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
在雙光子衰變通道中搜尋頂夸克味變中性希格斯耦合

Fast HGCal Detector Simulation via Score-Based Diffusion Models
Searching for Top Quark Flavor Changing Neutral Higgs Couplings in $H \to \gamma\gamma$ Decay Channel at $\sqrt{s} =$ 13.6 TeV within CMS Experiment

徐振華

Chen-Hua Hsu

指導教授：陳凱風 教授

Advisor: Prof. Kai-Feng Chen

中華民國114年3月

March 2025

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

NATIONAL TAIWAN UNIVERSITY

MASTER'S THESIS

Fast HGCal Detector Simulation via Score-Based Diffusion Models
Searching for Top Quark Flavor Changing Neutral Higgs Couplings in $H \to \gamma\gamma$ Decay Channel at $\sqrt{s} =$ 13.6 TeV within CMS Experiment

---

Author:
Chen-Hua Hsu

Supervisor:
Prof. Kai-Feng Chen

March 25, 2025

doi:10.6342/NTU202500790

ken91021615@hep1.phys.ntu.edu.tw

doi:10.6342/NTU202500790

國立臺灣大學碩士學位論文

口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

通過基於評分的擴散模型實現快速 HGCal 探測器模擬
在雙光子衰變通道中搜尋頂夸克味變中性希格斯耦合

Fast HGCal Detector Simulation via Score-Based Diffusion Models
Searching for Top Quark Flavor Changing Neutral Higgs Couplings in H → γγ Decay Channel at √s = 13.6 TeV within CMS Experiment

本論文係____徐振華____(姓名)_R12222013_（學號）在國立臺灣大學__物理所__(系/所/學位學程)完成之碩士學位論文，於民國 114 年 3 月 10 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Department / Institute of _____Physics_____
on __10__(date)__3__ (month)__2025___ (year) have examined a Master's thesis entitled above presented by____Chen-Hua Hsu____ (name)__R12222013___ (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

陳劭囿　　裴思達　　高英哲

（指導教授 Advisor）

王蝶生　　王子敬

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

感謝的話

轉眼就遇到碩士的終止線，像一個初學樂手一樣，到了出現的幾個音符前才發現。整首樂曲都是如此，突發奇想的開始，然後戛然中止。大二暑假的一次偶然決定提早開啓碩士人生，又是碩二的寒假，決定爲這首樂曲提早畫上終止線。其中最感謝的，是一直支持我各種創意想法的凱風老師，感謝他總是願意花時間與我聊聊，願意答應我突然決定在兩個月內畢業，願意分享自己對研究生涯的看法，更不用提每一次學術上的提點。也想感謝榮祥老師，感謝他與我們如此親近，感謝他每一次鉅細彌遺的經驗傳承，感謝他願意讓我加入硬體組有所貢獻，也感謝他分享在學術生涯中的每一次抉擇，爲我提供了許多新的人生想法與信心。當然也要感謝Prof. Stathes paganis， 感謝他願意爲我在最後關頭寫推薦信，願意給我最大的肯定以及分享他在美國與歐洲的研究生活。

當然，還要感謝在922的大家，很開心在碩一下學期有發現這片世外桃源，完全豐富了我的碩士生活，感謝大家願意接納我，接納一個半途加入的怪人，這裡的每個人都超好，是加入前無法想像的。感謝不管問什麼問題都會回答還會一起吃午餐的大師、願意從頭到尾講解分析並且不厭其煩回答問題的裕維、最早帶我加入922還教會我許多機器學習知識的奕安、雖然一來就失戀但還是完成口試並分享經驗的綱哥，在研究上帶領我的鼎翔、借我論文模板還總是第一時間回答問題的桐哥，即使看賽車和足球還是會抽空回答問題和我們一起吃飯的星輔，做什麼事都很認眞還一起談心的碩甫，一起去系籃練球一起討論HiggsDNA的泓漪，現在還是沒有搞懂但是特別愛我的秉霖，還有一起出去玩的Hedy、最近才加入就很搞笑還一起修課的翔宇、昱傑。感謝你們讓我的碩士生活不無聊，並且獲得好多知識與能力。

還有從大學就認識的一群好友，一起去峇里島畢業旅行的大家，一起吃午餐、晚餐的大家，一起去環島的大家，幫助我申請的大家，來聽我口試的大家，一起吃火鍋交換禮物的大家，一起談天說地的大家，一起去KTV唱歌的大家，陪我度過大學和碩士許多時刻的大家。

也感謝我的女友一路以來的陪伴，不管是下班後來實驗室陪我一起奮鬥，陪我去實驗室的每個假日，協助我製作超好看的海報，以及每一次的加油，陪我度過每一個熬夜的夜晚。

也感謝家人的一路扶持，支持我做的每一個決定，定期關心我不管是學術上還是生活上遇到的挑戰，有感受到來自家人的溫暖！

最後，很幸運能順利通過口試完成碩士學業，感謝一路上遇到的所有人事物，所有的一切都讓我擁有很棒的碩士生活，希望未來能繼續保持這份熱忱，迎接接下來的每一項挑戰！

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

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。

除此之外，我也探討了頂夸克味變中性希格斯耦合（TopFCNH）的搜尋作爲一個副專案。這種耦合在標準模型中被高度抑制，但在各種新物理理論中可以被顯著增強。通過分析CMS實驗中的雙光子衰變通道，本研究爲探測罕見的頂夸克過程做出了貢獻。

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

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

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.

Besides this machine learning-based simulation project, I also explore the search for top quark flavor-changing neutral Higgs (TopFCNH) interactions as a side project. These interactions are highly suppressed in the Standard Model but can be significantly enhanced in various new physics scenarios. By analyzing the $H \to \gamma\gamma$ decay channel at $\sqrt{s} = 13.6$ TeV within the CMS experiment, this study contributes to the ongoing effort to probe rare top quark processes.

Keywords: Fast Simulation, Diffusion Model, Transformer, CaloChallenge, HGCal, Top Quark, Flavor-Changing Neutral Higgs, BSM Physics.

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

Contents

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

List of Figures

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

List of Tables

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

Chapter 1

Introduction

1.1 Motivation

The upcoming High Luminosity phase of the Large Hadron Collider (LHC) [1] presents unprecedented opportunities to explore new physics in ATLAS [2] and CMS [3]. The increased luminosity enables the collection of vast experimental data, with Run 3 nearly doubling the luminosity of Run 2 [4].

At higher collision rates, the LHC will generate approximately 1 billion proton-proton (p-p) collisions per second, captured by detectors with nearly 100 million read-out channels. With just 25 nanoseconds between successive proton bunches, new collisions occur before previous interactions fully exit the detector. This immense data volume provides rich opportunities for discovery but also introduces significant challenges in data processing, storage, and simulation.

1.1.1 The Role of Simulation in High-Energy Physics

Simulation plays a critical role in high-energy physics, allowing researchers to compare experimental data with theoretical predictions. Every study must first validate that observed data aligns with background expectations and signals, ensuring a clear understanding of each channel's contributions. However, traditional simulation methods face computational bottlenecks, particularly as data rates increase. Accelerating simulation without sacrificing accuracy is essential for timely and reliable analysis.

Monte Carlo-based methods, such as those implemented in Geant4 [5], have long been the standard for simulating particle interactions and detector responses. These simulations provide high precision but are computationally expensive and struggle to keep pace with increasing data rates. As detector complexity grows, the time required for full simulations rises, making it increasingly difficult to scale traditional techniques to modern experimental demands.

A significant portion of high-energy physics computing resources is devoted to simulating particle propagation in dense materials, particularly within calorimeters,

doi:10.6342/NTU202500790

which measure deposited energy. Simulating electromagnetic and nuclear interactions in these dense environments is particularly challenging, requiring extensive computational power. Given the constraints on computing budgets, full Geant4 simulations are impractical for all events, leading to the development of fast simulation techniques. These methods replace detailed physics-based models with simplified parameterized approaches, which, while efficient, often fail to capture high-dimensional correlations and complex particle interactions.

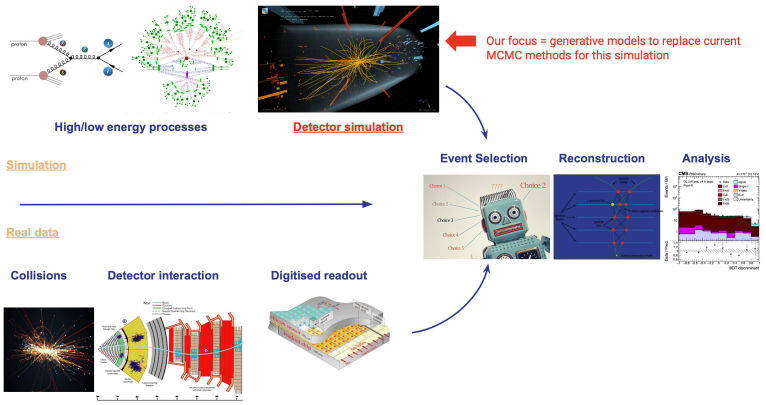


FIGURE 1.1: The importance of simulation. Credit: Joshuha Thomas-Wilsker

1.1.2 Generative Models for Fast Simulation

To address these challenges, generative models—particularly diffusion models—have emerged as promising alternatives for accelerating simulation while maintaining accuracy. Instead of replacing Geant4 entirely, the goal is to find an optimal balance between speed and precision, as illustrated in Figure 1.2. Recent works, such as Yang et al.'s score-based models [6] and diffusion-based calorimeter simulations [7], have significantly reduced computation time while preserving fidelity. Building on these advances, our project introduces a novel model that generates 3D point clouds representing energy distributions across spatial coordinates in a single step. Unlike previous models, which often focus on one-dimensional projections (e.g., energy vs. z-coordinate), our approach captures full 3D distributions in a single forward pass, enabling rapid and comprehensive simulations suited to high-luminosity experiments.

Deep learning offers a compelling alternative to traditional parametric models, with generative techniques such as Generative Adversarial Networks (GANs) [8], Variational Autoencoders (VAEs) [9], and Normalizing Flows (NFs) [10] increasingly adopted for fast detector simulations. GANs, for example, have demonstrated considerable success in generating calorimeter showers [11] and are now integrated into

doi:10.6342/NTU202500790

the ATLAS fast simulation framework [12]. However, they present optimization challenges and can suffer from mode collapse, where the generator fails to fully capture data diversity. NFs, while offering stable training and accurate density estimation, remain computationally expensive for high-dimensional data, limiting their feasibility for complex detector simulations. Additionally, their rigid model structure further constrains adaptability [13, 14].

1.1.3 Score-Based Generative Models for Simulation

This work explores score-based generative models [6], which learn the gradient of the data density rather than the density itself. This approach allows for more flexible network architectures without requiring Jacobian computation during training, enabling the use of bottleneck layers to reduce trainable parameters and improve scalability. Recent advancements in score-based models have demonstrated their potential in calorimeter simulation, achieving a balance between high-dimensional fidelity and computational efficiency—making them suitable for ultra-fine calorimeters and complex datasets [15, 16].

By leveraging score-based models, our project aims to address both the computational challenges of high-luminosity LHC experiments and the limitations of traditional fast simulation methods. Our approach enhances accuracy by capturing full 3D spatial distributions while significantly reducing simulation time, offering a scalable and reliable solution for next-generation collider experiments.

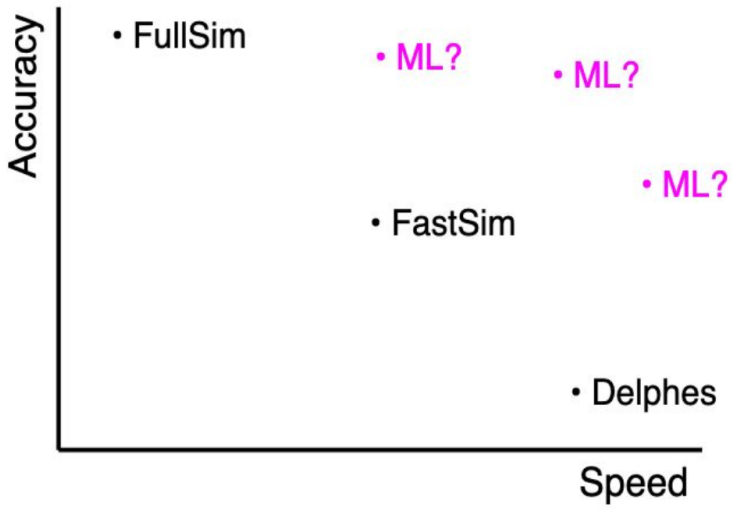


FIGURE 1.2: The balance between accuracy and speed in simulation. Credit: Joshuha Thomas-Wilsker

doi:10.6342/NTU202500790

1.2 Challenges

Generating a 3D point cloud to accurately model energy deposition across spatial coordinates presents several key challenges. Traditional approaches primarily focus on one-dimensional projections, modeling energy as a function of a single spatial dimension. While effective for simplified representations, these methods fail to capture the full complexity of particle interactions. Our model, in contrast, aims to reconstruct the complete three-dimensional energy distribution in a single forward pass, requiring a delicate balance between high-dimensional fidelity and computational efficiency.

To achieve this, we integrate advanced architectural components, including Gaussian Fourier Projection for time encoding and mean-field attention mechanisms with a class token, along with conditional guidance based on incident energy. These features enable precise control over both positional and energy distributions, addressing the intricate dependencies within the 3D spatial domain. However, training a model to learn the complex correlations between spatial coordinates, energy deposition, and incident energy introduces significant computational challenges. Ensuring that the model generalizes well across various particle types and energies, while maintaining efficiency, remains a non-trivial task.

The high-dimensional nature of this generative task demands careful conditioning to reflect realistic energy variations across spatial coordinates. Our approach requires the model to dynamically adjust its predictions based on incident energy, detector response, and local correlations. This complexity leads to a tradeoff: increasing fidelity often incurs substantial computational costs. Optimizing the architecture to maintain accuracy while reducing inference time is a key focus of our work.

Despite these challenges, our optimized approach achieves up to a 500-fold speedup over traditional Geant4-based simulations, offering a scalable alternative suited for next-generation collider experiments. By leveraging modern generative techniques, we not only enhance simulation efficiency but also improve the resolution and realism of synthetic data.

In summary, our model represents a step forward in 3D point cloud generation for high-energy physics simulations. By bridging the gap between scalability and fidelity, we address the computational limitations of conventional methods while enabling high-precision modeling of energy deposition patterns. These advancements pave the way for more efficient and realistic simulations, crucial for meeting the demands of high-luminosity experiments and future discoveries in particle physics.

doi:10.6342/NTU202500790

Chapter 2

Detector

2.1 The Large Hadron Collider (LHC)

Although the Standard Model of particle physics has been remarkably successful up to the 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 energy scales above the TeV range. It consists of a 27-kilometer ring of superconducting magnets and accelerating structures, enabling proton-proton collisions at an unprecedented energy of 13 TeV (with a design energy of 14 TeV). The primary goal of the LHC is to investigate electroweak symmetry breaking, for which the Higgs mechanism is presumed to be responsible, and to search for new physics beyond the Standard Model.

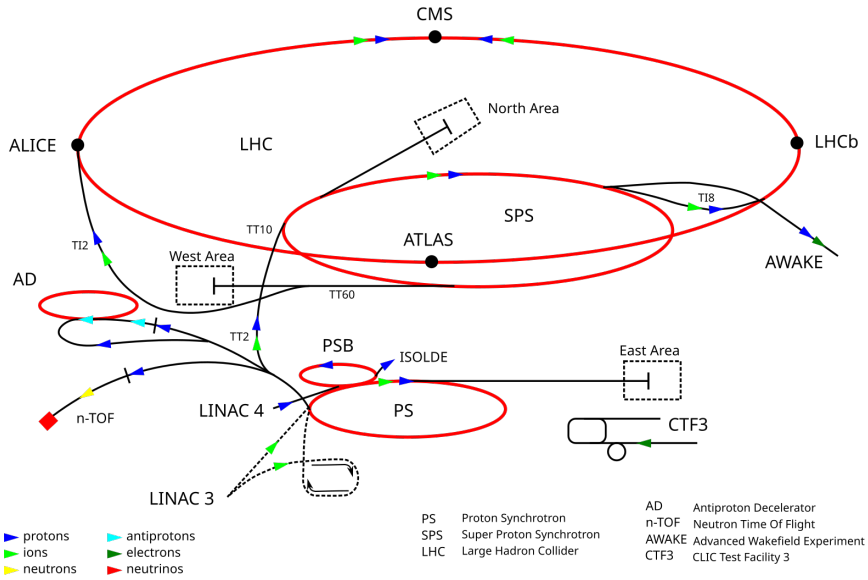


FIGURE 2.1: Schematic of the LHC
[17]

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

doi:10.6342/NTU202500790

ATLAS, LHCb, and ALICE, each optimized for specific research objectives. The injection system consists of the Proton Synchrotron (PS) and the 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 before injection into the Proton Synchrotron Booster (PSB) [18].
- **Proton Synchrotron Booster (PSB):** Strips electrons from H^- ions to produce protons and accelerates them to 2 GeV [19].
- **Proton Synchrotron (PS):** Further increases the beam energy to 26 GeV [20].
- **Super Proton Synchrotron (SPS):** Boosts the energy of protons to 450 GeV before injection into the LHC [21].

Each stage ensures that the beam achieves the required energy, intensity, and quality, culminating in proton-proton collisions at 13.6 TeV in Run 3.

Main Ring

The LHC ring consists of two counter-rotating beam pipes, maintained under ultra-high vacuum conditions to minimize interactions 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 [22].
- **Cryogenics:** The superconducting magnets operate at 1.9 Kelvin (-271), achieved using liquid helium cooling systems [23].

Experimental Sites

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

- **CMS (Compact Muon Solenoid):** Optimized for studying high-energy collisions, precision measurements, and new physics.
- **ATLAS (A Toroidal LHC Apparatus):** A general-purpose detector designed for a broad range of physics exploration.

doi:10.6342/NTU202500790

- **ALICE (A Large Ion Collider Experiment):** Specializes in heavy-ion collisions and studies of the quark-gluon plasma.
- **LHCb (LHC Beauty Experiment):** Dedicated to investigating matter-antimatter asymmetry by analyzing 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 in emergency situations.

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 enable the study of rare processes, such as Higgs boson production.

2.1.2 Technological Challenges

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

The LHC represents the 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. Its design focuses on:

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

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

doi:10.6342/NTU202500790

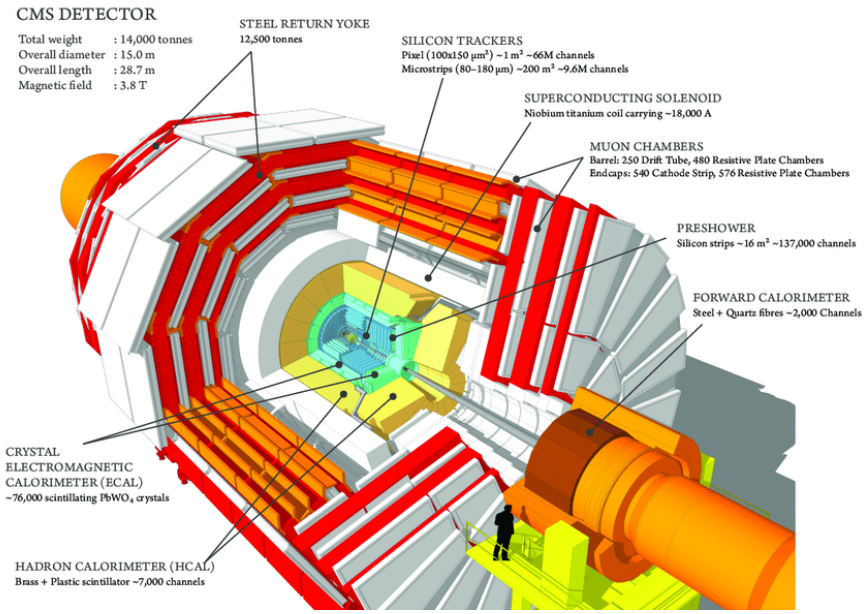


FIGURE 2.2: Exploded view of the CMS detector, showing its main components.

momentum measurements of charged particles. The solenoid is enclosed in a 10,000-tonne iron return yoke, which serves to contain the magnetic field and houses the muon detection system [24]. The CMS muon spectrometer consists of gaseous detectors embedded within the iron return yoke of the superconducting solenoid [25].

To better illustrate the CMS detector, the figure below presents a cross-sectional schematic showcasing its key components: the Silicon Tracker, Electromagnetic Calorimeter (ECAL), Hadron Calorimeter (HCAL), and Superconducting Solenoid. Next, this chapter will delve into the design and performance of the Silicon Tracker, ECAL, HCAL, Muon detector subsystems.

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 particle momentum, 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 $|\eta| < 2.5$ [26].

doi:10.6342/NTU202500790

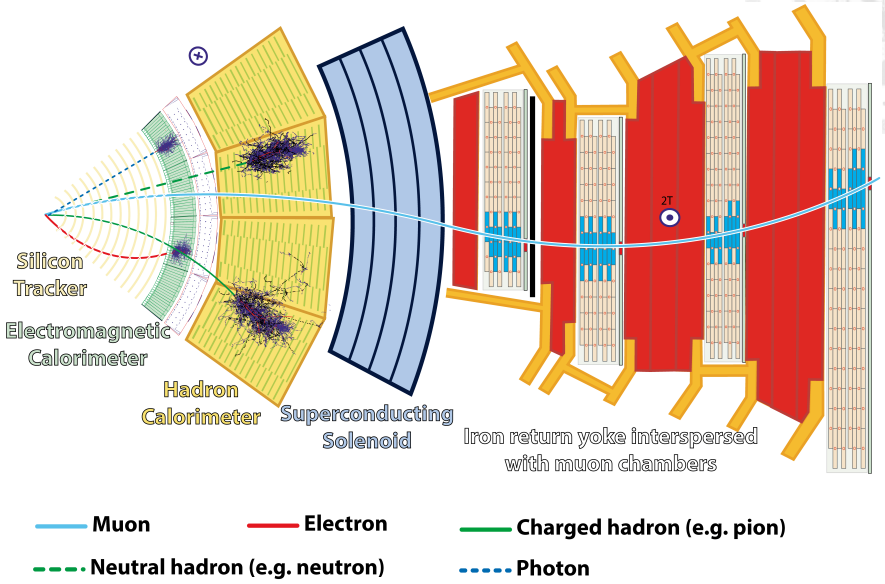


FIGURE 2.3: Cross-sectional schematic of the CMS detector.

The pixel detector is constructed using silicon sensors segmented into millions of tiny pixels, each measuring $100 \times 150\ \mu m^2$.

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 essential for identifying displaced vertices from the decays of short-lived particles such as B-mesons and τ leptons [27].

2.3.2 Silicon Strip Tracker

Surrounding the pixel detector is the silicon strip tracker, which extends 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 [26].

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 trajectory reconstruction [28].

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 its design include:

doi:10.6342/NTU202500790

- **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 GeV/c. This precision enables detailed studies of particle properties, including invariant mass reconstruction and decay vertex identification [26].

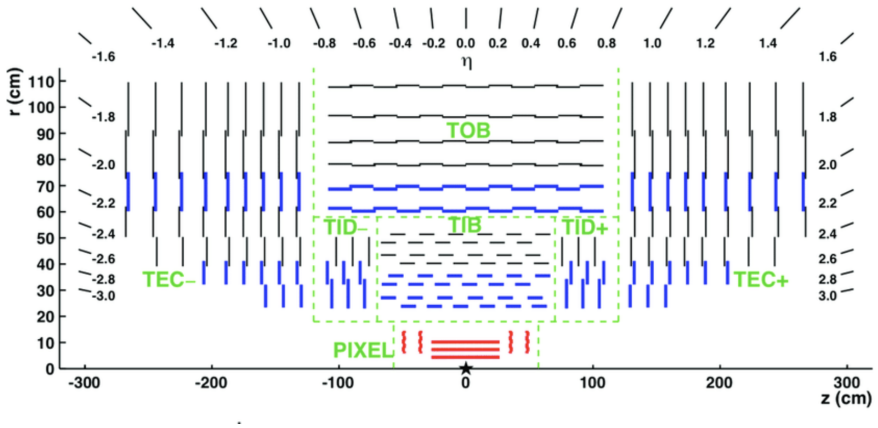


FIGURE 2.4: Cross-sectional schematic of the CMS tracker, showing the pixel and strip components.

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 $p_T \sim 100$ 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

doi:10.6342/NTU202500790

this research, since our dataset focuses primarily on photons and electrons, the ECAL is the main region of interest.

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 [29]. Lead tungstate is chosen due to its high density and short radiation length, allowing 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) in the barrel region, which offer excellent sensitivity and radiation resistance [29].

2.4.2 The ECAL Endcap (EE)

The ECAL Endcap extends the ECAL coverage to higher pseudorapidities, from $|\eta| = 1.479$ to $|\eta| = 3.0$. The endcap region consists of approximately 14,600 $PbWO_4$ crystals, arranged in a geometry optimized for forward physics studies [29]. 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 high-radiation environment in the endcap region 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 [29].

2.4.3 The Preshower Detector

The preshower detector is located in front of the ECAL Endcaps and is designed to enhance discrimination between photons and neutral pions (π^0). It consists of two layers of lead absorbers interleaved with silicon strip sensors [30]. 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 ability to identify isolated photons in a high-particle-density environment [30].

doi:10.6342/NTU202500790

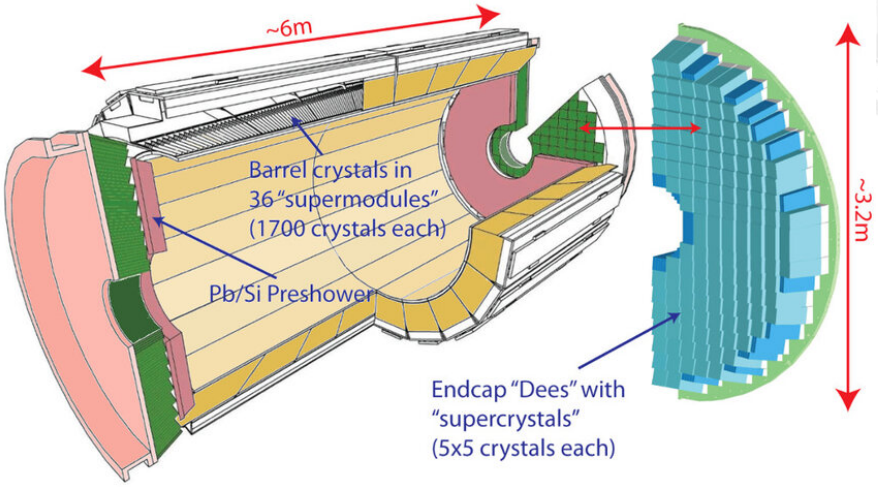


FIGURE 2.5: Structure of the ECAL showing barrel and endcap regions. [31]

2.4.4 Material Choices and Performance

The choice of materials for the ECAL and its preshower detector is driven by their unique properties:

- **High density and short radiation length ($PbWO_4$):** These properties allow electromagnetic showers to be contained within a compact volume, ensuring precise energy measurements.
- **Fast scintillation time ($PbWO_4$):** $PbWO_4$ crystals have a decay time of approximately 25 ns, matching the LHC's bunch crossing interval [29].
- **Radiation hardness ($PbWO_4$):** $PbWO_4$ is resistant to radiation damage, which is essential for maintaining detector performance over extended periods of operation.
- **Preshower detection (Silicon sensors):** Instead of using $PbWO_4$,the preshower detector uses layers of silicon sensors interleaved with lead to induce and detect electromagnetic showers. This enhances photon identification and discrimination.

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 [29]. This resolution allows the ECAL to distinguish between different particle species and

doi:10.6342/NTU202500790

measure their energies with high precision, making it indispensable 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. [32] 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.

doi:10.6342/NTU202500790

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. [32]

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. [33]

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. [34]

doi:10.6342/NTU202500790

2.5.5 Material Choices and Their Impact

The material choices for the HCAL are optimized to balance density, radiation hardness, and signal quality, ensuring effective energy containment and long-term performance across different detector regions.

- **HCAL Barrel (HB) and Endcap (HE)** - Brass and Plastic Scintillators: HB and HE use brass as the absorber for efficient hadron stopping and shower initiation, with plastic scintillators as the active medium for fast, high-yield light collection. HE employs thicker brass layers to handle higher-momentum particles at larger pseudorapidities.
- **HCAL Outer (HO)** - Steel and Plastic Scintillators: HO repurposes the steel return yoke of the solenoid as an absorber, with plastic scintillators capturing residual hadronic energy, improving energy containment and jet resolution.
- **HCAL Forward (HF)** - Steel and Quartz Fibers: HF, operating in a high-radiation environment, uses steel for absorption and quartz fibers to generate Cherenkov light for robust, radiation-resistant detection.

The strategic selection of these materials across the HCAL regions ensures optimal energy measurement, jet reconstruction, and missing transverse energy (E_T^{miss}) calculations. By tailoring the material composition to each pseudorapidity range, the HCAL effectively captures hadronic energy while withstanding the challenging conditions of high-energy collisions in the CMS detector.

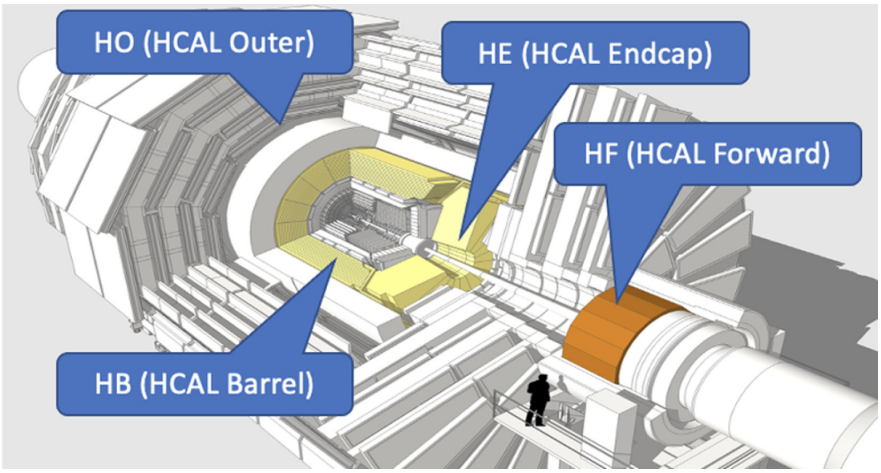


FIGURE 2.6: Schematic of the HCAL with barrel, endcap, and forward sections. [35]

doi:10.6342/NTU202500790

2.5.6 Performance

The HCAL provides energy resolution of: [36]

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

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

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 [37].

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 [37].

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 [37].

doi:10.6342/NTU202500790

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 [37].

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 TeV/c, enabling precise measurements of high-momentum muons [37]. This capability is critical for identifying rare processes, such as those involving heavy bosons or new particles.

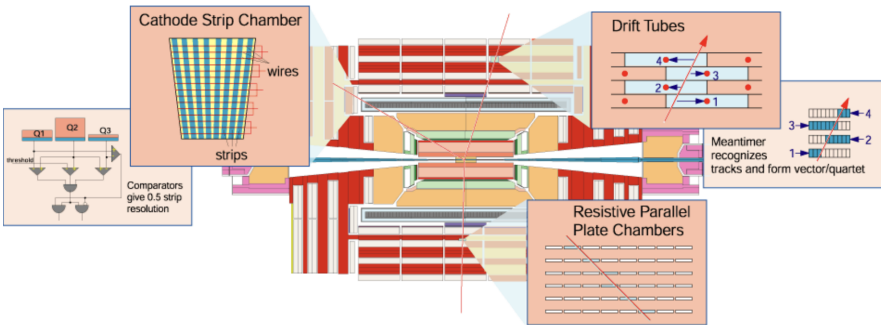


FIGURE 2.7: CMS Muon System layout, showing DTs, CSCs, and RPCs. [38]

The muon system achieves momentum resolution of $\Delta p/p \sim 10\%$ at 1 TeV/c, contributing significantly to global track reconstruction. [39]

doi:10.6342/NTU202500790

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).

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 [40]. 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 [40].

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 [40]. 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

 doi:10.6342/NTU202500790

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 [41].

CE-H: Hadronic Section The CE-H of the CMS High-Granularity Calorimeter (HGCAL) measures hadronic energy deposition and aids jet reconstruction at the HL-LHC. It features a sampling calorimeter with stainless steel absorber plates interleaved with silicon sensors in high-radiation areas and scintillator tiles with on-tile SiPMs in lower-radiation regions.

To ensure stability, copper cooling plates with biphase CO_2 coolant maintain operations at -35C, managing up to 125 kW of power dissipation. The 600-tonne structure is engineered for high precision, incorporating low-power, high-dynamic-range electronics and titanium wedges for support.

With high-granularity readout, radiation-hard materials, and advanced cooling, the CE-H delivers precise hadronic energy measurements, ensuring readiness for next-generation HL-LHC physics. [42].

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.

doi:10.6342/NTU202500790

- **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 [41].

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.

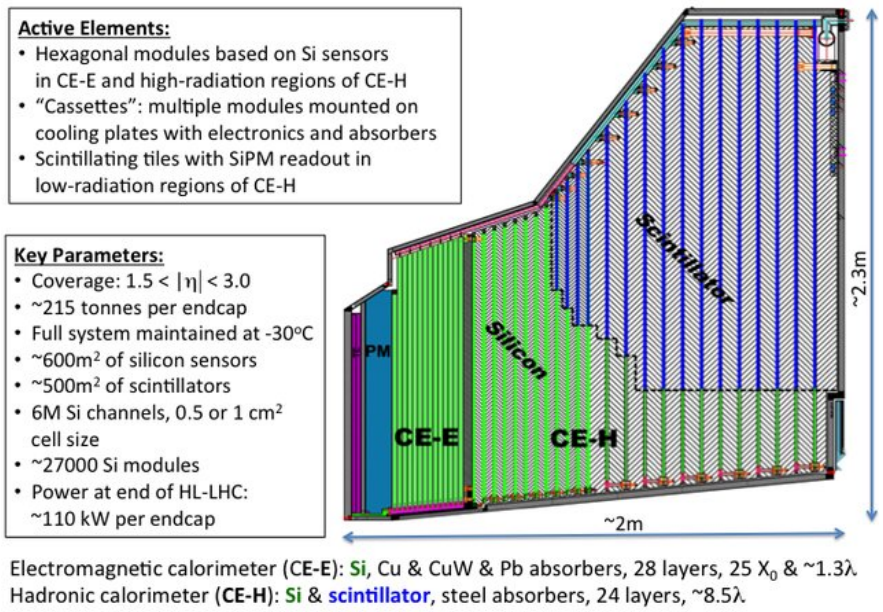


FIGURE 2.8: Schematic of the HGCal showing its layered structure and segmentation. (Image credit: CMS Collaboration)
[43]

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.

doi:10.6342/NTU202500790

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.

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

Chapter 3

Dataset

3.1 Geant4 Simulation

Geant4 is a powerful and widely used simulation toolkit for modeling particle interactions with matter. It provides detailed simulations of detector geometry, material interactions, and physics processes, enabling accurate predictions of detector responses. In the CMS experiment, Geant4 plays a crucial role in validating experimental results and designing detector upgrades such as the High-Granularity Calorimeter (HGCal).

3.1.1 Physics Processes

Geant4 provides 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 [44].

3.1.2 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 [44].

3.1.3 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.

doi:10.6342/NTU202500790

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.

3.1.4 Applications in HGCal Development

Geant4 has played a crucial role in optimizing the design of the HGCal. Through simulations of various configurations and material choices, researchers have fine-tuned the detector to achieve optimal performance in terms of energy resolution, granularity, and radiation tolerance. Additionally, these simulations aid in the development of reconstruction algorithms and calibration techniques specifically tailored to the unique characteristics of the HGCal [44].

Below is a demonstration of a Geant4 simulation for the HGCal, illustrating the interaction of a 20 GeV π^+ particle within the detector. An interesting aspect of Geant4 visualizations is the use of distinct colors to represent different particle types and interactions. In this simulation, charged particles are labeled in green, neutral particles in red, and interactions within the calorimeter in blue.

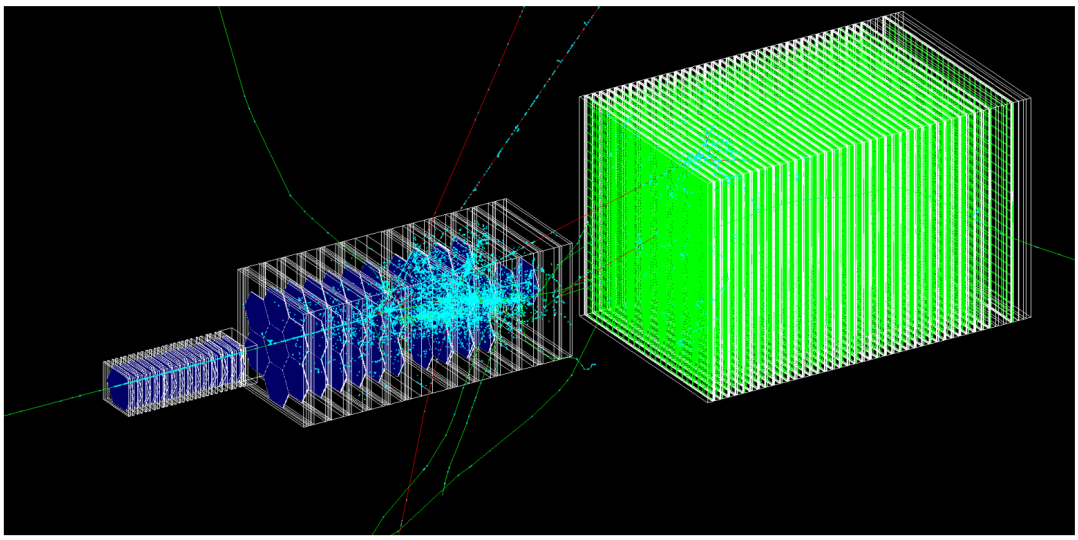


FIGURE 3.1: Visualization of a Geant4 simulation for the HGCal, showing particle showers in the calorimeter layers. (Image credit: Geant4 Collaboration)
[45]

doi:10.6342/NTU202500790

3.1.5 Challenges of Geant4

While Geant4 is a powerful and widely used simulation toolkit, it presents several challenges that impact its efficiency and usability in large-scale physics experiments.

- **Computational Intensity:**
  Geant4 simulations require significant computational resources, particularly for high-energy physics experiments involving dense materials and complex interactions. The need to track billions of particles through detailed detector geometries makes large-scale simulations computationally expensive [46].
- **High Complexity:**
  The object-oriented design of Geant4 provides flexibility but also introduces a steep learning curve. Users must understand multiple modules, including geometry definitions, physics processes, and tracking systems, which can slow down the implementation of new simulations [47].
- **Scaling Issues:**
  With increasing experimental data rates, such as those expected in the High-Luminosity Large Hadron Collider (HL-LHC) phase, Geant4 faces challenges in maintaining simulation accuracy while keeping computational costs manageable. Machine learning approaches, such as diffusion models, are being explored as potential solutions to accelerate Geant4-like simulations while retaining high fidelity.
- **Validation and Maintenance:**
  Ensuring the accuracy of Geant4 requires continuous validation against experimental data. Physics models must be regularly updated and optimized to reflect the latest theoretical and experimental findings. Additionally, software maintenance and debugging add further complexity to large-scale implementations [48].

Despite these challenges, Geant4 remains an essential tool in particle physics and detector simulations, with ongoing research focusing on improving its computational efficiency and expanding its applicability to future high-energy physics experiments.

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.

doi:10.6342/NTU202500790

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 [49].

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 single-particle shower 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.

- Voxel structure:
  - 368 voxels for photons
  - 533 voxels for pions
- Incident energy levels: 15 discrete values, spanning 256 MeV to 4 TeV in powers of two.
- Number of events: 10,000 per energy level, except at higher energies where limited statistics are available.

doi:10.6342/NTU202500790

This dataset serves as a baseline for evaluating generative models on relatively simple detector geometries and energy distributions.

Dataset 2: Multi-Layer Geometry with Electrons

This dataset simulates electron showers in a concentric cylindrical calorimeter, which consists of:

- 45 layers, each containing:
  - Active material (silicon)
  - Passive material (tungsten)
- Voxel segmentation:
  - 9 radial bins $\times$ 16 angular bins per layer
  - Total: 6,480 voxels ($45 \times 16 \times 9$)
- Incident energies: Sampled from a log-uniform distribution in the range 1 GeV to 1 TeV.
- Number of events: 100,000

This dataset introduces high-granularity segmentation, challenging models to accurately capture energy depositions across a complex detector geometry.

Dataset 3: High-Granularity Calorimeter Geometry

Dataset 3 extends the complexity of Dataset 2 by significantly increasing granularity:

- 45-layer calorimeter with active (silicon) and passive (tungsten) material.
- Higher voxel segmentation:
  - 18 radial bins $\times$ 50 angular bins per layer
  - Total: 40,500 voxels ($45 \times 50 \times 18$)
- Electron energies: Log-uniformly sampled between 1 GeV and 1 TeV.
- Number of events: 50,000

This dataset is specifically designed to evaluate the ability of generative models to generalize and simulate realistic high-granularity calorimeters, such as those planned for HL-LHC and future collider experiments.

doi:10.6342/NTU202500790

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.

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

doi:10.6342/NTU202500790

Chapter 4

Algorithm

4.1 Score-based Diffusion Model

Before diffusion models were introduced, generative models were primarily based on Variational Autoencoders (VAEs) and Generative Adversarial Networks (GANs). While VAEs are effective for compressing data, they struggle to generate high-quality, diverse samples due to their reliance on sampling from a normal distribution in the latent space. GANs, on the other hand, have demonstrated success in generating realistic samples but can be challenging to train and prone to mode collapse.

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.

4.1.1 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

doi:10.6342/NTU202500790

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 [50], 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:

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

4.1.2 Denoising Diffusion Probabilistic Model (DDPM)

Next, we are going to introduce the second method for diffusing models, the Denoising Diffusion Probabilistic Model (DDPM) [51]. Unlike SMLD, DDPM incorporates a

doi:10.6342/NTU202500790

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. [52] and Ho et al. [51], 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.

4.2 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 [50], 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:

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

doi:10.6342/NTU202500790

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.

4.3 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

 doi:10.6342/NTU202500790

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": [53]

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

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:

doi:10.6342/NTU202500790

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

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:

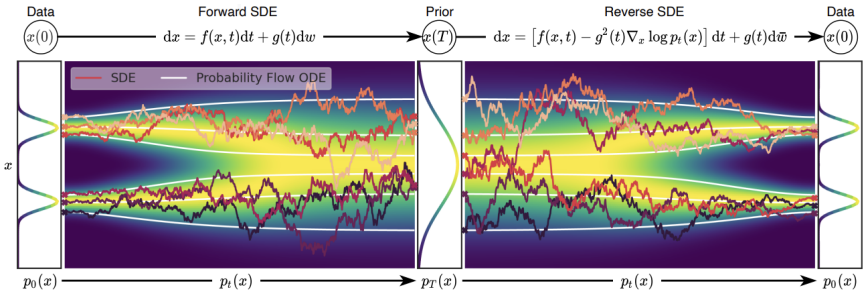


FIGURE 4.1: Forward and Backward Processes in Diffusion Models (The picture is from Song and Ermon (2019))

4.4 Loss Function for Score-Based Models

After deriving the backward process, the final step is to compute $\nabla_x \log p_{\text{data}}(x)$, which is where machine learning plays a crucial role. A neural network can be used to approximate this score function—the gradient of the log-density of the data distribution. To train the network, we minimize the following loss function:

$$L = \frac{1}{2}\mathbb{E}_{p_{data}(x)}\left[\|s_\theta(x,t) - \nabla_x \log p_{data}(x)\|_2^2\right]. \tag{4.16}$$

Since the true score function is not tractable, an alternative approach known as **Denoising Score Matching (DSM)** [54] is used. DSM does not estimate the score of the clean data directly but instead estimates the score of a perturbed version of the data. This is achieved by corrupting the original data with Gaussian noise:

$$p_\sigma(\tilde{x}|x) = \mathcal{N}(x, \sigma^2 I), \tag{4.17}$$

doi:10.6342/NTU202500790

where $\tilde{x}$ represents the perturbed data. The perturbed data distribution is then given by:

$$p_\sigma(\tilde{x}) = \int p_{\text{data}}(x) p_\sigma(\tilde{x}|x) dx. \tag{4.18}$$

To train a model to estimate the score function, we minimize the following objective:

$$L = \frac{1}{2}\mathbb{E}_{p_\sigma(\tilde{x}|x)p_{\text{data}}(x)}\left[\|s_\theta(\tilde{x}) - \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x)\|_2^2\right]. \tag{4.19}$$

Using the Gaussian perturbation kernel, we can compute the score function analytically:

$$\nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x) = \frac{x - \tilde{x}}{\sigma^2}. \tag{4.20}$$

Since $(x - \tilde{x}) \sim \mathcal{N}(0, \sigma^2)$, this follows:

$$\frac{x - \tilde{x}}{\sigma^2} \sim \frac{\mathcal{N}(0, \sigma^2)}{\sigma^2} = \mathcal{N}(0, \frac{1}{\sigma^2}). \tag{4.21}$$

Thus, we can rewrite the expectation term as:

$$\mathcal{N}(0, 1) = \sigma \cdot \nabla_{\tilde{x}} \log p_\sigma(\tilde{x}|x). \tag{4.22}$$

This leads to a practical loss function:

$$L = \frac{1}{2}\mathbb{E}\left[\|\sigma \cdot s_\theta(\tilde{x}, t) - \mathcal{N}(0, 1)\|_2^2\right]. \tag{4.23}$$

In implementation, this loss function simplifies to:

$$L = (\sigma \cdot s(x, t) - \text{torch.normal}(0, 1))^2. \tag{4.24}$$

The advantage of this approach is that it enables efficient computation without requiring knowledge of the full data distribution, making it a practical choice for training score-based generative models.

doi:10.6342/NTU202500790

4.5 VE, VP SDEs

4.5.1 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.

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:

$$x_i = x_{i-1} + \sqrt{\sigma_i^2 - \sigma_{i-1}^2} z_{i-1}, \quad i = 1, 2, \ldots, N, \tag{4.25}$$

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 4.31 with $\Delta t = \frac{1}{N}$ gives:

$$x(t + \Delta t) = x(t) + \sqrt{\sigma^2(t + \Delta t) - \sigma^2(t)} z(t) \approx x(t) + \sqrt{\frac{d\sigma^2(t)}{dt}} \Delta t z(t), \tag{4.26}$$

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

$$dx = \sqrt{\frac{d\sigma^2(t)}{dt}} dw. \tag{4.27}$$

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

$$d\mathbf{x} = \sigma_{\min} \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)^t \sqrt{2 \log \left(\frac{\sigma_{\max}}{\sigma_{\min}} \right)} d\mathbf{w}, \quad t \in (0, 1). \tag{4.28}$$

doi:10.6342/NTU202500790

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

$$\mathbf{x}_i = \sqrt{1-\beta_i}\mathbf{x}_{i-1} + \sqrt{1-\beta_i}z_{i-1}, \quad i = 1, 2, \ldots, N \tag{4.29}$$

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 4.31 as follows:

$$\mathbf{x}_i = \sqrt{1-\bar{\beta}_i}\mathbf{x}_{i-1} + \sqrt{1-\bar{\beta}_i}z_{i-1}, \quad i = 1, 2, \ldots, N \tag{4.30}$$

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 4.32 gives:

$$\begin{aligned}
\mathbf{x}(t+\Delta t) &= \sqrt{1-\beta(t+\Delta t)}\mathbf{x}(t) + \sqrt{1-\beta(t+\Delta t)}z(t) \\
&\approx \mathbf{x}(t) - \frac{1}{2}\beta(t+\Delta t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t+\Delta t)\Delta t}\mathbf{z}(t) \\
&\approx \mathbf{x}(t) - \frac{1}{2}\beta(t)\Delta t\mathbf{x}(t) + \sqrt{\beta(t)\Delta t}\mathbf{z}(t)
\end{aligned} \tag{4.31}$$

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

$$d\mathbf{x} = -\frac{1}{2}\beta(t)\mathbf{x}\,dt + \sqrt{\beta(t)}\,d\mathbf{w}. \tag{4.32}$$

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

$$d\mathbf{x} = -\frac{1}{2}\left(\beta_{\min} + t(\beta_{\max} - \beta_{\min})\right)\mathbf{x}\,dt + \sqrt{\beta_{\min} + t(\beta_{\max} - \beta_{\min})}\,d\mathbf{w}, \quad t \in (0, 1). \tag{4.33}$$

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):

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

doi:10.6342/NTU202500790

4.5.2 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 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. 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**
    $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$

doi:10.6342/NTU202500790

$\mathbf{x}_i \leftarrow \mathbf{x}_i + \epsilon_i s_\theta^* (\mathbf{x}_i, i) + \sqrt{2\epsilon_i}\mathbf{z}$

end for

end for

return $\mathbf{x}_0$

where ϵ is defined as

$$\epsilon = 2r^2 \frac{\|\mathbf{z}\|_2^2}{\|s_\theta\|_2^2} \tag{4.34}$$

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

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

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

doi:10.6342/NTU202500790

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 revolutionized deep learning by enabling efficient and scalable processing of sequential and structured data. Originally introduced for natural language processing, Transformers have since demonstrated remarkable versatility across various domains, including computer vision, time-series analysis, and scientific computing. Unlike traditional sequence models such as Recurrent Neural Networks (RNNs), which process data sequentially, Transformers utilize a self-attention mechanism that allows for parallel computation and long-range dependencies.

In high-energy physics, where data from particle detectors is vast, multidimensional, and often exhibits complex dependencies, Transformers provide significant advantages in both accuracy and efficiency. Their ability to capture intricate relationships between detector hits without being constrained by sequential processing makes them particularly suitable for simulations of particle showers, energy depositions, and collision dynamics. In the next section, we will explore the evolution from RNNs to

 doi:10.6342/NTU202500790

Transformers, highlighting the limitations of sequential models and how Transformers address these challenges.

5.1.2 The Evolution from RNNs to Transformers

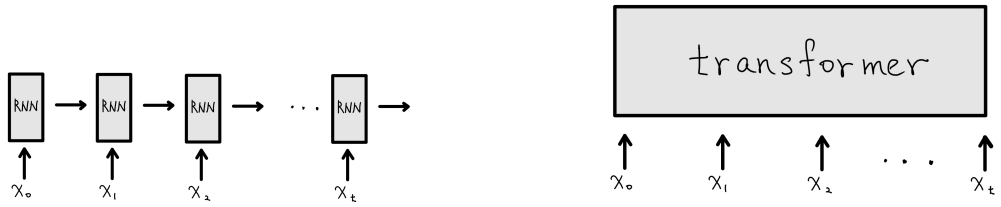
Figure 5.1 illustrates the fundamental differences in how RNNs and Transformers process sequential data. In the RNN model (A), the sequence is processed step by step, meaning that each input token x_0 is first passed into an RNN unit, which updates its internal state before moving to the next token x_1. At each step, the model relies on the hidden state from the previous timestep, which acts as a summary of all prior inputs. This recurrent dependency means that information flows sequentially through the network, making it impossible to process all tokens simultaneously. Instead, the model must first process x_0, then x_1, followed by x_2, and so on until x_t.

This sequential nature introduces several challenges. First, it creates a bottleneck in computation, as each step must wait for the previous step to finish before proceeding. This makes training slow and inefficient, especially for long sequences. Second, as the sequence length increases, information from earlier tokens may become difficult to retain, leading to what is known as the vanishing gradient problem. Since each update relies on a chain of transformations through multiple time steps, the influence of initial inputs weakens over time, making it difficult for RNNs to capture long-range dependencies effectively.

In contrast, the Transformer model (B) processes the entire sequence at once, leveraging a self-attention mechanism that allows every token to directly interact with all others. Instead of passing information step by step as in RNNs, the Transformer constructs global dependencies in a single operation, meaning that each token x_t can immediately access information from any other token, regardless of its position in the sequence. This parallel computation greatly accelerates training and removes the reliance on sequential updates.

For high-energy physics simulations, this difference is particularly relevant. Each input sequence in our case represents multiple showers, where each shower contains numerous hits with various energy levels and spatial coordinates. Unlike natural language, where word order matters, detector data does not follow a strict sequential pattern. An RNN would impose an artificial structure on the data, potentially obscuring meaningful relationships. Transformers, by contrast, can naturally capture interactions between all hits in a shower, ensuring that the model fully exploits the complex dependencies inherent in high-energy physics simulations. This ability to efficiently model both local and global relationships is one of the primary reasons Transformers have become the preferred choice over RNNs in modern deep-learning applications.

doi:10.6342/NTU202500790



(A) RNN Model (B) Transformer Model

FIGURE 5.1: Comparison of RNN and Transformer architectures.

5.1.3 Self-Attention Mechanism

Self-attention is the core mechanism that allows Transformers to process and understand relationships within a sequence efficiently. Unlike traditional models that rely on sequential computations, self-attention allows each token in an input sequence to consider every other token simultaneously. This mechanism is particularly well-suited for high-energy physics applications, where complex dependencies exist between detector hits, and the order in which data is collected does not necessarily dictate meaningful relationships.

At its essence, the **attention mechanism** determines how much focus each token should place on other tokens when computing its representation. This is achieved by transforming each input into three vectors: a **query** (Q), a **key** (K), and a **value** (V). The query represents what a token is looking for in other tokens, the key encodes what information each token contains, and the value carries the actual content that gets passed forward. The attention scores are computed by measuring the similarity between queries and keys, which determines how much influence one token should have on another.

The scaled dot-product attention mechanism follows the equation:

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

where d_k is the dimension of the key vectors, used as a scaling factor to stabilize gradients.

For detector simulations, this mechanism provides a significant advantage. Unlike models that process data sequentially, Transformers can immediately establish long-range dependencies, capturing interactions between hits that may be spatially distant but physically correlated. This ability to dynamically adjust attention across

 doi:10.6342/NTU202500790

the dataset ensures that important features are preserved, leading to more accurate and efficient simulations.

Figure 5.4a provides a visualization of this process, illustrating how input tokens are transformed and passed through attention layers. While the details of the computation involve matrix multiplications and scaling factors, the key idea remains straightforward: **each token learns to selectively focus on the most relevant information, allowing the model to capture both local and global dependencies simultaneously**. This capability is what makes Transformers particularly powerful, not only in natural language processing but also in scientific applications where capturing intricate relationships is essential.

5.1.4 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.

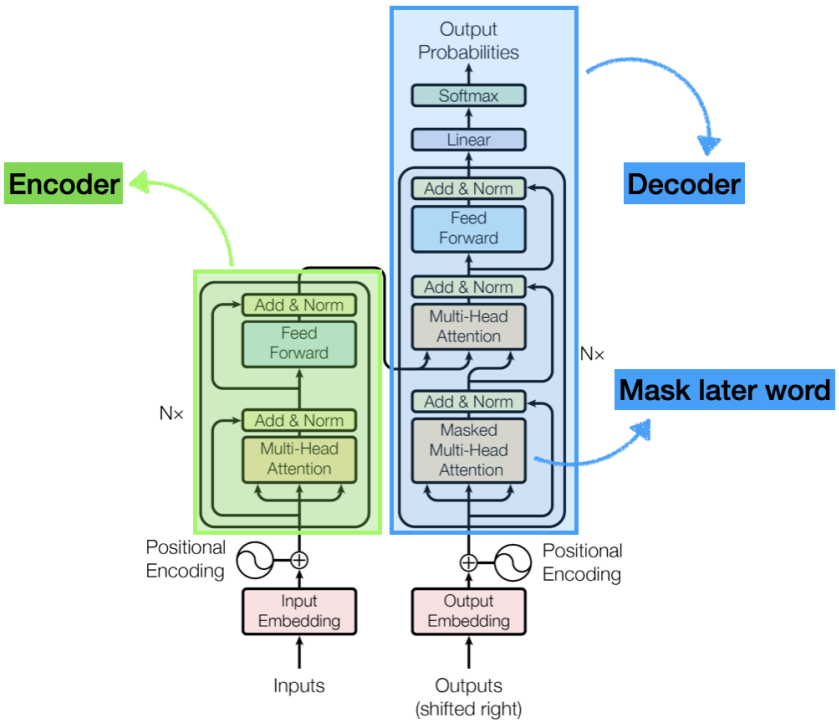


FIGURE 5.2: The structure of the original Transformer model. Adapted from *"Attention is All You Need,"* with additional annotations.

doi:10.6342/NTU202500790

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.

5.1.5 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. Given that our datasets exhibit rotational symmetry, an encoder-only model is the optimal choice, as it efficiently extracts and encodes essential features without introducing the additional complexity of a generative decoder. By focusing solely on representation learning, the encoder architecture ensures that the model captures the intricate relationships within the data while maintaining computational efficiency.

5.2 Our Model Structure

Our model architecture builds upon the Transformer framework, with specific modifications to optimize performance in detector simulations, as shown in Figure 5.3.

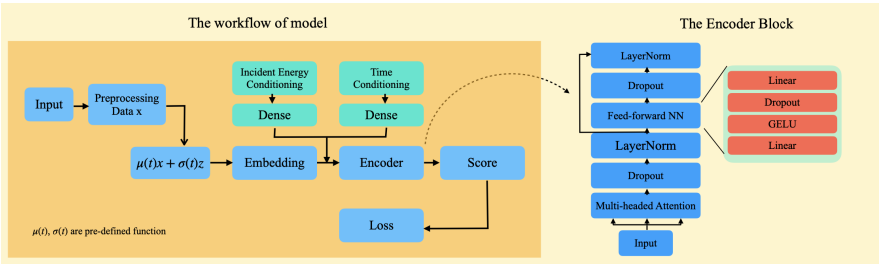


FIGURE 5.3: Custom Transformer model structure for detector simulations.

We incorporate **Gaussian Fourier Projections** [55] to effectively encode temporal information, dense layers to transform conditional variables, and **mean-field attention** [56] to efficiently aggregate global context. These architectural choices enable our model to capture complex dependencies, thereby enhancing the fidelity and accuracy of simulation outcomes.

doi:10.6342/NTU202500790

5.2.1 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:

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

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.

5.2.2 Mean-Field Attention in Detector Simulation

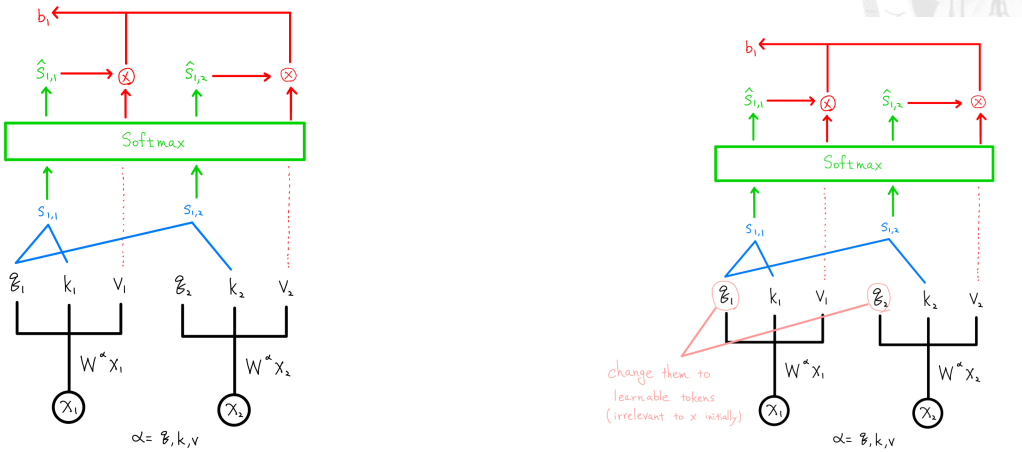
Our model utilizes a variation of self-attention called **mean-field attention**. Unlike traditional self-attention, mean-field attention employs a class token to aggregate information from all tokens, creating a global summary. This reduces computational complexity while preserving essential global 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 5.4 provides a comparison between self-attention and mean-field attention mechanisms.

5.3 Conclusion

Our custom Transformer model leverages specialized architectural choices to optimize performance in high-energy physics simulations. Key modifications include **Gaussian**

doi:10.6342/NTU202500790



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

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

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.

doi:10.6342/NTU202500790

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.

doi:10.6342/NTU202500790

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:

 doi:10.6342/NTU202500790

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

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:

doi:10.6342/NTU202500790

- **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.

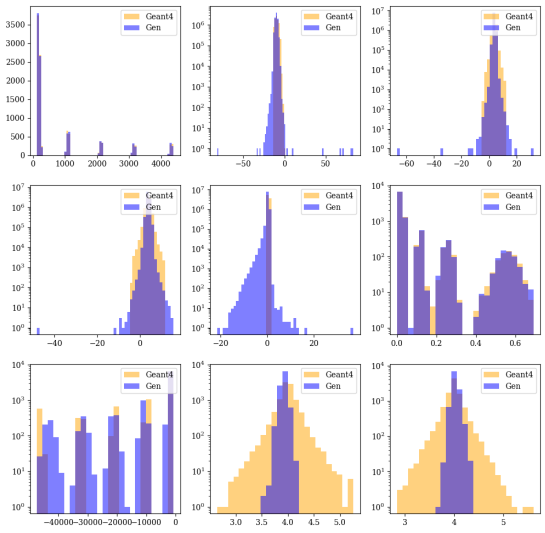


FIGURE 6.1: RobustScaler

- **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:

doi:10.6342/NTU202500790

- **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.

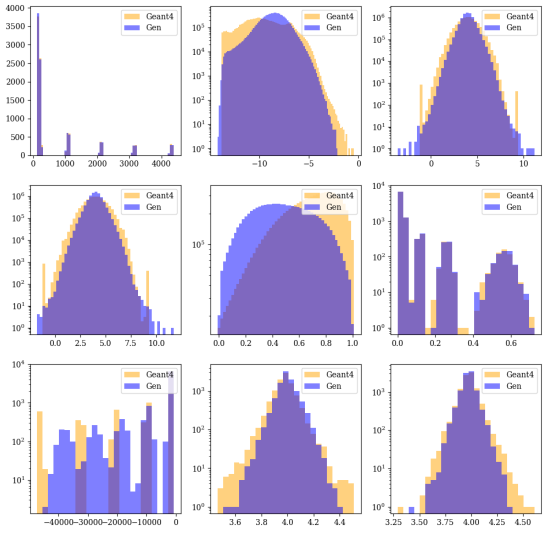


FIGURE 6.2: QuantileTransformer

- **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.

doi:10.6342/NTU202500790

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

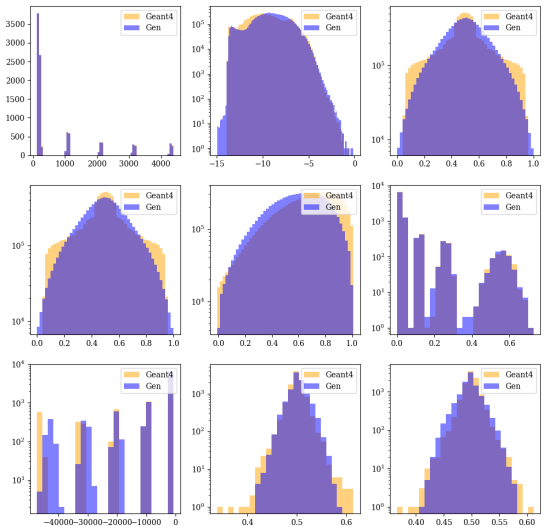


FIGURE 6.3: Exponential Transformation

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 [57]. 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 [58] for our calculations. The FID score is calculated as follows:

doi:10.6342/NTU202500790

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.

doi:10.6342/NTU202500790

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.

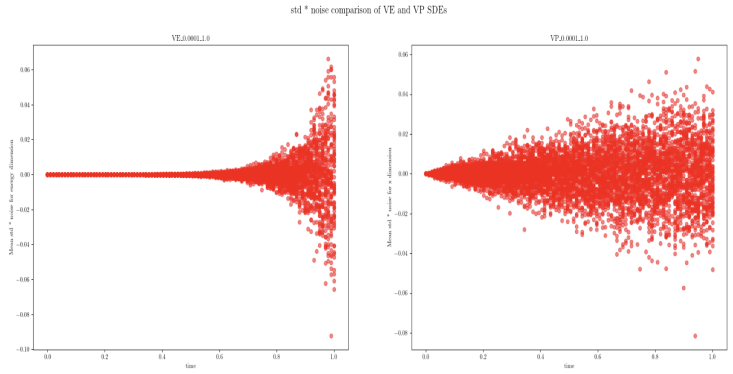


FIGURE 6.4: Comparison of VE and VP methods for both $\sigma_{max} = 1, \sigma_{min} = 0.0001$

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.

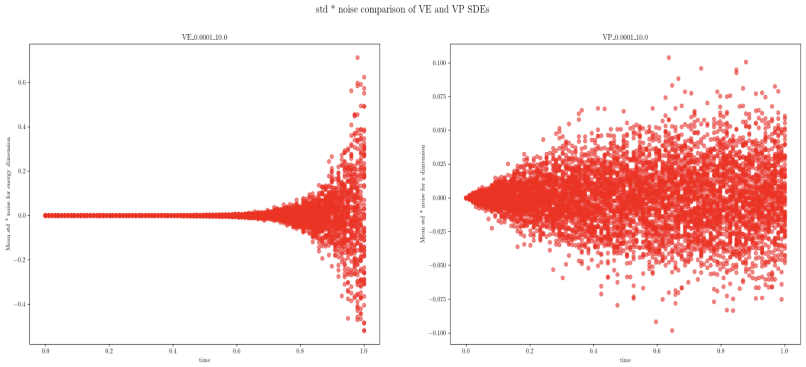


FIGURE 6.5: Comparison of VE and VP methods for both $\sigma_{max} = 5, \sigma_{min} = 0.0001$

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

doi:10.6342/NTU202500790

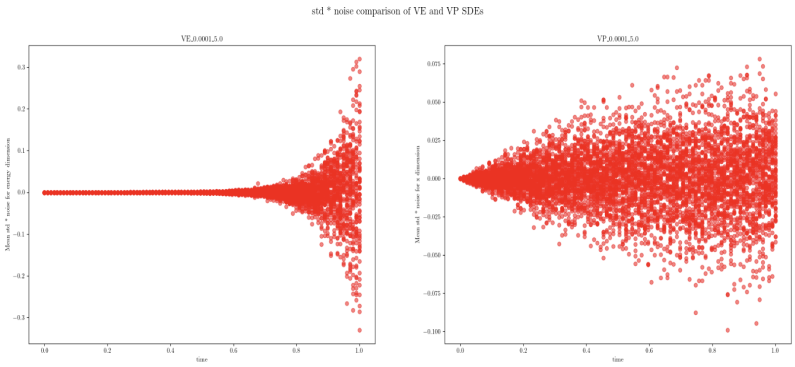


FIGURE 6.6: Comparison of VE and VP methods for both $\sigma_{max} = 10, \sigma_{min} = 0.0001$

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.

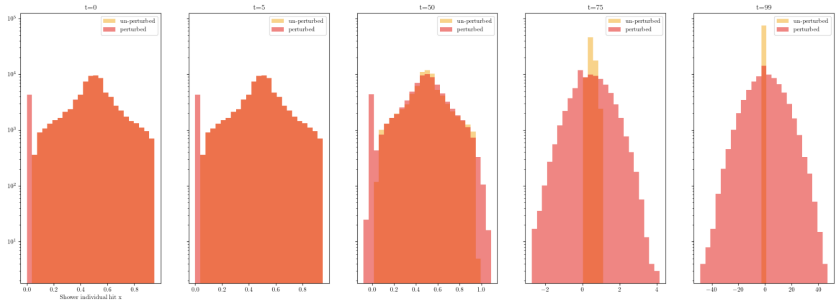


FIGURE 6.7: The distribution of the data after adding the noise using VE method.

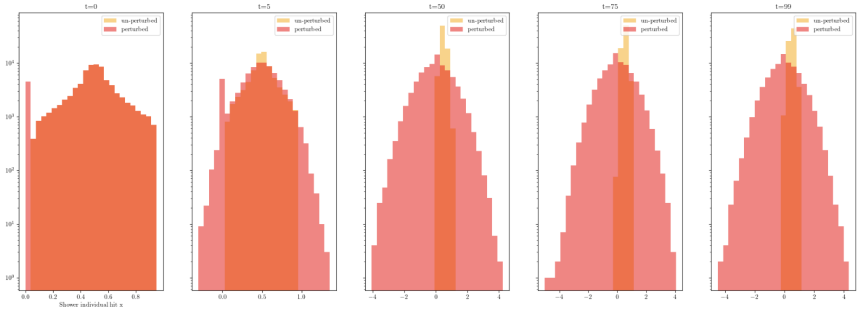


FIGURE 6.8: The distribution of the data after adding the noise using VP method.

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.

doi:10.6342/NTU202500790

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.

FID	**10**	**5**	**1**	**0.5**
VE	0.0312440	0.0349802	0.0224611	0.0190552
VP	0.0058973	10.533640	104.77893	0.0107441

FID_e	**10**	**5**	**1**	**0.5**
VE	0.0334921	0.0059905	0.0158524	0.0195042
VP	0.0356144	8.2458230	84.672020	0.0225893

FID_x	**10**	**5**	**1**	**0.5**
VE	0.0001294	0.0001524	0.0001418	0.0001736
VP	0.0001456	2.1163020	114.35691	0.0001689

FID_z	**10**	**5**	**1**	**0.5**
VE	0.0007689	0.0006219	0.0007465	0.0012151
VP	0.0015260	1.7053030	27.805210	0.0011892

TABLE 6.1: FID, FID_e, FID_x, and FID_z values for different SDE, σ_{max}, and σ_{min}.

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

 doi:10.6342/NTU202500790

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 abservation, 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.

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

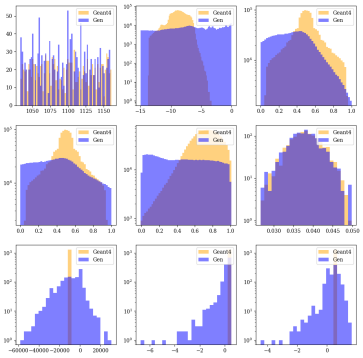
As shown in the results, the choice of σ_{max} and σ_{min} significantly impacts the performance of the model. Larger σ_{max} values can improve the diversity of the data and enhance the model's generalization ability. However, setting σ_{max} too high can lead to noisy samples and hinder the model's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

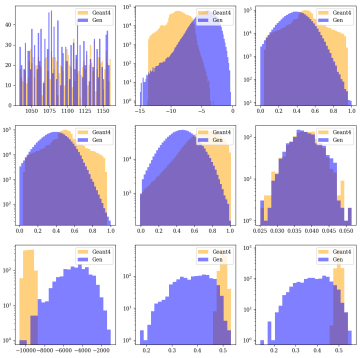
Besides, the parameters mentioned above there are also a lot of other parameters that can affect the performance of the model or the memory allocation. Thus, we conducted a parameter sweeping study using `wandb`. We experimented with various learning rates, batch sizes, and hidden dimensions. Our findings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

- Learning rate: 0.0003

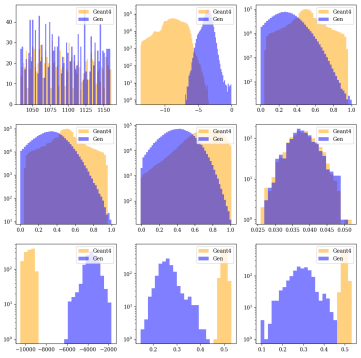
doi:10.6342/NTU202500790



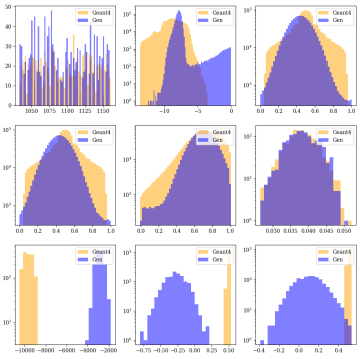
(A) VP with $\sigma_{max} = 20$



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



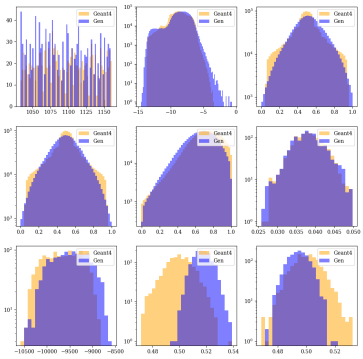
(C) VP with $\sigma_{max} = 5$



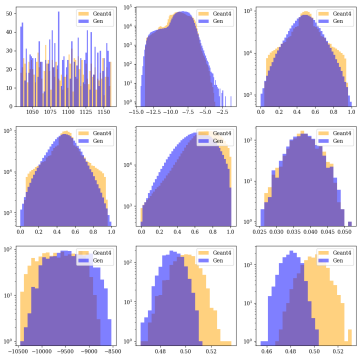
(D) VP with $\sigma_{max} = 1$

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

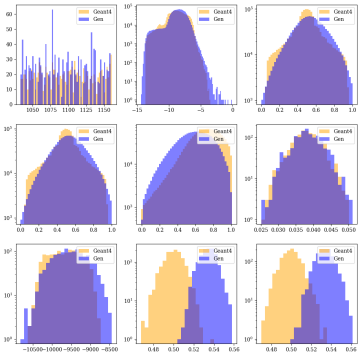
doi:10.6342/NTU202500790



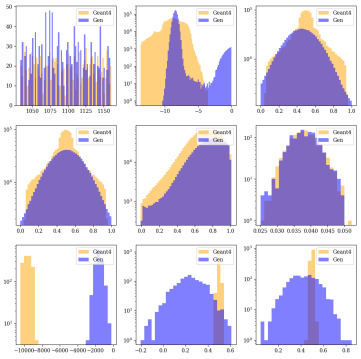
(A) VE with $\sigma_{max} = 20$



(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.

doi:10.6342/NTU202500790

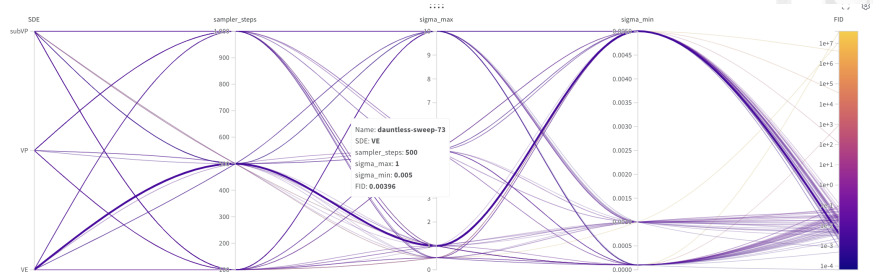


FIGURE 6.11: The result of different σ_{max} and σ_{min} in VE.

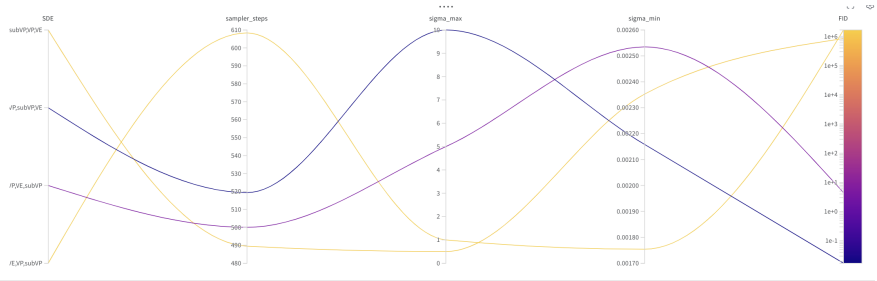


FIGURE 6.12: The result of fig 6.11, but grouped by σ_{max} in VE.

- 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

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

doi:10.6342/NTU202500790

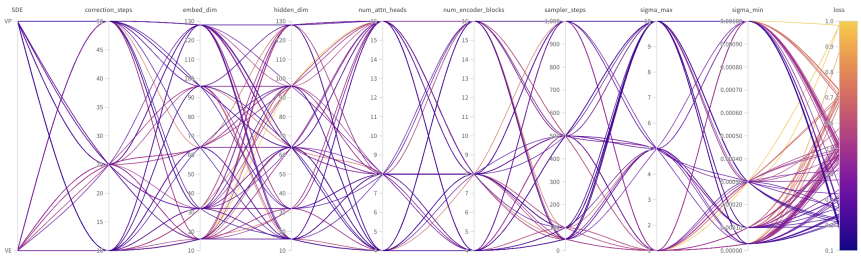


FIGURE 6.13: Visualization of parameter sweeping results.

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:

$$L = L_{\text{Original}} + \lambda L_{\text{cor}^2}, \tag{6.6}$$

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:

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

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

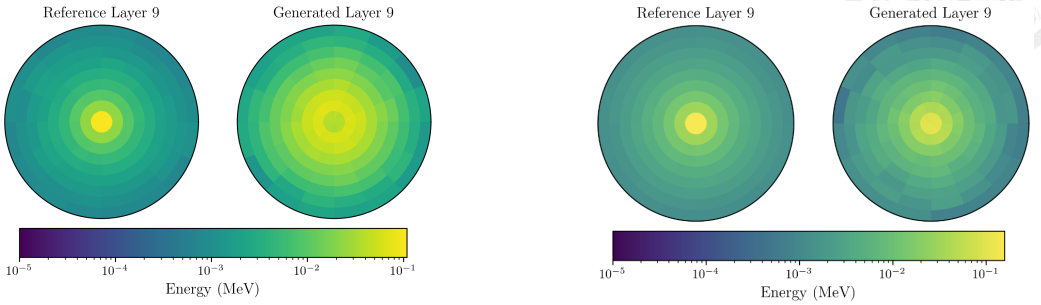
The reason why we don't apply the correlation term between energy and radius is that the relation between them is by experienced, although everyone would expect the result, it's not solid, we don't want to bias our model, or you can say we don't want to tell the answer of the relation to our model.

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.

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 6.14.

doi:10.6342/NTU202500790



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

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

FIGURE 6.14: The Comparison Picture after using QuantileTransformer.

6.7 Conditioning Issue

6.7.1 Incident energy

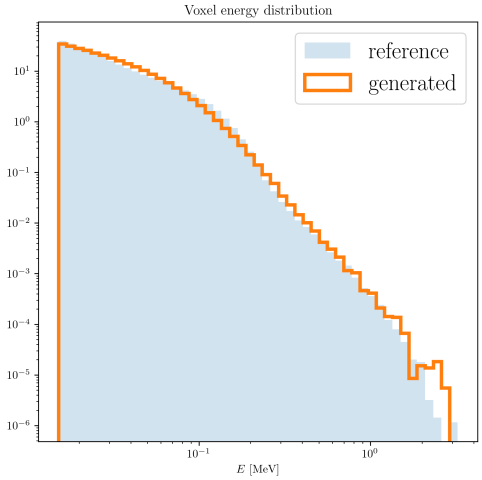
With the optimal settings, our model was able to generate the basic shapes of both the energy and spatial distributions. However, the model often produced an excessive number of hits (`nhits`) at higher energy levels, leading to overestimation. This issue was not observed when training on single-bucket 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.

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:

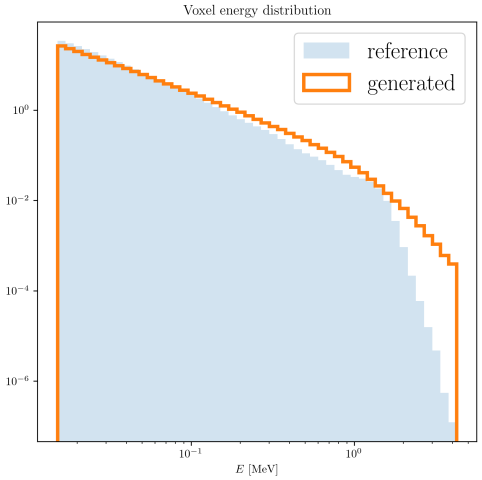
They are basically the same, indicating that the conditional variables are not working. Next, we also tried to concatenate the incident energy with the input data instead of adding them and the result is shown Figure 6.17.

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

doi:10.6342/NTU202500790

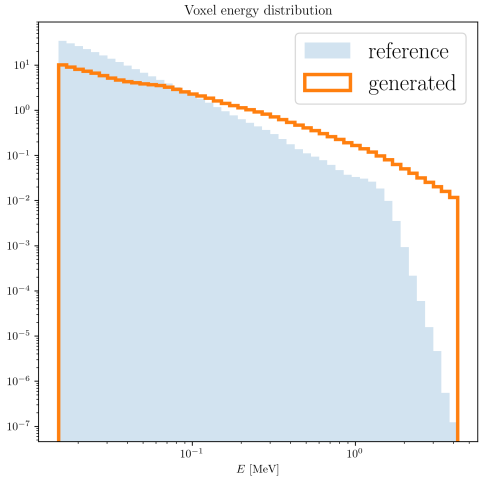


(A) The energy deposit of single bucket data.

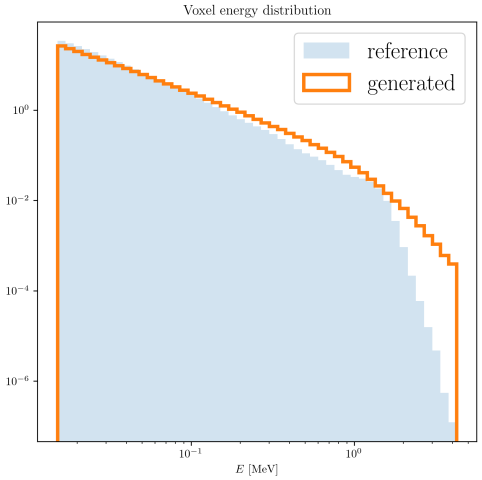


(B) The energy deposit of five bucket data.

FIGURE 6.15: The result of energy deposit of single bucket data and all bucket data.



(A) The energy deposit without incident energy.



(B) The energy deposit with incident energy.

FIGURE 6.16: The result of energy deposit with and without incident energy.

doi:10.6342/NTU202500790

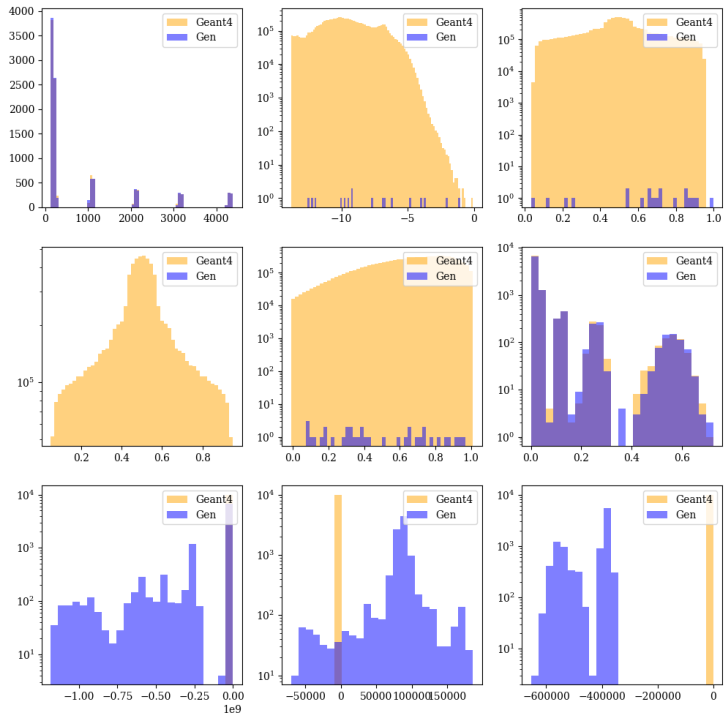


FIGURE 6.17: The result of energy deposit with incident energy concatenated with the input data.

6.7.2 Time

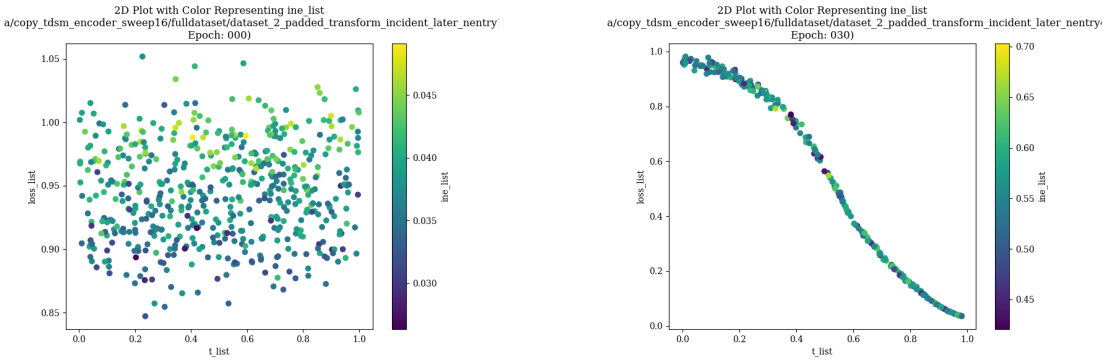
We also attempted to incorporate time as a conditional variable in our model. However, the results differ from those observed with incident energy. Even without explicitly using time as a condition, the output layer implicitly incorporates its effect by dividing by the standard deviation of the stochastic 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

 doi:10.6342/NTU202500790

inherently prioritize earlier time steps, making additional time conditioning redundant or ineffective.



(A) (B)

FIGURE 6.18: 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 6.18, 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.

doi:10.6342/NTU202500790

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.

 doi:10.6342/NTU202500790

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.

doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

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 100x 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 [59], 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.

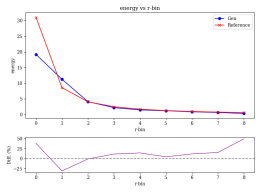
doi:10.6342/NTU202500790

doi:10.6342/NTU202500790

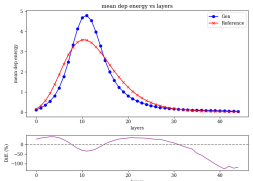
Appendix A

Figures

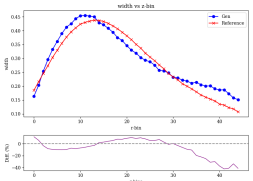
A.1 Best Result for Full Dataset



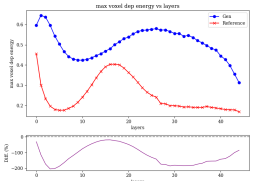
(A) Energy vs Radius



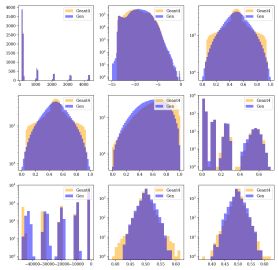
(B) Energy vs Z



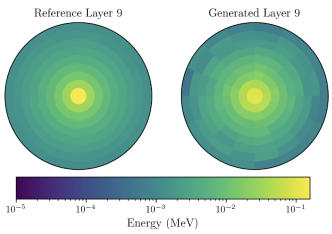
(C) R-width vs Layers



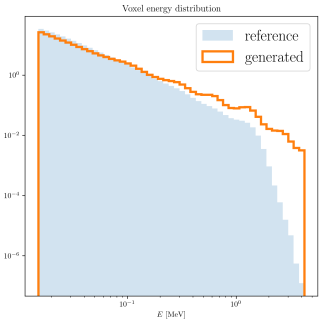
(D) Max Voxel Deposit vs Layers



(E) 1D Histogram



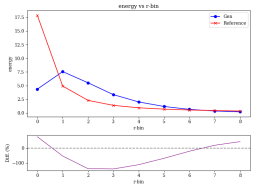
(F) Energy voxel comparison



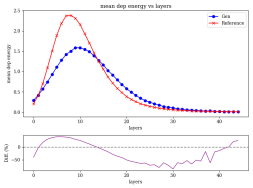
(G) Energy Deposit

doi:10.6342/NTU202500790

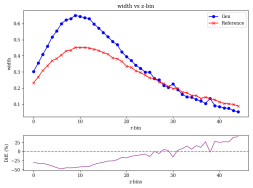
A.2 Best Result for Single Bucket Data



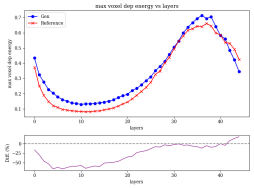
(A) Energy vs Radius



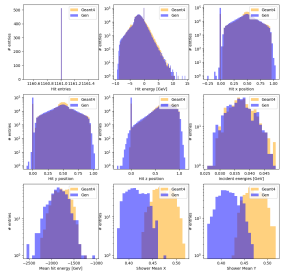
(B) Energy vs Z



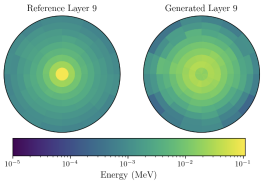
(C) R-width vs layers



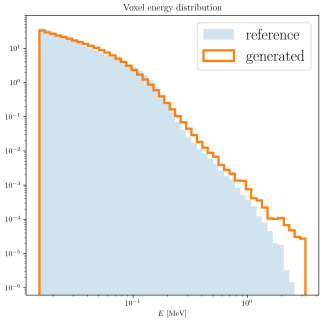
(D) Max Voxel Deposit vs Layers



(E) 1D Histogram



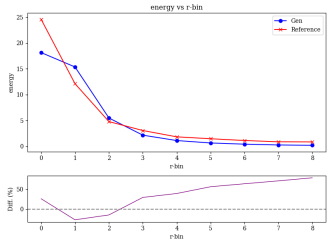
(F) Energy voxel comparison



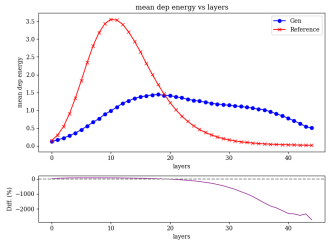
(G) Energy Deposit

doi:10.6342/NTU202500790

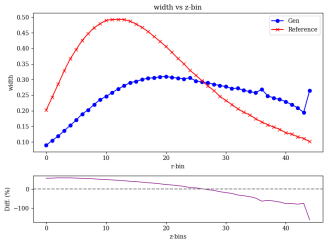
A.3 Result for using different Preprocessor



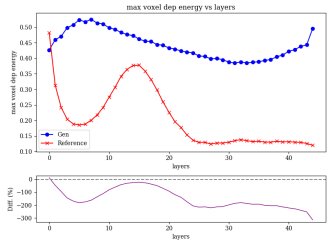
(A) Energy vs Radius



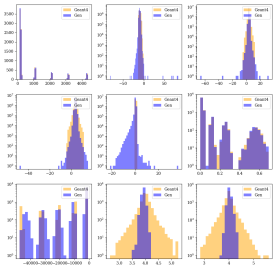
(B) Energy vs Z



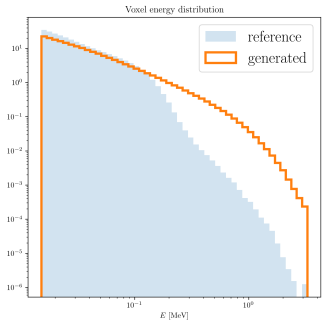
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers



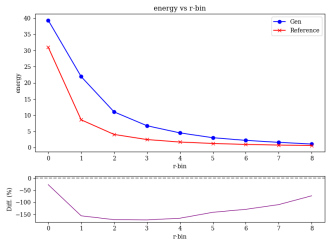
(E) 1D Histogram



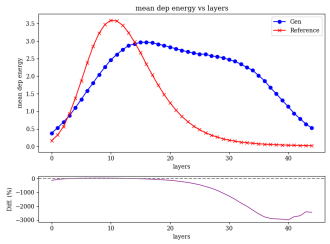
(F) Energy Deposit

FIGURE A.3: Result for using robust preprocessor

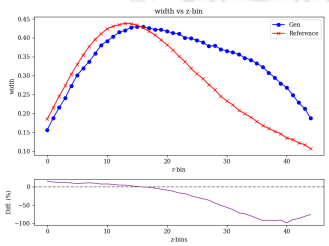
doi:10.6342/NTU202500790



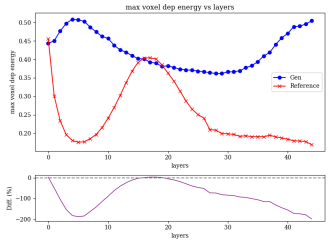
(A) Energy vs Radius



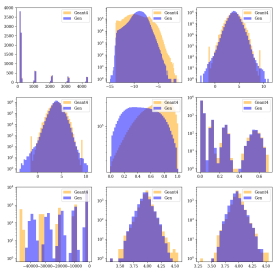
(B) Energy vs Z



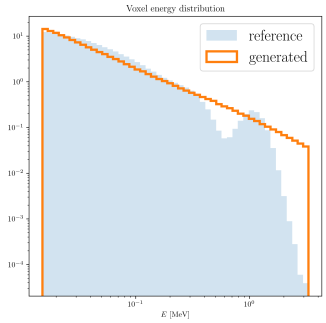
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers

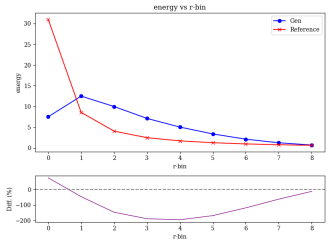


(E) 1D Histogram

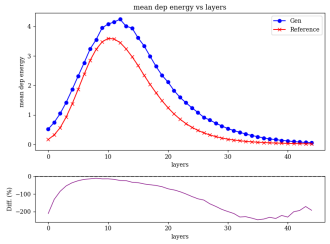


(F) Energy Deposit

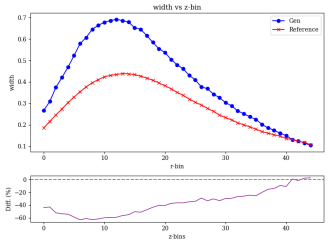
FIGURE A.4: Result for using quantile preprocessor



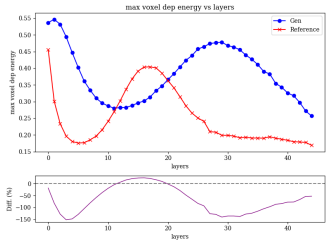
(A) Energy vs Radius



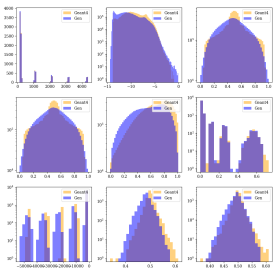
(B) Energy vs Z



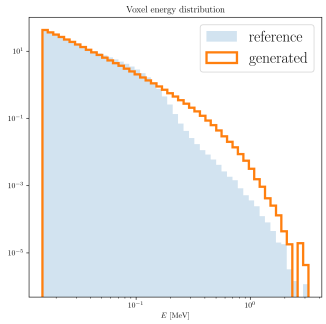
(C) R-width vs layers



(D) Max Voxel Deposit vs Layers



(E) 1D Histogram

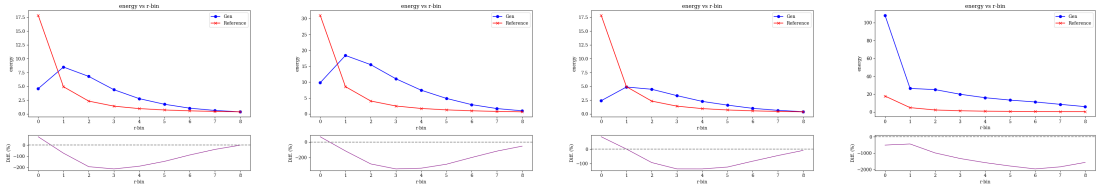


(F) Energy Deposit

FIGURE A.5: Result for using exponential preprocessor

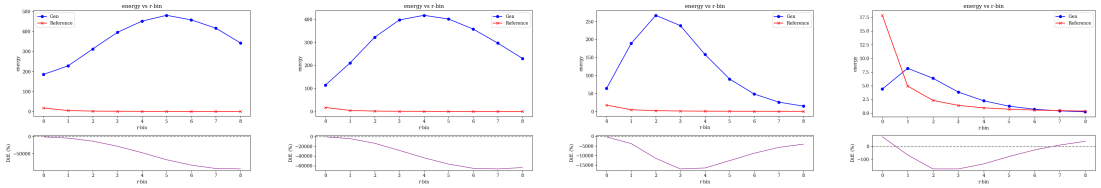
doi:10.6342/NTU202500790

A.4 Result for using different SDE settings



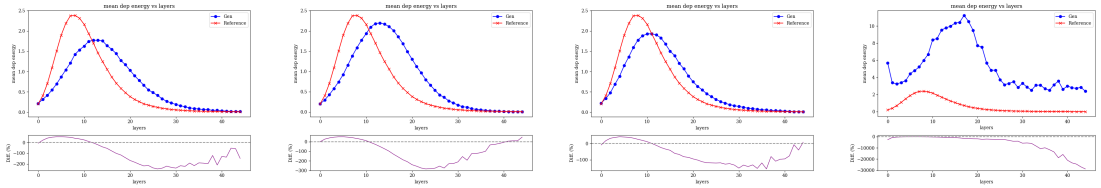
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.6: Result for Energy vs Radius for VE



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

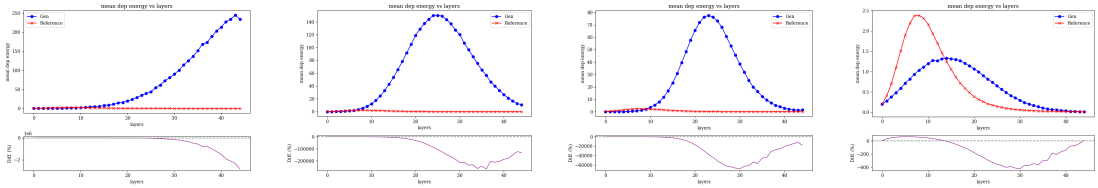
FIGURE A.7: Result for Energy vs Radius for VP



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

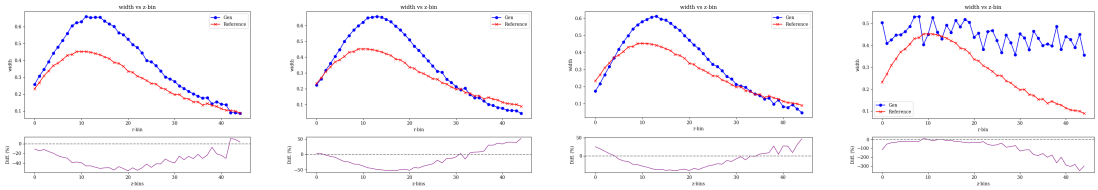
FIGURE A.8: Result for Energy vs Layers for VE

doi:10.6342/NTU202500790



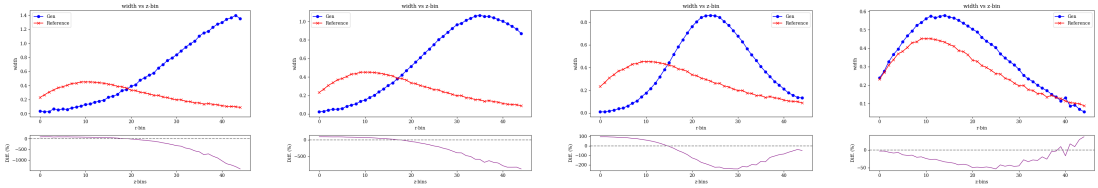
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.9: Result for Energy vs Layers for VP



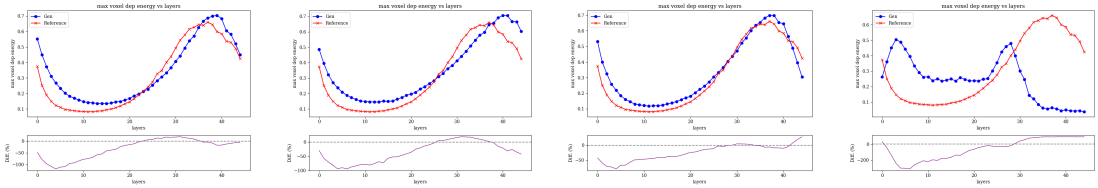
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.10: Result for R-width vs Layers for VE



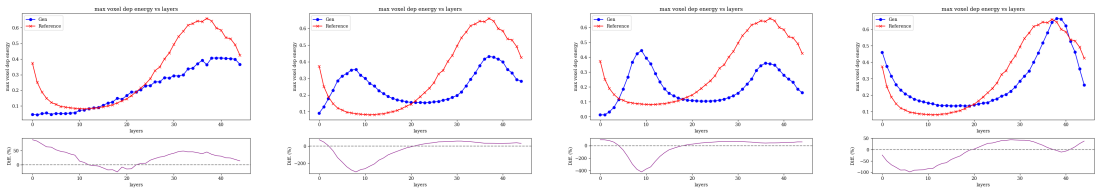
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.11: Result for R-width vs Layers for VP



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

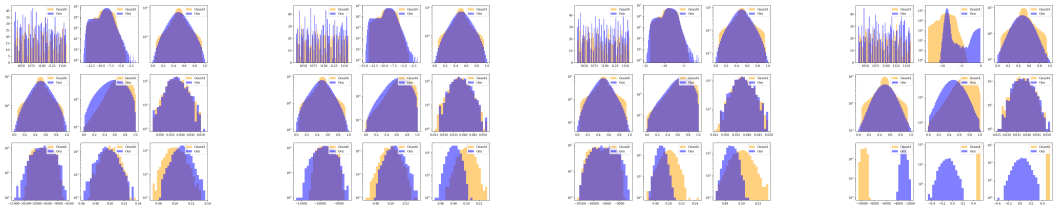
FIGURE A.12: Result for Max Voxel Deposit vs Layer for VE



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

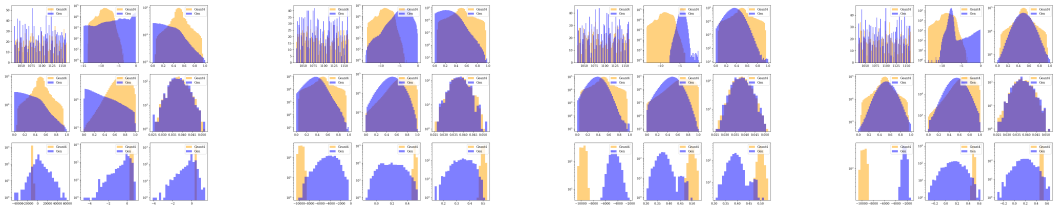
FIGURE A.13: Result for Max Voxel Deposit vs Layer for VP

doi:10.6342/NTU202500790



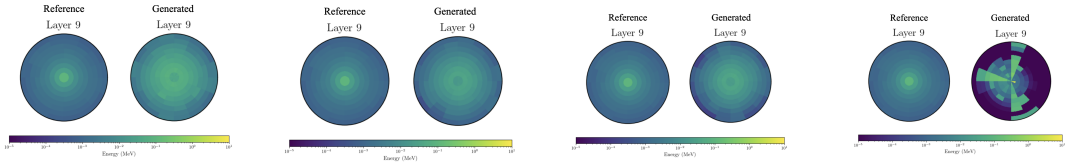
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.14: Result for Each Dimension VE



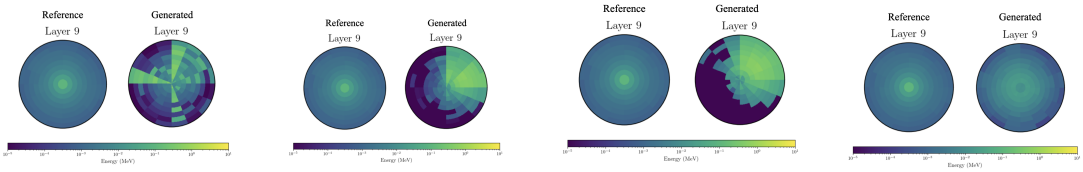
(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.15: Result for Each Dimension VP



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

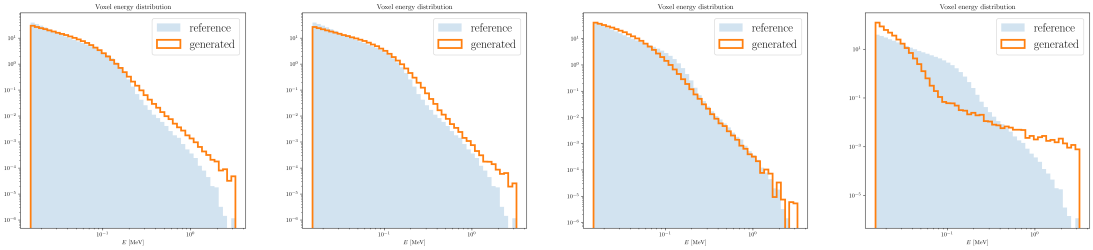
FIGURE A.16: Result for Energy Voxel Comparison for VE



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

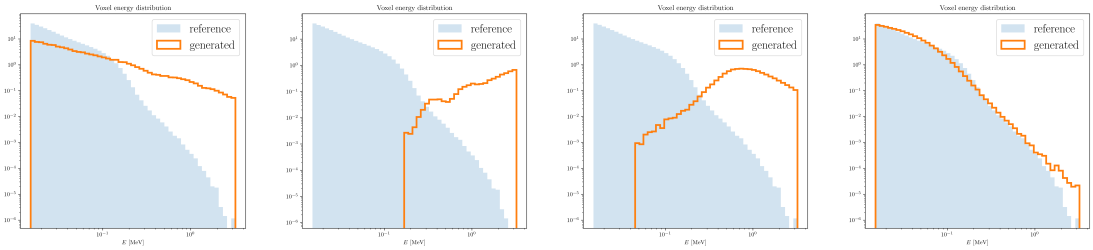
FIGURE A.17: Result for Energy Voxel Comparison for VP

doi:10.6342/NTU202500790



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.18: Result for Energy Deposit for VE



(A) $\sigma_{max} = 20$ (B) $\sigma_{max} = 10$ (C) $\sigma_{max} = 5$ (D) $\sigma_{max} = 1$

FIGURE A.19: Result for Energy Deposit for VP

doi:10.6342/NTU202500790

Appendix B

TopFCNH

B.1 Introduction

Besides my work on the Fast Calorimeter Simulation Challenge, I have also been involved in the TopFCNH project. For the sake of better understanding the concept and workflow of an analysis. This project aims to study the the interaction between top quark, higgs boson, and a light quark (u or c) in the context of the Standard Model Effective Field Theory (SMEFT) and search for new physics phenomena. It's just at the beginning stage, so what I have done includes roundtable presentation, gridpack preparation, monte carlo and data comparison. In this appendix, I will provide an overview of the TopFCNC project, the analysis workflow, gridpack generation, and the current status of the project.

B.2 Background

While higgs boson has been discovered in 2012, which is the newest particle, and the LHC is mainly designed for observing it, the top quark is the heaviest known elementary particle in the Standard Model (SM). The interaction between top quark and higgs boson is of great interest, as it can provide many insights into many unknown field.

The top-quark flavor-changing neutral current (TopFCNC) decay $t \to Hq$ (where $q = u, c$) is highly suppressed in the Standard Model (SM) due to the Glashow-Iliopoulos-Maiani (GIM) mechanism. [60] The predicted SM branching ratio for this process is $BR(t \to Hq) \sim 10^{-15} - 10^{-13}$, making it practically unobservable at the LHC. However, many beyond-the-SM (BSM) theories predict significantly enhanced branching ratios, making it a promising channel for new physics searches. As you can see the figure B.1

Several BSM frameworks predict an increase in the branching ratio. The Two-Higgs

doi:10.6342/NTU202500790

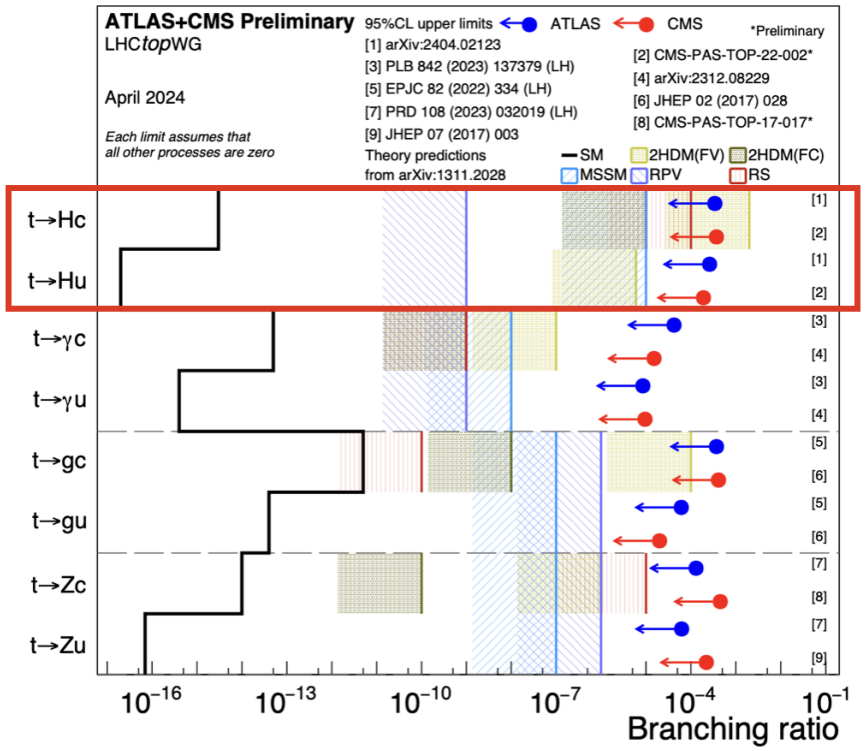


FIGURE B.1: The prediction and the result so far

Doublet Model (2HDM) suggests that $BR(t \to Hq)$ could reach $10^{-5} - 10^{-3}$.[61] Similarly, Supersymmetric Models (SUSY) predict comparable enhancements. Additionally, theories involving a Composite Higgs and Extra Dimensions indicate the possibility of increasing the branching ratio to $10^{-4} - 10^{-3}$. Given these enhancements, detecting $t \to Hq$ at the LHC would be a clear sign of new physics.

Among the Higgs boson decay channels, the $H \to \gamma\gamma$ (diphoton decay) is particularly attractive due to its clean experimental signature in the CMS electromagnetic calorimeter. The branching ratio of $H \to \gamma\gamma$ for a 125 GeV Higgs is approximately 0.2%, which is small but provides a well-reconstructed final state. Our research focuses on the process $pp \to t\bar{t}$, $pp \to tH$ and $pp \to tW-$, with $H \to \gamma\gamma$, where the diphoton final state can be efficiently detected using high-resolution electromagnetic calorimetry. The demonstrated Feymann diagram is shown in Figure B.2.

Photon triggers in CMS have high efficiency, with single-photon triggers reaching efficiencies above 99% and double-photon triggers capturing over 88% of events. Major backgrounds include prompt diphoton production ($pp \to \gamma\gamma +$ jets) and fake $\gamma + j$, which can be reduced by later analysis methods.

doi:10.6342/NTU202500790

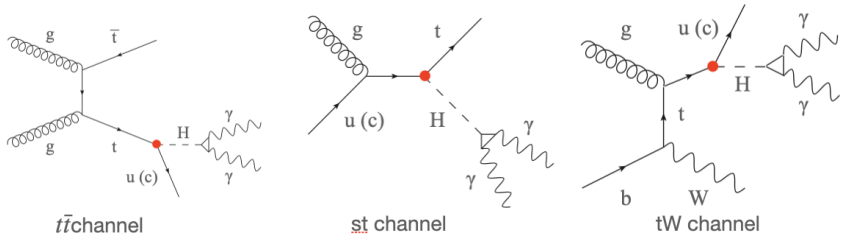


FIGURE B.2: The Feynman diagrams for the TopFCNC channels

B.3 Analysis Tool

Before we dive into the details of the analysis, let's first introduce the tool, HiggsDNA, we use for the analysis. HiggsDNA stands for Higgs diphoton NANOAOD, which is a tool for analyzing the Higgs boson decay to diphoton in the NanoAOD format. It is a pure Python package, which means the user can do things without CMS environment, that provides a set of functions to analyze the Higgs boson decay to diphoton.

Besides the environment independence, HiggsDNA also has some other changes and advantages compared to the traditional analysis tool.

First,traditional high-energy physics analyses often employ a per-event processing method, iterating through each event and its components sequentially. While straightforward, this approach can be computationally intensive and time-consuming. HiggsDNA adopts a columnar analysis paradigm, utilizing libraries such as awkward-array and coffea. This method processes data in a vectorized manner, enabling simultaneous operations on entire datasets. Such an approach not only accelerates computations but also enhances code clarity and maintainability.

Second, it provides robust tools to define and propagate these uncertainties throughout the analysis workflow, ensuring that results reflect both statistical and systematic variations.

Third, it also provides the studied corrections for each year, which can be used to correct the data and simulation.

In practice, we incorporate all these pre-studied corrections and uncertainties into a file called *run_analysis.py*, which calls a processor to handle the data, apply the necessary corrections, and propagate uncertainties to both the data and the simulation. In this process, our main tasks are to write the processor and the analysis code. Once these are prepared, we simply run *run_analysis.py* to obtain the results.

The overall workflow of the analysis is shown in Figure B.3.

doi:10.6342/NTU202500790

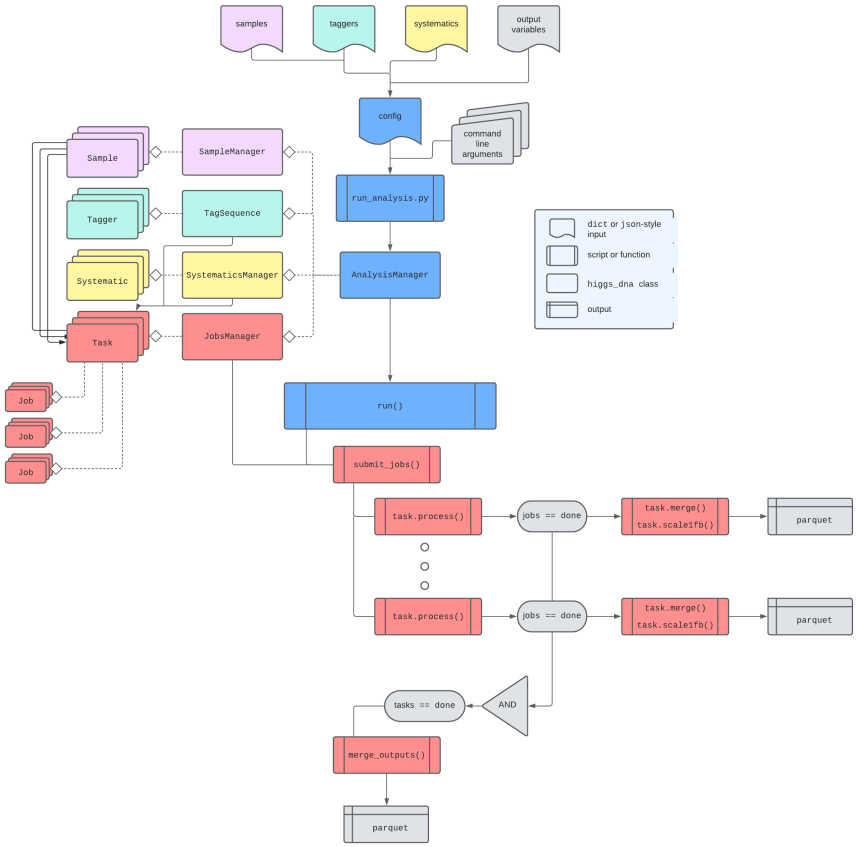


FIGURE B.3: The workflow of HiggsDNA.
[62]

doi:10.6342/NTU202500790

B.4 Workflow

Although the TopFCNC analysis is still in its early stages, we have established a preliminary workflow to guide our research. The workflow consists of several key steps, each contributing to the overall analysis process. The overall workflow consists of three main stages: Data-MC Samples Comparison & Top Reconstruction, Signal-Background Separation & Signal Region Optimization, and Statistical Analysis. This section outlines the key steps in each stage and their significance in the overall analysis.

B.4.1 Data-MC Samples Comparison & Top Reconstruction

The first step in the analysis workflow involves comparing data and Monte Carlo (MC) samples to validate the simulation's accuracy in modeling real experimental conditions. This stage focuses on reconstructing the top quark and verifying its properties against theoretical predictions. The main aspects of this step include:

- Utilizing Run 3 data collected between 2022 and 2024.
- Studying a total of 12 analysis channels, derived from three different production mechanisms and two possible decay modes:
  - Flavored Higgs couplings: Hut and Hct.
  - W boson decays into either leptonic or hadronic final states.
  - Single-top production (st), top-pair production ($t\bar{t}$), and associated top-W production (tW).
- Performing event reconstruction using:
  - A χ^2 method for selecting the most probable event topology.
  - Artificial Neural Network (ANN) training to improve event classification.

This stage ensures that the data used in the analysis is well understood and that the top quark events are reconstructed with high precision.

B.4.2 Signal-Background Separation & Signal Region Optimization

Once events are reconstructed, the next stage involves distinguishing signal events from background contributions. This process is critical for maximizing the sensitivity of the analysis. The main components of this stage are:

- **Signal-Background Separation:**

doi:10.6342/NTU202500790

  - Multi-Variate Analysis (MVA) techniques are employed, incorporating kinematic features and top reconstruction information.
  - Dedicated MVA classifiers are trained to differentiate between the Higgs signal and backgrounds, including Non-Resonant Background (NRB) and Standard Model Higgs Background (SMH).

- **Signal Region Optimization:**
  - Signal regions are defined using a two-dimensional phase space, where classification scores from NRB-MVA and SMH-MVA are used to optimize the separation of signal and background events.

These steps ensure that the analysis isolates the signal efficiently while minimizing background contamination, thereby improving the precision of the final measurement.

B.4.3 Statistical Analysis

The final stage of the workflow involves statistical modeling and interpretation of the extracted signal. This stage includes:

- **Modeling:**
  - The invariant mass of the diphoton system ($m_{\gamma\gamma}$) is used as the key observable.
  - Background modeling is performed separately for NRB and SMH components.
  - The sideband regions are defined within the ranges $[100, 115] \cup [135, 180]$ GeV.
  - The signal window is restricted to $[115, 135]$ GeV.
- **Results:**
  - A simultaneous signal-plus-background (S+B) fit is performed to extract the Higgs boson signal strength.
  - The final step involves setting an upper limit on the branching ratio (BR) of the targeted decay mode.

This stage quantifies the statistical significance of the observed signal and provides constraints on Higgs boson properties based on the analyzed dataset.

 doi:10.6342/NTU202500790

B.4.4 Summary of the Workflow

The entire analysis workflow is summarized in Figure B.4. It illustrates the three major stages, from data preparation to final statistical inference. Each step is designed to systematically refine the dataset, enhance the signal-to-background ratio, and extract meaningful physics results from the experimental data.

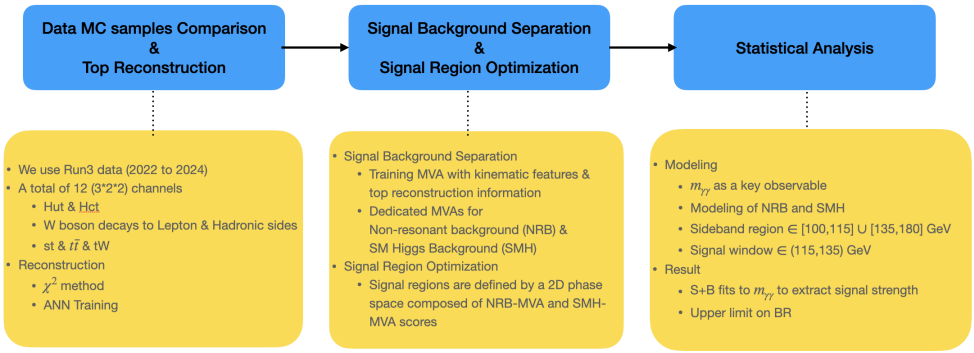


FIGURE B.4: The workflow of the HiggsDNA analysis framework.

This structured approach ensures a robust and efficient methodology for Higgs boson studies, leveraging advanced data analysis techniques and statistical tools.

B.5 Gridpack Generation

Originally, this shouldn't be a big problem. However, we faced some difficulties which will be explained later during doing the NLO calculation. In order to generate the signal samples for the TopFCNH ananlysis, we need to create gridpacks which contains all the parameters and configurations for the simulation. The gridpack generation process involves several key steps:

- **MadGraph5 Generation** The first step is to do the first decay, there are three channels in our research. The first channel is $pp \to t\bar{t}$, the second channel is $pp \to tH$, and the third channel is $pp \to tW-$. It is here that we do the NLO calculation. And the t and W will be decayed in madspin again.
- **Madspin Decay:** In this step, we will futher decay $t \to qH$ or $t \to bW+$ and $W \to jj$ or $W \to l\nu$, where $q = u, c$ and $l = e, \mu$.

However, due to the forbidden process $t \to qH$ in SM, we need to use special model with special parameters designed for TopFCNH. This is the main difficulty we faced during the gridpack generation. The original model in CMS failed to do the NLO calculation due to the improper parameter settings and python version inconsistency. We need to modify the model and the parameters to make it work. At the end, we also

doi:10.6342/NTU202500790

discussed with ATLAS modeling group to get the workable model and parameters. Thus, I also made a presentaion in formal meeting to introduce the work we have done and the problems we faced.

B.6 Current Status

The TopFCNH analysis is still in its early stages, with the gridpack generation being the primary focus. Besides the gridpack generation, we have also started to do the top reconstruction and compare the data and MC samples. For top reconstruction, because we don't have the signal samples yet, we used the ttbar samples to do the reconstruction also for the sake of practicing the HiggsDNA tool. The results are shown in Figure B.5.

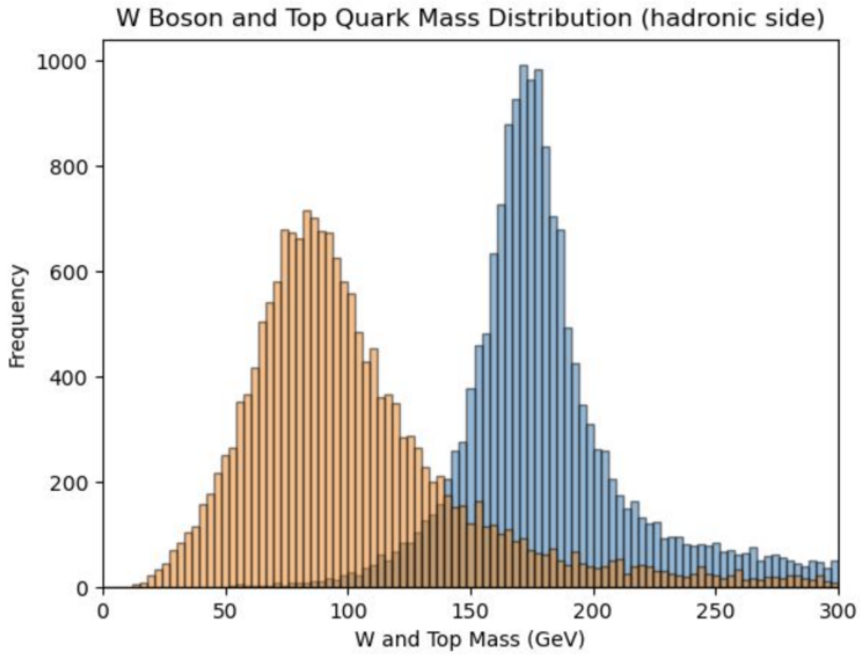


FIGURE B.5: Top quark reconstruction using Higgs DNA package (with ttH samples as practice)

doi:10.6342/NTU202500790

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

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

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

[11] 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.

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

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

doi:10.6342/NTU202500790

[14] 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).

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

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

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

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

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

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

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

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

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

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

[25] 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.

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

[27] *Silicon Pixels | CMS Experiment*.

[28] *Silicon Strips | CMS Experiment*.

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

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

[31] 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.

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

doi:10.6342/NTU202500790

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

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

[35] Candan Isik. "Phase 1 upgrade of the CMS Hadron Barrel Calorimeter". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1042 (2022), p. 167389. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2022.167389.

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

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

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

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

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

[41] 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.

[42] Hubert Gerwig. "Engineering challenges in mechanics and electronics in the world's first particle-flow calorimeter at a hadron collider: The CMS high-granularity calorimeter". In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1044 (2022), p. 167493. ISSN: 0168-9002. DOI: https://doi.org/10.1016/j.nima.2022.167493.

[43] 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.

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

[45] *celeritas-project/hgcal: A Geant4 simulation of the 2018 CMS HGCAL test-beam for geant-val.*

[46] S. Agostinelli et al. "Geant4—a simulation toolkit". In: *Nuclear Instruments and Methods in Physics Research A* 506.3 (2003), pp. 250–303. DOI: 10.1016/S0168-9002(03)01368-8.

[47] Geant4 Collaboration. *Geant4 User Documentation*. Available at: https://geant4-userdoc.web.cern.ch. 2024.

 doi:10.6342/NTU202500790

[48] J. Allison et al. "Recent developments in Geant4". In: *Nuclear Instruments and Methods in Physics Research A* 835 (2016), pp. 186–225. DOI: 10.1016/j.nima.2016.06.125.

[49] *CaloChallenge/homepage*. original-date: 2022-01-27T14:46:13Z. Dec. 2024.

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

[51] 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.

[52] Jascha SohlDickstein et al. "Deep Unsupervised Learning using Nonequilibrium Thermodynamics". In: *CoRR* abs/1503.03585 (2015).

[53] *Introduction to Generative Diffusion Models (Part 1): DDPM = Demolition + Construction - Scientific Spaces*.

[54] Pascal Vincent. "A connection between score matching and denoising autoencoders". In: *Neural Computation* 23.7 (2011), pp. 1661–1674. DOI: 10.1162/NECO_a_00142.

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

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

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

[58] *PyTorch*. en.

[59] Yilun Xu et al. *Restart Sampling for Improving Generative Processes*. 2023.

[60] Luciano Maiani. *The GIM Mechanism: origin, predictions and recent uses*. arXiv:1303.6154 [hep-ph]. Mar. 2013. DOI: 10.48550/arXiv.1303.6154.

[61] *FCNCHistory < LHCPhysics < TWiki*.

[62] *Table of Contents*. en-US.

 doi:10.6342/NTU202500790