
Deep Generative Models for Ultra-High Granularity Particle Physics Detector Simulation: A Voyage From Emulation to Extrapolation

Baran Hashemi



München 2023

Deep Generative Models for Ultra-High Granularity Particle Physics Detector Simulation: A Voyage From Emulation to Extrapolation

**Tiefe Generative Modelle für die Simulation von
Teilchenphysik-Detektoren mit ultrahoher Granularität:
Eine Reise von der Emulation zur Extrapolation**

Baran Hashemi

Dissertation
der Fakultät für Physik
der Ludwig-Maximilians-Universität
München

vorgelegt von
Baran Hashemi
aus Rasht, Iran

München, den 12. October 2023

Erstgutachter: Prof. Dr. Thomas Kuhr

Zweitgutachter: Prof. Dr. Lukas Heinrich

Tag der mündlichen Prüfung: 22.11.2023

Zusammenfassung

Die Simulation von Detektorantworten mit ultrahoher Granularität in der Teilchenphysik stellt eine kritische, jedoch rechenintensive Aufgabe dar. Diese Arbeit zielt darauf ab, diese Herausforderungen zu bewältigen, indem sie sich auf den Pixel-Vertex-Detektor (PXD) im Belle II-Experiment konzentriert, der über 7,5 Millionen Pixelkanäle verfügt – den höchstaufgelösten Detektorsimulationsdatensatz, der jemals mit tiefen generativen Modellen analysiert wurde. Als erste Beitrag führe ich das Intra-Event-Aware Generative Adversarial Network (IEA-GAN) ein, ein Modell, das relationales Denken und selbstüberwachtes Lernen integriert, um ein „Ereignis“ im Detektor zu simulieren. Diese Studie etabliert PXD-Daten als feinkörnigen Datensatz und unterstreicht die Bedeutung der intra-eventuellen Korrelation für nachgelagerte physikalische Analysen. IEA-GAN emuliert PXD-Signaturen auf eine geometriebewusste und korrelierte Weise.

Aufbauend darauf führt diese Arbeit YonedaVAE ein, ein fortschrittliches generatives Modell, das das offene Problem der Out-of-Distribution (OOD) Simulation von realen Daten angeht. Inspiriert durch die Kategorientheorie verwendet YonedaVAE ein lernbares Yoneda-Embedding, um die Gesamtheit eines Ereignisses anhand seiner Sensorbeziehungen zu erfassen, und formuliert eine formale Sprache für intra-eventuelles relationales Denken. Dies wird ergänzt durch einen selbstdestillierten Mengengenerator und einen adaptiven Top-q Abtastmechanismus, der es dem Modell ermöglicht, Punktwolken mit variablen intra- und inter-eventuellen Kardinalitäten weit über die Kardinalität der Trainingsdaten hinaus zu sammeln. Eine variable intra-event Kardinalität wurde zuvor noch nicht erreicht und ist von entscheidender Bedeutung, wenn man sich mit unregelmäßigen Detektorgeometrien und Treffermustern durch den Detektor beschäftigt.

Diese Studie präsentiert die ersten Ergebnisse für ein generatives Modell, das auf realen Daten in der Teilchenphysik mit ultrahoher Granularität trainiert wurde. Sie zeigt, dass das YonedaVAE-Modell, trainiert auf frühzeitigen Hintergrunddaten eines Experiments, eine vernünftige Simulation eines späteren Experiments mit fast doppelter Leuchtkraft erreichen kann, während es gleichzeitig eine signifikante Speicherentlastung bietet. Bemerkenswert ist, dass YonedaVAE diese Extrapolation ohne vorherige Exposition gegenüber Hochleuchtkraftdaten erreicht, was seine Robustheit und Generalisierbarkeit unterstreicht. Insgesamt reduzieren diese Modelle nicht nur erheblich den Rechenaufwand, sondern erreichen auch eine noch nie dagewesene Präzision in Detektorsimulationen mit ultrahoher Granularität. Dies eröffnet neue Wege sowohl für die Recheneffizienz als auch für die Genauigkeit in der Teilchenphysik.

Abstract

Simulating ultra-high-granularity detector responses in Particle Physics represents a critical yet computationally demanding task. This thesis aims to overcome these challenges by focusing on the Pixel Vertex Detector (PXD) at the Belle II experiment, which features over 7.5M pixel channels—the highest spatial resolution detector simulation dataset ever analyzed with deep generative models. As the first contribution, I introduce the Intra-Event Aware Generative Adversarial Network (IEA-GAN), a model incorporating relational reasoning and self-supervised learning to simulate an “event” in the detector. This study establishes PXD data as a fine-grained dataset and underscores the importance of intra-event correlation for downstream physics analyses. IEA-GAN emulates PXD signatures in a geometry-aware and correlated manner.

Building upon this, this thesis introduces YonedaVAE, a Category Theory-inspired generative model that tackles the open problem of Out-of-Distribution (OOD) simulation. Inspired by Category Theory, YonedaVAE introduces a learnable Yoneda Embedding to capture the entirety of an event based on its sensor relationships, formulating a formal language for intra-event relational reasoning. This is complemented by a self-supervised set generator and an adaptive Top-q sampling mechanism, enabling the model to sample point clouds with variable intra-event and inter-event cardinalities far beyond the training data cardinality. Intra-event variable cardinality has not been done before and is vital when one is dealing with irregular detector geometries and hit patterns through the detector.

This study presents the first results for a generative model trained on real data in ultra-high granularity particle physics. It shows that the YonedaVAE model, trained on an early experiment background data, can reach a reasonable simulation of a later experiment with almost double luminosity while providing a significant storage release. Remarkably, YonedaVAE achieves this extrapolation without previous exposure to high-luminosity data, showcasing its robustness and generalizability.

Collectively, these models not only substantially reduce computational overhead but also achieve unprecedented precision in ultra-high-granularity detector simulations, opening new avenues for both computational efficiency and accuracy in particle physics.

Contents

Zusammenfassung	iv
Abstract	vi
Contents	x
License	xi
1 Introduction	1
2 Belle II experiment: The PXD Saga	5
2.1 Belle II Experiment	5
2.1.1 Belle II Software Framework	8
2.2 Pixel Vertex Detector (PXD)	12
2.2.1 PXD Background: Types and Levels	14
2.3 PXD Background Simulation: Ideas and Challenges	17
2.3.1 Background Simulation	17
2.3.2 Random Trigger	18
2.4 PXD Background Simulation: Surrogate models	18
2.5 PXD Background Simulation: Figure of Merits	20
2.5.1 Low-Level Metrics	20
2.5.2 Neural Network-based Metrics	21
2.5.3 Clustering and Topological Data Analysis	21
2.5.4 Physics-Level Metrics	22
3 On the Shoulder of Giants: The Modern Machine Learning Tools	25
3.1 Introduction	25
3.2 Attention Mechanism and Transformers	26
3.2.1 Attention Mechanism and Self-Attention	26
3.2.2 Transformer	28
3.2.3 Problems with Transformer Models	31
3.3 Deep Generative Models and Simulation-Based Inference	32
3.3.1 Variational Autoencoders	33
3.3.1.1 Likelihood Estimation:	35

3.3.1.2	Problems with VAEs	37
3.3.2	Generative Adversarial Networks	37
3.3.2.1	Problems with GANs	38
3.3.3	BigGAN and ContraGAN	40
3.3.3.1	ContraGAN	44
3.3.4	Point Cloud Generation	45
3.3.4.1	Permutation Invariant loss	47
3.4	Self-Supervised Representation Learning	48
3.4.1	The Deep Metric Learning (Contrastive) Family	50
3.4.2	The Self-Distillation Family	52
4	Deep Generative Models for Detector Simulation: A Taxonomy	57
4.1	Introduction	57
4.2	Problem Definition	57
4.3	Algorithm Taxonomy	58
4.3.1	Latent Variable Approaches	58
4.3.1.1	Sampling Strategies	59
4.3.1.2	Generation Strategies	60
4.3.1.3	Representative Work	61
4.3.2	Non-Latent Variable Approaches	64
4.3.2.1	Representative Work	64
4.4	Applications	65
4.4.1	Statistics Amplification	66
4.4.2	Amortised Generation	67
4.4.3	Out-Of-Distribution (OOD) Generation	68
4.5	Conclusion	69
5	PXD Background Generation: Simulation Data	71
5.1	Introduction	71
5.2	PXD Images: Fine-Grained Image Analysis	71
5.2.1	PXD Generation Formulation	74
5.2.2	Challenges Of DGMs for PXD simulation	75
5.3	Prior Embedding GAN (PE-GAN)	76
5.3.1	Transition to BigGAN	77
5.3.2	Adopting Self-Attention GAN	78
5.3.2.1	Model Refinements	78
5.4	Intra-Event Aware GAN (IEA-GAN)	80
5.4.1	Introduction and Overview	80
5.4.2	Relational Reasoning	81
5.4.2.1	Event Approximation	83
5.4.2.2	Relational Reasoning Module	83
5.4.3	Intra-Event Aware Loss	85
5.4.4	Uniformity Loss	86

5.4.5	Model Details and Hyperparameters	88
5.5	Evaluation Results: IEA-GAN vs SOTAs	89
5.5.1	Neural Network-based Metrics: FID and KID	90
5.5.2	Marginal Distributions	91
5.5.3	Physics-Level Metric: Tracking	93
5.5.3.1	Physics Motivation for Intra-Event sensor correlation	97
5.5.3.2	IEA-GAN’s Tracking Performance	98
5.6	Ablation Studies and Things I Tried But Did Not Work	101
5.7	Conclusion and Discussion	103
6	PXD Background Generation: Real Data	107
6.1	Introduction	107
6.2	Real PXD data and OOD Simulation: Challenges and Motivations	108
6.2.1	Set Generation:	110
6.2.2	Length Extrapolation	112
6.2.3	Context Extrapolation:	116
6.3	Prologue: Category Theory and Yoneda Perspective	117
6.3.1	What is Category Theory?	117
6.3.2	Basic Definitions	118
6.3.3	Functors and Natural Transformations	118
6.3.4	The Yoneda Perspective	119
6.3.4.1	Yoneda Embedding and Yoneda Lemma	120
6.3.4.2	Presheaf	121
6.4	YonedaVAE	122
6.4.1	Problem setup	122
6.4.2	YonedaVAE: Yoneda Encoder	123
6.4.2.1	Eventformer and Attention Gating	124
6.4.2.2	Yoneda-Pooling	125
6.4.2.3	Yoneda-Pooling and Category Theory	127
6.4.3	YonedaVAE: Set Generation	129
6.4.3.1	Self-Distillation	130
6.4.3.2	Adaptive Top-q sampling	132
6.4.4	YonedaVAE: Decoder	134
6.5	Evaluation Results	137
6.5.1	Marginal Distributions	138
6.5.2	FID, KID, and Vendi Score	141
6.5.3	Topological and Geometrical Clustering for PXD data	150
6.5.3.1	Topological Data Analysis	150
6.5.3.2	Persistent Homology and filtration	151
6.5.3.3	Topological Data Analysis of PXD Background	157
6.5.3.4	Geometrical Clustering	159
6.5.4	Physics Analysis	163
6.6	Ablation Studies	172

6.7 Conclusion	175
7 Summary and Outlook	177
7.1 Summery	177
7.2 Future Directions and Opportunities	181
List of Publications	185
List of Figures	191
List of Tables	194
Bibliography	231
Acknowledgements	232

License

This work is licensed under

CC BY 4.0

<https://creativecommons.org/licenses/by/4.0/>

Chapter 1

Introduction

The current consensus to model the origin of our universe centers on an event known as the Big Bang [1, 2], which occurred 13.8 billion years ago. This explosive instant was characterized by extreme heat and dense energy, giving rise to elementary particles and their corresponding anti-particles. Mere fractions of a second later, cosmic inflation dramatically expanded the universe. What transpired next remains a fundamental enigma, serving as one of the motivating forces behind the creation of the Belle II experiment [3]. This crucial and enigmatic process, termed “Baryogenesis,” violated baryon number conservation, resulting in the observed predominance of baryons over anti-baryons.

In mainstream cosmology, the Universe underwent an initial inflationary phase, swiftly transitioning to a period of radiation domination. This was followed by an era of matter domination, ultimately giving way to the dark energy-dominated epoch we find ourselves in today [4]. While we can somewhat simulate baryon formation in particle colliders, the puzzle of the universe’s “missing” antimatter remains unsolved. To generate matter and antimatter at differing rates, a baryon-generating interaction must meet three critical conditions [5]: violation of the baryon number, breaking of C-symmetry and CP-symmetry, and a deviation from thermal equilibrium.

The Belle II experiment focuses particularly on CP-symmetry violation, which results in an imbalance between the number of left-handed and right-handed baryons and anti-baryons. This adds another layer of complexity to the baryogenesis conundrum. Current explanations within the Standard Model fail to account for the observed matter-antimatter disparity, signaling the need to delve into unexplored areas of physics. To navigate this uncharted territory, researchers employ two primary approaches. The first, known as the “high energy frontier,” is utilized by the Large Hadron Collider (LHC) and aims to directly create and analyze new particles through high-energy collisions. SuperKEKB (Belle II) takes a different tack and operates on the “intensity frontier,” focusing on high-precision experiments to identify deviations from the Standard Model and search for new, weakly-coupled mediators in the dark sector. Thus, the quest endures, the mysteries dating back to the universe’s earliest moments to today’s cutting-edge experiments, each discovery bringing us closer to understanding the nature of our existence.

The key to high-precision measurements at SuperKEKB is the collider’s ability to produce “clean” events. This enables precise measurements, particularly for events where particles like neutrinos escape undetected. High-precision measurement at Belle II is achieved by recording a large number of collisions to offer a statistically robust sample for analyses. Additionally, the excellent reconstruction of particle trajectories, particularly their decay vertices, is crucial. SuperKEKB has attained an unprecedented instantaneous luminosity of $4.7 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$ and aims for an even higher rate of $8 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$. Luminosity is a measure of how many particles pass through a given area in a specific time period. It’s essentially a measure of the “brightness” of the collider, and higher luminosity implies that more particle collisions are likely to occur, increasing the odds of observing rare processes.

Discoveries of New Physics at Belle II are heavily dependent on the precise reconstruction of particle decay vertices. To achieve this level of precision, the Belle II detector is equipped with a state-of-the-art PiXel vertex Detector (PXD) located very close to the interaction point. This serves as the innermost sub-detector of Belle II. The PXD, with its “ultra-high granular” matrix of sensors, records the passage of charged particles with a total readout of 7.68×10^6 channels per event. However, its position as the innermost sub-detector results in a high level of “background” noise. “background” refers to any detector hit that is not of interest but cannot be fully eliminated from the data. These are usually events that look similar to the desired signal but are caused by different processes. These backgrounds, as artifacts in the PXD, directly influence the precision of particle trajectory finding and reconstruction.

To validate our understanding of theoretical models in physical processes, Belle II, like other particle physics experiments, heavily relies on simulation for various tasks. These include data selection, statistical inference, and design optimization for new experiments. Consequently, emulating the PXD background responses is critical for studying experimental conditions. However, this emulation is both storage-intensive and computationally demanding. The current landscape for PXD background simulation includes three primary approaches: First is the somewhat accurate but storage-consuming and computationally intense “Geant4 simulation.” Second is the highly accurate but storage-intensive “random trigger” method. Lastly, there’s the “surrogate model,” known for its fast and efficient simulation capabilities.

In this thesis, my primary objective is to develop specialized Deep Generative Models (DGM) as surrogates to enhance computational efficiency and data storage for Belle II PXD detector simulation. To address the challenges presented by the ultra-high resolution of PXD data, I introduce a fresh perspective through “event-based relational reasoning.” In an event-based relational reasoning approach, each sensor in an event is studied in relation to the other sensors. The central premise is that understanding each PXD sensor solely based on its intrinsic features is insufficient for accurate simulation. Instead, how each sensor interacts with other sensors within the event provides a contextualized representation, very much like transitioning from “syntax” to “semantics” in linguistics. Through this thesis, I demonstrate that incorporating this relational inductive bias leads to the creation

of more accurate surrogate models for detector simulation.

Within the ensuing chapter, Chapter 2, I introduce the Belle II detector and its software framework. Then, there will be an in-depth discussion on the PXD background and various ways of simulating its detector signatures.

Chapter 3 lays the foundation for deep generative models, relational reasoning, and self-supervised learning, the three Machine Learning (ML) pillars of this thesis, by providing an overview of the prerequisite technologies and methodologies.

In Chapter 4, I present an exhaustive and taxonomically organized review of the use of deep generative models in Experimental Particle Physics, with a special focus on detector simulation. By the end of this chapter, I construct a holistic view to demonstrate that current state-of-the-art deep generative models are incomplete for tackling ultra-high granular PXD background generation.

Moving on to Chapter 5, I delve into the specifics of PXD background generation and offer a task-categorization perspective for it. I also introduce two new models that I have developed: the Prior Embedding GAN [6], founded on Contrastive Learning, and the Intra-Event Aware GAN (IEA-GAN) [7], which relies on Self-Supervised Learning and relational reasoning. These models are trained and tested on Geant4-simulated PXD background data. During this process, novel technologies in deep generative models tailored for fine-grained images will be introduced. For evaluation, I initially motivate the significance of “intra-event correlation” for the first time in the fast simulation domain via a study focused on Helix Parameter Resolution. Subsequently, I conduct a comprehensive evaluation of various figures of merit. As a result of this analysis, IEA-GAN has successfully integrated into the basf2 software [8] suite as a surrogate module for emulating PXD background on the fly for analysis.

In Chapter 6, I engage with the “real” PXD detector background data coming from the random trigger. This chapter explores the challenges and motivations behind the Out-Of-Distribution (OOD) simulation of PXD background data and extrapolation beyond current experimental limits, particularly concerning luminosity. I then seek to unify relational reasoning concepts through the lens of Category Theory, the abstract study of mathematical objects and their interrelations. As a result, I introduce YonedaVAE, a zero-shot point cloud deep generative model capable of generating PXD background hits with an unprecedented cardinality of 100 700 hits per event, despite being trained only on 7600 PXD hits per event from Experiment 12. Remarkably, YonedaVAE performs robustly when tested on data from Experiment 26, which has nearly double the luminosity of Experiment 12. The chapter presents results in two primary tasks: “length extrapolation,” where the model has access to an individual sensor condition, and “context extrapolation,” where it has access only to a global event-level condition during inference and has to solve an Inverse Problem. For evaluation, after an in-depth evaluation of low-level marginal distributions and NN-based metrics, I introduce a new diversity measure for detector simulation, the Vendi Score, previously used in Ecology and Protein Design. Then, I examine the geometrical and topological properties of the generated PXD background data via clustering analyses and Topological Data Analysis (TDA). Finally, I study the tracking analysis of the generated PXD background in comparison to the real PXD detector background data. Consequently,

I demonstrate the efficacy of YonedaVAE, which excels not only in the OOD simulation of PXD background but also effectively manages to condition its output based on both sensor locations and background levels.

In the final chapter, Chapter 7, a summary of the study's key findings will be presented, along with a thoughtful discussion regarding the limitations. These limitations offer crucial insights for potential refinements in the current methodologies and lay the groundwork for the next phase of investigations in this domain.

Chapter 2

Belle II experiment: The PXD Saga

The Belle II experiment, a precision marvel of modern particle physics located at the SuperKEKB accelerator in Tsukuba, Japan, initiated data collection [9] from electron-positron collisions in March 2019. It has achieved an unprecedented instantaneous luminosity, reaching up to $4.7 \times 10^{34} \text{ cm}^{-2}\text{s}^{-1}$, and accumulated data equivalent to 424 fb^{-1} [10]. A pivotal component of the Belle II apparatus is its Vertex Detector (VXD), which is instrumental for the precise reconstruction of both primary and secondary decay vertex locations. The VXD is comprised of an outer section with four layers of Silicon Vertex Detector (SVD). The inner section features a two-layer PiXel Detector (PXD), the protagonist of this thesis.

In the ensuing sections of this chapter, I will guide you through the experimental setup of the Belle II experiment and the PXD sub-detector. As we traverse this journey, we will delve into the nuances of the responses recorded by the PXD and their origin. Eventually, I will discuss the simulation of PXD data signatures and its challenges. By the end of the chapter, I discuss the plan for this thesis in the following chapters.

2.1 Belle II Experiment

The Belle II experiment [3] represents the next chapter in a rich legacy of collider experiments focused on B meson physics, collectively known as B factories. Situated at KEK in Tsukuba, Japan, Belle II is the successor to the original Belle experiment and operates in conjunction with the SuperKEKB accelerator. This state-of-the-art facility is an asymmetric electron-positron collider featuring a High Energy Ring (HER) for electrons at 7 GeV and a Low Energy Ring (LER) for positrons at 4 GeV. Designed to achieve a peak luminosity of $6.5 \times 10^{35} \text{ cm}^{-2}\text{s}^{-1}$, it aims to outperform its predecessor, KEKB, by a factor of 40.

The Belle II detector employs a Cartesian coordinate system that is right-handed for its spatial description (see Figure 2.1). The origin of this coordinate framework is situated at the point where particle interactions are expected to occur, commonly known as the nominal interaction point (IP). At this point, both beams are crossing at an angle of 11 mrad and collisions occur. The orientation of the axes is established as follows:

- z -axis: This axis is aligned with the magnetic field generated by the solenoid, and

parallel to the e^+ beam. It essentially provides a longitudinal perspective, running parallel to the primary axis of the detector.

- y -axis: Oriented in the vertical direction, this axis points towards the ceiling of the detector hall. It serves as the upward vertical reference for the system.
- x -axis: This axis extends radially, pointing outward in a direction perpendicular to the plane formed by the z and y axes. It is oriented towards the exterior of the accelerator ring.

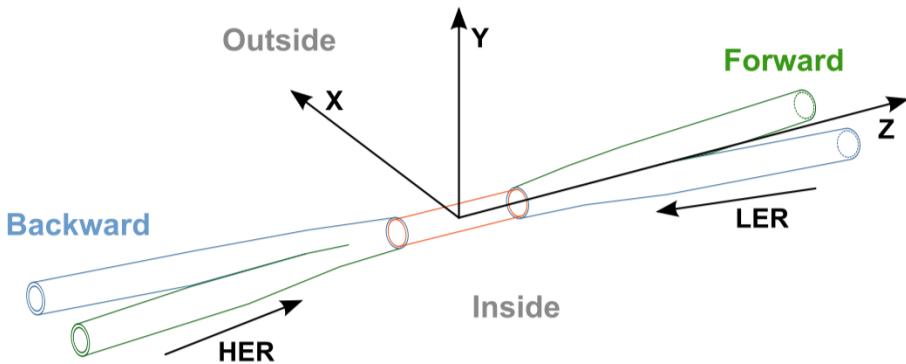


Figure 2.1: The Belle II coordinate system.

In this coordinate system, the positive z -axis is directed towards the “forward” orientation while the negative z -axis indicates the “backward” direction. Unless stated otherwise, all projections of the Belle II detector onto the xy -plane are designed to be viewed from the forward end to the backward end of the detector.

The collision dynamics are carefully controlled to yield a center-of-mass energy of $E_{\text{CMS}} = \sqrt{4E_{\text{HER}}E_{\text{LER}}} \approx 10.58 \text{ GeV}$, aligning closely with the mass of $\Upsilon(4S)$ resonance. In this setup, B and \bar{B} mesons are produced and almost immediately decay, initiating complex chains of subsequent decays. These chains are critical to the study of B meson physics, as they often yield a final set of stable particles whose properties can be thoroughly analyzed.

The asymmetry along the z -axis in Belle II’s detector design is a response to the unique collision dynamics, where the center-of-mass frame and the flight trajectories of produced B mesons are both directed towards the forward part of the detector. This forward focus introduces a Lorentz boost to the B mesons, effectively increasing their flight length within the detector. Thus, longer flight length yields a better resolution on lifetimes, enabling more precise measurements. This leads to a division of the angular acceptance into three separate regions in terms of polar angles: the Forward, Barrel, and Backward regions. The polar angle θ is the angle in the (x, z) -plane with respect to the z -axis. Specifically, the Forward region encompasses angles $17^\circ < \theta < 30^\circ$, the Barrel region is defined by $30^\circ < \theta < 125^\circ$, and the Backward region spans $125^\circ < \theta < 150^\circ$.

All but the outermost component of the Belle II detector operates within a uniform solenoidal magnetic field of magnitude 1.5 T, oriented parallel to the detector’s main axis.

A recorded collision, namely an “event”, involves mostly $e^+e^- \rightarrow \Upsilon(S) \rightarrow \bar{B}B$ reaction that typically yields around 12 tracks. The particles most commonly detected in such events are electrons, positrons, photons, muons, π^\pm mesons, K^\pm mesons, protons, and deuterons.

Belle II is engineered to focus on capturing decay products of B mesons due to their extremely short lifetimes. It features a cylindrical, multi-layered structure that envelopes the Interaction Point (IP), where the particle collisions take place. The sub-components of the Belle II detector are specialized for measuring various particle properties. Among these are the tracking sub-detectors, responsible for gauging particle momentum and locating decay vertices. Additionally, there are particle identification sub-detectors to classify the type of particle and a calorimeter tasked with reconstructing the energies of photons and electrons. The detector consists of a beam pipe and the following sub-detectors from inside out, as shown in Figure 2.2 and Section 2.1, the Vertex Detector (VXD) (the Pixel Vertex Detector (PXD) and Silicon Vertex Detector (SVD)), the Central Drift Chamber (CDC), The Time of Propagation Counter (TOP), The Aerogel Rich Detector (ARICH), The Electromagnetic calorimeter (ECL), and the KLong and Muon detector (KLM). Each plays a specific role, and together, they provide a comprehensive dataset essential for the reconstruction of B meson decays.

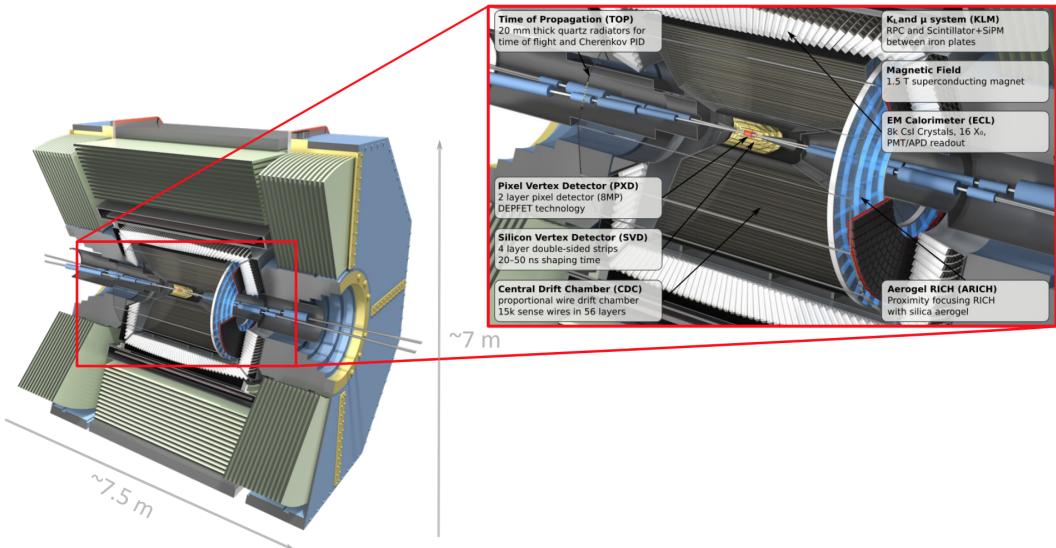


Figure 2.2: The Belle II detector with a zoomed-in view of the inner part adopted from [11]

The original Belle experiment and its contemporary BaBar set remarkable milestones, including the groundbreaking discovery of CP violation in the B meson system [12]. This discovery led to the validation of the Kobayashi-Maskawa model and earned a Nobel Prize in 2008 [13, 14]. By combining advancements in technology and insights from previous experiments like Belle and BaBar, the Belle II experiment aims to push the frontiers of our understanding of particle physics in the luminosity front.

Table 2.1: Summary of Belle II detector components, adopted from [15]

Purpose	Name	Component	Configuration	Readout channels
Beam pipe	Beryllium	Cylindrical, inner radius	10 mm, 10 μm Au, 0.6 mm Be, 1 mm paraffin, 0.4 mm Be	-
Tracking	PXD	Silicon Pixel (DEPFET)	Sensor size: $15 \times (L1\ 136, L2\ 170) \text{ mm}^2$, Pixel size: $50 \times (L1a\ 50, L1b\ 60, L2a\ 75, L2b\ 85) \mu\text{m}^2$	10M
	SVD	Silicon Strip	Rectangular and trapezoidal, strip pitch: $50(p)/160(n) - 75(p)/240(n) \mu\text{m}$	245k
	CDC	Drift Chamber	14336 wires in 56 layers, inner radius of 160mm outer radius of 1130 mm	14k
Particle ID	TOP	RICH with quartz radiator	16 segments in ϕ at $r \sim 120$ cm, 275 cm long, 2 cm thick quartz bars	8k
	ARICH	RICH with aerogel radiator	2×2 cm thick focusing radiators, HAPD photodetectors	78k
Calorimetry	ECL	CsI(Tl)	Barrel: $r = 125 - 162\text{cm}$, end-cap: $z = -102 - +196\text{cm}$	6624 (Barrel), 1152 (FWD), 960 (BWD)
Muon ID	KLM barrel	RPCs and scintillator strips	2 layers with scintillator strips and 12 layers with 2 RPCs	θ 16k, ϕ 16k
	KLM end-cap	scintillator strips	12 layers of $(7\text{-}10) \times 40 \text{ mm}^2$ strips	17k

2.1.1 Belle II Software Framework

Alongside hardware improvements, the experiment's analysis software has also been revamped to enhance both performance and user experience, culminating in a new software, basf2 [8, 16]. The methods described in this thesis are also implemented in the Belle II software, basf2. In the following, I briefly skim through the bird-eye view of the underlying processes in Belle II data-taking and the software pipeline based on [10]. Later, we go through the most relevant parts: simulation and reconstruction.

- 1. Data Collection and Event Triggers:** In the Belle II experiment, data acquisition occurs in specialized sessions characterized by a constant flux of electron-positron collisions. These collisions happen at an elevated frequency, and the detector system continuously scans the collision outcomes in real time. The focus is on identifying unique markers that signify the creation of $B\bar{B}$ meson pairs. An autonomous component within the Belle II online system (shown in Figure 2.3), known as the "trigger," oversees this detection process. The online system consists of the Data Acquisition (DAQ), Level 1 Trigger (L1), and the High-Level Trigger (HLT). When the trigger discerns that $B\bar{B}$ mesons have been produced, it activates a comprehensive readout of the entire detector, converting this information into a format suitable for downstream analysis and reducing the amount of data as much as possible before they reach the first storage. The DAQ system is there to make sure that all trigger signals are synchronously delivered to all sub-detectors, and provides the high-speed data links to read out the full detector data for each event and forward it to the HLT system. The Belle II trigger, responsible for starting the data readout of the whole detector for interesting events, is engineered to handle collision rates up to 30 kHz at full SuperKEKB design luminosity.
- 2. Event Classification and Data Segmentation:** In particle physics, the term "event" represents a specific, time-bound physical process, commonly a particle decay. Within the context of Belle II, a "signal event" refers to, e.g., the unique occurrence tied to $B\bar{B}$ meson pair and the subsequent decay products generated by each meson. The organized data captured by the full detector readout, instigated by the trigger, exemplifies one such signal event. Collections of these signal events are then grouped into datasets that serve as the primary material for scientific discovery. It is crucial

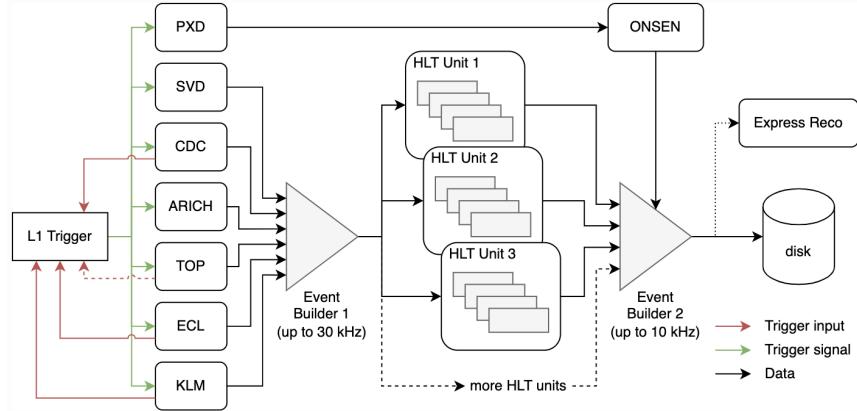


Figure 2.3: A schematic and simplified diagram of the Belle II data flow, taken from [15]

to distinguish these signal events from “background noise,” which includes extraneous physical processes that can interfere with the experimental results. They will be discussed in detail during the next section. As the luminosity of SuperKEKB escalates through a reduction of transverse beam size and a doubling of the beam currents, compared to its predecessor KEKB, the background noise at the Belle II detector is substantially raised. Generally, this background interference originates from two primary sources: beam-induced processes and luminosity backgrounds. Beam-induced backgrounds are a consequence of beam particles interacting with elements like residual gas inside the beam pipe, bending magnets, or other particles within the same bunch. On the other hand, luminosity backgrounds result from beam collisions that yield inconsequential physics phenomena, such as Bhabha scattering or two-photon interactions. Identifying and discarding these background processes is of paramount importance, especially for low-multiplicity analyses.

3. **Event generation and Simulation:** The ability to compare the observed data from the detector with theoretical expectations is paramount for the validation of results in particle physics. To fulfill this requirement, simulated events are generated that mimic the behavior of real detector events as closely as possible. This simulation process relies heavily on the Monte Carlo (MC) method, which involves the repeated sampling of pseudo-random numbers. Simulated data enables the estimation of the statistical uncertainties associated with the recorded data and the estimation of background noise. It is also used to validate the detector and its operation, estimate the detector’s efficiency, and perform calibration purposes. Simulation of data involves two steps, namely, event generation and the simulation of the detector response of each generated event. *Event generation* consists of simulating the physics interactions under study. Based on the initial conditions set for colliding electrons and positrons, a range of particles are generated in accordance with the specific physics model being examined—be it advanced theories. These generated events are often categorized into different samples for specific analyses: “signal MC” pertains to the decay process being

studied, whereas “generic MC” refers to basic standard model processes. Occasionally, additional “background MC” samples may be generated for processes that need to be excluded from the analysis. For Belle II, this phase is efficient and tailored for various study objectives.

4. **Detector Simulation:** Upon generating the essential 4-momentum vectors and taking the detector geometry and the magnetic field into account, the next task is to simulate their interactions with the detector material, capturing processes such as ionization, scintillation, Bremsstrahlung, pair production, and Cherenkov radiation, among others. The simulation of the detector response is called “digitization.” This part of simulation has been the subject of extensive research and development, culminating in the simulation software Geant4 [17]. Geant4 takes the generated particles and replicates their interactions within a virtual Belle II environment. Subsequently, the energy deposition and particle interactions within each sub-detector are computed. Geant4 performs the simulation by tracking the particles, one at a time, through the geometry. It takes into account the effect of the magnetic field on the particle, the energy loss, and multiple scattering the particle experiences while traversing material and various other electromagnetic and hadronic effects. If a particle decays, the decay products are added to the list of particles. The initial four-momentum vectors given to Geant4 are called *primary particles*, while all particles created from interactions or decay during the simulation are called *secondary particles*. Additional specialized software modules then convert these Geant4 outputs into realistic detector signals. The digitization is the last step in the processing chain used solely for Monte Carlo data. Post-digitization, the workflow for both Monte Carlo and real-world data aligns in terms of data processing steps. For instance, the pixel detector software translates the energy deposits into pixel activations, *PXD digits*. This part of the simulation is the computational bottleneck both from the time and space (storage) complexity perspective.

For example, the PXD background data has the highest storage consumption among all sub-detectors, with an utterly infeasible hypothetical online simulation time. A solution to alleviate the detector simulation problem is to use surrogate models under the topic of “Fast Simulation,” the main focus of this thesis.

5. **Reconstruction Procedures:** The term “reconstruction” in this setting refers to the act of methodically determining the attributes of a decaying B meson and its decay products. These reconstructed signal events serve as the foundational data for the downstream physical analyses. The reconstruction workflow adopts a bottom-up strategy, starting with the final state particles and their detector responses and moving backward to recreate each decay event until the originating $B\bar{B}$ pair (or one B) is reached. However, it’s never possible to uniquely identify all the particles in the interaction because of not only hadronic interactions and background processes but also because we a priori don’t know which decay products correspond to which mother particles. So, all we can do is look at the detector response, find a set of most likely

particles, and then leave it to the analyses to do a proper statistical analysis of the events. The first stages of reconstruction consist of two main procedures:

- **Clustering:** One of the first steps in reconstruction is “clustering”, where one needs to modulate the detector responses in each sub-detector if they are related. This is done by means of a clusteriser, which groups together adjacent detector signals (nearest neighbor) into clusters, performed for both real data and Monte Carlo data in the same way. One can then calculate geometric properties of these clusters like size, shape or center. For PXD, either a center of gravity or an analog head-tail algorithm (for cluster multiplicities larger than two hit pixels), is used. These algorithms aggregate and interpret data from pixelated detectors such as PXD that results in “PXD Clusters.” For real detector data, the simulation stage is skipped, and the input is the fired pixels, “PXD Digits.” The “Center of Gravity” (cog) method is widely used for localizing the point where a particle has interacted with a detector, especially in situations where the interaction affects more than one pixel. The idea is to take a weighted average of the positions of all the pixels in a cluster, with the weights being the ionization strengths (e.g., charge, energy deposition, etc.) in those pixels, as:

$$\text{cog} = \frac{\sum_{i=1}^N w_i \cdot x_i}{\sum_{i=1}^N w_i}$$

Where w_i is the weight (often the energy deposition) of the i^{th} pixel, and x_i is the position of the i^{th} pixel. N is the total number of pixels in the cluster. The “Analog Head-Tail” algorithm is used primarily when the cluster multiplicity is larger than two hit pixels. The algorithm aims to identify the ‘head’ and ‘tail’ of a cluster, which correspond to the entry and exit points of the particle in the detector plane. This is particularly useful for distinguishing between particles that might have similar total charge deposition but different directions.

The clustering of the PXD pixels is done in a row-wise manner, with increasing values for the column index and the row index. Starting with the pixel in the upper left corner. The clusteriser of basf2 checks each pixel to make sure that the ratio of its charge over a common noise level is above a given threshold. If it is, the left neighbor in the same row and the direct neighbors in the previous row are investigated. If one or more clusters have already been found in those neighboring pixels, the clusters are merged and the pixel is assigned to this cluster. Otherwise, a new cluster is created, and the pixel becomes its first member. The clusteriser proceeds with the pixel to the right of the current pixel or, if the pixel is the last pixel in the current row, with the first pixel of the next row. The procedure is repeated until the last pixel in the last row has been processed. This clustering scheme investigates each pixel only once and requires only the current and the previous pixel row to be stored in memory, making it an efficient and memory-saving pixel clustering method. After having grouped

all pixels into clusters, the position of each cluster is determined: if the size of the cluster, defined by the number of pixels belonging to the cluster, exceeds a given threshold, the head-tail algorithm is used to calculate its position. The default threshold is set to 3 pixels. The Analog Head-Tail algorithm calculates the position of the cluster by using the outermost pixels of a cluster (head and tail).

- **Track Reconstruction:** The Tracking (Track Reconstruction) of Belle II is designed to capture the spatial location of charged particles as they move through the detector. Reconstruction involves the inference of the trajectories of particles through the tracking detectors, called tracks. Multiple sensor layers work in tandem to collect this positional data, enabling the accurate reconstruction of particle paths, known as “tracking.” Belle II incorporates two primary tracking steps to ensure precise trajectory identification and thereby contribute to accurate decay vertex estimates, the track finding and the track fitting step. During the track finding step, the clusters from the PXD/SVD and the fired wires in the CDC that belong to the same track are identified. In other words, it finds patterns in the hits or hit clusters in the tracking detectors. Then, the identified clusters and wires are passed to the track fitting, which estimates the optimal track parameters. Here, the goal is to determine the best estimate of the kinematic variables describing the particle trajectories corresponding to each found hit/cluster pattern to obtain the particle position and momentum close to the interaction region as precisely as possible.

2.2 Pixel Vertex Detector (PXD)

The Pixel Vertex Detector (PXD) [11] serves as the innermost layer of the Belle II detector complex, shown in Figure 2.4 and Figure 2.6. This semiconductor-based apparatus employs DePFET (Depleted P-channel Field Effect Transistor) technology, a semi-conductor sensor that combines the detection of the passage of charged particles and the amplification of their deposited energy within one device. The PXD is the first full-scale detector employing the DePFET technology in High Energy Physics. The primary function of the PXD is to determine the positions of charged particles that emerge from collision events. A key objective is the highly accurate determination of decay vertices, a task for which the PXD is strategically located in close proximity to the interaction point (IP). The device spans a comprehensive polar angle range, specifically $17^\circ < \theta < 150^\circ$, within the Belle II setup. The PXD incorporates 40 pixel sensors, depicted in Figure 2.6b, which are organized into two concentric circles (annulus) around the IP, termed the inner and outer layers. These pixel sensors are paired into structures known as “ladders,” aligned along the z -axis. The geometrical arrangement of these ladders in the x - y plane resembles an octagon for the inner layer and a dodecagon for the outer layer.

Each of these pixel sensors integrates a 250×768 matrix of silicon pixels that are engineered to interact with charged particles, as shown in Figure 2.5. The activation

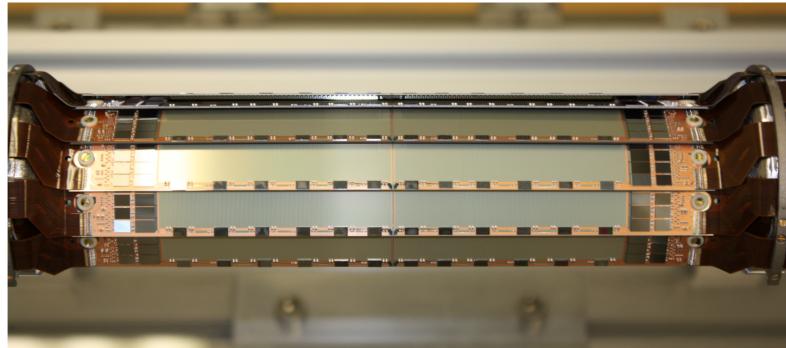


Figure 2.4: The first picture of the full PXD during the commissioning phase in DESY Hamburg.

of a pixel is closely linked with the ionization of underlying material within the silicon medium and producing electron-hole pairs. Data relating to the spatial coordinates of the particle-pixel interaction and the number of ionized electrons (accumulated charge) is then collected as “hits” (see Figure 2.6c).

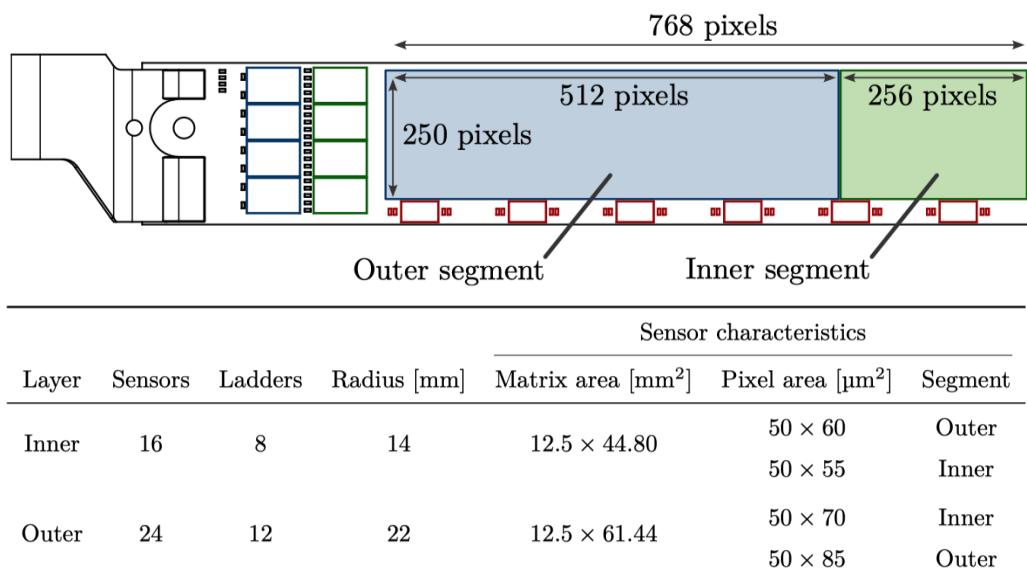


Figure 2.5: PXD specifications and detailed information adopted from [18]

The output from the pixel sensor is essentially a map detailing the hits and their corresponding levels of charge displacement, measured during the readout interval. This raw data undergoes noise filtering and digitization, converting it into 8-bit integers that range from 0 to 255. Given the technical constraints of silicon pixel technology, physical necessities, and the need for rapid readout, the pixel sensors are calibrated for optimal spatial resolution, thus ensuring adequate tracking precision [18].

Remarkably, the PXD can generate an “ultra-high granular” amount of raw data, owing to its intricate pixel configuration comprising a total of 7.68×10^6 pixel channels. The data production rate can reach up to 20 GB/s, outpacing the combined data rates of all other components in the detector by over a factor of ten. The PXD necessitates a parallel readout time of $20 \mu\text{s}$ from the individual pixel sensors which is considerably longer than the approximately $10 \mu\text{s}$ required for a beam particle to complete one circuit of the collider.

2.2.1 PXD Background: Types and Levels

Due to its proximity to the IP and the incoming particle beams, the PXD is susceptible to a huge amount of radiation and background levels. PXD hits generated by background shower particles deteriorate the detector’s physics performance. The rates of these background processes are correlated with multiple factors e.g. beam size, beam current, luminosity, accelerator status, and vacuum conditions [15]. Generally, the processes contributing to background in the detector can be classified into five main categories:

- **Touschek Background:** At Belle II, Touschek scattering is a predominant source of background. This phenomenon involves Coulomb interactions between two particles within the same beam bunch. A “bunch” refers to a group of particles that are packed together and travel around the accelerator ring almost as a single unit. These bunches are created to maximize the chances of interaction between particles when two such bunches cross paths in the detector. As a result of such interactions, the energy levels of the two participating particles diverge from the nominal beam energy; one particle experiences an energy increase while the other undergoes an energy loss. The rate of Touschek scattering is directly related to the square of the beam current and inversely related to both the number of bunches in the accelerator ring and the size of the beam [19].
- **Beam-Gas Background:** Scattering between the beam and residual gas molecules within the beam tube stands as another significant source of background noise at Belle II. Two types of beam-gas scatterings are prevalent: the elastic Coulomb scattering, which alters the trajectory of the beam particles, and the inelastic Bremsstrahlung, which diminishes their energy. The rate of such scatterings is directly linked to the residual gas pressure inside the tube and the current of the beam.
- **Luminosity-dependent Background:** Background arising from beam interactions at the Interaction Point (IP) is referred to as luminosity-dependent background, and its intensity is directly proportional to the luminosity, as shown in Figure 2.7¹. In the case of SuperKEKB, the target luminosity is approximately 40 times greater than KEKB’s peak luminosity, making the luminosity-dependent background a significant concern. A critical form of this background stems from radiative Bhabha scattering ($e^+e^- \rightarrow e^+e^-\gamma$). In this process, beam particles emit photons and deviate from their

¹Adopted from the PXD Analysis meeting 2.11.2022. by D.Pitzl

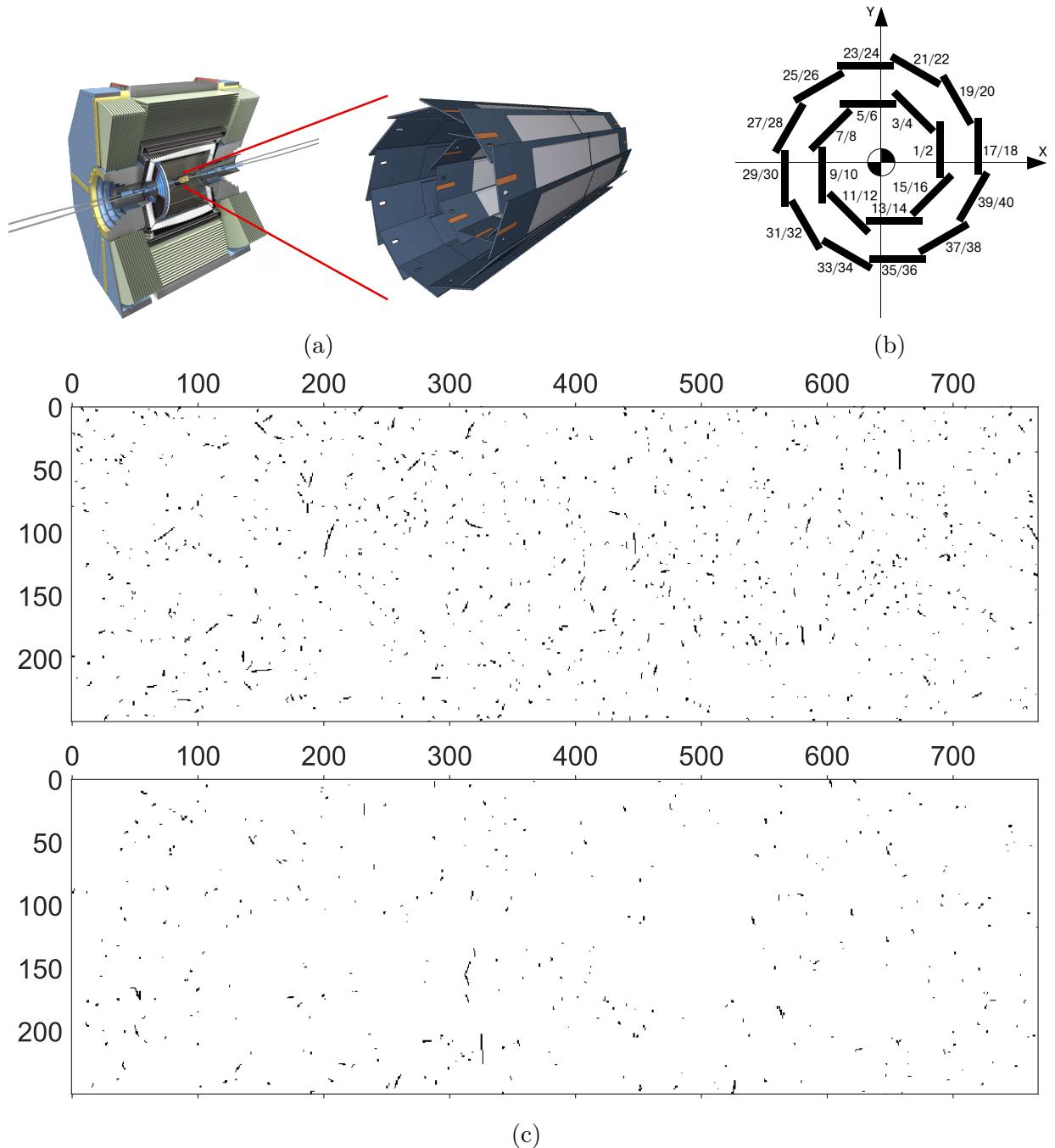


Figure 2.6: **Pixel detector layout** The pixel detector (PXD) is the inner-most sub-detector of the Belle II experiment (a) and is configured in a two-layered overlapping sensor structure (b). Geant4 Simulated PXD Background image examples (c) for sensors 7 (top) and 25 (down).

nominal paths. At a high luminosity regime, this background dominates over other

Belle II backgrounds. Two-photon processes ($e^+e^- \rightarrow e^+e^-e^+e^-$) represent another source of background. The emitting low-momentum electron-positron pairs cause the beam particles to lose energy, similar to the radiative Bhabha process. Moreover, when possessing low transverse momentum, the emitted electron and positron curl in the Belle II solenoid field, making multiple hits on the PXD.

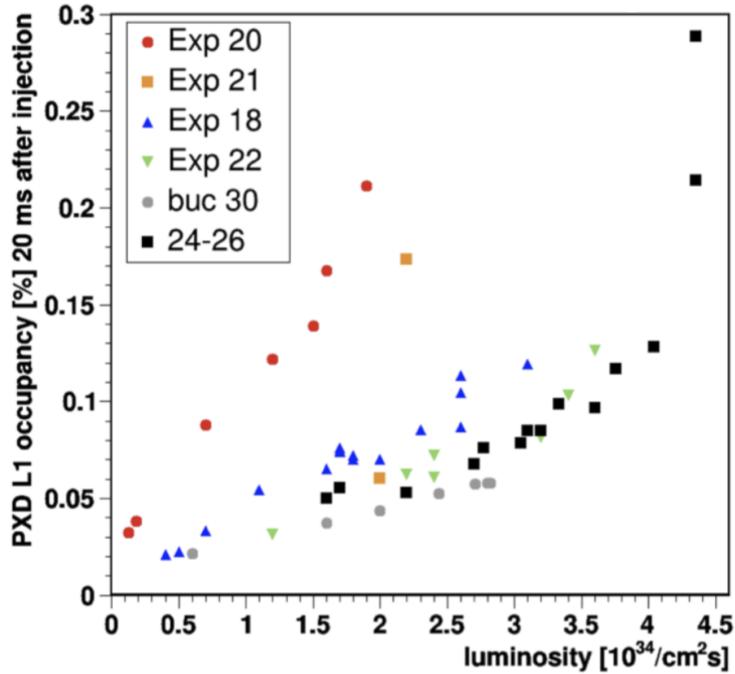


Figure 2.7: PXD occupancy (layer 1 here) increases with beam currents and luminosity, coming from raw data. The occupancy of PXD is defined as the ratio of the number of hits to the total number of PXD pixels. The Experiment numbering corresponds to the Belle II run periods.

- **Background from Synchrotron Radiation:** Another background component affecting the inner detectors of Belle II arises from Synchrotron Radiation. The intensity of this radiation is governed by the square of both the beam energy and the magnetic field strength. Thus, the HER electron beam predominantly contributes to the Synchrotron Radiation background. The energy of Synchrotron Radiation photons impacting the PXD and the SVD ranges from a few kiloelectronvolts (keV) to multiple tens of keV.
- **Background Due to Injection:** The beam lifetime in SuperKEKB is notably brief, necessitating frequent top-up injections through a betatron injection scheme [20]. When the overall beam current drops below 99% of the nominal current, additional charge is injected into low bunch-current buckets at varying frequencies (between

1–25 Hz). These new bunches oscillate horizontally around the main beam, causing temporary spikes in the background levels of the Belle II detector. These spikes last for milliseconds each time the newly-injected bunch crosses the interaction point.

Thus, these traversing background processes can deposit considerable energy, thereby reducing the lifespan of the detector and creating malfunctioning pixels or zones of reduced efficiency. Moreover, due to its proximity to the IP, PXD is also vulnerable to back-scattered low-energy Synchrotron Radiation photons. This huge amount of background manifests itself primarily in a sensor-level observable called the “occupancy.” The occupancy of a pixel sensor, defined as the ratio of the number of hits to the total pixel count in the matrix, is directly influenced by the level and type of background noise. Currently, the average PXD occupancy is below 0.3%.

In reality, there exists a constraint due to bandwidth limitations. For a 30 kHz trigger rate, data loss starts to become a factor when the average occupancy of the inner PXD layer exceeds 3%. Beyond this occupancy threshold, offline performance experiences a marked decline due to issues like cluster merging and a higher likelihood of incorrectly associating hits to particle tracks. It should be noted that performance degradation can begin even below this specified level.

2.3 PXD Background Simulation: Ideas and Challenges

To accurately replicate experimental conditions, emulating background processes is essential. Two methods are currently used, as depicted in Figure 2.9, for implementing background simulation: “background simulation” and “random trigger.” Both techniques have their pros and cons, and the choice between them depends on the specific requirements of the simulation and the resources available.

2.3.1 Background Simulation

In the “background simulation” approach, a collection of synthetically emulated background events serves as a resource. These backgrounds are simulated with the software framework Strategic Accelerator Design (SAD) [21]. SAD, initialized with beam optics parameters and detector elements, tracks scattered particles through the sequence of detector elements. The tracking simulation initiates by establishing uniformly spaced scattering zones around each ring, generating particle bunches in these regions. These particles are randomly created within a three-dimensional Gaussian distribution. Their momentum and statistical weight are calculated using established scattering theories, specifically Coulomb, Bremsstrahlung, and Touschek scattering. These background events are essentially the particle outcomes of specific background processes. Using Geant4 for detector response simulation, the PXD digits are then stored in a dedicated file. This file is then utilized to overlay background particles onto those from a signal or physics event, as shown in Figure 2.9a. This method

improves the quality of digitization by combining energy deposits to generate sensor responses. However, it neither fully reconciles the discrepancy between simulated and real background data nor is it computationally efficient. The reason is that it has to be pre-generated and stored, thus creating a high “space-complexity” from the computational point of view, as shown in Figure 2.8. On the other hand, it is extremely inefficient to run this on the fly and creates a high “time-complexity” issue.

2.3.2 Random Trigger

In the “random trigger” technique, an alternative to direct simulation, real experimental sensor responses to background noise are directly utilized randomly. This approach operates on the premise that each sensor’s response can be associated with a unique background event. The procedure begins with a focus on simulating the energy deposits of only the signal particles. Subsequently, sensor-specific responses from actual recorded background events—coming from random trigger events—are overlaid during the digitization stage, as shown in Figure 2.9b. The random trigger activates data collection separate from the event signature. Three kinds of random triggers exist [18]: a periodic one synced with SuperKEKB’s bunch crossing signal, a pseudo-random trigger based on an independent local clock, and a delayed Bhabha trigger that fires shortly after a specific bunch passes. This results in a more faithful simulation of real PXD conditions than the former technique. This method even allows for the incorporation of noise from detector electronics. However, it also comes with challenges, such as the potential loss of certain background properties due to threshold effects, the lack of real detector background response for higher (undetected) luminosity, and as always, the requirement for storage and transfer of a large volume of extra data. Thus, this also creates a high “space-complexity” issue from the computational point of view.

2.4 PXD Background Simulation: Surrogate models

A strong solution to the above challenges and issues for PXD simulation is “Surrogate models”. Surrogate models for fast detector simulation are simplified, computationally efficient approximations that emulate the behavior of more complex, detailed simulations of particle detectors on the fly of the analysis pipeline. Traditional detector simulation methods, as discussed above, are very computationally intensive and storage costly. A surrogate model aims to replicate the essential features of a full detector simulation but at a fraction of the computational cost. These models are constructed using Deep Generative Models (DGM), where the surrogate model is trained on a dataset either generated by the Geant4 simulation or the real detector data. Once trained, the model can generate samples that are statistically similar to their training data and even generalize to the data beyond the training data. There are some key requirements that the model should meet to be effective and efficient:

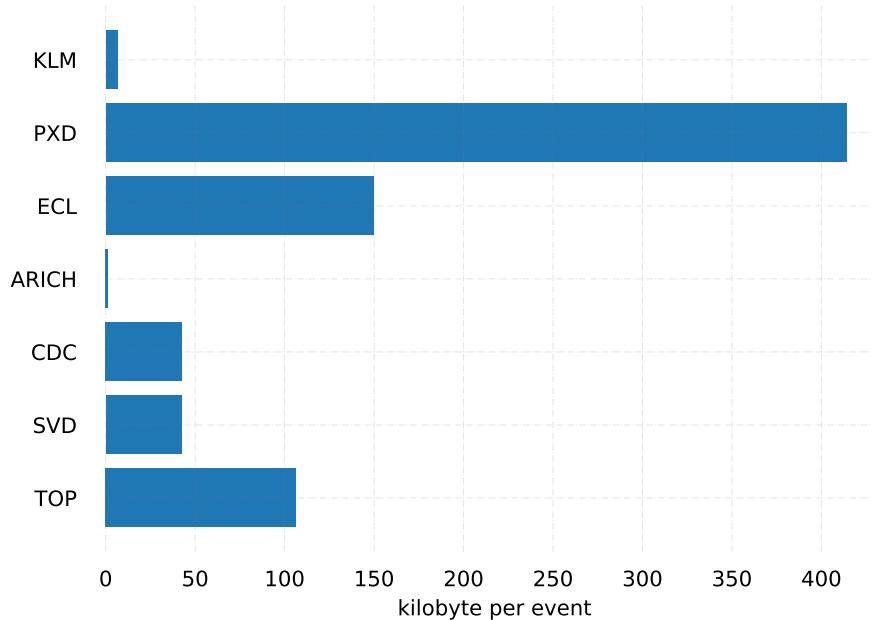


Figure 2.8: Geant4 simulated Background data volume comparison of PXD data per event compared to other sub-detectors at Belle II. The PXD background data is nearly as much as the combined background data for all other sub-detectors.

1. **Low time-complexity:** The surrogate model must be computationally efficient in order to facilitate fast simulations. Computational speed is crucial when performing large-scale simulations or when needing to iterate the model many times for optimization or fine-tuning. Thus, the model should take advantage of parallel processing capabilities and have to be optimized for the hardware it is expected to run on, whether it's a CPU or GPU.
2. **Low space-complexity:** It should come with a minimal storage cost. Hence, the underlying compression technique has to reduce the storage footprint without sacrificing too much in terms of the downstream physics analysis.
3. **Realistic and Diverse:** It has to generate samples as diverse as possible from the downstream physics analysis point of view. Thus, the sampling techniques should be capable of employing the nuanced behaviors and symmetries of the PXD to ensure that the model captures the diversity inherent in the real data.
4. **Extrapolation:** It has to be able to extrapolate to background levels beyond the current beam parameters and luminosities in order to analyze the PXD operation at higher luminosity and to do physics analysis beyond the current experimental limits. Therefore, the model should be robust against overfitting and incorporate a measure of control when extrapolating to give a range of plausible outcomes.

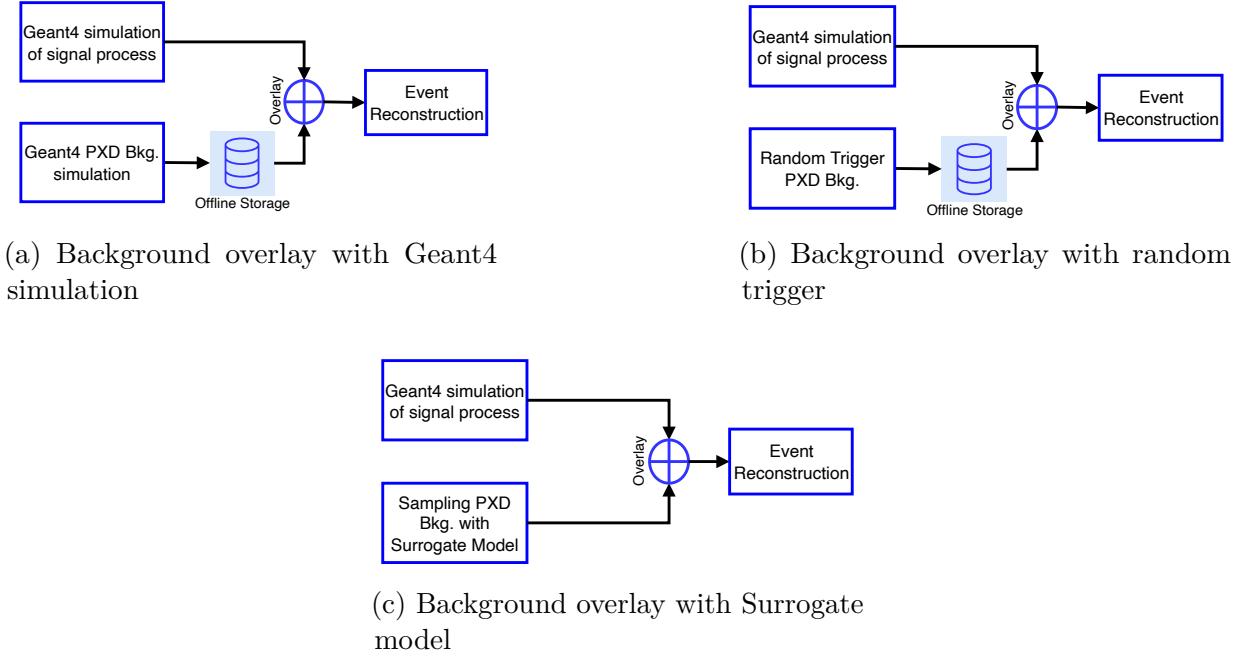


Figure 2.9: PXD Background simulation options.

In this thesis, I propose the surrogate modeling solution as an amortized simulator for the PXD background simulation as a fast and efficient simulation method, as shown in Figure 2.9c. First, a DGM is developed to emulate Geant4 simulated background data as an approximation of the real data with much lower data complexity and implement this approach into the Belle II software, basf2. Then, for the final challenge of real PXD detector data, I introduce a much more efficient DGM as a solution that successfully satisfies all the above conditions.

2.5 PXD Background Simulation: Figure of Merits

For evaluation, we have three main categories of metrics: Low-level Metrics, Neural Network-based Metrics, and Physics-Level Metrics. These metrics evaluate different aspects of the generated PXD background. Through the next chapters, I elaborate on and incorporate them to analyze and compare the results with each other. Some of them are being introduced for the first time in the Fast Detector Simulation domain. In the following, I briefly introduce them.

2.5.1 Low-Level Metrics

For Low-Level metrics, we are interested in analyzing the histogram projections of the data. Therefore, the Low-Level observables as marginal distributions include charge distribution

and average charge values per sensor, occupancy distribution and average occupancy per sensor, and the correlation between the average occupancies.

2.5.2 Neural Network-based Metrics

The necessity for incorporating Neural Network (NN)-based metrics in the evaluation process of PXD background simulation arises from two main factors:

- **Complexity of Data:** PXD background data is intrinsically complex and is influenced by numerous variables originating from the underlying Physics processes. Moreover, they exist in a high-dimensional space, often making evaluating their quality and diversity challenging using traditional metrics. Thus, simple 1D and 2D histogram-based Low-Level observables might not fully capture this complexity. NN-based metrics can distill this high dimensionality into a more manageable form by focusing on relevant feature spaces. This is particularly useful for understanding what aspects of the data are being captured (or missed) by the DGM, such as the data modality (mode collapse).
- **Quantitative Assessment:** While Low-Level metrics provide a direct evaluation of specific properties, they might lack the capacity for an overall quantitative assessment of data similarity. NN-based metrics offer a single quantitative score that can be compared across different models or iterations.

Thus, NN-based metrics offer a multifaceted evaluation approach that *complements* the Low-Level metrics. They enable a more comprehensive, nuanced, and rigorous analysis of the performance and reliability of PXD background simulations. The NN-based metrics that I incorporate in this thesis are the Frechet Inception Distance (FID) (see Chapter 5) [22], Kernel Inception Distance (KID) (see Chapter 5) [23], and Vendi Score (see Chapter 6) [24] and show how they stand out as particularly insightful and interpretable evaluation tools. FID and KID are used to measure the similarity between the distributions of generated data and real data in the feature space of an NN model that is pre-trained on the PXD background dataset with a very high precision. Vendi score, on the other hand, is to measure mode collapse (see Chapter 3) and the diversity evaluation problem (see Chapter 6) of the generated data.

2.5.3 Clustering and Topological Data Analysis

In PXD background simulation, a fundamental question arises concerning the spatial organization of PXD hits. Specifically, how are points in a PXD point cloud clustered? What is the complexity of these clusters, and what topological and geometrical features are discernible? Or, as we adjust our observation window, what shifts can be observed in the geometric representation of the PXD hits?

Topological Data Analysis (TDA) serves as an instrumental tool for investigating these questions. It allows us to delve into the intrinsic shape of the data by formalizing the notions

of proximity and continuity of data. TDA focuses on identifying topological features in data sets, such as connected components, loops, and higher-dimensional cycles (see Chapter 6). These features offer invaluable insights into the structural properties of PXD hits across multiple scales, which is especially useful for identifying underlying patterns and anomalies for both the real PXD data and the generated ones. This is the very first time that detector signatures are being analyzed through the lens of TDA.

Crucially, the reconstruction process in Belle II relies on clusters of PXD hits to perform its analyses. As such, it becomes imperative to accurately reproduce the background on this cluster level. To address this, I further complement the TDA perspective with clustering analysis implemented in the Belle II software. Integrating these two approaches aims to capture a more complete understanding of the structural, geometrical, and topological properties inherent in the PXD background data. Clustering analysis in Belle II offers a different set of evaluative tools that focus on grouping data points based on similarity measures such as distance or density. The complementary nature of these methods allows for a more robust validation mechanism, ensuring that the background is faithfully represented at the cluster level. This integrated approach provides an interpretable framework for the TDA and the clustering analysis, fulfilling the requirements for a precise and comprehensive background simulation.

2.5.4 Physics-Level Metrics

Eventually, within this study, I also analyze the effect of generated and simulated PXD backgrounds on high-precision charged track reconstruction and compare them. This is indeed crucial for both decay vertex measurement, quantifying and correcting systematic uncertainties, and any subsequent physics analyses. The Helix parameters quantify the relative distance to the Point of Closest Approach (POCA), a reference point on the particle track closest to the center of the coordinate system. The POCA is determined by extrapolating a particle track to the global detector z-axis. In the Belle II experiment, the trajectory of charged particles in a uniform magnetic field can be encapsulated by five helix parameters $\{d_0, z_0, \phi_0, \omega, \tan \lambda\}$ relative to POCA, depicted in Figure 2.10.

1. d_0 : Refers to the impact parameter, indicating the shortest perpendicular distance from POCA. This provides insight into the spatial offset of the helix trajectory from the coordinate system's center.
2. z_0 : Denotes the z-coordinate at the point where the helix is closest to the POCA, revealing the helix's vertical positioning relative to the central z-axis.
3. ϕ_0 : Represents the azimuthal angle of the particle's momentum at its nearest proximity to the POCA, specifying the helix's initial rotational direction on the xy-plane.
4. ω : Characterizes the curvature of the helix, which is inversely related to the radius of curvature R . The curvature provides a sense of the helix's tightness and is influenced by particle momentum and magnetic field intensity.

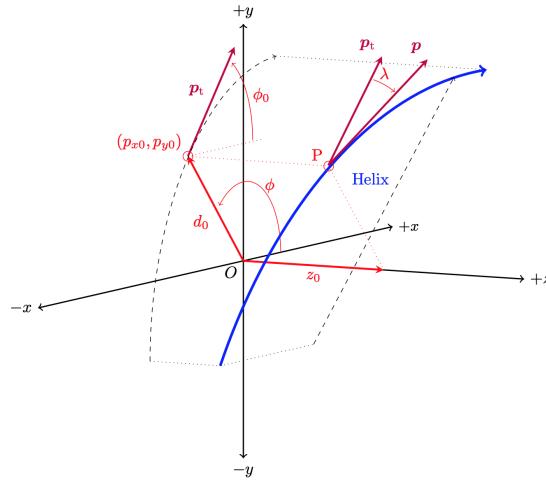


Figure 2.10: The perigee parametrization of the track helix adopted from [25]

5. $\tan \lambda$: Corresponds to the tangent of the dip angle λ , determining the helix's inclination with respect to the xy -plane. A flat helix exhibits $\tan \lambda = 0$, while a steeper helix is indicated by a larger $\tan \lambda$.

These helix parameters collectively offer a detailed geometric and kinematic description of a charged particle's trajectory. The resolution of these helix parameters serves as a compelling physics-level metric and comparative assessment for studying the effects of PXD background on charged track reconstruction (see Chapter 5).

Chapter 3

On the Shoulder of Giants: The Modern Machine Learning Tools

3.1 Introduction

The arena of Artificial Intelligence has been evolving at a relentless pace, fueled by an incessant pursuit to replicate and augment human cognitive abilities. In this quest, the development of advanced machine learning techniques has been instrumental, providing the bedrock for sophisticated AI systems. This chapter embarks on a journey to explore these techniques, focusing on three pivotal advancements that have revolutionized the landscape: Transformers and Attention Mechanism, Deep Generative Models, and Self-Supervised Learning.

The first section, “Transformers and Attention Mechanism”, dives deep into understanding these models that have redefined the field of natural language processing. With their ability to handle long-range dependencies in sequences, they form a key pillar in this study, providing us with the tools to extend and adapt these mechanisms to the context of detector simulation.

In the “Deep Generative Models” section, I venture into the realm of generative models, the vanguard of modern machine learning that strives to capture and mimic reality. By providing a deep dive into the two main latent variable models, Generative Adversarial Networks and Variational Auto Encoders, I aim to elucidate how these models serve as a launching pad for the innovative techniques and methods that I will introduce in the following chapters.

Finally, the “Self-Supervised Learning” section leads us to the frontier of machine learning research, the Dark Matter of Artificial Intelligence. By harnessing the power of SSL and enabling models to generate their own supervision, it has the potential to significantly improve the efficiency of learning algorithms. This concept will play a central role as I build upon it to devise novel learning strategies.

This chapter is designed to provide a comprehensive understanding of these three pillars of modern machine learning. By dissecting their inner workings, highlighting their strengths,

and understanding their limitations, I lay the groundwork for the novel mechanisms and strategies I will introduce in the coming chapters. As we navigate this journey, we are indeed standing on the shoulders of giants, harnessing their wisdom and insights to forge new paths in the exciting world of AI research.

3.2 Attention Mechanism and Transformers

The concept of attention in neural networks is a powerful mechanism that allows a model to enhance its predictive ability by selectively focusing on specific subsets of data. This idea, inspired by the human cognitive function, assigns learned weights to quantify the degree of attention, thereby forming the output as a weighted average. In essence, humans do not process all incoming information simultaneously; instead, attention is selectively allocated to relevant information when necessary. Initially implemented in the realm of computer vision to alleviate the computational load of image processing, attention was designed to concentrate on specific regions of images rather than the entire picture, effectively mimicking human perceptual tendencies. However, the inception of the attention mechanisms we recognize today is primarily traced back to the field of natural language processing. Bahdanau et al. [26]. employed attention in a machine translation model to rectify the structural issues inherent in recurrent neural networks, subsequently highlighting the benefits of attention. This endorsement paved the way for the refinement and popularization of attention techniques across a multitude of tasks.

3.2.1 Attention Mechanism and Self-Attention

Self-attention represents a unique form of attention mechanism that allows a model to make inferences about a specific section of a data sample by leveraging information from other portions of the same sample in a permutation-invariant way. This concept echoes the principles of non-local means [27], an image denoising technique, where each pixel in the output image is a function of all pixels in the input image.

Formally, an attention mechanism, is the data of $(\mathbf{K}, \mathbf{Q}, \mathbf{V}, \mathbf{s}, \mathbf{A})$. The vector spaces $\mathbf{K} \in \mathbb{R}^{N \times d_k}$, $\mathbf{Q} \in \mathbb{R}^{N \times d_k}$ and $\mathbf{V} \in \mathbb{R}^{N \times d_v}$ are the set of Keys, Queries, and Values. The bilinear map $\mathbf{s} : \mathbf{K} \times \mathbf{Q} \rightarrow \mathbb{R}^{N \times N}$ is a score function between the key and the query where the codomain is an attention score. The query serves as a request for information, and the corresponding attention score quantifies the relevance of the data encapsulated in the key vector with respect to the query. In the case of self-attention, the keys, queries, and values all come from the same sequence, allowing each element to attend to all others in that sequence. Different score functions are the additive [26], simple Dot-Product [28], Scaled Multiplicative [29], General [30], Activated General [31], and similarity-based [32]. Then, the attention, \mathbf{Att} , is defined as,

$$\mathbf{Att}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) := \mathbf{Align}(\mathbf{s}(\mathbf{K}, \mathbf{Q}))\mathbf{V} ,$$

where d_k and d_v are the dimensions of the corresponding vector spaces. The goal of the attention module is to produce a weighted average of the value vectors. Thus, the scores are redistributed via an alignment operation $\text{Align}(\cdot) : \mathbb{R}^d \rightarrow [a, b]$ that maps the non-compact distribution to a compact one between $[a, b]$. Different alignment functions are the soft or global alignment, like the Softmax function [28], that also introduces a probabilistic interpretation to the input vectors. Another variant, the hard alignment [33], forces the attention model to focus on exactly one feature vector in a deterministic sense.

For instance, the attention mapping used in the vanilla Transformer [29] adopts the scaled dot-product as the bilinear map between keys and queries as

$$\text{Att}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) := \text{softmax}\left(\frac{\mathbf{K}\mathbf{Q}^T}{\sqrt{d_k}}\right)\mathbf{V} . \quad (3.1)$$

As the dimensionality of the key, represented by d_k , increases, the dot product of the query and key can potentially escalate in magnitude. If these large values are then passed through the softmax function during the alignment stage, the resulting gradient can diminish significantly, impeding the model's ability to converge effectively. The incorporation of the normalization factor $\frac{1}{\sqrt{d_k}}$ serves as a countermeasure to this issue, effectively preventing the occurrence of vanishing gradients even when dealing with large inputs.

Rather than simply computing the attention once, the multi-head mechanism runs through the scaled dot-product attention of linearly transformed versions of keys, queries, and values multiple times in parallel via learnable maps W_i^k , W_i^q and W_i^v . The independent attention outputs over h number of heads are then aggregated and projected back into the desired number of dimensions via W^p ,

$$\text{MultiHead}(\mathbf{K}, \mathbf{Q}, \mathbf{V}) := \left[\biguplus_{i=1}^h \mathbf{H}_i \right] \mathbf{W}^p , \quad (3.2)$$

where \mathbf{H}_i is given by $\text{Att}(\mathbf{K}W_i^k, \mathbf{Q}W_i^q, \mathbf{V}W_i^v)$ and $\biguplus_{i=1}^h$ is the concatenation operation. Each query essentially asks for a different form of relevant information, allowing the attention model to incorporate more information into the computation of the context vector. Ergo, each head has the capability to learn and concentrate on distinct segments of the inputs, thus allowing the model to engage with a broader spectrum of information.

When used for processing feature vectors, the self-attention mechanism allows the model to summarize the information in the feature vectors that is important to the query. For instance, within the Natural Language Processing (NLP) domain, self-attention facilitates the extraction of inter-word relationships, such as the connections between verbs and their corresponding nouns or the associations between pronouns and the nouns they represent. For images, self-attention aids in identifying the interrelations between different regions of the image manifold. For multi-modal learning, self-attention can create links between different representations of the data.

3.2.2 Transformer

The original Transformer model, as proposed by Vaswani et al. [29], operates as a sequence-to-sequence model and encompasses an encoder and a decoder (shown in Figure 3.1), as commonly used in many Natural Machine Translation (NMT) models both of which are assembled from L identical layers. Each layer in the encoder primarily consists of a multi-head self-attention mechanism and a position-wise feed-forward network (FFN). To facilitate the construction of a deeper model, a residual connection [34], is integrated around each component, followed by the Layer Normalization [35]. Decoder layers, in contrast to encoder layers, incorporate cross-attention modules as well, situated between the multi-head self-attention mechanisms and the position-wise FFN. In cross-attention, the queries are projected from the outputs of the previous decoder layer, whereas the keys and values are projected using the outputs of the encoder. Additionally, the self-attention mechanisms within the decoder are modified to prevent each position from focusing on the positions that follow to create causal reasoning.

Position-wise FFN The position-wise FFN is a fully connected feed-forward module that operates separately and identically on each position. This can also be viewed as a convolutional layer with kernel size 1.

$$\text{FFN}(x) = \text{ReLU}(xW_1 + b_1)W_2 + b_2,$$

where x is the outputs of previous layer, and W_1, W_2, b_1, b_2 are trainable parameters.

Residual connection and Layer Normalization Proposed by He et al. [34], the residual connection is a shortcut connection that skips one or more layers. Formally, a residual block computes the output as $\text{FFN}(x) + x$, where $\text{FFN}(x)$ is the output of the neural network layers that are being skipped. The addition operation between $\text{FFN}(x)$ and x is element-wise. By doing so, residual connections alleviate the vanishing and exploding gradient problems, thus making it feasible to train much deeper models effectively.

Layer Normalization, introduced by [35], aims to mitigate the issue of internal covariate shift by normalizing the activations across a layer for each data point in a mini-batch. The normalization is carried out as follows,

$$\text{Norm}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}},$$

where μ and σ^2 are the mean and variance of the activations across the layer for each data point, and ϵ is a small constant for numerical stability.

In Transformers, both residual connections and layer normalization are extensively utilized to enhance the capabilities of the encoder and decoder structures, thereby facilitating the training of more robust and more complex models.

The Transformer architecture can be leveraged in three distinct manners:

- **Encoder–Decoder Configuration:** The complete Transformer architecture is employed. This is commonly utilized in sequence-to-sequence modeling tasks, such as neural machine translation, speech recognition, and video captioning. The main

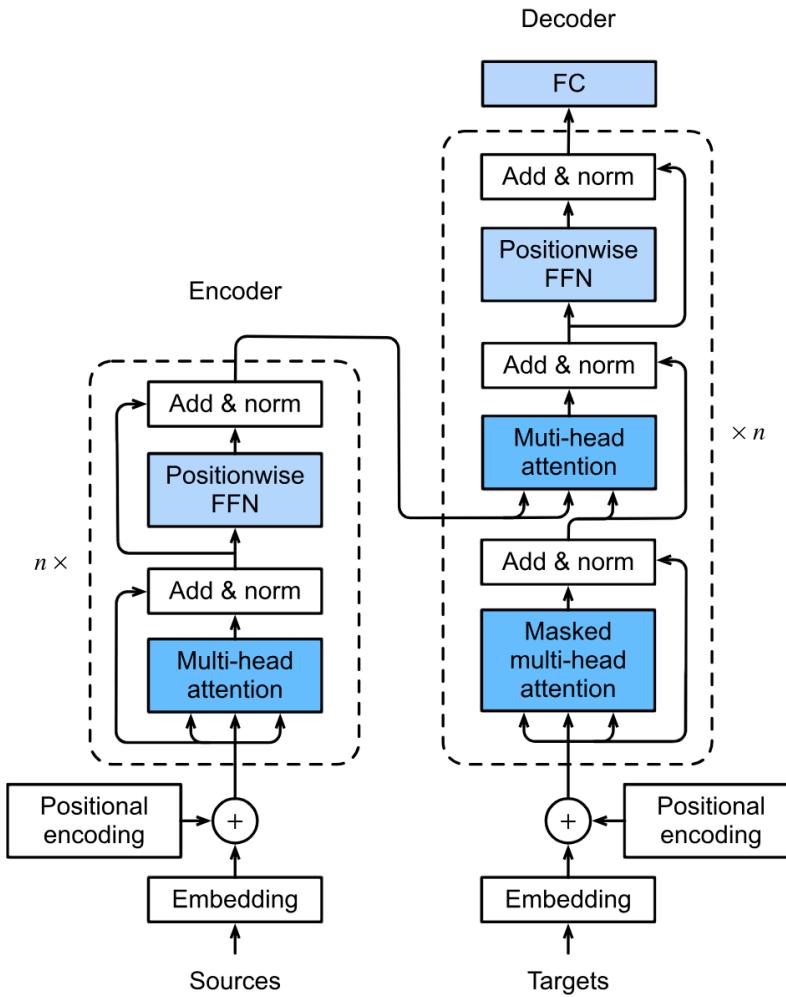


Figure 3.1: Transformer Encoder and Decoder Architecture, taken from [36]

inductive bias here is the assumption that the structure of the input and output sequences can be different and that a mapping between the two can be learned. This assumption comes into play in tasks like machine translation, where the length and structure of the source and target sentences can vary greatly. The self-attention mechanism in both the encoder and decoder allows the model to focus on different parts of the input sequence when generating each element of the output sequence. This configuration also assumes that information flow from any part of the input sequence to any part of the output sequence is possible. As the successor of Recurrent Neural Networks [37] and Long Short-Term Memory [38] architectures, Transformer models have several benefits such as parallel processing that prompts to performance and scalability increase and bidirectionality which allows understanding of ambiguous words and complex contexts.

- **Encoder-Only Configuration:** Only the encoder is utilized and the outputs of the

encoder serve as a representation of the input sequence, such as the Bert [39] family. BERT, Bidirectional Encoder Representations from Transformers, utilizes only the encoder part of the Transformers. In BERT, the input sequence is transformed into contextualized word embeddings. Unlike previous models that analyzed sentences from either left-to-right or right-to-left, BERT is able to analyze the context of a word in relation to all other words in the sentence by reading the input sequence in both directions. Bert utilizes a 2-stage training. During pre-training, it is trained on either of two self-supervised tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP). In the MLM task, some percentage of the input tokens are masked randomly, and the model must predict those masked tokens based only on their context. In the NSP task, the model learns to predict whether a sentence follows another sentence. This pre-training step allows BERT to learn a robust and general representation of language. Then, for fine-tuning on specific tasks, an additional output layer is added to the pre-trained BERT model, and all the parameters are fine-tuned on the task-specific data. This is frequently employed in Natural Language Understanding (NLU) tasks, such as text classification, sequence labeling, sentiment analysis, and named entity recognition.

- **Decoder-Only Configuration:** Only the decoder is used, with the encoder-decoder cross-attention module also being removed, such as the GPT [40] family of models. GPT, short for Generative Pretrained Transformer, employs a stack of Transformer decoders to generate a probability distribution over the vocabulary for the next token in a sequence, given all the previous tokens. Each token in the input sequence is processed in order autoregressively using masked attention, allowing each token to consider prior tokens in the same sequence. Importantly, this attention mechanism is “causal,” meaning that it only allows each token to attend to prior tokens in the sequence, preventing “future” tokens from influencing the output at the current position. This design enables GPT to generate coherent and contextually relevant text one token at a time. This is typically used for sequence generation tasks, such as language modeling, text generation, and music generation.

Compared with convolutional and recurrent networks, the Transformers carry a different inductive bias. Convolutional networks are recognized for their inductive biases of translation invariance and locality, facilitated through the use of shared local kernel functions. In a similar vein, Recurrent networks uphold the inductive biases of temporal invariance and locality via their Markovian structure [41]. In contrast, the Transformer architecture, without the positional encoding, makes minimal presumptions about the data, thereby establishing it as a versatile and universal model. However, this absence of structural bias can lead the vanilla Transformer to be susceptible to overfitting. The Transformer can also be perceived as a Graph Neural Network GNN [42] with message passing, designed over a fully connected graph, with each input serving as a node in the graph. A significant distinction between Transformers and GNNs lies in the fact that the former does not introduce any preconceived notions about the structure of the input data — the message-passing process in the Transformer is solely dictated by similarity measures over the content.

3.2.3 Problems with Transformer Models

While transformers have proven to be a powerful tool in many areas, they also have their limitations. Some of the key issues with transformer models include:

Temporal Information Loss: Research [43] has shown that despite their success in sequence modeling, transformers may not be the most ideal architecture for Long-Term Time Series Forecasting (TSF) problems. It was shown that self-attention mechanisms, even with positional encoding, despite their semantic correlation-capturing ability, can result in temporal information loss. When the transformers were evaluated on various datasets, they failed to capture the scale and bias of future data and had difficulty predicting trends on aperiodic data.

Limited Access to Higher Level Representations: In Transformers, a process of incremental abstraction is carried out, layer upon layer, to generate increasingly complex interpretations of the input sequence. This processing method involves treating the representations for the input sequence in a parallel manner across each layer. However, this parallel processing approach presents a drawback [44]. A significant feature that Transformers lacks is the utilization of previously computed top-tier representations to calculate the present representation. These top-tier representations refer to the most abstract and intricate interpretations of the input sequence, which have already been derived in the context of autoregressive models.

Complexity and Overfitting: Transformers often require larger training data sets to perform well due to their complexity. However, an experiment [45] showed that data set size was not a limiting factor for LTSF transformers, with models trained on a smaller training set performing marginally better. In another experiment [46], the authors discovered that the performance of the transformers only dropped slightly when the look-back window started at different time steps, suggesting that transformers may be overfitting to the provided data.

Destructive Bias from Improper Positional Embedding: Transformers also can exhibit a destructive bias when a proper positional embedding is not used [7]. Positional embeddings are crucial in transformers as they provide a sense of order/symmetry to the input data, which is inherently absent in the architecture due to its permutation equivariance [47]. However, the use of static positional embeddings can lead to limitations. For instance, these embeddings are fixed after training, regardless of the task or the word ordering system of the source or target language [48]. This can lead to a destructive bias, where the model fails to generalize well to unseen data. Furthermore, the lack of proper positional encoding can lead to inconsistencies in predictions under small shift perturbations, demonstrating a lack of shift-equivariance [49]. This destructive bias can significantly affect the model's performance, especially in tasks that heavily rely on the order or position of the input data.

In this thesis, we widely incorporate Transformer-based models and their inductive bias for both Event approximation (IEA-GAN) and point cloud generation (YonedaVAE). We also introduce tricks and methods to overcome these issues within this path.

3.3 Deep Generative Models and Simulation-Based Inference

Given a classification problem, the neural network is trained so that it can classify a proper class or condition y with a high probability $p(y|x)$, given the dataset x .

There are two main methods the model can utilize to reach decisions. One approach involves the explicit formulation of a classifier by modeling the conditional distribution $p(y|x)$. The problem is that $p(y|x)$ lacks the capability to deeply comprehend the data as it has no basis for understanding the uncertainties involved in decision-making. In other words, these models cannot simply acquire decision-making abilities without quantifying their beliefs about their environment in a probabilistic way [50].

Alternatively, one can opt for a joint distribution $p(x, y)$, that can be decomposed into $p(x, y) = p(y|x) p(x)$. To accomplish this, the estimation of the marginal distribution over objects, $p(x)$, becomes pivotal [50, 51]. The objective of “Deep Generative Models” (DGM) is to express and find the density of the data, $p(x)$, either implicitly or explicitly.

This chapter places an emphasis on latent Variable models as our formulated results, including PE-GAN, IEA-GAN, and YoendaVAE, all fall under this category of models. The fundamental concept of latent variable models involves postulating a latent manifold and the subsequent density estimation process:

$$z \sim p(z), \quad x \sim p(x|z).$$

In essence, the latent variables are associated with concealed factors within the data, and the conditional distribution $p(x|z)$ can be viewed as a **generator**. As a result, the joint distribution is factorized as $p(x, z) = p(x|z)p(z)$. However, since during training, one only has access to \mathbf{x} , the unknown \mathbf{z} should be marginalized out to get rid of all unobserved random variables. This leads us to the definition of the (marginal) likelihood function as follows:

$$p(x) = \int p(x|z)p(z)dz.$$

The pressing question then arises: *how can we compute this integral?* The integral over the latent manifold cannot be computed in practice efficiently, so we cannot directly evaluate the density of the observed data with a proper uncertainty evaluation. This means that we cannot directly find the maximum-likelihood estimators that best fit the given observations. The task of performing statistical inference when the data generating process does not have a tractable likelihood is known as **simulation-based inference** or **likelihood-free inference**. It is worth noting that the non-tractability amounts to not knowing the functional form of $p(x|z)$ but only being able to produce samples following the marginal distribution.

In the context of this thesis, there are two potential paths to tackle this issue with surrogate models as amortized simulators: Variational Autoencoders (VAEs) [52, 53] and Generative Adversarial Networks (GANs) [54].

In the case of VAEs, the tractability of inference is facilitated by using variational inference to approximate the posterior $p(z|x)$. This approach employs neural networks to parameterize the distributions, aiming to maximize the log-likelihood function, which measures the similarity between the data distribution and the model distribution in order to find a lower bound to the likelihood.

On the other hand, GANs operate based on a different principle. They utilize an adversarial loss function, where a discriminator, denoted as $D(\cdot)$, is used to differentiate between real data and synthetic data generated by the model. This generator operates in an implicit manner, defined as $p(x|z) = \delta(x - G(z))$, where $\delta(\cdot)$ represents the Dirac delta function. This adversarial mechanism encourages the generator to produce synthetic data that the discriminator cannot distinguish from real data, improving the model's ability to capture the data distribution. Although GANs do not learn the density directly, they do it implicitly, where one can indeed incorporate it for simulation-based inference [55].

3.3.1 Variational Autoencoders

Considering that we possess a latent-based model $p(x|z)$, with a prior $p(z)$ and a posterior $p(z|x)$, density estimation through maximum likelihood is intractable due to the integral, $p(x) = \int_z p(x|z)p(z)dz$. A naive approach would be to use the Monte Carlo approximation:

$$p(x) = \int p(x|z)p(z)dz$$

$$= E_{z \sim p(z)}[p(x|z)]$$

$$\approx \frac{1}{K} \sum_k p(x|z_k),$$

where $z_k \sim p(z)$ are samples from the latent prior. This method is simple and computationally feasible. However, if the manifold z is multi-dimensional, one encounters the curse of dimensionality immediately. This means that the number of samples needed to properly cover the space grows exponentially with respect to the dimension of the prior manifold. Alternatively, one can apply **variational inference** [56]. Variational inference reframes this challenge as an optimization problem by introducing an approximate version of the true, yet intractable, posterior which enables the creation of a tractable bound on $p(x)$. In other words, VAEs amortize the inference process.

Considering a family of known variational distributions parameterized by ϕ , represented as $\{q_\phi(z)\}_\phi$, we assume that they allocate non-zero probability mass to all latent priors, $z \in Z^M$. Subsequently, the logarithm of the marginal distribution can be approximated as follows:

$$\ln p(x) = \ln \int p(x|z)p(z)dz$$

$$\begin{aligned}
&= \ln \int \frac{p(x|z)p(z)}{q_\phi(z)} q_\phi(z) dz \\
&= \ln E_{z \sim q_\phi(z)} \left[\frac{p(x|z)p(z)}{q_\phi(z)} \right] \\
&\geq E_{z \sim q_\phi(z)} \left[\ln \frac{p(x|z)p(z)}{q_\phi(z)} \right] \\
&= E_{z \sim q_\phi(z)} [\ln p(x|z)] - E_{z \sim q_\phi(z)} [\ln \frac{q_\phi(z)}{p(z)}].
\end{aligned}$$

Here, the inequality arises from Jensen's inequality. Jensen's inequality states that the expectation of the convex function of the variable, $E[f(X)]$, is always greater than or equal to the function of the expectation of the variable, $f(E[X])$, $E[f(X)] \geq f(E[X])$.

By considering an amortized variational posterior, denoted as $q_\phi(z|x)$ instead of $q_\phi(z)$, we can rewrite our equation to:

$$\ln p(x) \geq \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p(x|z)] - \mathbb{E}_{z \sim q_\phi(z|x)} [\ln q_\phi(z|x) - \ln p(z)].$$

In Variational Bayesian methods, the lower bound of the log-likelihood function is known as the Evidence Lower BOund (ELBO). $p(z)$ is the marginal distribution or the prior. In amortization of the inference, one trains a singular model, like a neural network with specific weights, and it produces the parameters of a distribution for a given input. What emerges from this is a model similar to an Autoencoder, characterized by a probabilistic encoder, $q_\phi(z|x)$, and a probabilistic decoder, $p(x|z)$. An “Autoencoder” [57] is a neural network-based lossy compression model designed to learn an identity function in an unsupervised way to reconstruct the original input while compressing the data in a representation-efficient way. Thus, equipping it with the amortized variational posterior is a “Variational Auto-Encoder”. Two essential components form the ELBO: the first part, $\mathbb{E}_{z \sim q_\phi(z|x)} [\ln p(x|z)]$, is the negative reconstruction error, as x is encoded to z and then decoded back. The second part, $\mathbb{E}_{z \sim q_\phi(z|x)} [\ln q_\phi(z|x) - \ln p(z)]$, can be viewed as a regularization term. It is interesting that it looks suspiciously like a Kullback-Leibler (KL) divergence term.

From another point of view, in order to have a perfect reconstruction, one has to close the amortization gap, the gap between the ELBO and the true log-likelihood. In other words, the estimated posterior $q_\phi(z|x)$ should be very close to the intractable real posterior. One can use Kullback-Leibler divergence to quantify the distance between these two distributions. KL divergence $KL(X \parallel Y)$ measures how much information is lost if the distribution Y is used to represent X .

$$\begin{aligned}
\ln p(x) &= \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p(x)] \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\ln \frac{p(z|x)p(x)}{p(z|x)} \right] \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\ln \frac{p(x|z)p(z)}{p(z|x)} \right] \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\ln \frac{p(x|z)p(z)}{p(z|x)} \frac{q_\phi(z|x)}{q_\phi(z|x)} \right] \\
&= \mathbb{E}_{z \sim q_\phi(z|x)} \left[\ln p(x|z) - \ln \frac{q_\phi(z|x)}{p(z)} \right. \\
&\quad \left. + \ln \frac{q_\phi(z|x)}{p(z|x)} \right] \\
\implies \ln p(x) - \text{KL}[q_\phi(z|x) \parallel p(z|x)] &= \mathbb{E}_{z \sim q_\phi(z|x)} [\ln p(x|z)] \\
&\quad - \text{KL}[q_\phi(z|x) \parallel p(z)].
\end{aligned}$$

The term, $\text{KL}[q_\phi(z|x) \parallel p(z|x)]$, measures the discrepancy between the variational posterior and the real posterior, which is unknown. The LHS is exactly what we want to maximize when learning the true distributions.

Reparameterization Trick: The expectation term in the loss function invokes generating samples from the variational posterior, $q_\phi(z|x)$. The variational posterior typically assigns more probability mass to a smaller region than the prior. Nevertheless, the variational posteriors are almost deterministic, whereas the sampling process should be a stochastic process. A possible solution is the reparameterization trick [58, 59, 60]. With this trick, a random variable can be represented as a sequence of transformations of an independent random variable that has a simple distribution with known parameters. Incorporating the reparametrization of the Gaussian distribution can significantly decrease the variance of the gradient. This is because the source of randomness originates from the independent variable, and the gradient calculation is performed with respect to a deterministic function, not random entities.

3.3.1.1 Likelihood Estimation:

The log-likelihood is a critical stage in simulation-based inference. However, it's important to note that the ELBO is merely a lower bound for the log-likelihood and thus doesn't serve as an optimal approximation of it. There are several methods for likelihood estimation in VAEs, and we touch upon some of the most powerful ones here:

- **Importance Sampling:** As underscored in various studies [60, 61], an alternative approach is importance sampling that often suggested as a more effective means of

approximating the log-likelihood. Formally

$$\ln p(x) \approx \frac{1}{K} \sum_{k=1}^K \ln p(x|z_k)$$

where each z_k is sampled from the distribution $q_\phi(z_k|x)$. The key point to observe here is that the logarithm operation is applied outside the expected value. Given a sufficiently large sample space, the importance of weighting yields a better estimate of the log-likelihood.

- **Annealed Importance Sampling (AIS):** AIS is an advanced method of estimating the log-likelihood for VAEs [62, 63]. It gradually transforms and “anneals” the simple initial distribution into the complex target distribution through a series of intermediate distributions. This is done by introducing a temperature parameter β , varying from 0 to 1. At $\beta = 0$, the distribution is the prior $p(z)$ and at $\beta = 1$, it becomes the posterior $q_\phi(z|x)$. Formally, the unnormalized target distribution is defined as

$$\tilde{p}_\beta(z) = p(z)^{1-\beta} q_\phi(z|x)^\beta$$

AIS computes the importance weights along this temperature annealing path, and the log-likelihood can be approximated as

$$\ln p(x) \approx \frac{1}{K} \sum_{k=1}^K \ln \left[\frac{p(x, z_k)}{q_\phi(z_k|x)} \right]$$

where each z_k is sampled from the annealed distribution $\tilde{p}_\beta(z)$. This method provides a tighter estimate of the log-likelihood compared to the standard importance sampling, especially for complex data distributions.

- **Bidirectional Monte Carlo (BDMC):** BDMC [64, 65, 66] is another approach to improve the estimate of the log-likelihood in the context of VAEs. BDMC combines both forward and backward Markov chain Monte Carlo (MCMC) transitions for a more refined approximation of the log-likelihood. This method has been shown to provide more accurate log-likelihood estimates compared to methods like importance sampling or AIS, especially when the proposal distribution differs significantly from the target distribution. The idea is to correct the bias induced by the proposal distribution by considering both the forward and backward transitions. BDMC can be particularly beneficial in settings where the posterior distribution is multi-modal or when the variational approximation is not flexible enough. Formally, the log-likelihood estimate can be written as:

$$\ln p(x) \approx \frac{1}{K} \sum_{k=1}^K \ln \left[\frac{1}{2} \left(\frac{p(x|z_k^{(fwd)})}{q_\phi(z_k^{(fwd)}|x)} + \frac{p(x|z_k^{(bwd)})}{q_\phi(z_k^{(bwd)}|x)} \right) \right]$$

where each $z_k^{(fwd)}$ and $z_k^{(bwd)}$ is a sample from the forward and backward MCMC chains, respectively, initiated from the approximate posterior $q_\phi(z_k|x)$. As before, the logarithm operation is applied outside the expected value, and the average is taken over a large number of samples K . This approach provides a more balanced and accurate estimate of the log-likelihood when compared to unidirectional methods.

- **Nested Variational Inference (NVI):** NVI [67] is a unique approach that seeks to improve the estimation of the likelihood in the context of models with latent variables. The methodology introduces a sequence of forward and reverse densities, bridging the gap between an initial easy-to-sample density and the ultimate target density. This bidirectional approach aids in generating more sophisticated proposals for complex densities, deriving from simpler ones. NVI has demonstrated improved performance, particularly when the discrepancy between the proposal and target distributions is large. The approach becomes especially advantageous in scenarios where the posterior distribution is intricate or the variational approximation lacks sufficient flexibility. Formally, each intermediate joint density $\hat{p}(x, z_k)$ can be estimated using the self-normalized importance sampling method given by

$$\hat{p}(x, z_k) = \frac{w_k p(x, z_k)}{\hat{Z}}$$

where $w_k = p(x, z_k)/q_\phi(z_k|x)$, z_k are samples from the posterior $q_\phi(z|x)$, \hat{Z} is an estimate of the normalizer in the self-normalized importance sampling method, $\hat{Z} = \frac{1}{K} \sum_{k=1}^K w_k$, and K is the number of samples used. The log-likelihood is then approximated using weighted samples like in the AIS method. This framework offers a more refined and accurate estimate of the log-likelihood compared to the other methods.

3.3.1.2 Problems with VAEs:

Variational Autoencoders (VAEs) face a few challenges. A potential issue is the “posterior collapse” [68], which occurs when a powerful decoder treats the latent variable z as mere noise, leading to the regularization term being minimized for priors like the standard Gaussian. Another problem, known as the “hole problem” [69], arises from a mismatch between the aggregated posterior and the prior. If there are regions where the prior assigns high probability, but the aggregated posterior assigns low, sampling from these regions can result in low-quality output. In the next chapter (Chapter 4), we touch upon technologies that try to extend VAEs to approach these issues. Furthermore, in this study (see Chapter 6), with YonedaVAE, I introduce several tricks and methods to alleviate these issues.

3.3.2 Generative Adversarial Networks

Given the Monte Carlo approximation of the log-likelihood, one can turn the problem of calculating the integral into a problem of sampling from a known prior. In other words, one

needs only to model a parameterized $p_\theta(x|z)$ to generate samples. To gauge the difference between the generated samples $x \in p_g$ and real samples $x \in p_x$, an adversarial loss can be utilized to implicitly represent the empirical distribution. In such a setup, the generator in a single dimension behaves like a Dirac delta function, represented as $\delta(x - G(z))$. Consequently, the marginal distribution transpires as an infinite mixture of delta peaks in the observable space [70].

Given the generator G , a function $G : \mathbb{R}^d \rightarrow \mathbb{R}^n$, that maps a latent variable $z \in \mathbb{R}^d$ sampled from a distribution to the data manifold, $x \sim p(x|z) \in p_g$, and the discriminator D , a functional $D : \mathbb{R}^n \rightarrow [0, 1]$, that takes an image $x \sim p(x) \in p_x$ and assigns a probability to it, they are the players of the following two-player minimax game with value function $V(D, G)$ [54],

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \in \mathbb{R}^n} [\log D(x)] + \mathbb{E}_{z \in \mathbb{R}^d} [\log(1 - D(G(z)))].$$

For a fixed G , the objective for D can be reformulated as:

$$\begin{aligned} \max_D V(D, G) &= \mathbb{E}_{x \sim p_x} \left[\ln \frac{p(x)}{p(x) + p(x|z)} \right] + \mathbb{E}_{z \sim p_g} \left[\ln \frac{p(x|z)}{p(x) + p(x|z)} \right] \\ &= KL(p_d || \frac{1}{2}(p_d + p_g)) + KL(p_g || \frac{1}{2}(p_d + p_g)) + C. \end{aligned} \quad (3.3)$$

This loss is fundamentally linked to the Jensen-Shannon divergence, which measures the similarity between the generative distribution p_g and the data distribution p_d . Given ample capacity, the generator can implicitly reconstruct the data distribution.

Jensen-Shannon divergence has a particularly advantageous feature in that it behaves well when the magnitudes of both p_g and p_d are small. This attribute is an improvement over the asymmetric KL divergence employed in maximum likelihood models, which may not function well under these circumstances. However, it's worth mentioning that even the Jensen-Shannon divergence is not devoid of limitations. A notable instance arises when no probability mass is assigned to a data sample in a maximum likelihood model. While in such a case, the KL-divergence spikes to infinity, a GAN would experience no repercussions. This discrepancy highlights the unique properties and potential pitfalls of different divergence metrics in the context of generative models. As a result, a vast amount of research has been undertaken to improve its convergence and stability.

3.3.2.1 Problems with GANs

In general, training GANs is a highly brittle and painful task. It requires a significant amount of patience and hyperparameter tuning for domain-specific tasks.

Vanishing gradient: Should the discriminator reach an optimal state, the corresponding loss function diminishes to zero. Consequently, the gradients effectively vanish. This absence of a gradient halts the learning process, as the lack of an error signal hinders the ability to make adjustments based on the model's performance. The conundrum deepens when considering the opposite situation. If the discriminator performs poorly, the generator

lacks accurate feedback to guide its progress. In this scenario, the loss function fails to accurately mirror the reality of the data distribution. This double-edged sword scenario indeed poses a significant challenge to the successful training of GANs. The solution to this problem normally lies in balancing the complexity between the two models.

Mode collapse: A significant challenge that may arise during the training process of a GAN is a phenomenon known as 'Mode Collapse.' In this scenario, the generator model converges to a state where it consistently generates nearly identical outputs, despite the stochasticity of input noise vectors. While the generator, in this collapsed state, might succeed in deceiving the discriminator into believing that its outputs are real, it falls short in its primary goal: learning to accurately emulate the intricate and diverse $p(x)$. Formally speaking, if the generator's output is denoted by $G(z)$ for a random noise input z , the collapse can be seen when $G(z_1) \approx G(z_2)$ for $z_1 \neq z_2$. In this thesis, I introduce methods originating from Contrastive and Self-Supervised Learning as powerful solutions to prevent mode collapse.

Evading Nash equilibrium The training of a GAN is essentially a two-player non-cooperative game involving the generator and the discriminator, each aiming to optimize its own objective function. The goal of this game is to reach a Nash equilibrium, a state where no player can unilaterally improve their position by deviating from their current strategy while the other player's strategy remains fixed. GANs are typically trained using gradient descent techniques that are designed to find a low value of a cost function rather than to find the Nash equilibrium of a game [71]. However, this process does not explicitly account for the inherently interactive nature of the game being played. Each model updates its cost function independently, essentially ignoring the simultaneous adaptations of the other player. This strategy of concurrent gradient updates does not guarantee convergence to a Nash equilibrium. The models may continue to oscillate without settling into a stable state, or they may converge to a non-optimal solution. Several strategies, like Geometric GANs [72], Coulomb GANs [73], Energy-based GANs [74], moment matching GANs [75], f-GANs [76], contrastive GANs [77] have been proposed to address the problem of reaching Nash equilibrium in GAN training.

Low dimensional supports A manifold is a space that, at each point, locally is an Euclidean space. To be specific, if the dimension of this Euclidean space is denoted as n , we refer to the manifold as an n -manifold. For a given real-valued function, f , the "support" is the subset of the domain, n -manifold, that includes elements that are not mapped to zero. Arjovsky et al. [78] focuses on the issue of the supports of data being situated on low dimensional manifolds and the implications it has on the stability of GAN training. When the generator is tasked with creating a larger image based on a small dimensional random noise variable, z , it becomes challenging for it to entirely populate the high-dimensional space. The issue is further exacerbated when the intrinsic dimensions of the data are much lower than those of natural images, which can introduce biased information and lead to overfitting [79]. To address this, some approaches propose training the GAN using a low-dimensional representation of the dataset with the latent space of a pre-trained Autoencoder [80]. Other methods propose to let the prior match the embedding distribution rather than imposing the latent variables to fit the prior, which can help preserve the

geometric structure of the data manifold [81]. Despite these advances, the problem of low-dimensional supports in GANs remains a significant challenge in the field.

3.3.3 BigGAN and ContraGAN

BigGAN [82] is a model that has been trained at an unprecedented scale, with modifications introduced to improve both conditioning and scalability. These modifications have resulted in a model that sets a new state-of-the-art in class-conditional high-resolution image synthesis. In this subsection, we go through some of the most important methods they combined to reach such a high performance in conditional image generation.

Hinge Loss: BigGAN incorporates the hinge-loss variation of the adversarial loss [72],

$$\mathcal{L}_D^{\text{hinge}} = -\mathbb{E}_{x \in \mathbb{R}^n} [\min(0, -1 + D(x))] - \mathbb{E}_{z \in \mathbb{R}^d} [\min(0, -1 - D(G(z)))] , \quad (3.4)$$

$$\mathcal{L}_G^{\text{hinge}} = -\mathbb{E}_{z \in \mathbb{R}^d} [D(G(z))]. \quad (3.5)$$

The intuition behind the Hinge loss is to create a “margin” of separation in the feature space. It tries to ensure that positive and negative examples are not just correctly classified but are also separated by a wide margin. The Hinge loss function encourages the model to correctly classify examples and penalizes miss classifications. However, unlike other loss functions, such as cross-entropy loss, Hinge loss does not just care about correct classification. It also cares about the confidence of the classification.

For the discriminator, the Hinge loss tries to maximize the difference between the average output for real images and the average output for generated images. This encourages the discriminator to be confident in its decisions, which in turn provides stronger gradients for training the generator. On the other hand, for the generator, the Hinge loss tries to maximize the discriminator’s output for generated images. This encourages the generator to produce images that the discriminator is likely to classify as real.

Class Conditioning: For a Deep Generative Model (DGM) in order to capture and generate class-conditional samples, $p(x|z, y)$, many schemes for capturing the class conditions have been proposed since conditional GANs over input labels have been introduced [83]. The main idea is to minimize a specific measure between a class identification output of the discriminator and the actual labels after injecting an embedding of the conditional prior information into the generator. For example, ACGAN [84] tries to capture “data-to-class” relations by introducing an auxiliary classifier. However, BigGAN uses the “projection discrimination” (ProjGAN) mechanism [85] to generate conditional samples. ProjGAN tries to capture these data-to-class relations by projecting the class embeddings onto the output of the discriminator via an inner product that contributes to the adversarial loss. It is implemented by taking an inner product between the class embedding and the global mean-pooled feature map and then adding this to the output of the discriminator, as follows,

$$D(x, y) = D'(x) + e(y) \cdot D' D(x) \quad (3.6)$$

Where $D'()$ is the global mean-pooled feature map of the image x , $e(y)$ is the class label embedding. BigGAN also uses class-conditional BatchNorm [86] (CBN) to inject class labels into each of the subsequent layers. In standard conditional GANs, the class label information is typically incorporated into the model by simply concatenating the class label (or an embedding of the class label) with the noise vector that is input to the generator. However, this approach might not allow the class label information to sufficiently influence the generation process, especially in deeper layers of the generator. CBN addresses this issue by allowing the class label information to modulate the normalization parameters (i.e., the scale and shift parameters) of the BatchNorm layers in the generator. This means that the class label information can directly influence the activations of the neurons in each layer of the generator, allowing the model to more effectively condition the generation process on the class labels. In a general form, CBN can be formulated as follows

$$BN(x_i|y_i) = \gamma(y_i) \cdot \frac{x_i - \mu(X)}{\sqrt{\sigma^2(X) + \epsilon}} + \beta(y_i)$$

where, x_i is the input to the BatchNorm layer, y_i is the class label, $\mu(X)$ and $\sigma^2(X)$ are the mean and variance of the batch X , and $\gamma(y_i)$ and $\beta(y_i)$ are the scale and shift parameters, which are functions of the class label y_i .

Spectral Normalization: In order to stabilize the training, BigGAN employs Spectral Normalization [87] in all layers except the class label embedding layer for the generator. Spectral Normalization is a technique used to stabilize the training of the discriminator in GANs. It works by constraining the Lipschitz constant of a function, which helps to control the gradient norm and prevent the exploding gradient problem. The Spectral Normalization operation normalizes the weight matrix of a layer by the spectral norm, which is the largest singular value of the matrix. Formally, it can be written as

$$W_{SN} = \frac{W}{\sigma_1(W)} \quad (3.7)$$

Where, W is the weight matrix of a layer in G or D , $\sigma_1(W)$ is the spectral norm of the weight matrix, which is the largest singular value of W , and W_{SN} is the spectrally normalized weight matrix. This normalization operation ensures that the spectral norm (i.e., the largest singular value) of the weight matrix is 1, which helps to stabilize the training of the model.

Orthogonal Initialization: Orthogonal Initialization is a method used to initialize the weights of both G and D . Orthogonal Initialization has the property that the dot product of any pair of different rows or any pair of different columns is zero, and the dot product of a row with itself (or a column with itself) is 1. When the initial weight matrices are drawn from the orthogonal group, the width needed to guarantee efficient convergence of the model becomes independent of the depth of the layers [88]. This property can help to prevent vanishing and exploding gradients, which are common problems in the training GANs.

Skip-z connections: The generator is enhanced with this mechanism that directly links the input latent vector to the deeper layers within the network. So, instead of solely connecting to the initial layer, they introduced direct skip connections from the noise vector z to multiple layers within the generator. The underlying rationale for this approach is to enable the generator to directly manipulate features at varying resolutions and hierarchical levels using the latent space. The implementation of skip-z connections results in a moderate enhancement in performance.

Orthogonal Regularization: Orthogonal regularization [89] is a technique used to encourage the weights of the network to be orthogonal. This is done by adding a regularization term to the loss function that penalizes the network when its weights are not orthogonal. This can help to reduce overfitting and improve the generalization ability of the network, as it encourages the network to learn a set of diverse features that are not correlated with each other. Formally, the regularization term $R(W)$ is added to the generator's loss function during training

$$R(W) = \|W^T W - I\|_F^2 \quad (3.8)$$

Where, W is the weight matrix of a layer in the neural network, I is the identity matrix of the same size as $W^T W$, $\|\cdot\|_F$ denotes the Frobenius norm (which is a measure of the total magnitude of all the elements in the matrix), $R(W)$ is the orthogonal regularization term.

Architecture: Self-Attention non-local Blocks. The self-attention block used in BigGAN is borrowed from the Self-Attention GAN (SAGAN) [90], depicted in Figure 3.2, allows the model to focus on global relations and long-range dependencies in the image manifold. It works by calculating the attention score for each pair of positions in the input. The attention mechanism computes a weighted sum of features, where the weight assigned to each feature is determined by the attention score, as follows

$$o_i = \sum_j \frac{\exp(f_i^T g_j)}{\sum_k \exp(f_k^T g_k)} h_j.$$

Where o_i is the output of the self-attention block for position i . f_i and g_j are the feature vectors at positions i and j , respectively. These vectors are obtained by applying linear transformations (1×1 convolutions) to the input feature vectors. h_j is another transformed version of the input features at position j . The term $\exp(f_i^T g_j)$ computes the attention score. The softmax function, represented by $\frac{\exp(f_i^T g_j)}{\sum_k \exp(f_k^T g_k)}$, normalizes these scores across all positions k , ensuring the alignment.

Since traditional convolutional layers in a neural network are local and translation invariant, meaning they perform the same computation for every region of the input. However, they have a limited “field of view” (determined by the kernel size), and therefore, the long-range dependencies could be washed out over high-resolution images. The self-attention mechanism, on the other hand, allows the model to consider all positions in the input at once and learn to weigh their influence accordingly. This is particularly

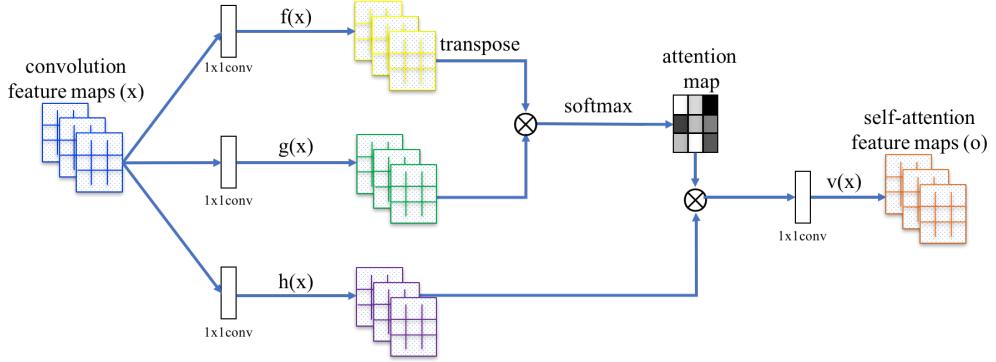


Figure 3.2: SAGAN architecture, taken from [90]

useful in scenarios where the relevant information is spread out and not confined to local neighborhoods, such as in images where the context of a pixel can be informed by distant pixels.

Architecture: BigGAN. The BigGAN model, shown in Figure 3.3, employs the ResNet GAN architecture [90, 87], but with a modified channel pattern in the discriminator. The first convolutional layer in each block of the discriminator has the same number of filters as the output filters. The model uses a shared class embedding in the generator and skip connections for the latent vector. The latent vector is divided into equal-sized chunks, each of which is combined with the shared class embedding and passed to a corresponding residual block. Each block’s conditioning is linearly projected to create per-sample gains and biases for the CBN layers.

Architecture: BigGAN-deep. The BigGAN-deep model differs from the original BigGAN in its structure and complexity. In the generator, It employs simplified skip-z conditioning where the entire z is concatenated over the class embedding and passed to each residual block in the CBN blocks. The model uses bottleneck residual blocks [34], which include two extra 1×1 convolutions to manage the number of channels before and after the 3×3 convolutions. In the generator, for channel reduction, they retain the first group of channels and drop the rest to produce the required number of channels. However, in the discriminator, they pass the input channels unperturbed and concatenate them with the remaining channels produced by a 1×1 convolution. At each resolution, there exist two blocks, whereas BigGAN uses one, making BigGAN-deep four times deeper than BigGAN. Despite this, BigGAN-deep has fewer parameters due to the bottleneck structure in its residual blocks.

With the right blend of these technologies and appropriate hyperparameter optimization, BigGAN and BigGAN-deep have achieved state-of-the-art results for high-resolution conditional image generation across numerous natural image datasets. However, these models have their shortcomings. They tend to exhibit low stability during the training and lack compatibility with fine-grained datasets. These datasets require the ability to distinguish between visually similar objects from subordinate categories, where there are high inter-class similarities and low intra-class similarities.

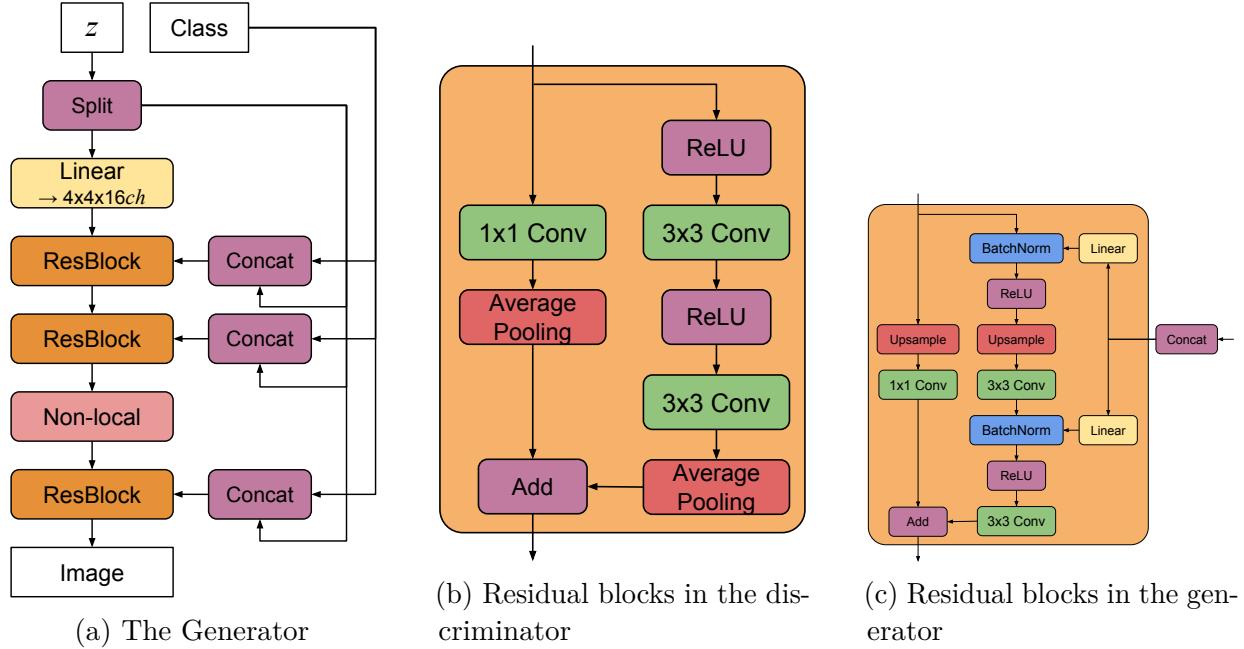


Figure 3.3: BigGAN's main building blocks, taken from [82]

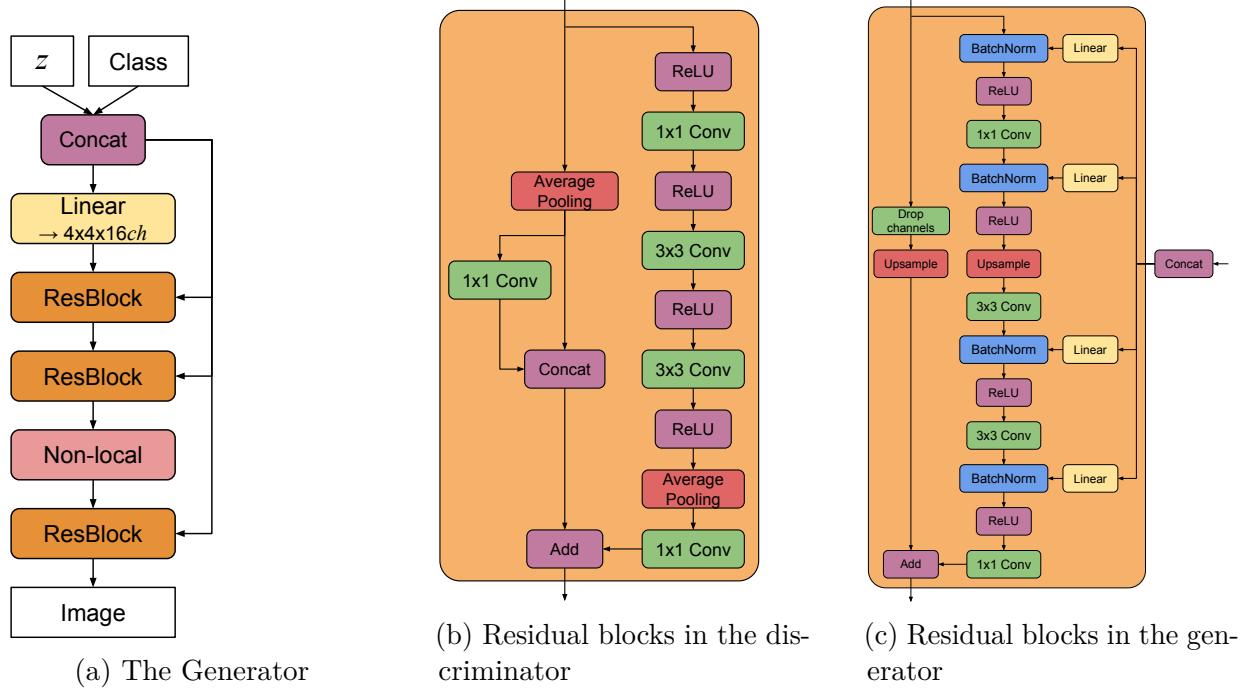


Figure 3.4: BigGAN-deep's main building blocks, taken from [82]

3.3.3.1 ContraGAN

BigGANs with a projection discriminator suffer from overfitting issues, which can lead to the collapse of adversarial training, especially when the number of classes increases [77].

ContraGAN [77] alleviates this overfitting problem with its conditional contrastive loss (2C loss). Incorporated from deep metric learning in order to seize “data-to-data” relations or “intra-class” relations, they introduce the 2C loss, derived from NT-Xent loss [91],

$$\ell_{2C}(x_i, y_i) = -\log\left(\frac{\exp(S_c(h(x_i)^\top e(y_i))) + \sum_{k=1}^m \mathbf{1}_{k=i} \exp(S_c(h(x_i)^\top h(x_k)))}{\exp(S_c(h(x_i)^\top e(y_i))) + \sum_{k=1}^m \mathbf{1}_{k \neq i} \exp(S_c(h(x_i)^\top h(x_k)))}\right). \quad (3.9)$$

Here, x_i are the images, y_i are the corresponding labels, $S_c(.,.)$ is a similarity metric, and $h(.)$ is the output of the image embeddings. Although ContraGAN benefits from this loss by capturing the intra-class characteristics among the images that belong to the same class, it is prone to class-confusion [92, 93] as different classes could also show similarity among themselves since their vector representation in the embedding space might not be orthogonal to each other, which is precisely what we are dealing with in a fine-grained dataset.

In this study, I try to address this issue by introducing an *Intra-Event Aware mechanism* that not only captures “intra-class” relations but “inter-class” correspondence. This is, in particular, very important in Particle Physics and Detector Simulation, where the signatures of the detector can be grouped together for each collision event as they originate from the same readout window.

3.3.4 Point Cloud Generation

We define a point cloud by a triplet $P = (V, X, Y)$, where V is the set of points, $X \in \mathbb{R}^{N \times D}$ is the feature matrix that stores the features (usually the coordinates) associated with each point, and $Y \in \mathbb{R}^{N \times F}$ is an optional attribute matrix that stores additional attributes associated with each point. D is the dimensionality of the point coordinates (usually 2 or 3), and F is the dimensionality of the optional attributes. Given a set of M observed point clouds $P = \{P_i\}_{i=1}^M$, a point cloud generative model learns the distribution of these point clouds $p(P)$, from which new point clouds can be sampled $P_{\text{new}} \sim p(P)$.

For point clouds as inherently unordered collections of points akin to set objects, exchangeability is often viewed as a pivotal characteristic of independent sampling. Let’s dive into this deeper.

Point clouds and sets (multi-sets), being unordered, exhibit a fundamental symmetry to the group of permutations, denoted as $S = \bigcup_{n \in \mathbb{N}^*} S_n$, encapsulating all possible permutations. A specific permutation $\pi \in S_n$ can operate on an $n \times n$ matrix A by reordering its rows and columns, which we denote as $\pi.A = \pi A \pi^T$. When dealing with a $n \times c$ matrix, the permutation alters the order of rows ($\pi.X = \pi X$), and leaves a vector $z \in \mathbb{R}^h$ untouched ($\pi.z = z$). This inherent symmetry serves as an irrelevant factor of variation in the data, which should be accounted for and removed in the latent space representation (that is, $\pi.z = z$).

Discriminative models account for these symmetries when a neural network f exhibits equivariance to the action of a group, $\pi.f(X) = f(\pi.X)$. However, when the input to f is a vector, imposing $\pi.f(z) = f(\pi.z) = f(z)$ only allows solutions where all rows are identical,

which is overly limiting. To circumvent this constraint, Vignac et al. [94] introduce a definition called (F, l) -equivariance, which extends the standard notion of equivariance and affords a more flexible condition in generative settings. In discriminative models, this is typically achieved by using an equivariant model with an invariant loss function. Formally, if $F_\Theta = \{f_\theta : X \rightarrow Y; \theta \in \Theta\}$ represents a class of hypothesis comprising G -equivariant functions mapping from X to Y (a neural network architecture parameterized by θ serves as an instance), then the loss function l must meet the following requirement:

For all f in F , for all g in G , and for all (X, Y) in $X \times Y$,

$$l(g.f(X), g.Y) = l(f(X), Y)$$

while the gradients with respect to the parameters satisfy,

$$\nabla_\theta l(g.f(X), g.Y) = \nabla_\theta l(f(X), Y).$$

This highlights that every adjustment to the parameters is independent of the group elements utilized in representing X and Y . As a consequence, the overall training dynamics become indifferent to the group elements used in data representation, thus eliminating the necessity for data augmentation. As a result, Vignac et al. [94], proposed the following definition for equivariance:

Definition 1: $((F, l)$ - equivariance). Consider a hypothesis class $F_\Theta \subset Y^X$, a group G that acts on X and Y , and a loss function l defined on Y . The pair (F_Θ, l) is equivariant to the action of G if the dynamics of $\theta \in \Theta$ trained with gradient descent on l do not depend on the group elements that are used to represent the training data,

$$\forall \theta \in \Theta, \forall g \in G, \forall (X, Y) \in X \times Y, \nabla_\theta l(f_\theta(g.X), g.Y) = \nabla_\theta l(f_\theta(X), Y).$$

Inherently, using an equivariant architecture in conjunction with an invariant loss is sufficient to achieve (F, l) -equivariance in discriminative scenarios. For typical generative architectures for sets and point clouds, they propose the following sufficient conditions:

Lemma 1. Sufficient conditions for (F, l) - equivariance:

- **GANs:** Given F as a GAN architecture featuring a permutation-invariant discriminator, and l as the standard GAN loss, the pair (F, l) is permutation equivariant. There are no restrictions imposed on the generator.

Proof: given a set generator f , a discriminator function d , and X_1, \dots, X_m as a training dataset, the standard loss function for GANs is formulated as

$$\frac{1}{m} l(f, d, X_1, \dots, X_m) = \sum_{i=1}^m \log(d(X_i)) + \mathbb{E}_Z[\log(1 - d(f(z)))].$$

In order to obtain $l(f, d, X_1, \dots, X_m) = l(f, d, \pi_1.X_1, \dots, \pi_m.X_m)$ for every choice of π_i , it is therefore sufficient to choose a permutation invariant discriminator.

- **VAEs:** Given F as an encoder-decoder architecture with a permutation-invariant encoder, and the reconstruction loss \hat{l} satisfies $\forall \pi \in S, \hat{l}(\pi.X, X) = \hat{l}(X, X)$, then the pair (F, l) is permutation equivariant. The decoder function is exempted from any constraints.

Proof: Assuming that the Autoencoder is comprised of a permutation invariant encoder enc and an arbitrary decoder f . For any set size n , set $X \in \mathbb{R}^{n \times d}$ and permutation $\pi \in S_n$, we have

$$\begin{aligned} l(\pi.X, \hat{X}) &= l(X, \hat{X}) \\ \Rightarrow l(\pi.X, f(enc(X))) &= l(X, f(enc(X))) \\ \Rightarrow l(\pi.X, f(enc(\pi.X))) &= l(X, f(enc(X))) \quad (\text{enc is invariant}) \\ \Rightarrow \nabla_{\theta} l(\pi.X, f(enc(\pi.X))) &= \nabla_{\theta} l(X, f(enc(X))) \end{aligned}$$

Within the context of generative models for point cloud generation, the aspired (F, l) - equivariance doesn't demand $\pi.f(z) = f(\pi.z)$. The stipulations offered by Lemma 1 are in accord with all point cloud generative models, and the common permutation invariant loss functions like the Chamfer loss, Wasserstein-2 distance, and Hungarian loss meet the requirements for the VAE loss. In the context of GANs and VAEs, the exchangeability condition is absent. A subtle point is that the model output in GANs and VAEs involves an implied mat-to-set conversion. Hence, a model creating uniformly permuted matrices equates to one producing exchangeable matrices. Thus, the output, being a set, not a matrix, is a presumption, not a proven fact. This insight explains why independent sampling doesn't surpass non-exchangeable set creation techniques like MLPs and First-n (see Chapter 6), leading us to incorporate a new creation mechanism, *Adaptive Top-q Sampling*, uncoupled from the constraint of model exchangeability, in Chapter 6.

3.3.4.1 Permutation Invariant loss

Thus far, we discussed the concept of (F, l) - equivariance and how it affects the inherent inductive bias in the generator (decoder) and the discriminator (encoder) model. Now, we discuss some commonly used permutation invariant assignment-based loss functions, such as the Chamfer loss [95] and the Hungarian loss [96]. Given the true set $Y = \{y_1, \dots, y_l\}$ and the predicted set, $\hat{Y} = \{\hat{y}_1, \dots, \hat{y}_k\}$, the assignment-based loss is as follows

$$L_A(\hat{Y}, Y) = \sum_i d(\hat{y}_i, y_{\pi(i)}) + \sum_j d(\hat{y}_{\sigma(j)}, y_j)$$

where d is a distance metric such as a Frobenius norm, and $\pi : \{k\} \rightarrow \{l\}$, $\sigma : \{k\} \rightarrow \{l\}$ are assignment functions that map indices from one set to another. The inherent lack of order in sets and point clouds brings about the challenge of deciding which elements should be compared as a combinatorial optimization problem. In the most generalized version, the problem can be depicted as follows: The instance of the problem involves a number of

generated points in a set and a number of truth-level points. Any generated set can be matched with any truth level set, with the cost of such a match possibly varying based on the specific generated-truth level set pairings. The aim is to align as many generated sets with truth level sets as possible, adhering to the condition that no more than one generated set is matched to a single truth level set and vice versa, with the intention of minimizing the overall cost of these matching. In graph theory language [97], the assignment problem consists of finding, in a weighted bipartite graph, a matching of a given size, in which the sum of weights of the edges is minimum. The variation in assignment-based set losses primarily lies in the strategies for assignment, which are represented by the choices for π and σ .

The Chamfer loss is one such example that solves an unbalanced assignment where each element in \hat{Y} is mapped to the nearest element in Y and vice versa. It is defined by $\pi(i) = \text{argmin}_j d(\hat{y}_i, y_j)$ and $\sigma(j) = \text{argmin}_i d(\hat{y}_i, y_j)$, resulting in:

$$L_C(\hat{Y}, Y) = \sum_i \min_j d(\hat{y}_i, y_j) + \sum_j \min_i d(\hat{y}_i, y_j)$$

For the Hungarian loss, π and σ create a bijection, thus necessitating identical set sizes $n = m$ to solve a balanced assignment problem,

$$L_H(\hat{Y}, Y) = \frac{1}{2} \left[\min_{\pi \in S_k} \sum_i d(\hat{y}_i, y_{\pi(i)}) + \min_{\sigma \in S_k} \sum_i d(\hat{y}_{\sigma(i)}, y_i) \right] = \min_{\pi \in S_k} \sum_i d(\hat{y}_i, y_{\pi(i)}) \quad (3.10)$$

where S_k is the collection of all permutations on sets of size k . These varying strategies of assignment bring about different metric spaces on sets. Both Chamfer and Hungarian losses present unique strengths and weaknesses. The Chamfer loss scales with the set sizes k and l at a computational cost of $O(kl)$, while the Hungarian loss is substantially more costly with a complexity of $O(k^3)$ [98, 99]. The Chamfer loss's lack of one-to-one assignments can be a disadvantage when comparing multi-sets or sets with numerous similar elements up to numerical precision. On the contrary, the stringent requirement for bijective assignments in the Hungarian loss becomes a disadvantage when comparing sets of different sizes in unbalanced assignment problems. Later, in the development of YonedaVAE, we will use the Hungarian loss as the reconstruction loss.

3.4 Self-Supervised Representation Learning

Self-supervised learning (SSL), dubbed “the dark matter of intelligence” [100], allows for the utilization of naturally occurring labels within data, harnessing the the abundant availability of unlabeled data. The process involves carefully crafting learning objectives to enable data self-supervision. The tasks within self-supervised learning, often referred to as pretext tasks, lead to a supervised loss function. The focus, however, isn't on the performance of these generated tasks but on the intermediate representation they develop. This representation is expected to hold semantic or structural significance, beneficial for multiple real-world

applications. For instance, randomly rotating images and training a model to predict these rotations serve as a pretext task. While the precision of this task isn't crucial, it assists the model in deriving high-quality latent variables for practical tasks, like developing an object recognition classifier with minimal labeled samples. From a representation learning perspective, it is possible to classify all generative models as falling under the umbrella of self-supervised learning, albeit with varied objectives. A common thread linking them is the concept of **contrastive learning**

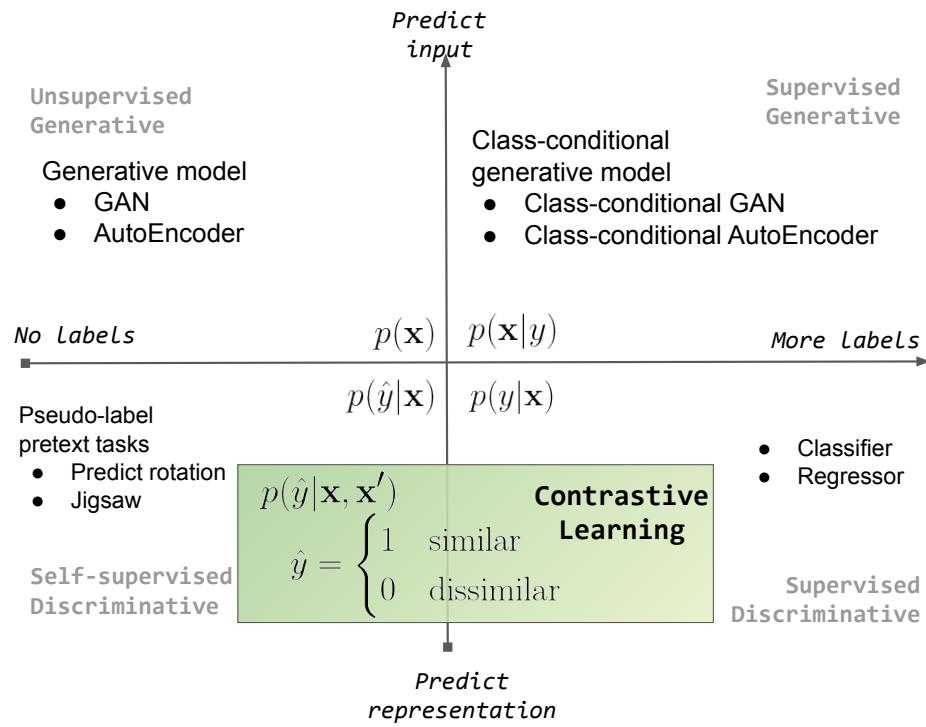


Figure 3.5: Contrastive learning at the interface of supervised and self-supervised learning, adopted from [101]

While distinct in their methods, self-supervised learning and contrastive representation learning are closely intertwined (shown in Figure 3.5), with both aiming to leverage the inherent structures or patterns in data for learning representations. Contrastive learning, often employed as a method within the self-supervised learning framework, operates on the principle of learning by comparison, distinguishing between similar and dissimilar data pairs. It complements self-supervised learning, which focuses on leveraging the inherent labels in the data, by further enhancing the data representation through comparison-based learning. The powerful combination of these two learning paradigms allows for the development of rich and meaningful representations without relying heavily on manually annotated data, thus maximizing the utility of vast amounts of unlabeled data.

Now, let's explore different three main families within the realm of SSL, each possessing

unique features and techniques for handling data and deriving valuable insights.

3.4.1 The Deep Metric Learning (Contrastive) Family

The Deep Metric Learning (DML) family of methods is based on the principle of encouraging similarity between semantically transformed versions of an input. DML originated with the idea of a contrastive loss, which allows the model to learn a discriminative embedding space (metric) that both maximizes inter-class distance and minimizes intra-class distance (see Figure 3.6).

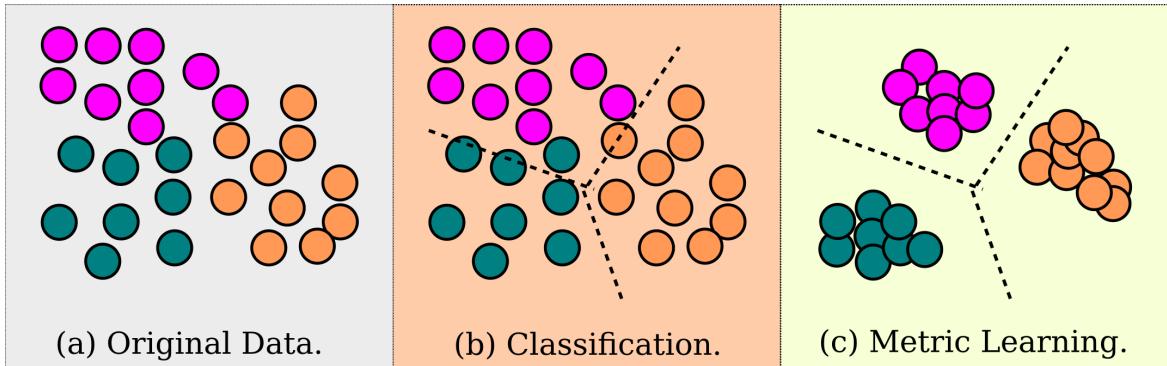


Figure 3.6: Deep Metric Learning vs. vanilla Classification

Classification in machine learning involves assigning predefined labels to instances based on learned patterns, focusing on differentiating classes. DML, however, is concerned with learning a distance function to measure the similarity or dissimilarity between instances, aiming to bring similar instances (positive pairs) closer and push dissimilar (negative pairs) ones apart in the representation space. The Contrastive loss [102, 103] was first introduced as the earliest training objective used for DML,

$$L_{\text{cont}}(\mathbf{Z}) = \sum_{(i,j) \in \mathbf{P}} \|z_j - z_i\|^2 + \sum_{(i,j) \notin \mathbf{P}} \text{ReLU} \left(m - \|z_i - z_j\|^2 \right)^2, \quad m > 0,$$

where $z_i \in \mathbf{Z}$ denotes the model representation of sample i , \mathbf{P} denotes the set of positive samples, and m is the margin hyperparameter, defining the lower bound distance between samples of different classes. The contrastive loss takes a pair of inputs and minimizes the embedding distance when they are from the same class but maximizes the distance otherwise. Improving the contrastive loss by focusing on relative distances rather than absolute distances, Triplet loss [104, 105] considered the relationships between three samples, an anchor, a positive sample (similar to the anchor), and a negative sample (dissimilar to the anchor), at a time instead of just two, leading to better generalization and improved performance.

$$L_{\text{triplet}}(Z) = \sum_{(i,j) \in \mathbf{P}} \sum_{(k,l) \notin \mathbf{P} | k=i} \text{ReLU} \left(\|z_i - z_j\|^2 - \|z_i - z_k\|^2 + m \right), \quad m > 0.$$

After the Triplet loss, many DML methods were introduced, but the transition from DML to the modern SSL can be traced back to [106, 107] presented the Contrastive Predictive Coding (CPC) loss. The innovative approach of using alternate positive sample views as the negative view for other pairs along with the incorporation of the Information Noise Contrastive Estimation (InfoNCE) loss were pivotal elements of this transition. Formally, the InfoNCE loss can be expressed as:

$$L_{\text{InfoNCE}} = - \sum_{(i,j) \in \mathbf{P}} \log \left(\frac{e^{\text{Sim}(z_i, z_j)/\tau}}{\sum_{k=1}^N e^{\text{Sim}(z_i, z_k)/\tau}} \right),$$

The innovation of the InfoNCE loss lies in its design. Instead of calculating the similarity with one negative sample at a time, InfoNCE loss considers multiple negative samples simultaneously, transforming the learning objective into a softmax-based, multi-class classification problem. This method introduces strong competition among the negative samples, which leads to a more balanced and effective gradient during the learning process. Additionally, by normalizing the similarities between all the negative samples and the anchor, InfoNCE loss ensures that the model's focus is not only on the hardest negatives but also on easy and moderate ones, thereby providing a more comprehensive understanding of the representation space. Also, InfoNCE loss is closely related to the mutual information between the representations of positive examples, which provides a solid theoretical foundation for the approach.

Among the most notable methodologies arising from this shift towards SSL within the domain of deep learning is SimCLR [91]. SimCLR, depicted in Figure 3.7, learns visual representations by formulating a correspondence between two altered views of the same image with contrastive learning. These views are derived by a series of transformations, which consist of random resizing, cropping, color distortion, and arbitrary blurring. Upon encoding every view, SimCLR utilizes a projector, typically an MLP, to relocate the initial embeddings into a compact manifold, a hypersphere, to prevent dimensional collapse in the representation space [108]. In this space, the contrastive loss is applied to promote congruity between the views. To maximize agreement between the representations, SimCLR modifies the InfoNCE loss by removing the i 'th from the denominator to introduce the NT-Xent loss,

$$L_{\text{NT-Xent}}(\mathbf{Z}) = - \sum_{(i,j) \in \mathbf{P}} \log \left(\frac{\exp(\text{Sim}(z_i, z_j))}{\sum_{k=1, k \neq i}^N \exp(\text{Sim}(z_i, z_k))} \right).$$

The negative sign is because it wants to maximize the similarities of positive pairs, which is equivalent to minimizing the negative log of this ratio. As a training challenge, SimCLR requires a large batch size and heavy augmentations to incorporate enough negative samples to achieve good performance.

The significant shift from DML to Contrastive SSL is rooted in several pivotal modifications. These include the adoption of data augmentation to procure positive/negative pairs instead of sampling, the utilization of deeper network structures with possible sources

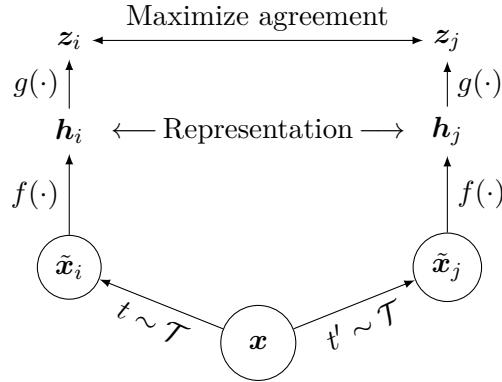


Figure 3.7: The framework for SimCLR. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two related views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a variant of InfoNCE loss, NT-Xent loss.

of noise, and the employment of a projector network. Next, we describe an alternative to deep metric learning based on self-distillation.

3.4.2 The Self-Distillation Family

Knowledge distillation [109] is the procedure of transferring knowledge from a large model (the target or teacher) to a more compact one (the online or student), often being used in the context of model compression. When both models have the same architecture, this procedure is called **self-distillation** [110] or the dark-knowledge [110]. Various forms of knowledge transfer can be broadly categorized into three types: Response-based knowledge, Feature-based knowledge, and Relation-based knowledge.

- **Response-Based Knowledge:** This type of knowledge transfer depends heavily on the output of the teacher model. Specifically, the teacher's predictions (also called 'soft targets') are used to guide the learning of the student. This is often done by using the teacher's class probabilities as targets in training the student, which can provide more information than the original hard labels, especially when the teacher model is confident about certain classes. It is the most direct form of knowledge transfer in distillation, and it is at the core of the original knowledge distillation method proposed by Hinton et al. [109].
- **Feature-Based Knowledge:** Feature-based knowledge refers to the intermediate activations (features) produced by the target model. In this case, the student model is trained to mimic the internal representations of the teacher model, with the aim of capturing the teacher's way of processing and representing the data. This approach [111] can be beneficial in cases where the soft targets do not provide enough guidance for the student, and the student can learn more by mimicking the teacher's feature maps.

- **Relation-Based Knowledge:** Relation-based knowledge takes into account the correspondence among different data points or features within the teacher model. It may include the pairwise similarities or relative rankings among different data points in the feature space or the correlations between different feature channels. By learning such relational knowledge [112], the student model may learn to capture more abstract and high-level information, which can be crucial for complex tasks.

Self-distillation techniques, such as BYOL (Bootstrap Your Own Latent) [113], SimSiam [114], and DINO (DIstillation of knowledge using NO labels)[115], function based on a straightforward yet effective mechanism. This mechanism involves presenting two distinct views of the same data point to a pair of encoder networks and subsequently using a predictor network to map the output of one encoder to that of the other. This process facilitates the transfer of learning from one view to the other, thus achieving self-distillation.

The self-distillation mechanism employed by these models is a significant contribution towards addressing one of the central challenges in SSL: **model collapse**. Model collapse, a situation where a model generates trivial or meaningless representations of data, often results in poor model performance. This issue is especially prevalent in contrastive representation learning methods, where the model’s task is to distinguish between different instances of data. A model can fall into the trap of model collapse if there is a shortage of negative samples or if the learning dynamics do not have enough constraints to prevent the model from collapsing toward these trivial solutions. However, self-distillation methods, like BYOL, SimSiam, and DINO, have been developed to circumvent this problem by introducing an asymmetry in the learning process. As such, they effectively counter model collapse by preserving the diversity in the representation space and encouraging meaningful learning from the input data. We will briefly review them in the following.

BYOL, or Bootstrap Your Own Latent [113], depicted in Figure 3.8, is a novel approach that initially deployed the concept of self-distillation with a mechanism to prevent model collapse. This method employs a pair of neural networks along with a predictor tasked with the duty of aligning the outputs from one network onto the other. The model operates by feeding each network with a uniquely transformed view of the same image, using transformations that include operations like random resizing, cropping, and alterations to color and brightness. The student network is dynamically updated during the training process through gradient descent. On the other hand, the teacher network is updated using the Exponential Moving Average (EMA) of the weights of the online network. The slow update pace, which is a consequence of using EMA, introduces a crucial asymmetry that significantly contributes to the successful performance of BYOL. The loss function for BYOL can be expressed as

$$L_{\text{BYOL}}(\theta) = \mathbb{E}_{(x, t, t') \sim (X, T, T')} \|\text{renorm}(q_\theta(g_\theta(t(x)))) - \text{renorm}(g_\xi(t'(x)))\|_2^2,$$

where the two vectors in the representation space are L2-normalized automatically. This normalization can be represented as:

$$\text{renorm}(v) = \frac{v}{\max(\|v\|_2 + \epsilon,)}.$$

g_θ is the online (student) encoder network, parameterized by θ , while q_θ is the predictor network parameterized by θ . Here, $x \sim X$ is the input drawn from the data distribution X , and $t(x)$, $t'(x)$ are two augmented views of x where $t \sim T$, $t' \sim T'$ are two data augmentations. The target network g_ξ shares the same architecture as the student and is only updated via EMA with τ controlling the extent to which the target network retains its history, as shown:

$$\xi \leftarrow \tau\theta + (1 - \tau)\theta.$$

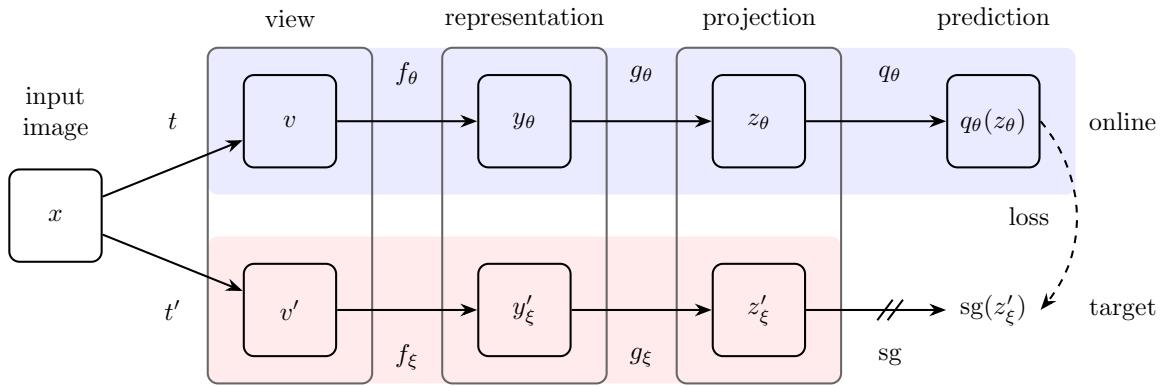


Figure 3.8: BYOL's overall architecture, adopted from [113]

SimSiam: SimSiam [114] can be thought of as a simplified version of the BYOL framework, as depicted in Figure 3.9. Through their exploration, they found that the use of EMA was not strictly necessary, even though its inclusion slightly improved performance. This discovery allowed for the formulation of a more streamlined loss function. The loss function for SimSiam is defined as follows:

$$L_{\text{SimSiam}}(\theta) = \mathbb{E}_{(x, t, t') \sim (X, T, T')} \left[\|\text{renorm}(q_\theta(f_\theta(t(x)))) - \text{sg}(\text{renorm}(f_\theta(t'(x))))\|_2^2 \right],$$

where $\text{sg}(\cdot)$ is the stop-gradient operation. The reason why BYOL and SimSiam evade model collapse and are vital to the success of these methods is the asymmetry introduced between the two network branches, along with certain dynamics of the training process, which implicitly regularizes the variance of the embeddings. It is this dynamic process and the careful management of variance that ensure the maintenance of a diverse and meaningful representation space. It has been observed [116, 117, 118] that the performance of these self-supervised learning methods tends to degenerate to the level of random guessing when the component of batch normalization is eliminated. Batch normalization indirectly institutes a form of contrastive learning, thereby playing a crucial role in these methods. The dependency on negative samples that arises through batch normalization is a significant element in preventing model collapse. For instance, if we were to represent every data

point as an all-zero vector, it would trigger a model collapse. However, the process of batch normalization inherently redistributes values, even if the batch of inputs is remarkably similar. By ensuring a diverse distribution of values, it keeps the model from falling into the trap of collapse.

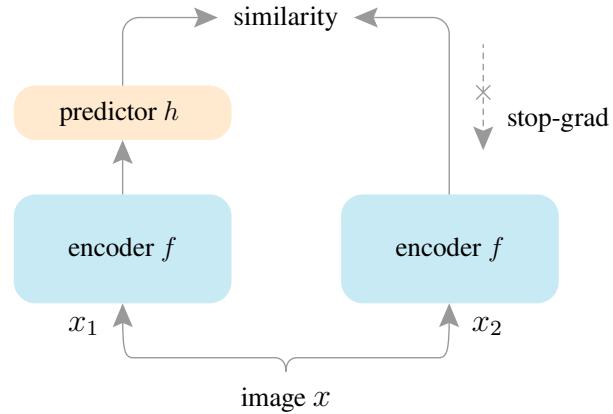


Figure 3.9: SimSiam’s overall architecture, adopted from [114]

The transformative journey from the Contrastive SSL to the Self-Distillation paradigm revolves around a number of crucial evolutions that try to address the central challenges in SSL, namely, model collapse and large batch size requirements. This includes the deployment of dual views of the same data instance, rather than generating positive/negative pairs via data augmentation. Furthermore, the strategic asymmetric updating of the networks using various mechanisms contributes significantly to this progression. Self-distillation methods offer several other advantages over contrastive methods for SSL, such as needing lower batch size due to not using negative pairs thus lower computation power and simpler Learning Dynamics. These innovations have led to a promising alternative to contrastive SSL.

In this thesis, we heavily incorporate self-distillation ideas, and introduce novel methods for both IEA-GAN (see Chapter 5) and YonedaVAE (see Chapter 6) models.

Chapter 4

Deep Generative Models for Detector Simulation: A Taxonomy

4.1 Introduction

The core concept of generative models is derived from the training of a density estimator that produces samples approximating the distribution of the training data. The initial generation of neural network-based generative models, also known as energy-based models [119, 120], attempted to accomplish this by establishing an energy function proportional to the likelihood for data points. However, these models faced challenges in scaling up to high-dimensional, complex data, like natural images. They also necessitated the use of Markov Chain Monte Carlo (MCMC) sampling [121], a method required during both training and inference stages. This method, characterized by its iterative nature, often resulted in a slow and inefficient process. The resurgence of interest in generative models in the past few years can be attributed to the availability of larger datasets and significant advancements in deep generative models (DGMs) for natural images. These DGMs have pushed the boundaries in terms of visual quality, sample diversity, and speed of sampling. In Particle Physics, the application of DGMs was first studied in the “Fast and Efficient Simulation” campaign [122] that sparked the search for faster and more storage-efficient simulation methods of collider physics experiments.

The remainder of this chapter is structured as follows. Firstly, I formulate the problem of detector simulation using deep generative models. Then, I give an algorithm taxonomy of existing methods in the category of latent variable approaches. This section presents a general framework, discusses common generation strategies and tasks in detail, and introduces a thorough review of works of each type.

4.2 Problem Definition

The detector response can be defined by a triplet $D_e = (\mathbf{L}_e, \mathbf{C}_e, \mathbf{H}_e)$ for each event e , where $\mathbf{L} \in \mathbb{Z}$ is the set of detector layers/sensors indicators, $\mathbf{C} \in \mathbb{R}^n$ is the global attribute of the

corresponding event, $\mathbf{H} \in \mathbb{R}^d$ is the hit manifold per sensor/layer like tracks or clusters which can be represented for instance by a grid-like an image and sequence, or a set like point clouds and graphs. Given a set of M observed detector responses $\mathbf{D} = \{D_i\}_{i=1}^M$, DGMs learn the distribution of these signatures $p(\mathbf{D})$, from which new responses can be sampled $D_{\text{new}} \sim p(\mathbf{D})$.

This chapter presents both the latent variable and non-latent variable approaches as the mainstream detector response generation models. latent variable approaches follow an encoder–sampler–decoder pipeline. It firstly maps the data into a hidden space through an encoding function, manipulates the hidden variables to reflect the desired properties of the detector response to be generated, and then generates new samples based on latent codes through a decoding function.

4.3 Algorithm Taxonomy

4.3.1 Latent Variable Approaches

Within the scope of latent variable techniques, the given data is mapped into a stochastic latent space. An *i.i.d* sample from this distribution is then fed into a decoder that reconstructs the original data structures. Prior to delving into an in-depth exploration of distinct models, I initially established a bird-eye-view pipeline that encompasses encoding, sampling, and decoding stages. This pipeline allows us to encapsulate the majority of the existing generative models used for detector simulation within a single unified framework. In accordance with this framework, I classify various methodologies based on their interaction with three pivotal components:

The Encoder. The encoding function $f_\theta(z|D)$ maps the detector response triplets to a dense, continuous (or quantized) topological space. To ensure the learned latent space is meaningful for generation, depending on the data type and the inductive bias, one incorporates various morphisms (e.g., Convolutional Neural Networks [123], Graph Neural Networks [124], DeepSets [125], etc) as the encoder. In this way, the encoder function f_θ generates the parameters for a stochastic distribution that adheres to a prior distribution denoted as $p(z)$.

The Sampler. Following the encoding process, the model that generates the detector response draws samples from the marginal distribution $z \sim p(z)$. There are generally two prevalent strategies for this sampling process: **random sampling** and **controlled sampling** [126]. Random sampling refers to randomly selecting latent codes from the learned or prior distribution. On the other hand, guided or controlled sampling is designed to sample the stratified latent code with the specific goal of generating samples that exhibit certain preferred characteristics. In most tasks, controlled sampling is model-dependent and necessitates an additional optimization component that goes beyond the scope of random generation.

The Decoder. Upon obtaining the latent codes drawn from the learned distribution, the decoding mechanism is responsible for stochastically reconstituting them into a data

manifold. Due to the multi-objective and fine-grained learning nature of detector simulation, the decoding phase is inherently more complex than the encoding stage. Typically, the decoders could be grouped into three categories: **sequential generation**, **one-shot generation**, and **zero-shot generation**. Sequential generation refers to generating sensor/layer information in a set of consecutive and autoregressive steps, usually done sensor-by-sensor or hit-by-hit. One-shot generation, instead, refers to generating the whole detector signature in one single step. Zero-shot generation is when the model generates an unseen set of detector information where the control parameter (e.g. incident energy, luminosity, or hit multiplicity) goes beyond the training data, thus the model has to generalize well to the Out-Of-Distribution (OOD) domain.

4.3.1.1 Sampling Strategies

After learning a latent space for representing the input data, the generative model samples from the learned distribution during the generation inference. The sampling strategies could be divided into two categories, *random sampling* and *controllable sampling*. Random sampling simply draws latent samples from the prior distribution, in which the model learns to approximate the distribution of the observed detector responses. The latter, on the contrary, samples detector responses with controls over the desired properties (i.e. detector geometry, angles, energy, luminosity, and beam parameters). Therefore, for latent variable approaches, random sampling is relatively trivial, while controllable sampling usually requires extra effort in algorithm design.

Controllable generation usually manipulates the randomly sampled $z \sim p(z)$ or the encoded vector $z \sim p(z|D)$ in the latent space to obtain a final representation vector \hat{z} , which is later decoded to the detector response representation with expected properties. There are three types of commonly used approaches:

- **Disentangled sampling** factorizes the latent vector z with each dimension z_n focusing on one property p_n , following the disentanglement regularization that encourages the learned latent variables to be disentangled from each other. Therefore, varying one latent dimension z_n of the latent vector z will lead to property change in the generated detector responses.
- **Conditional sampling** incorporates a conditional code c that explicitly controls the property of generated detector information. In this case, the final representation \hat{z} is usually a modulation of z and c .
- **Traverse-based sampling** searches over the latent space by directly optimizing the continuous latent vector z to obtain \hat{z} with specific properties or uses heuristic-based approach (e.g., linear or nonlinear interpolation from z to obtain \hat{z}), to control the property of the generated detector hits.

In this thesis, all the developed models, PE-GAN, IEA-GAN, and YonedaVAE highly depend on Conditional sampling. IEA-GAN uses adversarial conditional sampling ((see Chapter 5)), and YonedaVAE introduces a self-distilled Adaptive Top-q sampling (see Chapter 6)

as a specialized version of conditional sampling. In my exploration of PE-GAN, I experimented with traverse-based sampling through latent optimization [127]. However, both the training and testing phases proved to be computationally expensive.

4.3.1.2 Generation Strategies

The decoder restores the latent code back to the data manifold. In this study, due to the sparse, high dimensional, and unordered nature of detector data, the resulting outcome of the decoder often faced challenges in accurately reconstructing the data manifold or generating new ones. This led to artifacts or inaccuracies in the generated detector responses. To address this issue, existing works take three types of generation strategies for detector response generation, one-shot generation, sequential generation, and zero-shot generation.

One-shot generation. One-shot generation usually generates an event in one single step. It is achieved by feeding the latent representations to neural networks to obtain the desired representation. In practice, various neural networks could be utilized, including 2D Locally Connected Networks [122], 2D and 3D Convolutional Neural Networks (CNN), Graph Neural Networks (GNN), Multi-layer Perceptron (MLP), according to different types of detectors and representations to be generated. The advantage of one-shot generation is that it generates the whole event in a single step without sequential dependency on layer ordering or hit ordering. If the detector topology has a non-sequential ordering of sensors (non-cylindrical topology), it is not feasible to treat the sensor information, \mathbf{L} the same as the hit manifold \mathbf{H} . Such types of non-trivial topologies are also present in the Mesh representation of 3D objects which consists of a collection of vertices (like detector hits) and polygon faces (like sensor information).

The PXD detector carries such a non-trivial topology. Thus, in order to capture such topology-related intra-event relationship, I will introduce the *Relational Reasoning Module* (see Chapter 5) which is a modified Transformer Encoder model that acts over the event information with the advantage of bi-directional information aggregation.

Sequential generation. In contrast to one-shot generation, sequential generation generates the detector responses consecutively in a few steps. As there is ordering naturally defined for detector layers such as a Calorimeter detector, sequential generation has to follow a certain sequential inductive bias for the generation. This is usually done by generating probabilistic sensor features while sampling and feeding step-by-step the reconstructed detector response following a predefined node ordering. Despite slow sampling, sequential generative models, enjoy the benefit of auto-regressive reasoning which prompts a precise correlation modeling of the data. Therefore, it could be easily combined with constraint checking in each of the generation steps, when the responses to be generated should obey certain restrictions. Using sequential sampling, when generating a large detector with either high-granularity or long detector layers, the error will accumulate at each step, possibly resulting in discrepancies in the final generated and observed detector signatures. Also, in the case of non-cylindrical topologies like PXD, it could introduce a non-existing sequential bias into the generated data.

To alleviate these issues while benefiting from the auto-regressive reasoning, I will

incorporate a proper relative positional embedding to resolve the bias and introduce a gating mechanism in YonedaVAE to mitigate the error-aggregation (see chapter Chapter 6).

Zero-shot generation. Zero-shot detector response generation refers to the process of generating detector responses without any prior exposure to the specific representation or structure of the “new” target detector signatures. The new target detector signatures could belong to new beam parameters, higher incident energies, higher luminosity, or detector responses for uninstalled sensors. The term *Zero-Shot* comes from the concept of zero-shot learning, which is when a model can recognize or generate outputs for new, unseen categories or tasks without any training examples. This is achieved by leveraging the model’s pre-existing knowledge and generalization abilities, typically acquired during the pre-training phase on large datasets by incorporating symmetries and constraints directly into the generative model. This strategy leverages the latent space’s ability to capture the essential characteristics of the data manifold, enabling the generation of plausible detector responses for regions and conditions beyond the current data at hand. The core idea that is introduced in this study, is to extrapolate to the Out-Of-Distribution (OOD) data by designing a model to be more flexible and adaptive, allowing it to accommodate detector geometries, representations, and conditions beyond the training data.

In particular, to robustly model such capabilities, in Chapter 6 I introduce YonedaVAE, trained on only low luminosity data to integrate domain-specific knowledge into the generative framework, to extend its ability to reason about unobserved and generate plausible high luminosity PXD background.

4.3.1.3 Representative Work

In this subsection, I briefly discuss representative works for deep generative models in detector simulation with an emphasis on how they handle generation and sampling.

Variational Autoencoders. VAE is a simple yet flexible framework and could be adopted for controllable sampling by either modifying the loss function to enforce latent variables to be correlated with properties of interest or to feed the conditional information to different parts of the model. In [128], the ATLAS collaboration conditioned the encoder and decoder directly on the energy of the incident particle to generate showers corresponding to a specific energy. In [129, 130], they utilize the BIB-AE [131] model, conditioned on the incident photon energy. Thus the latent manifold is conditioned on the Energy. They introduce a post-processing module that relaxes the trade-off between the accuracy of the emulated hit energy spectrum and the reproduced shower shape. This module is an MLP-based network that fixes the hit energy spectrum resolution between the input and generated images. In the later efforts [132], along with improving their model, inspired by [133], the authors use a Kernel Density Estimator (KDE) [134] to fit the learned latent manifold for the inference time and use rejection sampling for the correct density estimation. In [135], they leverage the Sinkhorn Autoencoder [136] to have a trainable prior approximation, namely a noise generator, to encode and generate embeddings with the same distribution in the latent space. In order to overcome the mode collapse issue and promote diversity, they include additional regularisation on the autoencoder’s latent space. Following [137],

they compute a similarity matrix for the neural network's weights according to the cosine similarity between its different layers to assess the diversity. DeepRICH [138], designs a conditional latent space as a combination of CVAE [139] and infoVAE [140], where the latent variable σ is determined using a Bayesian Optimization [141]. The control variables in this conditional VAE are the kinematic parameters of each particle learned by an auxiliary classifier over the encoded latent manifold as a regularization. They only consider the reconstruction (emulation) of their dataset. GVAE [142], introduces a graph-based VAE architecture for learning the representation of collision events without any controllable sampling for emulation. Orzari et.al [143, 144], develop a VAE for generating constituents of hadronic jets as sets. Although they incorporate a permutation invariant loss, Chamfer distance [95], instead of the typical mean squared error (MSE) as the reconstruction loss, their model does break the permutation equivariance using 2d convolution layers in the encoder. They regularize the model by constraining the p_T and the invariant mass to follow the desired jet characteristics. The authors later in [145], fix this issue by using DGCNN [146] permutation equivariant layers. Abhishek at.al [147], incorporated a Discrete Variational Autoencoder (DVAE) based model [148, 149, 150] with hierarchical dependencies of latent variables and a Restricted Boltzmann Machine (RBM) [151] latent prior using block Gibbs sampling for generation of Calorimeter showers. They also tackle the sparsity of the showers with a learnable masking like [152]. Cresswell et.al [153], develops a manifold hypothesis-inspired model (density estimation) [154, 155, 156] to do a dimensional reduction to the Calorimeter data to speedup the inference-level sampling process.

Otten et.al [133], introduce a method called buffering density information given the encoded events. They construct a prior by aggregating a subset of the encoded training data by saving all the parameters of the Gaussian distributions for all events in the training data to a file, which constitutes the buffer. At inference time, to increase the variance and avoid overfitting to the training data, they sample also from the buffered Gaussian distributions with a variance control factor. Collins et.al [157, 158], using a ParticleFlow [159] beta-VAE [160] finds an interpretable and meaningful representation of the jets and their information complexity by analyzing the VAE's latent information. Moreover, they leverage β from a fixed hyperparameter to an input of both the encoder and decoder networks. Ilten et.al [161], for the first time, incorporates a conditional sliced-Wasserstein VAE [162] for full hadronization simulation.

Generative Adversarial Networks. GAN-based models, by design, allow easy implementation of controllable sampling due to introducing a property discriminator for desired properties. Although, taming its convergence is very difficult. The application of GAN-based generative models as implicit density estimators was embarked on by [122] where they simulated 2D jet images for high energy W bosons and QCD jets as their conditional classes while introducing 2d locally connected layers (LAGAN). CALogan [163, 164, 165] employs the LAGAN layers to generate layer-wise two-dimensional images that were conditioned on the primary particle energy (E_p) ranging uniformly from 1–100 GeV. Vallecorsa et.al [166, 167, 168] uses 3D ACGAN [84], to generate the calorimeter showers. In [169], they add incident angle conditioning as well. Erdmann et al [170], uses WGAN [76] with continuous air shower energy conditioning using a constrainer network (like ACGAN).

Musella et al [152], for the generation of sparse hadronic jets using a U-net [171] module for the generator, introduce a decision-making method by adding an additional channel to their output as a mask probability to decide if a pixel should be zero or not. Srebre et al. [172], used the WGAN-gp [173] to simulate the PXD background hitmaps with random sampling. This model is the baseline for the IEA-GAN study, as they also used the same dataset. In [174], they do controllable sampling by conditioning the WGAN-gp model for the LHCb Calorimeter images. In [175], they introduce an emulator-simulator setup that benefits from the Siamese Network [176, 177, 178]. They show that using their parameter-to-image grid 2-stage training pipeline, and they can model complex functions. In the pre-training stage, the goal is to learn an emulator distribution that matches the Monte-Carlo simulator distribution using the Siamese network to learn the similarity of the simulated and emulated images. Then, at the next stage, a generator will be trained to learn to map the random noise to the parameter space. All these stages follow an adversarial training regime. Diefenbacher et al. [179] introduces a method for refining the precision of GANs using a post-processing re-weighting and tuning function based on [180, 181]. Kansal et al. [182, 183] for the first time choose a more sparse representation of the detector data and introduce a graph GAN based on [184] layers. In [185], they introduce the Latent Space Refinement (LaSeR) protocol to enhance the precision and the topological obstruction of sampling by refining the predictions of a generator. In LaSeR, each generated sample is assigned a weight, which is then mapped to the corresponding latent space point. Rather than directly sampling from this weighted latent space, which could lead to biased results, the authors propose training a second generative model, the refiner, to transform the weighted latent space into an unweighted one. Shirobokov et.al [186], introduce a new approach that synergizes deep generative models and non-differentiable simulators. They show that one can both approximate the stochastic behavior of the simulator and enable direct gradient-based optimization of an objective by parameterizing the latent variable model with the relevant parameters of the simulator. Jaruskova et.al [187], improves the calorimeter simulations over the lower energy depositions with AdaGAN-based [188] ensemble of GANs.

Hashemi et.al [189], for the first time uses GAN for directly emulate high-level features computed from the reconstructed $Z \rightarrow \mu\mu$ events. DijetGAN [190] a simple GAN with random sampling for the simulation of QCD dijet events. Butter et.al [191], do top pair generation with a modified MMD-GAN [75] where the MMD kernel helps to describe on-shell resonances as well as tails of distributions, an improvement to [189]. They also study the statistical uncertainties and do ablation studies of the GAN approach to the event simulation using GANs. Carrazza et al. [192] employ CycleGAN [193] with the cycle-consistency loss to create mappings between two domains of Lund images [194], different categories of jets. Farrell et al. [195] apply GANs to generate full particle physics events conditioned on physics theory parameters. Li et.al [196], introduces a style-based [197] conditional GAN [83] to predict lepton decay angles in the rest frames of W bosons in Vector Boson Scattering (VBS) processes. Their approach addresses the challenge of missing neutrino information in the final state, which traditionally hampers the full determination of lepton angles. Alanazi et.al [198, 199] develop a GAN-based model that does an importance sampling over the

generated features that improve the sensitivity of the discriminator. Prieto et.al [200] proposes a style-based quantum GAN to generate events with a 3-qubit model. Howard et.al [201] incorporates the Sliced-Wasserstein VAE [162] and the theory-based physics constraints in an unsupervised setting for event generation. Anderlini et.al [202] introduce a distilled GAN from an ensemble of models [203] to reduce the variance for a maximally diverse set of models. In the inference time, they test the performance of their model in the Out-Of-Distribution (OOD) regions of the phase space.

4.3.2 Non-Latent Variable Approaches

Apart from the latent variable models, some generation methods do not map the observation into the latent space. In other words, they perform the generation in the raw space and directly generate the data representation.

4.3.2.1 Representative Work

Flow-based Models. Flow-based Models [53, 204] learn an explicit bijection between the data distribution and a known distribution (usually a simple one, like a multivariate normal). This allows for exact likelihood computation and exact sampling but does not involve a latent space in the same way as models like VAEs or GANs. The primary limitation of this category of models is their inadequacy in generating detector data with high resolution or granularity. Specifically, these models often struggle to capture the fine details and subtle variations that are crucial for applications requiring a high level of precision and fidelity, such as high-granularity detector simulation. Therefore, while normalizing flows offers advantages such as exact likelihood optimization and tractable inference, they might not be the most suitable choice for tasks that require the generation of high-resolution or granular detector data.

For the first time, the CaloFlow [205] applies the normalizing flow to the simplified calorimeter geometry with 504 cells. Their model is a combination of MADE blocks [206] and RQS transformations [207]. CaloFlow provides the additional benefit of tractable likelihoods with application to parameter inference for particle reconstruction. In CaloFlow II [205], they incorporate Knowledge distillation to transfer the probability density of the stronger model (teacher) to the student, an Inverse Autoregressive Flow (IAF) [208] model which is much faster in sampling. Xu et al. [209], develop a conditional Normalizing Flow based on [210] with the emphasis on modeling the correlation between the kinematic variables.

Diffusion Models. Diffusion models or deep score-based generative models [211, 212, 213, 214, 215] employ the concept of a stochastic diffusion process to gradually transform a known noise distribution into a target data distribution. This is very much like simulating a random walk over a series of time steps, each guided by a set of transition probabilities. One of the notable benefits of using diffusion models for high-resolution detector simulation data is their ability to capture intricate data structures. By taking numerous small steps to evolve from a simple distribution to a complex, high-dimensional target, diffusion models can effectively model the subtle nuances and details often required in high-fidelity simulations.

However, there are downsides to consider when applying diffusion models to high-resolution detector data such as PXD. One such drawback is computational cost. The diffusion process involves multiple steps, each of which typically requires its own round of computation, making them somewhat slow samplers and computationally intensive.

CaloScore [216] marks the first application of a score-based generative model for detector simulation. They construct the score function using Conv3D U-nets models [171], conditioned over the normalized incident energy. Mikuni et al. [217] apply a Transformer-based diffusion model to particle jets [183] conditioned fully on the initial jet type, kinematics, and multiplicity. In order to increase the sampling process, they use the progressive distillation mechanism [218], to transfer the knowledge of the Transformer-based teacher to an MLP-based student.

Autoregressive Models (ARM). Autoregressive Models [219, 220, 221], or ARMs, are designed to generate data in a sequential manner. Each new data point or feature generated by these models is conditioned on the preceding elements. Although these models might utilize internal hidden states during the generation process, it's worth noting that these states are not considered latent variables in the traditional probabilistic sense. In the context of detector simulation, ARMs offer the advantage of generating highly correlated data, thanks to their sequential conditioning on previously generated elements. This enables intricate probabilistic relationships to be captured, offering the potential for more realistic simulations. However, this strength comes with notable weaknesses, including computational inefficiency and potential error accumulation. The inherently sequential nature of ARMs makes them computationally intensive and difficult to parallelize, leading to slower data generation processes. Moreover, errors at the early stages of the sequential generation can propagate, potentially causing significant inaccuracies in the final high-resolution output.

Lu et al. [222] model the joint distribution of the data directly by breaking it down into a product of conditional distributions. In other words, each data point as pixels is modeled autoregressively as being conditionally dependent on the previous data points. This model takes sparseness into account during learning and generation, which is a key feature of this work. Liu et al. [223, 224], also generate Calorimeter responses autoregressively while taking into account variable detector sizes as a geometrical (detector size) conditioning for OOD detector geometries. Di Bello et al. [225], introduce a modified Transformer model that is a combination of GNN as an encoding module and Slot-Attention [226] layers as a context-injecting layer to the probabilistic decoder which is a GRU-based model [227]. Finke et al. [228], develops a masked Transformer Encoder-based model [29], based on TraDE (Transformers for Density Estimation) [229] for discretized and ordered jet constituents.

4.4 Applications

In this section, I give an overview of the applications of DGMs in detector simulation. Specifically, I categorize them in three concrete branches, *statistics amplification*, *Amortised generation*, and *OOD simulation*. Then, I illustrate their formulations in detector response generation and how different approaches and incorporation of inductive biases lead to

success in various applications. “Inductive biases” in DGMs refer to the assumptions the model makes to predict outputs for both in- and out-of-distribution inputs. These biases are inherent in the model’s architecture and learning algorithm such as symmetries and adversarial robustness that guide the model’s learning process [230, 41, 232, 233, 234, 235, 236, 237, 7, 231]. In general, for DGMs in particle physics, the inductive biases could include, distributional assumptions such as smoothness inductive bias, structural assumptions such as relational inductive bias, and Physics-informed Assumptions such as Energy-Momentum conservation bias.

4.4.1 Statistics Amplification

As surrogate models are being used for detector signature emulation to do sample amplification [238], it is very important to quantify the statistical power of the generated dataset. Sample amplification is a procedure where there exists a map that takes a finite, initial subset of data and generates an extended one. In this regime, DGMs work as classical parametric fits [239, 240] to the training that also works as data augmentation methods. The question then would be to which extent they can interpolate. In other words, *how can one quantify the diversity and uncertainty of DGMs?* This involves understanding the limitations and potential biases of the DGMs and possibly introducing methods to quantify the diversity or amplification factors [241, 242, 243]. For amplification of statistics, the smoothness assumption where the physics probability densities are smooth is very natural. In DGMs, the smoothness inductive bias is an assumption that similar inputs will produce similar outputs up to model interpolation. Regularization techniques in machine learning, which are used to encourage the model to learn smoother functions, are a common manifestation of the smoothness inductive bias.

The smoothness inductive bias is studied in [244, 243, 245] for a simplified VAE-GAN [130]. Their study revealed that individual-generated shower samples contain less information compared to a single real data point. However, when the number of generated shower samples increases significantly, the information contained within the generated sample set eventually levels off. As a result, they demonstrate the ability of DGMs to not only sample from implicitly defined distributions but also to leverage enhanced interpolation or fitting capabilities.

In Chapter 6, similar results are being achieved using an Information Theory approach incorporating a robust diversity measure, Vendi Score [24].

To improve interpolation, it’s crucial to consider how to diversify samples. In this study, by Chapter 5 I enhance the diversity and complexity [231, 246] of generated samples by introducing Self-Supervised Learning [247] techniques that can be incorporated in any DGMs. In particular, I introduce a Uniformity Loss that helps the model to avoid mode collapse and to generate more diverse samples by maintaining the “uniformity of information” inductive bias for the discriminator.

4.4.2 Amortised Generation

For detector simulation, amortized generation generally refers to a strategy where a computationally expensive simulation process is replaced or approximated by a faster and more efficient surrogate model. Detector effect simulations and parametrization can be computationally intensive due to a high space- and time-computation complexity. An “amortized” approach would involve training DGMs on a large set of data. Once trained, this model can generate new simulations much faster than the original simulation process. The term amortized in this context refers to the fact that the upfront cost of training the model is spread out, or “amortized,” over the many simulations that the model generates.

In an amortized generation, the sub-tasks are either “Fast Simulation” to overcome the high time complexity of detector simulation or “Data Compression” to overcome the high space complexity of detector simulation. Based on the different types of data manifolds, various inductive biases are being incorporated. One can categorize different representations into two main classes, “geometry-dependent” and “geometry-independent” representations. Geometry-dependent approaches look at the detector signatures as grid-like structures. For example, 1d fixed representations come with sequential local and translation invariance [248, 249]. 2d representation with local and translation-invariant inductive bias, which was studied for showers, jets, and detector hits in depth for different detectors. Then, in order to capture the detector’s layer-by-layer association and correspondence, 3d grid-based models [166, 167, 169, 168, 216] are studied. However, as a consequence of translation invariance, it has the drawback of stationary assumption over the temporal/spatial features.

Geometry-independent approaches, on the other hand, are more suitable for detector signature simulation due to the unordered and variable length of sensor data, and their heterogeneity and sparsity. For example, graph-based models [142, 182, 145, 183, 225], besides the variable-length assumption and relational inductive bias, it assumes graph isomorphism inductive bias as well. This bias ensures that the model focuses on the structural information contained in the graph, rather than the specific labeling of nodes. Another geometry-independent approach is considering the set representation of detector signatures. The most important property with set-based models are the permutation equivariant encoders for jet and shower constituents [228, 250, 251, 217, 252], and possibly permutation invariant loss functions [145, 225, 253]. For normalizing flows equivariance under permutation group action is more non-trivial. Proved by Köhler et al. [254], given a Flow-based model F such that the set creation yields an exchangeable distribution, the update is permutation equivariant and invertible, and p_θ denotes the model likelihood, then $(F, -\log p_\theta)$ is permutation equivariant. While maintaining the variable-length assumption, dropping the permutation equivariance of the encoders would correspond to a sequential inductive bias of detector responses either layer-by-layer [255, 256], or hit-by-hit [222, 223, 224]. While autoregressive models show a better predictive capability [256], the sequential nature of these models can be a disadvantage when it comes to computation as the sampling is rather slow. Moreover, it is noteworthy that ARM-only models lack a latent representation, therefore, it is not straightforward to manipulate their internal data representation which makes it less appealing for tasks like conditional compression and

metric learning.

In an amortized simulation, in all sub-tasks, there is a huge computational and engineering bottleneck, which is the size of the detector to be simulated. Given the previous works, I classify the detector simulation datasets into four categories, namely *the low granularity* such as the jets data with $\mathcal{O}(100)$ number of hits per event, *mid-granularity* such as simplified detector (calorimeter) data with $\mathcal{O}(1000)$ number of hits per event, *high-granularity* data such as ILD Calorimeter prototype with $\mathcal{O}(1e4)$ number of hits per event, and finally the *ultra-high-granularity* data such as PXD or High Granularity Calorimeter (HGCAL) with $+\mathcal{O}(1e6)$ number of hits per event. Previous work on mid-granularity calorimeter simulation is recent approaches [130, 205], incorporating a combination of several GAN-like or VAE-like and Flow-based architectures with less than 30k simulated channels, and 3DGAN [168] for high-granularity calorimeter simulation with only 65k pixel channels. Nonetheless, these studies barely scratch the surface of the profound challenges posed by the PXD detector simulation or the future detectors. Take, for instance, the impending High Granularity Calorimeter (HGCAL) - a component of the High Luminosity Large Hadron Collider (HL-LHC) [257] upgrade program at the Compact Muon Solenoid (CMS) experiment [258]. With an estimated 6.5 million detector channels distributed across 50 layers, the HGCAL's complexity far surpasses the capacity of existing methods, pointing to the urgency of developing more advanced simulation approaches.

4.4.3 Out-Of-Distribution (OOD) Generation

DGMs for OOD and zero-shot learning is an exciting area of research that holds significant potential across various fields including, but not limited to, Drug Discovery, material design, and weather forecasting. Traditional methods for the synthetic generation of objects with enhanced or specific properties are often iterative and costly, requiring extensive manual work or heavy computational resources. In contrast, DGMs with zero-shot capability can deal with new scenarios that are not explicitly present in the training data, making them highly desirable in a wide range of applications [259, 260, 261, 262, 263]. OOD generation of detector signatures is an emerging field in High Energy Physics (HEP) as well, especially through Simulation-Based Inference methods [264, 265]. The current main challenge remains to be the optimization of DGMs, based on available information while avoiding overfitting, and the generalization to cases in which information is scarce or altogether absent, such as extrapolation to beam parameters, energies, luminosities, and geometries where there are no data.

The first example of such OOD detector response generation is the CaloGAN paper [163], where they very briefly show they can generate showers beyond the training incident energy conditions. DijetGAN [190], also discusses extrapolation to new BSM-dependent OOD regions of dijet invariant mass. Anderlini et al. [202], evaluates the uncertainty of GANs in new momentum regions through Background efficiency comparison. Their analysis shows that for Kaons where the background efficiency does not decrease linearly in the OOD, high-momentum phase-space regions, their ensemble model fails to capture this behavior. Whereas, for Muons, due to the monotonic behavior of the background efficiency, the

extrapolation shows better results.

A very important inductive bias for OOD generation is the variable-length assumption, especially in the context of detector simulation, due to the inherently variable nature of the data, characterized by a fluctuating number of detector responses in individual events. DGMs with a variable-length inductive bias have the capability to generate sets/sequences of differing lengths, contingent on the inherent complexity or dynamism of the event being generated. This equips the model with a heightened adaptability when faced with novel and unseen scenarios. Liu et al. [223, 224], by using an ARM approach (length-independent) tries to extrapolate to unseen detector geometries where the extrapolation domain is calorimeter cell sizes. However, they show that the high granularity is a bottleneck to their approach.

4.5 Conclusion

The aim of this chapter has been to provide a structured overview of the landscape of deep generative models applied to the challenging domain of detector simulation in experimental particle physics. This survey is particularly important for the community as it not only categorizes various approaches but also highlights key areas that are currently under-explored or inadequately addressed. These include:

- An absence of effective methods and studies for capturing inter-sensor correlations within detector setups.
- Inadequate metrics for assessing the diversity of generated samples.
- A lack of focused research on length and attribute extrapolation in detector simulation.
- A complete dearth of studies for handling ultra-high granularity detector signatures.

Within this framework, my thesis offers two distinct contributions to the above issues. First, in Chapter 5, I present a fresh perspective for capturing inter-sensor correspondence that introduces a Bert-like transformer as a bidirectional encoder. This is aimed at encapsulating the complex correlations that exist between different sensors within a single event over the Geant4 simulated PXD data. Next, in Chapter 6, I introduce a unified approach that fuses the benefits of bidirectional encoding with auto-regressive reasoning. This results in a "geometry-aware" generative model that offers a more nuanced balance between efficient simulation and the ability to operate in out-of-distribution settings. Incorporating such a geometry-sensitive methodology promises to be a pivotal factor in enhancing the model's extrapolative capabilities. Moreover, I will introduce a data-driven and robust diversity measure for the generated detector responses.

Chapter 5

PXD Background Generation: Simulation Data

5.1 Introduction

As discussed in Chapter 2, classical PXD background simulation is inefficient both from the time-complexity (CPU demand) and space-complexity (storage demand) perspective. That is, this thesis incorporates Deep Generative Models (DGM) as the surrogate model to replace the old Geant4 detector simulation at Belle2.

This chapter elucidates the complexities inherent in emulating PXD data. First, it introduces the concept of Fine-Grained Image Analysis and argues that the PXD background generation task is a Fine-Grained generative one. Then, it presents a comprehensive elaboration of the surrogate model candidates, PE-GAN and IEA-GAN. These generative models, painstakingly crafted and continuously refined, aim to reproduce the multifaceted and intricate nature of ultra-high granularity data of the PXD, a task that presents unique challenges due to their inherent structure and characteristics. We delve into the methodological approaches employed, the obstacles encountered and overcome, and the improvements in both the data-level and downstream physics-level analysis. As we navigate the terrain of simulated PXD data emulation, I will also address the pressing questions surrounding the applicability, robustness, and limitations of deep generative models.

5.2 PXD Images: Fine-Grained Image Analysis

The innate abilities of the human visual system enable us to discern visual differences, allowing us to distinguish not just between broad categories like dogs and birds but between closely related ones such as a Siberian Husky and an Alaskan Malamute. This prowess led to the conceptualization of Fine-Grained Image Analysis (FGIA), intending to impart AI with a refined visual understanding. FGIA's central objective in computer vision involves recognizing and generating images from a variety of subcategories (objects) within a larger category, such as different animal species or various car models. The core challenge lies in

discerning the underlying distinctions that set apart objects that may otherwise appear highly similar.

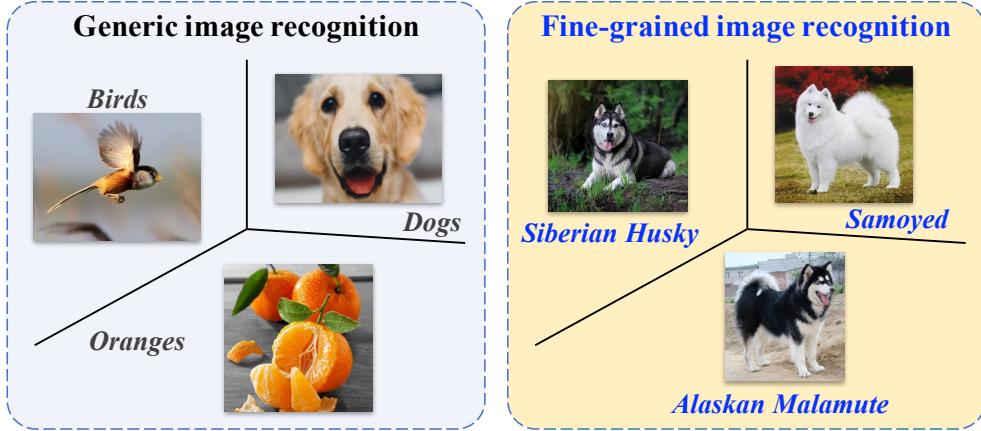


Figure 5.1: Fine-grained image analysis vs. generic image analysis (using visual classification as an example), adopted from [266].

Positioned between basic-level image analysis (generic image categorization) and instance-level analysis (identifying individual entities), FGIA, as shown in Figure 5.1, seeks to distinguish objects from multiple subordinate categories within a broader category (see Figure 5.2).

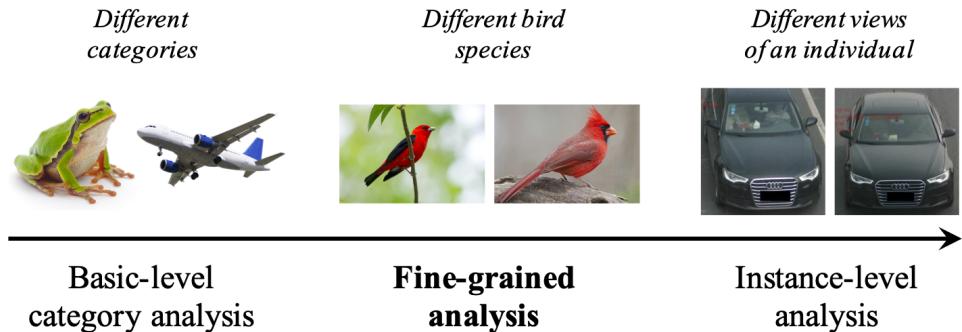


Figure 5.2: An illustration of fine-grained image analysis which lies in the continuum between the basic-level category analysis (i.e., generic image analysis) and the instance-level analysis (e.g. car identification), adopted from [266]

For instance, while generic image analysis might categorize images into broad classes like bird, fruit, or dog, FGIA delves deeper into categorizing breeds of the same species. To achieve this, it becomes imperative to pinpoint subtle visual nuances, such as distinct ear shapes or tail lengths. Recognizing these intricate features is also essential for other FGIA tasks, like generation. The challenge in FGIA stems from the minimal visual differences between sub-categories and the large differences that can exist within a sub-category due

to varying poses, scales, and rotations (refer to Figure 5.3). This complexity distinguishes FGIA from generic image analysis and marks its uniqueness.



Figure 5.3: Key challenges of fine-grained image analysis, small inter-class variations, and large intra-class variations. Different Tern species, one species per row, with different instances in the columns, adopted from [266]

On the other hand, instance-level analysis targets specific entities, not merely categories or sub-categories. Descending the granularity spectrum, identifying individuals, such as through face recognition, emerges as a unique form of fine-grained recognition where individual identity becomes the focal point. Consider person/vehicle re-identification tasks. Here, the aim is to discern if two images depict the same person or vehicle. These tasks deploy FGIA-like techniques, capturing distinguishing object features, harnessing coarse-to-fine structural information, or utilizing attribute-centric models.

The PXD data as pixelated images are created in high resolution and contain rich information that can be similar to the nuanced details considered in FGIA. The data captured by the PXD is extremely granular, allowing for very detailed information on the particle tracks and the different types of background manifested in various image-level artifacts. Inter-class variations in PXD background data refer to different types of background tracks being recorded. The features such as energy levels, trajectories (cluster information), and occupancy slightly vary from one class of sensors to another (see Figure 5.10). On the other

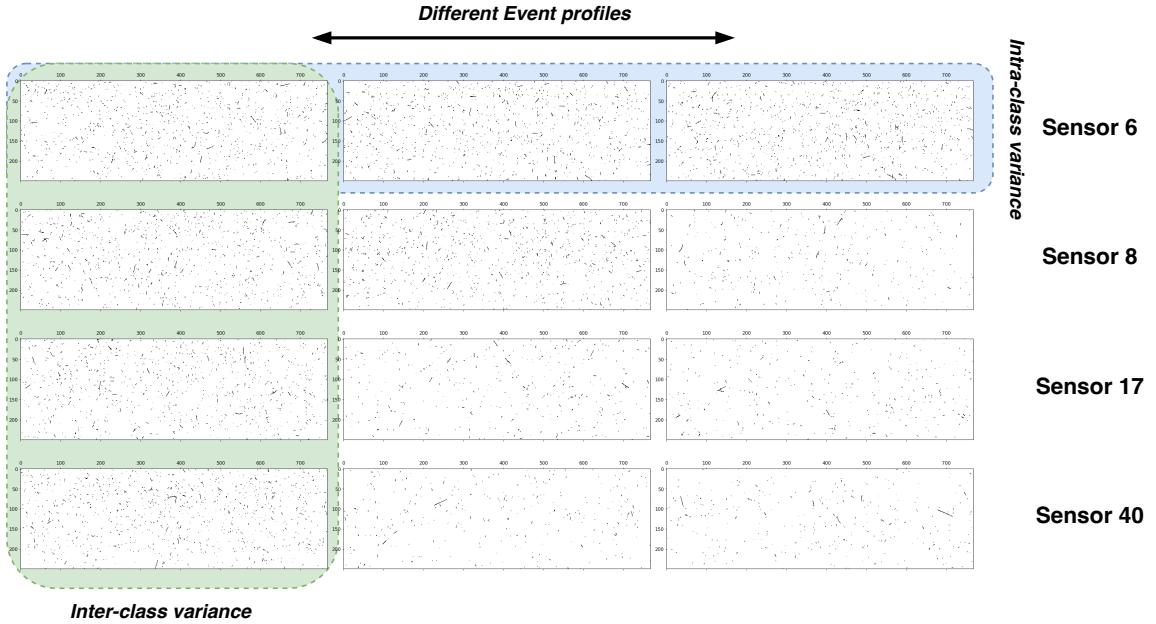


Figure 5.4: PXD background signatures across events for different sensors. The horizontal cases demonstrate various profiles (different clustering, occupancy, etc.) across different events. The vertical cases exhibit similar profiles across sensors.

hand, within each class (PXD sensor), intra-class variations occur due to differences in beam parameters, amount of background, and incident energies across various events.

5.2.1 PXD Generation Formulation

In PXD image generation, one is provided with a training dataset denoted as

$$\mathcal{D} = \left\{ \left(\mathbf{x}^{(n)}, y^{(n)} \right) \mid i = 1, \dots, N \right\}, \quad (5.1)$$

encompassing numerous images and their corresponding class labels, represented by \mathbf{x} and y , respectively. Here, y is drawn from the set $[1, \dots, 40]$, which corresponds to each PXD sensor. Each data pair (\mathbf{x}, y) is a member of the combined space comprising image and label spaces, represented as \mathcal{X} and \mathcal{Y} . This pair is drawn according to the distribution $p_r(\mathbf{x}, y)$.

$$(\mathbf{x}, y) \subset \mathcal{X} \times \mathcal{Y}. \quad (5.2)$$

Notably, the label space \mathcal{Y} is a combined space of all the C subspaces linked to the C categories (number of sensors), that is, $\mathcal{Y} = \bigcup_{c=1}^C \mathcal{Y}_c$. A generative deep network, symbolized by $g_\phi(\mathbf{z}; y)$, parameterized by ϕ , can then be trained for this generic conditional image generation. The training process aims to minimize the generative divergence:

$$\min_{\phi} \mathbb{E}_{(\mathbf{z}, y) \sim p_g(\mathbf{z}, y)} [\mathcal{D}(x, g_{\phi}(\mathbf{z}; y))], , \quad (5.3)$$

where $\mathcal{D}(\cdot, \cdot)$ is a divergence measure assessing the difference between the desired and generated distributions. In the framework of fine-grained generation, the goal is to precisely generate PXD instances of different subordinate granular categories, such as various background profiles, clustering, and occupancy information per sensor, from a certain meta-category (sensor labels), represented as:

$$(\mathbf{x}, y') \subset \mathcal{X} \times \mathcal{Y}_c, , \quad (5.4)$$

with y' standing for the fine-grained label, and \mathcal{Y}_c being the label space of class c regarded as the meta-category. A meta category is a high-level category that encompasses multiple subordinate categories. It's the umbrella term under which various more specific categories fall. A subordinate category is a specific, low-level category within a broader meta category in a hierarchical way. It is more specialized and is usually distinguished by more nuanced attributes or features. The optimization aim for fine-grained generation is:

$$\min_{\phi} \mathbb{E}_{(\mathbf{z}, y') \sim p_g(\mathbf{z}, y')} [\mathcal{D}(x, g_{\phi}(\mathbf{z}; y'))], , \quad (5.5)$$

In this chapter, I only introduce a control parameter over the meta-category (the sensor numbers) and allow the model, IEA-GAN, to explore and learn the subordinate categories (various intra-class variations) in a self-supervised way. In the next chapter (Chapter 6), YonedaVAE utilizes a control parameter over the occupancy (amount of background) of each sensor and event to generate samples with subordinate category attributes.

5.2.2 Challenges Of DGMs for PXD simulation

I have discussed in Chapter 3 and Chapter 4, the limitations of DGMs. However, the task of learning to generate ultra-high resolution detector responses such as the PXD background signatures poses several more intricate challenges. Here, I discuss them in detail:

- **Data dimensionality:** For PXD data, one is dealing with spatially asymmetric high-resolution heatmaps. Each event comes with $40 \times 250 \times 768$ pixel signatures, even more than the notorious high resolution, $1024 \times 1024 \times 3$ ImageNet dataset. The sheer volume and dimensionality of such data pose significant difficulties. Firstly, it necessitates the handling of vast quantities of information, demanding substantial computational resources and sophisticated techniques for efficient data processing and model training. Secondly, it complicates the task of effectively learning the latent variables and dependencies in the data due to the increased complexity associated with such high-dimensional spaces. Thirdly, the risk of mode-collapse escalates with the model potentially memorizing the training data rather than generalizing and learning underlying patterns. Lastly, the spatial asymmetry of the heatmaps adds another layer of complexity, requiring a nuanced understanding and careful modeling of these asymmetric features within the DGM framework.

- **Data frequency and sparsity:** This is underscored by the stochastic nature of background processes. The PXD captures a wide range of charged tracks, that prompts to a dramatic variation of occupancies from order of $\mathcal{O}(10)$ to as high as $\mathcal{O}(1M)$. This variability introduces a significant degree of sparsity in the data - a challenge that amplifies the complexity of generating realistic data using deep generative models as will be discussed in the next chapter. This variation in frequencies means that the model must be robust enough to generate events that span across these vast ranges even with imbalance. The sparsity of the data poses its own unique set of problems. It makes it harder for the model to learn meaningful representations, as the presence of many zero or near-zero elements can make it difficult for the model to discern the underlying structure or patterns in the data. This issue is further compounded when one considers the high dimensionality of the PXD data, as discussed previously. This sparsity can also lead to computational inefficiency. Traditional representations may lead to wastage of computational resources when dealing with sparse data. Therefore, appropriate data structures and algorithms are required to handle this sparsity effectively (see Chapter 6).
- **Fine-graininess and correlation:** The detector responses in an event, a single readout window after the collision of particles, share both statistical and semantic similarities with each other (see Figure 5.4). For example, the sparsity (occupancy) of each image within a class, defined as the fraction of pixels with a non-zero value, shows statistical similarities between detector components as shown in Figure 5.10. As the detector response images show extreme resemblance at the semantic and visual levels [267], they can be classified as fine-grained images. The small inter-class and considerable intra-class variation inherent to fine-grained image analysis make it a challenging problem [266]. The current state-of-the-art conditional models focus on class and intra-class level image similarity, in which intra-image [90], data-to-class [87], and data-to-data [77] relations are considered. However, in the case of detector simulation, classes become hierarchical and fine-grained, and the discrimination between generated classes that are semantically and visually similar becomes harder. Therefore, a wide range of SOTA models on popular natural datasets show extensive class confusion [92, 93] at the inter-class level. In addition, since the information in an event comes from a single readout window of the detector, the processes happening in this window affect all sensors simultaneously, leading to a correlation among them, as shown in Figure 5.11. This study will show how this fine-grain intra-event correlation plays an important role in the downstream physics analysis.

5.3 Prior Embedding GAN (PE-GAN)

In the realm of high-resolution image generation, especially in the intricate domain of PXD images, the choice of architecture, modeling techniques, and training strategies can significantly impact the efficacy of the generated output. With such complexity of data

and increasing demands for computational efficiency, there's a pressing need to identify and adapt the right methodologies that strike a balance between performance and computational feasibility. Over the years, a plethora of generative models have been proposed, each with its unique strengths and weaknesses. The challenge lies not just in selecting the right model, but in tuning it to cater to specific domain constraints and data peculiarities.

This section delves into my journey of identifying, customizing, and refining a generative model tailored for the PXD image generation task. After rigorous experimentation with various model structures and hyperparameters, informed by the evaluation metrics, I converged on a modified and deeper version of the SAGAN model, supplemented with contrastive conditioning, consistency regularization, and integration of prior knowledge on PXD sensors. The ensuing sections provide a detailed account of the model selection process, the challenges encountered, and the strategies employed to overcome these hurdles, culminating in the proposal of the PE-GAN model. This section is based on the paper [6].

5.3.1 Transition to BigGAN

My journey commenced with the integration of class labels, representing the 40 sensor numbers, into the previous model for PXD background generation, WGAN-gp [172]. While WGAN-gp served as the initial baseline, the primary challenges encountered were its slow event-based image generation due to its large kernel filter size and its inability to wholly capture the pixel intensity distribution, as depicted in Figure 5.5.

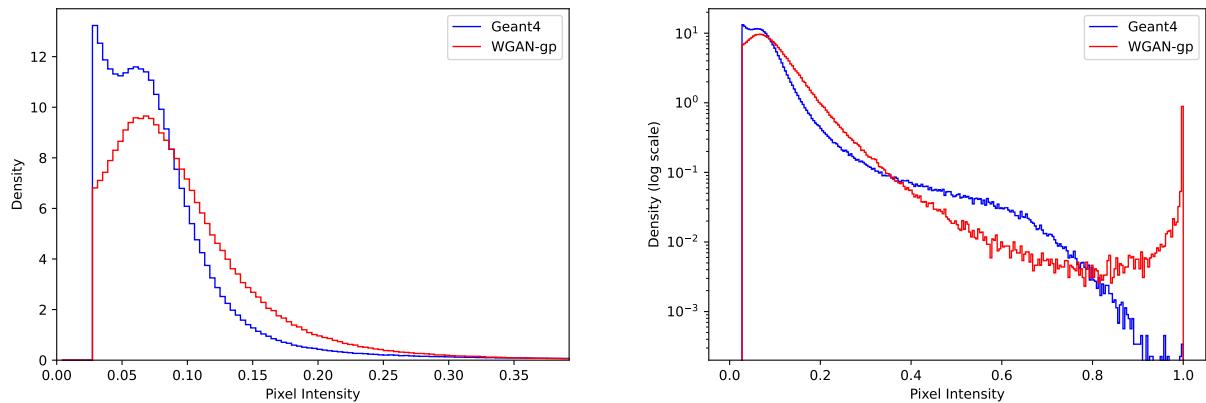


Figure 5.5: Charge intensity distribution for the WGAN-gp without conditioning [172] on PXD as the baseline.

Subsequently, my attention was directed towards more intricate but rapid models like BigGAN[82], which utilizes a reduced kernel size and a specialized residual block tailor-made for high-resolution image generation. During its training, however, I observed training collapse, emphasizing the need for early stopping. As outlined in BigGAN's paper [82], the first singular values of the weight matrices for both the generator and discriminator generally increased throughout training. However, any indications of mode collapse or

discriminator overfitting were accompanied by a drastic surge in these singular values for the generator.

Numerous countermeasures were attempted, including decreasing the discriminator’s learning rate and experimenting with various regularization techniques like Orthogonal Regularization, Dropout, and L2 regularization. Other approaches, such as Latent Optimization [127] and Feature Quantisation [268], also proved futile. This cascade of unsuccessful attempts guided my decision to adopt a more straightforward model.

5.3.2 Adopting Self-Attention GAN

Drawing from the intricacies of our dataset, memory constraints, and the feedback accrued during training, I transitioned to the Self-Attention GAN (SAGAN) [90] model. SAGAN equips both the generator and discriminator to model extended dependencies over the image manifold, enabling the generator to assimilate both local and global image details. Despite the inherent advantages of convolutional blocks like parameter sharing and translation invariance, they sometimes fall short in replicating complex structures necessitating long-range information. This deficiency is aptly addressed by the attention mechanism intrinsic to SAGAN. This architectural pivot offered invaluable insights regarding the influence of model variations on the results. The culmination of my efforts was a model deeper than SAGAN but relatively simpler than BigGAN.

5.3.2.1 Model Refinements

In the Prior Embedding GAN (PE-GAN) model, the depth of each block was increased for both the generator and discriminator, with each layer incorporating 32 multiplication channels. An in-depth evaluation of SAGAN prompted me to eliminate Projection Discrimination, a concept originally introduced in SNGAN [85]. The primary objective of Projection Discrimination is to enhance the projection of the real image embeddings onto the corresponding target embeddings while minimizing the inner-product values for fake images. Its limitation of solely harnessing the data-to-class relationship often led to discriminator overfitting and training collapse [77]. To address this, I incorporated the contrastive conditioning discrimination mechanism [77]. This technique is rooted in a conditional contrastive loss derived from metric learning that evaluates data-to-data relationships within each batch. Consequently, the discriminator can discern not just the data-to-class but also data-to-data interrelations among samples.

Given that our task is a fine-conditional image generation, I further appended embeddings of prior global sensor occupancy data along with the class labels to both the discriminator and generator. Including prior data provides more context, enabling the model to understand and internalize the nuances of each class more effectively. Such enriched representations can be crucial in tasks where class boundaries are intricate, and minor variations can lead to significant differences in outcomes. Let $\mathbf{X} = \{x_1, \dots, x_m\}$, where $x_i \in \mathbb{R}^{W \times H}$ be sampled training images and $\mathbf{y} = \{y_1, \dots, y_m\}$, where $y_i \in \mathbf{Z}$ be the corresponding labels from 1 to 40, and $\mathbf{s} = \{s_1, \dots, s_m\}$, where $s_i \in \mathbf{R}$ be the mean occupancies for each class of sensors.

Also, a linear projection function $h^{(d)} : \mathbb{R}^k \rightarrow \mathbb{S}^d$ and an embedding function $\mathbf{e} : y_i \rightarrow \mathbb{R}^d$ is defined. Then, using the linear projection of the output of the discriminator, one can define the conditional contrastive loss as,

$$L(x_i, y_i, s_i; t) = -\log \left(\frac{\exp(h_i^T a_i/t) + \sum_{k=1}^m \mathbf{1}_{k=i} \cdot \exp(h_i^T h_k/t)}{\exp(h_i^T a_i/t) + \sum_{k=1}^m \mathbf{1}_{k \neq i} \cdot \exp(h_i^T h_k/t)} \right)$$

By minimizing the above-mentioned contrastive loss, the distances between the embeddings of images with matching labels are reduced while distances between differing labels are maximized. This mechanism considers both the data-to-data relations $h_i^T h_k$ and data-to-class relations $h_i^T a_i$ without any augmentations. As a result, the discriminator is guided not solely by the image but is also informed by fine-grained prior information, functioning as an injection of inductive bias. This incorporation of prior semantic information, rooted in the distinct mean occupancies extracted from the simulated images, substantially enhanced the model's performance concerning the mean-occupancy distribution as shown in Figure 5.10. Additionally, it augmented the stability of the training process for nearly 20,000 iterations.

Lastly, the introduction of consistency regularization [269] significantly extended our training's stability and delayed potential collapses. The fundamental idea behind this technique is to ensure that the classifier remains invariant to specific semantics-preserving transformations.

Given an image \mathbf{x} and its perturbed counterpart $\mathbf{x}' = T(\mathbf{x})$, which is generated through semantics-preserving noise such as flipping or random shifts, the discriminator's outputs $D(\mathbf{x}; \theta)$ and $D(\mathbf{x}'; \theta)$ should ideally be consistent. Formally, the consistency regularization loss $\mathcal{L}_{\text{consistency}}$ can be expressed as:

$$\mathcal{L}_{\text{consistency}} = \mathbb{E}_{\mathbf{x} \sim p_g} \left[\|D(\mathbf{x}; \theta) - D(T(\mathbf{x}); \theta)\|_2^2 \right],$$

where $T(\cdot)$ represents the semantics-preserving transformation (e.g., image flip, random shift) applied to the image, and p_g is the distribution of the generated images. In our scenario, due to the generator's tendency to produce images with artifacts such as flips and random shifts, introducing such a regularization ensured that the discriminator did not penalize these perturbations excessively. By optimizing the model with this regularization, the robustness of the training process was enhanced, leading to improved stability and longevity in training cycles.

Eventually, PE-GAN, as depicted in Table 5.1, using the above technologies along with orthogonal initialized [89] residual blocks [82] for both the generator and the discriminator and Hinge loss [90], succeeded as the proof of concept sensor-dependent PXD hitmap generation. However, as I elaborate in the next section, PE-GAN had some serious problems including incomplete marginal distributions, low diversity, and very bad performance over the downstream Physics analysis.

Generator	Discriminator
$z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$	zero-padded gray-scale image
Embed(y) $\in \mathbb{R}^{128}$	$x \in \mathbb{R}^{256 \times 768 \times 1}$
$ch = 32$	
Linear(128+128) $\rightarrow 4 \times 12 \times 16ch$	ConvBlock $1 \rightarrow ch$
ResBlock $16ch \rightarrow 16ch$	ResBlock down $ch \rightarrow 2ch$
ResBlock up $16ch \rightarrow 16ch$	ResBlock $2ch \rightarrow 2ch$
ResBlock $16ch \rightarrow 16ch$	ResBlock down $2ch \rightarrow 4ch$
ResBlock up $16ch \rightarrow 8ch$	ResBlock $4ch \rightarrow 4ch$
ResBlock $8ch \rightarrow 8ch$	ResBlock down $4ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 8ch$	ResBlock $8ch \rightarrow 8ch$
ResBlock $8ch \rightarrow 8ch$	ResBlock down $8ch \rightarrow 8ch$
ResBlock up $8ch \rightarrow 4ch$	ResBlock $8ch \rightarrow 8ch$
Non-Local Attention Block	ResBlock down $8ch \rightarrow 16ch$
ResBlock $4ch \rightarrow 4ch$	ResBlock $16ch \rightarrow 16ch$
ResBlock up $4ch \rightarrow 2ch$	ResBlock $16ch \rightarrow 16ch$
ResBlock $2ch \rightarrow 2ch$	ReLU, Global sum pooling
ResBlock up $2ch \rightarrow ch$	Self-Attention [Proj(Embed(y) Linear(s))],
BN, ReLU, 3×3 Conv $ch \rightarrow 1$	Norm(Linear512 \rightarrow 1024)
Tanh	Conditional Contrastive loss

Table 5.1: The Architecture of the Generator and the Discriminator. “ ch ” denotes the number of multiplicative channels, “ResBlock”, the residual blocks introduced in BigGAN-deep, “BN”, the batch normalization, “ReLU”, the rectified linear unit, “Norm”, a normalization operator, “Proj” and “Linear”, a linear projection or an MLP.

5.4 Intra-Event Aware GAN (IEA-GAN)

Although PE-GAN introduced a conditional setup for PXD background generation and provided a significant speedup to the WGAN-gp [172], it lacks an intra-event grounding. Moreover, the use of PE-GAN’s generated samples in the downstream physics analysis is still not satisfactory (see Figure 5.13). Thus, I embarked on another journey to introduce Intra-Event Aware GAN. This section is based on the pre-print [7].

5.4.1 Introduction and Overview

IEA-GAN is a GAN-based deep generative model based on self-supervised relational reasoning. IEA-GAN’s discriminator, D , takes the set of detector response images $x_i \in \mathbf{R}^d$ coming from one event and embeds them as input nodes within a fully connected event graph in a self-supervised way. An Event graph is a weighted graph where the nodes are the

embedded detector images in an event and the edges are weighted by the degree of similarity between the detector images in each event (detailed description follows in Section 5.4.2.1). It approximates the concept of an event by contextual reasoning using a permutation equivariant Relational Reasoning Module (RRM). RRM is a novel, GAN-compatible, fully connected, multi-head Graph Transformer Network [41, 270] that groups the image tokens in an event based on their contextual similarity. For multi-modal contrastive reasoning, the discriminator also takes the sensor embedding of the detector as class tokens. In the end, it compactifies both image and class modalities information by projecting the normalized graph onto a hypersphere, as shown in Figure 5.6a.

To ensure that the Generator G has a proper understanding of an event and captures the intra-event correlation, it first samples from a Normal distribution, $\mathcal{N}(0, 1)$, at each event as random degrees of freedom (Rdof), and decorates the sensor embeddings with this four-dimensional learnable Rdof. Then, for a self-supervised contextual embedding of each event, the RRM acts on top of this. Notably, Rdof differs from the original GAN [54] Gaussian latent vectors. Rdof can be considered as an event-level learnable segment embedding [39] or perturbation [271] to the token embeddings, which can leverage the diversity of generated images. Combining these modules with the IEA Loss allows the Generator to gain insight and establish correlation among the samples in an event, thus improving its overall performance.

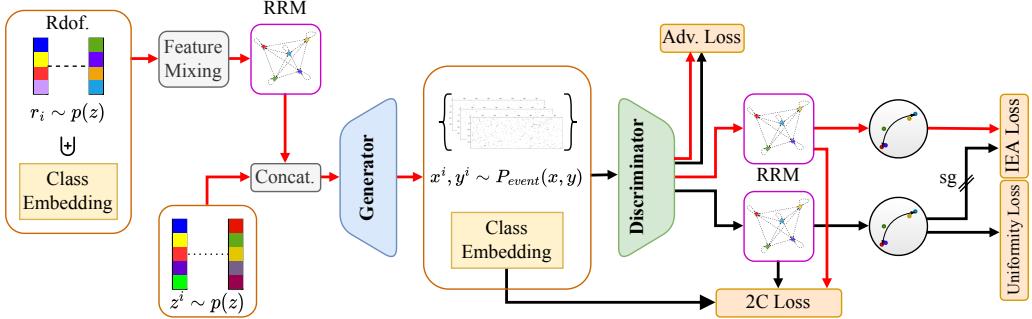
Apart from the adversarial loss, IEA-GAN also benefits from a self-supervised and contrastive set of losses. The model understands the geometry of the PXD detector through a proxy-based contrastive 2C loss [77] where the learnable proxies are the sensor embeddings over the hypersphere. Moreover, to improve the diversity and stability of the training, I introduce a Uniformity loss for the discriminator. The Uniformity loss can encourage the discriminator to give equal weight to all regions of the hypersphere [272], rather than just focusing on the areas where it can easily distinguish between real and fake data. Encouraging the discriminator to impose uniformity not only promotes more diverse and varied outputs but also mitigates issues such as mode collapse.

Another essential part of IEA-GAN is the IEA loss that addresses the class confusion [92, 93] problem of the conditional generative models for fine-grained datasets. In the IEA-loss the generator tries to imitate the discriminator’s understanding of each event through a dyadic information transfer with a stop-gradient for the discriminator. This can improve the ability of the generator to generate more fine-grained samples in the simulation process by being aware of the variability of conditions at each event.

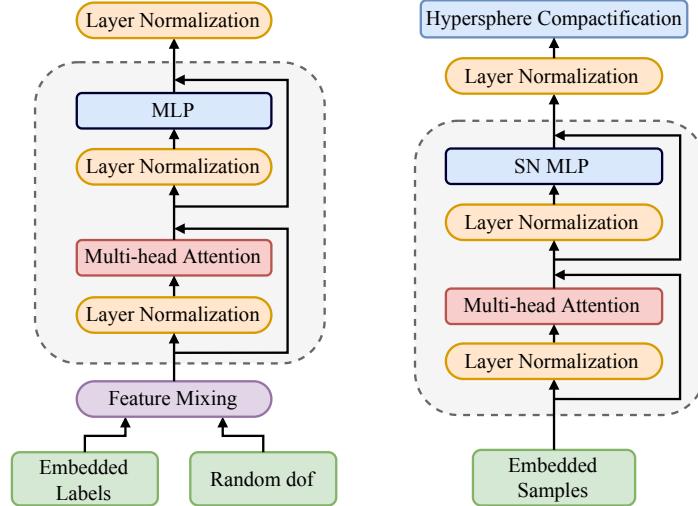
In the next section, we delve deeper into each of these methods and discuss their importance.

5.4.2 Relational Reasoning

Transformers [29] are widely used in different contexts. However, their application in Generative Adversarial Networks is either over the image manifold to learn long-range interactions between pixels [90, 273] or via pure Vision-Transformer based GANs [274] in which they utilize a fully Vision-Transformer [275] based generator and discriminator.



(a) Rdof stands for Random degrees of freedom, which decorates the generator's sensor/layer embedding with an event-level learnable embedding responsible for the generator's intra-event correlation. The Relational Reasoning Modules (RRM) in the generator and the discriminator do the intra-event reasoning by clustering class/image embeddings based on their contextual similarity, respectively. The red lines correspond to the forward and backward passes of the generator. The black lines correspond to the forward and backward passes of the discriminator. The discriminator is trained with the Adversarial Hinge loss, see Chapter 3, 2C loss, see Equation (3.9) and the Uniformity loss, see Equation (5.14). On the other hand, the generator uses the Adversarial loss, 2C loss, and the IEA loss, see Equation (5.11). Sg means stop-gradient for the discriminator from the IEA loss, a self-supervised dyadic-aware loss for the generator.



(b) The Relational Reasoning Module for the generator (left) and the discriminator (right)

Figure 5.6: IEA-GAN architecture (a) and Relational Reasoning Module components (b).

Given the fact that training the Transformers is notoriously difficult [276] and task-agnostic when determining the best learning rate schedule, warm-up strategy, decay settings, and gradient clipping, fusing and adapting a Transformer encoder over a GAN learning regime

is a highly non-trivial task. IEA-GAN successfully merges a Transformer-based module adapted to the GAN training schemes for the discriminator’s image and the generator’s class modalities without any of the aforementioned problems.

5.4.2.1 Event Approximation

An event, a single readout window after the collision of particles consisted of 40 of images each of which a sensor bitmap (image) of size 256×768 . Thus, each event represents a round of detector signature collection. In order to approximate the concept of an event, at each iteration, IEA-GAN should take an event with 40 sensor images. Therefore, I am conditioning the model with the sensor type $[[1, 40]]$ which can be thought of as a mixture of angle and radius conditioning. These conditions have to enter the model as learnable *tokens* as they are not absolute and are context-based. It is impossible to pre-define meaningful sparse connections among the sample nodes in an event. For instance, the relation between images from different sensors can vary from event to event, albeit cumulatively, they follow a particular distribution. Ergo, the model has to learn any dynamical inherited conditions from the data in context (through the Relational Reasoning Module).

To model the context-based similarity between the different detector sensors in each event rather than their absolute properties, I have to use a permutation-equivariant [277, 278] relational block such that can encode pairwise correspondence among elements in the input set. For instance, Max-Pooling (e.g. DeepSets [125]) and Self-Attention [29] are the common permutation equivariant modules for set-based problems. Performing attention on all token pairs in a set to identify which pairs are the most interesting enables Transformers like Bert [39] to learn a context-specific syntax as the different heads in the multi-head attention might be looking at different syntactic properties [279, 280].

Hence, I incorporate a self-attention mechanism with weighted sum pooling as a form of information routing to process meaningful connections between elements in the input set and create an event graph. Each sample in an event is viewed as a node in a fully connected event graph, where the edges represent the learnable degree of similarity. Samples in each event go into message propagation steps of the Relational Reasoning Module (RRM), a GAN-compatible fully connected multi-head Graph Transformer Network [41, 270].

5.4.2.2 Relational Reasoning Module

Specifically designed to be compatible with GAN training policies, the Relational Reasoning Module (RRM) can capture contextualized embeddings and cluster the image or class tokens in an event based on their inherent similarity.

Let $\mathbf{X} = \{x_1, \dots, x_m\}$ be the set of the sampled images in each event, where $x_i \in \mathbb{R}^d$, and $\mathbf{y} = \{y_1, \dots, y_m\}$ be the set of labels, with $y_i \in [[1, 40]]$ for 40 detector (PXD) sensors. I also define two linear hypersphere projection diffeomorphisms, $\mathbf{h}_x : \mathbb{R}^k \rightarrow \mathbb{S}^n$ and $\mathbf{h}_y : \mathbb{Z} \rightarrow \mathbb{S}^n$, which map the image embedding manifold and the set of labels to a unit n-sphere, respectively. The unit n-sphere is the set of points, $\mathbb{S}^n = \{s \in \mathbb{R}^{n+1} \mid \|s\|_2 = 1\}$, that is always convex and connected. The Relational Reasoning Module benefits from a variant of

the Pre-Norm Transformer [29] with a dot-product Multi-head Attention block such that

$$\mathbf{p}_i'^{(l)} = \mathbf{p}_i^{(l)} + \sum_{k=1}^h \sum_{j=1}^m a_{ij}^{(l,k)} \mathbf{W}_{\text{SN}}^{(l)} \mathbf{LN}(\mathbf{p}_j^{(l)}) , \quad (5.6)$$

$$\mathbf{p}_i^{(L)} = \mathbf{h}_x^{\text{LN}} \left(\mathbf{LN} \left(\bigcirc_{l=0}^L (\mathbf{p}_i'^{(l)} + \mathcal{F}_{\text{SN}}[\mathbf{LN}(\mathbf{p}_i'^{(l)})]) \right) \right) , \quad (5.7)$$

where $\mathbf{p}_i'^{(l)} \in \mathbb{R}^k$ is the embedding of each image via the discriminator for layer l of the RRM. \mathbf{LN} is the Layer Norm function [35] and h is the number of heads defined in Equation (3.2). $\mathcal{F}[\cdot]$ is a two layer MLP functional defined as $\mathcal{F}_{\text{SN}}[\mathbf{p}_i^{(l)}] = \text{ReLU}(p_i^{(l)} \mathbf{W}_{\text{SN}}^{(l,1)}) \mathbf{W}_{\text{SN}}^{(l,2)}$ with Spectral Normalization [87]. The logits $a_{ij}^{(l,k)}$ are the normalized Attention weights of the bilinear function that monitor the dyadic interaction between image embeddings in layer l and head k defined in Equation (3.1). $\mathbf{W}_{\text{SN}}^{(l)}$ in Equation (5.6) is the learnable multi-head projector at layer l defined in Equation (3.2) with Spectral Normalization. The output of the composition of all layers via the composition of L functionals, $\bigcirc_{l=0}^L \Phi^l := \phi_{w_L} \circ \dots \circ \phi_{w_0}[\mathbf{p}_i^{(l=0)}] \in \mathbb{R}^{m \times k}$, goes into a Layer Normalization layer where $\Phi^l = \mathbf{p}_i'^{(l)} + \mathcal{F}[\mathbf{LN}(\mathbf{p}_i'^{(l)})]$. $\mathbf{h}_x^{\text{LN}}(\cdot)$ in Equation (5.7) is the hypersphere compactification while the vectors are being standardized over the unit n-sphere \mathbb{S}^n by a Layer Normalization.

For the discriminator, this module takes the set of image embeddings as input nodes within a fully connected event graph applies a dot-product self-attention over them, and then updates each sample or node's embedding via the attentive message passing, as shown on the right of Figure 5.6b. In the end, it compactifies the information by projecting the normalized graph onto a hypersphere, as shown in Figure 5.6a. Embedding the samples in an event on the unit hypersphere provides several benefits. In modern machine learning tasks such as face verification and face recognition [281], when dot products are omnipresent, fixed-norm vectors are known to increase training stability. In our case, this avoids gradient explosion in the discriminator. Furthermore, as S^n is homeomorphic to the 1-point compactification of \mathbb{R}^n when classes are densely grouped on the n-sphere as a compact convex manifold, they are linearly separable, which is not the case for the Euclidean space [282].

For the generator's RRM, I use a simpler version of the above dot-product Multi-head Attention block without the last hypersphere compactification due to the stability issues., as shown on the left of Figure 5.6b. It finds a learnable contextual embedding for each event that will be fused to each class token via the feature mixing layer, which is a matrix factorization linear layer $\mathbf{W}_{\text{SN}}(\cdot)$. Formally we have,

$$\mathbf{q}_i^{(0)} = \mathbf{W}_{\text{SN}}(\mathbf{r}_i \uplus \mathbf{e}_i) , \quad (5.8)$$

$$\mathbf{q}_i'^{(l)} = \mathbf{q}_i^{(l)} + \sum_{k=1}^M \sum_{j=1}^m a_{ij}^{(l,k)} \mathbf{W}^{(l)} \mathbf{LN}(\mathbf{q}_j^{(l)}) , \quad (5.9)$$

$$\mathbf{q}_i^L = \mathbf{LN} \left(\bigcirc_{l=0}^L (\mathbf{q}_i'^{(l)} + \mathcal{F}[\mathbf{LN}(\mathbf{q}_i'^{(l)})]) \right) , \quad (5.10)$$

where $\mathbf{e}_i : \mathbb{Z} \rightarrow \mathbb{R}^t$ is the embedding of each class token via the embedding layer of the generator. The logits $a_{ij}^{(l,k)}$ are the normalized Attention weights of the bilinear function

that monitor the dyadic interaction between classes in the event embeddings in layer l and head k defined in Equation (3.1). $\mathbf{W}^{(l)}$ in Equation (5.9) is the learnable multi-head projector at layer l defined in Equation (3.2). The output of the composition of all layers via the composition of L functionals, $\bigcirc_{l=0}^L \Phi^l := \phi_{w_L} \circ \dots \circ \phi_{w_0}[\mathbf{q}_i^{(l=0)}] \in \mathbb{R}^{m \times t}$, goes into a Layer Normalization layer where $\Phi^l = \mathbf{q}_i'^{(l)} + \mathcal{F}[\mathbf{LN}(\mathbf{q}_i'^{(l)})]$ as shown in Equation (5.10).

One input to the generator is the embedded labels, which can be considered rigid token embeddings that will be learned as a global representation bias of each sensor. As sensor conditions change for each event as a set, having merely class embeddings, as used in conditional GANs [83], is insufficient because the context-based information will not be learned. Thus, the generator samples from a per-event shared distribution at each event as random degrees of freedom (Rdof). Rdofs are random samples from a shared Normal distribution for each class, $\mathbf{r}_i \sim \mathcal{N}(0, 1)$, that introduces four-dimensional learnable degrees of freedom for the generator, see Equation (5.8). This way, I ensure that the generator is aware of intra-event local changes, culminating in having an intra-event correlation among the generated images. Rdof can be interpreted as both perturbation [271] to the token embeddings and an event-level segment embedding [39], which can enhance the diversity of the generated images.

5.4.3 Intra-Event Aware Loss

Motivated by Self-Supervised Learning [283, 247], to transfer the intra-event contextualized knowledge of the discriminator to the generator in an explicit way, I introduce an Intra-Event Aware (IEA) loss, depicted in Figure 5.7, for the generator that captures class-to-class relations,

$$\ell_{\text{IEA}}(\mathbf{x}_r, \mathbf{x}_f) = \sum_{i,j} D_{\text{KL}} \left(\sigma \left(\mathbf{h}(x_i^{(r)})^\top \mathbf{h}(x_j^{(r)}) \right) \middle\| \sigma \left(\mathbf{h}(x_i^{(f)})^\top \mathbf{h}(x_j^{(f)}) \right) \right), \quad (5.11)$$

where $\mathbf{x}_r = \{x_i^{(r)}\}_{i=1}^m$ is the set of real images, and $\mathbf{x}_f = \{G(z^i, y^i, r^i) = x_i^{(f)}\}_{i=1}^m$ the set of generated images. The softmax function, $\sigma : \mathbb{R}^m \rightarrow [0, 1]^m$, normalizes the dot-product self-attention between the image embeddings. The map $\mathbf{h} : \mathbb{R}^k \rightarrow \mathbb{S}^n$ is the unit hypersphere projection of the discriminator. Therefore, the dot product is equivalent to the cosine distance. $D_{\text{KL}}(\cdot \parallel \cdot)$ is the Kullback-Leibler (KL) divergence [284] which takes two $m \times m$ matrices that have values in the closed unit interval (due to the softmax function). Hence, having a KL divergence is natural here as one wants to compare one probability density with another in an event. I also tested other distance functions reported in Section 5.6. By considering the linear interaction [285] between every sample in an event and assigning a weight to their similarity, the generator mimics the fine-grained class-to-class relations within each event and incorporates this information in its RRM module as shown in Figure 5.6a.

The KL divergence, a fundamental concept in information theory, has diverse interpretations that enrich our understanding of IEA loss. Primarily, the KL divergence is a measure of how much the generator's intra-event understanding Q deviates from the true distribution P of sensor-by-sensor relationships, or in simpler terms, how much P and

Q differ in a context where P is the true distribution. This lack of symmetry is evident in several ways. For instance, it represents the expected “surprise” when observing data with distribution P , assuming falsely that the intra-event distribution is Q . In hypothesis testing, it corresponds to the expected evidence for P over Q when P is true. In relation to Maximum Likelihood Estimators (MLEs), when P is an empirical distribution of data, $D_{KL}(P||Q)$ is minimized when the generated intra-event distribution Q is the MLE for P . Moreover, from the information theory perspective, it denotes the inefficiency or extra bits used when compressing a data source with distribution P using a code optimized for Q . Lastly, in the domain of Game Theory, it can indicate potential winnings when one understands the true distribution while another party (the generator) operates under false assumptions.

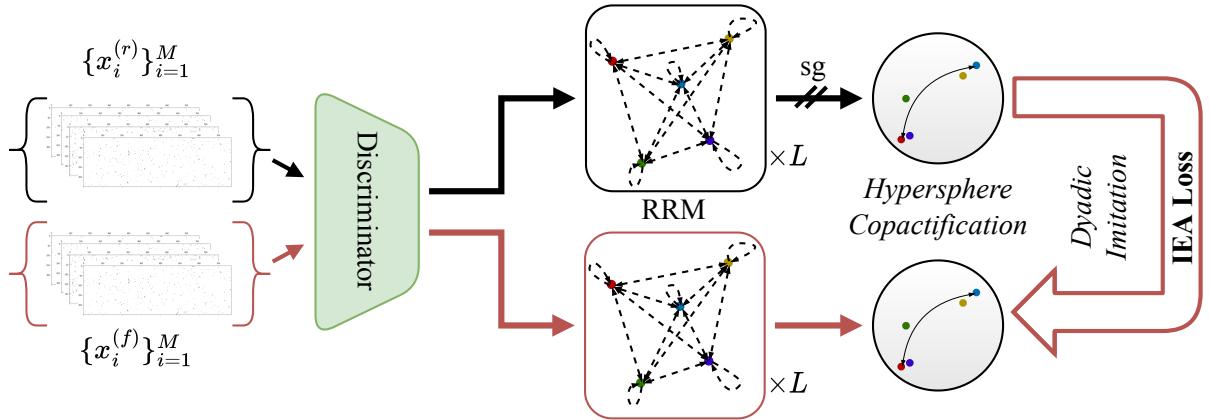


Figure 5.7: IEA-loss imposes a pair-wise fine-grained class-to-class imitation force for the generator. sg indicates that for the discriminator gradients are stopped and only the generator’s gradients will be updated.

Upon minimizing it for the generator (having the stop-gradient for the discriminator), it is putting a discriminator-supervised penalizing system over the intra-event awareness of the generator by encouraging it to look for more detailed dyadic connections among the images and be sensitive to even slight differences. Ultimately, it will maximize the consensus of data points on two unit hyperspheres of real images and generated image embeddings.

5.4.4 Uniformity Loss

The other crucial loss function comes from contrastive representation learning. With the task of learning fine-grained class-to-class relations among the images, I also want to ensure the feature vectors have as much hyperspherical diversity as possible. Thus, by imposing a uniformity condition over the feature vectors on the unit hypersphere, they preserve as much information as possible since the uniform distribution carries a high entropy. This idea stems from the Thomson problem [286], where a static equilibrium with minimal potential energy is sought to distribute N electrons on the unit sphere in the evenest manner.

It encourages hyperspherical diversity by pursuing the following learning objective for n d -dimensional vectors $\mathbf{H}_n = \{\mathbf{h}_1, \dots, \mathbf{h}_n \in \mathbb{R}^d\}$,

$$\min_{\{\mathbf{h}_1, \dots, \mathbf{h}_n \in \mathbb{S}^{d-1}\}} \{E_s(\mathbf{H}_n) := \sum_{i=1}^n \sum_{j=1, j \neq i}^n K_s(\mathbf{h}_i, \mathbf{h}_j)\} \quad (5.12)$$

where $h_i \equiv h(x_i)$, is the i -th vector projected onto the unit hypersphere $\mathbb{S}^{d-1} = \{\mathbf{h} \in \mathbb{R}^d \mid \|\mathbf{h}\| = 1\}$. $K_s(\cdot, \cdot)$ models the interaction between two vectors, typically following Riesz s -kernel function

$$K_s(\mathbf{h}_i, \mathbf{h}_j) = \begin{cases} \rho(\mathbf{h}_i, \mathbf{h}_j)^{-s}, & s > 0 \\ \log(\rho(\mathbf{h}_i, \hat{\mathbf{h}}_j)^{-1}), & s = 0 \\ -\rho(\mathbf{h}_i, \mathbf{h}_j)^{-s}, & s < 0 \end{cases} \quad (5.13)$$

where $\rho(\cdot, \cdot)$ is defined to measure the geodesic similarity on the unit hypersphere. In general, I can use either $\rho(\mathbf{h}_i, \mathbf{h}_j) = \|\mathbf{h}_i - \mathbf{h}_j\|_2$ (the standard Riesz s -kernel [47, 272]) or hyperspherical geodesic distance (angular distance) $\rho(\mathbf{h}_i, \mathbf{h}_j) = \arccos(\mathbf{h}_i^\top \mathbf{h}_j)$. Minimizing this pairwise energy sum asymptotically corresponds to the uniform distribution on the hypersphere [287]. To do this, I use the Riesz s -kernel as an auxiliary loss function,

$$\mathcal{L}_{\text{uniform}}(x; s) = \log \mathbb{E}_{x_i, x_j \sim p_{\text{event}}} [\exp(s \|\mathbf{h}(x_i) - \mathbf{h}(x_j)\|_2^2)]. \quad (5.14)$$

Upon minimizing this loss for the discriminator, it tries to maintain a uniform distance among the samples that are not well-clustered and thus not similar. In other words, eventually, we want to reach a maximum geodesic separation incorporating the Riesz s -kernel with $s = -2$ as a measure of geodesic similarity, to preserve maximal information over the Hypersphere.

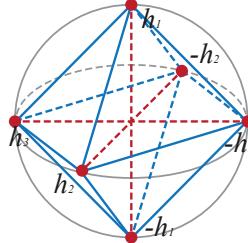


Figure 5.8: Uniformity over a 3-dimensional cross-polytope immersed in a unit hypersphere

By ensuring uniform distribution on the hypersphere for $2d + 2$ vectors in \mathbb{S}^d , one obtains a cross-polytope as described in [288]. The illustration in Fig. 5.8 showcases a 3-dimensional cross-polytope within \mathbb{S}^2 (consisting of 6 vectors). Supposing one has a set of unit-vectors with $d + 1$ vectors: $\{\mathbf{h}_1, \dots, \mathbf{h}_{d+1} \in \mathbb{S}^d\}$. By incorporating vectors in the opposite direction to this set, I generate a new collection comprising $2d + 2$ vectors: $\{\mathbf{h}_1, \dots, \mathbf{h}_{d+1}, -\mathbf{h}_1, \dots, -\mathbf{h}_{d+1} \in \mathbb{S}^d\}$. Aiming for uniformity on the hypersphere for vectors in this expanded set translates to pursuing orthogonality between the initial $d + 1$ vectors. This observation establishes a profound link between hyperspherical uniformity and vector orthogonality.

Inspired by [289], in order to examine the hyperspherical uniformity from a statistical test perspective uniformity, one can use the Sobolev test [290, 291, 292]. Using independent, identically distributed samples $\mathbf{h}_1, \dots, \mathbf{h}_n \in \mathbb{S}^{d-1}$ of a unit random vector \mathbf{h} , one can define the challenge of verifying uniformity on the hypersphere as the examination of the hypothesis $\mathcal{H}_0 : \mathbf{P} = \text{Uniform}(\mathbb{S}^{d-1})$ against its alternative $\mathcal{H}_1 : \mathbf{P} \neq \text{Uniform}(\mathbb{S}^{d-1})$, with \mathbf{P} signifying \mathbf{h} 's probability distribution. The underlying principle behind the Sobolev test is the transformation of the hypersphere \mathbb{S}^{d-1} into the Hilbert space $L^2(\mathbb{S}^{d-1}, \mu)$ of square-integrable functions with finite integral on \mathbb{S}^{d-1} . This is achieved via a function $t : \mathbb{S}^{d-1} \rightarrow L^2(\mathbb{S}^{d-1}, \mu)$ ensuring that when \mathbf{h} follows a uniform distribution on the hypersphere, the expectation value of $t(\mathbf{h})$ is zero. Denoting ϵ_k ($p_{d,k} = \dim \epsilon_k$), being associated with the k -th distinct eigenvalue of the Laplacian, the space of eigenfunctions from \mathbb{S}^{d-1} to \mathbb{R} . A mapping $t_k : \mathbb{S}^{d-1} \rightarrow \epsilon_k$ can be expressed as $t_k(\mathbf{h}) = \sum_{i=1}^{p_{d,k}} g_{i,k}(\mathbf{h}) g_{i,k}$ where $\{g_{i,k}\}_{i=1}^{p_{d,k}}$ creates an orthogonal basis of ϵ_k . With $\{v_k\}_{k=1}^{\infty}$ being a series satisfying $\sum_{k=1}^{\infty} v_k^2 p_{d,k} < \infty$, the function defined by $\mathbf{h} \rightarrow t(\mathbf{h}) := \sum_{k=1}^{\infty} v_k t_k(\mathbf{h})$ is a transformation from \mathbb{S}^{d-1} into the Hilbert space $L^2(\mathbb{S}^{d-1}, \mu)$ of square integrable real functions real functions over \mathbb{S}^{d-1} with respect to the uniform measure μ . The Sobolev test, refutes \mathcal{H}_0 for large values of the following test statistic,

$$S_n := \frac{1}{n} \sum_{i,j}^n \sum_{k=1}^{\infty} v_k^2 \langle t_k(\mathbf{h}_i), t_k(\mathbf{h}_j) \rangle \quad (5.15)$$

where $\langle f, g \rangle := \int_{\mathbb{S}^{d-1}} f(\mathbf{h}) g(\mathbf{h}) d\mu(\mathbf{h})$ denotes the inner product on $L^2(\mathbb{S}^{d-1}, \mu)$. For even values of k , let $v_k = 0$ and for odd k , set $v_k = (\pi k)^{-1}$. Then a variant of the Sobolev test, the Ajne test [293, 294], employs the following simplified test statistic.

$$A_n = \frac{n}{4} - \frac{1}{n\pi} \sum_{1 \leq i < j \leq n} \arccos(\mathbf{h}_i^\top \mathbf{h}_j) \quad (5.16)$$

This test rejects \mathcal{H}_0 when the A_n values are substantially large. Its relevance to Eq.5.14 is evident as both of them derive from pairwise relationships. Specifically, minimizing the this equation in relation to $\{\mathbf{h}_i\}_{i=1}^n$ corresponds directly to Eq.5.13 with a Riesz s -kernel, where $s = -2$ which is our Uniformity loss Equation (5.14).

This loss is beneficial for capturing the exact distribution of the mean occupancy distribution and balancing the inter-class pulling force of the Relational Reasoning module. As a result, not only does it help generate more diverse and varied outputs, but it also can prevent issues such as mode collapse or overfitting.

5.4.5 Model Details and Hyperparameters

For training and evaluation, 40 000 and 10 000 Monte Carlo simulated [8] events are used respectively. The data in each event consists of 40 grey-scale 256×768 zero-padded images. They are zero-padded on both sides from their original size of 250×768 to be divisible by 16 for training purposes.

To capture the intra-event mutual information among the images using the RRM and approximate the concept of an event, I have to sample properly at each iteration. Hence, a

one-sample per class sampler is used in data loading. Using this sampler, one can ensure that in each event, one has 40 unique classes of images from all 40 sensors that belong to the same event in the Monte-Carlo simulation. All hyperparameters are chosen based on the model’s stability and performance upon the evaluation set. The learning rates for the Generator and Discriminator are 5×10^{-5} . The Relational Reasoning Module of the Generator has two heads and one layer of non-spectrally normalized message propagation with an embedding dimension of 128 and ReLU non-linearity. The input to the generator’s RRM is embedded class tokens mixed with 4 random degrees of freedom by a spectrally normalized linear layer.

For the Discriminator, the RRM has four heads with one layer of spectrally normalized message propagation with the embedding dimension 1024 as the hypersphere dimension and ReLU non-linearity. All Generator and Discriminator modules use Orthogonal initialization [88]. For the IEA-loss in Equation (5.11), the coefficient $\lambda_{\text{IEA}} = 1.0$, defined in Algorithm 1 gives the best result. The most stable contribution of the Uniformity loss, defined in Equation (5.14) and Algorithm 1, is with $\lambda_{\text{uniform}} = 0.1$. For the backbone of both the discriminator and the generator, BigGAN-deep [82] with a non-local block at channel 32 for the discriminator only is used. Since in GAN training, there is no meaningful way to define a minimal loss, my stopping point is the divergence of the Frechet Inception Score (FID) [22], which is significantly correlated with the quality of other metrics.

5.5 Evaluation Results: IEA-GAN vs SOTAs

This study showcases a series of analyses and evaluations that demonstrate the performance of IEA-GAN in generating ultra-high-granularity detector responses of the Pixel Vertex Detector, consisting of over 7.5 million pixel channels. Furthermore, for the first time, this finding reveals that the FID [22] and KID [23] metrics for detector simulation is a very versatile estimator in conjunction with the marginal distributions (mentioned in Chapter 2), and is associated with the other image level metrics. I show that by using IEA-GAN, one is able to capture the underlying distributions such that one can generate and amplify detector response information with a very good agreement with the Geant4 distributions. I also found out that the SOTA models in high-resolution image generation even with an in-depth hyperparameter tuning analysis do not perform well in comparison.

For evaluation, I have two categories of metrics: image level and physics level. As I am interested in having the best pixel-level properties, diversity, and correlation of the generated images simultaneously while adhering to minimal generator complexity due to computational limitations, choosing the best iteration to compare results is challenging. Hence, I chose the models’ weights with the best FID for all comparisons.

This study compares IEA-GAN with three other models and the reference, which is the Geant4-simulated [17] dataset. The baselines are the SOTA in conditional image generation: BigGAN-deep [82] and ContraGAN [77]. I also compare IEA-GAN with the previous works

Algorithm 1 Intra-Event Aware GAN

Require: generator and discriminator parameters θ_G, θ_D , Intra-Event-aware coefficient λ_{IEA} , Uniformity coefficient λ_{uniform} and hyperparameter s , Adam [295] hyperparameters α, β_1, β_2 , event size M , number of discriminator iteration steps per generator iteration N_D

```

1: for number of training iterations do
2:   for  $t = 1, \dots, N_D$  do
3:     sample  $\{z^i\}_{i=1}^M \sim p(z)$ 
4:      $\{x^i, y^i\}_{i=1}^M \sim p_{\text{event}}(x, y)$ ,  $\{r^i\}_{i=1}^M \sim p_{\text{Rdof}}(z)$  ▷ Event Sampling
5:     for  $i = 1, \dots, M$  do
6:        $\ell_{D_{\text{hinge}}}^{(i)} \leftarrow \ell_{D_{\text{hinge}}}(x^{(i)}; G(z^i, y^i, r^i))$ 
7:     end for
8:      $\mathcal{L}_{D_{\text{hinge}}} \leftarrow \frac{1}{M} \sum_{i=1}^M \ell_{D_{\text{hinge}}}^{(i)}$ 
9:      $\mathcal{L}_{\text{uniform}} \leftarrow \mathcal{L}_{\text{uniform}}(x; s)$  ▷ The Uniformity Loss
10:     $\mathcal{L}_{2C}^{\text{real}} \leftarrow \frac{1}{M} \sum_{i=1}^M \ell_{2C}(x^i, y^i)$ 
11:     $\theta_D \leftarrow \text{Adam}(\mathcal{L}_{D_{\text{hinge}}} + \lambda_{2C} \mathcal{L}_{2C}^{\text{real}} + \lambda_{\text{uniform}} \mathcal{L}_{\text{uniform}}, \alpha, \beta_1, \beta_2)$ 
12:   end for
13:   sample  $\{z^i\}_{i=1}^M \sim p(z)$ 
14:   sample  $\{r^i\}_{i=1}^M \sim p_{\text{Rdof}}(z)$  ▷ Event Sampling
15:   for  $i = 1, \dots, M$  do
16:      $\ell_{G_{\text{hinge}}}^{(i)} \leftarrow \ell_{G_{\text{hinge}}}(G(z^i, y^i, r^i))$ 
17:   end for
18:    $\mathcal{L}_{G_{\text{hinge}}} \leftarrow \frac{1}{M} \sum_{i=1}^M \ell_{G_{\text{hinge}}}^{(i)}$ 
19:    $\mathcal{L}_{\text{IEA}} \leftarrow \frac{1}{M} \sum_{i=1}^M \ell_{\text{IEA}}(G(z^i, y^i, r^i), x^i)$  ▷ The Intra-Event Aware Loss
20:    $\mathcal{L}_{2C}^{\text{fake}} \leftarrow \frac{1}{M} \sum_{i=1}^M \ell_{2C}(G(z^i, y^i, r^i), y^i)$ 
21:    $\theta_G \leftarrow \text{Adam}(\mathcal{L}_{G_{\text{hinge}}} + \lambda_{2C} \mathcal{L}_{2C}^{\text{fake}} + \lambda_{\text{IEA}} \mathcal{L}_{\text{IEA}}, \alpha, \beta_1, \beta_2)$ 
22: end for

```

on the PXD image generation task: PE-GAN [6] and WGAN-gp [172]¹.

5.5.1 Neural Network-based Metrics: FID and KID

To compute the FID and KID scores, based on the recent Clean-FID project [296], I entirely fine-tuned the Inception-V3 [297] model on the PXD images, as the PXD images are very different from the natural images used in their initial training. The downstream task for the fine-tuning was multi-class classification, involving 40 different sensors with which it acquired the ability to discriminate sensors and their corresponding data manifold. The former FID implementations use the fixed-width bilinear interpolation which is independent of the resizing ratio. In contrast, the Clean-FID follows standard signal processing principles and adaptively stretches the filter to prevent aliasing [296]. This process can be done for any other detector dataset. FID measures the similarity of the generated samples' representations to those of samples from the real distribution. Given large sampling statistics, for each hidden activation of the Inception model, the FID evaluates the Fréchet distance, also known as Wasserstein-2 distance, between the first two moments of the activation distributions.

¹Only for FID.

As demonstrated to be useful and practical in the natural image analysis domain [298], FID performs well in terms of discriminability, diversity, and robustness, despite only modeling the first two moments of the distributions in the feature space. The lower the FID score, the more similar the distributions of the real and generated samples are. Kernel Inception Distance (KID) is another metric similar to FID, used for evaluating the quality of generative models. Unlike FID, KID uses a kernel two-sample test, which provides an unbiased estimate of the distance between distributions and is more robust to small sample sizes.

Table 6.3 demonstrates that generated images by IEA-GAN have the lowest FID and KID score compared to the other models and outperform them by 42%. This indicates that IEA-GAN is able to generate synthetic samples that are much closer to the target data than the samples generated by the other models.

Table 5.2: FID and KID comparison between models (all models in the benchmark are highly tuned to the current problem and dataset), averaged across six random seeds. The lower the FID and KID the better the image quality and diversity.

	WGAN-gp	BigGAN-deep	ContraGAN	PE-GAN	IEA-GAN
FID	12.09	4.40 ± 0.88	3.14 ± 0.74	2.61 ± 0.91	1.50 ± 0.16
KID	0.0096	0.0031 ± 0.0001	0.0015 ± 0.0002	0.0021 ± 0.0004	0.0010 ± 0.0002

In order to qualitatively analyze and interpret the FID flow during training, first I showcase the change of FID value with respect to the occupancy and charge distribution at different stages of the training as depicted in Figure 5.9.

Additionally, in Table 5.3 I demonstrate the sensitivity of FID to various types of image distortions directly linked to the underlying physics recorded by the corresponding sensor. This is achieved by introducing controlled changes or 'jitters' to the images and tracking their impact on the FID score.

5.5.2 Marginal Distributions

At the pixel level, there are the pixel intensity distribution, occupancy distribution, and mean occupancy. The pixel intensity distribution defines the distribution of the deposited charge of the background hits. The occupancy distribution and the pixel intensity distribution are evaluated over all sensors of a given number of events, while the mean occupancy corresponds to the mean value of sparsity across events for each sensor. This pixel-level information is essential since upon physics analysis via the basf2 software [16], when one wants to use the images and overlay the extracted information on the signal hits, the sparsity of the image defines the volume of the background hits on each sensor. The pixel intensity distribution, the occupancy distribution, as well as the mean occupancy per sensor are shown in Figure 5.10. The distributions for the IEA-GAN model show the closest agreement with the reference. The bimodal distribution of the occupancy comes from the

Image Jitterings	FID
None	0
Random Masking (dead zones)	14.58
Random Noise	87.23
Random Rotation (30 degrees)	23.69
Random Rotation (10 degrees)	2.81
Random Translation (0.1, 0.1)	1.99
Random Shear (10, 10)	23.53
Random Zoom	9.06
High Intensity smearing	3.16
Low Intensity smearing	47.24

Table 5.3: FID Score after Different Jittering Methods applied to the images

topology of the detector as the sensors are not in a cylindrical shape like a calorimeter but in an annulus shape in two layers. This indicates how challenging generating this detector signature is concerning both its geometry and resolution. In order to capture the correct bi-modality of the occupancy distribution, the RRM and the Uniformity loss play an important role. By using the Uniformity loss in the discriminator, the generator is incentivized to produce samples that are not biased towards a particular mode or class, leading to a wider bimodal distribution of generated samples.

Moreover, by utilizing the RRM module that considers the inter-dependencies and correlations among the samples within an event, the IEA-GAN exhibits a superior consistency with both high-intensity hits, which enhances the diversity of generated samples in regions with lower occurrence rates.

Along with all these image-level metrics, one also needs an intra-event sensitive metric. All the above metrics are equivariant under permutation between the samples among events. In other words, if I randomly shuffle the samples between events while fixing the sensor number, all the discussed metrics are unchanged. Hence, a metric that looks at the context of each event individually in its event space and goes beyond the sample space is needed. Ergo, I compute the Spearman’s correlation between the occupancy of the sensors along the population of generated events,

$$r_s = \mathbf{Corr}_p(R(\biguplus_{i=1}^{M=40} (\|x_i\|_0)), R(\biguplus_{i=1}^{M=40} (\|x_i\|_0))),$$

where $R(\cdot)$ is the rank operator, a function that assigns a rank to each number in a list as in the definition of Spearman’s correlation, and $\mathbf{Corr}_p(\cdot)$ is the Pearson Correlation function. \uplus is the disjoint union operator that symbolizes the concatenation operation. The norm with subscript 0, denoted by $\|\cdot\|_0$, is the L0 measure. It is a function that counts the number of non-zero elements in a vector. In this work, it is used to calculate the occupancy of the sensors, i.e., the number of non-zero elements in the sensor image x_i .

The coefficients by PE-GAN are random values in the range $[-0.2, 0.2]$, whereas IEA-GAN images show a meaningful correlation among their generated images. Even though the desired correlation is different from the reference, as shown in Figure 5.11, IEA-GAN understands a monotonically positive correlation for intra-layer sensors and a primarily negative correlation for inter-layer sensors.

In order to demonstrate that the learned correlation is actually meaningful, I incorporate the Mantel test [299, 300], which is a significance test of the correlation between two distance/correlation matrices excluding the diagonal part. The Mantel test works by comparing each pair of corresponding elements in the two matrices. The null hypothesis is that there's no relationship between the two sets of correlations, and the test statistic is a correlation coefficient. The significance of the observed correlation is evaluated using permutation testing. This involves randomly rearranging the elements of one matrix many times, recalculating the test statistic each time, and then seeing how extreme the observed test statistic is relative to this null distribution of test statistics. If the observed test statistic is very extreme, then the p-value is less than 0.05, and the null hypothesis is rejected. For IEA-GAN, the Mantel test results show a veridical correlation of 0.18 ± 0.02 with empirical p-value 0.0013. As the p-value is less than 0.05, the null hypothesis can be rejected, as there is significant evidence for the correlation between the two sets of matrices. This suggests that the sensor classes that are more correlated in the Geant4 samples tend to also be correlated in the generated ones by IEA-GAN. Whereas for PE-GAN, the Mantel test results show a veridical correlation of 0.002 with empirical p-value 0.96 in support of the null hypothesis.

5.5.3 Physics-Level Metric: Tracking

While image level metrics indicate the low-level quality of simulations, one must also confirm that the resulting simulations are reasonable physics-wise when the entire detector is considered as a whole. For this, a tracking analysis will be done to examine the Helix Parameter Resolutions (HPR). The quality of the tracking and HPRs directly impacts the precision and accuracy of the measurements. At the Belle II experiment, after each collision event, charged particle tracks propagating in vacuum in a uniform magnetic field move roughly along a helix path described by the five helix parameters $\{d_0, z_0, \phi_0, \omega, \tan \lambda\}$ with respect to a pivot point [16] (introduced in Chapter 2). These five helix parameters provide a comprehensive description of the trajectory of a charged particle moving in a uniform magnetic field, capturing its shape, orientation, and position in space relative to a reference or pivot point. The difference between the true and reconstructed helix parameters defines the resolution for the corresponding helix parameter. The track parameter resolution is affected by the number of hits, the hit intensity, and the underlying intra-event correlation. Understanding how the background effects impact the HPR can give us crucial insights into the overall performance of the detector and the quality of the data it produces. A better (lower) tracking resolution means that the path of a particle can be determined more precisely, which is critical for making accurate measurements and distinguishing between different particles or decay modes. This study utilizes the same event generation and track

reconstruction for the comparison, implying that the signal hits used in all simulations are essentially identical. Thus, the true track information is similar. The primary point of difference lies in the origin of the background. This distinct differentiation allows any disparities identified in the tracking parameter resolutions to be attributed largely to the different backgrounds and their generation origin, enabling a direct evaluation of the quality and performance of the IEA-GAN model in comparison to Geant4. First, let's see how important is the Intra-Event sensor correlation from the underlying tracking analysis.

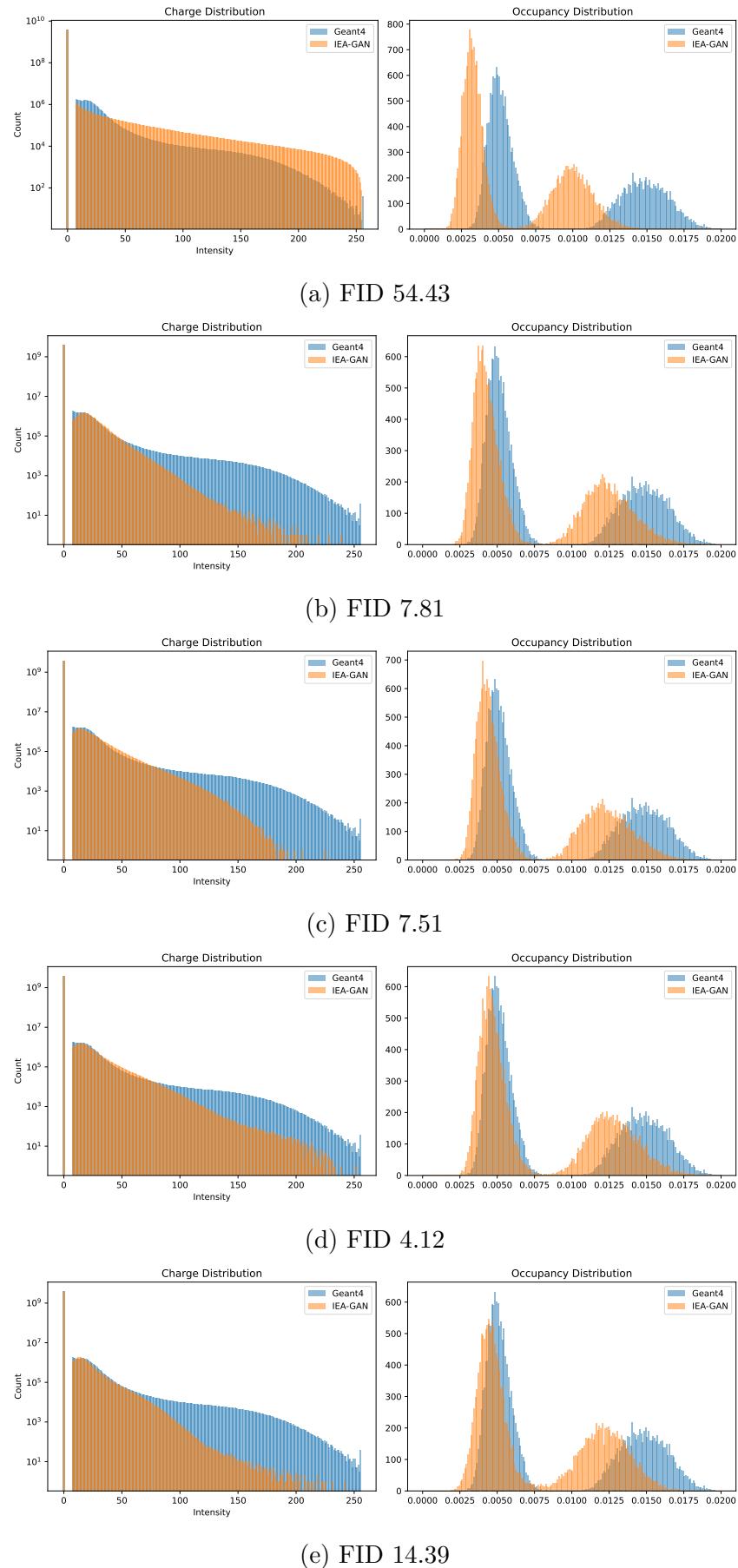


Figure 5.9: Different FID values with respect to the Charge and Occupancy distributions

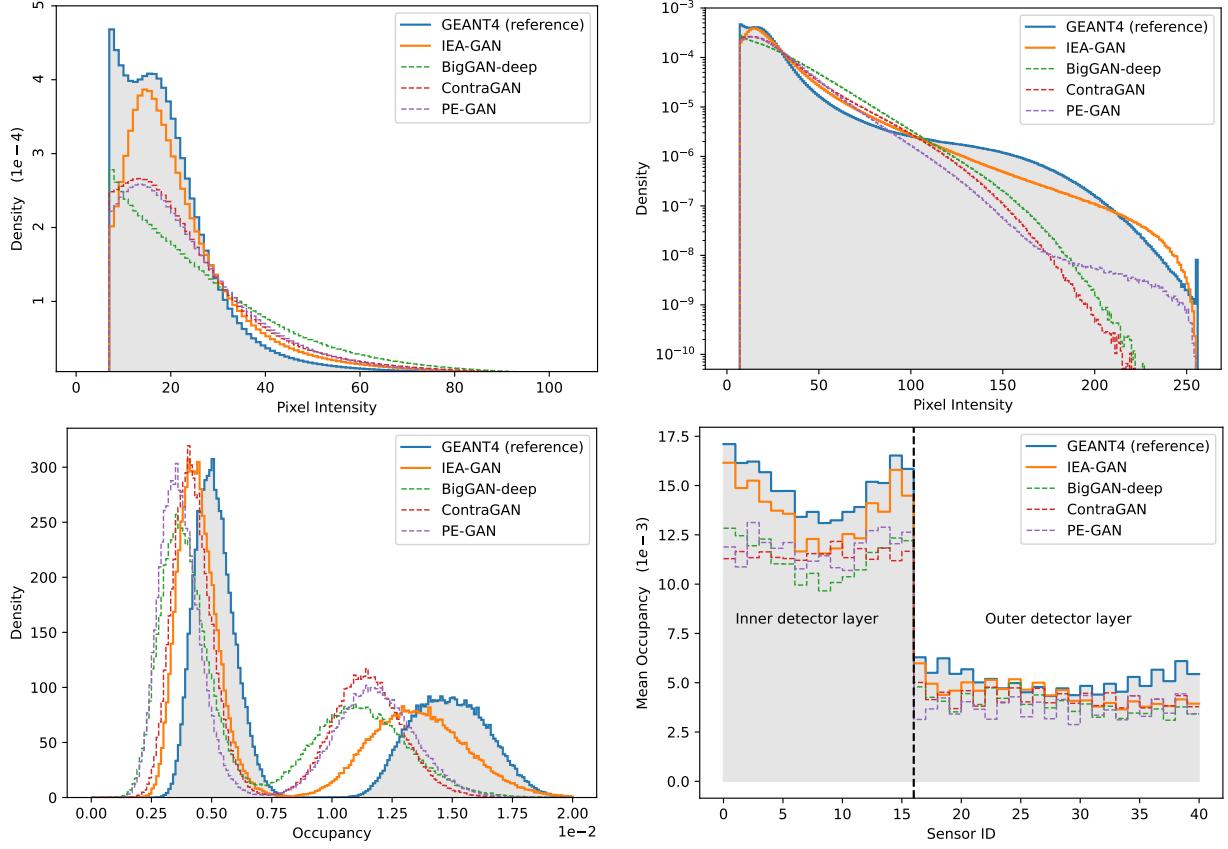


Figure 5.10: Pixel intensity distribution in linear (top left) and logarithmic scale (top right), the distribution of the occupancy (bottom left) and the mean occupancy per sensor (bottom right) for 10 000 events.

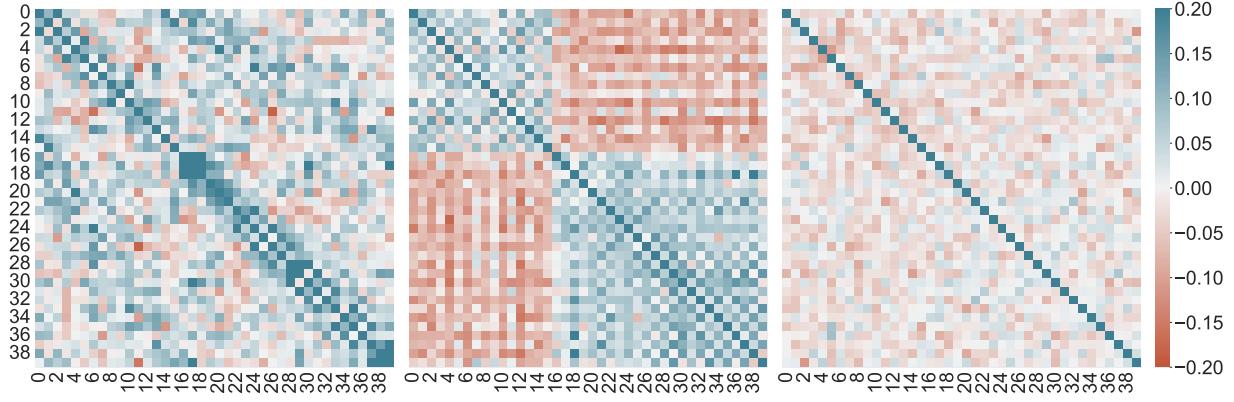


Figure 5.11: Spearman's correlation between the occupancy of Geant4 sensor images (left), and sensor images from IEA-GAN (center), sensor images from PE-GAN (right).

5.5.3.1 Physics Motivation for Intra-Event sensor correlation

As a physics motivation, I highlight the impact of the intra-event correlation by shuffling Geant4 samples. In other words, I show that in a physics analysis, the intra-event sensor-by-sensor correlation influences the performance of the tracking parameters. I examine the results by comparing the unbiased variance of the parameter resolutions and the 2-sample Kolmogorov–Smirnov test (KS test) between the shuffled and unshuffled Geant4 PXD background. The 2-sample Kolmogorov–Smirnov (KS) test is a non-parametric hypothesis test used to compare the distributions of two independent samples. Specifically, it tests the null hypothesis H_0 that the two samples are drawn from the same continuous distribution. The test statistic, D , is computed as the maximum absolute difference between the empirical cumulative distribution functions (ECDF) of the two samples. Formally, if $F_n(x)$ and $G_m(x)$ are the ECDFs of two samples of sizes n and m , respectively, the test statistic is defined as:

$$D = \max_x |F_n(x) - G_m(x)|$$

Under the null hypothesis H_0 , the expected value of D would be close to 0, while a larger value of D would indicate a significant difference between the two distributions, leading us to reject H_0 in favor of the alternative hypothesis H_1 which states that the two samples come from different distributions. The critical value or p-value for the test statistic can be obtained from the Kolmogorov distribution, enabling a formal test of the hypothesis.

The results for the high momentum tracks, with more than 0.4 GeV show that there is strong evidence that losing the intra-event sensor-by-sensor correlation would impact the resolution and thus the precision of the \mathbf{d}_0 , ϕ_0 and ω Helix parameters. For the \mathbf{z}_0 and $\tan \lambda$ parameter, there is no significant difference between the resolutions; however, the KS test for these parameters yields low p-values, indicating a high discrepancy between the shape of the two distributions.

Let's interpret the results in the context of each Helix parameter. For \mathbf{d}_0 impact parameter, the significant variance in resolution shows that the loss of correlation directly impacts how well one can measure the particle's closest approach to the origin in the transverse plane. Given that sensor-by-sensor correlations help to correctly associate track hits, losing it leads to a more spread out distribution of reconstructed values as shown in Figure 5.12. This could affect subsequent analyses, such as identifying primary and secondary vertices, especially in scenarios where particles have negligible deflection.

For the ϕ_0 parameter, the insignificant resolution variance difference and KS test result suggest that the lack of layer correlation doesn't significantly impact the distribution and precision of measurements of the azimuthal for high momentum tracks. The higher error in the variance of $\Delta\mathbf{z}_0$ in the shuffled data suggests that the lack of correlation between detector layers introduces more uncertainty in determining the longitudinal interaction point. High momentum tracks are less likely to deviate significantly in the z-direction. Combined with the significant KS test result, this indicates a fundamental difference in how particle trajectories are reconstructed in the z-direction without layer correlation. ω is a measure of the curvature of the particle's track, and it's inversely proportional to the particle's momentum. For high-momentum particles, one would expect the curvature to be

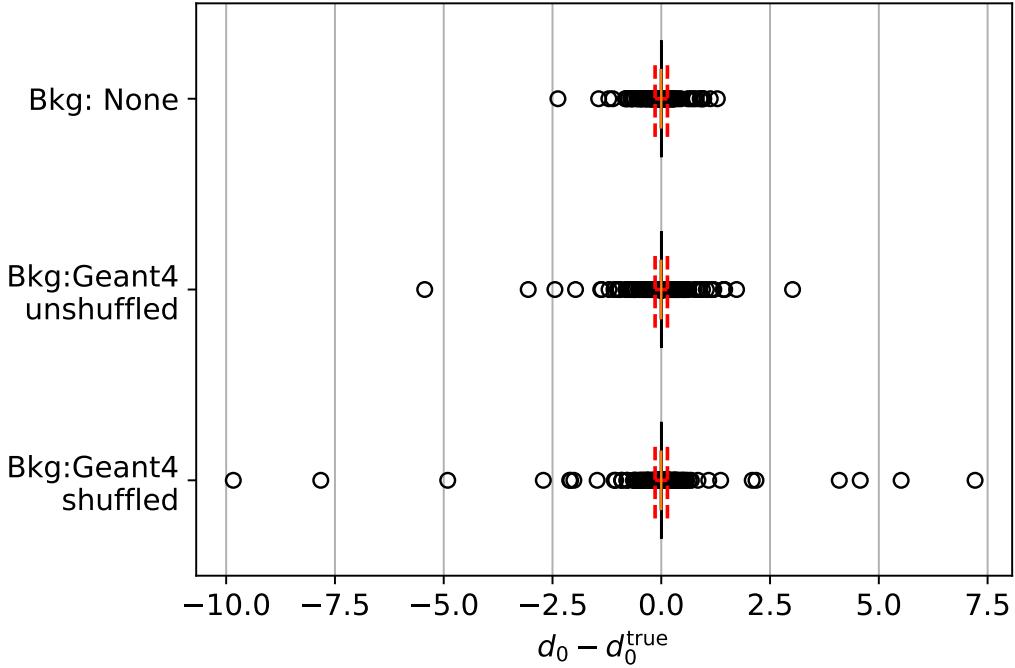


Figure 5.12: The boxplot for comparing the d_0 resolution in the presence of correlated background and uncorrelated background at high momentum regime. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.

smaller since higher-momentum particles travel more linearly. The variance for $\Delta\omega$ shows a slight discrepancy between the shuffled and unshuffled data, but the KS test doesn't show a significant difference. This indicates that while the overall distributions of the curvature resolution don't significantly differ, there's a minor difference in the precision with which the curvature is reconstructed, which could have implications for subsequent physics analyses that depend on accurate momentum information. Despite the insignificant resolution difference, the significant KS test result for $\Delta \tan \lambda$ suggests differences in the inclination distributions of high momentum tracks between the shuffled and unshuffled data. This might indicate that the inclination of the track, which is also related to the momentum in the longitudinal direction, is affected by the loss of correlation between layers and sensors.

5.5.3.2 IEA-GAN's Tracking Performance

This study compares IEA-GAN with PE-GAN for the resolutions of all five helix parameters as shown in Figure 5.13 and Table 5.5 for 5000 events for high momentum tracks ($P_T > 0.4$ GeV).

In the low momentum region, the resolution performance of the models is on par. My meticulous comparison revealed that the unbiased variance of these parameters, produced

Parameter	Unbiased Variance \pm error		KS statistic	p-value
	Shuffled Geant4	Unshuffled Geant4		
Δd_0	0.1343 ± 0.0007	0.0732 ± 0.0004	0.0067	0.7655
$\Delta \phi_0$	0.2158 ± 0.0011	0.1859 ± 0.0009	0.0066	0.7899
Δz_0	5.0076 ± 0.0253	4.9341 ± 0.0249	0.0152	0.0211
$\Delta \omega$	0.0010 ± 0.0001	0.0008 ± 0.0000	0.0138	0.0485
$\Delta \tan \lambda$	0.0388 ± 0.0002	0.0382 ± 0.0002	0.0167	0.0086

Table 5.4: Unbiased variance and KS test results for the shuffled and unshuffled Geant4 data across the 5 Helix parameters.

by the IEA-GAN model, approximates more closely to the Geant4 reference, outperforming the PE-GAN model in each instance. Moreover, the Kolmogorov-Smirnov test results further consolidated my findings, showing higher p-values for the IEA-GAN model, thus adhering more accurately to the Geant4 reference. Another interesting observation is that, in comparison with the shuffled Geant4, IEA-GAN shows a more significant KS test p-value for z_0 , ω , and $\tan \lambda$ resolutions and a more precise \mathbf{d}_0 reconstruction.

As a result, I observe a good agreement between the IEA-GAN and Geant4, both in the tail segments (variance) and precision of the resolutions where the largest difference between Geant4 and no background is found. Hence, not only does IEA-GAN demonstrate a close image level agreement with Geant4, but it maintains a proper reconstructed physical behavior during track reconstruction as well.

Model	Parameter	Unbiased Variance \pm error		KS statistic	p-value
		Model	Geant4		
PE-GAN	d_0	0.1709 ± 0.0009	0.0732 ± 0.0004	0.0156	0.0164
	ϕ_0	0.2207 ± 0.0011	0.1859 ± 0.0009	0.0120	0.1193
	z_0	6.9073 ± 0.0349	4.9341 ± 0.0249	0.0183	0.0029
	ω	0.0014 ± 0.0001	0.0008 ± 0.0001	0.0116	0.1425
	$\tan \lambda$	0.0579 ± 0.0003	0.0382 ± 0.0002	0.0179	0.0037
IEA-GAN	d_0	0.0762 ± 0.0004	0.0732 ± 0.0004	0.0104	0.2373
	ϕ_0	0.1905 ± 0.0010	0.1859 ± 0.0009	0.0109	0.1939
	z_0	5.1467 ± 0.0261	4.9341 ± 0.0249	0.0073	0.6814
	ω	0.0010 ± 0.0001	0.0008 ± 0.0001	0.0103	0.2537
	$\tan \lambda$	0.0412 ± 0.0002	0.0382 ± 0.0002	0.0068	0.7538

Table 5.5: Comparison of Unbiased Variance and KS test results for the PE-GAN and IEA-GAN models with the Geant4 reference across 5 Helix parameters for high momentum tracks.

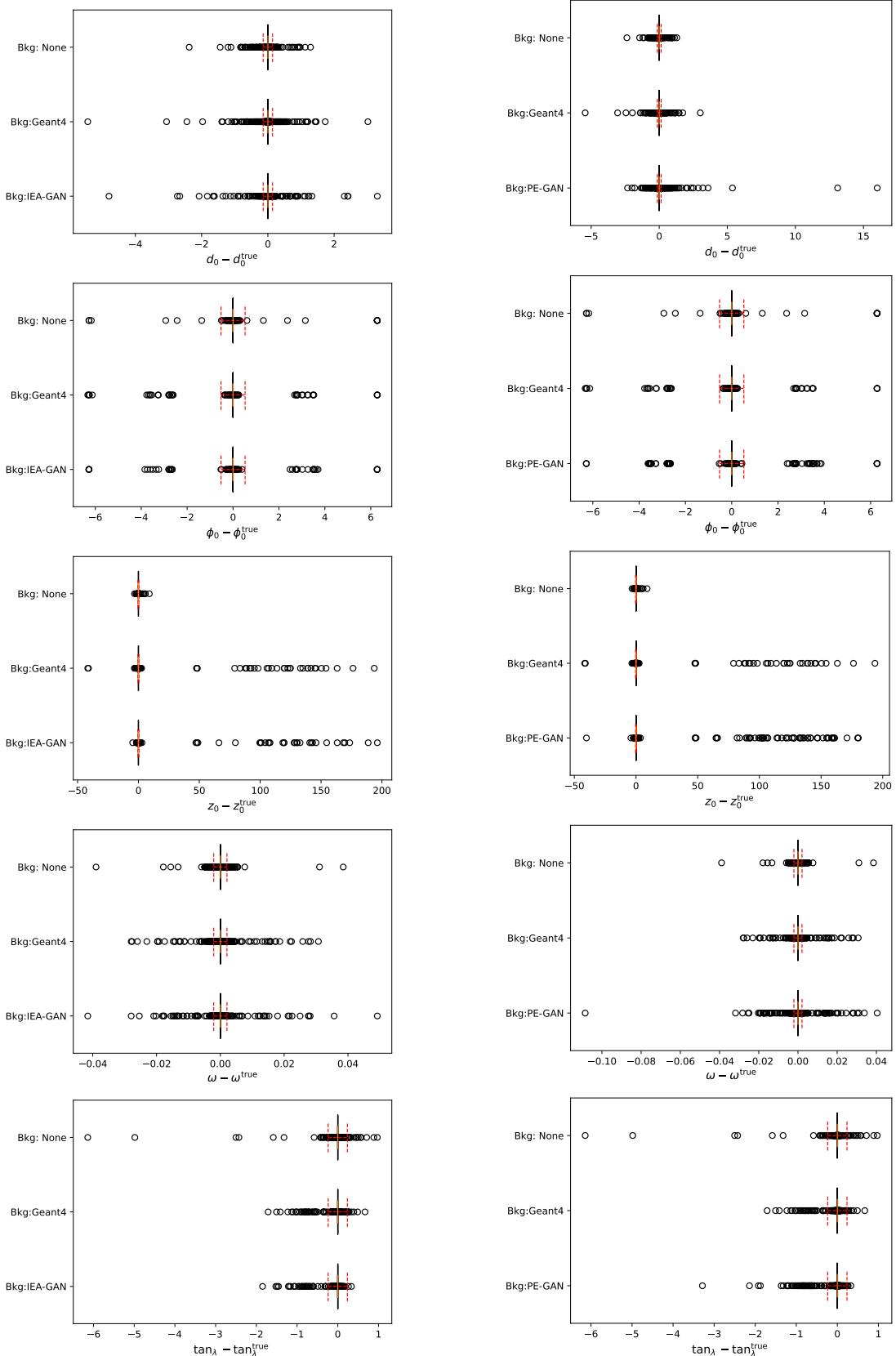


Figure 5.13: Helix parameter resolutions for d_0 (row 1), z_0 (row 2), ϕ_0 (row 3), ω_0 (row 4), and $\tan \lambda$ (row 5). For each parameter, the left figure corresponds to the IEA-GAN simulated background, and the right figure corresponds to the PE-GAN simulated background. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.

5.6 Ablation Studies and Things I Tried But Did Not Work

For the IEA-loss, I tested several losses in order to achieve the best stability, shown in Figure 5.14. Some of them have their own merits and downsides. I explored the KL divergence, the L1 loss, the Huber loss [301], and the L2 loss. KL divergence was more stable to capture differences between the real and fake self-similarities and more robust to outliers.

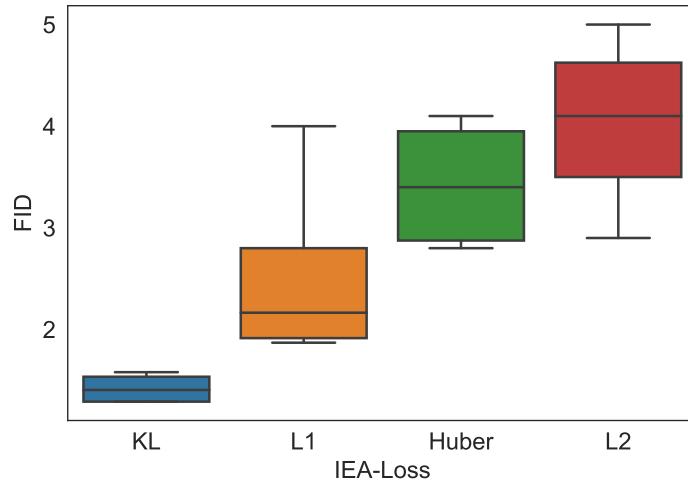


Figure 5.14: Comparison of the FID between different IEA-losses

I probed a range of coefficients for the IEA-loss and the Uniformity loss. For the KL divergence as the IEA-loss, I tried the values $\{0.1, 1, 5, 10\}$ and selected 1. For the L1 loss, as well as the IEA-loss, the best λ_{IEA} value is 10. For the Uniformity loss, I probed the values $\{0.01, 0.1, 0.5, 0.75, 1, 5, 10\}$ and selected 0.1. Moreover, IEA-GAN, without the IEA-loss and Uniformity loss suffers from the lack of agreement maximization penalty for the generator and information maximization for the discriminator. This study shows that having either of these losses without the other causes training instability, divergence, and lower fidelity as shown in Table 5.6.

Table 5.6: FID comparison between IEA-GAN, IEA-GAN with RRM only, IEA-GAN with Uniformity loss only, and IEA-GAN with both IEA-loss, averaged across six random seeds.

	IEA-GAN	Only RRM	RRM with Uniformity	RRM with IEA-loss
FID	1.50 ± 0.16	2.74 ± 0.62	2.29 ± 0.14	3.42 ± 0.52

For the hypersphere dimension, I probed the values $\{512, 768, 1024, 2048\}$ and selected 1024. For dimensions smaller than 512 the discriminator fails to converge. I also changed

the position of the hypersphere projection layer and put it before and after the Multi-head attention. The best position for the hypersphere projection is after the Multi-head attention and two layers of MLP. Moreover, for the hypersphere projection, I also tried an inverse Stereographic Projection $h : \mathbb{R}^N \rightarrow \mathbb{S}^N / \{p\}$ with p as a north pole on the n-sphere [302] instead of L2 compactification. This map is conformal thus it locally preserves angles between the data points. The results were more stable but the average FID was better with L2 compactification as shown in Table 5.7.

Table 5.7: FID comparison between two different Hypersphere projections for IEA-GAN’s discriminator, averaged across six random seeds.

	L2 compactification	Inverse Stereographic projection
FID	1.50 ± 0.16	2.01 ± 0.07

Inside the RRM I tried a GeLU [303] non-linearity instead of ReLU and the result was in favor of the latter. I also put the layer normalization before and after the Multi-head Attention. The pre-norm version seems to be much more stable and adaptable to GAN training intricacies. Another observation related to RRM is the weight normalization of the linear layers. I observed that for the discriminator spectrally normalized MLPs show the best results. For the Generator, applying Spectral Normalization to the linear layers destabilizes the training. My observation regarding the effect of RRM over the generator’s label embedding shows that without it, the RRM in the discriminator becomes also unstable and the training diverges very early.

For the random degrees of freedom (Rdof), first, I utilized the random vectors that are fed to the generator and applied the RRM on top of it. However, the FID did not reach values below 20, and there was no correlation. Hence, I introduced separate random sampling for event generation for which I probed dimensions $\{2, 4, 8, 16, 32\}$. 4 degrees of freedom was the most optimal choice. I observed that as the dimension of Rdof increases, the intra-event correlation fades away between the generated images. I also checked the Uniform distribution for event Rdof sampling, which did not lead to any stable result. Several ways to fuse the Rdof to the class embeddings were tested such as learnable neural network layer (matrix factorization), concatenating, summing, and having an MLP with non-linear part, but eventually chose a learnable neural network layer (matrix factorization) for the feature mixing layer.

I also looked at different combinations of learning rates for G and D. Using TTUR [22] regime results in a severe mode collapse. Thus, I used the same learning rates for both G and D. I swept through $\{1 \times 10^{-5}, 2.5 \times 10^{-5}, 5 \times 10^{-5}, 7.5 \times 10^{-5}, 1 \times 10^{-4}\}$ and selected 5×10^{-5} . For the backbone model, the shallow version of BigGAN-deep, BigGAN, leads to mode-collapse, therefore, I chose BigGAN-deep.

5.7 Conclusion and Discussion

This study has proposed a series of new methods for ultra-high-resolution, fine-grained, correlated PXD detector background response generation and conditional sampling. First, I argued that PXD background generation can be categorized as a fine-grained data analysis task with meta-categories being the sensor labels, and subordinate categories being the fine features such as various amounts of occupancy or different background types. Then PE-GAN was introduced as a proof-of-concept of having sensor-number conditioning. However, despite its performance in conditional sampling, it did not have any intra-event understanding.

As a result, IEA-GAN was proposed to capture the dyadic class-to-class relations and exhibit an explainable (weak) intra-event correlation among the generated detector images while all other models fail to capture any correlation. To achieve this, novel components, the Relational Reasoning Module (RRM) and the IEA-loss, with the Uniformity loss were introduced. The RRM introduced a self-supervised relational contextual embedding for the samples in an event, which is compatible with GAN training policies, a task that is very challenging. RRM dynamically clusters the images in a collider (PXD) event based on their inherent correlation culminating in approximating a collision event. The IEA-loss, a discriminator-supervised loss, helps the generator reach a consensus over the discriminator's dyadic relations between samples in each event. Finally, I have demonstrated that the Uniformity loss plays a crucial role for the discriminator in maximizing the homogeneity of the information entropy over the embedding space, thus helping the model to overcome mode-collapse and to capture a better bi-modality of generated occupancy. As a result, an improvement in all metrics compared to the previous SOTA occurs. Let's quickly review the results:

- Introducing the application of the FID and KID metric, for the detector simulation as a powerful tool for evaluating the performance of deep generative models in detector simulation. Moreover, it provides detector-response level interpretations of the FID as a data-driven robust NN-based figure of merit for PXD.
- Achieving an FID score of 1.50, an over 42% improvement, and a KID score of 0.0010, as presented in Table 6.3.
- Proving an improvement to all marginal distributions in comparison to SOTA models for high-resolution PXD generation. In particular, IEA-GAN provides a weak (showed by the Mantel test) intra-event correlations among the generated detector images while all other models fail to capture any correlation.
- Illustrating the vital role of intra-event sensor-by-sensor correlation in the downstream Physics analysis.
- Doing a tracking analysis over the IEA-GAN's generated PXD background where it shows a very close performance to the Geant4 simulated background. Consequently,

This study revealed that IEA-GAN, despite only capturing a weak correlation, surpasses PE-GAN and even outperforms the inter-event-shuffled (uncorrelated) Geant4 in certain HPRs.

Using IEA-GAN comes also with a storage release of more than 2 orders of magnitude. Furthermore, due to the dramatic CPU speed-up of $\times 147$ as shown in Table 5.8, It is now possible to employ the IEA-GAN as an online surrogate model for the ultra high-granularity PXD background simulation on the fly, a task that was unattainable before for such a high-resolution detector simulation. This work has a significant impact on high-granularity fast and efficient detector response and collider event simulations. Since they require fine-grained intra-event-correlated data generation, I believe that the Intra-Event Aware GAN (IEA-GAN) offers a robust controllable sampling for high granular particle physics simulations. For instance, the High-Luminosity Large Hadron Collider (HL-LHC) [257] is expected to surpass the LHC’s design integrated luminosity by increasing it by a factor of 10. The upcoming high-granularity Calorimeter (HGCAL) with roughly 6.5M channels, or the ITk 3D pixel detector at the HL-LHC [304] with around 1M information channels, will massively increase the geometry and precision complexity, leading to a dramatic increase in the time and storage to simulate the detector [305]. As a result, much more effective and efficient high-resolution detector simulations are required. IEA-GAN is the first potential candidate for simulating the corresponding high-resolution and high-granular detector signatures with the remarkable capability of generating more than 7.5M pixel channels.

Table 5.8: Computational performance of IEA-GAN and PE-GAN generators on a single core of an Intel Xeon Silver 4108 1.80GHz (CPU) and NVIDIA V100 with 32 GB of memory (GPU) compared to Geant4. For the generative models, the mean and standard deviation were obtained for sets of 10 000 events, meaning that the model generates these events one at a time, not in a batch of 10 000. The time for Geant4 refers to the theoretical time it would take to run the simulation of background processes on the fly, one event at a time. The storage consumption corresponds to storing 10 000 events of 1 times the PXD background simulated information.

Hardware	Simulator	time/event [s]	Storage [Mb]	Speed-up
CPU	Geant4	≈ 1500	≈ 2000	1
	PE-GAN	11.781 ± 0.357	≈ 47	$\approx \times 127$
	IEA-GAN	10.159 ± 0.208	≈ 47	$\approx \times 147$
GPU	PE-GAN	0.090 ± 0.010	≈ 47	$\approx \times 16667$
	IEA-GAN	0.070 ± 0.006	≈ 47	$\approx \times 21429$

The ability to capture the underlying correlation structure of the data in particle physics experiments where the physical interpretation of the results heavily relies on it is very important. The true correlation between the occupancy of the sensors is determined by the underlying physical processes within the simulation. Although the true correlation

differs from the one captured by IEA-GAN, the model might be learning biases or artifacts introduced by the training data or the discriminator. Therefore, while the IEA-GAN can provide valuable insights into the correlations and patterns present in the data, it is important to interpret its results in conjunction with the domain knowledge. To alleviate the discrepancy, I expect that incorporating perturbations directly into the discriminator's RRM module would improve its contextual understanding and generalization. For example, using random masking [306] or inter-event permutation [307] over the samples and asking the RRM module to predict the representation of perturbed sample could improve the robustness of the model, a concept that will be studied in the next chapter.

In the next chapter, I will be dealing with the real (random trigger) PXD background data. I will show that the sparsity (occupancy) profile of the real PXD background data is much more diverse than the Geant4 simulated data. Moreover, the intra-event correlation of the real PXD data is dynamic across Belle II runs and has different monotonic and linear behavior. Hence, the main question will be whether IEA-GAN can handle the real PXD sparsity and complex intra-event correlation or whether I need a fresh perspective if I want to deal with these new challenges.

Chapter 6

PXD Background Generation: Deus Ex Machina the Real Detector Data

6.1 Introduction

During the last chapter, with the story of IEA-GAN conditioned on sensor numbers (as meta categories), many new concepts were introduced. The whole study was based on the Geant4 simulated PXD background data. The current chapter approaches the final challenge of the PXD detector simulation, studying the real (random trigger) PXD background conditioned on the rate of background or occupancy (as subordinate categories). During this path, I ask a more fundamental question. Is it possible to develop a surrogate model not only for amortized generation but also for Out-Of-Distribution (OOD) simulation? One of the most intriguing open problems in the realm of Deep Generative Models for Particle Physics is the possibility of simulating particle detector signatures that extend beyond the known experimental conditions. Can one extend these simulations into the OOD territories in a truly daring venture?

The essence of this query is embedded in our quest to transcend the simulation boundaries. In Chapter 4, I elucidated the foundation of OOD generation in detector simulation, reviewing the current works being done in this domain. From the real PXD data perspective, the idea of generating PXD hits conditioned over the amount of background (rate of background) and luminosity even beyond the current experimental data is still an open problem. In this chapter, I dive deeper into this regime, elaborate on the challenges, and propose possible solutions.

This is where *YonedaVAE* enters, striving to attack this problem with Self-Supervised context extrapolation. *YonedaVAE* not only emulates real PXD signatures (an In-Distribution simulation regime) but also extrapolates to the amount and profile of background beyond the training data (Out-Of-Distribution simulation regime). Along the way, I set the stage for a unified perspective of relational reasoning, the Yoneda perspective. The Yoneda perspective, rooted in the foundations of Category Theory, is an abstract mathematical lens that deals with objects and their relationships in an algebraic context. This perspective

emphasizes a shift in focus from studying objects in isolation to studying them in relation to others. Moreover, I also introduce a new set creation mechanism, *Adaptive Top- q sampling*, to adaptively create sets with variable inter-category (inter-event) and intra-category (intra-event) cardinalities. As a result, YonedaVAE, a point cloud generative model trained on low luminosity data, not only possesses the power to generate sparse ultra-high granularity PXD point cloud data but to do so for double the training luminosity region with a finesse and adaptability that holds the promise to redefine OOD simulations.

Toward the end, I also provide novel evaluation methods for detector simulations using Topological Data Analysis (TDA). I demonstrate the fresh perspective of TDA to analyze how and to what order of pattern complexity the PXD background hits are oriented and clustered.

6.2 Real PXD data and OOD Simulation: Challenges and Motivations

In the realm of detector physics data, achieving OOD (Out-of-Distribution) simulation demands transcending conventional homogeneous data representations, such as images and sequences typically used in experimental particle physics. The intent is to amplify the adaptability of generative models, thus enabling them to cater to unordered and variable-sized sets [183, 225, 250]. The first motivation for adopting unordered and variable-sized representations such as point clouds for the real (random trigger) PXD background is its high sparsity and vast variance in number of hits compared to the Geant4 version (see Figure 6.1). For example, in the recent PXD data taking, Experiment 26, the number of hits of sensors varies in the range [10, 5300] with a standard deviation $\sigma = 400$ and the mean around 500 as shown in Figure 6.6. That is why models based on homogeneous data representations (image-based) such as IEA-GAN perform poorly (as will be shown in Table 6.3). This is because Convolutional Neural Network (CNN)-based methods rely on dense tensors, which makes them suboptimal for spatially sparse data.

Thus, data representation versatile multi-sets becomes crucial in detector simulation, given the data irregularities. Examples of these irregularities include high variances in number of hits, gaps in data due to developmental factors (handling scarce or absent detector information), irregular detector geometries, and parameter extrapolation.

Historically, the primary objectives behind the current multi-set (point cloud) generative methods were two-fold: either to replicate detector responses based on specific sensor attributes and overarching conditions or to generate incidents within a target kinematic zone. However, these methodologies often stumbled when it came to generalizing to situations where the data is sparse or entirely absent [190, 256, 308, 103] like going beyond the kinematic regions of the training data or the situations where the upper bound of set cardinalities is well-beyond the training data like a geometry-independent detector simulation. For the PXD background case, an issue with using real random trigger data is its dependence on the experimental runs. Thus, real luminosity and beam-parameter-

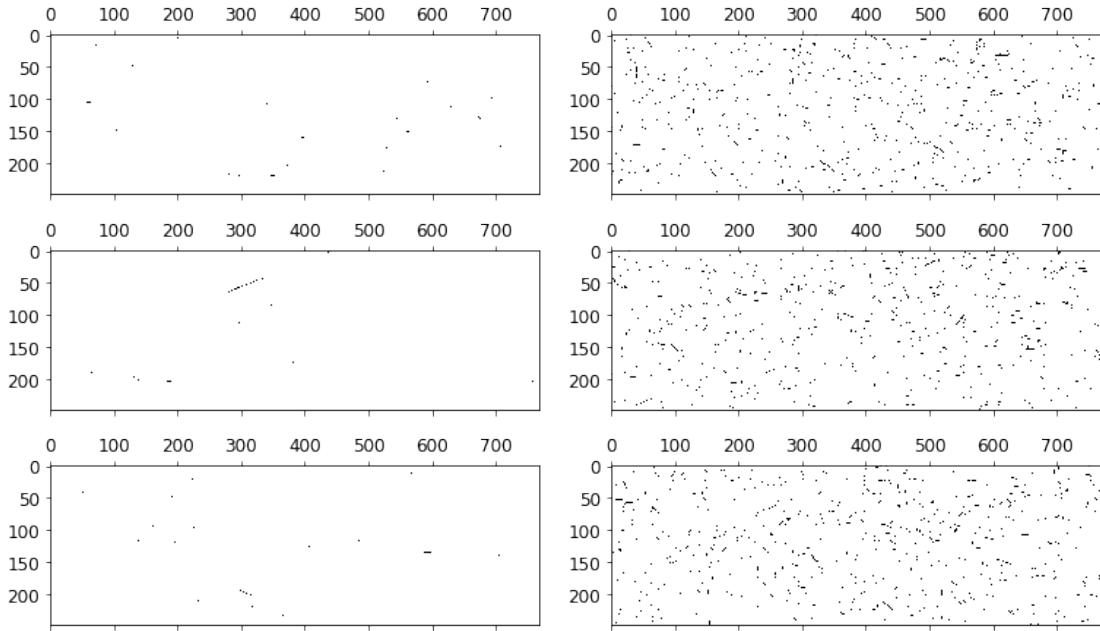


Figure 6.1: Random-trigger (real) PXD background (left) and Geant4 generated PXD background (right) in a bitmap format. As stated before, the sparsity of the real PXD background is much less than the Geant4 version.

dependent PXD background data beyond the current experiments will be absent. This signals a need for a surrogate model to generalize well to OOD luminosity regions.

Further complicating matters is the intrinsic challenge deep neural networks face in generating outputs that deviate from their training distribution [309, 310, 311, 312]. Looking beyond the domain of Experimental Particle Physics, the concept of OOD generation encompasses intriguing applications. One notable example is the *de Novo design*, prevalent in producing innovative protein structures or molecular compounds with desired attributes [313, 314]. Rather than merely reflecting existing data distributions, the *de Novo* approach leverages generative models to conjure entirely novel samples. The focus is tailoring these samples to exhibit preferred traits, be it stability, binding affinity, or protein foldability [315, 316, 317].

As one delves deeper into the OOD simulation for non-uniform data structures (multi-sets), three pivotal elements emerge:

1. The mechanism the model employs for generating sets that mirror the detector hit representation, whether represented as graphs or point clouds (multi-sets).
2. The model's capacity to do length extrapolation beyond the training data.
3. The model's adeptness at handling control parameters and performing context extrapolation beyond the training data.

Inspired by this question, this chapter develops a set of novel methods that enable a point cloud deep generative model to be able to extrapolate beyond the training data in a zero-shot manner. The proposed model trained on the detector response data with a specific intra-event correlation and limited cardinality profile can be generalized to a new set of intra-event correlation and cardinality profiles. In the subsequent sections, we'll embark on a more in-depth exploration of these elements.

6.2.1 Set Generation:

Graph or point cloud generation commonly engages in a two-fold process. Initially, point sampling occurs in a stochastic manner, primarily from a standard Gaussian distribution. Subsequently, the latent vector will be fused with each sampled point. Notably, this results in the manifestation of exchangeable distributions, whereby all potential permutations of a given set are equally probable. Such a property is often hailed as a generative model's counterpart to the concept of equivariance.

The main objective revolves around understanding the intricacies of designing a probabilistic decoder, denoted as f , whose goal is to transform latent vectors $\mathbf{z} \in \mathbb{R}^{d_{lat}}$ to sets (multi-sets). These sets, denoted as $\mathcal{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, encompass a variable number n of points, cardinality, $\mathbf{x}_i \in \mathbb{R}^d$. The challenge is to shape f in a manner that when \mathbf{z} is sourced from a prior distribution $p_Z(\mathbf{z})$, the resulting push-forward measure, represented as $f_{\#}(p_Z)$, aligns closely with an unknown distribution \mathcal{D} .

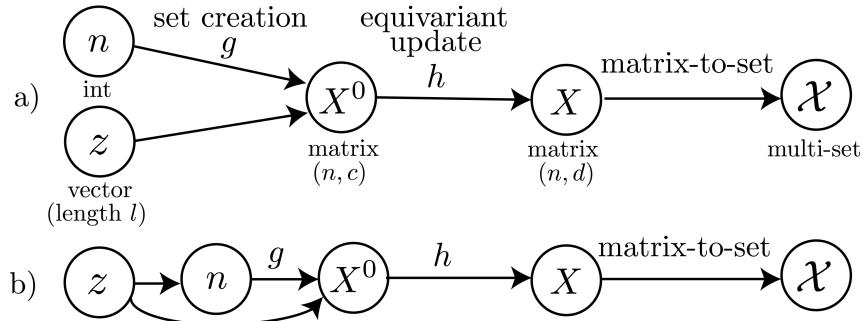


Figure 6.2: Illustrative models for set generation borrowed from [94]. The set's point count can either be derived from the dataset's distribution (a) or inferred from the latent vector (b). Although any equivariant function can serve for the update h , the set creation g encapsulates the intricacies of set generation. When generating graphs, edge weights complement the node features matrices \mathbf{M}^0 and \mathbf{M} .

Focusing on one-shot generation architectures, they predominantly align with the models depicted in Figure 6.2. The process commences by sampling a predetermined number of points for the set. A majority of methodologies presuppose the availability of set cardinalities during the training phase. During the generative phase, n is usually sampled from the distribution corresponding to set cardinalities from the training samples. Under this strategy,

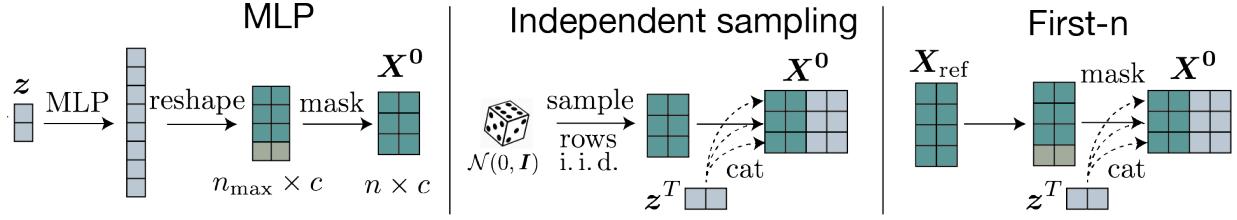


Figure 6.3: Existing creation methods, borrowed from [94], for mapping a latent vector \mathbf{z} to a set of points \mathbf{X}^0 . First-n creation empirically gives the best performance. It learns a reference set represented by a matrix \mathbf{X}_{ref} , and concatenates the latent vector to each point of this set.

the latent vector \mathbf{z} remains independent of the point count n , leading to the generative task of $p(\mathcal{X}|n, \mathbf{z}) \cdot p(n) \cdot p(\mathbf{z})$.

In contrast, [318] advocates for deducing the value of n directly from the latent vector via an MLP. Despite training this layer through an auxiliary loss, its predicted outcome only occurs during the generation stage. When relying on ground truth during training, the generative model transforms to $p(\mathcal{X}|\mathbf{z}, n) \cdot p(n|\mathbf{z}) \cdot p(\mathbf{z})$, indicating the interdependence of \mathbf{z} and n .

Upon determining n , the one-shot generation blueprints can be dissected into distinct components. The primary function, g (the *creation function*), maps the latent vector to an initial set $\mathbf{M}^0 \in \mathbb{R}^{n \times d_{\text{hid}}}$. Given the inherent simplicity of this function, it often falls short in modeling intricate inter-dependencies within each set. To overcome this, the initial set \mathbf{M}^0 may undergo further refinement via a secondary *update* functions, h that can be any equivariant modules. The succeeding segments delve into the proposed formulations for the set creation modules and the challenges they face with OOD set generation.

MLP Approach The adoption of MLPs is a recurring theme across several existing methodologies. These methods conventionally train an MLP to transform from $\mathbb{R}^{d_{\text{lat}}}$ to $\mathbb{R}^{n_{\text{max}} \times d_{\text{ch}}}$, where n_{max} symbolizes the maximum set cardinality encountered in the training dataset. The output, post-training, is a vector that is subsequently reshaped into a $n_{\text{max}} \times d_{\text{ch}}$ matrix. A masking procedure is applied to cater to the problem at hand, retaining only the initial n rows, as visualized in Figure 6.3.

MLPs, however, come with inherent constraints. Foremost among them is their inability to recognize and account for the symmetries intrinsic to the problem, which will be a huge problem with OOD set creation. Additionally, their training restricts them from generating a maximum of n_{max} points, effectively limiting their capacity to venture into extrapolating larger set sizes.

i.i.d. Method A widely embraced approach is the strategy of independent and identically distributed (i.i.d.) sampling. In this method, n points are drawn i.i.d. from a standard or low-dimensional Gaussian distribution. Following this, the latent vector is appended to each of these samples [254, 319, 318, 183, 225, 250]. The primary advantage of this methodology is its inherent preservation of permutation invariance, which is essential for many set and graph-related tasks.

However, a notable shortcoming is that DGMs built with an i.i.d. set generation mechanism fail to fit the train data correctly [320, 321, 322], which reflects in the poor quality/variability of the sampled sets due to excessive sources of randomness.

The First-n Approach Some researchers have adopted a distinct technique rather than the traditional practice of point sampling. Proposed by [323] and [321], this approach entails the use of a constant, learnable set, denoted as $\mathbf{X}_{\text{ref}} \in \mathbb{R}^{n_{\text{max}} \times d_{\text{ch}}}$. From this set, only the initial n rows are retained, leading to the terming of this technique as the *First-n creation*. Once established, this set has the latent vector attached to each of its individual points.

From empirical analyses, it has been discerned that the First-n strategy tends to achieve convergence at a more rapid pace when used with different sampling methods [321]. However, due to its deterministic nature, this method inherently restricts generation to a ceiling of n_{max} points, making it incapable of extrapolating towards larger sets. An implicit bias is also introduced due to the recurrent selection of the first n rows from the reference set \mathbf{X}_{ref} . This results in a disproportionate frequency of selection for the leading rows compared to those towards the end.

The Top-N Mechanism Moving away from the conventional i.i.d. sampling for set generation, Vignac et al. introduced a new approach [94]. Their method revolves around a trainable reference set, \mathbf{R}_{ref} . Instead of indiscriminately sampling from this set, the latent vector is employed to select the top N most relevant points, giving rise to the name *Top-N creation*. Once these points are identified, they are processed in conjunction with the prior vector. Empirical studies have shown that the Top-N mechanism surpasses traditional set generation in various benchmarks.

However, despite solving many issues in set generation for ID (In-Distribution) generation regime, they still lack the possibility to generate sets with variable sizes that belong to the same category. They do the sampling by grouping an equivalent class of sets with the same cardinality, which can easily bias the model toward more frequent cardinalities at inference time, indicating a lack of diversity in the generated set. Another limitation of this approach in OOD generation is its dependency on the predefined reference set \mathbf{R}_{ref} , which might not always capture the full diversity of the data if the inference set cardinality is much higher than the training set cardinality.

In this chapter, I go to the next level for OOD set generation by introducing *Adaptive Top-q Sampling*. With Adaptive Top-q Sampling, instead of relying on a fixed top-k or using a temperature parameter to control the shape of the distribution without sufficiently suppressing the unreliable tail, I propose sampling from the top- q_i portion of the probability mass where q_i is adaptively learned in context for each sample i , expanding and contracting the candidate pool dynamically for each sample.

6.2.2 Length Extrapolation

While sampling sets with correct cardinality stochastically and maintaining the symmetries of the problem during training is of paramount importance, it does not guarantee proper extrapolation during decoding inference. In other words, given a generated set, it has to be

mapped to the final representation (point cloud or graph) in such a way that the model can extrapolatively reason (update) to a longer set size. The length (cardinality) of the sets in the ID training, n_{tr} , is an important factor. Due to the above-mentioned motivations, one will use the model for substantially longer evaluation set cardinalities $n_{ex} \gg n_{tr}$ where only control parameters/attributes exist for the OOD. The discrepancy between n_{tr} and n_{ex} motivates the task of **length extrapolation** [324, 325]: *Can a set decoder model maintain equally good, if not better, perplexities when longer sequences are used in the testing stage?*

It is a well known fact that DeepSet-based models [125, 326, 250] have difficulty extrapolating to set sizes [318, 327, 328]. As an alternative, Attention-based models are the other equivariant candidates for set updates. However, it has been recently [325, 329] demonstrated that the vanilla Transformer Encoder fails to extrapolate to longer sequences, which is caused by the position embedding method. On the other hand, any positional information injection to the attention-based models will break the exchangeability of the set by order variance. From the DGM perspective, as I have shown in Chapter 3, sufficient conditions for (F, l) - equivariance, the VAE's decoder, and GAN's generator does not need to carry exchangeability condition for their corresponding set update. As a result, their length extrapolation can be enabled by changing the positional embedding policies.

Transformers, unlike RNNs, are designed to process data in parallel and require positional encoding to capture point order. Positional encodings can be either *absolute*, which signifies each exact position (like 1, 2, 3, ...), or *relative*, denoting the distance between tokens. Let's dive into various ways of incorporating positional information and their ability in length extrapolation.

Absolute Position Embedding (APE) converts each position i into a unique vector p_i , which is added to word embeddings before introducing them to the model as:

$$e_i = w_i + p_i$$

Where e_i represents the combined embedding for the token at position i , incorporating both its semantic information (from w_i) and its positional information (from p_i). For a given position i , the non-parametric approach to APE, as introduced in the original Transformer model [29], calculates the positional encoding using sine and cosine functions:

$$p_i^{(2k)} = \sin\left(\frac{i}{10000^{2k/d}}\right)$$

$$p_i^{(2k+1)} = \cos\left(\frac{i}{10000^{2k/d}}\right)$$

Where d is the dimensionality of the embeddings and k ranges from 0 to $d/2 - 1$. These functions ensure the model can generate positional embeddings for any position, even if it hasn't seen it during training.

In contrast, some models, including GPT3 [330] and OPT [331] instead of relying on fixed functions like sine and cosine, models, introduce trainable position embeddings. In this approach:

$$p_i = \text{LearnableParameter}(i)$$

Here, p_i is initialized randomly and updated during the training process along with other model parameters. The drawback, however, is that since these embeddings are learned from the training data, the model cannot naturally extrapolate to positions beyond the range it has seen during training. However, it's restricted to known positions and cannot extrapolate.

T5's Relative bias [332] computes the bias between token positions i and j using a lookup function, adding this bias to the self-attention mechanism's dot product. Distances beyond a specific range map to a common parameter, ensuring adaptability to unknown distances. Specifically, for any two token positions i and j , the relative distance d is given by:

$$d = i - j$$

The lookup function f is used to determine the bias associated with this relative distance:

$$\text{bias}(d) = f(d)$$

In the self-attention mechanism, the attention score between two positions is typically computed using the dot product of the query and key vectors. Let's denote the query for position i as Q_i and the key for position j as K_j . Without considering any bias, the attention score S would be:

$$S(i, j) = Q_i \cdot K_j^T$$

However, with T5's relative bias, the attention score is modified as follows:

$$S(i, j) = Q_i \cdot K_j^T + \text{bias}(d)$$

For values of d that exceed a certain threshold, say Δ , the lookup function maps them to a consistent bias value:

$$f(d) = \begin{cases} f(d) & \text{if } |d| \leq \Delta \\ f_{\max} & \text{otherwise} \end{cases}$$

Where f_{\max} is a common parameter value representing the maximum bias. This feature ensures the model remains resilient and adaptable even when encountering unfamiliar relative distances between tokens.

Rotary (ROPE) introduces a novel technique for encoding relative positions in the context of the self-attention mechanism of Transformers. In essence, the ROPE method revolves around rotation operations applied to the query and key representations based on their absolute positions, which in turn ensures that the attention mechanism effectively focuses on the relative distances between tokens. For any given token position i , the Rotary mechanism calculates a rotational positional encoding:

$$R(i) = \begin{bmatrix} \cos(\theta(i)) & -\sin(\theta(i)) \\ \sin(\theta(i)) & \cos(\theta(i)) \end{bmatrix}$$

Where $\theta(i)$ is a function of the absolute position i . Then, before feeding the query and key representations into the attention mechanism, they are rotated using their corresponding

positional encodings. Let's denote the query for position i as Q_i and the key for position j as K_j . Their adjusted representations are:

$$Q'_i = Q_i \times R(i)$$

$$K'_j = K_j \times R(j)$$

Afterward, the dot product attention score between the adjusted query and key representations will be $S(i, j) = Q'_i \cdot K'^T_j$. Because both Q'_i and K'_j have been rotated based on their absolute positions, the attention score $S(i, j)$ is inherently based on the relative distance $d = i - j$. In other words, despite the Rotary mechanism using absolute positional information to adjust the representations, the attention mechanism ultimately focuses on the relative distances between tokens, ensuring a more context-aware attention distribution.

By this method, the Rotary mechanism in models such as PaLM [333] and LLaMA [334] effectively leverages both absolute and relative positional information to enhance the self-attention mechanism's contextual understanding [335].

ALiBi, [325] in BLOOM [336], acts similarly to T5's Relative Bias but reduces a scalar bias from the attention score. The bias increases with the distance between tokens, leading to a tendency towards recent tokens. When using ALiBi, position embeddings are not added at any point in the network. The only modification applied is after the query-key dot product, where a static, non-learned bias is added:

$$\text{softmax}(q_i K^\top + m \cdot [-(i-1), \dots, -2, -1, 0])$$

The scalar m is a head-specific slope that is fixed before training. This bias penalizes the attention scores between distant query-key pairs, with the penalty increasing as the distance between a key and a query grows. The different heads increase their penalties at different rates, depending on the slope magnitude. For models with 8 heads, the slopes used are the geometric sequence:

$$\frac{1}{2^1}, \frac{1}{2^2}, \dots, \frac{1}{2^8}$$

This method introduces an inductive bias towards *recency*, penalizing attention scores between distant query-key pairs. The penalty increases as the distance between a key and a query grows. Depending on the slope magnitude, different heads increase their penalties at different rates.

XPOS, introduced in the Length-Extrapolatable Transformer [337], offers a novel approach to relative position modeling in Transformers. The core idea behind XPOS is to add the same decay as in ALiBi that leverages the extrapolation to all distances.

For a given token position j , XPOS calculates a position encoding based on exponential and rotational transformations:

$$fq(q, j) = q e^{\xi j + i\theta j}$$

$$fk(k, j) = k e^{-\xi j - i\theta j}$$

Where ξ and θ are parameters that determine the rate of exponential decay and rotation, respectively. If $\xi = 0$, the encoding is similar to the ROPE method. The transformation

provides a rotation on vectors, ensuring that the relative angle between vectors q and k determines the inner product. The attention scores between the transformed query and key representations inherently focus on the relative distance between tokens. This is achieved by ensuring that the inner product between vectors with a larger relative angle is smaller, thereby capturing the relative distances effectively.

It has been shown [325, 337] showed that the APE method in practice has very limited extrapolation capabilities. On the other hand, the rotary position method improves over the sinusoidal one, but it still does not achieve satisfying results. Interestingly, they observe that the T5 bias method leads to better extrapolation than either of these, concluding that extrapolation ability depends heavily on the position embedding.

In this study, I incorporate Rotary positional encoding for detector geometry and introduce modifications to ALiBi positional embeddings to be more compatible with bidirectional models (they were introduced for autoregressive models) and, in general, with OOD generalization.

6.2.3 Context Extrapolation:

Although there are very few works on Context Extrapolation for generative models, the common theme amongst the remaining is “controlled generation”. When one is dealing with a controlled generation, it involves generating samples characterized by a particular attribute, feature, or context. In Natural Language Processing (NLP), the context can be sentiment style as a prompt. In computational biology and chemistry, the context can represent attributes like stability, fluorescence, and binding affinity of a protein sequence. In event generation and detector simulation, the context can be the kinematic or luminosity profile, the number of detector hits (amount of background hits), and detector geometry.

OOD generation focuses on generating samples with context values that extrapolate beyond the training distribution. For example, in protein design, the problem of *de novo* design, simulation of novel protein sequences with respect to some context features such as binding affinity to a specific target, is of vital importance to drug discovery [338, 339, 340]. In event generation, going beyond the manifold of training events can bring physical insight into regions without any experimental data.

While controlled generation is prevalent in literature, its capability for extrapolation remains largely unexplored. The current approaches can be categorized into the following,

- **Control Code Methods:** These employ control codes encoding attributes either as discrete values (similar to style transfer) or scalar values [341, 342, 262]. However, discrete codes often face challenges when exposed to unseen attribute values.
- **Iterative Editing Methods:** Edit-based methods focus on iterative refinements [343, 344, 345]. Some recent works involved few-shot model learning by receiving feedback on generated samples and learning to edit itself [346, 347].
- **Latent Variable Models:** With latent variable models, control features are modeled inside latent variables. For instance, Genhance introduces a method where the latent

vector comprises both attribute-relevant and irrelevant components [348]. However, stability issues arise when these models operate on set-based data.

- **Scorer Model for Attribute Control:** Some models incorporate attribute information through a scorer model during inference, guiding an unconditional generative model [349, 350, 351]. While effective, this method is not always suitable for context extrapolation, especially when it goes beyond the training data distribution. The classifier-type model, when used as a reward model for reinforcement learning, also presents challenges [352, 353, 354, 355, 356] as the generator can exploit and amplify imperfections in the reward.

With YonedaVAE, I introduce methods as a fusion of Control Code Methods and Latent Variable Models and show that it can reach context extrapolation with Transformer-based models by synthesizing Self-Distillation and VAEs in conjunction with introducing a novel control-gating for the attention mechanism. However, before going deeper into the YonedaVAE, I first have to introduce a building block of this model: Category Theory. Using the concepts of category theory, I try to show how it relates mathematically to *Yoneda pooling* module, a relational pooling mechanism introduced in the YonedaVAE’s encoder.

6.3 Prologue: Category Theory and Yoneda Perspective

In this section, I briefly introduce the concept of Category theory (CT). Consequently, I will demonstrate how CT acts as a universal language to formulate the Yoneda pooling mechanism that involves relational intra-event dependencies.

6.3.1 What is Category Theory?

Category theory, initiated by Eilenberg and Mac Lane [357], provides a unifying and abstract framework in which mathematical structures and their interrelationships can be studied. CT can be seen as a language of structure, abstraction, and relationships. It allows us to describe mathematical (and non-mathematical) concepts in a highly general and structural way, focusing on the relationships between these concepts rather than their internal details. This powerful bird’s-eye view of mathematics makes CT an invaluable tool with diverse applications in areas such as logic, Physics, Chemistry, Computer Science, and AI [358, 359]. A Category is any collection of objects that can relate to each other via morphisms like functions with specific features, like composition and associativity. So, the collection of sets with functions forms a category, as does the collection of groups with group homomorphisms and topological spaces with continuous functions.

6.3.2 Basic Definitions

A *category* \mathcal{C} consists of the following data:

- A class $\text{ob}(\mathcal{C})$ of *objects*.
- For every pair of objects $A, B \in \text{ob}(\mathcal{C})$, a set $\text{Hom}_{\mathcal{C}}(A, B)$ of *morphisms* or *arrows* from A to B .
- For every object $A \in \text{ob}(\mathcal{C})$, a specified morphism $1_A \in \text{Hom}_{\mathcal{C}}(A, A)$ called the *identity* on A .
- For every triple of objects $A, B, C \in \text{ob}(\mathcal{C})$, a binary operation $\text{Hom}_{\mathcal{C}}(B, C) \times \text{Hom}_{\mathcal{C}}(A, B) \rightarrow \text{Hom}_{\mathcal{C}}(A, C)$ called *composition* of morphisms.

These data are subject to the following axioms:

- (Associativity) For every quadruple of objects $A, B, C, D \in \text{ob}(\mathcal{C})$ and every triple of morphisms $f \in \text{Hom}_{\mathcal{C}}(A, B)$, $g \in \text{Hom}_{\mathcal{C}}(B, C)$, $h \in \text{Hom}_{\mathcal{C}}(C, D)$, one has $(h \circ g) \circ f = h \circ (g \circ f)$.
- (Identity) For every pair of objects $A, B \in \text{ob}(\mathcal{C})$ and every morphism $f \in \text{Hom}_{\mathcal{C}}(A, B)$, one has $f \circ 1_A = f = 1_B \circ f$.

$$\begin{array}{ccc} A & \xrightarrow{f} & B \\ & \searrow^{g \circ f} & \downarrow g \\ & & C \end{array}$$

Figure 6.4: Commutative diagram in a category

6.3.3 Functors and Natural Transformations

A *functor* $F : \mathcal{C} \rightarrow \mathcal{D}$ between two categories \mathcal{C} and \mathcal{D} is a mapping that assigns to each object $A \in \text{ob}(\mathcal{C})$ an object $F(A) \in \text{ob}(\mathcal{D})$ and to each morphism $f \in \text{Hom}_{\mathcal{C}}(A, B)$ a morphism $F(f) \in \text{Hom}_{\mathcal{D}}(F(A), F(B))$, such that the following conditions hold:

- $F(g \circ f) = F(g) \circ F(f)$ for all morphisms $f, g \in \text{Hom}_{\mathcal{C}}$ for which the composition $g \circ f$ is defined.
- $F(1_A) = 1_{F(A)}$ for all $A \in \text{ob}(\mathcal{C})$.

A *natural transformation* $\eta : F \Rightarrow G$ between two functors $F, G : \mathcal{C} \rightarrow \mathcal{D}$ is a family of morphisms $\eta_A : F(A) \rightarrow G(A)$ in \mathcal{D} , one for each object $A \in \text{ob}(\mathcal{C})$, such that for every morphism $f : A \rightarrow B$ in \mathcal{C} , the following square commutes in \mathcal{D} :

$$\begin{array}{ccc}
 F(A) & \xrightarrow{\eta_A} & G(A) \\
 F(f) \downarrow & & \downarrow G(f) \\
 F(B) & \xrightarrow{\eta_B} & G(B)
 \end{array}$$

Figure 6.5: Commutative square for a natural transformation

The composition of natural transformations is defined component-wise. This turns the class of all functors between two fixed categories into a category itself, with natural transformations as morphisms. This concept of a category of functors is essential for the definition of a Yoneda lemma, a central concept in category theory.

An experimental Example: In an optical laboratory, think of each experimental setup - whether involving simple lenses, prisms, or mirrors - as *objects* in Category Theory. The transformations we apply to these setups, such as adding a filter to a light source or replacing a flat mirror with a concave one, act as *morphisms*, reflecting how one setup can evolve into another. The act of consecutively adding a filter and then changing a mirror represents the *composition* of morphisms. Importantly, the final experimental outcome remains consistent regardless of the order of these transformations, highlighting *associativity*. Some transformations, like placing an undistorted glass sheet in front of a light, might leave the setup unchanged, analogous to the *identity* morphism in Category Theory. Now, picture a neighboring acoustic lab where experiments with sound waves parallel our optical experiments. Transitioning our thought process from optics to acoustics while retaining the foundational concepts (like translating optical reflection to acoustic echo) serves as a *functor*, bridging these two distinct “categories” of experiments. Within the acoustics realm, subtle changes, such as transitioning from air to water as the medium and preserving the experiment’s essence, can be linked to *natural transformations* between functors. Through this lens, the optical lab analogy offers a tangible depiction of Category Theory’s essence.

In the context of AI, category theory offers a novel framework for understanding and unifying learning algorithms [358, 359, 360, 361]. As I will demonstrate in this thesis, category theory can also provide a powerful tool to have a fresh and unified understanding of intra-event relational reasoning, enabling the development of our proposed system, *YonedaVAE*.

6.3.4 The Yoneda Perspective

The Yoneda Perspective states that mathematical objects are completely determined by their relationships to other objects. If one wants to understand objects (sets, groups, topological spaces, etc.), then one has to understand the network of relationships they enjoy with all the other objects of their species [362].

6.3.4.1 Yoneda Embedding and Yoneda Lemma

Yoneda Embedding offers a deep insight into the structure of a category by analyzing the set of morphisms (or functors) from a given object (or functor) to all other objects (or functors) in the category. Consider a category \mathcal{C} and its category of functors $[\mathcal{C}^{op}, \text{Set}]$, where \mathcal{C}^{op} is the opposite category of \mathcal{C} and Set is the category of sets. For each object A in \mathcal{C} , there's a functor $h_A : \mathcal{C}^{op} \rightarrow \text{Set}$ given by:

$$h_A(B) = \text{Hom}_{\mathcal{C}}(B, A)$$

for every object B in \mathcal{C} , and for every morphism $f : B \rightarrow B'$ in \mathcal{C} , $h_A(f)$ maps a morphism $g : B' \rightarrow A$ to its composition $g \circ f$.

The Yoneda Embedding states that the object A in \mathcal{C} can be embedded into the category $[\mathcal{C}^{op}, \text{Set}]$ via the functor h_A .

The Yoneda Lemma establishes a one-to-one correspondence between the natural transformations from h_A to any functor $F : \mathcal{C}^{op} \rightarrow \text{Set}$ and the elements of $F(A)$. Formally, the Yoneda Lemma asserts that for any functor $F : \mathcal{C}^{op} \rightarrow \text{Set}$, there is a natural isomorphism:

$$\text{Nat}(h_A, F) \cong F(A)$$

where $\text{Nat}(h_A, F)$ denotes the set of all natural transformations from h_A to F .

Optical Lab Example Continuing with our experimental analogy, weaving in the Yoneda Lemma and Yoneda Embedding.

Within our optical lab, imagine there's a universally renowned experimental setup called the *Yoneda Experiment*. This setup is known for its unique property: for any other experiment in the lab, we learn about that experiment based on how it interacts with the Yoneda Experiment. For example, consider how an experimental setup with prisms interacts with the Yoneda Experiment: does it amplify the light? Refract it in a unique pattern? Or perhaps it even dims it? These interactions (or transformations) can be thought of as **morphisms** from our prism experiment to the Yoneda Experiment. Now, the collection of all such interactions from every experiment in the lab to the Yoneda Experiment is similar to the **Yoneda Embedding**. Just by looking at these interactions, we get a deeper insight into the structure of the whole lab and how each setup inherently behaves.

Meanwhile, the **Yoneda Lemma** would state something profound about our lab: If you know all the ways every setup in the lab can interact with the Yoneda Experiment, then you can predict how any given setup would behave in any conceivable scenario or with any conceivable tool or modification. For instance, imagine the lab gets a new, mysterious tool. If we understand how this tool interacts with the Yoneda Experiment and how every other setup interacts with the Yoneda Experiment, then we can predict how this tool will modify or affect any other setup in the lab. The Yoneda Lemma, in this context, is a powerful prediction tool. It asserts that if we understand these interactions deeply enough, we can foresee how any setup in our lab (object in our category) will behave under any new circumstances (with any functor).

6.3.4.2 Presheaf

A presheaf on a category \mathcal{C} is simply a functor from the opposite category \mathcal{C}^{op} to the category of sets Set . Formally, a presheaf F is defined by:

$$F : \mathcal{C}^{op} \rightarrow \text{Set}$$

For each object A in \mathcal{C} , $F(A)$ is a set, and for each morphism $f : A \rightarrow B$ in \mathcal{C} , $F(f)$ is a function from $F(B)$ to $F(A)$.

Expanding on the notion of presheaves, one can consider cases where, instead of sets, one assigns vector spaces to each object of \mathcal{C} . Such presheaves are referred to as vector-valued presheaves.

Formally, a vector-valued presheaf F on a category \mathcal{C} with values in the category Vect of vector spaces over a fixed field k is a functor:

$$F : \mathcal{C}^{op} \rightarrow \text{Vect}$$

For each object A in \mathcal{C} , F assigns a vector space $F(A)$. For each morphism in \mathcal{C} , F provides a corresponding linear transformation between these vector spaces.

For example, consider the category of open subsets of a topological space X , where the morphisms are inclusions. A vector-valued presheaf on this category might assign to each open subset U a vector space of functions from U to k that satisfy certain properties (e.g., continuous functions, smooth functions, etc.).

Optical Lab Example A **presheaf** in category theory is essentially a contravariant functor from a category to the category of sets. In the optical lab, it can be a systematic way of assigning, to every experimental setup, a set of *observations* based on the properties and results of that setup. For instance, for every experimental setup involving lenses, prisms, or mirrors, our presheaf could assign a set of observed light patterns, intensities, or colors produced by that setup. If you make a modification to a setup (e.g., adding a filter or changing a lens), this could correspond to a morphism in our category, and our presheaf would tell us how the set of observations changes in response.

The Yoneda Embedding, in the context of presheaves, involves understanding how each setup (or object) in the optical lab interacts with the Yoneda Experiment. In the context of our presheaf, this means looking at how the observations (or sets assigned by the presheaf) for each setup relate to each other and to those of the Yoneda Experiment.

Specifically, for every setup in the lab, we would consider the set of all possible interactions that the setup can have with the Yoneda Experiment. This “set of interactions” forms a set for each object, and this assignment from setups to sets, based on interactions with the Yoneda Experiment, is a presheaf! This special presheaf is called the **representable presheaf** associated with the Yoneda Experiment. The Yoneda Lemma then tells us that, for any other presheaf (any other way of assigning sets of observations to setups), the set of natural transformations from this representable presheaf to that presheaf is isomorphic to the set of observations assigned to the Yoneda Experiment by that other presheaf.

The most important message of the above analogy and discussion is as follows: In our optical lab, presheaves are like systematic observation logs detailing the various outcomes of each experimental setup. The Yoneda Embedding highlights the importance of understanding interactions with the Yoneda Experiment, and the Yoneda Lemma provides a bridge between the special observation log of the Yoneda Experiment and any other possible observation log in the lab. Through this analogy, we see the power of the Yoneda perspective: *by deeply understanding an experiment and its interactions with the other experiments, we can gain insights into the entire laboratory's structure and behavior.*

The following section introduces YonedaVAE and demonstrates an analogy between Yoneda perspective and intra-event reasoning. As a result, it introduces *Yoneda-pooling*, a potent relational pooling layer that will be of paramount importance for the YonedaVAE to do OOD point cloud generation.

6.4 YonedaVAE

YonedaVAE is a VAE-based deep generative model for multiset (point cloud and graph) generation with a self-distillation mechanism. It consists of an Encoder, Set Generator, and Decoder. In the following, I go through each module and the technologies YonedaVAE introduces to each to reach OOD simulation with context extrapolation.

6.4.1 Problem setup

Working with the real (random trigger) PXD data, and the fact that the installed PXD is incomplete (until experiment 26) and only 19 sensors are installed. For real PXD in the point cloud format, we denote an input event multiset with N_{evt} samples from an event (in our case 19 samples) where each sample comes with variable $\{N_l^i\}_{i=1}^{19}$ permutation invariant hit points as $\mathbf{x} = \{x_1, \dots, x_l\} \in \mathbb{R}^3$ and a sensor-level feature as $\mathbf{c} \in \mathbb{R}^{evt}$ such as cardinality, or energy deposits, and an event-level feature $\mathbf{e} \in \mathbb{R}$. \mathbf{e} denotes any event-level attribute such as luminosity, maximum occupancy, or mean occupancy that defines the controllable context for sampling. Samples in each event would have different cardinalities as each sensor could have various numbers of hits. During training, we are typically given a large supervised corpus $\mathcal{D}_{train} = \{(\mathbf{x}^{(m)}, \mathbf{e}^{(m)}, \mathbf{c}^{(m)})\}_{m=1}^{M_{train}}$ with maximum cardinality $N_{l_{max}} = 400$. I refer to this region as the *training range*. On the other hand, during inference we have only a set of Event-level attribute values, $\mathcal{D}_{test} = \{(\mathbf{e}^{(m)}, \mathbf{c}^{(m)})\}_{m=1}^{M_{test}}$ with maximum cardinality $N_{l_{max}} = 5300$. I refer to this region as the *extrapolation range*. In other words, a model trained on a max event cardinality of $19 \times 400 = 7600$ should extrapolate to a max event cardinality of $19 \times 5300 = 100700$. Our goal is to generate detector responses that have context profile and cardinality beyond the training region.

There are two main applications that I am targeting with this setup:

1. **PXD background simulation with length extrapolation:** Given that one has access to the fine sensor-level cardinalities of an event, for example, in the case of PXD

background amplification, one can sample from the cardinality distribution (even beyond the training region) to generate a background for any amount of multiplicity (e.g. $20\times$ the nominal background). In this case, we are dealing with a few-shot conditional generation task where length extrapolation becomes very important (see Figure 6.6), but context extrapolation is not as important as we have access to the fine-level context (sensor-level attributes). Therefore, during the training, the model has access to cardinalities within the $[1, 400]$ range. During inference, the model must generate point clouds with cardinalities in the $[400, 5300]$ range.

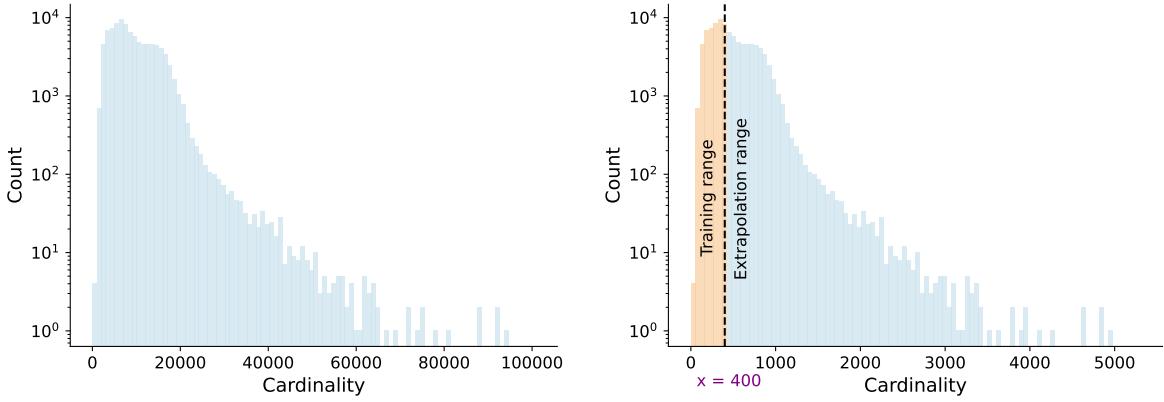


Figure 6.6: (right) The cardinality distribution in log scale for the training that belongs to Experiment 12 data and extrapolation (inference) region that belongs to Experiment 26 data. (left) The overall cardinality distribution of each event can go up to 100k points per event.

2. **OOD PXD background simulation with context extrapolation:** When one does not have access to the fine sensor-level cardinality profile, and only knows the coarse event-level attributes such as luminosity or the max cardinality of the whole event in the extrapolation region. In this case, we are dealing with a zero-shot generative task, for which context extrapolation is extremely important (see Figure 6.7) as we do not have access to fine details and only have access to coarser information (OOD event-level attributes). In particular, during the training, the model has access to the max cardinality of each event as the context within the range $[100, 400]$. During inference, the model gets max cardinalities as the context in the range $[400, 5300]$.

6.4.2 YonedaVAE: Yoneda Encoder

YonedaVAE's encoder, during training, takes the set of detector signatures event-by-event as a point cloud of shape $N_{evt} \times N_l \times 3$, along with each sensor's conditional parameter as inputs. The three feature channels in the input point cloud correspond to the (x, y, charge) data of the PXD hits. In our setup, the conditional parameters are the normalized occupancies for each sensor. The encoder comprises two main stages. First, the trunk of

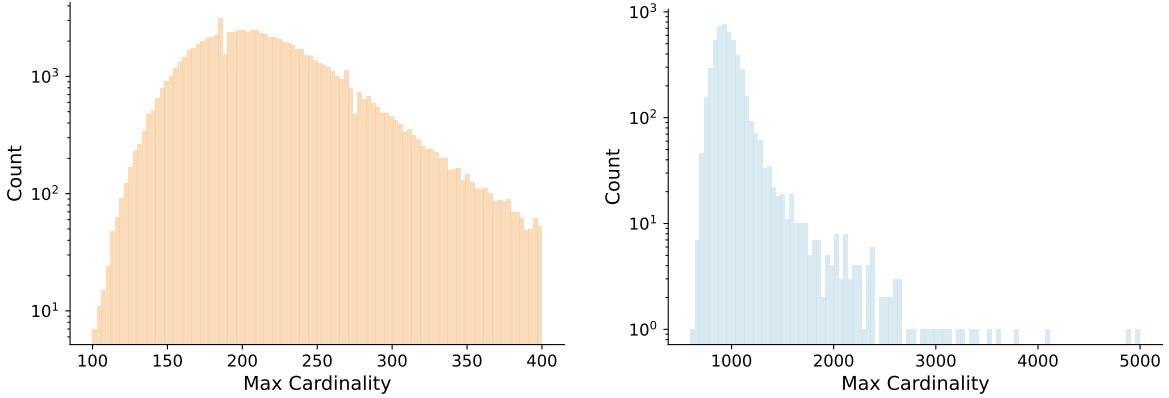


Figure 6.7: (left) The max cardinality distribution in log scale for the training that belongs to Experiment 12 data as event-level attributes. (left) The max cardinality distribution in the log scale for inference that belongs to Experiment 26 data as event-level attributes.

the network processes the inputs through multiple layers of an equivariant neural network block that I term *Eventformer*, as depicted in Figure 6.8, to output $N_{evt} \times N_l \times ch$ array that represents, N_{evt} number of samples in an event which is 19 for real PXD data, N_l the number of points representing the number of hits per sample, and ch the embedding dimension. The Eventformer blocks contain a number of attention-based components with a gating mechanism and cross-attention as I discuss below. The key point in the Eventformer block is the mechanism to inject intra-event context within each point cloud representation which enables direct reasoning over both the sensor and event relationships.

6.4.2.1 Eventformer and Attention Gating

After a two-layer MLP layer with Layer normalization, the point cloud information enters the Cross Attention module as the queries. From the other side, the embedded sample-level features (attributes for each sample in the event) also enter the Cross Attention module as Keys and Values. This is a mechanism to assign the sensor attribute to its corresponding sensor hit information.

Scale Normalization: Within both Cross Attention and Multi-Head Attention (MHA) I incorporate the residual connections [34] and Scale Normalization [363] instead of the Layer Normalization to achieve a faster convergence. Layer Normalization inspired by batch normalization [364], aims to reduce internal covariate shift by fixing the mean and variance of activation distributions. Both have been applied to self-attention [29, 365]. However, [366] shows that batch normalization’s success has little to do with covariate shift but comes instead from smoothing the loss landscape. As a result, [363] proposed replacing the Layer Normalization with Scale Normalization:

$$\text{ScaleNorm}(\mathbf{x}; g) = g \frac{\mathbf{x}}{\|\mathbf{x}\|}.$$

With Scale Normalization can be viewed as compactifying d -dimensional vectors onto a $(d-1)$ -dimensional hypersphere with learned radius g . This expresses the inductive bias that each sublayer’s activations have an ideal “global scale,” by replacing the $2d$ scale and shift parameters of Layer Normalization with a single learned scalar, improving computational and parameter efficiency.

Attention Gating: After the fusion of point clouds and their corresponding attributes, I introduce a gating mechanism over the attention weights to overcome the *over-smoothing* phenomenon that happens with multiple layers of MHA modules. Over-smoothing [367] refers to the exponential convergence of all point features towards the same constant value as the number of Attention layers increases. For Transformer-based models, this means that the excessive blending of representations, causes the model to assign very similar representation to each mode over successive layers and potentially leads to a loss in the granularity of information. In fine-grained cases, over-smoothed representations are less informative because distinct points become indistinguishable.

To address the issue of over-smoothing, this study introduces a *gating mechanism* on every MHA layer, regulating the layer’s output. Gating was initially devised to counter the Short-Term memory challenges seen in RNNs. RNNs, when handling extensive input sequences, sometimes struggle to retain information throughout the sequence’s entirety. Additionally, they are vulnerable to the vanishing gradient dilemma in initial layers, negatively impacting the learning process. The LSTM [368] architecture was designed to overcome these challenges. It employs gates within the standard RNN cells, which learn and decide which data to retain and which to dismiss. This decision-making process is facilitated by the sigmoid functions that adjust values between 0 and 1. In this setting, values closer to 0 indicate that specific cell information should be discarded, while values nearing 1 denote information worth retaining.

In our setup, the inclusion of the output gate signifies that certain information might need to be discarded after the self-attention by Hadamard multiplication of the sigmoid-transformed values to the attention matrix’s linear transformation. This provides a strong regulation over what information should be passed on to the subsequent layers for further refinement. There are certain points with a higher likelihood to form larger hit clusters than others, and so they are more important in determining the consistency of PXD hits. Thus, the gating mechanism provides an effective mechanism to prevent over-smoothing over multiple Attention layers which in the end enables the model to not overfit and be able to extrapolate. Without this gating mechanism, context extrapolation is not possible.

6.4.2.2 Yoneda-Pooling

Inspired by the Yoneda embedding, the trunk of the network is followed by a novel aggregator layer, *Yoneda-Pooling*, that introduces a learnable PNA-pooling [369] module that is also aware of samples relationship with each other. This module maps the input point clouds to the VAE’s latent manifold. The Key innovations in the Yoneda-pooling mechanism are the simultaneous global refinement over all samples within an event with respect to the event-level attribute, using the attention mechanism, and adding learnable weights to the

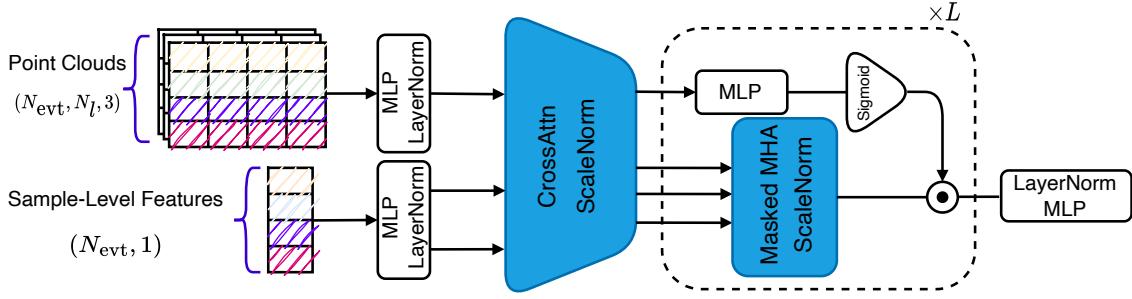


Figure 6.8: Eventformer

PNA-aggregator.

PNA Aggregator: The Principal Neighbourhood Aggregation (PNA) [369] mechanism offers an advanced approach to aggregate information in multisets. Instead of relying on traditional mean, sum, or max aggregations, PNA combines multiple aggregators and scalers in a systematic manner. The aggregation functions can include the mean, sum, max, min, and standard deviation, capturing diverse statistical properties of node features in the neighborhood. On the other hand, scalers are responsible for adjusting the aggregated values involving functions like identity, amplification, and attenuation. They propose the logarithmic scaler $S(d, \alpha)$:

$$S(d, \alpha) = \left(\frac{\log(d+1)}{\delta} \right)^\alpha, \quad \delta = \frac{1}{|\text{train}|} \sum_{i \in \text{train}} \log(d_i + 1), \quad d > 0, \quad -1 \leq \alpha \leq 1 \quad (6.1)$$

where δ is a normalization parameter computed over the training set, d is the degree of the point/node receiving the message, α is a variable parameter that is negative for attenuation, positive for amplification, or zero for no scaling. The integration of these aggregators can harness a broader spectrum of neighborhood information, leading to improved performance in multiset-related tasks.

However, the preset scalers and aggregators are fixed in their operations, meaning they lack adaptability to datasets or tasks where certain types of aggregations might be more pertinent than others. This data-agnostic strategy leads to sub-optimal performance when dealing with a high number of nodes (points) setup, the encoder does not show a generalizable representation learning. Hence, I introduce the *weighted learnable PNA* aggregation as a part of Yoneda Pooling. This data-centric method introduces learnable weights to the aggregation process, allowing the model to automatically prioritize certain types of aggregations over others based on the data and the task at hand:

$$\bigoplus_{\mathcal{W}} = \underbrace{\begin{bmatrix} I \\ S(D, \alpha = 1) \\ S(D, \alpha = -1) \end{bmatrix}}_{\text{scalers}} \otimes \underbrace{\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}}_{\text{Learnable Weights}} \odot \underbrace{\begin{bmatrix} \mu \\ \sigma \\ \max \\ \min \end{bmatrix}}_{\text{aggregators}} \quad (6.2)$$

Where \oplus_W is representing the weighted (learnable) PNA aggregator, \otimes represents the tensor product, and \odot represents the Hadamard product. This important modification extends the PNA formalism to learn the weights for each aggregation function resulting in dynamic adjustment of the fixed algebraic operations to prioritize certain aggregations over others based on their relevance to the data.

The Yoneda-pooling, depicted in Figure 6.9, along with incorporating the learnable PNA-aggregator, to provide dynamical summary statistics of points for each sample in an event, also introduces an Event-level attribute, $\langle Yon \rangle$. $\langle Yon \rangle$ can be thought of as a global summary statistics learnable token of the whole event that one has access to such as Luminosity, the maximum amount of background, or the type of background hits. In our case, it is the maximum cardinality of each event. After concatenating this token to the other samples in the event (we get now a mini-batch of size $[N_{evt} + 1, ch]$), everything goes into a Gated-MHA module that acts over the samples in the event. By doing this relational reasoning, the representation of samples in the event (mini-batch) will be reweighted with respect to the global (event-level) $\langle Yon \rangle$ token and also their relationship with each other.

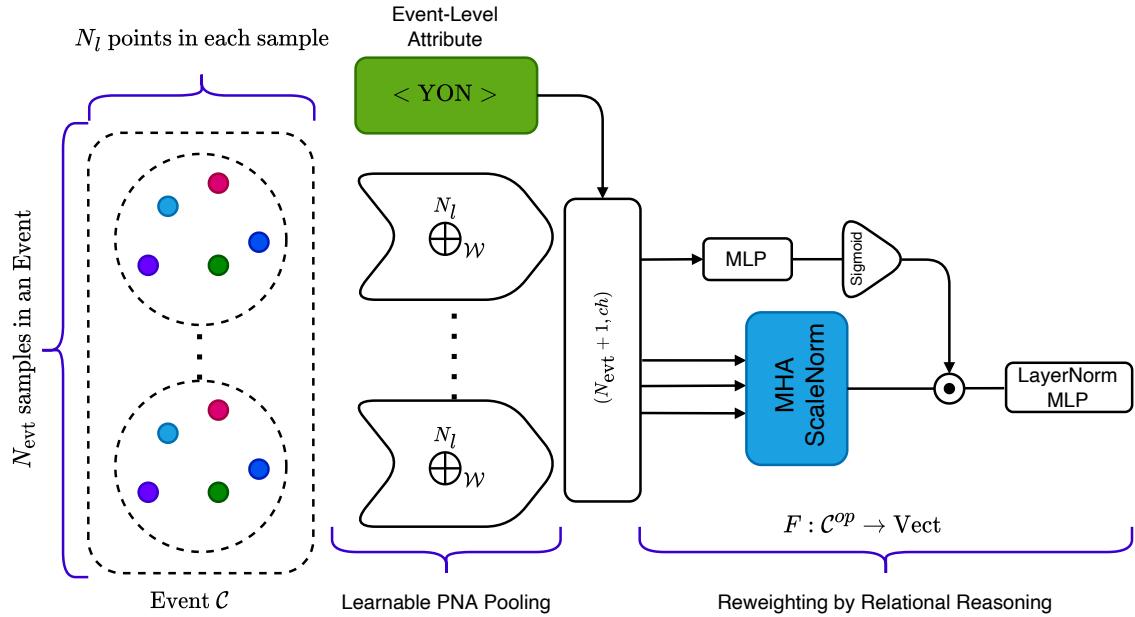


Figure 6.9: Yoneda-Pooling

6.4.2.3 Yoneda-Pooling and Category Theory

In order to create a universal mathematical language for relational reasoning, let's take a step back, and look at what Yoneda pooling is doing. Yoneda pooling is utilizing a learnable PNA aggregation of points, then mapping them to the VAE's latent manifold such each sample in the mini-batch is refined to log the information of the other samples in the

mini-batch (very much like the relational reasoning introduced in the last chapter). This general process is similar to the Yoneda lemma, demonstrating how objects in a category can be fully understood in terms of the morphisms (arrows) that go into them. Here the category \mathcal{C} is an event (self-dual as the morphisms are bidirectional), where the objects are the PXD sensor information represented as a \mathcal{G} -invariant multi-sets where \mathcal{G} is the permutation group \mathcal{S}_n (permutation invariant point clouds). For two G_{n_i} -invariant multisets A and B with multiplicities n_a and n_b , respectively, a morphism $\rho : A \rightarrow B$ is a function such that

- $\rho(g \cdot a) = g \cdot f(a)$ where $g \in G$
- $k \times \text{Multiplicity}(a) = \text{Multiplicity}(f(a))$, where $k = n_b/n_a$

This category is like “a bag of words” in which it sets up the basic framework for objects and morphisms without adding additional structure or interpretation. It specifies the kinds of objects (in this case, G_{n_i} -invariant multi-sets with elements in $[0, 1] \times [0, 1] \times [0, 1]$) and the rules for how they can be transformed. Much like syntax in natural language, which outlines how words can be combined but doesn’t specify their meaning, this category outlines how these specific multi-sets (PXD sensor information) exist individually without detailing what those PXD sensors might represent in a given event. On the other hand, The “contextual representation” or “semantics,” so to speak, would come from how one interprets these objects (PXD sensor information) and morphisms (their relation) in a specific context (event).

The Yoneda embedding P then assigns a vector-valued representation (presheaves) to each object in the category while taking the set of morphisms between the objects, by means of the Attention mechanism. Formally, one can define a presheaf $P : \mathcal{C}^{op} \rightarrow \text{Vect}_k$, where the category Vect_k refers to vector spaces over a field k and Linear transformations as the morphisms. For each object $A \in \mathcal{C}$, P assigns a vector space V_A over the field k . And for each morphism $\rho : A \rightarrow B$ in \mathcal{C} , P assigns a linear transformation $T_\rho : V_B \rightarrow V_A$. Given that P maps each object $A \in \mathcal{C}$ to a vector space $V_A \in \text{Vect}_k$, the presheaf P also associates to each morphism $\rho \in \mathcal{C}$ a k -linear map (denoted as $P(\rho)$) between the corresponding vector spaces $P(\rho) : V_B \rightarrow V_A$. Here $P(\rho)$ satisfies the following functorial properties for P

- Preservation of Identity: $P(\text{id}_A) = \text{id}_{V_A}$ for every object A in \mathcal{C} .
- Preservation of Composition: Given morphisms $\rho_1 : A \rightarrow B$ and $\rho_2 : B \rightarrow C$, one must have $P(\rho_2 \circ \rho_1) = P(\rho_1) \circ P(\rho_2)$.

Hence, given an object $A \in \mathcal{C}$, the Yoneda embedding maps A to the hom-functor $\text{Hom}_{\mathcal{C}}(-, A)$. This functor assigns to each object $B \in \mathcal{C}$ the vector space spanned by affine transformations from B to A , and to each morphism $\rho : B \rightarrow C$ in \mathcal{C} , the corresponding linear transformation induced by composition with g ,

$$\text{Hom}_{\mathcal{C}}(-, A) : \mathcal{C}^{op} \rightarrow \text{Vect}_k$$

$$B \mapsto \text{Hom}_{\mathcal{C}}(B, A)$$

$$g : B \rightarrow C \mapsto \text{Hom}_{\mathcal{C}}(g, A) : \text{Hom}_{\mathcal{C}}(C, A) \rightarrow \text{Hom}_{\mathcal{C}}(B, A)$$

where $\text{Hom}_{\mathcal{C}}(g, A)$ is given by function composition with g . As a result, Yoneda Lemma for Category \mathcal{C} states that for any presheaf $P : \mathcal{C}^{op} \rightarrow \text{Vect}_k$ and any object $A \in \mathcal{C}$, there is a natural isomorphism:

$$\text{Nat}(\text{Hom}_{\mathcal{C}}(-, A), P) \cong P(A).$$

This states that the natural transformations from the hom-functor represented by A to P correspond bijectively to the elements of the vector space $P(A)$. In essence, the Yoneda Lemma captures the idea that a set of points in \mathbb{R}^3 is fully described by how other sets relate to it, reflected in the context of vector spaces and linear transformations in Vect_k . In other words, an object $A \in \mathcal{C}$ is uniquely determined by its relationships to all other objects in \mathcal{C} . Going back to Yoneda-Pooling, in order to encode a sample A in an event, the Yoneda-Pooling mechanism considers all affine transformations from other samples (samples) to A . Thus, learning the information about the transformations of A could serve as a form of *relational compression*. This way, one makes sure that the latent space of the VAE also includes not only local information but also relational information.

From the earlier analogy, syntax tells us “which sensor is near which sensor,” while semantics tells you “what they mean in an event.” In a similar manner, the Yoneda Embedding doesn’t just tell us about the structure (syntax) of the event; it also tells us how to understand each sensor’s information in terms of its relationships/interactions with all other sensors (semantics). Given this, one can say that the Yoneda embedding and the resulting presheaf representation capture some form of “contextual representation” or “semantics” for each event.

6.4.3 YonedaVAE: Set Generation

YonedaVAE’s set generator has two components. The first part is to train a Self-Distilled Transformer decoder to learn to generate the cardinality of each sample given the event-level attribute (such as event luminosity or Max cardinality) as the condition, depicted in Figure 6.10. The second part is to use the Transformer decoder’s latent embedding and the generated (inference) cardinalities or true level (training) cardinalities to do the Top-q sampling (will be discussed shortly in Section 6.4.3.2).

Training During training, via self-distillation (discussed in Chapter 3, Section 3.2.1), the set generator is trained on generating the cardinalities given the event-level attribute and the Normal distribution prior and the VAE encoded latent. Then, it uses the Transformer decoder’s latent embedding and the true cardinalities to do the Top-q sampling. Afterward, the created set, along with the encoded latent (by the Yoneda Encoder) will be fed to the VAE decoder where the output is the generated point cloud.

Inference At inference time, the set generator generates the cardinalities given only the event-level attribute and the Normal distribution prior. Then, after Top-q sampling, based

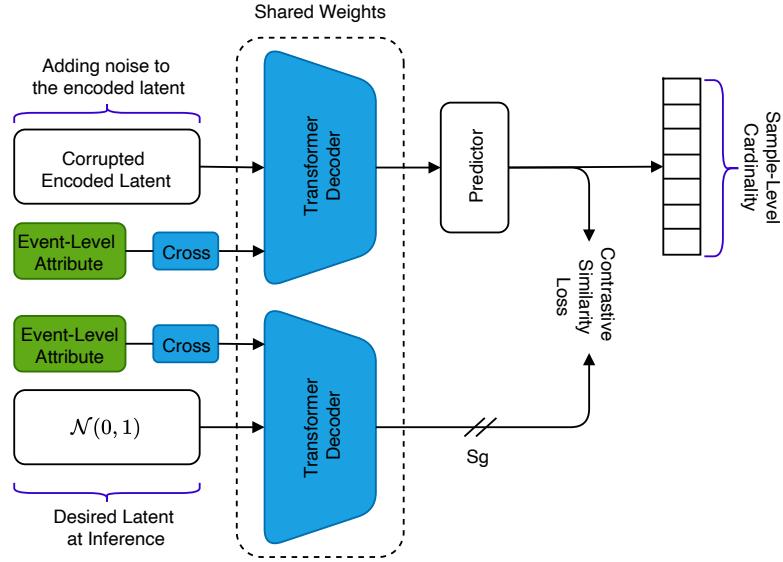


Figure 6.10: Self-Distillation of the learned latent space via the VAE encoder to the desired prior. It is termed “self-distillation” since the backbone models, the Transformer Decoders, share weights.

on the generated cardinalities, the generated set, along with the prior latent (the Normal distribution) will be fed to the VAE decoder where the output is the generated point cloud.

Now, let’s dive into the details step-by-step.

6.4.3.1 Self-Distillation

Since I am targeting OOD detector response simulation, the model needs to be able to generalize and predict each sensor’s number of hits and their correlation with each other accurately. In order to do so, YonedaVAE trains a Transformer decoder to generate them in a causal manner, given the latent vector (prior) and event-level attribute. By causal, I mean that during training the Transformer decoder utilizes a causal masking mechanism where the self-attention layer is only allowed to attend to earlier positions in the output sequence. This is done by masking future positions. This way the model successfully generates the cardinalities of each sensor in an auto-regressive manner. In order to take into account also the topology of the PXD detector (as they are not sequential but distributed in an annulus shape), and learn PXD sensor positions (angles and radius), learnable Rotary positional encodings (discussed in Section 6.2.2) of the PXD sensors are being added to the self-attention.

When one trains the model only given the encoded latent information, the model overfits the data (encoded latent), and cannot generalize at inference where I don’t have access to the true encoded data distribution during the inference. That is why I conjure Self-Supervised Learning, in particular, the *self-distillation* method to train the set generator’s transformer decoder.

Training with self-distillation normally is when one has two different views (positive pairs) of the same data and wants to train an encoder to subsequently use a predictor network to map the output of one encoder to that of the other. The upshot then would be to avoid mode collapse (discussed in Chapter 3, Section 3.2.1). However, in our setup, the trivial representation is the output of the Encoder, encoded latent space. The dual representation over which we want the model to generalize is the normal prior, to which we have access at the inference time. This introduces a new way for the VAE to learn a latent representation to be as similar as possible to a normal prior. In other words, the architecture takes as input two latent distributions that I want the model to consider being isomorphic, the corrupted encoded latent z_1 and the prior Normal distribution z_2 . The two latent spaces are processed by the Transformer decoder network g and a projection MLP head. The Transformer decoder, g , shares weights between the two latents. A prediction MLP head, denoted as h , transforms the output of one latent space and matches it to the other latent space. Denoting the two output vectors as $p_1 \equiv h(g(z_1))$ and $x_2 \equiv g(z_2)$, one minimizes their negative cosine similarity:

$$\mathcal{D}(\mathbf{p}_1, \mathbf{z}_2) = -\frac{\mathbf{p}_1}{\|\mathbf{p}_1\|_2} \cdot \frac{\mathbf{z}_2}{\|\mathbf{z}_2\|_2}, \quad (6.3)$$

where $\|\cdot\|_2$ is ℓ_2 -norm. Similar to SimSiam loss [114], one can define the self-distillation loss as,

$$\mathcal{L}_{SD} = \frac{1}{2} (\mathcal{D}(\mathbf{p}_1, \mathbf{sg}(\mathbf{z}_2)) + \mathcal{D}(\mathbf{p}_2, \mathbf{sg}(\mathbf{z}_1))), \quad (6.4)$$

where \mathbf{sg} represents the stop-gradient operation on the respective variables. Here, the encoder on z_2 receives no gradient from x_2 in the first term, but it receives gradients from p_2 in the second term (and vice versa for z_1).

Incorporating self-distillation then has a vital role for the set generator to be able to extrapolate well beyond the training context by detaching itself from the learned latent space of the Encoder. Since the model has access to the trivial encoded representation of the data through the learned latent vector, it still experiences a mode collapse (since it is easier for the model to map the learned latent space to cardinalities). That is why, as a solution, I propose to corrupt the encoded latent vector by random noise injection to make the training more balanced. From the perspective of the mainstream definition of self-distillation (like in BYOL), this can be viewed as using augmented views of the original input as a positive sample. Since the downstream task for the Transformer decoder is to predict the cardinalities correctly, along with the self-distillation loss, the model also uses the supervised loss as well over the predictor's output,

$$\mathcal{L}_{\ell_2} = \|h(g(z_1)) - \mathbf{e}\|_2^2 = \sum_i^{N_{\text{evt}}=19} (h(g(z_1))_i - e_i)^2, \quad (6.5)$$

where e_i is the true occupancy per sensor. It is important to note that since this is a predictive/generative setting, the set generator uses a Transformer decoder instead of an encoder. The transformer decoder has a causal inductive bias that, along with the

Rotary positional encoding of each sensor, helps to generate cardinalities with the correct correlation. For the Transformer decoder to generalize well to the extrapolation region, it benefits from the attention gating mechanism that I introduced earlier, Rotary positional encoding (to encode PXD sensor positions), and dropouts (embedding dropout, attention dropout, and layer dropout). The contextual event-level attribute, in our case, the max cardinality of the event, also gets injected into the Transformer decoder through the cross-attention mechanism. An interesting observation was that in order to encourage context extrapolation, one has to use Layer Normalization instead of Batch Normalization, which is frequently used in self-distillation setups [91]. I think the reason behind this phenomenon is that batch normalization uses the training batch statistics to learn its parameters (as they do not care about OOD generalization), while Layer Normalization does not rely explicitly on the statistics of the training data and only learns from the individual features through normalization.

6.4.3.2 Adaptive Top-q sampling

After encoding the point clouds and the corresponding attributes, the set generator creates the multiset from the latent vectors during training. Of course, the most important thing at this stage is to generate the points with the correct cardinality. However, the first challenge I face is that within an event, the cardinality of each sample differs from one another. Moreover, since the cardinality of an event describes the occupancy of the detector responses in that event, in the OOD regime, I do not have access to each sensor's occupancy during inference. I only have the maximum cardinality or the luminosity of the event and nothing more. That is why one cannot use any of the described set creation approaches in Section 6.2.1 as either they limit the extrapolation power or they lack the versatility to generate variable intra-event length sets.

Thus, it is of paramount importance for the set generator to be able to not only generate points with inter-event and intra-event variable cardinality but also to learn this in a zero-shot manner to extrapolate beyond the training region, as depicted in Figure 6.6. By inter-event variable cardinality, I mean when one is dealing with variable cardinality in general in the training/test set, and by intra-event variable cardinality, I mean when an event or mini-batch has variable cardinality. The former can be achieved without extrapolation with the Top-k method; however, the latter, handling intra-event variable cardinality, has not been done before and is vital when one is dealing with irregular detector geometries and hit patterns through the detector.

Hence, this study introduces a novel set creation mechanism, *Adaptive Top-q Sampling*. The key idea is to use the shape of the probability distribution to adaptively determine the multiplicity of points to be sampled from. Thus, instead of relying on a fixed and predefined Top-k or using a temperature parameter to control the shape of the distribution without sufficiently suppressing the unreliable tail, I propose sampling from the top-q portion of the reference probability distribution for each event where p is adaptively learned in context, expanding and contracting the candidate pool dynamically.

One might wonder why one needs to sample at all. As I described in Section 6.2.1 and

Chapter 4, in a multi-set (point cloud) generation task, during training or inference, one has two stages of decoding, a sampling stage where the best “candidate” points are being proposed by the set generator, then the update stage where the points become aware of each other and get updated to be mapped to the final output. For creating sets of point clouds stochastically, sampling is essential for several reasons, similar to why it’s used in natural language generation models. Sampling allows the model to capture the inherent variability in possible valid outputs for an event. If we were to deterministically choose the most probable points, we could end up with suboptimal and unrealistic point clouds that don’t capture the complexities of the whole dataset. By sampling from a distribution, the model inherently captures its own uncertainty about the “best” point or set of points, adding a layer of robustness and confidence assessment to the generated set. This is particularly useful in the challenging OOD scenarios where the model has to generalize beyond the training data.

Similarly to Top-k, Adaptive Top-q also uses a reference set with an arbitrary size n_0 , as depicted in Figure 6.11. Each point in this set is a pair (ϕ, \mathbf{r}) : the *reference angle* $\phi \in \mathbb{R}^a$ is used to decide when to select the point, and $\mathbf{r} \in \mathbb{R}^c$ contains the *reference representation* of the point. Given a latent vector $\mathbf{z} \in \mathbb{R}^{l \times 1}$, a reference set made of reference angles $\Phi \in \mathbb{R}^{n_0 \times a}$ and reference representations $\mathbf{R} \in \mathbb{R}^{n_0 \times c}$, as well as learnable matrices \mathbf{W}_1 and \mathbf{W}_2 (respectively of sizes $1 \times c$, $1 \times c$), Adaptive Top-q creation computes

$$\mathbf{a} = \text{MLP}(\text{TransformerDecoder}(\mathbf{z})) \in \mathbb{R}^a \quad (6.6)$$

$$\mathbf{n}_i = \text{TransformerDecoder}(\mathbf{z}) \in \mathbb{R}^{n_{\text{evt}} \times 1} \quad (6.7)$$

$$\mathbf{c} = \Phi \mathbf{a} / \text{vec}((\|\phi_i\|_2)_{1 \leq i \leq n_0}) \in \mathbb{R}^{n_0} \quad (6.8)$$

$$\tilde{\mathbf{c}} = \text{softmax}(\mathbf{c}) \in \mathbb{R}^{n_i \times 1} \quad (6.9)$$

$$\mathbf{s}_i = \text{argsort}_{\downarrow}(\mathbf{c})[: n_i] \in \mathbb{N}_i^n \quad (6.10)$$

$$\mathbf{X}_{n_i} = \mathbf{R}[\mathbf{s}_i] \odot \tilde{\mathbf{c}} \mathbf{W}_1 + \tilde{\mathbf{c}} \mathbf{W}_2 \in \mathbb{R}^{n_i \times c} \quad (6.11)$$

$$\mathbf{X}^0 = \text{ZeroPad}(\mathbf{X}_{n_i}) \in \mathbb{R}^{n \times c}. \quad (6.12)$$

In each event, the Transformer decoder adaptively proposes a set of cardinality for each sample in the event and a set of vectors (Transformer decoder’s intermediate layer), termed Angles, given the event-level attribute (max cardinality) and the latent space (or prior during inference). Then, the model computes the cosine similarity between these candidate’s angles and the reference angles. Then, based on their normalized similarity, the top n_i important points will be chosen for each sample. Note that, in Adaptive Top-q sampling, for each sample i , n_i differs and represents the top n_i portion of the reference probability mass. In the end, all samples get zero-padded to the maximum cardinality of the event for vectorization purposes.

The crux of the Adaptive Top-q module is to select the points adaptively that will be used to generate a dynamic set with variable multiplicity. This is the intra-event variability I was talking about. In practice, this means selecting the highest probability points whose multiplicity does not exceed the learned intra-event n_i for sample i . The size of the sampling

set will adjust dynamically based on this criteria at each time step. For instance, for low values of n_i , this is a small subset of points that takes up a majority of the reference probability mass.

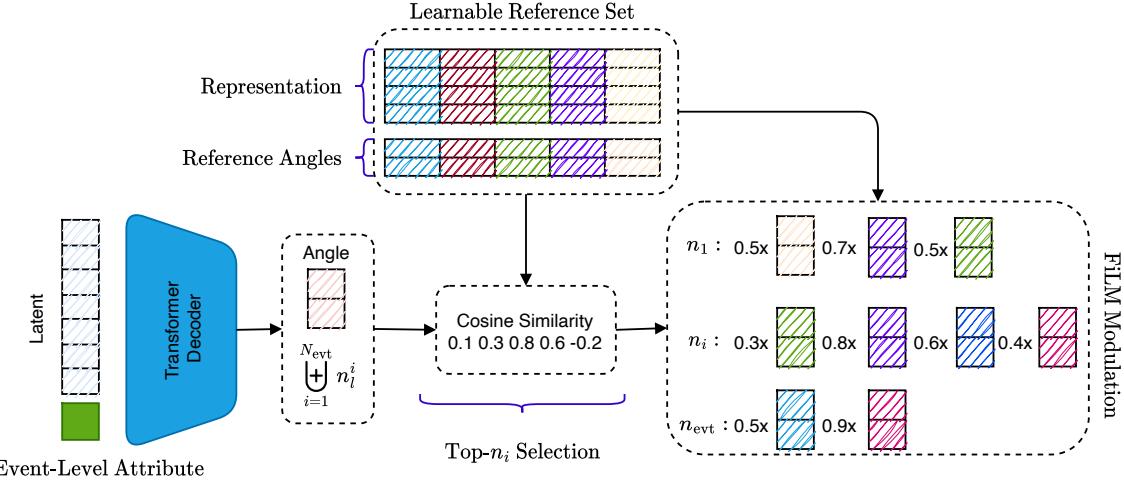


Figure 6.11: Adaptive Top-q sampling proposes a point sampling from the top n_i portion of the probability mass where n_i is adaptively learned in context, dynamically expanding and contracting the candidate pool in each event.

6.4.4 YonedaVAE: Decoder

The decoder of YonedaVAE comprises three main parts: the sample-level feature injection, set-latent space fusion, and the point cloud final decoder.

Training At training time, the decoder can access the true sample-level features. They go through a Transformer encoder to encode their relationship. Afterward, they will be injected into the created set by the set generator via the cross-attention mechanism to create an enriched set. While previous works usually append the latent vector to the enriched set, this study exploits the equivalence between summation and concatenation when applying a linear layer. Hence, the enriched set gets fused to the latent vector (still corrupted due to preventing overfitting) by a feature-wise linear modulation (FiLM) layer [370], which writes

$$\text{cat}(\mathbf{X}^0, \mathbf{1}_n \mathbf{z}^T) \mathbf{W} = \mathbf{X}^0 \mathbf{W}_1 + \mathbf{1}_n (\mathbf{z}^T \mathbf{W}_2), \quad (6.13)$$

for $\mathbf{W} = \text{cat}(\mathbf{W}_1, \mathbf{W}_2)$. Contrary to the concatenation (left-hand side), the sum (right-hand side) does not compute $\mathbf{z}^T \mathbf{W}_2$ several times, which reduces the complexity of this layer from $O(n(c + l)c)$ to $O(nc^2 + cl + nl)$. As a result, it combines the sum and multiplicative modulation in a FiLM layer to build a more expressive model. This feature-wise affine transformation based on conditional information generalizes well to challenging, new data of zero-shot regime. Then, the latent-injected enriched set goes into a Transformer Encoder network with Scale normalization to learn to produce output points in \mathbb{R}^3 with the proper

inter-dependencies, as depicted in Figure 6.12. The Transformer Encoder learns this point-by-point association by benefiting from the gated self-attention mechanism. In order to reach length extrapolation, as I discussed at the beginning of the section, one needs to incorporate a positional encoding mechanism. After several trials and in-depth exploration of different methods, I incorporated a modified version of ALiBi [325] embedding to be used in the bidirectional setup of the Transformer Encoder. Since it was originally proposed for autoregressive setups with causal masking, the modified version adopts it for the bidirectional setup of the Transformer Encoder without a causal mask. Another vital factor for extrapolation in this module is the dropout operation. The Transformer Encoder module benefits from 0.1 point embedding dropout, 0.1 layer dropout, and 0.1 Attention dropout.

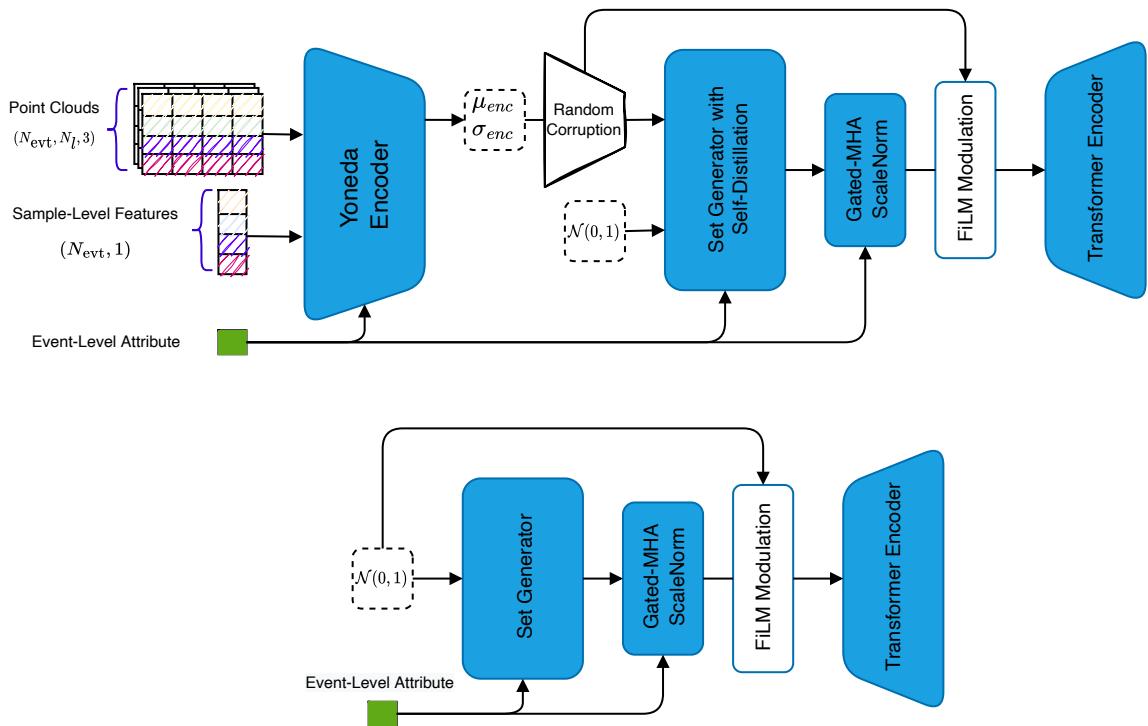


Figure 6.12: YonedaVAE in training (top) and inference (bottom)

For loss functions, the model needs to incorporate a permutation invariant reconstruction loss, as discussed in Chapter 3. YonedaVAE benefits from the Hungarian loss Equation (3.10), as the main reconstruction loss and the differentiable geometric loss based on Optimal Transport [371] as a loss over only the third channel of the point cloud, the charge. The version of geometric loss that this study uses computes the Sinkhorn divergence between two empirical measures. Given two sets of points $\mathcal{X} = \{x_1, x_2, \dots, x_m\}$ and $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ in \mathbb{R}^d , the empirical measures are $\mu = \frac{1}{m} \sum_{i=1}^m \delta_{x_i}$ and $\nu = \frac{1}{n} \sum_{j=1}^n \delta_{y_j}$. The Sinkhorn divergence $S_\epsilon(\mu, \nu)$ is defined as:

$$S_\epsilon(\mu, \nu) = W_\epsilon^p(\mu, \nu) - \frac{1}{2} W_\epsilon^p(\mu, \mu) - \frac{1}{2} W_\epsilon^p(\nu, \nu), \quad (6.14)$$

Where $W_\epsilon^p(\mu, \nu)$ is the entropically-regularized Wasserstein distance defined by:

$$W_\epsilon^p(\mu, \nu) = \min_{\gamma \in \Pi(\mu, \nu)} \langle \gamma, C \rangle + \epsilon H(\gamma)$$

Where, $\langle \gamma, C \rangle = \sum_{i=1}^m \sum_{j=1}^n \gamma_{ij} c(x_i, y_j)$ is the dot product of γ with the cost matrix C , where $c(x_i, y_j) = \|x_i - y_j\|^p$, $\Pi(\mu, \nu)$ is the set of joint measures with marginals μ and ν , $H(\gamma) = -\sum_{i,j} \gamma_{ij} \log(\gamma_{ij})$ is the entropy of γ , and ϵ is the regularization parameter. The optimal transport problem for γ is typically solved using the Sinkhorn-Knopp algorithm [372] to approximate the Wasserstein distance in high-dimensional spaces effectively with time complexity $O(n \log n)$. The parameter $p = 2$ specifies the power of the distance $\|x_i - y_j\|^p$ in the cost function $c(x_i, y_j)$. I apply this geometric loss only to the charge loss since, upon experiments, having this loss over all channels of the point cloud confuses the model and makes the training very long. However, since the essence of the third channel (charge) is somewhat different than the first two channel (x and y dimension of the PXD), using the geometric loss over the third channel actually acts as a learning disentanglement force for the model to learn the difference between charge and coordinate channels. Without this geometric loss over Charge, the model could not capture the full distribution of the PXD charge, especially the long-tail (high intensity) of the distribution.

For the VAE's prior regularization term (discussed in Chapter 3), I incorporate the Maximum mean discrepancy Variational Autoencoder (MMD-VAE) [373, 140] regularization approach. Maximum mean discrepancy (MMD) [373] is based on the idea that two distributions are identical if and only if all their moments are the same. Therefore, one can define a divergence by measuring how different the moments of two distributions $p(z)$ and $q(z)$ are. MMD can accomplish this efficiently via the kernel embedding trick:

$$\text{MMD}(p(z) \| q(z)) = \mathbb{E}_{p(z), p(z')}[k(z, z')] + \mathbb{E}_{q(z), q(z')}[k(z, z')] - 2\mathbb{E}_{p(z), q(z')}[k(z, z')] \quad (6.15)$$

where $k(z, z')$ is any universal kernel, such as the Gaussian kernel $k(z, z') = e^{-\frac{\|z-z'\|^2}{2\sigma^2}}$. A rough intuition of MMD is that if two distributions are identical, then the average “similarity” between samples from each distribution should be identical to the average “similarity” between mixed samples from both distributions. MMD regularization loss always prefers to maximize the mutual information between the data and the prior and avoid overfitting the data during training. Notice that the self-distillation mechanism that was discussed earlier has the same objective.

Hence, in the end, YonedaVAE benefits from the Hungarian loss (Equation (3.10)) and geometric loss (Equation (6.14)) as reconstruction losses, MMD loss (Equation (6.15)) as regularization loss, and Self-Distillation loss (Equation (6.4)) in conjunction with the supervised auxiliary loss to learn cardinalities (Equation (6.5)), all together as follows:

$$\mathcal{L}_{\text{YonedaVAE}} = \mathcal{L}_{\text{Rec}} + \mathcal{L}_{\text{Reg}} + \mathcal{L}_{\text{SelfD}} \quad (6.16)$$

$$\mathcal{L}_{\text{Rec}} = \sum_{i=1}^m \sum_{j=1}^n \min_{\pi} d(x_i, y_j) + S_{\epsilon}^{\text{geom}}(\mu, \nu) \quad (6.17)$$

$$S_{\epsilon}^{\text{geom}}(\mu, \nu) = W_{\epsilon}^p(\mu, \nu) - \frac{1}{2}W_{\epsilon}^p(\mu, \mu) - \frac{1}{2}W_{\epsilon}^p(\nu, \nu) \quad (6.18)$$

$$\mathcal{L}_{\text{Reg}} = \text{MMD}(p(z), q(z)) + D_{\text{KL}}(p(z) \| q(z)) \quad (6.19)$$

$$\text{MMD}(p(z), q(z)) = \mathbb{E}_{p(z), p(z')}[k(z, z')] + \mathbb{E}_{q(z), q(z')}[k(z, z')] - 2\mathbb{E}_{p(z), q(z')}[k(z, z')] \quad (6.20)$$

$$\mathcal{L}_{\text{SelfD}} = \frac{1}{2} (\mathcal{D}(\mathbf{p}_1, \text{sg}(z_2)) + \mathcal{D}(\mathbf{p}_2, \text{sg}(z_1))) + \|\mathbf{h}(g(z_1)) - \mathbf{e}\|_2^2. \quad (6.21)$$

Inference The only differences for the decoder at the inference time are the decoder’s lack of access to the true sample-level features and the learned latent space. In the inference time, the decoder has only access to the generated sample-level features by the set generator and the Normal latent distribution, that is, the prior distribution of the VAE. Everything else is exactly the same as the training time setup.

With YonedaVAE, one is now able to not only emulate PXD background signatures given the desired event profile but also simulate OOD PXD responses. The following section demonstrates the extent of success of YonedaVAE on OOD simulation (Length and context extrapolation), on low-level hit topology and geometry, and on the downstream physics analysis.

6.5 Evaluation Results

In order to evaluate the performance of YonedaVAE in generating ultra-high-granularity detector responses of the Pixel Vertex Detector, shown in Figure 6.13, this section provides results divided into two parts, “length extrapolation” and “context extrapolation”. As discussed in Section 6.4.1, with the length extrapolation, YonedaVAE has access to the sample-level occupancy/cardinality during inference and has to extrapolate to point cloud cardinalities beyond the training data. So, the extrapolation here is that the model has to infer how an event with a higher (than training) cardinality looks like. With the context extrapolation, YonedaVAE has only access to the event-level attribute (amount of background per event as max cardinality of the event) during inference and has to learn to generate sample-level occupancy/cardinality by itself in a zero-shot manner and extrapolate to point cloud cardinalities and intra-event correlation beyond the training data. So, the extrapolation here is that the model has to produce the correct distribution of cardinalities over all sensors, given only the maximum cardinality per event. In general, from the last chapter’s fine-grained image analysis perspective (Chapter 5, Section 5.2.1), one can say that YonedaVAE is now conditioned on both the sensor’s location as meta-conditions and the amount of PXD background as subordinate conditions.

This thesis does the evaluation by first going through the same analysis that I did in the last chapter, namely the low-level metrics at the hit level such as occupancy and charge distribution, mean occupancy and charge per sensor, sensor-by sensor intra-event correlation, FID [22] and KID [23]. In this chapter, I also introduce two new metrics for detector simulation, namely the topological and geometrical profiling of the PXD background signatures and Vendi Score [24], a diversity measure to quantify the diversity of the generated samples.

At high-level, this thesis goes through the downstream physics analysis of track reconstruction and studies the performance of YonedaVAE's generated samples, trained on experiment 12 random trigger data- recorded in experiment number 12 of Belle II data taking, with peak recorded luminosity of $1.42 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$, and evaluated on experiment 26 random trigger data- recorded in experiment number 26 of Belle II data taking, with almost double peak luminosity $2.68 \times 10^{34} \text{cm}^{-2}\text{s}^{-1}$ in order to demonstrate the extrapolation abilities of YonedaVAE.

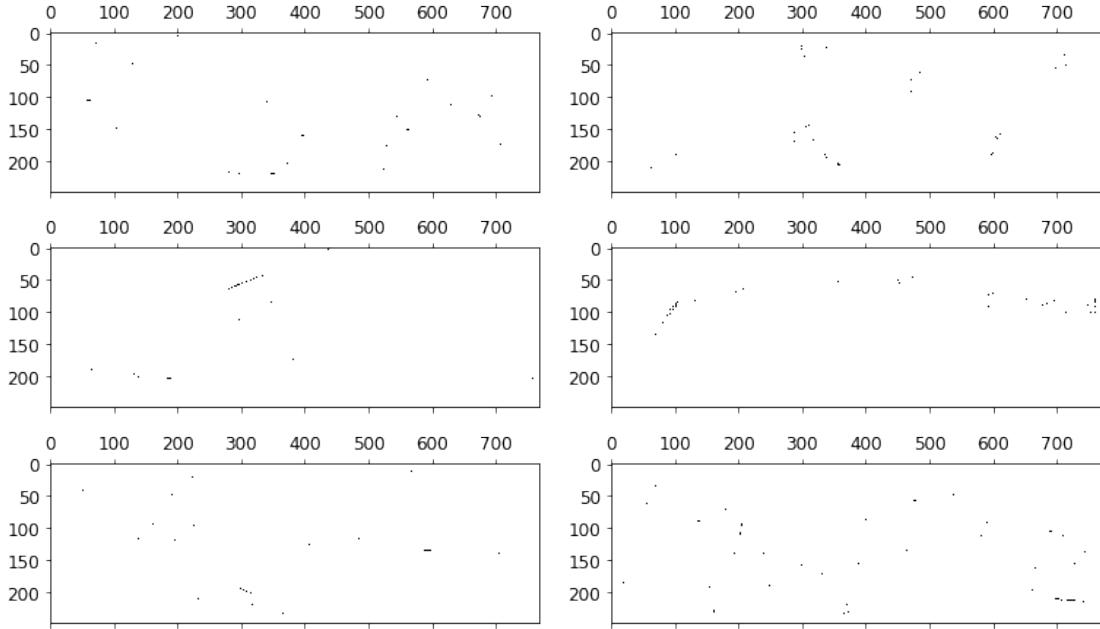


Figure 6.13: Random-trigger (real) PXD background (left) and YonedaVAE generated PXD background (right) in a bitmap format. As stated before, the sparsity of the real PXD background is much less than the Geant4 version (last chapter).

6.5.1 Marginal Distributions

As the first layer of evaluation, the hit-level marginal distributions are compared. The low-level metrics at the hit level are the occupancy distribution (a) and charge distribution in linear (c) and log (d) scale, mean cardinality (b), and charge (c) per sensor and sensor-by-sensor intra-event correlation. I analyze both Pearson (linear) and Spearman (monotonic)

sensor-by-sensor mean occupancy correlation. As a reminder, I have to emphasize that during training, the model only has access to events with up to 400 sensor cardinality (or 7600 event cardinality). This is done to make sure that the model does not *see* any data with cardinality beyond 400. During validation, the model is asked to generate point clouds with arbitrary cardinality over the Experiment 12 dataset. In Figure 6.14 and Figure 6.15 model has access to the cardinality distribution and is sampling from it. In Figure 6.16 and Figure 6.17 model does not have access to the cardinality distribution and only has access to the max occupancy of each event (event-level attribute). As depicted, YonedaVAE captures all the marginal distributions with good precision for the training and validation data.

As mentioned before, to demonstrate extrapolation beyond the training data, YonedaVAE generates previously unseen PXD background hits from experiment 26 (with almost double luminosity) in two regimes. First with length extrapolation regime where the model samples sensor-level features (sensor occupancy) depicted in Figure 6.18, Figure 6.19, and Figure 6.23. As shown in Figure 6.19 and Figure 6.23, Experiment 26 interestingly has a different linear and monotonic intra-event correlation profile with higher linear correlation than the monotonic one. These results demonstrate that while the model is sampling sensor-level occupancies, it can *fill* the previously unseen spots and infer how a background event with a higher cardinality looks like. For generic physics analysis where the analyst wants to do PXD background generation on the fly for analysis, conditioned on the amount of background and the geometry (sensor's position) of the PXD, this result shows one does not need to retrain the generative model for each experiment as far as there will be no strange beam-background issues. I have to note that this is from the perspective of marginal and low-level distributions. In the following, I also discuss other low-level and physics-level metrics to further study the results.

Eventually, with context extrapolation model has access only to event-level attribute (maximum cardinality of the event) depicted in Figure 6.21, Figure 6.22, and Figure 6.23. Thus, YonedaVAE not only needs to generate events with the desired amount of background but also has to learn to generalize to attribute profiles beyond the training data and generate point clouds with the correct cardinality and intra-event correlation. I have to again emphasize that during inference, the model is asked to generate events with up to 5300 sensor cardinality (or 100700 event cardinality) coming from Experiment 26. This is almost the 3% background occupancy limit discussed in Chapter 2. Context extrapolation results show that YonedaVAE can recover (generate) sensor-level information given only the event-level attribute very much like Inverse Problems [374, 375]. The inverse problem refers to using coarse observations to infer the values or the parameters that characterize the finer levels of the system and to estimate data that are not directly observed. For a physics analysis where the analyst wants to do PXD background generation on the fly of analysis, conditioned on the amount of background and the geometry (sensor's position) of the PXD, this result provides evidence that one can also generate PXD background for around 100K event cardinality for luminosities beyond the current threshold of the experiment. There is still a minor discrepancy between the intra-event correlation in the context extrapolation regime. However, it's important to note that the model is not without

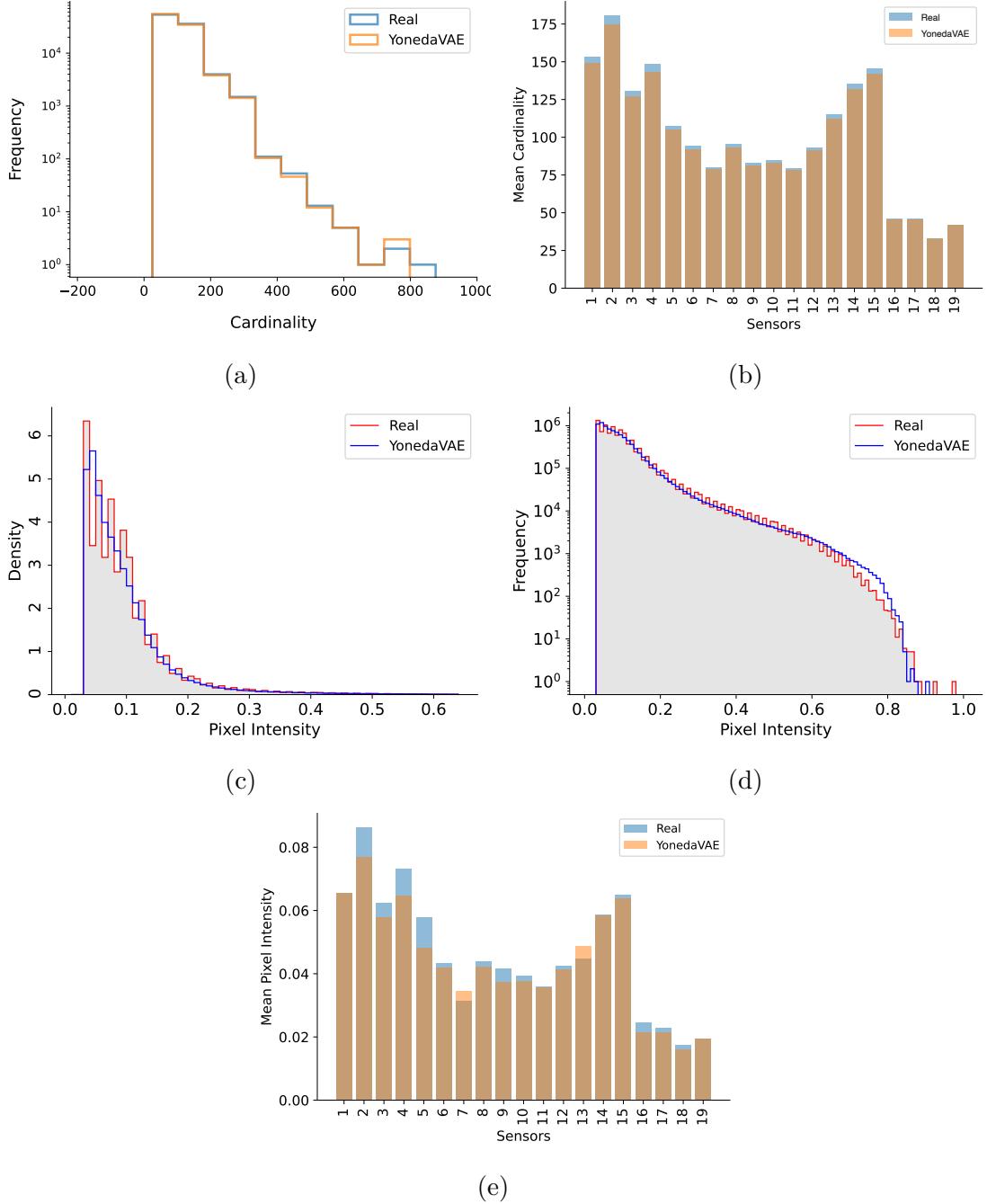


Figure 6.14: Marginal Distributions for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to sensor occupancies as sensor-level features.

its limitations. There are still minor inconsistencies in capturing the intra-event correlation in the context extrapolation domain. Despite these shortcomings, achieving the current level of accuracy in context extrapolation is an extraordinary feat in itself and represents a

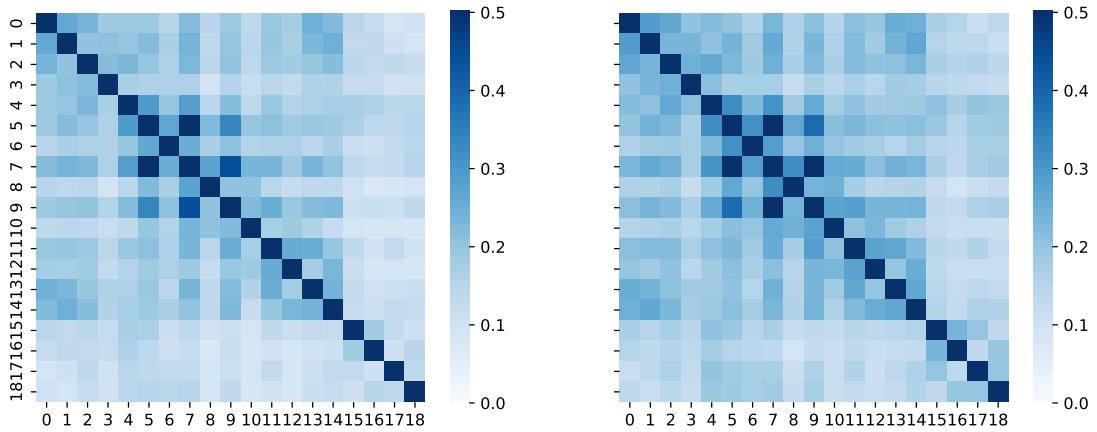


Figure 6.15: Pearson Correlation between sensor's mean occupancy for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to sensor occupancies as sensor-level features.

significant advance in the field. Again, I have to note that this is from the perspective of marginal and low-level distributions. I also discuss other low-level and physics-level metrics to further consolidate these results.

6.5.2 FID, KID, and Vendi Score

To compute the FID and KID, again, I incorporate the Clean-FID project [296], fine-tuned on the training and test PXD data with more than 100,000 events. Thus, the FID model in this setup is aware of both OOD and ID distribution of data. Since the YonedaVAE generates PXD responses in point cloud (continuous) representation in \mathbb{R}^3 , I quantize each point cloud into a grid-like representation of size $1 \times 250 \times 768$. The downstream task for this fine-tuning is multi-class classification between 19 PXD images.

Despite the fact that FID and KID implicitly measure the diversity and explicitly the quality of the generated samples, having a strong and reference-free metric is very absent in the Detector Simulation and event generation domain. Diversity is a criterion that has long been sought after in generative models, especially in surrogate modeling. A lack of diversity in generated samples can hinder the usefulness of surrogate models in large sampling regimes. It is, therefore, important to be able to measure diversity. Thus, to quantify the diversity of generated samples, I incorporate the *Vendi Score* for detector simulation, originating from ecology and statistical quantum mechanics. Although diversity is a vital property in detector simulation, limited efforts have been devoted to its rigorous understanding and quantification. The Vendi Score was first articulated in Friedman et al. [24]. The metric is formalized as e^H , where H is the Shannon entropy of the eigenvalues derived from a similarity matrix, which itself is generated using a data-centric similarity function. This makes the Vendi Score intrinsically connected to quantum statistical mechanics by being analogous to e^S , where S is the von Neumann entropy of a quantum system. In ecology,

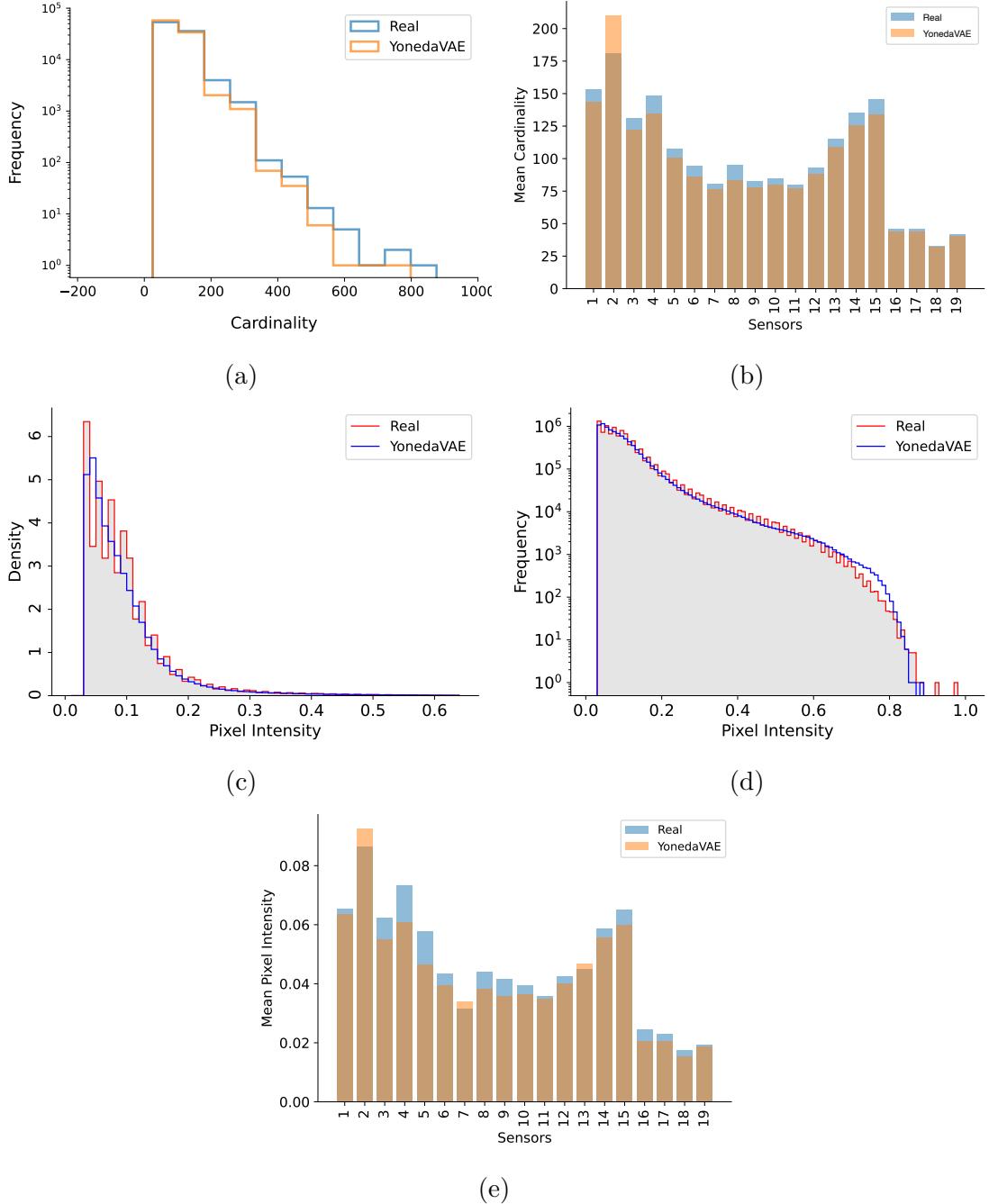


Figure 6.16: Marginal Distributions for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to max cardinality per event as event-level features.

diversity is commonly described as $e^{\log(n)}$, where n is the entropy of the species distribution under study. This serves as an adequate measure of ecological diversity. For a population evenly distributed across n species, the maximal diversity would be n , equivalent to a

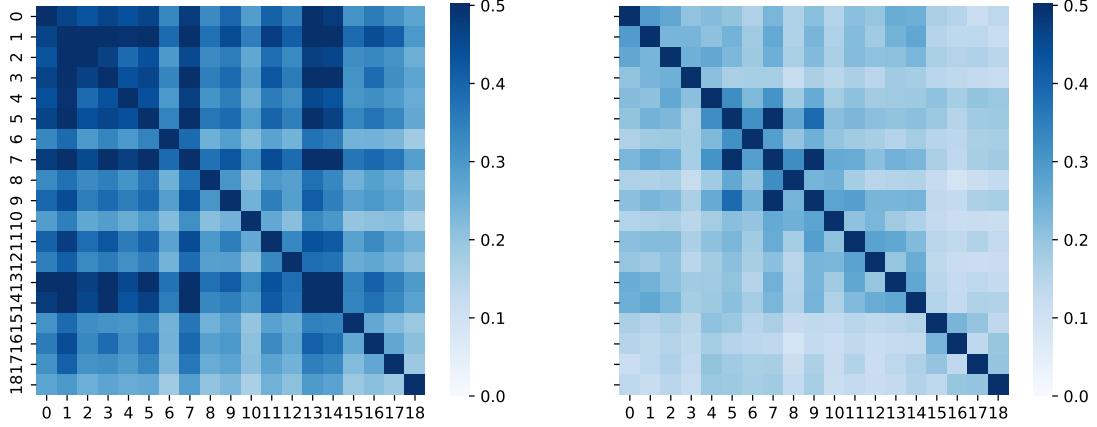


Figure 6.17: Pearson Correlation between sensor's mean occupancy for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to max cardinality per event as event-level features.

population comprising n distinct species. The metric decreases as the species distribution deviates from uniformity, reaching a minimum value of one when all members belong to a single species. Ideally, a diversity metric should rely solely on the samples under evaluation and should attain its maximum value when the samples are entirely diverse and its minimum value when all samples are the same. Hence, the Vendi Score is introduced as a solution to this problem.

Definition 1 (Vendi Score). Let $x_1, \dots, x_n \in \mathcal{X}$ denote a collection of samples, let $k : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ be a positive semi-definite similarity function, with $k(x, x) = 1$ for all x , and let $\mathbf{K} \in \mathbb{R}^{n \times n}$ denote the kernel matrix with entry $K_{i,j} = k(x_i, x_j)$. Denote by $\lambda_1, \dots, \lambda_n$ the eigenvalues of \mathbf{K}/n . The Vendi Score (VS) is defined as the exponential of the Shannon entropy of the eigenvalues of \mathbf{K}/n :

$$\text{VS}_k(x_1, \dots, x_n) = \exp \left(- \sum_{i=1}^n \lambda_i \log \lambda_i \right), \quad (6.22)$$

where I use the convention $0 \log 0 = 0$.

To understand the validity of the Vendi Score as a mathematical object, note that the eigenvalues of \mathbf{K}/n are non-negative (because k is positive semi-definite) and sum to one (because the diagonal entries of \mathbf{K}/n are equal to $1/n$). The Shannon entropy is, therefore, the Vendi Score is well-defined. In this form, the Vendi Score can also be seen as the effective rank of the kernel matrix \mathbf{K} . The effective rank of a matrix is defined as the exponential of the entropy of the normalized singular values. To interpret the Vendi Score it represents the effective count of non-identical elements within the evaluated sample. For instance, a sample with ten unique elements would yield a Vendi Score of 10, a figure that remains constant even when elements are duplicated.

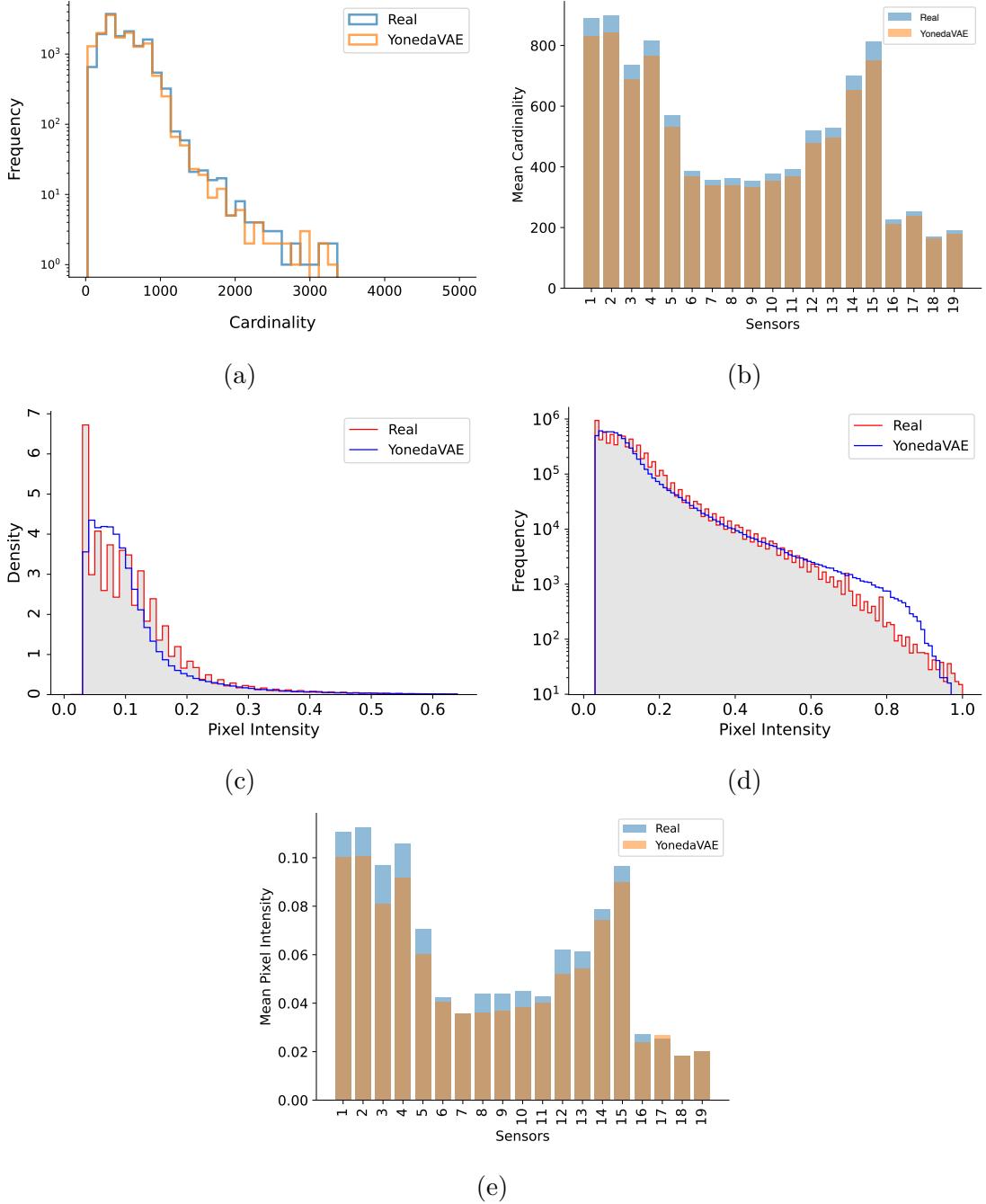


Figure 6.18: Marginal Distributions for YonedaVAE with length extrapolation on the OOD data (Experiment 26) with access to sensor occupancies, as sensor-level features.

In the case of PXD detector simulation, I incorporate the Vendi score and interpret it as the effective number of dissimilar elements in the PXD dataset. For example, if all PXD samples show that they belong to 3 clusters (from high-level to low-level clustering), the

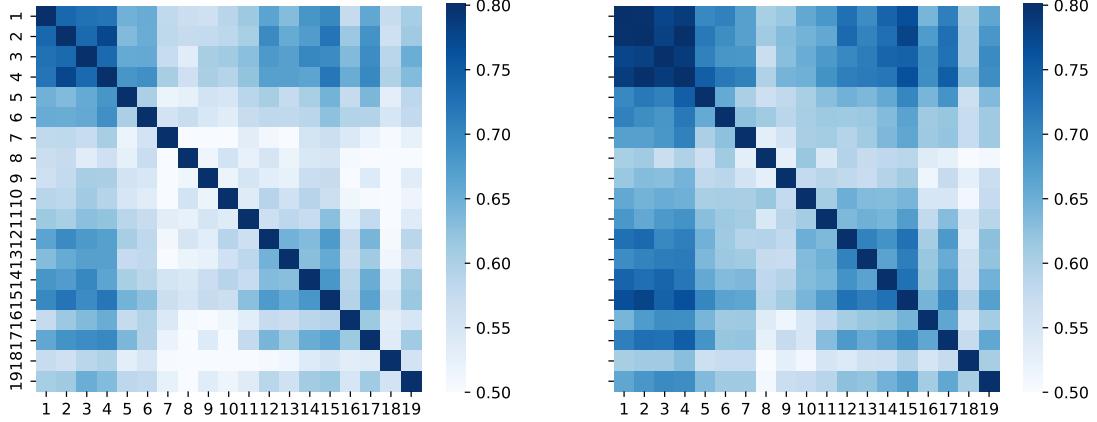


Figure 6.19: Pearson Correlation between sensor's mean occupancy in the length extrapolation regime for YonedaVAE on the OOD data (Experiment 26)

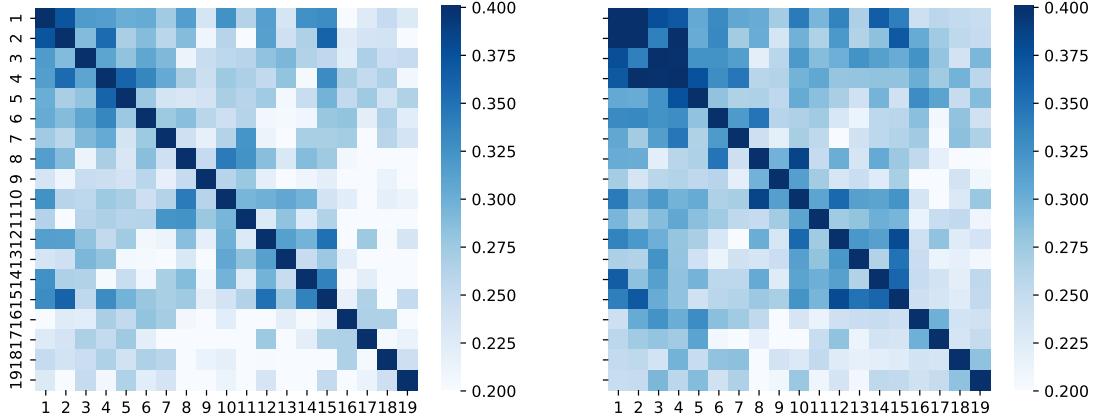


Figure 6.20: Spearman Correlation between sensor's mean occupancy in the length extrapolation regime for YonedaVAE on the OOD data (Experiment 26)

VS also shows 3 modes as exemplified in Figure 6.24. One can understand VS over the FID embedding of the samples to quantify the inter-event diversity. The goal here is to provide a measurable metric to quantify the diversity of generated samples using the covariance matrix from the point of view of a model that had access to both the training and test data. In this case, the kernel matrix is the covariance matrix, and the sample feature is the FID embedding. This way, I get a Vendi score for each sensor based on a large number of samples. In Table 6.3, I report this number.

I compare YonedaVAE with three other models with the previously unseen real PXD detector data coming from experiment 26. The baselines are the SOTA in the conditional point cloud generation, Transformer Set Prediction Network (TSPN) with i.i.d sampling [318], TSPN with Top-K set creation [94], and the previous PXD heatmap generation model (trained now on the real PXD data), IEA-GAN. All these models are being compared in the length

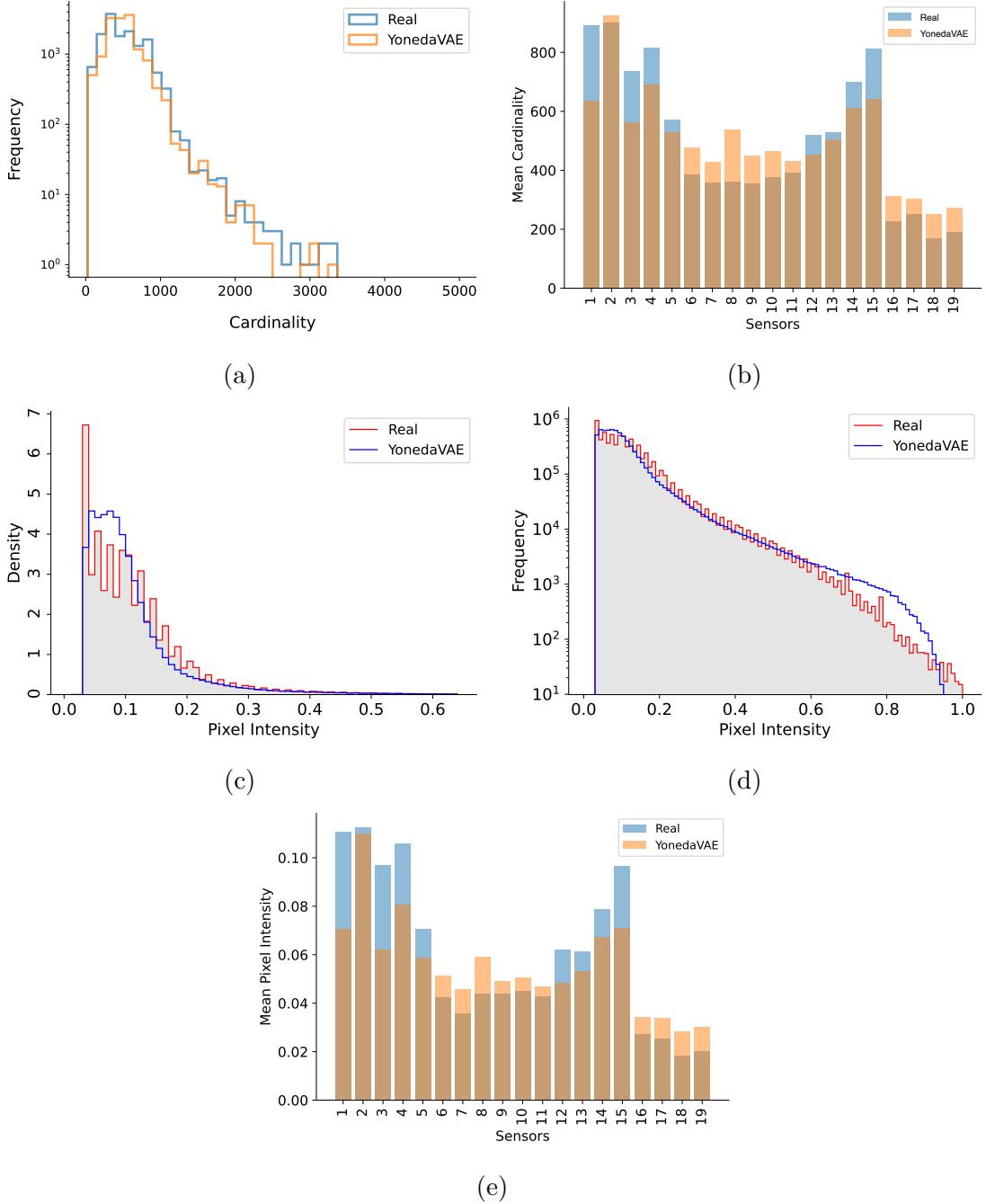


Figure 6.21: Marginal Distributions for YonedaVAE with context extrapolation on the OOD data (Experiment 26).

extrapolation (le) setup, where they are conditioned on the sample-level feature (occupancy per sensor) during inference. Other than YonedaVAE, none of the other models can handle the context-extrapolation (ce) setup where only the event-level attribute in the OOD region

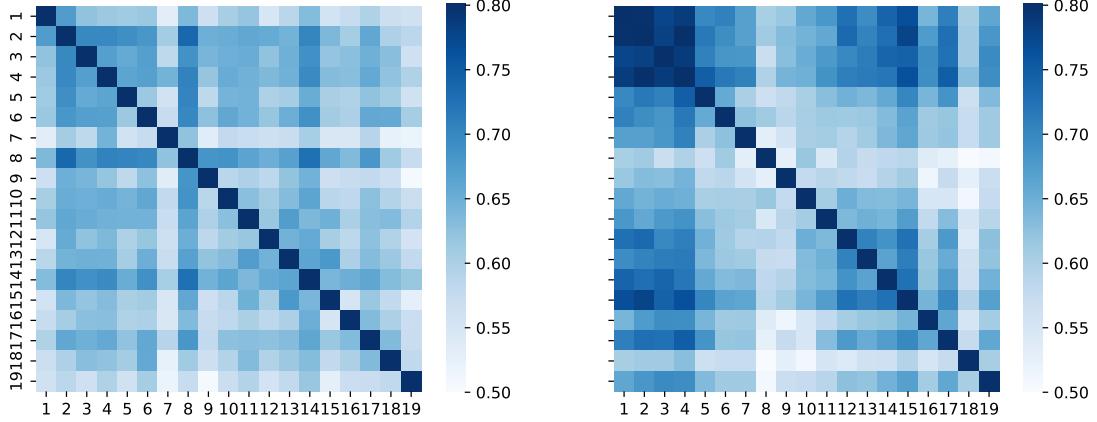


Figure 6.22: Pearson Correlation between sensor's mean occupancy in the context extrapolation regime for YonedaVAE on the OOD data (Experiment 26)

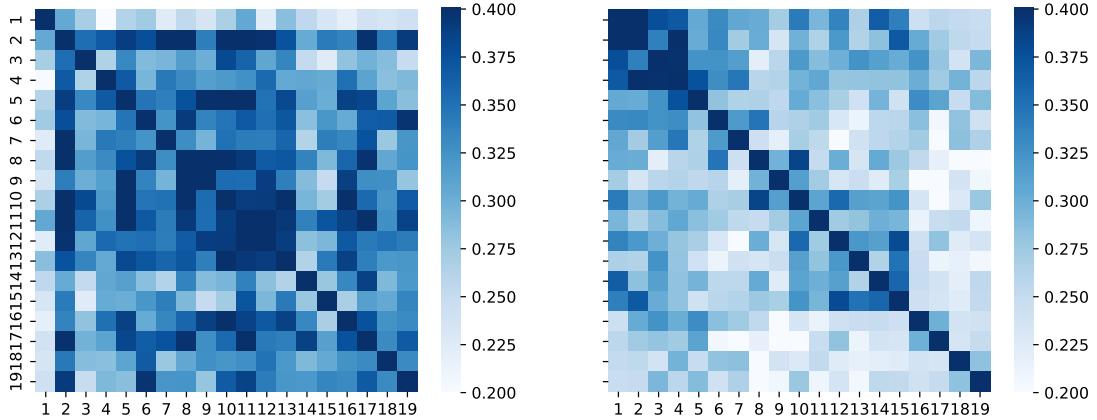


Figure 6.23: Spearman Correlation between sensor's mean occupancy in the context extrapolation regime for YonedaVAE on the OOD data (Experiment 26)

is present during inference.

Table 6.1: FID, KID, and VS score comparison between SOTA models for PXD generation averaged across four random seeds. They are computed over 5000 generated events. The lower the FID and KID, the better the sample quality and diversity. For the VS score, higher values indicate better sample diversity.

	TSPN (i.i.d)	TSPN (Top-k)	IEA-GAN	YonedaVAE ^{le}	YonedaVAE ^{ce}	Test Data
FID	49.46 ± 0.29	41.40 ± 0.48	37.84 ± 0.98	20.19 ± 0.31	22.49 ± 0.11	0
KID $(\times 10^{-4})$	339 ± 7	312 ± 1	283 ± 8	130 ± 2	132 ± 4	0
VS	2.81	3.25	1.71	4.91	4.73	3.86

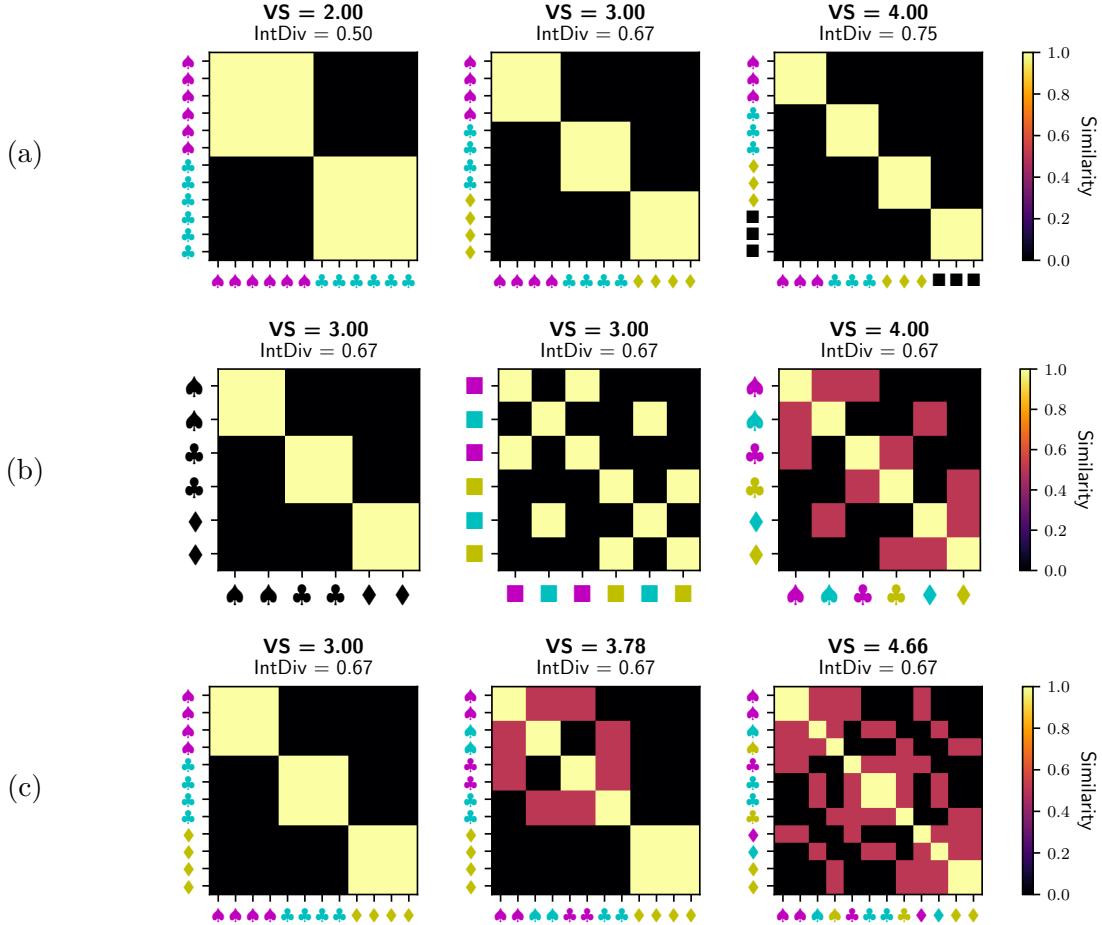


Figure 6.24: (a) The Vendi Score, VS in the figure, adopted from [24], can be interpreted as the effective number of unique elements in a set of events. It increases linearly with the number of modes in the dataset. IntDiv, the expected dissimilarity (inverse correlation), becomes less sensitive as the number of modes increases, converging to 1 (or becoming uncorrelated). (b) Combining distinct similarity measures can increase the Vendi Score, as should be expected of a diversity metric while leaving IntDiv ((inverse correlation)) unchanged. (c) Vendi Score takes into account correlations between samples. The Vendi Score is highest when the samples differ in many attributes and the attributes are not correlated with each other.

The Table 6.3 demonstrates that generated samples by YonedaVAE have the lowest FID and KID score compared to the other models both in context extrapolative and length extrapolative setups. Moreover, YonedaVAE also generates samples that with higher dissimilar modes. VS here means that across 5000 generated events, the number of main clusters of modes (diversity of modes) in the test data is 3.86. A very interesting observation was that VS was increasing as YonedaVAE was generating more samples. For instance, the VS was 3.42 in 100 samples, then it increased to 4.01 in 1000 samples and 4.91 in 5000

samples. This result could be in accord with [245] where generated samples by YonedaVAE outperform samples from a limited amount of test data from the diversity perspective. IEA-GAN, despite being conditioned on sensor occupancy trained on the real data (with very high instability), as expected, performs poorly. The main reason for this is that the real PXD background is much more sparse and diverse than the Geant4 simulated one, which makes it much harder for the IEA-GAN setup to adapt.

One can also utilize VS in order to quantify the diversity over the inter-sensor mean occupancy correlation, discussed in Section 6.5.1. The idea here is to measure the number of clusters or unique sensors in an event that contribute to the mean occupancy correlation and compare the true value with the generated value. In this case, the kernel matrix is the correlation matrix, and the features for each sensor are the distribution of mean occupancy per sensor over a large number of samples. This way, one can measure the diversity of sensor information in an event based on the intra-event occupancy correlation. In this scenario, the maximum VS diversity would be 19 (where no commonality is there between 19 sensors in an event), and the minimum VS diversity would be 1 (all sensors share the same information). Note that the VS results reported in Table 6.3, is different. There, the VS counts the number of modes across the whole dataset. It is like saying that over all the PXD background data, there are 3.86 unique clusters of information using the covariance matrix over their FID (inception-v3) embedding as the measure. While the results in Table 6.2 report that in one event, on average, there are 13.26 unique clusters of information using the Spearman kernel over the mean occupancies as the measure. As depicted in Table 6.2, the inter-sensor mean occupancy correlation shows a Vendi score of 4.92 and 13.26, which means that the effective number of unique contributions or clusters of similar behavior is around 4.92 when considering the mean occupancy rates using the Pearson correlation, and 13.26 when considering the mean occupancy rates using the Spearman correlation. Spearman shows higher diversity (more unique sensors in an event) than Pearson for the same data, possibly indicating that Spearman captures more non-linear relationships between the sensors. Thus, considering monotonic relationships, less sensors are similar. Although YonedaVAE shows a higher diversity (dissimilar sensors in an event), it also exhibits the same behavior.

Table 6.2: VS over the inter-sensor mean occupancy correlation for the real test data and YonedaVAE

	Test Data	YonedaVAE ^{le}	YonedaVAE ^{ce}
VS (Pearson kernel)	4.92	5.65	5.94
VS (Spearman kernel)	13.26	13.91	11.89

As a result, it is shown that YonedaVAE is able to generate samples that are diverse and of higher quality than all other SOTA models for the high-granularity PXD background generation task.

Table 6.3: FID, KID, and VS score comparison between SOTA models for PXD generation averaged across four random seeds. They are computed over 5000 generated events. The lower the FID and KID, the better the sample quality and diversity. For the VS score, higher values indicate better sample diversity.

	TSPN (i.i.d) ¹	TSPN (Top-k) ²	IEA-GAN ³	Set-VAE ⁴	YonedaVAE	Test Data
FID	49.46 ± 0.29	41.40 ± 0.48	37.84 ± 0.98	33.49 ± 0.11	20.19 ± 0.31	0
KID ($\times 10^{-4}$)	339 ± 7	312 ± 1	283 ± 8	181 ± 2	130 ± 2	0

6.5.3 Topological and Geometrical Clustering for PXD data

This section wants to answer the following questions:

How are the points of a PXD point cloud clustered? How many complex clusters do they create? How many “holes” does a PXD point cloud have? How many pieces is it constructed out of? As one increases an observation threshold window, at what scale in the PXD point cloud do we observe changes in some geometric representation of the data?

6.5.3.1 Topological Data Analysis

Topological Data Analysis (TDA) [376] is a way to explore these questions and the shape of data without concern for things like which metric to use. The knowledge of topological features (such as connected components, loops, and higher dimensional cycles) that are present in data sets provides a better understanding of their structural properties at multiple scales. It can be leveraged to improve statistical inference as well. TDA is the branch of data science that aims to detect and encode such topological features. TDA has the following pipeline:

1. The input is assumed to be a finite set of points (like PXD hits as a point cloud), coming with a notion of distance - or similarity - between them. This distance can be induced by the metric in the ambient space (e.g., the Euclidean metric when the data are embedded in \mathbb{R}^d) or come as an intrinsic metric defined by a pairwise distance matrix.
2. A continuous shape is built on top of the data in order to highlight the underlying topology or geometry. This is often a simplicial complex or a nested family of simplicial complexes, called filtration, that reflects the structure of the data at different scales. Simplicial complexes can be seen as higher dimensional generalizations of neighboring graphs that are classically built on top of data in many standard data analyses or learning algorithms.
3. Topological or geometric information is extracted from the structures built on top of the data. This may either result in a full reconstruction, typically a triangulation, of the shape underlying the data from which topological/geometric features can be

easily extracted or in crude summaries or approximations from which the extraction of relevant information requires specific methods, such as persistent homology.

4. The extracted topological and geometric information provides new families of features and descriptors of the data. They can be used to understand the data better.

6.5.3.2 Persistent Homology and filtration

Persistent homology is a powerful tool to compute, study, and encode efficiently multiscale topological features of simplicial complexes and topological spaces. A finite set of data points (point cloud) can be viewed as a (noisy) sampling from an underlying topological space. The most obvious way to convert these points in a metric space into a global object is to think of the points as the vertices of a combinatorial graph whose edges are determined by proximity (vertices within some specified distance ϵ). Such a graph, while capturing connectivity data, ignores a wealth of higher-order features beyond clustering. These features can be accurately discerned by thinking of the graph as a scaffold for a higher-dimensional object. Specifically, one completes the graph to a simplicial complex — a space built from simple pieces (simplices) identified combinatorially along faces. The choice of how to fill in the higher dimensional simplices of the proximity graph allows for different global representations.

Simplicial complexes can be seen as higher dimensional generalizations of graphs. They are mathematical objects that are both topological and combinatorial, a property that makes them particularly useful for TDA. Given a set $X = \{x_0, \dots, x_k\}$ which is a subset of \mathbb{R}^d and consists of $k + 1$ affinely independent points, I define a k -dimensional simplex σ as $[x_0, \dots, x_k]$, which is essentially the convex hull of X . The points in X are referred to as the vertices of σ , and the simplices formed by the subsets of X are termed the faces of σ . For instance, a 0-simplex is a vertex, a 1-simplex is an edge, a 2-simplex is a triangle, a 3-simplex is a tetrahedron, and so on; see Figure 6.25.

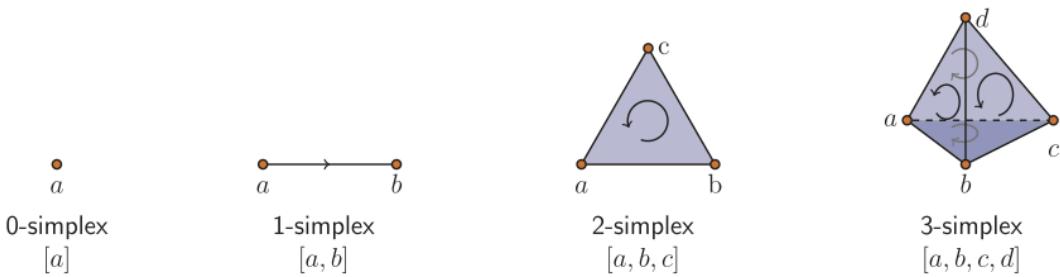


Figure 6.25: **Oriented k -simplices for $k = 0, 1, 2, 3$.** These k -simplices are the building blocks used to construct a simplicial complex from a point cloud of data.

Constructing simplicial complexes can vary depending on the underlying data set. Multiple techniques are available for creating these structures depending on the inductive

biases in the data—for instance, the Vietoris-Rips complex, Alpha Complexes, Čech Complexes, and Delaunay Triangulations.

Definition 2 (Čech Complex). Let $\{x_\alpha\}$ be a collection of points in Euclidean space \mathbb{R}^d . The Čech complex, \mathcal{C}_ϵ , is defined as an abstract simplicial complex. Its k -simplices are formed by unordered $(k + 1)$ -tuples of points $\{x_{\alpha_0}, \dots, x_{\alpha_k}\}$ such that their closed $\frac{\epsilon}{2}$ -ball neighborhoods have a point of common intersection.

Definition 3 (Rips Complex). Let $\{x_\alpha\}$ be a collection of points in Euclidean space \mathbb{R}^d . The Rips complex, \mathcal{R}_ϵ , is an abstract simplicial complex. Its k -simplices are determined by unordered $(k + 1)$ -tuples of points $\{x_{\alpha_0}, \dots, x_{\alpha_k}\}$ such that each pair of points in the tuple are within distance ϵ .

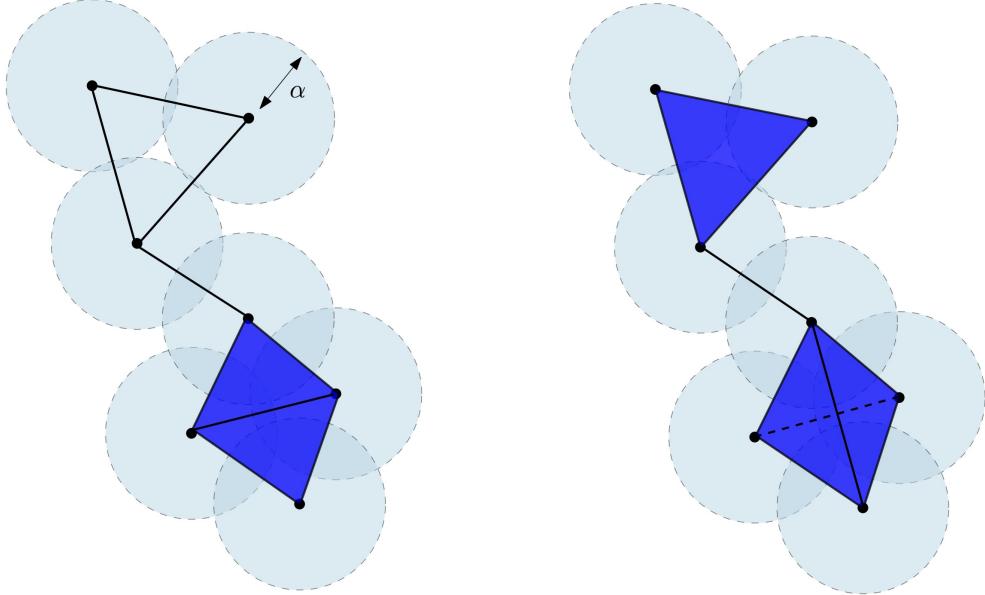


Figure 6.26: The Čech complex (left) and the Vietoris-Rips (right) of a finite point cloud in the plane \mathbb{R}^2 . The bottom part of the Čech complex is the union of two adjacent triangles, while the bottom part of the Rips complex is the tetrahedron spanned by the four vertices and all its faces. The dimension of the Čech complex is 2. The dimension of the Vietoris-Rips complex is 3.

Transforming a point cloud into an abstract simplicial complex, such as a Čech or Rips complex, for measuring the homology of data mandates the selection of a proximity scale parameter ϵ , the scale over which the simplicial connections are made. For small values of ϵ , the generated complex is essentially a disconnected set of points. Conversely, for large ϵ , it converges to a single, high-dimensional simplex.

For example, to form k -simplices using the Vietoris-Rips complex, one first defines a distance metric, which can be realized as a symmetric $N \times N$ matrix of pairwise distances between points. For each $\epsilon > 0$, I construct a Vietoris-Rips simplicial complex \mathcal{R}_ϵ in the

following way. In \mathcal{R}_ϵ , every collection of $k + 1$ data points is a k -simplex if the pairwise distance between points is less than ϵ . Thus, the 0-simplices are the data points themselves. A 1-simplex (an edge) is formed whenever two points are within ϵ of one another. A 2-simplex (a triangle) is formed whenever three points are pairwise within ϵ of one another; this occurs when there is a 3-cycle in the underlying graph formed by the vertices and edges in $S\mathcal{R}_\epsilon$. A 3-simplex (a tetrahedron) is formed whenever four points are pairwise within ϵ of one another. This raises a question: *Is there an optimal ϵ that accurately reflects the topology of the underlying data?*

Algebraic topology offers a mature set of tools for counting and collating holes and other topological features in spaces and maps between them. In the context of high-dimensional data, algebraic topology works like a telescope, revealing objects and features not visible to the naked eye. In order to handle topological measures, particularly numerical ones, that can be easily computed from the complex, one can consider the homology of the considered simplicial complexes. Homology is a classical concept in algebraic topology, providing a powerful tool to formalize and handle the notion of topological features of a simplicial complex in an algebraic way.

For any dimension k , the k -dimensional hole is represented by a vector space H_k whose dimension is intuitively the number of such independent features. For example, the 0-dimensional homology group H_0 represents the group of connected components of the complex, the 1-dimensional homology group H_1 represents the group of 1-dimensional loops, and the 2-dimensional homology group H_2 represents the group of 2-dimensional cavities. In terms of the topological measures (invariants), the k 'th *betti number*, b_k can be defined as the dimension of these groups. b_k equals the number of independent holes of dimension k . For instance, b_0 is the number of connected components, b_1 is the number of topological circles, b_2 is the number of trapped volumes, and so on. The topology of a simplicial complex may be described by the sequence of Betti numbers, $\mathbf{b} = (b_0, b_1, b_2, \dots)$. For instance, a topological circle has $\mathbf{b} = (1, 1, 0, \dots)$, a topological torus has $\mathbf{b} = (1, 2, 1, 0, \dots)$, and a topological sphere has $\mathbf{b} = (1, 0, 1, 0, \dots)$. Betti numbers are homological invariants, meaning that homologically equivalent spaces have the same Betti number.

Despite being both computable and insightful, the homology of a complex associated with a point cloud at a particular ϵ is insufficient: it is a mistake to ask which value of ϵ is optimal. Nor does it suffice to know a simple “count” of the number and types of holes appearing at each parameter value ϵ . Figure 6.27 presents an example. Homological invariants are not enough. One must declare which holes are essential and which can be safely ignored. The standard topological constructs of homology and homotopy offer no such slack in their strident rigidity: a hole is a hole, no matter how fragile or fine.

To address the above issue, the concept of persistence was initially formalized. In this framework, topological features that remain stable across a wide range of parameter values are considered as the “signal,” while features that exist for a short parameter range are treated as “noise.” One exploits the fact that as ϵ grows, so do the Rips complexes, giving an inclusion of complexes for small ϵ into those for larger values. To illustrate, let $\mathbf{R} = (R_i)_{i=1}^N$ be a sequence of Rips complexes generated from a fixed point cloud, using an increasing series of scale parameters $(\epsilon_i)_{i=1}^N$. The relationship between these complexes can

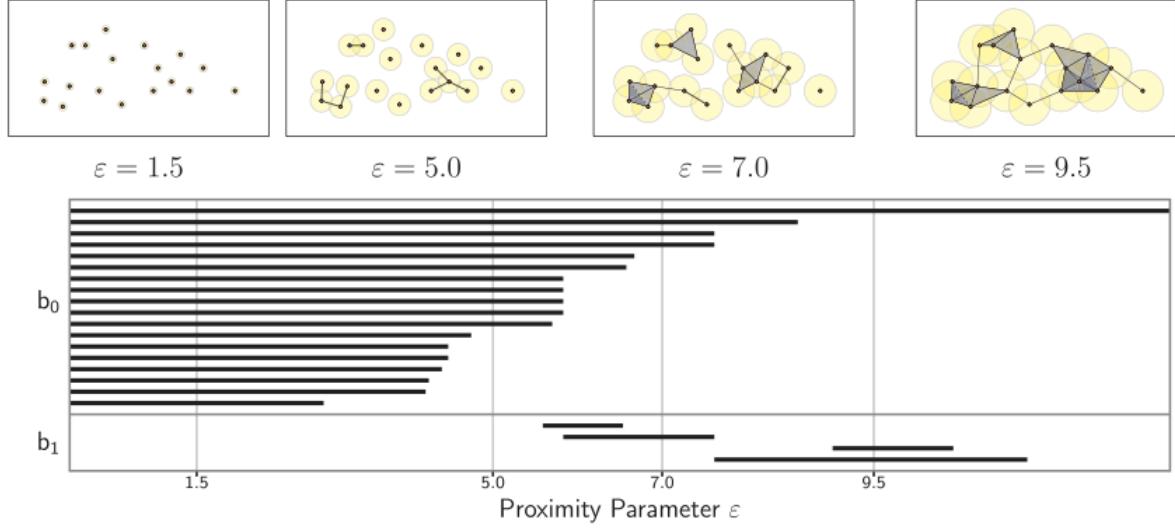


Figure 6.27: Example of the topological barcode of a Vietoris-Rips complex, adopted from [377]. The top four figures display the simplicial complex of 18 points for different values of the proximity parameter ϵ . The vertical lines in the barcode correspond to these four levels of ϵ . The number of horizontal bars intersecting each line gives the values of $\mathbf{b}(\epsilon) = (b_0(\epsilon), b_1(\epsilon))$. For the parameters selected, $\mathbf{b}(1.5) = (18, 0)$, $\mathbf{b}(5.0) = (11, 0)$, $\mathbf{b}(7.0) = (4, 1)$, and $\mathbf{b}(9.5) = (1, 2)$.

be understood through a sequence of natural inclusion maps,

$$R_1 \xrightarrow{\iota} R_2 \xrightarrow{\iota} \cdots \xrightarrow{\iota} R_{N-1} \xrightarrow{\iota} R_N. \quad (6.23)$$

This sequence is called a *filtration*. Persistent homology, then, tracks topological features which *persist* across a range of values of ϵ . Those features that persist over a large range are considered signals of the underlying topology, while the short-lived features are taken to be noise inherent in approximating a topological space with a finite sample. Thus, rather than focusing solely on the homology groups H^*R_i of each individual Rips complex R_i , one investigates the homomorphisms induced by the inclusion maps $\iota : H^*R_i \rightarrow H^*R_j$ where $i < j$. These induced maps disclose the persistent features within the sequence of complexes. The induced homomorphism $\iota^* : H^*R_\epsilon \rightarrow H^*R_{\epsilon'}$ unveils details not immediately discernible when examining H^*R_ϵ and $H^*R_{\epsilon'}$ in isolation. This can be seen as a precursor to the more generalized notion of persistent homology within an arbitrary sequence of chain complexes.

Persistent homology provides efficient algorithms to compute the homological invariants of each complex in the considered families and encodes the evolution of the homology groups of the nested complexes across the scales. If homology is to measure the shape of abstract spaces, then persistent homology is how the shape of a geometric data set can be quantified. Consequently, the topological features of a complex can be inferred through the persistent homology of the Vietoris-Rips filtration through which new connected components may

emerge, existing ones may merge, and higher-order topological features like loops and voids may form and vanish. Persistent homology tracks these evolving features by associating each a “lifetime.” The output is typically a “barcode,” a collection of intervals, or a point cloud in \mathbb{R}^2 , each point of which represents the birth and death times of a topological feature. Each homology space has a distinct barcode H_k from which I infer the Betti number b_k . As an example, see Figure 6.27. The horizontal axis corresponds to the proximity parameter ϵ , and the vertical axis is an (arbitrary) ordering of the homology generators, *i.e.*, the distinct homology classes of dimension k . Each homology class is visualized by a bar that persists for a given range of ϵ . Its leftmost endpoint is at the ϵ value at which the homology class forms and its rightmost endpoint is the ϵ value at which it disappears. At any given ϵ , the Betti number $b_k(\epsilon)$ is the number of bars that intersect the vertical line through ϵ . Those bars which persist over longer intervals generally correspond to real topological features, whereas short bars are considered noise.

In general, the culmination of tracking the persistent features across multiple scales is referred to as a *Persistence Diagram*. A persistence diagram is a multiset in $\mathbb{R} \times \mathbb{R} \cup \{\infty\}$, with each point (b, d) signifying a topological feature that is born at the scale b and dies entering the scale d . Formally, the i -th persistence diagram D_i is constituted by points (b_j, d_j) where the homological class α_j is born at $H^*(C_{b_j}^*)$ and dies entering $H^*(C_{d_j}^*)$. Intuitively, a persistence diagram is a way to summarize the “lifespan” of topological features as one ‘filters’ through a point cloud. In a persistence diagram, points near the diagonal are inferred to live short, while points further from the diagonal are considered topologically persist longer.

Let’s dive into an example in which I consider the filtration given by a union of growing balls centered on the finite set of points C , depicted in Figure Figure 6.28. This filtration is homotopy equivalent to the Čech filtration built on top of C . It shows several level sets of the filtration:

1. For the radius $\epsilon = 0$, the union of balls is reduced to the initial finite set of points, each of them corresponding to a 0-dimensional feature, *i.e.*, a connected component; an interval is created for the birth for each of these features at $\epsilon = 0$.
2. Some of the balls started to overlap, resulting in the death of some connected components that get merged together; the persistence diagram keeps track of these deaths, putting an endpoint to the corresponding intervals as they disappear.
3. New components have merged, giving rise to a single connected component, and so, all the intervals associated with a 0-dimensional feature have been ended, except the one corresponding to the remaining components; two new 1-dimensional features have appeared resulting in two new intervals (in blue) starting at their birth scale.
4. One of the two 1-dimensional cycles has been filled, resulting in its death in the filtration and the end of the corresponding blue interval.
5. All the 1-dimensional features have died. It only remains the long (and never dying) red interval. The final barcode can also be equivalently represented as a persistence

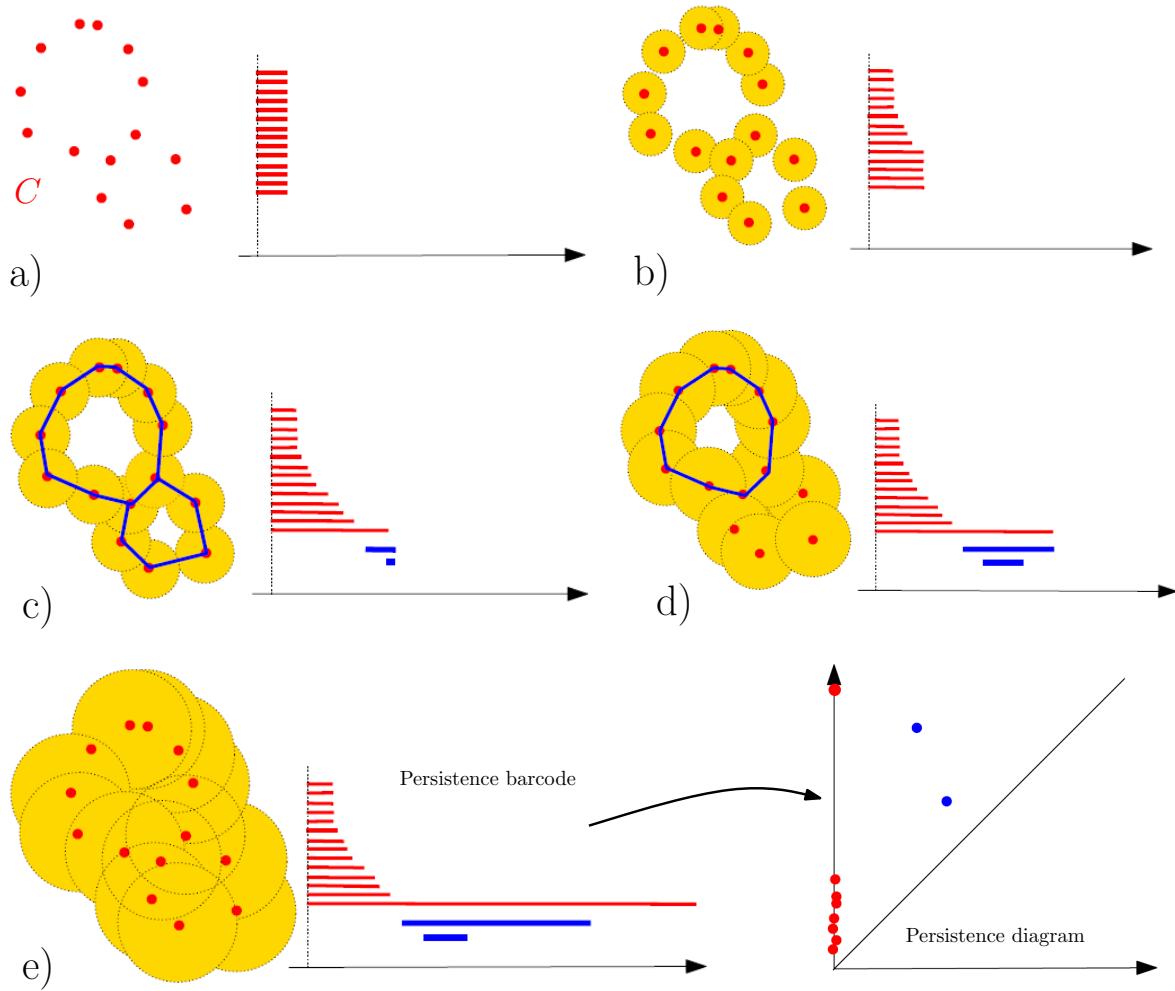


Figure 6.28: The set filtration of the distance function to a point cloud and the construction of its persistence barcode as the radius of balls increases, adopted from [376]. The blue curves in the unions of balls represent 1-cycles associated with the blue bars in the barcodes.

diagram where every interval (a,b) is represented by the point of coordinate (a,b) in \mathbb{R}^2 . Intuitively, the longer an interval in the barcode or, equivalently, the farther from the diagonal is the corresponding point in the diagram, the more persistent, and thus relevant, is the corresponding homological feature across the filtration. Notice also that for a given radius ϵ , the k 'th Betti number of the corresponding union of balls is equal to the number of persistence intervals corresponding to k -dimensional homological features and containing ϵ . So, the persistence diagram can be seen as a multiscale topological signature encoding the homology of the union of balls for all radii as well as its evolution across the values of ϵ .

After this brief introduction to TDA, let's find out how these methods can be applied to

the PXD background data to analyze their underlying clusters.

6.5.3.3 Topological Data Analysis of PXD Background

The clusters and distribution of PXD hits point clouds can also be studied through the lens of TDA. Traditional clustering algorithms may lack the capability to capture intricate spatial and topological features. Persistent homology, particularly in combination with Rips filtrations, offers a multi-scale, topologically rigorous approach to understanding this information. Clusters of PXD hits can be intuitively thought of as connected components in a topological space. Each hit will be its own connected component at low values of ϵ . As ϵ grows, these components will merge, capturing the notion of hits that are “close” to each other. The higher-dimensional features, such as loops and voids, could provide crucial insights into the organization and relations between hit clusters coming from background processes. For example, a line cluster of hits can be represented through a short-lived H_0 feature in a PXD point cloud as the filtration balls merge instantly for very close points, or a loop may indicate a spiral-like arrangement of hits, while a void could indicate an area devoid of hits surrounded by a high concentration of hits. The persistent diagram will indicate the scale at which these clusters form and disappear, providing a multiscale summary of clustering.

Let’s dive into some controlled examples where we can observe the association between possible PXD artifacts and persistent homology, as depicted in Figure 6.29. Figure (a) represents a dense linear set of points that exemplifies a long continuous hit cluster and its corresponding persistent diagram. Due to the closeness of dense points in a linear cluster, the persistent diagram shows a very low lifetime of 0.0055 for this line artifact. By adding two separate points, figure (b) illustrates how having a point-like (alone) set of points creates a long-lived persistent homology. In the persistent diagram, the upper points at death coordinate 0.47 and 0.18 belong to the two separated added point and their merge with the linear cluster, while the lower point at 0.0055 still belongs to the linear cluster. Figure (c) shows a curved cluster of points along with two separated points. Short-lived points in the persistent diagram again belong to the cluster. I also have a persistent H_1 feature that is due to the half-circular cluster revealing a one-dimensional or circular hole. Short-lived hits in the persistent diagram of Figure (d) again belong to the clusters in the dense circle-like points.

The most obvious conclusion from the examples in Figure 6.29, is how dense and line-like clusters create H_0 persistent features and how curvy-like clusters reveal the existence of H_1 persistent features. Persistence diagrams have been shown to be a powerful tool for quantifying shapes in geometric data. Moreover, one of their key properties is their stability with respect to perturbations in the input, which is crucial when dealing with noisy measurements. This study compares the real PXD hits with YonedaVAE’s output and random set of points. In Figure 6.30, as an example, compares the persistence diagrams of real PXD background hits with YonedaVAE’s output and random set of points. As expected, the first noticeable difference is the lifetime of H_0 features. As opposed to random points where no meaningful geometric clusters exist, real PXD hits and YonedaVAE exhibit

shorter-lifetime H_0 features that are caused by line-like clusters.

In Figure 6.31, the distribution of H_0 and H_1 lifetimes over 5000 samples are depicted for the three cases, real PXD background, YonedaVAE, and random set of points. The noticeable difference again is at the low lifetime H_0 feature that dies for clusters for both PXD and YonedaVAE samples. This signifies how well YonedaVAE captures the cluster of hits in the PXD data. On the other hand, both PXD and YonedaVAE exhibit more persistent H_1 features. In the persistent homology, a loop represents a 1-cycle that is not the boundary of a 2-chain in a simplicial complex and corresponds to a non-zero element in the first homology group H_1 . A persistent H_1 feature in the persistent diagram signifies a topological loop structure in the data that is robust across a range of scales. The emergence of a loop in the persistent diagram can be indicative of a spiral-like or curvy-like arrangement of hits. This could correspond to particles following a helical trajectory.

In order to numerically compare persistence diagrams of the real PXD data and from the YonedaVAE, one can incorporate different metrics such as Wasserstein distance, L_p distance between betti curves (the Betti numbers) as amplitudes, and the entropy of the persistence diagrams [378, 379]. Persistence entropy is a metric for characterizing the complexity of the persistence diagrams. It quantifies the uncertainty associated with the lifespan of topological features, thereby offering insights into the overall structural complexity of the data. The amplitude of a persistence diagram measures the “spread” or “complexity” of the points in the diagram. In other words, it quantifies how far the points in the diagram are from the diagonal, which in turn provides an indication of the prominence or importance of the corresponding topological features. For each persistence diagram in a collection, an amplitude is measured according to the following steps. First, the diagram is partitioned into subdiagrams according to the homology dimension. Then, the amplitude of each subdiagram is calculated according to a metric that gives a vector of amplitudes over the available homology dimensions. I report the average of these amplitudes as summary statistics across all tested samples.

Table 6.4: Quantitative comparison between persistence diagrams using amplitudes and entropy

	Real PXD	YonedaVAE	Random
Persistence Entropy H_0	5.54	5.64	7.54
Persistence Entropy H_1	2.34	2.16	5.78
Wasserstein Amplitude H_0	0.45	0.39	0.88
Wasserstein Amplitude H_1	0.048	0.048	0.201
betti Amplitude H_0	16.45	19.17	62.36
betti Amplitude H_1	0.56	0.49	7.52

As depicted in Table 6.4, across all metrics and both homology dimensions, the generated data by YonedaVAE closely mirrors the real PXD data. This indicates the model’s effectiveness in capturing both continuous clustering features (H_0) and higher-order clustering

structures (H_1). As expected, the random data set deviates significantly across all metrics, reinforcing the non-random nature of the real and generated PXD background data sets. Moreover, the persistence of H_1 features in the PXD data could be indicative of a specific type of interaction or decay process, which is an open question that can be studied in detail.

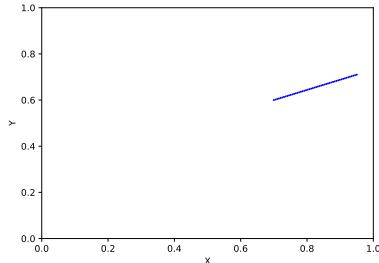
6.5.3.4 Geometrical Clustering

In order to consolidate the clustering analysis, the PXD clustering algorithm, *PXDClusterizer* module from the basf2 [8] is incorporated. As discussed in Chapter 2, the PXDClusterizer module is responsible for cluster formation and hit reconstruction. During its operation, the algorithm iteratively scans through all the digitized signals, or “digits,” from individual sensors. It then coalesces contiguous digits, centered around a seed pixel with a signal exceeding a predefined threshold, into consolidated cluster formations. After forming these clusters, they are then transformed into what are termed PXDHits. The transition to PXDHits involves deducing the hit location based on the cluster’s charge distribution profile. Specifically, the center of gravity algorithm is employed for clusters comprising two pixels, while the analog head-to-tail algorithm is utilized for larger clusters. In scenarios where only a single digit in one dimension contributes to the cluster, the hit is assigned to the pixel’s center. Subsequently, various attributes of the hit—such as its final position in the sensor, the cumulative charge, and its correlation to the originating cluster and particles—are recorded.

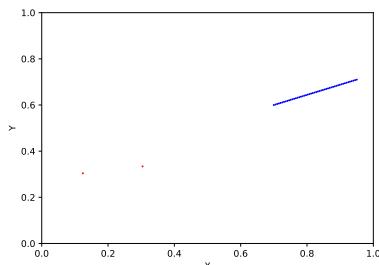
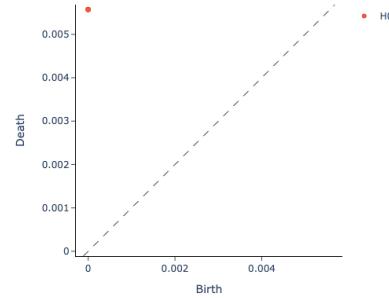
Clusters serve as a proxy for the interaction of a single particle with the PXD. Their properties are subject to this analysis as depicted in Figure 6.32. These attributes can include the total number of pixels constituting the cluster (Cluster size), the number of clusters for each sensor, the position along the u- and v-axes, the sum of charges across the cluster (cluster charge), and the charge at the seed pixel (seed charge). As depicted in Figure 6.32, YonedaVAE shows a close agreement with the real PXD clustering attributes. On the other hand, although point cloud-based approaches can handle sparse data, they exhibit a drawback when utilized for PXD background generation that must be studied further to have a conclusive result. This drawback can be conceptualized as a trade-off incurred when transitioning from a discrete (pixelated PXD bitmap) representation to a continuous (point cloud) representation, which results in information loss. To leverage point cloud approaches, data must transition from a pixelated format, where the positions of hits are defined by their exact pixel locations in \mathbb{Z} (e.g. [[1-768]]), to a continuous format, where the positions of hits are in \mathbb{R} (e.g. [[0,1]]) that possess a precision uncertainty. This issue is particularly more significant in the PXD case due to its ultra-high-dimensional size; the error introduced when reverting back to a pixelated format from the continuous point cloud format is amplified by the number of pixels in the PXD.

Encountering challenges in accurately reconstructing large cluster patterns, let’s explore in the following section how YonedaVAE performs in terms of downstream track reconstruction and its efficacy in generating hits that align with the tracks emerging from these clusters.

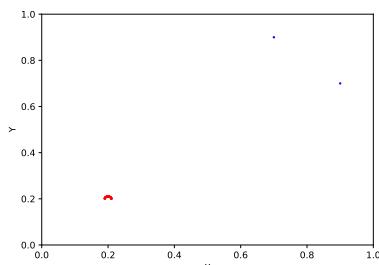
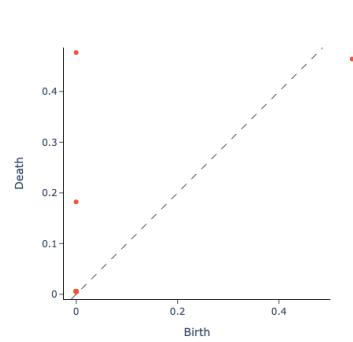
Figure 6.29: Persistent diagrams for each example of possible dense PXD hits



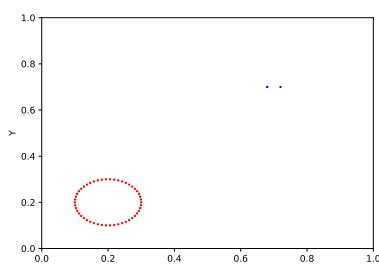
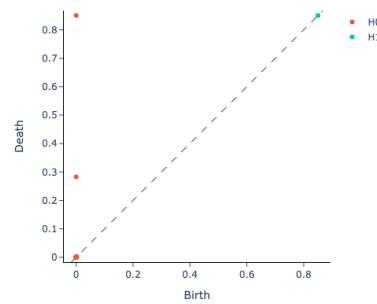
(a) Having a dense linear artifact



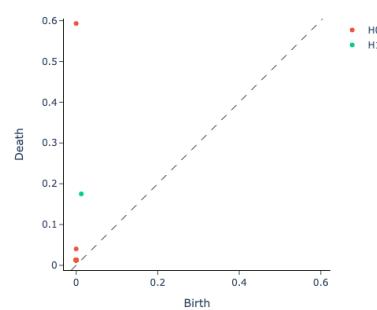
(b) Having a dense linear artifact and two separate points



(c) Having a dense curved-like artifact and two separate points



(d) Having a dense circle-like artifact and two separate points



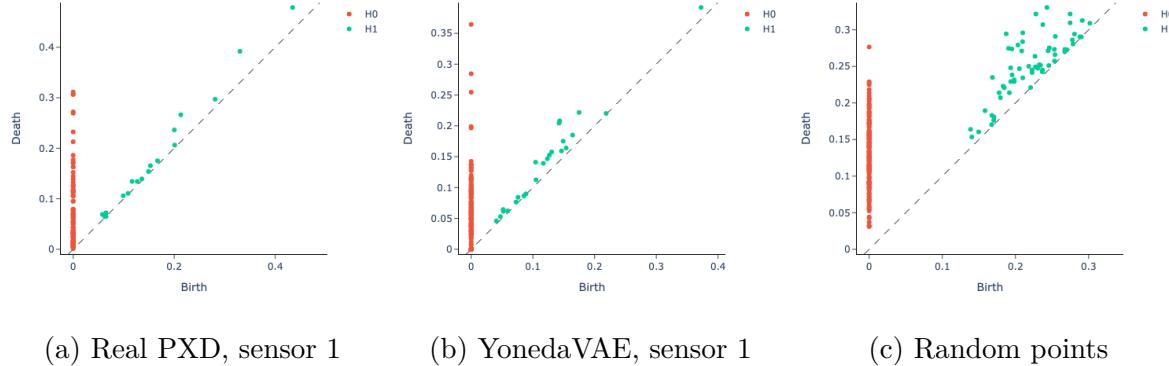


Figure 6.30: An example persistent diagram for a point cloud belonging to sensor 1 with 255 set cardinality. Notice the short-lived H_0 features from PXD and YonedaVAE.

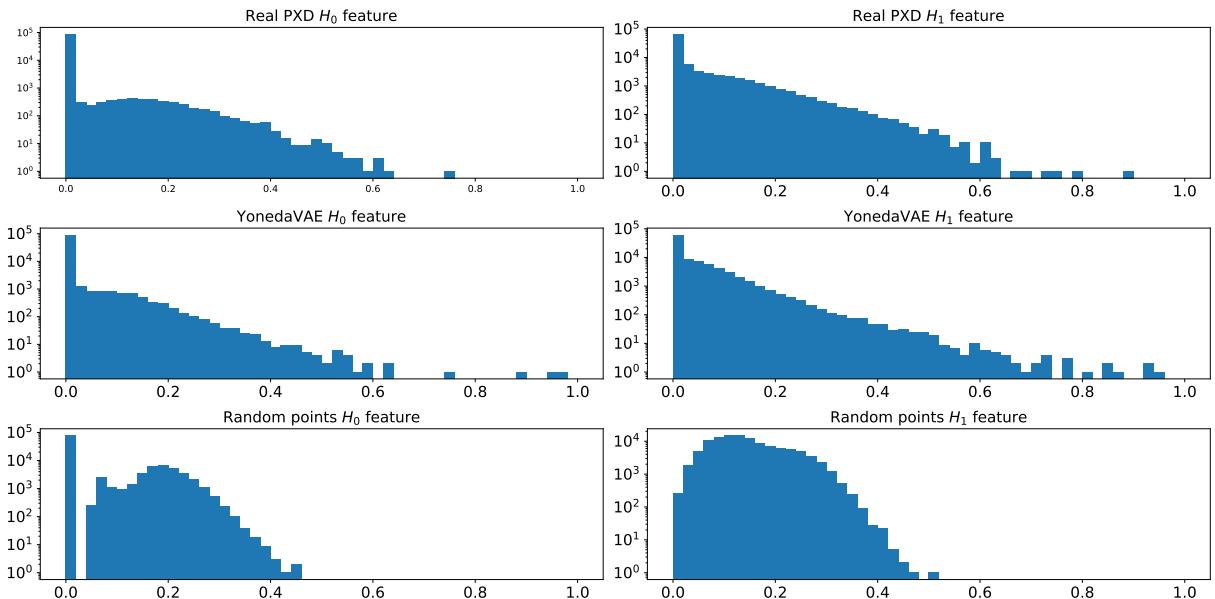


Figure 6.31: Histograms for H_0 and H_1 feature lifetimes for the real PXD hits, YonedaVAE generated samples, and samples with random hits.

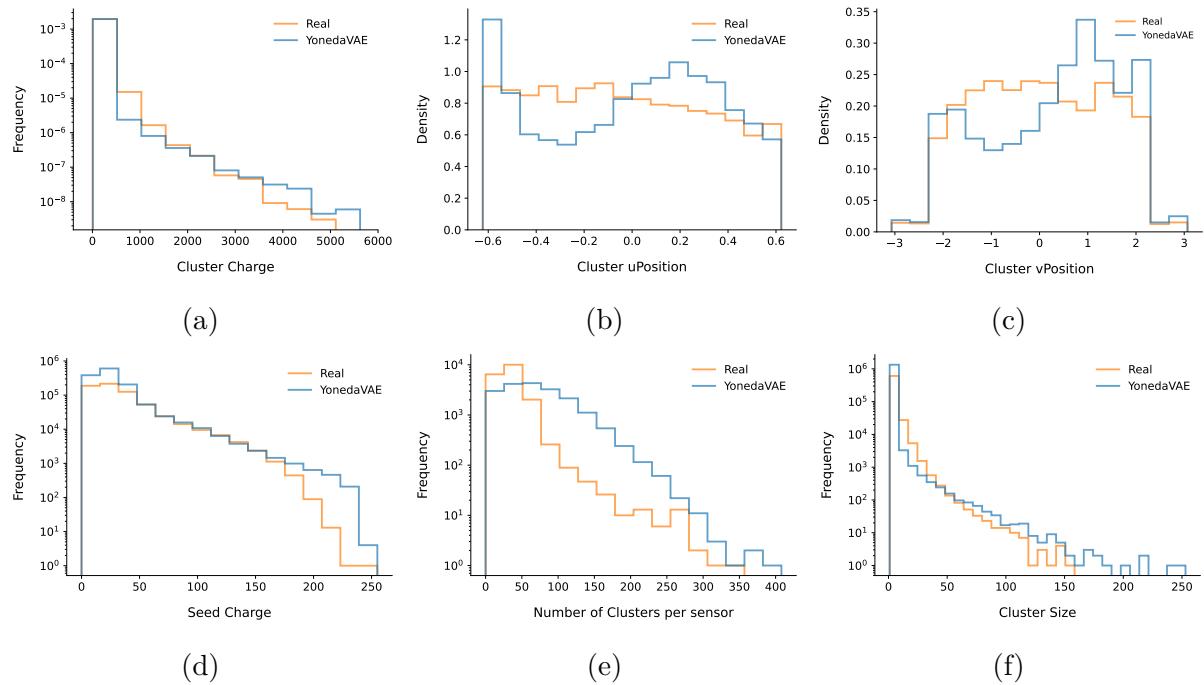


Figure 6.32: Comparison between the real PXD clustering information and the YonedaVAE generated ones.

6.5.4 Physics Analysis

This section dives into the tracking analysis by making a comparison between real PXD background and YonedaVAE-generated PXD background hits on the track parameter resolutions as introduced in Chapter 5. This study compares both the length extrapolative and context extrapolative YonedaVAE with the real PXD background of Experiment 26.

First, let's study the effect of intra-event correlation on the track reconstruction, specifically the Helix parameters. In the last chapter, I studied this for the Geant4 simulated data and showed how the lack of intra-event correlation negatively influences the Helix parameter resolution. Table 6.5, and Figure 6.33 shows the effect of losing the intra-event correlation for high momentum tracks, $p_t > 0.4$ GeV, is demonstrated through the huge discrepancy between the correlated (unshuffled) and uncorrelated (shuffled) events in the unbiased variance of the Δd_0 and Δz_0 , thus in the precision of the d_0 and z_0 reconstruction.

Table 6.5: Helix Parameter Resolution observable Comparison between correlated (Unshuffled) and uncorrelated (Shuffled) events with high momentum tracks, $p_t > 0.4$ GeV.

Parameter	Observable	Signal + Background		
		Shuffled	Unshuffled	No Bkg.
Δd_0	Unbiased Variance	0.1423 ± 0.0007	0.0770 ± 0.0004	0.0730 ± 0.0004
	KS Statistic	0.0074, p-value: 0.6638		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	22	27	18
	Num. Tracks $> 5\sigma_{\text{sig.}}$	19	24	17
$\Delta \phi_0$	Unbiased Variance	0.1650 ± 0.0008	0.1864 ± 0.0010	0.1706 ± 0.0009
	KS Statistic	0.0115, p-value: 0.1610		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	19	18	12
	Num. Tracks $> 5\sigma_{\text{sig.}}$	20	29	23
Δz_0	Unbiased Variance	4.6455 ± 0.0238	5.4540 ± 0.0279	4.6674 ± 0.0239
	KS Statistic	0.0189, p-value: 0.0022		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	2	1	0
	Num. Tracks $> 5\sigma_{\text{sig.}}$	32	37	26
$\Delta \omega$	Unbiased Variance	0.0008 ± 0.0001	0.0008 ± 0.0001	0.0006 ± 0.0001
	KS Statistic	0.0095, p-value: 0.3473		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	23	25	20
	Num. Tracks $> 5\sigma_{\text{sig.}}$	25	30	25
$\Delta \tan \lambda$	Unbiased Variance	0.0373 ± 0.0002	0.0399 ± 0.0002	0.0352 ± 0.0002
	KS Statistic	0.0237, p-value: 0.0000		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	38	48	36
	Num. Tracks $> 5\sigma_{\text{sig.}}$	7	7	7

First, let's take a look at the correlation between the Helix parameters, both the true and the reconstructed ones. A very interesting observation in Figure 6.34 is the lack of linear correlation (Pearson) but the monotonic correlation (Spearman) between z_0 and z_0^t . YonedaVAE also mimics the same behavior.

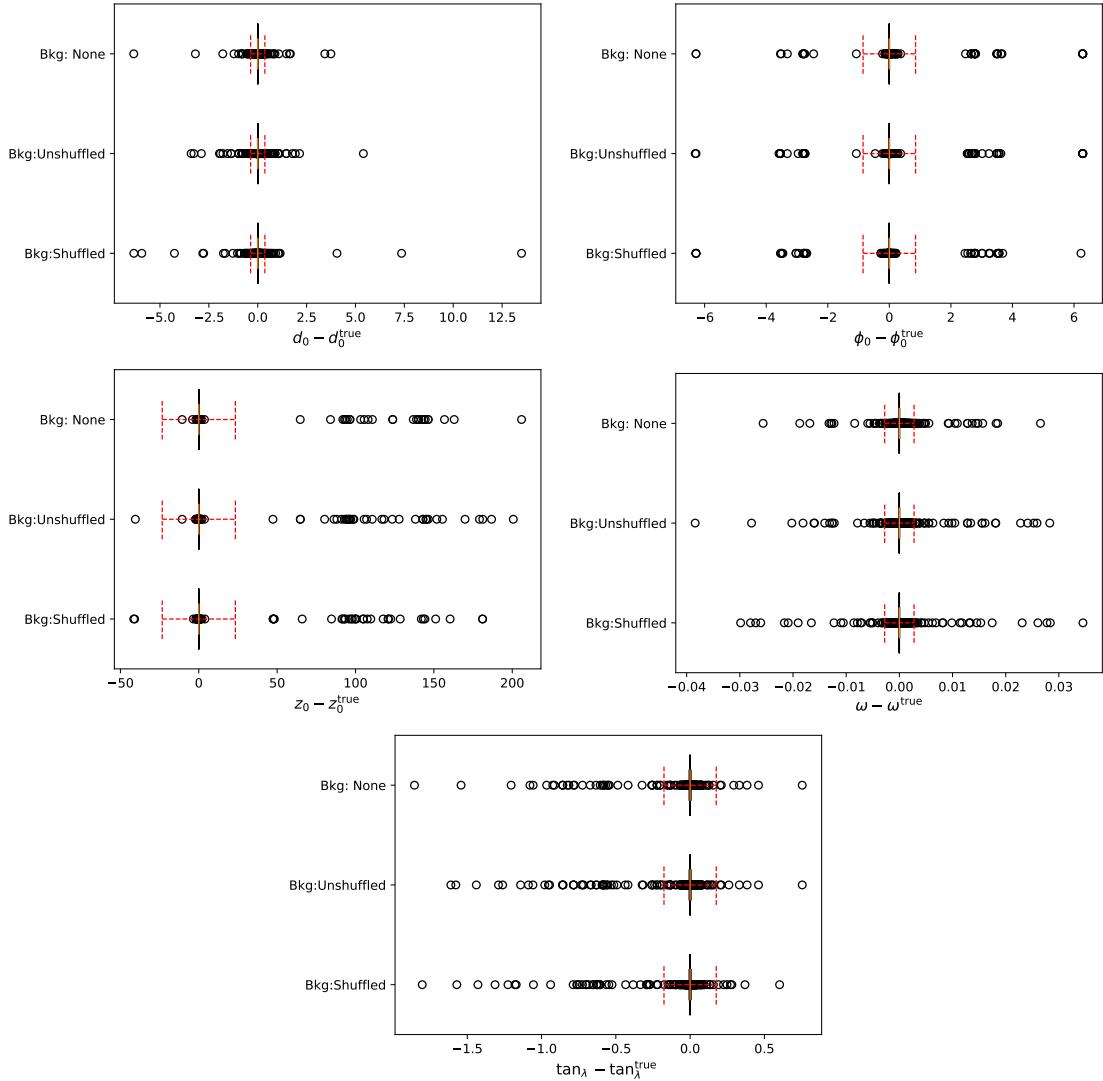


Figure 6.33: The boxplot for comparing the Helix parameter resolutions in the presence of correlated background and uncorrelated background at high momentum regime. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.

One can also compare the population of the number of PXD hits as depicted in Figure 6.35. 4 PXD hits show a slightly lower population as the background is overlaid. However, this could be coming from statistical fluctuations.

Dividing the hits to high momentum $p_t > 0.4$ GeV and low momentum $p_t < 0.4$ GeV regions, one can observe the effect of background in a more illuminating way in Figure 6.36 and Figure 6.39.

In the last chapter, analyzing the effect of the simulated background, the observation was that one of the effects of the background is the increased variance of the resolution.

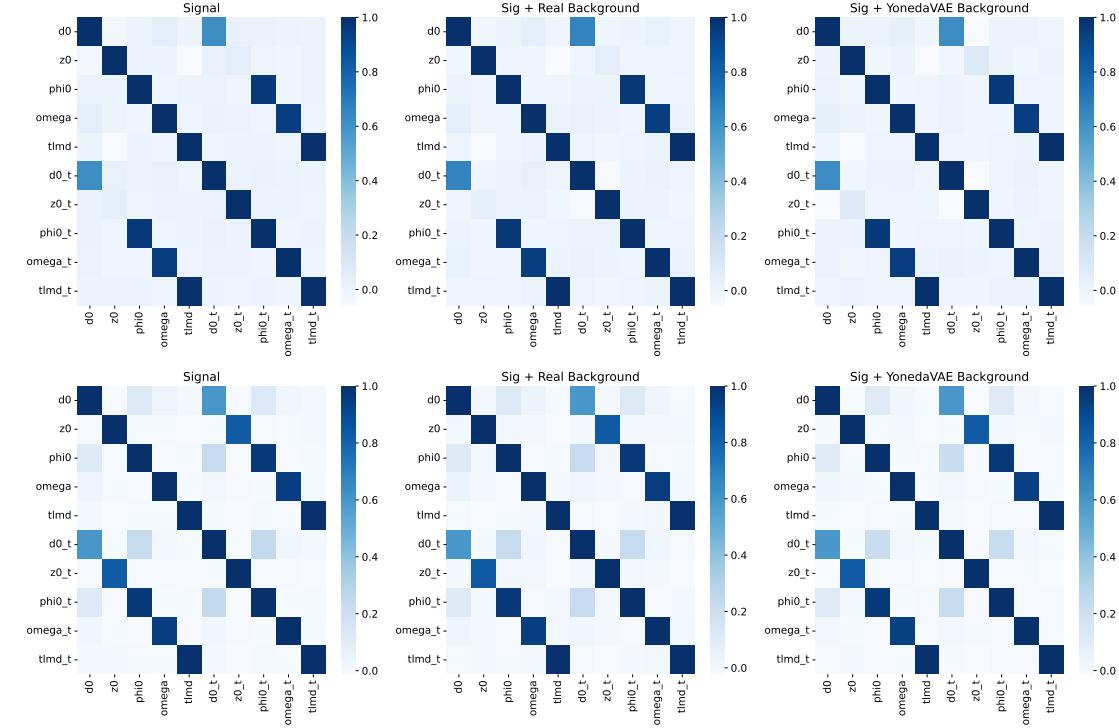


Figure 6.34: Pearson (top) vs Spearman (bottom) correlation between the true and reconstructed Helix parameters with and without background.

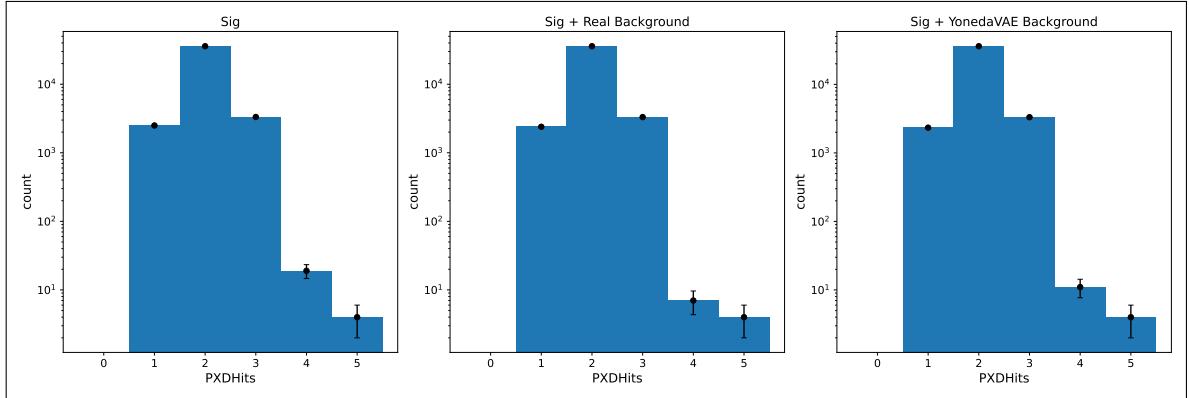


Figure 6.35: The population plot of each number of PXD hits. Notice the count is in the log scale.

With a real PXD background overlay, this effect still remains true. The performance of YonedaVAE_{le} and YonedaVAE_{ce} in approximating helix parameter resolutions is particularly noteworthy when evaluated against the overlay of random-trigger real background conditions as depicted in Figure 6.37 for YonedaVAE with length extrapolation and Figure 6.38 for YonedaVAE with context extrapolation. As shown in Table 6.6 and in Table 6.7,

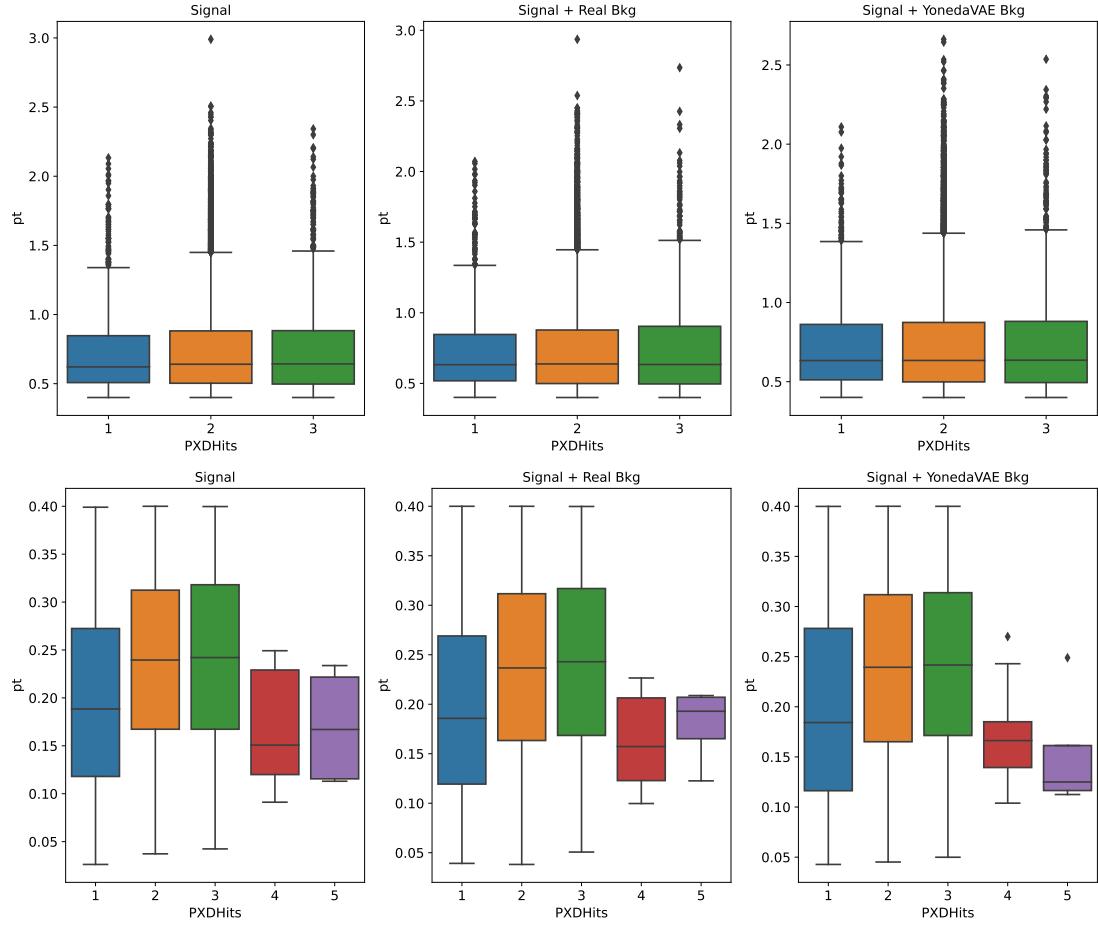


Figure 6.36: High momentum (top), and low momentum (bottom) momentum boxplot per number of PXD hits. The YonedaVAE here is with length extrapolation.

quantitatively, the unbiased variances for helix parameters, $\Delta d_0, \Delta \phi_0, \Delta z_0, \Delta \omega$, and $\Delta \tan \lambda$, are in close proximity to those obtained with real background data, often within the statistical uncertainties. Additionally, the Kolmogorov-Smirnov statistics yield p-values greater than 0.05, suggesting that the parameter distributions generated by YonedaVAE_{le} and YonedaVAE_{ce} are statistically indistinguishable from those generated under real background conditions. This congruence underscores YonedaVAE's robustness in mimicking the complex behaviors introduced by real background interference.

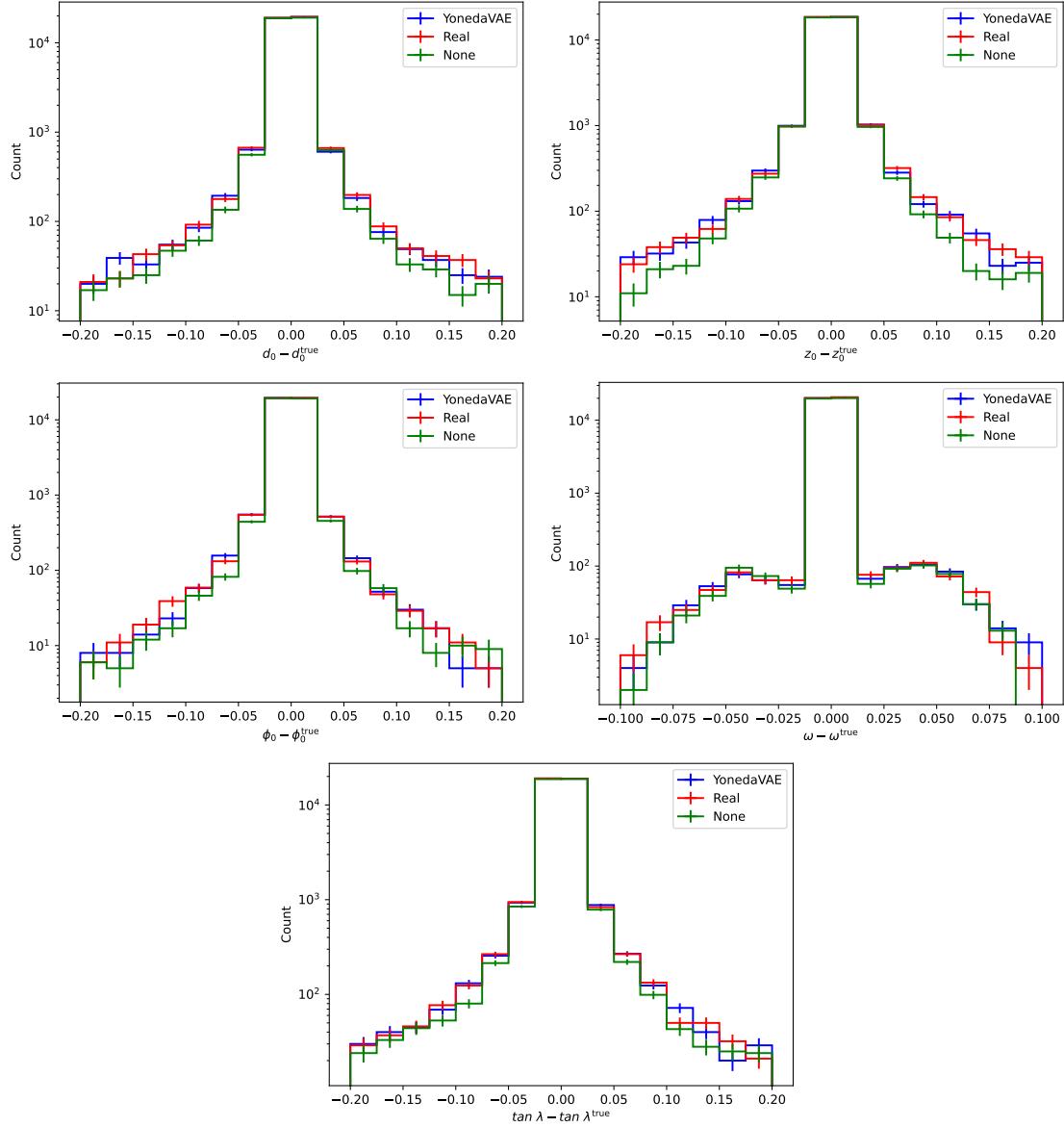


Figure 6.37: Helix parameter resolutions. For each parameter, blue represents YonedaVAE with length extrapolation, red represents Real PXD, and green represents resolution with no background overlay.

Table 6.6: Helix Parameter Resolution observable Comparison with the length extrapolative YonedaVAE

Parameter	Observable	Signal + Background		
		YonedaVAE _{le}	Real	No Bkg.
Δd_0	Unbiased Variance	0.1713 ± 0.0006	0.1643 ± 0.0006	0.1594 ± 0.0006
	KS Statistic	0.0058, p-value: 0.4910		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	138	146	106
	Num. Tracks $> 5\sigma_{\text{sig.}}$	148	136	112
$\Delta \phi_0$	Unbiased Variance	0.4444 ± 0.0015	0.4480 ± 0.0016	0.4429 ± 0.0016
	KS Statistic	0.0035, p-value: 0.9591		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	339	317	311
	Num. Tracks $> 5\sigma_{\text{sig.}}$	362	389	357
Δz_0	Unbiased Variance	5.9991 ± 0.0209	5.8735 ± 0.0204	5.7120 ± 0.0201
	KS Statistic	0.0040, p-value: 0.8993		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	64	71	62
	Num. Tracks $> 5\sigma_{\text{sig.}}$	273	261	238
$\Delta \omega$	Unbiased Variance	0.0066 ± 0.0001	0.0065 ± 0.0001	0.0061 ± 0.0001
	KS Statistic	0.0052, p-value: 0.6296		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	226	226	225
	Num. Tracks $> 5\sigma_{\text{sig.}}$	333	321	299
$\Delta \tan \lambda$	Unbiased Variance	0.0753 ± 0.0003	0.0726 ± 0.0003	0.0734 ± 0.0003
	KS Statistic	0.0044, p-value: 0.8238		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	248	250	226
	Num. Tracks $> 5\sigma_{\text{sig.}}$	66	69	59

Table 6.7: Helix Parameter Resolution Observable Comparison with the context extrapolative YonedaVAE

Parameter	Observable	Signal + Background		
		YonedaVAE _{ce}	Real	No Bkg.
Δd_0	Unbiased Variance	0.1721 ± 0.0006	0.1643 ± 0.0006	0.1594 ± 0.0006
	KS Statistic	0.0066, p-value: 0.3221		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	122	146	106
	Num. Tracks $> 5\sigma_{\text{sig.}}$	143	136	112
$\Delta \phi_0$	Unbiased Variance	0.4825 ± 0.0017	0.4480 ± 0.0016	0.4429 ± 0.0016
	KS Statistic	0.0066, p-value: 0.3178		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	400	317	311
	Num. Tracks $> 5\sigma_{\text{sig.}}$	387	389	357
Δz_0	Unbiased Variance	5.9736 ± 0.0207	5.8735 ± 0.0204	5.7120 ± 0.0201
	KS Statistic	0.0066, p-value: 0.3309		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	77	71	62
	Num. Tracks $> 5\sigma_{\text{sig.}}$	275	261	238
$\Delta \omega$	Unbiased Variance	0.0068 ± 0.0001	0.0065 ± 0.0001	0.0061 ± 0.0001
	KS Statistic	0.0063, p-value: 0.3833		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	281	226	225
	Num. Tracks $> 5\sigma_{\text{sig.}}$	357	321	299
$\Delta \tan \lambda$	Unbiased Variance	0.0790 ± 0.0003	0.0726 ± 0.0003	0.0734 ± 0.0003
	KS Statistic	0.0069, p-value: 0.2803		
	Num. Tracks $< -5\sigma_{\text{sig.}}$	276	250	226
	Num. Tracks $> 5\sigma_{\text{sig.}}$	77	69	59

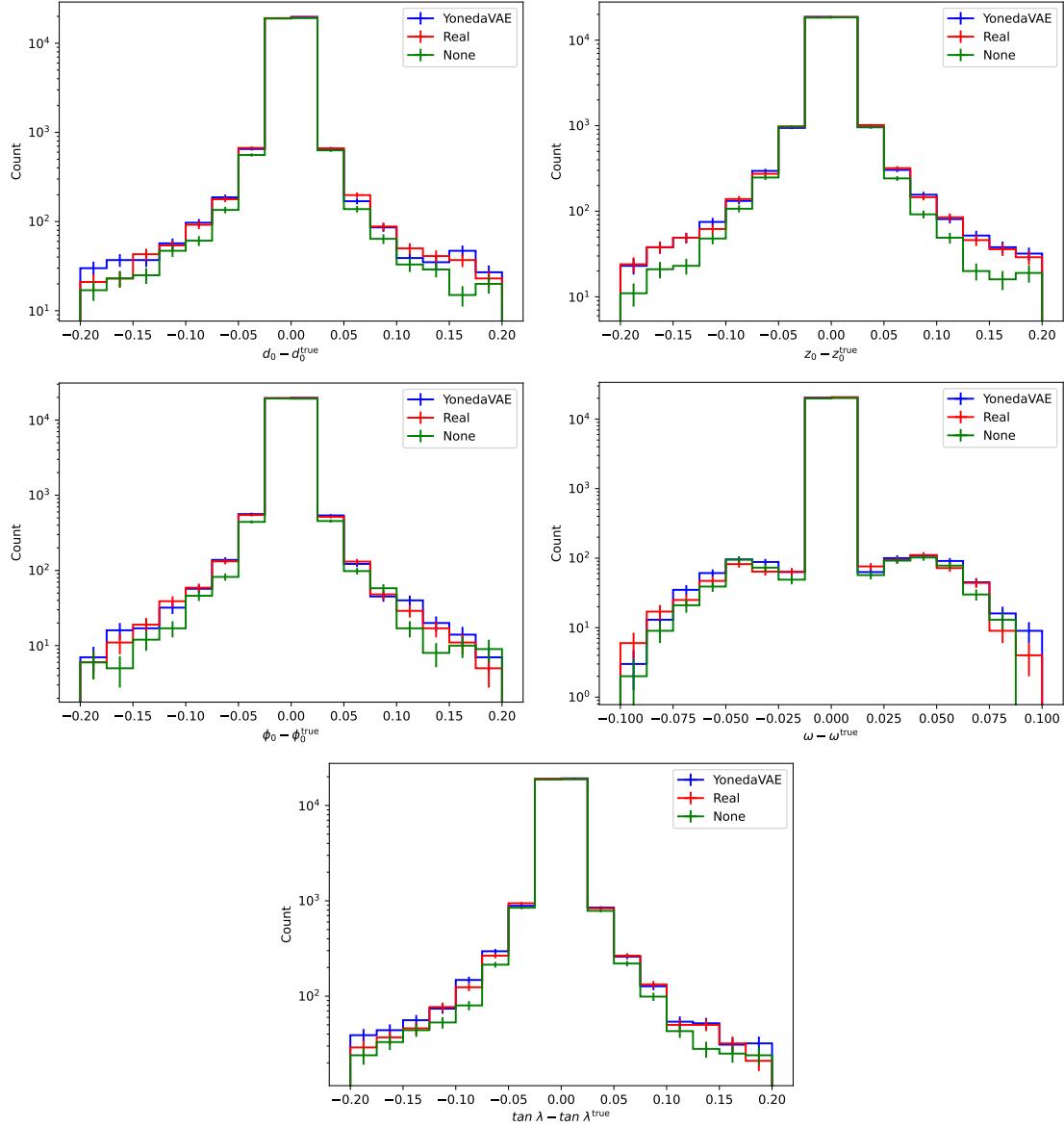


Figure 6.38: Helix parameter resolutions. For each parameter, blue represents YonedaVAE with context extrapolation, red represents Real PXD, and green represents resolution with no background overlay.

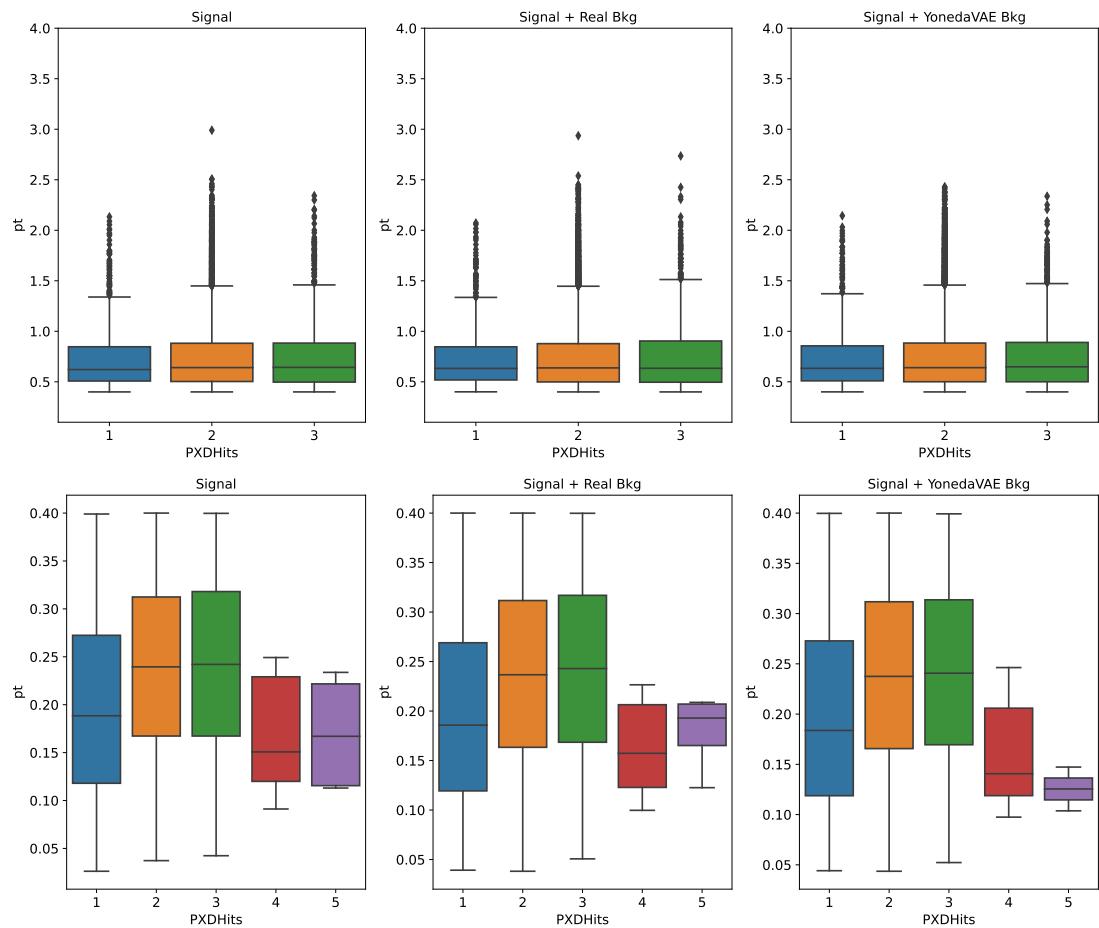


Figure 6.39: High momentum (top), and low momentum (bottom) momentum boxplot per number of PXD hits. The YonedaVAE here is with context extrapolation.

6.6 Ablation Studies

This section tries to motivate and justify various components of YonedaVAE through an ablation study. I do this by fixing the YonedaVAE’s architecture as far as possible and retraining it while a component of YonedaVAE is removed or changed as an intervention to observe the effect of the corresponding module. These results are for the context extrapolation setup where the model does not have access to the individual sensor feature (sensor occupancy) during inference. Due to the long training time, I could do this ablation only for Adaptive Top-q sampling (see Figure 6.41 and Figure 6.42), Attention gating (see Figure 6.43), self-distillation mechanism (see Figure 6.44), and Yoneda Pooling (see Figure 6.45). In this study, Adaptive Top-q sampling is compared with top-k and i.i.d sampling (described in Section 6.2.1). Note that I did not compare the original models that use these sampling methods (doing so would have ended in a huge difference in comparison as they were not designed for context extrapolation). Attention-gating and self-distillation mechanisms are compared with the situation where these mechanisms are removed. Yoneda pooling is compared with the original PNA pooling [369]. I have to emphasize that only these modules were removed/changed, and the other components stayed intact.

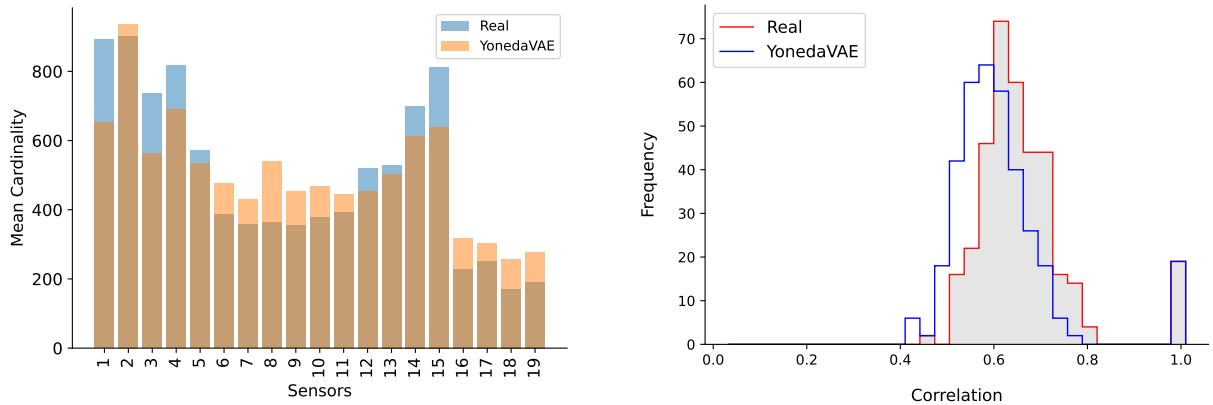


Figure 6.40: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE as a comparison baseline.

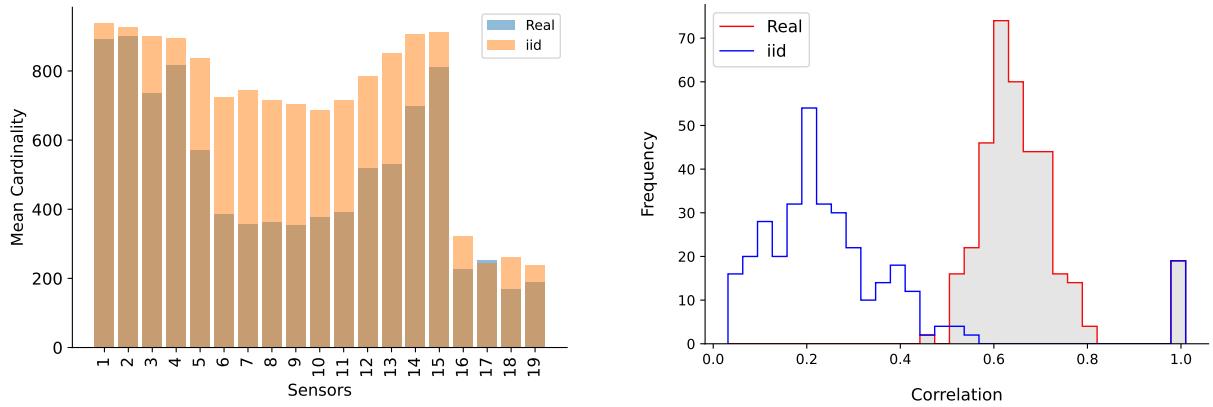


Figure 6.41: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Adaptive Top-q sampling, and only with i.i.d sampling.

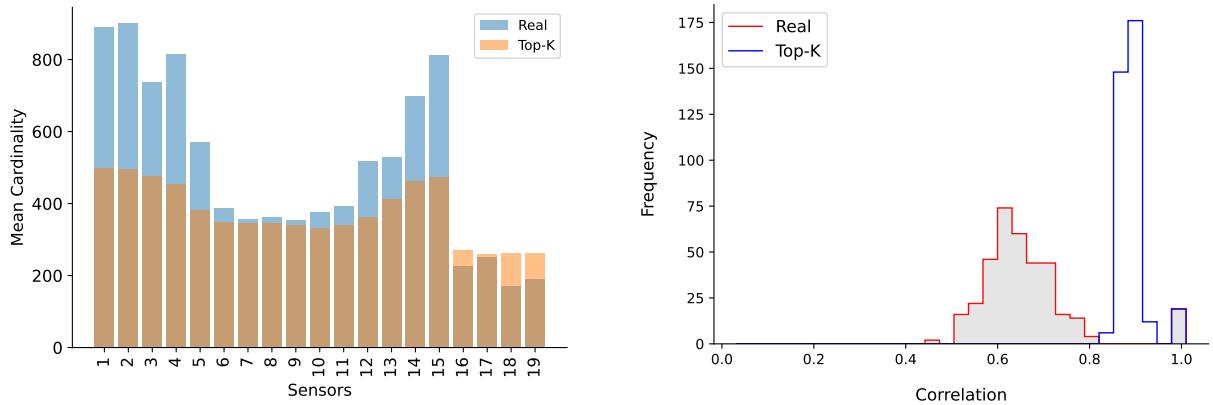


Figure 6.42: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Adaptive Top-q sampling, and only with Top-k sampling.

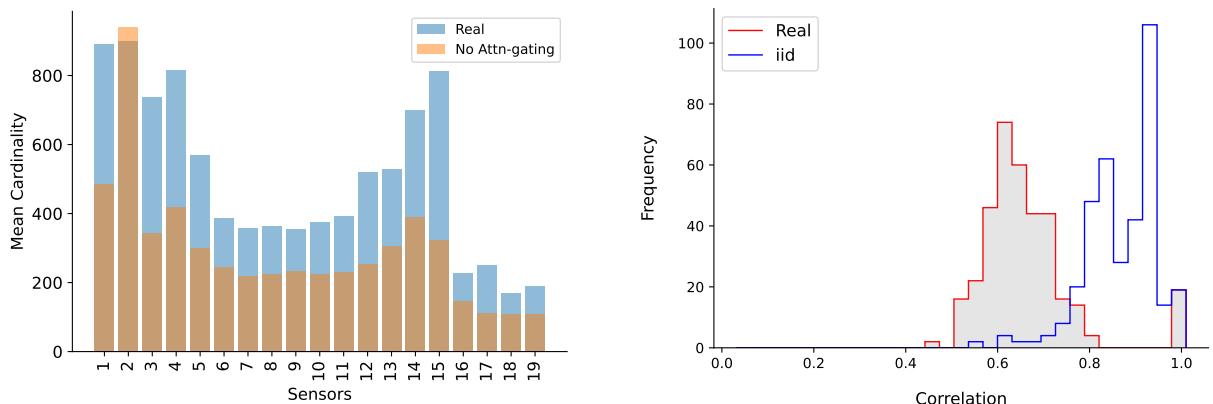


Figure 6.43: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Attention gating mechanism.

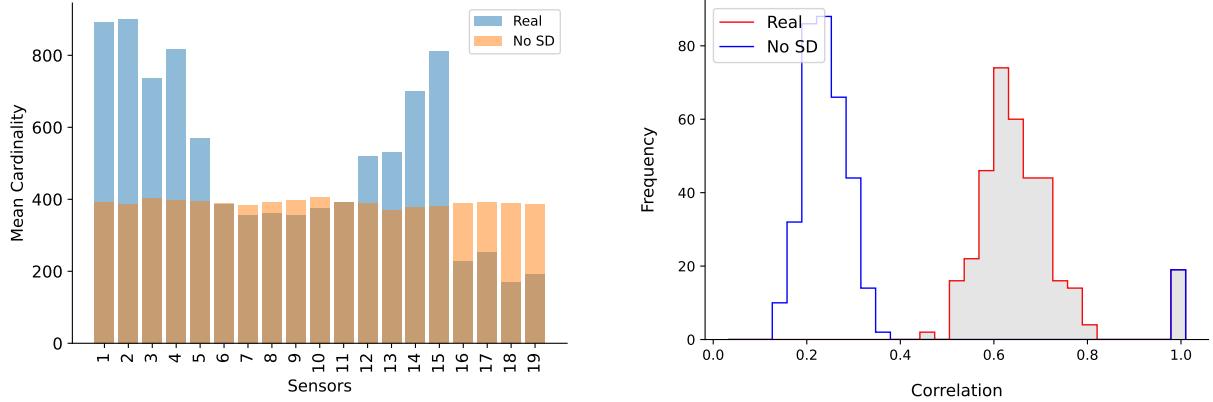


Figure 6.44: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the self-distillation mechanism during training.

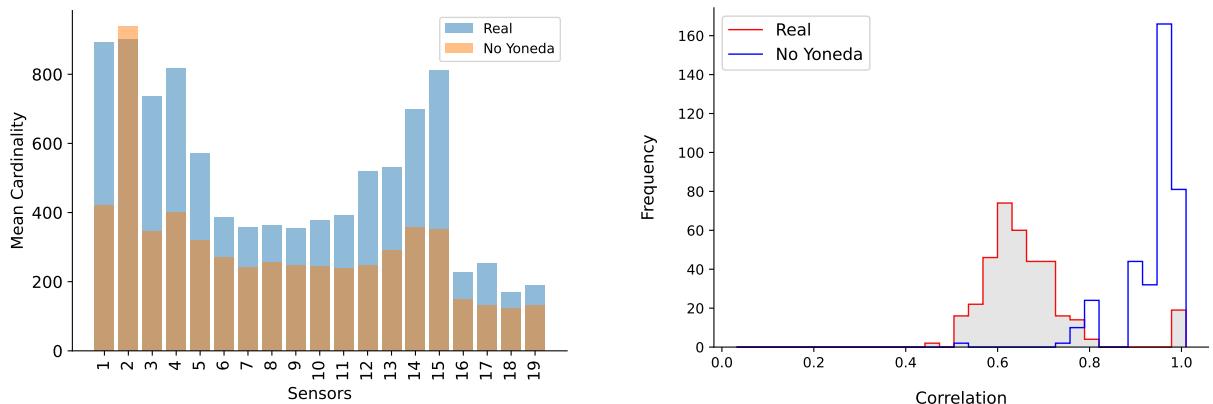


Figure 6.45: Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Yoneda pooling mechanism during training (only using the PNA).

6.7 Conclusion

This study provides an in-depth exploration of the complexities involved in simulating the real PXD background data with fine-grained features. The narrative started by discussing the motivation behind adopting unordered and variable-sized representations of the data and going to the Out-Of-Distribution (OOD) domain. The study then takes a closer look at the key elements crucial for OOD generation, specifically focusing on set generation, length extrapolation, and context extrapolation. It also addresses the associated challenges and limitations.

The issues discussed in this chapter are recognized as open questions within both Multi-Set generation and detector simulation communities. On the Multi-Set generation side, OOD simulation is particularly a notorious problem, for instance, in biological protein design, which is the problem of generating novel sequences/sets beyond the training data that exceeds the existing natural sequences in terms of certain attributes.

In the field of detector simulation, recent work has shown limitations. For example, the most current study [252] on detector point cloud simulation was only able to reconstruct in-distribution events with a cardinality of up to 6,000. In contrast, YonedaVAE has broken new ground by generating out-of-distribution events with a cardinality of up to 100,700.

This work begins with an introductory overview of Category Theory and the Yoneda perspective, leading to the introduction of YonedaVAE. This new VAE-based model is inspired by Category Theory for the generation of sparse ultra-high-granularity, variable-length, and correlated real PXD detector background responses. It does so while conditioning on both the amount of background and the geometry of the detector. The introduction of YonedaVAE is a paradigm shift by extending the capability of simulating PXD hits under conditions that have not been encountered during training.

During this path, first, I conjectured a unified and formal correspondence between relational reasoning and Yoneda lemma. Specifically, within the context of event-based reasoning, the idea is that Yoneda Embedding as an intra-event reasoning module offers more than just structural (syntactic) insights into an event. It also elucidates how to interpret sensor data in the context of its relationships and interactions with other sensors, thus adding a semantic layer. Consequently, the Yoneda Embedding and its resulting presheaf representation serve as a pooling mechanism, capturing what can be termed “contextual representation” for each event.

Then, for YonedaVAE’s set generator to be able to extrapolate well beyond the training data during inference, a self-distillation mechanism was introduced to distill the encoded (relational) information from the YonedaVAE’s encoder to a Normal prior distribution. Additionally, to handle detector hits with inter-event and intra-event variable cardinality, the novel *Adaptive Top- q sampling* was introduced. This method dynamically determines the number of points to be sampled for each PXD sensor based on the shape of the probability distribution for each event. This approach avoids relying on fixed parameters, offering a more flexible way to sample data and making the model more adaptive to the complexity of individual events.

During this study, many tricks and modifications were introduced, such as attention gating that facilitates the extrapolative abilities of the model. As a result of introducing YonedaVAE, all In Distribution level marginal distributions were accurately generated, surpassing the capabilities of IEA-GAN (Chapter 5). In particular, finally, the elusive intra-event correlation was captured.

In the Out-Of-Distribution region, YonedaVAE excelled at both length extrapolation and context extrapolation, maintaining high accuracy over marginal distributions and other lower-level metrics. The length extrapolation was manifested in the case where the model has to populate the previously unseen PXD hit spots and infer what a background event with a higher cardinality looks like. The context extrapolation was demonstrated through the model's ability to generate events with the desired amount of background and learn to generalize to attribute (context) profile beyond the training data and generate point clouds with the correct cardinality and intra-event correlation.

During the evaluation phase, this study also showcased the application Vendi Score, a metric to quantify the diversity of the generated samples, for PXD background diversity measurement. Then, for the first time, Topological Data Analysis (TDA) was studied for the PXD background data. Along with the PXD clustering analysis, It offered valuable insights into the patterns and intricacies of how the PXD point cloud hits cluster.

Yet, some open challenges persist. While it's crucial to acknowledge that the model has some limitations, particularly in the intra-event correlation *only* during context extrapolation, which is a luxury by itself, the current achievements shouldn't be understated. Attaining this level of precision in context extrapolation is marking a considerable step forward in the discipline. Moreover, the YonedaVAE's ability to capture long cluster patterns needs further exploration and possibly enhancement, especially considering its influence over the downstream tracking analysis and the fake rate of reconstruction. The efficacy of YonedaVAE in approximating helix parameter resolutions, as indicated by closely matching unbiased variances and statistically consistent Kolmogorov-Smirnov p-values, paints a promising picture of its utility for simulating real background conditions. However, the under-representation of such long clusters in the YonedaVAE-generated data as a point cloud generative model could introduce systematic biases or limitations in the track reconstruction efficiency. This aspect may not immediately manifest in global performance metrics like unbiased variance or p-values but could have repercussions in specialized scenarios, including track reconstruction in high-density environments or the estimation of rare track configurations. Hence, a deeper investigation into the relationship between cluster size and helix parameter resolution could provide additional layers of validation for YonedaVAE's robustness and reliability.

Chapter 7

Summary and Outlook

7.1 Summary

Rising storage cost and increasing computational time for detector simulations pose a considerable challenge for both existing and upcoming collider experiments. For Belle II with the full simulation of PXD background hits, the highest granularity sub-detector at Belle II is the most intense bottleneck of the detector simulation chain. This leads to a desire for new, fast, efficient, amortized, and accurate surrogate methods. This thesis narrated the story of how the current state-of-the-art deep generative models fall short in this task, and highlights imperative to develop new ideas while incorporating the inductive biases of this specific domain and our problem.

A significant challenge I encountered during my research was the absence of a thorough, taxonomic, and experiment-focused review or survey of deep generative models in the context of detector simulation and particle physics. This gap led to confusion and resulted in the loss of valuable insights in this domain, as well as unnecessary and time-consuming failed experiments. To address this issue, in Chapter 4, I aimed to offer a comprehensive taxonomy of deep generative models for detector simulation, drawing on nearly all published papers and works available as of the writing date. This survey delves into detailed discussions and categorizations from both algorithmic and application standpoints. I explore the three main use-cases: statistics amplification, amortized generation, and out-of-distribution (OOD) simulation, while also highlighting the strengths and weaknesses of current methods across various particle physics experiments. Additionally, I identify the limitations of state-of-the-art approaches in the specific context of ultra-high-granularity PXD background simulation.

The critical question in employing generative models for efficient simulation of PXD background centers on designing a model capable of sampling PXD data while preserving accurate correlations among sensors within a single event. This issue is intimately connected to the model's ability to grasp the inherent intra-event correlations, a task that becomes especially challenging due to the ultra-high resolution of PXD data. If a generative model can accurately capture these correlations, facilitated by the correct inductive biases, it can

be employed to simulate the entire PXD detector. While most existing surrogate methods concentrate on generating samples that account for intra-sensor (or layer) correlations, none have yet explored the significance of intra-event correlation or methods to accurately capture it.

Therefore, Chapter 5 tackles this issue by initially exploring the importance of intra-event correlation in subsequent track reconstruction analyses. The chapter demonstrates that a loss of intra-event correlation leads to less accurate Helix parameter resolution in the process of track reconstruction. Subsequently, I introduce the Intra-Event Aware GAN (IEA-GAN), a GAN-based surrogate model enhanced with self-supervised relational reasoning. This model is capable of generating PXD events with an unprecedented granularity of 7.68×10^6 , a level of resolution that has never before been attempted. Developing IEA-GAN was a story of patience and an extreme amount of hyper-parameter tuning, where each run took between 3 to 5 days over about 300 runs. The chapter classifies the PXD generation task as a fine-grained image analysis problem, characterized by large intra-class and small inter-class variations. This adds an additional layer of complexity to the generative task, given the ultra-high resolution of PXD data. As a solution to these challenges, I propose various methods, namely the relational reasoning module, Uniformity Loss, and Intra-Event Aware Loss. Each of these methods are designed to address specific issues and introduce domain-specific inductive biases to the model. The relational reasoning module is responsible of capturing the correlation among the samples in an event. The Uniformity Loss ensures that the generated samples exhibit as much diversity as possible. The Intra-Event Aware Loss is a novel, non-adversarial, self-supervised loss formulated to enhance the generator's understanding of intra-event dyadic relationships.

For evaluation purposes, I not only compared image-level marginal distributions, where IEA-GAN outperformed the highly-tuned state-of-the-art models in the conditional image generation task, but also introduced the data-driven FID [22] and KID [23] metrics for detector simulation. This study also provided interpretations for these metrics in the context of PXD's low-level image features, proving useful during both validation and hyper-parameter tuning as reliable, generic metrics. Ultimately, even though IEA-GAN captured only weak correlations, it proved to be a more potent candidate for replacing Geant4 PXD data compared to other state-of-the-art models. As a consequence, IEA-GAN has been implemented into the basf2 software as an amortized surrogate model for simulation. Geant4 PXD data has also been publicly released as the first ultra-high granularity dataset for future research. The integration of IEA-GAN as a surrogate model resulted in a speed-up factor of up to 147 and storage release by half.

Two main concerns remained regarding the results from IEA-GAN. First, although IEA-GAN had some success in capturing intra-event correlations, these correlations were relatively weak, and the true underlying relationships between the mean occupancies of sensors had yet to be fully understood. Second, there was the issue of data sparsity; the real PXD background data is significantly more sparse than the Geant4 simulated data, and this sparsity also exhibited high variance (or greater intra-class variation). As a result, it could be challenging for CNN-based models like IEA-GAN to meaningfully generate

grid-based (image) representations for PXD background data.

An important challenge in working with real PXD background data is its reliance on actual experiments for gathering the necessary random triggers. Consequently, real luminosity and beam-parameter-dependent PXD background data that go beyond current experimental limits are unavailable, leaving us reliant on computationally demanding simulations. This underscores the critical need for a surrogate model that can effectively generalize to Out-Of-Distribution (OOD) luminosity regions.

To address the aforementioned issues and problems, Chapter 7 introduces YonedaVAE, an equivariant point cloud VAE-based model featuring a self-distillation set generation mechanism and Adaptive Top-q sampling. YonedaVAE is conditioned on both the geometry of the PXD detector’s sensors— using rotary positional embedding— and the amount of PXD background (or occupancy) as control parameter. The model is trained on random trigger PXD background data from Experiment 12 of Belle II, which had a peak recorded luminosity of $1.42 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ and a mean occupancy of 0.06%. It was then tested on data from Experiment 26 of Belle II, which had nearly double the peak luminosity— $2.68 \times 10^{34} \text{ cm}^{-2} \text{ s}^{-1}$ —and a mean occupancy of 0.32%. This evaluation serves to demonstrate YonedaVAE’s capabilities for extrapolation into OOD and new luminosity regions.

YonedaVAE introduces novel technologies for zero-shot irregular multi-set generation, where both variable intra-category (intra-event) and variable inter-category (inter-event) multiplicity are crucial. In the encoder of YonedaVAE, Yoneda pooling serves as a learnable, relational pooling mechanism that extends PNA pooling to account for the relationships between objects within an event. As such, I provide a Category Theoretical formulation for relational reasoning and the intra-event aware mechanism. It conjectures that Yoneda Pooling generates a relational observable embedding that incorporates both instance-level information and inter-object relations. Unlike PNA pooling, which only describes the structure (syntax) of an event, Yoneda pooling delves into understanding each sensor’s information in the context of its relationships with other sensors (semantics). This allows one to assert that Yoneda pooling and the ensuing presheaf representation capture a form of “contextual representation.”

Illustrated in Section 6.5, YonedaVAE’s set generator introduces a novel training mechanism is of paramount importance for OOD generation: the self-distillation mechanism for the set generator. Drawing inspiration from self-supervised learning, this mechanism enables self-distillation in the Transformer decoder set generator to enforce an isomorphism between two distinct views of the latent space— the learned one by the encoder and the prior distribution. Through this approach, YonedaVAE attains enhanced generalization during inference where it only samples from from the prior Normal distribution.

Another crucial element of YonedaVAE is its Adaptive top-q sampling mechanism. This method accounts for the sparsity of the data and learns the potential skewness in the cardinality distribution within an event (or mini-batch). Consequently, it selects the most relevant set of top points, having the highest probability, ranging from $\mathcal{O}(10)$ to $\mathcal{O}(100000)$ number of points for each sample within an event.

As a result of deploying YonedaVAE, all In-Distribution (ID) level marginal distributions were accurately replicated, surpassing the performance of IEA-GAN. Specifically, the elusive intra-event correlation was finally captured. In Out-Of-Distribution (OOD) scenarios, YonedaVAE excelled in both length and context extrapolation, sustaining high accuracy in marginal distributions and other low-level metrics. Length extrapolation was evident when the model had to populate previously unseen PXD hit spots and infer the characteristics of a background event with higher cardinality. Context extrapolation was showcased by the model’s capability to generate events with the specified higher amount of background, learn to generalize to attribute (context) profiles beyond the training data, and produce point clouds with accurate cardinality and intra-event correlation.

After demonstrating YonedaVAE’s performance in marginal distributions and its advantages in FID and KID metrics, this thesis introduces additional metrics aimed at assessing the diversity of the generated samples. A key question in using surrogate models for detector simulation is the extent to which the model can sample diversely compared to real data. This question is intrinsically tied to the measurement and comprehension of a dataset’s diversity, given a particular similarity measure. Rooted in Ecology, the Vendi Score [24] provides a quantitative framework for evaluating data diversity. This thesis incorporated this metric in the context of PXD detector simulation and exhibited the amplification results that align with previous work [244], but from an information-theoretical standpoint.

Another important question that must be asked in detector signature simulation, especially for PXD, is how complex the hits are distributed from a geometrical and topological perspective. In particular, how are the points of a PXD point cloud clustered? And How many complex clusters do they create? This study provides, for the first time in particle physics detector simulation, a Topological Data Analysis (TDA) approach to answer these questions. TDA leverages the interpretable knowledge of topological features, such as connected points and loops in PXD data sets, to better understand their structural properties at multiple scales. As a result, this chapter provides insights into the clustering profile of PXD data. I complimented this study by analyzing the PXD background cluster information from the basf2 clusterizer’s point of view. Given the ultra-high granularity of the PXD background data, YonedaVAE, as a point cloud generative model, could reach a satisfactory and unpredictable result using the clustering algorithm of basf2. This, in particular, is very important from the downstream track reconstruction analysis as the clusters serve as a proxy for the interaction of charged particles with the PXD. Towards the end, as the final stage of the evaluation, I provide the track reconstruction analysis to compare the effect of experiment 26 real random trigger PXD background to the YonedaVAE generated PXD background. Despite some deviation in the clustering profile of YonedaVAE’s generated background from the real PXD ones, track reconstruction shows a good agreement between the two, even over the extrapolation region.

Another crucial question in detector signature simulation, particularly for PXD, concerns the geometric and topological complexity of hit distributions. Specifically, how are the points

in a PXD point cloud clustered, and how many complex clusters do they form? For the first time in particle physics detector simulation, this study introduces a Topological Data Analysis (TDA) approach to address these questions. TDA offers insights into topological features like connected points and loops in PXD hits, to better understand the multi-scale understanding of their structural properties. To complement this, I analyzed the clustering of PXD background data from the perspective of the basf2 clusterizer. Given the ultra-high granularity of PXD background data, YonedaVAE, as a point cloud generative model, achieved promising and unexpected results using basf2’s clustering algorithm. This is particularly significant for downstream track reconstruction analysis, where clusters act as indicators of charged particle interactions with the PXD. In the final stage of evaluation, I present a track reconstruction analysis that compares the effects of experiment 26’s real random trigger PXD background to those generated by YonedaVAE. Although there is some deviation in the clustering profile between the two, track reconstruction reveals a good agreement, even in extrapolation regions.

This outcome represents a significant milestone for both the fields of deep generative models and fast simulation in particle physics. It demonstrates the feasibility of context and length extrapolation beyond the training data. Specifically, YonedaVAE, trained on a maximum event cardinality of $19 \times 400 = 7600$, successfully extrapolated to a maximum event cardinality of $19 \times 5300 = 100700$ with different intra-event correlation profile. Consequently, YonedaVAE achieved an unparalleled hit (point cloud) cardinality of 100700 per event in detector simulation, far exceeding the previous record of 6000 set by point cloud-based fast simulation models [252].

7.2 Future Directions and Opportunities

The High-Luminosity Large Hadron Collider (HL-LHC) [257] is expected to surpass the LHC’s design integrated luminosity by increasing it by a factor of 10. For instance, the upcoming high-granularity Calorimeter (HGCAL) with roughly 6.5M channels, or the ITk 3D pixel detector at the HL-LHC [304] with around 1M information channels, will massively increase the geometry and precision complexity. This leads to a dramatic increase in the time and storage to simulate the detector [305]. However, these granularities far surpass the capacity of existing methods, pointing to the urgency of developing more advanced simulation approaches, while the current studies in generative models barely scratch the surface of the profound challenges posed by future detector simulations. Thus, much more effective and efficient ultra-high-granularity detector simulations are required.

This work has significant impacts on high-granularity fast and efficient detector simulations. YonedaVAE and IEA-GAN, with robust controllable sampling, are the first potential candidates for simulating the corresponding high-resolution and ultra-high-granular detector signatures with the remarkable capability of generating more than 7.5M information channels.

In addition, YonedaVAE presents OOD simulation for control parameters beyond the

current experimental limits, which has applications in both event generation and real detector simulation. YonedaVAE also presents potential applications in de novo protein design, which is a process that involves the generation of novel amino acid sequences to produce proteins with desired functions, such as enhanced stability and foldability, new binding specificity, or enzymatic activity [380]. Proteins can be grouped into different categories based on the arrangement and connectivity of their secondary structure features, such as alpha helices and beta sheets. Then, YonedaVAE can be incorporated for the extrapolative generation of novel foldable protein structures where category-level reasoning is of paramount importance.

The higher goal of this study was to demonstrate the importance of incorporating relational inductive bias and intra-event reasoning in the design of deep generative models for the emulation and extrapolative generation of ultra-high granular PXD backgrounds; however, several challenges will still have to be addressed.

For first and foremost important challenge for YonedaVAE was the clustering profile of the PXD hits. In general, capturing “continuity” is a notorious problem in point cloud generation, especially when the point cloud gets re-scaled by large and imbalanced factors (by [255, 250, 768]). Thus, in order to fully capture the clustering behavior of real PXD background, the future models have to incorporate clustering as an inductive bias in a differentiable way [381, 382, 383]. A related open problem is the possible interpretation of higher-order TDA artifacts (e.g. H_1). This thesis provides an interpretation of the lifetime of H_0 topological feature in the PXD background point cloud as the lifetime of connected components of large line-like clusters. However, a concrete explanation of the H_1 feature, especially in relation to the types of background hits, would be very interesting. In the case of a relationship, this can augment the surrogate model to generate PXD hits conditioned on the “type of background” by incorporating the desired H_1 profile during training.

Moreover, there existed a deviation from the true intra-event correlation of YonedaVAE only in the context extrapolation regime. It’s worth mentioning that generating out-of-distribution (OOD) data that perfectly captures the inherent context is a notorious challenge and the subject of extensive research across various domains. These include not just particle physics but also fields like drug discovery [338, 384], protein design [385], cell design [313], natural language processing [348], and weather forecasting [261]. The complexities inherent in this task indicate that the generative models presented in this thesis do not suffice for flawless context extrapolation. Additional refinements are essential for future models. Specifically, the incorporation of more refined inductive biases and regularization methods during the training phase could offer a pathway toward more reliable context extrapolation.

I think a promising avenue for future research lies in the direction of differentiable programming (DP) [386]. Through DP, software becomes differentiable through automatic differentiation (AD) [387], enabling efficient gradient computation to understand the influence of input variations on output predictions. Such gradient-based insights could be invaluable for various downstream tasks. Utilizing DP frameworks would allow particle physics simulation tools to be integrated into machine learning pipelines, thereby facilitating

joint optimization for enhanced computational efficiency. This integration paves the way for the creation of physics-informed surrogate models capable of more accurately addressing the complexities inherent in high-granularity detector simulations.

List of Publications

The findings presented in this thesis are the result of research conducted at the Ludwig-Maximilians-Universität München. The research outcomes span various stages of my academic dissemination. Specifically, the results detailed in Chapters 4, 5, and 6 have been submitted for publication or are actively being prepared for submission. Some of these findings have already been published, while others are currently under review or in preparation for submission to peer-reviewed journals.

- [1] M. Srebre, P. Schmolz, H. Hashemi, M. Ritter, and T. Kuhr 2020. **Generation of Belle II Pixel Detector Background Data with a GAN**, EPJ Web of Conferences, 245, 02010. [10.1051/epjconf/202024502010](https://doi.org/10.1051/epjconf/202024502010).
- [2] H. Hashemi, N. Hartmann, T. Kuhr, M. Ritter, and M. Srebre 2021. **PE-GAN: Prior Embedding GAN for PXD images at Belle II**, EPJ Web of Conferences, 251, 03031. [10.1051/epjconf/202125103031](https://doi.org/10.1051/epjconf/202125103031).
- [3] H. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn, and T. Kuhr 2022. **Intra-Event Aware Imitation Game for Fast Detector Simulation**, NeurIPS 2022 workshop paper (Machine learning and the physical sciences). [NeurIPS_ML4PS_2022_97](#).
- [4] H. Hashemi, N. Hartmann, S. Sharifzadeh, J. Kahn, and T. Kuhr 2023. **Ultra-High-Resolution Detector Simulation with Intra-Event Aware GAN and Self-Supervised Relational Reasoning**, Published in *Nature Communications*. [s41467-024-49104-4](https://doi.org/10.1038/s41467-024-49104-4)
- [5] B. Hashemi, C. Krause, 2023. **Deep Generative Models for Detector Signature Simulation: A Taxonomic Review**, Under Peer Review by *Reviews in Physics*. [j.revip.2024.100092](https://doi.org/10.4236/j.revip.2024.100092)
- [6] B. Hashemi, 2023. **YonedaVAE: Self-Supervised Out-of-Distribution Multi-Set Generation for Amortized Simulations**, Under Submission to *Machine Learning: Science and Technology*.
- [7] B. Hashemi, B. Rieck, 2024. **Topological Data Analysis for Particle Physics Detector signatures**, Under Submission to *Machine Learning: Science and Technology*.

List of Figures

2.1	The Belle II coordinate system.	6
2.2	The Belle II detector with a zoomed-in view of the inner part adopted from [11]	7
2.3	A schematic and simplified diagram of the Belle II data flow, taken from [15]	9
2.4	The first picture of the full PXD during the commissioning phase in DESY Hamburg.	13
2.5	PXD specifications and detailed information adopted from [18]	13
2.6	Pixel detector layout The pixel detector (PXD) is the inner-most sub-detector of the Belle II experiment (a) and is configured in a two-layered overlapping sensor structure (b) . Geant4 Simulated PXD Background image examples (c) for sensors 7 (top) and 25 (down).	15
2.7	PXD occupancy (layer 1 here) increases with beam currents and luminosity, coming from raw data. The occupancy of PXD is defined as the ratio of the number of hits to the total number of PXD pixels. The Experiment numbering corresponds to the Belle II run periods.	16
2.8	Geant4 simulated Background data volume comparison of PXD data per event compared to other sub-detectors at Belle II. The PXD background data is nearly as much as the combined background data for all other sub-detectors.	19
2.9	PXD Background simulation options.	20
2.10	The perigee parametrization of the track helix adopted from [25]	23
3.1	Transformer Encoder and Decoder Architecture, taken from [36]	29
3.2	SAGAN architecture, taken from [90]	43
3.3	BigGAN’s main building blocks, taken from [82]	44
3.4	BigGAN-deep’s main building blocks, taken from [82]	44
3.5	Contrastive learning at the interface of supervised and self-supervised learning, adopted from [101]	49
3.6	Deep Metric Learning vs. vanilla Classification	50
3.7	The framework for SimCLR. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two related views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a variant of InfoNCE loss, NT-Xent loss.	52
3.8	BYOL’s overall architecture, adopted from [113]	54

3.9	SimSiam’s overall architecture, adopted from [114]	55
5.1	Fine-grained image analysis vs. generic image analysis (using visual classification as an example), adopted from [266].	72
5.2	An illustration of fine-grained image analysis which lies in the continuum between the basic-level category analysis (i.e., generic image analysis) and the instance-level analysis (e.g. car identification), adopted from [266] . . .	72
5.3	Key challenges of fine-grained image analysis, small inter-class variations, and large intra-class variations. Different Tern species, one species per row, with different instances in the columns, adopted from [266]	73
5.4	PXD background signatures across events for different sensors. The horizontal cases demonstrate various profiles (different clustering, occupancy, etc.) across different events. The vertical cases exhibit similar profiles across sensors.	74
5.5	Charge intensity distribution for the WGAN-gp without conditioning [172] on PXD as the baseline.	77
5.6	IEA-GAN architecture (a) and Relational Reasoning Module components (b).	82
5.7	IEA-loss imposes a pair-wise fine-grained class-to-class imitation force for the generator. sg indicates that for the discriminator gradients are stopped and only the generator’s gradients will be updated.	86
5.8	Uniformity over a 3-dimensional cross-polytope immersed in a unit hypersphere	87
5.9	Different FID values with respect to the Charge and Occupancy distributions	95
5.10	Pixel intensity distribution in linear (top left) and logarithmic scale (top right), the distribution of the occupancy (bottom left) and the mean occupancy per sensor (bottom right) for 10 000 events.	96
5.11	Spearman’s correlation between the occupancy of Geant4 sensor images (left), and sensor images from IEA-GAN (center), sensor images from PE-GAN (right).	96
5.12	The boxplot for comparing the d_0 resolution in the presence of correlated background and uncorrelated background at high momentum regime. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.	98
5.13	Helix parameter resolutions for d_0 (row 1), z_0 (row 2), ϕ_0 (row 3), ω_0 (row 4), and $\tan \lambda$ (row 5). For each parameter, the left figure corresponds to the IEA-GAN simulated background, and the right figure corresponds to the PE-GAN simulated background. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.	100
5.14	Comparison of the FID between different IEA-losses	101
6.1	Random-trigger (real) PXD background (left) and Geant4 generated PXD background (right) in a bitmap format. As stated before, the sparsity of the real PXD background is much less than the Geant4 version.	109

6.2	Illustrative models for set generation borrowed from [94]. The set's point count can either be derived from the dataset's distribution (a) or inferred from the latent vector (b). Although any equivariant function can serve for the update h , the set creation g encapsulates the intricacies of set generation. When generating graphs, edge weights complement the node features matrices \mathbf{M}^0 and \mathbf{M}	110
6.3	Existing creation methods, borrowed from [94], for mapping a latent vector \mathbf{z} to a set of points \mathbf{X}^0 . First-n creation empirically gives the best performance. It learns a reference set represented by a matrix \mathbf{X}_{ref} , and concatenates the latent vector to each point of this set.	111
6.4	Commutative diagram in a category	118
6.5	Commutative square for a natural transformation	119
6.6	(right) The cardinality distribution in log scale for the training that belongs to Experiment 12 data and extrapolation (inference) region that belongs to Experiment 26 data. (left) The overall cardinality distribution of each event can go up to 100k points per event.	123
6.7	(left) The max cardinality distribution in log scale for the training that belongs to Experiment 12 data as event-level attributes. (left) The max cardinality distribution in the log scale for inference that belongs to Experiment 26 data as event-level attributes.	124
6.8	Eventformer	126
6.9	Yoneda-Pooling	127
6.10	Self-Distillation of the learned latent space via the VAE encoder to the desired prior. It is termed “self-distillation” since the backbones models, the Transformer Decoders, share weights.	130
6.11	Adaptive Top-q sampling proposes a point sampling from the top n_i portion of the probability mass where n_i is adaptively learned in context, dynamically expanding and contracting the candidate pool in each event.	134
6.12	YonedaVAE in training (top) and inference (bottom)	135
6.13	Random-trigger (real) PXD background (left) and YonedaVAE generated PXD background (right) in a bitmap format. As stated before, the sparsity of the real PXD background is much less than the Geant4 version (last chapter).	138
6.14	Marginal Distributions for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to sensor occupancies as sensor-level features.	140
6.15	Pearson Correlation between sensor's mean occupancy for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to sensor occupancies as sensor-level features.	141
6.16	Marginal Distributions for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to max cardinality per event as event-level features.	142

6.17 Pearson Correlation between sensor's mean occupancy for YonedaVAE on the validation data (Experiment 12 with arbitrary cardinalities) with access to max cardinality per event as event-level features.	143
6.18 Marginal Distributions for YonedaVAE with length extrapolation on the OOD data (Experiment 26) with access to sensor occupancies, as sensor-level features.	144
6.19 Pearson Correlation between sensor's mean occupancy in the length extrapolation regime for YonedaVAE on the OOD data (Experiment 26)	145
6.20 Spearman Correlation between sensor's mean occupancy in the length extrapolation regime for YonedaVAE on the OOD data (Experiment 26)	145
6.21 Marginal Distributions for YonedaVAE with context extrapolation on the OOD data (Experiment 26).	146
6.22 Pearson Correlation between sensor's mean occupancy in the context extrapolation regime for YonedaVAE on the OOD data (Experiment 26)	147
6.23 Spearman Correlation between sensor's mean occupancy in the context extrapolation regime for YonedaVAE on the OOD data (Experiment 26)	147
6.24 (a) The Vendi Score, VS in the figure, adopted from [24], can be interpreted as the effective number of unique elements in a set of events. It increases linearly with the number of modes in the dataset. IntDiv, the expected dissimilarity (inverse correlation), becomes less sensitive as the number of modes increases, converging to 1 (or becoming uncorrelated). (b) Combining distinct similarity measures can increase the Vendi Score, as should be expected of a diversity metric while leaving IntDiv ((inverse correlation)) unchanged. (c) Vendi Score takes into account correlations between samples. The Vendi Score is highest when the samples differ in many attributes and the attributes are not correlated with each other.	148
6.25 Oriented k-simplices for $k = 0, 1, 2, 3$. These k -simplices are the building blocks used to construct a simplicial complex from a point cloud of data.	151
6.26 The Čech complex (left) and the and Vietoris-Rips (right) of a finite point cloud in the plane \mathbb{R}^2 . The bottom part of the Čech complex is the union of two adjacent triangles, while the bottom part of the Rips complex is the tetrahedron spanned by the four vertices and all its faces. The dimension of the Čech complex is 2. The dimension of the Vietoris-Rips complex is 3.	152
6.27 Example of the topological barcode of a Vietoris-Rips complex, adopted from [377]. The top four figures display the simplicial complex of 18 points for different values of the proximity parameter ϵ . The vertical lines in the barcode correspond to these four levels of ϵ . The number of horizontal bars intersecting each line gives the values of $\mathbf{b}(\epsilon) = (\mathbf{b}_0(\epsilon), \mathbf{b}_1(\epsilon))$. For the parameters selected, $\mathbf{b}(1.5) = (18, 0)$, $\mathbf{b}(5.0) = (11, 0)$, $\mathbf{b}(7.0) = (4, 1)$, and $\mathbf{b}(9.5) = (1, 2)$	154

6.28	The set filtration of the distance function to a point cloud and the construction of its persistence barcode as the radius of balls increases, adopted from [376]. The blue curves in the unions of balls represent 1-cycles associated with the blue bars in the barcodes.	156
6.29	Persistent diagrams for each example of possible dense PXD hits	160
6.30	An example persistent diagram for a point cloud belonging to sensor 1 with 255 set cardinality. Notice the short-lived H_0 features from PXD and YonedaVAE.	161
6.31	Histograms for H_0 and H_1 feature lifetimes for the real PXD hits, YonedaVAE generated samples, and samples with random hits.	161
6.32	Comparison between the real PXD clustering information and the YonedaVAE generated ones.	162
6.33	The boxplot for comparing the Helix parameter resolutions in the presence of correlated background and uncorrelated background at high momentum regime. The ± 5 standard deviations interval for the no-background case is shown in all 3 cases for reference as a red dashed line.	164
6.34	Pearson (top) vs Spearman (bottom) correlation between the true and reconstructed Helix parameters with and without background.	165
6.35	Short Caption	165
6.36	High momentum (top), and low momentum (bottom) momentum boxplot per number of PXD hits. The YonedaVAE here is with length extrapolation.	166
6.37	Helix parameter resolutions. For each parameter, blue represents YonedaVAE with length extrapolation, red represents Real PXD, and green represents resolution with no background overlay.	167
6.38	Helix parameter resolutions. For each parameter, blue represents YonedaVAE with context extrapolation, red represents Real PXD, and green represents resolution with no background overlay.	170
6.39	High momentum (top), and low momentum (bottom) momentum boxplot per number of PXD hits. The YonedaVAE here is with context extrapolation.	171
6.40	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE as a comparison baseline.	172
6.41	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Adaptive Top-q sampling, and only with i.i.d sampling.	173
6.42	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Adaptive Top-q sampling, and only with Top-k sampling.	173
6.43	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Attention gating mechanism.	173
6.44	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the self-distillation mechanism during training.	174
6.45	Mean cardinality (left) and mean cardinality correlation (right), for YonedaVAE without the Yoneda pooling mechanism during training (only using the PNA).	174

List of Tables

2.1	Summary of Belle II detector components, adopted from [15]	8
5.1	The Architecture of the Generator and the Discriminator. “ <i>ch</i> ” denotes the number of multiplicative channels, “ResBlock”, the residual blocks introduced in BigGAN-deep, “BN”, the batch normalization, “ReLU”, the rectified linear unit, “Norm”, a normalization operator, “Proj” and “Linear”, a linear projection or an MLP.	80
5.2	FID and KID comparison between models (all models in the benchmark are highly tuned to the current problem and dataset), averaged across six random seeds. The lower the FID and KID the better the image quality and diversity.	91
5.3	FID Score after Different Jittering Methods applied to the images	92
5.4	Unbiased variance and KS test results for the shuffled and unshuffled Geant4 data across the 5 Helix parameters.	99
5.5	Comparison of Unbiased Variance and KS test results for the PE-GAN and IEA-GAN models with the Geant4 reference across 5 Helix parameters for high momentum tracks.	99
5.6	FID comparison between IEA-GAN, IEA-GAN with RRM only, IEA-GAN with Uniformity loss only, and IEA-GAN with both IEA-loss, averaged across six random seeds.	101
5.7	FID comparison between two different Hypersphere projections for IEA-GAN’s discriminator, averaged across six random seeds.	102
5.8	Computational performance of IEA-GAN and PE-GAN generators on a single core of an Intel Xeon Silver 4108 1.80GHz (CPU) and NVIDIA V100 with 32 GB of memory (GPU) compared to Geant4. For the generative models, the mean and standard deviation were obtained for sets of 10 000 events, meaning that the model generates these events one at a time, not in a batch of 10 000. The time for Geant4 refers to the theoretical time it would take to run the simulation of background processes on the fly, one event at a time. The storage consumption corresponds to storing 10 000 events of 1 times the PXD background simulated information.	104

6.1	FID, KID, and VS score comparison between SOTA models for PXD generation averaged across four random seeds. They are computed over 5000 generated events. The lower the FID and KID, the better the sample quality and diversity. For the VS score, higher values indicate better sample diversity.	147
6.2	VS over the inter-sensor mean occupancy correlation for the real test data and YonedaVAE	149
6.3	FID, KID, and VS score comparison between SOTA models for PXD generation averaged across four random seeds. They are computed over 5000 generated events. The lower the FID and KID, the better the sample quality and diversity. For the VS score, higher values indicate better sample diversity.	150
6.4	Quantitative comparison between persistence diagrams using amplitudes and entropy	158
6.5	Helix Parameter Resolution observable Comparison between correlated (Unshuffled) and uncorrelated (Shuffled) events with high momentum tracks, $p_t > 0.4$ GeV.	163
6.6	Helix Parameter Resolution observable Comparison with the length extrapolative YonedaVAE	168
6.7	Helix Parameter Resolution Observable Comparison with the context extrapolative YonedaVAE	169

Bibliography

- [1] C. L. Bennett et al. “Nine-Year Wilkinson Microwave Anisotropy Probe (WMAP) Observations: Final Maps and Results”. In: *The Astrophysical Journal Supplement Series* 208.2 (Sept. 20, 2013), p. 20. ISSN: 0067-0049, 1538-4365. DOI: [10.1088/0067-0049/208/2/20](https://doi.org/10.1088/0067-0049/208/2/20). arXiv: [1212.5225\[astro-ph\]](https://arxiv.org/abs/1212.5225). URL: <http://arxiv.org/abs/1212.5225> (cit. on p. 1).
- [2] Planck Collaboration et al. “Planck 2013 results. I. Overview of products and scientific results”. In: *Astronomy & Astrophysics* 571 (Nov. 2014), A1. ISSN: 0004-6361, 1432-0746. DOI: [10.1051/0004-6361/201321529](https://doi.org/10.1051/0004-6361/201321529). arXiv: [1303.5062\[astro-ph\]](https://arxiv.org/abs/1303.5062). URL: <http://arxiv.org/abs/1303.5062> (cit. on p. 1).
- [3] T. Abe et al. *Belle II Technical Design Report*. Nov. 1, 2010. DOI: [10.48550/arXiv.1011.0352](https://doi.org/10.48550/arXiv.1011.0352). arXiv: [1011.0352\[hep-ex, physics:physics\]](https://arxiv.org/abs/1011.0352). URL: <http://arxiv.org/abs/1011.0352> (cit. on pp. 1, 5).
- [4] P. J. E. Peebles and Bharat Ratra. “The Cosmological Constant and Dark Energy”. In: *Reviews of Modern Physics* 75.2 (Apr. 22, 2003), pp. 559–606. ISSN: 0034-6861, 1539-0756. DOI: [10.1103/RevModPhys.75.559](https://doi.org/10.1103/RevModPhys.75.559). arXiv: [astro-ph/0207347](https://arxiv.org/abs/astro-ph/0207347). URL: <http://arxiv.org/abs/astro-ph/0207347> (cit. on p. 1).
- [5] A. D. Sakharov. “Violation of CP Invariance, C asymmetry, and baryon asymmetry of the universe”. In: *Pisma Zh. Eksp. Teor. Fiz.* 5 (1967), pp. 32–35. DOI: [10.1070/PU1991v034n05ABEH002497](https://doi.org/10.1070/PU1991v034n05ABEH002497) (cit. on p. 1).
- [6] Hosein Hashemi et al. “Pixel Detector Background Generation using Generative Adversarial Networks at Belle II”. In: *EPJ Web of Conferences* 251 (2021). Publisher: EDP Sciences, p. 03031. ISSN: 2100-014X. DOI: [10.1051/epjconf/202125103031](https://doi.org/10.1051/epjconf/202125103031). URL: https://www.epj-conferences.org/articles/epjconf/abs/2021/05/epjconf_chep2021_03031/epjconf_chep2021_03031.html (cit. on pp. 3, 77, 90).
- [7] Hosein Hashemi et al. *Ultra-High-Resolution Detector Simulation with Intra-Event Aware GAN and Self-Supervised Relational Reasoning*. Mar. 7, 2023. DOI: [10.48550/arXiv.2303.08046](https://doi.org/10.48550/arXiv.2303.08046). arXiv: [2303.08046\[hep-ph, physics:physics\]](https://arxiv.org/abs/2303.08046). URL: <http://arxiv.org/abs/2303.08046> (cit. on pp. 3, 31, 66, 80).

[8] Thomas Kuhr. “Computing at Belle II”. In: *Journal of Physics: Conference Series* 331.7 (Dec. 2011), p. 072021. ISSN: 1742-6596. DOI: [10.1088/1742-6596/331/7/072021](https://doi.org/10.1088/1742-6596/331/7/072021). URL: <https://dx.doi.org/10.1088/1742-6596/331/7/072021> (cit. on pp. 3, 8, 88, 159).

[9] *SuperKEKB Phase 3 (Belle II Physics Run) Starts*. News. Mar. 11, 2019. URL: <https://www.kek.jp/en/newsroom/2019/03/11/1600/> (cit. on p. 5).

[10] *Belle II Software Documentation — basf2 development documentation*. URL: <https://software.belle2.org/development/sphinx/index.html> (cit. on pp. 5, 8).

[11] Georgios Giakoustidis et al. “Status of the BELLE II Pixel Detector”. In: *Proceedings of 10th International Workshop on Semiconductor Pixel Detectors for Particles and Imaging — PoS(Pixel2022)*. Vol. 420. Conference Name: 10th International Workshop on Semiconductor Pixel Detectors for Particles and Imaging. SISSA Medialab, May 8, 2023, p. 005. DOI: [10.22323/1.420.0005](https://pos.sissa.it/420/005). URL: <https://pos.sissa.it/420/005> (cit. on pp. 7, 12).

[12] Belle Collaboration and K. Abe. “Observation of Mixing-induced CP Violation in the Neutral B Meson System”. In: *Physical Review D* 66.3 (Aug. 29, 2002), p. 032007. ISSN: 0556-2821, 1089-4918. DOI: [10.1103/PhysRevD.66.032007](https://doi.org/10.1103/PhysRevD.66.032007). arXiv: [hep-ex/0202027](https://arxiv.org/abs/hep-ex/0202027). URL: [http://arxiv.org/abs/hep-ex/0202027](https://arxiv.org/abs/hep-ex/0202027) (cit. on p. 7).

[13] Makoto Kobayashi and Toshihide Maskawa. “CP-Violation in the Renormalizable Theory of Weak Interaction”. In: *Progress of Theoretical Physics* 49.2 (Feb. 1, 1973), pp. 652–657. ISSN: 0033-068X. DOI: [10.1143/PTP.49.652](https://doi.org/10.1143/PTP.49.652). URL: <https://doi.org/10.1143/PTP.49.652> (cit. on p. 7).

[14] *The Nobel Prize in Physics 2008*. NobelPrize.org. URL: <https://www.nobelprize.org/prizes/physics/2008/popular-information/> (cit. on p. 7).

[15] E. Kou et al. “The Belle II Physics Book”. In: *Progress of Theoretical and Experimental Physics* 2019.12 (Dec. 1, 2019), p. 123C01. ISSN: 2050-3911. DOI: [10.1093/ptep/ptz106](https://doi.org/10.1093/ptep/ptz106). arXiv: [1808.10567](https://arxiv.org/abs/1808.10567) [hep-ex, physics:hep-lat, physics:hep-ph]. URL: [http://arxiv.org/abs/1808.10567](https://arxiv.org/abs/1808.10567) (cit. on pp. 8, 9, 14).

[16] T. Kuhr et al. “The Belle II Core Software”. In: *Computing and Software for Big Science* 3.1 (Dec. 2019), p. 1. ISSN: 2510-2036, 2510-2044. DOI: [10.1007/s41781-018-0017-9](https://doi.org/10.1007/s41781-018-0017-9). arXiv: [1809.04299](https://arxiv.org/abs/1809.04299) [hep-ex, physics:physics]. URL: [http://arxiv.org/abs/1809.04299](https://arxiv.org/abs/1809.04299) (cit. on pp. 8, 91, 93).

[17] 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 (July 1, 2003), pp. 250–303. ISSN: 0168-9002. DOI: [10.1016/S0168-9002\(03\)01368-8](https://doi.org/10.1016/S0168-9002(03)01368-8). URL: <https://www.sciencedirect.com/science/article/pii/S0168900203013688> (cit. on pp. 10, 89).

[18] Andreas Moll. “Comprehensive study of the background for the Pixel Vertex Detector at Belle II”. PhD thesis. Ludwig-Maximilians-Universität München, July 31, 2015. URL: <https://edoc.ub.uni-muenchen.de/19106/> (cit. on pp. 13, 18).

[19] A. Natochii et al. “Measured and projected beam backgrounds in the Belle II experiment at the SuperKEKB collider”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1055 (Oct. 1, 2023), p. 168550. ISSN: 0168-9002. DOI: [10.1016/j.nima.2023.168550](https://doi.org/10.1016/j.nima.2023.168550). URL: <https://www.sciencedirect.com/science/article/pii/S0168900223005405> (cit. on p. 14).

[20] *Accelerator design at SuperKEKB / Progress of Theoretical and Experimental Physics / Oxford Academic*. URL: <https://academic.oup.com/ptep/article/2013/3/03A011/1556583> (cit. on p. 16).

[21] *SAD Home Page – Strategic Accelerator Design*. URL: <https://acc-physics.kek.jp/SAD/> (cit. on p. 17).

[22] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. Jan. 12, 2018. DOI: [10.48550/arXiv.1706.08500](https://doi.org/10.48550/arXiv.1706.08500). arXiv: [1706.08500\[cs, stat\]](https://arxiv.org/abs/1706.08500). URL: [http://arxiv.org/abs/1706.08500](https://arxiv.org/abs/1706.08500) (cit. on pp. 21, 89, 102, 138, 178).

[23] Mikołaj Bińkowski et al. *Demystifying MMD GANs*. Jan. 14, 2021. DOI: [10.48550/arXiv.1801.01401](https://doi.org/10.48550/arXiv.1801.01401). arXiv: [1801.01401\[cs, stat\]](https://arxiv.org/abs/1801.01401). URL: [http://arxiv.org/abs/1801.01401](https://arxiv.org/abs/1801.01401) (cit. on pp. 21, 89, 138, 178).

[24] Dan Friedman and Adji Bousso Dieng. *The Vendi Score: A Diversity Evaluation Metric for Machine Learning*. July 2, 2023. DOI: [10.48550/arXiv.2210.02410](https://doi.org/10.48550/arXiv.2210.02410). arXiv: [2210.02410\[cond-mat, stat\]](https://arxiv.org/abs/2210.02410). URL: [http://arxiv.org/abs/2210.02410](https://arxiv.org/abs/2210.02410) (cit. on pp. 21, 66, 138, 141, 148, 180).

[25] J.-F. Krohn et al. “Global Decay Chain Vertex Fitting at B-Factories”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 976 (Oct. 2020), p. 164269. ISSN: 01689002. DOI: [10.1016/j.nima.2020.164269](https://doi.org/10.1016/j.nima.2020.164269). arXiv: [1901.11198\[hep-ex\]](https://arxiv.org/abs/1901.11198). URL: [http://arxiv.org/abs/1901.11198](https://arxiv.org/abs/1901.11198) (cit. on p. 23).

[26] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. *Neural Machine Translation by Jointly Learning to Align and Translate*. arXiv.org. Sept. 1, 2014. URL: <https://arxiv.org/abs/1409.0473v7> (cit. on p. 26).

[27] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. “Non-Local Means Denoising”. In: *Image Processing On Line* 1 (Sept. 13, 2011), pp. 208–212. ISSN: 2105-1232. DOI: [10.5201/ipol.2011bcm_nlm](https://doi.org/10.5201/ipol.2011bcm_nlm). URL: https://www.ipol.im/pub/art/2011/bcm_nlm/?utm_source=doi (cit. on p. 26).

[28] Thang Luong, Hieu Pham, and Christopher D. Manning. “Effective Approaches to Attention-based Neural Machine Translation”. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. EMNLP 2015. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 1412–1421. DOI: [10.18653/v1/D15-1166](https://doi.org/10.18653/v1/D15-1166). URL: <https://aclanthology.org/D15-1166> (cit. on pp. 26, 27).

[29] Ashish Vaswani et al. *Attention Is All You Need*. Dec. 5, 2017. DOI: [10.48550/arXiv.1706.03762](https://doi.org/10.48550/arXiv.1706.03762). arXiv: [1706.03762\[cs\]](https://arxiv.org/abs/1706.03762). URL: <http://arxiv.org/abs/1706.03762> (cit. on pp. 26–28, 65, 81, 83, 84, 113, 124).

[30] Alessandro Sordoni et al. *Iterative Alternating Neural Attention for Machine Reading*. arXiv.org. June 7, 2016. URL: <https://arxiv.org/abs/1606.02245v4> (cit. on p. 26).

[31] Dehong Ma et al. *Interactive Attention Networks for Aspect-Level Sentiment Classification*. arXiv.org. Sept. 4, 2017. URL: <https://arxiv.org/abs/1709.00893v1> (cit. on p. 26).

[32] Alex Graves, Greg Wayne, and Ivo Danihelka. *Neural Turing Machines*. arXiv.org. Oct. 20, 2014. URL: <https://arxiv.org/abs/1410.5401v2> (cit. on p. 26).

[33] Kelvin Xu et al. “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention”. In: *Proceedings of the 32nd International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 1938-7228. PMLR, June 1, 2015, pp. 2048–2057. URL: <https://proceedings.mlr.press/v37/xuc15.html> (cit. on p. 27).

[34] Kaiming He et al. *Deep Residual Learning for Image Recognition*. Dec. 10, 2015. DOI: [10.48550/arXiv.1512.03385](https://doi.org/10.48550/arXiv.1512.03385). arXiv: [1512.03385\[cs\]](https://arxiv.org/abs/1512.03385). URL: <http://arxiv.org/abs/1512.03385> (cit. on pp. 28, 43, 124).

[35] [1607.06450] *Layer Normalization*. URL: <https://arxiv.org/abs/1607.06450> (cit. on pp. 28, 84).

[36] Aston Zhang et al. *Dive into Deep Learning*. arXiv.org. June 21, 2021. URL: <https://arxiv.org/abs/2106.11342v5> (cit. on p. 29).

[37] David E. Rumelhart and James L. McClelland. “Learning Internal Representations by Error Propagation”. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations*. Conference Name: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. MIT Press, 1987, pp. 318–362. ISBN: 978-0-262-29140-8. URL: <https://ieeexplore.ieee.org/document/6302929> (cit. on p. 29).

[38] *Long Short-Term Memory / Neural Computation*. URL: <https://dl.acm.org/doi/10.1162/neco.1997.9.8.1735> (cit. on p. 29).

[39] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. May 24, 2019. DOI: [10.48550/arXiv.1810.04805](https://doi.org/10.48550/arXiv.1810.04805). arXiv: [1810.04805\[cs\]](https://arxiv.org/abs/1810.04805). URL: <http://arxiv.org/abs/1810.04805> (cit. on pp. 30, 81, 83, 85).

[40] *Improving language understanding with unsupervised learning*. URL: <https://openai.com/research/language-unsupervised> (cit. on p. 30).

[41] Peter W. Battaglia et al. *Relational inductive biases, deep learning, and graph networks*. Oct. 17, 2018. DOI: [10.48550/arXiv.1806.01261](https://doi.org/10.48550/arXiv.1806.01261). arXiv: [1806.01261](https://arxiv.org/abs/1806.01261) [cs, stat]. URL: <http://arxiv.org/abs/1806.01261> (cit. on pp. 30, 66, 81, 83).

[42] Zonghan Wu et al. “A Comprehensive Survey on Graph Neural Networks”. In: *IEEE Transactions on Neural Networks and Learning Systems* 32.1 (Jan. 2021), pp. 4–24. ISSN: 2162-237X, 2162-2388. DOI: [10.1109/TNNLS.2020.2978386](https://doi.org/10.1109/TNNLS.2020.2978386). arXiv: [1901.00596](https://arxiv.org/abs/1901.00596) [cs, stat]. URL: <http://arxiv.org/abs/1901.00596> (cit. on p. 30).

[43] Ailing Zeng et al. *Are Transformers Effective for Time Series Forecasting?* Aug. 17, 2022. DOI: [10.48550/arXiv.2205.13504](https://doi.org/10.48550/arXiv.2205.13504). arXiv: [2205.13504](https://arxiv.org/abs/2205.13504) [cs]. URL: <http://arxiv.org/abs/2205.13504> (cit. on p. 31).

[44] Angela Fan et al. *Addressing Some Limitations of Transformers with Feedback Memory*. Jan. 25, 2021. DOI: [10.48550/arXiv.2002.09402](https://doi.org/10.48550/arXiv.2002.09402). arXiv: [2002.09402](https://arxiv.org/abs/2002.09402) [cs, stat]. URL: <http://arxiv.org/abs/2002.09402> (cit. on p. 31).

[45] Ali Hassani et al. *Escaping the Big Data Paradigm with Compact Transformers*. version: 4. June 7, 2022. DOI: [10.48550/arXiv.2104.05704](https://doi.org/10.48550/arXiv.2104.05704). arXiv: [2104.05704](https://arxiv.org/abs/2104.05704) [cs]. URL: <http://arxiv.org/abs/2104.05704> (cit. on p. 31).

[46] Julong Young et al. *Dateformer: Time-modeling Transformer for Longer-term Series Forecasting*. Feb. 21, 2023. DOI: [10.48550/arXiv.2207.05397](https://doi.org/10.48550/arXiv.2207.05397). arXiv: [2207.05397](https://arxiv.org/abs/2207.05397) [cs]. URL: <http://arxiv.org/abs/2207.05397> (cit. on p. 31).

[47] Xuanqing Liu et al. *Learning to Encode Position for Transformer with Continuous Dynamical Model*. version: 1. Mar. 12, 2020. DOI: [10.48550/arXiv.2003.09229](https://doi.org/10.48550/arXiv.2003.09229). arXiv: [2003.09229](https://arxiv.org/abs/2003.09229) [cs, stat]. URL: <http://arxiv.org/abs/2003.09229> (cit. on pp. 31, 87).

[48] Joyce Zheng, Mehdi Rezagholizadeh, and Peyman Passban. *Dynamic Position Encoding for Transformers*. version: 2. Oct. 22, 2022. DOI: [10.48550/arXiv.2204.08142](https://doi.org/10.48550/arXiv.2204.08142). arXiv: [2204.08142](https://arxiv.org/abs/2204.08142) [cs]. URL: <http://arxiv.org/abs/2204.08142> (cit. on p. 31).

[49] Peijian Ding et al. *Reviving Shift Equivariance in Vision Transformers*. version: 1. June 12, 2023. DOI: [10.48550/arXiv.2306.07470](https://doi.org/10.48550/arXiv.2306.07470). arXiv: [2306.07470](https://arxiv.org/abs/2306.07470) [cs]. URL: <http://arxiv.org/abs/2306.07470> (cit. on p. 31).

[50] *Probabilistic machine learning and artificial intelligence / Nature*. URL: <https://www.nature.com/articles/nature14541> (cit. on p. 32).

[51] J.A. Lasserre, C.M. Bishop, and T.P. Minka. “Principled Hybrids of Generative and Discriminative Models”. In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06). Vol. 1. ISSN: 1063-6919. June 2006, pp. 87–94. DOI: [10.1109/CVPR.2006.227](https://doi.org/10.1109/CVPR.2006.227) (cit. on p. 32).

[52] Durk P Kingma et al. “Improved Variational Inference with Inverse Autoregressive Flow”. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016. URL: https://papers.nips.cc/paper_files/paper/2016/hash/ddeebdeefdb7e7e7a697e1c3e3d8ef54-Abstract.html (cit. on p. 32).

[53] Danilo Jimenez Rezende and Shakir Mohamed. *Variational Inference with Normalizing Flows*. June 14, 2016. DOI: [10.48550/arXiv.1505.05770](https://doi.org/10.48550/arXiv.1505.05770). arXiv: [1505.05770](https://arxiv.org/abs/1505.05770) [cs, stat]. URL: [http://arxiv.org/abs/1505.05770](https://arxiv.org/abs/1505.05770) (cit. on pp. 32, 64).

[54] Ian J. Goodfellow et al. *Generative Adversarial Networks*. June 10, 2014. DOI: [10.48550/arXiv.1406.2661](https://doi.org/10.48550/arXiv.1406.2661). arXiv: [1406.2661](https://arxiv.org/abs/1406.2661) [cs, stat]. URL: [http://arxiv.org/abs/1406.2661](https://arxiv.org/abs/1406.2661) (cit. on pp. 32, 38, 81).

[55] Poornima Ramesh et al. *GATSBI: Generative Adversarial Training for Simulation-Based Inference*. Mar. 12, 2022. DOI: [10.48550/arXiv.2203.06481](https://doi.org/10.48550/arXiv.2203.06481). arXiv: [2203.06481](https://arxiv.org/abs/2203.06481) [cs, stat]. URL: [http://arxiv.org/abs/2203.06481](https://arxiv.org/abs/2203.06481) (cit. on p. 33).

[56] *An Introduction to Variational Methods for Graphical Models / SpringerLink*. URL: <https://link.springer.com/article/10.1023/A:1007665907178> (cit. on p. 33).

[57] G. E. Hinton and R. R. Salakhutdinov. “Reducing the Dimensionality of Data with Neural Networks”. In: *Science* 313.5786 (July 28, 2006). Publisher: American Association for the Advancement of Science, pp. 504–507. DOI: [10.1126/science.127647](https://doi.org/10.1126/science.127647). URL: <https://www.science.org/doi/10.1126/science.127647> (cit. on p. 34).

[58] *Random variate generation in one line of code / Proceedings of the 28th conference on Winter simulation*. URL: <https://dl.acm.org/doi/10.1145/256562.256623> (cit. on p. 35).

[59] Diederik P. Kingma and Max Welling. *Auto-Encoding Variational Bayes*. Dec. 10, 2022. DOI: [10.48550/arXiv.1312.6114](https://doi.org/10.48550/arXiv.1312.6114). arXiv: [1312.6114](https://arxiv.org/abs/1312.6114) [cs, stat]. URL: [http://arxiv.org/abs/1312.6114](https://arxiv.org/abs/1312.6114) (cit. on p. 35).

[60] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. *Stochastic Backpropagation and Approximate Inference in Deep Generative Models*. May 30, 2014. DOI: [10.48550/arXiv.1401.4082](https://doi.org/10.48550/arXiv.1401.4082). arXiv: [1401.4082](https://arxiv.org/abs/1401.4082) [cs, stat]. URL: [http://arxiv.org/abs/1401.4082](https://arxiv.org/abs/1401.4082) (cit. on p. 35).

[61] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. *Importance Weighted Autoencoders*. Nov. 7, 2016. DOI: [10.48550/arXiv.1509.00519](https://doi.org/10.48550/arXiv.1509.00519). arXiv: [1509.00519](https://arxiv.org/abs/1509.00519) [cs, stat]. URL: [http://arxiv.org/abs/1509.00519](https://arxiv.org/abs/1509.00519) (cit. on p. 35).

[62] Radford M. Neal. “Annealed importance sampling”. In: *Statistics and Computing* 11.2 (Apr. 1, 2001), pp. 125–139. ISSN: 1573-1375. DOI: [10.1023/A:1008923215028](https://doi.org/10.1023/A:1008923215028). URL: <https://doi.org/10.1023/A:1008923215028> (cit. on p. 36).

[63] Guodong Zhang et al. “Differentiable Annealed Importance Sampling and the Perils of Gradient Noise”. In: *Advances in Neural Information Processing Systems*. Nov. 9, 2021. URL: <https://openreview.net/forum?id=6rqjgrL7Lq> (cit. on p. 36).

[64] Roger B. Grosse, Zoubin Ghahramani, and Ryan P. Adams. *Sandwiching the marginal likelihood using bidirectional Monte Carlo*. Nov. 8, 2015. DOI: [10.48550/arXiv.1511.02543](https://doi.org/10.48550/arXiv.1511.02543). arXiv: [1511.02543\[cs, stat\]](https://arxiv.org/abs/1511.02543). URL: [http://arxiv.org/abs/1511.02543](https://arxiv.org/abs/1511.02543) (cit. on p. 36).

[65] Matthew D. Hoffman. “Learning Deep Latent Gaussian Models with Markov Chain Monte Carlo”. In: *Proceedings of the 34th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, July 17, 2017, pp. 1510–1519. URL: <https://proceedings.mlr.press/v70/hoffman17a.html> (cit. on p. 36).

[66] Yingzhen Li, Richard E. Turner, and Qiang Liu. *Approximate Inference with Amortised MCMC*. May 22, 2017. DOI: [10.48550/arXiv.1702.08343](https://doi.org/10.48550/arXiv.1702.08343). arXiv: [1702.08343\[cs, stat\]](https://arxiv.org/abs/1702.08343). URL: [http://arxiv.org/abs/1702.08343](https://arxiv.org/abs/1702.08343) (cit. on p. 36).

[67] Heiko Zimmermann et al. “Nested Variational Inference”. In: *Advances in Neural Information Processing Systems*. Vol. 34. Curran Associates, Inc., 2021, pp. 20423–20435. URL: <https://proceedings.neurips.cc/paper/2021/hash/ab49b208848abe14418090d95df0d590-Abstract.html> (cit. on p. 37).

[68] Samuel R. Bowman et al. *Generating Sentences from a Continuous Space*. May 12, 2016. DOI: [10.48550/arXiv.1511.06349](https://doi.org/10.48550/arXiv.1511.06349). arXiv: [1511.06349\[cs\]](https://arxiv.org/abs/1511.06349). URL: [http://arxiv.org/abs/1511.06349](https://arxiv.org/abs/1511.06349) (cit. on p. 37).

[69] Danilo Jimenez Rezende and Fabio Viola. *Taming VAEs*. Oct. 1, 2018. DOI: [10.48550/arXiv.1810.00597](https://doi.org/10.48550/arXiv.1810.00597). arXiv: [1810.00597\[cs, stat\]](https://arxiv.org/abs/1810.00597). URL: [http://arxiv.org/abs/1810.00597](https://arxiv.org/abs/1810.00597) (cit. on p. 37).

[70] Jakub M. Tomczak. *Deep Generative Modeling*. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-93157-5 978-3-030-93158-2. DOI: [10.1007/978-3-030-93158-2](https://doi.org/10.1007/978-3-030-93158-2). URL: <https://link.springer.com/10.1007/978-3-030-93158-2> (cit. on p. 38).

[71] Tim Salimans et al. “Improved Techniques for Training GANs”. In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016. URL: https://proceedings.neurips.cc/paper_files/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html (cit. on p. 39).

[72] Jae Hyun Lim and Jong Chul Ye. *Geometric GAN*. May 8, 2017. DOI: [10.48550/arXiv.1705.02894](https://doi.org/10.48550/arXiv.1705.02894). arXiv: [1705.02894\[cond-mat, stat\]](https://arxiv.org/abs/1705.02894). URL: [http://arxiv.org/abs/1705.02894](https://arxiv.org/abs/1705.02894) (cit. on pp. 39, 40).

[73] Thomas Unterthiner et al. *Coulomb GANs: Provably Optimal Nash Equilibria via Potential Fields*. Jan. 30, 2018. arXiv: [1708.08819\[cs, stat\]](https://arxiv.org/abs/1708.08819). URL: [http://arxiv.org/abs/1708.08819](https://arxiv.org/abs/1708.08819) (cit. on p. 39).

[74] Junbo Zhao, Michael Mathieu, and Yann LeCun. *Energy-based Generative Adversarial Network*. Mar. 6, 2017. DOI: [10.48550/arXiv.1609.03126](https://doi.org/10.48550/arXiv.1609.03126). arXiv: [1609.03126](https://arxiv.org/abs/1609.03126) [cs, stat]. URL: <http://arxiv.org/abs/1609.03126> (cit. on p. 39).

[75] Chun-Liang Li et al. *MMD GAN: Towards Deeper Understanding of Moment Matching Network*. Nov. 27, 2017. arXiv: [1705.08584](https://arxiv.org/abs/1705.08584) [cs, stat]. URL: <http://arxiv.org/abs/1705.08584> (cit. on pp. 39, 63).

[76] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. Dec. 6, 2017. DOI: [10.48550/arXiv.1701.07875](https://doi.org/10.48550/arXiv.1701.07875). arXiv: [1701.07875](https://arxiv.org/abs/1701.07875) [cs, stat]. URL: <http://arxiv.org/abs/1701.07875> (cit. on pp. 39, 62).

[77] Minguk Kang and Jaesik Park. *ContraGAN: Contrastive Learning for Conditional Image Generation*. Feb. 3, 2021. DOI: [10.48550/arXiv.2006.12681](https://doi.org/10.48550/arXiv.2006.12681). arXiv: [2006.12681](https://arxiv.org/abs/2006.12681) [cs]. URL: <http://arxiv.org/abs/2006.12681> (cit. on pp. 39, 44, 45, 76, 78, 81, 89).

[78] Martin Arjovsky and Léon Bottou. *Towards Principled Methods for Training Generative Adversarial Networks*. Jan. 17, 2017. DOI: [10.48550/arXiv.1701.04862](https://doi.org/10.48550/arXiv.1701.04862). arXiv: [1701.04862](https://arxiv.org/abs/1701.04862) [cs, stat]. URL: <http://arxiv.org/abs/1701.04862> (cit. on p. 39).

[79] Xingzhe Su et al. *A Unified GAN Framework Regarding Manifold Alignment for Remote Sensing Images Generation*. May 30, 2023. DOI: [10.48550/arXiv.2305.19507](https://doi.org/10.48550/arXiv.2305.19507). arXiv: [2305.19507](https://arxiv.org/abs/2305.19507) [cs, eess]. URL: <http://arxiv.org/abs/2305.19507> (cit. on p. 39).

[80] Emmanuel Martinez et al. *LD-GAN: Low-Dimensional Generative Adversarial Network for Spectral Image Generation with Variance Regularization*. Apr. 28, 2023. DOI: [10.48550/arXiv.2305.00132](https://doi.org/10.48550/arXiv.2305.00132). arXiv: [2305.00132](https://arxiv.org/abs/2305.00132) [cs, eess]. URL: <http://arxiv.org/abs/2305.00132> (cit. on p. 39).

[81] Cong Geng et al. *Generative Model without Prior Distribution Matching*. Sept. 23, 2020. DOI: [10.48550/arXiv.2009.11016](https://doi.org/10.48550/arXiv.2009.11016). arXiv: [2009.11016](https://arxiv.org/abs/2009.11016) [cs]. URL: <http://arxiv.org/abs/2009.11016> (cit. on p. 40).

[82] Andrew Brock, Jeff Donahue, and Karen Simonyan. *Large Scale GAN Training for High Fidelity Natural Image Synthesis*. Feb. 25, 2019. DOI: [10.48550/arXiv.1809.11096](https://doi.org/10.48550/arXiv.1809.11096). arXiv: [1809.11096](https://arxiv.org/abs/1809.11096) [cs, stat]. URL: <http://arxiv.org/abs/1809.11096> (cit. on pp. 40, 44, 77, 79, 89).

[83] Mehdi Mirza and Simon Osindero. *Conditional Generative Adversarial Nets*. Nov. 6, 2014. DOI: [10.48550/arXiv.1411.1784](https://doi.org/10.48550/arXiv.1411.1784). arXiv: [1411.1784](https://arxiv.org/abs/1411.1784) [cs, stat]. URL: <http://arxiv.org/abs/1411.1784> (cit. on pp. 40, 63, 85).

[84] Augustus Odena, Christopher Olah, and Jonathon Shlens. *Conditional Image Synthesis With Auxiliary Classifier GANs*. July 20, 2017. DOI: [10.48550/arXiv.1610.09585](https://doi.org/10.48550/arXiv.1610.09585). arXiv: [1610.09585](https://arxiv.org/abs/1610.09585) [cs, stat]. URL: <http://arxiv.org/abs/1610.09585> (cit. on pp. 40, 62).

[85] Takeru Miyato and Masanori Koyama. *cGANs with Projection Discriminator*. Aug. 14, 2018. DOI: [10.48550/arXiv.1802.05637](https://doi.org/10.48550/arXiv.1802.05637). arXiv: [1802.05637](https://arxiv.org/abs/1802.05637) [cs, stat]. URL: <http://arxiv.org/abs/1802.05637> (cit. on pp. 40, 78).

[86] Harm de Vries et al. *Modulating early visual processing by language*. version: 3. Dec. 18, 2017. DOI: [10.48550/arXiv.1707.00683](https://doi.org/10.48550/arXiv.1707.00683). arXiv: [1707.00683](https://arxiv.org/abs/1707.00683) [cs]. URL: <http://arxiv.org/abs/1707.00683> (cit. on p. 41).

[87] Takeru Miyato et al. *Spectral Normalization for Generative Adversarial Networks*. Feb. 16, 2018. DOI: [10.48550/arXiv.1802.05957](https://doi.org/10.48550/arXiv.1802.05957). arXiv: [1802.05957](https://arxiv.org/abs/1802.05957) [cs, stat]. URL: <http://arxiv.org/abs/1802.05957> (cit. on pp. 41, 43, 76, 84).

[88] Wei Hu, Lechao Xiao, and Jeffrey Pennington. *Provable Benefit of Orthogonal Initialization in Optimizing Deep Linear Networks*. version: 1. Jan. 16, 2020. arXiv: [2001.05992](https://arxiv.org/abs/2001.05992) [cs, math, stat]. URL: <http://arxiv.org/abs/2001.05992> (cit. on pp. 41, 89).

[89] Andrew Brock et al. *Neural Photo Editing with Introspective Adversarial Networks*. Feb. 6, 2017. DOI: [10.48550/arXiv.1609.07093](https://doi.org/10.48550/arXiv.1609.07093). arXiv: [1609.07093](https://arxiv.org/abs/1609.07093) [cs, stat]. URL: <http://arxiv.org/abs/1609.07093> (cit. on pp. 42, 79).

[90] Han Zhang et al. *Self-Attention Generative Adversarial Networks*. June 14, 2019. DOI: [10.48550/arXiv.1805.08318](https://doi.org/10.48550/arXiv.1805.08318). arXiv: [1805.08318](https://arxiv.org/abs/1805.08318) [cs, stat]. URL: <http://arxiv.org/abs/1805.08318> (cit. on pp. 42, 43, 76, 78, 79, 81).

[91] Ting Chen et al. *A Simple Framework for Contrastive Learning of Visual Representations*. June 30, 2020. DOI: [10.48550/arXiv.2002.05709](https://doi.org/10.48550/arXiv.2002.05709). arXiv: [2002.05709](https://arxiv.org/abs/2002.05709) [cs, stat]. URL: <http://arxiv.org/abs/2002.05709> (cit. on pp. 45, 51, 132).

[92] Minguk Kang et al. *Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training*. Nov. 1, 2021. DOI: [10.48550/arXiv.2111.01118](https://doi.org/10.48550/arXiv.2111.01118). arXiv: [2111.01118](https://arxiv.org/abs/2111.01118) [cs]. URL: <http://arxiv.org/abs/2111.01118> (cit. on pp. 45, 76, 81).

[93] Harsh Rangwani, Konda Reddy Mopuri, and R. Venkatesh Babu. *Class Balancing GAN with a Classifier in the Loop*. June 17, 2021. DOI: [10.48550/arXiv.2106.09402](https://doi.org/10.48550/arXiv.2106.09402). arXiv: [2106.09402](https://arxiv.org/abs/2106.09402) [cs]. URL: <http://arxiv.org/abs/2106.09402> (cit. on pp. 45, 76, 81).

[94] Clement Vignac and Pascal Frossard. *Top-N: Equivariant set and graph generation without exchangeability*. Apr. 1, 2022. DOI: [10.48550/arXiv.2110.02096](https://doi.org/10.48550/arXiv.2110.02096). arXiv: [2110.02096](https://arxiv.org/abs/2110.02096) [cs, stat]. URL: <http://arxiv.org/abs/2110.02096> (cit. on pp. 46, 110–112, 145).

[95] Haoqiang Fan, Hao Su, and Leonidas Guibas. *A Point Set Generation Network for 3D Object Reconstruction from a Single Image*. Dec. 6, 2016. arXiv: [1612.00603](https://arxiv.org/abs/1612.00603) [cs]. URL: <http://arxiv.org/abs/1612.00603> (cit. on pp. 47, 62).

[96] *The Hungarian method for the assignment problem - Kuhn - 1955 - Naval Research Logistics Quarterly - Wiley Online Library*. URL: <https://onlinelibrary.wiley.com/doi/10.1002/nav.3800020109> (cit. on p. 47).

[97] *Assignment problem*. In: *Wikipedia*. Page Version ID: 1159259304. June 9, 2023. URL: https://en.wikipedia.org/w/index.php?title=Assignment_problem&oldid=1159259304 (cit. on p. 48).

[98] N. Tomizawa. "On some techniques useful for solution of transportation network problems". In: *Networks* 1.2 (1971). _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/net.3230010206> pp. 173–194. ISSN: 1097-0037. DOI: [10.1002/net.3230010206](https://doi.org/10.1002/net.3230010206). URL: <https://onlinelibrary.wiley.com/doi/abs/10.1002/net.3230010206> (cit. on p. 48).

[99] *Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems / Journal of the ACM*. URL: <https://dl.acm.org/doi/10.1145/321694.321699> (cit. on p. 48).

[100] *Self-supervised learning: The dark matter of intelligence*. URL: <https://ai.facebook.com/blog/self-supervised-learning-the-dark-matter-of-intelligence/> (cit. on p. 48).

[101] Phuc H. Le-Khac, Graham Healy, and Alan F. Smeaton. "Contrastive Representation Learning: A Framework and Review". In: *IEEE Access* 8 (2020), pp. 193907–193934. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2020.3031549](https://doi.org/10.1109/ACCESS.2020.3031549). arXiv: [2010.05113\[cs,stat\]](https://arxiv.org/abs/2010.05113). URL: [http://arxiv.org/abs/2010.05113](https://arxiv.org/abs/2010.05113) (cit. on p. 49).

[102] Jane Bromley et al. "Signature Verification using a "Siamese" Time Delay Neural Network". In: () (cit. on p. 50).

[103] *Learning to simulate high energy particle collisions from unlabeled data / Scientific Reports*. URL: <https://www.nature.com/articles/s41598-022-10966-7> (cit. on pp. 50, 108).

[104] Kilian Q Weinberger, John Blitzer, and Lawrence Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification". In: *Advances in Neural Information Processing Systems*. Vol. 18. MIT Press, 2005. URL: <https://proceedings.neurips.cc/paper/2005/hash/a7f592cef8b130a6967a90617db5681b-Abstract.html> (cit. on p. 50).

[105] Gal Chechik et al. "Large Scale Online Learning of Image Similarity Through Ranking". In: *Journal of Machine Learning Research* 11.36 (2010), pp. 1109–1135. ISSN: 1533-7928. URL: <http://jmlr.org/papers/v11/chechik10a.html> (cit. on p. 50).

[106] Kihyuk Sohn. "Improved Deep Metric Learning with Multi-class N-pair Loss Objective". In: *Advances in Neural Information Processing Systems*. Vol. 29. Curran Associates, Inc., 2016. URL: https://papers.nips.cc/paper_files/paper/2016/hash/6b180037abbebea991d8b1232f8a8ca9-Abstract.html (cit. on p. 51).

[107] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. *Representation Learning with Contrastive Predictive Coding*. Jan. 22, 2019. DOI: [10.48550/arXiv.1807.03748](https://doi.org/10.48550/arXiv.1807.03748). arXiv: [1807.03748\[cs,stat\]](https://arxiv.org/abs/1807.03748). URL: [http://arxiv.org/abs/1807.03748](https://arxiv.org/abs/1807.03748) (cit. on p. 51).

[108] Li Jing et al. *Understanding Dimensional Collapse in Contrastive Self-supervised Learning*. Apr. 23, 2022. DOI: [10.48550/arXiv.2110.09348](https://doi.org/10.48550/arXiv.2110.09348). arXiv: [2110.09348\[cs\]](https://arxiv.org/abs/2110.09348). URL: <http://arxiv.org/abs/2110.09348> (cit. on p. 51).

[109] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. *Distilling the Knowledge in a Neural Network*. Mar. 9, 2015. DOI: [10.48550/arXiv.1503.02531](https://doi.org/10.48550/arXiv.1503.02531). arXiv: [1503.02531\[cs, stat\]](https://arxiv.org/abs/1503.02531). URL: <http://arxiv.org/abs/1503.02531> (cit. on p. 52).

[110] Tommaso Furlanello et al. *Born Again Neural Networks*. June 29, 2018. DOI: [10.48550/arXiv.1805.04770](https://doi.org/10.48550/arXiv.1805.04770). arXiv: [1805.04770\[cs, stat\]](https://arxiv.org/abs/1805.04770). URL: <http://arxiv.org/abs/1805.04770> (cit. on p. 52).

[111] Adriana Romero et al. “FitNets: Hints for Thin Deep Nets”. In: *CoRR* (Dec. 19, 2014). URL: <https://www.semanticscholar.org/paper/FitNets%3A-Hints-for-Thin-Deep-Nets-Romero-Ballas/cd85a549add0c7c7def36aca29837efd24b24080> (cit. on p. 52).

[112] Junho Yim et al. “A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). ISSN: 1063-6919. July 2017, pp. 7130–7138. DOI: [10.1109/CVPR.2017.754](https://doi.org/10.1109/CVPR.2017.754) (cit. on p. 53).

[113] Jean-Bastien Grill et al. *Bootstrap your own latent: A new approach to self-supervised Learning*. Sept. 10, 2020. DOI: [10.48550/arXiv.2006.07733](https://doi.org/10.48550/arXiv.2006.07733). arXiv: [2006.07733\[cs, stat\]](https://arxiv.org/abs/2006.07733). URL: <http://arxiv.org/abs/2006.07733> (cit. on pp. 53, 54).

[114] Xinlei Chen and Kaiming He. *Exploring Simple Siamese Representation Learning*. Nov. 20, 2020. DOI: [10.48550/arXiv.2011.10566](https://doi.org/10.48550/arXiv.2011.10566). arXiv: [2011.10566\[cs\]](https://arxiv.org/abs/2011.10566). URL: <http://arxiv.org/abs/2011.10566> (cit. on pp. 53–55, 131).

[115] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. May 24, 2021. DOI: [10.48550/arXiv.2104.14294](https://doi.org/10.48550/arXiv.2104.14294). arXiv: [2104.14294\[cs\]](https://arxiv.org/abs/2104.14294). URL: <http://arxiv.org/abs/2104.14294> (cit. on p. 53).

[116] *Understanding self-supervised and contrastive learning with "Bootstrap Your Own Latent" (BYOL)*. URL: <https://generallyintelligent.com/research/2020-08-24-understanding-self-supervised-contrastive-learning/> (cit. on p. 54).

[117] Suhong Moon et al. *An Embedding-Dynamic Approach to Self-supervised Learning*. July 7, 2022. arXiv: [2207.03552\[cs\]](https://arxiv.org/abs/2207.03552). URL: <http://arxiv.org/abs/2207.03552> (cit. on p. 54).

[118] Chaoning Zhang et al. *How Does SimSiam Avoid Collapse Without Negative Samples? A Unified Understanding with Self-supervised Contrastive Learning*. Mar. 30, 2022. arXiv: [2203.16262\[cs\]](https://arxiv.org/abs/2203.16262). URL: <http://arxiv.org/abs/2203.16262> (cit. on p. 54).

[119] Geoffrey E. Hinton and J. Sejnowski. “OPTIMAL PERCEPTUAL INFERENCE”. In: 1983. URL: <https://www.semanticscholar.org/paper/OPTIMAL-PERCEPTUAL-INFERENCE-Hinton-Sejnowski/1718965f492d4e9fe1d98a3fb83efe671a4aed2c> (cit. on p. 57).

[120] Geoffrey E. Hinton. “Training products of experts by minimizing contrastive divergence”. In: *Neural Computation* 14.8 (Aug. 2002), pp. 1771–1800. ISSN: 0899-7667. DOI: [10.1162/089976602760128018](https://doi.org/10.1162/089976602760128018) (cit. on p. 57).

[121] Geoffrey E. Hinton, Simon Osindero, and Yee-Whye Teh. “A fast learning algorithm for deep belief nets”. In: *Neural Computation* 18.7 (July 2006), pp. 1527–1554. ISSN: 0899-7667. DOI: [10.1162/neco.2006.18.7.1527](https://doi.org/10.1162/neco.2006.18.7.1527) (cit. on p. 57).

[122] Luke de Oliveira, Michela Paganini, and Benjamin Nachman. “Learning Particle Physics by Example: Location-Aware Generative Adversarial Networks for Physics Synthesis”. In: *Computing and Software for Big Science* 1.1 (Nov. 2017), p. 4. ISSN: 2510-2036, 2510-2044. DOI: [10.1007/s41781-017-0004-6](https://doi.org/10.1007/s41781-017-0004-6). arXiv: [1701.05927](https://arxiv.org/abs/1701.05927) [hep-ex, physics:physics, stat]. URL: <http://arxiv.org/abs/1701.05927> (cit. on pp. 57, 60, 62).

[123] Kunihiko Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 0340-1200, 1432-0770. DOI: [10.1007/BF00344251](https://doi.org/10.1007/BF00344251). URL: <http://link.springer.com/10.1007/BF00344251> (cit. on p. 58).

[124] Petar Veličković. “Everything is Connected: Graph Neural Networks”. In: *Current Opinion in Structural Biology* 79 (Apr. 2023), p. 102538. ISSN: 0959440X. DOI: [10.1016/j.sbi.2023.102538](https://doi.org/10.1016/j.sbi.2023.102538). arXiv: [2301.08210](https://arxiv.org/abs/2301.08210) [cs, stat]. URL: <http://arxiv.org/abs/2301.08210> (cit. on p. 58).

[125] Manzil Zaheer et al. *Deep Sets*. Apr. 14, 2018. DOI: [10.48550/arXiv.1703.06114](https://doi.org/10.48550/arXiv.1703.06114). arXiv: [1703.06114](https://arxiv.org/abs/1703.06114) [cs, stat]. URL: <http://arxiv.org/abs/1703.06114> (cit. on pp. 58, 83, 113).

[126] Roe Goodman and Leslie Kish. “Controlled Selection–A Technique in Probability Sampling”. In: *Journal of the American Statistical Association* 45.251 (1950). Publisher: [American Statistical Association, Taylor & Francis, Ltd.], pp. 350–372. ISSN: 0162-1459. DOI: [10.2307/2280293](https://doi.org/10.2307/2280293). URL: <https://www.jstor.org/stable/2280293> (cit. on p. 58).

[127] Yan Wu et al. *LOGAN: Latent Optimisation for Generative Adversarial Networks*. July 1, 2020. DOI: [10.48550/arXiv.1912.00953](https://doi.org/10.48550/arXiv.1912.00953). arXiv: [1912.00953](https://arxiv.org/abs/1912.00953) [cs, stat]. URL: <http://arxiv.org/abs/1912.00953> (cit. on pp. 60, 78).

[128] Aishik Ghosh and on behalf of the ATLAS Collaboration. “Deep generative models for fast shower simulation in ATLAS”. In: *Journal of Physics: Conference Series* 1525.1 (Apr. 2020). Publisher: IOP Publishing, p. 012077. ISSN: 1742-6596. DOI: [10.1088/1742-6596/1525/1/012077](https://dx.doi.org/10.1088/1742-6596/1525/1/012077). URL: <https://dx.doi.org/10.1088/1742-6596/1525/1/012077> (cit. on p. 61).

[129] Erik Buhmann et al. “Decoding Photons: Physics in the Latent Space of a BIB-AE Generative Network”. In: *EPJ Web of Conferences* 251 (2021), p. 03003. ISSN: 2100-014X. DOI: [10.1051/epjconf/202125103003](https://doi.org/10.1051/epjconf/202125103003). arXiv: [2102.12491](https://arxiv.org/abs/2102.12491) [hep-ex, physics:hep-ph, physics:physics]. URL: [http://arxiv.org/abs/2102.12491](https://arxiv.org/abs/2102.12491) (cit. on p. 61).

[130] Erik Buhmann et al. “Getting High: High Fidelity Simulation of High Granularity Calorimeters with High Speed”. In: *Computing and Software for Big Science* 5.1 (Dec. 2021), p. 13. ISSN: 2510-2036, 2510-2044. DOI: [10.1007/s41781-021-00056-0](https://doi.org/10.1007/s41781-021-00056-0). arXiv: [2005.05334](https://arxiv.org/abs/2005.05334) [hep-ex, physics:hep-ph, physics:physics]. URL: [http://arxiv.org/abs/2005.05334](https://arxiv.org/abs/2005.05334) (cit. on pp. 61, 66, 68).

[131] Slava Voloshynovskiy et al. *Information bottleneck through variational glasses*. Dec. 5, 2019. arXiv: [1912.00830](https://arxiv.org/abs/1912.00830) [cs]. URL: [http://arxiv.org/abs/1912.00830](https://arxiv.org/abs/1912.00830) (cit. on p. 61).

[132] Erik Buhmann et al. *Hadrons, Better, Faster, Stronger*. Dec. 17, 2021. arXiv: [2112.09709](https://arxiv.org/abs/2112.09709) [hep-ex, physics:hep-ph, physics:physics]. URL: [http://arxiv.org/abs/2112.09709](https://arxiv.org/abs/2112.09709) (cit. on p. 61).

[133] Sydney Otten et al. *Event Generation and Statistical Sampling for Physics with Deep Generative Models and a Density Information Buffer*. Feb. 25, 2021. arXiv: [1901.00875](https://arxiv.org/abs/1901.00875) [hep-ex, physics:hep-ph, physics:physics]. URL: [http://arxiv.org/abs/1901.00875](https://arxiv.org/abs/1901.00875) (cit. on pp. 61, 62).

[134] Emanuel Parzen. “On Estimation of a Probability Density Function and Mode”. In: *The Annals of Mathematical Statistics* 33.3 (1962). Publisher: Institute of Mathematical Statistics, pp. 1065–1076. ISSN: 0003-4851. URL: <https://www.jstor.org/stable/2237880> (cit. on p. 61).

[135] Kamil Deja et al. *End-to-end Sinkhorn Autoencoder with Noise Generator*. June 11, 2020. arXiv: [2006.06704](https://arxiv.org/abs/2006.06704) [cs, stat]. URL: [http://arxiv.org/abs/2006.06704](https://arxiv.org/abs/2006.06704) (cit. on p. 61).

[136] Giorgio Patrini et al. *Sinkhorn AutoEncoders*. July 15, 2019. arXiv: [1810.01118](https://arxiv.org/abs/1810.01118) [cs, stat]. URL: [http://arxiv.org/abs/1810.01118](https://arxiv.org/abs/1810.01118) (cit. on p. 61).

[137] Babajide O. Ayinde, Tamer Inanc, and Jacek M. Zurada. “Regularizing Deep Neural Networks by Enhancing Diversity in Feature Extraction”. In: *IEEE Transactions on Neural Networks and Learning Systems* 30.9 (Sept. 2019). Conference Name: IEEE Transactions on Neural Networks and Learning Systems, pp. 2650–2661. ISSN: 2162-2388. DOI: [10.1109/TNNLS.2018.2885972](https://doi.org/10.1109/TNNLS.2018.2885972) (cit. on p. 61).

[138] Cristiano Fanelli and Jary Pomponi. “DeepRICH: Learning Deeply Cherenkov Detectors”. In: *Machine Learning: Science and Technology* 1.1 (Apr. 28, 2020), p. 015010. ISSN: 2632-2153. DOI: [10.1088/2632-2153/ab845a](https://doi.org/10.1088/2632-2153/ab845a). arXiv: [1911.11717](https://arxiv.org/abs/1911.11717) [hep-ex, physics:nucl-ex, physics:physics]. URL: <http://arxiv.org/abs/1911.11717> (cit. on p. 62).

[139] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. “Learning Structured Output Representation using Deep Conditional Generative Models”. In: *Advances in Neural Information Processing Systems*. Vol. 28. Curran Associates, Inc., 2015. URL: https://papers.nips.cc/paper_files/paper/2015/hash/8d55a249e6baa5c06772297520da2051-Abstract.html (cit. on p. 62).

[140] Shengjia Zhao, Jiaming Song, and Stefano Ermon. *InfoVAE: Information Maximizing Variational Autoencoders*. May 30, 2018. DOI: [10.48550/arXiv.1706.02262](https://doi.org/10.48550/arXiv.1706.02262). arXiv: [1706.02262](https://arxiv.org/abs/1706.02262) [cs, stat]. URL: <http://arxiv.org/abs/1706.02262> (cit. on pp. 62, 136).

[141] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. “Practical Bayesian Optimization of Machine Learning Algorithms”. In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012. URL: https://papers.nips.cc/paper_files/paper/2012/hash/05311655a15b75fab86956663e1819cd-Abstract.html (cit. on p. 62).

[142] Ali Hariri, Darya Dyachkova, and Sergei Gleyzer. *Graph Generative Models for Fast Detector Simulations in High Energy Physics*. Aug. 24, 2021. arXiv: [2104.01725](https://arxiv.org/abs/2104.01725) [hep-ex]. URL: <http://arxiv.org/abs/2104.01725> (cit. on pp. 62, 67).

[143] Breno Orzari et al. *Sparse Data Generation for Particle-Based Simulation of Hadronic Jets in the LHC*. Sept. 30, 2021. arXiv: [2109.15197](https://arxiv.org/abs/2109.15197) [hep-ex, physics:physics]. URL: <http://arxiv.org/abs/2109.15197> (cit. on p. 62).

[144] Mary Touranakou et al. “Particle-based Fast Jet Simulation at the LHC with Variational Autoencoders”. In: *Machine Learning: Science and Technology* 3.3 (Sept. 1, 2022), p. 035003. ISSN: 2632-2153. DOI: [10.1088/2632-2153/ac7c56](https://doi.org/10.1088/2632-2153/ac7c56). arXiv: [2203.00520](https://arxiv.org/abs/2203.00520) [hep-ex, physics:hep-ph, physics:physics]. URL: <http://arxiv.org/abs/2203.00520> (cit. on p. 62).

[145] Steven Tsan et al. *Particle Graph Autoencoders and Differentiable, Learned Energy Mover’s Distance*. Nov. 24, 2021. arXiv: [2111.12849](https://arxiv.org/abs/2111.12849) [hep-ex, physics:physics]. URL: <http://arxiv.org/abs/2111.12849> (cit. on pp. 62, 67).

[146] Yue Wang et al. *Dynamic Graph CNN for Learning on Point Clouds*. June 11, 2019. arXiv: [1801.07829](https://arxiv.org/abs/1801.07829) [cs]. URL: <http://arxiv.org/abs/1801.07829> (cit. on p. 62).

[147] Abhishek Abhishek et al. *CaloDVAE : Discrete Variational Autoencoders for Fast Calorimeter Shower Simulation*. Oct. 13, 2022. arXiv: [2210.07430](https://arxiv.org/abs/2210.07430) [hep-ex, physics:physics, stat]. URL: <http://arxiv.org/abs/2210.07430> (cit. on p. 62).

[148] Jason Tyler Rolfe. *Discrete Variational Autoencoders*. Apr. 21, 2017. arXiv: [1609.02200](https://arxiv.org/abs/1609.02200) [cs, stat]. URL: <http://arxiv.org/abs/1609.02200> (cit. on p. 62).

[149] Arash Vahdat et al. *DVAE++: Discrete Variational Autoencoders with Overlapping Transformations*. May 25, 2018. arXiv: [1802.04920\[cs,stat\]](https://arxiv.org/abs/1802.04920). URL: <http://arxiv.org/abs/1802.04920> (cit. on p. 62).

[150] Amir H. Khoshaman and Mohammad H. Amin. *GumBolt: Extending Gumbel trick to Boltzmann priors*. Mar. 1, 2019. arXiv: [1805.07349\[cs,stat\]](https://arxiv.org/abs/1805.07349). URL: <http://arxiv.org/abs/1805.07349> (cit. on p. 62).

[151] Guido Montufar. *Restricted Boltzmann Machines: Introduction and Review*. June 19, 2018. arXiv: [1806.07066\[cs,math,stat\]](https://arxiv.org/abs/1806.07066). URL: <http://arxiv.org/abs/1806.07066> (cit. on p. 62).

[152] Pasquale Musella and Francesco Pandolfi. “Fast and accurate simulation of particle detectors using generative adversarial networks”. In: *Computing and Software for Big Science* 2.1 (Nov. 2018), p. 8. ISSN: 2510-2036, 2510-2044. DOI: [10.1007/s41781-018-0015-y](https://doi.org/10.1007/s41781-018-0015-y). arXiv: [1805.00850\[hep-ex,physics:hep-ph,physics:physics\]](https://arxiv.org/abs/1805.00850). URL: <http://arxiv.org/abs/1805.00850> (cit. on pp. 62, 63).

[153] Jesse C. Cresswell et al. *CaloMan: Fast generation of calorimeter showers with density estimation on learned manifolds*. Nov. 23, 2022. arXiv: [2211.15380\[hep-ex,physics:hep-ph,physics:physics\]](https://arxiv.org/abs/2211.15380). URL: <http://arxiv.org/abs/2211.15380> (cit. on p. 62).

[154] *Representation Learning: A Review and New Perspectives / IEEE Journals & Magazine / IEEE Xplore*. URL: <https://ieeexplore.ieee.org/abstract/document/6472238> (cit. on p. 62).

[155] Johann Brehmer and Kyle Cranmer. *Flows for simultaneous manifold learning and density estimation*. Nov. 13, 2020. arXiv: [2003.13913\[cs,stat\]](https://arxiv.org/abs/2003.13913). URL: <http://arxiv.org/abs/2003.13913> (cit. on p. 62).

[156] Bradley C. A. Brown et al. *Verifying the Union of Manifolds Hypothesis for Image Data*. Mar. 2, 2023. DOI: [10.48550/arXiv.2207.02862](https://doi.org/10.48550/arXiv.2207.02862). arXiv: [2207.02862\[cs,stat\]](https://arxiv.org/abs/2207.02862). URL: <http://arxiv.org/abs/2207.02862> (cit. on p. 62).

[157] Jack H. Collins. *An Exploration of Learnt Representations of W Jets*. Apr. 18, 2022. arXiv: [2109.10919\[hep-ex,physics:hep-ph\]](https://arxiv.org/abs/2109.10919). URL: <http://arxiv.org/abs/2109.10919> (cit. on p. 62).

[158] Jack H. Collins et al. *Machine-Learning Compression for Particle Physics Discoveries*. Dec. 18, 2022. arXiv: [2210.11489\[hep-ex,physics:hep-ph,physics:physics\]](https://arxiv.org/abs/2210.11489). URL: <http://arxiv.org/abs/2210.11489> (cit. on p. 62).

[159] Patrick T. Komiske, Eric M. Metodiev, and Jesse Thaler. “Energy Flow Networks: Deep Sets for Particle Jets”. In: *Journal of High Energy Physics* 2019.1 (Jan. 2019), p. 121. ISSN: 1029-8479. DOI: [10.1007/JHEP01\(2019\)121](https://doi.org/10.1007/JHEP01(2019)121). arXiv: [1810.05165\[hep-ex,physics:hep-ph,stat\]](https://arxiv.org/abs/1810.05165). URL: <http://arxiv.org/abs/1810.05165> (cit. on p. 62).

[160] Irina Higgins et al. “beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework”. In: International Conference on Learning Representations. July 21, 2022. URL: <https://openreview.net/forum?id=Sy2fzU9g1> (cit. on p. 62).

[161] Phil Ilten et al. *Modeling hadronization using machine learning*. Mar. 9, 2022. arXiv: 2203.04983[hep-ph]. URL: <http://arxiv.org/abs/2203.04983> (cit. on p. 62).

[162] Soheil Kolouri et al. *Sliced-Wasserstein Autoencoder: An Embarrassingly Simple Generative Model*. June 26, 2018. arXiv: 1804.01947[cs, stat]. URL: <http://arxiv.org/abs/1804.01947> (cit. on pp. 62, 64).

[163] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. “CaloGAN: Simulating 3D High Energy Particle Showers in Multi-Layer Electromagnetic Calorimeters with Generative Adversarial Networks”. In: *Physical Review D* 97.1 (Jan. 30, 2018), p. 014021. ISSN: 2470-0010, 2470-0029. DOI: 10.1103/PhysRevD.97.014021. arXiv: 1712.10321[hep-ex, physics:hep-ph, stat]. URL: <http://arxiv.org/abs/1712.10321> (cit. on pp. 62, 68).

[164] Luke de Oliveira, Michela Paganini, and Benjamin Nachman. “Controlling Physical Attributes in GAN-Accelerated Simulation of Electromagnetic Calorimeters”. In: *Journal of Physics: Conference Series* 1085 (Sept. 2018), p. 042017. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1085/4/042017. arXiv: 1711.08813[hep-ex, physics:physics]. URL: <http://arxiv.org/abs/1711.08813> (cit. on p. 62).

[165] Michela Paganini, Luke de Oliveira, and Benjamin Nachman. “Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeters”. In: *Physical Review Letters* 120.4 (Jan. 26, 2018), p. 042003. ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.120.042003. arXiv: 1705.02355[hep-ex, physics:hep-ph, stat]. URL: <http://arxiv.org/abs/1705.02355> (cit. on p. 62).

[166] Gul rukh Khattak, Sofia Vallecorsa, and Federico Carminati. “Three Dimensional Energy Parametrized Generative Adversarial Networks for Electromagnetic Shower Simulation”. In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. 2018 25th IEEE International Conference on Image Processing (ICIP). ISSN: 2381-8549. Oct. 2018, pp. 3913–3917. DOI: 10.1109/ICIP.2018.8451587 (cit. on pp. 62, 67).

[167] Sofia Vallecorsa, Federico Carminati, and Gulruk Khattak. “3D convolutional GAN for fast simulation”. In: *EPJ Web of Conferences* 214 (2019). Ed. by A. Forti et al., p. 02010. ISSN: 2100-014X. DOI: 10.1051/epjconf/201921402010. URL: <https://www.epj-conferences.org/10.1051/epjconf/201921402010> (cit. on pp. 62, 67).

[168] Gul Rukh Khattak et al. *Fast Simulation of a High Granularity Calorimeter by Generative Adversarial Networks*. Sept. 9, 2021. arXiv: 2109.07388[hep-ex, physics:physics]. URL: <http://arxiv.org/abs/2109.07388> (cit. on pp. 62, 67, 68).

[169] Dawit Belayneh et al. "Calorimetry with deep learning: particle simulation and reconstruction for collider physics". In: *The European Physical Journal C* 80.7 (July 31, 2020), p. 688. ISSN: 1434-6052. DOI: [10.1140/epjc/s10052-020-8251-9](https://doi.org/10.1140/epjc/s10052-020-8251-9). URL: <https://doi.org/10.1140/epjc/s10052-020-8251-9> (cit. on pp. 62, 67).

[170] Martin Erdmann et al. *Generating and refining particle detector simulations using the Wasserstein distance in adversarial networks*. Feb. 9, 2018. arXiv: [1802.03325](https://arxiv.org/abs/1802.03325) [astro-ph, physics:hep-ex]. URL: [http://arxiv.org/abs/1802.03325](https://arxiv.org/abs/1802.03325) (cit. on p. 62).

[171] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: [10.1007/978-3-319-24574-4_28](https://doi.org/10.1007/978-3-319-24574-4_28) (cit. on pp. 63, 65).

[172] Matej Srebre et al. "Generation of Belle II Pixel Detector Background Data with a GAN". In: *EPJ Web of Conferences* 245 (2020). Ed. by C. Doglioni et al., p. 02010. ISSN: 2100-014X. DOI: [10.1051/epjconf/202024502010](https://doi.org/10.1051/epjconf/202024502010). URL: <https://www.epj-conferences.org/10.1051/epjconf/202024502010> (cit. on pp. 63, 77, 80, 90).

[173] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. Dec. 25, 2017. arXiv: [1704.00028](https://arxiv.org/abs/1704.00028) [cs, stat]. URL: [http://arxiv.org/abs/1704.00028](https://arxiv.org/abs/1704.00028) (cit. on p. 63).

[174] Viktoria Chekalina et al. "Generative Models for Fast Calorimeter Simulation.LHCb case". In: *EPJ Web of Conferences* 214 (2019), p. 02034. ISSN: 2100-014X. DOI: [10.1051/epjconf/201921402034](https://doi.org/10.1051/epjconf/201921402034). arXiv: [1812.01319](https://arxiv.org/abs/1812.01319) [physics]. URL: [http://arxiv.org/abs/1812.01319](https://arxiv.org/abs/1812.01319) (cit. on p. 63).

[175] Saúl Alonso-Monsalve and Leigh H. Whitehead. "Image-based model parameter optimization using Model-Assisted Generative Adversarial Networks". In: *IEEE Transactions on Neural Networks and Learning Systems* 31.12 (Dec. 2020), pp. 5645–5650. ISSN: 2162-237X, 2162-2388. DOI: [10.1109/TNNLS.2020.2969327](https://doi.org/10.1109/TNNLS.2020.2969327). arXiv: [1812.00879](https://arxiv.org/abs/1812.00879) [hep-ex, stat]. URL: [http://arxiv.org/abs/1812.00879](https://arxiv.org/abs/1812.00879) (cit. on p. 63).

[176] Jane Bromley et al. "Signature Verification using a "Siamese" Time Delay Neural Network". In: *Advances in Neural Information Processing Systems*. Vol. 6. Morgan-Kaufmann, 1993. URL: <https://proceedings.neurips.cc/paper/1993/hash/288cc0ff022877bd3df94bc9360b9c5d-Abstract.html> (cit. on p. 63).

[177] S. Chopra, R. Hadsell, and Y. LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). Vol. 1.

ISSN: 1063-6919. June 2005, 539–546 vol. 1. DOI: [10.1109/CVPR.2005.202](https://doi.org/10.1109/CVPR.2005.202) (cit. on p. 63).

[178] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. “Siamese Neural Networks for One-shot Image Recognition”. In: () (cit. on p. 63).

[179] Sascha Diefenbacher et al. “DCTRGAN: Improving the Precision of Generative Models with Reweighting”. In: *Journal of Instrumentation* 15.11 (Nov. 2, 2020), P11004–P11004. ISSN: 1748-0221. DOI: [10.1088/1748-0221/15/11/P11004](https://doi.org/10.1088/1748-0221/15/11/P11004). arXiv: [2009.03796\[hep-ex, physics:hep-ph, physics:physics, stat\]](https://arxiv.org/abs/2009.03796). URL: <http://arxiv.org/abs/2009.03796> (cit. on p. 63).

[180] Anders Andreassen and Benjamin Nachman. “Neural Networks for Full Phase-space Reweighting and Parameter Tuning”. In: *Physical Review D* 101.9 (May 12, 2020), p. 091901. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.101.091901](https://doi.org/10.1103/PhysRevD.101.091901). arXiv: [1907.08209\[hep-ex, physics:hep-ph, stat\]](https://arxiv.org/abs/1907.08209). URL: <http://arxiv.org/abs/1907.08209> (cit. on p. 63).

[181] C. Badiali et al. *Efficiency Parameterization with Neural Networks*. May 17, 2020. arXiv: [2004.02665\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/2004.02665). URL: <http://arxiv.org/abs/2004.02665> (cit. on p. 63).

[182] Raghav Kansal et al. *Graph Generative Adversarial Networks for Sparse Data Generation in High Energy Physics*. Jan. 30, 2021. arXiv: [2012.00173\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2012.00173). URL: <http://arxiv.org/abs/2012.00173> (cit. on pp. 63, 67).

[183] Raghav Kansal et al. *Particle Cloud Generation with Message Passing Generative Adversarial Networks*. Jan. 21, 2022. arXiv: [2106.11535\[hep-ex\]](https://arxiv.org/abs/2106.11535). URL: <http://arxiv.org/abs/2106.11535> (cit. on pp. 63, 65, 67, 108, 111).

[184] Justin Gilmer et al. *Neural Message Passing for Quantum Chemistry*. June 12, 2017. arXiv: [1704.01212\[cs\]](https://arxiv.org/abs/1704.01212). URL: <http://arxiv.org/abs/1704.01212> (cit. on p. 63).

[185] Ramon Winterhalder, Marco Bellagente, and Benjamin Nachman. *Latent Space Refinement for Deep Generative Models*. Nov. 3, 2021. arXiv: [2106.00792\[hep-ex, physics:hep-ph, physics:physics, stat\]](https://arxiv.org/abs/2106.00792). URL: <http://arxiv.org/abs/2106.00792> (cit. on p. 63).

[186] Sergey Shirobokov et al. *Black-Box Optimization with Local Generative Surrogates*. June 15, 2020. arXiv: [2002.04632\[hep-ex, physics:physics, stat\]](https://arxiv.org/abs/2002.04632). URL: <http://arxiv.org/abs/2002.04632> (cit. on p. 63).

[187] K. Jaruskova and S. Vallecorsa. “Ensemble Models for Calorimeter Simulations”. In: *Journal of Physics: Conference Series* 2438.1 (Feb. 2023). Publisher: IOP Publishing, p. 012080. ISSN: 1742-6596. DOI: [10.1088/1742-6596/2438/1/012080](https://doi.org/10.1088/1742-6596/2438/1/012080). URL: <https://dx.doi.org/10.1088/1742-6596/2438/1/012080> (cit. on p. 63).

[188] Ilya Tolstikhin et al. *AdaGAN: Boosting Generative Models*. May 24, 2017. arXiv: [1701.02386\[cs, stat\]](https://arxiv.org/abs/1701.02386). URL: <http://arxiv.org/abs/1701.02386> (cit. on p. 63).

[189] Bobak Hashemi et al. *LHC analysis-specific datasets with Generative Adversarial Networks*. Jan. 16, 2019. arXiv: [1901.05282\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/1901.05282). URL: <http://arxiv.org/abs/1901.05282> (cit. on p. 63).

[190] Riccardo Di Sipio et al. “DijetGAN: A Generative-Adversarial Network Approach for the Simulation of QCD Dijet Events at the LHC”. In: *Journal of High Energy Physics* 2019.8 (Aug. 2019), p. 110. ISSN: 1029-8479. DOI: [10.1007/JHEP08\(2019\)110](https://doi.org/10.1007/JHEP08(2019)110). arXiv: [1903.02433\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/1903.02433). URL: <http://arxiv.org/abs/1903.02433> (cit. on pp. 63, 68, 108).

[191] Anja Butter, Tilman Plehn, and Ramon Winterhalder. “How to GAN LHC Events”. In: *SciPost Physics* 7.6 (Dec. 4, 2019), p. 075. ISSN: 2542-4653. DOI: [10.21468/SciPostPhys.7.6.075](https://doi.org/10.21468/SciPostPhys.7.6.075). arXiv: [1907.03764\[hep-ph\]](https://arxiv.org/abs/1907.03764). URL: <http://arxiv.org/abs/1907.03764> (cit. on p. 63).

[192] Stefano Carrazza and Frédéric A. Dreyer. “Lund jet images from generative and cycle-consistent adversarial networks”. In: *The European Physical Journal C* 79.11 (Nov. 2019), p. 979. ISSN: 1434-6044, 1434-6052. DOI: [10.1140/epjc/s10052-019-7501-1](https://doi.org/10.1140/epjc/s10052-019-7501-1). arXiv: [1909.01359\[hep-ex, physics:hep-ph, stat\]](https://arxiv.org/abs/1909.01359). URL: <http://arxiv.org/abs/1909.01359> (cit. on p. 63).

[193] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. Aug. 24, 2020. arXiv: [1703.10593\[cs\]](https://arxiv.org/abs/1703.10593). URL: <http://arxiv.org/abs/1703.10593> (cit. on p. 63).

[194] Frédéric A. Dreyer, Gavin P. Salam, and Grégory Soyez. “The Lund jet plane”. In: *Journal of High Energy Physics* 2018.12 (Dec. 11, 2018), p. 64. ISSN: 1029-8479. DOI: [10.1007/JHEP12\(2018\)064](https://doi.org/10.1007/JHEP12(2018)064). URL: [https://doi.org/10.1007/JHEP12\(2018\)064](https://doi.org/10.1007/JHEP12(2018)064) (cit. on p. 63).

[195] Steven Farrell et al. “Next Generation Generative Neural Networks for HEP”. In: *EPJ Web of Conferences* 214 (2019). Publisher: EDP Sciences, p. 09005. ISSN: 2100-014X. DOI: [10.1051/epjconf/201921409005](https://doi.org/10.1051/epjconf/201921409005). URL: https://www.epj-conferences.org/articles/epjconf/abs/2019/19/epjconf_chep2018_09005/epjconf_chep2018_09005.html (cit. on p. 63).

[196] Jimmian Li, Cong Zhang, and Rao Zhang. “Polarization measurement for the dileptonic channel of $W^+ W^-$ scattering using generative adversarial network”. In: *Physical Review D* 105.1 (Jan. 5, 2022), p. 016005. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.105.016005](https://doi.org/10.1103/PhysRevD.105.016005). arXiv: [2109.09924\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/2109.09924). URL: <http://arxiv.org/abs/2109.09924> (cit. on p. 63).

[197] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. Mar. 29, 2019. arXiv: [1812.04948\[cs, stat\]](https://arxiv.org/abs/1812.04948). URL: <http://arxiv.org/abs/1812.04948> (cit. on p. 63).

[198] Yasir Alanazi et al. “Simulation of electron-proton scattering events by a Feature-Augmented and Transformed Generative Adversarial Network (FAT-GAN)”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Aug. 2021, pp. 2126–2132. DOI: [10.24963/ijcai.2021/293](https://doi.org/10.24963/ijcai.2021/293). arXiv: [2001.11103](https://arxiv.org/abs/2001.11103) [hep-ex, physics:hep-ph, stat]. URL: <http://arxiv.org/abs/2001.11103> (cit. on p. 63).

[199] Luisa Velasco et al. “cFAT-GAN: Conditional Simulation of Electron–Proton Scattering Events with Variate Beam Energies by a Feature Augmented and Transformed Generative Adversarial Network”. In: *Deep Learning Applications, Volume 3*. Ed. by M. Arif Wani et al. Advances in Intelligent Systems and Computing. Singapore: Springer, 2022, pp. 245–261. ISBN: 9789811633577. DOI: [10.1007/978-981-16-3357-7_10](https://doi.org/10.1007/978-981-16-3357-7_10). URL: https://doi.org/10.1007/978-981-16-3357-7_10 (cit. on p. 63).

[200] Carlos Bravo-Prieto et al. “Style-based quantum generative adversarial networks for Monte Carlo events”. In: *Quantum* 6 (Aug. 17, 2022), p. 777. ISSN: 2521-327X. DOI: [10.22331/q-2022-08-17-777](https://doi.org/10.22331/q-2022-08-17-777). arXiv: [2110.06933](https://arxiv.org/abs/2110.06933) [hep-ph, physics:quant-ph]. URL: <http://arxiv.org/abs/2110.06933> (cit. on p. 64).

[201] Jessica N. Howard et al. “Learning to Simulate High Energy Particle Collisions from Unlabeled Data”. In: *Scientific Reports* 12.1 (May 9, 2022), p. 7567. ISSN: 2045-2322. DOI: [10.1038/s41598-022-10966-7](https://doi.org/10.1038/s41598-022-10966-7). arXiv: [2101.08944](https://arxiv.org/abs/2101.08944) [hep-ex, physics:hep-ph]. URL: <http://arxiv.org/abs/2101.08944> (cit. on p. 64).

[202] Lucio Anderlini et al. “Generative models uncertainty estimation”. In: *Journal of Physics: Conference Series* 2438.1 (Feb. 1, 2023), p. 012088. ISSN: 1742-6588, 1742-6596. DOI: [10.1088/1742-6596/2438/1/012088](https://doi.org/10.1088/1742-6596/2438/1/012088). arXiv: [2210.09767](https://arxiv.org/abs/2210.09767) [hep-ex, physics:hep-ph]. URL: <http://arxiv.org/abs/2210.09767> (cit. on pp. 64, 68).

[203] Andrey Malinin, Bruno Mlodzieniec, and Mark Gales. *Ensemble Distribution Distillation*. Nov. 25, 2019. arXiv: [1905.00076](https://arxiv.org/abs/1905.00076) [cs, stat]. URL: <http://arxiv.org/abs/1905.00076> (cit. on p. 64).

[204] Rianne van den Berg et al. *Sylvester Normalizing Flows for Variational Inference*. Feb. 20, 2019. DOI: [10.48550/arXiv.1803.05649](https://doi.org/10.48550/arXiv.1803.05649). arXiv: [1803.05649](https://arxiv.org/abs/1803.05649) [cs, stat]. URL: <http://arxiv.org/abs/1803.05649> (cit. on p. 64).

[205] Claudius Krause and David Shih. *CaloFlow II: Even Faster and Still Accurate Generation of Calorimeter Showers with Normalizing Flows*. May 5, 2023. arXiv: [2110.11377](https://arxiv.org/abs/2110.11377) [hep-ex, physics:hep-ph, physics:physics]. URL: <http://arxiv.org/abs/2110.11377> (cit. on pp. 64, 68).

[206] Mathieu Germain et al. *MADE: Masked Autoencoder for Distribution Estimation*. June 5, 2015. DOI: [10.48550/arXiv.1502.03509](https://doi.org/10.48550/arXiv.1502.03509). arXiv: [1502.03509](https://arxiv.org/abs/1502.03509) [cs, stat]. URL: <http://arxiv.org/abs/1502.03509> (cit. on p. 64).

[207] Conor Durkan et al. *Neural Spline Flows*. Dec. 2, 2019. DOI: [10.48550/arXiv.1906.04032](https://doi.org/10.48550/arXiv.1906.04032). arXiv: [1906.04032](https://arxiv.org/abs/1906.04032) [cs, stat]. URL: <http://arxiv.org/abs/1906.04032> (cit. on p. 64).

[208] Diederik P. Kingma et al. *Improving Variational Inference with Inverse Autoregressive Flow*. Jan. 30, 2017. arXiv: [1606.04934\[cs, stat\]](https://arxiv.org/abs/1606.04934). URL: <http://arxiv.org/abs/1606.04934> (cit. on p. 64).

[209] Allison Xu et al. *Generative Machine Learning for Detector Response Modeling with a Conditional Normalizing Flow*. Apr. 27, 2023. arXiv: [2303.10148\[hep-ex, physics:physics\]](https://arxiv.org/abs/2303.10148). URL: <http://arxiv.org/abs/2303.10148> (cit. on p. 64).

[210] George Papamakarios, Theo Pavlakou, and Iain Murray. *Masked Autoregressive Flow for Density Estimation*. June 14, 2018. DOI: [10.48550/arXiv.1705.07057](https://doi.org/10.48550/arXiv.1705.07057). arXiv: [1705.07057\[cs, stat\]](https://arxiv.org/abs/1705.07057). URL: <http://arxiv.org/abs/1705.07057> (cit. on p. 64).

[211] Yang Song and Stefano Ermon. *Generative Modeling by Estimating Gradients of the Data Distribution*. Oct. 10, 2020. DOI: [10.48550/arXiv.1907.05600](https://doi.org/10.48550/arXiv.1907.05600). arXiv: [1907.05600\[cs, stat\]](https://arxiv.org/abs/1907.05600). URL: <http://arxiv.org/abs/1907.05600> (cit. on p. 64).

[212] Yang Song and Diederik P. Kingma. *How to Train Your Energy-Based Models*. Feb. 17, 2021. DOI: [10.48550/arXiv.2101.03288](https://doi.org/10.48550/arXiv.2101.03288). arXiv: [2101.03288\[cs, stat\]](https://arxiv.org/abs/2101.03288). URL: <http://arxiv.org/abs/2101.03288> (cit. on p. 64).

[213] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Dec. 16, 2020. DOI: [10.48550/arXiv.2006.11239](https://doi.org/10.48550/arXiv.2006.11239). arXiv: [2006.11239\[cs, stat\]](https://arxiv.org/abs/2006.11239). URL: <http://arxiv.org/abs/2006.11239> (cit. on p. 64).

[214] Diederik P. Kingma et al. *Variational Diffusion Models*. Apr. 13, 2023. DOI: [10.48550/arXiv.2107.00630](https://doi.org/10.48550/arXiv.2107.00630). arXiv: [2107.00630\[cs, stat\]](https://arxiv.org/abs/2107.00630). URL: <http://arxiv.org/abs/2107.00630> (cit. on p. 64).

[215] Aditya Ramesh et al. *Hierarchical Text-Conditional Image Generation with CLIP Latents*. Apr. 12, 2022. DOI: [10.48550/arXiv.2204.06125](https://doi.org/10.48550/arXiv.2204.06125). arXiv: [2204.06125\[cs\]](https://arxiv.org/abs/2204.06125). URL: <http://arxiv.org/abs/2204.06125> (cit. on p. 64).

[216] Vinicius Mikuni and Benjamin Nachman. “Score-based Generative Models for Calorimeter Shower Simulation”. In: *Physical Review D* 106.9 (Nov. 28, 2022), p. 092009. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.106.092009](https://doi.org/10.1103/PhysRevD.106.092009). arXiv: [2206.11898\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2206.11898). URL: <http://arxiv.org/abs/2206.11898> (cit. on pp. 65, 67).

[217] Vinicius Mikuni, Benjamin Nachman, and Mariel Pettee. *Fast Point Cloud Generation with Diffusion Models in High Energy Physics*. Apr. 3, 2023. DOI: [10.48550/arXiv.2304.01266](https://doi.org/10.48550/arXiv.2304.01266). arXiv: [2304.01266\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/2304.01266). URL: <http://arxiv.org/abs/2304.01266> (cit. on pp. 65, 67).

[218] Tim Salimans and Jonathan Ho. “Progressive Distillation for Fast Sampling of Diffusion Models”. In: International Conference on Learning Representations. Jan. 28, 2022. URL: <https://openreview.net/forum?id=TIIdIXIpzhoI> (cit. on p. 65).

[219] Aaron van den Oord et al. *Conditional Image Generation with PixelCNN Decoders*. June 18, 2016. DOI: [10.48550/arXiv.1606.05328](https://doi.org/10.48550/arXiv.1606.05328). arXiv: [1606.05328\[cs\]](https://arxiv.org/abs/1606.05328). URL: <http://arxiv.org/abs/1606.05328> (cit. on p. 65).

[220] X. I. Chen et al. “PixelSNAIL: An Improved Autoregressive Generative Model”. In: *Proceedings of the 35th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, July 3, 2018, pp. 864–872. URL: <https://proceedings.mlr.press/v80/chen18h.html> (cit. on p. 65).

[221] Jiaxuan You et al. “GraphRNN: Generating Realistic Graphs with Deep Autoregressive Models”. In: () (cit. on p. 65).

[222] Yadong Lu et al. “SARM: Sparse Autoregressive Model for Scalable Generation of Sparse Images in Particle Physics”. In: *Physical Review D* 103.3 (Feb. 16, 2021), p. 036012. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.103.036012](https://doi.org/10.1103/PhysRevD.103.036012). arXiv: [2009.14017\[hep-ex, physics:physics\]](https://arxiv.org/abs/2009.14017). URL: <http://arxiv.org/abs/2009.14017> (cit. on pp. 65, 67).

[223] Junze Liu et al. *Geometry-aware Autoregressive Models for Calorimeter Shower Simulations*. Dec. 15, 2022. arXiv: [2212.08233\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2212.08233). URL: <http://arxiv.org/abs/2212.08233> (cit. on pp. 65, 67, 69).

[224] Junze Liu et al. *Generalizing to new calorimeter geometries with Geometry-Aware Autoregressive Models (GAAMs) for fast calorimeter simulation*. May 19, 2023. arXiv: [2305.11531\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2305.11531). URL: <http://arxiv.org/abs/2305.11531> (cit. on pp. 65, 67, 69).

[225] Francesco Armando Di Bello et al. *Conditional Generative Modelling of Reconstructed Particles at Collider Experiments*. Nov. 11, 2022. arXiv: [2211.06406\[hep-ex\]](https://arxiv.org/abs/2211.06406). URL: <http://arxiv.org/abs/2211.06406> (cit. on pp. 65, 67, 108, 111).

[226] Francesco Locatello et al. *Object-Centric Learning with Slot Attention*. Oct. 14, 2020. arXiv: [2006.15055\[cs, stat\]](https://arxiv.org/abs/2006.15055). URL: <http://arxiv.org/abs/2006.15055> (cit. on p. 65).

[227] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. Sept. 2, 2014. DOI: [10.48550/arXiv.1406.1078](https://doi.org/10.48550/arXiv.1406.1078). arXiv: [1406.1078\[cs, stat\]](https://arxiv.org/abs/1406.1078). URL: <http://arxiv.org/abs/1406.1078> (cit. on p. 65).

[228] Thorben Finke et al. *Learning the language of QCD jets with transformers*. Mar. 20, 2023. arXiv: [2303.07364\[hep-ph\]](https://arxiv.org/abs/2303.07364). URL: <http://arxiv.org/abs/2303.07364> (cit. on pp. 65, 67).

[229] Rasool Fakoor et al. *TraDE: Transformers for Density Estimation*. Oct. 14, 2020. arXiv: [2004.02441\[cs, stat\]](https://arxiv.org/abs/2004.02441). URL: <http://arxiv.org/abs/2004.02441> (cit. on p. 65).

[230] Shengjia Zhao et al. *Bias and Generalization in Deep Generative Models: An Empirical Study*. Nov. 7, 2018. DOI: [10.48550/arXiv.1811.03259](https://doi.org/10.48550/arXiv.1811.03259). arXiv: [1811.03259\[cs, stat\]](https://arxiv.org/abs/1811.03259). URL: <http://arxiv.org/abs/1811.03259> (cit. on p. 66).

[231] Jeff Z. HaoChen and Tengyu Ma. *A Theoretical Study of Inductive Biases in Contrastive Learning*. Apr. 8, 2023. DOI: [10.48550/arXiv.2211.14699](https://doi.org/10.48550/arXiv.2211.14699). arXiv: [2211.14699](https://arxiv.org/abs/2211.14699) [cs, stat]. URL: <http://arxiv.org/abs/2211.14699> (cit. on p. 66).

[232] Dominik Zietlow, Michal Rolinek, and Georg Martius. “Demystifying Inductive Biases for (Beta-)VAE Based Architectures”. In: *Proceedings of the 38th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, July 1, 2021, pp. 12945–12954. URL: <https://proceedings.mlr.press/v139/zietlow21a.html> (cit. on p. 66).

[233] Sven Krippendorf and Marc Syvaeri. *Detecting Symmetries with Neural Networks*. Mar. 30, 2020. DOI: [10.48550/arXiv.2003.13679](https://doi.org/10.48550/arXiv.2003.13679). arXiv: [2003.13679](https://arxiv.org/abs/2003.13679) [hep-th, physics:physics]. URL: <http://arxiv.org/abs/2003.13679> (cit. on p. 66).

[234] Barry M. Dillon et al. *Symmetries, Safety, and Self-Supervision*. Aug. 9, 2021. DOI: [10.48550/arXiv.2108.04253](https://doi.org/10.48550/arXiv.2108.04253). arXiv: [2108.04253](https://arxiv.org/abs/2108.04253) [hep-ph]. URL: <http://arxiv.org/abs/2108.04253> (cit. on p. 66).

[235] Gabriela Barenboim, Johannes Hirn, and Veronica Sanz. “Symmetry meets AI”. In: *SciPost Physics* 11.1 (July 15, 2021), p. 014. ISSN: 2542-4653. DOI: [10.21468/SciPostPhys.11.1.014](https://doi.org/10.21468/SciPostPhys.11.1.014). URL: <https://scipost.org/10.21468/SciPostPhys.11.1.014> (cit. on p. 66).

[236] Rupert Tombs and Christopher G. Lester. “A method to challenge symmetries in data with self-supervised learning”. In: *Journal of Instrumentation* 17.8 (Aug. 1, 2022), P08024. ISSN: 1748-0221. DOI: [10.1088/1748-0221/17/08/P08024](https://doi.org/10.1088/1748-0221/17/08/P08024). arXiv: [2111.05442](https://arxiv.org/abs/2111.05442) [hep-ph, physics:physics]. URL: <http://arxiv.org/abs/2111.05442> (cit. on p. 66).

[237] Krish Desai, Benjamin Nachman, and Jesse Thaler. “SymmetryGAN: Symmetry Discovery with Deep Learning”. In: *Physical Review D* 105.9 (May 26, 2022), p. 096031. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.105.096031](https://doi.org/10.1103/PhysRevD.105.096031). arXiv: [2112.05722](https://arxiv.org/abs/2112.05722) [hep-ph, physics:physics]. URL: <http://arxiv.org/abs/2112.05722> (cit. on p. 66).

[238] Brian Axelrod et al. *Sample Amplification: Increasing Dataset Size even when Learning is Impossible*. Dec. 2, 2019. DOI: [10.48550/arXiv.1904.12053](https://doi.org/10.48550/arXiv.1904.12053). arXiv: [1904.12053](https://arxiv.org/abs/1904.12053) [cs, math, stat]. URL: <http://arxiv.org/abs/1904.12053> (cit. on p. 66).

[239] Stefano Carrazza, Juan M. Cruz-Martinez, and Tanjona R. Rabemananjara. “Compressing PDF sets using generative adversarial networks”. In: *The European Physical Journal C* 81.6 (June 2021), p. 530. ISSN: 1434-6044, 1434-6052. DOI: [10.1140/epjc/s10052-021-09338-8](https://doi.org/10.1140/epjc/s10052-021-09338-8). arXiv: [2104.04535](https://arxiv.org/abs/2104.04535) [hep-ex, physics:hep-ph]. URL: <http://arxiv.org/abs/2104.04535> (cit. on p. 66).

[240] Ibrahim Chahrour and James D. Wells. “Comparing Machine Learning and Interpolation Methods for Loop-Level Calculations”. In: *SciPost Physics* 12.6 (June 8, 2022), p. 187. ISSN: 2542-4653. DOI: [10.21468/SciPostPhys.12.6.187](https://doi.org/10.21468/SciPostPhys.12.6.187). arXiv: [2111.14788\[hep-ph\]](https://arxiv.org/abs/2111.14788). URL: <http://arxiv.org/abs/2111.14788> (cit. on p. 66).

[241] Yi Hao et al. *Data Amplification: A Unified and Competitive Approach to Property Estimation*. Mar. 29, 2019. DOI: [10.48550/arXiv.1904.00070](https://doi.org/10.48550/arXiv.1904.00070). arXiv: [1904.00070\[cs,math,stat\]](https://arxiv.org/abs/1904.00070). URL: <http://arxiv.org/abs/1904.00070> (cit. on p. 66).

[242] Brian Axelrod et al. *On the Statistical Complexity of Sample Amplification*. Jan. 12, 2022. DOI: [10.48550/arXiv.2201.04315](https://doi.org/10.48550/arXiv.2201.04315). arXiv: [2201.04315\[cs,math,stat\]](https://arxiv.org/abs/2201.04315). URL: <http://arxiv.org/abs/2201.04315> (cit. on p. 66).

[243] Anja Butter et al. “GANplifying Event Samples”. In: *SciPost Physics* 10.6 (June 10, 2021), p. 139. ISSN: 2542-4653. DOI: [10.21468/SciPostPhys.10.6.139](https://doi.org/10.21468/SciPostPhys.10.6.139). arXiv: [2008.06545\[hep-ex,physics:hep-ph,physics:physics,stat\]](https://arxiv.org/abs/2008.06545). URL: <http://arxiv.org/abs/2008.06545> (cit. on p. 66).

[244] Anja Butter. “Amplifying Statistics using Generative Models”. In: () (cit. on pp. 66, 180).

[245] Sebastian Bieringer et al. “Calomplification – The Power of Generative Calorimeter Models”. In: *Journal of Instrumentation* 17.9 (Sept. 1, 2022), P09028. ISSN: 1748-0221. DOI: [10.1088/1748-0221/17/09/P09028](https://doi.org/10.1088/1748-0221/17/09/P09028). arXiv: [2202.07352\[hep-ex,physics:hep-ph\]](https://arxiv.org/abs/2202.07352). URL: <http://arxiv.org/abs/2202.07352> (cit. on pp. 66, 149).

[246] Ravid Shwartz-Ziv and Yann LeCun. *To Compress or Not to Compress- Self-Supervised Learning and Information Theory: A Review*. May 4, 2023. DOI: [10.48550/arXiv.2304.09355](https://doi.org/10.48550/arXiv.2304.09355). arXiv: [2304.09355\[cs,math\]](https://arxiv.org/abs/2304.09355). URL: <http://arxiv.org/abs/2304.09355> (cit. on p. 66).

[247] Randall Balestrieri et al. *A Cookbook of Self-Supervised Learning*. June 28, 2023. DOI: [10.48550/arXiv.2304.12210](https://doi.org/10.48550/arXiv.2304.12210). arXiv: [2304.12210\[cs\]](https://arxiv.org/abs/2304.12210). URL: <http://arxiv.org/abs/2304.12210> (cit. on pp. 66, 85).

[248] Wei Mu, Alexander I. Himmel, and Bryan Ramson. *Photon detection probability prediction using one-dimensional generative neural network*. Sept. 16, 2021. arXiv: [2109.07277\[hep-ex,physics:physics\]](https://arxiv.org/abs/2109.07277). URL: <http://arxiv.org/abs/2109.07277> (cit. on p. 67).

[249] Alberto Regadío, Luis Esteban, and Sebastián Sánchez-Prieto. “Synthesis of pulses from particle detectors with a Generative Adversarial Network (GAN)”. In: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 1033 (June 11, 2022), p. 166647. ISSN: 0168-9002. DOI: [10.1016/j.nima.2022.166647](https://doi.org/10.1016/j.nima.2022.166647). URL: <https://www.sciencedirect.com/science/article/pii/S0168900222002108> (cit. on p. 67).

[250] Erik Buhmann, Gregor Kasieczka, and Jesse Thaler. *EPiC-GAN: Equivariant Point Cloud Generation for Particle Jets*. Feb. 9, 2023. DOI: [10.48550/arXiv.2301.08128](https://doi.org/10.48550/arXiv.2301.08128). arXiv: [2301.08128\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2301.08128). URL: <http://arxiv.org/abs/2301.08128> (cit. on pp. 67, 108, 111, 113).

[251] Matthew Leigh et al. *PC-JeDi: Diffusion for Particle Cloud Generation in High Energy Physics*. Mar. 9, 2023. arXiv: [2303.05376\[hep-ex, physics:hep-ph\]](https://arxiv.org/abs/2303.05376). URL: <http://arxiv.org/abs/2303.05376> (cit. on p. 67).

[252] Erik Buhmann et al. *CaloClouds: Fast Geometry-Independent Highly-Granular Calorimeter Simulation*. May 8, 2023. DOI: [10.48550/arXiv.2305.04847](https://doi.org/10.48550/arXiv.2305.04847). arXiv: [2305.04847\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2305.04847). URL: <http://arxiv.org/abs/2305.04847> (cit. on pp. 67, 175, 181).

[253] Raghav Kansal et al. “Evaluating generative models in high energy physics”. In: *Physical Review D* 107.7 (Apr. 21, 2023), p. 076017. ISSN: 2470-0010, 2470-0029. DOI: [10.1103/PhysRevD.107.076017](https://doi.org/10.1103/PhysRevD.107.076017). URL: <https://link.aps.org/doi/10.1103/PhysRevD.107.076017> (cit. on p. 67).

[254] Jonas Köhler, Leon Klein, and Frank Noé. *Equivariant Flows: Exact Likelihood Generative Learning for Symmetric Densities*. Oct. 26, 2020. DOI: [10.48550/arXiv.2006.02425](https://doi.org/10.48550/arXiv.2006.02425). arXiv: [2006.02425\[physics, stat\]](https://arxiv.org/abs/2006.02425). URL: <http://arxiv.org/abs/2006.02425> (cit. on pp. 67, 111).

[255] Sascha Diefenbacher et al. *L2LFlows: Generating High-Fidelity 3D Calorimeter Images*. Feb. 22, 2023. DOI: [10.48550/arXiv.2302.11594](https://doi.org/10.48550/arXiv.2302.11594). arXiv: [2302.11594\[hep-ex, physics:hep-ph, physics:physics\]](https://arxiv.org/abs/2302.11594). URL: <http://arxiv.org/abs/2302.11594> (cit. on p. 67).

[256] Anja Butter et al. *Jet Diffusion versus JetGPT – Modern Networks for the LHC*. May 17, 2023. DOI: [10.48550/arXiv.2305.10475](https://doi.org/10.48550/arXiv.2305.10475). arXiv: [2305.10475\[hep-ph\]](https://arxiv.org/abs/2305.10475). URL: <http://arxiv.org/abs/2305.10475> (cit. on pp. 67, 108).

[257] O. Brüning and L. Rossi. “Chapter 1: High-Luminosity Large Hadron Collider”. In: *CERN Yellow Reports: Monographs* 10 (Dec. 17, 2020), pp. 1–1. ISSN: 2519-8076. DOI: [10.23731/CYRM-2020-0010.1](https://doi.org/10.23731/CYRM-2020-0010.1). URL: <https://e-publishing.cern.ch/index.php/CYRM/article/view/1153> (cit. on pp. 68, 104, 181).

[258] Nadja Strobbe and CMS Collaboration. “Status and progress on the CMS Phase-II detector upgrades”. In: *Proceedings of The Ninth Annual Conference on Large Hadron Collider Physics — PoS(LHCP2021)*. The Ninth Annual Conference on Large Hadron Collider Physics. zoom: Sissa Medialab, Oct. 26, 2021, p. 318. DOI: [10.22323/1.397.0318](https://doi.org/10.22323/1.397.0318). URL: <https://pos.sissa.it/397/318> (cit. on p. 68).

[259] Ivan Anishchenko et al. “De novo protein design by deep network hallucination”. In: *Nature* 600.7889 (Dec. 2021). Number: 7889 Publisher: Nature Publishing Group, pp. 547–552. ISSN: 1476-4687. DOI: [10.1038/s41586-021-04184-w](https://doi.org/10.1038/s41586-021-04184-w). URL: <https://www.nature.com/articles/s41586-021-04184-w> (cit. on p. 68).

[260] Pankaj Rajak et al. “Autonomous reinforcement learning agent for stretchable kirigami design of 2D materials”. In: *npj Computational Materials* 7.1 (July 9, 2021). Number: 1 Publisher: Nature Publishing Group, pp. 1–8. ISSN: 2057-3960. DOI: [10.1038/s41524-021-00572-y](https://doi.org/10.1038/s41524-021-00572-y). URL: <https://www.nature.com/articles/s41524-021-00572-y> (cit. on p. 68).

[261] Suman Ravuri et al. “Skilful precipitation nowcasting using deep generative models of radar”. In: *Nature* 597.7878 (Sept. 2021). Number: 7878 Publisher: Nature Publishing Group, pp. 672–677. ISSN: 1476-4687. DOI: [10.1038/s41586-021-03854-z](https://doi.org/10.1038/s41586-021-03854-z). URL: <https://www.nature.com/articles/s41586-021-03854-z> (cit. on pp. 68, 182).

[262] Ali Madani et al. “Large language models generate functional protein sequences across diverse families”. In: *Nature Biotechnology* 41.8 (Aug. 2023). Number: 8 Publisher: Nature Publishing Group, pp. 1099–1106. ISSN: 1546-1696. DOI: [10.1038/s41587-022-01618-2](https://doi.org/10.1038/s41587-022-01618-2). URL: <https://www.nature.com/articles/s41587-022-01618-2> (cit. on pp. 68, 116).

[263] Zhaonan Li and Hongfu Liu. *Zero-Knowledge Zero-Shot Learning for Novel Visual Category Discovery*. Feb. 8, 2023. DOI: [10.48550/arXiv.2302.04427](https://doi.org/10.48550/arXiv.2302.04427). arXiv: [2302.04427\[cs\]](https://arxiv.org/abs/2302.04427). URL: [http://arxiv.org/abs/2302.04427](https://arxiv.org/abs/2302.04427) (cit. on p. 68).

[264] Johann Brehmer. “Simulation-based inference in particle physics”. In: *Nature Reviews Physics* 3.5 (May 2021). Number: 5 Publisher: Nature Publishing Group, pp. 305–305. ISSN: 2522-5820. DOI: [10.1038/s42254-021-00305-6](https://doi.org/10.1038/s42254-021-00305-6). URL: <https://www.nature.com/articles/s42254-021-00305-6> (cit. on p. 68).

[265] Jan Boelts et al. “Flexible and efficient simulation-based inference for models of decision-making”. In: *eLife* 11 (July 27, 2022). Ed. by Valentin Wyart et al. Publisher: eLife Sciences Publications, Ltd, e77220. ISSN: 2050-084X. DOI: [10.7554/eLife.77220](https://doi.org/10.7554/eLife.77220). URL: <https://doi.org/10.7554/eLife.77220> (cit. on p. 68).

[266] [\[2111.06119\] Fine-Grained Image Analysis with Deep Learning: A Survey](https://arxiv.org/abs/2111.06119). URL: <https://arxiv.org/abs/2111.06119> (cit. on pp. 72, 73, 76).

[267] *Visual and semantic similarity in ImageNet / Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*. URL: <https://dl.acm.org/doi/10.1109/CVPR.2011.5995474> (cit. on p. 76).

[268] Yang Zhao et al. *Feature Quantization Improves GAN Training*. July 14, 2020. DOI: [10.48550/arXiv.2004.02088](https://doi.org/10.48550/arXiv.2004.02088). arXiv: [2004.02088\[cs, stat\]](https://arxiv.org/abs/2004.02088). URL: [http://arxiv.org/abs/2004.02088](https://arxiv.org/abs/2004.02088) (cit. on p. 78).

[269] Han Zhang et al. *Consistency Regularization for Generative Adversarial Networks*. Feb. 18, 2020. arXiv: [1910.12027\[cs, stat\]](https://arxiv.org/abs/1910.12027). URL: [http://arxiv.org/abs/1910.12027](https://arxiv.org/abs/1910.12027) (cit. on p. 79).

[270] Sahand Sharifzadeh, Sina Moayed Baharlou, and Volker Tresp. *Classification by Attention: Scene Graph Classification with Prior Knowledge*. Dec. 17, 2020. DOI: [10.48550/arXiv.2011.10084](https://doi.org/10.48550/arXiv.2011.10084). arXiv: [2011.10084\[cs\]](https://arxiv.org/abs/2011.10084). URL: [http://arxiv.org/abs/2011.10084](https://arxiv.org/abs/2011.10084) (cit. on pp. 81, 83).

[271] Dongxu Zhang and Zhichao Yang. *Word Embedding Perturbation for Sentence Classification*. Apr. 22, 2018. DOI: [10.48550/arXiv.1804.08166](https://doi.org/10.48550/arXiv.1804.08166). arXiv: [1804.08166](https://arxiv.org/abs/1804.08166) [cs]. URL: <http://arxiv.org/abs/1804.08166> (cit. on pp. 81, 85).

[272] Tongzhou Wang and Phillip Isola. *Understanding Contrastive Representation Learning through Alignment and Uniformity on the Hypersphere*. Aug. 15, 2022. DOI: [10.48550/arXiv.2005.10242](https://doi.org/10.48550/arXiv.2005.10242). arXiv: [2005.10242](https://arxiv.org/abs/2005.10242) [cs, stat]. URL: <http://arxiv.org/abs/2005.10242> (cit. on pp. 81, 87).

[273] Drew A. Hudson and Larry Zitnick. “Generative Adversarial Transformers”. In: *Proceedings of the 38th International Conference on Machine Learning*. International Conference on Machine Learning. ISSN: 2640-3498. PMLR, July 1, 2021, pp. 4487–4499. URL: <https://proceedings.mlr.press/v139/hudson21a.html> (cit. on p. 81).

[274] Yifan Jiang, Shiyu Chang, and Zhangyang Wang. *TransGAN: Two Pure Transformers Can Make One Strong GAN, and That Can Scale Up*. Dec. 8, 2021. DOI: [10.48550/arXiv.2102.07074](https://doi.org/10.48550/arXiv.2102.07074). arXiv: [2102.07074](https://arxiv.org/abs/2102.07074) [cs]. URL: <http://arxiv.org/abs/2102.07074> (cit. on p. 81).

[275] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. June 3, 2021. DOI: [10.48550/arXiv.2010.11929](https://doi.org/10.48550/arXiv.2010.11929). arXiv: [2010.11929](https://arxiv.org/abs/2010.11929) [cs]. URL: <http://arxiv.org/abs/2010.11929> (cit. on p. 81).

[276] Liyuan Liu et al. *Understanding the Difficulty of Training Transformers*. Sept. 18, 2020. DOI: [10.48550/arXiv.2004.08249](https://doi.org/10.48550/arXiv.2004.08249). arXiv: [2004.08249](https://arxiv.org/abs/2004.08249) [cs, stat]. URL: <http://arxiv.org/abs/2004.08249> (cit. on p. 82).

[277] Nicholas Guttenberg et al. *Permutation-equivariant neural networks applied to dynamics prediction*. Dec. 14, 2016. DOI: [10.48550/arXiv.1612.04530](https://doi.org/10.48550/arXiv.1612.04530). arXiv: [1612.04530](https://arxiv.org/abs/1612.04530) [cs, stat]. URL: <http://arxiv.org/abs/1612.04530> (cit. on p. 83).

[278] Siamak Ravanbakhsh, Jeff Schneider, and Barnabas Poczos. *Equivariance Through Parameter-Sharing*. June 13, 2017. DOI: [10.48550/arXiv.1702.08389](https://doi.org/10.48550/arXiv.1702.08389). arXiv: [1702.08389](https://arxiv.org/abs/1702.08389) [cs, stat]. URL: <http://arxiv.org/abs/1702.08389> (cit. on p. 83).

[279] Kevin Clark et al. “What Does BERT Look at? An Analysis of BERT’s Attention”. In: *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. BlackboxNLP 2019. Florence, Italy: Association for Computational Linguistics, Aug. 2019, pp. 276–286. DOI: [10.18653/v1/W19-4828](https://doi.org/10.18653/v1/W19-4828). URL: <https://aclanthology.org/W19-4828> (cit. on p. 83).

[280] Chulhee Yun et al. *Are Transformers universal approximators of sequence-to-sequence functions?* Feb. 24, 2020. DOI: [10.48550/arXiv.1912.10077](https://doi.org/10.48550/arXiv.1912.10077). arXiv: [1912.10077](https://arxiv.org/abs/1912.10077) [cs, stat]. URL: <http://arxiv.org/abs/1912.10077> (cit. on p. 83).

[281] Feng Wang et al. “NormFace: L2 Hypersphere Embedding for Face Verification”. In: *Proceedings of the 25th ACM international conference on Multimedia*. Oct. 19, 2017, pp. 1041–1049. DOI: [10.1145/3123266.3123359](https://doi.org/10.1145/3123266.3123359). arXiv: [1704.06369](https://arxiv.org/abs/1704.06369) [cs]. URL: <http://arxiv.org/abs/1704.06369> (cit. on p. 84).

[282] *[1703.01203] Stochastic Separation Theorems*. URL: <https://arxiv.org/abs/1703.01203> (cit. on p. 84).

[283] Lilian Weng. *Self-Supervised Representation Learning*. Section: posts. Nov. 10, 2019. URL: <https://lilianweng.github.io/posts/2019-11-10-self-supervised/> (cit. on p. 85).

[284] *On Information and Sufficiency*. URL: <https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-22/issue-1/On-Information-and-Sufficiency/10.1214/aoms/1177729694.full> (cit. on p. 85).

[285] Longbing Cao. “Coupling Learning of Complex Interactions”. In: *Information Processing & Management* 51.2 (Mar. 2015), pp. 167–186. ISSN: 03064573. DOI: [10.1016/j.ipm.2014.08.007](https://doi.org/10.1016/j.ipm.2014.08.007). arXiv: [2007.13534\[cs\]](https://arxiv.org/abs/2007.13534). URL: <http://arxiv.org/abs/2007.13534> (cit. on p. 85).

[286] *XXIV. On the structure of the atom: an investigation of the stability and periods of oscillation of a number of corpuscles arranged at equal intervals around the circumference of a circle; with application of the results to the theory of atomic structure: The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science: Vol 7, No 39*. URL: <https://www.tandfonline.com/doi/abs/10.1080/14786440409463107> (cit. on p. 86).

[287] *Asymptotics for Minimal Discrete Energy on the Sphere* on JSTOR. URL: <https://www.jstor.org/stable/117605> (cit. on p. 87).

[288] *The minimum of potential energy of a System of point charges*. URL: <https://www.degruyter.com/document/doi/10.1515/dma.1993.3.1.75/html?lang=en> (cit. on p. 87).

[289] Weiyang Liu et al. *Learning with Hyperspherical Uniformity*. Nov. 16, 2021. arXiv: [2103.01649\[cs\]](https://arxiv.org/abs/2103.01649). URL: <http://arxiv.org/abs/2103.01649> (cit. on p. 88).

[290] R. J. Beran. “Testing for uniformity on a compact homogeneous space”. In: *Journal of Applied Probability* 5.1 (Apr. 1968). Publisher: Cambridge University Press, pp. 177–195. ISSN: 0021-9002, 1475-6072. DOI: [10.2307/3212085](https://doi.org/10.2307/3212085). URL: [https://www.cambridge.org/core/journals/journal-of-applied-probability/article/abs/testing-for-uniformity-on-a-compact-homogeneous-space/1F98A87CC8BC7DE51D2C55D6CD66AA86](https://www.cambridge.org/core/journals/journal-of-applied-probability/article/testing-for-uniformity-on-a-compact-homogeneous-space/1F98A87CC8BC7DE51D2C55D6CD66AA86) (cit. on p. 88).

[291] *A New Sobolev Test for Uniformity on the Circle* on JSTOR. URL: <https://www.jstor.org/stable/2336748> (cit. on p. 88).

[292] Sreenivasa Rao Jammalamadaka, Simos Meintanis, and Thomas Verdebout. “On Sobolev tests of uniformity on the circle with an extension to the sphere”. In: *Bernoulli* 26.3 (Aug. 2020). Publisher: Bernoulli Society for Mathematical Statistics and Probability, pp. 2226–2252. ISSN: 1350-7265. DOI: [10.3150/19-BEJ1191](https://doi.org/10.3150/19-BEJ1191). URL: <https://projecteuclid.org/journals/bernoulli/volume-26/issue-3/On-Sobolev-tests-of-uniformity-on-the-circle-with-an/10.3150/19-BEJ1191.full> (cit. on p. 88).

[293] B. AJNE. “A simple test for uniformity of a circular distribution”. In: *Biometrika* 55.2 (July 1, 1968), pp. 343–354. ISSN: 0006-3444. DOI: [10.1093/biomet/55.2.343](https://doi.org/10.1093/biomet/55.2.343). URL: <https://doi.org/10.1093/biomet/55.2.343> (cit. on p. 88).

[294] Eduardo García-Portugués and Thomas Verdebout. *An overview of uniformity tests on the hypersphere*. Apr. 3, 2018. arXiv: [1804.00286 \[stat\]](https://arxiv.org/abs/1804.00286). URL: [http://arxiv.org/abs/1804.00286](https://arxiv.org/abs/1804.00286) (cit. on p. 88).

[295] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. Jan. 29, 2017. DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980). arXiv: [1412.6980 \[cs\]](https://arxiv.org/abs/1412.6980). URL: [http://arxiv.org/abs/1412.6980](https://arxiv.org/abs/1412.6980) (cit. on p. 90).

[296] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. *On Aliased Resizing and Surprising Subtleties in GAN Evaluation*. Jan. 20, 2022. DOI: [10.48550/arXiv.2104.11222](https://doi.org/10.48550/arXiv.2104.11222). arXiv: [2104.11222 \[cs\]](https://arxiv.org/abs/2104.11222). URL: [http://arxiv.org/abs/2104.11222](https://arxiv.org/abs/2104.11222) (cit. on pp. 90, 141).

[297] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. Dec. 11, 2015. DOI: [10.48550/arXiv.1512.00567](https://doi.org/10.48550/arXiv.1512.00567). arXiv: [1512.00567 \[cs\]](https://arxiv.org/abs/1512.00567). URL: [http://arxiv.org/abs/1512.00567](https://arxiv.org/abs/1512.00567) (cit. on p. 90).

[298] [PDF] *An empirical study on evaluation metrics of generative adversarial networks / Semantic Scholar*. URL: <https://www.semanticscholar.org/paper/An-empirical-study-on-evaluation-metrics-of-Xu-Huang/a476f45d867c9a2f953c44e14fec35a6e6af27a0> (cit. on p. 91).

[299] Nathan Mantel. “The Detection of Disease Clustering and a Generalized Regression Approach”. In: *Cancer Research* 27.2 (Feb. 1, 1967), pp. 209–220. ISSN: 0008-5472 (cit. on p. 93).

[300] *Biometry. The Principles and Practice of Statistics in Biological Research; Statistical Tables / Systematic Biology / Oxford Academic*. URL: <https://academic.oup.com/sysbio/article-abstract/19/4/391/1648697?redirectedFrom=fulltext> (cit. on p. 93).

[301] *Robust Estimation of a Location Parameter on JSTOR*. URL: <https://www.jstor.org/stable/2238020> (cit. on p. 101).

[302] Kajal Eybpoosh, Mansoor Rezghi, and Abbas Heydari. “Applying inverse stereographic projection to manifold learning and clustering”. In: *Applied Intelligence* 52.4 (Mar. 1, 2022), pp. 4443–4457. ISSN: 1573-7497. DOI: [10.1007/s10489-021-02513-0](https://doi.org/10.1007/s10489-021-02513-0). URL: <https://doi.org/10.1007/s10489-021-02513-0> (cit. on p. 102).

[303] Dan Hendrycks and Kevin Gimpel. *Gaussian Error Linear Units (GELUs)*. June 5, 2023. DOI: [10.48550/arXiv.1606.08415](https://doi.org/10.48550/arXiv.1606.08415). arXiv: [1606.08415 \[cs\]](https://arxiv.org/abs/1606.08415). URL: [http://arxiv.org/abs/1606.08415](https://arxiv.org/abs/1606.08415) (cit. on p. 102).

[304] *Frontiers / Novel 3D Pixel Sensors for the Upgrade of the ATLAS Inner Tracker*. URL: <https://www.frontiersin.org/articles/10.3389/fphy.2021.624668/full> (cit. on pp. 104, 181).

[305] Kevin Pedro. “Current and Future Performance of the CMS Simulation”. In: *EPJ Web of Conferences* 214 (2019). Publisher: EDP Sciences, p. 02036. ISSN: 2100-014X. DOI: [10.1051/epjconf/201921402036](https://doi.org/10.1051/epjconf/201921402036). URL: https://www.epj-conferences.org/articles/epjconf/abs/2019/19/epjconf_chep2018_02036/epjconf_chep2018_02036.html (cit. on pp. 104, 181).

[306] Yinhai Liu et al. *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. July 26, 2019. DOI: [10.48550/arXiv.1907.11692](https://doi.org/10.48550/arXiv.1907.11692) [cs]. URL: [http://arxiv.org/abs/1907.11692](https://arxiv.org/abs/1907.11692) (cit. on p. 105).

[307] Zhilin Yang et al. *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Jan. 2, 2020. DOI: [10.48550/arXiv.1906.08237](https://doi.org/10.48550/arXiv.1906.08237). arXiv: [1906.08237](https://arxiv.org/abs/1906.08237) [cs]. URL: [http://arxiv.org/abs/1906.08237](https://arxiv.org/abs/1906.08237) (cit. on p. 105).

[308] Yasir Alanazi et al. “A survey of machine learning-based physics event generation”. In: *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*. Aug. 2021, pp. 4286–4293. DOI: [10.24963/ijcai.2021/588](https://doi.org/10.24963/ijcai.2021/588). arXiv: [2106.00643](https://arxiv.org/abs/2106.00643) [hep-ph, physics:nucl-ex]. URL: [http://arxiv.org/abs/2106.00643](https://arxiv.org/abs/2106.00643) (cit. on p. 108).

[309] Sanjeev Arora et al. *Generalization and Equilibrium in Generative Adversarial Nets (GANs)*. Aug. 1, 2017. DOI: [10.48550/arXiv.1703.00573](https://doi.org/10.48550/arXiv.1703.00573). arXiv: [1703.00573](https://arxiv.org/abs/1703.00573) [cs, stat]. URL: [http://arxiv.org/abs/1703.00573](https://arxiv.org/abs/1703.00573) (cit. on p. 109).

[310] Keyulu Xu et al. *How Neural Networks Extrapolate: From Feedforward to Graph Neural Networks*. Mar. 2, 2021. DOI: [10.48550/arXiv.2009.11848](https://doi.org/10.48550/arXiv.2009.11848). arXiv: [2009.11848](https://arxiv.org/abs/2009.11848) [cs, stat]. URL: [http://arxiv.org/abs/2009.11848](https://arxiv.org/abs/2009.11848) (cit. on p. 109).

[311] Ruili Feng et al. *Principled Knowledge Extrapolation with GANs*. May 21, 2022. DOI: [10.48550/arXiv.2205.13444](https://doi.org/10.48550/arXiv.2205.13444). arXiv: [2205.13444](https://arxiv.org/abs/2205.13444) [cs]. URL: [http://arxiv.org/abs/2205.13444](https://arxiv.org/abs/2205.13444) (cit. on p. 109).

[312] Murat Kocaoglu et al. *CausalGAN: Learning Causal Implicit Generative Models with Adversarial Training*. Sept. 14, 2017. DOI: [10.48550/arXiv.1709.02023](https://doi.org/10.48550/arXiv.1709.02023). arXiv: [1709.02023](https://arxiv.org/abs/1709.02023) [cs, math, stat]. URL: [http://arxiv.org/abs/1709.02023](https://arxiv.org/abs/1709.02023) (cit. on p. 109).

[313] Mohammad Lotfollahi et al. “Conditional out-of-distribution generation for unpaired data using transfer VAE”. In: *Bioinformatics* 36 (Supplement_2 Dec. 30, 2020), pp. i610–i617. ISSN: 1367-4803. DOI: [10.1093/bioinformatics/btaa800](https://doi.org/10.1093/bioinformatics/btaa800). URL: <https://doi.org/10.1093/bioinformatics/btaa800> (cit. on pp. 109, 182).

[314] Seul Lee, Jaehyeong Jo, and Sung Ju Hwang. “Exploring Chemical Space with Score-based Out-of-distribution Generation”. In: (Sept. 29, 2022). URL: <https://openreview.net/forum?id=45TeQUJw9tn> (cit. on p. 109).

[315] *De novo Molecular Design / Wiley Online Books*. URL: <https://onlinelibrary.wiley.com/doi/book/10.1002/9783527677016> (cit. on p. 109).

[316] Joshua Meyers, Benedek Fabian, and Nathan Brown. “De novo molecular design and generative models”. In: *Drug Discovery Today* 26.11 (Nov. 2021), pp. 2707–2715. ISSN: 1878-5832. DOI: [10.1016/j.drudis.2021.05.019](https://doi.org/10.1016/j.drudis.2021.05.019) (cit. on p. 109).

[317] Hongming Chen et al. “The rise of deep learning in drug discovery”. In: *Drug Discovery Today* 23.6 (June 1, 2018), pp. 1241–1250. ISSN: 1359-6446. DOI: [10.1016/j.drudis.2018.01.039](https://doi.org/10.1016/j.drudis.2018.01.039). URL: <https://www.sciencedirect.com/science/article/pii/S1359644617303598> (cit. on p. 109).

[318] Adam R. Kosiorek, Hyunjik Kim, and Danilo J. Rezende. *Conditional Set Generation with Transformers*. July 1, 2020. DOI: [10.48550/arXiv.2006.16841](https://doi.org/10.48550/arXiv.2006.16841). arXiv: [2006.16841\[cs\]](https://arxiv.org/abs/2006.16841). URL: [http://arxiv.org/abs/2006.16841](https://arxiv.org/abs/2006.16841) (cit. on pp. 111, 113, 145).

[319] Shitong Luo and Wei Hu. *Diffusion Probabilistic Models for 3D Point Cloud Generation*. June 13, 2021. DOI: [10.48550/arXiv.2103.01458](https://doi.org/10.48550/arXiv.2103.01458). arXiv: [2103.01458\[cs\]](https://arxiv.org/abs/2103.01458). URL: [http://arxiv.org/abs/2103.01458](https://arxiv.org/abs/2103.01458) (cit. on p. 111).

[320] Hongwei Wang et al. *GraphGAN: Graph Representation Learning with Generative Adversarial Nets*. Nov. 22, 2017. arXiv: [1711.08267\[cs, stat\]](https://arxiv.org/abs/1711.08267). URL: [http://arxiv.org/abs/1711.08267](https://arxiv.org/abs/1711.08267) (cit. on p. 112).

[321] Igor Krawczuk et al. “GG-GAN: A Geometric Graph Generative Adversarial Network”. In: (Oct. 2, 2020). URL: <https://openreview.net/forum?id=qiAxL3Xqx1o> (cit. on p. 112).

[322] Lily H. Zhang, Mark Goldstein, and Rajesh Ranganath. *Understanding Failures in Out-of-Distribution Detection with Deep Generative Models*. July 16, 2021. DOI: [10.48550/arXiv.2107.06908](https://doi.org/10.48550/arXiv.2107.06908). arXiv: [2107.06908\[cs\]](https://arxiv.org/abs/2107.06908). URL: [http://arxiv.org/abs/2107.06908](https://arxiv.org/abs/2107.06908) (cit. on p. 112).

[323] *DeepFunc: A Deep Learning Framework for Accurate Prediction of Protein Functions from Protein Sequences and Interactions - Zhang - 2019 - PROTEOMICS - Wiley Online Library*. URL: <https://onlinelibrary.wiley.com/doi/10.1002/pmic.201900019> (cit. on p. 112).

[324] Beatrice Bevilacqua, Yangze Zhou, and Bruno Ribeiro. *Size-Invariant Graph Representations for Graph Classification Extrapolations*. July 6, 2021. arXiv: [2103.05045\[cs\]](https://arxiv.org/abs/2103.05045). URL: [http://arxiv.org/abs/2103.05045](https://arxiv.org/abs/2103.05045) (cit. on p. 113).

[325] Ofir Press, Noah A. Smith, and Mike Lewis. *Train Short, Test Long: Attention with Linear Biases Enables Input Length Extrapolation*. Apr. 22, 2022. DOI: [10.48550/arXiv.2108.12409](https://doi.org/10.48550/arXiv.2108.12409). arXiv: [2108.12409\[cs\]](https://arxiv.org/abs/2108.12409). URL: [http://arxiv.org/abs/2108.12409](https://arxiv.org/abs/2108.12409) (cit. on pp. 113, 115, 116, 135).

[326] Yan Zhang. *Deep Set Prediction Networks*. original-date: 2019-06-14T06:06:09Z. June 3, 2023. URL: <https://github.com/Cyanogenoid/dspn> (cit. on p. 113).

[327] Mateusz Jurewicz and Leon Strømberg-Derczynski. “Set-to-Sequence Methods in Machine Learning: a Review”. In: *Journal of Artificial Intelligence Research* 71 (Aug. 12, 2021), pp. 885–924. ISSN: 1076-9757. DOI: [10.1613/jair.1.12839](https://doi.org/10.1613/jair.1.12839). arXiv: [2103.09656\[cs\]](https://arxiv.org/abs/2103.09656). URL: <http://arxiv.org/abs/2103.09656> (cit. on p. 113).

[328] Jinwoo Kim, Saeyoon Oh, and Seunghoon Hong. *Transformers Generalize DeepSets and Can be Extended to Graphs and Hypergraphs*. Jan. 21, 2022. DOI: [10.48550/arXiv.2110.14416](https://doi.org/10.48550/arXiv.2110.14416). arXiv: [2110.14416\[cs\]](https://arxiv.org/abs/2110.14416). URL: <http://arxiv.org/abs/2110.14416> (cit. on p. 113).

[329] Santiago Ontanon et al. “Making Transformers Solve Compositional Tasks”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. ACL 2022. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 3591–3607. DOI: [10.18653/v1/2022.acl-long.251](https://doi.org/10.18653/v1/2022.acl-long.251). URL: <https://aclanthology.org/2022.acl-long.251> (cit. on p. 113).

[330] Tom B. Brown et al. *Language Models are Few-Shot Learners*. July 22, 2020. DOI: [10.48550/arXiv.2005.14165](https://doi.org/10.48550/arXiv.2005.14165). arXiv: [2005.14165\[cs\]](https://arxiv.org/abs/2005.14165). URL: <http://arxiv.org/abs/2005.14165> (cit. on p. 113).

[331] Susan Zhang et al. *OPT: Open Pre-trained Transformer Language Models*. June 21, 2022. DOI: [10.48550/arXiv.2205.01068](https://doi.org/10.48550/arXiv.2205.01068). arXiv: [2205.01068\[cs\]](https://arxiv.org/abs/2205.01068). URL: <http://arxiv.org/abs/2205.01068> (cit. on p. 113).

[332] Colin Raffel et al. *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer*. July 28, 2020. arXiv: [1910.10683\[cs, stat\]](https://arxiv.org/abs/1910.10683). URL: <http://arxiv.org/abs/1910.10683> (cit. on p. 114).

[333] Aakanksha Chowdhery et al. *PaLM: Scaling Language Modeling with Pathways*. Oct. 5, 2022. DOI: [10.48550/arXiv.2204.02311](https://doi.org/10.48550/arXiv.2204.02311). arXiv: [2204.02311\[cs\]](https://arxiv.org/abs/2204.02311). URL: <http://arxiv.org/abs/2204.02311> (cit. on p. 115).

[334] Hugo Touvron et al. *LLaMA: Open and Efficient Foundation Language Models*. Feb. 27, 2023. DOI: [10.48550/arXiv.2302.13971](https://doi.org/10.48550/arXiv.2302.13971). arXiv: [2302.13971\[cs\]](https://arxiv.org/abs/2302.13971). URL: <http://arxiv.org/abs/2302.13971> (cit. on p. 115).

[335] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. Aug. 8, 2022. DOI: [10.48550/arXiv.2104.09864](https://doi.org/10.48550/arXiv.2104.09864). arXiv: [2104.09864\[cs\]](https://arxiv.org/abs/2104.09864). URL: <http://arxiv.org/abs/2104.09864> (cit. on p. 115).

[336] BigScience Workshop et al. *BLOOM: A 176B-Parameter Open-Access Multilingual Language Model*. June 27, 2023. DOI: [10.48550/arXiv.2211.05100](https://doi.org/10.48550/arXiv.2211.05100). arXiv: [2211.05100\[cs\]](https://arxiv.org/abs/2211.05100). URL: <http://arxiv.org/abs/2211.05100> (cit. on p. 115).

[337] Yutao Sun et al. *A Length-Extrapolatable Transformer*. version: 1. Dec. 20, 2022. DOI: [10.48550/arXiv.2212.10554](https://doi.org/10.48550/arXiv.2212.10554). arXiv: [2212.10554\[cs\]](https://arxiv.org/abs/2212.10554). URL: <http://arxiv.org/abs/2212.10554> (cit. on pp. 115, 116).

[338] Chase R Freschlin, Sarah A Fahlberg, and Philip A Romero. “Machine learning to navigate fitness landscapes for protein engineering”. In: *Current Opinion in Biotechnology* 75 (June 1, 2022), p. 102713. ISSN: 0958-1669. DOI: [10.1016/j.copbi.2022.102713](https://doi.org/10.1016/j.copbi.2022.102713). URL: <https://www.sciencedirect.com/science/article/pii/S0958166922000465> (cit. on pp. 116, 182).

[339] Andy Hsien-Wei Yeh et al. “De novo design of luciferases using deep learning”. In: *Nature* 614.7949 (Feb. 2023). Number: 7949 Publisher: Nature Publishing Group, pp. 774–780. ISSN: 1476-4687. DOI: [10.1038/s41586-023-05696-3](https://doi.org/10.1038/s41586-023-05696-3). URL: <https://www.nature.com/articles/s41586-023-05696-3> (cit. on p. 116).

[340] Pablo Gainza et al. “De novo design of protein interactions with learned surface fingerprints”. In: *Nature* 617.7959 (May 2023). Number: 7959 Publisher: Nature Publishing Group, pp. 176–184. ISSN: 1476-4687. DOI: [10.1038/s41586-023-0593-x](https://doi.org/10.1038/s41586-023-0593-x). URL: <https://www.nature.com/articles/s41586-023-05993-x> (cit. on p. 116).

[341] Nitish Shirish Keskar et al. *CTRL: A Conditional Transformer Language Model for Controllable Generation*. Sept. 20, 2019. DOI: [10.48550/arXiv.1909.05858](https://doi.org/10.48550/arXiv.1909.05858). arXiv: [1909.05858\[cs\]](https://arxiv.org/abs/1909.05858). URL: [http://arxiv.org/abs/1909.05858](https://arxiv.org/abs/1909.05858) (cit. on p. 116).

[342] *ProGen: Language Modeling for Protein Generation* / bioRxiv. URL: <https://www.biorxiv.org/content/10.1101/2020.03.07.982272v2> (cit. on p. 116).

[343] Kelvin Guu et al. *Generating Sentences by Editing Prototypes*. Sept. 7, 2018. DOI: [10.48550/arXiv.1709.08878](https://doi.org/10.48550/arXiv.1709.08878). arXiv: [1709.08878\[cs, stat\]](https://arxiv.org/abs/1709.08878). URL: [http://arxiv.org/abs/1709.08878](https://arxiv.org/abs/1709.08878) (cit. on p. 116).

[344] Jonathan Mallinson et al. *EditT5: Semi-Autoregressive Text-Editing with T5 Warm-Start*. Oct. 26, 2022. DOI: [10.48550/arXiv.2205.12209](https://doi.org/10.48550/arXiv.2205.12209). arXiv: [2205.12209\[cs\]](https://arxiv.org/abs/2205.12209). URL: [http://arxiv.org/abs/2205.12209](https://arxiv.org/abs/2205.12209) (cit. on p. 116).

[345] Roman Novak, Michael Auli, and David Grangier. *Iterative Refinement for Machine Translation*. Apr. 13, 2018. DOI: [10.48550/arXiv.1610.06602](https://doi.org/10.48550/arXiv.1610.06602). arXiv: [1610.06602\[cs\]](https://arxiv.org/abs/1610.06602). URL: [http://arxiv.org/abs/1610.06602](https://arxiv.org/abs/1610.06602) (cit. on p. 116).

[346] Sean Welleck et al. *Generating Sequences by Learning to Self-Correct*. Oct. 31, 2022. DOI: [10.48550/arXiv.2211.00053](https://doi.org/10.48550/arXiv.2211.00053). arXiv: [2211.00053\[cs\]](https://arxiv.org/abs/2211.00053). URL: [http://arxiv.org/abs/2211.00053](https://arxiv.org/abs/2211.00053) (cit. on p. 116).

[347] Vishakh Padmakumar et al. *Extrapolative Controlled Sequence Generation via Iterative Refinement*. June 7, 2023. DOI: [10.48550/arXiv.2303.04562](https://doi.org/10.48550/arXiv.2303.04562). arXiv: [2303.04562\[cs, q-bio\]](https://arxiv.org/abs/2303.04562). URL: [http://arxiv.org/abs/2303.04562](https://arxiv.org/abs/2303.04562) (cit. on p. 116).

[348] Alvin Chan et al. *Deep Extrapolation for Attribute-Enhanced Generation*. Oct. 25, 2021. DOI: [10.48550/arXiv.2107.02968](https://doi.org/10.48550/arXiv.2107.02968). arXiv: [2107.02968\[cs, q-bio\]](https://arxiv.org/abs/2107.02968). URL: [http://arxiv.org/abs/2107.02968](https://arxiv.org/abs/2107.02968) (cit. on pp. 117, 182).

[349] Sumanth Dathathri et al. *Plug and Play Language Models: A Simple Approach to Controlled Text Generation*. Mar. 3, 2020. DOI: [10.48550/arXiv.1912.02164](https://doi.org/10.48550/arXiv.1912.02164). arXiv: [1912.02164\[cs\]](https://arxiv.org/abs/1912.02164). URL: <http://arxiv.org/abs/1912.02164> (cit. on p. 117).

[350] Kevin Yang and Dan Klein. “FUDGE: Controlled Text Generation With Future Discriminators”. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 3511–3535. DOI: [10.18653/v1/2021.naacl-main.276](https://doi.org/10.18653/v1/2021.naacl-main.276). arXiv: [2104.05218\[cs\]](https://arxiv.org/abs/2104.05218). URL: <http://arxiv.org/abs/2104.05218> (cit. on p. 117).

[351] *Denoising Diffusion-based Generative Modeling: Foundations and Applications*. Denoising Diffusion-based Generative Modeling: Foundations and Applications. URL: <https://cvpr2022-tutorial-diffusion-models.github.io> (cit. on p. 117).

[352] Hongyu Gong et al. “Reinforcement Learning Based Text Style Transfer without Parallel Training Corpus”. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. NAACL-HLT 2019. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 3168–3180. DOI: [10.18653/v1/N19-1320](https://doi.org/10.18653/v1/N19-1320). URL: <https://aclanthology.org/N19-1320> (cit. on p. 117).

[353] Christof Angermueller et al. “Model-based reinforcement learning for biological sequence design”. In: Eighth International Conference on Learning Representations. Apr. 2020. URL: https://iclr.cc/virtual_2020/poster_HklxbgBKvr.html (cit. on p. 117).

[354] Dario Amodei et al. *Concrete Problems in AI Safety*. July 25, 2016. DOI: [10.48550/arXiv.1606.06565](https://doi.org/10.48550/arXiv.1606.06565). arXiv: [1606.06565\[cs\]](https://arxiv.org/abs/1606.06565). URL: <http://arxiv.org/abs/1606.06565> (cit. on p. 117).

[355] Eric J. Michaud, Adam Gleave, and Stuart Russell. *Understanding Learned Reward Functions*. Dec. 10, 2020. DOI: [10.48550/arXiv.2012.05862](https://doi.org/10.48550/arXiv.2012.05862). arXiv: [2012.05862\[cs\]](https://arxiv.org/abs/2012.05862). URL: <http://arxiv.org/abs/2012.05862> (cit. on p. 117).

[356] Richard Yuanzhe Pang et al. *Reward Gaming in Conditional Text Generation*. June 1, 2023. DOI: [10.48550/arXiv.2211.08714](https://doi.org/10.48550/arXiv.2211.08714). arXiv: [2211.08714\[cs\]](https://arxiv.org/abs/2211.08714). URL: <http://arxiv.org/abs/2211.08714> (cit. on p. 117).

[357] Samuel Eilenberg and SAUNDERS MacLANE. “GENERAL THEORY OF NATURAL EQUIVALENCES”. In: () (cit. on p. 117).

[358] Pim de Haan, Taco S Cohen, and Max Welling. “Natural Graph Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 3636–3646. URL: <https://proceedings.neurips.cc/paper/2020/hash/2517756c5a9be6ac007fe9bb7fb92611-Abstract.html> (cit. on pp. 117, 119).

[359] Andrew Dudzik and Petar Veličković. *Graph Neural Networks are Dynamic Programmers*. Oct. 10, 2022. DOI: [10.48550/arXiv.2203.15544](https://doi.org/10.48550/arXiv.2203.15544). arXiv: [2203.15544\[cs,math,stat\]](https://arxiv.org/abs/2203.15544). URL: <http://arxiv.org/abs/2203.15544> (cit. on pp. 117, 119).

[360] Euan Ong and Petar Veličković. *Learnable Commutative Monoids for Graph Neural Networks*. Dec. 16, 2022. arXiv: [2212.08541\[cs, stat\]](https://arxiv.org/abs/2212.08541). URL: <http://arxiv.org/abs/2212.08541> (cit. on p. 119).

[361] Adam Nemecek. *Coinductive guide to inductive transformer heads*. Feb. 3, 2023. arXiv: [2302.01834\[cs\]](https://arxiv.org/abs/2302.01834). URL: <http://arxiv.org/abs/2302.01834> (cit. on p. 119).

[362] Barry Mazur. “When is one thing equal to some other thing?” In: () (cit. on p. 119).

[363] Toan Q. Nguyen and Julian Salazar. “Transformers without Tears: Improving the Normalization of Self-Attention”. In: (Nov. 2, 2019). DOI: [10.5281/zenodo.3525484](https://doi.org/10.5281/zenodo.3525484). arXiv: [1910.05895\[cs, stat\]](https://arxiv.org/abs/1910.05895). URL: <http://arxiv.org/abs/1910.05895> (cit. on p. 124).

[364] Sergey Ioffe and Christian Szegedy. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Mar. 2, 2015. DOI: [10.48550/arXiv.1502.03167](https://doi.org/10.48550/arXiv.1502.03167). arXiv: [1502.03167\[cs\]](https://arxiv.org/abs/1502.03167). URL: <http://arxiv.org/abs/1502.03167> (cit. on p. 124).

[365] Wouter Kool, Herke van Hoof, and Max Welling. *Attention, Learn to Solve Routing Problems!* Feb. 7, 2019. DOI: [10.48550/arXiv.1803.08475](https://doi.org/10.48550/arXiv.1803.08475). arXiv: [1803.08475\[cs, stat\]](https://arxiv.org/abs/1803.08475). URL: <http://arxiv.org/abs/1803.08475> (cit. on p. 124).

[366] Shibani Santurkar et al. *How Does Batch Normalization Help Optimization?* Apr. 14, 2019. DOI: [10.48550/arXiv.1805.11604](https://doi.org/10.48550/arXiv.1805.11604). arXiv: [1805.11604\[cs, stat\]](https://arxiv.org/abs/1805.11604). URL: <http://arxiv.org/abs/1805.11604> (cit. on p. 124).

[367] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. *A Survey on Oversmoothing in Graph Neural Networks*. Mar. 20, 2023. arXiv: [2303.10993\[cs\]](https://arxiv.org/abs/2303.10993). URL: <http://arxiv.org/abs/2303.10993> (cit. on p. 125).

[368] Sepp Hochreiter and Jürgen Schmidhuber. “Long Short-Term Memory”. In: *Neural Computation* 9.8 (Nov. 1, 1997), pp. 1735–1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735). URL: <https://doi.org/10.1162/neco.1997.9.8.1735> (cit. on p. 125).

[369] Gabriele Corso et al. *Principal Neighbourhood Aggregation for Graph Nets*. Dec. 31, 2020. DOI: [10.48550/arXiv.2004.05718](https://doi.org/10.48550/arXiv.2004.05718). arXiv: [2004.05718\[cs, stat\]](https://arxiv.org/abs/2004.05718). URL: <http://arxiv.org/abs/2004.05718> (cit. on pp. 125, 126, 172).

[370] Ethan Perez et al. *FiLM: Visual Reasoning with a General Conditioning Layer*. Dec. 18, 2017. DOI: [10.48550/arXiv.1709.07871](https://doi.org/10.48550/arXiv.1709.07871). arXiv: [1709.07871\[cs, stat\]](https://arxiv.org/abs/1709.07871). URL: <http://arxiv.org/abs/1709.07871> (cit. on p. 134).

[371] Jean Feydy. “Geometric data analysis, beyond convolutions”. In: () (cit. on p. 135).

[372] Philip A. Knight. “The Sinkhorn–Knopp Algorithm: Convergence and Applications”. In: *SIAM Journal on Matrix Analysis and Applications* 30.1 (Jan. 2008). Publisher: Society for Industrial and Applied Mathematics, pp. 261–275. ISSN: 0895-4798. DOI: [10.1137/060659624](https://doi.org/10.1137/060659624). URL: <https://pubs.siam.org/doi/10.1137/060659624> (cit. on p. 136).

[373] Arthur Gretton et al. “A Kernel Method for the Two-Sample-Problem”. In: *Advances in Neural Information Processing Systems*. Vol. 19. MIT Press, 2006. URL: https://proceedings.neurips.cc/paper_files/paper/2006/hash/e9fb2eda3d9c55a0d89c98d6c54b5b3e-Abstract.html (cit. on p. 136).

[374] Yanna Bai et al. “Deep learning methods for solving linear inverse problems: Research directions and paradigms”. In: *Signal Processing* 177 (Dec. 1, 2020), p. 107729. ISSN: 0165-1684. DOI: [10.1016/j.sigpro.2020.107729](https://doi.org/10.1016/j.sigpro.2020.107729). URL: <https://www.sciencedirect.com/science/article/pii/S0165168420302723> (cit. on p. 139).

[375] Derek Bingham, Troy Butler, and Don Estep. “Inverse Problems for Physics-Based Process Models”. In: *Annual Review of Statistics and Its Application* 11.1 (2024). _eprint: <https://doi.org/10.1146/annurev-statistics-031017-100108>, null. DOI: [10.1146/annurev-statistics-031017-100108](https://doi.org/10.1146/annurev-statistics-031017-100108). URL: <https://doi.org/10.1146/annurev-statistics-031017-100108> (cit. on p. 139).

[376] Frédéric Chazal and Bertrand Michel. *An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists*. arXiv.org. Oct. 11, 2017. URL: <https://arxiv.org/abs/1710.04019v2> (cit. on pp. 150, 156).

[377] Chad M. Topaz, Lori Ziegelmeier, and Tom Halverson. “Topological Data Analysis of Biological Aggregation Models”. In: *PLOS ONE* 10.5 (May 13, 2015). Ed. by Bard Ermentrout, e0126383. ISSN: 1932-6203. DOI: [10.1371/journal.pone.0126383](https://doi.org/10.1371/journal.pone.0126383). URL: <https://dx.plos.org/10.1371/journal.pone.0126383> (cit. on p. 154).

[378] Matteo Rucco et al. “Characterisation of the Idiotypic Immune Network Through Persistent Entropy”. In: *Proceedings of ECCS 2014*. Ed. by Stefano Battiston et al. Springer Proceedings in Complexity. Cham: Springer International Publishing, 2016, pp. 117–128. ISBN: 978-3-319-29228-1. DOI: [10.1007/978-3-319-29228-1_11](https://doi.org/10.1007/978-3-319-29228-1_11) (cit. on p. 158).

[379] Nieves Atienza, Rocio Gonzalez-Diaz, and Matteo Rucco. *Persistent Entropy for Separating Topological Features from Noise in Vietoris-Rips Complexes*. Jan. 18, 2017. arXiv: [1701.07857\[cs\]](https://arxiv.org/abs/1701.07857). URL: [http://arxiv.org/abs/1701.07857](https://arxiv.org/abs/1701.07857) (cit. on p. 158).

[380] Po-Ssu Huang, Scott E. Boyken, and David Baker. “The coming of age of de novo protein design”. In: *Nature* 537.7620 (Sept. 2016). Number: 7620 Publisher: Nature Publishing Group, pp. 320–327. ISSN: 1476-4687. DOI: [10.1038/nature19946](https://doi.org/10.1038/nature19946). URL: <https://www.nature.com/articles/nature19946> (cit. on p. 182).

[381] Shitong Luo and Wei Hu. “Differentiable Manifold Reconstruction for Point Cloud Denoising”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. Oct. 12, 2020, pp. 1330–1338. DOI: [10.1145/3394171.3413727](https://doi.org/10.1145/3394171.3413727). arXiv: [2007.13551\[cs\]](https://arxiv.org/abs/2007.13551). URL: [http://arxiv.org/abs/2007.13551](https://arxiv.org/abs/2007.13551) (cit. on p. 182).

[382] Kwangho Kim et al. *PLlay: Efficient Topological Layer based on Persistence Landscapes*. arXiv.org. Feb. 7, 2020. URL: <https://arxiv.org/abs/2002.02778v4> (cit. on p. 182).

- [383] Yujia Xie et al. “Differentiable Top-k with Optimal Transport”. In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 20520–20531. URL: <https://papers.nips.cc/paper/2020/hash/ec24a54d62ce57ba93a531b460fa8d18-Abstract.html> (cit. on p. 182).
- [384] Yuanfeng Ji et al. *DrugOOD: Out-of-Distribution (OOD) Dataset Curator and Benchmark for AI-aided Drug Discovery – A Focus on Affinity Prediction Problems with Noise Annotations*. arXiv.org. Jan. 24, 2022. URL: <https://arxiv.org/abs/201.09637v1> (cit. on p. 182).
- [385] Joseph L. Watson et al. “De novo design of protein structure and function with RFdiffusion”. In: *Nature* 620.7976 (Aug. 2023). Number: 7976 Publisher: Nature Publishing Group, pp. 1089–1100. ISSN: 1476-4687. DOI: [10.1038/s41586-023-06415-8](https://doi.org/10.1038/s41586-023-06415-8). URL: <https://www.nature.com/articles/s41586-023-06415-8> (cit. on p. 182).
- [386] Andreas Adelmann et al. *New directions for surrogate models and differentiable programming for High Energy Physics detector simulation*. Mar. 15, 2022. arXiv: [2203.08806\[hep-ex,physics:hep-ph,physics:physics\]](https://arxiv.org/abs/2203.08806). URL: [http://arxiv.org/abs/2203.08806](https://arxiv.org/abs/2203.08806) (cit. on p. 182).
- [387] Atilim Gunes Baydin et al. *Automatic differentiation in machine learning: a survey*. Feb. 5, 2018. DOI: [10.48550/arXiv.1502.05767](https://doi.org/10.48550/arXiv.1502.05767). arXiv: [1502.05767\[cs,stat\]](https://arxiv.org/abs/1502.05767). URL: [http://arxiv.org/abs/1502.05767](https://arxiv.org/abs/1502.05767) (cit. on p. 182).

Acknowledgements

This research was supported by the collaborative project IDT-UM (Innovative Digitale Technologien zur Erforschung von Universum und Materie) funded by the German Federal Ministry of Education and Research (BMBF) and the Deutsche Forschungsgemeinschaft under Germany's Excellence Strategy – EXC 2094 "ORIGINS" – 390783311.

This thesis is the capstone of four of the most adventurous years of my life. I learned a lot and gathered many unique experiences on this journey. None of this would have been possible without the assistance of a myriad of amazing individuals. While I may not be able to thank each and every one of them, I will certainly try.

First of all, I would like to thank Prof. Dr. Thomas Kuhr for his unwavering support and for sharing his expertise in detector physics and the exciting field of flavor physics. I vividly recall the day you introduced me to this subject matter. You outlined several potential topics, and when I asked which one was more challenging, you pointed to the PXD project. Thank you sincerely for allowing me the freedom to explore a multitude of different topics, even those outside our designated project.

I would like to extend my sincerest gratitude to Prof. Dr. Lukas Heinrich, Prof. Dr. Volker Tresp, Prof. Dr. Stefen Rullands, Prof. Dr. Daniel Gruen, and Prof. Dr. Lode Pollet for being part of my examination committee.

Dr. Nikolai Hartmann, thank you for generously dedicating your time to me and offering support at literally any hour of the week. Our discussions have always been vibrant and a synthesis of ideas. Your live coding sessions were truly epic, and I learned a lot from them. I'm also deeply grateful for your thorough review and insightful comments on this thesis.

Dr. Sahand Sharifzadeh, thank you for supporting and believing in me. I have always learned new things from our discussions. Your insights in creating new narratives and out-of-the-box thinking were always inspirational for me.

Dr. James Kahn, we have not met each other in person even once, but we had numerous online meetings that took hours discussing different ideas in both the LinkFEI and IEA-GAN projects. I learned many tricks and academic hacks from you.

I thank Dr. Thomas Lueck for his technical and insightful comments about the PXD problem. Thank you for helping me brush up on the first chapter of this thesis!

I thank my colleagues from the Computational Center for Particle and Astrophysics (C2PAP), who provided expertise and computation power that greatly assisted the research.

I also thank David Katheder for his assistance with the GitHub repository preparation and cleaning of my code.

Dr. Nikole Hartmann, thank you for imparting your academic experience. Your comments on publications were incredibly helpful and game-changing.

I want to thank my colleagues, Dr. Sviatoslav Bilokin, Yaroslav Kulii, Nathalie Eberlein, Boyang Yu, Pascal Schmolz, Caspar Schmitt, David Koch, Lorenz Gärtner, and Arul Prakash for their constant support and cheerful conversations.

Achieving this milestone would have been unimaginable without the abundant and heartfelt support from my family and friends. Your unwavering belief in me has been my keystone, making it possible for me to persevere through the most challenging times. Thank you for standing by me. I look forward to many more years filled with joy, love, and shared triumphs!

To my beloved wife, Setareh, your love has been the alchemy that transformed long nights of despair into hope and self-doubt into self-belief. Your kindness taught me that even in the toughest of times, love could light the way. Your wisdom served as the balance to my often one-track mind, reminding me to see life in all its beautiful complexities. You celebrated my smallest victories and stood by me in my biggest defeats. You were the calm in the academic storm, offering perspective when I most needed it. I am immensely grateful to have shared this journey with you, and I can't wait to embark on new adventures together.