Master's Thesis

The study on measurements of beam parameters in 250GeV ILC through the machine learning method

SHI KANG

Department of Physics
Graduate School of Science
Tohoku University
2021

The study on measurements of beam parameters in 250GeV ILC through the machine learning

SHI KANG

Abstract

The International Linear Collider is now under consideration as the next global project in particle physics. In ILC, it is essential to have very flat beams at the interaction point in order to obtain high luminosity.A measurement of the beam size is therefore extremely important for keeping luminosity as high as possible. In this research, Pair Monitor has been executed by utilizing the already implemented geometry of another forward calorimeter, namely BeamCal. By means of full simulation of process happened in linear collider, information is obtained from detector.Then building appropriate model of machine learning to precisely predict beam size by a number of training data from simulation. Otherwise, introduce data analysis method for reducing statistical fluctuations and improve accuracy in model. The result of prediction are almost close to 100% in defined range.

Acknowledgments

First and foremost I am extremely grateful to my supervisors, Prof.Sanuki and Dr.Yonamine for their invaluable advice, continuous support, and patience during my study. Their immense knowledge and plentiful experience have encouraged me in all the time of my academic research and daily life. I would also like to thank Prof.Ichikawa Dr.Yuichi for their technical support on my study. I would like to thank all the members in our Lab. It is their kind help and support that have made my study and life in the Japan a wonderful time. Finally, I would like to express my gratitude to my parents. Without their tremendous understanding and encouragement in the past few years, it would be impossible for me to complete my study.

Contents

List of Figures

Chapter 1

Introduction

In the high energy physics field, powerful accelerator,clever detector and precise measurement are significant part which helped made new discoveries. Meanwhile, artificial intelligence technologies swarm into many kinds of industry and field as a effective and importance tool in nearly a decade. Artificial intelligence technologies also have immense potential in high energy physics. For example, machine learning of Artificial Intelligence could made it possible to predict certain quantities which are hard work in past, and deeply seek for hidden information from events. The machine learning is an application of Artificial Intelligence that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. And it focuses on the use of data and algorithms to imitate the way that humans learn, gradually improving its accuracy, in the other word, use historical data as input to predict new output values. So it's a available algorithm for some prediction that beyond human's ability.

The International Linear Collider (ILC) has been designed to address many central physics issues. Compare with Large Hadron Collider, it will extend and complement the LHC physics program. And the ILC require very flat beams at the collision point in order to ensure maximum luminosity in running. Pair Monitor are designed for measuring the size of interacted beam and make sure the adequate luminosity can be provided. So the high precision

of measurement are most importance point in Pair Monitor[26]. After simulation of interaction process, it is obvious that information from Pair Monitor detector is sensitive to horizontal and vertical beam size. Therefore the machine learning model could predict beam size through observable from pair monitor. In earlier research[21][19], different machine learning model have been utilized, but the accuracy are short of expectation due to statistics fluctuations. So I introduced data analysis method for dealing with those problems.

In my thesis, first a short introduction of construction and interaction principle in the ILC. Then represent simulation about ILC detector environment and interaction process. Afterward introducing how to applied Machine Learning model to predict parameter of beam size, besides the method for developing model are implemented. Finally, some suggestions about how to improve future research on this topic would look like is discussed.

Chapter 2

International Linear Collider

2.1 Overview of the accelerator

The international Linear Collider (ILC)[7] is an electron-positron collider,will be a necessary tool for unlocking some of the deepest mysteries about the universe.Its best feature is it will allow physicists to precisely explore extremely high-energy regions between 250 GeV and 1000 GeV.The ILC consisting of two linear accelerators that will stretch approximately 31 kilometers in length, and it will smash electrons and their anti particles, positrons, together at nearly the speed of light and colliding nearly 7000 times every second.The ideal result is that the electrons and positrons will create an array of new particles that could help answer some of the most fundamental questions of all time: What is the higgs boson? What are dark matter and dark energy? Does supersymmetry exist? ILC will help achieve this goal by means of its clean environment, high luminosity and beam polarization. The following section would describe the chief components of the accelerator system.

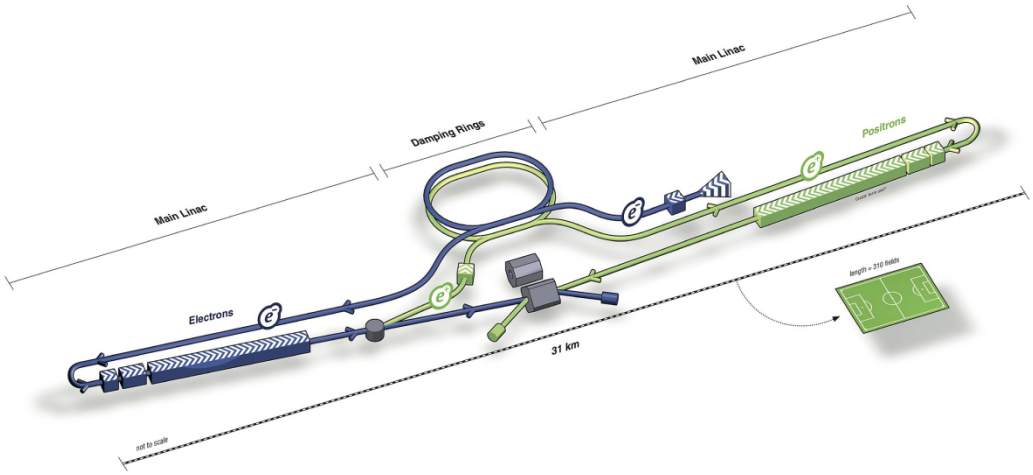

Figure 2.1: A schematic of the ILC[8]

2.2 Accelerator

The ILC accelerator which shown in Fig 2.1 consists of superconducting RF main Linacs, electron source,positron source, damping ring and a beam delivery system.

2.2.1 Superconducting RF main linacs

The heart of the ILC accelerator consists of the two superconducting Main Linacs that accelerate both beams from 5 to 125 GeV. The beams are accelerated in 1.3GHz nine-cell superconducting cavities made of niobium and operated at about 2 K(Fig 2.2). These are assembled into cryomodules comprising nine cavities or eight cavities plus a quadrupole / corrector / beam position monitor unit, and all necessary cryogenic supply lines. The cryogenic plants could cool a continuous length of 2.5 km of Linac. The main Linac is made with the curvature of the earth in mind to ensure smooth supply.

Figure 2.2: A 1.3 GHz superconducting niobium nine-cell cavity.[8]

2.2.2 Electron sources

The electron source design[4] is based on the SLC polarized electron source, which has demonstarted that the bunch charge, polarisation and cathode lifetime parameters are feasible.The laser impinging on a photocathode based on a strained GaAs/GaAsP superlattice structure, which will produce electron bunches with an expected polarisation of 85%.After bunching and accelerating to 76 MeV through normal conducting structure,superconducting solenoids rotate the spin vector into vertical and finally the beam is injected into a damping ring.

2.2.3 Positron sources

After electron beam is passed through a helical undulator, the hard gamma rays are produced which are converted to positron in a rotating target. Positrons are captured in a flux concentrator or a quarter wave transformer, accelerated to 400 MeV in two normal conducting preaccelerators followed by a superconducting accelerator very similar to the main Linac, before they are injected into the damping rings at 5 GeV.

2.2.4 Damping rings

The ILC includes two oval dampling rings of 3.2 km circumference, sharing a common tunnel in the central accelerator complex. The damping rings reduce the horizontal and vertical emittance of the beams by almost six orders of magnitude within a time span of only 100 ms, to provide the low emittance beams required at the interaction point.Both damping rings operate at an energy of 5 GeV.

2.2.5 Ring to Main Linac(RTML)

The Ring to Main Linac(RTML) system is responsible for transporting the beams from Damping Rings at the entrance of the accelerator complex to

the upstream ends of the Main Linacs, collimating the beam halo generated in the Damping Rings,and rotating the spin polarisation vector from the vertical to the desired angle at the IP(Interaction Point).

2.2.6 Beam delivery system

The Beam delivery system transports the electron-positron beam from the end of the main Linacs, focuses them to the require small beam spot at the Interaction Point,brings them into collision, and transports the spent beam to the main dumps.

2.3 International Linear Detector

The ILC detector are designed to make precision measurements on the Higgs boson,W,Z,τ,and other particles. The ILD are selected to meet the requirement for such measurements. It results in a large detector optimized for good energy and momentum resolution, with flexibility for operation at energies up to the TeV range, and employs a highly granular calorimeter, with minimal material between the interaction point and the calorimeter. The tracker is a Time Projection Chamber(TPC) providing continuous tracking for excellent pattern recognition and dE/dX capability.

The interaction region of the ILC is designed to host two detectors, which can be moved in and out of the beam position with a 'push-pull' scheme. The mechanical design of ILD and the overall integration of subdetectors takes these operational constraints into account.

The ILD concept shown in Fig 2.3 has been designed as a multi-purpose detector[1]. A high precision vertex detector is followed by a hybrid tracking layout, realised as a combination of silicon tracking with a time projection chamber, and a calorimeter system inside the large solenoid.

The momenta of the full set of final-state particles are best reconstructed with the Particles Flow Algorithm(PFA)[15]. This technique combines the

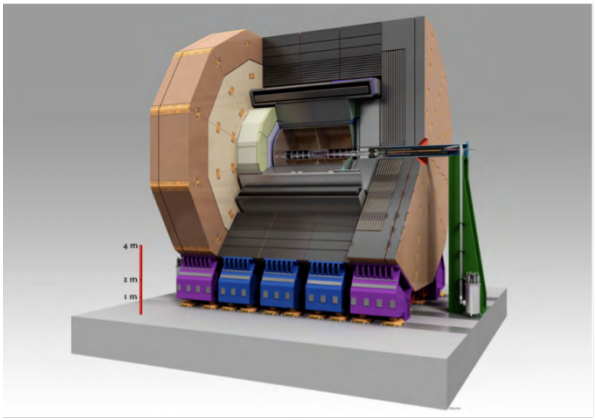

Figure 2.3: The ILD detector concept[5]

information from the tracking system and from the calorimetric systems to reconstruct the energy and the direction of all charged and neutral particles in the event. Besides, the requirements of large distance between interaction point and sweep charged particles away from the neutral, make particle reconstruction by PFA superior to previously used energy flow scheme.

2.3.1 Detector component

Vertex system

The vertex detector is realized as a multi-layer pixel vertex detector with three super-layers. The detector positioned very closely to the interaction point and has a pure barrel geometry to minimize the occupancy from background hit, the first layer located at a radius of about 1.6 cm. Besides it is optimized for point spatial resolution better than 3 μm and material budget below 0.15% χ_0/layer.

Silicon tracking system

The silicon part of the ILD tracking is made of four component: two barrel component, the Silicon Inner Tracker(SIT) and the Silicon External Tracker(SET). One end cap component behind the end-plate of the TPC.and the forward tracker. The SIT and SET help improving the overall momentum resolution

and linking the VTX detector with the TPC, and in extrapolating from the TPC to the calorimeter.

TPC

A distinct feature of ILD is a large volume Time Projection Chamber with up to 224 points per track that is optimized for 3-dimensional point resolution and minimum material in the field cage and in the end-plate. Its low material budget substantially for best calorimeter and PFA performance. To obtain good momentum resolution and to suppress backgrounds, the detector will be situated in a strong magnetic field of 3.5T. Under this condition a point resolution of better than 100 μm for complete drift and a double hit resolution of less than 2mm becomes possible.

Electromagnetic Calorimeter system

Electromagnetic Calorimeter is an experimental detector that measures the energy of electron, positron and photon that interact primarily via the electromagnetic interaction. It can distinguish between overlapping showers, do pattern recognition of the showers and reconstruct photons even in presence of particles nearby.

Hadronic Calorimeter system

Hadronic Calorimeter measure the energy of particles that interact via the strong nuclear force. It separates the charged hadrons from the neutral ones and thus contributes highly to particle flow resolution for jet energies up to 100 GeV.

Forward Calorimetry

In the very forward region shown in Fig 2.4, three systems: LumiCal, BeamCal and LHCAL, are proposed. Theses system serve as luminosity monitor

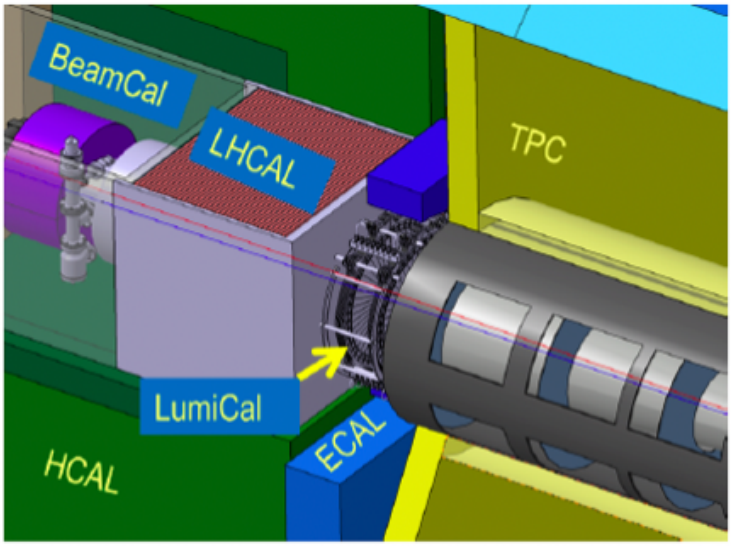

Figure 2.4: The very forward region of the ILD detector[2]

(LCAL) and beamstrahlungs monitor (BeamCal), and they close the coverage down to very small angles, also for neutral hadrons (LHCAL).

LumiCal

LumiCal carried out precision measurement of luminosity by means of Bhabha scattering. Its small pad size is very suitable for accurate measurement of showers with very small polar angles.

BeamCal

BeamCal is positioned just outside the beam-pipe and hit by massive amount of Beamstrahlung pairs and along with Pair Monitor it makes bunch-by-bunch measurement of luminosity and employs a shower finding algorithm which can detect pair backgrounds even at low polar angles.

Pair Monitor

Pair Monitor measure the parameter of beam size in front of BeamCal. And it's described at next section in detail.

2.4 Physics case

Today, the LHC experiments could achieve 20% uncertainties in their measurements of Higgs boson couplings. We believe the LHC is hard to lead to Higgs coupling measurements of the required high precision. So the core of the physics case for the ILC is to make high-precision measurements of the properties of the Higgs boson. As we know, the Higgs field has a central role in the SM which is responsible for not only the masses of all known elementary particles but also those aspects that are hardest to understand (such as hierarchy of quark and lepton masses, flavor mixing, CP violation and etc.). If we wish to learn more about these features of the fundamental laws of nature, an obvious course is to measure the Higgs boson as well as we are able. We will argue some physics case at ILC of 250GeV below.

- $e^+e^- \to Zh$

This is a different way to measure Higgs boson couplings provided by the ILC. It is measurement of the reaction $e^+e^- \to Zh$ at 250 GeV. At an e^+e^- collider at this energy, it is true to a first approximation that any Z boson observed with a lab energy of 110 GeV is recoiling against a Higgs boson. The background come from radiative $e^+e^- \to Z\gamma$ and $e^+e^- \to ZZ$.which should identify Higgs boson events independently of decay mode, allowing the measurement of the total cross section for Higgs production and the discovery of exotic and unanticipated Higgs decays.

- $e^+e^- \to W^+W^-$

The reaction $e^+e^- \to W^+W^-$ provides an excellent way to test for the presence of dimension-6 operators that involve the W and Z fields. The Feynman diagrams of reaction show in Fig 2.5.In the SM, there are large cancellations among these diagrams, but these are not respected by the dimension-6 contributions. Thus, the dimension-6 coefficients appear in the cross section formula enhanced by a factor S/m_W^2.[17]

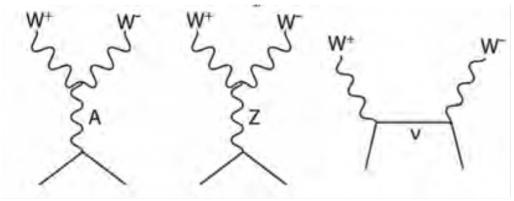

Figure 2.5: $e^+e^- \to W^+W^-$

- $e^+e^- \to f\bar{f}$

The Fermion pair production could provides a search for new forces that couple directly to the electron. In ILC, it operate at an appropriate energy but with much higher luminosity, which corresponds to the ability to observed new vector bosons at 5-6 TeV and contact interaction scales of 70 TeV, comparable to the projected reach of the HL-LHC.

- Search for pair-production of new particles

Due to insensitivity to new particles with electroweak interactions only that decay to an invisible partner with a mass gap of less than 5 GeV and insensitivity to production of pairs of invisible particles observed through radiation of an initial-state gluon, LHC has blind spots coreesponding to physically interesting models. Contrarily, the ILC can detect these new physics events for particle masses almost up to half of the collider center of mass energy.

From physics case described above, keeping luminosity as high as possible is indispensable for well running of ILC. Besides, measurement of luminosity is also a important role at ILC, that is the reason for implementation of Pair monitor

Chapter 3

Pair Monitor

3.1 Introduction

In ILC, it is essential to have very flat beams at the interaction point in order to obtain high luminosity[26]. Because of this goal, an precise meausrement of lunminosity is required. The quantity that measures the ability of a particle accelerator to produce the required number of interactions is called the luminosity and is the proportionality factor between the number of event per second dR/dt and the cross section σ_p:

$$\frac{dR}{dt}[S^{-1}] = L[S^{-1} * cm^{-2}] * \sigma_p[cm^{-2}] \tag{3.1}$$

For transverse Gaussian beam distributions, luminosity can be define as[13]:

$$L = \frac{N_1 N_2}{4\pi\sigma_x\sigma_y} n_b f H_D \tag{3.2}$$

Where N_1 and N_2 are number of particle in one bunch for incoming and outgoing beam respectively and transverse collision area is described as $4\pi\sigma_x\sigma_y$. f is the revolution frequency and n_b is the number of bunches in one $beam$.H_D is additional factor of luminosity due to the cross angles.It

shows parameter of beam sizes are inversely proportional to luminosity. So a appropriate beam size are benefit of maintaining stable luminosity at ILC.

3.2 Beamstrahlung

When a particle is forced on a curved trajectory by the other beam, it will emit radiation in a similar fashion as in a bending magnet. This radiation is called beamstrahlung. This leads to the formation of a luminosity spectrum and obviously impacts the performance of the physics experiments. The beamstrahlung can be described by its critical energy $\hbar w_c$, which can be wirrted as[23]:

$$\hbar w_c = \frac{3}{2}\frac{\hbar\gamma^3 c}{\rho} \tag{3.3}$$

where ρ is the bending radius of the particle trajectory,γ is gamma parameter and c is speed of light. The beamstrahlung parameter writted as:

$$\Upsilon = \frac{3}{2}\frac{\hbar w_c}{E} \tag{3.4}$$

The beamstrahlung spectrum is described by the Sokolov–Ternov spectrum:

$$\frac{d\dot{\omega}}{d\omega} = \frac{\alpha}{\sqrt{3}\pi\gamma^2}[\int_x^\infty K_{5/3}(x^{'})dx^{'} + \frac{\hbar\omega}{E}\frac{\hbar\omega}{E-\hbar\omega}K_{2/3}(x)] \tag{3.5}$$

where $x = \frac{\omega}{\omega_c}\frac{E}{E-\hbar\omega}$ and $K_{5/3}$ and $K_{2/3}$ are the modified Bessel function. If the limit $\Upsilon \ll 1$ the power of the photon radiation of a particle is proportional to Υ^2:

$$P = \frac{e^2}{6\pi\epsilon_0}\frac{c}{\rho^2}\gamma^4 = \frac{2}{3}\frac{r_e c}{\lambda_c^2}mc^2\upsilon^2 \tag{3.6}$$

with $\lambda_c = \hbar/(mc)$. The average Beastrahlung parameter defined as:

$$\langle\Upsilon\rangle = \frac{5}{6}\frac{Nr_e}{\alpha\sigma_z(\sigma_x+\sigma_y)} \tag{3.7}$$

where α is fine structure constant. The maximum beamstrahlung parameter is about:

$$\Upsilon_{max} \approx \frac{12}{5} \langle \Upsilon \rangle \tag{3.8}$$

Generally speaking,spectrum corresponds to synchrotron radiation and one speaks of the classical regime, $\Upsilon \ll 1$.

3.3 Choice of beam parameters

In the classical regime, the number of photons emitted per beam particle n_γ depends on the bunch charge and transverse dimensions:

$$n_\gamma \in \Upsilon \frac{\sigma_z}{\gamma} \in \frac{N}{\sigma_x + \sigma_y} \tag{3.9}$$

Similarly, the average energy E_γ of each photon as:

$$E_\gamma \in \Upsilon \frac{1}{\gamma} \in \frac{N}{\sigma_z(\sigma_x + \sigma_y)} \tag{3.10}$$

From Eq 3.9, for reducing effect of Beamstrahlung, the values of $\sigma_x + \sigma_y$ have to be large, meanwhile , for high luminosity, the values of $\sigma_x \sigma_y$ have to be small. For both goals can be simultaneously achieved, therefore, flat beam with $\sigma_x \ll \sigma_y$ is the right choice for purpose because dampling ring naturally delivers beams with larger horizontal emittance and small vertical emittance.

3.4 Pair Background

There are three major method of producing photons by interaction between oncoming electron and positron without come from Beamstrahlung which shown at Fig 3.1,

Breit-Wheeler($\gamma\gamma \rightarrow e^+e^-$),Landau- Lifschitz($ee\gamma\gamma \rightarrow e^+e^-e^+e^-$) and Bethe-Heitler($e\gamma\gamma \rightarrow e^\pm e^+e^-$). This case of Beamstrahlung real photons interacting with themselves or oncoming e^+e^- is called incoherent process.

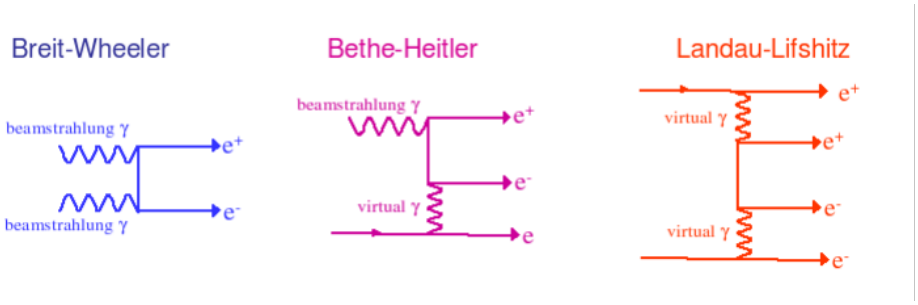

Figure 3.1: physical processes of photon production

3.5 Method of measurement

Many low-energy e^+e^- pair are expected to be created during beam crossing due to three incoherent processes. Where the γ is the beamstrahlung photon and this phenomena have been investigated as troublesome background for experiment at ILC. The particles of concern have the same charge as that of the oncoming beam, so most of them are deflected at larger angles than their inherent scattering angles by a strong electromagnetic force due to the oncoming beam and the particles with opposite charge are deflected with small angles like Fig 3.2. Since this potential is produced by the intense

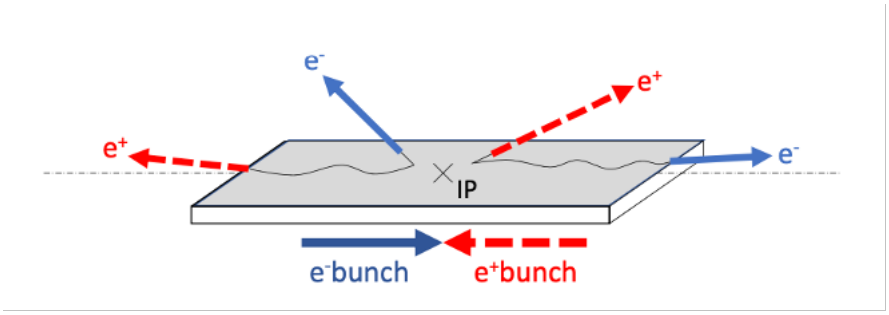

Figure 3.2: incoherent pair from Interaction Point[19]

electric charge of the oncoming beam, it is a function of the transverse beam size and intensity of the beam. So the deflected particles should carry this information, especially in their angular distribution and distance from the center axis. It is what I intend to measure. In this research, I assume the two beam have the same parameter of beam size.

3.6 Pair Monitor[3]

Additional and independent information on beam parameter will be obtained from the pair monitor.The device will consist of one layer of silicon pixel sensors, with pixel size of 400*400 μm^2, thickness of 300 μm and the distance from the interaction point is about 350 cm, just in front of BeamCal to measure the number density distribution of beamstrahlung pairs, that shown in Fig 3.3.

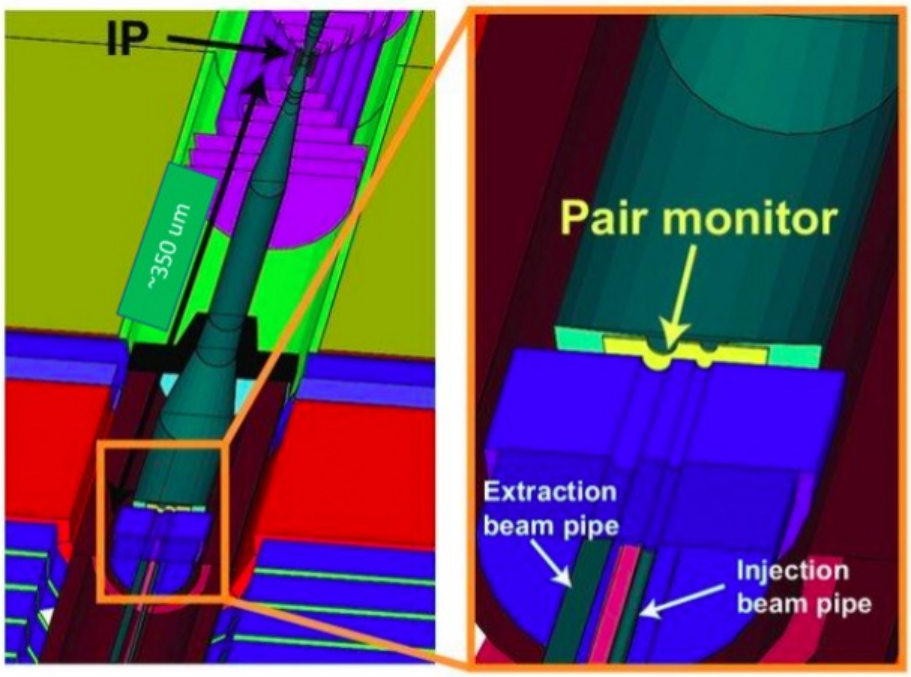

Figure 3.3: Detector geometry and location of the proposed pair monitor [25]

The nominal beam size of horizontal bunch sizes and vertical beam sizes are 729 nm and 7.7 nm respectively for 250 GeV. And the measurement of the beam size at the interaction point is crucial to maintaining a high luminosity because the luminosity is highly dependent on σ_x and σ_y. In my research, the full detector simulation of pair monitor has been undertaken, with the realization that the first layer of the BeamCal mimics pair monitor readout signal because of the same material and function.

Chapter 4

Simulations

4.1 Simulation of Beam-Beam Interaction

CAIN[18] is a standalone FORTRAN Monte-Carlo code for the interaction involving high energy electron, positron and photons.According to nominal beam size values (σ_x^*=729 nm,σ_y^*=7.7 nm), 0.8,0.9,1.0,1.1,1.2,1.3,1.4 times of the typical beam size values [9] have been simulated by CAIN. There are totally 7*7=49 sets of samples with different parameter combination of beam size. According to the convention of CAIN user's manual[18], the electron and positron beams have been simulated as right and left going beams respectively. The parameter used in the CAIN simulation described in Table 4.1. It is seen that each beam consist of 1312 bunches.But any observable from Pair Monitor by simulation is one bunch.

4.2 DD4hepSimulation Package[22]

The design of the DD4hep toolkit is shaped on the experience of detector description systems which developed for the Linear Collider community. DD4hep toolkit include a coherent set of tools, with most of the basic components already existing in one form or another.Fig 4.1 shows the architecture of the main components of the toolkit and their interfaces to the end-user

Energy(one beam)	125 GeV
Number of bunches	1312
Bunch Population	$2*10^{10}$
Number of macro-particles	100000
Collision Rate	Number of bunches *5Hz
Horizontal emittance	10
Vertical emittance	35
IP Horizontal β function	13.0 mm
IP Vertical β function	0.41mm
Slope($=\phi_c ross/2$)	7 mrad
External Field(B_x, B_y, B_z)	(0,0,3.5T)
Constant Field QED	BeamStrahlung Polarization Maximum event probability per time step= 0.5
Gaussian Tail cutoff $n_x, n_y, n_z, n_\epsilon$	4.5 (units of respective σ)
Polarization vector	$(\zeta_x, \zeta_y, \zeta_z)$ Electron Beam=(0,0,-1) Positron Beam=(0,0,1)
Beam-Beam Field	Horizontal bins=32 Vertical bins=128 Horizontal mesh width=12*$\sigma_x$height

Table 4.1: parameter of Simulation

applications, namely the simulation,reconstruction, alignment and visualization. The code is designed to optimize particle transport through complex structures and works standalone with respect to any Monte-Carlo simulation engine. The ROOT geometry package provides sophisticated 3D visualization functionality, which is ideal for building detector and event displays. The second component is the Geant4 simulation toolkit, which is used to simulate the detector response from particle collisions in complex designs. In DD4hep the geometrical representation provided by ROOT is the main source of information.

Currently the ideal detector description namely ILD-l5-v05, incorporates realistic detector geometry, solenoid field map, anti-DID field map and magnetic fields in the forward focusing magnets.Fig 4.2 and Fig 4.3 depicts the detector construction implemented by ILD-l5-v05 as a logarithmic color map

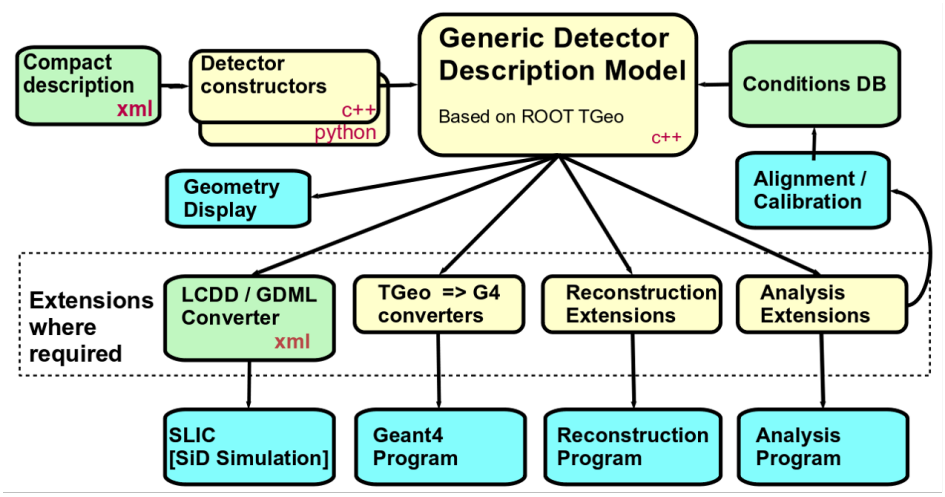

Figure 4.1: The components of the DD4hep detector geometry toolkit[10]

plot of number of radiation lengths in a bin with respect to the ILD detector coordinate system where y-axis point out of the page.

The tilted forward calorimetry is clearly visible here. Besides, the detector of interest i.e. BeamCal, which is about 300 cm away from the interaction point, is seen to have a very high radiation length.

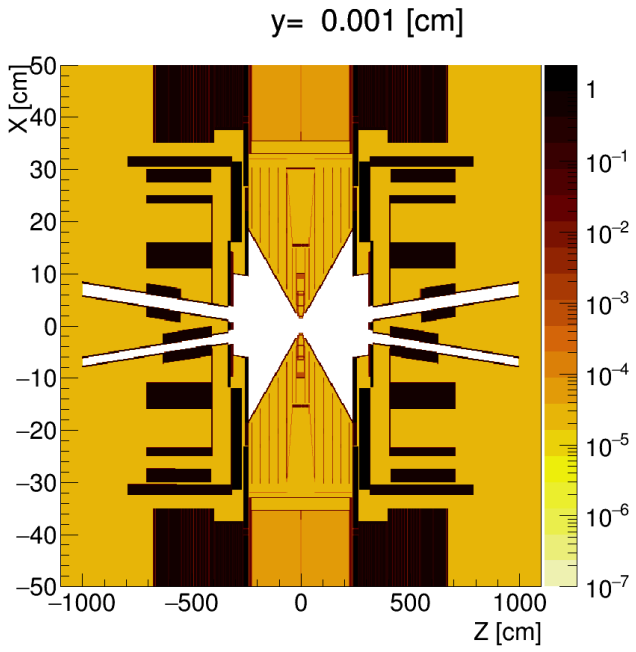

Figure 4.2: Cross Section of the ILD detector system[16]

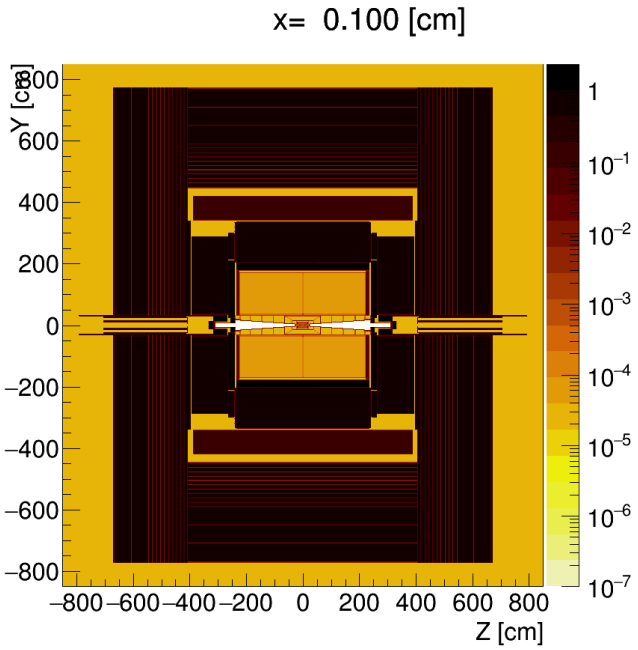

Figure 4.3: outline of the ILD detector system[16]

4.3 Marlin and LCIO packages

Marlin(Modular Analysis and Reconstruction for the Linear Collider) is a software framework based on ILCSOFT. LCIO is a persistency framework that defines a data model for linear collider detector studies, it is intended to be used in both simulation studies and analysis frameworks. Marlin implements a processor to digitise hit collection which then extracts the digitised collection through LCIO data model. [12]

Digitised information of an event are accessed inside a processor through the LCEvent class. Event information inside the BeamCal are accessed as LCCollection type object by means of SimCalorimeterHit class inside the processor. Each element of this collection corresponds to hits at the BeamCal cells. Each of this hit denote only the position of the cell hit by the particle and not the exact position i.e. coordinate values or layer number of the hit. Therefore, these collections are decoded by means of CellIDDecoder and layer number corresponding to every element of the BeamCal collection is obtained by passing the address of the element into decoder.

For the first layer, number of contributions by all MCParticles, inside each aforementioned element, is obtained using *getNMCContributions()*, which is a member function of *SimcalorimeterHit* class.

Due to multiple layers of BeamCal and the scatter while *MCparicles* pass through the layers,the number of contributions are higher than expect.Such contribution contain the energy, position, pdg etc. information calculated by Geant4.Only fisrt midpoint position between MCparticles and detector is required in PairMonitor, such MCparitcles are collected by using *getNMC-Contribution()*[20]. And the simulation of position distribution at first layer of BeamCal show at Fig 4.4. The central blank shape like a keyhole because of input pipe and output pipe.

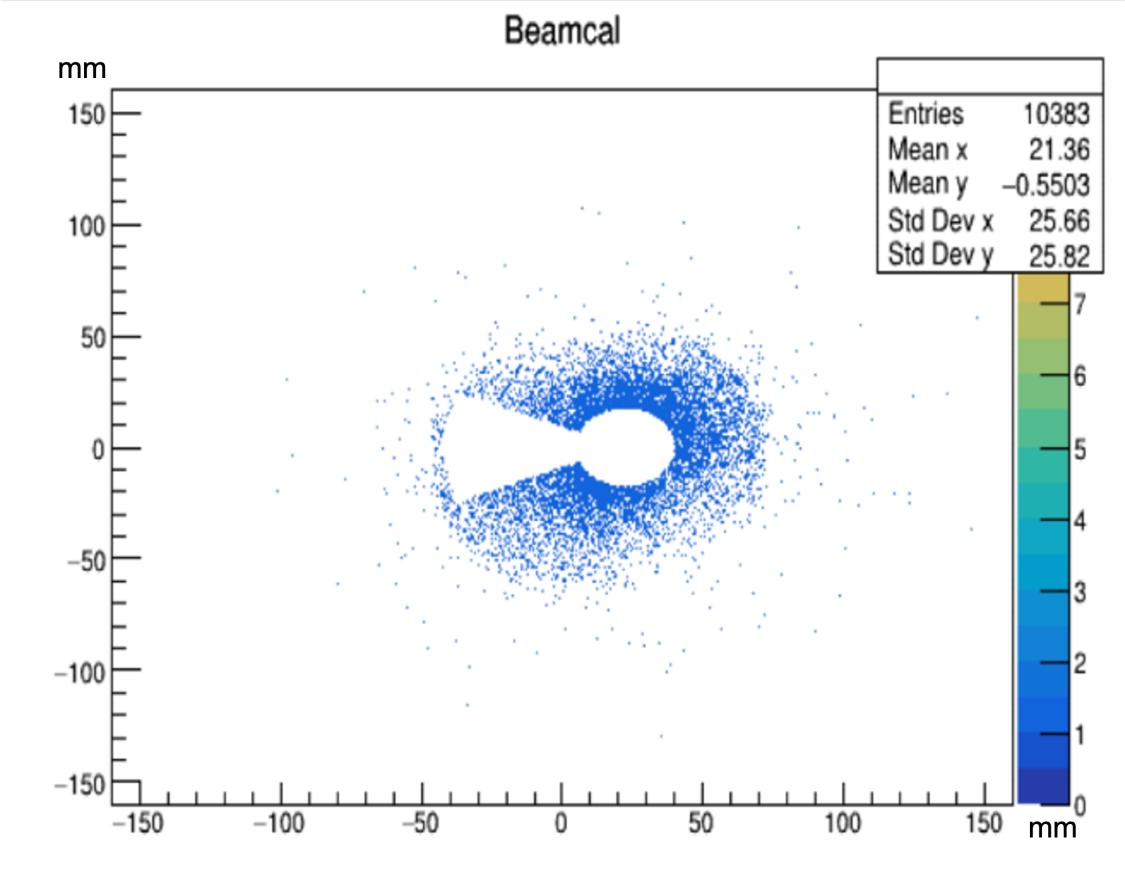

Figure 4.4: position distribution of the first layer of BeamCal(1 bunch) $(\sigma_x = \sigma_x^*, \sigma_y = \sigma_y^*)$

Chapter 5

Data analysis

As described at section 4.1, the beam like a train that contains a number of bunches(1312), each bunch look like a cube, the variables of beam size are σ_x and σ_y that is the width and length of the cube at Fig 5.1. According to different variables of beam size simulated different process of interaction and get different hit information from Pair Monitor that shown in Fig 5.2.There are 9 examples in results of simulation, it's difficult to look for difference and feature by naked eyes. So it's important to apply *dataanalysis* and $feature extract$ for discover the hidden information from hit distribution.

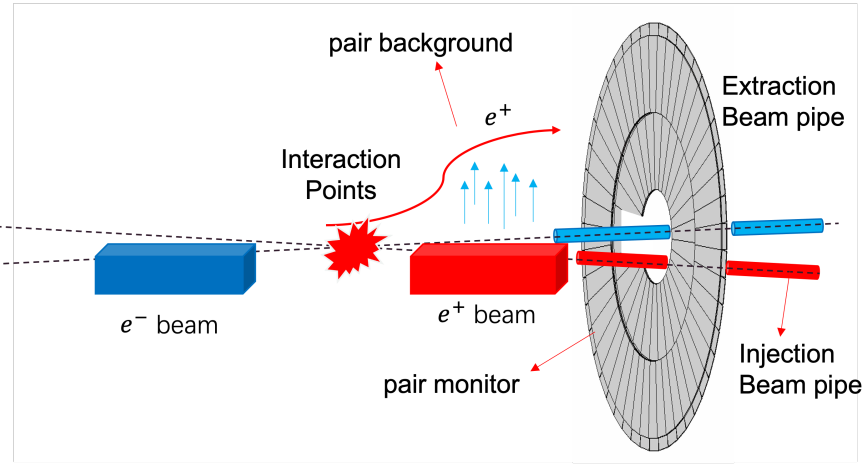

Figure 5.1: interaction after one bunch

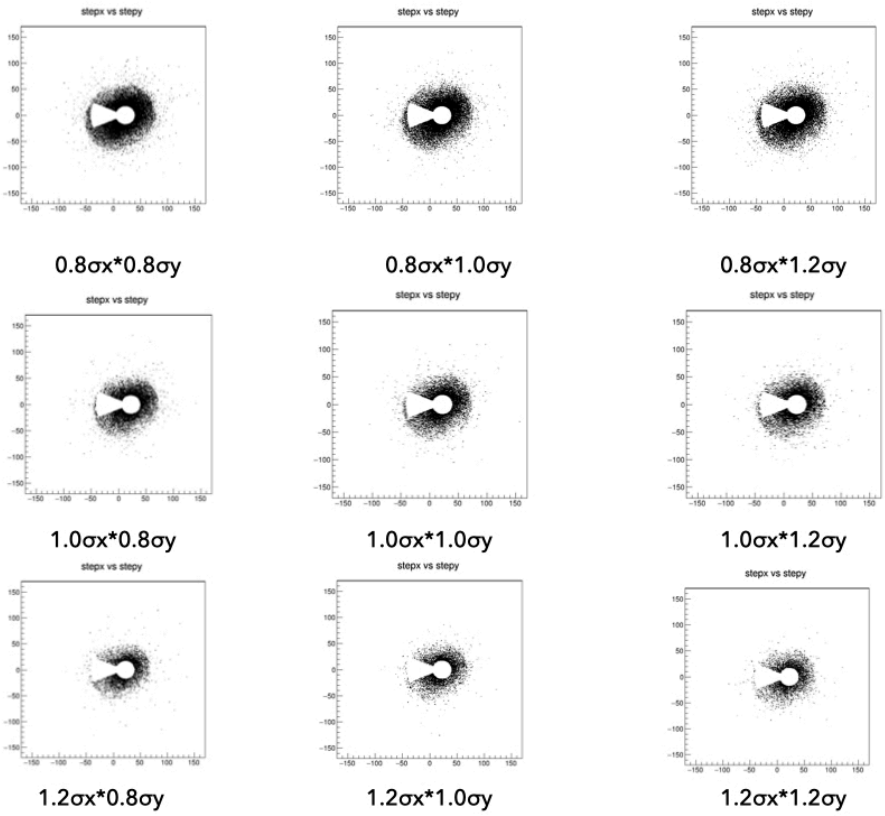

Figure 5.2: part of images of simulation(1 bunch)

5.1 Different parameter of beam size

As mentioned in section 4.1, simulation of 49 sets of parameters with each value of σ_x corresponding to 7 values of σ_y and vice versa, have been carried out. The Fig 5.3 and Fig 5.4 shows the azimuthal angle, $\phi = tan^{-1}(y/x)$ distributions for nominal σ_x with different σ_y and nominal σ_y with different σ_x respectively. *beam size pair* means one of 49 combinations of the horizontal and vertical beam size. The ϕ plots of different σ_x(Fig 5.4), vary across the whole region [-3,3] with small statistical fluctuations. But the fluctuations in the ϕ plot (Fig 5.3), that corresponds to different values of σ_y, are very vague and uncertain correlation of plots with the values of σ_y

The plot of ρ, $\rho = \sqrt{x^2 + y^2}$, that the distance of the particles from the center, also with different σ_x and different σ_y show in Fig 5.5 and Fig 5.6. The ρ plot have similar point with ϕ plot. There have very clear correlation of the plots with the different values of σ_x at the range of [10,80] in Fig 5.5.

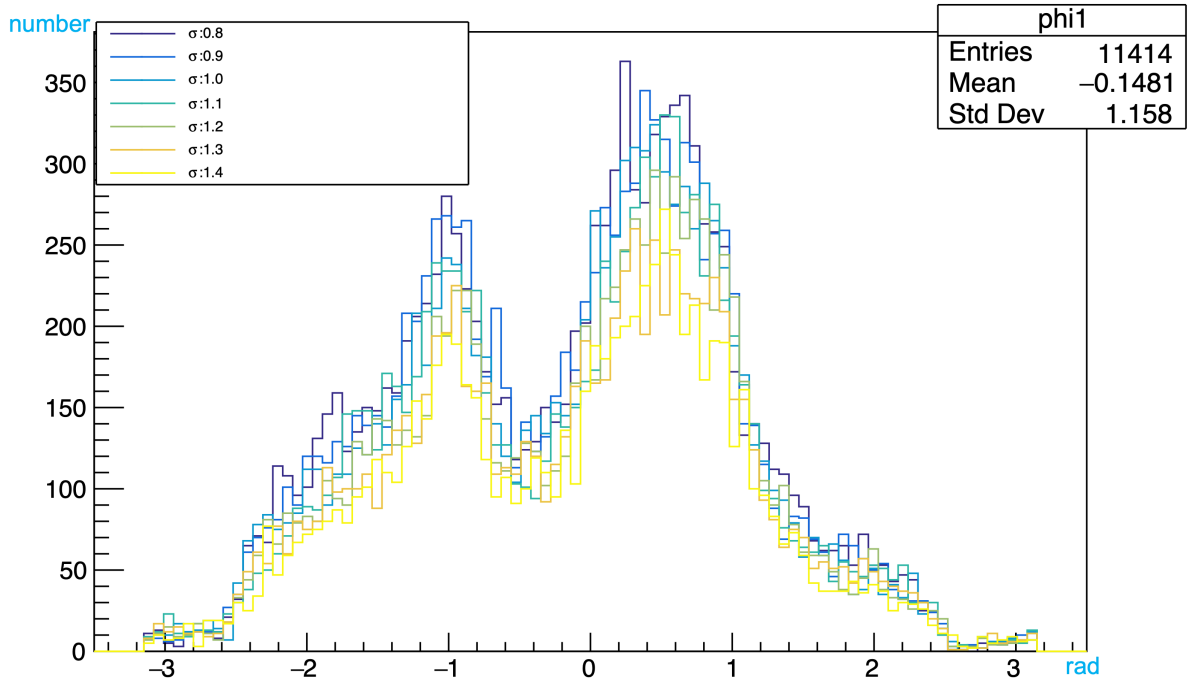

Figure 5.3: ϕ (radian) plots for different times to nominal values of σ_y and nominal σ_x (1 bunch/beam size pair)

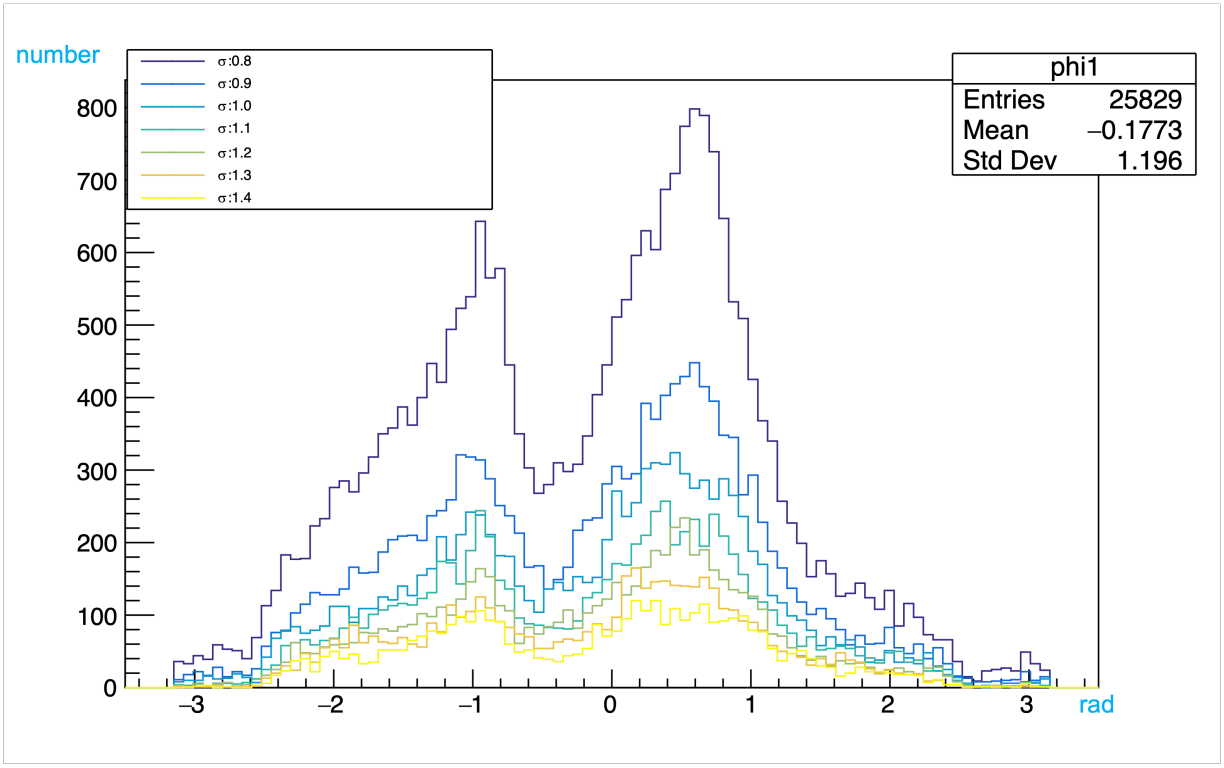

Figure 5.4: ϕ (radian) plots for different times to nominal values of σ_x and nominal σ_y (1 bunch/beam size pair)

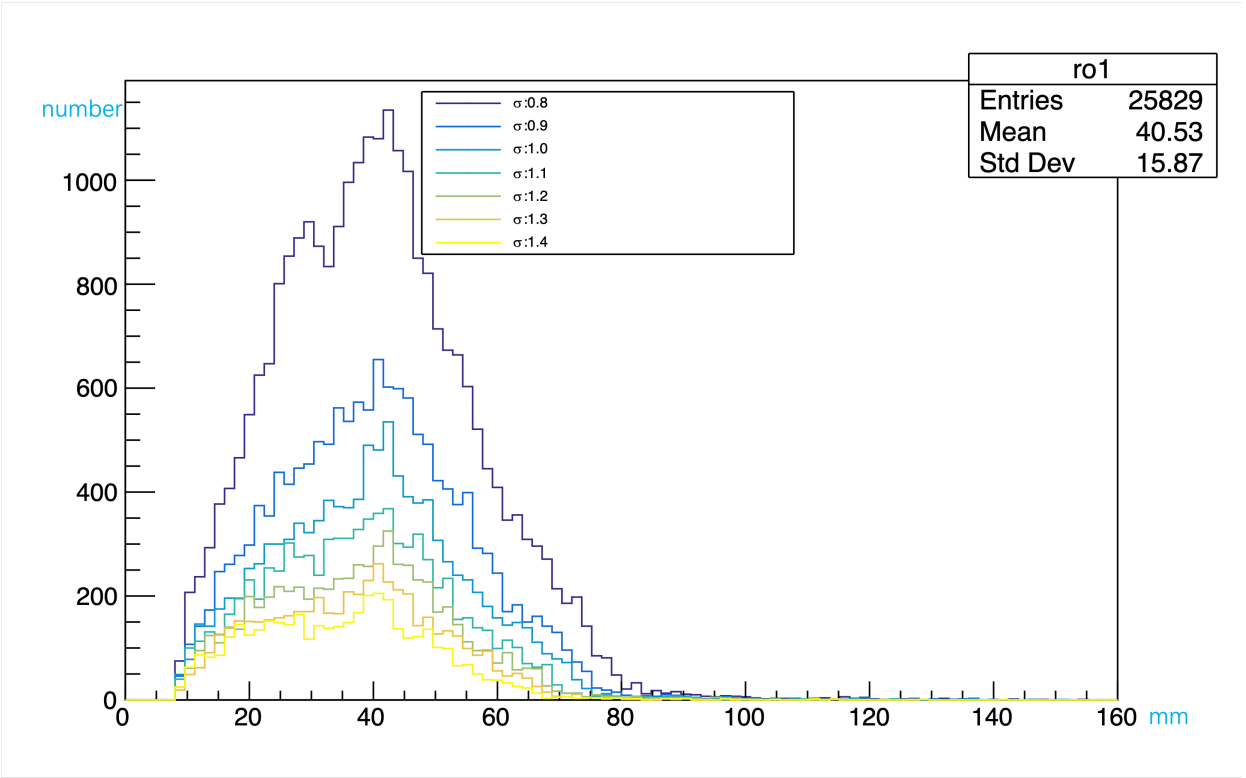

Figure 5.5: ρ plots for different times to nominal values of σ_x and nominal σ_y (1 bunch/beam size pair)

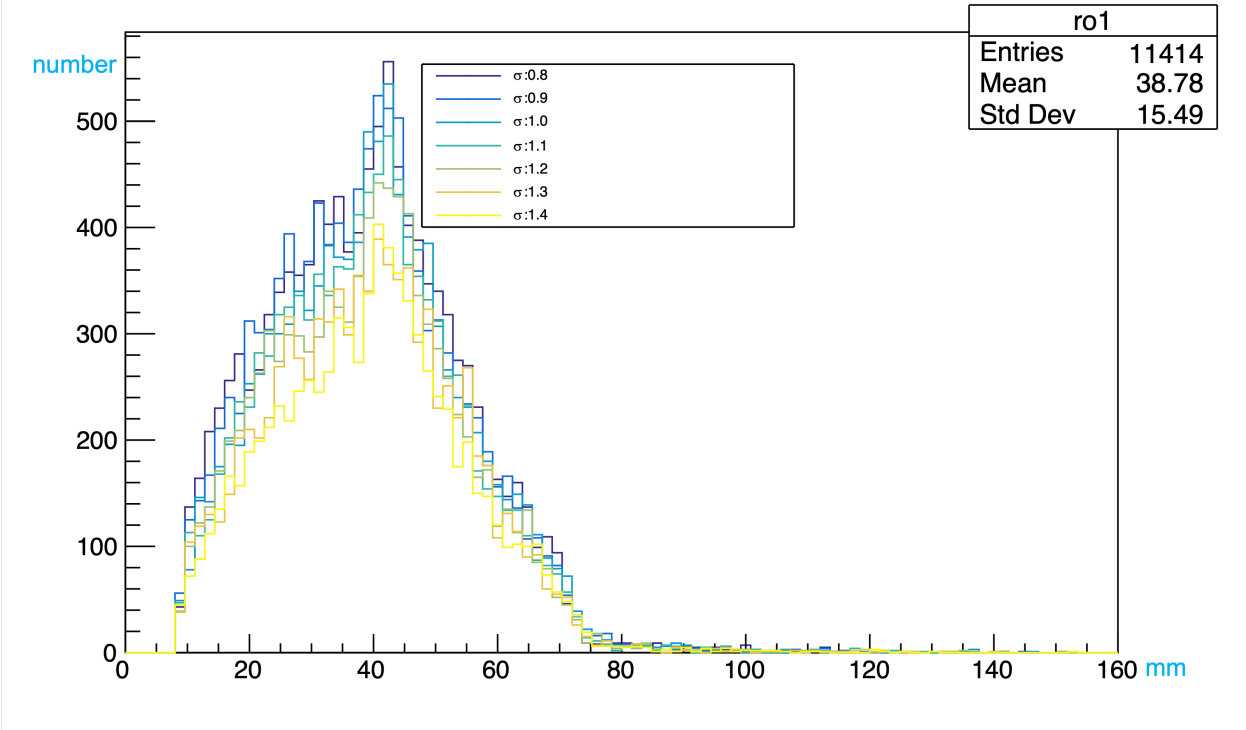

Figure 5.6: ρ plots for different times to nominal values of σ_y and nominal σ_x (1 bunch/beam size pair)

Meanwhile, the line segment are close and hard to distinguish with different values of σ_y in Fig 5.6. On the other hand, even though the same plots for different values of σ_y show some difference in some range, but it is not as drastic as the case of changing σ_x.

From the distribution as seen in Fig 5.3, 5.4,5.5,5.6, it can be seen that the number of particles is a major discriminant between different values of both horizontal and vertical beam size. Especially, the number of particles hit increases drastically when the values of σ_x is decreased. In case of decreasing values of σ_y, the number of hit increase slightly in most of the range in both ρ and ϕ distribution. The influence of the keyhole shape(as seen in Fig 4.4) is responsible for the asymmetry of the ϕ and ρ distribution.

5.2 EDA

Exploratory Data Analysis (EDA) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions. Current study in competition or application show that appropriate data analysis method will improve *quality* of data set that make it possible to increase accuracy of Machine Learning model.

5.2.1 Histogram Algorithm

In earlier research, the image from screen of Pair Monitor always be defined as pattern in training model. Although image recognition technique is popular and rapidly developmental in worldwide and the it is a visible method for expressing the hit distribution information on the detector, in this research it is a kind of unstructed data that is hard to intercomparison and do calculate with each other than digital data. Generally speaking, image store as RBG

matrix with 3 channel. Each pixel of image combination of color values with Red channel, Blue channel and Green channel that is tensor of n*m*3 (n*m is size of pixel). As seen at Fig 4.4, more deep color means more large number of hit particles. For more comfortable and direct performance, I used digit instead of color with 3 channel to displace the number of hit particle at any position of pixel.

Firstly, dividing the screen of detector into a form with 200*200 grid like Fig 5.7. Each element inside the grid is a digit. If once the MCparticle hit into any grid, the element in this grid plus one. Ultimately, a 200*200 matrix will be build, and the component of matrix means the number of hit particle which look likes a histogram of 3-dimension shown in Fig 5.8, the x-axis and y-axis are plane of screen in detector, and z-axis means the number axis of hit particles.

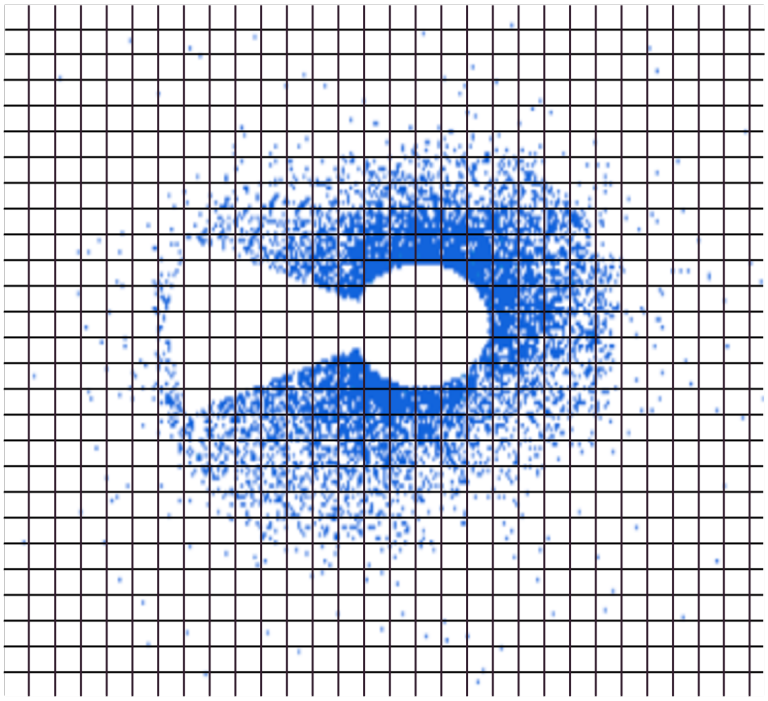

Figure 5.7: divide by 200*200 grid

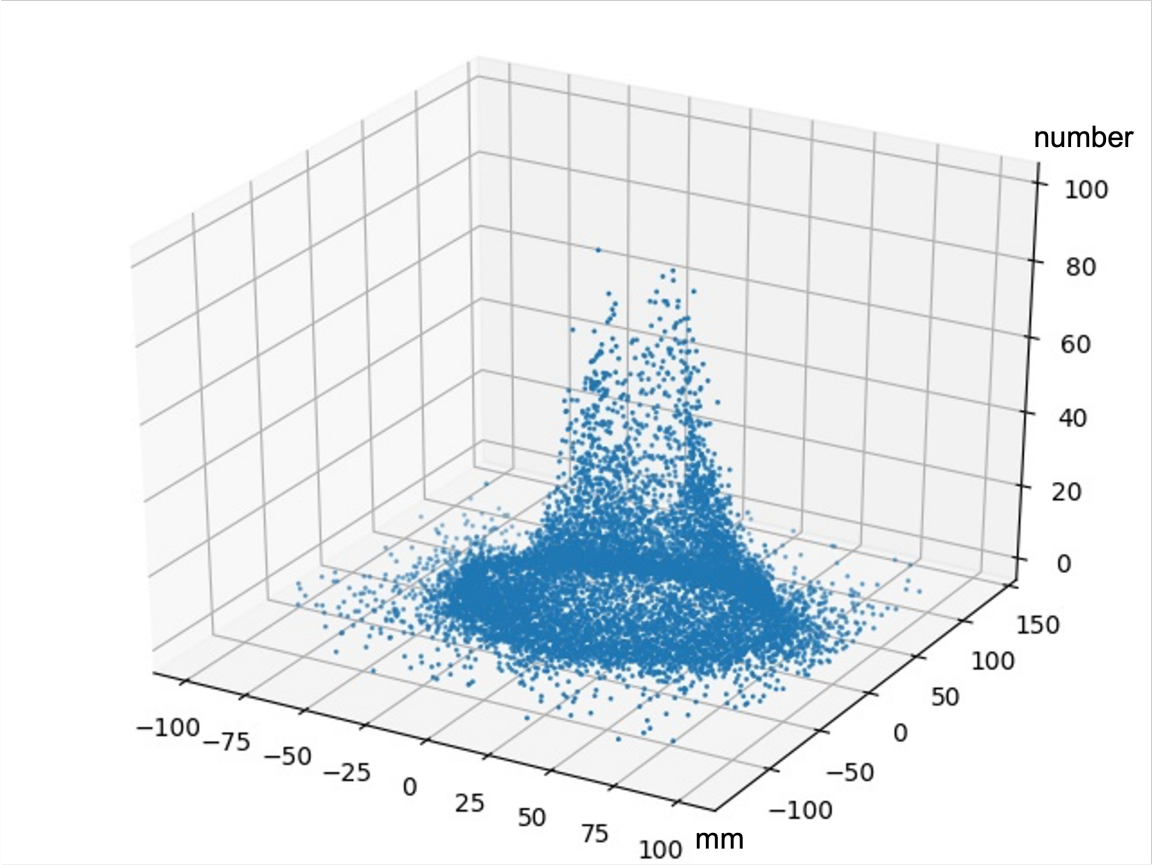

Figure 5.8: overhead viewing

5.3 High-Dimension Data

Though we transformed the pattern from image to matrix, but the dimensions of data are so huge that it is troublesome to find correlation and discrepancy among the different data.

What is dimension of data

We always describe an object from many different angles.If those angles describe an object completely, it could be regard as dimension of data. In my research, each matrix is a sample with label of beam size. It could be accepted similar sample with the same beam size and different sample with the different beam size respectively, but the common point of all samples is that those could be described by a 200*200 matrix. In that circumstances, the dimensionality of each sample could be seen as 200*200=400000, it also called *variables* of data set.

How to display a high-dimension data

For one dimensional data, it is easy to display it by a Number line. For two dimensional data, we could put it on a plane of two dimension. And for three dimensional data, it is available to see it at three dimensional space. But once the dimensionality exceed 3, the data are hard to visualize in our imagination. Before putting the data into model of Machine learning or Deep learning, it is required to make sure whether those sample could be distinguished by different label of beam size. Especially, due to huge dimensionality(40000), we could not get result directly. So first of all, try to make those data more easy to display.

5.4 Dimension Reduction

5.4.1 PCA

Principal Component Analysis(PCA), is a dimensionality-reduction method that often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set.

Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process.

So to sum up, reduce the number of variables of a data set, while preserving as much information as possible are our goal in this section.

Standardization

The first step is to standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis[14].

More specifically, the reason why it is critical to perform standardization prior to PCA, is that the latter is quite sensitive regarding the variances of the initial variables. That is, if there are large differences between the ranges of initial variables, those variables with larger ranges will dominate over those with small ranges (For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results. So, transforming the data to comparable scales can prevent this problem.

Mathematically, this is available by subtracting the mean and dividing by the standard deviation for each value of each variable.

$$z = \frac{value - mean}{standdevition} \tag{5.1}$$

Once the standardization is done, all the variables will be transformed to the same scale.

Covariance Matrix Computation

For understanding how the variables of the input data set are varying from the mean with respect to each other, or in other words, to see if there is any relationship between them. Because sometimes, variables are highly correlated in such a way that they contain redundant information. So, in order to identify these correlations, we compute the covariance matrix.

The covariance matrix is a p × p symmetric matrix (where p is the number of variables) that has as entries the covariances associated with all possible pairs of the initial variables. For example, for a 3-dimensional data set with 3 variables x, y, and z, the covariance matrix is a 3×3 matrix of this from:

$$\begin{bmatrix} Cov(x,x) & Cov(x,y) & Cov(x,z) \\ Cov(y,x) & Cov(y,y) & Cov(y,z) \\ Cov(z,x) & Cov(z,y) & Cov(z,z) \end{bmatrix}$$

Since the covariance of a variable with itself is its variance (Cov(a,a)=Var(a)), in the main diagonal (Top left to bottom right) we actually have the variances of each initial variable. And since the covariance is commutative (Cov(a,b)=Cov(b,a)), the entries of the covariance matrix are symmetric with respect to the main diagonal, which means that the upper and the lower triangular portions are equal.

If a n-dimension data after standardization:

$$X = (x_1, x_2, ..., x_n) \tag{5.2}$$

Convariance matrix:

$$C = \frac{1}{n} X X^T \tag{5.3}$$

If the convariance are positive, the two variables increase or decrease together i.e. correlate. If the covariance are negative, the one increase and the other decrease i.e. inversely correlated. We got the correlations between all the possible pairs of variables.

Principal Components[11]

Principal components are new variables that are constructed as linear combinations or mixtures of the initial variables. These combinations are done in such a way that the new variables (namely principal components) are uncorrelated and most of the information within the initial variables is squeezed or compressed into the first components. For example,a 10 dimensional data gives you 10 principal components, but PCA tries to put maximum possible information in the first component, then maximum remaining information in the second and so on.

In this research, 40000 variables of data are so huge that most of component are useless in describing the identified data.So organizing information

in principal components by this way, will allow us to reduce dimensionality without losing overfull information by discarding the components with low information and considering the remaining components as your new variables. An important thing to realize here is that, the principal components are less interpretable and don't have any real meaning since they are constructed as linear combinations of the initial variables.

Geometrically speaking, principal components represent the directions of the data that explain a maximal amount of variance, that is to say, the lines that capture most information of the data. The relationship between variance and information here, is that, the larger the variance carried by a line, the larger the dispersion of the data points along it, and the larger the dispersion along a line, the more the information it has. To put all this simply, just think of principal components as new axes that provide the best angle to see and evaluate the data, so that the differences between the observations are better visible.

So how to get the principal components? Eigenvectors and eigenvalues are the linear algebra concepts that we need to compute from the covariance matrix in order to determine the principal components of the data.Every eigenvector has an eigenvalue and their number is equal to the number of dimensions of the data. For example, for a 3-dimension data set, there are 3 variables, therefore there are 3 eigenvectors with 3 corresponding eigenvalues.

The first maximum principal component accounts for the largest possible variance in the data set. For example,to reduce dimension from 2 to 1.The scatter plot of our data set is as shown at Fig 5.9. The first principal component is approximately the line that minimize the average of the squared distances from the projected points to the line (variance).Comparing those two line, the u_1 is better than u_2 at minimizing variance. If the dimensionality larger than 2, the second principal component is calculated in the same way, with the condition that it is uncorrelated with (i.e., perpendicular to) the first principal component and that it accounts for the next highest variance.

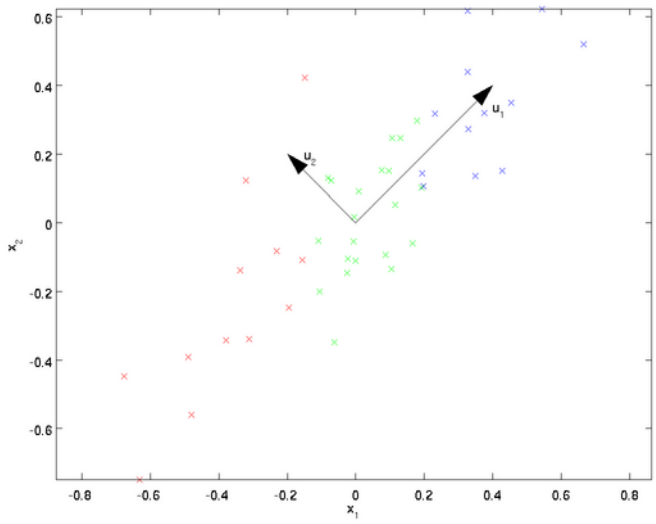

Figure 5.9: a case of data set

As describe above, the eigenvectors of the Covariance matrix are actually the directions of the axes where there is the most variance(most information) and that we call Principal Components. And eigenvalues are simply the coefficients attached to eigenvectors, which give the amount of variance carried in each Principal Component.By ranking your eigenvectors in order of their eigenvalues, highest to lowest, we get the principal components in order of significance.If we decided how many dimensionalities we want to keep, and discard those of lesser significance (of low eigenvalues), and form with the remaining ones a matrix of vectors that we call *Feature vector*.For example, if we choose to keep only p eigenvectors (components) out of n, the final data set will have only p dimensions that achieve goal of reducing dimension.

Don't forget to recover the data from standardization to original range by:

$$FinalDataSet = FeatureVector^T * StandardizedOriginalDataSet^T$$

5.4.2 T-SNE

In this research, although we can give up part of variable with low information and reduce the complexity of compute at the same time, around 5000

dimensionality still be required to completely describe the data. It is still hard to look for the correlation among the data set. How to display those data set by 2-dimensional plane or 3-dimensional space? There are a method called T-SNE dimension reduction, compare with the PCA reduction described above, what difference between them? PCA is a linear dimension reduction technique that seeks to maximize variance and preserves large pairwise distances,and T-SNE differs from PCA by preserving only small pairwise distances or local similarities[6].

The T-SNE algorithm calculates a similarity measure between pairs of instances in the high dimensional space and in the low dimensional space. It then tries to optimize these two similarity measures using a cost function.

Similarities

The main work are measuring similarities between points in the high dimensional space. Think of a bunch of data points scattered on a 2D space like Fig 5.10.

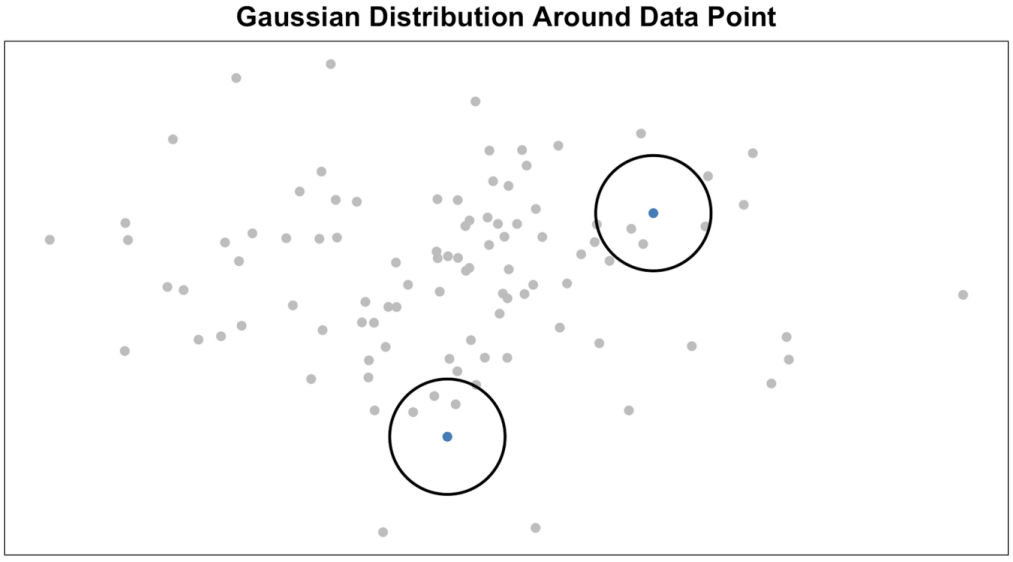

Figure 5.10: Measuring pairwise similarities in the high-dimensional space

For each data point (x_i) we'll center a Gaussian distribution over that point. Then we measure the density of all points (x_j) under that Gaussian distribution. Then renormalize for all points. This gives us a set of probabilities

$(P_{ij})$for all points. For $i \neq j$,define:

$$P_{i|j} = \frac{exp(-\left\|x_i - x_j\right\|^2 2\sigma_i^2)}{\sum_{k \neq j} exp(-\left\|x_i - x_k\right\|^2 2\sigma_i^2)} \tag{5.4}$$

if we set $P_{i|i} = 0$, then :

$$\sum_j P_{j|i} = 1 \tag{5.5}$$

Now,denfine:

$$P_{ij} = \frac{P_{j|i} + P_{i|j}}{2N} \tag{5.6}$$

And note that $p_{ij} = p_{ji}$,$p_{ii} = 0$ and $\sum_{i,j} P_{ij} = 1$

Those probabilities are proportional to the similarities. All that means is, if data points x_1 and x_2 have equal values under this gaussian circle then their proportions and similarities are equal and hence you have local similarities in the structure of this high-dimensional space. The Gaussian distribution or circle can be manipulated using what's called perplexity, which influences the variance of the distribution (circle size) and essentially the number of nearest neighbors. Normal range for perplexity is between 5 and 50[27].

Student T-distribution

Instead of Gaussian distribution, the Student T-distribution,which is also know as the Cauchy distribution gives us a second set of probabilities(Q_{ij}) at the low dimensional space.Q_{ij} is used to measure similarities between low-dimensional points in order to allow dissimilar objects to be modeled far apart in the map. As you can see the Student t-distribution has heavier tails than the normal distribution. The heavy tails allow for better modeling of far apart distances.

For $i \neq j$, define Q_{ij}:

$$Q_{ij} = \frac{(1 + - \left\|y_i - y_j\right\|^2)^-1}{\sum_k \sum_{l \neq k} (1 + \left\|y_k - y_l\right\|^2)^-1} \tag{5.7}$$

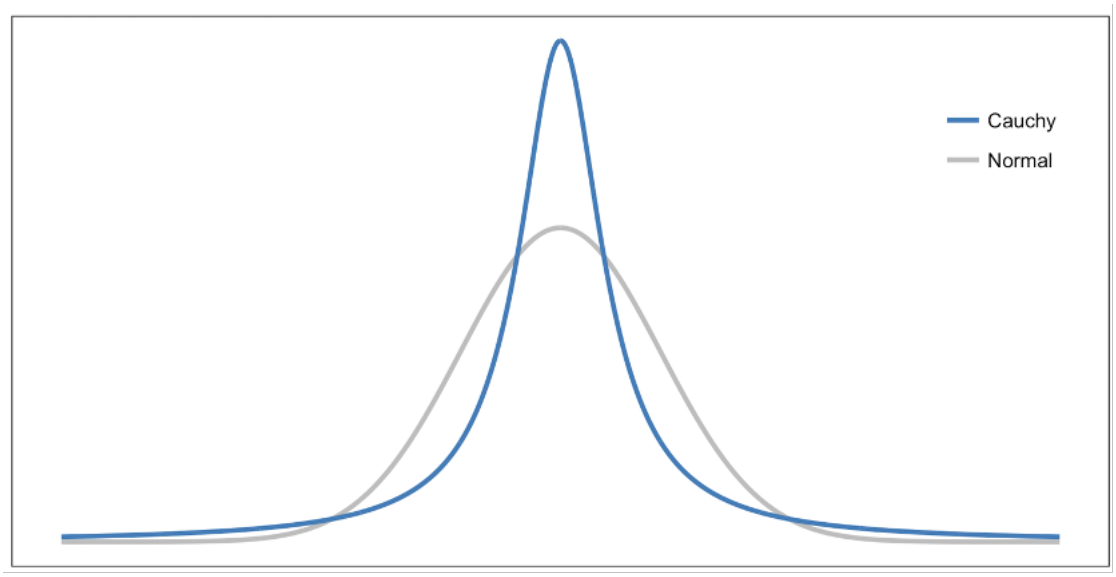

Figure 5.11: Gaussian vs Student t-distribution[28]

KL divergence

Finally we want these set of probabilities from the low-dimensional space (Q_{ij}) to reflect those of the high dimensional space (P_{ij}) as best as possible.We want the two map structures to be similar. We measure the difference between the probability distributions of the two-dimensional spaces using Kullback-Liebler divergence (KL).

$$KL(P\|Q) = \sum_{i \neq j} P_{ij} log(\frac{P_{ij}}{Q_{ij}}) \tag{5.8}$$

KL divergence is a statistical distance: a measure of how one probability distribution P is different from a distribution Q.The locations of the points y_i in low dimension space are determined by minimizing the KL divergence of the distribution P and distribution Q. The method of minimizing divergence are gradient descent, which are always utilized in Machine Learning model.

5.4.3 Visualization of data

In machine learning, a large number of data set are necessary to train a appropriate model. The result of simulation by technique described at section 4 shown in table 5.1:

σ_x \ σy	0.8	0.9	1.0	1.1	1.2	1.3	1.4
0.8	1088	1087	1079	1068	1077	1069	1099
0.9	1093	1039	1066	1072	1085	951	1092
1.0	1081	1041	1073	1096	1071	1063	1078
1.1	1077	1071	1085	1042	1072	1078	1050
1.2	1049	1010	1062	1046	1085	1082	1001
1.3	1032	1024	1002	1013	1001	1011	1026
1.4	1063	979	953	1045	921	1025	1098

Table 5.1: the number of the sample in different parameter by simulation

There are total around 50000 samples as data set.

It is also important to estimate whether those data set are distinguishable and rhythmic by using the algorithm that discussed above. Select 200 samples from each pair of σ_x and σ_y at visualization of data. We used T-SNE by package of *sklearn.manifold.TSNE* and PCA by package of *sklearn.decomposition.PCA* .

Firstly, we try to reduce the dimension from 200*200=40000 to 2-D space directly by T-SNE at range of σ_x and σ_y respectively. The result at Fig 5.12,5.13.(In color bar, 0-7 map to the multiple to nominal value of σ_x or σ_y from 0.8-1.4, the following is same.)

The result of figures are vague and disordered due to too many original dimension. So before using T-SNE, reduce part of variable from 200*200=40000 to 5000 by linear reduction namely PCA, the result shown in Fig 5.14,5.15,5.16. By the way, the axis in figures show the abstract distance between the points, so the unit of axis are common.

5.4.4 Conclusion

From the distributions as seen in Fig 5.14,5.15, it can be seen that the distance among the data of different σ_x are larger than the data of different σ_y, which accord with the result from section 5.1. Beside, Fig 5.16 show total 49 data distribution with different pair of σ_x and σ_y at high-dimension.It is obvious that the center of cluster are isolate, but there are many overlap area

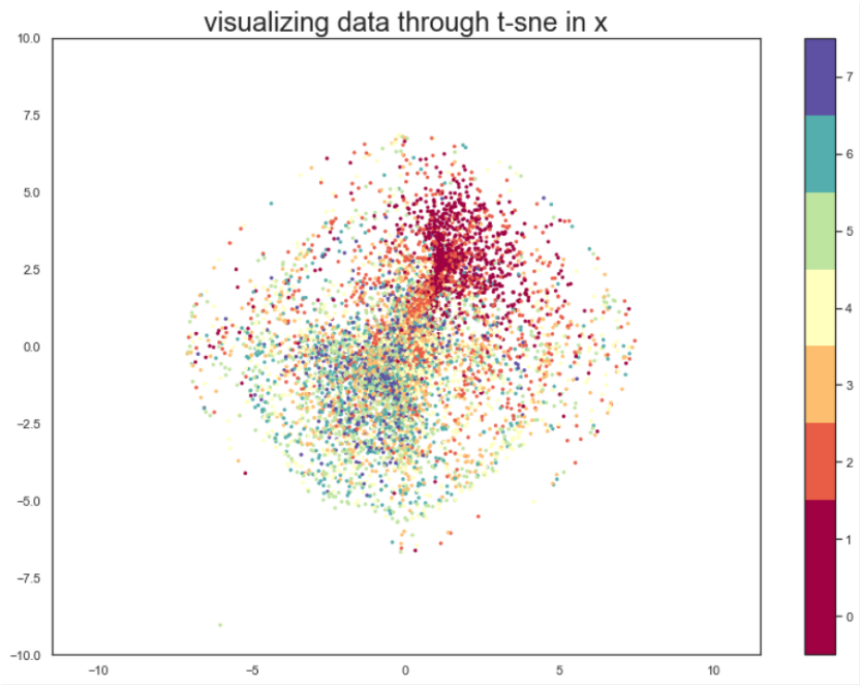

Figure 5.12: the data of different σ_x distribution at high-dimension

among different clusters. Even the data with the same beam size, it also has high statistical fluctuations under the result of each bunch.

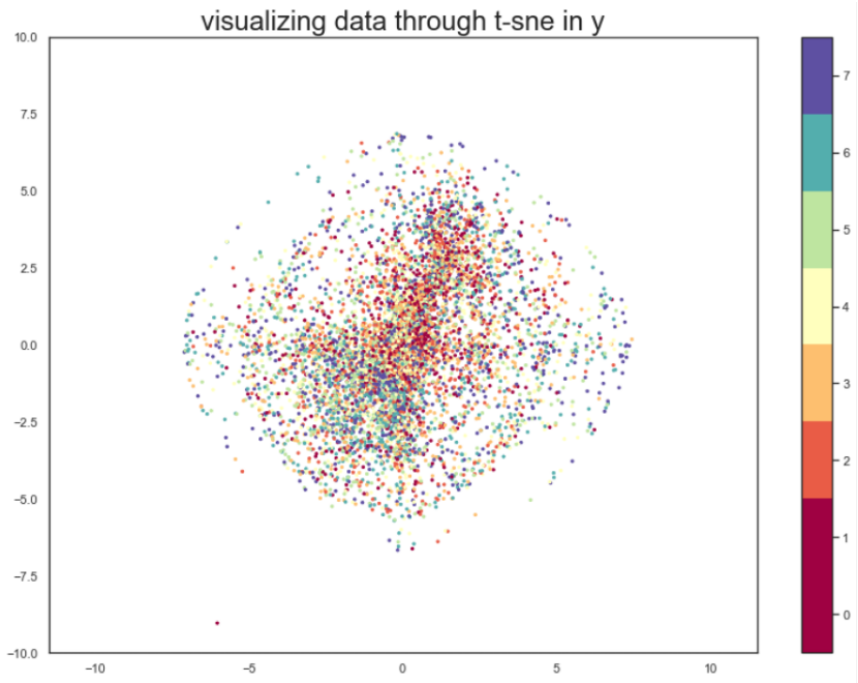

Figure 5.13: the data of different σ_x distribution at high-dimension

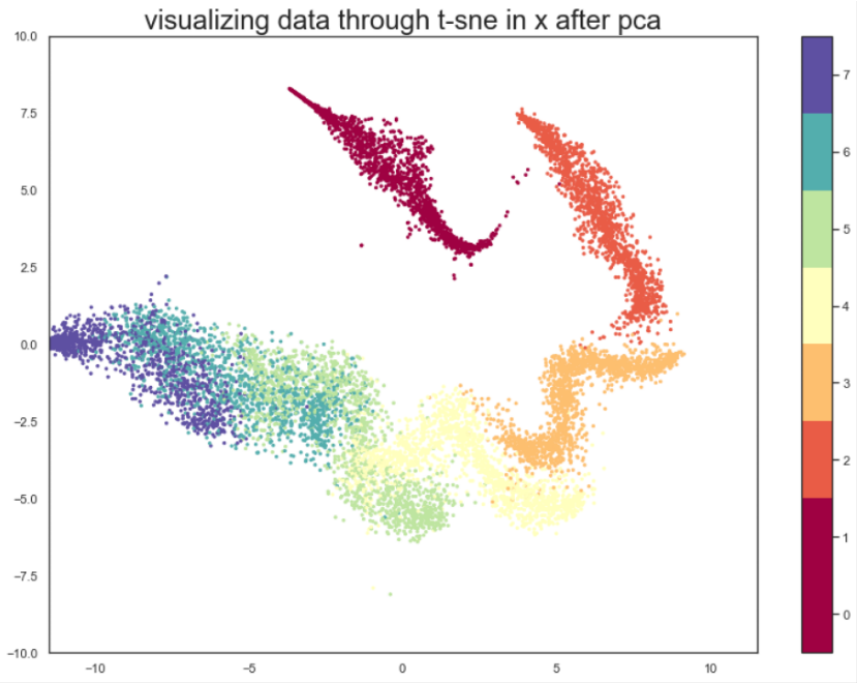

Figure 5.14: the data of different σ_x distribution at high-dimension after PCA

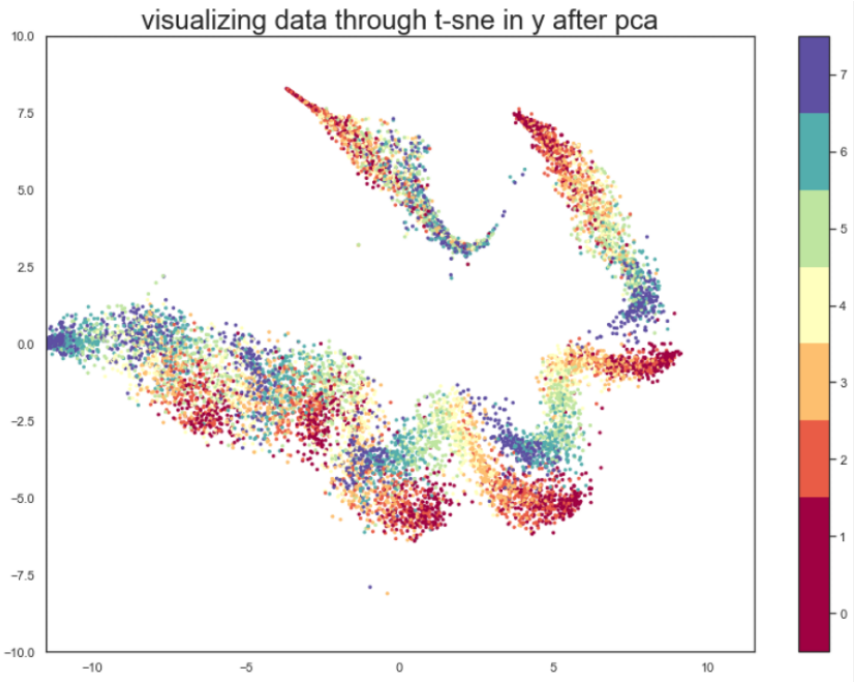

Figure 5.15: the data of different σ_y distribution at high-dimension after PCA

Figure 5.16: the data of different pair of σ_y and σ_y distribution at high-dimension after PCA

Chapter 6

Machine Learning

6.1 Introduce

Machine learning is an intersection of computer science and statistics and its study involves the design of algorithms in order to draw statistical inferences from observation.

In the language of machine learning, any observation can be considered to be an unknown function $y = h(x)$ where x denotes the independent variable. The purpose of machine learning is to define a hypothesis H which contains a set of many functions, and to choose a certain function h from H, so that $h \approx H$ in a strictly mathematical sense.

After many years of development, there are a great lot of excellent model in Machine Learning include *Logistic Regression*, *SVM*, *Bayesian Model* and so on. Otherwise, the advancement of Machine learning that Deep Learning also play a important role at many applications. I'd like to introduce two kinds of models employed in this research,*Neural Network* and *Ensemble Learning*, at next section.

Machine learning can be categorized into supervised learning, unsupervised learning, reinforcement learning etc. Machine learning also can be categorized into Regression and classification. In this research, supervised learning is suitable to address problem.

Regression

Regression analysis consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x).

Classification

Classification is a task of Machine Learning which assigns a label value to a specific class and then can identify a particular type to be of one kind or another.

In this research, the label of training data, which shown at table 5.1, look like continuous values but discrete values of 49 kinds of classes actually. So Classification Model are suitable model in this research.

6.2 Neural Network

Neural Networks are the functional unit of Deep Learning and are known to mimic the behavior of the human brain to solve complex data-driven problems. The input data is processed through different layers of artificial neurons stacked together to produce the desired output.

6.2.1 Neurons

The Neural Network architecture is made of individual units called *neurons* that mimic the biological behavior of the brain which shown in Fig 6.1.

Input

It is the set of features that are fed into the model for the learning process. For example, in this research, input values are variables in sample.

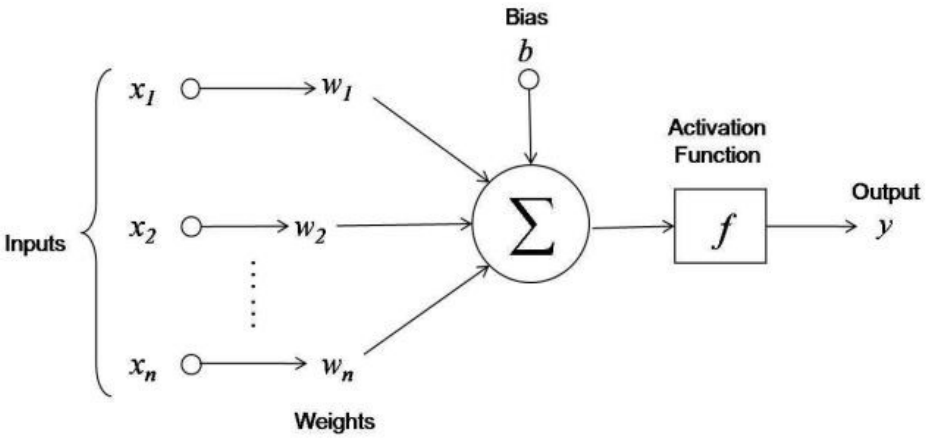

Figure 6.1: Neuron in Artificial Neural Network

Weight

It is main function is to give importance to those features that contribute more towards the learning. It does so by introducing scalar multiplication between the input value and the weight matrix.

Transfer Function

The job of the transfer function is to combine multiple inputs into one output value so that the activation function can be applied. It is done by a simple summation of all the inputs to the transfer function.

Activation Function

It introduces non-linearity in the working of perceptrons to consider varying linearity with the inputs. Without this, the output would just be a linear combination of input values and would not be able to introduce non-linearity in the network.

Bias

The role of bias is to shift the value produced by the activation function. Its role is similar to the role of a constant in a linear function.

The calculation function inside the neuron are:

$$y_k = f(\sum_{m=1}^{M}(x_m * w_{km} + b_k)) \tag{6.1}$$

When multiple neurons are stacked together in a row, they constitute a layer, and multiple layers piled next to each other are called a multi-layer neural network(Fig 6.2).

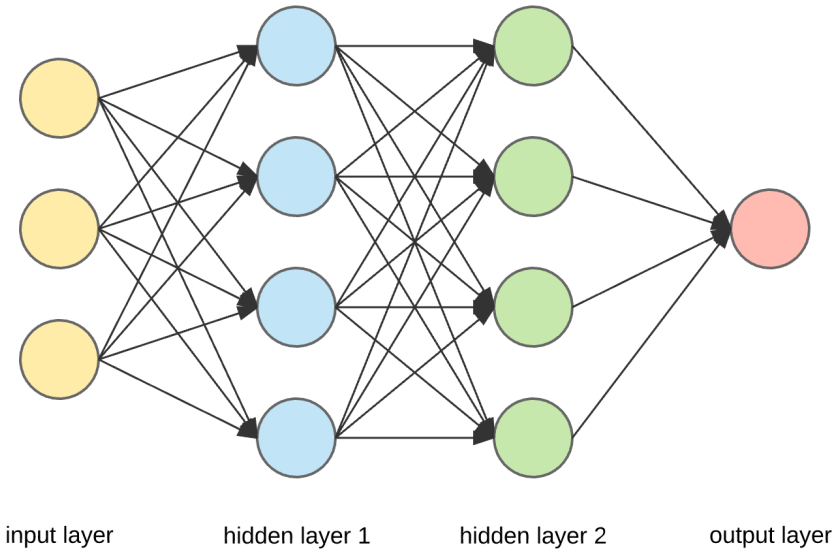

Figure 6.2: Multi-layer neural network

6.2.2 Layer

Input Layer

The data that we feed to the model is loaded into the input layer from external sources like a CSV file or a web service. It is the only visible layer in the complete Neural Network architecture that passes the complete information from the outside world without any computation.

Hidden Layers

The hidden layers are core of Multi-layer neural network what makes deep learning. They are intermediate layers that do all the computations and ex-

tract the features from the data. There can be multiple interconnected hidden layers that account for searching different hidden features in the data.

Output Layer

The output layer takes input from preceding hidden layers and comes to a final prediction based on the model's learnings. It is the most important layer where we get the final result. In the case of classification/regression models, the output layer generally has a single node. However, it is completely problem-specific and dependent on the way the model was built.

6.2.3 Forward Propagation

Neural network contains one or multiple layers, with each layer containing one or multiple nodes. Each node corresponds to a certain Activation Function φ_j, that is input into the next layer. If the first layer having M nodes, is M linear combinations of the input variables x_1,...,x_N:

$$a_j = \sum_{i=1}^{n} w_{ji}^{(1)} + w_{j0}^{(1)} \tag{6.2}$$

where j=1,...,n and the superscript (1) denotes the first layer, $w_{ji}^{(1)}$ denote the weights and $w_{j0}^{(1)}$ denotes the bias. And the bias can be absorbed into the weights by taking an input $x_0 = 1$ so that,

$$a_j = \sum_{i=0}^{N} w_{ij}^{(1)} x_i \tag{6.3}$$

And a non-linearity known as activation function is applied to this output,

$$z_j = \varphi(a_j) \tag{6.4}$$

Input layer output and hidden layer output can be combined to give a result at the output layer. For classification problem, $softmax$ usually be set as the

activated function of the output layer.

$$\sigma(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{6.5}$$

The i denote the number of neuron at output layer, is also the number of classes we want to distinguish. And the x_i is output values of output layer. Through double hidden layers and output layers, we get,

$$y_k(x, w) = \sigma(\sum_{j=0}^{M} w_{kj}^{(2)} \varphi(\sum_{i=0}^{D} w_{ji}^{(1)} x_i)) \tag{6.6}$$

The above method of calculating the output is called forward propagation. And weight and bias can be update by back propagation.

6.2.4 Back Propagation

The output obtained in this way is evaluated by means of a cost function, $E(w)$ which in the current study is the mean square error, which is defined as the following:

$$E(w) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i(w))^2 \tag{6.7}$$

for an input (x_i,y_i),where $x_i \in \mathbb{R}^{d+1}$ and prediction $\hat{y}_i(w)$ The cost function also be minimized by means of gradient descent, and the main idea of gradient descent is to update the parameters and then evaluate the cost function iteratively.

$$v_t = \eta_t \nabla_w E(w_t) \tag{6.8}$$

$$w_{t+1} = w_t - v_t \tag{6.9}$$

where $\nabla_w E(w_t)$ is the gradient of $E(w)$ with respect to w, ∇_t is the learning rate which controls the step to be taken in the direction of gradient at a step t. The model approach to ideal function that $h(x) \approx H$ by multiple iteration of Eq 6.8 and 6.9, namely *gradient descent*. But the presence of

multiple layers causes the error function to be a composite function of all the weight parameters used in the earlier layers. Because of this brute force calculation of *gradient descent* is not feasible in this case. Instead a special algorithm known as *back propagation* is used.

Change of the cost function with respect to the weighted input in a layer, i.e. the gradient $\Delta_j^{(l)}$ corresponding to the j-th neuron in the l-th layer can be defined as,

$$\Delta_j^{(l)} = \frac{\partial E}{\partial a_j^{(l)}} \frac{\partial \sigma(z_j^{(l)})}{\partial z_j^{(l)}} \tag{6.10}$$

Since error in layer l is propagated from the subsequent layer $l+1$, chain rule of differentiation can be used to write,

$$\Delta_j^{(l)} = (\sum_k \Delta_k^{(l+1)} w_{kj}^{(l+1)}) \frac{\partial \sigma(z_j^l)}{\partial z_j^{(l)}} \tag{6.11}$$

When final error is differentiated by the weight of the k-th neuron of a specific layer l, the result can be expressed by means of the product of $\Delta_j^{(l)}$ from above and j the output of the k-th from the previous layer, $l-1$. In such way the derivative of E for weights at all layers and neurons can be computed.

$$\frac{E}{w_{jk}^{(l)}} = \frac{E}{w_{jk}^{(l)}} = \Delta_j^{(l)} a_k^{(l-1)} \tag{6.12}$$

This approach makes it possible to update the weights:

$$w^{l+1} := w^l - \eta \frac{E}{w^{(l)}} \tag{6.13}$$

where the elements of the vector $w(l)$ correspond to all the weights in the neurons of a particular layer l and η denotes the learning parameter.

6.3 Ensemble learning

6.3.1 Introduce

Ensemble learning is a general meta approach to machine learning that seeks better predictive performance by combining the predictions from multiple models.

There are three main classes of ensemble learning methods are bagging, stacking, and boosting. In this research, a kind of boosting ensemble model be selected because of it's outstanding performance and accurate in many competition and experiments that is called Gradient Boosting Decision Trees(GDBT)

6.3.2 Decision Trees

A decision tree is a machine learning model that builds upon iteratively asking questions to partition data and reach a solution. It is the most intuitive way to zero in on a classification or label for an object. Visually too, it resembles and upside down tree with protruding branches and hence the name.

A decision tree is a flowchart-like tree structure where each node is used to denote feature of the data set, each branch is used to denote a decision, and each leaf node is used to denote the outcome.

The topmost node in a decision tree is known as the root node. It learns to partition on the basis of the feature value. It partitions the tree in a recursive manner, also call recursive partitioning. This flowchart-like structure shown in Fig 6.3 helps in decision making. Like Neural Network, Decision Tree also have a cost function(error function) to evaluate performance of model and update the model by minimizing the function. In current study, Gini impurity is a good cost function in Decision Tree.

For a data set $S = (\boldsymbol{x_1}, y_1), ..., (\boldsymbol{x_n}, y_n)$, $\boldsymbol{x_n} \in \mathbb{R}^{d+1}$ and $y_i \in 1, ..., c$ where c is the number of classes, the probability of picking up a certain label k can be denoted as $P_k = \frac{|S_K|}{s}$, where $S_K \in S$,$S_k = (\boldsymbol{x}, y) : y = k$ and $S = S_1 \cup ... \cup S_2$

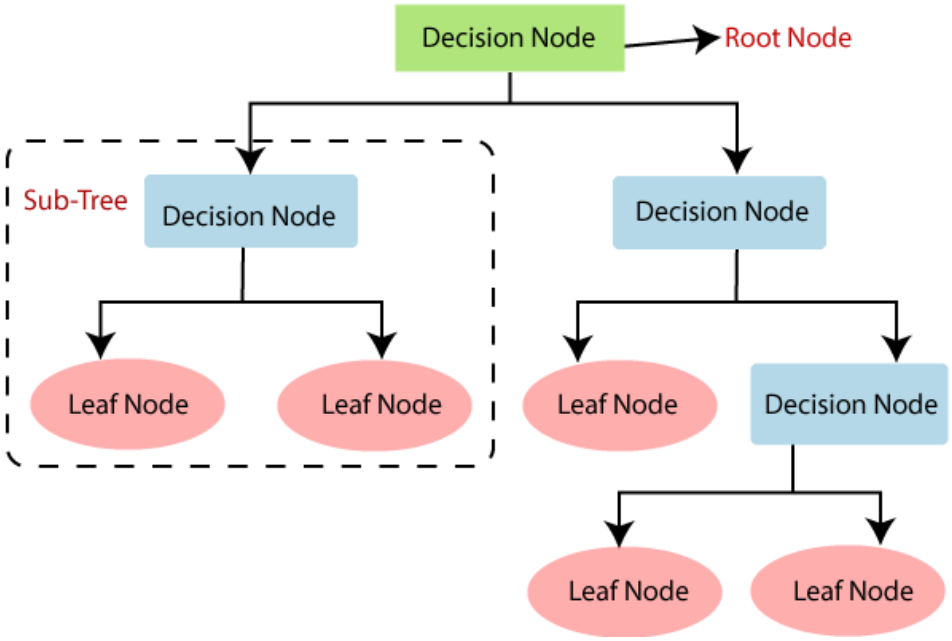

Figure 6.3: a framework of decision tree

Gini inpurity of a leaf, $G(S)$ is defined as:

$$G(S) = \sum_{k=1}^{c} p_k(1 - p_k) \tag{6.14}$$

when one leave is pure i.e. $p_k = 1$, the corresponding Gini impurity is 0. If a tree has two leaves, the maximum impurity corresponding to one leaf can therefore be 0.5.

An alternative to Gini impurity is cross-entropy, which for a leaf, can be defined as:

$$D = -\sum_{k=1}^{c} p_k log p_k \tag{6.15}$$

If p_k is 0 either 1, D will be 0. That is, entropy will be minimum if the leaf includes only one class.

6.3.3 Gradient Boosting Decision Trees

In gradient boosting decision trees, we combine many weak learners to come up with one strong learner. The weak learners here are the individual decision trees. For a data set $D = (\boldsymbol{x_i}, y_i)$, use K additive weak learners to combine a

strong learner:

$$\hat{y_i} = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{6.16}$$

where $F = f(x) = \omega(q(x))(q : \mathbb{R}^m \rightarrow T, \omega \in \mathbb{R}^T)$ Here q represents the structure of each tree that maps an example to the corresponding leaf index. T is the number of leaves in the tree. Each f_k corresponds to an independent tree structure q and leaf weights ω.

To learn the set of functions used in the model, we minimize the following loss objective function.

$$L(\phi) = \sum_{i} l(\hat{y_i}, y_i) + \sum_{k} \Omega(f_k) \tag{6.17}$$

where

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda ||\omega||^2 \tag{6.18}$$

Here l is a differentiable convex loss function that measures the difference between the prediction $\hat{y_i}$ and the target y_i. The second term Ω penalizes the complexity of the model,The additional regularization term helps to smooth the final learnt weights to avoid over-fitting.

The tree ensemble model in Eq 6.17 includes functions as parameters and cannot be optimized using traditional optimization methods. Instead, the model is trained in an additive manner. Formally, let $y_i^{(t)}$ be the prediction of the i-th instance at the t-th iteration, f_t needs to be added to minimize the following loss objective function.

$$L^{(t)} = \sum_{i=1}^{m} l(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \sum_{k} \Omega(f_k) \tag{6.19}$$

According to Taylor Series Expansion, Eq 6.19 can be write,

$$L^{(t)} \approx [\sum_{i=1}^{m} l(y_i, \hat{y_i}^{(t-1)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \sum_{k} \Omega(f_k) \tag{6.20}$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l$ are first and second order gradient statistics on the loss function. Then by removing the constant terms (constant terms do not effect gradient of objective function) and expand Ω as Eq 6.18, define $I_j = i|q(x_i) = j$.

$$\tilde{L}^{(t)} \approx \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)\omega_j + \frac{1}{2}(\sum_{i\in I_j)} h_i + \lambda)\omega_j^2] + \gamma T \tag{6.21}$$

Define $G_j = \sum_{i\in I_j} g_i$ and $H_j = \sum_{i\in I_j} h_i$,

$$\tilde{L}^{(t)} = \sum_{j=1}^{T}[G_i\omega_i + \frac{1}{2}(H_i + \lambda)\omega_j^2] + \gamma T \tag{6.22}$$

For a fixed structure $q(x)$, we can compute the optimal weight w_j^* of leaf j by

$$w_j^* = -\frac{G_j}{H_j + \lambda} \tag{6.23}$$

And calculate the corresponding optimal value by

$$\tilde{L}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T \tag{6.24}$$

Eq 6.24 can be used as a scoring function to measure the quality of a tree structure q. This score is like the impurity score for evaluating decision trees, except that it is derived for a wider range of objective functions.

Normally it is impossible to enumerate all the possible tree structures q. A greedy algorithm that starts from a single leaf and iteratively adds branches to the tree is used instead. Assume that IL and IR are the instance sets of left and right nodes after a split. Letting $I = I_L \cup I_R$, then the loss reduction after the split is given by

$$Gain = \frac{1}{2}[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} + \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}] - \gamma \tag{6.25}$$

To the sum, the process of GDBT are:

1.The algorithm produce new single decision tree each iteration.

2.Before starting iteration, calculate first and second order gradient statistics of loss function at every sample of training data.

3.The new decision tree produced by greedy algorithm, compute predictive values at every leaf by Eq 6.23.

4.Add the new decision tree into model.

5.Finish all of iteration, the model is strong learner what we need.

6.4 Application

6.4.1 Preprocessing

The data set display at table 5.1 which are 200*200 matrix that extracted from result of simulation.The total number of data set are around 50000. But before fed the data set into Machine learning model, the preprocess of data is important to reduce the compute complexity and risk of overfit, besides it can increase the accuracy of model.

dimension reduction

As we said at section 5.4, the dimension of single data are too large to training, so we do dimension reduction by means of PCA technique.

label

As supervised learning, label of data is a important component to measure the accuracy of model. The label of our data set are beam size that are σ_x and σ_y.

Standardization

As described at section 5.4, it transform values of data such that the mean of the values is 0 and the standard deviation is 1. In this approach, we

are constraining our data attribute to a particular container to develop a correlation among different data points. And what's interesting is that not all of model need standardization, for example, GBDT (said in section 6.3.3) can not do data standardization.

train-test split

Splitting 80% of data set as train data, and 20% as test data. Because even if the model perform well at train data due to overfit, other normal data performance would be shown at test data.

6.4.2 Training and Test

Neural Network

Building a classification model of full connection neural network, In table

Layer	Number of nodes
Input Layer	5000
Hidden Layer1	1024
Hidden Layer2	256
Hidden Layer3	64
Output Layer	49

Table 6.1: summary of full connection neural network

6.1, the number of nodes at input layer come from the variables of data after PCA reduction, and the Output Layer are classes number of different pair of σ_x and σ_y.

The activate function is $ReLU$:

$$f(x) = max(0, x) \tag{6.26}$$

The output layer activate function is $softmax$(described at Eq 6.5)

The loss function is $Cross\ Entropy$

$$Loss = -\frac{1}{N}\sum_{i}\sum_{c=1}^{M} y_{ic} log(p_{ic}) \tag{6.27}$$

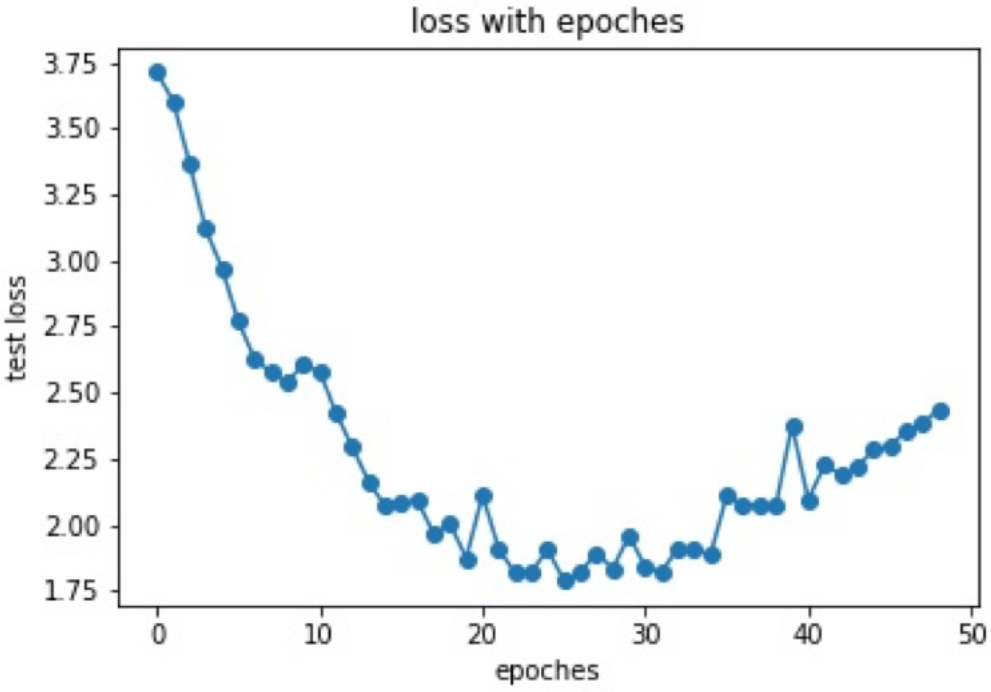

Figure 6.4: error of training data

where M is number of classes, $y_{ic} = 1$ if sample i belong to class c, or else $y_{ic} = 0$. p_{ic} is probability of sample i belong to class c namely the values from output layer.

Learning rate is 0.01 which means the speed of weight update. Method of training is *SGD* (stochastic gradient descent), that is stochastic training a part of data in every iteration. And one *epoch* means training the model by means of all the training data for one iteration. The number of epoch set as 50 that is 50 iteration. The result of training show at figure 6.4 and 6.5:

Ensemble Learning

The model used in this research called *XGBoost* which based on GBDT algorithm. Basic learner is *decision tree* which loss function also is *Cross Entropy*

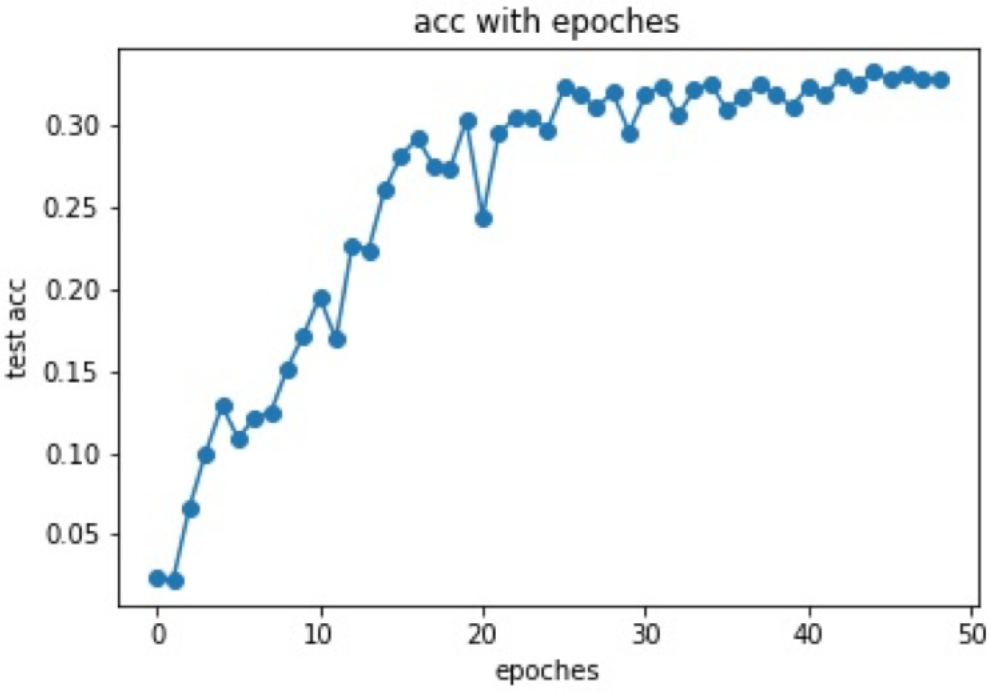

Figure 6.5: accuracy of test data

write at Eq 6.27. The parameter of XGBoost model set as following:

$$\begin{aligned} model = XGBClassifier(&max\_depth = 15,\\ &learning\_rate = 0.1,\\ &n\_estimators = 2000,\\ &objective =' multi : softprob',\\ &eval\_metric =' auc',\\ &reg\_alpha = 0,\\ &reg\_lambda = 0.4) \end{aligned}$$

Where max_depth denote the maximum depth of single decision tree, learning_rate is speed of iteration such as in Neural Network,n_estimators denote the number of weak learner(decision tree), objective function is $softprob$ that output a matrix of $n\_data * n\_class$ which values are probability between each data and each class, eval_metric is the Evaluation Metric for model, reg_alpha and reg_lambda is $L1regularization$ term and $L2regularization$

term. The accuracy define as :

$$acc = \frac{number\ of\ correctly\ classified\ sample}{total\ number\ of\ sample}$$

After training, the accuracy of test data is:

$$acc = 22.79\%$$

6.4.3 Resample

Performance of model are not good enough to our expect.Apart from modifying the model, we also can tune the data set.

In statistics and machine learning, the bias–variance tradeoff which shown in Fig 6.6 is the property of a model that the variance of the parameter estimated across samples can be reduced by increasing the bias in the estimated parameters.

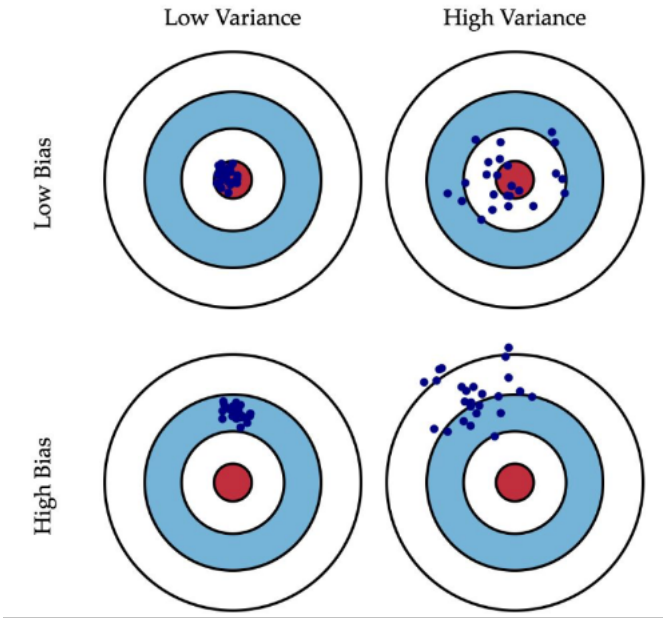

Figure 6.6: bias vs variance[24]

Bias

Bias is the difference between the average prediction of our model and the correct value which we are trying to predict. Model with high bias pays very little attention to the training data and oversimplifies the model. It always leads to high error on training and test data.

Varience

Variance is the variability of model prediction for a given data point or a value which tells us spread of our data. Model with high variance pays a lot of attention to training data and does not generalize on the data which it hasn't seen before. As a result, such models perform very well on training data but has high error rates on test data.

From Fig 5.14,5.15,5.5, we get result, that is the center of cluster are isolate, but there are many overlap area among different clusters. It's the expression of high bias.So if we can deal with the high statistical fluctuations of data, the accuracy of prediction will dramatic rise.

What I want are data superposition of the number of bunch for eliminating the effect of fluctuations.It easy to do matrix addition for data set which means recording hit information after the number of bunch.

The conduct at data set is randomly chose 20 samples at each label of beam size and do matrix addition as a new sample, then repeat it 200 times that means we have a new data set with 200 samples.

6.4.4 Conclusion

After T-SNE dimension reduction and Visulization, the result show in Fig 6.7,6.8 . It's obvious that the distinguishability at high dimensional space of new data set is better than origonal data set.

There are three series of charts which are original data after PCA, resample data without PCA and resample data after PCA respectively show

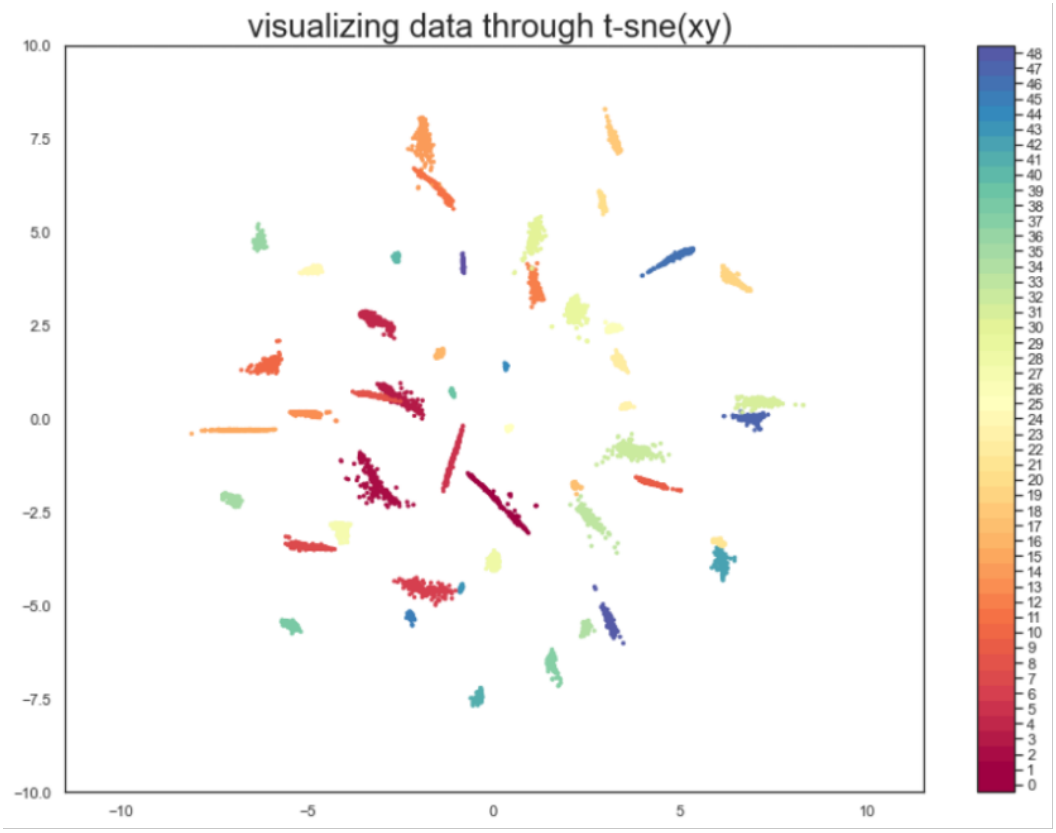

Figure 6.7: the new data of different pair of σ_x and σ_y distribution at high-dimension

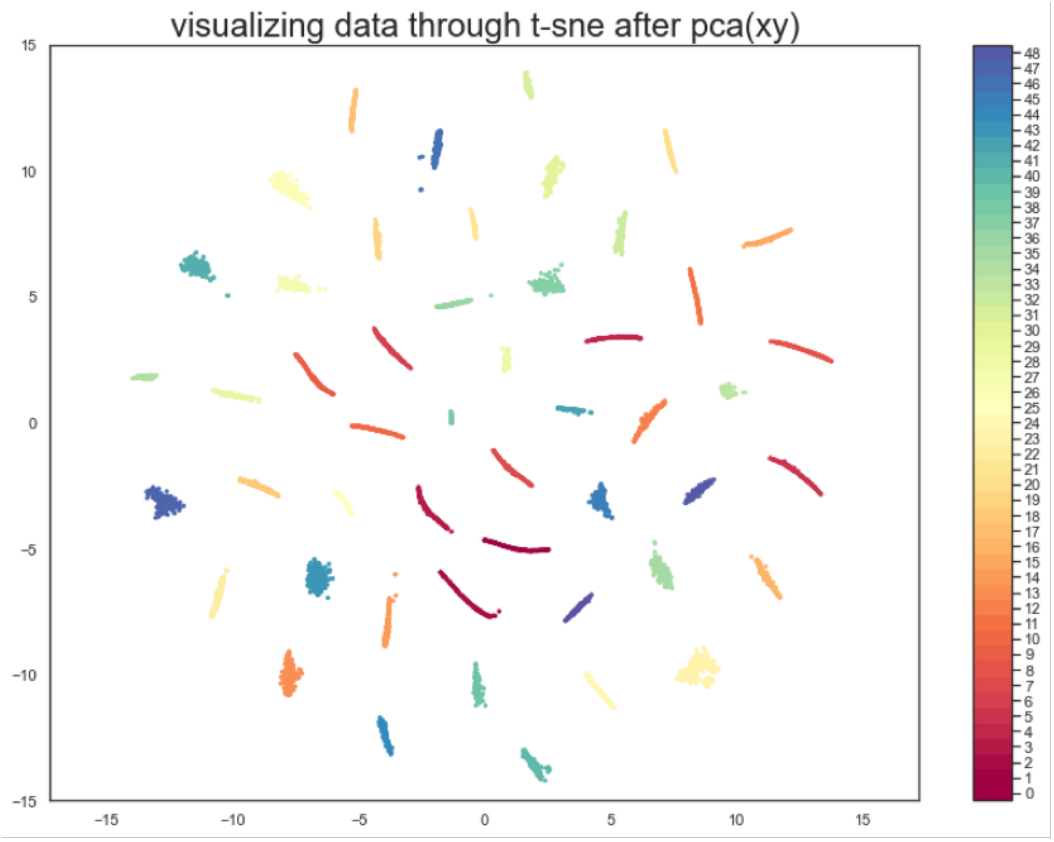

Figure 6.8: the new data of different pair of σ_x and σ_y distribution at high-dimension after PCA

in Fig 6.9,6.10,6.11 . From those charts we can learn that PCA could enhance the stability of model because it remove trifling dimensionality from input data, and resample could improve the accuracy of model as a result of decreasing statistical fluctuations of original data. By means of those two skills of data analysis, we could get a high precision and high stability model which could perfectly figure out the beam size of target sample in specify range σ_x(0.8,0.9,1.0,1.1,1.2,1.3 and 1.4 times σ_x nominal values) and σ_y(0.8,0.9,1.0,1.1,1.2,1.3 and 1.4 times σ_y nominal values). But how about the other sample out of the specify range ? A potential solution will be discussed at next chapter. Besides, the accuracy of ensemble learning model for resample data are also close to 100%. The difference between Neural Network and Ensemble Learning is that the latter are not required PCA reduction. So when dimensionality of data are large and all of them are too important to drop out by dimension reduction, Ensemble Learning will be a suitable model for high dimension data.

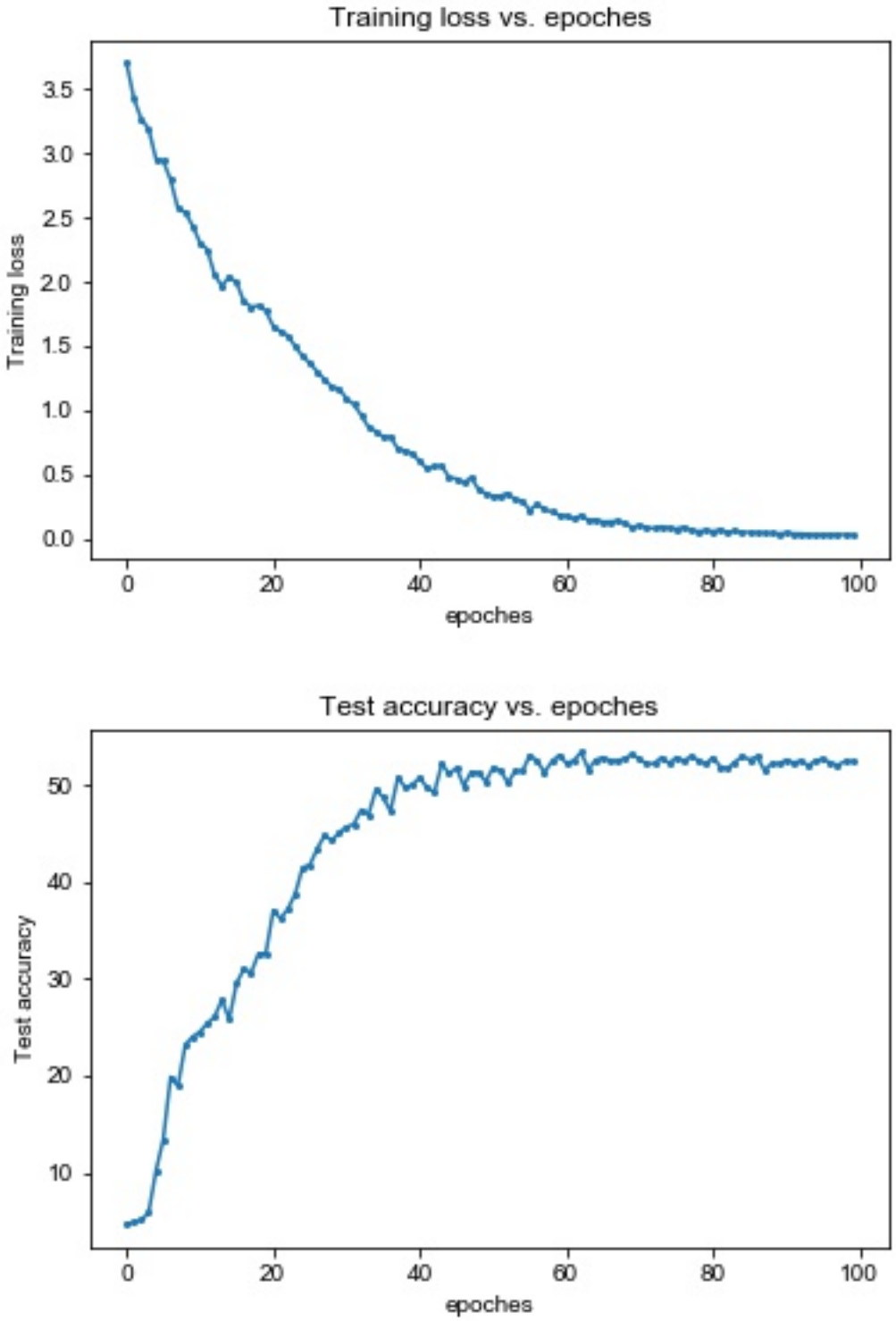

Figure 6.9: training loss and testing accuracy of original data after PCA

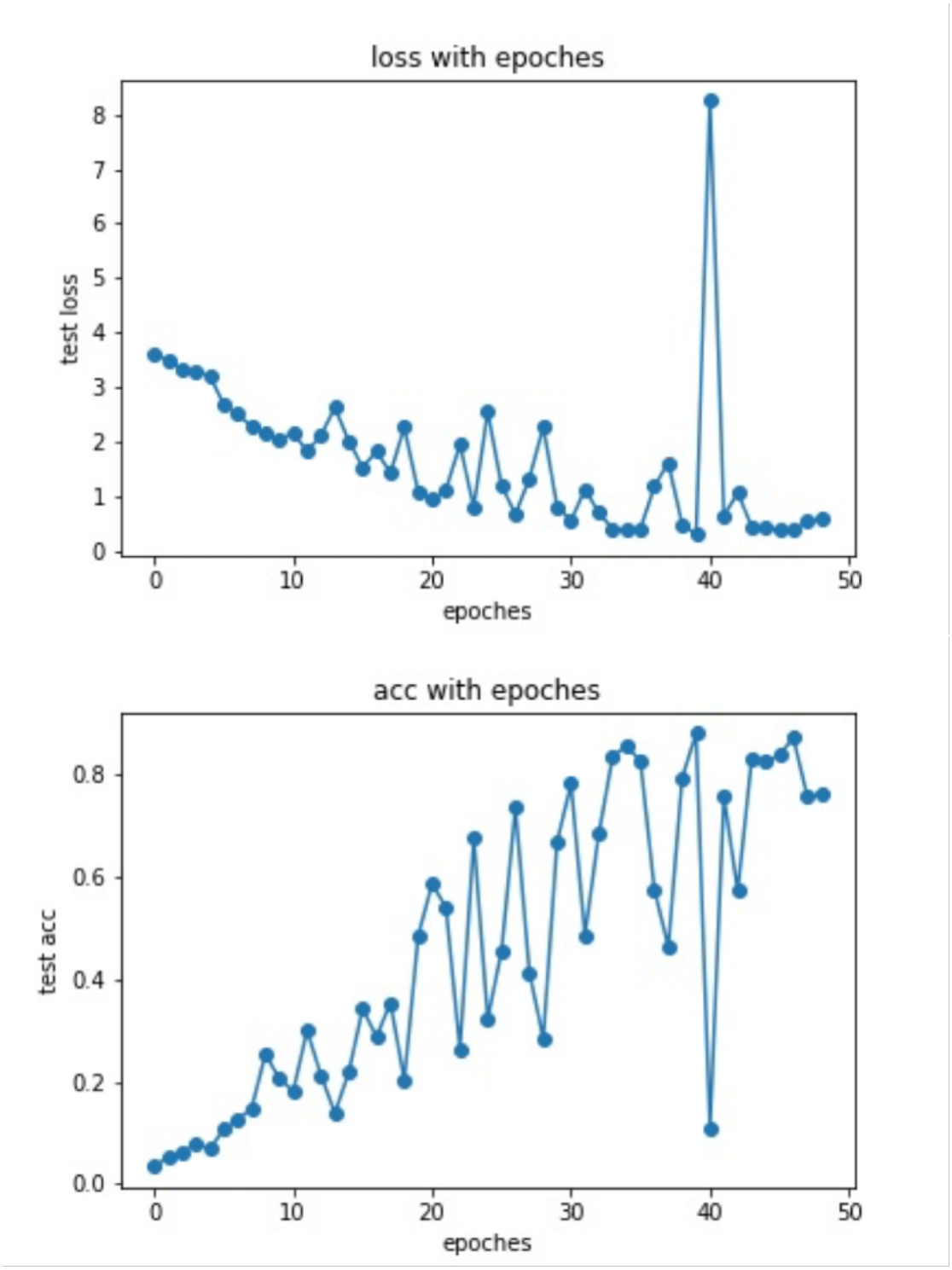

Figure 6.10: training loss and testing accuracy of resample data without PCA

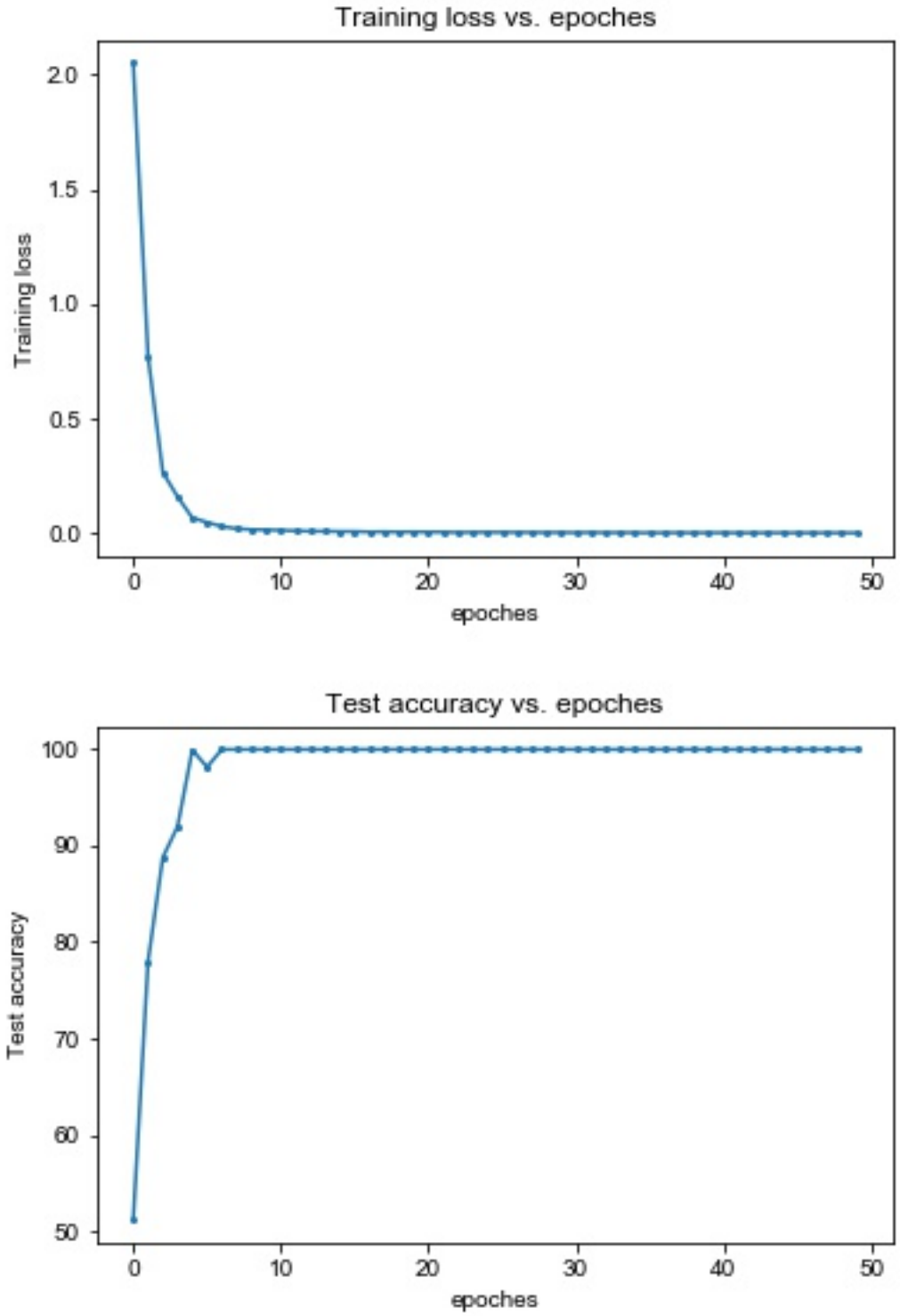

Figure 6.11: training loss and testing accuracy of resample data after PCA

Chapter 7

Discussion

In an earlier research, input data always be defined as image and applied in model designed for image recognition like Convolutional Neural Network. Compare with image, the matrix which elements relate to hit number could stands for the feature and quality of data directly. For example, as Fig 7.1, there are 4 lion pictures with different color, in computer language, that are 4 different matrix. But either human's eye or image recognition model, those 4 pictures always be regard as the same object. Because machine learning model always pay attention to compute and search the correlation between different pixel in the image not the values of pixel i.e. elements of matrix. However in this research, the number of hit particles are main feature for precisely defining data with definite beam size, it is a reason for transferring data from image to matrix.The next step is that choose appropriate model which is sensitive to numerical value, meanwhile utilizing dimension reduction for reducing compute complexity and improve model's accuracy. Besides, due to high statistical fluctuations i.e. high bias, the accuracy of neural network model and ensemble learning model are 50% and 40% respectively.

The method of solving high bias are *resample* that combine 20 bunches data to form a new sample, it is effective to deal with high bias which is obvious in T-SNE visualization at Fig 6.8. Applied new data set in the same model, the result of predict accuracy is 100% and 100%. It is seen that either

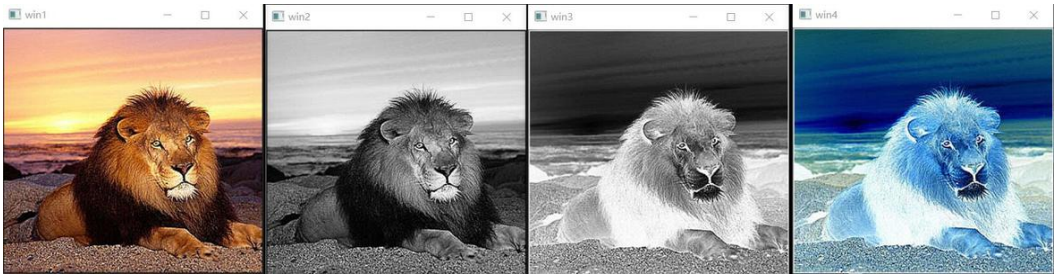

Figure 7.1: The same object with different color

neural network or ensemble learning are able to distinguish those 49 sets of data exactly which is also goal of this research.

For reducing error in measurement of beam size, instead of attempting different algorithm models in earlier research, we try to focus on preprocessor of data that is introduction of EDA, PCA and T-SNE method. It is good for magnifying useful information of data so that improve performance of input data in model which is obvious at conclusion of section 6.4. In other words, preprocessor work well for pair monitor analysis.

As described at section 6.1, there are many difference between classification problem and regression problem. When Machine learning applied in ILC project, regression model is a more reliable choice than classification model. In classification model, it could give answer which class the target sample belong to, more classes more precise. But if once the target sample not belong to any classes in model or out of range of classes, the result would be worse than expectation, so classification model are appropriate for data that located in defined range.It is limitations in classification model. As for regression model, its output are not possibility of class but values of beam size. So the regression model are more effective and useful at realistic experience. Although the accuracy of regression mode is not good as classification model inside the defined range, its performance at other range in high quality, in the other word, regression model are more universally than classification model.

However, there are complicated points in training of regression model. For regression model, the data set with continues values of label i.e. parameter of beam size are necessary. But it is conflict with defining the parameter

before simulation of data set. So how to convert classification problem into regression problem that make it more universally at reality experiment is important.

Besides, in our research, the factor which influence the luminosity in ILC are not only the σ_x and σ_y of beam size, but also other parameter of beam. Like the vertical distance the cross angle between two interaction beams. And other parameter also could be measured by Pair Monitor simultaneously. If we want to discover those information from Pair Monitor by Machine Learning, the a larger number of data that corresponding to different value of parameters are required. Actually, the biggest challenges at the moment are how to effectively simulate events under the different parameters, it also our goal in future research.

Bibliography

[1] Halina Abramowicz et al. “The international linear collider technical design report-volume 4: detectors”. In: *arXiv preprint arXiv:1306.6329* 12 (2013).

[2] Halina Abramowicz et al. “The international linear collider technical design report-volume 4: detectors”. In: *arXiv preprint arXiv:1306.6329* 12 (2013).

[3] Halina Abramowicz, Angel Abusleme, et al. “Forward instrumentation for ILC detectors”. In: *Journal of Instrumentation* 5.12 (2010), P12002.

[4] Chris Adolphsen, Maura Barone, et al. *The International Linear Collider Technical Design Report - Volume 3.II: Accelerator Baseline Design*. 2013. arXiv: 1306.6328 [physics.acc-ph].

[5] Hiroaki Aihara, Jonathan Bagger, et al. *The International Linear Collider A Global Project*, p. 34. URL: https://ilchome.web.cern.ch/content/ilc-european-strategy-document.

[6] Sanjeev Arora, Wei Hu, et al. “An analysis of the t-sne algorithm for data visualization”. In: *Conference On Learning Theory*. PMLR. 2018, pp. 1455–1462.

[7] Philip Bambade, Tim Barklow, et al. *The International Linear Collider A Global Project*. 2019.

[8] Ties Behnke, James E Brau, et al. “The international linear collider technical design report-volume 1: Executive summary”. In: *arXiv preprint arXiv:1306.6327* (2013).

[9] Ties Behnke, James E Brau, et al. “The international linear collider technical design report-volume 1: Executive summary”. In: *arXiv preprint arXiv:1306.6327* (2013).

[10] *DD4hep User Manual*. 2021.

[11] George H Dunteman. *Principal components analysis*. 69. Sage, 1989.

[12] Frank Gaede, Ties Behnke, et al. “LCIO-A persistency framework for linear collider simulation studies”. In: *arXiv preprint physics/0306114* (2003).

[13] Werner Herr and Bruno Muratori. “Concept of luminosity”. In: (2006).

[14] Zakaria Jaadi. *A Step-by-Step Explanation of Principal Component Analysis (PCA)*. 2021. URL: https://builtin.com/data-science/step-step-explanation-principal-component-analysis.

[15] Daniel Jeans. *Particle Flow Algorithms*. 2018. URL: http://ias.ust.hk/program/shared_doc/2018/201801hep/program/exp/HEP_20180118_0950_Daniel_Jeans.

[16] Daniel Jeans. *Scans of the simulated ILD models*. URL: hhttps://research.kek.jp/people/jeans/ctpics/v02-00-02/ILD_l5_v05/summary.

[17] K.Hagiwara, R.D.Peccei, et al. In: *Nucl. Phys. B282, 253* (1987).

[18] K.Yokoya. *User’s Manual of CAIN*. 2003. URL: https://ilc.kek.jp/~yokoya/CAIN/cain235/CainMan235.

[19] Y. Kobayashi. *The application of machine learning to an interaction-point beam profle monitor for the ILC*. 2019. URL: http://epx.phys.tohoku.ac.jp/eeweb/paper/2019_Mthesis_ykoba.

[20] *MARLIN - Modular Analysis and Reconstruction for the LINear collider*. URL: http://ilcsoft.desy.de/portal/software_packages/marlin/index_%5C%20eng.html.

[21] Ahmed Mustahid. *Full Detector Simulation of Pair Monitor at 250 GeV ILC and Application of Machine Learning*. 2019. URL: http://epx.phys.tohoku.ac.jp/eeweb/paper/2020_Mthesis_ahmed.

[22] Marko Petri, Markus Frank, et al. "Detector simulations with DD4hep". In: *Journal of Physics: Conference Series*. Vol. 898. 4. IOP Publishing. 2017, p. 042015.

[23] Daniel Schulte. *Beam-beam Effects in Linear Colliders*. 2017. URL: https://e-publishing.cern.ch/index.php/CYRSP/article/view/267.

[24] Seema Singh. *Understanding the Bias-Variance Tradeoff*. 2018. URL: https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229.

[25] "SOI readout ASIC of pair monitor for International Linear Collider". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 637.1 (2011), pp. 53–59. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2011.02.063. URL: https://www.sciencedirect.com/science/article/pii/S0168900211004190.

[26] Toshiaki Tauchi and Kaoru Yokoya. "Nanometer-beam-size measurement during collisions at linear colliders". In: *Physical Review E* 51.6 (1995), p. 6119.

[27] Laurens Van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE." In: *Journal of machine learning research* 9.11 (2008).

[28] Andre Violante. *An Introduction to t-SNE with Python Example*. 2018. URL: https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1.