Development of Muon Reconstruction Method Based on Machine Learning for ALICE Run 3 and Improvement of Performance for Verification of Origin of Hadronic Mass

---

Author:
Ren EJIMA

Supervisors:
Prof. Kenta SHIGAKI
Prof. Satoshi YANO

A thesis submitted in fulfilment
of the requirements for the degree of
Master of Science

Quark Physics Laboratory
Graduate School of Advanced Science and Engineering
Physics Program

Hiroshima University
Hiroshima, Japan

January 2023

Development of Muon Reconstruction Method Based on Machine Learning for ALICE Run 3 and Improvement of Performance for Verification of Origin of Hadronic Mass, © January 2023

Author:
Ren Ejima

Supervisors:
Prof. Kenta Shigaki
Prof. Satoshi Yano

Institute:
Hiroshima University, Hiroshima, Japan

CONTENTS

Abstract v

1 INTRODUCTION 1
1.1 QCD 1
1.2 Symmetry 2
1.3 Chiral transformation and chiral symmetry 2
1.4 Noether's theorem and chiral partners 3
1.5 Effective field theory (linear sigma model) 5
1.6 spontaneous breaking of symmetry 5
1.7 Relation between spontaneous breaking of chiral symmetry and hadron mass; experimental verification 8
1.8 ALICE 9
1.8.1 MUON spectrometer 10
1.8.2 MFT 12
1.8.3 Global muon tracking 16
1.9 Machine Learning 17
1.9.1 GBDT 18
1.10 Kalman filter 24
1.11 Purpose of this study 25

2 APPLICATION OF MACHINE LEARNING TO μ TRACK RECONSTRUCTION WITH MULTIPLE DETECTORS 27
2.1 Issues with the ALICE forward μ particle detector 27
2.2 Track matching using Kalman filter 28
2.3 Track matching by classification using GBDT 29
2.3.1 data-set 29
2.3.2 Handling imbalance data 31
2.4 Track matching by ranking using GBDT 40

3 REQUESTS FROM PHYSICS FOR TRACK MATCHING 45
3.1 Time evolution of QGP in the Transverse plane 45

4 PERFORMANCE COMPARISON OF TRACK MATCHING METHODS 53
4.1 Efficiency of track matching and Purity of tracks 53
4.1.1 Confirmation of Kalman filter performance 53
4.1.2 Performance of track matching with machine learning based classifiers 54
4.1.3 Performance of track matching by ranking using machine learning . . 58
4.2 Effect of matching method on invariant mass distribution of dimuon 61

5 DISCUSSION 65
5.1 Discussion 65

6 CONCLUSIONS 69
6.1 Major Contribution 69
6.2 Outlook and Future Work 70

BIBLIOGRAPHY 71

Acknowledgement 73

ABSTRACT

The ALICE experiment aims to elucidate the properties of the hot parton unconfined phase (QGP phase), which is the state of the universe immediately after the Big Bang, and the special physics that occurs in the QGP phase through high-energy nuclear collisions using the LHC accelerator at CERN. One of the important questions is how hadrons acquire mass as the universe grows; it is predicted that in the QGP, the spontaneous breaking of chiral symmetry is partially recovered and hadrons become lighter. Since this state cannot be observed directly, we use the decaying child particles as probes and reconstruct the mass of the parent particle from the information of the child particles. A good probe here is the μ pair, which does not interact strongly with the QGP and has few background events. Ideally, the hadrons to be measured should be light, with a pronounced mass change. Hadrons that satisfy the two requirements of being light and decaying into muon pairs are ρ meson, ω meson, and ϕ meson, but the momentum of μ originating from the decay of light hadrons is low. This causes multiple scattering of μ in the material layer for μ identification, and the resolution of the intertrack angles of μ pairs immediately after the decay is low. Since the mass of the parent hadron depends on this angle, angular resolution has been an issue in the measurement. To improve this resolution, a new trajectory detector was installed at the position before multiple scattering in the material layer for μ identification for LHC-Run 3, which started in July 2022. However, the new detector and the existing detector must be used in combination to compensate for each other's strengths and weaknesses, and the particle tracks detected by each detector must be correctly connected to each other. However, connecting the tracks is a challenging task due to the multiple scattering of μ and the huge number of tracks. In this study, we have attempted to address this issue using machine learning. We are currently developing an algorithm for connecting tracks using machine-learning classification by simulation, and the percentage of correctly connected tracks is higher than that of other algorithms. In order to further improve the accuracy and efficiency of connecting tracks, a completely different method using machine learning for ranking was also tested, and it was able to correctly connect tracks with higher efficiency than classification. In addition, for the simulation of Pb-Pb collisions at LHC-Run 3, we obtained the invariant mass distribution of hadrons using a machine learning classification algorithm and quantitatively evaluated the improvement in the signal-to-noise ratio of this distribution compared to that of the conventional method originally proposed in ALICE. The results showed a significant improvement of 6% in the signal-to-noise ratio. Further improvement of the signal-to-noise ratio can be expected by using machine-learning ranking and optimizing the cut of tracks.

1

INTRODUCTION

1.1 QCD

The current understanding of physics is that all forces consist of four types: gravitational, electromagnetic, weak, and strong forces. The current prevailing view is that all these forces were the same at the moment of the birth of the universe and were separated into the four types as we know them today due to the breaking of various mathematical Lagrangian symmetries and the construction of an ultimate theory that describes all forces in a unified manner is one of the ultimate goals of modern physics. QCD (Quantum Chromodynamics) is a theory that explains the strong force in terms of quantum field theory, and the following equation gives the Lagrangian of QCD [1].

$$\mathcal{L}_{QCD} = \bar{q}_{\alpha,f}(i\gamma^\mu D_\mu^{\alpha\beta} - m_f\delta^{\alpha\beta})q_{\beta,f} - \frac{1}{4}G^\alpha_{\mu\nu}G_a^{\mu\nu} \tag{1.1}$$

The first term is the Dirac field, representing the propagation of a single quark. The second and subsequent terms are expressed in expanded form as

$$\begin{aligned}
\mathcal{L}_{QCD} =& \bar{q}_{\alpha,f}(i\gamma^\mu\partial_\mu - m_f)q_{\alpha,f} && (1.2)\\
&- \frac{1}{4}(\partial_\mu A^a_\nu - \partial_\nu A^a_\mu)(\partial^\mu A^\nu_a - \partial^\nu A^\mu_a) && (1.3)\\
&- g\bar{q}_{\alpha,f}\gamma^\mu A^a_\mu (t^a)^{\alpha\beta} q_{\beta,f} && (1.4)\\
&+ (\partial_\mu A^a_\nu - \partial_\nu A^a_\mu) g f_{ade} A^\mu_d A^\nu_e + g f_{abc} A^b_\mu A^c_\nu (\partial^\mu A^\nu_a - \partial^\nu A^\mu_a) && (1.5)\\
&+ g^2 f_{abc} f_{ade} A^b_\mu A^c_\nu A^\mu_d A^\nu_e && (1.6)
\end{aligned}$$

The second term represents the propagation of gluons, which are gauge bosons that transmit the strong force; the third term represents the strong interaction between quarks and gluons, the fourth term represents the three-point interaction of gluons only, and the fifth term describes the four-point interaction of gluons only. As mentioned earlier, gluons are gauge bosons that transmit strong forces, but the interactions between gauge bosons (the fourth and fifth terms) are unique features of strong forces that are not found in other forces. The Lagrangian also contains a coupling constant that expresses the strength of each interaction. The interaction between quarks and gluons (third term) and the three-point interaction of gluons (fourth term) are multiplied by the coupling constant g, and the four-point interaction of gluons (fifth term) is further multiplied by g^2. This is a very tricky point in solving QCD. The coupling constant asymptotically approaches zero in the

high-energy region (long distance). It can be solved perturbatively, but in the low-energy region (short distance), such as inside hadrons, the coupling constant has a finite value that cannot be ignored, so the space-time cannot be solved perturbatively. The current situation is that it can only be solved numerically by lattice QCD, in which points on a lattice represent space-time.

1.2 SYMMETRY

In physics, a system is said to have symmetry when the form of the Lagrangian is invariant when a particular mathematical transformation is applied to the Lagrangian that represents the system. In this study, chiral symmetry is discussed. Chiral symmetry is symmetry when a chiral transformation is applied to a Lagrangian.

1.3 CHIRAL TRANSFORMATION AND CHIRAL SYMMETRY

For simplicity, we consider the Lagrangian of the Dirac field instead of the Lagrangian of QCD since we consider only the quark field.

$$\mathcal{L}_{Dirac} = \bar{q}(i\gamma^{\mu}\partial_{\mu} - m)q \tag{1.7}$$

The Dirac field is the first term of the Lagrangian of QCD, describing the propagation of quarks. We now consider splitting the quark field q into right-handed and left-handed.

$$q = q_L + q_R,\ \bar{q} = \bar{q}_L + \bar{q}_R \tag{1.8}$$

Where, right-handed and left-handed are defined as the chirality operator γ^5 acting on the field.

$$\psi_L = \frac{1-\gamma^5}{2}\psi,\ \psi_R = \frac{1+\gamma^5}{2}\psi \tag{1.9}$$

When the right- and left-handed quark fields are separated into right- and left-handed quark fields in the Lagrangian, the kinetic terms completely separate the right- and left-handed components, as shown below.

$$\bar{q}(i\gamma^{\mu}\partial_{\mu})q = \bar{q}_L(i\gamma^{\mu}\partial_{\mu})q_L + \bar{q}_R(i\gamma^{\mu}\partial_{\mu})q_R \tag{1.10}$$

However, the mass terms are a mixture of right-handed and left-handed.

$$-m\bar{q}q = -m\bar{q}_L q_R - m\bar{q}_R q_L \tag{1.11}$$

Next, we consider the chiral transformation for this Lagrangian. A chiral transformation is a transformation that rotates the quark flavor independently of right-handed and left-

handed. Consider the following operator that combines axial and polar transformations [2].

$$U = e^{it^a(\theta_V + \gamma^5\theta_A)} \tag{1.12}$$

The action of this operator is precisely the chiral transformation. Applying the chiral transformation to the right-handed and left-handed quark and antiquark fields, respectively, yields

$$q_L \to U_L q_L,\ \bar{q}_L \to \bar{q}_L U_L^\dagger \tag{1.13}$$

$$q_R \to U_R q_R,\ \bar{q}_R \to \bar{q}_R U_R^\dagger \tag{1.14}$$

It changes in this way. Given that the operator U of the chiral transformation has an exchange relation with γ^5, ∂_μ, and that the chirality operator γ^5 acting on the right-handed and left-handed quark fields yields eigenvalues $+1$ for the right-handed and -1 for the left-handed, the chiral transformation, whereas the Lagrangian kinetic terms are

$$\bar{q}_L(i\gamma^\mu\partial_\mu)q_L + \bar{q}_R(i\gamma^\mu\partial_\mu)q_R \to \bar{q}_L U_L^\dagger(i\gamma^\mu\partial_\mu)U_L q_L + \bar{q}_R U_R^\dagger(i\gamma^\mu\partial_\mu)U_R q_R \tag{1.15}$$

$$= \bar{q}_L(i\gamma^\mu\partial_\mu)q_L + \bar{q}_R(i\gamma^\mu\partial_\mu)q_R \tag{1.16}$$

and returns to the same form as before the chiral transformation. In other words, the kinetic term of the quark field has chiral symmetry. The mass term, on the other hand, is

$$-m\bar{q}_L q_R - m\bar{q}_R q_L \to -m\bar{q}_L U_L^\dagger U_R q_R - m\bar{q}_R U_R^\dagger U_L q_L \tag{1.17}$$

and differs from the original form. In other words, the mass term of the quark field breaks the chiral symmetry. When the Lagrangian has a symmetry-breaking form like this, it is called "explicit symmetry breaking," etc.

1.4 NOETHER'S THEOREM AND CHIRAL PARTNERS

Noether's theorem in quantum field theory is the same as Noether's theorem in analytical mechanics, which states that when a Lagrangian has a certain symmetry, a certain physical quantity of the system corresponding to that symmetry is conserved. In the equation, the following quantities are conserved [2].

$$0 = \delta\mathcal{L} = \frac{\partial\mathcal{L}}{\delta\phi}\delta\phi + \frac{\partial\mathcal{L}}{\delta\partial_\mu\phi}\delta\partial_\mu\phi = \partial_\mu\left[\frac{\partial\mathcal{L}}{\partial(\partial_\mu\phi)}\delta\phi\right] \tag{1.18}$$

Here, the conservation current of the chiral symmetry (since Noether's theorem comes from the differential form of the charge conservation law (the continuity formula), the

conserved quantity is called Noether current and the spatial integral of its time component is called Noether charge) can be written as follows [2].

$$\partial_\mu J_5^\mu = 2im\bar{\psi}\gamma^5\psi, \;\; J_5^\mu = \bar{\psi}\gamma^\mu\gamma^5\psi \tag{1.19}$$

The quark field also preserves the following currents because of the $U(1)$ symmetry.

$$\partial_\mu J^\mu = 0, \;\; J^\mu = \bar{\psi}\gamma^\mu\psi \tag{1.20}$$

Here, $U(1)$ symmetry is the symmetry of the following transformation.

$$\psi \to e^{-i\alpha}\psi \tag{1.21}$$

When the mass m of the quark field is 0, the Lagrangian has chiral symmetry, as confirmed in the previous section, and the conservation current of the chiral symmetry is conserved as follows.

$$J_{R,L}^\mu = \bar{\psi}\gamma^\mu\frac{1\pm\gamma^5}{2}\psi = \bar{\psi}_{R,L}\gamma^\mu\psi_{R,L}, \;\; \partial_\mu J_{R,L}^\mu = 0 \tag{1.22}$$

The conserved charge is the spatial integral of the conserved current, and $Q_{V,A}$ is defined as follows. Note that Q_V is the conserved charge of the polar transformation, and Q_A is the conserved charge of the axial transformation.

$$Q_{R,L} = \int d^3x J_{R,L}^0 = \int d^3x \psi_{R,L}^\dagger\psi_{R,L}, \;\; Q_V = Q_R + Q_L, \;\; Q_A = Q_R - Q_L \tag{1.23}$$

Then $Q_{V,A}$ has the following relationship with the parity operator P.

$$PQ_VP^{-1} = Q_V, \;\; PQ_AP^{-1} = -Q_A \tag{1.24}$$

Now consider a particle with mass M and parity π. That is, the eigenvalue π is obtained by acting the parity operator as follows.

$$P|M,\pi\rangle = \pi|M,\pi\rangle \tag{1.25}$$

Here, acting on the conserved charge of the axial transformation described earlier, we obtain

$$PQ_A|M,\pi\rangle = PQ_AP^{-1}P|M,\pi\rangle = -\pi Q_A|M,\pi\rangle \tag{1.26}$$

and we can consider $Q_A|M,\pi\rangle$ to be a particle with equal mass and inverted positive and negative eigenvalues of parity. Such a pair of particles $|M,\pi\rangle$ and $Q_A|M,\pi\rangle$ is called a chiral partner, and when the conservation current of chiral symmetry is conserved, that is, when the system has chiral symmetry, the chiral partner has such a state $|M,\pi\rangle$ and $Q_A|M,\pi\rangle$ are taken. Although the chiral symmetry of the quark Lagrangian is overtly broken by the mass term, the u,d,s quarks are light in mass and can be regarded as having

chiral symmetry preserved to a good approximation. However, in the real world, even in the u,d,s sector, the chiral partners are not in the states $|M, \pi\rangle$ and $Q_A|M, \pi\rangle$. This is because the chiral symmetry is spontaneously broken.

1.5 EFFECTIVE FIELD THEORY (LINEAR SIGMA MODEL)

To understand what it means for chiral symmetry to be spontaneously broken, as opposed to explicitly broken, we briefly introduce effective field theory (in this case, the linear sigma model).
First, since the linear sigma model deals with the effective field of nucleons, we introduce the usual Lagrangian of the nucleon field. The Lagrangians for nucleons (protons and neutrons) are as follows

$$\mathcal{L}_N = i\bar{\psi}\gamma^5\partial_\mu\psi - M_N\bar{\psi}\psi, \quad \psi = \begin{pmatrix} p \\ n \end{pmatrix} \tag{1.27}$$

Like the quark Lagrangian, the nucleon Lagrangian has a mass term that overtly breaks the chiral symmetry.
In the linear sigma model for π and its Like model is such that the Lagrangian is chiral symmetric. Its Lagrangian is as follows [2].

$$\mathcal{L} = \frac{1}{2}\partial_\mu\sigma\partial^\mu\sigma + \frac{1}{2}\partial_\mu\pi\cdot\partial^\mu\pi - \frac{\lambda}{4}\left[(\sigma^2+\pi^2)-\mu^2\right]^2 + \mathcal{L}'_N \tag{1.28}$$

Where $\mathcal{L}'_N$ is the Lagrangian for the effective field of nucleons as follows.

$$\mathcal{L}'_N = i\bar{\psi}\gamma^5\partial_\mu\psi - g\bar{\psi}(\sigma + i\tau\cdot\pi\gamma^5)\psi \tag{1.29}$$

Considering such a model makes it easier to understand what it means for symmetry to be spontaneously broken.

1.6 SPONTANEOUS BREAKING OF SYMMETRY

In the Lagrangian of the linear sigma model, the first and second terms are the kinetic terms of σ and π, respectively. After the third term, the fourth term is the Lagrangian of the nucleon. In analytical mechanics, the Lagrangian L is the difference between the kinetic energy K and the potential V, $L = K - V$. With this in mind, the third term can be viewed as $-V$, and in the following we consider the following potentials.

$$V(\sigma,\pi) = \frac{\lambda}{4}\left[(\sigma^2+\pi^2)-\mu^2\right]^2 \tag{1.30}$$

$$= \frac{\lambda^2}{4}(\sigma^2+\pi^2)^2 - \frac{\lambda^2\mu^2}{2}(\sigma^2+\pi^2) + \frac{\lambda^2\mu^4}{4} \tag{1.31}$$

Here, what is actually realized as the ground state of the vacuum is when this potential takes an extreme value. In other words,

$$\frac{\partial^2 V(\sigma,\pi)}{\partial\sigma\partial\pi} = 0 \tag{1.32}$$

Thethe form of this potential changes shape significantly with the positive and negative μ^2.
When $\mu^2 > 0$, the potential takes the following form.

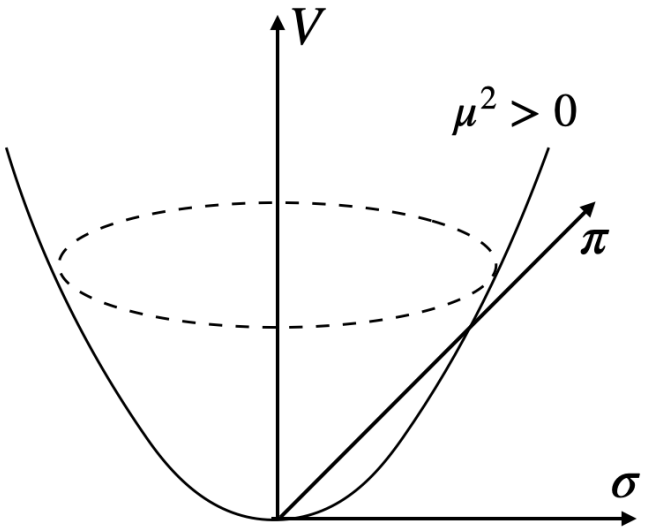


Figure 1.1: Wigner phase

Such a state is called the Wigner phase, and the ground state of the vacuum to be realized is $V(0,0) = 0$ at one point $(\sigma,\pi) = (0,0)$. And in this case, the masses of σ and π are both represented by $\sqrt{-\lambda^2\mu^2}$. In other words, the chiral partners have equal masses.
When $\mu^2 < 0$, the potential takes the following form.

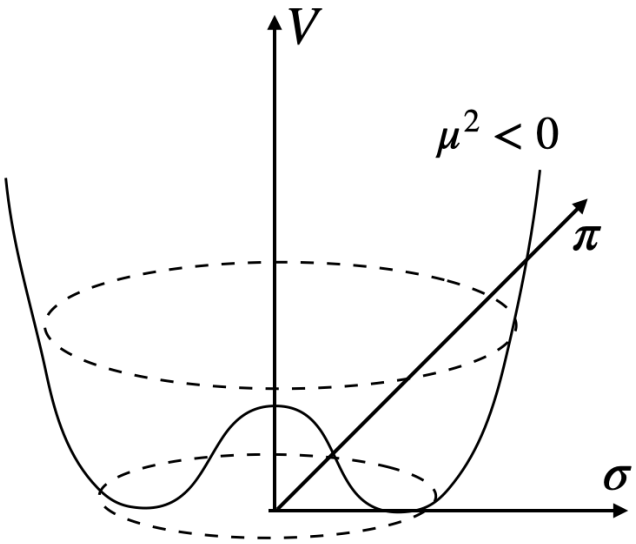


Figure 1.2: Goldstone phase

Such a state is called the Goldstone phase, and the ground state of the vacuum to be realized is (σ, π) such that it takes the extreme value of the potential, so it is at the origin $(0, 0)$ or on the circumference around it $\sigma^2 + \pi^2 = \mu^2$. However, since the origin is unstable stable, it spontaneously falls on the circumference with just a slight fluctuation. Now, consider rotating the σ, π plane (equivalent to a chiral transformation), where the vacuum to be realized is viewed from above the potential. In the Wigner phase, the vacuum is at the origin, so there is no change even if the coordinates are rotated, and the system has chiral symmetry. In the Goldstone phase, however, one point on the circumference is realized as the vacuum, so when the coordinates are rotated, the realized vacuum is also rotated accordingly. In this case there is no chiral symmetry. Thus, even though the Lagrangian of the system itself does not break the symmetry, when the realized vacuum breaks the symmetry, it is called "spontaneous symmetry breaking".
In the following, we will check the masses of σ and π when the chiral symmetry is spontaneously broken. Since the actual vacuum is realized on the circumference of $\sigma^2 + \pi^2 = \mu^2$, we assume here, for example, that the point $(\sigma, \pi) = (\mu, 0)$ is the actual vacuum realized. In this case, we consider a coordinate transformation that shifts the origin to the realized vacuum $(\sigma, \pi) = (\mu, 0)$. In other words, we consider such a transformation as $\sigma \to \sigma + \mu$,$\pi \to \pi$. In this case, the Lagrangian of the system is as follows.

$$\to \mathcal{L} = \frac{1}{2}\partial^\mu\sigma\partial_\mu\sigma - \frac{1}{2}2\lambda^2\mu^2\sigma^2 + \frac{1}{2}\partial^\mu\pi\partial_\mu\pi - \mu\lambda^2\sigma(\sigma^2+\pi^2) - \frac{1}{4}\lambda(\sigma^2+\pi^2)^2 + \mathcal{L}_N'' \quad (1.33)$$

Then the second term of this Lagrangian, $2\lambda^2\mu^2$, represents the mass of σ, and furthermore, the nucleon mass with a finite size of $-gn$ appears in $\mathcal{L}_N''$, which represented the nucleon Lagrangian. However, there is no π mass term anywhere. Thus, when the chiral symmetry is spontaneously broken, the masses of the chiral partners are different. Also, when the symmetry is spontaneously broken, it is a point on the circumference that can be realized as a vacuum, and no energy is required to move from one vacuum to another. When this is

the case, a particle with mass 0 can be considered, commonly called the Nambu-Goldstone particle. The π meson is a Nambu Goldstone particle due to spontaneous breaking of the chiral symmetry in the u.d sector. In the real world, however, π mesons have finite mass. To understand this, we consider tilting the potential of the Goldstone phase. We can tilt it by a magnitude of c in the $\sigma - V$ plane by adding the term $c\sigma$ to the Lagrangian as follows.

$$\mathcal{L}' = \frac{1}{2}\partial_\mu\sigma\partial^\mu\sigma + \frac{1}{2}\partial_\mu\pi\cdot\partial^\mu\pi - \frac{\lambda}{4}\left[(\sigma^2+\pi^2)-\mu^2\right]^2 + \mathcal{L}'_N + c\sigma \tag{1.34}$$

$$\rightarrow V(\sigma,\pi) = \frac{\lambda^2}{4}(\sigma^2+\pi^2)^2 - \frac{\lambda^2\mu^2}{2}(\sigma^2+\pi^2) + \frac{\lambda^2\mu^4}{4} \tag{1.35}$$

Then, the vacuum realized is

$$\frac{\partial^2 V(\sigma,\pi)}{\partial\sigma\partial\pi} = 0 \rightarrow \sigma_o(\sigma_0^2-\mu^2) = c/\lambda^2 \tag{1.36}$$

and if we rewrite the Lagrangian to move the origin to the point that satisfies this as before, the mass terms of σ and nucleon, as well as π appear.

$$M_N = -g\sigma_0,\ m_\sigma = \sqrt{\lambda^2(3\sigma_0^2-\mu^2)},\ m_\pi = \sqrt{\lambda^2(\sigma_0^2-\mu^2)} \tag{1.37}$$

We omit the detailed calculations here, but here the PCAC hypothesis ($\langle 0|i[Q^q_{L,R},\phi^b]|0\rangle = \delta^{a,b}\frac{2}{N_f}\langle 0|\bar{q}q|0\rangle$, ...), etc., the Gell-Mann Oakes-Renner relationship equation can be obtained [1].

$$f_\pi^2 m_\pi^2 = -\frac{m_u+m_d}{2}\langle\bar{q}q\rangle \tag{1.38}$$

However, this is just one example, and there are various other effective field theories (NJL model, QCD sum rule, chiral bag model, etc.).

1.7 RELATION BETWEEN SPONTANEOUS BREAKING OF CHIRAL SYMMETRY AND HADRON MASS; EXPERIMENTAL VERIFICATION

It has been suggested that the chiral symmetry described so far is related to the mass of hadrons. The mass of a hadron is much greater than the sum of the masses of the quarks that make up the hadron. It is believed that hadrons acquire their mass through interactions with the quark-antiquark condensed field, just as quarks acquire their mass through interactions with the Higgs field. The vacuum expectation value of the quark-antiquark condensate is as follows.

$$\langle\bar{q}q\rangle = \langle\Omega|\bar{q}q|\Omega\rangle \tag{1.39}$$

However, $|\Omega\rangle$ denotes the vacuum. This vacuum expectation value has exactly the same form as the mass term that has appeared in the Dirac field and so on, and breaks the chiral symmetry.

$$\bar{q}q = \bar{q}_L q_R + \bar{q}_R q_L \tag{1.40}$$

Therefore, when the vacuum has a finite $\langle \bar{q}q \rangle$, the vacuum spontaneously breaks the chiral symmetry. Thus, the physical quantity that expresses how much symmetry is spontaneously broken is called its order parameter of the spontaneous breaking of the symmetry. In other words, since the expected value of the vacuum $\langle \bar{q}q \rangle$ is an order parameter of the spontaneous breaking of the chiral symmetry, it is thought that hadrons acquire mass when the chiral symmetry is spontaneously broken = the vacuum is filled with quark-antiquark pairs. If the temperature is raised, the number of quark-antiquark pairs will decrease due to thermal motion overcoming the binding. Since there are no quark-antiquark pairs inside hadrons, the number of quark-antiquark pairs will decrease in regions where a large number of hadrons are densely packed. Taking advantage of this fact, experiments have been conducted to measure the mass of hadrons in space at high temperatures by high-energy nuclear collision experiments or in dense regions by launching hadron beams into nuclei of fixed targets. In both methods, it has been confirmed that the mass distribution is deformed by high temperature and high density. The reason for the deformation of the mass distribution may be not only the partial recovery of spontaneous breaking of chiral symmetry, but also many-body interactions among hadrons. Since all of the experimental data obtained can be explained only by hadron interactions, it is not possible to conclude that the mass of some hadrons changes due to the partial recovery of spontaneous chiral symmetry breaking in any of the experimental data.

1.8 ALICE

The ALICE experiment is one of the experiment using world's largest colliding circular accelerators, the LHC (Large Hadron Collider), built by CERN (European Organization for Nuclear Research). In Run 3, the collision energy in the beam center-of-mass system can reach $\sqrt{s} = 13.6\text{TeV}$ for proton-proton collisions and $\sqrt{s_{NN}} = 5.5\text{TeV}$ per nucleon pair for lead-lead collisions, which are the highest collision energies in the world. Only ALICE experiment has detectors dedicated to the study of quark-gluon plasmas in experiments using LHC and aims to elucidate their properties. In particular, the ALICE experiment specializes in particle measurements in the low transverse momentum region.

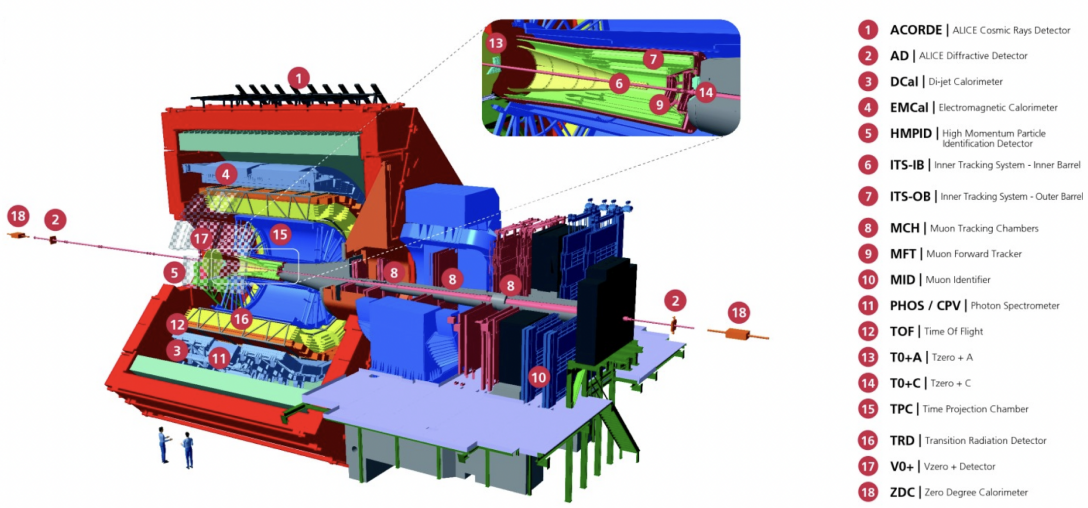


Figure 1.3: ALICE detector in LHC-Run 3.[3]

1.8.1 *MUON spectrometer*

Since particle identification of low-transverse momentum muons is possible in the forward region by satisfying both low and high transverse momentum, a group of muon detectors has been installed in the forward region in the ALICE experiment. First, muon spectrometer detectors will be explained.

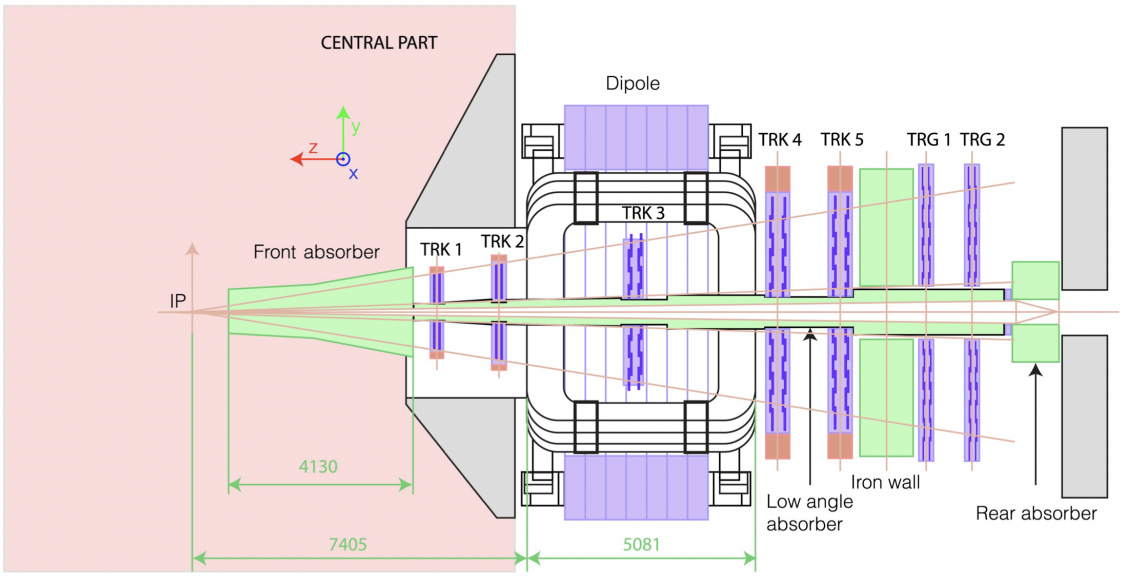


Figure 1.4: MUON spectrometer. It consists of a front absorber, MCH(TRK1 to TRK5), MID shield(Iron wall), MID(TRG1, 2).[4]

The pseudo-rapidity region covered by the MUON spectrometer is $-4 < \eta < -2.5$, and the MUON spectrometer consists of a front absorber, a dipole magnet, a tracking system (Muon CHamber, MCH), a secondary hadron absorber (MID shield), and a trigger system

(Muon IDentifier, MID)[5]. As shown in Figure1.4, MCH consists of five chamber stations. Each station has two chambers and uses a cathode pad chamber filled with Ar/CO_2 (80/20)[5]. The front absorber is a huge block about 4 m long made of concrete, steel, or carbon[5]. The front absorber stops the various particles produced in nuclear collisions, allowing only highly penetrating muons to pass through and be detected at MCH. The MID shield located between MCH and MID removes particles that could not be absorbed by the front absorber and secondary particles generated in the front absorber. Particles that pass through the front absorber are bent by a 3 T/m dipole magnet, and their tracks are measured by MCH and tracks which could pass through MID shield and be detected at MID are determined to be muons (Figure 1.5)[5].

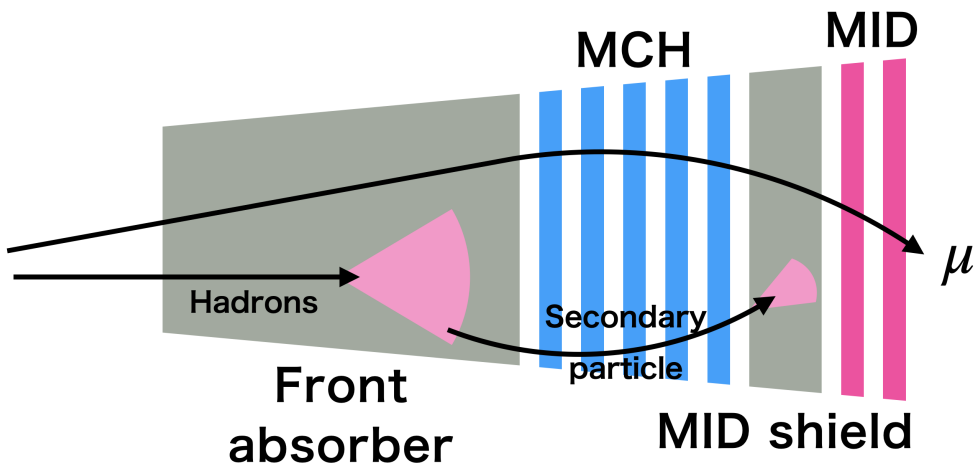


Figure 1.5: Roles of front absorber and MID shield. Particles which has momentum more than 4 GeV/c($p_T \sim 0.5$ GeV/c) can pass through the front absorber. Muons which have such a high momentum can pass through both front absorber and MID shield but hadrons and electrons are not able to pass through them and they make shower.

MCH has several pad which has different size. The smallest pad ($4.2 \times 6.3 mm^2$) has the highest occupancy and is used at the innermost perimeter of the first station[5]. Thin plate cathode pad chambers are installed in the third, fourth, and fifth stations, with the largest cathode pad($5 \times 100 mm^2$) in the last station[5]. MID uses a streamer-mode resistive chamber (RPC)[5]. The MID consists of two stations with two chambers, with 18 RPC modules per chamber[5]. The typical size of the RPC module is $70 \times 300 cm^2$ and a position resolution of about 1 cm is achieved by using an RPC filled with $Ar/C_2H_2F_4/i-buthane/SF_6$ (50.5/41.3/7.2/1) with a 2 mm gas gap[5].
Transverse momentum is calculated from the distance between the point measured at the MID and the point of impact in a straight line to the measured point at the third station of the MCH.

1.8.2 *MFT*

Muon Forward Tracker, MFT (Figure 1.6) is a new detector from LHC-Run 3.

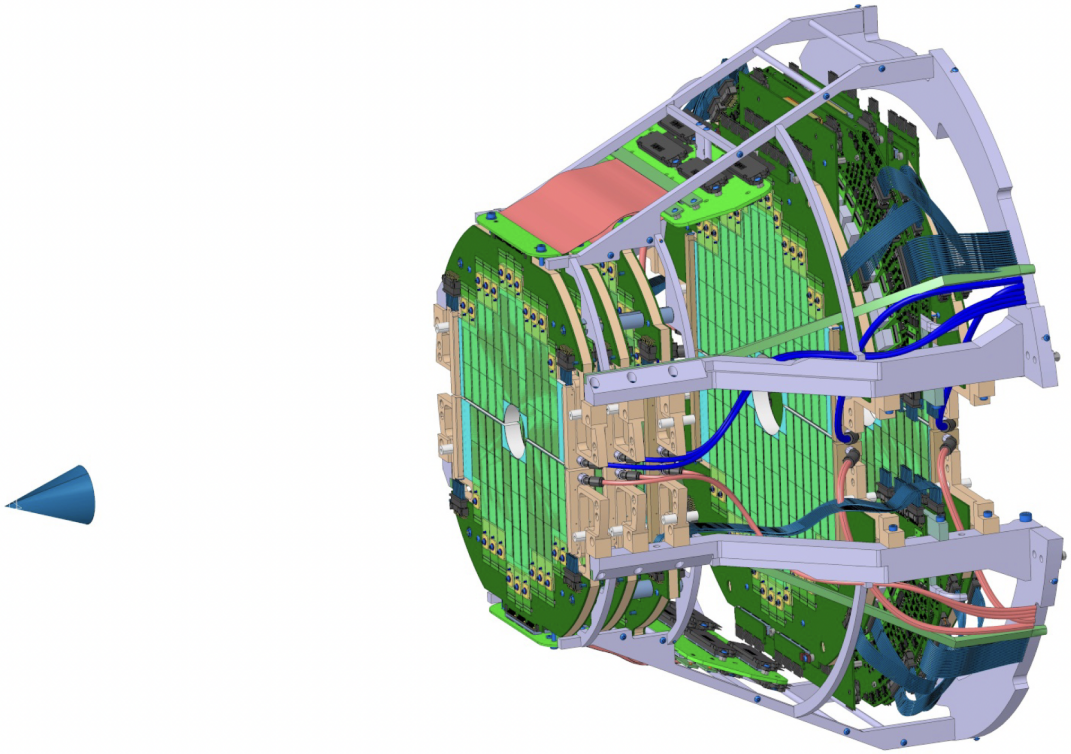


Figure 1.6: Muon Forward Tracker, MFT.[6]

Until LHC-Run 2, the only forward muon detector was a muon spectrometer. This system uses a front absorber for muon selection, eliminating many hadrons, but muons also cause front absorbers and multiple Coulomb scattering. The issue was that the tracks of muons undergoing multiple Coulomb scattering would be measured, but this would blur the tracks near the collision point before the multiple Coulomb scattering. Suppose the tracks near the collision point are blurred. In that case, the decay angle of the light vector meson when it decays into a muon pair will be ambiguous, adversely affecting the mass resolution when an invariant mass is reconstructed from the muon pair. To achieve higher mass resolution, we installed a new MFT in Run 3 that can track all charged particles that fly forward once before the muon undergoes multiple Coulomb scattering.

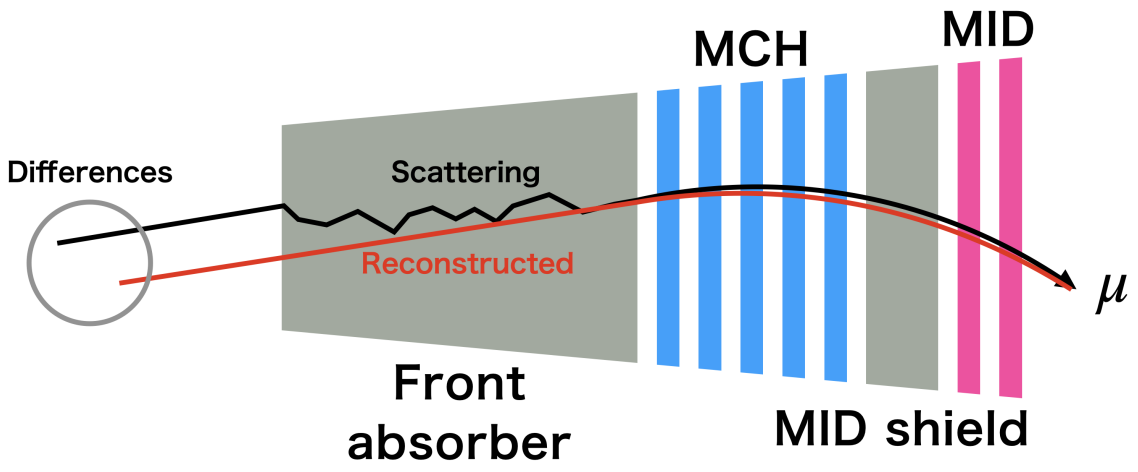


Figure 1.7: Multiple Coulomb scattering in the front absorber. This makes differences between the reconstructed track and the trajectory of the particle.

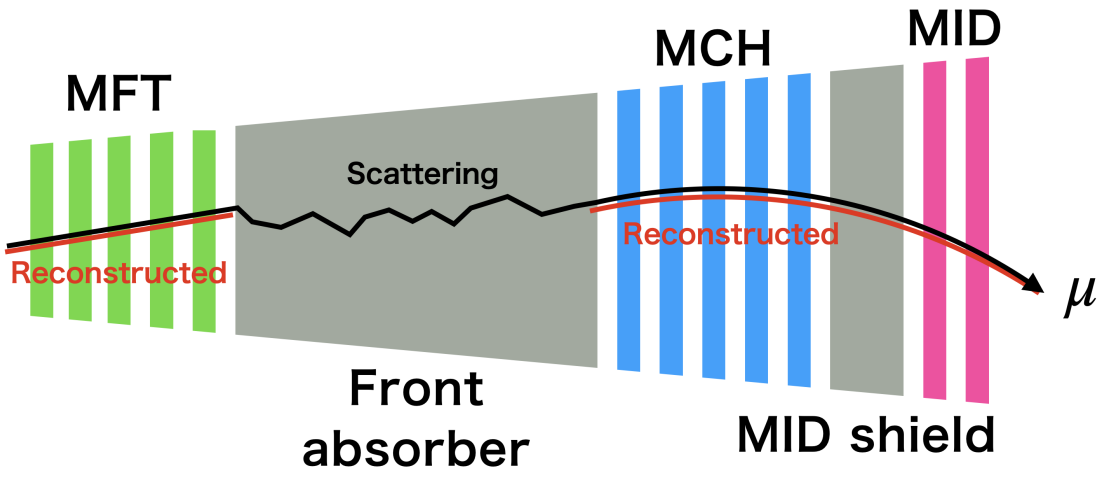


Figure 1.8: The role of MFT is to measure all charged particles before multiple Coulomb scattering in the front absorber. MFT is expected to contribute to improving mass resolution.

MFT consists of 5 discs, each disc consists of 2 half-discs (Figure 1.9), and MFT is located along the beam axis between -460mm ~ -768mm from the impact point[7]. MFT's acceptance is $-3.6 < \eta < -2.5$ to cover the same pseudo-rapidity region as MCH as possible[7].

The detector element is a silicon pixel sensor called ALPIDE, developed for both Inner Tracking System (ITS) and MFT[7]. The multiplicity of charged particles passing through the MFT is very high because the MFT is placed relatively close to the impact point. The efficiency of MFT in detecting charged particles and reconstructing their tracks depends on the spatial resolution of the sensor, the noise-induced fake hit rate, and the level of charge accumulation in the sensor. To get the best performance of MFT, MFT is required to have

the tolerance to radiation and electric power to limit heat dissipation inside and outside the detector. The silicon pixel sensor developed for this purpose is ALPIDE, which has dimensions of $1.5 \times 3\text{cm}^2$ and a pixel pitch of $27 \times 27\mu\text{m}$[7]. A total of 936 ALPIDE chips are used in the MFT, and $2 \sim 5$ ALPIDEs are mounted on a printed circuit board called a ladder, for a total of 280 ladders in the MFT[7]. Each half-disk in the MFT is divided into four zones, each of which is supplied with a common power supply to reduce wiring, and each zone is connected to a single readout unit base (RU)[7].

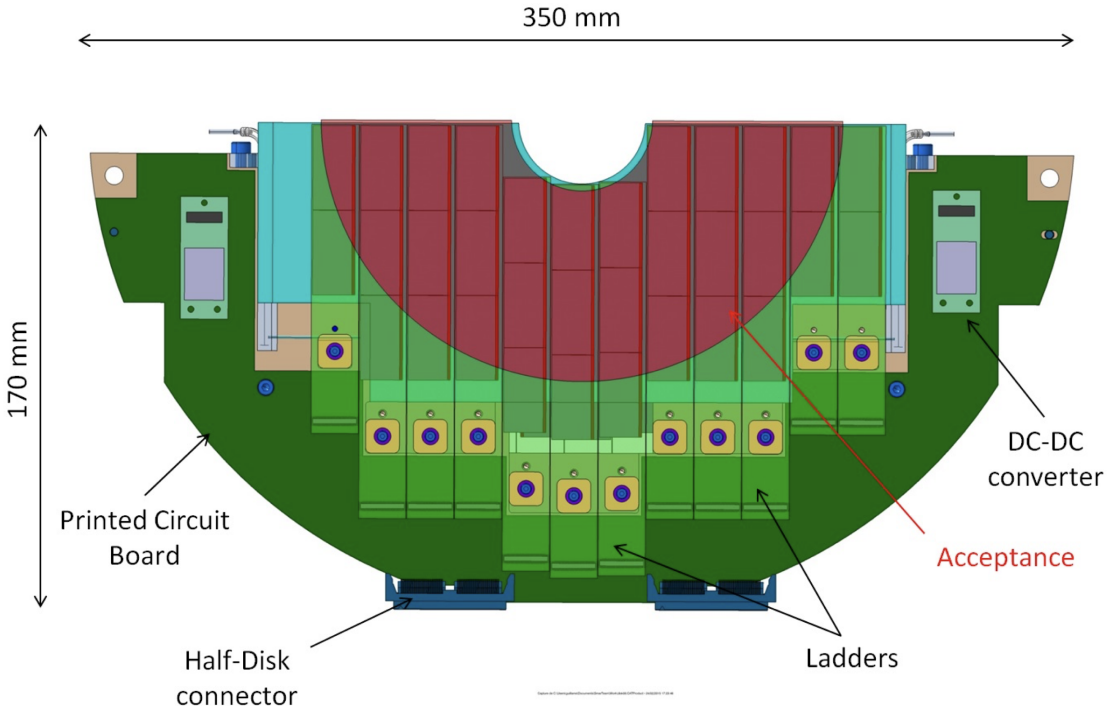


Figure 1.9: Half disk of MFT[7].

1.8.2.1 *MFT standalone tracking*

The particles detected by MFT are charged particles produced by hadron interactions and background events such as very high-multiplicity low-energy QED electrons. The MFT tracks reconstruction must be very accurate since it must also consider fake clusters produced by pixel electron noise that exceeds the threshold used in the readout. There are two algorithms for reconstructing the track in the MFT: the Cellular Automaton algorithm and the Liner Tracking Finder algorithm.

Celluler Automaton algorithm, CA

In Cellular Automaton (CA), detector hits from a single particle are grouped into clusters, and the clusters are pre-paired into smaller units called cells. A cell is defined in MFT as a segment connecting two clusters from two consecutive disks. The cell must point to the point of particle creation within a certain angular range. The CA then considers the given cell and proceeds to the next MFT disk in the direction the particle has traveled, starting with the disk on the side closest to the

MFT collision point and considering the clusters on each disk. In this case, the condition of track continuity is applied based on the angle between the tracks made by the cells on the previous disk and the angle formed by the cells on the next disk. The state of the cell is initially one, but is increased by one each time another compatible cell is added to its end. A specific chain of cells is selected from a large number of cells so that the status of the downstream cells is 4 (Figure 1.10). From such a chain of cells, a candidate track is constructed[7].

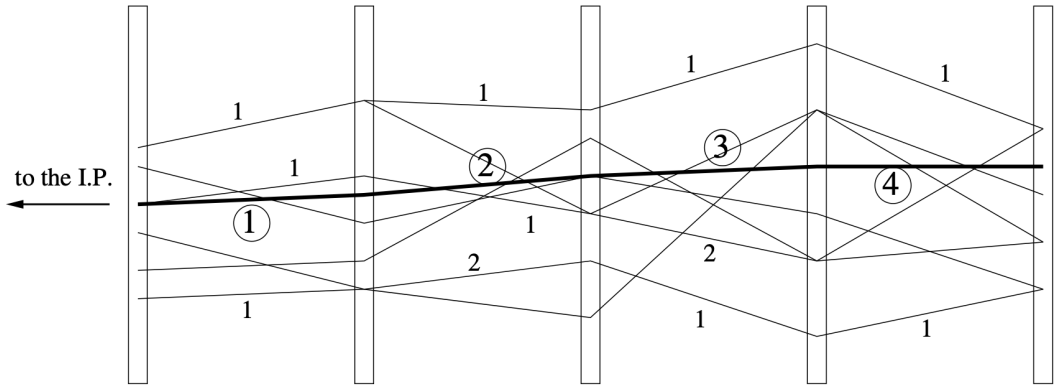


Figure 1.10: Celluler Automaton algorithm, CA: A track is a continuous chain of adjacent cells (segments connecting clusters in a continuous disk), where the status of a cell increases by one unit from one cell to the next. A cell with a status of 4 on the far right indicates that a candidate track of four segments in length starts there[7].

Linear Tracking Finder algorithm, LTF

The Liner Track Finder (LTF) is justified because the magnetic field of the ALICE solenoid magnet is almost parallel to the forward flying particles and is not subject to strong Lorentz forces. LTF requires that the hit information on all disks be linearly fitted. It is obtained from the cluster of disks farthest from the hit point in the MFT as the first point in the tracks. However, the LTF search efficiency is expected to decrease in the low momentum region due to the multiple scattering of particles as they pass through the MFT and the influence of magnetic fields. This has a greater impact than the CA algorithm, which uses only local conditions in the neighborhood. However, it also has the advantage of being computationally faster than CA[7].

A comparison of the track detection efficiency of CA and LTF for transverse momentum and momentum is shown in Figure 1.11. For momentum $p > 8\text{GeV/c}$, the track detection efficiencies of CA and LFT are the same, but the track reconstruction efficiency of CA increases as momentum decreases. On the other hand, CA's procedure for selecting combinations is slower in high particle multiplicity environments. The procedure for track reconstruction is such that LTF is used to find high transverse momentum tracks, and CA is used to search for low transverse momentum tracks using the other hit information[7].

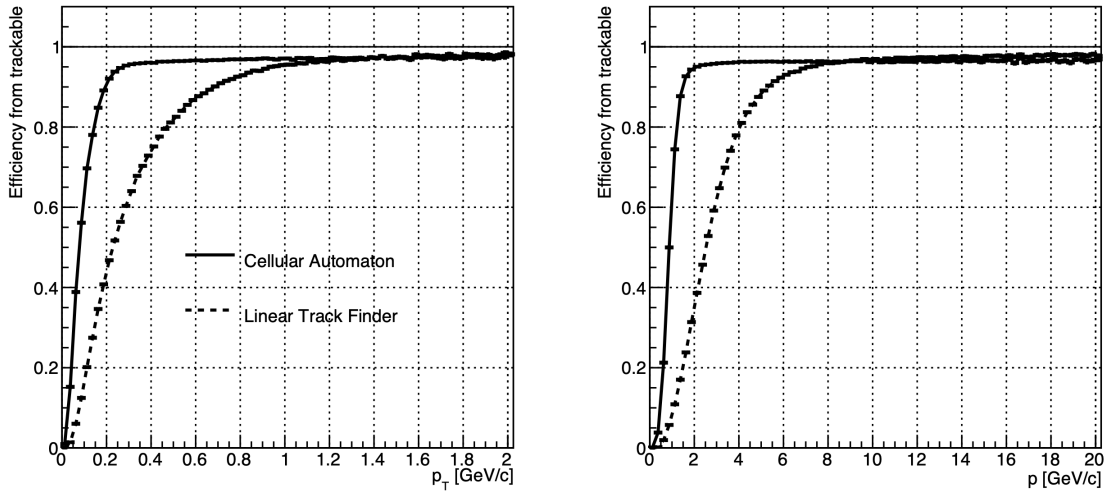


Figure 1.11: Track reconstruction efficiency with respect to transverse momentum (left panel) and with respect to momentum (right panel). Comparison of track reconstruction efficiencies of muons generated by simulations using Cellular Automaton (solid line) and Liner Track Finder (dotted line)[7].

1.8.3 *Global muon tracking*

With the introduction of MFT, it is now possible to measure the tracks of particles before scattering. However, MFT doesn't have the ability of identification whether muon or not. Therefore, we have to use both MFT and MUON spectrometer but this alone does not tell us which tracks reconstructed by the MFT correspond to which tracks reconstructed by the MUON spectrometer; the tracks must be correctly linked to each other in order to measure the MFT and MUON spectrometer together. This is called track matching (Figure 1.12).

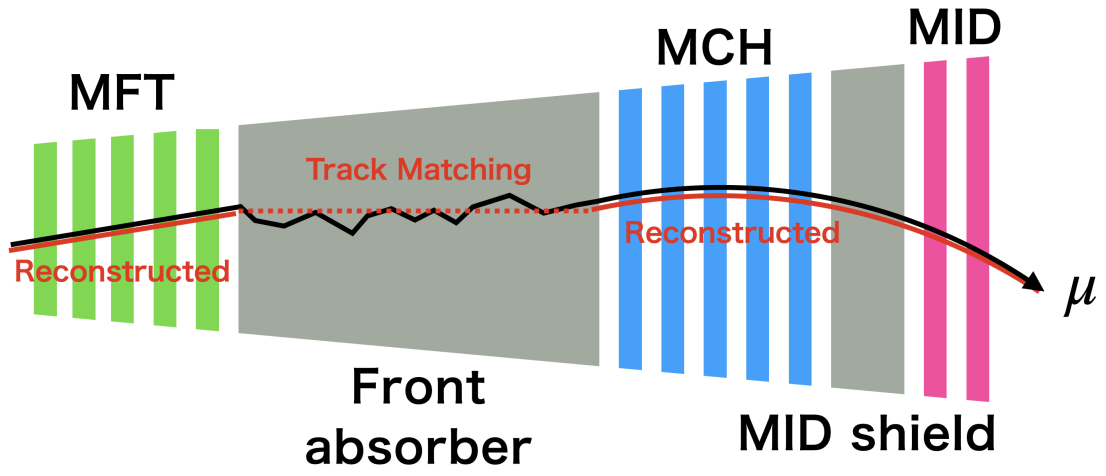


Figure 1.12: GlobalTracking

To perform track matching, set an arbitrary z-plane, the Matching Plane, and extrapolate the tracks reconstructed by the MFT and the tracks reconstructed by the MUON spectrometer to the Matching Plane (Figure 1.13). Then, the parameters of each tracks in the Matching Plane are compared to determine which combination of tracks is correct. The algorithm for determining which combination of tracks is correct is currently being studied within the ALICE experiment.

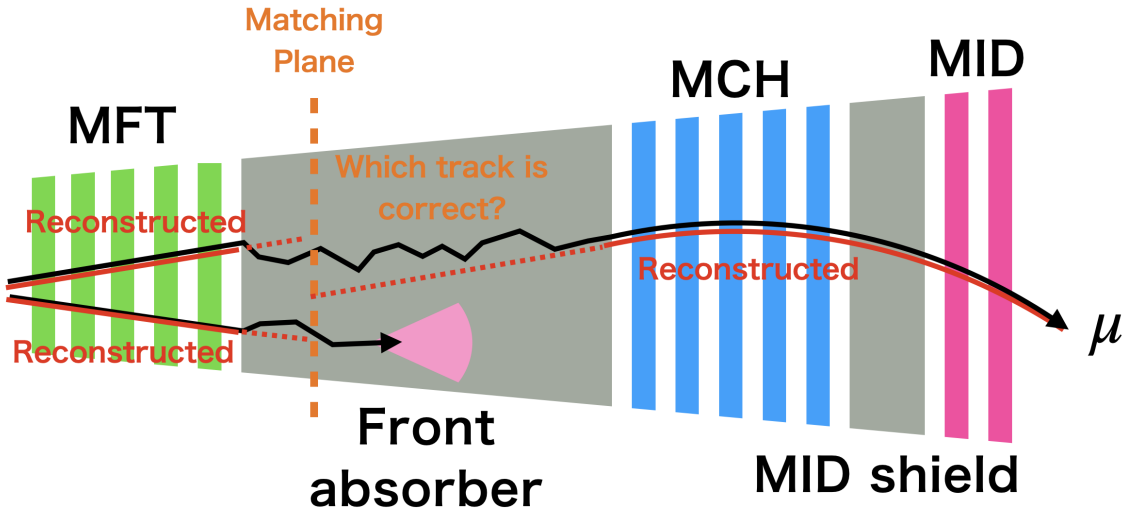


Figure 1.13: Track Matching

1.9 MACHINE LEARNING

Machine learning is a method of analyzing data in which a "machine" (computer) automatically "learns" from the data to discover rules and patterns behind the data. In recent years, the emphasis has shifted to "prediction and decision making" based on the results of the learning process.
In this study, machine learning is used as a one of the method for track matching.
There are three types of machine learning to learn the rules and patterns behind the data: 1) supervised learning, 2) unsupervised learning, and 3) reinforcement learning.

1) supervised learning

Input data and output data (answers) are available to estimate output data from input data. There are several methods to analyze the relationship between inputs and outputs, including regression analysis, a statistical method that is performed mechanically on all combinations of data. A typical example is the prediction of sales based on factors such as weather, prices, and sales promotions.

2) unsurpervised learning

It finds hidden patterns and structures in the background of data from a series of

input data. Compared to "supervised learning", there is no target variable (output data), so data is divided into groups or connections between data are estimated by calculating the closeness or similarity between each piece of data. Clustering methods not used in statistics are often used. A typical example is the logic used to make recommendations in online shopping.

3) reinforcement learning

Unlike supervised/unsupervised learning, this learning method is not based on data from the beginning, but rather on the system's own trial-and-error process to increase accuracy. For example, when considering how to increase the walking distance of a robot, instead of estimating the walking distance from input data, the robot tries out new ways of walking by itself to increase the walking distance, and learns the results to find the optimal way of walking (algorithm and rules). The key point is that the robot learns through trial and error.

In this study, we used supervised learning. Simulate a heavy ion collision experiment and let the computer learn whether the combination of tracks reconstructed by the detector is the correct combination. Since it was a simulation, the computer could tell if its judgment was correct and repeated the learning process to correct its mistakes.

1.9.1 *GBDT*

GBDT (Gradient Boosted Decision Tree) is one of the machine learning algorithms used in various situations in recent years. It is a method that combines gradient descent, ensemble learning with boosting, and decision trees.

Gradient descent

The batch gradient descent method finds the minimum value by focusing on the function's slope, or first derivative. To illustrate the batch gradient descent method, consider a simple linear prediction model. Assume that this model predicts the target variable y by multiplying the explanatory variable (characteristic) x by a certain parameter w. To simplify the model here, we do not consider constants.

$$y = wx \tag{1.41}$$

A model with good prediction performance minimizes the error between the prediction and the true value. In other words, the training data $x1, ..., x_n$ are substituted into this model. Then the model which outputs predictions that minimize the difference between the actual true value $y_1, ..., y_n$ and the prediction $y_1^{\text{pred}}, ..., y_n^{\text{pred}}$ is the best. We define the mean squared error between the prediction and the true value as the loss function L. The loss function L is then written by

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{\text{pred}})^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - wx_i)^2 \tag{1.42}$$

Since the training data never changes, to minimize the loss function L, we can change the value of the parameter w. Here, we first randomly select one value $w^{(0)}$, which is the value of w.Once $w = w^{(0)}$ is determined, the value of the loss function L is also determined; to find out if $L(w^{(0)})$ is minimal when $w = w^{(0)}$, we can differentiate the loss function L at $w = w^{(0)}$ and examine its first derivative. The loss function is a convex quadratic function concerning w. Therefore, if the first derivative $L'(w^{(0)}) < 0$, $w^{(0)}$ is to the left of w giving the minimum value, and if $L'(w^{(0)}) > 0$, $w^{(0)}$ is to the right of w giving the minimum value. If $L'(w^{(0)}) < 0$, then $w^{(1)}$ is the value of $w^{(0)}$ shifted slightly to the right, and $w = w^{(1)}$, and the first derivative of the loss function can be examined again. By repeating this operation a sufficient number of times, w can be found such that the first derivative of the loss function L is 0.

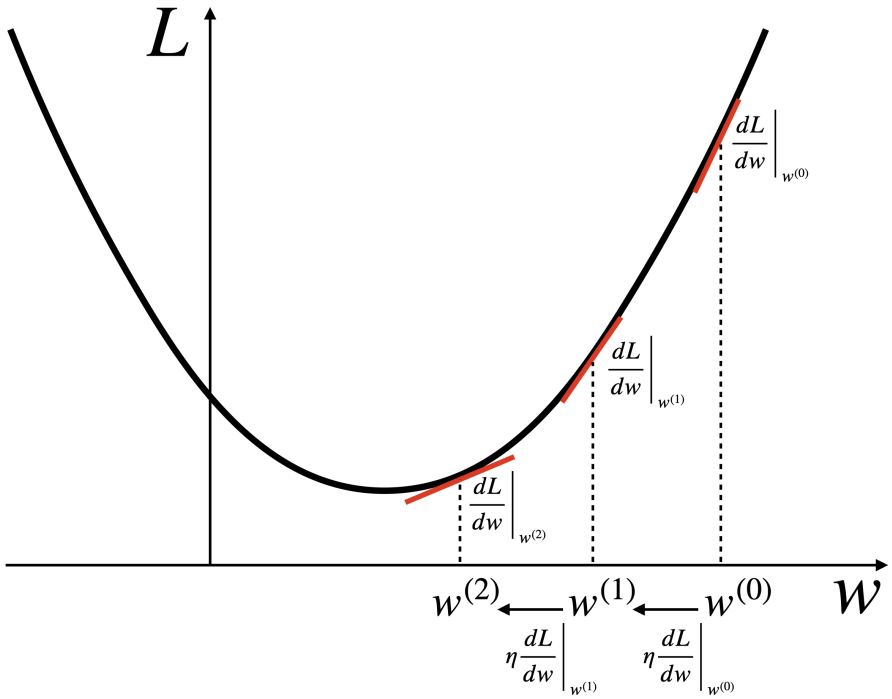


Figure 1.14: Gradient descent

When the first derivative L' of the loss function L is not zero, the value of w is moved slightly to the right or slightly to the left, depending on the sign of L', to approach the optimal w. This process is called learning. Various methods have been proposed to determine how much learning is actually required. The most basic method is to learn as follows. In this case, η $(0 < \eta < 1)$ is the learning rate.

$$w^{(j+1)} = w^{(j)} - \eta \left.\frac{dL}{dw}\right|_{w^{(j)}} \tag{1.43}$$

Boosting

Boosting is one of the methods of ensemble learning. Ensemble learning is the combination of several weak learners to create a strong learner. In boosting, weak learners

are connected in series: the first weak learner calculates the prediction of the objective variable from the explanatory variables; the second weak learner learns to reduce the loss of the first weak learner by turning this loss into the objective variable (Figure 1.15). Repeating this process is boosting, and boosting learns to focus on improving the areas where it made mistakes in its predictions.

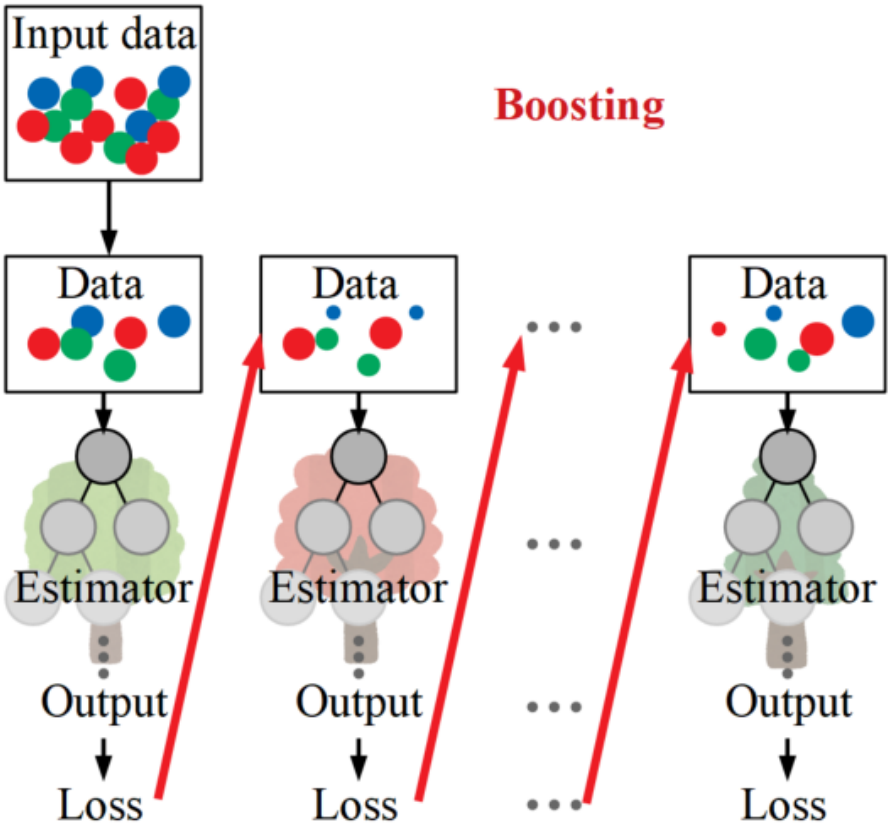


Figure 1.15: Boosting

Decision Tree

Decision tree is a machine learning technique that finds clumps of data in which specific characteristics often appear and generates classification rules for them. Expressly, the objective variable and explanatory variables are set, and data groups are found in which the characteristics of the objective variable are clumped together, and the concentration of the characteristics of the objective variable within the data area is increased by dividing the data using multiple explanatory variables. In other words, we divide the data by combining the conditions of the explanatory variables so that the characteristics of the objective variable are biased as much as possible, in other words, so that the variation of the data within the region is reduced. A decision tree is a method to generate a tree structure of branching rules (If-Then conditional rules) for the data consisting of the conditions of the explanatory variables obtained in this way (Figure 1.16).

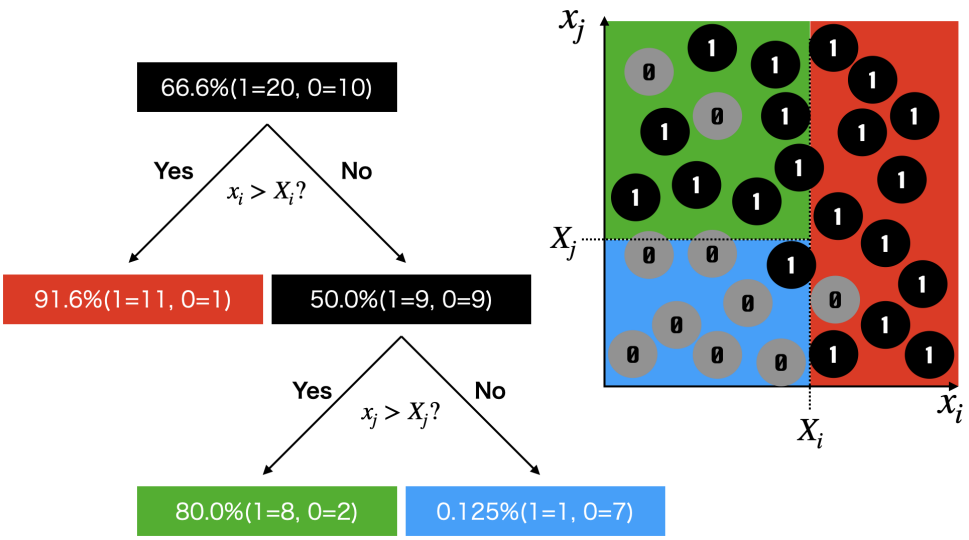


Figure 1.16: Decision Tree

The GBDT combines all three of these learning methods.
Consider the following data set.

$$\begin{pmatrix} x_1 \\ \vdots \\ x_i \\ \vdots \\ x_n \end{pmatrix}, \begin{pmatrix} y_1 \\ \vdots \\ y_i \\ \vdots \\ y_n \end{pmatrix} \tag{1.44}$$

For this data set, consider finding a function $F(x)$ that best approximates y by entering x. We treat this as a minimization problem for the loss function $L(y, F(x))$. First, consider the function $F(x)$ to be the sum of functions $h_m(x)$ [8].

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) + const. \tag{1.45}$$

The first function is initialized with a constant γ that minimizes the loss function [8].

$$F_0(x) = \underset{\gamma}{\operatorname{argmin}} \sum_{i-1}^{n} L(y_i, \gamma) \tag{1.46}$$

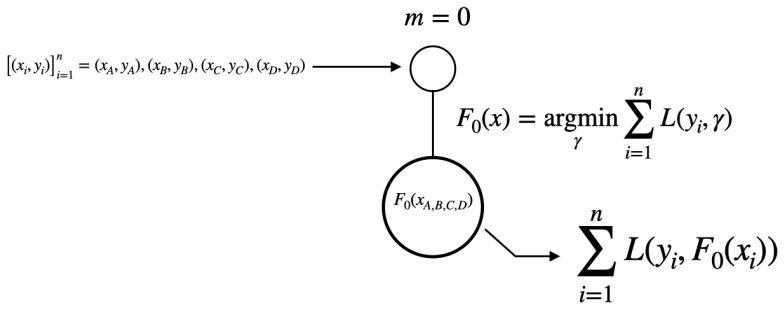


Figure 1.17: First round of GBDT.

The second and subsequent functions are then considered as follows [8].

$$F_m(x) = F_{m-1}(x) + \underset{h_m(x)}{\text{argmin}} \left[\sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + h_m(x_i)\right) \right] \tag{1.47}$$

The first term roughly predicts y using the l, and the second term predicts the loss of that prediction with the function $h_m(x)$. In this way, the loss is reduced by repeatedly adding a term to the prediction of one previous model that corrects for the loss of that prediction. This is precisely what boosting is. Now, to consider what kind of function $h_m x$ should be, we use the gradient descent method. We first find the derivative of the loss function in one previous prediction [8].

$$\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \tag{1.48}$$

The function $F_m(x)$ is then determined as follows [8].

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)} \tag{1.49}$$

where γ is a positive parameter that minimizes the loss as follows

$$\gamma_m = \underset{\gamma_m}{\text{argmin}} \sum_{i=1}^{n} L\left(y_i, F_m(x_i)\right) \tag{1.50}$$

$$= \underset{\gamma_m}{\text{argmin}} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) - \gamma_m \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right) \tag{1.51}$$

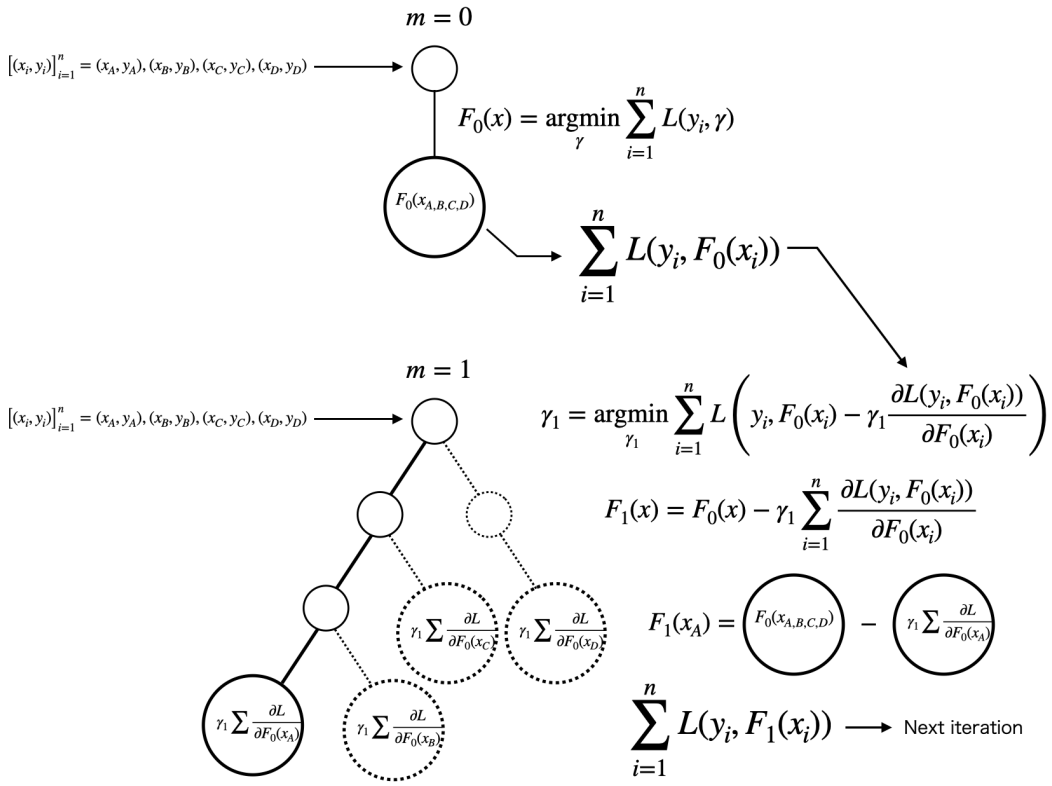


Figure 1.18: Second round of GBDT.

For the mth prediction, we want the loss to be smaller than the $m-1$th prediction. To do so, we calculate the derivative of the loss function and update the prediction so that it is down the slope of the loss function. The update is done as follows: $F_m(x) = F_{m-1}(x) - \gamma_m \partial L_{m-1} / \partial F_{m-1}$. In this case, γ_m is determined so that the loss of $F_m(x)$ is minimized.

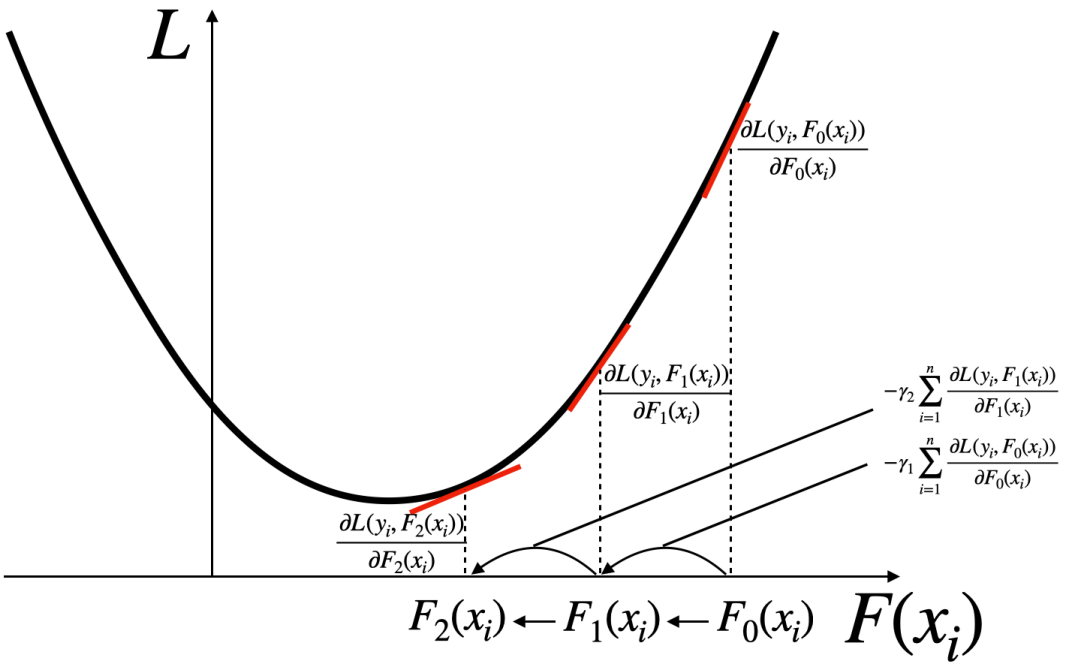


Figure 1.19: Predicted probability by GBDT.

1.10 KALMAN FILTER

The Kalman filter is an algorithm for estimating the next state value from one previous state value using a model and observations [9]. The Kalman filter assumes that both the state values estimated by the model and the observations have noise that follows a Gaussian distribution [9].

$$q_t^{odo} = Fq_{t-1}^{odo} + \omega q_t^{obs} = q_t^{odo} + v \tag{1.52}$$

where q_t^{odo} represents the state value estimated by model F at time t, q_t^{obs} represents the observed value at time t, and ω and v represent the model noise and observed noise, respectively. Because each state value and observed value estimated by the model has noise, both values deviate from the true state value. Therefore, the Kalman filter uses a matrix K_t called Kalman gain to weigh how much to trust either information to estimate more accurate state values [9].

$$q_t^{est} = (I - K_t H_t) q_t^{odo} + K_t q_t^{obs} \tag{1.53}$$

where q_t^{est} is the state value estimated using the Kalman filter at time t, and H_t is a Jacobian-like matrix that allows the state q^{odo} estimated by the model at time t to be treated like the measured value. The state at each time is estimated by repeating this process at each time, starting from the first time. In addition to estimating the central state value, the Kalman filter also estimates a matrix called the estimation error covariance matrix from the state

and observed values at the previous time as well. This covariance matrix represents the certainty of the estimated state value at that time. Since the Kalman gain is determined so that the covariance matrix is small, the state value can be estimated plausibly.

1.11 PURPOSE OF THIS STUDY

The main goal of this study is to improve track matching performance. We measure the decay of light vector mesons, represented by rho, omega, and phi, into dimuons to verify the relationship between hadron mass and chiral symmetry, as mentioned before. However, the branching ratio of this decay mode is meager. Therefore, a high signal-to-noise ratio and mass resolution are required. Currently, ALICE employs a Kalman filter-based algorithm for track matching, but we propose a method using machine learning in this study. We hope to obtain cleaner invariant mass spectra.

2

APPLICATION OF MACHINE LEARNING TO μ TRACK RECONSTRUCTION WITH MULTIPLE DETECTORS

2.1 ISSUES WITH THE ALICE FORWARD μ PARTICLE DETECTOR

As mentioned above, ALICE installed MFT from Run3, enabling the detection of μ particle tracks with high positional resolution. However, the MFT is a detector that covers the forward rapidity region ($-3.6 < \eta < -2.5$), and since a magnetic field is applied in the z-axis direction in the region where the MFT exists, particles flying in the forward direction where the MFT exists are almost always parallel to the magnetic field and are hardly affected by Lorentz force. Therefore, the track detected by the MFT is almost straight and not curved, making it difficult to measure momentum. Another forward μ detector is the MCH, which has a dedicated magnet that bends the track due to the Lorentz force and can measure momentum from the curvature of the track. However, since MCH measures particles that have passed through a material layer as long as 4 m for μ identification, the position resolution is low due to multiple Coulomb scattering in the material layer.

To investigate the relationship between the spontaneous breaking of chiral symmetry and hadron masses, it is necessary to measure hadron masses, so it is necessary to reconstruct the invariant masses of the parent particles from the μ pairs formed by the decay of hadrons. We can calculate the invariant mass using the following equation

$$M^2 = m_{\mu_1}^2 + m_{\mu_2}^2 + 2(E_{\mu_1}E_{\mu_2} - p_{\mu_1} \cdot p_{\mu_1}) \tag{2.1}$$

$$= m_{\mu_1}^2 + m_{\mu_2}^2 + 2\left(\sqrt{m_{\mu_1}^2 + p_{\mu_1}^2}\sqrt{m_{\mu_2}^2 + p_{\mu_2}^2} - |p_{\mu_1}||p_{\mu_2}|\cos\theta\right) \tag{2.2}$$

Where θ is the angle between the momentum vectors of the μ pairs, this equation includes momentum and angle, and invariant masses can be calculated by measuring them. As mentioned earlier, MFT has excellent position (angle) resolution but poor momentum resolution. On the other hand, MCH has excellent momentum resolution and low position (angle) resolution. Therefore, combining these two detectors is necessary to reconstruct invariant masses with high accuracy.

To use the two detectors in combination, it is necessary to connect the tracks detected by each detector correctly. In following this is called track matching. The following two

issues must be addressed to perform track matching.

i) The number of particles that pass through the MFT is huge, about 2000∼3000 per event.

ii) Multiple Coulomb scattering inside the material layer for μ identification causes blurring of position and momentum while passing through the material layer.

With these issues, the performance of how well track matching can be done depends on the track matching algorithm, and ALICE has been studying algorithms for track matching. In this paper, we describe machine learning-based track matching in detail.

2.2 TRACK MATCHING USING KALMAN FILTER

The track matching algorithm previously proposed in ALICE is based on the track finding algorithm with Kalman filter[10]. The Kalman filter was a method of estimating the actual state value based on the state value at a certain time t, using a model to estimate the state value at the next time $t+1$, and two measured values at time t+1. In track matching, the tracks reconstructed by MFT and MCH are extrapolated to the matching plane. Extrapolating the track reconstructed by MCH means that the track's state values (position, angle, momentum, etc.) on the matching plane are estimated from the track reconstructed by MCH using a physical model. The track reconstructed by MFT is then treated as an observed value on the matching plane. At this time, the covariance matrix is calculated for each pattern to determine which track reconstructed by MFT should be matched with the track reconstructed by MCH the most. The track with the most minor residue is selected.

$$\chi^2 = r_{k|k-1}^T R_{k|k-1}^{-1} r_{k|k-1} \tag{2.3}$$

$$r_{k|k-1} = m_k - h_k(q_{k|k-1}) \tag{2.4}$$

$$R_{k|k-1} = V_k + H_k C_{k|k-1} H_k^T \tag{2.5}$$

where $r_{k|k-1}$ is the residue of the state values of the MFT-reconstructed and MCH-reconstructed tracks, and $R_{k|k-1}$ is the residue of the covariance matrix. m_k and V_k are the parameters and covariance matrix of the MFT-reconstructed tracks in the matching plane, which are generally $q_{k|k-1}$ and $C_{k|k-1}$ are the parameters. Covariance matrices of the MCH reconstructed tracks in the matching plane are estimated by the model from the state at $k-1$ in the general Kalman filter k, which corresponds to the state at k in the general Kalman filter. In addition, $h_k(q)$ and H_k are functions and matrices to treat the predictions the same as the observed values in the general Kalman filter. Still, since the parameters and covariance matrices are the same for the MFT-reconstructed and MCH-reconstructed tracks, the algorithm uses $h_k(q)$ and H_k are unitary matrices.

2.3 TRACK MATCHING BY CLASSIFICATION USING GBDT

Here, machine learning is used to pair the MFT-reconstructed tracks with the MCH-reconstructed tracks and classify the pairs as binary, i.e., whether they are correct combinations consisting of the same particles or incorrect combinations that are not.
There are various types of machine learning methods, such as DNN, random forests, etc. However, since the features to be handled in this study are based on a table of numerical data, such as the momentum and position of the tracks, we decided to use GBDT, which is currently considered very accurate when handling table data.

2.3.1 *data-set*

The data set used for this training is the following table data. (Note that the data below is an example, and the numbers differ from those used.)

Table 2.1: data-set

No.	1	2	3	4	...	1001	1002	1003	1004	...
X_{MFT}	5.4	-7.4	1.3	-4.2	...	5.4	-7.4	1.3	-4.2	...
Y_{MFT}	13.1	-21.4	5.7	4.3	...	13.1	-21.4	5.7	4.3	...
ϕ_{MFT}	1.4	4.8	1.5	2.2	...	1.4	4.8	1.5	2.2	...
$\tan\lambda_{MFT}$	3.2	9.8	-32.4	15.4	...	3.2	9.8	-32.4	15.4	...
$q/p_{T,MFT}$	3.5	-0.4	-2.1	4.9	...	3.5	-0.4	-2.1	4.9	...
X_{MCH}	42.9	42.9	42.9	42.9	...	21.4	21.4	21.4	21.4	...
Y_{MCH}	52.5	52.5	52.5	52.5	...	-11.9	-11.9	-11.9	-11.9	...
ϕ_{MCH}	1.5	1.5	1.5	1.5	...	5.3	5.3	5.3	5.3	...
$\tan\lambda_{MCH}$	4.1	4.1	4.1	4.1	...	25.2	25.2	25.2	25.2	...
$q/p_{T,MCH}$	3.6	3.6	3.6	3.6	...	-0.2	-0.2	-0.2	-0.2	...
ΔX	-37.5	-50.3	-41.6	-47.1	...	-16.0	-28.8	-20.1	-25.6	...
ΔY	-39.4	-73.9	-46.8	-48.2	...	25.0	-9.5	17.6	16.2	...
...	...	...	...	...	...	...	...	...	...	...
Ratio X	0.13	-0.17	0.03	-0.10	...	0.25	-0.35	0.06	-0.20	...
Ratio Y	0.25	-0.41	0.11	0.08	...	-1.10	1.80	-0.48	0.36	...
...	...	...	...	...	...	...	...	...	...	...
MFT χ^2	0.1	2.0	0.4	0.9	...	0.1	2.0	0.4	0.9	...
MFT $N_{cluster}$	3	12	8	3	...	3	12	8	3	...
Truth	1	0	0	0	...	0	0	1	0	...

All tracks detected by MFT and MCH, respectively, are extrapolated to some arbitrary z-plane (matching plane) and learned by comparing the parameters of each track on the matching plane. $X_{MFT}, Y_{MFT}, X_{MCH}, Y_{MCH}$ are the position coordinates of tracks detected by MFT and MCH on the matching plane, ϕ_{MFT}, ϕ_{MCH} are the azimuth angle of tracks detected by MFT and MCH, $\tan\lambda_{MFT}, \tan\lambda_{MCH}$ is the tangent of the angle between the x,y plane and the tracks detected by MFT and MCH respectively, $q/p_{T,MFT}, q/p_{T,MCH}$ is the charge of the tracks detected by MFT and MCH respectively divided by the transverse momentum of the tracks. Those with Δ are the difference of $X, Y, \phi, \tan\lambda, q/p_T$ at MFT and MCH, respectively, and those with Ratio are the ratio. Also, MFT χ^2 is a quantity that represents the quality of the reconstruction of the tracks reconstructed by MFT. And Truth is the quantity that represents whether the combination of tracks is correct, where 0 represents a wrong combination, and 1 represents a correct combination.
The coordinate system used in this study has the following coordinate system as shown in the following figure[11].

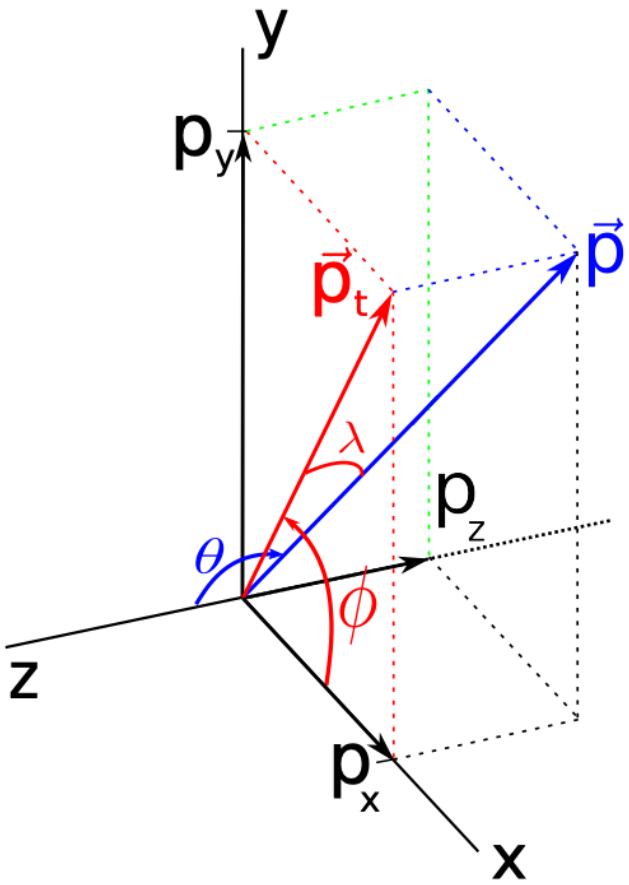


Figure 2.1: The coordinate system used in this study[11].

In the data example shown above, 1000 tracks are reconstructed by MFT in one event, and these 1000 tracks are compared with each track reconstructed at MCH. In only one event, the number of tracks in the MFT ($\mathcal{O}(10^3)$) and the number of tracks in the MCH ($\mathcal{O}(1)$) are not equal. The number of positive examples is (almost) equal to the number

of tracks in MCH, so the amount of data is too small for machine learning to capture the features. Therefore, data from multiple events are merged into a single table data. In this study, $\sqrt{s_{NN}} = 5.5$TeV, 500 events of minimum-bias Pb-Pb collisions were prepared as training data. We then allocated 70% of them as training data, 15% as validation data, and 15% as test data.

2.3.2 *Handling imbalance data*

There is only one correct combination of tracks per particle, and all other combinations are wrong. Therefore, in the present data set, the number of samples for each class of teacher data is very different, and the number of samples for the correct combinations is less than the number of samples for the wrong combinations. The number of samples for the correct combinations is only about 1/80th of the number of samples for the wrong combinations. Suppose that in such an unbalanced case, learning proceeds in such a way as to maximize the percentage of simple correct answers, for example, as follows.

$$\text{logloss} = y_{\text{true}} \log y_{\text{pred}} + (1 - y_{\text{true}}) \log(1 - y_{\text{pred}}) \tag{2.6}$$

If the number of samples in each class is approximately equal, learning proceeds without problems, but if there is an imbalance, even if the prediction for the minority class is wrong, the losses will only be small when added together because it is a minority class. On the other hand, since the majority class is large in number, if the loss of the majority class can be reduced, the combined loss can be reduced to a small loss. In other words, even if the minority class makes a mistake, its penalty can be ignored because it is small.
Suppose there is one combination of correct tracks and 99 combinations of wrong tracks, and a machine learning model predicts that they are all wrong. In that case, we can say that the model has 99% prediction accuracy because 99 combinations of tracks are successfully predicted. If the loss is low and the evaluation index is high at that time, machine learning will stop learning there.
However, in the case of track matching, it is useless if learning stops in such a situation, since the key is to determine that the correct combination of tracks is the correct one for the correct combination of tracks.
Therefore, we tried each of the following measures for unbalanced data.
i)PR-AUC is used as the evaluation index.
ii)Downsampling of training data[12].
iii)Weighted learning.
iv)Loss functions such as Focal loss are used[13].

The PR-AUC is defined as the area bounded by the Precision-Recall curve, but a confusion matrix is shown below first to explain the definitions of Precision and Recall.

Table 2.2: Confusion matrix

	Positive (prediction)	Negative (prediction)
Positive (truth)	TP: True Positive	FN: False Negative
Negative (truth)	FP: False Positive	TN: True Negative

In standard binary classification, a good learning model can classify positive examples as positive and negative examples as negative. Hence, a model with more TP and TN and less FN and FP is ideal. (However, this is not always the case in track matching, which will be explained in the explanation of learning by ranking). First, we show Accuracy as a representative evaluation index.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{TN} + \text{FN}} \tag{2.7}$$

It is the percentage of all predictions that are correctly classified. The closer this value is to 1, the better, which is the same 99% prediction accuracy mentioned above for one positive sample and 99 negative samples. In the case of such unbalanced data, if we predict that all the data are negative samples, 99% of them will be TN and 1% will be FN, resulting in an accuracy of 0.99, which is an easy way to obtain high accuracy. Still, a model that predicts that all the data are negative samples is useless. Therefore, Accuracy is not a good measure of the quality of a classification model for unbalanced data.
Next, Precision and Recall are explained.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2.8}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{2.9}$$

Precision represents the percentage of those predicted to be positive samples that are really positive samples, and Recall represents the percentage of those whose true values are positive samples that are predicted to be positive samples. Since Precision is a quantity like purity and Recall is an efficiency measure, they are trade-offs. Since these two measures do not use TN in common, they do not evaluate the correctness of predictions about negative samples, as is the case with Accuracy.
As mentioned in the explanation of Accuracy, when data are unbalanced, they are often classified as all being in the majority class, which tends to result in a forecast with a large number of FNs when there is an overwhelming number of negative samples, as in the present case. Therefore, it would be good to aim for a model that increases the Recall that includes FNs in the denominator. However, if one wants to increase the value of Recall, it would be better to consider a model that classifies all of the cases as positive samples. By looking at Precision simultaneously, it is possible to prevent a model that predicts all of the samples as positive samples since Precision is only 1% for 1:99 data. Therefore, it is necessary to look at these two metrics simultaneously.

These Precision and Recall are calculated with the TP, FP, TN, and FN values shown in the confusion matrix. To begin with, these numbers are counted by considering a threshold value for the prediction. In machine learning classification, the probability that a model is a positive sample for a given data is calculated. Then, an arbitrary threshold is set, and if the probability is higher than the threshold, it is classified as a positive sample, and if it is lower than the threshold, it is classified as a negative sample. This determines the number of TP, FP, TN, and FN. Therefore, if the threshold value is moved, these four values will also change, and the Precision and Recall values will also change. If, as in this case of track matching, there is no particular restriction on the threshold value, and it is simply sufficient to classify correctly, we would like to use an evaluation index that does not depend on the threshold value. This is where PR-AUC comes in, which, as mentioned earlier, uses a Precision-Recall curve. This curve is drawn by plotting Precision and Recall while moving the threshold value. The area under this curve is called PR-AUC.

Next, downsampling is explained. Downsampling is randomly thinning out too many negative samples and reducing them to the same number as positive samples[12]. By learning with down-sampled data, the ratio of positive and negative samples is reduced to 50% each so that even if Accuracy is used as an evaluation index, as in the case of unbalanced data, the model will not predict that all cases are negative. Both positive and negative samples can be learned equally. However, down-sampling is intended to eliminate imbalance during training so that the minority class is learned with the same importance as the majority class. Still, the binary classification problem the trained model faces is naturally imbalanced, so the validation and test data must remain unbalanced. It is also known that training on down-sampled data can introduce a bias in the prediction probability[14]. This is because when training on down-sampled data, especially near the boundary between positive and negative samples, the density of the distribution of the negative sample class is lower than the distribution before sampling, making the distribution of the positive sample class appear to be wider than it originally was, resulting in a higher prediction probability of being a positive sample. The bias can be corrected (calibrated) from the biased prediction probability $p(y|x, s = 1)$ using the following equation[14].

$$p(y = 1|x) = \frac{\beta p(y|x, s = 1)}{\beta p(y|x, s = 1) - p(y|x.s = 1) + 1}, \beta = p(s = 1|y = 0) \tag{2.10}$$

where x is the explanatory variable and y is the objective variable. Also, s is a number called the sampling variable that takes one if some data before sampling are still included after sampling and 0 if not. In actual use, we need to find β, and since it is the probability that negative sample data remain after down sampling, we can say $\beta = p(s = 1|y = 0) \simeq N^+/N^-$.
Downsampling allows positive samples to be important even when they are unbalanced during training, and calibration allows bias-free predictions to be made for unknown data[14].

In the following figures, the horizontal axis shows the predicted probability, and the vertical axis shows the percentage of positive samples in the data with that predicted probability. The orange histogram shows the distribution of the prediction probability. When the prediction probability is 1, the data is predicted to be a positive sample with certainty, so all such data should also have a true positive value. When the prediction probability is 0.5, the prediction is half and half, so only half of the data with such a prediction should have a true positive value. Thus, when there is data that gives a predicted probability of being a positive sample, the percentage of positive examples in that data should be equal to the predicted probability. However, if only down-sampling is performed, the predicted probability and the percentage of positive samples in the data will not match, as shown below.

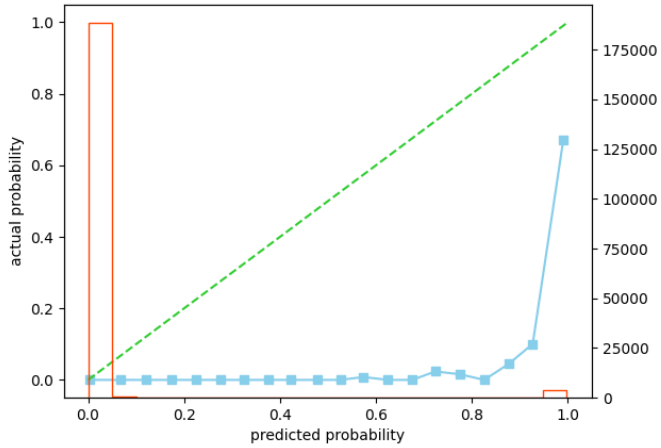


Figure 2.2: Calibration plot of the prediction with downsampling only, no calibration.

In contrast, with the calibration shown above,

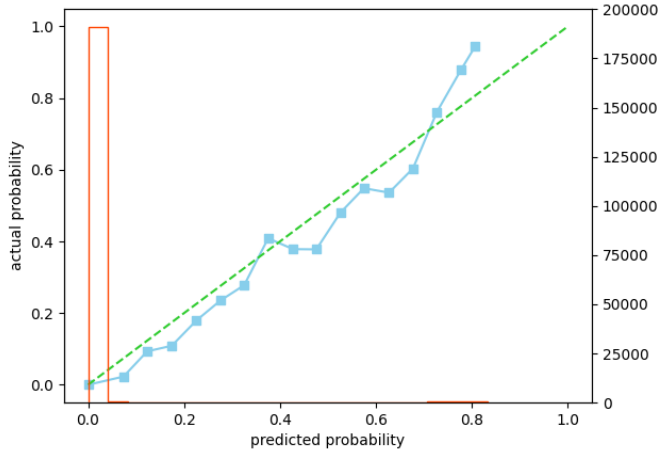


Figure 2.3: Calibration plot of the prediction with calibration after downsampling

This is what happened, and the bias was removed. Let's look at the PR-AUC in each case to see how much the downsampling and calibration improved the classification performance.

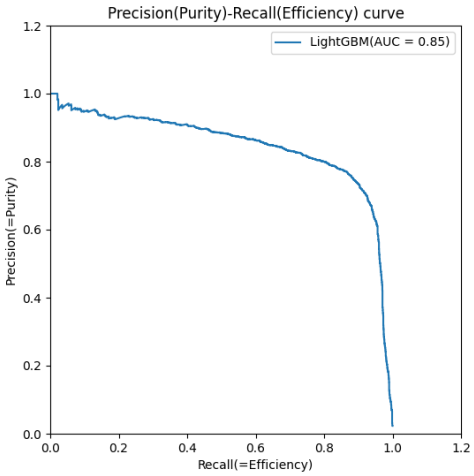


Figure 2.4: Precision-Recall curves of classification for $\sqrt{s} = 14$TeV minimum bias pp collision data.

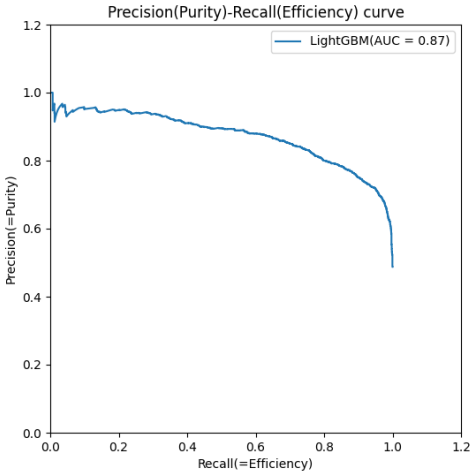


Figure 2.5: Precision-Recall curves when trained on downsampled pp collision data and calibrated for predictive probability.

PR-AUC was higher with downsampling. In other words, the classification performance for positive samples improved. However, down-sampling has drawbacks. Since the training data is randomly thinned out, if the thinned out data is important, the feature cannot be learned, and since the data size is simply reduced, prediction for unknown data may become difficult. Therefore, the comparison here is not about down-sampling, but rather about checking how much the prediction on unbalanced data improves when the positive and negative samples are aligned by sampling. We have aligned the total number of data before and after sampling to avoid the influence of differences in total data volume.

Next, we will discuss weighted learning. In downsampling, the data are directly edited to allow learning without neglecting minority groups. In weighted learning, instead of editing the data, we add weights to the loss function at the time of learning. Even if a prediction for a negative sample, which is the majority, is missed by a large amount, the loss is not large, but if a prediction for a positive sample, which is the minority, is missed by a large amount, the loss is larger than for a negative sample, even if the same amount is missed. This allows learning without editing the data set and without neglecting positive samples. For example, consider weighting with a general loss function called logloss.

$$\text{logloss} = y_{\text{true}} \log y_{\text{pred}} + (1 - y_{\text{true}}) \log(1 - y_{\text{pred}}) \tag{2.11}$$

The logloss with weighting taken into account is as follows

$$\text{weighted logloss} = \frac{y_{\text{true}}}{N^+} \log y_{\text{pred}} + \frac{(1 - y_{\text{true}})}{N^-} \log(1 - y_{\text{pred}}) \tag{2.12}$$

The weighting was done by the reciprocal of the number of elements within each class. This results in a larger loss for positive samples (N^+ is small) and a smaller loss for negative samples (N^- is large).
In this case, as with down-sampling, it is necessary to weight the loss function only during training with training data and to treat the unweighted loss function during validation. The Precision-Recall curve for the case of weighted learning is shown below.

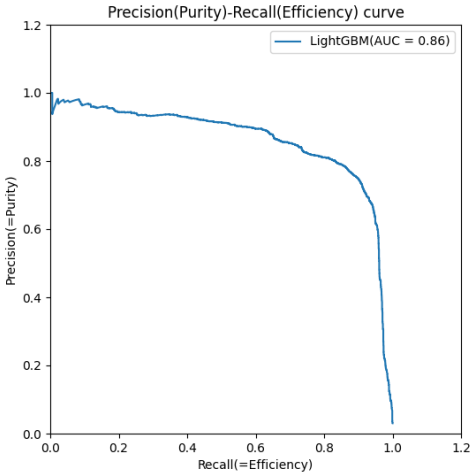


Figure 2.6: Precision-Recall curve for weighted learning with downsampled pp collision data.

In this case, the PR-AUC was also improved, but not as much as in the downsampling case. Comparing the three cases of learning without any modification (here called the original), downsampling, and weighted learning, we see that following table.

Table 2.3: Results of measures for imbalance data

	original	down sampling	weighted learning
Number of positive samples in training data	55	2290	55
Number of negative samples in training data	4525	2290	4525
Total amount of training data	4580	4580	4580
Loss function	logloss	logloss	weighted logloss
metric for early stopping	PR-AUC	PR-AUC	PR-AUC
Precision when the threshold is 0.5	0.826	0.762	0.747
Recall when the threshold is 0.5	0.729	0.889	0.902
PR-AUC	0.847	0.873	0.862

In the original, Precision is high and Recall is low. This means that there are fewer FPs and more FNs. In other words, there are more cases where a negative sample is wrongly predicted to be a negative sample when the true value is a positive sample than vice versa. On the other hand, in the case of downsampling and weighted learning, Recall is higher than Precision, meaning that there are more cases where a negative sample is predicted to be a positive sample. The trend observed in the original (Precision>Recall) is precisely due to unbalanced data. Because of the overwhelming number of negative samples, even if we apply early termination using PR-AUC as the evaluation index, the loss function is the usual logloss, so learning proceeds in the direction of reducing the logloss. As mentioned in the introduction, it is less lossy to predict a negative sample even at the expense of a few positive samples, which are in the minority. This is the reason why this trend was observed. In the down-sampling process, learning can proceed without being affected by imbalance because the distribution of positive samples is larger than in the normal case. When predictions are then made for unknown unbalanced data, most of the prediction probabilities are slightly below the green dotted line in Figure 2.3 (= overall prediction probabilities tend to be too high), although the effect is suppressed due to the calibration of the prediction probabilities. This trend should diminish as the calibration becomes cleaner. The weighted learning uses a loss function normalized by the number of data, but we explained that the challenge in track matching is not only imbalance (high multiplicity due to high energy Pb-Pb collisions), but also multiple Coulomb scattering inside hadron absorbers located between the MFT and MCH. As shown below, when track matching is performed by machine learning, the most essential feature is the difference in position on the matching plane ($\Delta X, \Delta Y$).

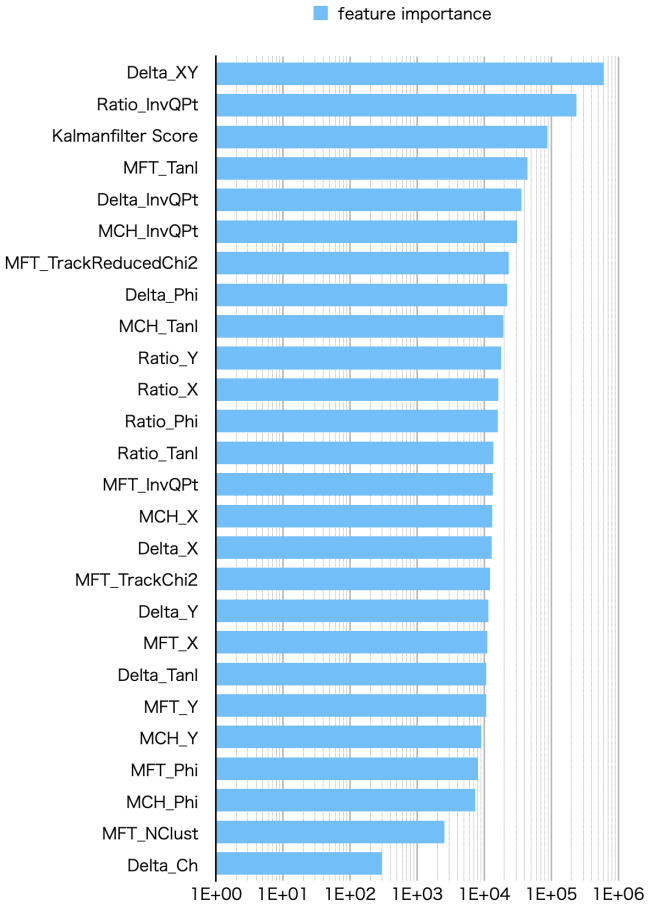


Figure 2.7: Feature importances in track matching

The width of the position spread due to multiple Coulomb scattering is then expressed by the following equation[15]

$$\theta_0 = \frac{13.6\text{MeV}}{\beta cp} z \sqrt{\frac{x}{X_0}} \left(1 + 0.038 \ln \frac{x}{X_0;}\right) \tag{2.13}$$

$$y_0 = \frac{1}{\sqrt{3}} x\theta_0 \tag{2.14}$$

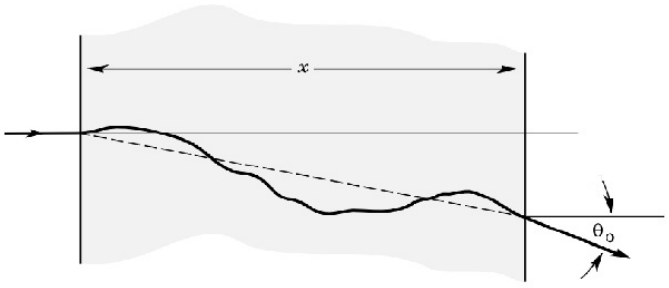


Figure 2.8: Multiple Coulomb scattering effects[15].

where X is the length of the scatterer and X_0 is the radiation length of the scatterer. As can be seen from this equation, for multiple Coulomb scattering, the scattering angle and position spread are larger at lower momentum. The transverse momentum distribution of charged particles in the MFT and MCH acceptances is shown below.

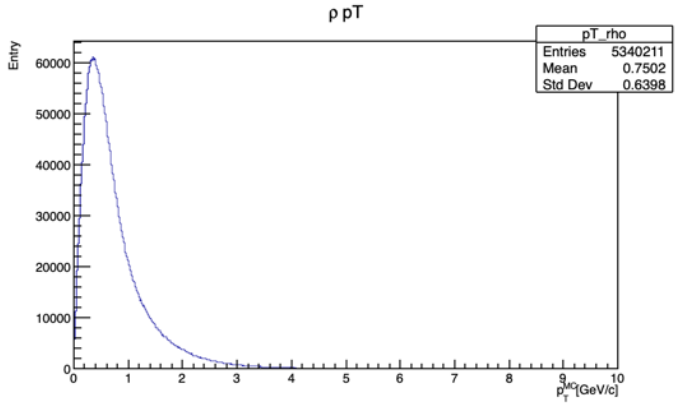


Figure 2.9: Transverse momentum distribution of charged particles in the MFT-MCH acceptances.(ρ)

The main tracks are of low transverse momentum, and the effect of multiple Coulomb scattering is not small, so that even if the combination of tracks is correct, there will be a finite difference in position ($\Delta X, \Delta Y$) on the matching plane. If there happens to be another track with the same or smaller $\Delta X, \Delta Y$, track matching becomes very difficult. On the other hand, a combination of wrong tracks at a considerable distance, for example, can be clearly shown to be wrong. Thus, even after removing the imbalance in the data, it is often more difficult to predict a positive sample to be a positive sample and easier to predict a negative sample to be a negative sample. Therefore, even if the loss is normalized by the number of samples in each class, the prediction for a positive sample has almost always larger loss than the loss of negative samples in the first place. Therefore, machine learning can suppress the loss for a positive sample by predicting that it is a positive example in order to reduce the loss, and the loss for a negative sample will increase, but the loss will be less if one negative sample is wrong than if one positive sample is wrong due to the normalization factor. Therefore, the tendency of Recall>Precision was observed, as more and more cases were predicted to be positive samples for negative samples.
In track matching, however, whether Recall or Precision should be higher depends on what you want to do after track matching, although both should be higher. In this case, it is better to succeed in reconstructing the invariant mass with high efficiency because the invariant mass can be reconstructed correctly when the tracks of two μ that decayed from the same parent particle are both reconstructed correctly when reconstructing the invariant mass. Therefore, a model with a higher Recall is better from the viewpoint of di-muon rather than single-muon. However, it should be noted that Recall and Precision are not entirely synonymous with efficiency and purity of track matching.
Finally, a further loss function devise is a countermeasure for unbalanced data. Even when it is easy to get a correct answer (= mostly negative samples), the predicted probability al-

most never takes a value such as a perfectly stated 0 or 1, so the log argument is often not 1, but close to 1. Thus, the loss for easy predictions is small, but when the loss is non-negligible, machine learning tries to reduce the loss by increasing the prediction performance for negative samples, which are easy to predict. In other words, when there is not only a difference in the number of positive samples and negative samples, but also a difference in the degree of difficulty, it may not be sufficient to normalize the loss function based on the number alone. FocalLoss is defined as follows[13].

$$\text{FocalLoss} = -\frac{N^-}{N^+}(1-y_{\text{pred}})^\gamma \log y_{\text{pred}} - \frac{N^+}{N^-} y_{\text{pred}}^\gamma \log(1-y_{\text{pred}}) \tag{2.15}$$

One major difference from the weighted logloss is that $(1-y_{\text{pred}})^\gamma$ is multiplied as a coefficient. The γ is a free parameter and must be optimized[13]. For large errors, this coefficient approaches 1, so the magnitude of the loss is close to the normal weighted logloss, but for small errors, this coefficient approaches 0, so the loss is negligible[13]. This coefficient weights the loss according to the difficulty of classification. This makes it possible to learn to give more weight to the minority of positive samples that are difficult to classify.
However, ALICE uses C++ to speed up data processing such as track reconstruction [16], while such machine learning is done using Python. Therefore, the trained models created on Python are used in C++. As a converter between different languages for this purpose, ALICE uses ONNXRuntime which is developed by MicroSoft and also used in ATLAS experiment, etc; in ONNXRuntime, only logloss is available as a loss function for the classifier, and no special loss functions such as Focal loss are available(31st Jan. 2023).
Because of the above and because we want to improve the performance of dimuon reconstruction over single muon, we will use weighted learning with logloss, which had a large Recall.
However, as a future perspective, it is possible to perform track matching without using ONNXRuntime using models trained with Focal loss in offline processing, and we would like to study the use of such a special loss function later.

2.4 TRACK MATCHING BY RANKING USING GBDT

Above, we considered using machine learning to classify whether the combination of tracks reconstructed by MFT and MCH was a combination of correct tracks made by the same particle or other wrong tracks. However, we thought it was not optimal to view track matching as a classification task.
In ALICE's data collection and analysis framework, track matching is a process in which a certain track reconstructed by MCH is combined with all candidate tracks reconstructed by MFT, and the most likely (=highest predicted probability) track combination is selected and stored as a μ track for analysis. Then, it does not matter whether the positive samples are correctly classified as positive samples (i.e., the predicted probability for positive samples exceeds the threshold value) or whether there are no negative samples mixed in among those classified as positive samples (i.e., the predicted probability for negative

samples does not exceed the threshold value). And also, it does not matter whether the prediction probability for a positive sample is below or above the threshold as long as the prediction probability for each positive sample is the largest among all the candidates.
To explain what this means, consider the search for the correct combination of tracks among the tracks reconstructed by MFT for a certain track reconstructed by MCH. In this case, the only candidate tracks are those in the same collision event, and there is only one (if any) correct combination of tracks for one track reconstructed by MCH. Track matching succeeds as long as machine learning gives a higher prediction probability for the correct MFT-reconstructed tracks than for any other candidate in the same event. Therefore, the predicted probability for a positive sample doesn't need to be well separated from the predicted probability for a negative sample at a threshold value across multiple events.
In the classification task, machine learning looks at a large amount of data (multiple tracks reconstructed by MCH in multiple events) at one time, characterizes the positive and negative samples by the differences in the distribution of various parameters, and finds the boundaries between positive and negative samples. The boundary line is then applied to the unknown data to discriminate between positive and negative examples. In this process, even though positive samples are correctly classified as positive and most negative samples are classified as negative (= high classification performance), only one negative sample crosses the boundary and is mistakenly classified as a positive sample, and its prediction probability exceeds the prediction probability of the combination of tracks that are truly positive samples. In most cases, this negative sample was adopted as a plausible combination of tracks, and track matching failed. To improve classification performance, the classification task handles multiple sets of data at the same time to capture features. As long as the boundaries are sought so that they can be classified as well as possible all at once, there will inevitably be data that must be given up on to be classified correctly. This was a major problem when treating track matching as a classification task.
Therefore, stop treating track matching as a classification task and treat it as a new ranking task. In the ranking, instead of looking at data from multiple events at once to get a sense of the characteristics, only tracks reconstructed by MFT that are candidates for a certain track reconstructed by MCH (= in the same event) are considered and ranked against those candidates. Then, when each MCH-reconstructed track and its corresponding rank table pair are created, we can see at once the track-rank table pairs in multiple events and the rank of the correct combination of tracks (=truth information) to get the characteristics of what tracks are plausible.
However, there is a caveat in this ranking. If the correct combination of tracks only needs to be ranked first among the candidates, it is enough to predict that all candidates are correct. This is the same as when we try to maximize only the recall in classification. This is not the model we are looking for. However, since there are multiple combinations of wrong tracks and no likelihood ranking is defined for each, the correctness of the ranking of negative samples cannot be considered either. Therefore, it is important that in track matching, only the correct tracks combination should be ranked first, all other negative samples should be ranked second or lower, and the order below second should not be evaluated.

Therefore, we will now discuss the loss function for ranking.
First, let s_i be the predicted probability of being a positive sample, obtained by inputting the features of the i-th dataset into the model f.

$$s_i = f(x_i) \tag{2.16}$$

Next, the difference between s_i and s_j is converted to a value between 0 and 1 using a sigmoid function.

$$P_{ij} = \frac{1}{1 + e^{-\sigma(s_i - s_j)}} \tag{2.17}$$

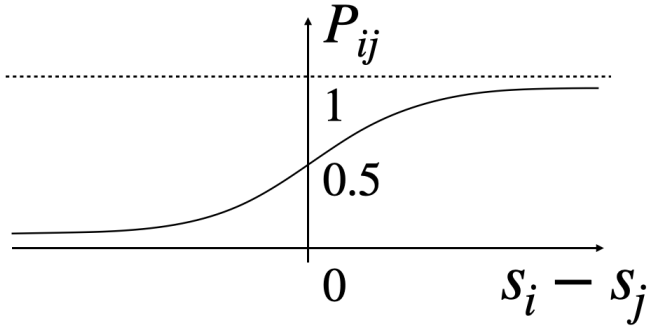


Figure 2.10: Sigmoid function

If the P_{ij} values are close to 0, the jth is more positive than the ith, and if they are close to 1, the ith is more positive than the jth. If the two data sets are ordered correctly, then all of them should also be ordered correctly. This concept is called "pair-wise". In other words, the task of ranking can be treated as a binary classification problem. Then, consider the following loss function C.

$$C = \begin{cases} \log P_{ij} \ (y_i = 1, y_j = 0) \\ \log(1 - P_{ij}) \ (y_i = 0, y_j = 1) \\ 0 \ (y_i = 0, y_j = 0) \end{cases} \tag{2.18}$$

This loss function evaluates only the difference in predicted probability between the positive sample ($y = 1$) and the other negative examples ($y = 0$), not the greater or lesser relationship between the predicted probabilities of the negative examples. Therefore, when learning so that this loss function is minimized, the relationship between the predicted probabilities of the positive sample and the other negative examples will be corrected, and learning will not proceed for the negative examples. Then, learning proceeds so that the predicted probability of the positive sample is higher than the predicted probability of the other negative samples for every negative sample, resulting in the positive sample being

the only one that ranks first.
Also, this loss function C is

$$\frac{\partial C}{\partial s_i} = \frac{\partial C}{\partial P_{ij}} \frac{Pij}{s_i} = \begin{cases} \frac{1}{P_{ij}} \frac{\partial P_{ij}}{\partial s_i} \ (y_i = 1, y_j = 0) \\ \frac{1}{1-P_{ij}} \frac{\partial (1-P_{ij})}{\partial s_i} \ (y_i = 0, y_j = 1) \\ 0 \ (y_i = 0, y_j = 0) \end{cases} \tag{2.19}$$

$$= \begin{cases} \frac{\sigma e^{-\sigma(s_i - s_j)}}{1+e^{-\sigma(s_i - s_j)}} = -\frac{\partial C}{\partial s_j} \ (y_i = 1, y_j = 0) \\ \frac{\sigma}{\left(1+e^{-\sigma(s_i - s_j)}\right)^2} = -\frac{\partial C}{\partial s_j} \ (y_i = 0, y_j = 1) \\ 0 \ (y_i = 0, y_j = 0) \end{cases} \tag{2.20}$$

differentiable with predicted probability s_i and can be differentiated by the second-order derivative in the same way, allowing for learning using the gradient descent method.
In this study, such ranking using the loss function C was performed through machine learning.

3

REQUESTS FROM PHYSICS FOR TRACK MATCHING

Here, we discuss the requirements for track matching. We aim to provide experimental verification of the relationship between hadron mass and spontaneous breaking of the chiral symmetry using μ tracks reconstructed by matching together MFT and MCH tracks.

3.1 TIME EVOLUTION OF QGP IN THE TRANSVERSE PLANE

In the environment in which we live today, quarks and gluons always form hadrons, and particles that exist alone always have a color charge of 0. Therefore, quarks and gluons do not exist by themselves. However, in the case of ultrahigh temperatures, such as the universe immediately after the Big Bang, or ultrahigh densities, such as the interiors of neutron stars, quarks and gluons are released from their hadron confinement by the phase transition, and particles with color charge can exist on their own. The phase diagram of QCD is shown below.

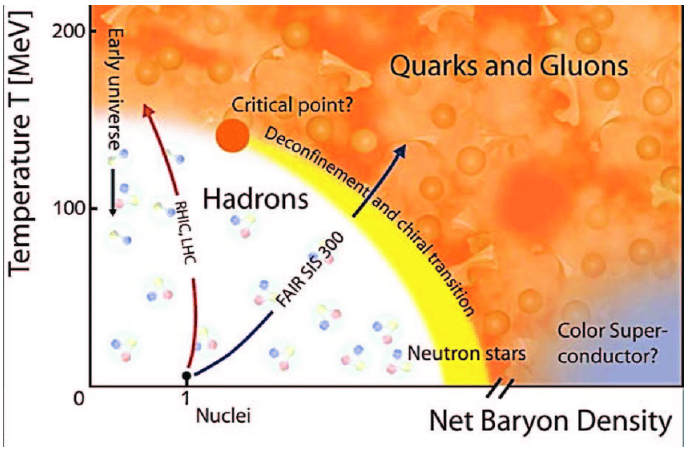


Figure 3.1: QCD phase diagram[17]

Similarly, spontaneous chiral symmetry breaking is predicted to be partially restored at very high temperatures and densities[18]. As mentioned above, we can think qualitatively this is because intense thermal motion causes the quark-antiquark pairs condensed in the vacuum to unravel, lowering their vacuum expectation value, and that the higher density of hadrons reduces the space available for quark-antiquark pairs in that space, lowering their vacuum expectation value.

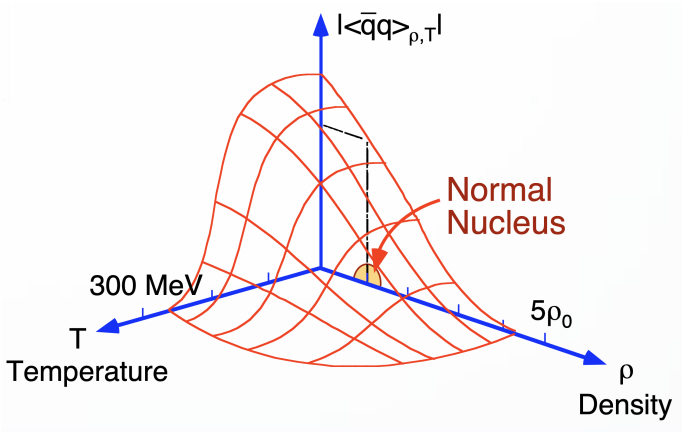


Figure 3.2: Temperature and density dependence of $\langle \bar{q}q \rangle$ [19]

In ALICE, the phase transition temperature of 170 MeV can be exceeded by instantaneous ultrahigh temperatures in high-energy heavy-ion collision experiments, and the QGP phase can be realized. In the QGP phase, spontaneous chiral symmetry breaking should be partially recovered. In the following, we will consider a simple model of how the space of the QGP phase generated by high-energy heavy-ion collisions evolves.
The time evolution of heavy-ion collisions generally occurs first with hard scattering, and thermal equilibration occurs at times less than 1 fm/c. Here soft scattering occurs, and hadronization gradually begins. This is followed by the end of inelastic scattering (chemical freeze out) and, finally, the end of elastic scattering (kinematic freeze out). There are various models that treat the thermally equilibrated QGP as a fluid. Still, for the sake of simplicity, we will consider a straightforward model in which the QGP phase space cools with expanding at a constant rate immediately after the collision in the Transverse plane.
Let $t = 0$ be the time when the heavy ion (lead in this case) beam collides with the system, and let the initial size of the system be a circle with radius r_{Pb} equal to the radius of the lead nucleus. The initial radius of the system r_{Pb} can be written using the atomic number A of lead as follows.

$$r_{\mathrm{Pb}} = r_0 A^{1/3} \tag{3.1}$$

where r_0 is a constant of approximately 1.3. Then consider that this circle expands at a constant speed of $0.66c$. Here c represents the speed of light.
Next, consider the temperature of the system.

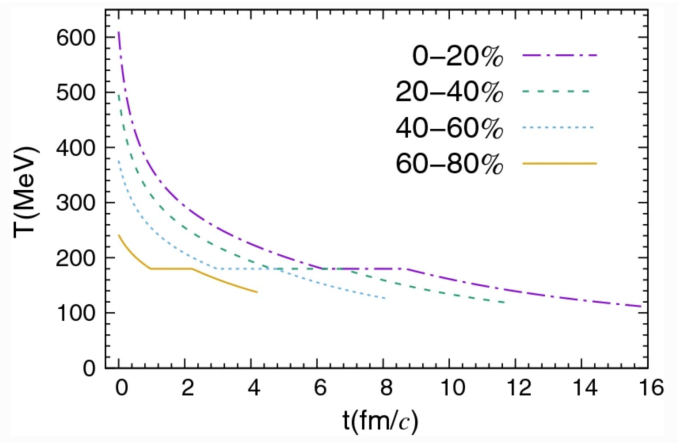


Figure 3.3: Time evolution of system temperature[20].

This figure shows the time evolution of the temperature of the medium produced by a Pb-Pb collision with $\sqrt{s_{NN}} = 5$TeV, based on the thermal fireball model[20]. The temperature of the simple model considered here is assumed to have a temperature at a central degree of $0 \sim 20\%$ at each time point. According to this figure, the temperature remains constant for approximately $6 \sim 8$fm/c, which is the thermal equilibrium time. We assume that hadronization starts exactly in the middle of this period, at $t = 7$[fm/c]. Then at $t = 15$[fm/c], the temperature cools down to about 120MeV. This is about 70% of the QCD phase transition temperature $T_c = 170$[MeV]. The expected value of the vacuum varies continuously with temperature, but for simplicity, we assume that spontaneous chiral symmetry breaking is recovered at $T > 0.7T_c$ and that the chiral symmetry is spontaneously broken at $T < 0.7T_c$.

Next, we consider the events in which ρ, ω, ϕ are randomly created inside the system and decay to dimuon. At the same time, the spontaneous breaking of chiral symmetry is recovered in this model ($7 < t < 15$[fm/c]). If these hadrons decay without leaving the medium between their creation and decay, then detecting the dimuon and reconstructing the invariant mass will result in an invariant mass that contains information about the mass change due to the recovery of spontaneous breaking of the chiral symmetry. Suppose these hadrons decay after they exit out of the medium. In that case, the invariant masses of the dimuons are not likely to change due to the spontaneous breaking of the chiral symmetry since hadrons with masses in the hadronic phase will decay.

The transverse momentum distribution of ρ, ω, ϕ within the MFT-MCH acceptances are obtained from the simulation using Pythia8 HI $\sqrt{s_{NN}} = 5.5$TeV, minimum bias Pb-Pb collision. Each distribution are parameterized and shown.

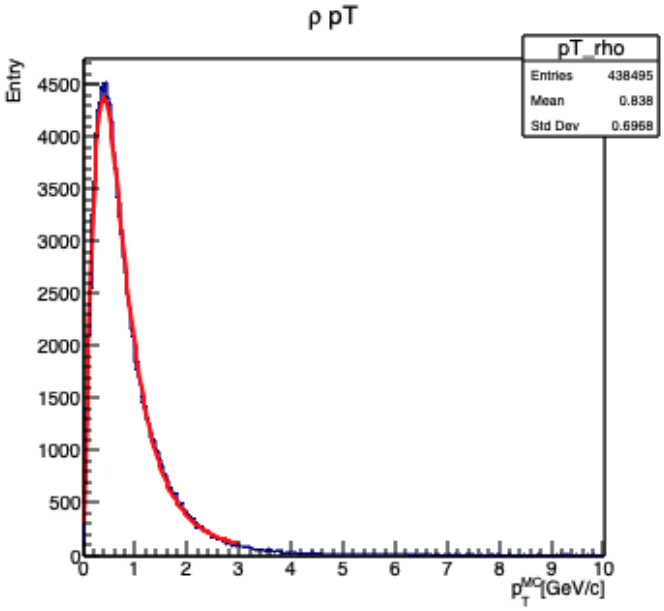


Figure 3.4: ρ meson's transverse momentum distribution.

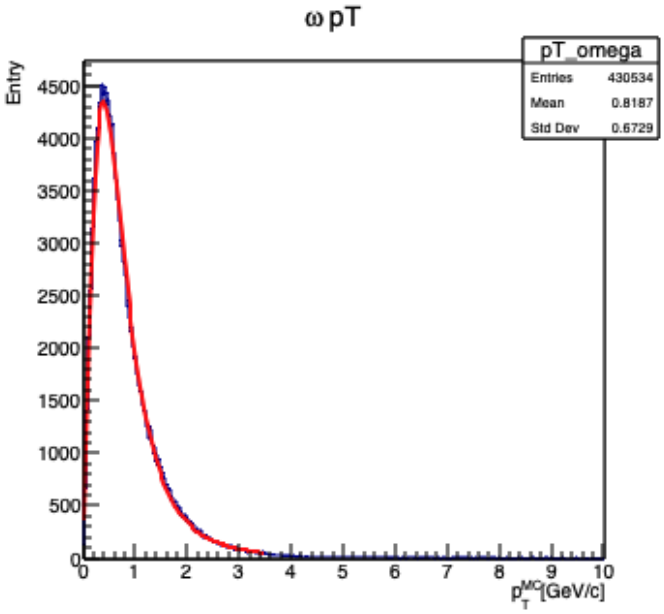


Figure 3.5: ω meson's transverse momentum distribution.

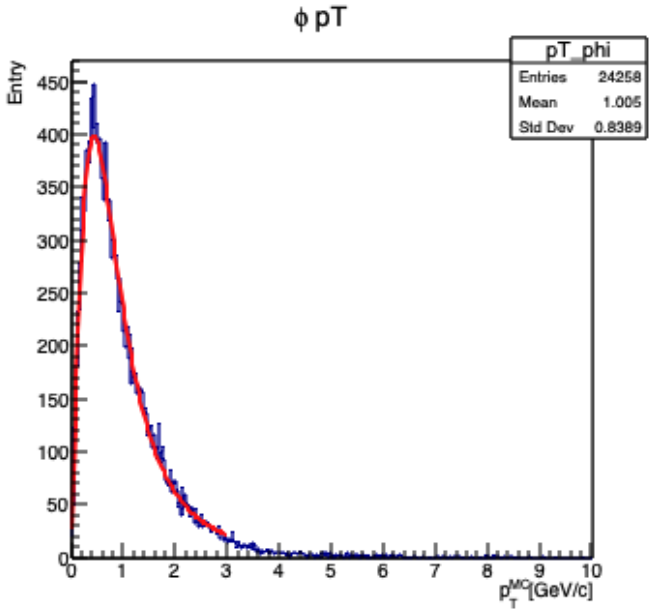


Figure 3.6: ϕ meson's transverse momentum distribution.

These distributions were parameterized by the following equation.

$$\frac{dN_\rho}{dp_T} = p_0 \frac{p_T}{\left[p_1 + (\frac{p_T}{p_2})^{p_3}\right]^{p_4}} \tag{3.2}$$

The parameters of each distribution are as follows

Table 3.1: Paramters of each hadron's transverse momentum distributions

Parameter	ρ	ω	ϕ
p_0	1.7×10^2	5.23×10^1	4.82×10^1
p_1	4.43×10^{-1}	3.76×10^{-1}	6.20×10^{-1}
p_2	3.60×10^0	3.86×10^0	5.04×10^{-1}
p_3	8.75×10^{-1}	9.05×10^{-1}	7.39×10^{-1}
p_4	1.02×10^1	9.70×10^0	1.41×10^1

We also used $\Gamma_\rho = 149\text{MeV}, \Gamma_\omega = 8.68\text{MeV}, \Gamma_\phi = 4.249\text{MeV}$ for the decay width, respectively. The decay probabilities were made to follow an exponential function in the rest frame of each hadron, the coordinates of the production point were uniformly random inside the medium, and the production cross section was constant from hadronization to spontaneous chiral symmetry breaking.
The resulting Lorentz β of ρ, ω, ϕ that decays inside the medium is shown in the following figures.

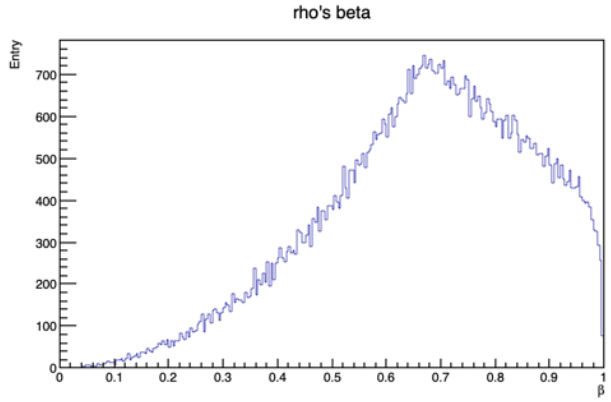


Figure 3.7: ρ meson's lorentz beta distribution.

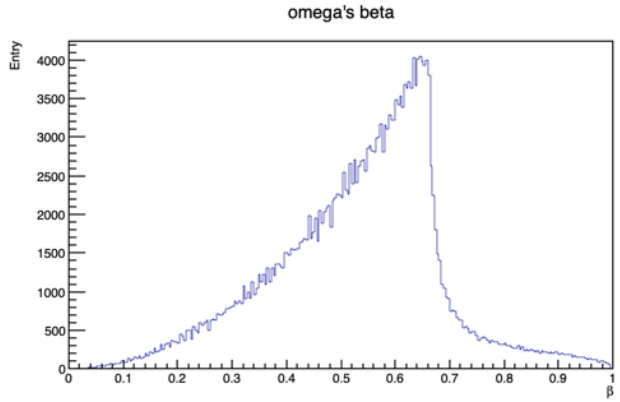


Figure 3.8: ω meson's lorentz beta distribution.

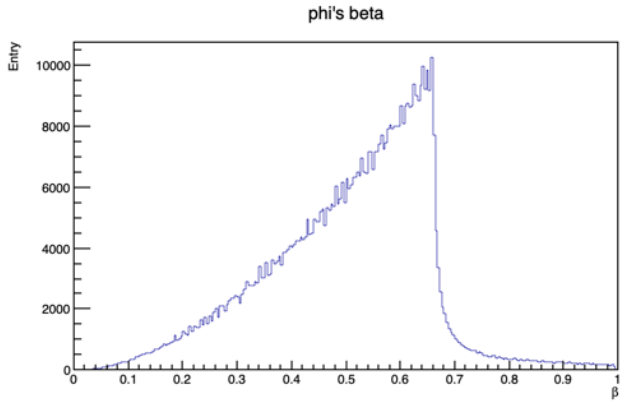


Figure 3.9: ϕ meson's lorentz beta distribution.

The Lorentz β of hadrons decaying inside the medium basically decreases rapidly after the expansion rate of the medium for all three. However, only ρ's lifetime is shorter than others, so it can decay inside the medium even if it has a Lorentz β that is larger than the expansion rate of the medium. The reason for the low Lorentz β distribution in the region

below the medium expansion velocity may be due to the shape of the hadron transverse momentum distribution.
Next, the percentage of decay inside the medium was determined for each transverse momentum.

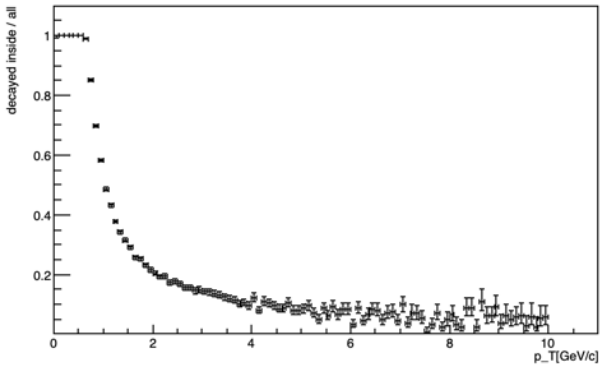


Figure 3.10: Percentage of ρ mesons that decay inside medium

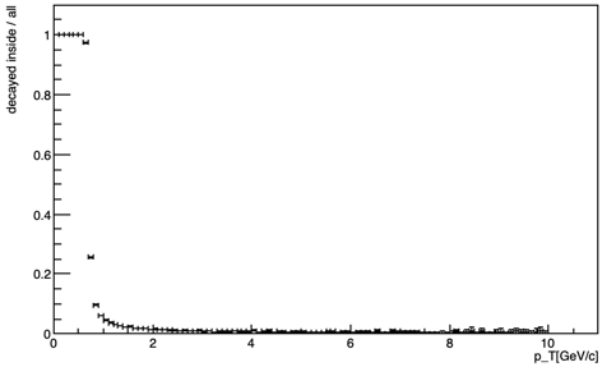


Figure 3.11: Percentage of ω mesons that decay inside medium

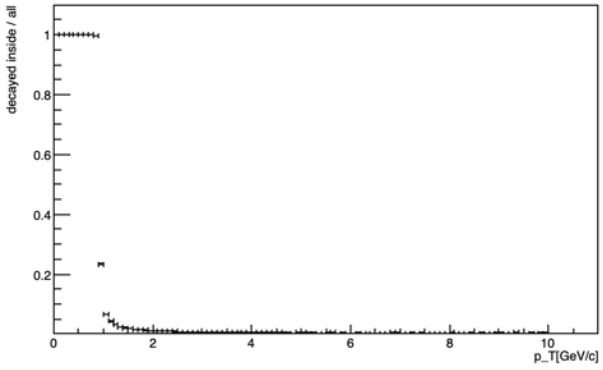


Figure 3.12: Percentage of ϕ mesons that decay inside medium

Thus, all three mesons can certainly decay inside the medium if their transverse momentum is less than 1 GeV/c. Still, if it exceeds 1 GeV/c, the fraction of mesons that decay inside the medium decreases drastically. However, the ρ's lifetime is significantly shorter than the other two, so it can decay inside the medium if its transverse momentum is about 20% at 2 GeV/c compared to the other two.
In other words, the lower the transverse momentum, the more ρ, ω, ϕ can decay inside the medium, so the mass tends to change due to partial recovery of spontaneous chiral symmetry breaking and considering that about half of the transverse momentum is inherited by each μ when ρ, ω, ϕ decays to dimuon, the performance of track matching at $p_{T,\mu} < 1$GeV/c is required to observe as much mass change as possible.

4

PERFORMANCE COMPARISON OF TRACK MATCHING METHODS

4.1 EFFICIENCY OF TRACK MATCHING AND PURITY OF TRACKS

First, as a performance comparison of track matching itself, we define Purity and Efficiency as follows

$$\text{Purity} = \frac{N_{\text{matched correctly}}}{N_{\text{matched}}} \tag{4.1}$$

$$\text{Efficiency} = \frac{N_{\text{matched correctly}}}{N_{\text{matchable correctly}}} \tag{4.2}$$

where $N_{\text{matched correctly}}$ is the number of matches that were actually correctly matched, N_{matched} is the number of matches that were actually matched (regardless of whether they were correct or not), and $N_{\text{matchable correctly}}$ is the number that was originally the number of correct matches (regardless of whether the matching was actually correct or not). In the case of a cut that excludes some tracks, even if the cut makes it impossible to match some tracks that were originally matched correctly, this is included in $N_{\text{matchable correctly}}$.
In the following, we also performed track matching on 500 events with $\sqrt{s_{NN}} = 5.5\text{TeV}$ and minimum bias Pb-Pb collisions, the same as the training data and evaluated Purity and Efficiency.

4.1.1 *Confirmation of Kalman filter performance*

The Kalman filter is a matching method that has been mentioned as a candidate within ALICE for some time. Its performance is shown below.

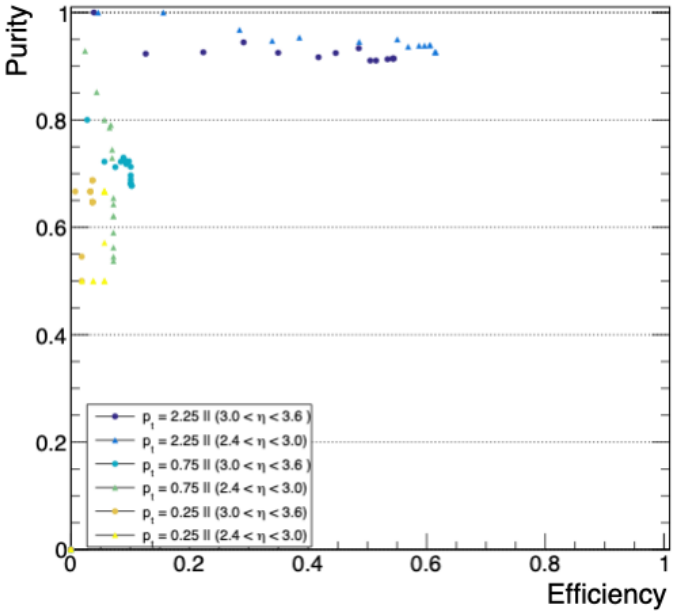


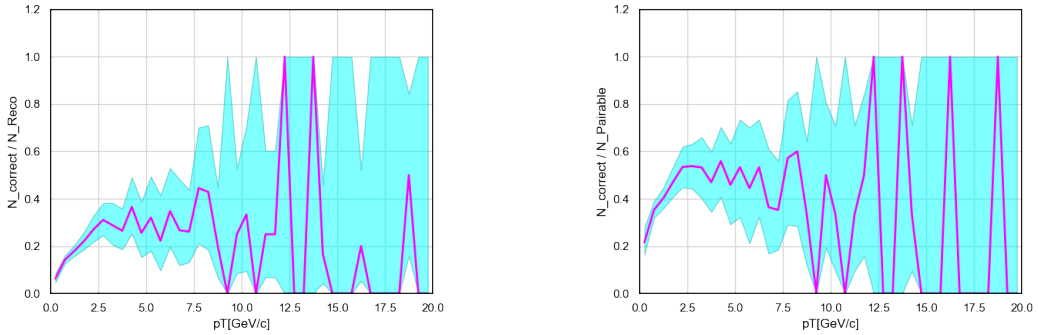
Figure 4.1: The result of track matching by Kalman filter.

The vertical axis represents Purity, and the horizontal axis represents Efficiency. When using the Kalman filter, the smaller the residuals assigned to each track reconstructed by MFT, the more plausible the track is. The figure above shows the values of Purity and Efficiency when an arbitrary threshold is set for the residuals and only tracks below the threshold are used, and the threshold values are varied. Kalman filter track matching tends to have high Purity, but at the same time, Efficiency is very low. This leaves almost all muon tracks in the low transverse momentum region unreconstructed, with only those that are easily matchable being matched. Therefore, in the low transverse momentum region, there are very few muon tracks, which is very unfavorable given the low branching ratio of the $\rho, \omega, \phi \to \mu^+\mu^-$ decay modes.

4.1.2 *Performance of track matching with machine learning based classifiers*

In the following, we will confirm the performance of the machine learning based track matching method developed in this study. Similar to the figure used to check the performance of the Kalman filter, the probability that the combination of tracks reconstructed by MFT and MCH is the correct combination is expressed as a number between 0 and 1, an arbitrary threshold is set for the probability, and only tracks with a combination higher than the threshold are used for track matching and the Purity and Efficiency values are shown for different threshold values. Also shown are the transverse momentum dependence of Purity and Efficiency when no threshold is set. The transverse momentum dependence can be seen in each figure, but as discussed in the previous section, performance is required in the low transverse momentum region around 1 GeV.In the figure for the dependence of Purity and Efficiency on transverse momentum, the blue band represents the statistical error.

The machine learning based method used weighted learning to improve the accuracy of learning, but to further improve the accuracy and efficiency of learning, try cuts on the candidate tracks to be matched. In the following, we will check the performance of the method in three cases: when no cut is used, when a cut is applied based on the distance from the track reconstructed by MCH to the candidate, and when, in addition to the cut, the difference in acceptances of the two detectors is cut.
First, we show the performance of track matching by a machine learning based classifier when no cuts are used that exclude tracks.



(a) Purity (b) Efficiency

Figure 4.2: Transverse momentum dependence of Purity and Efficiency for track matching with a machine-learned classifier in the absence of cuts. The band represents the statistical error.

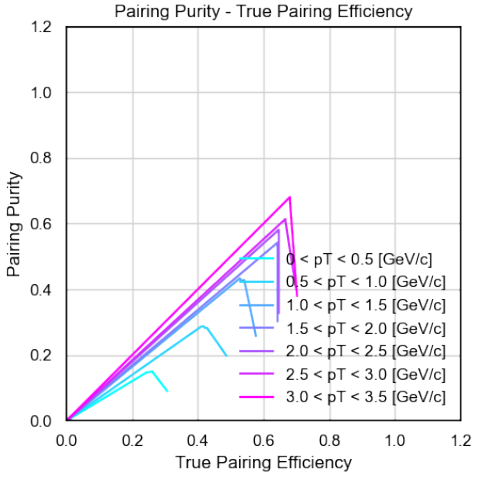
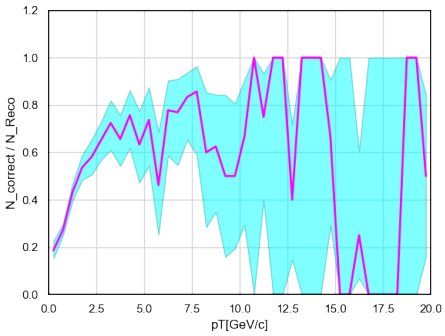
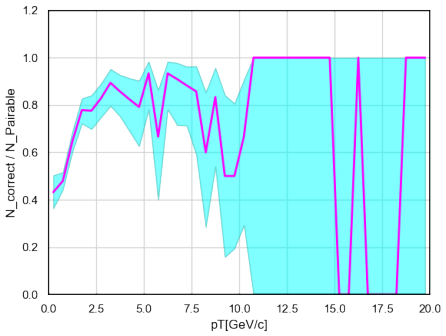


Figure 4.3: Purity and Efficiency of track matching with machine learning based classifiers when no cuts are used.

Next, use cuts that exclude some tracks to improve track matching performance. Cuts should be used consistently from training to testing.
First, we trained only for those falling within $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\mathrm{cm}]$ and used this cut also during testing. In this case, we show the performance of track matching by the classifier using machine learning.



(a) Purity



(b) Efficiency

Figure 4.4: Transverse momentum dependence of Purity and Efficiency for track matching with a machine-learned classifier using cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$. The band represents the statistical error.

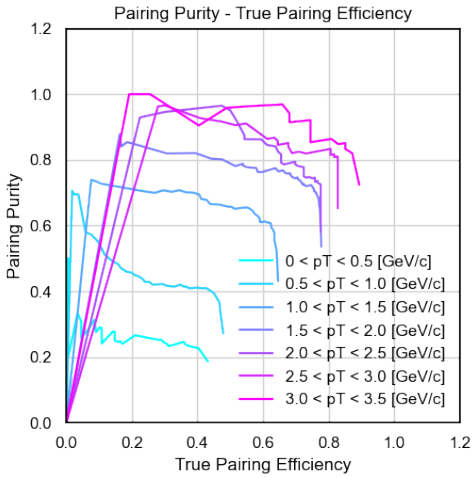
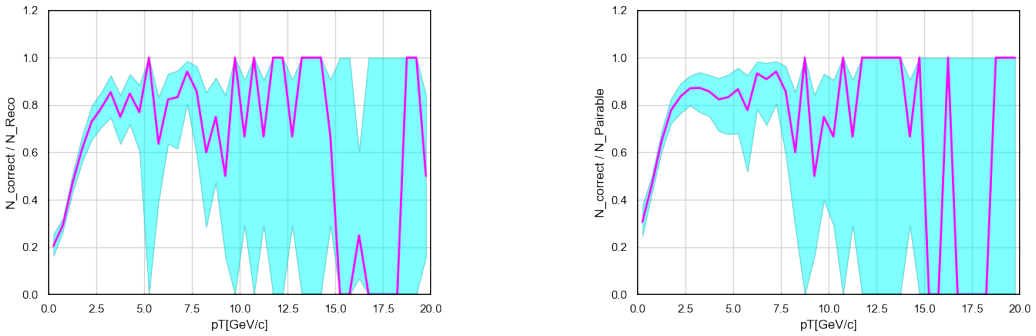


Figure 4.5: Purity and Efficiency of track matching with machine learning based classifiers using cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$.

We also performed training to testing with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ as a stronger cut. In this case, the performance of track matching with a machine learning based classifier is shown.



(a) Purity (b) Efficiency

Figure 4.6: Transverse momentum dependence of Purity and Efficiency for track matching with a machine-learned classifier when $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ cuts. The band represents the statistical error.

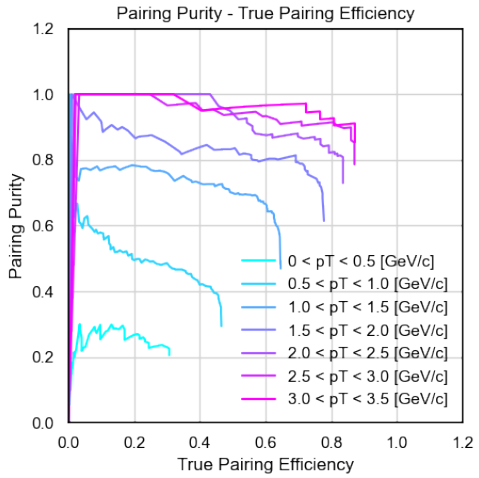


Figure 4.7: Purity and Efficiency of track matching with machine learning based classifiers using cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$.

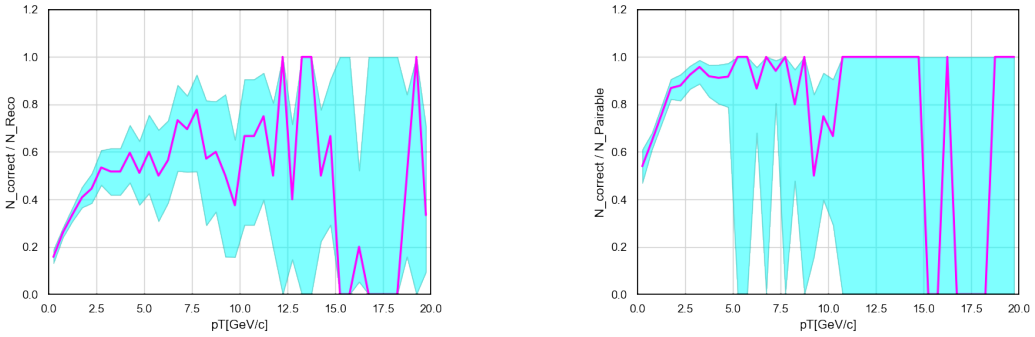
MFT and MCH have slightly different acceptance regions, with the $-3.75 < \eta < -3.6$ region being undetectable with MFT but detectable with MCH. Therefore, particles passing through this region often hit only in MCH, so we cut tracks in this region.

We have checked the performance of the same machine learning based classifier with different cuts, and compared to no cut, applying a cut of $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ improved both Purity and Efficiency. However, when we cut the difference in acceptances as an even stricter cut, Purity improved, but Efficiency decreased slightly. This shows that the cut applied at the distance between the track reconstructed by MCH and the candidate track reconstructed by MFT was moderately able to eliminate the wrong candidate, but the cut in the eta region, which is the difference in acceptances, was too severe.

The consistently lower Purity and Efficiency in the low transverse momentum region may be because the effect of multiple Coulomb scattering is more significant at low transverse momentum.

4.1.3 *Performance of track matching by ranking using machine learning*

The performance evaluation of track matching by ranking using machine learning is also performed in the same way as before, with three types of cuts: no cut, $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$, $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ cuts.
First, without cuts, the results were as follows.



(a) Purity (b) Efficiency

Figure 4.8: Transverse momentum dependence of Purity and Efficiency for track matching by ranking using machine learning without cuts. The band represents the statistical error.

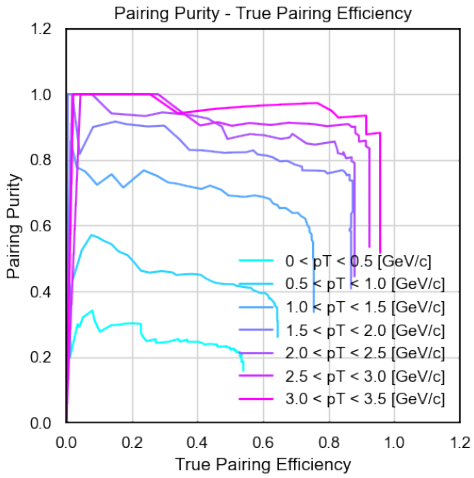
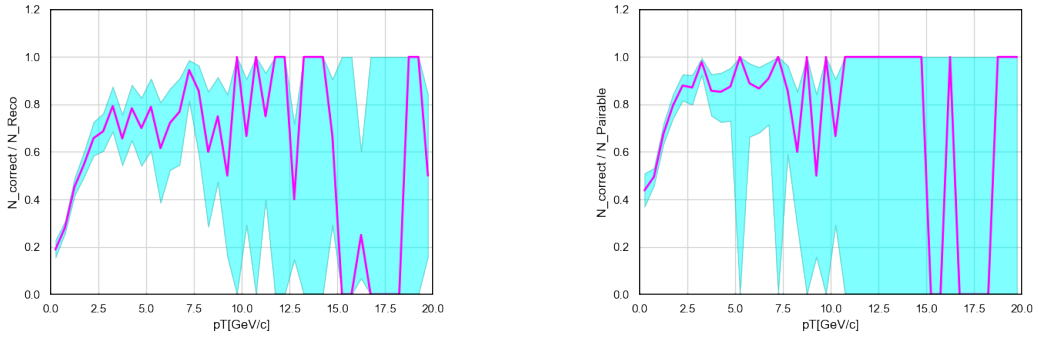


Figure 4.9: Purity and Efficiency of track matching by ranking using machine learning when no cuts are used.

When machine learning is used to rank, it can achieve higher Efficiency than any of the classifiers with cuts, even though it does not use cuts. In classification using machine learning, the classifier only learns to increase the Purity and Efficiency of the classified positive example class, so even if the Purity and Efficiency are high, the predicted probability for a positive example when track matching is performed is only the predicted probability of being in the positive example class, never been always. Hence, the highest predicted probability among the candidates. This proves that the performance was lower than the ranking system's because the probability was not necessarily the highest.

Now, since it would be good if we could increase Purity while maintaining this high Efficiency, we first tried cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$. We did training and testing only on tracks that were within the $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ region. The results are shown below.



(a) Purity (b) Efficiency

Figure 4.10: Transverse momentum dependence of Purity and Efficiency for track matching by ranking using machine learning and cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$. The band represents the statistical error.

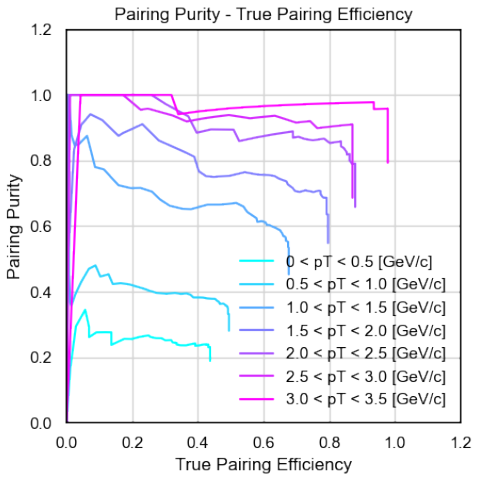
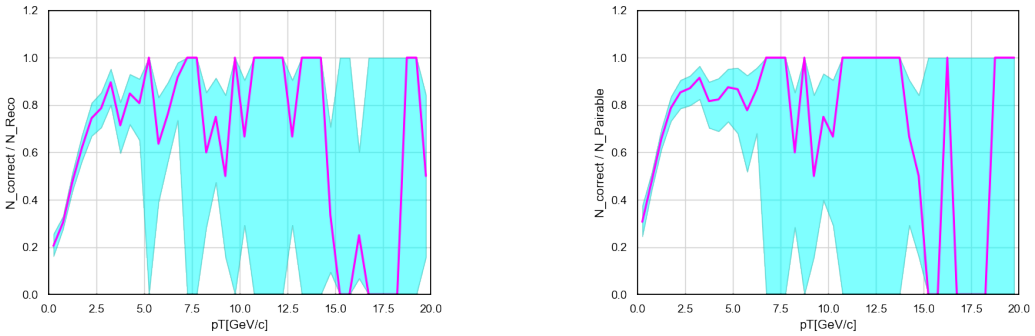


Figure 4.11: Purity and Efficiency of track matching by ranking using machine learning and cuts with $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$.

The use of cuts did not work as expected, resulting in lower Efficiency. And in the transverse momentum region higher than 1 GeV/c, Purity is increased, but in the transverse momentum region lower than 1 GeV/c, Purity is also slightly decreased. Finally, the case of $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ is shown.



(a) Purity (b) Efficiency

Figure 4.12: Transverse momentum dependence of Purity and Efficiency for track matching by ranking using machine learning when $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ cuts. The band represents the statistical error.

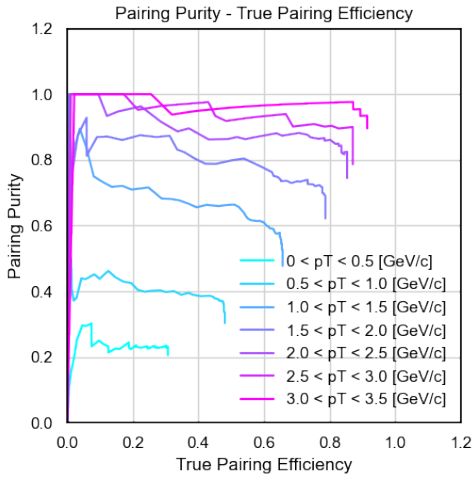


Figure 4.13: Purity and Efficiency of track matching by ranking using machine learning when $\sqrt{\Delta X^2 + \Delta Y^2} < 15[\text{cm}]$ and $\eta > -3.6$ cuts.

In this case, Efficiency is further reduced. And Purity is not greatly improved either. While ranking using machine learning succeeded in correctly matching a more significant number of positive samples, simple cuts did not increase Purity because they did not reduce only the wrong ones in the ranking.

Based on previous results, the maximum Efficiency for muons with transverse momentum around 1 GeV is about 40% for track matching based on machine learning classification and about 60% for track matching based on ranking. On the other hand, the Kalman filter has an efficiency of only about 10%. This means that muons with transverse momentum around 1 GeV are not correctly matched with a probability of about 90% when we using the Kalman filter. In contrast, track matching by classification could be reduced to 60%, and track matching by ranking could be reduced to 40%. Furthermore, the probability of incorrectly matching both muon pairs was reduced to 36% with classification and 16% with ranking, compared to about 81% with the Kalman filter.

4.2 EFFECT OF MATCHING METHOD ON INVARIANT MASS DISTRIBUTION OF DIMUON

So far, we have discussed the quality of single-muon reconstruction using Purity and Efficiency. However, even when the wrong tracks are matched, Purity and Efficiency cannot be used to evaluate whether it is a terrible or regrettable mistake. In addition, we would like to discuss the quality of dimuon reconstruction rather than single-muon reconstruction since dimuon must be measured to actually verify the relationship between spontaneous chiral symmetry breaking and hadron mass in experiments. Therefore, we reconstruct the invariant mass of the dimuon with each track matching method and obtain the mass distribution of hadrons. Then, we will discuss differences in the ratio of the signal component to the background event component and the width of the peak of the signal component of that hadron.

Here we use $\sqrt{s_{NN}} = 5.5$TeV, minimum bias Pb-Pb collision events simulated by Pythia8 HI with 10 ρ, a_1, ω, ϕ mesons each embedded. Note that Figures 3.4, 3.5, and 3.6 were used for the ρ, ω, ϕ transverse momentum distributions, and the a_1 meson transverse momentum distribution was m_T scaling from ρ transverse momentum distribution. The pseudo-rapidity distributions for each of ρ, ω, ϕ as well as the transverse momentum distributions, were obtained by simulating a $\sqrt{s_{NN}} = 5.5$TeV, minimum-bias Pb-Pb collision event in Pythia8 HI. Also, the pseudo-rapidity distribution for a_1 is taken directly from ρ. The results of reconstructing the invariant masses with each track matching method are as follows.

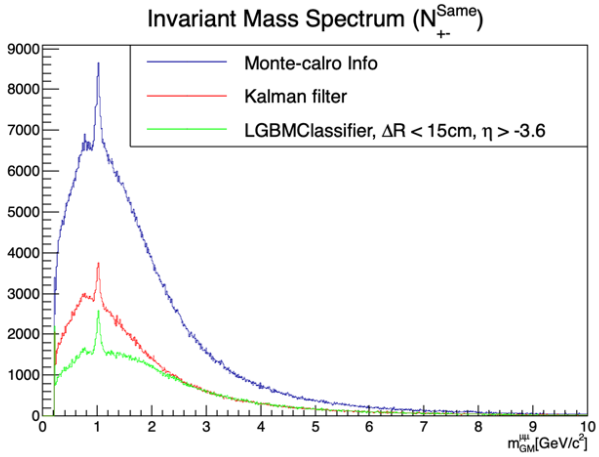


Figure 4.14: Spectra of invariant masses reconstructed from tracks reconstructed by three different track matching methods, reconstructing invariant masses between tracks of different charges in the same event. The blue line shows the spectrum of track matching using the truth information from a Monte Carlo simulation. The red line represents the spectrum when track matching is performed using the Kalman filter. The green line represents the spectrum when track matching was performed with machine learning based classification "LGBMClassifier" is the name of classifier used in this study. LGBMClassifier is developed by Microsoft. Note that to increase the performance of classification using machine learning, we use a trained model learned using cuts of $\sqrt{\Delta X^2 + \Delta Y^2} < 15$[cm].

The case of track matching using information from a Monte Carlo simulation represents the case where all track matching is done perfectly, so no better performance can be achieved using MFT, MCH, and MID. Track matching using Kalman filter or machine learning has a different spectral height than track matching using Monte Carlo simulation information. In the case of using Monte Carlo information, all the tracks are perfectly track-matched, and the invariant mass is reconstructed using all the tracks after matching. However, in the others, whether Kalman filter or machine learning, scores or predicted probabilities are calculated for all combinations of tracks, and the combinations with the highest scores or predicted probabilities are matched against each other. Therefore, for a given track reconstructed by MCH, all the candidate tracks reconstructed by MFT may have a low score or predicted probability. For example, in the case of a classifier that assumes a positive sample if the predicted probability is greater than 0.5 and a negative sample if it is less than 0.5 if all the candidates have a prediction probability of less than 0.5, it is wise not to perform matching in the first place. Therefore, a score or predicted probability threshold is set when using Kalman filter or machine learning to perform track matching. The spectrum is, therefore, lower than when using information from Monte Carlo simulations. Regarding the difference between using the Kalman filter and machine learning, suppose that when an algorithm makes a mistake in track matching, it mistakes a track for another track similar to the correct one. In this case, we consider that the parameters $x, y, \phi, \tan\lambda, q/p_T$ that the track has have changed as if the noise was added to the parameters when the track matching was initially done correctly. In this case, the invariant mass also changes as if it were noisy compared to the case where track matching was performed correctly. The accuracy of track matching depends on the momentum of the particles due to multiple scattering in the material layer located between the MFT and MCH. As seen in Efficiency and Purity, the smaller the momentum is, the larger the multiple scattering is, and the more difficult track matching is. Therefore, tracks with low transverse momentum are noisier than tracks with high transverse momentum in terms of the parameters that the tracks have. Given that the parent hadron decays two-body to dimuon, the invariant mass of the parent hadron is

$$M^2 = m_{\mu_1}^2 + m_{\mu_2}^2 + 2(E_{\mu_1}E_{\mu_2} - p_{\mu_1} \cdot p_{\mu_1}) \tag{4.3}$$

$$= m_{\mu_1}^2 + m_{\mu_2}^2 + 2\left(\sqrt{m_{\mu_1}^2 + p_{\mu_1}^2}\sqrt{m_{\mu_2}^2 + p_{\mu_2}^2} - |p_{\mu_1}||p_{\mu_2}|\cos\theta\right) \tag{4.4}$$

In the forward rapidity region where MFT and MCH are located, the range of θ is determined to some extent so that, roughly speaking, the parent hadron of a dimuon with little momentum has a light mass and the parent hadron of a dimuon with large momentum has a heavy mass. In other words, the mass distribution of the light parent hadron is noisier than that of the heavy parent hadron, resulting in a large difference between the different track matching methods in the light invariant mass region. In contrast, the mass distributions are identical in the heavy invariant mass region. Next, we show the results of subtracting the uncorrelated background using the Like-Sign method.

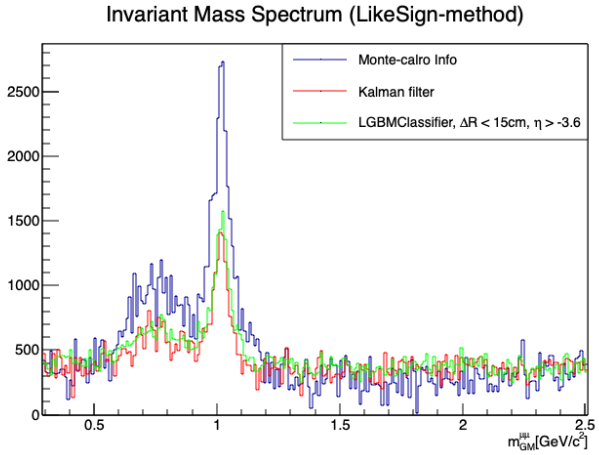


Figure 4.15: Invariant mass spectrum with uncorrelated background removed by the Like-Sign method.

This figure shows that the peak height is lower for the other track matching methods than the perfect track matching method. The peak near $1\mathrm{GeV}/c^2$ is the peak of ϕ mesons, but the height is different for each track matching method. The width, however, does not change significantly. We calculate the signal-to-noise ratio for ϕ meson peaks for performance comparison between the Kalman fiter based track matching method and track matching with machine learning based classification in the following.
First, we will perform background fitting for each track matching method.

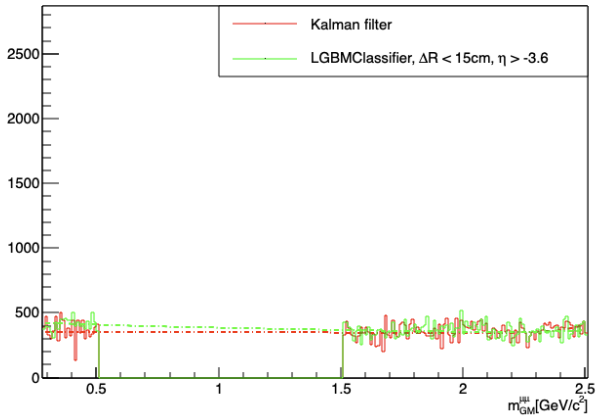


Figure 4.16: Fitting for background.

The background was then subtracted, and what remained was taken as the signal, and the peak of the omega meson in the region of $0.9\,\mathrm{GeV}/c^2$ to $1.1\,\mathrm{GeV}/c^2$ was integrated. This value is denoted as I_{Sgn}.

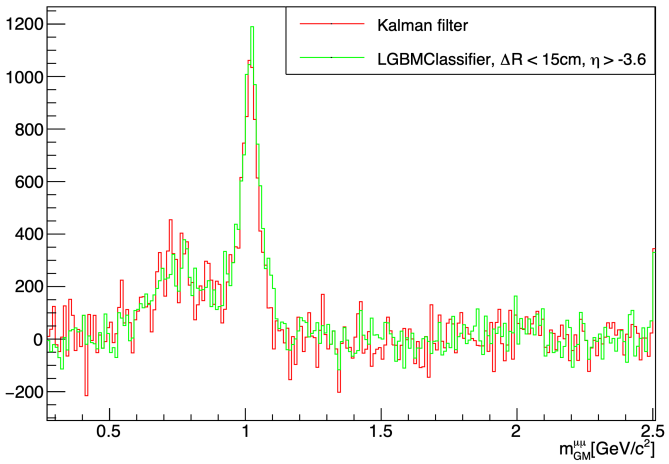


Figure 4.17: Background was subtracted.

If I_{All} is the integrated value of the spectrum in the same mass range before the background is subtracted, the background component I_{Bkg} can be expressed as $I_{All} - I_{Sgn}$. The signal-to-noise ratio is defined as follows.

$$S/N \text{ ratio} = \frac{I_{Sgn}}{I_{Bkg}} \tag{4.5}$$

Calculating this signal-to-noise ratio for each track matching method yielded the following values.

$$\text{Kalman filter} : 1.21 \pm 0.03 \tag{4.6}$$

$$\text{LGBMClassifier} : 1.29 \pm 0.03 \tag{4.7}$$

This confirms that track matching by machine learning based classification has a significantly higher signal-to-noise ratio than the Kalman fiter based method. Also, comparing the signal-to-noise ratio between the Kalman filter and the machine learning based classification, the machine learning based classification has a signal-to-noise ratio 1.06 times higher than that of the Kalman filter.
Originally, the branching ratio of decay modes of light vector mesons to dimuons was meager, and statistics were needed to analyze the mass spectra. Still, the use of machine learning in this study successfully improved the signal-to-noise ratio.

5

DISCUSSION

5.1 DISCUSSION

In the previous chapter, we first checked the accuracy of the single muon reconstruction using Purity and Efficiency.
As a result, qualitatively, the Kalman filter achieved high Purity but very low Efficiency. Track matching by machine learning-based classification achieved lower Purity than the Kalman filter, but also had sufficient Efficiency. Also, the cut for candidates improved Purity and Efficiency, but tightening the cut further increased Purity but lowered Efficiency. Machine learning-based ranking showed that the highest Efficiency was achieved without cuts, but Purity was low. When the cut was used, Purity was increased, but Efficiency was decreased.
We want to discuss these behaviors. First, in all track matching methods, as the threshold used for track matching is tightened, first, Efficiency remains constant and only Purity improves. Then, Efficiency begins to decrease from the middle of the process, and Purity increases slowly. This is because, in the beginning, only combinations of clearly wrong tracks are eliminated, so only Purity increases while Efficiency remains constant. When the threshold reaches a point where it is difficult to determine whether it is a wrong combination of tracks or a right combination of tracks, the right combination of tracks and the wrong combination of tracks are eliminated together. Therefore, Efficiency begins to fall, and Purity rises more slowly.
The problem is the magnitude of Efficiency when no threshold is set, and the magnitude of Purity when the threshold is tightened, and Efficiency begins to fall.
The Kalman filter has an extremely low Efficiency even if the threshold is not set from the beginning. In other words, it cannot provide small residuals for almost all combinations that can be track-matched correctly. And by making the threshold quite strict, only those combinations that can easily be correctly track-matched remain, and Purity can be made high. This feature means that the Kalman filter has a fundamentally poor ability to determine whether a combination of tracks is correct. The Kalman filter uses five parameters $X, Y, \phi, \tan\lambda, q/p_T$ and its covariance matrix. When track matching is performed, extrapolating the MCH and MFT reconstructed tracks to an arbitrary matching plane, Purity was higher when this plane was set to the MFT side than the front absorber[21]. What this means is that the Kalman filter places more importance on momentum information than on position. The reason is that MFT has a higher position resolution, and MCH has a higher momentum resolution. Therefore, if the position is essential, the track reconstructed

by MFT should be extrapolated longer so that the position error does not become large. If momentum is essential, the track reconstructed by MCH should be extrapolated longer so that the momentum error does not become large. Therefore, it can be said that the Kalman filter emphasizes momentum. However, as shown in Figure 2.7, machine learning places more importance on position and achieves higher performance than the Kalman filter. Therefore, the Kalman filter looks at momentum when the position should be more important than momentum, and this is why the performance of track matching was low.
Next, in the case of classification using machine learning, first of all, the overwhelmingly higher Efficiency than the Kalman filter is mainly because machine learning was able to determine that location is more important. In addition, it can simultaneously delimit an overwhelmingly larger number of parameters in a multidimensional space than the Kalman filter. Furthermore, the Kalman filter does not have a learning function, but in machine learning, once a trial track matching is performed, it is iteratively improved to improve the value of the loss function. Because these features were not found in the Kalman filter, higher performance was achieved than with the Kalman filter when machine learning was used. And without the cut, ranking had a higher Efficiency than classification because, as discussed in Chapter 2, classification learns to maximize the purity and efficiency of each of the two classes created by simply classifying. Still, when it comes to actual track matching, all algorithm has to do is just set the highest predicted probability to the correct track pair, regardless of whether the classification fails. However, when it comes to actual track matching, even if the classification fails, the correct track combination should be given a higher predicted probability than any other candidate track. In this respect, ranking is more efficient than classification for increasing Efficiency and Purity. Therefore, we believe that the ranking would have performed better when cuts were not used. When cuts were used, there was a tendency for Purity to increase instead of Efficiency to decrease in both classification and ranking. This is natural. However, both Efficiency and Purity improved in classification when a cut was added based on the distance on the matching plane between the track reconstructed by MCH and its matching candidates. We believe that this is because using the cut improves the learning process and prediction accuracy. This is because a cut such as the one used in this study does not reduce the information on the distribution boundaries in classification but can exclude candidates for obviously wrong tracks that are far from the boundaries. Therefore, the effect is similar to down-sampling. However, a stricter cut will still lower the Efficiency.
Next, the signal-to-noise ratio of the invariant mass distribution of dimuon was compared between the machine learning-based classification and the Kalman filter, and the machine learning-based classification had a higher signal-to-noise ratio. Although the Kalman filter has higher Purity, it does not have enough Efficiency to reconstruct two correlated muons in the first place, so when the combinatorial background is subtracted using the Like-Sign method, the number of signal components that remain is reduced. On the other hand, classification using machine learning has more Efficiency but less Purity than the Kalman filter, which means that there are some signal components but also many noise components. However, even if the track matching did not match correctly, we believe that the

shape of the distribution and the signal-to-noise ratio did not differ significantly because it makes regrettable mistakes rather than serious ones.

6

CONCLUSIONS

6.1 MAJOR CONTRIBUTION

In ALICE, a new detector MFT was introduced from LHC-Run3 and used in combination with the existing detector MUONspectrometer, and the Kalman Filter was proposed as an algorithm to correctly link tracks reconstructed by the two detectors. Track matching using the Kalman filter was not very practical in the low Transverse momentum region (around 1 GeV), because the Purity was very high (about 70%) but the Efficiency was very low (about 10%). When examining the relationship between hadronic masses and spontaneous breaking of the chiral symmetry, the reconstruction of invariant masses from low transverse momentum muon pairs produced by decay modes with very small branching ratios($\rho, \omega, \phi \rightarrow \mu\mu$) requires that both muon pairs be reconstructed even with some unwanted tracks in the mix. If Efficiency is low(10%), either one of the muon pairs (about 90% probability) or both of them (about 81% probability) will not be reconstructed correctly due to the small Efficiency. Therefore, low efficiency is a disadvantage. In order to improve the Efficiency, we proposed a new method using machine learning. As a result, we succeeded in performing track matching with significantly higher Efficiency than the Kalman filter. Track matching based on machine learning classification was able to improve Efficiency to about 40% for single muons with transverse momentum around 1 GeV, although Purity dropped to about 40%, by using cuts for matching candidates. Furthermore, track matching based on machine-learning ranking improved Efficiency to about 60% while maintaining the same level of Purity as track matching based on classification without using any cuts. This means that machine learning classification-based track matching reduces the probability that one of the muon pairs around 1 GeV of Transverse momentum is not correctly reconstructed to 60% and the probability that both are not correctly reconstructed to 36%, while rank-based track matching with machine learning reduces the probability that one of the muon pairs around 1 GeV of Transverse momentum is not correctly reconstructed to the probability of one being incorrectly reconstructed was reduced to 40%, and the probability of both being incorrectly reconstructed was reduced to 16%. The invariant spectra of the muon pairs were also reconstructed with the Kalman filter and track matching with machine learning based classification, respectively. The machine learning-based classification improved the signal-to-noise ratio by about 6% over the Kalman filter. There was concern that the S/N ratio would deteriorate because Purity was reduced in machine learning instead of increasing Efficiency, but it appears that the increase in Efficiency was more than enough to improve the S/N ratio. This brings us closer

to verifying the relationship between hadronic masses and spontaneous breaking of chiral symmetry.

6.2 OUTLOOK AND FUTURE WORK

The GBDT used in this study has various hyperparameters. These have not yet been optimized, and it is expected that adjusting them will further improve the signal-to-noise ratio. In addition, other S/N ratios even better than the present results could be obtained by reconstructing the invariant mass spectrum with a different threshold value for the predicted probability of machine learning.
We have evaluated the signal-to-noise ratio of invariant mass spectra for track matching by classification using machine learning. Still, we have not yet evaluated the signal-to-noise ratio of invariant mass spectra for track matching by ranking using machine learning. Since ranking was more successful in track matching with higher efficiency than classification, we believe that a higher signal-to-noise ratio can be expected. Furthermore, classification and ranking using deep learning instead of GBDT is currently under study and is expected to have even higher performance than GBDT.

BIBLIOGRAPHY

[1] K. Yagi, T. Hatsuda, Y. Miyake, *Quark-Gluon Plasma*. 2005, ISBN: 13 978-0-521-56108-2.

[2] M. E. Peskin, D. V. Schroeder, *An Introduction To Quantum Field Theory*. 1995, ISBN: 13 978-0201503975.

[3] ALICE Collaboration, *ALICE Schematics as during RUN3 (after upgrade)*, ser. ALICE-PHO-SKE-2017-001-5.

[4] ——, *Addendum of the Letter of Intent for the upgrade of the ALICE experiment: The Muon Forward Tracker*, ser. CERN-LHCC-2013-014.

[5] ——, *Technical Design Report for the Dimuon Forward Spectrometer*, ser. CERN-LHCC 99-22.

[6] ALICE, MFT Collaboration, *Schematic view of the ALICE MFT detector*, ser. OPEN-PHO-EXP-2020-004-13.

[7] ALICE Collaboration, *Technical Design Report for the Muon Forward Tracker*, ser. CERN-LHCC-2015-001.

[8] J. H. Friedman, *Greedy Function Approximation: A Gradient Boosting Machine*. 1999.

[9] Kalman, R. E., *A New Approach to Linear Filtering and Prediction Problems*, ser. Transaction of the ASME—Journal of Basic Engineering, pp. 35-45. 1960.

[10] A. Strandlie, R. Fru hwirth, *Track and vertex reconstruction: From classical to adaptive methods*. DOI: 10.1103/RevModPhys.82.1419.

[11] Lucas Nunes Lopes, Luis Gustavo Pereira, Rafael P. Pezzi, *Transformations between MFT and MCH coordinate systems*.

[12] Andrea Dal Pozzolo, Giacomo Boracchi, Oliver Caelen, Giauluca Bontempi, *When is undersampling effective in unbalanced classification tasks?*, ser. Machine Learning and Knowledge Discovery in Databases. Springer.

[13] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar, *Focal Loss for Dense Object Detection*, ser. ICCV.

[14] Andrea Dal Pozzolo, Oliver Caelen, Reid A. Johnson, Giauluca Bontempi, *Calibrating Probability with Undersampling for Unbalanced Classification*, ser. Dune Chronicles, Book 3. DOI: 10.1109/SSCI2015.33.

[15] H. Bichsel, D. Grooms, S. Klein, *27. Passage of Particles through Matter 1*, ser. Corpus ID: 8104407.

[16] ALICE Collaboration, *Update of the online-offline computing system technical design report*, ser. CERN-LHCC-2015-006 ; ALICE-TDR-019.

[17] Gauss Centre for Supercomputing, *Numerical Determination of the Phase Diagram of Nuclear Matter*.

[18] T. Hatsuda, T. Kunihiro, *QCD Phenomenology based on a Chiral Effective Lagrangian*, ser. Phys.Rept.247:221-367,1994. DOI: https://doi.org/10.1016/0370-1573(94)90022-1.

[19] W.Weise, *Nuclear aspects of chiral symmetry*, ser. Nuclear Physics A Volume 553, 1 March 1993, Pages 59-72. DOI: https://doi.org/10.1016/0375-9474(93)90615-5.

[20] Wu, B., Du, X., Sibila, M. et al., *X(3872) transport in heavy-ion collision*, ser. Eur. Phys. J. A 57, 122 (2021). 2021. DOI: https://doi.org/10.1140/epja/s10050-021-00435-6.

[21] Ren Ejima, *Foward Muon Track Reconstruction Considering Multiple Coulomb Scattering in ALICE Run3(ALICE jikken Run3 ni okeru zenpou mu ryushi no tajuu sanran wo kouryo shita hiseki saikousei, in Japanese)*. 2021.

ACKNOWLEDGEMENT

I would like to take this opportunity to thank all those who have assisted me in carrying out this research.
First, I would like to express my utmost gratitude to my primary supervisor, Kenta Shigaki. He has provided us with a wide range of physics discussions based on his extensive experience and knowledge. I would also like to express my utmost gratitude to Satoshi Yano. He is also involved in track matching using machine learning, and we have discussed and improved our machine learning techniques and knowledge together on a daily basis. In addition to Satoshi Yano, Rita Sadek, Maxime Guilbaud, Robin Caron, Lucas Nunes Lopes, and Maurice Louis Coquet are members of the ALICE track matching project, and we discuss and share information on a daily basis. They have made the research process very smooth. I have also had discussions with Yorito Yamaguchi and Motomi Oya in the same laboratory on the status of MFT and physics. I would also like to thank Rafael Pezzi for his long technical support. I would also like to thank Stefano Panebianco, Antonio Uras, and the rest of the MFT team for their help during our weekly meetings. I would also like to thank Kensuke Homma, Takahiro Miyoshi and Masanori Ogino. They give me so many general comments and it was very important for my glowing as a researcher. I would also like to thank Keisuke Tomohiro and Koki Soeda and the rest students in the same lab, with whom I had regular discussions about physics and detectors. Thank you very much.