Open Task

Quantum Computing

Quantum computing is an emerging paradigm that fundamentally departs from classical computation by leveraging superposition, entanglement, and quantum interference to solve problems more efficiently than classical computers. The fundamental difference lies in the way information is processed: classical computers use bits, which exist in a definite state of either 0 or 1, whereas quantum computers use qubits, which can exist in a superposition of both 0 and 1 simultaneously. This enables quantum parallelism, allowing certain computations to be performed exponentially faster. Entanglement creates strong correlations between qubits, enabling instant information sharing regardless of distance, a property that has no classical counterpart. Quantum interference plays a key role in optimizing computational outcomes by amplifying correct computational paths while canceling out incorrect ones.

But quantum computers are still in their infancy, with Noisy Intermediate-Scale Quantum (NISQ) machines being the state of the art right now. These computers possess a limited number of qubits and are prone to decoherence and noise, leading to computational errors. *Quantum error correction (QEC)* and *quantum error mitigation (QEM)* are being researched to make quantum computations more reliable. Also, hybrid quantum-classical approaches are being investigated to leverage the strengths of classical computation together with quantum speedups and make them more usable for near-term applications.

Quantum computing holds vast promise in optimization, cryptography, drug discovery, finance, and materials science. Quantum algorithms like Shor's algorithm for integer factorization, Grover's search algorithm, and Variational Quantum Eigensolver (VQE) for chemistry simulations show the strength of quantum systems. Nevertheless, executing these algorithms on existing quantum hardware needs to take into account circuit depth, gate fidelity, and noise reduction methods to obtain meaningful results.

Quantum Machine Learning

Quantum Machine Learning (QML) is an emerging intersection of quantum computing and machine learning that seeks to leverage quantum mechanics for computational benefit in classification, pattern recognition, optimization, and generative modeling.

The key strength of QML lies in processing data in exponentially huge Hilbert spaces without calling for a proportionate rise in physical memory. This is especially critical in quantum kernel approaches where the classical methods lack performance in dealing with high-dimensional feature spaces. Quantum-enhanced support vector machines (QSVMs) and quantum kernels embed data into complex quantum feature spaces, allowing for more efficient classification and clustering. This is of enormous potential in areas like genomics, material discovery, finance, and drug screening, where pattern extraction from large-scale, high-dimensional data is key.

One of the basic building blocks of QML is Variational Quantum Circuits (VQCs), which are trainable quantum neural networks. These circuits utilize parameterized quantum gates, with the optimization carried out via classical gradient-based techniques. VQCs feature prominently in Quantum Generative Adversarial Networks (QGANs) and Quantum Boltzmann Machines (QBMs), which are currently under consideration for applications such as data generation and quantum-fortified generative modeling.

However, QML is in its infancy and has a lot of challenges to overcome. The most urgent one is the **barren plateau problem**, where gradients exponentially disappear as circuits increase in depth, complicating training. The other significant limitation is **hardware noise and decoherence**, which causes errors in quantum computations. Hybrid quantum-classical models, in which quantum circuits are applied for feature transformation but classical neural networks are utilized for end classification, offer a realistic means of avoiding these challenges.

My Research in QML

As a part of my research work in QML, I have investigated several different methods of incorporating quantum computing into actual machine-learning problems. One of the significant contributions has been the creation of Quantum-Enhanced Spiking Neural Networks (QSNNs) with Temporal Encoding, in which spiking neuron models were integrated into quantum circuits to encode dynamic temporal dependencies. This work was validated on fraud detection, network security, and signal datasets with the result being that quantum spiking models may match if not perform better than classical deep learning architectures.

Another work of mine focussed on Quantum Graph Neural Networks (QGNNs). Although conventional Graph Neural Networks (GNNs) function optimally when working with graph-structured data, they suffer from feature space limitations and do not scale up. During my summer internship at TCS, I explored how quantum-upgraded message-passing algorithms could be employed to learn higher-order dependencies in graph structures. Encoding node and edge features into quantum states, the QGNN and QGRNN (Quantum Graph Recurrent Neural

Network) models enhanced accuracy for molecular property prediction and financial transaction fraud detection tasks.

In addition, I have researched Quantum Convolutional Neural Networks (QCNNs) for medical image classification, where high-dimensional imaging data were encoded into quantum circuits using quantum feature maps. Surprisingly, this maintained classification accuracy even when decreasing the number of available features, which emphasizes the performance capability of QCNN even in feature constrained environment.

Conclusion

Though such models have shown great promise, there are still challenges. Variational circuit training is still non-trivial, and the applicability of QML models depends frequently on the limited availability of qubits and noisy intermediate-scale quantum (NISQ) hardware. Upcoming studies need to address hardware-efficient quantum embeddings, training methods resilient to noise, and improved interfacing of quantum models with deep learning networks.

Quantum Algorithm: Warm-Start Quantum Approximate Optimization Algorithm (WS-QAOA)

The Warm-Start Quantum Approximate Optimization Algorithm (WS-QAOA) is a variant of the regular QAOA, to boost the efficiency of combinatorial optimization problem-solving using classical warm-starting methods. Contrary to regular QAOA, which starts with an uninformed superposition state, WS-QAOA exploits previously acquired classical knowledge via approximation methods like linear programming (LP) relaxations to start the quantum state more informed. This method considerably enhances convergence rate, decreases circuit depth, and enhances final solution quality.

Standard QAOA and Its Derivation

QAOA is a variational quantum algorithm used to discover approximate solutions of combinatorial optimization problems, frequently written as:

$\min_{x \in \{0,1\}^n} C(x)$

where C(x) is the classical cost function representing the optimization problem, such as Max-Cut, Traveling Salesman, or Portfolio Optimization.

The QAOA ansatz consists of alternating layers of two unitary operators:

1. **Phase Operator**, which encodes the cost function into the quantum state:
   $U_C(\gamma)=e^{-i\gamma C}$
2. **Mixing Operator**, which encourages exploration of the solution space:
   $U_B(\beta)=e^{-i\beta B}$ where $B=\sum_i X_i$ is the mixing Hamiltonian, and X_i is the Pauli-X operator acting on qubit i.

The quantum state is initialized in a uniform superposition:

$|\psi 0\rangle=H^{\otimes n}|0\rangle^{\otimes n}$

where H is the Hadamard gate applied to all qubits. The final QAOA state is given by:

$|\psi(\gamma,\beta)\rangle=U_B(\beta_p)U_C(\gamma_p)\ldots U_B(\beta_1)U_C(\gamma_1)|\psi_0\rangle$

The expectation value of the cost function is then minimized using classical optimization:

$\min_{\gamma,\beta}\langle\psi(\gamma,\beta)|C|\psi(\gamma,\beta)\rangle$

One of the major limitations of traditional QAOA is that it begins with an uninformed uniform superposition state, which makes no use of any prior classical information. In WS-QAOA, a warm-start classical solution is added to the quantum initialization, enhancing efficiency and performance.

Rather than employing the uniform superposition state, WS-QAOA initializes the quantum state with a rounded classical solution, often obtained from a classical LP relaxation. The quantum state is initialized as:
$|\psi_{ws}\rangle=^{i=1}\otimes_n(\cos(\theta_i)|0\rangle+\sin(\theta_i)|1\rangle)$

where θ_i is computed based on the probability of variable i taking value 1 in the classical approximation:

$\theta_i=\arcsin(p_i)$

where p_i represents the probability obtained from the classical solution. This initialization ensures that the quantum state is biased toward the most probable classical configurations, leading to a more efficient exploration of the solution space.

Evolution Using WS-QAOA Operators

After the warm-start initialization, WS-QAOA follows a similar alternation of unitary operators as in standard QAOA. However, a key difference is in the **modified mixing operator**, which preserves prior classical correlations:

$U_B(\beta)=e^{-i\beta B_{ws}}$

where the **warm-start mixing Hamiltonian** is defined as:

$B_{ws}=_i\sum(1-2p_i)X_i$

This modification ensures that the evolution remains close to the classical approximation while still allowing quantum exploration.

Cost Function Evaluation and Optimization

Once the circuit is executed, the expectation value of the cost function is measured:

$F(\gamma,\beta)=\langle\psi_{ws}|U_B^\dagger U_C^\dagger C U_C U_B|\psi_{ws}\rangle$

Classical optimizers are then used to update the variational parameters γ,β to minimize the function $F(\gamma,\beta)$.

Advantages of WS-QAOA Over Traditional QAOA

WS-QAOA has several important advantages over traditional QAOA:

Improved Convergence: As the initialization is closer to the optimal solution, WS-QAOA needs fewer optimization steps.
Improved Solution Quality: With the use of classical information, WS-QAOA obtains better approximations with fewer iterations.
Adaptive Balance Between Classical and Quantum: It cleverly combines classical pre-optimization with variational quantum methods, the best of both worlds.

My work

As a Runner-Up in the TCS Quantum Challenge, I worked on optimizing airline fleet allocation using quantum computing techniques, focusing on the Warm-Start Quantum Approximate Optimization Algorithm (WS-QAOA) to enhance classical optimization approaches. The problem involved assigning aircraft to flight routes while minimizing operational costs, maximizing efficiency, and respecting constraints—a combinatorial optimization challenge that scales exponentially with fleet size. By leveraging WS-QAOA, I integrated classical pre-optimized solutions to initialize quantum states, reducing circuit depth and improving convergence speed. This approach significantly outperformed standard QAOA by achieving faster convergence and higher-quality solutions while remaining practical for near-term Noisy Intermediate-Scale Quantum (NISQ) devices.

Quantum Software: Qiskit, Pennylane

To efficiently implement and run quantum algorithms, stable quantum software frameworks like Qiskit and Pennylane are critical. The two frameworks provide specialized capabilities applicable to a range of use cases, from the development of quantum algorithms to quantum machine learning (QML) and hybrid quantum-classical computing.

Qiskit, built by IBM, is a versatile open-source framework developed mainly for gate-based quantum computing. It is a complete-stack quantum development environment with tools for the design of quantum circuits, noise-aware simulation, and running on IBM's quantum processors. Qiskit Aer is one of its main features, as it is a high-performance simulator that allows quantum algorithms to be tested and benchmarked before execution on hardware. Qiskit features Qiskit Machine Learning, a module that combines quantum kernels and variational quantum circuits (VQCs) for hybrid learning applications. Because of its native hardware compatibility and robust community support, Qiskit is extensively utilized in research and industry for quantum optimization, quantum chemistry simulations, and machine learning applications.

Pennylane, however, is designed for quantum machine learning (QML) and hybrid quantum-classical computing. It was developed by Xanadu and has support for various quantum backends such as Qiskit, Cirq, and photonic quantum hardware. Pennylane's differentiability engine allows for gradient-based optimization of quantum circuits, making it an ideal choice for training variational quantum circuits (VQCs), quantum neural networks (QNNs), and quantum generative models. A standout feature of Pennylane is its seamless integration with PyTorch and TensorFlow, enabling direct implementation of quantum-classical architectures without extensive modifications.

Regarding my experience with these frameworks, I have been actively developing with Qiskit for more than two years, using it in an array of projects. A majority of my quantum algorithmic development and implementations, such as my research on Quantum-Enhanced Spiking Neural Networks (QSNNs), Warm-Start QAOA, and Quantum Convolutional Neural Networks (QCNNs), have been written using Qiskit.

Yet, I worked on Quantum Graph Neural Networks (QGNNs) with Pennylane specifically as part of my TCS Summer Internship. The project concerned using Pennylane's hybrid QML features and automatic differentiation to design quantum graph-based models and see how quantum embeddings along with message-passing approaches may improve graph representation learning. Working on this allowed me to have practical experience working with Pennylane, specifically in the quantum-enhanced graph networks and the quantum variational models.

Methods I think I am good at and would like to work on:

My research has primarily been focused on quantum-enhanced neural networks, quantum graph-based architectures, and hybrid quantum-classical learning frameworks applied to diverse machine learning tasks. I have worked extensively on Spiking Quantum Neural Networks (SQNNs), where I developed encoding schemes to integrate temporal dependencies into quantum circuits. I designed and compared two different SQNN models against classical LSTMs for time-series forecasting, fraud detection, and network security. Additionally, I constructed a fully quantum circuit leveraging the Quantum Fourier Transform (QFT) for efficient encoding, achieving function modeling accuracy comparable to classical optimizers. This work has been published in the **IEEE International Conference on Quantum Computing and Engineering (QCE24)** and **Quantum Techniques in Machine Learning (QTML)** under the titles *Quantum-enhanced spiking neural networks* and *Quantum-enhanced spiking neural networks with temporal encoding*, respectively. In quantum graph-based learning, I implemented Quantum Graph Neural Networks (QGNNs) and Quantum Graph Recurrent Neural Networks (QGRNNs) for molecular property prediction. I explored quantum phase kickback, Rabi oscillations, and entanglement fidelity in multi-qubit systems, demonstrating the potential of quantum graphs in chemistry and material science.

For quantum convolutional architectures, I developed a Quantum Convolutional Neural Network (QCNN) for medical image classification, showcasing its ability to maintain high accuracy even when the number of available features was significantly reduced. This work has been published in the **IEEE International Conference on Trends in Quantum Computing and Emerging Business Technologies (TQCEBT) 2024** under the title *Quantum Convolutional Neural Network for Medical Image Classification: A Hybrid Model*. Additionally, my experience in Quantum Approximate Optimization Algorithm (WS-QAOA) for airline fleet allocation and quantum-enhanced combinatorial optimization has strengthened my expertise in developing scalable and hardware-aware quantum machine learning models.

Moving forward, I am mostly interested in projects involving Quantum Kolmogorov-Arnold Networks (QKANs), Quantum Transformers (QViT, QTF), Quantum Representation Learning, and Quantum Graph Neural Networks (QGNNs) for High-Energy Physics (HEP). These models offer promising directions for solving challenging optimization, large-scale function approximation, and structured learning in quantum machine learning. Quantum Kolmogorov-Arnold Networks (QKANs) provide a quantum-boosted efficient method for universal function approximation, and they are suitable for high-dimensional modeling with fewer parameters than traditional networks. Quantum Transformers (QViT, QTF) open up new possibilities in long-range dependencies and sequence modeling, which may transform fields like NLP, genomics, and sequential decision-making. Quantum Representation Learning is another research area I would like to pursue, especially under contrastive learning, metric learning, and quantum embeddings, as these techniques would potentially provide an influential framework for extracting features from quantum and classical data. Also, Quantum Graph Neural Networks (QGNNs) for HEP are particularly interesting for LHC jet classification, rare event detection, and particle tracking, taking advantage of quantum strengths in processing structured data with high-energy physics use cases.