uc3m | Universidad Carlos III de Madrid

CSIC
Consejo Superior de Investigaciones Científicas

Master in Quantum Technologies & Engineering

FINAL THESIS

Hardware-Adapted Quantum Machine Learning

Ignacio Benito Acedo Blanco

Carlos Ruiz Pastor

Javier González Conde

Academic course 2024-2025

AVOID PLAGIARISM

The University uses the Turnitin Feedback Studio program within the Aula Global for the delivery of student work. This program compares the originality of the work delivered by each student with millions of electronic resources and detects those parts of the text that are copied and pasted. Plagiarizing in a TFM is considered a **Serious Misconduct**, and may result in permanent expulsion from the University.

Hardware-Adapted Quantum Machine Learning

Quantum Machine Learning (QML) has largely focused on theoretical models, often overlooking practical challenges posed by real-world quantum hardware. In this work, we introduce pulsed quantum machine learning (PQML), a framework that replaces traditional parameterized quantum gates with quantum pulses, addressing hardware constraints in the noisy intermediate-scale quantum (NISQ) era. This approach offers benefits such as enhanced expressivity and more efficient classical data embedding. We also introduce an encoding map that aims to preserve the inherent structure of datasets on the Bloch sphere through the use of control pulses. As a proof of concept, we implemented the PQML approach via numerical simulations with the data re-uploading model, demonstrating superior accuracy and performance across a range of datasets compared to traditional gate-based methods. Our results highlight PQML's potential to advance QML by bridging the gap between theoretical development and practical hardware implementation, paving the way for more robust and efficient quantum algorithms.

I. INTRODUCTION

Quantum computing has emerged as one of the most exciting frontiers in modern science and technology, offering the potential to revolutionize a wide range of fields, from cryptography and optimization to drug discovery and materials science [1]. The foundation of quantum computing lies in its ability to leverage quantum mechanical properties, such as superposition, entanglement, and interference, to solve computational problems that are intractable for classical computers. In fact, the promise of exponential speedups for certain tasks has generated significant interest from academia, industry, and governments around the world, with substantial resources being invested in the development of both quantum algorithms and hardware [2].

Despite this enthusiasm, the practical realization of quantum computing faces significant challenges. Even though some promising advances glimpse the beginning of a feasible implementation of quantum error correction codes [3], current quantum computers still remain in the noisy intermediate-scale quantum (NISQ) era [4], characterized by a limited number of qubits, short coherence times, and relatively high error rates. These hardware constraints prevent the implementation of many well-known quantum algorithms, such as Shor's or Grover's algorithms, at scales of practical relevance [5]. As a result, much of the focus in quantum computing research has shifted toward identifying applications and algorithms that can provide near-term advantages, leveraging the capabilities of NISQ devices.

One of the most promising areas of quantum computing is quantum machine learning (QML) [6], an interdisciplinary field that integrates quantum computing with machine learning techniques. QML seeks to exploit the inherent parallelism of quantum systems to improve the performance of machine learning tasks such as classification, clustering, regression, and generative modeling. In this sense, QML has the potential to reduce the complexity of high-dimensional data analysis and enhance the representational power of machine learning models. However, the practical deployment of QML models on current hardware remains a significant challenge. In this regard, there exist many QML frameworks designed from an abstract theoretical perspective, often neglecting some particular hardware limitations of NISQ devices [7]. As a result, there is a pressing need to develop QML models that are both theoretically sound and optimized for NISQ hardware.

In this article, we introduce a novel approach to QML, termed "pulsed quantum machine learning" (PQML), which replaces the parametrized set of gates in circuit-based models with parametrized pulses as building blocks. This innovation enhances the expressivity of quantum models by enabling richer feature representations while improving their practical implementation on current quantum hardware through programming on a more native layer, additionally enabling faster operations. By leveraging pulse programming, our method aligns more closely with the capabilities of NISQ devices.

Previous efforts [8, 9] explored similar ideas, focusing primarily on the variational quantum eigensolver algorithm. However, these efforts lacked a clearly defined ansatz or model and did not effectively optimize training procedures. Instead, they proposed an iterative circuit-building process during training, which struggled to identify optimal parameters. Furthermore, they did not address the essential challenge of encoding quantum features—an area central to our work.

By aligning quantum models more closely with the native capabilities of the hardware, PQML not only enhances model expressivity but also improves computational efficiency. This makes it particularly suited for tasks requiring rich feature representations. Moreover, we introduce a novel encoding method that leverages the increased expressivity of control pulses to further enhance the model's performance.

To demonstrate the versatility of PQML, we apply it to a quantum neural network inspired by the framework introduced in [10], which employs data re-uploading for model construction. While the original design demonstrates strong theoretical potential, it does not fully address hardware-level constraints. Our approach bridges this gap by replacing parametrized gates with parametrized pulses, taking advantage of pulse-level control to optimize the model's implementation.

To substantiate our claims, we present a detailed discussion of the theoretical advantages of the proposed model and validate its feasibility through numerical simulations. The results demonstrate that our approach has the potential for effective implementation on real quantum hardware, showcasing its

practicality as a tool for QML research and applications.

The rest of the manuscript is organized as follows. In Section II, we review the theoretical foundation of quantum optimal control and its applicability to the current hardware implementation of quantum computing with superconducting qubits. In Section III, we examine the core concepts of QML used in the model presented in [10], detailing its principles and limitations. In Section IV, we introduce our proposed modifications, combining QML with quantum control and pulse programming techniques to enhance its implementation. Section V presents the experimental setup and results, providing evidence of the practical advantages of our approach. Finally, in Section VI, we discuss the broader implications of our findings and propose future research directions in QML.

II. QUANTUM CONTROL

Quantum optimal control (QOC) is a field that focuses on manipulating and optimizing the dynamics of quantum systems through external controls, typically in the form of electromagnetic fields, to achieve desired quantum states or operations [11]. By exploiting the principles of quantum mechanics, quantum control techniques allow for precise adjustments to qubit behavior, enhancing the fidelity and performance of quantum computations.

Pulse programming, a crucial aspect of quantum control, represents the most direct and fundamental way to interact with quantum hardware, essentially "*communicating*" with quantum computers in its native language. Unlike gate-level programming, which relies on abstracted quantum operations carefully calibrated regularly and built from limited predefined blocks, pulse programming gives precise control over the electromagnetic signals that manipulate qubit dynamics at a granular level. This low-level access provides a deeper understanding of quantum hardware, enabling researchers to fine-tune quantum gates, reduce latency and error rates, increase the accuracy of certain processes, and design custom operations that surpass the limitations of conventional fixed set of allowed operations [12].

In this section, we introduce the key concepts of pulse programming on qubits based on superconducting circuits, laying the groundwork for constructing a PQML model.

A. Transmon qubits

Up to the moment of realization of this work, superconducting quantum computers represent one of the most advanced and well-developed quantum computing technologies [3, 13, 14]. In the architectures built on this platform, the information is encoded in the quantum degrees of freedom of nanofabricated, anharmonic oscillators composed of superconducting circuit elements. At the core of these circuits are the so-called *transmon qubits* [15], which are built using Josephson junctions. The dynamics of transmon qubits are described by the Hamiltonian

$$\hat{H}_0 = 4E_c(\hat{n} - n_g)^2 - E_J \cos\hat{\varphi}, \tag{1}$$

where the charge operator is defined as $\hat{n} = -\hat{Q}/2e$, representing the number of Cooper pairs crossing the Josephson junction. The charging energy is given by $E_c = e^2/(2(C_J + C))$, with C_J the junction capacitance and C the shunt capacitance, while the Josephson energy is $E_J = \Phi_0 I_c/2\pi$ and I_c the critical current. The flux operator is defined as $\hat{\varphi} = 2\pi\hat{\Phi}/\Phi_0$, where Φ_0 is the magnetic flux quantum, and $n_g = Q_g/2e$ accounts for an offset charge due to capacitive coupling with external charges. A schematic of the circuit implementation for a transmon qubit is shown in Fig. **1**.

The ratio E_J/E_c is crucial in determining the qubit's robustness to charge fluctuations. Changes in the gate charge n_g can strongly influence the qubit's transition frequency, leading to dephasing. Operating in the transmon regime, where $E_J \gg E_c$, delocalizes the charge degree of freedom, making the energy levels nearly insensitive to the gate charge and thereby reducing dephasing [16]. In this regime, the offset charge n_g can often be neglected.

Using the ladder operators, $\hat{a}$ and $\hat{a}^\dagger$, the flux and charge can be expressed as

$$\hat{n} = i\sqrt[4]{\frac{E_J}{32E_c}}(\hat{a}^\dagger - \hat{a}) \quad \text{and} \quad \hat{\varphi} = \sqrt[4]{\frac{2E_C}{E_J}}(\hat{a}^\dagger + \hat{a}), \tag{2}$$

and therefore the Hamiltonian can be rewritten as

$$\hat{H}_{1q} = \omega_q \hat{a}^\dagger\hat{a} + \frac{\alpha}{2}\hat{a}^\dagger\hat{a}^\dagger\hat{a}\hat{a}. \tag{3}$$

Here, the first term corresponds to the qubit frequency $\omega_q = \sqrt{8E_cE_J} - E_c$, and the second term represents the anharmonicity, defined as $\alpha = -E_c$. In practice only the two lowest energy levels are used to approximate the system as a two-level qubit, effectively ignoring the higher energy levels. This yields to the simplified qubit Hamiltonian

$$\hat{H}_{1q} \simeq \frac{\omega_q}{2}\hat{\sigma}^z. \tag{4}$$

Note that this simplification is valid under typical operational conditions, though it introduces potential errors due to the excitation of higher levels, especially when the microwave pulse parameters are not perfectly optimized [17].

The transmon system is manipulated using microwave pulses, which enables a precise control over the state of the anharmonic oscillator. By tuning the pulse parameters, the population distribution across its energy levels can be accurately adjusted. After applying certain approximations and transformations [18, 19], the driven effective Hamiltonian in the interaction picture of a single transmon qubit can be expressed as

$$\hat{H}_d = -\frac{\Delta_\omega}{2}\hat{\sigma}^z + \frac{\Omega s(t)}{2}(\cos\gamma\,\hat{\sigma}^x + \sin\gamma\,\hat{\sigma}^y), \tag{5}$$

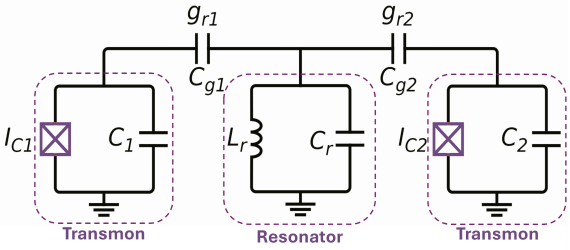

FIG. 1. Capacitive coupling of two transmon qubits via a coupler in form of a linear resonator. Each transmon is made with a Josephson junction, I, and a capacitor, C. The transmon qubits interact through an intermediate resonator, which is composed of an inductance, L, and a capacitor, C. Figure adapted from [18].

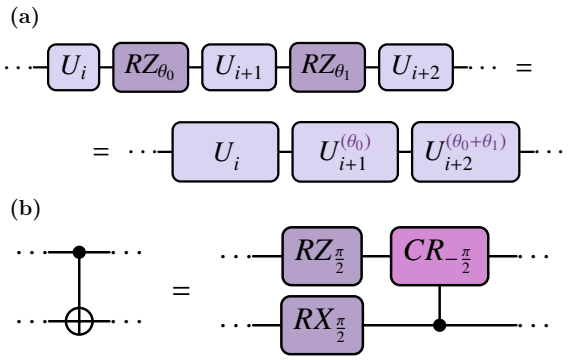

FIG. 2. (a) Virtual Z rotations scheme. U gates represent an arbitrary gate. The upper indices denote the phase offset of the pulse. (b) CNOT gate from the three basic blocks for transmon. Here CR represents a cross resonance gate. Virtual rotations and CNOT gate transpilation with transmon architecture.

where the applied pulse takes the form $\Omega s(t)\cos(\omega_d t + \gamma)$. Here, Ω denotes the driving pulse amplitude, $s(t)$ the pulse shape, ω_d is the driving frequency and γ denotes the phase. Finally, we define $\Delta_\omega = \omega_q - \omega_d$ as the *detuning* between the qubit and the drive frequencies.

From this Hamiltonian, the abstraction of gate operations becomes apparent, as it describes the interaction between the qubit and an external field. By choosing a pulse with a frequency resonant with the qubit frequency, *i.e.* $\Delta_\omega = 0$, one can perform rotations in the XY-plane where the axis of rotation is determined by the pulse phase, γ, and the rotation angle is controlled by the duration and the Rabi frequency of the pulse, $\Omega s(t)$. For example, setting $\gamma = \pi/2$ enables the implementation of Y-rotations.

On the other hand, exact rotations around the Z axis can be achieved using a technique known as virtual Z (VZ) gates [20]. This method enables perfect Z-rotations without requiring physical operations. Instead, the rotation is *virtually* implemented by introducing a phase shift to subsequent pulses as illustrated in Fig. 2a. Consequently, VZ rotations are both cost-free and error-free, as they are executed instantaneously.

These two transformations—XY-rotations with frequency-tuned pulses and VZ rotations—enable universality in the Hilbert space of a single qubit, as any SU(2) unitary matrix, U, can be decomposed into three rotations around the Y and Z axes,

$$U = R_Z(\theta_3)R_Y(\theta_2)R_Z(\theta_1). \tag{6}$$

These rotation angles are commonly referred to as Euler angles

B. Interaction between transmon qubits

Qubits can be interconnected in various ways to enable interactions. Inspired by the IBM quantum computer implementation, this work adopts capacitive coupling via a coupler in form of a linear resonator [18], eliminating the need for tunable qubit frequencies. A schematic representation of this coupling mechanism is shown in Fig. 1.

The effect of a coupling via a capacitor adds an interacting term to the transmon Hamiltonian, see Eq. (1), given by $\hat{H}_{\text{int}} = C_g \hat{V}_1 \hat{V}_2$, where C_g denotes the coupling capacitor and V_q the voltage operator of the corresponding voltage node being connected. Identifying as $\hat{V}_q \propto \hat{n}_q \propto i(\hat{a} - \hat{a}^\dagger)$, the Hamiltonian of two coupled transmons reads [18]

$$\hat{H}_{2q} = \sum_{q=1,2}\left(\omega_q \hat{a}_q^\dagger \hat{a}_q + \frac{\alpha_q}{2}\hat{a}_q^\dagger \hat{a}_q^\dagger \hat{a}_q \hat{a}_q\right) + \omega_r \hat{a}_r^\dagger \hat{a}_r + \sum_{q=1,2} g_{qr}(\hat{a}_q^\dagger a_r + \hat{a}_q \hat{a}_r^\dagger). \tag{7}$$

In this Hamiltonian, the summation over $q = 1, 2$ represents the two qubits, where ω_q denotes their respective frequencies and α_q their anharmonicities. The term $\omega_r \hat{a}_r^\dagger \hat{a}_r$ accounts for the energy levels of the resonator, characterized by the frequency ω_r. Finally, the coupling term g_{qr} describes the interaction between each qubit and the resonator.

In the dispersive limit, where $g_{qr} \ll \Delta_{q,r} := \omega_q - \omega_r$, the resonator can be treated as an isolated system, allowing the coupling term to be adiabatically eliminated, which effectively transfers the interaction between the qubits [21]. Projecting into the zero resonator excitation subspace, the effective Hamiltonian that describes the full system in the absence of an external driving reads

$$\hat{H}_{2q} \simeq \sum_{q=1,2}\frac{\omega_q}{2}\hat{\sigma}_q^z + J(\hat{\sigma}_1^x \hat{\sigma}_2^x + \hat{\sigma}_1^y \hat{\sigma}_2^y), \tag{8}$$

where we defined $J = g_1 g_2(\omega_1 + \omega_2 - 2\omega_r)/2(\omega_1 - \omega_r)(\omega_2 - \omega_r)$ and took a two-level approximation.

Once we have described the two transmon system, the focus shifts to controlling both qubit states and their entanglement. On the one hand, when $J \ll \omega_1, \omega_2$, the system can be

effectively described by the individual eigenstates of each qubit. In this regime, applying a driving pulse to one qubit, with its frequency precisely calibrated to match the qubit's resonance frequency, can be treated as affecting the qubits separately with negligible error [18]. Under these conditions, Eq. (5) remains valid, allowing for rotations on the XY-plane as before. VZ rotations also remain applicable, preserving the universality of single-qubit operations. Therefore, achieving two-qubit universality requires only the implementation of an entanglement operation between qubits. Commonly, as in case of the IBM quantum computers that have showed a prominent scalability, the entanglement operation chosen is the so-called cross-resonance (CR) gate [22]. This gate is achieved via a single microwave pulse that entangles a pair of fixed-frequency qubits, leveraging static coupling and non-tunable frequency qubits.

Although the mathematical details of the CR gate can be intricate [23], the underlying logic is relatively simple. The CR gate involves driving one qubit with a pulse tuned to the frequency of the other qubit, effectively ducing an XZ-rotation. Therefore, the effective Hamiltonian of an applied pulse to qubit 1 with a driving corresponding to the frequency of qubit 2 (*i.e.* $\omega_d = \omega_2$), yields the expression [19]

$$\begin{aligned}\hat{H}_{\mathrm{CR}} = &-\frac{\Delta_{12}}{2}\hat{\sigma}^z \otimes \hat{\mathbb{I}} \\ &+\frac{\Omega(t)}{2}\Big[\cos\gamma\left(\hat{\sigma}^x \otimes \hat{\mathbb{I}} + \mu\,\hat{\sigma}^z \otimes \hat{\sigma}^x + \nu\,\hat{\mathbb{I}} \otimes \hat{\sigma}^x\right) \\ &\quad + \sin\gamma\left(\hat{\sigma}^y \otimes \hat{\mathbb{I}} + \mu\,\hat{\sigma}^z \otimes \hat{\sigma}^y + \nu\,\hat{\mathbb{I}} \otimes \hat{\sigma}^y\right)\Big].\end{aligned} \tag{9}$$

where $\Delta_{12} = \omega_1 - \omega_2$ is the difference between the qubit frequencies, the coefficient ν represents the cross-talk effect, and $\mu = \frac{J\alpha_2}{\Delta(\alpha_2+\Delta)}$ the CR-interaction term.

In summary, there are three fundamental pulse operations used to manipulate superconducting quantum computers: a pulse resonant with the qubit frequency for single-qubit XY-rotations, VZ rotations to implement Z-axis single-qubit rotations, and the CR gate to entangle qubits and enable multi-qubit rotations. Any multi-qubit operation can be transpiled using these three foundational building blocks. For example, the CNOT gate can be constructed in this manner, as illustrated in Fig. 2b.

III. QUANTUM MACHINE LEARNING

QML leverages quantum systems to enhance machine learning tasks, with the hope of achieving unique advantages such as greater representational power and reduced computational complexity [24]. However, deploying QML on current NISQ hardware remains a challenge due to noise, limited coherence times, and constrained qubit resources. In this context, strategies that align theoretical advancements with practical hardware constraints are essential to unlock the full potential of QML.

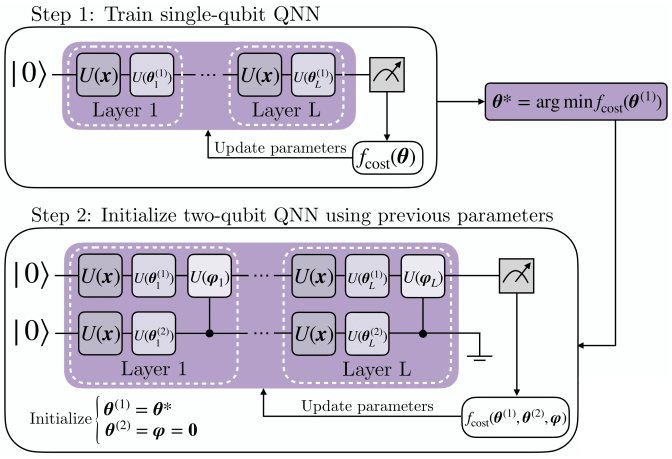

FIG. 3. The iterative training process for a two-qubit data re-uploading QNN begins by training a single-qubit QNN to determine optimal parameters, $\boldsymbol{\theta}^*$. These parameters are reused in the two-qubit QNN, with additional parameters initialized to zero. Figure adapted from [10].

A key area of focus in the NISQ era is the development of variational quantum circuits (VQC), which combine quantum circuits with classical optimization to address the limitations of noisy hardware [25]. These hybrid approaches allow parameterized quantum circuits to adapt to specific tasks, balancing theoretical expressiveness with practical feasibility. Within this framework, techniques like *data re-uploading* are crucial, as they enhance the representational capacity of variational circuits while staying compatible with hardware constraints.

The remaining of section is dedicated to providing an in-depth explanation of the theoretical model described in [10], which leverages the concept of data re-uploading to construct a QNN.

The QNN model based on data re-uploading, proposed in Ref. [26], takes inspiration from the layered structure of classical neural networks. It relies on alternating encoding operations with parametrized trainable operations, enabling a universal quantum classifier with just a single qubit. The expressivity of this model can be understood by analyzing the Fourier components of the function it generates, as discussed in Ref. [27]. Repeatedly applying encoding operations allows the model to access a richer frequency spectrum, determined by the eigenvalues of the data-encoding Hamiltonians. On the other hand, the parametrized operations provide the flexibility to adjust the weight of each Fourier component, enhancing the model's ability to capture complex patterns on data. This makes the model especially well-suited for the NISQ era since it maximizes expressive power with a minimal number of qubits.

To create multi-qubit models, Ref. [26] suggests a straightforward method for extending this approach; however, it does not provide any guarantees regarding scalability. In Ref. [10], an iterative training method is proposed to scale the data re-

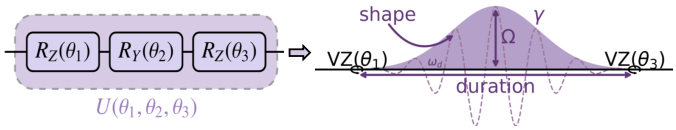

FIG. 4. Pulse-based gate replacement. Each Z-rotation gate is implemented as a VZ rotation, introducing neither physical errors nor additional duration. The fixed Y-rotation gate is replaced by a parameterized single-qubit pulse, enabling arbitrary rotations within the XY-plane. As a result, the trainable parameters consist of the pulse phase, pulse amplitude, and the two VZ rotation angles. Note that the original β parameter is linked to the pulse amplitude Ω, the pulse shape $s(t)$, and the pulse duration.

uploading model to multiple qubits. The training process begins by defining and optimizing a single-qubit QNN. The encoding block uses a general SU(2) rotation (e.g., as defined in Eq. (6)), with the Euler angles representing the Cartesian coordinates of the data point. The parametrized block consists of an arbitrary rotation gate, where the angles are treated as trainable parameters. After forwarding each input point through the one-qubit network, the fidelity of the resulting state is measured against the computational basis states, with one basis state assigned to each data label. The objective function is designed to maximize the fidelity across the entire dataset.

After optimizing the single-qubit model, an additional qubit register is introduced, maintaining the same structure as the single-qubit model: arbitrary rotations in both the encoding and parametrized blocks, along with an arbitrary controlled gate applied after each layer. To ensure scalability, the newly introduced parameters are initialized to zero, ensuring that the added qubit is only activated if it improves the model's performance. This approach allows for a smooth transition to multi-qubit models.

IV. HARDWARE-ADAPTED QML

One of the primary challenges facing VQC is the barren plateau problem, which renders these models untrainable as the number of qubits increases [28, 29]. Furthermore, achieving sufficient expressivity typically requires a large number of layers [27], which is impractical in the NISQ era, as each additional gate introduces noise to the model. Building these models at the pulse level offers a promising approach to addressing both issues. Recent studies have shown that minimizing the fidelity of a target unitary using gradient-based parameterization through pulses can effectively circumvent barren plateaus [30]. This provides a theoretical basis for the accurate training results previously observed experimentally during gate calibration. Additionally, pulse-level optimization enhances expressivity, enabling access to a broader range of frequencies with greater precision.

A. Parameterized Pulse Circuits

Our aim is to adapt the data re-uploading QNN model to the pulse programming level. To this end, we propose creating parameterized pulsed circuits instead of parameterized gates. Analogously to single-qubit rotations being characterized by three Euler angles, a general one-qubit transformation can be physically characterized by a pulse, described in terms of parameters such as its shape, amplitude, duration, phase, and frequency. Since an SU(2) rotation has three degrees of freedom, the five pulse parameters are not all independent and capable of producing distinct rotations—for instance, with a constant pulse shape, a pulse with amplitude Ω and duration τ has the same effect as one with amplitude $\Omega/2$ and duration 2τ. Although these parameters are redundant in terms of achieving the same rotation, they can be optimized for other purposes, such as enhancing the physical robustness of the pulse or reducing noise [31]. However, such optimizations are beyond the scope of this discussion.

Training all these parameters does not improve expressivity but increases computational complexity. Therefore, we choose to leave the pulse shape and duration as fixed parameters. The remaining trainable parameters are the pulse phase (primarily responsible for selecting the axis rotation in the XY plane), the pulse amplitude (which represents the rotation angle) and the pulse frequency (which plays a more complex role, contributing to Z-rotations, qubit entanglement, and other effects [19]). In total, we have three independent parameters for the three degrees of freedom that a SU(2) rotation has.

The gate replacement is illustrated in Fig. **4**. As Z-rotations are free in virtue of the VZ gates, we can add as much as desired without negative impacts, so we will leave these rotations. However, it is not necessary to restrict the middle rotation to the Y axis, but we can leverage of Eq. (5) with $\Delta_\omega = 0$ to have a more general rotation that combines X and Y components [32]. It is important to note that this does not enhance expressivity, as it remains an arbitrary SU(2) transformation as before. Nonetheless, it may facilitate faster or more effective learning in the model. In summary, the parameterized one-qubit pulse consists of 4 parameters: two Z-rotation angles, the pulse phase and pulse amplitude.

Regarding the two-qubit gates, the original model did not focus on the practical implementation of two-qubit gates in real devices but instead used general controlled SU(2) rotations. When converted to pulses, this approach would require multiple pulses, which could negatively impact experimental performance due to errors from gate-to-pulse noise [33] and increased execution times, leading to higher decoherence. For instance, a simple CNOT gate requires two pulses, as shown in Fig. **2**b. To adopt a more realistic model, we propose using a single pulse to successfully entangle the qubits, inspired by the CR gate. Specifically, we target the first qubit added to the model and apply a pulse tuned to the natural frequency of the newly introduced qubit. The pulse's amplitude and

phase are treated as learnable parameters. However, instead of limiting the operation to a pure CR gate, we introduce an additional parameter that offsets the pulse frequency slightly from the control qubit's frequency. This offset allows the model to explore alternative forms of entanglement beyond a pure XZ-rotation. It also provides the flexibility to reduce or even eliminate entanglement if it is not beneficial for the task.

As a result, each two-qubit interaction is characterized by three learnable parameters of the pulse: the amplitude, the phase, and the frequency detuning. This setup provides flexibility in tailoring interactions between qubits, allowing the model to optimize both the degree and type of entanglement required for the problem at hand.

B. Pulse encoding

A key aspect not yet addressed in the literature is how to perform data encoding when focusing on driving pulses. For three-dimensional data, one possible approach, as suggested in the original model, is to apply an arbitrary SU(2) rotation whose Euler angles are determined by the Cartesian coordinates of the data points. If $\vec{x} = (x_1, x_2, x_3)$ denotes the point, this encoding is realized in hardware by applying a VZ rotation with angle x_1, followed by a pulse with a fixed phase of $\pi/2$ and amplitude determined by x_2, and another VZ rotation with angle x_3. Thus, in virtue of VZ rotations, this encoding is hardware-efficient.

However, we now propose a novel method that utilizes the flexibility of pulse implementation, enabling the definition of more customized operations, which may enhance the efficiency of the encoding process. This strategy aims to position each data point on the Bloch sphere in a way that reflects its original location in the dataset. Given a point represented by its spherical coordinates (r, θ, φ), the encoding process begins by applying a pulse whose phase is determined by the azimuthal angle φ, positioning the point at the correct latitude. The amplitude of the pulse is set by the radial coordinate r, ensuring that the largest value of r corresponds to the maximum latitude (π) on the Bloch Sphere. Finally, the longitude is adjusted using a VZ rotation with an angle of θ. This process is depicted in Fig. 5.

C. Further modifications

After defining the core protocol of PQML, we introduce additional modifications to the original QNN model to improve performance, even though they are not directly related to quantum control. In the original design, class label states were fixed in the computational basis, restricting the decision hyperplane to $\langle \hat{\sigma}^z \rangle = 0$ and limiting the QNN to rotating points on the Bloch sphere for class separation. We propose parametrizing this decision hyperplane, where class label states are defined by general qubit states: $|s_0(\theta, \varphi)\rangle = \cos(\theta)|0\rangle + e^{i\varphi}\sin(\theta)$ and $|s_1(\theta, \varphi)\rangle = \left|s_0^{\perp}(\theta, \varphi)\right\rangle = -\sin(\theta)|0\rangle + e^{i\varphi}\cos(\theta)$, with θ and φ as learnable parameters. This enables the QNN to not only rotate points during training but also adapt the decision boundary, potentially reducing the training steps for convergence.

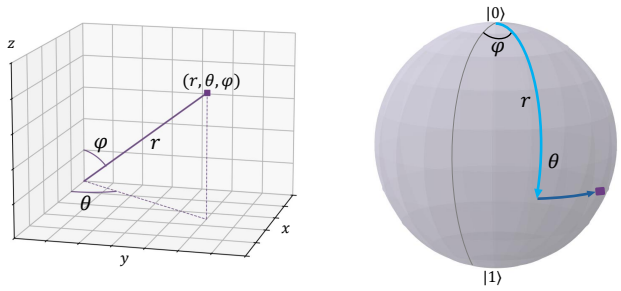

FIG. 5. Illustration of the process for encoding a single point using the new pulse-based approach starting at $|0\rangle$. The first step, depicted in light blue, involves placing the point at the correct latitude on the Bloch Sphere. The longitude is then set, as shown by the dark blue arrow.

To enhance the encoding for the gate-based model, we rescale the dataset to fit within the cube $[-\pi, \pi]^3 \subset \mathbb{R}^3$ and encode it using Euler angles for the rescaled points. This encoding is independent of the dataset's original scale, making the method more flexible for different data structures. These scaled points are also used for pulse encoding to ensure a fair comparison of results.

V. RESULTS

This section validates the performance of the proposed model through numerical simulations, employing a specific architecture and parameters of a selected quantum device. The device's evolution is simulated under the influence of tailored control pulses, demonstrating the model's ability to adapt to hardware-specific constraints while maintaining strong performance.

A. Experimental setup

To study the proposed model performance, we remained aligned with our primary goal, creating QML models tailored for implementation on real hardware. For this purpose, we simulated IBM's superconducting quantum computer `ibm_brisbane` [35]. In this regard, simulating qubits, particularly their exact evolution under the interaction with microwave pulses, is computationally demanding [36]. Therefore, we restricted our study to the case of one and two qubits. In particular, we selected the qubits indexed as 1 and 2 in the Brisbane topology. Their parameters, including T_1, T_2, and qubit frequencies, are listed in Table I.

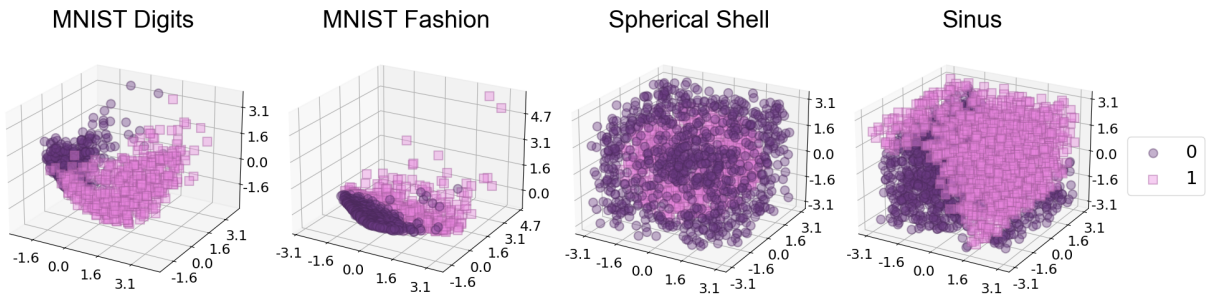

FIG. 6. Datasets for binary classification [34], including the MNIST dataset (classifying digits 0 vs 8), the Fashion MNIST dataset (distinguishing dresses vs shirts), and two synthetic datasets: one representing a spherical shell and the other a boundary defined by the sine function. The MNIST datasets are reduced to three dimensions using classical PCA techniques.

Qubit	T1 (us)	T2 (us)	Frequency (GHz)	Anharmonicity (GHz)	1-Qubit Gate Time (ns)	Coupling (GHz)	2-Qubit Gate Time (ns)
1	180	180	4.8	-0.31	300	0.013	660
2	310	250	4.6	-0.31	300		

Table I. Brisbane qubit 1 and 2 statistics.

Regarding the exact form of the pulse shape, we chose a constant amplitude shape, so the total pulse takes the form $\Omega \sin(\omega_d t + \gamma)$. With this choice, the angle of a one-qubit rotation is directly given by ΩT, where T represents the pulse duration.

B. Datasets and models

The datasets used in this study were selected to reflect a range of challenges and include both real-world and synthetic examples, all designed for binary classification. From the MNIST Fashion dataset, we considered a task distinguishing dresses (label 3) from shirts (label 6), while the MNIST Handwritten Digits dataset focused on classifying digits 8 and 0. Complementing these, we included synthetic datasets inspired by the original model's article, specifically the sinusoidal and spherical shell datasets, which are also binary. These datasets are shown in Fig. 6. To ensure compatibility with the quantum embedding, the MNIST datasets were reduced to three dimensions using classical PCA.

We compare the new PQML model with the traditional gate-based approach across various datasets, evaluating train and test accuracies for three models: the original gate-based QNN (GateQNN), the pulse-based model with the original Euler angle encoding (MixedQNN), and the model that integrates both the new encoding and the proposed parameterized pulse architecture (PulsedQNN). This comparison enables us to assess the performance of the parameterized pulse model relative to the traditional gate-based approach by comparing GateQNN with MixedQNN, and to evaluate the effectiveness of the new pulse encoding by comparing MixedQNN with PulsedQNN.

C. Encoding

First, to compare the pulse-based encoding with the gate-based encoding, we apply an encoding step to the datasets and analyze how they are encoded in the Bloch sphere. The results are shown in Fig. 7. As observed, encoding a dataset with spherical symmetry, such as the spherical shell, highlights the perfect alignment achieved by the pulse-based encoding method, as the dataset is perfectly organized within the Bloch sphere. In contrast, the gate-based encoding produces chaotic results, destroying the original structure of the dataset. For a general dataset like the MNIST Fashion dataset, the newly introduced encoding clearly preserves the dataset's structure, while the gate-based encoding results appear less organized. This refined encoding enables the model to effectively study the Fourier components of the data, leading to better classification results.

D. Model performance

To conclude this study, the performance of the models is evaluated across the different datasets. For each dataset, 500 training points and 250 test points are randomly sampled, using five different seeds to ensure robustness. The three models are trained using architectures with one to eight layers, employing both one-qubit and two-qubit architectures. These configurations ensure that the total time required to execute all pulses remains within the decoherence times of the physical device, T_1 and T_2 as listed in Table I. Classification performance is assessed by comparing the mean and standard deviation of

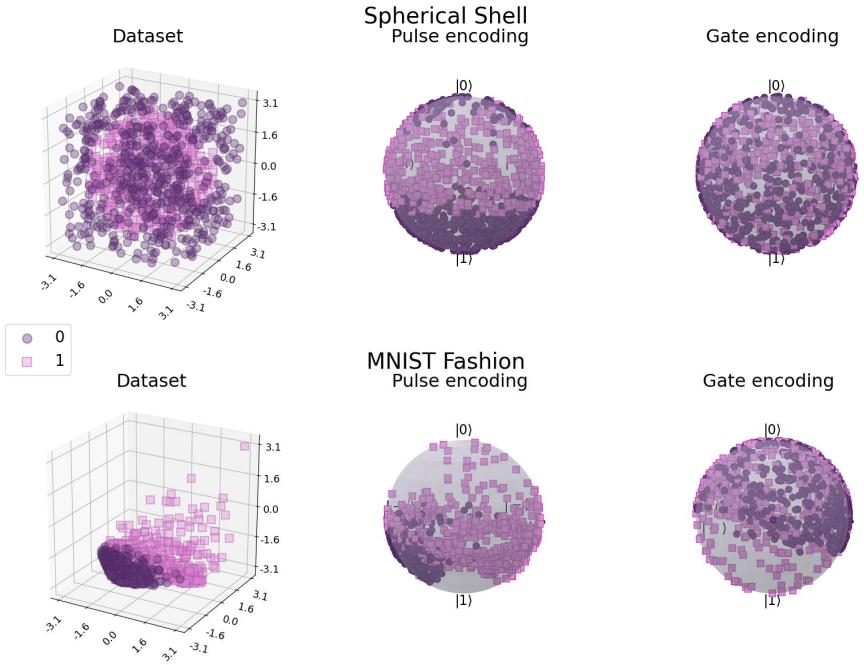

FIG. 7. Encoding proposed for the shell and corners dataset compared to the original model's encoding. As depicted, the new encoding aims to map the spherical coordinates of the dataset into the Bloch Sphere.

training and test accuracies. The results for the various datasets are presented in Fig. **8**.

Regarding the new pulsed encoding, the PulsedQNN model outperforms other models across all datasets, except for the MNIST fashion dataset—where precision remains comparable. Overall, this encoding appears particularly effective in capturing complex patterns and structures in the data, leading to improved generalization. The consistent performance across different datasets suggests that the pulsed encoding is robust and adaptable, especially for more complex feature spaces, such as those in the synthetic datasets.

Turning to the results for the MixedQNN model, the final accuracy results show that its performance is essentially comparable to the original gate-based model. However, the gate-based model has greater access to a richer two-qubit interaction space, as it can perform any controlled rotation for two-qubit interactions. In contrast, the new parameterized pulse strategy is constrained to single pulses, with the main entangling interaction term corresponding to the CR gate (i.e., a *XZ*-multirotation). Despite these limitations, the results underscore the feasibility of hardware-adapted QML, as the performance of the GateQNN and MixedQNN models, that use the same encoding strategy, is very similar .

It is also crucial to consider other limitations in the practical implementation of the gate-based model. This model requires prior calibration of the gates using QOC, which can add complexity and introduce noise from imperfectly calibrated gates. Additionally, experimental noise and imperfections can accumulate from the execution of multiple pulses, and the increased time required to execute these pulses may lead to greater decoherence. On the other hand, the new pulse-parameterized model directly uses custom-designed pulses, bypassing the need for this calibration or transpilation into multiple pulses. This highlights the practical advantages of parameterized pulse models.

Additional numerical results, along with the simulation source code, are available in this GitHub repository: Hardware-Adapted-Quantum-Machine-Learning [34], and in the *Supplementary materials* document [37].

Finally, it is important to highlight that this QNN model can be leveraged as a kernel method within a embedding quantum kernel (EQK) framework, as the original reference shows [10]. By doing so, the model not only enhances the accuracy of quantum machine learning tasks but also enables more expressive feature mappings.

VI. CONCLUSIONS

In this work, we presented Pulsed Quantum Machine Learning (PQML), a new pulse-based model designed for practical experimental implementation of QML models, along with a novel pulse-based encoding strategy. This approach leverages the unique advantages of pulse control in quantum systems,

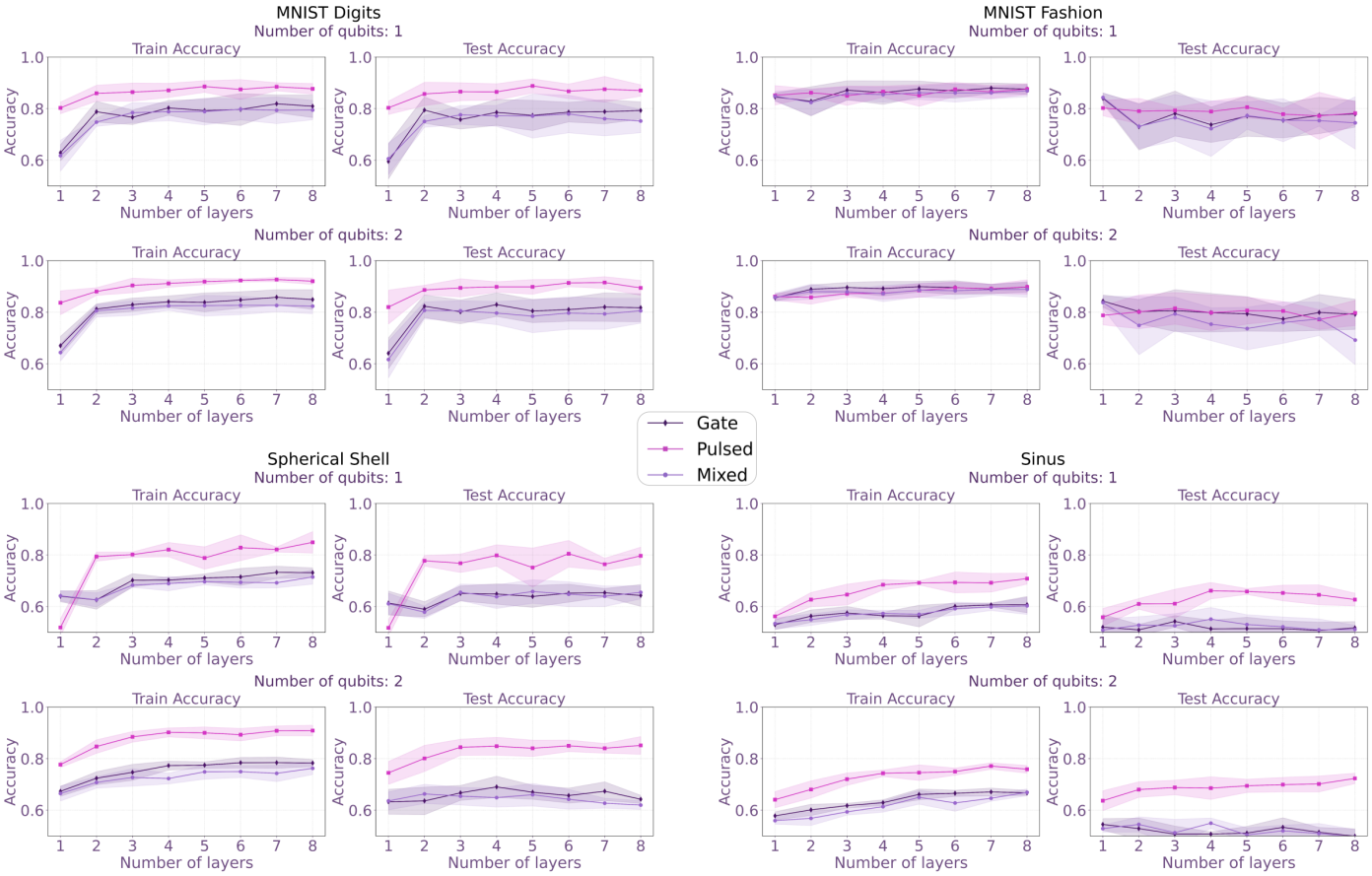

FIG. 8. Performance comparison of the newly proposed pulsed QNN models: traditional encoding (MixedQNN, lavender) and new pulse encoding (Pulse QNN, pink) against the original gate-based model (GateQNN, dark violet). The comparison considers variations in the number of qubits and layers across four datasets.

offering a potential pathway for overcoming some of the limitations faced by traditional gate-based models. We conducted simulated experiments on a variety of datasets and compared the results with a general gate-based model. The numerical results suggest that pulse-based methods may be better suited for real-world implementation due to their enhanced compatibility with the inherent physical constraints and operational limitations of quantum hardware. The results also show that the experimental implementation of this approach holds promise, as it is capable of achieving competitive accuracy levels.

Furthermore, the proposed encoding surpasses the gate-based approach by providing a more versatile mapping of data to quantum states. Although it does not reduce the number of operations, the use of pulses facilitates custom-defined mappings, enabling the design of an intuitive encoding that effectively represents the underlying data structure on the Bloch sphere.

The results highlight exciting opportunities that arise with increasing model complexity, presenting new challenges that can drive innovation. Simulating pulse dynamics on classical hardware, while computationally intensive, inspires the development of more efficient optimization strategies and advanced computational techniques. This complexity motivates the exploration of novel approaches beyond traditional gradient-based strategies, as classical simulations provide opportunities to refine methods that better capture pulse behavior. These challenges open the door to enhancing parameter updates and discovering innovative pathways toward convergence to optimal solutions, pushing the boundaries of what can be achieved in this domain.

Finally, it is worth highlighting that we applied the PQML methodology to a specific model, the data re-uploading model, as a proof of concept. However, the potential of this methodology extends far beyond this example. Similar parameterized-pulse approaches can be employed in any QML context, particularly in VQCs, where the fine-grained control over pulse-level parameters offers new avenues for optimizing performance and adapting to hardware constraints.

A. Futher work

Future work will focus on further exploring the advantages of pulse-based models, especially their potential to mitigate decoherence and reduce noise in real quantum hardware. To gain a deeper understanding of their behavior under realistic conditions, we will conduct numerical simulations that include noise, followed by experiments on real quantum devices to assess the results. Additionally, in this study, we have overlooked errors related to measurement, which may be taken into account in future work. These studies will provide valuable insights into the robustness and performance of pulse-based models compared to gate-based ones, highlighting their practical applicability and effectiveness in addressing the challenges of quantum hardware.

A more thorough theoretical investigation of the encoding strategies used in pulse-based models will also be undertaken. This will involve extending the analysis to more general scenarios, such as encoding information into multi-qubit states, as well as exploring related approaches that leverage the extended expressivity of pulse-based methods. These efforts aim to uncover novel ways to enhance the flexibility and applicability of pulse-based quantum machine learning.

REFERENCES

[1] S. S. Gill et al. "Quantum Computing: Vision and Challenges". arXiv:2403.02240 (2024).

[2] McKinsey & Company. Steady progress in approaching the quantum advantage. Accessed on: 2025-01-16. (2024).

[3] R. Acharya et al. "Quantum error correction below the surface code threshold". *Nature* (2024).

[4] J. Preskill. "Quantum Computing in the NISQ era and beyond". *Quantum* **2** (2018).

[5] J. Yamaguchi et al. "Estimation of Shor's Circuit for 2048-bit Integers based on Quantum Simulator" (2023).

[6] J. Biamonte et al. "Quantum machine learning". *Nature* **549.7671** (2017).

[7] Y. Gujju et al. "Quantum machine learning on near-term quantum devices: Current state of supervised and unsupervised techniques for real-world applications". *Phys. Rev. Appl.* **21** (2024).

[8] Z. Liang et al. "NAPA: Intermediate-level Variational Native-pulse Ansatz for Variational Quantum Algorithms". arXiv:2208.01215 (2024).

[9] David C. McKay et al. "Efficient *Z* gates for quantum computing". *Phys. Rev. A* **96** (2017).

[10] P. Rodriguez-Grasa et al. "Training embedding quantum kernels with data re-uploading quantum neural networks". arXiv:2401.04642 (2024).

[11] D. D'Alessandro. *Introduction to Quantum Control and Dynamics*. 1st. Chapman and Hall/CRC, (2007).

[12] T. Alexander et al. "Qiskit pulse: programming quantum computers through the cloud with pulses". *Quantum Science and Technology* **5.4** (2020).

[13] F. Arute et al. "Quantum Supremacy using a Programmable Superconducting Processor". *Nature* **574** (2019).

[14] IBM Corporation. IBM Unveils Breakthrough 127-Qubit Quantum Processor. Accessed on: 2024-12-25.

[15] Jens Koch et al. "Charge-insensitive qubit design derived from the Cooper pair box". *Physical Review A* **76.4** (2007).

[16] A. Blais et al. "Circuit quantum electrodynamics". *Reviews of Modern Physics* **93.2** (2021).

[17] M. J. Peterer et al. "Coherence and Decay of Higher Energy Levels of a Superconducting Transmon Qubit". *Physical Review Letters* **114.1** (2015).

[18] P. Krantz et al. "A quantum engineer's guide to superconducting qubits". *Applied Physics Reviews* **6.2** (2019).

[19] Ken Xuan Wei et al. "Characterizing non-Markovian off-resonant errors in quantum gates". *Physical Review Applied* **21.2** (2024).

[20] D. C. McKay et al. "Efficient Z gates for quantum computing". *Physical Review A* **96.2** (2017).

[21] E. Magesan and Jay M. Gambetta. "Effective Hamiltonian models of the cross-resonance gate". *Physical Review A* **101.5** (2020).

[22] Jerry M. Chow et al. "Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits". *Phys. Rev. Lett.* **107** (2011).

[23] The process can be mathematically understood by performing a Schrieffer-Wolff transformation on the Hamiltonian (8) to eliminate the interaction between qubits.

[24] M. et al. Schuld. *Machine Learning with Quantum Computers*. Quantum Science and Technology. Cham: Springer, (2021).

[25] M. Cerezo et al. "Variational quantum algorithms". *Nature Reviews Physics* **3.9** (2021).

[26] Adrián Pérez-Salinas et al. "Data re-uploading for a universal quantum classifier". *Quantum* **4** (2020).

[27] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. "Effect of data encoding on the expressive power of variational quantum-machine-learning models". *Phys. Rev. A* **103** (2021).

[28] J. R. McClean et al. "Barren plateaus in quantum neural network training landscapes". *Nature Communications* **9.1** (2018).

[29] P. Easom-Mccaldin et al. "On Depth, Robustness and Performance Using the Data Re-Uploading Single-Qubit Classifier". *IEEE Access* **9** (2021).

[30] H. Tao el al. "On the Role of Controllability in Pulse-based Quantum Machine Learning Models". arXiv:2405.09135 (2024).

[31] K. et al. Williams. "Quantification of Robustness, Leakage, and Seepage for Composite and Adiabatic Gates on Modern NISQ Systems". *Intelligent Computing* **3** (2024).

[32] G. T. Genov et al. "Efficient and robust signal sensing by sequences of adiabatic chirped pulses". *Phys. Rev. Res.* **2** (2020).

[33] J Gil-Lopez et al. "Improved non-linear devices for quantum applications". *New Journal of Physics* **23.6** (2021). URL: https://dx.doi.org/10.1088/1367-2630/ac09fd.

[34] I.B. Acedo. Hardware-Adapted Quantum Machine Learning GitHub Repository. *GitHub*. Accessed on: 2025-01-22.

[35] IBM Corporation. Brisbane Quantum Computer Description. Accessed on: 2024-12-12.

[36] On the code implementation, pulses are simulated using the Trotter-Suzuki approximation, with each pulse divided into dozens of gates. The gradient for each gate is computed during training, making even a simple QNN computationally intensive.

[37] I.B. Acedo. Supplementary Materials. Hardware-Adapted Quantum Machine Learning. *GitHub*. Accessed on: 2025-01-22.