UNIVERSITY OF COPENHAGEN

A proposed model to estimate flow coefficients from charged-particle densities using Deep Learning

Zlatko Šaldić

UNIVERSITY OF COPENHAGEN

Name of Department:	Niels Bohr Institute
Name of Faculty:	SCIENCE
Author:	Zlatko Saldic
Email:	zlatko.saldic@nbi.ku.dk
Title:	A proposed model to estimate flow coefficients from charged-particle densities using Deep Learning
Project:	A 45 ECTS points thesis presented for the degree of Master of Science in Physics at the High Energy Heavy-Ion Group
Supervisor:	Prof. Ian G. Bearden
Handed in:	16.07.2020
Defended:	31.07.2020

Name ______________________________

Signature ___________________________

Date _______________________________

Abstract

In the first microseconds of the Big Bang, all known matter in the Universe was in a state of Quark-Gluon Plasma (QGP). The primary purpose of heavy-ion collisions is to investigate QGP's formation and properties. For this purpose, measuring the so-called flow coefficients is an essential observable.

In this thesis, we approach the challenge of estimating the flow coefficients v_n for $n = 1, 2, ..., 5$ from charged-particle densities $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$ using Deep Learning (DL).

For this reason, we construct a controlled experiment by simulating nuclei collisions with a Glauber model. Next, we use the calculations from the Glauber to produce charged particles from various particle production models. Lastly, we distribute the final state particles in the relevant coordinate system (η, φ).

We propose a model based on a Convolutional Neural Network (CNN) capable of analyzing patterns found in charged-particle densities to predict flow coefficients, with confidence intervals quantified by systematic uncertainties. We found that the proposed model can estimate the flow coefficient event by event, with high accuracy and precision for central and semi-central collisions. The performance was evaluated on a test set of 3.000 simulated Pb-Pb events, and yielded a Pearson product-moment correlation coefficient (PPMCC) of 0.42 for directed v_1, 0.90 for elliptical v_2, 0.86 for both triangular v_3 and quintuple v_5, and 0.87 for the quadruple v_4 flow coefficients. The proposed model lays the groundwork for a novel technical approach towards estimating flow coefficients with the possibility of assisting already established methods.

Preface

In the summer of 2018, a somewhat unexpected opportunity resulted in a brief and exciting journey into the world of heavy-ion physics. I had heard of a giant particle accelerator, and this mystical place known as CERN, but mostly from the Dan Brown novel I'll have to admit. Something about a God particle and astonishing pictures of colorful tracks of particle trajectories, and that was it. I had no idea (I still don't, but now, at least, I know all the things that I do not know) whatsoever about heavy nuclei collisions, Quark-Gluon Plasma (QGP), and ALICE, which luckily turned out to be all the exciting stuff. Since then, I have visited CERN multiple times. First time at the ALICE tutorial week in November 2018, next at a GEANT4 beginner course in January 2019, then a beautiful two-day trip to the ALICE detector in February 2019, and finally two times at the ALICE week; first in August and afterward in November 2019.

The idea of this thesis emerged from the last trip. Until then, the focus was to gain experience with the new software framework ALICE O2, a framework necessary to accommodate the Large Hadron Collider upgrade in 2021. In the summer of 2019, I became part of the Fast Interaction Trigger (FIT) software team, where my task was to test the detectors' performance in regards to computation and memory consumption inside ALICE O2. I learned many things during this time, but the satisfaction to discover something more science-related seemed far away.

My original project's long-term goal was to find out how well FIT could measure flow, an observable of QGP, with the established methods. There were multiple challenges with this, one of them being that at the time, even a Monte Carlo event generator capable of simulating flow was not present in ALICE O2. It isn't easy to test a thing's response or phenomena when the thing does not exist. Combined with my admiration of Machine Learning (ML) and its application to multiple problems, I began wondering if ML could be used to investigate flow, independent of software frameworks and the detectors inside of them.

The particles produced from a nuclei collision deposit their energy in the detectors surrounding the interaction point. What if we could measure flow solely by analyzing where the particles have deposited their energy, the so-called final-state? No tracks, no energies, no momenta, no ensemble of multiple events to be analyzed together. Only the image of the final distribution of whatever charged particles. So, I decided to knock on some doors. The knocking thankfully paid off, and this thesis represents the work made from the initial idea to the first results.

Acknowledgements

This thesis would never have seen the light if it were not to some special people. It is obligatory to begin the acknowledgments by thanking the supervisor. I am not going to diverge from that tradition at any cost, so here it is: thank you, Ian, for allowing me to be part of the High Energy Heavy Ion (HEHI) group. The opportunity came with many experiences I wouldn't trade for anything. Thank you for the encouraging words along the way, and for never saying no regardless if it was: taking courses at CERN, requesting hard drives, or thinking AI would solve our problems. Thank you to the guys at the FIT software team, Maciej, Jacek, and Alla. It was a pleasure to work with you even though the work is not presented in this thesis. An exceptional thanks to Christan Holm Christensen, for his guidance, cooperation, ideas, and discussions, especially during the corona crisis. A special thanks to the guys at HEHI: Jens Jørgen Gaardhøje for your support and comments, Børge Svane Nielsen, for giving me and others a guided tour into the core of the ALICE detector (a genuinely magical place), and Christian Bierlich for exciting discussions and helpful remarks. I also would like to thank Freja Thoresen that welcomed me with open arms into the office from the very first day, always prepared to help me out regardless the circumstances. To the rest of the HEHI'ers, thanks for the beers, hangman, coffee, and good (pre-corona) times.

Lastly, a big thank you to my family and friends. In particular my brothers in arms in physics: Aleksandar Topic for our discussion about Neural Networks, Alexander Rathcke for your comments and support, and Henrik Munch Jessen for your proofreading and valuable remarks. I would also like to thank my father and my grandparents for their support during both good and bad times. Finally, my fiance—you made this weird journey so much more enjoyable, thank you.

Zlatko Šaldić
København, July 2020

Contents

1

Introduction

The ultimate goal of experimental particle physics is to detect and analyze particle collisions and their products to enhance our knowledge of the fundamental forces and properties of the Universe. These are glamorous words, but they nonetheless describe the genuine truth, or at least the aim. Particle physics is, like many other areas of science, divided into narrower areas, each focusing on solving their piece of the puzzle. The puzzle, of course, being the Universe and everything in it.

The attention in heavy-ion physics is towards creating a fireball formed in heavy nucleus-nucleus collisions that has the same properties as the Universe had a few moments after the Big Bang. Again, these seem to be glittering words, but maybe reminding ourselves of what we are aiming at is perhaps the most appropriate thing to do when the physics or the tools to describe it become complicated. Not surprisingly, the task is only possible with cooperation among people sharing this goal of solving puzzles. This thesis attempts to take a one-piece of jigsaw puzzle, observe it, find out if and where it fits, and if it doesn't, put it back in the box.

The jigsaw piece is the application of Machine Learning (ML) in the field of high energy heavy-ion physics, with an emphasis on the use of Deep Learning (DL) techniques such as Convolutional Neural Networks (CNN) to Monte Carlo (MC) simulated nucleus-nucleus collisions. The ambition is to see if an DL model can extract five different continuous numerical values, the so-called *flow coefficients* v_n. The flow coefficients are sensitive to the initial conditions of a nucleus-nucleus collision. There is evidence to suggest that the initial geometry of a collision, defined by the so-called *eccentricities* ε_n, is directly proportional to the values of the flow coefficients [1]. It is rather intriguing to investigate the possibility of accurately predicting the initial geometry based on how the created particles from a collision are distributed (also known as the final state) event by event. Already established methods are capable of estimating flow. However, these methods require an ensemble of multiple events and careful selection of particle tracks that minimize effects not related to flow [2].

We, therefore, want to test and train a CNN capable of identifying specific patterns in charged particle densities and associating those with the collision's initial geometry. By constructing a controlled environment of simulated data, we aim at investigating this possibility. There exist multiple toy MC models offering components of simulating everything from nuclei collisions, to charged particle production, and their distribution.

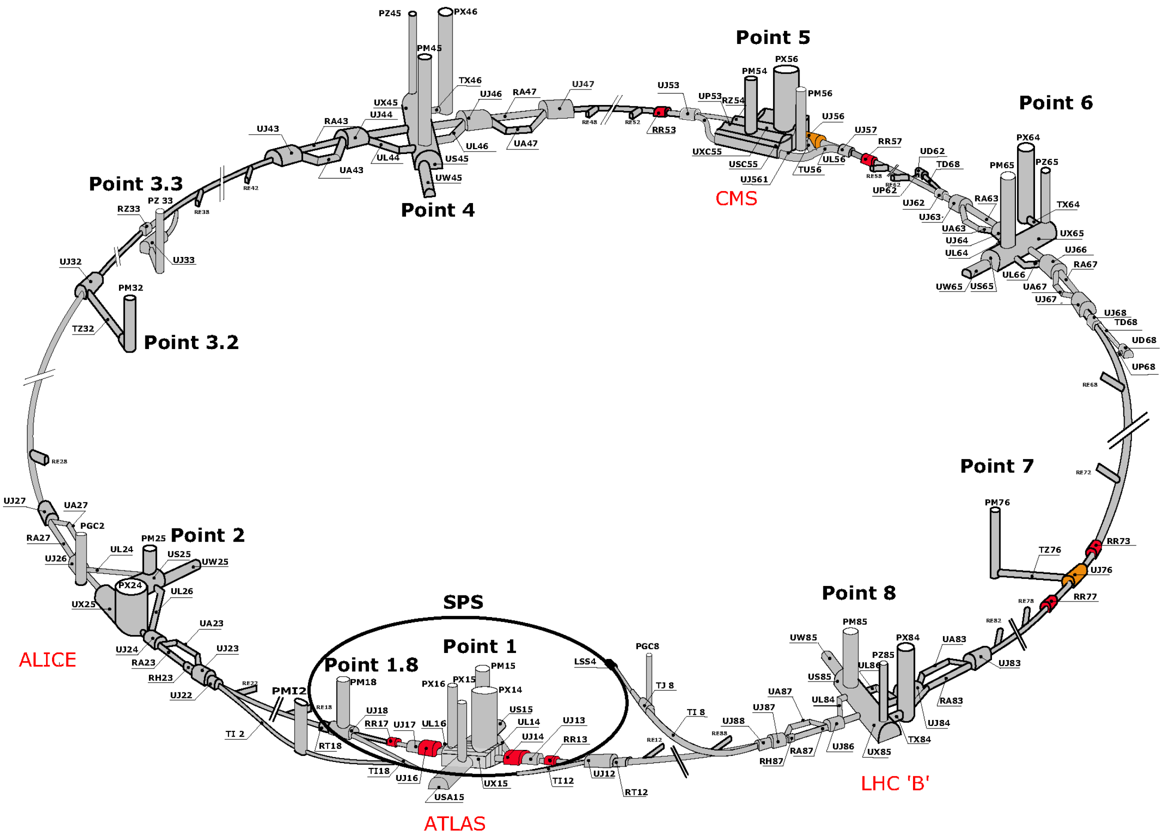


Figure 1.1. Layout of the Large Hadron Collider (LHC) tunnel with its four experiments ALICE, CMS, LHCb, and ATLAS on its circumference © CERN.

Together with state of the art software frameworks offered in the field of ML, we aim to build a proficient and versatile CNN that predicts its targets with uncertainties regardless of the underlying toy MC model.

1.1 Experimental particle physics in practice

The entire foundation of this thesis relies on what we can observe from real experiments. Therefore, it is worth mentioning how we, in practice, collide nuclei with ultra-relativistic speed and detect what happens from the collision.

1.1.1 Collision

A particle accelerator is the basis of experimental particle physics. The largest particle accelerator in the world [3] is located under the French-Swiss border near the town of Geneva. Its name is the Large Hadron Collider (LHC), and its creator is the European center for nuclear research, CERN (*Conseil Européen pour la Recherche Nucléair*). The LHC is 27 km in circumference and consists primarily of a pipe surrounded by superconducting electromagnets in a vacuum tank cooled down to near absolute zero ($0\,\mathrm{K} \approx 273^\circ\mathrm{C}$). Located on its circumference are four experiments—one of them being A Large Ion Experiment (ALICE), see Figure 1.1. The main goal of ALICE is to study heavy-ion collisions. Inside the pipe, particles are traveling as a beam[1] close to the speed of light in opposite directions. The beam covers the distance of the LHC tunnel 11245 times each second. Each beam can have energies up to 5.7 TeV, and a large amount of energy is therefore released during a collision. The arrangement of the pipe is such that the particles will collide in the center of an experiment. Squeezing (caused by quadrupole magnets) the particles together despite their size and speed and making them collide with such precision is no less than an engineering masterpiece. Besides protons, heavy nuclei, such as lead (Pb) are also collided in the LHC. Lead atoms have all of their electrons stripped away to become fully ionized (hence the term heavy-ion) so that they can be accelerated and kept in place by electric (radiofrequency (RF) cavities) and magnetic (electromagnetic diploes) fields, respectively.

1.1.2 Detection

The violent collision of heavy-ions causes the available energy to produce numerous new particles and radiation. Most of them quickly decay to other forms of matter via fundamental forces—Gamma-radiation, e.g., can decay into electron-positron pairs (the reverse of annihilation).

Particle detectors seek to describe what happened, what was produced, and how. There are designate detectors that determine the exact position and time of the interaction point, and the actual number of produced particles is also known as the *multiplicity*. Other detectors focus on distinguishing between leptons (by electromagnetic calorimeters) and hadrons (by hadronic calorimeters), and seperate detectors focus on particle trajectories (like the Time Projection Chamber (TPC) in ALICE) from the interaction point where it was created. An experiment can, therefore, consist of many sub-detectors constructed in layers, which work together in tandem to give a clear picture of what happened.

There are many methods of detecting particles, but common to all of them is that they involve some variety of energy deposition in a material—the particles have to leave a mark and interact with the detector following the same laws that it tries to uncover. For this reason, there are constraints to what is measurable. How can we measure a particle that neither absorbs nor emits energy through light, as it is the case of Dark Matter? If a particle cannot interact through the electromagnetic force in some way or another, it is not detectable with current technology, and we are in deep trouble of ever measuring it. *Charged particles* are the only thing measurable in experiments[2]; fortunately, there are many ways a charged particle can deposit its energy. A large particle detector is composed of many layers, as mentioned, and each with properties that can distinguish particles or other relevant information.

For example, a scintillation counter is a detector that absorbs charged particles. An electron of a molecule in the scintillator gets excited to a higher energy, and then emits a photon when returning to the ground state. The scintillating material is also

[1] A beam can consist of $\sim 10^{14}$ protons.

[2] Not to be confused with that we can only detect particles with charge—instead we refer to that we are only capable of detecting particles interacting via the electromagnetic force.

attached to a photocathode. When the emitted photon strikes the photocathode, it gets converted to an electron as prescribed by the photoelectric effect. The photocathode is coupled to photomultiplier tubes (PMT), whose purpose it is to enhance the small energy of the electron to an electric pulse. This electric pulse is what is measured from the electronics of the detector. Therefore, by studying the pulse, experimenters can infer the number of electrons originating from the photocathode. Knowing the number of electrons allows us to determine the energy of the photons hitting the photocathode. The energy of the photon is directly related to the energy of the charged particle hitting the detector. Scintillation detectors can also be useful in estimating the centrality (how much the colliding nuclei overlap) in the forward region of the collision, which is an essential parameter, as we will later see.

Another way to identify charged particles could be through a detector that utilizes the Cherenkov effect. A Cherenkov detector consists of a dielectric medium[3] that gets polarized when a charged particle transverses it. The charged particle (a charged pion, for example) has to travel faster than light would in that medium. When it does, a cone of light appears whose shape (angle) depends on the incoming particle's velocity. If there are two Cerenkov detectors on each side of the interaction point, the collision time can be measured very precisely (down to a few picoseconds).

1.1.3 Future of ALICE

These types of detectors are the cornerstones of the new so-called forward and trigger detectors used in ALICE [5]. As well as the rest of LHC, ALICE undergoes a significant upgrade in the years of 2019-2021 [6]. The upgrade of LHC aims to increment the number of possible collisions—this leads to more data and, thus, higher statistics that should lead to discoveries. The new forward and trigger detector system for the upgraded ALICE is known as the Fast Interaction Trigger (FIT) [7] and is a system of sub-detectors that include the principles of scintillation and Cherenkov radiation. The goal is to have a fast and continuous readout for event selection (triggering), which is the primary purpose of FIT. Event selection is essential in order to characterize the collisions, but also to identify events worthwhile recording.

The accelerated progress of Deep Learning (DL) in computer vision and image processing has also found its way to the LHC community [8], where the application of DL for jet classification [9] and event selection [10] is currently being developed. Integrating DL with known methods may perhaps constrain properties of both collision parameters and collision analysis otherwise not achievable when studied separately.

[3] An example of a suitable Cherenkov medium is the quartz crystal. Other possibilities include water in solid form. The IceCube experiment [4] uses ice on the South Pole to measure neutrinos. The neutrino can interact (very rarely) with atoms and produce charged leptons which then emit Cherenkov radiation in the ice.

1.2 Outline

The outline of this thesis is as follows: we begin by considering the problem at hand; why even bother spending time on ultra-relativistic nuclear collisions? The background chapter will briefly cover the Big Bang the Standard Model and in particular, the interaction of the strong force and its consequences. Next, it is necessary to create a controlled experiment. The experiment is the simulation of particle collisions. For this purpose, we introduce the Glauber model for particle collisions. After that, we show how we distribute the created charged particles N_{ch} from the collision to their final state densities $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$ by using the calculations from Glauber. Chapter 4 is devoted to ML principles and an introduction to the main concepts of neural networks.

The main novel contribution of this thesis is to estimate the flow coefficients v_n event by event, entirely from the final state densities of charged particles $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$. On these grounds, we propose a CNN and address the considerations made in constructing it. Lastly, we evaluate the model's performance on 10.000 events and examine the systematic uncertainties.

2

Background

This chapter briefly summarizes the concepts needed to recognize the importance of studying nucleus-nucleus collisions. We begin with the Big Bang and the Standard Model of particle physics, where we consider their relation to the cosmic evolution. We close off by describing Quark-Gluon plasma in the early Universe and how it can be studied in experiments by measuring flow.

2.1 Big Bang

The Big Bang theory is reasonably new in the scientific world. It was ridiculed by the European physics communities in the 1920s when it was suggested the first time by priest and professor in physics Georges Lemaître. Nevertheless, observational data eventually changed their minds. The first data taken originates from an observatory on the mountaintop of Mt. Wilson, where now-famous Edward Hubble observed some strange behavior in the motion of galaxies [11]. He knew the distance r to the nearest galaxies because of the newfound application of standard candles. When he analyzed their spectral lines, he simultaneously found their velocity v. The conclusion was that most of the galaxies had redshifted spectral lines and were thus receding from Earth. What is more, the further the galaxy is away, the higher its redshift and thereby its velocity. This linear relation $v = Hr$, is now known as Hubble's law and implies that our Universe is expanding. Hubble was not very precise in his measurement of this linear constant H (now named the Hubble constant). However, today we know that $H = 70\,\mathrm{km/s/Mpc}$[1], which means that galaxies that are $\sim 100\,\mathrm{Mpc}$ away (like galaxies in the Coma cluster) will recede from us with a velocity of $v = 7000\,\mathrm{km/s}$. Hubble's law is the first observational evidence of the Big Bang model and since then followed multiple other compelling evidence such as the total abundance of Hydrogen and Helium in the Universe and the Cosmic Microwave Background (CMB) [12].

2.2 Standard Model

The Standard Model (SM) of particle physics is the currently most accepted scientific theory of the structure of matter and forces in Nature. It is a description of all elementary particles and fundamental forces. There are two distinct groups of matter particles

[1] $1\,\mathrm{Mpc} = 1\,\mathrm{Mega\,Parsec}$, where $1\,\mathrm{Parsec} \sim 3 \cdot 10^{13}\,\mathrm{km}$.

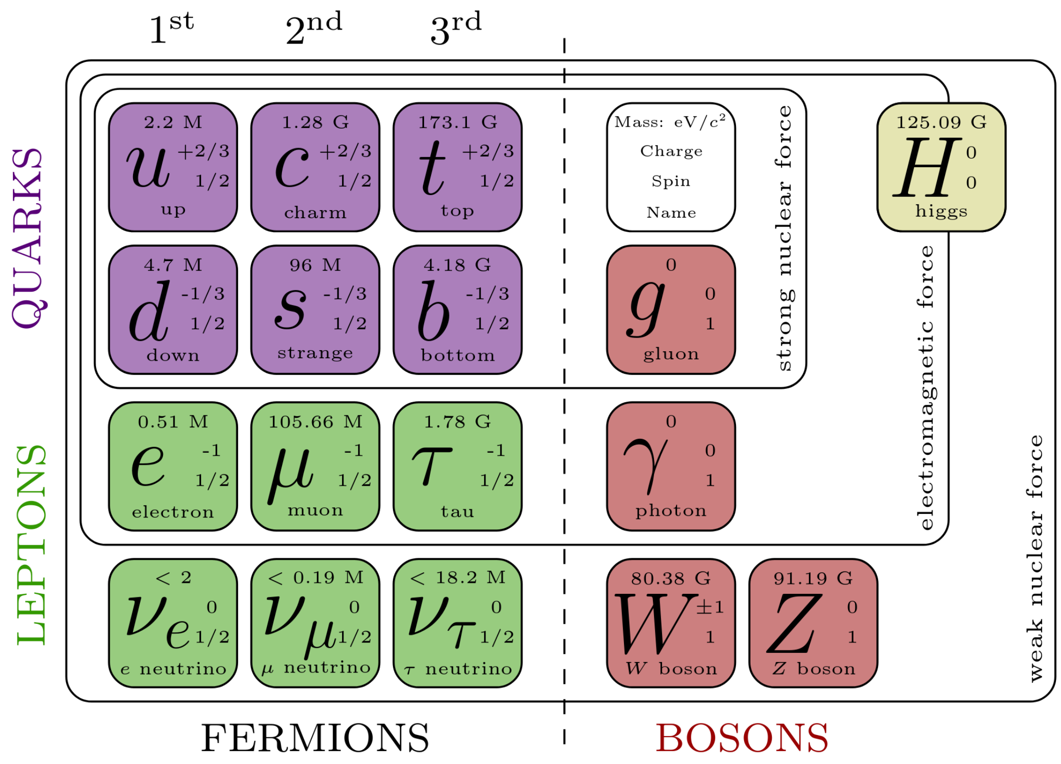


Figure 2.1. The Standard Model of elementary particles and fundamental forces [14].

(*fermions*)—a group of *quarks*, and a group of *leptons*. The distinction between quarks and leptons is by how they react to the fundamental forces. In SM, the *bosons* manifest the fundamental forces by exchanging themselves among the fermionic particles. Photons are the exchange particles of the electromagnetic force, which couples to electric charge. Likewise, the gluon is the exchange particle of the strong nuclear force and couples to color charge (red, green, and blue). The weak nuclear force enables the transformation of quarks (observed as decay), and the force carriers are the W and Z bosons. The particles of the SM can be seen together in Figure 2.1. The forces are fundamental, meaning that, in the framework of the SM, nothing else is taking place except for those forces at smaller scales (presumably). To emphasize what this means, when anything happens it boils down to one of four things: attraction, repulsion, decay, or annihilation. How exactly and why some processes happen, and others not, is mainly governed by conservation laws [13].

Although the Standard Model is a compelling theory, some of the phenomena recently observed, such as Dark Matter (DM) and the absence of antimatter in the Universe, cannot be accounted for in the model. Likewise, incorporating gravity into SM is also missing; therefore, much work still lies ahead for the scientific community to reveal the mysteries of our world.

2.3 Cosmic Evolution

It is possible to join the findings from particle physics and cosmology to study the Universe as a whole. Cosmology is the scientific field of the origin and evolution of the Universe described by the theory of General Relativity (GR). Alexander Friedmann was able to simplify Einstein equations GR to a relation between the expansion rate of the Universe and the energy density within it. If the Universe expands a tiny bit more tomorrow, it implies that it was a tiny bit smaller yesterday. The average temperature of the Universe today is $\sim 2.7\,\mathrm{K}$ as measured from the CMB. However, when the CMB photons decoupled[2] after the formation of atoms, they had a temperature of $\sim 3000\,\mathrm{K}$. This last scattering finalized the epoch of *recombination*. It is possible to characterize different epochs in the early Universe, since particular interactions only take place at distinct energies or temperatures, see Figure 2.2. The justification to why it is possible to relate temperature with interaction energies is due to observational data suggesting that the CMB is a *perfect blackbody* [15]. Given that the CMB is a perfect blackbody implies that the Universe at that time was in thermal equilibrium. The main property of an object in thermal equilibrium is that the temperature is equal for all of its constituents. Only frequent interactions among the constituents can keep the object remained in thermal equilibrium.

An interaction only happens with a particular probability which can be described by a *cross-section* σ. The mean free path $\lambda = \frac{1}{n\sigma}$ is the length which, e.g., a photon covers before it interacts, where n is the number density of the interacting particle (electrons for example). The *interaction rate* Γ is then the frequency of interactions. Dimensional analysis implies that the interaction rate is simply the photon's velocity divided by the cross-section of some interaction involving the photon $\Gamma = \frac{c}{\lambda} = n\sigma c$. Interactions are only possible if the interaction rate is larger than the expansion rate H (the Hubble constant) of the Universe, or if the mean free path is smaller than the *Hubble size*, defined as c/H. The reason is that, when the Universe expands the temperature drops, and the necessary energy required to accommodate particular interactions is not present. Because photons interact so frequently with particles until the epoch of recombination, all components of the Universe are in thermal equilibrium. Therefore the energy available for particular interactions is the same as the temperature of the photons. Hence, one can trace back the cosmic evolution to investigate what properties the Universe had shortly after the Big Bang. It suggests that the Universe had a temperature of around 200 MeV some $20\,\mu$s after the Big Bang.

What was the state and structure of matter at this point? What happens to the fundamental forces and elementary particles, or in particular the strong force and color charge particles at these temperatures and energies? With this question, we enter the realm of heavy-ion physics.

[2] Decoupling hinges on the sudden seize of additional interactions.

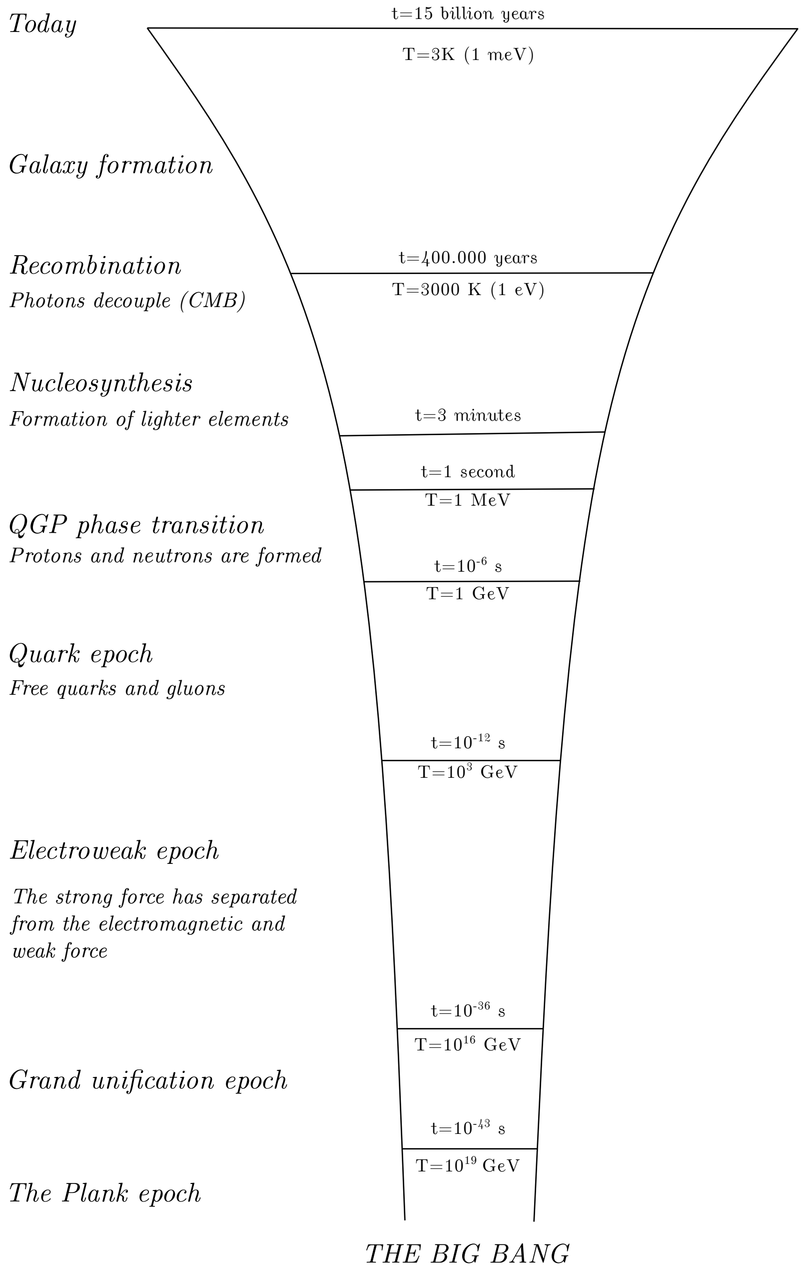


Figure 2.2. The history of the Universe with temperatures and timescales corresponding to the various epochs and major events.

2.4 Quark-Gluon Plasma

The state of matter a few microseconds after the Big Bang was a Quark-Gluon Plasma (QGP). During the quark epoch, between 1 ps and 1μs after the Big Bang, the gluons and quarks were free to move. When the temperature dropped below a critical value, T_c, the free quarks and gluons underwent a phase transition from a fluid to a gas of hadronic bound color states. Multiple experimental observables from nucleus-nucleus collisions suggests that the QGP has the properties of a *perfect fluid*[3] [16]. To understand why QGPs occur, we need to dive into the strong force.

2.4.1 The Strong Force

If the distance increases between two electrical charges, the potential drops as the inverse of that distance. Physicists have great familiarity with the inverse square law of both electromagnetism and gravity. The strong force behaves quite unlike those two forces. If the distance increases between two color charges, the potential between them *rises*. The reason is that gluons themselves carry color charge. Unlike photons, gluons can interact with themselves, and this means that the color fields surrounding color charges gain strength with increasing distance. The consequence of this is that quarks are always bound in color-neutral states called hadrons. No one has ever found a free quark or gluon, and if you try to separate quarks inside a nucleon, the energy of a strong field rises enough to materialize a new quark-antiquark pair. This concept is also known as *color confinement*.

An analogy often used to illustrate the strong force involves a rubber band. You have a rubber band, and you stretch it as much as possible between your hands. Provided you are reasonably strong, and the rubber band you use is small, the tension you create will eventually break the rubber band into two pieces. You can know start all over again with the two rubber pieces only to still find yourself with proportionally many rubber pieces (and a mess). Conversely, if you take a rubber band between your hands and shorten the distance, the tension disappears completely. This analogy is useful to introduce the concept of *asymptotic freedom*. QGP is a direct consequence of asymptotic freedom. The properties of the strong force makes it a short-range force which only works on small length scales (1 fm or 10^{-15} m). If the energy scale rises, or conversely the length scale decreases, asymptotic freedom is attainable, and a QGP forms. This idea wraps up the begging of this section, namely, that the Universe at the begging had a size and temperature that corresponded to the scales of asymptotic freedom resulting in QGP. Correspondingly, it is possible to attain the energy density necessary in nucleus-nucleus collisions to "melt" the nuclei. With this knowledge in mind, we can now focus our attention on the colliding nuclei.

2.5 Flow

An important observable of QGP is the so-called phenomenon of *flow*. When nuclei collide, the energy density and temperature are high enough for the constituents (quarks

[3] This concept relates to how the shear viscosity over entropy density ratio $\eta/s \approx 1/4\pi$.

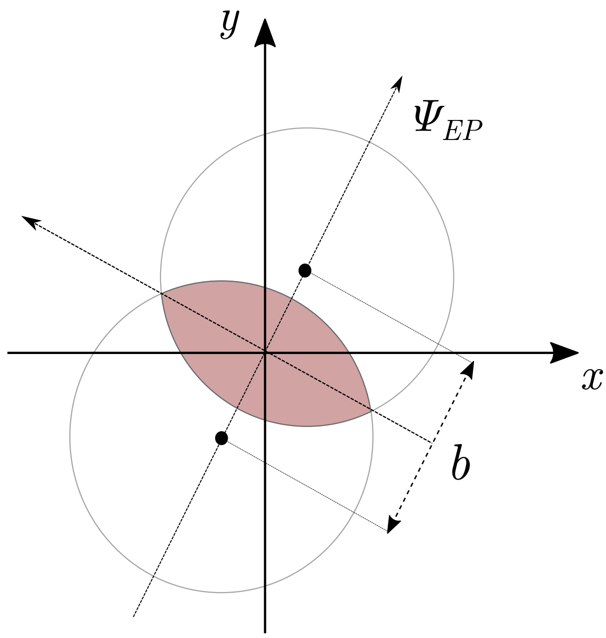


Figure 2.3. Two overlapping nuclei in the (x, y) plane. Ψ_{EP} is the angle of the *Event Plane* relative to the x-axis, and is spanned by the *impact parameter* b. The impact parameter characterizes the distance between the centers of the colliding nuclei.

and gluons) to no longer be bound in the nucleons—they become asymptotically free and form a fireball of QGP. If the collision is peripheral, the overlapping area of the colliding nuclei can resemble an American football, see Figure 2.3. The fireball will seek to return to equilibrium with its surroundings and therefore expands to cool off. If the QGP is a fluid, then the mean free path is small, and the constituents all couple to each other. Because of its American football shape, and its attempt to restore equilibrium, the pressure needs to be much stronger along the side of the football. The energy from the higher pressure is transferred to the individual particles during the hadronization phase, where the plasma decouples to a hadron gas. Therefore, the geometry of the initial collisions directly influences how the charged particles are distributed in the detectors. We will later see how to generate multiplicity distributions resembling real-world experiments from the initial conditions and thereby simulate the effect of flow.

2.5.1 Flow coefficients

The study of particle distributions in the azimuthal angle can be done with Fourier analysis [17]. Fourier analysis involves decomposing a function in periodic parts, and in the case of the azimuthal charged particle distribution it is given by

$$\frac{\mathrm{d}N_{\mathrm{ch}}}{\mathrm{d}\varphi} \sim f(\varphi) = 1 + \sum_{n=1}^{5} v_n \cos\left(n\left[\varphi + \Psi_n\right]\right) \quad , \tag{2.1}$$

where N_{ch} is the number of charged particles, v_n are the *flow coefficients*, n the harmonic order, and Ψ_n is the harmonic plane (the orientation of the specific flow coefficient). All

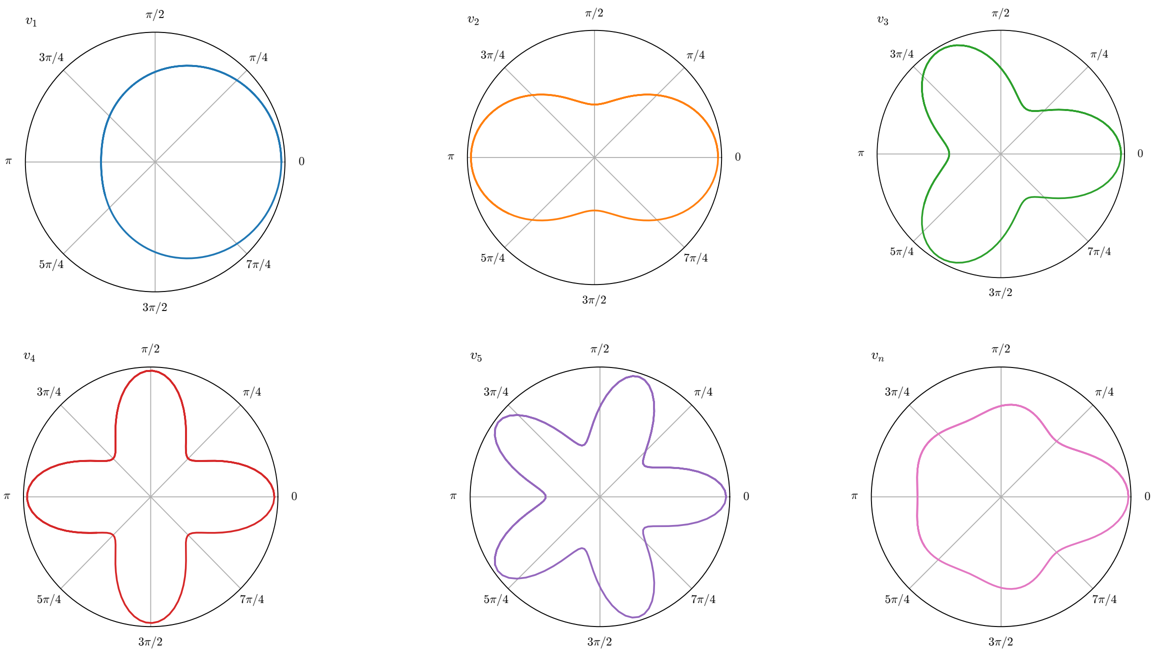


Figure 2.4. A depiction of the flow coefficients v_1, v_2, v_3, v_4, v_5 and the combined v_n.

of the flow coefficients represent a separate structure of the Quark-Gluon Plasma, see Figure 5.2. For every harmonic n there is a distinct nature to it, e.g., for $n = 1$ we have v_1 which is labeled as *directed flow*, whereas v_2 is *elliptical flow* , v_3 *triangular flow*, v_4 *quadruple flow*, and v_5 *quintuple flow* .

There is reason to suggest that the flow coefficients are proportionally dependent on the collision's initial state or geometry [18]. Knowing the initial condition thus helps in further investigation of the QGP properties. These properties include not only the critical temperature of the phase transition but also the very nature of going from the unbound liquid state to a confined gas state. In more advanced simulations, various so-called hydrodynamic models use the initial conditions in great detail for the transportation and evolution of the QGP. Therefore, measuring flow coefficients is a crucial step toward understanding the early Universe and the strong force. In the next chapter, we will quantify the initial geometry, which is labeled by the eccentricity ε_n, and we will distribute the particles as in Eq. **??** by scaling the flow coefficients linearly from the eccentricities $v_n \propto \varepsilon_n$.

3

Experiment

This chapter describes the steps made to simulate the phenomena of flow, an observable of the Quark-Gluon Plasma in nucleus-nucleus collisions. We will collide nuclei together and replicate the distributions seen from real experiments of the charged particles formed from the collision. We begin by building the atomic nuclei from scratch by placing its nucleons using a nuclear density distribution. Afterwards, we create the Glauber model and produce collisions following an interaction and cross-section profile. We finish off by using the calculations of the initial conditions from the Glauber model to generate the charged particle multiplicity distributions.

An advantage of computer experiments is the ability to have full control of the underlying calculations and assumptions. What is more, we can use simulations to model things that are not directly observable in a real-world experiment. Nucleus-nucleus collisions are an example of such an experiment. An atomic nucleus is a complex object that consists of nucleons (protons and neutrons), all having a size of about $1\,\mathrm{fm} = 10^{-15}\,\mathrm{m}$. Because nucleons have such small sizes, it is impossible to observe the collisions between them directly. Additionally, the collision of nuclei and formation of QGP is unfortunately very brief, corresponding to only a few fm/c, i.e., $\sim 0.00000000000000000000001\,\mathrm{s} = 10^{-23}\,\mathrm{s}$. Therefore, no camera or detector in the world can directly capture what exactly happens and how.

The only things which are measurable are the charged particles. If the nuclei collide head-on, both of them will substantially overlap, and the number of independent nucleons participating in the collision N_{part} is high. We can parameterize the overlap with the impact parameter b, which is the distance between the centers of the two nuclei. In this manner, two (Pb) lead-ions colliding head-on, i.e. with $b = 0$, each having $A = 208$ nucleons can, in principle, both have $N_{\mathrm{part}} = 2 \times 208 = 416$ participants. We term such collisions as *central.* A typical Pb has a size of about $\sim 14\,\mathrm{fm}$, and this puts constraints on the impact parameter. If the impact parameter is large, all the way up to $b \approx 14\,\mathrm{fm}$, the two nuclei will graze slightly, and the collision is said to be *peripheral.* Consequently, the participating nucleons in a Pb-Pb collision can range from just $N_{\mathrm{part}} = 2$ (a single proton-proton collision) to the maximum possible, $N_{\mathrm{part}} = 416$.

The nucleons are small and well separated inside the nucleus, and a head-on (elastic) collision between two nucleons is rare. The most common types of collisions are the inelastic ones. We will not distinguish between the two in our model. Instead, we will characterize the probability of a collision as the nucleus-nucleus cross-section σ_{NN}[1] and define an interaction profile that quantifies how we apply the cross-section.

Furthermore, the nucleus is an extended three-dimensional object, but the collision only happens in the transverse plane (x, y) relative to the axis of the collision z. The number of collisions can exceed the number of participating N_{part} nucleons—the approximation is that the nucleons do not deflect or otherwise change direction after they have collided the first time. This arrangement will allow participating nucleons to interact with more than one nucleon in the other nucleus. It makes the nucleus-nucleus collision more energetic, resulting in many more particles created. We label this as N_{coll} or the number of *binary collisions*. The impact parameter, the number of participants N_{part}, and the number of binary collisions N_{coll} are all parameters that are not directly observable in particle collision experiments. However, they all define the initial conditions critical for the formation of the QGP and its observables. We have to rely on assumptions and models that can be simulated on a computer. An example of such a model is the Glauber model [19]. The Glauber model is one of many methods to simulate particle collisions. The main reason for choosing the Glauber model is because the geometry and initial conditions of the collision are essential for how QGPs evolve. We will build the Glauber model from the ground up, carefully placing nucleons inside the nuclei, sample a random impact parameter, and collide the nuclei with some interaction profile defined by a cross-section. By building a model, we can calculate the initial state geometry defined as the eccentricities ε_n.

3.1 Glauber Model

The initial conditions of nucleus-nucleus collisions directly dictate the production of charged particles and the observables of QGP. We attempt to compute these initial conditions using the Glauber model, i.e., to summarize from preceding sections, we aim at defining the following:

- Two nuclei A and B built from a nucleon distribution
- An interaction profile given by a cross-section σ_{NN}
- A random impact parameter b

From then on, we intend to calculate:

- The number of participating nucleons N_{part}
- The number of binary collisions N_{coll}
- The eccentricity ε (geometry of the collision)

[1] It is customary for a particle physicist to quantify cross-section in milibarns $1\,\mathrm{mb} = 10^{-31}\mathrm{m}^2$.

The bullet points all contribute to what we will encapsulate as the Glauber model in this thesis[2].

3.1.1 Building the nuclei

The first step before colliding is to build the atomic nucleus. The nucleons are not uniformly distributed in the nucleus but instead arranged in a way that can be modeled by a nuclear charge density distribution (or nucleon distribution) which is defined by a probability density function (PDF). This arrangement means that each coordinate of a nucleon

$$(x, y, z) \quad ,$$

is distributed according to some probability.In practice, this implies that we draw a number

$$r = \sqrt{x^2 + y^2 + z^2} \tag{3.1}$$

which characterizes the distance from the center of the nucleus, and then we place the nucleon by adding a random polar θ and azimuthal angle ϕ, respectively. Thereby,

$$\begin{aligned} x &= r\cos\phi\sin\theta \\ y &= r\sin\phi\sin\theta \\ z &= r\cos\theta \end{aligned}$$

The polar θ and azimuthal ϕ angles are both drawn from a uniform distribution

$$\begin{aligned} \phi &\sim \mathcal{U}[0, 2\pi] \\ \theta &\sim \mathcal{U}[0, \pi] \quad . \end{aligned}$$

The method of sampling nucleons in this manner does not ensure that the nucleons do not overlap in the nucleus. We will manage this unphysical technicality by checking for overlaps and redistributing nucleons until no overlaps are present. We additionally also recenter the nucleons such that the center-of-mass is at zero in the (local) rest frame of the nucleus. This recentering ensures that we can collide the nuclei properly, later on, when we have to pass a random impact parameter in the process.

3.1.1.1 Nucleon distribution

There are many variations of a nucleon distribution, depending on the type of nucleus we are interested in building. In the appendix, we show different types of density distributions. In this thesis, we will be focusing on colliding (Pb) lead-ions. The placement of nucleons inside Pb (which is approximately spherical) can follow the three-parameter Eerhart-Fermi charge distribution [21]

$$f(r; W, R, A) = r^2 \frac{1 + W\left(\frac{r}{R}\right)^2}{1 + e^{\frac{r-R}{A}}} \quad ,$$

[2] As part of this thesis project, the Glauber model's implementation happened in cooperation with Associate Professor Christian Holm Christensen. It works now as a learning tool at the Niels Bohr Institute, University of Copenhagen. The full source code with documentation is available online [20].

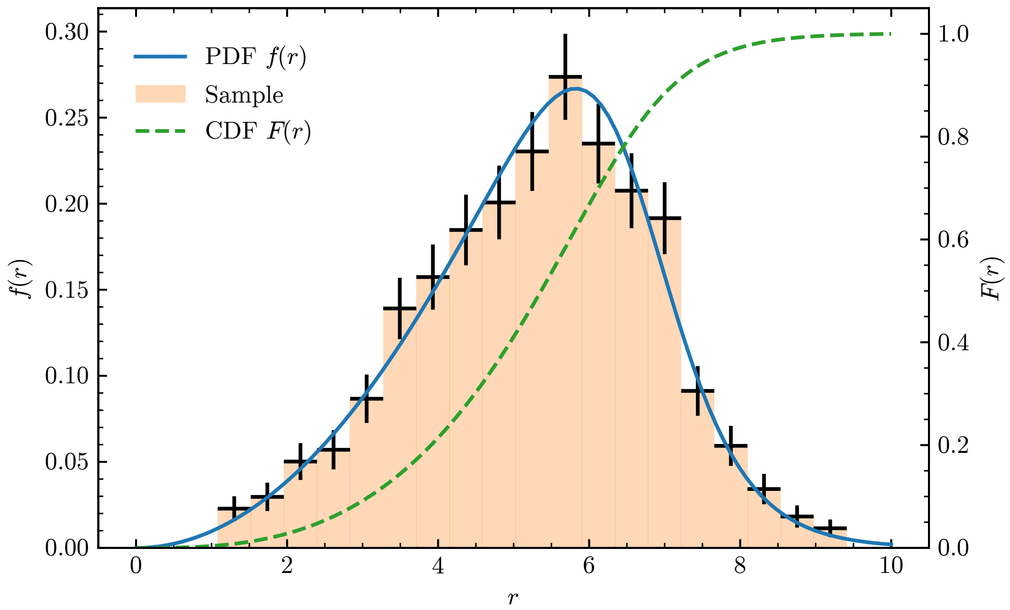


Figure 3.1. Nucleons sampled from the Eerhart-Fermi charge distribution. Here we are sampling 208 nucleons of a Pb nucleus with parameters $R = 6.624$, $A = 0.549$, and $W = 0$.

where r is the radius, W is the suppression near $r = 0$ (or deviation from a spherical shape), R is the half-density radius, and A is the skin parameter[3]. We sample nucleons from this probability distribution using the inverse transformation Monte Carlo (MC) numerically, see Figure 3.1.

3.1.2 Colliding nuclei

Now that we have a method for creating nuclei, we proceed with the practice of colliding them. In order to collide the nuclei, we first need to define an interaction profile and a cross-section. When we have these in place, we can continue with the Glauber implementation by picking a random impact parameter and collide the nuclei together.

3.1.2.1 Interaction profile

We can define an interaction profile as the PDF that, when evaluated with a particular probability (cross-section) and (squared) distance between two nucleons, determines whether the collision happens or not. The most common interaction profile applied in a typical Glauber model is the *black-disc* model, which is a simple step function that defines an interaction if the squared distance between two opposing nucleons is smaller than a given cross-section

[3]Heavy nuclei often have a more neutron rich outer layer referred to as the "neutron skin".

$$f(r^2;\sigma_{\mathrm{NN}}) = \begin{cases} 1 & r^2 < \frac{\sigma_{\mathrm{NN}}}{10\pi} \\ 0 & \text{otherwise} \end{cases} \quad . \tag{3.2}$$

Here $1 = \text{True} = \text{Interaction}$, and $0 = \text{False} = \text{No-interaction}$, and we divide with factor 10 since $1\,\mathrm{fm}^2 = 10\,\mathrm{mb}$. A PDF-like interaction profile could instead be the Gaussian (normal) distribution

$$f(r^2;\sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\frac{r^2}{\sigma^2}}, \quad \text{with} \quad \sigma^2 = \frac{\sigma_{\mathrm{NN}}}{20\pi} \quad . \tag{3.3}$$

For these types of smooth interaction profiles, the procedure that determines whether an interaction happens or not involves the following: given some distance between two nucleons, we can compute the probability of an interaction by evaluating the PDF with this distance (like the black-disc). We then draw a random number from a uniform distribution between zero and one and check whether this probability is smaller than the one evaluated with the PDF. If the probability is indeed smaller, then we have an interaction.

We also have other possibilities, such as the so-called *defuse model* [22]

$$f(r^2;\omega,R) = \left[1 - \Gamma\left(\frac{1}{\omega}, \frac{r^2}{R^2\omega}\right)\right], \text{ with } \omega = \frac{\delta^2\sigma_{\mathrm{NN}}}{\sigma_{\mathrm{NN}}^2} \text{ and } R^2 = \frac{\sigma_{\mathrm{NN}}}{10\pi} \quad . \tag{3.4}$$

Here $\Gamma(a,x)$ is the lower incomplete Γ-function given by

$$\Gamma(a,x) = \frac{1}{\Gamma(a)} \int_0^x dt\, e^{-t} t^{a-1} \quad . \tag{3.5}$$

The parameter ω is a measure of the relative fluctuation in the nucleon-nucleon cross-section as prescribed by the fluctuation parameter $\delta\sigma_{\mathrm{NN}}$. Having an $\omega \to 0$ corresponds to a black-disc, while $\omega \to 1$ coincides to a normal distribution. The interaction probability of the different profiles can be viewed in Figure 3.2. We will mainly apply the black-disc model in this thesis, but we will use other interaction profiles to constrain the systematic errors in the proposed Deep Learning model.

3.1.2.2 Cross-section

We introduced a more smooth interaction profile that allows interactions when nucleons are far from each other with some (small) probability. This idea relates to a cross-section that can vary around a mean in a relatively small range, e.g., $70 \pm 5\,\mathrm{mb}$. However, we can imagine other ways that the cross-section varies from collision to collision, or even from nucleon-nucleon pair within a collision.

One theoretical treatment of a fluctuating cross-section is the so-called *Glauber-Gribov* picture. The proposal is that the color state of each nucleon determines the interaction probability of a nucleon-nucleon pairs. Highly relativistic nucleons can have

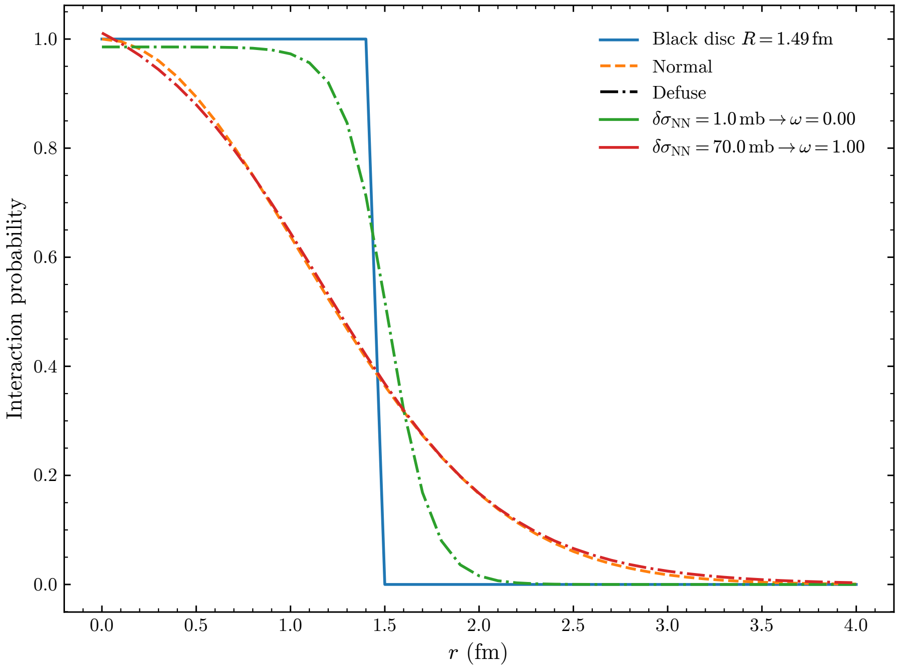


Figure 3.2. Variations of different interaction profiles that can be implemented in the Glauber model. The implementation of the defuse model has been simplified to reduce computations in practice—details are available in the source code.

fluctuating color states, and possibly interact with all nucleons (the extreme case) in the opposing nuclei regardless of the impact parameter. In any case, introducing fluctuations in cross-section affects the changes in the number of collisions to be significantly larger than for a fixed cross-section. The fluctuations can, therefore, also lead to rather dramatic or unusual events. We will consider two variations of these types of fluctuating cross-sections. The different types can be seen in Figure 3.4. In one type we assume that the individual color state of each nucleon is fixed during the nucleus-nucleus collision. The assumption is that the collision's duration is too short for the nucleons to fluctuate in the time of the collision, i.e., all nucleons have the same fluctuating cross-section at collision time. The other type allows for fluctuations between individual nucleons, i.e., the nucleons in a nucleus can all have separate cross-sections. The Gribov PDF [23] gives the fluctuating cross-section

$$f(x;\sigma_{NN},A,\delta\sigma_{NN}) = \frac{1}{A}\frac{x}{x+\sigma_{NN}}e^{-\frac{(x/\sigma_{NN}-1)^2}{\delta^2\sigma_{NN}}}\;,\;\text{with } x=\sigma/A \quad . \tag{3.6}$$

Here $\delta\sigma_{NN}$ is the uncertainty on σ_{NN}, A is the normalisation, and σ is the fluctuating nucleon-nucleon cross-section. A sample of the PDF can be seen in Figure 3.3

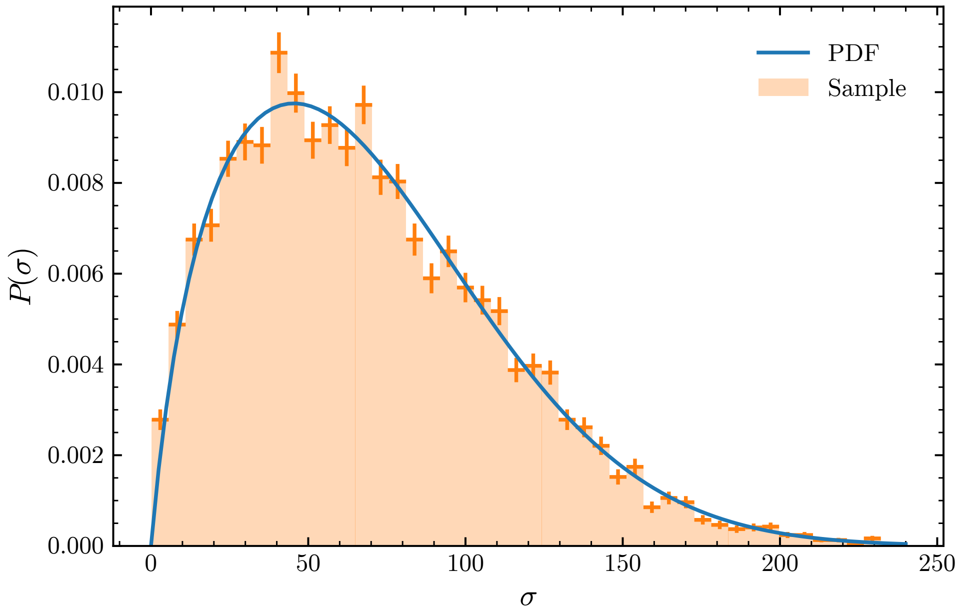


Figure 3.3. The Gribov PDF describing the fluctuation of cross-section.

3.1.2.3 Collision

We now have almost all the pieces in place to perform collisions. The only thing missing is to sample an impact parameter (the distance between the centers of the two colliding nuclei). We sample the impact parameter from a uniform distribution times the squared maximum impact parameter, since the interaction probability is proportional, as shown in Eq. 3.2, to the overlap area

$$b \sim \sqrt{\mathcal{U}[0, b_{\text{max}}^2]\,\text{fm}^2} \quad .$$

The impact parameter, cross-section, and interaction profile are the variables that together influence the characteristics of an event. When we know the impact parameter, we can directly compute the distance from Eq. 3.1 between all nucleons in both nuclei. Which nucleons collide is based on a cross-section, an interaction profile, and a probability (if any). Besides the specific nucleons and their coordinates, we also compute the number of binary collisions each nucleon partook in, see Figure 3.5.

Later we will see how the number of participants and the number of binary collisions influence the charge multiplicity distributions when they get combined to a single *source* of particle creation (the so-called number of *ancestors* N_{anc}). Next, we perform calculations based on the colliding nucleons' coordinates to compute the initial geometry of the collision.

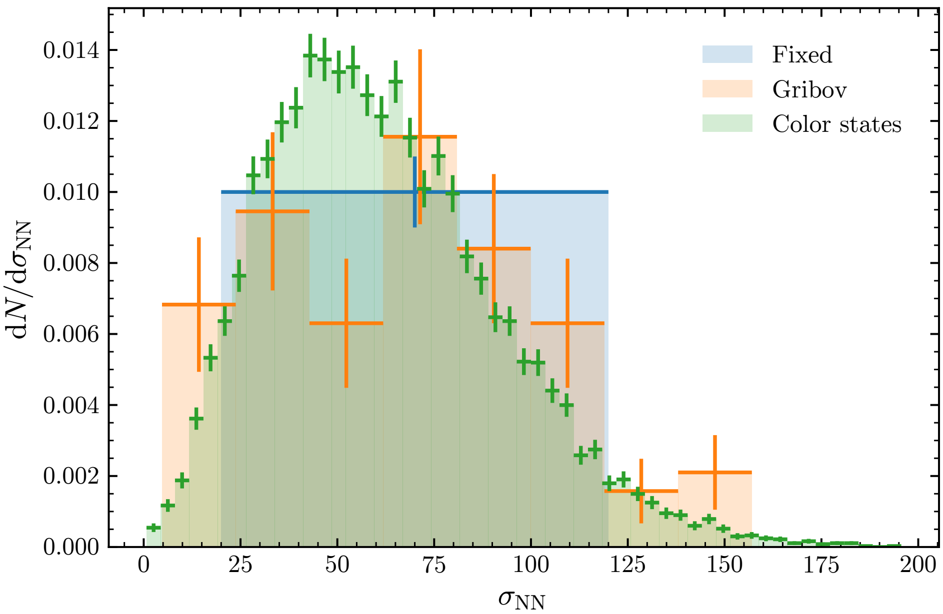


Figure 3.4. Different variations of cross-section models. The histograms display the *profile* or probability density function of a particular cross-section. We can imagine the cross-section fluctuates from collision to collision, and these profiles give the probability of an interaction given the specific cross-section. For a fixed cross-section (blue) we have chosen a large uncertainty (± 50 mb). We see that the probability of each cross-section is equally likely, but outside this range the probability is zero. For the color-state fluctuation (green), the probability depends on each nucleon. It is, therefore, finer-grained, whereas the Gribov-Glauber picture (orange) constrains the cross-section for all nucleons in the same collision.

3.2 Eccentricity

The eccentricity ε_n is what characterizes the initial geometry of a nucleus-nucleus collision. It is also the eccentricity that defines how created particles are distributed in space. We will go into more detail of how later, but for now we will only mention the flow coefficients v_n that we are interested in knowing for every event are proportional to ε_n. As earlier stated, we could simplify the overlapping area of the colliding nuclei to resemble an American football. This idea, however, is only valid in the approximation of spherical nuclei. Nonetheless, we have in the previous sections seen how this is not the case. The nucleus is not spherical, but instead consists of nucleons distributed randomly according to a PDF, and therefore the shape of the nucleus fluctuates event by event. What it also means is that the overlapping geometry is much more complicated than a downscaled American football. Ultimately, it is the participating nucleons that determine the geometry. Hence, we will use the coordinates of the participants and the number of collisions to calculate the eccentricity. The eccentricity is given by [24]

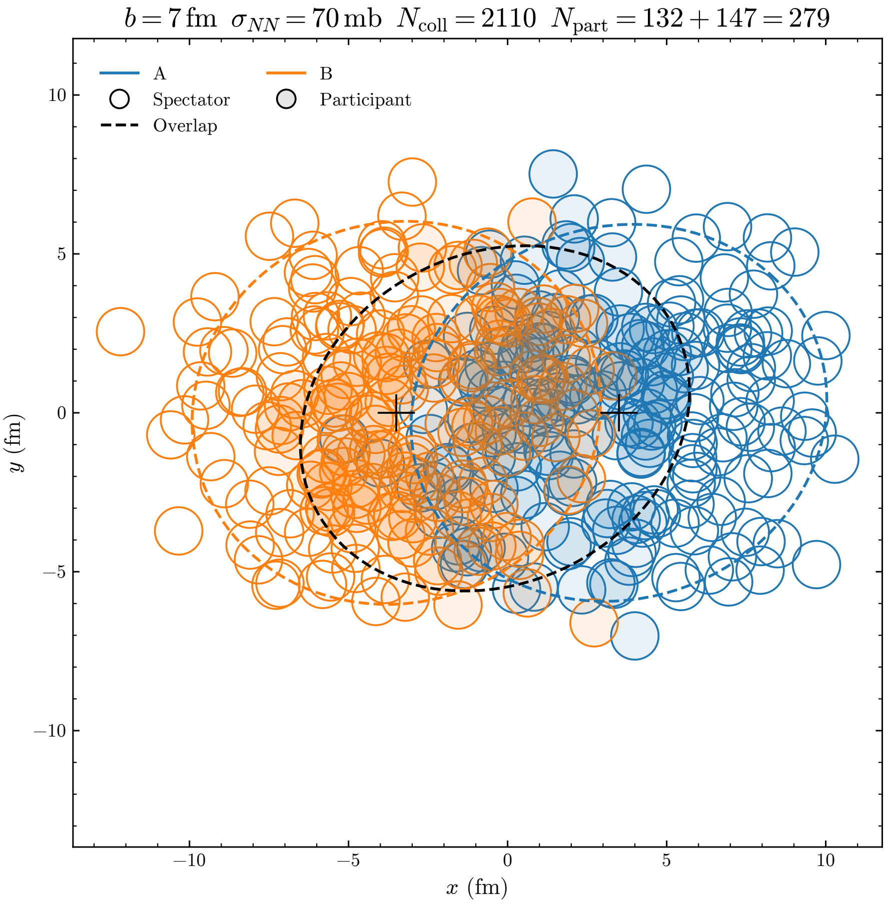


Figure 3.5. A depiction of a collision between two nuclei A and B with an impact parameter of $b = 7\,\mathrm{mb}$. The participating nucleons of both nuclei have slightly shaded circles. In contrast, spectating (non-participating) nucleons are shown with no fill. The black dashed line shows where the nuclei overlap. The overlap region and the region that defines the nuclei (blue and orange dashed lines) are both confidence contours based on the nucleon positions. The number of binary collisions is $N_{\mathrm{coll}} = 2110$, while the number of participants is $N_{\mathrm{part}} = 279$. For this event, we used a fixed cross-section of $\sigma_{\mathrm{NN}} = 70 \pm 5\,\mathrm{mb}$ (where $\pm 5\,\mathrm{mb} = \delta_{\mathrm{NN}}$, is just an uncertainty on σ_{NN}), together with a black disc interaction profile.

$$\varepsilon_n = \frac{\sqrt{\overline{r^n \cos n\phi}^2 + \overline{r^n \sin n\phi}^2}}{\overline{r^n}} \quad , \tag{3.7}$$

where

$$\overline{r^n \cos n\phi} = \frac{1}{N_{\text{part}}} \sum_{i=1}^{N_{\text{part}}} r_i'^n \cos n\phi_i' \tag{3.8}$$

$$\overline{r^n \sin n\phi} = \frac{1}{N_{\text{part}}} \sum_{i=1}^{N_{\text{part}}} r_i'^n \sin n\phi_i' \tag{3.9}$$

$$\overline{r^n} = \frac{1}{N_{\text{part}}} \sum_{i=1}^{N_{\text{part}}} r_i'^n \quad , \tag{3.10}$$

and

$$x_i' = x_i - \overline{x} \tag{3.11}$$

$$y_i' = y_i - \overline{y} \tag{3.12}$$

$$r' = \sqrt{x_i'^2 + y_i'^2} \tag{3.13}$$

$$\phi_i' = \tan^{-1} \frac{y_i'}{x_i'} \quad , \tag{3.14}$$

where $\overline{x}, \overline{y}$ is the weighted average of the participating nucleons positions. They quantify the number of times each nucleon has participated in a collision, and are given by

$$\overline{x} = \frac{\sum_{i=1}^{N_{\text{part}}} w_i x_i}{\sum_{i=1}^{N_{\text{part}}} w_i} \tag{3.15}$$

$$\overline{y} = \frac{\sum_{i=1}^{N_{\text{part}}} w_i y_i}{\sum_{i=1}^{N_{\text{part}}} w_i} \quad , \tag{3.16}$$

where w_i is the number of collisions each participant engaged in. Lastly, the harmonic plane (or orientation of the eccentricity), is specified as

$$\psi_n = \tan^{-1} \frac{\overline{r^n \sin n\phi}}{\overline{r^n \cos n\phi}} \quad . \tag{3.17}$$

In Figure 3.6 we display the same event as in Figure 3.5, except that we also include the eccentricities ε_n. Figure 3.8 shows the variation between the elliptical eccentricity ε_2 for various combinations of interaction and cross-section profiles. A summary plot of 10.000 Pb-Pb events for a black-disc interaction profile can be seen in Figure 3.7, which displays how the different quantities of the Glauber model are distributed.

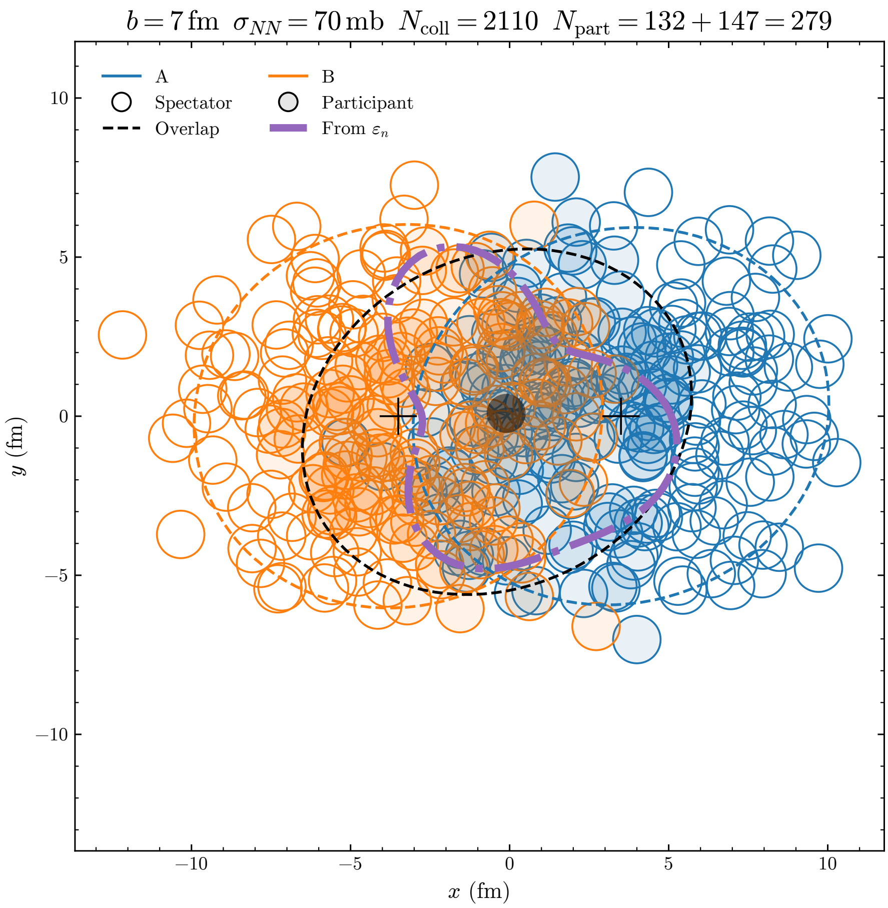


Figure 3.6. The same Glauber event as in Figure 3.5 also showing the eccentricity (purple) of the collision.

3.3 Simulation of Charged Particle Production

Colliding nuclei with high energy in particle accelerators is both a destructive and creative process. The acceleration of the nuclei to near light speed results in extremely high energy densities during the impact of the collision, where the high temperatures cause the nuclei to melt. However, the excess energy available excites the force fields, which leads to the creation of new particles following Einsteins mass-energy equivalence $E = mc^2$. The process has such high levels of complexity that to observe, let alone simulate, all phenomena and interactions occurring in the split of a second the particles collide and new particles emerge, is currently at the limits of our understanding.

A toy model is a powerful tool used to investigate the complexity of nature. It is a highly simplified and idealized model of the real world, and we will, in the following, use

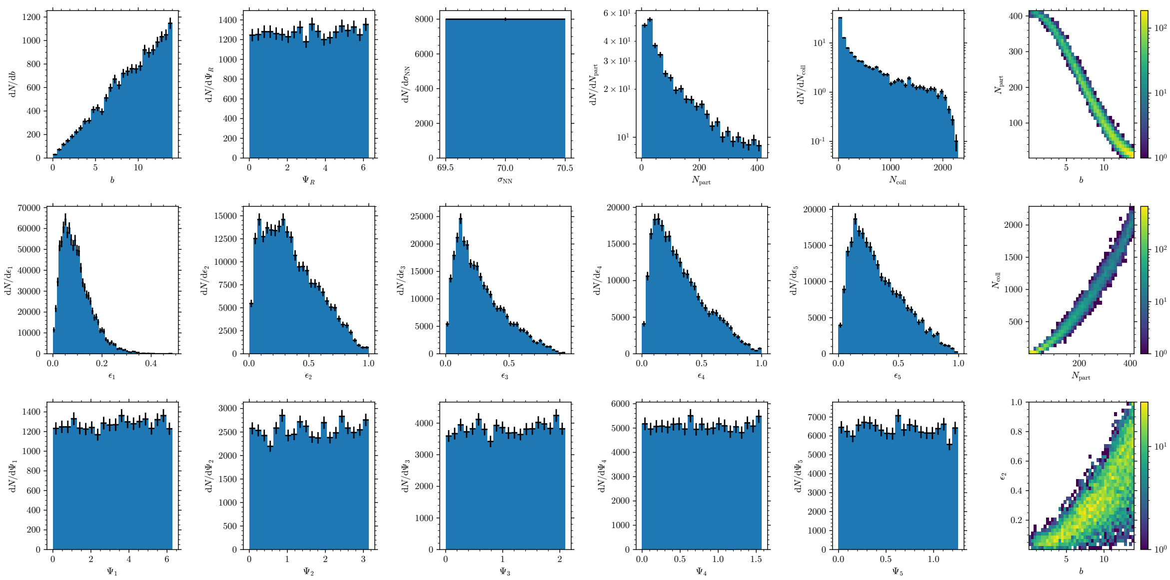


Figure 3.7. A summary plot of 10.000 Pb-Pb events with a black-disc profile and a fixed cross section. In appendix the summary plots are available for other combinations. In the top right corner of the the plot the number of participants N_{part} is shown as a function of impact parameter b. We see that the number of participants decreases as the impact parameter increases. This phenomena is characteristic for all combinations of interaction and cross section profiles, and is a representation of the dependence between central (small b) collisions leading to many interactions and peripheral (high b) collisions leading to few interactions.

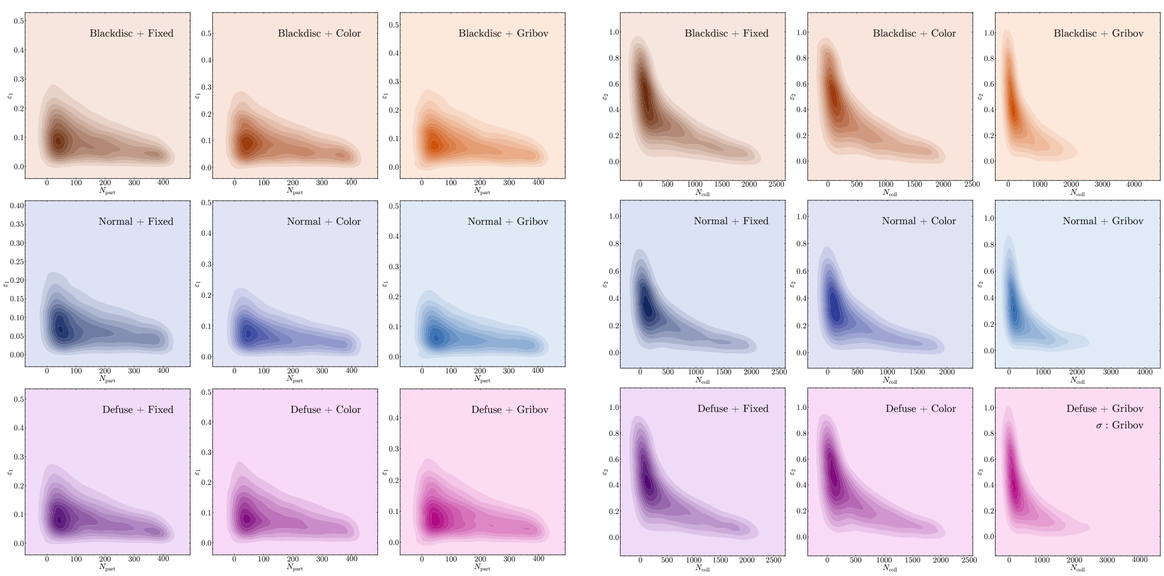


Figure 3.8. Density of ε_2 versus N_{part} (left) and N_{coll} (right) for various combinations of interaction profiles and cross-sections for 10.000 events. We refer to appendix C and D for the review of other eccentricities. The conclusion regardless of interaction and cross-section profile is that the eccentricity is increasing as the number of participants N_{part} decreases. A high number of participants corresponds to a central collision leaving little room for geometrical significance.

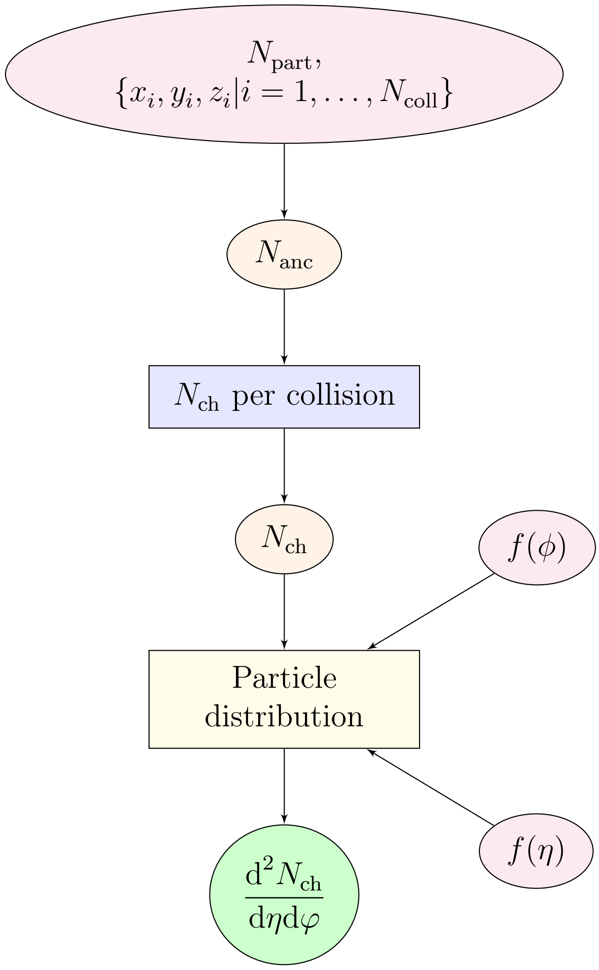


Figure 3.9. Simulation of particle production. Starting from the collisions generated by the Glauber model (see Figure 3.10), we calculate the number of *ancestors* N_{anc}. We then generate a random number of produced particles based on a parameterisation (typically a Negative Binomial distribution) to obtain the total number of charged particles N_{ch}. Given probability distributions for the azimuth angle $f(\varphi)$ and the pseudorapidity $f(\eta)$, we can finally build up our charged-particle density $\frac{\mathrm{d}^2 N_{\text{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$.

a toy model[4] to simulate the number of charged particles N_{ch}. The Glauber model simulates nuclei collisions and computes the number of participants and the coordinates of the interacting nucleons $N_{\text{part}}, \{x_i, y_i, z_i | i = 1, \ldots, N_{\text{coll}}\}$. We will use these calculations as an input to our toy model, see Figure 3.9. After the number of charged particles is known, it will be possible to distribute them spatially in the relevant coordinate system (η, φ). The last step involves using the distributions $\frac{\mathrm{d}^2 N_{\text{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$, representing the observed final state of particles in detectors, in an analysis. For an overall view of the program, see Figure 3.10.

[4] The source code, documentation, and description of the full implementation is available at: https://github.com/zsaldic/sim-ch-particles.

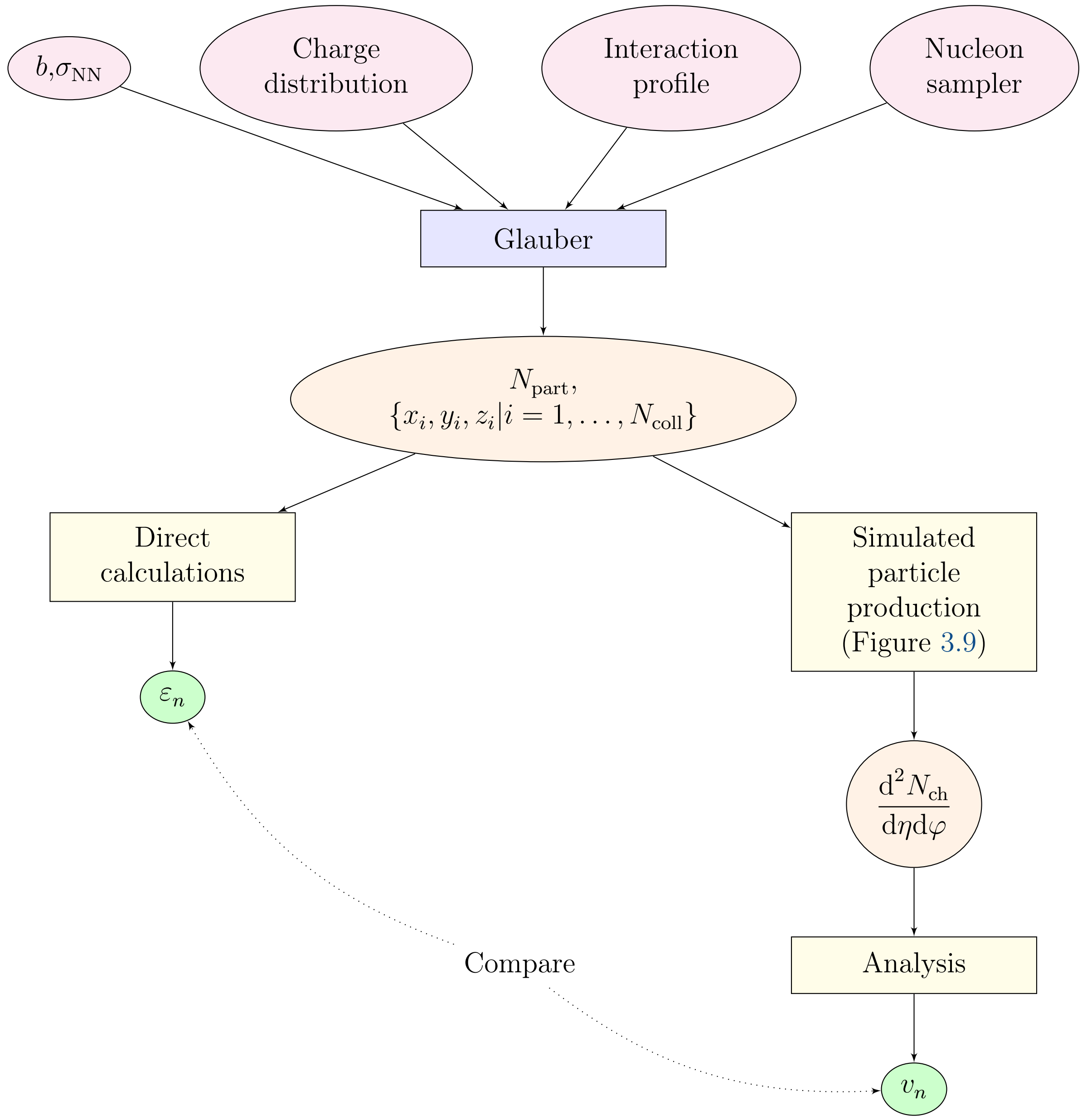


Figure 3.10. Overall flow diagram. The input to the Glauber calculations consist of an impact parameter b, a nucleon-nucleon cross-section σ_{NN}, and parameterisations of the charge distribution, interaction profile, and a method for sampling nucleon positions within a nucleus. The Glauber algorithm outputs number of participants N_{part} and a list of N_{coll} collision coordinates. From these we can directly calculate the eccentricity ε_n. We can also simulate the charged particle production and obtain $\frac{d^2 N_{ch}}{d\eta d\varphi}$ from which we can calculate the flow harmonics v_n.

3.3.1 Number of Charged Particles

3.3.1.1 A Random Experiment

There is a genuine level of randomness associated with many phenomena in nature, and particle production is no exception. A *random variable* characterizes the outcome (event) that can occur from a random experiment. All possible outcomes of the experiment has an associated probability that is defined by the PDF. The event, therefore, describes the random phenomena of the experiment which is manifested as a random variable X, and defined as $X \sim \mathrm{PDF(X = x)}$[5]. Various probability distributions can parameterize the number of charged particles N_{ch}. However, it is conventional to use the Negative Binomial Distribution (NBD) since the random variable from an NBD reflects our understanding of the random process of particle production [25] [26].

A binomial distribution portrays a random experiment by the probability of observing the number of *Bernoulli trials* with a specific outcome. A Bernoulli trial is defined by having a binary outcome:

- True *or* False
- Success *or* Failure
- *A or B*

Thus, we can say that the binomial distribution describes the probability of k A's out of n possible—represented as counts of the number of successes given a certain number of trials. Contrary, the NBD describes the probability of k B's before n A—represented as counts of the number of failures before a given number of successes (or counting the number of successes before a given number of failures) [6].

3.3.2 The Toy Model

The NBD suggests to be appropriate at describing the random nature of particle production. Nevertheless, it is not immediately apparent how it can be implemented in a toy model, together with the Glauber calculations. For this purpose, a formalism that conveys the participants N_{part} and binary collisions N_{coll} into a particle production model is required. The idea is to combine N_{part} and N_{coll} to a single *source* that charged particles N_{ch} can be built from. This source is referred to as *ancestors* N_{anc}, and the model that describes how they are computed is labeled an *ancestor model.* Examples of ancestor models are

- Two-component $N_{\mathrm{anc}} = \alpha\, N_{\mathrm{part}} + (1-\alpha)\, N_{\mathrm{coll}} \quad , \alpha \in [0,1]$
- Hard two-component $N_{\mathrm{anc}} = (1-\alpha)\, N_{\mathrm{part}} \,/\, 2 + \alpha\, N_{\mathrm{coll}} \quad , \alpha \in [0,1]$

[5] A random variable (capital letter) X defines the event, where the possible values it can take are (small letter) x

[6] Because the number of failures is a label we are assigning an *outcome* of a Bernoulli trial, we are allowed to switch the labeling as required.

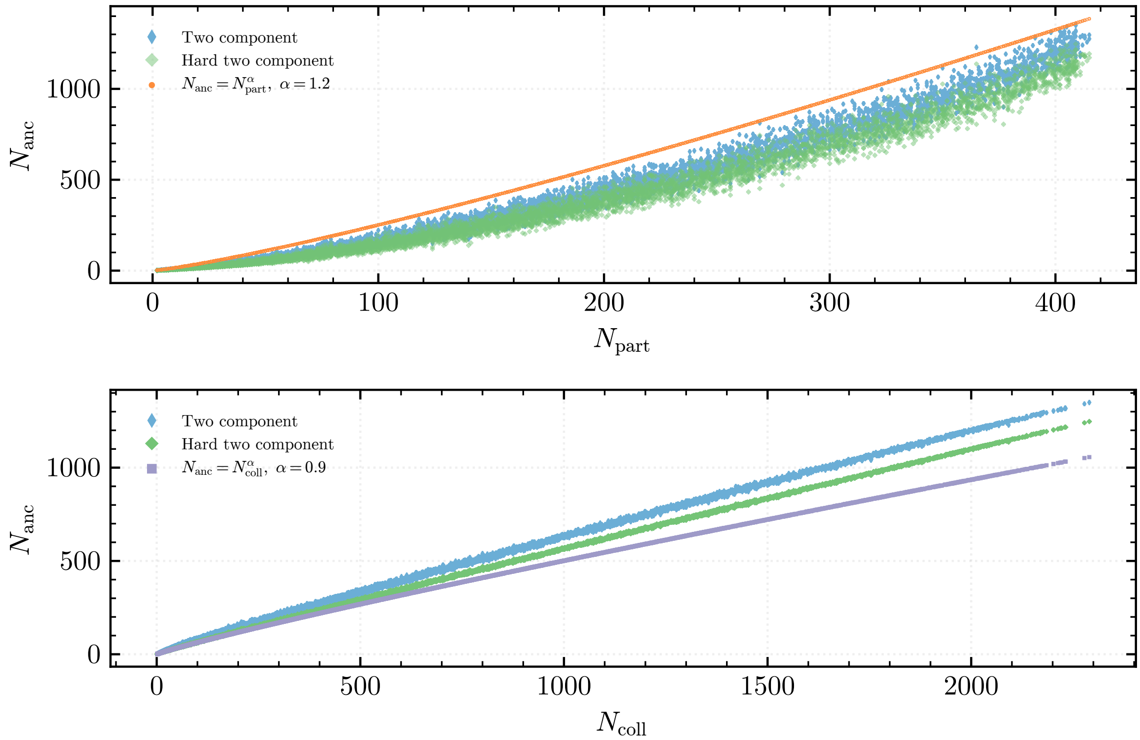


Figure 3.11. The number of ancestors N_{anc} versus the number of participants N_{part} (top) and binary collisions N_{coll} (bottom) with $\alpha = 0.5$ for the Two Component and Hard two component ancestor models.

- Power-law in number of participants $N_{\mathrm{anc}} = N_{\mathrm{part}}^{\alpha}$, $\alpha \in [0.5, 1.4]$
- Power-low in number of binary collisions $N_{\mathrm{anc}} = N_{\mathrm{coll}}^{\alpha}$, $\alpha \in [0.5, 1.1]$

Where α is the ancestor parameter. We show the depends N_{part} and N_{coll} has on the number of ancestors N_{anc} in Figure 3.11.

Using either of these models, we arrive at our number of sources, N_{anc}. Given the number of sources, and a model of charged particle production *per* source, we can calculate the number of charged particles produced. Examples of charged-particle production models are

- NBD $N_{\mathrm{ch}} \sim \mathrm{NB}[N_{\mathrm{anc}}k, \frac{\mu}{\mu+k}]$
- Poisson $N_{\mathrm{ch}} \sim \mathcal{P}[N_{\mathrm{anc}}\mu]$
- Normal $N_{\mathrm{ch}} \sim \mathcal{N}[N_{\mathrm{anc}}\mu, k\sqrt{N_{\mathrm{anc}}\mu}]$
- Constant $N_{\mathrm{ch}} = N_{\mathrm{anc}}\mu$

Thus, we can, among others, distribute the produced number of particles per source according to an NBD. Consequently, we describe the probability of n particles produced before we get k failures in a series of $n + k$ independent trials for a given ancestor.

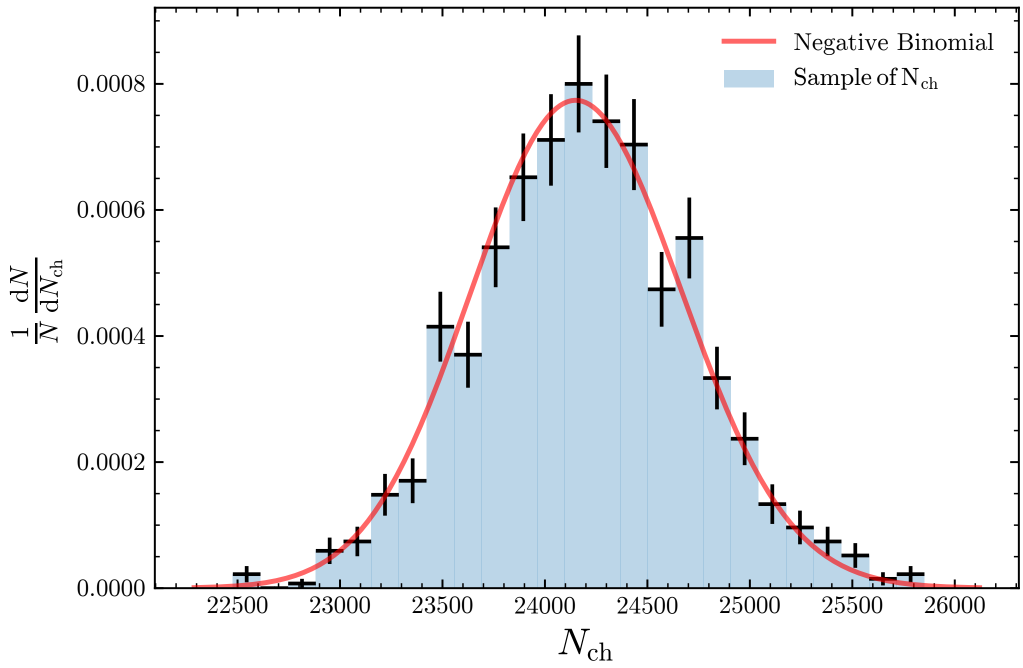


Figure 3.12. The plot showing the NBD when sampling $N_{ch} = 24000$. The histogram represent a realization of the underlying PDF. Here we have chosen $\mu = 20$, $k = 2$, and an $\alpha = 1/2$ in the Hard Two Component model.

3.3.3 The Negative Binomial Distribution

The probability mass function of the number of failures k before the n^{th} success in *independent trials* is given by

$$f(k) = \binom{k+n-1}{n-1} p^n (1-p)^k \quad , \tag{3.18}$$

where p is the probability of a single success. We wish to flip this around to express the exact opposite, i.e., the probability density of the number of *successes* n before the k^{th} *failure*

$$f(n) = \binom{n+k-1}{k-1} p^k (1-p)^n \quad . \tag{3.19}$$

Here we let n define the number of charged particles produced *per* ancestor.
The expectation value

$$\sum_{x=0}^{\infty} x \,\mathrm{PDF}(X = x)$$

of the PDF in Eq. 3.19 is given by

$$\mathbb{E}[n] = \mu = \frac{k(1-p)}{p} \quad ,$$

and characterize the mean value (in this case, the average of the number of charged particles produced *per* ancestor). It also allows us to find the expression for the probability of failure as $p = \frac{k}{k+\mu}$, and the probability of success as $1 - p = \frac{\mu}{\mu+k}$.

3.3.3.1 The γ-Poisson mixture

We will, in the following, show how to rewrite the NBD (Eq. 3.19) as a γ-Poisson mixture. The mixture allows us to sample with the number of ancestors in a continuum $N_{\mathrm{anc}} \in \mathbb{R}$. We begin with the sample of n particles per ancestor from a Poisson with the parameter Λ where, $\Lambda = \lambda$ (the mean of Poisson) itself is a random variable distributed according to $\Lambda \sim \gamma(a, b)$. Here the γ distribution is given by

$$\gamma(\Lambda = \lambda; a, b) = \frac{1}{\Gamma(a) b^a} \lambda^{a-1} \mathrm{e}^{-\lambda/b} \quad , \tag{3.20}$$

where we will come back to the values a and b (the shape and scale parameters). Joining this distribution together with a Poisson we get

$$f(x; a, b, \lambda) = \frac{1}{\Gamma(a) b^a} \lambda^{a-1} \mathrm{e}^{-\lambda/b} \frac{\lambda^x \mathrm{e}^{-\lambda}}{x!} \quad . \tag{3.21}$$

The idea of letting λ be a random variable and allowing it to alternate can be utilized by integrating over all possible (positive) λ values. In this, manner we also obtain the *marginal* (or unconditional) distribution, i.e. a distribution that no longer depends on λ

$$f(x; a, b, \lambda) = \frac{b^{-a}}{\Gamma(a)\Gamma(x+1)} \int_0^\infty \mathrm{d}\lambda\, \lambda^{x+a-1} \mathrm{e}^{-\lambda(1+1/b)} \quad , \tag{3.22}$$

where $x! = \Gamma(x+1)$ holds for x as positive integer values. After integrating and evaluating we get

$$f(x; a, b) = \frac{\Gamma(a+x)}{\Gamma(a)\Gamma(x+1)} b^{-a} \left(1 + \frac{1}{b}\right)^{-a} \left(1 + \frac{1}{b}\right)^{-x} \quad . \tag{3.23}$$

Collecting the terms and rewriting the Γ functions as a binomial coefficient, we finally obtain

$$\binom{x+a-1}{a-1} \left(\frac{1}{1+b}\right)^a \left(\frac{b}{b+1}\right)^x \tag{3.24}$$

which we recognize as the NBD in Eq. 3.19 with the number of failures as $a = k$, the probability of *success* as $p = \frac{b}{b+1}$, and $b = \frac{p}{(1-p)} = \frac{\mu}{k}$.

To model the number of charged particles N_{ch} we will have to let a be proportional to the number of ancestors because the particle production model is modeling for the number of charged particles *per* ancestor. In the following step, we will redefine a by letting it be proportional to the number of ancestors as $a = N_{\text{anc}}k$, and affirm that the found expectation value (mean) μ is the average number of charged particles *per* ancestor. Thereby, we ensure to sample N_{ch} and not $N_{\text{ch}}/N_{\text{anc}}$. After inserting $b = \frac{\mu}{k}$ we arrive to the final distribution of charged particles

$$
\begin{aligned}
N_{\text{ch}} \sim P(N_{\text{ch}} = n_{\text{ch}}; N_{\text{anc}}, k, \mu) &= \binom{n_{\text{ch}} + N_{\text{anc}}k - 1}{N_{\text{anc}}k - 1} \left(\frac{1}{1+\mu/k}\right)^{N_{\text{anc}}k} \left(\frac{\mu/k}{1+\mu/k}\right)^{n_{\text{ch}}} \\
&= \binom{n_{\text{ch}} + N_{\text{anc}}k - 1}{N_{\text{anc}}k - 1} \left(\frac{k}{k+\mu}\right)^{N_{\text{anc}}k} \left(\frac{\mu}{\mu+k}\right)^{n_{\text{ch}}} .
\end{aligned}
\tag{3.25}
$$

The expectation value for the number of charged particles with this particle production model is $\mathbb{E}[N_{\text{ch}}] = N_{\text{anc}}\mu$.

3.3.4 Putting it together

In Figure 3.12 the probability of producing $N_{\text{ch}} = 24000$ particles can be seen for a Hard Two Component model as the ancestor model and the NBD as production model. The *hard* part adds on the idea that N_{coll} scales as the number of collisions with larger Q^2 (momentum transfer), while N_{part} scales as the number of interactions with low Q^2. The most central collision in real experiments produces $N_{\text{ch}} \sim 24000$ particles [27]. We have chosen to tune the shape parameters μ and k such that for the most central collision of $N_{\text{coll}} = 2000$ and $N_{\text{part}} = 416$ will result in $N_{\text{ch}} \sim 24000$. Furthermore, we have chosen an ancestor parameter of $\alpha = 1/2$. If α is zero it favors N_{coll}, i.e. the number of ancestors only depends on the number of collisions. Likewise if α is equal to one it favours only N_{part}.

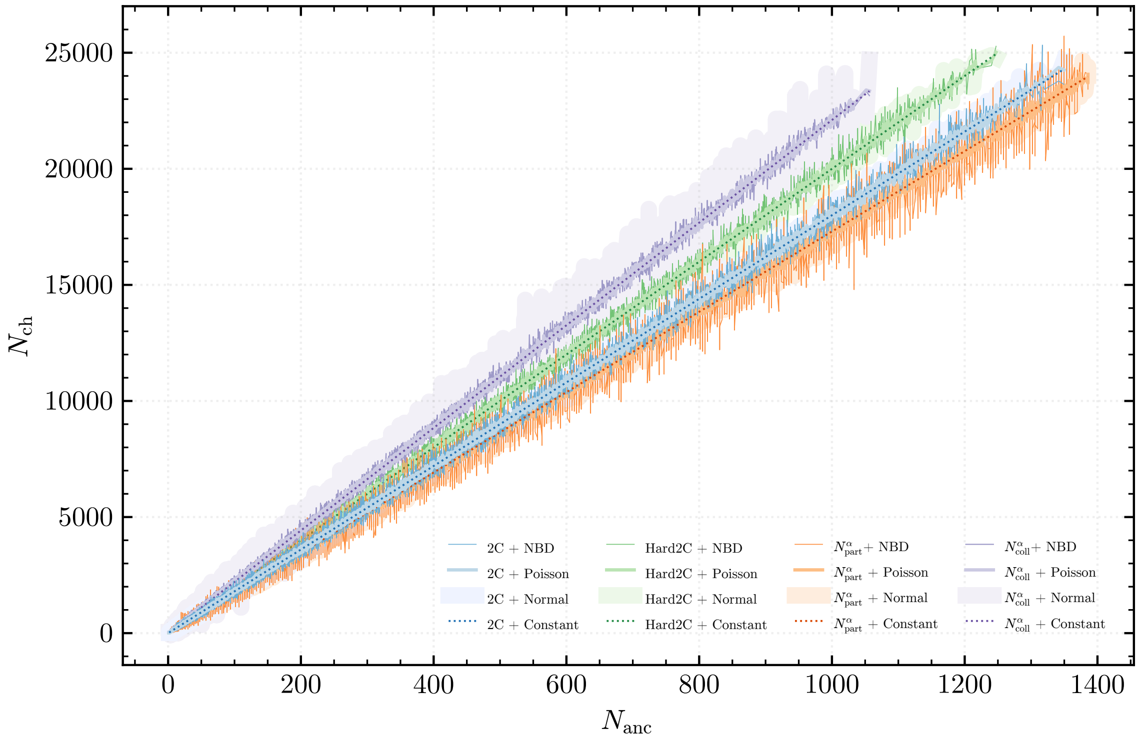
In Figure 3.13a the number of charged particles N_{ch} as a function of ancestors N_{anc} are shown for different combinations of ancestor models. Each of the models had their parameters tuned towards producing $N_{\text{ch}} \sim 24000$. A similar plot replaced by the number of participants N_{part} and number of binary collisions N_{coll} can be found in Figure 3.14. Keeping the μ and k parameters the same for all combinations of toy models results in larger variations in the number of produced particles as shown in Figure 3.15. Nonetheless, the toy models share the same phenomenology, i.e., that the number of charged particles produced from a nuclei collision is monotonically increasing with the increasing number of participants N_{part} and binary collisions N_{coll}.

3.4 Multiplicity Distributions

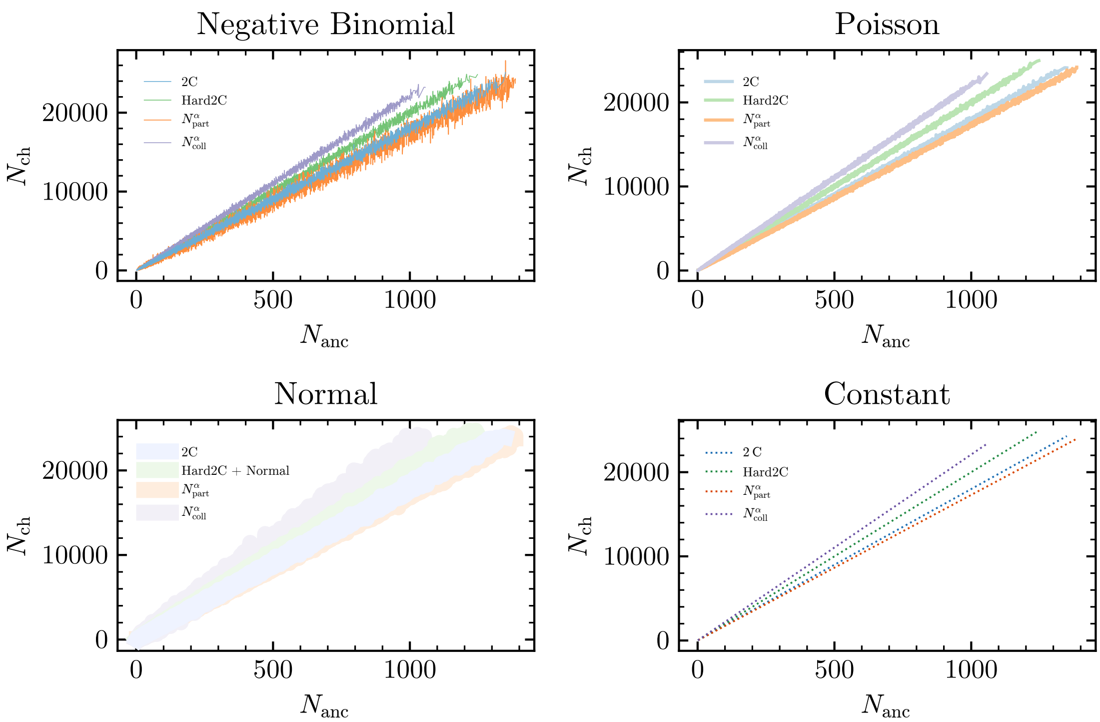
Given the number (multiplicity) of charged particles N_{ch}, the following procedure involves how N_{ch} can be distributed spatially, in pseudorapidity η and azimuthal angles φ. For this purpose, we create the charged particle densities $\frac{\mathrm{d}N_{\text{ch}}}{\mathrm{d}\eta}$ and $\frac{\mathrm{d}N_{\text{ch}}}{\mathrm{d}\varphi}$.

3.4.1 Pseudorapidity Densities

The term *pseudorapidity* originates from of its approximation to *relativistic rapidity*. Rapidity is a way of relating a particle's energy and momentum (along the beam axis) to its position, and is given by



(a) The number of charged particles N_{ch} as a number of ancestors N_{anc}. The same plot separated in can be seen in Figure 3.13b.



(b) Same as in Figure 3.13a but separated in the various production models. A production model following a normal distribution leads to the highest spread while a constant production model has no spread (the number of ancestors it requires to build charged particles from is strict).

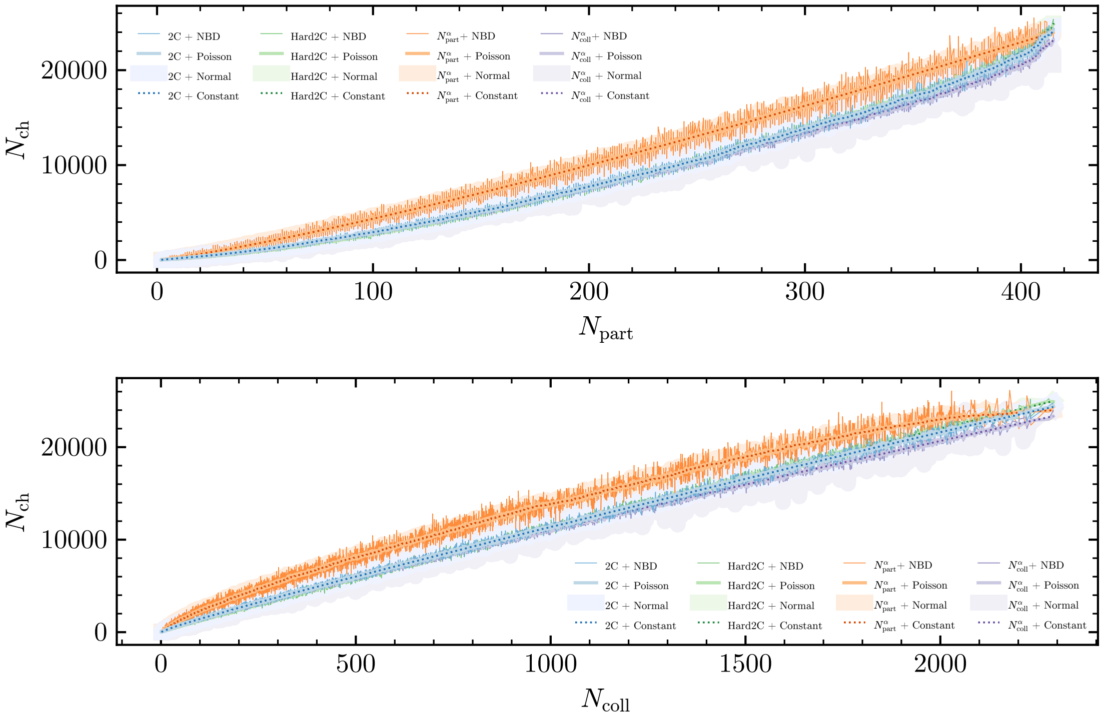


Figure 3.14. The number of charged particles as N_{part} (top) and N_{coll} (bottom). The parameters α, k and μ has been tuned to produce approximately $N_{\text{ch}} \sim 24000$ for the most central collision where $N_{\text{part}} = 416$ and $N_{\text{coll}} \sim 2000$. In Figure 3.15 the effect of $\mu = 20$ and $k = 2$ fixed for all models can be seen.

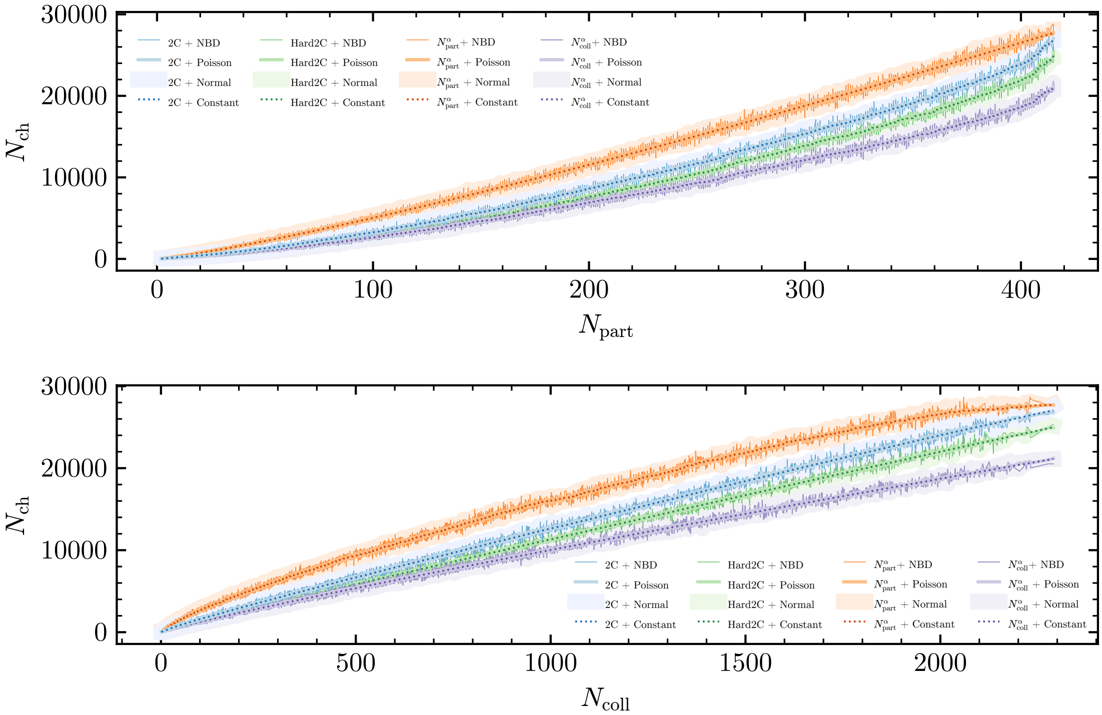


Figure 3.15. Same as in Figure 3.14 only that here the paramters of $\mu = 20$ and $k = 2$ are fixed for all particle production models. We see that this influences how many participants and binary collisions it requires to produced a given number of charged particles.

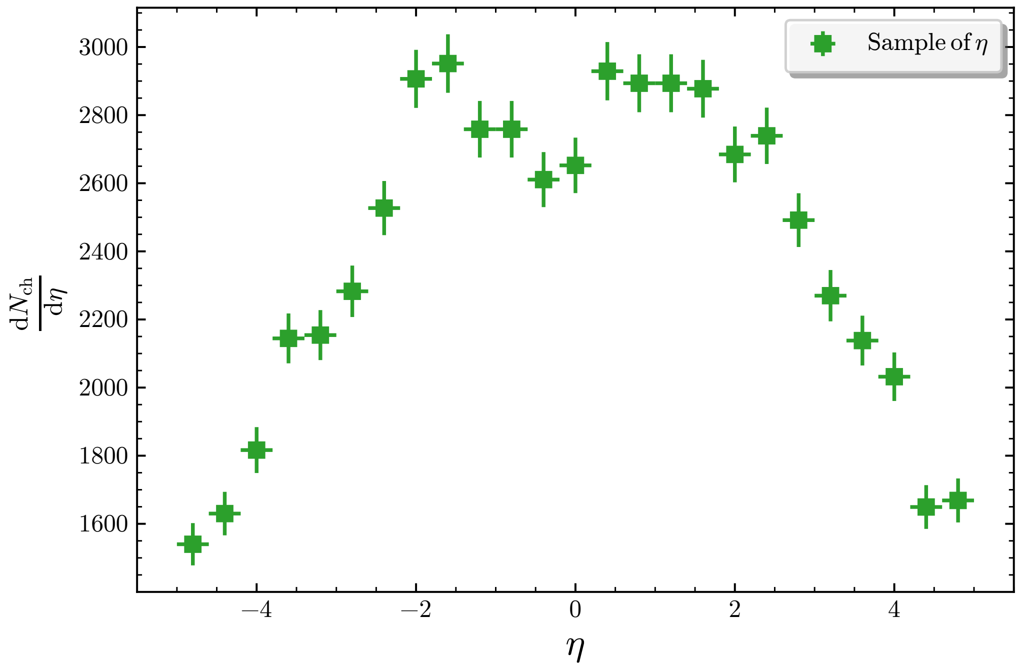


Figure 3.16. A pseudorapidity distribution of the $N_{ch} = 24000$ particles.

$$y = \log\left[\frac{E+p_z}{E-p_z}\right] \quad . \tag{3.26}$$

The reason particle physicists use rapidity as a measure of position is because the difference in the rapidity of two particles is Lorentz-invariant under boost along the beam direction (z-axis). Different observers, i.e., detectors, measure the same difference in rapidity regardless of where the observation has taken place. This fact assists in identifying both the exact origin of the collision and the exact position of all particle products from the lab frame, when things get messy due to the effects of special relativity. The challenging is to measure the energy and total momentum of every particle. This challenge is the reason why pseudorapidity is introduced in the first place. If the momentum of a particle is much higher than its mass, i.e., $p \gg m$, most of the particle's energy *is* momentum, $E \to p$ (which is the case for ultra-relativistic experiments). If we insert these limits in Eq. 3.26 we obtain the pseudorapidity

$$\eta = -\log\left[\tan\left(\frac{\theta}{2}\right)\right] \quad , \tag{3.27}$$

where θ is the polar angle relative to the beam axis. Hence, pseudorapidity approximates towards relativistic rapidity $\eta \approx y$ in ultra-relativistic nuclei collisions.

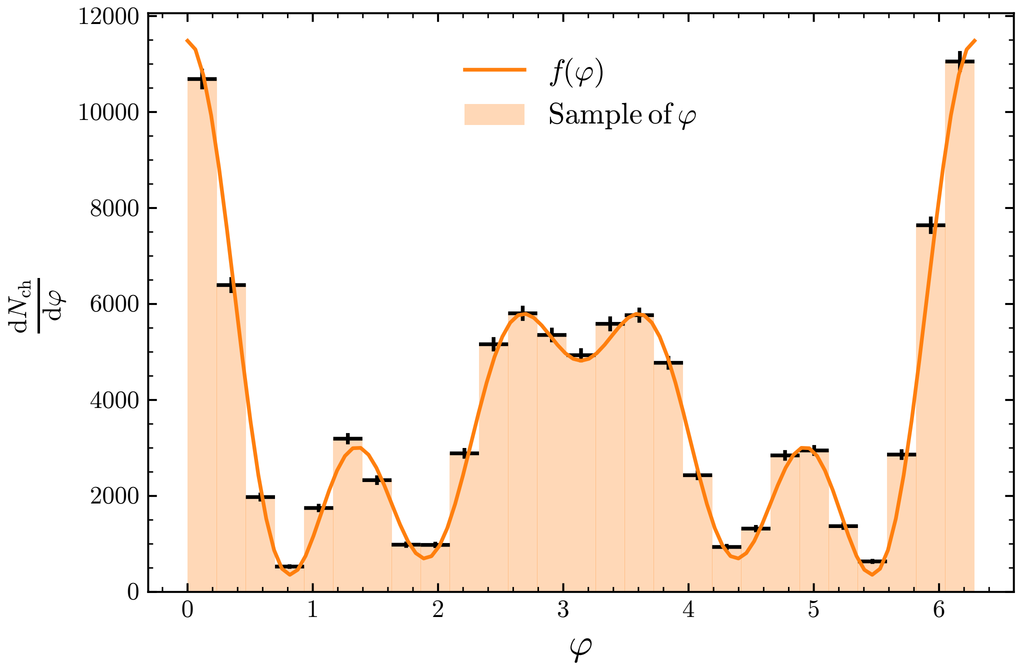


Figure 3.17. The azimuthal distribution of the $N_{ch} = 24000$ particles.

We will use a simple method [28] to distribute the number of charged particles in pseudorapidity—where the first step is to generate the rapidity distributions. The rapidity y distribution can be modeled with a Normal distribution

$$y \sim \mathcal{N}(0, \sigma)$$

where the spread is $\sigma = 4$. We transform rapidity to pseudorapidity by

$$\eta = -\frac{1}{2}\log\left(-\frac{a^2t^2 + t^2 - 2t\sqrt{a^2+1}\sqrt{a^2t^2+1}+1}{t^2-1}\right) \quad , \tag{3.28}$$

where $t = \tanh y$, and $a = 1/2$ is the mass to transverse momentum ratio set equally for all particles. Figure 3.16 shows an event of $N_{ch} = 24000$ particles distributed in pseudorapidity.

3.4.2 Azimuthal φ Densities

Provided the calculations of the eccentricities ε_n and symmetry plane angles Ψ_n from the Glauber model we can distribute N_{ch} in the azimuthal angles φ according to Eq. 2.1. However, a conversion of ε_n to the flow coefficients v_n is necessary. We assume a linear relation with the constants κ_n

$$\kappa_n = [0.5, 2, 1, 1, 1] \tag{3.29}$$

such that

$$v_n = \kappa_n \varepsilon_n \quad . \tag{3.30}$$

The factors[7] represent the assumptions that the system couples more to the total transverse momentum in the direction perpendicular to the event plane as suggested by experiments [30]. Furthermore, we also rotate the angle of emission relative to the epsilon (symmetry) planes. This rotation is to simulate how the particles flow out more in the plane than out of plane due to the pressure gradient. A vanilla distribution without any rotation or extra phases can be seen in Figure 3.17.

3.5 Final Charged Particle Multiplicities

We end this chapter by merging the distributions in pseudorapidity η and azimuthal angles φ to a common density $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$, as shown in Figure 3.18 and 3.19. We use a 32×32 bins detector granularity with a full coverage of the azimuthal angle, and a pseudorapidity coverage $-5 < \eta < 5$ corresponding to the ALICE experiment [31]. We have simulated the end product of a nuclei collision, the so-called final state multiplicities in (η, φ). Hence, as particles gets produced in a nuclei collision and scatter off in multiple direction they will eventually interact with a detector, i.e., the particles deposit their energy in the respective particle detectors (hadronic or electromagnetic calorimeters) surrounding the interaction point, where each pixel in Figure 3.18 and 3.19 corresponds to a particular detector segment. The pixel colors represent the number of particles hitting a particular detector segment—with a brighter pixel corresponding to a higher number of particle hits, and a darker pixel representing few or no particle hits.

The task open for investigation is whether it is possible to create a Machine Learning model capable of associating the patterns present in the final state densities of charged-particles $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$ with the flow coefficients $v_1, v_2, ..., v_5$, assuming that the flow coefficients are proportional to the initial state geometry $\varepsilon_1, \varepsilon_2, ..., \varepsilon_5$ of the nuclei collision.

[7] The the factors κ_n are more complicated nature, and depends on multiple parameters of the QGP, such as the equation of state, shear viscosity, initial densities, and freeze-out temperature [29].

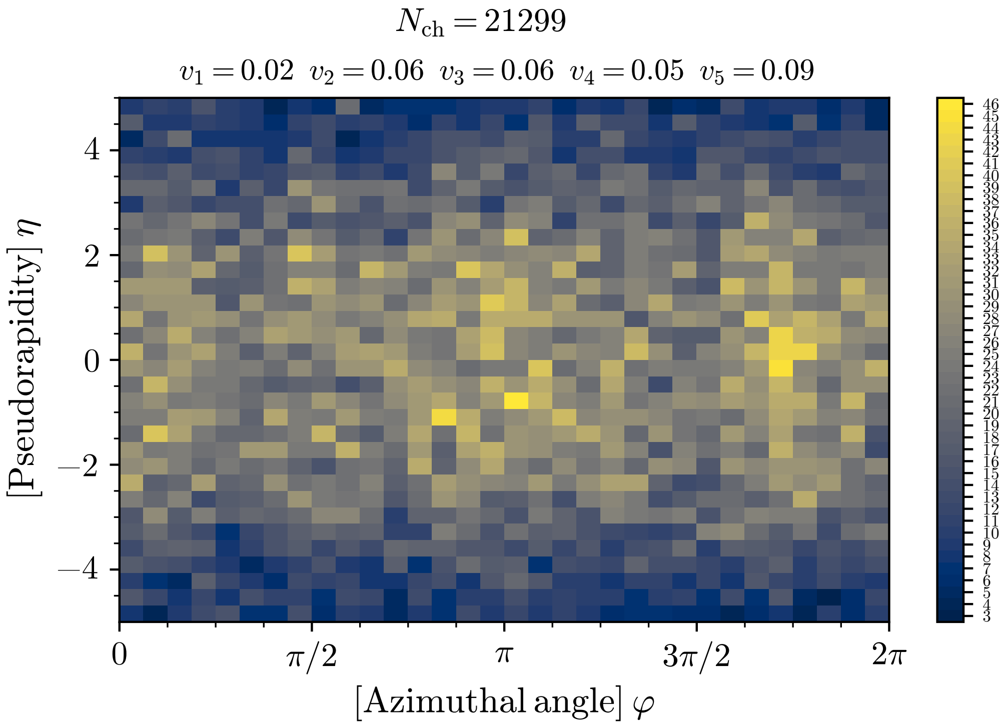


Figure 3.18. A highly central event producing $N_{\rm ch} = 21299$ charged particles. Notice that v_2 is smaller compared to a more peripheral event in Figure 3.19.

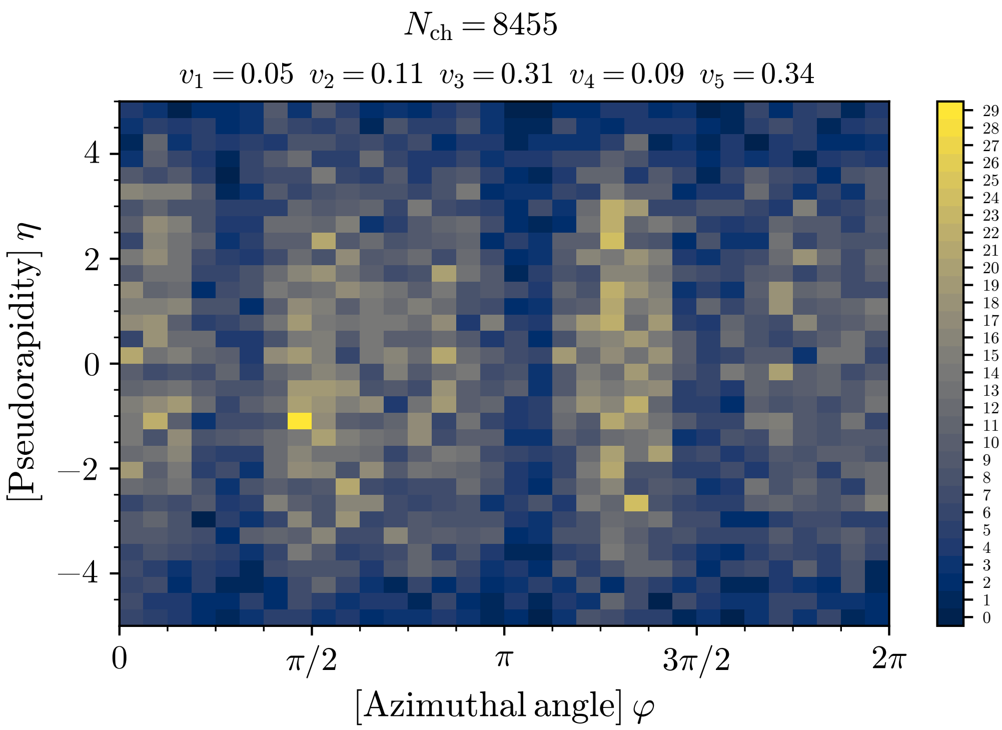


Figure 3.19. A semi-central event producing $N_{\rm ch} = 8455$. This event has a lot more structure in it compared to the more central event in Figure 3.18.

4

Methodology

We begin this chapter with a quick overview of the basic concept of Machine Learning. Next, we briefly consider Deep Learning's capabilities and limitations. We continue from there by introducing neural networks and how we apply them in to our particular problem. Lastly, we present the final model architecture and the measures made to justify it.

The idea of Machine Learning (ML) can be boiled down to building a model by constructing a computer program capable of finding patterns in data. ML is, therefore, more an umbrella term of several methods that try to solve different kinds of problems related to many scientific fields and industries. ML's general approach goes along the lines of: We have some data that consists of an input $\mathbf{x}$ and a *target* value y. The target value represents the thing we are trying to predict. There are possibly more *features* that describe the input data $\mathbf{x} = [x_0 \ x_1 \ \dots \ x_n]$. We apply a specific function f that consists of some parameters θ to this input data

$$f(\theta) : \mathbf{x} \to \hat{y} \quad .$$

The primary purpose is now to update the function's parameters via some optimization algorithm such that the error between the *true target* y and *predicted target* $\hat{y}$ is as small as possible. Here the function f represents the ML model.

A classical example of the application of ML is in the real estate business, where one tries to estimate the (best) price of a house. One data point (one house) has many different features, such as size, number of bedrooms, and condition. The goal is to estimate the price (predict the target value) given the attributes of the house (the feature of the data). House pricing is a classic example of a regression problem in ML. Another popular type of problem could be a classification problem that assigns a label to the target value instead of a certain number. This label could be sick/not sick, pion/not pion, pass/fail. In this thesis, we wish to build an ML regression model capable of predicting five distinct target values (the flow coefficients $v_1, v_2, ..., v_5$) from the input data that represents an image (the final state density of charged particles $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$) and consists of $32 \times 32 = 1024$ features (particle hits).

4.1 Deep Learning

Deep Learning refers to the potentially many layers of connected neurons in a neural network. It is a rather intriguing concept that a Deep Learning model in principle can approximate any continuous function, both linear and non-linear, as long as the model has enough neurons implemented. In this manner, a neural network can learn the features of a sub-sample of multidimensional data to a high degree. This concept is known as the *universal approximation theorem* [32]. One challenge with this concept is to balance the number of neurons with the computation time required to execute the model training. However, it is desirable to have a neural network that is both high in prediction/accuracy as well as being able to *generalize*. Generalization implies that the model can infer a reasonable estimate on data that it has not seen before[1]. The dilemma also goes under the name of the *bias-variance tradeoff*, but can also be criticized by the *No Free Lunch theorem.* This theorem suggests that if a highly specified algorithm perform well in some region of sample space, it will perform poorly or even worse than a completely random utilized algorithm in other regions.

4.1.1 Neural Networks

We will now turn our attention to the working principles of a neural network (NN). A NN consists of interconnected neurons performing matrix multiplication (dot product) on its input and then composes this sum to an activation function. Consequently, each neuron is associated with a set of so-called *weights* w. Each weight w_i is multiplied on a particular input value x_i. For this reason, the number of weights is the same as the number of inputs. Furthermore, a numerical value, also known as a *bias*, can be added to the weighted sum. The purpose of the bias is to constrain further whether the particular neuron is activated or not. Thus, the value of its weights and biases determines the strength of its activation. The activation function is there to pass on a signal to the next layer of neurons. A popular activation function is the *Rectified Linear Unit* (ReLU) [33] which outputs a signal linearly if the weighted sum is positive and outputs zero if the weighted sum is negative

$$\mathrm{ReLU}(z) = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases} \quad , \tag{4.1}$$

where $z = \sum w + bias$, is the weighted sum plus a bias of the neuron. Our goal is to create a neural network capable of associating a particular pattern of an image with five numerical values. We remind ourselves that the pixels of the image represent particle hits. Some events (images) have many particle hits distributed in some pattern-like fashion, while others have few particles hits with no apparent pattern (to humans). We want the NN to learn how the pixels' values and their spatial distance between one another are related to the flow coefficients. Dividing a single image with its maximum hit value would render an image with all pixels lying in the range between zero and one. This

[1]Think about an ML model capable of predicting house prices but only has trained on houses from Lolland. The model will likely underpredict the price if it were presented with a house from North Zealand.

arrangement allows us to view the particle densities $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$ as a simple greyscale image. If the image works as the first layer of the network (the input layer), then each pixel's grayscale values determines the activation of each neuron in the input layer.

The classic approach towards analyzing an image is by introducing the basic feed-forward NN. The first step consists of flattening the 32×32 greyscale image. We now have a vectorized image of $32 \times 32 = 1024$ columns (or rows). All the activations of the neurons in the input layer determine all the activations in the following layers. Utilizing a single neuron on this flatten image vector would require the initialization of 1024 weights to perform the matrix multiplication. For this reason, a single neuron has 1024 parameters plus a bias. A model consisting of only 100 neurons would already yield 102.500 parameters. We will come back to how this problem can be avoided later by introducing Convolutional Neural Networks (CNN) that are powerful at analyzing images.

4.1.1.1 How a machine learns?

Gradient descent is the central underlying idea of how a Neural Network (NN) or machine learning algorithm learns in general. During training, the NN is provided with images and the labels of the flow coefficients. The idea is now that we make an algorithm adjust the neurons' weights such that the overall performance of the NN improves. The goal is then that the NN generalizes to predict the flow coefficients from images it has not seen before (referred to as test data). The weights and biases are initialized randomly, to begin with, resulting in poor performance. Improving the performance requires the introduction of a *loss* or *error* function. The fundamental meaning of a NN learning simplifies towards minimizing the loss function.

An example of a loss function is the L1 loss, also known as the mean absolute error (MAE). For a single image i the loss function yield

$$\mathcal{L}_i = \sum_{n=1}^{5} |\hat{v}_n - (v_i)_n| \quad , \tag{4.2}$$

where the total loss is given by

$$\text{L1 loss} = \mathcal{L} = \frac{1}{N} \sum_i \mathcal{L}_i \quad . \tag{4.3}$$

Here N is the size of the training sample (number of images), and $\hat{v}_n = \hat{v}_n(w)$ is the prediction of the flow coefficients, which is depends on the weights w in the NN. This loss function computes the absolute difference between each actual flow coefficient with the one predicted from the NN. Furthermore, the loss function depends on the parameters of the NN. It takes the weights and biases as an input and returns a single value corresponding to the entire training set's variation from the actual flow coefficients.

We are now interested in finding the values of the weights and biases that minimize the loss function. This idea leads us to compute the gradient of the loss function.

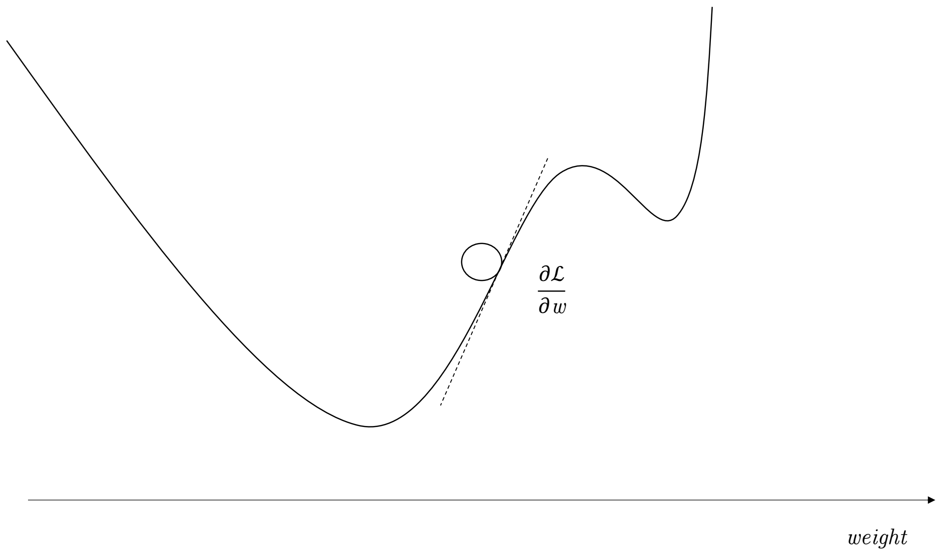


Figure 4.1. A schematic representation of the optimization scheme (the ball) with the gradient (dashed line) in the process of finding a minimum of the loss function $\mathcal{L}$.

Knowing the gradient (the direction of the steepest descent) enables us to update the weights accordingly by taking a small step in the direction of the gradient, which is the notion of gradient descent, until a local minimum is reached

$$\frac{\partial \mathcal{L}(w)}{\partial w} = 0 \quad . \tag{4.4}$$

The small step is also known as the learning rate of the NN. The gradient gives an estimate of how the weights should be changed to minimize the loss most efficiently. However, there is no guarantee, given the random initialization of the neuron weights, that ensures finding the global minimum. The main algorithm responsible for computing the gradient is known as *backpropagation*—while we deem the derivation of the mathematical details of this technique beyond the scope of this thesis, we note that this technique is well described in the literature. However, it is worth mentioning that the global gradient can be computed, and the weights adjusted accordingly by utilizing the chain rule from calculus. Since the activation of the final output depends on previous activations of neurons, the amount the final activation has to be changed (determined by the loss) is influenced by how much all previous activations need to be changed.

Computing the gradient, updating the weights, and determining the minimum of the loss is altogether summarized as *optimization*. There exist multiple optimization routines; a particular one is the Adam (adaptive moment estimation) [34] optimizer. The strength of the Adam optimizer is that it facilitates the adjustment of learning rates during training. The adjustment can allow the optimization to gain momentum over small "bumps" in the loss function avoiding getting stuck in a sub-optimal local minima.

The rapid increase in the number of fully connected neurons could quickly become a challenge concerning computation time. However, there is one more reason why this method would be inadequate for images of our type. Flattening the image removes the information in a single-pixel strength and structure relative to other pixels in the image. When we flatten an image to a vector, each pixel only has two neighbors, and the composition relative to all other pixels is lost. This information is crucial for our task. The intention is that the strength, distance, and distribution of the pixels relative to one another in the image, contains enough information to estimate the collision's original geometry. The Convolutional Neural Network (CNN) avoids the problem altogether, and we will briefly summarize its core concepts.

4.1.1.2 Convolutional Neural Networks

By introducing CNNs, it is no longer necessary to flatten an image beforehand. A CNN takes an image as input, searches for some particular properties, and outputs a new image representing activations of these properties, respectively. Consequently, two main ideas lay the foundation of the CNN: *sparse connectivity* and *weight sharing*. Sparse connectivity refers to how neurons in a CNN only connect to parts of the image instead of being fully connected. Hence, each neuron in a CNN layer applies the same form of matrix multiplication as earlier introduced, but only on a specific patch of the image. Weight sharing refers to how the neurons' weights are shared and, therefore, *fixed* for the entire layer of neurons. The shared values of these weights is also defined as a *filter* or sometimes *kernel*. For this reason, the filter size also determines the size of the previously mentioned patch. Hence, the concept of a CNN layer is to slide (convolve) over an image, pixel by pixel, with a characteristic filter. A CNN layer can be considered a feature detector that searches for a specific pattern or feature in the image—these are the distinct kind of patterns that the filter learns to search for during the optimization of the CNN.

4.1.2 Network Architecture Considerations

The first attempt at a network architecture was within the strategy of keeping it as simple as possible, i.e., constraining the number of parameters to reduce the computation time and prevent overfitting (failure of generalization) while ensuring a decent performance. The general rule of thumb, to build a CNN, is to stack multiple CNN layers each time rising in the number of filters (depth) followed by fully connected layers [35]. The main idea is that the first layer of filters receives a general knowledge of the image, whereas the subsequent layers learn the importance of each region in more detail. This idea matches our objective of a network capable of finding a more general pattern of the $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$ image. A large filter size is thereby also more appropriate for this task.

Furthermore, it is suggested that it is preferable to stack multiple layers than having a single or a few CNN layers with large filters. Stacking with a fewer number of parameters opens the possibility of capturing same levels of complexity as models that have substantially higher number of parameters. Because the filters re-outputs an image of the same size, a technique to reduce the dimensionality of the image known as pooling can be applied potentially after each layer. Max pooling, e.g., returns the maximum of a patch of 2×2 pixels. Other models suggest reducing dimensionality by alternating the stride of the filters [36]. The above leads us to a typical architecture [37] usually in the form of

$$\text{INPUT} \rightarrow [[\text{CONV} \rightarrow \text{ACTIVATION}]\times N \rightarrow \text{POOL}]\times M \rightarrow [\text{FC} \rightarrow \text{ACTIVATION}]\times K \rightarrow \text{FC(OUTPUT)}$$

where usually $0 \leq N \leq 3$, $M \geq 0$, and $0 \leq K < 3$.

4.1.2.1 Hyperparameters

The actual number of filters, filter size, activation function, pooling type, and learning rate are all subjects to the model hyperparameters. Relying on finding the optimal combination by plain trial and error can quickly turn into an inefficient process. Instead, it is often beneficial to finetune the model. Finetuning involves training a model with many combinations of hyperparameters via a search algorithm such as random search. Before any advanced finetuning algorithm, it is advisable to overfit a small sample of data with some initial architecture. The process ensures that the model can converge towards the desired output. Therefore the initial process of building the model consisted of training on a data set of only 20 images.

4.2 The Proposed Model

We use a Convolutional Neural Network (CNN) to predict five continuous numerical values v_1, v_2, v_3, v_4, v_5, the so-called flow coefficients directly proportional to the initial state ε_n of nuclei-nuclei collision, by analyzing the final state charged-particle multiplicity densities $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$. The model's implementation was made with the Machine Learning platform TensorFlow [38] with the Deep Learning API Keras [39].

The model consists of three convolutional layers, each followed by a 2×2 max-pooling layer. The first layer has 8 filters with a filter size of 11×11, the second layer has 16 filters with a filter size of 5×5, and the last layer has 32 filters with a filter size of 3×3. After the last pooling layer the output is flattened to a 256 input vector, where we follow up with two fully connected layers, with 32 neurons each. The output layer consists of 5 neurons corresponding to the five flow coefficients v_1, v_2, v_3, v_4, v_5, as seen Figure 4.2. The last layer's activation function is by default chosen to be the ReLU since it is non-physical to have negative flow coefficients. We found that the best performance was achieved with a LeakyReLU

$$\text{LeakyReLU}(z) = \begin{cases} z & \text{if } z > 0 \\ \alpha z & \text{if } z \leq 0 \end{cases}, \tag{4.5}$$

with $\alpha = 0.001$, in all other layers. Furthermore, we use the Adam optimization routine with an initial learning rate of $\mathrm{lr} = 0.0003$. The initial weights were specified with the glorot normal initializer, also called Xavier normal initializer [40]. The final considerations regarding the specific architecture were based on reducing the number of model parameters as much as possible and still achieving satisfying performance. Furthermore, reducing parameters ensured a simple model less prone to overfitting, thereby avoiding techniques such as regularization and dropout layers. The decision for the optimal filter

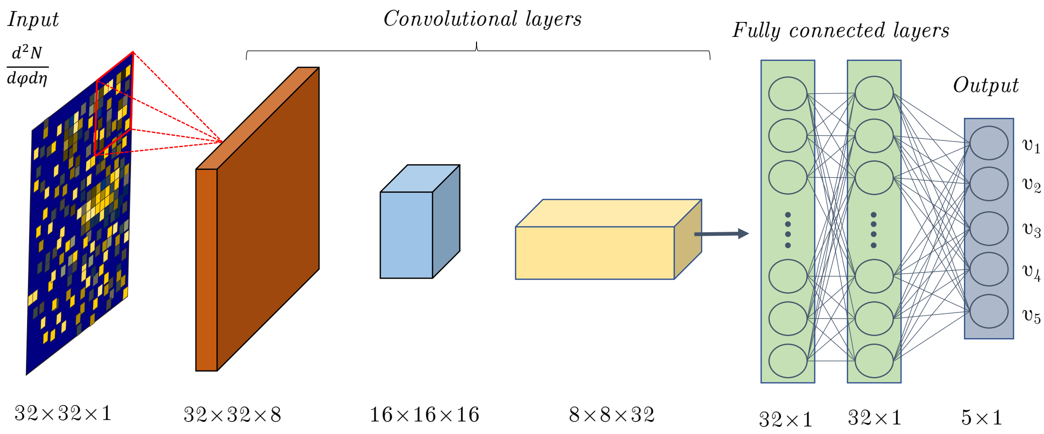


Figure 4.2. The overall model architecture. The input image and CNN layers are zero padded on the edges to preserve the input size when we apply the filters. Each convolutional layer is also followed by a maxpooling layer of factor 2 to reduce dimensionality. The last CNN layer is flattened to a 512×1 vector before being fully connected to the first dense layer.

size, activation function (and α), weight initialization, learning rate, and loss function, were altogether based on the HyperBand search algorithm offered by the Keras Tuner Library [41]. This algorithm is a form of random search that trains on multiple different models specified by the given hyperparameters for a small number of iterations and continuous training with the best performing models. Therefore, it works as a guide towards finding the optimal hyperparameters by ranking each model on its performance. The choice of performance in this search was the validation loss. We used a mini-batch strategy [42] during training with a batch size of 64 images to achieve faster convergence through backpropagation.

5

Results

5.1 Performance

There exists multiple ways to evaluate the performance of an ML algorithm. We evaluate our model with the Mean Absolute Error (MAE), which shares the same properties as the loss function in Eq. 4.3

$$\mathrm{MAE} = \frac{1}{N}\sum_{i}^{N} |y_i - \hat{y}_i| \quad , \tag{5.1}$$

where y is the true value and $\hat{y}$ is the predicted value. The data set used to train the CNN was composed of a mixture of interaction and cross-section profiles, with the

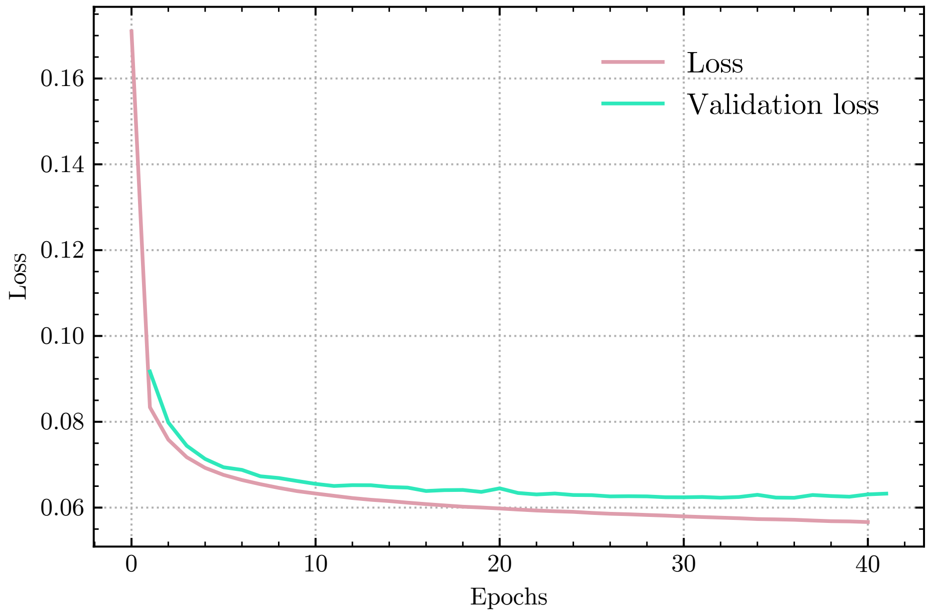


Figure 5.1. The training progress on a sample of 100.000 events consisted of an even mixture of interaction and cross-section profiles. The ancestor and production models were held fixed with a Two Component ($\alpha = 0.5$) as ancestor model and NBD ($\mu = 20$, $k = 2$) as the particle production model.

Two-Component ($\alpha = 0.5$) as ancestor model and the NBD ($\mu = 20$, $k = 2$) as particle production model. We used an initial training set consisting of a total of 90.000 events: 10.000 events for each combination of interaction profiles (black-disc, Normal, and defuse) and cross-section profiles (fixed, gribov, and color). The progression of the optimization routine is shown in Figure 5.1 and demonstrates adequate convergence. The model trained on accelerator hardware with the NVIDIA TESLA K80 GPU run from a virtual machine provided by the Google Cloud Platform compute engine. In Figure 5.2, we plot the residuals (errors) of the predicted flow coefficients. The error describes the difference between the real value and the predicted value, and therefore, is a measure for how much our model predictions deviate from the actual value. Good performance and overall strong peaks can be seen for all v_n's except for v_1. Next, we compute the Pearson product-moment correlation coefficient (PPMCC) (also known as the Pearson correlation coefficient)

$$\text{PPMCC} = \frac{\mathbb{E}[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \sigma_Y} \quad , \tag{5.2}$$

where $X = \hat{v}_n$ in this case are the predicted flow coefficients and $Y = v_n$ are the true flow coefficients. A PPMCC = 1 implies a perfect prediction, whereas PPMCC = 0 suggests no correlation between the predicted and the true values. In Figure 5.3 the PPMCC as well as a scatter plot between true and predicted are shown. We see a strong correlation (prediction) for all flow coefficients except v_1. However, it can be seen that there are fluctuations present for $v_3, v_4, v_5 > 0.3$ and $v_2 > 0.9$. We would like to investigate whether these fluctuations are due to the systematic uncertainties of the model or due to statistical uncertainties in the data.

5.2 Uncertainties & Discussion

If the deviation from the true values is caused by the Neural Network (NN) then it is subject to a systematic uncertainties provided that adding more data does not decrease the error further. It is expected that the statistical uncertainty σ decreases as the sample size N increases, as indicated by the inquiry shown in Figure 5.4. Generally

$$\sigma = 1/\sqrt{N} \quad ,$$

where the total uncertainty (or variance) of the data is given by

$$\sigma^2 = 1/N + c \quad , \quad \text{with } c > 0.$$

Here c is a measure of the systematic uncertainty. There are multiple suggestions to how exactly the systematic uncertainties of a NN should be estimated. Even though this part of the ML field is under rapid development, the methods are still under debate, and a unified consensus is still missing. A popular method includes MC dropout[43], but since we are not using dropout in our CNN (or other regularization techniques), this method is not suitable for our proposed model.

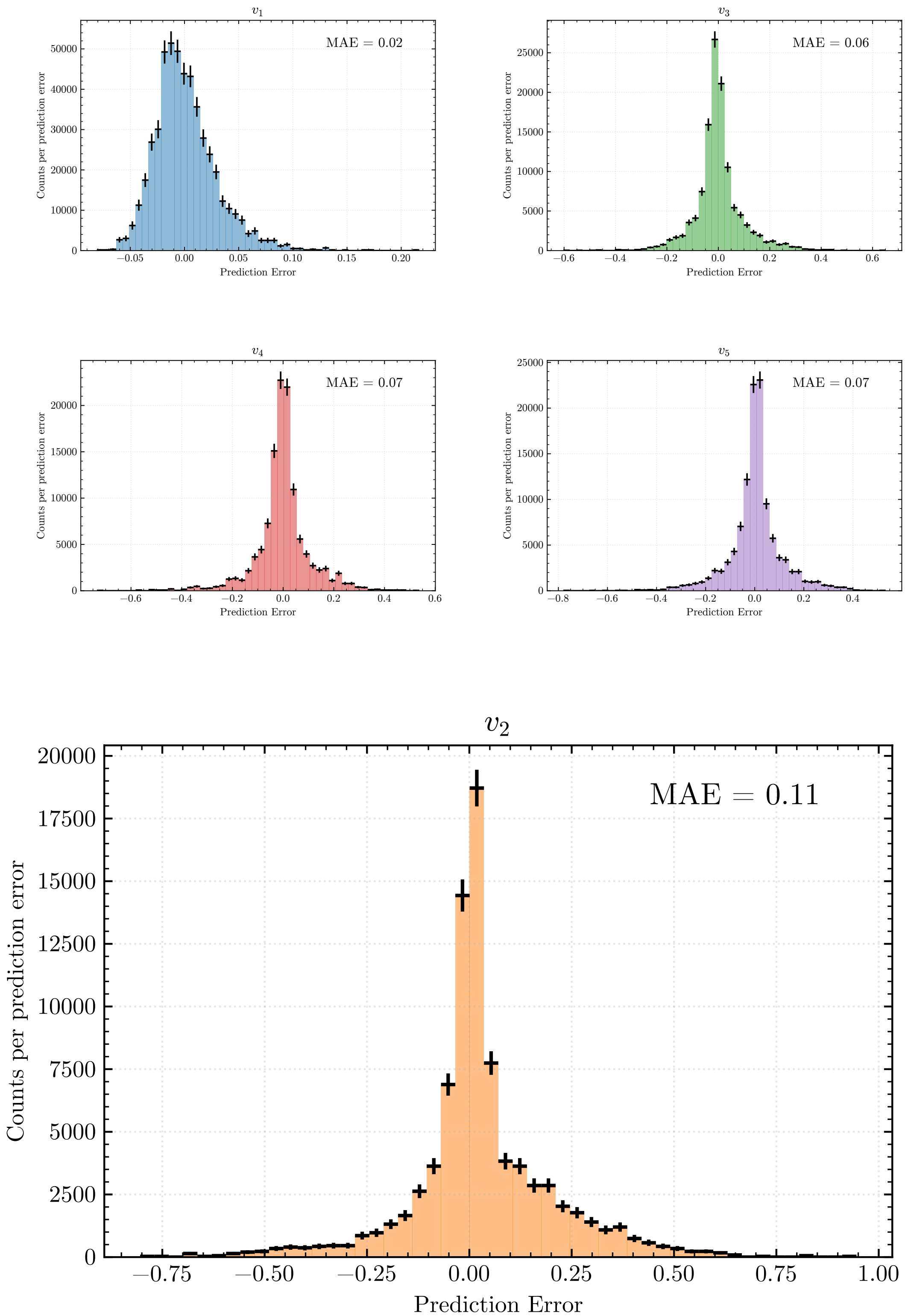


Figure 5.2. The residuals (difference between predicted and true values) for all five flow coefficients of 3.000 events. We remind the reader that the scales for each flow coefficient is different, e.g. $v_1 \in [0, 0.3]$, $v_3, v_4, v_5 \in [0, 1]$, and $v_2 \in [0, 2]$. Therefore, the MAE should be considered in respect to these ranges.

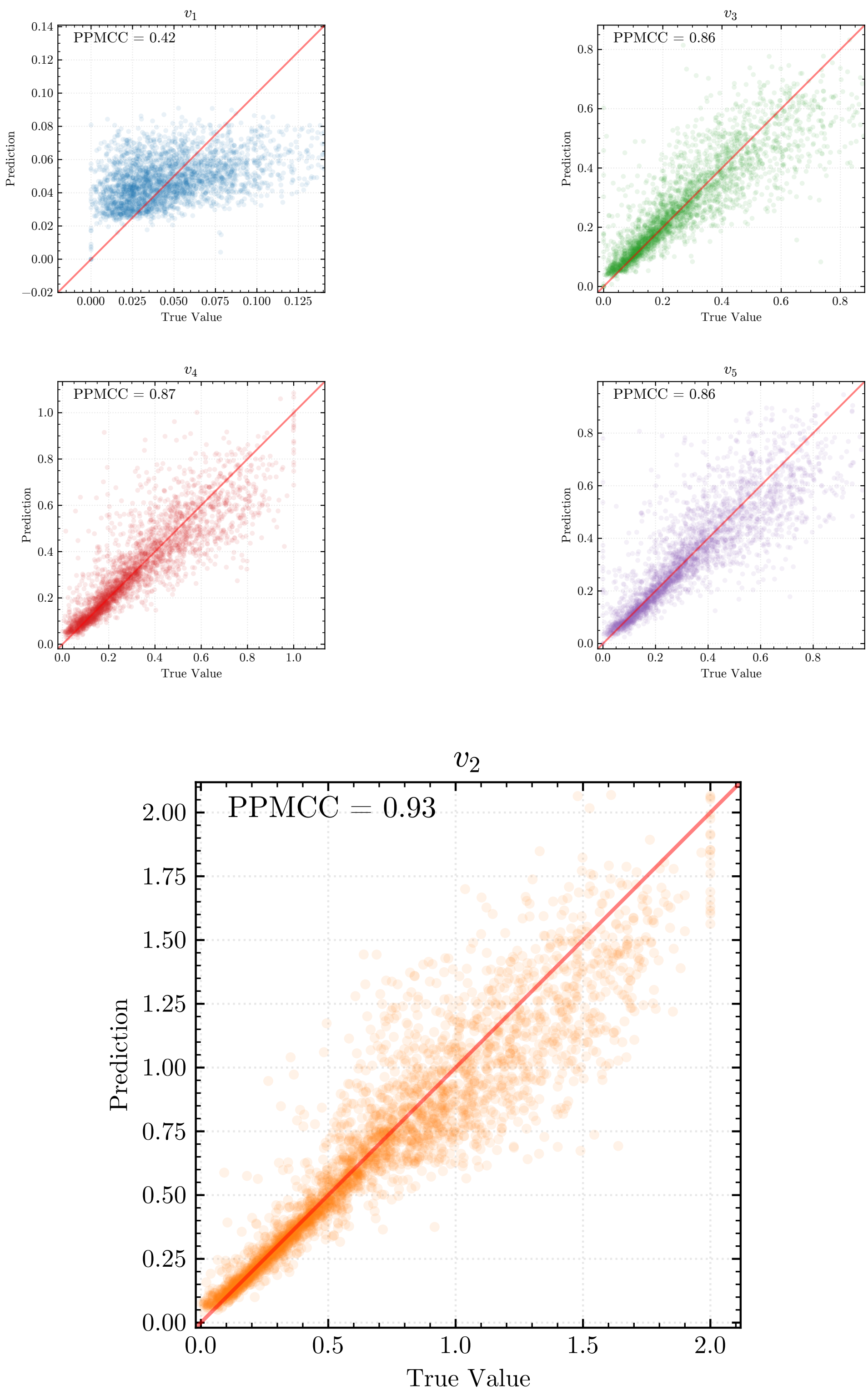


Figure 5.3. True versus predicted flow coefficients of 3.000 events, together with their respective PPMCC performance measures. The strongest correlation between the predicted and true values are for v_2 (orange dots), while the weakest are for v_1(blue dots).

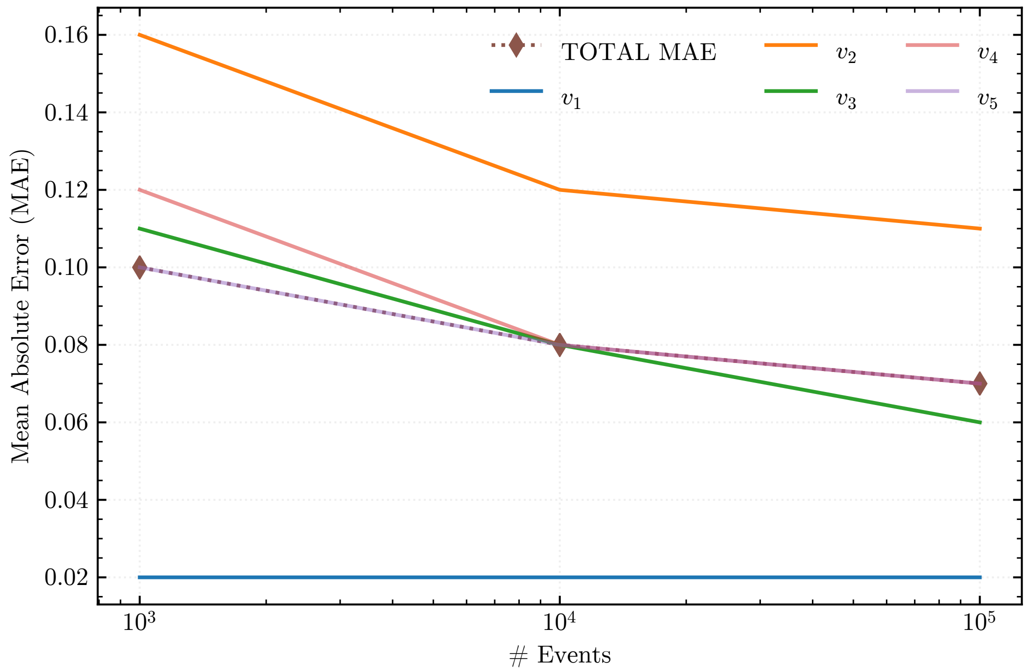


Figure 5.4. The Mean Absolute Error evolution for various sample sizes. The sample size does not affect the uncertainty on v_1, suggesting that the model is unable to capture its features.

An independent proposition is to take a sample of N different toy models (N different combinations of interaction profiles + cross-section profiles + ancestor models + particle production models) and train M *different* Neural Networks (NNs) on each of those. For this purpose, we test the M various NNs on a test set similar to the N toys used for training. This procedure results in a matrix with entries corresponding to each NNs performance error on the respective toy models. The systematic uncertainty is arguably the variation of the errors in this matrix. However, this would result in M *independent* proposed models, because each NN does not have the same access to the complete parameter space of the data. This proposition is consequently not suitable to estimate the systematic uncertainty of a distinct proposed model.

If we briefly return to the analogy of predicting house prices, we could imagine that we have built several different ML models where each of them has trained with houses from specific areas of neighborhoods in Denmark. One model only considered houses on Lolland, another one only on houses in central Jutland, and the last one in Northern Zealand. We cannot isolate a true systematic uncertainty of a "unified" house prediction model (even if the model parameters are the same) if we tested the models with houses from Funen, because each model has a bias towards its own data space. The model from Lolland may consistently underestimate, while the model from Nothern Zealand may consistently overestimate the house prices from Funen. In order to have a robust ML model and genuinely capture its bias (systematic uncertainty), it would be favorable to train on the entire data parameter space available. Therefore the optimal solution in this case is to train on a data set consisting of all 144 combinations of toy models and

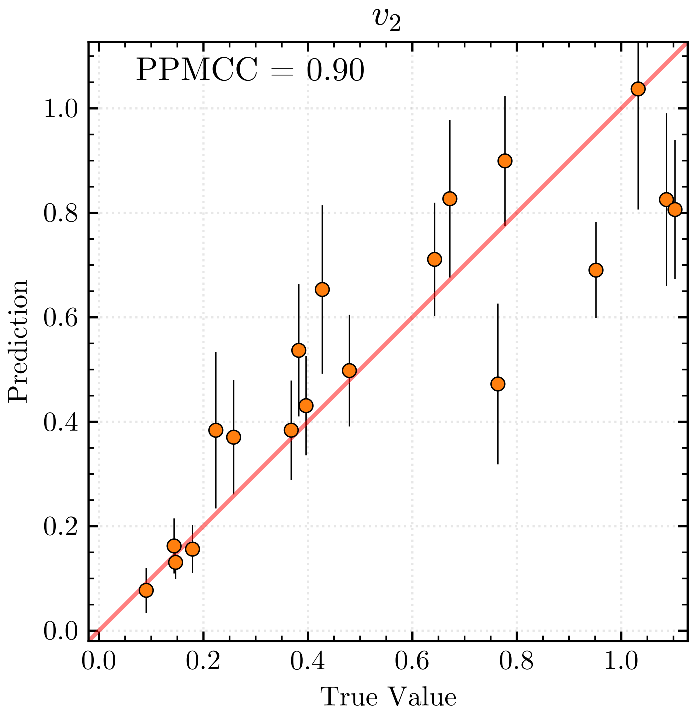


Figure 5.5. The prediction of 20 events (available for display in the appendix). The model predicts v_2 with a strong correlation, where it clearly estimates low v_2 with high accuracy and precision, whereas the uncertainty rises for higher v_2 (from semi-central and beyond). The same plot for v_1, v_3, v_4, v_5 can be found in Figure 5.6.

perhaps extend it further to consider various combinations of α, μ, and k on each of those. However, it is still possible to capture some systematic uncertainties within the chosen parameter space mentioned in the previous section.

We can estimate the systematic uncertainties by applying concepts defined under the Bayesian Neural Networks (BNNs) framework [44]. The idea of estimating weights as point predictions are instead replaced by drawing random variables from a probability distribution. In the Bayesian case, the initial weight distribution are the priors, and the objective is to update the priors by replacing them with the learned posteriors. The prior is usually Normal distribution. Thereby, the weights are incidentally random variables drawn from Normal distribution with a mean of zero and variance one $\mathcal{N}(0, 1)$. The learning process consists of updating the prior by shifting the mean and variance towards values that appropriately fit the data. We decided to incorporate the method of BNNs to estimate the systematic uncertainties (leaving further technicalities for future work) by replacing the first layer of the CNN with a so-called probabilistic CNN layer using the TensorFlow Probability library [45].

We present the results including systematic uncertainties in Figure 5.6 for v_1, v_3, v_4, v_5 and Figure 5.5 for v_2. The model predicts the flow coefficients from 20 random events, all

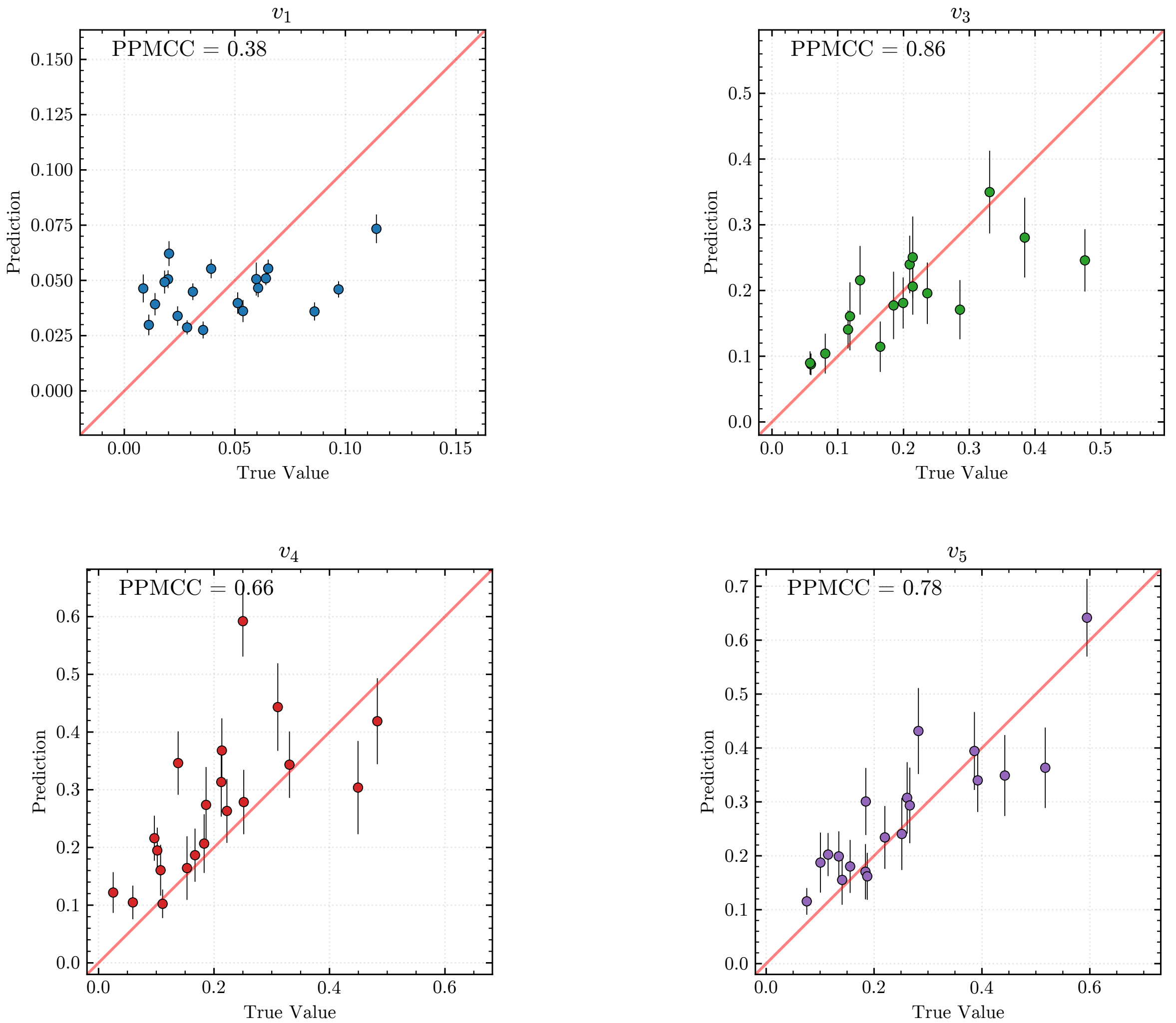


Figure 5.6. Estimated flow coefficients with systematic uncertainties characterized by replacing the first CNN layer with probabilistic features. It is clearly shown that the model is confident with v_1 yet the PPMCC is weakly correlated. Furthermore, the model seem to have a bias towards overestimating v_4. The same plot for v_2 can be found in Figure 5.5.

Event	v_1	$\hat{v}_1$	v_2	$\hat{v}_2$	v_3	$\hat{v}_3$	v_4	$\hat{v}_4$	v_5	$\hat{v}_5$
# 1	0.064	0.051 ± 0.004	0.777	0.89 ± 0.15	0.209	0.24 ± 0.05	0.331	0.34 ± 0.06	0.392	0.33 ± 0.07
# 2	0.039	0.055 ± 0.004	1.372	1.00 ± 0.16	0.384	0.28 ± 0.06	0.483	0.40 ± 0.07	0.442	0.34 ± 0.07
# 3	0.028	0.029 ± 0.003	0.147	0.15 ± 0.05	0.059	0.09 ± 0.02	0.111	0.11 ± 0.03	0.075	0.12 ± 0.03
# 4	0.114	0.072 ± 0.006	1.032	1.07 ± 0.19	0.663	0.50 ± 0.06	0.250	0.58 ± 0.06	0.595	0.63 ± 0.06
# 5	0.020	0.050 ± 0.004	1.087	0.83 ± 0.14	0.476	0.24 ± 0.04	0.138	0.33 ± 0.06	0.517	0.35 ± 0.07
# 6	0.020	0.062 ± 0.005	0.224	0.40 ± 0.12	0.626	0.47 ± 0.07	0.310	0.45 ± 0.06	0.282	0.43 ± 0.06
# 7	0.014	0.040 ± 0.006	0.259	0.39 ± 0.12	0.286	0.18 ± 0.05	0.183	0.21 ± 0.06	0.251	0.24 ± 0.06
# 8	0.051	0.040 ± 0.006	0.479	0.51 ± 0.10	0.185	0.18 ± 0.05	0.102	0.20 ± 0.04	0.135	0.20 ± 0.04
# 9	0.097	0.045 ± 0.003	1.103	0.81 ± 0.13	0.199	0.17 ± 0.03	0.222	0.26 ± 0.04	0.220	0.22 ± 0.05
# 10	0.054	0.037 ± 0.005	0.368	0.40 ± 0.11	0.119	0.17 ± 0.06	0.107	0.16 ± 0.04	0.101	0.19 ± 0.05
# 11	0.011	0.030 ± 0.005	0.179	0.16 ± 0.05	0.081	0.11 ± 0.03	0.025	0.12 ± 0.03	0.184	0.17 ± 0.05
# 12	0.036	0.028 ± 0.003	0.144	0.16 ± 0.06	0.058	0.09 ± 0.02	0.059	0.10 ± 0.03	0.141	0.15 ± 0.04
# 13	0.065	0.055 ± 0.003	0.672	0.80 ± 0.14	0.331	0.34 ± 0.06	0.214	0.36 ± 0.05	0.386	0.37 ± 0.06
# 14	0.018	0.048 ± 0.005	0.428	0.62 ± 0.14	0.214	0.24 ± 0.06	0.213	0.30 ± 0.06	0.185	0.28 ± 0.06
# 15	0.031	0.044 ± 0.004	0.952	0.70 ± 0.11	0.236	0.19 ± 0.05	0.097	0.21 ± 0.04	0.115	0.20 ± 0.04
# 16	0.086	0.036 ± 0.003	0.396	0.42 ± 0.10	0.116	0.14 ± 0.03	0.167	0.19 ± 0.04	0.156	0.18 ± 0.05
# 17	0.009	0.046 ± 0.005	0.383	0.53 ± 0.13	0.134	0.21 ± 0.05	0.186	0.28 ± 0.06	0.261	0.29 ± 0.07
# 18	0.024	0.033 ± 0.004	0.090	0.08 ± 0.04	0.165	0.11 ± 0.03	0.153	0.16 ± 0.05	0.187	0.16 ± 0.04
# 19	0.060	0.050 ± 0.007	0.764	0.47 ± 0.15	0.625	0.25 ± 0.07	0.449	0.30 ± 0.06	0.767	0.38 ± 0.08
# 20	0.060	0.047 ± 0.004	0.643	0.71 ± 0.11	0.214	0.21 ± 0.04	0.252	0.29 ± 0.05	0.266	0.29 ± 0.06

Table 5.1. The true flow coefficients v_n side by side with the predicted $\hat{v}_n$ from Figure 5.5 and 5.6. The actual events can be found in appendix E.

originating from a black disc interaction profile and fixed cross-section. We see a strong correlation between the uncertainties and the structure available in the images being analyzed[1], i.e., the model is confident (estimating low uncertainties) for lower values of v_n since these correspond to $\frac{d^2N_{ch}}{d\eta d\varphi}$ images with lots of structure. These structure rich images are correlated with central collisions producing a high number of particles.

A critic of the proposed model is whether it captures the multiplicity of particles instead of the actual patterns of the densities, and thereby correlating the eccentricities ε_n with the multiplicity of charge particles, instead of their distribution in (η, φ). It is not unreasonable to assume this dependency since the eccentricities are zero for impact parameters equal zero $\varepsilon_n \approx 0 \rightarrow b \approx 0$ and monotonically rising with the impact parameter. As the impact parameter *rises* multiplicities *decreases*. We account for this dependency with the pre-processing induced on the original images. We remind the reader that we normalize each picture with the maximum pixel value of the image in the pre-processing, such that the highest value of any event is equal to unity. In this regard, we ensure that the model is independent of the multiplicity of the event.

Another observation is the model's inadequacy of predicting v_1. This observation is presumably linked to how v_1 has zero to no structure as viewed in Figure , combined with its suppressed values by the factor $\kappa_1 = 0.5$. Because the systematic uncertainties are small for v_1 it implies that the error is subject large statistical uncertainties. It is therefore questionable if the construction of a more complex model would benefit the estimated $\hat{v}_1$.

[1] The events can be found in appendix E.

6

Conclusion

We began this thesis by asking whether it would be possible to create a Deep Learning (DL) model capable of predicting the flow coefficients v_n solely from the final state densities of charged particles $\frac{\mathrm{d}^2 N_{\mathrm{ch}}}{\mathrm{d}\eta \mathrm{d}\varphi}$. To investigate this, we created a controlled experiment by simulating nuclei collisions with the Glauber model, and their products by implementing particle production models. The experiments lead to a variety of toy models, all having specific assumptions on how particles collide, interact, and form. Nonetheless, all the models capture the same phenomenology, i.e., they all have in common that the initial state geometry ε_n depends on the impact parameter b, and that all the charged particles are distributed in the same fashion. Still, the various toy models and their flexibility allowed us the possibility of assembling a diverse data set, suitable for training a DL model.

We constructed a DL model in the form of a Convolutional Neural Network (CNN). The considerations concerning building the model were two-folded. The ambition was a versatile and simple model capable of capturing features regardless of the underlying toy models or assumptions. The task of decreasing the number of parameters and still achieved good performance was accomplished by fine-tuning particular hyper parameters. We found that the proposed model can estimate the flow coefficient event by event, with high accuracy and precision for central and semi-central collisions. The performance of the proposed model regarding the analysis of 3.000 minimum bias Pb-Pb events with a black disc interaction profile and fixed cross-section of 70 mb yielded a Pearson product-moment correlation coefficient (PPMCC) of 0.42 for directed flow v_1, 0.90 for elliptical v_2, 0.86 for triangular v_3 and quintuple v_5, and 0.87 for the quadruple v_4 flow coefficients.

6.1 Outlook

This thesis's work can be considered a technical approach to the problem of estimating flow coefficients and lays the initial groundwork for the possibility of assisting already established methods such as the Generic Framework [46] (GF). Hence, the prospective work might include how this model compares with GF and how they can complement each other. Other prospective work includes the reduction of the statistical and systematic errors. For this reason, it would be intriguing to investigate the (close to) full parameter space of the various combinations of toy models.

Bibliography

[1] H. Niemi et al. "Event-by-event distributions of azimuthal asymmetries in ultrarelativistic heavy-ion collisions". In: *Phys. Rev. C* 87 (5 May 2013), p. 054901. DOI: 10.1103/PhysRevC.87.054901. URL: https://link.aps.org/doi/10.1103/PhysRevC.87.054901.

[2] S. Acharya et al. "Investigations of Anisotropic Flow Using Multiparticle Azimuthal Correlations in pp, $p-$Pb, Xe-Xe, and Pb-Pb Collisions at the LHC". In: *Phys. Rev. Lett.* 123 (14 Oct. 2019), p. 142301. DOI: 10.1103/PhysRevLett.123.142301. URL: https://link.aps.org/doi/10.1103/PhysRevLett.123.142301.

[3] CERN. *The Large Hadron Collider*. [Accessed on August 21, 2019]. c2019. URL: https://home.cern/topics/large-hadron-collider.

[4] Markus Ahlers, Klaus Helbing, and Carlos Pérez de los Heros. "Probing particle physics with IceCube". In: *The European Physical Journal C* 78.11 (2018), p. 924. DOI: 10.1140/epjc/s10052-018-6369-9. URL: https://doi.org/10.1140/epjc/s10052-018-6369-9.

[5] Wladyslaw Henryk Trzaska. "New ALICE detectors for Run 3 and 4 at the CERN LHC". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* (2019). ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2019.04.070. URL: http://www.sciencedirect.com/science/article/pii/S0168900219305443.

[6] B Abelev et al and. "Upgrade of the ALICE Experiment: Letter Of Intent". In: *Journal of Physics G: Nuclear and Particle Physics* 41.8 (July 2014), p. 087001. DOI: 10.1088/0954-3899/41/8/087001. URL: https://doi.org/10.1088%2F0954-3899%2F41%2F8%2F087001.

[7] A. I. Maevskaya. "Fast Interaction Trigger for the upgrade of the ALICE experiment at CERN: design and performance". In: *EPJ Web Conf.* 204 (2019), p. 11003. DOI: 10.1051/epjconf/201920411003. arXiv: 1812.00594 [physics.ins-det].

[8] Dan Guest, Kyle Cranmer, and Daniel Whiteson. "Deep Learning and Its Application to LHC Physics". In: *Annual Review of Nuclear and Particle Science* 68.1 (2018), pp. 161–181. DOI: 10.1146/annurev-nucl-101917-021019. eprint: https://doi.org/10.1146/annurev-nucl-101917-021019. URL: https://doi.org/10.1146/annurev-nucl-101917-021019.

[9] Luke de Oliveira et al. “Jet-images — deep learning edition”. In: *Journal of High Energy Physics* 2016.7 (July 2016). ISSN: 1029-8479. DOI: 10.1007/jhep07(2016)069. URL: http://dx.doi.org/10.1007/JHEP07(2016)069.

[10] A. Schwartzman et al. “Image Processing, Computer Vision, and Deep Learning: new approaches to the analysis and physics interpretation of LHC events”. In: *Journal of Physics: Conference Series* 762 (Oct. 2016), p. 012035. DOI: 10.1088/1742-6596/762/1/012035. URL: https://doi.org/10.1088%2F1742-6596%2F762%2F1%2F012035.

[11] Edwin Hubble. “A relation between distance and radial velocity among extra-galactic nebulae”. In: *Proceedings of the National Academy of Sciences* 15.3 (1929), pp. 168–173. ISSN: 0027-8424. DOI: 10.1073/pnas.15.3.168. eprint: https://www.pnas.org/content/15/3/168.full.pdf. URL: https://www.pnas.org/content/15/3/168.

[12] A. W. Jones and A. N. Lasenby. “The Cosmic Microwave Background”. In: *Living Reviews in Relativity* 1.1 (1998), p. 11. DOI: 10.12942/lrr-1998-11. URL: https://doi.org/10.12942/lrr-1998-11.

[13] John F. Donoghue, Eugene Golowich, and Barry R. Holstein. *Dynamics of the Standard Model*. 2nd ed. Cambridge Monographs on Particle Physics, Nuclear Physics and Cosmology. Cambridge University Press, 2014. DOI: 10.1017/CBO9780511803512.

[14] Lubej Matic. URL: http://www-f9.ijs.si/~lubej/.

[15] John C Mather et al. “Measurement of the cosmic microwave background spectrum by the COBE FIRAS instrument”. In: *The Astrophysical Journal* 420 (1994), pp. 439–444.

[16] K. Adcox et al. “Formation of dense partonic matter in relativistic nucleus–nucleus collisions at RHIC: Experimental evaluation by the PHENIX Collaboration”. In: *Nuclear Physics A* 757.1-2 (Aug. 2005), pp. 184–283. ISSN: 0375-9474. DOI: 10.1016/j.nuclphysa.2005.03.086. URL: http://dx.doi.org/10.1016/j.nuclphysa.2005.03.086.

[17] S. Voloshin and Y. Zhang. “Flow study in relativistic nuclear collisions by Fourier expansion of azimuthal particle distributions”. In: *Zeitschrift fur Physik C Particles and Fields* 70.4 (May 1996), pp. 665–671. ISSN: 1431-5858. DOI: 10.1007/s002880050141. URL: http://dx.doi.org/10.1007/s002880050141.

[18] Peter F. Kolb and Ulrich Heinz. *Hydrodynamic description of ultrarelativistic heavy-ion collisions*. 2003. arXiv: nucl-th/0305084 [nucl-th].

[19] Michael L. Miller et al. “Glauber Modeling in High-Energy Nuclear Collisions”. In: *Annual Review of Nuclear and Particle Science* 57.1 (Nov. 2007), pp. 205–243. ISSN: 1545-4134. DOI: 10.1146/annurev.nucl.57.090506.123020. URL: http://dx.doi.org/10.1146/annurev.nucl.57.090506.123020.

[20] Christian Holm Christensen & Zlatko Šaldić. *Glauber Monte Carlo Simulation - Initial conditions in nucleus-nucleus collisions*. 2020. URL: https://cholmcc.gitlab.io/nbi-python/mechanics/Glauber.html.

[21] T. A. Minelli, A. Pascolini, and C. Villi. "Shell model potentials for finite nuclei". In: *Il Nuovo Cimento A (1965-1970)* 90.2 (Oct. 2007), p. 185. ISSN: 1826-9869. DOI: 10.1007/BF02724231. URL: https://doi.org/10.1007/BF02724231.

[22] Maciej Rybczyński and Zbigniew Włodarczyk. "The nucleon–nucleon collision profile and cross section fluctuations". In: *Journal of Physics G: Nuclear and Particle Physics* 41.1 (Dec. 2013), p. 015106. ISSN: 1361-6471. DOI: 10.1088/0954-3899/41/1/015106. URL: http://dx.doi.org/10.1088/0954-3899/41/1/015106.

[23] M. Alvioli and M. Strikman. "Color fluctuation effects in proton–nucleus collisions". In: *Physics Letters B* 722.4-5 (May 2013), pp. 347–354. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2013.04.042. URL: http://dx.doi.org/10.1016/j.physletb.2013.04.042.

[24] B. Alver and G. Roland. *Collision geometry fluctuations and triangular flow in heavy-ion collisions*. 2010. arXiv: 1003.0194 [nucl-th].

[25] R. E. Ansorge et al. "Charged particle multiplicity distributions at 200 and 900 GeV c.m. energy". In: *Zeitschrift für Physik C Particles and Fields* 43.3 (1989), pp. 357–374. DOI: 10.1007/BF01506531. URL: https://doi.org/10.1007/BF01506531.

[26] S. S. Adler et al. "Nuclear effects on hadron production ind+Aucollisions at-sNN=200GeV revealed by comparison withp+pdata". In: *Physical Review C* 74.2 (Aug. 2006). ISSN: 1089-490X. DOI: 10.1103/physrevc.74.024904. URL: http://dx.doi.org/10.1103/PhysRevC.74.024904.

[27] J. Adam et al. "Centrality dependence of the pseudorapidity density distribution for charged particles in Pb–Pb collisions atsNN=5.02 TeV". In: *Physics Letters B* 772 (Sept. 2017), pp. 567–577. ISSN: 0370-2693. DOI: 10.1016/j.physletb.2017.07.017. URL: http://dx.doi.org/10.1016/j.physletb.2017.07.017.

[28] Christian Holm Christensen. *Statistics Overview with Python*. Chapter 8, page 97–99. 2020. URL: https://cholmcc.gitlab.io/nbi-python/statistics/Statistik_en.pdf.

[29] S. Acharya et al. "Anisotropic flow in Xe–Xe collisions at sNN=5.44 TeV". In: *Physics Letters B* 784 (2018), pp. 82–95. ISSN: 0370-2693. DOI: https://doi.org/10.1016/j.physletb.2018.06.059. URL: http://www.sciencedirect.com/science/article/pii/S037026931830515X.

[30] STAR Collaboration et al. *Fluctuations of charge separation perpendicular to the event plane and local parity violation in sqrt(sNN)=200 GeV Au+Au collisions at RHIC*. 2013. arXiv: 1302.3802 [nucl-ex].

[31] Kenneth Aamodt et al. "The ALICE experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08002.

[32] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. "Multilayer feedforward networks are universal approximators". In: *Neural Networks* 2.5 (1989), pp. 359–366. ISSN: 0893-6080. DOI: https://doi.org/10.1016/0893-6080(89)90020-8. URL: http://www.sciencedirect.com/science/article/pii/0893608089900208.

[33] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. MIT Press, 2016.

[34] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2014. arXiv: 1412.6980 [cs.LG].

[35] Stéphane Lathuilière et al. *A Comprehensive Analysis of Deep Regression*. 2018. arXiv: 1803.08450 [cs.CV].

[36] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net*. 2014. arXiv: 1412.6806 [cs.LG].

[37] Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition. *Convolutional Neural Networks (CNNs / ConvNets)*. [Visited on May 21, 2020]. 2020. URL: https://cs231n.github.io/convolutional-networks/.

[38] Martìn Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[39] François Chollet et al. *Keras*. https://keras.io. 2015.

[40] Xavier Glorot and Y. Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Journal of Machine Learning Research - Proceedings Track* 9 (Jan. 2010), pp. 249–256.

[41] Tom O'Malley et al. *Keras Tuner*. https://github.com/keras-team/keras-tuner. 2019.

[42] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444. DOI: 10.1038/nature14539. URL: https://doi.org/10.1038/nature14539.

[43] Yarin Gal and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning". In: *international conference on machine learning*. 2016, pp. 1050–1059.

[44] Kumar Shridhar, Felix Laumann, and Marcus Liwicki. *A Comprehensive guide to Bayesian Convolutional Neural Network with Variational Inference*. 2019. arXiv: 1901.02731 [cs.LG].

[45] Joshua V. Dillon et al. *TensorFlow Distributions*. 2017. arXiv: 1711.10604 [cs.LG].

[46] Ante Bilandzic et al. "Generic framework for anisotropic flow analyses with multiparticle azimuthal correlations". In: *Physical Review C* 89.6 (June 2014). ISSN: 1089-490X. DOI: 10.1103/physrevc.89.064904. URL: http://dx.doi.org/10.1103/PhysRevC.89.064904.

List of Figures

A

Nuclear charge distributions

There exists other possibilities of nuclear charge distributions besides the three-parameter Eerhart-Fermi distribution that was mainly used in this thesis. These distributions include the three-parameter Taper-Gaussian [21] charge distribution

$$f(r) = r^2 \frac{1 + W r^2/R^2}{1 + e^{\frac{r^2 - R^2}{A^2}}} \quad ,$$

the Hulthen [26] charge distribution which is in particular relevant for deuterons

$$f(r; R, A) = r^2 \frac{RA(R + A)}{2\pi (R - A)^2} \left(e^{-Rr} - e^{-Ar} \right)^2 \quad ,$$

and different variations of proton charge distributions

$$f(r; R) = r^2 \begin{cases} e^{-r/R} \\ e^{-r^2/(2R^2)} \\ \frac{(1-a)}{R^3} e^{-r^2/R^2} + \frac{a}{(0.4R)^3} e^{-r^2/(0.4R)^2} \end{cases} \quad .$$

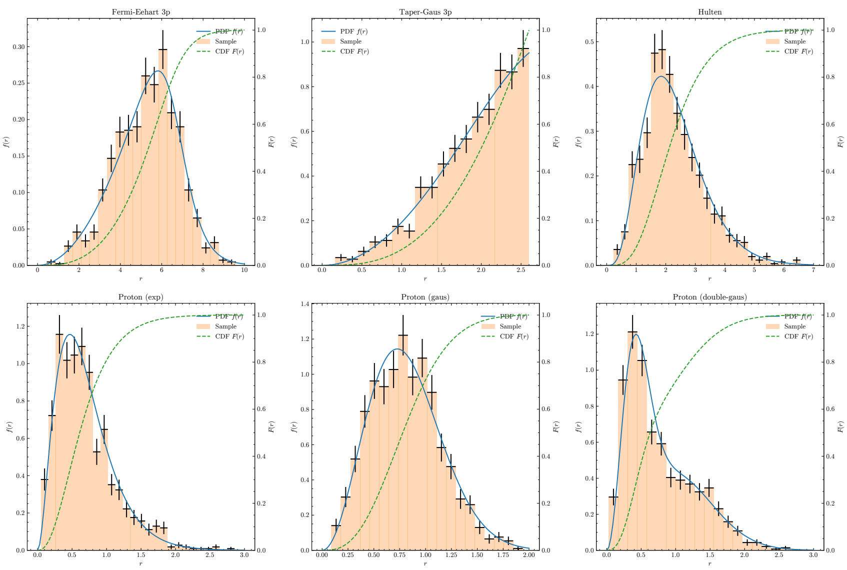


Figure A.1. Various types of nuclear charge density distributions.

B

Event summaries

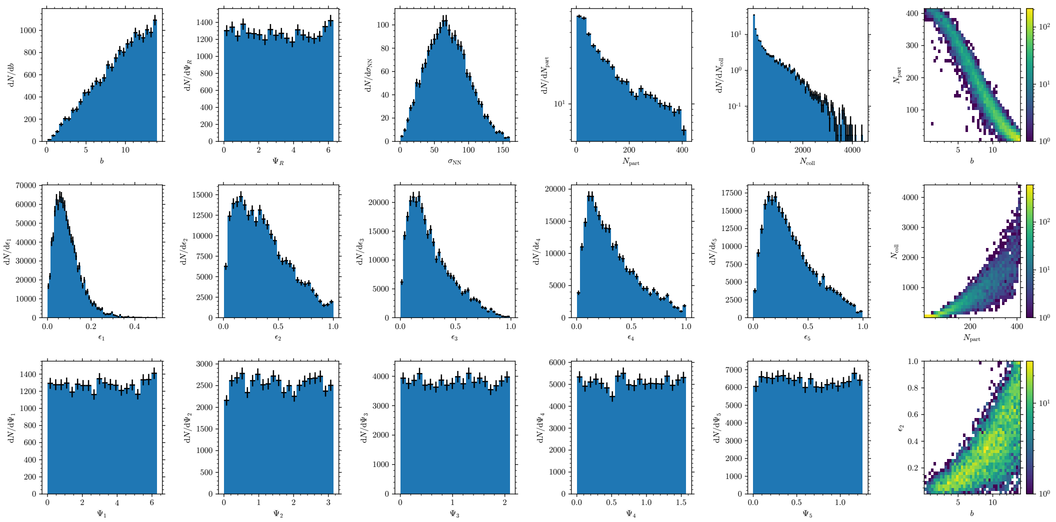


Figure B.1. Black disc profile and Gribov fluctuations

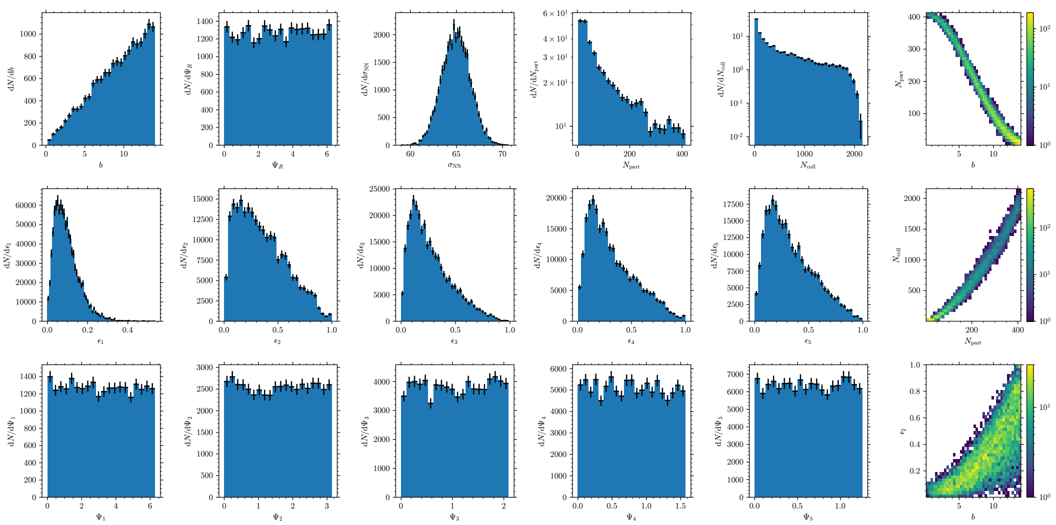


Figure B.2. Black disc profile and color fluctuations

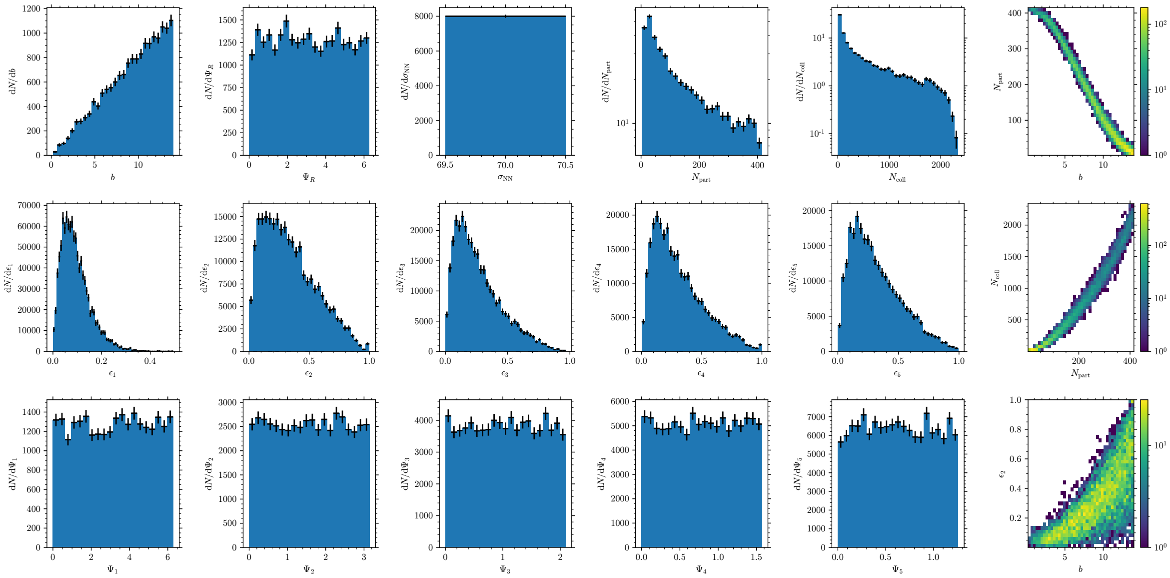


Figure B.3. Defuse interaction profile and fixed fluctuations

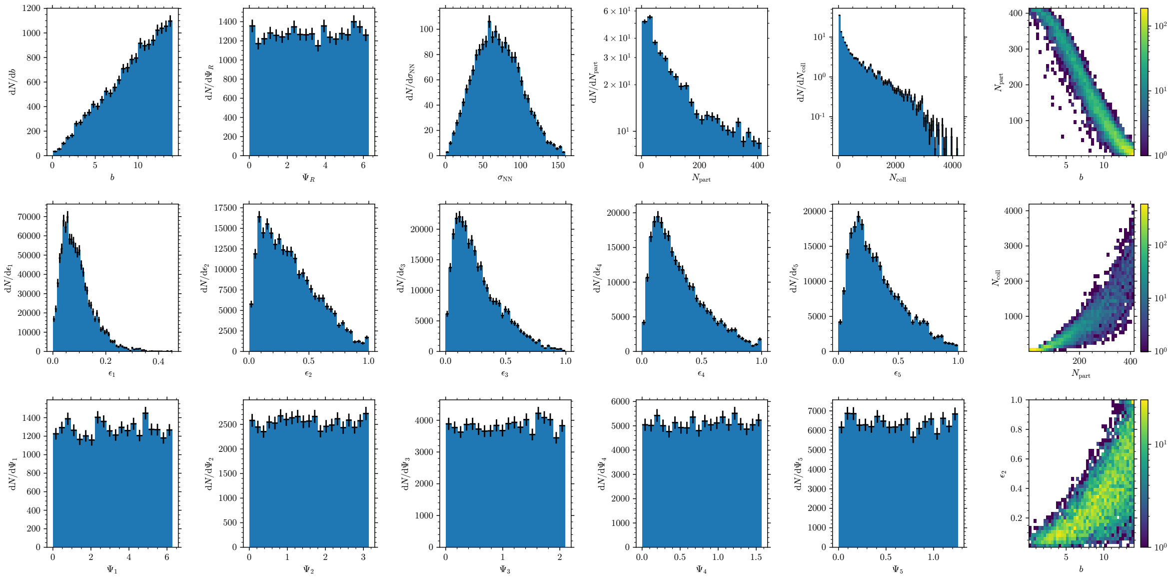


Figure B.4. Defuse interaction profile and Gribov fluctuations

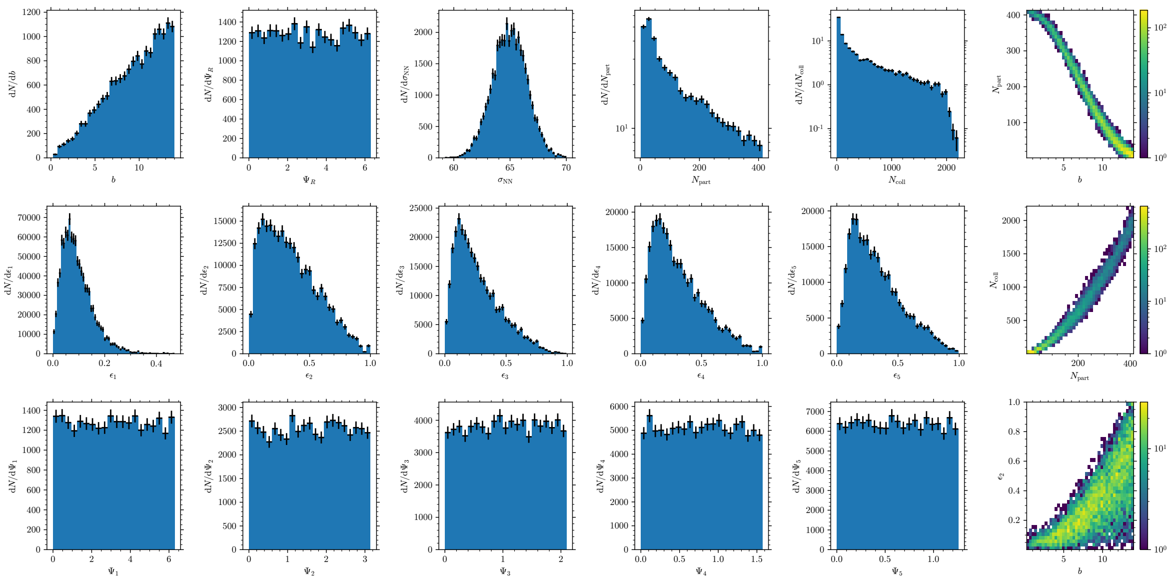


Figure B.5. Defuse interaction profile and color fluctuations

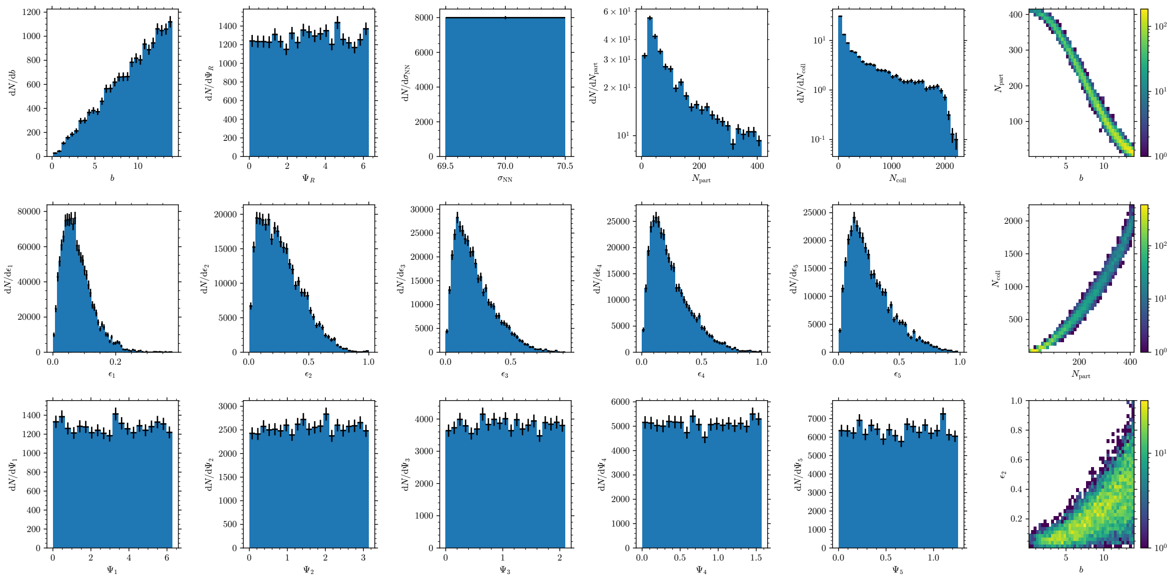


Figure B.6. Normal interaction profile and fixed fluctuations

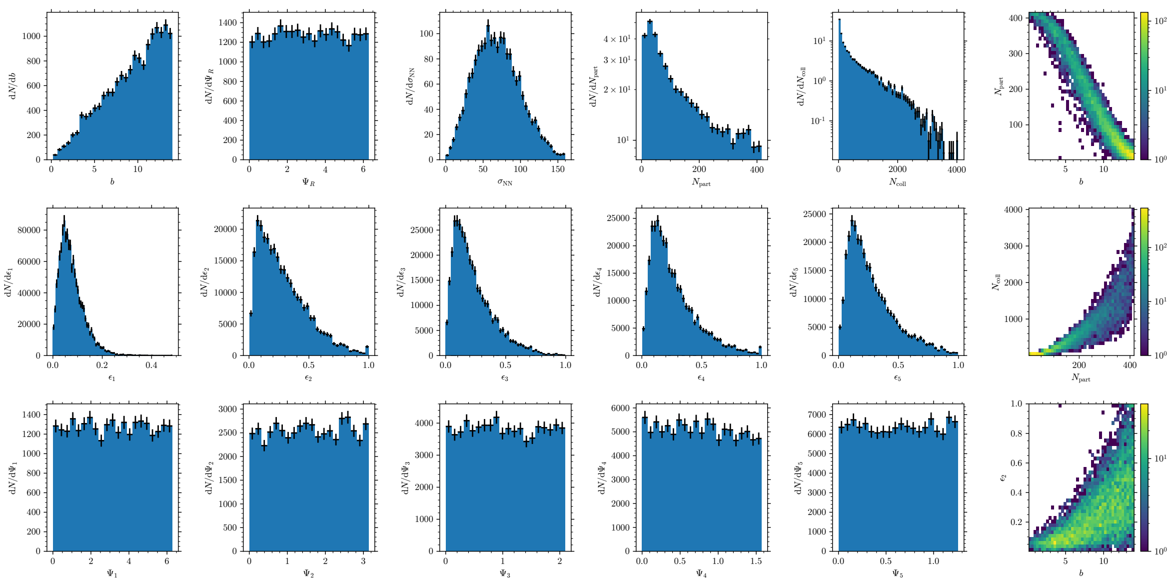


Figure B.7. Normal interaction profile and Gribov fluctuations

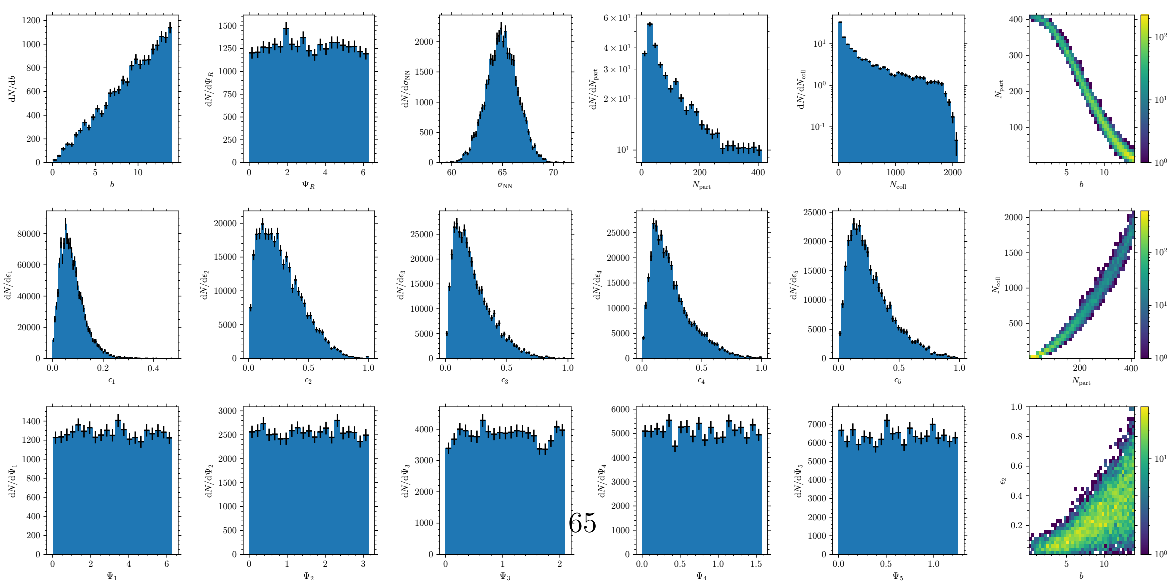


Figure B.8. Normal interaction profile and color fluctuations

C

Eccentricities versus N_{part}

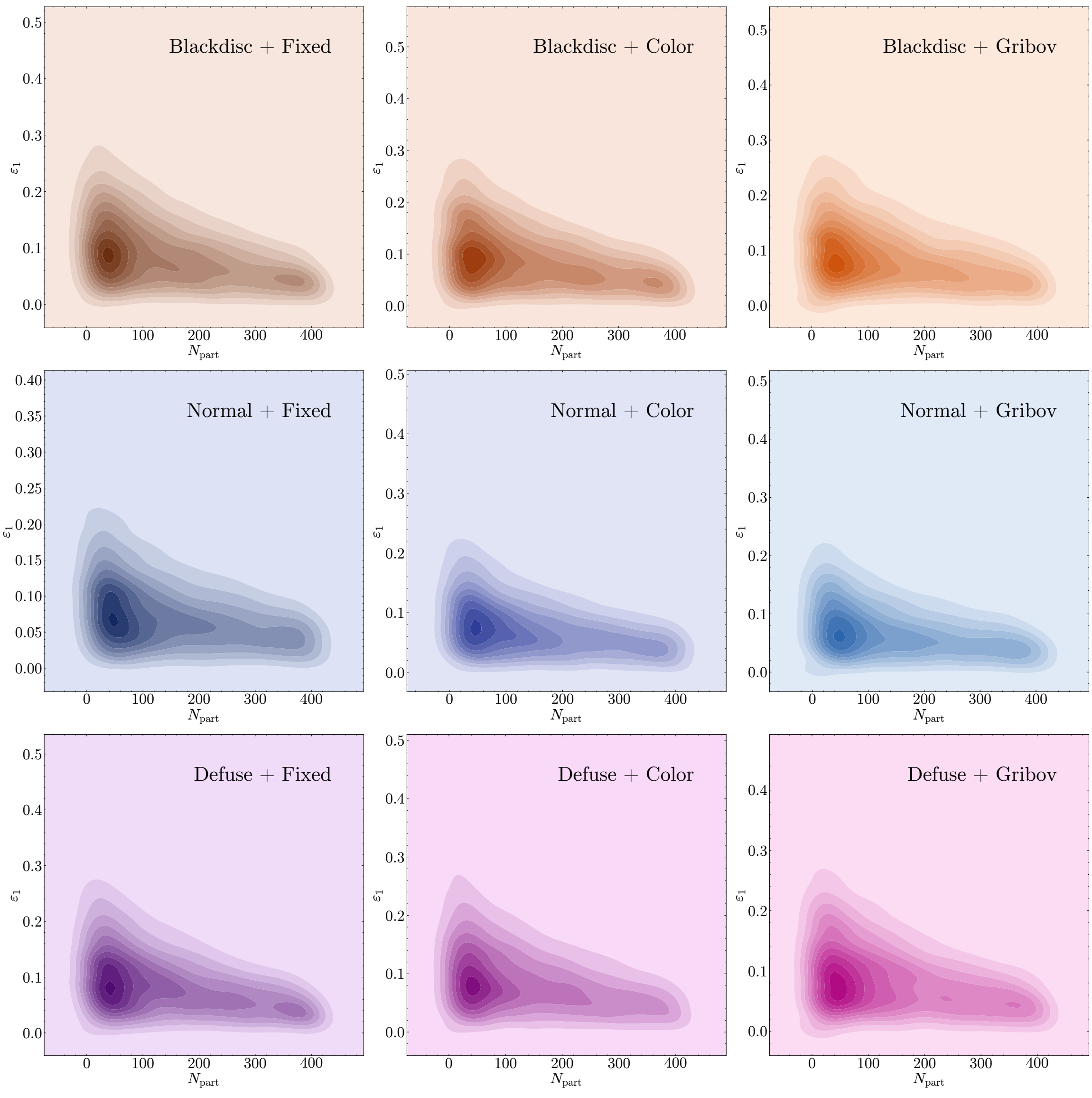


Figure C.1. ε_1 vs N_{part}

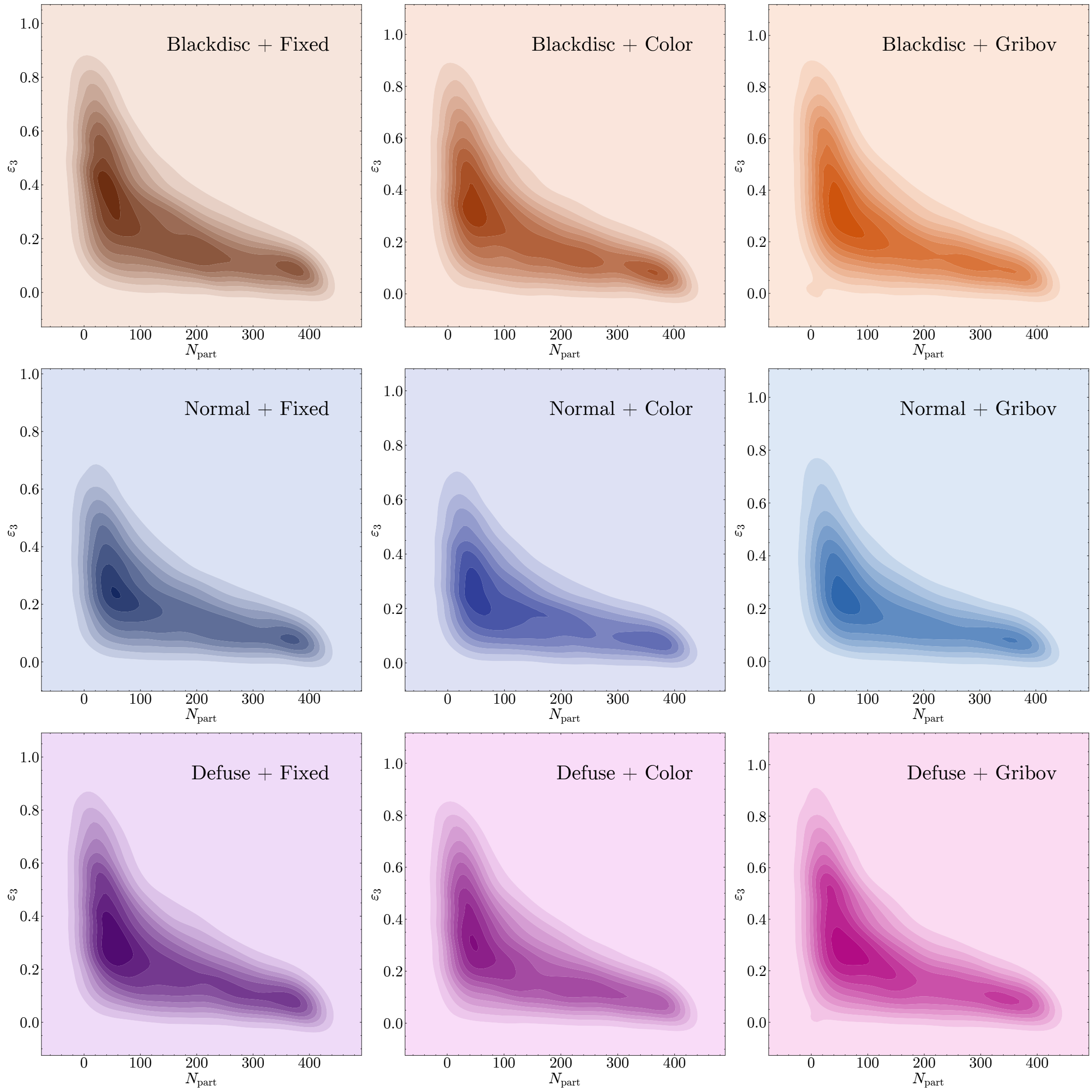


Figure C.2. ε_3 vs N_{part}

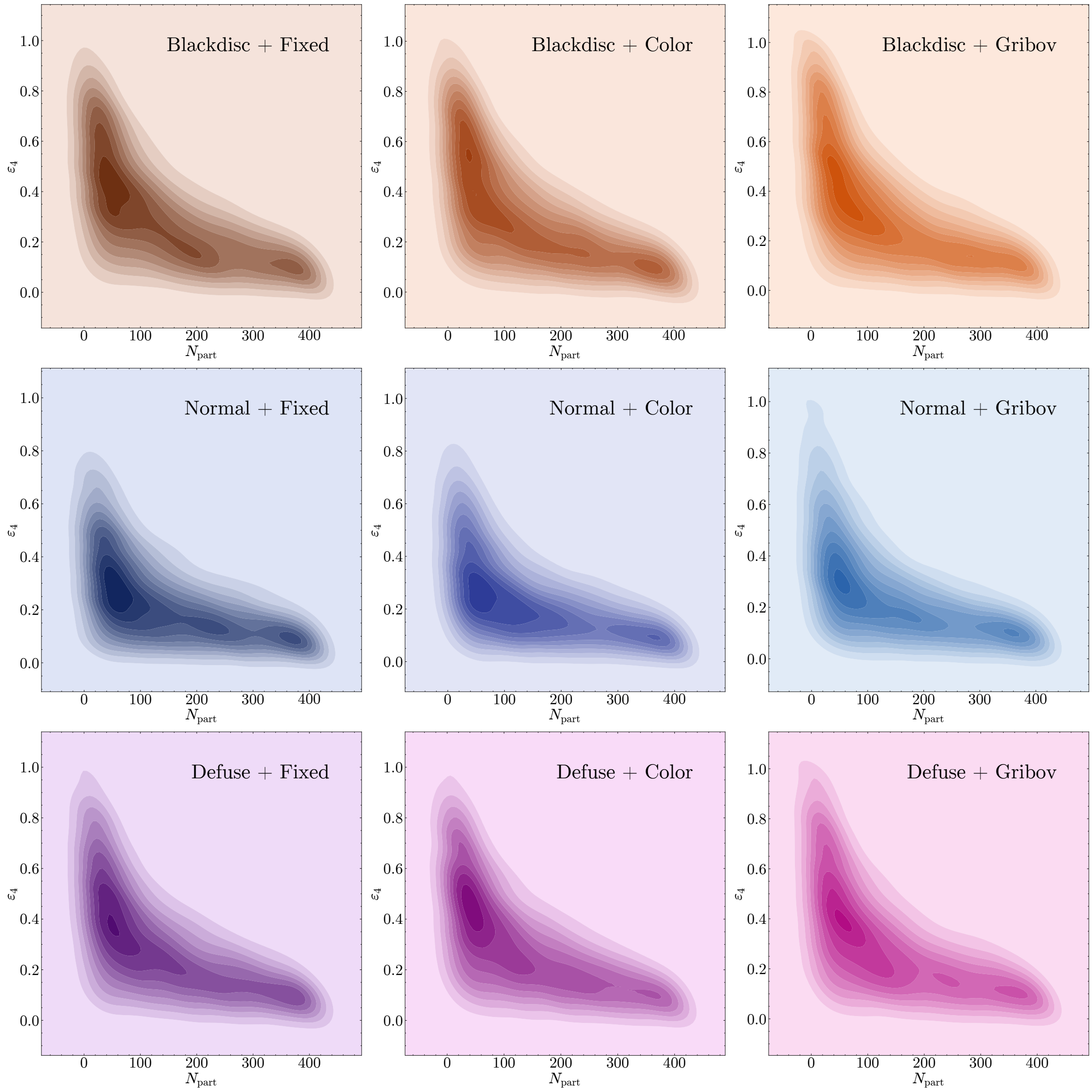


Figure C.3. ε_4 vs $N_{\rm part}$

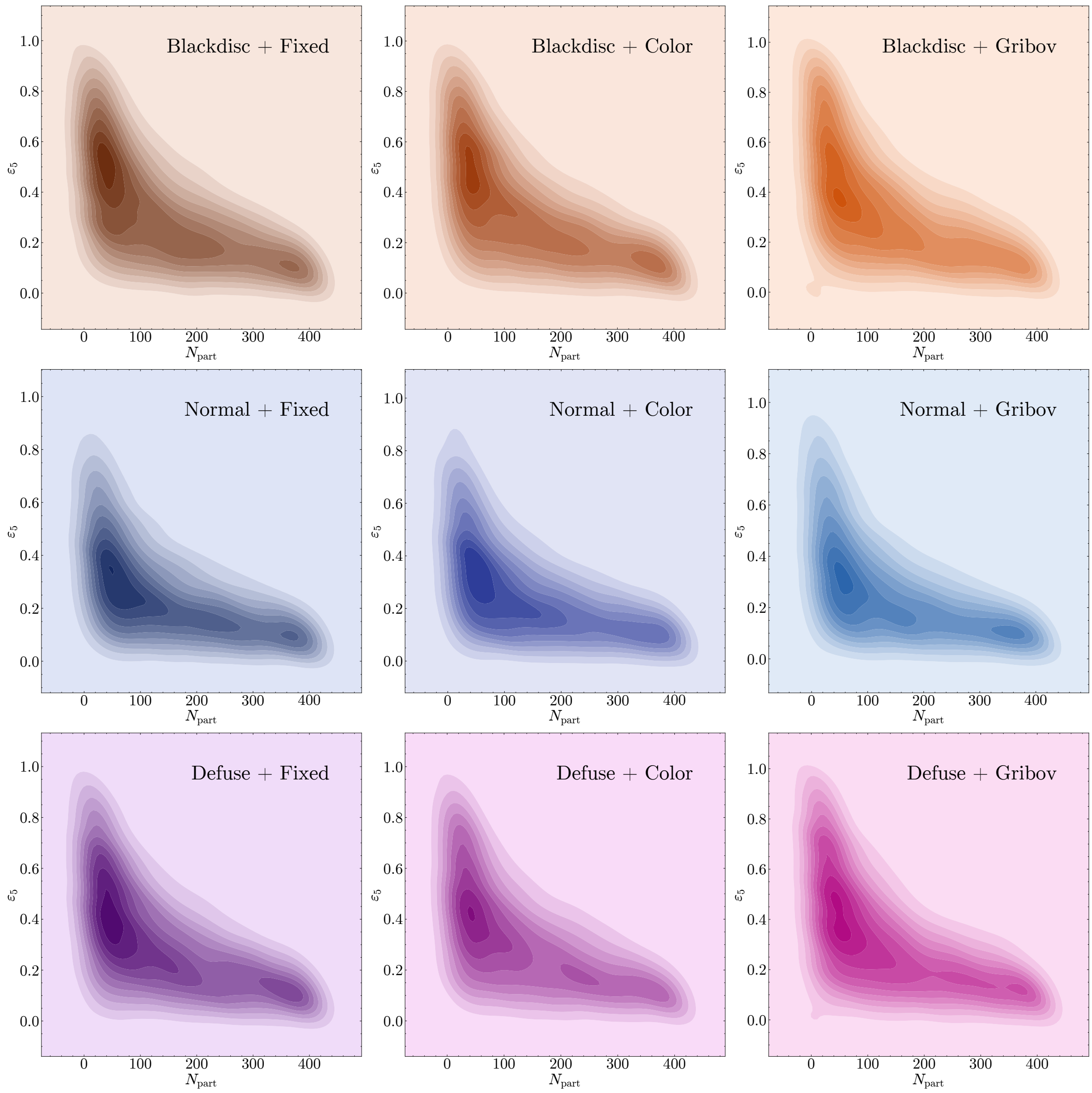


Figure C.4. ε_5 vs N_{part}

D

Eccentricities versus N_{coll}

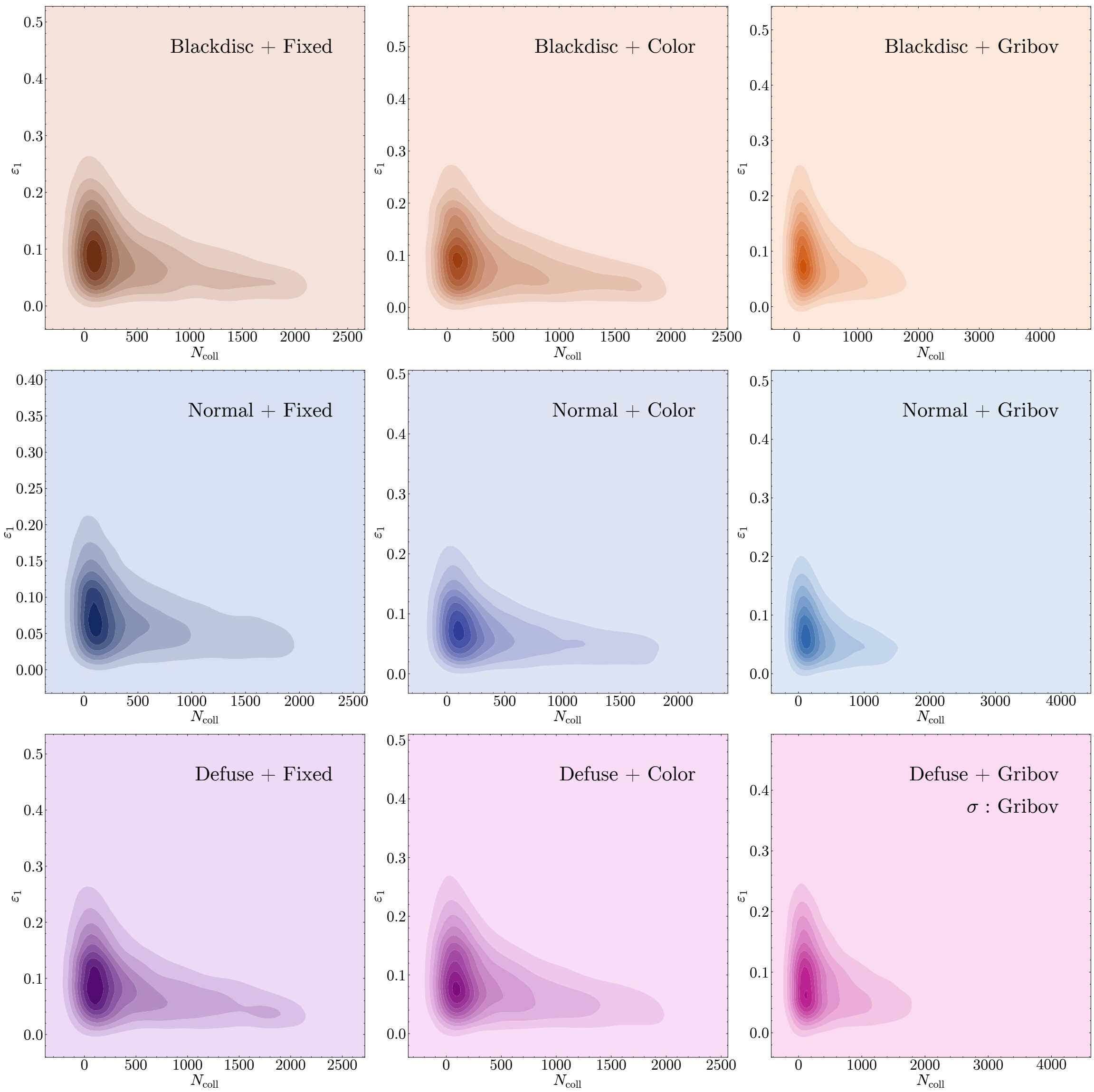


Figure D.1. ε_1 vs N_{coll}

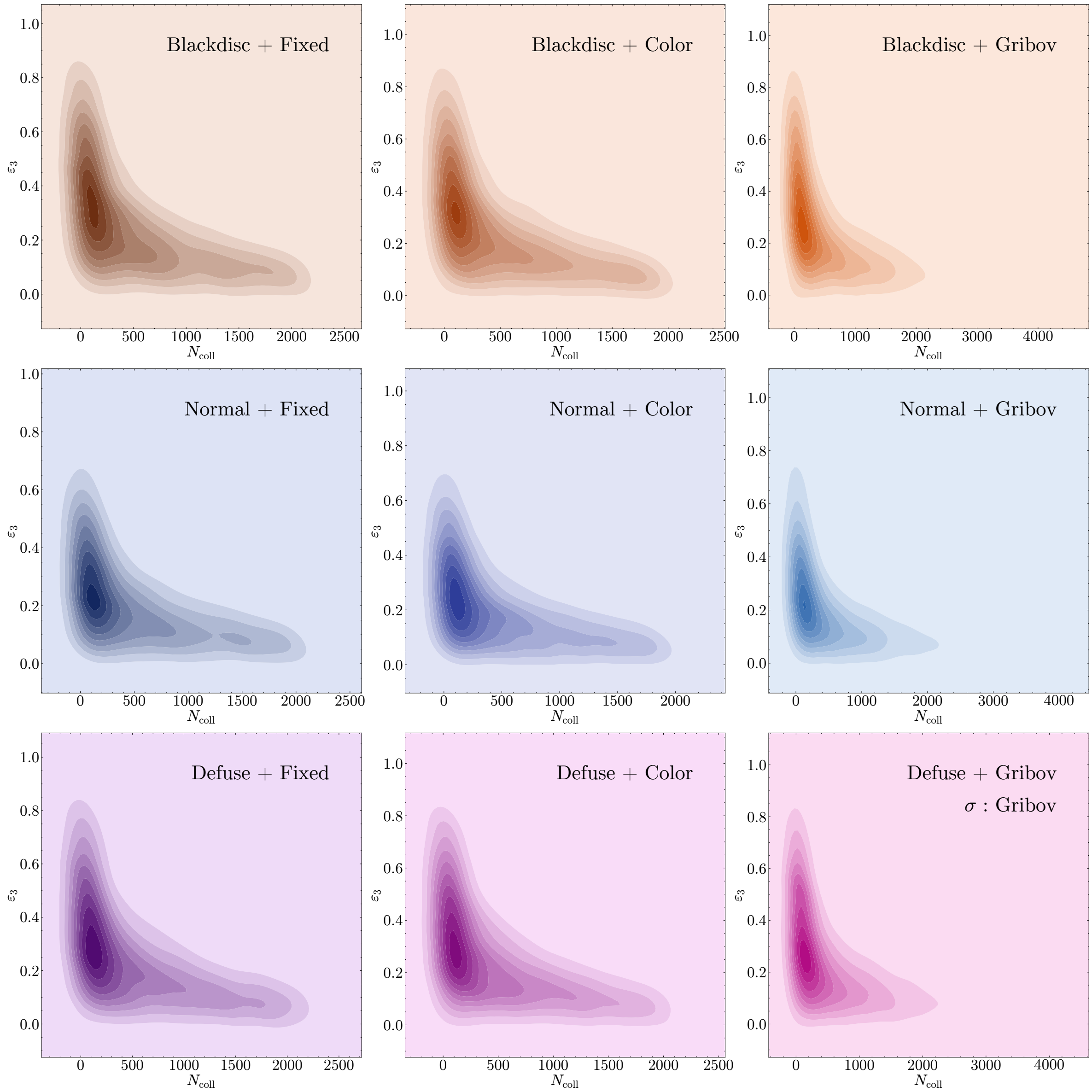


Figure D.2. ε_3 vs N_{coll}

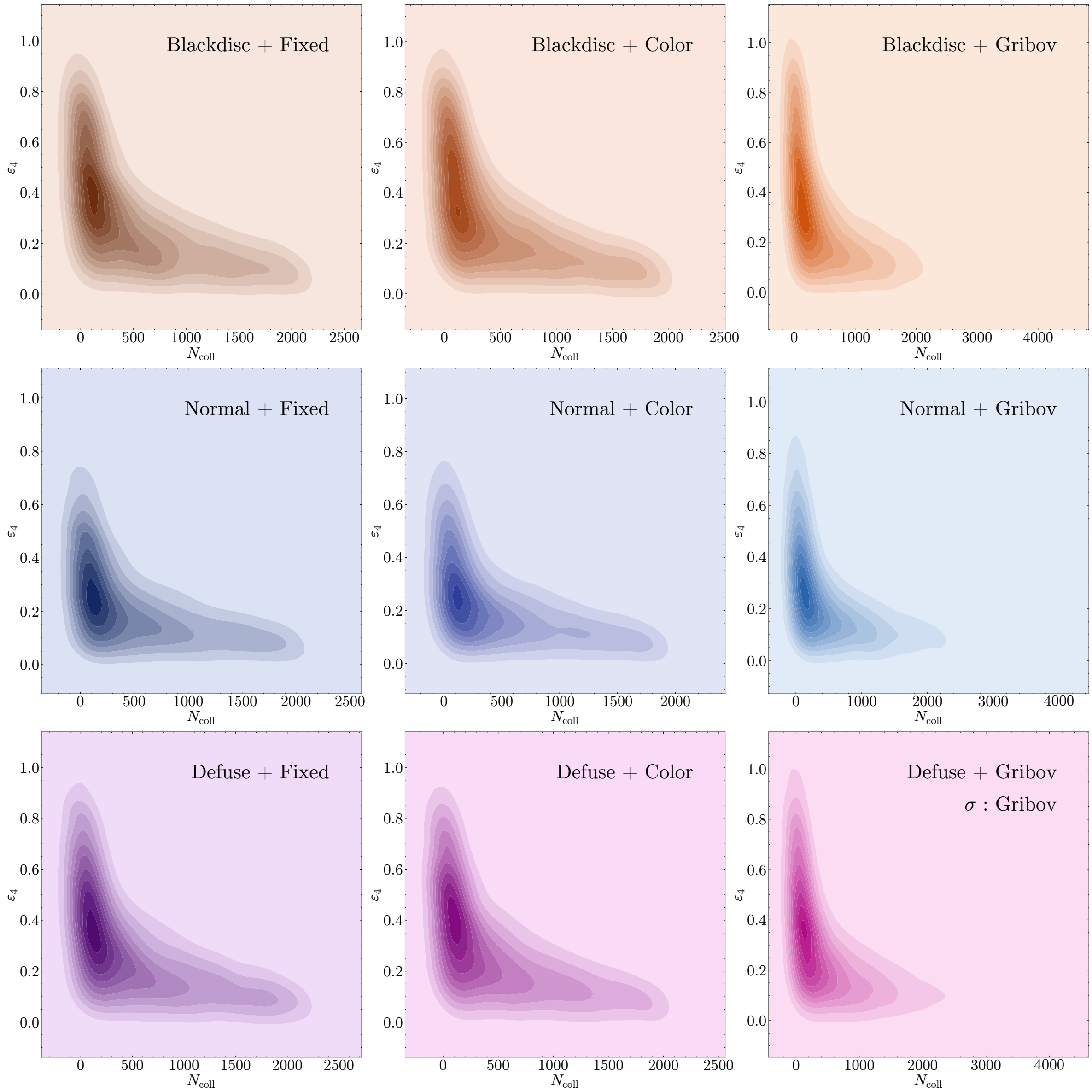


Figure D.3. ε_4 vs N_{coll}

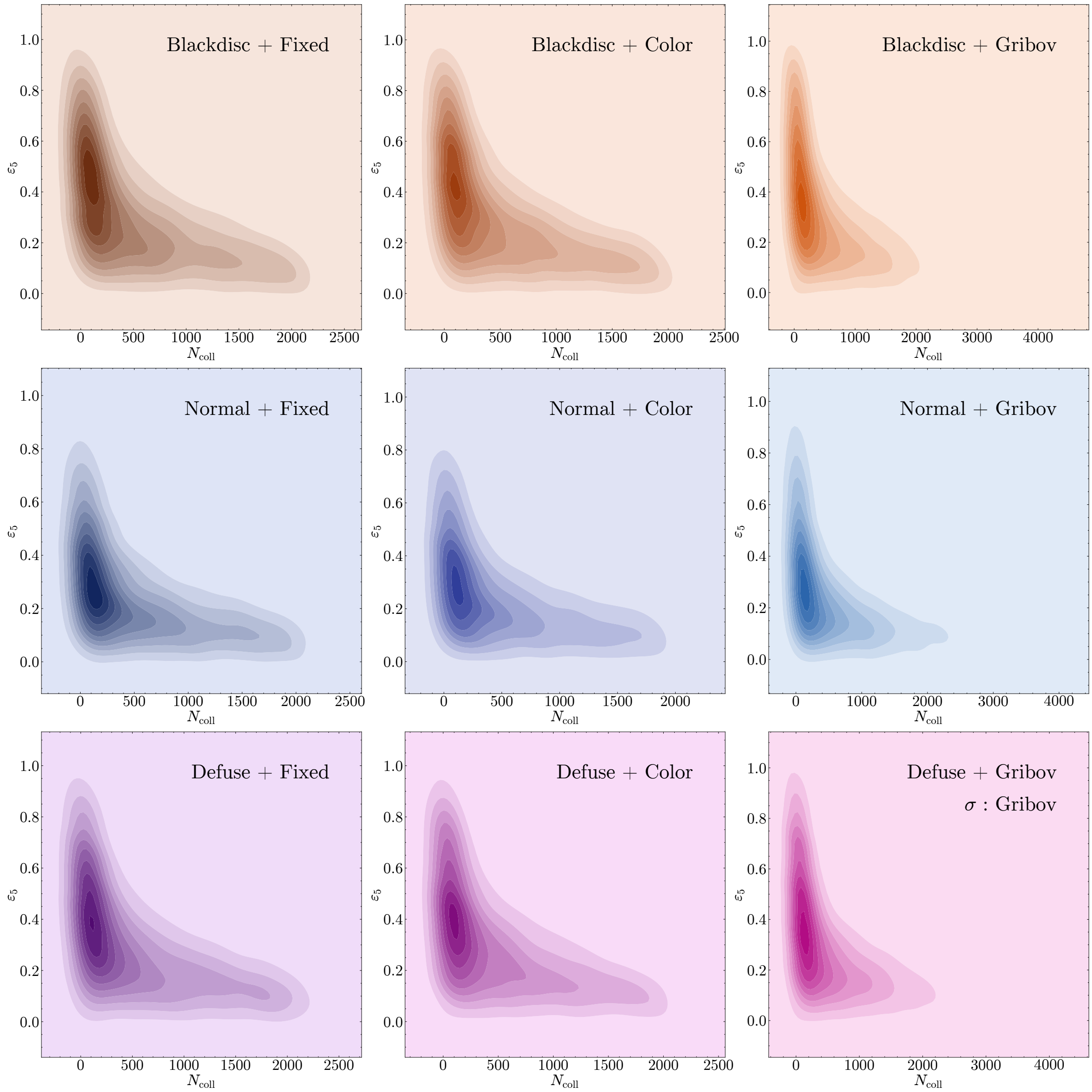
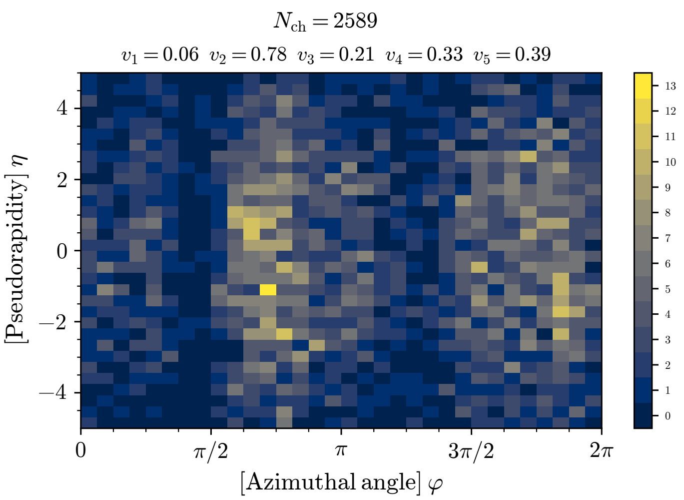


Figure D.4. ε_5 vs N_{coll}

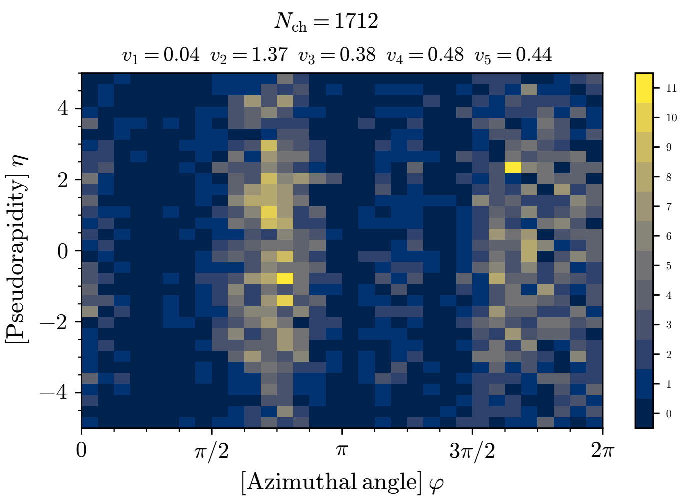
E

Events analyzed by the proposed model

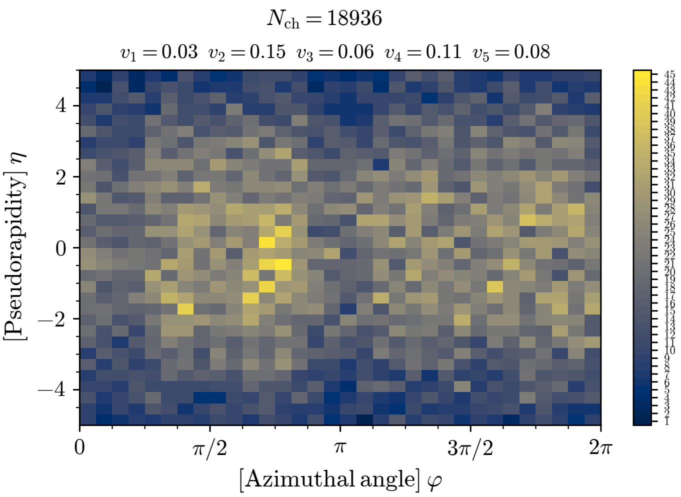
The events shown below have been used to estimate the flow coefficients presented in the *Results* chapter. These random events has been produced with a black disc interaction profile with a fixed cross section.



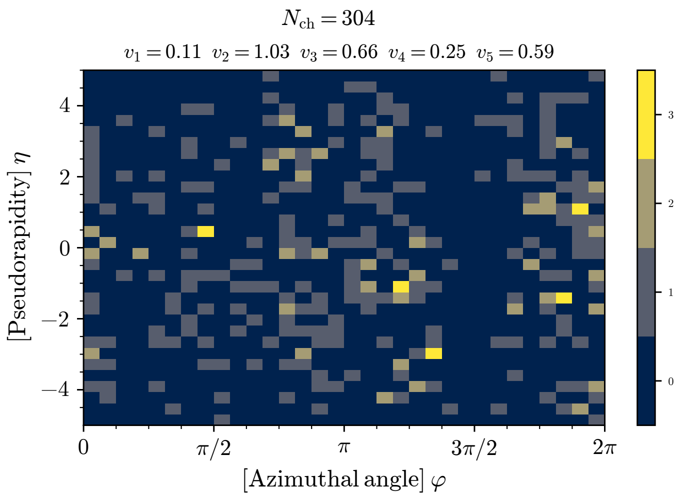
Event # 1



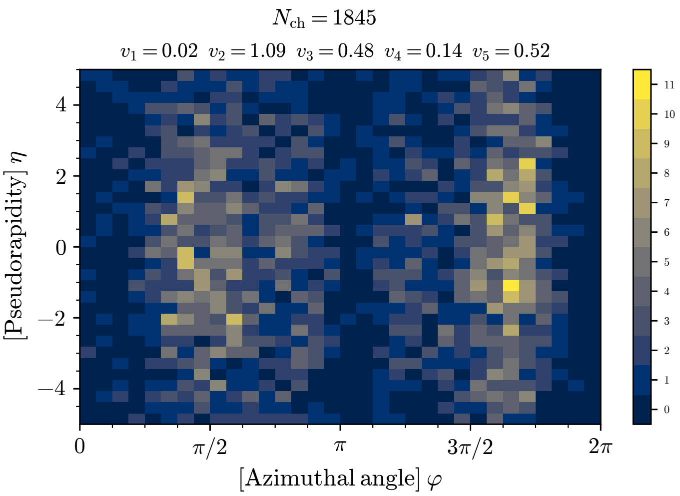
Event # 2



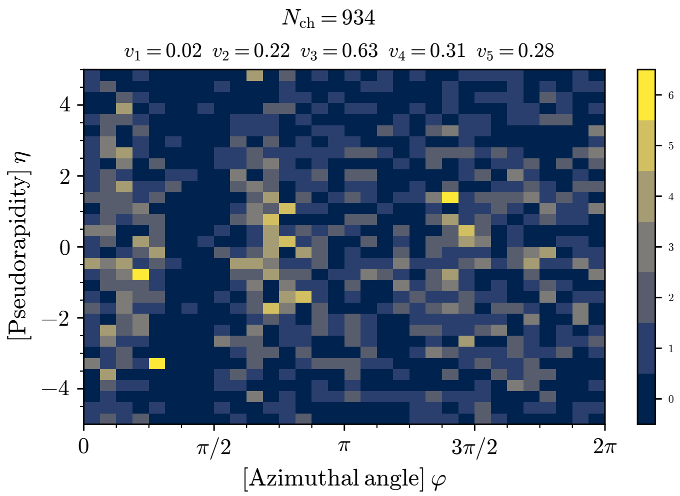
Event # 3



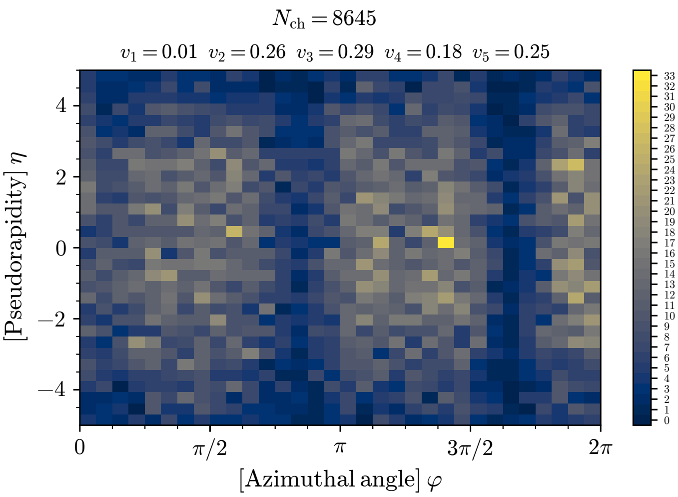
Event # 4



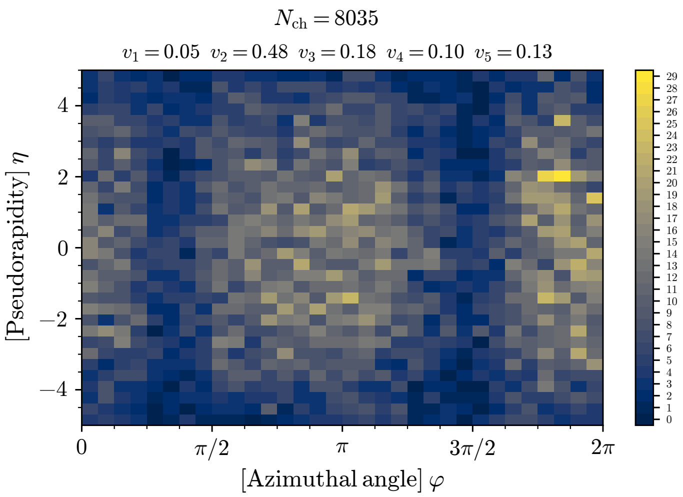
Event # 5



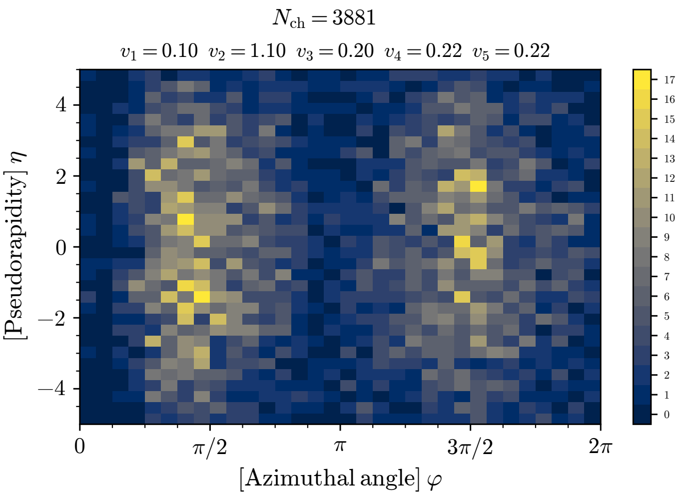
Event # 6



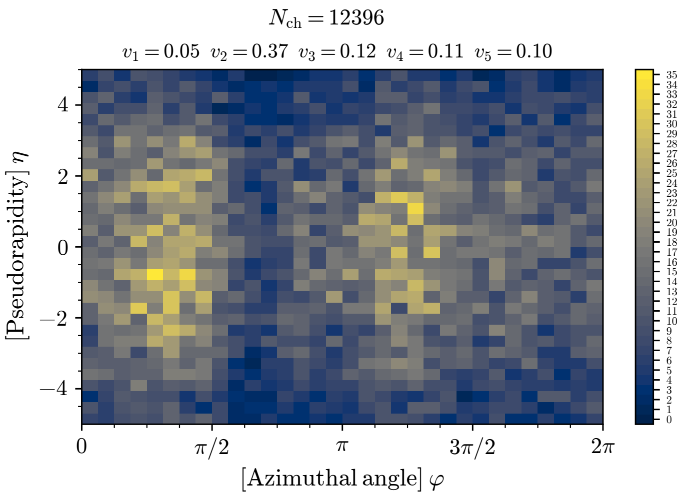
Event # 7



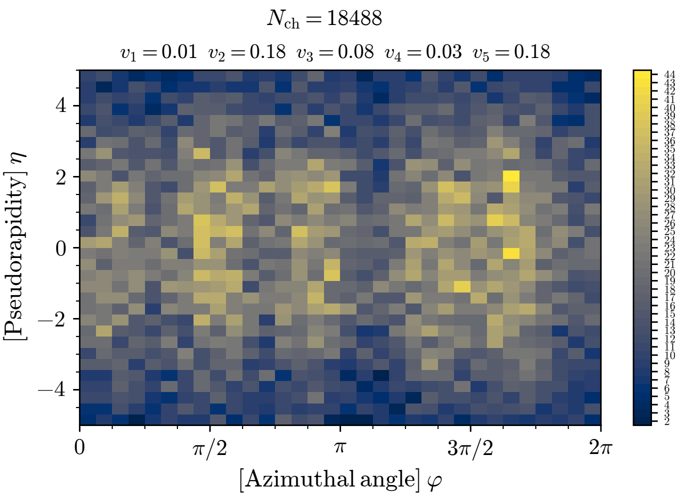
Event # 8



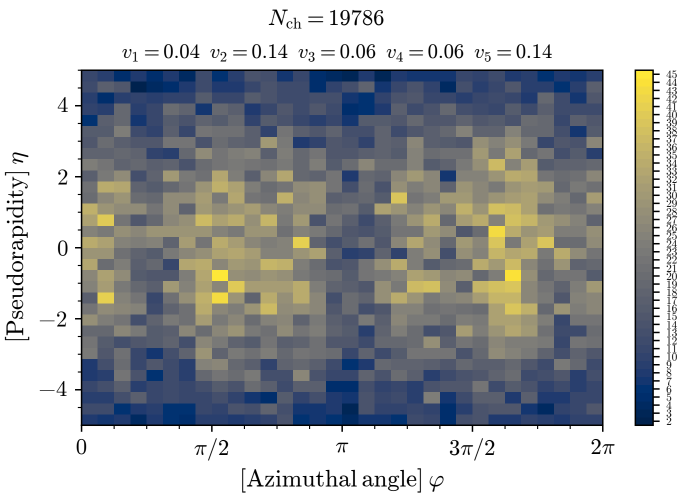
Event # 9



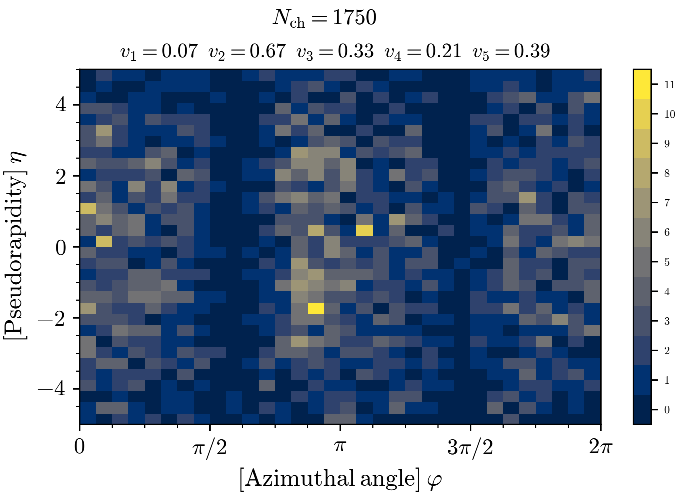
Event # 10



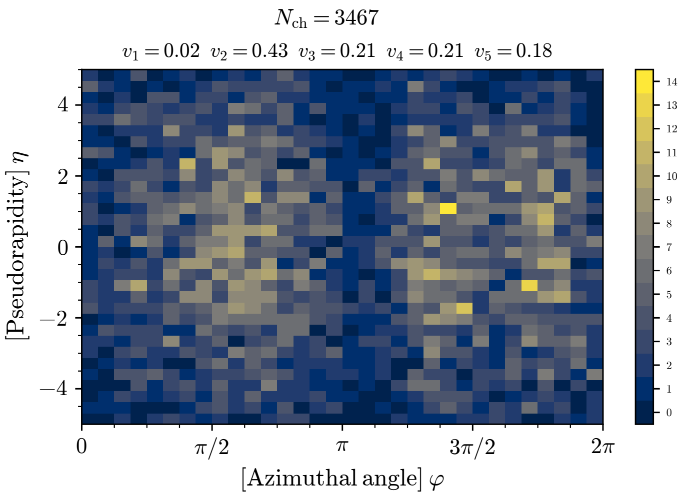
Event # 11



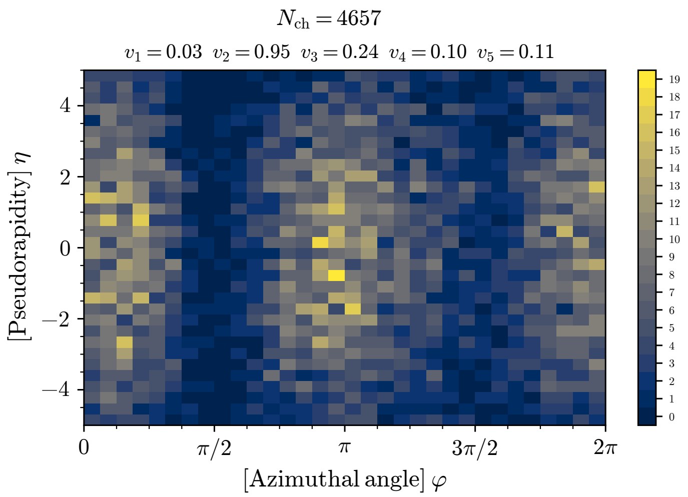
Event # 12

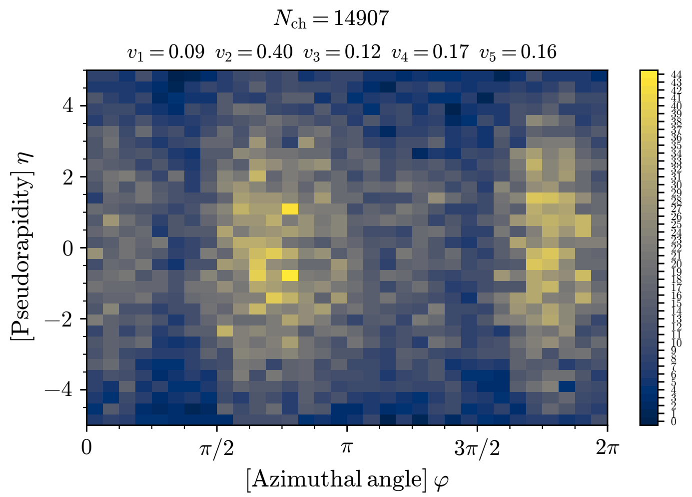


Event # 13

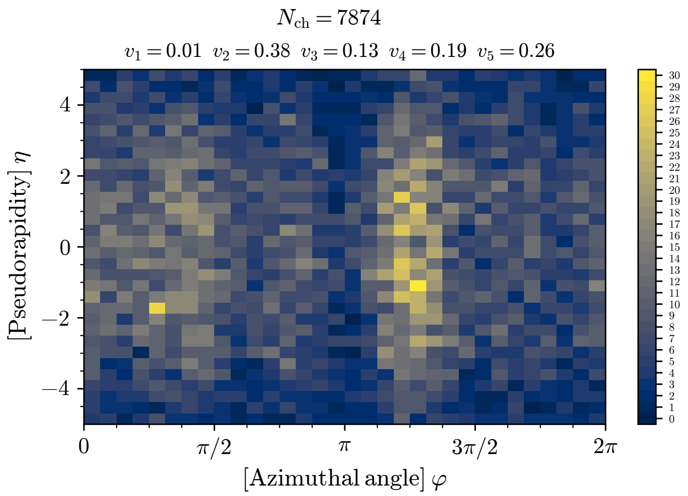


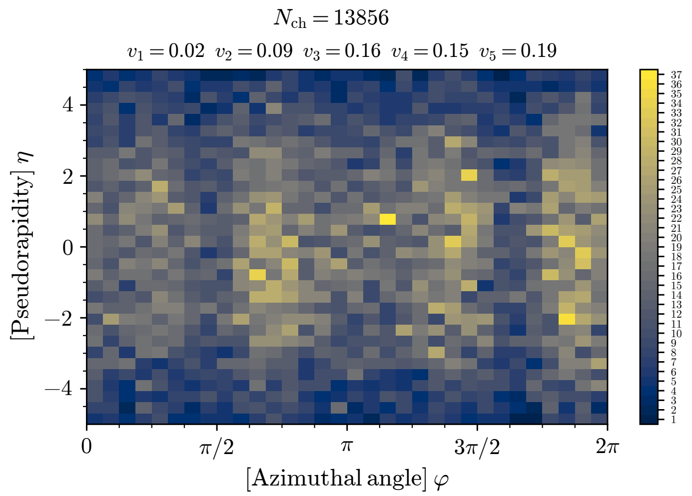
Event # 14



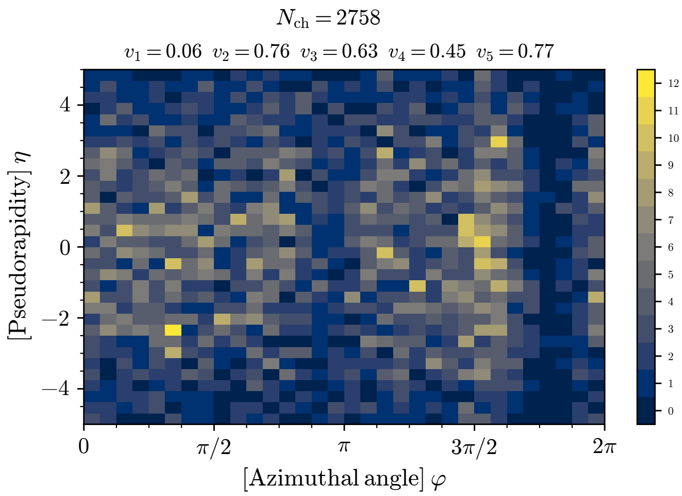


Event # 15 Event # 16

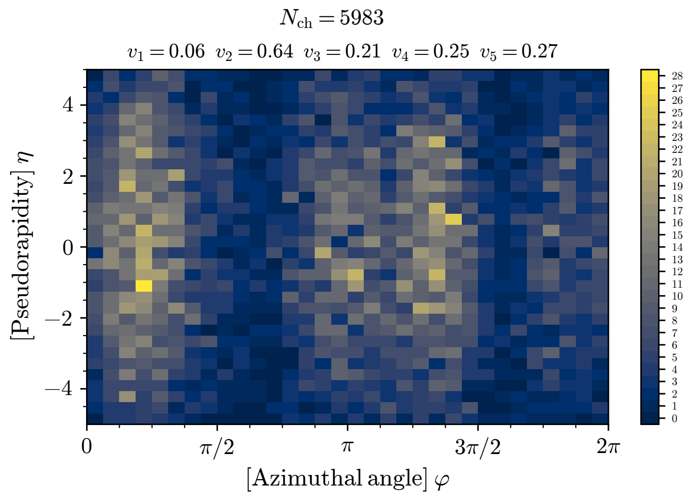




Event # 17 Event # 18



Event # 19



Event # 20