Google Summer of Code 2025

Quantum Kolmogorov-Arnold Networks for High Energy Physics at LHC

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, Gitter: @kushaltrivedi19032005

TABLE OF CONTENTS

1. Overview

1.1. Project Synopsis

The Large Hadron Collider (abbreviated as LHC) is the world's largest and highest-energy particle accelerator.

A major upgrade of the LHC, called the High-Luminosity Large Hadron Collider (HL-LHC) aims to increase the machine's luminosity (simply the collision frequency per unit area) by a factor of between five and ten. The goal is to study in more detail the fundamental components of matter and the forces that bind them together.

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

- **Computational Bottlenecks**
  The upgraded machine is planned to operate at the same energy as the LHC and to produce more than 250 inverse femtobarns of data per year (about 3-5 times the amount of data LHC produces per year) [7]. Handling such large amounts of data presents two challenges (and researching on quantum computing and QKANs could provide solutions to these issues):

  1. Developing data storage facilities capable of managing vast amounts of data.
  2. Designing new data processing hardware and software tools.

  Quantum computing can be brought in to handle the exponential rise in data.

- **Improved pattern recognition for Rare Event Detection** It took the ATLAS and CMS experiments more than two years to gather sufficient evidence for the existence of the Higgs boson. Its production is extremely rare—occurring in only about 1 in 1,000,000,000 collisions. This necessitates the development of new methods for enhanced pattern recognition to detect potential new subparticles or signals. [7]

- **Potential Speedup over classical calculations and function approximation** With large volumes of data, pre-processing and numerical analysis become increasingly challenging. Additionally, the scientific community often relies on deriving formulas based on observed events. Research on Quantum Kolmogorov-Arnold Networks (QKANs) would be highly suitable for accurate function approximation in this context.

1.2. Impact on Scientific Community

- Exploring applications of classical and quantum KANs in particle physics and other scientific applications.
- Enhancing rare signal detection for new physics discoveries.
- Advancing quantum computing applications in HEP.

1.3. Background Research

Large Hadron Collider and the Universe:
The universe is composed of four fundamental forces: the strong force, weak force, gravitational force, and electromagnetic force, as proposed in the Standard Model of particle physics. The early universe, just after the Big Bang, was a hot soup of subatomic particles called Quark-Gluon Plasma (QGP) with high energy density and temperature, some of which combined to form the particles we observe today [1].

The primary purpose of the LHC is to accelerate and collide two particles of the same kind—hadrons (particles composed of quarks), resulting in the production of particle jets, which can lead to the creation of particles never seen before, albeit for a very short moment of time before they decay.

Kolmogorov-Arnold Networks [5] (or KANs) were proposed in April 2024, primarily differing from standard Multi-Layer Perceptrons (MLPs) in the following ways:

- Each weight parameter is replaced by a learnable 1D function parametrized as a spline.
- Sum operation on nodes, and learnable activation function on edges.
- Interpretable and accurate representation of special functions.

Although KANs are known to have their downsides too, like slow training and catastrophic forgetting during continual learning, KANs are proved suitable for scientific purposes. Some of the problems faced by KANs with B-Spline activations are solved partially by **SineKANs** [4] such as catastrophic forgetting due to continuous and periodic activations.

KANs are mathematically based on the Kolmogorov-Arnold theory stating: any multivariate continuous function f can be written as a finite composition of a single variable and the binary operation of addition. Mathematically,

$$f(x) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^{n} \phi_{q,p}(x_p) \right)$$

Although originally proposed for a 2-layer structure, KANs can also be extended to multiple layers.

Application of Quantum KANs for QMLHEP tasks:

- Signal Extraction: Overcoming classical limitations like SNR ratio, QKANs can utilize quantum phenomena such as superposition and entanglement to extract complex patterns from multiple frames of particle collisions due to their ability to capture information in qubits.
- Modeling Non-linear and High-dimensional Functions: Using QKANs to model complex processes, such as particle decay rates and energy distributions, which are typically difficult to represent with classical networks due to their non-linearity and high-dimensional nature.
- Data Storage: QKANs, along with dimensionality reduction techniques like QPCA, can be used for the extraction of important features from high-dimensional datasets.

Quantum KANs Architecture:

A brief quantum circuit for QKANs is shown below:

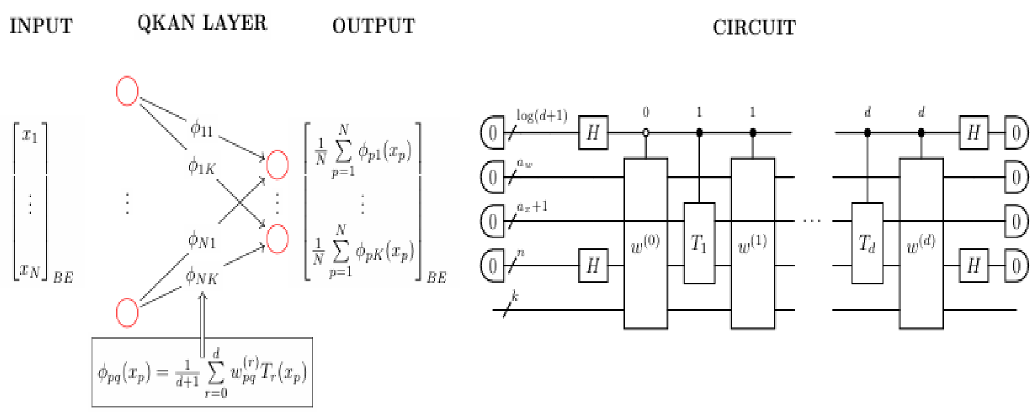

Figure 1.1: QKAN Circuit

To explain in brief the architecture of the circuit, it takes qubits in the $|0\rangle$ state and performs a series of operations like applying the Hadamard gates (H) to block-encode them into quantum states.

Transformation on bits is the next stage involving $w_{pq}^{(r)}$ which are learnable weights and $T_r(x)$ i.e. the Chebyshev polynomial evaluations. Mathematically, the transformations can be represented as,

$$\phi_{pq}(x) := \frac{1}{d+1} \sum_{r=0}^{d} w_{pq}^{(r)} T_r(x_p)$$

or,

$$\Phi(x) = \left(\frac{1}{N} \sum_{p=1}^{N} \phi_{p1}(x_p), \ldots, \frac{1}{N} \sum_{p=1}^{N} \phi_{pK}(x_p) \right)^{\top}$$

After all the transformations, the output vector is a K-dimensional vector, which can be used for further quantum tasks or direct classical measurement.

The construction of Cheb-QKAN is based on these fundamental steps- DILATE, CHEB, MUL, LCU and SUM.

Below is a flowchart which briefly describes each operation:

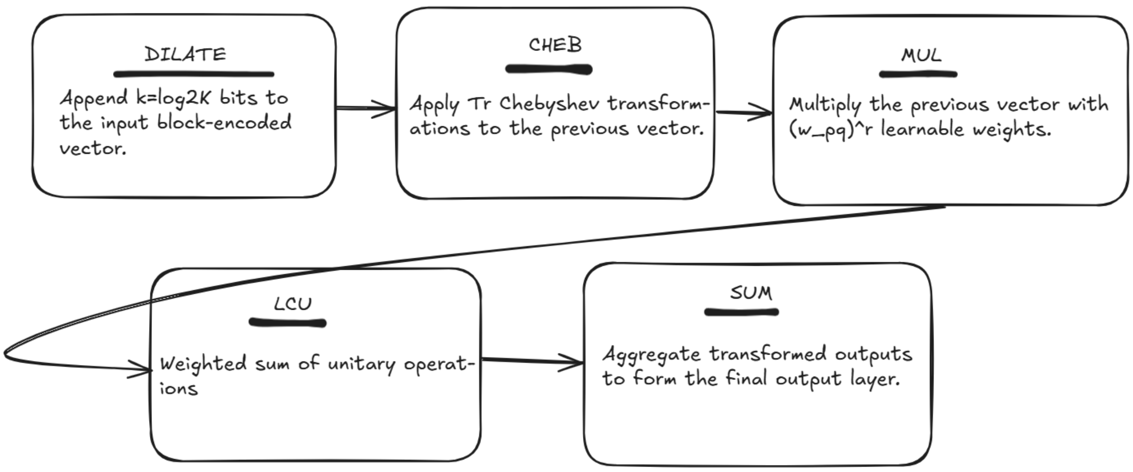

Figure 1.2: Flowchart for CHEB-QKAN Architecture

2. Goals and Deliverables

2.1. Deliverables

1. Start with a classical KAN and test its abilities on datasets, such as the HEP dataset.

2. Extend the idea of classical KANs to quantum KANs, compare both models and benchmark the models on the basis of -

- **Performance Metrics:** Accuracy, F1 Score, ROC-AUC Score, etc.
- **Resource Efficiency:** Training and testing time (time complexity), memory utilization (space complexity).
- **Quantum-specific Metrics:** Circuit dimensions (width and depth), fidelity, and avoiding barren plateaus.

3. Well-maintained and up-to-date documentation, including analysis and reports of the work completed.

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

As mentioned in Task IX of the QMLPHEP document, I started my work by implementing the classical KAN model on the MNIST dataset.

Since the pykan library only works for function approximation on numerical data, I implemented a separate class for the KAN using the BSpline activation function from the SciPy library.

(Note: I ended up overshooting the loss due to training for an extra number of epochs. Training for fewer epochs and using methods like an LR scheduler and early stopping could help improve the results moving forward.)

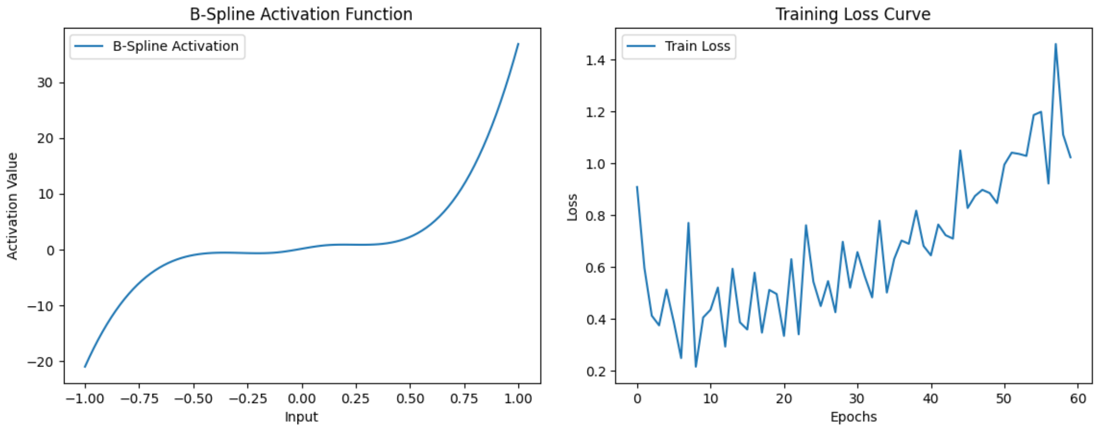

Figure 2.1: (Left) BSpline Activation Function used in the KAN model. (Right) Loss curve during the training of the KAN model.

To compare the performance of the classical KAN, I tested it against various models, such as SineKAN, Quantum Convolutional Neural Network (QCNN), Convolutional Neural Network (CNN), and a standard MLP. I have listed the results in the order of the models described above:

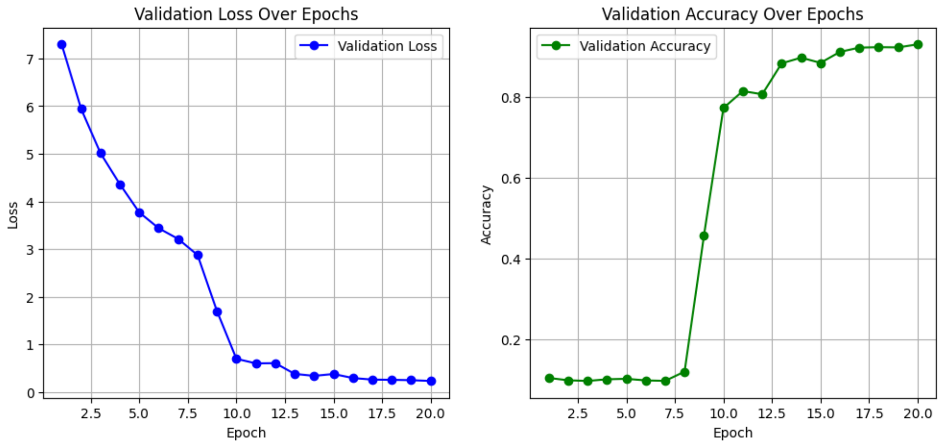

Figure 2.2: SineKAN Architecture

2.4. Future Research Directions for GSoC

- With classical KANs, it has been shown that SineKAN outperforms KANs with B-Spline activation and FourierKAN. Similarly, we could implement a SineQKAN and compare its results with the Cheb-QKAN, which has been proposed so far.
- Currently, QKANs have only been benchmarked using standard metrics. Further research on the extension of QKAN applications to scientific phenomena is still needed.
- Similar to Generative Adversarial Networks (GANs), where hybrid models exist with either the generator or discriminator (usually the generator) in quantum implementations, similar experiments can be conducted with KANs.

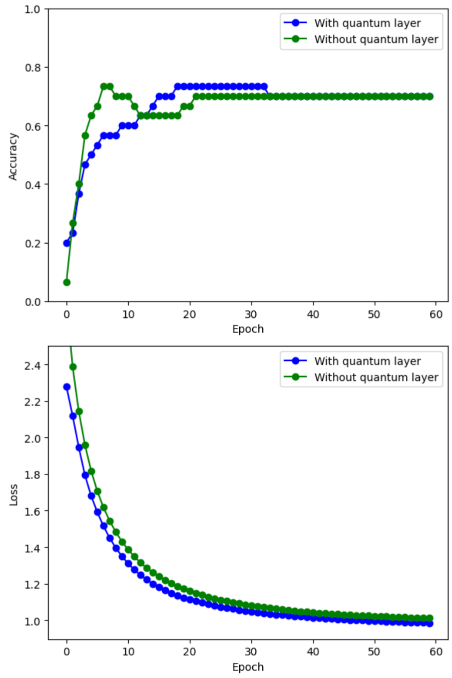

Figure 2.3: Quantum Convolutional Neural Network (QCNN)

Figure 2.4: Convolutional Neural Network (CNN)

- Trying different optimizers like LGBFS apart from Adam optimizer.

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 the QKANs project. Additionally, I will use this one-month period to deepen my theoretical understanding of QKANs 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, especially since this project is being introduced for the first time in ML4SCI. This is expected to take some time. 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):

- **Week 1: Classical KAN Model Implementation**
  - Set up the project environment for training and testing (including necessary libraries/tools).
  - Implement the classical KAN model as a baseline on the new HEP dataset.
  - Identify strengths and weaknesses of the classical KAN model.
- **Week 2 and Week 3: Implementing QKAN Model**
  - Research and implement the Quantum Kolmogorov-Arnold Network (QKAN) model.
  - Train and test the QKAN model on HEP dataset.
  - Compare the performance of the QKAN model with the classical KAN model.
- **Week 4: Performance Metrics and Evaluation**
  - Define performance metrics (e.g., accuracy, loss, training time).
  - Evaluate both the classical KAN and QKAN models using these metrics.
  - Document findings and identify areas for further research and improvements.

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

- **Week 5: QKAN Model Refinement**
  - Complete any incomplete work related to the QKAN model.
  - Test the performance of the QKAN model with different loss functions and optimizers.
- **Week 6: SineQKAN Extension**
  - Extend QKANs to SineQKANs as suggested in previous sections.
  - Implement and test the performance of SineQKAN.
- **Week 7 and Week 8: Hybrid KANs and Documentation**
  - Explore the possibility of a hybrid version of KANs, if feasible.
  - Thoroughl 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 my academic pursuits, I enjoy participating in nationwide hackathons. I was a winner of the Smart India Hackathon '24, a government-initiated competition aimed at developing hardware and software solutions to real-world challenges. In this event, my team developed a multimodal chatbot for Bharat Electronics Limited, a major public-sector organization supplying defense utilities to the Indian Army. Additionally, I have participated in several other hackathons, including 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 (3-4 hours daily)**
  Preferred Timings: 8 p.m. to 2 a.m. IST
- **Working Timings for Weekends (4-6 hours daily)**
  Preferred Timings: 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/` and `https://home.cern/science/accelerators/large-hadron-collider`.

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

[3] Liu, et al., "KAN: Kolmogorov-Arnold Networks," *arXiv*, 2024. Available at: `https://arxiv.org/abs/2404.19756`.

[4] Reinhrdt, E., Ramakrishnan, D., Gleyzer, S., "SineKAN: Kolmogorov-Arnold Networks using sinusoidal activation functions," *Frontiers in Artificial Intelligence*, 2024. Available at: `https://www.frontiersin.org/journals/artificial-intelligence/articles/10.3389/frai.2024.1462952/full`.

[5] Ivashkov, P., Huang, P.-W., Koor, K., Pira, L., Rebentrost, P., "QKAN: Quantum Kolmogorov-Arnold Networks," *arXiv*, 2024. Available at: `https://arxiv.org/pdf/2410.04435`.

[6] B-Splines. Available at: `https://www.youtube.com/watch?v=YMl25iCCRew` and `https://rohangautam.github.io/blog/b_spline_intro/#b-splines`.

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