UNIVERSITAT POLITÈCNICA DE VALÈNCIA

School of Informatics

Use of Deep Learning generative models for Monte Carlo event simulation in the context of LHC experiments

End of Degree Project

Bachelor's Degree in Informatics Engineering

AUTHOR: Balanzá García, Raúl

Tutor: Gómez Adrian, Jon Ander

External cotutor: SALT CAIROLS, JOSE FRANCISCO

Experimental director: RUIZ DE AUSTRI BAZAN, ROBERTO

ACADEMIC YEAR: 2021/2022

Agradecimientos

Para mí, este trabajo significa mucho más que el final de un grado universitario. Personalmente, marca el cumplimiento de uno de mis objetivos desde hace varios años: conseguir obtener una titulación en el ámbito de estudio de la informática, que representa una de mis mayores vocaciones en la vida.

Para ello, ha sido necesario superar diferentes etapas de estudio en los últimos años, que han requerido un gran trabajo y esfuerzo, durante las cuales he estado motivado por la convicción de ser capaz de conseguir este logro.

Sin embargo, no tengo dudas de que si tengo la posibilidad de estar escribiendo estas palabras es gracias a la ayuda y apoyo de diferentes personas, a quienes me gustaría mostrar mi agradecimiento.

En primer lugar, agradecer a mi profesor y tutor de este trabajo en la UPV Jon Ander Gómez Adrián su apoyo y dedicación, que me ha aportado gran cantidad de conocimiento. Estoy seguro de que, sin su mentorización, este trabajo no habría sido posible.

En segundo lugar, a mis dos tutores externos del IFIC (Centro Mixto CSIC-Universitat de València): Roberto Ruiz de Austri Bazan y José Salt Cairols, que han resuelto todas mis dudas teóricas en relación con la física y han aportado las muestras de datos que han permitido realizar la experimentación que se expone en el trabajo.

En tercer lugar, a mis padres, por apoyarme incondicionalmente durante todos los años de esfuerzo que han conducido hasta este trabajo y dotarme de todos los recursos y facilidades que he necesitado.

Finalmente, agradezco a mis amigos de todas las etapas que siempre han creído en mí (en ocasiones más que yo mismo) su apoyo y motivación, que ha sido determinante en muchos momentos del camino que me ha llevado hasta aquí.

Resum

El Model Estàndard, que és el marc teòric físic més acceptat en l'actualitat, presenta certes deficiències i manca d'explicació per a algunes qüestions fonamentals sobre el comportament de la matèria. Per a resoldre aquestes qüestions es va crear, entre d'altres, el projecte ATLAS: dissenyat per a aprofitar al màxim el potencial del *Large Hadron Collider* en cerca de senyals de la denominada Nova Física. Aquests tipus de senyals es tracten d'una extensió de l'actual model que permetrien explicar les incògnites presents en l'actualitat. Comparant dades reals, obtingudes a partir de l'experimentació, amb dades simulades, procedents de models teòrics, és possible determinar si s'han trobat aquests senyals, validant així el descobriment de nous successos.

Com què la simulació de dades amb els mètodes tradicionals és temporalment i computacionalment molt costosa, nosaltres tractem d'utilitzar mètodes generatius amb un cost inferior que permeten generar esdeveniments de Monte Carlo simulats d'una qualitat similar, facilitant així el disseny de detectors i el descobriment de Nova Física.

En aquest treball, proposem utilitzar models generatius basats en Deep Learning com generadors ràpids d'esdeveniments de Monte Carlo i simuladors de detectors en el context del LHC, reduint el cost temporal i energètic dels mètodes actuals, i facilitant una estimació completa i detallada d'errors sistemàtics. En particular, explorem diferents escenaris de Nova Física i comparem el rendiment de les actuals tècniques que pertanyen a l'estat de l'art.

Paraules clau: Deep Learning, Esdeveniments de Monte Carlo, Nova Física, Models generatius basats en Deep Learning, Large Hadron Collider

Resumen

El Modelo Estándar, que es el marco teórico físico más aceptado en la actualidad, presenta ciertas deficiencias y carece de explicación para algunas cuestiones fundamentales sobre el comportamiento de la materia. El proyecto ATLAS, entre otros, se creó para resolver estas cuestiones. Este proyecto fue diseñado para aprovechar al máximo el potencial del *Large Hadron Collider* en la búsqueda de señales de la denominada Nueva Física. Estos tipos de señales se tratan de una extensión del actual modelo que permitirían explicar las incógnitas presentes en la actualidad. Comparando datos reales, obtenidos en base a la experimentación, con datos simulados, procedentes de modelos teóricos, es posible determinar si se han encontrado dichas señales, validando así el descubrimiento de nuevos sucesos.

Debido a que la simulación de datos con los métodos tradicionales es temporal y computacionalmente muy costosa, tratamos de utilizar modelos generativos con un coste inferior que permitan generar eventos de Monte Carlo simulados de una calidad similar, facilitando así el diseño de detectores y el descubrimiento de Nueva Física.

En este trabajo, proponemos utilizar modelos generativos basados en Deep Learning como generadores rápidos de eventos de Monte Carlo y simuladores de detectores en el contexto del LHC, reduciendo el coste temporal y energético de los métodos utilizados en la actualidad, y facilitando una estimación completa y detallada de errores sistemáticos. En particular, exploramos diferentes escenarios de Nueva Física y comparamos el rendimiento de técnicas que pertenecen al actual estado del arte.

Palabras clave: Deep Learning, Eventos de Monte Carlo, Nueva Física, Modelos generativos basados en Deep Learning, Large Hadron Collider

Abstract

The Standard Model, which is currently the most accepted physics theoretical framework, presents certain deficiencies and lacks an explanation for some fundamental aspects of the behavior of matter. Project ATLAS, among others, was created to solve those questions. This project was designed to make the most of the capabilities of the *Large Hadron Collider* while looking for signals of the so-called Physics Beyond the Standard Model. These kinds of signals are an extension of the current model that should be able to explain the current unknowns. Comparing real data, obtained from experimentation, with simulated data, coming from theoretical models, it is possible to determine if those signals have been found, validating the discovery of new events.

Because data simulation with traditional models is temporally and computationally very demanding, we try to use generative models with a lower cost that allow generating simulated Monte Carlo events with a similar quality, making the design of detectors and the discovery of Physics Beyond the Standard Model easier.

In this work, we propose the use of generative models based on Deep Learning as fast Monte Carlo event generators and detector simulators in the LHC context, reducing the time and energy cost of currently used methods, and making it easier to obtain a complete and detailed estimation of systematic errors. In particular, we explore several New Physics scenarios and compare the performance of existing state-of-the-art techniques.

Key words: Deep Learning, Monte Carlo events, New Physics, Deep Generative Models, Large Hadron Collider

Contents

Contents **v**

List of Figures **vii**

List of Tables **ix**

Acronyms **xi**

1 Introduction **1**
- 1.1 Motivation and problem description . . . 2
- 1.2 Objectives . . . 3
- 1.3 Structure of this work . . . 3

2 State of the art **5**
- 2.1 Analysis of the current situation . . . 6

3 The dataset **7**
- 3.1 Data generation . . . 7
- 3.2 Data format . . . 8

4 Proposed solution **11**
- 4.1 Autoencoders . . . 11
- 4.2 Variational Autoencoders . . . 12
  - 4.2.1 β-VAE . . . 12
  - 4.2.2 α-VAE . . . 13
- 4.3 Gaussian Mixture Models . . . 13

5 Used technologies **15**
- 5.1 Software environment . . . 15
  - 5.1.1 Model definition and training . . . 15
  - 5.1.2 Data representation . . . 16
- 5.2 Hardware infrastructure . . . 16

6 Experimentation **19**
- 6.1 Data partitioning . . . 19
- 6.2 Data analysis and preprocessing . . . 20
- 6.3 Experiment I: Initial model . . . 21
  - 6.3.1 Training results . . . 24
- 6.4 Experiment II: Input subdivision . . . 24
  - 6.4.1 Without BGMMs . . . 25
  - 6.4.2 With BGMMs . . . 36
- 6.5 Experiment III: Introducing the α-VAE . . . 43
  - 6.5.1 Without BGMMs . . . 44
  - 6.5.2 With BGMMs . . . 51
- 6.6 BSM event generation . . . 58

7 Conclusions **63**
- 7.1 Future work . . . 63

Bibliography **65**

Appendices

List of Figures in Appendices **69**

A Activation functions **73**
A.1 Linear 73
A.2 Rectified Linear Unit 73
A.3 Softmax 74

B Loss functions **75**
B.1 Categorical Crossentropy 75
B.2 Mean Squared Error 75

C Keras layers **77**
C.1 *Dense* 77
C.2 *BatchNormalization* 77
C.3 *Flatten* and *Reshape* 77

D Data distributions **79**
D.1 Standard model 79
D.2 Beyond the Standard Model 85

E Full experiment results **91**
E.1 Experiment I 91
E.2 Experiment II 94
E.2.1 Without BGMMs 94
E.2.2 With BGMMs 113
E.3 Experiment III 132
E.3.1 Without BGMMs 132
E.3.2 With BGMMs 140

F Sustainable Development Goals **171**

List of Figures

1.1 Computer-generated image of the ATLAS detector located at CERN. . . . 2

4.1 Scheme of the internal structure of a Variational Autoencoder. 12

5.1 Used software and hardware technologies diagram. 15
5.2 NVIDIA TU104 graphics chip scheme. 17

6.1 Representation of the circular buffer of open files when creating a batch. . 20
6.2 Histogram of b-jet object parameter E in the whole SM dataset. 20
6.3 Histogram of b-jet object parameter E after applying log_{10} in the SM dataset. 20
6.4 VAE architecture scheme for the initial model (variator omitted for simplicity). 22
6.5 Training the β-VAE using events from type T. 23
6.6 Training one BGMM per event type using encodings from the β-VAE. . . . 23
6.7 Sampling from the BGMM of events of type T using the β-VAE. 23
6.8 Histogram of first jet object parameter E with *ttbar* events ($\beta = 0.01, \gamma = 50$). 24
6.9 Histogram of first jet object parameter ϕ with *ttbar* events ($\beta = 0.01, \gamma = 50$). 24
6.10 VAE architecture scheme in experiment II (variator omitted for simplicity). 25
6.11 Generating events with the β-VAE without using BGMMs. 26
6.12 Histograms of `MET` event parameter in experiment II with different β values. 26
6.13 Histograms of `METphi` event parameter in experiment II with different β values. 27
6.14 Histograms of first jet parameter E in experiment II with different β values. 28
6.15 Histograms of first lepton parameter E in experiment II with different β values. 29
6.16 Histograms of first jet parameter η in experiment II with different β values. 30
6.17 Histograms of first lepton parameter η in experiment II with different β values. 31
6.18 Histograms of first jet parameter ϕ in experiment II with different β values. 32
6.19 Histograms of first lepton parameter ϕ in experiment II with different β values. 33
6.20 Histograms of first jet parameter p_T in experiment II with different β values. 34
6.21 Histograms of first lepton parameter p_T in experiment II with different β values. 35
6.22 Comparison of `MET` event parameter with and without BGMMs ($\gamma = 100$). 36
6.23 Comparison of `METphi` event parameter with and without BGMMs ($\gamma = 100$). 37
6.24 Comparison of first jet particle feature E with and without BGMMs ($\gamma = 100$). 38
6.25 Comparison of first lepton particle feature E with and without BGMMs ($\gamma = 100$). 38
6.26 Comparison of first jet particle feature η with and without BGMMs ($\gamma = 100$). 39
6.27 Comparison of first lepton particle feature η with and without BGMMs ($\gamma = 100$). 40
6.28 Comparison of first jet particle feature ϕ with and without BGMMs ($\gamma = 100$). 40

6.29 Comparison of first lepton particle feature ϕ with and without BGMMs ($\gamma = 100$). 41
6.31 Comparison of first jet particle feature p_T with and without BGMMs ($\gamma = 100$). 42
6.32 Comparison of first lepton particle feature p_T with and without BGMMs ($\gamma = 100$). 42
6.33 Training the α-VAE using one kind of event. 43
6.34 Generating events with the α-VAE without using BGMMs. 44
6.35 Histograms of `MET` event parameter in experiment III with different α values. 44
6.36 Histograms of `METphi` event parameter in experiment III with different α values. 45
6.38 Histograms of first jet parameter E in experiment III with different α values. 46
6.39 Histograms of first lepton parameter E in experiment III with different α values. 46
6.40 Histograms of first jet parameter η in experiment III with different α values. 47
6.42 Histograms of first lepton parameter η in experiment III with different α values. 48
6.43 Histograms of first jet parameter ϕ in experiment III with different α values. 48
6.44 Histograms of first lepton parameter ϕ in experiment III with different α values. 49
6.46 Histograms of first jet parameter p_T in experiment III with different α values. 50
6.47 Histograms of first lepton parameter p_T in experiment III with different α values. 50
6.48 Training the BGMM using the generated latent representations from the α-VAE. 51
6.49 Sampling from the BGMM to generate events from the α-VAE. 51
6.50 Comparison of `MET` event parameter with (🟢, 🔵) and without (🔴) BGMMs. 52
6.51 Comparison of `METphi` event parameter with (🟢, 🔵) and without (🔴) BGMMs. 52
6.52 Comparison of first jet parameter E with (🟢, 🔵) and without (🔴) BGMMs. 53
6.54 Comparison of first lepton parameter E with (🟢, 🔵) and without (🔴) BGMMs. 54
6.55 Comparison of first jet parameter η with (🟢, 🔵) and without (🔴) BGMMs. 54
6.56 Comparison of first lepton parameter η with (🟢, 🔵) and without (🔴) BGMMs. 55
6.58 Comparison of first jet parameter ϕ with (🟢, 🔵) and without (🔴) BGMMs. 56
6.59 Comparison of first lepton parameter ϕ with (🟢, 🔵) and without (🔴) BGMMs. 56
6.60 Comparison of first jet parameter p_T with (🟢, 🔵) and without (🔴) BGMMs. 57
6.62 Comparison of first lepton parameter p_T with (🟢, 🔵) and without (🔴) BGMMs. 58
6.63 Comparison of event parameter `MET` using the trained models in the BSM dataset. 59
6.64 Comparison of event parameter `METphi` using the trained models in the BSM dataset. 59
6.65 Comparison of first jet parameter E using the trained models in the BSM dataset. 59
6.66 Comparison of first lepton parameter E using the trained models in the BSM dataset. 60
6.67 Comparison of first jet parameter η using the trained models in the BSM dataset. 60
6.68 Comparison of first lepton parameter η using the trained models in the BSM dataset. 60
6.69 Comparison of first jet parameter ϕ using the trained models in the BSM dataset. 61
6.70 Comparison of first lepton parameter ϕ using the trained models in the BSM dataset. 61

6.71 Comparison of first jet parameter p_T using the trained models in the BSM dataset. 61
6.72 Comparison of first lepton parameter p_T using the trained models in the BSM dataset. 62

List of Tables

3.1 Definition of symbols representing final-state objects of each event. 8
3.2 Generated SM and BSM signal processes with their identification. 9

Acronyms

AE Autoencoder. 11

AI Artificial Intelligence. 16

ANN Artificial Neural Network. 73, 77

API Application Programming Interface. 15

BGMM Bayesian Gaussian Mixture Model. vii, viii, 13, 15, 21, 23–26, 36–44, 51–58, 63

BSM Beyond the Standard Model. v, ix, 1, 2, 5, 6, 9, 11, 58, 59, 61, 63, 69, 79, 90

CERN European Organization for Nuclear Research. 1, 3, 5

CPU Central Processing Unit. 16, 17

CUDA Compute Unified Device Architecture. 15

DNN Deep Neural Network. 11

GAN Generative Adversarial Network. 5, 6, 11, 63

GMM Gaussian Mixture Model. 13

GPU Graphics Processing Unit. 16, 17

LHC Large Hadron Collider. ii, iii, 1, 5

MC Monte Carlo. 2, 3, 5, 13, 28, 29, 31, 32, 34, 37, 41, 43, 46, 50, 54, 56–58, 62, 63, 91

MSE Mean Squared Error. 27, 36

PRHLT Pattern Recognition and Human Language Technology. 17

SM Standard Model. vii, ix, 1, 6, 9, 20, 21, 63, 69, 79, 84

VAE Variational Autoencoder. v, vii, viii, 1, 5, 6, 11–13, 21–23, 25, 26, 36, 43, 44, 51, 62, 63

CHAPTER 1
Introduction

Currently, the **Standard Model** is the theoretical framework that provides an explanation of how matter is constituted and describes the forces that govern the different physical processes exhibited by these constituents, with the Higgs boson [1] being the most recent discovery. However, there are some notable discrepancies in some observables that do not agree with theoretical predictions and the model does not explain several fundamental aspects that are key in physics. For instance, it cannot explain some experimentally observed phenomena like neutrino masses, gravity, and dark matter. The SM also features some unexplained parameters, such as the relatively low value of the Higgs mass despite its large quantum corrections, which implies a lack of understanding.

For that reason, at the Large Hadron Collider (LHC), the most powerful and largest collider in the world, located at the European Organization for Nuclear Research (CERN), particles are accelerated to high velocities and collided in order to study their interactions and to give an insight to the fundamental laws of nature.

At the LHC, there exists a program for obtaining **real data**. In that program, events from the proton-proton collisions (that are a very small fraction of the total) are recorded. Online selection of events is made from a trigger, and then the recorded events are processed and reconstructed. Data is collected from four collision points using seven detectors by different experiments [2] on the 27-kilometer circular ring of the LHC.

Throughout the life of an experiment, simulated data is produced in both fast and detailed ways in order to compare possible models with the characteristics of the real events. Then, results from analyses of the collected data are discussed.

In addition to precision measurements of Standard Model observables, experiments search for new Physics Beyond the Standard Model that could explain some of the previously mentioned shortcomings of the current model.

For now, a significant amount of data at the LHCb, ATLAS, ALICE and CMS experiments has been taken. ATLAS and CMS are so-called general purpose experiments with the objective of researching about the origin of masses and Physics Beyond the Standard Model. While there is no clear sign of confirmed new models of physics for now, it is expected that some promising results could follow when more experimentation is done. LHCb instead focuses on B physics, matter-antimatter asymmetry and CP violation. ALICE is a quite different experiment as it investigates Heavy Ion physics.

In this work, we focus on the ATLAS experiment (fig. 1.1), trying to use Deep **Generative Models**[1] to provide a faster way to simulate physical events. In particular, we focus on different variants of a particular type of model: the Variational Autoencoders (VAEs), which have already obtained significant results in similar experiments.

[1] Unsupervised learning tasks in machine learning that involve learning patterns in input data so that the model can be used to generate new samples that follow the same patterns found in the original dataset.

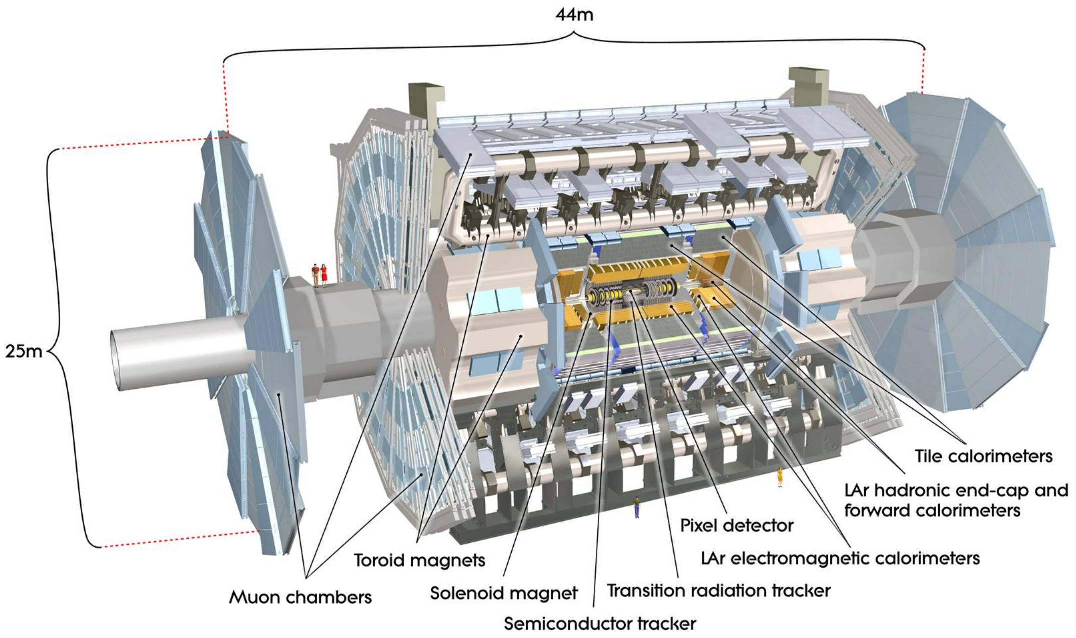

44m
25m
Tile calorimeters
LAr hadronic end-cap and forward calorimeters
Pixel detector
LAr electromagnetic calorimeters
Toroid magnets
Transition radiation tracker
Muon chambers
Solenoid magnet
Semiconductor tracker

Figure 1.1. Computer-generated image of the ATLAS detector located at CERN. [3]

1.1 Motivation and problem description

The simulation of physical processes following the Monte Carlo method is usually performed in two steps:

- Firstly, pseudo-random numbers are sampled.
- Secondly, these numbers are transformed by an algorithm into simulated physical events, which in this case are high energy particle collisions.

Nowadays, a relevant problem with these numerical simulations is their massive need for computational resources. Because of that, scientific progress related to this topic is restricted due to the speed of and budget for simulation. For example, the generation in particle physics experiments of an MC event, including the detector response, may take up to 10 minutes and depends on non-optimal algorithms like VEGAS [4].

In the next few years, billions of events will need to be generated due to the accumulation of data that will be obtained in the collider. On the one hand, the required generation times with current methods will be infeasible. On the other hand, if those methods were used, they would have an enormous impact on the environment in terms of energy consumption. This makes it imperative to optimize the event generation for it to be more efficient.

A significant speedup for signal studies could be achieved by accelerating the event generation pipeline with the help of artificial intelligence and, in particular, Deep Generative Models. This improvement would allow, for example, more extensive and accurate searches for signals of Physics Beyond the Standard Model.

However, the use of generative models will allow not only to produce simulated data in large quantities while saving temporal, energy and economic resources during physics studies and analysis. It will also allow to make a complete and detailed estimation of systematic errors. These errors always have to be estimated when measurements of different physical magnitudes are made and usually mass productions of simulated data are created taking into account the different relevant factors in obtaining these measurements.

Another issue is the inability to specify the properties of the events the simulation produces exactly. Data analysis often requires the generation of kinematically similar events to those seen in the data. Current event generators generate a large number of events and then select the interesting ones, discarding the rest of events. It is of interest to investigate how to avoid the generation of irrelevant events.

1.2 Objectives

After describing the experiment properties, we define some objectives to achieve as the result of this work in an effort to overcome some of its current difficulties.

First, we require to create a framework that is suitable for generating events from existing Monte Carlo data in a similar way to the current methods, taking as the starting point already generated events (known as *ground truth*).

Then, the obtained results will be compared with the classical approaches to verify that our newly designed framework performs similarly regarding data quality (i.e., events are generated accurately) while being more efficient in terms of time and/or required resources.

Finally, several approaches shall be considered to decide which one best suits the presented purpose, obtaining a balance of efficiency and quality of the generated events. It is important to obtain event distributions that adjust properly with real Monte Carlo events.

1.3 Structure of this work

This paper is structured as follows: in the first chapter, we first make a brief explanation of the problem and its related ongoing experiments at CERN, followed by a description of the current shortcomings and a description of the defined objectives to accomplish. Then, in the following chapter, we analyze the current state-of-the-art with a concise description of classical and similar approaches, together with machine learning solutions that are close to what we try to achieve.

After that, in chapter three, we describe the dataset that we are working with, detailing the types of events and particles, the information that is recorded for each of them, and the format in which data is provided. In chapter four we propose our own solution, attempting to improve the models of the previously mentioned references and obtaining a valid model that meets the specified objectives. Then, before beginning with the details of the practical work, we present the technologies that make all the experiments possible.

In chapter six, we present and describe our experimentation results, explaining the framework for each experiment and focusing on the results and some analysis that guides the exploration of different models until reaching the best approach. We conclude in chapter seven with an interpretation of how suitable the chosen approach is for the desired experiment, and which is the best model architecture comparing all the studied possibilities.

CHAPTER 2
State of the art

Several attempts have been made in the past aiming to detect signals of new physics at the LHC using supervised and unsupervised machine learning algorithms. For now, no signal of BSM physics was found, so our methods are crucial to facilitate the investigation and to create a solid framework for future experiments.

Among the already performed experiments, the closest approach to the study presented here was a study for the generation of events from a physical process [4], that already used Deep Generative Models. That study investigated the feasibility of learning the event generation and the correct frequency of occurrence to produce events like MC generators. In the beginning, several Generative Adversarial Network architectures with recommended hyperparameters and the Variational Autoencoder with a standard normal prior failed to correctly produce events with the right frequency of occurrence. Finally, a particular VAE setup, the β-VAE, obtained promising results. By buffering density information of encoded events given the encoder of a VAE, it was possible to construct a prior for the sampling of new events from the decoder, obtaining distributions that were in very good agreement with real MC events and were generated several orders of magnitude faster.

Another relevant event in the scientific community related to this work was *The Dark Machines Anomaly Score Challenge*, whose results are detailed in [5]. The dataset used in our work comes from this challenge, and its properties are detailed in chapter 3. The challenge aimed to detect signals of new physics at the LHC using unsupervised machine learning algorithms. First, an implementation of an anomaly score to define model-independent signal regions was proposed. Then, a wide range of anomaly detection and density estimation algorithms, developed in the context of the data challenge, was reviewed and their performance in a set of realistic analysis environments was measured. This challenge lays the foundation for new benchmarks and the development of better approaches for detecting BSM signals.

The ATLAS project team at CERN also investigated in this matter, defining their own strategy to find potential indications of new physics [6]. In their approach, events were classified according to their final state into many event classes. For each event class an automated search algorithm tested whether the data were compatible with the MC simulated expectation in several distributions sensitive to the effects of new physics. The significance of a deviation was quantified using pseudo-experiments. A data selection with a significant deviation defined a signal region for a dedicated follow-up analysis with an improved background expectation. The analysis of the data-derived signal regions on a new dataset allowed statistical interpretation. The sensitivity of the approach was discussed using Standard Model processes and benchmark signals of new physics.

To conclude, it is worth mentioning several approaches that were explored in different areas of the event processing field of study. Regarding **event generation**, there have been a wide number of attempts to use GAN-based models [7, 8, 9, 10], but the possibilities of VAEs have not been explored deeply. Concerning **detector simulation**, we can find some GAN-based models [11, 12, 13], but also some VAE-based ones [11, 12, 14]. Also, in relation to **Monte Carlo integration**, we observe the use of regression in many studies [15, 16] and also some attempts to use GANs [17]. Finally, recent research has also been made in the use of Flow models [18].

2.1 Analysis of the current situation

After analyzing some of the most relevant and recent contributions related to the topic of the present work, we obtain several relevant conclusions and observe specific facts that allow us to have a perspective of the status we are at:

- Most physics research has been done using supervised learning methods. In the last years, unsupervised learning has started to become popular in the community, so there is still a lot to explore about that topic.
- The study performed in [4] already experimented with VAEs. However, that experiment featured a very limited number of processes.
- The data challenge described in [5] created a solid framework for evaluating the performance of different event generating approaches, without focusing on a particular method.
- While there have been many attempts to use GANs for event generation and detector simulation, the possibilities of VAEs have been less extensively explored.

Taking the previous observations into account, we focus here on exploring the possibilities of unsupervised learning by using VAE-based models while considering a dataset with a wide variety of SM and BSM processes, trying to create a valid model that can be trained for any type of process.

CHAPTER 3
The dataset

As mentioned in the previous chapter, the dataset that we used for our experimentation is one of those generated to be used in *The Dark Machines Anomaly Score Challenge* [5], and it can be obtained in [19]. A detailed explanation of the data generation procedure format is described in [20], but we give here a brief summary with the most relevant details for our study.

3.1 Data generation

The simulated events are proton-proton collisions similar to those occurring at the LHC at a center-of-mass energy of 13 TeV[1]. The generation of events for the signal and the background processes was performed using classical methods and tools. After generation, the final-state physics objects as described in table 3.1 were stored in a one-line-per-event CSV text file.

A collision event results in up to 20 objects, and for each of them the full energy (E), transverse momentum (p_T), pseudo-rapidity (η), and azimuthal angle (ϕ), are recorded. An event is stored (i.e., included in the dataset) when at least 1 of the following requirements is fulfilled:

- At least one **jet** or ***b*-jet** with $p_T > 60$ GeV and $|\eta| < 2.8$
- At least one **electron** with $p_T > 25$ GeV and $|\eta| < 2.47$, except for $1.37 < |\eta| < 1.52$
- At least one **muon** with $p_T > 25$ GeV and $|\eta| < 2.7$
- At least one **photon** with $p_T > 25$ GeV and $|\eta| < 2.37$.

Some additional requirements that impacted the `event weight` were applied. However, both in the original data challenge and in this study, that variable is not considered for simplicity. The requirements on the final objects stored in the dataset were:

- **jet** or ***b*-jet**: $p_T > 20$ GeV and $|\eta| < 2.8$
- **electron** or **muon**: $p_T > 15$ GeV and $|\eta| < 2.7$
- **photon**: $p_T > 20$ GeV and $|\eta| < 2.37$

[1] Amount of kinetic energy gained or lost by a single electron accelerating from rest through an electric potential difference of 1 volt in vacuum.

The aim was to create a flexible dataset that allowed for several types of studies that might require different selection criteria. The full list of generated processes can be seen in table 3.2.

Symbol ID	Object
j	jet
b	b-jet
e-	electron (e^-)
e+	positron (e^+)
m-	muon (μ^-)
m+	antimuon (μ^+)
g	photon (γ)

Table 3.1. Definition of symbols representing final-state objects of each event.

3.2 Data format

As introduced in the previous section, data is provided in a one-line-per-event CSV file, where each line has variable length and contains 3 event-specifiers, followed by the kinematic features for each object in the event.

Each file corresponds to a process. For each of them, the total number of generated events is at least the number that is needed for obtaining $10fb^{-1}$ of data[2] ($N_{10fb^{-1}}$). Additionally, when $N_{10fb^{-1}} < 20000$, a second file with 20000 lines is provided. The format of all CSV files is:

```
event ID; process ID; event weight; MET; METphi; obj1, E1, pt1, eta1, phi1;
obj2, E2, pt2, eta2, phi2; ...
```

The `event ID` is an event specifier. It is an integer to identify the generation of a particular event for debugging purposes. The `process ID` is a string referring to the process that generated the event, as mentioned in table 3.2. The `event weight` depends on the cross section for a particular process and the number of events in a single file, but it is not taken into account in our experiments. These conclude the event specifiers of each line in the CSV file.

Regarding the kinematic features, the `MET` and `METphi` entries are the magnitude E_T^{miss} and the azimuthal angle $\phi_{E_T^{miss}}$ of the missing transverse energy vector of the event. These values represent the transverse energy and azimuthal angle of those objects that genuinely escape detection.

The object identifiers (`obj1`, `obj2`, ...) are strings identifying each object in the event, following table 3.1. Each object identifier is followed by 4 comma-separated values fully specifying the features of the object: `E1` (E), `pt1` (p_T), `eta1` (η), `phi1` (ϕ). The features correspond with those mentioned in the previous section.

The ordering of particles inside each event shows, in this order: b-jets, jets, leptons, and photons. Inside each type, they are sorted in descending order according to their p_T. Therefore, in the results section when referring to the *leading* particle, we consider the one with the highest p_T inside its class.

All of the events are separated by process in 65 files with a total size of 66.5 GB of data. However, for the experimentation part only the largest files for each type of process

[2] The inverse femtobarn is a measurement of particle-collision events per femtobarn; a measure of both the collision number and the amount of data collected ($1fb^{-1} \approx 10^{12}$ proton-proton collisions).

were chosen (i.e., for processes that reached $10fb^{-1}$ with less than 20K events, the file with 20K events was chosen).

SM processes	
Physics process	*Process ID*
$pp \to jj$	njets
$pp \to W^{\pm}(+2j)$	w_jets
$pp \to \gamma(+2j)$	gam_jets
$pp \to Z(+2j)$	z_jets
$pp \to t\bar{t}(+2j)$	ttbar
$pp \to W^{\pm}t(+2j)$	wtop
$pp \to W^{\pm}\bar{t}(+2j)$	wtopbar
$pp \to W^{+}W^{-}(+2j)$	ww
$pp \to t + jets(+2j)$	single_top
$pp \to \bar{t} + jets(+2j)$	single_topbar
$pp \to \gamma\gamma(+2j)$	2gam
$pp \to W^{\pm}\gamma(+2j)$	Wgam
$pp \to ZW^{\pm}(+2j)$	zw
$pp \to Z\gamma(+2j)$	Zgam
$pp \to ZZ(+2j)$	zz
$pp \to h(+2j)$	single_higgs
$pp \to t\bar{t}\gamma(+1j)$	ttbarGam
$pp \to t\bar{t}Z$	ttbarZ
$pp \to t\bar{t}h(+1j)$	ttbarHiggs
$pp \to \gamma t(+2j)$	atop
$pp \to t\bar{t}W^{\pm}$	ttbarW
$pp \to \gamma\bar{t}(+2j)$	atopbar
$pp \to Zt(+2j)$	ztop
$pp \to Z\bar{t}(+2j)$	ztopbar
$pp \to t\bar{t}t\bar{t}$	4top
$pp \to t\bar{t}W^{+}W^{-}$	ttbarWW

BSM processes	
Physics process	*Process ID*
$pp \to \tilde{g}\tilde{g}$ (1 TeV)	gluino_01
$pp \to \tilde{g}\tilde{g}$ (1.2 TeV)	gluino_02
$pp \to \tilde{g}\tilde{g}$ (1.4 TeV)	gluino_03
$pp \to \tilde{g}\tilde{g}$ (1.6 TeV)	gluino_04
$pp \to \tilde{g}\tilde{g}$ (1.8 TeV)	gluino_05
$pp \to \tilde{g}\tilde{g}$ (2 TeV)	gluino_06
$pp \to \tilde{g}\tilde{g}$ (2.2 TeV)	gluino_07
$pp \to \tilde{t}_1\tilde{t}_1$ (220 GeV), $m_{\tilde{\chi}_1^0} = 20$ GeV	stop_01
$pp \to \tilde{t}_1\tilde{t}_1$ (300 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	stop_02
$pp \to \tilde{t}_1\tilde{t}_1$ (400 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	stop_03
$pp \to \tilde{t}_1\tilde{t}_1$ (800 GeV), $m_{\tilde{\chi}_1^0} = 100$ GeV	stop_04
$pp \to Z'$ (2 TeV)	Zp_01
$pp \to Z'$ (2.5 TeV)	Zp_02
$pp \to Z'$ (3 TeV)	Zp_03
$pp \to Z'$ (3.5 TeV)	Zp_04
$pp \to Z'$ (4 TeV)	Zp_05

Table 3.2. Generated SM and BSM signal processes with their identification.

CHAPTER 4

Proposed solution

Taking into account the current state of the art as discussed in chapter 2, and after deciding and analyzing the dataset to work with, we propose a solution based in using a Variational Autoencoder (VAE) to try to obtain relevant results while generating events.

The approach of using VAEs has been chosen due to the existence of less studies choosing this model and deciding to use GANs instead, mostly due to the complexity of the theoretical foundations of the former. Together with GANs and Flow models, VAEs are the most popular and used kind of Deep Generative Models in the AI field, and we believe that promising results can be obtained with this technique.

When we obtain a model that properly fits the data distribution of our dataset, this will allow for more extensive searches for signals of Physics Beyond the Standard Model, making it possible to confirm new theoretical models. A brief description of the theoretical concepts that define autoencoders and its evolution into the architecture of VAEs follows.

4.1 Autoencoders

We define an autoencoder (AE) [21] as a class of Deep Neural Network composed by a central hidden layer of lower dimension than the input layer, and a target space coinciding with the input space. They can be trained to reconstruct the features of the input data, with a bottleneck structure that prevents them from simply learning the identity map. The dimension of the central layer is a critical variable, as it determines the amount of compression of the features.

The AE architecture is therefore deconstructed into:

- An **encoder** part, which compresses the input data into the latent space.
- A **decoder** part, which uses information from the lower dimensional latent space to extrapolate to the full input space.

This class of DNN has already been used in some High Energy Physics applications, such as [22]. For instance, an AE trained extensively on SM data and reaching reasonably low reconstruction errors could be used as an anomaly detection algorithm. When presented with new data, the model could tag anomalous events from their comparatively higher reconstruction error.

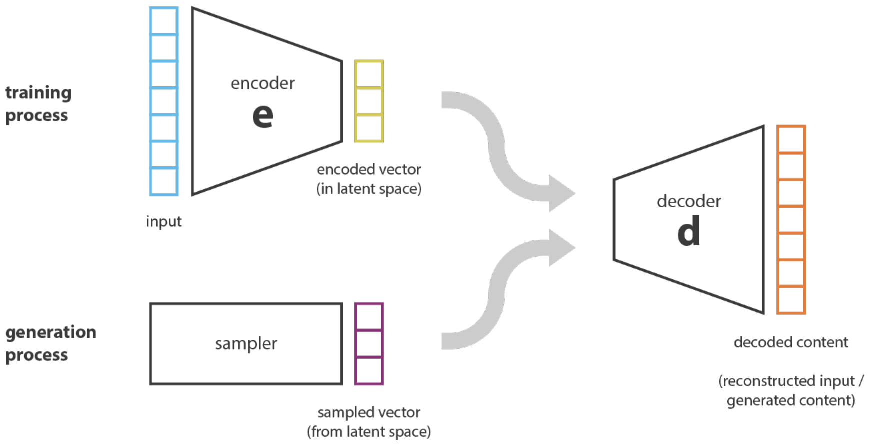

training process
input
encoder
e
encoded vector (in latent space)
generation process
sampler
sampled vector (from latent space)
decoder
d
decoded content
(reconstructed input / generated content)

Figure 4.1. Scheme of the internal structure of a Variational Autoencoder. [23]

4.2 Variational Autoencoders

A Variational Autoencoder (VAE) [23, 21] can be defined as a class of autoencoder architecture whose encodings distribution is regularised during the training process to avoid overfitting and to guarantee that its latent space has good properties, allowing to generate new data from arbitrary numbers. The term *variational* comes from the close relation that exists between the regularisation and the variational inference method known in statistics.

Similarly to a standard autoencoder, the architecture of a VAE (fig. 4.1) is composed of both an **encoder** and a **decoder**. However, in this approach the output is equal to the input, so the model is trained to minimise the reconstruction error between the encoded-decoded data and the initial data.

In order to introduce some regularisation of the latent space, during the encoding-decoding process a small modification is made: instead of encoding an input as a single point, it is encoded as a distribution over the latent space. Here, the bottleneck layer is generated by letting the encoder output two numbers per latent space dimension, which generally represent a mean and standard deviation for a normal distribution. This way, we ensure both a local (variance) and global (mean) regularisation of the latent space.

We can describe the **training process** as follows. First, the input is encoded as a distribution over the latent space. Then, a point from the latent space is sampled from that distribution. After that, the sampled point is decoded and the reconstruction loss can be computed. Finally, the error is backpropagated through the network.

The **loss function** that is minimised when training a VAE is composed of a *reconstruction term* (on the final layer), that tends to improve the performance of the encoding-decoding scheme, and a *regularisation term* (on the latent layer), that is proportional to the Kullback-Leibler (KL) divergence and tends to regularise the organisation of the latent space by making the distributions returned by the encoder close to a standard normal distribution.

4.2.1. β-VAE

In some cases, assigning the same importance to both terms of the loss function does not give proper results. In order to circumvent this problem, sometimes it is a good approach to balance out the relative importance of the reconstruction term and the regularisation

(KL-divergence) term. To do so, a term β is sometimes introduced as a hyperparameter to control this relative importance. The loss function then becomes:

$$L_{VAE} = (1 - \beta)MSE + \beta KL \tag{4.1}$$

The version of Variational Autoencoders using this form of the loss function are known as β-VAE. We propose using this kind of model for generating accurate MC events during our experimentation, as it is described in section 6.

4.2.2. α-VAE

In this work, we also present a new approach consisting of an alternative model of Variational Autoencoder. This version, that we call α-VAE, features some differences when compared to the previously described β-VAE.

The main feature is that, after encoding the input in the usual way, and together with a normal distribution with $\mu = 0$ and $\sigma = \alpha$, it adds Gaussian noise to the latent representation of the input data. This effectively replaces the Kullback-Leibler divergence layer.

We introduce this kind of model for generating events during our experimentation, also describing our implementation details, in section 6.5.

4.3 Gaussian Mixture Models

A Gaussian Mixture Model (GMM) [24] is a parametric probability density function represented as a weighted sum of Gaussian component densities. They are commonly used as a parametric model of the probability distribution of continuous measurements or features in unsupervised learning. The parameters are estimated from training data using the Maximum *A Posteriori* (MAP) estimation from a well-trained prior model or the iterative Expectation-Maximization (EM) algorithm.

In our case, we use a Bayesian Gaussian Mixture Model (BGMM), that implements a variant of the GMM with variational inference algorithms. The implementation included in Scikit-learn includes two types of prior for the weights distribution: a finite mixture model with *Dirichlet distribution*, and an infinite mixture model with the *Dirichlet Process*. In practice, the latter is approximated and uses a truncated distribution with a fixed maximum number of components. The number of components actually used almost always depends on the data.

The relevant parameters considered here were the type of the weight concentration prior explained before, the number of maximum components to use, and the type of covariance parameters to use. Regarding this last aspect, there were four possible choices, from which we focused on two of them:

- **Full**, where each component has its own general covariance matrix
- **Diagonal**, where each component has its own diagonal covariance matrix
- Tied, where all components share the same general covariance matrix
- Spherical, where each component has its own single variance

After initially deciding to focus on the full and diagonal covariance matrix choices, in the end only the first option was used due to noticing a high correlation in the different variables.

It must also be noted that the number of components is strongly related with the training time, increasing noticeably when the number of components increased, and therefore not allowing to perform as many experiments as it would be desirable due to time limitations.

CHAPTER 5

Used technologies

After many years of study and development, today we can find a large number of libraries and tools that make developing models and testing different configurations in the field of AI very straightforward.

During the development of this project, several software and hardware tools (fig. 5.1) have been chosen to help building and training the designed models without requiring to know all the theoretical foundations and low-level implementation details.

5.1 Software environment

Regarding the software environment, the system used during our work was Linux-based, and every experiment was coded using Python. This language was chosen as it is the most widely used language in the field with support from all major libraries.

5.1.1. Model definition and training

For our experimentation, Tensorflow and Keras were chosen as the libraries for model definition and training, effectively defining a solid framework for running the whole process. Together with both of these libraries, Scikit-learn was also used to include the BGMM component into the architecture.

TensorFlow [26] is a flexible and scalable software library for numerical computations using data-flow graphs. This library and related tools enable users to efficiently program and train neural network and other machine learning models and deploy them to production. The core algorithms of this library are written in highly optimized C++ and CUDA, a parallel computing platform and API created by NVIDIA. It has APIs available

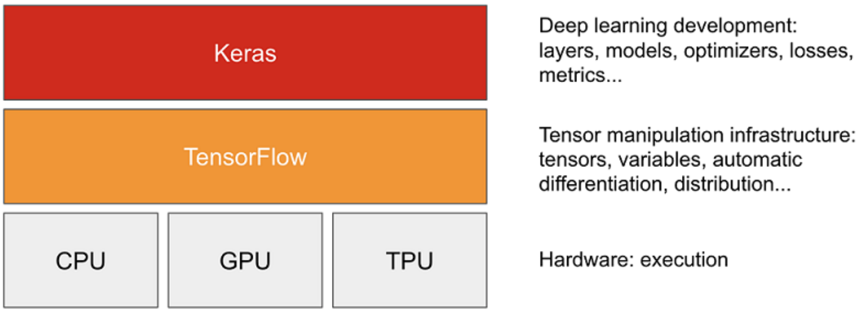

Keras
Deep learning development: layers, models, optimizers, losses, metrics...
TensorFlow
Tensor manipulation infrastructure: tensors, variables, automatic differentiation, distribution...
CPU
GPU
TPU
Hardware: execution

Figure 5.1. Used software and hardware technologies diagram. [25]

in several languages, being the Python implementation the most complete and stable one. Other officially supported languages include JavaScript, C++, Java, Go, and Swift.

Keras [27] is a compact and easy-to-learn high-level Python library for deep learning that runs on top of TensorFlow, which acts as the backend. It allows developers to focus on the main concepts of deep learning, such as creating layers for neural networks, while taking care of the internal details of tensors, their shapes, and their mathematical particularities. Keras allows running deep learning applications without interacting with the relatively complex TensorFlow. There are two major kinds of framework: the sequential API and the functional API. The sequential API is based on the idea of a sequence of layers and is the most commonly found.

Scikit-learn [28] is a Python module integrating a wide range of state-of-the-art machine learning algorithms for medium-scale supervised and unsupervised problems. It focuses on bringing machine learning to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency. It has minimal dependencies and its use is encouraged in both academic and commercial settings. In this work, it was mainly used to implement Bayesian Gaussian Mixture Models as a component of the different experimentation frameworks that we designed.

5.1.2. Data representation

For determining the quality of the obtained results and for evaluating whether they reach the desired objectives, a visual and easy to analyze data representation must be created. To do so, several well-known libraries and software were also used in this work. Precisely, we used Matplotlib and Netron to display our data results and model architectures, respectively.

Matplotlib [29] is a 2D plotting and imaging Python package aimed primarily at visualization of scientific, engineering, and financial data. It can be used from the Python shell or scripts, or embedded in a GUI application. Many popular hardcopy outputs are supported including JPEG, PNG, PostScript and SVG. Features include the creation of multiple axes and figures per page, interactive navigation, many predefined line styles and symbols, images, antialiasing, alpha blending, date and financial plots, W3C compliant font management and FreeType2 support, legends and tables, pseudocolor plots, mathematical text and more.

Netron is a multiplatform software designed to be a viewer for neural network, deep learning and machine learning models that creates an easy-to-understand diagram representing the distribution of layers of such model, allowing to identify its different parts and showing basic information about every component. It also allows exporting the model schemes into popular formats such as SVG or PNG.

5.2 Hardware infrastructure

The training and testing process of an AI model requires performing a high amount of calculations of a particular type, and it usually needs to repeat experiments while trying different configurations. Concerning hardware, GPUs are the most optimal device to compute the required calculations, achieving a higher performance when compared with CPUs.

Because of that, the hardware infrastructure devoted to this project consisted of a server featuring 128 GB of DDR4 RAM memory and two high-performance *NVIDIA*

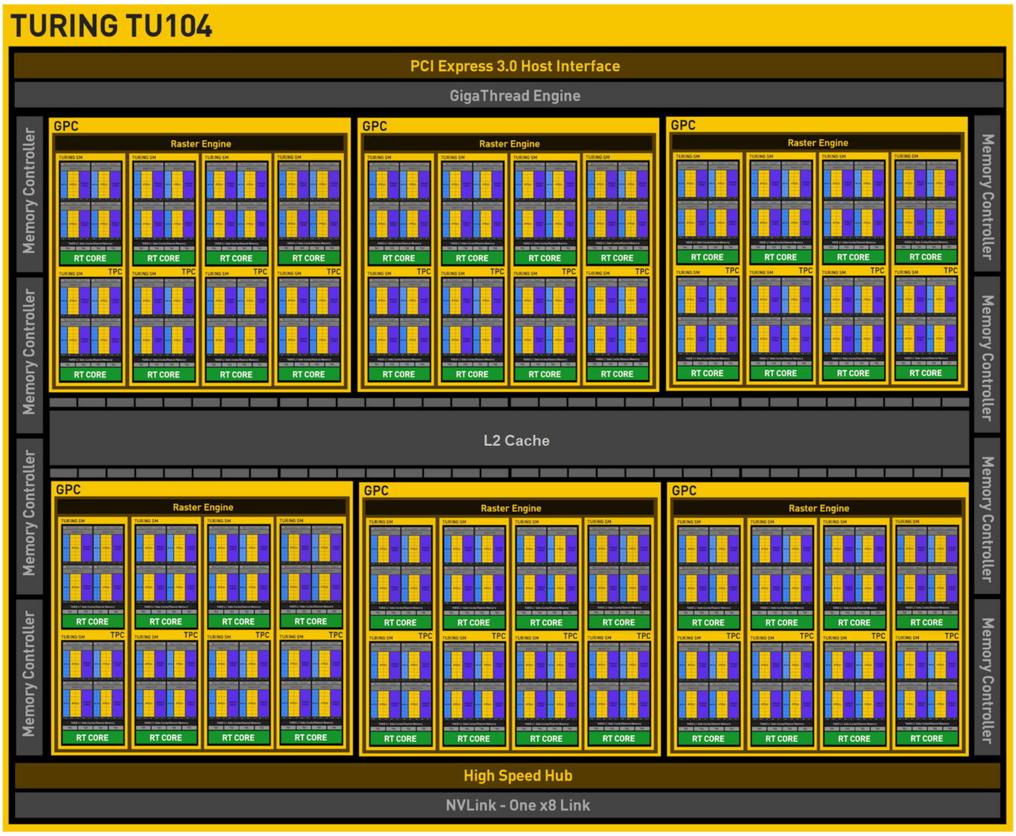


Figure 5.2. NVIDIA TU104 graphics chip scheme. [30]

GeForce RTX 2080 graphics cards, that include the TU104 chip (figure 5.2) with several machine learning specifically designed components, and feature 2944 CUDA cores each.

However, the server also had a high-performance CPU: the Intel Core i7-7800X, with 6 cores and 12 threads running at 4 GHz. This processor was crucial for training the BGMM component of our models, that was not implemented with GPU processing support.

The availability of these computing resources was possible thanks to the infrastructure located at the Pattern Recognition and Human Language Technology (PRHLT) research center, which belongs to the *Universitat Politècnica de València*.

CHAPTER 6

Experimentation

In this section we describe the development and experimentation that was carried out following the proposed solution detailed in chapter 4 and using the technologies that were described in chapter 5, analyzing the results of each experiment.

Before getting into details of the different approaches, we first specify some details about the required data partitioning of our large dataset, and the preprocessing that was applied following the observations of that analysis.

6.1 Data partitioning

The dataset used for all the experiments contains a large amount of data, with a total of 66.5 GB, as it was described in chapter 3. Although a plain text representation of the data would fit into the system that we used, which contained 128 GB of main memory, the internal representation required together with the applied preprocessing to train some of the models made the data size too big for our machine. Moreover, it is always desirable to perform some optimization in order to load in memory only the data that is useful at every moment to the model and not the whole dataset.

Because of that, it was needed to develop a strategy to process this dataset by loading small portions or **batches** of data, one at a time, using every file uniformly. To do so, we created our own implementation of the *Sequence*[1] class provided by the TensorFlow library. This class allows their instantiated objects to be provided as the input and output of Keras models, and provides the interface of the required functions to handle batch processing appropriately.

The class has a variable representing the **batch size**[2], another one defining the total amount of batches, and it also includes every file opened (but not loaded in memory) to extract events from it on demand. When the model requests a batch to be loaded, the list of files is traversed as a circular array, obtaining 1 event from each file (fig. 6.1), and therefore ensuring to load events from every file distributing them equally. Furthermore, when the end of a file is reached it is also treated as a *circular array of lines*, and its internal pointer returns to the beginning. In this way, we avoid to penalize processes with a smaller amount of events.

In all our experiments, we used a batch size of 10000 events and 1000 batches in each **epoch**[3], meaning that VAEs received a total of 10^7 events per epoch.

[1] https://www.tensorflow.org/api_docs/python/tf/keras/utils/Sequence

[2] Amount of events included in each batch.

[3] Hyperparameter that defines how many times the learning algorithm goes through the training dataset.

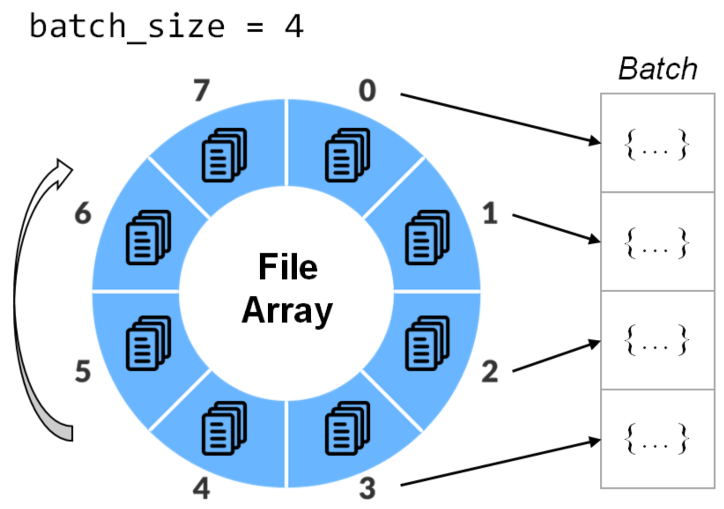


Figure 6.1. Representation of the circular buffer of open files when creating a batch.

6.2 Data analysis and preprocessing

Before performing any experiment, the dataset was first analyzed to see if the different variables followed any known distribution. After considering separately all kinds of particles that can be found in the events (listed in table 3.1) it was then observed that, among the four features that describe each particle, E and p_T seemed to have the shape of a log-normal distribution[4] as it can be seen in figs. 6.2 and 6.3, and ϕ followed a uniform distribution, while η did not seem to describe any known shape. Most of the histograms that were generated during this data analysis process are available in annex D.

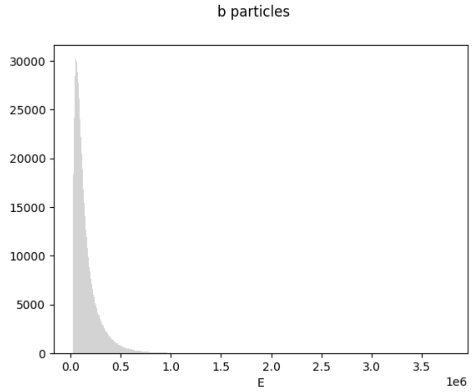


Figure 6.2. Histogram of b-jet object parameter E in the whole SM dataset.

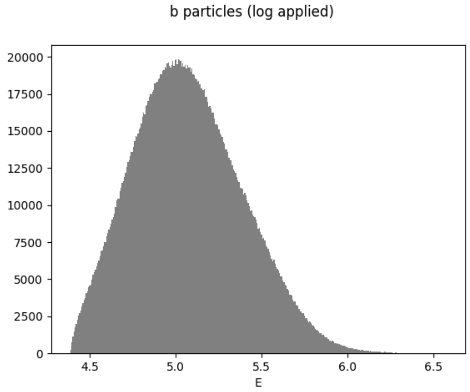


Figure 6.3. Histogram of b-jet object parameter E after applying log_{10} in the SM dataset.

Then, after carrying out the described analysis, the data was converted to a proper format to be provided as input to our model, applying some transformations as a consequence of the previously obtained information. First, it was decided that each **event** would be described as a tensor composed of 20 objects with four features per **object**.

Apart from the object types described in table 3.1, the `MET` and `METphi` values were included as a special type of object with $E = MET$ and $\phi = MET\phi$, leaving the other values set to 0. After that, if an event consisted of $k < 20$ objects, the $(20 - k)$ remaining objects were filled using a special *mask* object, with a special ID and all fields set to 0.

[4] Continuous probability distribution of a random variable whose logarithm is normally distributed.

In each object, the features E and p_T were converted into $log_{10}E$ and $log_{10}p_T$, and the symbol ID was represented using *One-Hot Encoding*[5]. The reason for this last choice instead of a numerical constant was to avoid the algorithm to misinterpret that constant as having some meaning apart from a simple identification of the object.

After applying these transformations, the final format of an event would be:

$(0, 0, 0, 0, 0, 0, 0, 1, 0)$	$(log_{10}MET, 0, 0, MET\phi)$,	$MET/MET\phi$ special object
$(x, x, x, x, x, x, x, x\ x)$	$(log_{10}E, log_{10}p_T, \eta, \phi)$,	Particle object
$(x, x, x, x, x, x, x, x\ x)$	$(log_{10}E, log_{10}p_T, \eta, \phi)$,	Particle object
... ,		
$(0, 0, 0, 0, 0, 0, 0, 0, 1)$	$(0, 0, 0, 0)$	*Mask* special object

where there is a total of 20 objects, and the vectors filled with x characters represent the *One-Hot Encoding* of the different objects present in each event. The result is a tensor of dimension $20x9$ containing object identification, and another one of dimension $20x4$ containing object features. With the data already analyzed and converted to a format that is ideal for our experiments, we can begin trying different configurations of our model.

6.3 Experiment I: Initial model

In our investigation to find the optimal model for generating events, we decided to focus in using one kind of event from the Standard Model which had lots of event samples: *ttbar*, and then to apply that model to other types of events. In this process, defined in table 3.2, the proton collision produces two top quarks which decay to other particles.

The model that we first designed for generating the events features a β-VAE and a BGMM as its main components. The whole process is divided in three steps, and is followed by the evaluation part.

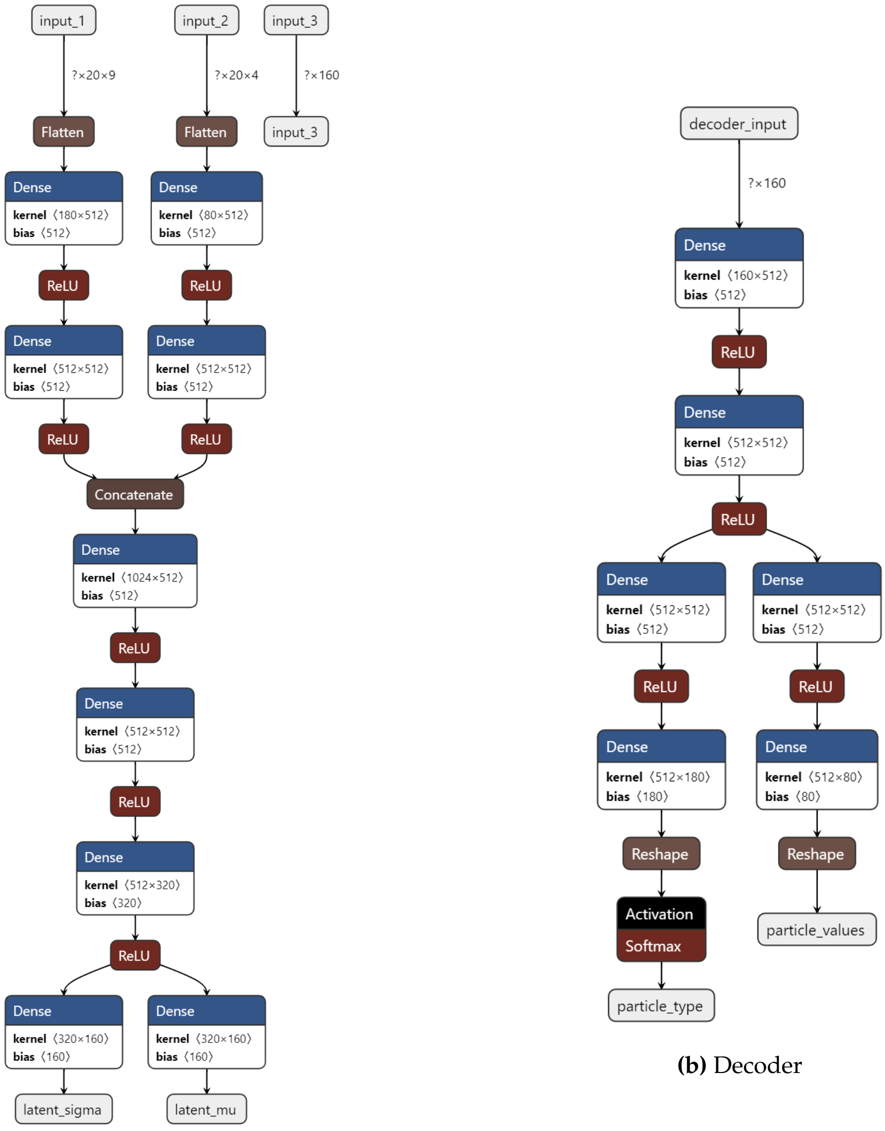
First, a Variational Autoencoder is trained for 100 epochs using **all the events of the *ttbar* dataset** (fig. 6.5). This VAE has three sub-components, unlike the usual autoencoder architectures. This design choice was made for convenience, in order to connect some VAE components with the BGMM in further steps. Our implementation is structured as follows (fig. 6.4):

- an **encoder**, which transforms the input data in the format described in section 6.2 into an encoded representation in the latent space, producing the logarithm of the variance and mean vector for all latent space dimensions as outputs.

  The encoder of this first model features a *Flatten* layer, followed by two blocks composed by a *Dense* layer of 512 nodes and a ReLU activation function, all of it applied to each of the two inputs separately. Then, both inputs are concatenated and two blocks of a *Dense* layer with 512 nodes + a ReLU activation function are applied after the concatenation. After that, a *Dense* layer of twice the latent dimensionality ($20x8$) together with Batch Normalization is added to the model. Finally, we put two *Dense* layers of as many nodes as the latent dimensionality to sample the means and logarithms of the variance, providing the data that the next component requires.

- a **variator**, that receives the outputs of the encoder and, together with a randomly generated ϵ from a normal distribution with $\mu = 0$ and $\sigma = 1$, calculates the Kullback-Leibler divergence and outputs z, calculated as follows:

[5] A vector with as many values as the amount of different objects where, for each object, a 1 is placed in the position in which that object ID belongs, leaving the others with 0.



(b) Decoder

(a) Encoder

Figure 6.4. VAE architecture scheme for the initial model (variator omitted for simplicity).

$$z = (\epsilon * e^{log(\sigma^2)*0.5}) + \mu$$

The dimensions of the z tensor match the latent dimensionality. The variator component is usually part of the encoder in most implementations.

- a **decoder**, that transforms the encoded data from the latent space back into its original dimensions.

  The decoder of this approach is composed by two blocks of a *Dense* layer of 512 nodes, followed by a ReLU activation layer. Then, the output is divided into the particle type and particle layer tensors, with a *Dense* layer of 512 nodes in each + a ReLU activation function, followed by another *Dense* layer with linear activation of $20x9 = 180$ and $20x4 = 80$ nodes respectively, connected to a *Reshape* layer to *unflatten* the data. In the case of the particle type, a final Softmax activation layer is included.

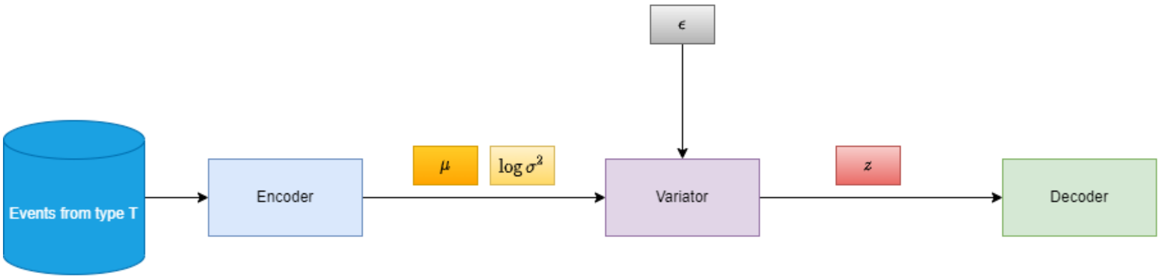


Figure 6.5. Training the β-VAE using events from type T.

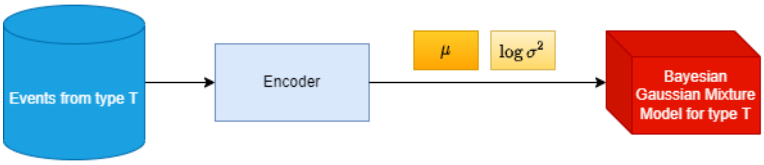


Figure 6.6. Training one BGMM per event type using encodings from the β-VAE.

Our VAE used *Categorical Crossentropy* as the loss function for the particle type detection, and the *Mean Squared Error* for learning the different object features. See appendix B for a detailed description of loss functions.

It must be noted that a variant of this VAE using Batch Normalization after some of the *Dense* layers was also tested, but it was finally discarded due to the lack of improvement in the results.

Second, the already trained encoder is detached from the rest of the VAE and its output becomes the input of a Bayesian Gaussian Mixture Model (fig. 6.6) in form of 200K generated encoded events. This BGMM is trained **for a particular type of process**, creating a model that is capable of generating valid encodings for that particular kind of events.

Third, the output of each BGMM is connected to the rest of the previously trained VAE (fig. 6.7), producing new event encodings that can be decoded into the desired events.

Once this procedure is defined, we can now test several VAE configurations and tweak the parameters of the model to look for the optimal one.

Several tests were performed, combining different values of β with a varying number of BGMM components γ, but always using a full covariance matrix for each component. In particular, the following values for both variables were tested:

$$\beta \in \{0.0, 0.001, 0.01, 0.1, 0.2, 0.5\}$$
$$\gamma \in \{10, 50, 100\}$$

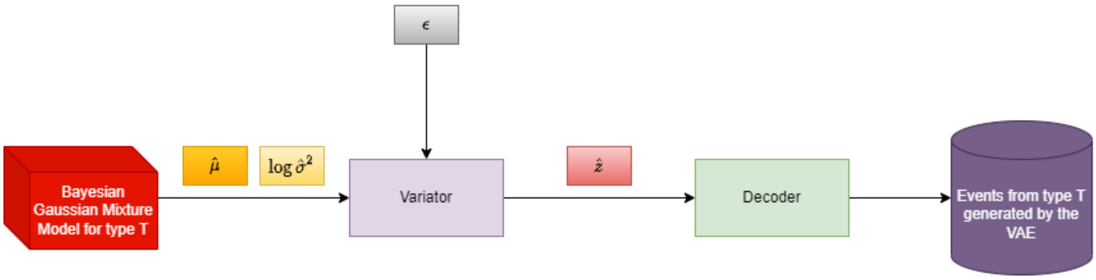

ϵ
Bayesian Gaussian Mixture Model for type T
$\hat{\mu}$
$\log\hat{\sigma}^2$
Variator
$\hat{z}$
Decoder
Events from type T generated by the VAE

Figure 6.7. Sampling from the BGMM of events of type T using the β-VAE.

It must be noted that BGMMs took a considerably higher training time when compared to the previous experiment, mainly due to the fact that the used libraries did not allow to use the GPU hardware for this process.

6.3.1. Training results

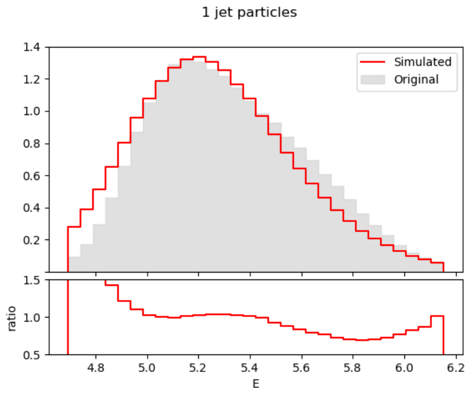


Figure 6.8. Histogram of first jet object parameter E with *ttbar* events ($\beta = 0.01, \gamma = 50$).

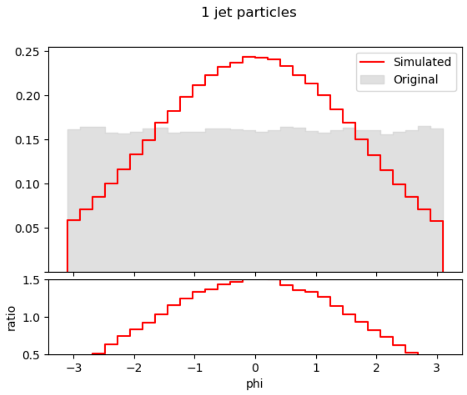


Figure 6.9. Histogram of first jet object parameter ϕ with *ttbar* events ($\beta = 0.01, \gamma = 50$).

The results, in this case, showed similar distributions in most particles (see figure 6.8) when comparing the original data with simulated events. However, there were still some differences that needed to be corrected, and, most importantly, data that followed uniform distributions was represented with normal distributions (see figure 6.9) in the simulation process.

A slight modification was considered: it consisted in taking particle feature values, calculating statistics (mean and standard deviation) to normalize them, scaling particle features before passing them as an input to the model, and then unscaling them after obtaining the model output. However, this modification did not improve the results so it was discarded.

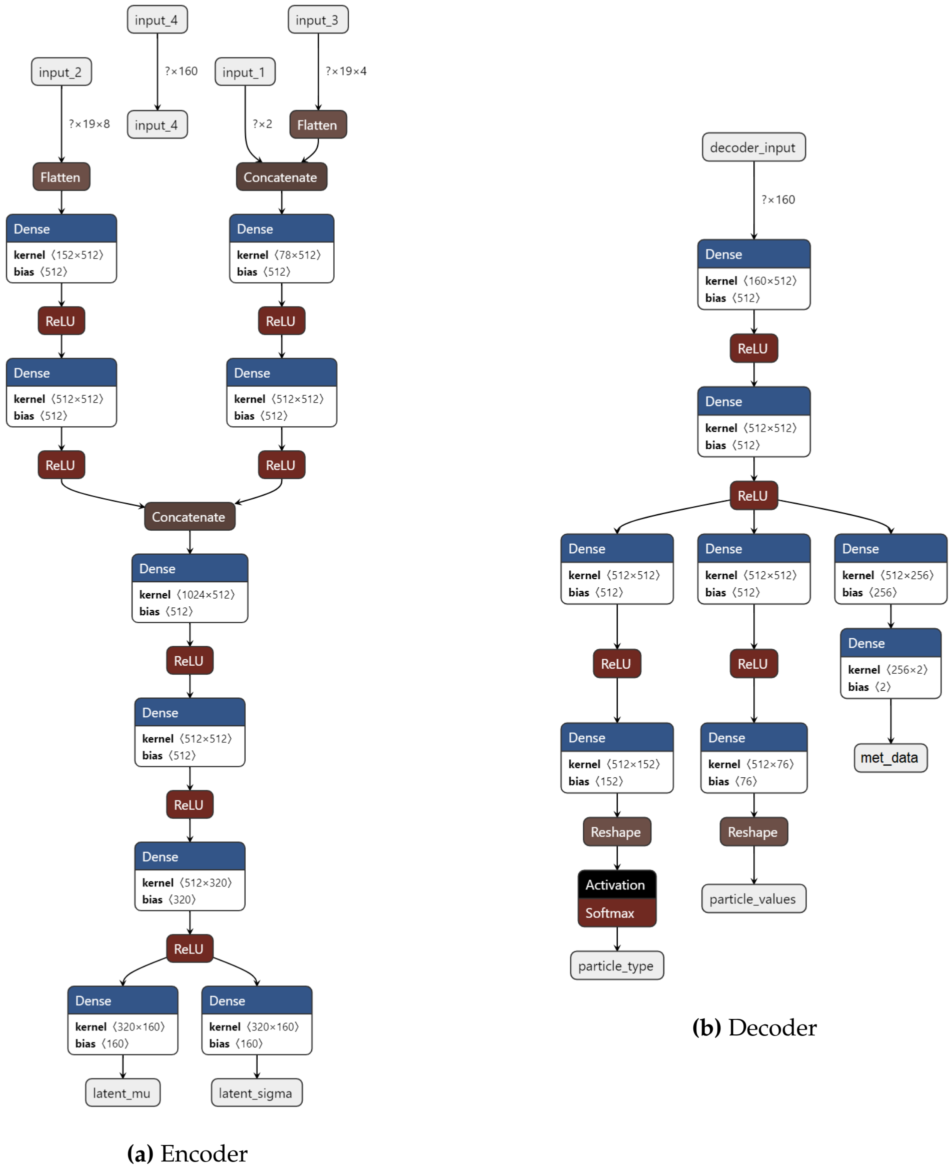
All the results of this experiment for $\beta = 0.01$ and $\gamma = 50$ are available in appendix E.1. With these results so far, we decided that a change in the model architecture was required in order to allow the model to learn uniform distributions accurately, leading to the following experiment.

6.4 Experiment II: Input subdivision

In the previous experiment, data was passed into the model using two inputs plus a third one that represented a random normal distribution to add variability. Considering the obtained results, we decided that it was needed to subdivide input data in more detail.

As we described in section 6.2, every event has two attributes: `MET` and `METphi`, representing the missing transverse energy, and we included them into the data as an special type of particle, mixing it with the rest of particles that formed the event, and adding two particle features with a fixed 0 value. Now, those attributes were no longer considered as a particle and that information was included as a third data input, separated from the rest of the event.

This implies that the maximum amount of particles that an event could have was reduced to 19, and that the number of different types of particle was reduced to 8, so input



(a) Encoder

(b) Decoder

Figure 6.10. VAE architecture scheme in experiment II (variator omitted for simplicity).

dimensions were adjusted accordingly. Now, the model had three data inputs together with the random normal distribution that we mentioned previously.

With this modified architecture (represented in figure 6.10), several new experiments were executed: first, some of them without using BGMMs, and then including that component again. This decision was made because we noticed in experiment I that one of the main needs here was to adjust properly to uniform distributions, an aspect that BGMMs were not able to correct by themselves, and had no apparent reason to occur.

6.4.1. Without BGMMs

The results in this first stage of the experiment without using Bayesian Gaussian Mixture Models provided some interesting simulations. Once again, different values of β were tested, particularly $\beta \in \{0.0, 0.001, 0.01, 0.1, 0.2, 0.5, 0.7, 1\}$, obtaining a wide variety of results. The full result set can be seen in appendix E.2.1.

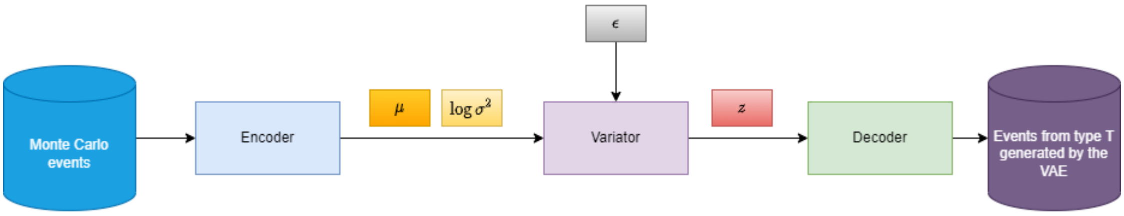
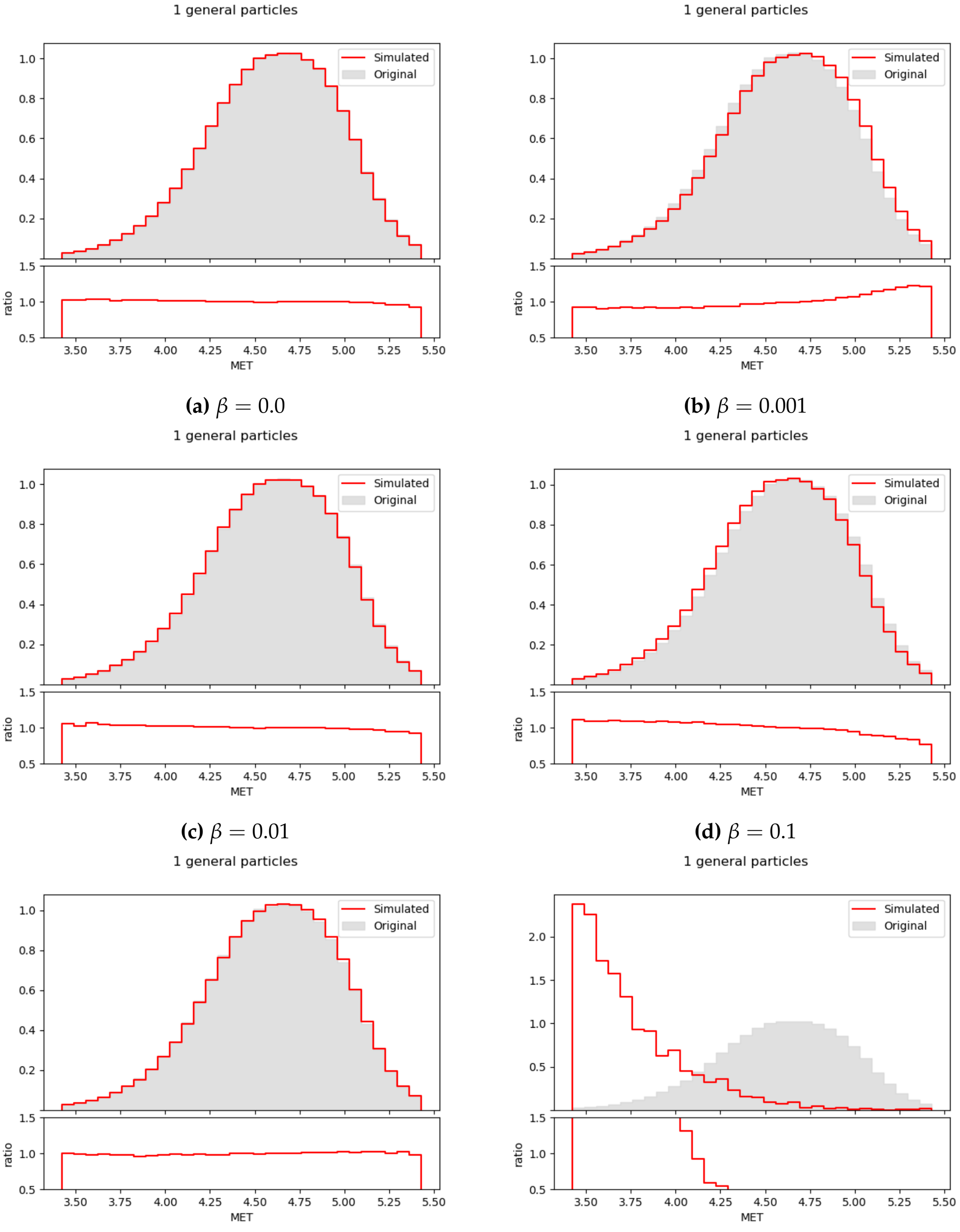


Figure 6.11. Generating events with the β-VAE without using BGMMs.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.1$

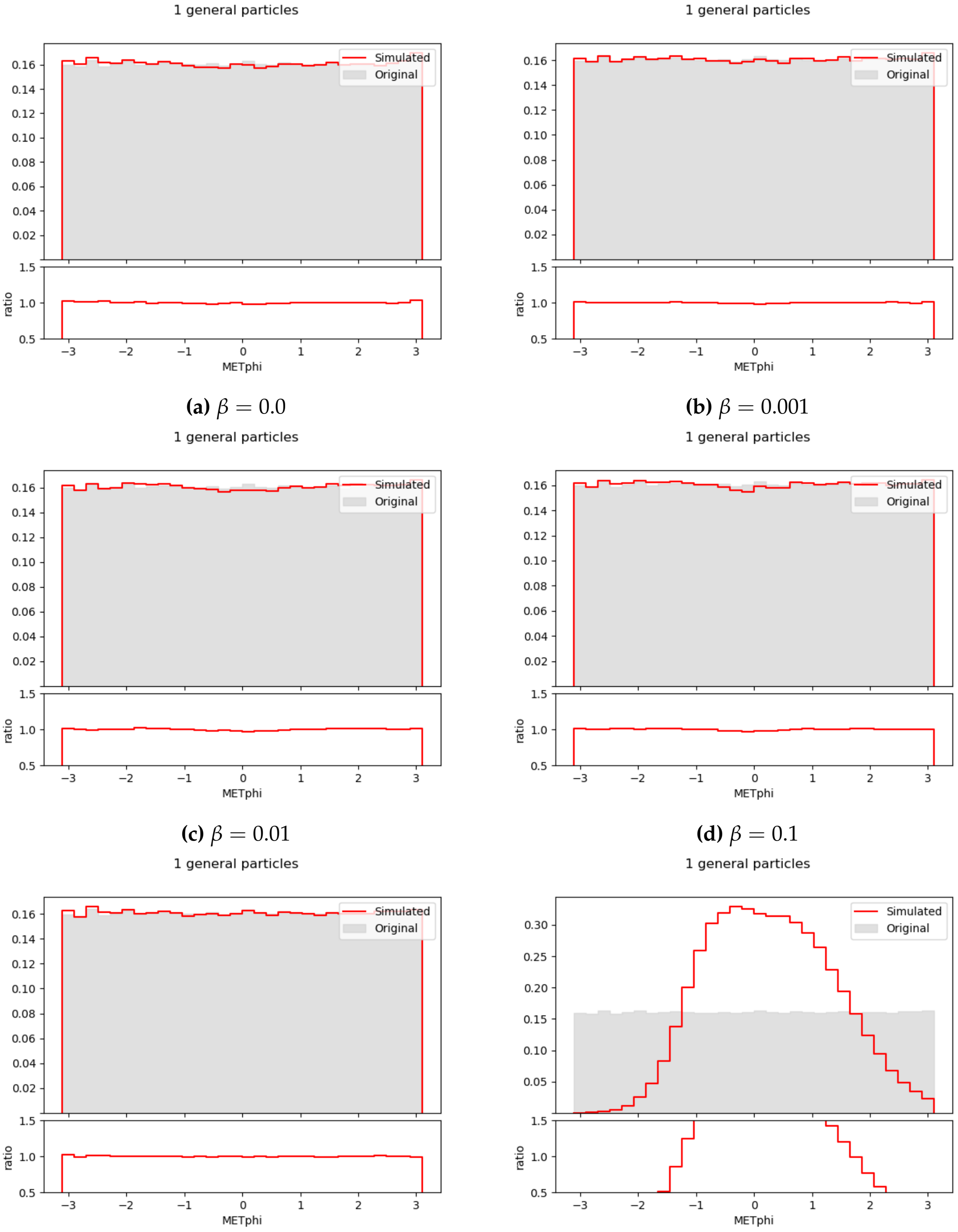
(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.12. Histograms of `MET` event parameter in experiment II with different β values.

First of all, by separating `MET` and `METphi` into a different input we achieved an almost perfect simulation regarding those event features for all tested values of $\beta \leq 0.7$ (even in the uniform distribution that the `METphi` feature follows), as we can observe in figures 6.12 and 6.13.

Here, some small differences in the results (results get worse with $\beta = 0.001$ and then better again with $\beta = 0.01$) could be due to the stochastic nature of the experiments, that are executed using random sampling each time, and therefore repeating an experiment twice with the same configuration could lead to slightly different results.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.1$

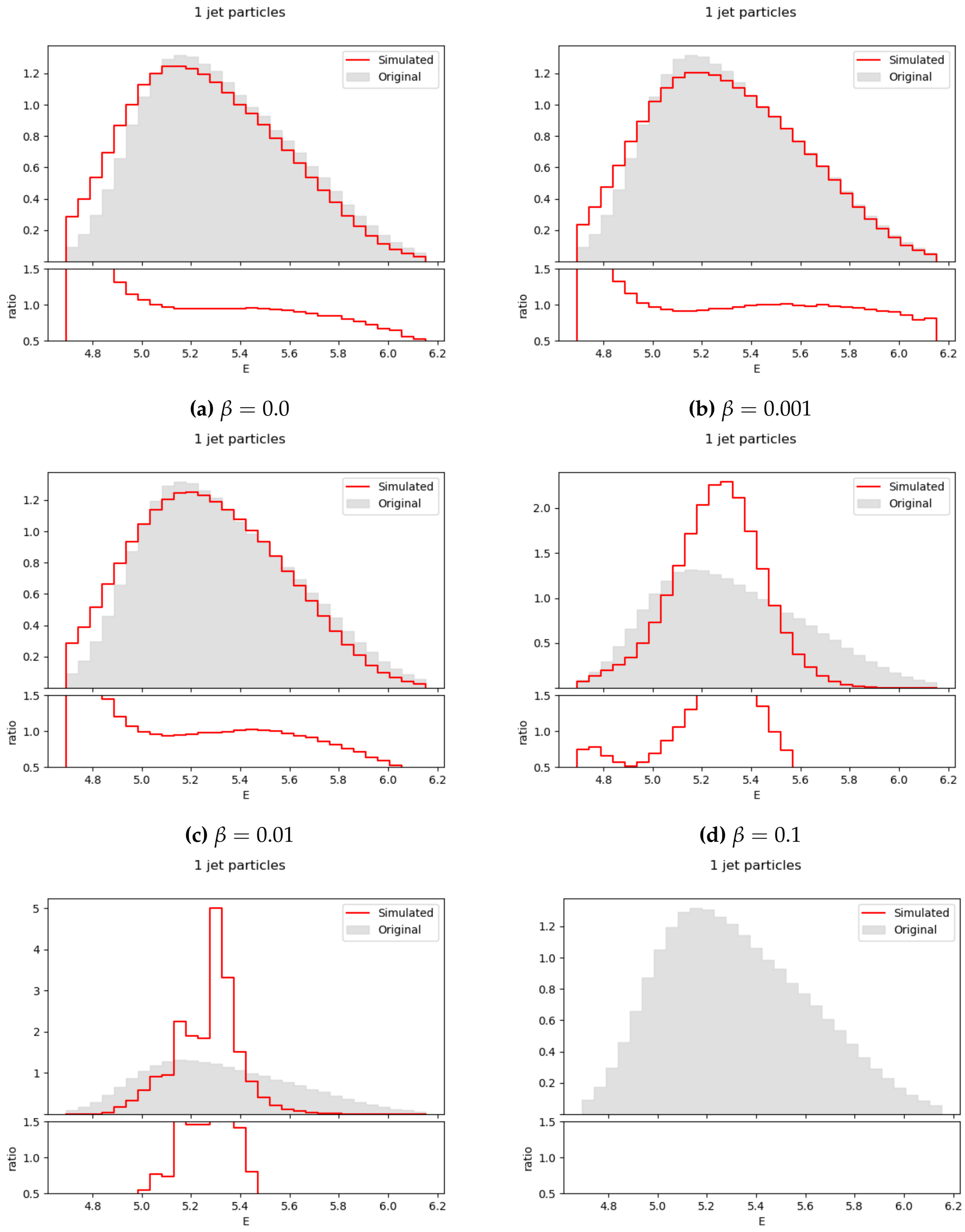
(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.13. Histograms of `METphi` event parameter in experiment II with different β values.

We can observe that, as the value of β increases, the MSE is almost removed from the loss function, effectively rendering the decoder useless when it reaches 1.0, which proved to be very ineffective.

Regarding particle features, we noticed that values of $\beta > 0.01$ generated events that contained particles with distributions that were notably worse than all of the previously shown experiments. The result was drastically different for low values of β, where we finally achieved a great result learning uniform distributions properly after several tries, apart from a great adjustment to the rest of distributions.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

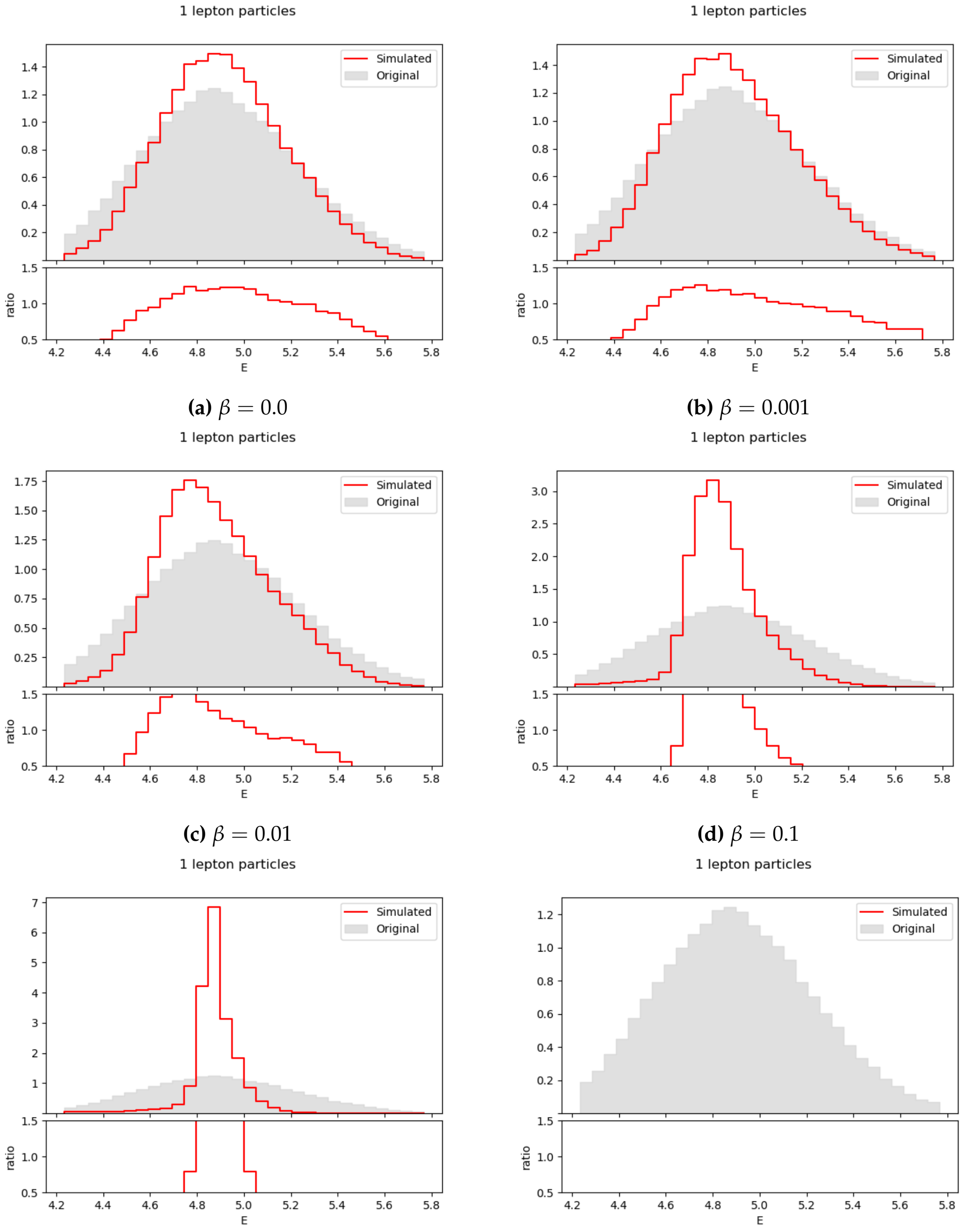
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.14. Histograms of first jet parameter E in experiment II with different β values.

Analyzing the E particle feature of the first jet, we can see a great fit of the data when compared to the original MC events in the range of $\beta \in [0.0; 0.01]$. However, as β increases, the values of the generated events focus on the central part of the Gaussian distribution. Finally, when $\beta = 1.0$, the values are too different from the original MC events to be represented together.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

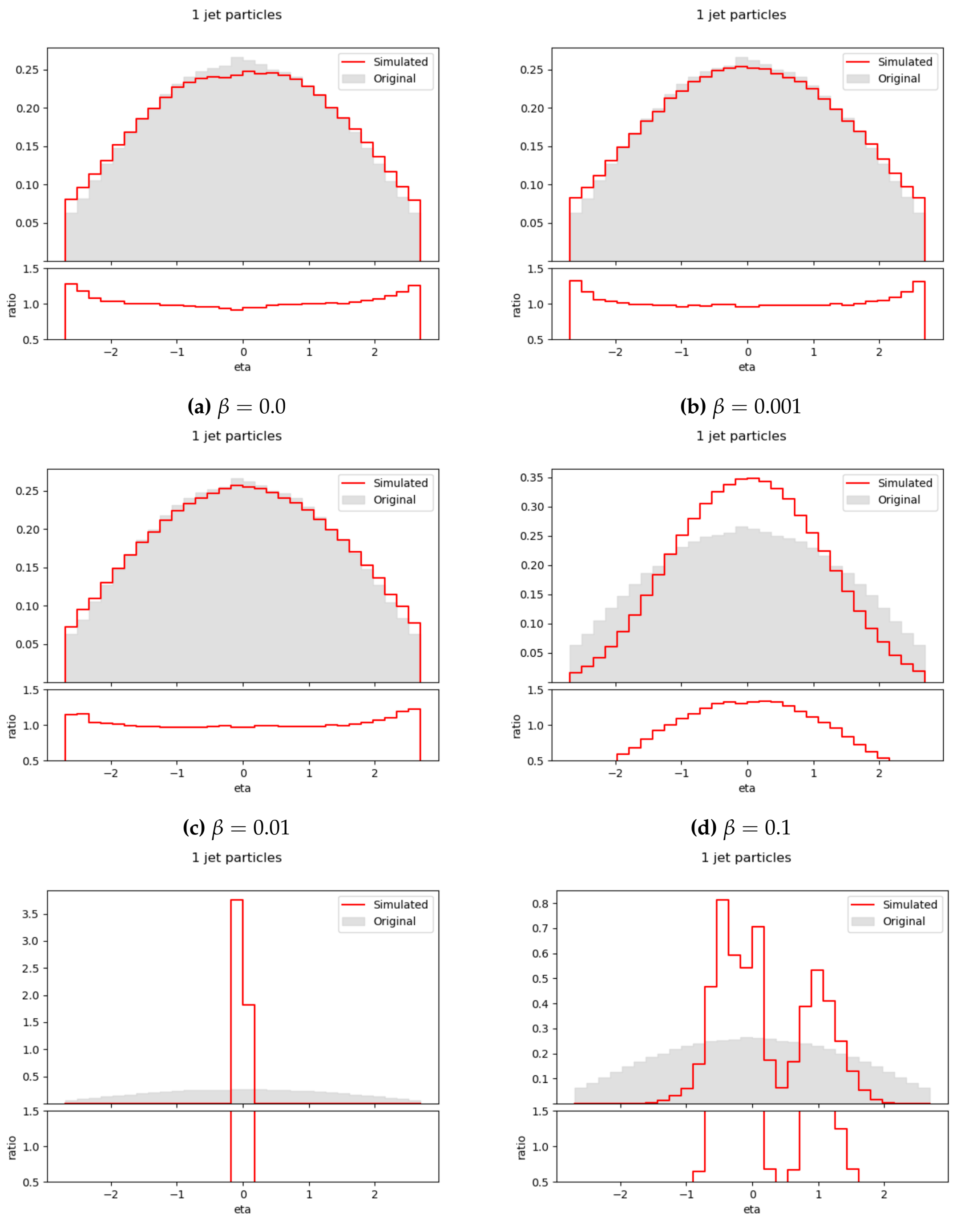
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.15. Histograms of first lepton parameter E in experiment II with different β values.

Observing the E particle feature of the first lepton, we can see a similar behaviour to the one of the first jet: data fits properly when compared to the original MC events if $\beta \in [0.0; 0.01]$, but when it increases, the values of the generated events focus on the central part of the Gaussian distribution, reaching a value where data is too different to be represented together.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

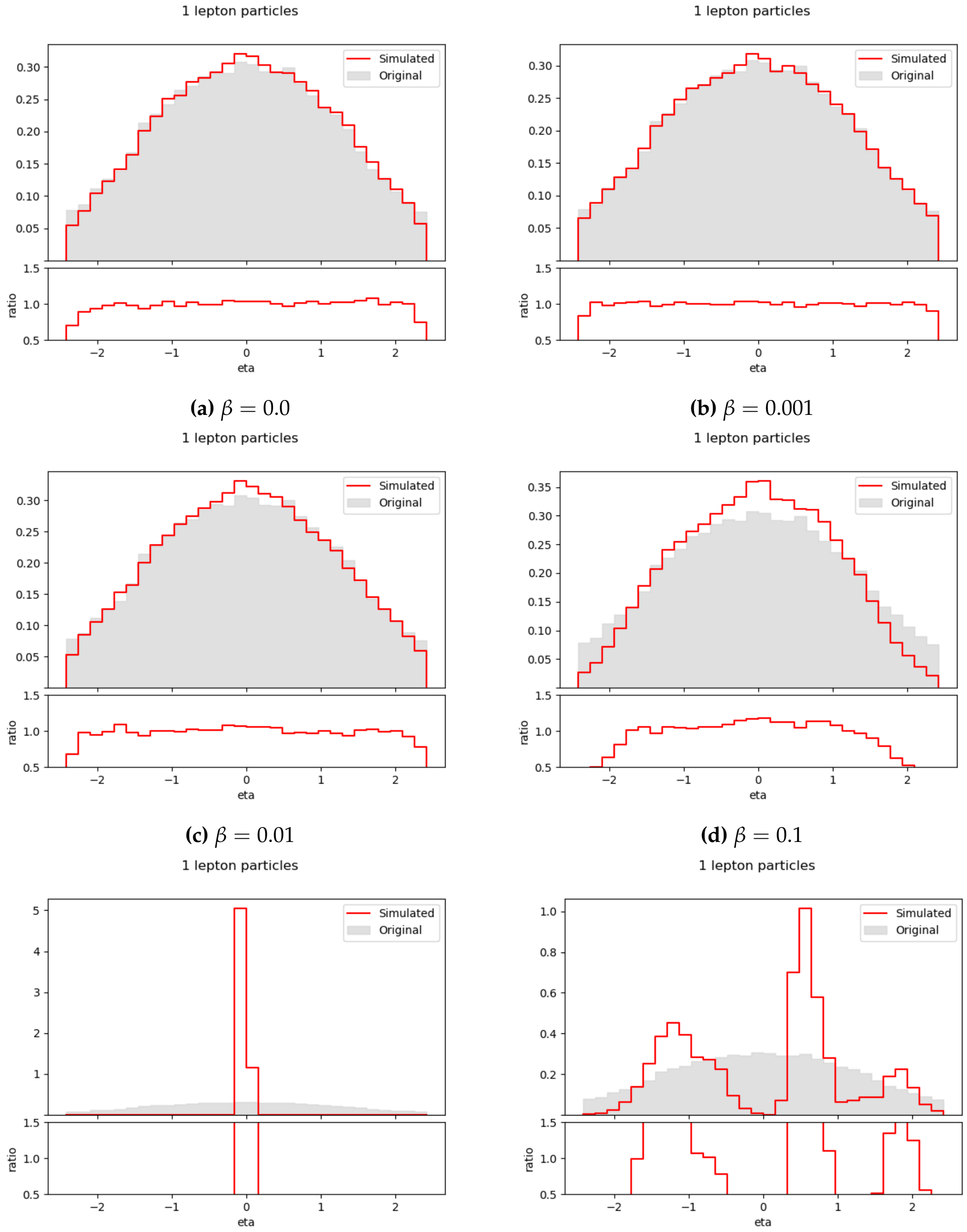
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.16. Histograms of first jet parameter η in experiment II with different β values.

Here, in the η particle feature of the first jet, we can see a better fit in the range of $\beta \in [0.0; 0.01]$, and the results for $\beta = 0.1$ are not bad either. However, when $\beta > 0.1$, the values of the generated events focus heavily on the central part of the Gaussian distribution. Finally, when $\beta = 1.0$, the values seem to describe two Gaussian distributions mixed together, that do not match the original data distribution.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

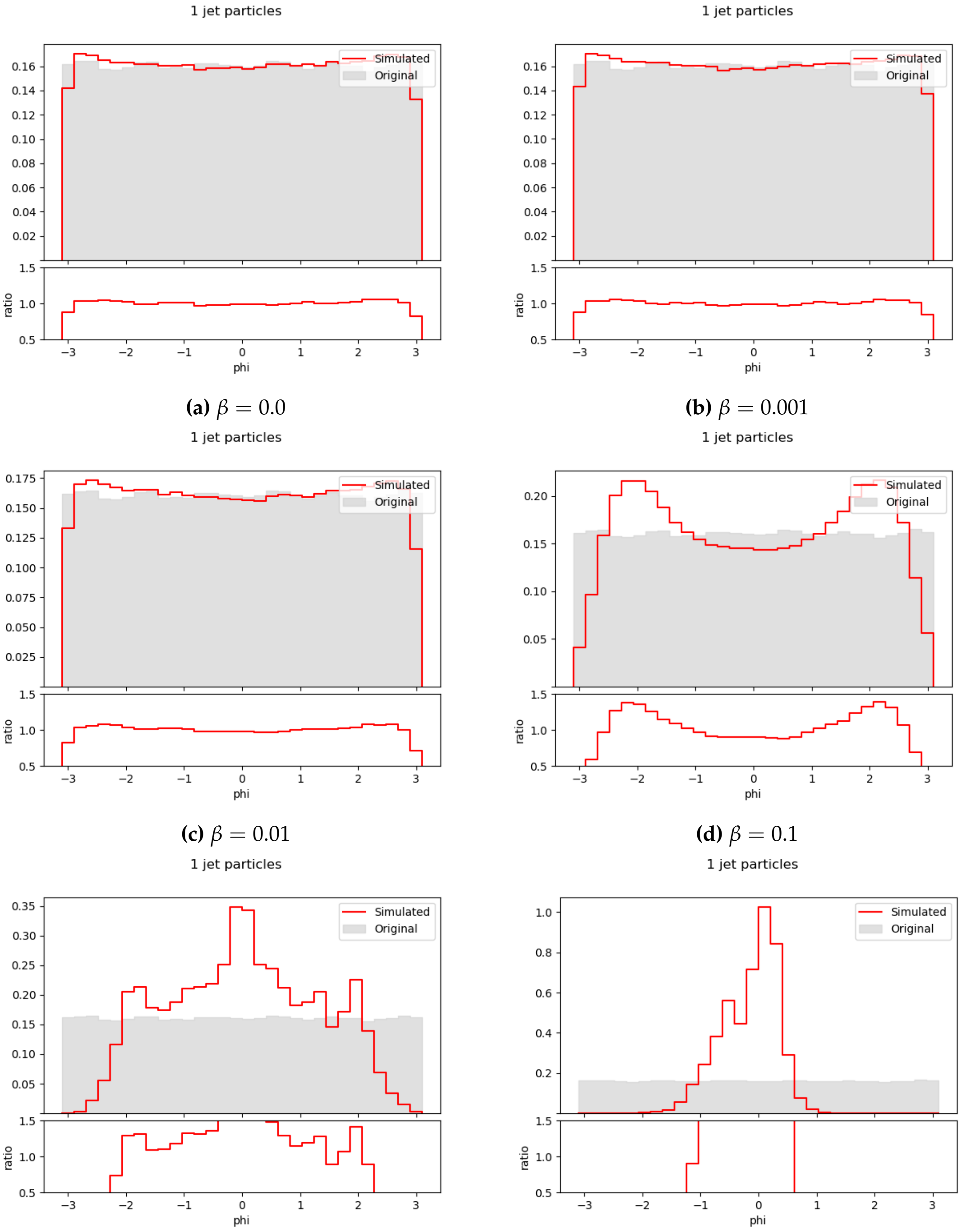
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.17. Histograms of first lepton parameter η in experiment II with different β values.

The η particle feature of the first lepton achieves to model the original MC data in the range of $\beta \in [0.0; 0.1]$. However, when $\beta > 0.1$, the values of the generated events get worse drastically, focusing heavily on the central part of the Gaussian distribution. Finally, when $\beta = 1.0$, the values seem to describe three Gaussian distributions mixed together, that are very different from the original data distribution.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.1$

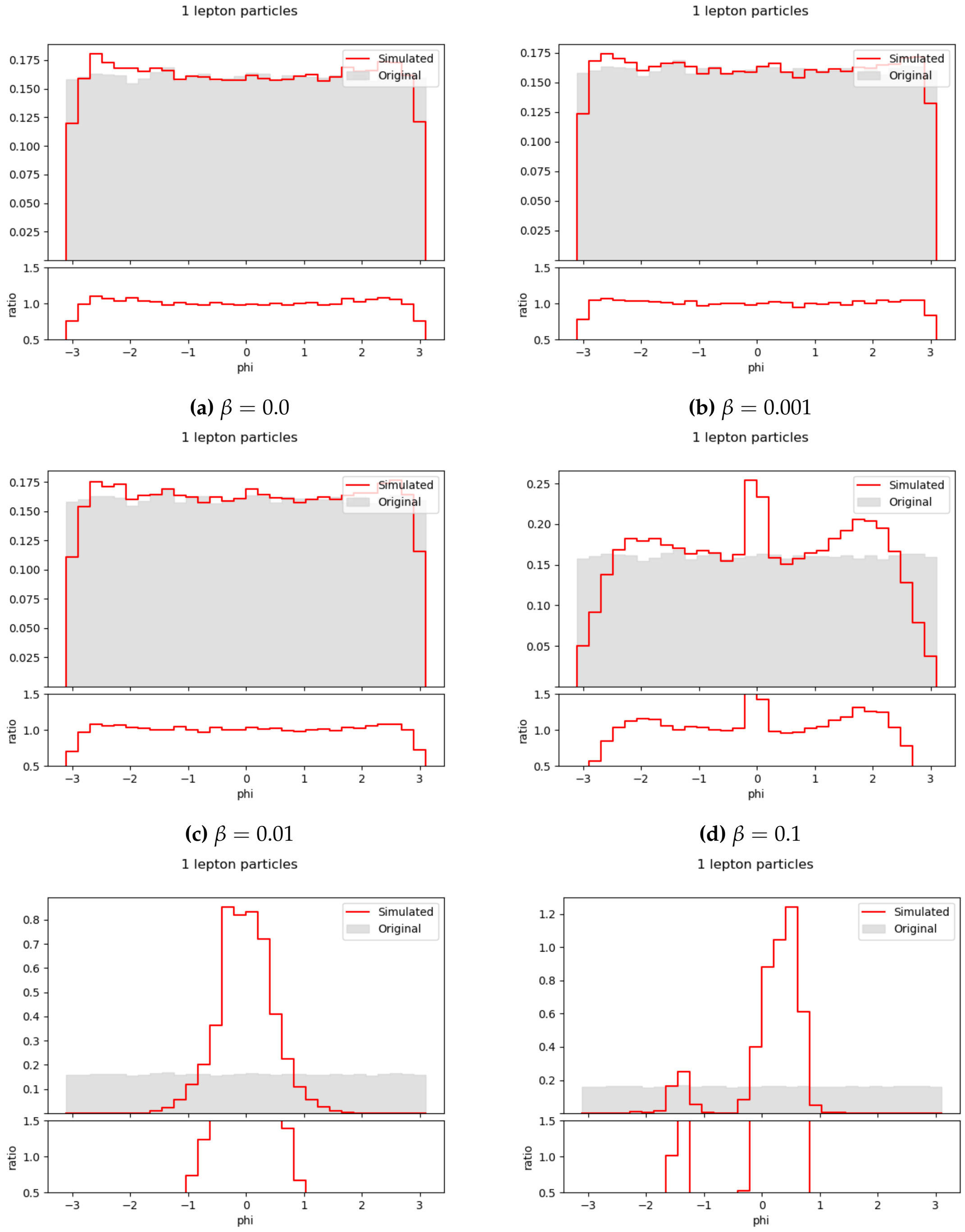
(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.18. Histograms of first jet parameter ϕ in experiment II with different β values.

The ϕ particle feature of the first jet is our first result with a uniform distribution in the original MC data (known as *ground truth*). Analyzing the generated events, we can see a great fit of the data that models the uniform distribution properly when compared to the original MC events in the range of $\beta \in [0.0; 0.01]$. As β increases, the values of the generated events first seem two normal distributions mixed together. Finally, when $\beta = 1.0$, the values are a normal distribution that is not related with the original values at all.

It is worth mentioning that ϕ and `METphi` data are present for validation purposes, as they should resemble a uniform distribution for the results to be considered correct. However, its physics meaning is not of high relevance.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

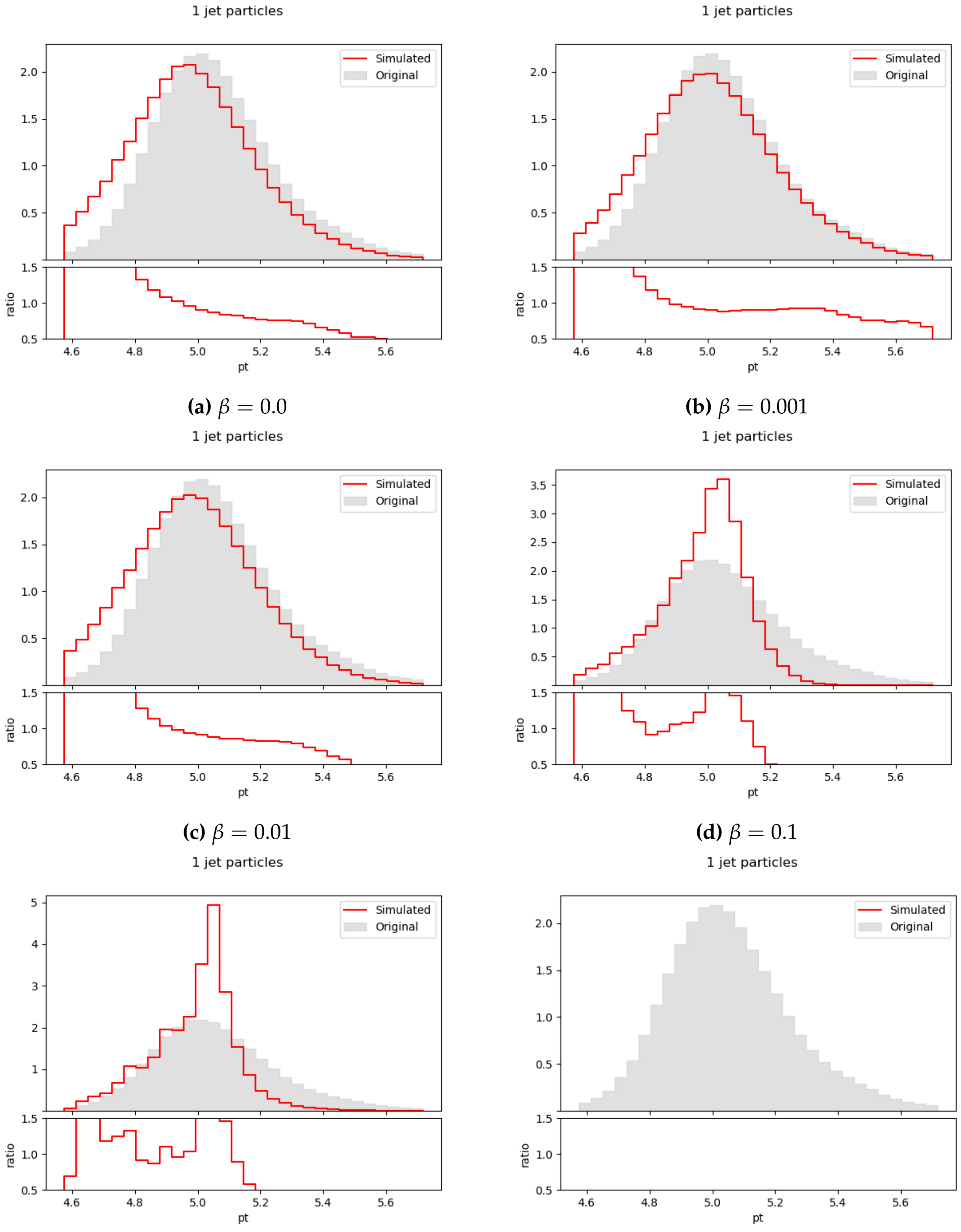
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.19. Histograms of first lepton parameter ϕ in experiment II with different β values.

The ϕ particle feature of the first lepton also achieves a great fit of the data that models the uniform distribution properly in the range of $\beta \in [0.0; 0.01]$. This time, as β increases, the values of the generated events get worse faster than in the case of jets: they quickly become many Gaussian distributions mixed together, and when $\beta \geq 0.5$, the values are again a Gaussian distribution.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

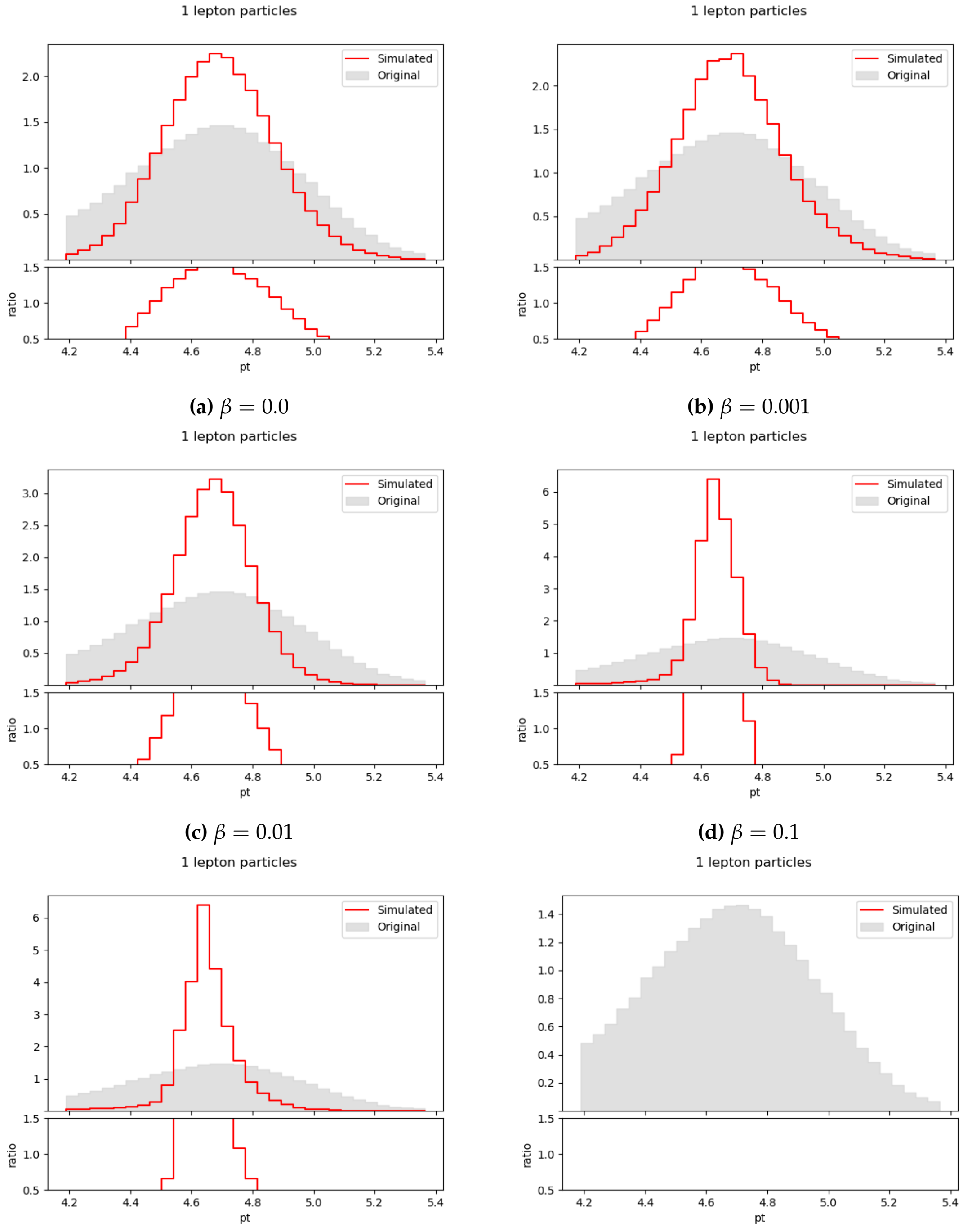
(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.20. Histograms of first jet parameter p_T in experiment II with different β values.

Analyzing the p_T particle feature of the first jet, we can see that data fits properly when compared to the original MC events when $\beta \in [0.0; 0.01]$ (even though there is a small shift to the left in the distribution). However, when it increases, the values of the generated data that were in the right tail of the distribution seem to move to the central part. Finally, data reaches values where it is too different to be represented together with the original MC simulated events.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.1$

(e) $\beta = 0.5$

(f) $\beta = 1.0$

Figure 6.21. Histograms of first lepton parameter p_T in experiment II with different β values.

Finally, observing the p_T particle feature of the first lepton, we see that from the lowest β value, the values from the right and left tails of the *ground truth* events distribution are located in the central part of the generated events distribution. This effect is more evident as β increases, reaching a point where data is too different to be represented together.

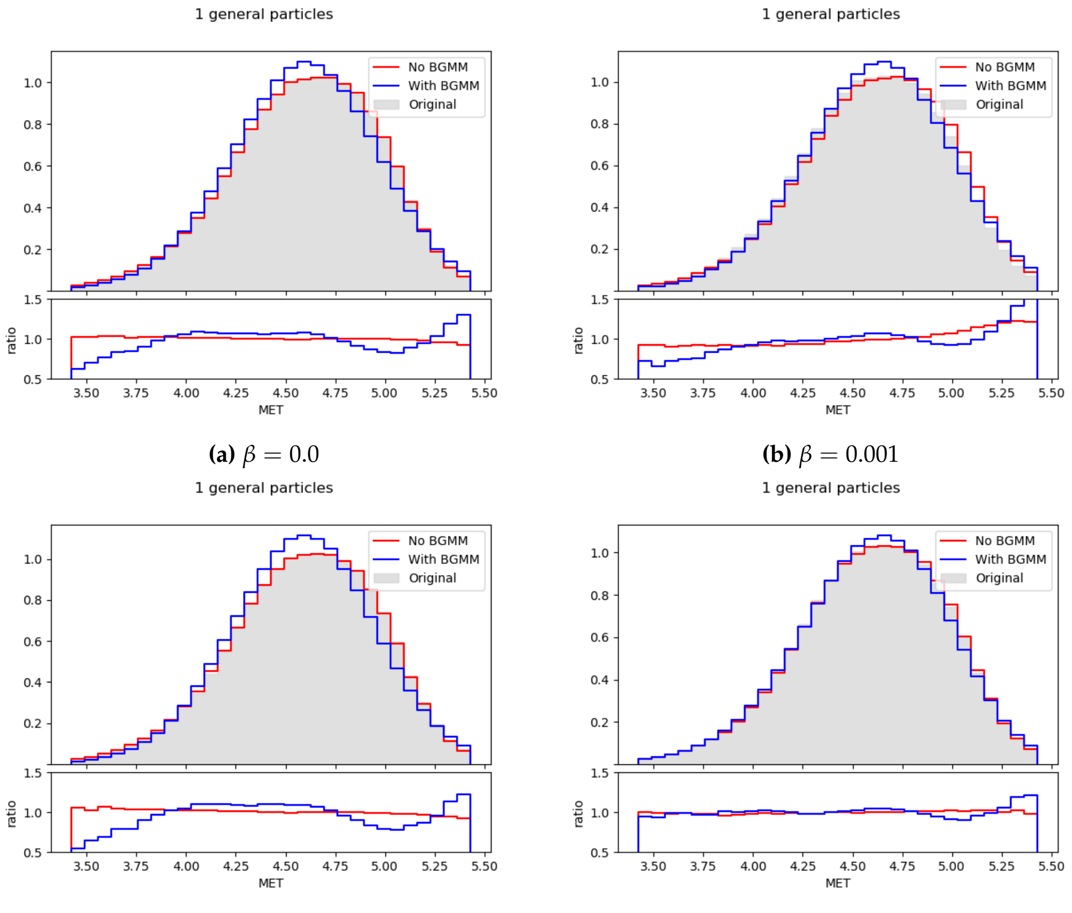
Considering the previous analysis, we conclude that in this case the weight assigned to the Kullback-Leibler divergence in the loss function of our β-VAE must be significantly lower when compared to the one of the MSE for the results to be acceptable.

Here, the decoder received as input the latent representation of encoded events and a value ϵ, sampled from a normal distribution, to add variability and generate new events. Another option was to provide the decoder with randomly generated values for the latent representation instead of values from already existing (encoded) events. This last option was discarded as it obtained very low quality results due to a lack of information given to the decoder, even though this component was already trained.

6.4.2. With BGMMs

After analyzing the promising results obtained by the variant of this model without Bayesian Gaussian Mixture Models, we decided to include that component again to check whether the obtained results could improve further.

We used the same values of β than in the previous approach, and $\gamma = 100$ with full covariance matrices. A comparison of the most relevant results is presented here. However, the full result set can be seen in appendix E.2.2.



(a) $\beta = 0.0$

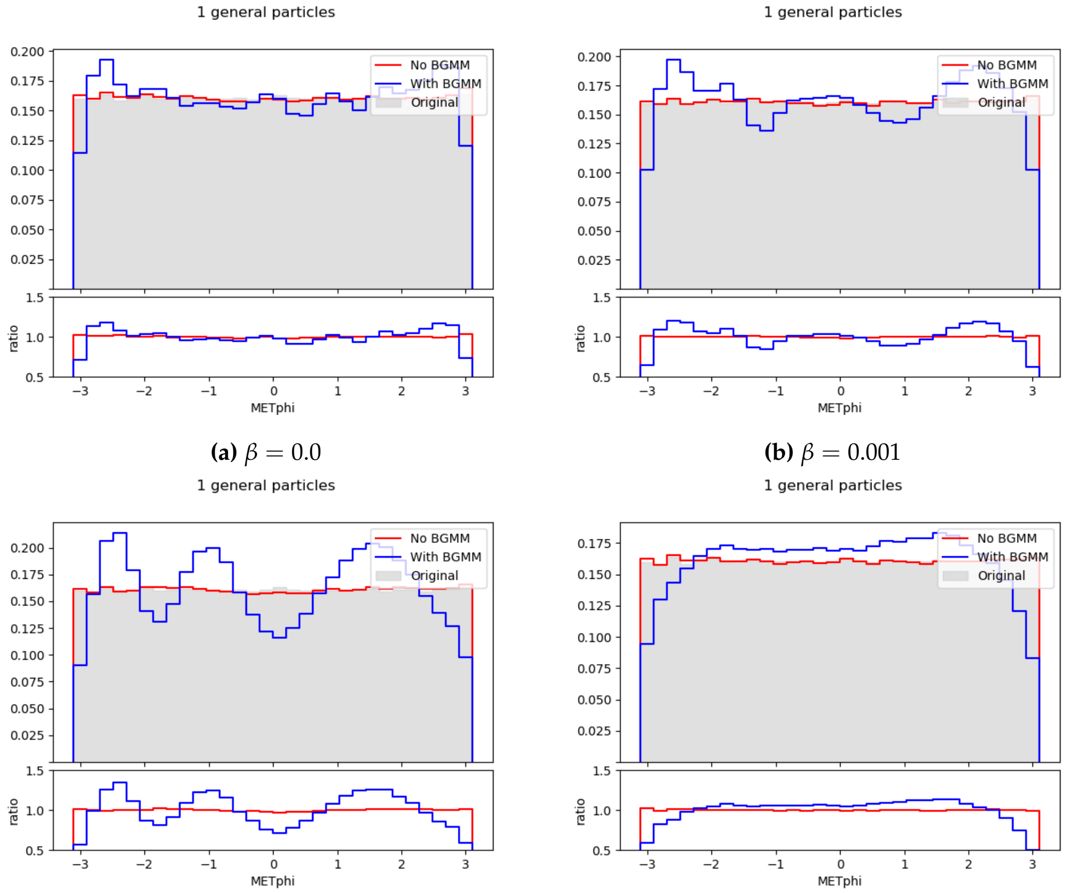
(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.22. Comparison of MET event parameter with and without BGMMs ($\gamma = 100$).

Observing the MET values of all the events, we can see that both the BGMM and non-BGMM approaches model the data accurately (considering a certain margin of error). Nevertheless, in the ratio histogram we see the non-BGMM model performing better.

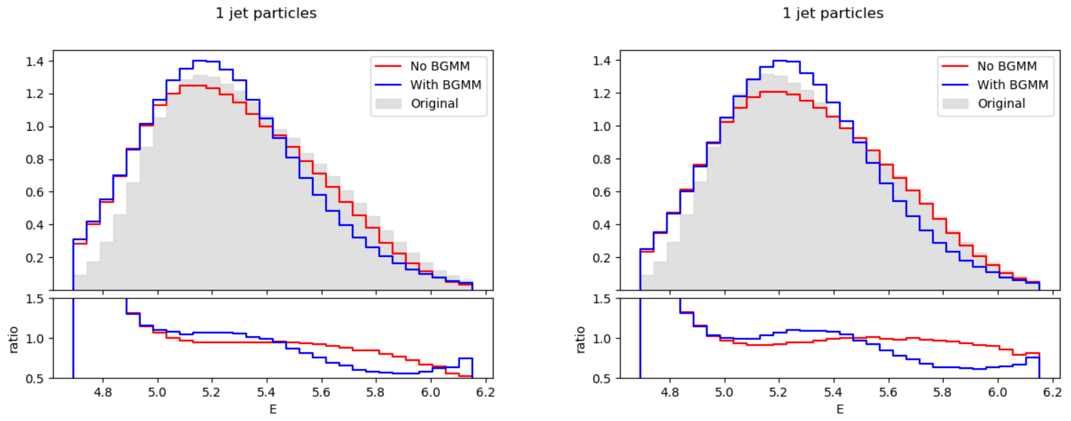


(a) $\beta = 0.0$ **(b)** $\beta = 0.001$

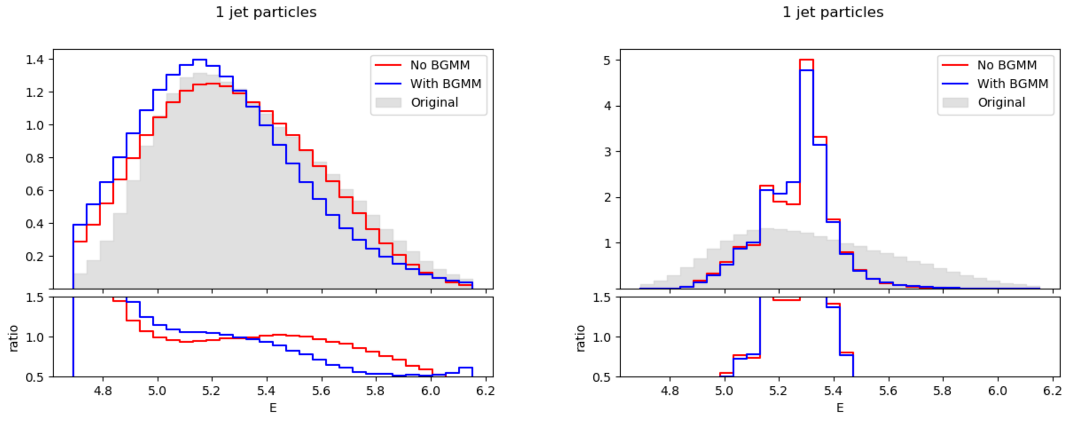
(c) $\beta = 0.01$ **(d)** $\beta = 0.5$

Figure 6.23. Comparison of `METphi` event parameter with and without BGMMs ($\gamma = 100$).

When comparing the `METphi` values, here the result varies significantly, as the great fit to the uniform distribution that was achieved in the previous non-BGMM model is lost. The results when $\beta = 0.0$ still maintain a great ratio with respect to the original MC data. However, as β increases, the previously obtained almost-uniform distribution turns into several normal distributions mixed together.



(a) $\beta = 0.0$ **(b)** $\beta = 0.001$

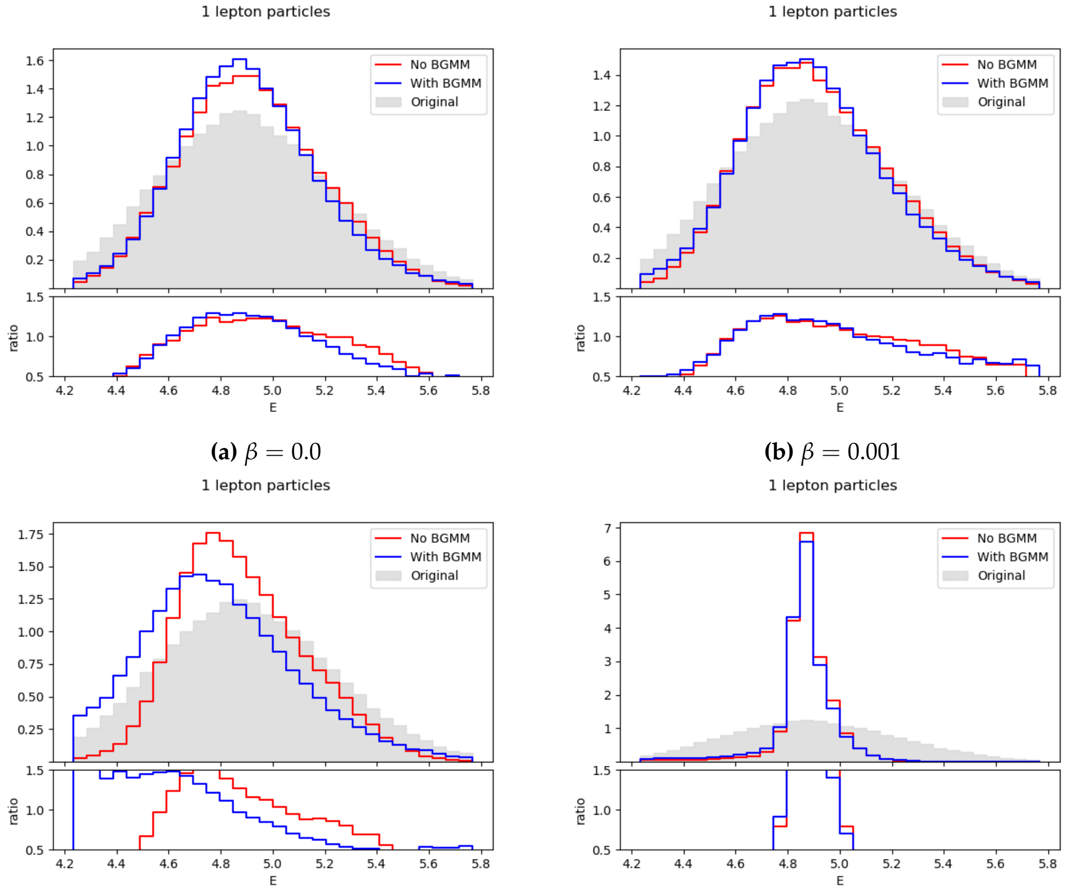


(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.24. Comparison of first jet particle feature E with and without BGMMs ($\gamma = 100$).

The E particle feature of the first jet concentrates a few values of the right tail of the original distribution in the central and left part, obtaining worse events than the non-BGMM variant. When $\beta > 0.01$, the data groups in the middle of the original distribution.



(a) $\beta = 0.0$

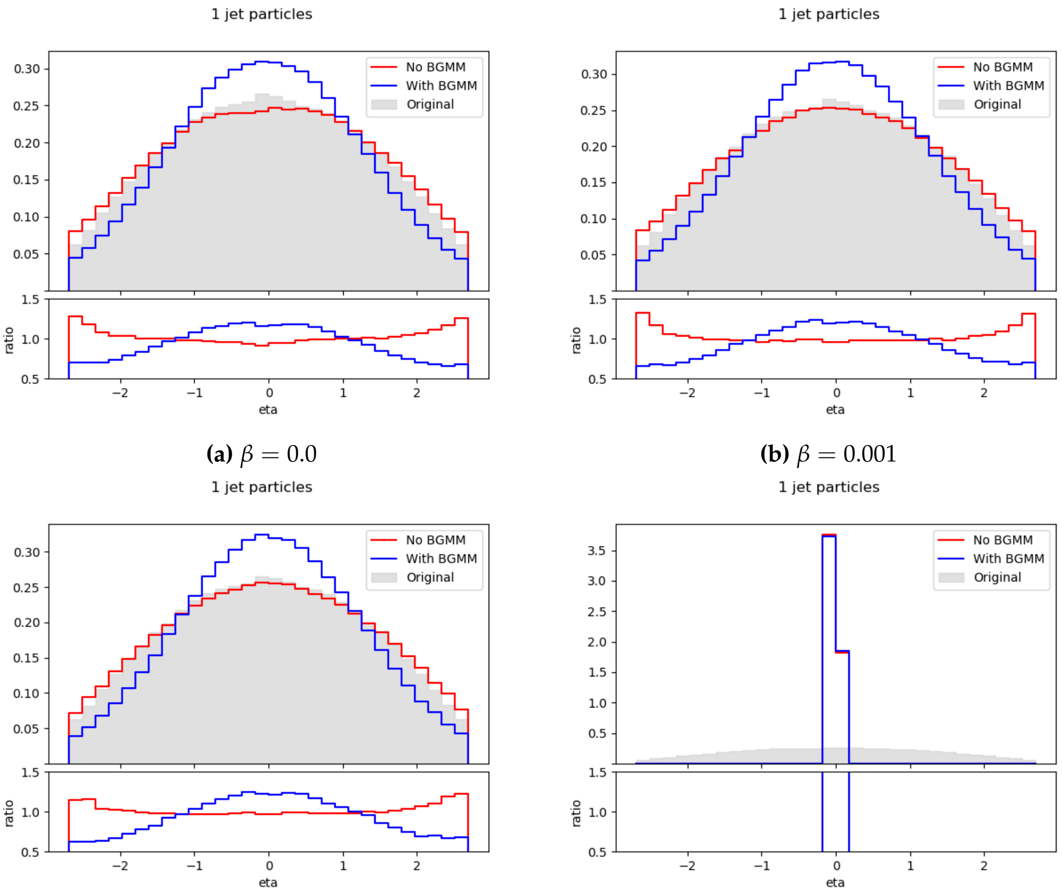
(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.25. Comparison of first lepton particle feature E with and without BGMMs ($\gamma = 100$).

In the E particle feature of the first lepton we find a similar distribution in the BGMM and non-BGMM variant when $\beta \in [0.0; 0.001]$. When $\beta = 0.01$, we obtain a more 'flat' distribution when compared with the non-BGMM variant, that still does not adjust properly to the *ground truth* data. When $\beta > 0.01$, the same centering effect that happened with the first jet occurs.



(a) $\beta = 0.0$

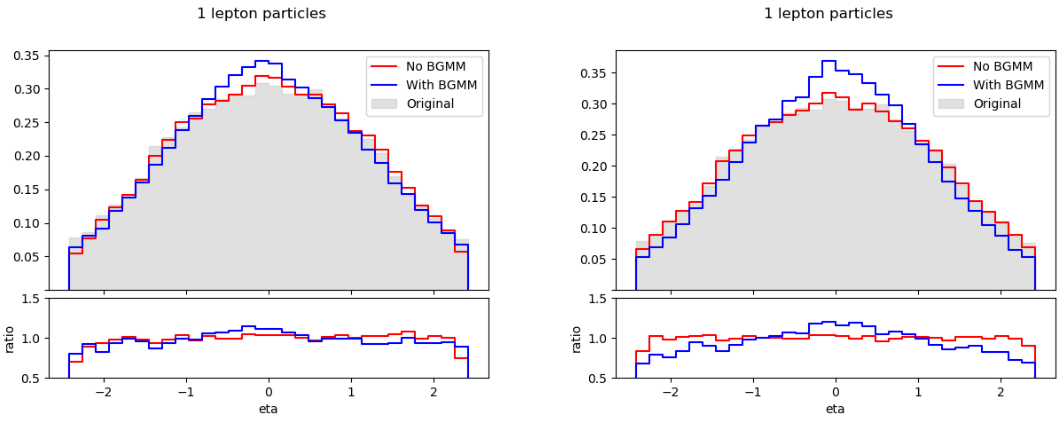
(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

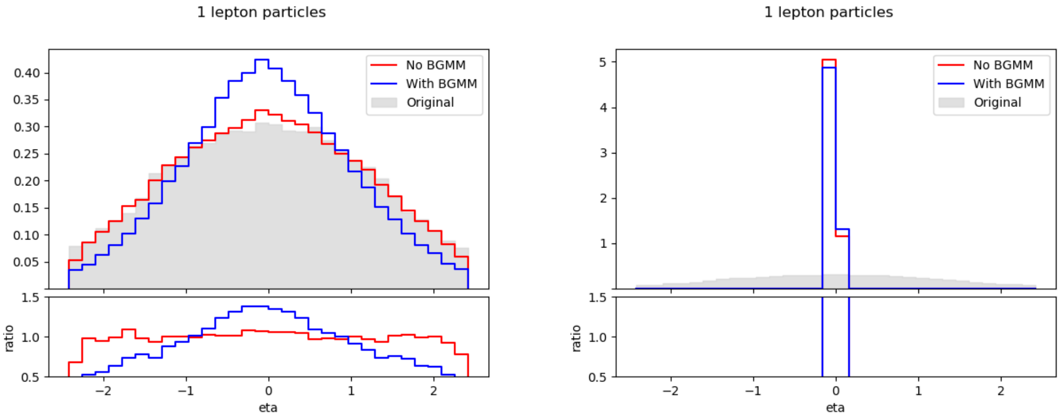
Figure 6.26. Comparison of first jet particle feature η with and without BGMMs ($\gamma = 100$).

Comparing the η particle feature of the first jet, we lose the great fit that we obtained in the non-BGMM variant, resulting in a more spiky distribution that concentrates some values from the tails of the *ground truth* distribution in the center of the generated events. When $\beta > 0.01$, almost all values are in the two middle bars of the histogram, all grouped without a clear distribution.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

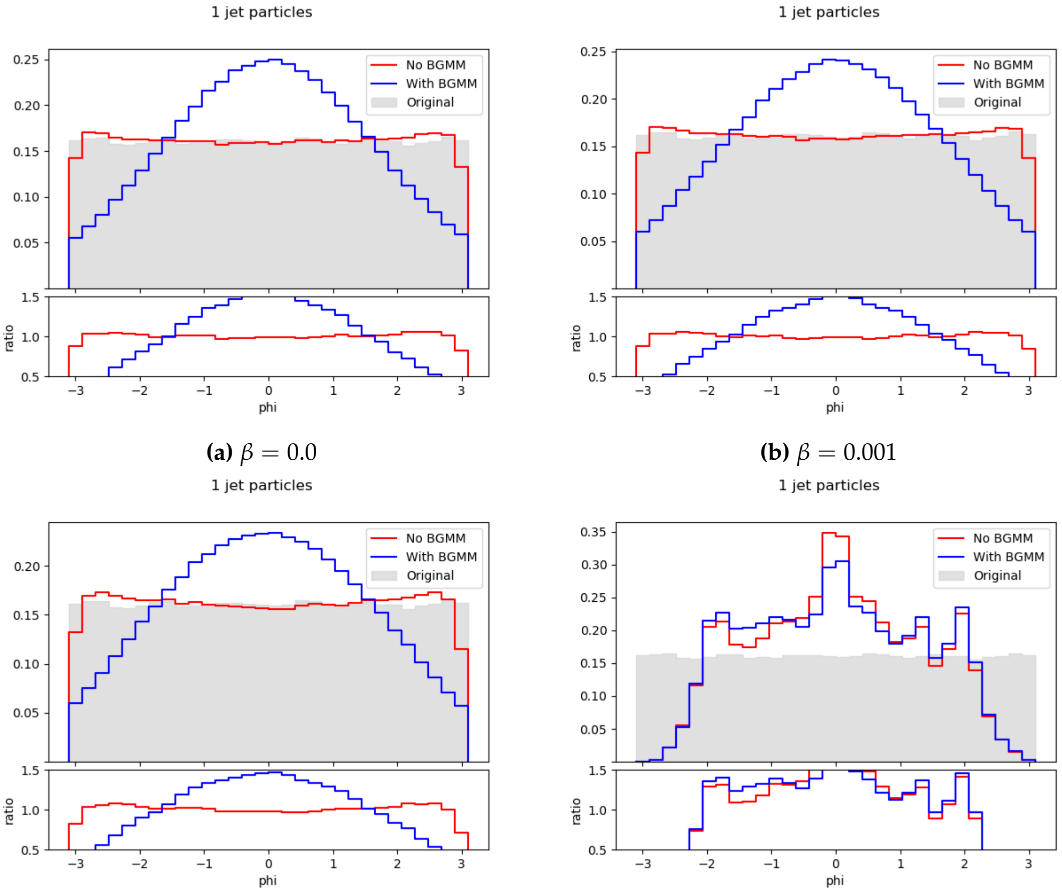


(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.27. Comparison of first lepton particle feature η with and without BGMMs ($\gamma = 100$).

The η particle feature of the first lepton behaves similarly than what was observed for the first jet. Nevertheless, in this case when $\beta \in [0.0; 0.001]$, the results are equivalent to those obtained in the non-BGMM variant.



(a) $\beta = 0.0$

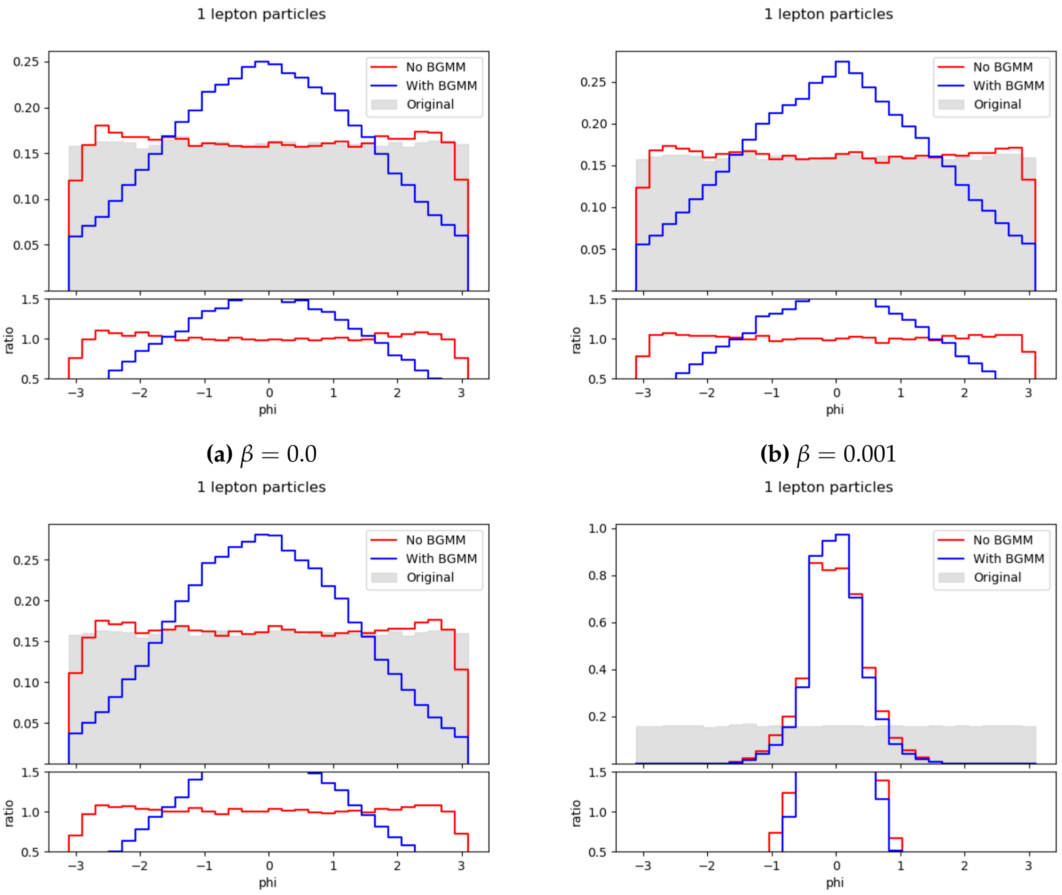
(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.28. Comparison of first jet particle feature ϕ with and without BGMMs ($\gamma = 100$).

Regarding the ϕ particle feature of the first jet, the accurate fit to the uniform distribution achieved in the previous non-BGMM model is also lost, resembling now to a normal distribution. This also happened (but more softly) in the `METphi` case. The effect is present already in the smallest β values.



(a) $\beta = 0.0$

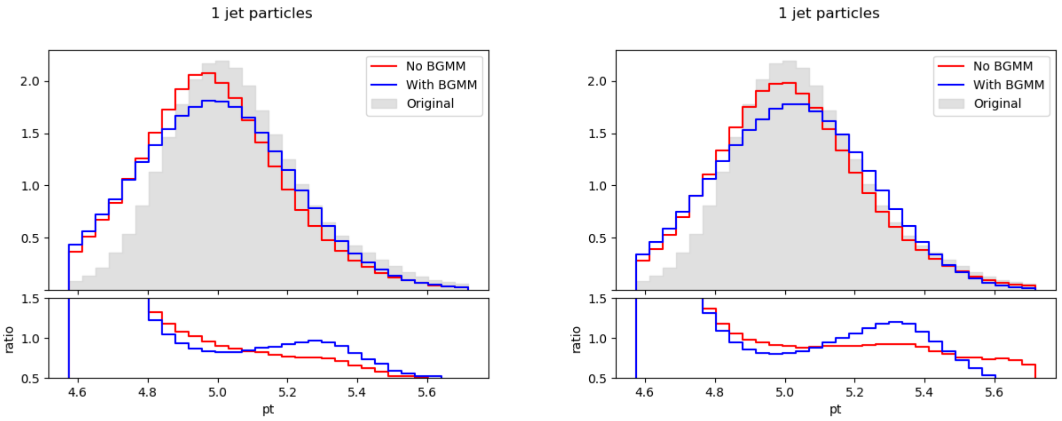
(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

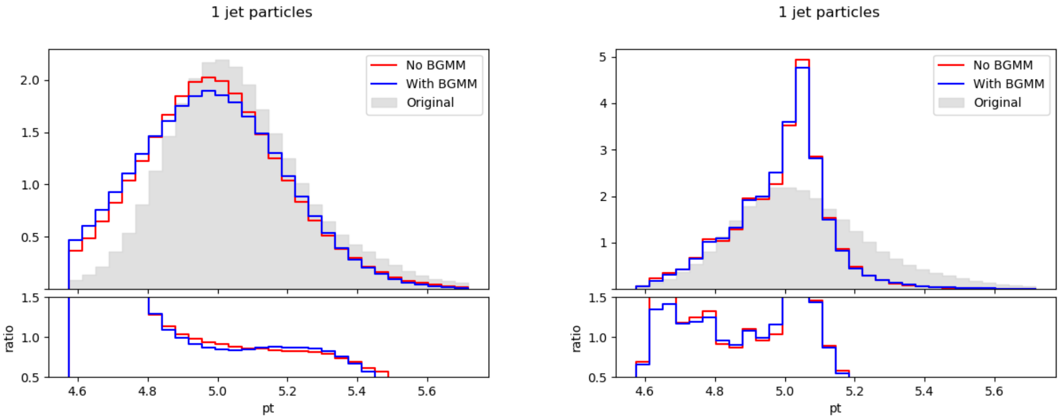
Figure 6.29. Comparison of first lepton particle feature ϕ with and without BGMMs ($\gamma = 100$).

In the ϕ particle feature of the first lepton, we find a resemblant behaviour to that of the same feature of the first jet. The difference is that now, when $\beta > 0.01$, the generated events define a Gaussian distribution that is very dissimilar to the uniform distribution of the original MC events.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

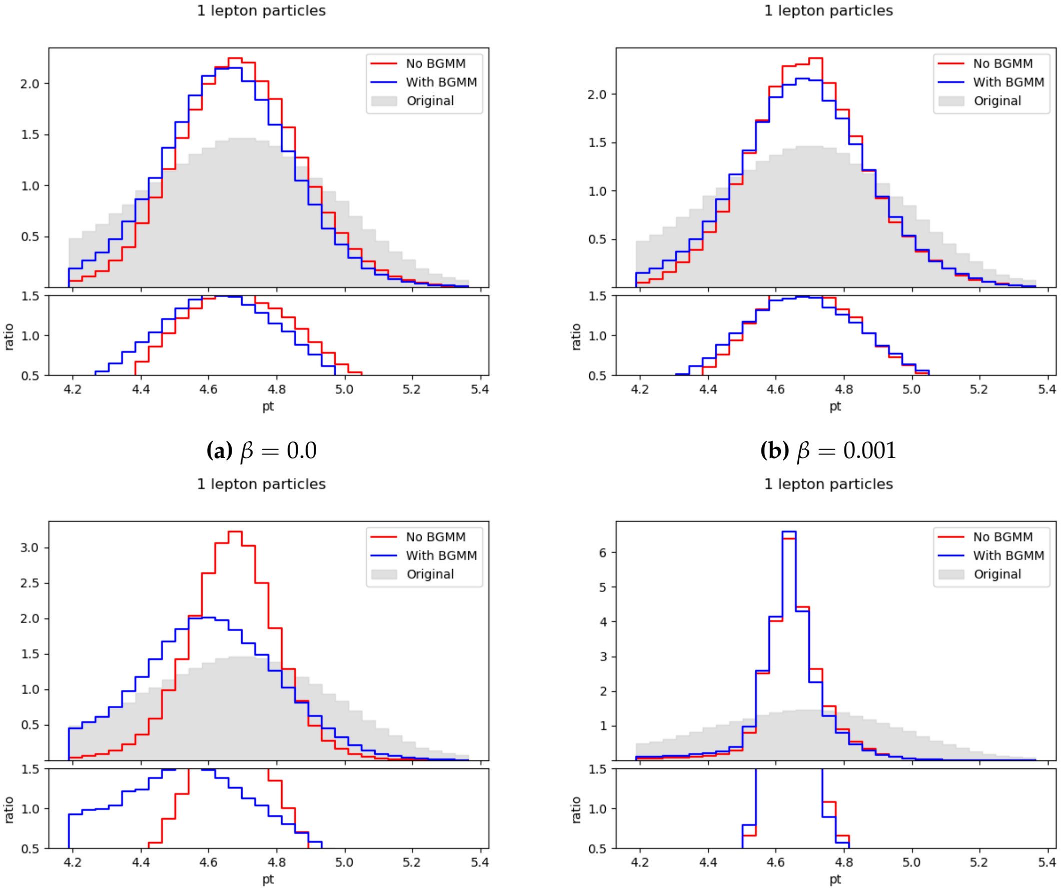


(a) $\beta = 0.01$

(b) $\beta = 0.5$

Figure 6.31. Comparison of first jet particle feature p_T with and without BGMMs ($\gamma = 100$).

The p_T particle feature of the first jet presents a similar distribution to the non-BGMM variant in all cases: some of the data in the middle of the original distribution is in the left tail of the generated events distribution when $\beta \in [0.0; 0.01]$. Finally, when $\beta \geq 0.5$, we notice some data in the right tail of the original distribution now in the central part of the generated data.



(a) $\beta = 0.0$

(b) $\beta = 0.001$

(c) $\beta = 0.01$

(d) $\beta = 0.5$

Figure 6.32. Comparison of first lepton particle feature p_T with and without BGMMs ($\gamma = 100$).

The p_T particle feature of the first lepton behaves similar to the non-BGMM variant when $\beta \leq 0.001$. In $\beta = 0.01$, we notice a more similar distribution to the original data than the one obtained in the non-BGMM model (with some values in the right tail of the original distribution moved into the central part of the generated data distribution). Finally, when $\beta \geq 0.5$, the generated events define a distribution that is very different to the original MC events.

All things considered, we conclude that with this model including the BGMM is not beneficial and does not help correcting the inability to adjust to the uniform distributions: a problem that was present in all the previous experiments and is resolved in the non-BGMM variant. The reason for this could be the need to add more components (γ), but additional experimentation regarding this topic was not possible due to a lack of time.

6.5 Experiment III: Introducing the α-VAE

For the last experiment, we used our alternative model of Variational Autoencoder described in section 4.2.2: the α-VAE, first to generate events by itself, and then together with a BGMM. The whole event generation process was again divided in different steps.

In both variants, first our new VAE model is trained focusing in only one kind of event for 30 epochs (fig. 6.33), as we noticed that results did not change much when reducing the number of epochs to that number. This α-VAE has three sub-components, and features some differences when compared to the previously used β-VAE. This design choice allows to connect some VAE components with the BGMM in further steps.

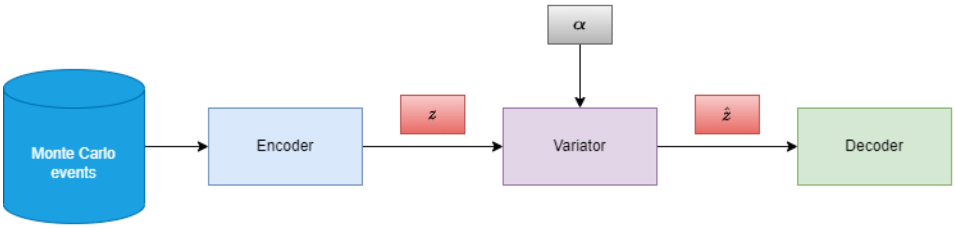


Figure 6.33. Training the α-VAE using one kind of event.

Our implementation takes as a starting point the one depicted in figure 6.10 with some changes to the inputs and outputs of the components, and is structured as follows:

- an **encoder**, which transforms the input data in the format described in section 6.2 into an encoded representation in the latent space z, producing this encoded representation as its unique output.
- a **variator**, that receives the output of the encoder and, together with a normal distribution with $\mu = 0$ and $\sigma = \alpha$, adds Gaussian noise to the latent representation and outputs $\hat{z}$.

  The dimensions of the $\hat{z}$ tensor match the latent dimensionality z. The variator component is usually part of the encoder in most implementations.
- a **decoder**, that transforms the encoded data from the latent space back into its original dimensions. This component is identical to that of the β-VAE.

This VAE used *Categorical Crossentropy* as the loss function for the particle type detection, and the *Mean Squared Error* for learning the different object features (including `MET` and `METphi`).

Once the α-VAE is trained, each variant of the experiment follows a different procedure. The selected values of α to run the experiments were $\alpha \in \{0.1, 0.2, 0.3\}$.

6.5.1. Without BGMMs

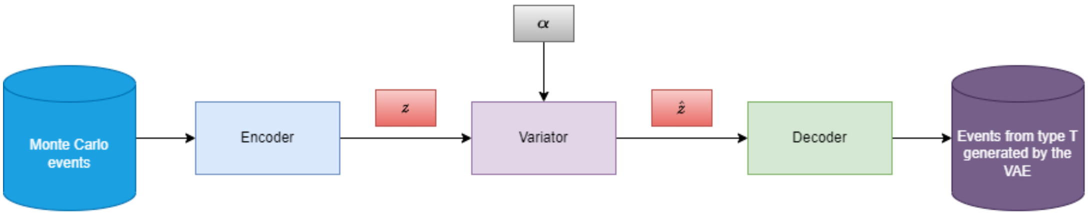
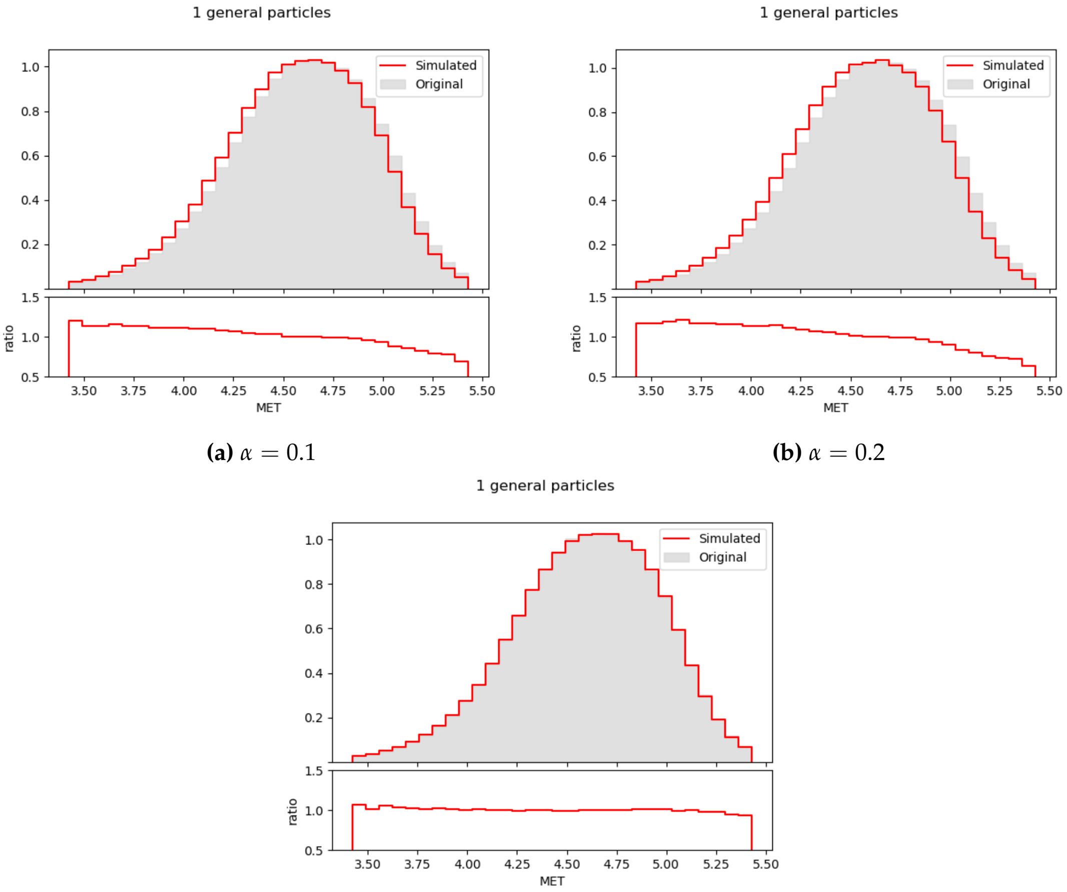


Figure 6.34. Generating events with the α-VAE without using BGMMs.

In the non-BGMM variant, the already trained α-VAE is used to generate encodings of events that are then altered with Gaussian noise and then decoded into newly generated events, as depicted in figure 6.34. Here, we compare all the chosen α values for each event and particle features, and the full result set can be seen in appendix E.3.1.



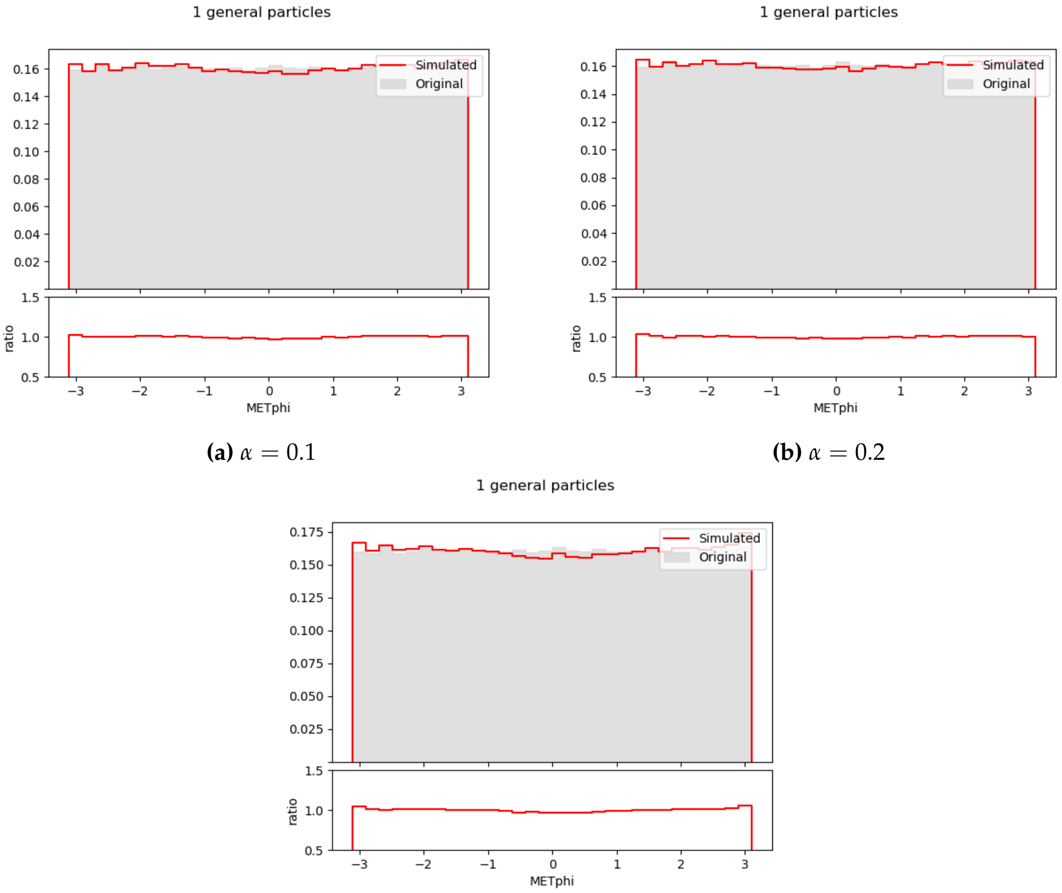
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.35. Histograms of `MET` event parameter in experiment III with different α values.

First, examining the `MET` event feature we can observe a great fit for all values of α. Here, the results get slightly better as α increases, contrary to what was previously the case when using the β-VAE.



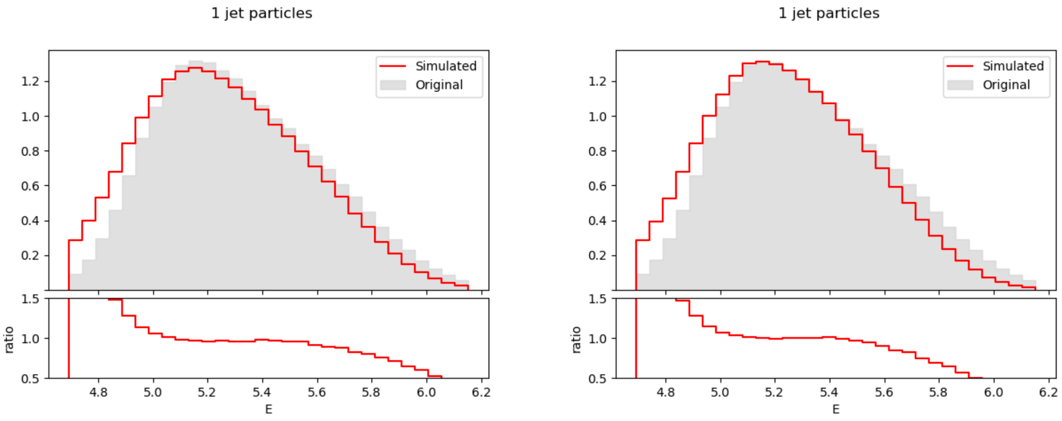
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

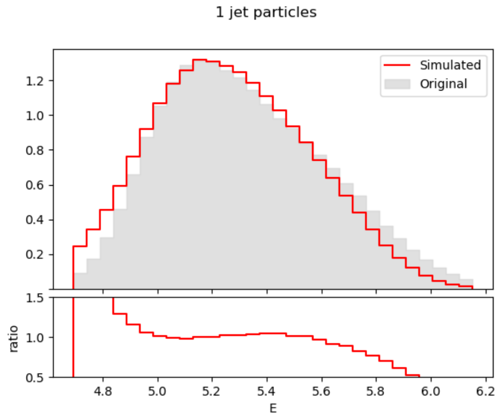
Figure 6.36. Histograms of `METphi` event parameter in experiment III with different α values.

Observing the `METphi` event feature, the effect is similar to what happened before in the `MET` feature. This almost perfect adjustment was obtained obtained by separating both of these event features into a separate input layer.



(a) $\alpha = 0.1$

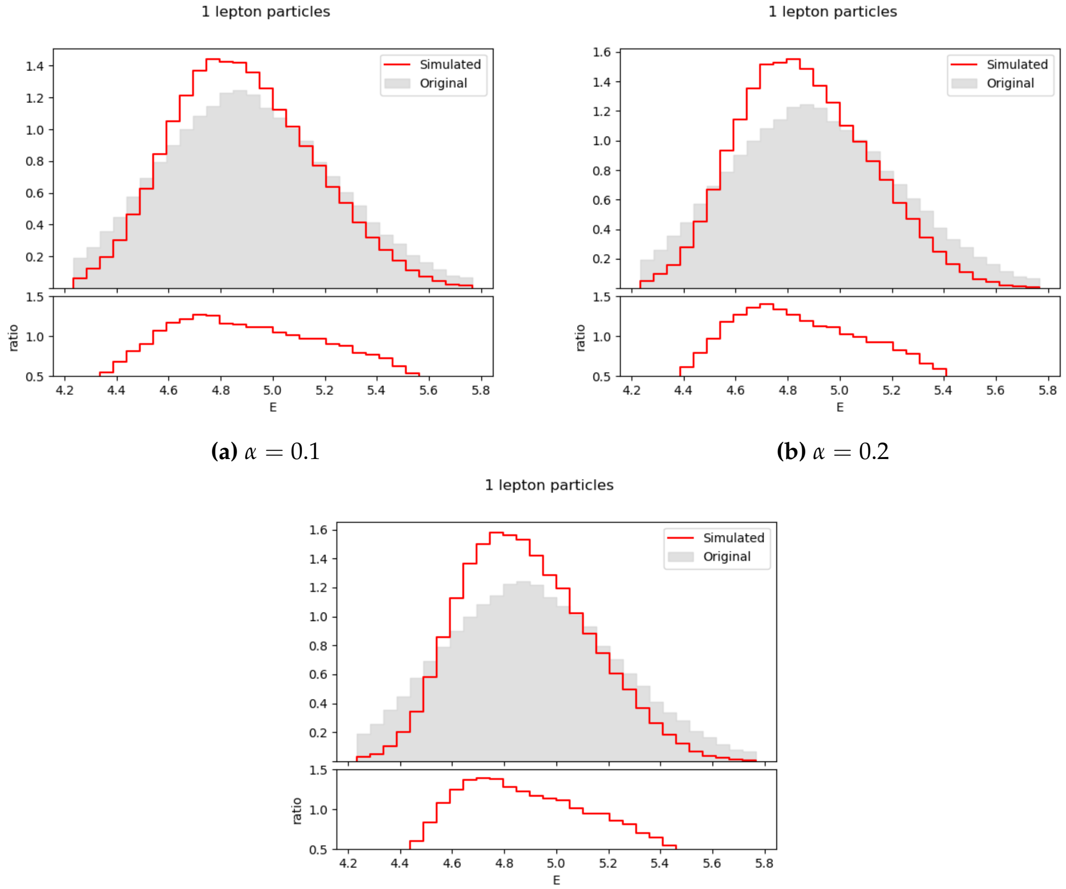
(b) $\alpha = 0.2$



(a) $\alpha = 0.3$

Figure 6.38. Histograms of first jet parameter E in experiment III with different α values.

Here, in the E particle feature of the first jet, the adjustment to the original MC data distribution is also pretty accurate, with the exception of some data from the right tail of the original distribution moving into the left tail of the generated one.



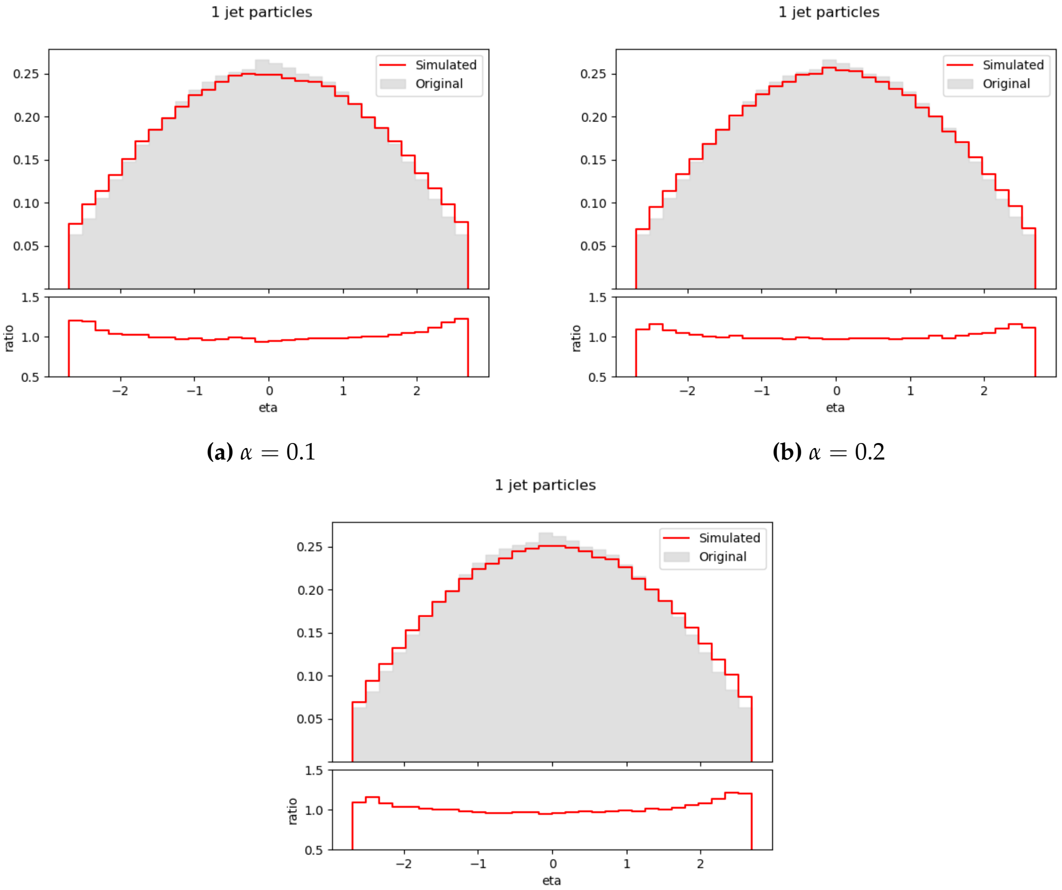
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.39. Histograms of first lepton parameter E in experiment III with different α values.

Analyzing the E particle feature of the first lepton, we see that some data from the left and right tails of the original distribution is concentrated in the middle of the generated one. Despite that, the result is an acceptable fit that does not change as α increases.



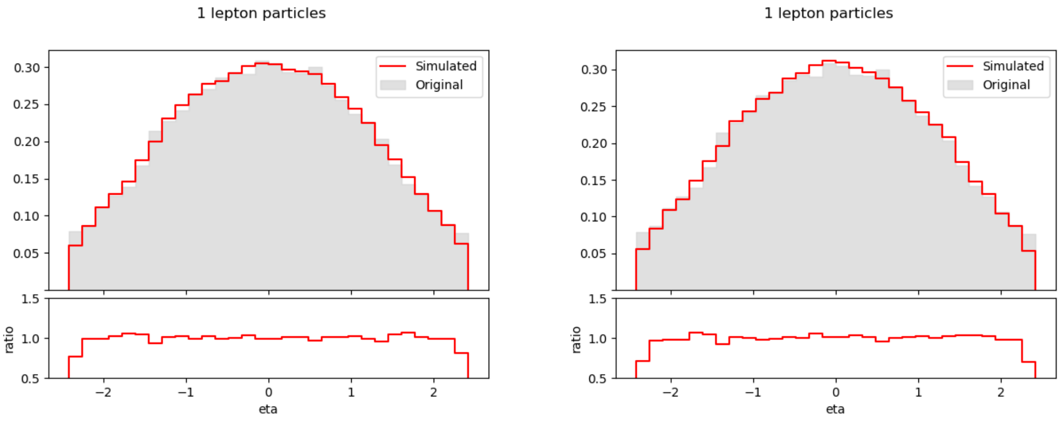
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

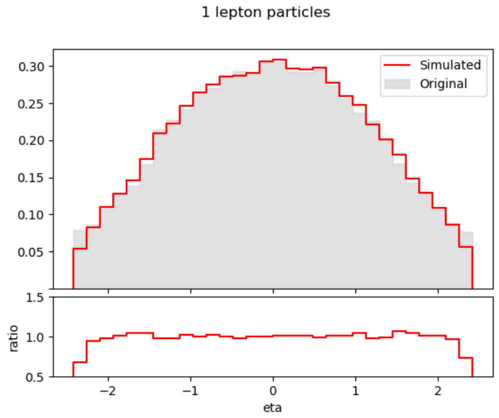
Figure 6.40. Histograms of first jet parameter η in experiment III with different α values.

The η particle feature of the first jet in this experiment features a very similar distribution when compared to the *ground truth* data, only losing a few values from the middle of the distribution. We notice that the ratio is always close to 1.0 and the effect of α is not relevant once again.



(a) $\alpha = 0.1$

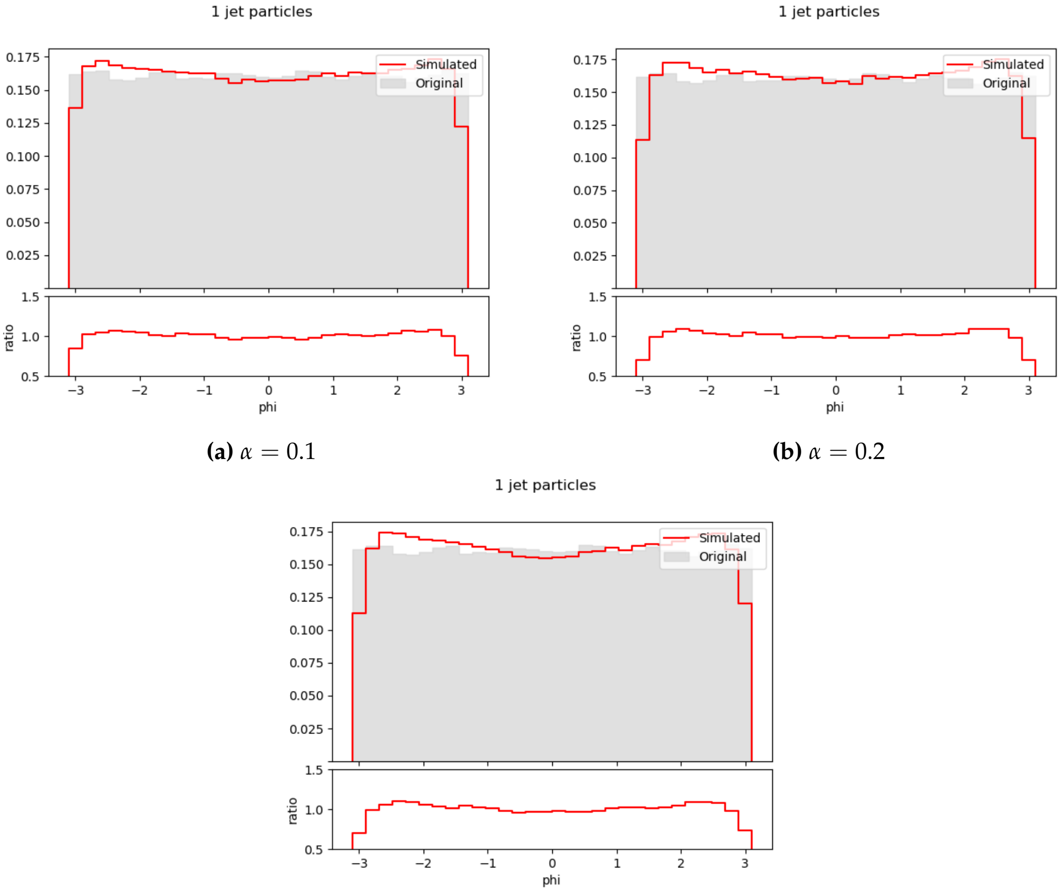
(b) $\alpha = 0.2$



(a) $\alpha = 0.3$

Figure 6.42. Histograms of first lepton parameter η in experiment III with different α values.

The η particle feature of the first lepton, however, shows a difference when the value of α increases, improving the results a bit. Even though this change is noticeable in this case, the results are great for all values of α.



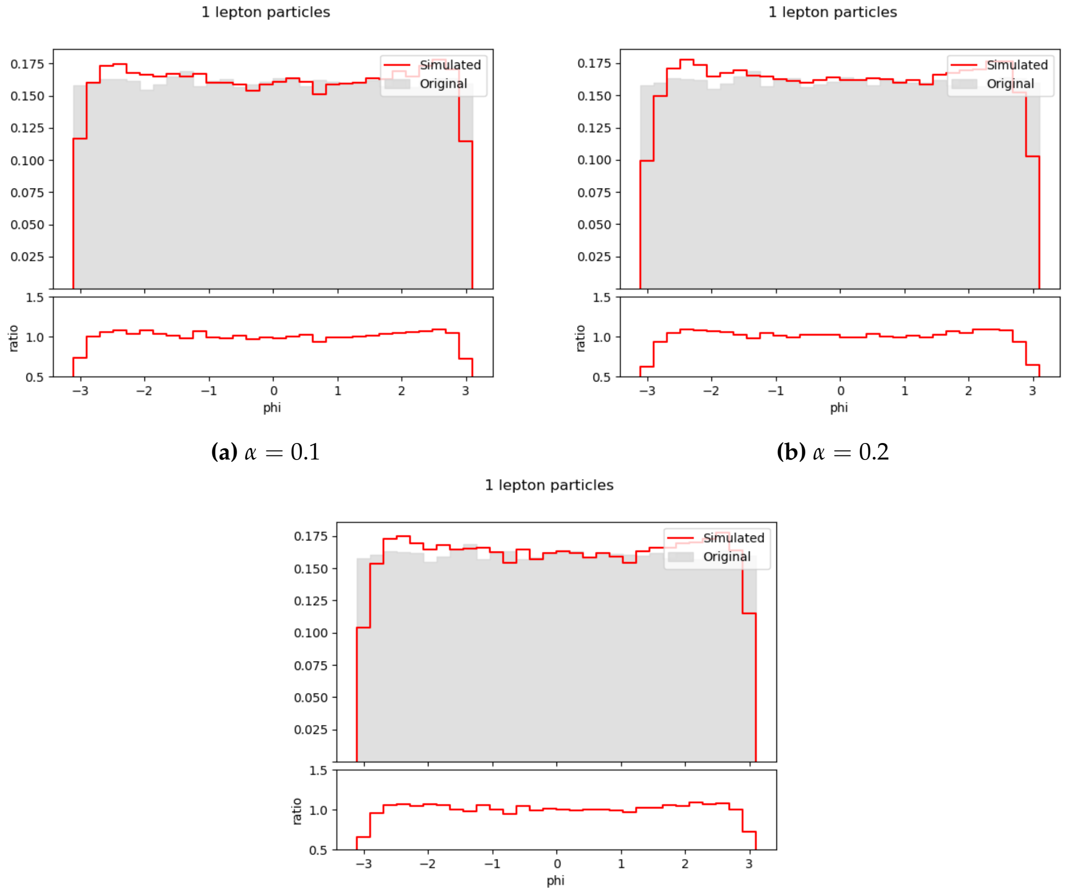
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.43. Histograms of first jet parameter ϕ in experiment III with different α values.

Analyzing the ϕ particle feature of the first jet, we observe that the model learns how to adjust to the uniform distribution very accurately, only showing some differences in the sides of the generated distribution. The ratio holds stable values around 1.0.



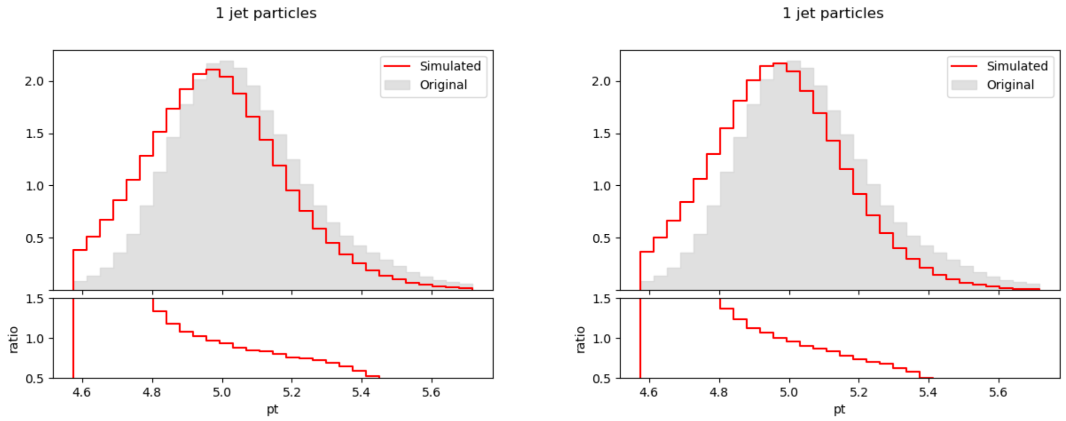
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

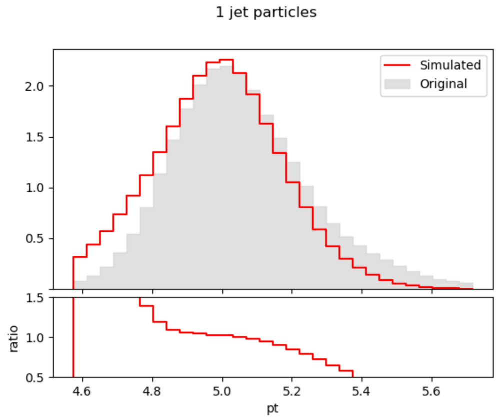
Figure 6.44. Histograms of first lepton parameter ϕ in experiment III with different α values.

The ϕ particle feature of the first lepton has a very similar behaviour to the one of the same feature of the first jet, and shows an accurate adjustment to the original data, also having a uniform distribution.



(a) $\alpha = 0.1$

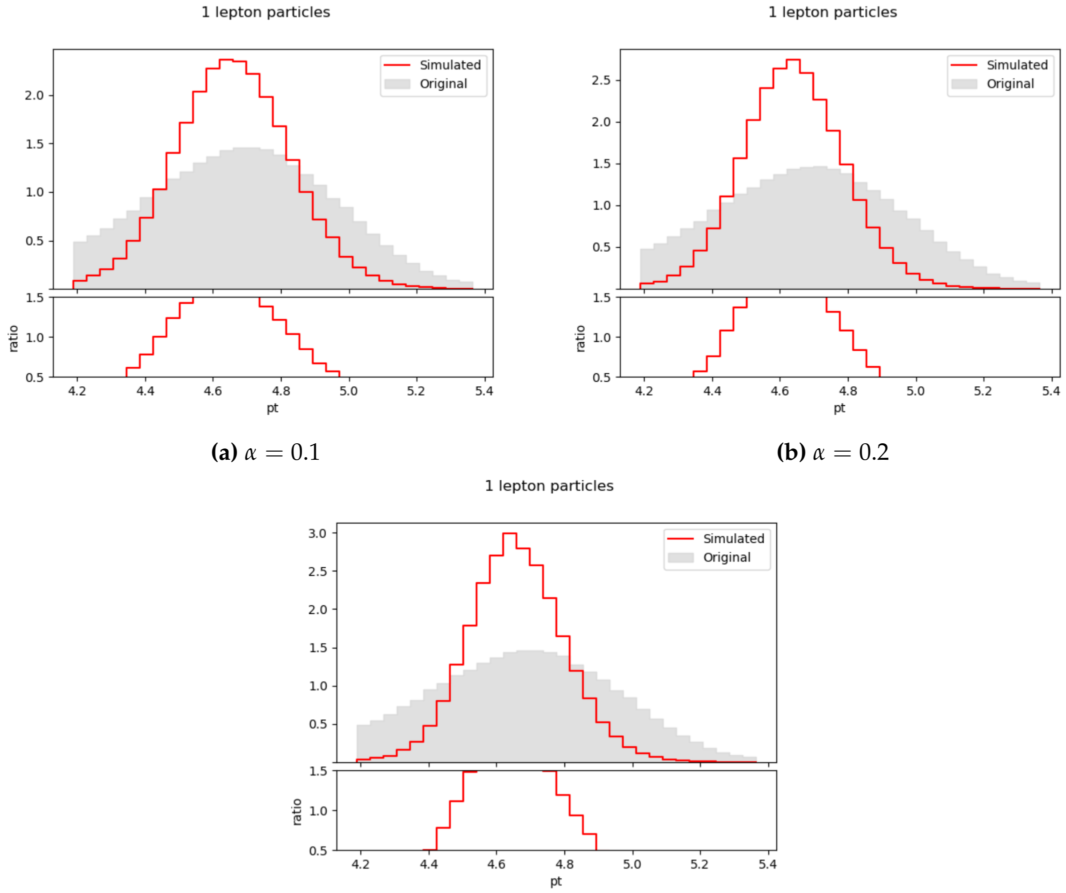
(b) $\alpha = 0.2$



(a) $\alpha = 0.3$

Figure 6.46. Histograms of first jet parameter p_T in experiment III with different α values.

Observing the p_T particle feature of the first jet, we see a small shift of the values of the right tail to the left, that are probably represented in the left tail of the generated values. The ratio is not as stable as some other features.



(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.47. Histograms of first lepton parameter p_T in experiment III with different α values.

Regarding the p_T particle feature of the first lepton, this time we see that for all values of α, the values from the right and left tails of the original MC events distribution are located in the central part of the generated events distribution.

In general, the results using this approach were very accurate and showed that this version of VAE has great potential when generating events from already existing ones.

6.5.2. With BGMMs

In the BGMM variant, the already trained encoder from the first step is detached from the rest of the VAE and its output becomes the input of a Bayesian Gaussian Mixture Model (fig. 6.48) in form of 200K generated latent representations. This BGMM is trained, creating a model that is capable of generating valid encodings for simulated events.

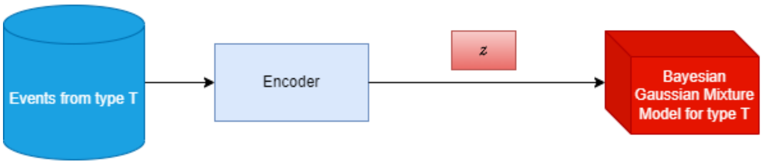


Figure 6.48. Training the BGMM using the generated latent representations from the α-VAE.

Then, the output of the BGMM is connected to the rest of the previously trained VAE (fig. 6.49), producing new event encodings that can be decoded into the desired events.

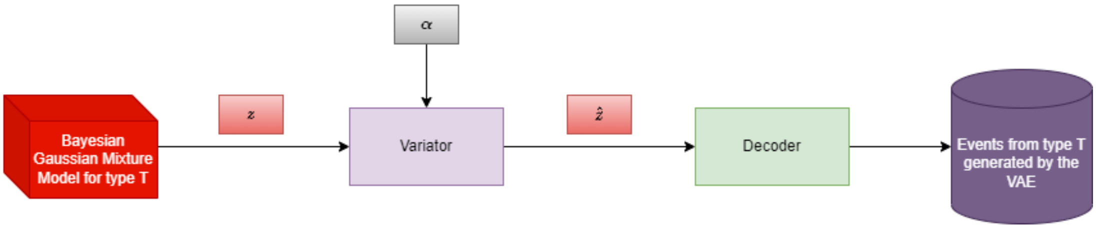
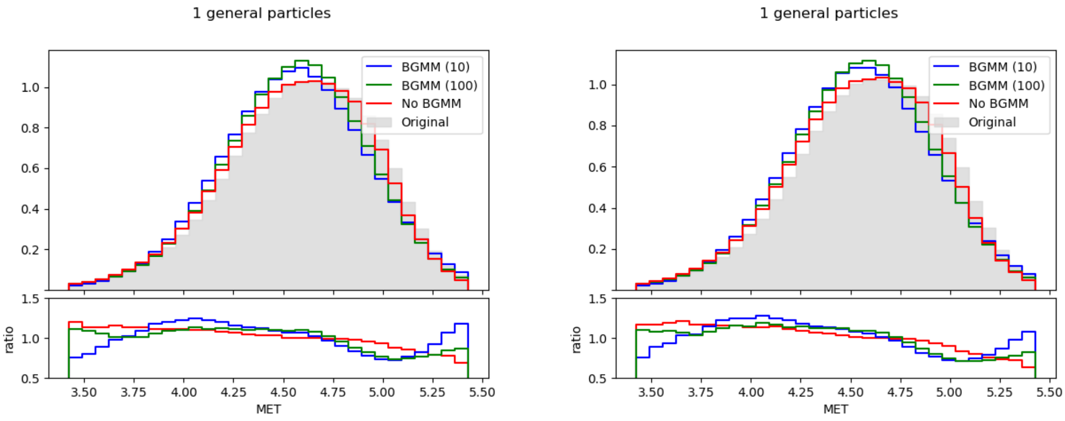

α
Bayesian Gaussian Mixture Model for type T
z
Variator
$\hat{z}$
Decoder
Events from type T generated by the VAE

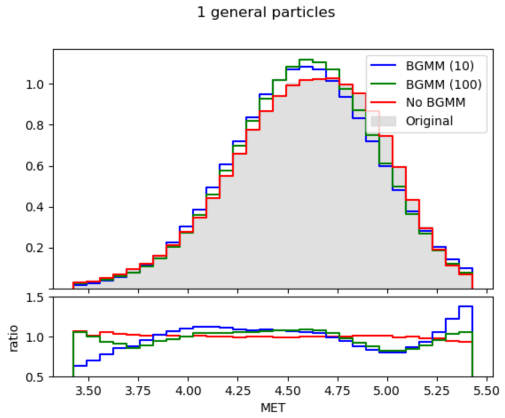
Figure 6.49. Sampling from the BGMM to generate events from the α-VAE.

When this procedure was properly defined, we tested several α-VAE configurations to look for the optimal one. Many tests were performed, combining the previously specified values of α with a different number of BGMM components $\gamma \in \{10, 20, 50, 100\}$, always using a full covariance matrix for each component.

We noticed that for a number of components $\gamma \geq 20$ the results are very similar so, for the results to be easily comparable, we show here a comparison using only $\gamma \in \{10, 100\}$. However, the full result set can be found in appendix E.3.2.



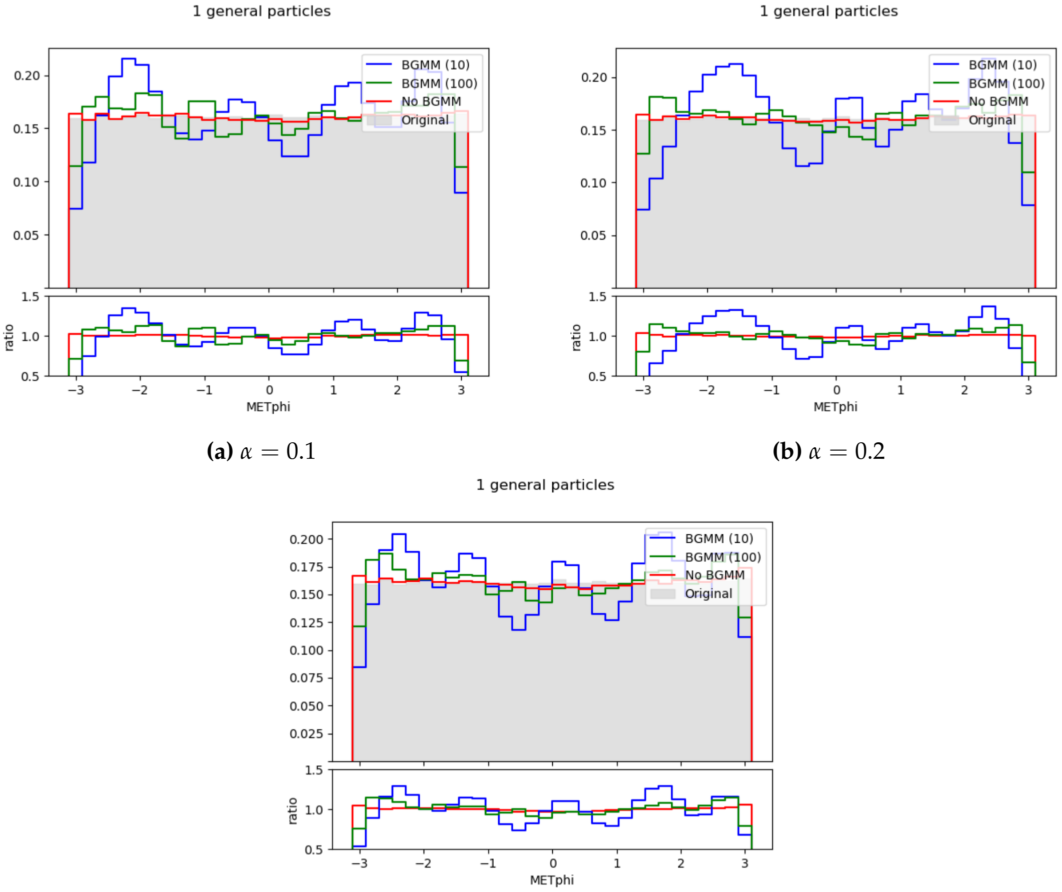
(a) $\alpha = 0.1$ **(b)** $\alpha = 0.2$



(c) $\alpha = 0.3$

Figure 6.50. Comparison of `MET` event parameter with (●, ●) and without (●) BGMMs.

Comparing the `MET` values of all the events, we observe that both the BGMM and non-BGMM approaches model the data accurately (with a certain margin of error). However, in the ratio histogram we see the non-BGMM model being more stable.



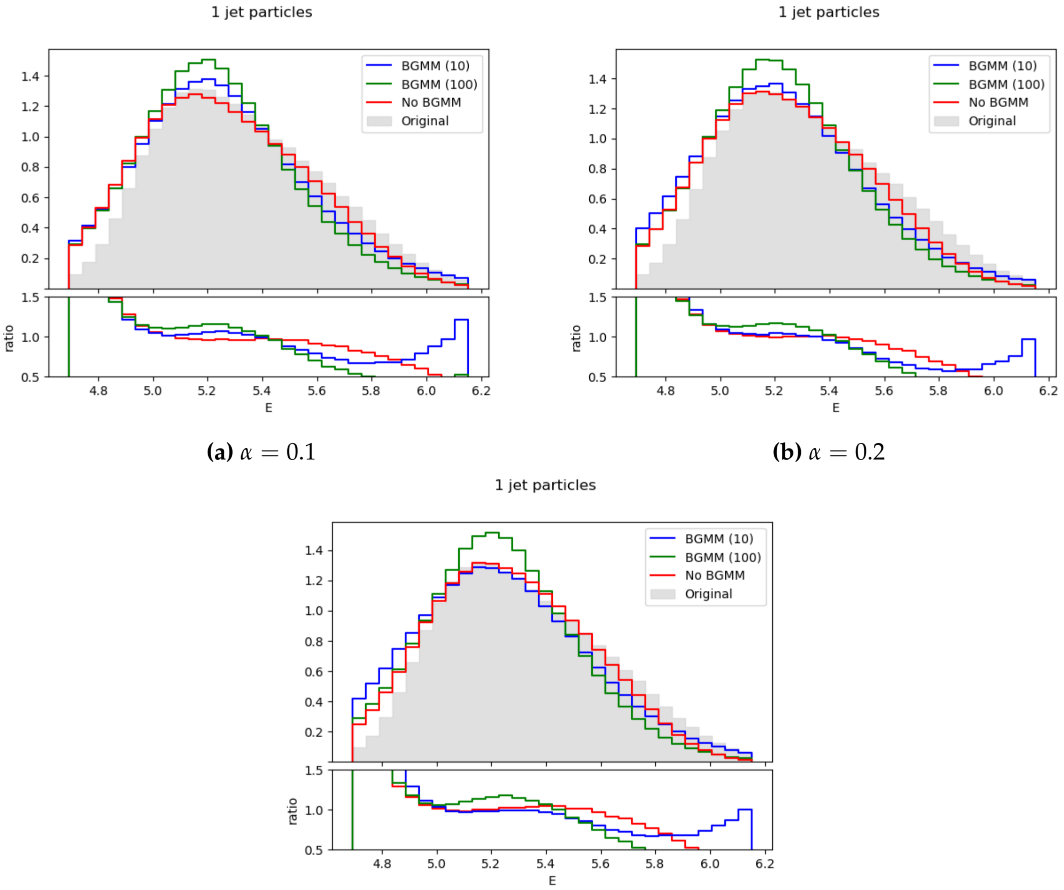
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.51. Comparison of `METphi` event parameter with (●, ●) and without (●) BGMMs.

Observing the `METphi` values, here the result varies significantly, as the adjustment to the uniform distribution that was achieved in the non-BGMM model is no longer present. The data for all values of α turns into several normal distributions mixed together.



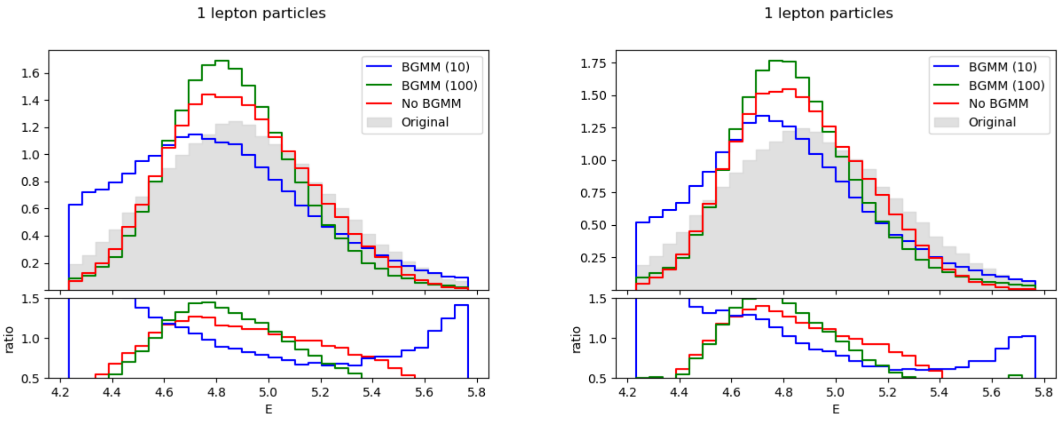
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

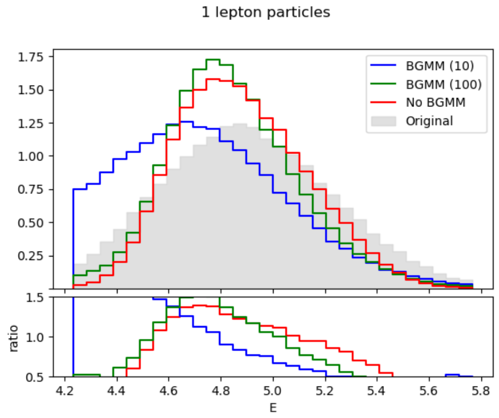
Figure 6.52. Comparison of first jet parameter E with (●, ●) and without (●) BGMMs.

In the E particle feature of the first jet, we see a few values of the right tail of the original distribution grouped in the central and left part of the generated data, obtaining worse events than the non-BGMM variant. When $\gamma = 100$, the distribution is more spiky and concentrates more values in the middle part.



(a) $\alpha = 0.1$

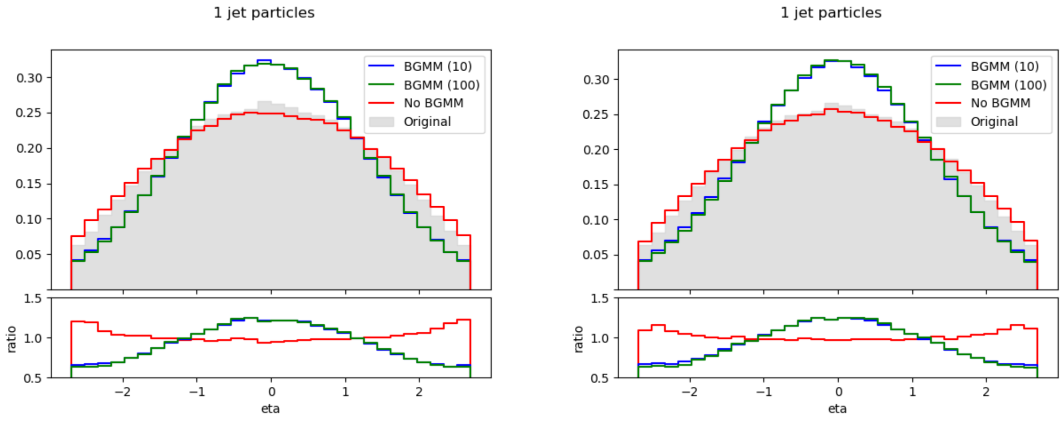
(b) $\alpha = 0.2$



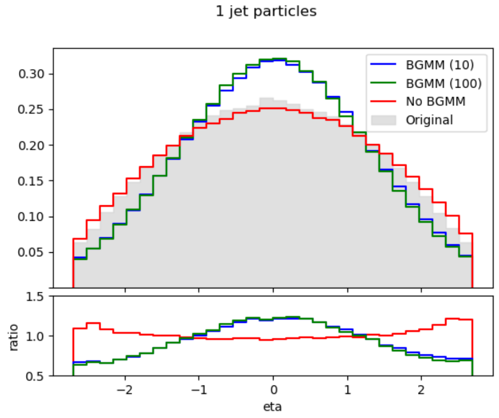
(a) $\alpha = 0.3$

Figure 6.54. Comparison of first lepton parameter E with (●, ●) and without (●) BGMMs.

The E particle feature of the first lepton shows a similar distribution in both variants when $\gamma = 100$. Nevertheless, when $\gamma \geq 20$, we obtain a more 'flat' distribution in the BGMM variant, that still does not adjust properly to the original MC data.



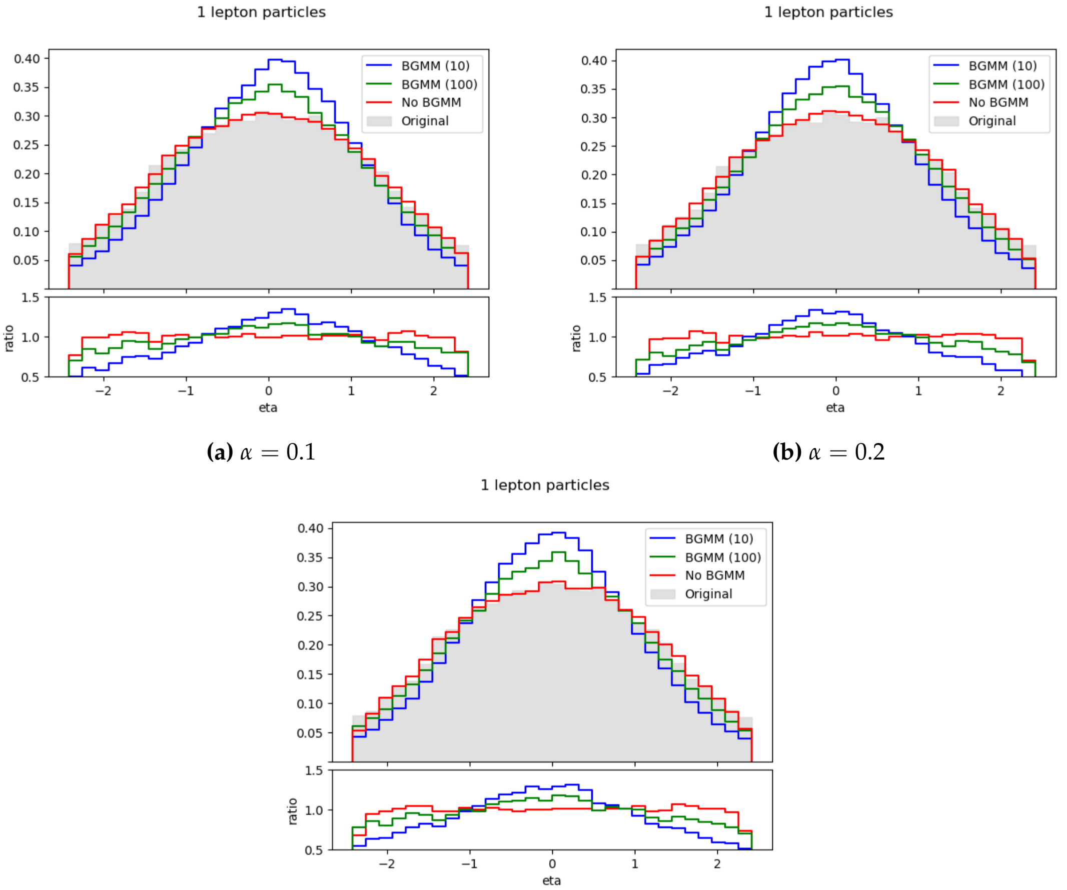
(a) $\alpha = 0.1$ **(b)** $\alpha = 0.2$



(c) $\alpha = 0.3$

Figure 6.55. Comparison of first jet parameter η with (●, ●) and without (●) BGMMs.

Analyzing the η particle feature of the first jet, we lose once again the accurate fit obtained in the non-BGMM variant, resulting in a distribution that concentrates some values from the tails of the *ground truth* distribution in the center of the generated events.



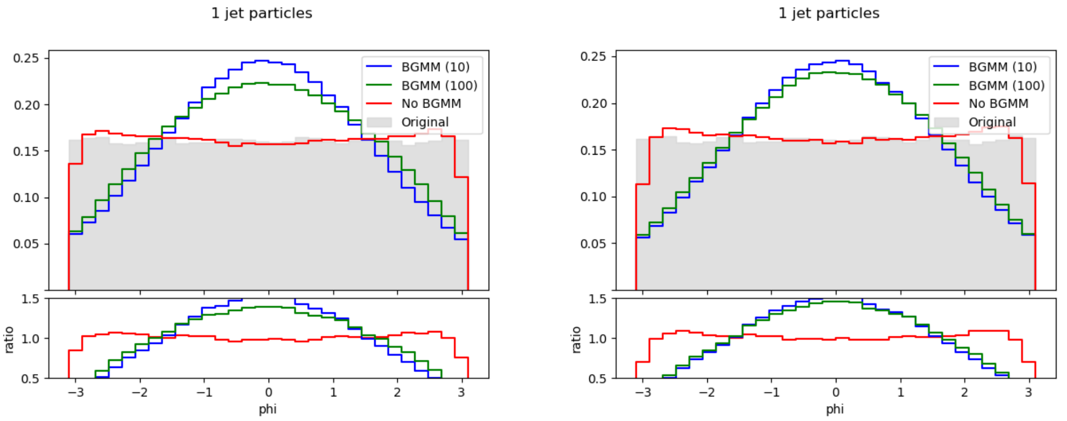
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

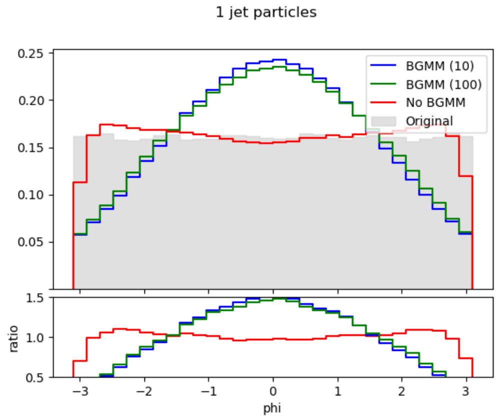
Figure 6.56. Comparison of first lepton parameter η with (●, ●) and without (●) BGMMs.

The η particle feature of the first lepton behaves similarly than the first jet, reaching a better adjustment without using a BGMM and with a more stable ratio. However, in this case when $\gamma \geq 20$, the results are closer to those obtained in the non-BGMM variant.



(a) $\alpha = 0.1$

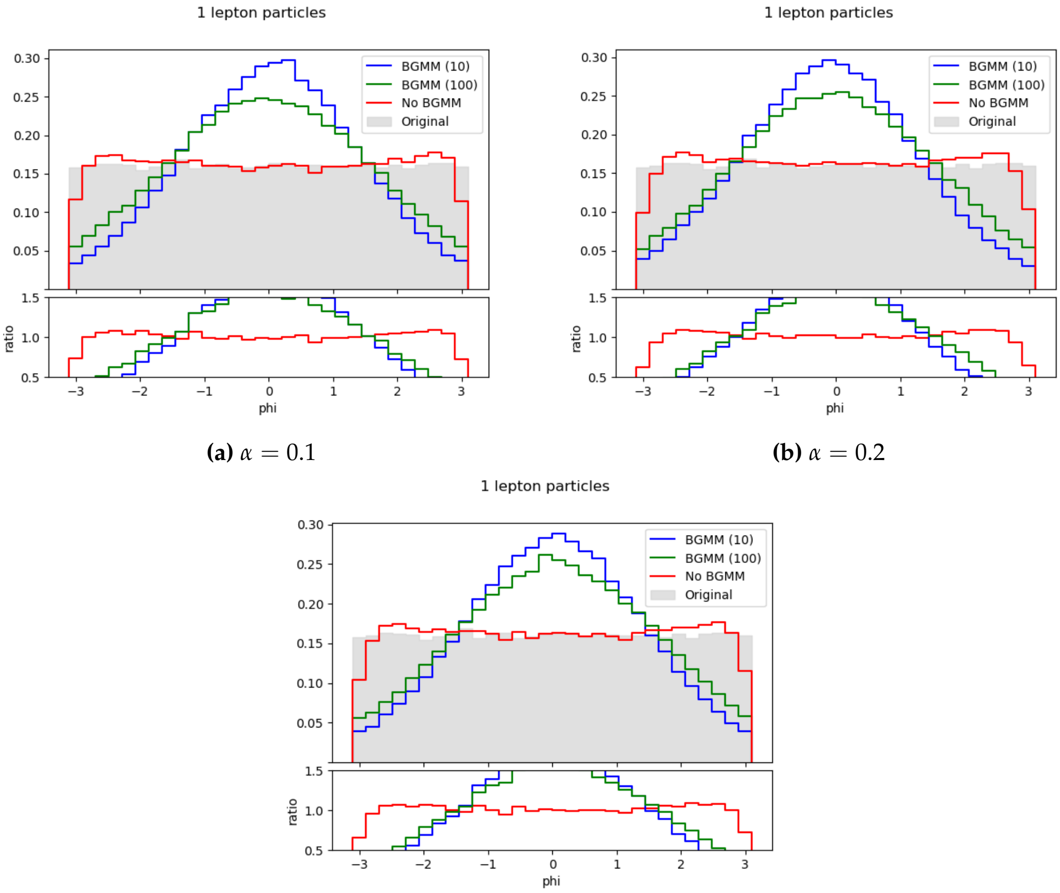
(b) $\alpha = 0.2$



(a) $\alpha = 0.3$

Figure 6.58. Comparison of first jet parameter ϕ with (●, ●) and without (●) BGMMs.

Regarding the ϕ particle feature of the first jet, here the great fit to the original MC data distribution that was achieved in the previous non-BGMM model is also lost, similarly than what happened in the METphi case.



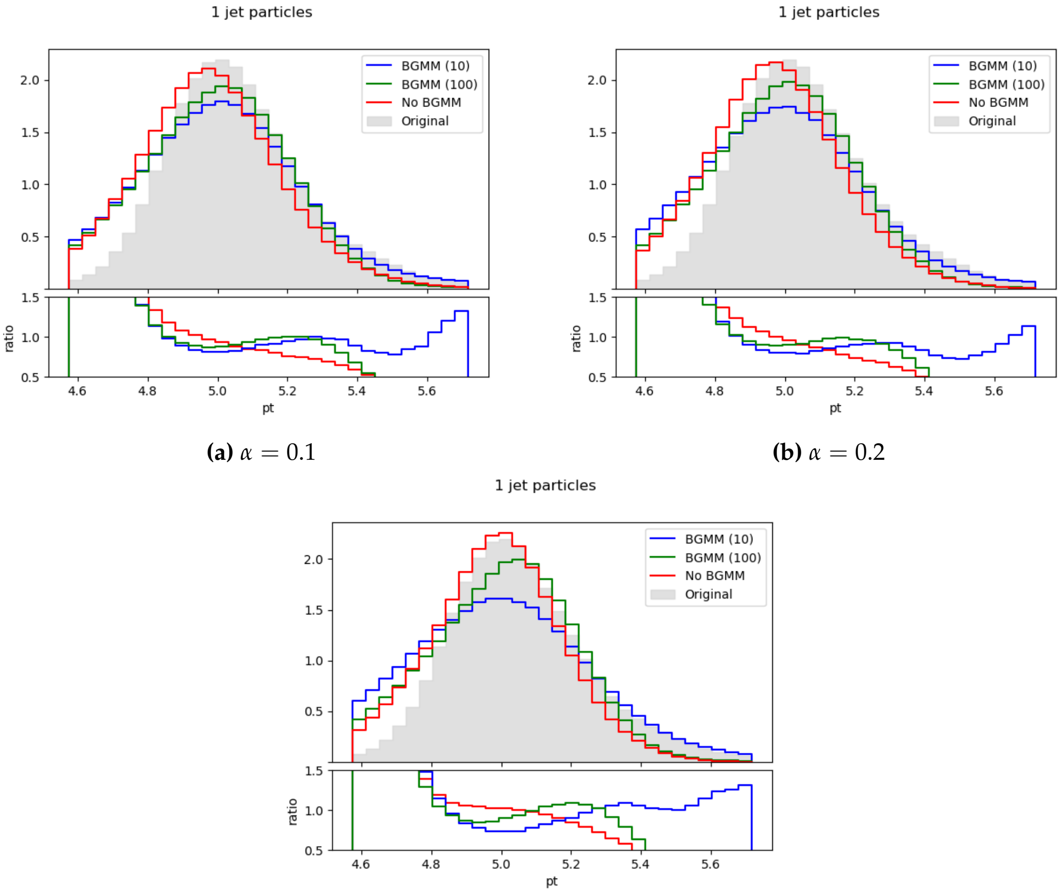
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

Figure 6.59. Comparison of first lepton parameter ϕ with (●, ●) and without (●) BGMMs.

In the ϕ particle feature of the first lepton, we find a resemblant behaviour when compared to the first jet. The uniform distribution is also transformed into a normal one, being the use of a BGMM the most probable reason of this effect.



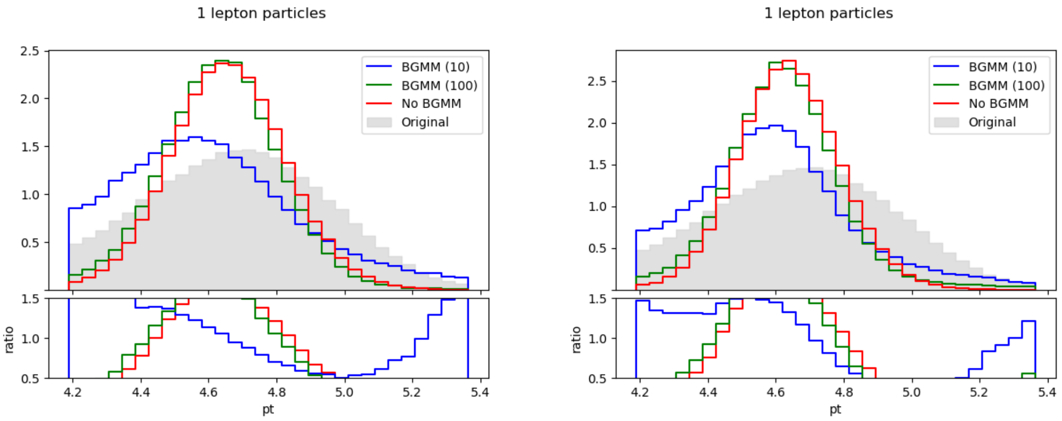
(a) $\alpha = 0.1$

(b) $\alpha = 0.2$

(c) $\alpha = 0.3$

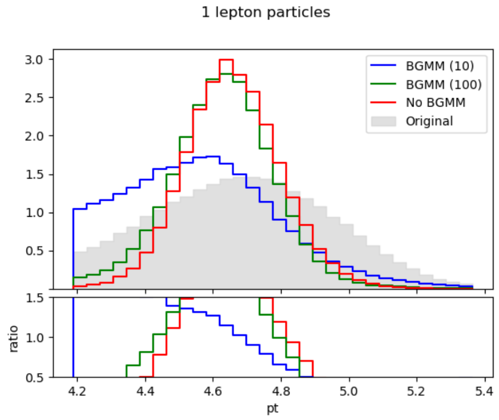
Figure 6.60. Comparison of first jet parameter p_T with (●, ●) and without (●) BGMMs.

Analyzing the p_T particle feature of the first jet, we see that it presents a similar distribution to the non-BGMM model for most combinations of α and γ: some of the data in the right tail of the original distribution is in the left tail of the generated events. Finally, when $\alpha = 0.3$ and $\gamma = 10$, we notice a more 'flat' shape, concentrating some values from the middle part of the MC events distribution.



(a) $\alpha = 0.1$

(b) $\alpha = 0.2$



(a) $\alpha = 0.3$

Figure 6.62. Comparison of first lepton parameter p_T with (●, ●) and without (●) BGMMs.

The p_T particle feature of the first lepton behaves similar to the non-BGMM variant when $\gamma \geq 20$, describing a normal distribution that is very different from the original MC events. In $\gamma = 10$, we notice a more similar distribution to the original data than the one obtained in the non-BGMM model (with some values in the right tail of the original distribution moved into the central/left part of the generated data distribution).

After analyzing all the results from this experiment, and considering its variants with and without the use of Bayesian Gaussian Mixture Models, we can state that this component affects negatively in the adjustment to uniform distributions, and generates events with a slightly worse fit to the original data.

6.6 BSM event generation

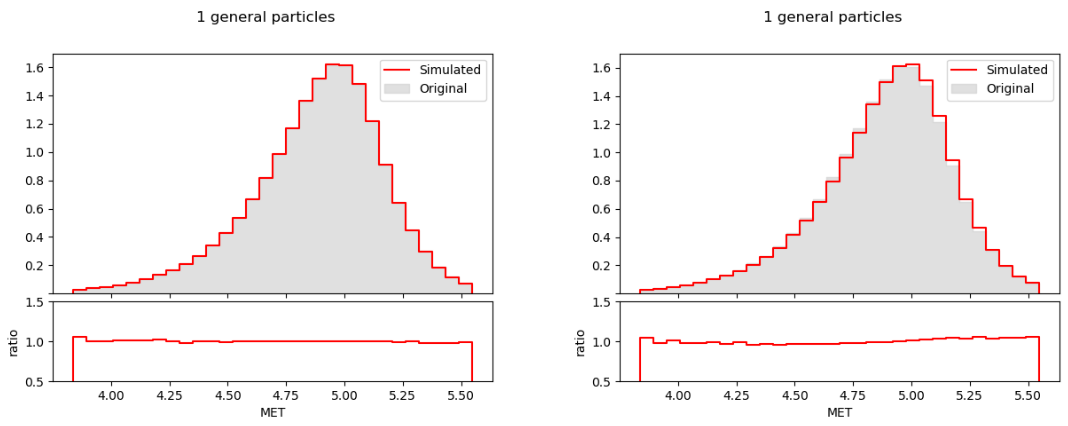
In experiment I, we trained a model using the whole dataset of a particular kind of event, obtaining some results that did not adjust properly to the original MC (*ground truth*) data distributions, but we decided to try some variants with the objective of expanding it further to other kinds of events once we found a model that was good enough. We improved our model during experiments II and III, until obtaining a promising approach that fitted the distributions of *ttbar* events very accurately.

To conclude our work, we decided to train the two models that obtained the best results among all the executed experiments with a different type of event: *stop_02*, from the Beyond the Standard Model event dataset (described in table 3.2). In this way, we can see how well this models perform for any kind of event, and we see whether we can generate BSM events that help proving if the theoretical foundations of these events can be observed in practice or not.

The chosen models that we highlight here were the following ones:

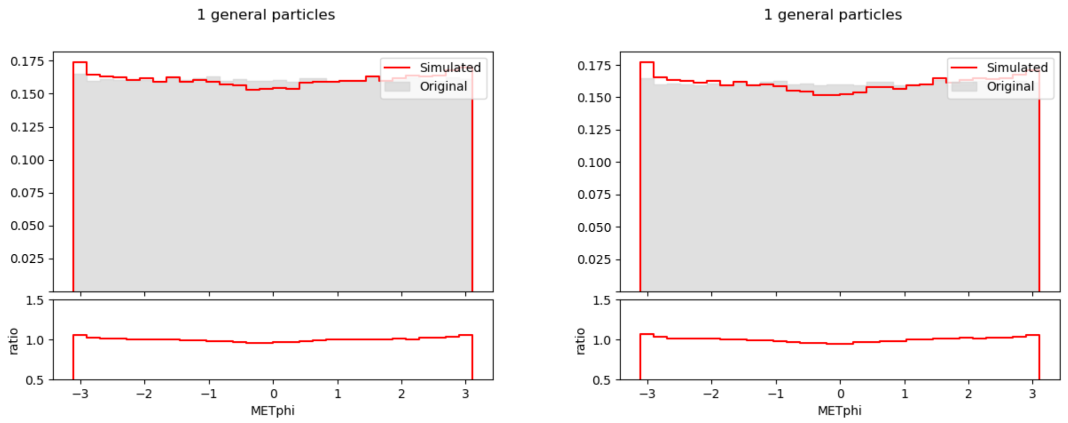
- Experiment II on its non-BGMM variant using $\beta = 0.001$
- Experiment III on its non-BGMM variant using $\alpha = 0.2$

Additional tests were run for experiment III using $\alpha \in \{0.1, 0.3\}$. However, the results were very similar to the version with $\alpha = 0.2$ presented here, so they are not included here for simplicity. Model variants with BGMMs were not tested due to lack of time, as they take a very long time to train, and also because the results of these models in previous experiments were not as good as its respective non-BGMM variants.



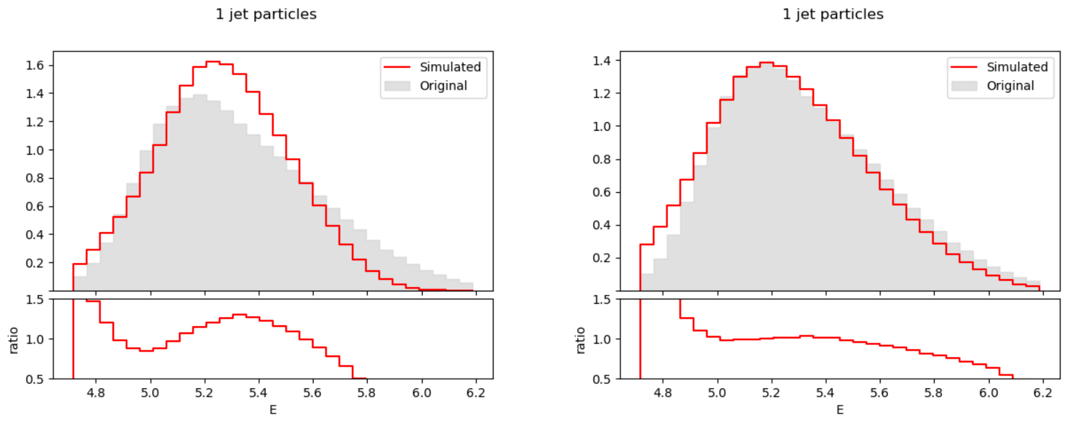
(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

Figure 6.63. Comparison of event parameter `MET` using the trained models in the BSM dataset.



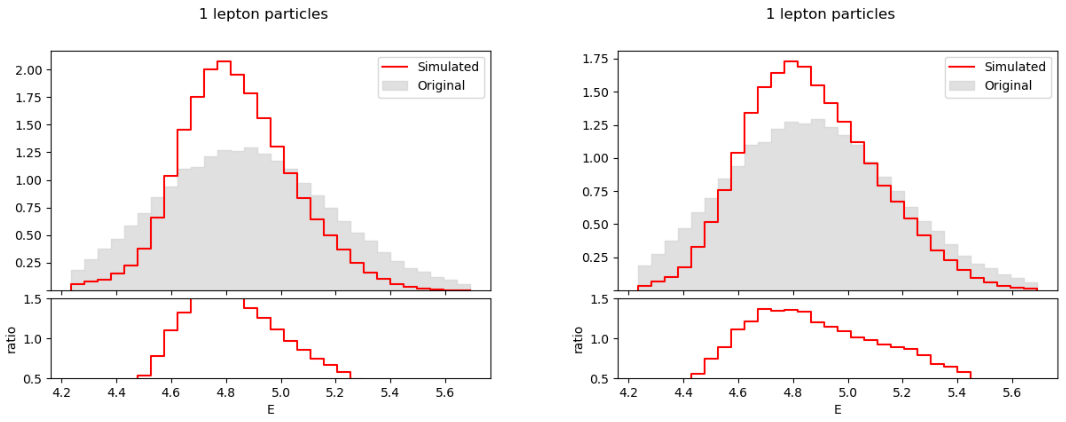
(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

Figure 6.64. Comparison of event parameter `METphi` using the trained models in the BSM dataset.



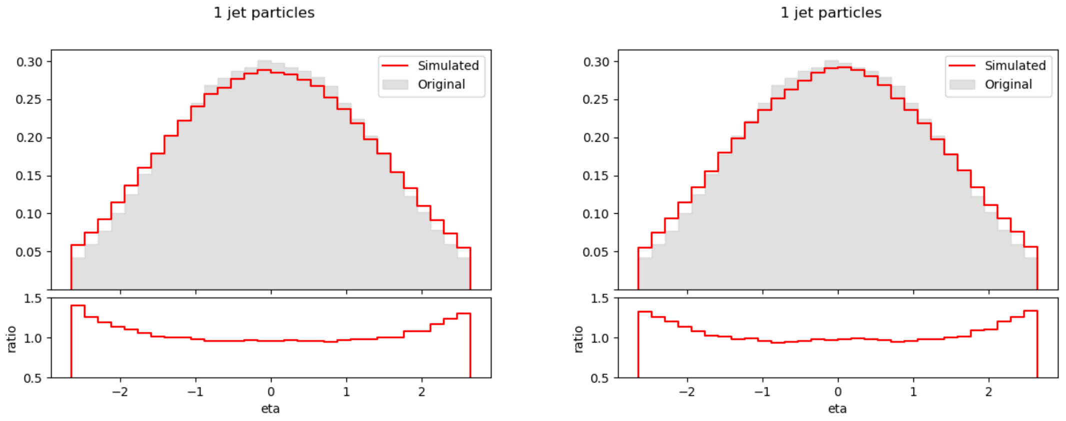
(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

Figure 6.65. Comparison of first jet parameter E using the trained models in the BSM dataset.



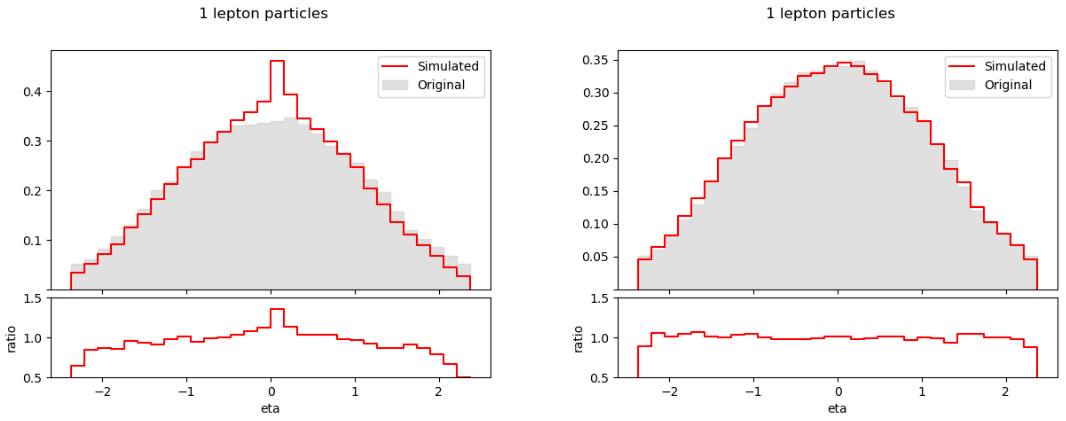
(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

Figure 6.66. Comparison of first lepton parameter E using the trained models in the BSM dataset.



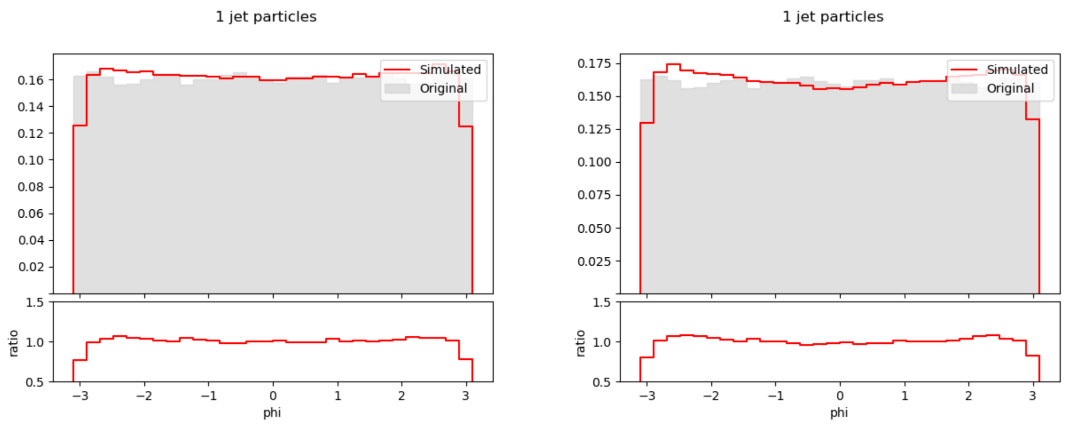
(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

Figure 6.67. Comparison of first jet parameter η using the trained models in the BSM dataset.



(a) β-VAE with $\beta = 0.001$ **(b)** α-VAE with $\alpha = 0.2$

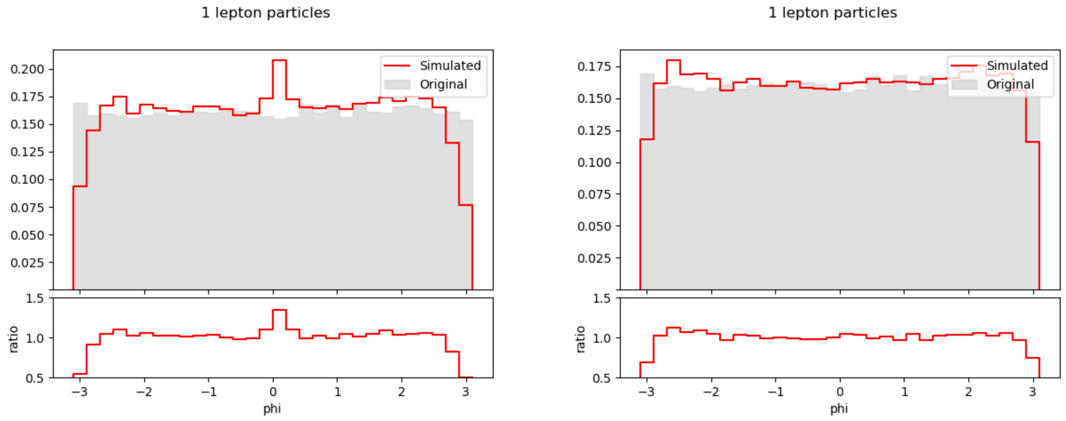
Figure 6.68. Comparison of first lepton parameter η using the trained models in the BSM dataset.



(a) β-VAE with $\beta = 0.001$

(b) α-VAE with $\alpha = 0.2$

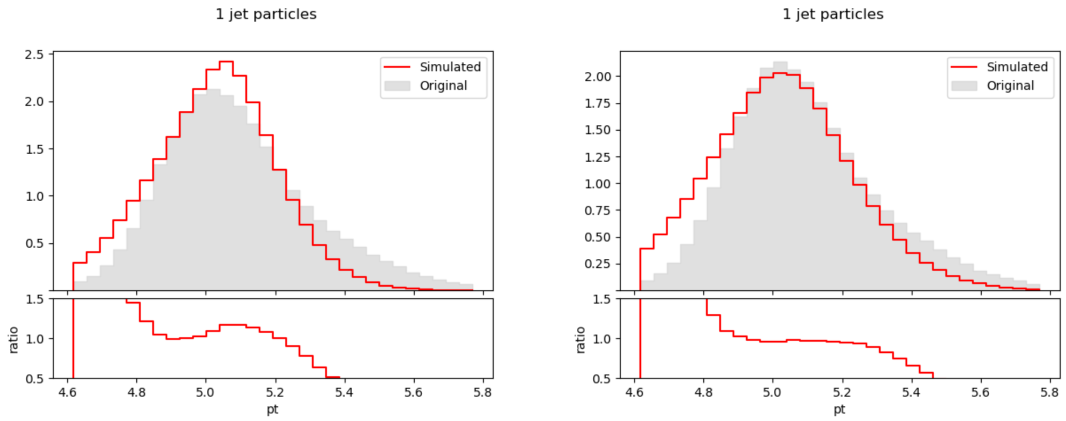
Figure 6.69. Comparison of first jet parameter ϕ using the trained models in the BSM dataset.



(a) β-VAE with $\beta = 0.001$

(b) α-VAE with $\alpha = 0.2$

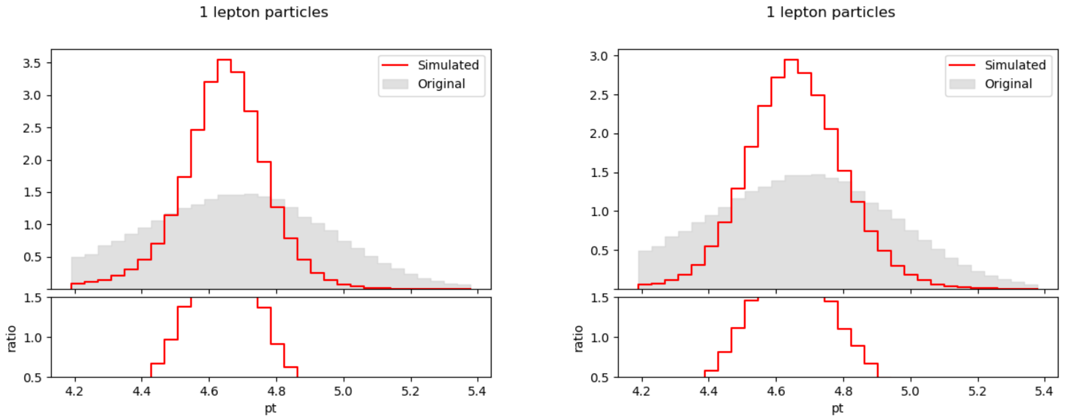
Figure 6.70. Comparison of first lepton parameter ϕ using the trained models in the BSM dataset.



(a) β-VAE with $\beta = 0.001$

(b) α-VAE with $\alpha = 0.2$

Figure 6.71. Comparison of first jet parameter p_T using the trained models in the BSM dataset.



(a) β-VAE with $\beta = 0.001$

(b) α-VAE with $\alpha = 0.2$

Figure 6.72. Comparison of first lepton parameter p_T using the trained models in the BSM dataset.

In this final event generation results, we can observe very interesting results that represent in an accurate way the original MC data, except for the p_T object feature in leptons, that is shifted or presents a spiky distribution that does not adjust properly to the original data. This could happen because there is a much higher probability of jets being present in an event than the one of leptons, so the model has less information about leptons and therefore their distribution is learnt in a less accurate way.

There are some cases in which the α-VAE obtains considerably better results than the β-VAE, such as the first jet E particle feature. However, for most features, both models perform similarly. Considering that the model has not been tweaked for this specific type of event, the results are promising.

CHAPTER 7
Conclusions

After developing and experimenting with several approaches, we obtained many interesting results in our work. Taking as a starting point some techniques used in recent studies, we generated events from the Standard Model and Beyond the Standard Model datasets in a more efficient way than traditional Monte Carlo methods.

We presented models using β-VAEs that performed better when using low β values ($\beta \leq 0.01$), and we also introduced our own variation of Variational Autoencoder: the α-VAE, that showed the best results when using relatively high values of α (≈ 0.3).

Although the previous approaches obtained great results, there is still a need for tweaking its internal architecture and parameters in order to refine the generation of some particle features, that did not adjust to their original distributions as well as it would have been desirable.

Moreover, when combining both of the previous approaches with an additional component, known as Bayesian Gaussian Mixture Model, the results were not as good as the ones obtained in previous variants.

Nevertheless, taking into account the initially defined objectives, we can conclude that they were fulfilled, as we created a framework for event generation that improved drastically the current traditional methods in terms of time and computational efficiency, while keeping (with reasonable margins) their accuracy.

7.1 Future work

Because the topic presented here still has lots of possibilities to explore, and due to its critical importance in the future, when it will be required to generate a huge amount of events, we propose future work that could be done starting from what we created.

In the first place, it would help to create several metrics to quantify more accurately the ability of a model to adjust to the original data, so that we can use them together with histograms to evaluate potential new models.

In the second place, because we believe that BGMMs could have some potential to improve our results, a greater number of components than the ones we have experimented with could be tested, so that it can be either confirmed that the component is not suitable for our task, or its results can improve and become even better than the current variants that do not use BGMMs. However, this potential improvement must be evaluated taking into account the required training time, which is higher than other model variants.

Finally, it would be interesting to expand our investigated techniques by including other kinds of models in our research. Generative Adversarial Networks have been tested before, but new implementations could improve current results. Flow models were recently used in this field for similar objectives, so they could also be a research option.

Bibliography

[1] ATLAS Collaboration. Observation of a new particle in the search for the Standard Model Higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1):1–29, sep 2012, doi:10.1016/j.physletb.2012.08.020.

[2] Jory Sonneveld. Searches for physics beyond the standard model at the LHC. 2019, doi:10.48550/ARXIV.1905.06239.

[3] ATLAS Experiment. Computer-generated image of the ATLAS detector compared with human figures, 2011. [Online; accessed May 27, 2022]. Available at: https://www.livescience.com/cern-atlas-experiment.

[4] Sydney Otten, Sascha Caron, Wieske de Swart, Melissa van Beekveld, Luc Hendriks, Caspar van Leeuwen, Damian Podareanu, Roberto Ruiz de Austri, and Rob Verheyen. Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer, 2019, doi:10.48550/ARXIV.1901.00875.

[5] Thea Aarrestad, Melissa van Beekveld, Marcella Bona, Antonio Boveia, Sascha Caron, Joe Davies, Andrea de Simone, Caterina Doglioni, Javier Duarte, Amir Farbin, et al. The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider. *SciPost Physics*, 12(1), jan 2022, doi:10.21468/scipostphys.12.1.043.

[6] ATLAS Collaboration. A strategy for a general search for new phenomena using data-derived signal regions and its application within the ATLAS experiment. *The European Physical Journal C*, 79(2), feb 2019, doi:10.1140/epjc/s10052-019-6540-y.

[7] Bobak Hashemi, Nick Amin, Kaustuv Datta, Dominick Olivito, and Maurizio Pierini. LHC analysis-specific datasets with Generative Adversarial Networks, 2019, doi:10.48550/ARXIV.1901.05282.

[8] Riccardo Di Sipio, Michele Faucci Giannelli, Sana Ketabchi Haghighat, and Serena Palazzo. DijetGAN: a Generative-Adversarial Network approach for the simulation of QCD dijet events at the LHC. *Journal of High Energy Physics*, 2019(8), aug 2019, doi:10.1007/jhep08(2019)110.

[9] Joshua Lin, Wahid Bhimji, and Benjamin Nachman. Machine learning templates for QCD factorization in the search for physics beyond the standard model. *Journal of High Energy Physics*, 2019(5), may 2019, doi:10.1007/jhep05(2019)181.

[10] Anja Butter, Tilman Plehn, and Ramon Winterhalder. How to GAN LHC events. *SciPost Physics*, 7(6), dec 2019, doi:10.21468/scipostphys.7.6.075.

[11] Deep generative models for fast shower simulation in ATLAS. Technical report, CERN, Geneva, Jul 2018. All figures including auxiliary figures

are available at: https://atlas.web.cern.ch/Atlas/GROUPS/PHYSICS/PUBNOTES/ATL-SOFT-PUB-2018-001.

[12] Aishik Ghosh. Deep generative models for fast shower simulation in ATLAS. Technical report, CERN, Geneva, Jun 2019, doi:10.1088/1742-6596/1525/1/012077.

[13] Stefano Carrazza and Frédéric A. Dreyer. Lund jet images from generative and cycle-consistent adversarial networks. *The European Physical Journal C*, 79(11), nov 2019, doi:10.1140/epjc/s10052-019-7501-1.

[14] Erik Buhmann, Sascha Diefenbacher, Engin Eren, Frank Gaede, Gregor Kasieczka, Anatolii Korol, and Katja Krüger. Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed. *Computing and Software for Big Science*, 5(1), may 2021, doi:10.1007/s41781-021-00056-0.

[15] Fady Bishara and Marc Montull. (Machine) Learning amplitudes for faster event generation, 2019, doi:10.48550/ARXIV.1912.11055.

[16] Simon Badger and Joseph Bullock. Using neural networks for efficient evaluation of high multiplicity scattering amplitudes. *Journal of High Energy Physics*, 2020(6), jun 2020, doi:10.1007/jhep06(2020)114.

[17] Joshua Bendavid. Efficient Monte Carlo Integration Using Boosted Decision Trees and Generative Deep Neural Networks, 2017, doi:10.48550/ARXIV.1707.00028.

[18] Pratik Jawahar, Thea Aarrestad, Nadezda Chernyavskaya, Maurizio Pierini, Kinga A. Wozniak, Jennifer Ngadiuba, Javier Duarte, and Steven Tsan. Improving Variational Autoencoders for New Physics Detection at the LHC With Normalizing Flows. *Frontiers in Big Data*, 5, 2022, doi:10.3389/fdata.2022.803685.

[19] DarkMachines community. LHCsimulationProject, Feb 2020, doi:10.5281/zenodo.3685861. Available at: https://zenodo.org/record/3685861.

[20] G. Brooijmans, A. Buckley, S. Caron, A. Falkowski, B. Fuks, A. Gilbert, W. J. Murray, M. Nardecchia, J. M. No, R. Torre, et al. Les Houches 2019 Physics at TeV Colliders: New Physics Working Group Report, 2020, doi:10.48550/ARXIV.2002.12220.

[21] T. Aarrestad, M. van Beekveld, M. Bona, A. Boveia, S. Caron, J. Davies, A. De Simone, C. Doglioni, J. M. Duarte, A. Farbin, et al. The Dark Machines Anomaly Score Challenge: Benchmark Data and Model Independent Event Classification for the Large Hadron Collider. *SciPost Phys.*, 12:43, 2022, doi:10.21468/SciPostPhys.12.1.043.

[22] Wulff, Eric. Deep Autoencoders for Compression in High Energy Physics, 2020. Student Paper.

[23] Joseph Rocca and Baptiste Rocca. Understanding Variational Autoencoders (VAEs), Sep 2019. Available at: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73.

[24] Douglas A Reynolds. Gaussian mixture models. *Encyclopedia of biometrics*, 741(659-663), 2009.

[25] Gargeya Sharma. Top 3 ways to Write your tensorflow code, Jul 2021. Available at: https://www.analyticsvidhya.com/blog/2021/07/top-3-ways-to-write-your-tensorflow-code/.

[26] Bo Pang, Erik Nijkamp, and Ying Nian Wu. Deep Learning With TensorFlow: A Review. *Journal of Educational and Behavioral Statistics*, 45(2):227–248, 2020, doi:10.3102/1076998619872761.

[27] Navin Kumar Manaswi. *Understanding and Working with Keras*, pages 31–43. Apress, Berkeley, CA, 2018.

[28] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, 12:2825–2830, 2011.

[29] Paul Barrett, John Hunter, J Todd Miller, J-C Hsu, and Perry Greenfield. matplotlib–A Portable Python Plotting Package. In *Astronomical data analysis software and systems XIV*, volume 347, page 91, 2005.

[30] VideoCardZ/NVIDIA. NVIDIA TU104 graphics chip block diagram, 2019. [Online; accessed April 15, 2022]. Available at: https://videocardz.net/gpu/nvidia-tu104.

[31] Loss functions: Build an AI model: Peltarion platform. Available at: https://peltarion.com/knowledge-center/documentation/modeling-view/build-an-ai-model/loss-functions.

[32] Keras. Keras Documentation: BatchNormalization layer. Available at: https://keras.io/api/layers/normalization_layers/batch_normalization/.

List of Figures in Appendices

A.1 Comparison between Linear and ReLU activation functions. 74

B.1 Comparison of both loss functions used in the presented experiments. . . 75

D.1 Data distributions per object type and feature in the whole SM dataset. . . 84
D.2 Data distributions per object type and feature in the whole BSM dataset. . 90

E.1 Event MET and METphi distributions in experiment II ($\beta = 0.01$, $\gamma = 50$). . . 91
E.2 First jet particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$). . 92
E.3 First lepton particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$). 92
E.4 Second jet particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$). 93
E.5 Second lepton particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$). 93
E.6 Event MET and METphi distributions in experiment II ($\beta = 0.0$). 94
E.7 First jet particle feature distributions in experiment II ($\beta = 0.0$). 94
E.8 First lepton particle feature distributions in experiment II ($\beta = 0.0$). 95
E.9 Second jet particle feature distributions in experiment II ($\beta = 0.0$). 95
E.10 Second lepton particle feature distributions in experiment II ($\beta = 0.0$). . . 96
E.11 Event MET and METphi distributions in experiment II ($\beta = 0.001$). 96
E.12 First jet particle feature distributions in experiment II ($\beta = 0.001$). 97
E.13 First lepton particle feature distributions in experiment II ($\beta = 0.001$). . . . 97
E.14 Second jet particle feature distributions in experiment II ($\beta = 0.001$). . . . 98
E.15 Second lepton particle feature distributions in experiment II ($\beta = 0.001$). . 98
E.16 Event MET and METphi distributions in experiment II ($\beta = 0.01$). 99
E.17 First jet particle feature distributions in experiment II ($\beta = 0.01$). 99
E.18 First lepton particle feature distributions in experiment II ($\beta = 0.01$). . . . 100
E.19 Second jet particle feature distributions in experiment II ($\beta = 0.01$). 100
E.20 Second lepton particle feature distributions in experiment II ($\beta = 0.01$). . . 101
E.21 Event MET and METphi distributions in experiment II ($\beta = 0.1$). 101
E.22 First jet particle feature distributions in experiment II ($\beta = 0.1$). 102
E.23 First lepton particle feature distributions in experiment II ($\beta = 0.1$). 102
E.24 Second jet particle feature distributions in experiment II ($\beta = 0.1$). 103
E.25 Second lepton particle feature distributions in experiment II ($\beta = 0.1$). . . 103
E.26 Event MET and METphi distributions in experiment II ($\beta = 0.2$). 104
E.27 First jet particle feature distributions in experiment II ($\beta = 0.2$). 104
E.28 First lepton particle feature distributions in experiment II ($\beta = 0.2$). 105
E.29 Second jet particle feature distributions in experiment II ($\beta = 0.2$). 105
E.30 Second lepton particle feature distributions in experiment II ($\beta = 0.2$). . . 106
E.31 Event MET and METphi distributions in experiment II ($\beta = 0.5$). 106
E.32 First jet particle feature distributions in experiment II ($\beta = 0.5$). 107
E.33 First lepton particle feature distributions in experiment II ($\beta = 0.5$). 107
E.34 Second jet particle feature distributions in experiment II ($\beta = 0.5$). 108
E.35 Second lepton particle feature distributions in experiment II ($\beta = 0.5$). . . 108

E.36 Event MET and METphi distributions in experiment II ($\beta = 0.7$). 109
E.37 First jet particle feature distributions in experiment II ($\beta = 0.7$). 109
E.38 First lepton particle feature distributions in experiment II ($\beta = 0.7$). 110
E.39 Second jet particle feature distributions in experiment II ($\beta = 0.7$). 110
E.40 Event MET and METphi distributions in experiment II ($\beta = 1.0$). 111
E.41 First jet particle feature distributions in experiment II ($\beta = 1.0$). 111
E.42 First lepton particle feature distributions in experiment II ($\beta = 1.0$). 112
E.43 Second jet particle feature distributions in experiment II ($\beta = 1.0$). 112
E.44 Second lepton particle feature distributions in experiment II ($\beta = 1.0$). . . 113
E.45 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.0$).113
E.46 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.0$). 114
E.47 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.0$). 114
E.48 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.0$). 115
E.49 Second lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.0$). 115
E.50 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.001$). 116
E.51 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.001$). 116
E.52 First lepton particle feature using a BGMM in experiment II ($\beta = 0.001$). . 117
E.53 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.001$). 117
E.54 Second lepton particle feature using a BGMM in experiment II ($\beta = 0.001$). 118
E.55 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.01$). 118
E.56 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.01$). 119
E.57 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.01$). 119
E.58 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.01$). 120
E.59 Second lepton particle feature using a BGMM in experiment II ($\beta = 0.01$). 120
E.60 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.1$).121
E.61 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.1$). 121
E.62 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.1$). 122
E.63 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.1$). 122
E.64 Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.1$). 123
E.65 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.2$).123
E.66 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.2$). 124
E.67 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.2$). 124
E.68 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.2$). 125

E.69 Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.2$). 125
E.70 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.5$).126
E.71 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.5$). 126
E.72 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.5$). 127
E.73 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.5$). 127
E.74 Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.5$). 128
E.75 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.7$).128
E.76 First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.7$). 129
E.77 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.7$). 129
E.78 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.7$). 130
E.79 Event MET and METphi distributions using a BGMM in experiment II ($\beta = 1.0$).130
E.80 First jet particle feature distributions using a BGMM in experiment II ($\beta = 1.0$). 131
E.81 First lepton particle feature distributions using a BGMM in experiment II ($\beta = 1.0$). 131
E.82 Second jet particle feature distributions using a BGMM in experiment II ($\beta = 1.0$). 132
E.83 Event MET and METphi distributions in experiment III ($\alpha = 0.1$). 132
E.84 First jet particle feature distributions in experiment III ($\alpha = 0.1$). 133
E.85 First lepton particle feature distributions in experiment III ($\alpha = 0.1$). 133
E.86 Second jet particle feature distributions in experiment III ($\alpha = 0.1$). 134
E.87 Second lepton particle feature distributions in experiment III ($\alpha = 0.1$). . . 134
E.88 Event MET and METphi distributions in experiment III ($\alpha = 0.2$). 135
E.89 First jet particle feature distributions in experiment III ($\alpha = 0.2$). 135
E.90 First lepton particle feature distributions in experiment III ($\alpha = 0.2$). 136
E.91 Second jet particle feature distributions in experiment III ($\alpha = 0.2$). 136
E.92 Second lepton particle feature distributions in experiment III ($\alpha = 0.2$). . . 137
E.93 Event MET and METphi distributions in experiment III ($\alpha = 0.3$). 137
E.94 First jet particle feature distributions in experiment III ($\alpha = 0.3$). 138
E.95 First lepton particle feature distributions in experiment III ($\alpha = 0.3$). 138
E.96 Second jet particle feature distributions in experiment III ($\alpha = 0.3$). 139
E.97 Second lepton particle feature distributions in experiment III ($\alpha = 0.3$). . . 139
E.98 Event MET and METphi distributions using a BGMM ($\alpha = 0.1, \gamma = 10$). . . . 140
E.99 First jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 10$). . . 140
E.100First lepton particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 10$). 141
E.101Second jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 10$). . 141
E.102Second lepton particle feature distributions using BGMM ($\alpha = 0.1, \gamma = 10$). 142
E.103Event MET and METphi distributions using a BGMM ($\alpha = 0.1, \gamma = 20$). . . . 142
E.104First jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 20$). . . 143
E.105First lepton particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 20$). 143
E.106Second jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 20$). . 144
E.107Second lepton particle feature distributions using BGMM ($\alpha = 0.1, \gamma = 20$). 144
E.108Event MET and METphi distributions using a BGMM ($\alpha = 0.1, \gamma = 50$). . . . 145
E.109First jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 50$). . . 145

E.110First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$). 146
E.111Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$). . 146
E.112Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 50$). 147
E.113Event MET and METphi distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$). . . 147
E.114First jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$). . 148
E.115First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$). 148
E.116Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$). 149
E.117Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 100$).149
E.118Event MET and METphi distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$). . . . 150
E.119First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$). . . 150
E.120First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$). 151
E.121Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$). . 151
E.122Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 10$). 152
E.123Event MET and METphi distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$). . . . 152
E.124First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$). . . 153
E.125First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$). 153
E.126Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$). . 154
E.127Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 20$). 154
E.128Event MET and METphi distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$). . . . 155
E.129First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$). . . 155
E.130First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$). 156
E.131Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$). . 156
E.132Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 50$). 157
E.133Event MET and METphi distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$). . . 157
E.134First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$). . 158
E.135First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$). 158
E.136Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$). 159
E.137Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 100$).159
E.138Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$). . . . 160
E.139First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$). . . 160
E.140First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$). 161
E.141Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$). . 161
E.142Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 10$). 162
E.143Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$). . . . 162
E.144First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$). . . 163
E.145First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$). 163
E.146Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$). . 164
E.147Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 20$). 164
E.148Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$). . . . 165
E.149First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$). . . 165
E.150First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$). 166
E.151Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$). . 166
E.152Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 50$). 167
E.153Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$). . . 167
E.154First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$). . 168
E.155First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$). 168
E.156Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$). 169
E.157Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 100$).169

APPENDIX A
Activation functions

The elemental unit of neural networks are neurons themselves. In ANNs, like in the human brain, there are connections between neurons, some stronger than others, and pathways are established among them that can activate different brain functions. However, not all neurons are activated simultaneously: some are activated and others are deactivated depending on the activity being performed. In the case of neural networks, neurons are units that that perform a computation similar to the one described in A.1.

$$out = \sum(weight * input) + bias \tag{A.1}$$

Furthermore, the neuron's output does not have to be a binary output that allows us to easily determine whether the neuron is activated or not. Thus, to establish the relationship between the output values of a neuron and its activation state, we use activation functions.

A.1 Linear

The simplest activation function is the **linear** one, also known as "no activation" (multiply by 1.0), where the activation is proportional to the input. The function leaves the weighted sum of the input as it is, simply returning the value it was given.

A.2 Rectified Linear Unit

The **rectified linear** (ReLU) function is a piecewise linear function that will output the input that it was given directly if it is positive, and otherwise it will output zero. It has become the default activation function for many types of neural networks because models that use it are easier to train and often achieve better results.

A comparison of this function against the basic linear one can be seen in figure A.1.

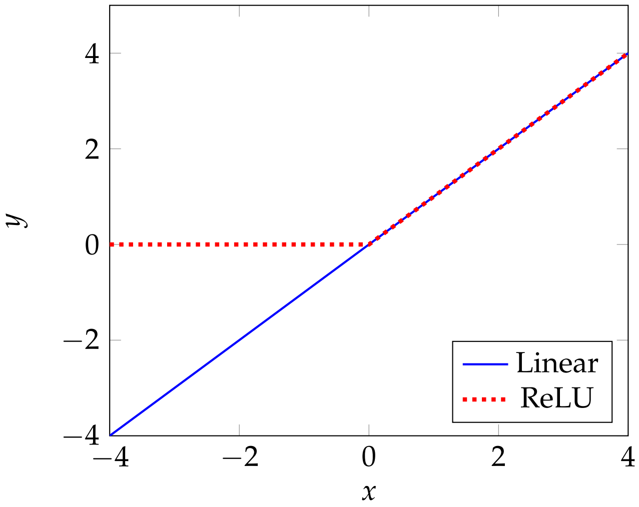


Figure A.1. Comparison between Linear and ReLU activation functions.

A.3 Softmax

The **softmax** activation function is widely used in classification tasks, as it transforms the input values into a vector of probabilities with as many components as the number of classes. Each component of the vector represents the probability of the sample belonging to the class represented by that component.

This activation function is often used in the last layer of neurons of a network dedicated to classify the samples into several classes. The last layer has as many neurons as classes and the softmax function transforms the output of the layer into a vector of probabilities.

APPENDIX B
Loss functions

A **loss function** in the field of neural networks is a measure to quantify the difference between the expected outcome and the outcome generated by the model. From this function, we can derive the gradients that will be used to update the weights of a model.

A machine learning model attempts to learn the probability distribution underlying the given dataset, and evaluates the quality of the estimation at each step of the training process using these kind of functions.

B.1 Categorical Crossentropy

Categorical Crossentropy is a loss function that is used in multi-class classification tasks (i.e., tasks where an example can only belong to one out of many possible categories, and the model must decide which one), such as the particle type classification in each event.

Formally, it is designed to quantify the difference between two probability distributions. It is recommended to combine it with a *Softmax* activation function (see A.3).

B.2 Mean Squared Error

The Mean Squared Error (MSE) loss function calculates how close a regression line is to a set of points. To compute this distance, it takes the distances from the points to the regression line (known as the "errors") and squares them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It is called the mean squared error as it is the average of a set of errors.

In our case, this function is used to measure the error in the estimation of the different object features of each event.

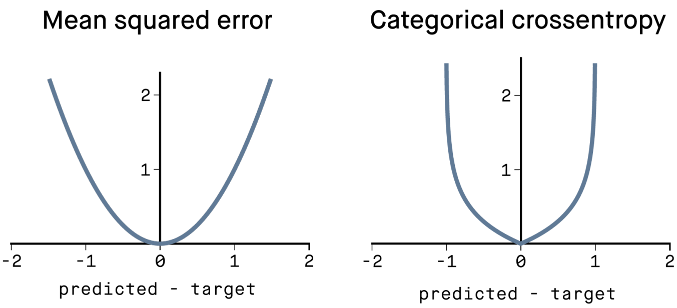


Figure B.1. Comparison of both loss functions used in the presented experiments. [31]

APPENDIX C
Keras layers

As it was explained in section 5.1, **Keras** is one of the two main Python libraries that was used to implement all the models of the performed experiments. In order to allow us to focus on designing the models without having to consider all the implementation details and potentially making mistakes on that side, this library already provides an implementation for the most commonly used layers in ANN architectures.

C.1 *Dense*

Dense layers represent what are known as fully-connected layers. This kind of layer is provided with the number of neurons (nodes) that it should have, and is characterized by the fact that all neurons are connected to each of the neurons of the layer that follows it.

C.2 *BatchNormalization*

Batch Normalization [32] applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1. It must be noted that this layer works differently during training and during inference.

During **training**, the layer normalizes its output using the mean and standard deviation of the current batch of inputs. During **inference**, the layer normalizes its output using a moving average of the mean and standard deviation of the batches it has seen during training.

As such, the layer will only normalize its inputs during inference after having been trained on data that has similar statistics as the inference data.

C.3 *Flatten* and *Reshape*

Sometimes it is convenient for the input to be a one-dimensional array, but the data is not provided in such format. For that purpose, we can use the *Flatten* operation, that converts the size of the data into a 1D array. Conversely, if the data in our model requires to be converted to a multidimensional format at some point, the *Reshape* layer transforms the data into the desired shape.

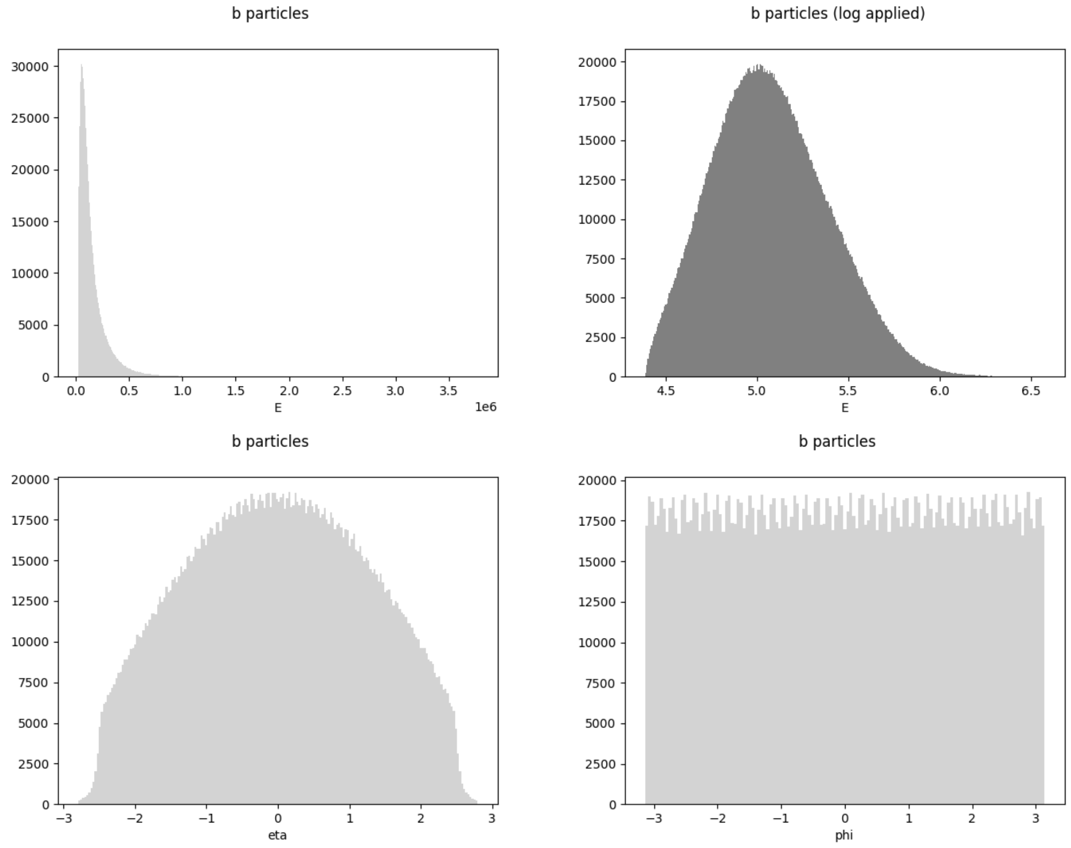
In our case, the data is originally represented in a multidimensional format and we use a *Flatten* layer to convert it into a 1D vector, then it goes through the model, and finally we convert it back into the original format using the *Reshape* layer.

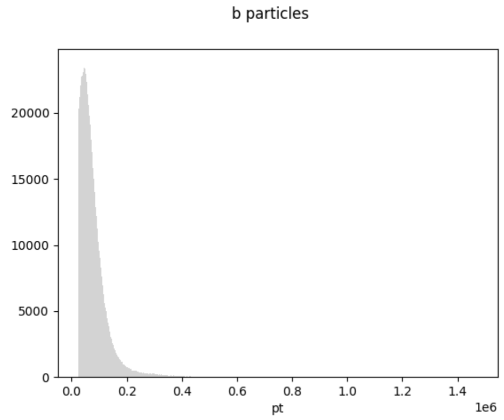
APPENDIX D

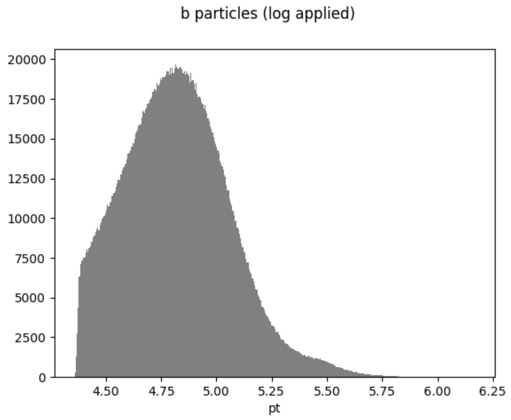
Data distributions

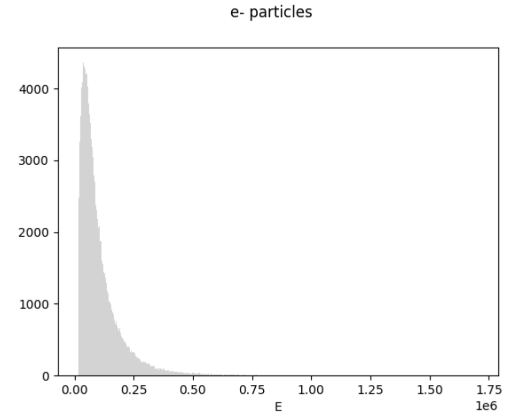
As part of the data analysis process (detailed in section 6.2) that was carried out before beginning with the experimentation, the statistical distributions of the source data were calculated. In this appendix we show the distributions of the different particle features for both the SM and BSM datasets.

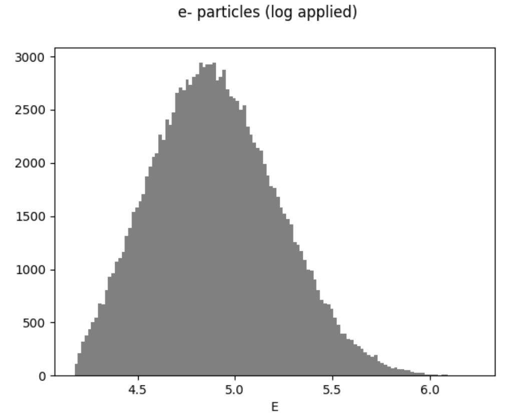
D.1 Standard model

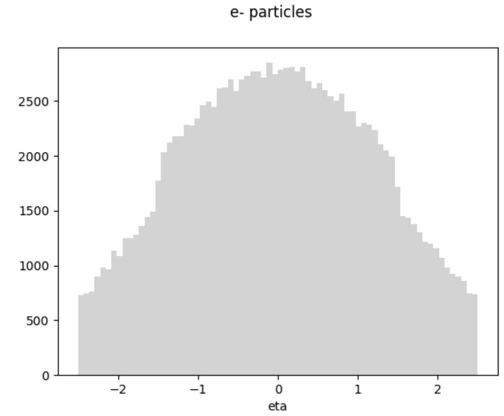


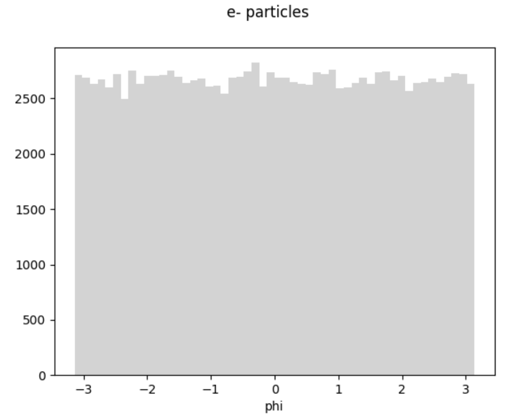

b particles
0
5000
10000
15000
20000
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1e6
pt

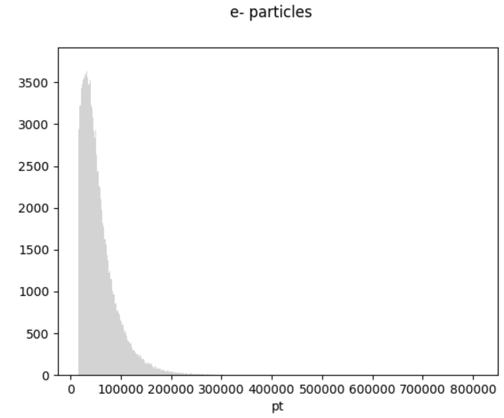

b particles (log applied)
0
2500
5000
7500
10000
12500
15000
17500
20000
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

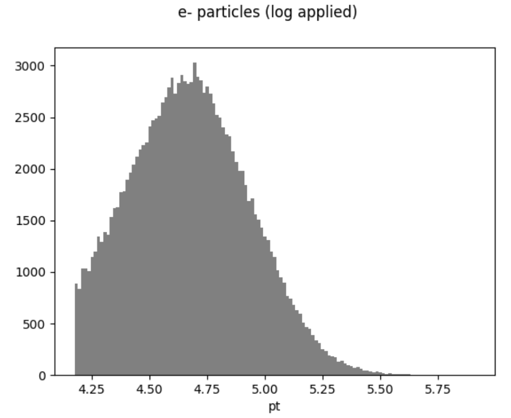

e- particles
0
1000
2000
3000
4000
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
1e6
E

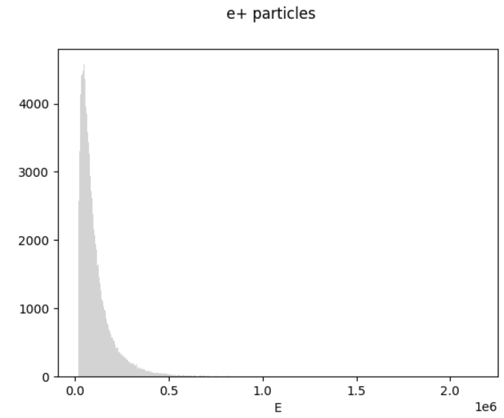

e- particles (log applied)
0
500
1000
1500
2000
2500
3000
4.5
5.0
5.5
6.0
E

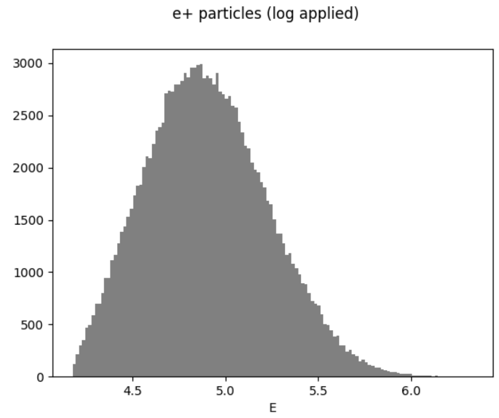

e- particles
0
500
1000
1500
2000
2500
−2
−1
0
1
2
eta

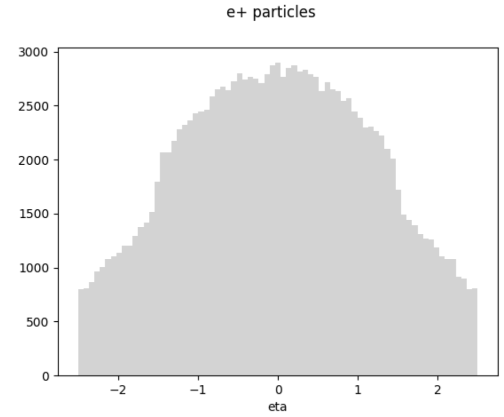

e- particles
0
500
1000
1500
2000
2500
−3
−2
−1
0
1
2
3
phi

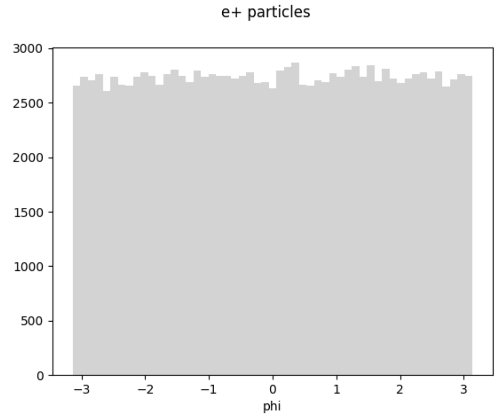

e- particles
0
500
1000
1500
2000
2500
3000
3500
0
100000
200000
300000
400000
500000
600000
700000
800000
pt

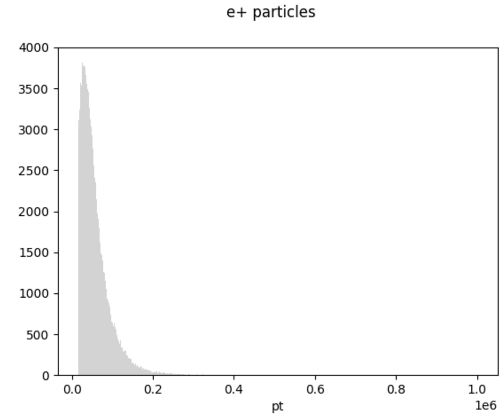

e- particles (log applied)
0
500
1000
1500
2000
2500
3000
4.25
4.50
4.75
5.00
5.25
5.50
5.75
pt

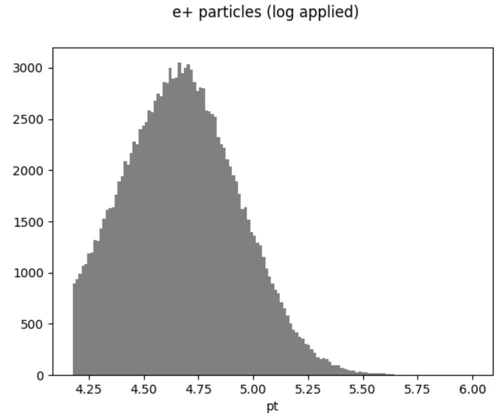

e+ particles

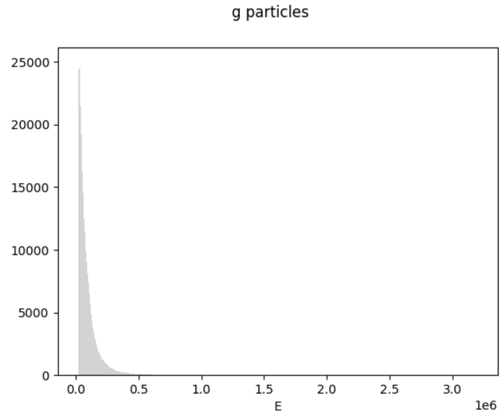

e+ particles (log applied)

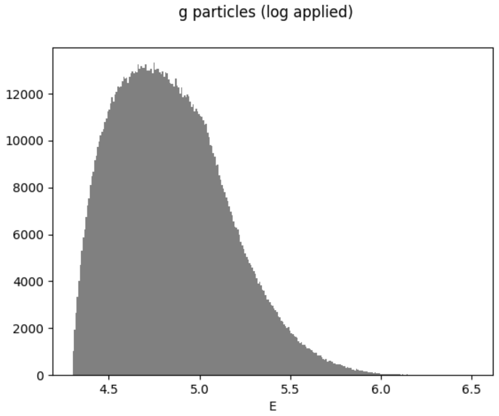

e+ particles

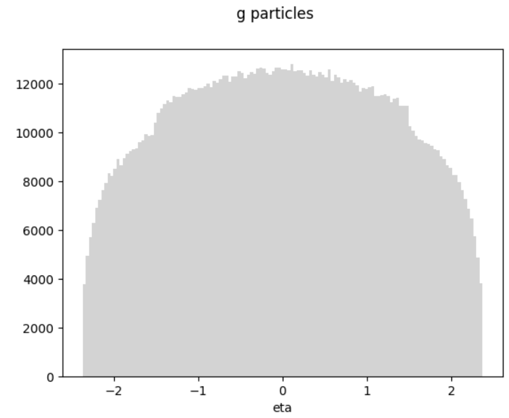

e+ particles

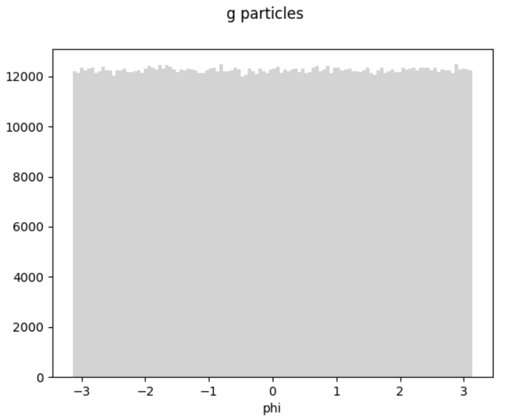

e+ particles

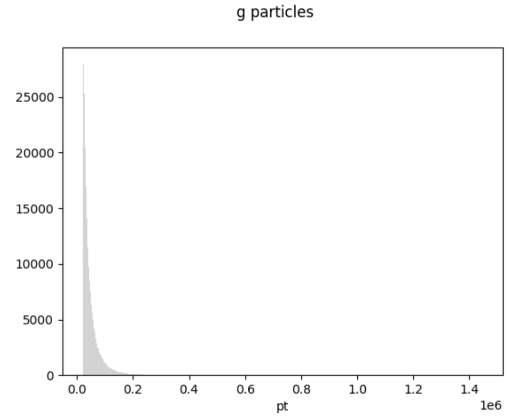

e+ particles (log applied)

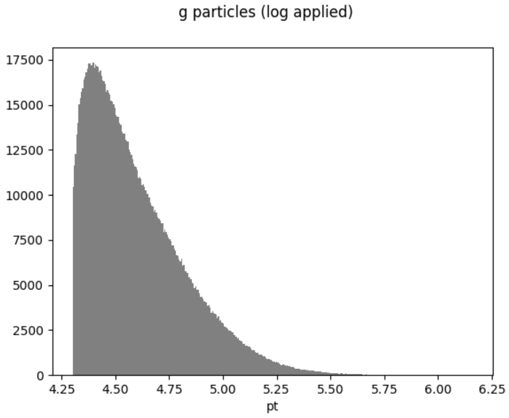

g particles

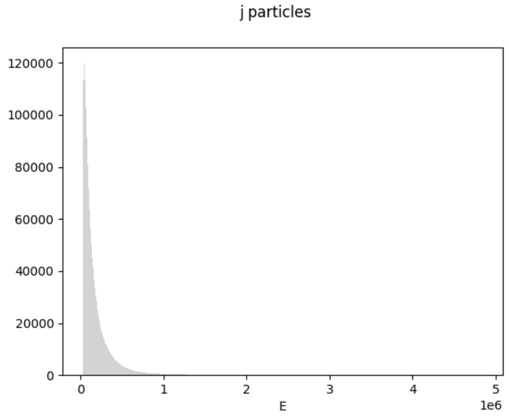

g particles (log applied)

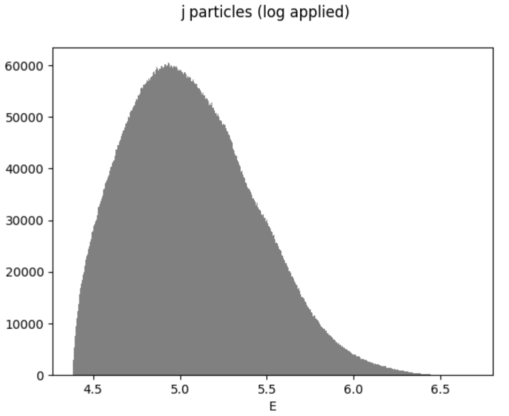

g particles
12000
10000
8000
6000
4000
2000
0
−2
−1
0
1
2
eta

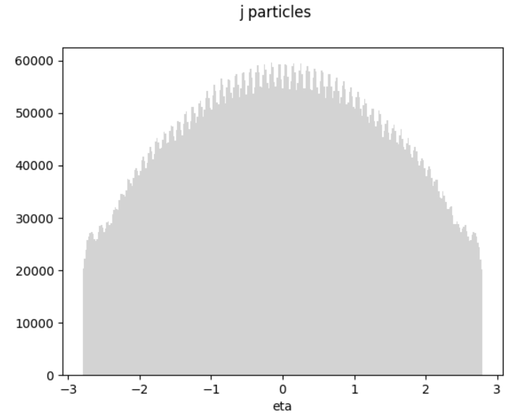

g particles
12000
10000
8000
6000
4000
2000
0
−3
−2
−1
0
1
2
3
phi

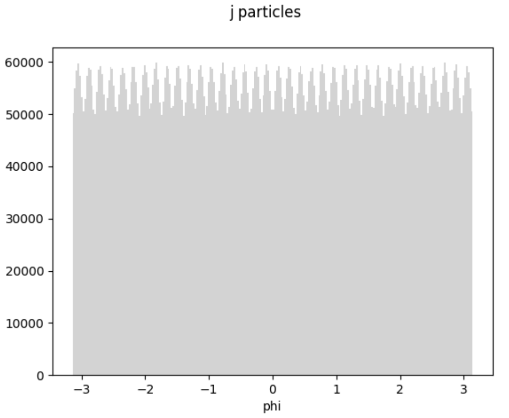

g particles
25000
20000
15000
10000
5000
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
pt
1e6

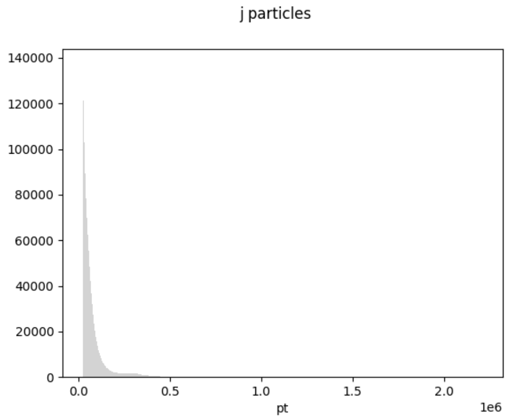

g particles (log applied)
17500
15000
12500
10000
7500
5000
2500
0
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

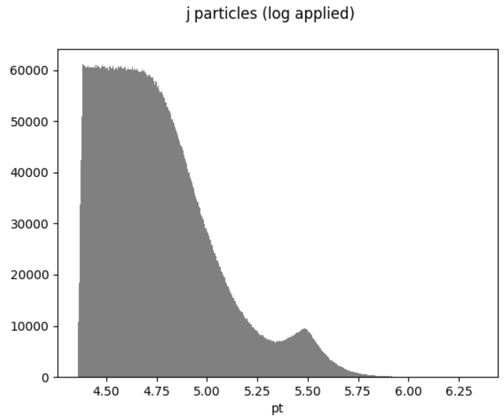

j particles
120000
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
E
1e6

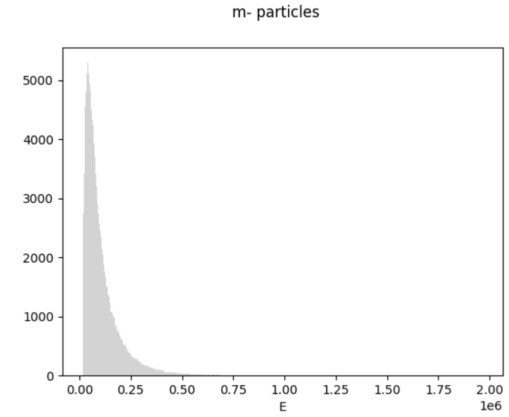

j particles (log applied)
60000
50000
40000
30000
20000
10000
0
4.5
5.0
5.5
6.0
6.5
E

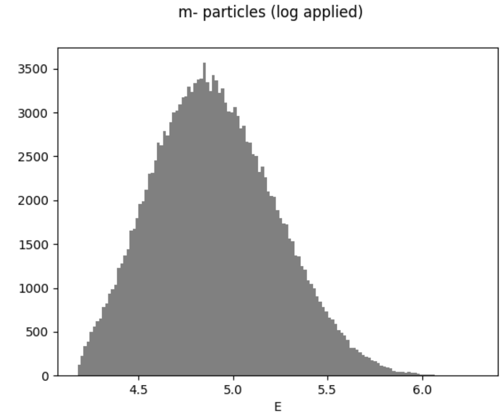

j particles
60000
50000
40000
30000
20000
10000
0
−3
−2
−1
0
1
2
3
eta

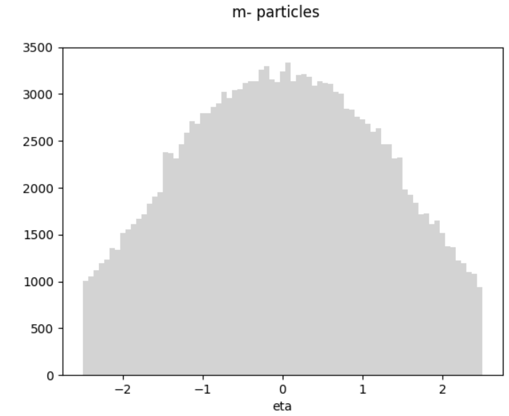

j particles
60000
50000
40000
30000
20000
10000
0
−3
−2
−1
0
1
2
3
phi

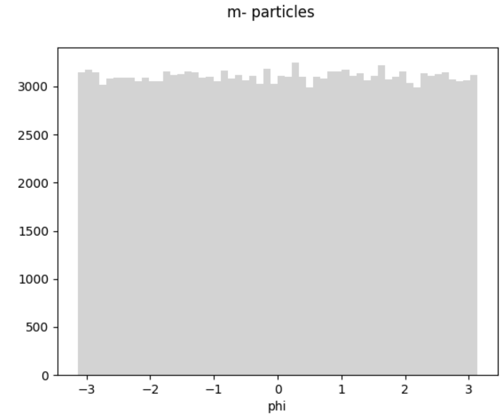

j particles
140000
120000
100000
80000
60000
40000
20000
0
0.0
0.5
1.0
1.5
2.0
1e6
pt

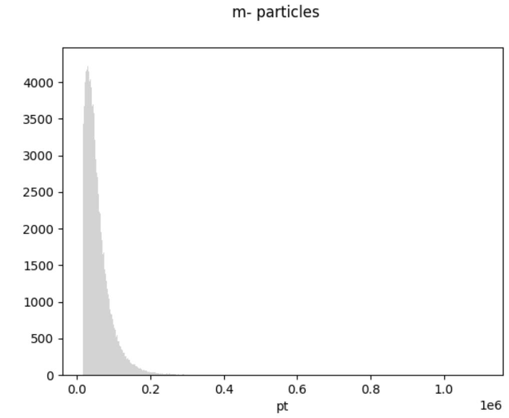

j particles (log applied)
60000
50000
40000
30000
20000
10000
0
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

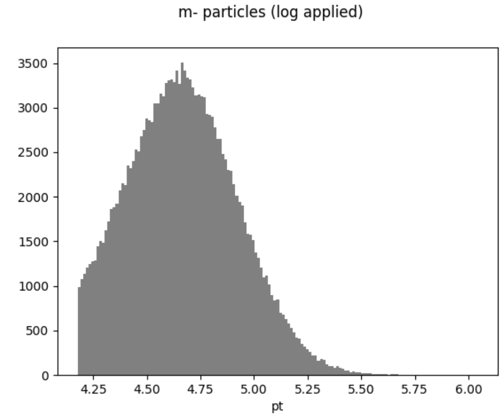

m- particles
5000
4000
3000
2000
1000
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
1e6
E


m- particles (log applied)
3500
3000
2500
2000
1500
1000
500
0
4.5
5.0
5.5
6.0
E


m- particles
3500
3000
2500
2000
1500
1000
500
0
−2
−1
0
1
2
eta


m- particles
3000
2500
2000
1500
1000
500
0
−3
−2
−1
0
1
2
3
phi


m- particles
4000
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
1e6
pt


m- particles (log applied)
3500
3000
2500
2000
1500
1000
500
0
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
pt

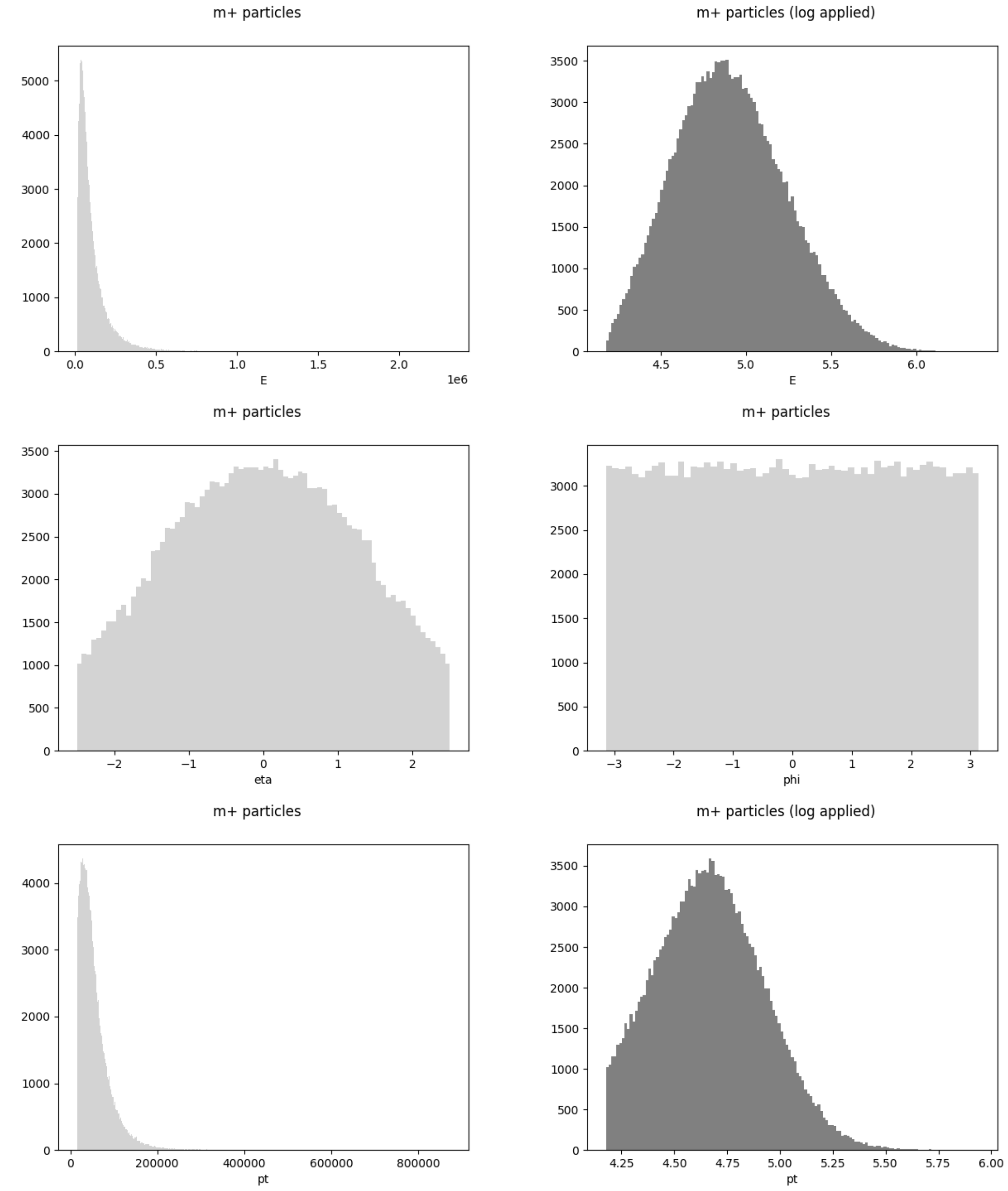
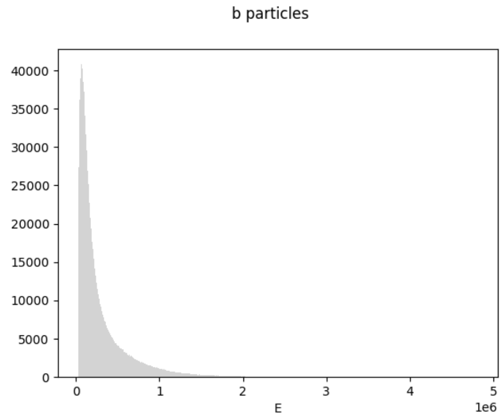
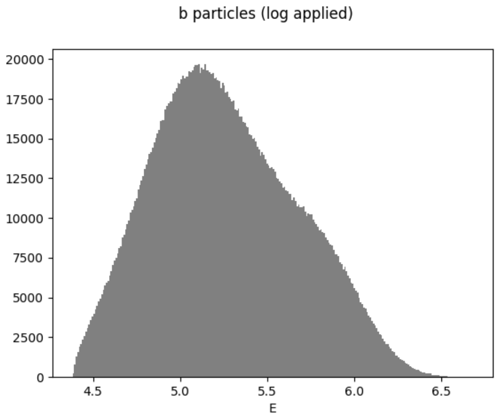
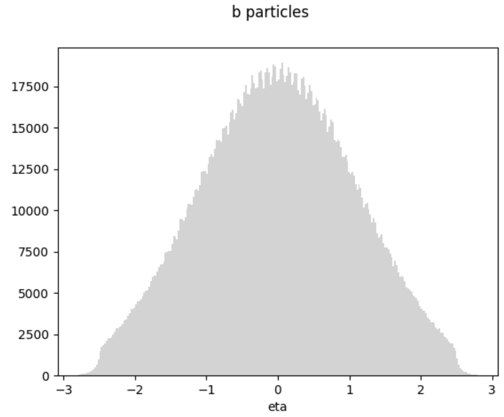


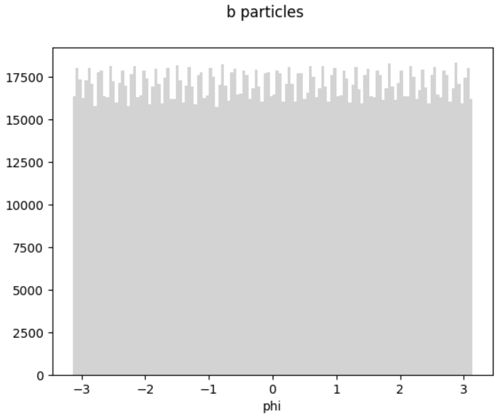
Figure D.1. Data distributions per object type and feature in the whole SM dataset.

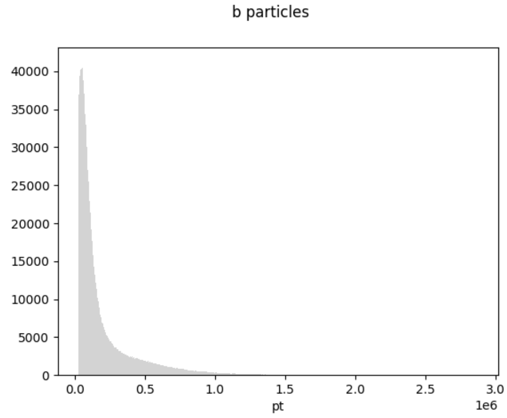
D.2 Beyond the Standard Model

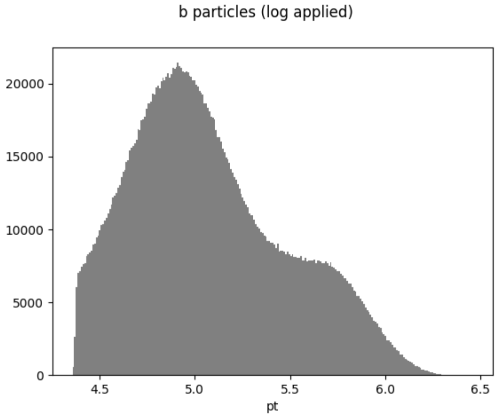


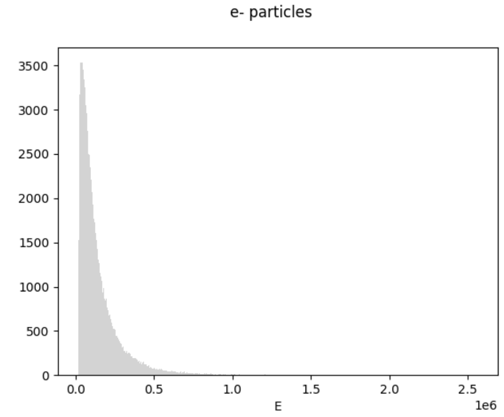


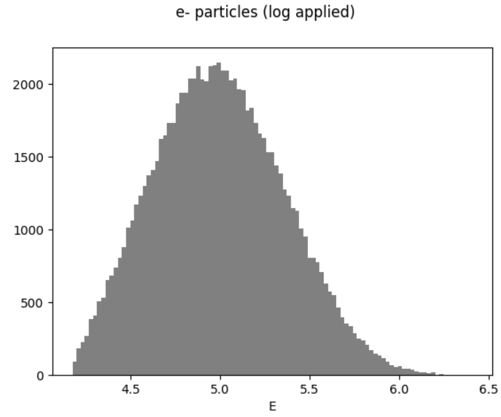


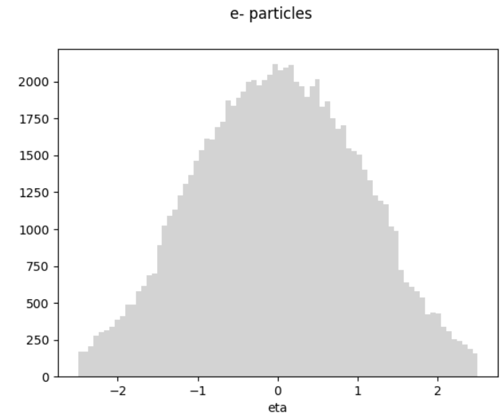


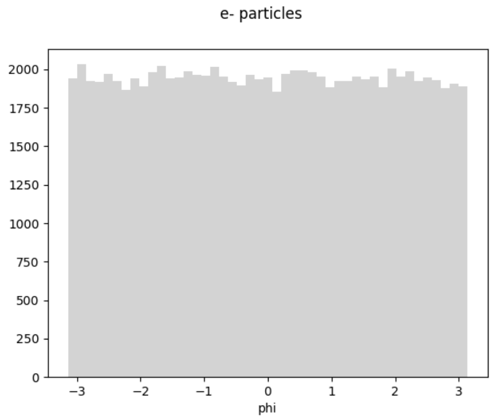


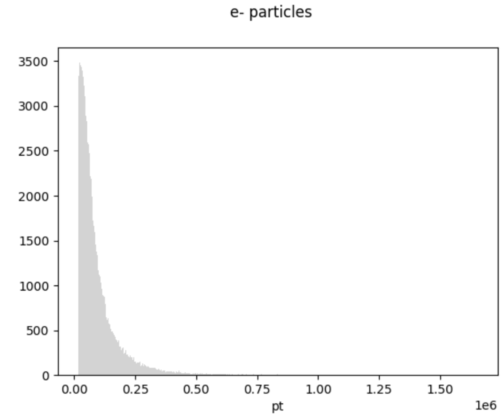


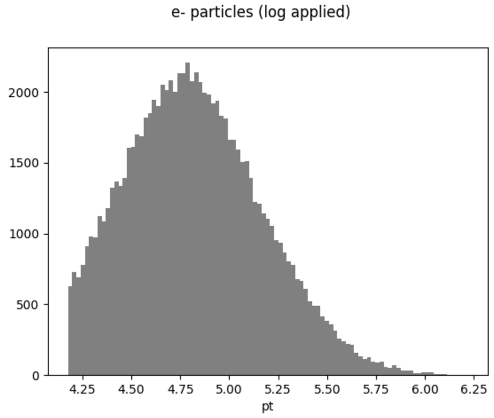


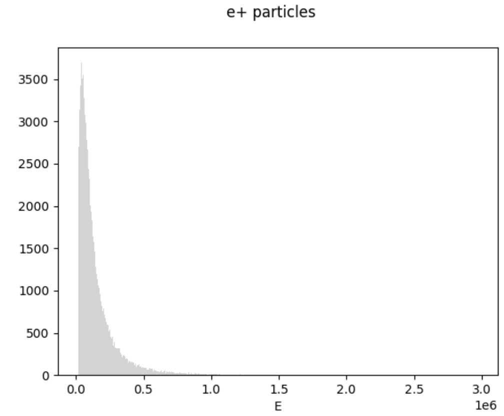


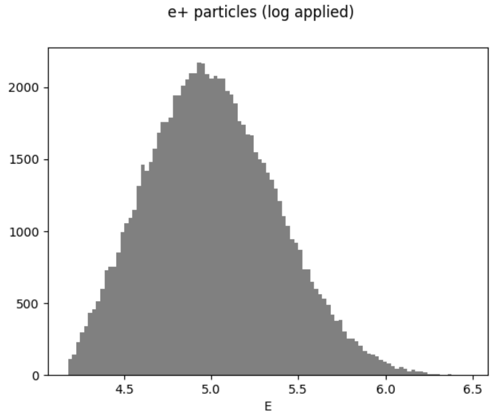

e- particles
0
250
500
750
1000
1250
1500
1750
2000
−2
−1
0
1
2
eta

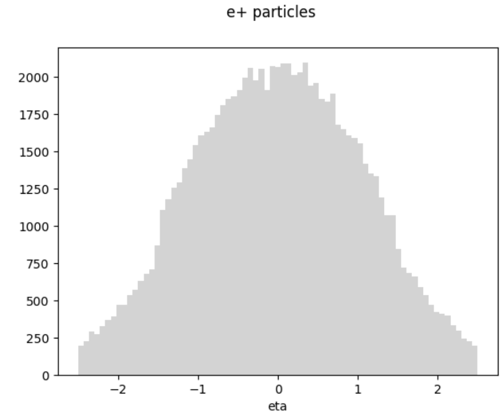

e- particles
0
250
500
750
1000
1250
1500
1750
2000
−3
−2
−1
0
1
2
3
phi

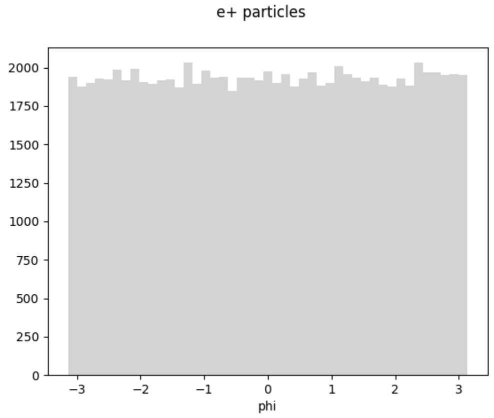

e- particles
0
500
1000
1500
2000
2500
3000
3500
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1e6
pt

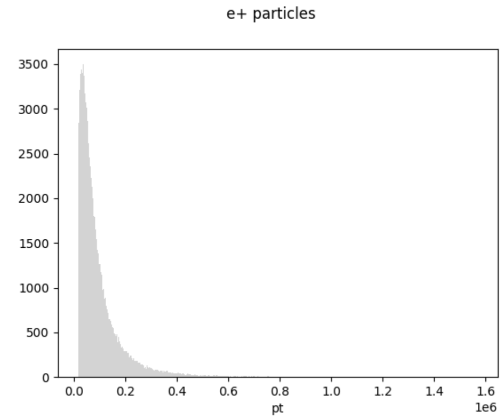

e- particles (log applied)
0
500
1000
1500
2000
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

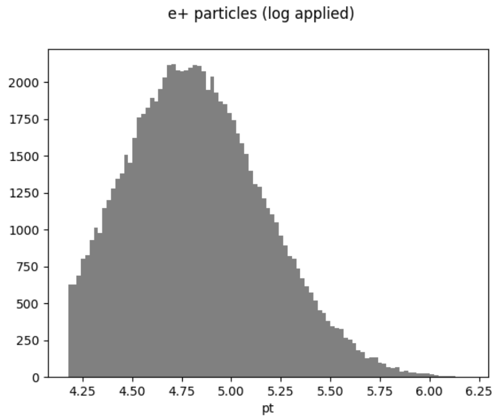

e+ particles
0
500
1000
1500
2000
2500
3000
3500
0.0
0.5
1.0
1.5
2.0
2.5
3.0
1e6
E

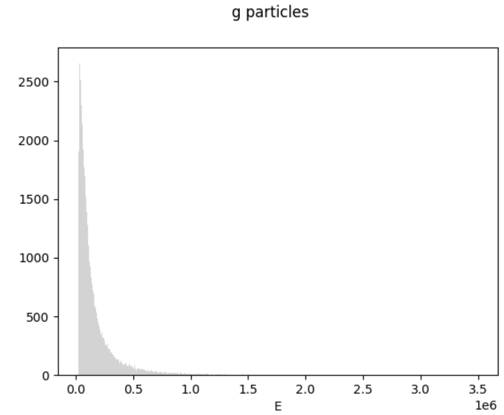

e+ particles (log applied)
0
500
1000
1500
2000
4.5
5.0
5.5
6.0
6.5
E

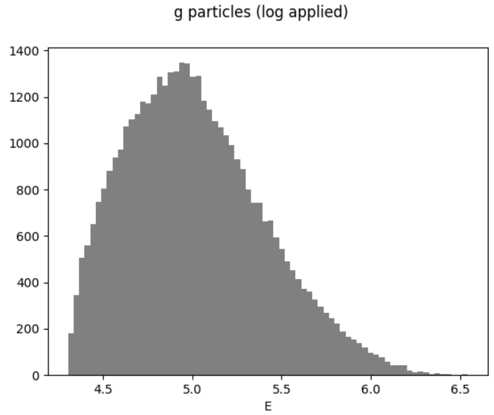

e+ particles
0
250
500
750
1000
1250
1500
1750
2000
−2
−1
0
1
2
eta

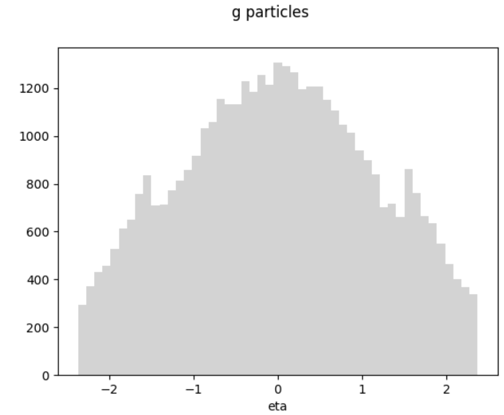

e+ particles
0
250
500
750
1000
1250
1500
1750
2000
−3
−2
−1
0
1
2
3
phi

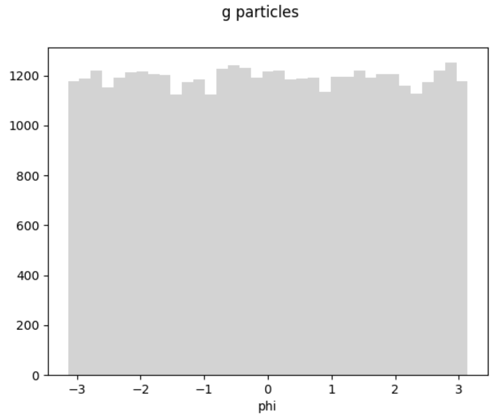

e+ particles
3500
3000
2500
2000
1500
1000
500
0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
pt
1e6

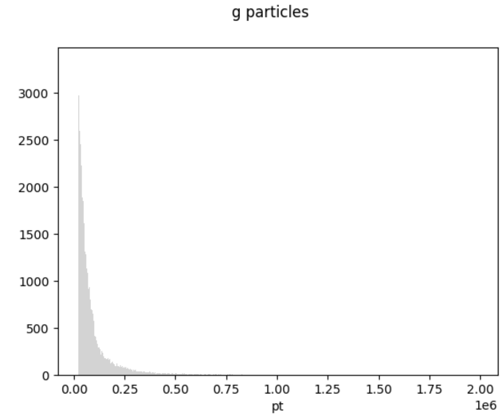

e+ particles (log applied)
2000
1750
1500
1250
1000
750
500
250
0
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

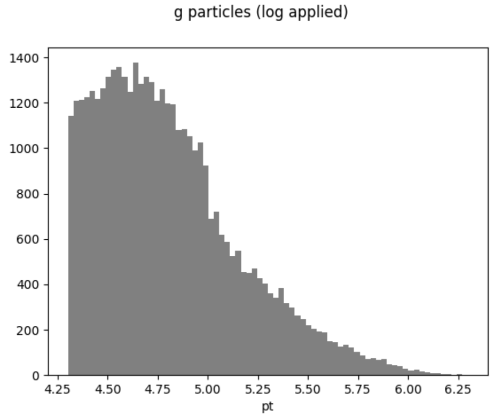

g particles
2500
2000
1500
1000
500
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
E
1e6

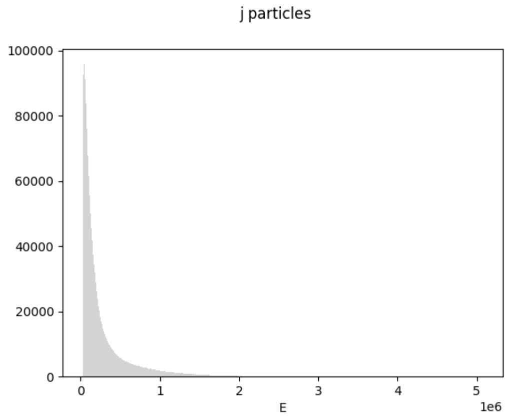

g particles (log applied)
1400
1200
1000
800
600
400
200
0
4.5
5.0
5.5
6.0
6.5
E

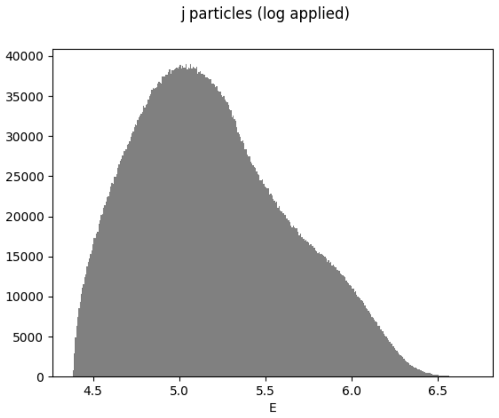

g particles
1200
1000
800
600
400
200
0
−2
−1
0
1
2
eta

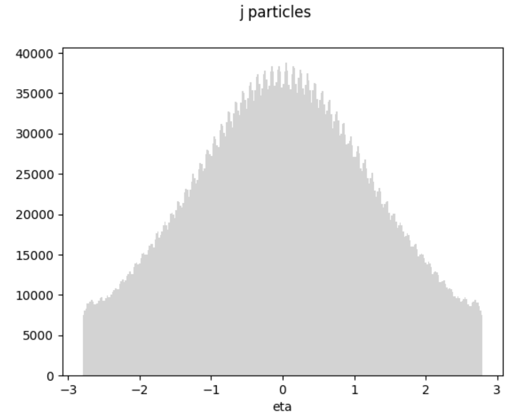

g particles
1200
1000
800
600
400
200
0
−3
−2
−1
0
1
2
3
phi

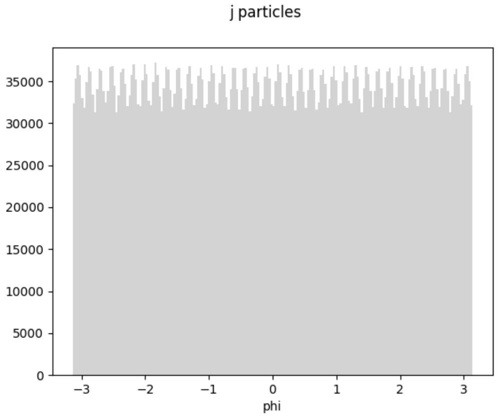

g particles
3000
2500
2000
1500
1000
500
0
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
pt
1e6

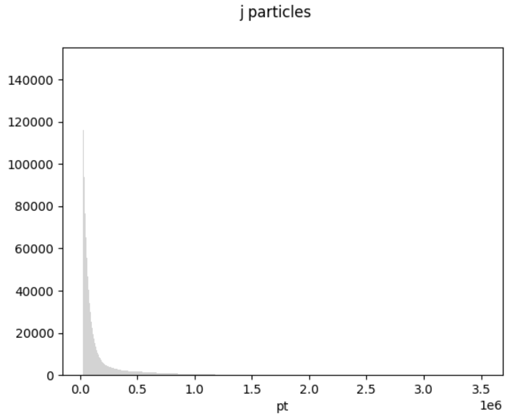

g particles (log applied)
1400
1200
1000
800
600
400
200
0
4.25
4.50
4.75
5.00
5.25
5.50
5.75
6.00
6.25
pt

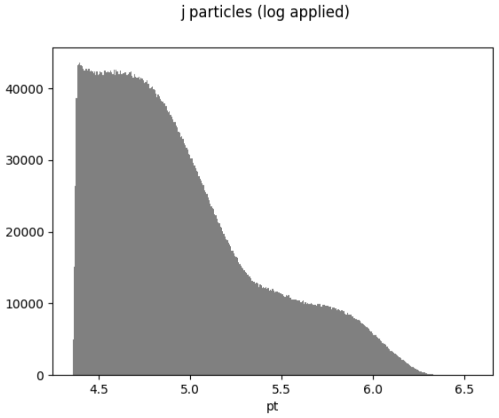

j particles
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
E
1e6

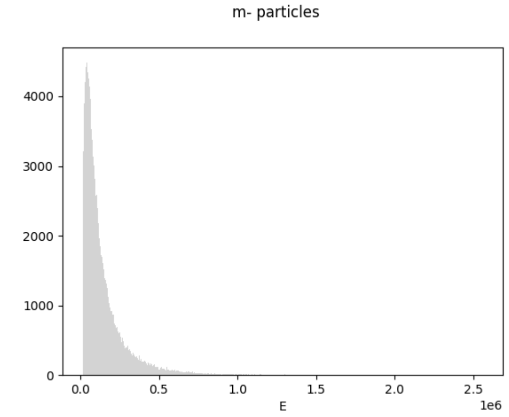

j particles (log applied)
40000
35000
30000
25000
20000
15000
10000
5000
0
4.5
5.0
5.5
6.0
6.5
E

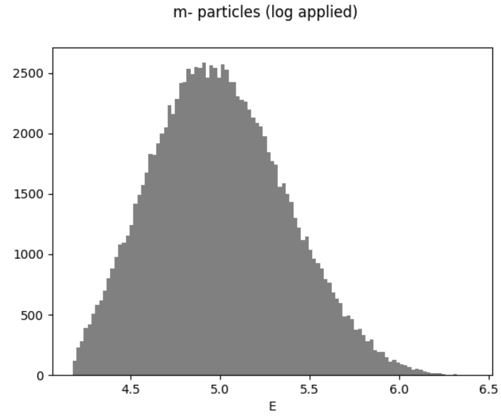

j particles
40000
35000
30000
25000
20000
15000
10000
5000
0
−3
−2
−1
0
1
2
3
eta

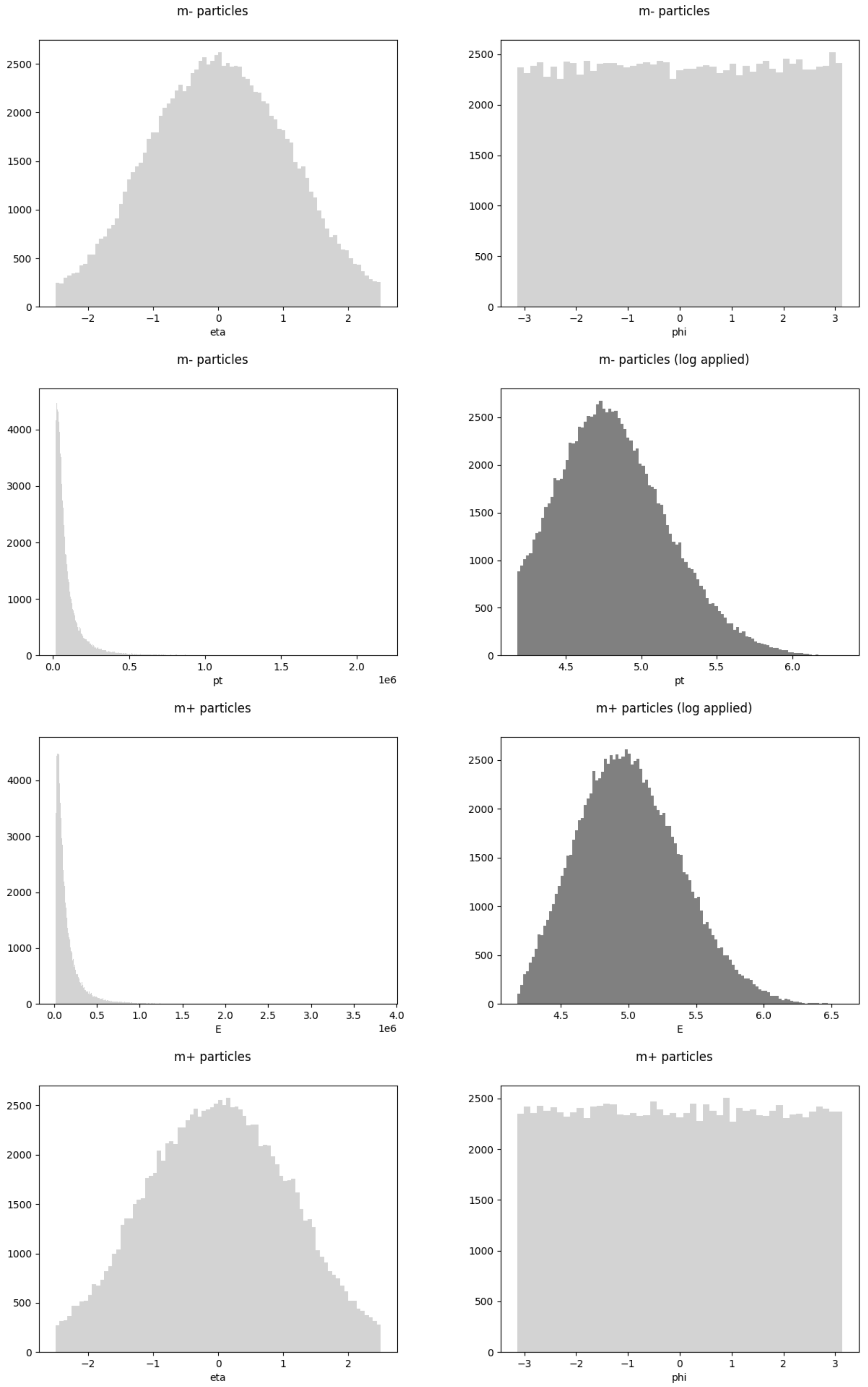

j particles
35000
30000
25000
20000
15000
10000
5000
0
−3
−2
−1
0
1
2
3
phi


j particles
140000
120000
100000
80000
60000
40000
20000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
pt
1e6


j particles (log applied)
40000
30000
20000
10000
0
4.5
5.0
5.5
6.0
6.5
pt


m- particles
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
2.5
E
1e6


m- particles (log applied)
2500
2000
1500
1000
500
0
4.5
5.0
5.5
6.0
6.5
E


m- particles
2500
2000
1500
1000
500
0
−2
−1
0
1
2
eta
m- particles
2500
2000
1500
1000
500
0
−3
−2
−1
0
1
2
3
phi
m- particles
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
1e6
pt
m- particles (log applied)
2500
2000
1500
1000
500
0
4.5
5.0
5.5
6.0
pt
m+ particles
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
1e6
E
m+ particles (log applied)
2500
2000
1500
1000
500
0
4.5
5.0
5.5
6.0
6.5
E
m+ particles
2500
2000
1500
1000
500
0
−2
−1
0
1
2
eta
m+ particles
2500
2000
1500
1000
500
0
−3
−2
−1
0
1
2
3
phi

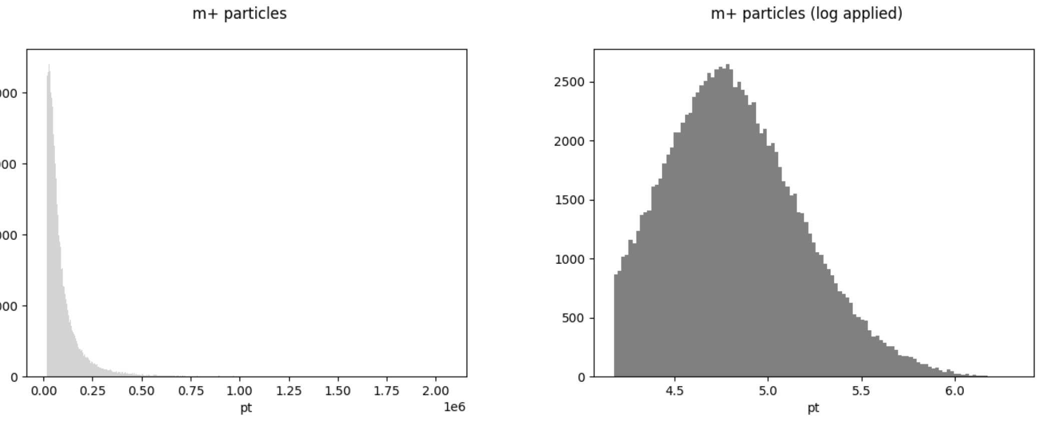


Figure D.2. Data distributions per object type and feature in the whole BSM dataset.

APPENDIX E
Full experiment results

In this appendix we show all the results of most of the experiments that were carried out as a result of this work. Some of them have already been commented in previous chapters.

In some histograms we can see some particle features with an irregular shape that does not fit any known probability distribution. This is due to the lack of presence of some particle types in certain kinds of events (e.g., in *ttbar* events, some experiments generate events that do not have two leptons).

Other histograms show only the original events (in gray), without showing any simulated values (in red). Although this situation could mean that there are no simulated events for that configuration, it actually represents that values for the simulated events are too different when compared to original MC events, so they are not shown when plotted using original events as a base.

E.1 Experiment I

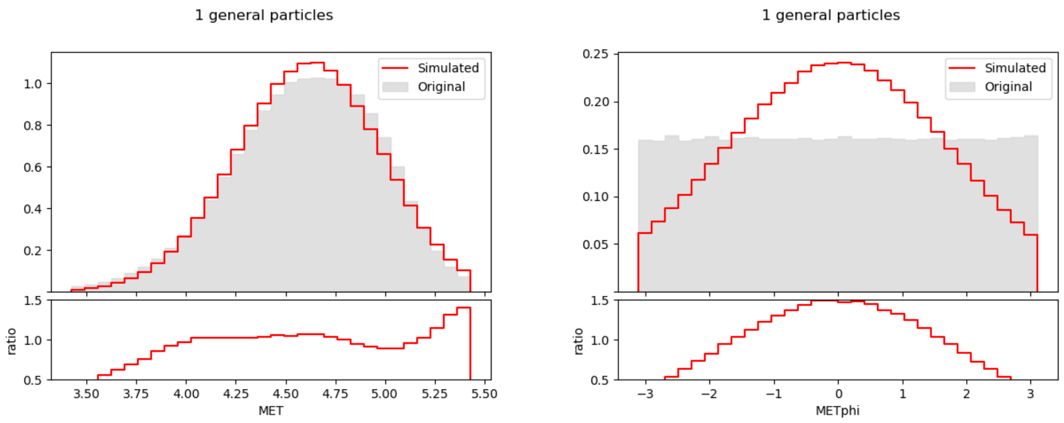


Figure E.1. Event MET and METphi distributions in experiment II ($\beta = 0.01$, $\gamma = 50$).

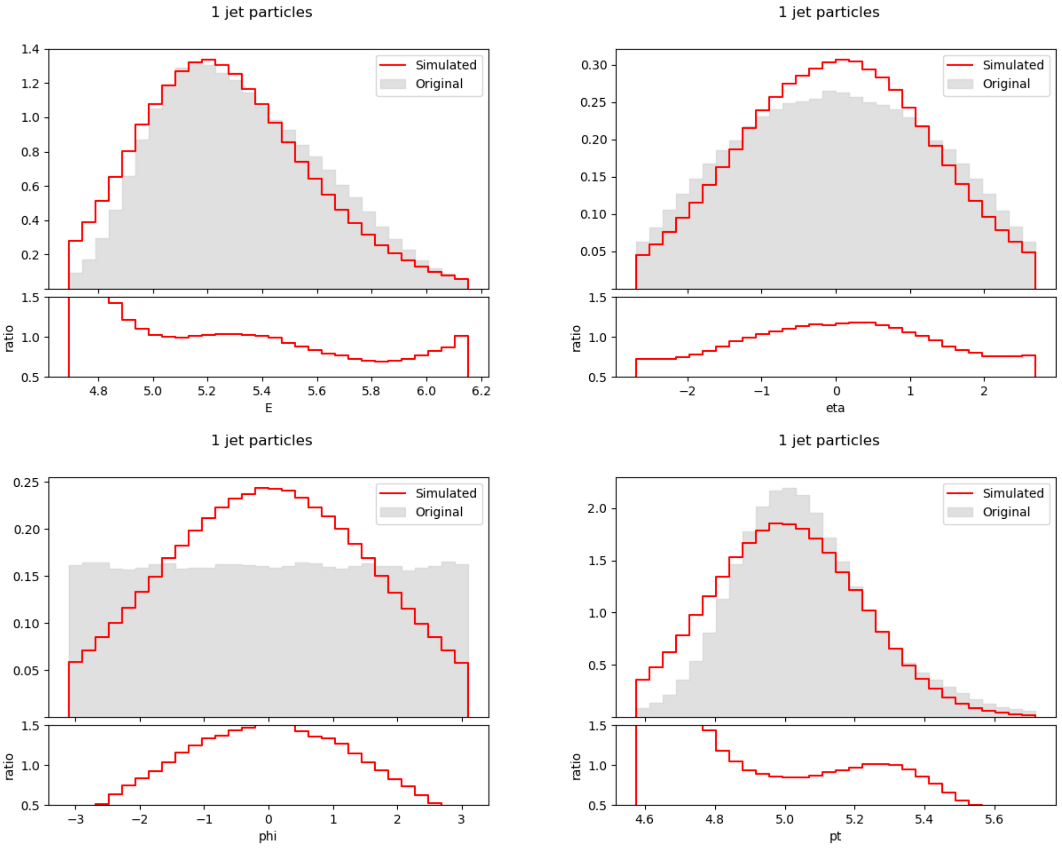


Figure E.2. First jet particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$).

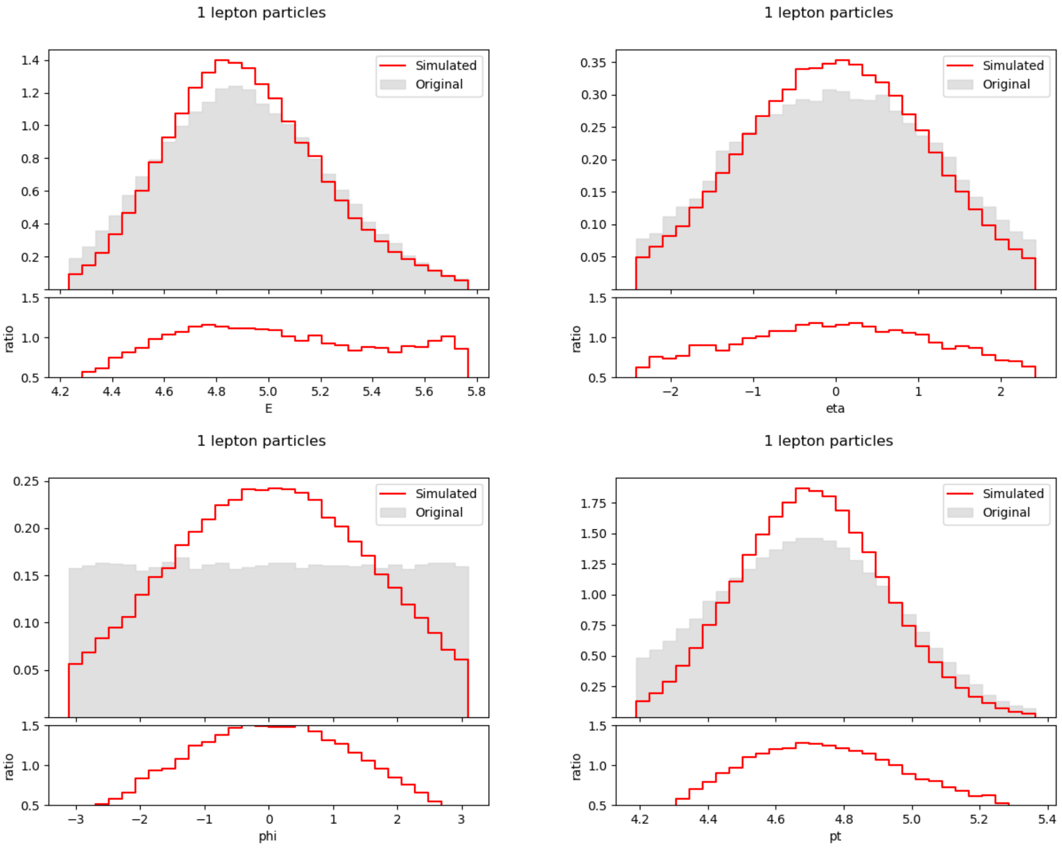


Figure E.3. First lepton particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$).

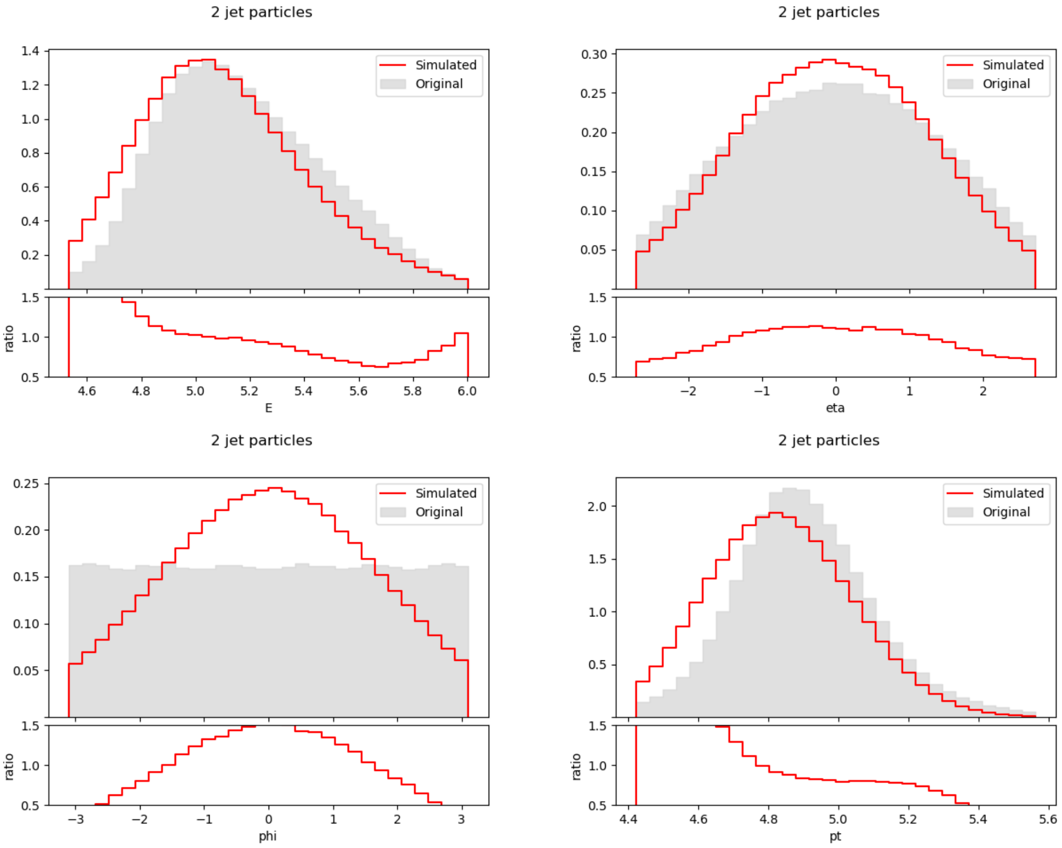


Figure E.4. Second jet particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$).

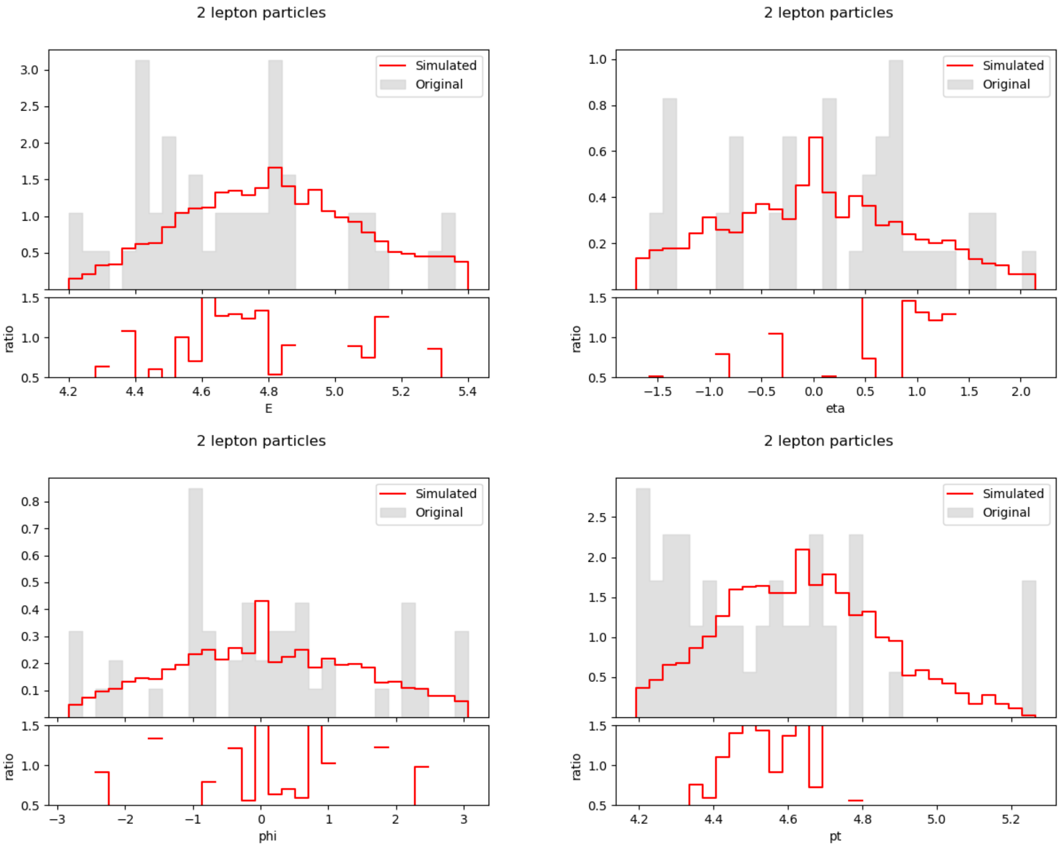


Figure E.5. Second lepton particle feature distributions in experiment II ($\beta = 0.01$, $\gamma = 50$).

E.2 Experiment II

E.2.1. Without BGMMs

$\beta = 0.0$

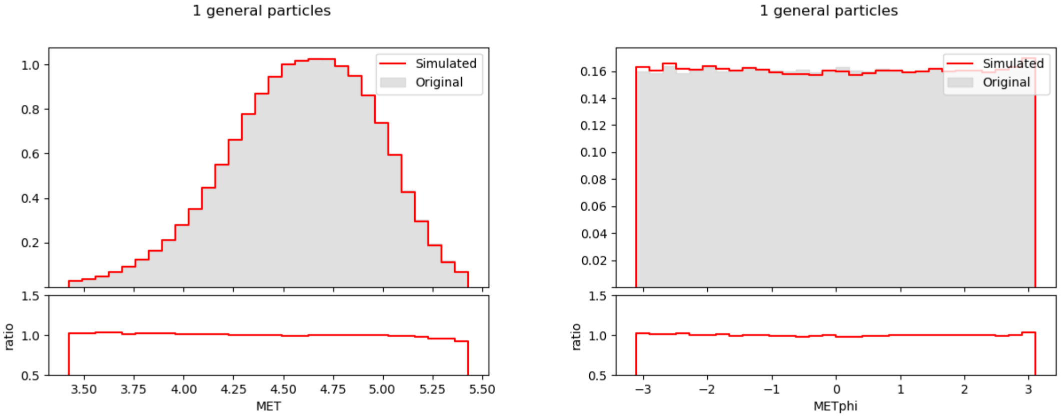


Figure E.6. Event `MET` and `METphi` distributions in experiment II ($\beta = 0.0$).

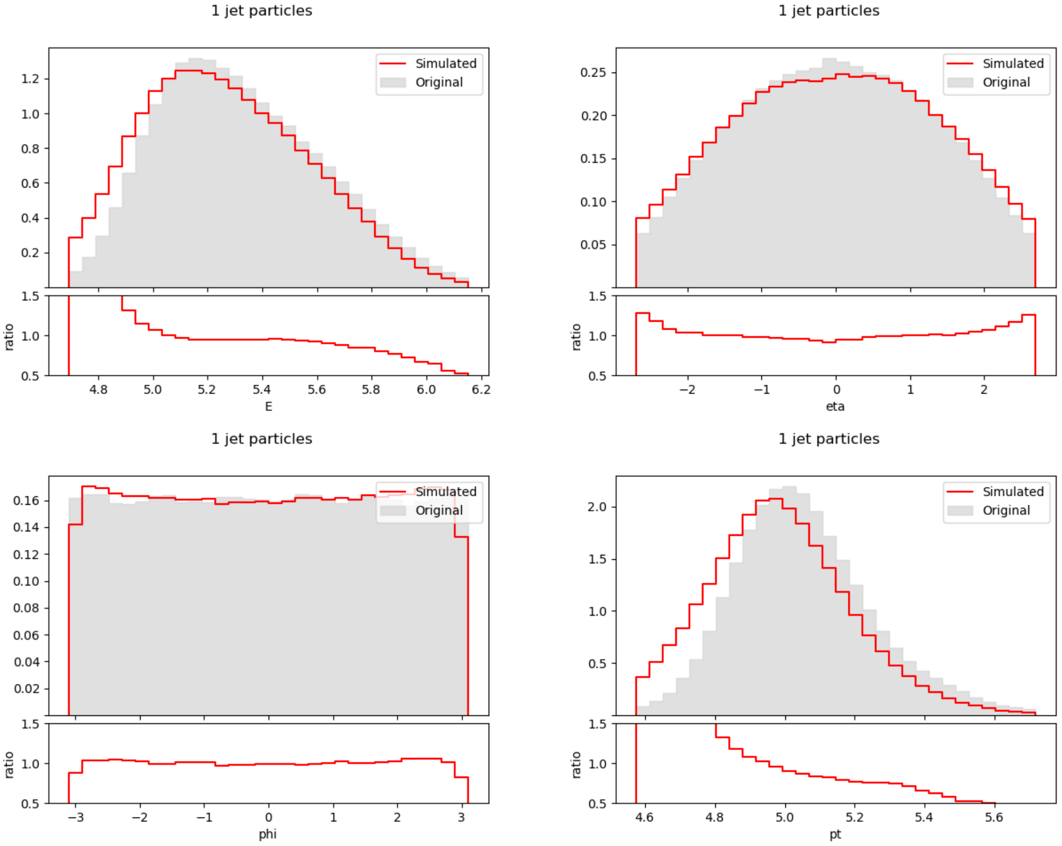


Figure E.7. First jet particle feature distributions in experiment II ($\beta = 0.0$).

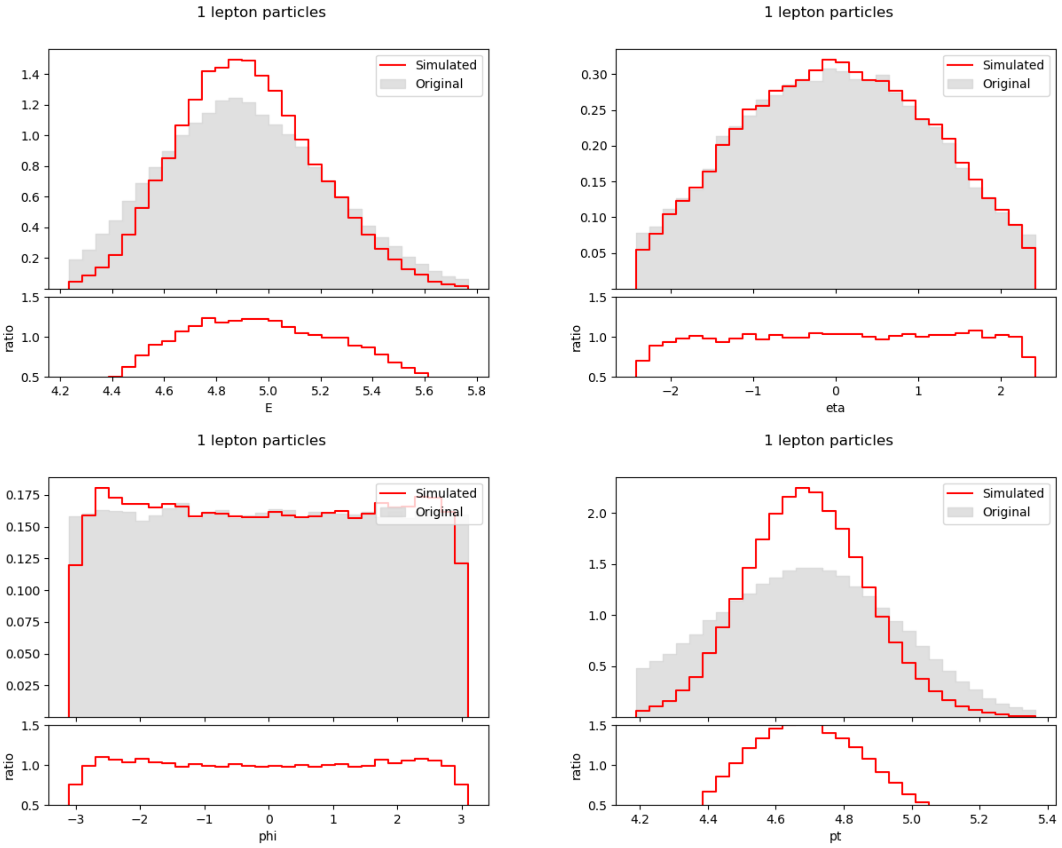


Figure E.8. First lepton particle feature distributions in experiment II ($\beta = 0.0$).

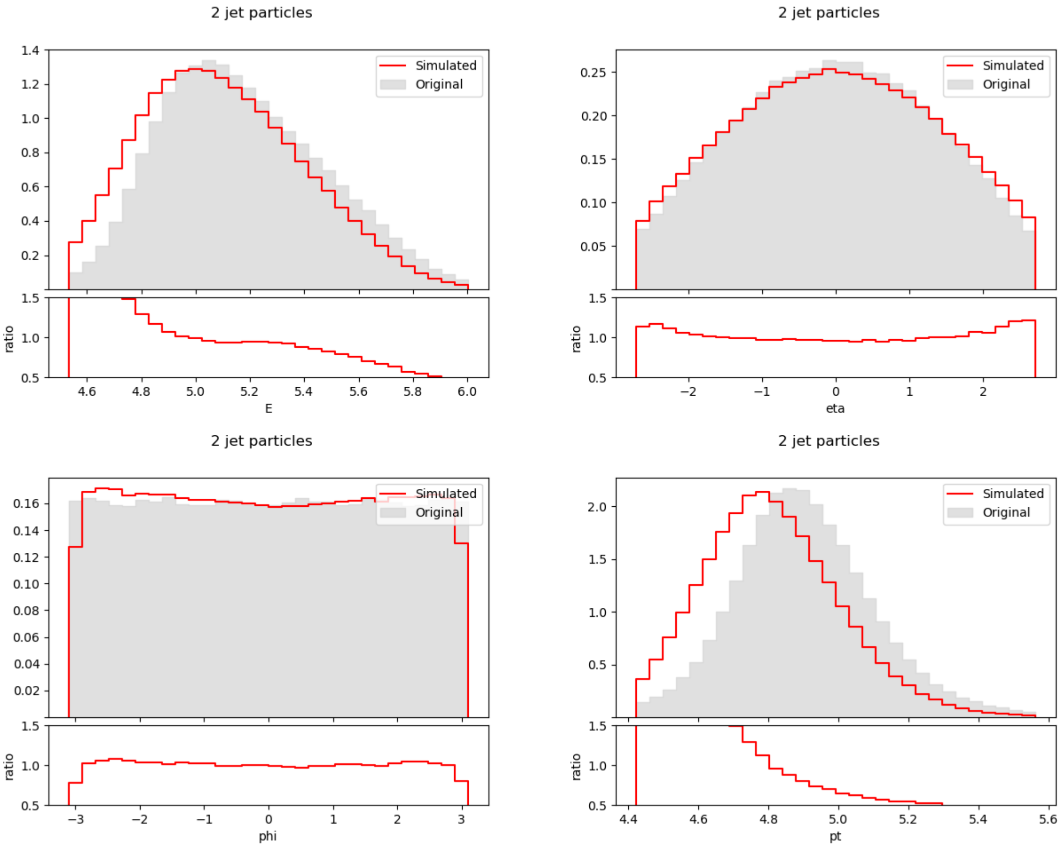


Figure E.9. Second jet particle feature distributions in experiment II ($\beta = 0.0$).

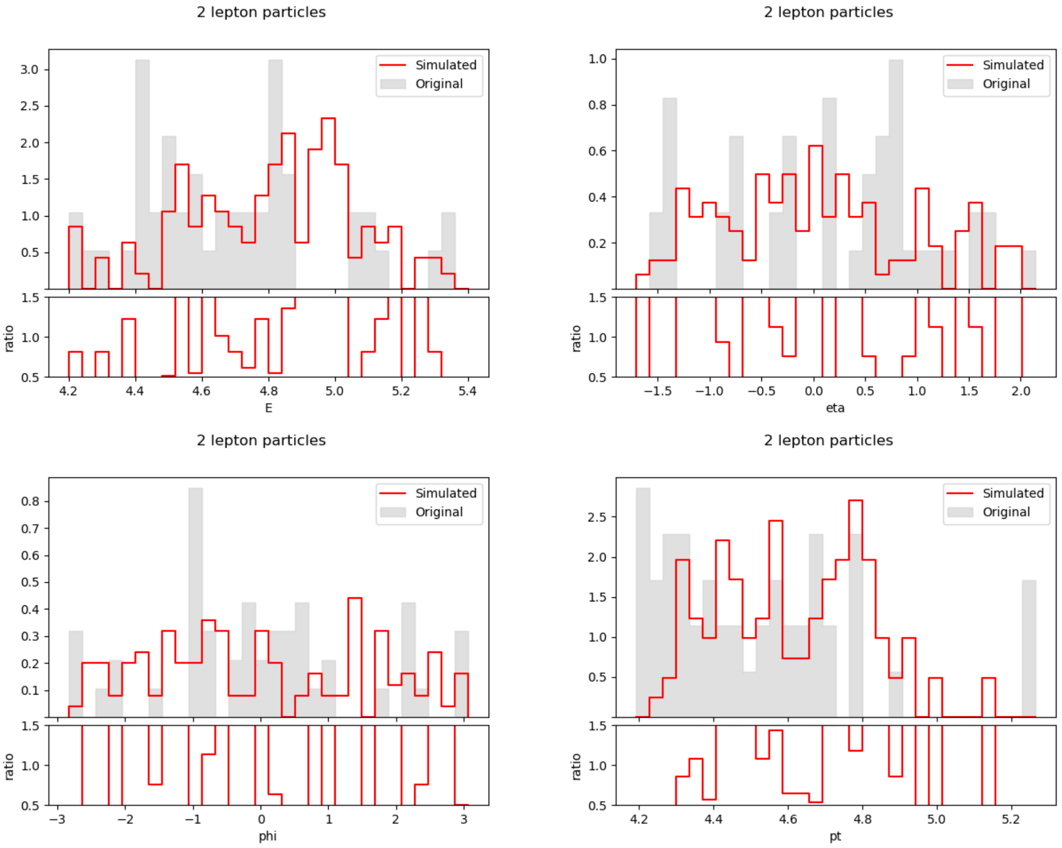


Figure E.10. Second lepton particle feature distributions in experiment II ($\beta = 0.0$).

$\beta = 0.001$

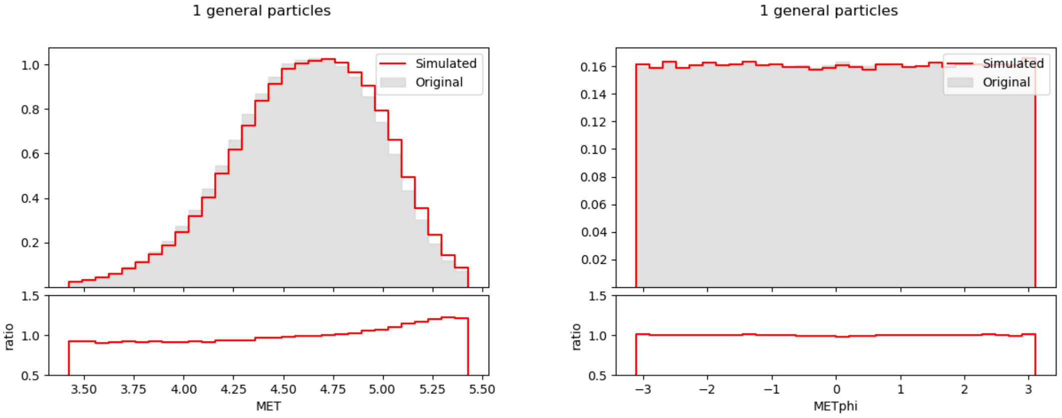


Figure E.11. Event `MET` and `METphi` distributions in experiment II ($\beta = 0.001$).

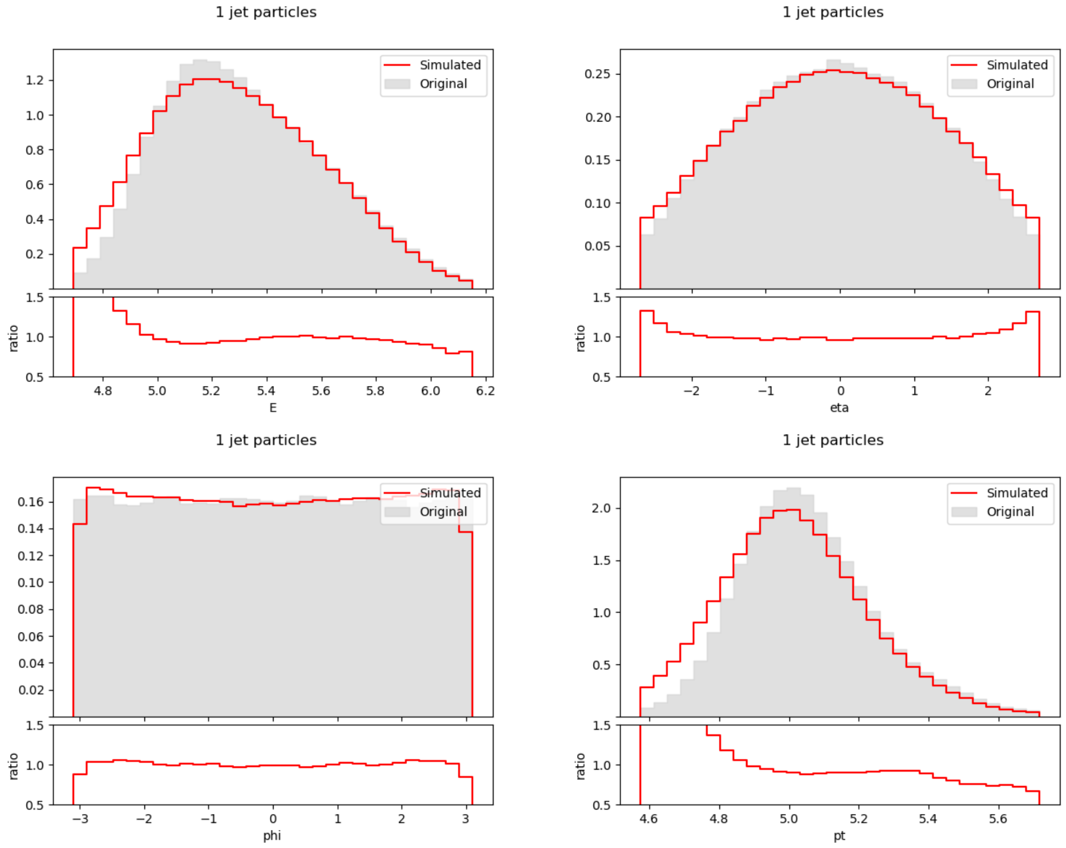


Figure E.12. First jet particle feature distributions in experiment II ($\beta = 0.001$).

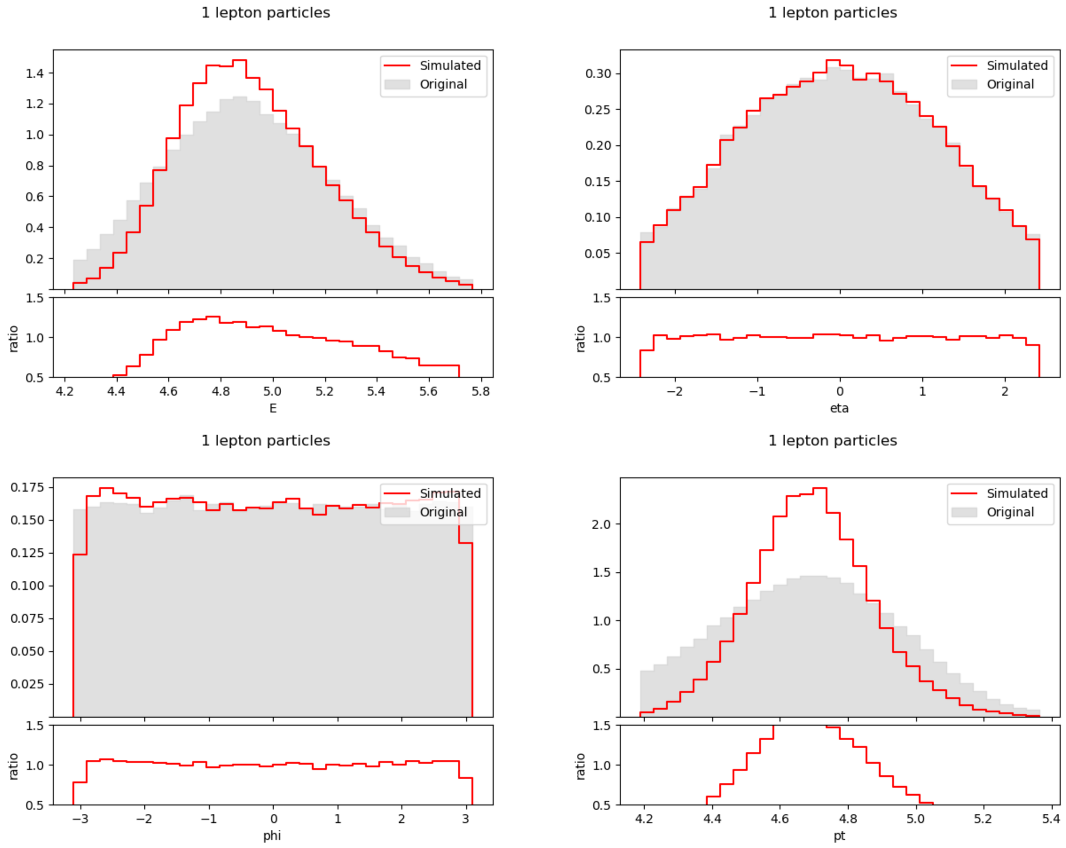


Figure E.13. First lepton particle feature distributions in experiment II ($\beta = 0.001$).

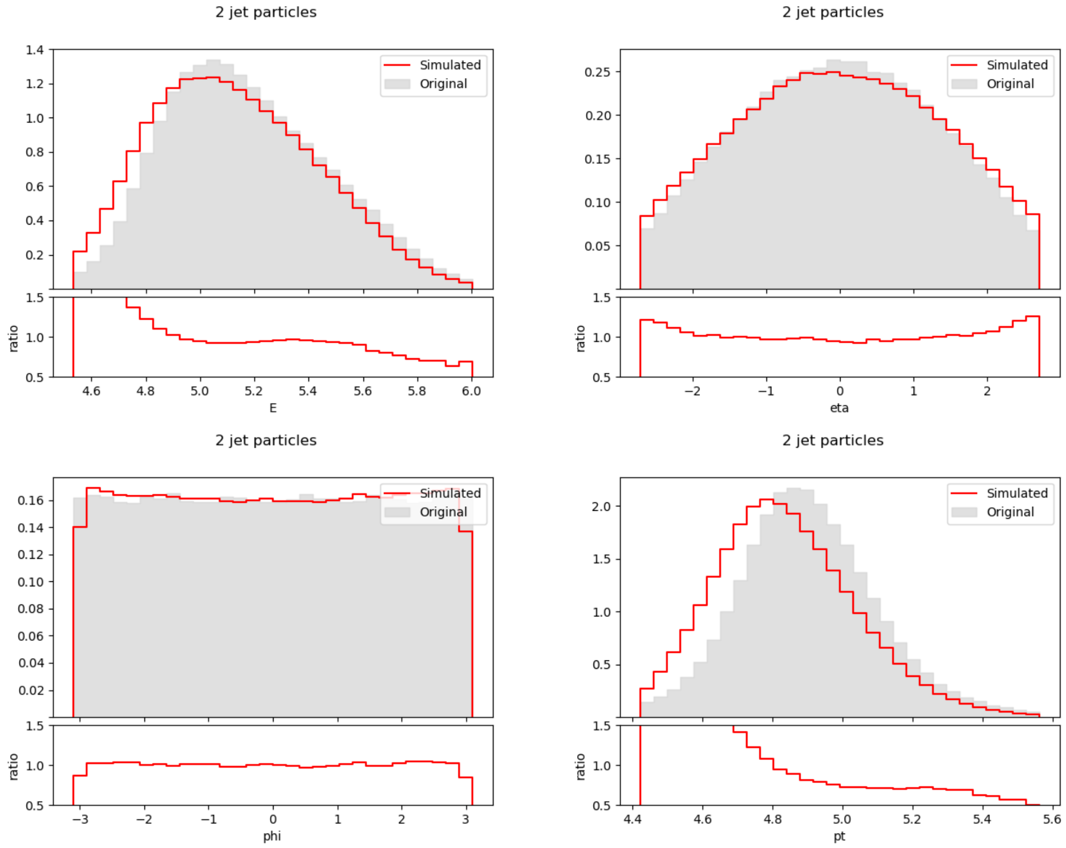


Figure E.14. Second jet particle feature distributions in experiment II ($\beta = 0.001$).

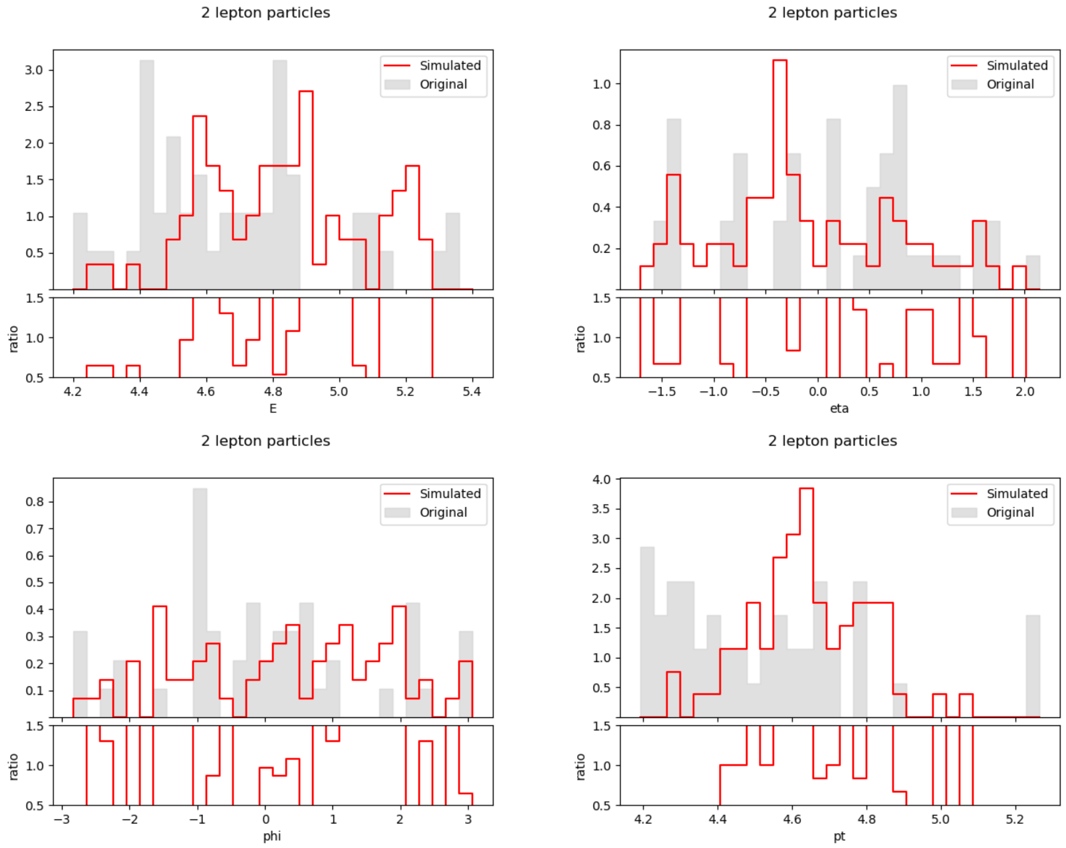


Figure E.15. Second lepton particle feature distributions in experiment II ($\beta = 0.001$).

$\beta = 0.01$

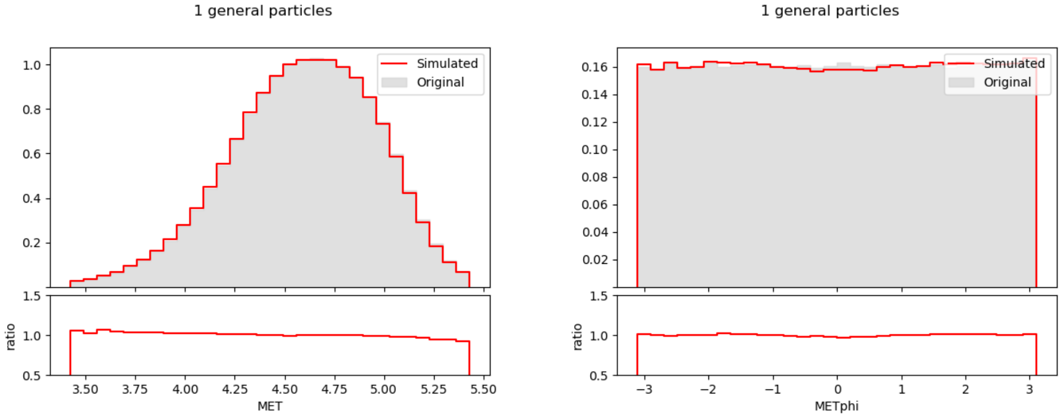


Figure E.16. Event `MET` and `METphi` distributions in experiment II ($\beta = 0.01$).

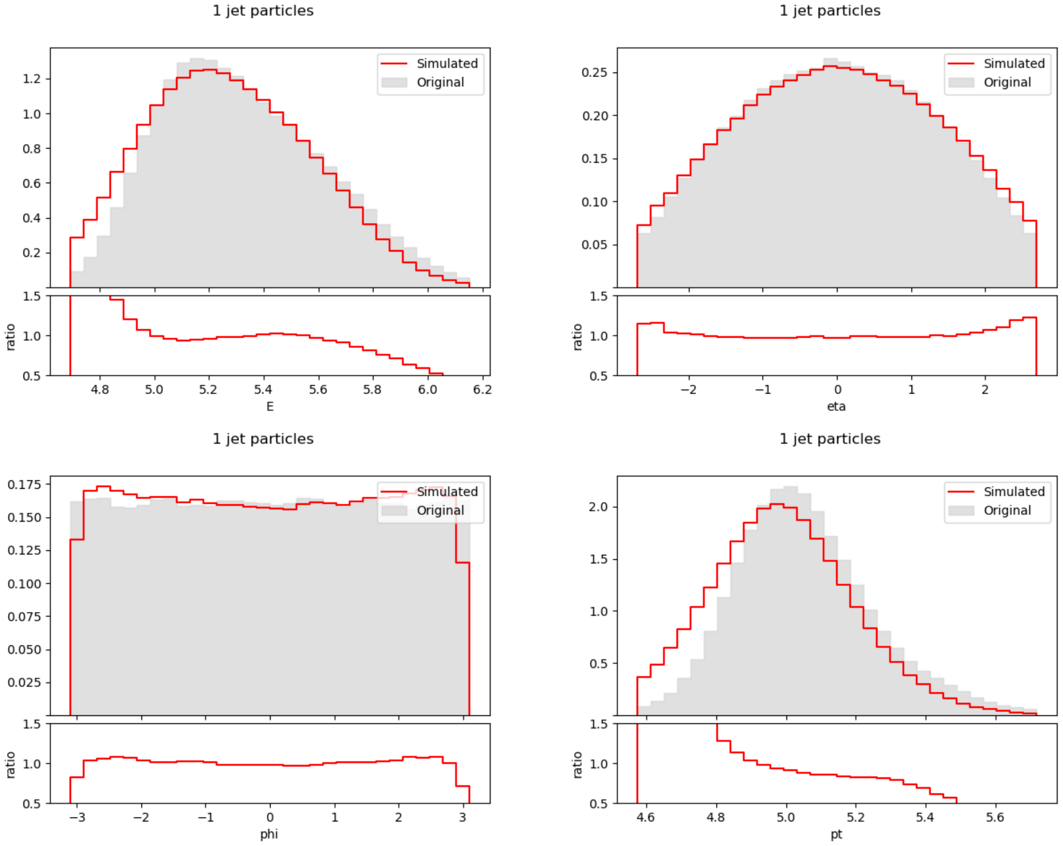


Figure E.17. First jet particle feature distributions in experiment II ($\beta = 0.01$).

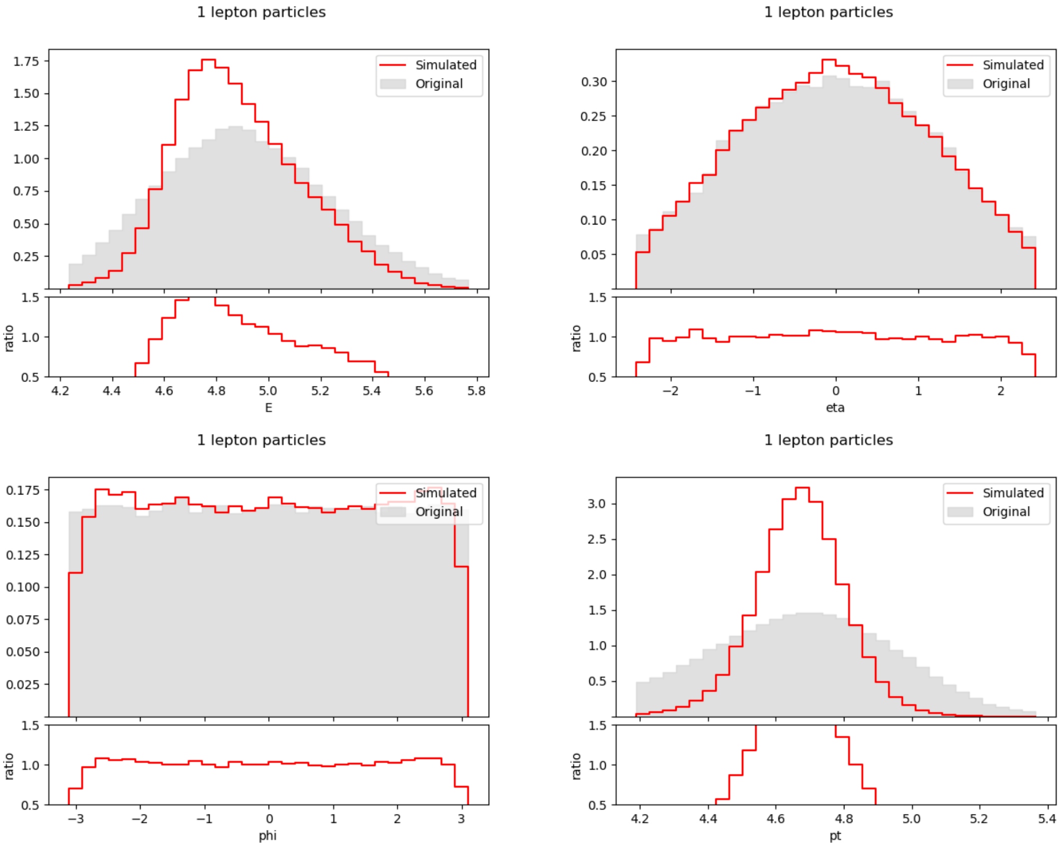


Figure E.18. First lepton particle feature distributions in experiment II ($\beta = 0.01$).

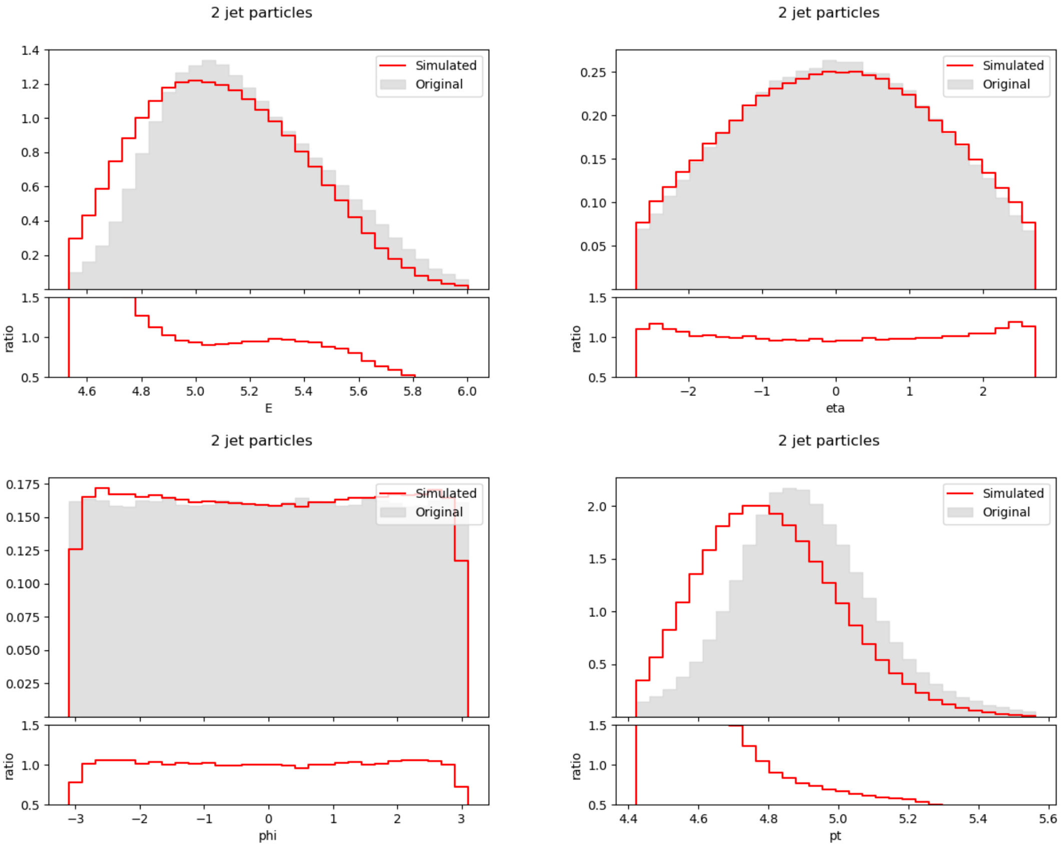


Figure E.19. Second jet particle feature distributions in experiment II ($\beta = 0.01$).

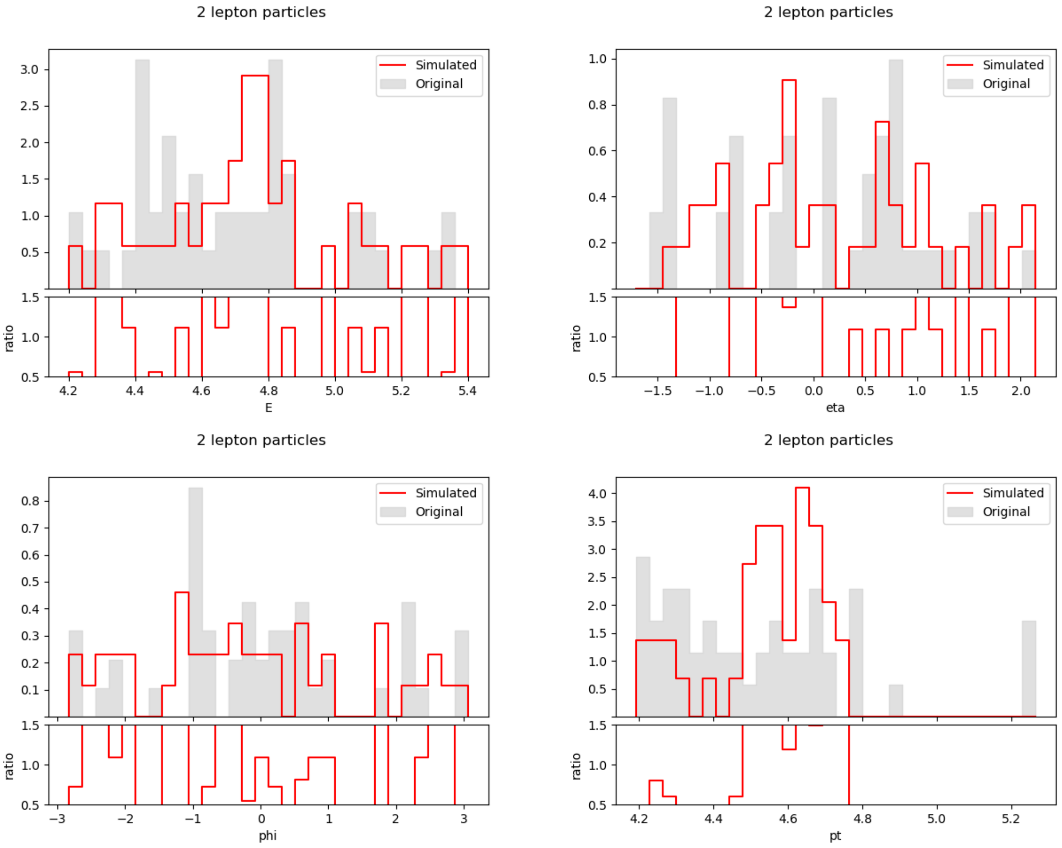


Figure E.20. Second lepton particle feature distributions in experiment II ($\beta = 0.01$).

$\beta = 0.1$

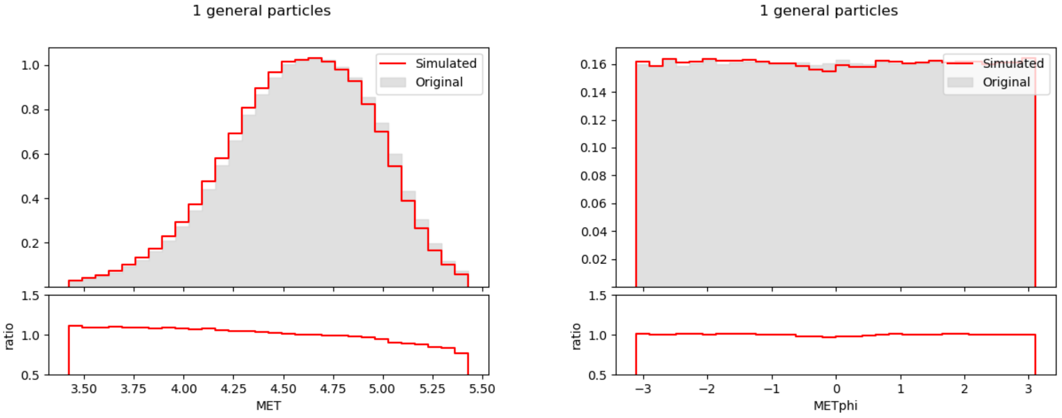


Figure E.21. Event MET and METphi distributions in experiment II ($\beta = 0.1$).

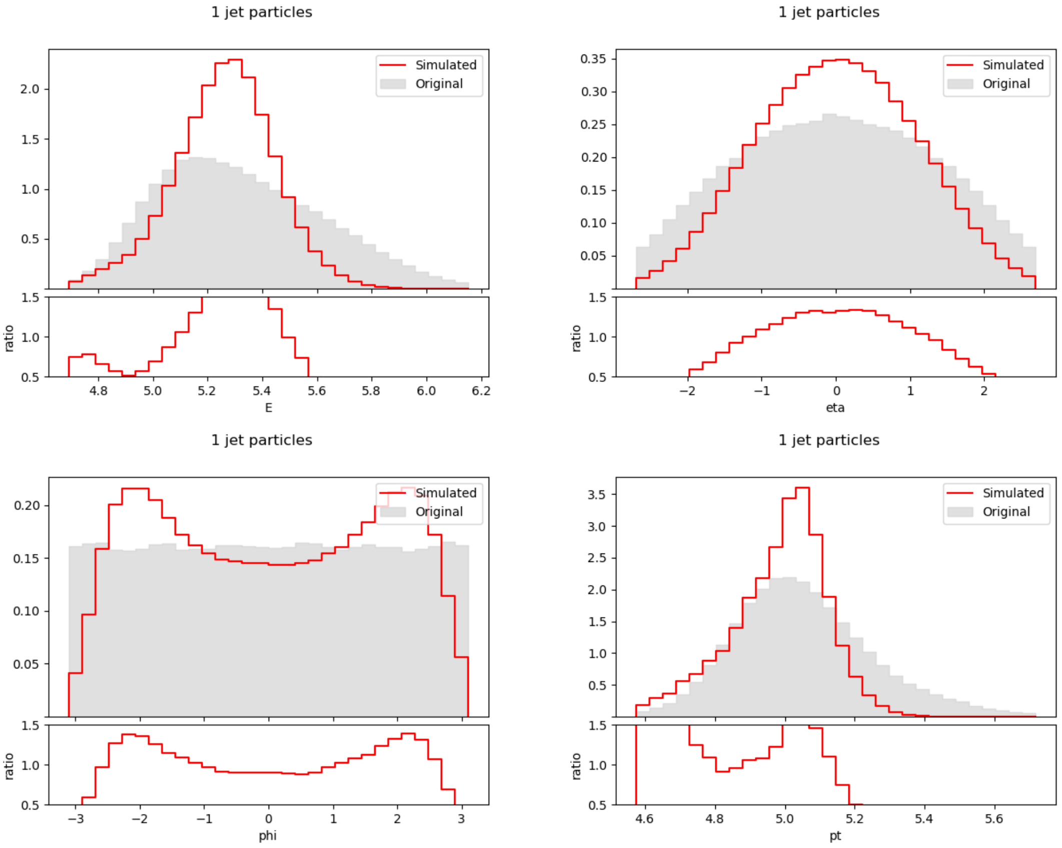


Figure E.22. First jet particle feature distributions in experiment II ($\beta = 0.1$).

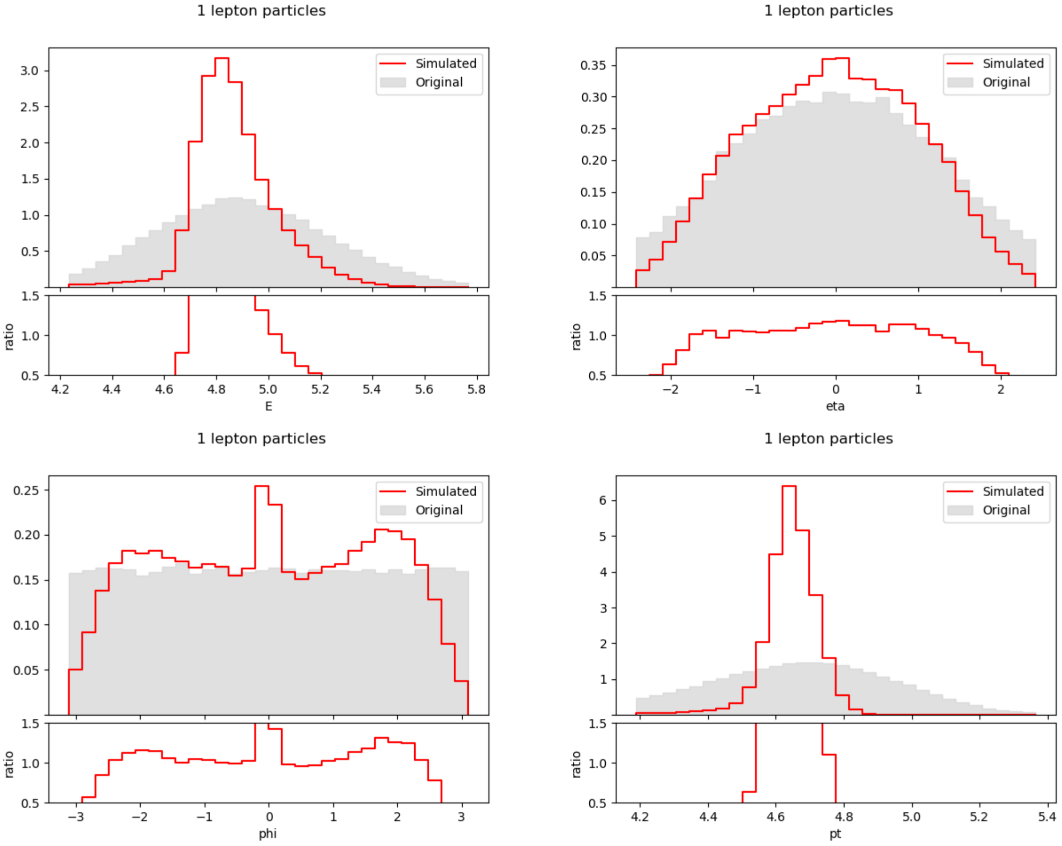


Figure E.23. First lepton particle feature distributions in experiment II ($\beta = 0.1$).

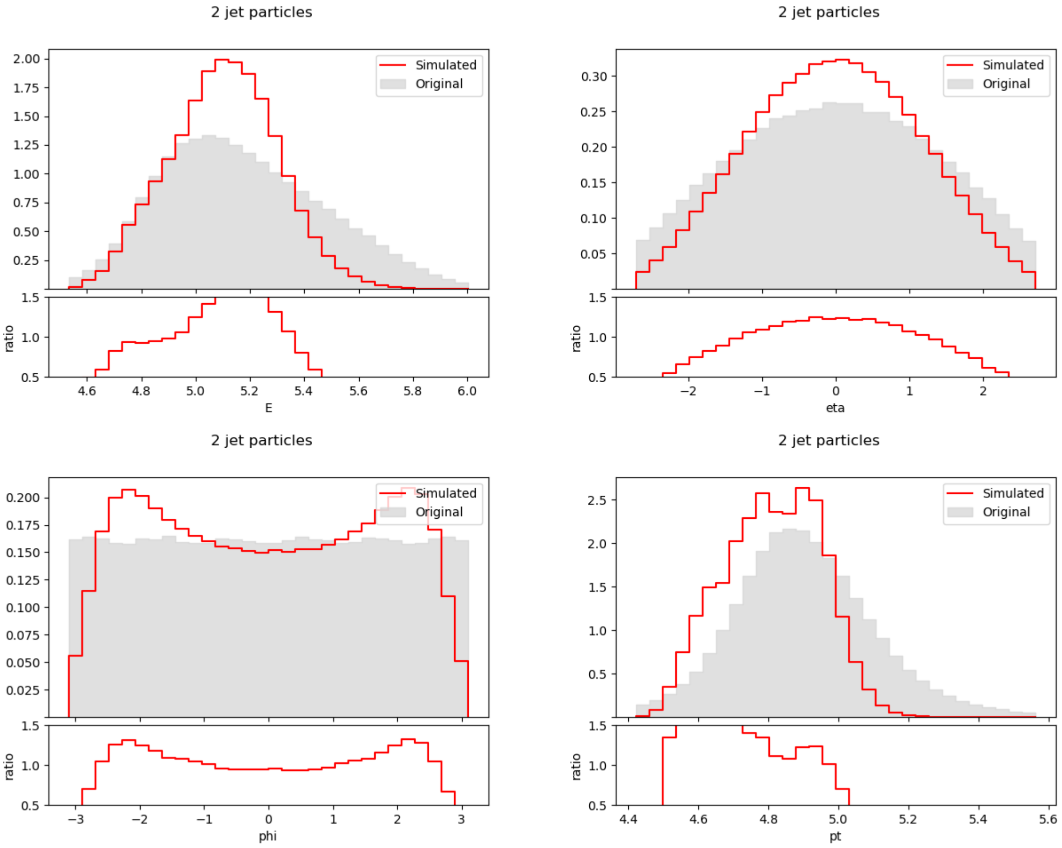


Figure E.24. Second jet particle feature distributions in experiment II ($\beta = 0.1$).

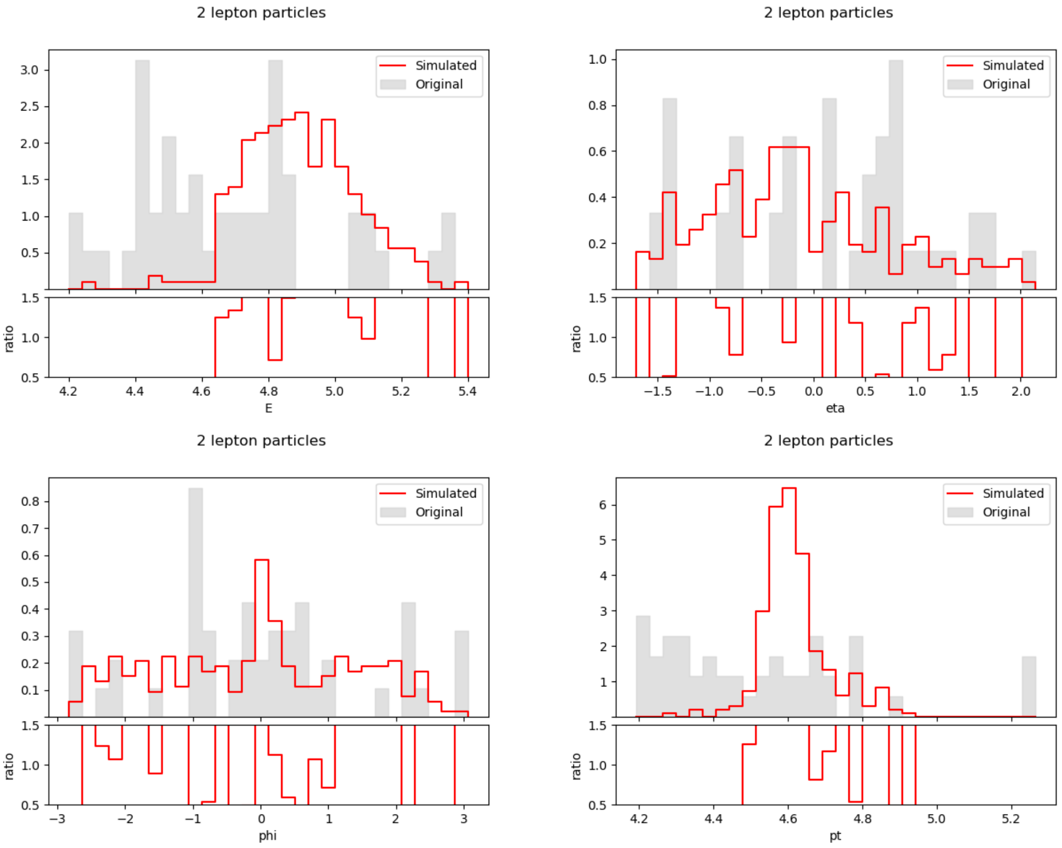


Figure E.25. Second lepton particle feature distributions in experiment II ($\beta = 0.1$).

$\beta = 0.2$

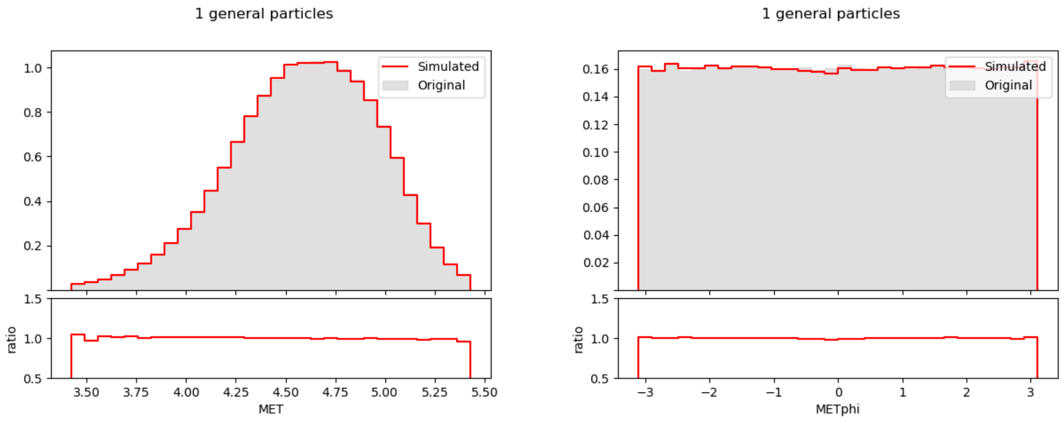


Figure E.26. Event MET and METphi distributions in experiment II ($\beta = 0.2$).

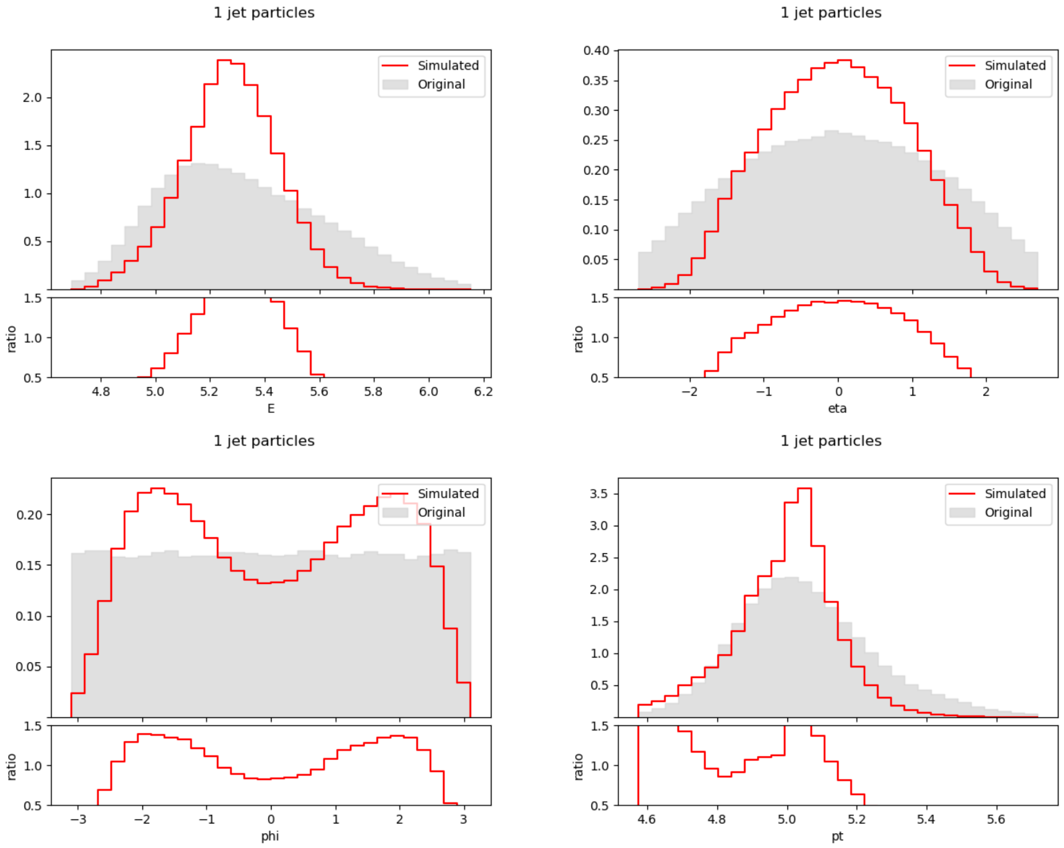


Figure E.27. First jet particle feature distributions in experiment II ($\beta = 0.2$).

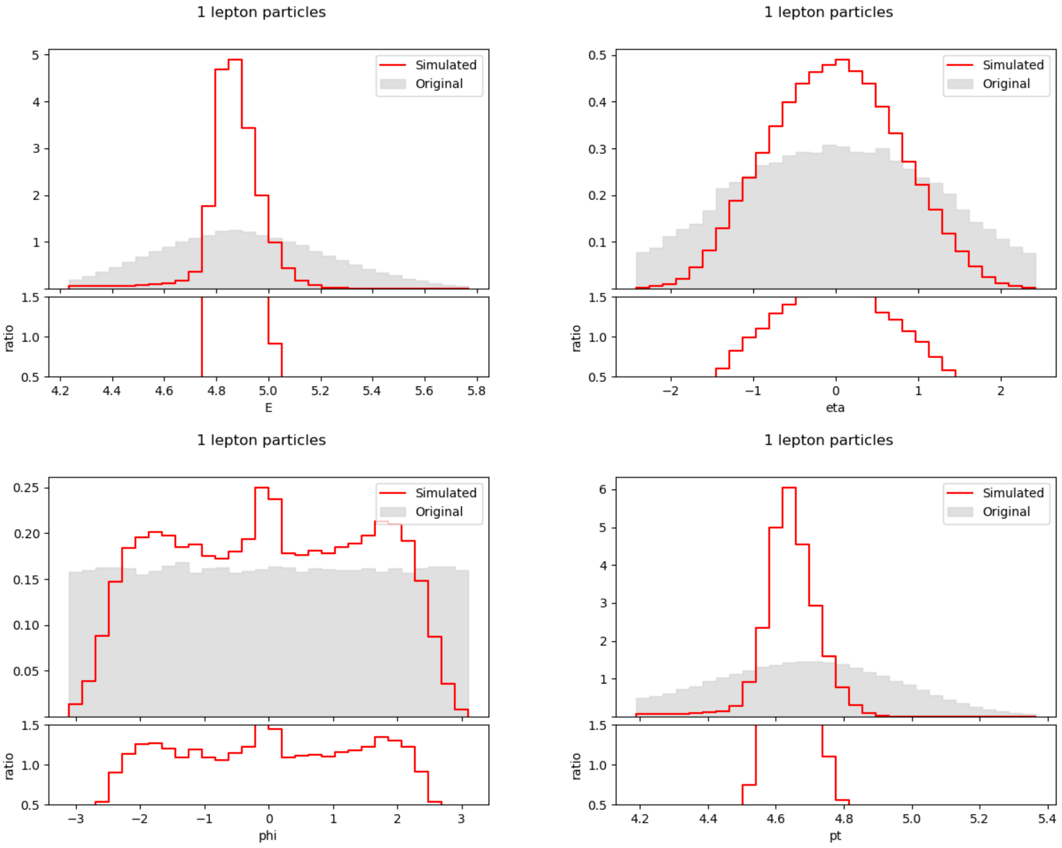


Figure E.28. First lepton particle feature distributions in experiment II ($\beta = 0.2$).

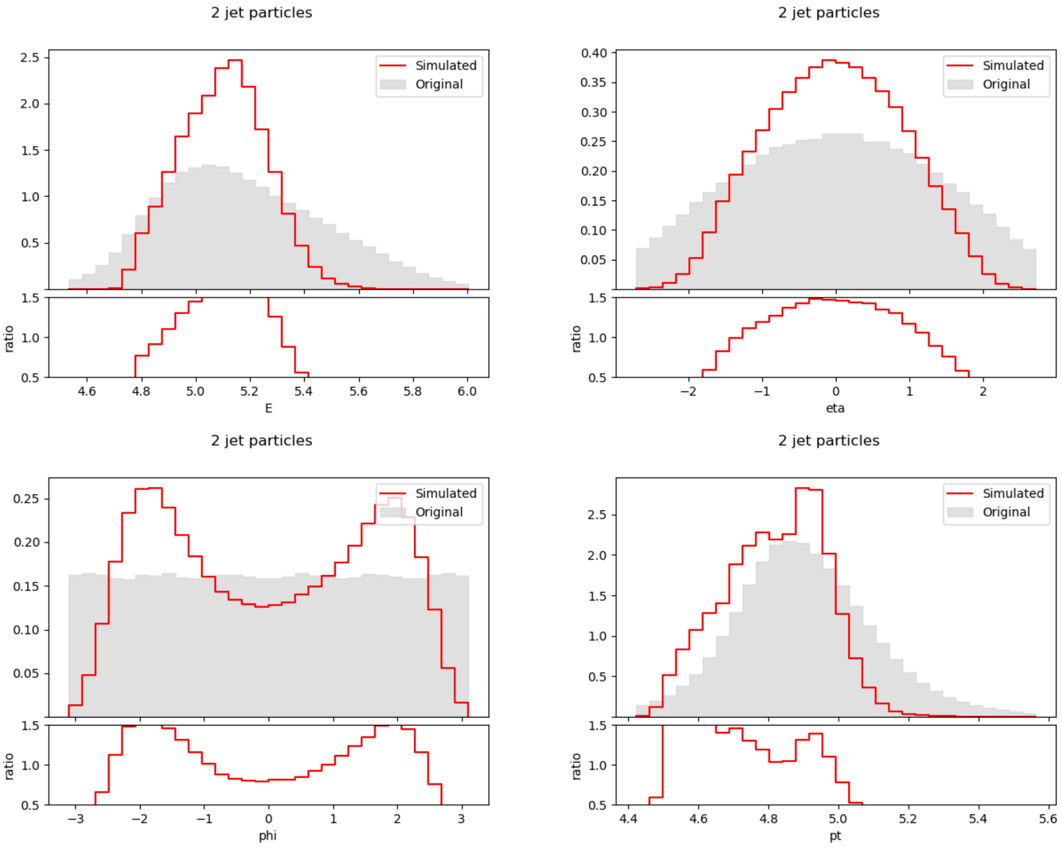


Figure E.29. Second jet particle feature distributions in experiment II ($\beta = 0.2$).

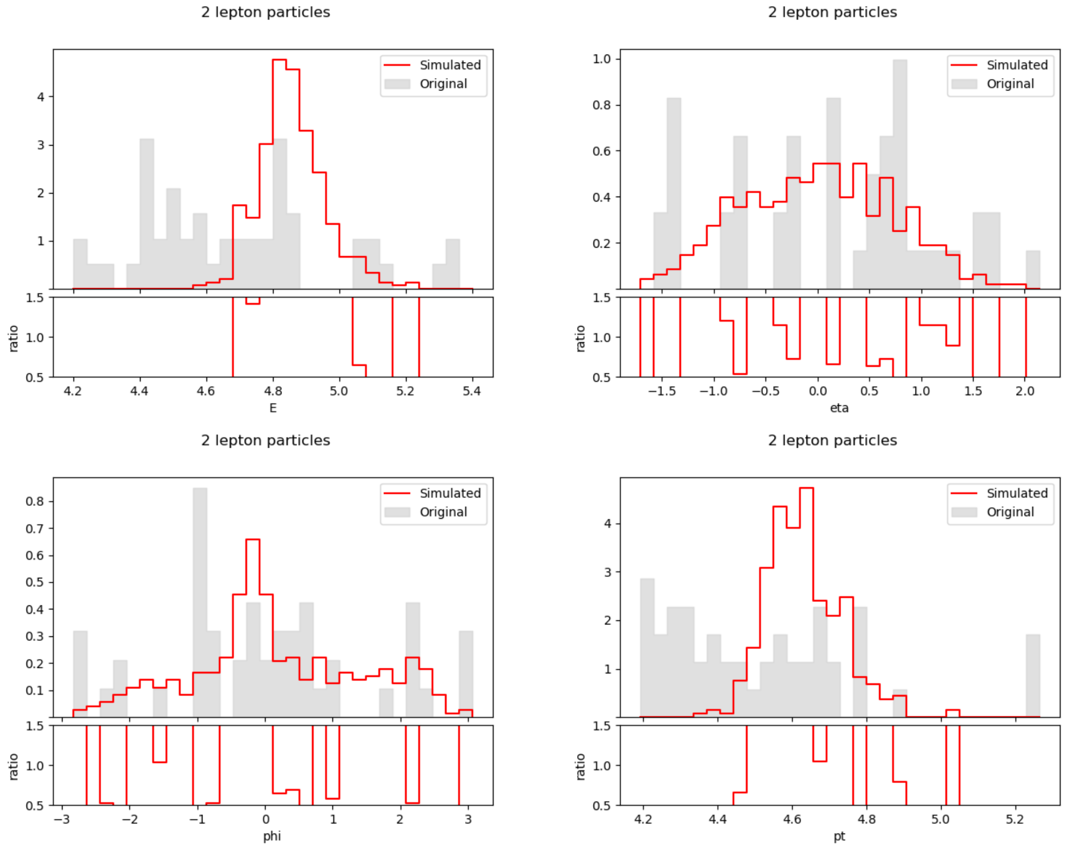


Figure E.30. Second lepton particle feature distributions in experiment II ($\beta = 0.2$).

$\beta = 0.5$

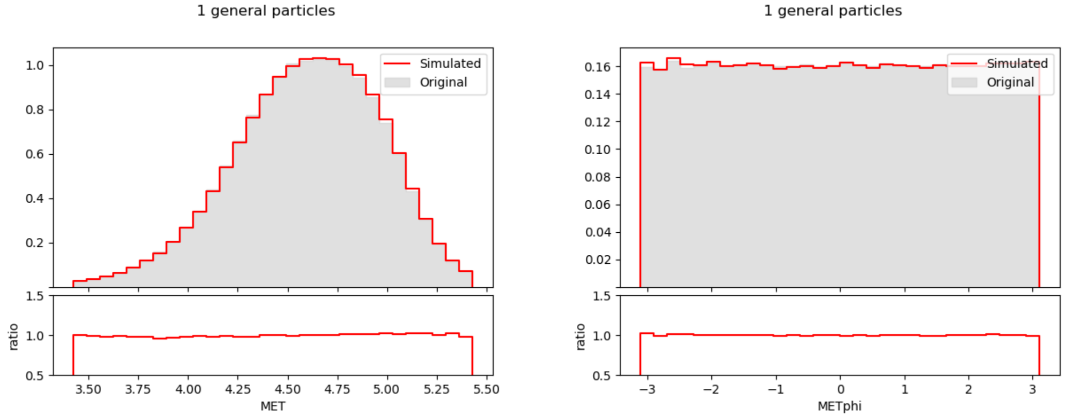


Figure E.31. Event `MET` and `METphi` distributions in experiment II ($\beta = 0.5$).

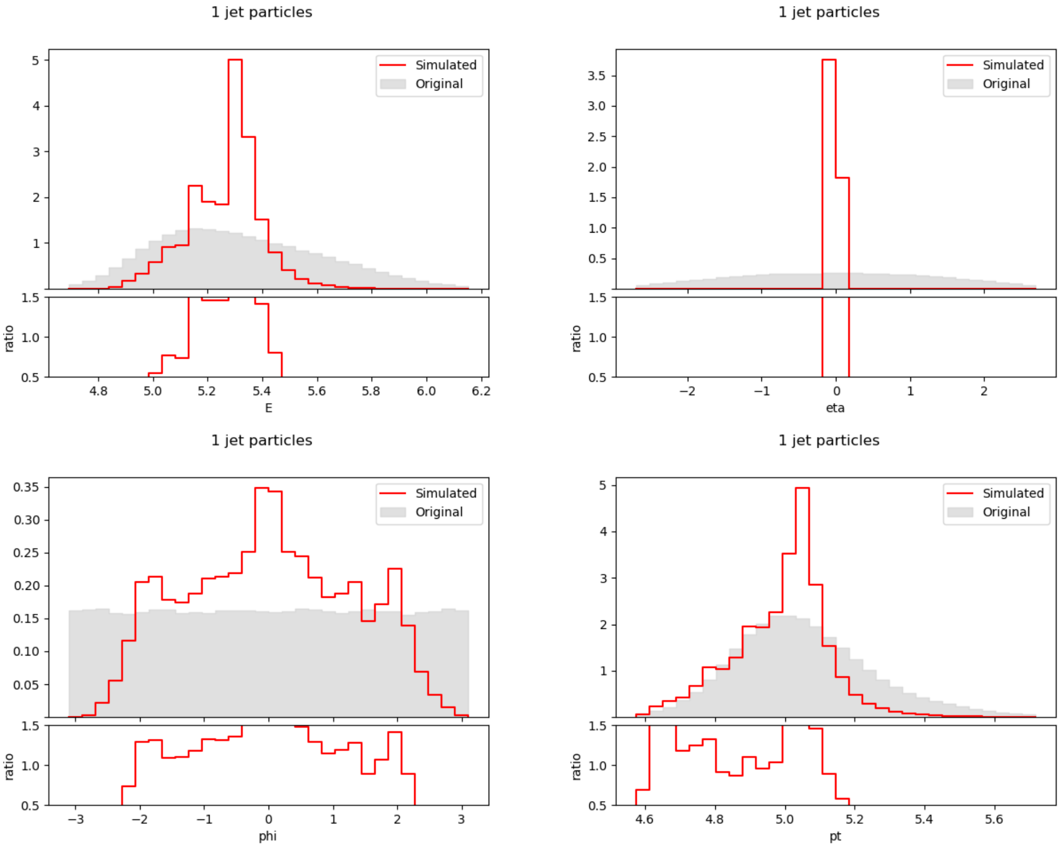


Figure E.32. First jet particle feature distributions in experiment II ($\beta = 0.5$).

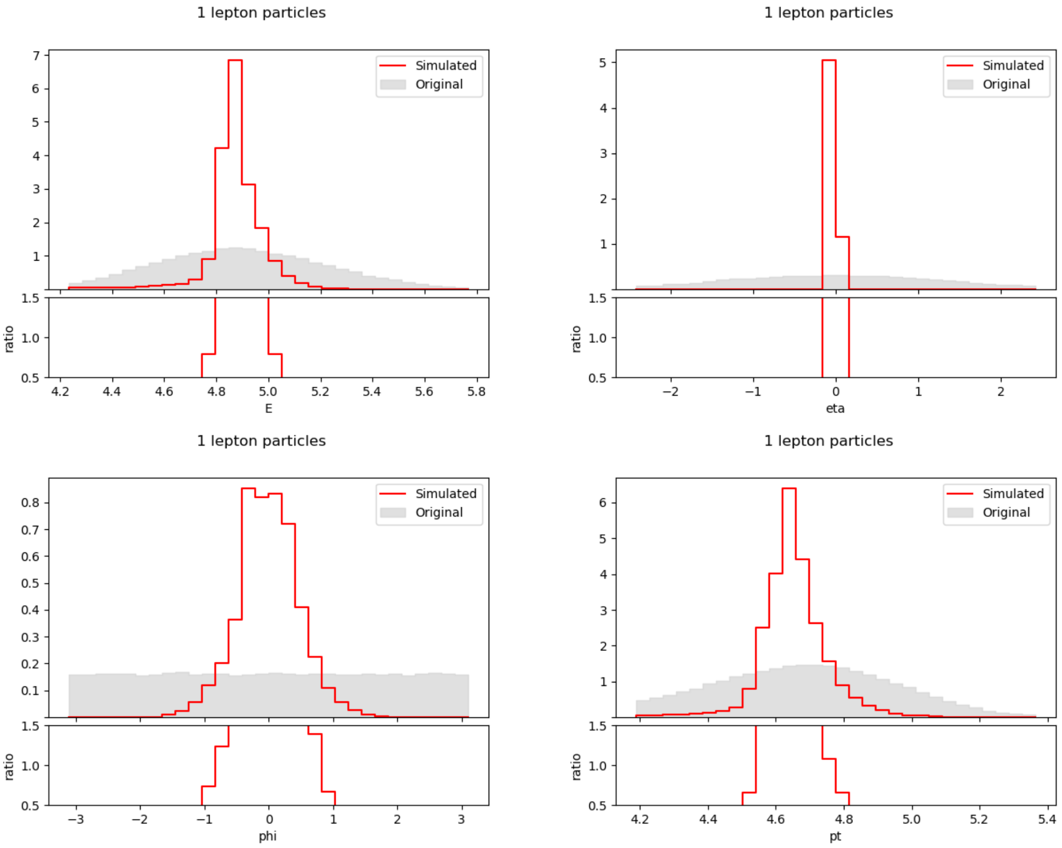


Figure E.33. First lepton particle feature distributions in experiment II ($\beta = 0.5$).

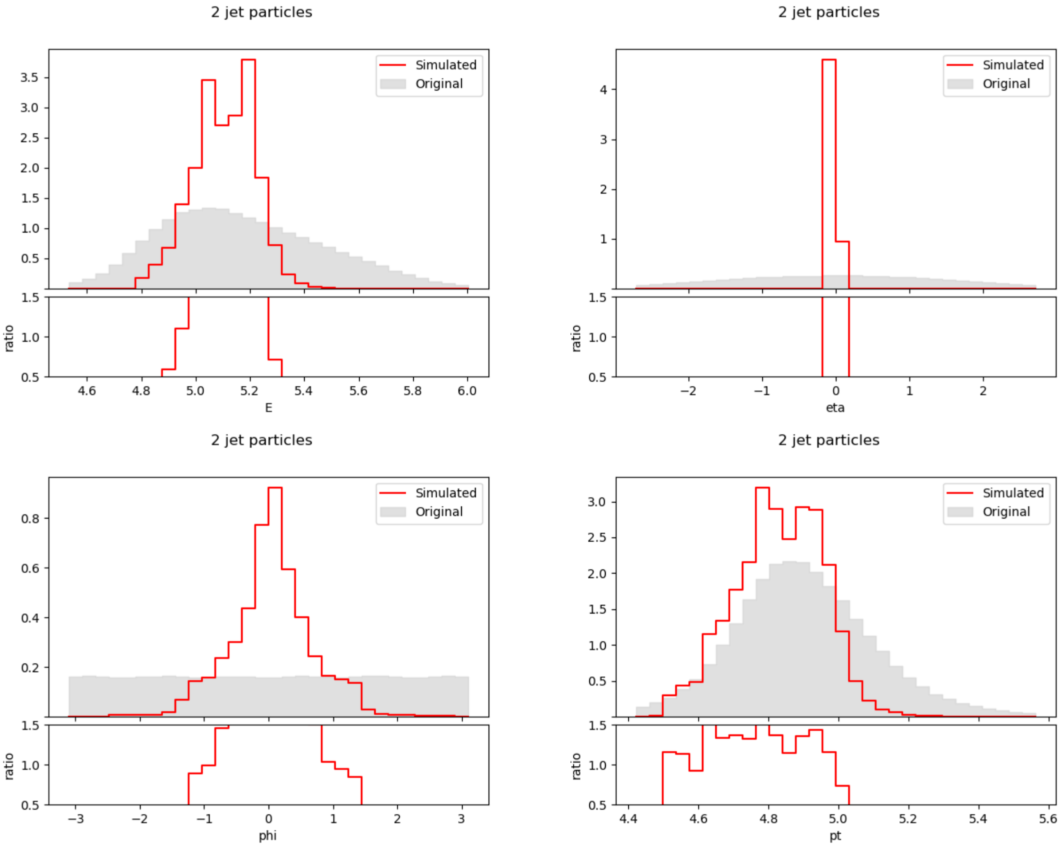


Figure E.34. Second jet particle feature distributions in experiment II ($\beta = 0.5$).

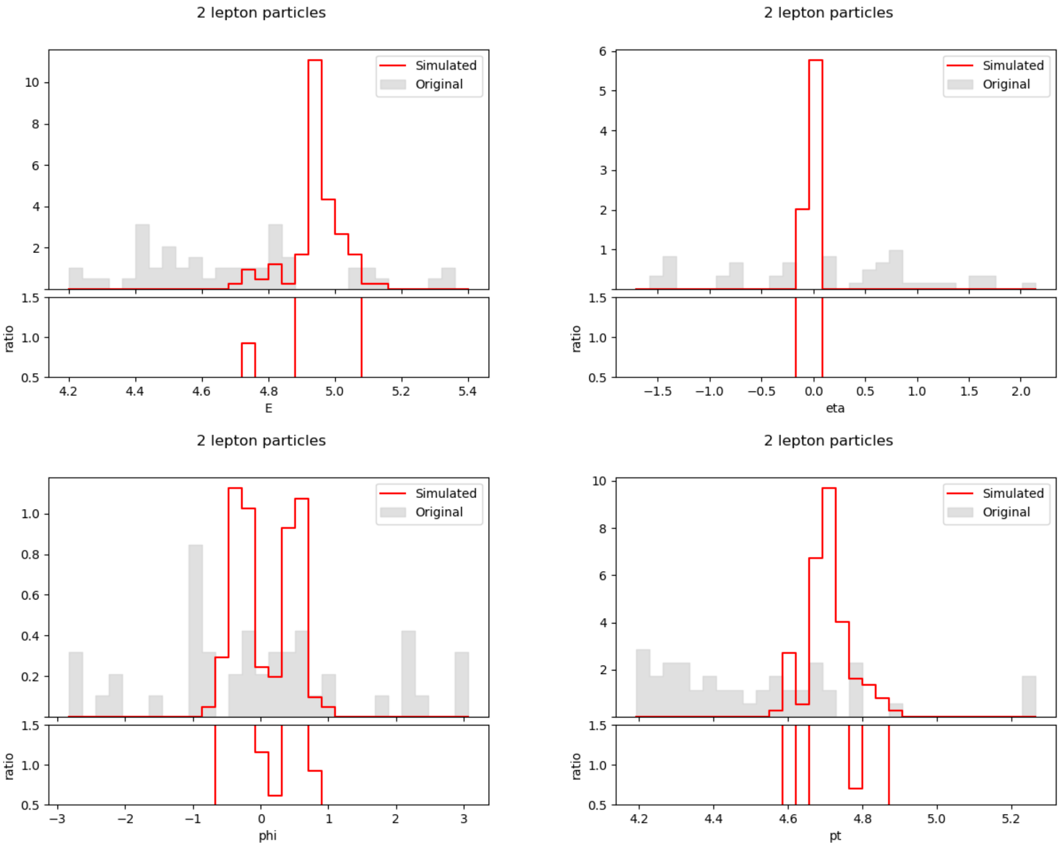


Figure E.35. Second lepton particle feature distributions in experiment II ($\beta = 0.5$).

$\beta = 0.7$

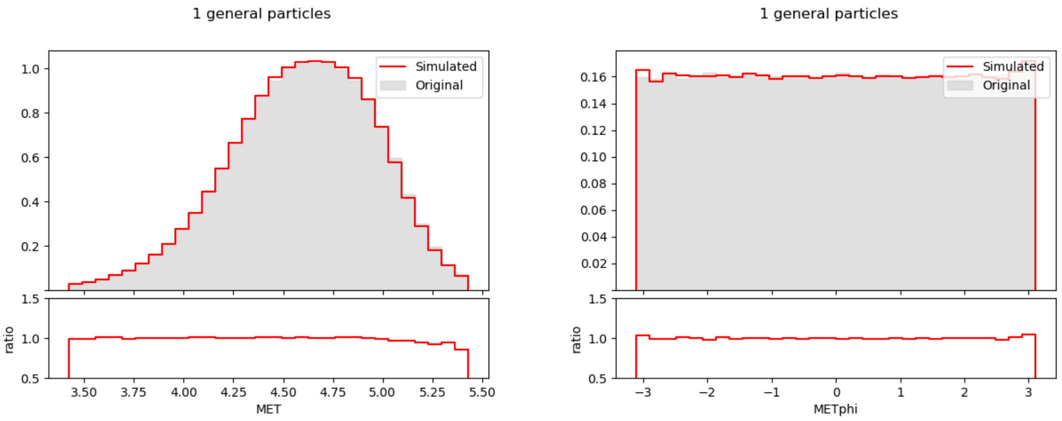


Figure E.36. Event MET and METphi distributions in experiment II ($\beta = 0.7$).

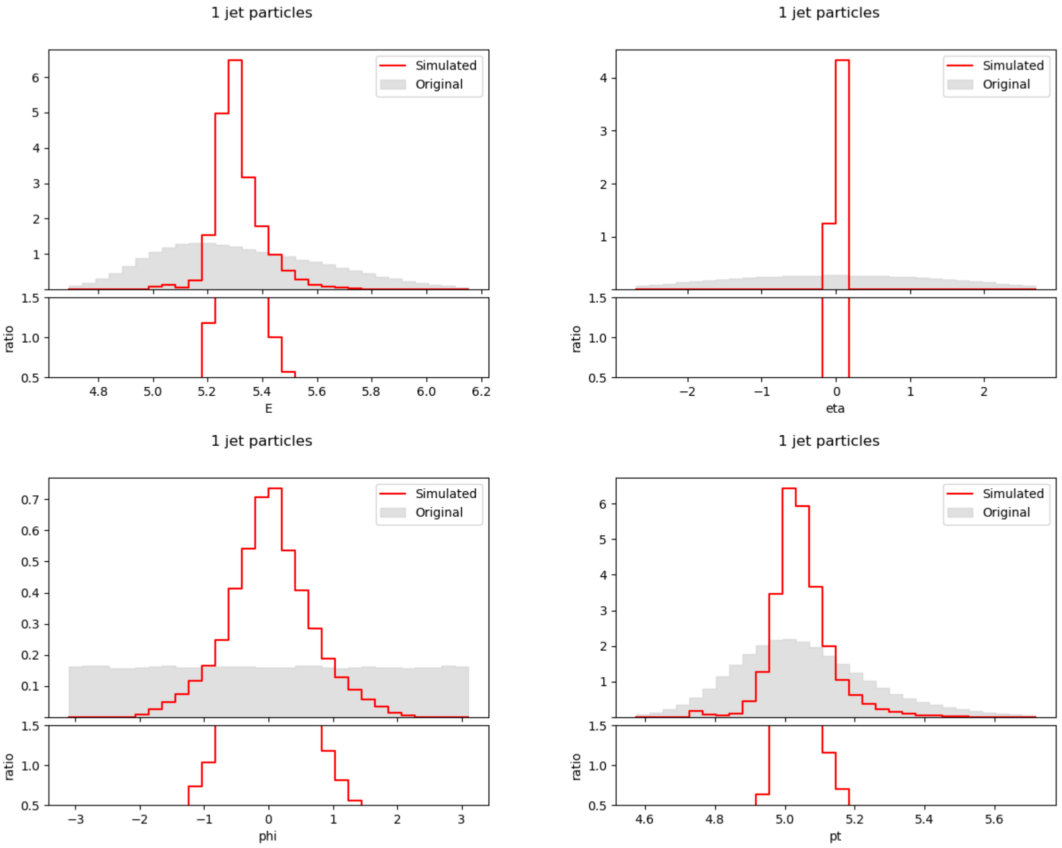


Figure E.37. First jet particle feature distributions in experiment II ($\beta = 0.7$).

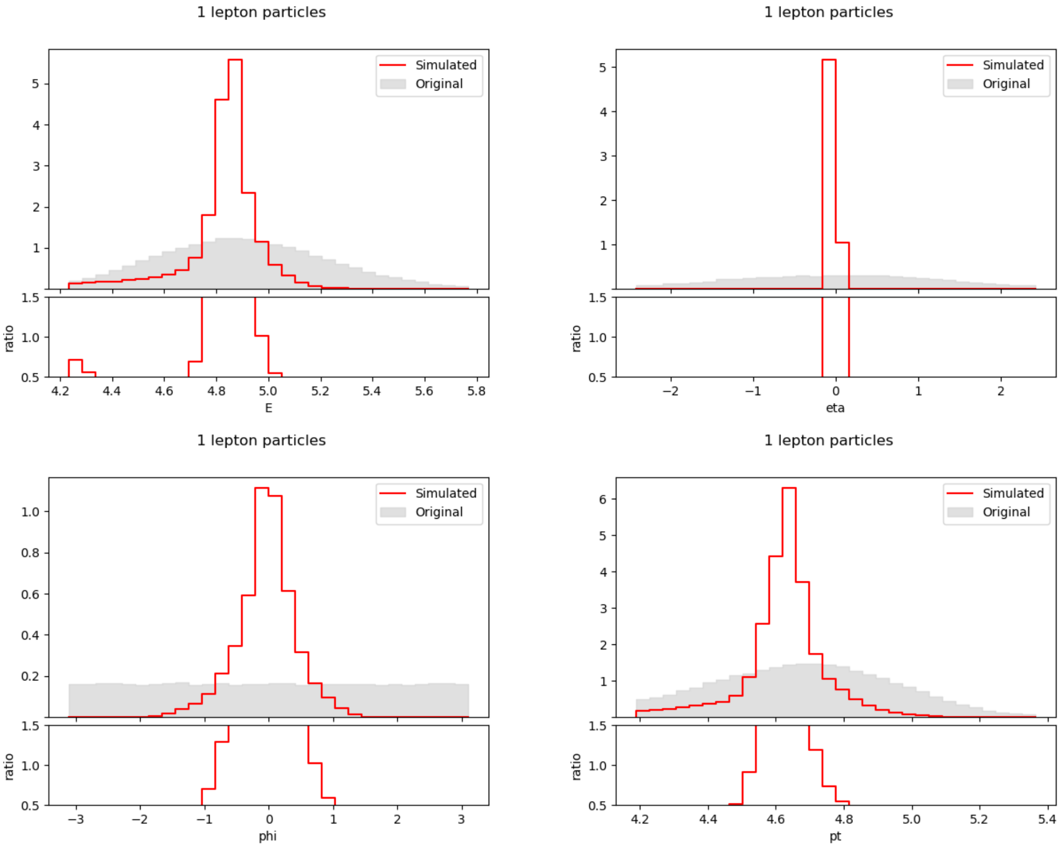


Figure E.38. First lepton particle feature distributions in experiment II ($\beta = 0.7$).

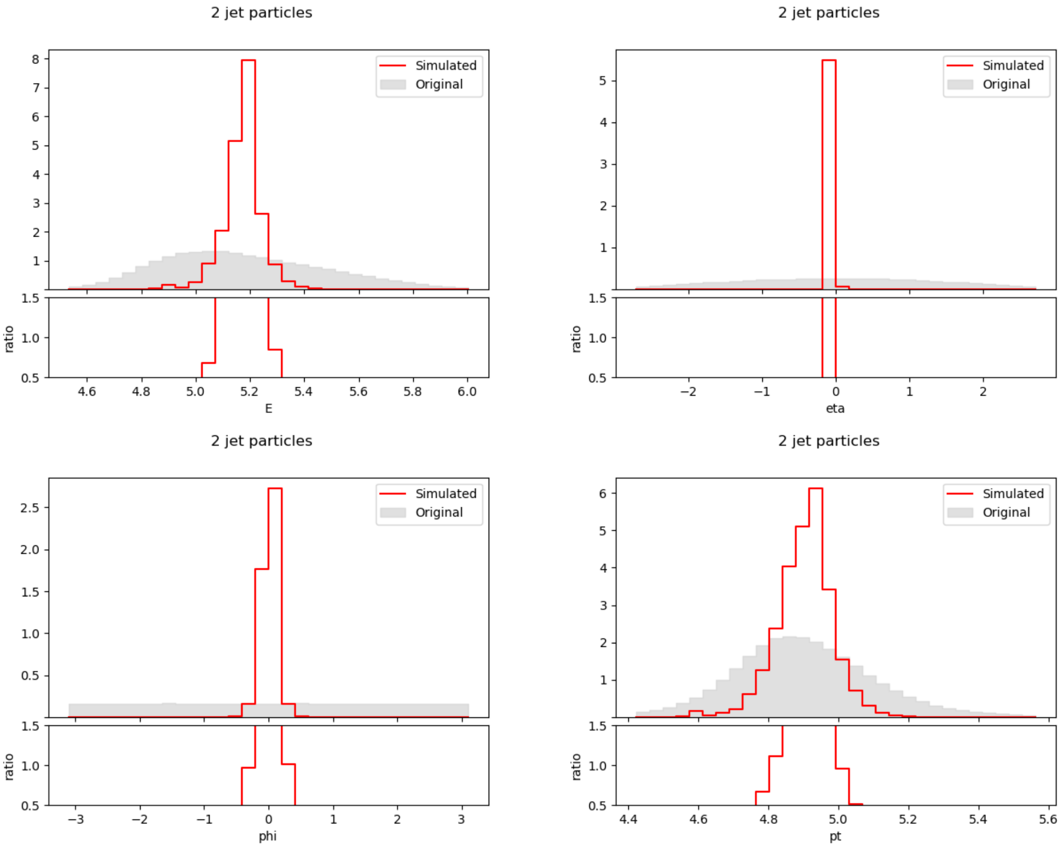


Figure E.39. Second jet particle feature distributions in experiment II ($\beta = 0.7$).

No second leptons were present in the simulated events using $\beta = 0.7$.

$\beta = 1.0$

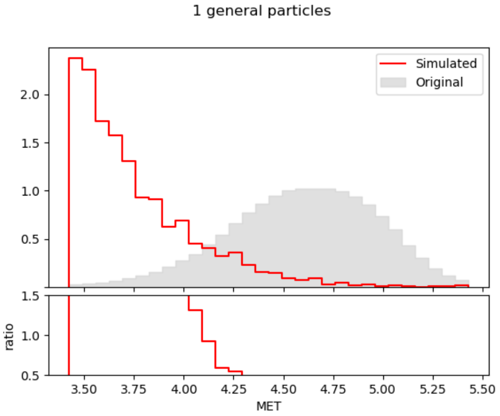


Figure E.40. Event MET and METphi distributions in experiment II ($\beta = 1.0$).

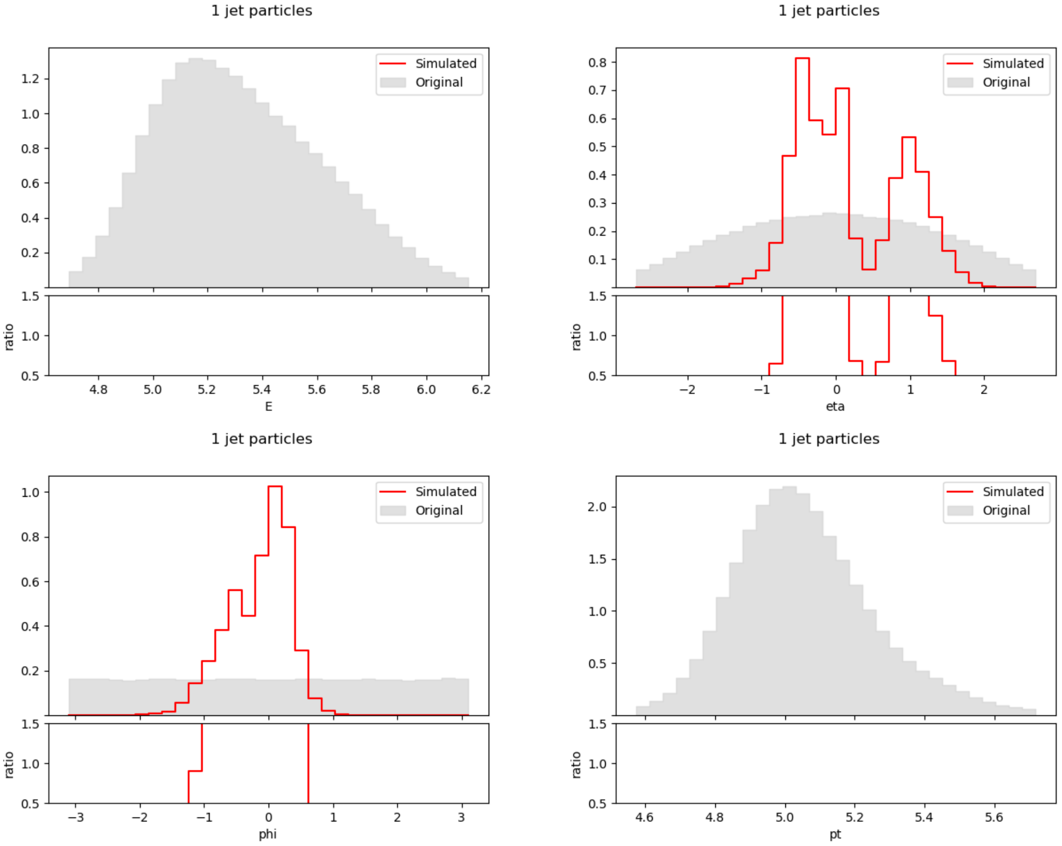


Figure E.41. First jet particle feature distributions in experiment II ($\beta = 1.0$).

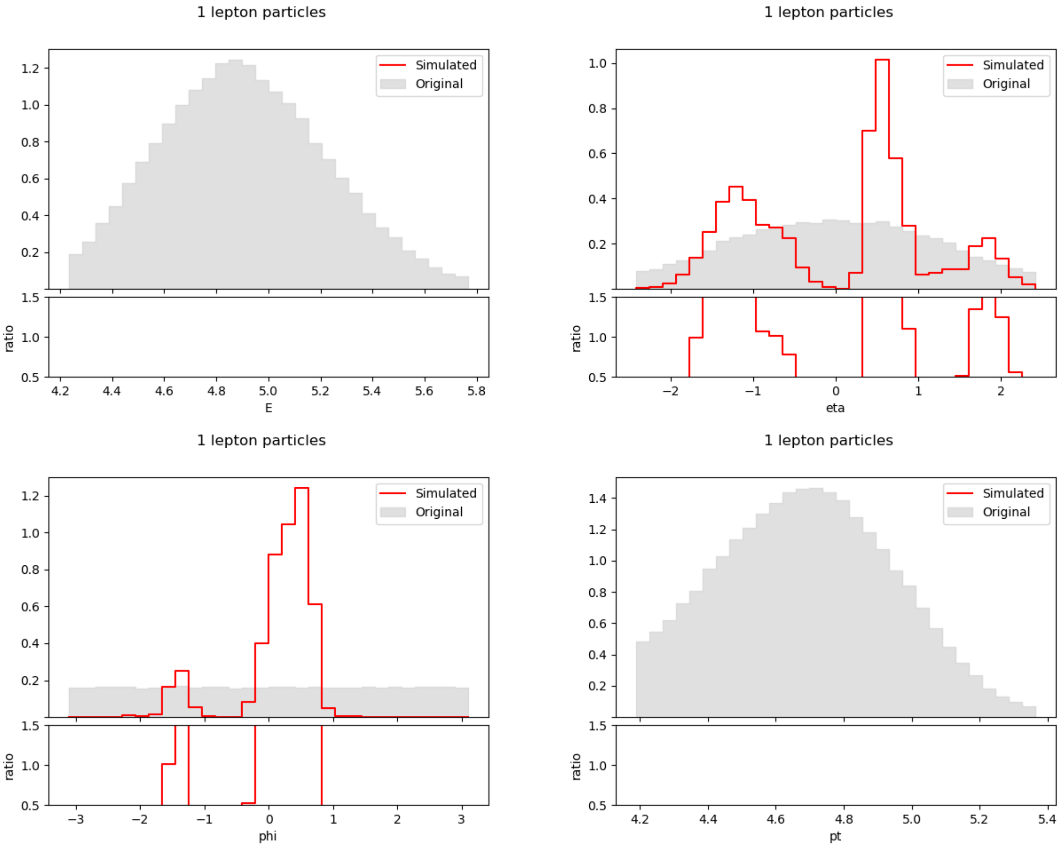


Figure E.42. First lepton particle feature distributions in experiment II ($\beta = 1.0$).

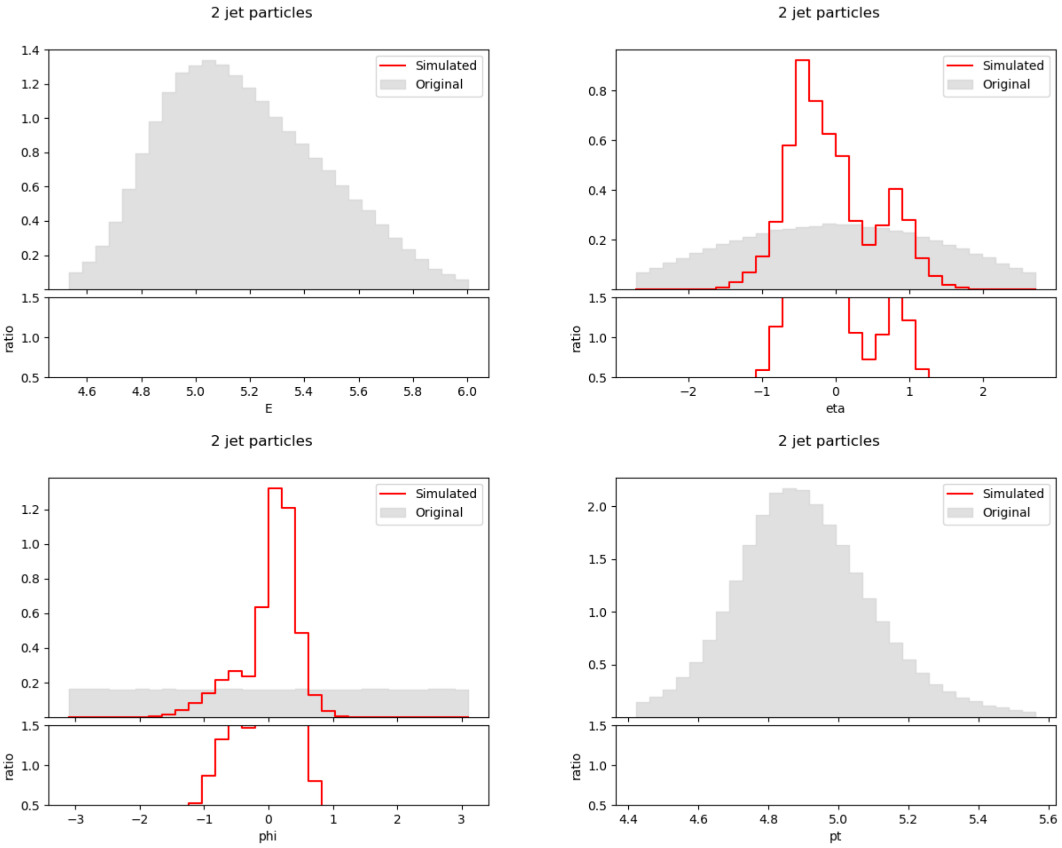


Figure E.43. Second jet particle feature distributions in experiment II ($\beta = 1.0$).

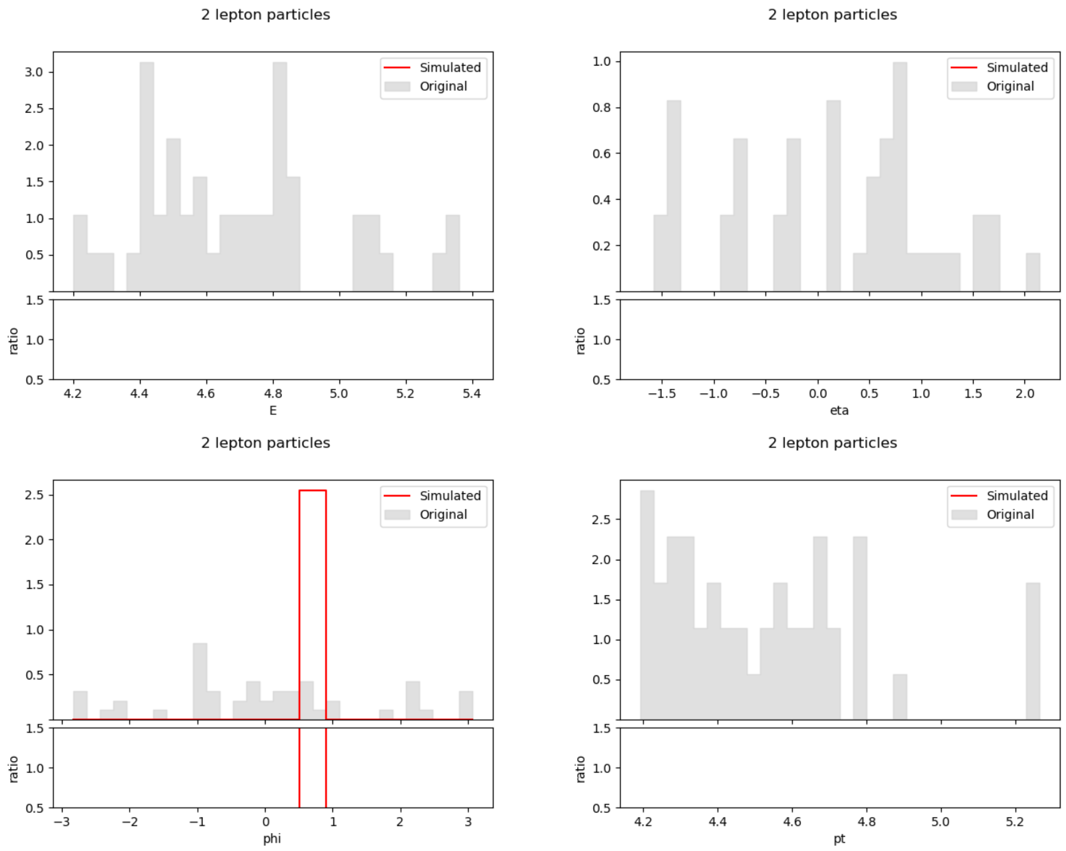


Figure E.44. Second lepton particle feature distributions in experiment II ($\beta = 1.0$).

E.2.2. With BGMMs

$\beta = 0.0, \gamma = 100$

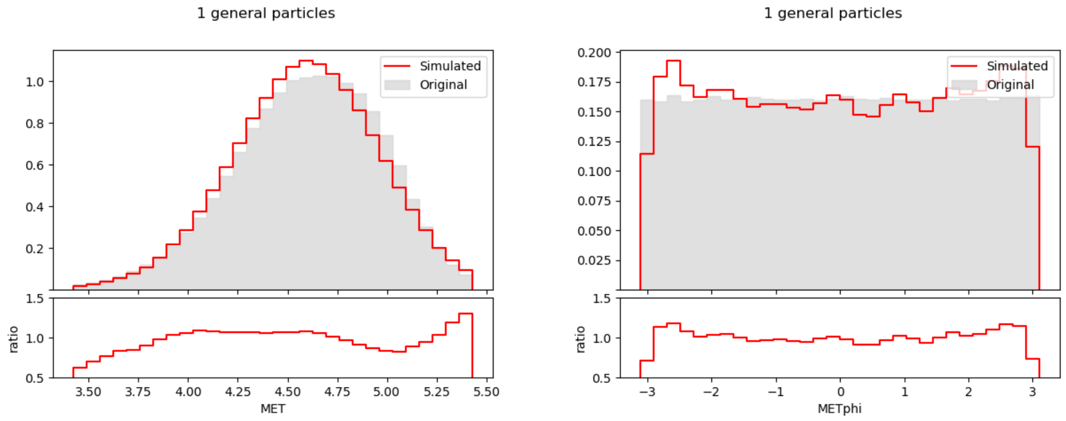


Figure E.45. Event `MET` and `METphi` distributions using a BGMM in experiment II ($\beta = 0.0$).

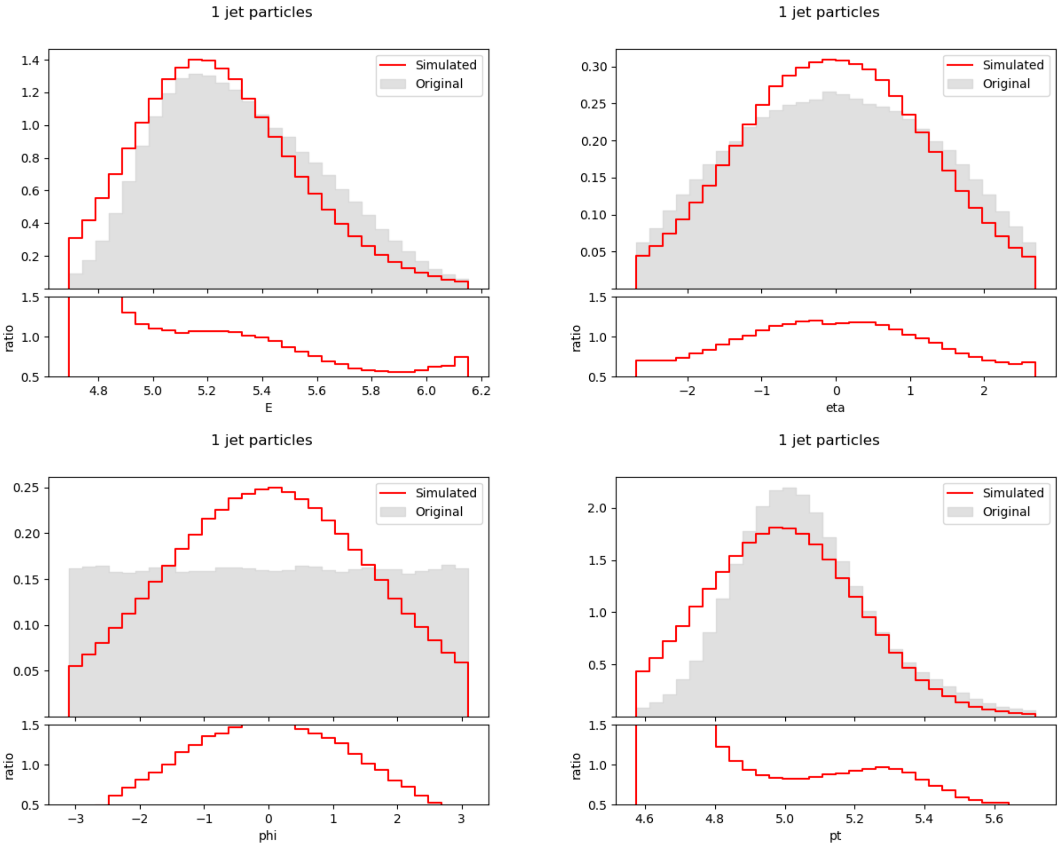


Figure E.46. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.0$).

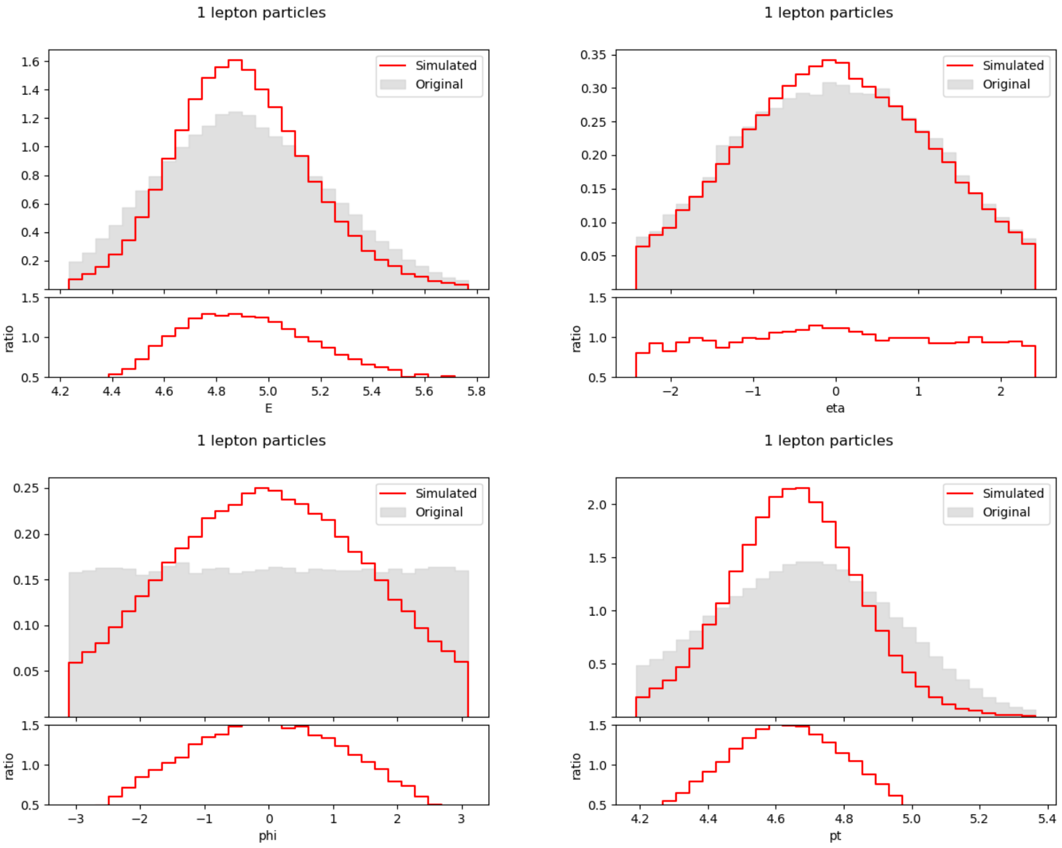


Figure E.47. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.0$).

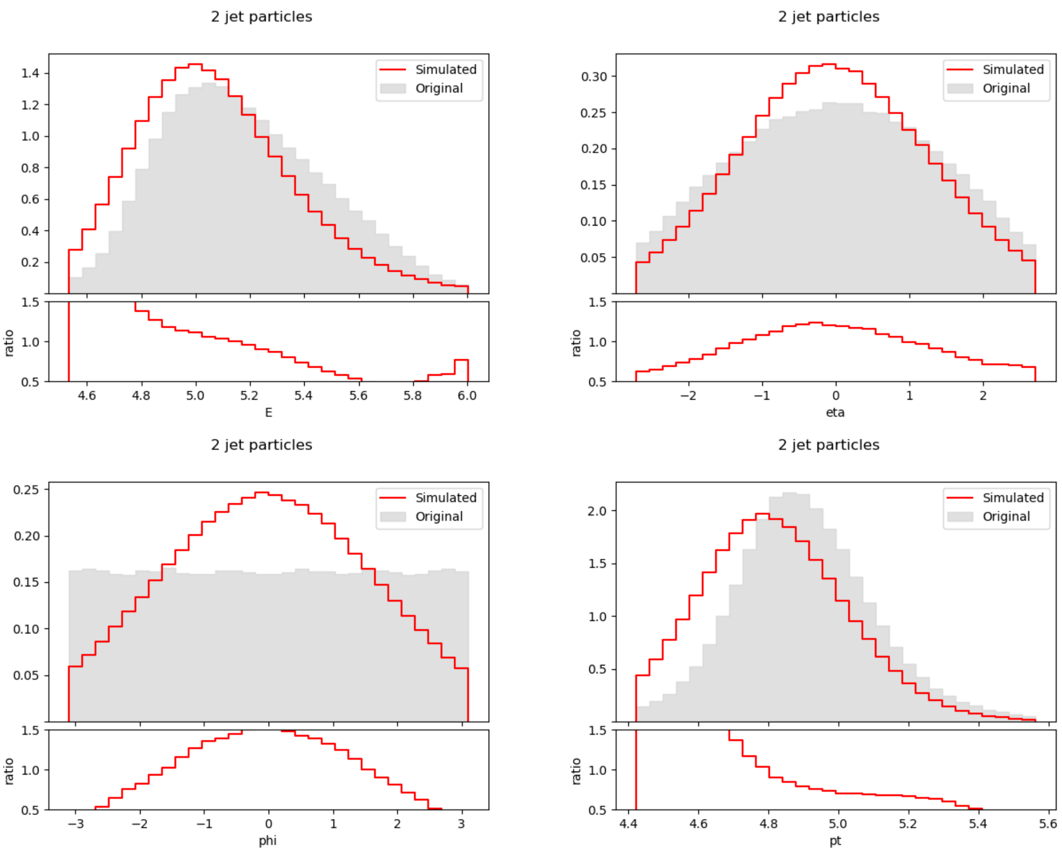


Figure E.48. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.0$).

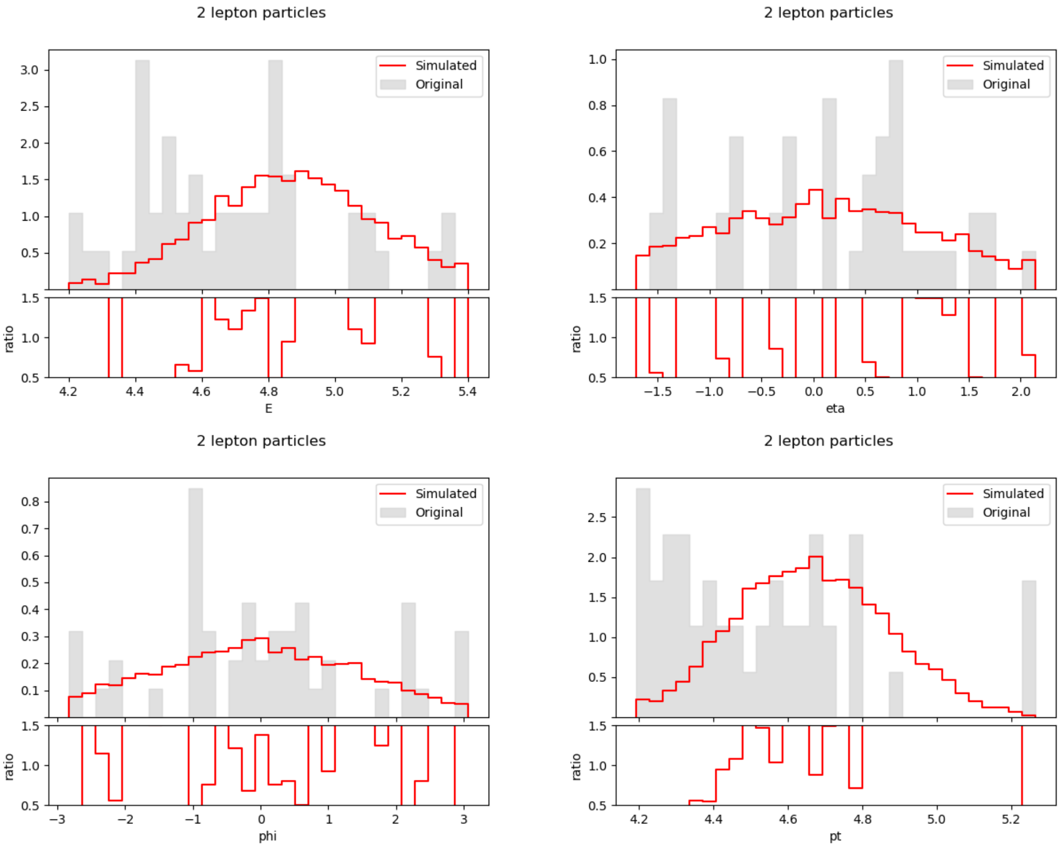


Figure E.49. Second lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.0$).

$\beta = 0.001, \gamma = 100$

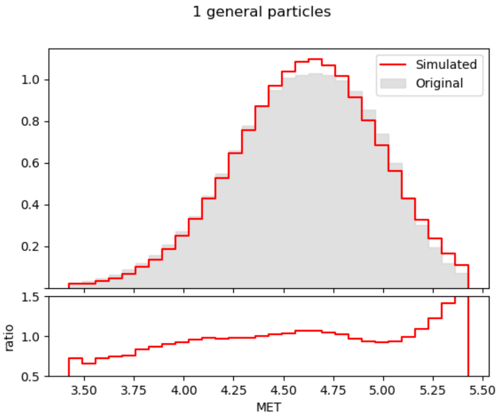
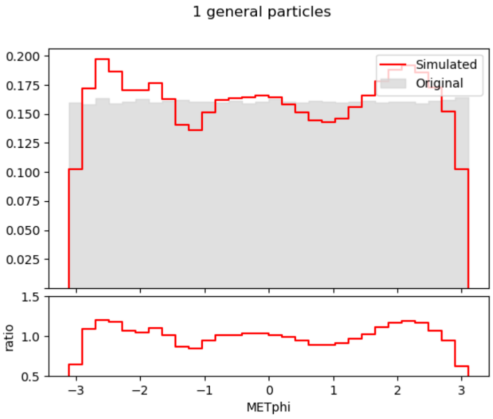


Figure E.50. Event `MET` and `METphi` distributions using a BGMM in experiment II ($\beta = 0.001$).



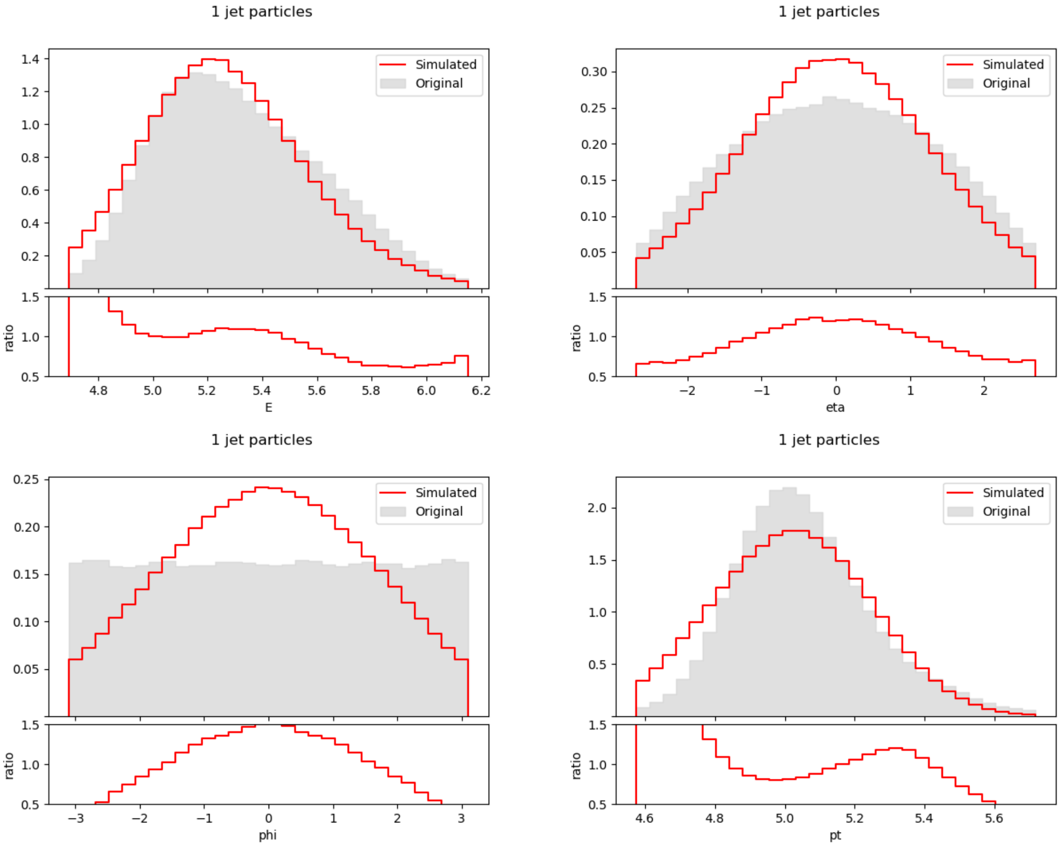


Figure E.51. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.001$).

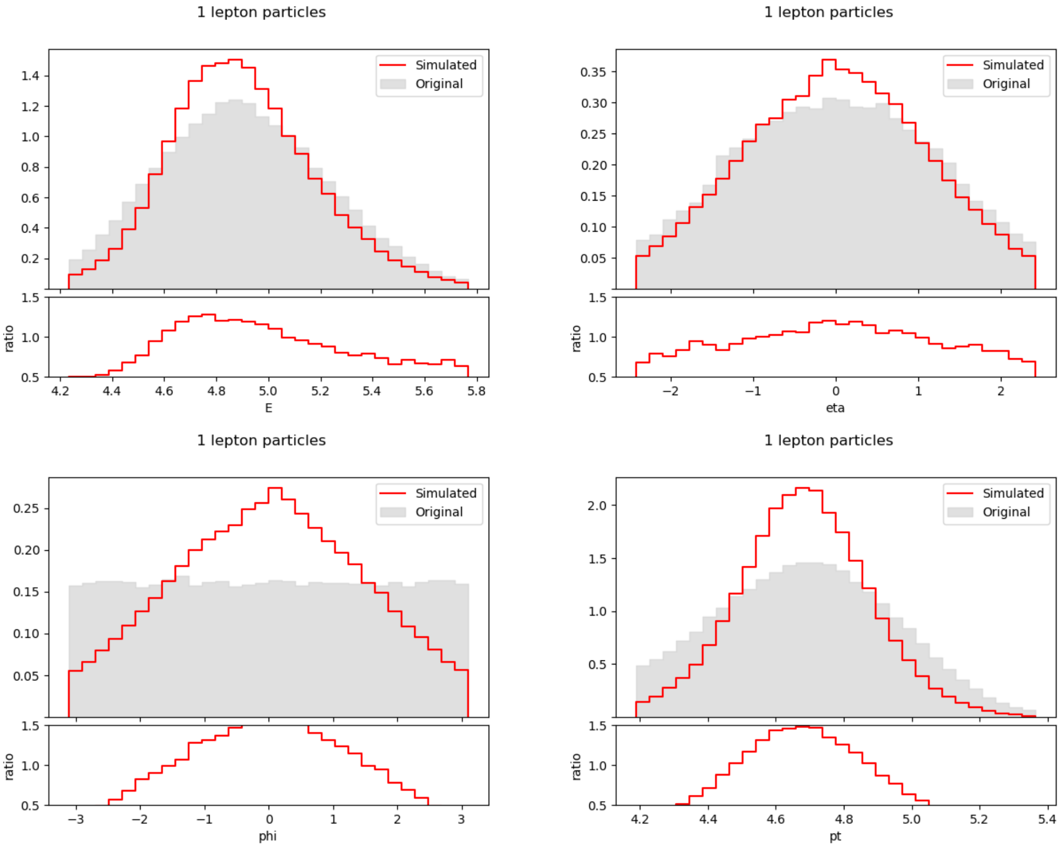


Figure E.52. First lepton particle feature using a BGMM in experiment II ($\beta = 0.001$).

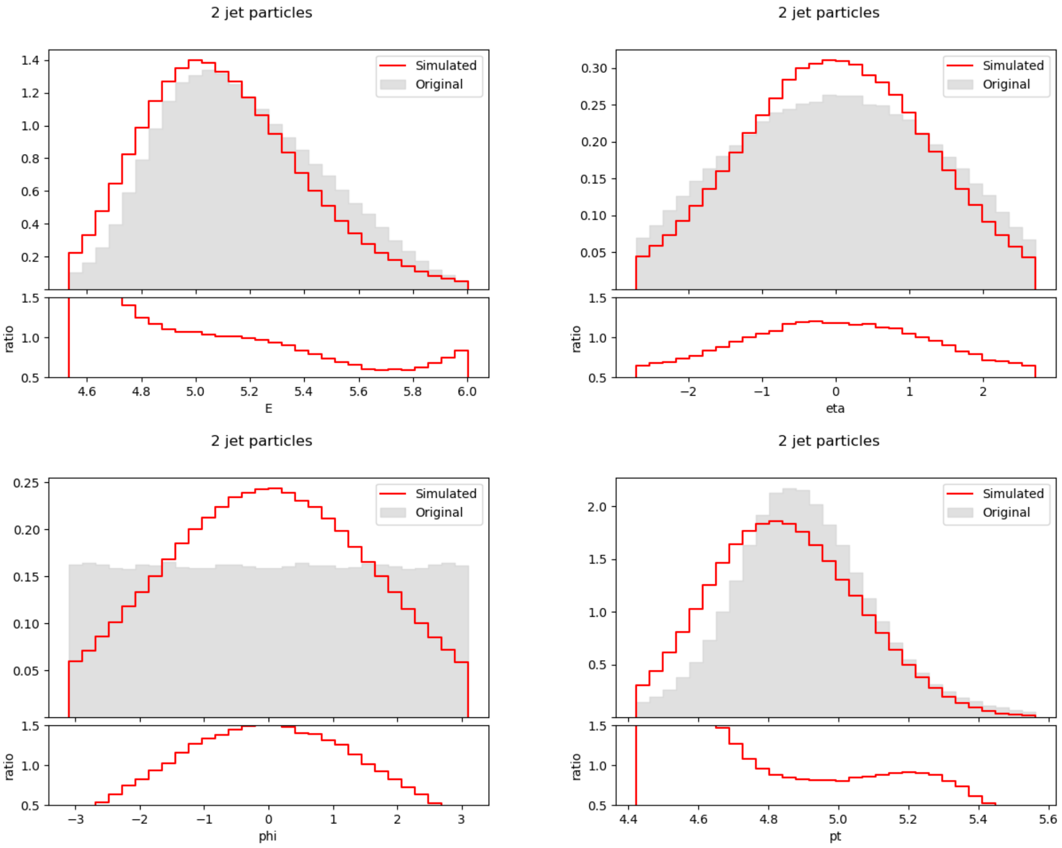


Figure E.53. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.001$).

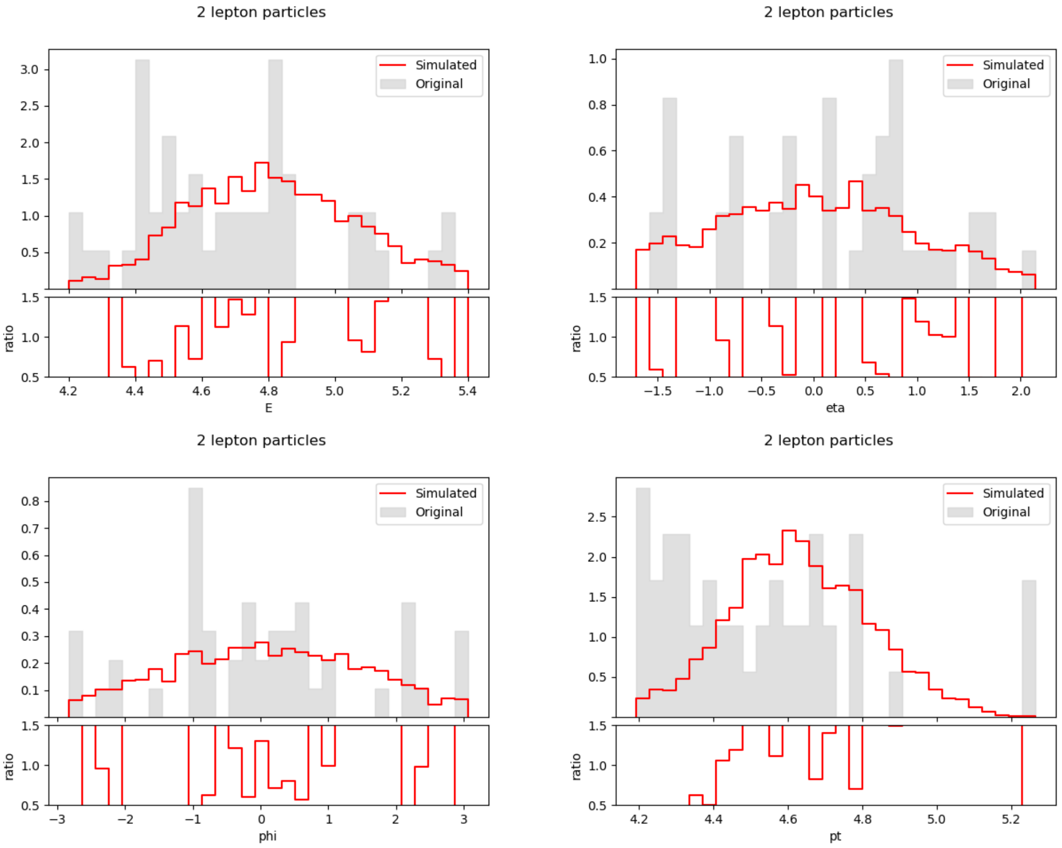


Figure E.54. Second lepton particle feature using a BGMM in experiment II ($\beta = 0.001$).

$\beta = 0.01$, $\gamma = 100$

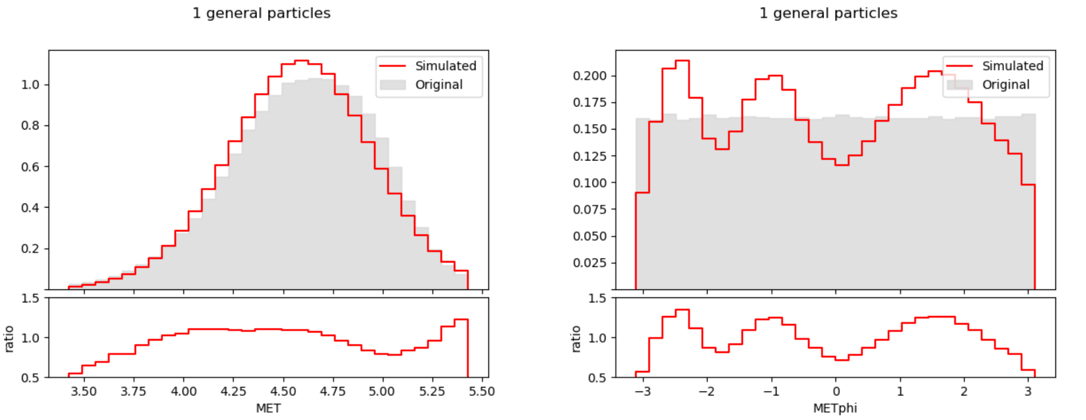


Figure E.55. Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.01$).

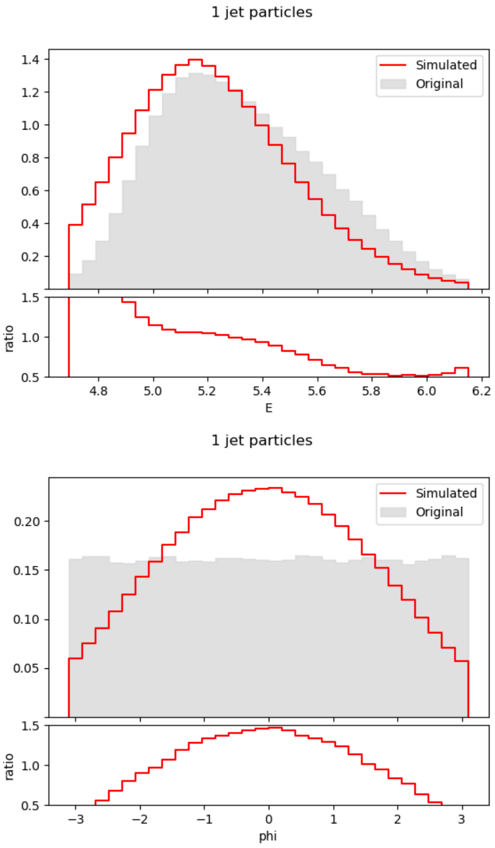


Figure E.56. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.01$).

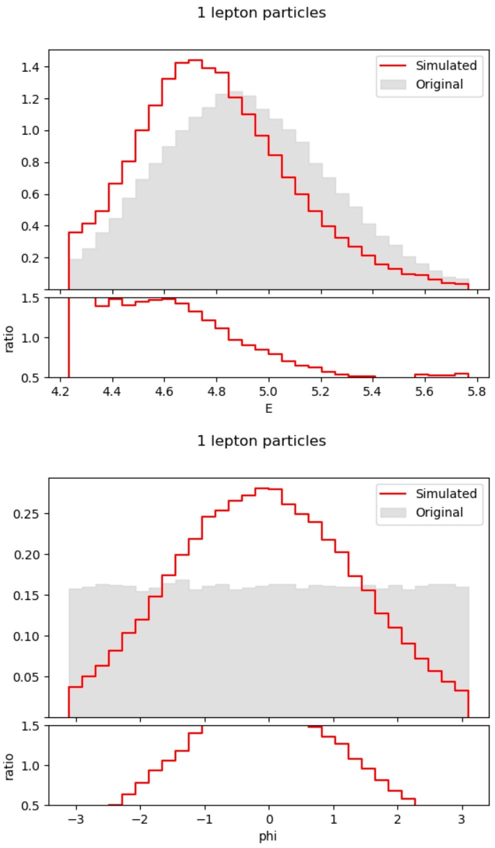


Figure E.57. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.01$).

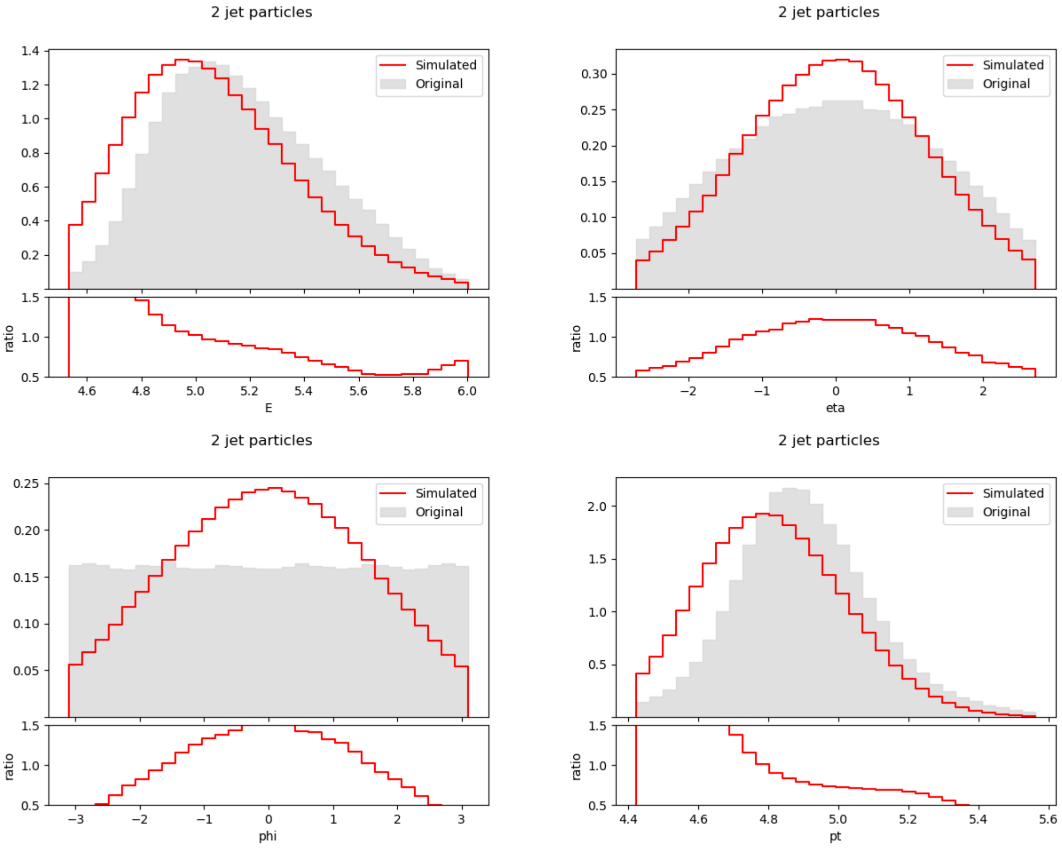


Figure E.58. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.01$).

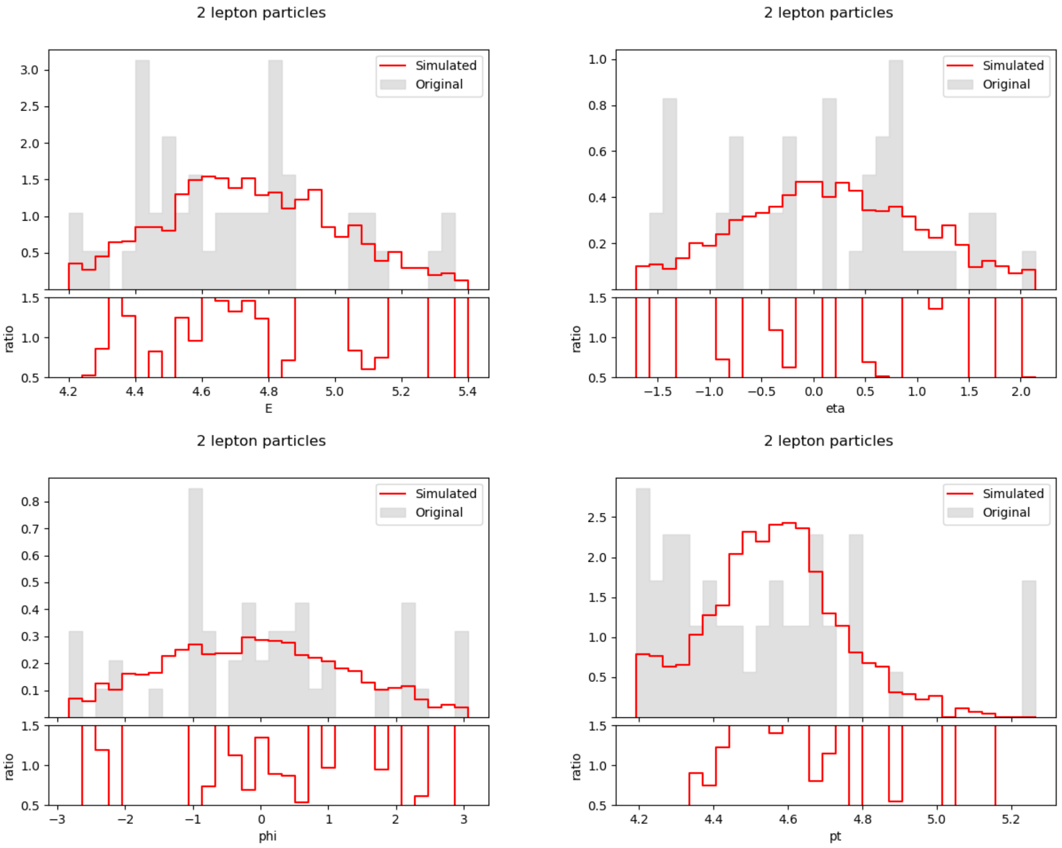


Figure E.59. Second lepton particle feature using a BGMM in experiment II ($\beta = 0.01$).

$\beta = 0.1, \gamma = 100$

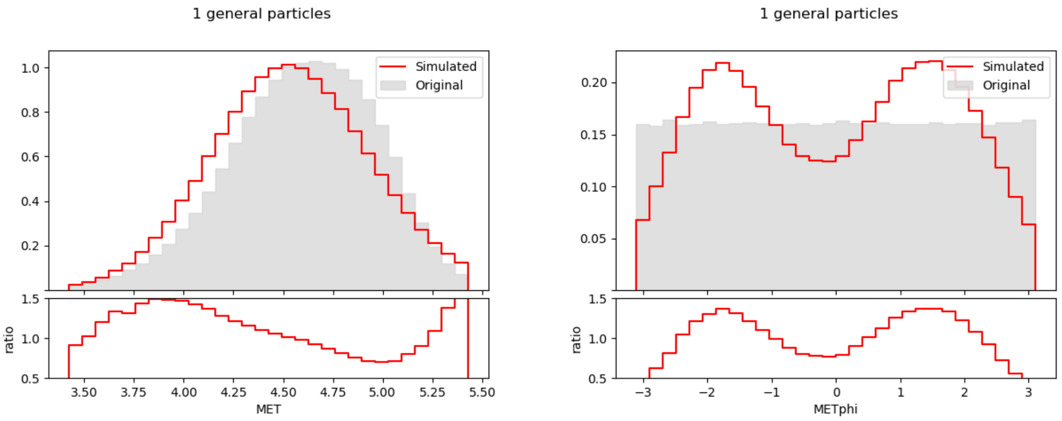


Figure E.60. Event `MET` and `METphi` distributions using a BGMM in experiment II ($\beta = 0.1$).

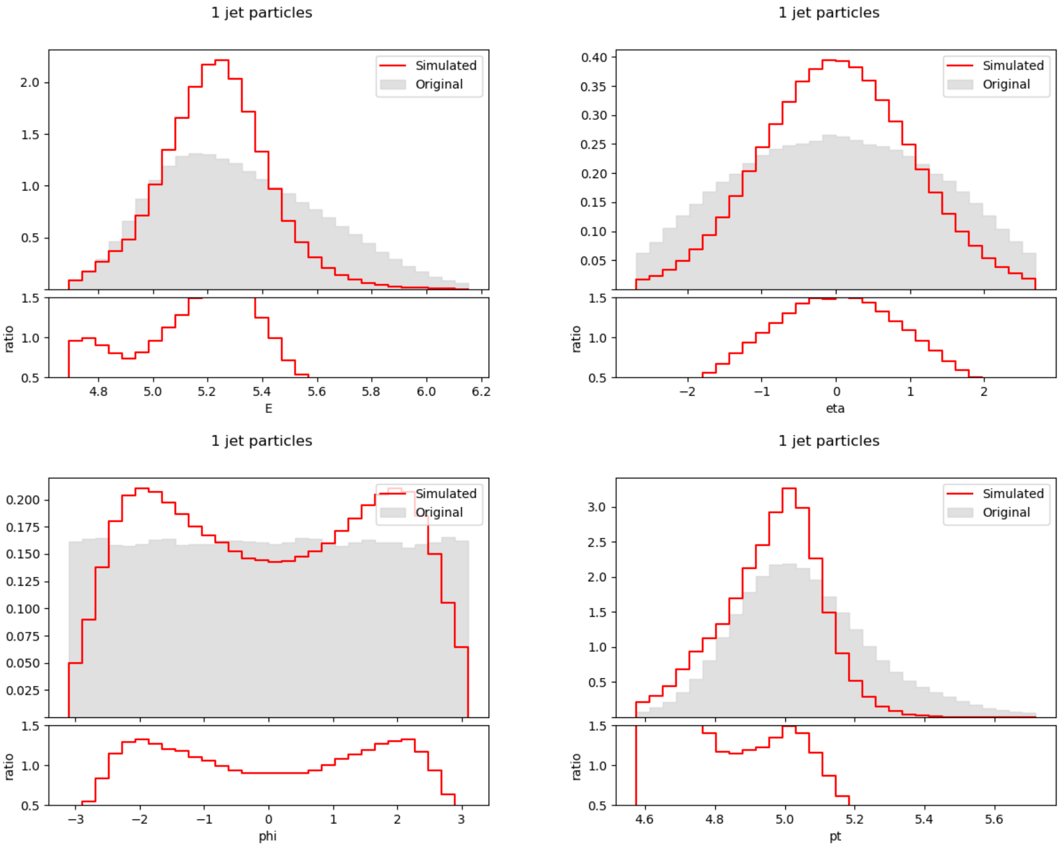


Figure E.61. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.1$).

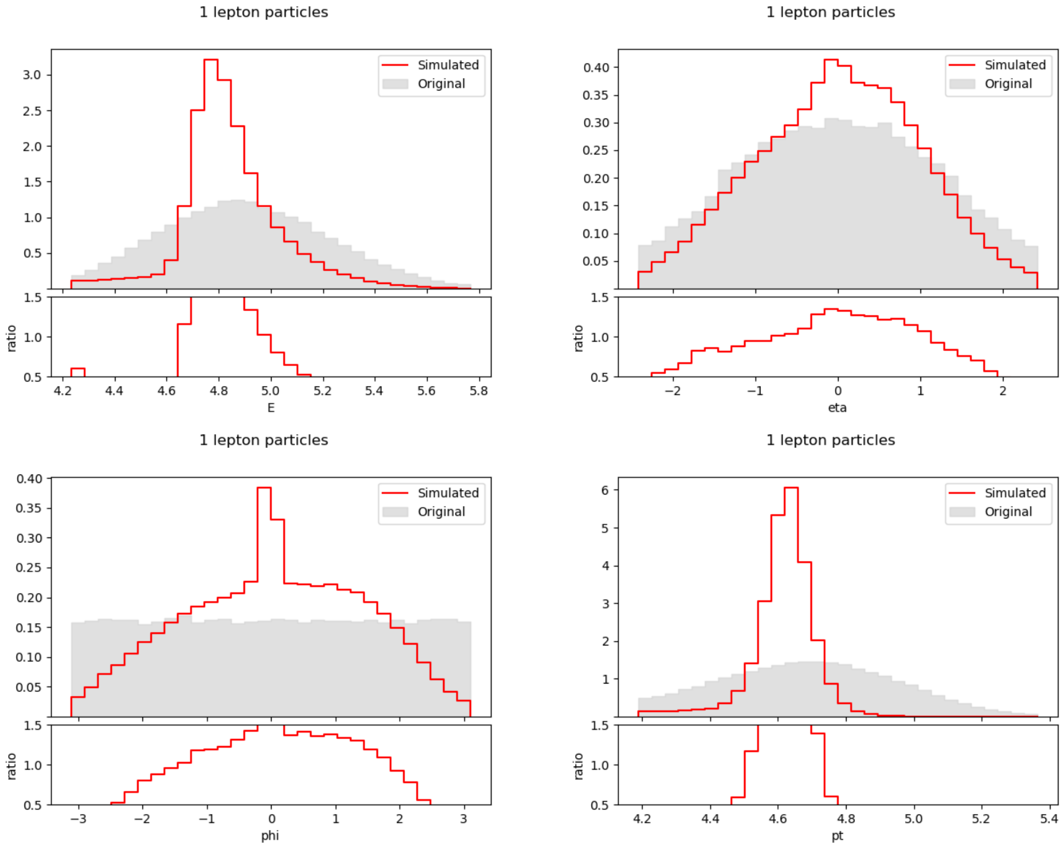


Figure E.62. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.1$).

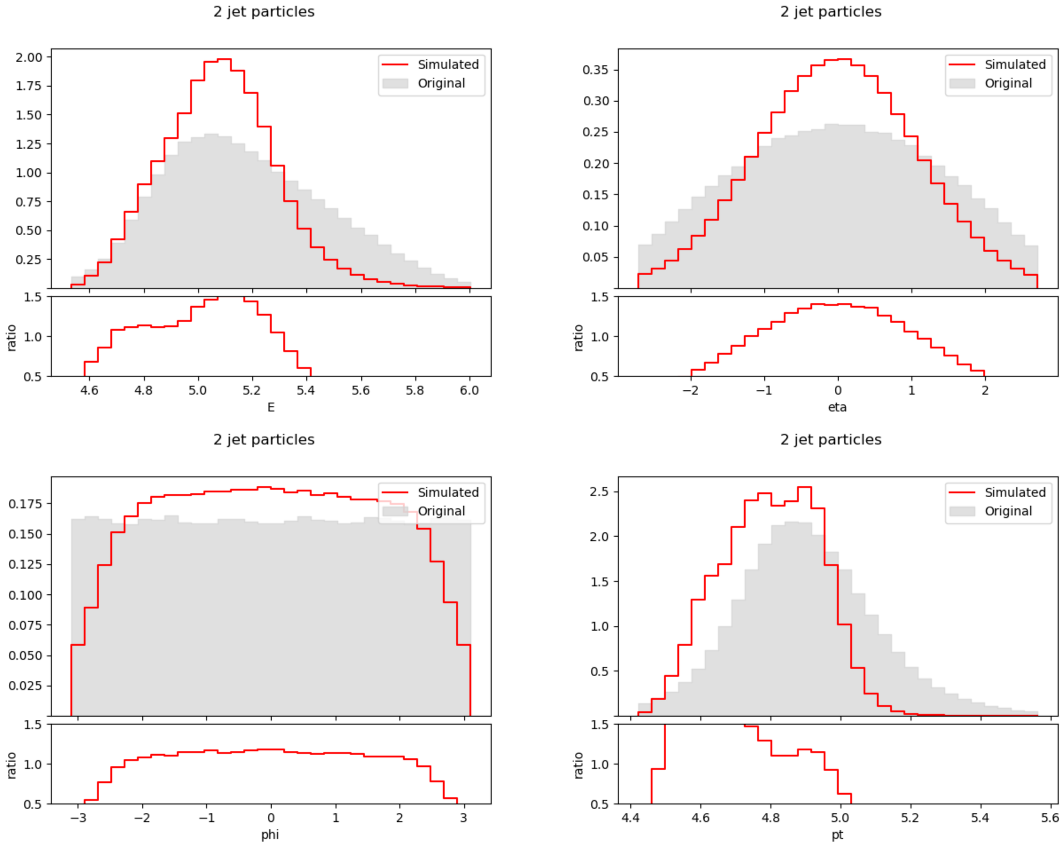


Figure E.63. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.1$).

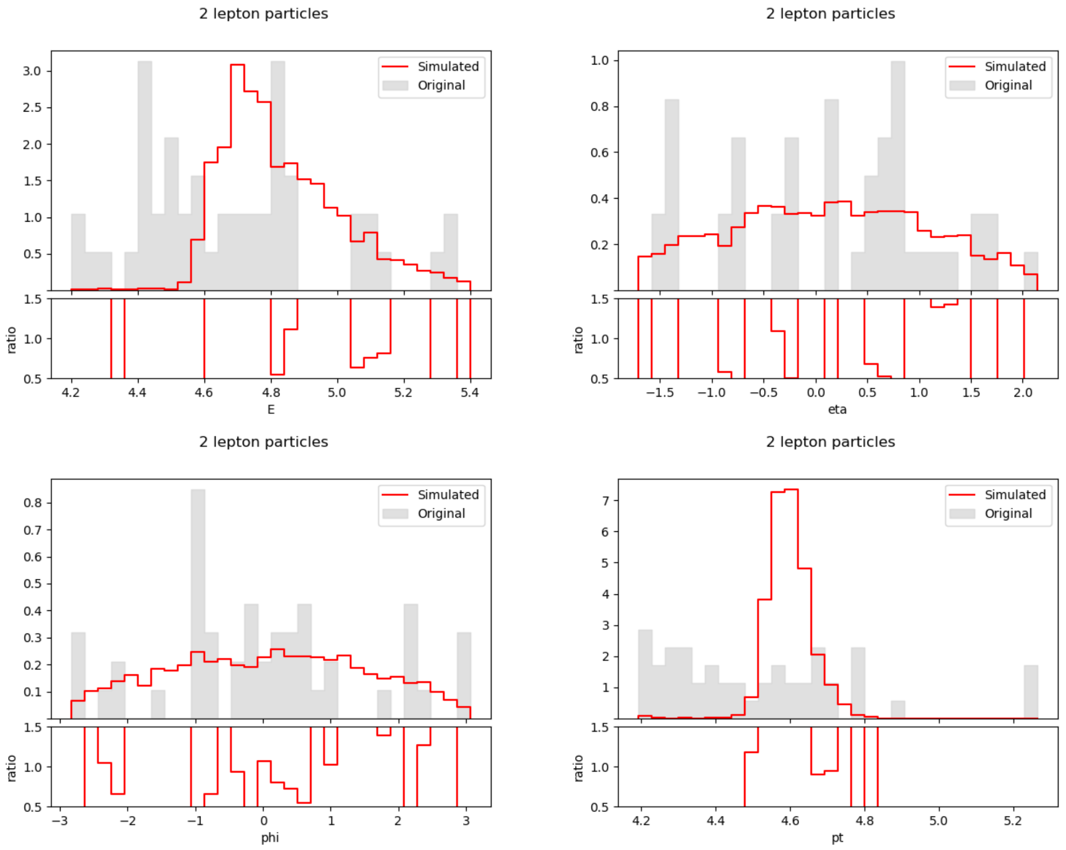


Figure E.64. Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.1$).

$\beta = 0.2, \gamma = 100$

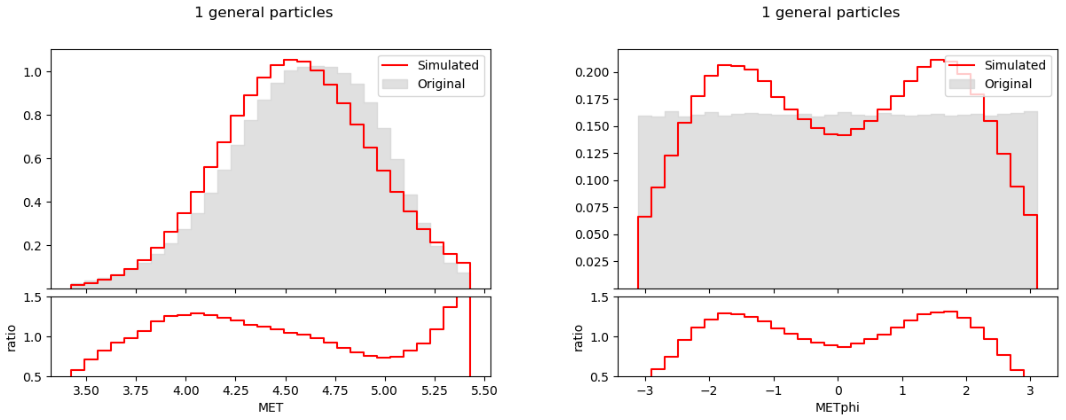


Figure E.65. Event `MET` and `METphi` distributions using a BGMM in experiment II ($\beta = 0.2$).

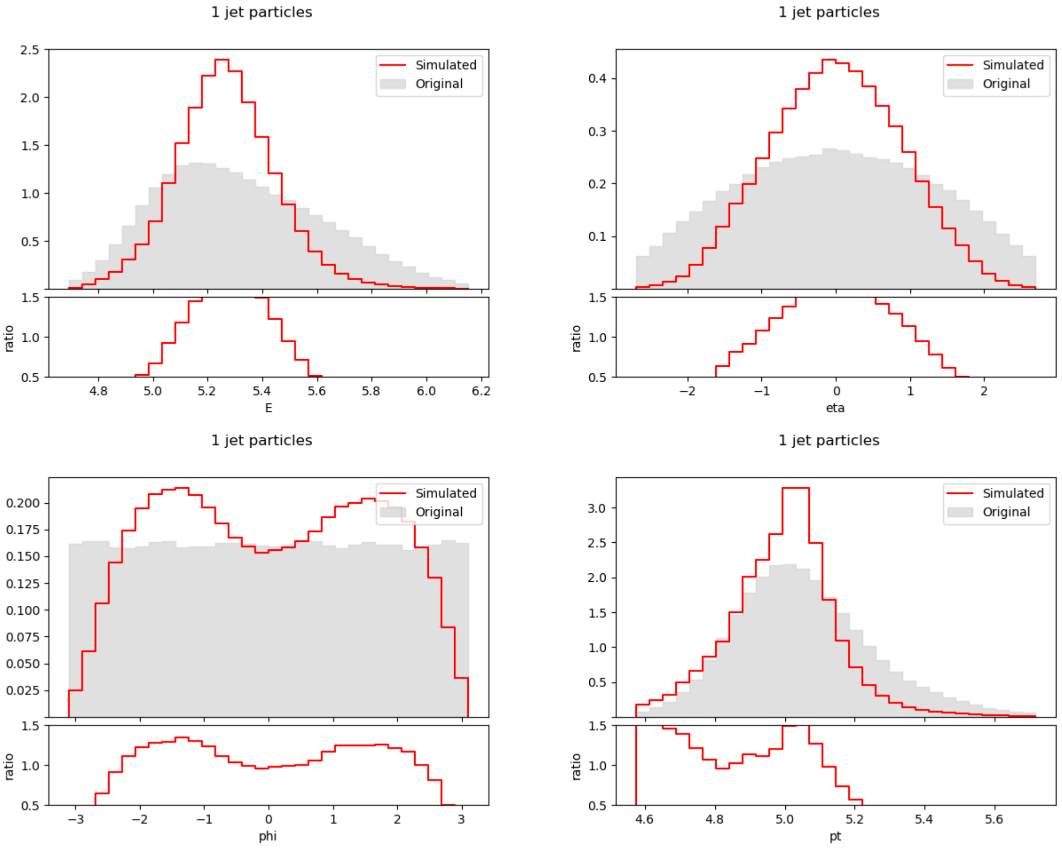


Figure E.66. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.2$).

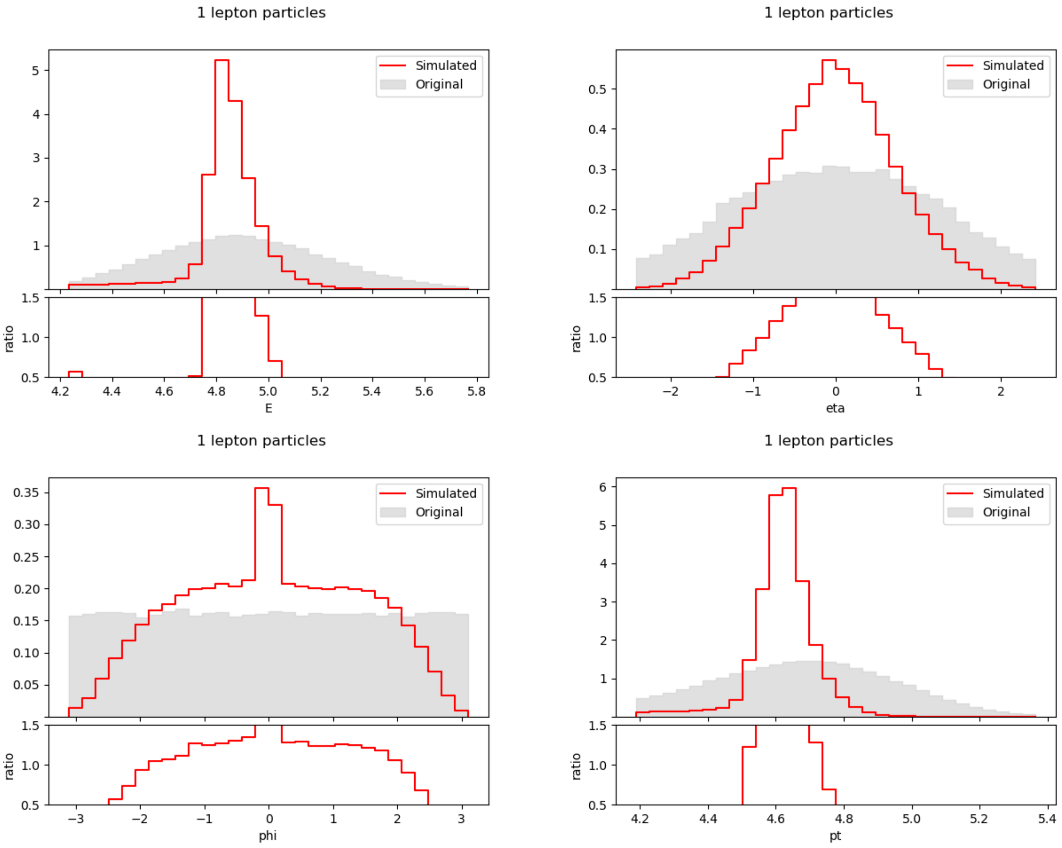


Figure E.67. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.2$).

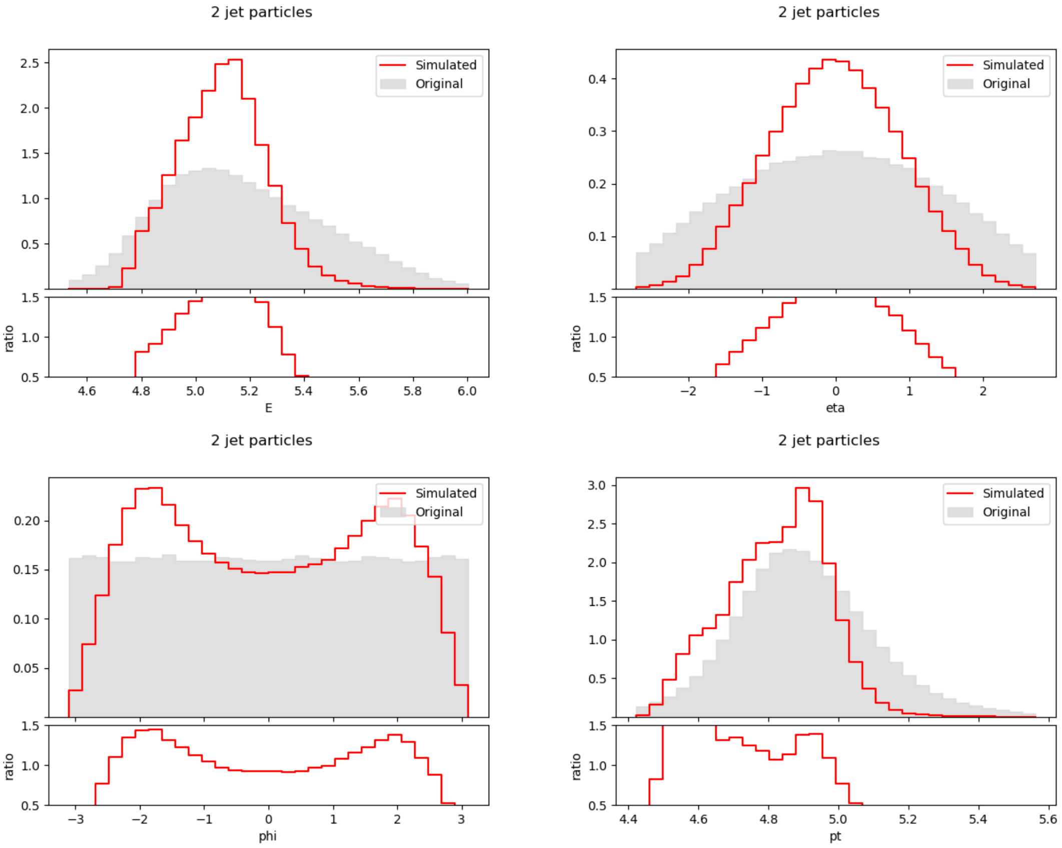


Figure E.68. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.2$).

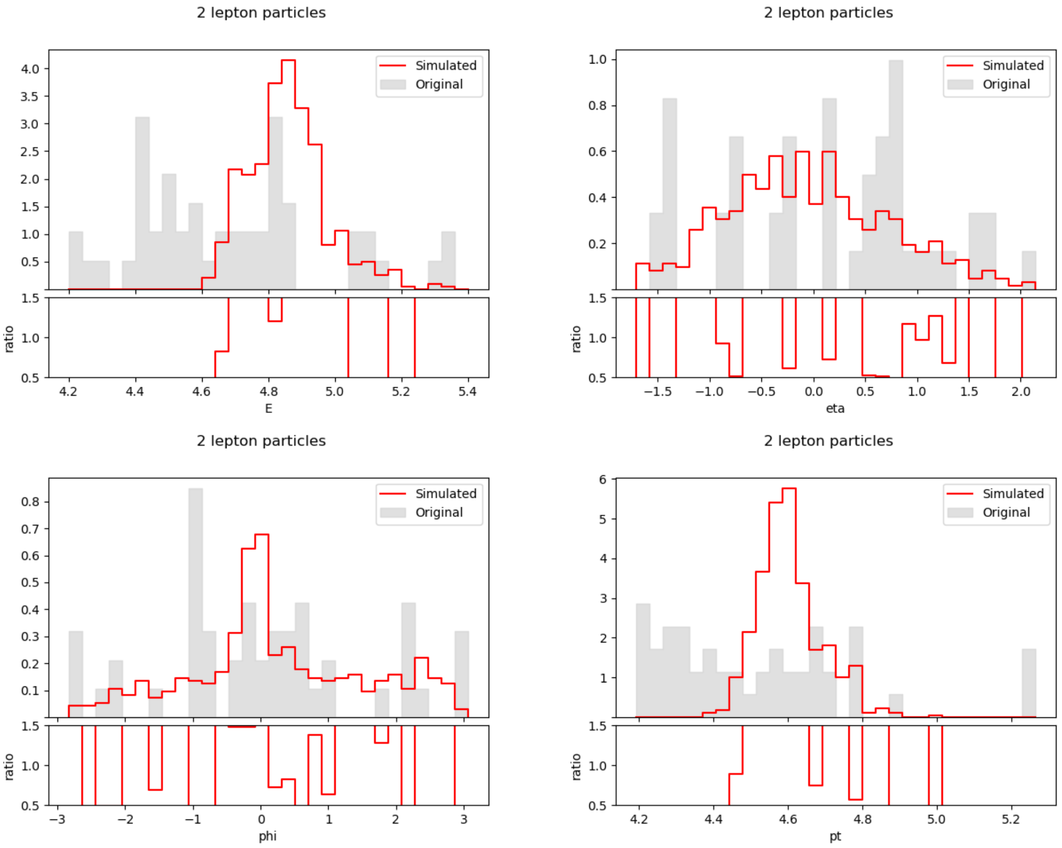


Figure E.69. Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.2$).

$\beta = 0.5, \gamma = 100$

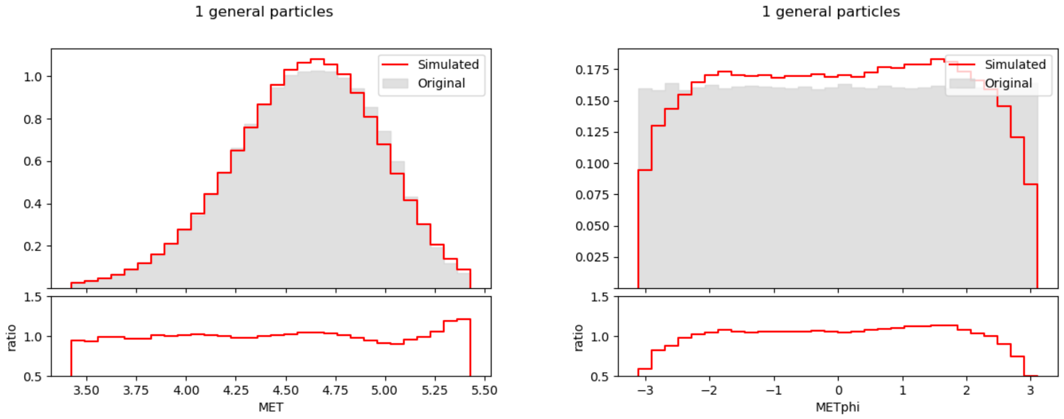


Figure E.70. Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.5$).

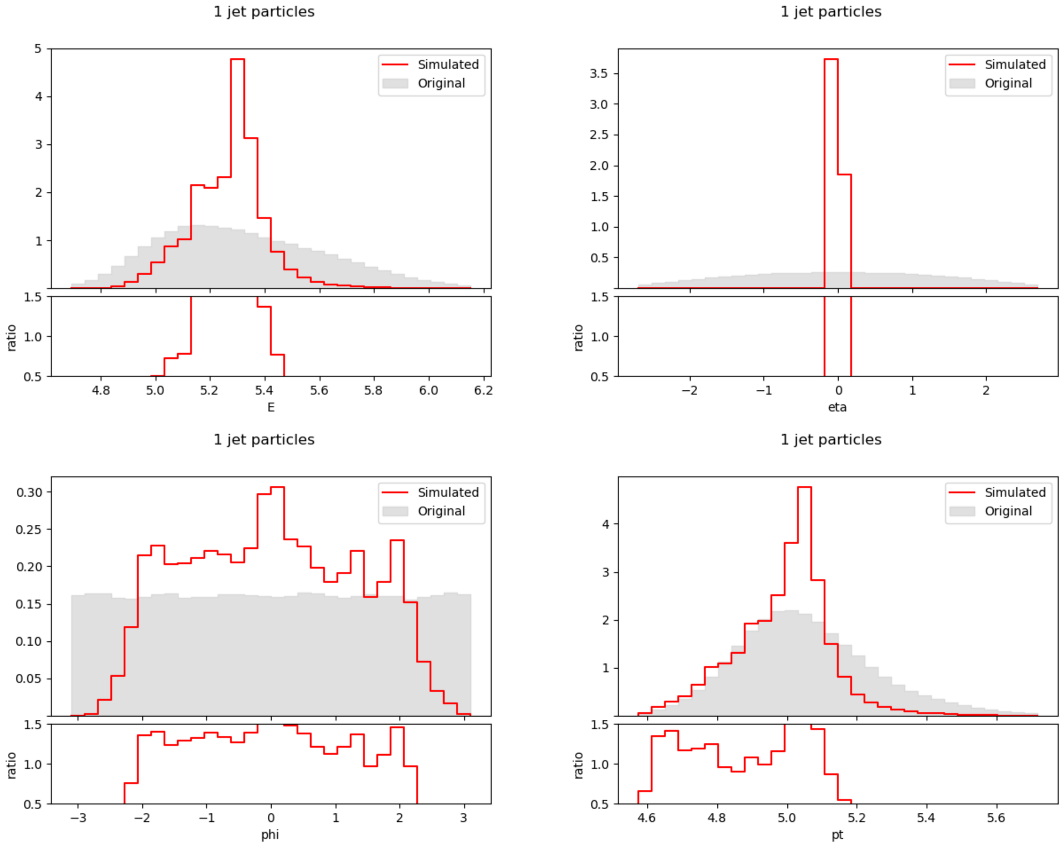


Figure E.71. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.5$).

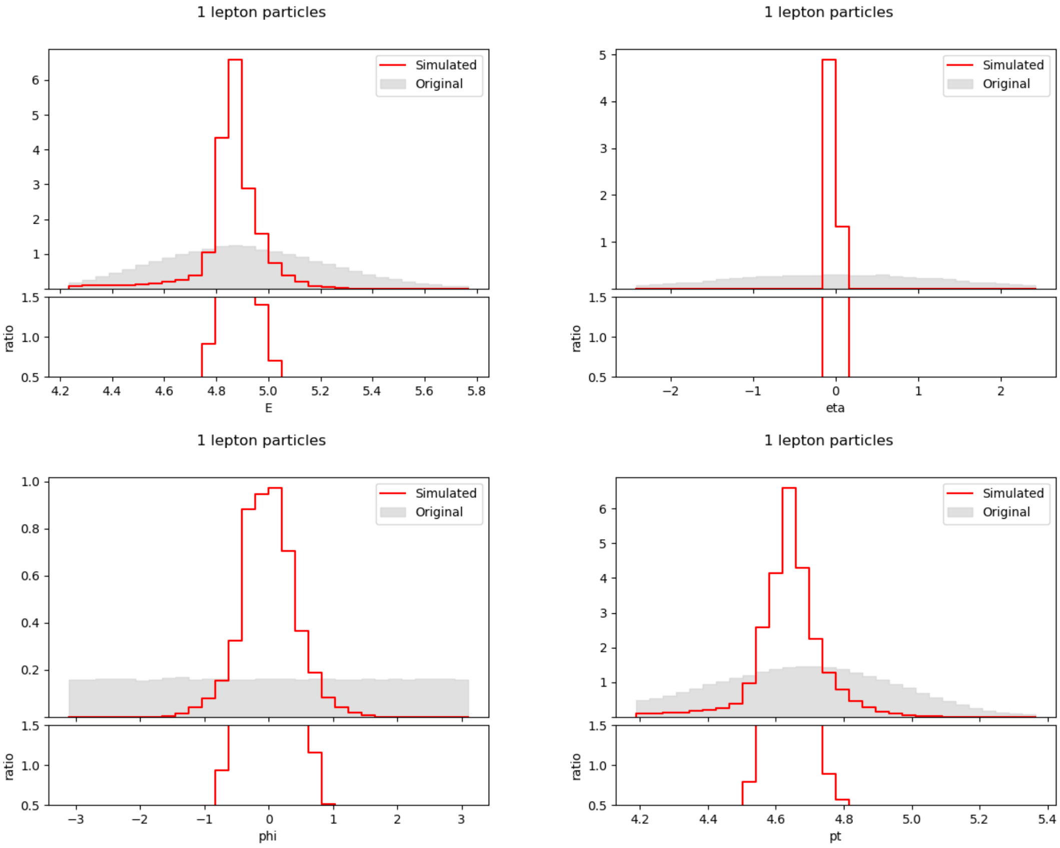


Figure E.72. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.5$).

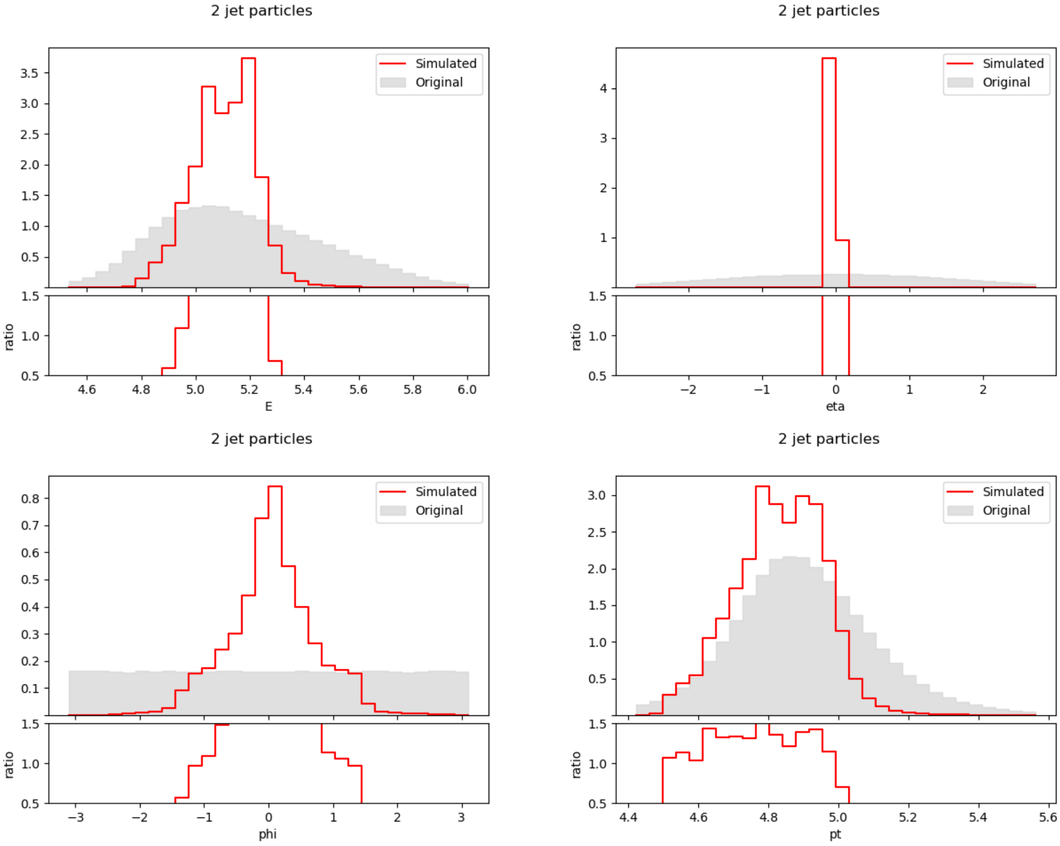


Figure E.73. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.5$).

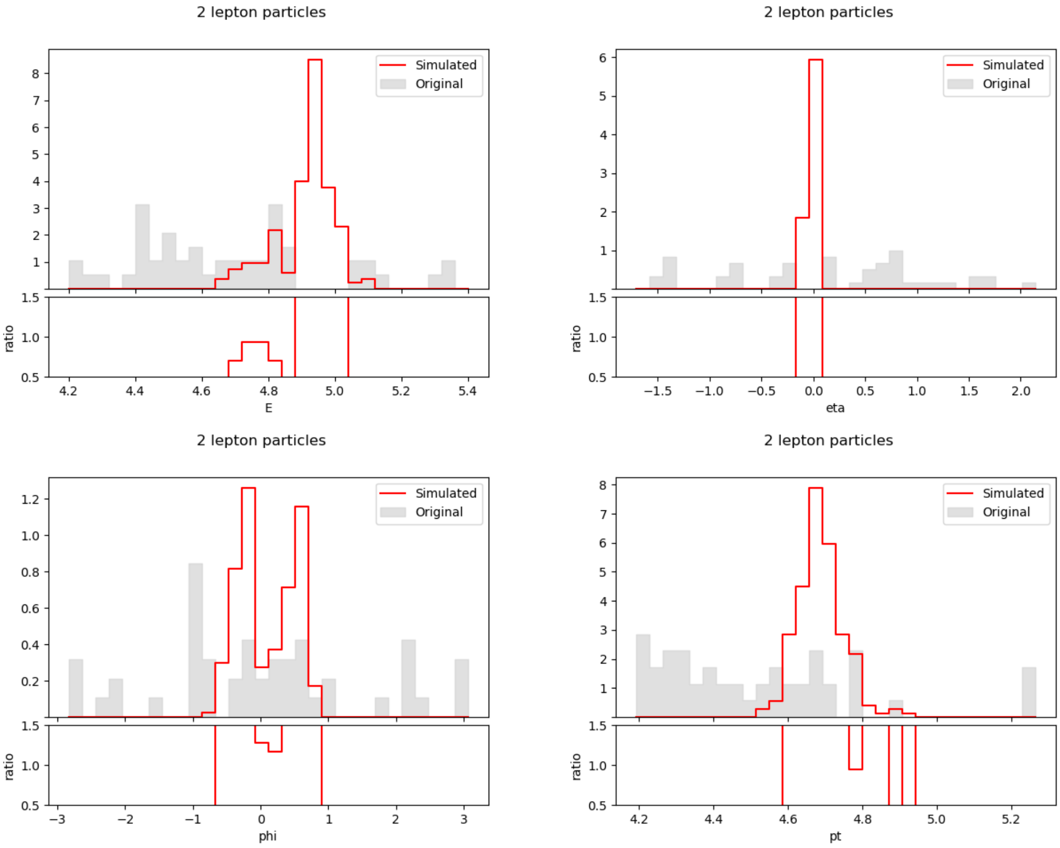


Figure E.74. Second lepton particle feature distributions using BGMM in experiment II ($\beta = 0.5$).

$\beta = 0.7, \gamma = 100$

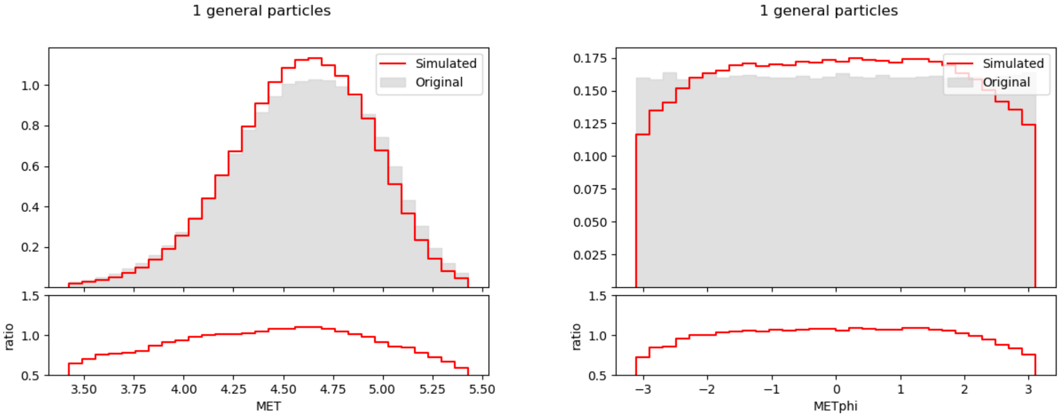


Figure E.75. Event MET and METphi distributions using a BGMM in experiment II ($\beta = 0.7$).

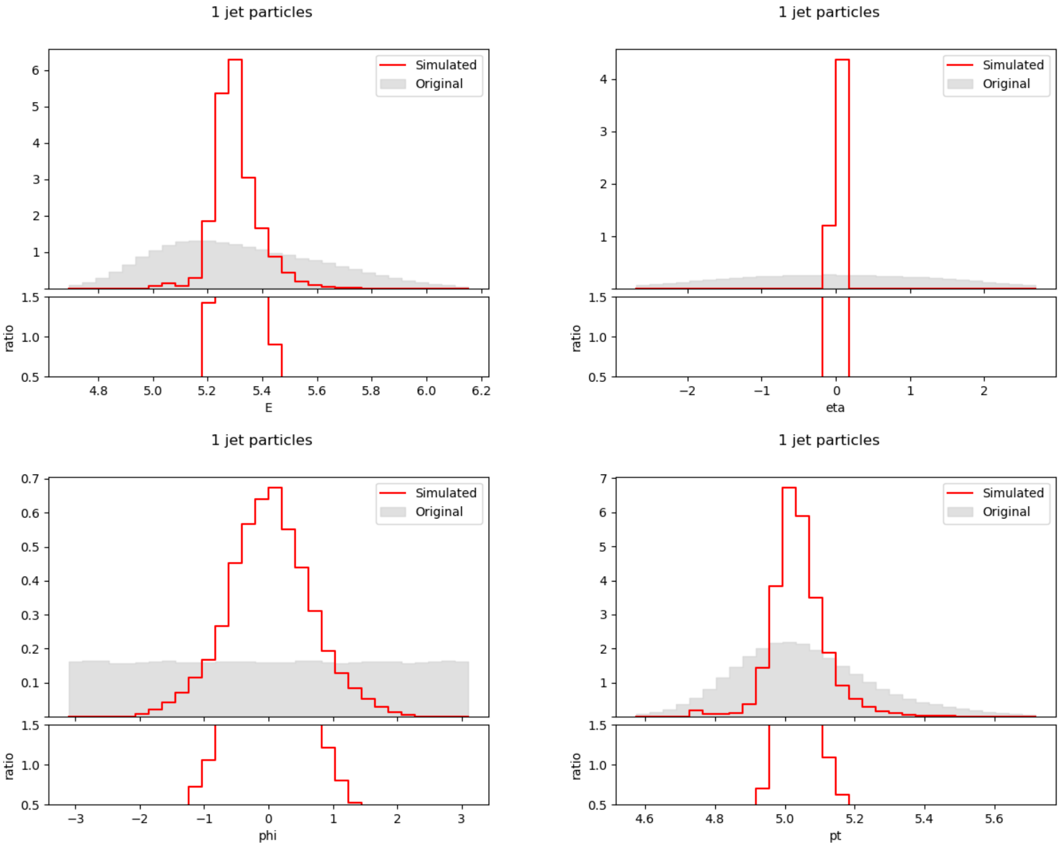


Figure E.76. First jet particle feature distributions using a BGMM in experiment II ($\beta = 0.7$).

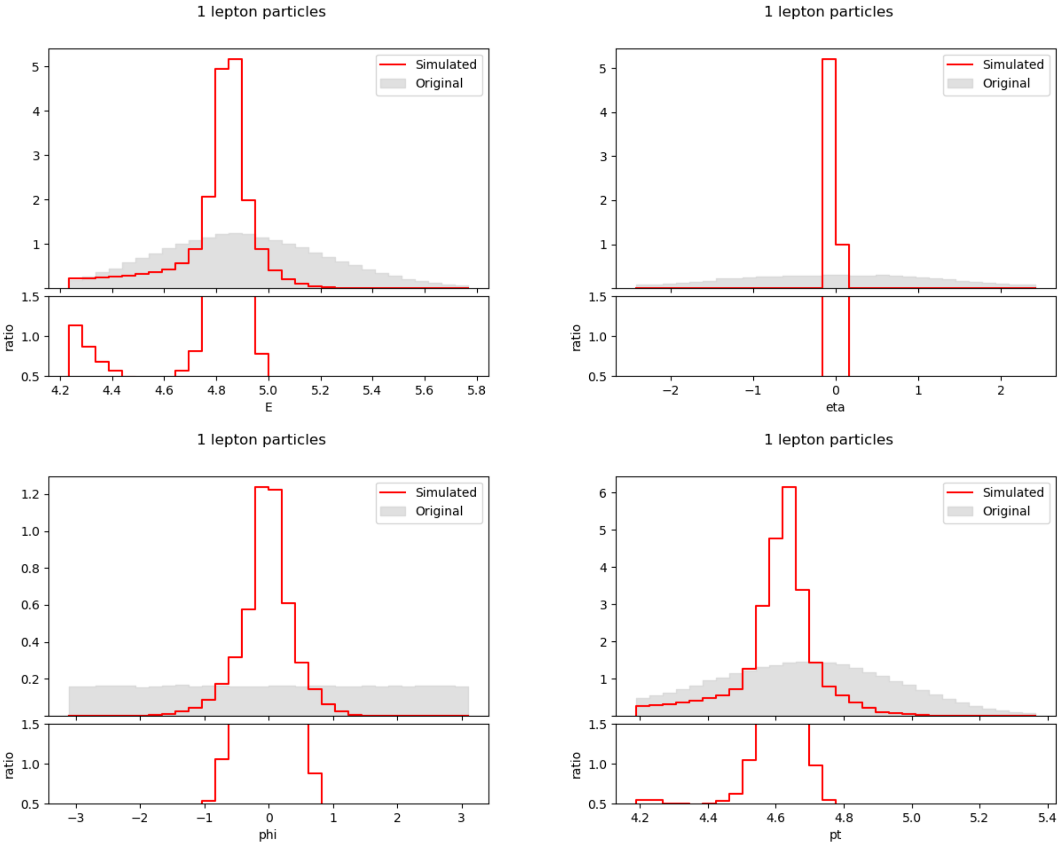


Figure E.77. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 0.7$).

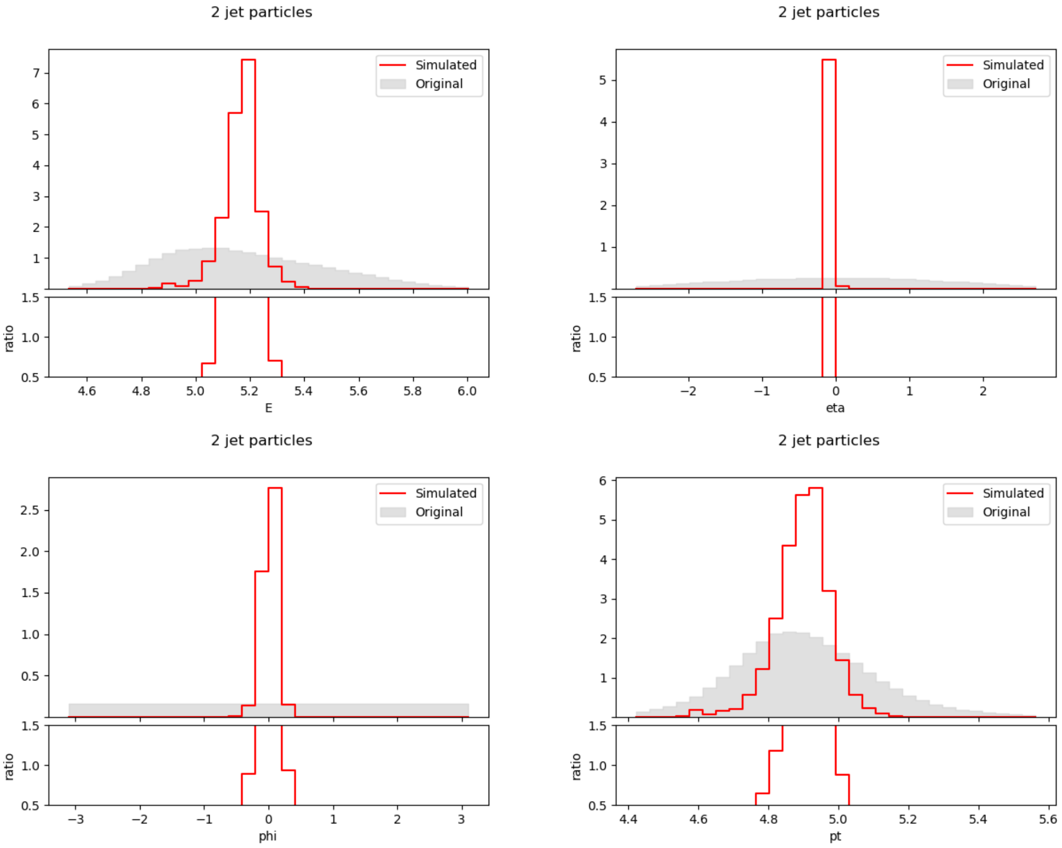


Figure E.78. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 0.7$).

No second leptons were present in the simulated events using $\beta = 0.7$.

$\beta = 1.0, \gamma = 100$

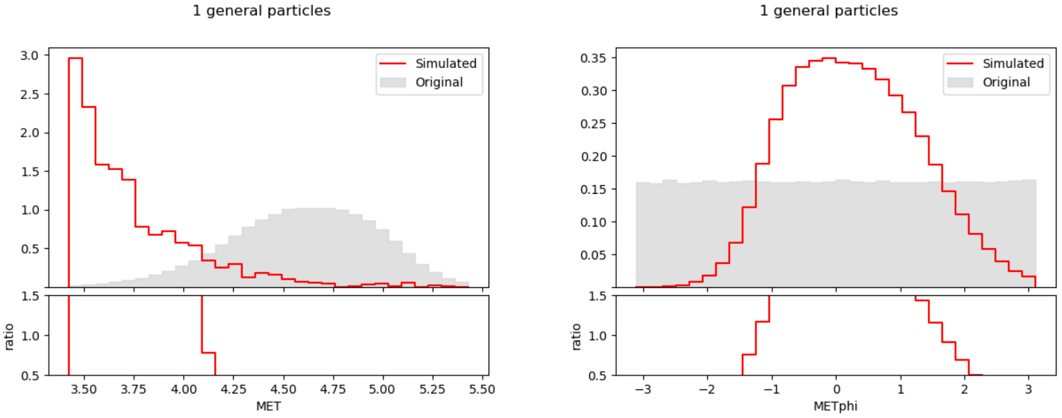


Figure E.79. Event `MET` and `METphi` distributions using a BGMM in experiment II ($\beta = 1.0$).

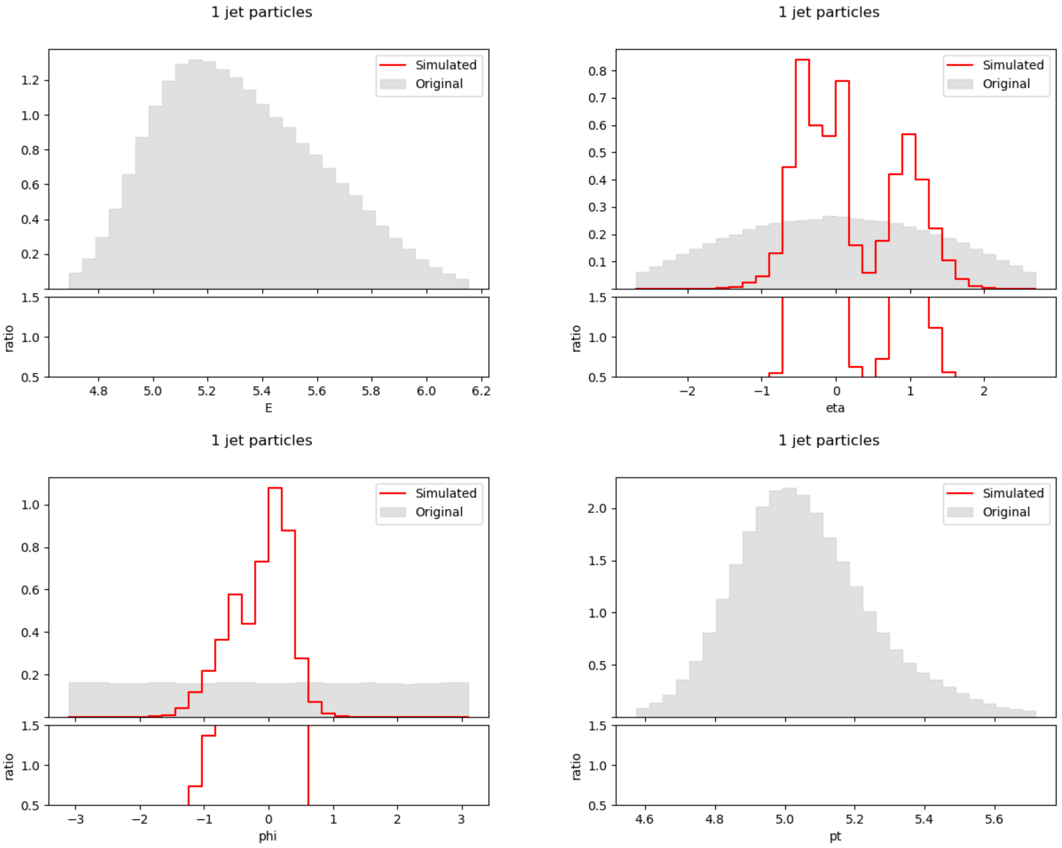


Figure E.80. First jet particle feature distributions using a BGMM in experiment II ($\beta = 1.0$).

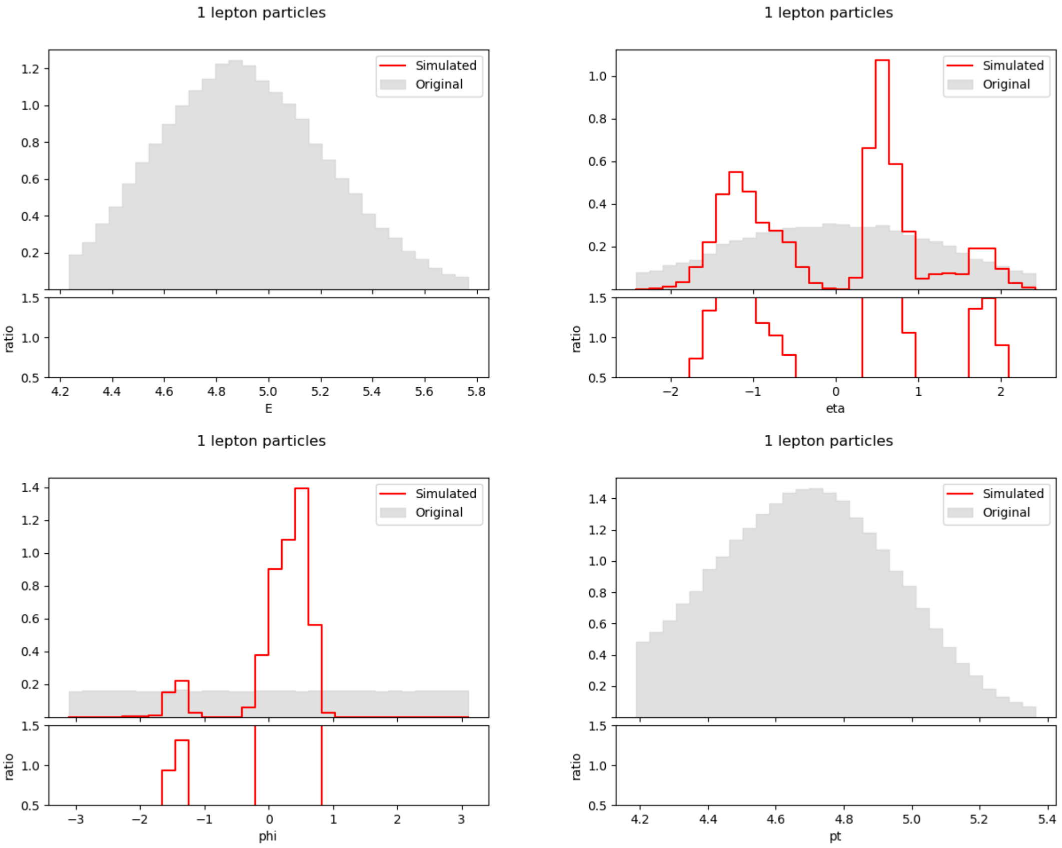


Figure E.81. First lepton particle feature distributions using a BGMM in experiment II ($\beta = 1.0$).

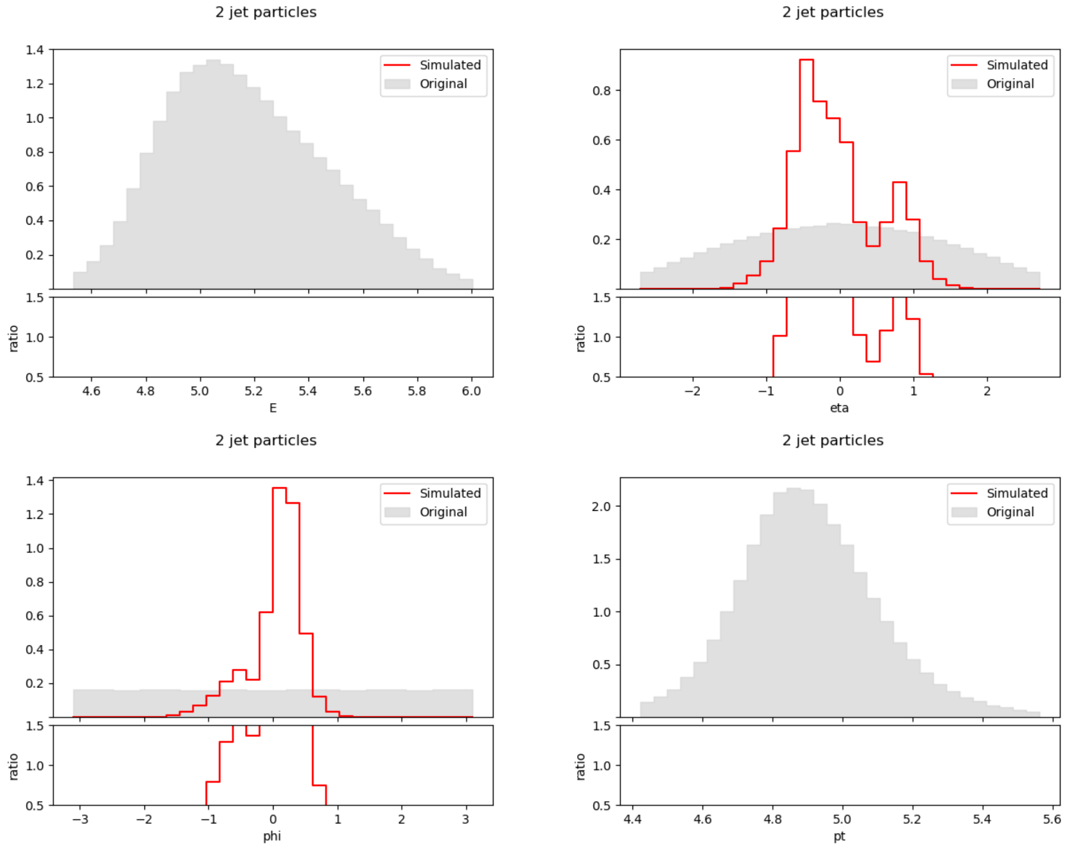


Figure E.82. Second jet particle feature distributions using a BGMM in experiment II ($\beta = 1.0$).

No second leptons were present in the simulated events using $\beta = 1.0$.

E.3 Experiment III

E.3.1. Without BGMMs

$\alpha = 0.1$

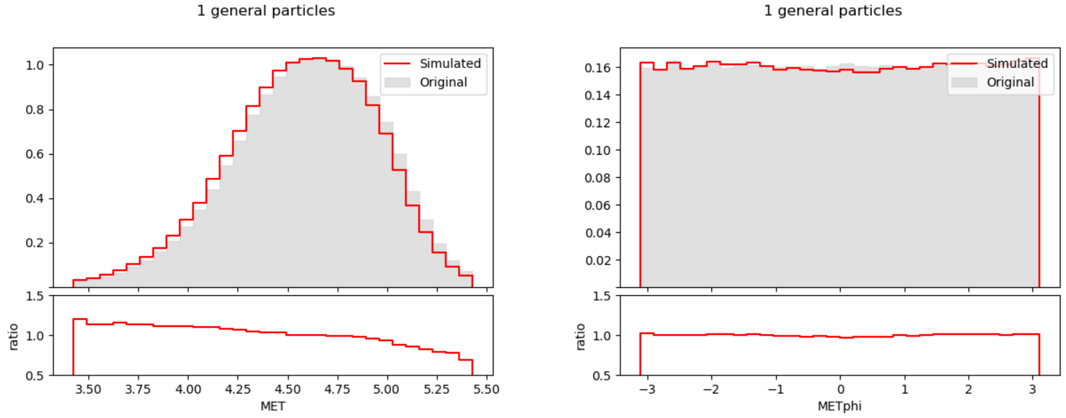


Figure E.83. Event `MET` and `METphi` distributions in experiment III ($\alpha = 0.1$).

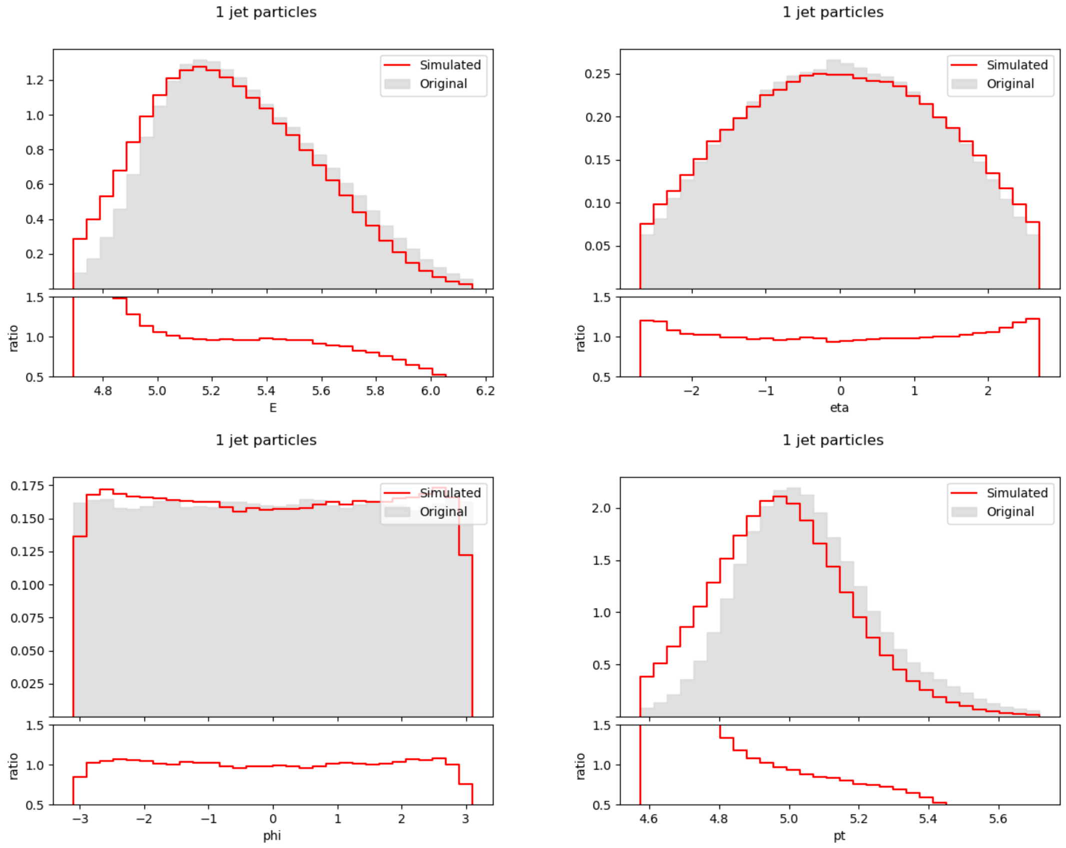


Figure E.84. First jet particle feature distributions in experiment III ($\alpha = 0.1$).

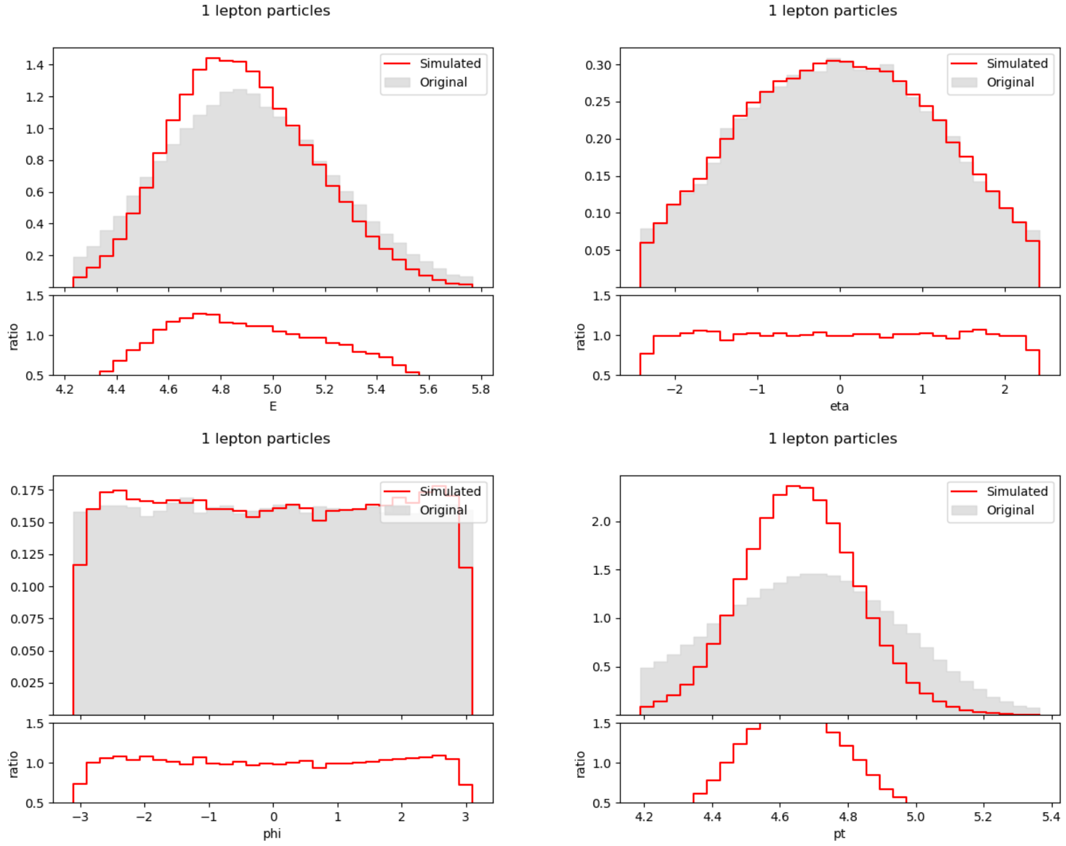


Figure E.85. First lepton particle feature distributions in experiment III ($\alpha = 0.1$).

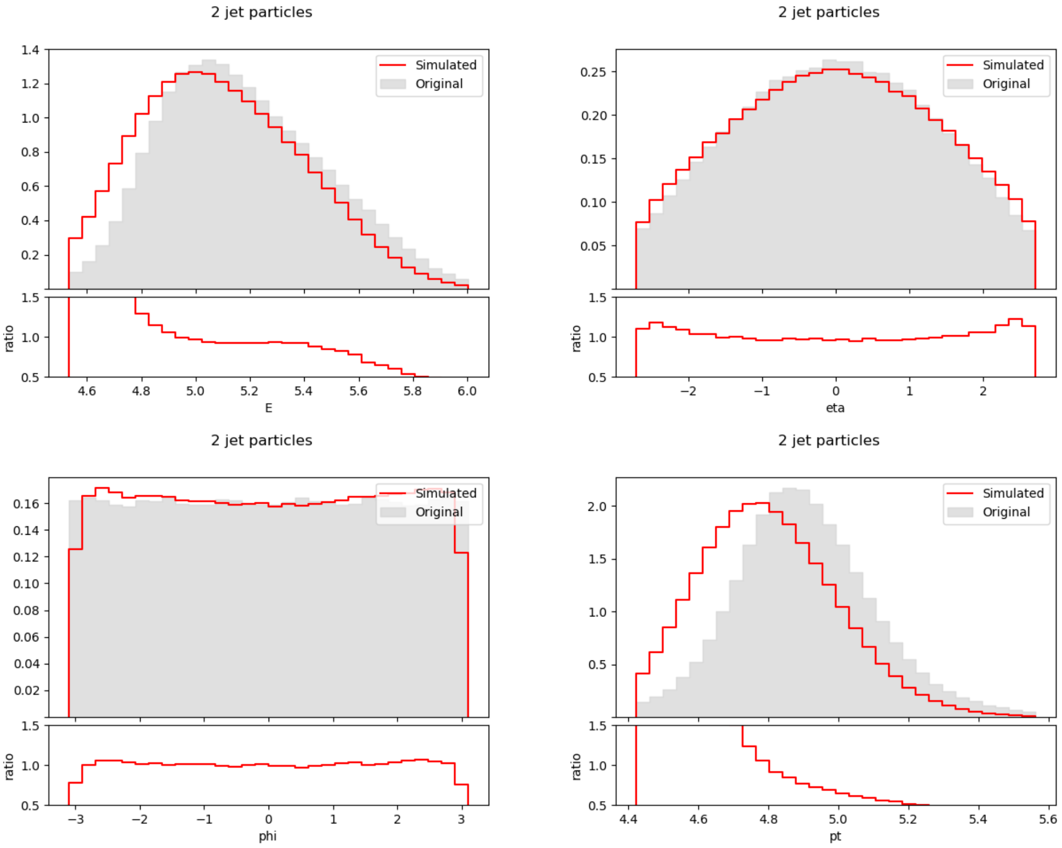


Figure E.86. Second jet particle feature distributions in experiment III ($\alpha = 0.1$).

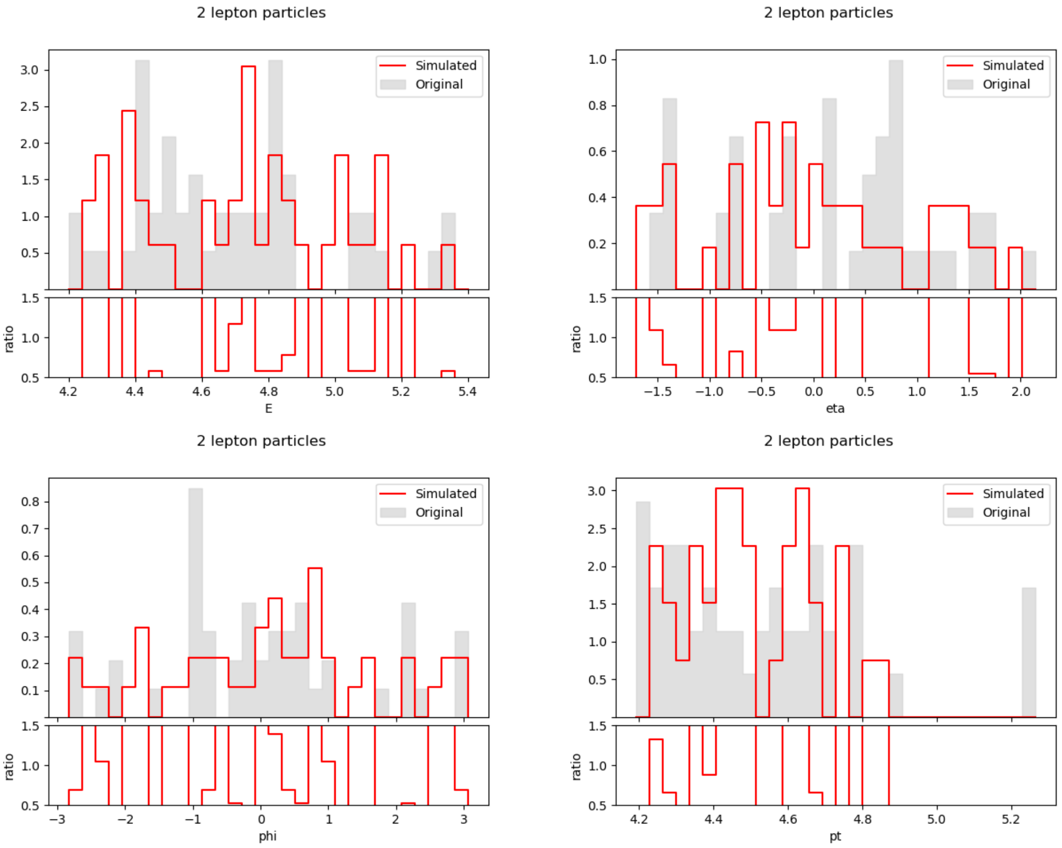


Figure E.87. Second lepton particle feature distributions in experiment III ($\alpha = 0.1$).

$\alpha = 0.2$

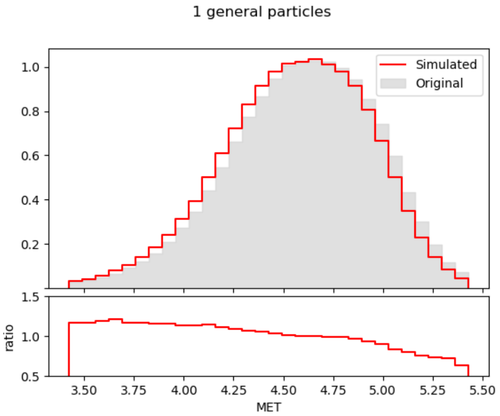


Figure E.88. Event MET and METphi distributions in experiment III ($\alpha = 0.2$).

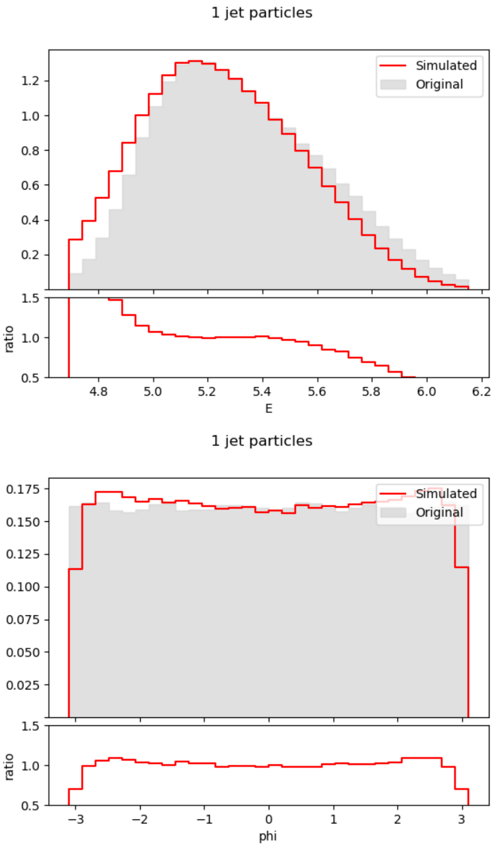


Figure E.89. First jet particle feature distributions in experiment III ($\alpha = 0.2$).

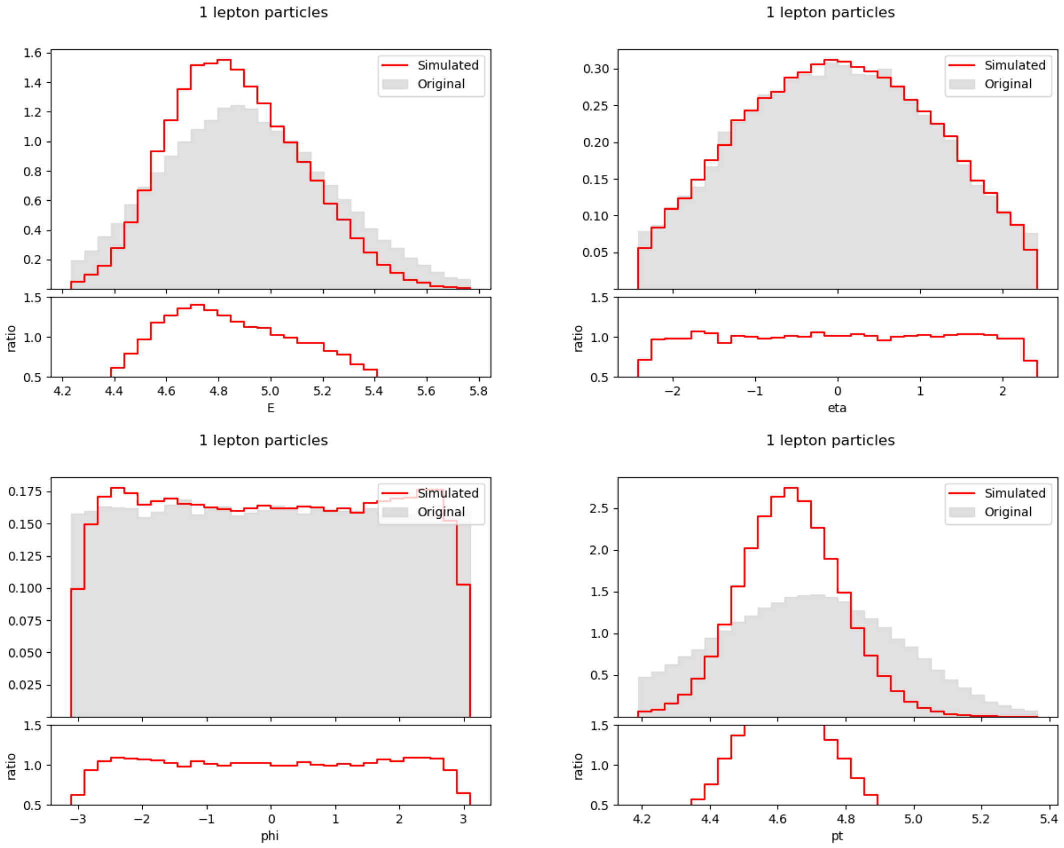


Figure E.90. First lepton particle feature distributions in experiment III ($\alpha = 0.2$).

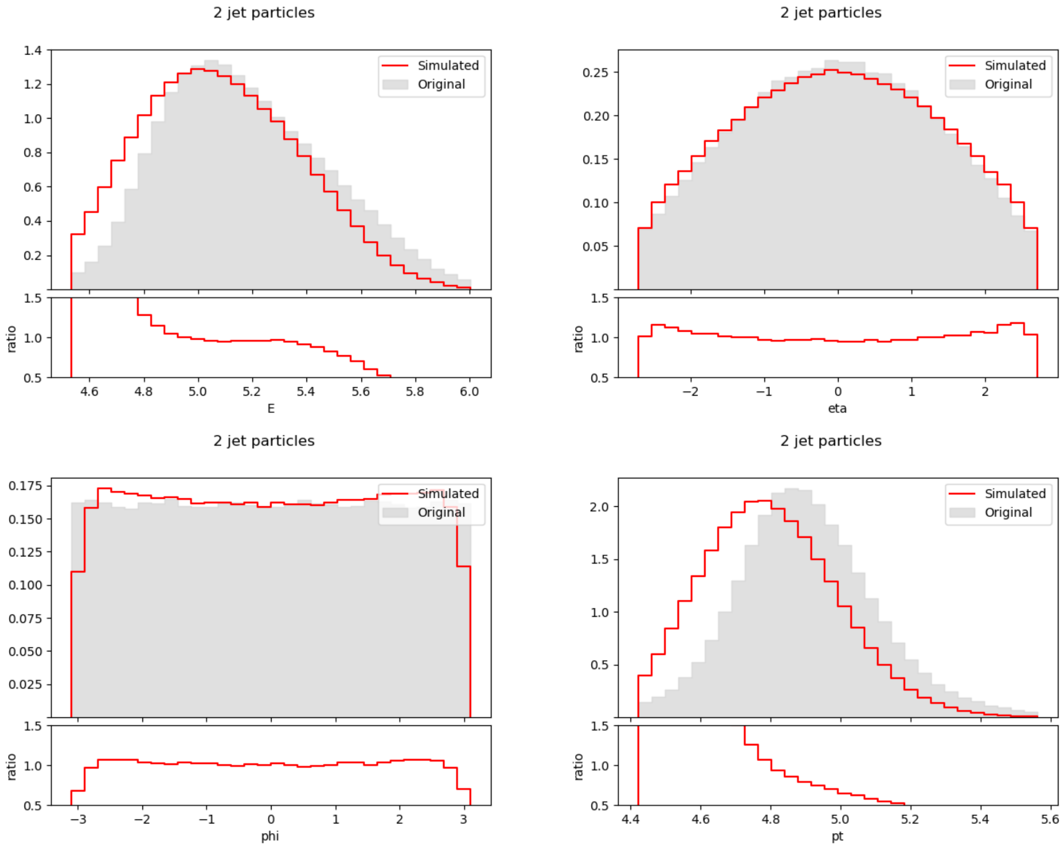


Figure E.91. Second jet particle feature distributions in experiment III ($\alpha = 0.2$).

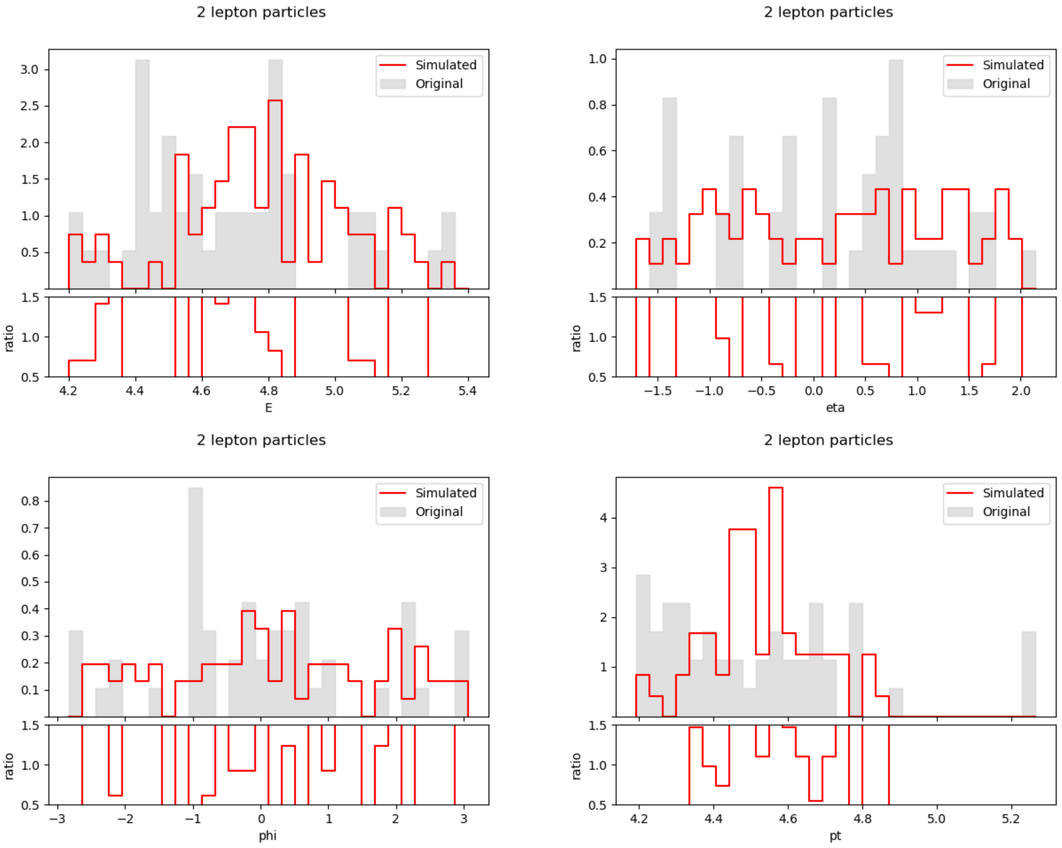


Figure E.92. Second lepton particle feature distributions in experiment III ($\alpha = 0.2$).

$\alpha = 0.3$

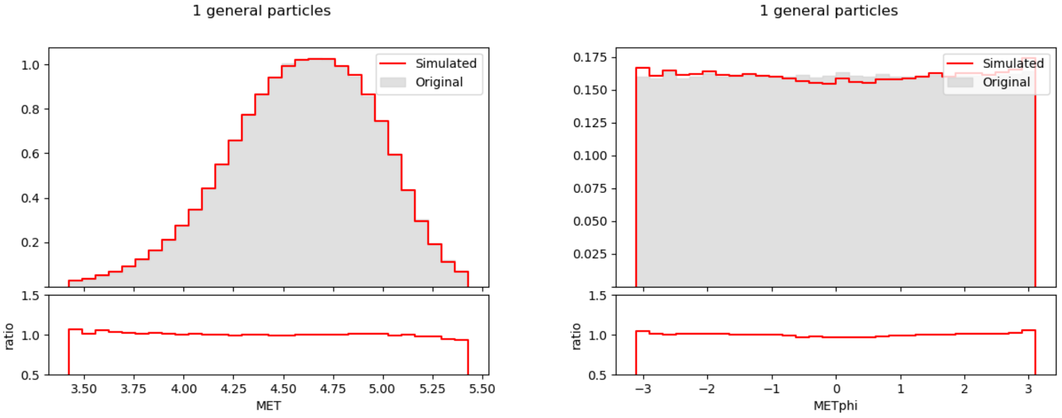


Figure E.93. Event MET and METphi distributions in experiment III ($\alpha = 0.3$).

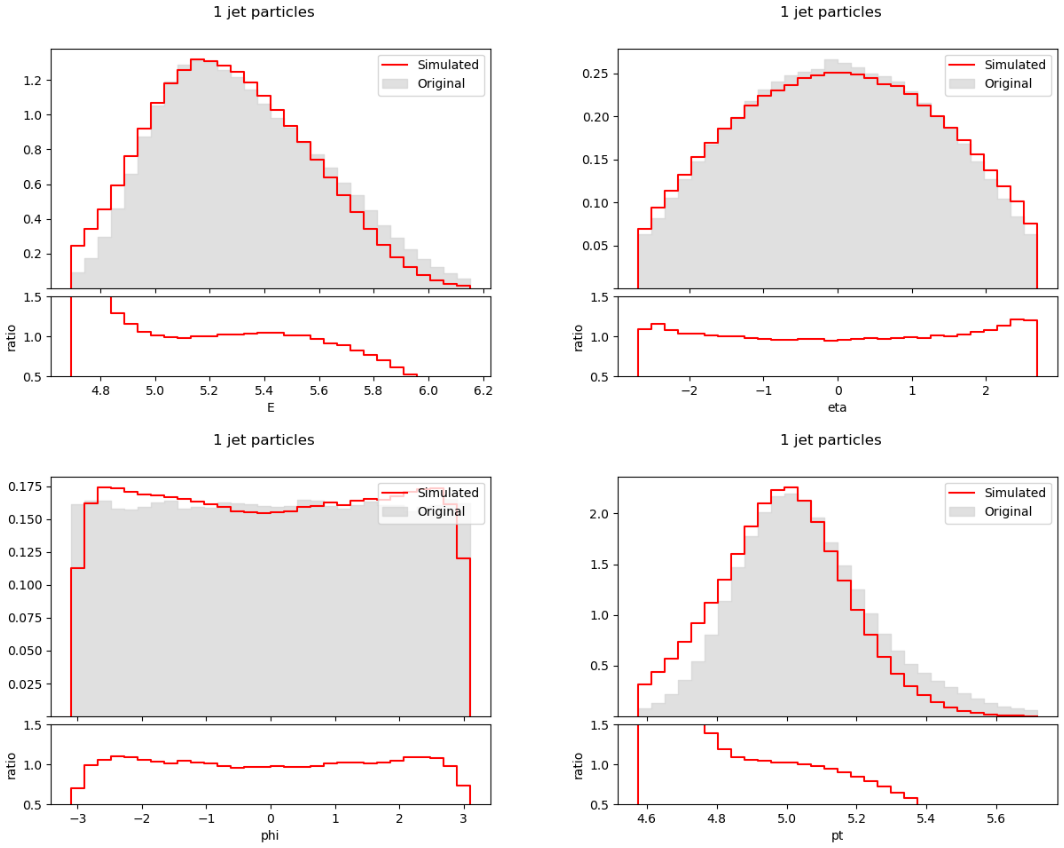


Figure E.94. First jet particle feature distributions in experiment III ($\alpha = 0.3$).

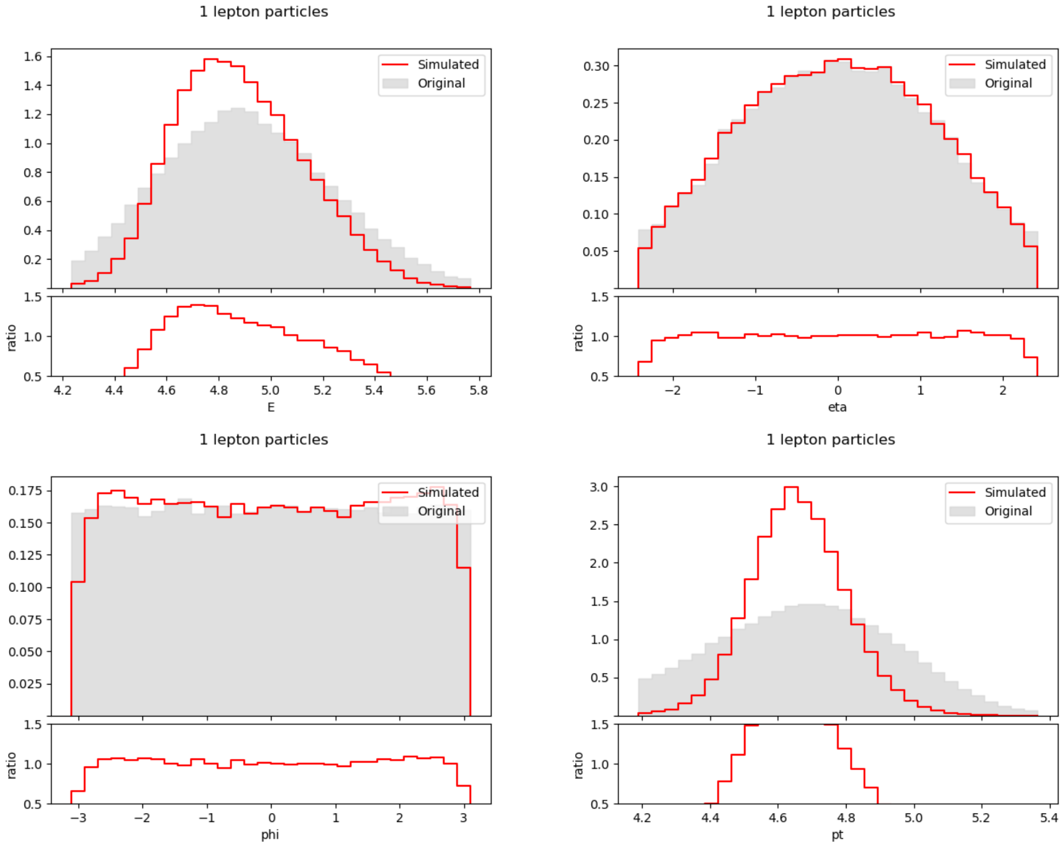


Figure E.95. First lepton particle feature distributions in experiment III ($\alpha = 0.3$).

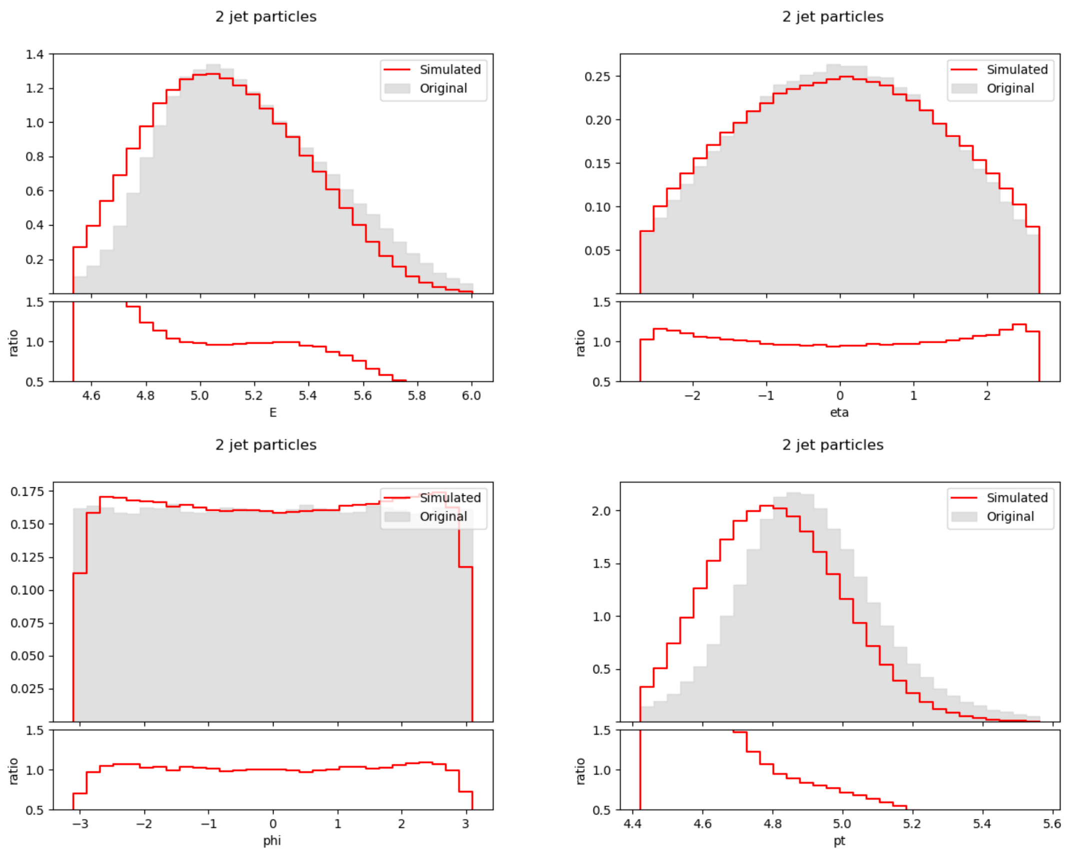


Figure E.96. Second jet particle feature distributions in experiment III ($\alpha = 0.3$).

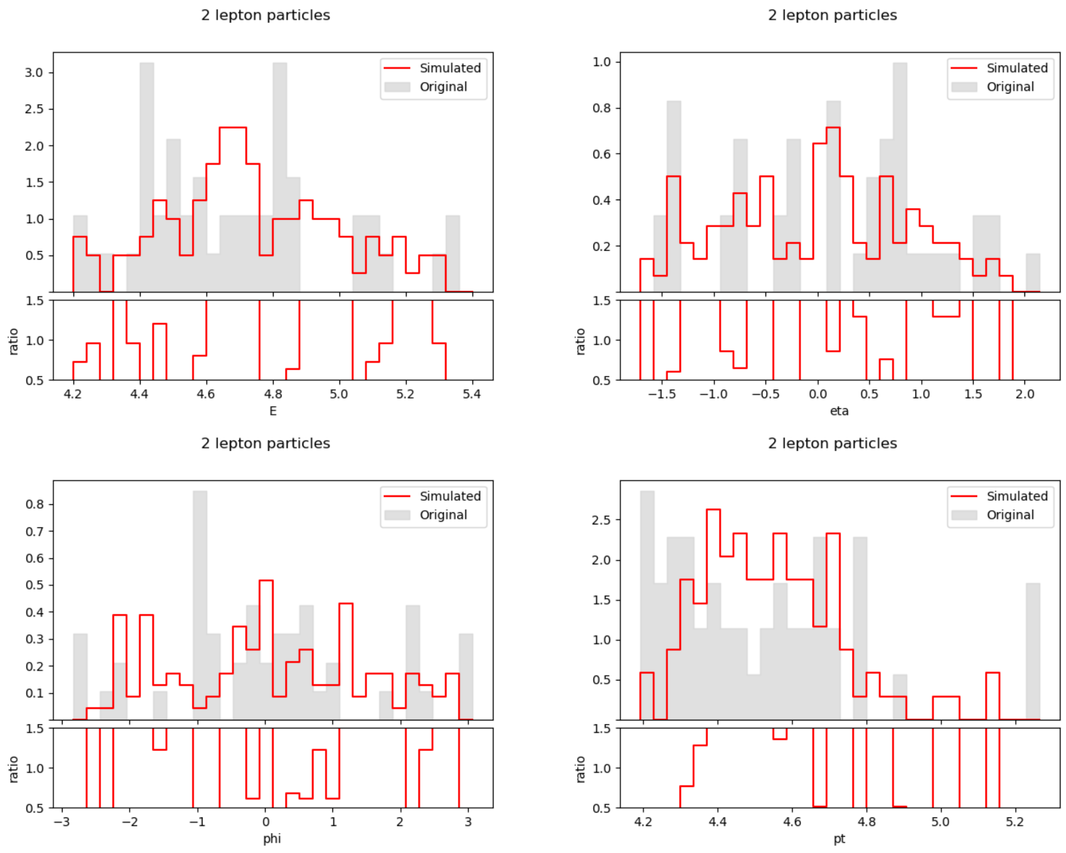


Figure E.97. Second lepton particle feature distributions in experiment III ($\alpha = 0.3$).

E.3.2. With BGMMs

$\alpha = 0.1, \gamma = 10$

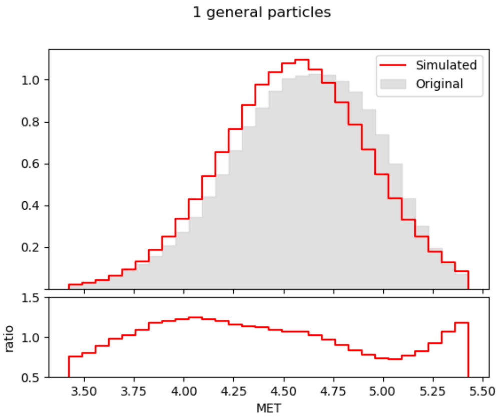
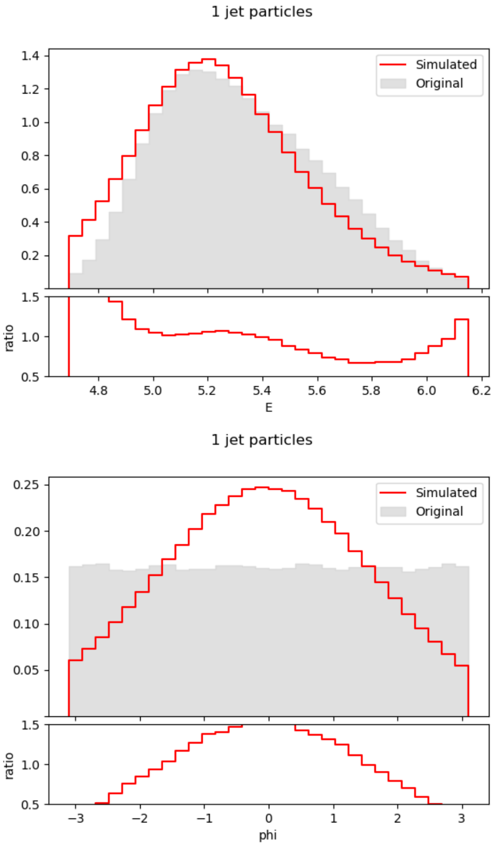
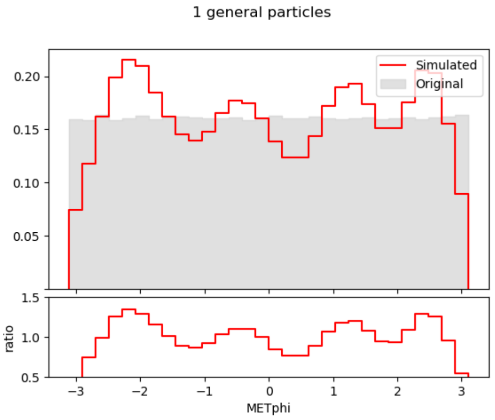


Figure E.98. Event MET and METphi distributions using a BGMM ($\alpha = 0.1, \gamma = 10$).





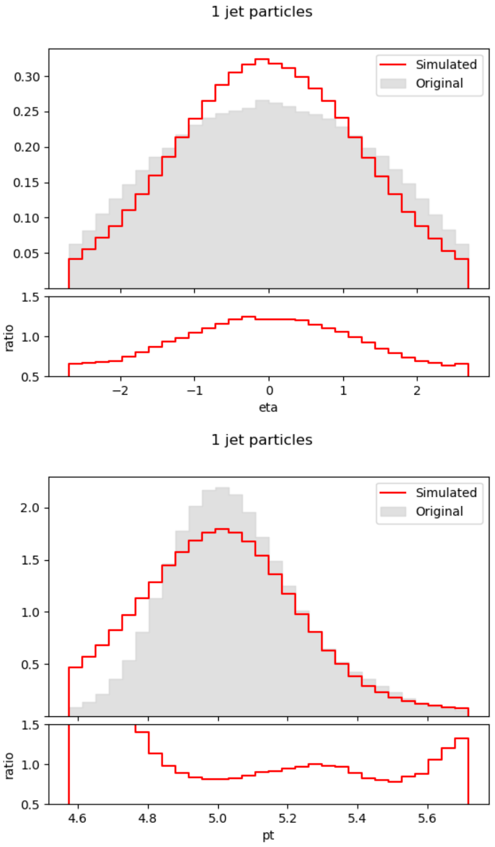


Figure E.99. First jet particle feature distributions using a BGMM ($\alpha = 0.1, \gamma = 10$).

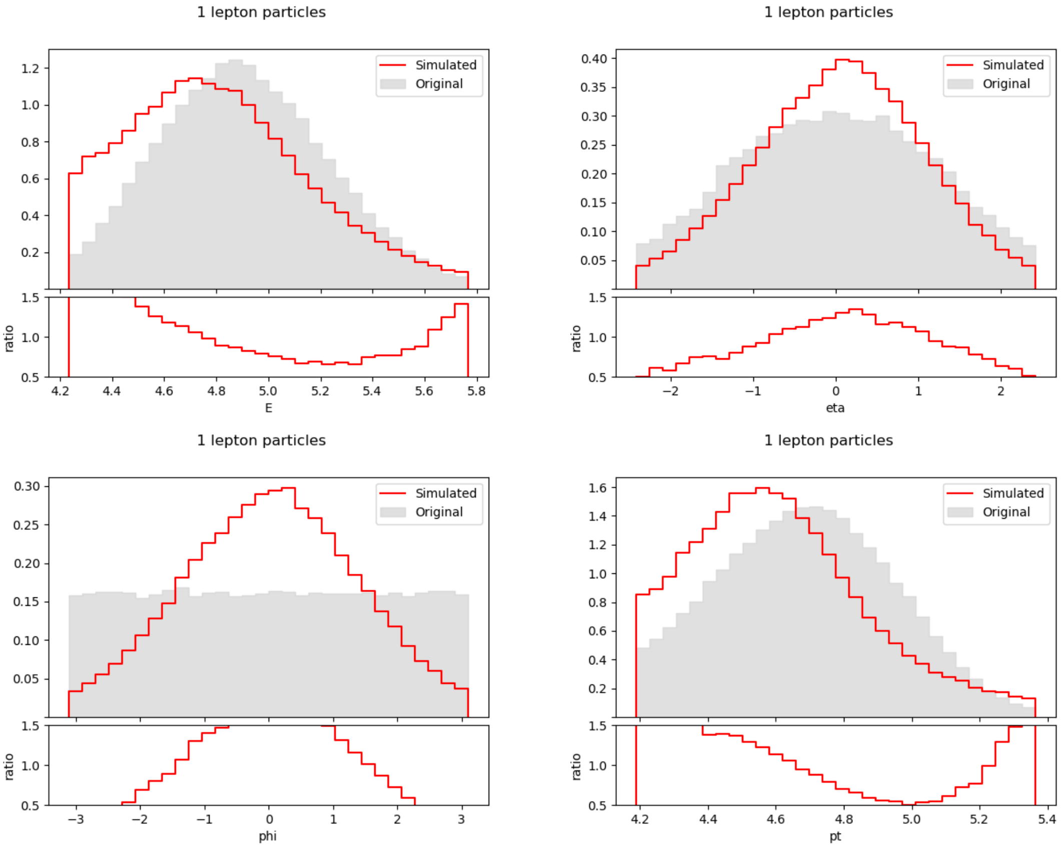


Figure E.100. First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 10$).

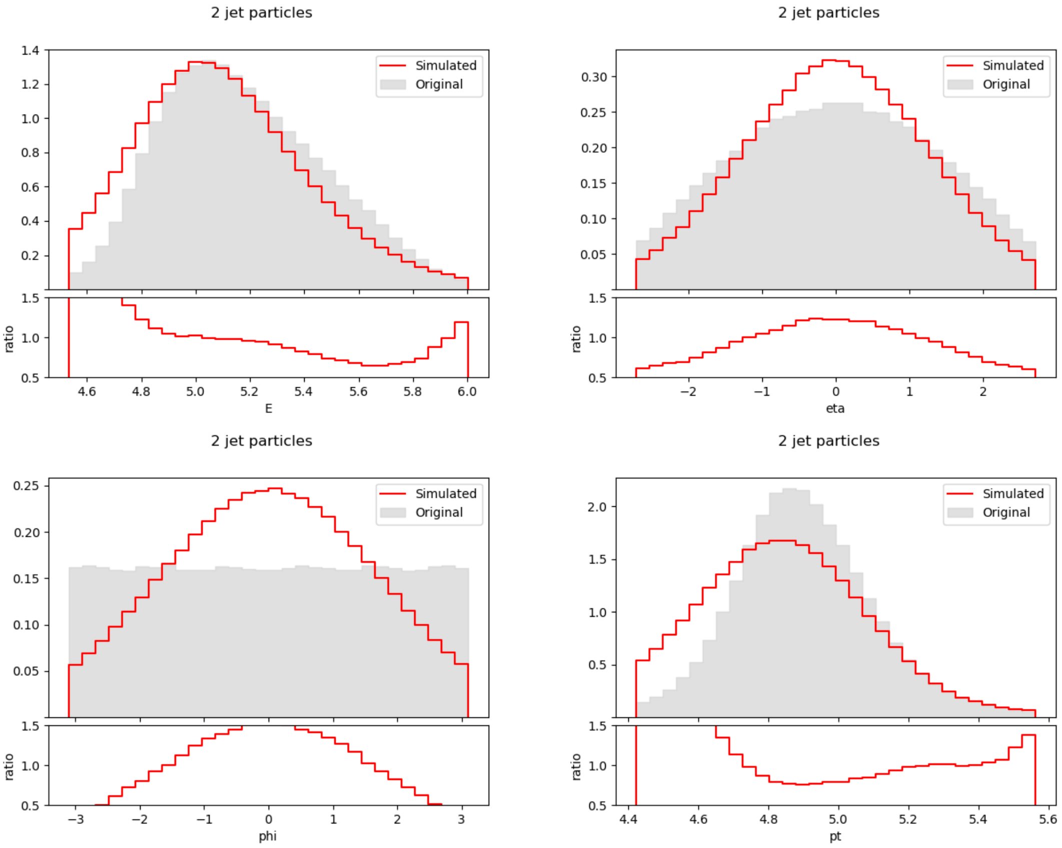


Figure E.101. Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 10$).

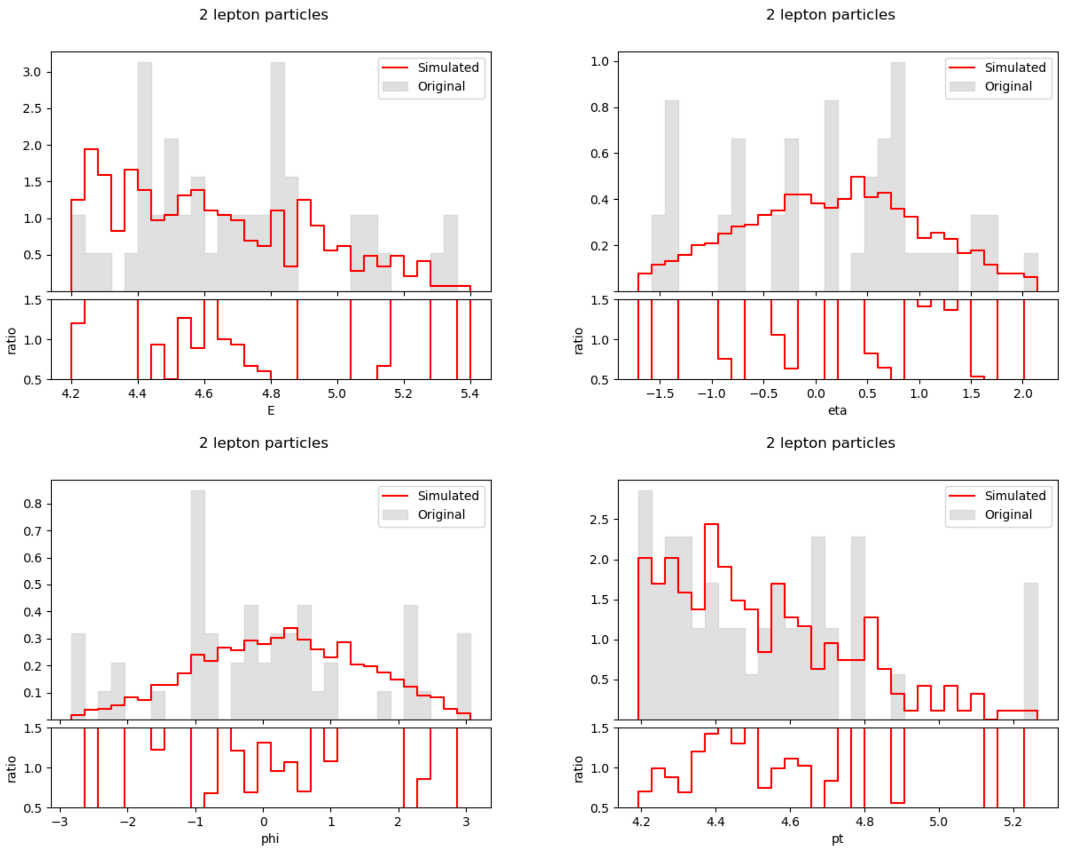


Figure E.102. Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 10$).

$\alpha = 0.1, \gamma = 20$

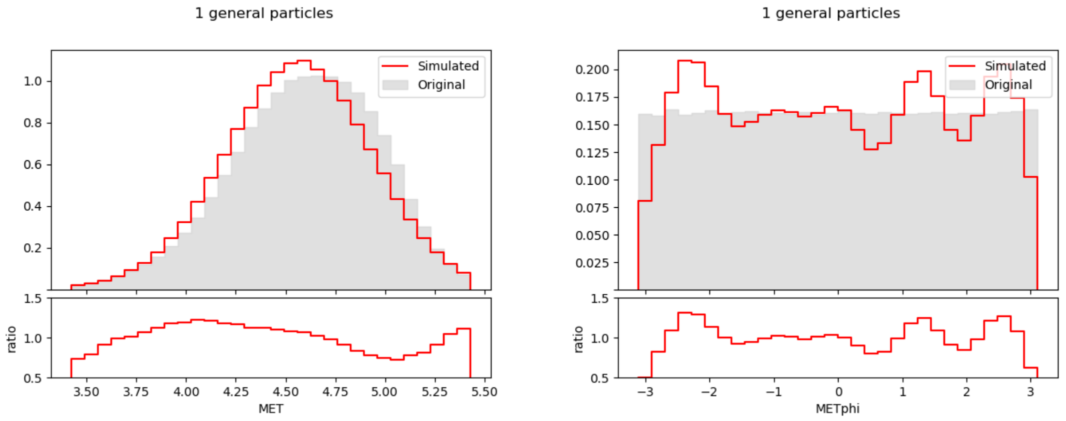


Figure E.103. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.1$, $\gamma = 20$).

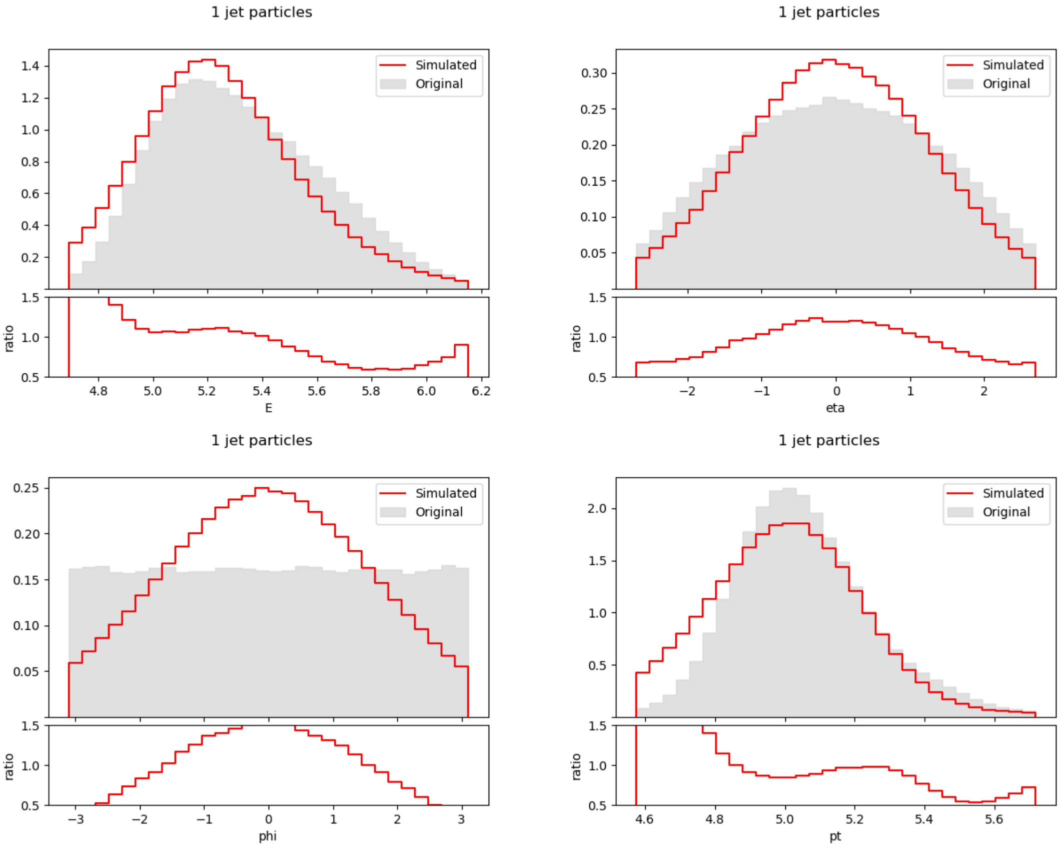


Figure E.104. First jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 20$).

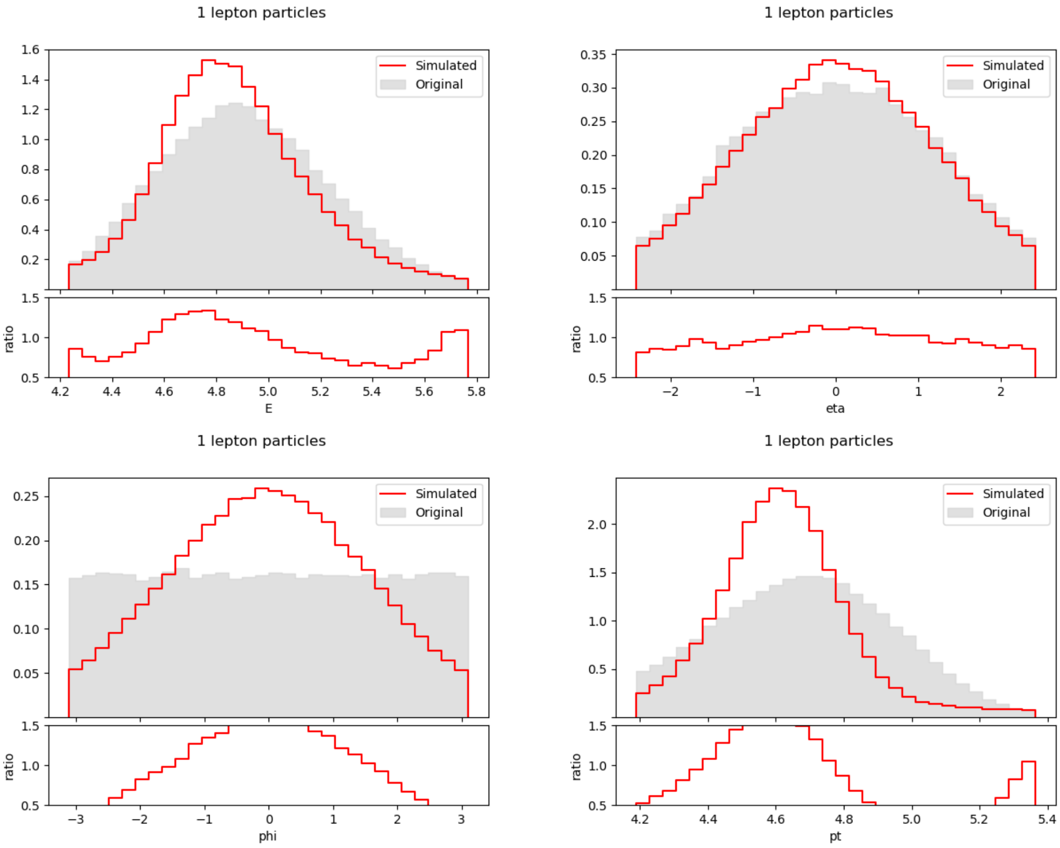


Figure E.105. First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 20$).

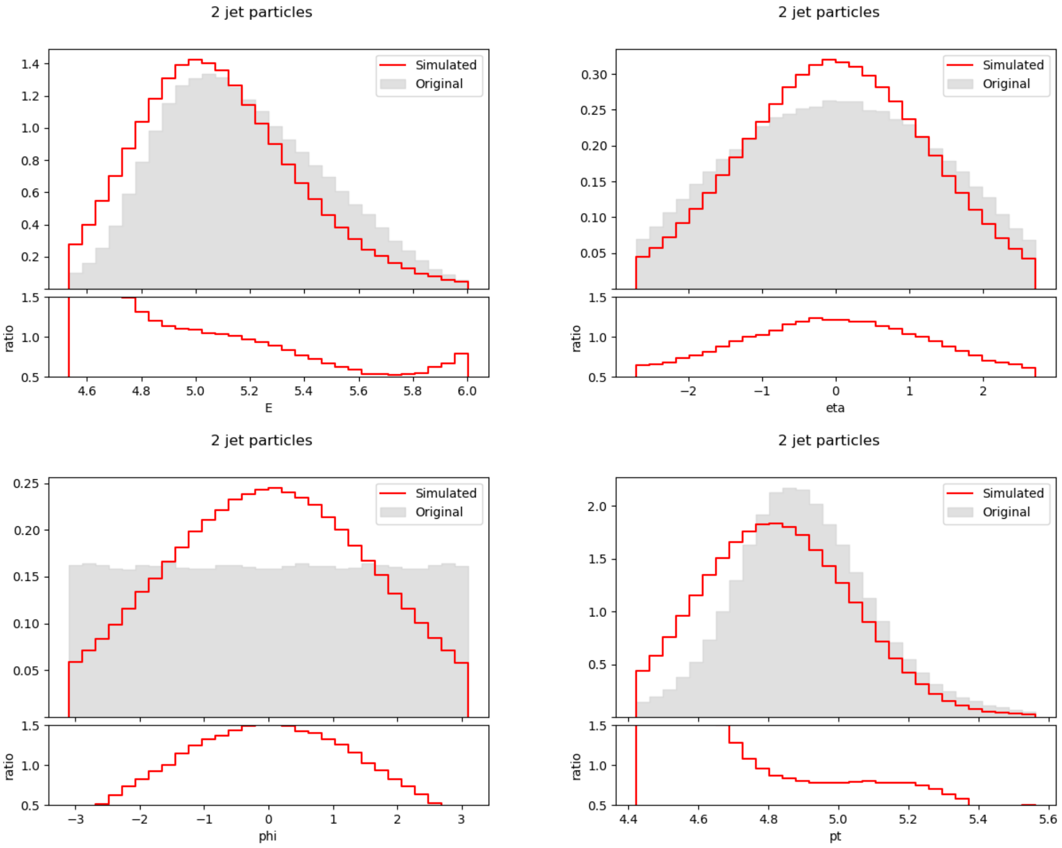


Figure E.106. Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 20$).

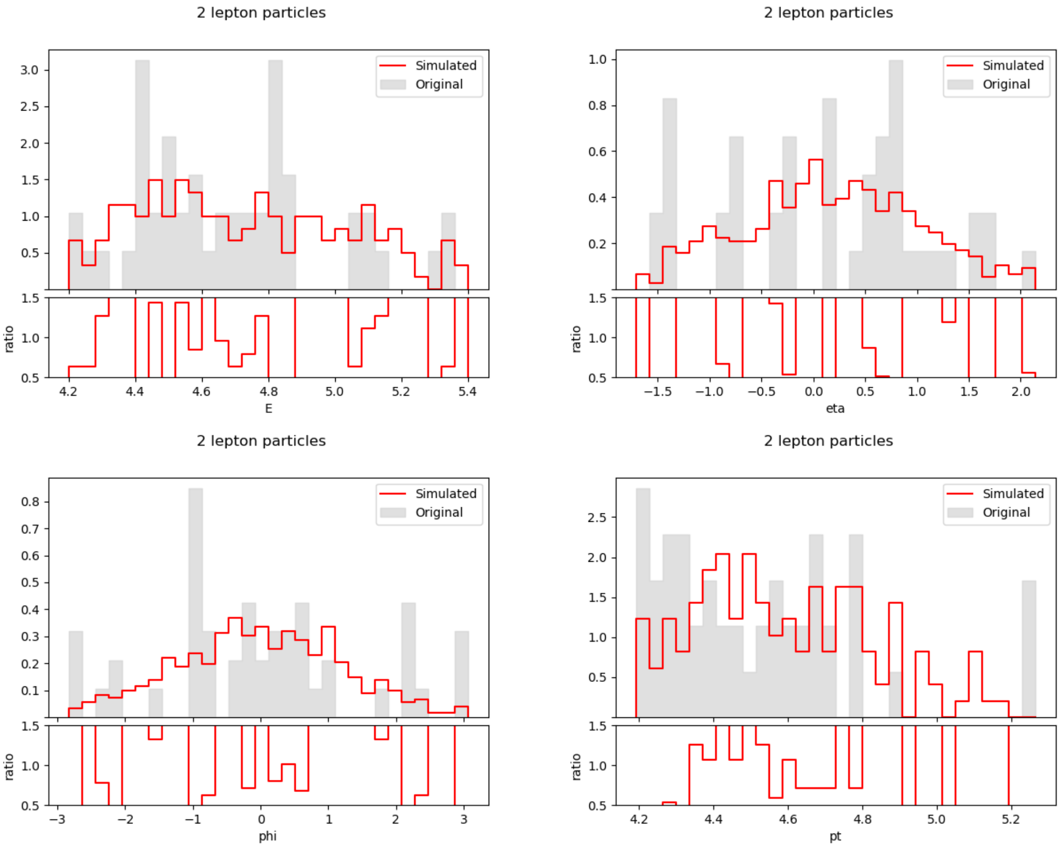


Figure E.107. Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 20$).

$\alpha = 0.1, \gamma = 50$

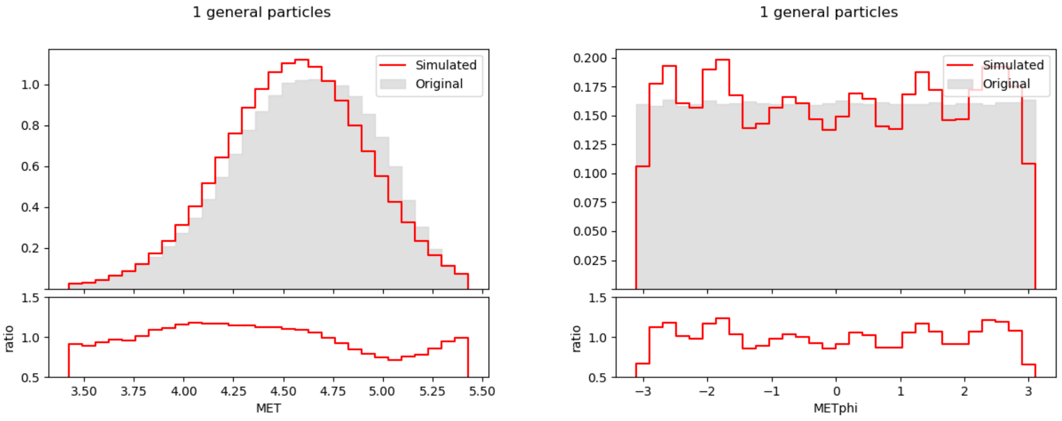


Figure E.108. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$).

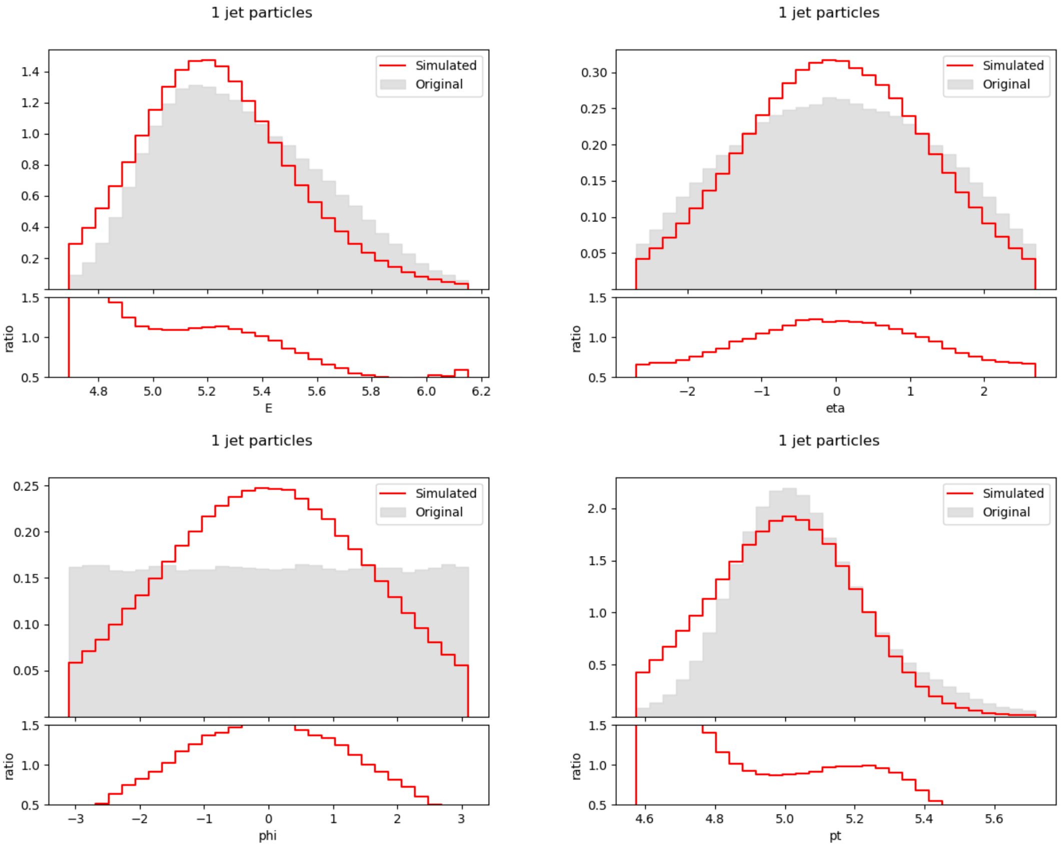


Figure E.109. First jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$).

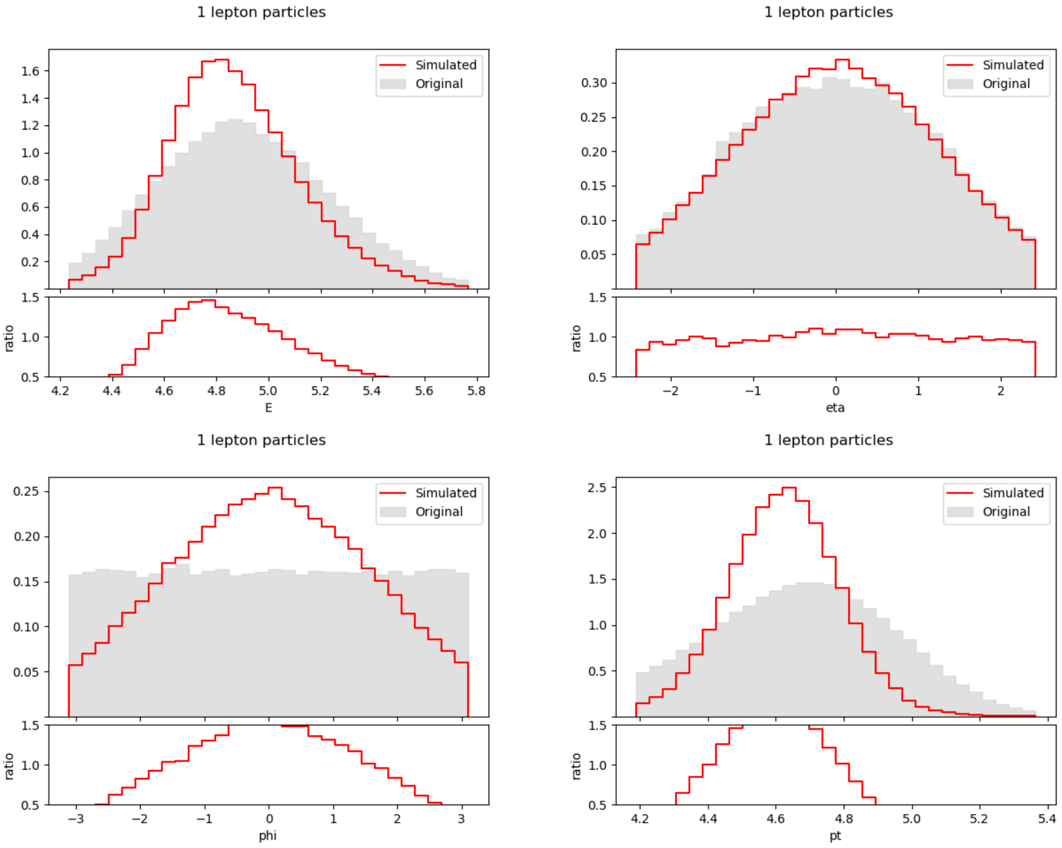


Figure E.110. First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$).

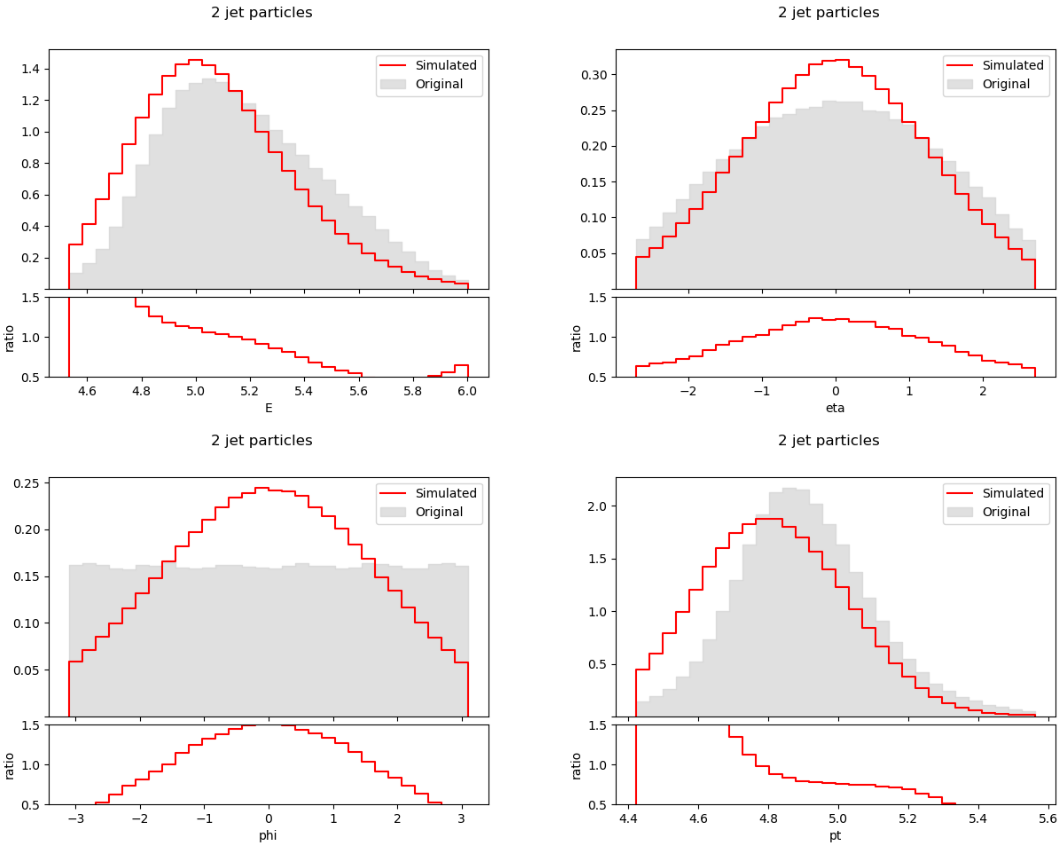


Figure E.111. Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 50$).

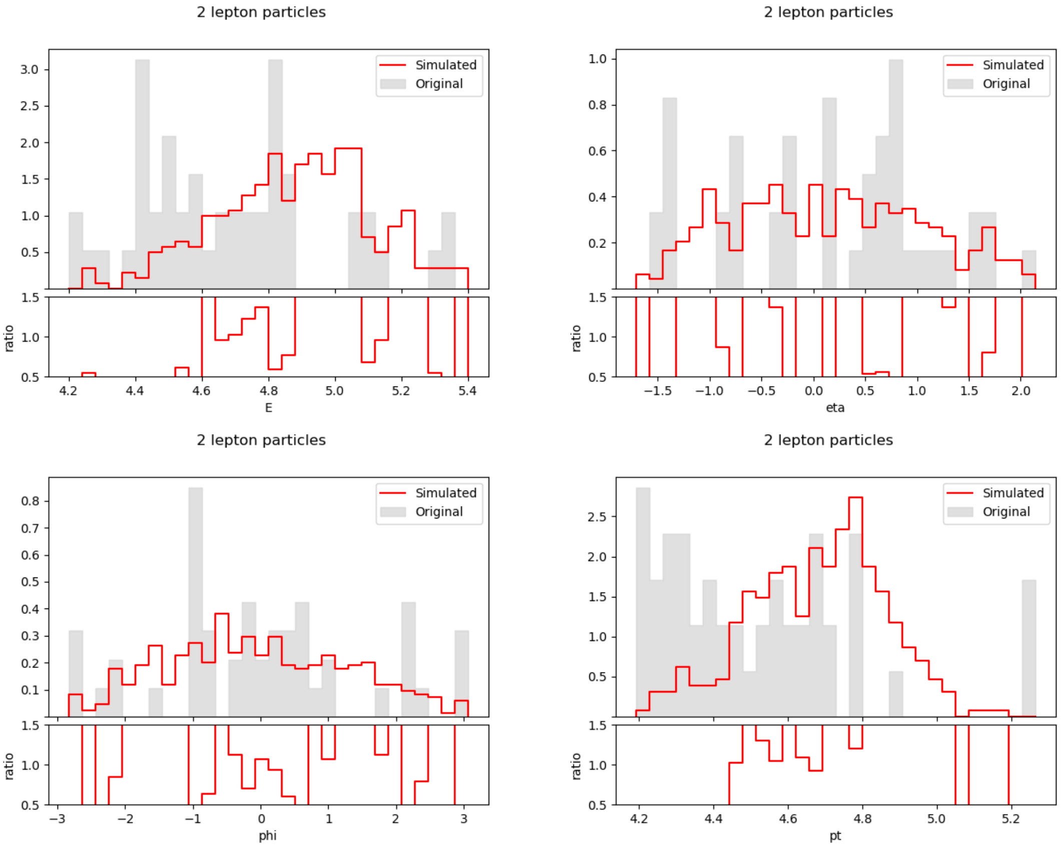


Figure E.112. Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 50$).

$\alpha = 0.1, \gamma = 100$

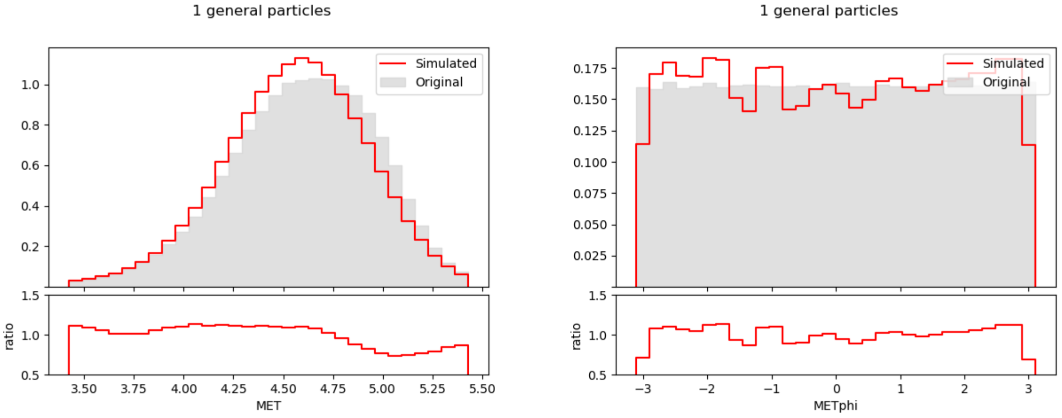


Figure E.113. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$).

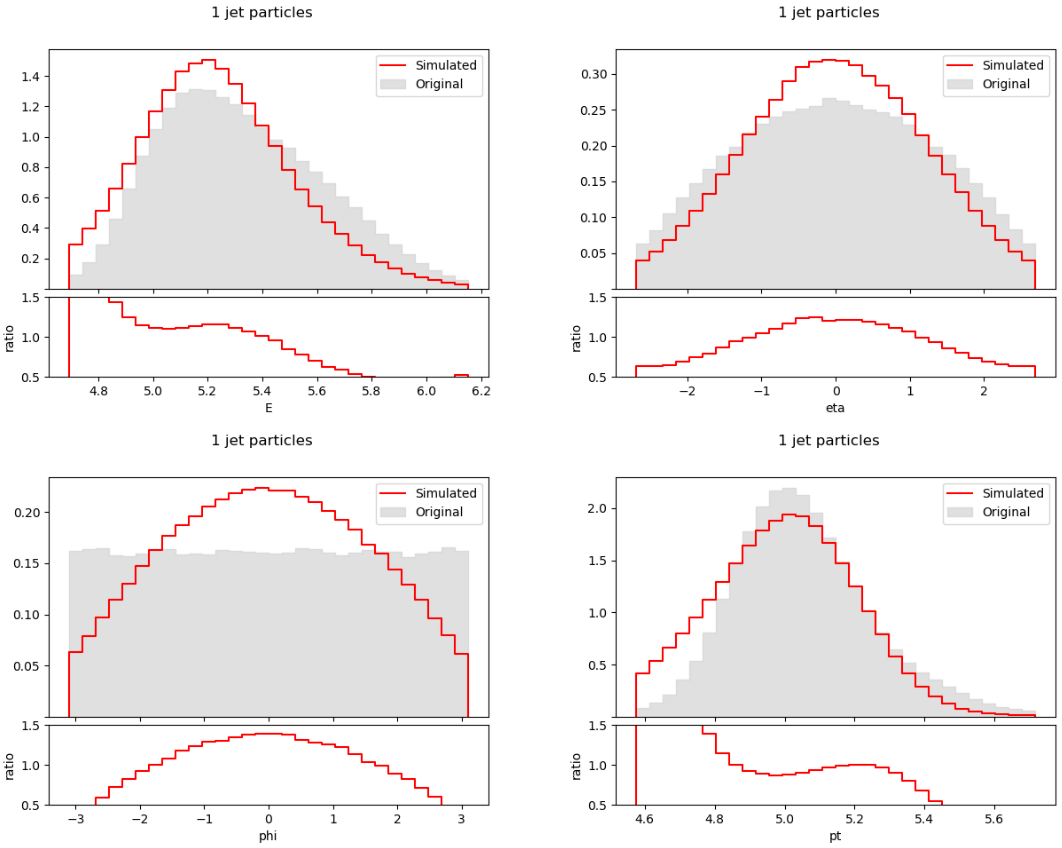


Figure E.114. First jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$).

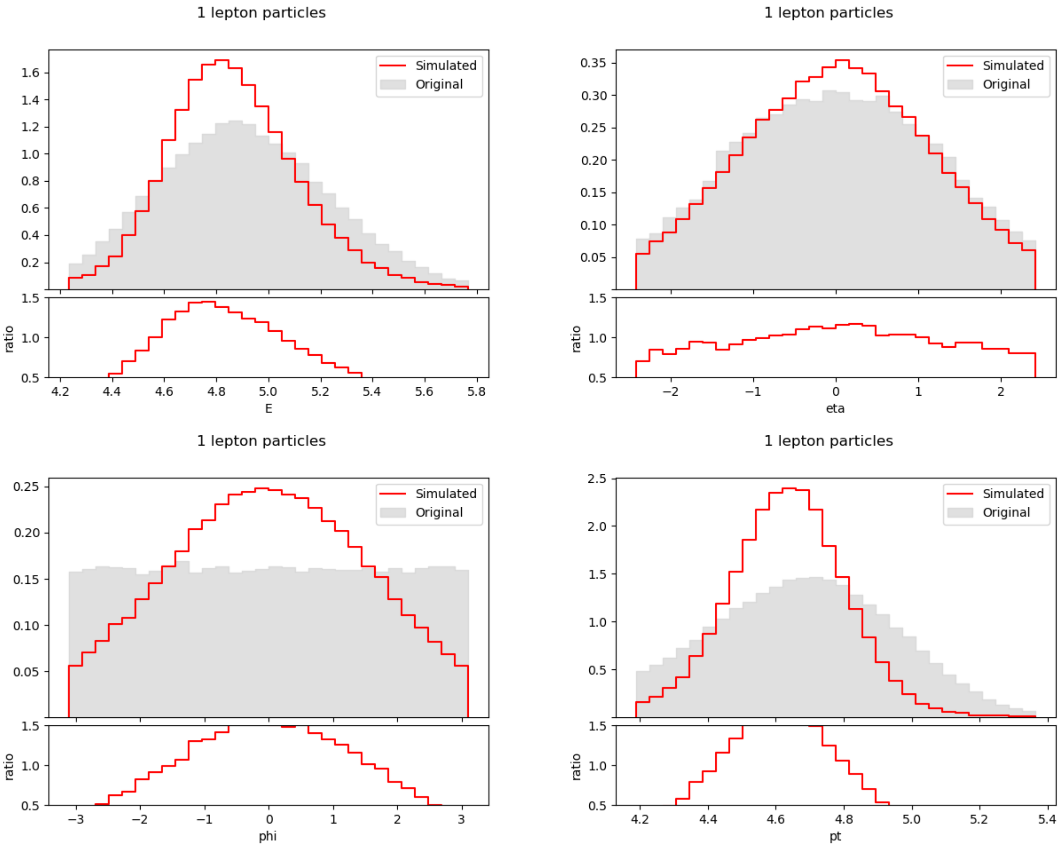


Figure E.115. First lepton particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$).

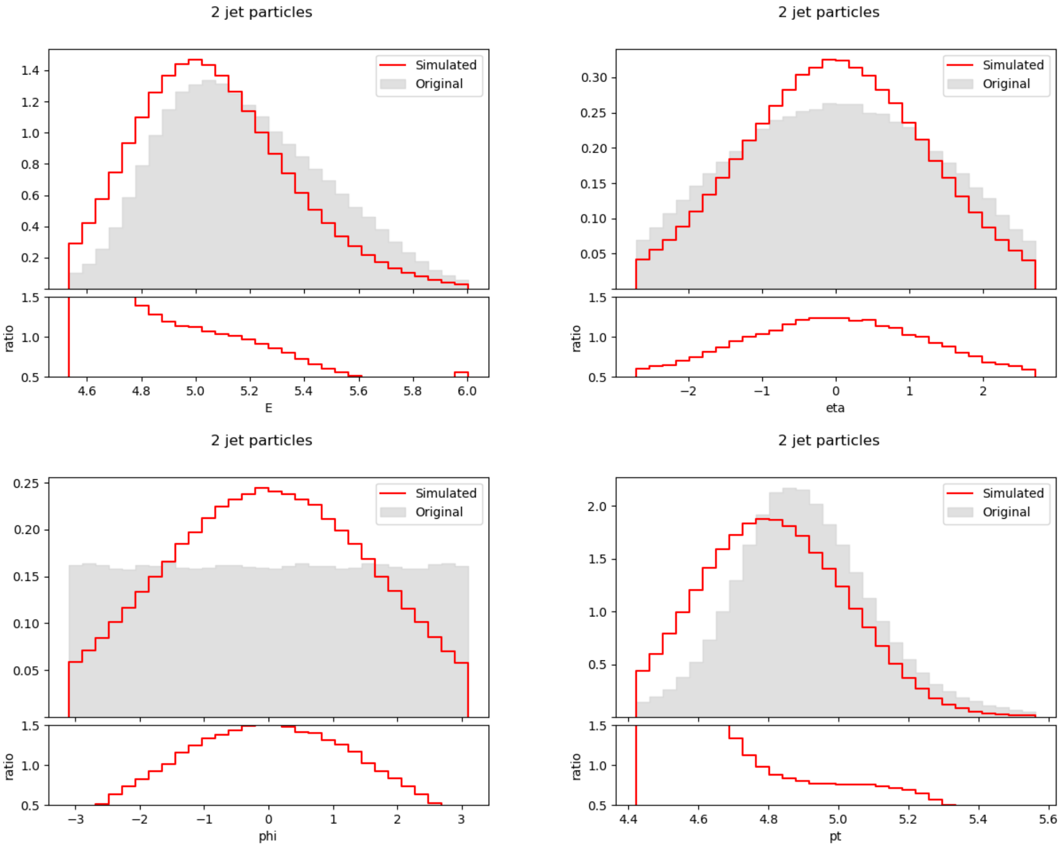


Figure E.116. Second jet particle feature distributions using a BGMM ($\alpha = 0.1$, $\gamma = 100$).

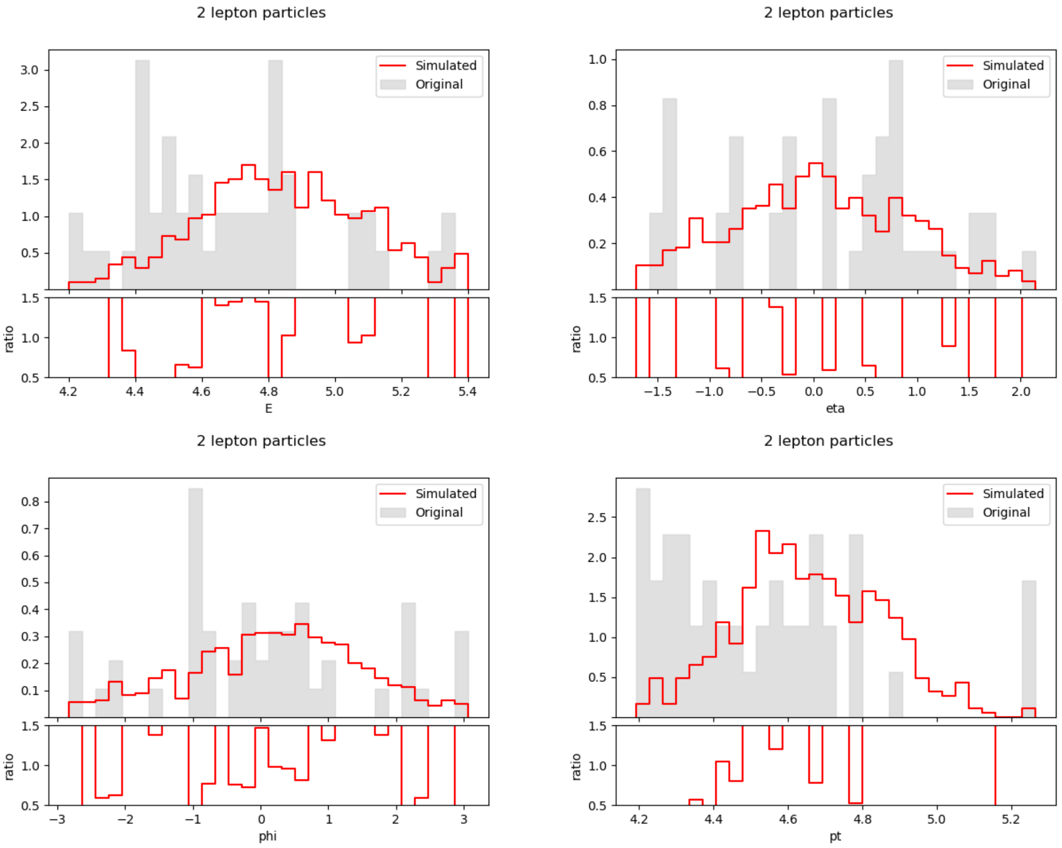


Figure E.117. Second lepton particle feature distributions using BGMM ($\alpha = 0.1$, $\gamma = 100$).

$\alpha = 0.2, \gamma = 10$

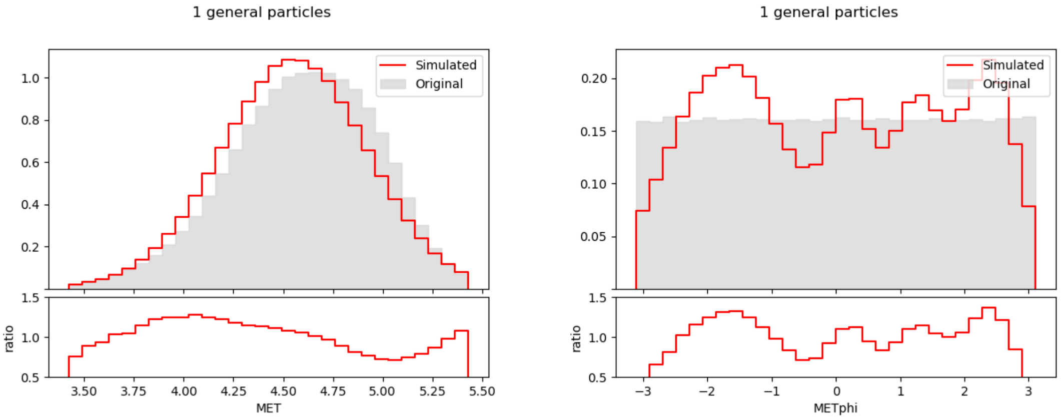


Figure E.118. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$).

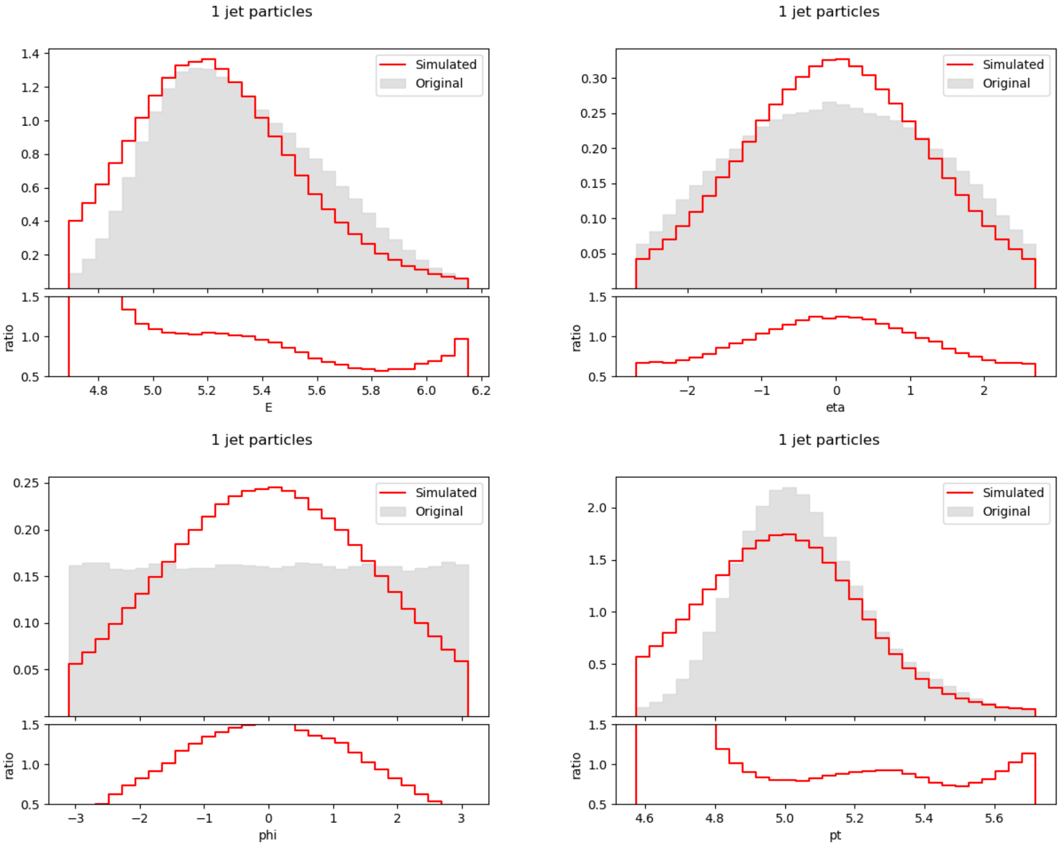


Figure E.119. First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$).

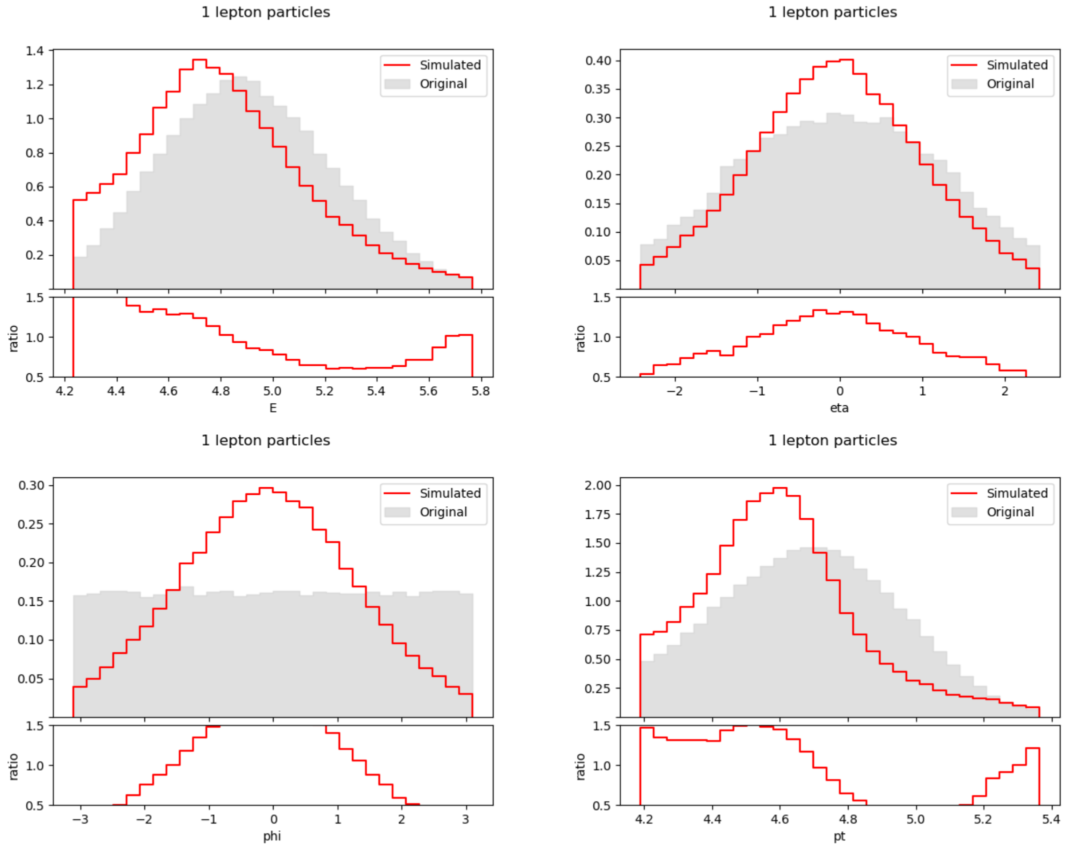


Figure E.120. First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$).

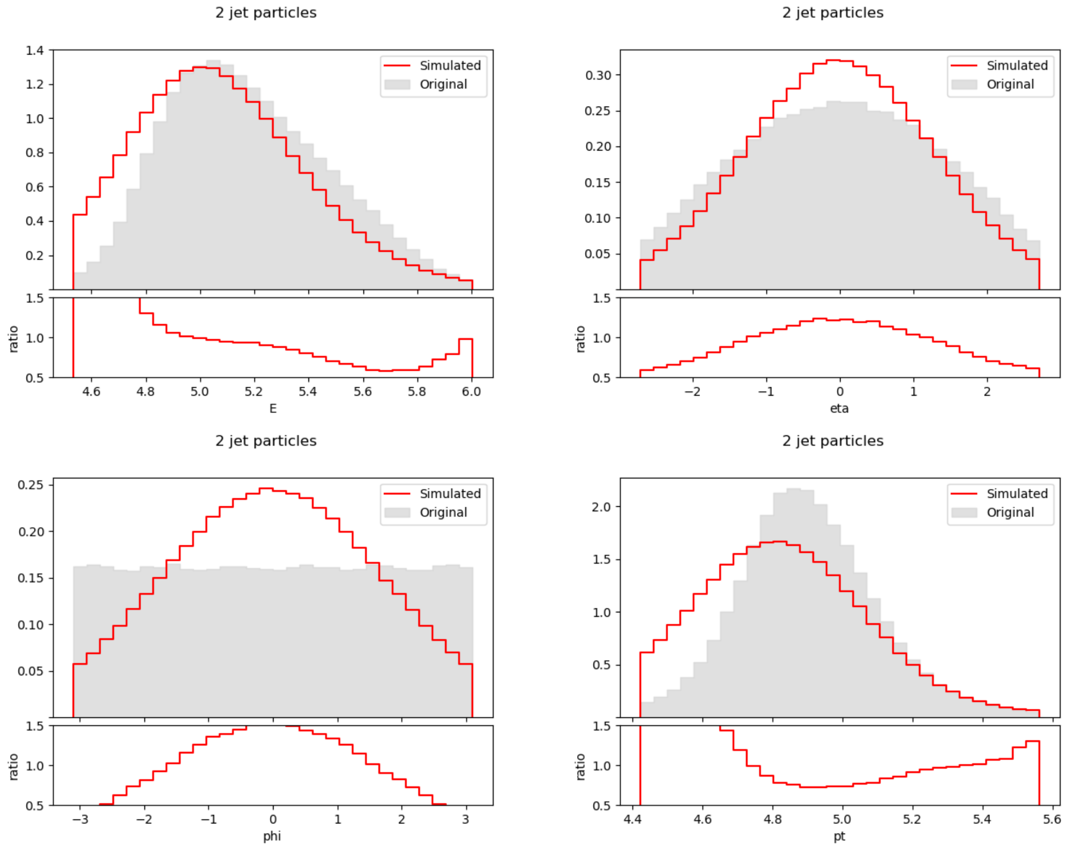


Figure E.121. Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 10$).

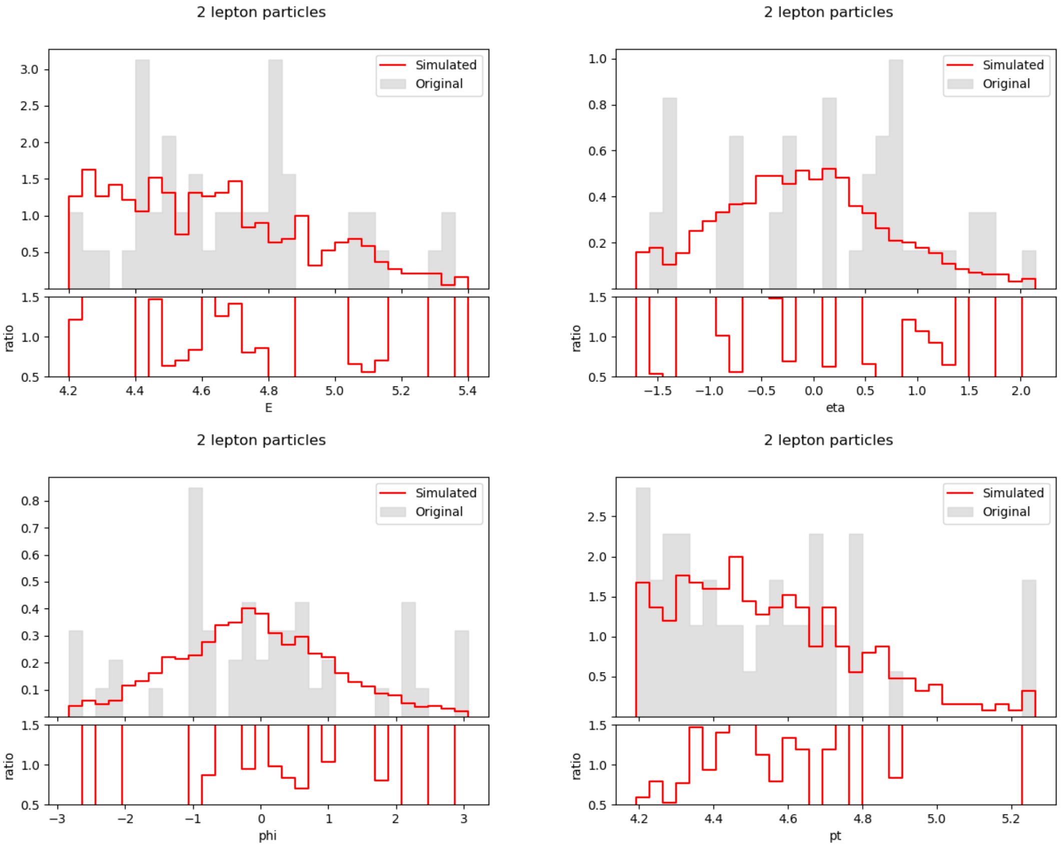


Figure E.122. Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 10$).

$\alpha = 0.2, \gamma = 20$

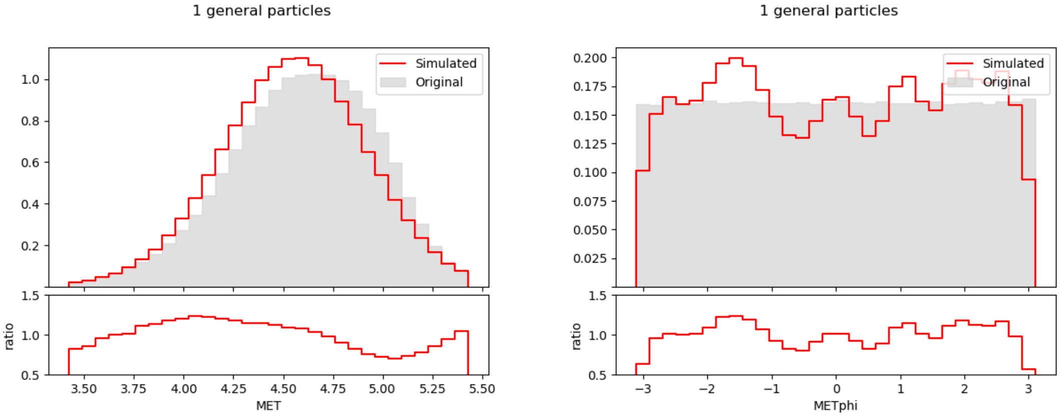


Figure E.123. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$).

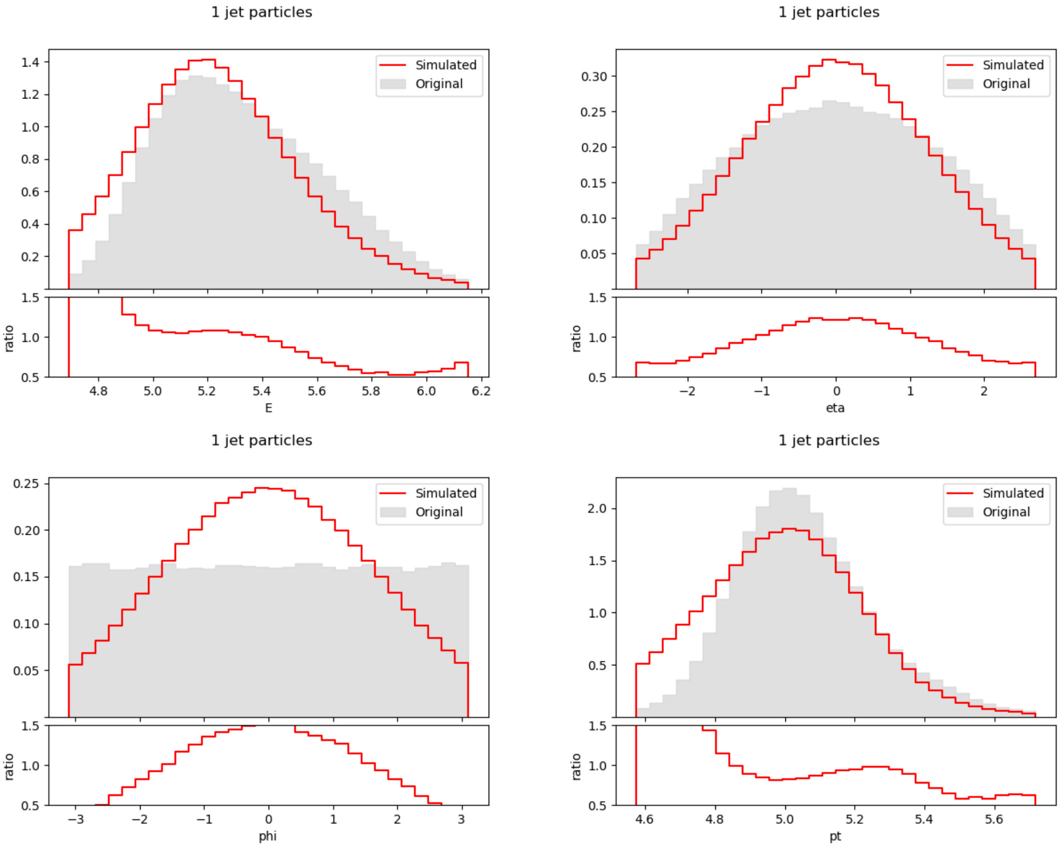


Figure E.124. First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$).

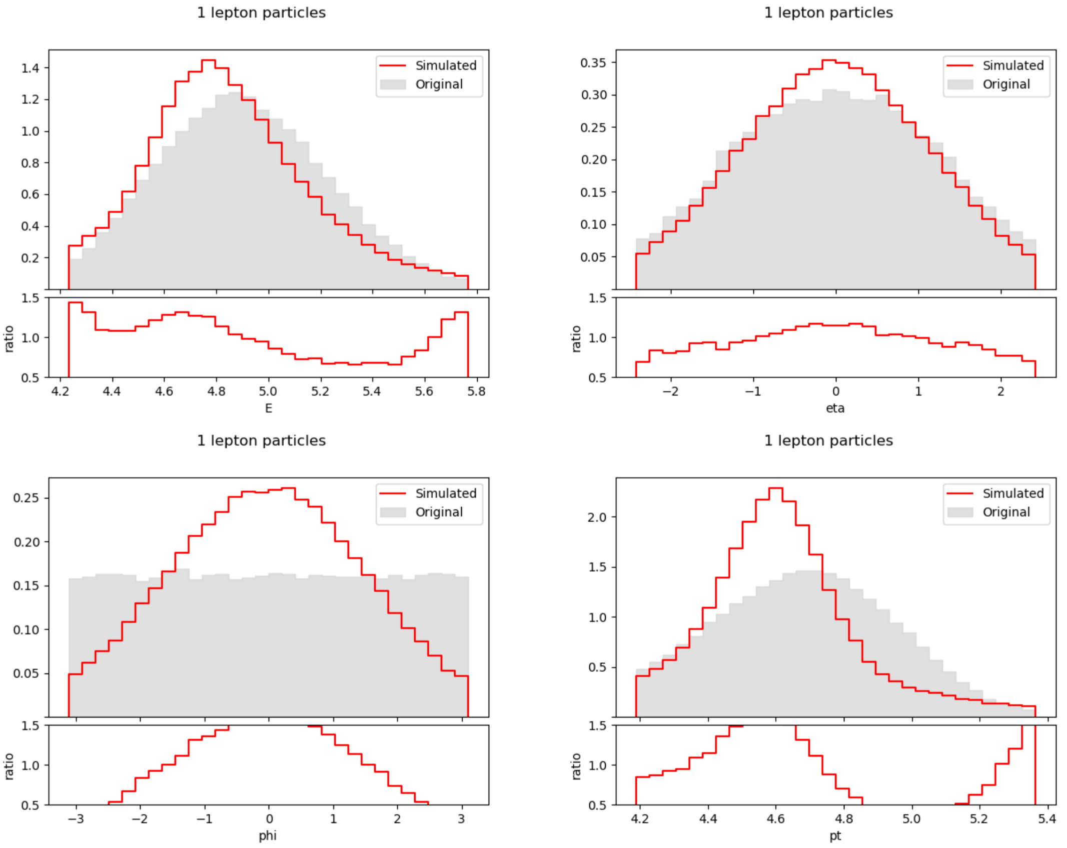


Figure E.125. First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$).

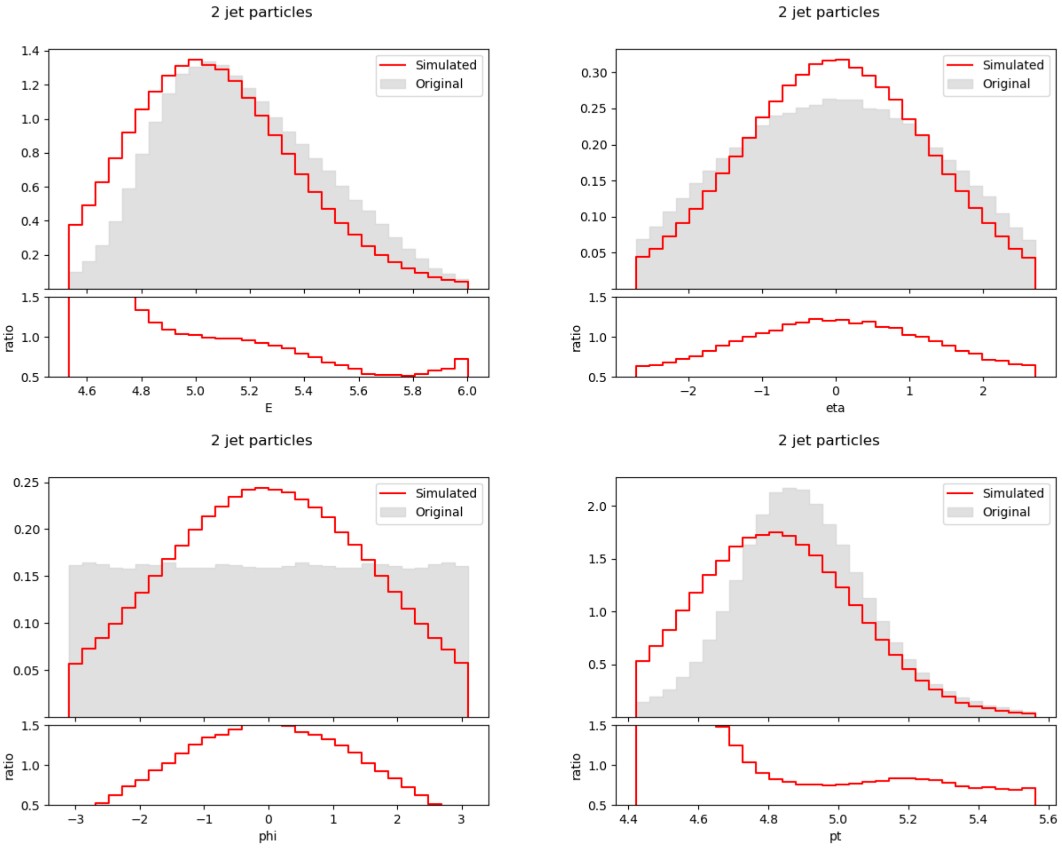


Figure E.126. Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 20$).

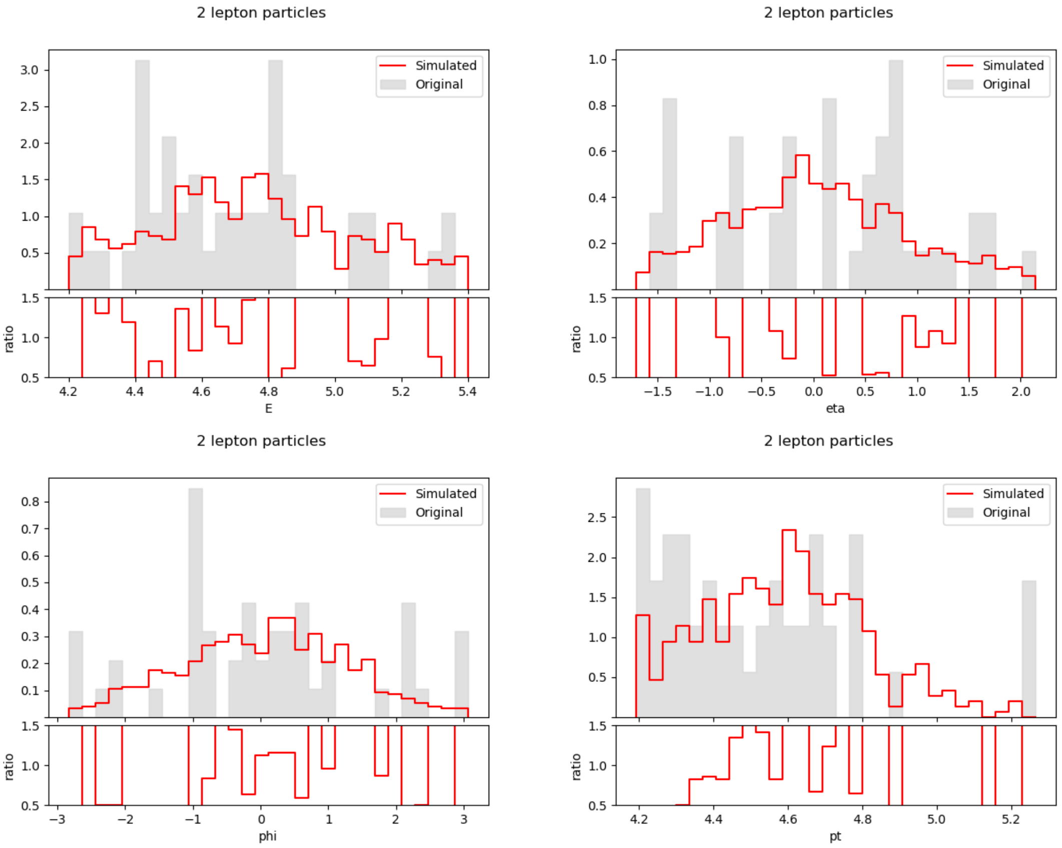


Figure E.127. Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 20$).

$\alpha = 0.2, \gamma = 50$

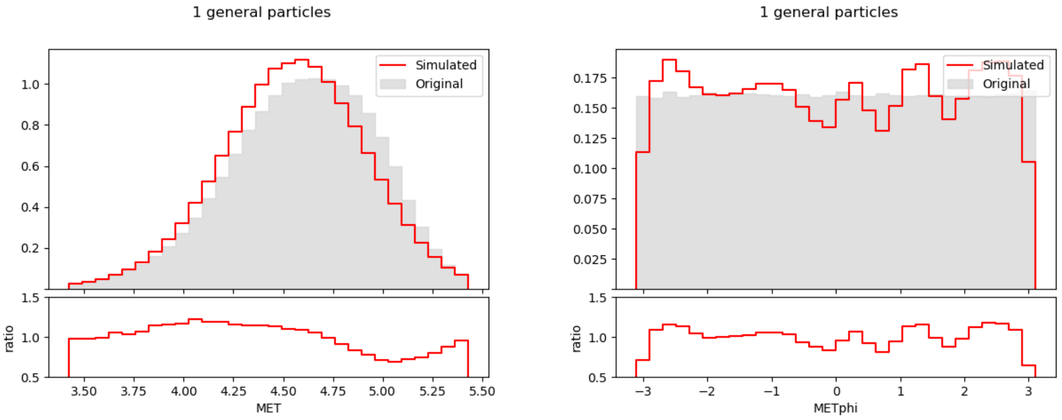


Figure E.128. Event MET and METphi distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$).

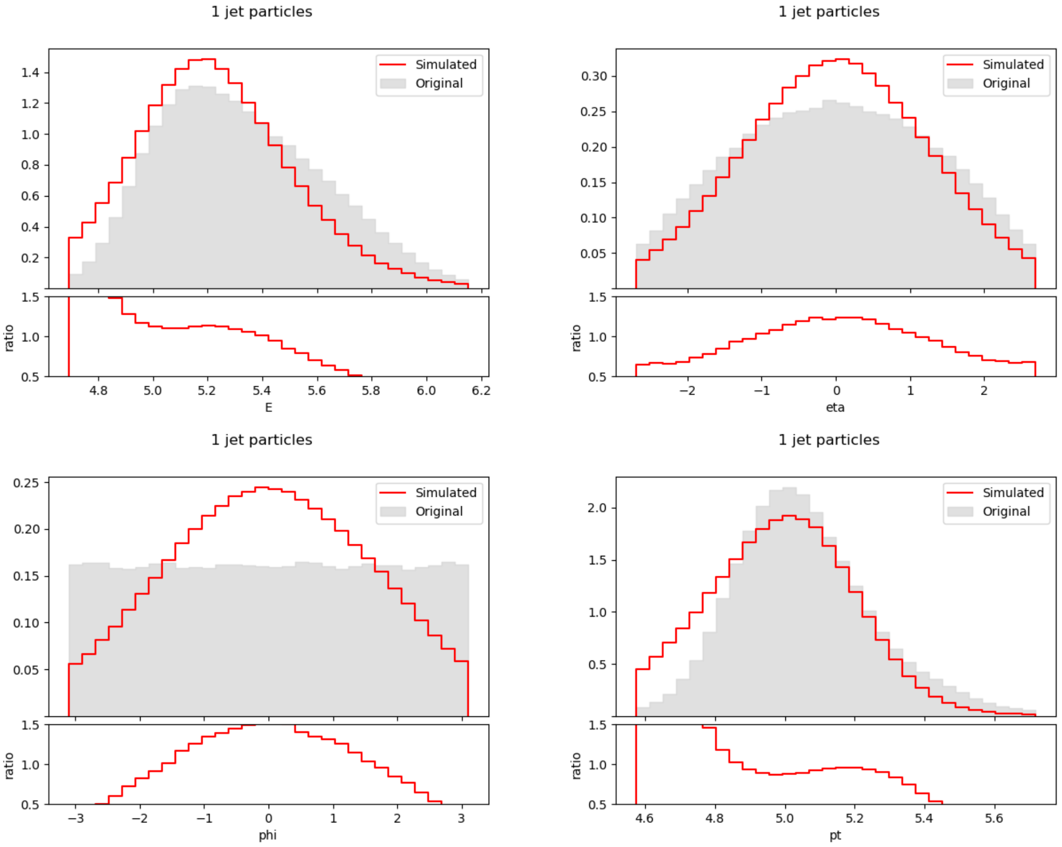


Figure E.129. First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$).

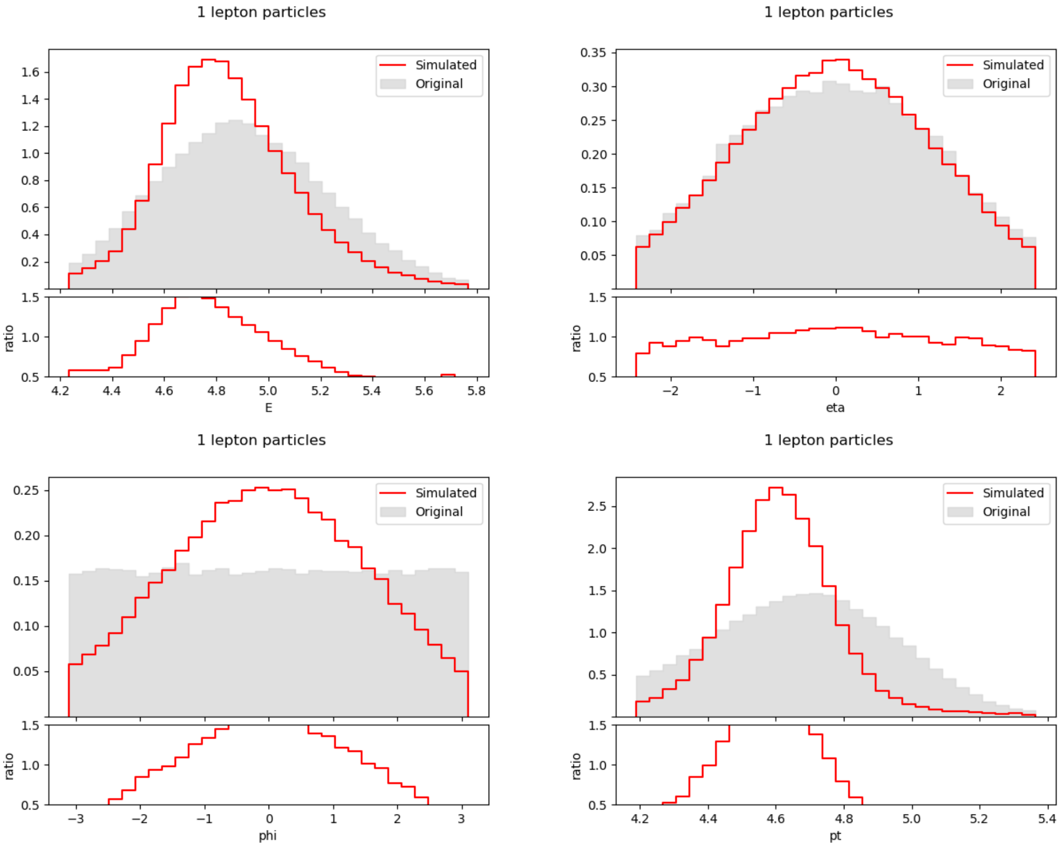


Figure E.130. First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$).

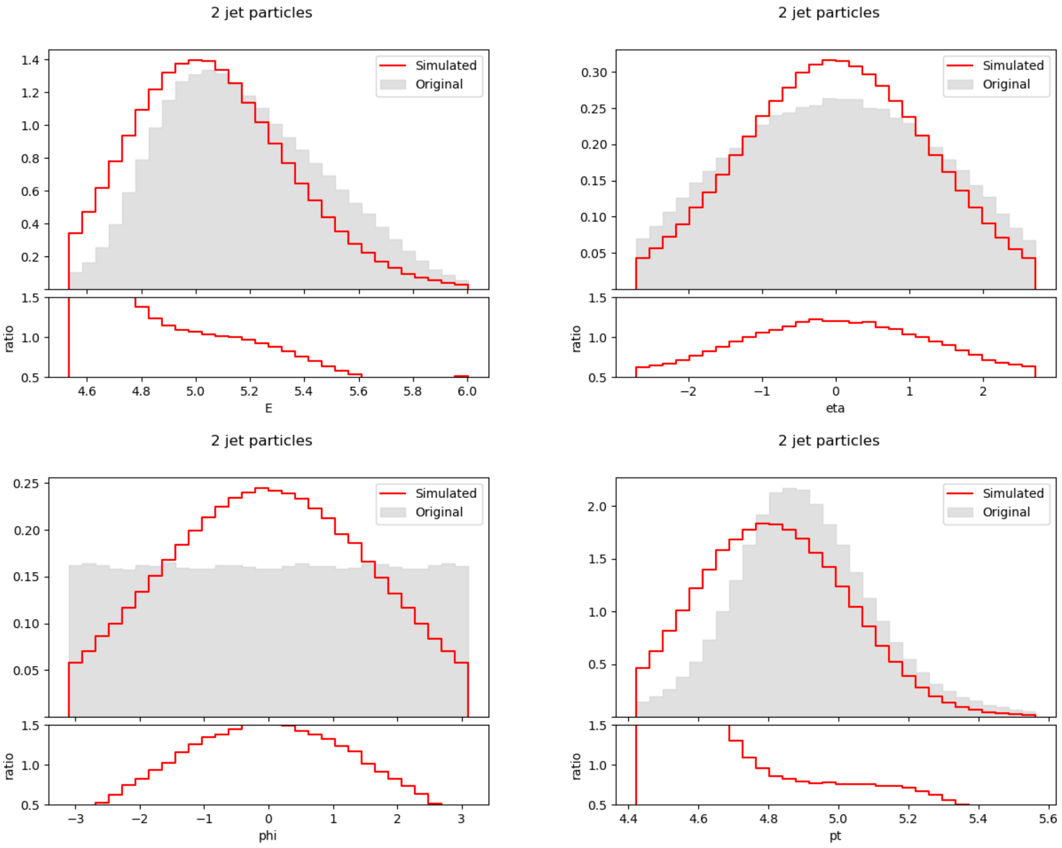


Figure E.131. Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 50$).

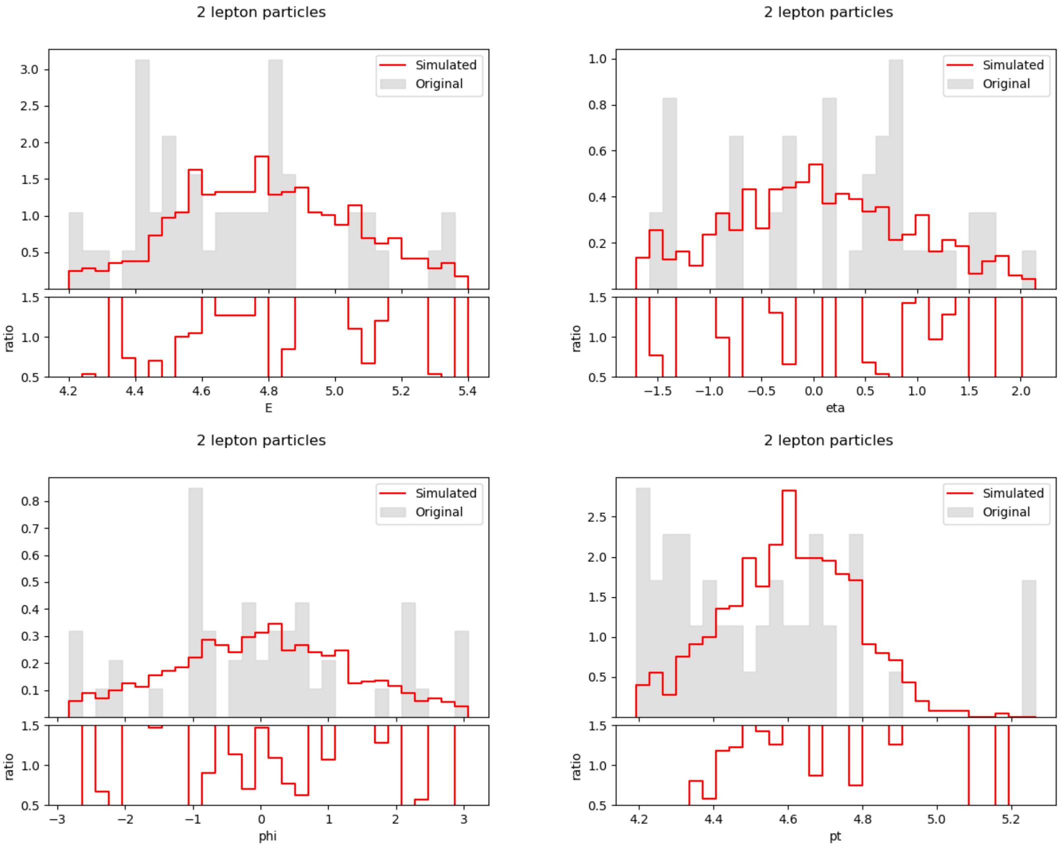


Figure E.132. Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 50$).

$\alpha = 0.2, \gamma = 100$

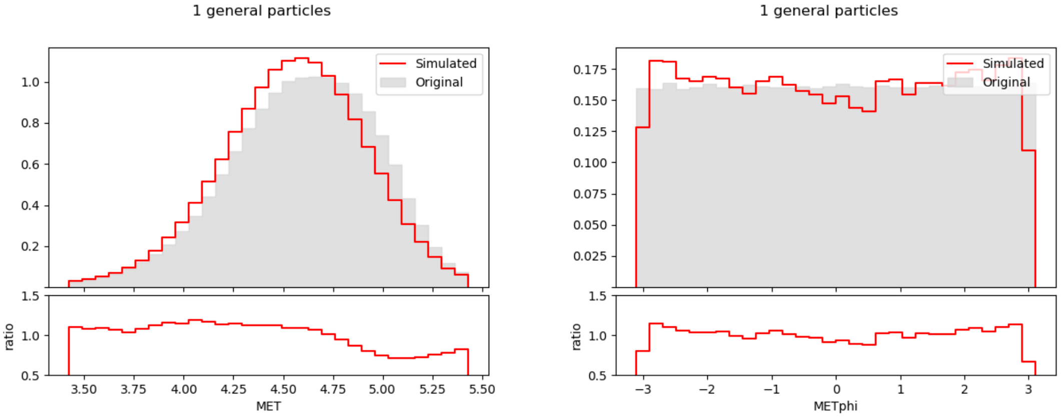


Figure E.133. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$).

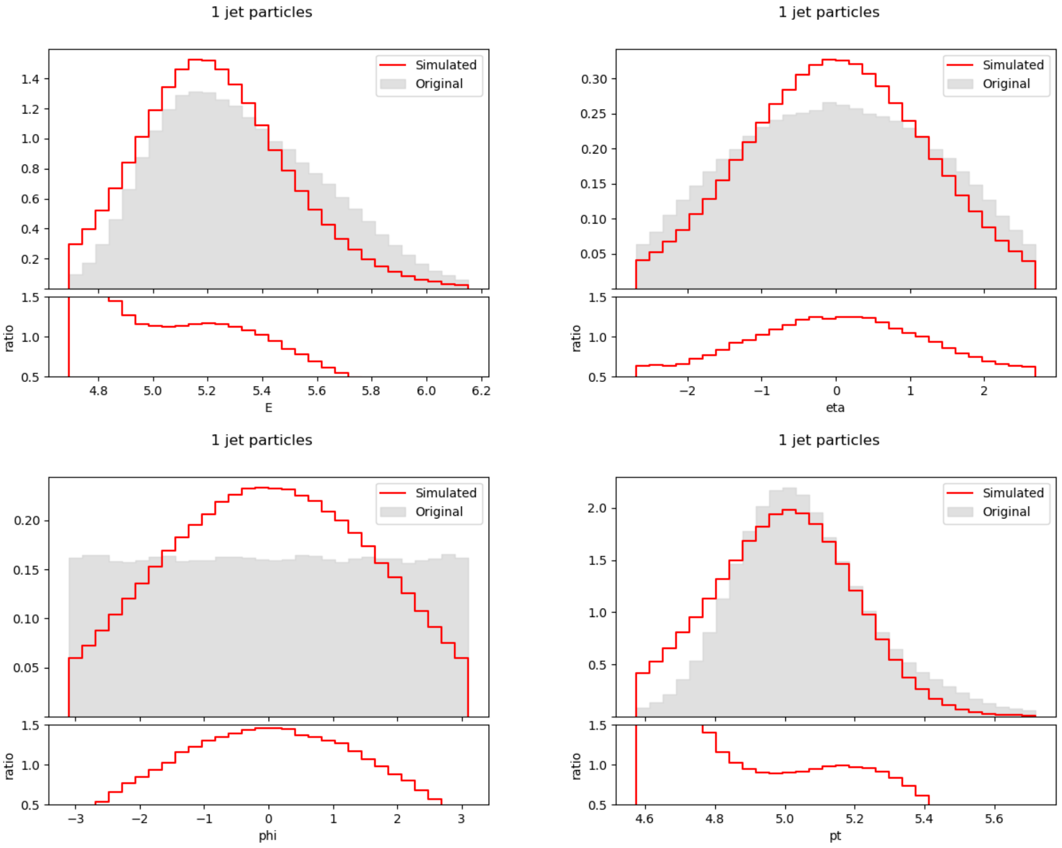


Figure E.134. First jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$).

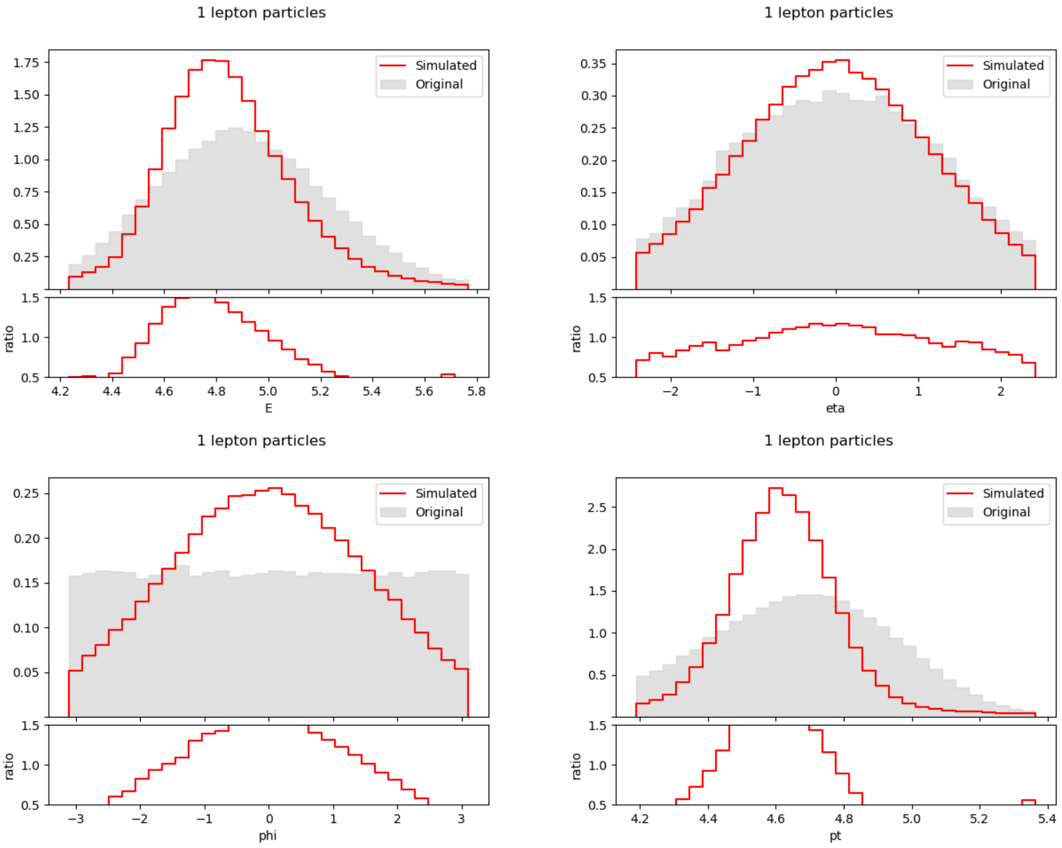


Figure E.135. First lepton particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$).

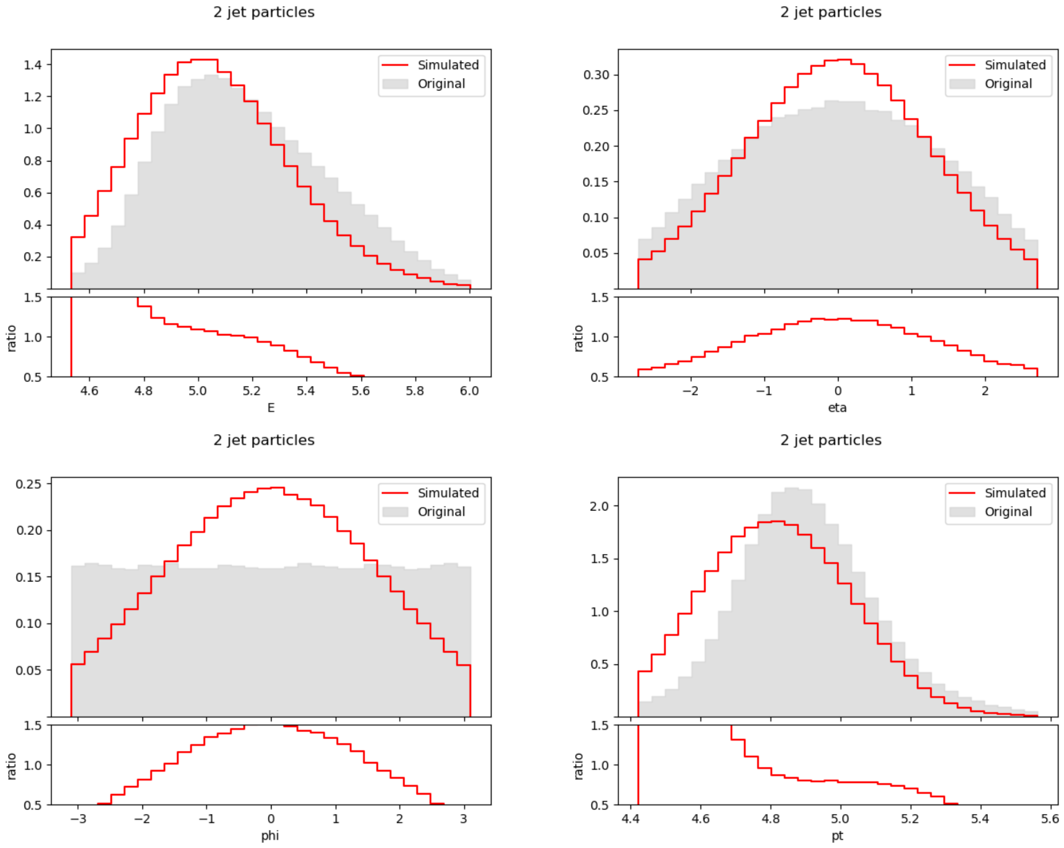


Figure E.136. Second jet particle feature distributions using a BGMM ($\alpha = 0.2$, $\gamma = 100$).

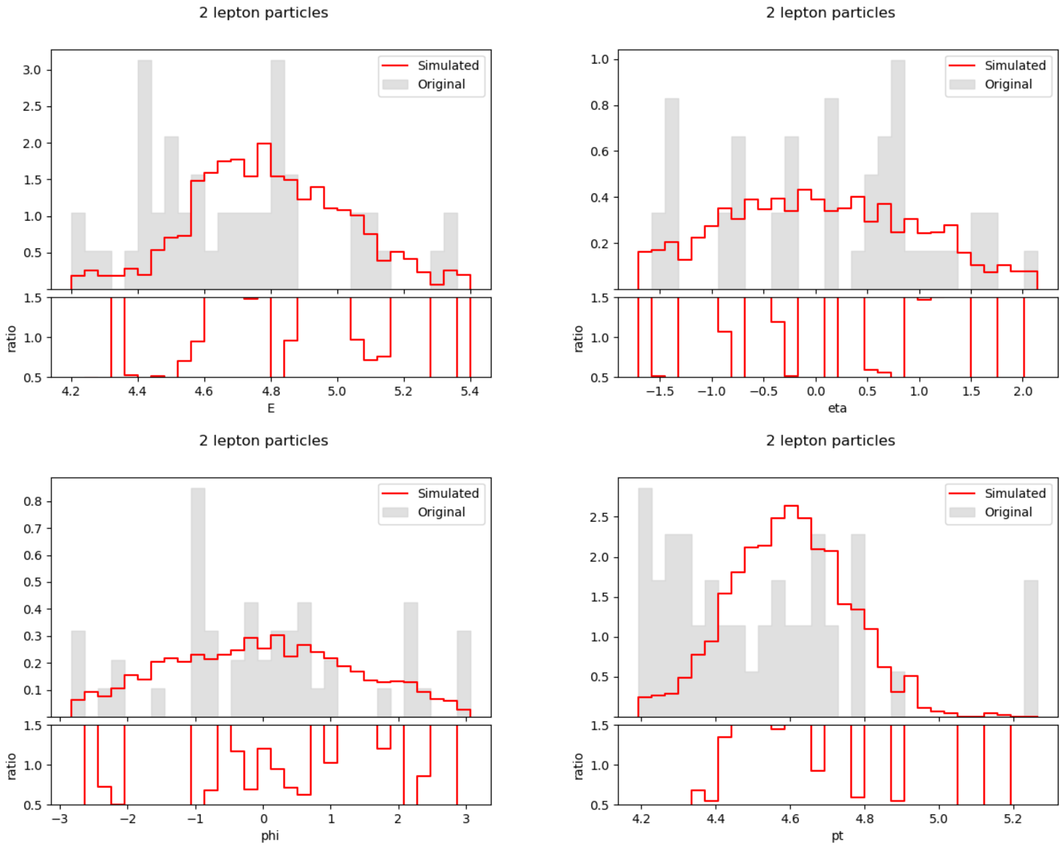


Figure E.137. Second lepton particle feature distributions using BGMM ($\alpha = 0.2$, $\gamma = 100$).

$\alpha = 0.3, \gamma = 10$

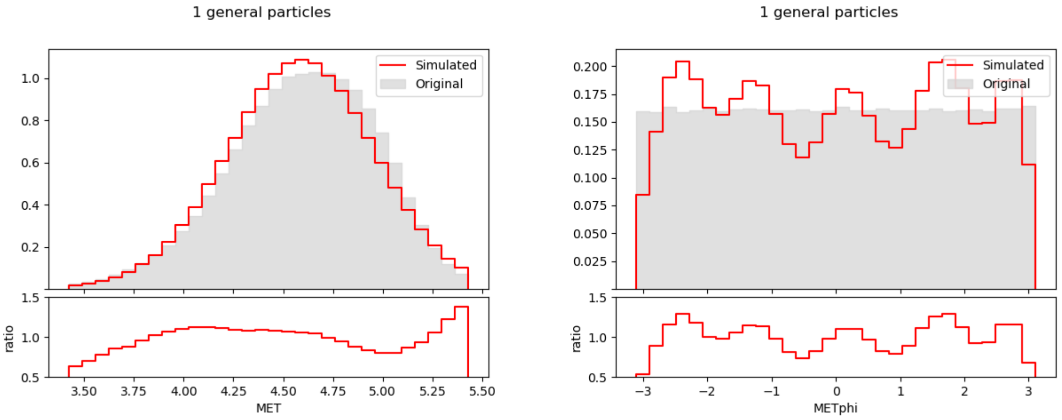


Figure E.138. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$).

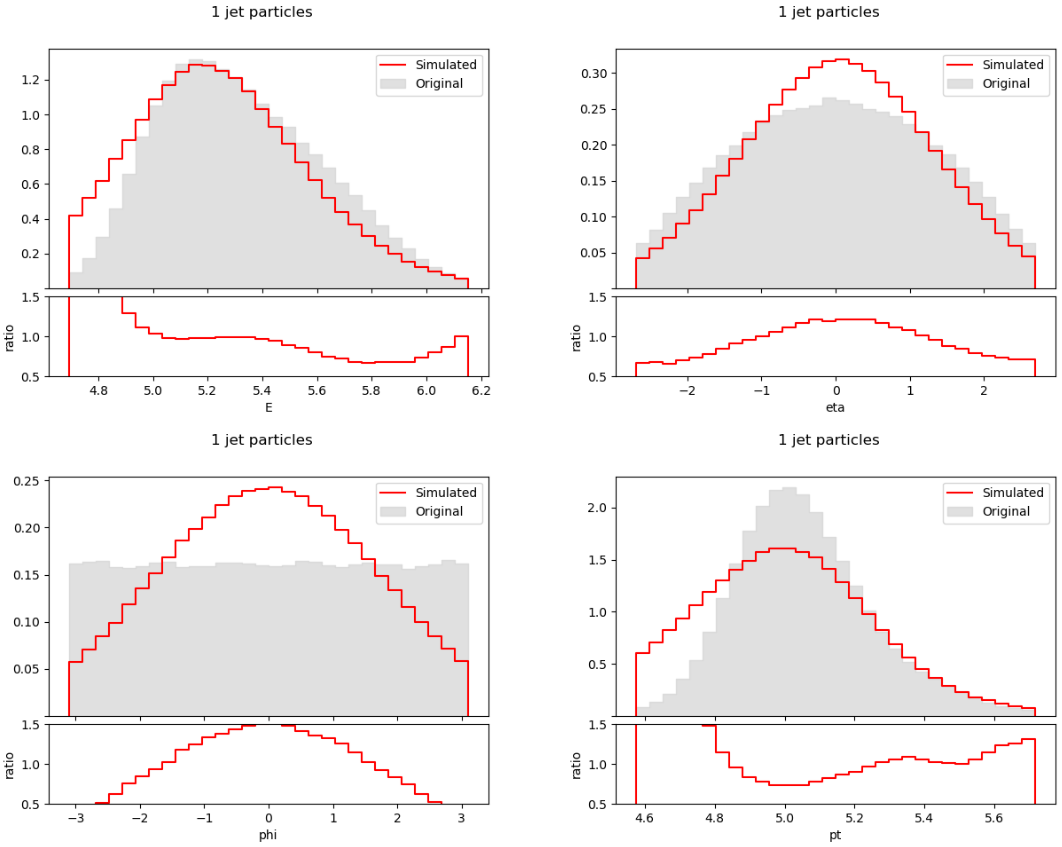


Figure E.139. First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$).

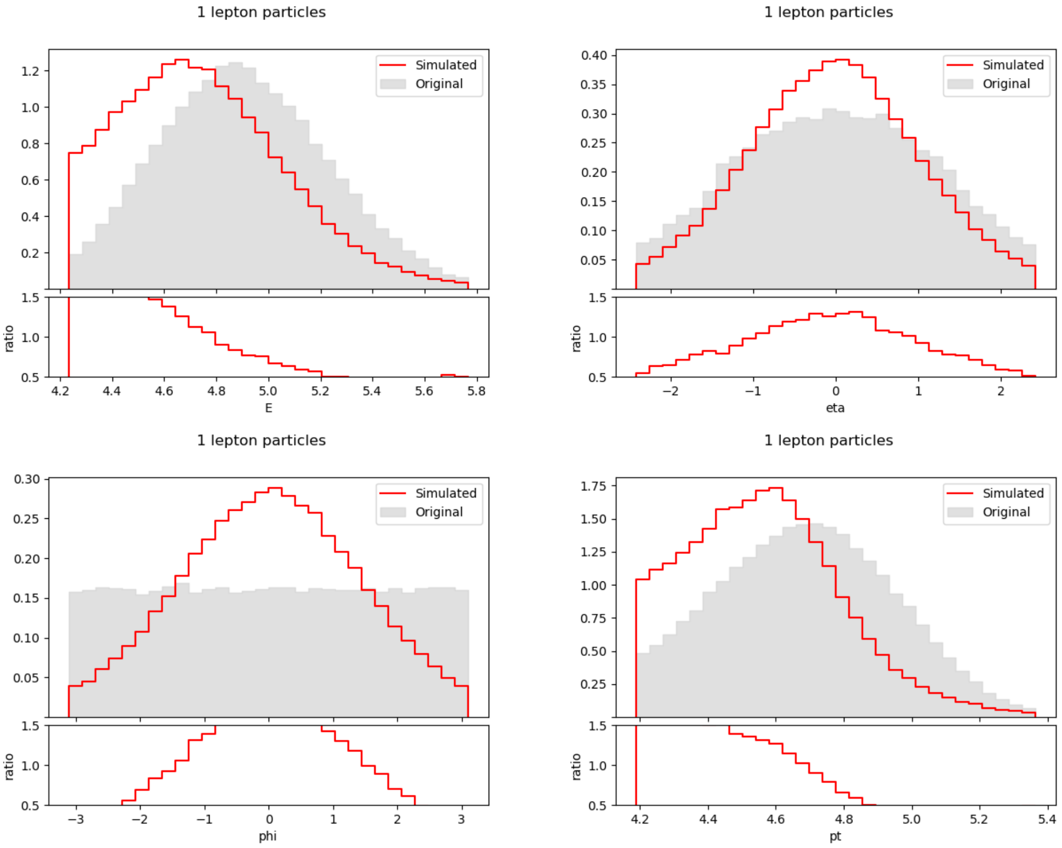


Figure E.140. First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$).

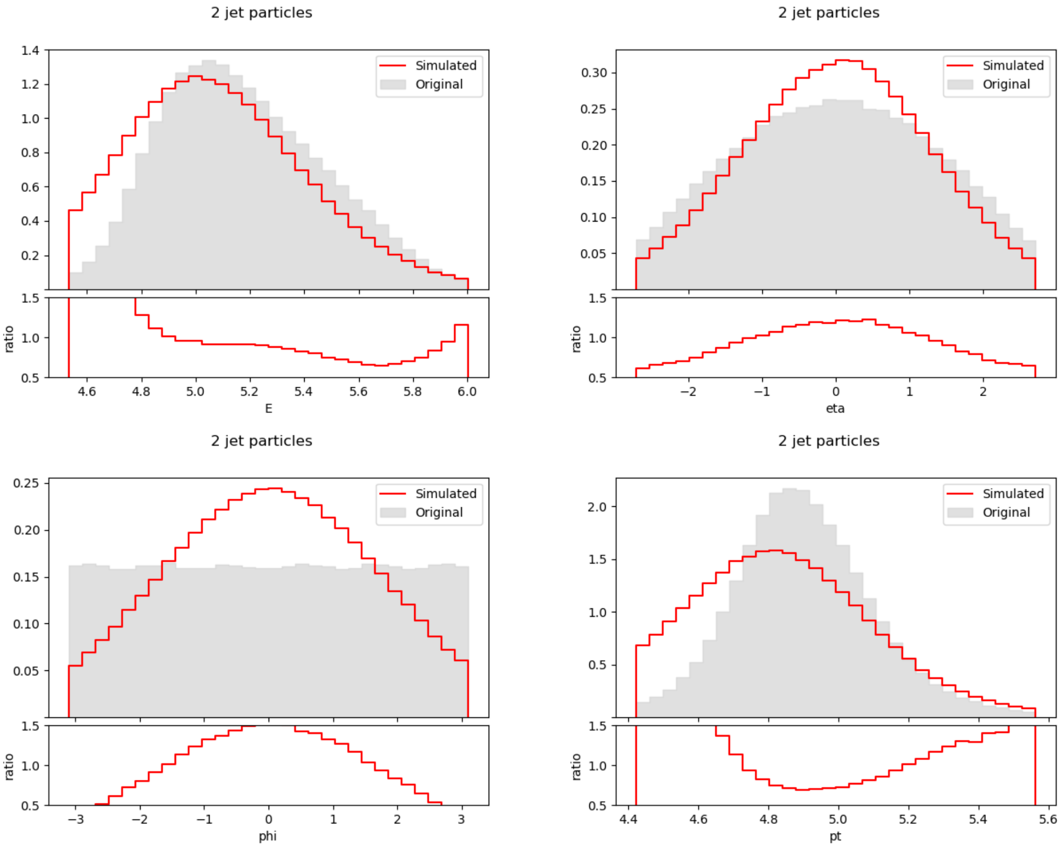


Figure E.141. Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 10$).

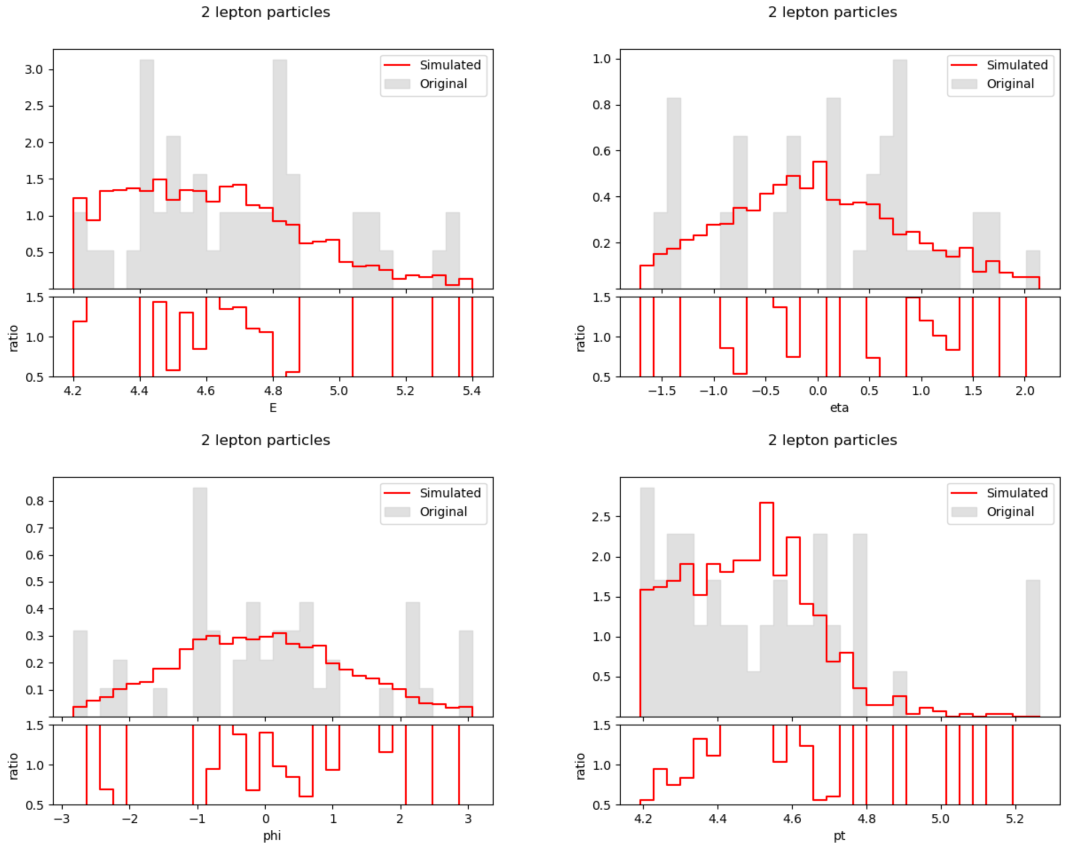


Figure E.142. Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 10$).

$\alpha = 0.3$, $\gamma = 20$

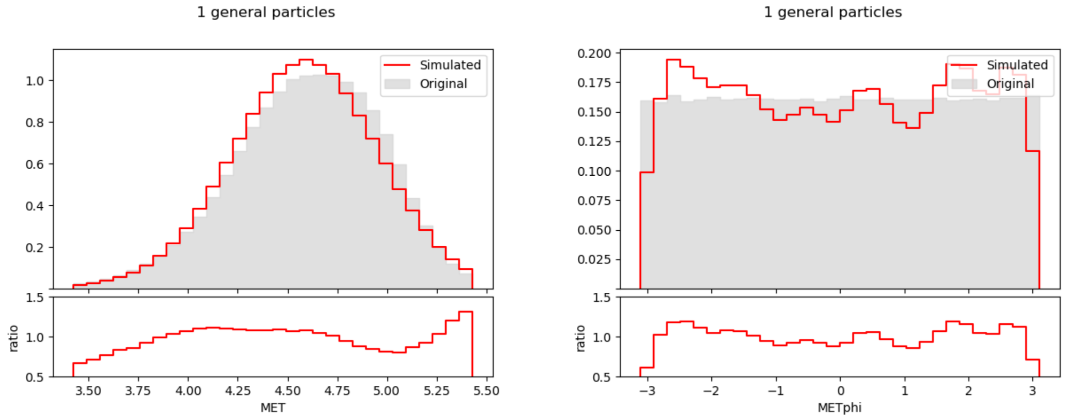


Figure E.143. Event `MET` and `METphi` distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$).

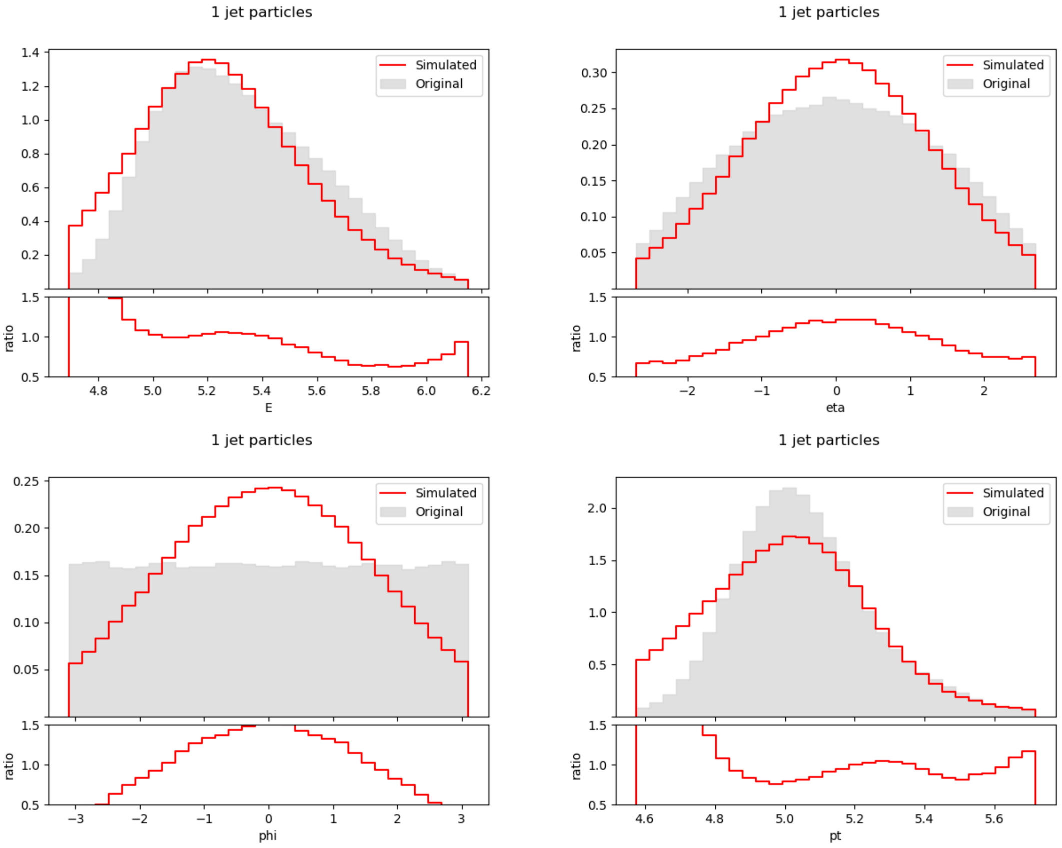


Figure E.144. First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$).

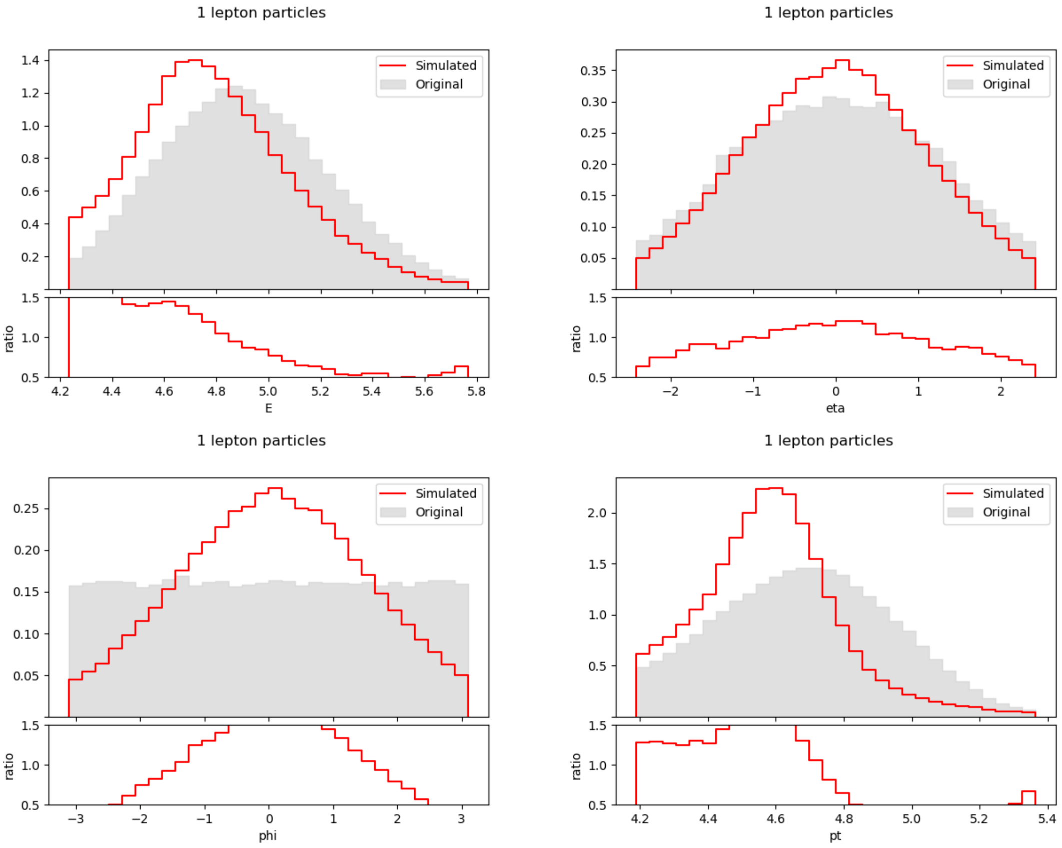


Figure E.145. First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$).

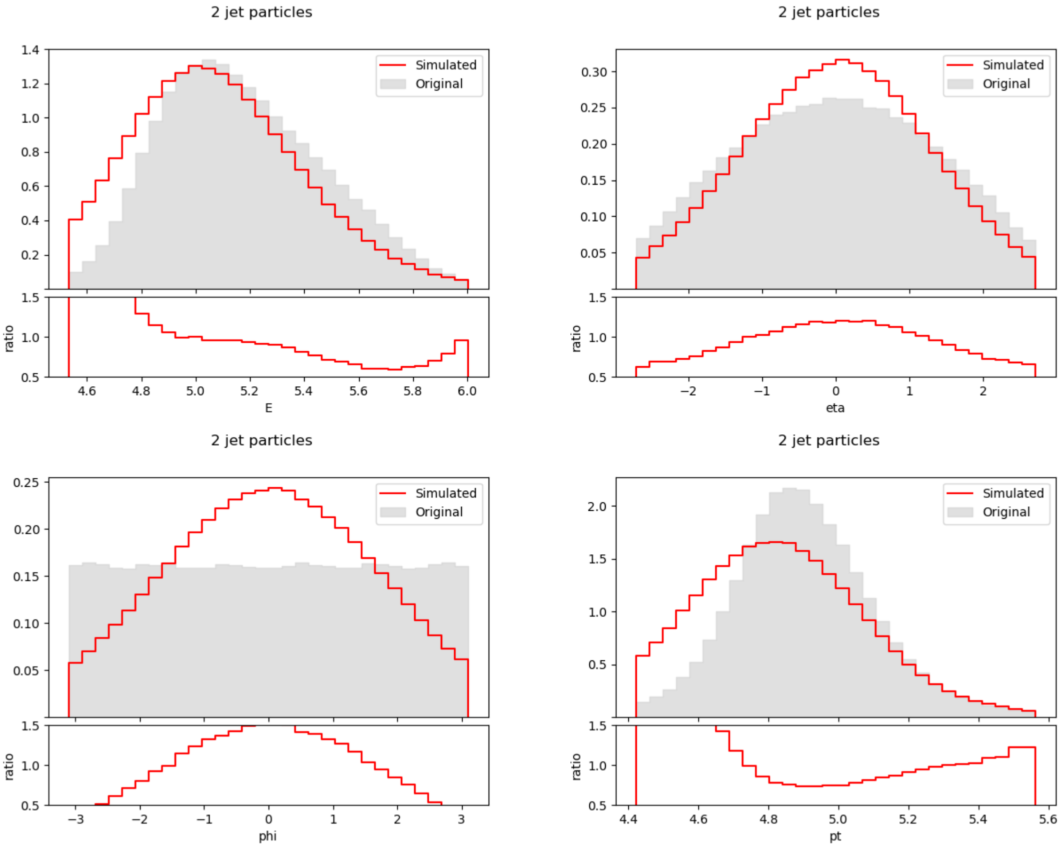


Figure E.146. Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 20$).

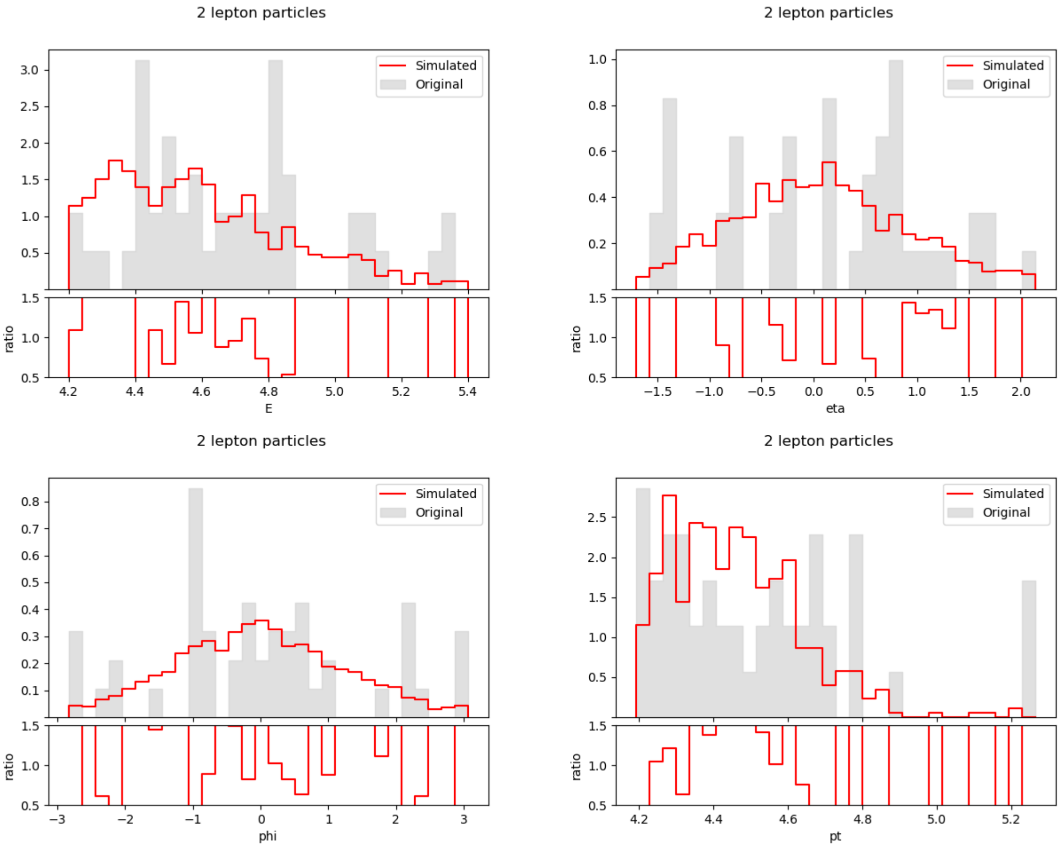


Figure E.147. Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 20$).

$\alpha = 0.3, \gamma = 50$

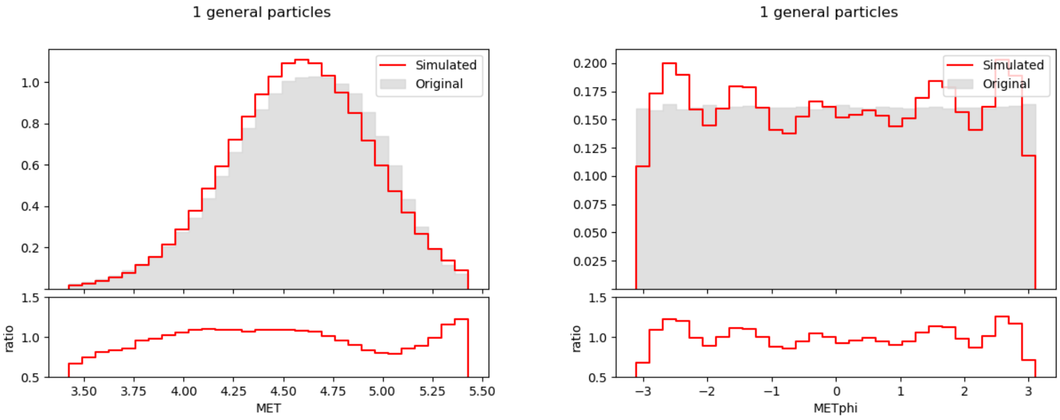


Figure E.148. Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$).

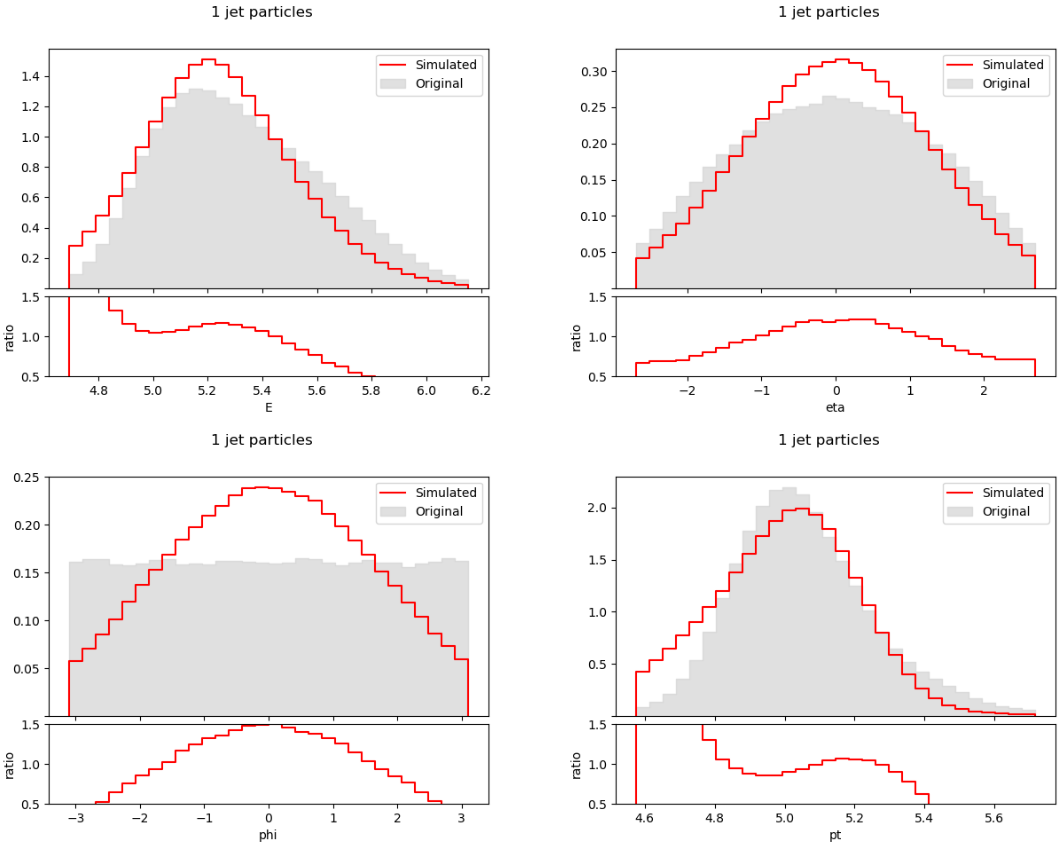


Figure E.149. First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$).

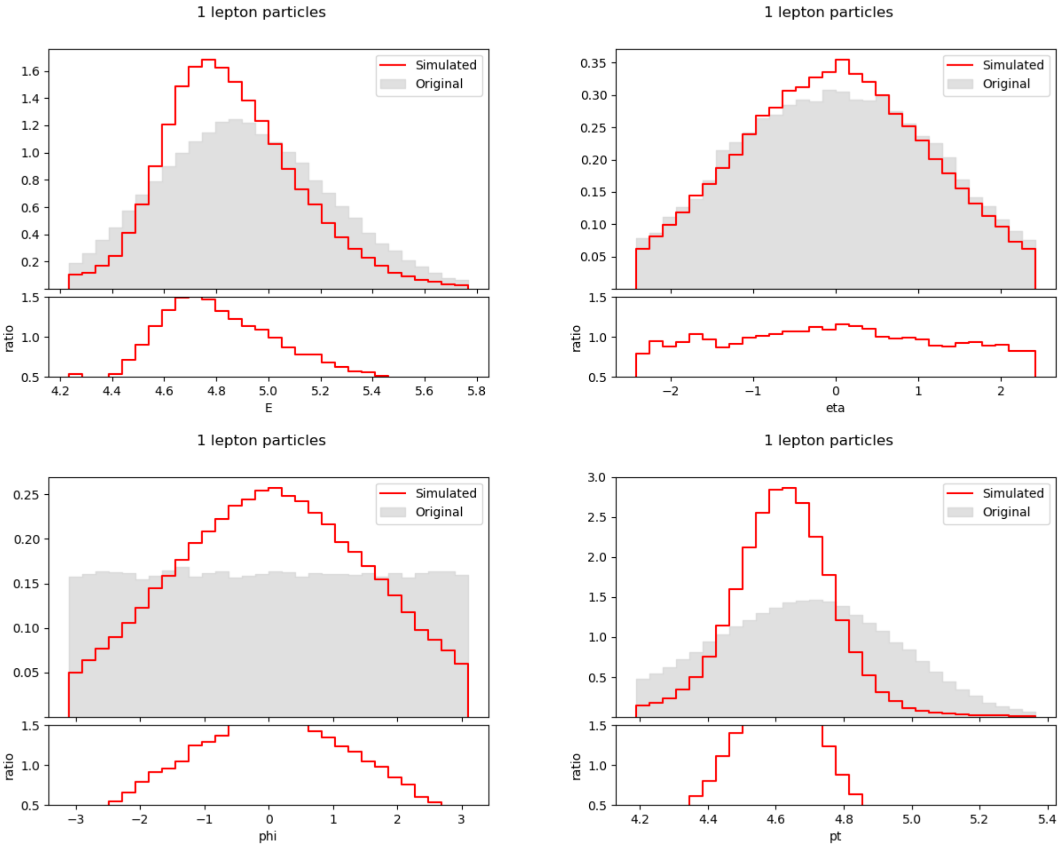


Figure E.150. First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$).

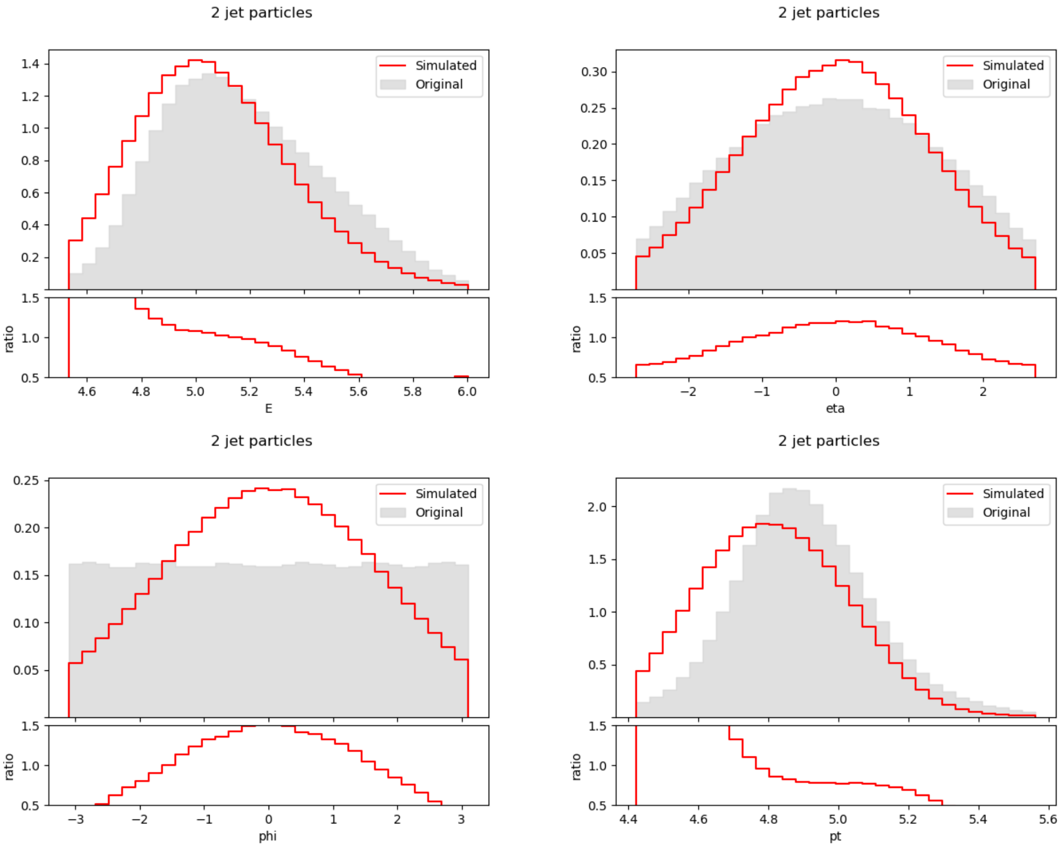


Figure E.151. Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 50$).

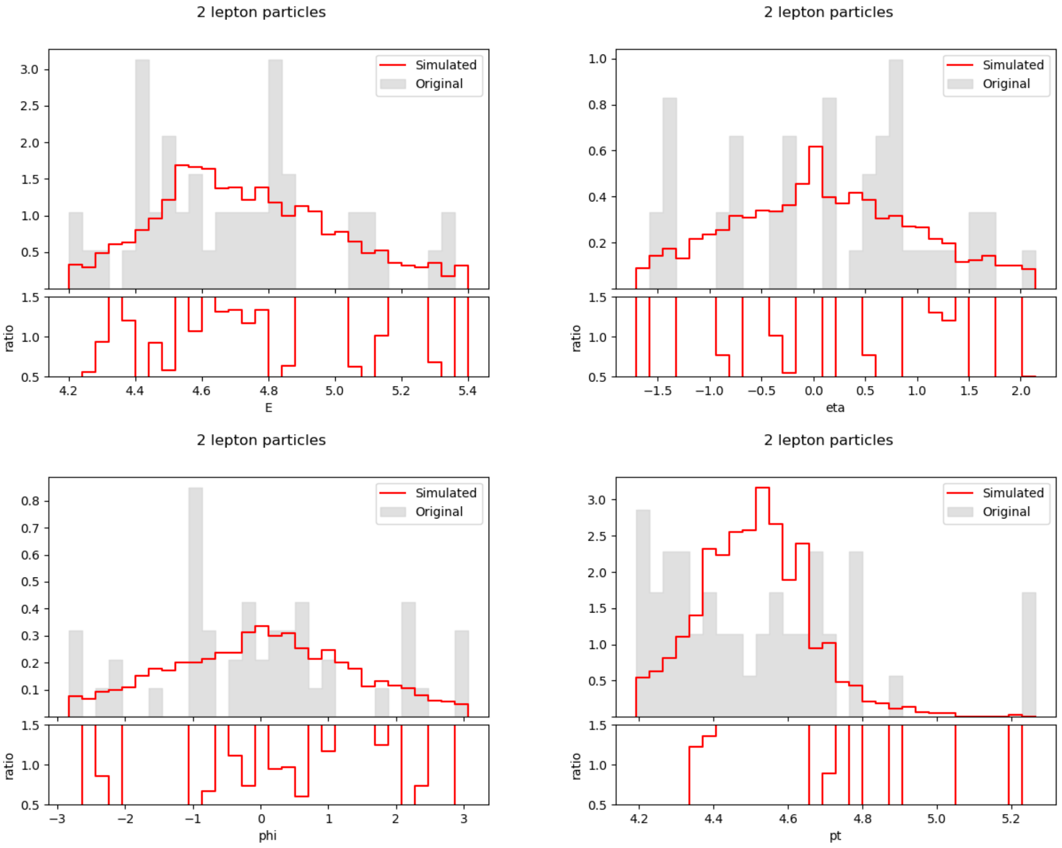


Figure E.152. Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 50$).

$\alpha = 0.3$, $\gamma = 100$

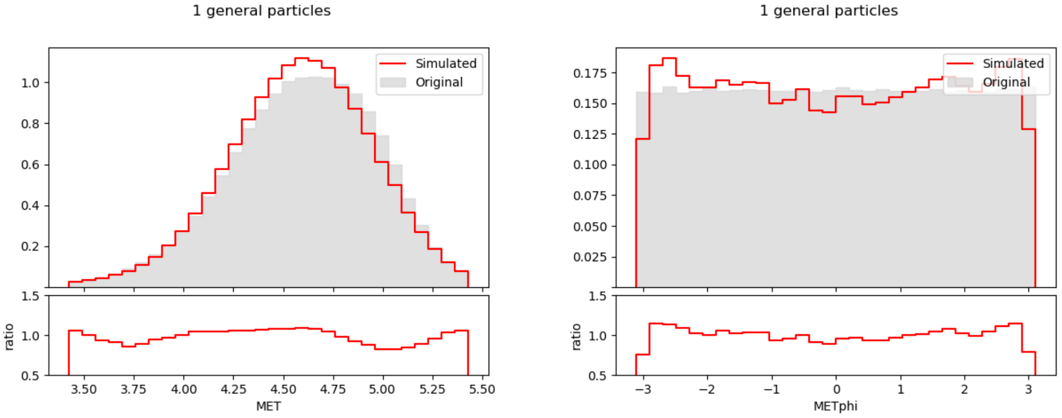


Figure E.153. Event MET and METphi distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$).

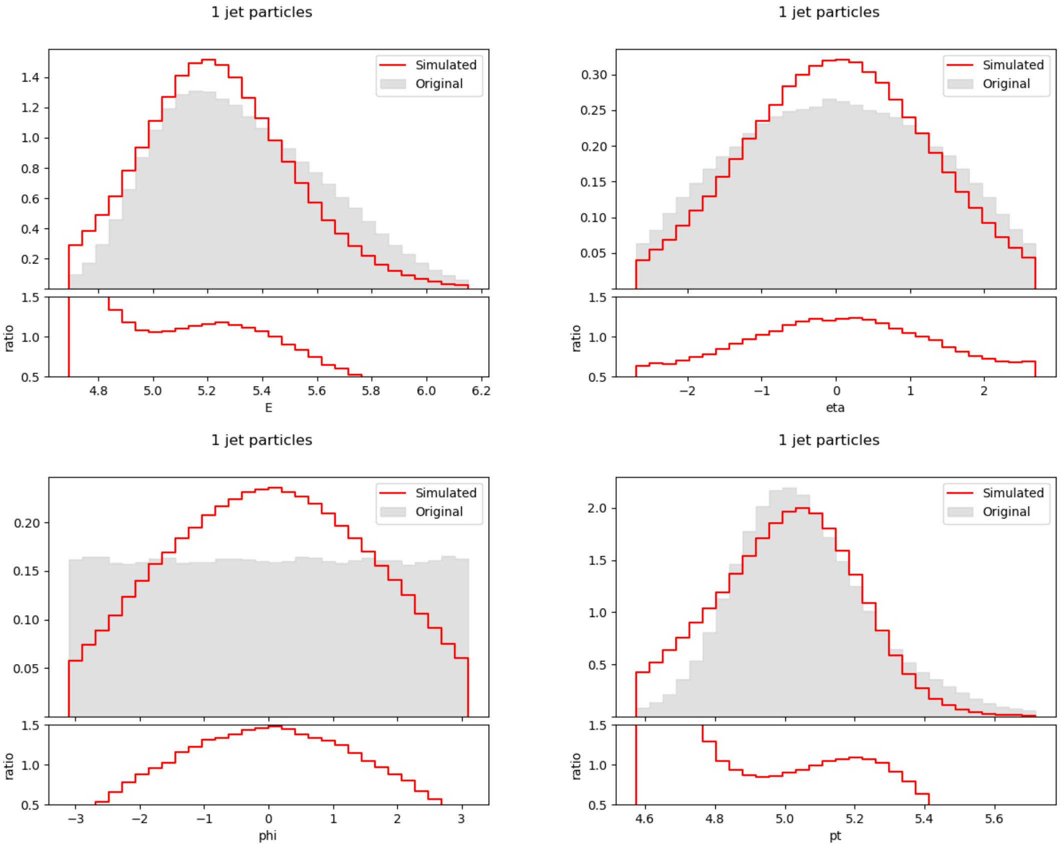


Figure E.154. First jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$).

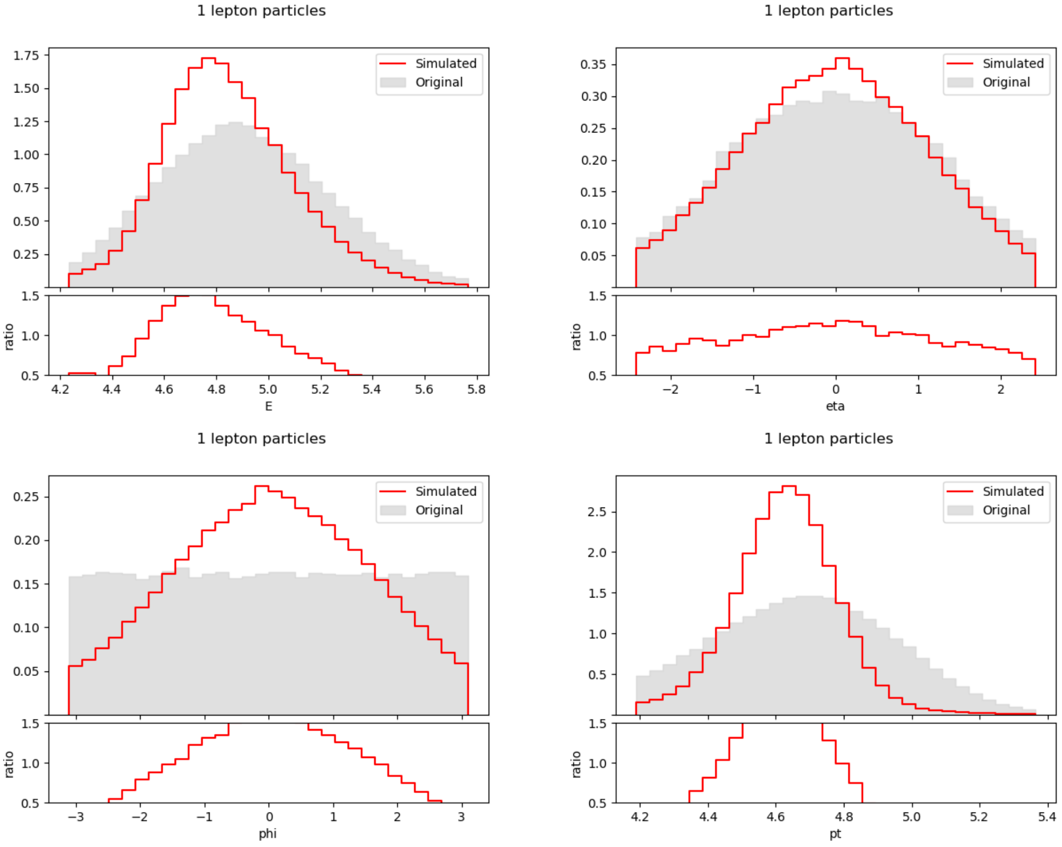


Figure E.155. First lepton particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$).

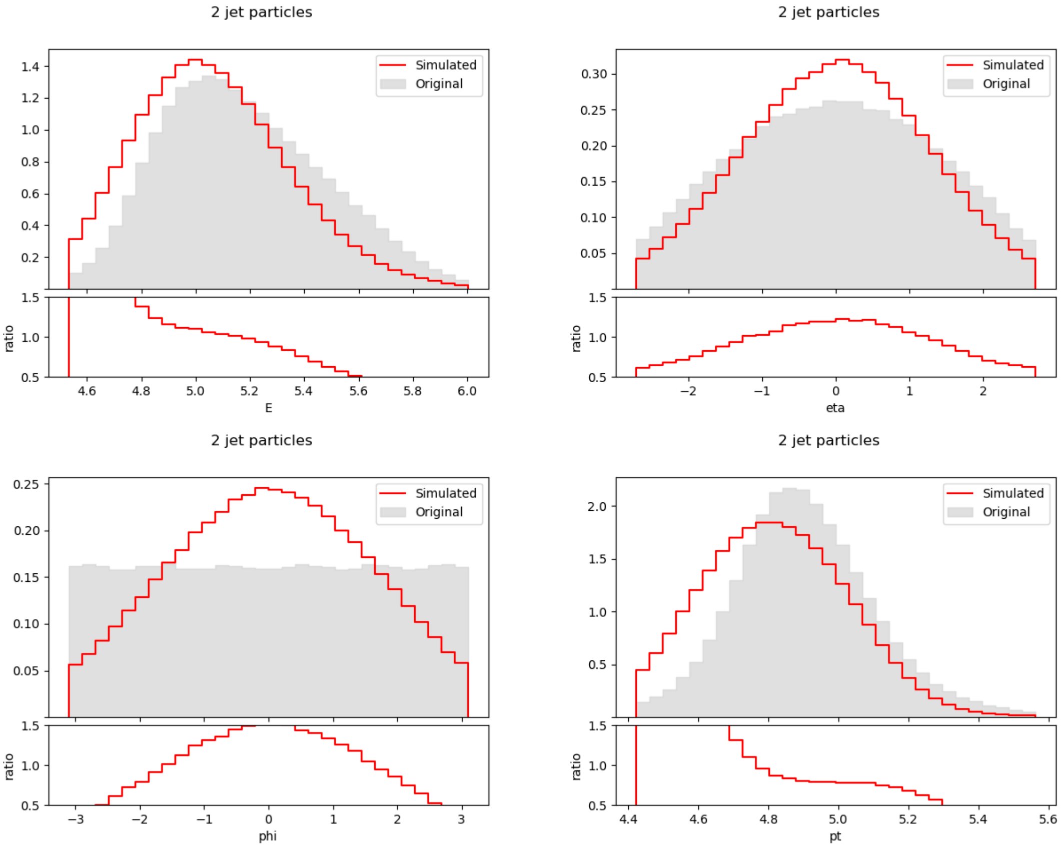


Figure E.156. Second jet particle feature distributions using a BGMM ($\alpha = 0.3$, $\gamma = 100$).

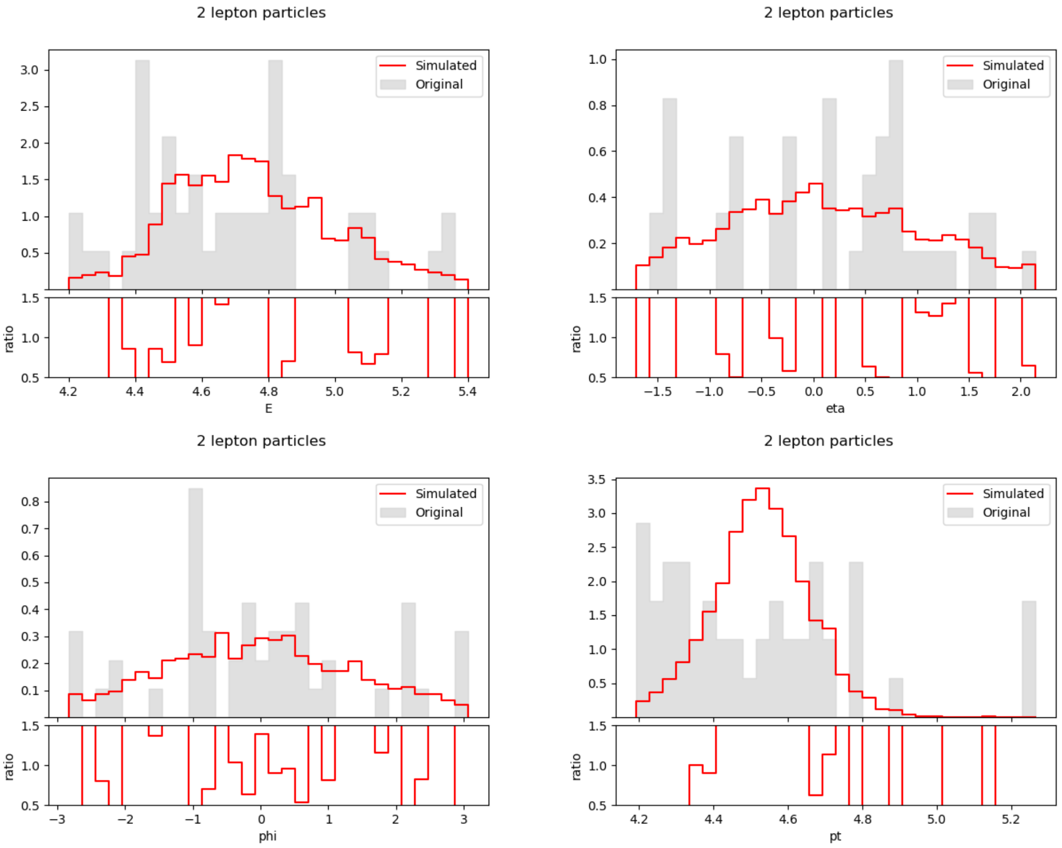


Figure E.157. Second lepton particle feature distributions using BGMM ($\alpha = 0.3$, $\gamma = 100$).

APPENDIX F
Sustainable Development Goals

In this appendix we discuss the degree of relationship of our work with the Sustainable Development Goals (SDG), established by the United Nations.

Sustainable Development Goals	**High**	**Medium**	**Low**	**Unrelated**
SDG 1. **No poverty.**				X
SDG 2. **Zero hunger.**				X
SDG 3. **Good health and well-being.**				X
SDG 4. **Quality education.**				X
SDG 5. **Gender equality.**				X
SDG 6. **Clean water and sanitation.**				X
SDG 7. **Affordable and clean energy.**		X		
SDG 8. **Decent work and economic growth.**				X
SDG 9. **Industry, innovation and infrastructure.**			X	
SDG 10. **Reduced inequalities.**				X
SDG 11. **Sustainable cities and communities.**				X
SDG 12. **Responsible consumption and production.**				X
SDG 13. **Climate action.**	X			
SDG 14. **Life below water.**				X
SDG 15. **Life on land.**				X
SDG 16. **Peace, justice and strong institutions.**				X
SDG 17. **Partnerships for the goals.**			X	

Discussion on the relationship of this work with the selected SDGs

From the previously mentioned Sustainable Development Goals, this work is mainly related to four of them.

In the first place, it presents a broad relationship with the objective of **Climate action**, as one of the main objectives of this work consisted in achieving the fastest way of generating physics events when compared to traditional methods while keeping the accuracy of those events as high as possible. By completing this objective, we managed to reduce the amount of energy consumption used in computing time since the required computers for the task of generating events will be on for a lower amount of time. This reduction will be even more beneficial in the future, when a huge number of events will need to be generated, and the energy saving will be substantial.

In the second place, it is also related to the objective of **Affordable and clean energy** because, even considering that it consumes a high amount of energy, CERN obtains the required electricity mainly from France, where the 88% of energy production is carbon-free. Moreover, CERN has decided that by 2024, its direct greenhouse emissions will be reduced by 28% by replacing fluorinated gases with carbon dioxide[1], which has several thousand times lower global-warming potential. Moreover, the reduction in terms of energy consumption obtained by our models when compared to current methods has also a great impact as it helps saving more energy.

In the third place, it keeps a certain relationship with the objective of **Industry, innovation and infrastructure** by contributing to the potential confirmation of new physics models. This could lead to a revolution in some processes that are used today in the industry, reaching in this way a high innovation degree thanks to the experiments that are executed in the ATLAS detector located at CERN, and that our work helps to develop in a more efficient way.

Finally, in the fourth place, it presents some relationship with the objective of **Partnerships for the goals**, as this work is the result of a collaboration with the *Instituto de Física Corpuscular* (*Centro Mixto CSIC-Universitat de València*), having as its mission to combine the experiments that are executed in the investigation field of High Energy Physics with the capacity of simulating data in an accurate and efficient way of the field of Artificial Intelligence to reach the previously mentioned objectives.

We believe that this project is not widely related to the rest of the SDGs, as it consists of a low-level investigation of physics aspects whose discoveries could lead to new applications that could contribute to the other objectives in the future.

Therefore, we conclude that the project is committed in different ways to the SDGs established in 2015 by the United Nations in the 2030 Agenda for Sustainable Development. These goals define a great opportunity for all of us to begin a new journey that allows to improve the quality of life, while maintaining awareness of the impact of any project during its implementation.

[1] https://cerncourier.com/a/cern-publishes-first-environmental-report/