國立臺灣大學 理學院物理學研究所

碩士論文

Department of Physics
College of Science
National Taiwan University
Master's Thesis

通過基於評分的擴散模型實現快速HGCal探測器模擬

Fast HGCal Detector Simulation via Score-Based Diffusion Models

徐振華

Chen-Hua Hsu

指導教授：陳凱風 教授

Advisor: Kai-Feng Chen, Ph.D.

中華民國113年10月

October 2024

NATIONAL TAIWAN UNIVERSITY

MASTER'S THESIS

Fast HGCal Detector Simulation via Score-Based Diffusion Models

---

Author:
Chen-Hua Hsu

Supervisor:
Dr. Kai-Feng Chen

January 30, 2025

ken91021615@hep1.phys.ntu.edu.tw

Acknowledgements

中文摘要

隨著對撞機的不斷擴建和升級，物理學家面臨著越來越複雜的實驗需求，這導致對計算資源的需求急劇增加。現有的計算能力將難以持續支撐Geant4軟體完成精確且大規模的全套物理計算模擬，因此，尋求一種更加高效、快速的模擬方法已成爲當前的研究重點。在此論文中，我們提出了使用擴散模型作爲核心演算法，並結合transformer模型，嘗試模擬粒子能量在探測器內部的空間分佈。這一方法不僅能夠顯著加速模擬過程，還保持了與Geant4模擬結果相似的精度。本研究的最大特色在於其能夠生成與Geant4預測高度一致的三維能量分佈圖，而不僅僅是如同大多數類似研究所展示的在一維空間上的能量分佈。

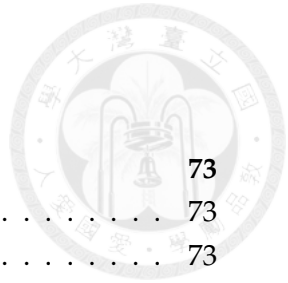
關鍵詞：快速模擬、擴散模型、Transformer、CaloChallenge、HGCal。

Abstract

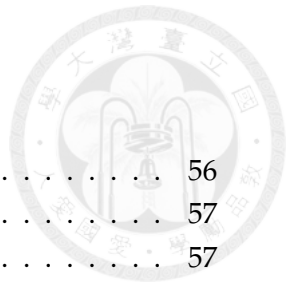
As particle colliders continue to expand and upgrade, physicists face increasingly complex experimental demands, which in turn have led to a sharp rise in the need for computational resources. The current computational power will struggle to support full-scale and precise simulations using Geant4 software, especially as the scale of experiments grows. Therefore, finding a more efficient and fast simulation method has become a pressing priority in current research. In this thesis, we propose using a diffusion model as the core algorithm, coupled with a transformer model, to simulate the spatial distribution of particle energy within the detector. This approach not only significantly accelerates the simulation process but also maintains a level of accuracy comparable to Geant4 simulations. The key feature of this research lies in its ability to generate three-dimensional energy distributions that closely match those predicted by Geant4, rather than the one-dimensional energy distributions typical of most similar studies.

Keywords: Fast Simulation, Diffusion Model, Transformer, CaloChallenge, HGCal.

Contents



List of Figures



List of Tables

Chapter 1

Introduction

1.1 Motivation

The upcoming High Luminosity phase of the Large Hadron Collider (LHC) [1] offers unprecedented opportunities to explore new physics in both ATLAS [2] and CMS [3], with its increased luminosity enabling the collection of vast amounts of experimental data. Notably, from Run 2 to Run 3, the luminosity increased by approximately twofold [4].

With a higher collision rate, the collider is expected to produce around 1 billion proton-proton (p-p) collisions per second, captured by detectors containing nearly 100 million readout channels. With only 25 nanoseconds between consecutive groups of colliding protons, new collisions occur even before the previous interactions have fully exited the detector. This massive volume of data not only represents a rich source for scientific discoveries but also poses immense challenges in terms of data processing, storage, and simulation requirements.

Simulation plays a critical role in high-energy physics, as it enables researchers to determine if experimental data aligns with theoretical models. Before delving into deeper analyses, every study must first validate that the observed data is consistent with both background expectations and the signal. This essential step ensures that we have a solid understanding of the main background and signal contributions to each channel, allowing us to apply suitable analysis strategies. However, the high computational demands of this simulation process create bottlenecks, especially as data rates increase. Thus, accelerating the simulation process without compromising accuracy is crucial for timely and reliable analysis.

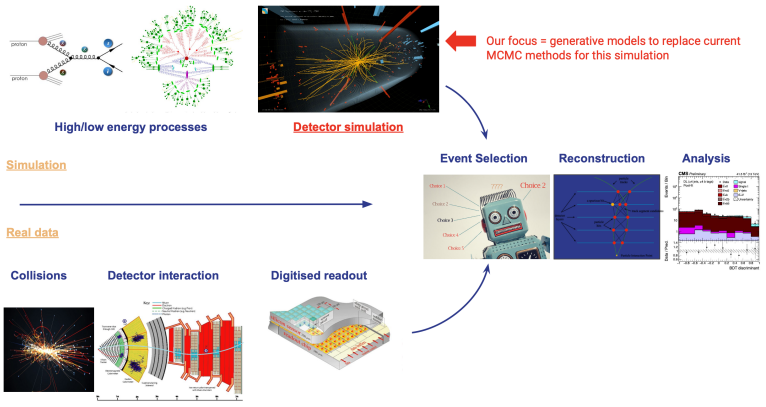


FIGURE 1.1: The importance of simulation. credit: Joshuha Thomas-Wilsker

The simulation of particle interactions and detector responses, traditionally carried out by Monte Carlo methods and implemented with tools like Geant4 [5], has been a foundational aspect of high-energy physics research. However, these methods are computationally intensive and struggle to keep pace with the data rates expected in the Run3 and even future. As the complexity of detector and collider setups continues to grow, so does the time required for full simulations, making it increasingly difficult to scale traditional techniques to meet the demands of modern experiments.

In response to the challenges posed by increasing data rates, generative models—especially diffusion models—have shown promising potential to accelerate simulations without sacrificing quality. Our goal is not to fully replace Geant4 simulations but rather to find a balance between accuracy and speed, as illustrated in Figure 1.2 . Recent works, including Yang et al.'s score-based models [6] and other diffusion approaches in calorimeter simulations [7], have achieved significant reductions in computation time while maintaining fidelity. Building on these innovations, our project introduces a novel model designed to generate 3D point clouds representing energy distributions across spatial coordinates in one step. Unlike previous models, which often focus on single-dimensional energy profiles (e.g., energy vs. z-coordinate), our approach captures the full 3D energy distribution in a single forward pass, allowing for rapid and comprehensive simulations that could match the data collection demands of high-luminosity experiments.

Detailed detector simulations are crucial in particle and nuclear physics data analysis. They allow researchers to compare particle-level predictions with observed data and to account for detector effects, making it possible to accurately interpret experimental results and compare them with theoretical predictions. Simulations also play a significant role in designing future experiments, guiding adjustments to optimize detector performance [8, 9]. Geant4-based simulations [10] have become a standard

tool in high-energy physics due to their precision, but achieving this precision is computationally demanding. Particle propagation in dense materials produces numerous secondary particles that undergo complex electromagnetic and nuclear interactions, making calorimeters—whose role is to measure deposited energy—the most challenging detectors to simulate accurately. In fact, a large portion of computing resources in high-energy physics is dedicated to simulating particle propagation in dense materials using Geant4.

The experiments at the Large Hadron Collider (LHC) generate billions of events per run, each with hundreds to thousands of individual calorimeter showers. Due to computing budget constraints, it is not feasible to use Geant4 simulations for all events, so experiments have developed fast simulation methods. These methods replace physics-based models with simpler parametric models calibrated to full simulations. While these fast simulations are efficient, their simplified parameterizations limit their accuracy, especially for modeling complex, high-dimensional correlations. Often, only a few one-dimensional observables are optimized, which may not fully capture the intricacies of particle interactions.

Deep learning provides a compelling alternative to traditional parametric models, with generative approaches like Generative Adversarial Networks (GANs) [11], Variational Autoencoders (VAEs) [12], and Normalizing Flows (NFs) [13] increasingly adapted for fast detector simulations. GANs, for example, have demonstrated considerable speed and adaptability in generating calorimeter showers [14] and are now even integrated into the ATLAS experiment's fast simulation framework [15]. Nonetheless, GANs present optimization challenges and can suffer from mode collapse,where the generator produces a narrow range of outputs, failing to capture the diversity of the data distribution. Conversely, NFs provide robust training and accurate density estimation, yet they remain computationally demanding when applied to high-dimensional data, which limits their practicality for simulating complex detector responses [16, 17].

In this work, we explore score-based generative models [6], which learn the gradient of the data density rather than the density itself, enabling a more flexible network architecture without requiring the Jacobian computation during training. This flexibility supports the use of bottleneck layers, reducing trainable parameters and improving scalability. Recent advances in score-based generative models have shown potential in calorimeter simulation, achieving a balance between high-dimensional fidelity and computational efficiency, making them suitable for ultra-fine calorimeters and other high-complexity datasets [18, 19].

By leveraging score-based models, our project aims to address both the demands of

the high-luminosity phase of the LHC and the limitations of traditional fast simulation methods. Our approach enhances accuracy by capturing full 3D spatial distributions while significantly reducing the time required for simulation, thus providing a scalable, reliable solution for next-generation collider experiments.

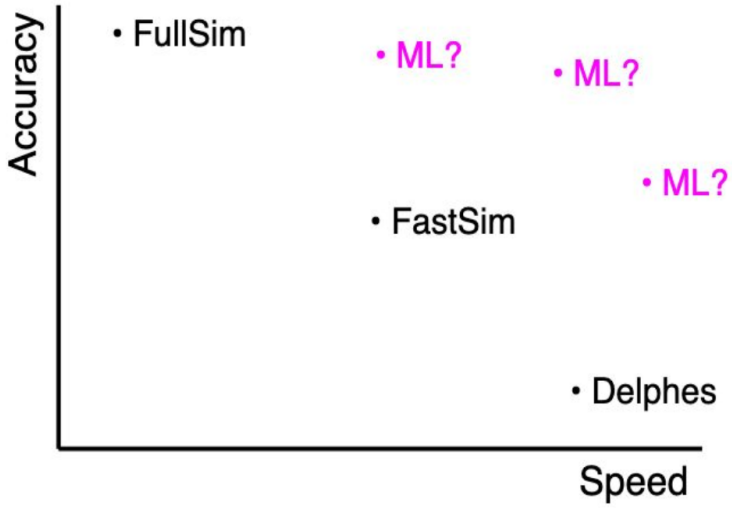


FIGURE 1.2: The balance between accuracy and speed in simulation. credit: Joshuha Thomas-Wilsker

1.2 Challenges

The generation of a 3D point cloud to depict energy deposition across spatial coordinates introduces unique challenges. Existing approaches primarily model the relationship between energy and a single spatial dimension, typically generating only partial representations of energy distributions. Our model, by contrast, aims to capture the complete three-dimensional energy profile in a single forward pass, which requires balancing high-dimensional fidelity and computational efficiency.

To achieve this goal, our model leverages advanced features, including Gaussian Fourier Projection for time encoding and mean-field attention mechanisms with a class token, in addition to conditional guidance based on incident energy. These architectural choices allow us to control both positional and energy distributions, addressing the intricacies of accurate 3D spatial modeling. However, managing the computational load and also let model learn every relation between each variables is quite hard.

This high-dimensional generative task requires careful conditioning to reflect realistic variations in energy deposition across multiple spatial coordinates, especially

given the model's need to dynamically adjust based on the incident energy. Achieving this balance involves a tradeoff between accuracy and computational load, as the high fidelity demanded in multi-dimensional output often requires extensive computation. Nevertheless, our optimized approach achieves up to a 100-fold speedup over traditional simulation methods, providing a scalable solution that addresses the needs of next-generation collider experiments.

In summary, our project seeks to address both the demands of high luminosity and the constraints of traditional simulations, aiming to bridge the gap between scalability and fidelity in particle shower simulations. Our advancements in 3D point cloud generation not only enhance the efficiency of simulations but also mark a step forward in producing realistic, high-dimensional data essential for future discoveries in high-energy physics.

Chapter 2

Detector

2.1 The Large Hadron Collider (LHC)

Although the Standard Model of particle physics has been remarkably successful up to TeV scale, several fundamental questions remain unanswered. The Large Hadron Collider (LHC) at CERN is the most powerful particle accelerator ever built, designed to explore above the TeV energy scale. It consists of a 27-kilometer ring of superconducting magnets and accelerating structures, enabling proton-proton collisions at an unprecedented energy of 13 TeV (design energy of 14 TeV). The main purpose of the LHC is to explore the electroweak symmetry breaking for which the Higgs mechanism is presumed to be responsible and search for new physics beyond the Standard Model.

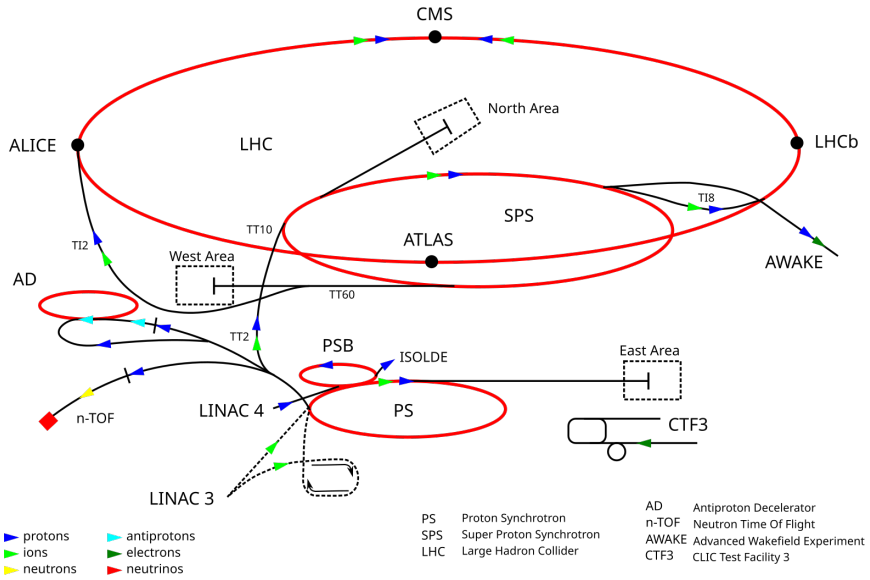


FIGURE 2.1: The schema of LHC [20]

The LHC features a high collision rate with 25 ns bunch spacing, producing up to 10^9 interactions per second. The facility includes key experimental sites like CMS,

ATLAS, LHCb, and ALICE, each optimized for specific research goals. The injection system consists of the Proton Synchrotron (PS) and Super Proton Synchrotron (SPS), ensuring high beam luminosity and energy.

2.1.1 Key Components of the LHC

Injector Chain

The LHC relies on a sequence of pre-accelerators to prepare the particle beams:

- **Linear Accelerator (Linac4):** Replaced Linac2 and accelerates negative hydrogen ions (H^-) to 160 MeV. Synchrotron Booster (PSB). [21]
- **Proton Synchrotron Booster (PSB):** Strips electrons from H^- ions to produce protons and accelerates them to 2 GeV. [22]
- **Proton Synchrotron (PS):** Further increases the beam energy to 26 GeV.[23]
- **Super Proton Synchrotron (SPS):** Boosts the energy of protons to 450 GeV before injection into the LHC.[24]

Each stage ensures the beam achieves the required energy, intensity, and quality, culminating in proton-proton collisions at 13.6 TeV in the LHC in Run3.

Main Ring

The LHC ring consists of two counter-rotating beam pipes, maintained under ultra-high vacuum conditions to avoid particle collisions with residual gas.

- **Superconducting Magnets:** Approximately 1,232 dipole magnets steer the beams around the circular path, while quadrupole magnets focus them to maintain stability. [25]
- **Cryogenics:** The superconducting magnets operate at 1.9 Kelvin (-271), achieved using liquid helium cooling systems. [26]

Experimental Sites

The LHC includes four main experiments, strategically placed along the ring:

- **CMS (Compact Muon Solenoid):** Focused on studying high-energy collisions for precision measurements and new physics.
- **ATLAS (A Toroidal LHC Apparatus):** Another general-purpose detector designed for broad physics exploration.
- **ALICE (A Large Ion Collider Experiment):** Specializes in studying heavy-ion collisions and the quark-gluon plasma.

- **LHCb (LHC Beauty Experiment):** Dedicated to investigating the matter-antimatter asymmetry by studying b-hadron decays.

Collimation and Beam Dumps

The LHC is equipped with a sophisticated collimation system to remove stray particles and protect sensitive components. Beam dumps allow controlled termination of particle beams after experiments or emergencies.

Collision Points

Particles are brought to collision points within the detectors, achieving a luminosity of $10^{34}\ \mathrm{cm}^{-2}\mathrm{s}^{-1}$. These conditions facilitate rare particle processes, such as Higgs boson production.

2.1.2 Technological Challenges

- **Radiation Damage:** Extensive shielding is required to protect equipment and personnel from high levels of radiation.
- **Alignment Precision:** The alignment of the LHC's components must be maintained within micrometers to ensure proper beam steering.
- **Data Volume:** Experiments generate petabytes of data annually, necessitating advanced computational infrastructure for storage and analysis.

The LHC represents a pinnacle of human engineering and scientific collaboration, involving thousands of scientists and engineers worldwide.

2.2 The Compact Muon Solenoid (CMS)

CMS is a general-purpose detector optimized for high-precision measurements and searches for rare physics events. The detector design focuses on:

- Precise tracking for charged particles.
- High-resolution electromagnetic and hadronic calorimetry.
- Efficient muon identification and momentum resolution.
- Robust missing transverse energy measurement.

In order to detect the momentum of muons The CMS detector features a 4 Tesla superconducting solenoid with a 6-meter diameter and 12.5-meter length, providing a strong magnetic field essential for accurate momentum measurements of charged particles. The solenoid is enclosed inside a 10,000-tonne iron return yoke, which serves

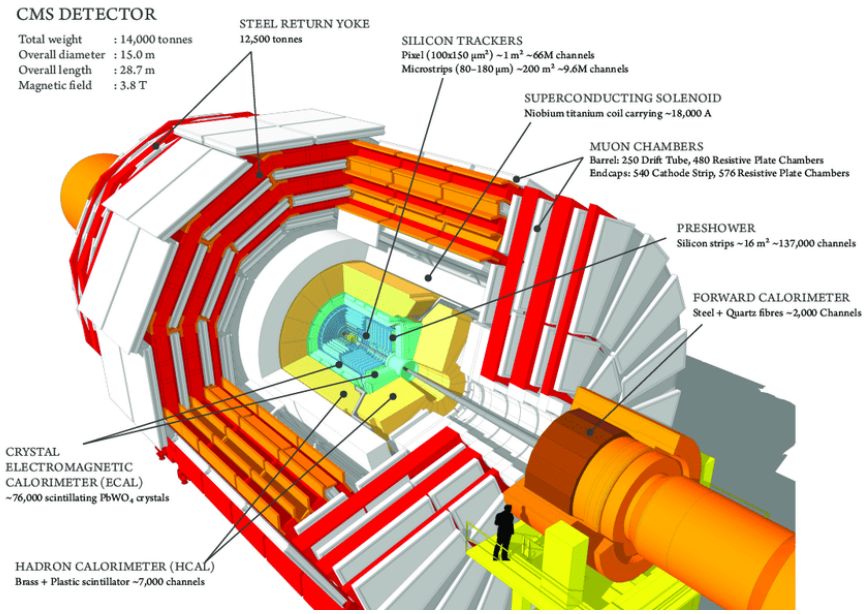


FIGURE 2.2: Exploded view of the CMS detector, showing its main components.

to contain the magnetic field and also houses the muon detection system.[27] The CMS muon spectrometer is based on gaseous detectors placed inside the iron return yoke of the superconducting solenoid.[28]

In order to better illustrate the CMS detector, the figure below is the Cross-sectional schematic of the CMS detector showcasing its key components: the Silicon Tracker, Electromagnetic Calorimeter (ECAL), Hadron Calorimeter (HCAL), and Superconducting Solenoid.

2.3 Silicon Tracker

The tracker system in the CMS detector is designed to reconstruct the trajectories of charged particles produced in high-energy collisions with unparalleled precision. This subsystem plays a vital role in measuring the momentum of particles, identifying particle types, and reconstructing primary and secondary vertices.

2.3.1 Silicon Pixel Detector

The innermost layer of the tracker is the silicon pixel detector, which provides high-resolution tracking near the interaction point. It consists of three barrel layers and two endcap disks on either side, covering a pseudorapidity range of $|\eta| < 2.5$ [29]. The pixel detector is constructed using silicon sensors segmented into millions of tiny pixels, each measuring $100 \times 150\ \mu m^2$.

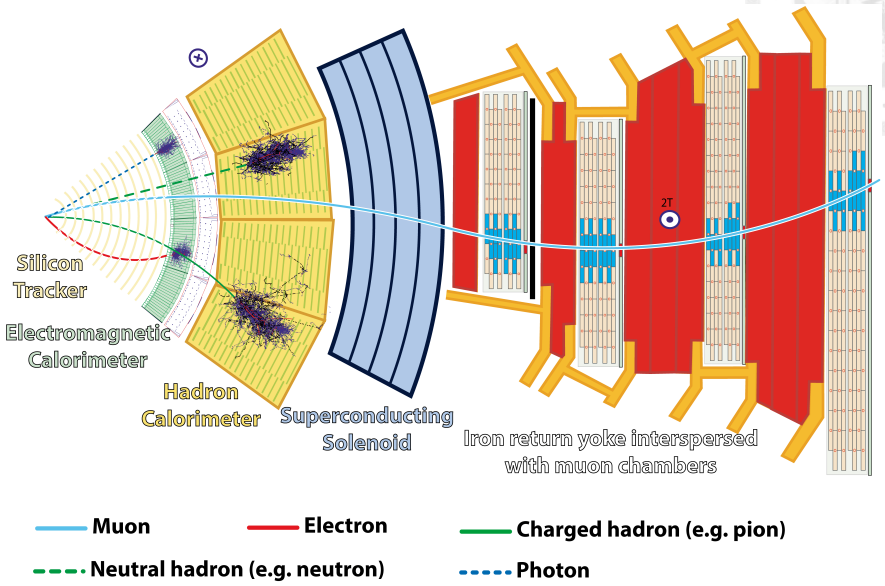


FIGURE 2.3: Cross-sectional schematic of the CMS detector

The pixel detector is designed to withstand intense radiation levels and high particle flux near the beamline. Its fine granularity ensures excellent spatial resolution, which is critical for identifying displaced vertices from the decays of short-lived particles such as B-mesons and au leptons [30].

2.3.2 Silicon Strip Tracker

Surrounding the pixel detector is the silicon strip tracker, which extends the tracking coverage to larger radii and provides additional layers for trajectory reconstruction. The strip tracker is divided into the Tracker Inner Barrel (TIB), Tracker Outer Barrel (TOB), Tracker Endcaps (TEC), and Tracker Inner Disks (TID). These components collectively cover a radial distance of 20 to 110 cm from the beamline [29].

The silicon strips are oriented in parallel arrays, with each strip measuring several centimeters in length and a few hundred microns in width. By combining signals from multiple layers, the strip tracker achieves precise momentum measurements and improves the robustness of the trajectory reconstruction. [31]

2.3.3 Material Choices and Performance

The tracker is constructed entirely from silicon sensors, chosen for their excellent resolution and radiation hardness. Key considerations in the design include:

- **Lightweight support structures:** Minimize material interactions that can scatter particles and degrade tracking performance.

- **Radiation-tolerant electronics:** Ensure reliable operation in the high-radiation environment of the LHC.
- **High granularity:** Allows for precise reconstruction of particle trajectories even in the presence of multiple simultaneous collisions (pile-up).

The tracker achieves a transverse momentum resolution of approximately $\Delta p_T/p_T = 1\%$ for particles with p_T around $100\ \mathrm{GeV/c}$. This precision enables detailed studies of particle properties, including invariant mass reconstruction and decay vertex identification [29].

The tracker is designed to withstand high radiation levels and provides a momentum resolution of $\Delta p_T/p_T \approx 1\%$ for particles with $p_T \sim 100\ \mathrm{GeV/c}$. The low-mass design minimizes material interactions, reducing the impact on photon and electron measurements. Cooling systems maintain stable operation despite the intense radiation environment.

2.4 Electromagnetic Calorimeter (ECAL)

The Electromagnetic Calorimeter (ECAL) in the CMS detector is a crucial subsystem designed to measure the energy of electrons and photons with high precision. The ECAL achieves this by utilizing scintillating lead tungstate ($PbWO_4$) crystals as the active medium, coupled with photodetectors to convert scintillation light into electrical signals. Its design, divided into the Barrel (EB), Endcap (EE), and Preshower Detector (ES), ensures optimal performance across a wide range of pseudorapidity. In this research, because our dataset is mainly focused on photons and electrons, the ECAL is actually the region we mainly focus on.

2.4.1 The ECAL Barrel (EB)

The ECAL Barrel covers the central pseudorapidity region, $|\eta| < 1.479$, and consists of approximately 61,200 $PbWO_4$ crystals. These crystals are characterized by their high density, fast scintillation time, and radiation hardness [32]. Lead tungstate is chosen due to its high density and short radiation length, which allows electromagnetic showers to develop within a compact volume. This compactness ensures that the ECAL can achieve high resolution while fitting within the spatial constraints of the CMS detector.

Each crystal is aligned quasi-projectively towards the interaction point, ensuring minimal gaps in coverage and precise angular resolution. The scintillation light produced in the crystals is detected by avalanche photodiodes (APDs) for the barrel region, which offer excellent sensitivity and radiation resistance [32].

2.4.2 The ECAL Endcap (EE)

The ECAL Endcap extends the coverage of the ECAL to higher pseudorapidities, from $|\eta| = 1.479$ to $|\eta| = 3.0$. The endcap region is composed of roughly 14,600 $PbWO_4$ crystals, arranged in a geometry optimized for forward physics studies [32]. Due to the higher radiation levels and particle flux in this region, the photodetectors used are vacuum phototriodes (VPTs), which are more robust against radiation damage compared to APDs.

The higher radiation environment in the endcap region also necessitates additional cooling and monitoring systems to maintain the performance of the crystals and photodetectors. The EE plays a critical role in measuring photons and electrons produced at small angles relative to the beamline, ensuring comprehensive detector coverage [32].

2.4.3 The Preshower Detector

The preshower detector is located in front of the ECAL Endcaps and is designed to enhance the discrimination between photons and neutral pions (π^0). It consists of two layers of lead absorbers, interleaved with silicon strip sensors [33]. The lead layers initiate electromagnetic showers, while the silicon sensors measure the spatial distribution of the resulting particles.

This design allows the preshower detector to effectively distinguish between single photons and π^0 decays, which produce two closely spaced photons. This capability is crucial for improving the ECAL's performance in identifying isolated photons in a high-particle-density environment [33].

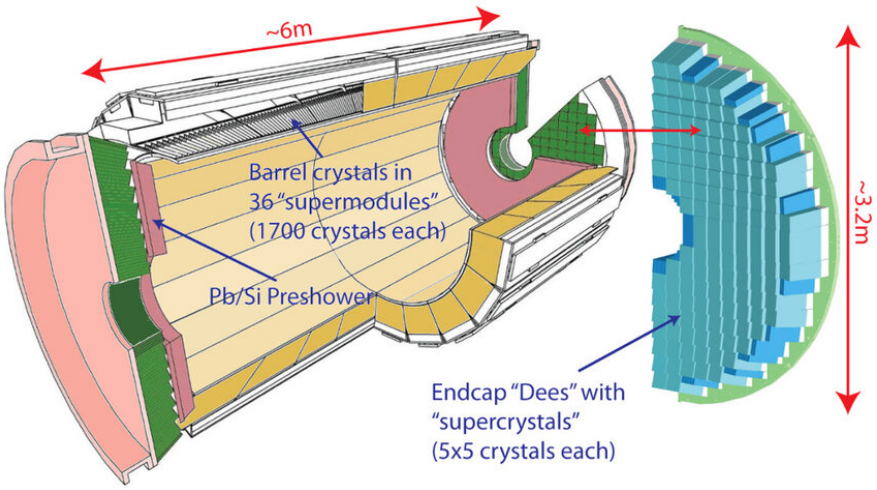


FIGURE 2.4: Structure of the ECAL showing barrel and endcap regions. [34]

2.4.4 Material Choices and Performance

The choice of $PbWO_4$ as the active material for the ECAL is driven by its unique properties:

- **High density and short radiation length:** These properties allow electromagnetic showers to be contained within a compact volume, ensuring precise energy measurements.
- **Fast scintillation time:** $PbWO_4$ crystals have a decay time of approximately 25 ns, matching the LHC's bunch crossing interval [32].
- **Radiation hardness:** $PbWO_4$ is resistant to radiation damage, which is critical for maintaining detector performance over extended periods of operation.

The ECAL achieves an excellent energy resolution, parameterized as:

$$\frac{\sigma_E}{E} = \frac{S}{\sqrt{E}} \oplus \frac{N}{E} \oplus C,$$

where S is the stochastic term, N represents the noise, and C is the constant term [32]. This resolution allows the ECAL to distinguish between different particle species and measure their energies with high precision, making it an indispensable tool for studies of Higgs boson decays, rare processes, and new physics searches.

2.5 Hadronic Calorimeter (HCAL)

The HCAL measures hadronic energy, complementing the ECAL in reconstructing jets and missing transverse energy. It employs a sampling design with brass absorbers and plastic scintillators.

The Hadronic Calorimeter (HCAL) in the CMS detector is an essential component designed to measure the energy of hadrons produced in high-energy collisions. The HCAL achieves this through a carefully engineered combination of absorber and active materials, divided into distinct regions optimized for different pseudorapidity ranges. These regions include the HCAL Barrel (HB), HCAL Endcap (HE), HCAL Forward (HF), and HCAL Outer (HO). The selection of materials and their specific configurations in each section is driven by the requirements of energy containment, radiation hardness, and detector efficiency.

2.5.1 The HCAL Barrel (HB)

The HCAL Barrel is the central component of the HCAL, covering the region close to the interaction point with a pseudorapidity range of $|\eta| < 1.3$. The HB is constructed

using brass as the absorber material and plastic scintillators as the active medium. Brass is chosen due to its high density and structural stability, which allow it to efficiently stop high-energy hadrons and initiate hadronic showers.[35] The dense nature of brass ensures that the hadronic showers are contained within a compact volume, which is critical for the limited space available in the detector.

The active medium in the HB consists of plastic scintillator tiles, which emit light when traversed by charged particles generated in the hadronic showers. This scintillation light is collected by photodetectors, such as silicon photomultipliers, and converted into an electrical signal proportional to the energy deposited in the calorimeter. The use of plastic scintillators ensures a fast response time, high light yield, and excellent linearity, all of which contribute to the precision of energy measurements.

2.5.2 The HCAL Endcap (HE)

The HCAL Endcap extends the coverage of the HCAL to higher pseudorapidities, from $|\eta| = 1.3$ to $|\eta| = 3.0$. Similar to the HB, the HE uses brass as the absorber material and plastic scintillators as the active medium. However, the endcap is designed to handle particles with higher momenta, which require increased thickness of the absorber layers to fully contain the hadronic showers.

The higher density and thickness of the brass absorbers in the HE ensure that the energy of the hadronic showers is completely absorbed, even for particles at extreme angles. The endcap region is critical for capturing the energy of forward jets and particles produced at small angles relative to the beamline, ensuring no significant gaps in the detector's acceptance.[35]

2.5.3 The HCAL Forward (HF)

The HCAL Forward is specifically designed to handle the extreme forward region, covering $3.0 < |\eta| < 5.0$. This region experiences the highest particle flux and radiation levels, necessitating the use of radiation-hard materials such as steel for the absorbers and quartz fibers for the active medium. Steel is chosen for its durability and ability to withstand the intense radiation environment in the forward region. It also provides the density required to stop high-energy hadrons effectively.

The active medium in the HF consists of quartz fibers, which generate Cherenkov light when traversed by relativistic charged particles produced in the hadronic showers. Cherenkov light is collected by specialized photodetectors, providing a robust signal in an environment where plastic scintillators would suffer significant degradation. This combination of materials ensures that the HF maintains its performance over long periods of operation, even in the harshest conditions.

The HF plays a crucial role in studying forward physics phenomena, including parton distribution functions and diffractive events. Its design also contributes to the accurate measurement of missing transverse energy (E_T^{miss}) by reducing the likelihood of undetected particles escaping.[36]

2.5.4 The HCAL Outer (HO)

The HCAL Outer is located outside the superconducting solenoid and complements the energy measurements of the HB. The HO uses the steel return yoke of the solenoid as its absorber, with additional layers of plastic scintillators serving as the active medium. The primary purpose of the HO is to act as a "tail catcher," capturing energy from high-energy particles that pass through the HB and the solenoid without being fully absorbed.

Using the steel return yoke as an integral part of the calorimeter minimizes the overall size and weight of the detector while maintaining its energy containment capabilities. The additional scintillator layers ensure that any residual energy from penetrating particles is measured, providing a complete picture of the event's energy balance. [37]

2.5.5 Material Choices and Their Impact

The material choices for the HCAL are driven by the need to balance density, radiation hardness, and signal quality. Dense materials such as brass and steel are used to contain hadronic showers within a compact volume, minimizing leakage and ensuring precise energy measurements. Plastic scintillators are employed in regions with lower radiation exposure due to their high light yield and fast response time. In contrast, quartz fibers are used in the forward region, where their radiation tolerance and ability to generate Cherenkov light make them the ideal choice.

This careful selection of materials and their strategic placement within the HCAL ensures that the detector meets the stringent requirements for hadronic energy measurement. By providing accurate jet energy reconstruction and missing transverse energy measurements, the HCAL plays a vital role in the CMS detector's ability to explore physics at the energy frontier. [38]

2.5.6 Performance

The HCAL provides energy resolution of:[39]

$$\frac{\sigma_E}{E} = \frac{S}{\sqrt{E}} \oplus C.$$

The combination of ECAL and HCAL ensures accurate jet energy reconstruction and E_T^{miss} measurements, critical for new physics searches.

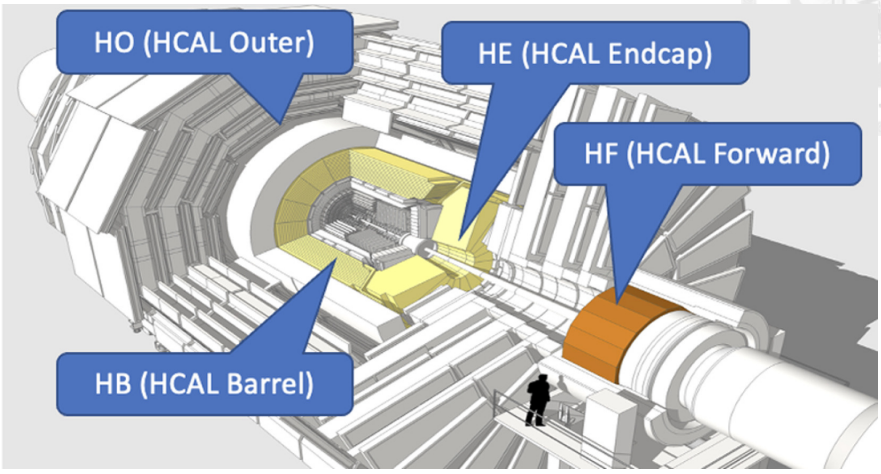


FIGURE 2.5: Schematic of the HCAL with barrel, endcap, and forward sections.
[hgcal_picture]

2.6 Muon Detector

The muon detector in the CMS experiment is a crucial subsystem designed to identify and measure the momentum of muons, which are often key signatures in high-energy collisions. The muon system provides the outermost layer of the CMS detector, ensuring precise muon tracking and efficient triggering across a wide range of pseudorapidity.

2.6.1 Muon Chambers: Drift Tubes (DT)

Drift tubes are the primary technology used in the barrel region of the CMS detector, covering $|\eta| < 1.2$. They consist of gas-filled chambers with wires running along their length. When a muon passes through the chamber, it ionizes the gas, and the resulting electrons drift toward the central wire under the influence of an electric field [40].

The time taken by the electrons to reach the wire allows for precise measurements of the muon's position. The DTs are arranged in layers, providing redundancy and improving spatial resolution. The use of drift tubes in the barrel region ensures robust performance in areas with lower radiation exposure and relatively uniform magnetic fields.

2.6.2 Muon Chambers: Cathode Strip Chambers (CSC)

Cathode strip chambers are employed in the endcap regions, where the pseudorapidity ranges from $1.2 < |\eta| < 2.4$. The CSCs are designed to operate in areas with higher radiation levels and non-uniform magnetic fields. They consist of multi-layered gas chambers with cathode strips and anode wires arranged perpendicularly [40].

When a muon traverses a CSC, it ionizes the gas, and the resulting charge is collected on the strips and wires. The perpendicular arrangement allows for precise two-dimensional position measurements. This design ensures high efficiency and excellent spatial resolution in the endcap regions, where particle flux and radiation are more intense [40].

2.6.3 Resistive Plate Chambers (RPC)

Resistive plate chambers are used in both the barrel and endcap regions, providing fast timing information and additional redundancy for triggering. RPCs consist of parallel resistive plates separated by a thin gas layer. When a muon passes through the gas, it creates an avalanche of electrons, resulting in a detectable signal [40].

The fast response time of RPCs makes them ideal for the Level-1 trigger system, which is responsible for selecting events of interest in real time. Their simple design and robust performance contribute significantly to the overall efficiency of the muon detector.

2.6.4 Material Choices and Performance

The materials and technologies used in the muon detector are carefully chosen to meet the demands of high-energy particle physics experiments:

- **Gas-filled chambers:** Used in DTs and CSCs for their ability to provide precise spatial measurements and operate in high-radiation environments.
- **Resistive materials:** Employed in RPCs to ensure fast timing and robust performance under high particle flux.
- **Redundant layering:** Multiple layers of chambers improve tracking resolution and ensure reliability in detecting muons.

The muon system achieves a momentum resolution of $\Delta p/p \sim 10\%$ at $1\ \mathrm{TeV/c}$, enabling precise measurements of high-momentum muons [40]. This capability is critical for identifying rare processes, such as those involving heavy bosons or new particles.

The muon system achieves momentum resolution of $\Delta p/p \sim 10\%$ at $1\ \mathrm{TeV/c}$, contributing significantly to global track reconstruction.[42]

2.6.5 Trigger and Reconstruction

The CMS trigger system is essential for managing the vast amount of data generated by the detector, selecting only the most relevant events for further analysis. The trigger operates in two levels: the Level-1 Trigger and the High-Level Trigger (HLT).

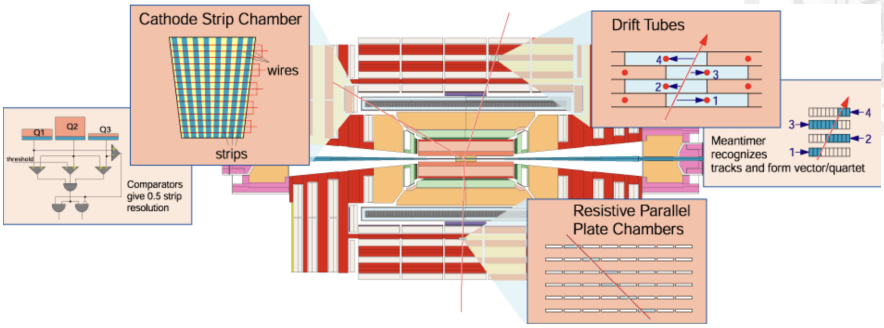


FIGURE 2.6: CMS Muon System layout, showing DTs, CSCs, and RPCs. [41]

2.6.6 Level-1 Trigger

The Level-1 Trigger is a hardware-based system designed to process data in real time and reduce the event rate from 40 MHz to approximately 100 kHz [43]. It uses custom electronics located close to the detector to analyze data from the calorimeters and muon chambers. This system identifies candidate particles such as muons, electrons, and jets, and makes decisions within microseconds.

The Level-1 Trigger ensures that only events with significant physics potential, such as those involving high-energy muons or missing transverse energy, are passed on to the next stage [43].

2.6.7 High-Level Trigger (HLT)

The High-Level Trigger is a software-based system that further reduces the event rate from 100 kHz to approximately 1 kHz, suitable for storage and offline analysis [43]. The HLT uses a computing farm to reconstruct full events in real time, applying more sophisticated algorithms to refine the selection criteria.

This stage enables detailed analysis of particle trajectories and energy deposits, ensuring that only the most promising events are retained for later study. The combination of the Level-1 Trigger and HLT allows CMS to efficiently manage the enormous data flow while preserving the ability to capture rare and significant physics phenomena.

2.7 The High-Granularity Calorimeter (HGCal)

The High-Granularity Calorimeter (HGCal) is a significant upgrade to the Compact Muon Solenoid (CMS) detector at the Large Hadron Collider (LHC). It is designed to operate efficiently in the intense radiation environment of the High-Luminosity LHC

(HL-LHC). Replacing the endcap electromagnetic and hadronic calorimeters, the HGCal features a highly granular sampling calorimeter, enabling precise energy measurements and particle identification under challenging conditions.

However, the unprecedented granularity of the HGCal introduces substantial computational challenges for traditional simulation methods, which struggle to efficiently model the complex detector geometry and interactions. To address this, our research focuses on leveraging deep learning methods to improve simulation performance. By integrating these advanced techniques, we aim to enable faster and more accurate simulations, making the study of high-granularity detectors both feasible and impactful. This goal is crucial to unlocking the full potential of the HGCal and advancing our understanding of fundamental physics.

2.7.1 Structure and Components

The HGCal comprises two main sections: the electromagnetic calorimeter (CE-E) and the hadronic calorimeter (CE-H). Each section is constructed from a series of hexagonal sensor modules, arranged in layers and interleaved with absorber plates.

CE-E: Electromagnetic Section The CE-E is designed to measure the energy of electromagnetic particles such as photons and electrons. It uses silicon sensors as the active material, chosen for their excellent resolution and radiation hardness. These sensors are segmented into hexagonal cells, with each cell covering an area of approximately 1 cm^2. The absorber plates, made of lead, are optimized to initiate electromagnetic showers within a compact volume [44].

CE-H: Hadronic Section The CE-H is responsible for measuring the energy of hadrons and providing complementary information for jet reconstruction. This section uses a combination of silicon sensors and scintillator tiles as active materials, interleaved with steel absorber plates. The hybrid design balances cost and performance, ensuring robust measurements across a wide range of particle energies [44].

2.7.2 Design and Innovations

The HGCal introduces several innovations to meet the demands of the HL-LHC environment:

- **High Granularity:** With over six million readout channels, the HGCal provides unparalleled spatial resolution, allowing for detailed reconstruction of particle showers.

- **Radiation Hardness:** The use of radiation-tolerant silicon sensors ensures long-term performance under intense radiation conditions.
- **Timing Capability:** The HGCal incorporates timing measurements with a precision of a few tens of picoseconds, enabling precise identification of collision vertices and pile-up mitigation [44].

2.7.3 Performance and Applications

The high granularity and timing capabilities of the HGCal significantly enhance the CMS detector's performance in several areas:

- **Particle Flow Reconstruction:** The fine segmentation enables accurate separation of overlapping showers, improving the resolution of energy measurements for jets and missing transverse energy.
- **Pile-Up Mitigation:** The timing information allows for the discrimination of signals from different interaction vertices, reducing the impact of pile-up.

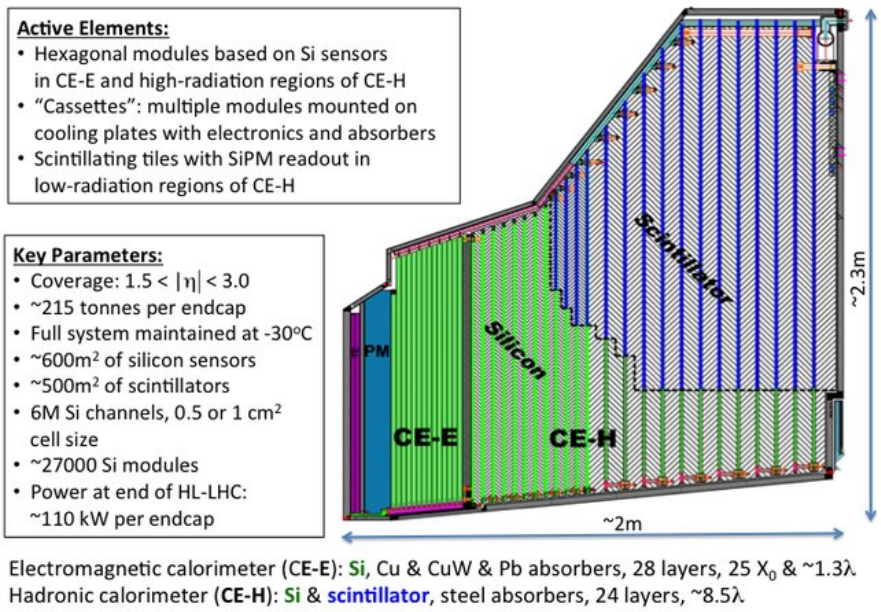


FIGURE 2.7: Schematic of the HGCal showing its layered structure and segmentation. (Image credit: CMS Collaboration)

[45]

The HGCal represents a major technological advancement for calorimetry in high-energy physics, providing the tools necessary to explore the physics potential of the HL-LHC.

2.8 Conclusion

The CMS detector integrates advanced subsystems, including the tracker, calorimeters, and muon chambers, to provide comprehensive coverage and high precision. These capabilities enable CMS to explore the rich physics opportunities at the LHC.

Chapter 3

Dataset

3.1 Geant4 Simulation

Geant4 is a powerful and widely used simulation toolkit for modeling the passage of particles through matter. It provides detailed simulations of detector geometry, material interactions, and physics processes, enabling accurate predictions of detector responses. In the context of the CMS detector, Geant4 plays a critical role in validating experimental results and designing upgrades like the HGCal.

3.1.1 Physics Processes

Geant4 includes a comprehensive suite of physics processes covering electromagnetic, hadronic, and optical interactions. For the HGCal, electromagnetic processes such as ionization, bremsstrahlung, and photon interactions are particularly important in the CE-E section, while hadronic processes are crucial for modeling particle showers in the CE-H [46].

3.1.2 Geometry and Materials

Geant4 enables users to define complex and highly detailed detector geometries with exceptional precision and flexibility. Taking the High-Granularity Calorimeter (HGCal) as an example, the arrangement of silicon sensors, scintillator tiles, and absorber plates is accurately modeled in Geant4. Each component is defined in terms of its precise geometry and physical properties, including parameters such as density, radiation length, and interaction cross-sections.

Through Geant4, the HGCal geometry is meticulously constructed layer by layer. Silicon sensors, segmented into hexagonal cells, simulate active regions where particles interact to generate measurable signals. Absorber materials like lead and steel are defined to induce particle showers, while scintillator tiles are incorporated to detect the resulting secondary particles. This level of detail ensures that simulations replicate real-world interactions, providing reliable data for performance optimization and physics studies.

3.1.3 Applications in HGCal Development

Geant4 has been instrumental in optimizing the design of the HGCal. By simulating different configurations and material choices, researchers have fine-tuned the detector to achieve the desired performance in terms of energy resolution, granularity, and radiation tolerance. These simulations also help in developing reconstruction algorithms and calibrations tailored to the unique characteristics of the HGCal [46].

FIGURE 3.1: Visualization of a Geant4 simulation for the HGCal, showing particle showers in the calorimeter layers. (Image credit: Geant4 Collaboration)

Geant4 remains an indispensable tool in the development and operation of the CMS detector, enabling detailed studies of particle interactions and supporting advancements in high-energy physics.

3.2 The Fast Calorimeter Simulation Challenge (CaloChallenge)

The Fast Calorimeter Simulation Challenge, or CaloChallenge, is an initiative designed to advance the development of fast, accurate, and efficient generative models for calorimeter shower simulations. This challenge bridges the gap between traditional simulation methods like GEANT4 and novel machine learning approaches, providing datasets, benchmarks, and metrics for evaluation [**calochallenge**].

3.2.1 Objectives

CaloChallenge has the following primary goals:

- Encourage the development of generative models capable of fast and accurate calorimeter shower simulation.
- Provide standardized datasets and metrics for consistent evaluation and benchmarking.
- Foster collaboration across the high-energy physics and machine learning communities.

3.2.2 Datasets

The CaloChallenge offers three distinct datasets, each increasing in complexity, to evaluate model performance in diverse scenarios. The datasets are as follows:

Dataset 1: ATLAS GEANT4 Open Datasets

Dataset 1 is based on simulations using the ATLAS detector geometry. It includes two particle types: photons and charged pions. The voxelized shower information

is derived from single particles produced at the calorimeter surface in the η range of 0.2-0.25. The detector geometry consists of 5 layers for photons and 7 layers for pions, with the number of radial and angular bins varying by layer and particle type. This results in 368 voxels for photons and 533 voxels for pions.

The dataset spans 15 discrete incident energy levels, ranging from 256 MeV to 4 TeV in powers of two. Each energy level contains 10k events, except for the higher energies, where fewer events are available due to statistical limitations. This dataset serves as a benchmark for evaluating generative models on simpler detector geometries and energy distributions.

Dataset 2: Multi-Layer Geometry with Electrons

Dataset 2 focuses on simulations of electrons interacting with a concentric cylindrical detector geometry. The detector comprises 45 layers, each with both active (silicon) and passive (tungsten) material. Each layer is divided into 9 radial bins and 16 angular bins, resulting in a total of 6480 voxels ($45 \times 16 \times 9$).

The electron energies are sampled from a log-uniform distribution ranging from 1 GeV to 1 TeV, offering a continuous spectrum of energy levels. The dataset contains 100k events, enabling models to explore and learn intricate energy depositions across the detector geometry. This dataset challenges models to handle the complexities of high-granularity detectors.

Dataset 3: High-Granularity Calorimeter Geometry

Dataset 3 simulates a high-granularity calorimeter with an advanced detector geometry. Like Dataset 2, it features 45 layers with active (silicon) and passive (tungsten) material. However, the granularity is significantly higher, with each layer containing 18 radial bins and 50 angular bins. This results in a total of 40,500 voxels ($45 \times 50 \times 18$).

The dataset consists of electron showers with energies sampled from a log-uniform distribution ranging from 1 GeV to 1 TeV. Each file contains 50k events, offering robust training and evaluation datasets. This dataset is designed to test models' ability to generalize and simulate realistic particle physics scenarios with highly detailed detector geometries.

3.2.3 Data Format

Each dataset is stored as one or more HDF5 files created using Python's `h5py` module with gzip compression. The files include:

- `incident_energies`: An array of shape (`num_events`, 1) containing the incoming particle energies in MeV.
- `showers`: An array of shape (`num_events`, `num_voxels`) storing the energy depositions (in MeV) for each voxel, flattened in a specific order.

The mapping of voxel indices to spatial coordinates follows the detector segmentation. Helper functions are provided for reshaping and handling the data.

3.2.4 Evaluation Metrics

CaloChallenge evaluates the generative models using multiple metrics, including:

- A binary classifier trained to distinguish between real GEANT4 samples and model-generated samples.
- Chi-squared comparisons between histograms of high-level features, such as layer energies and shower shapes.
- Speed and resource usage metrics, such as training time, generation time, and memory footprint.
- Interpolation capabilities to test generalization across unseen particle energies.

3.2.5 Community Engagement

Participants are encouraged to share their findings and contribute to community discussions. The challenge concludes with a workshop to present results, compare approaches, and collaborate on a community paper documenting the outcomes. For communication and updates, participants can join the ML4Jets Slack channel and the Google Groups mailing list [**calochallenge**].

For further details, visit the official CaloChallenge GitHub repository: `https://github.com/CaloChallenge/homepage`.

Chapter 4

Algorithm

4.1 AE

4.2 VAE

test

4.3 Score-based Diffusion Model

One drawback of Variational Autoencoders (VAEs) is the inclusion of the KL divergence term in their loss function. While VAEs are effective for compressing data (encoding), they struggle to generate high-quality, diverse samples. This limitation stems from their reliance on sampling from a normal distribution in the latent space. Although VAEs are trained to bring the posterior distribution close to a Gaussian, in practice, the match is often not precise enough to ensure that samples drawn from this distribution will be of high quality.

Therefore, an alternative approach, introduced in 2015, is the “diffusion model,” which can be implemented using either score-matching or denoising techniques. Diffusion models aim to generate synthetic data based on a set of independent, identically distributed (i.i.d.) samples drawn from an unknown data distribution. The key concept is to simulate new samples by either employing denoising Score Langevin Dynamics (SMLD) or implementing Denoising Diffusion Probabilistic Model (DDPM), where a deep neural network approximates the score, or gradient, of the log-density of the data distribution. Next we will discuss the two methods in detail.

4.3.1 Denoising Score Matching with Langevin Dynamics (SMLD)

Langevin Dynamics in generative modeling is a way to generate samples by simulating a process that gradually moves from random points in space toward areas with high probability density, where most of the real data is located. It does this by changing along the directions defined by the gradient of the probability distribution, called the "score" in our context. At each step, a small amount of Gaussian noise is added

to introduce randomness, ensuring that each path taken is unique and prevents the sampling process from getting "stuck" in local regions.

In simpler terms, think of Langevin Dynamics as a guided walk starting from a random spot and following a path that gradually leads toward more typical or likely values of the data (like images, text, etc.). The direction of each step is influenced by both the data structure (moving toward areas where data is dense) and a bit of noise to keep things varied, which helps to explore the whole space more effectively. This makes Langevin Dynamics an effective sampling method for creating new data points in generative modeling.

In this approach, we define a perturbation mechanism $p_\sigma(\tilde{x}|x) = \mathcal{N}(\tilde{x}; x, \sigma^2 I)$, which acts as a Gaussian kernel centered at x with variance σ^2. This perturbation is integrated over the data distribution $p_{\text{data}}(x)$ to yield the broader distribution $p_\sigma(\tilde{x}) = \int p_{\text{data}}(x) p_\sigma(\tilde{x}|x)\, dx$.

We consider a range of increasing noise scales, where $\sigma_{\min} = \sigma_1 < \sigma_2 < \cdots < \sigma_N = \sigma_{\max}$. Typically, $\sigma_{\min}$ is chosen to be small enough that $p_{\sigma_{\min}}(x) \approx p_{\text{data}}(x)$, capturing the original data distribution, while $\sigma_{\max}$ is set large enough so that $p_{\sigma_{\max}}(x) \approx \mathcal{N}(x; 0, \sigma_{\max}^2 I)$, resembling a Gaussian prior.

Following the work of Song and Ermon [47], we train a Noise Conditional Score Network (NCSN), denoted $s_\theta(x, \sigma)$, by minimizing a weighted sum of denoising score matching objectives as follows:

$$\theta^* = \arg\min_\theta \sum_{i=1}^{N} \sigma_i^2 \, \mathbb{E}_{p_{\text{data}}(x)} \, \mathbb{E}_{p_{\sigma_i}(\tilde{x}|x)} \left[\| s_\theta(\tilde{x}, \sigma_i) - \nabla_{\tilde{x}} \log p_{\sigma_i}(\tilde{x}|x) \|_2^2 \right]. \tag{4.1}$$

Given sufficient data and model capacity, the resulting score-based model $s_\theta^*(x, \sigma)$ estimates the gradient $\nabla_x \log p_\sigma(x)$ across noise scales $\sigma \in \{\sigma_i\}_{i=1}^N$.

So the Langevin Dynamics process can be described as follows:

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2}\mathbf{z}_{i-1}, i = 1, 2, \ldots, N \tag{4.2}$$

4.3.2 Denoising Diffusion Probabilistic Model (DDPM)

Next, we are going to introduce the second method for diffusing models, the Denoising Diffusion Probabilistic Model (DDPM) [48]. Unlike SMLD, DDPM incorporates a scaling factor for x, which modifies the approach slightly. The basic idea is to define the conditional probability distribution as follows: $p(x_i|x_{i-1}) = \mathcal{N}\left(x_i; \sqrt{1-\beta_i} x_{i-1}, \beta_i I\right)$.

Following Sohl-Dickstein et al. (2015) and Ho et al. (2020), let us consider a set of positive noise scales $0 < \beta_1, \beta_2, \ldots, \beta_N < 1$. For each data point $x_0 \sim p_{\text{data}}(x)$, we define a discrete Markov chain $\{x_0, x_1, \ldots, x_N\}$, with each transition given by $p(x_i|x_{i-1}) = \mathcal{N}\left(x_i; \sqrt{1-\beta_i}x_{i-1}, \beta_i I\right)$. Consequently, we can write the marginal distribution $p_{\alpha_i}(x_i|x_0) = \mathcal{N}\left(x_i; \sqrt{\alpha_i}x_0, (1-\alpha_i)I\right)$, where $\alpha_i := \prod_{j=1}^{i}(1-\beta_j)$.

As in SMLD, we also train it by minimizing the denoising score matching objective:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} (1-\alpha_i)\, \mathbb{E}_{p_{\text{data}}(x)}\, \mathbb{E}_{p_{\alpha_i}(\tilde{x}|x)} \left[\| s_\theta(\tilde{x}, \alpha_i) - \nabla_{\tilde{x}} \log p_{\alpha_i}(\tilde{x}|x) \|_2^2 \right]. \tag{4.3}$$

where again, $1-\alpha_i$ is just a weighting factor.

What's more, we can define the perturbed data distribution as $p_{\alpha_i}(\tilde{x}) := \int p_{\text{data}}(x) p_{\alpha_i}(\tilde{x}|x) dx$. The noise scales are chosen so that x_N approximates a standard normal distribution $\mathcal{N}(0, I)$. So the simialr form as SMLD will be

$$x_{t-1} = \sqrt{1-\beta_t}x_t + \sqrt{\beta_t}z_t, \quad t = N, N-1, \ldots, 1 \tag{4.4}$$

where $z_t \sim \mathcal{N}(0, I)$ are standard normal samples. The final sample x_0 is drawn from the data distribution $p_{\text{data}}(x)$. The process is repeated for each data point, and the final samples are generated by running the Markov chain for T steps. The resulting samples are expected to approximate the data distribution $p_{\text{data}}(x)$ when $T \to \infty$ under suitable conditions.

4.4 Forward Process

So far, we have discussed two ways of simulating new samples from a given data distribution. Although they look different, both methods are based on the same principle: iteratively transforming a sample from a simple distribution (e.g., a Gaussian) to a more complex one (e.g., the data distribution).

Based on the work of Yang Song [47], we can generalize this concept through what is called the **forward process** in diffusion models.

Our goal is to construct a diffusion process x_t indexed by a continuous time variable $t \in [0, T]$, such that:

$$x_0 \sim p_0 \tag{4.5}$$

for which we have a dataset of independent and identically distributed (i.i.d.) samples, and

$$x_T \sim p_T \tag{4.6}$$

for which we have a tractable form to generate samples efficiently. In other words, p_0 is the data distribution and p_T is the prior distribution.

This diffusion process can be modeled as the solution to an Itô stochastic differential equation (SDE):

$$dx = f(x,t)\,dt + g(t)\,dw \tag{4.7}$$

where:

- x is the state variable,
- $f(x,t)$ is the drift coefficient,
- $g(t)$ is the diffusion coefficient,
- w is a Wiener process (Brownian motion).

For later we can show that this has a slightly better result than orginal DDPM and SMLD.

4.5 Backward Process

With the forward process established, we can now construct the **backward process**. The aim of this process is to generate samples from the data distribution p_0, given samples from the prior distribution p_T.

The continuous form of this process is defined by the following stochastic differential equation (SDE):

$$d\mathbf{x} = \mathbf{f}_t(\mathbf{x})\,dt + g_t\,d\mathbf{w} \tag{4.8}$$

To direclty prove the reverse SDE formula in continuous form will be a little complex. But we can get the same spirit from discrete form, as $\Delta t \to 0$, the continuous equation above can be approximated by:

$$\mathbf{x}_{t+\Delta t} - \mathbf{x}_t = \mathbf{f}_t(\mathbf{x}_t)\,\Delta t + g_t\sqrt{\Delta t}\,\varepsilon, \quad \varepsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}) \tag{4.9}$$

The discrete form of the stochastic differential equation (SDE) is especially valuable for practical computer implementations. By breaking down the continuous process into discrete steps, we can simulate both the diffusion and reverse processes incrementally, allowing us to generate samples using numerical methods. This approach enables us to approximate continuous dynamics with a series of discrete updates, making the computations more manageable and efficient.

In this way, using the SDE framework to describe diffusion models provides a clear distinction between theoretical analysis and practical implementation. We can rely on the mathematical properties of continuous SDEs for analysis, while in actual applications, we have the flexibility to choose any appropriate discretization method for efficient numerical calculation.

In probabilistic terms, Equation (4.9) implies that the conditional probability is given by

$$\begin{aligned} p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t) &= \mathcal{N}\left(\mathbf{x}_{t+\Delta t}; \mathbf{x}_t + \mathbf{f}_t(\mathbf{x}_t)\,\Delta t, g_t^2\,\Delta t\,\mathbf{I}\right) \\ &\propto \exp\left(-\frac{\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \mathbf{f}_t(\mathbf{x}_t)\,\Delta t\|^2}{2g_t^2\Delta t}\right) \end{aligned} \tag{4.10}$$

Now since our goal is to use the forward process to derive the backward process, which means obtaining $p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t})$, we apply Bayes' theorem, as shown in "A Discussion on Generative Diffusion Models: DDPM = Bayesian + Denoising":

$$\begin{aligned} p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) &= \frac{p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)p(\mathbf{x}_t)}{p(\mathbf{x}_{t+\Delta t})} \\ &= p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)\exp\left(\log p(\mathbf{x}_t) - \log p(\mathbf{x}_{t+\Delta t})\right) \\ &\propto \exp\left(-\frac{\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \mathbf{f}_t(\mathbf{x}_t)\Delta t\|^2}{2g_t^2\Delta t} + \log p(\mathbf{x}_t) - \log p(\mathbf{x}_{t+\Delta t})\right) \end{aligned} \tag{4.11}$$

It is not difficult to see that when Δt is sufficiently small, $p(\mathbf{x}_{t+\Delta t}|\mathbf{x}_t)$ will be significantly non-zero only when $\mathbf{x}_{t+\Delta t}$ is close to $\mathbf{x}_t$. Conversely, only in this case will $p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t})$ be significantly non-zero. Therefore, we only need to conduct an approximate analysis for situations where $\mathbf{x}_{t+\Delta t}$ is close to $\mathbf{x}_t$. For this, we can use a Taylor expansion:

$$\log p(\mathbf{x}_{t+\Delta t}) \approx \log p(\mathbf{x}_t) + (\mathbf{x}_{t+\Delta t} - \mathbf{x}_t)\cdot\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t) + \Delta t\,\frac{\partial}{\partial t}\log p(\mathbf{x}_t) \tag{4.12}$$

It is important not to neglect the term $\frac{\partial}{\partial t}$, because $p(\mathbf{x}_t)$ is a function of both t and $\mathbf{x}_t$, while $p(\mathbf{x}_{t+\Delta t})$ is a function of $t+\Delta t$ and $\mathbf{x}_{t+\Delta t}$. Thus, $p(\mathbf{x}_t)$ must include an additional time derivative. Substituting this into Equation (4.11) allows us to derive:

$$p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) \propto \exp\left(-\frac{\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \left[\mathbf{f}_t(\mathbf{x}_t) - g_t^2\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t)\right]\Delta t\|^2}{2g_t^2\Delta t} + \mathcal{O}(\Delta t)\right) \tag{4.13}$$

As $\Delta t \to 0$, the term $\mathcal{O}(\Delta t)$ becomes negligible, thus:

$$p(\mathbf{x}_t|\mathbf{x}_{t+\Delta t}) \propto \exp\left(-\frac{\|\mathbf{x}_{t+\Delta t} - \mathbf{x}_t - \left[\mathbf{f}_t(\mathbf{x}_t) - g_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right] \Delta t\|^2}{2g_t^2 \Delta t}\right) \tag{4.14}$$

Finally, the above formula indicates that the reverse process also contains a deterministic part and a stochastic part. The deterministic part consists of $\mathbf{f}_t(\mathbf{x}_t) - g_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$, while the stochastic part is $g_t\sqrt{\Delta t}\varepsilon$.

Thus, our reverse process is defined as:

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t - \left[\mathbf{f}_t(\mathbf{x}_t) - g_t^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)\right] \Delta t + g_t\sqrt{\Delta t}\varepsilon \tag{4.15}$$

We can use the picture below to illustrate the forward and backward processes in a diffusion model:

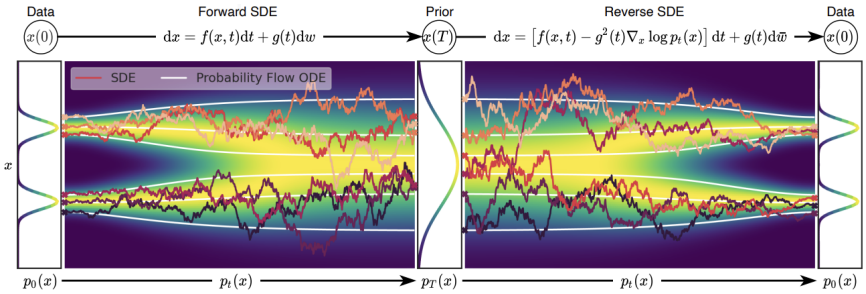


FIGURE 4.1: Forward and Backward Processes in Diffusion Models (The picture is from Song and Ermon (2019))

4.6 VE, VP SDEs

4.6.1 Continuos Forward Process

After we established the general form of the forward and backward processes, we can now go back to see how to apply them on SMLD (VE mthoed) and DDPM (VP method).

So in this section, we try to present detailed derivations demonstrating that the noise perturbations in SMLD (Score-based generative modeling via Langevin Dynamics) and DDPM (Denoising Diffusion Probabilistic Models) are discretizations of the Variance Exploding (VE) and Variance Preserving (VP) Stochastic Differential Equations (SDEs), respectively. We also introduce sub-VP SDEs, which are modifications of VP SDEs that often yield improved performance in terms of sample quality and likelihoods.

First, when utilizing a total of N noise scales, each perturbation kernel $p_{\sigma_i}(x|x_0)$ for SMLD can be derived from the following Markov chain:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, 2, \ldots, N, \tag{4.16}$$

where $z_{i-1} \sim \mathcal{N}(0, I)$ and $x_0 \sim p_{\text{data}}$. Here, we introduce $\sigma_0 = 0$ for simplicity. As $N \to \infty$, the Markov chain $\{x_i\}_{i=1}^N$ converges to a continuous stochastic process $\{x(t)\}_{t=0}^1$, and $\{\sigma_i\}_{i=1}^N$ becomes a function $\sigma(t)$, while z_i transitions to $z(t)$. We denote the continuous time variable as $t \in [0, 1]$ instead of the integer index $i \in \{1, 2, \ldots, N\}$. Let $\mathbf{x}(\frac{i}{N}) = \mathbf{x}_i$, $\sigma(\frac{i}{N}) = \sigma_i$, $\mathbf{z}(\frac{i}{N}) = \mathbf{z}_i$, for $i = 1, 2, \ldots, N$. Rewriting Equation (20) with $\Delta t = \frac{1}{N}$ gives:

$$x(t + \Delta t) = x(t) + \sqrt{\sigma^2(t + \Delta t) - \sigma^2(t)} z(t) \approx x(t) + \sqrt{\frac{d\sigma^2(t)}{dt}} \Delta t z(t), \tag{4.17}$$

where the approximation holds as $\Delta t \to 0$. In the limit of $\Delta t \to 0$, we obtain the VE SDE:

$$dx = \sqrt{\frac{d\sigma^2(t)}{dt}} dw. \tag{4.18}$$

Furthermore, we usually let σ sequence to be a geometric sequence. We have $\sigma(\frac{i}{N}) = \sigma_i = \sigma_{\min}(\frac{\sigma_{\max}}{\sigma_{\min}})^{\frac{i-1}{N-1}}$ for i ranges from 1 to N. If $N \to \infty$

The corresponding VE SDE is

$$d\mathbf{x} = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t \sqrt{2 \log \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)} d\mathbf{w}, \quad t \in (0, 1). \tag{4.19}$$

For the perturbation kernels $p_{\alpha_i}(x|x_0)$ used in DDPM, the discrete Markov chain is given by:

$$\mathbf{x}_i = \sqrt{1 - \beta_i} \mathbf{x}_{i-1} + \sqrt{1 - \beta_i} z_{i-1}, \quad i = 1, 2, \ldots, N, \tag{4.20}$$

where $z_{i-1} \sim \mathcal{N}(0, I)$. To obtain the limit of this Markov chain as $N \to \infty$, we define an auxiliary set of noise scales $\{\bar{\beta}_i\}_{i=1}^N$ and rewrite Equation (22) as follows:

$$\mathbf{x}_i = \sqrt{1 - \bar{\beta}_i} \mathbf{x}_{i-1} + \sqrt{1 - \bar{\beta}_i} z_{i-1}, \quad i = 1, 2, \ldots, N, \tag{4.21}$$

As $N \to \infty$, the noise scales $\{\bar{\beta}_i\}_{i=1}^N$ converge to a function $\beta(t)$ indexed by $t \in [0, 1]$. Let $\{\bar{\beta}_i\}_N = \beta$ and $\{x_i\}_N = x$ and $\{z_i\}_N = z$. Rewriting Equation (23) gives:

$$\begin{aligned}\mathbf{x}(t+\Delta t) &= \sqrt{1-\beta(t+\Delta t)}\mathbf{x}(t) + \sqrt{1-\beta(t+\Delta t)}z(t) \\ &\approx \mathbf{x}(t) - \frac{1}{2}\beta(t+\Delta t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\ &\approx \mathbf{x}(t) - \frac{1}{2}\beta(t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t)\Delta t}\mathbf{z}(t)\end{aligned} \tag{4.22}$$

where the approximation holds as $\Delta t \to 0$. Therefore, in the limit of $\Delta t \to 0$, we obtain the VP SDE:

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,dt + \sqrt{\beta(t)}\,d\mathbf{w}. \tag{4.23}$$

As in DDPM, β is typically an arithmetic sequence where $\beta_i = \beta_{\min} + t(\beta_{\max} - \beta_{\min})$ for t ranges from 0 to 1 if $N \to \infty$. This will then give us the VP SDE as

$$d\mathbf{x} = -\frac{1}{2}\left(\beta_{\min} + t(\beta_{\max} - \beta_{\min})\right)\mathbf{x}\,dt + \sqrt{\beta_{\min} + t(\beta_{\max} - \beta_{\min})}\,d\mathbf{w}, \quad t \in (0,1). \tag{4.24}$$

In conclusion, the contents above indicate that

For **SMLD (Variance Exploding SDE - VE)**:

- $f(x,t) = 0$
- $g(t) = \sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t \sqrt{2\log\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)}$

For **DDPM (Variance Preserving SDE - VP)**:

- $f(x,t) = -\frac{1}{2}\beta(t)x$
- $g(t) = \sqrt{\beta(t)}$

4.6.2 Continuos Backward Process - PC Sampler

Here we can of course use the $f(x,t)$ and $g(t)$ to do the reverse process as equation (4.15) shows. However, here we possess additional insights that can enhance our solution methods. Specifically, with our score-based model $s_{\theta^*}(x,t) \approx \nabla_x \log p_t(x)$, we can utilize score-based Markov Chain Monte Carlo (MCMC) techniques, such as Langevin MCMC (Parisi, 1981; Grenander & Miller, 1994)to sample directly from the distribution p_t and refine the outputs of a numerical SDE solver.

At each time step, the numerical SDE solver provides an initial estimate for the sample at the next time step, functioning as a "predictor." Subsequently, the score-based MCMC method adjusts the estimated sample's marginal distribution, acting as

a "corrector." This approach is reminiscent of Predictor-Corrector methods, which are a class of numerical continuation techniques used for solving systems of equations (Allgower & Georg, 2012). We similarly refer to our hybrid sampling algorithms as Predictor-Corrector (PC) samplers.

The PC samplers extend the original sampling methodologies of SMLD and DDPM: the SMLD method employs an identity function as the predictor and utilizes annealed Langevin dynamics as the corrector. In contrast, the DDPM method adopts ancestral sampling as the predictor and the identity function as the corrector.

Algorithm 1 *PC sampling (VE SDE)*

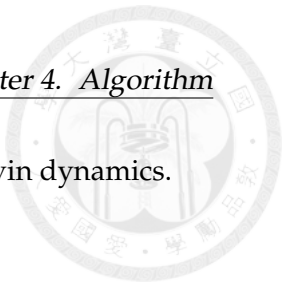
$\mathbf{x}_N \sim \mathcal{N}(0, \sigma_{\max}^2 \mathbf{I})$
for $i = N-1$ to 0 **do**
  $\mathbf{x}_i' = \mathbf{x}_i - g^2(t) s_\theta^*(\mathbf{x}_i, \sigma_i)\,\Delta t$
  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{x}_i = \mathbf{x}_i' + g(t)\sqrt{\Delta t}\mathbf{z}$
  for $j = 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i s_\theta^*(\mathbf{x}_i, \sigma_i) + \sqrt{2\epsilon_i}\mathbf{z}$
  end for
end for
return $\mathbf{x}_0$

Algorithm 2 *PC sampling (VP SDE)*

$\mathbf{x}_N \sim \mathcal{N}(0, \mathbf{I})$
for $i = N-1$ to 0 **do**
  $\mathbf{x}_i' = (f(x,t) - g^2(t) * s_\theta^*(\mathbf{x}_{i+1}, i+1))\Delta t$
  $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
  $\mathbf{x}_i = \mathbf{x}_i' + g(t)\sqrt{\Delta t}\mathbf{z}$
  for $j = 1$ to M **do**
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i s_\theta^*(\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$
  end for
end for
return $\mathbf{x}_0$

where ϵ is defined as

$$\epsilon = 2r^2 \frac{\|\mathbf{z}\|_2^2}{\|s_\theta\|_2^2} \tag{4.25}$$



and r is a hyperparameter that controls the step size of the Langevin dynamics.

4.7 Conclusion

Chapter 5

Model Structure

In the previous chapter, we introduced the foundational algorithms employed in this research project. This chapter delves into the structure of our custom Transformer-based model, designed to predict the "score" or gradient in detector simulations. Built upon the Transformer architecture—a cutting-edge model in deep learning—our model incorporates several modifications to enhance its applicability in high-energy physics detector simulations.

We chose the Transformer architecture not only for its power and versatility but also for its unique suitability for data with rotational invariance. In our research, each input consists of multiple showers, each shower containing several hits, and each hit represented by four features, as introduced in Chapter 3. This structure makes our data rotationally invariant, meaning that the relationships within the data remain consistent even if the order of hits within a shower or the showers within an input are rearranged. Transformers are particularly well-suited for handling such properties. Their self-attention mechanism enables them to learn and capture relationships between data points in a way that is invariant to transformations like rotation. This flexibility is especially advantageous for our detector simulations, where capturing invariant relationships is crucial for making accurate predictions.

We will begin by exploring the evolution of Transformers from Recurrent Neural Networks (RNNs), highlighting how Transformer architectures overcame the limitations of sequential models. Following this, we will examine the core components of the Transformer model, including its different architectural types (encoder-only, decoder-only, and encoder-decoder models) and the self-attention mechanism, which lies at the heart of the Transformer's ability to model long-range dependencies.

After establishing an understanding of the original Transformer architecture, we

will discuss the custom modifications introduced in our model to optimize it for detector simulations. Key innovations include the **Gaussian Fourier Projection** for encoding temporal information, which allows the model to capture high-frequency dependencies by transforming time and incident energy into sinusoidal features. Additionally, we introduce a specialized **mean-field attention mechanism**, a variant of self-attention tailored to efficiently aggregate global context. Mean-field attention leverages a class token to summarize information across the sequence, reducing computational complexity while retaining essential global information.

Furthermore, our model incorporates residual network structures and layer normalization to stabilize and expedite the training process. We will explain how these modifications, along with our encoder-only architecture, facilitate efficient information flow, enabling the model to focus on capturing the relationships within the data rather than generating sequences. We also employ **Weights & Biases (wandb)** for parameter tuning, using its sweep functionality to systematically explore hyperparameters such as the number of encoder blocks, attention heads, and dropout rates to achieve optimal performance.

In summary, this chapter provides a comprehensive overview of our custom Transformer model, from its foundational components to the innovative adjustments that make it well-suited for high-energy physics applications. Through these design choices, our model efficiently captures both local and global dependencies, thereby enhancing the accuracy and fidelity of detector simulations.

5.1 Transformer

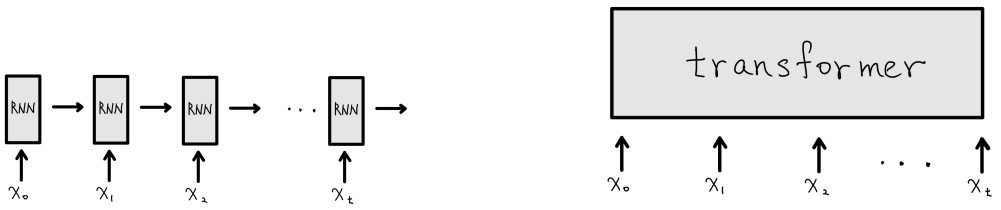
5.1.1 Introduction

Transformer models have transformed deep learning applications across various domains, providing significant advantages in handling complex and large datasets. In high-energy physics, where data from detectors is vast and multidimensional, advanced models like Transformers enhance both the accuracy and efficiency of simulations designed to emulate particle collisions and energy depositions within detectors.

5.1.2 The Evolution from RNNs to Transformers

Prior to Transformers, Recurrent Neural Networks (RNNs) were widely used for sequence modeling due to their ability to capture dependencies between sequential elements. However, RNNs are inherently sequential, making them slow and prone to issues like vanishing gradients, particularly with long sequences.

The introduction of Transformers by Vaswani et al. in their seminal paper, *"Attention is All You Need,"* addressed these limitations by introducing the self-attention mechanism. This innovation enables Transformers to capture long-range dependencies without the need for recurrent connections, allowing for faster and more efficient training.



(A) RNN Model (B) Transformer Model

FIGURE 5.1: Comparison of RNN and Transformer architectures.

5.1.3 Types and Structure of Transformer Architectures

The original Transformer architecture, as introduced by Vaswani et al., consists of both an encoder and a decoder. The encoder processes the input sequence, while the decoder generates the output sequence. This setup is particularly effective for tasks like machine translation. However, in practice, different applications benefit from using only the encoder or decoder.

The three main types of Transformer architectures are as follows:

- **Encoder-only Models**: Encoder-only models, such as BERT, create contextual embeddings by attending to all tokens bidirectionally. These models are ideal for tasks requiring sequence understanding, like classification.
- **Decoder-only Models**: Decoder-only models, like GPT, are designed for unidirectional generation. Each token attends only to previous tokens, making these models suitable for tasks like language modeling.
- **Encoder-Decoder Models**: The original Transformer model combines both an encoder and a decoder, making it effective for sequence-to-sequence tasks such as machine translation. Examples include BART and T5.

5.1.4 Choosing an Encoder-Only Model for Detector Simulation

In the context of detector simulation, our objective is to generate a high-quality representation of input data, such as particle collisions, without generating new sequences.

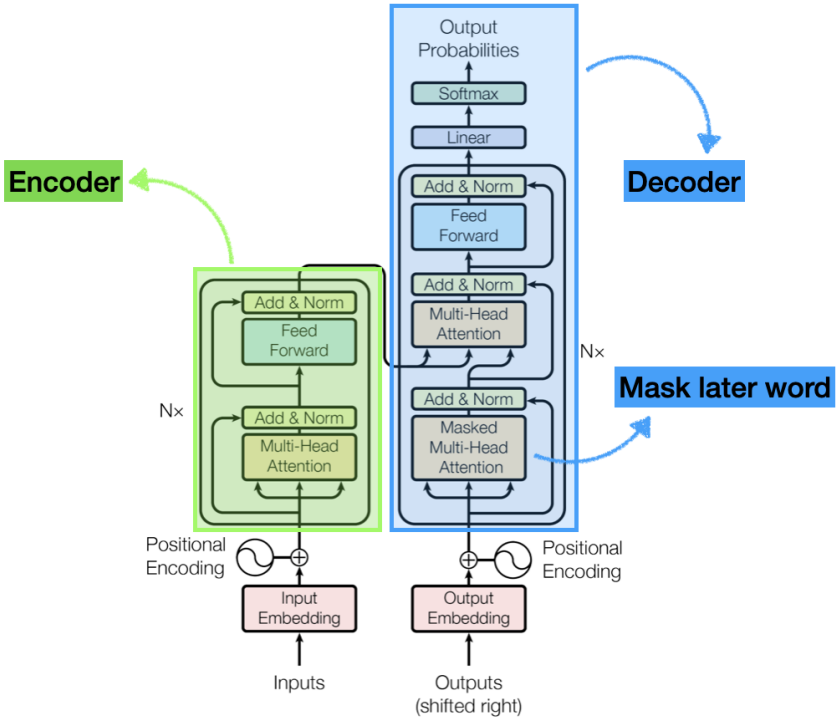


FIGURE 5.2: The structure of the original Transformer model. Adapted from *"Attention is All You Need,"* with additional annotations.

Thus, an encoder-only model is more appropriate, as it efficiently processes and encodes input data, capturing necessary features without the additional complexity of a generative decoder.

5.2 Self-Attention Mechanism

Self-attention is central to the Transformer model, enabling each token in a sequence to attend to all other tokens. For each token, the attention scores are computed based on the query Q, key K, and value V vectors:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{5.1}$$

This mechanism allows the model to capture relationships between distant tokens, which is crucial for high-dimensional data, such as particle detector simulations. Figure ?? illustrates the self-attention mechanism, showcasing how each token dynamically weights other tokens in the sequence.

5.3 Our Model Structure

Our model architecture builds upon the Transformer framework, with specific modifications to optimize performance in detector simulations, as shown in Figure .

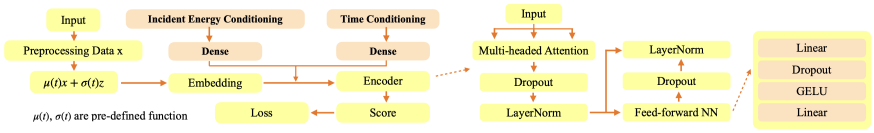


FIGURE 5.3: Custom Transformer model structure for detector simulations.

We incorporate **Gaussian Fourier Projections** [49] to effectively encode temporal information, dense layers to transform conditional variables, and **mean-field attention** [50] to efficiently aggregate global context. These architectural choices enable our model to capture complex dependencies, thereby enhancing the fidelity and accuracy of simulation outcomes.

5.3.1 Gaussian Fourier Projection for Temporal Encoding

The Gaussian Fourier Projection component encodes temporal information using Gaussian random features. This technique allows the model to incorporate high-frequency time-dependent information, in our case time and incident energy, which is crucial for capturing the dynamics of particle interactions within detectors.

In our model, we apply a Fourier feature mapping γ to featurize input coordinates before passing them through a coordinate-based multilayer perceptron (MLP). This approach improves both convergence speed and generalization.

The mapping function γ transforms input points $\mathbf{v} \in [0,1)^d$ onto the surface of a higher-dimensional hypersphere using sinusoidal functions:

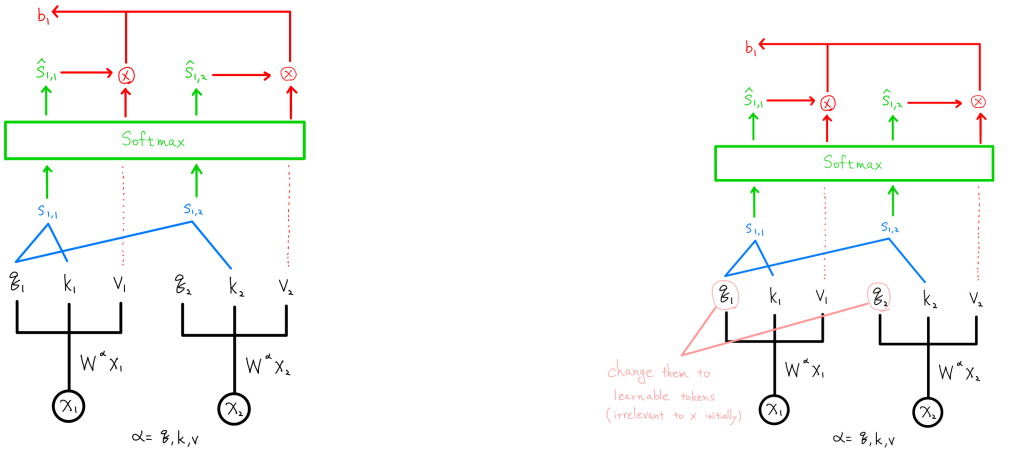
$$\gamma(\mathbf{v}) = \begin{bmatrix} a_1 \cos(2\pi \mathbf{b}_1^T \mathbf{v}) \\ a_1 \sin(2\pi \mathbf{b}_1^T \mathbf{v}) \\ \vdots \\ a_m \cos(2\pi \mathbf{b}_m^T \mathbf{v}) \\ a_m \sin(2\pi \mathbf{b}_m^T \mathbf{v}) \end{bmatrix} \tag{5.2}$$

where a_i and $\mathbf{b}_i$ are parameters that control the scaling and frequency of each sinusoid. We set $a = 1$ for all cases and experiment with different values of $\mathbf{b}$ to identify optimal performance. The results are presented in subsequent sections.

5.3.2 Mean-Field Attention in Detector Simulation

Our model utilizes a variation of self-attention called **mean-field attention**. Unlike traditional self-attention, mean-field attention employs a class token to aggregate information from all tokens, creating a global summary. This reduces computational complexity while preserving essential global context.

Mean-field attention allows the class token to encapsulate the sequence's essential features by attending to each token once. This mechanism is computationally efficient and well-suited for high-energy physics applications, where capturing global properties of particle collisions is more important than individual token interactions. Figure ?? provides a comparison between self-attention and mean-field attention mechanisms.



(A) Self-Attention Mechanism (B) Mean-Field Attention Mechanism

FIGURE 5.4: Comparison of self-attention and mean-field attention mechanisms.

5.3.3 Parameter Tuning

Once the model structure is established, tuning the parameters is essential for optimal performance.

To optimize the model, we utilize **Weights & Biases (wandb)** for parameter tuning. Using wandb's sweep functionality, we systematically explore hyperparameters, including:

- **Number of blocks**: Controls the depth of the Transformer model.
- **Number of heads**: Determines the number of attention heads in the multi-head attention mechanism.
- **Hidden dimension**: Sets the size of the hidden layers in the model.

- **Embed dimension**: Specifies the embedding size for the model's input.
- **Batch size**: Number of samples per batch.
- **Learning rate**: Determines the rate at which the model updates during training.
- **Dropout rate**: The fraction of nodes dropped during training to prevent overfitting.
- **Sampling steps**: The number of sampling steps when solving SDE.
- **Correction steps**: The number of correction steps in each sampling step.
- **Scale in Fourier features**: The scaling factor for Fourier features.

The results of this tuning process are presented in later chapters.

5.4 Conclusion

Our custom Transformer model leverages specialized architectural choices to optimize performance in high-energy physics simulations. Key modifications include **Gaussian Fourier projections** for encoding time and incident energy, and **mean-field attention** for capturing global context beyond immediate shower information. The addition of a class token enables the model to represent both local and global dependencies, making it particularly suitable for scenarios with strong temporal and energetic relationships.

The mean-field attention mechanism enhances computational efficiency by reducing complexity while preserving essential global information. Parameter tuning plays a crucial role in achieving optimal performance, as we demonstrate with our use of wandb. By employing an encoder-only model, we capture inter-token relationships within the data, making our approach well-suited for high-energy physics applications.

Chapter 6

Strategies and Results

6.1 Data Preprocessing

6.1.1 Bucketing

Before we explain why we need bucketing, we can first explain the structure of our data. When one particle interacts with the detector, it will produce a series of hits, which we call one shower. So in one shower, we have several hits, while one hit means one point in the detector labeled by the energy. One hit has several features, such as the hit energy, x, y and z coordinates. What's more, we will send several showers make it to be one batch to our model. So the structure of our data is actually a 3D tensor, where the first dimension is the number of showers, the second dimension is the number of hits in one shower, and the third dimension is the number of features in one hit.

In chapter 5, we have discussed that our model is a transformer-based model. While transformer implement the self-attention mechanism, it requires the length of the sequence to be fixed in each batches. However, the number of hits in each event varies, which makes it difficult to feed the data into the transformer. To address this issue, we would need to pad the sequences to a fixed length. What's more, the length of the data can vary from 1 to 5500, which means that the padding will be very large. This will lead to a waste of memory and computation. To solve this problem, we employed a bucketing strategy to group events with similar numbers of hits into the same bucket. This allowed us to pad the sequences within each bucket to a fixed length, making it easier to feed the data into the transformer. Based on the principle of similar memory usage, we divided the data into 45 buckets, each containing events with a similar number of hits. This bucketing strategy significantly improved the efficiency of the model and reduced the computational burden. Another advantage of bucketing is that we can first train the model on a smaller bucket to see if the model can learn the data well. If the model can learn the data well, we can then train the model on a larger bucket. This allows us to gradually increase the complexity of the data and ensure that the model can handle the data effectively.

6.1.2 Preprocessor

Preprocessing is a crucial step in preparing data for machine learning. Raw data often contains missing values, outliers, and features on different scales, which can negatively impact model performance. Effective preprocessing cleans and standardizes the data, ensuring consistency and enabling accurate predictions. It also helps models learn specific relationships between features more effectively.

A key role of preprocessing is improving data quality. Techniques like imputation, normalization, and outlier removal address missing or noisy values, allowing models to focus on meaningful patterns rather than irrelevant or erroneous information. Preprocessing also standardizes feature scales, ensuring equal contributions to models, which is especially critical for distance-based algorithms like neural networks or support vector machines.

Additionally, preprocessing optimizes computational efficiency by simplifying data complexity through methods like dimensionality reduction or sampling. This is vital for large-scale datasets, enabling faster and more resource-efficient training while preserving essential information. Overall, preprocessing is foundational for reliable, robust machine learning systems.

One important point to note is that we chose to use the x,y coordinate system instead of the cylindrical coordinate system. The primary reason for this choice is the discontinuity at $\theta = 0$ and $\theta = 2\pi$, which is unphysical and can introduce challenges during training. Although the cylindrical coordinate system aligns better with the detector structure and may simplify learning the relationship between radius and energy, we opted for the x,y coordinate system to ensure continuity and avoid such complications.

From the reasons above, we employ three different data preprocessing techniques for detector hit information: **RobustScaler**, **QuantileTransformer**, and **Exponential Transformation**. While the the comparison between three methods focus on transforming the x and y coordinates, the energy and z coordinate are processed using the same methodology across all three approaches. This consistent treatment of energy and z coordinates allows for a direct comparison of the methods and highlights the benefits of the different transformations applied to x and y.

Energy Transformation

The energy transformation applies a **logit-based rescaling**, ensuring numerical stability and normalization. Given the raw hit energy e, the transformation is defined as:

$$e' = \log\left(\frac{1 + (1 - 2 \times 10^{-6})\frac{e}{E_{\text{incident}}}}{1 - \frac{e}{E_{\text{incident}}}}\right) \tag{6.1}$$

where E_{incident} is the incident particle energy. This formulation ensures stability while preserving the ratio of the deposited energy to the incident energy. The advantages of this approach include:

- **Prevents numerical instability**: The small offset ensures that divisions by zero do not occur.
- **Incident Related Logit Transformation**: The transformation densifies the distribution of energy values and reduces variance between different incident energy levels.

z-Coordinate Transformation

The z-coordinate transformation applies a linear rescaling:

$$z' = \frac{z - z_{\min}}{z_{\max} - z_{\min}} \tag{6.2}$$

ensuring values remain within a fixed range while preserving spatial relationships. Benefits include:

- **Normalization improves model stability**: A fixed range enhances model generalization.
- **Outlier Cut Easier**: Knowing detector boundaries ($z_{\max}, z_{\min}$), we can discard predictions outside $(0, 1)$.

Above are the preprocessing methods for energy and z. Next, we introduce preprocessing methods for x and y.

- **RobustScaler on x and y**

  The `RobustScaler` removes the median and scales data using the interquartile range (IQR), making it **robust to outliers**. The transformation is:

  $$x' = \frac{x - \text{Median}}{\text{IQR}}$$

  where IQR $= Q3 - Q1$ represents the range between the 75th and 25th percentiles. This transformation is particularly effective in datasets with extreme values. Advantages include:

- **Resistant to Outliers**: Using the median and IQR minimizes the influence of extreme values.
- **Preservation of Relative Distances**: The transformation retains the original distribution while normalizing the scale.
- **Effective for Skewed Data**: Works well on data with heavy tails or asymmetric distributions.

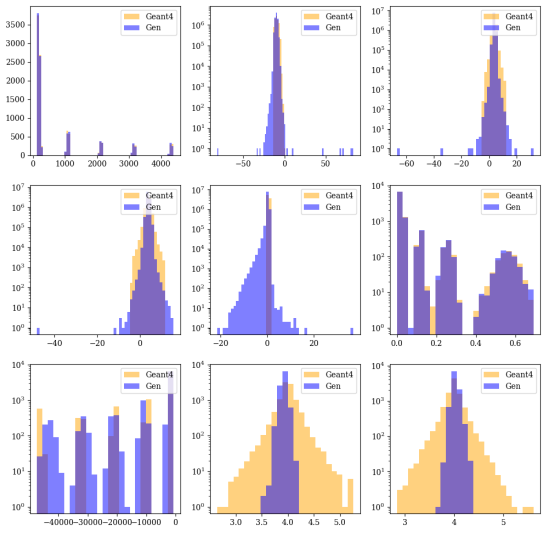


FIGURE 6.1: RobustScaler

- **QuantileTransformer on x and y**

  As for `QuantileTransformer`, it is a non-linear transformation that maps data to a uniform or normal distribution. Here I choose the normal one. It applies a non-linear transformation using the empirical cumulative distribution function (ECDF) to reshape the feature's distribution. This ensures that each feature closely resembles the desired target distribution.

  This method is particularly useful when the data distribution has heavy tails or abrupt changes, as our x and y coordinates do. By transforming the data to a normal distribution, the `QuantileTransformer` can help the model learn the underlying patterns more effectively. This is especially beneficial for our data, as it can improve the model's ability to capture the relationship between energy and radius. The advantages of the `QuantileTransformer` include:

- **Uniform-to-Normal Mapping**: Converts arbitrary distributions into a normal distribution, aiding model interpretability.
- **Outlier Robustness**: Reduces the influence of extreme values using empirical percentiles.
- **Smooth Data Representation**: Reshapes skewed or heavy-tailed distributions into a well-behaved normal form.

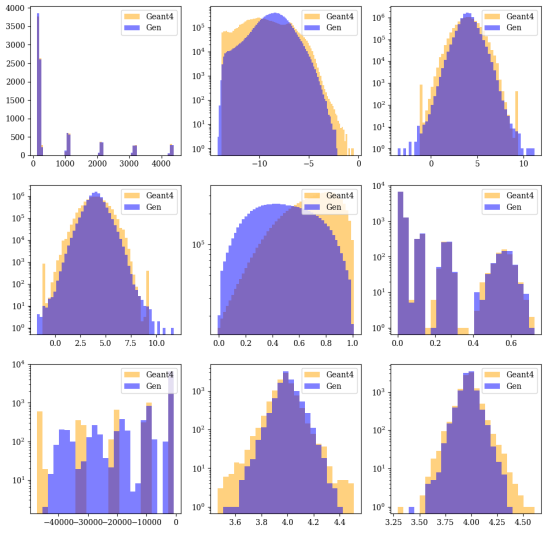


FIGURE 6.2: QuantileTransformer

- **Exponential Transformation**

  The **Exponential Transformation** follows a **sigmoid-based scaling**:

$$x', y' = \frac{1}{1 + e^{-0.07 \cdot (x,y)}} \tag{6.3}$$

  which maps original x, y coordinates into a compressed range, preventing extreme values from dominating. Advantages include:

  - **Soft bounding of values**: Ensures large deviations do not dominate the scale.
  - **Improved gradient stability**: The sigmoid function provides smooth gradients, improving model training.

– **Consistent mapping**: Unlike statistics-based transformations, this method applies a continuous function, making it robust for out-of-distribution inputs.

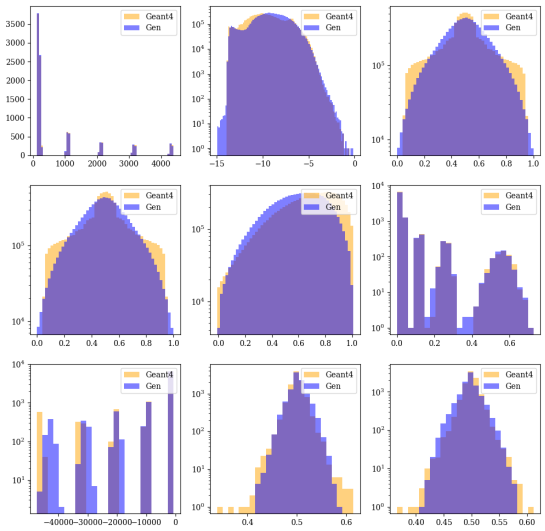


FIGURE 6.3: Exponential Transformation

The figures 6.1, 6.2, and 6.3 show the distribution of data after applying different preprocessing methods. More figures can be found in Appendix A.

From these results, we see that the `QuantileTransformer` performs best for x and y because it transforms data into a normal distribution, allowing the model to better capture spatial patterns and the relationship between energy and radius.

6.2 Metrics

6.2.1 FID Score

To evaluate the performance of our model, we employed the Fréchet Inception Distance (FID) score as a key metric. The FID score is widely used to assess the quality of generated samples by measuring the distance between the feature representations of real and generated images using the InceptionV3 model [51]. A lower FID score indicates that the generated samples are closer to the real samples in terms of their statistical distribution. We utilized the PyTorch library's implementation of the FID score [52] for our calculations. The FID score is calculated as follows:

For two multivariate Gaussian distributions with means μ_{real} and μ_{gen} and covariance matrices Σ_{real} and Σ_{gen}, the FID score is given by:

$$\text{FID} = ||\mu_{\text{real}} - \mu_{\text{gen}}||^2 + \text{Tr}(\Sigma_{\text{real}} + \Sigma_{\text{gen}} - 2(\Sigma_{\text{real}}\Sigma_{\text{gen}})^{1/2}), \tag{6.4}$$

In order to measure what's the performance on each dimension, we also calculate the FID score on each dimension. Then the FID score on each dimension is calculated as follows:

$$\text{FID}_{\text{dim}} = ||\mu_{\text{real}} - \mu_{\text{gen}}||^2 + \text{Tr}(\sigma_{\text{real}} + \sigma_{\text{gen}} - 2(\sigma_{\text{real}}\sigma_{\text{gen}})^{1/2}), \tag{6.5}$$

One important point to note is that sometimes the FID score is not enough to evaluate the performance of the model. For example, if the FID score is low, it means that the generated samples are close to the real samples in terms of their statistical distribution. However, the generated samples may not capture the underlying physics of the data, for example, the shape of the data may not be the gaussian distribution. In this case, the FID score may not be a good metric to evaluate the performance of the model. So when we evaluate the performance of the model, we still need to rely on other metrics and observation.

6.2.2 Classifier

As mentioned earlier, the FID score alone is insufficient for evaluating the performance of the model. To complement it, we employ classifiers to assess the model's ability to generate realistic samples. These classifiers are binary, designed to distinguish between real and generated samples. The structure of the classifiers is primarily based on deep neural networks (DNNs). The input features for the classifiers can range from high-level features, such as energy distributions across layers or θ bins, to low-level features like the energy values in each voxel. Regardless of the input, real samples are labeled as 1, and generated samples are labeled as 0. The loss function used is the Binary Cross-Entropy Loss (`BCEWithLogitsLoss`).

However, our classifiers consistently achieve very high performance, with an AUC of 99–100%. This indicates that it is relatively easy for the classifier to distinguish between real and generated samples. This issue is not unique to our study; many papers report similar findings, even when their models achieve low FID scores and realistic data shapes. One plausible explanation is that generated samples tend to exhibit higher continuity, while real data has inherent discreteness due to the limitations of the detector. This mismatch in continuity could make it easier for classifiers to identify generated samples.

6.3 VE and VP Studies

As mentioned before, there are two main ways to add the noise into the data, which are Variance Exploding (VE) and Variance Preserving (VP). In this section, we will discuss the performance of the model trained with these two methods. First, we can observe the standard deviation times the noise we add in, which is the change of every step. We can see that it is more steep and the value is bigger for VE method. This means that the VE method has more power to push the data to the random noise, which is the initial state of the sampling space. That is why we guess the VE method will have a better performance than the VP method.

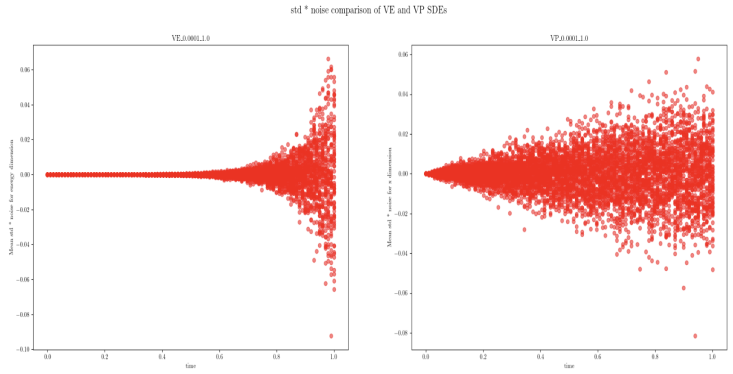


FIGURE 6.4: Comparison of VE and VP methods for both $\sigma_{max} = 1, \sigma_{min} = 0.0001$

From Figure 6.4, it may not be obvious that the value of VE is larger, but later if we see Figure 6.5 and 6.6 when $\sigma_{max} = 5, \sigma_{max} = 10$, we can see that the value of VE is larger than VP.(0.3 vs 0.075) This is consistent with our guess that the VE method will have a better performance than the VP method.

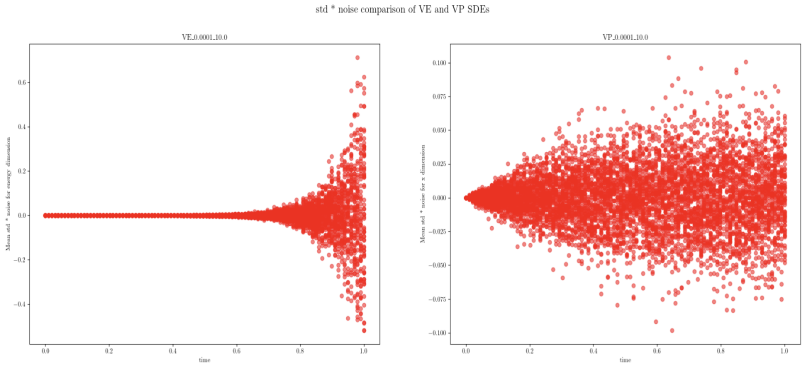


FIGURE 6.5: Comparison of VE and VP methods for both $\sigma_{max} = 5, \sigma_{min} = 0.0001$

Next, we can further compare the actual distribution change after adding the noise. We can see that the distribution of the data after adding the noise in the VE method is

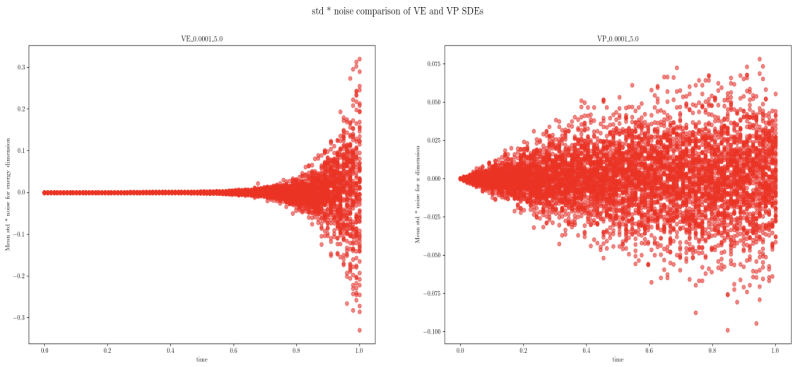


FIGURE 6.6: Comparison of VE and VP methods for both $\sigma_{max} = 10, \sigma_{min} = 0.0001$

more close to the random noise than the VP method. This is consistent with our guess that the VE method will have a better performance than the VP method.

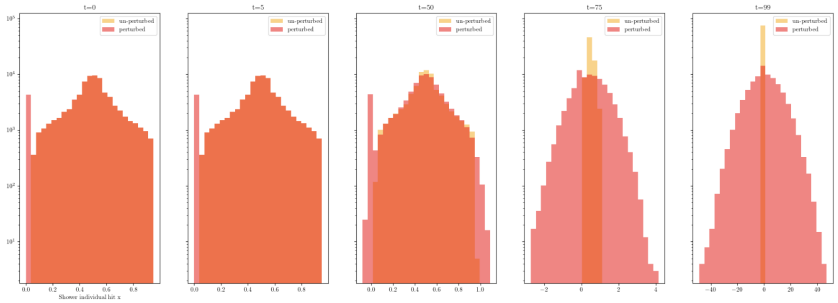


FIGURE 6.7: The distribution of the data after adding the noise using VE method.

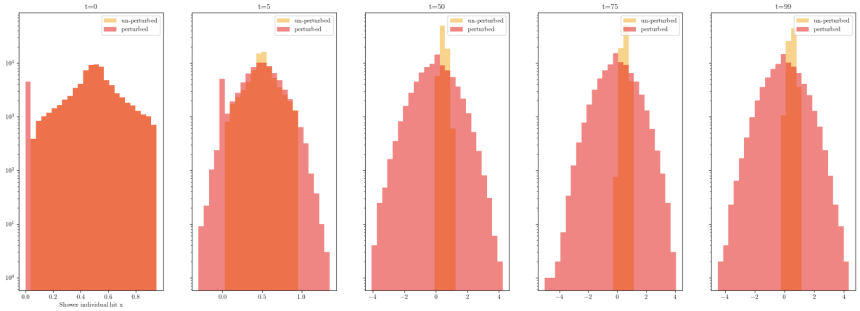


FIGURE 6.8: The distribution of the data after adding the noise using VP method.

From Figure 6.7 and 6.8, we can observe two points. Firstly, the VE method requires more steps to effectively disrupt the original data distribution, allowing the reverse process to provide the model with additional opportunities to capture the true distribution, which is advantageous. Secondly, the distribution of data subjected to noise through the VE method appears closer to random noise than that of the VP method.

This observation supports our hypothesis that the VE method is likely to outperform the VP method.

We also compared the FID scores of models trained with the VE and VP methods. The results showed that the VE method resulted in a lower FID score compared to the VP method. This suggests that the VE method is more effective at pushing the model toward generating random samples that better represent the initial sampling space.

TABLE 6.1: Comparison of FID scores for VE and VP methods.

In conclusion, the VE method outperformed the VP method in terms of FID score. We guess this is because it has more power to push our data to random noise, which is the initial state of sampling space. So our model know how to do the reverse process at the beginning in VE method. For example, if we see the standard deviation of both VE and VP methods, one can find out VE has the steeper slope than VP, which means it has the power to push the data to the random noise.

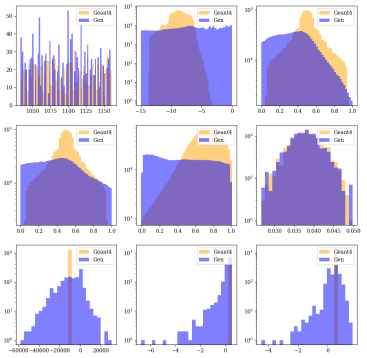
6.4 σ_{max} and σ_{min} Studies

Among all fo the parameters, the σ_{max} may be the most important one. In the context of diffusion models, the parameters σ_{max} and σ_{min} play a crucial role in determining the noise levels introduced during the forward and backward processes. These parameters define the range of noise scales, influencing both the quality of the generated samples and the training stability of the model. This section explores the impact of σ_{max} and σ_{min} on model performance and provides insights into selecting optimal values for these parameters.

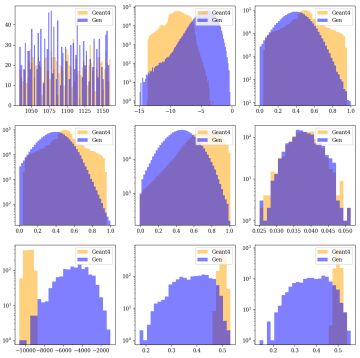
6.4.1 The Role of σ_{max} and σ_{min}

The parameter σ_{min} represents the minimum noise level in the forward process and also used as the step size of σ series. In this case as you can imagine, σ_{min} is typically set close to zero. However, based on our observation, σ_{min} won't actually affect too mcuh on the performance of our model. Conversely, σ_{max} deos. It defines the maximum noise level and is set high enough to approximate a standard normal distribution. And it also determines the power to change our data distribution during the training. These noise levels influence the progression of the diffusion process, as the model learns to reverse the added noise during training.

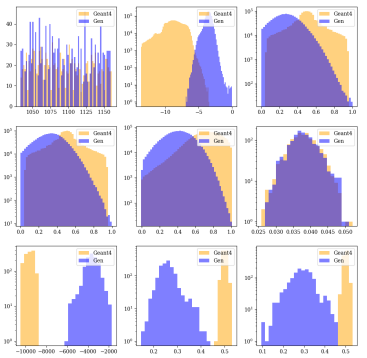
Larger σ_{max} ensures sufficient diversity in the data during the forward process, helping the model generalize better. Yet, if σ_{max} is too large, it can result in excessively noisy samples, making it challenging for the model to learn the reverse process effectively.



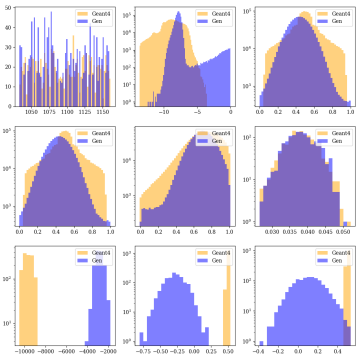
(A) VP with $\sigma_{max} = 20$



(B) VP with $\sigma_{max} = 10$



(C) VP with $\sigma_{max} = 5$



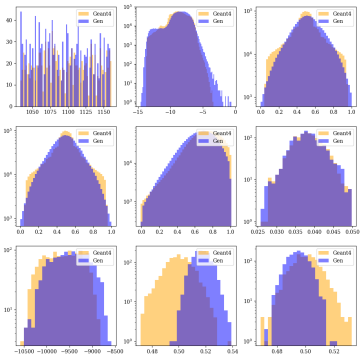
(D) VP with $\sigma_{max} = 1$

FIGURE 6.9: The result of different σ_{max} in VP.

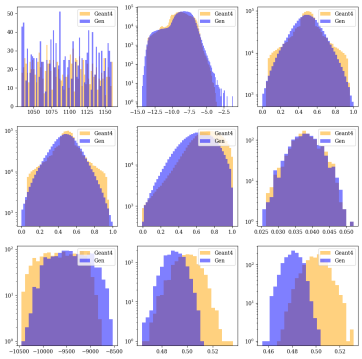
And we can further check the sweep in Figure 6.11 and 6.12. One can see that the performance of the model is better when σ_{max} is larger. This actually fit with our prediction as the reasons above.

6.4.2 Conclusions

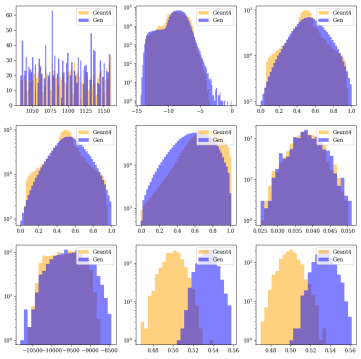
As shown in the results, the choice of σ_{max} and σ_{min} significantly impacts the performance of the model. Larger σ_{max} values can improve the diversity of the data and enhance the model's generalization ability. However, setting σ_{max} too high can lead to noisy samples and hinder the model's learning process. On the other hand, σ_{min}



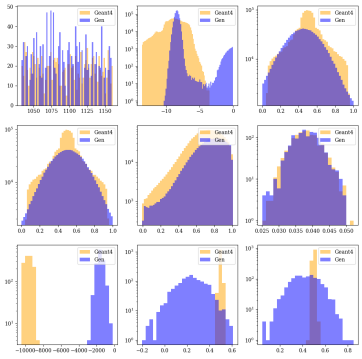
(A) VE with $\sigma_{max} = 20$



(B) VE with $\sigma_{max} = 10$



(C) VE with $\sigma_{max} = 5$



(D) VE with $\sigma_{max} = 1$

FIGURE 6.10: The result of different σ_{max} in VE.

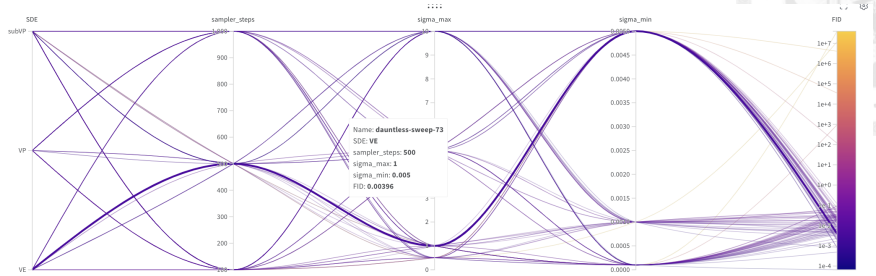


FIGURE 6.11: The result of different σ_{max} and σ_{min} in VE.

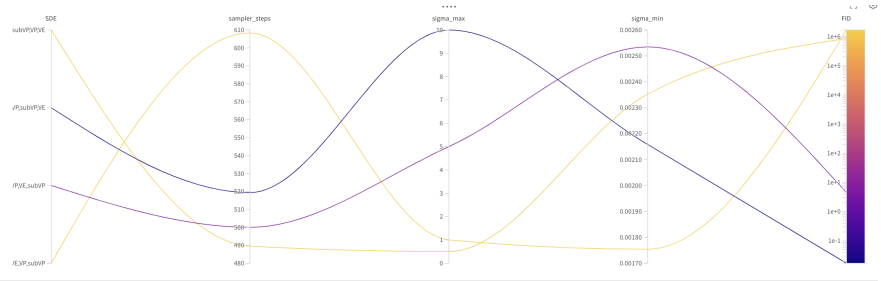


FIGURE 6.12: The result of fig 6.11, but grouped by σ_{max} in VE.

has a less pronounced effect on model performance, as it primarily serves as the step size for the noise levels. And based on our data scale, we choose σ_{min} to be 0.0003, and σ_{max} to be 5.0 in VE.

6.5 Overall Parameter Sweeping

Besides, the parameters mentioned above there are also a lot of other parameters that can affect the performance of the model or the memory allocation. Thus, we conducted a parameter sweeping study using `wandb`. We experimented with various learning rates, batch sizes, and hidden dimensions. Our findings indicated that the best-performing parameter configuration was:

- Learning rate: 0.0003
- Batch size: 128
- Embedding dimension: 96
- Hidden dimension: 96
- Number of Attention Heads: 8
- Number of Encoder Blocks : 16
- Dropout rate: 0.2

- Sampler Step: 100
- Correction Step: 25
- SDE : VE
- Sigma Max: 5.0
- Sigma Min: 0.0003

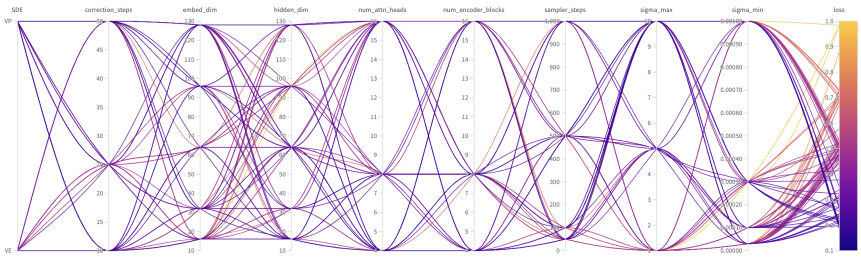


FIGURE 6.13: Visualization of parameter sweeping results.

6.6 Centralization

After visualizing 2D or 3D plots revealed that the model failed to capture the relationship between energy and radius. A key observation was that the model could not learn that higher energy values should be concentrated near the center (smaller radii). Consequently, while the 1D plots were satisfactory, the generated samples lacked proper centralization.

To address this, we first tried to transform our data into spherical coordinate and introduce a correlation term between energy and theta in the loss function to try to suppress relation between energy and theta, hoping our model can thus learn more about the relation between energy and radius.

The new loss function is defined as:

$$L = L_{\text{MSE}} + \lambda L_{\text{cor}}, \tag{6.6}$$

where L_{MSE} is the mean squared error loss, L_{cor} is the correlation loss, and λ is a weighting factor for the correlation loss. The correlation loss is defined as:

$$L_{\text{cor}} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(y_i - \bar{y}), \tag{6.7}$$

where x and y are the variables of interest, and $\bar{x}$ and $\bar{y}$ are their respective means.

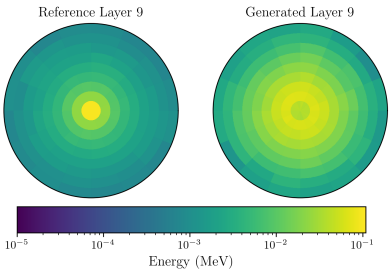
The reason why we don't apply the correlation term between energy and radius is that the relation between them is by experienced, although everyone would expect the result, it's not solid, we don't want to bias our model, or you can say we don't want to tell the answer of the relation to our model.

However, although the correlation term was added to the loss function and it indeed suppressed the relation between energy and theta, the centralization of the generated samples did not improve significantly. This suggests that the correlation term alone is not sufficient to address the centralization issue.

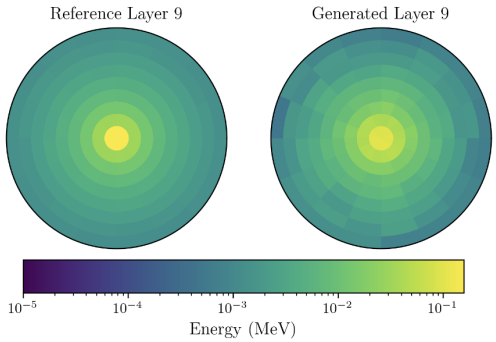
FIGURE 6.14: The Picture after adding the correlation term.

After that, one time when we tried to use QuantileTransformer to preprocess the data, we found that the centralization of the data is improved. This is because the QuantileTransformer can transform the data to follow a uniform or a normal distribution. This can help the model to learn the data better, especailly the x,y distribution. This also makes it is able to learn the relation between energy and radius better.

The result compared to the original one is shown in Figure 6.15.



(A) The energy in each voxel with original transformation.



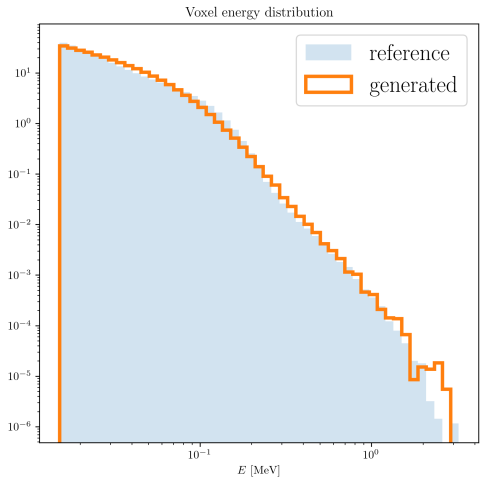
(B) The energy in each voxel with QuantileTransformer transformation.

FIGURE 6.15: The Comparison Picture after using QuantileTransformer.

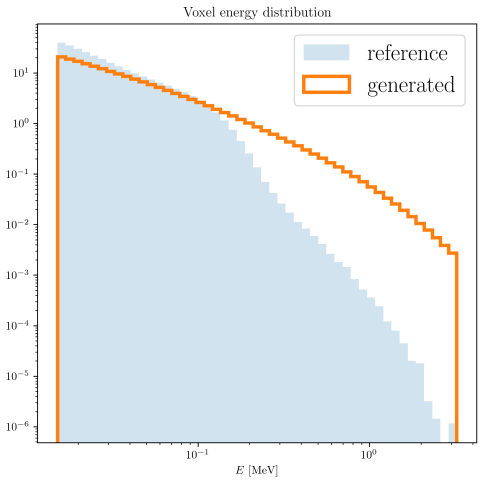
6.7 Conditioning Issue

6.7.1 Incident energy

With the optimal settings, our model was able to generate the basic shapes of both the energy and spatial distributions. However, the model often produced an excessive number of hits (`nhits`) at higher energy levels, leading to overestimation. This issue was not observed when training on single-bucket data, indicating that the model struggles to differentiate between data from different buckets. This suggests that our conditional variables are not functioning effectively. As you can see the result of energy deposit of single bucket data and all bucket data, the model can generate the data well in single bucket data, but it failed to generate the data well in all bucket data. This is because the model can't learn the condition well.



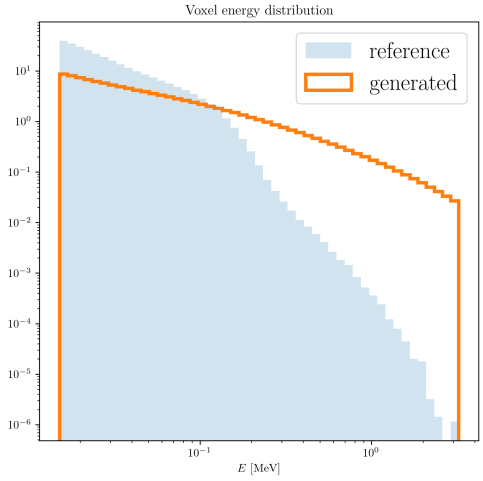
(A) The energy deposit of single bucket data.



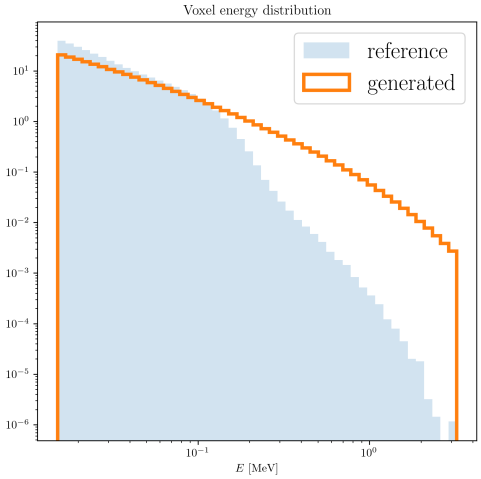
(B) The energy deposit of five bucket data.

FIGURE 6.16: he result of energy deposit of single bucket data and all bucket data.

To address this issue, we first need to make sure if our conditional variables aren't really working. So we tried to add the incident energy as the conditional variables and not. The result is shown below:



(A) The energy deposit without incident energy.



(B) The energy deposit with incident energy.

FIGURE 6.17: Main caption for both figures

They are basically the same, indicating that the conditional variables are not working. Next, we also tried to concatenate the incident energy with the input data instead of adding them and the result is shown Figure 6.18.

As you can see, the result is totally a disaster. To be honest, we still don't know why this happened. And that is also one of the things we need to figure out in the future.

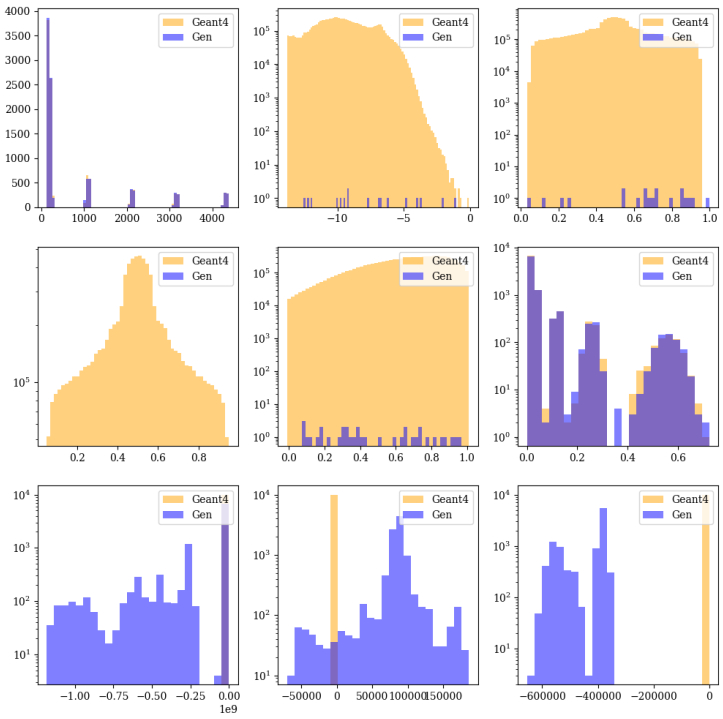


FIGURE 6.18: The result of energy deposit with incident energy concatenated with the input data.

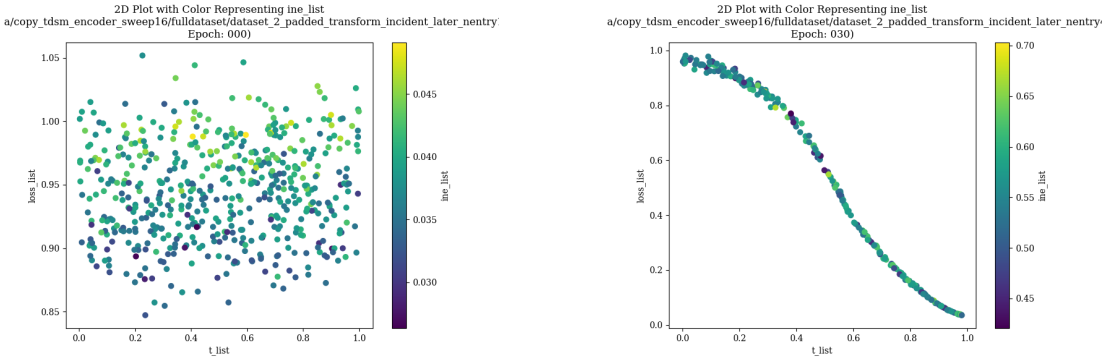
We will probably try to use fewer hits and focus on one or two dimensions to find out the reason.

6.7.2 Time

We also attempted to incorporate time as a conditional variable in our model. However, the results differ from those observed with incident energy. Even without explicitly using time as a condition, the output layer implicitly incorporates its effect by dividing by the standard deviation of the stochastic differential equation (SDE), which is time-dependent. Despite this, the inclusion of time as a conditional variable does not appear to enhance the model's performance.

The plot of loss versus time reveals consistent behavior across epochs, showing that the shape of this plot remains virtually unchanged. Notably, the loss value at $t = 0$ is almost identical to the initial loss, indicating that the time condition fails to improve the model's capacity to learn the data effectively.

Time close to $t = 0$ represents the critical phase where the model transitions toward generating real data, whereas near $t = 1$, the model predominantly learns the structure of Gaussian noise. This suggests that the model's learning mechanism may inherently prioritize earlier time steps, making additional time conditioning redundant or ineffective.



(A) (B)

FIGURE 6.19: The left figure shows the loss at epoch 0, which is quite normal it's still caotic. The right figure actually represent the loss after 10 epochs.

From the Figure 6.19, We can see that the loss value near t equals to 0 is always aorund 1, which is as same as it is at the initial state. This means that our model learns nothing in that region. And the region stands for our model to predict the real data. That's may be the reason why our model can't learn the real shape of the data well. It only learns the approximate shape of the data from time is above 0.4.

6.8 Conclusion

In this work, we explored various data preprocessing techniques and model configurations to improve the performance of our transformer-based generative model for detector hit data. We implemented multiple strategies to optimize the data representation and ensure efficient learning.

First, we introduced a **bucketing strategy** to handle the variability in sequence lengths, significantly reducing memory overhead while improving computational efficiency. This allowed us to maintain a structured approach to feeding data into the transformer, ensuring stable training dynamics.

For **preprocessing**, we applied three distinct transformations—**RobustScaler, QuantileTransformer, and Exponential Transformation**—to normalize the hit coordinate data while ensuring robustness to outliers. We observed that **QuantileTransformer

provided the best performance**, as it effectively reshaped the data into a normal distribution, improving the model's ability to capture spatial relationships and energy dependencies.

Through extensive experimentation with **variance exploding (VE) and variance preserving (VP) stochastic differential equations**, we found that **VE outperforms VP** in terms of pushing the data distribution towards an effective generative space, resulting in lower FID scores and better model convergence.

We also examined the effect of **key hyperparameters, particularly σ_{max} and σ_{min}**, in controlling the diffusion process. Our results indicate that **a larger σ_{max} improves the diversity and realism of generated samples** by facilitating a more expressive transformation of the data, while σ_{min} had a minor impact on overall performance.

To assess model quality, we used the **Fréchet Inception Distance (FID) score**, supplemented with a **classifier-based evaluation**. While the classifier achieved near-perfect performance in distinguishing real and generated samples, we observed that **real-world constraints and detector properties introduce inherent discreteness** that can be challenging for the generative model to replicate.

One significant challenge we encountered was the **conditioning issue**, particularly with incident energy and time as conditional variables. Despite various conditioning strategies, including direct concatenation and implicit conditioning through normalization, the model struggled to fully leverage these inputs. This suggests that additional work is required to refine the conditional mechanisms to improve control over generated samples.

Additionally, we observed a **centralization issue** in the generated data, where the model failed to accurately capture the expected energy-radius relationship. Our attempts to enforce correlation constraints showed limited improvement, but the **QuantileTransformer preprocessing unexpectedly enhanced centralization**, highlighting its potential importance in data representation.

Moving forward, future work will focus on:

- Improving the **conditioning mechanism** to ensure that incident energy and other physical parameters effectively guide the generation process.
- Investigating **alternative loss functions** and **regularization techniques** to better capture the physical constraints of the detector.
- Exploring **architectural modifications**, such as hybrid transformer-CNN approaches, to better leverage spatial dependencies in hit distributions.

- Refining the **preprocessing pipeline** by testing other transformations that could further enhance the model's ability to generalize across different hit distributions.

In conclusion, while our model demonstrates strong generative capabilities and promising results, further refinement is needed to fully capture the underlying physics of detector hit data. The findings in this work provide a solid foundation for future improvements in data-driven generative modeling in high-energy physics applications.

Chapter 7

Future Goals

Looking ahead, there are two primary objectives for future work:

7.1 Further Acceleration of the Model

The first goal is to further improve the speed of the model. Currently, our model achieves a 500x speedup compared to Geant4 simulations. However, there is potential for even greater acceleration by exploring alternative methods. For instance, replacing the Stochastic Differential Equation (SDE) framework with an Ordinary Differential Equation (ODE) approach, or implementing a restart method as suggested in [**restart**], could lead to significant improvements in computational efficiency.

7.2 Layer Relationship Learning and Tracking

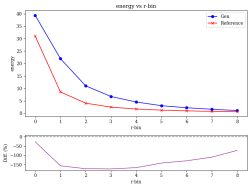
The second goal is to enhance the model's ability to learn the relationships between layers. Specifically, we aim to train the model to identify which hits in one layer correspond to hits in the previous layer. This capability would enable the development of a model for particle tracking, providing a more comprehensive and detailed understanding of the underlying physical processes.

Achieving these goals would not only improve the current model but also open new possibilities for its application in simulation and analysis.

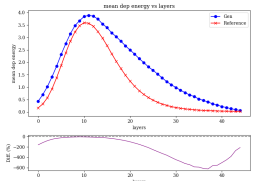
Appendix A

Figures

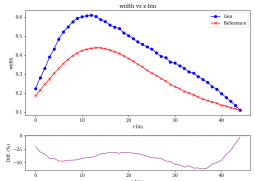
A.1 Best Result for Full Dataset



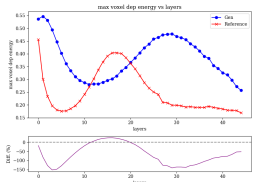
(A) Energy vs Radius



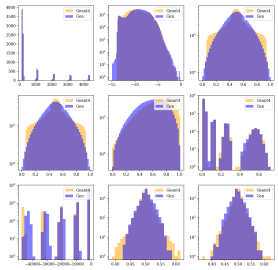
(B) Energy vs Z



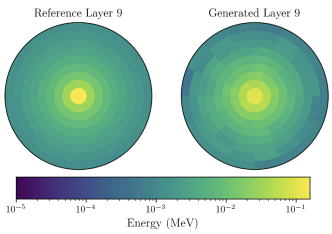
(C) R-width vs Layers



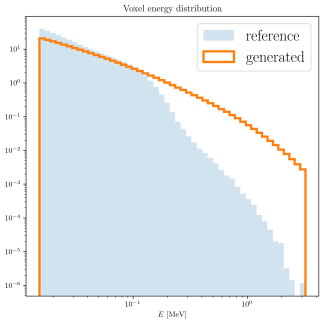
(D) Max Voxel Deposit vs Layers



(E) 1D Histogram

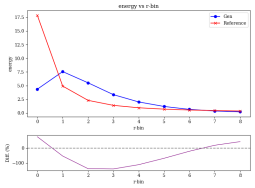


(F) Energy voxel comparison

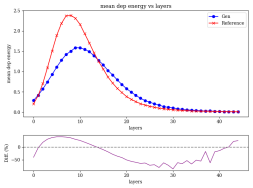


(G) Energy Deposit

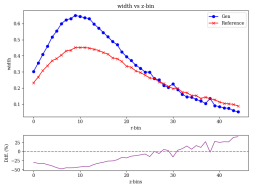
A.2 Best Result for Single Bucket Data



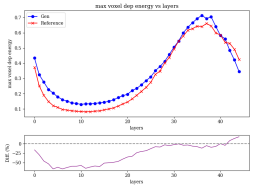
(A) Energy vs Radius



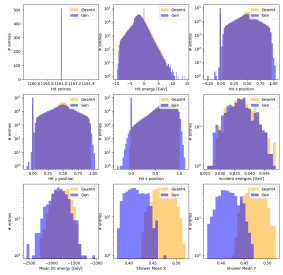
(B) Energy vs Z



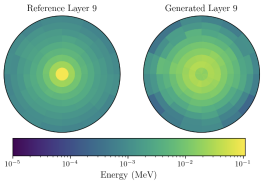
(C) R-width vs layers



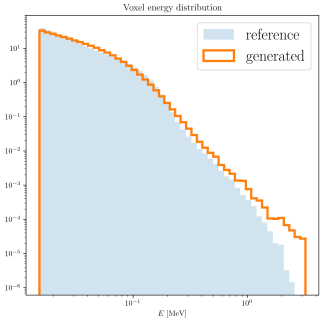
(D) Max Voxel Deposit vs Layers



(E) 1D Histogram

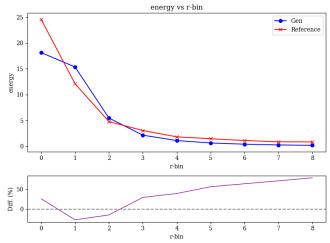


(F) Energy voxel comparison

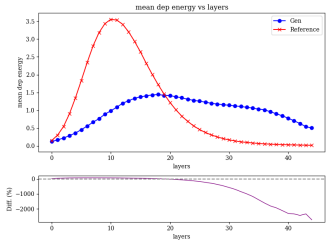


(G) Energy Deposit

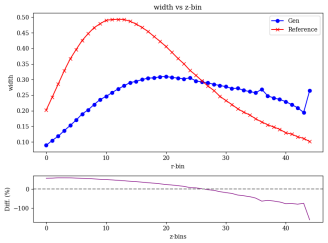
A.3 Result for using different Preprocessor



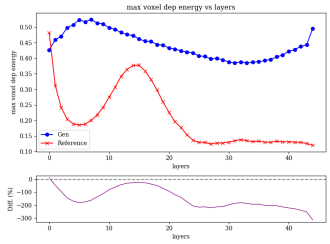
(A) Energy vs Radius



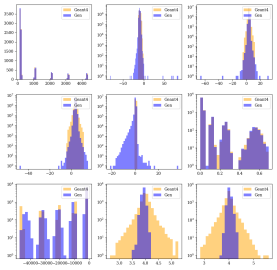
(B) Energy vs Z



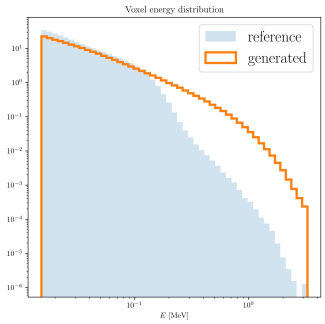
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers

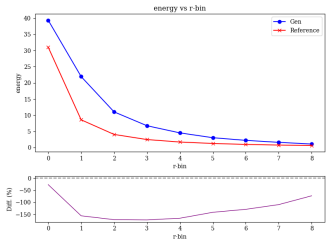


(E) 1D Histogram

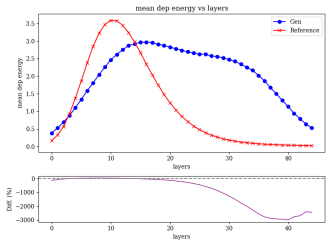


(F) Energy Deposit

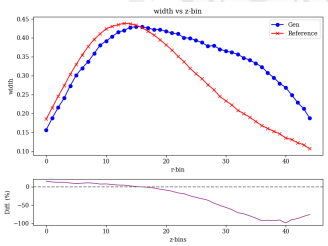
FIGURE A.3: Result for using robust preprocessor



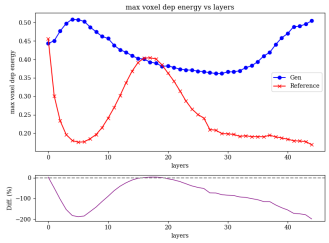
(A) Energy vs Radius



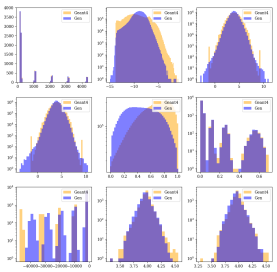
(B) Energy vs Z



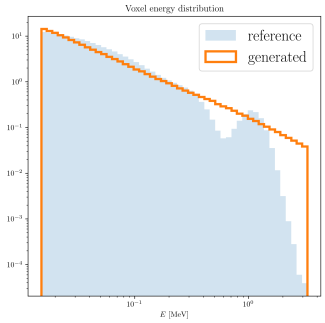
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers

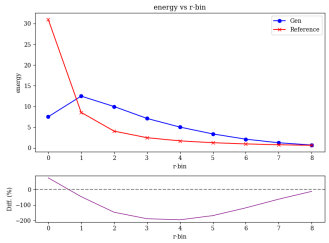


(E) 1D Histogram

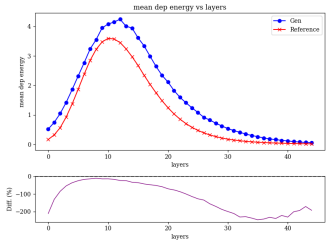


(F) Energy Deposit

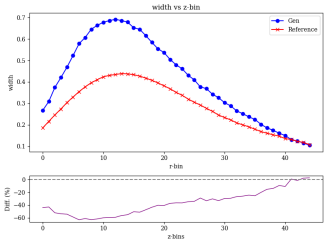
FIGURE A.4: Result for using quantile preprocessor



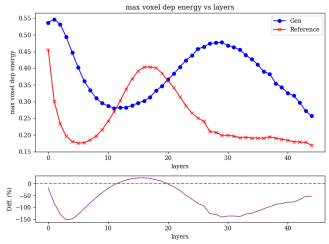
(A) Energy vs Radius



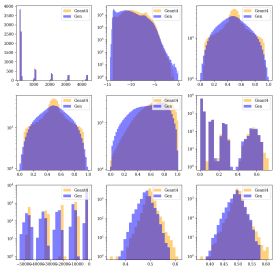
(B) Energy vs Z



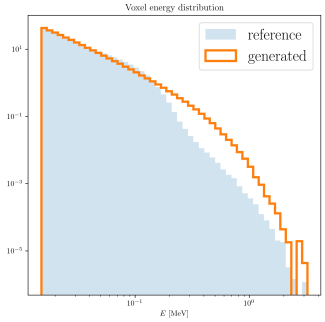
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers

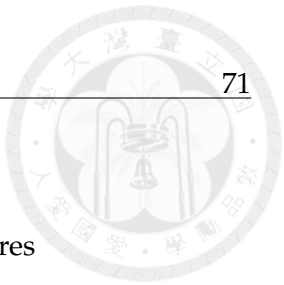


(E) 1D Histogram



(F) Energy Deposit

FIGURE A.5: Result for using exponential preprocessor



A.4 Result for using different SDE settings

every 7 pictures for VE,VP for sde =1,5,10,20. Total 7*2*4=56 pictures

Appendix B

TopFCNC

B.1 Introduction

B.2 Background

B.3 Analysis Workflow

B.4 Gridpack Generation

B.5 Current Status

Bibliography

[1] Lyndon Evans and Philip Bryant. "LHC Machine". In: *Journal of Instrumentation* 3.08 (2008), S08001. DOI: 10.1088/1748-0221/3/08/S08001.

[2] The Atlas Collaboration et al. "The ATLAS Experiment at the CERN Large Hadron Collider". en. In: *Journal of Instrumentation* 3.08 (Aug. 2008), S08003–S08003. ISSN: 1748-0221. DOI: 10.1088/1748-0221/3/08/S08003.

[3] The CMS Collaboration et al. "The CMS experiment at the CERN LHC". In: *Journal of Instrumentation* 3.08 (2008), S08004. DOI: 10.1088/1748-0221/3/08/S08004.

[4] G Apollinari et al. *High Luminosity Large Hadron Collider HL-LHC*. en. 2015. DOI: 10.5170/CERN-2015-005.1.

[5] S. Agostinelli et al. "Geant4—a simulation toolkit". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 506.3 (2003), pp. 250–303. ISSN: 0168-9002. DOI: https://doi.org/10.1016/S0168-9002(03)01368-8.

[6] Y. Song and S. Ermon. "Score-Based Generative Modeling through Stochastic Differential Equations". In: *arXiv preprint arXiv:2011.13456* (2020).

[7] V. Mikuni and B. Nachman. "CaloScore: A Conditional Generative Model for Calorimeter Shower Simulation". In: *arXiv preprint arXiv:2106.00792* (2021).

[8] S. Agostinelli et al. "Geant4: A Simulation Toolkit". In: *Nuclear Instruments and Methods in Physics Research Section A* 506.3 (2003), pp. 250–303.

[9] J. Allison et al. "Geant4 Developments and Applications". In: *IEEE Transactions on Nuclear Science* 53.1 (2006), pp. 270–278.

[10] J. Allison et al. "Recent Developments in Geant4". In: *Nuclear Instruments and Methods in Physics Research Section A* 835 (2016), pp. 186–225.

[11] I. Goodfellow et al. "Generative Adversarial Networks". In: *arXiv preprint arXiv:1406.2661* (2014).

[12] D. P. Kingma and M. Welling. "Auto-Encoding Variational Bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[13] L. Dinh, J. Sohl-Dickstein, and S. Bengio. "Density Estimation Using Real NVP". In: *arXiv preprint arXiv:1605.08803* (2016).

[14] M. Paganini, L. de Oliveira, and B. Nachman. "CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks". In: *Physical Review D* 97.1 (2018), p. 014021. DOI: 10.1103/PhysRevD.97.014021.

[15] ATLAS Collaboration. *Fast Calorimeter Simulation with Generative Adversarial Networks*. Tech. rep. ATL-SOFT-PUB-2018-001, 2018.

[16] S. Verheyen and B. Krislock. "CaloFlow: Fast and Accurate Generation of Calorimeter Showers with Normalizing Flows". In: *arXiv preprint arXiv:2106.05285* (2021).

[17] S. Verheyen and B. Krislock. "CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows". In: *arXiv preprint arXiv:2107.13684* (2021).

[18] CMS Collaboration. *The CMS High Granularity Calorimeter for HL-LHC Upgrade*. Tech. rep. CERN-LHCC-2017-023, 2017.

[19] CMS Collaboration. "Design and Performance of the CMS Beam Radiation, Instrumentation, and Luminosity Detectors". In: *Journal of Instrumentation* 13.10 (2018), P10034.

[20] Forthommel. *English: Map of the CERN accelerator complex*. May 2011.

[21] *Linear accelerator 4*. en. Dec. 2024.

[22] *The Proton Synchrotron Booster*. en. Dec. 2024.

[23] *The Proton Synchrotron*. en. Dec. 2024.

[24] *The Super Proton Synchrotron*. en. Dec. 2024.

[25] *Pulling together: Superconducting electromagnets*. en. Dec. 2024.

[26] *The Large Hadron Collider*. en. Dec. 2024.

[27] A Hervé. "The CMS detector magnet". In: *IEEE Trans. Appl. Supercond.* 10.1 (2000), pp. 389–94. DOI: 10.1109/77.828255.

[28] M.C Fouz. "The CMS Muon detectors". In: *2007 IEEE Nuclear Science Symposium Conference Record*. Vol. 3. 2007, pp. 1885–1890. DOI: 10.1109/NSSMIC.2007.4436524.

[29] CMS Collaboration. *The Tracker Technical Design Report*. Tech. rep. CERN/LHCC 98-006. CERN, 1998.

[30] *Silicon Pixels | CMS Experiment*.

[31] *Silicon Strips | CMS Experiment*.

[32] CMS Collaboration. *The Electromagnetic Calorimeter Technical Design Report*. Tech. rep. CERN/LHCC 97-033. CERN, 1997.

[33] CMS Collaboration. *The Preshower Detector Technical Design Report*. Tech. rep. CERN/LHCC 99-033. CERN, 1999.

[34] Rosalinde Pots. "Investigation of new technologies to improve light collection from scintillating crystals for fast timing". PhD thesis. May 2022. DOI: 10.18154/RWTH-2022-04865.

[35] CMS Collaboration. *The Hadronic Calorimeter Technical Design Report*. Tech. rep. CERN/LHCC 96-041. CERN, 1997.

[36] CMS Collaboration. "Forward Hadron Calorimeter Design and Performance". In: *CERN-PH-EP/2006-002* (2006).

[37] CMS Collaboration. *Outer Hadron Calorimeter Technical Performance Report*. Tech. rep. CERN/LHCC 98-030. CERN, 1998.

[38] CMS Collaboration. "The CMS Detector at the LHC". In: *Journal of Instrumentation* 3.S08004 (2008).

[39] CMS Collaboration. *The Hadronic Calorimeter Technical Design Report*. CERN/LHCC 96-041. 1996.

[40] CMS Collaboration. *The Muon Technical Design Report*. Tech. rep. CERN/LHCC 97-032. CERN, 1997.

[41] ChristopherStephen. *The CMS Muon Detector*. 2025.

[42] CMS Collaboration. *The Muon Technical Design Report*. CERN/LHCC 97-032. 1997.

[43] CMS Collaboration. *The Trigger and Data Acquisition Technical Design Report*. Tech. rep. CERN/LHCC 2000-038. CERN, 2000.

[44] CMS Collaboration. *The Technical Design Report for the High-Granularity Calorimeter for the Phase-2 Upgrade of the CMS Experiment*. Tech. rep. CERN-LHCC-2017-023. CERN, 2017.

[45] Nural Akchurin et al. "First beam tests of prototype silicon modules for the CMS High Granularity Endcap Calorimeter". In: *Journal of Instrumentation* 13 (Oct. 2018), P10023–P10023. DOI: 10.1088/1748-0221/13/10/P10023.

[46] S. Agostinelli et al. "Geant4: A Simulation Toolkit". In: *Nuclear Instruments and Methods in Physics Research A* 506.3 (2003), pp. 250–303.

[47] Yang Song and Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. arXiv:1907.05600. Oct. 2020.

[48] Jonathan Ho, Ajay Jain, and Pieter Abbeel. "Denoising Diffusion Probabilistic Models". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle et al. Vol. 33. Curran Associates, Inc., 2020, pp. 6840–6851.

[49] Matthew Tancik et al. *Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains*. en. arXiv:2006.10739 [cs]. June 2020.

[50] Benno Käch, Isabell Melzer-Pellmann, and Dirk Krücker. *Pay Attention To Mean Fields For Point Cloud Generation*. en. arXiv:2408.04997 [hep-ex]. Aug. 2024.

[51] PyTorch-Ignite Contributors. *FID — PyTorch-Ignite v0.5.1 Documentation*. en.

[52] *PyTorch*. en.