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 Δ: each splitting is characterized by a scale that decreases when evolving down the tree from root to leaves.

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.

As a result, we build our model as follows. Each node of the jet tree represents a particle and encodes its momentum vector. 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. The L (R) child's momentum is obtained from subtracting (adding) a vector of magnitude Δ to half of the parent's momentum vector. This prescription ensures *momentum conservation* and *permutation invariance*.

We consider a 2D model to be able to define an angular *distance measure*. Also, the angular separation between a parent and its L/R child's momentum is expected to decrease as the physics shower moves forward from the root to the leaves of the tree. The magnitude of the momentum of each node is also expected to drop in the same way, i.e. "the radiation gets softer when evolving down the tree". Both requirements are satisfied by the *running of the splitting scale* Δ, which we achieve by rescaling the value of Δ by a factor r. (r is drawn from a decaying exponential distribution each time

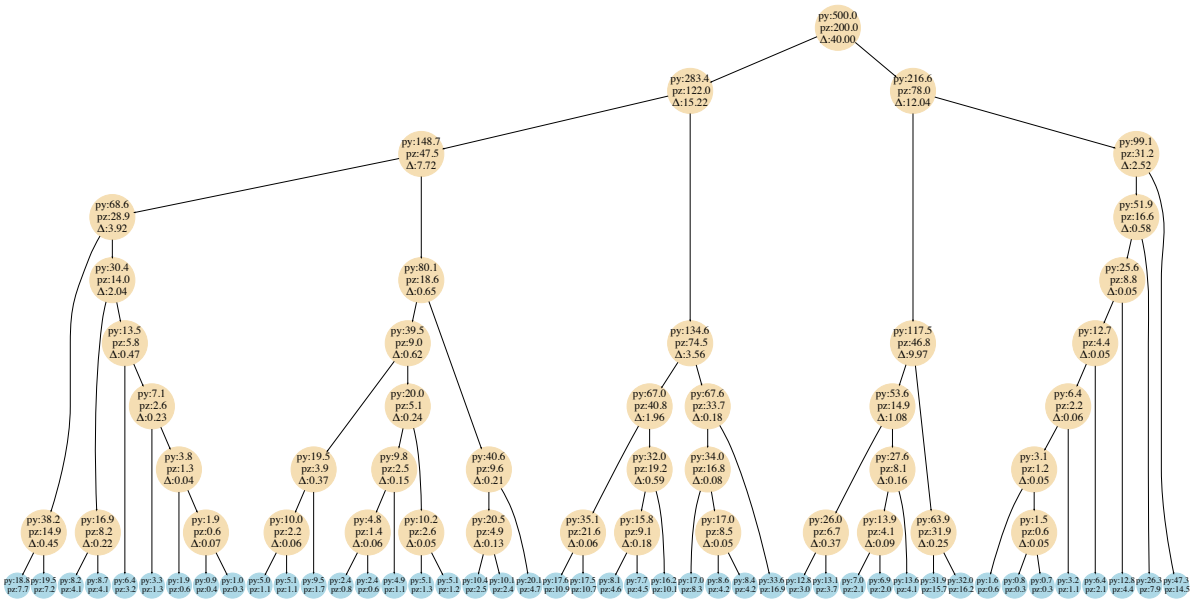

Figure 1: Tree visualization of a sample jet generated with our model that represents a W boson jet, as described in section 2.1.1. We show the values of $\vec{p} = (p_y, p_z)$ for each node and the scale Δ for the splitting of the inner nodes. The horizontal ordering of the leaves corresponds to the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

a new splitting is produced.) This way, we also assign a Δ value to each node of the tree.

We build our 2D model in the (y, z) plane, where $\hat{z}$ is the direction of the beam axis and $\hat{y}$ the transverse direction.[1] We define the transverse momentum as $p_{\mathrm{T}} = |p_y|$. We show in Fig. 1 a tree visualization plot of a sample jet generated with our model.

2.1 Generative process

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

- $\vec{p}_0$: momentum of the jet. This will be the input value for the root node of the tree.
- λ: decaying rate for the exponential distribution.

[1] At the Large Hadron Collider, jets are produced from the collision of two beams of protons moving in opposite directions.

- Δ_0: Initial scale of the splitting.
- Δ_{cut}: cut-off scale to stop the showering process.

Next, we describe the splitting of a node. Given a parent node with momentum $\vec{p}_{\text{p}}$, and a scalar value Δ_{p} that sets the scale of the splitting, we define the splitting function as follows:

1. We draw a value ϕ_{p} for the angle in the (y,z) plane, from a uniform distribution in the range $\{0, 2\pi\}$ and define

$$\vec{\Delta}_{\text{p}} = \Delta_{\text{p}} \ (\sin\phi_{\text{p}}, \cos\phi_{\text{p}}) \tag{1}$$

2. We obtain the L,R children nodes momentum,

$$\begin{aligned} \vec{p}_{\text{L}} &= \frac{1}{2}\vec{p}_{\text{p}} - \vec{\Delta}_{\text{p}} \\ \vec{p}_{\text{R}} &= \frac{1}{2}\vec{p}_{\text{p}} + \vec{\Delta}_{\text{p}} \end{aligned} \tag{2}$$

3. We separately draw r_{L} and r_{R} from an exponential distribution

$$f(r|\lambda) = \lambda e^{-\lambda r} \tag{3}$$

and generate

$$\begin{aligned} \Delta_{\text{L}} &= \Delta_{\text{p}} \ r_{\text{L}} \\ \Delta_{\text{R}} &= \Delta_{\text{p}} \ r_{\text{R}} \end{aligned} \tag{4}$$

We start the showering process with the root node as the parent node and build the jet binary tree recursively as follows:

- We split the parent node, and get $\vec{p}_{\text{L}}$, $\vec{p}_{\text{R}}$, Δ_{L}, Δ_{R}.
- If $\Delta_{\text{L/R}} > \Delta_{\text{cut}}$, we promote the L/R node to a parent node (i.e. we promote $\vec{p}_{\text{L/R}}$ to $\vec{p}_{\text{p}}$ and $\Delta_{\text{L/R}}$ to Δ_{p}), and split again.

The algorithm is outlined in more detail in Algorithm 1. After running the algorithm, the final lists with the tree structure (*tree*) and momentum of each node (*content*) are obtained.

Algorithm 1: Toy Parton Shower Generator

function Exp2DShower $(\vec{p}_{\rm p}, \Delta_{\rm p}, \Delta_{\rm cut}, \lambda, \text{tree})$

Input : parent momentum $\vec{p}_{\rm p}$, parent splitting scale $\Delta_{\rm p}$, cut-off scale $\Delta_{\rm cut}$, rate for the exponential distribution λ, binary tree *tree*

Add parent node to tree.

if $\Delta_p > \Delta_{cut}$ **then**

draw ϕ from uniform distribution in $\{0, 2\pi\}$

$\vec{\Delta}_{\rm p} = \Delta_{\rm p}\ (\sin\phi_{\rm p}, \cos\phi_{\rm p})$

$\vec{p}_{\rm L} = \frac{1}{2}\vec{p}_{\rm p} - \vec{\Delta}_{\rm p}$

$\vec{p}_{\rm R} = \frac{1}{2}\vec{p}_{\rm p} + \vec{\Delta}_{\rm p}$

draw r_L, r_R from the exponential distribution in Eq. 3.

$\Delta_{\rm L} = \Delta_{\rm p}\ r_{\rm L}$

$\Delta_{\rm R} = \Delta_{\rm p}\ r_{\rm R}$

Exp2DShower $(\vec{p}_{\rm L}, \Delta_{\rm L}, \Delta_{\rm cut}, \lambda, \text{tree})$

Exp2DShower $(\vec{p}_{\rm R}, \Delta_{\rm R}, \Delta_{\rm cut}, \lambda, \text{tree})$

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 use $\Delta_{\rm p} = \frac{m_X}{2}$ to split the root node.

To model a QCD like jet, we split the root node with $\Delta_{\rm p} = \Delta_0\ r_{\rm root}$, where $r_{\rm root}$ is drawn from the exponential distribution (3).

2.1.2 Reconstruct $\{\vec{p}_{\mathbf{p}}, \Delta_{\mathbf{p}}, \phi_{\mathbf{p}}, r_L, r_R\}$ from $\{\Delta_L, \Delta_R, \vec{p}_{\mathbf{L}}, \vec{p}_{\mathbf{R}}\}$

Next, we show how to reconstruct $\{\vec{p}_{\rm p}, \Delta_{\rm p}, \phi_{\rm p}, r_L, r_R\}$ from a bottom-up approach for each splitting. From $\{\Delta_L, \Delta_R, \vec{p}_{\rm L}, \vec{p}_{\rm R}\}$ we can obtain

$$\begin{aligned} \vec{p}_{\rm p} &= \vec{p}_{\rm L} + \vec{p}_{\rm R} \\ \vec{\Delta}_{\rm p} &= \frac{1}{2}(\vec{p}_{\rm R} - \vec{p}_{\rm L}) \end{aligned} \tag{5}$$

which can be used to calculate

$$\begin{aligned} \Delta_{\rm p} &= |\vec{\Delta}_{\rm p}| \\ \phi_{\rm p} &= \tan^{-1} \frac{(\Delta_{\rm p})_y}{(\Delta_{\rm p})_z} \\ r_{\rm L} &= \frac{\Delta_{\rm L}}{\Delta_{\rm p}} \\ r_{\rm R} &= \frac{\Delta_{\rm R}}{\Delta_{\rm p}} \end{aligned} \tag{6}$$

If the L/R node is a leaf, then we set

$$\Delta_{\rm L/R} = \Delta_{\rm cut} \tag{7}$$

2.2 Other ideas for Toy Generative Models

We could define $\vec{\Delta}$, following the kinematics of a 2-body decay, as :

$$\vec{\Delta} = E\sqrt{1 - \frac{m^2}{E^2}}\ [\hat{r} + (\gamma - 1)(\hat{r} \cdot \hat{n})\hat{n}] = \vec{\Delta}(m, \phi) \tag{8}$$

We draw ϕ from a uniform distribution. We can get m in a way to resemble the sudakov factor approach as:

$$m_{\rm L} = m_{\rm R} = \frac{m_{\rm p}}{2}\ r \tag{9}$$

where $r_{\rm L/R}$ is drawn from an exponential distribution as in (3) with $r \in [0, 1)$. The prescription of (9) solves one of the problems of the traditional parton showers given that it satisfies $m_{\rm L} + m_{\rm R} \leq m_{\rm p}$[2]. Then, we could think of $m_{\rm L/R}$ as the off-shell mass value, to avoid the required reshuffling. This results in leaves where each pair of siblings have the same mass, all different among pairs of siblings.

This case would be closer to a physics parton shower but more complex and time consuming.

2.2.1 General case, with $m_{\mathbf{L}} \neq m_{\mathbf{R}}$

We could also build a model for this case, adding extra features. We should replace (9) by

$$\begin{aligned} m_{\rm L} &= m_{\rm p}\ r_{\rm L} \\ m_{\rm R} &= (m_{\rm p} - m_{\rm L})\ r_{\rm R} \end{aligned} \tag{10}$$

[2]Traditional parton showers draw $m_{\rm L}$ and $m_{\rm R}$ independently, so they should check the constraint is satisfied.

where r_{L} and r_{R} are independently drawn from an exponential distribution as in (3) for $r \in [0, 1)$.