Google Summer of Code 2025

Quantum Representations of Classical HEP Data with Contrastive Learning

Details:

GSoC Contributor Name: Kushal Trivedi

Degree: Integrated B.Tech+M.Tech in IT

College: Indian Institute of Information Technology, Gwalior, India

E-Mail ID: kushal.trivedi.2005@gmail.com

Contact Number: +91-9152203790

Timezone: India/UTC+5:30, Indian Standard Time

Languages: English, Hindi, Gujarati, French

Social Handles: LinkedIn profile, X profile, Gitter: @kushaltrivedi19032005

TABLE OF CONTENTS

Index of Contents **1**

1 Overview **2**

1.1 Project Synopsis . . . 2

1.2 Impact on Scientific Community . . . 3

1.3 Background Research . . . 3

2 Goals and Deliverables **6**

2.1 Deliverables . . . 6

2.2 Prerequisite Tests . . . 6

2.3 Implemented Work and Observations . . . 6

2.4 Future Research Directions for GSoC . . . 7

3 Schedule of Deliverables **8**

3.1 Application Review Period . . . 8

3.2 Community Bonding Period . . . 9

3.3 Programming Period . . . 9

4 Biographical Infomation **10**

4.1 Academic Details . . . 10

4.2 Motivation for Quantum Machine Learning . . . 10

5 Availability Schedule **11**

5.1 Working Hours . . . 11

5.2 Mentor-Mentee Meetings and Updates . . . 11

5.3 Post Google Summer of Code ’25 . . . 12

1. Overview

1.1. Project Synopsis

The Large Hadron Collider (LHC) is the world's largest particle accelerator. With the upcoming High-Luminosity Large Hadron Collider (HL-LHC), the need for extensive compute resources will rise. The HL-LHC aims to increase the machine's luminosity (simply the collision frequency per unit area) by a factor of five to ten [1]. However, balancing the demand for compute resources with replacing current classical methods of handling and generating data representations with potentially better quantum methods requires research.

With the coming up of the new HL-LHC, there are three main challenges which need to be solved:

- **Event Classification and Anomaly Detection:** Tracking and distinguishing different types of particle collisions at the LHC is essential. Manually labeling data is a cumbersome task, but learning features without labels based on the inherent structure of the particles takes more time for training but less time for data handling. The discovery of anomalous particles, such as the Higgs boson in 2012 [7], has led to greater curiosity about improving the task of detecting anomalies among normal observations.
- **Dimensionality Reduction and Feature Learning:** Particle collision data often exists in very high-dimensional spaces, with data points from multiple detectors (such as energy, momentum, etc.). High-dimensional data makes pre-processing, analyzing, and visualizing the data more time-consuming and computationally heavy. Therefore, projecting this data to low-dimensional spaces is important, but without losing a significant amount of essential information. Extracting correlations between different features becomes easier as well. This is particularly important for tasks like distinguishing between jets originating from quarks or gluons, where subtle differences in the particle signatures are critical.

In this project, we will implement Quantum Machine Learning methods for LHC-HEP analysis using the Pennylane framework.

1.2. Impact on Scientific Community

- *Improved Data Analysis:* Better representations of data in scientific applications like particle physics or medical imaging.
- *Advancement in QML:* Extension of research into the potential of quantum solutions.

1.3. Background Research

Contrastive learning is an approach that helps in setting up a contrast between positive pairs and negative pairs in a learned embedding space. It is a discriminative task based on the assumption that similar instances should be closer and dissimilar instances should be farther away in the embedding space.

There are various ways to project images into the embedding space, one being the population augmentation graph [5]. In this approach, high-probability edges exist between images with similar content, while those with different semantic content, like dogs with different poses, have fewer connections. Similar images are linked via interpolated transformations. Euclidean distance or cosine similarity can be used to compare distances between embeddings.

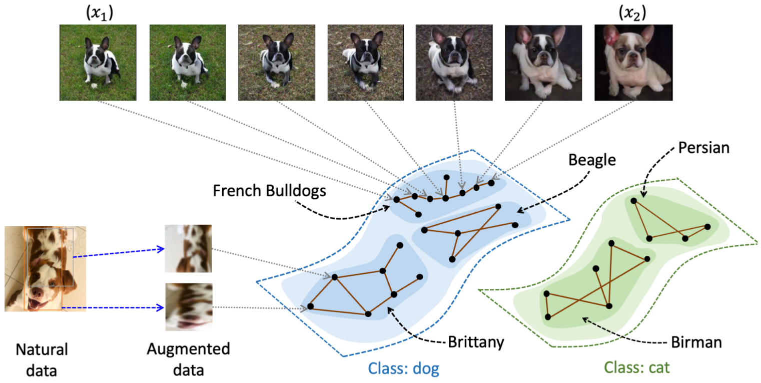


Figure 1.1: Embedding space of data images

However, contrastive learning can further be classified into supervised and self-supervised forms. Our project is based on the idea of self-supervised contrastive learning, in which no labels are provided for two images, and the task is to bring the embeddings of similar instances closer while pushing those of dissimilar instances farther apart. However, this method requires a larger training dataset, introducing the need for data augmentation.

A simple model for contrastive loss can be illustrated as follows [3]:

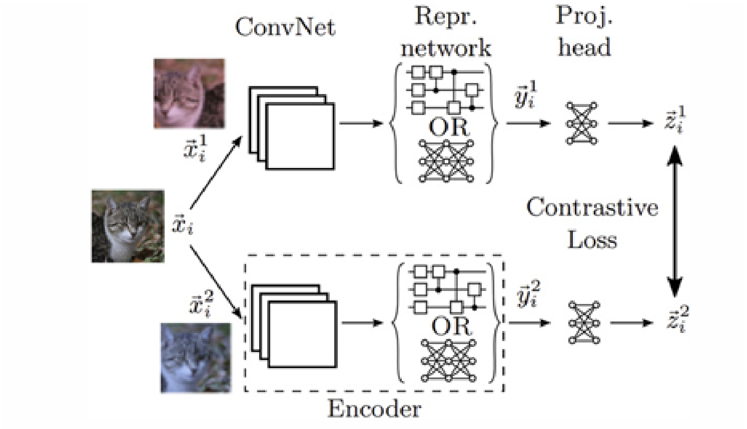


Figure 1.2: Classical architecture for contrastive loss model

In this architecture, the images are fed into ConvNets to extract high-level and low-level features. The representation network maps the extracted features to a representation space. Lastly, the projection head reduces the dimensionality of these embeddings and projects them into a new space, called the embedding space, to enhance the network's discriminative power.

There are different contrastive learning frameworks [4], such as SimCLR, MoCo, and BYOL, that could be employed for our project.

Various loss functions [6] can be used in contrastive learning, including triplet loss, N-pair loss, contrastive loss (which I have used), and InfoNCE. InfoNCE is similar to contrastive loss but utilizes cosine similarity, the softmax function, and a temperature parameter, which controls the sharpness of the distribution. This allows for a balance between prioritizing harder negatives and treating all negative pairs more evenly.

The InfoNCE loss function is defined as:

$$L = -\log \frac{\exp(\text{sim}(z_i, z_i^+)/\tau)}{\sum_j \exp(\text{sim}(z_i, z_j)/\tau)}$$

The contrastive loss function is given by:

$$L = (1-Y) \cdot \frac{1}{2} D^2 + Y \cdot \frac{1}{2} \max(0, m - D)^2$$

where: Y indicates whether the pair is positive ($Y = 0$) or negative ($Y = 1$), - D represents the Euclidean distance between the embeddings, - m is the margin parameter.

The quantum extension of this is also proposed earlier, in which the input data vector is encoded into qubits, followed by variational ansatz layers of qubit rotations. (A similar but not entirely identical approach was used in the code part as well).

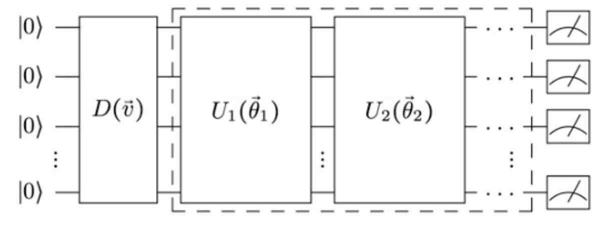


Figure 1.3: General structure of QNN

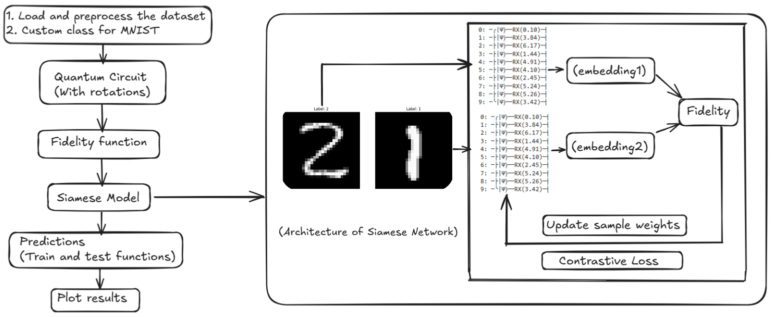
The quantum extension of this is also proposed earlier, in which the input data vector is encoded into qubits, followed by variational ansatz layers of qubit rotations.

Each variational ansatz layer was shown to produce the best representations when each qubit was given an $R_{\hat{y}}$ rotation followed by controlled $R_{\hat{x}}$ rotations in a ring topology fashion.

Application of a New Siamese Network for QMLHEP Tasks:

- *Jet tagging and particle classification in calorimeter images*, which are grid-like energy heatmap structures used in high-energy physics experiments. These models help improve the identification of different particle types.
- *Detection of rare or new particles* by analyzing patterns in experimental data, aiding in the discovery of new physics beyond the Standard Model.

Proposed Model Architecture:



```
0: ─╭|Ψ⟩──RX(0.10)─┤
1: ─├|Ψ⟩──RX(3.84)─┤
2: ─├|Ψ⟩──RX(6.17)─┤
3: ─├|Ψ⟩──RX(1.44)─┤
4: ─├|Ψ⟩──RX(4.91)─┤
5: ─├|Ψ⟩──RX(4.10)─┤
6: ─├|Ψ⟩──RX(2.45)─┤
7: ─├|Ψ⟩──RX(5.24)─┤
8: ─├|Ψ⟩──RX(5.26)─┤
9: ─╰|Ψ⟩──RX(3.42)─┤
```

```
0: ─╭|Ψ⟩──RX(0.10)─┤
1: ─├|Ψ⟩──RX(3.84)─┤
2: ─├|Ψ⟩──RX(6.17)─┤
3: ─├|Ψ⟩──RX(1.44)─┤
4: ─├|Ψ⟩──RX(4.91)─┤
5: ─├|Ψ⟩──RX(4.10)─┤
6: ─├|Ψ⟩──RX(2.45)─┤
7: ─├|Ψ⟩──RX(5.24)─┤
8: ─├|Ψ⟩──RX(5.26)─┤
9: ─╰|Ψ⟩──RX(3.42)─┤
```

Figure 1.4: Quantum adaptation of Siamese Network

2. Goals and Deliverables

2.1. Deliverables

1. Start with a function that converts HEP dataset images into quantum states.
2. Experiment with different circuits, each yielding different quantum embeddings and fidelity results.
3. Compare each quantum model with its previous versions to evaluate performance improvements.
4. Compare the representation spaces of quantum encoding models and classical encoding models.

2.2. Prerequisite Tests

The solutions to common and specific tasks I have completed can be found here:
`https://github.com/KushalTrivedi19032005/QMLHEP`

2.3. Implemented Work and Observations

As mentioned in Task VI of the QMLPHEP document, I began my work by creating a simple function that takes an image from the MNIST dataset (which had been loaded and pre-processed earlier) and returns the quantum state (which was later extended to include two images as arguments too).

There are primarily three ways of encoding the data of image pixels into quantum information: Amplitude Encoding, Angle Encoding, and Basis Encoding.

For amplitude encoding, we first normalize the pixel values, and then we map this normalized pixel vector to quantum states. Mathematically,

$$\mathbf{p} = \left[\frac{p_1}{\sum p_i}, \frac{p_2}{\sum p_i}, \dots, \frac{p_N}{\sum p_i}\right]$$

$$|\psi\rangle = \sum_{i=1}^{N} \alpha_i |i\rangle$$

where,

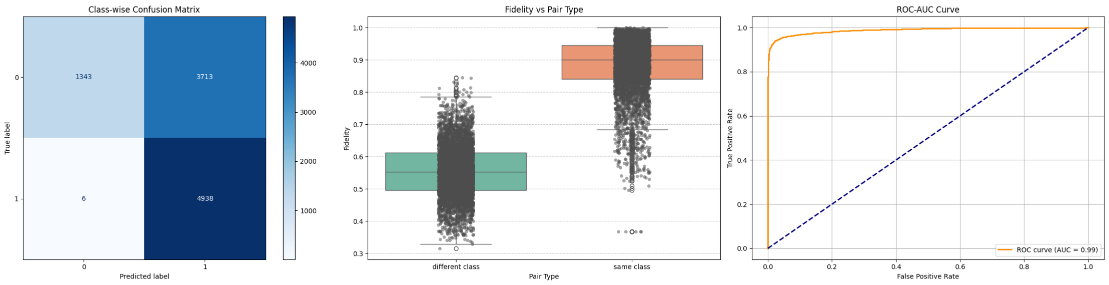
- $\alpha_i = \sqrt{\frac{p_i}{\sum p_i}}$ are the amplitudes, and
- $|i\rangle$ are the computational basis states.

Further, I implemented the SWAP test and functions for calculating fidelity, creating quantum embeddings using the above circuit, and defining the class for contrastive loss.

I also implemented a custom class to obtain pairwise images from the MNIST dataset—positive pairs (images from the same class) and negative pairs (images from different classes).

The main task was to implement the Siamese model. It flattens the images from a pair, calls the quantum embeddings function, and compares them. After each epoch, the sample weights (which are parameters of the model) are updated during backpropagation, and the loss value is updated.

Note : Since no pre-processing was applied, the training time per epoch was long. Therefore, the model was trained for just five epochs.



A few important observations from these plots are :

- The high number of False Positives (3713) suggests that the model tends to predict class 1 more often, potentially leading to a high recall but lower precision for class 1.
- For different-class pairs, median fidelity score is around 0.55, and for same-class pairs it is around 0.90. Some overlap might suggest misclassifications.
- AUC = 0.99 is an excellent measure.

2.4. Future Research Directions for GSoC

- After training the model, it was observed that the loss started saturating earlier than expected. Experimenting with hyperparameters such as the margin parameter and replacing the simple learning rate with learning rate schedulers could yield interesting results.

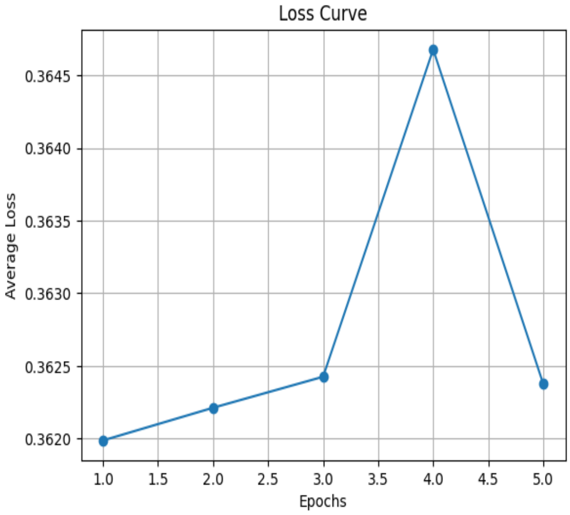


Figure 2.1: Loss curve for the Siamese network with Contrastive loss

- Pre-processing MNIST images could reduce computational costs. Since no pre-processing was applied and all 28×28 pixels were processed directly in the model, a future direction could involve methods such as Principal Component Analysis (PCA).
- Amplitude encoding is difficult to scale for large images, so trying angle encoding in the quantum circuit and comparing performances could be beneficial.
- Visualizing both quantum and classical embeddings in a low-dimensional t-SNE space could provide deeper insights.
- Using different contrastive learning frameworks like SimCLR or MoCo and different loss function like InfoNCE.

3. Schedule of Deliverables

3.1. Application Review Period

(8th April - 8th May): During this assessment period, I will further enhance my programming skills in PennyLane and the necessary libraries for this project. Additionally, I will use this one-month period to deepen my theoretical understanding of quantum computing and explore potential experimental ideas that could be implemented during the coding phase.

3.2. Community Bonding Period

(8th May - 1st June): During this period, I will focus on getting to know the mentors, thoroughly reading the project documentation, and familiarizing myself with the details. Afterward, I will discuss ideas with the mentors, gather feedback, finalize the datasets, iterate on improvements, and ultimately begin working once all conditions are deemed satisfactory.

3.3. Programming Period

Phase 1 (2nd June - 14th July):

- Set up the project environment for training and testing (including necessary libraries/tools).
- Implement the classical Siamese network and Triplet loss
- Next, compare the quantum model with the classical model
- Fine-tuning of hyper-parameters

Phase 2 (14th July - 1st September):

- Work on the suggestions provided in the future research directives.
- Thorough documentation of all work completed during Google Summer of Code (GSoC '25) in GitHub README files, Medium blog posts, and presentations for the end-term evaluations.
- The documentation will cover:
  - Prior research and initial contributions before the cohort began.
  - Discussions and brainstorming sessions with mentors during the bonding period.
  - Final implementation of the models.

4. Biographical Infomation

4.1. Academic Details

I am Kushal Trivedi, a second-year undergraduate (sophomore) pursuing an Integrated B.Tech + M.Tech degree in Information Technology at the Atal Bihari Vajpayee Indian Institute of Information Technology and Management, Gwalior, India. Currently, I am in my fourth semester with a CGPA of 8.41 out of 10, placing me in the top 15% of my batch.

From 2021 to 2023, I prepared for the Joint Entrance Examination (JEE) Mains and Advanced, highly competitive entrance exams for admission to India's prestigious research and engineering institutes. I secured a rank in the top 1 percentile in both exams, which are taken by approximately 12–13 lakh students each year, leading to my admission to this institute.

Over the past two years, I have been actively practicing Machine Learning and Artificial Intelligence. I am familiar with Rust, Java, and C, and I consider myself proficient in C++ and Python.

Alongside my passion for computer science, I have always been deeply interested in the natural sciences, particularly Physics and Mathematics. My fascination with particle physics began in middle school after reading the globally renowned book *A Brief History of Time* by *Stephen Hawking*. More recently, attending the International Conference on Applied AI and Scientific Machine Learning provided me with further insights into the intersection of physics and programming.

Beyond academics, I actively participate in nationwide hackathons. I was a winner of Smart India Hackathon '24, where my team developed a multimodal chatbot for Bharat Electronics Limited, supporting defense utilities for the Indian Army. I have also competed in hackathons like IIT Roorkee's TechFest Hackathon, Convolve 3.0, and more.

4.2. Motivation for Quantum Machine Learning

Simply put, I truly value the opportunity provided by Google and ML4SCI QML-HEP. My passion for quantum physics dates back to my school days when I had little knowledge of advanced sciences. However, as I started preparing for the JEE and later took it as a course in college, I began to understand it piece by piece, admiring both its complexity and its real-world applications. Last summer, I started studying Quantum Computing using textbooks and online resources. However, there was no way to apply the knowledge or gain experience since it is still a niche field with limited opportunities. Therefore, this initiative is the perfect platform to connect with like-minded individuals, further hone my skills, and also provide me with a pathway to pursue my postgraduate studies in the subject.

5. Availability Schedule

5.1. Working Hours

I can commit the required time for the project to achieve the timely deliverables as outlined below:

- **Working Timimgs for Weekdays :**
  Preferred Timings: Between 8 p.m. to 2 a.m. IST
- **Working Timings for Weekends :**
  Preferred Timings: Between 10 a.m.to 1 p.m. IST and/or 4 p.m. to 8 p.m.

I have summer holidays from May 4 to July 28, which allows me to work flexibly according to the needs of the project, as directed by the mentors. After the summer break, I have college classes scheduled from 9 a.m. to 5 p.m. (with some free slots), which will allow me to work comfortably in the late evenings or nights, from 8 p.m. to 2 a.m., depending on the project requirements.

As additional information, I am also actively seeking research internship opportunities. Should I be selected for one, I assure you that it will not delay the project's progress. In such a case, I will be able to work more comfortably in the late evenings onwards.

5.2. Mentor-Mentee Meetings and Updates

I can assure you that I will work in the following manner:

- Weekly meetings with mentors (preferably on Google Meet/Zoom, but I'm flexible with any platform) to discuss new progress, including both theoretical ideas and code implementations.

- Pushing the code to GitHub repositories after seeking feedback, and correcting the code if negative feedback is received, as well as proper documentation of every step.
- Finalizing the current steps and creating a list of the next achievable tasks.

5.3. Post Google Summer of Code ’25

I believe that open-source initiatives are not limited to the timespan of the program; contributions beyond that period also matter. I would love to continue working on these projects after GSoC and certainly apply as a mentee next year, or perhaps even as a mentor!

References

[1] Information on LHC and HL-LHC. Available at: https://hilumilhc.web.cern.ch/ https://home.cern/science/accelerators/large-hadron-collider.

[2] Pennylane Documentation. Available at: https://pennylane.ai/.

[3] B. Jaderberg, L. W. Anderson, W. Xie, S. Albanie, M. Kiffner, and D. Jaksch, ”Quantum Self-Supervised Learning.” Available at: https://iopscience.iop.org/article/10.1088/2058-9565/ac6825.

[4] ”Full Guide to Contrastive Learning.” Available at: https://encord.com/blog/guide-to-contrastive-learning/.

[5] ”Understanding Deep Learning Algorithms that Leverage Unlabeled Data.” Available at: https://ai.stanford.edu/blog/understanding-contrastive-learning/.

[6] ”An In-Depth Guide to Contrastive Learning: Techniques, Models, and Applications.” Available at: https://myscale.com/blog/what-is-contrastive-learning/.

[7] LHC Facts and Figures. Available at: https://cds.cern.ch/record/2809109/files/CERN-Brochure-2021-004-Eng.pdf.

[8] Previous Work in GSoC ’24. Available at: https://sanyananda.github.io/ML4Sci_QuantumContrastiveLearning/ https://medium.com/@sanya.nanda/quantum-contrastive-learning-on-lhc-hep-dataset-1b3084a0b141.