

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**NEUTRINO PARTICLE DIRECTION ESTIMATION USING
TRANSFORMER-BASED DEEP LEARNING**



M.Sc. THESIS

Şevval COŞKUN

Department of Physics Engineering

Physics Engineering Programme

FEBRUARY 2026

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL

**NEUTRINO PARTICLE DIRECTION ESTIMATION USING
TRANSFORMER-BASED DEEP LEARNING**



M.Sc. THESIS

**Şevval COŞKUN
(509201121)**

Department of Physics Engineering

Physics Engineering Programme

**Thesis Advisor: Prof. Dr. Emre Onur KAHYA
Thesis Co-Advisor: Academic Dr. Doğu SIRT**

FEBRUARY 2026

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**TRANSFORMER TABANLI DERİN ÖĞRENME KULLANILARAK
NÖTRİNO PARÇACIK YÖNÜ KESTİRİMİ**

YÜKSEK LİSANS TEZİ

**Şevval COŞKUN
(509201121)**

Fizik Mühendisliği Anabilim Dalı

Fizik Mühendisliği Programı

**Tez Danışmanı: Prof. Dr. Emre Onur KAHYA
Eş Danışman: Öğr. Gör. Dr. Doğu SIRT**

ŞUBAT 2026

Şevval COŞKUN, a M.Sc. student of ITU Graduate School student ID 509201121, successfully defended the thesis entitled “NEUTRINO PARTICLE DIRECTION ESTIMATION USING TRANSFORMER-BASED DEEP LEARNING”, which she prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor : **Prof. Dr. Emre Onur KAHYA**
Istanbul Technical University

Co-advisor : **Academic Dr. Doğu SIRT**
Istanbul Technical University

Jury Members : **Prof. Dr. Mehmet ÖZKAN**
Istanbul Technical University

Doç. Dr. Şener ÖZÖNDER
Boğaziçi University

Doç. Dr. Ayşe Nihan KATIRCI
Doğuş University

Date of Submission : 1 January 2026
Date of Defense : 9 February 2026





To my family,



FOREWORD

This thesis was created as part of the Master of Science program at the Graduate School of Istanbul Technical University. The main goal of this research was to develop a deep learning method to solve a key problem in neutrino astronomy: accurately determining the direction neutrinos arrive from, using data from the IceCube experiment. The study process has proven to be both arduous and intellectually stimulating. The intricacy of the challenge required not just a profound comprehension of high-energy physics and machine learning but also the application of sophisticated computational techniques like distributed training.

I extend my heartfelt appreciation to my supervisor, Prof. Dr. Emre Onur KAHYA, and my co-supervisor, Academic Dr. Doğu SIRT, for their indispensable support, intellectual acumen, and patience during this research endeavor. Their encouragement to investigate innovative methodologies and their constructive critiques were important in the formulation of this thesis.

I extend my profound gratitude to the National Center for High Performance Computing (UHeM) at Istanbul Technical University. The computational infrastructure furnished by UHeM was crucial for the execution of the comprehensive distributed training experiments, encompassing both MPI and DDP methodologies, that formed the foundation of this investigation. I am grateful to my friends for their unsparing help and support during the long periods of study; their friendship played a significant role in overcoming the obstacles encountered. Furthermore, I extend my heartfelt thanks to my family for their unwavering love and support.

February 2026

Şevval COŞKUN
(Banker)



TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xv
SYMBOLS	xix
LIST OF TABLES	xxi
LIST OF FIGURES	xxiii
SUMMARY	xxv
ÖZET	xxvii
1. INTRODUCTION	1
1.1 The Neutrino Sector as a Source of Novel Physics.....	1
1.2 Issues with Experiments and the Function of Large-Scale Detectors	1
1.3 The Shift from Engineered Features to Deep Learning (DL)	2
1.4 What Advanced Neural Network Architectures Can Do	3
1.4.1 Convolutional neural networks (CNNs) for visual data.....	3
1.4.2 Graph neural networks (GNNs) for geometries that aren't uniform.....	4
1.4.3 Transformer architectures and self-attention mechanisms.....	4
1.5 Contributions of the Thesis	5
2. REVIEW OF THE LITERATURE	7
2.1 Classical and Statistical Approaches in Neutrino Event Reconstruction	7
2.2 Computer Vision Approaches and Advanced Deep Learning Architectures.....	8
2.2.1 From hand-crafted features to end-to-end learning.....	8
2.2.2 The evolution of CNN architectures in neutrino experiments	10
2.2.2.1 Early implementations: the GoogLeNet-inspired CVN for NOvA ...	10
2.2.2.2 Squeeze-and-excitation residual network for the DUNE experiment	10
2.2.3 Quantifying performance improvements and classification power	11
2.2.3.1 Performance gains in the NOvA experiment	11
2.2.3.2 State-of-the-art performance in the DUNE experiment	12
2.2.4 Strategies for handling sparse data and semantic segmentation in LArTPCs	12
2.2.4.1 Computational inefficiencies and submanifold sparse networks	13
2.2.4.2 Semantic segmentation and the U-ResNet architecture	13
2.2.4.3 Context-enriched architectures and multi-task learning	14
2.2.5 Deep learning for neutrino direction and vertex reconstruction	14
2.2.5.1 Accelerating analysis with enhanced computational efficiency	14
2.2.5.2 Improving physics reach with superior regression performance	15
2.2.6 Advanced deep learning architectures in the IceCube observatory	15
2.2.6.1 Geometry-aware architectures: hexagonal CNNs.....	16
2.2.6.2 Generative data augmentation: superresolution	16
2.2.6.3 Hybrid methods and the galactic plane	16
2.3 Foundations and Advanced Applications of GNNs	17
2.3.1 Theoretical basis and the scaling imperative	17

2.3.2 The GNN design pipeline and point cloud processing.....	18
2.3.2.1 Dynamic graph construction for topological inference.....	18
2.3.2.2 Graph-based regression for directional reconstruction	19
2.3.3 Advanced architectures and benchmarks in neutrino physics.....	19
2.3.3.1 Performance on irregular geometries: the IceCube benchmark	19
2.3.3.2 DynEdge: precise reconstruction for IceCube events	20
2.3.3.3 Evolving GNNs: attention mechanisms and NuGraph2	20
2.3.4 GraphNeT: standardizing deep learning for neutrino telescopes	20
2.4 Deep Neural Networks for Reconstructing Neutrino Events	21
2.4.1 The rise of deep neural networks in neutrino physics	21
2.4.2 Full event reconstruction.....	22
2.4.2.1 How well shallow detectors work	22
2.4.2.2 A comparison of different detector geometries.....	22
2.4.3 Advanced methods for uncertainty quantification	23
2.4.3.1 Methods for quantifying uncertainty in in-ice radio detectors	23
2.4.3.2 A thorough evaluation of UQ methodologies	24
2.4.4 Modeling complex physical and systematic effects	24
2.4.4.1 Consequences of birefringence	25
2.4.4.2 How emission models affect sensitivity	25
2.4.5 Additional methodological advancements	26
2.5 Transformer Architectures in Reconstructing Neutrino Events	27
2.5.1 How transformer architecture can be used for visual tasks.....	27
2.5.2 Reasons for using transformers in neutrino data analysis	29
2.5.3 Implementations and changes in neutrino physics.....	30
2.5.3.1 Directional reconstruction in IceCube.....	30
2.5.3.2 Breaking down vertex activity	31
2.5.3.3 Finding and starting UHE neutrinos.....	31
2.5.4 A comparison with other architectures.....	32
2.5.5 Current limitations and future research opportunities.....	33
3. EXPERIMENTAL SETUP AND DATASET	35
3.1 Dataset Selection and Rationale	35
3.1.1 Structural compatibility with transformer architecture	35
3.1.2 Comparability and dataset characteristics	36
3.2 The Properties of Neutrinos.....	36
3.3 The IceCube Neutrino Observatory.....	37
3.3.1 Detector geometry and components.....	37
3.4 The Principle of Detection is Cherenkov Radiation.....	39
3.4.1 The first interaction: from neutrino to lepton.....	39
3.4.2 The cherenkov effect: an optical shockwave in ice.....	40
3.4.3 Signal transduction: from photons to digital waveforms	40
3.5 Event Topologies and Signatures	41
3.5.1 Track topologies: the muon signature	41
3.5.2 Cascade topologies: the shower signature.....	42
3.5.3 The reconstruction challenge and the drive for research.....	43
3.6 Dataset Structure and Event Representation	44
3.6.1 Data format and sparsity	44
3.7 Input Variables and Feature Space	44
3.8 Target Variables: Directional Reconstruction	45
4. METHODOLOGY.....	47
4.1 Choosing and Explaining the Model Architecture: Why Transformer?	47

4.1.1 A sequential modeling and long-range dependencies	47
4.1.2 Positional awareness	47
4.1.3 Efficiency and scalability of computation	47
4.1.4 Success in experiments and place in the literature.....	48
4.2 Model Architecture and Experimental Design	48
4.2.1 Data preparation	48
4.2.1.1 Feature normalization.....	49
4.2.1.2 Sequence length and padding strategy	50
4.2.1.3 Data splitting	51
4.2.2 Feature engineering	51
4.2.3 Model architecture	52
4.2.3.1 Positional encoding	52
4.2.3.2 Multi-head self-attention.....	53
4.2.3.3 Position-wise feed-forward networks	54
4.2.3.4 Layer normalization and residual connections.....	54
4.2.3.5 Output layer.....	55
4.3 Training Infrastructure and Configuration	56
4.3.1 Distributed computing.....	56
4.3.2 Optimization.....	56
4.3.3 Loss functions	58
4.4 Performance Metrics	59
5. RESULTS AND DISCUSSION	61
5.1 Comparison of Experimental Results.....	61
5.2 Performance Analysis	62
5.2.1 Performance of the baseline model (Exp1 - Exp2)	62
5.2.2 Constraints in intermediate experiments (Exp3 - Exp5).....	62
5.2.3 Superiority of the proposed method (Exp6 - Exp7).....	63
5.3 Analysis of the Best and Worst Possible Outcomes	64
5.4 Discussion on Metric Inconsistency (R^2 Analysis).....	65
5.5 Visual Analysis of Direction Prediction Vectors	66
6. FUTURE WORKS	71
REFERENCES.....	73
CURRICULUM VITAE.....	81



ABBREVIATIONS

ATWD	: Analog Transient Waveform Digitizer
AUC	: Area Under the Curve
BDTs	: Boosted Decision Trees
CC	: Charged-Current
CCSN	: Core-Collapse Supernovae
CDR	: Conceptual Design Report
CKM	: Cabibbo-Kobayashi-Maskawa
CNN	: Convolutional Neural Network
CP	: Charge-Parity
CVN	: Convolutional Visual Network
DDP	: Distributed Data Parallel
DGCNN	: Dynamic Graph Convolutional Neural Network
DL	: Deep Learning
DNN	: Deep Neural Network
DOM	: Digital Optical Module
DT	: Decision Tree
DUNE	: Deep Underground Neutrino Experiment
EDL	: Evidential Deep Learning
fADC	: Fast Analog-to-Digital Converter
FFC	: Fast Neutrino Flavor Conversion
FLOPs	: Floating-Point Operations
FNN	: Feed-Forward Network
GAN	: Generative Adversarial Network
GeV	: Giga Electronvolt
GNN	: Graph Neural Network
GPU	: Graphics Processing Unit
GRU	: Gated Recurrent Unit
HLC	: Hard Local Coincidence
HL-LHC	: High-Luminosity Large Hadron Collider
IoU	: Intersection over Union

KNN	: k-Nearest Neighbors
LArTPC	: Liquid Argon Time Projection Chamber
LC	: Local Coincidence
LPM	: Landau-Pomeranchuk-Migdal
LR	: Logistic Regression
LSTM	: Long Short-Term Memory
MAE	: Mean Angular Error
MCD	: Monte Carlo Dropout
ML	: Machine Learning
MLH	: Maximum-Likelihood
MLP	: Multi-Layer Perceptron
MPID	: Multiple Particle Identification
MSE	: Mean Squared Error
NC	: Neutral Current
NLP	: Natural Language Processing
NOvA	: The NuMI Off-axis ν_e Appearance
ns	: Nanosecond
NSM	: Neutron Star Merger
PDF	: Probability Density Function
p.e.	: Photoelectrons
PeV	: Peta Electronvolt
PMT	: Photomultiplier Tube
RNO-G	: Radio Neutrino Observatory Greenland
RAPCal	: Reciprocal Active Pulsing Calibration
RCNN	: Region-based Convolutional Neural Network
ResNet	: Residual Networks
RMS	: Root Mean Square
RNN	: Recurrent Neural Network
Se-ResNet	: Squeeze-and-Excitation Residual Network
SGD	: Stochastic Gradient Descent
SLC	: Soft Local Coincidence
SSCN	: Submanifold Sparse Convolutional Network
SuperFGD	: Super Fine-Grained Detector
SVM	: Support Vector Machine
TeV	: Tera Electronvolt

UHE	: Ultra-High-Energy
UheM	: Istanbul Technical University National Center for High Performance Computing
UV	: Ultraviolet
UQ	: Uncertainty Quantification
VAE	: Variational Autoencoder
ViT	: Vision Transformer
2D	: 2-Dimensional
3D	: 3-Dimensional





SYMBOLS

q	: Charge
CC	: Charged-Current
χ^2	: Chi-squared
e^-	: Electron
ν_e	: Electron neutrino
γ	: Gamma
K	: Key
μ^-	: Muon
μs	: Microsecond
ν_μ	: Muon neutrino
π^0	: Nötr pion
ϕ	: Phi (azimuth angle)
π	: Pi (radians)
$\pi^\pm$	: Pion
p	: Proton
Q	: Query
σ	: Sigma
ν_τ	: Tau neutrino
t	: Time
V	: Value
θ_{zenith}	: Zenith angle



LIST OF TABLES

	<u>Page</u>
Table 3.1 : Presents a Sample of the Sensor Geometry Data.....	45
Table 3.2 : Presents a Sample of the Parquet Data Structure Read from the Train Folder.....	45
Table 3.3 : Presents a Sample from the Train Meta Parquet File.....	46
Table 5.1 : Experimental Results and Configurations.	61





LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : The Transformer Model Architecture. Encoder (left) and Decoder (right) [7].	28
Figure 3.1 : Schematic View of the IceCube Neutrino Observatory [58].....	38
Figure 3.2 : Sensor Geometry.	39
Figure 3.3 : A Muon Track Topology Observed in IceCube [16].	42
Figure 3.4 : A Particle Shower (Cascade) Topology Observed in IceCube [16].....	43
Figure 4.1 : Model Architecture for Exp6 and Exp7.	52
Figure 5.1 : Training and Validation Loss Curves for Exp6.....	63
Figure 5.2 : Learning Curves During the Fine-Tuning Phase (Exp7).....	63
Figure 5.3 : Visualization of the Best and Worst Prediction Scenarios from Exp7..	65
Figure 5.4 : 3D Comparison of True and Predicted Direction Vectors for Exp1.	66
Figure 5.5 : 3D Comparison of True and Predicted Direction Vectors for Exp2.	67
Figure 5.6 : 3D Comparison of True and Predicted Direction Vectors for Exp3.	67
Figure 5.7 : 3D Comparison of True and Predicted Direction Vectors for Exp4.	68
Figure 5.8 : 3D Comparison of True and Predicted Direction Vectors for Exp5.	68
Figure 5.9 : 3D Comparison of True and Predicted Direction Vectors for Exp6.	69
Figure 5.10 : 3D Comparison of True and Predicted Direction Vectors for Exp7... ..	69



NEUTRINO PARTICLE DIRECTION ESTIMATION USING TRANSFORMER-BASED DEEP LEARNING

SUMMARY

The neutrino sector represents a crucial area for advancing our comprehension of the cosmos, extending beyond the confines of the Standard Model of particle physics. Nevertheless, the weak interaction cross-sections, coupled with the intricate data structures derived from large-scale detectors such as the IceCube Neutrino Observatory, present substantial obstacles in the detection and reconstruction of these elusive particles. Conventional algorithms and standard Convolutional Neural Networks (CNNs) encounter considerable limitations when processing the irregular, non-grid geometry of the IceCube detector, especially in the context of "cascade" events, where directional resolution is intrinsically constrained. Consequently, this thesis introduces a Deep Learning methodology, grounded in the Transformer architecture, to estimate neutrino arrival directions, specifically azimuth and zenith angles.

The main goal is to use the self-attention mechanism in Transformers to capture global dependencies and long-range spatiotemporal correlations in pulse series of varying lengths. This strategy allows for the direct processing of the sequential raw data, independent of fixed geometric constraints. This approach uses the complete simulation dataset from the "IceCube - Neutrinos in Deep Ice" competition, which contains over 138 million events. The research followed a structured experimental process, from Exp1 to Exp7, to improve the model's architecture and training methods. The input data, which consisted of pulse series, went through thorough preprocessing. This included adding Spherical coordinates (radial distance and angular orientation) to standard Cartesian coordinates, along with normalization, to better represent the detector's geometry.

The sequences were analyzed using a Transformer Encoder architecture, which included eight attention heads. The training process changed from using the usual Mean Squared Error (MSE) loss functions to Cosine Similarity Loss, a method better suited for angular data. Learning rate schedulers, including warm-up periods, were also used. Some of the experimental work was done on the National Center for High Performance Computing (UHeM) infrastructure at Istanbul Technical University. In addition, the training workflows were improved using distributed computing methods, such as PyTorch Distributed Data Parallel (DDP). Although early experiments showed difficulties in geometric generalization, the final configurations (Exp6 and Exp7) demonstrated both stability and the ability to generalize spatially.

In the Exp7 phase, the fine-tuned model attained a Mean Angular Error of 63.80 degrees on the test set, indicating an approximate 11-degree enhancement relative to the baseline model. Visual assessments corroborated the model's capacity to resolve the head-tail ambiguity present in particle track reconstruction, enabling it to differentiate between individual event topologies instead of merely predicting the

dataset mean. Moreover, the model exhibited its applicability to real-time data streams, achieving a high inference speed that allowed for the processing of roughly 375 events per second. Based on the experimental results, it can be stated that the Transformer architecture shows promise for this field. Furthermore, it is concluded that further success could be achieved by combining the Transformer with Graph Neural Networks (GNNs), which have demonstrated general success in the literature, within a hybrid architecture.



TRANSFORMER TABANLI DERİN ÖĞRENME KULLANILARAK NÖTRİNO PARÇACIK YÖNÜ KESTİRİMİ

ÖZET

Nötrino sektörü, parçacık fiziğinin Standart Modeli'nin ötesindeki evren anlayışımızı genişletmek adına kritik bir sınır teşkil etmektedir. Kozmik haberciler olarak bilinen bu parçacıklar, evrenin en uzak ve en enerjik köşelerinden, manyetik alanlardan veya maddeden etkilenmeden doğrudan bilgi taşıyabilme kapasitesine sahiptir. Ancak, nötrinoların madde ile etkileşime girme olasılıklarının tesir kesitlerinin son derece düşük olması, bu hayalet parçacıkların tespit edilmesini ve analiz edilmesini modern astrofiziğin en zorlu problemlerinden biri haline getirmektedir. Bu zorluğu aşmak amacıyla Antarktika buzullarının derinliklerine inşa edilen ve bir kilometreküp hacmi kapsayan IceCube Nötrino Gözlemevi, buz atomları ile etkileşime giren nötrinoların yaydığı zayıf Cherenkov ışınmasını tespit etmek üzere tasarlanmıştır.

Bu tez çalışmasındaki temel deneysel zorluk, IceCube dedektöründen elde edilen simülasyon verileri kullanarak, nötrino geliş yönlerinin (Azimuth ve Zenith açıları) hassas bir şekilde tahmin edilmesidir. IceCube dedektörü, buzun içine dikey halatlar boyunca yerleştirilmiş 5.000'den fazla Dijital Optik Modülden (DOM) oluşmaktadır. Bir nötrino etkileşimi gerçekleştiğinde, bu sensörler fotonları yakalayıp zaman, yük ve konum bilgisini içeren atım serileri üretir. Ancak üretilen bu veri, son derece seyrek, düzensiz ve gürültülü bir yapıya sahiptir.

Olay topolojileri, etkileşime giren parçacığın kütlesi ve enerji kaybı mekanizmalarına bağlı olarak İz (Track) ve Kaskad (Cascade) olmak üzere iki ana sınıfa ayrılır. Müon nötrinoları, müonun yüksek kütlesi sebebiyle düşük ışıma yaparak kilometrelerce uzunluğunda düz bir iz bırakır ve bu iz üzerinden yön tayini yapmak görece mümkündür. Buna karşın, Elektron ve Tau nötrinolarının baskın olduğu Cascade olaylarında durum çok daha karmaşıktır. Bu parçacıklar, enerjilerini çok kısa mesafelerde (yaklaşık 10-20 metre) elektromanyetik duşlar yoluyla boşaltırlar. Dedektör sensörleri arasındaki geniş boşluklar sebebiyle, bu olaylar noktasal ışık patlamaları olarak görülür ve yönsel bilgi büyük ölçüde kaybolur.

Geleneksel istatistiksel rekonstrüksiyon algoritmaları ve standart Evrişimli Sinir Ağları (CNN), IceCube dedektörünün ızgara yapısında olmayan, altıgen ve düzensiz geometrisini işlerken ciddi kısıtlarla karşılaşmaktadır. Özellikle, yönsel çözünürlüğün doğası gereği sınırlı olduğu ve parçacık izinin net bir şekilde görülmediği cascade tipi olaylarda, mevcut yöntemlerin hassasiyeti düşmekte ve hesaplama maliyetleri artmaktadır.

Bu tez, söz konusu zorlukları aşmak ve rekonstrüksiyon hassasiyetini artırmak amacıyla, nötrino geliş yönlerini tahmin etmek için Transformer mimarisine dayalı bir derin öğrenme metodolojisi önermektedir. Çalışmanın temel hedefi, değişken uzunluklu atım serileri içindeki küresel bağımlılıkları ve uzun menzilli uzamsal-zamansal korelasyonları yakalamak için Transformer'ların Öz-Dikkat mekanizmasından yararlanmaktır. Bu mekanizma, bir olay içerisindeki her bir sensör verisinin diğer tüm sensör verileriyle olan ilişkisini ağırlıklandırarak öğrenir; böylece model, dedektörün hangi bölgesinde ve hangi zaman aralığında kritik bilgi olduğunu, geometrik bir önyargı olmaksızın keşfedebilir.

Metodoloji, “IceCube - Neutrinos in Deep Ice” yarışması kapsamında sağlanan ve 138 milyondan fazla olay içeren kapsamlı bir simülasyon veri setini kullanmaktadır. Bu veri seti, modelin genelleme yeteneğini artırmak için büyük bir fırsat sunmaktadır. Araştırma süreci, model mimarisini, veri ön işleme tekniklerini ve eğitim stratejilerini optimize etmek amacıyla yedi farklı aşamadan (Exp1'den Exp7'ye) oluşan sistematik bir deneysel ilerleme izlemiştir. Girdi verileri, modelin daha iyi temsil elde edebilmesi adına özellik mühendisliği sürecinden geçirilmiştir. Her bir nötrino olayı, matematiksel olarak $(N \times F)$ boyutunda bir tensör yapısına dönüştürülmüştür; burada N dizi uzunluğunu (Sequence Length), F ise özellik boyutunu (Feature Dimension) temsil etmektedir. Bu yapı, deneysel süreç boyunca optimize edilmiştir. Başlangıç deneylerinde (Exp1-Exp5) girdi vektörü yalnızca zaman, yük, auxiliary durumu ve Kartezyen koordinatlarda (x, y, z) sensör konumlarını içeren 6 temel değişkene ($F = 6$) sahiptir. Optimize edilmiş nihai aşamada (Exp6-Exp7) bu yapıya sensör konumlarının Küresel Koordinatlardaki (r, θ, ϕ) karşılığı da entegre edilerek özellik uzayı 9 boyuta ($F = 9$) genişletilmiştir. Standart Kartezyen koordinatların yanı sıra, Küresel Koordinatların da sisteme dahil edilmesi, modelin uzaydaki yönelimleri daha hızlı kavramasını sağlamıştır. Ayrıca, zaman ve yük verileri üzerinde uygulanan logaritmik dönüşümler ve normalizasyon teknikleri, modelin sayısal kararlılığını artırmıştır.

Önerilen model, nötrino olaylarını işlemek için özelleştirilmiş, yalnızca kodlayıcıdan (Encoder-only) oluşan bir Transformer mimarisine dayanmaktadır. Süreç, dedektörden gelen ham atım serilerinin, zenginleştirilmiş öznitelik vektörlerine dönüştürülmesiyle başlar. Bu vektörler, doğrusal bir katman aracılığıyla daha yüksek boyutlu gizli bir uzaya iz düşürülerek modelin işleyebileceği matematiksel forma getirilir. Ardından, mimarinin çekirdeğini oluşturan Çok Başlı Öz-Dikkat (Multi-Head Self-Attention) mekanizması devreye girerek, olay içindeki tüm sensör verileri arasındaki uzun menzilli küresel bağımlılıkları ve geometrik ilişkileri analiz eder. Bu katmanlarda elde edilen bağlamsal bilgiler, ağırlıklı bir havuzlama (Weighted Pooling) işleminden geçirilerek olayı temsil eden tek bir özet vektöre indirgenir. Son aşamada ise, başlangıç deneylerinde kullanılan basit doğrusal çıktı katmanının yerini alan gelişmiş bir Çok Katmanlı Algılayıcı (MLP) yapısı, bu soyut özet vektörü fiziksel uzaya tercüme ederek nötrinin geliş yönünü belirler. Bu çıktı mekanizması, deneysel stratejinin evrimine paralel olarak, başlangıçta doğrudan açılabilir değerleri, nihai konfigürasyonda ise bu açıların türetildiği 3 boyutlu yön vektörlerini (x, y, z) tahmin edecek şekilde yapılandırılmıştır.

Başlangıç deneylerinde (Exp1 ve Exp2) kayıtp fonksiyonu olarak, standart Hata Kareler Ortalaması (MSE) kayıp fonksiyonu kullanılmıştır. Ancak, açısal verilerin döngüsel doğası, standart MSE fonksiyonunun başarısız olmasına, eğitimin kararsızlaşmasına yol açmıştır. Model, bu aşamada tüm olaylar için veri setinin ortalama yönünü tahmin etme eğilimi göstermiştir. Bu problemi aşmak için ara deneylerde (Exp3-Exp5), optimizasyon hedefi açısal uzaydan vektörel uzaya taşınmış ve DirectionMSE fonksiyonu denenmiştir. Bu yaklaşımda açılar (x, y, z) birim vektörlerine dönüştürülerek Öklid mesafesi üzerinden hata hesaplanmış, ancak donanım kısıtları nedeniyle tam kararlılık sağlanamamıştır.

Son deneylerde (Exp6-Exp7), eğitim stratejisi daha dirençli ve GPU hızlandırmalı bir çerçeveye evrilmiştir. Bu yeni çerçevede, vektörlerin uzaydaki hizalanmasını doğrudan maximize eden Kosinüs Benzerliği Kaybı (Cosine Similarity Loss) kullanılmıştır. Bu kayıp fonksiyonu, açısal verilerdeki periyodiklik ve süreksizlik problemini çözerek modelin yönelimsel öğrenme kapasitesini açığa çıkarmıştır. Ayrıca, ısınma periyotlu öğrenme oranı zamanlayıcılarının kullanımı, modelin yerel minimumlardan kurtularak daha iyi bir global optimuma ulaşmasını sağlamıştır.

Çalışmanın ölçeklenebilirliğini sağlamak ve veri setini makul sürelerde işleyebilmek için İstanbul Teknik Üniversitesi Ulusal Yüksek Başarımlı Hesaplama Merkezi (UHeM) altyapısının yanı sıra Kaggle ve Google Colab Pro platformlarının sunduğu yüksek performanslı GPU kaynaklarından yararlanılmıştır. Eğitim süreçleri, tekil GPU kullanımından çoklu GPU ve çoklu düğüm yapılarına taşınmış; bu süreçte altyapı, manuel senkronizasyon gerektiren Mesaj Geçiş Arayüzü'nden (MPI), PyTorch'un modern ve optimize edilmiş Dağıtık Veri Paralellliği (DDP) modülüne evrilmiştir.

İlk deneyler geometrik genelleştirmede zorlanırken, optimize edilmiş nihai konfigürasyon (Exp6 ve Exp7) hem kararlılık hem de uzamsal genelleme yeteneği kazanmıştır. Özellikle, Exp7 aşamasındaki ince ayarlı model, test setinde 0.45 seviyesinde bir MSE skoruna ulaşmış ve fiziksel bir metrik olan Ortalama Açısal Hata değerini 63.80 dereceye düşürmüştür. Bu değer, temel modele kıyasla yaklaşık 11 derecelik bir iyileşmeyi temsil etmektedir. Rastgele tahminin ortalama hatasının 90 derece civarında olduğu ve cascade olaylarının zorluğu göz önüne alındığında, bu sonuç modelin olay topolojisini başarılı bir şekilde öğrendiğini göstermektedir.

Dahası, modelin hesaplama verimliliği, gerçek dünya uygulamaları için büyük bir potansiyel taşımaktadır. Geliştirilen model, saniyede yaklaşık 375 olayı işleyebilen yüksek bir hıza sahiptir. Bu hız, modelin IceCube ve gelecekteki nötrino teleskoplarında (örneğin IceCube-Gen2), olayları anlık olarak filtreleyen ve kaydeden tetikleme sistemlerinde gerçek zamanlı olarak kullanılabileceğini göstermektedir. Bu bulgular, Transformer mimarisinin, fiziğe dayalı özellik mühendisliği ve uygun kayıp fonksiyonları ile birleştirildiğinde, nötrino olay rekonstrüksiyonu için güçlü, hassas ve ölçeklenebilir bir alternatif sunduğunu ortaya koymaktadır. Gelecek çalışmalarda, Graf Sinir Ağlarının (GNN) hibrit bir mimariye entegre edilmesiyle hem yerel geometrik nüansların hem de küresel bağlamın daha iyi yakalanması ve performansın daha da artırılması öngörülmektedir.



1. INTRODUCTION

1.1 The Neutrino Sector as a Source of Novel Physics

The neutrino sector presents a crucial domain for probing phenomena that elude the boundaries of the Standard Model of particle physics. Although the Standard Model represents a considerable theoretical achievement, it does not furnish a complete account of key experimental observations and evidence. These encompass the existence of dark matter and dark energy, the non-zero masses of neutrinos, and the observed imbalance between matter and antimatter in the universe [1]. Neutrinos are the most mysterious particles in the Standard Model, and the things that have been seen about them suggest that there is new physics. The discovery of neutrino oscillations, in which neutrinos shift between flavor eigenstates during propagation, unequivocally indicates that neutrinos have mass, a phenomenon not addressed by the Standard Model. This raises important questions about how they get their mass and opens the door to other new physical phenomena. Moreover, the possibility of charge-parity (CP) violation in the neutrino sector may be associated with leptogenesis, a fundamental principle that elucidates the observed preeminence of matter over antimatter. It is important to answer these unanswered questions in order to learn more about fundamental particles and possibly get a better picture of the early Universe [2]. The urgent, unanswered questions are driving the development of a new generation of advanced experiments that will study the neutrino sector with unparalleled precision.

1.2 Issues with Experiments and the Function of Large-Scale Detectors

The Experiment of neutrino physics presents considerable technical and statistical obstacles. Because neutrinos interact only very weakly with matter, detecting them directly is quite difficult. To overcome this challenge and effectively separate the signal from background noise, data processing is as important as building large, complex detectors. Therefore, advanced data processing techniques are essential for handling the complex data from these detectors and successfully identifying these elusive particles.

The IceCube Neutrino Observatory is a large detector with sensors spread out over a cubic kilometer of ice in Antarctica. The sparse and irregular sensor design generates data inadequate for conventional image-based analytical techniques, necessitating innovative methods to reconstruct particle interactions [2].

The NuMI Off-axis ν_e Appearance (NOvA) experiment is a long-baseline study that uses detectors filled with scintillating mineral oil. The design makes it easy to make two-dimensional, image-like pixel maps of how particles interact from two different angles. This makes it a great subject for computer vision methods [3,4].

The upcoming Deep Underground Neutrino Experiment (DUNE) will use Liquid Argon Time Projection Chambers (LArTPCs) technology. LArTPCs give very accurate, high-resolution, picture-like data of the paths of particles.

However, the data obtained from these large detectors is typically spatially sparse. This implies that while particle interactions produce distinct signals, they occupy only a small fraction of the total detector volume. Consequently, these sophisticated detectors generate vast and complex datasets that are difficult to process and interpret, necessitating the adoption of modern computing methods [2,3].

1.3 The Shift from Engineered Features to Deep Learning (DL)

The advent of deep learning in the early 2010s has precipitated a significant transformation in the methodology of data analysis within particle physics. Although machine learning techniques have been employed in this domain for an extended period, established practices primarily depended on shallow learning approaches. Within this established paradigm, domain specialists manually devised high-level attributes exemplified by the invariant mass of a decaying particle drawing upon their physical understanding. These engineered features were subsequently analyzed by shallow classifiers, including Boosted Decision Trees (BDTs) or single-layer neural networks, to differentiate between signal and background events [4,5].

This plan works, but it takes a lot of work and has some problems that are already there. It might not be able to see all the details in the low-level detector readouts and might not be able to capture the full complexity of the data. Because of better algorithms and stronger hardware like Graphics Processing Units (GPUs) [2], DL has become a new way to do things. DL algorithms can work directly with low-level, high-

dimensional data, such as the raw pixel values from a detector image. They can figure out the most important parts by themselves, without needing an operator. This method directly fixes the main problem with the traditional model. It lowers the risk of knowledge loss by letting the computer find complex patterns that human experts might miss.

Deep neural networks (DNNs) trained on low-level inputs have either performed as well as or better than shallow methods that use physics-engineered high-level features. This shows that they can find the underlying physics that is important on their own [5]. This change has made some neural network architectures better for the unique data formats used in modern particle physics research.

1.4 What Advanced Neural Network Architectures Can Do

It's hard to evaluate modern neutrino detectors because of their unique and often complicated data structures. To go from regular grid-based images to sparse, irregular point clouds, it has been necessary to make and improve many different kinds of DL architectures. Every architecture is built to handle a different problem with data analysis.

1.4.1 Convolutional neural networks (CNNs) for visual data

Convolutional Neural Networks (CNNs), developed for computer vision applications, exhibit significant utility in analyzing data from detectors that produce structured, grid-like outputs. Their inherent capacity to identify spatial patterns makes them especially appropriate for scrutinizing image-like data from experiments such as NOvA and those utilizing Liquid Argon Time Projection Chambers (LArTPCs). For instance, the NOvA experiment utilized a Convolutional Visual Network (CVN) based on the GoogLeNet architecture. By utilizing the detector's two orthogonal views as image channels, the CVN effectively differentiated between neutrino interactions. This approach yielded a 40% increase in electron neutrino selection efficiency at a fixed purity compared to traditional reconstruction methods. This significant progress highlights the potential of CNNs in enhancing physics analysis [3,4].

CNNs have been very useful for finding neutrino interaction signals among a lot of cosmic rays in LArTPC studies. Faster- Region-based Convolutional Neural Network (RCNN) and other algorithms are used to make a bounding box around the particles

that are made when neutrinos hit something. This does a good job of separating the signal of interest from the background noise in the detectors complicated images [4].

1.4.2 Graph neural networks (GNNs) for geometries that aren't uniform

In studies with sparse data and irregular detector geometries, GNNs have shown to be more effective and robust. A GNN is a type of function that can be trained to work with graphs. Graphs are data structures that have nodes (like individual detector hits) and edges that show how the nodes interact with each other [6]. This representation is flexible by nature and not limited by a fixed grid structure. The IceCube Neutrino Observatory is a good example of this because its sensors are spread out and not evenly spaced in the Antarctic ice. As highlighted in a review by Psihas et al. a GNN developed by the IceCube collaboration for differentiating neutrino-induced signals from cosmic-ray backgrounds significantly outperformed CNNs and other conventional methods. This GNN architecture identified 6.3 times more signal events and achieved a signal to noise ratio that was three times higher than what was seen with standard baseline methods [2].

GNNs will also be used in the next DUNE experiment. The high-resolution but sparse data from its LArTPC can be used to make 2-Dimensional (2D) detector hit graphs. This lets GNNs put things back together by grouping the edges that connect hits based on the type of particle, such as muons, electrons, or hadrons. This is a very important step in figuring out how particles move in the sparse LArTPC data [3].

1.4.3 Transformer architectures and self-attention mechanisms

The Transformer model is a new type of architecture that doesn't use recurrence or convolutions; instead, it uses an attention mechanism [7]. Self-attention is a way to make a representation of a sequence by finding links between different parts of that sequence. A Transformer is a model that works on a full graph, which means that every node is connected to every other node. It dynamically figures out how strong the edges (attention weights) are and then uses that information to send messages. This architecture is made to work with sets of different lengths that stay the same when they are permuted, which is common in particle physics [1,6]. The increasing intricacy of these sophisticated supervised architectures highlights a significant reliance on high-quality simulated data for training, which introduces distinct challenges and potential constraints.

1.5 Contributions of the Thesis

This thesis proposes a comprehensive examination of the contemporary Transformer architecture’s capabilities in light of the limitations of existing analytical methods and the sparse, irregularly geometric data from the IceCube detector. We hypothesize that the self-attention mechanism can accurately represent the intricate geometry and extensive correlations of the IceCube detector. To achieve this, a validation study will utilize the high-fidelity simulation dataset from the IceCube Neutrinos in Deep Ice Kaggle Competition [8].





2. REVIEW OF THE LITERATURE

2.1 Classical and Statistical Approaches in Neutrino Event Reconstruction

The main challenge in neutrino astronomy is figuring out how a particle moved and how much energy it had based on the small patterns of Cherenkov light that an array of photomultiplier tubes (PMT) picks up. The accuracy of this reconstruction is of paramount importance, as it constitutes the primary technique for distinguishing upward-going neutrino signals from the prevalent background of downward-going atmospheric muons a critical concern addressed in several key studies. Separating astrophysical neutrinos from this background is a computational problem that a series of algorithms solve by creating a particle hypothesis with better physical accuracy and statistical reliability over time[9,10].

The complicated process of reconstruction usually starts with a quick, analytical first guess method that gives an initial seed for more resource-intensive fitting methods. The Line-Fit algorithm is a common way to do things that doesn't take into account the shape of the Cherenkov cone and instead uses a straight line. This makes it easier to find the initial track vector quickly and analytically. Different methods for recognizing patterns, like the Direct Walk algorithm, find possible track components by looking for pairs of hits that are linked by cause. These early algorithms are computationally efficient, but they purposely ignore the random nature of how Cherenkov light travels. To attain the requisite precision for astrophysical discovery, more stringent statistical methodologies that faithfully depict the underlying physics are imperative [9,10].

The maximum likelihood method is what modern reconstruction is built on. It looks at the arrival times and spatial distribution of recorded photons to find the best probability density function (PDF). A lot of the time, these probabilities use complicated parameterizations, like the Pandel function, to show how light moves through Antarctic ice [9]. Advanced reduction methods like MultiNest are used to find the global maximum in the multi-dimensional and often complicated probability space

[11]. One useful method is χ^2 minimization, which finds the best track parameters by minimizing a quality function based on the time and charge residuals of the hits [12].

The maximum probability and χ^2 methods assume that the signal is pretty clean. But noise from instruments or complicated topologies can make real events hard to understand because there are so many particles. Strong statistical methods are added to classical methods to lessen the effect of outlier hits. For instance, a Huber fit punishes outlier hits in a linear way instead of a quadratic way, which makes them have less of an effect on the final fit [10]. Also, modular, multi-algorithm frameworks break down pattern recognition into a series of separate topological tasks, such as hit clustering, followed by merging and splitting of clusters to find individual particle candidates [13].

After the first reconstruction, machine learning (ML) techniques such as BDTs are frequently employed to categorize events, thereby enhancing the rejection of background events [11]. The transition from a sparse light pattern to a high-fidelity particle track occurs in multiple stages, each of which adds complexity to the statistics. The same principles can be used with different detection methods. In-ice radio detectors replicate the electric field of radio signals through methods including template correlation and Information Field Theory [14]. These reconstruction pipelines create very accurate event directions and energies that are necessary for later statistical analyses, such as using nonparametric test statistics to look at anisotropies in neutrino arrival directions [15]. While geometric reconstruction primarily aims to optimize pointing accuracy, distinct algorithms are required to estimate energy by analyzing the stochastic energy losses and total light deposition along the track [16].

2.2 Computer Vision Approaches and Advanced Deep Learning Architectures

2.2.1 From hand-crafted features to end-to-end learning

A primary challenge in experimental high-energy physics is the precise categorization of particle interactions as either signal or background. This task is strategically significant for attaining critical physics objectives, including the precise measurement of neutrino oscillation a quantum mechanical phenomenon that provides insights into fundamental inquiries regarding our universe. Historically, physicists approached this classification problem through a multi-step process that relied on painstakingly

developed reconstruction algorithms. The new paradigm of DL, however, has fundamentally transformed this landscape, enabling more powerful and direct methods of analysis.

The old way of doing things was to first use raw detector data to put together high-level physics objects, like the tracks left by muons or the showers made by electrons. The properties of these objects their energy, direction, and shape were then summarized into a handful of quantities in a process known as feature engineering. These hand-crafted features were subsequently fed into ML algorithms, like BDTs, to perform the final signal-versus-background classification [17]. While successful for many years, this feature-engineering approach has inherent limitations. As Aurisano et al. (2016) note, it suffers from two critical failings. The ultimate classification is susceptible to inaccuracies originating from the preliminary high-level reconstruction phase; an error in track or shower identification can cascade through the analytical process, resulting in a flawed outcome. Furthermore, the features employed for classification are constrained by the pre-existing conceptualizations and implementations of physicists, which may preclude the identification of more nuanced or intricate data patterns that could enhance the efficacy of separation.

The recent revolution in computer vision has introduced a new paradigm that overcomes these challenges. CNNs, the algorithms that power modern image recognition, enable a shift away from constructed features toward direct, end-to-end learning. This approach is particularly well-suited for neutrino detectors like LArTPCs, which record energy depositions throughout their volume. When combined, these measurements produce what is essentially an image of the physics interaction, making the data perfectly suited for analysis with computer vision tools [17]. The primary advantage of the CNN approach is its ability to robustly and automatically extract learned features directly from the raw, pixel-level detector data. This makes the final classification independent of potential errors in intermediate high-level reconstruction algorithms like track-finding or shower reconstruction. By learning from the data itself, CNNs can discover the most salient features for a given classification task, by passing the limitations of human-engineered variables. This fundamental shift in methodology has been enabled by the specific evolution of CNN architectures, which have been progressively adapted for the unique challenges of neutrino physics [17,18].

2.2.2 The evolution of CNN architectures in neutrino experiments

The utilization of CNNs within the realm of neutrino physics has demonstrated a dynamic trajectory. This evolution has paralleled the advancements observed in computer vision architectures, transitioning from initial, foundational models to increasingly intricate and deeper networks. This advancement has been fueled by the imperative to extract progressively refined and abstract features from the intricate event topologies generated by contemporary detectors, consequently enhancing classification performance [19].

2.2.2.1 Early implementations: the GoogLeNet-inspired CVN for NOvA

A pioneering example of this new paradigm was the CVN developed for the NOvA experiment. The design was directly influenced by the powerful GoogLeNet architecture, which was great at classifying large numbers of images. The NOvA CVN used GoogLeNet's main idea, the inception module, which is a sub-network that uses filters at different scales to capture features of different sizes at the same time. The NOvA collaboration made a few important changes to the model so that it could work with physics data. They most importantly used a new split, parallel architecture. Because the NOvA detector provides two distinct, interleaved two-dimensional projections (an X-view and a Y-view) of a three-dimensional interaction, the CVN was designed with separate branches to process each view independently. The high-level features learned from each view were only concatenated and passed to a final, shared inception module at a later stage in the network. The collaboration also used a shortened version of the full GoogLeNet architecture because they thought that the neutrino event images were topologically simpler than the complex natural images that the original network was made for [17].

2.2.2.2 Squeeze-and-excitation residual network for the DUNE experiment

DUNE shows the next big step in the evolution of architecture. The DUNE CVN is built on a newer and more powerful architecture called a 34-layer Squeeze-and-Excitation Residual Network (SE-ResNet). The switch to a ResNet architecture gives it a big technical edge over older models like GoogLeNet. The main new thing about ResNet is that it uses residual connections, which are like shortcuts that let information skip layers in the network. This feature is very important for analyzing LArTPC data because it lets the fine-grained detail from the high-resolution detector images move

deeper into the network without being lost. NOvA used a shallower, GoogLeNet-inspired network that didn't have this feature [20].

The DUNE CVN architecture is built on this strong base. It uses parallel branches to process the different detector views, just like its NOvA predecessor. In the case of DUNE's LArTPC design, there are three views. But it also has a complex multi-output structure. This lets the network do more than one classification task at the same time. It can predict the type of neutrino interaction, the type of interaction, and the number of different final-state particles (protons, pions, etc.) from just one input image. These improvements in architecture have led to direct improvements in classification performance, setting a new standard for the field [20].

2.2.3 Quantifying performance improvements and classification power

The real value of this architectural change from early CNNs to modern deep networks is shown by measurable improvements in important performance metrics. In physics experiments, it is very important to be able to improve signal efficiency (the percentage of desired events that are correctly identified) while keeping background rejection high. These gains directly lead to better physics sensitivity, which means more accurate measurements and a better chance of finding new things.

2.2.3.1 Performance gains in the NOvA experiment

A quantitative evaluation of the NOvA CVN's performance demonstrates its substantial influence relative to the preceding non-CNN-based algorithms employed by the collaboration. The CVN achieved a signal efficiency of 49% for the electron neutrino (ν_e) appearance channel, which is a measurement that is very limited by statistics. This was a significant gain over the 35% efficiency of the last selection method. The CVN had a 58% success rate for the muon neutrino (ν_μ) disappearance channel. This was only a modest improvement over the 57% from the previous method, but it was done while keeping the sample purity almost the same. These results confirmed the initial use of DL, demonstrating that even first-generation CNNs could surpass highly advanced, manually optimized algorithms, especially for the most difficult classification tasks [17].

2.2.3.2 State-of-the-art performance in the DUNE experiment

The more advanced SE-ResNet-based CVN created for the DUNE experiment has set a new standard for how well it can classify things. It can use fine-grained detector information through a much deeper network architecture, which makes it very efficient at finding key signal events. The DUNE CVN's best performance metrics are as follows:

- Electron Neutrino (ν_e) Selection: 90% efficiency (94% for antineutrinos).
- Muon Neutrino (ν_μ) Selection: 96% efficiency (97% for antineutrinos).
- Overall Purity: 90% for the selection of both electron neutrinos and antineutrinos.

The results are much better than what was predicted in the DUNE Conceptual Design Report (CDR), which used a more standard classification scheme. This shows a real improvement in the physics potential of the experiment that can be directly linked to the advanced DL method. These results are corroborated by an independent student study that replicated a comparable architecture. As stated in a corrigendum to the publication, it is important to note that “This work is intended as a student investigation into statistical analysis and is not intended to address DUNE classification capabilities”[20].

2.2.4 Strategies for handling sparse data and semantic segmentation in LArTPCs

Modern CNNs do a great job of classifying data, but data from high-resolution detectors like LArTPCs is different and presents its own computational challenges. The data is very sparse, which means that the particle tracks and showers that hold the important physics information take up only a small part of the total pixel volume. This lack of data makes it very hard for computers to work with, which makes using traditional dense CNNs in a simple way inefficient and impossible to scale. To get past these problems, we need to switch from dense, computationally heavy methods to specialized sparse convolutional techniques.

2.2.4.1 Computational inefficiencies and submanifold sparse networks

The structure of LArTPC data and the design of traditional dense CNNs are fundamentally different. Standard CNNs were made to work with photographic images, where each pixel usually has useful information in it. LArTPC images, on the other hand, are mostly made up of zero-value pixels, which show empty space in the detector volume. When you use dense CNNs on this sparse data, the computations are very inefficient because they mostly involve multiplying dense matrices with inputs that are mostly zero. The submanifold dilation problem makes this problem worse by making the sparse data denser with each convolution layer, which makes the computational load grow exponentially. A quantitative comparison shows this difference: a sparse network only needs 0.044 GB of memory to process a single 192px 3-Dimensional (3D) LArTPC image, while a dense network needs 16 GB. The time it takes to process the image goes from 3.3 seconds to 0.10 seconds [21].

Submanifold Sparse Convolutional Networks (SSCNs) use operations that are best for sparsity to fix this. To ignore pixels with a value of zero, they use Sparse Convolutions. To keep output sparsity in check, they use Submanifold Sparse Convolutions, which makes sure that data doesn't get denser with each layer. These improvements make DL methods scalable and useful for the huge datasets that next-generation LArTPC detectors are expected to produce [21].

2.2.4.2 Semantic segmentation and the U-ResNet architecture

The strategic use of DL in LArTPC analysis has changed from classifying whole images to doing pixel-level semantic segmentation. This change makes it possible to understand particle interactions in much more detail by giving each pixel with a non-zero energy deposit a specific class, like track or shower [22,23]. The U-ResNet architecture is a strong hybrid model that works well for this job. It puts together the U-Net, which is good for biomedical image segmentation, and ResNet. The architecture is based on an auto-encoder, with an encoding path that learns abstract features by down-sampling the image and a decoding path that restores resolution by up-sampling the features. A key design element is the use of skip connections that link feature maps from the encoding path directly to the decoding path, keeping high-precision spatial information [21,22]. This pixel-level data is very important for event

reconstruction because it makes it easier to do hard tasks like finding vertices and grouping particles [22,23].

2.2.4.3 Context-enriched architectures and multi-task learning

To get to the next level of accuracy, you need to use the full event context. The main idea behind context-enriched learning is to give the network not only the particle of interest but also the activity going on around it. A new four-tower Siamese-type architecture has been created to handle inputs from both the specific particle cluster and the whole interaction at the same time [24]. This method has a lot of benefits for performance. For example, the simple selection efficiency for finding pions goes up by 11%, and the number of pions that are mistakenly identified as photons goes down by 8 percentage points. Additionally, multi-task learning models such as the Multiple Particle Identification (MPID) network yield concurrent probability scores for various particle types (e^- , γ , μ^- , $\pi^\pm$ and p) from a single inference pass. Occlusion Analysis applied to MPID validates that these models acquire physically significant features; obscuring the Bragg peak of a proton track or the core of an electromagnetic shower results in substantial reductions in the corresponding particle scores, thereby affirming the network's dependence on distinctive physics signatures [23].

2.2.5 Deep learning for neutrino direction and vertex reconstruction

DL can also be used for regression tasks that are important for event reconstruction, building on the successes of classification and segmentation. Conventional reconstruction techniques frequently encounter difficulties associated with high computational costs and dependence on explicit physical models. CNNs are a revolutionary, data-driven option that can learn important features directly from detector data to greatly speed up and improve the accuracy of finding neutrino direction and interaction locations.

2.2.5.1 Accelerating analysis with enhanced computational efficiency

In modern neutrino observatories, being efficient is a big plus. Recent studies show that there are huge gains in efficiency. Using GPU hardware, a CNN-based method for direction reconstruction in the IceCube experiment was up to 10,000 times faster than the traditional likelihood-based method. When using a GPU, the CNN method takes only 0.0044 seconds to process each event, while the likelihood-based method takes

44.97 seconds on a CPU [25]. Processing a high-resolution LArTPC image with a CNN on a GPU takes about ten milliseconds [26].

2.2.5.2 Improving physics reach with superior regression performance

CNNs have also demonstrated exceptional performance in regression tasks, frequently surpassing well-known traditional algorithms. In the context of likelihood replacement for giga elektronvolt (GeV) scale events in IceCube, a CNN is utilized to reconstruct the neutrino zenith angle (θ), serving as a direct replacement for standard likelihood-based methods. This approach has yielded significant improvements in angular resolution; specifically, the CNN method in IceCube achieves a Root Mean Square (RMS) reconstruction error that is 10.3% smaller than the likelihood-based method, which consequently makes the 68% containment contours narrower [25].

Consequently, the utility of CNNs is evident in radio detection applications, where an end-to-end CNN demonstrates the capability to ascertain the trajectory of ultra-high energy neutrinos with a precision of only a few degrees [27, 28]. This level of accuracy is preserved even in the context of intricate electron neutrino CC interactions, despite the substantial signal fluctuations induced by the Landau-Pomeranchuk-Migdal (LPM) effect [26]. These developments are also relevant to the identification of vertices within LArTPCs. As an illustration, the Faster-RCNN architecture, a state-of-the-art object detection network, utilizes bounding box predictions to pinpoint neutrino interaction vertices within MicroBooNE images.

The performance of this system is measured using the Intersection over Union (IoU) metric, which demonstrates the system's capability to accurately localize interaction points [26].

2.2.6 Advanced deep learning architectures in the IceCube observatory

While general CNN architectures have worked well with many types of detectors, the IceCube Neutrino Observatory has its own set of problems that need very specific architectural innovations. The detector's shape and the size of the instrumented volume require methods that go beyond regular grid-based convolutions.

2.2.6.1 Geometry-aware architectures: hexagonal CNNs

The Digital Optical Modules (DOMs) in IceCube are arranged in a triangular grid, which is not the best fit for standard CNNs. To solve this problem, hexagonal CNN architectures have been made. This is a two-step process: first, changing the hexagonal grid into a standard orthogonal grid by adding padding in the right places; second, changing the convolution kernels by setting the corner elements to zero to match the physical layout of the hexagonal sensor. The raw data is also a time series with a variable length. Nine summary statistics (like total charge and charge-weighted mean time) are used to change it into fixed-size feature vectors. The network is set up as a multi-label regression task to guess the deposited energy and direction vectors, which gives important estimates of uncertainty for each event [29].

2.2.6.2 Generative data augmentation: superresolution

The large distance between sensors in IceCube makes it hard to get enough photon signals. To get around this, a superresolution method uses generative DL to guess where photons will hit virtual sensors that are placed between real ones. This uses a VAE-UNet architecture, which means that a Variational Autoencoder (VAE) compresses the time series into a latent space and a UNet works in that space to guess what the virtual sensor data will be. This method improves the average angular resolution by about 0.8 degrees, and for events below 1 Tera Electronvolt (TeV), it can improve it by as much as 1.3 degrees, all with very little extra computational cost [30].

2.2.6.3 Hybrid methods and the galactic plane

Lastly, a hybrid reconstruction method was created to make the detector sensitive enough to see neutrinos from the Milky Way. This method combines DL with classic maximum-likelihood (MLH) methods. A neural network is trained to find the parameters that describe the complicated link between an event hypothesis and expected light yields. This is a quick way to get an idea of the PDF. By lowering energy thresholds, this hybrid method keeps more than 20 times as many events as older methods. By using it on ten years of IceCube data, scientists made a discovery; the first observation of high-energy neutrino emission from the Galactic plane, with a significance of 4.48σ for the π^0 model. This success shows how powerful it can be to combine DL's ability to recognize patterns with strict physics frameworks [31].

2.3 Foundations and Advanced Applications of GNNs

2.3.1 Theoretical basis and the scaling imperative

The strategic choice of an artificial intelligence model is determined by its foundational assumptions, referred to as inductive biases. These biases are not restrictions but are necessary limits that enable effective learning. For physical systems, which are fundamentally characterized by the interactions among their constituent entities, a model's most essential inductive bias is its capacity to represent and analyze relationships. This ability to learn from compositional structure allows for a first-principles architectural choice that reflects the basic structure of the problem domain. GNNs represent a primary option in this context due to theoretical properties that make them structurally analogous to real physical systems.

GNNs are characterized by a robust relational inductive bias. This bias, essentially a set of guiding principles, steers the learning algorithm to emphasize the relationships and interactions among entities. Consequently, the model is predisposed to favor interpretations grounded in relational structure, irrespective of individual observations. This encoding of assumptions serves to improve both sample efficiency and generalizability. This architectural design is especially well-suited for modeling physical systems. This is because graphs inherently accommodate pairwise relational structures, which directly correspond to physical frameworks where forces or decay processes establish relationships between particles [32,33].

GNNs are different from other common architectures. For example, CNNs focus on local features and spatial translation invariance, while Recurrent Neural Networks (RNNs) are designed for sequential data and temporal invariance. These specific design choices, although helpful for image and text analysis, often don't work well with the complex, non-sequential, and non-local relationships found in high-energy physics events [33].

These theoretical advantages are particularly relevant given the substantial computational demands anticipated in future experimental endeavors. The forthcoming High-Luminosity Large Hadron Collider (HL-LHC) is projected to produce an unparalleled quantity of collision data, thereby establishing a scale at which conventional reconstruction algorithms may encounter significant limitations. Within this context, GNNs present a distinct benefit by representing the entire event as a graph

structure, as opposed to a grid. Consequently, by employing message-passing mechanisms to model intricate dependencies within extensive, non-Euclidean datasets, GNNs furnish a scalable and efficient deep learning framework well-suited for such high-dimensional challenges [34].

2.3.2 The GNN design pipeline and point cloud processing

To use GNNs effectively on physical systems, there needs to be a systematic way to turn raw detector data into a graph representation that can be processed by computers. Graphs are a special type of non-Euclidean data structure that shows how a group of objects (nodes) are related to each other (edges). They don't have a regular structure like image grids. For instance, Abbasi et al. demonstrated the practical application of this by modeling the IceCube detector array as a graph, where active optical sensors are treated as nodes and their spatial proximity defines the edges, facilitating the classification of low-energy events [29]. There are four steps in the general design pipeline: (1) figuring out the graph structure, (2) choosing the type and scale of the graph, (3) making a loss function that works for the task, and (4) putting together the model by putting together computational modules [34].

2.3.2.1 Dynamic graph construction for topological inference

Processing 3D point cloud data is particularly challenging because irregular collections of coordinates do not have clear topological information. To solve this problem, the EdgeConv module builds local graphs that show how geometric objects are related to each other. EdgeConv doesn't look at each point on its own. Instead, it makes edge features that show how a point is related to its neighbors. The Dynamic Graph CNN (DGCNN) architecture expands upon this by introducing the idea of dynamic graph recomputation. The graph structure is not static; it is dynamically recalculated at each layer, with neighborhood proximity established in the learned feature space rather than solely in the initial 3D coordinate space. This allows the model to learn semantic groupings, where points that are distant in physical space but share a semantic identity can become neighbors in the network's deeper layers [35].

2.3.2.2 Graph-based regression for directional reconstruction

Dynamic graph structures, while enabling intricate feature learning, also prove advantageous in regression tasks, including the reconstruction of incoming particle trajectories. Vijayakumar et al. employed GNNs in this capacity, specifically for neutrino direction detection. This application, in contrast to classification tasks that categorize events, concentrates on predicting continuous geometric variables derived from detector data. By utilizing the graph structure to encapsulate the spatial dependencies inherent in the detected signals, the model can more directly estimate the neutrino's incoming direction compared to conventional grid-based methods, thereby illustrating the adaptability of GNNs in addressing the sparse and irregular topology characteristic of neutrino events [36].

2.3.3 Advanced architectures and benchmarks in neutrino physics

Standard models like CNNs look good, but neutrino detectors have strange shapes that need more advanced solutions. GNNs have become a better option for these difficult classification tasks, especially in places like the IceCube neutrino observatory.

2.3.3.1 Performance on irregular geometries: the IceCube benchmark

GNNs can naturally handle sparse and irregular data by modeling the detector array as a graph, thereby eliminating the need for rebinning or the approximations typically required by grid-based 3D CNNs. These models generate edge weights through a learned kernel function that evaluates the spatial distance between sensors, allowing them to adapt to the specific detector geometry without overfitting.

The advantages of this approach are clearly demonstrated in the IceCube signal classification benchmark, where the GNN model significantly outperformed a 3D CNN architecture by identifying 5.772 signal events per year, compared to only 1.815 for the CNN. This performance gap is further highlighted by the signal-to-noise ratio; the GNN achieved a ratio of 2.980, which is over three times better than the 3D CNN's ratio of 0.937. Furthermore, GNNs exhibit a distinct advantage in scenarios requiring a very low False Positive Rate, a critical requirement for effectively isolating rare neutrino events from pervasive background noise [37].

2.3.3.2 DynEdge: precise reconstruction for IceCube events

The DynEdge model represents a highly successful and specific application of Graph Neural Networks (GNNs) within the IceCube observatory. DynEdge interprets neutrino events as point cloud graphs, where each detected pulse is treated as an individual node. This architectural approach preserves the full resolution of the raw data without necessitating any grid modifications [38,39].

The model has demonstrated significant improvements in classification tasks, increasing signal efficiency by 18% at a constant background rate. Alternatively, at a fixed signal efficiency, DynEdge reduces background contamination by more than eightfold, bringing it down to less than 0.5% [38]. In terms of regression, DynEdge outperforms the state-of-the-art Retro maximum likelihood algorithm within the critical energy range of 1 GeV to 30 GeV [29]. It consistently enhances the resolution of deposited energy, zenith angle, and interaction vertex by a margin of 15% to 20% . Perhaps most significantly, DynEdge exhibits robust real-time processing capabilities. By utilizing GPU acceleration, the model achieves an average inference speed of 30.6 kHz, which is nearly twice the median IceCube trigger rate. This performance level indicates that the full online datastream can be processed in real time for the first time, marking a major milestone in the observatory’s data handling capabilities [38].

2.3.3.3 Evolving GNNs: attention mechanisms and NuGraph2

In addition to static geometries, enforcing 3D consistency is necessary to reconstruct a single 3D interaction from several 2D views. The NuGraph2 model solves this problem by using a special multi-head self-attention mechanism called “categorical cross-attention”. It also adds a Nexus Block that lets information flow between different 2D detector planes. Features are sent to virtual nexus nodes, mixed together to make a new feature set, and then sent back to the planar subgraphs. This architecture gets a cross-plane semantic consistency of 94.81%, which is a pretty good improvement over the 67.38% that models without the nexus block got [40].

2.3.4 GraphNeT: standardizing deep learning for neutrino telescopes

In the past, neutrino astronomy has made progress in fits and starts because of codebases that are only useful for certain experiments. GraphNeT is an open-source DL library that works with any type of detector. It aims to bring the field together [38].

- **Detector-Agnostic Framework:** GraphNeT takes experiment-specific information and puts it into a Detector component that can be used with any experiment. This lets core model architectures be used with data from IceCube, KM3NeT, and future projects in a flexible way, getting around the problems with traditional geometry-dependent algorithms [33,38].
- **Community Impact:** GraphNeT showed its worth in the Kaggle competition “IceCube - Neutrinos in Deep Ice.” The library’s DynEdge model was used as a starting point. Even though it was only trained on 8% of the data, it inspired a lot of competitors, and the best solutions were later added back to the library. This cycle of open competition and integration strengthens GraphNeT’s role as a collaborative platform for speeding up shared innovation in neutrino science [38].

2.4 Deep Neural Networks for Reconstructing Neutrino Events

2.4.1 The rise of deep neural networks in neutrino physics

DNNs have become an important instrument for scrutinizing the extensive and intricate datasets produced by contemporary neutrino detectors. These sophisticated computational methodologies present a prospective alternative to conventional, frequently labor-intensive, analytical approaches for the complex process of reconstructing neutrino attributes from unprocessed detector signals, in the pursuit of elucidating the most energetic phenomena within the cosmos. DNNs adeptly model complex, non-linear relationships in detector data, enabling the complete reconstruction of vital neutrino characteristics, including energy and direction, directly from recorded antenna waveforms [28].

Recent studies have rapidly progressed beyond merely demonstrating the basic reconstruction capabilities of these networks. The focus has now shifted to improving their reliability for formal scientific measurement, mainly by using advanced uncertainty quantification (UQ) and the ability to learn and fix complex physical effects. These improvements are making DNNs an important part of the basic steps that go into reconstructing neutrino events.

2.4.2 Full event reconstruction

A major application of DNNs in neutrino physics is the complete reconstruction of a neutrino's energy and path directly from raw antenna waveforms. This method gets rid of the need for fitting intermediate parameters, which speeds up the analysis process. The method has been successfully employed in simulations of various in-ice radio detector designs, demonstrating its versatility and effectiveness in handling high-dimensional, noisy data.

2.4.2.1 How well shallow detectors work

Initial research demonstrated the successful application of a VGG-like DNN architecture, characterized by iterative blocks of convolutional and pooling layers, in simulations of shallow in-ice radio detector stations. This work represented a significant advancement as it was the first time a DNN accurately predicted neutrino direction across all event topologies. This includes highly complex electron neutrino charged-current (ν_e -CC) interactions, which generate intricate radio signals due to the LPM effect a phenomenon that has historically challenged traditional reconstruction methods [28].

The performance achieved by these shallow stations proved to be robust, with the network computing energy levels at a standard deviation that meets the scientific requirements for ultra-high energy (UHE) neutrino detectors [28].

2.4.2.2 A comparison of different detector geometries

The IceCube-Gen2 Collaboration then expanded this study to assess DNN performance in both the shallow and deep components of a proposed hybrid detector station. This comparative analysis revealed a compromise in reconstruction capabilities directly linked to the physical configuration of the antennas. The results showed that the two designs had different strengths: deep components had a better energy resolution of $\sigma \approx 0.2$ in $\log_{10}(\text{Esh})$ (with Esh being the shower energy from the neutrino interaction), while shallow components had a better directional resolution of 4° compared to 7.5° for the deep components [41].

The differences in performance are directly related to the shape of the detectors. The deep stations have better energy resolution because they have more antennas, which better define the Cherenkov cone of the radio emission. Conversely, the shallow

stations achieve enhanced angular resolution by utilizing more sensitive, horizontally polarized antennas, which are crucial for precisely determining signal polarization and, consequently, the direction of the neutrino's arrival [41,42]. Additionally, the study highlighted the importance of the training dataset's structure. The improved energy resolution in the deep component analysis was probably due to using a training dataset with a uniform energy spectrum. This got rid of the biases that were seen in earlier studies at both low and high energies, which led to a more even performance across the whole energy spectrum [41]. This conclusion underscores the intrinsic link between the efficacy of these networks and the quality and content of the Monte Carlo training sets, highlighting the critical necessity for the development of comprehensive, impartial simulation libraries for forthcoming detectors. While these findings demonstrate the exceptional predictive power of DNNs, a singular optimal value is insufficient for scientific investigation; it is imperative to evaluate the uncertainties associated with these predictions.

2.4.3 Advanced methods for uncertainty quantification

The core reconstruction performance is good, but the non-Gaussian tails found in angular resolution and the energy-dependent biases show that a single optimal value is not enough. To use these predictions in formal physics observations, we need strong UQ. To have faith in a measurement and understand its statistical significance, it is important to accurately assess the uncertainty of a prediction. UQ-enabled networks are crucial for scientific integrity and provide practical benefits, such as the capacity to reject erroneous predictions and detect out-of-distribution samples that may indicate new physics or instrumental anomalies [43].

2.4.3.1 Methods for quantifying uncertainty in in-ice radio detectors

Recent progress in in-ice radio detectors has utilized intricate UQ methodologies to surpass the estimation of a singular optimal value. One well-known method is the use of conditional normalizing flows, which are a type of deep generative model that can learn complex probability distributions. This technique allows a DNN to predict the entire posterior PDF for every unique event. Instead of giving a single point estimate for the direction of the neutrino, the network makes a full probability map of the sky. This makes it easier to model the complex, non-Gaussian uncertainty contours that come with these measurements [41,42].

2.4.3.2 A thorough evaluation of UQ methodologies

The application of normalizing flows in astroparticle physics represents a facet of a broader inquiry into UQ methodologies within the physical sciences. A comprehensive examination of diverse UQ methodologies was conducted within the context of data analysis for LArTPCs, a pivotal technology in neutrino physics [43]. This study examined several notable UQ methods alongside normalizing flows:

- **Model Ensembling:** Involves training multiple instances of the same network architecture with different random initializations. The differences in their predictions show how uncertain things are.
- **Monte Carlo Dropout (MCD):** An approximation of a Bayesian neural network that makes many predictions from one network by running stochastic forward passes with dropout layers turned on during testing.
- **Evidential Deep Learning (EDL):** A method in which the network learns to predict the parameters of a prior distribution (for example, a Dirichlet distribution) that mathematically describes the posterior, giving a direct measure of uncertainty from just one forward pass [43].

The main finding from this comparison was that simple model ensembling methods usually got the best accuracy and the best calibration of output probability values for different particle classification tasks. This result, while seemingly advocating for a simpler method, likely suggests that ensembles can more effectively traverse the complex, high-dimensional loss landscapes characteristic of physics data, thus providing a more robust sampling of model uncertainty than approaches constrained by specific prior distributions. This shows that while more complex methods are powerful, simple ones can also be very useful for accurately measuring uncertainty, even though they cost more to compute during training. These new methods provide the statistical framework necessary for comprehensive analysis, while other research explores how DNNs can adapt to complex physical events previously regarded as systematic constraints.

2.4.4 Modeling complex physical and systematic effects

DNNs are not only good at core reconstruction and figuring out how much uncertainty there is, but they are also good at learning and fixing complex physical phenomena

that affect how signals are made and spread. These phenomena, usually seen as systematic uncertainties in standard analyses, can be directly modeled by a DNN trained on realistic simulations, allowing it to learn the basic physics of the detector system and turn a systematic uncertainty into a new observable.

2.4.4.1 Consequences of birefringence

A notable example is the first study utilizing a DNN to measure the effect of birefringence an anisotropic property of the South Pole ice that alters radio wave polarization on event reconstruction [42]. The study yielded unexpected yet physically illuminating results. When a DNN was trained on data that included birefringence effects, it showed a small drop in directional resolution but a big rise in energy resolution. The physical significance of this result is that the network has gained the capacity to employ a new piece of information introduced by the effect. Birefringence causes a delay in time between the different polarization eigenstates of the radio wave. This delay is directly related to how far away the neutrino interaction vertex is. The DNN used this information to improve vertex reconstruction, which led to a more precise energy measurement. This is an important discovery because it shows that the network is not just fitting a function; it is also finding and using a physical correlation the linear relationship between polarization time-delay and propagation distance that was not clearly encoded.

2.4.4.2 How emission models affect sensitivity

DNNs show a strong sensitivity to the subtle differences in the basic physics models used to make their training data. A study assessed the effectiveness of a DNN trained on two different Askaryan emission models: the simpler frequency-domain Alvarez2009 model and the more precise time-domain ARZ2020 model [28]. The conclusion showed that the DNN did better when it was trained on the more complex and physically accurate “ARZ2020” data. This means that the network was able to learn and use the extra structural information that was present in the more accurate physical simulations, such as the radio pulse shape’s asymmetries that simpler models don’t have. This result has important effects on simulation-based inference because it shows that the accuracy of the physics generators used to create training data limits how well the reconstruction process can work. These improvements represent a paradigm shift, changing DNNs from basic tools for reconstructing events into

important parts of the analytical framework that can learn about complex physical effects and give the precise UQ needed for scientific measurements of discovery-class.

2.4.5 Additional methodological advancements

The ongoing challenge of accurately reconstructing neutrino events has led to the development of sophisticated computational techniques, frequently merging traditional physics-based models with cutting-edge ML methodologies [44]. Neutrino detectors, like the IceCube Neutrino Observatory, use strict statistical and analytical methods to accurately separate the rare astrophysical neutrino signal from the common atmospheric noise [45]. Recent literature indicates a significant trend towards enhancing reconstruction accuracy through the integration of hybrid ML and statistical analysis, while simultaneously prioritizing computational efficiency, often by minimizing the necessity for extensive simulated training data. A significant area of methodological advancement involves the precise enhancement of statistical analyses through ML algorithms to optimize event observables [45]. Enhancements to IceCube point source searches employ a DNN to approximate muon energy at the detector's entrance, serving as a surrogate for neutrino energy. This method gives unbiased estimates over the whole muon energy range and improves energy resolution by up to 40% [45]. To parameterize the estimated angular uncertainty, a BDT is used. This makes the spatial PDF independent of event declination again, even at sub-TeV energies [45].

Supplementary research demonstrates the efficacy of a hybrid methodology that amalgamates the domain expertise intrinsic to MLH techniques with the universal approximation capabilities of DL [44]. This hybrid methodology employs a generative neural network to estimate the intricate, high-dimensional likelihood function, facilitating the explicit integration of physical invariances and detector attributes, including precise detector geometry and the attenuating influence of the dust layer. This new generative method greatly improves the angular resolution for cascade events, going beyond what both traditional maximum likelihood and only CNN methods can do [44].

ML is essential for classifying neutrino events, especially when detector simulations are computationally intensive [46]. Transfer learning offers a feasible approach to alleviate the substantial costs associated with training CNN models on extensive

simulated interactions [46]. A study showed that fine-tuning a pre-trained ResNet18 network that was originally made for classifying photographic images using a small set of LArTPC neutrino images made it work better than networks that were trained from random initialization with the same number of samples [46].

Various statistical ML classifiers, including Logistic Regression (LR), k-nearest neighbors (KNN), Support Vector Machine (SVM), and Decision Tree (DT), have been effectively utilized to identify intricate nonlinear astrophysical phenomena, such as fast neutrino flavor conversions (FFCs) in core-collapse supernovae (CCSN) and neutron star mergers (NSMs) [47]. Notably, simpler algorithms like LR, especially when improved with a polynomial transformation of degree $n=2$, showed better generalization accuracy on real CCSN simulation data. This shows how important it is to pick models that match the complexity of real-world datasets [47]. An analysis employing RNNs specifically Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), and a hybrid LSTM+GRU model on IceCube Observatory data revealed that while RNNs demonstrated competitive accuracy, they consistently yielded higher Mean Angular Errors (MAEs) compared to the baseline GNN model [48]. The noted performance discrepancy suggests that RNNs, although skilled in modeling the sequential attributes of event data, may struggle to accurately depict the intricate spatial relationships effectively modeled by GNNs [48]. The current academic environment emphasizes the application of advanced, adaptive ML techniques to improve all phases of neutrino research, from fundamental reconstruction to advanced classification tasks.

2.5 Transformer Architectures in Reconstructing Neutrino Events

2.5.1 How transformer architecture can be used for visual tasks

The Transformer architecture represents a pivotal evolution in sequence modeling. Originally intended for natural language processing (NLP), its strategic importance lies in its core mechanism, self-attention, which offers a strong and highly parallelizable alternative to the recurrent and convolutional layers that have traditionally dominated DL. This method lets a model look at a whole data sequence at once and figure out how each part affects all the others, which helps it find complex, long-range connections. Vaswani et al. wrote the important paper “Attention Is All You Need” [7] in which they laid out the basic ideas behind the Transformer as shown

in Figure 2.1. The self-attention method lets the model find global connections between inputs, no matter how far apart they are in a sequence. Self-attention creates a representation of the sequence by directly linking several points within the same sequence. This is different from recurrent models, which process input one at a time.

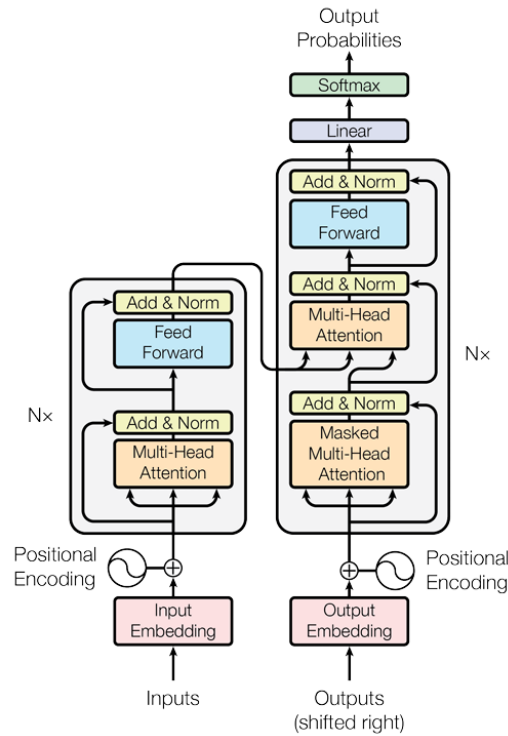


Figure 2.1 : The Transformer Model Architecture. Encoder (left) and Decoder (right) [7].

To get an output, which is a weighted sum of the values, you need to match a query with a set of key-value pairs. The alignment of the query with the key determines how much weight each value gets. This lets the model focus on the most important parts of the input data for a specific task. At first, it was thought that using self-attention directly on high-dimensional data like photos was not possible because it costs a lot of computing power compared to the number of input pixels. The Vision Transformer (ViT) got around this problem, which was a very important change that made the architecture useful for computer vision [49]. The ViT looks at an image as a series of smaller, fixed-size pieces. After that, these patches are treated as tokens in a phrase, which lets a normal Transformer encoder work with them. This patch-based method made the self-attention mechanism scalable for image recognition tasks and demonstrated that a pure transformer, with minimal image-specific inductive biases,

could outperform state-of-the-art convolutional networks when pre-trained on sufficiently large datasets. The effective use of Transformers in general computer vision has led to their study in specialized scientific areas, revealing important uses in particle and neutrino physics.

2.5.2 Reasons for using transformers in neutrino data analysis

For neutrino physics to move forward, it is very important to come up with new ways to reconstruct and find neutrinos. Neutrino observatories collect data that has its own set of problems, like being very high-dimensional, very sparse, and having complicated event topologies that need to be looked at to get useful physical quantities. Because Transformer models can naturally capture global dependencies in sequential or graph-structured data, they are a good way to deal with these problems. The primary goals of utilizing Transformer models in neutrino physics are varied, including the mitigation of systematic uncertainty and the implementation of fundamentally innovative architectural approaches.

For future high-precision experiments like DUNE and Hyper-Kamiokande, it is very important to know exactly what particles are doing at the neutrino interaction vertex. Standard analytical methods for measuring vertex activity often depend on arbitrary nuclear models to make sense of detector data. This could lead to big mistakes in the final reconstruction of neutrino energy. DL, especially the Transformer architecture, offers a promising model-agnostic methodology. These models can figure out the number, type, and energy of final-state particles by looking at complex patterns in simulated data, without using any existing physics models [50].

Because UHE neutrinos are so rare, we need to make very sensitive detector trigger systems. To get the most out of scientific work, observatories like Radio Neutrino Observatory Greenland (RNO-G) and the planned IceCube-Gen2 need advanced triggers that can pick up weak neutrino signals while getting rid of background noise. Transformers are being looked into for this purpose because they can make triggers that are much more sensitive than the ones we have now, which could greatly improve the rate at which neutrinos are detected. This enhancement enables the identification of fainter signals that might otherwise be disregarded, thereby directly expanding the scientific scope of these studies [51].

In direct competitive tests, transformer-based designs have clearly outperformed other top ML models, like CNNs, GNNs, and RNNs. In the public Kaggle competition “IceCube – Neutrinos in Deep Ice,” Transformer-based models were the most popular among the best ways to figure out where neutrinos were coming from. This accomplishment highlights that the Transformer architecture represents an innovative and highly effective approach for examining the unique geometric and temporal attributes of IceCube data [52,53].

These strong reasons have led to a lot of research into specific uses of Transformer models in many areas of neutrino physics.

2.5.3 Implementations and changes in neutrino physics

The Transformer architecture’s flexibility has made it useful for a number of specific tasks in neutrino physics, such as accurately reconstructing particle trajectories, analyzing complicated interaction events, and making real-time detection triggers for next-generation observatories.

2.5.3.1 Directional reconstruction in IceCube

The Kaggle competition “IceCube – Neutrinos in Deep Ice” was a great way to test out new ML architectures. Transformer-based models often came out on top [53]. The winning entries showed new ways to use attention mechanisms on the sparse, irregular point-cloud data that is common in neutrino detectors. They often did better than the standard GNN baseline. The best solutions used a lot of different architectural styles to get the best performance. The winning solution used a lightweight hybrid model that combined EdgeConv layers with a Transformer encoder. This took advantage of both architectures and made training more efficient by using techniques like mixed-precision training and sequence bucketing [52]. The second-place solution used a more real Transformer method that was based on the BEiT architecture. Major improvements included a Fourier encoder for handling continuous input variables and a unique attention bias based on the Minkowski space-time gap. This made it easier to encode causal links between detector pulses. This model examined the utilization of relative position embeddings, a technique that enhances the self-attention mechanism by explicitly incorporating the relative distances among input elements [54], a feature lacking in the original Transformer architecture [52]. A later model, ISeeCube, showed that a simplified Transformer architecture using the TorchScale (BEiT-3) framework

could perform as well as or better than the second-place solution while using much less complicated code. This shows how effective a well-optimized, simplified Transformer can be [55].

2.5.3.2 Breaking down vertex activity

To deal with overlapping-sparse pictures at the neutrino interaction vertex, a decomposing transformer has been made [50]. This model deals with the problem of distinguishing the signatures of many low-momentum particles that can't be told apart by normal methods. The model works in steps, starting with predicting the kinematic properties (type, energy, direction) of the most energetic particle in the event. Then, the decoder predicts the next most energetic particle by breaking the composite event down into its parts in order of energy. Adding a fully differentiable Generative Adversarial Network (GAN) as an image generator makes this system even better. The GAN can make fake particle pictures based on the kinematic properties that the transformer predicts. Because the GAN is differentiable, its output can be directly compared to the original detector image. This lets gradient-descent optimization minimize any differences. This creates a feedback loop that makes it easier to improve the transformer's initial predictions, which in turn makes the reconstructed and original event images match up better [50].

2.5.3.3 Finding and starting UHE neutrinos

Currently, Encoder-exclusive Transformer models are being actively developed to serve as a triggering system for UHE neutrino detectors [51]. You can use these models in two different ways: as a second-stage trigger, where they look at a data stream that has already been chosen by a simpler, lower-level trigger system, or as a first-stage trigger, where they look at a continuous raw data stream coming directly from the antennas. Performance evaluations have shown a lot of promise. Compared to the baseline system, a Transformer model used as a second-stage trigger on phased array data increased the rate of neutrino detection by 50%. A smaller and more efficient model still got a 40% improvement. The model achieved an impressive 128% increase in detection rate as a first-stage trigger on LPDA data, but it came at a much higher cost in terms of computation. These results highlight a substantial trade-off between sensitivity and computational expense, particularly for models intended to function on power-constrained hardware at remote detection sites [51].

The Transformer architecture is very flexible, as shown by the wide range of applications it can be used for, from post-hoc reconstruction to real-time event detection. This means that it needs to be directly compared to other neural network frameworks.

2.5.4 A comparison with other architectures

For a full assessment of the pros and cons of using Transformer models in neutrino physics, it is necessary to compare them to other types of neural networks, such as RNNs, CNNs, and GNNs. This comparative analysis clarifies the specific contexts in which Transformers offer distinct performance or efficiency advantages.

Transformers offer significant advantages in parallelization. They process entire sequences at once, avoiding the inherently sequential and thus slower nature of RNNs [7]. Furthermore, they excel at handling long-range dependencies because self-attention connects all positions with a constant path length ($O(1)$), making it easier to learn dependencies between distant elements in a sequence, a key challenge for RNNs [7]. However, a trade-off is that the base architecture does not inherently process sequential order; this must be injected via positional encodings [7].

A key advantage of Transformers is learning without inductive bias. They lack the built-in locality and translation equivariance of CNNs. While this can be a weakness on small datasets, it allows them to learn more general patterns directly from data when pre-trained on massive datasets, ultimately outperforming CNNs [49]. Regarding the demonstrated efficacy of UHE neutrino triggering, a Transformer model exhibited a 39% enhancement in performance relative to a preceding CNN-based model [51]; furthermore, the Kaggle competition results corroborated the superiority of Transformers over CNNs [52]. The principal drawbacks, however, are data inefficiency, necessitating considerably larger training datasets to achieve robust generalization compared to CNNs [49], and computational expense, given that superior performance may entail a substantially greater number of floating-point operations (FLOPs) than those required by comparable CNNs [51].

In terms of performance and speed, Transformer-based solutions in the IceCube Kaggle competition outperformed the GNN (DynEdge) baseline. The 2nd place solution noted that for a given computational budget, their Transformer model achieved a significant improvement in angular resolution over a DynEdge model [52].

However, GNNs are naturally suited to the irregular geometry of neutrino detector data. Some top solutions found success by creating a hybrid architecture, using a GNN component like EdgeConv to inform a Transformer, suggesting a synergistic relationship rather than pure replacement [52].

This analysis highlights that while Transformers offer compelling performance advantages, their successful application depends on navigating their specific data and computational requirements, which in turn informs current challenges and future research directions.

2.5.5 Current limitations and future research opportunities

Transformers have a lot of potential for analyzing neutrino data, but it's important to be aware of the problems that already exist so that future research can be guided and their full potential can be realized in experimental physics. Several essential fields require further investigation to overcome existing limitations and improve the applicability of these models.

The high computational cost, measured in FLOPs, of large Transformer models is still a major problem. This is especially true for real-time applications like online event triggering, where algorithms have to run on hardware with limited power at remote detection sites. Future research ought to focus on developing more efficient Transformer variants or hardware-specific adaptations to facilitate the extensive implementation of these powerful models [51]. The quest for superior model designs is an ongoing effort. Research from the IceCube Kaggle competition shows that combining Transformers with GNN parts, like EdgeConv, can work very well. More research is needed to find the best model depth and width ratios for some physics tasks [52]. Subsequent research has examined the simplification of complex structures while maintaining performance by assessing various components, including those from the BEiT-3 framework and the application of Register Tokens [55].

One of the most important things about Transformers is that they need a lot of data to learn from before they can be used because they don't have enough inductive bias. The requirement for extensive, frequently simulated datasets presents a logistical challenge. A crucial area for future research is the development of more effective self-supervised pre-training methods, like masked patch prediction, to reduce dependence on large labeled datasets and improve efficiency when training from the start on small

datasets [49]. Transformers are useful for directional reconstruction and event decomposition, which makes them useful for more tasks in computer vision and particle physics. Future research opportunities include the use of Transformers for event segmentation, reducing particle pile-up effects in high-intensity environments, and expanding their use from directional reconstruction to precise energy reconstruction [49,50,55].

In conclusion, Transformer structures represent a dynamic and rapidly evolving field in the analysis of neutrino physics data. Their demonstrated performance and adaptability signify a significant enhancement over numerous previous methodologies. The physics community can fully use these models to push the boundaries of scientific discovery if they carefully address the current issues with computing costs, architectural design, and data efficiency.

3. EXPERIMENTAL SETUP AND DATASET

3.1 Dataset Selection and Rationale

In neutrino physics research, the structural representation of detector data is the fundamental element determining the architecture of the artificial intelligence model to be used. An examination of the literature reveals that classifiers developed for the NOvA experiment represent detector activity in the form of X-Z and Y-Z views, utilizing a grid structure resembling grayscale images [17]. Similarly, DUNE experiment simulations employ fixed 500x500 images where pixels represent energy deposition [20]. Such fixed-size and image-based data structures have led to the predominant adoption of CNN-based approaches in the literature [20,46].

However, in this thesis study, the dataset presented within the scope of the “IceCube-Neutrinos in Deep Ice” competition has been preferred [8]. The basis of this preference lies in the compatibility of the data’s natural structure with the Transformer architecture, the magnitude of the data volume, and the advantages of comparability with the literature. These rationales are detailed below.

3.1.1 Structural compatibility with transformer architecture

The Transformer-based approach proposed in this study requires processing the data as a sequential series (sequence) rather than a spatial image. Unlike the image formats in other experiments, the selected IceCube dataset possesses a variable-length pulse series structure. Each event is represented by a series of pulses detected by the optical modules (DOMs) in the detector, and each pulse includes 6 basic features: the sensor’s spatial position (x, y, z) , time (t) , charge (q) , and an auxiliary flag indicating signal quality [8].

This data structure creates an ideal foundation for Transformer architectures, primarily due to its inherent sequential processing capability. Specifically, events in the dataset vary depending on the pulse count; while most events contain fewer than 200 pulses, this number can reach up to tens of thousands. This structure can allow the data to be

fed directly into Transformer encoders as a matrix of size $N \times P$ (Number of Pulses $\times$ Number of Features), eliminating the need for the image rendering process typically required by CNN models, which can lead to data loss [55].

3.1.2 Comparability and dataset characteristics

The reliability and comparability of results in scientific research are fundamentally rooted in the standardization of the underlying dataset. The dataset selected for this study offers significant advantages, particularly regarding its scale and representational accuracy. Containing over 138 million simulated neutrino events across a broad energy spectrum (100 GeV to 100 Peta Electronvolt (PeV)), the dataset provides the necessary statistical power to satisfy the big data requirements of deep learning models [53, 55]. Furthermore, by incorporating realistic detector conditions such as noise pulses and events failing local coincidence (LC) criteria (marked by auxiliary flags) the dataset ensures that the developed model is tested against practical constraints rather than idealized signals alone [53].

3.2 The Properties of Neutrinos

To develop an effective DL model on this selected dataset, it is essential to first understand the physical origin of the data, specifically the nature of neutrinos. To tackle the difficult problem of reconstructing neutrino direction, one must first have a solid understanding of the particle's basic characteristics. Neutrinos are basic particles that fall into two groups: fermions (spin 1/2) and leptons. They are electrically neutral. There are three different types of active neutrinos: the electron neutrino (ν_e), the muon neutrino (ν_μ), and the tau neutrino (ν_τ). These correspond to the weak eigenstates, which are represented by $|\nu_\ell\rangle$, that are created in weak interactions with their charged lepton partners. The discovery of neutrino oscillations, the process by which neutrinos change flavor as they travel, offered the first conclusive evidence for physics beyond the original Standard Model, which had assumed neutrinos were massless. This means that the mass is not zero, but it is very small [56].

Neutrinos primarily interact with other matter through the weak nuclear force and gravity, which also govern their production in weak decays and their detection through charged current processes. A significant outcome of this interaction mode is an extraordinarily diminutive interaction cross-section. Neutrinos can pass through a lot

of dense matter, like the whole Earth, almost without any problems because they don't interact very strongly. Because they are so hard to find, it is very hard to observe them. Since they are very difficult to detect, meticulous calculation and detection mechanisms must be developed to increase the chance of capturing rare interaction signatures. [56].

3.3 The IceCube Neutrino Observatory

This research employs a dataset derived from simulation data supplied by the 'IceCube-Neutrinos in Deep Ice' competition, orchestrated by the IceCube Neutrino Observatory. The Amundsen-Scott South Pole Station in Antarctica has a cubic-kilometer particle detector. The detector is buried deep in the Antarctic glacier, where the ice's high purity and optical clarity make it a good Cherenkov medium for finding particles [8, 57].

The IceCube detector has a high triggering rate of about 3 kHz, which is mostly caused by muons in the atmosphere [58]. This means that the raw data stream is about 1 TB per day [59]. The detector logs almost 100 billion events every year, but the number of atmospheric neutrinos it finds is much lower, at about 100,000 events per year [58, 59].

The main thing that IceCube does is find Cherenkov radiation. When a high-energy neutrino hits an atomic nucleus inside or near the detector volume, it makes relativistic secondary charged particles like muons, electrons, or taus. Cherenkov radiation is a blue light cone that comes from charged particles moving through ice faster than the phase velocity of light in that medium. By looking at the shape and timing of this light emission, you can figure out the energy, direction, and flavor of the neutrino that made it [57].

3.3.1 Detector geometry and components

The physical structure of the observatory consists of a three-dimensional array of sensors embedded deep within the ice sheet, covering an instrumented volume located between 1,450 and 2,450 meters below the surface. The fundamental sensing unit of the detector is the DOM, which serves as an independent data acquisition system housed within a pressure-resistant borosilicate glass sphere. Each DOM is equipped with a downward-facing 10-inch Hamamatsu R7081-02 PMT and integrated

digitization electronics [8, 57]. As shown in Figures 3.1 and 3.2, the array has 5,160 DOMs distributed across 86 vertical cables, or strings, arranged in a specific spatial pattern. In the main array, the strings are arranged in a hexagonal grid, with horizontal spacing of about 125 meters, and the DOMs on these strings are spaced vertically about 17 meters apart [58]. In addition, the detector includes a more densely instrumented sub-array, called DeepCore, designed to improve detection capabilities. Within this area, special strings are used, with a vertical DOM spacing reduced to 7 meters, to increase sensitivity to lower-energy neutrinos [58].

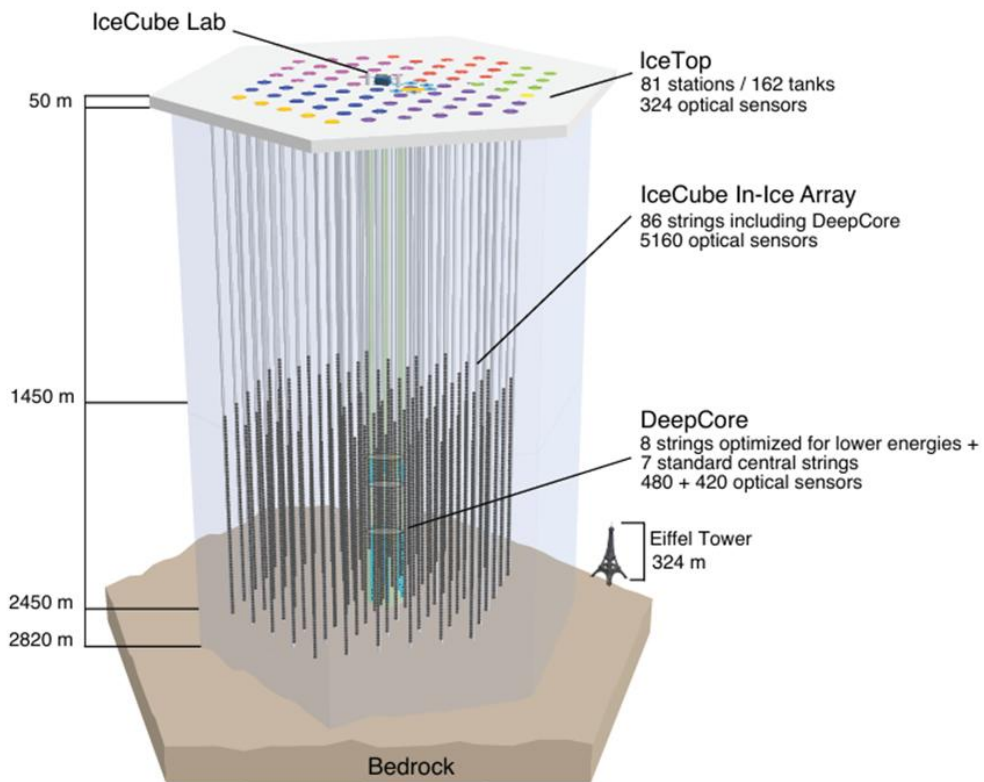


Figure 3.1 : Schematic View of the IceCube Neutrino Observatory [58].

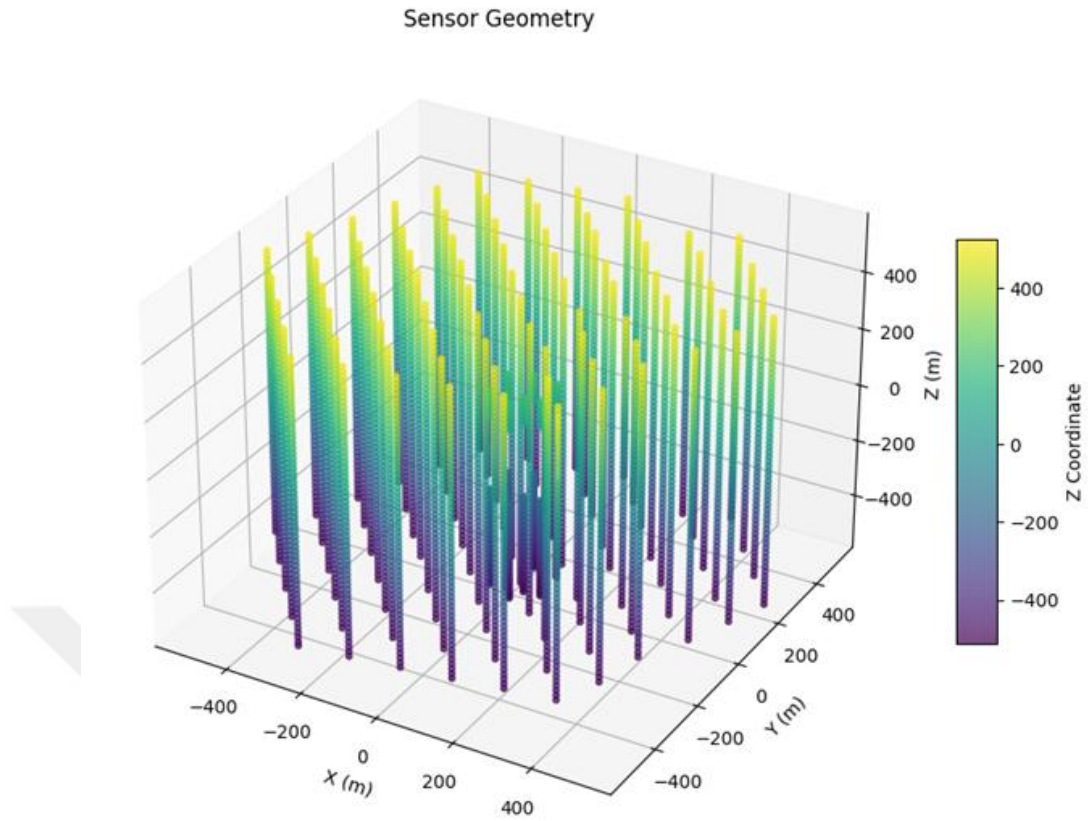


Figure 3.2 : Sensor Geometry.

3.4 The Principle of Detection is Cherenkov Radiation

The main problem with neutrino astronomy is that the particle is very hard to find. Neutrinos can pass through the whole Earth without leaving a trace, which makes finding them a very difficult technical task. The IceCube Observatory solves this problem by turning a cubic kilometer of dead Antarctic ice into an active particle detector. Cherenkov radiation is a physical phenomenon that connects the invisible neutrino to the real data that the sensor array picks up. This principle lets the weak, short-lived signature of a neutrino interaction be turned into a measurable signal. This means that a gigaton of ice can be turned into the telescope, which can show a neutrino's energy, direction, and flavor from the way light is captured [59].

3.4.1 The first interaction: from neutrino to lepton

A high-energy neutrino that has traveled across space may, on rare occasions, interact with the nucleus of a water molecule in the Antarctic ice. This interaction is what caused the detection event to happen. In a CC interaction, the neutrino creates a secondary charged lepton (like a muon for a muon neutrino) that moves on at speeds

close to the speed of light. On the other hand, a Neutral-Current (NC) interaction happens when the neutrino gives some of its energy to the nucleus, which only makes a hadronic shower [59].

In the CC case, the secondary lepton takes a lot of the energy from the parent neutrino. This amount can be anywhere from 50% at lower energies (around 10 GeV) to 80% at higher energies. The rest of the neutrino's energy goes to the nucleus it hit, which causes a localized burst of particles called a hadronic shower. The high-energy lepton and the hadronic shower both help make light, which is what makes the event visible to the IceCube array [59].

3.4.2 The cherenkov effect: an optical shockwave in ice

The Cherenkov effect makes it possible to see the path of the secondary lepton. This phenomenon is the fundamental detection mechanism and transpires under a specific condition: the charged particle must exceed the speed of light in the surrounding medium. The speed of light in the ice of Antarctica is only about 75% of its speed in a vacuum. A neutrino interaction can easily make a high-energy lepton go faster than this local speed limit [59].

The light that comes out is like a sonic boom in the optical world. When the particle moves faster than its own electromagnetic field, it makes a coherent shockwave of photons. This radiation spreads out in a clear cone from the path of the particle. Most of it is blue and near-ultraviolet (UV) light. As this cone of light travels through the very clear ice, it gets bigger. This lets its signature be picked up by many sensors that are far apart, which sends information about the direction and energy of the particle that made it [59].

3.4.3 Signal transduction: from photons to digital waveforms

The final stage of the detection chain involves the transformation of faint Cherenkov light into digital signals, a critical process executed by the 5,160 DOMs comprising the IceCube array [59]. This procedure begins with photon-to-electron conversion; as Cherenkov photons strike a DOM, the internal PMT captures them and utilizes the photoelectric effect to convert the energy of each incident photon into a subtle electrical pulse [59]. Following this, the DOM's internal electronics process these analog signals, meticulously recording the current pulse's morphology and

subsequently converting it into a timestamped digital waveform. To facilitate precise event reconstruction, the local clocks that generate these timestamps are calibrated to a nanosecond (ns) precision [59]. This waveform, in turn, encapsulates the intensity and timing of the light detected by each individual sensor, thereby serving as the fundamental raw data for all subsequent analyses. The collection, filtration, and systematic examination of these individual digital waveforms across the extensive detector array are, therefore, the critical steps in reconstructing the original neutrino's trajectory and energy [59].

3.5 Event Topologies and Signatures

The first thing that determines the topology of the Cherenkov light produced in the IceCube detector is the flavor of the neutrino and the type of interaction it has, which can be either CC or NC scattering. These resulting light patterns can be divided into two main groups, each with its own set of features. These different topologies are the main things that are used to sort events, measure energy, and figure out which way something is going. To know what the detector can and can't do, it's important to look closely at its properties. This is what this proposed research is based on [16].

3.5.1 Track topologies: the muon signature

Track topologies are the most clear and unambiguous signs of muon neutrino interactions. They are also very important for neutrino astronomy because they have great pointing resolution. A track is a long, straight line of light that is made when a high-energy muon moves through the detector volume as illustrated in Figure 3.3. The CC interactions of muon neutrinos (ν_μ) are mostly what make these signatures. The interaction physics say that a hadronic cascade happens at the vertex, but the track signature that is most common in IceCube is usually an isolated, through-going muon. This is because the muon's production vertex is usually outside the instrumented volume. The main benefit of track events is their geometric range. The muon's path can go for several kilometers, which gives it a long lever arm that lets you reconstruct the direction with much better angular resolution than you can with cascades. The muon's mass, which is about 200 times that of an electron, is what makes this range so long. Because muons are heavier than electrons, they lose very little radiative energy (Bremsstrahlung) and act like minimum ionizing particles. This lets them go through

kilometers of ice without slowing down much. This great pointing ability, on the other hand, is not as useful when it comes to the other main event class, which puts its energy into a much smaller space [16, 56].

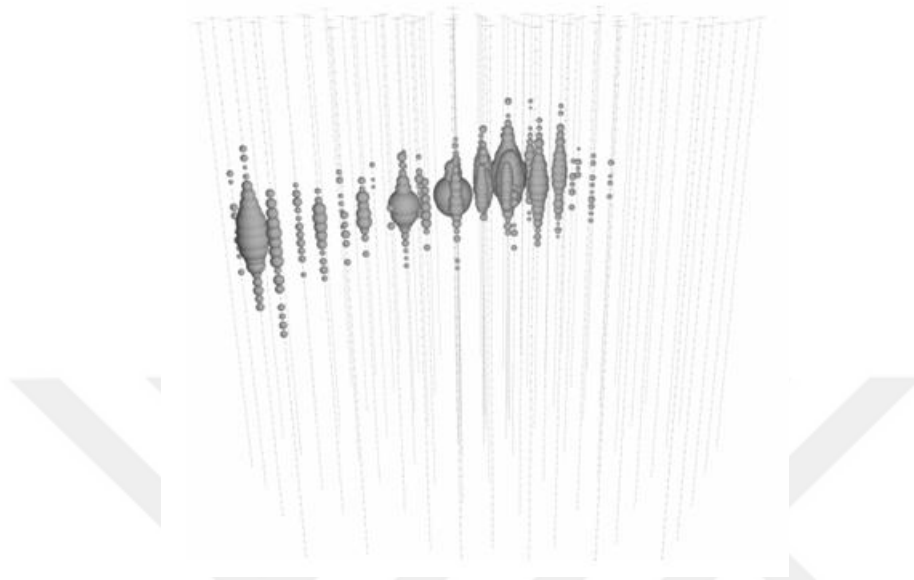


Figure 3.3 : A Muon Track Topology Observed in IceCube [16].

3.5.2 Cascade topologies: the shower signature

Most of the interactions between neutrinos in IceCube create shower or cascade topologies. This important but more complicated event category includes all NC events. To fully understand the neutrino flux, it is important to accurately reconstruct this category. A cascade is a burst of light that looks like a point and is roughly spherical as depicted in Figure 3.4. It happens when a shower of particles is contained. The electromagnetic or hadronic showers that are underneath them are about 10 meters long. The IceCube array usually can't see this scale because the sensor strings are spaced 125 meters apart horizontally, making the event look like a point source of light. The confined nature of cascades is due to fast energy dissipation mechanisms, which is different from tracks. Because they have a low mass, electrons cause electromagnetic showers by releasing a lot of energy in a short amount of time. Tau leptons also decay almost immediately, even though they are very heavy. This is because they have a very short lifetime, which stops them from traveling far before making a shower. Cascades are the hallmark of CC interactions involving both electron neutrinos (ν_e) and tau neutrinos (ν_τ), in addition to all-flavor NC interactions. At energies of PeV and higher, CC ν_τ interactions can create a unique “Double Bang”

topology that can be resolved. Otherwise, they are put together as single cascades. These events have a near-pointlike topology, which means that no matter where they come from, there is an inherent trade-off between great energy calorimetry and difficult directional reconstruction [16, 56].

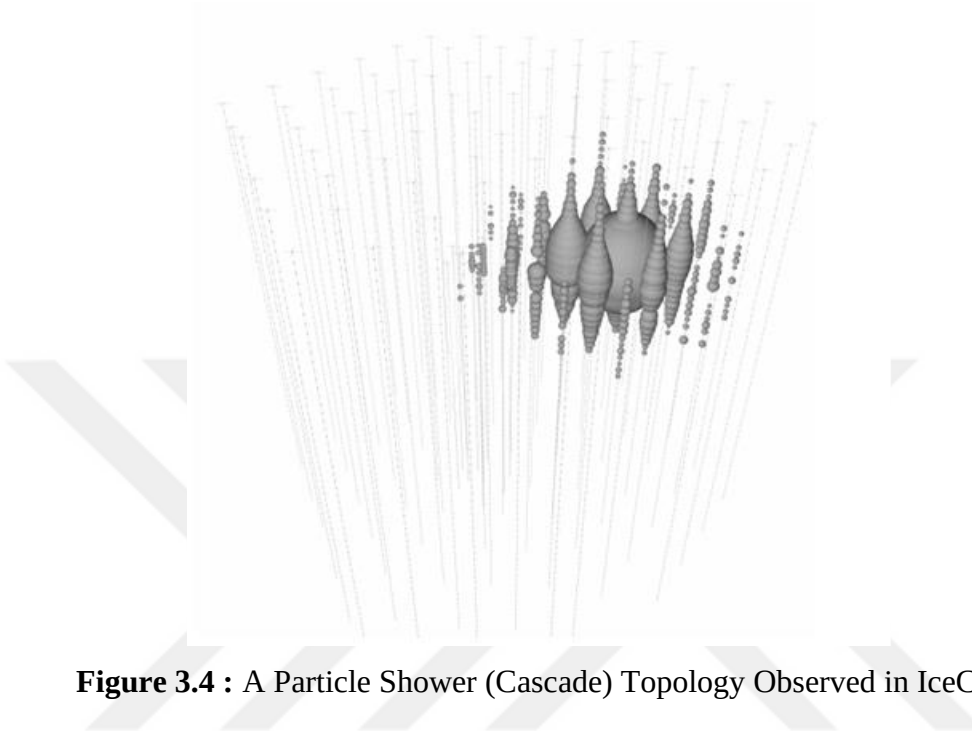


Figure 3.4 : A Particle Shower (Cascade) Topology Observed in IceCube [16].

3.5.3 The reconstruction challenge and the drive for research

The primary analytical challenge addressed in this research stems from the inherent reconstruction trade-offs associated with cascade events. While cascades provide a robust calorimetric signal with the majority of the shower’s energy contained within the detector volume their nearly spherical morphology complicates the determination of their arrival direction. This directional uncertainty currently limits the utility of cascades in identifying the astronomical sources of high-energy neutrinos [16]. The divergence between the two primary event classes underscores the current state of the field: tracks offer precise directional information but typically only provide a lower limit on initial neutrino energy when muons enter or exit the detector, whereas cascades offer superior, self-contained energy measurements at the expense of directional resolution [16].

Currently, reconstructing the direction of cascade events remains a significant obstacle; for events exceeding 100 TeV, the optimal angular resolution is approximately 15 degrees. This fundamental limitation is a direct result of event

topology, as the light emitted from a compact, quasi-symmetric shower offers few geometric cues for precise directional inference [16].

3.6 Dataset Structure and Event Representation

The dataset analyzed in this research originates from the “IceCube - Neutrinos in Deep Ice” competition held on Kaggle. There are a number of events in the data, and each event is a triggered time window with a series of photon detections.

3.6.1 Data format and sparsity

The observatory makes a lot of raw data, so the dataset is stored in the Apache Parquet format, which lets you store and compress data in columns. There are more than 5,000 sensors in the detector, but only a few of these DOMs go off when a neutrino event happens. This is why the data isn’t saved as dense tensors. Instead, it’s saved as pulse series (sequences of different lengths) that go with each event_id [8].

3.7 Input Variables and Feature Space

The dataset characterizes each recorded event as a temporal sequence of pulses, where each individual pulse represents a digital waveform captured by a specific DOM. These pulses are defined by four primary scalar features that serve as the fundamental input variables for the reconstruction algorithms. Primarily, the DOM_id acts as a spatial index that uniquely maps each pulse to a specific sensor within the detector array. By referencing the complementary sensor_geometry.csv file, this ID is converted into Cartesian coordinates in meters. As illustrated in Table 3.1, the origin (0,0,0) of this coordinate system is situated approximately 2,000 meters below the surface, aligned with the geometric center of the IceCube array [58].

Temporal topography is defined by the time variable, which records the arrival of the photon pulse at the DOM in ns. To maintain array-wide synchronization, the Reciprocal Active Pulsing Calibration (RAPCal) system ensures that each DOM’s independent clock is aligned with a surface master clock, achieving a timing precision within a ns [58]. Furthermore, the charge variable represents the signal magnitude, measured in photoelectrons (p.e.). The DOM utilizes two distinct digitization systems to acquire this signal: an Analog Transient Waveform Digitizer (ATWD) for high-precision waveform capture and a fast Analog-to-Digital Converter (fADC) for signals

of longer duration [58]. The charge value increases proportionally with light intensity, serving as a primary metric for calorimetric energy deposition.

Finally, the auxiliary flag is a boolean variable used as a signal quality indicator, reflecting the LC logic of the detector’s data acquisition system [58]. A value of False indicates a Hard Local Coincidence (HLC) hit. This happens when a DOM detects a signal close in time to its nearest or next-nearest neighbors. These HLC hits, which usually provide detailed waveform data, are likely caused by physical particle interactions, not background noise.

Conversely, a value of True indicates a Soft Local Coincidence (SLC) hit. These isolated pulses fail to meet the neighbor coincidence criteria; while they may contain physical information, they predominantly consist of background noise, such as the radioactive decay of Potassium-40 within the DOM’s glass housing [58].

Table 3.1 : Presents a Sample of the Sensor Geometry Data.

Sensor_id	x	y	z
0	-256.14	-521.08	496.03
1	-256.14	-521.08	479.01
2	-256.14	-521.08	461.99
3	-256.14	-521.08	444.97
4	-256.14	-521.08	427.95

Table 3.2 shows an example of the time, charge, and extra information that is in the training files.

Table 3.2 : Presents a Sample of the Parquet Data Structure Read from the Train Folder.

Sensor_id	Time	Charge	Auxiliary
3918	5928	1.325	True
4157	6115	1.175	True
3520	6492	0.925	True
5041	6665	0.225	True
2948	8054	1.575	True

3.8 Target Variables: Directional Reconstruction

The regression task for the dataset’s main goal is to find out where the neutrino came from. The ground truth for each event is a unit vector that shows where the particle is going.

As shown in Table 3.3, the direction is represented in the dataset using a spherical coordinate system defined by two primary angular variables. The first variable, the Zenith Angle (θ), describes the angle that the z-axis makes with the vertical, ranging from 0 to π radians. Within this coordinate convention, a zenith angle of 0 signifies a particle arriving from directly above, whereas a zenith angle of π indicates an arrival from below. Complementary to this, the Azimuth Angle (ϕ) represents the orientation within the horizontal xy-plane. This angle is measured from the grid East (x-axis) and is aligned with the Prime Meridian, which corresponds to grid North [58].

Table 3.3 : Presents a Sample from the Train Meta Parquet File.

Batch_id	Event_id	First_pulse_index	Last_pulse_index	Azimuth	Zenith
1	24	0	60	5.029555	2.087498
1	41	61	111	0.417742	1.549686
1	59	112	147	1.160466	2.401942
1	67	148	289	5.845952	0.759054
1	72	290	351	0.653719	0.939117

To get the variable-length pulse series for each event from the flat data structure, two metadata indices, `first_pulse_index` and `last_pulse_index`, are used. The data for a specific event is in the rows between these two indices.

4. METHODOLOGY

4.1 Choosing and Explaining the Model Architecture: Why Transformer?

The existing research suggests that the Transformer architecture performs better than traditional architectures, such as CNNs and RNNs, when processing high-dimensional neutrino data. This study uses a Transformer-based approach, considering its theoretical advantages and the success seen in the IceCube Kaggle competition. This section, unlike the previous ones that discussed Transformers in general, will explain why we chose this architecture. It will also describe the changes we made to fit the specific requirements of our research.

4.1.1 A sequential modeling and long-range dependencies

Consistent with the capability to capture global dependencies discussed in Section 2.5.4, our methodology leverages the Transformer’s self-attention mechanism to model the neutrino detection pulses as a sequence modeling problem.

4.1.2 Positional awareness

The standard (Vanilla) Transformer architecture does not have a sense of sequence order and processes all input elements at the same time (permutation-invariant). Nonetheless, encoding the sequence of observations is paramount in time series modeling [60]. To address this challenge, our study has utilized positional encoding strategies. To fix the standard Transformer’s problem, we created a PositionalEncoding class based on sine and cosine and added it to the input vectors (time, charge, coordinates). This way, the model has learned a lot about the data, including what it is and when it happened [60].

4.1.3 Efficiency and scalability of computation

The Transformer architecture’s ability to work with large datasets is one of its best features. Research shows that Transformer models work better when there is more data, as long as they follow known scaling laws [52].

Also, studies from the IceCube competition have shown that Transformer models use GPUs more efficiently than GNNs [52]. For example, studies have shown that a Transformer model with 7.4 million parameters needs the same amount of computing power as a GNN model (DynEdge) with 1.4 million parameters [52].

4.1.4 Success in experiments and place in the literature

The success of Transformer-based models in the “IceCube - Neutrinos in Deep Ice” competition, as discussed in Section 2.5, provides strong experimental support for their use in this research. As the literature review showed, the winning solutions were limited by the competition’s short time frame and hardware constraints, which prevented the development of more complex and larger models. Therefore, the full potential of Transformers in this area hasn’t been fully explored. Therefore, the full potential of Transformers in this field has not yet been fully explored. Consequently, this study uses the Transformer architecture not only to demonstrate that it can perform well, but also to investigate this unexplored potential. Our approach aims to overcome the limitations of previous competitive applications and to enhance the current capabilities of the architecture. [52].

4.2 Model Architecture and Experimental Design

This section describes the architecture, mathematical principles, and experimental setups of the DL model based on the Transformer Encoder architecture. This model was created to estimate the arrival directions, specifically the azimuth and zenith angles, of neutrino particles. Specifically, the progressive development of the model is detailed through seven distinct experiments (Exp1–Exp7), providing a comparative evaluation of different strategies regarding data handling, distributed training, and loss function formulations.

4.2.1 Data preparation

The raw dataset, as described before, has undergone several numerical changes within the NeutrinoDataset class. These changes were made to prepare the data for the training process of the DL model. The fundamental preprocessing steps applied to increase the model’s convergence speed and optimize computational load are as follows.

4.2.1.1 Feature normalization

In DL, using input data with different sizes can hinder the optimization process. To speed up how quickly the model converged, the input features were scaled. In all experimental setups, the main features were scaled using the same methods. In contrast, the final phase, Exp6 and Exp7, included an additional implicit normalization method for the new characteristics.

The sensor's spatial coordinates, represented by the variables x, y, z , have been reduced by a factor of 500. The IceCube detector's reach extends roughly 500 meters from its core. Dividing the physical coordinates by 500 changes their range to the unit interval, which is $[-1, 1]$. This method helps to reduce the instability in a neural network's weights, which can be caused by large values in the input data.

The temporal data underwent normalization, which was accomplished using the formula $(t - 10^4) / (3 \times 10^4)$. The initial time values are substantial, showing a concentration around a specific starting point. To bring the distribution to zero, we subtract 10^4 . After this, the time span of the events is standardized to the unit interval by dividing by 3×10^4 . This normalization method ensures that the temporal information has the same impact on the loss function as the spatial features.

The charge variable was changed using the formula $\log_{10} \text{charge} / 3$. The number of p.e., which indicates the amount of charge, follows a distribution with a noticeable tail. The data show a wide range of values, from 1 to more than 1000 p.e.. The logarithmic transformation, by compressing the dynamic range, allows the model to learn from signals of different strengths, including both weak and strong signals. Dividing by 3.0 normalizes the logarithmic values, where $\log_{10} 1000 = 3$, to a range from 0 to 1.

In Exp6 and Exp7, the model's spatial understanding improved by adding three new spherical coordinate features: r, θ and ϕ . In contrast to temporal and charge data, which required explicit scaling, these features used implicit normalization. The radial distance, represented by (r) , is calculated directly from the normalized Cartesian coordinates (x, y, z) . This method ensures that the values stay within a unit scale, where r ranges from 0 to approximately the $\sqrt{3}$, which means we don't need to divide them again. The angular parameters, represented by θ and ϕ , are expressed in radians. Because the radian scale restricts values to specific numerical ranges (specifically, θ

between 0 and π , and ϕ between $-\pi$ and π), it works well with the network's weight initialization. This implies that further expansion isn't necessary.

4.2.1.2 Sequence length and padding strategy

Because neutrino events vary in duration, as indicated by the number of sensor hits, transforming them into a compatible tensor format required different strategies across the experiments. This involved managing both the length of the sequences and the padding.

In the initial experiments (Exp1 – Exp3), no specific upper limit was set. The length of the sequence was determined by the longest event in each batch. To match this changing length, shorter events were filled with zeros. In contrast, a naive padding approach was used. This method treated the zero values as valid inputs, which introduced noise into the attention mechanism. As a result, this caused instability in the model's memory.

In the Exp4, to standardize the input for disk-based loading, a fixed sequence length of 4096 was used. Although this approach reduced memory fluctuations, the padding method remained naive. As a result, the model still used computational resources to process empty zeros.

In the Exp5, a Boolean Padding Mask was incorporated, all while keeping the sequence length at a constant 4096. This mask tells the Transformer's Multi-Head Attention layers to disregard padded positions, effectively giving them an attention score of negative infinity. This was a crucial move to get rid of the unwanted noise generated by zero-padding.

In the Exp6 and Exp7, a strategic trade-off was made to enable RAM-Caching, which is the Optimized In-Memory Strategy. Because loading the entire dataset, which has a length of 4096, would exceed the hardware's memory limits, the sequence length was reduced to 192. To offset this decrease:

- The feature set was expanded from six to nine, including spherical coordinates.
- In contrast to earlier experiments that used a simple average of the output, which diluted the signal with zeros, Exp6 and Exp7 used a Weighted Pooling method. As a result, the final event vector is created using only the valid sensor readings, completely excluding the padded areas from the final representation.

4.2.1.3 Data splitting

To assess the model's capacity to generalize and to mitigate overfitting, the dataset underwent partitioning into training, validation, and test sets. Overfitting occurs when a model learns the specific details and noise in the training data, which reduces its ability to make accurate predictions on new, unseen data. In this case, while the training loss decreases significantly, the validation loss begins to increase [61]. In the first five experiments, a data split of 70% Training, 15% Validation, and 15% Testing was applied. In Exp6 and Exp7, the training set ratio was increased to maximize the model's learning capacity. A distribution of 80% Training, 10% Validation, and 10% Testing was adopted to provide the model with more data for learning complex spatial patterns.

4.2.2 Feature engineering

In the first five experimental configurations (Exp1-Exp5), the model was provided with input features comprising time, charge, auxiliary data, and sensor positions represented by Cartesian coordinates (x, y, z) , [time, charge, auxiliary, x, y, z], with the objective of estimating the azimuth and zenith angles.

However, due to the limitation of Cartesian coordinates in directly representing directional orientation, a feature engineering approach was adopted in the Exp6 and Exp7. In this context, the geometric representation of the problem was optimized for the model by applying a spherical coordinate transformation, which directly yields radial distance and angular orientation. Consequently, in the current phase (Exp6), the input vector has been expanded to include spherical components (r, θ, ϕ) in addition to the standard Cartesian coordinates (x, y, z) , [time, charge, auxiliary, x, y, z, r, θ, ϕ].

After the feature expansion, the final input representation is organized as a high-dimensional tensor so that the Transformer architecture can process batches of data. For the inference process, the input is a tensor $X \in \mathbb{R}^{B \times N \times F}$, where B is the batch size, N is the fixed sequence length, and F is the input feature dimension. The Linear Embedding layer then takes this structured tensor and uses it to project the F -dimensional feature space into the model's 128-dimensional hidden state.

4.2.3 Model architecture

This study uses a model based on the Transformer Encoder architecture to analyze complex spatiotemporal relationships in neutrino events. The model, built using the PyTorch library and the NeutrinoTransformer Model class, is a version of the original architecture proposed by Vaswani et al. (2017), but adapted for neutrino physics as illustrated in Figure 4.1 [7].

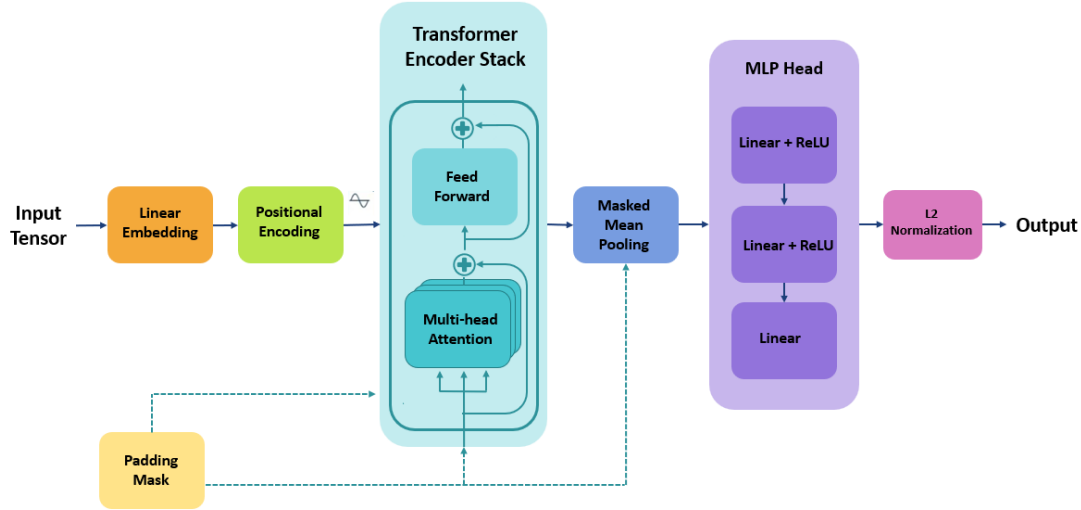


Figure 4.1 : Model Architecture for Exp6 and Exp7.

The Transformer architecture operates on high-dimensional vector spaces (d_{model}). However, the preprocessed neutrino event data consists of low-dimensional feature vectors (9 features in Exp6 and Exp7). Therefore, before the positional encoding step, a learnable Linear Projection layer is employed to map the input features to the model's hidden dimension space ($d_{\text{model}}=128$). This transformation allows the model to extract initial relationships between the heterogeneous input variables (spatial, temporal, and charge data) and prepares the sequence for the self-attention mechanism.

This projection is mathematically represented as Equation 4.1:

$$X_{\text{embed}} = X_{\text{input}} W_p + b_p \quad (4.1)$$

W_p is the weight matrix and b_p is the bias vector.

4.2.3.1 Positional encoding

Since the Transformer architecture processes data in parallel (permutation-invariant), it does not possess inherent knowledge regarding the order of data. To introduce the

flow of signals over time to the model, Positional Encoding vectors based on sine and cosine functions have been added to the input vectors. The formulation used in our study is as shown in Equation 4.2 and 4.3:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (4.2)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right) \quad (4.3)$$

This operation enables the model to learn the sequence of occurrence of sensor pulses and their temporal distances from each other [7].

4.2.3.2 Multi-head self-attention

In Experiments 1 through 5, the model was set up with nhead=4. This meant the embedding dimension was divided into four subspaces, which allowed the model to learn different types of relationships.

In Exp6 and Exp7, the number of attention heads was increased to nhead=8. This change was made to help the model better understand complex and varied spatiotemporal relationships. As a result, this change allows the model to consider a wider range of signal characteristics at the same time. Increasing the number of heads from 4 to 8 in Exp6 serves a specific physical purpose. In the context of neutrino detection, distinct heads can theoretically specialize in different aspects of the event topology: while some heads may attend to the temporal causality of the Cherenkov cone (tracking the wavefront), others may focus on spatial clustering to identify high-charge regions indicative of the interaction vertex. This parallel processing capability is crucial for distinguishing the directional signal from the sparse noise inherent in the IceCube detector.

The attention mechanism is mathematically expressed by the Scaled Dot-Product Attention formula (Equation 4.4):

$$Attention(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (4.4)$$

Here, the Q (Query), K (Key), and V (Value) matrices are obtained by multiplying input vectors with learnable weights. Thanks to this formula, the model can calculate

which sensor pulses in a neutrino event are more related to each other, regardless of the distance between them [7].

4.2.3.3 Position-wise feed-forward networks

Data emerging from the attention layer is passed to a fully connected Feed-Forward Network (FFN) applied independently and identically for each position. The architecture consists of two linear transformations, with a ReLU (Rectified Linear Unit) activation function between them. This function is known for helping to stabilize the training of deep networks. Equation 4.5 provides the mathematical representation of this operation.

$$FFN(x) = \max(0, xW_1 + b_1) W_2 + b_2 \quad (4.5)$$

The ReLU function, defined as $f(x) = \max(0, x)$, sets negative inputs to zero. This property creates sparsity in the network, which then reduces the computational cost. Unlike traditional functions like Sigmoid or Tanh, the ReLU function's constant derivative (1) in the positive range is crucial for preventing the vanishing gradient problem, which is often seen in DL models. In our model, this layer is used to learn more complex and non-linear representations of features [62].

In all experimental setups, this FNN structure is implemented via the `nn.TransformerEncoderLayer` module of the PyTorch library. The standard architecture utilizes a hidden dimension size of 2048 and applies the ReLU activation function by default, consistent with Equation 4.5.

4.2.3.4 Layer normalization and residual connections

To stabilize the training of deep networks, Residual Connections and Layer Normalization are used at the output of both the Attention and FFN layers. The `nn.TransformerEncoderLayer` structure in our code automatically performs this operation, as shown in Equation 4.6.

$$Output = LayerNorm(x + Sublayer(x)) \quad (4.6)$$

This structure ensures that gradients propagate back through the network without vanishing and allows for the healthy training of models at the varying depths investigated in this study.

Specifically, for Exp1, Exp2, Exp3, Exp4, and Exp5, the model depth was set to $\text{num_layers}=2$, while $\text{num_layers}=6$ was used for Exp6 and Exp7.

4.2.3.5 Output layer

The final prediction is made by taking the average (Global Average Pooling) of the 128-dimensional feature vectors ($d_{\text{model}} = 128$) that come from the Encoder block. Although this pooling mechanism remained consistent, the subsequent architecture of the output layer includes experiment-based differences to better handle the physical constraints of neutrino direction estimation.

In the Exp1 and Exp2, a simple linear transformation was used without activation functions. However, this approach failed to bound the predictions, occasionally producing non-physical angles. In the Exp3-Exp5 to address the bounding issue, the output layer was modified to use the Sigmoid activation function to normalize the results. The sigmoid function, which has a “S” shape, is a differentiable activation function. It takes any real number, from negative to positive infinity, and changes it to a value between 0 and 1. This method is often used in binary classification problems, or when the goal is to produce a probability or a ratio. In this study, it was used as a scaling factor to normalize outputs, bringing them into a specific angular range, either from 0 to π or from 0 to 2π [63].

The Sigmoid activation function can be shown as specified in Equation 4.7:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (4.7)$$

The results are normalized as follows:

- Azimuth Prediction: $\text{Sigmoid}(x_1) \times 2\pi$
- Zenith Prediction: $\text{Sigmoid}(x_2) \times \pi$

This operation guarantees that the angles produced by the model remain within physically meaningful ranges.

In the Exp6 and Exp7, although the Sigmoid approach successfully bounded the angles, it did not fully resolve the periodicity issue (e.g., the proximity of 0 and 2π). Therefore, in the Exp6 and Exp7, the output layer was upgraded to a Multi-Layer Perceptron (MLP) predicting 3D Cartesian coordinates x, y, z . Instead of Sigmoid

scaling, an L_2 normalization was applied to project the output onto the unit sphere, providing the most geometric robust representation.

4.3 Training Infrastructure and Configuration

The computational experiments in this study were conducted using two different high-performance computing environments. The first experiments, labeled Exp1 through Exp5, were performed using the computing resources at the Istanbul Technical University National Center for High Performance Computing (UHeM). Exp6 and Exp7 were conducted on the Kaggle cloud platform. This change allowed for the use of interactive GPU accelerators, like the NVIDIA P100 or T4. It also enabled the implementation of the improved ‘In-Memory’ data loading method within a containerized environment.

4.3.1 Distributed computing

Because the dataset was so large, at 117 GB, and because Transformer models are computationally intensive, using just one processor wasn’t enough for efficient training. Therefore, the training strategy evolved through different distributed computing protocols used in the experiments.

In Exp 1, 2, 6 and 7, the native PyTorch module, `torch.utils.data.distributed`, was used for DDP. This method allows the model to be replicated across multiple GPUs, and it synchronizes gradients during the backward pass. This approach offers strong stability when training on a large scale. In Exp 3, 4, and 5, the `mpi4py` library was used to manage communication between processes.

4.3.2 Optimization

Adaptive Moment Estimation (Adam) optimization algorithm was used to update the model parameters. Adam is a variation of the stochastic gradient descent (SGD) algorithm. The method calculates separate and adaptive learning rates for each parameter. This method uses both the average (first moment) and the spread (second moment) of the gradients. The chosen method was preferred because it converged faster than the SGD algorithm. Moreover, this method reduced the chance of getting stuck in local minima. This is especially important when working with sparse and

noisy data, like the information collected by IceCube. The update rule for the parameter vector, denoted as θ , is given by the Equation 4.8 [64].

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{\vartheta}_t} + \epsilon} \hat{m}_t \quad (4.8)$$

In this context, the symbol θ represents the model parameters, η is the learning rate, $\hat{m}_t$ is the corrected first moment, and $\hat{\vartheta}_t$ is the corrected second moment [64].

In all experiments, the Adam algorithm was used. However, the hyperparameters were refined in Exp6 and Exp7 to ensure stable training. In Exp1 through Exp5, the initial experiments used a constant learning rate of 1e-4 throughout the training. In the sixth experiment, two key changes were made to speed up the model's convergence and reduce instability during the early training stages.

To reduce sudden and large changes in model weights at the beginning, a linear warmup scheduler was used. This method gradually increases the learning rate, starting at 0 and reaching a maximum of 2e-4 over the first 1000 steps. The use of a linear warmup scheduler in Exp6 is necessitated by the specific optimization landscape of Transformer models. Without a warmup phase, the adaptive learning rates in the Adam optimizer can show high variance in the initial updates. This is because of the lack of reliable gradient statistics. As a result, the model often converges to a suboptimal local minimum early in the training process.

Gradually ramping up the learning rate enables the model to converge to a stable region within the parameter space prior to commencing comprehensive optimization; this approach is especially important when training on irregular and sparse geometries, such as those encountered in neutrino events. To mitigate the frequent occurrence of exploding gradients in deep Transformer models, gradient clipping was implemented. This method constrained the magnitude of the gradients, establishing a maximum norm of 1.0 to promote stable training.

Exp7 represents a fine-tuning stage, succeeding the preliminary training conducted in Exp6. To promote stable training dynamics and facilitate precise convergence to a local minimum, the learning rate was decreased to 5e-5, a quarter of its initial setting. In contrast to the preceding phase, a warmup scheduler was not implemented; training

commenced directly at this diminished rate. This strategy sought to optimize the pre-trained weights without altering the model's established feature representations.

4.3.3 Loss functions

This thesis study is a regression problem where continuous angle values are estimated. Regression is a statistical method that models the relationship between a dependent variable (target) and one or more independent variables (features). Unlike classification, regression models predict continuous numerical values instead of discrete classes [65].

In the first two experiments, referred to as Exp1 and Exp2, the Standard Mean Squared Error (MSE) was used as the main loss function. MSE, a widely used loss function in regression, measures how well a model works by calculating the average of the squared differences between the predicted angular values and the actual values.

As shown in Equation 4.9, this function squares the error terms, which means it penalizes larger errors more than smaller ones. This traditional method, while useful for general regression, treats azimuth and zenith angles as if they were linear variables.

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (4.9)$$

Here, n represents the number of samples, y_i the true value, and $\hat{y}_i$ the predicted value. It is the average of the squares of the differences between the predicted value $\hat{y}$ and the true value y [65].

In the Exp3, Exp4 and Exp5, to resolve the boundary discontinuity problem of angular values (cyclicity), the optimization objective was shifted from angular space to vector space. The DirectionMSELoss function was implemented. First, the predicted angles are converted into Cartesian unit vectors using the transformation in Equation 4.10:

$$v = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \quad (4.10)$$

In the Exp6 and Exp7 the objective was refined to focus purely on the alignment of direction rather than point-wise distance. A Cosine Similarity Loss was adopted. Mathematically, this function maximizes the cosine of the angle between the predicted vector ϑ_{pred} and the true vector ϑ_{true} .

The loss value approaches 0 as the vectors become perfectly aligned. The mathematical formulation is given in Equation 4.11.

$$L_{cos} = 1 - \frac{\vartheta_{pred} \cdot \vartheta_{true}}{||\vartheta_{pred}|| \cdot ||\vartheta_{true}||} \quad (4.11)$$

4.4 Performance Metrics

In the first five experiments, the metrics used to evaluate the model's success are mathematically identical to the loss functions detailed in Section 4.3.3.

- The results reported for Exp1 and Exp2 correspond to the Standard Angular MSE.
- The results reported for Exp3, Exp4, and Exp5 correspond to the Vector MSE (DirectionMSELoss), which measures the distance on the unit sphere. Standard MSE cannot manage the cyclicity in angular data (e.g., the difference between 359° and 1° should be 2° , but mathematically it comes out as 358). To overcome this problem, DirectionMSELoss was developed.
- In the Exp6 and Exp7, although the model was trained on cosine similarity, the performance was evaluated in terms of physical angular deviation. Distinct from the training loss, this metric expresses the angle between the predicted vector and the true vector in degrees ($^\circ$). This conversion was performed using the following equation 4.12:

$$\theta_{error} = \arccos(\vartheta_{pred} \cdot \vartheta_{true}) \frac{180}{\pi} \quad (4.12)$$

Here, the term $\vartheta_{pred} \cdot \vartheta_{true}$ represents the dot product between two unit vectors. The $\arccos$ function converts this similarity into an angle in radians, and the factor $\frac{180}{\pi}$ converts it to degrees. This metric presents the deviation in the neutrino arrival direction estimation in the most interpretable format.

Throughout all experimental phases, the Coefficient of Determination (R^2) was consistently calculated to monitor the general statistical fit of the models. R^2 is a statistical measure indicating how much of the variance (variability) in the data a regression model can explain. It generally takes values between -1 and 1, approaching 1 indicates that the model's predictions fit the real data very well. Its formulation is given in Equation 4.13 and Equation 4.14.

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \bar{y})^2} \quad (4.13)$$

$$R^2 = 1 - \frac{SS_{residual}}{SS_{total}} \quad (4.14)$$

Here, $\bar{y}$ is the mean of the true values. $SS_{residual}$ is the sum of squared residuals, and SS_{total} is the total sum of squares [66].



5. RESULTS AND DISCUSSION

In this section, the performance of the deep learning models developed for estimating neutrino arrival directions is comprehensively analyzed. The process, extending from the baseline models (Exp1-Exp2) to the intermediate experiments where optimization challenges were encountered (Exp3-Exp5), and finally to the mature stages of the proposed method (Exp6-Exp7), is evaluated using quantitative data.

5.1 Comparison of Experimental Results

The cumulative impact of different hardware infrastructures, loss functions, and training strategies on model success is summarized in Table 5.1.

Table 5.1 : Experimental Results and Configurations.

Experiment Name	Training Duration(min.)	Average Training Loss	Batc Size	Epoch Number	Sub-Sample Size	MSE	R^2	Mean Angular Error
Exp1	79.38	1.71 (MSELoss)	1	2	90000	1.70(MSE)	0.32	-
Exp2	742.91	1.64 (MSELoss)	1	2	900000	1.66(MSE)	0.34	-
Exp3	650.52	0.68(DirectionMSELoss)	1	25	1000	0.67(3DMSE)	-1	-
Exp4	804.85	0.65(DirectionMSELoss)	1	5	1000	0.64(3DMSE)	-0.95	-
Exp5	742.59	0.65(DirectionMSELoss)	1	5	1000	0.61(3DMSE)	-0.87	-
Exp6	538.8	0.704(Cosine Similarity Loss)	64	15	1.000.000	0.45	-0.35	64.80°
Exp7	883.75	0.675(Cosine Similarity Loss)	64	10	1.000.000	0.44	-0.33	63.80°

Examination of Table 5.1 reveals that the instabilities experienced in the CPU-based experiments (Exp1-5) were overcome with Exp6, and the continued training (fine-tuning) performed in Exp7 moved the model to the optimal point, reducing the error to the 63.80° level.

5.2 Performance Analysis

5.2.1 Performance of the baseline model (Exp1 - Exp2)

The results of Exp1 and Exp2 showed that, when the problem was approached as a direct angular regression, the model demonstrated some ability to learn, but its capacity was limited.

- When the Angular MSE value of 1.708 is converted to radians, it corresponds to an average error of about 74.5 degrees.
- The R^2 score was 0.325. A positive R^2 value indicates that the model can successfully show a linear relationship between the input data and the resulting angles. However, the low value of this score (0.33) suggests that the model can only explain 33% of the data's variability. This means it doesn't account for the remaining 67%, particularly the inaccuracies in the boundary areas.

5.2.2 Constraints in intermediate experiments (Exp3 - Exp5)

Although the vector-based approach (Exp3-5) theoretically solved the boundary problem, its practical use was limited by hardware constraints.

The average vector MSE showed a decrease, going from 0.67 in Experiment 3 to 0.61 in Experiment 5. The model's understanding of geometric orientation seems to have improved compared to the initial experiment (Exp1 and Exp2). However, the results were still not as good as those achieved with the final method.

Despite the apparent reduction in average loss, the training logs revealed significant instability. In Exp3, the batch loss showed considerable fluctuation, with values ranging from 0.006 to 1.23. The considerable variation suggests that the model struggles to find a global minimum when using the CPU infrastructure.

In this phase, the R^2 values were -1 for Exp3, -0.95 for Exp4, and -0.87 for Exp5. While R^2 isn't usually the best way to measure spherical regression, values this low, especially those near or below -1.0, indicate a major problem with the training process. Mathematically, this suggests that the model's predictions were less accurate than a horizontal line representing the average value. The results confirm that the CPU-based model couldn't reach a stable state, showing significant oscillations, as indicated by the log loss value of 1.23.

5.2.3 Superiority of the proposed method (Exp6 - Exp7)

The switch to a GPU-based architecture, along with using Cosine Similarity loss, led to a clear and significant improvement compared to the initial models. The training logs from Exp6 showed consistent stability. As illustrated in Figure 5.1, The training loss decreased from 0.79 to 0.66, eliminating the fluctuations seen in earlier phases.

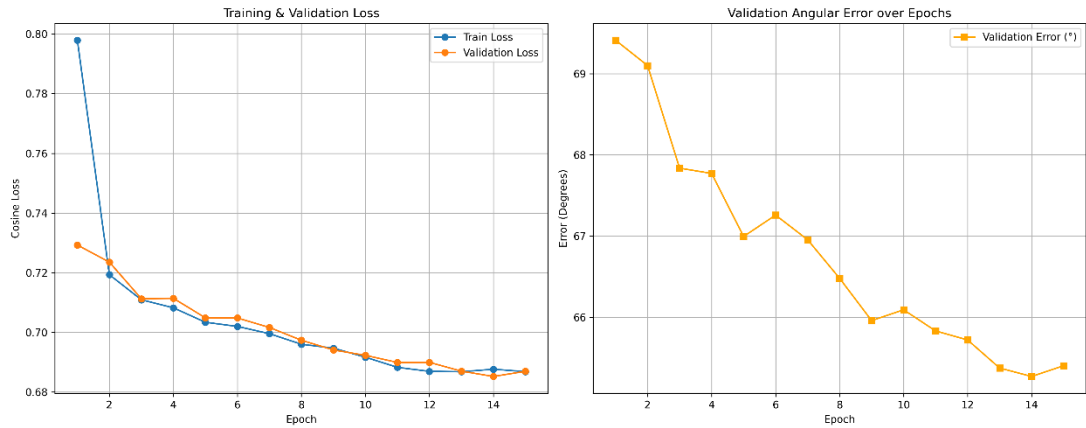


Figure 5.1 : Training and Validation Loss Curves for Exp6.

The graph highlights the stability of the training process and the sharp decrease in loss values.

Following the weight optimization in Exp6, the Exp7 phase focused on further refining the model through fine-tuning. As shown in Figure 5.2, this phase resulted in a slow but steady decrease in the validation error, which went from 64.68° to 64.44° . This trend confirms that the model continued to learn and generalize, even after the initial convergence.

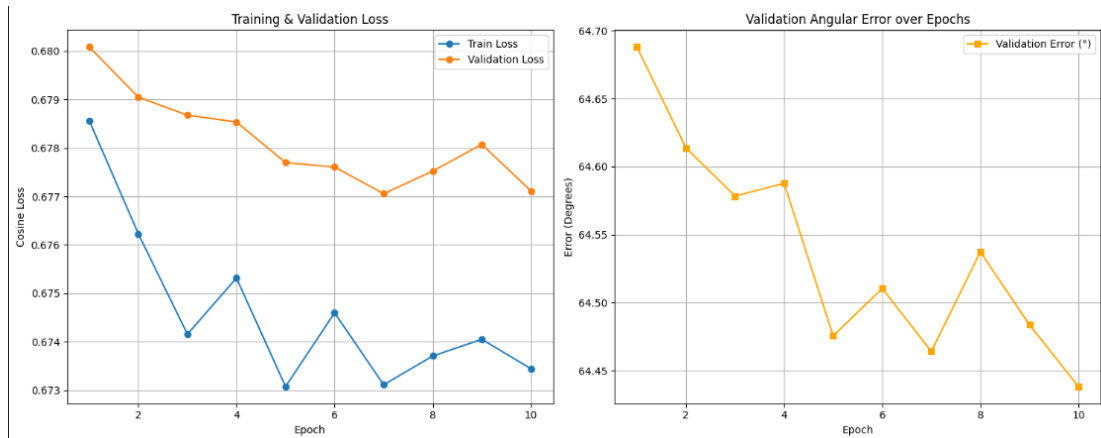


Figure 5.2 : Learning Curves During the Fine-Tuning Phase (Exp7).

The graph depicts the gradual minimization of the validation angular error over additional epochs.

The final adjustments yielded a mean angular error of 63.80° on the test set, a notable enhancement. This represented an approximate 11 degree increase in accuracy relative to the baseline Exp1, which exhibited an error of approximately 74.5° . Regarding statistical analysis, the R2 score demonstrated a slight improvement, shifting from -0.355 in Exp6 to -0.336 in Exp7. Although the score remained negative, a result of the problem's geometric nature, its increasing value (approaching zero) reflects the decreasing angular error. This finding supports the effectiveness of the fine-tuning procedure employed in Exp7, which improved the model's orientation capabilities beyond the initial convergence observed in Exp6.

5.3 Analysis of the Best and Worst Possible Outcomes

Relying solely on average error to evaluate a model's performance can be misleading. Examining the outliers in the Exp7 results offers a deeper understanding of the model's geometric comprehension. In the most successful prediction of the model, the error margin was measured at 0.0751° . This prediction is almost perfect, showing that the model can accurately determine the geometry and orientation of the detector in events with enough signal strength. The largest error margin observed was 179.22° . As illustrated in Figure 5.3, this error, which is very close to 180 degrees, indicates the "Head-Tail Ambiguity" problem. The model correctly identified the axis of the neutrino's path, but it predicted the direction of travel incorrectly. This is a common topological challenge in high-energy physics experiments that use tracks.

It suggests that the model's errors aren't random, but instead, it's learned the fundamental physics involved.

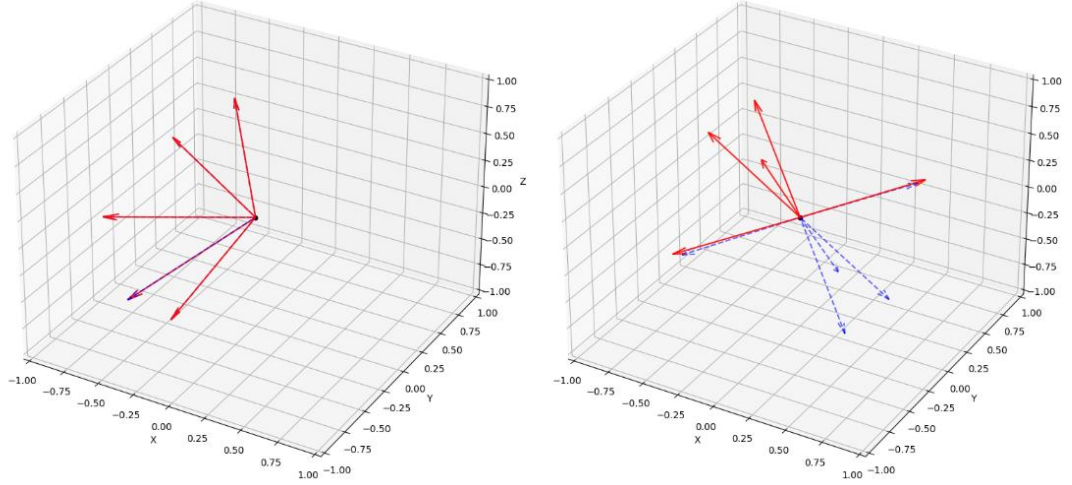


Figure 5.3 : Visualization of the Best and Worst Prediction Scenarios from Exp7.

In the Figure 5.1, the blue markers and lines show the actual direction of the neutrino, while the red markers and lines represent the model's predictions. In the best-case scenario, an angular error of 0.0751° indicates nearly perfect alignment. (Right) In the worst-case scenario, an angular error of 179.22° illustrates the “Head-Tail Ambiguity”. This occurs when the model correctly identifies the track axis but predicts the opposite direction.

5.4 Discussion on Metric Inconsistency (R^2 Analysis)

The negative R^2 values, when analyzed, reveal that they stem from two different fundamental causes, depending on the experimental stage. The results from the middle experiments (Exp3-Exp5), which dropped to -1, indicate significant training problems. This suggests the model couldn't learn even the basic spatial layout, likely because of hardware limitations and a lack of optimization. In contrast, the improvement to -0.336 in the final phase (Exp6-Exp7), where the proposed method was used, shows that the model has stabilized. However, the continued negative values are due to a problem with the metric, not a failure of the model itself. Although the model showed physical accuracy, with a 63.80° error, the R^2 score was still negative. This was due to geometric limits on the unit sphere and the mathematical rule that any angular error greater than 60° results in a negative value.

This finding confirms that the Mean Angular Error metric, which directly measures physical alignment, is a more accurate way to assess performance in spherical orientation problems than the variance-based R^2 score.

5.5 Visual Analysis of Direction Prediction Vectors

The complex structure of neutrino events and the noise ratio within the dataset directly impact the model's direction estimation performance. The following visual analyses document the evolutionary process, spanning from early stages where the model struggled to extract features from raw data, to the final stage where it achieved generalization capabilities through the integration of physics-based feature engineering.

The 3D comparison of True and Predicted Neutrino Direction Vectors for each experiment is presented sequentially below. The blue arrows indicate the true directions, while the red arrows represent the predicted directions. The experimental process chronologically reflects the difficulties the model faced in learning from raw data and the solution strategies applied.

As shown in Figures 5.4, 5.5, 5.6, and 5.7, during the initial stages, the model could only establish a weak correlation between the input data and output vectors; systematic directional deviations and axis alignment problems were observed in the predictions (Red).

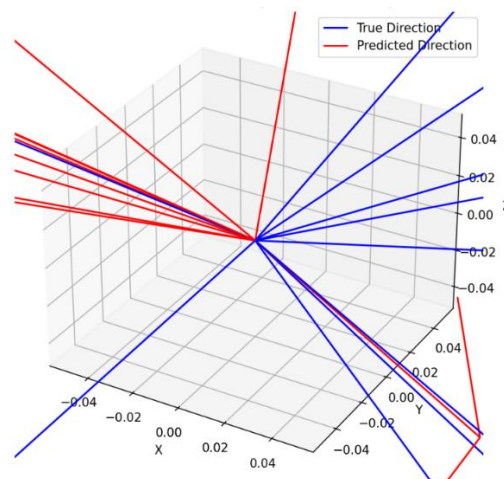


Figure 5.4 : 3D Comparison of True and Predicted Direction Vectors for Exp1.

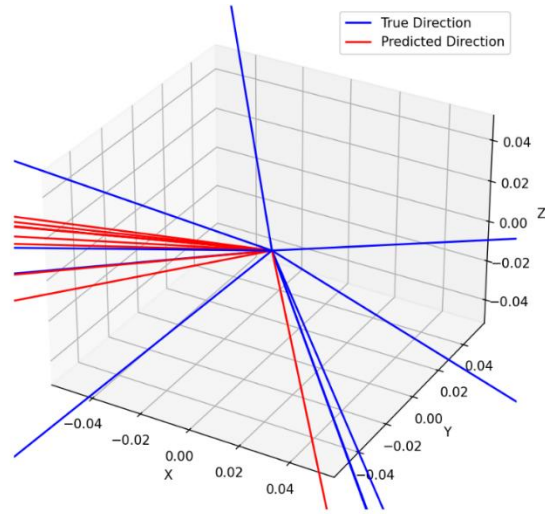


Figure 5.5 : 3D Comparison of True and Predicted Direction Vectors for Exp2.

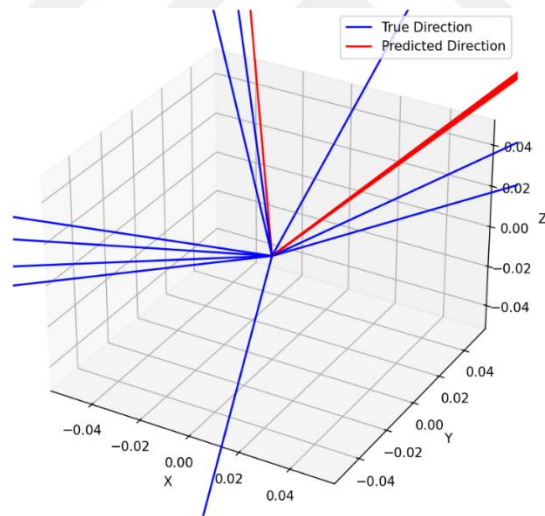


Figure 5.6 : 3D Comparison of True and Predicted Direction Vectors for Exp3.

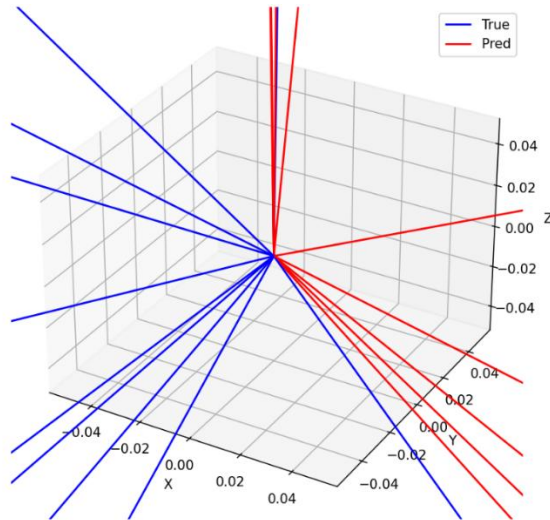


Figure 5.7 : 3D Comparison of True and Predicted Direction Vectors for Exp4.

The visualization for Exp5 (Figure 5.8) illustrates “Mode Collapse,” which was the most critical bottleneck in the learning process. Unable to distinguish between input data, the model became trapped in a local minimum and began predicting the mean direction of the training set (as a single cluster) for all events.

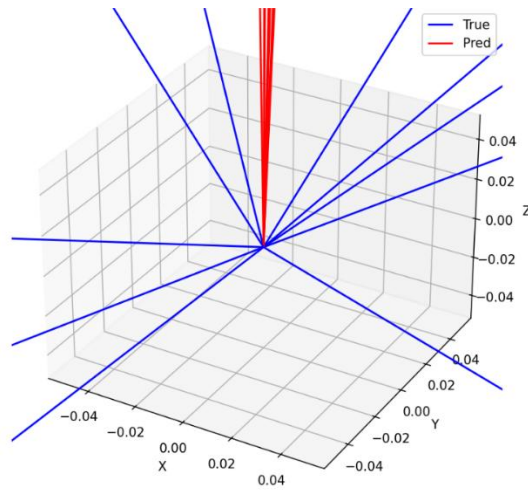


Figure 5.8 : 3D Comparison of True and Predicted Direction Vectors for Exp5.

In Exp6 (Figure 5.9), this problem was overcome by applying physics-based feature engineering methods instead of using raw data.

The fact that the predictions (Red) achieved a spherical distribution indicates that the model can now individually distinguish each neutrino event (Blue) and has gained spatial generalization capabilities.

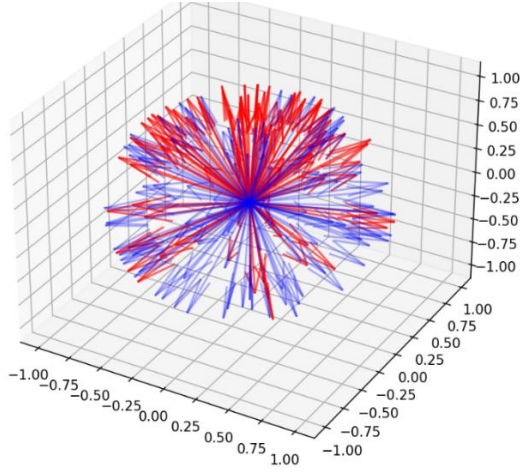


Figure 5.9 : 3D Comparison of True and Predicted Direction Vectors for Exp6.

Finally, in Exp7 (Figure 5.10), as a result of fine-tuning, the model has become more stable, and the angular error has been reduced while preserving variance. This progress demonstrates that data representation plays as critical a role as model architecture in neutrino direction estimation.

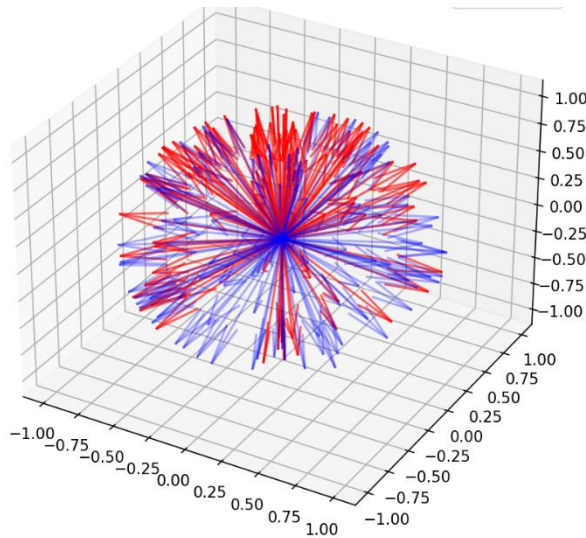


Figure 5.10 : 3D Comparison of True and Predicted Direction Vectors for Exp7.



6. FUTURE WORKS

The Transformer-based architecture developed in this work has demonstrated promising results in modeling the spatiotemporal dynamics of neutrino events. Nevertheless, considering the IceCube detector's complex, sparse, and non-uniform 3D geometry, integrating geometric deep learning into the current architecture presents a vital direction for subsequent research. Experimental findings, particularly from Exp7, indicate that although Transformers are proficient in sequence modeling, inputs based solely on coordinates might fail to capture local geometric details. Consequently, future studies should integrate GNN techniques, such as EdgeConv, to serve as feature extractors for local neighborhood geometries. These hybrid models would combine the Transformer's ability to capture long-range global dependencies with the GNN's capacity to analyze topological patterns, especially in contexts characterized by low-energy and sparse data.

In addition to architectural improvements, a central research objective is to shift the model's output from fixed point estimations to a probabilistic structure. Within high-energy physics, the assessment of a prediction's reliability is of equal importance to the prediction itself. Consequently, the incorporation of loss functions derived from directional statistical distributions, such as the Von Mises-Fisher distribution, would enable the model to concurrently learn the direction vector and its corresponding uncertainty. Moreover, the model's functionality should be enhanced through Multi-Task Learning methodologies. Given that neutrino events display interconnected physical properties including energy, flavor, and direction utilizing a unified backbone to predict these attributes concurrently would promote a more thorough comprehension of event physics.

Employing the self-attention mechanism to consolidate total charge deposition renders the architecture particularly well-suited for energy regression tasks. Precise energy resolution is essential for differentiating astrophysical neutrinos from atmospheric noise. To enhance data quality, integrating unsupervised Denoising Autoencoder architectures before the input layer offers a practical method for reducing background

noise, including cosmic rays, in events characterized by a low signal-to-noise ratio. Ultimately, to promote the practical application of these models in experimental physics, the black box characteristic of deep neural networks necessitates consideration. Future research should concentrate on visualizing self-attention maps to ascertain which pulses, such as high-charge or early-time pulses, the network identifies as most influential.



REFERENCES

- [1] **Karagiorgi, G., Kasieczka, G., Kravitz, S., Nachman, B., & Shih, D.** (2022). Machine learning in the search for new fundamental physics. *Nature Reviews Physics*, **4**, 399-412. <https://doi.org/10.1038/s42254-022-00455-1>
- [2] **Psihas, F., Groh, M., Tunnell, C., & Warburton, K.** (2020). A Review on Machine Learning for Neutrino Experiments. *International Journal of Modern Physics A*, **35**(33), 2043005. <https://doi.org/10.1142/S0217751X20430058>.
- [3] **Poonthottathila, N.** (2024). Machine learning in experimental neutrino physics. *Eur. Phys. J. Spec. Top.*, **233**, 2687-2698. <https://doi.org/10.1140/epjs/s11734-024-01280-6>
- [4] **Radovic, A., Williams, M., Rousseau, D., Kagan, M., Bonacorsi, D., Himmel, A., Aurisano, A., Terao, K., & Wongjirad, T.** (2018). Machine learning at the energy and intensity frontiers of particle physics. *Nature*, **560**, 41-48. <https://doi.org/10.1038/s41586-018-0361-2>
- [5] **Baldi, P., Sadowski, P., & Whiteson, D.** (2014). Searching for exotic particles in high-energy physics with deep learning. *Nature Communications*, **5**, 4308. <https://doi.org/10.1038/ncomms5308>
- [6] **Shlomi, J., Battaglia, P., & Vlimant, J.-R.** (2021). Graph neural networks in particle physics. *Mach. Learn.: Sci. Technol.*, **2**(021001). <https://doi.org/10.1088/2632-2153/abbf9a>
- [7] **Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I.** (2017). Attention Is All You Need. arXiv:1706.03762. <https://arxiv.org/abs/1706.03762>.
- [8] **Url-1** <<https://www.kaggle.com/competitions/icecube-neutrinos-in-deep-ice/data>>, date retrieved 26.11.2025.
- [9] **Hellwig, M., Herquet, P., Hill, G. C., Hubert, D., Hughey, B., Hulth, P. O., Hultqvist, K., Hundertmark, S., Jacobsen, J., Karle, A., . . . Przybylski, G. T.** (2004). Muon Track Reconstruction and Data Selection Techniques in AMANDA. *Nuclear Instruments and Methods in Physics Research Section*, **A524**, 169-194 <https://doi.org/10.1016/j.nima.2004.01.065>.

- [10] Aartsen, M. G., Abbasi, R., Abdou, M., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Altmann, D., Andeen, K., Anderson, T., . . . Zolla, M. (2014). Improvement in Fast Particle Track Reconstruction with Robust Statistics. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, **736**, 143-149. <https://doi.org/10.1016/j.nima.2013.10.074>
- [11] Hellauer, R., Hickford, S., Hignight, J., Hill, G. C., Hoffman, K. D., Hoffmann, R., Hokanson-Fasig, B., Hoshina, K., Huang, F., Huber, M., . . . Köpke, L. (2018). Measurement of Atmospheric Neutrino Oscillations at 6-56 GeV with IceCube. *Physical Review Letters*, **120**(7), 071801. <https://doi.org/10.1103/PhysRevLett.120.071801>.
- [12] Coniglione, R., Costantini, H., Cottini, N., Coyle, P., Curtil, C., Decowski, M. P., Dekeyser, I., Deschamps, A., Distefano, C., Donzaud, C., . . . de Wolf, E. (2011). A Fast Algorithm for Muon Track Reconstruction and its Application to the ANTARES Neutrino Telescope. *Astroparticle Physics*, **34**(9), 652-662. <https://doi.org/10.1016/j.astropartphys.2011.01.003>.
- [13] Acciarri, R., Adams, C., An, R., Anthony, J., Asaadi, J., Auger, M., Bagby, L., Balasubramanian, S., Baller, B., Barnes, C., Barr, G., Bass, M., Bay, M., Bishai, M., Blake, A., Bolton, T., Camilleri, L., Caratelli, D., Carls, B., . . . Zhang, C. (2018). The Pandora multi-algorithm approach to automated pattern recognition of cosmic-ray muon and neutrino events in the MicroBooNE detector. *The European Physical Journal C*, **78**(1), 82. <https://doi.org/10.1140/epjc/s10052-017-5481-6>
- [14] Aguilar, J. A., Allison, P., Beatty, J. J., Bernhoff, H., Besson, D., Bingsfors, N., Botner, O., Bouma, S., Buitink, S., Carter, K., Cataldo, M., Clark, B. A., Curtis-Ginsberg, Z., Connolly, A., Dasgupta, P., de Kockere, S., de Vries, K. D., Deaconu, C., DuVernois, M. A., . . . Zink, A. (2022). Reconstructing the neutrino energy for in-ice radio detectors. *The European Physical Journal C*, **82**(2), 147. <https://doi.org/10.1140/epjc/s10052-022-10034-4>
- [15] Kovalev, Y. Y., Plavin, A. V., & Troitsky, S. V. (2022). Galactic contribution to the high-energy neutrino flux found in track-like IceCube events. *The Astrophysical Journal Letters*, **940**(2), L41. <https://doi.org/10.3847/2041-8213/aca1ae>
- [16] Aartsen, M. G., Abbasi, R., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Altmann, D., Argüelles, C., Auffenberg, J., Bai, X., Baker, M., Barwick, S. W., Baum, V., Bay, R., Beatty, J. J., Becker Tjus, J., Becker, K.-H., BenZvi, S., Berghaus, P., . . . Zoll, M. (2014). Energy reconstruction methods in the IceCube Neutrino Telescope. *Journal of Instrumentation*, **9**(3), P03009. <https://doi.org/10.1088/1748-0221/9/03/P03009>.

- [17] Aurisano, A., Radovic, A., Rocco, D., Himmel, A., Messier, M. D., Niner, E., Pawloski, G., Psihas, F., Sousa, A., & Vahle, P. (2016). A convolutional neural network neutrino event classifier. *Journal of Instrumentation*, **11**(9), P09001. <https://doi.org/10.1088/1748-0221/11/09/P09001>.
- [18] Abbasi, R., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Ahrens, M., Alispach, C., Alves Jr., A. A., Amin, N. M., An, R., . . . Koirala, R. (2021). A Convolutional Neural Network Based Cascade Reconstruction for the IceCube Neutrino Observatory. *Journal of Instrumentation*, **16**(07), P07041. <https://doi.org/10.1088/1748-0221/16/07/P07041>.
- [19] Novotný, A., & Franc, J. (2021). Application of Convolutional Neural Networks in Neutrino Physics. *Journal of Physics: Conference Series*, **1730**, 012116. <https://doi.org/10.1088/1742-6596/1730/1/012116>
- [20] Abi, B., Brunetti, G., Buchanan, N., Budd, H., Caiulo, D., Calafiura, P., Calcutt, J., Calin, M., Calvez, S., Calvo, E., . . . Zhang, C. (2020). Neutrino interaction classification with a convolutional neural network in the DUNE far detector. *Physical Review D*, **102**(092003). <https://doi.org/10.1103/PhysRevD.102.092003>.
- [21] Dominé, L., & Terao, K. (2020). Scalable Deep Convolutional Neural Networks for Sparse, Locally Dense Liquid Argon Time Projection Chamber Data. *Physical Review D*, **102**(012005). <https://doi.org/10.1103/PhysRevD.102.012005>.
- [22] Adams, C., Alrashed, M., An, R., Anthony, J., Asaadi, J., Ashkenazi, A., Auger, M., Balasubramanian, S., Baller, B., Barnes, C., . . . Zhang, C. (2019). Deep neural network for pixel-level electromagnetic particle identification in the microboone liquid argon time projection chamber. *Physical Review D*, **99**(092001). <https://link.aps.org/doi/10.1103/PhysRevD.99.092001>.
- [23] Abratenko, P., Convery, M., Cooper-Troendle, L., Crespo-Anadón, J. I., Del Tutto, M., Devitt, D., Diurba, R., Domine, L., Dorrill, R., Duffy, K., Dytman, S., Eberly, B., Ereditato, A., Escudero Sanchez, L., Evans, J. J., Fiorentini Aguirre, G. A., Fitzpatrick, R. S., Fleming, B. T., Foppiani, N., Franco, D., . . . Wospakrik, M. (2021). Convolutional neural network for multiple particle identification in the MicroBooNE liquid argon time projection chamber. *Phys. Rev. D*, **103**(092003). <https://doi.org/10.1103/PhysRevD.103.092003>.
- [24] Psihas, F., Groh, M., Tunnell, C., & Warburton, K. (2019). Context-enriched identification of particles with a convolutional network for neutrino events. *Physical Review D*, **100**(7), 073005. <https://doi.org/10.1103/PhysRevD.100.073005>.

- [25] **IceCube Collaboration.** (2021). Direction reconstruction using a CNN for GeV-scale neutrinos in IceCube. *Journal of Instrumentation*, **16**(11), C11001. <https://doi.org/10.1088/1748-0221/16/11/C11001>.
- [26] **Acciarri, R., Adams, C., An, R., Asaadi, J., Auger, M., Bagby, L., Baller, B., Barr, G., Bass, M., Bay, F., Bishai, M., Blake, A., Bolton, T., Bugel, L., Camilleri, L., Caratelli, D., Carls, B., Castillo Fernandez, R., Cavanna, F., . . . Zhang, C.** (2017). Convolutional neural networks applied to neutrino events in a liquid argon time projection chamber. *Journal of Instrumentation*, **12**(3), P03011. <https://doi.org/10.1088/1748-0221/12/03/P03011>.
- [27] **Stjärnholm, S., Ericsson, O., & Glaser, C.** (2021). Neutrino direction and flavor reconstruction from radio detector data using deep convolutional neural networks. *PoS (ICRC2021)*, 1055. <https://doi.org/10.22323/1.395.1055>
- [28] **Glaser, C., McAleer, S., Stjärnholm, S., Baldi, P., & Barwick, S. W.** (2023). Deep-learning-based reconstruction of the neutrino direction and energy for in-ice radio detectors. *Astroparticle Physics*, **145**, 102781. <https://doi.org/10.1016/j.astropartphys.2022.102781>.
- [29] **Abbasi, R., Ackermann, M., Adams, J., Aggarwal, N., Aguilar, J. A., Ahlers, M., Ahrens, M., Alameddine, J. M., Alves Jr., A. A., Amin, N. M., . . . Zhelnin, P.** (2022). Graph Neural Networks for low-energy event classification & reconstruction in IceCube. *Journal of Instrumentation*, **17**(11), P11003. <https://doi.org/10.1088/1748-0221/17/11/P11003>.
- [30] **Yu, F. J., Kamp, N., & Argüelles, C. A.** (2025). Enhancing events in neutrino telescopes through deep-learning-driven superresolution. *Physical Review D*, **111**(4), L041301. <https://doi.org/10.1103/PhysRevD.111.L041301>.
- [31] **Abbasi, R., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Ahrens, M., Alispach, C., Alves Jr., A. A., Amin, N. M., An, R., . . . Zhelnin, P.** (2023). Observation of high-energy neutrinos from the galactic plane. *Science*, **380**(6652), 1338-1343. <https://doi.org/10.1126/science.adc9818>.
- [32] **Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., . . . Pascanu, R.** (2018). Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261. <https://doi.org/10.48550/arXiv.1806.01261>.
- [33] **Aurisano, A., Hewes, V., Cerati, G., Kowalkowski, J., Lee, C. S., Liao, W., Grzenda, D., Gumpula, K., & Zhang, X.** (2024). Graph neural network for neutrino physics event reconstruction. *Physical Review D*, **110**(3), 032008. <https://doi.org/10.1103/PhysRevD.110.032008>.

- [34] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications. *AI Open*, *1*, 57-81. <https://doi.org/10.1016/j.aiopen.2021.01.001>.
- [35] Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. (2019). Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics*, *1*(1), Article 1. <https://doi.org/10.1145/3326362>.
- [36] Vijayakumar, K., Padmanaban, A., & Rajesh Kumar, D. (2024). Neutrino Direction Detection using GNN. In *Proceedings of 2024 International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS 2024)*. IEEE. <https://doi.org/10.1109/ICC-ROBINS60238.2024.10534013>
- [37] Choma, N., Monti, F., Gerhardt, L., Palczewski, T., Ronaghi, Z., Prabhat, P., Bhimji, W., Bronstein, M. M., Klein, S. R., & Bruna, J. (2018). Graph Neural Networks for IceCube Signal Classification. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)* (pp. 386-391). <https://doi.org/10.48550/arXiv.1809.06166>.
- [38] Søgaard, A., Ørsøe, R. F., Bozianu, L., Holm, M., Iversen, K. E., Guggenmos, T., Ha Minh, M., Eller, P., & Petersen, T. C. (2022). GraphNeT: Graph neural networks for neutrino telescope event reconstruction. arXiv:2210.12194. <https://doi.org/10.48550/arXiv.2210.12194>.
- [39] Ju, X., Farrell, S., Calafiura, P., Murnane, D., Prabhat, Gray, L., Klijsma, T., Pedro, K., Cerati, G., Kowalkowski, J., Perdue, G., Spentzouris, P., Tran, N., Vlimant, J.-R., Zlokapa, A., Pata, J., Spiropulu, M., An, S., Aurisano, A., & Hewes, V. (2020). Graph Neural Networks for Particle Reconstruction in High Energy Physics detectors. arXiv:2003.11603. <https://doi.org/10.48550/arXiv.2003.11603>.
- [40] Ørsøe, R. F., Rosted, A., & GraphNeT Team. (2025). GraphNeT 2.0 -- A deep learning library for neutrino telescopes. arXiv:2501.03817. <https://doi.org/10.48550/arXiv.2501.03817>.
- [41] Heyer, N., Glaser, C., & Glösenkamp, T. (2023). Deep Learning Based Event Reconstruction for the IceCube-Gen2 Radio Detector. arXiv:2308.00164. <https://doi.org/10.48550/arXiv.2308.00164>.
- [42] Heyer, N., Glösenkamp, T., & Glaser, C. (2024). Deep learning reconstruction of neutrino direction, energy, and flavor with complete uncertainty predictions. arXiv:2410.09573. <https://doi.org/10.48550/arXiv.2410.09573>.
- [43] Koh, D. H., Mishra, A., & Terao, K. (2021). Evaluating deep learning uncertainty quantification methods for neutrino physics applications. (NeurIPS 2021 Workshop). Stanford University.

- [44] Hünnefeld, M., Abbasi, R., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Ahrens, M., Alispach, C., Alves Jr., A. A., Amin, N. M.,... Zhang, Z. (2021). Combining Maximum-Likelihood with Deep Learning for Event Reconstruction in IceCube. *PoS (ICRC2021)*, 1065. <https://doi.org/10.22323/1.395.1065>.
- [45] Bellenghi, C., Glauch, T., Haack, C., Kontrimas, T., Niederhausen, H., Reimann, R., & Wolf, M. (2021). A new and improved IceCube point source analysis. *Journal of Instrumentation*, **16**(11), C11002. <https://doi.org/10.1088/1748-0221/16/11/C11002>.
- [46] Chappell, A., & Whitehead, L. H. (2022). Application of transfer learning to neutrino interaction classification. *Eur. Phys. J. C*, **82**(1099), 1099. <https://doi.org/10.1140/epjc/s10052-022-11066-6>.
- [47] Abbar, S., & Nagakura, H. (2023). Detecting fast neutrino flavor conversions with machine learning. arXiv:2310.03807. <https://doi.org/10.48550/arXiv.2310.03807>.
- [48] Chiyangwa, T. B., & Mhlanga, S. T. (2024). Exploring the performance of deep learning models for neutrino direction prediction in high-energy astrophysics. In K. Arai (Ed.), *Proceedings of the Future Technologies Conference (FTC)*, **3**, 452-467. Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-031-73125-9_29.
- [49] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929. <https://doi.org/10.48550/arXiv.2010.11929>.
- [50] Alonso-Monsalve, S., Sgalaberna, D., Zhao, X., Molines, A., McGrew, C., & Rubbia, A. (2024). Deep-learning-based decomposition of overlapping-sparse images: application at the vertex of simulated neutrino interactions. *Communications Physics*, **7**(1), 173. <https://doi.org/10.1038/s42005-024-01669-8>.
- [51] Alin, H. (2024). *Development of a transformer model for the detection of ultra-high energy neutrinos* (PhD thesis), Uppsala Universitet. <https://www.diva-portal.org/smash/record.jsf?pid=diva2:1860270>.
- [52] Bukhari, H., Chakraborty, D., Eller, P., Ito, T., Shugaev, M. V., & Ørsøe, R. (2023). IceCube - Neutrinos in Deep Ice: The Top 3 Solutions from the Public Kaggle Competition. arXiv:2310.15674. <https://doi.org/10.48550/arXiv.2310.15674>.

- [53] **Eller, P., Abbasi, R., Ackermann, M., Adams, J., Agarwalla, S. K., Aguilar, J. A., Ahlers, M., Alameddine, J. M., Amin, N. M., Andeen, K., . . . Zimmerman, M.** (2023). Public Kaggle Competition “IceCube - Neutrinos in Deep ice”. *PoS (ICRC2023)*, 1609. <https://doi.org/10.22323/1.444.1609>.
- [54] **Shaw, P., Uszkoreit, J., Vaswani, A.** (2018). Self-Attention with Relative Position Representations. arXiv:1803.02155. <https://doi.org/10.48550/arXiv.1803.02155>.
- [55] **Li, C., Cai, H., & Jiang, X.** (2024). Refine Neutrino Events Reconstruction with BEiT-3. *Journal of Instrumentation*, **19**(6), T06003. <https://doi.org/10.1088/1748-0221/19/06/T06003>.
- [56] **Particle Data Group (Naves, S., et al.).** (2024). Neutrino properties. In Review of particle physics. *Physical Review D*, **110**(3), 030001. <https://doi.org/10.1103/PhysRevD.110.030001>.
- [57] **Url-2** <<https://icecube.wisc.edu/>> , date retrieved 26.11.2025.
- [58] **Aartsen, M. G., Ackermann, M., Adams, J., Aguilar, J. A., Ahlers, M., Ahrens, M., Altmann, D., Andeen, K., Anderson, T., Anseau, I., Anton, G., Archinger, M., Argüelles, C., Auer, R., Auffenberg, J., Axani, S., Baccus, J., Bai, X., Barnett, S., . . . Zoll, M.** (2017). The IceCube Neutrino Observatory: Instrumentation and online systems. *Journal of Instrumentation*, **12**(3), P03012. <https://doi.org/10.1088/1748-0221/12/03/P03012>.
- [59] **Halzen, F., & Klein, S. R.** (2010). IceCube: An instrument for neutrino astronomy. *Review of Scientific Instruments*, **81**(8), 081101. <https://doi.org/10.1063/1.3480478>.
- [60] **Wen, Q., Zhou, T., Zhang, C., Chen, W., Ma, Z., Yan, J., & Sun, L.** (2023). Transformers in time series: A survey. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI 2023)* (pp. 6750-6760). International Joint Conferences on Artificial Intelligence Organization. <https://doi.org/10.24963/ijcai.2023/759>.
- [61] **Hawkins, D. M.** (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, **44**(1), 1-12. <https://doi.org/10.1021/ci0342472>.
- [62] **Nair, V., & Hinton, G. E.** (2010). Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*.
- [63] **Goodfellow, I., Bengio, Y., & Courville, A.** (2016). Deep Learning. *MIT Press*. (Section 6.2.2). <https://www.deeplearningbook.org/>.

- [64] **Kingma, D. P., & Ba, J.** (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. <https://doi.org/10.48550/arXiv.1412.6980>.
- [65] **Bishop, C. M.** (2006). Pattern Recognition and Machine Learning. *Springer*. <https://doi.org/10.1007/978-0-387-45528-0>.
- [66] **Cameron, A. C., & Windmeijer, F. A.** (1997). An R-squared measure of goodness of fit for some common nonlinear regression models. *Journal of Econometrics*, 77(2), 329-342. [https://doi.org/10.1016/S0304-4076\(96\)01818-0](https://doi.org/10.1016/S0304-4076(96)01818-0).



CURRICULUM VITAE

Name Surname : Şevval Coşkun

EDUCATION :

- **B.Sc.** : 2021, Yıldız Technical University, Faculty of Arts and Sciences, Department of Physics

PROFESSIONAL EXPERIENCE:

- 2023-Present: Albaraka Türk Participation Bank, Business Excellence and Innovation Department, Open Banking Specialist.

PUBLICATIONS, PRESENTATIONS AND PATENTS ON THE THESIS:

- **Coşkun, Ş.**, Kahya, E. O. 2025: Neutrino Particle Direction Estimation Using Transformer-Based Deep Learning. 4th International Graduate Research Symposium (IGRS'25), May 12-14, 2025, Istanbul, Türkiye, p. 2478. (Extended Abstract)
- **Coşkun, Ş.**, Kahya, E. O. 2025: Applying Transformer Architectures for Neutrino Direction Estimation. Turkish Physical Society 41st International Physics Congress (TPS 41), September 1-5, 2025, Bodrum, Türkiye. (Poster Presentation)