

Received 23 September 2024; revised 13 January 2025; accepted 7 February 2025; date of publication 13 February 2025;  
date of current version 27 March 2025.

Digital Object Identifier 10.1109/TQE.2025.3541882

# Benchmarking Quantum Machine Learning Kernel Training for Classification Tasks

DIEGO ALVAREZ-ESTEVEZ<sup>ID</sup>

CITIC Research Center, Universidade da Coruña, 15071 A Coruña, Spain

Corresponding author: Diego Alvarez-Estevez (e-mail: diego.alvarez@udc.es).

This work was supported in part by the MCIN/AEI/10.13039/501100011033 and the European Social Fund Plus (ESF+) under Project RYC2022-038121-I and in part by the MCIU/AEI/10.13039/501100011033 and the European FEDER program under Project PID2023-147422OB-I00.

**ABSTRACT** Quantum-enhanced machine learning is a rapidly evolving field that aims to leverage the unique properties of quantum mechanics to enhance classical machine learning. However, the practical applicability of these methods remains an open question, particularly beyond the context of specifically crafted toy problems, and given the current limitations of quantum hardware. This study focuses on quantum kernel methods in the context of classification tasks. In particular, it examines the performance of quantum kernel estimation and quantum kernel training (QKT) in connection with two quantum feature mappings, namely, ZZFeatureMap and CovariantFeatureMap. Remarkably, these feature maps have been proposed in the literature under the conjecture of possible near-term quantum advantage and have shown promising performance in ad hoc datasets. This study aims to evaluate their versatility and generalization capabilities in a more general benchmark, encompassing both artificial and established reference datasets. Classical machine learning methods, specifically support vector machines and logistic regression, are also incorporated as baseline comparisons. Experimental results indicate that quantum methods exhibit varying performance across different datasets. Despite outperforming classical methods in ad hoc datasets, mixed results are obtained for the general case among standard classical benchmarks. The experimental data call into question a general added value of applying QKT optimization, for which the additional computational cost does not necessarily translate into improved classification performance. Instead, it is suggested that a careful choice of the quantum feature map in connection with proper hyperparameterization may prove more effective.

**INDEX TERMS** Benchmarking, quantum kernel estimation (QKE), quantum kernel training (QKT), quantum machine learning (QML).

## I. INTRODUCTION

In the crossover of artificial intelligence and quantum computing, the emerging field of quantum machine learning (QML) arises with the aim to harness the distinctive capabilities of quantum mechanics to boost the capabilities of current state-of-the-art (classical) machine learning methods. Work in this field is rapidly evolving, and considerable progress has been made in recent years. However, to what extent QML might live up to its expectations is yet a subject of debate. This motivates the necessity of carrying out benchmark studies that provide a structured framework to assess the performance and potential use of these methods. Indeed, there are still many open questions and challenges to be addressed, in particular, in relation to the limited capabilities of real quantum computing hardware available today. In this context, benchmarking models via classical simulations on small, but heterogeneous, and still challenging datasets might

offer valuable insights to drive the field forward before more capable quantum hardware becomes available. Following this line, this work carries out experimentation in the context of quantum kernel methods to test the performance of the so-called quantum kernel alignment technique [1], [2]. For that purpose, benchmarking is performed that involves different artificial and small-scale reference classification datasets to compare both quantum realizations with their classical counterparts.

The specific interest in kernel methods arises in part as it has recently been shown that variational supervised QML models (often referred to as quantum neural networks) fundamentally reduce to kernel methods. However, kernel methods come with additional advantages, such as the guarantee of finding better or equally good solutions, convex optimization, or even improved performance under certain conditions [3]. Moreover, quantum kernel methods can lead to a

provable separation between classical and quantum learners for certain problems, opening the door for possible quantum advantage achievable even in the current noisy intermediate-scale quantum (NISQ) era [4].

Motivated by these properties, the literature on potential practical applications of quantum kernel methods is growing rapidly. Specific use cases range from finance [5], fraud detection [6], [7], healthcare [8], Earth sciences [9], [10], [11], or physics [12], [13], among others [14], [15]. However, there is a need for benchmark studies that systematically assess the performance analysis of these methods from a more general perspective [16], [17].

Following this line, this study contributes to the existing body of work on performance evaluation by following a general benchmarking approach that involves alternative datasets and methods. In particular, it takes as reference two quantum feature maps, namely, ZZFeatureMap and CovariantFeatureMap, for which the potential for quantum advantage has been postulated in the context of quantum kernel estimation (QKE) using ad hoc data [2], [18]. However, a more extensive evaluation of these methods' capabilities on general-purpose benchmarks, including a comparison to baseline classical counterparts, is still necessary. This is particularly true in the case of CovariantFeatureMap. The study methodically examines the impact of quantum (alignment) kernel training on the generalization performance of the resulting models. Two distinct parameterization strategies are examined to this end: "shared" and "dedicated," applicable to both feature mappings. To the best of the author's knowledge, the impact of these strategies on model generalization has not been previously evaluated in the literature. Furthermore, the "weighted" version of target alignment is used, in contrast to the prevailing practices that utilize the "unweighted" alignment version derived from the seminal work of Hubregsten et al. [1]. As discussed in the work of Glick et al. [2], this "weighted" version theoretically optimizes the support vector machine (SVM) upper bound on the generalization error. However, the benchmarking of the weighted kernel alignment on a wide set of datasets has not yet been performed.

## II. METHODS

The article follows with a concise exposition of kernel methods in the context of classification problems. It then describes the use of quantum kernel estimation (QKE). Finally, it shows how the approach can be extended with parameterized adaptation, resorting to quantum kernel training (QKT).

### A. KERNEL METHODS

Kernel methods are an important class of algorithms in machine learning. The underlying idea is that of transforming the input data into a higher dimensional space where the problem at hand becomes easier to solve. This transformation (i.e., feature mapping) is done using a kernel function, which computes the inner product of two inputs as a measure of

their similarity in the induced higher dimensional space. This procedure is core for a wide range of methods that can be expressed in terms of these operations. The beauty of kernel methods is that the kernel function is designed in such a way that it implicitly computes the inner product in the high-dimensional space, however, without ever explicitly accessing the coordinates of the data in that space, which often is computationally expensive. This is known as the "kernel trick," and for this to be possible, the mapping function has to meet the so-called Mercer's conditions [19].

Kernel methods find application in various areas and cover many machine learning settings, including both supervised and unsupervised learning. They are particularly useful when data are not linearly separable in their original space. By transforming data into a higher dimensional space, kernel methods allow linear algorithms to be applicable to solve nonlinear problems. This study focuses on which arguably is the most famous algorithm within kernel methods, the SVM. More specifically, with application to classification problems, leading to the so-called support vector classifier (SVC).

Let us consider a binary classification dataset of samples  $\{\vec{x}_i\}_{i=1\dots m}$  from input space  $\Omega \subset \mathbb{R}^d$ , with corresponding class labels  $y_i \in \{-1, 1\}$ . The binary SVC classifies examples according to

$$y(\vec{x}) = \text{sgn}(\langle \vec{w}, \phi(\vec{x}) \rangle + b) \quad (1)$$

where the vector  $\vec{w} \in \mathbb{R}^{d_f}$  represents the hyperplane normal to the linear decision boundary that separates the two classes, and lives in the induced feature space that results from the mapping  $\phi$  (and thus, usually  $d_f >> d$ ) and  $b \in \mathbb{R}$  is the intercept, also referred as offset or bias term.

Training the classifier following the SVM formulation corresponds to finding  $\vec{w}$  perpendicular to the separating hyperplane with maximum possible distance between samples lying in the margins of the two classes (also called support vectors). Mathematically, this is equal to solving the following quadratic problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \|\vec{w}\|_2^2 + C \sum_{i=1}^m \xi_i \\ \text{subject to:} \quad & y_i(\langle \vec{w}, \phi(\vec{x}) \rangle + b) + b \geq 1 - \xi_i \\ & \xi_i \geq 0 \quad \forall i = 1, \dots, m \end{aligned} \quad (2)$$

with  $C \geq 0$  acting as regularization parameter for datapoints that are not linearly separable (even in the feature space), for which additional slack variables  $\xi_i$  are introduced acting as penalization terms. This is a convex optimization problem, for which by applying the Lagrangian formulation and noticing that  $\vec{w}$  can be expressed as the weighted sum of the embedded data points  $\vec{w} = \sum_i^m \alpha_i \phi(\vec{x}_i)$ , the following

equivalent Wolfe dual version is obtained:

$$\begin{aligned} \text{maximize } & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \langle \phi(\vec{x}_i), \phi(\vec{x}_j) \rangle \\ = & \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \kappa(\vec{x}_i, \vec{x}_j) \quad (3) \\ \text{subject to: } & \sum_{i=1}^m y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C \quad \forall i = 1, \dots, m \end{aligned}$$

in which application of the “kernel trick” results straightforward, and noticing that for any new data point  $\vec{z}$  not included in the training set, the classification decision can be efficiently calculated as

$$y(\vec{z}) = \text{sgn} \left( \sum_{i=1}^m y_i \alpha_i \kappa(\vec{x}_i, \vec{z}) + b \right). \quad (4)$$

## B. QUANTUM KERNEL ESTIMATION

Variational quantum algorithms, a hybrid quantum–classical approach that relies on the use of parameterized quantum circuits (PQCs), represent the current workhorse of QML methods in the current NISQ era.

QKE is a subclass of these methods where the kernel function is calculated using a PQC that acts as a quantum feature map, embedding datapoints in the original space (classical data) into the induced Hilbert space of quantum states [18], [20]. Let us denote by  $U(\vec{x})$  the unitary representation of the PQC that acts on  $n$ -qubits initialized to the ket-zero state, and the following data-dependent state vector results

$$|\phi(\vec{x})\rangle = U(\vec{x})|0^{\otimes n}\rangle. \quad (5)$$

The quantum state vector representation, however, does only have physical meaning up to global phase. This is problematic, for which the inner product of two physically identical state vectors might result in different values. Thus, in practice, the mapping is considered in the space of density matrices, which for pure states is given by

$$\Phi : \vec{x} \in \Omega \rightarrow \rho(\vec{x}) = |\phi(\vec{x})\rangle \langle \phi(\vec{x})|. \quad (6)$$

Hence, given two samples  $\vec{x}_i$  and  $\vec{x}_j$  and using the Hilbert–Schmidt inner product between the corresponding density matrices  $\rho(\vec{x}_i)$  and  $\rho(\vec{x}_j)$ ,  $\langle \rho(\vec{x}_i), \rho(\vec{x}_j) \rangle = \text{Tr}\{\rho(\vec{x}_i)\rho(\vec{x}_j)\}$ , the following kernel function is naturally derived:

$$\kappa(\vec{x}_i, \vec{x}_j) = |\langle \phi(\vec{x}_j) | \phi(\vec{x}_i) \rangle|^2 \quad (7)$$

which corresponds to the fidelity or overlap between the corresponding  $|\phi(\vec{x}_i)\rangle$  and  $|\phi(\vec{x}_j)\rangle$  quantum states. Following (5) we have

$$\kappa(\vec{x}_i, \vec{x}_j) = |\langle 0^{\otimes n} | U^\dagger(\vec{x}_j) | U(\vec{x}_i) | 0^{\otimes n} \rangle|^2. \quad (8)$$

There are several algorithms to estimate the overlap of two arbitrary quantum states. One straightforward approach is to

use a SWAP circuit. It has the advantage that it works with mixed states, but it requires the use of some auxiliary qubits resulting in a circuit of width  $2n + 1$ . Also, depending on the quantum hardware architecture, extra depth might be required for the implementation of the required controlled SWAP operations. However, because, in this case, the author knows exactly how the state was created, an alternative approach is to apply the encoding circuit, followed by the adjoint of its unitary,  $U^\dagger(\vec{x})$ , and finally perform measurement to record the frequency and the output is found to be  $|0^{\otimes n}\rangle$ . That is, the approach literally follows optimization expression (3) and results in a circuit of depth  $2n$ . However, one has to bear in mind that this protocol is only applicable in the context of pure states [18].

Among the nice properties of QKE, it follows that as the kernel computation does only represent a small subroutine in the general hybrid pipeline, usually long coherence times are not required. In addition, because kernel values can be estimated using circuits, i.e., short-depth circuits, it makes it an ideal candidate in the context of current NISQ devices. Furthermore, regularization and error-mitigation techniques have been discussed that make QKE estimation method described above robust even in the presence of certain amount of noise [1], [18].

One crucial consideration in the pursuit of potential quantum advantage, in terms of the infeasibility of classical simulation, regards the complexity of the quantum circuit. In this regard, it is important that the underlying represented state does not simplify to a product state. QKE may also offer additional benefits beyond the discussion on the achievement of possible quantum speed-ups. These include the potential to access more expressive and possibly better separable induced feature spaces. This subject is currently an active area of research.

## C. QUANTUM KERNEL TRAINING

One should notice that under the QKE technique described in the previous section, the unitary  $U(\vec{x})$  remains purely dependent on input data, and it admits no further optimization once the specific structure of the representative PQC has been chosen. However, the proper choice of the optimal feature map depends on the specific targeted problem in practice. To partially mitigate this challenge, one might consider to fit additional parameters  $\theta$  for a variational family of kernels, leading to QKT technique that is introduced next.

### 1) CLASSICAL BACKGROUND

Classical kernel-target alignment technique relies on the idea of using of a proxy quantity, namely, alignment, which enables efficient evaluation of the appropriateness of choosing a specific kernel function for the targeted learning problem [21]. Indeed, the success of kernel methods is very much dependent on the appropriate choice of a kernel function such that the geometrical structure of the mapped data in the feature space fits the specific learning target. In the case of SVMs, it would conceptually translate to the possibility of

achieving maximum margin linear separability between the target classes in the induced feature space.

Let  $K_1$  and  $K_2$  be two kernel matrices, and the empirical alignment derives from evaluating and normalizing their inner product, that is

$$A(K_1, K_2) = \frac{\langle K_1, K_2 \rangle}{\sqrt{\langle K_1, K_1 \rangle \langle K_2, K_2 \rangle}} \quad (9)$$

which corresponds to calculation of their cosine similarity in the case  $\langle K_1, K_2 \rangle$  is a real nonnegative value, a condition that is guaranteed for which valid kernel matrices are positive semidefinite.

Given a parameterized kernel family function  $\kappa_\theta$ , Cristiani et al. [21] showed that, by choosing  $K_1 = \kappa_\theta(\vec{x}_i, \vec{x}_j)$  and  $K_2 = y_i y_j$ , and optimizing (maximizing)  $A(K_1, K_2)$  on the training set, one is expected to keep high alignment and achieve good classification generalization on the test set as well. Of note, substituting in (9) and operating

$$\text{TA}(\kappa_\theta) = \frac{\sum_{i,j=1}^m y_i y_j \kappa_\theta(\vec{x}_i, \vec{x}_j)}{\sqrt{(\sum_{i,j=1}^m \kappa_\theta(\vec{x}_i, \vec{x}_j)^2)(\sum_{i,j=1}^m y_i^2 y_j^2)}}. \quad (10)$$

Hence, iterative optimization methods can be applied to choose the best parameter set  $\vec{\theta}_{\text{opt}}$  that maximizes  $\text{TA}(\kappa_\theta)$ . This is in contrast to the more complex (often widespread) approach of considering kernel optimization as a problem of model selection using uninformed cross-validation approaches that involve computationally intensive training of kernel machines. For a more comprehensive overview of (classical) kernel alignment, its mathematical justification, and applications, the author refers to the nice review by Wang et al. [22].

## 2) QUANTUM APPROXIMATIONS

Inspired by ideas on classical kernel–target alignment, QKT has been proposed as hybrid quantum–classical optimization algorithm in the context of QKE [1]. In this setting, kernel–target alignment is used as a cost function from which to iteratively derive new parameters for an embedding quantum circuit, which is dependent on both the input data points  $\vec{x}$  and variational parameters  $\vec{\theta}$ . The quantum processing unit (QPU) is used at each variational step to obtain the corresponding kernel matrix entries  $\kappa_\theta(\vec{x}_i, \vec{x}_j)$ . That is, the data embedding process described in (5) now becomes

$$|\phi_\theta(\vec{x})\rangle = U_\theta(\vec{x})|0^{\otimes n}\rangle \quad (11)$$

and the associated QKE routine

$$\kappa_\theta(\vec{x}_i, \vec{x}_j) = |\langle 0^{\otimes n}|U_\theta^\dagger(\vec{x}_j)|U_\theta(\vec{x}_i)|0^{\otimes n}\rangle|^2. \quad (12)$$

Alternatively, the use of weighted kernel alignment has been recently introduced for the optimization of the fiducial state in the context of (quantum) covariant quantum kernels for data spaces that have a group structure [2].

Under their approach, the SVM optimization problem (3) directly translates to

$$\min_{\vec{\theta}} \max_{\vec{\alpha}} F(\vec{\alpha}, \vec{\theta}) = \sum_{i=1}^m \alpha_i - \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \kappa_\theta(\vec{x}_i, \vec{x}_j) \quad (13)$$

subject to the same  $\alpha_i$  constraints.

Indeed, noticing from (10) that

$$\text{TA}(\kappa_\theta) \propto \sum_{i,j=1}^m y_i y_j \kappa_\theta(\vec{x}_i, \vec{x}_j) \quad (14)$$

it derives that the second term of (13) involves the kernel alignment weighted by the Lagrange multipliers, hence the name. Notably, whereas the unweighted version of the alignment, cf., (10), implies finding kernels that maximize the intraclass and minimize the interclass overlap for all points included in the training set, its weighted counterpart aims as well to maximize the margin gap around the decision boundary. Consequently, only the subset of support vectors contribute to the alignment (for all the rest  $\alpha_i = 0$ ). Also remarkable, because of the negative sign in the second term, solving the quadratic SVM dual problem involves minimizing over the kernel parameters  $\theta$ . It can be shown that solving this min–max problem corresponds to finding  $\theta_{\text{opt}}$  such that  $\kappa_{\theta_{\text{opt}}}$  minimizes the upper bound of the SVM generalization error [2]. In general, it is also noted that kernel optimization using the kernel–target alignment is closely related to the “metric learning” approach proposed in [23] for the training of quantum feature embeddings.

The problem (13) is solved following a variant of the stochastic classical–quantum iterative algorithm proposed by the same authors. The algorithm evaluates kernel matrices on a quantum processor and updates parameters  $\theta$  and  $\alpha$  for the next iteration with classical optimizers. The use of simultaneous perturbation stochastic approximation (SPSA) is proposed for the evolution of  $\theta$ , for which minimization of (13) with respect to the kernel parameters leads to highly nonconvex landscape in general. For this purpose, the method approximates the gradient of the loss with respect to  $\theta$  using only two objective function evaluations at each iteration, regardless of the dimension of  $\theta$ . This involves evaluating the corresponding kernel matrices, which is done using the QPU following (12). Once kernel matrices have been calculated, standard SVM classical solvers can be used for maximization with respect to  $\alpha$  in the min–max problem according to (13), which remains concave. One additional advantage is that SPSA regards its robustness in the context of noisy objective functions, which is convenient given kernels are evaluated on noisy quantum hardware.

## III. EXPERIMENTAL SETTING

The main focus of the followed experimentation (and of this work) is dual purpose. First, it aims to test the versatility and generalization capabilities of some specific quantum feature mapping circuits that have been proposed in the context of QML-related literature (and more specifically, QKE), for

which the possibility of quantum advantage has been conjectured [2], [18]. The quantum feature mappings in question have been shown to be able to achieve perfect separability on specifically targeted (ad hoc) classification learning problems. However, in the related reference works, no attempt was performed to solve the same task using classical learning counterparts. Moreover, it remains an open question to which extent these mappings could find some practical application in solving other learning tasks beyond the aforementioned artificially generated datasets. For this purpose, in addition to trying to reproduce results shown in these seminal works, the target is to expand the benchmarking to include additional well-known classical machine learning datasets, as well as the use of classical SVMs as baseline methods. Second, taking as reference the same benchmark database, the objective is to evaluate the contribution of QKT using weighted alignment following the methods described in Section II-C.

#### A. FEATURE MAPPINGS

In this work, two quantum feature maps, namely, ZZFeatureMap and CovariantFeatureMap are taken as reference.

The first one, related to the IQP circuit family introduced in [24] (and thus sometimes also regarded as IQP-circuit throughout the literature), derives from a family of encoding circuits that transform input data  $\vec{x} \in \mathbb{R}^d$  applying the following  $n$ -qubit unitary to evolve from the  $|0\rangle^{\otimes n}$  initialization state [18]:

$$U(\vec{x}) = \prod_d \text{PE}_{\Phi(\vec{x})} H^{\otimes n}$$

with PE denoting the Pauli expansion

$$\text{PE}_{\Phi(\vec{x})} = \exp \left( i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{k \in S} P_i \right)$$

where

$$\phi_S : \vec{x} \mapsto \begin{cases} x_i, & \text{if } S = \{i\} \\ (\pi - x_i)(\pi - x_j), & \text{if } S = \{i, j\} \end{cases}$$

and in which  $P_i \in \{I, X, Y, Z\}$ , i.e., the set of Pauli matrices,  $S \in \binom{[n]}{k}$  combinations,  $k = 1, \dots, n$ , describes the qubit connectivity, and  $d$  is the number of repetitions that determine the circuit depth. More specifically, the ZZFeatureMap results from choosing  $d = 2$ ,  $k = 2$ ,  $P_0 = Z$ , and  $P_1 = ZZ$ , leading to the following relatively shallow circuit, which has been conjectured to be hard to estimate classically [18]:

$$U_{ZZ}(\vec{x}) = \left[ \exp \left( i \sum_{jk} \phi_{\{j,k\}}(\vec{x}) Z_j \otimes Z_k \right) \times \exp \left( i \sum_j \phi_{\{j\}}(\vec{x}) Z_j \right) H^{\otimes n} \right]^2. \quad (15)$$

The second quantum feature map at test, i.e., CovariantFeatureMap, derives from the already mentioned work on covariant quantum kernels for data with group structure [2]. The motivational hypothesis is that advanced classical kernels for group data are computationally expensive to evaluate, and thus in practice, one has to resort to approximation methods for their computation. Thus, a potential advantage of quantum kernels is that such approximation may not be necessary [2]. This hypothesis is supported by previous work showing that covariant quantum kernels can lead to provable quantum advantage over classical learners for specifically constructed learning problems [4]. Generalizing this result, Glick et al. [2] proposed covariant feature maps characterized as  $U_\theta(\vec{x}) = D(\vec{x})V(\vec{\theta})$ , with  $D(\vec{x})$  corresponding to the unitary representation of a group  $G$ , with  $\vec{x} \in G$ , and  $V(\vec{\theta})$  representing a parameterized (fiducial) state, such that  $|\phi_\theta(\vec{x})\rangle = D(\vec{x})|\phi_\theta\rangle = D(\vec{x})V(\vec{\theta})|0^{\otimes n}\rangle$ .

More specifically, based on the synthetic labeling cosets with error (LCE) problem, the specific parameterized (ad hoc) mapping was characterized by choosing

$$D(\vec{x}) = \otimes_{k=1}^n R_X(x_{2k-1})R_Z(x_{2k}) \quad (16)$$

$$V(\theta) = \prod_{(k,t) \in E} CZ_{k,t} \prod_{k \in V} R_{Y_k}(\theta) \quad (17)$$

where  $R_X$  and  $R_Y$  stand for the single-qubit rotation operators around the corresponding axes of the Bloch sphere, with  $R_{Y_k}$  the expanded version of the latter in the  $n$ -qubit space affecting qubit  $k$ , and CZ the two-qubit corresponding Controlled-Z entanglers indexed by  $k$ , with target  $t$ , according to the (graph-stabilizer) subgroup  $S_{\text{graph}} < G = SU(2)^{\otimes n}$ , for the graph with edges  $E$  and vertices  $V$ . From (17), it follows that parameterization of the fiducial state  $|\phi_\theta\rangle = V(\theta)|0^{\otimes n}\rangle$  in this case is made dependent on one single scalar  $\theta$ .

Arranging from (16) and (17) it results into the following data-dependent unitary:

$$U_{\text{Cov}}(\vec{x}) = [\otimes_{k=1}^n R_X(x_{2k-1})R_Z(x_{2k})] \prod_{k=1}^n CZ_{k,k+1} \quad (18)$$

thus grouping together all  $\theta$ -independent operations, leaving aside the preparation of the fiducial state (described in more detail in the next section). From (18) it is also noted that the group-specific CZ mapping, originally optimized for the LCE problem, was replaced for a more general linear entangling layer. In doing so, one seeks to come up with a more general expression that enable us to apply the resulting mapping to a wider range of problems where the possible underlying graph structure is a priori unknown.

#### 1) OPTIMIZATION OF THE KERNEL PARAMETERS

Here, the weighted kernel alignment procedure described above is followed to fit

$$\kappa_\theta(\vec{x}_i, \vec{x}_j) = |\langle 0^{\otimes n} | V_\theta^\dagger U_x^\dagger | U_x V_\theta | 0^{\otimes n} \rangle|^2. \quad (19)$$

**TABLE 1.** Summary of Datasets Included in the Benchmarking

Dataset	Feature dimension	Training dist.	Test dist.
Ad-hoc-ZZ	2	100 - 100	100 - 100
Ad-hoc-COV	14	32 - 32	32 - 32
Linear-IRIS	4	35 - 35	15 - 15
Nonlinear-IRIS	4	35 - 35	15 - 15
MNIST-PCA-4	4	125 - 125	125 - 125
MNIST-PCA-8	8	125 - 125	125 - 125
MNIST-1D-PCA-4	4	125 - 125	50 - 50
MNIST-1D-PCA-8	8	125 - 125	50 - 50

The number of positive and negative instances in the corresponding training and testing set partitions are shown in the last two columns.

In the experimental setting,  $U_x \in \{U_{ZZ}(\vec{x}), U_{COV}(\vec{x})\}$ , respectively, from (15) and (18), and  $V_\theta \in \{V_{shared}(\vec{\theta}), V_{dedicated}(\vec{\theta})\}$ , where

$$V_{shared}(\vec{\theta}) = \otimes_{k=1}^n R_{XYZ}(\theta_1, \theta_2, \theta_3) \quad (20)$$

$$V_{dedicated}(\vec{\theta}) = \otimes_{k=1}^n R_{XYZ}(\theta_{3(k-1)+1}, \theta_{3(k-1)+2}, \theta_{3(k-1)+3}) \quad (21)$$

with  $R_{XYZ}$  the universal single-qubit rotation gate with three Euler angles

$$R_{XYZ}(\theta_1, \theta_2, \theta_3) = \begin{pmatrix} \cos\left(\frac{\theta_1}{2}\right) & -e^{i\theta_2} \sin\left(\frac{\theta_1}{2}\right) \\ e^{i\theta_3} \sin\left(\frac{\theta_1}{2}\right) & e^{i(\theta_3+\theta_2)} \cos\left(\frac{\theta_1}{2}\right) \end{pmatrix}. \quad (22)$$

One should notice that by choosing  $U_x = U_{COV}(\vec{x})$  and  $V_{shared}(\theta_1, \theta_2 = 0, \theta_3 = 0)$ , the mapping (16) is retrieved for a linearly connected graph (for which  $R_{XYZ}(\theta, 0, 0) = R_Y(\theta)$ ). Thus, once again, by using (20) and (21), the aim is to generalize the expressivity of the resulting kernel mapping, allowing additional degrees of freedom for adaptability to wider set of classification problems. More specifically, and as the naming suggests,  $V_{shared}(\vec{\theta})$  leads to all  $n$ -qubits in the mapping circuit to share the same parameterized rotation throughout all possible configurations of the Bloch sphere, whereas  $V_{dedicated}(\vec{\theta})$  involves a dedicated rotation for each qubit.

## B. BENCHMARK DATASETS

Here, a benchmark dataset is described to test the performance of QKE and QKT techniques on diverse classification tasks using the methods and quantum feature mappings described above.

One proceeds to that objective, fully aware of the limitations: gathering a well-diverse and general-enough benchmark dataset that covers all possible situations and enables extraction of unequivocal conclusions in comparing performance between classical and QML is a difficult endeavor, if not Utopian [17]. Regardless, despite the challenges, one can still strive for a “fair enough” dataset that captures a broad spectrum of scenarios. The effort is still justified in serving as a valuable resource toward continuing unraveling the potential of quantum computing in machine learning.

With that in mind, a mix of quantum-inspired artificially generated data are assembled, together with some reference

classical machine learning datasets to test versatility and generalization capabilities of both quantum and classical methods in heterogeneous settings.

## 1) AD HOC QUANTUM DATASETS

An obvious (baseline) choice is to use synthetic ad hoc data representing classification problems that should be easily separable by the investigated quantum feature mappings, but are theoretically hard for classical methods.

In this regard, here it is denoted by Ad-hoc-ZZ dataset, to a 2-D 400 sample of ad hoc data generated according to the procedure described in [18] for the ZZFeatureMap. Data are split in 200 samples corresponding to train and test partitions, each including a balanced representation of the positive and negative classes. Default  $\Delta = 0.3$  separation gap parameter is used.

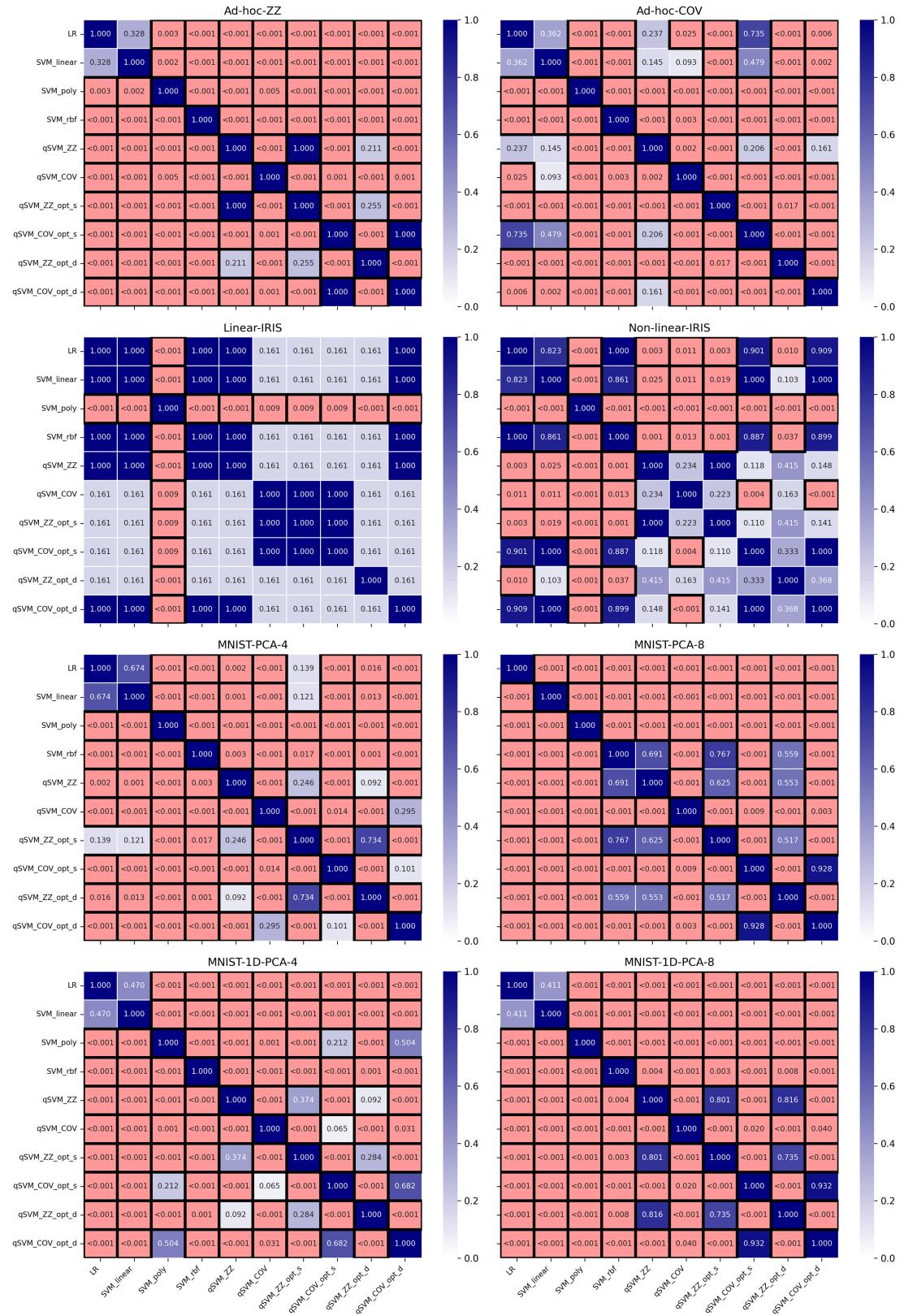
Analogously, the LCE classification problem is considered, outlined in the work describing covariant quantum mappings [2]. More specifically, a seven-qubit instance of the problem is used, which is available online in a related experimental repository on GitHub [25]. The resulting Ad-hoc-COV dataset comprises 14-D feature vectors with independently sampled training and testing partitions, each consisting of 32 data points per label.

## 2) REFERENCE CLASSICAL DATASETS

Different instances of two well-known datasets in classical machine learning are considered, namely, the IRIS and the MNIST classification datasets. The author also includes a third one, namely, MNIST-1D, especially designed low-scale model evaluation.

First, by Linear-IRIS, it is denoted the data collection that consists of the first two classes (*setosa* versus *versicolor*) of the famous classification dataset. As the name suggests, these two classes are known to be linearly separable. The resulting dataset involves 100 4-D data samples (50 for each class). Train and test partitions are arranged using a 70%–30% stratified split. A second dataset, referred to as Nonlinear-IRIS, is composed that comprises the *versicolor* versus *virginica* classification task, which cannot be fully separated without resorting to nonlinear feature projections. Similarly, in this case, a 70%–30% rule is used for training and test splitting.

From the reference MNIST dataset, two additional instances of a downscaled version of the three versus five digits’ classification task are considered. In particular, the 4-D and 8-D versions of the MNIST-PCA benchmark from [17] are taken as reference. As the name suggests, the generation of these variants involves the application of principal component analysis (PCA) to the original MNIST dataset for the targeted dimensionality reduction, after which a stratified subsample of 250 training and test points is taken to constrain the resulting complexity within the reach range of quantum kernel methods. The two resulting datasets are denoted as MNIST-PCA-4 and MNIST-PCA-8, respectively.



**FIGURE 1.** P-values for t-test comparing paired classification accuracy scores in the testing set for each combination of tested models and for each dataset.

**TABLE 2.** Experimental Results for Quantum Methods

Dataset	Mapping	QKT approach (# parameters)	ACC <sub>TR</sub>	$\kappa_{TR}$	MF <sub>TR</sub>	ACC <sub>TS</sub>	$\kappa_{TS}$	MF <sub>TS</sub>
Ad-hoc-ZZ	<b>ZZFeatureMap</b>	—	<b>0.99 (0.01)</b>	<b>0.99 (0.03)</b>	<b>0.99 (0.01)</b>	<b>0.99 (0.02)</b>	<b>0.98 (0.03)</b>	<b>0.99 (0.02)</b>
		Shared (3)	<b>0.99 (0.01)</b>	<b>0.99 (0.03)</b>	<b>0.99 (0.01)</b>	<b>0.99 (0.01)</b>	<b>0.98 (0.03)</b>	<b>0.99 (0.01)</b>
		Dedicated (6)	<b>0.99 (0.01)</b>	<b>0.99 (0.03)</b>	<b>0.99 (0.01)</b>	<b>0.99 (0.02)</b>	<b>0.98 (0.05)</b>	<b>0.99 (0.02)</b>
	CovariantMap	—	0.60 (0.03)	0.19 (0.07)	0.59 (0.03)	0.57 (0.04)	0.15 (0.08)	0.57 (0.04)
		Shared (3)	0.64 (0.05)	0.29 (0.10)	0.64 (0.05)	0.61 (0.07)	0.22 (0.14)	0.61 (0.07)
		Dedicated (3)	0.64 (0.05)	0.29 (0.10)	0.64 (0.05)	0.61 (0.07)	0.22 (0.14)	0.61 (0.07)
Ad-hoc-COV	<b>ZZFeatureMap</b>	—	<b>0.99 (0.01)</b>	<b>0.99 (0.02)</b>	<b>0.99 (0.01)</b>	0.92 (0.07)	0.84 (0.14)	0.92 (0.07)
		Shared (3)	0.77 (0.07)	0.55 (0.14)	0.77 (0.07)	0.72 (0.07)	0.44 (0.13)	0.71 (0.07)
		Dedicated (42)	0.79 (0.07)	0.58 (0.14)	0.79 (0.07)	0.74 (0.04)	0.48 (0.09)	0.73 (0.04)
	CovariantMap	—	0.97 (0.02)	0.94 (0.05)	0.97 (0.02)	0.88 (0.03)	0.76 (0.06)	0.88 (0.03)
		Shared (3)	<b>0.99 (0.02)</b>	0.97 (0.04)	<b>0.99 (0.02)</b>	0.91 (0.03)	0.81 (0.06)	0.91 (0.03)
		Dedicated (21)	<b>0.99 (0.01)</b>	0.98 (0.02)	<b>0.99 (0.01)</b>	<b>0.94 (0.04)</b>	<b>0.88 (0.08)</b>	<b>0.94 (0.04)</b>
Linear-IRIS	<b>ZZFeatureMap</b>	—	<b>1.00 (0.00)</b>					
		Shared (3)	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>	<b>1.00 (0.00)</b>	0.99 (0.02)	0.99 (0.05)	0.99 (0.03)
		Dedicated (12)	<b>1.00 (0.00)</b>	<b>1.00 (0.01)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.02)</b>	<b>1.00 (0.05)</b>	<b>1.00 (0.03)</b>
	CovariantMap	—	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	0.99 (0.02)	0.99 (0.05)	0.99 (0.03)
		Shared (3)	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	<b>1.00 (0.00)</b>	0.99 (0.02)	0.99 (0.05)	0.99 (0.03)
		Dedicated (6)	<b>1.00 (0.00)</b>					
Non-linear-IRIS	<b>ZZFeatureMap</b>	—	0.95 (0.02)	0.90 (0.03)	0.95 (0.02)	0.93 (0.03)	0.85 (0.06)	0.93 (0.03)
		Shared (3)	0.95 (0.02)	0.90 (0.04)	0.95 (0.02)	0.93 (0.03)	0.85 (0.06)	0.93 (0.03)
		Dedicated (12)	0.96 (0.02)	0.93 (0.03)	0.96 (0.02)	0.93 (0.04)	0.86 (0.08)	0.93 (0.04)
	CovariantMap	—	0.96 (0.02)	0.92 (0.03)	0.96 (0.02)	0.92 (0.04)	0.84 (0.07)	0.92 (0.04)
		Shared (3)	<b>0.97 (0.02)</b>	<b>0.94 (0.03)</b>	<b>0.97 (0.02)</b>	<b>0.94 (0.05)</b>	<b>0.88 (0.09)</b>	<b>0.94 (0.05)</b>
		Dedicated (6)	0.96 (0.02)	0.93 (0.04)	0.96 (0.02)	<b>0.94 (0.04)</b>	<b>0.88 (0.09)</b>	<b>0.94 (0.04)</b>
MNIST-PCA-4	<b>ZZFeatureMap</b>	—	0.83 (0.04)	<b>0.67 (0.09)</b>	<b>0.83 (0.04)</b>	<b>0.80 (0.04)</b>	<b>0.61 (0.07)</b>	<b>0.80 (0.04)</b>
		Shared (3)	0.83 (0.07)	0.66 (0.13)	<b>0.83 (0.08)</b>	<b>0.80 (0.06)</b>	0.60 (0.12)	0.79 (0.07)
		Dedicated (12)	<b>0.84 (0.05)</b>	<b>0.67 (0.11)</b>	<b>0.83 (0.06)</b>	<b>0.80 (0.04)</b>	0.60 (0.08)	<b>0.80 (0.04)</b>
	CovariantMap	—	0.69 (0.02)	0.38 (0.03)	0.69 (0.02)	0.69 (0.03)	0.38 (0.06)	0.68 (0.03)
		Shared (3)	0.71 (0.04)	0.41 (0.07)	0.70 (0.04)	0.71 (0.04)	0.42 (0.08)	0.70 (0.05)
		Dedicated (6)	0.70 (0.03)	0.40 (0.06)	0.70 (0.03)	0.70 (0.03)	0.39 (0.07)	0.69 (0.04)
MNIST-PCA-8	<b>ZZFeatureMap</b>	—	<b>0.97 (0.01)</b>	0.94 (0.03)	<b>0.97 (0.01)</b>	<b>0.95 (0.01)</b>	<b>0.89 (0.02)</b>	<b>0.95 (0.01)</b>
		Shared (3)	<b>0.97 (0.01)</b>	<b>0.95 (0.03)</b>	<b>0.97 (0.01)</b>	<b>0.95 (0.01)</b>	<b>0.89 (0.03)</b>	<b>0.95 (0.01)</b>
		Dedicated (24)	<b>0.97 (0.01)</b>	<b>0.95 (0.02)</b>	<b>0.97 (0.01)</b>	0.94 (0.02)	<b>0.89 (0.03)</b>	0.94 (0.02)
	CovariantMap	—	0.91 (0.01)	0.81 (0.03)	0.91 (0.01)	0.88 (0.02)	0.76 (0.05)	0.88 (0.02)
		Shared (3)	0.92 (0.03)	0.84 (0.05)	0.92 (0.03)	0.89 (0.03)	0.78 (0.05)	0.89 (0.03)
		Dedicated (12)	0.92 (0.03)	0.83 (0.05)	0.92 (0.03)	0.89 (0.02)	0.78 (0.05)	0.89 (0.02)
MNIST-1D-PCA-4	<b>ZZFeatureMap</b>	—	<b>0.89 (0.03)</b>	0.78 (0.06)	<b>0.89 (0.03)</b>	0.85 (0.04)	0.70 (0.08)	0.85 (0.04)
		Shared (3)	<b>0.89 (0.03)</b>	0.78 (0.06)	<b>0.89 (0.03)</b>	0.85 (0.04)	<b>0.71 (0.08)</b>	0.85 (0.04)
		Dedicated (12)	<b>0.89 (0.03)</b>	<b>0.79 (0.06)</b>	<b>0.89 (0.03)</b>	<b>0.86 (0.04)</b>	<b>0.71 (0.07)</b>	<b>0.86 (0.04)</b>
	CovariantMap	—	0.74 (0.05)	0.47 (0.10)	0.73 (0.05)	0.71 (0.06)	0.41 (0.11)	0.70 (0.06)
		Shared (3)	0.76 (0.05)	0.51 (0.10)	0.75 (0.05)	0.73 (0.06)	0.46 (0.11)	0.73 (0.06)
		Dedicated (6)	0.76 (0.05)	0.52 (0.10)	0.76 (0.05)	0.74 (0.06)	0.47 (0.12)	0.73 (0.06)
MNIST-1D-PCA-8	<b>ZZFeatureMap</b>	—	0.95 (0.02)	0.90 (0.05)	0.95 (0.02)	<b>0.92 (0.03)</b>	<b>0.84 (0.06)</b>	<b>0.92 (0.03)</b>
		Shared (3)	0.95 (0.02)	0.90 (0.05)	0.95 (0.02)	<b>0.92 (0.03)</b>	<b>0.84 (0.06)</b>	<b>0.92 (0.03)</b>
		Dedicated (24)	<b>0.96 (0.02)</b>	<b>0.92 (0.04)</b>	<b>0.96 (0.02)</b>	<b>0.92 (0.03)</b>	<b>0.84 (0.06)</b>	<b>0.92 (0.03)</b>
	CovariantMap	—	0.79 (0.02)	0.57 (0.05)	0.78 (0.02)	0.80 (0.03)	0.60 (0.07)	0.80 (0.03)
		Shared (3)	0.81 (0.05)	0.61 (0.10)	0.80 (0.05)	0.81 (0.05)	0.63 (0.09)	0.81 (0.05)
		Dedicated (12)	0.80 (0.05)	0.61 (0.10)	0.80 (0.05)	0.81 (0.05)	0.63 (0.09)	0.81 (0.05)

Average (standard deviation) values for all  $n = 30$  repetitions are reported; ACC = accuracy,  $\kappa$  = Cohen's kappa, MF = Macro F1-score  
Best values for each metric are highlighted in bold. Best model is marked as the one maximizing testing accuracy.

Finally, following similar procedures, two additional 4-D and 8-D instance versions are arranged, using as reference the MNIST-1D dataset, a minimalist, low-memory, and low-compute alternative to the classical MNIST [26]. In a recent discussion posed by Bowles et al. [17], it was suggested that this dataset may be a much more suitable alternative to use for benchmarking of QML models.

Table 1 summarizes the characteristics of the datasets included in the resulting benchmark. In order to avoid the potential for bias and to facilitate subsequent statistical analysis, experimental procedures involve multiple repetitions with random resampling of the corresponding training and testing partitions for each dataset.

### 3) CLASSICAL BASELINE MODELS

To set a reference with respect to classification performance using quantum kernel methods, a set of classical machine learning counterparts are considered.

For this purpose, classical SVMs are taken with the use of linear, polynomial, and radial basis function (RBF) kernels to target the same classification benchmark. The author also includes a logistic regression (LR) classifier for additional reference. Both the logistic classifier and the linear kernel SVM operate in the original feature space and are limited to the generation of linear decision boundaries. Polynomial and RBF kernels are able to produce nonlinear decisions in the induced feature space, as described in Section II-A. For

**TABLE 3.** Results of Classical Methods

Dataset	Model	ACC <sub>TR</sub>	$\kappa_{TR}$	MF <sub>TR</sub>	ACC <sub>TS</sub>	$\kappa_{TS}$	MF <sub>TS</sub>
Ad-hoc-ZZ	LR	0.55 (0.03)	0.09 (0.06)	0.55 (0.03)	0.51 (0.04)	0.01 (0.09)	0.50 (0.04)
	SVM linear	0.55 (0.03)	0.10 (0.06)	0.55 (0.03)	0.51 (0.04)	0.02 (0.07)	0.51 (0.04)
	SVM poly (d=4)	0.57 (0.03)	0.13 (0.07)	0.52 (0.06)	0.54 (0.03)	0.08 (0.05)	0.49 (0.05)
	<b>SVM rbf</b>	<b>0.99 (0.02)</b>	<b>0.98 (0.05)</b>	<b>0.99 (0.02)</b>	<b>0.87 (0.04)</b>	<b>0.74 (0.09)</b>	<b>0.87 (0.04)</b>
Ad-hoc-COV	<b>LR</b>	<b>0.98 (0.02)</b>	<b>0.97 (0.04)</b>	<b>0.98 (0.02)</b>	<b>0.90 (0.04)</b>	<b>0.80 (0.09)</b>	<b>0.90 (0.04)</b>
	<b>SVM linear</b>	<b>0.98 (0.02)</b>	<b>0.97 (0.04)</b>	<b>0.98 (0.02)</b>	<b>0.90 (0.05)</b>	0.79 (0.11)	<b>0.90 (0.05)</b>
	SVM poly (d=17)	0.69 (0.08)	0.38 (0.16)	0.65 (0.11)	0.50 (0.00)	0.00 (0.00)	0.34 (0.01)
	SVM rbf	0.97 (0.07)	0.93 (0.13)	0.97 (0.07)	0.83 (0.07)	0.67 (0.15)	0.83 (0.08)
Linear-IRIS	<b>LR</b>	<b>1.00 (0.00)</b>					
	<b>SVM linear</b>	<b>1.00 (0.00)</b>					
	SVM poly (d=7)	0.99 (0.01)	0.97 (0.03)	0.99 (0.01)	0.97 (0.03)	0.94 (0.06)	0.97 (0.03)
	<b>SVM rbf</b>	<b>1.00 (0.00)</b>					
Non-linear-IRIS	<b>LR</b>	0.97 (0.02)	0.93 (0.04)	0.97 (0.02)	<b>0.94 (0.04)</b>	<b>0.88 (0.07)</b>	<b>0.94 (0.04)</b>
	<b>SVM linear</b>	0.97 (0.02)	0.94 (0.03)	0.97 (0.02)	<b>0.94 (0.03)</b>	<b>0.88 (0.06)</b>	<b>0.94 (0.03)</b>
	SVM poly (d=7)	0.84 (0.03)	0.69 (0.05)	0.84 (0.03)	0.81 (0.05)	0.62 (0.11)	0.80 (0.06)
	<b>SVM rbf</b>	<b>0.98 (0.02)</b>	<b>0.95 (0.04)</b>	<b>0.98 (0.02)</b>	<b>0.94 (0.04)</b>	<b>0.88 (0.08)</b>	<b>0.94 (0.04)</b>
MNIST-PCA-4	LR	0.78 (0.02)	0.56 (0.04)	0.78 (0.02)	0.78 (0.03)	0.56 (0.05)	0.78 (0.03)
	SVM linear	0.78 (0.02)	0.56 (0.03)	0.78 (0.02)	0.78 (0.03)	0.56 (0.06)	0.78 (0.03)
	SVM poly (d=7)	<b>0.87 (0.04)</b>	<b>0.73 (0.08)</b>	<b>0.87 (0.04)</b>	0.75 (0.04)	0.50 (0.08)	0.74 (0.04)
	<b>SVM rbf</b>	0.85 (0.03)	0.69 (0.06)	0.85 (0.03)	<b>0.82 (0.03)</b>	<b>0.65 (0.06)</b>	<b>0.82 (0.03)</b>
MNIST-PCA-8	LR	0.93 (0.02)	0.86 (0.03)	0.93 (0.02)	0.93 (0.02)	0.86 (0.03)	0.93 (0.02)
	SVM linear	0.93 (0.02)	0.87 (0.04)	0.93 (0.02)	0.92 (0.02)	0.85 (0.03)	0.92 (0.02)
	SVM poly (d=11)	0.86 (0.04)	0.72 (0.07)	0.86 (0.04)	0.77 (0.05)	0.53 (0.09)	0.75 (0.05)
	<b>SVM rbf</b>	<b>0.98 (0.01)</b>	<b>0.97 (0.03)</b>	<b>0.98 (0.01)</b>	<b>0.95 (0.01)</b>	<b>0.89 (0.03)</b>	<b>0.95 (0.01)</b>
MNIST-1D-PCA-4	LR	0.82 (0.02)	0.65 (0.04)	0.82 (0.02)	0.82 (0.03)	0.64 (0.07)	0.82 (0.03)
	SVM linear	0.83 (0.02)	0.65 (0.04)	0.83 (0.02)	0.82 (0.03)	0.65 (0.07)	0.82 (0.03)
	SVM poly (d=7)	<b>0.92 (0.03)</b>	<b>0.85 (0.05)</b>	<b>0.92 (0.03)</b>	0.74 (0.04)	0.49 (0.08)	0.73 (0.05)
	<b>SVM rbf</b>	0.89 (0.03)	0.79 (0.06)	0.89 (0.03)	<b>0.87 (0.04)</b>	<b>0.74 (0.08)</b>	<b>0.87 (0.04)</b>
MNIST-1D-PCA-8	LR	0.87 (0.02)	0.73 (0.03)	0.87 (0.02)	0.87 (0.03)	0.73 (0.07)	0.87 (0.03)
	SVM linear	0.87 (0.02)	0.73 (0.03)	0.87 (0.02)	0.87 (0.03)	0.74 (0.06)	0.87 (0.03)
	SVM poly (d=11)	<b>0.96 (0.01)</b>	<b>0.92 (0.02)</b>	<b>0.96 (0.01)</b>	0.70 (0.03)	0.41 (0.06)	0.68 (0.04)
	<b>SVM rbf</b>	<b>0.96 (0.03)</b>	<b>0.92 (0.06)</b>	<b>0.96 (0.03)</b>	<b>0.93 (0.03)</b>	<b>0.86 (0.05)</b>	<b>0.93 (0.03)</b>

Average (standard deviation) values for all  $n = 30$  repetitions are reported; ACC = accuracy,  $\kappa$  = Cohen's kappa, MF = Macro F1-score  
Best values for each metric are highlighted in bold. Best model is marked as the one maximizing testing accuracy.

polynomial kernels, it is well established that the dimensionality of the induced space is of order  $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ , where  $n$  is the original data dimension, and  $k$  is the degree of the polynomial. This result shall be considered when choosing the specific degree of the polynomial for each dataset. RBF-kernels can theoretically map data up to infinite-dimensional feature spaces. However, in practice, the effective dimension is constrained by the number of available training samples  $m$  [19].

### C. DATA PREPROCESSING AND OPTIMIZATION SETTINGS

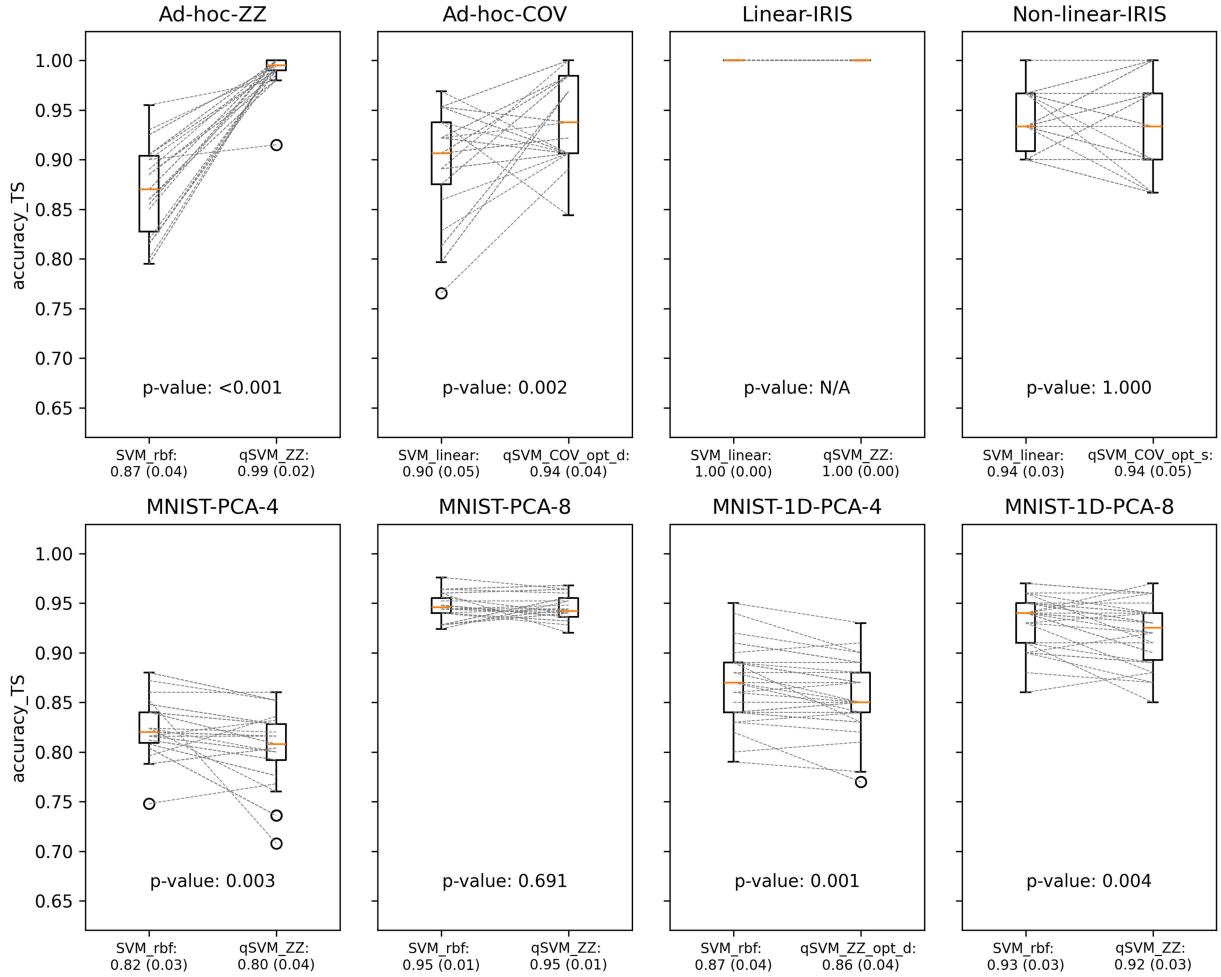
For the quantum models, the input features were scaled to the  $0 - 2\pi$  range for all datasets. This is a reasonable choice given that both quantum feature mappings tackled in this work implement angle encoding [27]. For classical models, the standard approach of removing the mean and scaling features to the unit variance (z-score normalization) [19] is followed. In any case, one should note that the scaling parameters in both cases are learned exclusively from the training set and subsequently applied to the test set, under the assumption that access to the test set during training is not permitted.

A hyperparameter search for all models and datasets is scheduled using a full-grid approach. For the classical methods, this includes, as a minimum, optimization of the regularization parameter  $C$  in the range  $[0.01, 0.1, 1, 10, 100]$ .

In addition, for the RBF kernel, different configurations for the bandwidth parameter  $\gamma \in [0.001, 0.001, 0.01, 1, 10]$  are also explored. For classical polynomial kernels, the degree of the polynomial  $k$  was set such that the dimensionality of the resulting induced space approximately matches that of the quantum kernels for the specific target dataset. The following rule-of-thumb is used: set  $k$  to the closest natural number such that  $\binom{n}{k} \leq 2^{2n}$ , i.e., the space of density matrices for a  $n$ -qubit system [cf., the mapping (6)].

In the case of the quantum kernels, the hyperparameter search includes the same logarithmic progression for  $C$  as for classical methods. In addition, the optimization of the (quantum) kernel bandwidth  $\lambda$  is also targeted, exploring the range  $[0.001, 0.01, 0.1, 0.5, 1.0]$ . As discussed in related work [28], [29], this parameter acts as a tradeoff between the generalization and expressiveness capabilities of the quantum kernel and can play a role in mitigating the effects of quantum kernel concentration [30], [31].

Exploration of the hyperparameter space grid was carried out in all cases following a fivefold cross-validation on the training set partition of the corresponding dataset. The accuracy score is used to select the best model. It is noted that whereas this metric can be misleading for some classification tasks, especially in the context of multiclass or unbalanced data, it is chosen here for its simplicity and clarity



**FIGURE 2.** Comparison of the best classical and quantum models per dataset among  $n = 30$  repetitions.

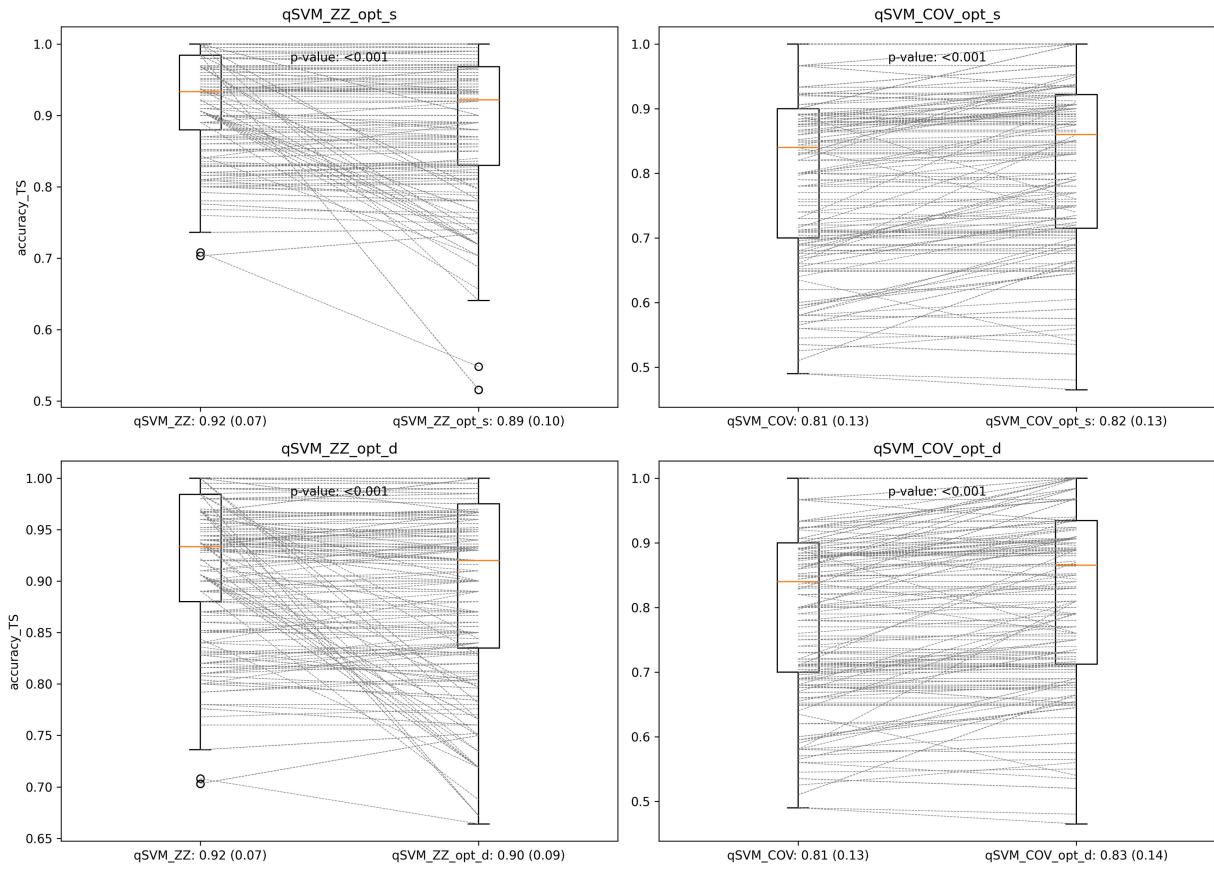
of interpretation, and since the datasets are all binary and balanced. Once the best hyperparameter set has been decided for each model, generalization performance is evaluated in the corresponding testing set. More generally, to quantify the performance of each investigated classifier, in addition to the accuracy score, Cohen's kappa ( $\kappa$ ) and macro F1-score were also calculated.

For QKT optimization, a maximum of 400 iterations of the associated SPSA hybrid algorithm described in the methods section are allowed. More specifically, a second-order version that includes the Hessian estimation of the loss is used for improved convergence. This version also resorts to automatic calibration to avoid uninformed fixing of the learning rate and of the perturbation parameter. As starting point, the parameter vector  $\vec{\theta}$  is initialized to all zeros. This point is chosen as it corresponds with the absence of effective rotation applied by means of  $V_\theta$  with respect to the  $|0^{\otimes n}\rangle$  state vector initialization. See (19) with regard to the associated QKE procedure. Once the maximum number of allowed iterations is reached,  $\theta_{\text{opt}}$  is finally set to the point with corresponding minimal loss (maximal weighted kernel alignment) obtained during the training phase.

All experiments were carried out in Python using Qiskit [32] (v1.1.0) in combination with scikit-learn [33] (v1.4.2.) packages. The latter was used for the implementation of the classical models, general hyperparameter tuning, metric evaluations, and to solve the SVM optimization problem. For QKT optimization, the SPSA implementation provided by the qiskit-algorithms (v0.3.0) extension package was used. The estimation of the quantum kernel matrices for QKE were carried out using a state vector simulation on a classical computer. For this purpose, the implementation relied on the functionality of the TrainableFidelityStatevectorKernel and QuantumKernelTrainer classes contained in the qiskit-machine-learning (v0.7.2) package.

#### IV. RESULTS

Experimental simulations for each model and dataset were repeated a total of  $n = 30$  times. For each repetition, the corresponding training and testing partitions were regenerated using random resampling following the distributions outlined in Table 1. This was in order to minimize potential biases and to enable subsequent statistical analyses of the results.



**FIGURE 3.** Differences in test accuracy score between baseline QKE and after the quantum kernel was optimized using QKT with weighted alignment. Performance values are aggregated across all datasets. P-values for paired t-tests between the corresponding metric samples are superimposed on each plot.

Table 2 gives the corresponding performance values obtained for the quantum models. In the table, the first column describes the dataset, the second column describes the applied quantum feature map, and the third column refers to the QKT parameterization strategy. Rows without an entry description under this category signify the absence of kernel training, indicating the implementation of “plain” QKE. Otherwise, either the text shared or dedicated is shown, cf., (20) and (21), respectively, together with the number of involved parameters. Notice that this number depends not only on the followed parameterization strategy but also on the circuit layout that results from the application of the associated quantum feature map. The subsequent columns show the corresponding averaged and standard deviation results for each of the evaluated metrics, obtained at training and testing times, respectively. Performance results obtained by the analyzed classical methods on each case are shown in Table 3.

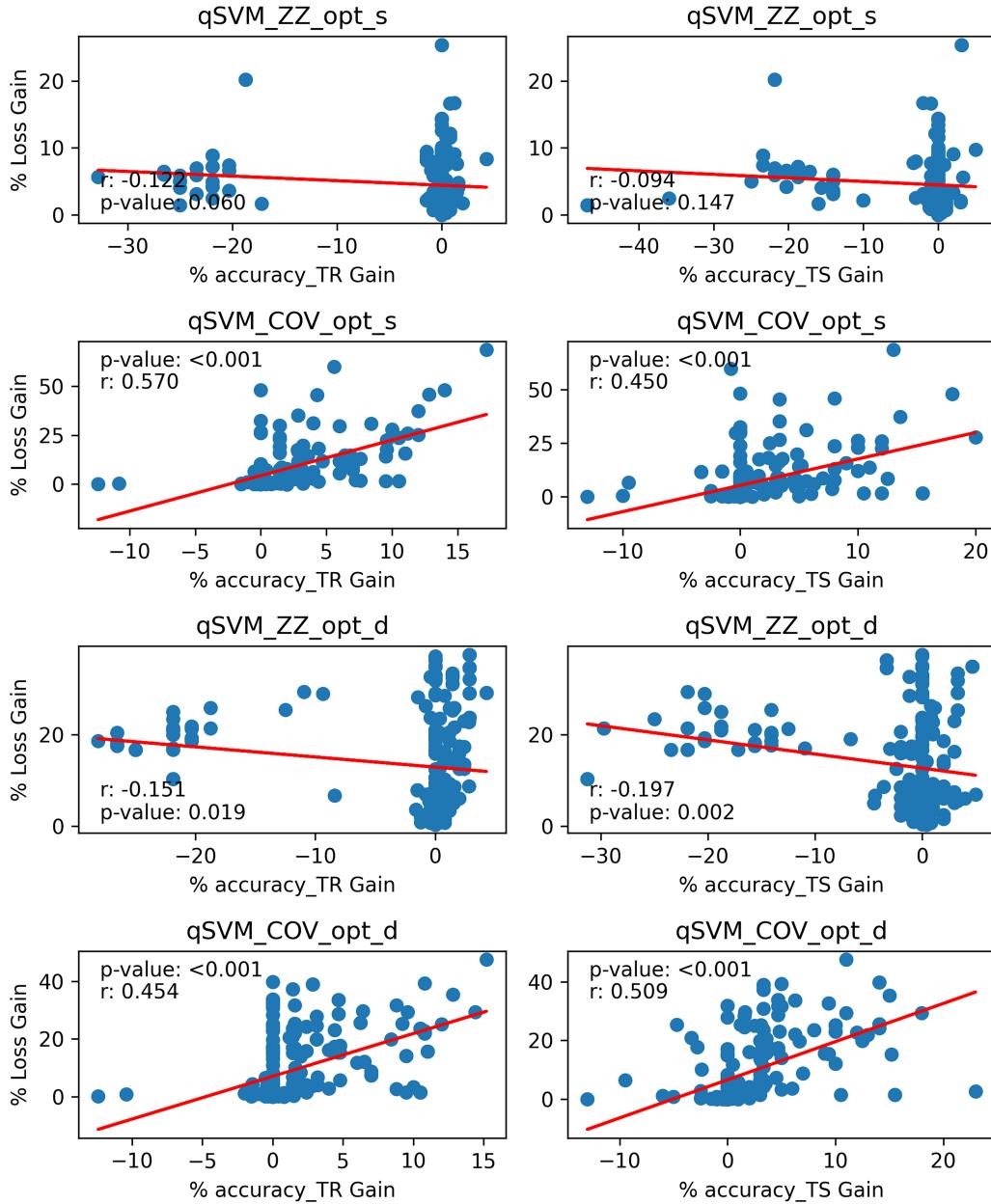
To analyze the statistical significance of the differences found across the models, paired t-tests were executed among the corresponding metrics distributions. The results are shown in Fig. 1, where the classification accuracy achieved in the testing set is taken as reference.

A more detailed comparison between the respective classical and quantum methods that perform best on each

dataset is shown in Fig. 2. It can be seen that quantum methods outperform classical models in the subset of ad hoc datasets, but not in the general case of classical problems. In some cases (Linear-IRIS, Nonlinear-IRIS, and MNIST-PCA-8), quantum methods are able to produce results that are statistically indistinguishable from classical ones. It is also noted that both approaches are capable of achieving perfect class separation in the Linear-IRIS. However, in others (MNIST-PCA-4, MNIST-1D-PCA-4, and MNIST-PCA-1D-8), a significant slight degradation in performance is observed.

The experimental data suggest an overall better performance of quantum methods based on variants of the ZZ-mapping over the Covariant alternative. Apart from the otherwise logical trend observed in the respective ad hoc datasets, variants of the ZZ-mapping achieve the best performance in four out of the six classical datasets. However, in the two exceptions (Linear-IRIS and Nonlinear-IRIS), it follows from Fig. 1 that the differences between the two quantum mappings do not reach the level of statistical significance.

To provide a general evaluation of the contribution of the weighted alignment QKT with respect to baseline QKE, as well as to analyze the influence of the associated parameterization strategy, aggregation of the



**FIGURE 4.** Relationship between evolution of the target loss and resulting accuracy score.

corresponding performance metrics is considered across all datasets. The results are shown in Fig. 3. An opposite trend can be observed depending on the specific quantum mapping. Whereas for the Covariant mapping, QKT seems to contribute to an overall improvement, regardless of the parameterization strategy, a downgrading effect is observed in the case of the ZZ feature map. The statistical significance of the results at the dataset levels shows mixed behavior.

One can further analyze the relationship between the minimization of the target loss during the training phase and the resulting performance gain. The results in terms of the respective training and testing classification accuracies are shown in Fig. 4. The data confirm the trends of Fig. 3,

observing that while the achieved improvement in the target loss goes up to 50% in some cases, it does not always translate into an effective increase in the resulting classification performance. Furthermore, a general downgrading effect can be observed for the ZZ feature map, and even in some specific cases when using Covariant kernels (negative values on the  $x$ -axis). Nevertheless, general positive correlation between the weighted target alignment and the classification performance is observed for the latter.

## V. DISCUSSION AND CONCLUSION

Following the results of the experiments, one initial observation is that there is no one-fits-to-all method; the best approach in terms of classification performance,

whether quantum or classical (with their corresponding variants) depends very much on the specific targeted dataset. Perhaps unsurprisingly, quantum methods outperformed classical alternatives in the Ad-hoc-ZZ and Ad-hoc-COV datasets. More specifically, the optimal points for these datasets were achieved when using the corresponding quantum generative mappings, ZZFeatureMap and CovariantMap, respectively. This is consistent with the expected behavior predicted in the original works [2], [18]. However, these works did not use classical methods to compare performance on the corresponding ad hoc data. Thus, the results confirm these hypotheses with experiments. The quantum suitability nevertheless vanishes for the subset of classical datasets. This does not imply that quantum methods were ineffective or unable to learn. With some exceptions (MNIST-PCA-4, MNIST-1D-PCA-4, and MNIST-1D-PCA-8), they achieved results that were statistically indistinguishable from the best classical alternatives. And even then, the average performance scores were not far off from those of the classical methods.

In terms of the quantum mapping, the data suggest better overall performance in the case of the ZZ-mapping variant compared to the Covariant counterparts. However, it is important to recall that in order to extend the use of the Covariant mapping beyond the specific LCE problem, the actual optimal coupling for the Ad-hoc-COV data was replaced by a more general linear entanglement layer. This modification comes with the cost of preventing the mapping from achieving perfect classification in the corresponding Ad-hoc-COV dataset. The extent to which the differences in entanglement levels between the two mappings may have conditioned their overall performance on the classical datasets remains a topic for further research.

Perhaps relatedly, when analyzing the effect of QKT over baseline QKE, opposite trends are observed between the two quantum mappings. Whereas QKT seemed to contribute to an overall slight improvement for the Covariant mapping, a downgrading effect was observed for the ZZ feature maps. One could speculate on whether part of this trend could be explained by the fact that the room for improvement with respect to baseline QKE was more restricted for the ZZFeatureMap. However, the data in Fig. 4 seem to discard possible overfitting effects.

As for the influence of the parameterization strategy, the data show that the dedicated approach always achieves equal or better performance than the shared strategy. Although these differences do not always reach the level of statistical significance, it is a logical result for which the space of shared solutions is an obvious subset of the dedicated parameter vectors. That is, where all dedicated rotations share exactly the same parameter values.

Altogether, the results of the benchmarking call into question a generalized added value of applying QKT optimization. As discussed later, the additional computational cost associated with QKT optimization is significant, and without

a clear benefit in terms of performance gain, it is difficult to justify its use.

In fact, efforts toward optimization of the kernel hyperparameters may pay off more effectively. As a reference, one can compare the results of the present study with preliminary work that relied on fixed default parameterization [34], noting significant changes in the simulation results that boost the resulting performance of both classical and quantum approaches. Notably, in addition to the regularization parameter  $C$ , this study included the quantum kernel bandwidth in the hyperparameter search. As discussed in related work [28], [29], this parameter acts as a tradeoff between the generalization and expressiveness capabilities of the quantum kernel and can play a role in mitigating the effects of quantum kernel concentration. To the best of the author's knowledge, this is the first study to consider this parameterization in conjunction with QKT. Although an explicit analysis of the relationship between quantum kernel alignment and optimal bandwidth is beyond the scope of this work, it motivates further studies to deeply understand the underlying implications of this interplay.

From a general point of view, it is also important to consider that quantum feature maps can be constructed with considerable freedom regarding their architecture. In this work, the ZZFeatureMap and the CovariantMap have been chosen as a basis for which their conjectured association with quantum advantage. Whereas the study's primary focus targeted quantum kernel optimization, rather than kernel selection [1], one still has to address the problem of selecting the kernel family (e.g., the structure of the parameterized circuit layout). One should expect this choice to largely influence the foreseeable performance on the targeted dataset, potentially more so than by QKT parameter optimization.

From a nonfunctional point of view, when comparing classical versus quantum kernel methods, one should also consider important aspects that go beyond the mere performance metrics. These include issues related to their scalability, efficiency, robustness, or practical usability. In this regard, kernel methods (including classical ones) need to store and compute the kernel matrix, which grows quadratically with the number of training samples. Hence, these methods become computationally expensive as the dataset size increases. In fact, in the current era of Big Data, kernel methods have been superseded by (deep) neural networks, which usually scale linearly on the number of training samples. This is without prejudice to other advantages that kernel-based methods bring to (still challenging) problems of a more moderated size. These include improved efficiency, interpretability, or convergence guarantees, partly derived from the well-defined mathematical framework supporting these methods.

However, in the case of quantum kernels, the scalability aspect becomes even more crucial. Specifically, current practical implementation of the quantum fidelity estimation procedure requires multiple repetitions of the

compute–uncompute circuit with increased cost penalty over classical counterparts [18]. Moreover, variational QKT optimization involves additional computational overhead over QKE, for which each iteration of the hybrid algorithm requires recomputation of the kernel matrix with the new set of candidate parameters. In addition, one should take into account current NISQ-era hardware limitations, which make it necessary to increase the number of sampling shots to compensate for the noise effects. Recent work has also discussed the related problem of exponential concentration, whereby values of the kernel matrix tend to concentrate around constant values, and for which precision of the related estimated entries needs to be increased to allow learning to occur [30], [31].

Whether these issues may potentially invalidate the theoretical exponential speedups attributed to the use of quantum kernels is still under debate. Recent work has explored alternative approaches, such as the use of projected kernels [14] or the optimization of the number of shots [31]. Likewise, it has been suggested that kernel concentration effects may strongly depend on the characteristics of the targeted dataset [35], and various links have been established with respect to the depth and expressivity of the associated quantum circuits [36], [37]. These topics deserve more attention and are currently among the most active areas of research in the QML field.

Quantum kernel methods can still offer better scaling than alternative QML variational approaches, in addition to the already mentioned guarantees in accessing true global optima [3]. In the longer term, if reliability of quantum hardware progresses enough (and while improving, this is still a big if), alternative implementations of quantum vector machines may become accessible, for which it is known that training time can grow linearly with the size of the data [38].

In future work, the authors is committed to extend the current benchmark to include additional optimization strategies and benchmark datasets. In the first case, this includes comparing weighted versus unweighted alignment, multi-class and unbalanced problems, and exploring the suitability of additional quantum feature maps. The author also anticipates the need to investigate the case for real-world and domain-specific applications that involve potentially more challenging data.

## DATA AVAILABILITY

Data and source code for the reproducibility of experimental procedures in this study are available at [https://github.com/diegoalvarez/qkt\\_benchmarking](https://github.com/diegoalvarez/qkt_benchmarking).

## REFERENCES

- [1] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. S. Derkx, P. K. Faehrmann, and J. J. Meyer, “Training quantum embedding kernels on near-term quantum computers,” *Phys. Rev. A*, vol. 106, no. 4, Oct. 2022, Art. no. 042431, doi: [10.1103/PhysRevA.106.042431](https://doi.org/10.1103/PhysRevA.106.042431).
- [2] J. R. Glick et al., “Covariant quantum kernels for data with group structure,” *Nature Phys.*, vol. 20, no. 3, pp. 479–483, Mar. 2024, doi: [10.1038/s41567-023-02340-9](https://doi.org/10.1038/s41567-023-02340-9).
- [3] M. Schuld, “Supervised quantum machine learning models are kernel methods,” 2021, *arXiv:2101.11020*, doi: [10.48550/arXiv.2101.11020](https://doi.org/10.48550/arXiv.2101.11020).
- [4] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” 2020, *arXiv:2010.02174*, doi: [10.48550/arXiv.2010.02174](https://doi.org/10.48550/arXiv.2010.02174).
- [5] S. Miyabe et al., “Quantum multiple kernel learning in financial classification tasks,” 2023, *arXiv:2312.00260*, doi: [10.48550/arXiv.2312.00260](https://doi.org/10.48550/arXiv.2312.00260).
- [6] M. Grossi et al., “Mixed quantum–classical method for fraud detection with quantum feature selection,” *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 3102812, doi: [10.1109/TQE.2022.3213474](https://doi.org/10.1109/TQE.2022.3213474).
- [7] A. Di Pierro and M. Incudini, “Quantum machine learning and fraud detection,” in *Protocols, Strands, and Logic*. Cham, Switzerland: Springer, 2021, pp. 139–155, doi: [10.1007/978-3-030-91631-2\\_8](https://doi.org/10.1007/978-3-030-91631-2_8).
- [8] Z. Krunic, F. Flotter, G. Seegan, N. Earnest-Noble, and S. Omar, “Quantum kernels for real-world predictions based on electronic health records,” *IEEE Trans. Quantum Eng.*, vol. 3, 2022, Art. no. 2500311, doi: [10.1109/TQE.2022.3176806](https://doi.org/10.1109/TQE.2022.3176806).
- [9] A. Miroszewski et al., “Detecting clouds in multispectral satellite images using quantum-kernel support vector machines,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 7601–7613, 2023, doi: [10.1109/JSTARS.2023.3304122](https://doi.org/10.1109/JSTARS.2023.3304122).
- [10] A. M. Wijata, A. Miroszewski, B. L. Saux, N. Longépé, B. Ruszczak, and J. Nalepa, “Detection of bare soil in hyperspectral images using quantum-kernel support vector machines,” in *Proc. IGARSS 2024-2024 IEEE Int. Geosci. Remote Sens. Symp.*, 2024, pp. 817–822, doi: [10.1109/IGARSS53475.2024.10641442](https://doi.org/10.1109/IGARSS53475.2024.10641442).
- [11] A. Miroszewski et al., “Cloud detection in multispectral satellite images using support vector machines with quantum kernels,” in *Proc. IGARSS 2023-2023 IEEE Int. Geosci. Remote Sens. Symp.*, 2023, pp. 796–799, doi: [10.1109/IGARSS52108.2023.10282416](https://doi.org/10.1109/IGARSS52108.2023.10282416).
- [12] S. L. Wu et al., “Application of quantum machine learning to HEP analysis at LHC using quantum computer simulators and quantum computer hardware,” *Eur. Phys. Soc. Conf. High Energy Phys.*, vol. EPS-HEP2021, p. 842, 2022, doi: [10.22323/1.398.0842](https://doi.org/10.22323/1.398.0842).
- [13] E. Peters et al., “Machine learning of high dimensional data on a noisy quantum processor,” *npj Quantum Inf.*, vol. 7, 2021, Art. no. 161, 10.1038/s41534-021-00498-9. [Online]. Available: <https://api.semanticscholar.org/CorpusID:231698587>
- [14] H.-Y. Huang et al., “Power of data in quantum machine learning,” *Nature Commun.*, vol. 12, no. 1, May 2021, Art. no. 2631, doi: [10.1038/s41467-021-22539-9](https://doi.org/10.1038/s41467-021-22539-9).
- [15] R. Mengoni and A. D. Pierro, “Kernel methods in quantum machine learning,” *Quantum Mach. Intell.*, vol. 1, pp. 65–71, 2019, doi: [10.1007/s42484-019-00007-4](https://doi.org/10.1007/s42484-019-00007-4).
- [16] F. D. Marcantonio, M. Incudini, D. Tezza, and M. Grossi, “Quantum advantage seeker with kernels (QuASK): A software framework to speed up the research in quantum machine learning,” *Quantum Mach. Intell.*, vol. 5, pp. 1–11, 2022, doi: [10.1007/s42484-023-00107-2](https://doi.org/10.1007/s42484-023-00107-2).
- [17] J. Bowles, S. Ahmed, and M. Schuld, “Better than classical? The subtle art of benchmarking quantum machine learning models,” Mar. 2024, *arXiv:2403.07059*, doi: [10.48550/arXiv.2403.07059](https://doi.org/10.48550/arXiv.2403.07059).
- [18] V. Havlíček et al., “Supervised learning with quantum-enhanced feature spaces,” *Nature*, vol. 567, no. 7747, pp. 209–212, Mar. 2019, doi: [10.1038/s41586-019-0980-2](https://doi.org/10.1038/s41586-019-0980-2).
- [19] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006, doi: [10.5555/1162264](https://doi.org/10.5555/1162264).
- [20] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Phys. Rev. Lett.*, vol. 122, Feb. 2019, Art. no. 040504, doi: [10.1103/PhysRevLett.122.040504](https://doi.org/10.1103/PhysRevLett.122.040504).
- [21] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, “On kernel-target alignment,” in *Advances in Neural Information Processing Systems*, T. Dietterich, S. Becker, and Z. Ghahramani, Eds., vol. 14. Cambridge, MA, USA: MIT Press, 2001. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2001/file/f71e393b3809197ed66df836fe833e5-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2001/file/f71e393b3809197ed66df836fe833e5-Paper.pdf)
- [22] T. Wang, D. Zhao, and S. Tian, “An overview of kernel alignment and its applications,” *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 179–192, Feb. 2015, doi: [10.1007/s10462-012-9369-4](https://doi.org/10.1007/s10462-012-9369-4).

- [23] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, “Quantum embeddings for machine learning,” 2020, *arXiv.2001.03622*, doi: [10.48550/arXiv.2001.03622](https://doi.org/10.48550/arXiv.2001.03622).
- [24] M. J. Bremner, A. Montanaro, and D. J. Shepherd, “Average-case complexity versus approximate simulation of commuting quantum computations,” *Phys. Rev. Lett.*, vol. 117, Aug. 2016, Art. no. 080501, doi: [10.1103/PhysRevLett.117.080501](https://doi.org/10.1103/PhysRevLett.117.080501).
- [25] B. Fuller, J. R. Glick, and C. Johnson, “Quantum kernel training toolkit,” 2021, Accessed: Aug. 4, 2024. [Online]. Available: <https://github.com/qiskit-community/prototype-quantum-kernel-training>
- [26] S. Greydanus and D. Kobak, “Scaling down deep learning with MNIST-ID,” in *Proc. 41st Int. Conf. Mach. Learn.*, vol. 235, 2024, pp. 16404–16415, Art. no. 653, doi: [10.48550/arXiv.2011.14439](https://doi.org/10.48550/arXiv.2011.14439).
- [27] M. Schuld and F. Petruccione, *Machine Learning With Quantum Computers*. Cham, Switzerland: Springer, 2021, doi: [10.1007/978-3-030-83098-4](https://doi.org/10.1007/978-3-030-83098-4).
- [28] R. Shaydulin and S. M. Wild, “Importance of kernel bandwidth in quantum machine learning,” *Phys. Rev. A*, vol. 106, no. 4, Oct. 2022, Art. no. 042407, doi: [10.1103/PhysRevA.106.042407](https://doi.org/10.1103/PhysRevA.106.042407).
- [29] A. Canatar, E. Peters, C. Pehlevan, S. M. Wild, and R. Shaydulin, “Bandwidth enables generalization in quantum kernel models,” *Trans. Mach. Learn. Res.*, 2023, doi: [10.48550/arXiv.2206.06686](https://doi.org/10.48550/arXiv.2206.06686).
- [30] S. Thanasilp, S. Wang, and M. Cerezo, “Exponential concentration in quantum kernel methods,” *Nat. Commun.*, vol. 15, p. 5200, 2024, doi: [10.1038/s41467-024-49287-w](https://doi.org/10.1038/s41467-024-49287-w).
- [31] A. Miroszewski, M. F. Asiani, J. Mielczarek, B. L. Saux, and J. Nalepa, “In search of quantum advantage: Estimating the number of shots in quantum kernel methods,” 2024, *arXiv.2407.15776*, doi: [10.48550/arXiv.2407.15776](https://doi.org/10.48550/arXiv.2407.15776).
- [32] A. Javadi-Abhari et al., “Quantum computing with qiskit,” 2024, *arXiv:2405.08810*, doi: [10.48550/arXiv.2405.08810](https://doi.org/10.48550/arXiv.2405.08810).
- [33] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011, doi: [10.5555/1953048.2078195](https://doi.org/10.5555/1953048.2078195).
- [34] D. Alvarez-Estevez, “Benchmarking quantum machine learning kernel training for classification tasks,” 2024, *arXiv:2408.10274v1*. [Online]. Available: <https://arxiv.org/abs/2408.10274v1>
- [35] A. Miroszewski, B. L. Saux, N. Longépé, and J. Nalepa, “Utility of quantum kernel machines in remote sensing applications,” in *Proc. IGARSS 2024-2024 IEEE Int. Geosci. Remote Sens. Symp.*, 2024, pp. 799–803, doi: [10.1109/IGARSS53475.2024.10640823](https://doi.org/10.1109/IGARSS53475.2024.10640823).
- [36] E. Peters and M. Schuld, “Generalization despite overfitting in quantum machine learning models,” *Quantum*, vol. 7, Dec. 2023, Art. no. 1210, doi: [10.22331/q-2023-12-20-1210](https://doi.org/10.22331/q-2023-12-20-1210).
- [37] E. Gil-Fuster, J. Eisert, and V. Dunjko, “On the expressivity of embedding quantum kernels,” *Mach. Learn.: Sci. Technol.*, vol. 5, no. 2, Apr. 2024, Art. no. 025003, doi: [10.1088/2632-2153/ad2f51](https://doi.org/10.1088/2632-2153/ad2f51).
- [38] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for Big Data classification,” *Phys. Rev. Lett.*, vol. 113, no. 13, Sep. 2014, Art. no. 130503, doi: [10.1103/PhysRevLett.113.130503](https://doi.org/10.1103/PhysRevLett.113.130503).