

Complementary Matrix-Gated QKAN Fast-Weight Programmers for Quantum Dynamics Forecasting

Kuo-Chung Peng^{1,2,*}, Samuel Yen-Chi Chen^{3,*}, Jiun-Cheng Jiang^{1,4,5}, Chen-Yu Liu⁶, En-Jui Kuo⁷, Yun-Yuan Wang⁴, Tzung-Chi Huang⁴, Prayag Tiwari⁸, Chi-Sheng Chen⁹, Chun-Hua Lin^{1,2}, Yu-Chao Hsu^{2,10}, Tai-Yue Li², Saif Al-Kuwari¹², Simon See¹¹, Kuan-Cheng Chen^{12,†}, Nan-Yow Chen^{2,‡}, Hsi-Sheng Goan^{1,5,6,13,§}

¹Department of Physics and Center for Theoretical Physics, National Taiwan University, Taipei, Taiwan

²National Center for High-Performance Computing, National Institutes of Applied Research, Hsinchu, Taiwan

³Wells Fargo, New York, NY, USA

⁴NVIDIA AI Technology Center, NVIDIA Corp., Taipei, Taiwan

⁵Center for Quantum Science and Engineering, National Taiwan University, Taipei, Taiwan

⁶Graduate Institute of Applied Physics, National Taiwan University, Taipei, Taiwan

⁷Department of Electrophysics, National Yang Ming Chiao Tung University, Hsinchu, Taiwan

⁸School of Information Technology, Halmstad University, Sweden

⁹Beth Israel Deaconess Medical Center & Harvard Medical School, Boston, MA, USA

¹⁰School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea

¹¹NVIDIA AI Technology Center, NVIDIA Corp., Singapore, Singapore

¹²Qatar Center for Quantum Computing, College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

¹³Physics Division, National Center for Theoretical Sciences, Taipei, Taiwan

*These authors contributed equally to this work.

Emails: [†]kchen@hbku.edu.qa, [‡]nanyow@nchc.narl.org.tw, [§]goan@phys.ntu.edu.tw

Abstract—Sequence models must decide what to write into memory and what to retain. In quantum and quantum-inspired sequence learning, nonlinear recurrent updates often require repeated circuit evaluations and sequential backpropagation through time, making long contexts costly. Gated fast-weight programmers (FWPs) based on quantum-inspired Kolmogorov–Arnold networks (QKANs) alleviate this bottleneck by storing context in time-varying fast parameters. However, their scalar gate applies one retention–write balance to every fast-state coordinate, forcing all parameters to share a memory timescale. We introduce Self-Modulating QKAN-based FWPs, which replace this broadcast gate with low-rank-generated element-wise modulation of the new-proposal branch, a bounded old-state branch, or both. We further propose Complementary Matrix Gating (CMG), which uses one sigmoid matrix gate to retain the old state and its complement to write the new proposal. CMG provides coordinate-wise memory control while preserving the bounded convex update and affine prefix-scan structure of scalar gating, at the modulation-head cost of a single-branch rule. We compare four self-modulating rules with scalar gating across four FWP architectures combining classical and QKAN-based slow and fast programmers. Across seven single-step forecasting benchmarks and five sequence lengths, CMG gives the most consistent improvements for architectures whose fast programmer incorporates a QKAN-based module. In direct multi-step forecasting of Jaynes–Cummings and transmon–resonator dynamics simulated with CUDA-Q Dynamics, CMG models maintain mean-squared errors on the order of 0.001 or lower across forecasting horizons of 4, 8, and 16 steps, while improving on their scalar-gated counterparts by at least 91.2%.

The views expressed in this article are those of the authors and do not represent the views of Wells Fargo. This article is for informational purposes only. Nothing contained in this article should be construed as investment advice. Wells Fargo makes no express or implied warranties and expressly disclaims all legal, tax, and accounting implications related to this article.

These results establish coordinate-wise complementary modulation as a stable and effective update for QKAN-based FWPs.

Index Terms—fast weight programming, quantum-inspired machine learning, Kolmogorov–Arnold networks, sequence modeling

I. INTRODUCTION

Sequence models must decide what to write into memory and what to retain from the past. Long short-term memory (LSTM) networks and gated recurrent units (GRUs) make this trade-off explicit with input, update, and forget gates [1]–[4]. In quantum machine learning (QML), recurrent memory can be especially expensive: nonlinear recurrent updates generally require unparallelizable, step-by-step circuit evolution, rendering backpropagation through time (BPTT) over long sequences highly time-consuming [5]–[9]. Fast-weight programming offers a more parallelizable alternative by storing temporal context in dynamically updated parameters rather than in a recurrent hidden state [10]–[12]. Quantum Fast Weight Programmers (QFWPs) implement this idea with a slow classical programmer that updates a fast variational quantum circuit (VQC) [13]. Quantum-inspired Kolmogorov–Arnold Network (QKAN)-based Fast-Weight Programmers (QKAN-FWPs) improve scalability by replacing multi-qubit fast circuits with QKAN modules, whose Data Re-Uploading ActivationN (DARUAN) edge functions are single-qubit data re-uploading circuits [14]–[17]. While in ref. [14], scalar gates are stable and parameter-efficient, broadcasting a single retention/write coefficient to every fast-state coordinate, forcing all fast parameters to share the same memory timescale. The

central question of this paper is therefore whether QKAN-FWPs can benefit from coordinate-wise memory control without losing boundedness or the affine parallel-prefix structure that makes fast-weight recurrences efficient [18].

We answer this question by implementing self-modulating fast-state updates in QKAN-FWP and by introducing Complementary Matrix Gating (CMG) as a new self-modulating rule. Following the self-modulating QFWP [19], we let the slow programmer emit low-rank-generated element-wise modulators for the new-update branch, the old-state branch, or both, yielding Only-new, Only-old, and Full variants. We bound the old-state modulation branch with $\tanh$ so each coordinate can retain, suppress, or sign-adjust memory without geometric amplification [20]. CMG, in contrast, ties old-state retention and new-state writing through one sigmoid matrix gate. The gate weights the old fast state, and its complement weights the new proposal. Thus CMG is an element-wise generalization of gated QKAN-FWPs that preserves its bounded convex update and affine scan-compatible form. Empirically, CMG is the most reliable update rule for single-step prediction benchmarks. Across seven benchmarks and five sequence lengths, it gives the most consistent improvements over scalar gating. On Jaynes–Cummings and transmon–resonator quantum dynamics direct multi-step forecasting [21], models that deploy the CMG update rule maintain prediction mean-squared error(MSE) of order 10^{-4} or lower across horizons, improving over gated update rule by at least 91.2%. Only-old and Full are competitive but either lack complementary write control or require two modulation matrices.

In summary, we introduce CMG, a coordinate-wise self-modulating update rule for QKAN-FWPs that preserves bounded convex dynamics and prefix-scan compatibility. Through systematic comparisons with write-side, memory-side, and full self-modulation, we show that coordinate-wise memory control yields more reliable gains than scalar gating, with CMG providing the most consistent improvements across single-step and direct multi-step forecasting benchmarks.

II. RELATED WORK

a) QML sequence models and fast-weight memory:

QML sequence models commonly insert quantum reservoirs, recurrent quantum circuits, or VQCs into classical temporal architectures for time-series forecasting or reinforcement learning tasks [5]–[7], [22]–[25]. These models can be expressive, but long sequence lengths still require repeated temporal circuit evaluation. Fast-weight programming instead accumulates context in adaptive fast parameters [10]–[12]. Self-modulating QFWPs further extend QFWP with element-wise fast-parameter gating [13], [19], [20].

b) KAN/QKAN sequence models and element-wise gating:

Kolmogorov–Arnold networks (KANs) replace fixed node activations and linear weights with learnable univariate edge functions, inspiring KAN forecasters, temporal KAN cells, mixture-of-expert variants, and decomposition or frequency modules for time series [16], [26]–[29]. QKAN substitutes

TABLE I
PROGRAMMER BACKBONES FOR THE FOUR ARCHITECTURAL VARIANTS. “CLASSICAL” DENOTES A MULTILAYER-PERCEPTRON (MLP) SLOW PROGRAMMER OR A LINEAR FAST PROGRAMMER; “HQKAN” DENOTES AN ENCODER–QKAN–DECODER MODULE.

Model	Slow programmer S_ψ	Fast programmer $F(\cdot; \Theta_t)$
FWP	Classical	Classical
QKANFWP	Classical	HQKAN
QKAN-FWP	HQKAN	Classical
QKAN-QKANFWP	HQKAN	HQKAN

spline-style edge functions with DARUAN functions, combining KAN edge-wise function learning with parameter-efficient quantum-inspired nonlinear modeling [15], [17]. Recent QKAN sequence models incorporate QKAN blocks into LSTM-style gates, fast-weight programmers, or transformers [14], [15], [30]–[32]. Coordinate-wise gating is well established in recurrent models, with LSTMs and gated recurrent units using element-wise gates to balance retained memory and candidate states [1], [3]. Later models extend this feature-wise control through lightweight recurrence, adaptive decay, or element-wise attention [33]–[36]. CMG bridges gated QKAN-FWPs and self-modulating QFWPs through low-rank-generated matrix gates with complementary old/new weighting.

III. METHODS

A. QKAN and HQKAN programmer backbones

Fast-weight programming separates a slow programmer from a fast programmer. Let x_t be the scalar input, S_ψ the slow programmer, and $F(\cdot; \Theta_t)$ the fast programmer with time-dependent fast state Θ_t . The slow programmer reads x_t and updates fast states, Θ_t , while the fast programmer predicts with the updated state. Temporal information is stored in the trajectory of $\{\Theta_t\}$ rather than in a recurrent hidden state.

We use either classical linear/multi-layer perceptron (MLP) programmers or QKAN-based programmers. QKAN replaces spline KAN edge functions with DARUAN functions [15]. For an input z , $\phi_\vartheta(z) = \langle 0|U^\dagger(z; \vartheta)\hat{O}U(z; \vartheta)|0\rangle$, where $U(z; \vartheta)$ is a parameterized single-qubit data-reuploading unitary and $\hat{O}$ is the measured observable [17]. In QKAN-based programmers we use a hybrid QKAN (HQKAN) module: a classical encoder maps inputs to a latent space, a QKAN block applies nonlinear transformations, and a classical decoder maps back to the required output dimension. Thus the “QKAN layer” in Fig. 1 denotes the quantum-inspired core inside an encoder–QKAN–decoder module. Table I lists the four architectures: fully classical FWP, QKANFWP with an HQKAN fast programmer, QKAN-FWP with an HQKAN slow programmer, and QKAN-QKANFWP with HQKAN in both roles.

B. Gated fast-weight update

For a generic fast state Θ_t , the slow programmer first proposes $\Delta_t = S_\psi^\Delta(x_t)$, with the same shape as Θ_t . The

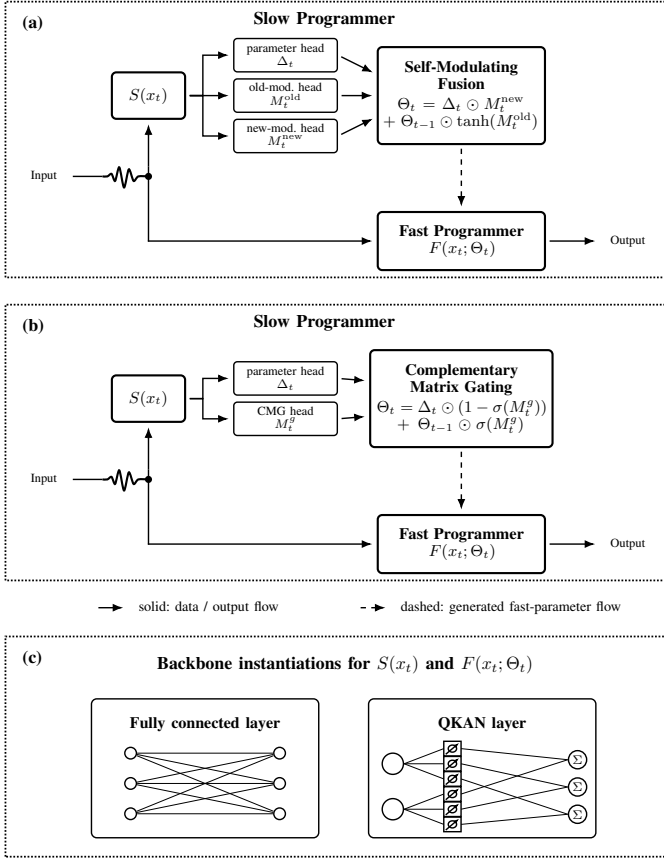


Fig. 1. Self-modulating update rules and programmer backbones. (a) Full self-modulation uses separate old- and new-modulation heads. (b) CMG uses a single low-rank-generated matrix gate with complementary weights for the retained and proposed states. (c) The slow and fast programmers may use classical fully connected modules or QKAN-based HQKAN modules. Solid arrows show data/output flow. Dashed arrows show generated fast-parameter flow.

slow programmer of the gated baseline [14] also produces $g_t = \sigma(S_{\psi}^g(x_t))$, $g_t \in [0, 1]$, and updates

$$\Theta_t = g_t \Theta_{t-1} + (1 - g_t) \Delta_t. \quad (1)$$

C. Low-rank self-modulating updates

Self-modulation replaces the scalar gate with element-wise modulation. For any architecture in Table I, the fast state is reshaped to $\Theta_t \in \mathbb{R}^{P \times Q}$. For each branch $r \in \{\text{new}, \text{old}\}$, the slow programmer generates a rank-one modulation matrix from the outer product of two affine-head outputs:

$$M_t^r = m_t^{r,P} \left(m_t^{r,Q} \right)^\top, \quad m_t^{r,P} \in \mathbb{R}^P, \quad m_t^{r,Q} \in \mathbb{R}^Q.$$

Following self-modulating QFWP [19], the raw update family is

$$\begin{aligned} \Theta_t &= \Delta_t \odot M_t^{\text{new}} + \Theta_{t-1} \odot M_t^{\text{old}}, & \text{Full,} \\ \Theta_t &= \Delta_t \odot M_t^{\text{new}} + \Theta_{t-1}, & \text{Only-new,} \\ \Theta_t &= \Delta_t + \Theta_{t-1} \odot M_t^{\text{old}}, & \text{Only-old.} \end{aligned}$$

The new branch controls write amplitude; the old branch controls coordinate-wise retention, attenuation, amplification, or sign reversal.

Because old-state modulation is multiplied recurrently, we bound it by

$$\widetilde{M}_t^{\text{old}} = \tanh(M_t^{\text{old}}), \quad |\widetilde{M}_{t,pq}^{\text{old}}| \leq 1.$$

The bounded Full and Only-old updates are

$$\begin{aligned} \Theta_t &= \Delta_t \odot M_t^{\text{new}} + \Theta_{t-1} \odot \widetilde{M}_t^{\text{old}}, & \text{bounded Full,} \\ \Theta_t &= \Delta_t + \Theta_{t-1} \odot \widetilde{M}_t^{\text{old}}, & \text{bounded Only-old.} \end{aligned}$$

For coordinate $j = (p, q)$, with $a_{t,j} = \widetilde{M}_{t,j}^{\text{old}}$ and $d_{t,j} = [\Delta_t]_j$, bounded Only-old unrolls as

$$\theta_{t,j} = \theta_{0,j} \prod_{u=1}^t a_{u,j} + \sum_{s=1}^t d_{s,j} \prod_{u=s+1}^t a_{u,j}, \quad |a_{u,j}| \leq 1.$$

Thus the recurrent kernel cannot amplify stored updates through products of old-state multipliers. The same expression applies to bounded Full after replacing $d_{s,j}$ by $[\Delta_s \odot M_s^{\text{new}}]_j$.

D. Complementary Matrix Gating

CMG uses the same low-rank factorization but generates one gate matrix,

$$M_t^g = m_t^{g,P} \left(m_t^{g,Q} \right)^\top, \quad G_t = \sigma(M_t^g) \in [0, 1]^{P \times Q}.$$

The update is

$$\Theta_t = G_t \odot \Theta_{t-1} + (1 - G_t) \odot \Delta_t.$$

Equivalently, CMG replaces the scalar gate in Eq. (1) with an element-wise matrix gate. This is the new self-modulating rule proposed in this paper: the slow programmer generates the gate, but the old and new branches remain complementary. Unlike Full self-modulation, CMG uses one modulation matrix, giving the same modulation-head size as Only-new and Only-old.

CMG inherits the scalar gate's bounded convex update coordinate-wise. For coordinate j , let $g_{t,j} \in [0, 1]$ and $d_{t,j} = [\Delta_t]_j$. Then

$$\theta_{t,j} = g_{t,j} \theta_{t-1,j} + (1 - g_{t,j}) d_{t,j},$$

and

$$\theta_{t,j} = \theta_{0,j} \prod_{u=1}^t g_{u,j} + \sum_{s=1}^t (1 - g_{s,j}) d_{s,j} \prod_{u=s+1}^t g_{u,j}. \quad (2)$$

The nonnegative update weights in Eq. (2) sum to $1 - \prod_{u=1}^t g_{u,j} \leq 1$; including the initial-state coefficient gives a convex combination. Hence, under the same bounded-proposal condition used for scalar gating, each coordinate remains within the same bound without requiring a global scalar gate [14].

All update rules considered here can still be evaluated by parallel prefix scan. After the slow-programmer heads are computed for all time steps, each update has the affine form

$$\Theta_t = A_t \odot \Theta_{t-1} + B_t.$$

For Only-new, $A_t = 1$ and $B_t = \Delta_t \odot M_t^{\text{new}}$; for Only-old, $A_t = \widetilde{M}_t^{\text{old}}$ and $B_t = \Delta_t$; for Full, $A_t = \widetilde{M}_t^{\text{old}}$ and

$B_t = \Delta_t \odot M_t^{\text{new}}$; for CMG, $A_t = G_t$ and $B_t = (1 - G_t) \odot \Delta_t$. The pairs compose associatively,

$$(A', B') \circ (A, B) = (A' \odot A, A' \odot B + B'),$$

so the full fast-state trajectory can be obtained with the same affine prefix-scan principle used in fast-weight recurrences [14].

IV. EXPERIMENTAL PROTOCOL

A. Tasks and preprocessing

We evaluate seven univariate time-series benchmarks and follow the same setup used in prior works [14], [20]. Each scalar sequence is min–max normalized to $[-1, 1]$, converted to chronological sliding-window samples, and split into 80% train and 20% test sets.

a) *Damped simple harmonic motion*: The damped simple harmonic motion (SHM) task uses the angular velocity of a nonlinear damped pendulum, $\frac{d^2\theta}{dt^2} + \frac{b}{m} \frac{d\theta}{dt} + \frac{g}{L} \sin \theta = 0$, with $g = 9.81$, $b = 0.15$, $L = m = 1$, $\theta(0) = 0$, and $\dot{\theta}(0) = 3$. It tests smooth damped oscillatory prediction.

b) *Bessel function*: The Bessel task uses the second-order Bessel function of the first kind, $J_2(x)$, to test nonlinear approximation with changing amplitude and phase.

c) *NARMA-5 and NARMA-10*: The nonlinear autoregressive moving-average (NARMA) tasks use memory orders $n = 5$ and $n = 10$ with the standard recurrence $y_{t+1} = \alpha y_t + \beta y_t \sum_{j=0}^{n-1} y_{t-j} + \gamma u_{t-n+1} u_t + \delta$, $n \in \{5, 10\}$.

d) *Delayed quantum control*: The delayed quantum control (DQC) task is a non-Markovian feedback-like signal formed by decaying localized pulses, $x(t) = \sum_{n=0}^{10} \exp[-10(t - 2n)^2] \exp(-t/16)$, $t \in [-2, 20]$.

e) *Open Jaynes–Cummings dynamics*: This benchmark is an open Jaynes–Cummings system with a two-level qubit coupled to a cavity truncated to 5 Fock levels, simulated with CUDA-Q Dynamics [21]. The Hamiltonian is

$$H = \omega_c a^\dagger a + \omega_q \sigma_+ \sigma_- + g(\sigma_- a^\dagger + \sigma_+ a),$$

with $\omega_c = \omega_q = 2\pi$ and $g = \pi$. Photon loss uses $C = \sqrt{\gamma} a$ with $\gamma = 0.05$. The initial state is $\rho_0 = |g, 1\rangle\langle g, 1|$, and the target observable is qubit excitation expectation probability $\langle \sigma_+ \sigma_- \rangle(t)$ for $t \in [0, 50]$.

f) *Dispersive transmon–resonator dynamics*: The second task is a closed dispersive transmon–resonator model with a two-level transmon and a resonator truncated to 20 Fock levels, also simulated with CUDA-Q Dynamics. The Hamiltonian is

$$H = \frac{1}{2} \omega'_{01} \sigma_z + (\omega'_r + \chi \sigma_z) a^\dagger a.$$

We use $\omega_{01} = 3.0 \cdot 2\pi$ GHz, $\omega_r = 2.0 \cdot 2\pi$ GHz, $\chi = 0.025 \cdot 2\pi$ GHz, $\omega'_{01} = \omega_{01} + \chi$, and $\omega'_r = \omega_r$. The initial state is $(|0\rangle + |1\rangle)/\sqrt{2}$ for the transmon and $|\alpha = 2.0\rangle$ for the resonator. The target is the expectation value of the resonator’s position quadrature operator $\langle \hat{x} \rangle(t)$ for $t \in [0, 25]$ ns. Both CUDA-Q Dynamics trajectories contain 3000 equally spaced time steps.

TABLE II
BEST MODELS AT $N = 16$ FOR EACH DATASET. FINAL TEST MSE IS REPORTED AS MEAN (STANDARD DEVIATION) OVER FIVE SEEDS.

Dataset	Selected arm	Family	Rule	Epoch-100 test MSE
Bessel	CMG QKAN-QKANFWP	QKAN-QKANFWP	CMG	$4.94(6.7) \times 10^{-6}$
Damped SHM	CMG FWP	FWP	CMG	$1.42(1.2) \times 10^{-5}$
Narma5	G-QKANFWP [14]	QKANFWP	Gated	$5.21(3.9) \times 10^{-6}$
Narma10	Only-new QKANFWP	QKANFWP	Only New	$2.55(2.5) \times 10^{-6}$
DQC	CMG QKANFWP	QKANFWP	CMG	$6.94(3.5) \times 10^{-6}$
Jaynes–Cummings	CMG QKAN-QKANFWP	QKAN-QKANFWP	CMG	$3.18(1.6) \times 10^{-6}$
transmon-resonator	CMG QKAN-QKANFWP	QKAN-QKANFWP	CMG	$2.65(1.9) \times 10^{-6}$

TABLE III
FINAL TEST MSE OF THE SELECTED MODELS ON THE SMOOTH-DYNAMICS BENCHMARKS FOR $N = \{4, 8, 32, 64\}$. VALUES ARE REPORTED AS MEAN (STANDARD DEVIATION) OVER FIVE SEEDS. BEST/SECOND-BEST RESULTS ARE SHOWN IN BOLD/UNDERLINED.

Model	$N = 4$	$N = 8$	$N = 32$	$N = 64$
<i>Bessel</i>				
CMG QKAN-QKANFWP	$6.24(3.3) \times 10^{-6}$	$1.86(3.2) \times 10^{-5}$	$1.92(2.1) \times 10^{-5}$	$8.37(5.3) \times 10^{-6}$
CMG FWP	$3.23(2.6) \times 10^{-5}$	<u>$1.86(1.9) \times 10^{-5}$</u>	$6.07(11.7) \times 10^{-4}$	$2.04(3.1) \times 10^{-5}$
G-QKANFWP [14]	$8.15(15.3) \times 10^{-4}$	$4.08(5.0) \times 10^{-5}$	$1.23(1.5) \times 10^{-3}$	$1.47(1.8) \times 10^{-3}$
Only-new QKANFWP	$1.49(1.3) \times 10^{-5}$	$6.38(11.4) \times 10^{-5}$	$3.53(0.3) \times 10^{-3}$	$3.45(0.6) \times 10^{-3}$
CMG QKANFWP	<u>$1.67(1.2) \times 10^{-5}$</u>	$8.73(4.6) \times 10^{-6}$	$2.90(2.2) \times 10^{-6}$	$6.37(11.1) \times 10^{-6}$
<i>Damped SHM</i>				
CMG QKAN-QKANFWP	$1.16(1.3) \times 10^{-4}$	$1.43(1.4) \times 10^{-5}$	$3.60(1.4) \times 10^{-6}$	$1.87(2.7) \times 10^{-5}$
CMG FWP	$4.41(6.5) \times 10^{-4}$	$3.24(2.4) \times 10^{-5}$	$2.85(2.9) \times 10^{-5}$	$6.00(7.6) \times 10^{-5}$
G-QKANFWP [14]	$1.14(1.2) \times 10^{-3}$	$1.14(1.2) \times 10^{-3}$	$1.03(1.2) \times 10^{-3}$	$1.21(1.4) \times 10^{-3}$
Only-new QKANFWP	$1.18(1.4) \times 10^{-4}$	$3.87(4.2) \times 10^{-4}$	$2.08(1.9) \times 10^{-4}$	$9.04(6.9) \times 10^{-5}$
CMG QKANFWP	$5.97(2.5) \times 10^{-5}$	$1.17(0.7) \times 10^{-5}$	<u>$2.51(3.7) \times 10^{-5}$</u>	$1.00(1.1) \times 10^{-5}$

B. Single-step prediction

For input sequence length N , each input is $\mathbf{x}_{t,N} = [x_{t-N}, \dots, x_{t-1}]$ and the target is x_t . The model processes the N observations sequentially, updates the fast state at each internal step, and predicts after the final observation. We evaluate $N \in \{4, 8, 16, 32, 64\}$ to test how each update rule retains information as sequence length grows.

C. Direct multi-step forecasting

For direct forecasting, the input length is fixed to $N = 64$ and the model outputs the entire horizon in one forward pass: $\mathbf{x}_{t,64} = [x_{t-64}, \dots, x_{t-1}]$ and $\mathbf{y}_{t,H} = [x_t, x_{t+1}, \dots, x_{t+H-1}]$, with $H \in \{4, 8, 16\}$. The protocol avoids recursive feedback of predictions and isolates whether the fast state contains enough history for longer-horizon prediction.

D. Training and evaluation

We use a GPU-efficient FlashQKAN implementation in PyTorch [37], accelerated with CuTe DSL [38] for fused operators and block tiling on CUDA devices, building on the open-source QKAN repository [39]. For each dataset, architecture, update rule, sequence length, and horizon, we train with five independent random seeds for 100 epochs using Adam [40] with learning rate 10^{-3} and batch size 4. The primary metric is MSE. For multi-step forecasting, MSE over the horizon is calculated as $\text{MSE} = \frac{1}{H} \sum_{h=0}^{H-1} (\hat{x}_{t+h} - x_{t+h})^2$, then averaged across samples and seeds. Relative improvement over the scalar-gated baseline is

$$\Delta_{\text{rel}} = \frac{M_{\text{gated}} - M_{\text{self-modulated}}}{M_{\text{gated}} + 10^{-12}}. \quad (3)$$

Positive values indicate lower MSE than the corresponding scalar-gated model.

TABLE IV

FINAL TEST MSE OF THE SELECTED MODELS ON THE NARMA BENCHMARKS FOR $N = \{4, 8, 32, 64\}$. VALUES ARE REPORTED AS MEAN (STANDARD DEVIATION) OVER FIVE SEEDS. BEST/SECOND-BEST RESULTS ARE SHOWN IN BOLD/UNDERLINED.

Model	$N = 4$	$N = 8$	$N = 32$	$N = 64$
<i>Narma-5</i>				
CMG QKAN-QKANFWP	$3.15(1.8) \times 10^{-5}$	$1.90(1.9) \times 10^{-5}$	$1.34(1.1) \times 10^{-5}$	$1.57(1.2) \times 10^{-5}$
CMG FWP	$2.45(1.1) \times 10^{-5}$	$1.32(1.3) \times 10^{-5}$	$1.59(1.6) \times 10^{-5}$	$1.48(0.6) \times 10^{-5}$
G-QKANFWP [14]	$1.70(0.9) \times 10^{-5}$	$1.05(1.2) \times 10^{-5}$	$1.99(2.2) \times 10^{-5}$	$1.38(0.7) \times 10^{-5}$
Only-new QKANFWP	$2.04(0.7) \times 10^{-5}$	$4.91(6.7) \times 10^{-6}$	$5.90(3.1) \times 10^{-5}$	$4.72(1.5) \times 10^{-5}$
CMG QKANFWP	<u>$2.62(0.8) \times 10^{-5}$</u>	<u>$8.94(10.0) \times 10^{-6}$</u>	$7.13(3.6) \times 10^{-6}$	$9.58(5.0) \times 10^{-6}$
<i>Narma-10</i>				
CMG QKAN-QKANFWP	$1.16(0.5) \times 10^{-4}$	$4.60(3.1) \times 10^{-5}$	$2.09(3.0) \times 10^{-5}$	$2.84(3.4) \times 10^{-5}$
CMG FWP	<u>$1.14(0.3) \times 10^{-4}$</u>	$2.25(0.5) \times 10^{-5}$	$2.47(2.1) \times 10^{-5}$	$1.57(0.9) \times 10^{-5}$
G-QKANFWP [14]	$8.36(2.3) \times 10^{-5}$	$4.79(1.9) \times 10^{-5}$	$1.33(0.7) \times 10^{-5}$	$1.50(0.5) \times 10^{-5}$
Only-new QKANFWP	$1.17(0.4) \times 10^{-4}$	$4.04(2.1) \times 10^{-5}$	$1.22(0.6) \times 10^{-4}$	$8.39(1.5) \times 10^{-5}$
CMG QKANFWP	$1.37(0.7) \times 10^{-4}$	<u>$2.68(0.9) \times 10^{-5}$</u>	<u>$1.48(1.1) \times 10^{-5}$</u>	$1.36(0.5) \times 10^{-5}$

TABLE V

FINAL TEST MSE OF THE SELECTED MODELS ON THE QUANTUM-DYNAMICS BENCHMARKS FOR $N = \{4, 8, 32, 64\}$. VALUES ARE REPORTED AS MEAN (STANDARD DEVIATION) OVER FIVE SEEDS. BEST/SECOND-BEST RESULTS ARE SHOWN IN BOLD/UNDERLINED.

Model	$N = 4$	$N = 8$	$N = 32$	$N = 64$
<i>DQC</i>				
CMG QKAN-QKANFWP	$4.02(2.8) \times 10^{-6}$	$3.52(1.9) \times 10^{-6}$	$7.82(4.0) \times 10^{-6}$	$4.77(1.2) \times 10^{-6}$
CMG FWP	$2.44(2.8) \times 10^{-5}$	$1.07(1.3) \times 10^{-5}$	$1.16(0.7) \times 10^{-5}$	$1.96(1.9) \times 10^{-5}$
G-QKANFWP [14]	$6.42(6.9) \times 10^{-5}$	$1.21(0.1) \times 10^{-4}$	$1.19(0.2) \times 10^{-4}$	$1.47(0.2) \times 10^{-4}$
Only-new QKANFWP	$1.24(0.7) \times 10^{-5}$	$1.10(1.0) \times 10^{-5}$	$8.81(9.0) \times 10^{-5}$	$3.33(1.9) \times 10^{-5}$
CMG QKANFWP	<u>$4.04(1.8) \times 10^{-6}$</u>	<u>$4.57(3.4) \times 10^{-6}$</u>	$1.19(1.1) \times 10^{-5}$	<u>$1.44(1.0) \times 10^{-5}$</u>
<i>Jaynes-Cummings</i>				
CMG QKAN-QKANFWP	$5.42(2.8) \times 10^{-5}$	$3.17(1.5) \times 10^{-5}$	$1.04(1.3) \times 10^{-5}$	$1.47(1.6) \times 10^{-5}$
CMG FWP	$4.31(2.9) \times 10^{-5}$	$1.66(0.9) \times 10^{-5}$	$5.21(3.3) \times 10^{-5}$	$1.04(1.1) \times 10^{-4}$
G-QKANFWP [14]	$4.27(3.1) \times 10^{-4}$	$7.02(1.0) \times 10^{-4}$	$7.30(0.9) \times 10^{-4}$	$7.98(1.4) \times 10^{-4}$
Only-new QKANFWP	$3.56(3.3) \times 10^{-5}$	$1.40(1.2) \times 10^{-4}$	$1.26(0.8) \times 10^{-3}$	$1.53(0.8) \times 10^{-3}$
CMG QKANFWP	$3.47(3.4) \times 10^{-5}$	$3.98(3.1) \times 10^{-5}$	$9.10(10.8) \times 10^{-6}$	$7.04(7.0) \times 10^{-6}$
<i>transmon-resonator</i>				
CMG QKAN-QKANFWP	$1.08(0.9) \times 10^{-5}$	$1.10(1.5) \times 10^{-5}$	$8.89(11.1) \times 10^{-6}$	$6.90(6.3) \times 10^{-6}$
CMG FWP	$2.35(1.9) \times 10^{-5}$	$4.97(7.3) \times 10^{-5}$	$1.68(1.5) \times 10^{-5}$	$1.78(1.5) \times 10^{-5}$
G-QKANFWP	$2.75(2.2) \times 10^{-3}$	$4.67(0.2) \times 10^{-3}$	$3.74(1.9) \times 10^{-3}$	$3.77(1.9) \times 10^{-3}$
Only-new QKANFWP	$1.56(0.5) \times 10^{-5}$	$2.14(1.2) \times 10^{-5}$	$3.80(3.7) \times 10^{-5}$	$4.12(1.6) \times 10^{-3}$
CMG QKANFWP	<u>$4.64(6.8) \times 10^{-5}$</u>	$2.49(1.4) \times 10^{-6}$	$7.09(3.4) \times 10^{-6}$	$6.40(4.6) \times 10^{-6}$

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. Single-step prediction

We first evaluate all update rules across the four model families and sequence lengths $N \in \{4, 8, 16, 32, 64\}$. Figure 2 establishes that CMG provides the most consistent improvement over the scalar-gated baseline. The paired scatter plots in Fig. 3 support the same conclusion. CMG produces the cleanest shift below the diagonal, especially for QKANFWP and QKAN-QKANFWP. In contrast, Only-new has many failures above the diagonal, Only-old mainly confirms the value of memory-side modulation, and Full shows that adding both old- and new-side modulation is not always necessary. QKAN-FWP is the main exception. Its scalar-gated form is already competitive, and the self-modulation variants provide only marginal or inconsistent gains. Thus, CMG’s benefit is best interpreted as stable coordinate-wise old/new balancing that is most useful when the quantum-inspired structure is used directly in the fast programmer.

Following prior experimental conventions [14], we isolate the top-performing model at $N = 16$ for each dataset and re-evaluate these selected arms across the remaining sequence lengths as reported in Tables II to V. The cross-sequence results strongly favor the QKAN-based backbones over classical FWP. Across the 28 configurations, CMG variants achieve the best MSE in 24 cases. Specifically, CMG QKAN-QKANFWP and CMG QKANFWP decisively outperform classical FWP

TABLE VI

PARAMETER COUNTS AND AREA UNDER THE TEST-MSE LEARNING CURVE (AULC) FOR SINGLE-STEP FORECASTING AT $N = 64$. AULC VALUES ARE AVERAGED OVER THE JAYNES-CUMMINGS AND TRANSMON-RESONATOR DATASETS AND FIVE SEEDS. PARAMETER RATIOS ARE RELATIVE TO THE CORRESPONDING GATED MODEL. LOWER AULC IS BETTER.

Model	Update	Params	Params/Gated	Test-MSE AULC $\downarrow$
FWP	Only-old	209	152.6%	2.41×10^{-4}
	Full	299	218.2%	2.86×10^{-4}
	CMG	209	152.6%	9.17×10^{-5}
QKANFWP	Only-old	210	162.8%	9.48×10^{-5}
	Full	300	232.6%	1.02×10^{-4}
	CMG	210	162.8%	4.45×10^{-5}
QKAN-QKANFWP	Only-old	234	139.3%	1.20×10^{-4}
	Full	306	182.1%	1.37×10^{-4}
	CMG	234	139.3%	2.99×10^{-5}

across most tasks. While the NARMA datasets exhibit slight heterogeneity at shorter sequences, CMG QKANFWP still achieves the best MSE at the longest sequence length ($N = 64$), confirming that CMG becomes increasingly critical as the effective context window grows.

Finally, we analyze learning dynamics at the longest sequence length ($N = 64$) on the two CUDA-Q dynamics datasets. Figure 4 shows that CMG’s advantage is not merely a final-epoch artifact. On Jaynes-Cummings, CMG stays among the lowest-error rules after early training. On transmon-resonator, it reaches the low-error regime quickly and remains close to Only-old and Full through epoch 100. In contrast, scalar gating and Only-new remain in a higher-error regime, especially on transmon-resonator. Table VI further contextualizes this by comparing parameter counts and the Area Under the Learning Curve (AULC) [41], [42] for the self-modulating variants. CMG matches the parameter efficiency of Only-old while requiring fewer parameters than Full self-modulation. In these benchmarks, CMG QKANFWP and CMG QKAN-QKANFWP yield the lowest test-MSE AULC, indicating lower average error throughout training. Based on the single-step results, we select the two top-performing models, QKANFWP and QKAN-QKANFWP for the subsequent direct multi-step prediction tasks.

B. Direct multi-step prediction

We next evaluate direct multi-step forecasting on the two CUDA-Q Dynamics datasets with $N = 64$ and $H \in \{4, 8, 16\}$. We focus on QKANFWP and QKAN-QKANFWP because the single-step study identifies them as the most robust HQKAN-based families.

The heatmaps in Figs. 5 and 6 show that memory-side element-wise updates remain strong in long-horizon forecasting. For CMG, every QKANFWP and QKAN-QKANFWP cell is no worse than 9.3×10^{-4} , placing the direct expectation-value forecasts at the 10^{-4} scale rather than the 10^{-2} scale of scalar gating. Only-old, Full, and CMG reduce MSE by orders of magnitude over scalar gating on both quantum systems, whereas Only-new is worse than the gated baseline in every multi-step setting. CMG is the best rule in five of the twelve

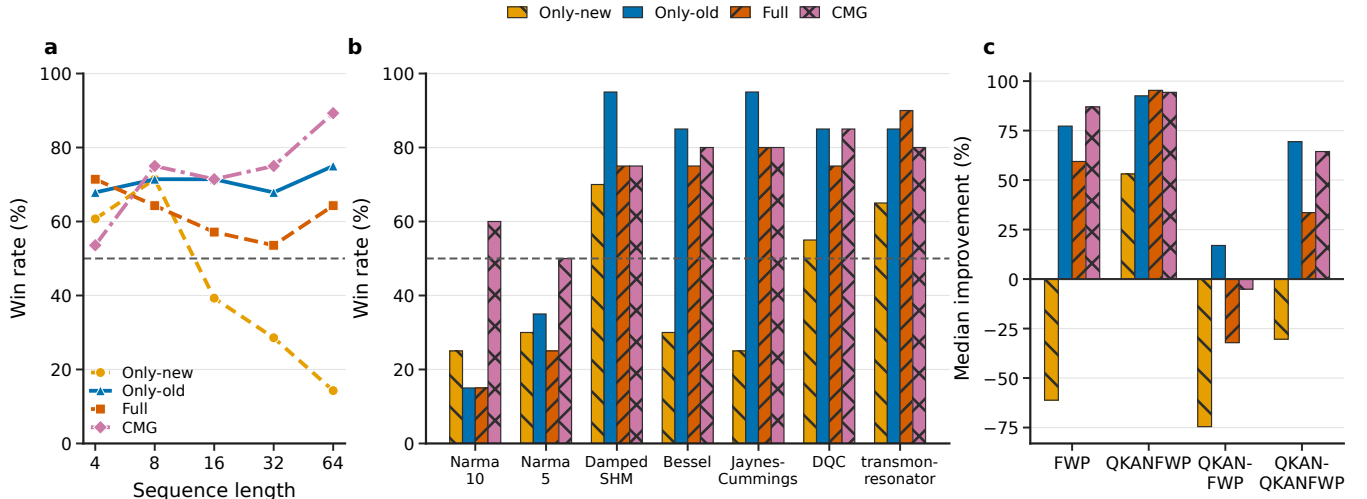


Fig. 2. Performance of the self-modulating update rules relative to their paired gated baselines. (a) Win rate by sequence length, aggregated across datasets and model families. (b) Win rate by dataset, aggregated across sequence lengths and model families. (c) Median relative MSE improvement [Eq. (3)] by model family, aggregated across datasets and sequence lengths. Dashed lines in (a) and (b) mark a 50% win rate; the horizontal line in (c) marks zero improvement.

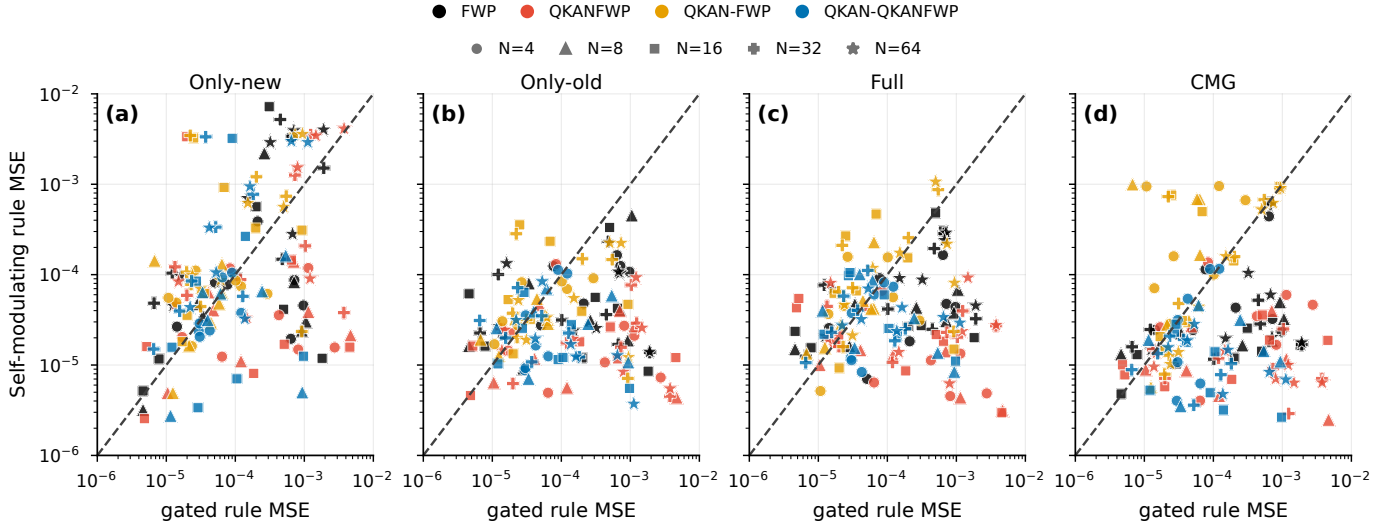


Fig. 3. Final test MSE of each self-modulating rule versus its paired gated baseline across all datasets, sequence lengths, and model families: (a) Only-new, (b) Only-old, (c) Full, and (d) CMG. The diagonal indicates equal MSE; points below it favor self-modulation. Colors denote model families, and markers denote sequence lengths.

dataset-horizon cells: QKANFWP on Jaynes–Cummings at $H = 8$, QKANFWP on transmon–resonator at $H = 4, 8$, QKAN-QKANFWP on Jaynes–Cummings at $H = 4$, and QKAN-QKANFWP on transmon–resonator at $H = 16$. Full is best in four cells, and Only-old in three. Thus CMG reaches the same competitive regime with the same modulation-head size as Only-old and fewer heads than Full. The relative-improvement heatmap clarifies the Only-new failure mode. For QKAN-QKANFWP, Only-new worsens Jaynes–Cummings by 64.8%, 124%, and 193% for $H = 4, 8, 16$, and worsens transmon–resonator by 66.4%, 94.7%, and 223%. In contrast, CMG achieves at least 91.2% relative improvement across all multi-step cells and reaches 99.9%–100% improvement on transmon–resonator. These results indicate that direct forecasting benefits from stable retention or complementary retention/write gating, rather than write-only modulation.

The qualitative forecasts in Fig. 7 match the quantitative results. Full and CMG QKAN-QKANFWP track the held-out oscillatory trajectory with accurate phase and amplitude over non-overlapping forecast blocks. The gated model captures the broad oscillation but shows larger phase/amplitude mismatch and wider seed variation. CMG remains visually comparable to Full despite using only one matrix-gating head.

VI. CONCLUSION

We presented Self-Modulating QKAN-FWP, a framework that equips quantum-inspired fast-weight programmers with coordinate-wise memory control, and CMG as its central update rule. CMG generalizes the scalar gate of gated QKAN-FWPs into a single low-rank-generated matrix gate whose complement writes the new proposal. It therefore preserves the two properties that make scalar gating attractive—coordinate-wise bounded-convex stability and affine parallel-prefix-scan

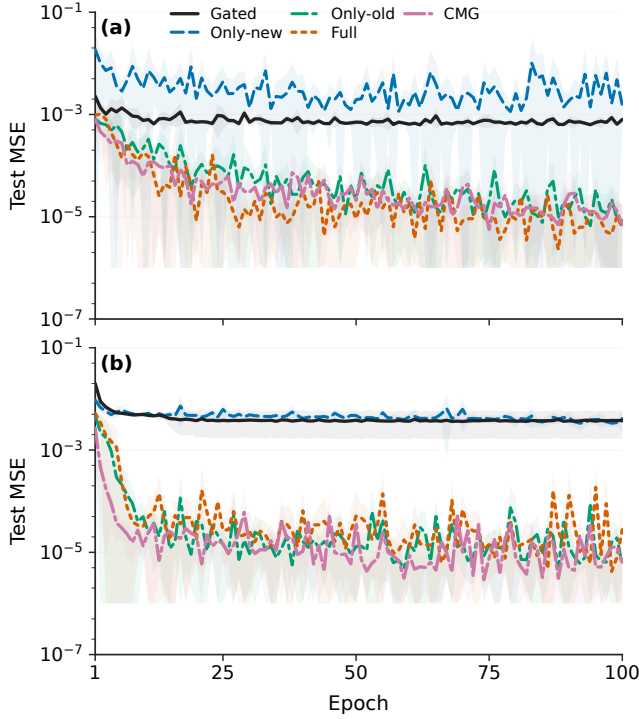


Fig. 4. Test MSE versus training epoch for QKANFWP at $N = 64$: (a) Jaynes-Cummings and (b) transmon-resonator. Curves show the mean over five seeds, and shaded bands show ± 1 standard deviation.

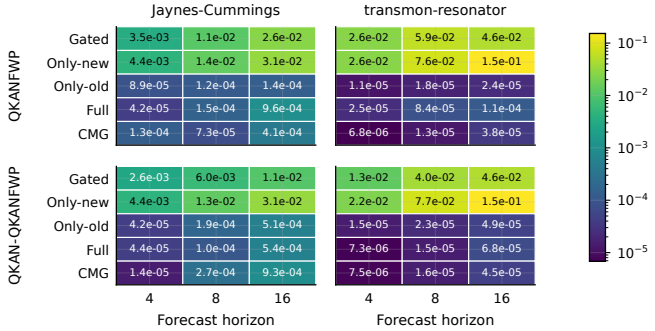


Fig. 5. Mean test MSE for direct multi-step forecasting with QKANFWP and QKAN-QKANFWP at $N = 64$, separated by dataset, forecast horizon, and update rule. Values are averaged over five seeds; colors encode MSE on a logarithmic scale. CMG uses one low-rank-generated gate matrix, whereas Full uses separate old- and new-modulation matrices.

compatibility—while relaxing the constraint that one coefficient govern every fast-state coordinate, at the modulation-head cost of Only-old and below that of Full self-modulation.

Across seven single-step benchmarks and five sequence lengths, CMG is the most consistent update rule for QKANFWP and QKAN-QKANFWP, and in direct multi-step forecasting of Jaynes-Cummings and transmon-resonator dynamics it holds expectation-value error at the 10^{-4} scale or below across $H \in \{4, 8, 16\}$, improving on scalar gating by at least 91.2%. Together with the failure of write-only modulation at long horizons, these results indicate that the decisive ingredient is not additional modulation capacity but a stable, complementary balance between retained memory and new writes.

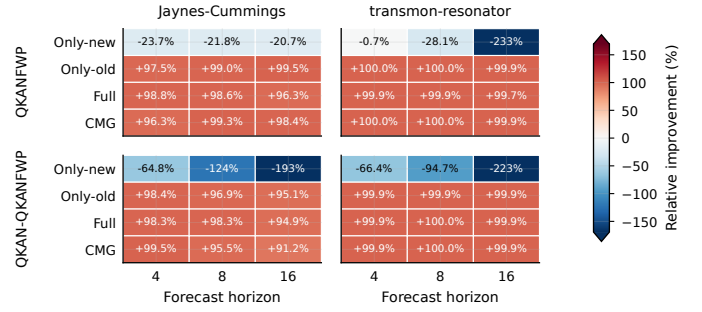


Fig. 6. Relative MSE improvement over the paired gated baseline [Eq. (3)] for direct multi-step forecasting at $N = 64$. Rows group update rules within QKANFWP and QKAN-QKANFWP; columns group forecast horizons within the Jaynes-Cummings and transmon-resonator datasets. Values are computed from mean MSE over five seeds. Red denotes improvement, whereas blue denotes degradation.

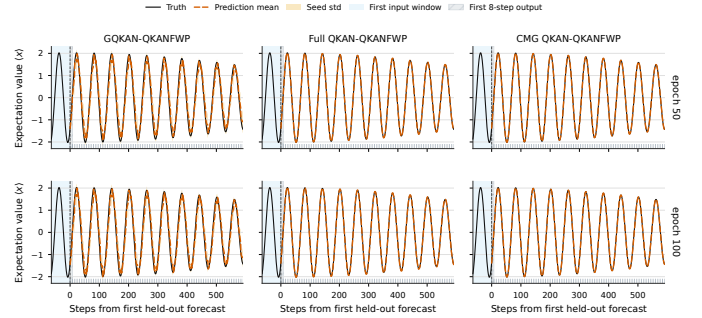


Fig. 7. Direct multi-step forecasts on the transmon-resonator test set for QKAN-QKANFWP with $N = 64$ and $H = 8$. Columns compare scalar-gated, Full, and CMG updates; rows show model checkpoints after 50 and 100 epochs. Curves show the mean prediction over five seeds with ± 1 standard-deviation bands and the ground truth. Shading identifies the first input window and first eight-step output block.

The present formulation targets univariate temporal memory and does not explicitly represent spatial structure among variables. Extending complementary matrix gating to structured spatio-temporal fast states is a natural next step toward multi-variate physical dynamics and quantum-control forecasting.

ACKNOWLEDGMENT

K.-C. Peng, J.-C. Jiang, Y.-C. Hsu and C.-H. Lin thank the National Center for High-Performance Computing (NCHC), National Institutes of Applied Research (NIAR), Taiwan, for providing computational and storage resources supported by the National Science and Technology Council (NSTC), Taiwan, under Grants No. NSTC 114-2119-M-007-013. H.-S. Goan acknowledges support from the NSTC, Taiwan, under Grants No. NSTC 113-2112-M-002-022-MY3, No. NSTC 113-2119-M-002-021, No. NSTC 114-2119-M-002-018, No. NSTC 114-2119-M-002-017-MY3, and from the National Taiwan University under Grants No. NTU-CC-115L8937, No. NTU-CC-115L893704 and No. NTU-CC-115L8512. H.-S. Goan is also grateful for the support of the Center for Advanced Computing and Imaging in Biomedicine through the Featured Areas Research Center Program within the framework of the Higher Education Sprout Project by the Ministry of Education, Taiwan, the support of Taiwan Semiconductor

Research Institute through the Joint Developed Project and the support from the Physics Division, National Center for Theoretical Sciences, Taiwan. E.-J. Kuo acknowledges financial support from the NSTC of Taiwan under Grant No. NSTC 114-2112-M-A49-036-MY3.

REFERENCES

- [1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [2] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.
- [3] K. Cho, B. Van Merriënboer, Ç. Gulçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1724–1734.
- [4] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Gated feedback recurrent neural networks," in *International conference on machine learning*. PMLR, 2015, pp. 2067–2075.
- [5] J. Bausch, "Recurrent quantum neural networks," *Advances in neural information processing systems*, vol. 33, pp. 1368–1379, 2020.
- [6] 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.
- [7] Y. Takaki, K. Mitarai, M. Negoro, K. Fujii, and M. Kitagawa, "Learning temporal data with a variational quantum recurrent neural network," *Physical Review A*, vol. 103, no. 5, p. 052414, 2021.
- [8] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*. Pmlr, 2013, pp. 1310–1318.
- [9] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.
- [10] J. Schmidhuber, "Learning to control fast-weight memories: An alternative to dynamic recurrent networks," *Neural Computation*, vol. 4, no. 1, pp. 131–139, 1992.
- [11] J. Ba, G. E. Hinton, V. Mnih, J. Z. Leibo, and C. Ionescu, "Using fast weights to attend to the recent past," *Advances in neural information processing systems*, vol. 29, 2016.
- [12] I. Schlag, K. Irie, and J. Schmidhuber, "Linear transformers are secretly fast weight programmers," in *International conference on machine learning*. PMLR, 2021, pp. 9355–9366.
- [13] 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.
- [14] K.-C. Peng, S. Y.-C. Chen, J.-C. Jiang, C.-Y. Liu, E.-J. Kuo, Y.-Y. Wang, P. Tiwari, A. Ceschini, C.-S. Chen, Y.-C. Hsu, C.-H. Lin, T.-Y. Li, A. Rosato, M. Panella, S. See, S. Al-Kuwari, K.-C. Chen, N.-Y. Chen, and H.-S. Goan, "Gated QKAN-FWP: Scalable quantum-inspired sequence learning," 2026. [Online]. Available: <https://arxiv.org/abs/2605.06734>
- [15] J.-C. Jiang, M. Y.-C. Huang, T. Chen, and H.-S. Goan, "Quantum variational activation functions empower Kolmogorov-Arnold networks," *arXiv preprint arXiv:2509.14026*, 2025.
- [16] Z. Liu, Y. Wang, S. Vaidya, F. Ruehle, J. Halverson, M. Soljagic, T. Hou, and M. Tegmark, "KAN: Kolmogorov–Arnold networks," in *International conference on learning representations*, vol. 2025, 2025, pp. 70367–70413.
- [17] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.
- [18] G. E. Blelloch, "Prefix sums and their applications," 1990.
- [19] S. Y.-C. Chen, Y. Peng, K.-C. Peng, J.-C. Jiang, C.-H. Lin, J. J. Park, H.-H. Tseng, H.-Y. Lin, K.-C. Chen, C.-Y. Liu, and S. Yoo, "Self-modulating quantum fast-weight programmers for efficient adaptive sequential learning," 2026. [Online]. Available: <https://arxiv.org/abs/2606.24933>
- [20] K.-C. Peng, J.-C. Jiang, C.-H. Lin, Y. Peng, J. J. Park, H.-H. Tseng, H.-Y. Lin, K.-C. Chen, C.-Y. Liu, S. Yoo, and S. Y.-C. Chen, "Stable self-modulating quantum fast-weight programmers with bounded memory gates," 2026. [Online]. Available: <https://arxiv.org/abs/2607.02363>
- [21] J.-S. Kim *et al.*, "Cuda quantum: The platform for integrated quantum-classical computing," in *2023 60th ACM/IEEE Design Automation Conference (DAC)*. IEEE, 2023, pp. 1–4.
- [22] K. Fujii and K. Nakajima, "Harnessing disordered-ensemble quantum dynamics for machine learning," *Physical Review Applied*, vol. 8, no. 2, p. 024030, 2017.
- [23] Y. Li, Z. Wang, R. Han, S. Shi, J. Li, R. Shang, H. Zheng, G. Zhong, and Y. Gu, "Quantum recurrent neural networks for sequential learning," *Neural Networks*, vol. 166, pp. 148–161, 2023.
- [24] Y. Li, Z. Wang, R. Xing, C. Shao, S. Shi, J. Li, G. Zhong, and Y. Gu, "Quantum gated recurrent neural networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 47, no. 4, pp. 2493–2504, 2024.
- [25] A. Kundu, A. Sarkar, P. Tiwari, and S. Feld, "Reinforcement learning for quantum circuit optimization: A review," *Openreview*, 2026.
- [26] K. Xu, L. Chen, and S. Wang, "Kolmogorov-Arnold networks for time series: Bridging predictive power and interpretability," *arXiv preprint arXiv:2406.02496*, 2024.
- [27] X. Han, X. Zhang, Y. Wu, Z. Zhang, and Z. Wu, "Are KANs effective for multivariate time series forecasting?" *arXiv preprint arXiv:2408.11306*, 2024.
- [28] S. Huang, Z. Zhao, C. Li, and L. Bai, "TimeKAN: KAN-based frequency decomposition learning architecture for long-term time series forecasting," *arXiv preprint arXiv:2502.06910*, 2025.
- [29] A. Noorizadegan, S. Wang, L. Ling, and J. P. Dominguez-Morales, "A practitioner's guide to Kolmogorov–Arnold networks," *Computer Science Review*, vol. 62, p. 100991, 2026. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1574013726000997>
- [30] Y.-C. Hsu, J.-C. Jiang, C.-H. Lin, K.-C. Peng, N.-Y. Chen, S. Y.-C. Chen, E.-J. Kuo, and H.-S. Goan, "QKAN-LSTM: Quantum-inspired Kolmogorov–Arnold long short-term memory," in *2026 International Conference on Quantum Communications, Networking, and Computing (QCNC)*. IEEE, 2026, pp. 650–659.
- [31] K.-C. Peng, J.-C. Jiang, C.-H. Lin, T.-Y. Li, N.-Y. Chen, and S. Y.-C. Chen, "Parameter-efficient quantum-inspired fast weight programmers for traffic-matrix forecasting," 2026. [Online]. Available: <https://arxiv.org/abs/2606.27821>
- [32] Y.-C. Lin, Y.-C. Hsu, I.-S. Tsai, C.-H. Lin, K.-C. Peng, J.-C. Jiang, Y.-Y. Wang, T.-C. Huang, T.-Y. Li, K.-C. Chen, S. Y.-C. Chen, and N.-Y. Chen, "Generative quantum-inspired Kolmogorov-Arnold eigensolver," 2026. [Online]. Available: <https://arxiv.org/abs/2605.04604>
- [33] I. Schlag and J. Schmidhuber, "Gated fast weights for on-the-fly neural program generation," in *NIPS Metalearning Workshop*, 2017.
- [34] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, 2018, pp. 4470–4481.
- [35] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, "Recurrent neural networks for multivariate time series with missing values," *Scientific reports*, vol. 8, no. 1, p. 6085, 2018.
- [36] P. Zhang, J. Xue, C. Lan, W. Zeng, Z. Gao, and N. Zheng, "Eleatt-rnn: Adding attentiveness to neurons in recurrent neural networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 1061–1073, 2019.
- [37] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," 2019.
- [38] C. Cecka, "CuTe layout representation and algebra," 2026. [Online]. Available: <https://arxiv.org/abs/2603.02298>
- [39] J.-C. Jiang, "QKAN: Quantum-inspired Kolmogorov-Arnold network," 2025. [Online]. Available: <https://github.com/Jim137/qkan>
- [40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [41] T. Viering and M. Loog, "The shape of learning curves: a review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7799–7819, 2022.
- [42] D. Mazzoni and K. Wagstaff, "Active learning in the presence of unlabelable examples," in *European Conference on Machine Learning*, 2004.