Learning to Program Quantum Measurements for Machine Learning

Samuel Yen-Chi Chen
Wells Fargo
New York NY, USA
yen-chi.chen@wellsfargo.com

Huan-Hsin Tseng
AI Department
Brookhaven National Laboratory
Upton NY, USA
htseng@bnl.gov

Hsin-Yi Lin
Department of Mathematics
and Computer Science
Seton Hall University
South Orange NJ, USA
hsinyi.lin@shu.edu

Shinjae Yoo
AI Department
Brookhaven National Laboratory
Upton NY, USA
syjoo@bnl.gov

Abstract—The rapid advancements in quantum computing (QC) and machine learning (ML) have sparked significant interest, driving extensive exploration of quantum machine learning (QML) algorithms to address a wide range of complex challenges. The development of high-performance QML models requires expert-level expertise, presenting a key challenge to the widespread adoption of QML. Critical obstacles include the design of effective data encoding strategies and parameterized quantum circuits, both of which are vital for the performance of QML models. Furthermore, the measurement process is often neglected-most existing QML models employ predefined measurement schemes that may not align with the specific requirements of the targeted problem. We propose an innovative framework that renders the observable of a quantum system—specifically, the Hermitian matrix-trainable. This approach employs an end-to-end differentiable learning framework, enabling simultaneous optimization of the neural network used to program the parameterized observables and the standard quantum circuit parameters. Notably, the quantum observable parameters are dynamically programmed by the neural network, allowing the observables to adapt in real time based on the input data stream. Through numerical simulations, we demonstrate that the proposed method effectively programs observables dynamically within variational quantum circuits, achieving superior results compared to existing approaches. Notably, it delivers enhanced performance metrics, such as higher classification accuracy, thereby significantly improving the overall effectiveness of QML models.

Index Terms—Quantum neural networks, Variational quantum circuits, Quantum architecture search, Learning to learn, Quantum measurements

I. INTRODUCTION

Quantum computing (QC) promises computational capabilities that surpass classical computing for a range of intractable problems [1]. In parallel, recent advances in artificial intelligence (AI) and machine learning (ML) have enabled intelligent agents to achieve remarkable performance in domains

such as sequential decision-making [2] and natural language processing [3]. Motivated by the complementary strengths of these two paradigms, a growing body of research has focused on hybrid quantum-classical frameworks aimed at harnessing the best of both worlds. While today's quantum hardware has yet to realize large-scale, fault-tolerant quantum algorithms, hybrid frameworks such as variational quantum algorithms (VQAs) provide a practical bridge for near-term applications [4], [5].

VQAs represent a hybrid computing paradigm in which computational tasks are decomposed into subcomponents delegated to either quantum or classical processors, depending on their respective strengths. Quantum subroutines are typically implemented using variational quantum circuits (VQCs), while classical routines handle gradient computation and optimization. VQAs have demonstrated effectiveness in applications such as quantum chemistry [6] and combinatorial optimization [7].

Building on this foundation, VQCs have been widely adopted as learnable modules in quantum machine learning (QML), enabling end-to-end training pipelines where VQC parameters are updated using classical gradient methods [8], [9]. VQC-based QML has shown promise across diverse ML tasks, including classification [8]–[13], sequence modeling and regression [14], [15], generative modeling [16], natural language processing [17]–[20], model compression and quantum distillation [21]–[29], and reinforcement learning [30]–[40].

Despite this progress, most quantum neural networks (QNNs) are still trained using fixed observables $\hat{B}_k$—typically the Pauli-X, Y, or Z operators—whose spectral range is limited to $\lambda = \pm 1$. As a result, the output of a VQC is confined to $\langle \psi | H | \psi \rangle \in [-1, 1]$, regardless of the specific parameterization of the quantum circuit $U(\vec{x}), W(\Theta)$. According to the Rayleigh quotient, we know that $\lambda_{\min} \leq \langle \psi | H | \psi \rangle \leq \lambda_{\max}$ for any normalized state $|\psi\rangle$, suggesting that increasing the spectral range of the observable can expand the expressivity of the VQC—a desirable property for tasks such as classification and regression.

Prior work has demonstrated that parameterizing and training the observable-represented as a Hermitian matrix-alongside the VQC parameters can yield substantial gains over

fixed measurements [41]. However, in this setting, the observable is static during inference, which limits its adaptability. While this is more flexible than fixed Pauli observables, it still cannot adjust to data-specific measurement needs.

In this paper, we propose a dynamic framework in which an external neural controller, referred to as the slow programmer, is trained to generate both the VQC rotation parameters and the observable (Hermitian matrix) on a per-input basis, as illustrated in Figure 1. This transforms the VQC into a data-reactive quantum model whose measurement scheme adapts to the input distribution. Numerical simulations across various classification benchmarks show that the proposed method consistently outperforms existing alternatives, offering both higher accuracy and improved training stability.
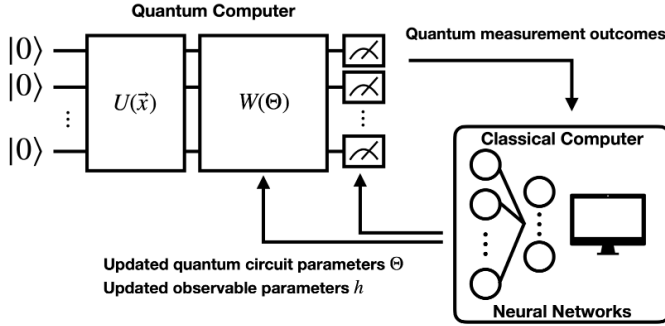

Fig. 1. **Hybrid quantum-classical computing with learnable observable programmed by the neural network.**

II. RELATED WORK

Prior work has demonstrated that learning the quantum observable, represented as a parameterized Hermitian matrix, can significantly enhance the performance of variational quantum circuits (VQCs). In particular, [41] showed that making the observable trainable leads to substantial gains over fixed choices such as the Pauli-Z matrix. Moreover, using separate optimizers for circuit parameters and observable parameters further improves convergence stability and final accuracy. This idea is rooted in the broader trend of parameterizing and optimizing more components of a quantum model beyond rotation angles. One prominent direction is differentiable quantum architecture search (DiffQAS) [42]–[44], which adapts the classical differentiable architecture framework [45] to the quantum domain by assigning trainable weights to candidate operations, allowing joint structure-parameter optimization. Beyond direct gradient-based optimization, the Fast Weight Programmer (FWP) framework [46] introduces a meta-learning mechanism in which a "slow" neural controller dynamically generates parameters for a "fast" neural actor. This paradigm has been extended to quantum settings in Quantum FWP (QFWP) [29], [47], where a classical controller network generates data-dependent parameters for a VQC. The present work advances this line by enabling the slow neural controller to program not only the quantum circuit parameters but also the measurement observable. This transforms the entire hybrid quantum-classical model into a data-driven, reactive system. Additionally, unlike previous works where the observable is fixed after training, our framework supports on-the-fly observable adaptation at inference time. Together, these extensions enable a more expressive, flexible, and dynamically reprogrammable quantum learning architecture.

III. QUANTUM NEURAL NETWORKS

A typical variational quantum circuit (VQC) comprises three primary components: an encoding circuit $U(\vec{x})$ that maps classical input data into quantum states, a parameterized variational circuit $W(\Theta)$ consisting of trainable quantum gates, and a final quantum measurement step using one or more observables $\hat{B}_k$. The encoding circuit prepares the initial state $U(\vec{x})|0\rangle^{\otimes n}$, where $\vec{x}$ is the input feature vector and n is the number of qubits. This encoded state is then transformed by the variational circuit, yielding the final quantum state

$$|\Psi\rangle = W(\Theta)U(\vec{x})|0\rangle^{\otimes n}, \tag{1}$$

where the variational circuit is composed of M layers of trainable unitary blocks:

$$W(\Theta) = \prod_{j=M}^{1} W_j(\vec{\theta}_j), \quad \Theta = \{\vec{\theta_1} \cdots \vec{\theta_M}\}. \tag{2}$$

Information is extracted from the quantum system via measurement of Hermitian operators $\hat{B}_k$, yielding expectation values $\langle\hat{B}_k\rangle = \langle\Psi|\hat{B}_k|\Psi\rangle$. In practice, this expectation can be evaluated through repeated measurements on real quantum hardware (sampling over multiple shots), or computed directly in simulation. The VQC can thus be interpreted as a quantum function $\vec{f}(\vec{x};\Theta) = (\langle\hat{B}_1\rangle,\ldots,\langle\hat{B}_K\rangle)$, where the choice of observable $\hat{B}_k$ determines how quantum information is projected back into the classical domain. As we will discuss, replacing fixed observables with trainable or input-conditioned operators opens up new expressive capabilities for quantum models.
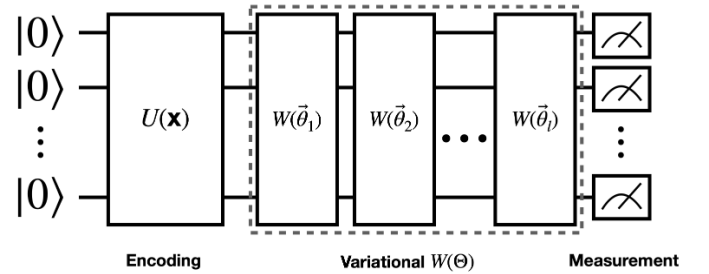

Fig. 2. **Basic structure of a variational quantum circuit (VQC).**

IV. PROGRAMMABLE QUANTUM MEASUREMENTS

A typical assumption in machine learning is that data reside on a differentiable manifold [48], [49]. When classical data are transformed to quantum states, their geometric structure are mapped to the Hilbert space of the quantum domain accordingly. To extract the intrinsic structure of data in the

quantum setting, we introduce a dynamically evolving inner product defined via a fibration in an extended geometric space. This formulation considers the space of quantum states as a complex vector bundle, where the metric structure adapts in response to variations in the underlying data manifold. This perspective naturally aligns with gauge-theoretic principles [50], [51], wherein the choice of fiber structure and the corresponding connection provide insights into quantum embeddings and their evolution.

The mathematical description of our method is as follows. Consider a set of classical data $\mathcal{D} = \{(x_i, y_i) \in \mathbb{R}^k \times \mathbb{R}^m\}$ distributed on a (differentiable) manifold M. To conduct QML, the classical data are embedded into quantum states $|\psi_{x_i}\rangle$ of n-qubit Hilbert space $\mathbb{C}^N$ with $N = 2^n$. This "classical-quantum" conversion can be resulted from a (smooth) *section* from M to a complex vector bundle (E, π) of rank N with $\pi : E \to M$ the projection (see Figure 3) [52]. Denote a bundle chart of a point $x \in M$ by (φ, W) with $x \in W \subseteq M$, then the fiber $E_x := \pi^{-1}(x)$ is isomorphic to a usual n-qubit space $\mathbb{C}^N$. Together with a local section $s : W \to E$, each sample $x \in M$ is mapped to a quantum state $s(x)$ (or $|s(x)\rangle$) that has its own qubit space $\{x\} \times \mathbb{C}^N$.

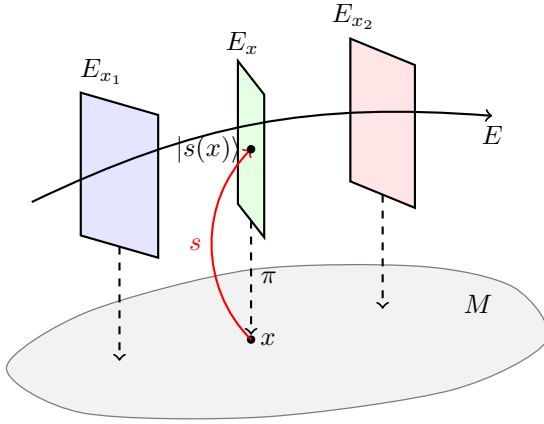Fig. 3. Our Learning to Program Quantum Measurements is described by a vector bundle (E, π, M) with $\pi : E \to M$ the bundle projection. On each classical data point $x \in M$, the living Hilbert space of quantum states is given by $E_x = \pi^{-1}(x) \cong \mathbb{C}^N$ along with a Hermitian inner product $h(x)$ on E_x smoothly changing with respect to x. A local section $s : W \subseteq M \to E$ encodes (converts) $x \in M$ into a quantum state $|s(x)\rangle \in E_x$. The inner product $h(x)(|s(x)\rangle, |s(x)\rangle)$ induces a Hermitian operator $H(x)$ via $\langle s(x)| H(x) |s(x)\rangle$.

To perform a quantum measurement, a bundle metric is required on (E, π).

Definition 1. *Let $E \xrightarrow{\pi} M$ be a complex vector bundle and $\overline{E}$ be the complex conjugate vector bundle defined by $\alpha * v = \overline{\alpha} \cdot v$ for $\alpha \in \mathbb{C}$, $v \in \pi^{-1}(x)$, $x \in M$. Then a **Hermitian metric** is a smooth section $h : M \to E \otimes \overline{E}^*$ such that*

$$h(x)(v, w) = \overline{h(x)(w, v)} \qquad (\text{for all } v, w \in E_x)$$
$$h(x)(v, v) > 0 \qquad (\text{for any } 0 \neq v \in E_x) \tag{3}$$

where $\overline{E}^*$ is the dual bundle of $\overline{E}$. Endowing a bundle metric on a complex vector bundle allows us to measure each quantum state *dynamically*, depending on each classical
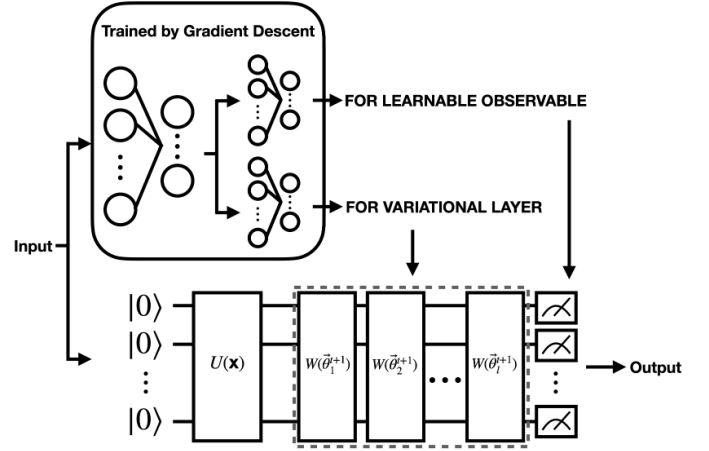

Fig. 4. **Schematic of the proposed framework integrating fast weight programmers (FWP) with variational quantum circuits (VQCs).** An external neural network controller, trained via gradient descent, takes the classical input and generates two sets of parameters: one for the variational layers of the VQC and another for the learnable quantum observable. The input is encoded via $U(\vec{x})$ and processed through trainable quantum layers $W(\vec{\theta}_j)$ with dynamically generated parameters. The final measurement is performed using a data-conditioned observable, enabling fully input-adaptive quantum inference.

sample $x \in M$. Indeed, given a frame field (basis) $\{e_j\}_{j=1}^N$ in E, from (3) we see

$$h(x)(e_j, e_k) = \overline{h(x)(e_k, e_j)} \tag{4}$$

which yields a Hermitian matrix $H_{jk} = \overline{H}_{kj}$ at each sample $x \in M$ by denoting $H_{jk} = h(x)(e_j, e_k)$. Conversely, whenever a smooth mapping $x \in M \mapsto H(x)$ with values in Hermitian matrices is given, we can define

$$\widetilde{h}(x)(v, w) := \sum_{j,k} \overline{v_j}\, w_k \cdot H_{jk}(x) \tag{5}$$

Then one verifies that $\widetilde{h}$ is a Hermitian metric satisfying (3). The scheme by (5) is proposed as our "*Learning to Program Quantum Measurements*".

Consequently, a quantum state $|\psi_x\rangle \in E_x$ corresponding to $x \in M$ has measurement value $h(x)(\psi_x, \psi_x)$ by (3), or $\langle \psi_x| H |\psi_x\rangle$ in Dirac notations.

In this study, a classical neural network is used to create a smooth map from $x \mapsto H(x)$ of a Hermitian matrix, *i.e.,* an observable. However, since Hermitian matrices do not form a Lie group, the following parametrization method is constructed to ensure neural network predictions are Hermitian matrices.

Definition 2. *A Hermitian matrix B is used as an* observable *in quantum mechanics, for it yields real-valued expectations $\langle \Psi| B |\Psi\rangle$ under a wave function $|\Psi\rangle$. The condition $B = B^\dagger$ requires each matrix element $b_{ij} = \overline{b_{ji}} \in \mathbb{C}$. Consequently, a Hermitian matrix can be generated by $2 \times \frac{N(N-1)}{2} + N = N^2$*

real parameters,

$$B = \begin{pmatrix} d_{11} & a_{12} + ic_{12} & a_{13} + ic_{13} & \cdots & a_{1N} + ic_{1N} \\ * & d_{22} & a_{23} + ic_{23} & \cdots & a_{2N} + ic_{2N} \\ * & * & d_{33} & \cdots & a_{3N} + ic_{3N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ * & * & * & \cdots & d_{NN} \end{pmatrix}$$

where $*$ denotes the corresponding complex conjugate and a_{ij}, c_{ij}, d_{ii} are arbitrary real numbers.

The Hermitian matrix can be initialized randomly and optimized iteratively via gradient-based methods. To elucidate the process, we can write the Hermitian matrix of a n-qubit system with $N = 2^n$ as the parametrization of coefficient $\vec{b} = (b_{11}, \ldots, b_{NN}) \in \mathbb{C}^{N \times N}$ as $B(\vec{b}) = \sum_{i=1}^{N} \sum_{j=1}^{N} b_{ij} E_{ij}$, where $b_{ij} = \overline{b_{ji}}$ and E_{ij} as the indicating matrix with only one non-zero at entry (i, j),

$$E_{ij} = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 0 \end{pmatrix} \tag{6}$$

Given a quantum state $|\Psi\rangle$ (e.g. the one shown in Equation 1), the expectation value with the $B(\vec{b})$ can be written as,

$$\langle \Psi | B(\vec{b}) | \Psi \rangle = \langle \Psi | \sum_{i}^{N} \sum_{j}^{N} b_{ij} E_{ij} | \Psi \rangle \tag{7}$$

$$= \sum_{i}^{N} \sum_{j}^{N} b_{ij} \langle \Psi | E_{ij} | \Psi \rangle \tag{8}$$

The total differential $\langle \Psi | B(\vec{b}) | \Psi \rangle$ is

$$\nabla \langle \Psi | B(\vec{b}) | \Psi \rangle = \begin{pmatrix} \nabla_\theta \langle \Psi | B(\vec{b}) | \Psi \rangle \\ \nabla_{\vec{b}} \langle \Psi | B(\vec{b}) | \Psi \rangle \end{pmatrix} \tag{9}$$

where the differentiation of $\langle \Psi | B(\vec{b}) | \Psi \rangle$ can be written as,

$$\frac{\partial \langle \Psi | B(\vec{b}) | \Psi \rangle}{\partial b_{k\ell}} = \frac{\partial \langle 0 | U(\vec{x})^\dagger W(\Theta)^\dagger B(\vec{h}) W(\Theta) U(\vec{x}) | 0 \rangle}{\partial b_{k\ell}}$$

$$= \frac{\partial}{\partial b_{k\ell}} \sum_{i}^{N} \sum_{j}^{N} b_{ij} \langle 0 | U(\vec{x})^\dagger W(\Theta)^\dagger E_{ij} W(\Theta) U(\vec{x}) | 0 \rangle$$

$$= \sum_{i}^{N} \sum_{j}^{N} \delta_{ik} \delta_{j\ell} \langle 0 | U(\vec{x})^\dagger W(\Theta)^\dagger E_{ij} W(\Theta) U(\vec{x}) | 0 \rangle$$

$$= \langle 0 | U(\vec{x})^\dagger W(\Theta)^\dagger E_{k\ell} W(\Theta) U(\vec{x}) | 0 \rangle$$

$$= \overline{(W(\Theta) U(\vec{x}))}_{k1} (W(\Theta) U(\vec{x}))_{\ell 1}$$

where the bar is the complex conjugate and $k, \ell \in \{1, \ldots, N\}$ denote matrix indices. Θ represents the parameters of the variational quantum circuit. The last equality shows the explicit dependency on $W(\Theta)$. As illustrated in Figure 4, under the FWP framework, an external neural network can be employed to generate the VQC rotation parameters, the parameters

controlling the measurement observables, or both. When FWP is applied to a single component (either the rotation angles or the observable), the architecture consists of a single-layer feedforward neural network whose input size matches the feature dimension and whose output size corresponds to the number of trainable parameters for that component. In the case where FWP is used to jointly program both the VQC and the observable, the framework adopts an encoder-decoder design: the input is first encoded into a latent representation, which is then processed by two separate neural networks to produce the VQC rotation angles and the observable parameters, respectively.

V. EXPERIMENTS

To evaluate the effectiveness of the proposed method, we conduct a series of experiments under various VQC learning configurations, as summarized in Table I. In the table, if the "Observable" field is marked as "None", a fixed Pauli-Z operator is used as the measurement observable. If the "Circuit Param" is labeled "None", the corresponding VQC parameters are randomly initialized and remain untrained throughout. Entries marked as "NN" indicate that the corresponding component is dynamically generated by an external neural network controller. All experiments follow a unified training protocol unless otherwise specified: batch size = 20, training set size = 200, testing set size = 100, VQC depth = 2. The learning rate is set to 0.01 for all FWP-based methods and for direct optimization of circuit parameters, while a higher learning rate of 0.1 is used for optimizing observables directly. Each experiment is repeated five times with different random seeds to report the mean and standard deviation for robust statistical comparison. For the `make_moons` and `make_circles` tasks, we employ a 4-qubit system. In the `make_classification` experiments, the number of qubits is matched to the input feature dimension, using 8, 10, and 12 qubits, respectively. To evaluate the consistency and

TABLE I
VARIOUS VQC LEARNING SETTINGS CONSIDERED.

	Circuit Param	Observable
VQC	RMSProp	None
VQC Learnable Observable	RMSProp	RMSProp
VQC Learnable Observable Separate Opt	RMSProp	Adam
VQC Learnable Observable Only	None	RMSProp
FWP VQC Parameters	NN	None
FWP VQC Learnable Observable	None	NN
FWP VQC Paramaters and Learnable Observable	NN	NN

effectiveness of various VQC learning strategies, we conduct a series of experiments on the `make_moons` dataset with Gaussian noise (noise = 0.1). Each configuration is trained across five different random seeds, and the reported results in Figure 6 show the mean and standard deviation of both training loss and classification accuracy over 40 epochs. The results reveal a clear advantage for FWP-based models over traditional optimizer-based VQCs. Notably, the setting where both the circuit parameters and measurement observable are generated by a data-conditioned neural network (*FWP VQC*
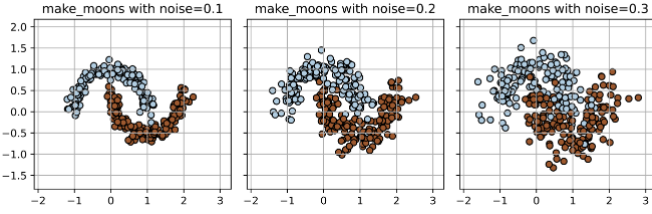
Fig. 5. **Visualization of `make_moons` dataset with different noise configurations.**

Fig. 6. **Comparison of different VQC models in `make_moons` dataset with noise = 0.1.** Top: Mean training loss across 5 random seeds (± standard deviation). Bottom: Mean classification accuracy (± standard deviation). FWP-based configurations, particularly *FWP VQC Parameters and Learnable Observable*, achieve faster convergence and higher accuracy, suggesting that neural generation of both circuit parameters and observables enhances training efficiency and robustness.

Parameters and Learnable Observable) achieves the most rapid convergence and highest accuracy, while maintaining low variance across seeds. In contrast, traditional VQCs that rely solely on classical optimizers, especially those optimizing only the observable (*VQC Learnable Observable Only*), exhibit slower and less stable learning. These results highlight the benefits of one-shot parameter generation and further support the hypothesis that decoupling parameter optimization from iterative procedures can mitigate gradient-related training challenges in quantum models.

To evaluate model robustness under increased input uncertainty, we repeat the classification experiment with a higher noise level (noise = 0.2) in the make_moons dataset. As shown in Figure 7, all models experience performance degradation to varying degrees, but the proposed FWP-based configurations remain substantially more resilient. In particular, *FWP VQC Parameters and Learnable Observable* maintains a rapid convergence rate and achieves near-perfect accuracy across runs. The low variance bands indicate consistent performance across different seeds, even in the presence of noisy inputs. This suggests that one-shot parameter generation conditioned on the input distribution may offer a regularizing effect and improve generalization. Traditional VQCs optimized via classical optimizers suffer more from the increased noise, with the baseline *VQC* and *VQC Learnable Observable Only* showing higher variance and slower accuracy growth. These results reinforce the benefits of integrating neural modules (FWP) to bypass direct optimization of quantum parameters, particularly in scenarios with data uncertainty.

To further examine model robustness under challenging conditions, we repeat the classification task with noise = 0.3, simulating severe input uncertainty. As shown in Figure 8, traditional optimization-based VQCs exhibit noticeable degradation, with slower convergence and wider performance variance. Notably, the *VQC Learnable Observable Only* configuration becomes increasingly unstable, suggesting that training observables in isolation may be insufficient in high-noise regimes. By contrast, FWP-based architectures remain highly resilient. The configuration utilizing neural generation for both circuit parameters and observables (*FWP VQC Parameters and Learnable Observable*) continues to outperform other settings, achieving near 0.95 accuracy with minimal variance across all seeds. These results highlight the potential of neural-controlled quantum models to generalize well even under
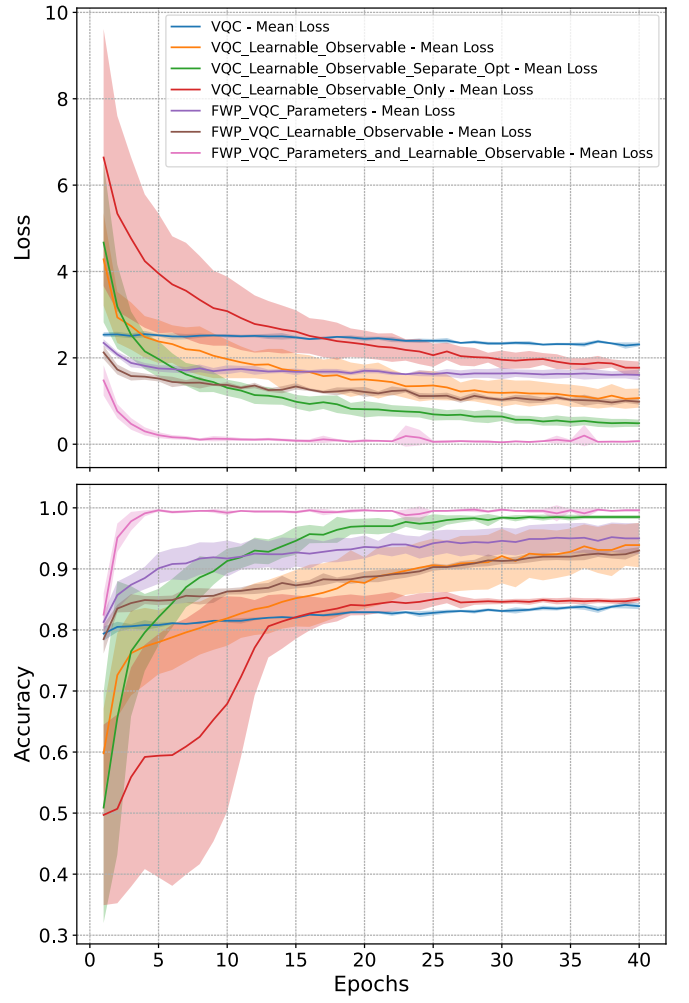
adverse conditions. Across all noise levels (noise = 0.1, 0.2, and 0.3), the proposed FWP-based configurations consistently outperform their classical counterparts. In particular, the full FWP model that generates both VQC rotation parameters and observables maintains high accuracy, low loss, and reduced variance regardless of data perturbations. These findings suggest that integrating a data-conditioned neural prior into VQC training may enhance generalization and stability in real-world noisy quantum learning scenarios. On the make_circles dataset with low noise (noise = 0.05), the benefit of FWP-based configurations remains evident in terms of rapid convergence and early-stage accuracy (shown in Figure 10). Notably, *FWP VQC Parameters and Learnable Observable* achieves the fastest loss reduction and reaches high accuracy within the first few epochs. However, unlike previous experiments,
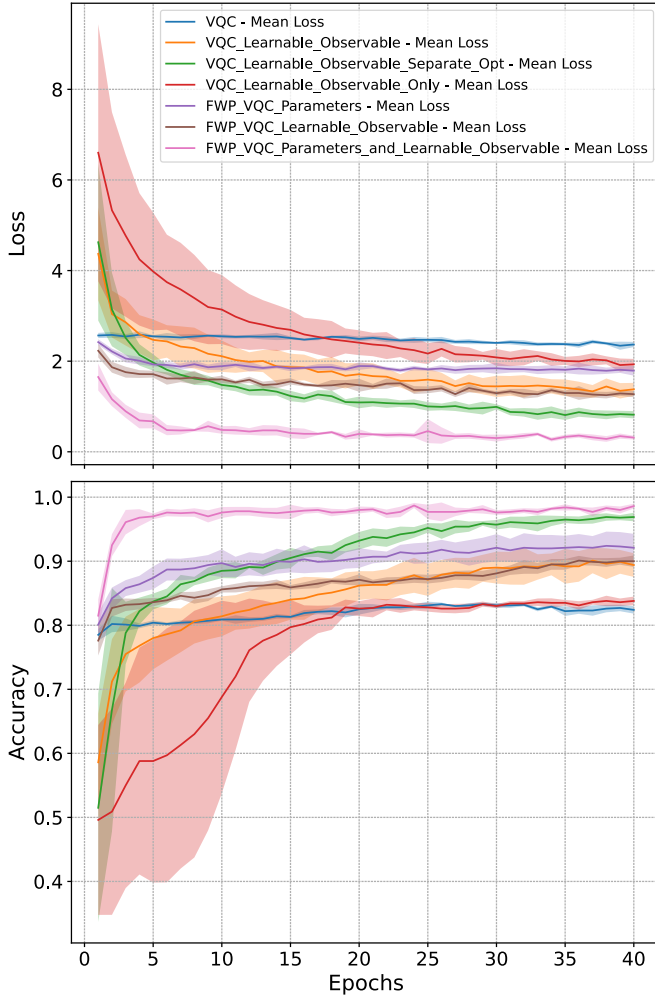
Fig. 7. **Comparison of different VQC models in `make_moons` dataset with noise = 0.2.** Top: Mean training loss over 5 runs (± standard deviation). Bottom: Classification accuracy across 40 epochs. The *FWP VQC Parameters and Learnable Observable* setting maintains superior performance despite the added noise, achieving near-perfect accuracy with low variance.

Fig. 8. **Comparison of different VQC models in `make_moons` dataset with noise = 0.3.** Top: Mean training loss over 5 random seeds (± standard deviation). Bottom: Mean classification accuracy (± standard deviation). Despite increased data noise, FWP-based models, especially the dual-generation variant, maintain strong accuracy and low loss variance, demonstrating superior robustness compared to traditional optimizer-based VQCs.

the *VQC Learnable Observable Separate Opt* setting performs surprisingly well in this task, ultimately matching or slightly exceeding the final accuracy of FWP models. This indicates that separating the optimization strategies for circuit parameters and observables may lead to better convergence behavior in classically trained VQCs-at least under low-noise, geometrically structured inputs. These results suggest that while neural parameter generation (FWP) offers clear advantages in generalization and early-stage learning, certain hand-crafted optimizer configurations can still compete under favorable conditions. When the noise level is increased to noise = 0.1 in the *make_circles* dataset, the gap between FWP-based configurations and classical VQCs becomes more evident (Figure 11). While the *VQC Learnable Observable Separate Opt* setting still reaches competitive final accuracy, its early-stage learning is slower and its performance variance is significantly higher. In contrast, the FWP configura-
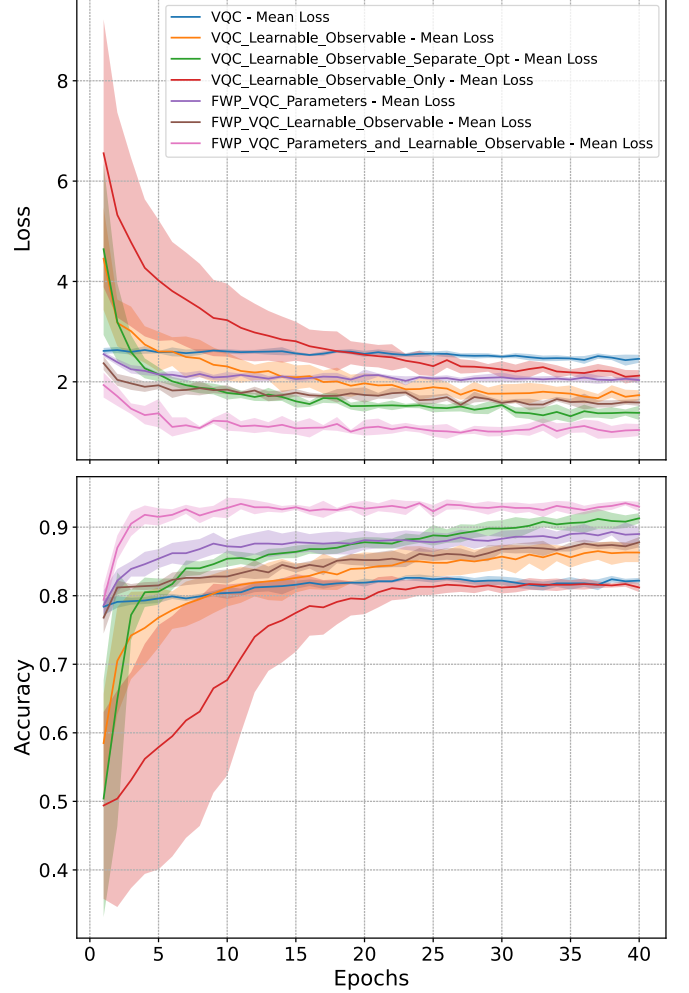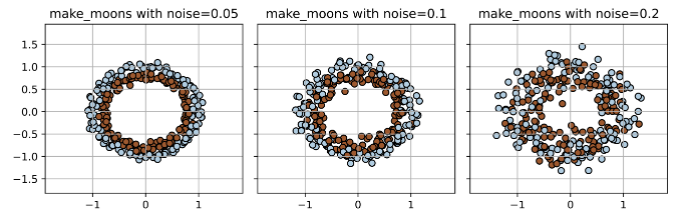

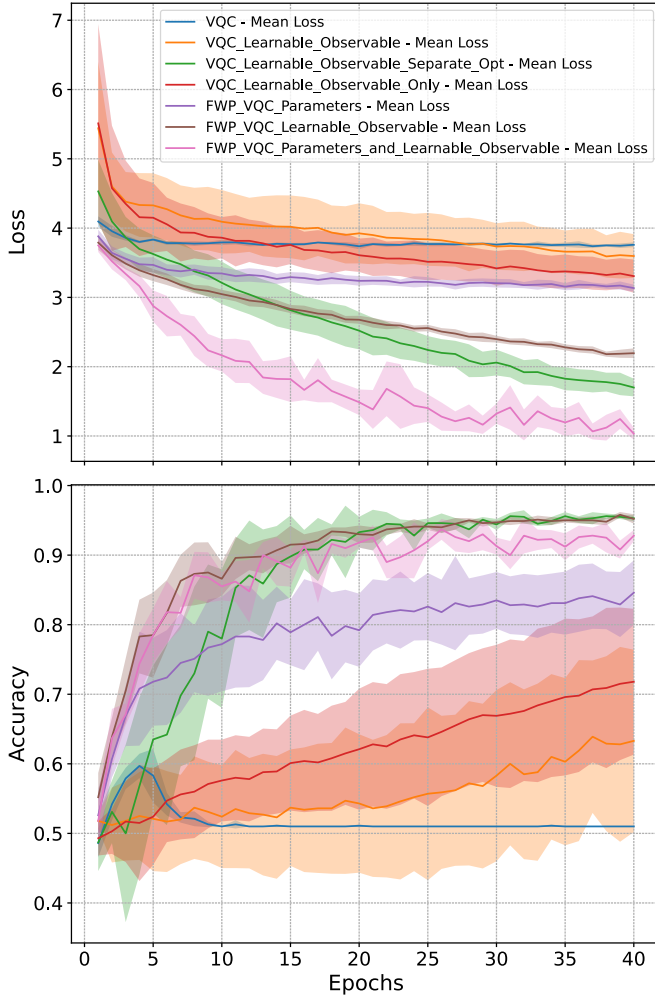Fig. 9. **Visualization of `make_circles` dataset with different noise configurations.**

Fig. 10. **Comparison of different VQC models in `make_circles` dataset with noise = 0.05.** Top: Mean training loss over 5 random seeds (± standard deviation). Bottom: Mean classification accuracy. While FWP-based models achieve faster convergence, the *VQC Learnable Observable Separate Opt* setting also reaches competitive final accuracy, suggesting that careful optimizer assignment can partially mitigate classical limitations.
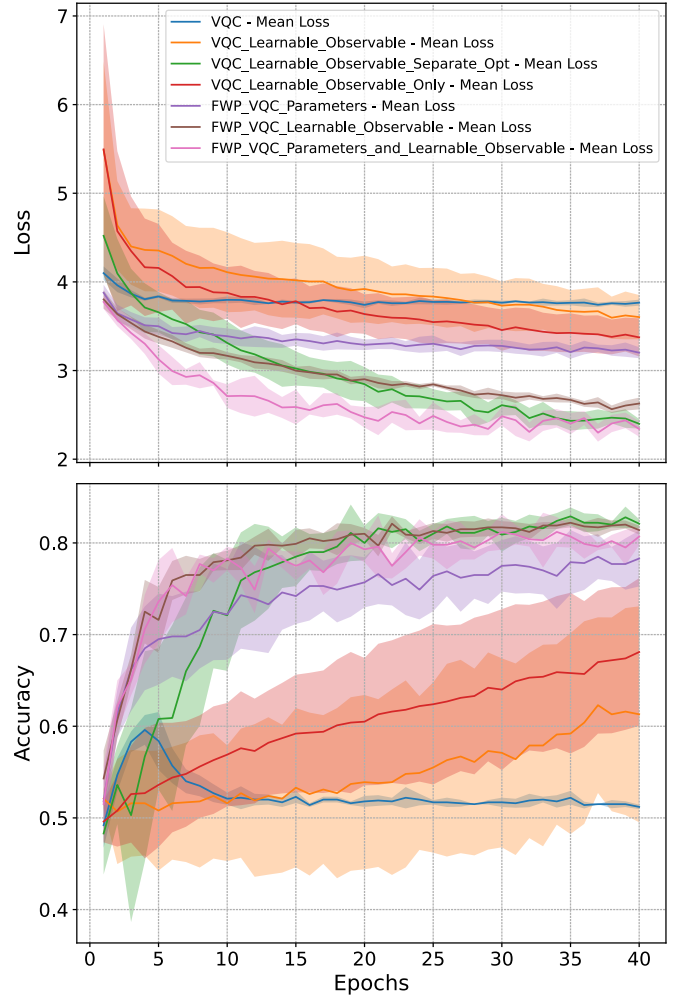
Fig. 11. **Comparison of different VQC models in `make_circles` dataset with noise = 0.1.** Top: Mean loss over 5 random seeds (± standard deviation). Bottom: Accuracy progression over 40 epochs. Classical optimizer-based VQCs exhibit reduced stability, while FWP-based models retain both faster convergence and superior accuracy. Notably, the dual-generator model consistently outperforms all baselines.

tions—especially the dual-generator (*FWP VQC Parameters and Learnable Observable*)-achieve rapid convergence, stable training curves, and the highest overall accuracy. Even under moderate noise, classical baselines such as *VQC* and *VQC Learnable Observable* suffer from underfitting and instability. These findings highlight the robustness of data-conditioned quantum parameter generation under noisy conditions, and suggest that FWP-based architectures can serve as more reliable foundations in real-world applications where clean data is not guaranteed. When evaluated under high noise conditions (noise = 0.2) in the `make_circles` dataset, most classical VQC settings exhibit significant performance degradation (Figure 12). The standard *VQC* and *VQC Learnable Observable Only* configurations fail to generalize and produce accuracy curves barely exceeding 55%. While *VQC Learnable Observable Separate Opt* had previously shown

promise, its performance becomes increasingly unstable in this noisy regime. In contrast, the FWP-based models—especially the dual-generator setting—maintain stable training curves and achieve the highest accuracy among all tested configurations, with minimal variance. These results reinforce the hypothesis that neural parameter generation mechanisms such as FWP are more resilient to noisy inputs, offering a promising direction for quantum machine learning in real-world, imperfect data environments. In the synthetic `make_classification` task with 8 input features, all models are able to reach high final accuracy, reflecting the relatively low task complexity. However, significant differences arise in training dynamics and convergence speed (Figure 13). The *FWP VQC Parameters and Learnable Observable* configuration achieves near-perfect accuracy within the first few epochs, with the lowest average loss and smallest variance. Other FWP configura-
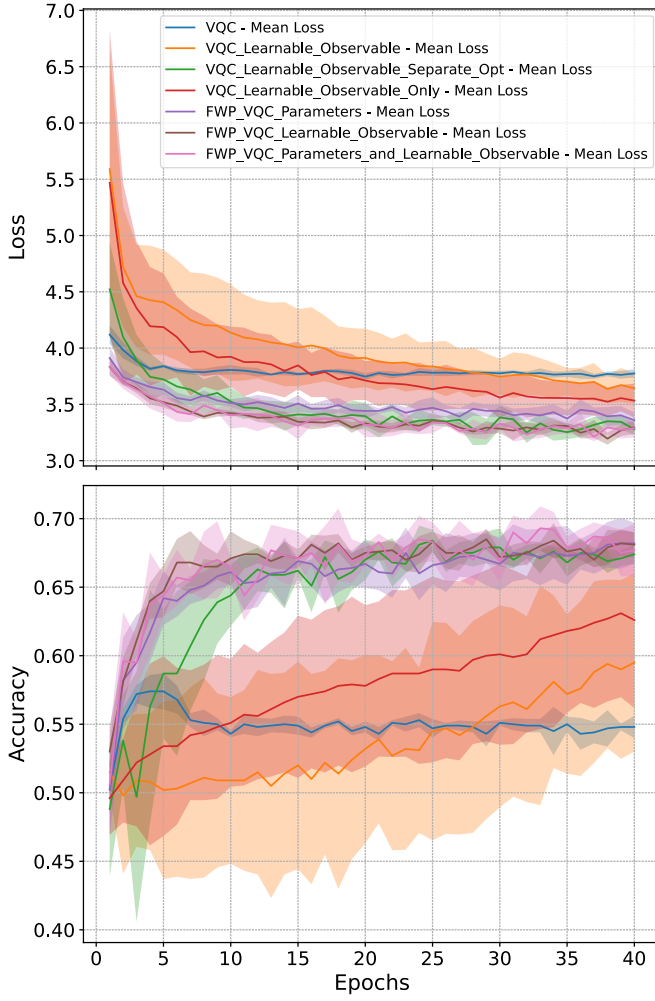
Fig. 12. **Comparison of different VQC models in `make_circles` dataset with noise = 0.2.** Top: Mean training loss (± std) across 5 runs. Bottom: Classification accuracy over 40 epochs. With increased noise, classical optimizer-based VQCs exhibit performance collapse, while FWP-based models retain relatively stable accuracy and learning dynamics.

Fig. 13. **Comparison of different VQC models in `make_classification` dataset with 8 features.** Top: Mean training loss over 5 random seeds (± standard deviation). Bottom: Classification accuracy. All models eventually reach high accuracy, but FWP-based configurations—especially the dual-generator-achieve significantly faster convergence and lower loss throughout training.

tions also converge rapidly, outperforming classical optimizer-based settings in both early-stage and stable performance. In contrast, traditionally optimized VQCs-especially *VQC* and *VQC Learnable Observable*-exhibit slower convergence and higher loss curves, highlighting the inefficiency of classical gradient methods in encoding meaningful representations early in training. The *VQC Learnable Observable Only* configuration continues to underperform, validating the need for full-model adaptability in hybrid quantum-classical setups. In the more challenging setting with 10 input features from the `make_classification` dataset, the performance gap between FWP-based models and classical VQC configurations becomes increasingly pronounced (Figure 14). While the *VQC Learnable Observable Separate Opt* previously showed competitive accuracy, it now plateaus near 70%, indicating its limited scalability to higher-dimensional feature spaces. Conversely, all FWP-based models maintain strong general-
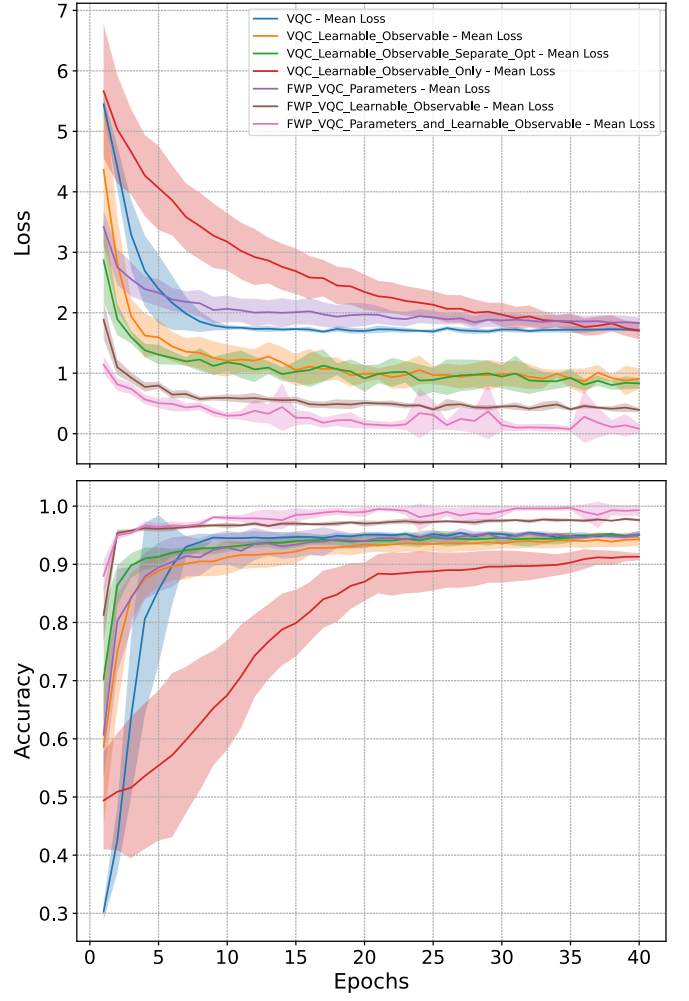
ization and fast convergence, with *FWP VQC Parameters and Learnable Observable* achieving nearly perfect classification performance. Although this dual-generator model shows minor fluctuations in loss, its accuracy remains consistently superior. The *FWP VQC Parameters* variant also demonstrates robust training stability, striking a favorable balance between efficiency and reliability. These results affirm the scalability of FWP-based approaches to more complex learning tasks, highlighting their potential for practical applications involving high-dimensional quantum-classical data. As the number of input features increases to 12, classical VQC models exhibit further degradation in both convergence and final accuracy (Figure 15). Most traditional settings, including *VQC*, *VQC Learnable Observable*, and *VQC Learnable Observable Separate Opt*, plateau at ~70% accuracy and struggle to reduce training loss. In contrast, FWP-based models maintain their
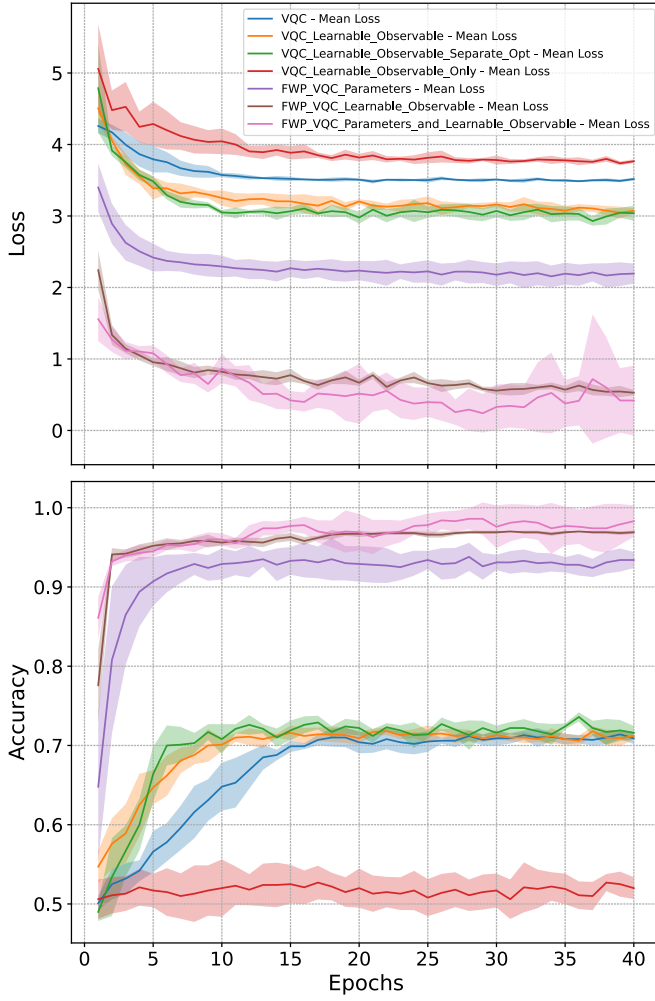
Fig. 14. **Comparison of different VQC models in `make_classification` dataset with 10 features.** Top: Mean loss over 5 random seeds (± standard deviation). Bottom: Classification accuracy over training epochs. As input dimensionality increases, traditional optimizer-based models plateau below 75% accuracy, while FWP-based configurations continue to scale effectively and reach over 95%.
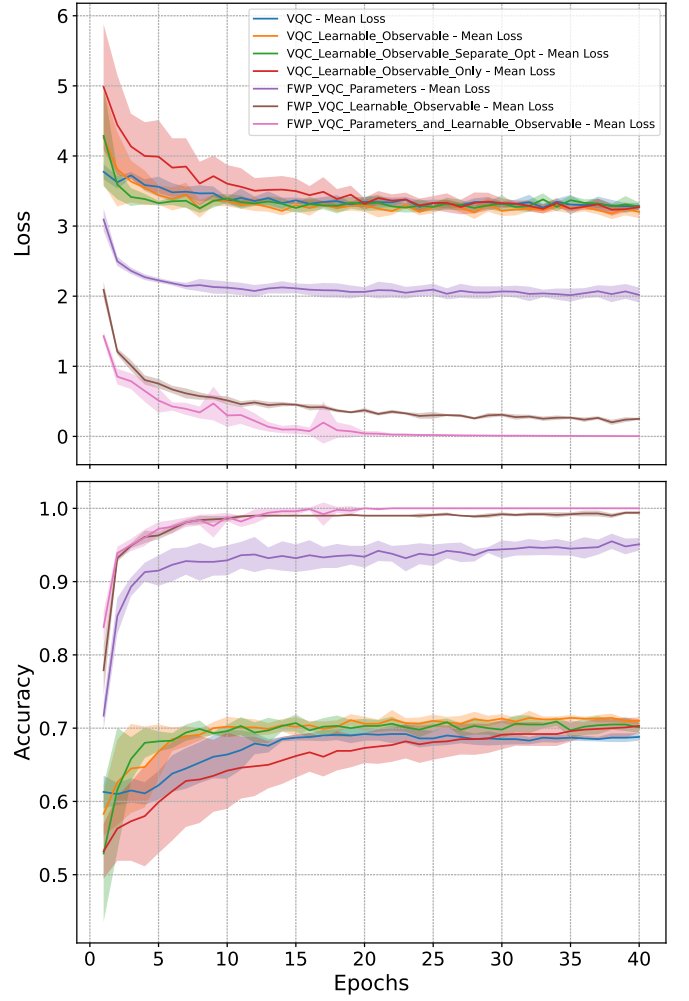
Fig. 15. **Comparison of different VQC models in `make_classification` dataset with 12 features.** Top: Mean loss across 5 seeds (± standard deviation). Bottom: Classification accuracy. Classical optimizer-based VQCs fail to scale effectively, while all FWP-based models maintain prediction performance, with the dual-generator achieving near-perfect accuracy.

performance advantage and demonstrate excellent scalability. The *FWP VQC Parameters and Learnable Observable* configuration once again achieves near-perfect accuracy and minimal training variance. The *FWP VQC Learnable Observable* setting also performs exceptionally well, followed closely by *FWP VQC Parameters*, which remains above 90% accuracy throughout training. These findings highlight the structural limitation of classical optimizer-based approaches when facing increased input complexity, and further validate the effectiveness of FWP-based architectures in high-dimensional quantum learning.

VI. CONCLUSION

In this study, we conducted a comprehensive ablation analysis of various learning configurations for variational quantum circuits (VQCs), with a particular focus on classifying synthetic datasets under different noise levels and input

dimensionalities. Our comparisons span classical optimizer-based approaches and neural network-based parameter generation, implemented via the proposed Fast Weight Programmer (FWP). Numerical results across multiple benchmark tasks—including make_moons, make_circles, and make_classification-consistently demonstrate the superiority of FWP-based configurations. The dual-generator variant, which jointly produces VQC parameters and observables conditioned on input data, achieves rapid convergence, low variance, and superior classification accuracy even under high noise or feature-rich settings. In contrast, classical VQC training pipelines struggle with scalability, often failing to generalize in noisy or complex regimes. These findings suggest that replacing classical optimizers with input-aware neural generation mechanisms offers a promising direction for training hybrid quantum-classical models more efficiently.

The proposed FWP framework not only alleviates gradient-based optimization bottlenecks but also improves robustness and adaptability in quantum learning tasks.

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information*. Cambridge university press, 2010.

[2] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[3] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[4] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke *et al.*, "Noisy intermediate-scale quantum algorithms," *Reviews of Modern Physics*, vol. 94, no. 1, p. 015004, 2022.

[5] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[6] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'brien, "A variational eigenvalue solver on a photonic quantum processor," *Nature communications*, vol. 5, no. 1, p. 4213, 2014.

[7] E. Farhi, J. Goldstone, and S. Gutmann, "A quantum approximate optimization algorithm," *arXiv preprint arXiv:1411.4028*, 2014.

[8] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, p. 032309, 2018.

[9] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Physical Review A*, vol. 101, no. 3, p. 032308, 2020.

[10] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, and Y.-J. Kao, "An end-to-end trainable hybrid classical-quantum classifier," *Machine Learning: Science and Technology*, vol. 2, no. 4, p. 045021, 2021.

[11] S. Y.-C. Chen, T.-C. Wei, C. Zhang, H. Yu, and S. Yoo, "Quantum convolutional neural networks for high energy physics data analysis," *Physical Review Research*, vol. 4, no. 1, p. 013231, 2022.

[12] J. Qi, C.-H. H. Yang, and P.-Y. Chen, "Qtn-vqc: An end-to-end learning framework for quantum neural networks," *Physica Scripta*, vol. 99, 12 2023.

[13] H.-Y. Lin, H.-H. Tseng, S. Y.-C. Chen, and S. Yoo, "Quantum gradient class activation map for model interpretability," in *2024 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2024, pp. 165–170.

[14] S. Y.-C. Chen, S. Yoo, and Y.-L. L. Fang, "Quantum long short-term memory," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8622–8626.

[15] C.-S. Chen, S. Y.-C. Chen, A. H.-W. Tsai, and C.-S. Wei, "Qeegnet: Quantum machine learning for enhanced electroencephalography encoding," in *2024 IEEE Workshop on Signal Processing Systems (SiPS)*. IEEE, 2024, pp. 153–158.

[16] C. Chu, G. Skipper, M. Swany, and F. Chen, "Iqgan: Robust quantum generative adversarial network for image synthesis on nisq devices," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[17] S. S. Li, X. Zhang, S. Zhou, H. Shu, R. Liang, H. Liu, and L. P. Garcia, "Pqlm-multilingual decentralized portable quantum language model," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[18] C.-H. H. Yang, J. Qi, S. Y.-C. Chen, Y. Tsao, and P.-Y. Chen, "When bert meets quantum temporal convolution learning for text classification in heterogeneous computing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8602–8606.

[19] R. Di Sipio, J.-H. Huang, S. Y.-C. Chen, S. Mangini, and M. Worring, "The dawn of quantum natural language processing," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8612–8616.

[20] J. Stein, I. Christ, N. Kraus, M. B. Mansky, R. Müller, and C. Linnhoff-Popien, "Applying qnlp to sentiment analysis in finance," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 2. IEEE, 2023, pp. 20–25.

[21] C.-Y. Liu, E.-J. Kuo, C.-H. A. Lin, J. G. Young, Y.-J. Chang, M.-H. Hsieh, and H.-S. Goan, "Quantum-train: Rethinking hybrid quantum-classical machine learning in the model compression perspective," *arXiv preprint arXiv:2405.11304*, 2024.

[22] C.-H. A. Lin, C.-Y. Liu, and K.-C. Chen, "Quantum-train long short-term memory: Application on flood prediction problem," *arXiv preprint arXiv:2407.08617*, 2024.

[23] C.-Y. Liu, C.-H. A. Lin, C.-H. H. Yang, K.-C. Chen, and M.-H. Hsieh, "Qtrl: Toward practical quantum reinforcement learning via quantum-train," *arXiv preprint arXiv:2407.06103*, 2024.

[24] C.-Y. Liu and S. Y.-C. Chen, "Federated quantum-train with batched parameter generation," *arXiv preprint arXiv:2409.02763*, 2024.

[25] C.-Y. Liu, C.-H. A. Lin, and K.-C. Chen, "Quantum-train with tensor network mapping model and distributed circuit ansatz," *arXiv preprint arXiv:2409.06992*, 2024.

[26] C.-H. A. Lin, C.-Y. Liu, S. Y.-C. Chen, and K.-C. Chen, "Quantum-trained convolutional neural network for deepfake audio detection," *arXiv preprint arXiv:2410.09250*, 2024.

[27] C.-Y. Liu, C.-H. H. Yang, M.-H. Hsieh, and H.-S. Goan, "A quantum circuit-based compression perspective for parameter-efficient learning," *arXiv preprint arXiv:2410.09846*, 2024.

[28] K.-C. Chen, S. Y.-C. Chen, C.-Y. Liu, and K. K. Leung, "Quantum-train-based distributed multi-agent reinforcement learning," *arXiv preprint arXiv:2412.08845*, 2024.

[29] C.-Y. Liu, S. Y.-C. Chen, K.-C. Chen, W.-J. Huang, and Y.-J. Chang, "Programming variational quantum circuits with quantum-train agent," *arXiv preprint arXiv:2412.01173*, 2024.

[30] S. Y.-C. Chen, C.-M. Huang, C.-W. Hsing, H.-S. Goan, and Y.-J. Kao, "Variational quantum reinforcement learning via evolutionary optimization," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015025, 2022.

[31] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, "Variational quantum circuits for deep reinforcement learning," *IEEE access*, vol. 8, pp. 141 007–141 024, 2020.

[32] S. Y.-C. Chen, "Efficient quantum recurrent reinforcement learning via quantum reservoir computing," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 13 186–13 190.

[33] N. Meyer, C. Ufrecht, M. Periyasamy, D. D. Scherer, A. Plinge, and C. Mutschler, "A survey on quantum reinforcement learning," *arXiv preprint arXiv:2211.03464*, 2022.

[34] S. Y.-C. Chen, "Quantum deep recurrent reinforcement learning," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[35] O. Lockwood and M. Si, "Reinforcement learning with quantum variational circuit," in *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. 16, no. 1, 2020, pp. 245–251.

[36] A. Skolik, S. Jerbi, and V. Dunjko, "Quantum agents in the gym: a variational quantum algorithm for deep q-learning," *Quantum*, vol. 6, p. 720, 2022.

[37] S. Jerbi, C. Gyurik, S. Marshall, H. Briegel, and V. Dunjko, "Parametrized quantum policies for reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 34, pp. 28 362–28 375, 2021.

[38] R. Coelho, A. Sequeira, and L. Paulo Santos, "Vqc-based reinforcement learning with data re-uploading: performance and trainability," *Quantum Machine Intelligence*, vol. 6, no. 2, p. 53, 2024.

[39] S. Y.-C. Chen, "Asynchronous training of quantum reinforcement learning," *Procedia Computer Science*, vol. 222, pp. 321–330, 2023, international Neural Network Society Workshop on Deep Learning Innovations and Applications (INNS DLIA 2023). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050923009365

[40] W. J. Yun, J. Park, and J. Kim, "Quantum multi-agent meta reinforcement learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 9, 2023, pp. 11 087–11 095.

[41] S. Y.-C. Chen, H.-H. Tseng, H.-Y. Lin, and S. Yoo, "Learning to measure quantum neural networks," *arXiv preprint arXiv:2501.05663*, 2025.

[42] D. Martyniuk, J. Jung, and A. Paschke, "Quantum architecture search: a survey," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2024, pp. 1695–1706.

[43] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, "Differentiable quantum architecture search," *Quantum Science and Technology*, vol. 7, no. 4, p. 045023, 2022.

[44] S. Y.-C. Chen, "Differentiable quantum architecture search in asynchronous quantum reinforcement learning," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 1. IEEE, 2024, pp. 1516–1524.

[45] H. Liu, K. Simonyan, and Y. Yang, "Darts: Differentiable architecture search," *arXiv preprint arXiv:1806.09055*, 2018.

[46] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, vol. 4, no. 1, pp. 131–139, 1992.

[47] S. Y.-C. Chen, "Learning to program variational quantum circuits with fast weights," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–9.

[48] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[49] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation*, vol. 15, no. 6, pp. 1373–1396, 2003.

[50] D. Bleecker, *Gauge theory and variational principles*. Addison-Wesley Pub. Co., Advanced Book Program/World Science Division, 1981.

[51] H.-H. Tseng, "Gravitational theories with torsion," *arXiv preprint arXiv:1812.00314*, 2018.

[52] J. Jost and J. Jost, *Riemannian geometry and geometric analysis*. Springer, 2008, vol. 42005.