# Note of *Monte Carlo and Jet Tutorial*

Tong Ou

# 1 Basic Concepts

## 1.1 Parton Shower and Hadronization

### 1.1.1 Physics Principles

In LHC collisions, single partons are emitted from the hard scattering. These partons continue to split into mutiple partons through the QCD process, which finally produces the Parton Shower. In the process shown in Figure 1, it can be shown that the $n+1$ leg Feynman diagram ($\mathcal{M}_{n+1}$) can be built from the $n$ leg Feynman diagram ($\mathcal{M}_n$) through:

$$|\mathcal{M}_{n+1}|^2 \sim \frac{1}{\theta^2} C_A F(z) \,|\mathcal{M}_n|^2 \tag{1}$$

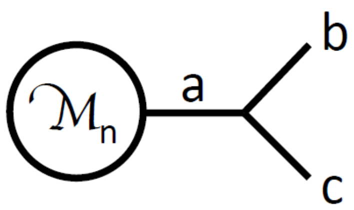



Figure 1: The Feynman diagram representing an $n+1$ leg matrix element $\mathcal{M}_{n+1}$, built from an $n$ leg process $\mathcal{M}_n$ and an additional parton splitting.

Where $z = \frac{E_b}{E_a}$ is the relative energy of the more energetic parton after splitting to that of the initial one and $\theta$ is the opening angle of the splitting between $b$ and $c$. For the $ggg$ and $qqg$ vertices, $F(z)$ generally follows a functional form of $1/z$. The $1/z$ and $1/\theta$ components of Eq (1) indicate the preference for low energy and low opening angle in the splitting process.

When the energy scale becomes low enough, splitting is no longer supported, while the hadronization among partons occurs. This means the partons with low energy would combine to an ensemble of hadrons.

It should be noted that in both the processes of splitting and hadronization, randomness exists due to the random nature of Quantum Mechanics.

### 1.1.2 Generation of Parton Shower by MC Methods

There are two stochastic factors in the generation of Parton Shower:

- $z$: To avoid the singularity of $1/z$ at $z=0$, we set the $PDF(z) = \frac{1}{z+1}$ and allow the range of possible $z$ value spans $[0,1]$.

- $\theta$: Also to avoid the singularity, we set $PDF(\theta) = \frac{1}{\theta+1}$ and allow the range of possible $\theta$ value spans $[0, \pi/2]$. (In fact, I am not sure why $PDF(\theta)$ should be set to $1/\theta$ instead of $1/\theta^2$ as indicated in Section 1.1.1. However, I tried both of these $PDF$s and saw no significant differences between the results.)

Thus, the key point of generating a Parton Shower is to generate a sequence of $[z, \theta]$ pairs according to the $PDF$s with the Monte Carlo methods.

## 1.2 Monte Carlo Simulation

There are two methods to generate a list of random numbers with some $PDF$s from a list of evenly distributed random numbers:

- Accept-Reject Monte Carlo: Generate an extra evenly distributed random number $y$ with the desired random number $x$. Compare $y$ with normed $PDF(x)$ to decide whether $x$ should be accepted.

- Inverse Transform Method: Generate an evenly distributed random number $y$. Get the desired random number $x$ through solving the Equation $y = CDF(x) = \int_{x_{min}}^{x} PDF(x)dx$.

## 1.3 Jet Reconstruction

### 1.3.1 Jet Finding

Given the final states (pseudojets) of a Parton Shower, a set of four momentum vectors of partons, we are able to find jets using the *jet clustering algorithm*. Two variables are defined in this algorithm: the distance between any two pseudojets $d_{ij}$ and the distance between a pseudojet and the beam $d_{iB}$:

$$d_{ij} = min(k_{ti}^{2p}, k_{tj}^{2p})\frac{\Delta_{ij}^2}{R^2} \tag{2}$$

$$d_{iB} = k_{ti}^{2p} \tag{3}$$

The procedures for performing jet finding are:

- Calculate the $d_{ij}$ and $d_{iB}$ matrices. Loop over these two matrices to find the minimum value.

- If the minimum value is $d_{iB}$, pick out the corresponding pseudojet to become a real jet and no longer consider it in the following comparison.

- If the minimum value is $d_{ij}$, combine the pseudojets $i$ and $j$ to get a new pseudojet and recalculate the $d_{ij}$ and $d_{iB}$ matrices.

- Stop when all pseudojets are combined to jets.

The procedures above are for inclusive jet finding. Complementarily, the exclusive jet finding procedures are similar with the inclusive ones except the differences[1]: (a) when a $d_{iB}$ is the smallest value, that particle is considered to become part of the beam jet (i.e., is discarded); (b) clustering is stopped when all $d_{ij}$ and $d_{iB}$ are above some $d_{cut}$. The exclusive jet finding is commonly performed with $p \geq 0$ and $R = 1$.

[1] Cacciari M, Salam G P, Soyez G. FastJet user manual[J]. The European Physical Journal C, 2012, 72(3): 1896.

#### 1.3.2 Jet Observables

Since the jets are built from combination of initial pseudojets (jet constituents), the observables of jets should be functions taking the set of jet constituents as arguments, i.e., $O(Jet) = function([j_1, j_2, ..., j_n])$, where $j_i$ represents the jet constituent. For example, the pseudomass of a jet can be computed as:

$$SimpleMass(J) = E(j_1) \times E(j_2) \times \Delta(j_1, j_2) \tag{4}$$

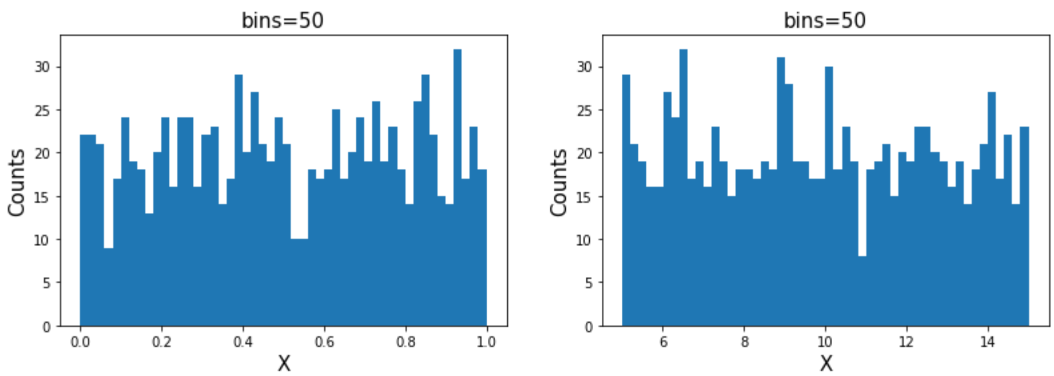
where $j_1$ and $j_2$ are the two highest $p_T$ constituents of the jet $J$.

# 2 Results of Practice in the Tutorial

## 2.1 Basic Monte Carlo Simulation

**Practice 1** [2] Practice writing a program in Python that generates random numbers. In the first case, write a program to generate 1000 random numbers distributed uniformly between [0,1] and make a histogram of these numbers. In the second case, write a program that distributes the random numbers between [5,15] and makes a histogram. Remember, it is only allowed to use the `random.random()` functionality in Python. As a hint, think about how to make the minimum and maximum of the [0,1] distribution into the minimum and maximum of the [5,15] distribution.

**Results:** The histograms of the generated random numbers distributed uniformly between [0,1] and [5,15] are shown in Figure 2.




(a) (b)

Figure 2: Results of Practice 1: (a) Range [0,1]; (b) Range [5,15].

**Practice 2** Practice writing a program in Python that generates random numbers, distributed according to a Gaussian distribution with a mean $\mu$ of 5 and a width $\sigma$ of 2 in the range of [0,10].

**Results:** The random numbers are generated by Accept-Reject Method. The histogram of the generated numbers is shown in Figure 3.

[2] All of the codes can be found here: https://github.com/Tong-Ou/MC-and-Jet-Tutorial/tree/codes

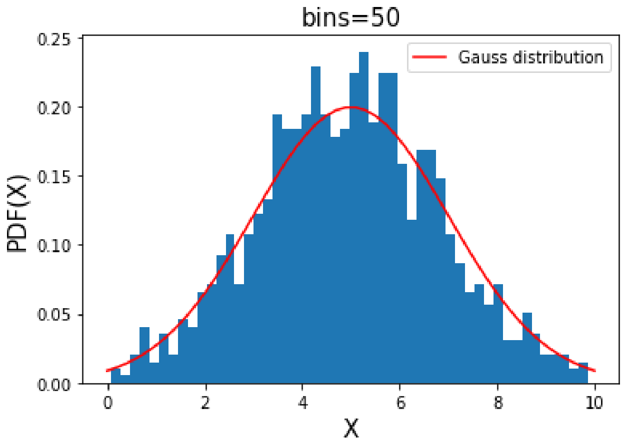



Figure 3: Random numbers distributed according to a Gaussian distribution in the range of [0,10].

**Practice 3** Calculate $\pi$ by accept-reject method. We now want to use statistics to estimate the uncertainty. You already plotted a histogram with several $\pi$ values. One method to calculate and reduce the uncertainty is to calculate $\pi$ several times and determine the mean $\mu$ and standard deviation $\sigma$ of the distribution of calculated $\pi$ values. Increase the number of $\pi$-values you use and examine how $\sigma$ evolves. Another thing that will impact the precision of the determination of $\pi$ is to the number of random points used in each accept-reject calculation. Again calculate several $\pi$ values and get the mean and $\sigma$. Keep the number of $\pi$-values constant and alter the number of random points. How does $\sigma$ change? Does this make sense?

**Results:** (a) Calculate $\pi$ for 1000 times with 1000 random points generated. The histogram is shown in Figure 4. Fit the histogram with a Gauss distribution and the results are: $\mu = 3.144$, $\sigma = 0.0511$.

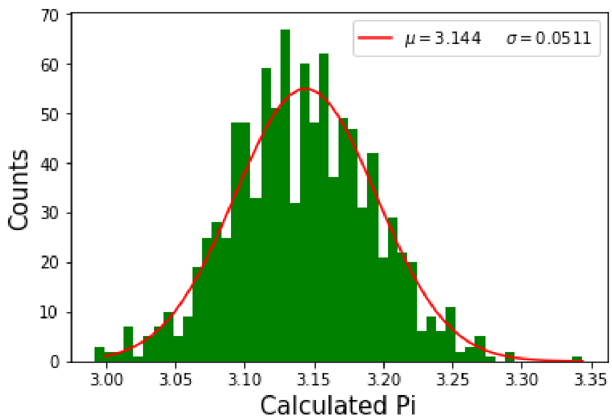



Figure 4: Calculated $\pi$-values. Calculation times: 1000. Random points generated: 1000.

(b) Increase the calculation times to examine how $\mu$ and $\sigma$ evolve. The result is shown in Figure 5. It is observed that $\mu$ and $\sigma$ vary quite irregularly with increase of calculation times.

(c) Alter the number of random number points and the evolutions of $\mu$ and $\sigma$ are shown in Figure 6. Obviously, with the increase of number of random points, $\sigma$ decreases significantly and $\mu$ tends to the real value of $\pi$ (3.14).

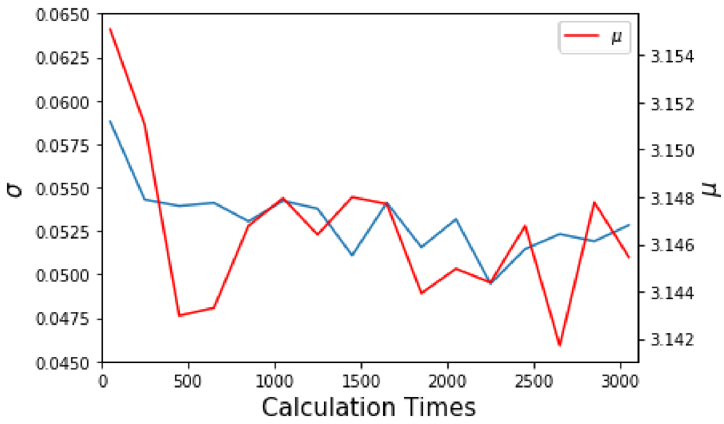



Figure 5: Evolutions of $\mu$ and $\sigma$ with increase of calculation times.

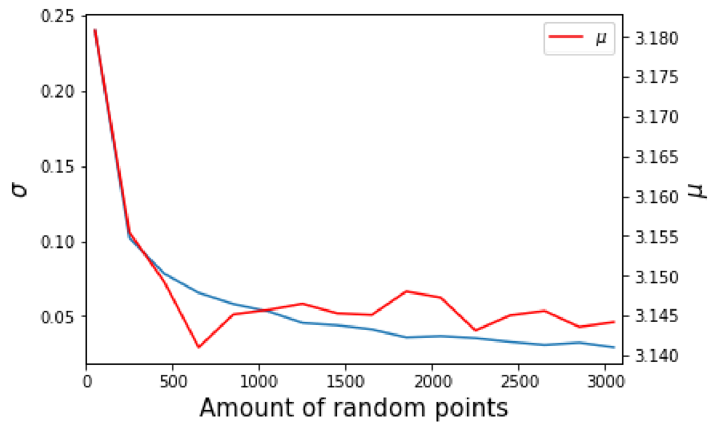



Figure 6: Evolutions of $\mu$ and $\sigma$ with increase of number of random points.

Thus, a preliminary conclusion can be drawn from the results of (b) and (c): The number of random points has a greater impact on the calculation precision in this case than the calculation times.

**Practice 4** Practice writing a program in Python that generates random numbers, distributed according to a falling distribution of the form $e^{-x}$ in the range [0,5]. After doing this, generate the same distribution using the accept-reject method and determine the efficiency of the generation, defined as the fraction of accepted x values with respect to the total generated x values. How does this efficiency compare for the accept-reject method and the inverse transform method?

**Results:** The histogram of the random numbers distributed according to the form $e^{-x}$ is shown in Figure 7.

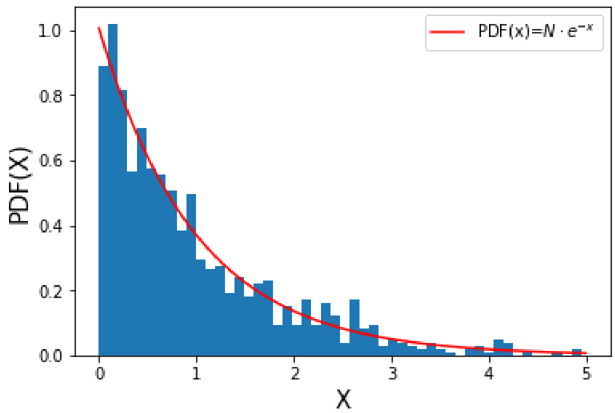



Figure 7: Random numbers distributed according to the form $e^{-x}$.

The efficiency of inverse transform method is 100% as indicated by its definition. The efficiency of accept-reject method is approximately 0.2 in this case. The evolution of the efficiency with increase of accepted $X$ values is shown in Figure 8.

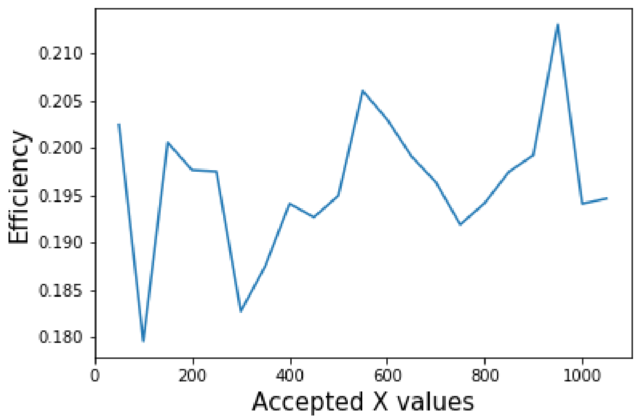



Figure 8: Evolution of efficiency of accept-reject method with increase of accepted $X$ values.

## 2.2 Parton Shower Generation and Jet Reconstruction

**Practice 5** Can you determine which simplifying assumption it is that allows for the full determination of the final state? The main point is to recognize the energy scales we are talking about here in comparison to the typical energy scale of stuff at the LHC (TeV).

**Results:** The simplifying assumption should be $E \approx p$, i.e., neglecting the rest mass of the partons. With this assumption, it is easy to apply conservation of energy and momentum in both Parton Shower generation and Jet Clustering.

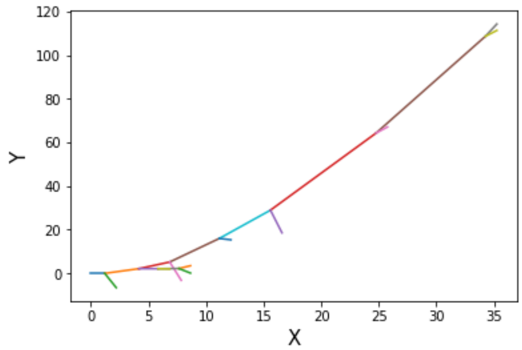
**Practice 6** This parton shower model should be implemented such that, in addition to the output set of four momentum vectors, a graphical representation of the parton shower tree is produced from your program. This takes careful organization of the flow of information and can be achieved in multiple ways.

**Results:** Two assumptions (approximations) are made in the design of the code:

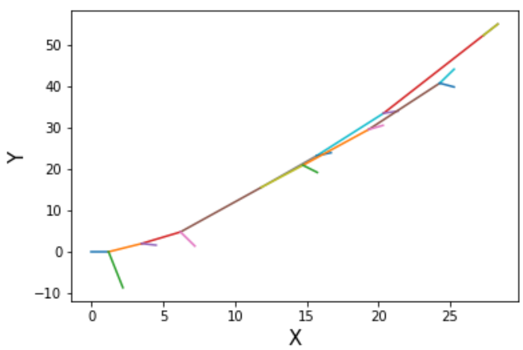
(a) Considering the energy scale $E \gg m_{parton}$, $m_{parton}$ is neglected here and $E \approx p$.

(b) $PDF(\theta) = 1/(\theta + 1)$ implies a preference for small splitting angle in Parton Shower generation, thus it is resonable to apply the small angle approximation $\sin\theta \sim \theta$ and $\cos\theta \sim 1$ in the calculation.

Two of the generated parton shower trees are shown in Figure 9. Most of the results accord with small angle approximation, while some of them behave weirdly with over-large splitting angles and seem to violate the conservation of energy and momentum, which may be caused by the over-simplification made here.




(a)




(b)

Figure 9: Two of the generated parton shower trees. The initial parton departs from [0,0] in both examples.

**Practice 7: Jet Reconstruction** (a) From the previous Monte Carlo model you have created, you have the ability to simulate parton showers/hadronizations that are representative of hadronic activity in LHC collisions. At this stage, implement the clustering algorithm described here to cluster a single event, including proper validation of the procedure. Note that you will have to specify the parameters n and R which are choices of the algorithm. Make a histograma of the number of jets that result from this algorithm for 1000 parton showers and do this for the set of n values of [-1,0,1] and R values of [0.01,0.05,0.1,0.5,1.0]. What differences do you see?

A histogram is a great way to check if your program is doing what it is supposed to do. Think about what this histogram should look like and ask youself Does it make sense?

(b) The algorithm described here is commonly referred to as inclusive jet finding. A complementary algorithm, called exclusive jet finding, exists that performs clustering in a slightly different way and truncates based on a different condition. Find a description of this algorithm and compare and contract it to inclusive jet finding. Implement this algorithm in your program such that it is possible to cluster jets in either fashion.

(c) Based on the jets that you have clustered in the previous section, calculate the jet pseudo-mass as

$$SimpleMass(J) = E(j_1) \times E(j_2) \times \Delta(j_1, j_2) \tag{5}$$

where $j_1$ and $j_2$ are the two highest $p_T$ constituents of the jet $J$. In the case that a given parton shower and clustering produces more than one jet, perform this calculation for the jet of highest energy. Perform this calculation for a large number of parton shower generations and make a histogram of the pseudo-mass for all of these generated showers.

**Results:** The program is composed of several parts:

- Parton Shower generation (in 3D space): Based on the 2-dimensional parton shower model, an additional variable, the azimuthal angle $\phi$ (uniformly distributed in the range $[0, 2\pi]$), is employed to decide the kinematics of the partons. With this additional dimension, a necessary coordinate transformation (from the natural coordinate attached with the splitting parton to the lab frame) is performed every time when two new partons are produced from the splitting.

- Jet Clustering: It is performed for the parton showers generated from last step following the procedures described in Section 1.3.1.

- Calculation of pseudomass: To perform this calculation, an array is used to store the two highest $p_T$ constituents of every pseudojet and is updated every time when two pseudojets combine together. At the same time, the four momentum vectors of the real jets are recorded in another array, from which we find the one with highest energy to calculate its pseudomass.

- Exclusive Jet finding: The key point of the exclusive algorithm is to set an appropriate $d_{cut}$.

Given the same two partons initially coming from the $pp$ collision, the parton shower generation and jet clustering are performed for 1000 times for each pair of $[p, R]$. The histograms for number of jets and pseudomass of jets are shown in Figure 10 to Figure 12.

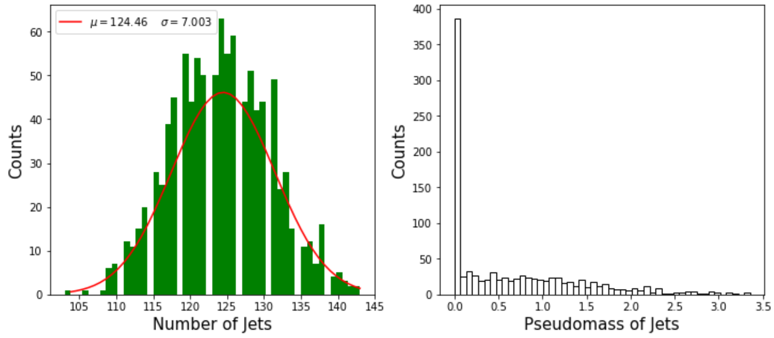
Several features are observed from these results:

(a) Given the same $p$ value, the number of jets significantly decrease with the increase of $R$, correspondingly, the pseudomass of jets increase in this process. This result should be expected from the principles of the jet clustering algorithm, since the parameter $R$ can be roughly interpretated as the "radius" of the jet.

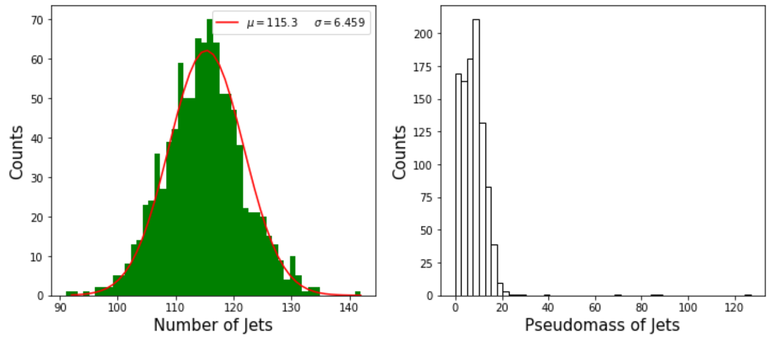
(b) The Gauss fitting results of the histogram show that not only the mean, but also the standard deviation decreases with the increase of $R$. Even though the fitting quality is poor when $R = 0.5$ and $R = 1.0$, the tendency of $\sigma$ decreasing still appears in such cases. It indicates that the clustering is more stable with a larger $R$, which should also be expected from the definition of $R$.

(c) No significant differences among the results of same $R$ but different $p$ are observed in this program. It seems to be reasonable based on the discussion [3] that the $p$ value mainly influences the "shape" of jet instead of number of jets.

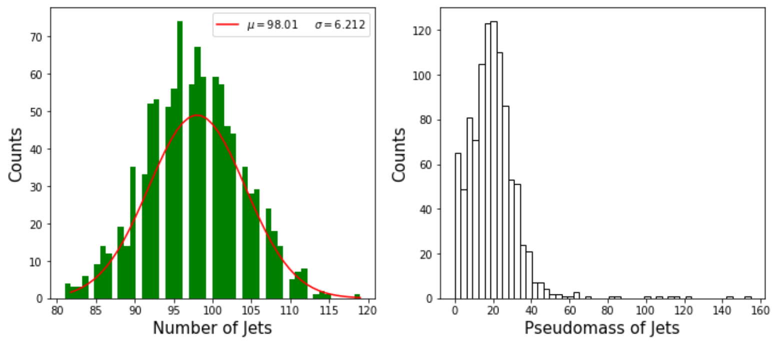
[3] Cacciari M, Salam G P, Soyez G. The anti-$k_t$ jet clustering algorithm[J]. Journal of High Energy Physics, 2008, 04(4):403-410.



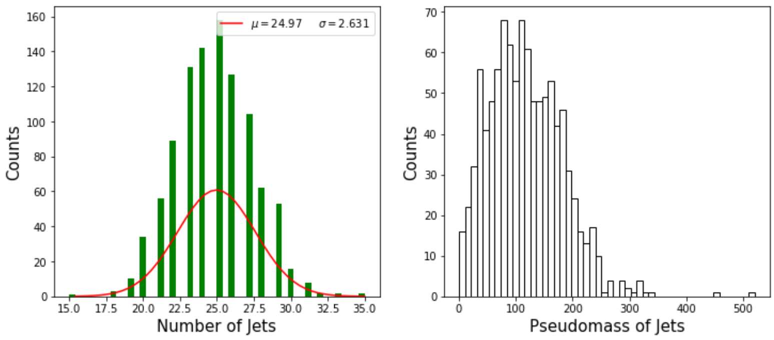

(a) R=0.01



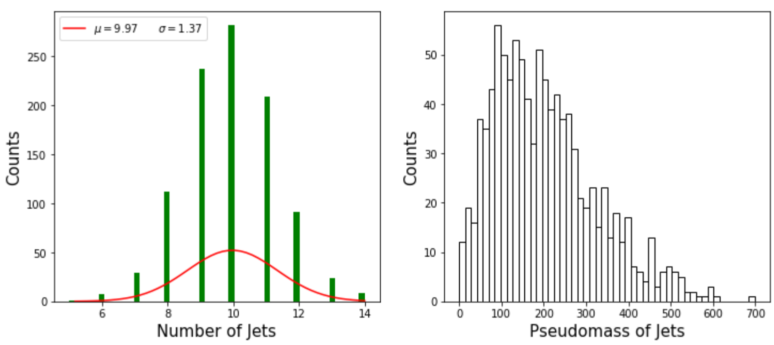

(b) R=0.05




(c) R=0.1

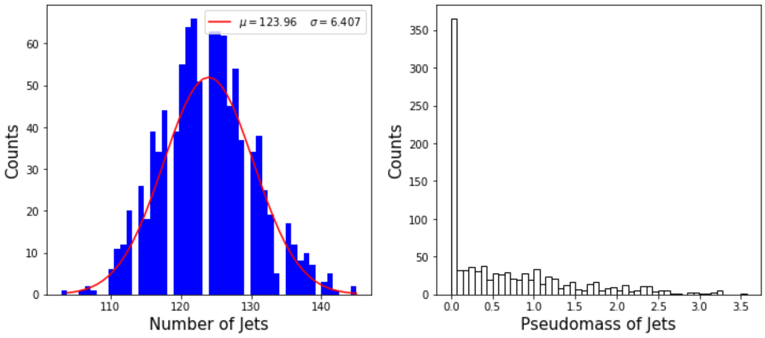



(d) R=0.5

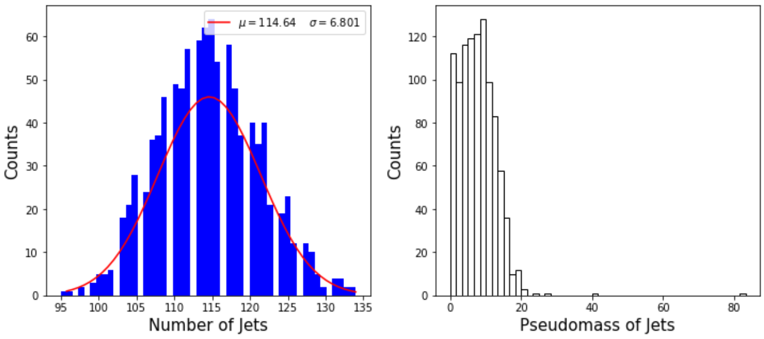



(e) R=1.0

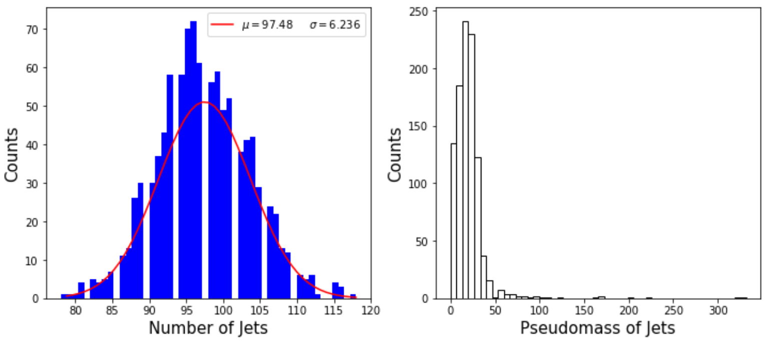
Figure 10: Anti-$k_T$ jet algorithm ($p$=-1)



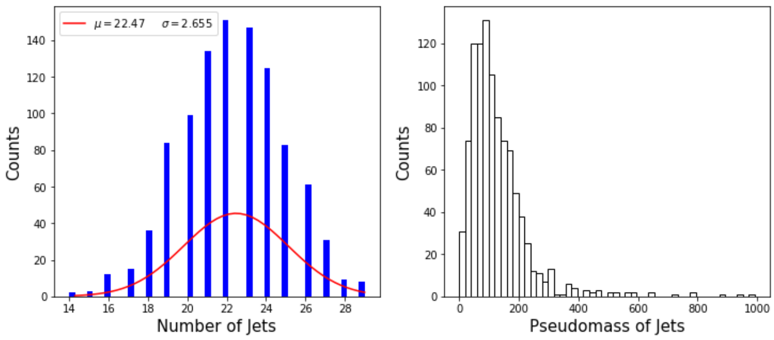

(a) R=0.01



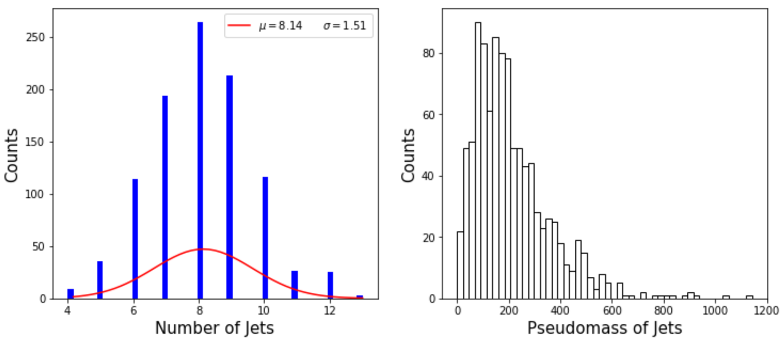

(b) R=0.05




(c) R=0.1

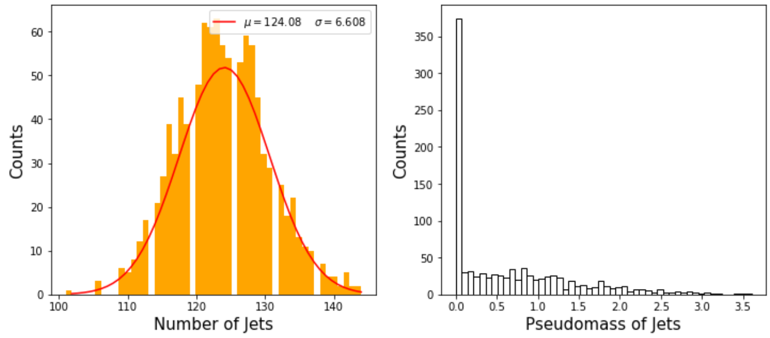



(d) R=0.5

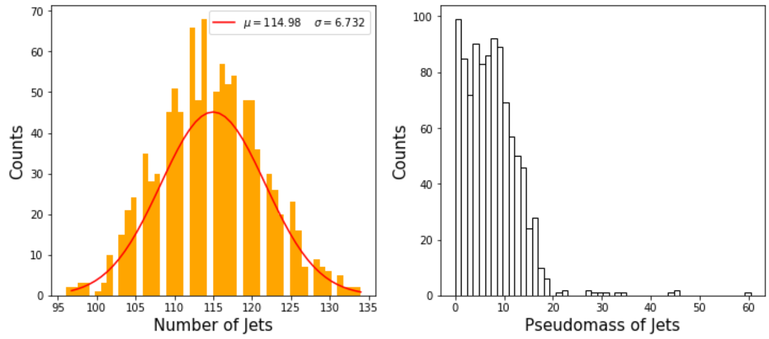



(e) R=1.0

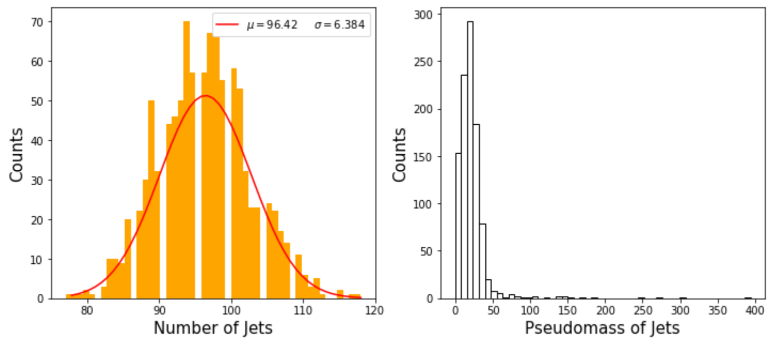
Figure 11: Cambridge/Aachen jet algorithm ($p$=0)



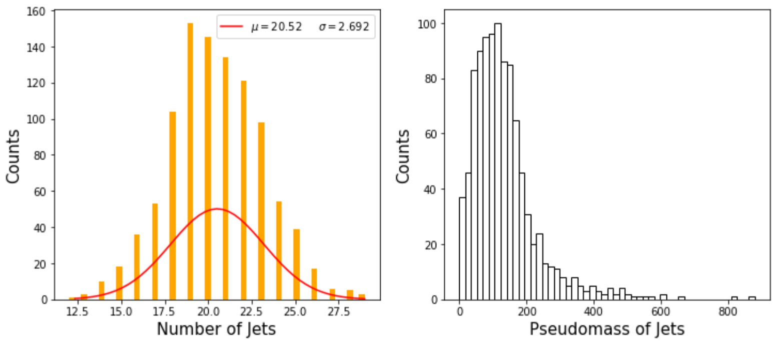

(a) R=0.01



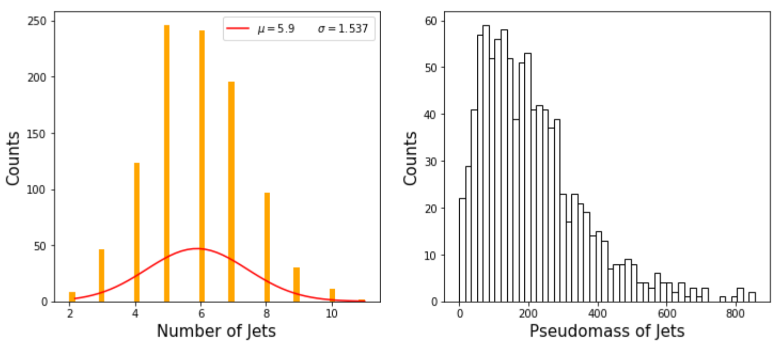

(b) R=0.05




(c) R=0.1



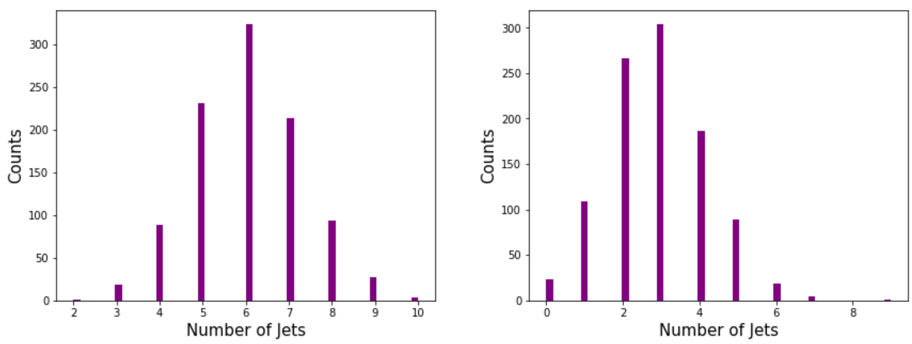

(d) R=0.5




(e) R=1.0

Figure 12: $k_T$ jet algorithm ($p$=1)

Exclusive jet finding is performed for $[p, R] = [0, 1]$ and $[p, R] = [1, 1]$ with $d_{cut} = 5000$. Results are shown in Figure 13. Compared with the same cases of inclusive jet finding, the mean and deviation of number of jets slightly decrease for both cases. However, the results of exclusive jet finding heavily depend on the choice of $d_{cut}$.




(a) $p = 0, R = 1$ (b) $p = 1, R = 1$

Figure 13: Exclusive jet finding.