

Quantum Resonators Circuit for EEG Signals Pattern Matching

Hillol Biswas

M Manoj Kumar

S K Srivastava

WAPCOS Limited, Power Division, Gurgaon, India

Abstract:

Applications of quantum computing are expanding into new and crossing fields, yet there is a noticeable difference in how quantum machine learning visualization is approached, i.e., when compared with simulation conducted using any simulator. When working with a large dataset, this simulation results in a situation where, depending on the problem, either a probability or a count histogram can be visualized in a Qiskit circuit application. When dealing with large amounts of data, the quantum machine learning technique is typically used, which solves the classification problem by the selection of a suitable classifier; however, the usual simulation results plot visualization takes a backseat. Given this, a Qiskit quantum circuit-based method for pattern resonance reveals target quantum states for real-world applications where a baseline state of the input data is suitably encoded for seeing any target pattern. Since EEG data is spontaneous in the real world, in this work, this approach is further applied to mimic stream data to observe the state pattern and for comparison purposes. The results are consistent and encouraging for further application.

Keywords: Quantum Circuit, EEG signals, patterns, Qiskit

Introduction:

Quantum computing requires a series of numbers, a guideline to generate numbers, and an algebra with a universal set of operators to work on numbers and digits. The unit of quantum information in quantum computing is called a quantum bit, or qubit. A two-state quantum system represents the qubit, and further expansion to qudit for n-dimensional states. One can employ a variety of two-state quantum systems as multiple qubits. The application of particular quantum computing hardware is based on quantum devices. For instance, by designating the $+1/2$ spin state as basis state $|0\rangle$ and the $-1/2$ spin state as basis state $|1\rangle$, the states of a spin $1/2$ particle can be utilised as qubit basis states. By designating the vertical polarisation as basis state $|1\rangle$ and the horizontal polarisation as basis state $|0\rangle$, the photon polarisation can also be utilised as a qubit. Atomic or quantum dot energy levels can be utilised as qubit states. The basic state $|0\rangle$ is associated with the presence of an electron at energy level E_0 . The basic state $|1\rangle$ is related to the presence of an electron at energy level E_1 . There are currently plans to employ superconducting rings as qubits as well. Operator actions that cause state vectors in the Hilbert space to rotate are known as quantum computations. While this is not possible to visualize them, there have been several attempts to use a model to depict quantum calculations.

The gate model, often known as the circuit model of quantum computing, is the most popular and is being employed nearly entirely. A circuit can represent any quantum processing, no matter how complicated. Quantum circuits comprise qubits, quantum registers, and quantum gates and are therefore the basic ingredients used to describe quantum calculations. Quantum circuits use sequential operations of quantum gates on qubits rather than information flowing from gate to gate and quantum registers, which store information. A two-qubit quantum register's state vector has a length of one and is located in a four-dimensional Hilbert space. Since it is impossible to perform quantum computations with several qubits and computation time steps by hand, specialised software tools have been created to perform quantum computations on

conventional computers. These have been referred to these instruments as quantum simulators. However, the number of qubits increases exponentially with the processing time that conventional computers need.

Wavefunctions describe the states of qubits and quantum registers. A quantum computation solves the Schrödinger equation subject to the initial conditions imposed by initial qubit states. The input for quantum computation is the initial states. The output, or result, of quantum computations is the final states. The beginning and final states are driven by the unitary operator U . All of the quantum gates are part of the unitary operator. It is necessary to identify the initial qubit states and the unitary operator, or quantum circuit, for the particular problem to create a novel quantum computing (quantum algorithm).

Richard Feynman [1] envisioned a quantum computer that could imitate quantum physics by using the equations of quantum mechanics in his well-known 1982 lecture. This is regarded as one of the earliest ideas of quantum computing. He proposed that, as nature is not classical, a computer system based on quantum mechanical principles would be required to replicate natural events. Quantum computers present these opportunities, allowing computers to take advantage of quantum mechanical features like superposition and entanglement to provide the enormous computational power required for complicated quantum system simulations. Because the proposed quantum mechanical properties are only observed at the most fundamental scale of nature (such as electron spins or photon polarisation), which are extremely difficult to manipulate due to technological limitations, the initial progress towards developing quantum computer hardware was relatively slow. Nonetheless, the science of quantum computing has advanced quickly in recent years and has become one of the hottest debated scientific topics. Due to the potential for quantum computing to provide processing powers that will outstrip those of current supercomputers, industry and academia are interested in creating the first quantum machine in history. Countries, research centres, universities, and businesses are developing quantum computing and quantum technology systems. The race to develop the first large-scale universal quantum computer is now being vigorously pursued by several giant corporations, including IBM, Google, Microsoft, and Intel, as well as numerous ambitious start-ups, including Rigetti and IonQ. The development of quantum software and algorithms has advanced significantly in recent years, in tandem with the development of quantum hardware.

Unsurprisingly, currently it has applications in many domains, viz. cryptography, communication, drug simulation and discovery, traffic optimization, climate change, and machine learning. In particle physics anomaly detection [2], adversarial machine learning benchmarking [3], framework equivariant quantum neural networks [4], addressing machine learning generalization issues when training data are only a few [5], quantum machine learning constructive framework that captures all standard [6] models based on parametrized quantum circuits that of linear quantum models, real world application of quantum machine learning techniques [7] are some of the recent related directions research are being undertaken.

Qiskit Quantum Computing Applications:

In 2017, IBM launched Qiskit as an open-source quantum computing toolset. The Qiskit ecosystem is flourishing over ten years after it was first released. The field of quantum computing is developing quickly, and reliable software tools like Qiskit are becoming increasingly crucial for supporting research, teaching, and solving computationally challenging issues on quantum computers. A quantum algorithm outlines a computational problem and how quantum circuits can solve it. The classical problem must be translated to the quantum realm in this stage. To optimise and translate circuits to the target instruction set architecture (ISA), the transpiler rewrites them in several passes. Qiskit uses the term "transpiler" to highlight that it is a circuit-to-circuit rewriting tool rather than a complete compilation down to controller binaries, which is required to run circuits. However, another way to think of the transpiler is as an optimising compiler for quantum programmes. First, circuits that encode a classical problem are created to transfer it to quantum processing. Although Qiskit offers a handy circuit creation API to manage big circuits, domain-specific software or professionals are best suited

to handle this stage. The circuits are then modified to enable them to run on specific hardware. Since it is a circuit-to-circuit rewriting process rather than a complete compilation down to the classical controller instructions, we generally refer to this step as transpilation. The circuits are then assessed on a target backend using crude calculations. Lastly, a solution to the initial problem is obtained by post-processing the results[8]. Qiskit is a comprehensive open-source software library for quantum computing that covers all aspects of the stack, from simulation and emulation to application-level algorithms and actual interface with the IBM Q hardware. As a result, the tool is organised into four libraries named after the four classical elements: ignis, terra, aqua, and aer [9]. Since then, version updates have encouraged cited applications in a wide range of various disciplines to keep up with the speed of research and development. But Qiskit version 2.0, which was just released, offers an even wider range of implementation options, whether simulators or real-world quantum computers with 127 qubits or more. Among the intriguing recent application studies mentioned are social network analysis using quantum circuits [10], the non-Hermitian skin effect (NHSE) in a variety of classical metamaterials and even ultracold atomic arrays with its interaction with many-body dynamics [11], reinforcement learning [12], integrating with other applications that transform Qiskit is Sparse PauliOp in real-time operator evolution [13], and maintaining Hamming weights in variational quantum circuits [14]. Scale, hardware, speed, and quality have been extensively addressed in relation to Qiskit circuits, patterns, and architectures in the future period of potential larger applications, ranging from airlines to logistics [15].

Quantum Computing and Patterns

Trugenberger examined quantum pattern recognition in 2002. An energy functional is usually used to hold the information about the patterns to be remembered in classical associative memories. The memory functional drives the evolution of the input configuration to the associated output during information retrieval. A phase change in the statistical ensemble controlled by the memory energy functional is the cause of the capacity limitation. Because quantum associative memories are devoid of false memories, they perform better than classical ones [16].

Other works, in addition to quantum circuit patterns, try to offer high-level architectural design patterns that are specifically made to cover classical-quantum software systems. Consequently, a dataset containing quantum circuits, identified design patterns, and metrics for describing each circuit is provided. These metrics are used to analyse the utilisation of those patterns and characterise them. Eighty quantum circuits were examined in total. The five design patterns seen in both Qiskit and QASM circuits are also displayed in the study. Three distinct design patterns were identified: oracle, uniform superposition, and initialisation. As a result, the initial understanding of the extent of use of quantum software design patterns may help practitioners better understand the features and settings in which these patterns are used. i) This information can help developers understand how and when to apply specific patterns; ii) this understanding can help establish other design patterns to be applied in similar/different scenarios [17]. These are the primary implications for researchers and practitioners.

In order to measure the Hamming distance on a quantum computer, Schuld et al. proposed a quantum pattern classification technique based on Trugenberger's idea [18]. They also discussed the benefits of this algorithm using handwritten digit recognition from the MNIST database [19].

A few years ago, quantum pattern matching [20] was also mentioned in relation to the Grover search method [21].

“You have an online dictionary with 1,000,000 words in which the words are arranged alphabetically. You could program it to look for the solution to the puzzle so that it typically solves it after looking through 500,000 words. It is very difficult to do much better than this. But this is: only if you limit yourself to a classical computer. A quantum computer can be in multiple states at the same time and, by proper design, can carry out multiple computations simultaneously. In case the above dictionary were available on a quantum computer, it would be possible to carry out the search in only about 1,000 steps by using the quantum search algorithm.” – L K Grover[22].

Though it is a groundbreaking quantum search algorithm, it can indirectly be related to pattern search problem areas thus plausibly applied in other type of domain problems.

For the first time, Ortolano et al. show that there is a quantum advantage in the multi-cell pattern recognition issue. We demonstrate the use of entangled probe states and photon-counting to achieve quantum advantage in classification error over that achieved with classical resources through experimental realizations of digits from the MNIST handwritten digit dataset and the application of advanced classical post-processing. This confirms that the advantage gained through quantum sensors can be maintained throughout pattern recognition and intricate post-processing [23].

The approach described in an article encodes a classification issue for A Pattern Recognition approach for Quantum Annealers in the high-luminosity Large Hadron Collider (HL-LHC). Tracks from earlier works are built from n successive hits, resulting in $n - 1$ doublets. The algorithm aims to identify which subset of the enormous set of possible doublets from hits in the detector corresponds to the charged particle trajectories. The program seeks to increase the input doublet set's purity while maintaining efficiency [24].

Das et al. explore the feasibility of implementing a quantum pattern recognition protocol based on the swap test for quantum approaches in graphical data, such as photographs, and validate the concept using IBMQ noisy intermediate-scale quantum (NISQ) devices. While the noise in the real devices becomes detrimental for three or more qubits, they discover that the swap test may effectively and with acceptable fidelity determine the similarity between two patterns with a two-qubit protocol [25].

Building the most powerful quantum software or the largest fleet of utility-scale quantum processors is not enough to introduce practical quantum computing. Additionally, we must enable consumers to utilise our developed technologies effectively and efficiently. We are accomplishing that with the launch of Qiskit Code Assistant, which is currently accessible as a private preview through the IBM Quantum Premium Plan.

To help you develop better Qiskit code with less effort, Qiskit Code Assistant combines the pooled knowledge of Qiskit users throughout the quantum community with the advanced large language models (LLMs) of IBM® Watsonx™. Its ability to generate quantum code not only increases the efficiency and accessibility of quantum computing, but it also gives users a fresh, practical approach to learning how to develop Qiskit code. In order to help users learn how to write better code, streamline their development process, optimise their quantum programmes to produce better quantum circuits, and complete their projects faster, we expect that Qiskit Code Assistant will open up the world of quantum computing [26].

Quantum approach with EEG data

Binary class categorized EEG data for quantum machine learning with variational quantum classifier (VQC) where data is encoded with Rx rotation using TensorFlow Quantum and Cirq simulator [27]. A recurrent quantum neural network (RQNN) was implemented to increase signal separability in electroencephalogram (EEG) signals [28]. Using amplitude and phase encoding through Rx and Rz gates, the neuromarketing dataset used EEG signals, a hybrid quantum convolutional neural network-based classification work has been cited [29]. As a forward-looking strategy, employing a quantum-classical hybrid neural network, which combines quantum computing with the classical EEGNet architecture, to enhance EEG encoding and analysis acknowledges that the outcomes may not always outperform conventional techniques, but it does demonstrate its potential [30].

In the Qiskit SDK, using an open-source EEG dataset, quantum states resonance-based matching with target states is the contribution of the paper. Usually, a single quantum circuit is applied for a specific problem comprising a data list unless used for quantum machine learning purposes. In such a case, however, it normally becomes beyond the scope of

visualization, the application of simulation shots, and the count results. Applying and comparing with a significant stream dataset for identifying quantum state target pattern using the quantum circuits with simulation for understanding the plausibility of deployment using a simulator/quantum real hardware on a real-time basis is the novelty of this work.

Methods:

It is becoming increasingly evident that quantum advantage might be on the horizon as more and more academics use IBM's fleet of utility-scale quantum computers to investigate issues at the very edge of classical processing. Making use of Qiskit 2.0 [31].

The work flow rhythm comprises identification, and encoding Technique Selection of Data, Building QC by appropriate gate selection, Visualizing QC States by appropriate drawings, Checking and Applying Applicable Criteria, Data, simulation/Analysis for Born rule for probabilistic outcome, fidelity, entanglement entropy or mutual information to justify performance, and subsequent application for a mimicking stream data for observing the quantum states and thus pattern by comparing results.

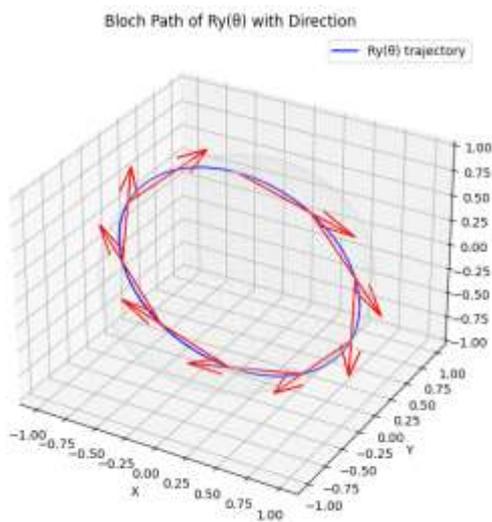


Figure 1: Ry rotation typical in Bloch sphere

The data encoding for signal can be categorised as either binarized or direct analogue input. For example, row 1 of Table 1, the first four variable values, 4.2068, -0.22239, -1.045500, and 1.2418 can be binarized considering a threshold value of 0.0, thus yielding 1001 which can be applied in a 4-qubit quantum circuit. However, as the numerical value differs, this may not be appropriate to uphold the data integrity without information loss or oversimplification. On the contrary, in an angle encoding, the values are normalized in the range from 0 to π to be fit for Bloch sphere. Therefore, the angle encoding method is selected for the direct analogue data input.

Rotating around the Y axis by angle in qubit is given by Eq. 1, where states 0 and 1 are the quantum basis states.

$$R_Y(\theta) = \cos\left(\frac{\theta}{2}\right) |0\rangle + \sin\left(\frac{\theta}{2}\right) |1\rangle \tag{1}$$

A typical Ry rotation in Bloch sphere, Figure 1, causes moving along the y-axis depending upon the value of angle.

$$|\psi_{input}\rangle = \sum_{x=0000}^{1111} \alpha_x |x\rangle \tag{2}$$

Eq. 2 is the four-qubit superposition state of all possible combinations. The coefficients α_x denote how weight or probability amplitude each basis state contributes.

$$\mu = \frac{1}{M} \sum_{k=1}^M r^k \tag{3}$$

Eq. 3 depicts the mean Bloch vector, where M is the total number of such vectors, r^k is the Bloch individual vector.

$$\alpha = \sqrt{\frac{1}{M} \sum_{k=1}^M (r^k - \mu)^2} \tag{4}$$

Eq. 4 gives the standard deviation of the Bloch vectors about their mean, Eq. (3).

Standardized Euclidean distance or Mahalanobis distance

$$d = \sqrt{\sum \left(\frac{r_{stream,i} - \mu_i}{\sigma + \epsilon} \right)^2} \tag{5}$$

U_2 and U_3 are universal single-qubit rotation gates customized for IBM Qiskit frameworks [32] for parametrized unitary operations, matrix representations as given below:

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos \frac{\theta}{2} & -e^{i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i(\phi+\lambda)} \cos \frac{\theta}{2} \end{bmatrix} \tag{6}$$

$$U_2(\phi, \lambda) = U\left(\frac{\pi}{2}, \phi, \lambda\right) \tag{7}$$

And $U_3(\theta, \phi, \lambda) = R_z(\phi)R_y(\theta)R_z(\lambda)$ (8)

Dataset selection

Open-source EEG dataset from Kaggle has been selected for this study [33]. A sample dataset comprises of 31000 samples each with 19 channels of EEG recording which can be treated as variables. To make a sample study, four variables have been chosen for 4 Qubit quantum circuit. Each qubit is encoded with angle encoding data as each variable.

The size of the dataset is (31,000, 19), Table I, provides the first five rows of four variables selected for this work. More information about the dataset is available in the citation link [33].

Table 1: Data of 1st five rows of the EEG signals.

	COL0	COL1	COL2	COL3
0	4.2068	-0.22239	-1.045500	1.2418
1	6.4815	-1.67370	-0.788550	2.2359
2	8.8589	-2.95130	0.023804	3.5748
3	11.0770	-3.68900	1.491700	4.9318
4	12.9220	-3.68790	3.552300	5.8837

Framing the Quantum Circuit

The quantum circuit of the problem statement comprises 4 qubits with two layers of R_y rotation gates. The first layer contains the encoded data for each value; thereafter, a fixed rotation of $\pi/2$ is applied to adjust the final states before measurement. Each qubit is row-wise encoded with each variable of data for the selected row, Figure 11. It is followed by the measurement gates, each qubit with a classical register. Post measurement density matrix is applied for all the qubits.

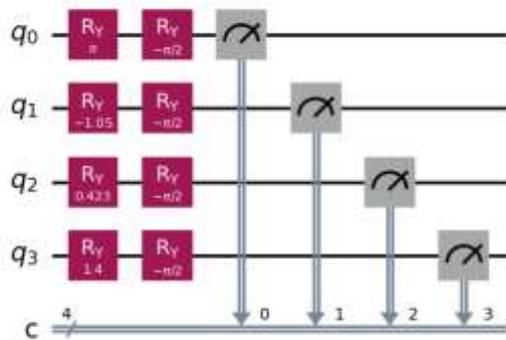


Figure 2: Quantum circuit with a row of EEG data encoding

As shown in Figure 2, the quantum circuit is configured with four qubits and classical registers for each qubit for measurement. The R_y gate is applied, Eq. 1, with angle rotation in terms of 0 to π , depending upon the row and the first four variables of the selected dataset. A second R_y gate is applied, with a fixed rotation angle for aligning the state vectors. Figure 3 is the transpiled circuit with application of U_2 and U_3 gates for a hardware-configured application. In the transpiled circuit, the unitary operations U_2 , Eq. 6, 7, and U_3 , Eq. 6, 8, are customized to the hardware requirement.

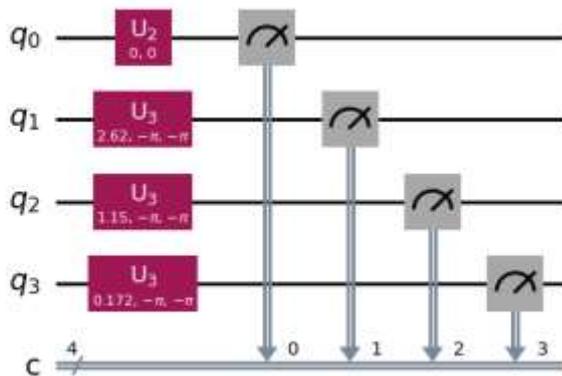


Figure 3: transpiled circuit

The depth, width, counts operations define circuit complexity, depending upon the problem type. Figure 4 and 5 reveal the gate counts of the original and transpiled circuit.

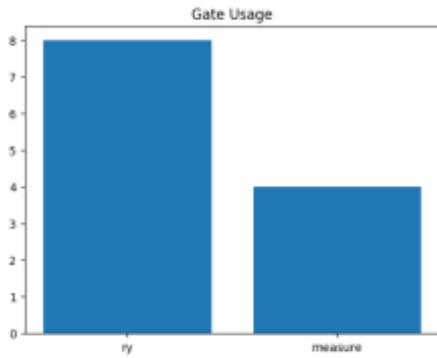


Figure 4: Gate Counts for circuit

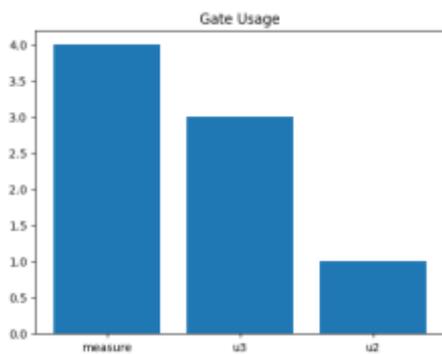


Figure 5: Gate counts for transpiled circuit

Results:

The array to latex of the statevector of quantum circuit yields as four-qubit 16-dimensional Hilbert space ($2^4 = 16$), and the system is in a structured superposition as per the Eq. 9, where each term $c_i|i\rangle$ is the amplitude-basis pair in the Hilbert space.

$$|\psi\rangle = \sum_{i=0}^{15} c_i|i\rangle \tag{9}$$

$$0.1533973983|0000\rangle + 0.1533973983|0001\rangle - 0.571406355|0010\rangle - 0.571406355|0011\rangle - 0.0991613712|0100\rangle - 0.0991613712|0101\rangle + 0.3693767841|0110\rangle + 0.3693767841|0111\rangle - 0.0132300992|1000\rangle - 0.0132300992|1001\rangle + 0.0492822097|1010\rangle + 0.0492822097|1011\rangle + 0.0085523926|1100\rangle + 0.0085523926|1101\rangle - 0.0318577208|1110\rangle - 0.0318577208|1111\rangle$$

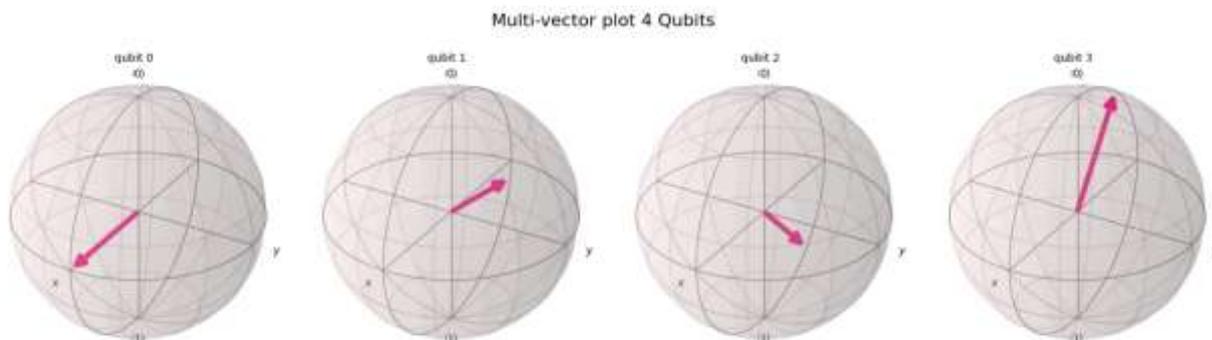


Figure 6: Multi-vector plot of quantum circuit

The multi-vector plot, Figure 6, reveals each qubit's state. Qubits 0 to 2 are in non-trivial bias and coherence states, and qubit 3 is separable. The first three qubits indicate entanglement.

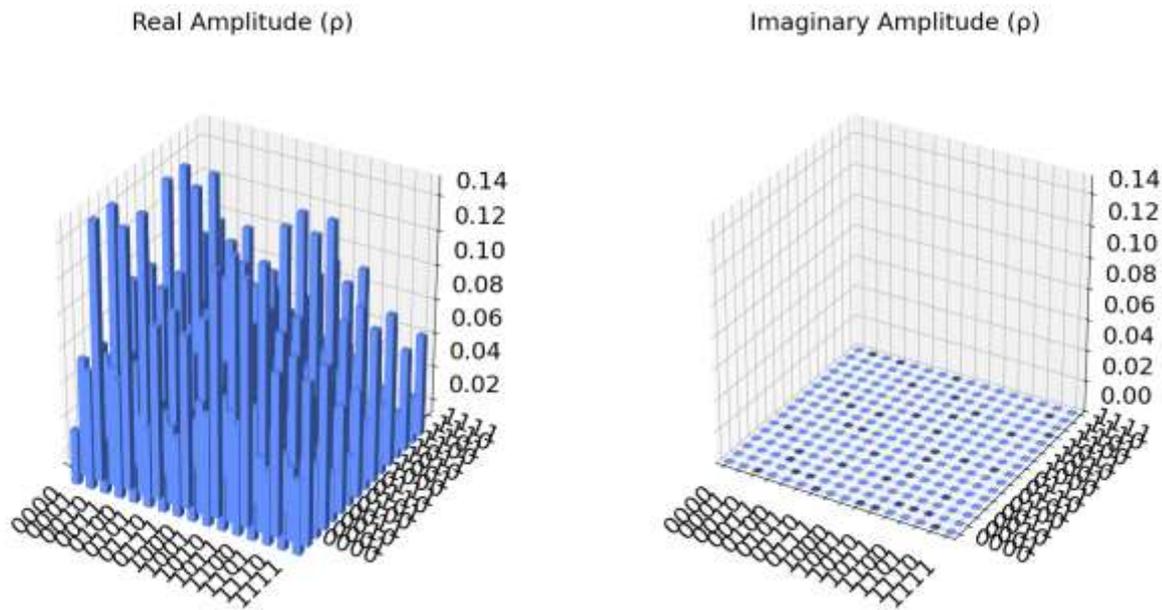


Figure 7: 3-D State-city plot of quantum circuit

Figure 7, the off-diagonal elements in the real part are indicative of coherence and presence of entanglement.

Hinton plot 4 Qubits

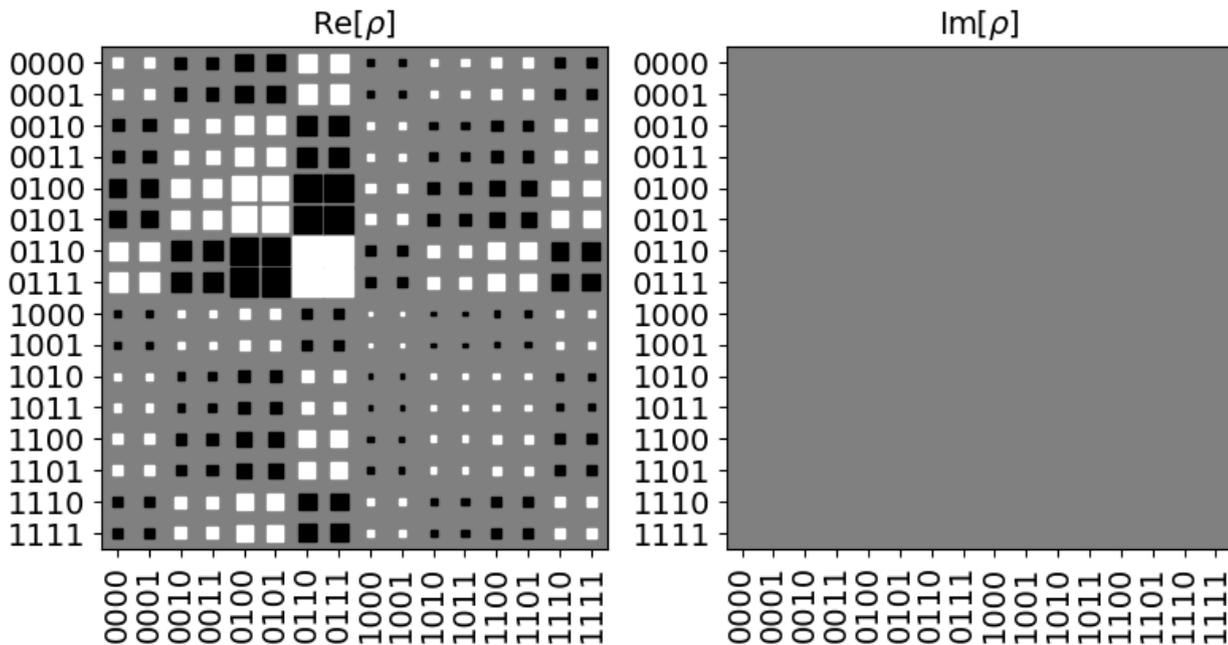


Figure 8: Hinton plot of quantum circuit

Figure 8, density matrix compliments the city state bars and the multi-vector plots.

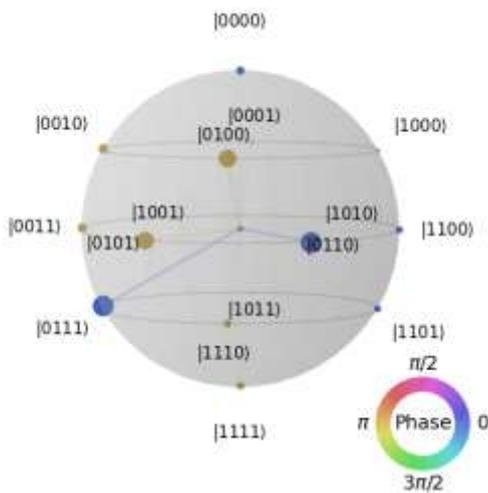


Figure 9: The Qsphere plot of the quantum circuit

Figure 9, Qsphere plot of all the qubits indicate phase clusters from 0(blue) to π (brown). The four qubit system is almost real. Qiskit also computes Purity of the system is $1.0000+0.0000j$, Von Neumann Entropy is 0.0000, meaning no classical uncertainty, Coherence (L1 norm of off-diagonals) is 9.8707, indicating strong coherence. Individually, Purity of Qubit 0 is $1.0000+0.0000j$, Purity of Qubit 1: $1.0000+0.0000j$, Purity of Qubit 2: $1.0000+0.0000j$, and Purity of Qubit 3: $1.0000+0.0000j$. In summary, A highly coherent, non-entangled, pure quantum superposition. The Bloch vectors, Qsphere, and Hinton plots all match the indications from other Figures. Coherence is just a signature of quantum superposition, not entanglement. Figure 10, is the outcome of the simulation run depicting all sixteen basis states from $|0000\rangle$ to $|1111\rangle$ and their probability amplitude. The states $|0110\rangle$ and $|0111\rangle$ resonate strongly with the probabilities of 0.22. The quantum signature of EEG signals is corresponding to the bursts or dominant signals.

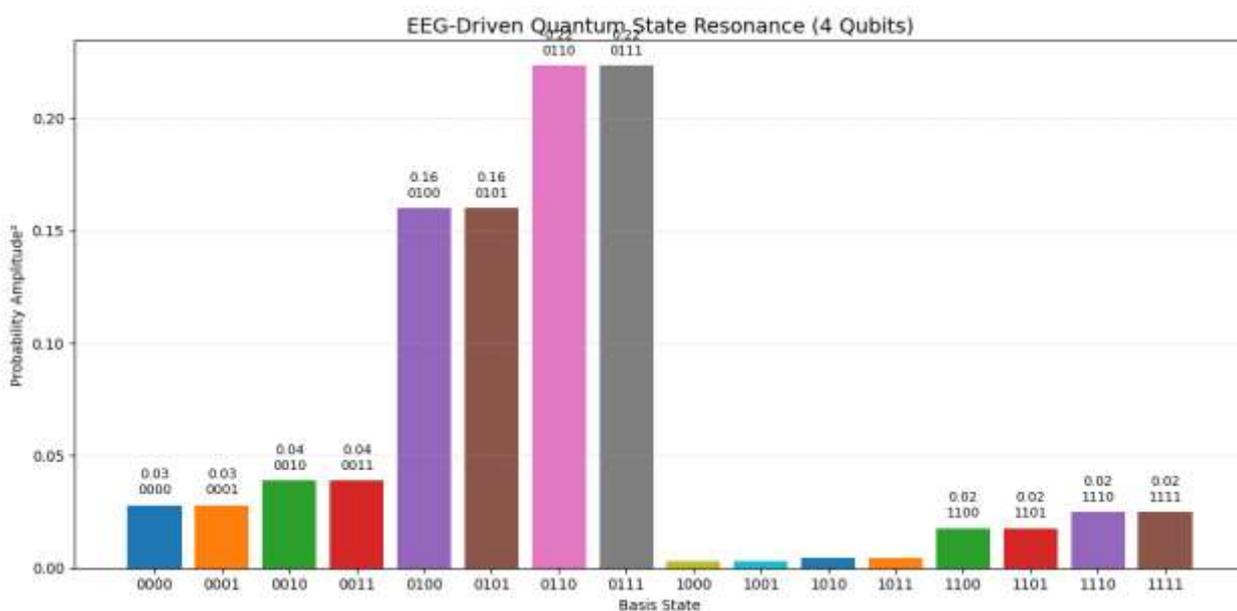


Figure 10: Quantum State Resonance of EEG data

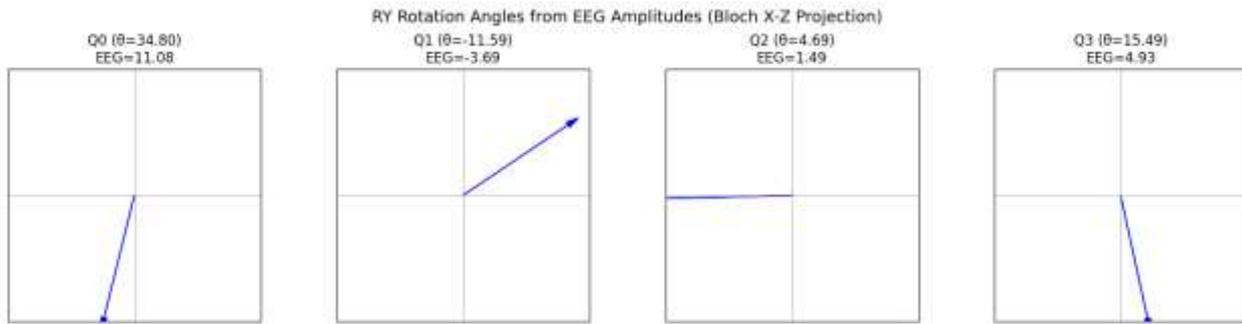


Figure 11: Ry rotation angles from EEG amplitudes

Data encoding in quantum circuit entail different techniques [34] viz. amplitude encoding, state encoding, angle encoding are a few normally used. Figure 11 depicts EEG amplitude encoding of the EEG data, the amplitude and angle relation for each qubit is exhibited separately. The EEG data is encoded in quantum states.

Figure 12 and 13 are the Aer Simulator plot of 1024 shots counts and quasi- probability distributions. It corroborates with the Figure 10 with the most counted states and their distributions.

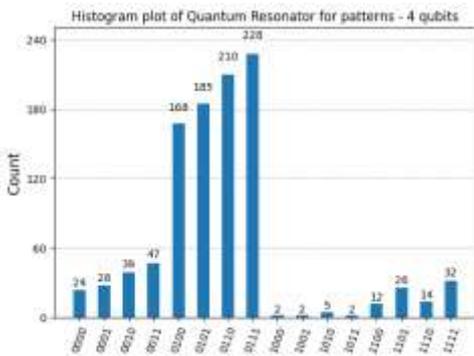


Figure 12: Aer Simulator Count Plot

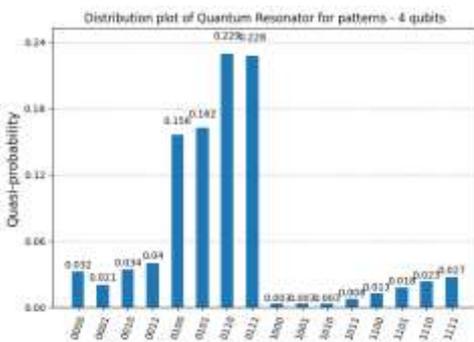


Figure 13: Aer Simulator Distribution plot

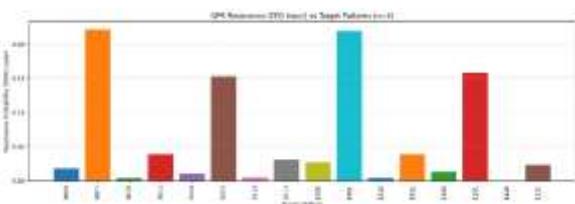


Figure 14: Resonance plots of quantum states

The resonance probability, depicted in Figure 14, depicts the similarity between the quantum state and each target state of the target data variables. The highest bars show which cognitive basis it resonates with the target basis.

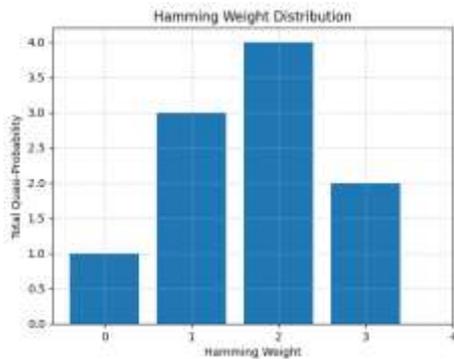
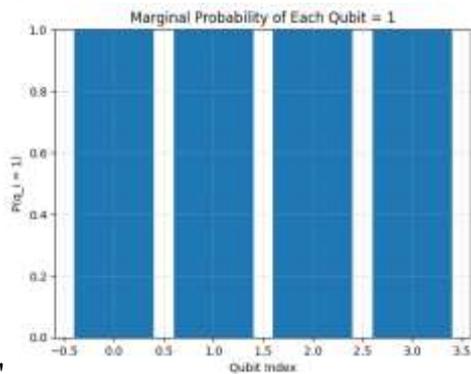


Figure 15: Hamming distance weights

Based on quantum data, specifically from a quantum state measurement or quasi-probability analysis across basis states in the computational foundation, the simulation results' Hamming weights, Figure 15, are revealed. All basis states with the same Hamming weight have their probabilities (or quasi-probabilities) added up, producing a distribution over



bitstring "weight."

Figure 16: Marginal probability

Figure 16 reveals the marginal probability of all the states being measured as 1.0. The Shannon entropy is calculated in qiskit as 2.26 bits. Figure 17, is the joint probabilities or counts of the measurements.

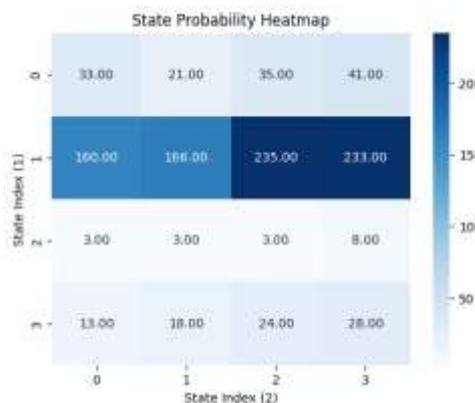


Figure 17: State probability heatmap

Further, using Qiskit noise model function, introducing some noise with depolarizing error, the [NOISY] Purity comes out as 0.1492+0.0000j, [NOISY] Von Neumann Entropy: 3.1224, and [NOISY] Coherence (L1 norm of off-diagonals): 0.0000. This indicates the mixed state, the Von Neuman entropy approaching 4 (log₂ (16)) revealing almost no quantum

information. Zero coherence yields no superposition being left in the system. Overall, it reveals that noise can have a significant effect on the system, in contrast to a structured superimposition and entangled state.

Baseline-based streaming data check on cosine similarity (using scikit-learn library, computes similarity as the normalized dot product of X and Y), threshold of 0.95 on the statevector baseline data and the target streaming real data.

Further, distance is calculated as per Eq. 5 with the mean, standard deviation of the max value and a threshold value of 3.0. Both the baseline and the live streaming data mean over channel is compared with the matching criteria. The result is acceptable if cosine similarity AND distance are within the respective threshold value. The streaming data is input from a different EEG set comprises of 31000 entries with 19 variables. A sample result of 5 rows, Table 2, reveals the match, cosine similarity and the distance calculated. Row 4 reveals a match criterion with the cosine similarity of 0.96 and the distance of 1.274, implying matching with the baseline parameters.

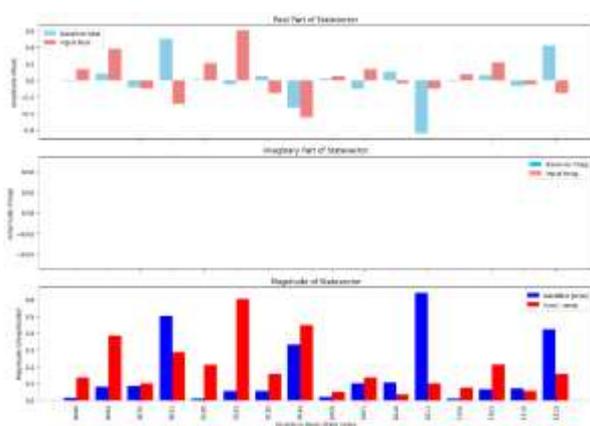


Figure 18: Baseline and amplitude state vector of real and imaginary part

Further, the quantum circuit is tested with stream data, taking another sample dataset from the EEG signals. Figures 19 and 20 represent the baseline and target states comparison, revealing the quantum signature similarity. The probability amplitude-squared reveals the closeness of the baseline and target pattern states' quantum states.

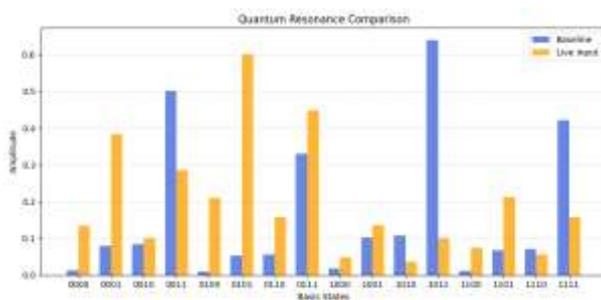


Figure 19: Quantum resonance comparison

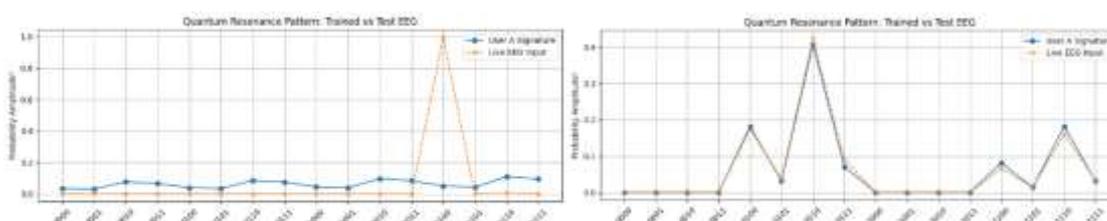


Figure 20: 2 different samples baseline and user data plot comparison (left- spike at state $|1100\rangle$), (right, mostly match)

Table 2: Match, cosine similarity and distance comparison

	MATCH	COSINE	DISTANCE
1	False	0.544	2.219
2	False	0.879	2.290
3	False	0.576	4.013
4	True	0.963	1.274
5	False	0.456	2.753

Table 3: Channel numbers and the statistical parameters of the streaming data

CHANNEL	MEAN	STD	MIN	MAX
26002	-2.877	9.024	-	18.223
			31.321,	
26009	0.606	9.698	-36.509	23.349
26013	1.531	9.857	-29.303	23.206
26015	0.940	10.350	-25.988	25.280
26018	0.065	11.738	-21.955	32.621

The baseline data is fed considering as enroll_windows and the mimic stream data as eeg_live. The shape of enroll_windows is (4, 31000, 19), whereas, when checked with mean values, the Shape of eeg_live is (31,000, 19). The data is fed in the quantum circuit by matching with the number of qubits. Table 3 provides a glimpse of sample mimic of streaming data parameters, which is finally matched with the target in this works.

Discussion:

A four-qubit system quantum circuit is built corresponding to four variables for angle data encoding based on an EEG dataset. The quantum pattern is matched with a target EEG sample and compared. Further, making a baseline with statistical parameters, the approach is applied for mimicking stream data, and quantum patterns are matched. The work shows that EEG signal resonance based on quantum states can be captured from a given signal. Moreover, successful application in streaming data encourages possible real-world application cum deployment for effectively monitoring the inflow of data for pattern identification. The work is however, extendable for n number of qubits matching with number of EEG variables, subject to the availability of quantum hardware, either the simulator or a real computer, albeit in case of a real hardware error correction will need to be considered.

Conclusion:

This paper demonstrates that a qiskit circuit can be applied for quantum state resonance-based EEG signal pattern matching and comparison without any necessity of additional gate introduction, thus resembling the Occam's razor principle of classical machine learning that a model should not be made more complex than necessary for bias and variance trade-off. This also indicates that the plausibility of a qiskit circuit for relatively large-scale or real-time applications for dealing with spontaneous incoming data. While a four qubit system is used corresponding to four EEG channel data as variables, the study can be generalized for n number of qubits and thus variables. It is also possible to extend by using other criteria, viz., wavelet functions for considering the streaming data rather than mean values for comparison. The findings provide a future direction for possible explorable application in different domains, viz., brain-computer interface

(BCI), quantum sensing application, baselining with selected Earth signals, and testing with SETI signals at an appropriate scale.

References

- [1] R. P. Feynman, "Simulating Physics with Computers," vol. 21, pp. 467–488, 1982.
- [2] V. Belis, P. Odagiu, and T. K. Aarrestad, "Machine learning for anomaly detection in particle physics," *Rev. Phys.*, vol. 12, p. 100091, 2024, doi: 10.1016/j.revip.2024.100091.
- [3] M. T. West, S. M. Erfani, C. Leckie, M. Sevier, L. C. L. Hollenberg, and M. Usman, "Benchmarking adversarially robust quantum machine learning at scale," *Phys. Rev. Res.*, vol. 5, no. 2, pp. 1–19, 2023, doi: 10.1103/PhysRevResearch.5.023186.
- [4] Q. T. Nguyen *et al.*, "Theory for Equivariant Quantum Neural Networks," *PRX Quantum*, vol. 5, no. 2, p. 1, 2024, doi: 10.1103/PRXQuantum.5.020328.
- [5] M. C. Caro *et al.*, "Generalization in quantum machine learning from few training data," *Nat. Commun.*, vol. 13, no. 1, 2022, doi: 10.1038/s41467-022-32550-3.
- [6] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, "Quantum machine learning beyond kernel methods," *Nat. Commun.*, vol. 14, no. 1, pp. 1–8, 2023, doi: 10.1038/s41467-023-36159-y.
- [7] Y. Gujju, A. Matsuo, and R. Raymond, "Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications," *Phys. Rev. Appl.*, vol. 21, no. 6, 2024, doi: 10.1103/PhysRevApplied.21.067001.
- [8] A. Javadi-Abhari *et al.*, "Quantum computing with Qiskit," pp. 1–19, 2024, [Online]. Available: <http://arxiv.org/abs/2405.08810>
- [9] R. Wille, R. Van Meter, and Y. Naveh, "IBM's Qiskit Tool Chain: Working with and Developing for Real Quantum Computers," *Proc. 2019 Des. Autom. Test Eur. Conf. Exhib. DATE 2019*, pp. 1234–1240, 2019, doi: 10.23919/DATE.2019.8715261.
- [10] S. S. Singh, S. Kumar, S. K. Meena, K. Singh, S. Mishra, and A. Y. Zomaya, "Quantum social network analysis: Methodology, implementation, challenges, and future directions," *Inf. Fusion*, vol. 117, no. September 2024, p. 102808, 2025, doi: 10.1016/j.inffus.2024.102808.
- [11] R. Shen, T. Chen, B. Yang, and C. H. Lee, "Observation of the non-Hermitian skin effect and Fermi skin on a digital quantum computer," *Nat. Commun.*, 2023, doi: 10.1038/s41467-025-55953-4.
- [12] D. Kremer, V. Villar, H. Paik, I. Duran, I. Faro, and J. Cruz-Benito, "Practical and efficient quantum circuit synthesis and transpiling with Reinforcement Learning," 2024, [Online]. Available: <http://arxiv.org/abs/2405.13196>
- [13] T. Begušić and G. K. L. Chan, "Real-Time Operator Evolution in Two and Three Dimensions via Sparse Pauli Dynamics," *PRX Quantum*, vol. 6, no. 2, pp. 1–13, 2025, doi: 10.1103/PRXQuantum.6.020302.
- [14] L. Monbroussou, J. Landman, A. B. Grilo, R. Kukla, and E. Kashefi, "Trainability and Expressivity of Hamming-Weight Preserving Quantum Circuits for Machine Learning," no. 2, 2023, [Online]. Available: <http://arxiv.org/abs/2309.15547>
- [15] M. AbuGhanem, *IBM quantum computers: evolution, performance, and future directions*, vol. 81, no. 5. Springer US, 2025. doi: 10.1007/s11227-025-07047-7.
- [16] C. . A. . Trugenberger, "Quantum pattern recognition," *2013 Conf. Lasers Electro-Optics Eur. Int. Quantum Electron. Conf. CLEO/Europe-IQEC 2013*, 2013, doi: 10.1109/CLEOE-IQEC.2013.6801621.
- [17] R. Perez-Castillo, M. Fernandez-Osuna, J. A. Cruz-Lemus, and M. Piattini, "A preliminary study of the usage of design patterns in quantum software," *Proc. - 2024 IEEE/ACM 5th Int. Work. Quantum Softw. Eng. Q-SE 2024*, pp. 41–48, 2024, doi: 10.1145/3643667.3648220.

- [18] C. A. Trugenberger, “Probabilistic Quantum Memories,” *Phys. Rev. Lett.*, vol. 87, no. 6, pp. 67901-1-67901-4, 2001, doi: 10.1103/PhysRevLett.87.067901.
- [19] M. Schuld, I. Sinayskiy, and F. Petruccione, “Quantum computing for pattern classification,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8862, pp. 208–220, 2014, doi: 10.1007/978-3-319-13560-1.
- [20] K. K. . Soni and A. Rasool, “Pattern Matching: A quantum Approach”.
- [21] L. K. Grover, “A fast quantum mechanical algorithm for database search,” *Proc. Annu. ACM Symp. Theory Comput.*, vol. Part F1294, pp. 212–219, 1996, doi: 10.1145/237814.237866.
- [22] L. O. V. K. Grover, “From Schrodinger ’ s equation to the quantum search,” no. January 2000, pp. 333–348, 2001.
- [23] G. Ortolano *et al.*, “Quantum-Enhanced Pattern Recognition,” *Phys. Rev. Appl.*, vol. 20, no. 2, pp. 1–13, 2023, doi: 10.1103/PhysRevApplied.20.024072.
- [24] F. Bapst *et al.*, “A Pattern Recognition Algorithm for Quantum Annealers,” *Comput. Softw. Big Sci.*, vol. 4, no. 1, pp. 1–7, 2020, doi: 10.1007/s41781-019-0032-5.
- [25] S. Das, J. Zhang, S. Martina, D. Suter, and F. Caruso, “Quantum pattern recognition on real quantum processing units,” *Quantum Mach. Intell.*, vol. 5, no. 1, 2023, doi: 10.1007/s42484-022-00093-x.
- [26] D. G. Almeida, J. Cruz-Benito, and R. Davis, “Introducing Qiskit Code Assistant,” 2024. Accessed: Apr. 19, 2025. [Online]. Available: <https://www.ibm.com/quantum/blog/qiskit-code-assistant>
- [27] I. D. Lins *et al.*, “Quantum machine learning for drowsiness detection with EEG signals,” *Process Saf. Environ. Prot.*, vol. 186, pp. 1197–1213, 2024, doi: <https://doi.org/10.1016/j.psep.2024.04.032>.
- [28] V. Gandhi, G. Prasad, D. Coyle, L. Behera, and T. M. McGinnity, “Quantum neural network-based EEG filtering for a brain-computer interface,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 25, no. 2, pp. 278–288, 2014, doi: 10.1109/TNNLS.2013.2274436.
- [29] S. Aishwarya, V. Abeer, B. B. Sathish, and K. N. Subramanya, “Quantum Computational Techniques for Prediction of Cognitive State of Human Mind from EEG Signals,” *J. Quantum Comput.*, vol. 2, no. 4, pp. 157–170, 2020, doi: 10.32604/jqc.2020.015018.
- [30] C. S. Chen, S. Y. C. Chen, A. H. W. Tsai, and C. S. Wei, “QEEGNet: Quantum Machine Learning for Enhanced Electroencephalography Encoding,” *IEEE Work. Signal Process. Syst. SiPS Des. Implement.*, pp. 153–158, 2024, doi: 10.1109/SiPS62058.2024.00035.
- [31] Qiskit Team, “Release News: Qiskit SDK v2.0 is here!” Accessed: Apr. 19, 2025. [Online]. Available: <https://www.ibm.com/quantum/blog/qiskit-2-0-release-summary>
- [32] IBM Qiskit, “UGate.” Accessed: Apr. 20, 2025. [Online]. Available: <https://docs.quantum.ibm.com/api/qiskit/qiskit.circuit.library.UGate>
- [33] “Complete EEG dataset.” Accessed: Mar. 20, 2025. [Online]. Available: <https://www.kaggle.com/datasets/amananandrai/complete-eeg-dataset>
- [34] H. Biswas, “Data Encoding for VQC in Qiskit , A Comparison With Novel Hybrid Encoding”.