Developing and testing Unfolding Algorithms

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Nukulsinh Parmar

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

July, 2021

Supervisor: Prof. Florencia Canelli

Certificate

This is to certify that this dissertation entitled Developing and testing Unfolding Algorithms towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Nukulsinh Parmarat Indian Institute of Science Education and Research under the supervision of Prof. Florencia Canelli, Professor, Department of Physics , during the academic year 2016-2021.

Florencia Canelli

Prof. Florencia Canelli

Committee:
Prof. Florencia Canelli
Prof. Seema Sharma

Everyone who reads this thesis, I dedicate this thesis to You!

Declaration

I hereby declare that the matter embodied in the report entitled Developing and testing Unfolding Algorithms are the results of the work carried out by me at the Department of Physics, Indian Institute of Science Education and Research, Pune, under the supervision of Prof. Florencia Canelli and the same has not been submitted elsewhere for any other degree.

2/6/21

Nukulsinh Parmar

Acknowledgments

I want to express my sincere gratitude to my supervisor Prof. Florencia Canelli for providing me with the opportunity to work on the Master's thesis and support and encourage me throughout the project's duration. I am deeply grateful to Dr. Kyle Cormier for his guidance and assistance at every stage of the thesis. I am thankful for his valuable advice, insights and support on various research related topics to communication skills and way of life in Zürich. I am grateful to the UZH CMS group for insightful group discussions, which provided me with a broader perspective of the physics at CMS. I am extremely grateful to UZH for providing me with the opportunity and the financial support to stay in Zürich.

I express my sincere gratitude to my TAC member, Dr Seema Sharma, for providing me with insightful comments on my thesis and for her constant support and encouragement on the previous research projects in IISER Pune. I learned a lot of experimental particle physics and data analysis techniques through the discussions. I am also thankful to Dr. Sourabh Dube and my group-mates Vinay, Shubham, Aditee, Bhumika and Alpana at IISER Pune for all the discussions and guidance they provided me during my studies at IISER Pune. I also express my gratitude to KVPY Scholarship for providing me with financial support throughout my undergraduate studies.

I would like to acknowledge my friends at IISER Pune and at Cäsar-Ritz-Strasse in Zürich for motivating me and their communion. Lastly, I would like to thank my parents for their support and warmth throughout my life.

Abstract

The observed kinematic distributions of the particles produced in proton proton collision events are generally different from what is expected for an ideal detector mainly because of the detector effects like imperfect energy resolution, acceptance of the detector, reconstruction efficiency, to name a few. We need to consider these effects so that we can compare the collider observables distributions at the "truth level" with theoretical predictions and with measurements from the different experiments. These comparisons can enhance the understanding of the Standard Model, tune Monte Carlo event generator parameters and enable precision searches for new physics. Unfolding algorithms are used to obtain these truth distributions from the detector's measured information by correcting these detector effects.

Performing high-dimensional measurements at particle colliders are essential as these are the measurements that keep as much information as possible. They are the key to understanding the correlations across different measurements and how they impact interpretations of differential cross-section, Wilson coefficients in top physics, to name a few. Current unfolding methods depends on the binning chosen for the histogram of the measured observable, which causes problems when unfolding is done on several variables simultaneously. All possible auxiliary features that control the detector response are not taken in account by the traditional unfolding algorithms. Hence, to extract as much information as possible using the high-dimensional measurements, there is a requirement of an unfolding method which do not depend on the binning chosen and performs well for multi-dimensional measures.

We explore different unfolding algorithms to develop and test an unfolding method that can be performed in an unbinned, multi-dimensional fashion preserving as much event information as possible. An unfolding method that performs unfolding using the Energy Mover's distance [1] metric is explored and is compared with OmniFold [2] which is a new deep learning-based unbinned unfolding method. A specific case study is used to show specific gains and advantages of these new unbinned unfolding methods.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The Standard Model (SM) of particle physics is the most successful theory to date in explaining the fundamental constituents of matter and their interaction. The theoretical predictions have excellent agreement with almost all published experimental data from accelerators such as LEP, Tevatron and the LHC, to name a few. Even though the Standard Model has strong predictive power, there are reasons to believe that the SM is not a complete theory and it is just an effective theory of a more fundamental theory. It does not have a proper theoretical structure to include the gravitational force to have a unification between all fundamental interactions. Furthermore, the SM does not have answers to questions like the compositions of dark matter, neutrino oscillations and stability of Higgs boson mass to name a few. The recent muon g-2 experiment results [10] by Fermilab and anamolies like the b-physics anomalies [11] hints us towards the possibility of Beyond Standard Model physics, which is very exciting! Mainly, there are two methods to challenge the Standard Model. First, search for direct signatures of particular Beyond Standard Model physics in specific phase space regions. Second, perform precision measurements of the Standard Model parameters and look for the discrepancies with respect to the SM predictions.

In this chapter, an overview of the theoretical background of the Standard Model is given, along with its limitations. The Standard Model is challenged by the differential cross-section measurements at LHC, and the importance of unfolding is discussed in relation to these measurements. Our focus then shifts to unfolding algorithms and developing a new approach to solving the unfolding problem. At the end of this chapter, we discuss our approach to the unfolding algorithm.

1.1 Standard Model

The theoretical structure of the Standard Model of particle physics was developed mainly in the late 20^{th} century when quantum mechanics alone without special relativity could not explain the fundamental interactions. The Standard Model is based on quantum field theory (QFT), which reconciles quantum mechanics and special relativity. Here the quantum fields are used to describe the elementary particles. The Electromagnetic interactions are governed by Quantum Electro-Dynamics (QED). The Strong interactions are described by Quantum Chromo-Dynamics (QCD). Abdus Salam[12], Steven Weinberg[13] and Sheldon Glashow[14] unified the Electromagnetic and the Weak interactions and incorporated the Higgs Boson into electroweak theory. This model correctly predicted the weak interaction mediating $W^{\pm}$ and Z bosons. The SM also includes a mass generating mechanism for all the fundamental particles, i.e., Brout-Englert-Higgs (BEH) mechanism[15, 16]. The SM was further validated by the Higgs boson discovery in 2012 at the LHC (CERN) by the ATLAS[17] and the CMS[18] detectors.

1.1.1 Theoretical Background

In the SM, the matter particles are fermions (spin-half) and interactions among are mediated by vector bosons (spin-one). Fermions are half-integer spin particles, and they follow the Fermi-Dirac statistics. Leptons and quarks are fermions that make up all matter and have three generations based on mass hierarchy. Each generation of leptons has an electrically charged particle ($e^{\mp}, \mu^{\mp}, \tau^{\mp}$) and an electrically neutral particle called neutrinos (ν_e, ν_μ, ν_τ). Each generation of leptons has an associated quantum number - $L_e = 1$, $L_\mu = 1$ and $L_\tau = 1$. In the SM, neutrinos are massless and only interacts via weak interactions. The quarks carry fractional electric charge ($\frac{2}{3}e$, $\frac{-1}{3}e$) and also carry a color charge (*red, green, blue*) which helps them to interact via strong interactions. The quark generation (u, d),(c, s), (t, b) have quantum numbers called charm (C=1), strangeness (S=1), bottomness (b=1) and topness (t=1) and isospins. Confinement of quarks in hadrons leads to no observed free quarks in nature; only colour-neutral states can be observed. According to the CPT theorem, each fermion has its anti-particle with opposite quantum numbers and the same mass. Bosons, on the other hand, are integer spin particles that follow Bose-Einstein statistics. Gauge bosons mediate the fundamental interactions. In the the SM, Gluons(g) photon(γ), W and Z are vector boson (spin = 1) and the Higgs boson is the scalar boson (spin = 0). The $W^{\pm}$, Z bosons are massive mediates the Weak interaction, while photon and gluons are massless bosons, and mediate the electromagnetic and strong forces, respectively. The weak interaction is a chirality dependent theory where it only interacts with the left-handed fundamental particles. The scalar Higgs boson is a massive boson responsible for masses of the elementary fermions and bosons. A detailed picture of fundamental particles and their classification according to their properties is shown in Figure 1.1. The characteristics of the fundamental forces are shown in table 1.1

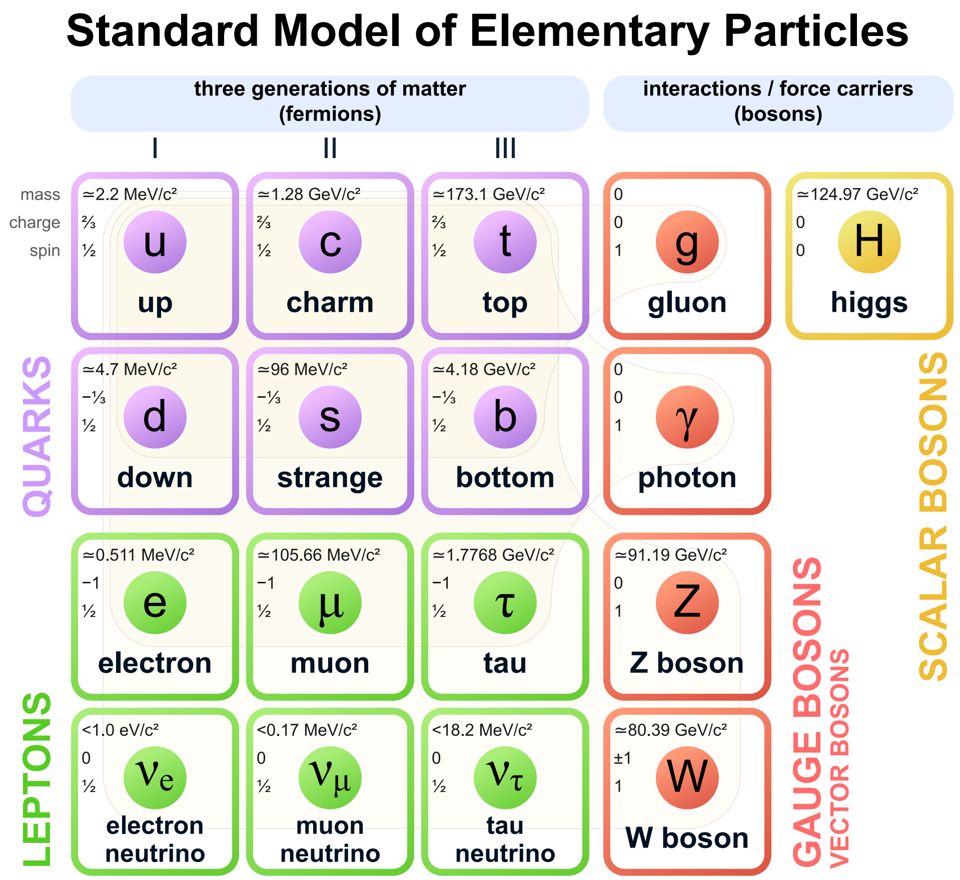

Figure 1.1: The Standard Model picture includes 12 fundamental fermions and 5 fundamental bosons with their name and properties like mass, charge, spin. All the fermions listed are arranged in three generations from left to right. The loops show the boson's coupling to the fermions.

In terms of group theory, the SM obeys an $SU(3)_C \otimes SU(2)_L \otimes U(1)_Y$ gauge symmetry. The $SU(3)_C$ group has eight generators corresponding to eight different types of gluons of the QCD theory. The $SU(2)_L \otimes U(1)_Y$ corresponds to Glashow, Weinberg, and Salam

Interaction	**Theory**	**Mediator**	**Strength**	**Behavior**	**Range**
Weak	EW	$W^{\pm}, Z$	10^{25}	$\frac{1}{r}\exp^{-m_{W,Z}r}$	$10^{-18}m$
Strong	QCD	g	10^{38}	$\sim r$	$10^{-15}m$
EM	QED	γ	10^{36}	$\sim \frac{1}{r}$	∞

Table 1.1: Summary of the fundamental forces that are incorporated in the SM.

(GWS) theory. It has 3 + 1 generators corresponding to three gauge mediators of the weak interactions and one mediator of the electromagnetic interactions. Thus in total, there are 12 force-carrying gauge particles, and the conserved quantities arising from $U(1)_Y$, $SU(2)_L$ and $SU(3)_C$ symmetries are weak hyper-charge, weak isospin and colour charge, respectively. The scalar Higgs field spontaneously breaks the $SU(2) \otimes U(1)$ symmetry group and produces the massive gauge boson - $W^{\pm}, Z$ and massless γ boson via the Higgs mechanism [19]. The short version of the SM Lagrangian incorporating the described fundamental interactions is shown in equation 1.1.

$$\mathcal{L} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} \quad + i\bar{\psi}\not{D}\psi + h.c. \quad + \bar{\psi}_i y_{ij}\psi_j\phi + h.c. \quad + |D_\mu\phi|^2 - V(\phi) \tag{1.1}$$

1.1.2 Limitations

Even though the SM agrees with almost all confirmed experimental data, it still has many theoretical issues. It contains at least 19 arbitrary parameters, which includes three independent gauge couplings, three generalized Cabibbo weak mixing angles, one CP-violating strong-interaction parameter, six quark and three charged-lepton masses, and the CP-violating Kobayashi-Maskawa phase and two independent masses for weak bosons [20]. There are multiple evidences that there is physics Beyond the Standard Model (BSM) that has yet to be discovered. Some of the unanswered questions which limits the SM from being a perfect theory are -

- **Dark Matter and Dark Energy:** Astronomical observations[21] suggest that there is a type of matter which explains the rotation curve of galaxies, but the SM does not have any candidate for such matter (which is invisible by all means so far except Gravity) to explain such observations. This invisible matter is called the “Dark matter,” and the SM cannot explain its abundance.

- **Neutrino masses :** Neutrinos are assumed to be massless in the SM, but the observations of neutrino oscillations[22, 23] indicates that these particles have mass.

o **Matter/anti-matter asymmetry :** From the early stage of the universe, both matter and anti-matter must be created in an equal amount according to the SM, but observations show the contrary. A small amount of this asymmetry can be explained by CP violation in SM, but for the current models of the early universe, this would not be enough. The SM does not provide a mechanism by which matter dominates over anti-matter.

o **Hierarchy Problem :** The particle mass correction comes from the loop interactions with other particles and depends on the energy scale. Taking this into consideration, it is seen that the mass of Higgs must scale quadratically in the energy scale and the correction coming from the top quark loop is -

$$\Delta m_H^2 = -\frac{|\lambda_t|^2}{8\pi^2}[\lambda_{UV}^2 + ..]$$

where λ_t is the Yukawa coupling (~ 1) and Λ_{UV} is the ultraviolet cutoff. This indicates $m_H > 10^{16}$GeV. Experimentally, the mass of Higgs is 125GeV [18, 17]. For this, very fine-tuning in mass is needed, which is not explained by the SM[24, 25].

o **Unification :** Weak and electromagnetic interaction has been unified by the Electroweak theory when considering high enough energy scales, i.e., the observed weak and electromagnetic interactions are actually mixtures of the $U(1)_Y$ and $SU(2)_L$ groups, but this only becomes apparent at higher energies. Currently, the SM does not have a unification of all the three forces, and also, it does not include Gravity.

From these limitations of the SM, there are plenty of good reasons to search for Beyond Standard Model theories. One of the most promising BSM theory is Supersymmetry or SUSY, which is not covered here, and more information can be found in Ref [26].

1.2 Precision experiments at LHC

Direct searches for the beyond standard model physics and precision measurements of Standard Model parameters are two experimental methods to look for physics beyond the SM. The Standard Model can precisely predict the outcome of a large number of experiments, measuring, for example, the probability and cross-sections of various particle interactions. There are 19 free parameters in the Standard Model and measuring them precisely is essential to validate the theory. One of the most well-known examples of precision measurement is the observation of the Higgs boson and the measurements of its decay to other particles by the ATLAS [17], and the CMS [18] detectors at the LHC.

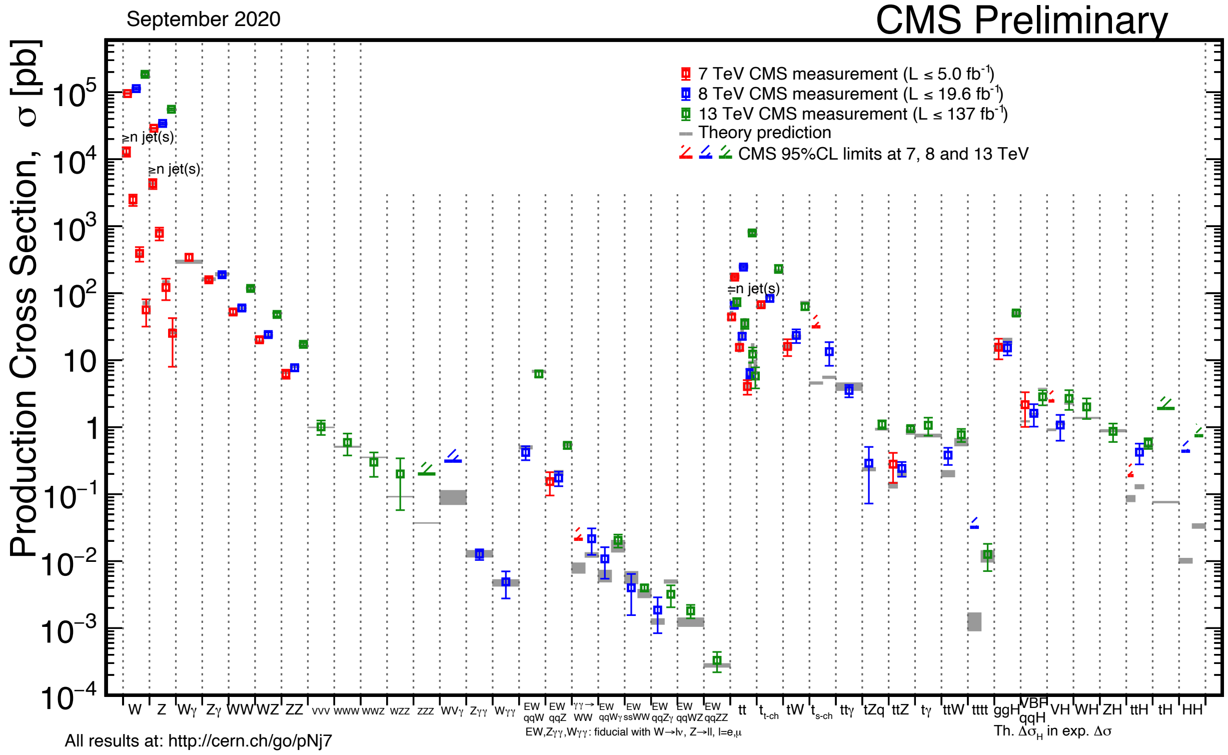

Figure 1.2: Summary of the cross-section measurements of Standard Model processes at CMS [3].

The production cross-sections measurements for various SM processes by the CMS experiment is shown in the figure 1.2 [3]. The CMS precision measurements of the cross-section and the theoretically predicted value are very close.

1.3 Unfolding

Precision experimental measurements like cross-sections require taking into account the detector effects. The detector acceptance and efficiency affects the particle-level distributions (real particle collisions), and the smearing that is introduced by the detector response, and kinematic reconstruction algorithms leads the measured values of an observable to be slightly different from their true values. Unfolding is the procedure that accounts for these detector and physics effects using a non-parametric estimator to get the particle level or the parton

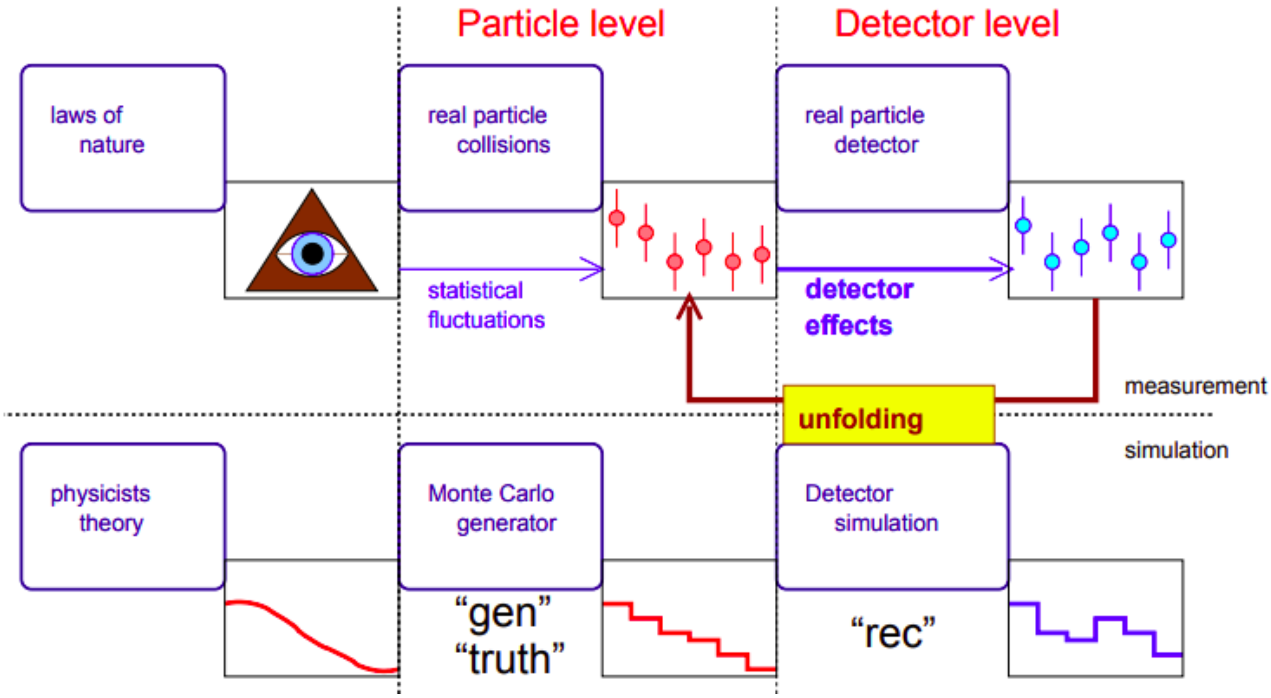

Figure 1.3: Schematic diagram of unfolding process [4]

level space often called "Truth level". Figure 1.3 summarises the unfolding process. Unfolding uses the information from the Monte Carlo generators on which detector simulation is performed, and the "Truth level" distributions are acquired by applying it to the detector level data.

Unfolding is important as it unsmears the detector effects and gives the "truth level" distributions which are useful in comparing the distributions of collider observables. This process ensures the independence of the results from the specific experiments and enables comparisons between different experiments. These comparisons are important as they help in -

- enhancing and understanding of the Standard Model,
- tuning Monte Carlo event generator parameters by fitting the parameters involved in the theory (e.g., QCD) to the data,
- enable comparison between collider observables and theoretical predictions and also with the measurements from different experiments,
- aiding precision searches for new physics, and

- allowing exploratory data analysis

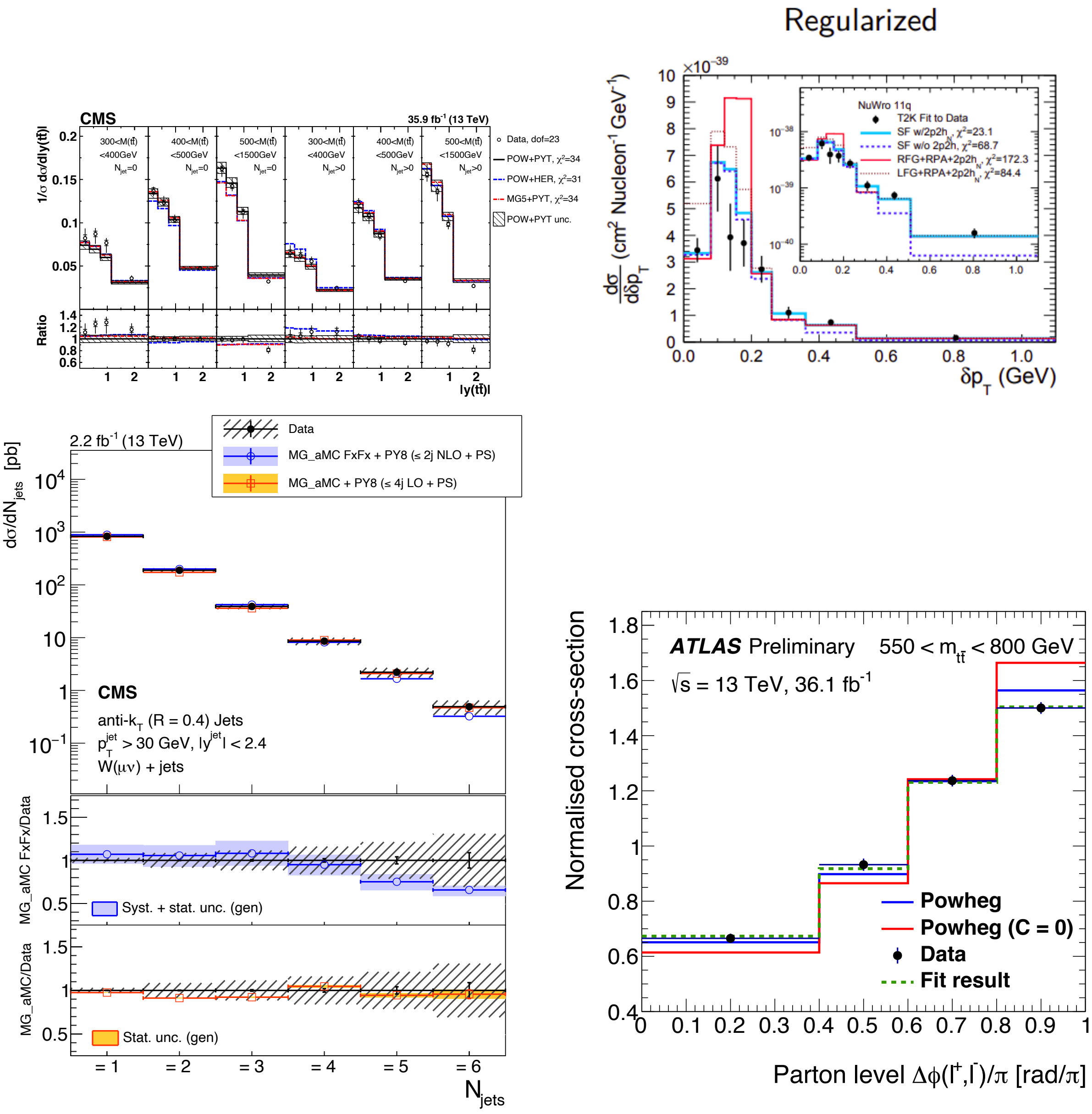

Figure 1.4: Some examples of published results where unfolding was used. The Top left plot shows the $t\bar{t}$ normalized multi-differential cross-section [5]. The top right plot shows the Muon-neutrino scattering at the T2K experiment [6]. The bottom left plot shows the differential cross-section measurement for the associated production of the W boson + jets [7]. The bottom right plot shows the top-quark pair spin correlation measurement in the eμ channel at the ATLAS experiment [8].

Unfolding is most often used in measurement analysis, where the goal is to calculate cross-section for QCD, top and electro-weak processes. A few published results where unfolding was used are shown in Figure 1.4.

Unfolding algorithms are used to mitigate any effects from detector response. As a case study, we will be focusing on the unfolding in the CMS detector; however, we will be using simulations of the CMS detector. The CMS detector is made of trackers and calorimeters, each of them have their own efficiencies and acceptances in resolution of the energy deposited by the particles. In the following subsections, we first introduce the CMS detector geometry and then provide the status of Unfolding in the CMS detectors.

1.3.1 CMS detector

The Large Hadron Collider (LHC) is the world's largest particle accelerator, with a circumference of 27 km. It has four collision points where the detectors are constructed, namely, ATLAS, CMS, LHCb and ALICE. It is designed for proton-proton collisions with the center of mass energy $\sqrt{s} = 14$GeV and a luminosity $\mathscr{L} = 10^{34} cm^{-2} s^{-1}$.

The Compact Muon Solenoid (CMS) is a general-purpose detector studying the SM particle properties and searching for signatures of new physics produced in p-p collisions at the LHC. The CMS detector is cylindrical in geometry with the beam pipe aligning with the axis of the cylinder and has different layers specialised in detecting specific particles as shown in the Figure 1.5.

The superconducting solenoid has an internal diameter of 6m, creating a magnetic field of 3.8 T, which helps bend the particles' trajectories according to their charge. Closest to the beam pipe is the silicon pixel and strip detectors, which help determine charged particle tracks and their momentum. The Electromagnetic Calorimeter (ECal) is made of scintillating lead tungstate crystals, which helps in identifying energy deposit from photons and electrons. The Hadronic Calorimeter (HCal) comprises brass and scintillator where neutral and charged hadrons deposit energy. Muons are measured in the muon chamber, which are gas ionisation detectors embedded in the steel flux-return yoke outside the solenoid. A more detailed description of the CMS detector, along with the coordinate system used and the relevant kinematic variables, can be found in Ref [27].

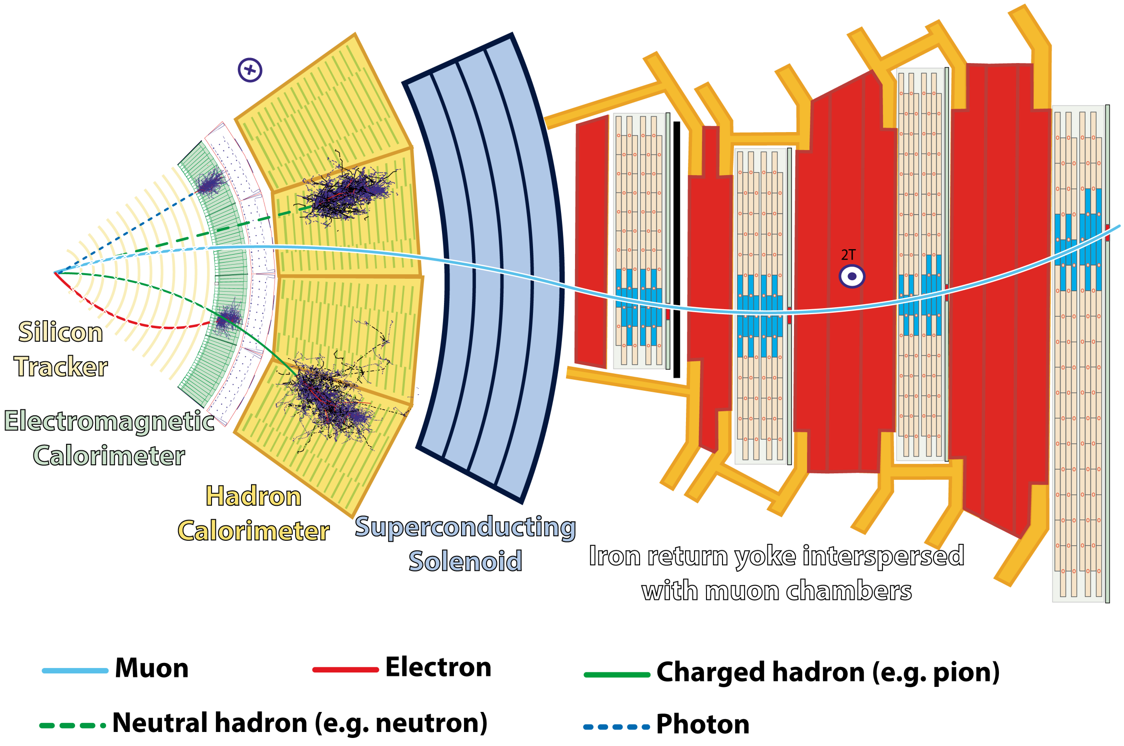
2T
Silicon Tracker
Electromagnetic Calorimeter
Hadron Calorimeter
Superconducting Solenoid
Iron return yoke interspersed with muon chambers
Muon
Electron
Charged hadron (e.g. pion)
Neutral hadron (e.g. neutron)
Photon

Figure 1.5: Transverse slice of CMS detector showing different type of particles depositing their energy from the point of collision to the muon chamber [9].

1.3.2 Current status of Unfolding in CMS

Unfolding in CMS is commonly done using the TUnfold algorithm [28]. TUnfold is interfaced to the ROOT analysis package [29] and is used for correcting the migration and background effects for multi-dimensional distributions. It implements unfolding the data distribution using a least square method with Tikhonov regularisation (discussed in the theory section 2.2.3) where the regularisation parameter is determined using the L-curve method and scans of global correlation coefficients [28]. Apart from the Tikhonov regularisation, Iterative Bayesian Unfolding (IBU) is also widely used for unfolding. Both TUnfold and IBU algorithms can be accessed using the RooUnfold [30] package, which aims to provide a framework for different Unfolding algorithms.

In recent years, there has been a lot of interest in calculating the multi-differential cross-section of various SM processes. These types of measurements requires unfolding to be in a multi-dimensional space where understanding the correlation between the observables is crucial. These measurements are currently performed as a function of various observables.

A few examples of multi-dimensional unfolding used are -

1. "The measurement of $t\bar{t}$ normalized multi-differential cross-sections in pp collisions at $\sqrt{s} = 13$TeV, and simultaneously measuring the strong coupling strength, top quark pole mass and parton distribution function at the CMS experiment" [5]. In this analysis, the unfolding is done in three dimensions as a function of the invariant mass, rapidity of the $t\bar{t}$ system and the multiplicity of the additional jets at the particle level.

2. "Measurement of jet-substructure observables in top quark, W boson and light jet production in proton-proton collisions at $\sqrt{s} = 13$TeV with the ATLAS detector" [31].

1.4 Unfolding Algorithm limitations

One important point we will notice in the theory chapter 2 is that the traditional unfolding algorithms are done for individual binned observables, i.e., they are performed on a binned histogram. This comes from the requirement of a binned response matrix which is initialised using existing histograms of the observables that are to be unfolded. This binning is chosen manually and is determined ahead of time. This binning also prevents us from unfolding a large number of observables simultaneously, and ultimately the measurements of multi-differential cross-section beyond two or three dimensions are not feasible. Working with binned observables also introduces mathematical challenges with sparse binning. Binning also makes it much more challenging for comparing measurements with different experiments and to the theoretical predictions. The RooFit framework can do two or three-dimensional unfolding using the multi-dimensional IBU [32] but cannot do higher dimensional unfolding. Hence, these unfolding algorithms do not generalise well to multi-dimensional unfolding, loses correlation information, and the results are sub-optimal with a potential bias if the detector response depends on auxiliary features.

1.5 Our Approach

Performing high-dimensional measurements at particle colliders are important as these measurements keep as much information as possible. This helps in understanding the correlations across different measurements, which helps in making better comparisons. One approach for performing high-dimensional measurements is to move away from binning and perform unfolding in an unbinned fashion. Omnifold [2] is one such algorithm that performs unfolding at the particle level and is essentially unbinned unfolding. It uses a deep learning method to perform reweighting on the simulated data using the likelihood ratio approach. Detailed information is discussed in the section 2.3.

Optimal transport is a method that is widely used in image processing and on point cloud data. Optimal transport is a very interesting method to consider as it allows morphing from one distribution to another, and it works even for the sparse distribution. This makes optimal transport a good candidate to explore for performing multi-dimensional unbinned unfolding. In this thesis, the traditional unfolding methods are compared with the non-traditional unfolding (OmniFold) algorithm, and we propose an unfolding algorithm based on Optimal transport, which is an unbinned high-dimensional unfolding preserving as much event information as possible.

Chapter 2

Theory

2.1 Mathematics of Unfolding

Unfolding, in mathematics, can generally be classified as an inverse problem and is called deconvolution or unsmearing. In general, the problem can be formulated as to determine the probability density function (pdf) $f(y)$ of a random variable y using a sample of data $y_1, ..., y_n$. In the case where a parametric form for the pdf ($f(y;\theta)$) is known, then standard techniques like maximum likelihood function are used to obtain estimators $\hat{\boldsymbol{\theta}}$.

If the parametric form is not available, we construct a histogram of y with M bins and let the expectation value of the number of entries in bin i of the histogram be represented by μ_i. The goal of unfolding is to construct estimators for the M parameters $\boldsymbol{\mu} = (\mu_1, ..., \mu_M)$. In the case where the parametric form is not available and the measured values of y are subject to further random fluctuations because of measurement errors and statistics. Each observation is characterized by a true value (unknown) y and a measured value x. The pdfs for x and y are related by a convolution as defined in equation 2.1.

$$f_{meas}(x) = \int R(x|y) f_{true}(y) dy \tag{2.1}$$

where $R(x|y)$ is the response function which is known and assumed to only depend on the measuring apparatus [33]. In particle physics, usually the response function is modelled from a simulated sample based on an assumption $f(y)^{model}$

Considering that both x and y pdfs are represented by histograms (usually the case in particle physics), equation 2.1 becomes -

$$\nu_i = \sum_{j=1}^{M} R_{ij}\mu_j, i = 1, ..., N. \tag{2.2}$$

where $\boldsymbol{\mu} = (\mu_1, .., \mu_N)$ is the expectation value for the histogram of y and $\boldsymbol{\nu} = (\nu_1, .., \nu_N)$ is the expectation value for the histogram of x. R is the response matrix, and in terms of conditional probability it is defined as -

$$R_{ij} = P(\text{observed in bin i} \mid \text{true value in bin j}) \tag{2.3}$$

and,

$$\sum_{i=1}^{N} R_{ij} = P(\text{observed anywhere} \mid \text{true value in bin j}) = \epsilon_j \tag{2.4}$$

where the ε_j gives the efficiency, and it depends on the bin j of the true histogram.

Considering the background events in bin i as β_i, the equation 2.2 is modified to equation 2.5 for the data **n** -

$$E[\mathbf{n}] = \boldsymbol{\nu} = R\boldsymbol{\mu} + \boldsymbol{\beta} \tag{2.5}$$

2.2 Traditional Unfolding Algorithms

2.2.1 Correction factors

Deriving correction factors from Monte Carlo simulations is a simple method to construct estimators. For the ith bin, the estimator of the unfolded distribution can be given by -

$$\hat{\mu_i} = (\nu_i - \beta_i)\frac{N_i^{gen}}{N_i^{reco}} \tag{2.6}$$

where N_i^{gen}, N_i^{reco} are the number of generated and reconstructed Monte Carlo events in bin i. This method is only applicable for $M_y = M_x$, i.e., bins on detector level and truth level have a clear correspondence. In the correction factor method, the results have much smaller variances, but they are biased significantly by the underlying Monte Carlo distributions. The estimators are often pulled towards the prediction of the simulation (Monte Carlo) model used for the correction factors by the bias. This method is usually disfavoured due to the simplicity and the fact that it does not account for bin-to-bin migrations. [33, 34].

2.2.2 Matrix Inversion

In the case where the response matrix can be inverted, the equation 2.5 can be written as -

$$\hat{\boldsymbol{\mu}} = R^{-1}(\boldsymbol{\nu} - \boldsymbol{\beta}) \tag{2.7}$$

The Matrix inversion yield unbiased results as there is no assumption on the y_i, but the variance is very high. When the off-diagonal terms in the response matrix are too large, then the estimators have significant variances and strong negative correlations between neighbouring bins. When we apply R^{-1} to get $\boldsymbol{\mu}$, the statistical fluctuations in the data histograms end up with large oscillations between neighbouring bins in the estimators [33]. An example of such oscillations is shown in the unfolding result shown in the figure 2.1.

2.2.3 Matrix Inversion with Regularization

One method to solve the significant variance seen in response matrix inversion is to regularise the solution. Regularisation methods for inversion problems involve a trade-off between the quality of the fit and the size of the regularised solution, which is controlled by the regularisation parameter. The large bin to bin fluctuations of the unfolded result is reduced by the addition of a regularisation term suggested by Tikhonov [35]. In the most general form, the equation of the Tikhonov regularisation is -

$$\boldsymbol{\mu_\lambda} = \arg\min\{||R\boldsymbol{\mu} - \boldsymbol{\nu}||_2^2 + \lambda^2||L(\boldsymbol{\mu} - \boldsymbol{\mu_0})||_2^2\} \tag{2.8}$$

where λ is the regularisation parameter, $||L(\boldsymbol{\mu} - \boldsymbol{\mu_0})||_2$ is the size of the regularisation solution and the fit is measured by $||R\boldsymbol{\mu} - \boldsymbol{\nu}||_2$ of the residual vector. The bias vector $\boldsymbol{\mu_0}$ is often set to zero or equal to the simulation. The matrix L is the regularisation condition, and it is usually called the penalty matrix. The choice of penalty matrix depends on the matrix inversion, and it penalizes the unwanted features (e.g. oscillations) of the unfolded distributions. The most commonly used penalty matrix are Curvature matrix (discretized version of the laplacian/second derivative operator), Identity matrix, Difference matrix (discretized version of the difference operator). The Curvature and the Difference matrix as the penalty matrices long range oscillations while the Identity matrix penalizes the overall norm.

The Tikhonov solution $\boldsymbol{\mu_\lambda}$ is usually given by the rewriting the eq 2.8 as -

$$(R^T R + \lambda^2 L^T L)\boldsymbol{\mu_\lambda} = R^T \boldsymbol{\nu} + \lambda^2 L^T L \boldsymbol{\mu_0} \tag{2.9}$$

The regularisation parameter λ is arbitrary and one way to determine the value of this parameter using the L-curve. It can be noted that if regularisation parameter is very high, then it will not fit the data $\boldsymbol{\nu}$ and the residual $(||R\boldsymbol{\mu_\lambda} - \boldsymbol{\nu}||_2)$ will be too large, whereas if regularisation parameter is too small, the fit will be good but it will be affected by the large oscillation, i.e., $||L(\boldsymbol{\mu_\lambda} - \boldsymbol{\mu_0}||_2$ will be very large. The L-curve is obtained by plotting the residual norm $(||R\boldsymbol{\mu_\lambda} - \boldsymbol{\nu}||_2)$ vs the solution norm $(||L(\boldsymbol{\mu_\lambda} - \boldsymbol{\mu_0}||_2)$. The regularisation parameter is selected corresponding to the point where change of slope is maximum i.e., the kink in the L-curve[36, 33, 34].

The significant variances seen in the response matrix inversion can be solved using the Tikhonov regularisation. We see an example that will show how the difference can be seen when Matrix inversion is applied vs the Tikhonov regularised solution. We first create an exponentially generated distribution called "Truth lepton p_T", and for detector simulation, the exponential distribution is smeared using a normal distribution. The smeared distribution is called "Detector lepton p_T". The response matrix is created by plotting Detector lepton p_T vs Truth lepton p_T which is shown in the top left plot of Figure 2.1. We see from the Matrix inversion unfolding (top right plot 2.1) for this example that the variance is very significant. The unfolded distribution looks nothing like the truth distribution with wild fluctuations and negative values in some bins. When inverting the matrix, the solution amplifies the small fluctuations leading to wildly different results. Performing the Tikhonov regularisation unfolding, first, the regularisation parameter is selected from the L-curve kink (bottom left plot in 2.1, where the curvature matrix is used as the penalty matrix for penalizing the oscillations of the unfolded distribution, which is shown in the bottom right plot in 2.1. Using the regularisation, the wild fluctuations when inverting the matrix got penalized in the unfolded distribution. For example purpose, we specifically chose to showcase an extreme case of matrix inversion going wrong. Matrix inversion does not perform this poorly and it largely depends on the response matrix, and its eigenvalues.

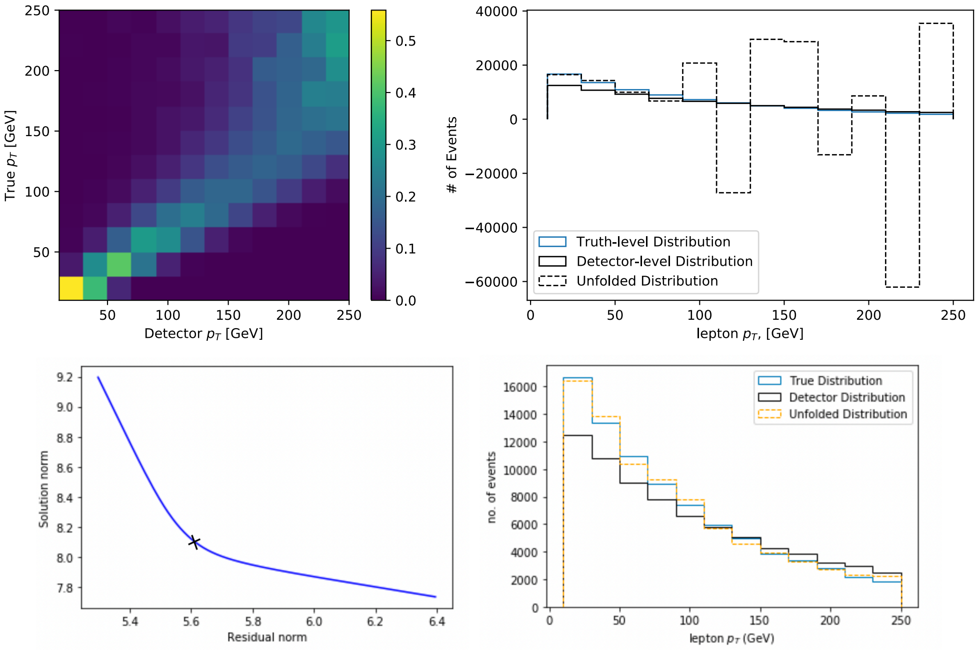

Figure 2.1: A comparison of Matrix inversion and Tikhonov regularisation using an example is shown in the plots. The top left plot shows the response matrix for lepton p_T. The top right plot shows the unfolded distribution using the Matrix inversion. The bottom left plot shows the L-curve between the residual norm and the solution norm, where the cross indicates the regularisation parameter chosen from the L-curve. The bottom right plot shows the unfolded distribution using the Tikhonov regularisation.

2.2.4 Iterative Bayesian Unfolding

D'Agostini [37] proposed an iterative method for unfolding, which makes use of the Bayes theorem. Let y denote unfolded distribution, and x denote data distribution, then the equation for the Iterative Bayesian Unfolding is shown in equation 2.10.

$$y_j = \sum_i \Pr_{n-1}(y_j|x_i)\,\Pr(x_i) = \sum_i \frac{R_{ij} y_j^{n-1}}{\sum_k R_{ik} y_k^{(n-1)}} \times x_i \tag{2.10}$$

where i, j are the bin numbers and n is the number of iterations. In IBU, the number of iterations act as the regularisation parameters and is the arbitrary parameter of the theory. The number of iterations is decided by looking at the simulation test data. As the n becomes large, it brings the large variances, and the estimators approach the oscillating solutions that we see in the Matrix inversion [33]. The convergence rate for IBU towards the unregularized estimate is very slow compared to the other traditional unfolding methods, and the number of iterations is expected to grow with the number of bins squared [38].

2.2.5 TUnfold

TUnfold algorithm estimates $\boldsymbol{\mu}$ using a least square method with Tikhonov regularisation and optional area constraint. As discussed earlier, let ν be the observables, $\boldsymbol{\mu}$ be the truth level distribution and $\mathbf{R}$ be the response matrix. Let $\mathbf{V}_{\boldsymbol{\nu\nu}}$ be the covariance matrix of $\boldsymbol{\nu}$. For the TUnfold algorithm to work, the dimension of $\boldsymbol{\mu}$ is less or equal to the dimension of $\boldsymbol{\nu}$.

The TUnfold algorithm determines the stationary point of the Lagrangian

$$\mathcal{L}(\mu, \lambda) = \mathcal{L}_1 + \mathcal{L}_2 + \mathcal{L}_3 \tag{2.11}$$

$$\begin{aligned}
\mathcal{L}_1 &= (\boldsymbol{\nu} - \mathbf{R}\boldsymbol{\mu})^T \mathbf{V}_{\boldsymbol{\nu\nu}}{}^{-1} (\boldsymbol{\nu} - \mathbf{R}\boldsymbol{\mu}), \\
\mathcal{L}_2 is &= \tau^2 (\boldsymbol{\mu} - f_b \boldsymbol{\mu}_0)^T (\mathbf{L}^T \mathbf{L}) (\boldsymbol{\mu} - f_b \boldsymbol{\mu}_0), \\
\mathcal{L}_3 &= \lambda (Y - e^T \boldsymbol{\mu}), \\
Y &= \sum_i \nu_i, \\
e_j &= \sum_i R_{ij}
\end{aligned} \tag{2.12}$$

The term $\mathcal{L}_1$ is from the least square minimisation. The term $\mathcal{L}_2$ describes the regularisation where the regularisation parameter is τ^2 which helps to dampen the fluctuations in $\boldsymbol{\nu}$. The term $\mathcal{L}_3$ is an optional area constraint, and the λ is the Lagrangian parameter. The stationary point of the $\mathcal{L}(\boldsymbol{\mu}, \lambda)$ is discussed in more detail in literature [28].

2.3 Machine Learning Based Unfolding - OmniFold

OmniFold is an unfolding method where the detector-level quantities are iteratively unfolded, and machine learning is used to handle phase space of any dimensionality without requiring binning. OmniFold is an unfolding method that generalizes the iterative Bayesian equation 2.10 [37] to the unbinned, full phase space using the likelihood ratio approach. To understand it, consider samples X and X', and $p_{w,X}(x)$, $p_{w',X'}(x)$ is the probability density of x estimated from the empirical weights w & w' and the samples X & X' respectively. A classifier is trained to distinguish (w, X) from (x', X') as shown in eq 2.13.

$$L[(w, X), (w', X')](x) = \frac{p_{w,X}(x)}{p_{w',X'}(x)} \tag{2.13}$$

The function $L[(w,X),(w',X')](x)$ is approximated using classifier trained to distinguish (w,X) and (w',X'). Neural network classifiers are used to iteratively reweight the particle and detector level Monte Carlo weights, which results in an unfolding procedure.

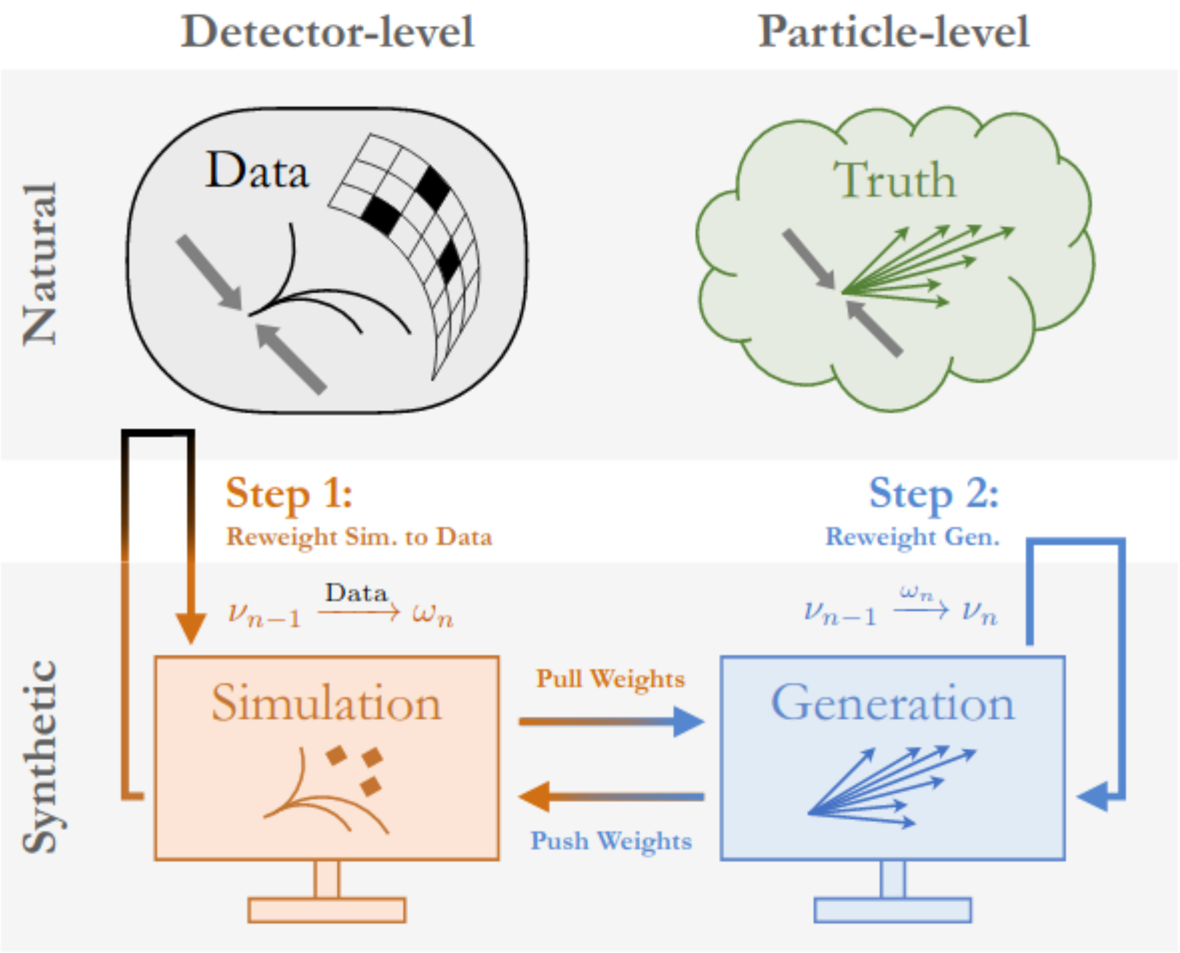

Figure 2.2: A schematic diagram of the OmniFold algorithm [2]. Here the "Natural" level is the physics that we observe in nature, and "Synthetic" is the simulation of the nature.

The OmniFold technique is illustrated in Fig 2.2. The Monte Carlo dataset (Simulation) is used where each particle level event is defined as t and the detector level event is defined as m. First, the OmniFold algorithm starts with the prior weights defined as ν_0, which is for the first iteration assumed to be one. The detector level data at the Natural and the Synthetic level are matched, i.e. the Simulation weights are reweighted to match the Data which is defined as ω_1 and calculated by using the Likelihood ratio eq 2.13, where the function L is approximated using the classifier trained to distinguish (ν_0, Sim) & $(1, Data)$. These weights ω_1 are "pulled back," and the weights are induced on the particle level synthetic data, i.e. Generation. These "pulled weights" are used to compare Generation with previous weights of the Generation and the weights of the initial Generation is reweighted to match the new weighted Generation which results in the weight ν_1. These ν_1 weights are then "pushed forward" to the Simulation so that the new Simulation is induced. This process is then iterated. The following equations show how the method for n^{th} reweighting -

1 $\omega_n(m) = \nu_{n-1}^{push}(m)L[(1, Data), (\nu_{n-1}^{push}, Sim.)](m)$,

2 $\nu_n(t) = \nu_{n-1}(t)L[(\omega_n^{pull}, Gen.), (\nu_{n-1}, Gen.)](t)$.

The first step yields new detector-level weights $\omega_n(m)$, which are pulled back to particle-level weights $\omega_n^{pull} = \omega_n(m)$ using the same synthetic pairs (to an event at detector, m, matched to the same simulated event at gen-level, t). The second step ensures that ν_n is a valid weighting function of the particle-level quantities.

The OmniFold algorithm has the following variations depending on the input -

o UniFold - Input - Single observables. It is also an unbinned version of IBU.

o MultiFold - Input - Multiple observables. For example, jet mass, multiplicity, jet pT, etc as the input.

o OmniFold - Input - Full event, using the full phase space information.

The OmniFold algorithm performed better at unfolding jet substructure observables on simulated sets of events [2], than the traditional Iterative Bayesian Unfolding (IBU), where the implementation used dense networks with three layers of one hundred nodes each and a two-node output layer.

2.4 Optimal Transport

Optimal transport is a tool to transport between two probabilities measures. Optimal transport can be best thought of as moving a large pile of sand to transform it into a target pile with a prescribed shape in such a way that minimizes the total effort of moving the sand. Now, let us consider the two different piles of sand of the same volume as two different probability distributions. We compare them and consider all of the possible ways to morph, transport or reshape the first pile into second and associate a ”global” cost to every such transport, using the ”local” consideration of how much it costs to move a grain of sand from one place to another. The effort or the costs defines a distance between the two probability distributions and also enriches the geometric structure on the space of probability distributions [39].

2.4.1 Introduction to the Problem

There are two formulations of optimal transport formulations - Monge formulation and Kantorovich formulation.

Monge Formulation

Optimal transport givers a framework for comparing measures μ and ν in a Lagrangian framework. Let μ and ν be two probability measures on space X and Y respectively, i.e., $P(X)$ & $P(Y)$. Let $c : X \times Y \to [0, +\inf)$ be the cost function where $c(x, y)$ measures the cost of transporting one unit of mass from $x \in X$ to $y \in Y$. The goal is to minimize the cost c while transporting $\mu \to \nu$. Define T called the transport map where $T : X \to Y$ transports $\mu \in \mathcal{P}(X)$ to $\nu \in \mathcal{P}(Y)$ given the condition

$$\nu(B) = \mu(T^{-1}(B))$$

for all ν-measurable sets B. Figure 2.3 [39]. shows that for any ν-measurable B and $A = x \in X : T(x) \in B$ that $\mu(A) = \nu(B)$. The defination of the Monge's Optimal Transport Problem is given as -

Definition 2.4.1. *Given $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y)$, minimise $M(T) = \int_X c(x, T(x))d\mu(x)$ over μ-measurable maps $T : X \to Y$ subject to $\nu = T_{\#}\mu$*

Monge's formulation is difficult due to the non-linearity in the constraint 2.4.1 mentioned in the definition of T. It is challenging to apply this formulation in the discrete cases where the existence of the maps T such that the condition 2.4.1 is followed is not known.

Kantorovich Formulation

In the Monge formulation, mass was mapped $x \to T(x)$, but in the Kantorovich formulation, this mass is split. Consider a measure $\pi \in \mathcal{P}(X \times Y)$ and $d\pi(x, y)$ as the amount of

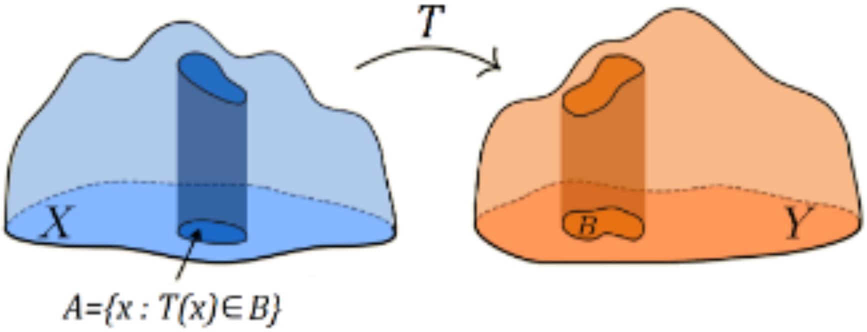

Figure 2.3: Schematic diagram of a Monge's Transport map T which is used to transport probability distribution $A \in X$ to $B \in Y$.

mass transferred from x to y, then mass can be transferred from x to multiple locations. Let $\Pi(\mu, \nu)$ be the set of transport plans between μ and ν, which contains πs such that [39] -

$$\begin{aligned} &\pi(A \times Y) = \mu(A), \\ &\pi(X \times B) = \nu(B) \text{ for all measurable set} A \subseteq X, B \subseteq Y. \end{aligned} \tag{2.14}$$

The Kantorovich's formulation of optimal transport is given as -

Definition 2.4.2. *Given $\mu \in \mathcal{P}(X)$ and $\nu \in \mathcal{P}(Y), minimise K(\pi) = \in_{X \times Y} c(x,y) d\pi(x,y)$ over $\pi \in \Pi(\mu, \nu)$.*

2.4.2 Wasserstein Distance

Given the cost function as $c(x, y) = |x - y|^p$, to find the optimal transport maps T, given $X \sim P_0$ we need to minimize

$$\mathbb{E}\Big[||X - T(X)||^p\Big] = \int ||x - T(x)||^p dP_0(x) \tag{2.15}$$

over all maps T such that $T(X) \sim P_1$ where P_1 & P_0 are two distributions. Fig 2.4[40] shows an example of a transport map.

The Optimal transport problem helps in defining the distance between the two distribution. The metric itself is a type of transport. This distance is called the Wasserstein distance,

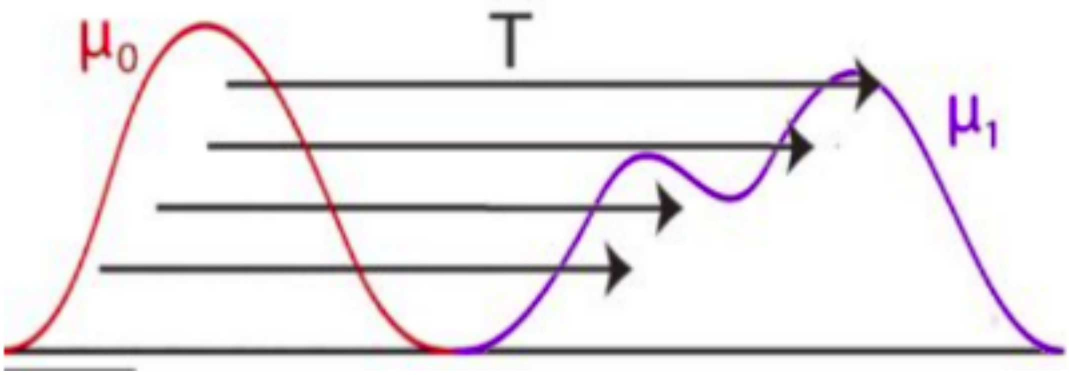

Figure 2.4: Schematic diagram of transport map

which is define as

$$W_p(X, Y) \equiv W_p(P_0, P_1) = \Big(\int ||x - T^*(x)||^p dP_0(x) \Big)^{1/p} \tag{2.16}$$

where T^* is the optimal transport map. W_1 is called the Earth Mover's Distance. A more rigorous definition with proof for metric properties can be found in the Ref [41].

An intuitive way of looking at the Wasserstein distance is to consider the probability distribution as the pile of sand on a given metric space M. Here M can be any metric space like square-euclidean, on which the probability distribution is defined. Then the Earth Mover's distance W_1 is the minimum cost of moving one pile of sand from one point to another and turning the pile to another pile, which takes into consideration the amount of pile moved and the distance it travelled. This can be used to define a distance function between two probabilities on a given metric space.

2.4.3 Energy Mover's Distance

Inspired from the Earth Mover's Distance (W_1), a distance can be defined on collider events called the Energy Mover's Distance (EMD)[1]. This notion of distance can be used as a definition of the similarity between collider events. EMD allows comparison between events with different total energies. The EMD can be thought as the minimum "work" required to rearrange one event ε into the other ε' by moving the energy f_{ij} from particle i in one event to particle j in the other.

$$
\begin{aligned}
EMD(\varepsilon, \varepsilon') = \min_{f_{ij}} \sum_{ij} f_{ij} \frac{\theta_{ij}}{R} + \Big| \sum_i E_i - \sum_j E'_j \Big|, \\
f_{ij} \geq 0, \sum_j f_{ij} \leq E_i, \sum_i f_{ij} \leq E'_j, \sum_{ij} f_{ij} = E_{min}
\end{aligned}
\tag{2.17}
$$

where i and j index particles in events ε and ε', E_i is the particle energy, θ_{ij} is an angular distance between particles, and $E_{min} = \min(\sum_i E_i, \sum_j E'_j)$ is the smaller of the two total energies, R is a parameter that controls the relative importance of the two terms. The EMD has dimensions of energy, where the first term quantifies the difference between the two radiation patterns and the second term accounts for the creation or destruction of energy [1].

EMD is a true metric, and it satisfies the triangle inequality for the condition -

1. θ_{ij} is a metric
2. $R \geq \frac{1}{2}\theta_{max}$, where θ_{max} is the maximum attainable angular distance between particles.

One feature of EMD is that it metricizes the energy flow by treating events differing only by soft particles or collinear splittings identically, which hints at a connection to infrared and collinear (IRC) safety of observables. Fig 2.5 [1] shows an example of the EMD. Here, two top Jets (blue & red) are compared and the Energy Mover's Distance is calculated between them. The particles are shown in the jet reference frame, with the point area proportional to the p_T of the particle.

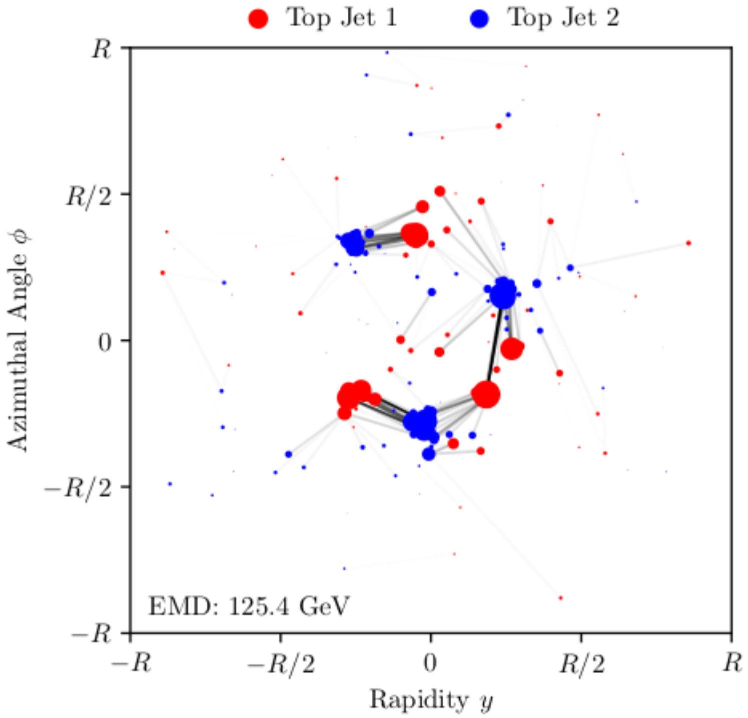

Figure 2.5: Example of EMD - The optimal movement to rearrange one top jet (red) into another (blue) is shown in the plot of the particles where each point is a particle and their area is proportional to their transverse momenta in hte rapidity-azimuth plane. Darker lines indicate more transverse momentum movement. The energy mover's distance in Eq. 2.17 is the total "work" required to perform this rearrangement [1].

Chapter 3

Event Simulation

Monte-Carlo simulations are used to model p-p collisions at the LHC. We will be using the same dataset that the OmniFold paper [2] used for comparison purposes. Herwig 7.1.5 [42] and Pythia 8.243 [43] are used to generate events where Herwig is a substitute for the "nature" (LHC data) and Pythia is used as "synthetic" distributions. This ensures that "nature" and "synthetic" distributions are substantially different. We use Delphes 3.4.2 [44] with fast simulation of the CMS detector is used for to simulate the detector effects, which uses particle flow reconstruction. Jets with radius parameter R = 0.4 are clustered using anti-k_T algorithm [45] implemented in FastJet 3.3.2 [46]. The Z + jet events are produced using the above setup with Z boson $p_T^{min} > 150$ GeV and $\sqrt{s} = 14$TeV. Only the leading jets are considered for the analysis, where the Z boson transverse momentum $p_T^Z > 200$GeV.

The nomenclature that will be used is -

- Gen - Pythia generator level information
- Sim - Pythia reconstructed level information
- Truth - Herwig generator level information
- Data - Herwig reconstructed level information (substitute for LHC data)

Pythia simulated events are used to derive the unfolding correction along with using Herwig reconstructed level information, which will act as the LHC data. The aim is to perform unfolding and compare the unfolded result to the "truth" i.e., Herwig generator level information. The dataset used in this studies is publicly available on zenodo website [47].

Jets are the ideal environment to compare unfolding techniques, since detector effects often account for a significant portion of the experimental measurement uncertainties for many jet substructure observables. To investigate the unfolding performance, the jet substructure observables considered for unfolding are - Jet Mass, Jet Multiplicity, Jet Width, N-subjettiness Ratio (τ_{21}), Grommed Momentum fraction (zg) and Groomed Jet Mass (m_{SD}).

The Jet Mass is the invariant mass of the jet four-vector, Jet multiplicity variable is defined as the number of particles in the Jet. Jet width is the measure of the broadness of the jet along the jet axis, which can be used to understand the nature of the jet. As an example, in the $W + jets$ events, if the p_T^W is high, then the particles are mostly collimated near the jet axis giving low Jet width value [48]. N-subjettiness is a jet shape variable that determines how consistent a jet is with having N subjets, and it defined as the τ_N, i.e. the degree to which the jet can be called a jet of N-subjets [49]. The N-subjettiness ratio $\tau_{21} = \tau_2/\tau_1$, measures how two-pronged the jet is as compared to how 1-prong like it is. Jet grooming has lots of interest as it systematically removes the radiation from the jets, specifically targeting the soft and wide-angle radiation. Removing the unwanted soft radiation allows identification of underlying hard substructure, for example - two-prong W-boson or three-prong top quark decay [50]. The Soft Drop algorithm recursively declusters the jet and removes the soft radiations unless the soft drop condition is satisfied. After the grooming, the invariant mass of the jet four-vector is defined as the Groomed Jet Mass (m_{SD}). The Groomed momentum fraction zg is the energy-sharing of the prongs after grooming. zg can be used to determine the scale of the soft radiation and the angular distance between two particles for identifying the wide-angle radiation [50]. The jet observables for the “synthetic” and “nature” datasets used in the analysis are plotted in Figure 3.1.

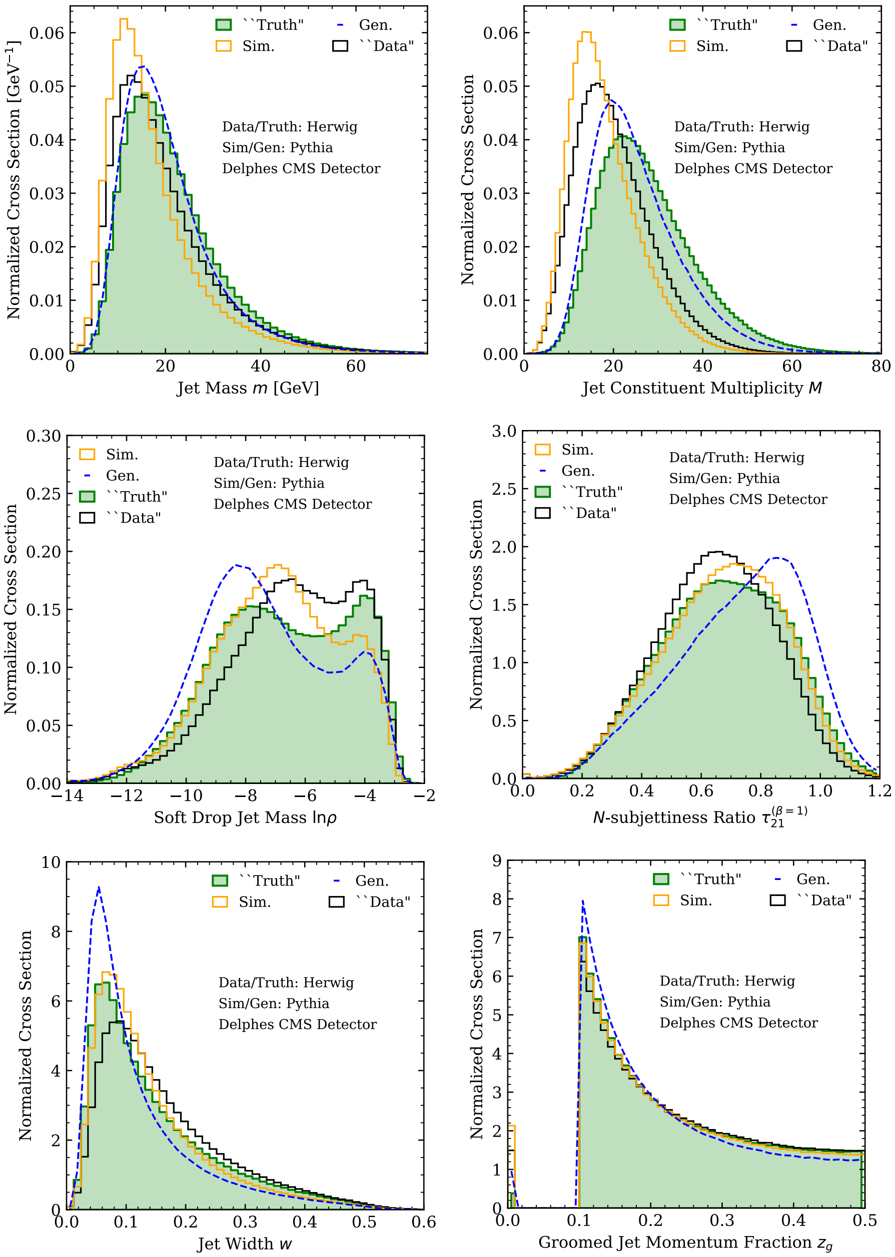

Figure 3.1: Plots shows the jet observable comparison between Gen, Sim, Truth and Data.

Chapter 4

Results

In this chapter, we will start by discussing and comparing the OmniFold and Traditional Unfolding Algorithms and see the performance of OmniFold. Then, we introduce a new approach to unbinned multidimensional unfolding using the Optimal transport and show results for one dimensional and multidimensional cases and compare it with the OmniFold and traditional unfolding results.

4.1 OmniFold and Traditional Unfolding Algorithms

We start our analysis by comparing the ML based OmniFold with the traditional Iterative Bayesian Unfolding (IBU) and Tikhonov regularization for one dimensional observables using the dataset discussed in Event Simulation chapter 3.

The Triangle discriminator measures the similarity, or differences between two distributions p and q, by taking the square of their differences and dividing by the sum of the distributions at each point. Considering each point as an individual bin in the histogram, we can use the triangle discriminator to measure the unfolding performance of various unfolding methods. The equation of the triangle discriminator is given by -

$$\Delta(p,q) = \frac{1}{2}\sum_{i}^{N} \frac{(p(x_i) - q(x_i))^2}{p(x_i) + q(x_i)} \times 10^3 \tag{4.1}$$

where p is the unfolded (binned) histogram, q is the truth (binned) histogram, and x_i is the bin number.

We use Tikhonov regularization and optimize the free regularization parameter λ using the L-curve method 2.9 i.e., solution norm vs residual norm is plotted, which looks like an L shaped curve, and the λ is selected to be the kink of the L-curve, as this represents a good trade-off between the fit to the data and suppression of the unwanted features given by the penalty matrix. We also vary the penalty term and add a bias vector in the regularization. The penalty term, i.e., L matrix in the equation 2.8, used for performing Tikhonov unfolding are - curvature matrix, second-order curvature matrix and the identity matrix. The Curvature matrix is the matrix version of the Laplace operator, and it penalizes the oscillations that arise from the matrix inversion. The bias vector used is the "Gen" vector (Pythia Generator level information). For finding the statistical uncertainties, we use multiple toy samples (smearing the simulation data by reweighting it with a Poisson distribution) for unfolding, and the uncertainties are found from the multiple unfolded results. For IBU, the number of iteration used is $n = 3$. The Response matrix, which is the input to both the Tikhonov and IBU, is calculated using the PYTHIA "Gen" and "Sim" information.

Instead of OmniFold, we use MultiFold, where all the jet observables described above are used as input to derive the detector response. For step 1 and step 2 as mentioned in Theory chapter 2.3, PFN architecture and training parameters with latent space dimension $l = 256$ are used which are implemented in the ENERGYFLOW Python package [51]. Keras and TensorFlow with Adam optimization algorithm are used for training the Neural networks. For the first iteration, the models are randomly initialized and for the subsequent iterations the model from the previous iteration are used. The number of epochs = 100, batch size = 500 are used, and the whole process is iterated three times ($n = 3$). For the ML architecture, dense networks with three layers of one hundred nodes each and a two-node output layer are used.

The unfolding performance of MultiFold, Tikhonov and IBU are shown in Fig 4.1 and the triangle discriminator values are shown in Table 4.1. It is seen that MultiFold (ML-based unfolding method) performs better than IBU and Tikhonov (Traditional unfolding methods) in most of the cases. Tikhonov performs the worst for all the observables except the Jet Multiplicity. It is seen that the plots for Tikhonov unfolding do not compare well with the "Truth" and for the z_g observable, it shows oscillatory behaviour. This oscillatory behaviour mainly comes from the inversion of the response matrix of Groomed Jet momentum fraction z_g as shown in example plots of Tikhonov regularisation 2.1. The Tikhonov regularisation unfolding is done with multiple penalty terms, and only the plots with Curvature matrix as the penalty term and "Gen" vector as the bias term is shown in the Figure 4.1. The Tikhonov regularisation unfolding plots with different penalty and the bias vector mentioned in the table 4.1 are shown in the appendix A.

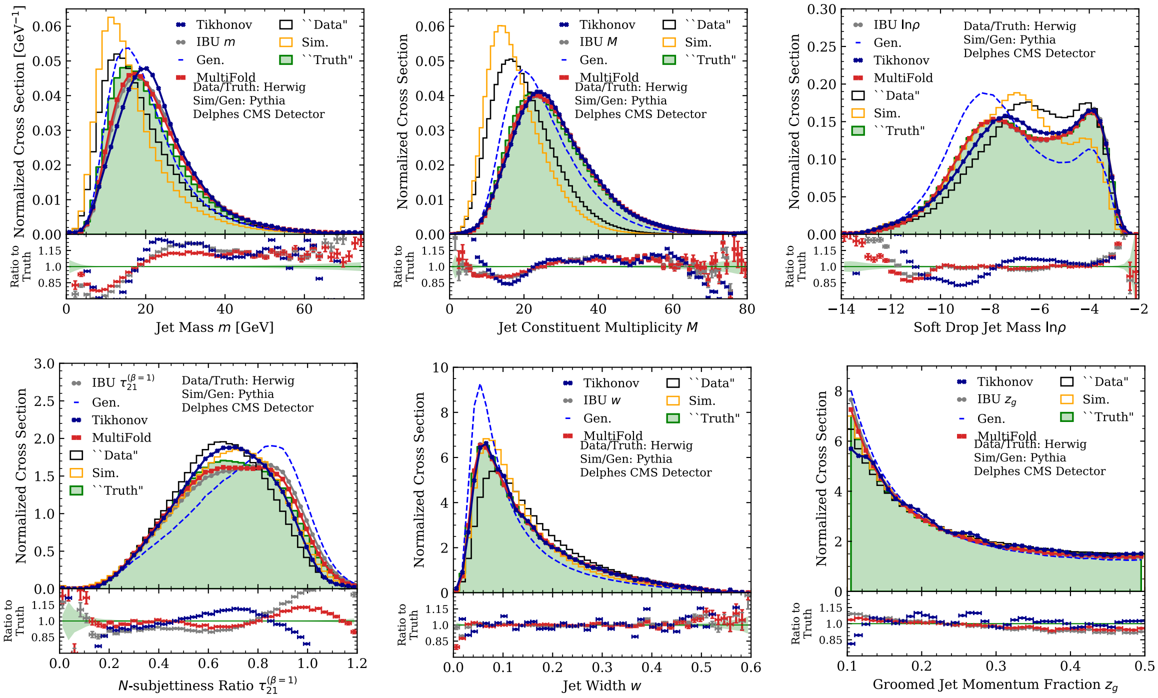

Figure 4.1: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Tikhonov. For the Tikhonov plots shown here, unfolding was done using the curvature matrix as the penalty term and "Gen" vector as the bias. The goal here is to get the unfolded results as close to the green histogram as possible. From the plots, it is seen that MultiFold's unfolding performance is better than the IBU and Tikhonov.

It is seen that MultiFold improves the performance if the full phase space contains auxiliary features that are relevant for the detector response. By construction of the MultiFold method, it is an unbinned and multidimensional unfolding that is performing better than the one dimensional binned traditional unfolding methods, i.e., Tikhonov and IBU.

4.2 Optimal Transport Unfolding (OTU)

As discussed in the theory section on optimal transport 2.4, OT helps in defining a metric on the collider events space and enables comparison between the events. Considering events

Methods	Mass	Multiplicity	width	tau21	zg	SDMass
MultiFold	1.93	1.12	35.87	71.51	66.44	1.1
IBU	4.77	1.11	7.59	143.24	99.42	1.92
Tikh - 1 (no bias)	8.26	1.89	35.66	474.76	165.95	11.95
Tikh - 1 (MC bias)	9.12	1.89	86.54	213.19	222.27	10.91
Tikh - 2 (no bias)	8.45	2.04	173.93	333.87	153.44	7.75
Tikh - 2 (MC bias)	8.97	0.89	193.63	183.74	253.63	10.73
Tikh - 3 (MC bias)	27.06	0.23	8284.56	323.74	245.32	22.9

Table 4.1: Triangle discriminator values for various unfolding methods. Tikh-1 is Tikhonov unfolding with Curvature matrix as the penatly term. Tikh-2 is Tikhonov unfolding with Second-order curvature matrix as the penalty term. Tikh-3 is Tikhonov unfolding with Identity matrix as the penalty term. (MC bias) means that the "Gen" vector was used as the bias term during the unfolding.

as a distribution on the collider event space, applying optimal transport, we compare each event and show similarity between the events. Using this information, we can compare the "Data" and "Sim" events in the collider event space, i.e., we want to find the optimal way to transform "Data" distribution to the "Sim" distribution. From this comparison, we can find an optimal transport flow matrix T, which can be used to transform "Data" to look like "Sim" distribution on the collider event space. There is a known relationship between each event at the detector and generator level for the simulated events. In Monte Carlo simulations, each event is individually simulated and can be tracked from generator level to detector level, and so its migration can be tracked on an individual event basis. Using this relation between "Gen" to "Sim", we invert this relationship and apply it to the "Data" distribution to get the "Unfolded" distribution in the collider event space. Performing unfolding in the event space and not on the binned histogram makes the algorithm by construction an unbinned unfolding method. A schematic diagram is shown in the Fig 4.2.

In the algorithm 1, solving the optimal transport and finding the optimal flow matrix between the "Data" distributions and the "Sim" distribution can be understood as an Association, and applying the inversion of the relationship between "Gen" and "Sim" as the Transport function, which transports the "Data" to the "Truth" distribution. We define $f(G, S)$ to be the transport function that incorporated the known relationship between each event at the detector and generator level for the simulated events, i.e., a function between "Gen" and "Sim". This function takes us from the "Gen" distribution to "Sim" distribution for each event, and it includes the detector effects. Now inverting this function and applying

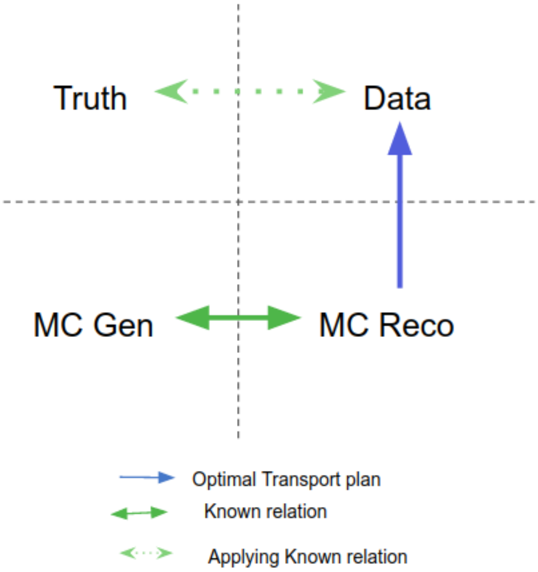

Figure 4.2: Schematic diagram of the application of Optimal transport for unfolding. Here, the Known relation is the relation between the "Gen" and "Sim" that is obtained from the fact that each event is individually simulated and can be tracked from generator to the detector level.

it to the "Data" distribution gives us the "unfolded" distribution.

Algorithm 1: Optimal transport unfolding

```
Initialize the variables $D \leftarrow$ "Data" observable, $S \leftarrow$ "Sim" observable and
  $G \leftarrow$ "Gen" observable";
$M$ = calculate cost matrix between the $D$ and $S$ for the observables;
$T$ = calculate EMD flow matrix from $D$ to $S$ using M and event weights ;
$V = T \times D^{\top}$;
$f(G, S)$ = function that incorporates the relation between $G$ and $S$;
Unfold = $f^{-1}(V, f(G, S))$;
Result: Unfolded observable
```

4.2.1 One Dimensional OTU

Traditionally, unfolding is often done for single observables, which is being measured, for example, the momentum of a jet. Although it is possible to use more information, i.e., more inputs to facilitate unfolding, often only the information from the variable of interest is used. As a starting point, we, therefore, consider a similar case for optimal transport unfolding. Here, we use the value of the variable itself to define the distance between two events. For example, if we measured the momentum of the most energetic jet, the distance between two events would be defined as the square of the difference between the momentum of the most energetic jet in the two events. Using the single observable, we first create a metric space of collider events using a metric (square-euclidean) defined on the observable of interest. Then compare "Data", and "Sim" events using the Earth Mover's distance as implemented in the Optimal Transport python library (POT) [52] and calculate the optimal transportation map T. We then rearrange the "Data" to match the "Sim" events using the map T. From the knowledge of "Gen" and "Sim" relation, we can apply this relation on "Data" to get the unfolded result. For one dimensional unfolding, the input is just the observables that are to be unfolded. We can define the function between the "Gen" and "Sim" to be the difference or ratio, i.e., $f(G, S) = G - S$ or $f(G, S) = G/S$. More complex functions can be defined but in our analysis we mainly use the difference between the "Gen" and "Sim" distributions.

One big obstacle to Optimal transport is the computational cost. Computing p-Wasserstein distance between two events requires $O(n^3)$ operations, where n is the number of particles in each event. We need to compute the pairwise p-Wasserstein distances between the entire collection of N_{evt} requiring $O(N_{evt}^2)$ computations[53]. Due to computational limitations, the number of events was limited to 5000 events.

The unfolding performance of MultiFold, IBU and Optimal transport are shown in Fig 4.4 & 4.3 and the triangle discriminator values are shown in Table 4.2. From the plots and triangle discriminator values, we see that our one-dimensional Optimal transport unfolding performs better than MultiFold and IBU for Jet Mass and Jet Multiplicity observables, but unfolding performance is not great for other variables. In the plots, we see that the histograms are not smooth, and this is due to the limited number of events used for unfolding. In order to smooth out some of the fluctuations, we just increased the binning size shown in the histograms here, which would allow the patterns to be more clearly visible and compared. For the unbinned methods, there is no difficulty with changing the binning. Fig 4.4 uses the "Gen" to "Sim" relation as the $f(G, S) = G/S$ and Fig 4.3 uses $f(G, S) = G - S$. Comparing the plots, it is observed that both difference and the ratio functions produce similar outputs. We will be using the difference between "Gen" and "Sim" as the function $f(G, S)$ in performing the Optimal Transport Unfolding. In the table 4.2, the difference function is used to perform 1D OTU for the triangle discriminator values.

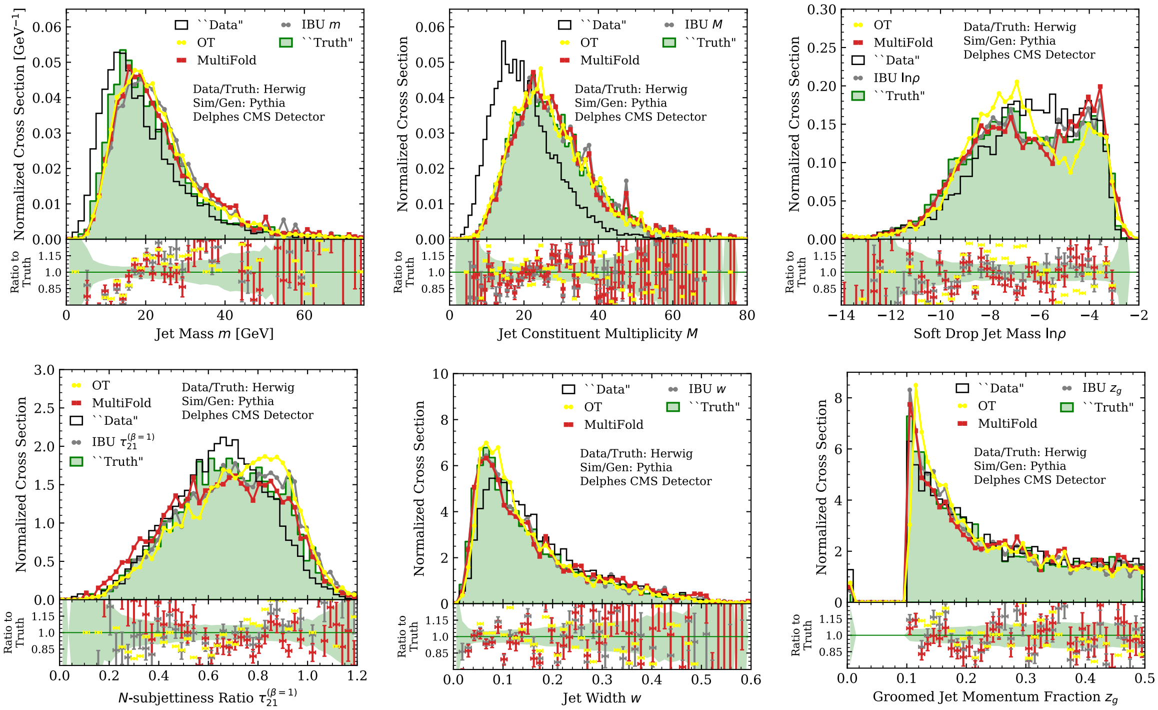

Figure 4.3: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and one-dimensional Optimal Transport. The goal here is to get the unfolded results as close to the green histogram as possible. Here the function used is the difference between "Gen" and "Reco", i.e., $f(G, S) = G - S$. We see that the one-dimensional optimal transport unfolding is doing reasonable when compared to Multifold and IBU.

It is expected that the performance of the unfolding methods changes with the change in the parts of the dataset used. We are only using 5k events, which is a subset of the dataset 3. We create multiple such subsets of 5k events, and we calculate the values of the triangle discriminator for the unfolding methods by varying these subsets and calculate the average and standard deviation values are shown in table 4.2. Also, as MultiFold is a machine learning-based unfolding method where the weight of the PFN architecture are initialized randomly, it is expected that the performance of the MultiFold will vary for each run on the same dataset. These errors are shown in the brackets in the table 4.2.

4.2.2 Different inputs to the OTU algorithm

We already saw the OTU results for one-dimensional inputs, which motivates the use of different inputs to the optimal transport unfolding algorithm 1. In one dimensional OTU,

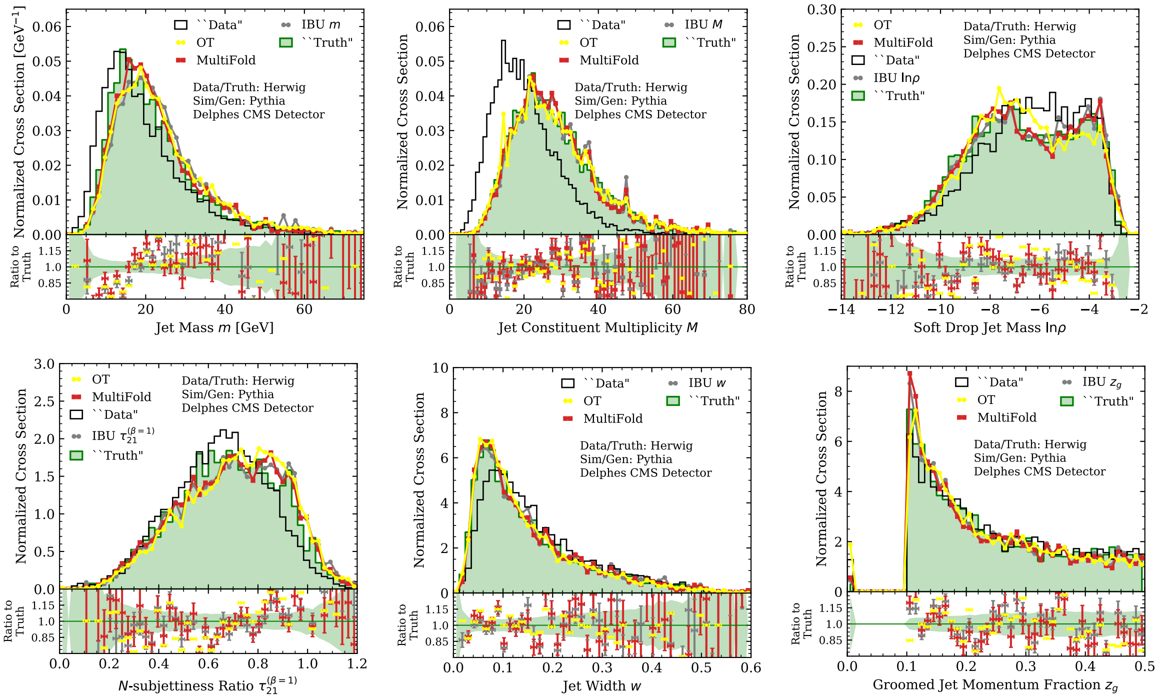

Figure 4.4: The plots show the unfolded distributions for the jet substructure observables. Herwig (“Data”/“Truth”) and Pythia (“Sim”/“Gen”) are used to perform unfolding with IBU, MultiFold and one-dimensional Optimal Transport. The goal here is to get the unfolded results as close to the green histogram as possible. Here the function used is the ratio between “Gen” and “Reco”, i.e., $f(G, S) = G/S$. We see that the one-dimensional optimal transport unfolding is doing reasonable when compared to Multifold and IBU.

Methods	Mass	Multiplicity	width	tau21	zg	SDMass
MultiFold	6.14 +- 1.46(2.43)	10.29 +- 1.96(3.52)	565.299 +- 74.43(278.35)	329.90 +- 147.07(98.94)	613.00 +- 173.08(743.26)	23.56 +- 3.43(10.92)
IBU	9.30 +- 2.05	11.28 +- 1.73	551.29 +- 106.76	397.88 +- 115.30	542.01 +- 52.64	26.47 +- 6.59
OT	6.22 +- 0.82	7.26 +- 0.82	725.87 +- 94.19	514.47 +- 83.12	1108.83 +- 479.51	45.53 +- 5.40

Table 4.2: Triangle discriminator values for MultiFold, IBU and one dimensional OT. Multiple parts of the dataset were used to calculate the average and the variance of each unfolding method. One dimensional OT unfolding outperforms MultiFold and IBU for Jet Mass and Jet Observables and performs reasonably in other cases except for the Groomed Jet Momentum Fraction z_g.

we aimed to understand the unfolding using the optimal transport and perform an unbinned unfolding of an observable of interest, with only one observable as the input. In this section, we want to understand the unfolding performance of the OTU using in a multidimensional case. We observe that the input to the ground metric can be multidimensional, which can help in preserving more correlations than the single observable input as more information was used to perform the unfolding. The inputs to the ground metric calculation that we explored were Jet momentum four-vector and particle momentum four-vectors. Here, we are just changing the "association function" (i.e., changing the ground matric and how different events get associated to each other) and keeping the observable of interest as the "transport function".

For the case where Jet momentum four vectors is provided as the input, we perform the unfolding in the similar manner to 1 by first calculating the metric using Jet 4 vector, then finding the optimal transport matrix T by matching "Sim" and "Data". Then after rearranging, "Data", the Unfolded distribution is estimated using the relation between the "Gen" and "Sim" and applying it on "Data".

For the case where the particle momentum four-vector is provided as the input, there is a tweak to the algorithm 1. Here, we first calculate the Energy Mover's Distance of each event using the implementation in EnergyFlow python library [51]. These Energy Mover's Distance (EMD) will act as a measure to compare two collider events. These values are then used for calculating the cost matrix and which is used to get the optimal transport matrix T using the python optimal transport library (POT). Hence, we applied OT twice, once for calculating EMD between particles of each event and second for performing unfolding. It is important to notice that by structure, this is an unbinned, multidimensional unfolding. The comparison of MultiFold, IBU and Jet four vector OTU are shown in the figure B.1 and the same with Particle four vector OTU are shown in the figure B.2 in the appendix B.1

Methods	Mass	Multiplicity	width	tau21	zg	SDMass
1D	5.76	4.73	523.27	368.79	750.91	38.58
Jet4vector	5.71	3.2	668.35	1178.87	9256.17	74.84
Particle level	6.87	5.55	329.12	889.7	7925.02	36.77

Table 4.3: Triangle discriminator values for one dimensional, jet 4 vector and particle 4 vector OT unfolding.

We compare the OT unfolding with different inputs, i.e., one dimensional, jet four-vector and particle four-vector. Fig 4.5 shows the comparison plots and table 4.3 triangle discriminator comparisons. The number of events used for the plots is 15000. We see that for Jet mass and multiplicity, all the OT unfolding are performing similarly. For soft drop jet mass and Jet width, particle level unfolding is performing better than others. However, in N-subjettiness Ratio, jet and particle level unfoldings have a large concentration of events in

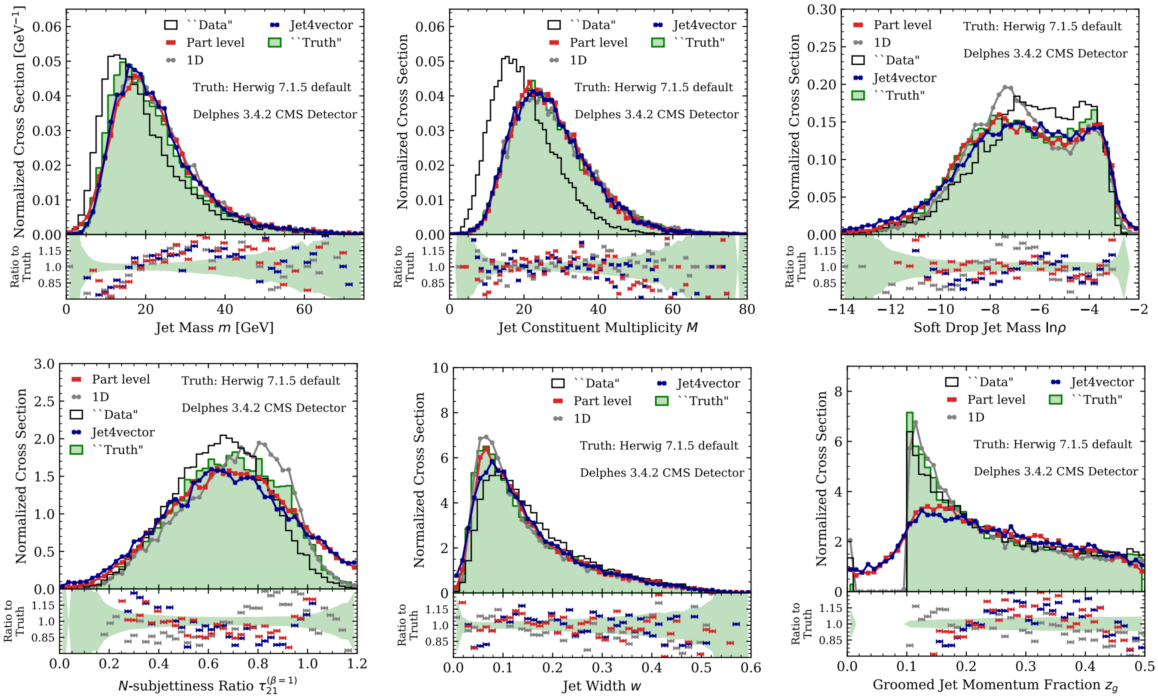

Figure 4.5: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with different inputs to the Optimal Transport problem. The goal here is to get the unfolded results as close to the green histogram as possible.

the higher region. The jet and particle level unfolded distribution shows an interesting behaviour of distributions having a sharp cut like the one in Groomed Jet Momentum Fraction z_g. There is a significant concentration of events in the region below the cut. It is possible that the number of events where gen $z_g < 0.1$ is less than that for sim z_g. It seems like due to this, there is a bias that is pulling all these events towards (0, 0.1) interval. Another possible reason might be that, the events which are close in the case for Jet momentum 4 vector or the Particle momentum 4 vector as the inputs for the "Association function", these events are not necessarily close in the observable of interest for calculating the "Transport function". Hence, when there is a sharp edge in the observable z_g, an event in the "Data" which is close to that edge might get matched to the "Sim" event which is far away from the that edge and has a transport vector which pull it over the edge.

In this subsection, we compared the OTU with different inputs to the algorithm. Using the Jet four-momentum vector as the input to the OTU algorithm helps in preserving the correlations more than the one-dimensional OTU. Apart from some specific cases, we see that the Particle four-momentum vector as the input has good unfolding performance and gives an interesting method of comparing the events. We first compare the events at the particle level and determine a value to the similarity of those events using Energy Mover's Distance, and then we compare the events at the event level. Comparing at the particle level helps in preserving most of the correlations. In the next section, we will extend the idea of comparing the events at the particle level using the EMD values and then use these EMD values as the underlying ground metric to get unfolded particles.

4.2.3 Particle Level OTU

In the previous algorithm 1, we solved the optimal transport problem between the "Data" and "Sim" and obtained the flow matrix at the level of event space and produced the unfolded distribution using it. One of the drawbacks of this algorithm is that it directly get the unfolded distribution of the observables. An approach to get the unfolded particles can give freedom and flexibility in calculating various observables and preserving correlations. This section proposes an optimal transport unfolding method that can perform unfolding in the particle space giving unfolded particles.

Extending the ideas taken from the unfolding done with particle four-momentum as the input to the algorithm 1, we can first solve the optimal transport problem in the particle space for each event in the "Data" and "Sim" and find the Energy Mover's Distance (EMD) for each of the pairs of events in "Data" and "Sim" using the EnergyFlow python package [51]. Using the EMD values as the ground metric on the collider events, we solve the optimal transport problem in the event space and obtain the Transport matrix T, which is used to compare "Data" and "Sim" events. Now, instead of using the observables that we want to unfold as the input, we use a different step to obtain unfolded particles as described in the algorithm 2.

The algorithm 2 make use of the Linearized optimal transport (LOT) [53]. We first take a look at the LOT. As mentioned earlier in 4.2.1, the computational cost of computing the p-Wasserstein distance between two events is $O(n^3)$. Also, computing EMD, i.e. pairwise distance between all events for N_{evt} dataset is $O(N_{evt}^2)$. LOT proposes a framework for computing the distance by reducing the computational cost. It introduces a use of reference vector which helps in speeding up the computation of the optimal transport distances. We use LOT as it allows us to approximate a transport map derived on one event (distirbution)

Algorithm 2: Optimal transport unfolding in particle space

```
Initialize the variables $D \leftarrow$ "Data" observable, $S \leftarrow$ "Sim" observable and
 $G \leftarrow$ "Gen" observable";
M = Cost matrix with EMD values between $D$ and $S$ at the particle level;
T = EMD flow matrix from $D$ to $S$ in the event space using the M cost matrix;
Match $D$ and $S$ events using T;
R = Initialize a reference event;
for number of events do
    $T^{sim}$ = EMD flow matrix from $R$ to $S$ at the particle level;
    $T^{gen}$ = EMD flow matrix from $R$ to $G$ at the particle level;
    $T^{data}$ = EMD flow matrix from $R$ to $D$ at the particle level;
    $s = T^{sim} \times S$;
    $g = T^{gen} \times G$;
    $d = T^{data} \times D$;
    unfolded event = $d + (g - s)$;
    unfolded jets = FastJet clustering on the unfolded particles;
end
Result: Unfolded particles for each event
```

to another event (distribution), which we don't know how to do with the full EMD. We are more interested in the concept of the reference event and use it to enable vector calculation in the particle space.

These reference event contains particles which are used as a reference for the energy flow. First, we solve the optimal transport problem between reference event R and event E (where event E will be "Data", "Sim" and "Gen" event) at the particle level so that we have a Transport matrix T in the particle four momentum vector space. Let the t_{ij} denote the optimal transport plan from reference event R to an event E, which intuitively can be viewed as the transport plan t_{ij} send energy from particle i in the event E to many different particles in the reference event R. Let t_{ij}^{data}, t_{ij}^{gen} and t_{ij}^{sim} be the optimal transport plans from R to "Data" (D), R to "Gen" (G) and R to "Sim" (S) respectively. Using these optimal transport plans a map from event E to a vector r_i is created which is in n-dimensional Euclidean space of $\mathbb{R}^n$, where n is the number of particles in the reference event. This r_i vector is defined as -

$$r_i \equiv \frac{1}{R_i} \sum_j t_{ij} \varepsilon_j \tag{4.2}$$

The energies of the particles in the "Data", "Sim" and "Gen" events are sent to the reference particles using the optimal transport matrix, and by using the eq 4.2 vectors d_i, s_i and g_i are created to have vectors in the same dimensions as the reference event. By applying the knowledge of the relation between "Gen" and "Sim" to "Data" in the vector space created using the reference event, we get Unfolded particles. These unfolded particles are then clustered into jets using the pyjet python package, a python implementation of Fastjet[46]. With these unfolded particles, we can calculate any jet substructure variable needed for the analysis. This unfolding algorithm is an unbinned, multidimensional unfolding that preserves correlations by construction.

In the algorithm 2, it can be noted that the choice of reference event is arbitrary. The reference event is introduced to perform the Linearized Optimal Transport. The choice of reference event has a considerable impact on the quality of the unfolding results. In our analysis, we choose data as the reference event. To understand the algorithm 2, we perform unfolding for a straightforward case. Let the case be defined using the following particle structure -

1. data = [20,1,1,0], [20,-1,-1,0], [5,0,0,0], [5,0,0,0]

2. reco = [20,1,1,0], [20,-1,-1,0], [10,0,0,0]

3. gen = [20,2,2,0], [20,-2,-2,0], [10,0,0,0]

4. truth = [20,2,2,0], [20,-2,-2,0], [5,0,0,0], [5,0,0,0] = Unfolded

In this example, we took a four-particle data event, a three-particle MC event, and the particle at the centre in gen splits into two equal pT in the truth event. For this simple case, figure 4.6 shows the plots between all possible combinations of data, reco, gen, truth, and unfolded events. We see that the data-reco are the same, which is reflected by the EMD value. The EMD values for gen-reco and data-unfolded is the same, which is expected for this simple case, and it is what we want to achieve from the unfolding, i.e. when we perform vector additions to get the unfolded particle distributions, data-unfolded plots should show the similar distributions as that of gen-reco plots. When unfolding is done for the dataset, the EMD values for data-unfolded should be similar to that of gen-reco, and the EMD value for unfolded-gen should be similar to data-reco.

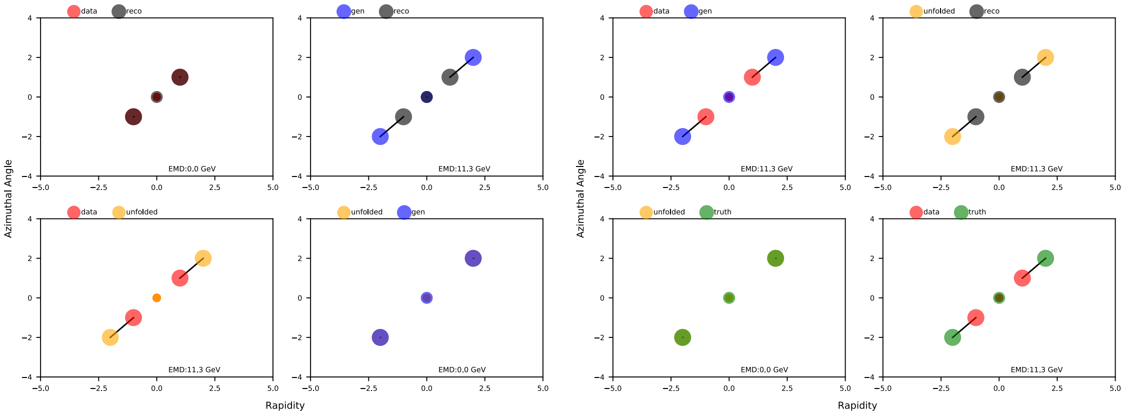

Figure 4.6: Particle level plots of the simple case. Point shows the particles and their area is proportional to their transverse momentum in the rapidity-azimuth plane. Darker lines indicate more transverse momentum movement.

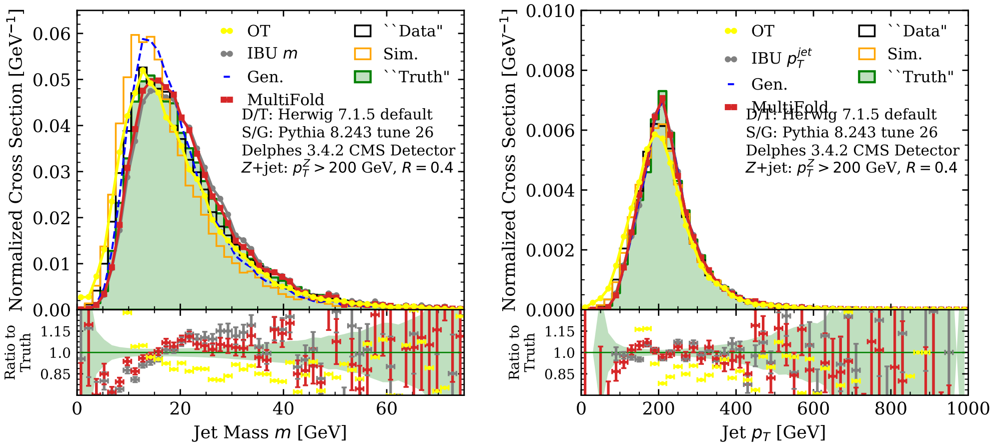

Figure 4.7: The unfolding results for Jet Mass and Jet pT observables using Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are shown in this plots. Particle level OTU is compared with MultiFold and IBU.

Methods	Mass	Jet_pT
MultiFold	1.86	0.06
IBU	3.37	0.07
Particle level OTU	11.99	1.21

Table 4.4: Triangle discriminator values for MultiFold, IBU and Particle level OTU.

We apply the algorithm 2 to the dataset and perform unfolding to get the unfolded particles. We compare this OT unfolding with MultiFold and IBU. Figure 4.7 shows the comparison plots and table 4.2.3, i.e., different inputs to the association function. From the plots and triangle discriminator values, we see that the MultiFold is performing better than our Particle level OTU for the Jet Mass and Jet pT observables. We observe that the current results of Particle level OTU are inconclusive, but the advantage of Particle level OTU can be reflected in its great flexibility that the unfolded particles provide.

It is observed that the Particle level OTU is similar to the data, and the development of the algorithm is in progress to remove this bias. We are also exploring the results with different reference vectors and this is a work in progress.

Chapter 5

Conclusion

This thesis consisted of an unfolding algorithm study in which an innovative unbinned multidimensional unfolding algorithm is proposed. The disadvantages of the traditional unfolding methods, mainly the restrictions coming from the binning and the algorithms' not scaling well for higher dimensions and unfolding in higher dimensions, help conserve the correlations that can be used in comparing results and performing precision analysis with more freedom. Unfolding algorithms for an unbinned and multidimensional measurement which can preserve as much correlation as possible is essential.

We first looked into the Machine Learning based unbinned unfolding algorithm - OmniFold, which can handle phase space of any dimensionality. It was seen in our comparison that OmniFold performs better than the traditional unfolding algorithms. It utilizes the full phase space information to mitigate the problem of auxiliary features that are controlling the detector response. It can be observed that by the structure of the OmniFold algorithm, it unfolds the full-radiation pattern, which can be used to probe new quantities which are challenging to unfold with the traditional unfolding methods [2].

From the motivation of OmniFold's unbinned unfolding, we proposed a novel unfolding algorithm that is unbinned and handles any dimensionality. The method is based on solving the Optimal transport problem and exploiting the transport matrix to perform unfolding. The main concepts used here is the comparison of two events and defining a metric space on the event space. We showed that Optimal transport unfolding gives promising results compared to the traditional unfolding method in specific cases, and it can act as a non deep-learning based alternative to the OmniFold algorithm, although, further studies are required to understand its limitations, and under which conditions it does, and does not perform well. Further improvements to handle some of the cases where it does not perform very well may also be possible.

The Particle level OTU algorithm that unfolds the whole event to produce unfolded particles is an interesting way of unfolding. This presents a paradigm shift towards unfolding where the whole event at the particle level which preserve most of the correlations by the construction of the algorithm. Although, the results presented in the thesis do not give any conclusion on the power of Particle level OTU, the approach of performing unfolding at the event level where we get the unfolded particles provides lots of freedom for precision analysis. These approaches have broad applicability in particle physics; if we came up with a new jet substructure variable, then to compare the results from the unfolded particles, we can cluster these particles however necessary to calculate this new jet substructure very easily, and this can aid to the multi-differential measurements. Calculation of observables which seems daunting using the traditional unfolding methods can be performed effortlessly using the raw data, i.e., unfolded particles.

We have demonstrated the OTU method in a specific case of interest, and we unfolded the full events at the particle level and at the event level and showed the different implementations of optimal transport problem. The OTU algorithms allowed us to unfold more complex dataset statistics. There are a few disadvantages in the current OTU implementation; mainly, the unfolded event has the multiplicity of the reference event. The comparisons of Particle level OTU with different reference events and polishing the OTU algorithm to have better performance is required and currently being implemented but has not to be included in this thesis.

Bibliography

[1] P. T. Komiske, E. M. Metodiev, and J. Thaler, "Metric space of collider events," *Physical review letters*, vol. 123, no. 4, p. 041801, 2019.

[2] A. Andreassen, P. T. Komiske, E. M. Metodiev, B. Nachman, and J. Thaler, "Omnifold: A method to simultaneously unfold all observables," *Physical Review Letters*, vol. 124, no. 18, p. 182001, 2020.

[3] CMS Collaboration, "Summaries of cms cross section measurements." [Online]. Available: https://twiki.cern.ch/twiki/bin/view/CMSPublic/PhysicsResultsCombined

[4] Stefan Schmitt. Unfolding in high energy physics. DESY. [Online]. Available: https://www.desy.de/~sschmitt/talks/UnfoldStatSchool2014.pdf

[5] C. collaboration *et al.*, "Measurement of $t\bar{t}$ normalised multi-differential cross sections in pp collisions at $\sqrt{s} = 13$ tev, and simultaneous determination of the strong coupling strength, top quark pole mass, and parton distribution functions," *arXiv preprint arXiv:1904.05237*, 2019.

[6] K. Abe, J. Amey, C. Andreopoulos, L. Anthony, M. Antonova, S. Aoki, A. Ariga, Y. Ashida, Y. Azuma, S. Ban *et al.*, "Characterization of nuclear effects in muon-neutrino scattering on hydrocarbon with a measurement of final-state kinematics and correlations in charged-current pionless interactions at t2k," *Physical Review D*, vol. 98, no. 3, p. 032003, 2018.

[7] A. M. Sirunyan, A. Tumasyan, W. Adam, F. Ambrogi, E. Asilar, T. Bergauer, J. Brandstetter, E. Brondolin, M. Dragicevic, J. Erö *et al.*, "Measurement of the differential cross sections for the associated production of a w boson and jets in proton-proton collisions at s= 13 tev," *Physical Review D*, vol. 96, no. 7, p. 072005, 2017.

[8] A. Collaboration *et al.*, "Measurements of top-quark pair spin correlations in the eμ channel at√ s= 13 tev using pp collisions in the atlas detector," *European Physical Journal C*, vol. 80, no. 8, p. 754, 2020.

[9] D. Barney, "CMS Detector Slice," Jan 2016, CMS Collection. [Online]. Available: https://cds.cern.ch/record/2120661

[10] B. Abi *et al.*, “Measurement of the Positive Muon Anomalous Magnetic Moment to 0.46 ppm,” *Phys. Rev. Lett.*, vol. 126, no. 14, p. 141801, 2021.

[11] D. Buttazzo, A. Greljo, G. Isidori, and D. Marzocca, “B-physics anomalies: a guide to combined explanations,” *Journal of High Energy Physics*, vol. 2017, no. 11, Nov 2017. [Online]. Available: http://dx.doi.org/10.1007/JHEP11(2017)044

[12] A. Salam and J. C. Ward, “Electromagnetic and weak interactions,” *Phys. Lett.*, vol. 13, pp. 168–171, 1964.

[13] S. Weinberg, “A Model of Leptons,” *Phys. Rev. Lett.*, vol. 19, pp. 1264–1266, 1967.

[14] S. L. Glashow, “Partial Symmetries of Weak Interactions,” *Nucl. Phys.*, vol. 22, pp. 579–588, 1961.

[15] F. Englert and R. Brout, “Broken Symmetry and the Mass of Gauge Vector Mesons,” *Phys. Rev. Lett.*, vol. 13, pp. 321–323, 1964.

[16] P. W. Higgs, “Broken symmetries, massless particles and gauge fields,” *Phys. Lett.*, vol. 12, pp. 132–133, 1964.

[17] G. Aad *et al.*, “Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC,” *Phys. Lett. B*, vol. 716, pp. 1–29, 2012.

[18] S. Chatrchyan *et al.*, “Observation of a New Boson at a Mass of 125 GeV with the CMS Experiment at the LHC,” *Phys. Lett. B*, vol. 716, pp. 30–61, 2012.

[19] M. E. Peskin and D. V. Schroeder, *An Introduction to quantum field theory.* Reading, USA: Addison-Wesley, 1995.

[20] J. R. Ellis, “Limits of the standard model,” in *PSI Zuoz Summer School on Exploring the Limits of the Standard Model*, 11 2002.

[21] E. Corbelli and P. Salucci, “The Extended Rotation Curve and the Dark Matter Halo of M33,” *Mon. Not. Roy. Astron. Soc.*, vol. 311, pp. 441–447, 2000.

[22] Y. Fukuda *et al.*, “Evidence for oscillation of atmospheric neutrinos,” *Phys. Rev. Lett.*, vol. 81, pp. 1562–1567, 1998.

[23] Q. R. Ahmad *et al.*, “Measurement of day and night neutrino energy spectra at SNO and constraints on neutrino mixing parameters,” *Phys. Rev. Lett.*, vol. 89, p. 011302, 2002.

[24] R. Barbieri and G. F. Giudice, “Upper Bounds on Supersymmetric Particle Masses,” *Nucl. Phys. B*, vol. 306, pp. 63–76, 1988.

[25] S. Dimopoulos and G. F. Giudice, "Naturalness constraints in supersymmetric theories with nonuniversal soft terms," *Phys. Lett. B*, vol. 357, pp. 573–578, 1995.

[26] S. P. MARTIN, "A supersymmetry primer," *Advanced Series on Directions in High Energy Physics*, p. 1–98, Jul 1998. [Online]. Available: http://dx.doi.org/10.1142/9789812839657_0001

[27] S. Chatrchyan *et al.*, "The CMS Experiment at the CERN LHC," *JINST*, vol. 3, p. S08004, 2008.

[28] S. Schmitt, "TUnfold: an algorithm for correcting migration effects in high energy physics," *JINST*, vol. 7, p. T10003, 2012.

[29] R. Brun and F. Rademakers, "ROOT: An object oriented data analysis framework," *Nucl. Instrum. Meth. A*, vol. 389, pp. 81–86, 1997.

[30] T. Adye, "Unfolding algorithms and tests using roounfold," 2011.

[31] M. Aaboud, G. Aad, B. Abbott, D. C. Abbott, O. Abdinov, A. Abed Abud, D. K. Abhayasinghe, S. H. Abidi, O. S. AbouZeid, and et al., "Measurement of jet-substructure observables in top quark, w boson and light jet production in proton-proton collisions at

$$\sqrt{s}$$

= 13 tev with the atlas detector," *Journal of High Energy Physics*, vol. 2019, no. 8, Aug 2019. [Online]. Available: http://dx.doi.org/10.1007/JHEP08(2019)033

[32] G. D'Agostini, "A multidimensional unfolding method based on bayes' theorem," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 362, no. 2-3, pp. 487–498, 1995.

[33] G. Cowan, "A survey of unfolding methods for particle physics," in *Prepared for Conference on Advanced Statistical Techniques in Particle Physics, Durham, England*, 2002, pp. 18–22.

[34] S. Schmitt, "Data unfolding methods in high energy physics," *EPJ Web of Conferences*, vol. 137, p. 11008, 2017. [Online]. Available: http://dx.doi.org/10.1051/epjconf/201713711008

[35] A. N. Tikhonov, "On the solution of ill-posed problems and the method of regularization," in *Doklady Akademii Nauk*, vol. 151, no. 3. Russian Academy of Sciences, 1963, pp. 501–504.

[36] P. C. Hansen, "The l-curve and its use in the numerical treatment of inverse problems," 1999.

[37] G. D'Agostini, "A multidimensional unfolding method based on bayes' theorem," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, vol. 362, no. 2-3, pp. 487–498, 1995.

[38] H. N. Multhei and B. Schorr, "On an Iterative Method for the Unfolding of Spectra," *Nucl. Instrum. Meth. A*, vol. 257, p. 371, 1987.

[39] G. Peyre and M. Cuturi, "Computational optimal transport," *Foundations and Trends in Machine Learning*, vol. 11, no. 5-6, pp. 355–607, 2019.

[40] S. Kolouri, S. R. Park, M. Thorpe, D. Slepcev, and G. K. Rohde, "Optimal mass transport: Signal processing and machine-learning applications," *IEEE signal processing magazine*, vol. 34, no. 4, pp. 43–59, 2017.

[41] G. Peyré and M. Cuturi, "Computational optimal transport," 2020.

[42] J. Bellm, M. H. Seymour, P. Kirchgaeßer, F. Loshaj, C. Reuschle, A. Papaefstathiou, R. Podskubka, G. Nail, S. Webster, S. Gieseke *et al.*, "Herwig 7.1 release note: arxiv," Tech. Rep., 2017.

[43] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, "An introduction to pythia 8.2," *Computer physics communications*, vol. 191, pp. 159–177, 2015.

[44] J. De Favereau, C. Delaere, P. Demin, A. Giammanco, V. Lemaitre, A. Mertens, M. Selvaggi, D. . Collaboration *et al.*, "Delphes 3: a modular framework for fast simulation of a generic collider experiment," *Journal of High Energy Physics*, vol. 2014, no. 2, p. 57, 2014.

[45] M. Cacciari, G. P. Salam, and G. Soyez, "The anti-kt jet clustering algorithm," *Journal of High Energy Physics*, vol. 2008, no. 04, p. 063, 2008.

[46] ——, "FastJet User Manual," *Eur. Phys. J. C*, vol. 72, p. 1896, 2012.

[47] A. Andreassen, P. Komiske, E. Metodiev, B. Nachman, and J. Thaler, "Pythia/Herwig + Delphes Jet Datasets for OmniFold Unfolding," *Zenodo*, 2019.

[48] A. J. Larkoski, I. Moult, and B. Nachman, "Jet substructure at the large hadron collider: A review of recent advances in theory and machine learning," *Physics Reports*, vol. 841, p. 1–63, Jan 2020. [Online]. Available: http://dx.doi.org/10.1016/j.physrep.2019.11.001

[49] J. Thaler and K. Van Tilburg, "Identifying boosted objects with n-subjettiness," *Journal of High Energy Physics*, vol. 2011, no. 3, Mar 2011. [Online]. Available: http://dx.doi.org/10.1007/JHEP03(2011)015

[50] R. Kogler, B. Nachman, A. Schmidt, L. Asquith, E. Winkels, M. Campanelli, C. Delitzsch, P. Harris, A. Hinzmann, D. Kar, and et al., "Jet substructure at the large hadron collider," *Reviews of Modern Physics*, vol. 91, no. 4, Dec 2019. [Online]. Available: http://dx.doi.org/10.1103/RevModPhys.91.045003

[51] P. T. Komiske, E. M. Metodiev, and J. Thaler, "The Hidden Geometry of Particle Collisions," *JHEP*, vol. 07, p. 006, 2020.

[52] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, L. Gautheron, N. T. Gayraud, H. Janati, A. Rakotomamonjy, I. Redko, A. Rolet, A. Schutz, V. Seguy, D. J. Sutherland, R. Tavenard, A. Tong, and T. Vayer, "Pot: Python optimal transport," *Journal of Machine Learning Research*, vol. 22, no. 78, pp. 1–8, 2021. [Online]. Available: http://jmlr.org/papers/v22/20-451.html

[53] T. Cai, J. Cheng, N. Craig, and K. Craig, "Linearized optimal transport for collider events," *Physical Review D*, vol. 102, no. 11, Dec 2020. [Online]. Available: http://dx.doi.org/10.1103/PhysRevD.102.116019

Appendix A

Tikhonov Regularisation Plots

In section 4.1, we performed the Tikhonov unfolding using different penalty matrix and using "Gen" vector as the bias vector. In this appendix, we show the plots of the Tikhonov unfolding with the different penalty matrix, namely, Curvature matrix, Second order differential matrix and Identity matrix with "Gen" as bias vector. Following are the list of figures showing the comparison of Tikhonov unfolding with Multifold and IBU -

1. In figure A.1 Tikhonov unfolding is done with Curvature matrix as the penalty matrix no bias vector is used.

2. In figure A.2 Tikhonov unfolding is done with Identity matrix as the penalty matrix "Gen" vector is used as the bias vector.

3. In figure A.3 Tikhonov unfolding is done with Identity matrix as the penalty matrix no bias vector is used.

4. In figure A.4 Tikhonov unfolding is done with discretized second-order differential matrix as the penalty matrix "Gen" vector is used as the bias vector.

5. In figure A.5 Tikhonov unfolding is done with discretized second-order differential matrix as the penalty matrix no bias vector is used.

From the plots, it is observed that the Tikhonov unfolding with Identity matrix as the penalty matrix does not perform well as it is not penalizing the off-diagonal terms, which is causing large oscillations. Both Curvature and Second-order differential matrix penalize the oscillations, but the Unfolding with Curvature matrix performs better than the Second-order matrix from the triangle discriminator values in table 4.1.

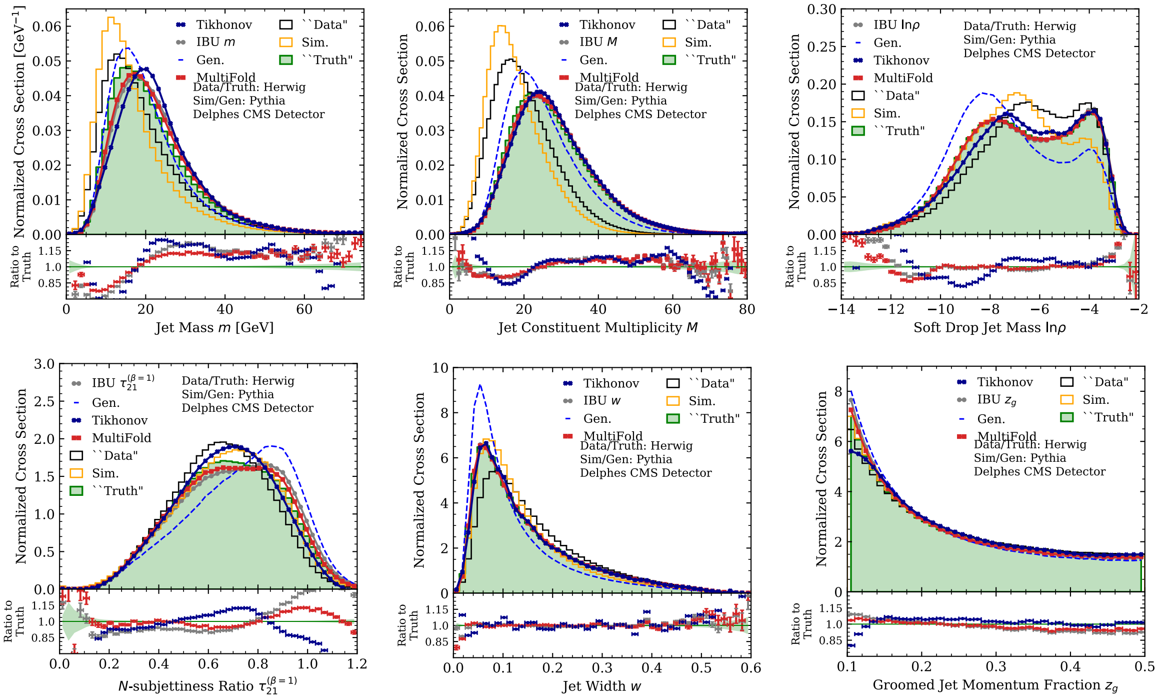

Figure A.1: The plots show the unfolded distributions for the jet substructure observables. Herwig (“Data”/“Truth”) and Pythia (“Sim”/“Gen”) are used to perform unfolding with IBU, MultiFold and Tikhonov. Unfolding using Tikhonov is done using the curvature matrix as the penalty term with no bias term added. The goal of the unfolding algorithms is to get the unfolded results as close to the green “Truth” histogram as possible.

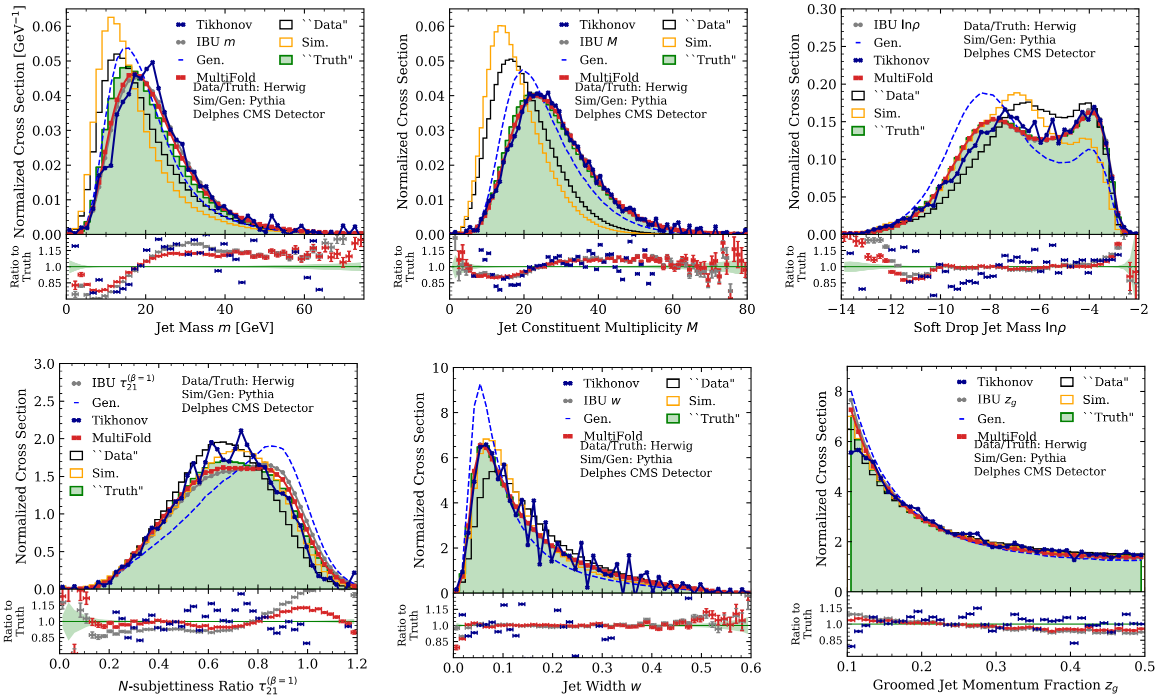

Figure A.2: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Tikhonov. Unfolding using Tikhonov is done using the Identity matrix as the penalty term with "Gen" vector as the bias term added. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.

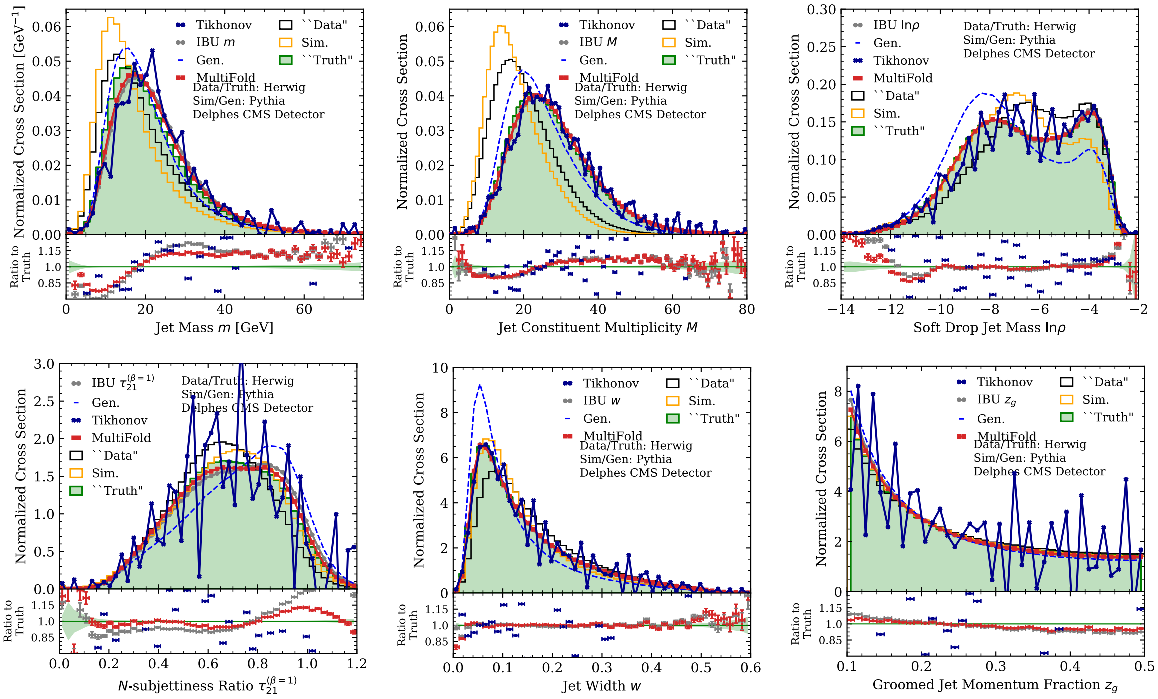

Figure A.3: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Tikhonov. Unfolding using Tikhonov is done using the Identity matrix as the penalty term with no bias term added. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.

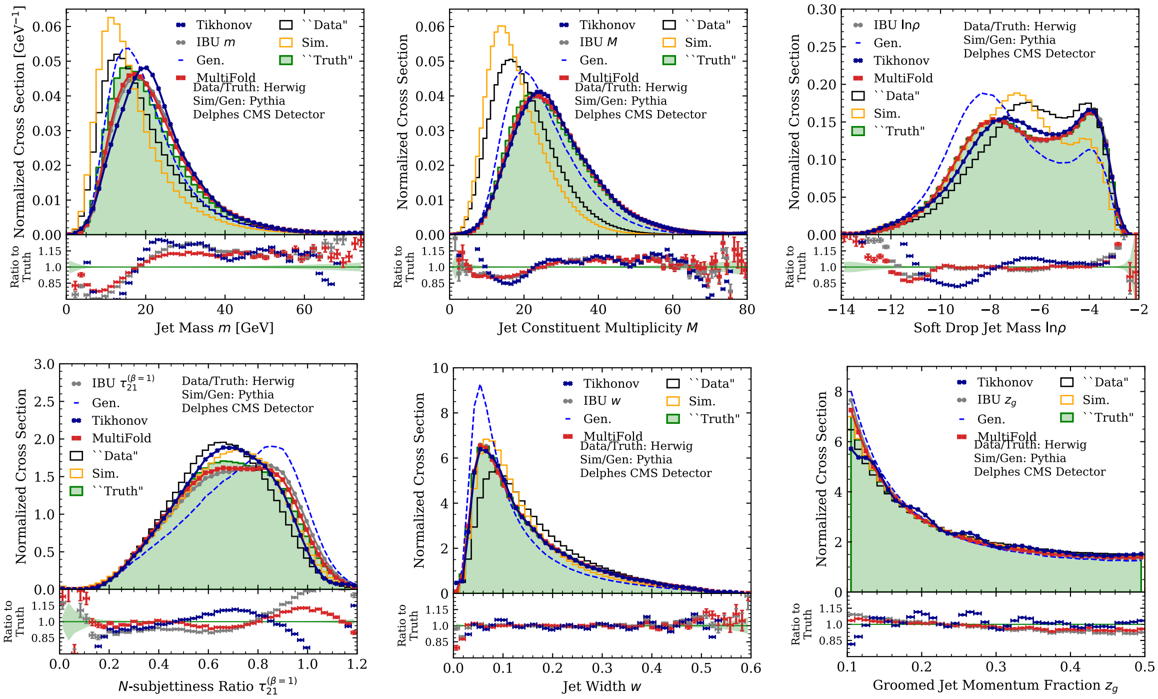

Figure A.4: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Tikhonov. Unfolding using Tikhonov is done using the Second order differential matrix as the penalty term with "Gen" vector as the bias term added. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.

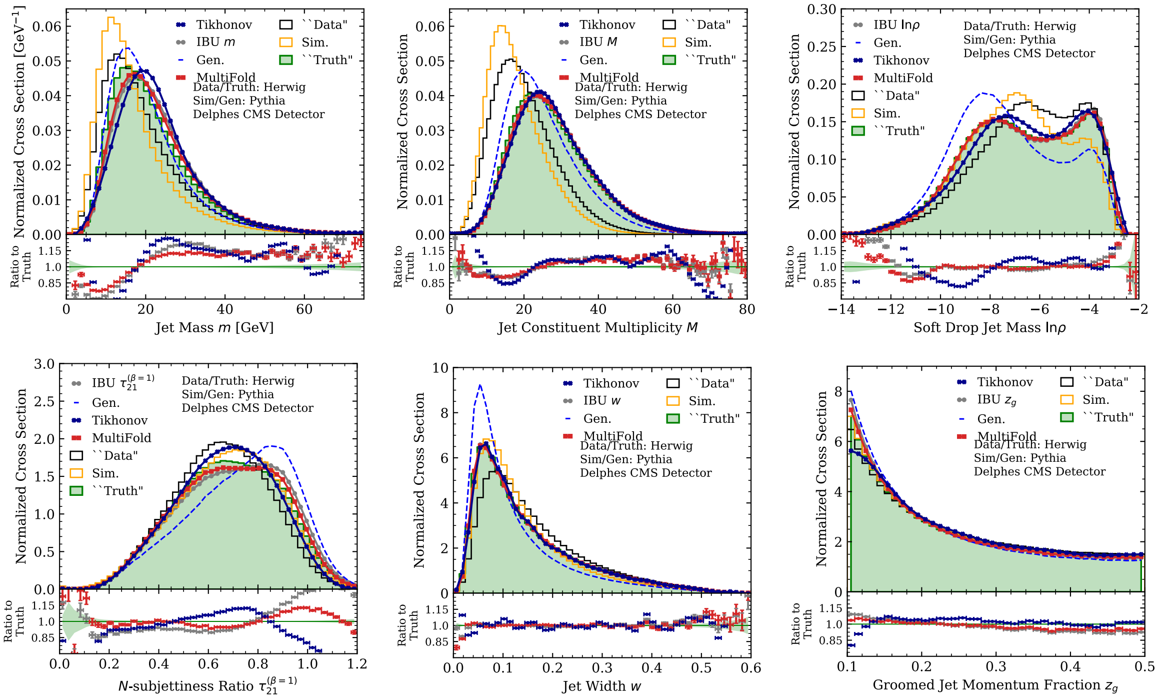

Figure A.5: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Tikhonov. Unfolding using Tikhonov is done using the Second order differential matrix as the penalty term with no bias term added. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.

Appendix B

Optimal transport Unfolding

B.1 Different Input OTU plots

In section 4.2, we showed the comparisons of Optimal Transport Unfolding results for the one-dimensional collider observable input with MultiFold and IBU. Comparison of one-dimensional, jet momentum four-vector and particle momentum four-vector as the input for calculating the ground metric in the OTU algorithm 1 were shown. In this appendix, we show the comparison of Optimal Transport Unfolding results for the jet momentum four-vector B.1 and particle four-vector B.2 with the MultiFold and IBU for the jet substructure observables.

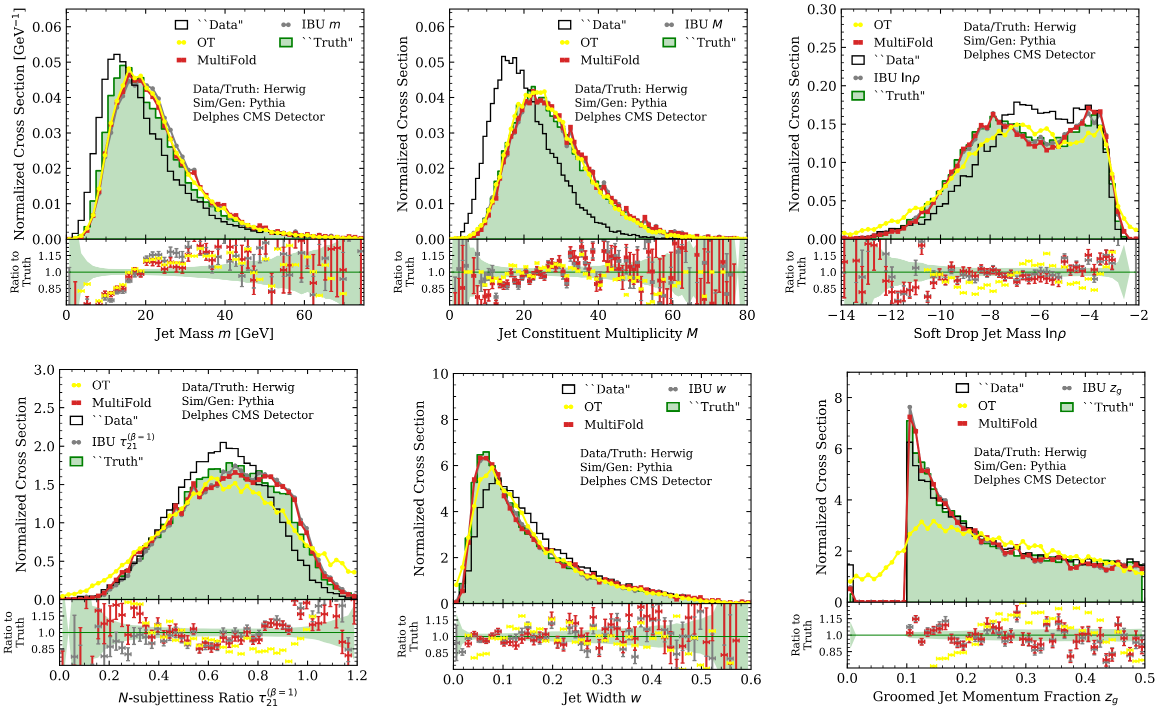

Figure B.1: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Optimal transport unfolding. Unfolding using OTU is done using the jet momentum four vector as the input for calculating the cost matrix. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.

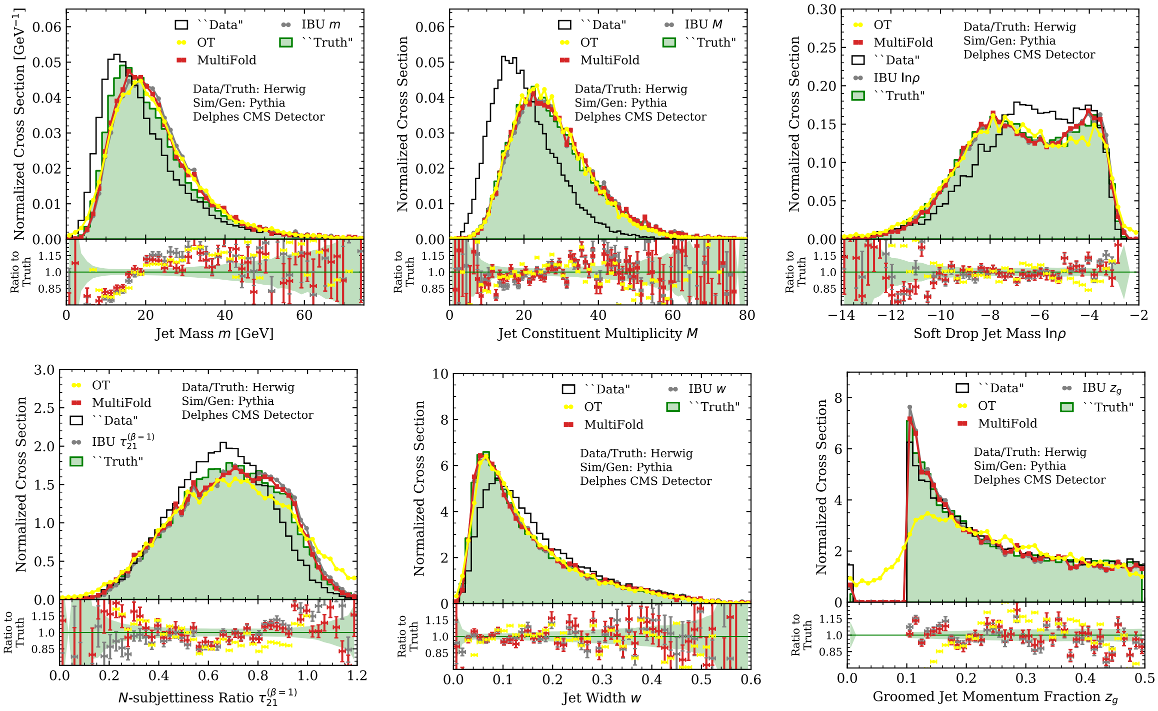

Figure B.2: The plots show the unfolded distributions for the jet substructure observables. Herwig ("Data"/"Truth") and Pythia ("Sim"/"Gen") are used to perform unfolding with IBU, MultiFold and Optimal transport unfolding. Unfolding using OTU is done using the particle momentum four vector as the input for calculating the cost matrix. The goal of the unfolding algorithms is to get the unfolded results as close to the green "Truth" histogram as possible.