Compact Feedback-Enhanced Quantum Fast Weight Programmer for Time Series Forecasting

Francesco Sanetti*, Andrea Ceschini*, Andrew Hornback†,
Samuel Yen-Chi Chen‡, Antonello Rosato* and Massimo Panella*
*Department of Information Engineering, Electronics and Telecommunications (DIET)
University of Rome "La Sapienza", Via Eudossiana 18, 00184 Rome, Italy
Email: {andrea.ceschini, antonello.rosato, massimo.panella}@uniroma1.it
†School of Computational Science, Georgia Institute of Technology, Atlanta, GA 30332, USA
‡Wells Fargo Bank, USA

***Abstract*—Quantum recurrent models are principled for temporal learning, yet their reliance on backpropagation through time and deep unrolling is poorly matched to Noisy Intermediate-Scale Quantum constraints. Quantum Fast Weight Programming alleviates this depth bottleneck via shallow, online parameter adaptation, but remains sensitive to noise-induced variance and gradient degradation, especially when the programmer has limited feedback about the current fast-weight state. This work tries to overcome these challenges by introducing the Compact Feedback-Enhanced Quantum Fast Weight Programmer, a hybrid architecture that strengthens quantum fast-weight programming with an explicit state-aware controller while preserving shallow, hardware-efficient quantum execution. To maintain parameter efficiency, the controller generates rank-1 parameter increments via the outer product of layer-wise and qubit-wise modulation vectors, avoiding full-matrix generation while supporting expressive adaptation. The proposed approach is assessed on three real-world forecasting benchmarks spanning disparate fields and dynamics, consistently improving predictive accuracy over both quantum fast weight programming approaches and quantum deep recurrent neural networks, while requiring also substantially less training time than quantum baselines.**

I. Introduction

The application of Quantum Machine Learning (QML) to sequential data and signal processing has gained significant momentum. While traditional quantum recurrent architectures, such as Quantum RNNs (QRNN) and Quantum LSTM (QLSTM) networks, successfully extend classical gated logic into the quantum domain [1], [2], they face severe scalability hurdles. Their reliance on Backpropagation Through Time (BPTT) and deep, unrolled circuit structures imposes a heavy computational burden that often exceeds the capabilities of current Noisy Intermediate-Scale Quantum (NISQ) hardware [3]–[5].

A promising alternative is Quantum Fast Weight Programming (QFWP) [6], which replaces explicit quantum recurrence with a hybrid 'slow-fast' mechanism. At each time step t, a classical slow programmer g_θ (parameterized by trainable weights θ) computes an update $\Delta\Phi_t$ for the parameters of a shallow Variational Quantum Circuit (VQC). The VQC parameters, denoted by Φ_t, evolve according to

$$\Phi_{t+1} = \Phi_t + \Delta\Phi_t\,, \qquad \Phi_0 = 0\,, \tag{1}$$

where the subscript t indicates the time index and $\Delta\Phi_t$ denotes the increment produced at time t. In this formulation, the model's memory is not carried by a recurrent quantum state but is instead encoded in the trajectory of circuit parameters $\{\Phi_t\}_{t=1}^{T}$. this design avoids deep circuit unrolling and enables gradient estimation through the parameter-shift rule [7] while preserving hardware-efficient, shallow quantum circuits. Such a configuration brings a key advantage in regimes where circuit depth and trainability are limiting factors [8], [9]. However, in standard QFWP the slow programmer computes the update $\Delta\Phi_t$ only from the current input x_t, without explicitly observing the current fast-weight state Φ_t. As a result, the update mechanism is input-driven but not state-aware: it can react to the incoming sample, yet it cannot directly adapt its update according to the current trajectory already followed by the VQC.

In this paper, we introduce the Compact Feedback-Enhanced Quantum Fast Weight Programmer (CFE-QFWP), a novel recurrent extension of QFWP that aims to improve forecasting accuracy without sacrificing QFWP's computational efficiency. The key idea is to make the slow programmer state-aware by injecting a compact feedback signal derived from the current fast-weight configuration. This explicit recurrence is realized within the classical programmer rather than the quantum circuit, thereby incorporating temporal dependence while avoiding the prohibitive cost of applying BPTT through a deeply unrolled quantum computation.

We validate CFE-QFWP on three real-world univariate forecasting benchmarks with different temporal scales and dynamics: (i) monthly sunspot number series with stochastic fluctuations and quasi-periodic structure; (ii) hourly wind power generation characterized by strong short-term variability and weather-driven regime changes: (iii) hourly global horizontal irradiance, which exhibits diurnal patterns and weather-induced intermittency. Experimental results show that the proposed architecture consistently improves over both QLSTM and the original QFWP, while also outperforming classical LSTM. These gains come with lower training time than QLSTM, supporting CFE-QFWP as a practical and scalable hybrid approach for time series forecasting in the NISQ setting. More broadly, the results suggest that parameter-efficient

quantum-enhanced models may offer a complementary path toward more energy-efficient learning and deployment.

II. Related Work

The emergence of QML has introduced new paradigms for addressing complex signal processing and sequential data modeling challenges. Central to many QML approaches are Parameterized Quantum Circuits (PQCs), also referred to as VQCs, which combine quantum principles with classical optimization routines [10]. These hybrid architectures provide an expressive yet resource-efficient framework for representing high-dimensional data and have demonstrated promise across a range of multiple Machine Learning (ML) tasks [11]–[13], including time series forecasting and reinforcement learning, where capturing temporal dependencies is critical [14], [15].

To explicitly handle sequential information, classical recurrent architectures have been extended into the quantum domain through QRNNs and QLSTM networks [16], [17]. As said, while these models aim to leverage quantum-enhanced representations for sequence modeling, they are fundamentally constrained by their reliance on BPTT. In fact, in quantum settings, BPTT leads to a rapid increase in circuit evaluations and exacerbates training challenges such as barren plateaus, particularly as sequence length or circuit depth increases [18]–[20]. These issues are further amplified by the limited coherence times and noise characteristics of current NISQ hardware, motivating the development of shallow, locality-aware architectures that maintain computational efficiency without sacrificing predictive performance [21].

In response to these limitations, the paradigm of Fast Weight Programmers (FWPs), originally proposed in classical ML for rapid adaption to sequential data [22], [23], has been translated into the quantum domain. In the classical setting, FWPs employ a dual-network structure in which a slow controller network computes updates to the fast weights of a secondary model, enabling efficient temporal credit assignment without explicit recurrence. This dual-layered approach facilitates high-speed adaptation and the effective management of temporal dependencies. QFWP extends this idea to PQCs, marrying the adaptive nature of FWPs with the inherent expressive advantages of quantum architectures [6].

Within the QFWP framework, a compact slow programmer g_θ calculates step-wise updates $\Delta\Phi_t$ for a fast PQC, which allows temporal credit assignment to be embedded directly into the evolving parameter trajectory. This mechanism removes the necessity for explicit recurrence and bypasses the computational burdens of BPTT. Thus, QFWP offers a more scalable alternative to QRNNs and QLSTM networks when operating under the constraints of NISQ hardware. Furthermore, the training of QFWPs is characterized by superior computational efficiency; by leveraging analytic gradients via parameter-shift rules, a PQC with P_q trainable parameters requires only $O(P_q)$ circuit evaluations per observable. This is in sharp contrast to the $O(TP_q)$ scaling typically required by unrolled recurrent training, where T represents the sequence length.

III. Proposed Methodology

The study presents a novel two-module QFWP architecture which introduces a recurrent mechanism inside the slow programmer Fig. 1. Consider $\{(x_t, y_t)\}_{t=1}^{T}$ the sequence of input target pairs. At any time step t, the state of the Fast Programmer (the VQC) is defined by its trainable parameters $\Phi_t \in \mathbb{R}^{L\times N}$, where L is the number of variational layers and N is the number of qubits. To facilitate direct comparison and isolate the impact of the proposed controller, the variational circuit architecture employed is identical to that of the original QFWP framework. This VQC design prioritizes NISQ hardware efficiency through a low-depth heuristic ansatz. Following initialization into a uniform superposition and angle encoding via R_y gates, the circuit applies L layers of fixed nearest-neighbor CNOTs and dynamic R_y rotations. The rotation angles Φ_t are updated by the classical Slow Programmer, and outputs are determined via Pauli-Z expectation values.

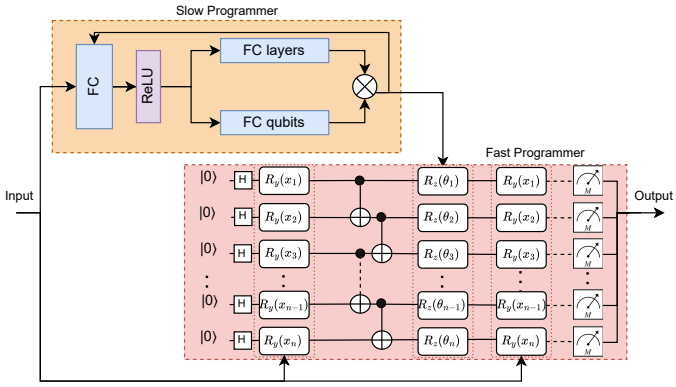

Fig. 1. Schematic overview of the proposed CFE-QFWP model, incorporating the new feedback mechanism in the Slow Programmer with an additional Rectified Linear Unit (ReLU) layer.

The said slow programmer, denoted by a function g_ψ parametrized by the classical weights ψ, computes the quantum parameters update by means of a feedback mechanism. To enable state-dependent adaptation, a context vector z_t is defined as the concatenation of the current input features and the flattened vector of the current quantum parameters:

$$z_t = x_t \oplus \mathrm{vec}(\Phi_t) \in \mathbb{R}^{d_{\mathrm{in}}+L\times N}, \tag{2}$$

where $\oplus$ denotes vector concatenation, $\mathrm{vec}(\cdot)$ the vectorization operation and d_{in} is the dimension of the input data. Intuitively, z_t acts as a compact readout of the current memory stored in the fast programmer. By concatenating the current input with the current quantum parameter state, the slow programmer can determine whether the ongoing fast-weight trajectory should be reinforced or corrected before producing the next update. This is the key difference from standard QFWP, where $\Delta\Phi_t$ is computed from x_t alone and therefore cannot explicitly react to the current fast-weight configuration.

The Slow Programmer processes z_t through a classical encoder to generate a hidden representation h_t. To improve

gradient flow during BPTT, a Rectified Linear Unit (ReLU) activation is adopted:

$$h_t = \mathrm{ReLu}\left(W_{\mathrm{enc}} z_t + b_{\mathrm{enc}}\right), \tag{3}$$

where W_{enc} and b_{enc} are the learnable weights of the encoder.

For maintaining computational efficiency consistent with the original QFWP framework, the Slow Programmer predicts a rank-1 modification to the existing parameters that avoids generating the full parameter matrix Φ directly. The hidden state h_t is projected into two modulation vectors: a layer-wise modulation vector $l_t \in \mathbb{R}^{\mathbb{L}}$ and a qubit-wise modulation vector $q_t \in \mathbb{R}^{\mathbb{N}}$:

$$l_t = W_l h_t + b_l, \qquad q_t = W_q h_t + b_q\,. \tag{4}$$

The parameter update matrix $\Delta\Phi_t$ is then computed via the outer product of these vectors:

$$\Delta\Phi_t = l_t \otimes q_t\,. \tag{5}$$

Consequently, the element-wise update rule for the parameter connecting the ith layer and the jth qubits is:

$$\theta_{ij}^{t+1} = \theta_{ij}^{t} + (l_t)_i \times (q_t)_j\,. \tag{6}$$

The update parameters Φ_{t+1} are immediately applied to the VQC (the fast programmer).

An end-to-end complete model training is carried out to minimize the prediction error across the sequence. The total loss function $\mathcal{L}$ is defined as the sum of the Mean Squared Error (MSE) and an L_2 regularization term applied to the slow programmer weights ψ to prevent overfitting:

$$\mathcal{L} = \frac{1}{T}\sum_{t=1}^{T} ||\hat{y}_t - y_t||^2 + \lambda||\psi||_2^2\,, \tag{7}$$

where $\hat{y}_t = f(x_t;\Phi_t)$ is the prediction generated by the quantum circuit at step t, y_t is the ground truth and λ is the regularization coefficient.

In the original QFWP framework, the slow programmer operates as a simple Feedforward function mapping the current input x_t directly to the parameter updates:

$$\Delta\Phi_t = g_\psi(x_t)\,. \tag{8}$$

Crucially, this update is independent of the current VQC configuration at time step t, limiting the slow programmer's ability to adapt based on the circuit's evolving trajectory. By contrast, the proposed slow controller incorporates an explicit feedback mechanism that conditions parameter updates on both the input and the fast programmer state:

$$\Delta\Phi_t = g_\psi(z_t)\,, \tag{9}$$

where z_t is the context vector defined by (2). This enables informed, state-dependent adaptation of circuit parameters Φ.

Notably, the BPTT introduced by the feedback mechanism is restricted to the slow controller, thereby bypassing the computational burden of applying BPTT to quantum parameters. This strategy enables the model to maintain $O(P_q)$ circuit evaluations per observable, significantly reducing the simulation overhead on classical hardware associated with quantum recurrent models like QRNNs and QLSTM networks.

IV. Illustrative Examples

A. Data and Experiment Setup

To evaluate the proposed model across different temporal scales and dynamics, we considered three univariate datasets. The first consists of 3265 monthly averaged sunspot numbers from January 1749 to July 2018, obtained from the Solar Physics Research Department of the Royal Observatory of Belgium. Due to their intrinsic stochasticity and quasi-periodic behavior, sunspots represent a standard benchmark for recurrent models on complex temporal dynamics.

The second dataset contains hourly wind power generation data from Italy for June 2022, comprising 720 observations (i.e., 30 days) measured in GWh. Wind energy forecasting is particularly challenging because of its strong non-linearity and volatility induced by stochastic atmospheric conditions.

The third dataset consists of 8759 hourly global horizontal irradiance observations collected at Oak Ridge National Laboratory in 2018, where cloud-driven intermittency makes future irradiance difficult to predict.

All data were scaled to the range $[-1, 1]$ to avoid periodicity issues in the VQC angle embedding and activation saturation in the classical network, thus improving training stability. The simulations were run on a commercial machine with 64 GB RAM and an AMD® Ryzen™ 7 5800X CPU (8 cores, 3.80 GHz), using Python 3.10 and Pennylane for quantum circuit definition and simulation. Adam was used for CFE-QFWP, QFWP, and LSTM due to its high efficiency in these setups, while QLSTM was trained with RMSprop, following [16], as it yielded superior performance with this model. Due to the high cost of exhaustive hyperparameter tuning for quantum models, all methods were evaluated under the same configuration and experimental settings reported in Table I.

We employed a 80%/20% split rather than 67%/33% in the Wind Power dataset because it contained 720 observations only. A larger training portion was retained in this case to avoid an overly small training set while still preserving a sufficiently informative held-out test segment. Furthermore, for this reason, only the following subsets of circuit configurations were considered: 6-qubit architectures with 1 and 2 layers, and 8-qubit architectures with 1 layer. To ensure the statistical robustness of our results, each experiment was repeated five times with different random initializations. The reported values represent the mean and standard deviation. In the subsequent analysis all metrics are computed using the best-performing configuration for each quantum model and the LSTM with hidden size 32.

TABLE I
Experimental Settings Adopted for Each Dataset

Dataset	Input	LR	Batch	Epochs	Split
Sunspot	4	0.001	16	100	67%/33%
Wind Power	5	0.001	16	130	80%/20%
Photovoltaic Power	4	0.001	16	50	67%/33%

TABLE II
AVERAGE TRAINING EFFICIENCY AND FINAL ACCURACY ON THE SUNSPOT DATASET

Model	# qubits	# layers	MSE ≤ 1.80		MSE ≤ 1.70		MSE ≤ 1.60		Final MSE (100 Epochs)	
			Epochs	Time (s)	Epochs	Time (s)	Epochs	Time (s)	Value	Time (s)
LSTM (h=4)	-	-	16.8	13	25.2	20	49.2	39	1.574 ± 0.036	80
QLSTM	6	1	43.19	2856	72.4	3667	*	*	1.660 ± 0.019	3600
QFWP	6	1	16.4	111	21.79	140	38.8	314	1.564 ± 0.043	744
CFE-QFWP	6	1	12	134	14.8	165	25.2	296	1.563 ± 0.014	899
LSTM (h=5)	-	-	13	11	21.60	19	44.60	39	1.575 ± 0.005	72
QLSTM	6	2	18.22	1037	30.97	1785	*	*	1.604 ± 0.018	5954
QFWP	6	2	5.2	75.4	10.8	184	*	*	1.611 ± 0.036	1450
CFE-QFWP	6	2	6	95	6.80	108	12.40	197	1.558 ± 0.095	1592
QLSTM	8	1	44.99	3070	73.80	4086	*	*	1.664 ± 0.019	6502
QFWP	8	1	22.99	325	32.99	468	43.2	606	1.570 ± 0.050	1288
CFE-QFWP	8	1	8.62	123	12.2	175	22.50	328	1.574 ± 0.025	1592
LSTM (h=32)	-	-	9.40	8	14.40	12	28.2	24	1.567 ± 0.027	83
Naive	-	-	-	-	-	-	-	-	1.787	-

MSE values are scaled by 10^{-2} and they are always computed on the test set. Epochs and training times correspond to the values required to train a model whose MSE on the test set falls below the specified threshold; an asterisk (*) indicates that at least one run did not reach the threshold.

TABLE III
AVERAGE TRAINING EFFICIENCY AND FINAL ACCURACY ON THE WIND POWER DATASET

Model	# qubits	# layers	MSE ≤ 5.00		MSE ≤ 4.00		MSE ≤ 3.25		Final MSE (130 Epochs)	
			Epochs	Time (s)	Epochs	Time (s)	Epochs	Time (s)	Value	Time (s)
LSTM (h=4)	-	-	93.6	21	*	*	*	*	3.339 ± 0.049	25
QLSTM	6	1	*	*	*	*	*	*	7.887 ± 0.409	1342
QFWP	6	1	*	*	*	*	*	*	5.415 ± 1.445	468
CFE-QFWP	6	1	57	176	69.4	214	83	256	2.369 ± 0.288	415
LSTM (h=5)	-	-	78.2	17	95.6	20	114.4	24	2.778 ± 0.025	27
QLSTM	6	2	*	*	*	*	*	*	6.117 ± 0.540	3564
QFWP	6	2	*	*	*	*	*	*	4.070 ± 0.985	649
CFE-QFWP	6	2	35.4	160	46.4	211	68	311	2.676 ± 0.378	603
QLSTM	8	1	48.33	2859	76.67	4844	*	*	8.008 ± 0.418	3190
QFWP	8	1	*	*	*	*	*	*	6.856 ± 3.104	536
CFE-QFWP	8	1	55.2	239	79	343	*	*	2.615 ± 0.448	570
LSTM (h=32)	-	-	44.6	10	56	12	67.8	15	2.034 ± 0.079	33
Naive	-	-	-	-	-	-	-	-	3.253	-

MSE values are scaled by 10^{-3} and they are always computed on the test set. Epochs and training times correspond to the values required to train a model whose MSE on the test set falls below the specified threshold; an asterisk (*) indicates that at least one run did not reach the threshold.

B. Numerical Results

The results for the Sunspot dataset, averaged over five runs, are reported in Table II. For each model configuration, the table reports the mean number of epochs and wall-clock time required to reach progressively stricter MSE thresholds (i.e., 0.018, 0.017, and 0.016, respectively), together with the final MSE after 100 epochs (mean and standard deviation) and the corresponding training time. The CFE-QFWP achieved a marginal performance improvement over the QFWP (-0.4%) and the LSTM benchmark (-0.57%), while significantly outperforming the QLSTM by 2.9%. Regarding computational efficiency, training times were slightly higher than the QFWP but remained significantly lower than the QLSTM. Notably, the CFE-QFWP demonstrated a convergence rate twice as fast as the classical baseline in terms of test MSE.

Similarly, the results for the Wind Power dataset are shown in Table III. This domain proved more challenging for the quantum models, with both the QFWP and QLSTM failing to surpass the baseline threshold established by the naive predictor. While the CFE-QFWP outperformed classical models of comparable size, it did not reach the performance of the larger LSTM (hidden dimension 32), resulting in a performance gap of 16.47%.

Finally, the results for the Solar Irradiance dataset are presented in Table IV. In this case, the CFE-QFWP outperformed all other models, with a significant reduction in final error compared to QFWP (-29.27%) and QLSTM (-13.63%). Remarkably, it also surpassed the classical LSTM, yielding an error reduction of 2.52%. Furthermore, the model demonstrated superior efficiency, reducing the average number of epochs to converge to the lowest test set threshold by 37.84%.

Overall, the results demonstrate that the CFE-QFWP outperformed the LSTM on two out of three datasets, despite

TABLE IV
AVERAGE TRAINING EFFICIENCY AND FINAL ACCURACY ON THE SOLAR IRRADIANCE DATASET

Model	# qubits	# layers	MSE ≤ 1.70		MSE ≤ 1.60		MSE ≤ 1.50		Final MSE (50 Epochs)	
			Epochs	Time (s)	Epochs	Time (s)	Epochs	Time (s)	Value	Time (s)
LSTM (h=4)	-	-	12.6	24	19.4	37	33.8	65	0.143 ± 0.034	94
QLSTM	6	1	*	*	*	*	*	*	2.752 ± 0.217	4056
QFWP	6	1	*	*	*	*	*	*	1.754 ± 0.114	1314
CFE-QFWP	6	1	4.4	139	5.2	160	10.2	290	1.356 ± 0.056	1303
LSTM (h=5)	-	-	11.6	23	17.4	35	23	46.5	1.446 ± 0.038	27
QLSTM	6	2	*	*	*	*	*	*	0.193 ± 0.129	5802
QFWP	6	2	12.8	479	*	*	*	*	1.570 ± 0.063	1869
CFE-QFWP	6	2	12	415	17	576	30.2	1079	1.406 ± 0.077	1172
QLSTM	8	1	*	*	*	*	*	*	0.254 ± 0.140	6085
QFWP	8	1	*	*	*	*	*	*	1.742 ± 0.087	1745
CFE-QFWP	8	1	4.6	163	5.8	207	9.2	327	1.367 ± 0.026	1760
LSTM (h=32)	-	-	6.4	13	9.6	20	14.8	31	1.391 ± 0.031	105
Naive	-	-	-	-	-	-	-	-	2.513	-

MSE values are scaled by 10^{-2} and they are always computed on the test set. Epochs and training times correspond to the values required to train a model whose MSE on the test set falls below the specified threshold; an asterisk (*) indicates that at least one run did not reach the threshold.

the LSTM possessing approximately $30\times$ more parameters as summarized in Table V. This performance advantage is further reinforced by the model's efficiency; the CFE-QFWP consistently required fewer training epochs to reach the minimum test error threshold compared to its classical counterparts.

TABLE V
ARCHITECTURE DETAILS FOR ALL ADOPTED MODELS

Model	# qubits	# layers	# hidden dim.	# quantum params	# classical params	total params
LSTM	-	-	4	-	117	117
QLSTM	6	1	-	24	6	30
QFWP	6	1	-	6	66	72
CFE-QFWP	6	1	-	6	96	102
QLSTM	6	2	-	48	6	54
QFWP	6	2	-	12	73	85
CFE-QFWP	6	2	-	12	133	145
LSTM	-	-	5	-	166	166
QLSTM	8	1	-	32	8	40
QFWP	8	1	-	8	102	110
CFE-QFWP	8	1	-	8	158	166
LSTM	-	-	32	-	4513	4513
MLP-FWP	-	-	4	-	162	162

C. Discussion

We also conducted an ablation study to assess the role of quantum computation in CFE-QFWP. Specifically, we replaced the VQC in the fast programmer with a two-layer multilayer perceptron (MLP) with a $\tanh(\cdot)$ activation between layers (MLP-FWP). The MLP-FWP was designed to have a parameter count comparable to CFE-QFWP (Table V), ensuring a fair comparison. As shown in Table VI, this replacement causes a clear performance drop, suggesting that the quantum component contributes meaningfully to the effectiveness of the proposed architecture.

Overall, the experimental evidence supports the design choice of this new proposed algorithm. QRNN/QLSTM replicate gated recurrence in the quantum setting, but their BPTT-based training requires deep unrolling and scales as $O(TP_q)$, which clashes with NISQ limitations. QFWP mitigates this by shifting temporal credit assignment to a shallow VQC updated by a classical controller $O(P_q)$, yet its programmer remains input-driven and unaware of the current fast-weight state. CFE-QFWP closes this gap by adding a compact classical feedback signal, making updates explicitly state-dependent while keeping quantum execution shallow. From the experiments, it is evident that CFE-QFWP consistently improves over QLSTM in training efficiency and predictive accuracy, and systematically outperforms the original QFWP with only a modest increase in classical computation. The gains are particularly visible in datasets with structured temporal dynamics and regime shifts, where state-aware fast-weight adaptation appears beneficial (e.g., solar irradiance).

Conversely, in the wind power benchmark, performance remains below the strongest classical baseline, plausibly due to limited data availability and higher randomness, which reduce the statistical signal guiding the update trajectory. From a scalability perspective, the main advantage of CFE-QFWP is that the added recurrence is absorbed by the classical controller, while the quantum component remains shallow

TABLE VI
MSE RESULTS OF THE ABLATION STUDY USING A TWO-LAYER MLP

Model	Dataset	Final MSE
MLP-FWP	Sunspot	$1.569 \times 10^{-2} \pm 7.791 \times 10^{-5}$
CFE-QFWP	Sunspot	$1.558 \times 10^{-2} \pm 9.505 \times 10^{-4}$
LSTM (h=32)	Sunspot	$1.567 \times 10^{-2} \pm 2.704 \times 10^{-5}$
MLP-FWP	Wind Power	$3.522 \times 10^{-3} \pm 5.159 \times 10^{-4}$
CFE-QFWP	Wind Power	$2.369 \times 10^{-3} \pm 2.883 \times 10^{-4}$
LSTM (h=32)	Wind Power	$2.034 \times 10^{-3} \pm 7.867 \times 10^{-5}$
MLP-FWP	Solar Irradiance	$1.571 \times 10^{-2} \pm 1.241 \times 10^{-3}$
CFE-QFWP	Solar Irradiance	$1.356 \times 10^{-2} \pm 5.643 \times 10^{-4}$
LSTM (h=32)	Solar Irradiance	$1.391 \times 10^{-2} \pm 3.142 \times 10^{-4}$

and continues to operate through compact rank-1 parameter updates. Consequently, increasing sequence length does not require the deep quantum unrolling characteristic of QLSTM-style models. On larger or more diverse datasets, the proposed mechanism is expected to be most beneficial when the data exhibit structured temporal regimes, regime switches, or context-dependent dynamics, since these are precisely the settings in which state-aware modulation of the fast weights should provide the greatest advantage. Contrarily, in highly irregular or data-scarce scenarios, as suggested by the wind power benchmark, the benefit of the feedback signal may be reduced by the weaker statistical regularity available to guide the update trajectory.

Finally, the ablation study shows that the gains are not due to the feedback mechanism alone: replacing the VQC with a parameter-matched classical MLP degrades performance, suggesting that the quantum circuit provides a non-trivial inductive bias within the fast-weight framework. Despite these findings, quantum simulation costs still limit extensive hyperparameter tuning, while robustness under realistic noise and extension to multivariate forecasting remain important directions for future work.

V. Conclusion

This paper presented CFE-QFWP, a hybrid forecasting architecture designed to balance temporal modeling capacity with NISQ-era constraints. The core idea is to make fast-weight programming state-aware by feeding the evolving quantum parameter state back into the classical slow programmer. In this way, recurrence is introduced where it is computationally cheaper, while avoiding the cost of applying BPTT to a deeply unrolled quantum model. The update mechanism remains compact through rank-1 parameter increments, preserving the hardware-efficient VQC backbone of QFWP while enabling richer, trajectory-dependent adaptation.

Across the experiments, CFE-QFWP consistently outperforms the original QFWP and QLSTM in forecasting accuracy, and achieves convergence rates comparable to, or better than, a strong LSTM baseline despite using about 30 times fewer parameters. Beyond predictive performance, these results highlight a potential advantage of QML in the context of Green AI. As classical deep learning models continue to grow, their energy demands become increasingly relevant. CFE-QFWP suggests that quantum-enhanced hybrid architectures can achieve strong performance with a much smaller parameter budget, potentially supporting more sustainable scaling and edge deployment. Overall, these results position feedback-enhanced fast-weight programming as a practical route for quantum-enhanced time series learning under near-term hardware constraints.

Several directions remain open for future work. Extending the evaluation to multivariate time series would better assess the model across richer dependencies and application domains. In addition, the feedback pathway could be further refined, for instance through LSTM-style gating to improve information flow over time, stability, and predictive accuracy.

References

[1] P. Gohel and M. Joshi, "Quantum time series forecasting," in *Sixteenth International Conference on Machine Vision (ICMV 2023)*, vol. 13072. SPIE, 2024, pp. 390–398.

[2] L. Bergadano, A. Ceschini, P. Chiavassa, E. Giusto, B. Montrucchio, M. Panella, and A. Rosato, "Q-SCALE: Quantum Computing-Based Sensor Calibration for Advanced Learning and Efficiency," in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 01, 2024, pp. 306–314.

[3] N. Shrestha and S. K. Ghimire, "Comparative analysis of quantum lstm and classical lstm models," in *International Conference on Information and Communication Technology for Intelligent Systems*. Springer, 2025, pp. 255–268.

[4] A. Ceschini, A. Rosato, and M. Panella, "A variational approach to quantum gated recurrent units," *Journal of Physics Communications*, vol. 8, no. 8, p. 085004, aug 2024.

[5] ——, "Design of an LSTM cell on a quantum hardware," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 3, pp. 1822–1826, 2022.

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

[7] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, "General parameter-shift rules for quantum gradients," *Quantum*, vol. 6, p. 677, 2022.

[8] M. Cerezo, A. Arrasmith, R. Babbush *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[9] M. Cerezo, G. Verdon, H.-Y. Huang, L. Cincio, and P. J. Coles, "Challenges and opportunities in quantum machine learning," *Nature computational science*, vol. 2, no. 9, pp. 567–576, 2022.

[10] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, "Parameterized quantum circuits as machine learning models," *Quantum Science and Technology*, vol. 4, no. 4, p. 043001, Nov 2019.

[11] D. Zhu *et al.*, "Training of quantum circuits on a hybrid quantum computer," *Science Advances*, vol. 5, no. 10, p. eaaw9918, 2019.

[12] V. Leyton-Ortega, A. Perdomo-Ortiz, and O. Perdomo, "Robust implementation of generative modeling with parametrized quantum circuits," 2019. [Online]. Available: https://arxiv.org/abs/1901.08047

[13] B. Coyle, D. Mills, V. Danos, and E. Kashefi, "The born supremacy: quantum advantage and training of an Ising Born machine," *npj Quantum Information*, vol. 6, no. 1, Jul. 2020.

[14] Y. Takaki, K. Mitarai, M. Negoro, K. Fujii, and M. Kitagawa, "Learning temporal data with a variational quantum recurrent neural network," *Phys. Rev. A*, vol. 103, p. 052414, May 2021.

[15] M. Guo, Y. Weng, L. Ye, and Y. C. Lai, "Continuous variational quantum algorithms for time series," in *2023 International Joint Conference on Neural Networks (IJCNN)*, 2023, pp. 01–08.

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

[17] Y. Li *et al.*, "Quantum recurrent neural networks for sequential learning," *Neural Networks*, vol. 166, pp. 148–161, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S089360802300360X

[18] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, no. 1, p. 4812, 2018.

[19] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, "Cost function dependent barren plateaus in shallow parametrized quantum circuits," *Nature Communications*, vol. 12, p. 1791, 2021.

[20] J. D. Viqueira, D. Faílde, M. M. Juane, A. Gómez, and D. Mera, "Density matrix emulation of quantum recurrent neural networks for multivariate time series prediction," *Machine Learning: Science and Technology*, vol. 6, no. 1, p. 015023, jan 2025.

[21] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Phys. Rev. A*, vol. 99, p. 032331, Mar 2019.

[22] J. Schmidhuber, "Learning to control fast-weight memories: an alternative to dynamic recurrent networks," *Neural Comput.*, vol. 4, no. 1, p. 131–139, Jan. 1992.

[23] ——, "Reducing the ratio between learning complexity and number of time varying variables in fully recurrent nets," in *ICANN 1993*, ser. Lecture Notes in Computer Science, S. Gielen and H. J. Kappen, Eds. Springer, London, 1993, vol. 635, pp. 460–465.