maximize this function by taking $t \to 0$ and $t_{\mathrm{P}} = t_{\mathrm{cut}}$.

b) **Leaves**

$$f(t|\lambda, t_{\mathrm{P}}) = \frac{1}{1 - e^{-\lambda}} \left(1 - e^{-\frac{\lambda}{t_{\mathrm{P}}} t_{\mathrm{cut}}}\right) \tag{8}$$

We maximize it by taking $t_{\mathrm{Pi}} = \min_{\{j \neq i,\, j \text{ in leaves}\}} t(p_i + p_j)$