Looking into Elliptic Flow in Heavy-ion collisions at LHC energies using Deep Neural Network

M.Sc. Thesis

By

Bhavin Jogi

Department of Physics
Indian Institute of Technology Indore
June 2022

Looking into Elliptic Flow in Heavy-ion collisions at LHC energies using Deep Neural Network

A Thesis

Submitted in partial fulfillment of the requirements for the award of the degree

of

Master of Science

By

Bhavin Jogi

Department of Physics
Indian Institute of Technology Indore
June 2022

INDIAN INSTITUTE OF TECHNOLOGY INDORE

CANDIDATE'S DECLARATION

I hereby certify that the work which is presented in this thesis entitled **Looking into Elliptic Flow in Heavy-ion collisions at LHC energies using Deep Neural Network** in the partial fulfillment of the requirements for the award of the degree of **Master of Science** and submitted in the **Department of Physics, Indian Institute of Technology Indore**, is an authentic record of my own work carried out during the time period from July 2021 to June 2022 under the supervision of Dr. Raghunath Sahoo, Professor, Department of Physics, Indian Institute of Technology Indore.

The matter presented in this thesis has not been submitted by me or by anyone else for the award of any other degree of this or any institute.

05/06/2022
Signature of student with date
(Bhavin Jogi)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Signature of the Supervisor
Prof. Raghunath Sahoo

Dr. Raghunath Sahoo | डॉ. रघुनाथ साहू
Professor | प्राध्यापक
Department of Physics | भौतिकी विभाग
Indian Institute of Technology Indore, India
भारतीय प्रौद्योगिकी संस्थान इन्दौर, भारत

Bhavin Jogi has successfully given his M.Sc. Oral Examination held on **03 June 2022**.

Signature of supervisor of MSc thesis
Date: 03/06/22

Convener, DPGC
Date: 07/06/2022

Signature of PSPC member no. 1
Date: 06/06/2022

Signature of PSPC member no. 2
Date: 07/06/2022

Acknowledgement

First and foremost, I would like to express my gratitude to my supervisor Prof. Raghunath Sahoo for giving me this opportunity for working under his guidance. I am thankful for his continuous support and encouragement in every way possible. I would also like to thank my PSPC committee members Prof. Subhendu Rakshit and Prof. Manavendra Mahato for their support.

I am very thankful to all my lab members, Experimental High Energy Physics group, for their constant support and fruitful discussions. I want to thank Mr. Neelkamal Mallick and Mr. Suraj Prasad for continuously helping me throughout the project work. I am thankful to my friends for their constant support.

Abstract

The goal of relativistic heavy-ion collision program is to create and study a hot and dense thermalized state of matter known as the quark-gluon plasma (QGP). Several indirect probes are available to study the formation and properties of QGP in such collisions as direct measurements are not possible due to the very transient nature of the medium. In heavy-ion collisions, the anisotropic flow is one such probes which refers to the final state momentum anisotropy caused by the initial spatial anisotropy. It can be very helpful to study the medium evolution in heavy-ion collisions. Elliptic flow is defined as the second order Fourier coefficient of azimuthal momentum distribution. In this work, we use Deep Neural Network (DNN) to estimate event-by-event elliptic flow coefficient (v_2) and flow vector (q_2). The DNN is trained on Pb-Pb collisions at $\sqrt{s_{\rm NN}} = 5.02$ TeV minimum bias events simulated with a multi-phase transport (AMPT) model. We find that, the DNN is efficient to learn from minimum bias events and predict v_2 and q_2 simultaneously for minimium bias events as well as for events with different centralities.

Contents

List of Figures

Chapter 1

Introduction

1.1 Relativistic Heavy-ion Collisions and Quark-Gluon Plasma

Around a hundred years ago, atoms were thought to be impenetrable and resemble fundamental objects like miniature versions of billiard balls. Atoms were considered as the smallest pieces of matter. From Rutherford's α-particle scattering experiment, we now know that atoms do have inner structures. At the center of each atom, there is a very high dense nucleus which accounts for almost all of its mass and have a net positive charge. In the outer region of the atom, there are tiny light weight particles known as electrons, carrying negative charge which makes an atom charge-neutral.

The central nucleus is made up of hadrons, known as protons and neutrons each carrying positive and neutral charges, respectively. Among these three, proton, neutron and electron, electron appears to be a true fundamental particle having no inner structure. But proton and neutron do have inner structures. They are found to be clusters of smaller fundamental particles called as the quarks and gluons bound under the strong force. The force carrier particles of strong force are known as gluons.

It is believed that just after the Big Bang, the universe was filled with a very hot and dense state of quarks and gluons [1]. Quarks and gluons form a

very hot and dense thermally equilibrated state of matter, known as quark-gluon plasma (QGP), which is extremely short-lived (lifetime of the order of fm/c). Under regular conditions of temperature and energy density, quarks and gluons are confined inside hadrons. This phenomena is well understood in QCD, known as the Color Confinement and is attributed to the asymptotic freedom. It tells that, free quarks and gluons can not exist beyond hadronic dimensions as the strong coupling constant (α_s) increases sharply as one tries to separate the quark pairs. In other way, it also tells that, with increasing energy density, the strength of strong interaction becomes weaker (*i.e.* α_s decreases at larger energy) [2]. Such high energy density can be achieved in experiments of heavy-ion collisions in which two nuclei are accelerated up to relativistic speeds and then are smashed into each other. As a result, large number of particles are produced in the final state. Figure 1.1 shows a typical heavy-ion collision inside a cylindrical barrel shaped detector with the colored tracks showing the produced particles as they interact with the detector materials. The goal of heavy-ion collision programs such as the Large Hadron Collider (LHC), CERN, Switzerland and the Relativistic Heavy Ion Collider (RHIC), BNL, USA is to produce and study the properties of this primordial state of matter known as the quark-gluon plasma (QGP) and recreate similar conditions as of the Big Bang in a miniature scale in the labs.

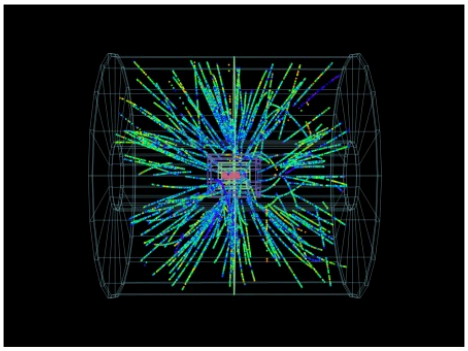

Figure 1.1: Production of large number of particles in heavy-ion collision [3]

Under the extreme conditions of pressure and energy during the ultra-relativistic collisions of nuclear matter, the hadrons melt down and exhibit a

free state of its constituent particles, quarks and gluons, which upon thermalization produce the QGP medium for a very short duration. Figure 1.2 shows

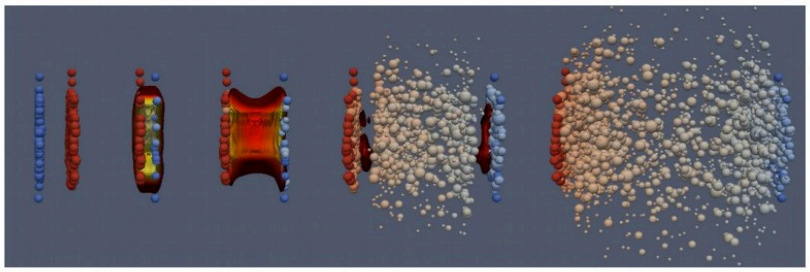

Figure 1.2: Different stages of heavy-ion collisions [4]

the various stages of a heavy-on collision event. In heavy-ion collisions, QGP is formed by colliding the heavy nuclei at very high speeds (approximately 99.999% of the speed of light, c). At such high speeds, the nuclei are Lorentz contracted and the energy per nucleon in the center of mass frame is very high (of the order of a few GeV). The contraction of nuclei causes them to interact during the collisions only. Due to Lorentz contraction of nuclei, the energy deposition is maximum during the collision. The transverse area of the overlap region is different for each collision. A large overlap corresponds to a large number of participant particles, while a small overlap takes a small number of participants particles. After the collision of heavy nuclei at the center of mass energy of the order of TeV, the constituent particles of proton and neutron which are quarks and gluons (also known as partons) are deconfined. But they are not in thermal equilibrium. The partons collide inelastically and thermal equilibrium is achieved as the system expands and cools down. The mixture of quarks and gluons in thermal equilibrium is the quark-gluon plasma. As the system expands and cools down further, the hadronization process begins. Hadronization is a complex mechanism in which quarks and gluons combine to form hadrons. After a specific temperature known as the chemical freeze-out temperature, T_{ch}, the inelastic collisions inside the medium cease and the chemical equilibrium is achieved. At this point, the formation of new hadrons is ceased. All the particles remain in the hadron gas phase thereafter and undergo elastic collisions only. Eventually, as the system expands further, the

system size becomes larger than the mean free path of the particles in the hadron gas and hence no further exchange of momentum could occur. This process is called the kinetic freeze-out and the temperature is called as the kinetic freeze-out temperature ($T_{\rm kin}$). Particles carrying fixed momentum travel towards the detector. An event is denoted by a single collision between a pair of nuclei. Each event can produce thousands of charged particles depending upon the type of colliding nuclei and their center of mass energy.

We can not observe any evidence of the possible formation of the QGP medium directly due to its very short lifetime. But we observe some indirect signatures of QGP such as jet-quenching [5], J/ψ suppression [6], strangeness enhancement, etc [7]. These signatures suggest that the formation of the QGP medium is highly probable in heavy-ion collisions at LHC and RHIC energies.

Every event in heavy-ion collisions can generally be classified by the impact parameter (b) between two colliding nuclei. It is defined as the distance between their centers in the plane transverse to the initial beam axis. The impact parameter defines the overlap region of the colliding nuclei. For small b, the overlap between the nuclei is large and *vice versa*. Impact parameter is also related to the number of participant nucleons (N_{part}) in the collision. For small b, N_{part} is large and for large b, N_{part} is small. The impact parameter can not be measured directly in the experiments. So experimentally, the events are classified using the centrality of collisions as a proxy to impact parameter. Centrality of the collision can be defined using the multiplicity of an event or the energy of the spectators (forward energy). The event multiplicity distribution is sliced in segments in such a way that the 0-5% centrality class indicates those five percent events which have the highest multiplicity.

In the next section, we will discuss about the anisotropic flow which is an important observable in heavy-ion collisions and tells us more about the medium formation and its evolution.

1.2 Anisotropic Flow:

In nucleus-nucleus collisions, the reaction plane is defined by the beam direction (z) and the impact parameter direction (b). The $x-y$ plane is defined as the transverse plane or the azimuthal plane (Fig. 1.3). Here, z-axis is the beam direction, ϕ is the azimuthal angle of the produced particles and $\psi_{\rm RP}$ is the reaction plane angle.

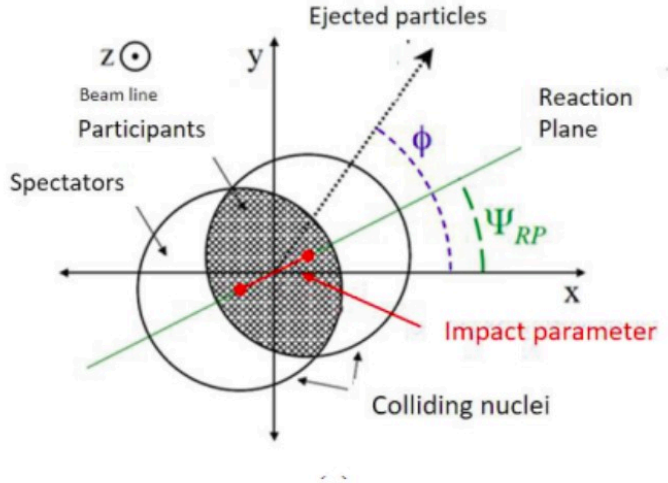

Figure 1.3: Initial collisional geometry of two nuclei

The collective expansion of the final state particles is called flow. In a non-central collision, the overlap region of the nuclei is like an almond-shaped. During the system evolution, if the system is reached to thermal equilibrium, then it can be described in terms of thermodynamic quantities like T and P. Due to the spatial anisotropy of the overlap region, a strong pressure gradient is created. The pressure gradients ensures the anisotropy in the momentum of the final state outgoing particles in the transverse plane. So, the initial spatial anisotropy is converted into the momentum space anisotropy (Fig. 1.4). The non-zero finite azimuthal anisotropy in momentum space can be expressed as the Fourier series expansion in the azimuthal angle ϕ [8, 9, 10]:

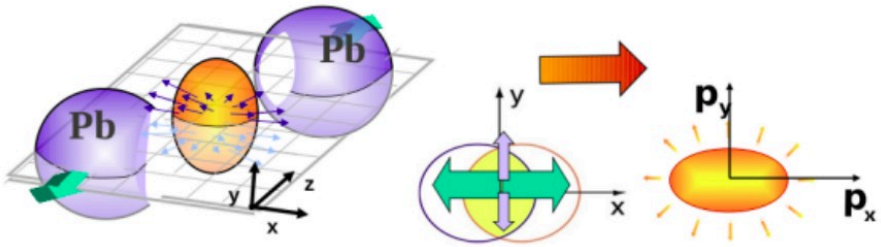

Figure 1.4: Conversion of spatial anisotropy into momentum anisotropy in Pb-Pb collision

$$E\frac{d^3N}{dp^3} = \frac{d^2N}{2\pi p_T dp_T dy}\left(1 + 2\sum_{n=1}^{\infty} v_n \cos[n(\phi - \psi_n)]\right) \tag{1.1}$$

Here, ϕ is the azimuthal angle of outgoing particles and ψ_n is the n^{th} order harmonic event plane angle determined by the initial geometry of the system [11]. The Fourier coefficient v_n is called the flow coefficient and can be written as:

$$v_n = \langle \cos[n(\phi - \psi_n)] \rangle \tag{1.2}$$

The angle brackets indicate an average over particles in all events. v_n is the magnitude of the anisotropic flow of different order n. The coefficients v_1 corresponding to $n = 1$ is often called the directed flow, v_2 corresponding to $n = 2$ is called the elliptic flow and v_3 corresponding to $n = 3$ is called the triangular flow and so on.

The flow vectors of order n, Q_n are defined as [11, 12, 13]:

$$Q_{n,x} = Q_n \cos(n\psi_n) = \sum_{i}^{M} \cos(n\phi_i) \tag{1.3}$$

$$Q_{n,y} = Q_n \sin(n\psi_n) = \sum_{i}^{M} \sin(n\phi_i) \tag{1.4}$$

Here, ψ_n is the event plane angle of order n. We can write Q_n as:

$$Q_n = \sum_{i}^{M} e^{in\phi_i} \tag{1.5}$$

The reduced flow vector can also be defined as q_n, where:

$$q_n = \frac{Q_n}{\sqrt{M}} = \frac{\sum_i^M e^{in\phi_i}}{\sqrt{M}} \tag{1.6}$$

where M is the multiplicity of an event and ϕ_i is the azimuthal angle of i^{th} particle. The flow coefficients v_n and the flow vector q_n are observed to be almost correlated linearly [12].

1.3 Motivation

In every heavy-ion collisions, very large number of particles are generated depending upon the initial geometry. Each final state particle has various observables associated to it, such as charge (q), mass (m), transverse momentum (p_T), azimuthal angle (ϕ), polar angle (θ), etc. So in every collisions, a very large dataset is generated. In recent times, the machine learning (ML) algorithms have become very popular to handle such large amount of data and extract the useful information from it [14, 15, 16, 17]. One of very popular ML techniques is neural network (NN). In this project, we have tried to implement deep neural network (DNN) based regression technique to predict the elliptic flow coefficient (v_2) and flow vector (q_2) using the final state particle properties such as event-by-event transverse momentum (p_T) spectra, pseudorapidity (η) spectra and azimuthal angle (ϕ) spectra for Pb-Pb collisions at $\sqrt{s_{\rm NN}} = 5.02$ TeV.

In next section, we discuss about machine learning and deep neural network (DNN) architecture. Then we will see the methodology of this project in which we will see the input observables, output observables and the DNN model structure used. At the end, the results are presented in chapter 4.

Chapter 2

Deep Learning

2.1 Machine Learning

Machine Learning (ML), data science and statistics are very important fields that are related to large data. They describe how to learn from training data and make predictions. Machine learning is the study of computer algorithms that improve through experience. It involves the computer discovering how it can perform tasks without being programmed explicitly to do so. It is about making computers adapt or learn the actions (whether these are classifying objects, or predicting an output for a given input) so that these actions get more accurate. The accuracy is the measure of how the machine can perform a task correctly.

There are different types of problems in machine learning: Regression, classification, time-series forecasting etc. When there in only one numerical output value for every given input, the problem is of regression type. For example, prediction of house prize. If the input needs to be classified in one of the different classes, then it is classification type of problem. For example, classifying the animal into dog or cat. If the output of any input is dependent on time-series data, the it is the time-series forecasting type of problem. For example, prediction of temperature of a region.

There are mainly three types of machine learning algorithms: Supervised, Unsupervised and Reinforcement. In supervised learning, a training set of

examples with labeled outputs is given to machine, and based on the training set, the machine tries to recognize the relation pattern between input and output, and generalises to give output correctly to all possible inputs. In unsupervised learning, the training set is provided to machine without labeled output. The machine tries to recognize the pattern and grouping of data on its own and predicts the output. Reinforcement learning can be seen as a combination of supervised and unsupervised learning. The machine learns through the method of reward and punishments. The machine gets rewards (positive feedback) for the desired actions and punishments (negative feedback) for the undesired actions.

The general work flow of a supervised machine learning algorithm can be shown as Fig. (2.1). In every such problem, the first thing we do is to convert the data into a suitable form so that the machine can recognize and learn patterns in the input data. A machine learning model is prepared. Then this data is split into training and test sets. The training data set is given as input to machine. At every input of the input set, the model predicts some output. For every deviated output from the true output, the model is modified via feedback process in which model takes feedback from the amount of error in predicting the output and gets modified. After the feedback, the model is re-trained for the next input. In such a way, model learns the hidden data patterns between input and output and the model is said to be fully trained at the end if it can predict the correct output for any suitable input. This trained model can now predict output for the test set. If the loss function continues to decrease, then model is called to be underfitted or if the loss function starts increasing after several epochs of training, then the model is said to be overfitted. In case of underfitting of model, the number of training epochs are increased and model can be made more complex. While in case of overfiiting of model, the number of training epochs are reduced and model can be made simpler.

There are different machine learning techniques, but two are the most used in high energy physics which are decision tree (DT) and neural network (NN). They are widely used for classification, regression and pattern recognition. A

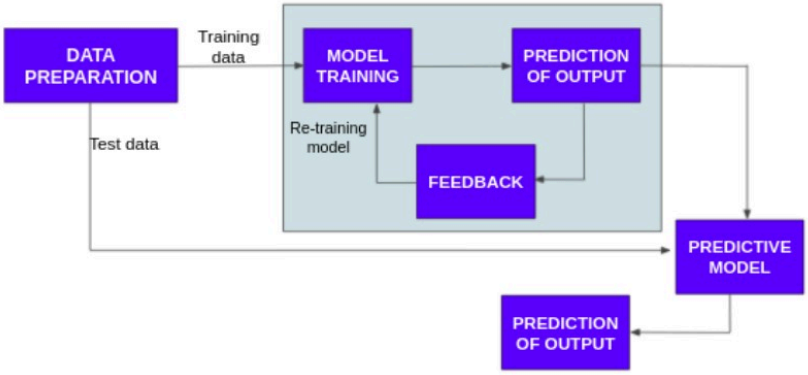

Figure 2.1: General workflow of a machine learning algorithm

decision tree takes a set of input features and splits input data recursively based on those features [14, 18]. The decision trees are usually built from top to bottom. Figure 2.2 shows the general architecture of a decision tree. NN

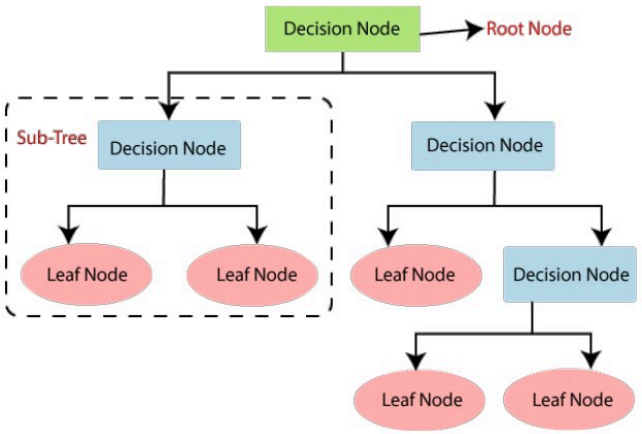

Figure 2.2: General architecture of decision tree

is a computational model which is made up of several interconnected groups of nodes similar to the human brain [16, 19, 20]. Figure 2.3 shows the basic architecture of NN.

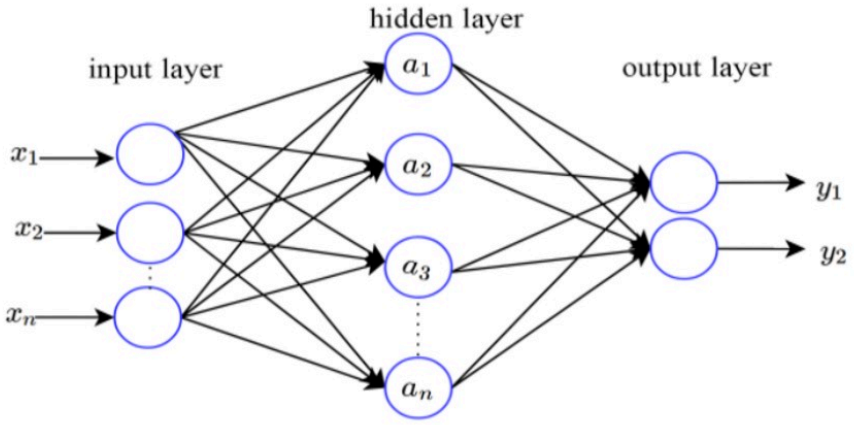

Figure 2.3: General architecture of neural network

2.2 Deep Neural Network (DNN)

Deep Learning is a subset of machine learning, where deep neural network is used for training and predicting the outputs. Deep Neural Network (DNN) means a neural network having several number of hidden layers. So the word 'Deep' in deep learning indicates several number of hidden layers in a neural network. Figure 2.3 shows a simple deep neural network having only one hidden layer. The smallest element of a neural network is a neuron which works in a similar way of neuron in human brain. It is also called a node. Several neurons together make different layers of neural network. Every layers are interconnected fully or partially depending upon the requirements. We will see different important components of a deep neural network

2.2.1 Layers

There are three types of layers in a neural network: Input layer, Hidden layers and Output layer. The input data we give to neural network is loaded to the first layer of neural network which is called the input layer. The complete input information is passed to next layer without any type of transformation. Hidden layers lie between input layer and output layer. They are called hidden in a sense that they can not be communicated directly. Most of the processes on the

input data are done by these layers and they extract the hidden pattern and features of the data. The last hidden is connected to the output layer. Output layer gives the output predicted by the network. Output layer has different number of neurons depending upon the type of problem. For regression type of problems with one output variable, output layer has only one neuron; while for classification type of problems, output layer has at least two neurons.

2.2.2 Neurons

Neurons are the elemental component of a neural network. A neuron and various processes associated to it are shown in fig. 2.4.

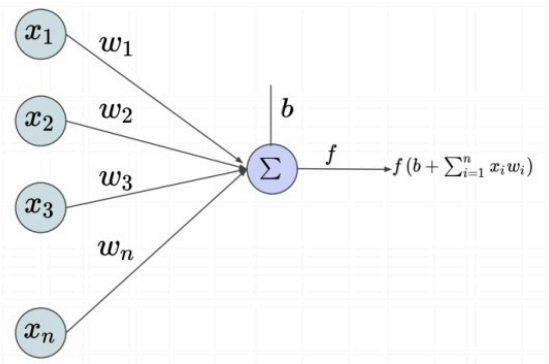

Figure 2.4: A neuron in a neural network

Every neuron takes input from different neurons of previous layer connected with it, transforms the input and gives the output to next layer. In fig. 2.4, $\{x_1, x_2, x_3, ..., x_n\}$ is a set of inputs to this neuron. Weights are the coefficients associated each input and their main job is to prioritize the inputs which help our network in learning. $\{w_1, w_2, w_3, ..., w_n\}$ is a set of weights. The bias (b) is a constant similar to the constant in any linear function $y = mx + c$. It just shifts the output value. Activation function (f) is a very important part of each layer. Any neuron should be activated or not is decided by the activation function so that the network can learn as rapidly as possible. There are many different activation functions available: Linear, Sigmoid, softmax, ReLU, etc. So, the output at each layer, in general, can be written in the form of a matrix

multiplication as $Y = f(Wx + b)$, where Y is a output column matrix, W is a coefficient matrix, x is an input column matrix and b is a bias matrix.

Linear activation function gives the output which is linearly proportional to input (Fig. 2.5). The equation of function it has is:

$$f(x) = mx + c \tag{2.1}$$

The activation function of the output layer in a regression type of problems

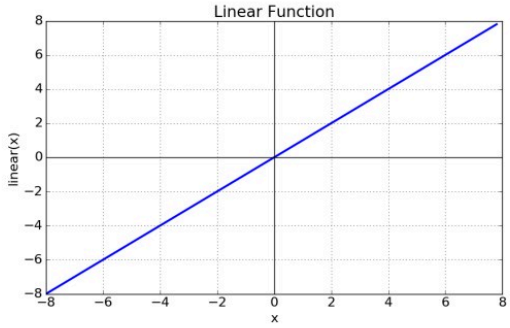

Figure 2.5: Linear activation function

is always the linear activation function. It introduces the non-linearity for the non-linear mapping of output from input because $(Wx + b)$ is just a linear expression which cannot map the input non-linearly into output. Sigmoid activation function ensures to give output value between 0 and 1 (Fig. 2.6) and its equation is:

$$\phi(z) = \frac{1}{1 + e^{-z}} \tag{2.2}$$

The activation function of the output layer in a binary classification type of problems is always the sigmoid activation function. Softmax activation function also guarantees to give output value between 0 and 1 and its equation is:

$$S(y)_i = \frac{\exp(y_i)}{\sum_{j=1}^{n} \exp(y_j)} \tag{2.3}$$

The activation function of the output layer in multiclass classification (logistic regression) type of problems is always the softmax activation function. ReLU stands for the rectified linear unit activation function (Fig. 2.7). Its equation

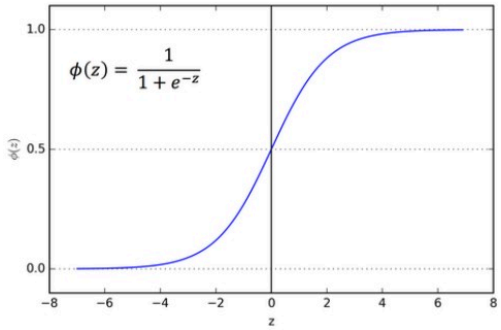

Figure 2.6: Sigmoid activation function

is:

$$R(z) = max(0, z) \tag{2.4}$$

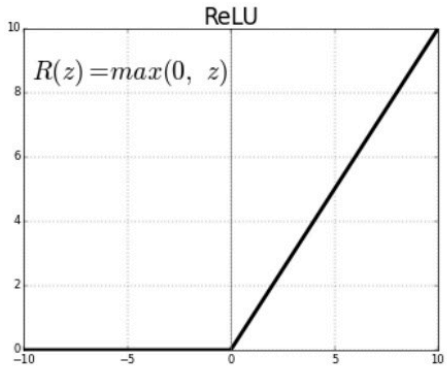

Figure 2.7: ReLU activation function

2.2.3 Loss function and Optimizer

Loss function defines a measure of error in predicted output values compared to true output values. There are different loss functions each having its own importance in a particular type of problem. Mean squared error (MSE) is used for regression type of problems, binary crossentropy is used for binary classification type of problems and categorical crossentropy is used for logistic regression type of problems. Optimizer function helps to improve the weight matrix and consequently the learning of a neural network and reduces the loss.

Examples of optimizer are stochastic gradient descent (SDG), Adam, rmsprop, etc.

These are some of the important components of a DNN. The next section is the methodology part in which we will see the workflow of this project.

Chapter 3

Methodology

In this chapter, we discuss the various stages of work done in this project. The events data used in this project is generated using AMPT. AMPT (A multi-phase transport model) is a Monte Carlo event generator for simulating the collision events of p-A and A-A at very high energies from 5 GeV to 5.5 TeV. We have used 150K minimum bias events for Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV.

3.1 Input observables

As proposed, we try to use the final state particle properties p_T spectra, η spectra and ϕ spectra to predict v_2 and q_2. So the input to DNN would be these three observables. If the initial direction of motion of colliding particle is z-axis, then the transverse momentum of any final state particle is defined as:

$$p_T = \sqrt{p_x^2 + p_y^2} \tag{3.1}$$

where p_x and p_y are the momenta of particle in x and y direction, respectively. The rapidity of any particle having Lorentz boost along z-direction can be written in terms of E and p_z as:

$$y = \frac{1}{2} \ln \left(\frac{E + p_z}{E - p_z} \right) \tag{3.2}$$

If a final state particle is emitted making an angle θ with the initial beam axis, then the rapidity of particle is:

$$y = \frac{1}{2}\ln\left(\frac{E + p_L}{E - p_L}\right) = \frac{1}{2}\ln\left[\frac{\sqrt{m^2 + p^2} + p\cos\theta}{\sqrt{m^2 + p^2} - p\cos\theta}\right] \tag{3.3}$$

At very high energy, $p >> m$, so

$$y = \frac{1}{2}\ln\left[\frac{p + p\cos\theta}{p - p\cos\theta}\right] = -\ln\left(\tan\frac{\theta}{2}\right) \equiv \eta \tag{3.4}$$

Here, we call η the pseudorapidity. The outgoing direction making an angle ϕ with the preferred x-direction in the detector is called the azimuthal angle ϕ of the emitted particle.

We are taking these three event-by-event observable spectra as input. Each of these spectra is divided into 40 bins and given as input to DNN. The input for each event contains 40 bins of η-spectra, 40 bins of ϕ-spectra and 40 bins of p_T-spectra, making it total 120 bins. Figure 3.1, 3.2 and 3.3 show the input for a random event. In the input dataset, $p_T \in [0.2, 5.0]$ GeV/c, $\eta \in [-0.8, 0.8]$ (mid-rapidity region) and $\phi \in [0, 2\pi]$.

3.2 Output observables

The output for DNN would be to predict the elliptic flow coefficient (v_2) and flow vector (q_2). From equation 1.2 & 1.6, we can write

$$v_2 = \langle\cos[2(\phi - \psi_2)]\rangle \tag{3.5}$$

and

$$q_2 = \frac{\sum_i^M e^{i2\phi}}{\sqrt{M}} \tag{3.6}$$

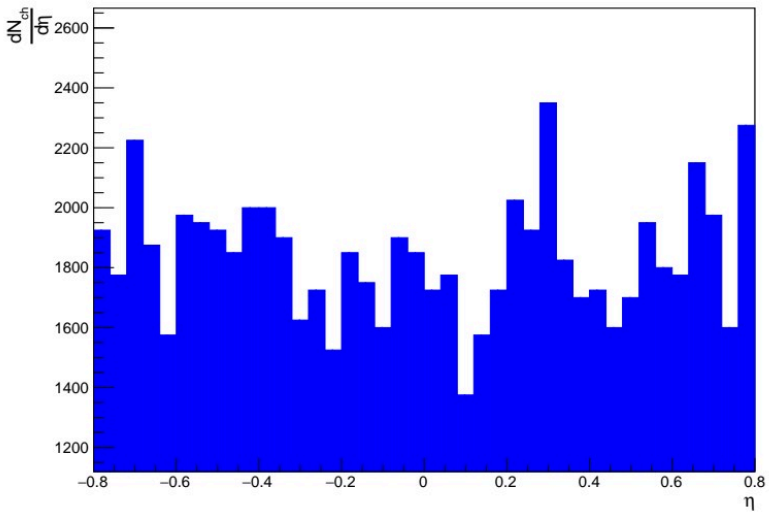

Figure 3.1: Event-by-event η-spectra of a random event of Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV generated using AMPT

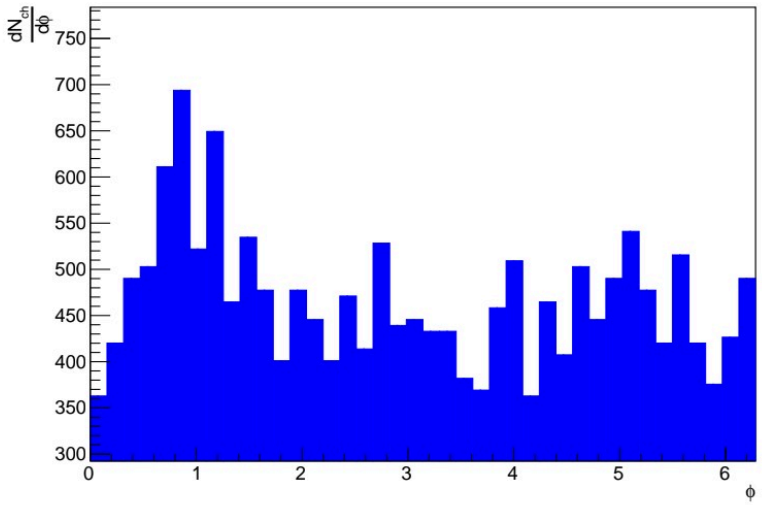

Figure 3.2: Event-by-event ϕ-spectra of a random event of Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV generated using AMPT

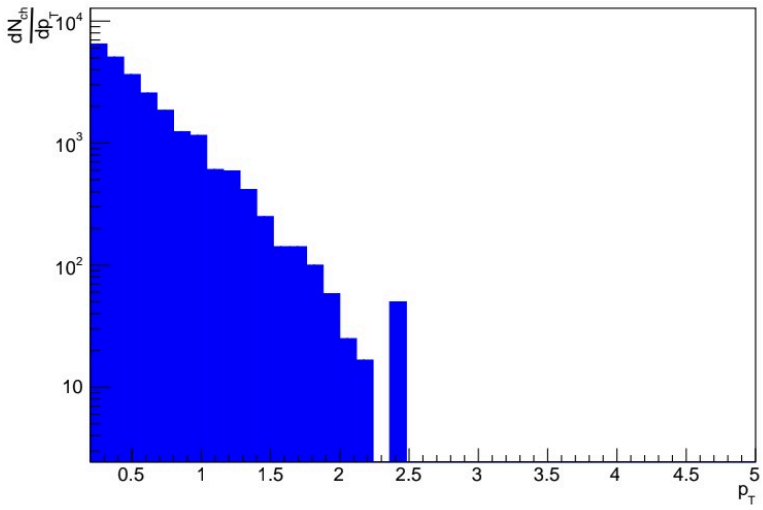

Figure 3.3: Event-by-event p_T-spectra of a random event of Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV generated using AMPT

3.3 Model definition

The deep neural network (DNN) we have used has an architecture as shown in fig. 3.4. All the layers are dense layers. The input layer takes input of 120 bins for one event at a time. This layer is connected to the first hidden layer. The model has four hidden layers having 128, 256, 256 and 256 number of nodes respectively. The last hidden layer is connected to the output layer which predicts the two outputs simultaneously namely v_2 and q_2. For each layer except the output layer, the ReLU (Eq. 2.4) activation function is used. For output layer, the linear activation function is used. The DNN has the adam optimizer for the optimization of weights and loss; and the mean-squared-error (MSE) as a measure of loss function.

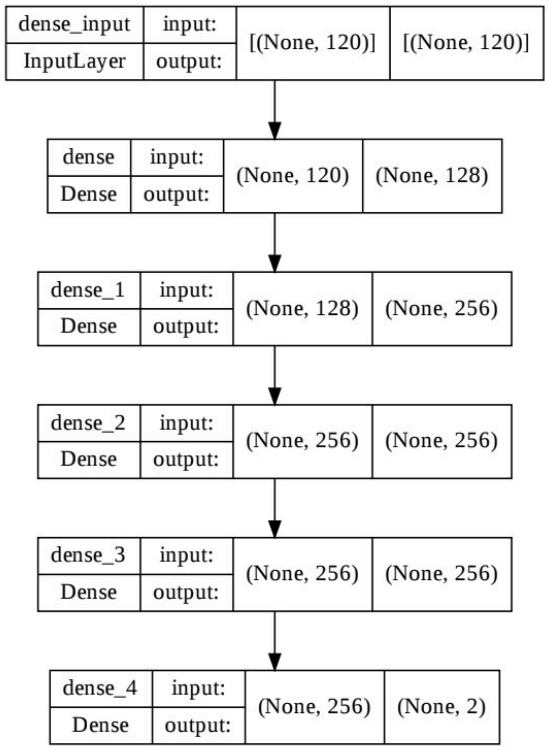

Figure 3.4: DNN architecture

Chapter 4

Results and Discussion

4.1 Training of network

We have about 150K minimum bias events for Pb-Pb collisions. We split it into training set and test set. Out of 150K events, 80% events have been used for training of the model specified in figure 3.4 and the remaining events have used for testing and validation purpose. The model has been trained for 156 epochs. The loss function as a function of epochs is shown in figure 4.1.

After the training process, the model uses the test set to predict the outputs which are v_2 and q_2. Figure 4.2 shows the prediction of v_2 by the DNN vs the true values of v_2. Similarly, fig. 4.3 shows the prediction of q_2 by the DNN vs the true value of q_2. We can see that the network gives very good predictions for test set.

4.2 Prediction

After training and testing the DNN using the minimum bias events for Pb-Pb collisions, we tried to use this trained DNN to predict v_2 and q_2 for events having different centralities. We can see that the DNN gives very good predictions of v_2 and q_2 for different centralities also as shown in fig. 4.4, 4.5. We can see that the predicted outputs of v_2 and q_2 are almost coinciding with the

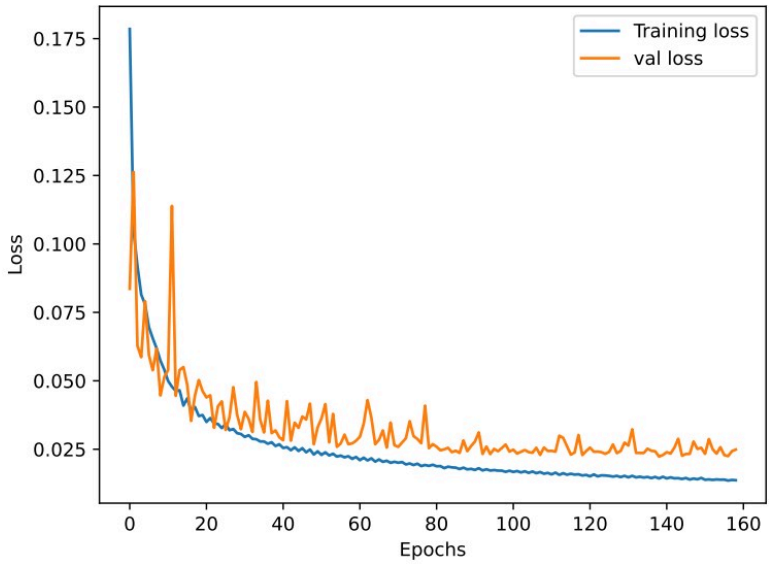

Figure 4.1: Loss function vs Epoch during the training process

actual values of v_2 and q_2 generated using AMPT. We have also compared the predicted and true values of v_2 with the ALICE data for Pb-Pb collisions at $\sqrt{s_{\mathrm{NN}}} = 5.02$ TeV [9]. And we observe that the predicted and true values of v_2 deviates from the ALICE data points at higher centralities.

4.3 v_2 vs q_2

We can see that we have almost a linear correlation between the elliptic flow coefficient (v_2) and flow vector (q_2) [12] as shown in figure 4.6. So we drew the same histogram of v_2 vs q_2 for predicted values of v_2 and q_2 by the DNN for different centralities shown. We see that the histograms for predicted values and for true values both are almost coinciding with each other for centralities shown.

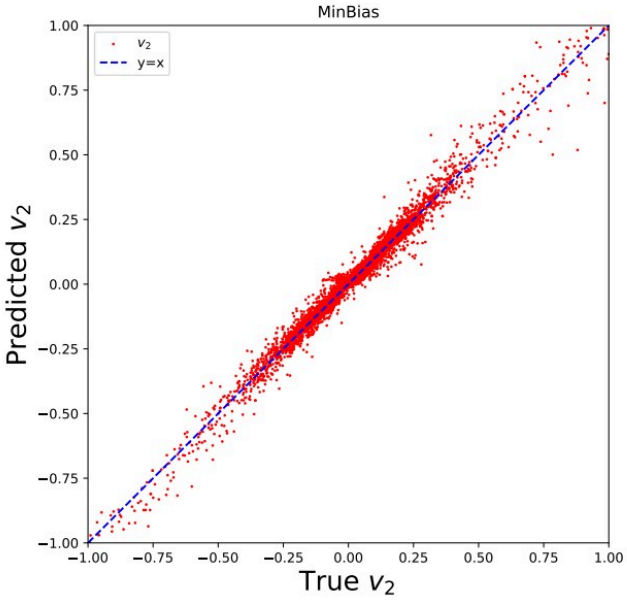

Figure 4.2: Predicted v_2 vs true v_2 for test dataset

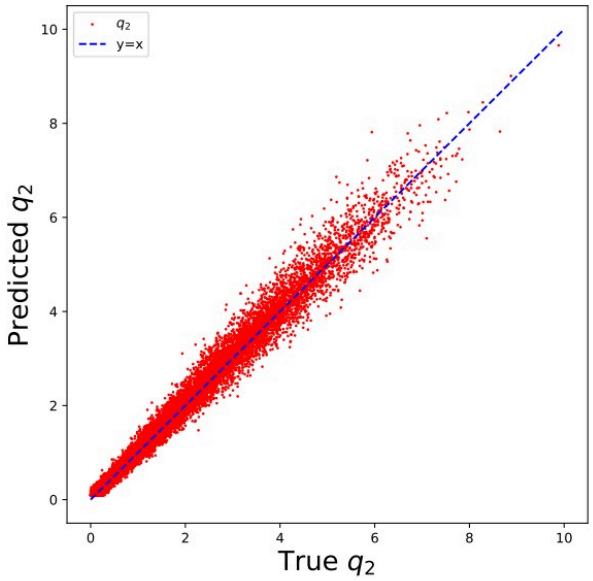

Figure 4.3: Predicted q_2 vs true q_2 for test dataset

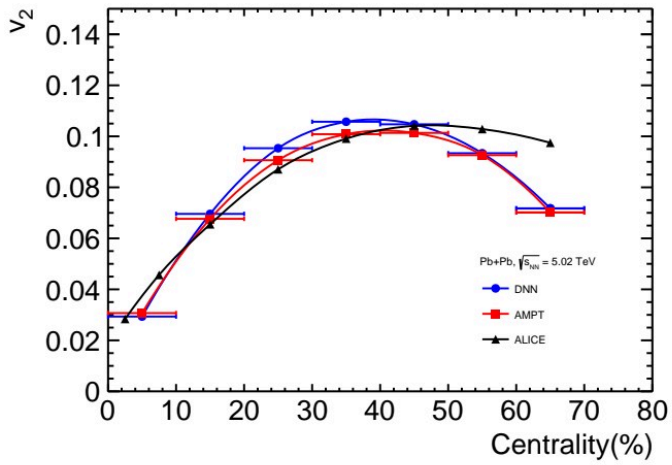

Figure 4.4: v_2 vs. centrality for Pb-Pb collisions, $\sqrt{s_{\rm NN}} = 5.02$ TeV for AMPT simulation and DNN prediction. The ALICE results are shown for comparison.

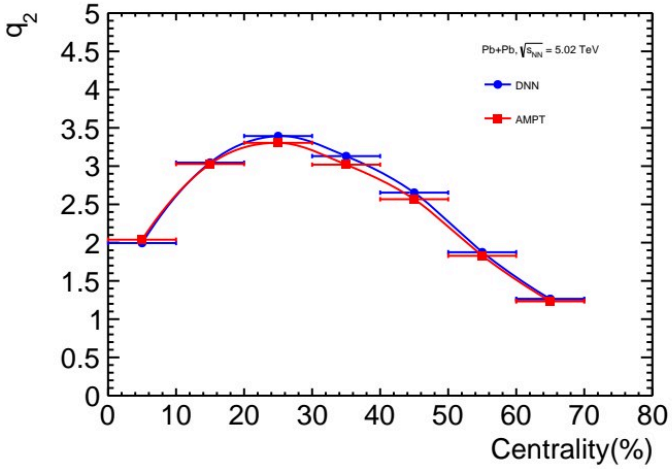

Figure 4.5: q_2 vs centrality for Pb-Pb collisions, $\sqrt{s_{\rm NN}} = 5.02$ TeV for AMPT simulation and DNN prediction.

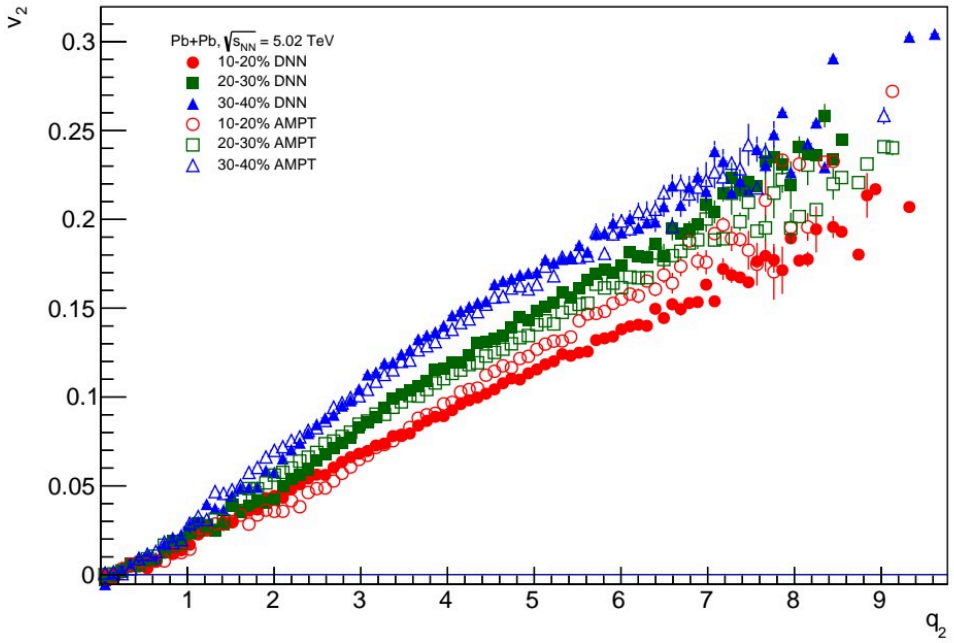

Figure 4.6: v_2 vs. q_2 for (10-20)%, (20-30)%, and (30-40)% central Pb-Pb collisions, $\sqrt{s_{\rm NN}} = 5.02$ TeV for AMPT simulation and DNN prediction.

Chapter 5

Summary and Outlook

We started with making the input dataset using the track properties of final state particles as shown in fig. 3.1, 3.2, 3.3, generated in the minimum bias Pb-Pb collisions at $\sqrt{s_{\rm NN}} = 5.02$ TeV energy and used them to train our DNN as defined in figure 3.4. The model was trained using 80% of 150K events and was tested and validated by using the remaining events. The model gives pretty good training accuracy to predict the elliptic flow coefficient (v_2) and elliptic flow vector (q_2) for minimum which can be seen from figure 4.2 & 4.3. The model can not only predict v_2 and q_2 for minimum bias events, but can also predict v_2 and q_2 for different centrality classes as shown in fig. 4.4, 4.5. The linear correlation between v_2 and q_2 can also be seen preserved while predicting v_2 and q_2 from fig. 4.6.

Bibliography

[1] R. Sahoo and T. K. Nayak, Curr. Sci. **121**, 1403 (2021).

[2] D. d'Enterria, P. Z. Skands, S. Alekhin, A. Banfi, S. Bethke, J. Blümlein, K. G. Chetyrkin, G. Dissertori, X. Garcia Tormo, i and A. H. Hoang, *et al.*

[3] Available online: `https://home.cern/news/news/experiments/heavy-ion-collisions-where-size-matters` (Accessed on 26 May 2022)

[4] Available online: `https://webhome.phy.duke.edu/~jp401/music_manual/` (Accessed on 26 May 2002)

[5] S. Cao, X. Wang, Reports on Progress in Physics **84**, 024301 (2021)

[6] T. Matsui and H. Satz, Phys. Lett. B **178**, 416 (1986)

[7] T. Niida and Y. Miake, AAPPS Bull. **31**, 12 (2021).

[8] S. Voloshin and Y. Zhang, Z. Phys. C **70**, 665 (1996).

[9] J. Adam *et al.* [ALICE], Phys. Rev. Lett. **116**, 132302 (2016).

[10] N. Mallick, R. Sahoo, S. Tripathy and A. Ortiz, PoS **LHCP2021**, 227 (2021).

[11] A. M. Poskanzer and S. A. Voloshin, Phys. Rev. C **58**, 1671 (1998).

[12] J. Schukraft, A. Timmins and S. A. Voloshin, Phys. Lett. B **719**, 394 (2013).

[13] C. Adler *et al.* [STAR], Phys. Rev. C **66**, 034904 (2002).

[14] N. Mallick, S. Tripathy, A. N. Mishra, S. Deb and R. Sahoo, Phys. Rev. D **103**, 094031 (2021).

[15] P. Xiang, Y. S. Zhao and X. G. Huang, [arXiv:2112.03824 [hep-ph]].

[16] N. Mallick, S. Prasad, A. N. Mishra, R. Sahoo and G. G. Barnaföldi, [arXiv:2203.01246 [hep-ph]] [To appear in Phys. Rev. D (2022)].

[17] L. Lonnblad, C. Peterson and T. Rognvaldsson, Phys. Rev. Lett. **65**, 1321 (1990).

[18] Y. Coadou, EPJ Web Conf. **55**, 02004 (2013).

[19] F. Sergeev, E. Bratkovskaya, I. Kisel and I. Vassiliev, Int. J. Mod. Phys. A **35**, 2043002 (2020).

[20] J. Therhaag, EPJ Web Conf. **55**, 02003 (2013).