Kernel-based optimization of measurement operators for quantum reservoir computers

Markus Gross[1,*] and Hans-Martin Rieser[1]

[1] *Institute for AI Safety and Security, German Aerospace Center (DLR), Sankt Augustin and Ulm, Germany*

(Dated: April 27, 2026)

Finding optimal measurement operators is crucial for the performance of quantum reservoir computers (QRCs), since they employ a fixed quantum feature map. We formulate the training of both stateless (quantum extreme learning machines, QELMs) and stateful (memory dependent) QRCs in the framework of kernel ridge regression. We thus extend the kernel viewpoint of supervised quantum models to recurrent QRCs by deriving an exact Hilbert–Schmidt kernel representation of the optimal readout observable on history space. This approach renders an optimal measurement operator that minimizes prediction error for a given reservoir and training dataset. For large qubit numbers, this method is more efficient than the conventional training of QRCs. We discuss efficiency and practical implementation strategies, including Pauli basis decomposition and operator diagonalization, to adapt the optimal observable to hardware constraints. To demonstrate the effectiveness of this approach, we present numerical experiments on image classification and time series prediction tasks, including chaotic and strongly non-Markovian systems. The developed method can also be applied to other quantum machine learning models.

I. INTRODUCTION

Since quantum systems are inherently disturbed by measurements, finding efficient and optimal ways to extract information is a crucial part of any quantum machine learning (QML) model, and numerous studies have addressed this problem [1–14]. For variational QML models, measurement operators are often chosen based on hardware constraints or heuristics. Due to the training of the internal layers, this usually does not pose a severe problem for the accuracy of the model. However, in the case of QRCs, training happens only at the readout and thus depends crucially on the information extracted by the measurement operator. Finding optimal measurement operators is thus a key challenge in QRC design [15–19].

In [20], the notion of an optimal measurement operator has been introduced based on the equivalence between supervised quantum ML models and quantum kernel methods [20–23]. Such optimal measurement operators have subsequently been identified in a number of quantum ML studies [24–26], including a recent discussion of double descent in QELM-like models, which employ a fixed quantum feature map [27]. The measurement problem for QELMs from a more general quantum-information perspective has been addressed in [17]. On the side of classical reservoir computing, various methods have been proposed to construct equivalent kernel representations [28–31]. Besides this, no systematic study of optimal measurement operators for QRCs has been performed.

In the present work, we discuss the problem of determining an optimal measurement operator for QRCs using kernel-based methods. QRCs have been introduced as a hardware efficient and noise-resilient quantum ML approach [32–35], and have been successfully applied in tasks such as

* markus.gross@dlr.de

Symbol	Meaning
$\mathcal{H}$	Hilbert space
$\mathcal{M}(\mathcal{H})$	Space of Hermitian operators on $\mathcal{H}$
N	Total number of qubits
N_I, N_A	Number of input/ancilla qubits
D	Hilbert space dimension ($D = 2^N$)
d	Input data dimension ($\mathbf{x} \in \mathbb{R}^d$)*
d'	Time series dimension ($\mathbf{z}_t \in \mathbb{R}^{d'}$)*
P	Number of training samples
L	QRC memory length
T	Time series length
m	Number of measurement operators
$\mathbf{x}$	Input vector *
$\mathbf{z}_t$	Time series value at time t *
$\mathbf{y}$	Target vector
$\rho(\mathbf{x})$	Density matrix of the quantum state (feature map)
$\rho(x_t)$	D.m. after evolving over the input history [Eq. (2.3)]
η_I, η_M	D.m. of the input and memory states
$f(\mathbf{x})$	Classical readout function
M_k	Measurement operators
w_k	Readout weights
$K(\mathbf{x}, \mathbf{x}')$	Quantum kernel
α^*	Optimal kernel coefficients
M^*	Optimal measurement operator
P_k	N-qubit Pauli operators
σ^j	Pauli matrices ($j \in \{0, x, y, z\}$)

Table I. Summary of notation used in this work. *For QELMs, $\mathbf{x} = \mathbf{z}_t$ and $d = d'$, while for stateful QRCs, the identification in Eq. (2.3) is necessary.

image classification [36, 37] and time series prediction [16, 32, 38]. We demonstrate that using an optimal measurement operator enables systematic improvements of QRC performance in these tasks. Besides considering QELMs, which employ a memoryless quantum feature map and are thus close to untrained conventional quantum ML models [39], we show that our method can also be applied to recurrent QRCs with internal memory. The latter systems are typically used for time series processing [32, 33] and we show that the kernel framework underlying supervised quantum models naturally extends to the recurrent setting by constructing an exact Hilbert–Schmidt kernel on history space.

II. THEORY

A. Model

We consider an N-qubit model, with dimension $D = 2^N$ of its Hilbert space $\mathcal{H}$, described by the feature map

$$\mathbf{x} \mapsto \rho(\mathbf{x}), \tag{2.1}$$

which encodes input data $\mathbf{x} \in \mathbb{R}^d$ into a quantum state represented by a density matrix $\rho(\mathbf{x})$. The density matrix is an element of the $(D^2 = 4^N)$-dimensional space $\mathcal{M}(\mathcal{H})$ of Hermitian operators acting on $\mathcal{H}$ and can include the action of various quantum channels (such as unitary time evolution or noise) on some initial state ρ_0 (see Section III for specific examples). The classical output of a QRC is given by

$$f(\mathbf{x}) = \sum_{k=1}^{m} w_k \mathrm{tr}[M_k \rho(\mathbf{x})], \tag{2.2}$$

where $\{M_k\}$ are a set of chosen measurement operators (total number m) and w_k are trainable weights. We discuss the optimization problem in detail in the next section.

Importantly, for the present purpose, the feature map (2.1) is assumed to be *stateless* in the sense that the state ρ is always fully determined by the input vector $\mathbf{x}$. Such a model is naturally realized by a *quantum extreme learning machine* (QELM) [33, 34, 36]. This seems at first to preclude QRCs with internal *memory*, as commonly used for time series processing [32–34], because their current state $\rho(t)$ would not only depend on the present input, but also on the inputs preceding it. To address this issue, let $\{\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_T\}$ be a time series ($T+1$ samples in $\mathbb{R}^{d'}$) and assume that the QRC has a memory of length L. This approximation is typically warranted due to the echo state property of a QRC. Then, an effective input x_t at time $t \geq L$ can be defined as

$$x_t := (\mathbf{z}_{t-L}, \ldots, \mathbf{z}_t). \tag{2.3}$$

This quantity can be understood as an ordered set of $L+1$ samples $\mathbf{z}_{t-L}, \ldots, \mathbf{z}_t$ such that processing these samples through the QRC is sufficient to obtain the unique state $\rho(t)$ associated with x_t. Accordingly, the QRC output is fully determined by any of the $P = T - L$ unordered effective inputs $\{x_t\}_{t=L}^{T-1}$ and can thus be considered stateless in this setting [40]. Prescriptions like (2.3) to define truncated input histories are also used in kernelization of classical reservoir computing models [30, 31].

B. Measurement operator optimization

When training a RC-type model, there are in principle three formalisms available: (A) constrained primal optimization, (B) unconstrained primal optimization, and (C) dual space optimization [41]. Method (C) seems not to have been discussed previously in the context of QRCs.

Constrained primal optimization. The conventional way of training a QELM like in Eq. (2.2) consists in selecting a set of measurement operators $\{M_k\}$ and optimizing the weights w_k via regression on a training data set $\{(\mathbf{x}_i, y_i)\}_{i=1}^{P}$ of P samples:

$$\min_{\mathbf{w} \in \mathbb{R}^m} \sum_{i=1}^{P} \mathcal{L}\left(\sum_{k=1}^{m} w_k \mathrm{tr}[M_k \rho(\mathbf{x}_i)],\, y_i \right) + \frac{\lambda}{2} \|\mathbf{w}\|_2^2, \tag{2.4}$$

	(A) Fixed measurements (constrained primal)	**(B) Free observable (unconstrained primal)**	**(C) Kernel form (dual of B)**
Given data	$\{(\mathbf{x}_i, y_i)\}_{i=1}^P$, reservoir states $\rho_i := \rho(\mathbf{x}_i) \in \mathcal{M}(\mathcal{H})$		
Trainable object	$\mathbf{w} \in \mathbb{R}^m$	$M \in \mathcal{M}(\mathcal{H})$	$\boldsymbol{\alpha} \in \mathbb{R}^P$
Allowed observables	$M = \sum_{k=1}^m w_k M_k$ $\in S := \mathrm{span}\{M_k\}$	any $M \in \mathcal{M}(\mathcal{H})$	$M = \sum_{i=1}^P \alpha_i \rho_i$ (representer form)
Prediction	$f(\mathbf{x}) = \sum_{k=1}^m w_k \mathrm{tr}(M_k \rho(\mathbf{x}))$	$f(\mathbf{x}) = \mathrm{tr}(M\rho(\mathbf{x}))$	$f(\mathbf{x}) = \sum_{i=1}^P \alpha_i K(\mathbf{x}_i, \mathbf{x})$
Training objective	$\min_{\mathbf{w}} \frac{1}{2}\sum_{i=1}^P \Big(y_i - \sum_{k=1}^m w_k \mathrm{tr}(M_k\rho_i)\Big)^2 + \frac{\lambda}{2}\|\mathbf{w}\|_2^2$	$\min_M \frac{1}{2}\sum_{i=1}^P \big(y_i - \mathrm{tr}(M\rho_i)\big)^2 + \frac{\lambda}{2}\mathrm{tr}(M^2)$	$\min_{\boldsymbol{\alpha}} \frac{1}{2}\|\mathbf{y} - \mathbf{K}\boldsymbol{\alpha}\|_2^2 + \frac{\lambda}{2}\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}$
Matrices	$\Phi_{ik} := \mathrm{tr}(M_k\rho_i)$	$K_{ij} := \mathrm{tr}(\rho_i\rho_j)$	$K_{ij} := \mathrm{tr}(\rho_i\rho_j)$
Solution	$\mathbf{w}^\star = (\Phi^\top\Phi + \lambda\mathbb{1})^{-1}\Phi^\top\mathbf{y}$	$M^\star = \sum_{i=1}^P \alpha_i^\star \rho_i$	$\boldsymbol{\alpha}^\star = (\mathbf{K} + \lambda\mathbb{1})^{-1}\mathbf{y}$
Corresponding kernel	$K_{\mathrm{meas}}(\mathbf{x}, \mathbf{x}') = \boldsymbol{\varphi}(\mathbf{x})^\top\boldsymbol{\varphi}(\mathbf{x}')$, $\varphi_k(\mathbf{x}) = \mathrm{tr}(M_k\rho(\mathbf{x}))$	$K(\mathbf{x}, \mathbf{x}') = \mathrm{tr}(\rho(\mathbf{x})\rho(\mathbf{x}'))$	$K(\mathbf{x}, \mathbf{x}') = \mathrm{tr}(\rho(\mathbf{x})\rho(\mathbf{x}'))$
Equivalence statement	not equivalent to (B)/(C) unless S is complete on the data manifold (no information loss)	exactly equivalent to (C) (same objective, different parametrization)	exactly equivalent to (B) (same objective, different parametrization)
Meaning of "optimal"	minimizer over $M \in S$ (best within chosen measurement family)	minimizer over all $M \in \mathcal{M}(\mathcal{H})$ (best linear readout)	minimizer in RKHS induced by K (same solution as B)

Table II. Three training formulations for a QRC. The training objective and solution are specialized here to the case of least-squares loss and assuming an orthonormal operator basis. For other kinds of losses, usually no analytic expressions for the weights $\mathbf{w}^*$ exist. If the QRC carries an internal memory state, the above formulations remain intact but the data $\mathbf{x}_i \hat{=} x_t$ represents an input history [see Eq. (2.3)] and $\rho(\mathbf{x}) \hat{=} \rho(x_t)$ is the state resulting by sequentially processing that history (see Sections III A and F).

where $\mathcal{L}$ is a loss function and λ is a regularization parameter. For regression and time-series prediction tasks, one typically uses the least-squares loss $\mathcal{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$, in which case the optimal weights are given by

$$\mathbf{w}^* = (\boldsymbol{\Phi}^\top\boldsymbol{\Phi} + \lambda\mathbb{1})^{-1}\boldsymbol{\Phi}^\top\mathbf{y}, \qquad \Phi_{ik} = \mathrm{tr}(M_k\rho(\mathbf{x}_i)), \tag{2.5}$$

where $\mathbf{y} \in \mathbb{R}^P$ is the sample vector of target values and $\varphi_k(\mathbf{x}) = \mathrm{tr}(M_k\rho(\mathbf{x}))$ can be considered as the feature map. For classification tasks, one often uses the hinge loss $\mathcal{L}_{\mathrm{hinge}}(\hat{y}, y) = \max(0, 1 - y\hat{y})$ (with $y \in \{-1, +1\}$), in which case the resulting optimal weights then generally do not have a closed-form expression. For multi-class classification with C classes, a common strategy is to train C separate readouts (one-vs-rest), i.e., solve Eq. (2.4) for each class $c = 1, \ldots, C$ with targets $y_i^{(c)} \in \{-1, +1\}$ and separate weights $\mathbf{w}^{(c)}$.

Unconstrained primal optimization. Crucially, unless the $\{M_k\}$ span the full operator space $\mathcal{M}(\mathcal{H})$, the optimization in Eq. (2.4) does not guarantee that all relevant information encoded in the quantum state ρ can actually be accessed. The resulting trained QRC may thus perform poorly on the given data set. In variational QML, this issue is addressed by training parametrized unitaries acting on ρ before performing the measurement. Alternatively, and specifically for QRCs, we can exploit the freedom in the choice of measurement operators and hence optimize over the *full*

Hermitian operator space $\mathcal{M}(\mathcal{H})$:

$$\min_{M\in\mathcal{M}(\mathcal{H})} \sum_{i=1}^{P} \mathcal{L}\big(\mathrm{tr}[M\rho(\mathbf{x}_i)],\, y_i\big) + \frac{\lambda}{2}\mathrm{tr}(M^2) \tag{2.6}$$

with the same loss function $\mathcal{L}$ as in Eq. (2.4). In practice, this optimization over $\mathcal{M}(\mathcal{H})$ can be achieved by using in (2.4) as observables $\{M_k\}$ a complete orthonormal $m = D^2$-dimensional basis of $\mathcal{M}(\mathcal{H})$ (e.g., the normalized Pauli basis $\{2^{-N/2}P_k\}$). The associated optimal weights w_k^* are given formally by the same expression as in Eq. (2.5), and yield the optimal observable

$$M^* = \sum_{k=1}^{D^2} w_k^* M_k. \tag{2.7}$$

For the least-squares loss, the solution weights w_k^* are given by Eq. (2.5).

Dual optimization. The above unconstrained optimization procedure becomes especially natural in its dual formulation, resting on the connection between supervised quantum ML models and quantum kernels [20]. Identifying the unconstrained feature map as $\phi(\mathbf{x}) = \rho(\mathbf{x})$ allows one to define the kernel $K(\mathbf{x}, \mathbf{x}') = \mathrm{tr}(\rho(\mathbf{x})\rho(\mathbf{x}'))$, which induces a reproducing kernel Hilbert space (RKHS) [42]. The representer theorem implies that the optimizer M^* (*optimal measurement operator*) of the problem (2.6) is of the form (see also Section A)

$$M^* = \sum_{j=1}^{P} \alpha_j^* \rho(\mathbf{x}_j), \tag{2.8}$$

i.e., it necessarily sits in the span of the training features [20, 42]. In terms of the representer coefficients $\boldsymbol{\alpha} \in \mathbb{R}^P$, the corresponding dual problem reads

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^P} \sum_{i=1}^{P} \mathcal{L}\big((\mathbf{K}\boldsymbol{\alpha})_i,\, y_i\big) + \frac{\lambda}{2}\,\boldsymbol{\alpha}^\top \mathbf{K}\boldsymbol{\alpha}. \tag{2.9}$$

For the least-squares loss, the canonical solution coefficients are available in closed form as

$$\boldsymbol{\alpha}^* = (\mathbf{K} + \lambda\mathbb{1})^{-1}\mathbf{y}, \qquad K_{ij} = \mathrm{tr}(\rho(\mathbf{x}_i)\rho(\mathbf{x}_j)). \tag{2.10}$$

For other loss functions, the solution coefficients must typically be computed numerically. Independently of the loss function $\mathcal{L}$, an optimally trained QELM [Eq. (2.2)] admits the kernel form

$$f(\mathbf{x}) = \sum_{j=1}^{P} \alpha_j^* \mathrm{tr}(\rho(\mathbf{x}_j)\rho(\mathbf{x})) = \mathrm{tr}(M^*\rho(\mathbf{x})) \tag{2.11}$$

The generalization to multi-dimensional target values $\mathbf{y}_i \in \mathbb{R}^C$ $(i = 1, \ldots, P)$ is straightforward and requires a separate kernel construction like in Eq. (2.11) for each class (this includes multi-class classification in a one-vs-rest scheme).

When the measurement operators $\{M_k\}$ form a complete orthonormal basis (or at least span the relevant subspace containing M^* [17]), primal and dual formulations are *equivalent* and Eqs. (2.7)

and (2.8) describe the same object. Expanding the optimal observable M^* in a complete orthogonal basis leads to a relation between the coefficients $\mathbf{w}^*$ and $\boldsymbol{\alpha}^*$ via [cf. Eqs. (A11) and (E5)]

$$w_k^* = \frac{1}{g_k} \sum_{j=1}^{P} \alpha_j^* \mathrm{tr}(M_k \rho(\mathbf{x}_j)). \tag{2.12}$$

Here, $\mathrm{tr}(M_k M_l) = g_k \delta_{kl}$ with $g_k = 2^N$ for the orthogonal Pauli basis. The optimization formulations are summarized in Table II.

Remarks. (i) Neither the primal nor the dual optimization methods assume uncorrelated samples. Both are therefore compatible with the sliding window construction (2.3). (ii) Global conjugation $\rho(\mathbf{x}) \mapsto U\,\rho(\mathbf{x})\,U^\dagger$ leaves the expression (2.11) unchanged and only transforms the optimal observable as $M^* \mapsto U M^* U^\dagger$. (iii) The constrained primal formulation can be "dualized", too, by using the kernel induced by the constrained feature map (see Table II). While this does not render a lower training error, it may offer computational benefits.

C. Enhancements and practical considerations

Using the Hermitian $2^N \times 2^N$ matrix M^* directly as a measurement operator can be infeasible in a practical setting with large qubit numbers N. After addressing below the computational efficiencies of the primal vs. dual method, we consider possible approaches to reducing the implementation complexity.

1. *Efficiency of primal vs. dual optimization*

When taking the complete $m = D^2 = 4^N$-dimensional basis of measurement operators $\{M_k\}$ in Eq. (2.2), the primal method is equivalent to the dual one, and the preference for either method becomes a matter of computational efficiency:

- The *primal* formulation requires solving for m weights, involving the inversion of a $m \times m$ matrix (covariance matrix) with time complexity $\mathcal{O}(m^3)$ and memory complexity $\mathcal{O}(m^2)$. In the case of a complete operator basis, $m = D^2 = 4^N$, leading to a scaling of $\mathcal{O}(64^N)$ in time and $\mathcal{O}(16^N)$ in memory.
- In contrast, the *dual* formulation solves for P coefficients by inverting the $P \times P$ kernel matrix, which scales as $\mathcal{O}(P^3)$ in time and $\mathcal{O}(P^2)$ in memory.

Thus, the dual method is preferable when the number of samples is small compared to the feature space dimension ($P \ll m$), which is typical for large qubit numbers N due to the exponential scaling of D. If one chooses to truncate the operator basis to $m < D^2$, the primal method can become more efficient, however at the expense of losing the optimality guarantee.

Note that the resulting optimal measurement operator M^* [Eq. (2.8)] is still a $D \times D$ matrix. Explicitly constructing and storing M^* requires $\mathcal{O}(D^2) = \mathcal{O}(4^N)$ memory, which eventually limits the feasibility of the full optimization for large N. Conversely, if the number of measurement operators is small (e.g., due to truncation) or the dataset is very large such that $P \gg m$, the primal formulation becomes more efficient. Specialized classical and quantum algorithms for the inversion of large matrices exist that can mitigate the exponential scaling (see, e.g., [43]).

We finally remark that, in practice, one does not have to explicitly construct the samples as in Eq. (2.3) for the optimization in the stateful case. Instead, one can simply run the QRC over the time series $\mathbf{z}_t$ and (after some equilibration period) collect the output states $\{\rho_t\}_{t=L}^{T-1} \hat{=} \{\rho(\mathbf{x}_j)\}_{j=1}^{P}$ or a subset of them. The sliding window construction is implicit in this procedure.

2. Pauli basis projection

In this approach, the Hermitian operator M^* is expanded in the Pauli (or any other operator) basis,

$$M^* = \sum_{k=0}^{4^N-1} c_k P_k, \qquad c_k = \frac{1}{2^N}\mathrm{tr}(M^* P_k), \tag{2.13}$$

where the P_k are all possible tensor products of N Pauli operators (including the identity operator) and the coefficients c_k follow from orthogonality of the Pauli basis. The absolute values $|c_k|$ indicate the importance of the observable P_k for the given prediction task. Selecting only the most relevant observables or restricting to a subset of a given maximum Pauli weight renders a set of implementable observables $\mathcal{O}^*_{\mathrm{Pauli}} = \{P_k\}_{k=1}^{K}$. The resulting prediction error is discussed in Section D.

A problem with this approach is that the elements of the set $\mathcal{O}^*_{\mathrm{Pauli}}$ do not necessarily mutually commute, requiring in principle multiple executions of the quantum circuit to obtain the full prediction. Moreover, the Pauli expansion is highly redundant, as we will see in the following.

3. Diagonalization

Since M^* is Hermitian, it can be diagonalized via a unitary V as

$$M^* = V \Lambda V^\dagger, \qquad \Lambda = \sum_{\mathbf{b}\in\{0,1\}^N} \lambda_{\mathbf{b}} \, |\mathbf{b}\rangle\langle\mathbf{b}|, \tag{2.14}$$

where $\{|\mathbf{b}\rangle\}_{\mathbf{b}\in\{0,1\}^N}$ is the computational basis and $\lambda_{\mathbf{b}} \in \mathbb{R}$ are eigenvalues. For any state $\rho(\mathbf{x})$,

$$\mathrm{tr}(M^*\rho(\mathbf{x})) = \mathrm{tr}\big(\Lambda\, V^\dagger \rho(\mathbf{x}) V\big) = \sum_{\mathbf{b}\in\{0,1\}^N} \lambda_{\mathbf{b}} \, \langle\mathbf{b}|V^\dagger\rho(\mathbf{x})V|\mathbf{b}\rangle. \tag{2.15}$$

The prediction is the weighted average of the observed bitstrings using the eigenvalues $\{\lambda_{\mathbf{b}}\}$. Thus, the optimal observable can be implemented by a single basis rotation $V^\dagger$ followed by a standard computational-basis measurement. This is equivalent to measuring all 2^N distinct Pauli-Z strings $\{Z_{\mathbf{s}}\}_{\mathbf{s}\in\{0,1\}^N}$, since (see Section C)

$$\Lambda = \sum_{\mathbf{s}\in\{0,1\}^N} \beta_{\mathbf{s}} \, Z_{\mathbf{s}} \tag{2.16}$$

with $\beta_{\mathbf{s}}$ given by Eq. (C2).

The unitary V may be treated as part of the readout (applied once just before measurement) or absorbed into the reservoir by redefining the last layer as $U_R \mapsto V^\dagger U_R$. This global conjugation leaves the kernel $K(\mathbf{x}, \mathbf{x}')$ and the trained predictor unchanged (see Section II B) and only

affects the hardware realization of the optimal measurement. While, strictly, a pre-rotation followed by computational-basis measurement is sufficient and equivalent to measuring the exact M^*, we will include in our numerical experiments [Section IV] also the Pauli observable decomposition [Eq. (2.13)], as both approaches can lead to slight differences in numerical stability.

III. QRC MODELS

The optimization methods described above are general and independent of the encoding scheme or other quantum channels acting on the readout state ρ. In order to test our methods in practice, we consider some specific QRC models.

A. QRCs and QELMs

The standard (stateful) QRC model introduced in [32, 33] consists of a set of coherently evolving (memory/ancilla) qubits (N_A) periodically coupled to a set of input qubits (N_I). At time t, a classical input $\mathbf{z}_t \in \mathbb{R}^{d'}$ is encoded into an input state with density matrix $\eta_I(\mathbf{z}_t)$. The joint state with the coherent ancilla state $\eta_M(t)$ is formed and time-evolved by a unitary U:

$$\eta(t) = U\left(\eta_I(\mathbf{z}_t) \otimes \eta_M(t)\right) U^\dagger. \tag{3.1}$$

This $\eta(t)$ represents the feature map $\rho(x_t)$ on the input sequence defined by Eq. (2.3) [corresponding to the one in (2.1)]. Subsequently, the input is reset and discarded, and the memory alone is carried to the next step:

$$\eta_M(t+1) = \mathrm{tr}_I[\eta(t)]\,, \tag{3.2}$$

where tr_I denotes partial trace over the input register I (see Section F for further details).

We assume the encoding state to be pure, $\eta_I(\mathbf{z}) = |\psi(\mathbf{z})\rangle\langle\psi(\mathbf{z})|$, with the state vector $|\psi(\mathbf{z})\rangle \in \mathcal{H}_I$ (living in the N_I-qubit input Hilbert space) given by

$$|\psi(\mathbf{z})\rangle = S(\mathbf{z})|\Phi_0\rangle \tag{3.3}$$

in terms of the encoding unitary $S(\mathbf{z})$ and the initial state $|\Phi_0\rangle = |0\rangle^{\otimes N_I}$. The *reservoir* unitary U is defined by gate operations or Hamiltonian time evolution, e.g., $U(t) = \exp(-\mathrm{i}tH_R)$ with a reservoir Hamiltonian H_R. Specifically, we consider a transverse field Ising Hamiltonian with coupling parameters h and J:

$$H_R = -h\sum_{j=1}^{N}\sigma_j^x - J\sum_{j=1}^{N-1}\sigma_j^z\sigma_{j+1}^z, \tag{3.4}$$

as well as the variant with σ^x and σ^z exchanged. Here, σ_j^p, $p \in \{0, x, y, z\}$ ($\sigma_j^0 \equiv \mathbb{1}$) denote single-qubit Pauli matrices acting on the j-th qubit.

For a *QELM*, the coherent qubits are absent ($N_A = 0$) and Eq. (3.1) reduces to a stateless feature map

$$\rho(\mathbf{x}) = \eta(\mathbf{x}) = U\eta_I(\mathbf{x})U^\dagger \tag{3.5}$$

without time evolution. The sliding window prescription (2.3) is not relevant in this case and one can take $\mathbf{x} = \mathbf{z}_t$ to process time series. Moreover, in this setting, any data-independent unitary drops out from the trained predictor in Eq. (2.11) as

$$K(\mathbf{x}, \mathbf{x}') = \mathrm{tr}\big(\rho(\mathbf{x})\rho(\mathbf{x}')\big) = \mathrm{tr}(\eta_I(\mathbf{x})\eta_I(\mathbf{x}')) = \big|\langle\Phi_0|S^\dagger(\mathbf{x})S(\mathbf{x}')|\Phi_0\rangle\big|^2. \tag{3.6}$$

This does not hold for the QRC dynamics (3.2) with memory.

B. Encoding schemes

The encoding $S(\mathbf{z})$ is a crucial component of any quantum ML model. In our numerical experiments, we use parallel product state encodings of the form [44]

$$S(\mathbf{z}) = \bigotimes_{j=1}^{N_I} S_j(z_j), \tag{3.7}$$

with the unitary operator S_j encoding a single scalar z_j (thus $N_I = d$) and acting on the j-th qubit. We consider the following specific encoding schemes.

(1) *Rotational* encoding:

$$S_j(z_j) = R_{p_j}(2\pi z_j), \tag{3.8}$$

where

$$R_p(\theta) \equiv \exp\left(-i\frac{\theta}{2}\sigma^p\right) = \cos\left(\frac{\theta}{2}\right)\mathbb{1} - i\sin\left(\frac{\theta}{2}\right)\sigma^p \tag{3.9}$$

are standard rotation gates. We will mostly use R_Y encoding ($p = y$) for all qubits, which results in the state

$$|\psi_{\mathrm{rot}}(z)\rangle = R_Y(2\pi z)|0\rangle = \cos(\pi z)|0\rangle + \sin(\pi z)|1\rangle. \tag{3.10}$$

This implies a useful data scale of $z \in [0, 1]$, since $z \in [1, 2]$ produces only an irrelevant global phase relative to $z \in [0, 1]$. An R_X gate acts similarly, but introduces a complex phase, which is undesired and not needed for classical data processing. R_Z gates alone are not suitable for encoding as they just shift the phase of the $|0\rangle$ state. They can, however, be meaningfully combined with R_Y gates to encode two scalars into the same qubit (dense encoding):

$$\widehat{S}_j(z_j, z_{j+1}) = R_Z(2\pi z_{j+1})R_Y(2\pi z_j), \tag{3.11}$$

where now $N_I = d/2$ [45, 46].

(2) *Amplitude* encoding: inputs are scaled to $z \in [0, 1]$ and prepared as single-qubit states that traverse half a great circle on the Bloch sphere:

$$|\psi_{\mathrm{amp}}(z)\rangle = \sqrt{z}|0\rangle + \sqrt{1-z}|1\rangle. \tag{3.12}$$

A symmetric variant maps $z \in [-1, 1]$ to the full great circle on the Bloch sphere:

$$|\psi_{\mathrm{amp'}}(z)\rangle = z|0\rangle + \sqrt{1-z^2}|1\rangle. \tag{3.13}$$

We remark that these encodings can be expressed in terms of single R_Y rotations, e.g., $|\psi_{\mathrm{amp'}}(z)\rangle = z|0\rangle + \sqrt{1-z^2}|1\rangle = R_Y(2\arccos z)|0\rangle$.

C. Measurement readout

In practice, the measured expectation values in Eq. (2.2) can be augmented by nonlinear features in order to enhance the expressivity of the model [38, 47–49]. Defining the vector of measurement outcomes

$$\mathbf{v}(\mathbf{x}) := (v_1(\mathbf{x}), \ldots, v_m(\mathbf{x}))^\top, \qquad v_k(\mathbf{x}) = \mathrm{tr}[M_k \rho(\mathbf{x})], \tag{3.14}$$

we construct a readout vector $\mathbf{R}(\mathbf{x})$ consisting of all distinct monomials in the components of $\mathbf{v}(\mathbf{x})$ of total degree up to $p_{\max}$. Introducing the multi-index $\boldsymbol{\alpha} \in \mathbb{N}_0^m$ with $|\boldsymbol{\alpha}| := \sum_{k=1}^m \alpha_k$, we can write

$$\mathbf{R}(\mathbf{x}) := \left(\prod_{k=1}^m v_k(\mathbf{x})^{\alpha_k} \right)_{1 \le |\boldsymbol{\alpha}| \le p_{\max}} \in \mathbb{R}^{m_{\mathrm{ro}}}, \tag{3.15}$$

where any fixed ordering of the multi-indices $\boldsymbol{\alpha}$ specifies the components R_j of $\mathbf{R}$. The number of distinct monomials of total degree p equals $\binom{m+p-1}{p}$, so that the total readout dimension is

$$m_{\mathrm{ro}} = \sum_{p=1}^{p_{\max}} \binom{m+p-1}{p} = \binom{m+p_{\max}}{p_{\max}} - 1. \tag{3.16}$$

Taking into account C possible output channels/classes with targets $\mathbf{y}_i = (y_i^{(1)}, \ldots, y_i^{(C)})^\top \in \mathbb{R}^C$, the complete QRC output function becomes

$$\mathbf{f}(\mathbf{x}) = \mathbf{W}^\top \mathbf{R}(\mathbf{x}), \qquad \mathbf{W} = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(C)}] \in \mathbb{R}^{m_{\mathrm{ro}} \times C}. \tag{3.17}$$

Note that the measurement-operator optimization theory developed here applies only to the linear readout case.

IV. NUMERICAL EXPERIMENTS

In the following, we present numerical experiments on QRCs and QELMs trained using the kernel-based measurement operator optimization. We will typically show test accuracy as a function of the number of measurement operators kept after thresholding. The maximum number of operators resulting from the diagonalization and Pauli projection methods [Sections II C 2 and II C 3] is $m_{\mathrm{diag}} = 2^N \times D_{\mathrm{out}}$ and $m_{\mathrm{pauli}} = 4^N \times D_{\mathrm{out}}$, respectively, where D_{out} is the output dimension. In the case of setups without reservoir unitaries, the effective maximum number can be lower, since expectation values involving Pauli-Y's vanish identically on states encoded via (3.10), (3.12), or (3.13) (see Section G). After having determined the optimal operator set, we generally re-fit the QRC using this set [alternatively, one may use Eq. (2.12)].

A. Image classification task

We first consider the standard QELM task of classifying handwritten digits from the MNIST dataset [50], which contains about $P \simeq 70\,000$ images of size 28×28 pixels [36, 51, 52]. We use a QELM with $N = 5$ qubits and dimensionally reduce the images to their $2N$ dominant principal

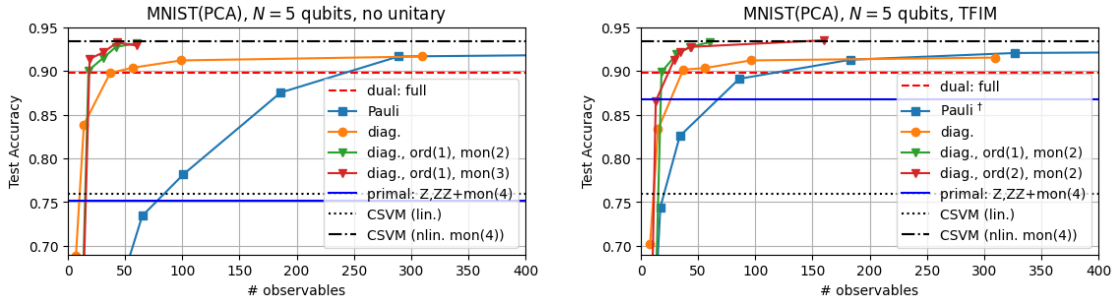

Figure 1. Optimization of the measurement operator for a QELM trained on the MNIST (first $2N$ principal components) classification task (left) without and (right) with a reservoir unitary (transverse field Ising model). The dashed line represents the test accuracy obtained using the full operator M^* (i.e., one per digit class), while the connected symbols correspond to the Pauli decomposition and diagonalization approaches [Eqs. (2.13) and (2.14)]. The notation ord(k) and mon(p) refers to the observable order k and monomial order p in Eq. (3.15). The curve labeled 'primal: Z,ZZ+mon(4)' represents primal optimization using only the Pauli-Z and -ZZ observables enhanced by readout monomials up to 4th order, while CSVM refers to a classical SVM trained on (non-)linear features constructed from the principal components. Since there is no intrinsic randomness in the model, error bars arise only from the selection of samples and are negligible here. †Here, we optimized the kernel only over a single class (to limit combinatorial explosion in the number of readout terms), but used the resulting operator set to re-train the QELM for all classes.

components, which we encode via R_Y-R_Z rotation operators (allowing for two features per qubit). For the kernel-based optimization, we use a random subset of approximately 10 000 samples. For comparison to a classical model, we consider an SVM with polynomial features (up to degree 4) constructed from the principal components.

The prediction accuracy on a held-out test data set ($P_{\text{test}} = 0.2P$) is given by the ratio of correctly classified over total number of samples:

$$\mathcal{A} = \frac{1}{P_{\text{test}}} \sum_{i=1}^{P_{\text{test}}} \delta_{y_i,\hat{y}_i}, \quad \text{with } \hat{y}_i = \arg\max_c \hat{f}^{(c)}(\mathbf{x}_i). \tag{4.1}$$

Figure 1 shows the resulting test accuracies for the MNIST classification problem in the case without (left panel) and with (right panel) a reservoir unitary.

QELM without a reservoir. We find that the test accuracy of the classical SVM saturates at around $\mathcal{A}_{\text{CSVM}} \approx 93.5\%$, which appears to be also an upper limit for the QELM. This limit is reached when the QELM readout is enhanced by monomials up to order $p_{\max} = 2$ [see Eq. (3.17)], both for the Pauli-decomposition and diagonalization approaches [Fig. 1(a)]. In this case, it suffices even to restrict the observables to weight-1 and -2 Pauli strings. Notably, without additional monomial features, the QELM test accuracy saturates at $\mathcal{A}_{\text{QELM}} \approx 92\%$, but requires a significantly larger number of observables. This quantum-to-classical performance gap exists because powers $\langle\sigma_j\rangle^q$ of expectation values of observables acting on the *same* qubit j cannot be represented in the operator basis for the present encoding.

QELM with a reservoir. Figure 1(b) shows the test accuracies for a QELM in the same setup as above, but *with* a reservoir unitary based on the TFIM (3.4) (with parameters $th \approx tJ \approx 10$). When optimizing via the primal route using only Z- and ZZ-Pauli observables enhanced with monomials up to order $p_{\max} = 4$, we obtain a higher test accuracy in the case with than without

a reservoir unitary. We explain this by the information scrambling property of the unitary, which makes information that sits in the Pauli-X- and Y-sectors accessible in the Z sector. In general, the results obtained for the QELM optimized via the kernel method are similar both with and without a reservoir unitary, demonstrating the robustness of the approach.

We summarize this section with some common findings: (i) Unless one uses a complete operator basis or techniques like temporal multiplexing [32], the primal optimization is generally inferior to the kernel-based approach: in the latter, it suffices to restrict the readout to Pauli-Z observables enhanced by 2nd-degree monomials to reach the classical SVM accuracy $\mathcal{A}_{\mathrm{CSVM}}$. The fundamental reason is of course that the kernel-based method implicitly renders an optimized pre-rotation before measurement. (ii) Performing the kernel-based optimization only over a single class can significantly reduce the computational load without sacrificing accuracy, provided the QELM is retrained for all classes. (iii) Low order monomial in the readout (especially squares) are generally helpful to close the quantum-to-classical accuracy gap (depending on the specific encoding).

B. Time series prediction

We turn to autoregressive prediction tasks [32, 34, 53], where, given a time series $\{\mathbf{z}_t\}_{t=0}^{T}$, the QRC is trained to predict the next step $\hat{\mathbf{z}}_{t+1} = f(x_t)$ (with the prescription (2.3) where appropriate). A complete forecasted trajectory is generated by repeatedly applying the prediction function, i.e., $\hat{\mathbf{z}}_{t+1} = f(\hat{x}_t)$. As was the case for static data, enhancing the readout with monomials typically improves forecasting capability [38, 49]. However, they obscure a clear assessment of the performance of the quantum parts of the model, and we therefore consider here only setups without hand-engineered features in the readout. As a performance measure, we consider, beside the root-mean-square training error $E_{\mathrm{train}} = \left[\frac{1}{P}\sum_{t=0}^{P-1}\|\mathbf{z}_{t+1} - f(\mathbf{z}_t)\|^2\right]^{1/2}$, the forecast horizon

$$T_{\mathrm{fch}} = \inf\{t : \|\mathbf{z}(t) - \hat{\mathbf{z}}(t)\| > \sigma\} \tag{4.2}$$

where $\sigma^2 = \frac{1}{P}\sum_{i=0}^{P-1}\|\bar{\mathbf{z}} - \mathbf{z}(t_i)\|^2$ is the variance of the trajectory and $\bar{\mathbf{z}} = \frac{1}{P}\sum_{i=0}^{P-1}\mathbf{z}(t_i)$ is its mean. In the case of chaotic systems, the forecast horizon is expressed in units of the Lyapunov time $T_L = 1/\lambda_1$ with largest Lyapunov exponent λ_1.

1. *Memoryless QRCs (QELMs)*

Time series generated by Markovian dynamical systems can be learned using QELMs, i.e., memoryless QRCs (cf. [54–56] for similar minimal QRC approaches and [57] for a discussion within a classical RC framework). We can thus train the QRC directly on the individual samples of the time series, $\mathbf{x}_t = \mathbf{z}_t$ and ignore the prescription (2.3). As a standard benchmark for reservoir computing prediction tasks, we take the ($d = 3$)-dimensional Lorenz-63 chaotic system, described by

$$\begin{aligned}\dot{z}_1 &= \sigma(z_2 - z_1)\\ \dot{z}_2 &= z_1(\rho - z_3) - z_2\\ \dot{z}_3 &= z_1 z_2 - \beta z_3\end{aligned} \tag{4.3}$$

with parameters $\sigma = 10$, $\rho = 28$, $\beta = 8/3$ and largest Lyapunov exponent $\lambda_1 \approx 0.89$ [58]. The trajectory is numerically generated using a RK4 (Runge-Kutta) scheme with tolerances of 10^{-10}.

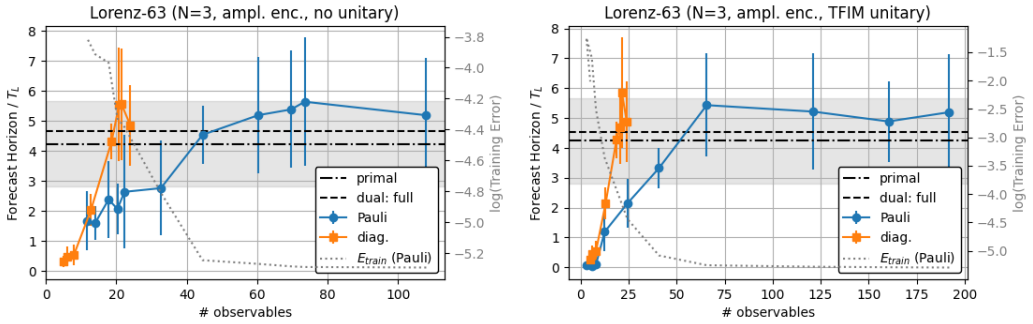

Figure 2. Optimization of the measurement layer for a memoryless QRC without ancilla qubits (QELM), for the cases *without* and *with* a (TFIM-based) reservoir unitary. The QRC is trained on a 3-dimensional chaotic time series [Eq. (4.3)]. The error bars on the forecast horizon represent the standard deviation (not error of the mean) after averaging over different initial conditions. (Their magnitude is comparable to classical RC results for chaotic systems.) The dashed line represents the forecast horizon when using the kernel-based optimal measurement operator M^* directly, while the connected points correspond to decompositions into subsets of observables [Eqs. (2.13) and (2.14)]. The dash-dotted line gives the forecast horizon (with its standard deviation indicated by the gray area) for the primal optimization over the complete operator basis. E_{train} denotes the (RMS) training error. The horizontal axis represents the number of observables ranked by their relevance (left = least relevant; maximum number = $64 \times 3 = 192$ for right panel).

We use the amplitude encoding (3.12) with $N_I = 3$ qubits. We find that the other tested encoding schemes yield similar training accuracy.

The resulting forecast horizons are illustrated in Fig. 2 for the case without (left) and with a reservoir unitary (right). Due to using an optimal measurement operator M^*, the presence of a reservoir unitary does not have a significant impact on the largest achievable forecast horizon. Interestingly, both for the Pauli-decomposition and the diagonalization approach, the largest forecast horizon is reached for a slightly reduced number of observables. We explain this by the numerical conditioning effects and redundancy in the operator basis for the present dataset and encoding scheme.

2. QRCs with internal memory

We now turn to a *stateful* QRC, obtained by extending the above QELM model by $N_A \geq 1$ ancilla qubits, coupled via a TFIM unitary. The measurement operator acts on all $N = N_I + N_A$ qubits and is optimized using the kernel-based approach. We first revisit the forecasting problem for the Lorenz-63 model and then consider two strongly non-Markovian time series: a harmonic signal and the Mackey-Glass system.

Lorenz-63 time series. Using memory states significantly enhances prediction accuracy both for the primal and dual optimization method (Fig. 3), yielding forecast horizons of $T_{\text{fch}}/T_L \approx 10$ (which is comparable to recent literature results [38]). Notably, using the optimal operator M^* directly leads in this case to a significantly lower forecast horizon and higher training error ($E_{\text{train}} \approx 2\times10^{-6}$) compared to the Pauli decomposition or diagonalization methods ($E_{\text{train}} \approx 4 \times 10^{-8}$). This result appears to be specific to the current QRC setup or data, as we do not observe strong discrepancies in the other benchmarks (cf. Figs. 2 and 4).

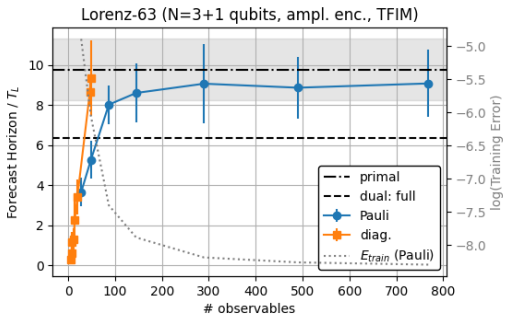

Figure 3. Optimization of the measurement operator for a QRC with internal memory ($N_A = 1$ ancilla qubit coupled via a TFIM unitary to $N_I = 3$ input qubits), trained to predict the Lorenz-63 time series [Eq. (4.3)] encoded via amplitude encoding [Eq. (3.12)]. The dashed line represents the forecast horizon using the optimal operator M^* directly, while the connected points correspond to decompositions into subsets of observables [Eqs. (2.13) and (2.14)]. The forecast horizon obtained from the primal optimization over the full operator basis is shown by the dashed-dotted line, with the gray area representing the standard deviation.

Harmonic time series. A scalar harmonic time series consists of n frequencies with random amplitudes $\{a_k, b_k\}$:

$$z(t) = \sum_{k=1}^{n} \left[a_k \cos(k\omega_0 t) + b_k \sin(k\omega_0 t)\right]. \tag{4.4}$$

Learning this time series with a classical linear autoregressive model requires a memory capacity of $2n$ delays [**?** **?**]. Due to the nonlinearity generated by the encoding and the ancilla coupling of a QRC like Eq. (3.2), this bound is only an upper limit in terms of actual number of required linear encoding features. Specifically, the number of ancilla qubits N_A required to store the delays is much less than $2n$. While a detailed study will be presented elsewhere, we find here that $N_A = 3$ and a well tuned memory capacity (via the TFIM parameters) suffices to learn $n \sim 10$ frequencies. We use amplitude encoding [Eq. (3.13)] and a resolution $\Delta t = 0.2$ to feed the time samples $z_n = z(n\Delta t)$ into the QRC. The valid region of Δt follows from the requirements that (i) the fading memory window covers a full period and (ii) the sampling is rate larger than $2f_{\max}$ (Nyquist criterion).

Figure 4(a) illustrates the forecast horizons obtained from the optimized QRC using either the single optimal measurement operator M^* (Eq. (2.8), dashed line) or decompositions into subsets of observables [Eqs. (2.13) and (2.14)]. Using the full M^* provides a stable forecast up to the maximum tested time, which is also reached when using all Pauli basis operators (dots). When using the Pauli Z-observables, the maximum forecast time is only reached for certain random seeds used in constructing the time series (reflected by the rather large error bars). The forecast can become unstable when using a small number of measurement operators, which essentially reflects numerical instabilities unrelated to the optimization process. These issues can, in principle, be alleviated by tuning the regularization parameter, input scale, or input noise (to be analyzed in detail elsewhere).

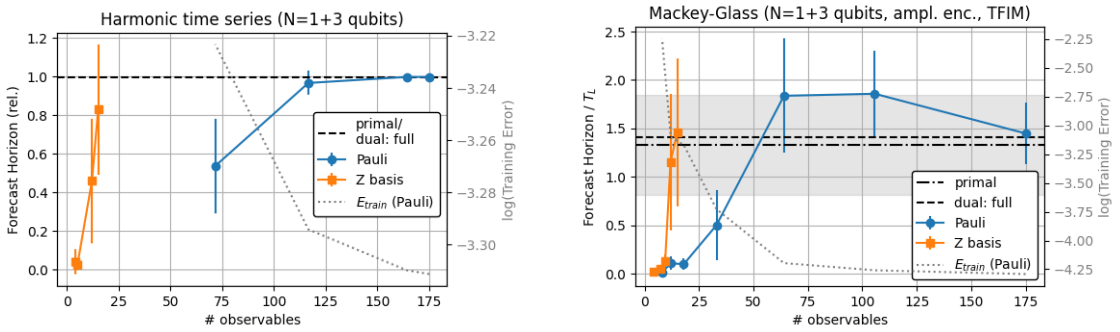

Figure 4. Optimization of the measurement layer for a QRC *with* internal *memory* ($N_A = 3$ ancilla qubits, coupled to $N_I = 1$ input qubits via a TFIM-based unitary). In (a), the QRC is trained on a random harmonic signal [Eq. (4.4)] with $n = 9$ frequencies, encoded via amplitude encoding (3.13). In (b), the Mackey-Glass time series [Eq. (4.5)] is encoded via Eq. (3.12). The dashed line represents the forecast horizon (relative to the maximum tested time in (a)) using the optimal measurement operator M^* (as computed by the kernel-based optimization), while the connected points correspond to decompositions into subsets of observables (Eqs. (2.13) and (2.14), ranked by their relevance, along horizontal axis). The error bars on the forecast horizon represent the standard deviation after averaging over different initial conditions. The forecast horizon obtained from the primal optimization over the full operator basis is shown by the dashed-dotted line, with the gray area representing the standard deviation. E_{train} denotes the (RMS) training error.

Mackey-Glass time series. This model has nonlinear and non-Markovian dynamics and is defined by the delay differential equation [59]

$$\dot{z}(t) = \beta \frac{z(t-\theta)}{1+z(t-\theta)^n} - \gamma z(t) \tag{4.5}$$

with chaotic-regime parameters $\theta = 17$, $\beta = 0.2$, $\gamma = 0.1$, $n = 10$. We evolve Eq. (4.5) numerically with a time step $\Delta t = 1$. Samples $z_n = z(n\Delta t)$ are scaled to the interval $[0, 0.3]$ and injected with the amplitude encoding variant of Eq. (3.12) into a QRC with TFIM-based unitary and $N_A = 3$ ancilla qubits. As seen in Fig. 4(b), the optimized QRC can forecast the Mackey-Glass time series up to ≥ 250 time steps. The Pauli-decomposition with slightly reduced observable count reaches even to ~ 300 time steps, possibly due to numerical stabilization effects not present when using the full operator M^* or the Pauli-Z decomposition. Note that the forecast horizon is necessarily finite due to the intrinsic Lyapunov instability of the Mackey-Glass system, with Lyapunov exponents reported of $\lambda_1 \simeq 6 \times 10^{-3}$ [60].

V. SUMMARY

We have provided a framework, based on the connection between quantum kernel methods and supervised quantum ML models [20], to optimize the measurement operator in quantum reservoir computers (QRCs). This establishes a quantum kernel formulation on history space for recurrent QRCs, connecting the optimal readout observable to the kernel methods previously employed for memoryless quantum models. Our method is thus applicable to both stateless (QELM) and stateful

Step	Stateless (QELM)	Stateful (QRC)
1. Data	Input pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^{P}$	Time series $\{\mathbf{z}_t\}_{t=0}^{T}$; Targets $y_t = \mathbf{z}_{t+1}$
		Effective inputs $x_t := (\mathbf{z}_{t-L}, \ldots, \mathbf{z}_t)$
		$\{(x_t, y_t)\}_{t=L}^{T-1}$ can be treated as unordered samples
2. Encoding	Map each $\mathbf{x}_i$ to state:	To obtain $\rho(x_t)$, evolve for $\tau = t-L, \ldots, t$:
	$\rho_i \equiv \rho(\mathbf{x}_i)$	$\eta(\tau) = U(\eta_I(\mathbf{z}_\tau) \otimes \eta_M(\tau))U^\dagger$, $\eta_M(\tau+1) = \mathrm{tr}_I \eta(\tau)$;
		Then $\rho_t \equiv \rho(x_t) = \eta(t)$
3. Kernel	Gram matrix: $K_{ij} = K(x_i, x_j) = \mathrm{tr}(\rho_i \rho_j)$	
4. Training	Solve for dual coefficients (least-squares): $\boldsymbol{\alpha}^* = (\mathbf{K} + \lambda \mathbb{1})^{-1}\mathbf{y}$	
5. Optimal Observable	$M^* = \sum_{k=1}^{P} \alpha_k^* \rho_k$	
6. Prediction	For new input $\hat{\mathbf{x}}$:	For new input $\hat{x}_t = (\hat{\mathbf{z}}_{t-L}, \ldots, \hat{\mathbf{z}}_t)$:
	$f(\hat{\mathbf{x}}) = \mathrm{tr}(M^* \rho(\hat{\mathbf{x}}))$	$f(\hat{x}_t) = \mathrm{tr}(M^* \rho(\hat{x}_t)) = \sum_{t'} \alpha_{t'}^* K(x_{t'}, \hat{x}_t)$
	$= \sum_i \alpha_i^* K(\mathbf{x}_i, \hat{\mathbf{x}})$	

Table III. Algorithm flow for the kernel-based (dual) optimization of the measurement operator in stateless (memoryless) and stateful (recurrent) QRCs. Note that $\mathbf{y} \in \mathbb{R}^P$ is the vector of target samples and the expression for the dual coefficients α^* thus applies componentwise. In the stateful case, $P = T - L$ is the number of time steps available for training. For details, see Sections A and F.

models, as summarized by the algorithm flow chart in Table III. Since the essence of the method does not rely on the QRC-specific form (2.2) of the output function, insights gained from this work can be transferred to other quantum ML models with a well-defined (and already trained) feature map $\mathbf{x} \mapsto \rho(\mathbf{x})$, enhancing previous discussions of primal-dual approaches in QML [23, 61].

Exact implementation of the optimal measurement operator M^* on an N-qubit system requires measuring all 2^N Z-strings (or, equivalently, measuring in the computational basis) and a pre-measurement rotation of the quantum state by a global (training data-dependent) unitary. This unitary can be understood to capture the outcome otherwise achieved by training a parametrized QML model with fixed measurement operator. Empirically, we find that using operator subsets (e.g., Pauli Z-strings) obtained from decomposing M^* can sometimes improve forecasting performance compared to using the single operator M^* directly. This could be related to the emergence of numerical instabilities or conditioning effects in certain QRC setups and should be investigated further.

Having determined an optimal measurement operator raises the question whether and to which extent the prediction performance of a QRC can be further improved. As will be further discussed elsewhere, in QML models with a time-independent initial-state encoding strategy, the available features are fully determined by the encoding and are subsequently linearly processed by the quantum channels [62]. This processing can manifest as information scrambling and reduce the prediction performance of QML models. In the case of stateless QELMs, the unitary associated with the optimal observable can undo this unitary scrambling [see Eq. (2.14)], which is reflected by the invariance of the trained predictor (2.11) under global conjugation. Accordingly, for stateless QELMs with a unitary channel, optimization of the reservoir unitary can be traded against optimization

of the measurement operator, up to the final measurement/readout parametrization. By contrast, for stateful QRCs with decoherence/reset, or generally for non-unitary channels, optimizing the reservoir dynamics can change the kernel and thus the attainable predictor class. Further meaningful optimizations pertain to the encoding, memory depth and possibly hand-engineered readout features. Alternatively to determining the optimal measurement operator, one may use temporal multiplexing [32, 33] or measuring over a complete measurement basis (full state tomography).

We have numerically demonstrated the feasibility of the optimization for QRCs with $N \lesssim 10$ qubits, noting that, for our considered datasets, it suffices to use a representative subset of the full training samples to evaluate the kernel matrix. Application to larger systems is instead bottlenecked by memory consumption of the 4^N-dimensional operator matrix and requires further methodological developments. Another avenue for future work is a closer look into generalization capability entailed by the optimal measurement operator, since both primal and dual optimization methods provide only error bounds on the training set [12, 63, 64].

ACKNOWLEDGMENTS

This project was made possible by the DLR Quantum Computing Initiative and the Federal Ministry for Research, Technology and Space; https://qci.dlr.de/NeMoQC

Appendix A: QELM as a kernel method

In the following, we describe in detail the mapping between a QRC/QELM and a quantum kernel method and the ensuing measurement operator optimization. This builds on standard material of quantum ML [20, 27, 65–67], adapted specifically to the case of QRCs. In the following, we focus on the QELM formulation and discuss the stateful QRC separately in Section F below.

We consider an N-qubit system with Hilbert space dimension $D = 2^N$, on which a density matrix acts as a feature map $\mathbf{x} \mapsto \rho(\mathbf{x})$. On the real vector space of Hermitian operators $\mathcal{M}(\mathcal{H})$ we use the Hilbert–Schmidt inner product $\langle A, B\rangle = \mathrm{tr}(AB)$ with induced norm $(|A|^{\mathrm{HS}})^2 = \mathrm{tr}(A^2)$. A QELM with a linear readout layer specifies an observable $M \in \mathcal{M}(\mathcal{H})$ and predicts the output

$$f_M(\mathbf{x}) = \mathrm{tr}\big(M\rho(\mathbf{x})\big). \tag{A1}$$

Given training data $(\mathbf{x}_i, y_i)_{i=1}^P$, we optimize the measurement operator M using a Tikhonov-regularized loss function $\mathcal{L}$

$$\min_{M\in\mathcal{M}(\mathcal{H})} \sum_{i=1}^{P} \mathcal{L}\big(f_M(\mathbf{x}_i), y_i\big) + \frac{\lambda}{2}(|M|^{\mathrm{HS}})^2, \qquad \lambda > 0. \tag{A2}$$

Typical loss functions are the squared-error loss $\mathcal{L}(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$ (regression problem) and the hinge loss $\mathcal{L}(\hat{y}, y) = \max(0, 1 - y\hat{y})$ (classification problem).

The representer property holds directly in this operator space [20, 68]. Decompose any M as $M = M_\parallel + M_\perp$, where $M_\parallel \in \mathrm{span}\{\rho(\mathbf{x}_1), \ldots, \rho(\mathbf{x}_P)\}$ and $M_\perp$ is Hilbert–Schmidt orthogonal to this span. Then $\mathrm{tr}(M_\perp \rho(\mathbf{x}_i)) = 0$ for all i, so the data-fit term in Eq. (A2) depends only on $M_\parallel$, while $(|M|^{\mathrm{HS}})^2 = (|M_\parallel|^{\mathrm{HS}})^2 + (|M_\perp|^{\mathrm{HS}})^2$. Replacing M by $M_\parallel$ strictly decreases the objective unless

$M_\perp = 0$. Therefore, there exists an optimal observable M^* in the span of the training data, i.e.,

$$M^* = \sum_{i=1}^{P} \alpha_i^* \rho(\mathbf{x}_i). \tag{A3}$$

Substituting Eq. (A3) into Eq. (A2) produces an optimization over the coefficients $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_P)^\top$. Define the Gram matrix $\mathbf{K} \in \mathbb{R}^{P\times P}$ with $K_{ij} = \mathrm{tr}(\rho(\mathbf{x}_i)\rho(\mathbf{x}_j))$ and the label vector $\mathbf{y} = (y_1, \ldots, y_P)^\top$. For each training point,

$$f_M(\mathbf{x}_i) = \mathrm{tr}\Big(\sum_{j=1}^{P} \alpha_j \rho(\mathbf{x}_j)\, \rho(\mathbf{x}_i)\Big) = \sum_{j=1}^{P} \alpha_j K_{ij}, \tag{A4}$$

and the Hilbert–Schmidt norm evaluates to

$$(|M|^{\mathrm{HS}})^2 = \mathrm{tr}\Big(\sum_i \alpha_i \rho(\mathbf{x}_i) \sum_j \alpha_j \rho(\mathbf{x}_j)\Big) = \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}. \tag{A5}$$

Hence, Eq. (A2) reduces to the convex optimization problem

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^P} \sum_{i=1}^{P} \mathcal{L}\big([\mathbf{K}\boldsymbol{\alpha}]_i, y_i\big) + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}. \tag{A6}$$

For the squared-error loss, this is the kernel ridge regression objective

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^P} \frac{1}{2} \|\mathbf{K}\boldsymbol{\alpha} - \mathbf{y}\|_2^2 + \frac{\lambda}{2} \boldsymbol{\alpha}^\top \mathbf{K} \boldsymbol{\alpha}, \tag{A7}$$

which has a closed solution $\boldsymbol{\alpha}^*$ determined by the linear system $(\mathbf{K} + \lambda \mathbb{1})\, \boldsymbol{\alpha}^* = \mathbf{y}$. For the hinge loss, this becomes equivalent to the dual "quadratic programming problem" [41]:

$$\max_{\boldsymbol{\beta}\in\mathbb{R}^P} \sum_{i=1}^{P} \beta_i - \frac{1}{2\lambda} \sum_{i,j=1}^{P} \beta_i \beta_j y_i y_j K_{ij} \quad \text{subject to} \quad 0 \le \beta_i \le 1, \tag{A8}$$

with $\alpha_i^* = y_i \beta_i^* / \lambda$ (and no bias). In this case, the solution coefficients $\boldsymbol{\alpha}^*$ are not available in closed form, but can be computed numerically.

For any loss function $\mathcal{L}$ in Eq. (A2), the optimal observable and predictor admit the kernel representation

$$M^* = \sum_{i=1}^{P} \alpha_i^*\, \rho(\mathbf{x}_i), \qquad f^*(\mathbf{x}) = \mathrm{tr}\big(M^* \rho(\mathbf{x})\big) = \sum_{i=1}^{P} \alpha_i^*\, K(\mathbf{x}, \mathbf{x}_i), \tag{A9}$$

with kernel

$$K(\mathbf{x}, \mathbf{x}') = \mathrm{tr}\big(\rho(\mathbf{x})\rho(\mathbf{x}')\big). \tag{A10}$$

Thus, for the squared-error loss, measurement optimization for a QELM is exactly kernel ridge regression with kernel Eq. (A10), and the QELM predictor equals the kernel predictor on all inputs when the optimal observable Eq. (A3) is used.

Expanding M^* in terms of an orthogonal operator basis $\{M_k\}$ with $G_{kl} \equiv \mathrm{tr}(M_k M_l) = g_k \delta_{kl}$, yields $M^* = \sum_k w_k M_k$ with

$$w_k^* = \frac{1}{g_k} \mathrm{tr}(M_k M^*) = \frac{1}{g_k} \sum_{i=1}^{P} \alpha_i^* \mathrm{tr}(M_k \rho(\mathbf{x}_i)). \tag{A11}$$

For the Pauli basis, $g_k = 2^N$. Thus, taking

$$\phi_k(\mathbf{x}) = \mathrm{tr}(M_k \rho(\mathbf{x})) \tag{A12}$$

as the feature map [69] and $\Phi_{ik} = \phi_k(\mathbf{x}_i)$ as the feature matrix $\Phi \in \mathbb{R}^{P \times m}$ (with rows $\phi(\mathbf{x}_j)^T$), the QELM predictor can be written in the form of a classical ELM [see Eqs. (E1) and (E5) below]:

$$f^*(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w}^*, \qquad \mathbf{w}^* = G^{-1} \Phi^T \boldsymbol{\alpha}^*. \tag{A13}$$

Using the HS completeness identity $\sum_{k,l} \mathrm{tr}(M_k A)(G^{-1})_{kl} \mathrm{tr}(M_l B) = \mathrm{tr}(AB)$ for any Hermitian operators A, B, it is easy to see that, indeed, $f^*(\mathbf{x}) = \sum_i \alpha_i^* K(\mathbf{x}, \mathbf{x}_i)$.

Consider now the case where $\{M_k\}_{k=1}^m$ is *not a complete basis*, i.e., one cannot generally assume that there exists a $\mathbf{w}$ with $\sum_k w_k M_k = M^*$. The natural generalization of Eq. (A11) is to find a $\mathbf{w}$ that minimizes the (squared) Hilbert–Schmidt distance $\| \sum_k w_k M_k - M^* \|_{\mathrm{HS}}^2 = \mathrm{tr}\big[(\sum_k w_k M_k - M^*)^2\big]$. Solving this optimization problem yields $\mathbf{w} = G^+ \Phi^T \alpha^*$, or component-wise

$$w_k = \sum_{l=1}^{m} (G^+)_{kl} \sum_{j=1}^{P} \alpha_j^* \mathrm{tr}(M_l \rho(\mathbf{x}_j)). \tag{A14}$$

Appendix B: Generalization to multiple outputs/classes

Extension of the framework in Section A to multiple outputs/classes is straightforward. Consider first *regression* with squared-error loss: for C-dimensional targets $\mathbf{y}_i \in \mathbb{R}^C$, one assembles $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_P]^\top \in \mathbb{R}^{P \times C}$, solves $\mathbf{A}^* = (\mathbf{K} + \lambda \mathbb{1})^{-1} \mathbf{Y}$, sets $M_c^* = \sum_{i=1}^P \alpha_{ic}^* \rho(\mathbf{x}_i)$, and predicts $f_c^*(\mathbf{x}) = \sum_{i=1}^P \alpha_{ic}^* K(\mathbf{x}, \mathbf{x}_i)$.

For multi-class *classification* with labels $y_i \in \{1, \ldots, C\}$, one can use one-hot targets and interpret the components $f_c^*(\mathbf{x})$ as class scores, predicting $\hat{y}(\mathbf{x}) = \arg\max_c f_c^*(\mathbf{x})$ (or applying a softmax to obtain class scores). One thus replaces the single observable M by C observables $\{M_c\}_{c=1}^C$ (equivalently a coefficient matrix $\mathbf{A} \in \mathbb{R}^{P \times C}$) with regularizer $\frac{\lambda}{2} \sum_c (|M_c|^{\mathrm{HS}})^2$. The representer form then holds component-wise as $M_c^* = \sum_i \alpha_{ic}^* \rho(\mathbf{x}_i)$, and one-vs-rest hinge/logistic training reduces to C standard kernel problems with the same Gram matrix $\mathbf{K}$.

Appendix C: Diagonalization

Because the computational basis is the joint eigenbasis of all single-qubit Z operators, any diagonal operator admits an expansion in commuting Pauli-Z strings:

$$\Lambda = \sum_{\mathbf{s} \in \{0,1\}^N} \beta_{\mathbf{s}} \, Z_{\mathbf{s}}, \qquad Z_{\mathbf{s}} := \bigotimes_{j=1}^{N} (\sigma_z)^{s_j}, \tag{C1}$$

with coefficients given by a Walsh–Hadamard transform of the spectrum $\{\lambda_b\}$ (see, e.g., [70, 71]),

$$\beta_{\mathbf{s}} = 2^{-N}\mathrm{tr}(\Lambda Z_{\mathbf{s}}) = 2^{-N}\sum_{\mathbf{b}\in\{0,1\}^N} \lambda_{\mathbf{b}}\,(-1)^{\mathbf{s}\cdot\mathbf{b}}. \tag{C2}$$

Consequently, the optimal measurement operator M^* can be expanded in terms of Z-basis measurement operators $Z_{\mathbf{s}}$:

$$M^* = V\Lambda V^\dagger = \sum_{\mathbf{s}\in\{0,1\}^N} \beta_{\mathbf{s}}\, V Z_{\mathbf{s}} V^\dagger. \tag{C3}$$

After the pre-rotation $V^\dagger$, all $Z_{\mathbf{s}}$ can be evaluated from the same computational-basis samples by taking products of single-qubit outcomes; combining these estimates with weights $\{\beta_{\mathbf{s}}\}$ yields the same predictor as the eigenvalue-weighted average in Eq. (2.15). Both procedures use a single circuit per shot.

Appendix D: Pauli basis truncation

Assume we measure only a subset of Pauli observables in the expansion of M^* [Eq. (2.13)]. Let the corresponding index set be $S \subset \{0,\ldots,D^2-1\}$ and define the truncated observable $M_{\mathrm{trunc}} = \sum_{k\in S} c_k P_k$. By Eq. (A1), the pointwise prediction error on any input $\mathbf{x}$ is $\Delta f(\mathbf{x}) = \mathrm{tr}\big((M^* - M_{\mathrm{trunc}})\rho(\mathbf{x})\big)$. Using the Cauchy–Schwarz inequality in the Hilbert–Schmidt inner product yields $|\Delta f(\mathbf{x})| \le |M^* - M_{\mathrm{trunc}}|^{\mathrm{HS}}\,|\rho(\mathbf{x})|^{\mathrm{HS}}$. Since for any feature state $|\rho(\mathbf{x})|^{\mathrm{HS}} \le 1$, the worst-case prediction error is bounded by the magnitude of the omitted coefficients:

$$|\Delta f(\mathbf{x})| \le \|M^* - M_{\mathrm{trunc}}\|_{\mathrm{HS}} = \Big(2^N \sum_{k\notin S} c_k^2\Big)^{1/2}, \tag{D1}$$

where we used orthogonality of the basis.

Appendix E: Classical ELM

It is instructive to review the classical extreme learning machine (ELM) [72] as a paradigmatic example for a kernel formulation. The ELM is essentially a generalized linear regression model [41, 73] with a set of m fixed features $\phi(\mathbf{x}) \in \mathbb{R}^m$ and trainable weights $\mathbf{w} \in \mathbb{R}^m$:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w}. \tag{E1}$$

Recall that a QELM can be mapped to this form by identifying $\phi_k(\mathbf{x}) = \mathrm{tr}[M_k\rho(\mathbf{x})]$ for a set of m observables $\{M_k\}$ [see Eq. (A13)]. Assuming a least-squares loss,

$$\mathcal{L}(\{\mathbf{x}_j, y_j\}) = \frac{1}{2}\sum_{j=1}^{P}[y_j - f(\mathbf{x}_j)]^2 + \frac{\lambda}{2}\|\mathbf{w}\|^2, \tag{E2}$$

and P samples, the optimal solution follows as

$$\mathbf{w}^* = (\Phi^T\Phi + \lambda\mathbb{1})^{-1}\Phi^T\mathbf{y} \tag{E3}$$

with the feature matrix $\Phi \in \mathbb{R}^{P\times m}$ (with rows $\phi(\mathbf{x}_j)^T$) and sample vector $\mathbf{y}$. From the feature vector $\phi(\mathbf{x})$, an associated kernel can be defined as [41, 73, 74]

$$k(\mathbf{x},\mathbf{x}') = \phi(\mathbf{x})^T\phi(\mathbf{x}') \in \mathbb{R}, \tag{E4}$$

which gives rise to the Gram matrix $\mathbf{K} \in \mathbb{R}^{P\times P}$ with $K_{ij} = k(\mathbf{x}_i,\mathbf{x}_j) = (\Phi\Phi^T)_{ij}$. Noting that $(\Phi^T\Phi + \lambda\mathbb{1})^{-1}\Phi^T = \Phi^T(\Phi\Phi^T + \lambda\mathbb{1})^{-1}$, the primal solution in Eq. (E3) can be rewritten as

$$\mathbf{w}^* = \Phi^T\boldsymbol{\alpha}^* = \sum_{j=1}^{P} \alpha_j^*\,\phi(\mathbf{x}_j) \tag{E5}$$

with the dual coefficients

$$\boldsymbol{\alpha}^* = (\mathbf{K} + \lambda\mathbb{1})^{-1}\mathbf{y}. \tag{E6}$$

Accordingly, the trained ELM predictor [Eq. (E1)] can be equivalently written as a kernel model:

$$f(\mathbf{x}) = \phi(\mathbf{x})^T\mathbf{w}^* = \phi(\mathbf{x})^T\Phi^T\boldsymbol{\alpha}^* = \sum_{j=1}^{P}\phi(\mathbf{x})^T\phi(\mathbf{x}_j)\alpha_j = \sum_{j=1}^{P}\alpha_j k(\mathbf{x},\mathbf{x}_j). \tag{E7}$$

The mapping (E5) shows that, even in a potentially infinite feature space, the representer theorem [65, 74] guarantees that the optimal solution can be expressed in terms of a finite number of samples. Primal and dual optimizations thus render the same set of functions. Note that the expressions (E3) and (E5) pertain to the least-squares loss (E2). For other loss functions, the solutions typically have to be determined numerically.

Appendix F: QRC with stateful reservoir

We describe here in some more detail how the kernel-based optimization can be applied to the case of a QRC with internal memory, complementing Section A and demonstrating the viability of prescription Eq. (2.3). Let the input register of the QRC have N_I qubits and the memory register N_A qubits. At time t, a classical input $\mathbf{z}_t \in \mathbb{R}^{d'}$ (typically $d' = N_I$) is encoded via the density matrix $\eta_I(\mathbf{z}_t)$. The joint state $\eta_I(\mathbf{z}_t)\otimes\eta_M(t)$ is time-evolved by a unitary U, giving

$$\eta(t) = U\left(\eta_I(\mathbf{z}_t)\otimes\eta_M(t)\right)U^\dagger. \tag{F1}$$

Subsequently, the input is measured/reset and discarded, and the memory alone is carried to the next step, described by the CPTP map

$$\eta_M(t+1) = \mathrm{tr}_I[\eta(t)]\,. \tag{F2}$$

The stateful QRC dynamics above induces the quantum feature map

$$(\mathbf{z}_0,\ldots,\mathbf{z}_t) =: x_t \longmapsto \rho(x_t) = \eta(t) \in \mathcal{M}(\mathcal{H}), \tag{F3}$$

where $\rho(x_t)$ is obtained after sequentially processing the $\{\mathbf{z}_{t'}\}$ via Eqs. (F1) and (F2). Due to the fading memory of the reservoir, the dependence on inputs earlier than time $t-L$ is negligible. We assume that any transient associated with the reservoir initialization has already decayed before

the observed sequence begins, e.g. by driving the reservoir for times $t < 0$. We therefore work, instead of (F3), with the effective input history (window)

$$x_t := (\mathbf{z}_{t-L}, \dots, \mathbf{z}_t) \in (\mathbb{R}^{d'})^{L+1}, \qquad t = L, \dots, T-1, \tag{F4}$$

and define $P = T - L$ training samples. Let $\rho(x_t)$ denote the reservoir state obtained from driving the reservoir with that history from some fixed initialization (the precise initialization is irrelevant by assumption). The corresponding readout is

$$f_M(x) = \mathrm{tr}\big(M\rho(x)\big). \tag{F5}$$

This exactly corresponds to Eq. (2.2), except that the state $\rho(x_t)$ is to be determined by evolving the reservoir over the input sequence $(\mathbf{z}_{t-L}, \dots, \mathbf{z}_t)$.

For targets $\{y_t\}_{t=L}^{T-1}$, we consider the optimization problem

$$\min_{M \in \mathcal{M}(\mathcal{H})} \frac{1}{2} \sum_{t=L}^{T-1} \big(\mathrm{tr}(M\rho(x_t)) - y_t\big)^2 + \frac{\lambda}{2}\mathrm{tr}(M^2), \tag{F6}$$

which can be straightforwardly extended to other loss functions. By the representer theorem in $(\mathcal{M}(\mathcal{H}), \langle\cdot,\cdot\rangle_{\mathrm{HS}})$, there exists an optimal observable [65, 74],

$$M^* = \sum_{t=L}^{T-1} \alpha_t^* \rho(x_t), \tag{F7}$$

which is the stateful analogue of Eq. (2.8). We now define the kernel on histories/windows

$$K(x, x') := \mathrm{tr}\big(\rho(x)\rho(x')\big), \tag{F8}$$

and the Gram matrix $K_{tt'} := K(x_t, x_{t'})$. Then the coefficients satisfy

$$\boldsymbol{\alpha}^* = (\mathbf{K} + \lambda \mathbb{I})^{-1}\mathbf{y}, \tag{F9}$$

with $\boldsymbol{\alpha}^* = (\alpha_t^*)_{t=L}^{T-1}$ and the target vector $\mathbf{y} = (y_t)_{t=L}^{T-1} \in \mathbb{R}^P$. For any test history/window x (obtained from any test sequence driven through the same reservoir),

$$f^*(x) = \sum_{t=L}^{T-1} \alpha_t^* K(x_t, x). \tag{F10}$$

For a fixed driving sequence, one may write $K(t, t')$ as shorthand for $K(x_t, x_{t'})$. This is the direct stateful counterpart of the kernel representation in Eq. (2.11), with the feature map $\rho(x)$ replaced by the stateful reservoir map $(\mathbf{z}_{t-L}, \dots, \mathbf{z}_t) =: x_t \mapsto \rho(x_t)$. For vector-valued targets $\mathbf{y}_t \in \mathbb{R}^C$ (for example, when predicting all components of $\mathbf{z}_{t+1}$), one can apply the same construction componentwise, as in the stateless case discussed below Eq. (2.10).

Appendix G: Product state encodings and measurements

If $\rho = \bigotimes_{j=1}^N \rho_j$ and $P = \bigotimes_{j=1}^N \sigma^{p_j}$ is a Pauli string, then the measurement on the N-qubit product state gives

$$\mathrm{tr}(P\rho) = \prod_{j=1}^N \mathrm{tr}(\sigma^{p_j}\rho_j). \tag{G1}$$

Any single-qubit state can be expanded in the Pauli basis as

$$\rho_j = \frac{1}{2}\Big(\mathbb{1} + \sum_{p\in\{x,y,z\}} \mathrm{tr}(\sigma^p \rho_j)\,\sigma^p\Big). \tag{G2}$$

For single-qubit rotational encoding (3.10) one has

$$\rho_j(z) = |\psi_{\mathrm{rot}}(z)\rangle\langle\psi_{\mathrm{rot}}(z)| = \begin{pmatrix} \cos^2(\pi z) & \cos(\pi z)\sin(\pi z) \\ \cos(\pi z)\sin(\pi z) & \sin^2(\pi z) \end{pmatrix} = \frac{1}{2}\Big(\mathbb{1}+\sin(2\pi z)\,\sigma^x+\cos(2\pi z)\,\sigma^z\Big), \tag{G3}$$

while for amplitude encoding (3.12):

$$\rho_j(z) = \begin{pmatrix} z & \sqrt{z(1-z)} \\ \sqrt{z(1-z)} & 1-z \end{pmatrix} = \frac{1}{2}\Big(\mathbb{1} + 2\sqrt{z(1-z)}\,\sigma^x + (2z-1)\,\sigma^z\Big). \tag{G4}$$

Table IV summarizes the single-qubit expectation values obtained for rotational and amplitude encoding.

	$\mathrm{tr}(\sigma^p\rho_j)$	
σ^p	rotational	amplitude
$\mathbb{1}$	1	1
σ^x	$\sin(2\pi z)$	$2\sqrt{z(1-z)}$
σ^y	0	0
σ^z	$\cos(2\pi z)$	$2z-1$

Table IV. Single-qubit expectation values for rotational and amplitude encoding.

This implies, in particular, that measuring the above encoded product states with a Pauli string P containing a Y-factor will always yield zero [75]. Action of a unitary breaks Eq. (G1) as it typically spreads any σ^p into a mixture of all other Paulis, such that measuring a Y then will result in nonzero contributions.

[1] J. Bae and L.-C. Kwek, Quantum state discrimination and its applications, J. Phys. A.: Math. Theor. **48**, 083001 (2015), arXiv:1707.02571 [quant-ph].

[2] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nat. Phys. **16**, 1050 (2020).

[3] G. García-Pérez, M. A. Rossi, B. Sokolov, F. Tacchino, P. K. Barkoutsos, G. Mazzola, I. Tavernelli, and S. Maniscalco, Learning to Measure: Adaptive Informationally Complete Generalized Measurements for Quantum Algorithms, PRX Quantum **2**, 040342 (2021).

[4] A. Glos, A. Nykänen, E.-M. Borrelli, S. Maniscalco, M. A. C. Rossi, Z. Zimborás, and G. García-Pérez, Adaptive POVM implementations and measurement error mitigation strategies for near-term quantum devices (2022), arXiv:2208.07817 [quant-ph].

[5] R. Leporini and D. Pastorello, An efficient geometric approach to quantum-inspired classifications, Scientific Reports **12**, 8781 (2022).

[6] C. Gyurik, V. Dyon Vreumingen, and V. Dunjko, Structural risk minimization for quantum linear classifiers, Quantum **7**, 893 (2023).

[7] K. Nakaji, S. Endo, Y. Matsuzaki, and H. Hakoshima, Measurement optimization of variational quantum simulation by classical shadow and derandomization, Quantum **7**, 995 (2023).

[8] D. Lee, K. Baek, J. Huh, and D. K. Park, Variational quantum state discriminator for supervised machine learning, Quantum Science and Technology **9**, 015017 (2024), arXiv:2303.03588 [quant-ph].

[9] S. Y.-C. Chen, H.-H. Tseng, H.-Y. Lin, and S. Yoo, Learning to Program Quantum Measurements for Machine Learning (2025), arXiv:2505.13525 [quant-ph].

[10] S. Y.-C. Chen, H.-H. Tseng, H.-Y. Lin, and S. Yoo, Learning to Measure Quantum Neural Networks, in *2025 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)* (2025) pp. 1–5.

[11] C.-Y. Liu, L. Placidi, K.-C. Chen, S. Y.-C. Chen, and G. Matos, You Only Measure Once: On Designing Single-Shot Quantum Machine Learning Models (2025), arXiv:2509.20090 [cs].

[12] A. Li, T. Li, and F. Li, Enhancing the performance of variational quantum models by optimizing observable measurement based on generalization bounds, Quantum Information Processing **24**, 265 (2025).

[13] H.-Y. Lin, H.-H. Tseng, S. Y.-C. Chen, and S. Yoo, Adaptive Non-local Observable on Quantum Neural Networks (2025), arXiv:2504.13414 [quant-ph].

[14] M. Vetrano, G. Lo Monaco, L. Innocenti, S. Lorenzo, and G. M. Palma, State estimation with quantum extreme learning machines beyond the scrambling time, npj Quantum Information **11**, 20 (2025).

[15] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. H. Liew, Quantum reservoir processing, npj Quantum Information **5**, 35 (2019).

[16] P. Mujal, R. Martínez-Peña, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Time Series Quantum Reservoir Computing with Weak and Projective Measurements, npj Quantum Information **9**, 16 (2023), arXiv:2205.06809 [quant-ph].

[17] L. Innocenti, S. Lorenzo, I. Palmisano, A. Ferraro, M. Paternostro, and G. M. Palma, Potential and limitations of quantum extreme learning machines, Communications Physics **6**, 118 (2023).

[18] W. Xiong, G. Facelli, M. Sahebi, O. Agnel, T. Chotibut, S. Thanasilp, and Z. Holmes, On fundamental aspects of quantum extreme learning machines (2023), arXiv:2312.15124 [quant-ph, stat].

[19] A. Suprano, D. Zia, L. Innocenti, S. Lorenzo, V. Cimini, T. Giordani, I. Palmisano, E. Polino, N. Spagnolo, F. Sciarrino, G. M. Palma, A. Ferraro, and M. Paternostro, Experimental Property Reconstruction in a Photonic Quantum Extreme Learning Machine, Phys. Rev. Lett. **132**, 160802 (2024).

[20] M. Schuld, Supervised quantum machine learning models are kernel methods, Journal of Machine Learning Research 10.1145/3451902.3451903 (2021).

[21] M. Schuld and N. Killoran, Quantum machine learning in feature hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[22] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[23] H.-Y. Huang, M. Broughton, M. Mohseni, *et al.*, Power of data in quantum machine learning, Nature 10.1038/s41586-021-03391-3 (2021).

[24] B. Y. Gan, D. Leykam, and S. Thanasilp, A Unified Framework for Trace-induced Quantum Kernels (2023), arXiv:2311.13552 [quant-ph].

[25] N. Polson, V. Sokolov, and J. Xu, Quantum Bayesian computation, Applied Stochastic Models in Business and Industry **39**, 869 (2023).

[26] P. Rodriguez-Grasa, Y. Ban, and M. Sanz, Neural quantum kernels: Training quantum kernels with quantum neural networks, Phys. Rev. Research **7**, 023269 (2025).

[27] M. Kempkes, A. Ijaz, E. Gil-Fuster, C. Bravo-Prieto, J. Spiegelberg, E. van Nieuwenburg, and V. Dunjko, Double descent in quantum kernel methods, PRX Quantum **7**, 010312 (2026), arXiv:2501.10077 [quant-ph].

[28] M. Hermans and B. Schrauwen, Recurrent Kernel Machines: Computing with Infinite Echo State Networks, Neural Computation **24**, 104 (2012).

[29] J. Dong, R. Ohana, M. Rafayelyan, and F. Krzakala, Reservoir Computing meets Recurrent Kernels and Structured Transforms (2020), arXiv:2006.07310 [stat].

[30] L. Gonon, L. Grigoryeva, and J.-P. Ortega, Reservoir kernels and Volterra series (2022), arXiv:2212.14641 [cs].

[31] L. Grigoryeva, H. L. J. Ting, and J.-P. Ortega, Infinite-dimensional next-generation reservoir computing, Phys. Rev. E **111**, 035305 (2025).

[32] K. Fujii and K. Nakajima, Harnessing Disordered-Ensemble Quantum Dynamics for Machine Learning, Phys. Rev. Applied **8**, 024030 (2017).

[33] K. Fujii and K. Nakajima, Quantum reservoir computing: A reservoir approach toward quantum machine learning on near-term quantum devices (2020), arXiv:2011.04890 [quant-ph].

[34] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, Opportunities in Quantum Reservoir Computing and Extreme Learning Machines, Advanced Quantum Technologies **4**, 2100027 (2021).

[35] L. Domingo, G. Carlo, and F. Borondo, Taking advantage of noise in quantum reservoir computing (2023), arXiv:2301.06814 [quant-ph].

[36] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, Quantum Extreme Reservoir Computation Utilizing Scale-Free Networks, Phys. Rev. Applied **17**, 064044 (2022).

[37] M. Kornjača, H.-Y. Hu, C. Zhao, J. Wurtz, P. Weinberg, M. Hamdan, A. Zhdanov, S. H. Cantu, H. Zhou, R. A. Bravo, K. Bagnall, J. I. Basham, J. Campo, A. Choukri, R. DeAngelo, P. Frederick, D. Haines, J. Hammett, N. Hsu, M.-G. Hu, F. Huber, P. N. Jepsen, N. Jia, T. Karolyshyn, M. Kwon, J. Long, J. Lopatin, A. Lukin, T. Macrì, O. Marković, L. A. Martínez-Martínez, X. Meng, E. Ostroumov, D. Paquette, J. Robinson, P. S. Rodriguez, A. Singh, N. Sinha, H. Thoreen, N. Wan, D. Waxman-Lenz, T. Wong, K.-H. Wu, P. L. S. Lopes, Y. Boger, N. Gemelke, T. Kitagawa, A. Keesling, X. Gao, A. Bylinskii, S. F. Yelin, F. Liu, and S.-T. Wang, Large-scale quantum reservoir learning with an analog quantum computer (2024), arXiv:2407.02553 [quant-ph].

[38] J. Steinegger and C. Räth, Predicting three-dimensional chaotic systems with four qubit quantum systems, Scientific Reports **15**, 6201 (2025), arXiv:2501.15191 [quant-ph].

[39] By "conventional", we mean quantum ML models that have trainable internal layers instead of only a final trainable classical regression layer.

[40] In practice, an initial washout/equilibration time for the QRC must be considered, which, however, does not affect the present theoretical framework.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning* (Springer, Berlin, Heidelberg, 2006).

[42] B. Schölkopf, R. Herbrich, and A. J. Smola, A Generalized Representer Theorem, in *Computational Learning Theory*, edited by D. Helmbold and B. Williamson (Springer, Berlin, Heidelberg, 2001) pp. 416–426.

[43] K. Bükrü, S. Leger, M. L. Hickmann, H.-M. Rieser, R. Sturm, and T. Siefkes, A Hybrid Quantum Solver for Gaussian Process Regression (2025), arXiv:2510.15486 [quant-ph].

[44] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[45] R. LaRose and B. Coyle, Robust data encodings for quantum classifiers, Phys. Rev. A **102**, 032420 (2020), arXiv:2003.01695 [quant-ph].

[46] Y. Wang, Y. Wang, C. Chen, R. Jiang, and W. Huang, Development of variational quantum deep neural networks for image recognition, Neurocomputing **501**, 566 (2022).

[47] C. Yang, J. Qiao, H. Han, and L. Wang, Design of polynomial echo state networks for time series prediction, Neurocomputing **290**, 148 (2018).

[48] J. Pauwels, G. Van der Sande, G. Verschaffelt, and S. Massar, Photonic Reservoir Computer with Output Expansion for Unsupervized Parameter Drift Compensation, Entropy **23**, 955 (2021).

[49] C. Zhu, P. J. Ehlers, H. I. Nurdin, and D. Soh, Minimalistic and Scalable Quantum Reservoir Computing Enhanced with Feedback (2024), arXiv:2412.17817 [quant-ph].

[50] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, Handwritten Digit Recognition with a Back-Propagation Network, in *Advances in Neural Information Processing Systems*, Vol. 2 (Morgan-Kaufmann, 1989).

[51] R. Kharsa, A. Bouridane, and A. Amira, Advances in Quantum Machine Learning and Deep Learning for Image Classification: A Survey, Neurocomputing **560**, 126843 (2023).

[52] A. D. Lorenzis, M. P. Casado, M. P. Estarellas, N. L. Gullo, T. Lux, F. Plastina, A. Riera, and J. Settino, Harnessing Quantum Extreme Learning Machines for image classification, Phys. Rev. Applied **23**, 044024 (2025), arXiv:2409.00998 [quant-ph].

[53] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, Dynamical Phase Transitions in Quantum Reservoir Computing, Phys. Rev. Lett. **127**, 100502 (2021).

[54] O. Ahmed, F. Tennie, and L. Magri, Optimal training of finitely-sampled quantum reservoir computers for forecasting of chaotic dynamics (2024), arXiv:2409.01394 [quant-ph].

[55] O. Ahmed, F. Tennie, and L. Magri, Prediction of chaotic dynamics and extreme events: A recurrence-free quantum reservoir computing approach (2024), arXiv:2405.03390 [quant-ph].

[56] G. McCaul, J. S. T. Gongora, W. Otieno, S. Savelev, A. Zagoskin, and A. G. Balanov, Minimal Quantum Reservoirs with Hamiltonian Encoding (2025), arXiv:2505.22575 [quant-ph].

[57] M. Gross, Flow Map Learning and Harmonic Representations in Nonlinear Vector Autoregressive Models (2026).

[58] J. C. Sprott, *Elegant Chaos: Algebraically Simple Chaotic Flows* (World Scientific, 2010).

[59] M. C. Mackey and L. Glass, Oscillation and Chaos in Physiological Control Systems, Science **197**, 287 (1977).

[60] A. Sadath, T. K. Uchida, and C. P. Vyasarayani, Approximating strange attractors and Lyapunov exponents of delay differential equations using Galerkin projections (2018), arXiv:1810.01016 [physics].

[61] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nature Communications **14**, 517 (2023).

[62] For the product encoding schemes considered in Section III B, the features are essentially polynomials of certain basis functions (e.g., Fourier exponentials or monomials in the input data).

[63] A. Canatar, E. Peters, C. Pehlevan, and S. Wild, Bandwidth enables generalization in quantum kernel models, Trans. Mach. Learn. Res. **2023** (2022).

[64] M. Gross, M. Lange, B. Bujnowski, and H.-M. Rieser, Expressivity vs. Generalization in Quantum Kernel Methods, in *33rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, ESANN 2025*, edited by M. Verleysen (Ciaco - i6doc.com, Bruges, Belgium, 2025) pp. 543–548.

[65] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, 2004).

[66] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge ; New York, 2010).

[67] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers* (Springer, Cham, 2021).

[68] J. Kübler, S. Buchholz, and B. Schölkopf, The Inductive Bias of Quantum Kernels, in *Advances in Neural Information Processing Systems*, Vol. 34 (Curran Associates, Inc., 2021) pp. 12661–12673.

[69] For a complete orthonormal operator basis, ϕ_k represents the expansion coefficients of the feature map $\rho(\mathbf{x})$ under the HS-inner product $\mathrm{tr}(\cdot)$ and thus becomes identical to the *constrained* feature map φ defined in Eq. (2.5) and Table II.

[70] J. Welch, D. Greenbaum, S. Mostame, and A. Aspuru-Guzik, Efficient quantum circuits for diagonal unitaries without ancillas, New. J. Phys. **16**, 033040 (2014).

[71] T. N. Georges, B. K. Berntson, C. Sünderhauf, and A. V. Ivanov, Pauli decomposition via the fast Walsh-Hadamard transform, New. J. Phys. **27**, 033004 (2025).

[72] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, Extreme learning machine: Theory and applications, Neurocomputing Neural Networks, **70**, 489 (2006).

[73] K. P. Murphy, *Probabilistic Machine Learning: An Introduction* (The MIT Press, Cambridge, Massachusetts, 2022).

[74] T. Hofmann, B. Schölkopf, and A. J. Smola, Kernel methods in machine learning, The Annals of Statistics **36**, 1171 (2008).

[75] This does not apply to the dense encoding in Eq. (3.11).