Project Proposal

Quantum Machine Learning for Exoplanet Characterization (EXXA)

Organisation : ML4SCI

Mentors :

- Katia Matcheva (University of Alabama)
- Konstantin Matchev (University of Alabama)
- Sergei Gleyzer (University of Alabama)
- Alex Roman (University of Alabama)
- Emilie Panek (University of Alabama)

Contact Information:

Full Name: Sourish Phate

Degree: Bachelor of Technology in Computer Engineering (B. Tech)

University: Veermata Jijabai Technological Institute (VJTI) , Mumbai

Location: Mumbai, Maharashtra ,India

Time Zone: Indian Standard Time (UTC+5.30)

Email: sourishphate@gmail.com, spsourish_b23@ce.vjti.ac.in

Github Profile: @sourishphate

Resume: Resume

1.1 About Me: Introduction & Background

Hello! Sourish Phate here. I am an undergraduate student in my second year at Veermata Jijabai Technological Institute (VJTI), Mumbai pursuing Computer Engineering as my major and a minor in Entrepreneurship.

1. As for relevant courses in the institute, I have taken Data Structures, Design and Analysis of Algorithms, Automata Theory, Operating Systems, Database Management Systems and Artificial Intelligence.
2. I have been developing projects and participating in various competitions since my first year.
    - Lip Reading: Changing the style of text while preserving its meaning.
    - Sustainable Travel Planner: The Sustainable Travel Planner is a web-based application that helps users plan customized trips efficiently. It leverages AI, real-time data to suggest an itinerary based on user preferences. Users can also access pre-made itineraries.
3. I have been contributing to open source for a brief period of time. Successfully completed Hacktoberfest in 2024.

1.2 Other Commitments

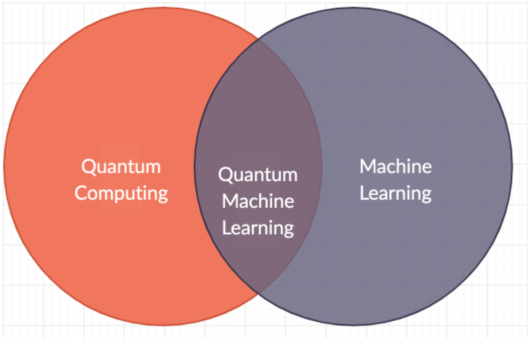
I can give 4 to 5 hours to the project everyday, compiling it to **30-35 hours per week** or even more than that if required.
I do not have any strict commitments to fulfil during this summer and can easily **increase my duration** if required. I am **flexible** with any working hours and can adjust them as per the time zone of my mentors.
I have my summer vacation from **19 May to 14 July** so I can easily get ahead of schedule in that duration. I will provide **regular updates** to my mentors regarding progress and will regularly attend the developer meets. I'll continue to post updates on my blog every **two weeks** for reference.
After the end of my vacation period I will continue my work each day after college hours on weekdays and put in extra hours on the weekends to ensure I have a steady progress and cover up on any delays I might encounter on other days.I might have **mid-semester exams in late August** and may need up to **two weeks off**, but I will plan ahead to ensure timely completion of milestones.

2.1 Project Description

Understanding exoplanet atmospheres requires analysing large and complex spectral data collected from telescopes. Classical machine learning methods often struggle with this high-dimensional data, making it challenging to extract meaningful patterns. **Quantum Machine Learning (QML)** offers a new approach by using quantum algorithms to process this data more efficiently and accurately.

This project aims to develop **QML-based models for exoplanet characterization**, focusing on three key objectives:

1. **Detecting atmospheric anomalies** that may indicate unique planetary conditions or potential habitability.
2. **Learning meaningful latent representations** of exoplanet transmission spectra to improve data interpretation.
3. **Benchmarking QML models** against classical machine learning techniques to evaluate their advantages and limitations



To achieve these, we will explore Quantum Neural Networks (QNNs) and variational quantum circuits for feature extraction, anomaly detection, and spectral reconstruction. These techniques will be tested on both simulated and real telescope data to assess their effectiveness.

The expected outcome is a set of QML models capable of accurately modeling exoplanet spectra, identifying anomalies, and improving data-driven exoplanet characterization. These advancements will enhance our ability to interpret observational data, refine atmospheric models, and support future space missions in their search for habitable worlds.

How Are Exoplanets Analysed?

Exoplanets are planets orbiting stars beyond our solar system. Their atmospheres are studied using transmission and emission spectroscopy, where light from the host star passes through or is emitted by the planet's atmosphere, leaving behind unique spectral signatures.

These spectra, captured by telescopes like Hubble, JWST, and ALMA, provide insights into chemical compositions, cloud structures, and atmospheric conditions, helping scientists assess exoplanet habitability and climate.

The most common methods include:

1. **Transit Method** – When a planet passes in front of its host star, telescopes observe a slight dip in the star's brightness, revealing details about the planet's size and orbit.
2. **Radial Velocity Method** – Measures tiny shifts in a star's light spectrum caused by a planet's gravitational pull, helping estimate its mass.

3. **Direct Imaging** – Some large exoplanets can be directly photographed using specialized instruments that block out the star's light.

2.2 Deliverables

1. Variational Quantum Circuit (VQC) for Feature Extraction: Develop and implement VQCs to identify latent representations of exoplanet transmission spectra, specifically capturing atmospheric features like chemical abundances and cloud structures.

2. Quantum Neural Network (QNN) for Classification – Train a QNN on extracted features to classify spectra and detect anomalous exoplanet atmospheres.

3. Benchmarking Against Classical Models – Evaluate the performance of QML models by comparing them with classical machine learning approaches.

4. Application to Real and Simulated Data – Test and validate the models on synthetic exoplanet spectra and real observational data from missions like JWST and Hubble.

5. Comprehensive Documentation and Usage Guide – Provide detailed documentation on model implementation, training procedures, and evaluation methods.

2.3 Implementation : Plan of action

Basic Goal Of Project :

The primary objective of this project is to develop a quantum-classical hybrid pipeline for exoplanet characterization, specifically focusing on classification and atmospheric characterization using Quantum Machine Learning (QML).

We will leverage Variational Quantum Circuits (VQCs) for feature extraction and use these latent representations to train a Quantum Neural Network (QNN) for classification and regression tasks. This approach integrates quantum and classical computing to enhance feature learning, classification, and parameter estimation from transmission spectra data.

Step 1: Spectral Simulation and Atmospheric Retrieval:

Before applying quantum processing, we will generate and analyse exoplanetary spectra using **state-of-the-art models**. These simulations provide well-controlled training data, allowing us to develop and benchmark quantum models before testing on real observations.

Datasets & Methods:

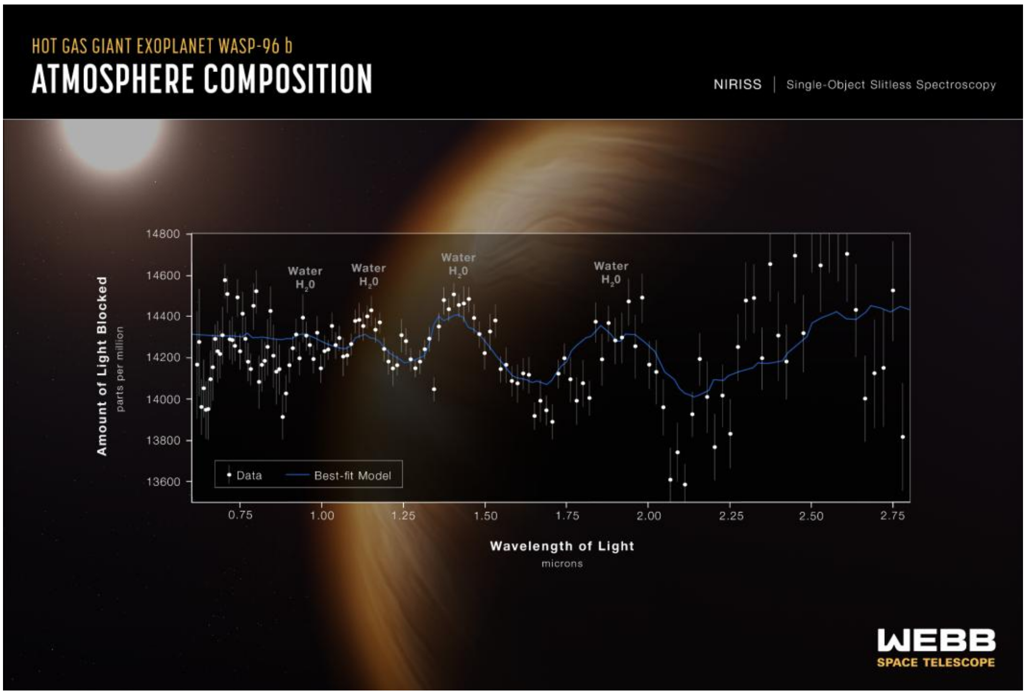
1. **petitRADTRANS (Radiative Transfer for Spectral Modeling)**
   - Simulates transmission and emission spectra of exoplanets using radiative transfer.
   - Models molecular absorption based on gas abundances, temperature-pressure profiles, and clouds.

2. **Poseidon (Bayesian Atmospheric Retrieval)**
   - Extracts atmospheric properties (gas abundances, temperature profiles, cloud structures) from spectra.
   - Performs Bayesian retrieval to provide atmospheric properties from spectra.
   - Uses nested sampling to compute posterior distributions, quantifying uncertainties.

Atmospheric Compositions to Simulate:

1. **Primary Molecules:** We will simulate key molecules like H_2O, CO_2, CH_4, CO, and NH_3 in varying abundances.
2. **Metal-Rich Atmospheres:** Simulations will include enhanced metallicities with trace elements like Na and K, and vary C/O ratios to capture spectral diversity.
3. **Cloud and Haze Models:** We will model both clear and cloudy/hazy atmospheres using Mie scattering with variable cloud opacity and altitude.
4. **Thermal Profiles:** Atmospheres will range from isothermal to non-isothermal with possible thermal inversions, spanning equilibrium temperatures from 300 K to over 2000 K.

5. **Planet Types:** Spectra will be simulated for hot Jupiters, sub-Neptunes, mini-Neptunes, and super-Earths, accounting for different planetary gravities and radii.



Ground Truth and Model Refinement:

1. The simulated spectra, generated using **petitRADTRANS or Poseidon**, are labeled with their respective atmospheric compositions based on known molecular absorption features.
2. These labeled spectra serve as the ground truth for training the QML model, ensuring it learns to associate spectral patterns with specific atmospheric compositions.
3. The labeled data also helps validate model predictions and refine quantum feature extraction for improved atmospheric characterization.

Why Use Simulated Data?

1. **Controlled Environment**: Simulated spectral data enables precise control over planetary conditions, unlike real telescope data, which is limited and noisy.
2. **Scalability**: We can generate as much data as needed for training, ensuring robust model development.

3. **Domain Adaptation**: Models trained on simulations can generalize to real exoplanet data (e.g., JWST spectra), improving performance in real-world applications.

Step 2: Quantum Feature Extraction with VQCs

1. Data Preprocessing and Encoding

To prepare the transmission spectra for quantum processing, we will implement the following steps:

1. **Normalization:** Standardize flux values and wavelengths to maintain numerical stability during encoding and ensure molecular absorption features are preserved during encoding.
    - **Spectral Interpolation**:
      All spectra will be interpolated to a fixed wavelength grid . Flux values will be normalized to remove scale differences across samples.
    - **Noise Injection**:
      We'll add realistic noise based on HST and JWST instrument profiles to the synthetic data, improving generalization to actual telescope observations.
2. **Dimensionality Reduction:** Principal Component Analysis (PCA) will primarily be used to retain key spectral features, particularly those influenced by atmospheric molecular absorption. It is computationally efficient but lacks the capability to capture complex quantum correlations.
3. **Quantum Autoencoders (QAEs)**, on the other hand, are suited for quantum data due to their ability to exploit quantum entanglement , offering a better compression of high-dimensional data.
4. **Spectral features** directly corresponding to molecular absorption patterns will be emphasized, ensuring critical atmospheric information is captured in quantum processing.

Encoding Strategy:

1. **Angle Encoding** is the preferred method for exoplanet spectra. It is well-suited because we map the normalized flux values to qubit rotation angles (RX, RY),

maintaining the precision required for spectral analysis.

2. **Amplitude Encoding:** An alternative approach, where entire spectra are encoded into quantum states to exploit **superposition advantages**—used if hardware constraints permit.

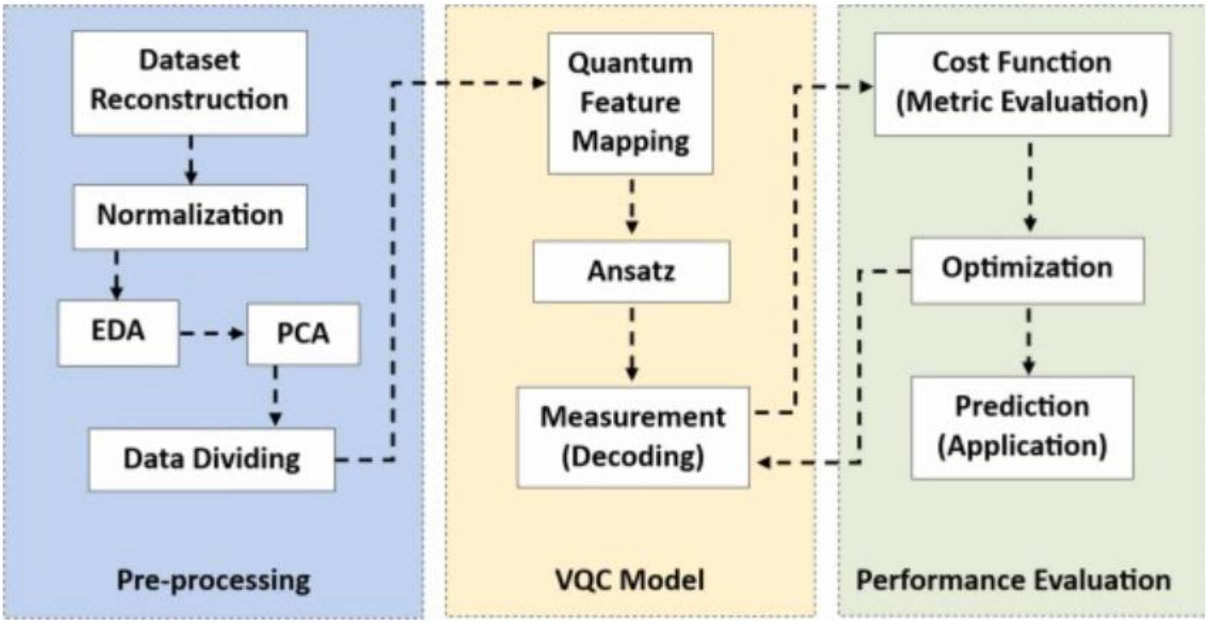
3. **Quantum Kernel Encoding:** We will explore optimized quantum kernels designed for time-series and spectral data to improve feature separability in quantum space.

   We will explore optimized quantum kernels tailored for spectral data, such as the Fidelity Kernel and Gaussian Quantum Kernel.

   These kernels will help improve feature separability by mapping spectral features into a high-dimensional Hilbert space, enhancing classification performance.

2. Variational Quantum Circuit (VQC) Design:

We will develop Parameterized Quantum Circuits (PQCs) for spectral feature extraction, transforming encoded spectral data into quantum embeddings.



1. Quantum Feature Mapping & Ansatz Selection

Encoding: We use techniques like Angle, Amplitude, and Quantum Kernel Encoding to map classical spectral data to quantum states.

Ansatz Selection:

- For quantum feature extraction, we will primarily use the **Quantum Self-Attention Ansatz**, as it effectively captures long-range dependencies in spectral data, ensuring better representation of molecular signatures in our model.

- Other ansatz options include the **Hardware-Efficient Ansatz** for efficient encoding, the **Alternating Layer Ansatz** for balancing expressivity and trainability, and the **Quantum Convolutional Ansatz** for hierarchical feature extraction.

We will experiment and choose the best Ansatz for feature extraction. The choice of ansatz will be optimized based on the complexity of spectral features, ensuring accurate representation of atmospheric signatures.

2. Circuit Optimization

To ensure efficient training and stability, quantum circuits will be optimized using:

- **Classical Optimizers**: COBYLA, SPSA, and Adam for parameter optimization.
- **Layer-wise Training**: Sequentially optimize layers to mitigate barren plateaus.
- **Noise-aware Initialization**: Use robust initial parameters to minimize noise impacts.
- **Circuit Depth Optimization**: Balance expressivity and reduce decoherence effects.

3. Post-Measurement Feature Refinement

Once quantum embeddings are extracted, we will refine them using:

- **Quantum Entanglement-Based Feature Selection**: Identifies significant qubits contributing to the feature representation.
- **Classical vs. Quantum Embedding Comparison**: Evaluates the information gain from quantum embeddings over classical methods like PCA.

Step 3: Quantum Neural Network (QNN) for Classification:

After extracting quantum features using **Variational Quantum Circuits (VQC)**, we will utilize these quantum embeddings to train a **Quantum Neural Network (QNN)** for exoplanet classification and anomaly detection tasks.

The QNN will operate within a **quantum-classical hybrid framework**, leveraging the quantum embeddings to enhance model accuracy and provide deeper insights into the atmospheric characterization of exoplanets.

QNN Architecture:

The QNN will consist of several layers that interact with the quantum embeddings, designed for classification and regression tasks. Here's how we will structure the architecture:

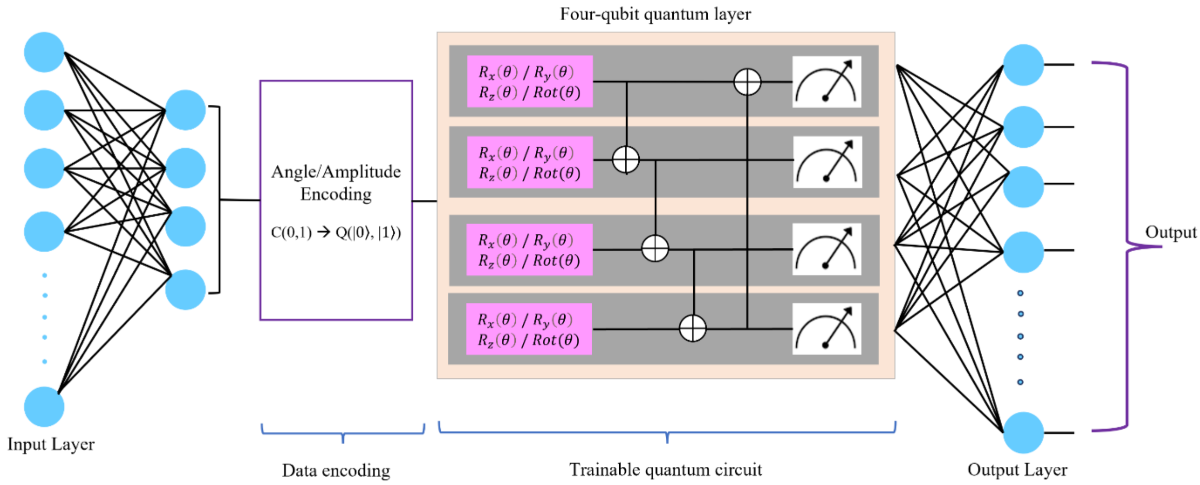
1. **Quantum Input Layer:** The quantum embeddings obtained from the Variational Quantum Circuit (VQC) step act as the input to the QNN. These embeddings encode critical features of exoplanetary atmospheres from transmission spectra, ensuring molecular absorption patterns are central to classification.

2. **Quantum Hidden Layers:** The network uses Parameterized Quantum Circuits (PQCs) with quantum gates (RX, RY, RZ, CNOT) to model complex feature interactions. The Pauli-Z measurements introduce non-linearity, improving the learning process.

3. **Hybrid Quantum-Classical Interaction:** This architecture leverages the strengths of both quantum and classical systems. Quantum layers extract high-dimensional features using superposition and entanglement, while classical layers refine these features, ensuring a hybrid approach for optimal predictions.

4. **Quantum Output Layer:** The final decision is determined by measurements in the Pauli-Z basis, enabling the network to provide either binary or multi-class classification based on the learned quantum embeddings.

5. **Multi-class Classification:** The QNN will be trained to associate specific quantum embeddings with known atmospheric compositions, ensuring accurate characterization of exoplanetary atmospheres.



Training the QNN:

1. Variational Quantum Training:

The QNN will utilize Variational Quantum Circuits (VQCs), where quantum parameters adapt during training to extract complex features.

2. Loss Function:

- Classification: Cross-entropy loss (e.g., classifying exoplanet types).
- Regression: Mean Squared Error (MSE) for predicting atmospheric properties.

3. Backpropagation & Optimization:

- **Parameter Shift Rule:** Used for efficient quantum parameter updates.
- **Hybrid Optimization:** Combines VQE (quantum) with classical optimizers like COBYLA and Adam for fine-tuning or employ Quantum Natural Gradient Descent and layer wise training.
- **Use Transfer Learning** by integrating pre-trained classical models into hybrid architectures.
- **Quantum Regularization:** Quantum Dropout and Gradient Clipping will stabilize training and prevent overfitting.

4. Hybrid Quantum-Classical Training:

- **Quantum Layers**: Quantum layers will extract complex, high-dimensional features from the input data (spectral embeddings) by leveraging quantum superposition and entanglement.
- **Classical Layers**: Classical layers (e.g., fully connected layers) will refine the predictions by incorporating the quantum-extracted features into the final

output.

5. Regression for Atmospheric Parameter Estimation

- **Prediction of Molecular Abundances**: The QNN will predict molecular abundances, such as **H_2O**, **CO_2**, and **CH_4**, from the spectral embeddings, using the learned quantum features.
- **Quantum Self-Attention**: To improve time-series learning, **Quantum Self-Attention** will be applied, allowing the model to focus on relevant spectral features and improve the handling of sequential data.
- **Quantum-Optimized Loss Function**: A quantum-optimized loss function will be used to ensure better convergence during training, further enhancing the model's ability to estimate atmospheric parameters with high accuracy.

Step 4: Hybrid Integration with Classical Processing:

1. To address quantum device limitations like qubit count and coherence time, classical preprocessing and postprocessing steps are integrated to enhance performance:

2. **Preprocessing:** Before quantum encoding, redundancy in the classical data is reduced using domain-specific feature selection. Classical CNN feature extractors may enhance spectral features before quantum transformation, focusing on molecular absorption features like H_2O, CO_2, CH_4, and NH_3.

3. **Postprocessing:** After quantum processing, quantum outputs are aggregated for improved accuracy. Hybrid integration with classical algorithms will refine predictions, ensuring robust atmospheric characterization.

   Furthermore, optimizing quantum embeddings before the final classification or regression aligns with the efforts made in refining quantum features from the VQC.

Step 5: Anomaly Detection and Experimental Validation

For robust classification, we apply anomaly detection to detect unusual spectral patterns that may indicate instrument noise or new astrophysical phenomena.

Chosen Method: Quantum Autoencoders (QAEs)

We will train **Quantum Autoencoders (QAEs)** to learn normal spectral patterns and identify anomalies as deviations from expected reconstructions. QAEs are well-suited for exoplanet data because they effectively compress and reconstruct high-dimensional spectra, leverage quantum entanglement for richer feature representations, and use reconstruction errors to identify anomalies.

Alternative Methods:

- **Hybrid One-Class SVM :** Uses quantum kernel methods on extracted spectral embeddings.
- **Isolation Forests on Quantum Features:** Classical Isolation Forests applied to quantum embeddings.

Implementation:

- **Quantum Embeddings**: Extract quantum embeddings from previous steps to serve as input for anomaly detection.

- **Anomaly Identification**: The QNN classifier processes these embeddings to differentiate between valid spectral data and anomalous patterns.

- **Refinement and Validation:** Flagged anomalies are further refined using hybrid classical processing, ensuring accurate identification and eliminating false positives before final validation.

Step 6: Benchmarking and Testing

To validate the hybrid quantum model, we will:

1. **Model Comparison**: Compare the accuracy of the hybrid model against classical ML baselines (e.g., CNNs, SVMs, Random Forests). Use metrics like accuracy, precision, recall, F1 score (for classification), or MSE, R-squared (for regression).

2. **Quantum Resource Analysis**: Evaluate quantum resource requirements such as qubit count, circuit depth, and noise impact.

3. **Experimental Testing**: Run experiments on quantum simulators (e.g., Qiskit, Cirq) and real quantum hardware (e.g., IBM Quantum, Rigetti) to assess performance.

4. **Real-World Validation**: Test the model on real astrophysical datasets (e.g., JWST, Hubble) to validate its performance in realistic conditions.

5. **Scalability & Efficiency**: Measure the training time and scalability of the hybrid model as data size increases. Analyse computational efficiency and resource usage.

6. **Performance Visualization**: To evaluate the effectiveness of our quantum-classical hybrid model, we use targeted visualizations that analyze spectral variations, model behavior, and retrieval performance.

    - Scatter plots comparing predicted vs. true molecular abundances (e.g., H_2O, CO_2) with R^2 scores assess prediction accuracy. Uncertainty is shown using error bars.
    - Visualize how changes in temperature, C/O ratio, and cloudiness affect spectra, highlighting which features are most sensitive to atmospheric parameters.
    - Visualize different exoplanet classes (e.g., hot Jupiters vs. super-Earths) to show spectral diversity.
    - Show training/validation curves to monitor learning. Confusion matrices and label scatter plots validate class-level performance.

Additional Considerations:

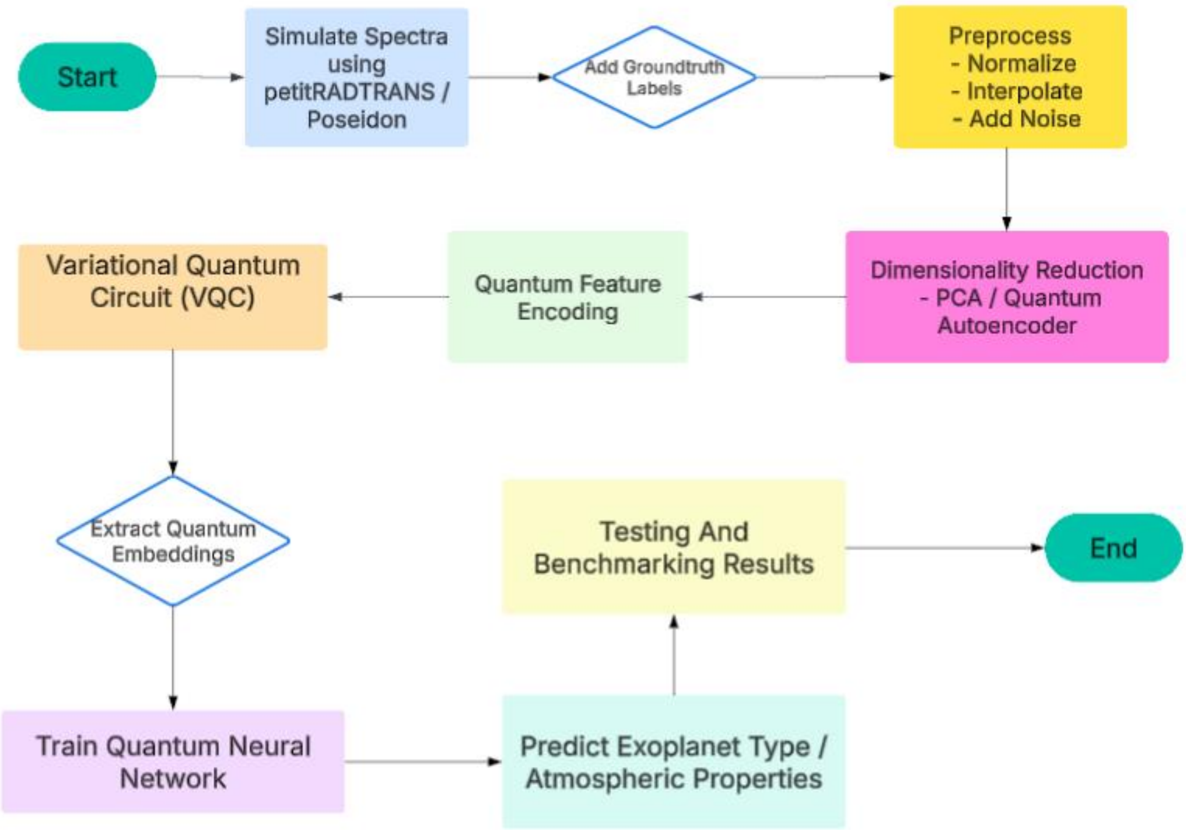
1. **Model Comparisons:** Compare quantum-classical architectures, such as pure QNN models versus hybrid QNN-classical models, to highlight where quantum advantages (e.g., feature extraction) emerge in complex tasks.

2. **Ablation Study:** Evaluate the impact of different quantum feature encodings (Angle vs. Amplitude vs. Kernel Encoding) on model performance to identify the most effective encoding strategy.

3. **Resource Estimation & Computational Cost Analysis:**
    - **Challenges & Solutions:** Optimize circuit depth to reduce noise; use PCA and quantum autoencoders for dimensionality reduction; offload preprocessing to GPUs.
    - **Computational Cost:** Classical ML models train in minutes to hours, while QNNs require hours to days on quantum hardware. Benchmarking
    - will compare execution time and accuracy to assess feasibility for real JWST spectral data.
    - **Quantum Hardware:** IBM Quantum is preferred for simulations due to its strong PennyLane integration and reliable superconducting qubits. IonQ ,Rigetti and Borealis are alternatives for experimental validation.

- **Classical Hardware:** Google Colab Pro for initial experiments; AWS EC2 (T4, A100 GPUs) for large-scale training and preprocessing.

Flowchart Diagram of Overall Implementation Plan :



Quantum Decoherence Mitigation & Error Handling

Quantum machine learning faces significant challenges due to noise and errors in near-term quantum devices. Decoherence, gate errors, and barren plateaus can severely impact the accuracy and stability of computations. To address these issues, various error mitigation and correction techniques can be applied to enhance the reliability of quantum models. Below are some potential strategies to handle these challenges effectively.

1. Quantum Error Correction : QEC protects quantum information by encoding one **logical qubit** into many **physical qubits**. This allows the system to detect and fix small errors without measuring the actual quantum state .

Key Steps in QEC:

- **Encoding**: Spread one qubit's information across multiple physical qubits.
- **Error Detection**: Identify where an error happened without looking at the actual qubit values.
- **Error Correction**: Apply a fix to bring the logical qubit back to the correct state.
- **Common codes**: Shor code, 5-qubit code, Surface code.

2. Mitigating in VQC Optimization:

- Use **layer-wise training** to avoid vanishing gradients.
- **Quantum Natural Gradient (QNG)**: Optimizes using the quantum information metric to improve convergence.

3. Other Approaches

- Use states immune to certain noise.
- Minimize time qubits are exposed to noise.

2.4 Project Timelines

Phase	Activities	Estimated Hours(175 hrs)
Community Bonding Period	- Familiarize myself with project documentation and requirements. - Set up development environment (install Qiskit, TensorFlow Quantum, etc.). - Study exoplanet datasets (light curves, protoplanetary disk images, etc.). - Plan initial steps for project execution, including model architecture and data preprocessing.	**N/A**

Weeks 1 & 2 (Initial Setup, Data Simulation & Preprocessing)	- Generate simulated exoplanet datasets for testing. - Study and clean datasets (light curves, imaging data, etc.). - Set up the framework for quantum models. - Plan and design model architecture. - Engage with mentors for feedback on data handling and initial steps.	**35 hrs**
Weeks 3 & 4 (Model Development and Optimization)	- Begin implementing quantum models for exoplanet characterization i.e to analyse atmospheric signatures of exoplanets. - Work on feature extraction techniques for better classification. - Work on data augmentation techniques to enhance model robustness. - Begin optimization of the models using quantum feature maps and quantum circuit techniques. - Compare different quantum classifiers for efficiency.	**40 hrs**
Week 5 (Buffer Period)	- Buffer time to handle unexpected delays or challenges. - Refine models based on mentor feedback. - Test and compare the performance of QML models on different datasets. - Make any necessary adjustments to model architectures and fine-tune hyperparameters.	**15 hrs**
Midterm Evaluation Period	- Submit Midterm Evaluation. - Review progress with mentors. - Adjust project approach based on feedback. - Prepare for the second phase of coding, with a focus on optimization and scaling.	**10 hrs**

Weeks 6 - 7 (Model Optimization and Automation)	- Optimize models for scalability and performance. - Work on automating the pipeline for running experiments on new exoplanet datasets. - Integrate quantum models with classical machine learning approaches to leverage both parts effectively.	**35 hrs**
Weeks 8 - 9 (Anomaly Detection Implementation & Testing)	- Implement QML-based anomaly detection for rare exoplanet signals. - Compare quantum anomaly detection with classical ML methods. - Optimize the model for real-world applicability and robustness.	**20 hrs**
Final Week (Preparation and Submission)	- Prepare final project deliverables, including documentation, code comments, and blog posts. - Conduct final testing, resolve any remaining bugs, and finalize the project. - Submit the final work product and mentor evaluations. - Engage with mentors for final feedback and wrap-up.	**20 hrs**
Post-GSoC Phase	- Continue contributing to the project, as needed. - Work on further improvements to models, adding more advanced features . - Engage with the community by assisting with ongoing discussions and issues.	**N/A**

Efforts to Further My Knowledge

To ensure I am well-prepared for this project, I have actively worked on strengthening my understanding of Quantum Machine Learning (QML) and its applications. My efforts include:

- **Going Through PennyLane Documentation and Tutorials:** I have explored PennyLane's official documentation and tutorials to build a strong foundational understanding of quantum circuits, parameterized quantum gates, and their applications in quantum machine learning.
- **Implementing PennyLane Quantum Circuits:** I have designed and tested multiple quantum circuits using PennyLane, focusing on parameterized quantum circuits and their applications in machine learning. My implementations include Single Qubit and Multi-Qubit Circuits , Quantum Encoding Techniques, Quantum Neural Networks and Parameterized Quantum Gates.
- **Quantum Neural Networks (QNN) on MNIST:** I have successfully trained a QNN for MNIST classification using Pennylane, understanding how quantum layers can be integrated into deep learning models.
- **EXXA Test Evaluation Tasks:** As part of the ML4Sci application, I have completed the General and Sequential Tests. These involved:
    - **General Test:** Implementing unsupervised clustering on protoplanetary disk images from ALMA data.
    - **Sequential Test:** Simulating exoplanet transit light curves and training a classifier to detect exoplanets using machine learning.

These experiences have solidified my foundational understanding of QML, ensuring I am equipped to tackle this project effectively.

Task Folder: https://drive.google.com/drive/folders/19lCarTa5K-4zcD8V-KTiRMmi4uOfGIXh

All implementations are available at:
https://drive.google.com/drive/folders/1QUzm5KM-PqpzE43OZyxzRJGChMYZFpzE

Other Comments

My long standing fascination with space and astrophysics, combined with my background in machine learning, led me to this project. The opportunity to apply

Quantum Machine Learning (QML) to exoplanet research aligns with my keen interest in advancing computational methods for scientific discovery.

Through this project, I aim to deepen my understanding of QML and astronomy as a whole, and I look forward to continuing contributions to the project even after GSoC 2025, improving models and advancing exoplanet research.

Benefits to the Community

Machine learning (ML) is already widely used in exoplanet research for tasks such as detecting and classifying exoplanets from telescope data. However, quantum machine learning (QML) is a relatively new approach that has the potential to revolutionize this field by leveraging quantum computing's ability to process high-dimensional data efficiently.

This project will contribute to the advancement of QML in exoplanet characterization by:

- Developing **quantum-enhanced models** to analyse exoplanetary spectral data, helping identify chemical compositions, atmospheric processes, and potential habitability.
- **Paving the way for future research** by benchmarking QML models against classical ML counterparts, assessing where quantum methods provide an advantage.
- **Contributing to open science** by providing well-documented QML architectures and datasets, enabling other researchers to build upon this work.

References

❖ Approaches and Potential Datasets:
https://medium.com/@shuklag554/exoplanet-atmosphere-characterization-gsoc24-ml4sci-5f78f85faa13
https://petitradtrans.readthedocs.io/en/latest/
https://petitradtrans.readthedocs.io/en/latest/content/notebooks/getting_started.html
https://poseidon-retrievals.readthedocs.io/en/latest/

- QML:
  https://arxiv.org/pdf/2502.01146
  https://qml-tutorial.github.io/chapter1/
  https://arxiv.org/html/2409.00294v1

- VQC:
  https://arxiv.org/html/2312.13798v1/#S5
  https://neuralsorcerer.medium.com/building-a-variational-quantum-classifier-from-scratch-a-step-by-step-guide-c8adcbf13418
  https://docs.pennylane.ai/en/stable/introduction/templates.html

- QNNs:
  https://medium.com/mit-6-s089-intro-to-quantum-computing/quantum-neural-networks-7b5bc469d984
  https://arxiv.org/pdf/1802.06002
  https://qml-tutorial.github.io/chapter4/

- Pennylane:
  https://docs.pennylane.ai/en/stable/development/guide.html
  https://qml-tutorial.github.io/code/
  https://docs.pennylane.ai/en/stable/code/api/pennylane.qnn.KerasLayer.html

- Quantum Error and Noise:
  https://q-ctrl.com/topics/what-is-quantum-error-correction
  https://en.wikipedia.org/wiki/Quantum_error_correction