This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/TQE.2020.DOI

Quantum Conformal Prediction for Reliable Uncertainty Quantification in Quantum Machine Learning

SANGWOO PARK, (Member, IEEE), and OSVALDO SIMEONE, (Fellow, IEEE)

King's Communications, Learning & Information Processing (KCLIP) lab, Centre for Intelligent Information Processing Systems (CIIPS), Department of Engineering, King's College London, London WC2R 2LS, United Kingdom (email: {sangwoo.park, osvaldo.simeone}@kcl.ac.uk)

Corresponding author: Sangwoo Park (email: sangwoo.park@kcl.ac.uk).

"This work was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Research and Innovation Programme (grant agreement No. 725732), by the European Union's Horizon Europe project CENTRIC (101096379), by an Open Fellowship of the EPSRC (EP/W024101/1), by the EPSRC project (EP/X011852/1), and by Project REASON, a UK Government funded project under the Future Open Networks Research Challenge (FONRC) sponsored by the Department of Science Innovation and Technology (DSIT)."

ABSTRACT Quantum machine learning is a promising programming paradigm for the optimization of quantum algorithms in the current era of noisy intermediate scale quantum (NISQ) computers. A fundamental challenge in quantum machine learning is generalization, as the designer targets performance under testing conditions, while having access only to limited training data. Existing generalization analyses, while identifying important general trends and scaling laws, cannot be used to assign reliable and informative "error bars" to the decisions made by quantum models. In this article, we propose a general methodology that can reliably quantify the uncertainty of quantum models, irrespective of the amount of training data, of the number of shots, of the ansatz, of the training algorithm, and of the presence of quantum hardware noise. The approach, which builds on probabilistic conformal prediction, turns an arbitrary, possibly small, number of shots from a pre-trained quantum model into a set prediction, e.g., an interval, that *provably* contains the true target with any desired coverage level. Experimental results confirm the theoretical calibration guarantees of the proposed framework, referred to as *quantum conformal prediction*.

Code can be found at https://github.com/kclip/quantum-CP

INDEX TERMS Quantum machine learning, conformal prediction, generalization analysis, uncertainty quantification.

I. INTRODUCTION

QUANTUM machine learning is currently viewed as a promising paradigm for the optimization of algorithms that can leverage existing noisy intermediate scale quantum (NISQ) computers [1]–[3]. As for classical machine learning, the principle underlying quantum machine learning is *generalization*: While the optimization is based on a limited training set, the designer targets the performance of the algorithm *outside* the training set, under testing conditions that are statistically similar – ideally equivalent – to those encountered during training data collection. The central technical challenge is therefore that of understanding and controlling accuracy and uncertainty levels for decisions taken by a trained algorithm on test data [4], [5].

As we will summarize in the next subsection, most work on the subject of generalization for quantum machine learning has focused on studying scaling laws on the amount of data required to ensure desired performance levels on test data (see, e.g., [6]). In contrast, this article introduces a novel *operational* framework for the derivation of practically relevant generalization guarantees. The approach, referred to as *quantum conformal prediction* (QCP), applies a post-hoc calibration step based on held-out calibration, or validation, data to pre-trained quantum machine learning models. QCP builds on *conformal prediction* (CP), a general calibration methodology that is currently experiencing renewed attention in the area of classical machine learning [7]–[10]. Unlike classical CP, QCP takes into account the unique nature of quantum models as *probabilistic* machines, due to the inherent randomness of quantum measurements [3], [11].

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

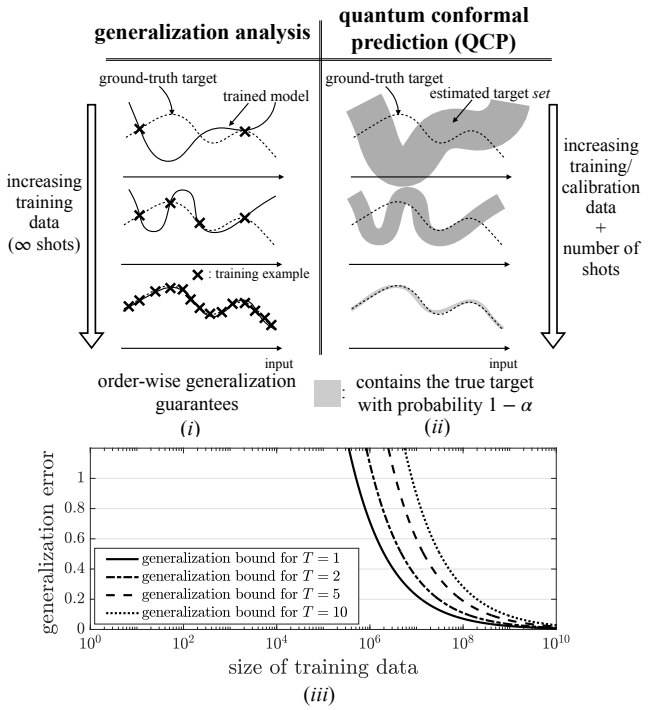


FIGURE 1. Comparison between (*i*) quantum generalization analysis [6], [12]–[15] and (*ii*) quantum conformal prediction (QCP), which is introduced in this work. Quantum generalization analysis provides *analytical* bounds on the generalization error that explicitly capture the general dependence on the number of training examples (typically assuming an infinite number of shots). Based on such bounds, one is able to conclude that, if the number of data points scales sufficiently quickly with respect to the model complexity, the trained model generalizes well outside the training data. In contrast, QCP provides an *operational* way of quantifying the uncertainty of the decisions made outside the training set (shaded areas). The resulting "error bars" are *guaranteed* to contain the ground-truth output with a desired probability, regardless of the amount of data, of the number of measurements, of the ansatz of the quantum machine learning model, of the training algorithm, and of the presence of quantum hardware noise. (*iii*) As an illustration of the results that can be obtained via generalization analysis, this panel shows the generalization bounds derived in [6] as a function of number of training examples for different numbers T of trainable local quantum gates (see Appendix A for details). As suggested by the plot, while very useful to identify general trends and scaling laws, generalization analyses only provide numerically meaningful bounds with a very large number of training examples.

A. GENERALIZATION ANALYSIS VS. CONFORMAL PREDICTION

As illustrated in Fig. 1, *generalization analysis* focuses on the identification of *analytical* scaling laws on the amount of data required to ensure desired performance levels on test data [4], possibly as a function of the training algorithm [16], [17] and of the data distribution [18], [19] (see also [5]). Related studies have also been initiated for quantum machine learning, with recent results including [6], [12]–[15].

As a notable example, reference [6] reveals the important insight that the *generalization error*, i.e., the discrepancy between training and test losses, for quantum models grows as the square root of the number of gates T and with the inverse of the square root of the size of the data set. While critical to gauge the feasibility to train quantum circuits, such results provide limited *operational* guidelines concerning the uncertainty associated with decisions made on test data when training data are limited (see Fig. 1 and Appendix A for details).

Most papers on quantum machine learning, including on generalization analyses, treat the output of a quantum model as *deterministic*, implicitly assuming that expected values of observables can be calculated exactly [6], [12]–[15]. In practice, for this to be an accurate modelling assumption, one needs to carry out a sufficiently large – strictly speaking, infinite – number of measurements, or *shots*, at the output of the quantum circuit (see Fig. 1, left column). These measurements are averaged to obtain the final prediction. From a statistical viewpoint, this assumption conveniently makes a quantum models *likelihood-based*, in the sense that one can evaluate exactly the probability of any output given an input and the model parameters (see, e.g., [5]).

In practice, however, quantum models are more properly described as being *implicit*, or *simulation-based*, in the sense that they only provide random samples from a given, inaccessible, likelihood [20]. The generalization capabilities of *generative* quantum models – an example of implicit models – have been recently studied in a separate line of research, including in [21], [22].

In contrast to generalization analysis, CP [7] does not aim at obtaining analytical conclusions concerning sample efficiency. Rather, it provides a general methodology to obtain "error bars" with formal guarantees on the probability of covering the correct test output. Such guarantees hold irrespective of the size of the training data set [7], [9]. We will specifically focus here on *split*, or *validation-based*, CP [7].

Validation-based CP is a post-hoc *calibration* methodology that applies to *pre-trained* machine learning models. It produces set predictors, such as intervals for regression problems, that are provably guaranteed to contain the true target with probability no smaller than the predetermined coverage level (see Fig. 1, right column). This goal is accomplished by utilizing a *calibration data set*, and is formalized in terms of probabilities with respect to the random generation of testing and calibration data.

To the best of our knowledge, as we further discuss next, this is the first work that proposes to leverage CP for the purpose of reliably quantifying uncertainty for quantum machine learning models.

B. QUANTUM CONFORMAL PREDICTION

The goal of the proposed framework, referred to as QCP, is to assign *well-calibrated* "error bars" to the decisions made by the quantum machine learning models. The "error-bars" are formally subsets of the output space, and *calibration* refers to the property of containing the true target with probability no smaller than a predetermined *coverage* level. Importantly, with QCP, calibration holds irrespective of the number of training data set, of the ansatz of the quantum machine learning model, of the training algorithm, of the number of shots, and of the presence of quantum hardware

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

noise [11]. The only assumption required for QCP is the *finite exchangeability* of calibration data and test data, which is practically satisfied under the conventional independent and identically distributed (i.i.d.) assumption [7]. We refer to Fig. 2, left bar, for an illustration of the properties of QCP.

An important challenge in applying CP to quantum machine learning models is that conventional CP targets *deterministic*, likelihood-based, models that produce point predictions, such as conventional neural networks, while quantum models are inherently *stochastic*, or sampling-based. Recent work [23] introduced a novel CP scheme, referred to as *probabilistic CP* (PCP), that applies to *classical* sampling-based predictors. The main motivation of reference [23] in leveraging probabilistic outputs is to address a mismatch between the assumed probabilistic model and the ground-truth distribution, which may require disconnected predictive sets (see [23, Fig. 1], as well as Fig. 3 for a preview). QCP is inspired by reference [23], although our motivation stems from the fact that quantum models are *inherently* implicit probabilistic models. The proposed method is aligned with the framework in [24] in that it operates with any finite number of copies of a quantum state, not requiring the exact evaluation of an expected value.

Our main contributions are as follows:

- We introduce QCP, a post-hoc calibration methodology for quantum machine learning models that produces predictive sets with controllable guarantees on the generalization performance.
- To keep the article self-contained, we provide an introduction to classical CP and PCP.
- We present, for reference, a direct application of CP to conventional deterministic and likelihood-based quantum machine learning that assume the use of an infinite number of shots.
- We detail experimental results based on both simulation and quantum hardware implementation. The experiments bring evidence on the merits of QCP, confirming its theoretical calibration guarantees.
- Among our main conclusions, we show that, when quantum models are augmented with QCP, it is generally advantageous not to average over the shots, as typically done in the literature. Rather, treating the shots as separate samples allows QCP to obtain more informative error bars.

The rest of the article is organized as follows. In Sec. II, conventional CP for classical deterministic models is reviewed, and a direct application of CP to quantum machine learning models is presented that assumes an infinite number of shots. Sec. III describes PCP for classical probabilistic models as proposed in [23]. Then, QCP for quantum machine learning models is introduced in Sec. IV, with Sec. V addressing the aspect of performance evaluation. Numerical and experimental settings are provided in Sec. VI, with results presented in Sec. VII. Sec. VIII concludes the article.

II. CONFORMAL PREDICTION FOR DETERMINISTIC MODELS

In this section, we first review conventional CP, which applies a post-hoc calibration step to *deterministic* predictive models [7]. Then, we introduce a direct application of this approach to quantum models, and we highlight its limitations. This discussion will motivate the proposed QCP method, which is presented in Sec. IV.

A. CALIBRATING DETERMINISTIC PREDICTORS VIA SET PREDICTION

Consider a class of classical parametric predictors $f(\cdot|\theta)$ with parameter vector θ. The predictor produces an output $y \in \mathcal{Y}$ given an input $x \in \mathcal{X}$ as

$$\hat{y} = f(x|\theta). \tag{1}$$

The domains $\mathcal{X}$ and $\mathcal{Y}$ of input and output, respectively, can be either discrete or continuous depending on the problem of interest. The parameter vector θ is optimized using a *training data set*

$$\mathcal{D}^{\text{tr}} = \{z^{\text{tr}}[i] = (x^{\text{tr}}[i], y^{\text{tr}}[i])\}_{i=1}^{|\mathcal{D}^{\text{tr}}|} \tag{2}$$

via an arbitrary training algorithm that returns a parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ and a corresponding predictor $\hat{y} = f(x|\theta_{\mathcal{D}^{\text{tr}}})$.

As we detail in the next subsection, CP is a post-hoc calibration scheme that processes the prediction $\hat{y}$ by using a *calibration data set*

$$\mathcal{D}^{\text{cal}} = \{z[i] = (x[i], y[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|} \tag{3}$$

to obtain a *well-calibrated* set prediction. Given a test input x, a *set predictor* $\Gamma(x|\mathcal{D}^{\text{cal}}, \theta) \subseteq \mathcal{Y}$ returns a subset of the possible output values in set $\mathcal{Y}$. The set $\Gamma(x|\mathcal{D}^{\text{cal}}, \theta)$ is said to be *well calibrated* if it contains the true target y with probability no smaller than a predetermined *coverage level* $1 - \alpha$.

To formalize the notion of calibration for set predictors, let us make the mild assumption that the calibration examples $z[i] = (x[i], y[i])$ for $i = 1, ..., |\mathcal{D}^{\text{cal}}|$ in set $\mathcal{D}^{\text{cal}}$, along with the test example $z = (x, y)$, are (finitely) exchangeable random variables. As a point of notation, observe that we use bold fonts to denote random variables.

Assumption 1 (Finite exchangeability [25, Sec. II])**.** *Calibration data set $\boldsymbol{\mathcal{D}}^{cal}$ and a test data point $\mathbf{z}$ are finitely exchangeable random variables, i.e., the joint distribution $p(\mathcal{D}^{cal}, z) = p(z[1], ..., z[|\mathcal{D}^{cal}|], z)$ is invariant to any permutation of the variables $\{\mathbf{z}[1], ..., \mathbf{z}[|\mathcal{D}^{cal}|], \mathbf{z}\}$. Mathematically, we have the equality $p(z[1], ..., z[|\mathcal{D}^{cal}| + 1]) = p(z[\pi(1)], ..., z[\pi(|\mathcal{D}^{cal}|+1)])$ with $z = z[|\mathcal{D}^{cal}|+1]$, for any permutation operator $\pi(\cdot)$. Note that the standard assumption of i.i.d. random variables satisfies finite exchangeability.*

It is important to emphasize that only exchangeability of calibration and test data is assumed (Assumption 1), while no requirements are imposed on the training data set $\mathcal{D}^{\text{tr}}$, or on the accuracy of the trained model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

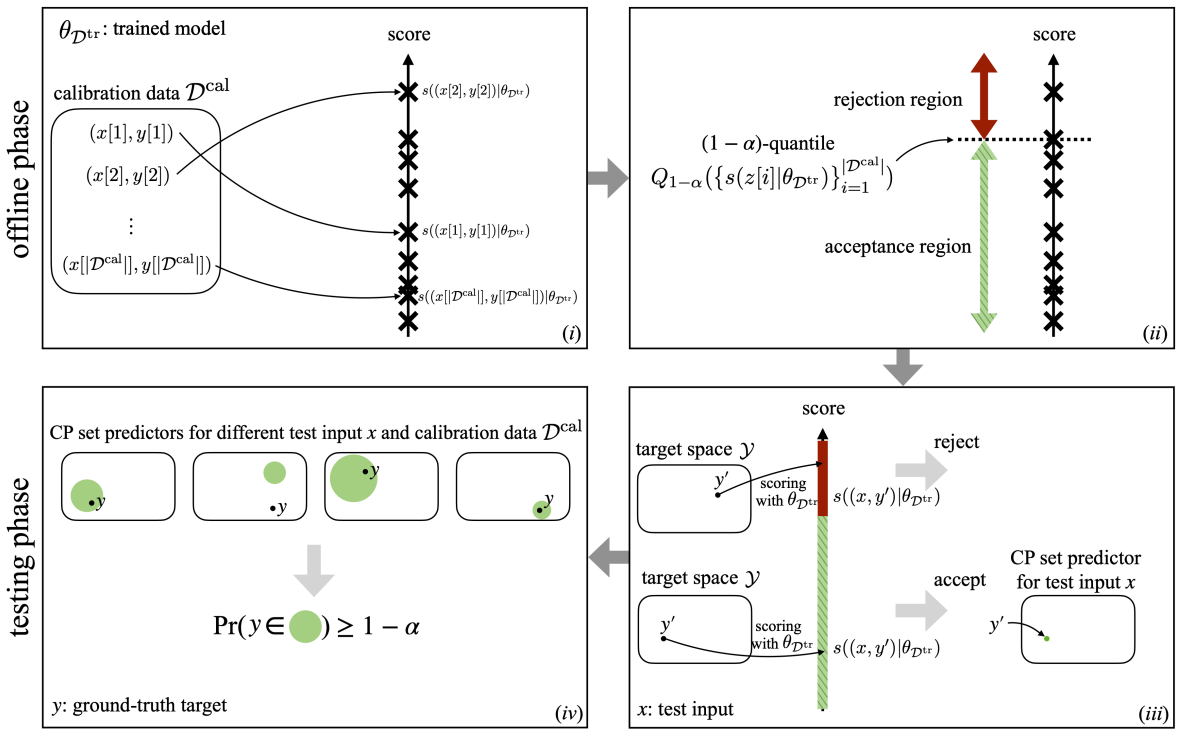


FIGURE 2. Illustration of conventional CP for deterministic (likelihood-based) models: (*i*) Based on any trained model parameter vector $\theta_{\mathcal{D}^{\rm tr}}$ and given a scoring function $s(\cdot|\theta_{\mathcal{D}^{\rm tr}})$, e.g., the quadratic loss, CP first computes scores for every calibration example in an offline phase; (*ii*) Based on the $|\mathcal{D}^{\rm cal}|$ calibration scores, CP divides the real line into an acceptance region and a rejection region by computing the $(1-\alpha)$-quantile of the calibration scores; (*iii*) CP constructs a set predictor for a test input x by collecting all candidate outputs $y' \in \mathcal{Y}$ whose scores lie in the acceptance region; (*iv*) The obtained CP set predictor (green circle) is guaranteed to satisfy the validity condition (4).

Given a trained model parameter vector $\theta_{\mathcal{D}^{\rm tr}}$, a set predictor $\Gamma(\mathbf{x}|\boldsymbol{\mathcal{D}}^{\rm cal}, \theta_{\mathcal{D}^{\rm tr}})$ is said to be *well calibrated at the coverage level* $1-\alpha$, for some $\alpha \in (0,1)$, if the following inequality holds

$$\Pr\left(\mathbf{y} \in \Gamma(\mathbf{x}|\boldsymbol{\mathcal{D}}^{\rm cal}, \theta_{\mathcal{D}^{\rm tr}})\right) \geq 1-\alpha, \tag{4}$$

where the probability $\Pr(\cdot)$ is taken over the joint distribution $p(\mathcal{D}^{\rm cal}, z)$ of calibration data set and test data point.

B. CONFORMAL PREDICTION

As mentioned, CP starts with a pre-trained model $\hat{y} = f(x|\theta_{\mathcal{D}^{\rm tr}})$, and extracts from it a well-calibrated set predictor satisfying the condition (4). To this end, CP relies on the specification of a *scoring function* $s(\cdot|\theta_{\mathcal{D}^{\rm tr}})$. The scoring function quantifies the *performance loss* accrued by the trained model on the input-output pair $z = (x, y)$. Accordingly, a scoring function should ideally increase when the error between prediction $\hat{y}$ and the true target y increases [7].

As an example of scoring functions, one may consider the quadratic loss $s(z|\theta) = (y - f(x|\theta_{\mathcal{D}^{\rm tr}}))^2$ for continuous target variables with $\mathcal{Y} = \mathbb{R}$, or the logistic loss $s(z|\theta) = \log(1 + \exp(-yf(x|\theta_{\mathcal{D}^{\rm tr}})))$ for binary random variables with $\mathcal{Y} = \{-1, 1\}$ [5].

Based on the calibration set $\mathcal{D}^{\rm cal}$ and the target coverage level $1-\alpha$, as illustrated in Fig. 2, CP divides the possible values returned by the scoring function for the trained model $f(x|\theta_{\mathcal{D}^{\rm tr}})$ into an *acceptance region* and a *rejection region*. A set predictor $\Gamma(x|\mathcal{D}^{\rm cal}, \theta_{\mathcal{D}^{\rm tr}})$ for a test input x is then obtained by including all candidate outputs $y' \in \mathcal{Y}$ whose scores $s((x, y')|\theta_{\mathcal{D}^{\rm tr}})$ lie in the acceptance region for the trained predictor.

The threshold dividing the acceptance and rejection regions is obtained by examining the calibration set $|\mathcal{D}^{\rm cal}|$. Specifically, in an *offline* phase, the scores assigned by the pre-trained model are evaluated for all data points in the calibration set, obtaining the set of $|\mathcal{D}^{\rm cal}|$ values $\{s(z[1]|\theta_{\mathcal{D}^{\rm tr}}), ..., s(z[|\mathcal{D}^{\rm cal}|]|\theta_{\mathcal{D}^{\rm tr}})\}$. Then, CP chooses the threshold such that a fraction $(1-\alpha)$ of calibration points returns a score value lower than the threshold. Intuitively, this ensures that all accepted output values also have probability no smaller than $1-\alpha$ due to the exchangeability assumption in Assumption 1 [7].

More formally, let us define as $Q_{1-\alpha}(\{s(z[i]|\theta_{\mathcal{D}^{\rm tr}})\}_{i=1}^{|\mathcal{D}^{\rm cal}|})$ the $\lceil (1-\alpha)(|\mathcal{D}^{\rm cal}|+1) \rceil$-th smallest value in the set $\{s(z[1]|\theta_{\mathcal{D}^{\rm tr}}), ..., s(z[|\mathcal{D}^{\rm cal}|]|\theta_{\mathcal{D}^{\rm tr}}), \infty\}$. The CP set predictor is defined as

$$\Gamma(x|\mathcal{D}^{\rm cal}, \theta_{\mathcal{D}^{\rm tr}}) = \Big\{ y' \in \mathcal{Y} : s((x, y')|\theta_{\mathcal{D}^{\rm tr}}) \leq Q_{1-\alpha}\big(\{s(z[i]|\theta_{\mathcal{D}^{\rm tr}})\}_{i=1}^{|\mathcal{D}^{\rm cal}|}\big) \Big\}. \tag{5}$$

Fig. 2 provides an illustration of the overall CP process.

As long as Assumption 1 holds, conventional CP guarantees that the set predictor (5) is well calibrated for any trained model $\theta_{\mathcal{D}^{\rm tr}}$.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

Theorem 1 (Calibration of deterministic CP [7])**.** *Under Assumption 1, for any miscoverage level $\alpha \in (0,1)$, for any trained model $\theta_{\mathcal{D}^{\text{tr}}}$, and for any scoring function $s(\cdot|\theta_{\mathcal{D}^{\text{tr}}})$, the CP set predictor* (5) *satisfies the validity condition* (4)*, i.e., $\Pr(\mathbf{y} \in \Gamma(\mathbf{x}|\boldsymbol{\mathcal{D}}^{cal}, \theta_{\mathcal{D}^{\text{tr}}})) \geq 1-\alpha$, with probability taken over the joint distribution of the test data $\mathbf{z}$ and calibration data set $\boldsymbol{\mathcal{D}}^{cal}$.*

For completeness, and in order to keep the presentation self-contained, a proof of Theorem 1 can be found in Appendix B.

C. CONVENTIONAL CONFORMAL PREDICTION FOR QUANTUM MODELS

In this section, we introduce a direct application of conventional CP, as described in the previous subsection, to predictions obtained via quantum machine learning models. Throughout this article, we focus on general quantum machine learning models implemented by a *parameterized quantum circuit* (PQC) (see, e.g., [2], [3] for an introduction). Furthermore, we assume in this subsection that outputs of the model are obtained by averaging over an infinite number of shots, as in most of the literature on the subject (see, e.g., [2]).

A PQC encodes the classical input x into a *parameterized quantum embedding* defined by the state of an n-qubits state. The state is described by a $2^n \times 2^n$ density matrix $\rho(x|\theta)$, which is a positive semi-definite matrix with unitary trace, i.e., $\text{Tr}(\rho(x|\theta)) = 1$, where $\text{Tr}(\cdot)$ represents the trace operator. We write $N = 2^n$ for the dimension of the Hilbert space on which density matrix $\rho(x|\theta)$ operates. The state $\rho(x|\theta)$ is obtained via the application of a parameterized unitary matrix $U(x|\theta)$ to a fiducial state $|0\rangle$ for the register of n qubits, yielding

$$\rho(x|\theta) = U(x|\theta)|0\rangle\langle 0|U(x|\theta)^{\dagger}, \tag{6}$$

where $\dagger$ represents the complex conjugate transpose operation.

Deterministic predictors can be in principle implemented via a PQC by considering the expected value of some *observable* O as the output of the model. An observable is defined by an $N \times N$ Hermitian matrix O. By the spectral theorem, an observable can be hence expressed in terms of its eigendecomposition (see, e.g., [3, Sec. 5])

$$O = \sum_{j=1}^{N'} o_j \Pi_j \tag{7}$$

with real eigenvalues $\{o_j\}_{j=1}^{N'}$ and projection matrices $\{\Pi_j\}_{j=1}^{N'}$ satisfying the resolution-of-identity equality $\sum_{j=1}^{N'} \Pi_j = I$, where $N' \leq N$ is the number of distinct eigenvalues of the observable O.

Accordingly, for a scalar target variable, the deterministic predictor is given by the expectation

$$\hat{y} = \text{Tr}(O\rho(x|\theta)) = \langle O \rangle_{\rho(x|\theta)}. \tag{8}$$

Note that multi-dimensional target spaces $\mathcal{Y}$ can be handled by considering a number of observables [6]. Henceforth, we will assume a scalar target variable $y \in \mathbb{R}$. Furthermore, we will assume that the discrete output space of the PQC $\hat{\mathcal{Y}} = \{o_j\}_{j=1}^{N'}$ is included in the target space $\mathcal{Y}$, i.e., $\hat{\mathcal{Y}} \subseteq \mathcal{Y}$. These assumptions are made to simplify the presentation, and they can be alleviated at the cost of a more cumbersome notation.

Given the deterministic parametric predictor (8), training can be carried out based on training set $\mathcal{D}^{\text{tr}}$ using local or global optimization schemes [20], [26]–[29], producing a trained parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$. Given the trained model $\text{Tr}(O\rho(x|\theta_{\mathcal{D}^{\text{tr}}}))$, one can directly apply CP as described in the previous subsection in order to obtain well-calibrated set predictors.

This direct application of CP to quantum machine learning has the important limitation of relying on the assumption that the expected value (8) can be evaluated exactly. In fact, in practice, the expectation in (8) can only be estimated based on a finite number of measurements, or shots, obtained by running the PQC multiple times and applying the projective measurement defined by projectors $\{\Pi_j\}_{j=1}^{N'}$ to the quantum embedding $\rho(x|\theta)$.

III. CONFORMAL PREDICTION FOR CLASSICAL PROBABILISTIC MODELS

This section reviews the PCP approach introduced in [23], which is a variant of CP that applies post-hoc calibration to samples produced by a *classical* probabilistic model.

Consider a parametric probabilistic predictor defined by a conditional distribution $p(y|x,\theta)$ of target output $\mathbf{y}$ given input $\mathbf{x} = x$. For example, in regression, the distribution $p(y|x,\theta)$ may describe a Gaussian random variable $\mathbf{y}$ with mean and covariance dependent on input x and parameter vector θ; or it may describe a categorical random variable $\mathbf{y}$ with logit vector dependent on input x and parameter vector θ. The mentioned functions are typically implemented as neural networks with weight vector θ and input x. Model $p(y|x,\theta)$ can be trained using standard tools from machine learning, yielding a trained model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ [3], [4].

PCP constructs a set predictor not directly as a function of the likelihood $p(y|x,\theta_{\mathcal{D}^{\text{tr}}})$ of the trained model, but rather as a function of a number of random predictions $\hat{\mathbf{y}}$ generated from the model $p(y|x,\theta_{\mathcal{D}^{\text{tr}}})$. As mentioned in Sec. I, the motivation of PCP is to obtain more flexible set predictors that can describe disjoint error bars [23, Fig. 1]. An illustration of PCP can be found in Fig. 3.

To elaborate, given test input x and trained model $p(y|x,\theta_{\mathcal{D}^{\text{tr}}})$, PCP generates M i.i.d. predictions $\hat{\mathbf{y}}^{1:M} = \{\hat{\mathbf{y}}^m\}_{m=1}^M$, with each sample drawn from the model as

$$\hat{\mathbf{y}}^m \sim p(y|x,\theta_{\mathcal{D}^{\text{tr}}}). \tag{9}$$

It then calibrates these predictions by using the calibration set. To this end, given the calibration set $\mathcal{D}^{\text{cal}}$, in an *offline* phase, for each calibration example $z[i] \in \mathcal{D}^{\text{cal}}$, PCP generates M i.i.d. random predictions $\hat{\mathbf{y}}^{1:M}[i] = \{\hat{\mathbf{y}}^m[i]\}_{m=1}^M$,

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

with each sample obtained from the model as $\hat{\mathbf{y}}^m[i] \sim p(y|x[i], \theta_{\mathcal{D}^{\text{tr}}})$.

Like CP, PCP also relies on the use of a scoring function that evaluates the loss of the trained model on each data point. Unlike CP, the scoring function of PCP is not a function of a single, deterministic, prediction $\hat{y}$, but rather of the M random predictions $\hat{\mathbf{y}}^{1:M}$ generated i.i.d. from the model $p(y|x, \theta_{\mathcal{D}^{\text{tr}}})$ for the given test input x. Accordingly, we write as $s((x, y)|\hat{\mathbf{y}}^{1:M})$ the scoring function, which measures the loss obtained by the trained model on an example $z = (x, y)$ based on the random predictions $\hat{\mathbf{y}}^{1:M}$.

The work [23] proposed the scoring function

$$s(z = (x, y)|\hat{\mathbf{y}}^{1:M}) = \min_{m \in \{1,\ldots,M\}} |y - \hat{\mathbf{y}}^m|. \tag{10}$$

This loss metric uses the best among the M random predictions $\hat{\mathbf{y}}^m$ to evaluate the score on example z as the Euclidean distance $|y - \hat{\mathbf{y}}^m|$ between the best prediction $\hat{\mathbf{y}}^m$ and the output y [23].

Having defined the scoring function as in (10), PCP obtains a set predictor (5) as for CP by replacing the scoring function $s(z|\theta)$, e.g., the quadratic loss $(y - f(x|\theta))^2$, with the scoring function (10) for both test and calibration pairs. This yields the set predictor

$$\begin{aligned}\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) = \Big\{ & y' \in \mathcal{Y} : s((x, y')|\hat{\mathbf{y}}^{1:M}) \\ & \leq Q_{1-\alpha}\Big(\big\{s(z[i]|\hat{\mathbf{y}}^{1:M}[i])\big\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}\Big)\Big\}.\end{aligned} \tag{11}$$

When evaluated using the scoring function (10), this results in a generally disjoint set of intervals for scalar variables (and circles for two-dimensional variables) [23].

As long as the examples in the calibration set $\boldsymbol{\mathcal{D}}^{\text{cal}}$ and the test example $\mathbf{z}$ are finitely exchangeable random variables (Assumption 1), the PCP set predictor (11) is well calibrated as formalized next.

Theorem 2 (Calibration of PCP [23])**.** *Under Assumption 1 with sampling procedure following* (9)*, for any miscoverage level $\alpha \in (0, 1)$ and for any trained model $\theta_{\mathcal{D}^{\text{tr}}}$, the PCP set predictor* (11) *satisfies the inequality*

$$\Pr(\mathbf{y} \in \mathbf{\Gamma}_M(\mathbf{x}|\boldsymbol{\mathcal{D}}^{cal}, \theta_{\mathcal{D}^{tr}})) \geq 1 - \alpha, \tag{12}$$

with probability taken over the joint distribution of the test data $\mathbf{z}$, of the calibration data set $\boldsymbol{\mathcal{D}}^{cal}$, and also over the independent random predictions $\hat{\mathbf{y}}^{1:M}$ and $\{\hat{\mathbf{y}}^{1:M}[i]\}_{i=1}^{|\mathcal{D}^{cal}|}$ produced by the model for test and calibration points.

A proof of Theorem 2 is provided for completeness in Appendix B.

IV. QUANTUM CONFORMAL PREDICTION

In this section, we describe the proposed QCP method, a post-hoc calibration scheme for quantum machine learning that treats PQCs as implicit, i.e., likelihood-free, probabilistic models.

A. PQCS AS IMPLICIT PROBABILISTIC MODELS

As discussed in Sec. II-C, the output of PQCs is typically taken to be the expectation (8) of one or more observables. However, the exact evaluation of this quantity requires the capacity to run the PQC for an arbitrarily large number of times in order to average over a, theoretically infinite, number of shots. In contrast, in this section, we view the output of the PQC over any, potentially small, number of shots as probabilistic.

To elaborate, let us fix a trained model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$, as well as a projective measurement defined by the projection matrices $\{\Pi_j\}_{j=1}^{N'}$ and corresponding numerical outputs $\{o_j\}_{j=1}^{N'}$ as in (7). For any input x, the output $\hat{\mathbf{y}}$ obtained from the model equals value o_j with probability

$$p(\hat{\mathbf{y}} = o_j|x, \theta_{\mathcal{D}^{\text{tr}}}) = \text{Tr}(\Pi_j \rho(x|\theta_{\mathcal{D}^{\text{tr}}})). \tag{13}$$

The distribution (13) is generally not directly accessible. Rather, the model is implicit, in the sense that all that can be observed by a user are samples $\hat{\mathbf{y}} \sim p(y|x, \theta_{\mathcal{D}^{\text{tr}}})$ drawn from it.

Treating a PQC as an implicit probabilistic model has the additional advantage that one can seamlessly account for the presence of *quantum hardware noise*. To this end, let us write as $\tilde{\rho}(x|\theta_{\mathcal{D}^{\text{tr}}})$ the quantum embedding produced by a noisy quantum computer. The density matrix $\tilde{\rho}(x|\theta_{\mathcal{D}^{\text{tr}}})$ accounts also for the effect of quantum channels describing gate noise [30]–[33]. Then, the noisy PQC can be viewed as an implicit probabilistic model that, for any input x, produces a sample $\hat{\mathbf{y}}$ equal to o_j with probability

$$p(\hat{\mathbf{y}} = o_j|x, \theta_{\mathcal{D}^{\text{tr}}}) = \text{Tr}(\Pi_j \tilde{\rho}(x|\theta_{\mathcal{D}^{\text{tr}}})) \tag{14}$$

for all $j = 1, ..., N'$. One can also similarly account for other sources of noise, including in the preparation and measurement phases [34]–[36].

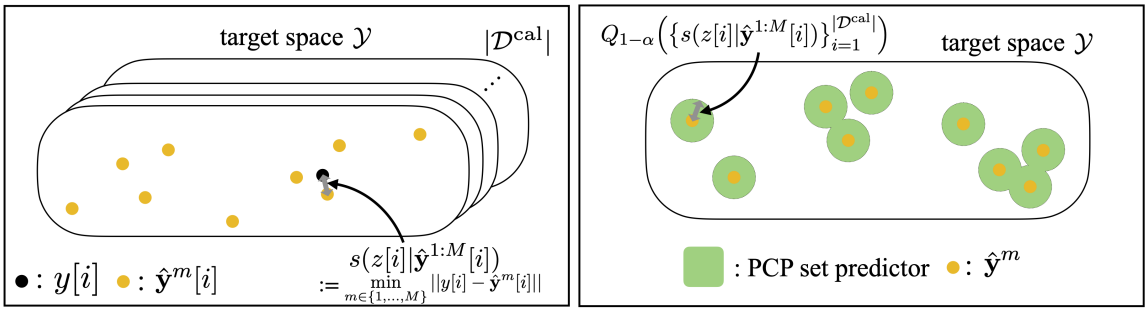
We will follow the standard assumption that each new m-th measurement, or shot, for a given input x produces an independent output $\hat{\mathbf{y}}^m$ with distribution (14). Accordingly, given an input x, we will assume the availability of M independent measurements $\hat{\mathbf{y}}^{1:M} = \{\hat{\mathbf{y}}^m\}_{m=1}^M$ from distribution (14). This is in line with the assumption made in PCP, which considers the output predictions in (9) to be i.i.d. While we will not elaborate on this point further, we note that it is also possible to extend the properties of QCP to more general measurement models characterized by correlation across the measurement outputs $\hat{\mathbf{y}}^{1:M}$ produced for each input x, as long as there is no correlation across shots for distinct inputs x. This correlation may arise, for instance, due to a system or diffusive drift in the quantum noise processes [37]. We note that, in practice, correlations may also extend across multiple inputs x [24].

B. QUANTUM CONFORMAL PREDICTION

For a test input x, QCP processes the M predictions $\hat{\mathbf{y}}^{1:M} = \{\hat{\mathbf{y}}^m\}_{m=1}^M$ made via the PQC with the goal of producing a well-calibrated set predictor with generalization guarantees.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224



(*i*) sampling-based scoring for calibration example (*ii*) constructing the PCP set predictor

FIGURE 3. (*i*) In PCP, the score for each calibration example $z[i] = (x[i], y[i])$ is evaluated based on independent random predictions $\hat{\mathbf{y}}^{1:M}[i]$ produced by the trained model as in (9). As an example, using the scoring function (10), the score is obtained by evaluating the minimum distance between the true output $y[i]$ and the M random predictions $\hat{\mathbf{y}}^m[i]$. (*ii*) The PCP set predictor uses the M random predictions $\hat{\mathbf{y}}^{1:M}$ for test input x in a similar way in order to evaluate the score of every possible candidate output $y' \in \mathcal{Y}$. With the scoring function (10), PCP yields a predictive set given by disjoint regions in the plane. This set is guaranteed to contain the true output $\mathbf{y}$ with the predetermined coverage level $1-\alpha$.

As detailed in the previous subsection, the M predictions are obtained via M independent measurements of the output of the PQC, which are distributed as in (14). Following the CP framework, QCP leverages a calibration data set $\mathcal{D}^{\text{cal}}$ consisting of $|\mathcal{D}^{\text{cal}}|$ calibration examples $z[i]$ for $i = 1, ..., |\mathcal{D}^{\text{cal}}|$. In the rest of this subsection, we detail the operation of QCP, which is illustrated in Fig. 4 and summarized in Algorithm 1.

Like CP and PCP, QCP is divided into an offline phase, taking place only once based on the calibration data set, and a testing phase in which set predictions are produced for a set of desired test inputs. Both phases rely on the specification of a *scoring function* $s(z|\hat{\mathbf{y}}^{1:M})$, which, given an example $z = (x, y)$, quantifies the prediction loss associated with samples $\hat{\mathbf{y}}^{1:M}$ produced by the PQC in response to an input x when the correct output is y. The scoring function may be selected as in (10), as suggested in [23]. More general choices, which will be proved to be advantageous in Sec. VII, are presented in Sec. IV-D.

During the *offline phase*, for each calibration point $z[i]$, the PQC with trained model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ generates M independent predictions $\hat{\mathbf{y}}^{1:M}[i] = \{\hat{\mathbf{y}}^m[i]\}_{i=1}^M$ from distribution (13) (Fig. 4-(*i*)). Then, following the CP methodology detailed in Sec. II-B, QCP divides the possible values returned by the scoring function into an *acceptance region* and a *rejection region* (Fig. 4-(*ii*)). The acceptance threshold is given by the $\lceil (1-\alpha)(|\mathcal{D}^{\text{cal}}|+1)\rceil$-th smallest value in the set $\{s(z[1]|\hat{\mathbf{y}}^{1:M}[1]), ..., s(z[|\mathcal{D}^{\text{cal}}|]|\hat{\mathbf{y}}^{1:M}[|\mathcal{D}^{\text{cal}}|]), \infty\}$ of scores for the calibration data points. As in Sec. II-B, this quantity is denoted as $Q_{1-\alpha}(\{s(z[i]|\hat{\mathbf{y}}^{1:M}[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|})$.

In the *testing phase*, for a test input x, the PQC with the same trained model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ produces M predictions $\hat{\mathbf{y}}^{1:M}$ (Fig. 4-(*iii*, dotted box)). Finally, QCP produces the prediction set $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) \subseteq \mathcal{Y}$ of the *target space* $\mathcal{Y}$, by including in the set predictor $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$ all candidate outputs $y' \in \mathcal{Y}$ whose scores $s((x, y')|\hat{\mathbf{y}}^{1:M})$ lie in the acceptance region (Fig. 4-(*iii*)). Accordingly, the QCP set predictor is defined as

$$\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) = \Big\{y' \in \mathcal{Y} : s((x, y')|\hat{\mathbf{y}}^{1:M}) \leq Q_{1-\alpha}\Big(\{s(z[i]|\hat{\mathbf{y}}^{1:M}[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}\Big)\Big\}. \tag{15}$$

In Algorithm 1, which summarizes QCP, the complexity of the offline phase (lines 3–7) can be amortized by reusing the threshold identified during this phase across multiple test inputs. In this regard, for general scoring functions, it may not be feasible to try out all values of $y' \in \mathcal{Y}$ for inclusion in the predictive set (lines 11–14). This problem can be addressed in one of two ways. First, for specific scoring functions, such as (10), the predictive set can be obtained in closed form, as we elaborate on in Sec. IV-D. Second, it is possible to quantize the target space in order to reduce the complexity of the search among values of $y' \in \mathcal{Y}$ to include in the predictive set [38].

C. THEORETICAL CALIBRATION GUARANTEES

QCP inherits the calibration properties of CP and PCP. In particular, as long as the examples in the calibration set $\boldsymbol{\mathcal{D}}^{\text{cal}}$ and the test example $\mathbf{z}$ are finitely exchangeable random variables (Assumption 1), the QCP set predictor (15) is well calibrated at coverage level $1-\alpha$, as detailed next.

Theorem 3 (Calibration of QCP)**.** *Under Assumption 1 with sampling procedure following* (14)*, for any coverage level* $1-\alpha \in (0, 1)$ *and for any trained PQC with model parameter vector* $\theta_{\mathcal{D}^{tr}}$*, the QCP set predictor* (11) *satisfies the inequality*

$$\Pr(\mathbf{y} \in \mathbf{\Gamma}_M(\mathbf{x}|\boldsymbol{\mathcal{D}}^{cal}, \theta_{\mathcal{D}^{tr}})) \geq 1-\alpha, \tag{16}$$

with probability taken over the joint distribution of the test data $\mathbf{z}$*, of the calibration data set* $\boldsymbol{\mathcal{D}}^{cal}$*, and of the independent random predictions* $\hat{\mathbf{y}}^{1:M}$ *and* $\{\hat{\mathbf{y}}^{1:M}[i]\}_{i=1}^{|\mathcal{D}^{cal}|}$ *produced by the PQC for test and calibration points, respectively.*

A proof of Theorem 3 is provided in Appendix B.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

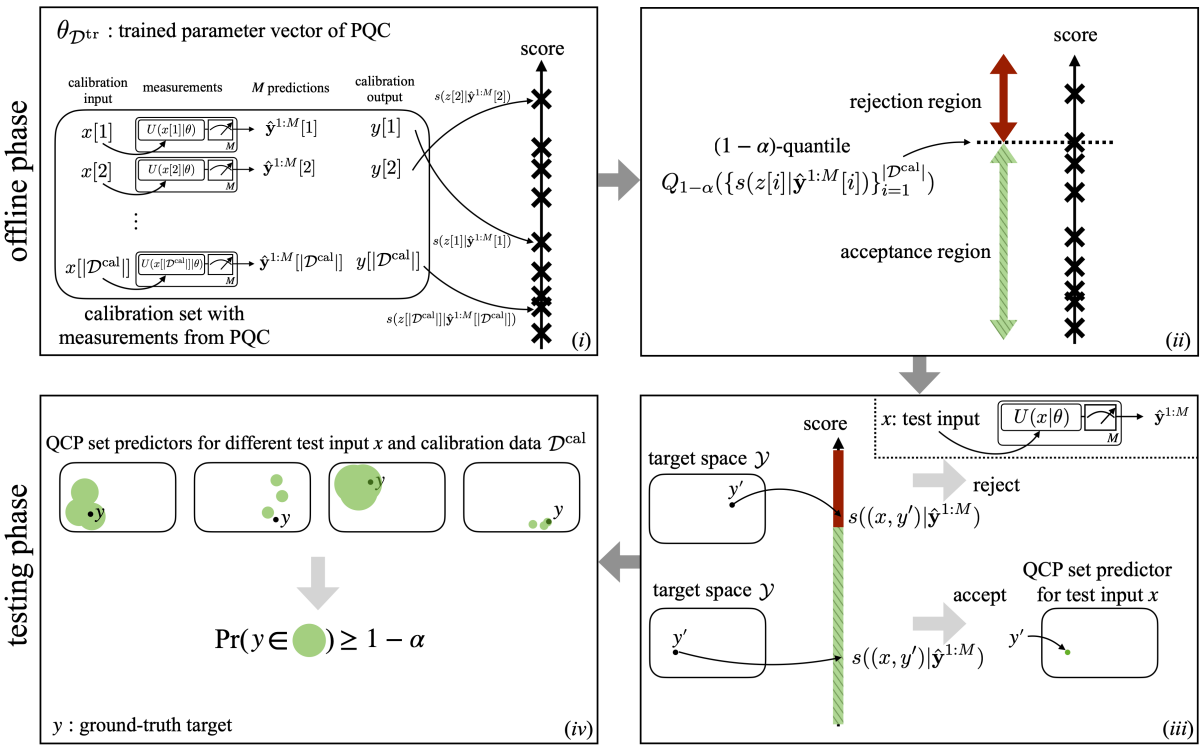


FIGURE 4. Illustration of QCP for a PQC with trained parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ and for a scoring function $s((x,y)|\hat{\mathbf{y}}^{1:M})$ (see Sec. IV-D): In the *offline phase*, (*i*) QCP first computes a score for each calibration example based on M random predictions produced by the PQC; (*ii*) then, based on the obtained $|\mathcal{D}^{\text{cal}}|$ calibration scores, QCP divides the real line into an acceptance region and a rejection region by using as a threshold the $(1-\alpha)$-quantile of the calibration scores. In the *testing* phase, (*iii*) QCP constructs a set predictor for a test input x by including in the set all candidate outputs $y' \in \mathcal{Y}$ whose scores lie in the acceptance region. (*iv*) The obtained QCP set predictor (green circle) is guaranteed to satisfy the validity condition (16).

D. SCORING FUNCTION

In this subsection, we elaborate on possible choices for the scoring function to be used in QCP. We note that the same considerations apply also to PCP. We start by interpreting the scoring function (10) assumed by PCP as the loss associated with a *non-parametric estimate* of the conditional distribution $p(y|x,\theta)$ of the target variable $\mathbf{y}$ given the input variable $\mathbf{x}$, where the estimate is obtained based on the samples $\hat{\mathbf{y}}^{1:M}$ produced by the model. We then observe that any loss function for a probabilistic predictor, coupled with an estimate of the conditional distribution $p(y|x,\theta)$, yields a valid scoring function.

To elaborate, we review the notion of loss function for an *explicit* probabilistic predictor $p(y|x,\theta)$ (see, e.g., [7, Sec. 2.2.2]). An explicit probabilistic predictor is one that outputs the probability distribution $p(y|x,\theta)$. Given an example $z=(x,y)$, the loss accrued by probabilistic model $p(y|x,\theta)$ can be evaluated as $g(p(y|x,\theta))$, where $g(\cdot)$ is a strictly decreasing function. Accordingly, the loss increases if the model assigns a lower probability $p(y|x,\theta)$ to the correct output y given input x. Typical examples of such functions include the negative logarithm, which yields the *log-loss* $-\log p(y|x,\theta)$, and the *inverse function* $1/p(y|x,\theta)$ (see, e.g., [3, Sec. 6.7] for further examples).

The class of loss functions outlined in the previous paragraph is not applicable to implicit models, which do not produce the probabilities $p(y|x,\theta)$, but only samples from this distribution. However, given M samples $\hat{\mathbf{y}}^{1:M}$ generated i.i.d. from the distribution $p(y|x,\theta)$, one can obtain an estimate $\hat{p}(y|x,\theta)$ of the probability $p(y|x,\theta)$. The key observation of this subsection is that such an estimate can be used to define a scoring function as follows.

Definition 1 (Scoring function for implicit models)**.** *Given M i.i.d. predictions $\hat{\mathbf{y}}^{1:M}$ drawn from a model $p(y|x,\theta)$ for input x, a scoring function for input-output pair $z=(x,y)$ is defined as*

$$s(z|\hat{\mathbf{y}}^{1:M}) = g(\hat{p}(y|x,\theta)), \tag{17}$$

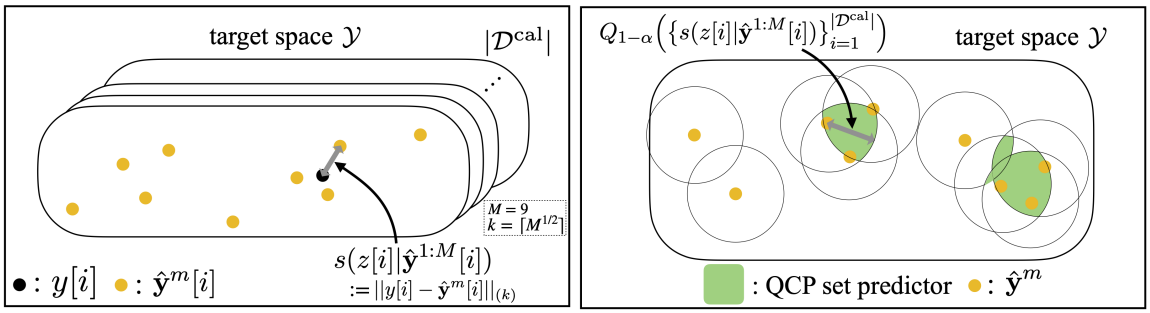
where $g(\cdot)$ is a monotonically decreasing function and $\hat{p}(y|x,\theta)$ is an estimator of the probability $p(y|x,\theta)$ obtained as a function of the samples $\hat{\mathbf{y}}^{1:M}$.

To define a score function as in (17) one can generally use either *non-parametric* density estimators, including kernel density estimation [39], variable kernel density estimation [40], [41], and histogram-based density estimation [42]; or *parametric* density estimators, including contrastive density learning [43], and normalizing flows [44] (see, e.g., [5, Sec. 7.3] for a summary).

We elaborate here on the use of non-parametric estimators, and we focus as done throughout the article on a scalar target variable y. Given M samples $\hat{\mathbf{y}}^{1:M}$ drawn i.i.d. from

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224



(*i*) sampling-based scoring for calibration example (*ii*) constructing the QCP set predictor

FIGURE 5. (*i*) In QCP, the score for each calibration example $z[i] = (x[i], y[i])$ is evaluated based on independent random predictions $\hat{\mathbf{y}}^{1:M}[i]$ produced by the trained model as in (9). Using the proposed scoring function (17)-(18), the score is obtained by evaluating the k-th smallest distance between the true output $y[i]$ and the M random predictions $\hat{\mathbf{y}}^m[i]$. (*ii*) The QCP set predictor uses the M random predictions $\hat{\mathbf{y}}^{1:M}$ for test input x in order to evaluate the score of every possible candidate output $y' \in \mathcal{Y}$. QCP yields a predictive set given by disjoint regions in the plane. By not relying solely on the closest prediction as PCP (see Fig. 3), the prediction sets produced by QCP can be more robust to noisy predictions (see Sec. VII). This set is guaranteed to contain the true output $\mathbf{y}$ with the predetermined coverage level $1 - \alpha$.

Algorithm 1: Quantum Conformal Prediction (QCP)

Input: trained parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ for the PQC; calibration data set $\mathcal{D}^{\text{cal}} = \{z[i] = (x[i], y[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}$; test input x; number of shots $M \geq 1$; desired coverage level $1 - \alpha$

Output: well-calibrated predictive set $\Gamma_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$ at coverage level $1 - \alpha$

choose scoring function $s((x, y)|\hat{\mathbf{y}}^{1:M})$ as outlined in Sec. IV-D
`Offline phase`
for i-th calibration data example $z[i]$ with $i = 1, ..., |\mathcal{D}^{\text{cal}}|$ **do**
  given input $x[i]$, produce M independent predictions $\hat{\mathbf{y}}^{1:M}[i] = \{\hat{\mathbf{y}}^m[i]\}_{m=1}^M$ via M shots of the PQC
  compute the corresponding score $s(z[i]|\hat{\mathbf{y}}^{1:M}[i])$
end
find the $\lceil (1 - \alpha)(|\mathcal{D}^{\text{cal}}| + 1) \rceil$-th smallest value among the obtained scores, which is denoted as $Q_{1-\alpha}(\{s(z[i]|\hat{\mathbf{y}}^{1:M}[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|})$
`Testing phase for input` x
initialize predictive set $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) \leftarrow \{\}$
for test input x, make M predictions $\hat{\mathbf{y}}^{1:M} = \{\hat{\mathbf{y}}^m\}_{m=1}^M$ via PQC
for candidate output $y' \in \mathcal{Y}$ **do**
  compute the corresponding score $s((x, y')|\hat{\mathbf{y}}^{1:M})$
  if $s((x, y')|\hat{\mathbf{y}}^{1:M}) \leq Q_{1-\alpha}(\{s(z[i]|\hat{\mathbf{y}}^{1:M}[i])\}_{i=1}^{|\mathcal{D}^{cal}|})$
    $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) \leftarrow \mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) \cup \{y'\}$
end
return the predictive set $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$ for test input x

distribution $p(y|x, \theta)$, and a positive integer $k \leq M$, the *k-nearest neighbor (k-NN) density estimator* of $p(y|x, \theta)$ is given by [41]

$$\hat{p}(y|x, \theta) = \frac{k}{2M|y - \hat{\mathbf{y}}^m|_{(k)}}, \tag{18}$$

where $|y - \hat{\mathbf{y}}^m|_{(k)}$ is the k-th smallest Euclidean distance in the set of distances $\{|y - \hat{\mathbf{y}}^m|\}_{m=1}^M$. This estimate is asymptotically consistent: As long as k grows suitably with M, the estimate (18) tends to the true distribution $p(y|x, \theta)$ pointwise in probability [41]. In particular, parameter k should be selected in such a way that the limits $\lim_{M\to\infty} k(M) = \infty$ and $\lim_{M\to\infty} k(M)/M = 0$ are satisfied. An example of such dependence is given by [41]

$$k(M) = \lceil M^{1/2} \rceil. \tag{19}$$

With this background in place, we can now show that the PCP scoring function (10) corresponds to a special case of the class of scoring functions (17). In fact, with the inverse function for $g(\cdot)$ and the 1-NN estimator (18), the scoring function (17) reads

$$\begin{aligned} s(z = (x, y)|\hat{\mathbf{y}}^{1:M}) &= \frac{1}{\hat{p}(y|x, \theta)} \\ &= 2M|y - \hat{\mathbf{y}}^m|_{(1)}, \end{aligned} \tag{20}$$

which equals (10) apart from an inessential multiplicative constant.

In the experiments in Sec. VII, we will consider the more general scoring function in class (20) obtained with the k-NN predictor (18). With this choice, the QCP set predictor (15), as well as the PCP set predictor (11) can be expressed as

$$\begin{aligned} &\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}}) \\ &= \left\{ y' \in \mathcal{Y} : \sum_{m=1}^M \mathbb{1}\left(y' \in B_{q(\mathcal{D}^{\text{cal}})}(\hat{\mathbf{y}}^m)\right) \geq k \right\}, \end{aligned} \tag{21}$$

in which $B_{q(\mathcal{D}^{\text{cal}})}(y)$ is the closed interval with center point y, i.e., $B_r(y) = \{y' \in \mathcal{Y} : |y - y'| \leq r\}$, and radius $q(\mathcal{D}^{\text{cal}}) = Q_{1-\alpha}\left(\left\{|y[i] - \hat{\mathbf{y}}^m[i]|_{(k)}\right\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}\right)$. Illustration of the QCP set

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

predictor with k chosen as (19) can be found in Fig. 5, while experimental study regarding the impact of k on the size of the set predictor (21) can be found in Appendix C.

E. APPLICATION TO QUANTUM DATA

So far we have focused on situations in which input data x is of classical nature. Accordingly, given a pre-trained PQC, one can produce M copies of the quantum embedding state $\rho(x|\theta_{\mathcal{D}^{\text{tr}}})$ to be used by QCP in order to yield the set predictor $\mathbf{\Gamma}_M(x|\mathcal{D}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$. However, QCP can be equally well applied to settings in which the pre-trained model takes as input quantum data. We briefly elaborate on this applications in the rest of this subsection by focusing on the classification of quantum data.

Assume that there are C classes of quantum states, indexed by variable $y \in \{1, ..., C\}$, such that the y-th class corresponds to a density matrix $\rho(y)$. The data generating mechanism is specified by a distribution $p(y)$ over the class index y and by the set of density matrices $\rho(y)$ with $y \in \{1, ..., C\}$. Given any pre-designed positive operator-valued measurement (POVM) defined by positive semidefinite matrices $\mathcal{P} = \{\mathrm{P}_1, ..., \mathrm{P}_C\}$, the POVM produces output $\hat{\mathbf{y}}$ with probability

$$\hat{\mathbf{y}} \sim p(y|\rho, \mathcal{P}) = \text{Tr}(\mathrm{P}_y \rho). \tag{22}$$

Note that the POVM $\mathcal{P}$ can be implemented via a pre-trained PQC with parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ followed by a projective measurement as per the setting considered above, yielding (13) when removing input x. Here we adopt a more general formulation. As we will see in Sec. VII-C, this allows us to consider other quantum detectors such as pretty good measurements [45], [46].

The goal of QCP is to use the predesigned model operating on M copies of a test state $\rho \in \{\rho(1), ..., \rho(C)\}$, denoted as $\rho^{\otimes M}$, as well as M copies of each example in the calibration data set $\mathcal{D}^{\text{cal}}$, to produce a set predictor $\mathbf{\Gamma}_M(\rho^{\otimes M}|\mathcal{D}^{\text{cal}}, \mathcal{P})$ that contains the true label with predetermined coverage level $1 - \alpha$. The calibration data is of the form $\mathcal{D}^{\text{cal}} = \{(\rho(y[i])^{\otimes M}, y[i])\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}$, where label $y[i] \in \{1, ..., C\}$ and M copies of corresponding state $\rho(y[i])$ are available for each i-th calibration example. Mathematically, the reliability requirement can be expressed as the inequality

$$\Pr\left(\mathbf{y} \in \mathbf{\Gamma}_M(\rho(\mathbf{y})^{\otimes M}|\boldsymbol{\mathcal{D}}^{\text{cal}}, \mathcal{P})\right) \geq 1 - \alpha, \tag{23}$$

where the probability $\Pr(\cdot)$ is taken over the i.i.d. calibration and test labels $\mathbf{y}, \mathbf{y}[1], ..., \mathbf{y}[|\mathcal{D}^{\text{cal}}|] \sim p(y)$ and over the independent random predictions $\hat{\mathbf{y}}^{1:M}$ and $\{\hat{\mathbf{y}}^{1:M}[i]\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}$ in (22).

QCP can be directly applied to the test predictions $\hat{\mathbf{y}}^{1:M}$ and to the calibration predictions $\{\hat{\mathbf{y}}^{1:M}[i]\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}$ as detailed in Algorithm 1. By Theorem 3, QCP satisfies the reliability condition (16). We also emphasize, following the discussions in Sec. IV-A, that the result applies also in the presence of quantum noise affecting the copies of the input states. Such noise can be generally expressed as a channel $\mathcal{N}(\cdot)$ acting on the M copies as $\mathcal{N}(\rho^{\otimes M})$, as long as the same channel $\mathcal{N}(\cdot)$ affects both test and calibration input examples.

V. CALIBRATION GUARANTEES OVER A FINITE NUMBER OF EXPERIMENTS

In the previous sections, we have described CP schemes that satisfy calibration conditions, namely (4), (12), and (16), that are defined on average over the calibration and test data points. For PCP and QCP, the average is also taken with respect to the random predictions produced for calibration and test data points. In this section, we elaborate on the practical significance of this expectation.

Suppose that we run CP, PCP or QCP (Algorithm 1) over K runs that use independent calibration and test data. What is the fraction of such runs that meet the condition that the true output is included in the predictive set? Ideally, this fraction will be close to the desired target $1 - \alpha$ with high probability. In fact, by the results in the previous sections and by the law of large numbers, as K grows large, this fraction of "successful" experiments will tend to $1 - \alpha$. What can be guaranteed for a finite number K of experiments? In the following, we address this question for conventional CP first, and then for PCP and QCP.

A. CONVENTIONAL CP

Following the setting described above, let us consider K experiments, such that in each k-th experiment we draw calibration and test data from the joint distribution $p(\mathcal{D}^{\text{cal}}, z)$, i.e., $(\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \mathbf{z}_k) \sim p(\mathcal{D}^{\text{cal}}, z)$. For each experiment, we evaluate whether the prediction set (5) produced by CP contains the true target label $\mathbf{y}_k$ or not. Accordingly, the fraction of "successful" experiments is given by

$$\hat{\mathbf{P}} = \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\big(\mathbf{y}_k \in \Gamma(\mathbf{x}_k|\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})\big), \tag{24}$$

where $\mathbb{1}(\cdot)$ is the indicator function ($\mathbb{1}(\text{true}) = 1$ and $\mathbb{1}(\text{false}) = 0$). To restate the question posed at the beginning of this section, given K, how large can we guarantee the success rate $\hat{P}$ to be?

By the exchangeability of calibration and test data (Assumption 1), which implies the exchangeability of the $|\mathcal{D}^{\text{cal}}| + 1$ scores evaluated on calibration and test data (see Appendix B), the distribution of random variable $K\hat{\mathbf{P}}$ is given by the binomial

$$K\hat{\mathbf{P}} \sim \text{Binom}\left(K, \frac{\lceil (1-\alpha)(|\mathcal{D}^{\text{cal}}| + 1)\rceil}{|\mathcal{D}^{\text{cal}}| + 1}\right), \tag{25}$$

if ties between the $|\mathcal{D}^{\text{cal}}| + 1$ scores occur with probability zero (see [8], [47] for other uses of this assumption). Note that the distribution (25) can be recovered from [10, Sec. C] by setting the number of test points to be one (see also [48]).

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

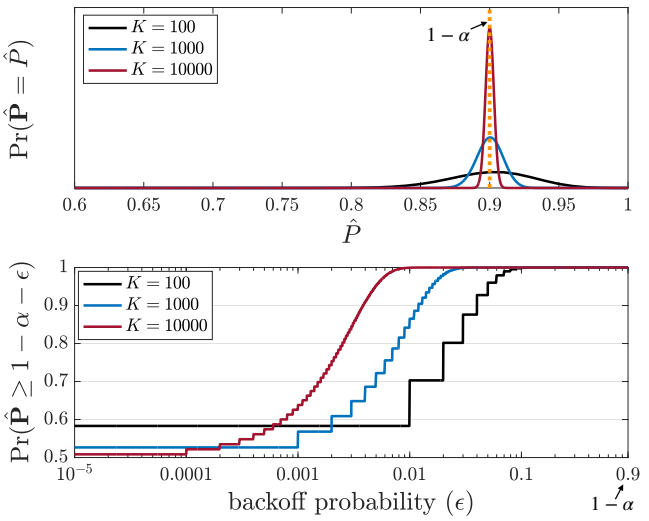


FIGURE 6. (top) Normalized density of the empirical coverage rate $\hat{\mathbf{P}}$ obtained from $K = 100, 1000, 10000$ independent trials. (bottom) Probability that the empirical coverage rate $\hat{\mathbf{P}}$ satisfies the validity condition with tolerance level, or backoff probability, ϵ as a function of ϵ, for $K = 100, 1000, 10000$ independent trials ($\alpha = 0.1, |\mathcal{D}^{\text{cal}}| = 9$).

This implies that the success rate $\hat{\mathbf{P}}$ is larger than $1-\alpha-\epsilon$ for any $\epsilon > 0$ with probability

$$\Pr(\hat{\mathbf{P}} \geq 1-\alpha-\epsilon) \tag{26}$$
$$= I_{\frac{\lceil (1-\alpha)(|\mathcal{D}^{\text{cal}}|+1) \rceil}{|\mathcal{D}^{\text{cal}}|+1}} (\lceil K(1-\alpha-\epsilon) \rceil, \lfloor K(\alpha+\epsilon) \rfloor + 1),$$

with regularized incomplete beta function $I_x(a,b) = B(x;a,b)/B(a,b)$, where $B(x;a,b) = \int_0^x t^{a-1}(1-t)^{b-1}\mathrm{d}t$ and $B(a,b) = B(1;a,b)$ [49].

Fig. 6 shows the probability distribution of random variable $\hat{\mathbf{P}}$, as well as the probability (26) as a function of tolerance level ϵ for different number of experiments $K = 100, 1000, 10000$, given $\alpha = 0.1$ and $|\mathcal{D}^{\text{cal}}| = 9$. The top figure confirms that, by the law of large numbers, the distribution of success rate $\hat{\mathbf{P}}$ concentrates around the value $1-\alpha = 0.9$. The bottom figure can be used to identify a value of the *backoff probability* ϵ that allows one to obtain finite-K guarantees on the success rate $1-\alpha-\epsilon$. For instance, when $K = 1000$, we observe that setting $\epsilon = 0.03$ guarantees a success rate $\hat{\mathbf{P}}$ no smaller than 0.87 with probability larger than 0.999.

B. PROBABILISTIC AND QUANTUM CP

In the case of PCP and QCP, each k-th experiment involves also the predictions $\{\hat{\mathbf{y}}_k^{1:M}[i]\}_{i=1}^{|\mathcal{D}^{\text{cal}}|}$ for the calibration points $\boldsymbol{\mathcal{D}}_k^{\text{cal}}$, and $\hat{\mathbf{y}}_k^{1:M}$ for the test data $\mathbf{z}_k$, following either (9) (PCP) or (14) (QCP). Despite the presence of the additional randomness due to the stochastic predictions, the finite-K guarantees (25)-(26) still hold for the fraction of "successful" experiments

$$\hat{\mathbf{P}}_M := \frac{1}{K}\sum_{k=1}^{K} \mathbb{1}\big(\mathbf{y}_k \in \boldsymbol{\Gamma}_M(\mathbf{x}_k|\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})\big) \tag{27}$$

for PCP and QCP if ties between the $|\mathcal{D}^{\text{cal}}|+1$ scores occur with probability zero. This is because the $|\mathcal{D}^{\text{cal}}|+1$ scores for calibration and test data are exchangeable also for PCP and QCP due to the exchangeability of calibration and test data (Assumption 1), and to the independence of M predictions (9), (14) for distinct inputs.

VI. EXPERIMENTAL SETTINGS

In the rest of this article, we demonstrate the validity of the proposed QCP method by addressing both an unsupervised learning task, namely density learning, and a supervised learning task, namely regression. We start in this section by describing the experimental settings, including problem definition and assumed PQC ansatzes, while the next section presents the experimental results. We cover first density learning and then regression.

A. UNSUPERVISED LEARNING: DENSITY LEARNING

Given a data set $\mathcal{D} = \{y[i]\}_{i=1}^{|\mathcal{D}|}$, with training samples $y[i]$ following an unknown population distribution $p^*(y)$ on the real line, density estimation aims at inferring some properties about the distribution $p^*(y)$ (see, e.g., [5, Sec. 7.3]). We make the standard assumption that the data points in set $\mathcal{D}$ are drawn i.i.d. from the population distribution $p^*(y)$. Following [51], we specifically focus on the problem of reliably identifying a collection of intervals that are guaranteed to contain a test sample $y \sim p^*(y)$ with coverage probability at least $1-\alpha$, as illustrated in Fig. 7-(*i*). We recall that, with CP, PCP, and QCP, the coverage probability is evaluated also with respect to the calibration data set. The ground-truth population distribution $p^*(y)$ is the mixture of Gaussians $p^*(y) = \frac{1}{2}\mathcal{N}(-0.75, 0.1^2) + \frac{1}{2}\mathcal{N}(0.75, 0.1^2)$ as also shown in Fig. 7-(*i*).

1) Benchmarks

For deterministic CP and for QCP, set predictors are obtained from (5) and (15), respectively by disregarding the input x. Accordingly, we denote the corresponding set predictors as $\Gamma(\boldsymbol{\mathcal{D}}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$ and $\boldsymbol{\Gamma}_M(\boldsymbol{\mathcal{D}}^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})$, respectively. As benchmarks, we consider an ideal support estimator and a naïve support predictor based on a pre-trained PQC.

The *smallest support set with coverage* $1-\alpha$ is given by

$$\Gamma^{\text{opt}} = \arg\min_{\Gamma \in 2^{\mathcal{Y}}} |\Gamma| \text{ s.t. } \int_{y\in\Gamma} p^*(y)\mathrm{d}y \geq 1-\alpha. \tag{28}$$

This is depicted in Fig. 7-(*i*) using a gray shaded area. This set offers an idealized solution, and it cannot be evaluated in practice, since the ground-truth distribution $p^*(y)$ is not known.

Consider now a trained PQC that implements an implicit probabilistic model $p(y|\theta_{\mathcal{D}})$, where $\theta_{\mathcal{D}}$ is the parameter vector optimized based on data set $\mathcal{D}$. Replacing the ground-truth distribution $p^*(y)$ with the trained probabilistic model

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

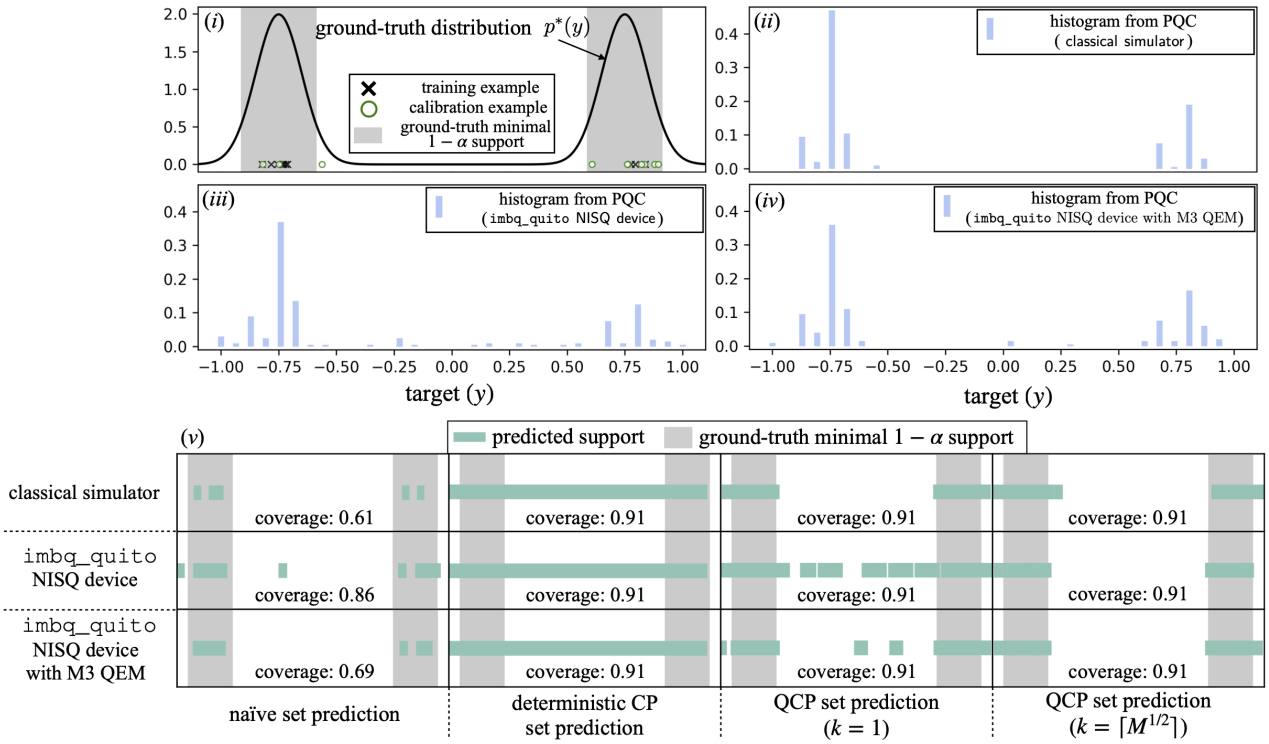


FIGURE 7. Illustration of the unsupervised learning problem of support estimation, in which the goal is to use the samples drawn from the PQC, as well as additional calibration data $\mathcal{D}^{\text{cal}}$, to estimate the support at coverage level $1-\alpha$ of the ground-truth probability density $p^*(y)$. (*i*) Ground-truth (unknown) distribution $p^*(y)$ and smallest support set (28) with coverage $1-\alpha=0.9$ (gray area), along with $|\mathcal{D}^{\text{tr}}|=10$ training examples (crosses) and $|\mathcal{D}^{\text{cal}}|=10$ calibration examples (circles); Histogram of the $M=200$ samples $\hat{\mathbf{y}}^{1:M}$ obtained via a PQC trained using the data set shown in part (*i*) and implemented via: (*ii*) a classical simulator; (*iii*) the `imbq_quito` NISQ device made available through IBM Quantum; and (*iv*) the `imbq_quito` NISQ device with M3 quantum error mitigation (QEM) [50]. (*v*) Predicted intervals produced by the naïve set predictor (29), by deterministic CP, and by QCP ($k=1$ and $k=\lceil M^{1/2}\rceil$) (15) for one realization of the calibration data set. The estimated empirical coverage probability for the set predictors is also indicated.

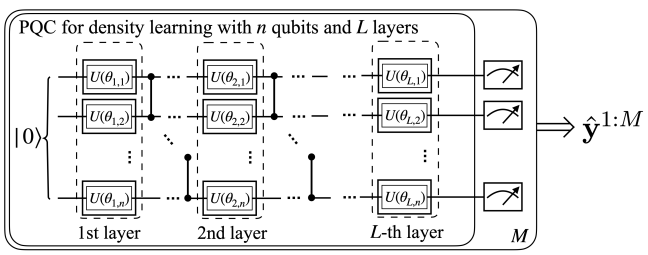


FIGURE 8. Illustration of the assumed hardware-efficient ansatz for density learning.

$p(y|\theta_{\mathcal{D}})$ in (28), a *naïve support predictor* can be obtained as

$$\mathbf{\Gamma}_M^{\text{naïve}}(\theta_{\mathcal{D}}) = \arg\min_{\Gamma\in 2^{\mathcal{Y}}} |\Gamma| \text{ s.t. } \int_{y\in\Gamma} p(y|\theta_{\mathcal{D}})\mathrm{d}y \geq 1-\alpha. \tag{29}$$

To evaluate the set in (29), samples $\hat{\mathbf{y}}^{1:M}$ drawn from the trained model $p(y|\theta_{\mathcal{D}})$ are used to approximate the integral in (29) via Monte Carlo integration.

2) PQC Ansatz

We adopt the standard *hardware-efficient ansatz*, which has also been previously used for the related unsupervised learning task of training generative quantum models [21], [22], [52]. As shown in Fig. 8, the parameterized unitary matrix $U(\theta)$ operates on a register of n qubits, which are initially in the fiducial state $|0\rangle$. The parameterized unitary matrix $U(\theta)$ consists of L layers applied sequentially as

$$U(\theta) = U_L(\theta)\cdot U_{L-1}(\theta)\cdots U_1(\theta), \tag{30}$$

with $U_l(\theta)$ being the unitary matrix corresponding to the l-th layer.

Following the hardware-efficient ansatz, the unitary $U_l(\theta)$ can be written as (see, e.g., [3, Sec. 6.4.2])

$$U_l(\theta) = U_{ent}(R(\theta_{l,1}^1,\theta_{l,1}^2,\theta_{l,1}^3)\otimes\cdots\otimes R(\theta_{l,n}^1,\theta_{l,n}^2,\theta_{l,n}^3)), \tag{31}$$

where U_{ent} is an *entangling unitary*, while

$$R(\theta^1,\theta^2,\theta^3) = R_Z(\theta^1)R_Y(\theta^2)R_Z(\theta^3) \tag{32}$$

represents a general parameterized single-qubit gate with Pauli rotations $R_P(\theta)=\exp(-i\frac{\theta}{2}P)$ for $P\in\{Y,Z\}$. By (31), the parameter vector θ is the collection of the angles $[\theta_{l,k}^1,\theta_{l,k}^2,\theta_{l,k}^3]$ for all k-th qubits and all l-th layers, i.e., $\theta=\{\{\theta_{l,k}^1,\theta_{l,k}^2,\theta_{l,k}^3\}_{k=1}^n\}_{l=1}^L$.

Given the PQC described above, for a fixed model parameter vector θ, M predictions $\hat{\mathbf{y}}^{1:M}$ are obtained by measuring a quantum observable O under the quantum state

$$\rho(\theta) = U(\theta)|0\rangle\langle 0|U(\theta)^\dagger \tag{33}$$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

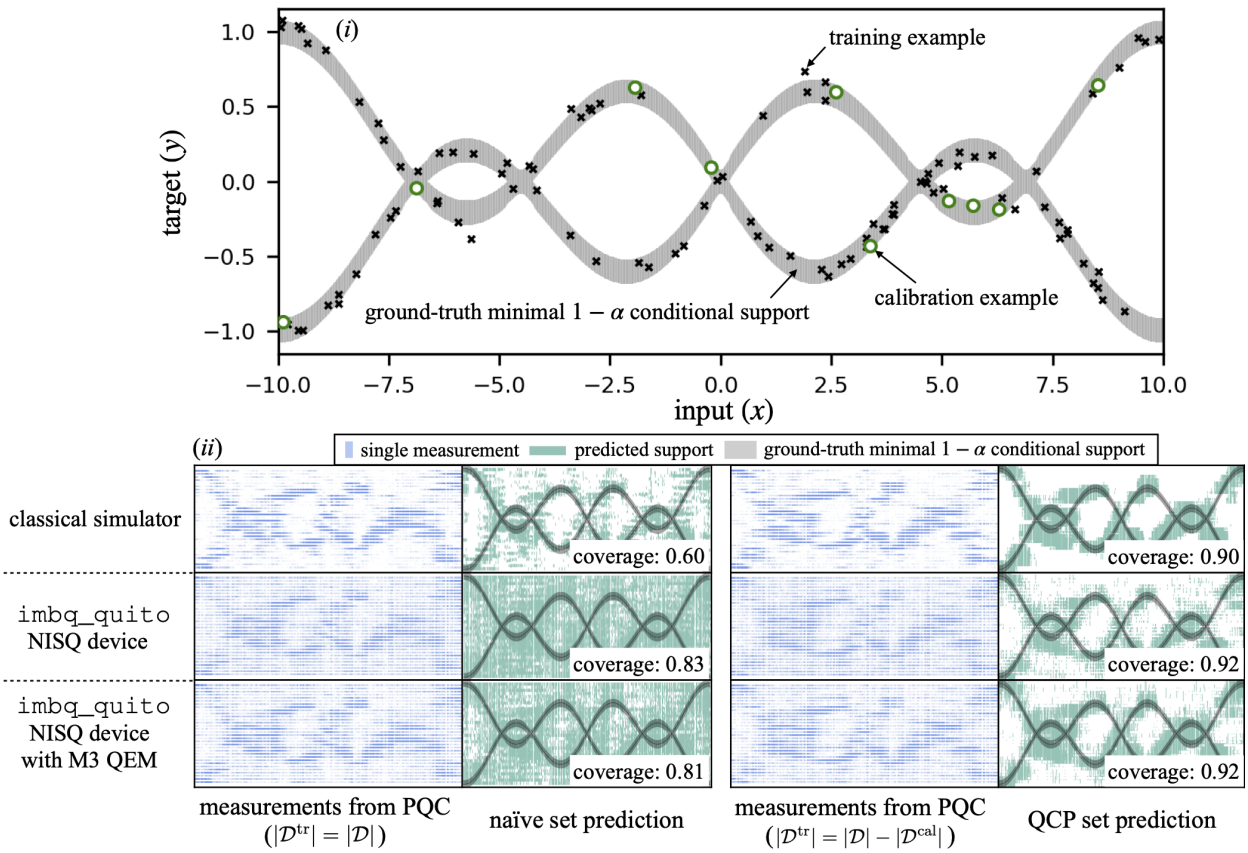


FIGURE 9. Illustration of the supervised learning problem of regression under study, in which the goal is to use the samples drawn from the PQC, as well as additional calibration data $\mathcal{D}^{\text{cal}}$, in order to produce a subset of predicted values at coverage probability $1-\alpha$ for the target y given test input x. (*i*) Ground-truth (unknown) minimal $1-\alpha$ conditional support (35) (gray area), along with $|\mathcal{D}^{\text{tr}}| = 90$ training examples (crosses) and $|\mathcal{D}^{\text{cal}}| = 10$ calibration examples (circles); (*ii*) First and third columns: $M = 100$ measurements obtained from a PQC with trained model based on the entire data set, $\theta_{\mathcal{D}}$ (first column) and based only on the data partition, $\theta_{\mathcal{D}^{\text{tr}}}$ (third column), with the PQC being implemented either using a classical simulator or the `imbq_quito` NISQ device made available through IBM Quantum with or without M3 QEM [50]. (*ii*) Second and fourth columns: Predicted intervals produced by the naïve set predictor (36) and by QCP (15) for one realization of the calibration data set. The estimated empirical coverage probability via naïve set prediction and QCP set prediction is also indicated.

produced as the output of the PQC. Measurements are implemented in the computational basis, i.e., with projection matrices $\Pi_j = |j\rangle\langle j|$, denoting as $|j\rangle$ the one-hot amplitude vector with 1 at position j for $j = 1, ..., N$. We choose the possible values $\{o_j\}_{j=1}^N$ to be equally spaced in the interval $[-1, 1]$. Accordingly, we set the j-th eigenvalue of the observable O as $o_j = -1 + 2(j-1)/(N-1)$.

We adopt a PQC with $n = 5$ qubits, and the entangling unitary consists of controlled-Z (CZ) gates connecting successive qubits, i.e., $U_{ent} = \prod_{k=1}^{n-1} C^Z_{k,k+1}$, where $C^Z_{k,k+1}$ is the CZ gate between the k-th qubit and $(k+1)$-th qubit. We set number of layers to $L = 2$. We implemented the described PQC on (*i*) a classical simulator; (*ii*) on the `imbq_quito` NISQ device made available through IBM Quantum; and (*ii*) on the `imbq_quito` NISQ device with M3 quantum error mitigation [50].

B. SUPERVISED LEARNING: REGRESSION

In the regression problem under study, we aim at predicting a scalar continuous-valued target y given scalar input x, with input and output following the unknown ground-truth joint distribution $p^*(x, y)$. We assume access to a data set $\mathcal{D} = \{z[i] = (x[i], y[i])\}_{i=1}^{|\mathcal{D}|}$, with training samples $z[i] \sim p^*(x, y)$ drawn in an i.i.d. manner.

We specifically consider the mixture of two sinusoidal functions for the ground-truth distribution, as given by

$$p^*(y|x) = \frac{1}{2}\mathcal{N}(\mu(x), 0.05^2) + \frac{1}{2}\mathcal{N}(-\mu(x), 0.05^2)$$
$$\text{and } p^*(x) = \mathcal{U}(-10, 10), \tag{34}$$

where we have denoted as $\mathcal{U}(a, b)$ the uniform distribution in the interval $[a, b]$, and we set $\mu(x) = 0.5\sin(0.8x) + 0.05x$. We note that bimodal distributions such as (34) cannot be represented by standard classical machine learning models used for regression that assume a Gaussian conditional distribution $p(y|x, \theta)$ with a parameterized mean [53]–[55].

1) Benchmarks

Benchmarks for regression generalize the approaches described in Sec. VI-A1 for unsupervised learning. Accordingly, the *smallest prediction interval with coverage* $1-\alpha$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

for input x is given by

$$\Gamma^{\text{opt}}(x) = \arg\min_{\Gamma\in 2^{\mathcal{Y}}} |\Gamma| \quad \text{s.t.} \int_{y\in\Gamma} p^*(y|x)\mathrm{d}y \geq 1-\alpha. \tag{35}$$

This is illustrated in Fig. 9-(*i*) as a gray shaded area. Since this ideal interval cannot be computed, a *naïve set predictor* alternative is to replace the ground-truth distribution $p^*(y|x)$ with a model $p(y|x,\theta_{\mathcal{D}})$ trained using the available data $\mathcal{D}$. This yields

$$\mathbf{\Gamma}_M^{\text{naïve}}(x|\theta_{\mathcal{D}}) = \arg\min_{\Gamma\in 2^{\mathcal{Y}}} |\Gamma| \quad \text{s.t.} \int_{y\in\Gamma} p(y|x,\theta_{\mathcal{D}})\mathrm{d}y \geq 1-\alpha, \tag{36}$$

where the integral is approximated via Monte Carlo integration using samples $\hat{\mathbf{y}}^{1:M}$ drawn from the model $p(y|x,\theta_{\mathcal{D}})$.

2) PQC Ansatz

As for the problem of density estimation, we adopt the *hardware-efficient ansatz*, with the key difference that the parameterized unitary matrix $U(x|\theta)$ also encodes the input x. In this regard, we follow the *data reuploading* strategy introduced in [56], which encodes the input x across all layers of the PQC architecture. This has been shown to offer significant advantages in terms of model expressivity [56]. We explore three different data encoding solutions in order of complexity: (*i*) conventional *fixed angle encoding* [57]; (*ii*) *learned linear angle encoding* as studied in [56]; and (*iii*) *learned non-linear angle encoding*, which appears not to have been considered before and may be of independent interest.

The parameterized unitary matrix $U(x|\theta)$ with L layers is defined as (cf. (30))

$$U(x|\theta) = U_L(x|\theta)\cdot U_{L-1}(x|\theta)\cdots U_1(x|\theta), \tag{37}$$

with the unitary matrix $U_l(x|\theta)$ for each l-th layer modelled as (cf. (31))

$$U_l(x|\theta) = U_{ent}(R(\phi^1_{l,1},\phi^2_{l,1},\phi^3_{l,1})\otimes\cdots\otimes R(\phi^1_{l,n},\phi^2_{l,n},\phi^3_{l,n})), \tag{38}$$

with U_{ent} and $R(\phi^1,\phi^2,\phi^3)$ defined as in Sec. VI-A2. Unlike Sec. VI-A2, however, the three angles $[\phi^1_{l,k},\phi^2_{l,k},\phi^3_{l,k}]$ for k-th qubit at the l-th layer may encode information about the input x. Note that we did not indicate this dependence explicitly in the notation in order to avoid clutter. We set $L=5$.

Angle encoding can be generally expressed with a parameterized (classical) function $f(\cdot|\theta_{l,k}):\mathcal{X}\to[0,2\pi]\times[0,2\pi]\times[0,2\pi]$ that takes as input x and outputs three angles. The function is parameterized by a vector $\theta_{l,k}$, and is generally written as

$$f(x|\theta_{l,k}) = [\phi^1_{l,k},\phi^2_{l,k},\phi^3_{l,k}]. \tag{39}$$

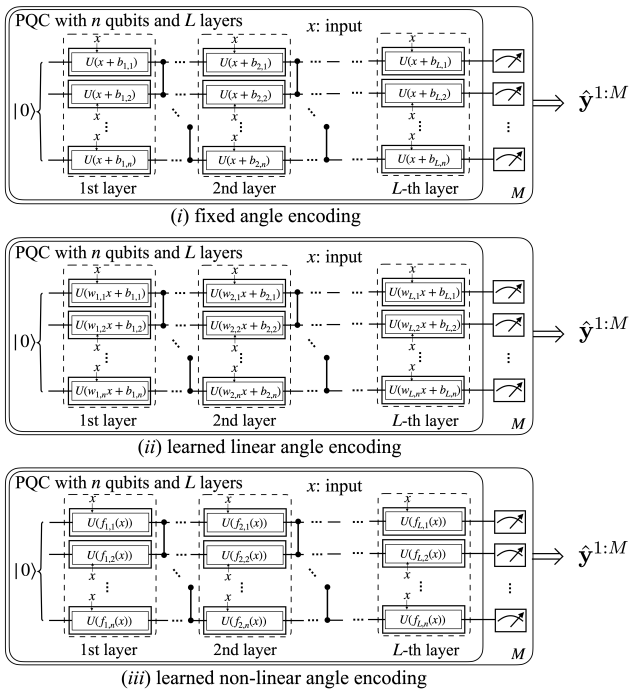


FIGURE 10. Illustration of the considered angle encoding strategies for the hardware-efficient ansatz adopted for the regression problem: *(i)* fixed angle encoding [57] has no trainable parameters for the data encoding block; *(ii)* learned linear angle encoding assumes trainable linear weight parameters that are multiplied with the input x [56]; *(iii)* learned *non-linear* angle encoding strategy maps the input x to the gate-controlling angles via a classical neural network.

Accordingly, the parameter vector θ contains n parameter vectors $\theta_{l,k}$ for each l-th layer, i.e., we have $\theta = \{\{\theta_{l,k}\}_{k=1}^n\}_{l=1}^L$. As mentioned, we consider three types of encoding functions, which are listed in order of increased generality.

Denoting as $f^i(x|\theta_{l,k})$ the i-th output of the function $f(x|\theta_{l,k})$ in (39), *conventional angle encoding* sets [56]

$$\phi^i_{l,k} = f^i(x|\theta_{l,k}) = x + b^i_{l,k}, \tag{40}$$

where the model parameter vector encompasses the scalar biases $\theta_{l,k} = \{b^i_{l,k}\}_{i=1}^3$.

Learning linear angle encoding sets the function $f(x|\theta_{l,k})$ as [56]

$$\phi^i_{l,k} = f^i(x|\theta_{l,k}) = w^i_{l,k}x + b^i_{l,k}, \tag{41}$$

with parameters $\theta_{l,k} = \{w^i_{l,k}, b^i_{l,k}\}_{i=1}^3$ containing scalar weights and scalar biases.

Finally, *learned non-linear angle encoding* implements function $f(x|\theta_{l,k})$ as a multi-layer neural network. In this case, the parameter vector $\theta_{l,k}$ includes the synaptic weights and biases of the neural network. In particular, we consider a fully connected network with input x followed by two hidden layers with 10 neurons having exponential linear unit (ELU) activation in each layer, that outputs the $3nL$ angles $\{\{\phi^1_{l,k},\phi^2_{l,k},\phi^3_{l,k}\}_{k=1}^n\}_{l=1}^L$ for the PQC, i.e., $[\phi^1_{1,1},\phi^2_{1,1},\phi^3_{1,1},\phi^1_{1,2},...,\phi^3_{1,n},\phi^1_{2,1},...,\phi^3_{L,n}]$. We note that non-parametric non-linear angle encoding [3, Sec. 6.8.1] and

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

exponential angle encoding [58] are also non-linear forms of angle encoding, which can be approximated via a suitable choice of the neural network function $f(x|\theta_{l,k})$ [59].

C. TRAINING

In this subsection, we elaborate on the implementation of training for the PQCs described in the previous subsections. We first note that the naïve predictors (29) and (36) use the entire data set $\mathcal{D}$ for training, while QCP splits data set $\mathcal{D}$ into a training set $\mathcal{D}^{\text{tr}}$ and a calibration set $\mathcal{D}^{\text{cal}}$. We adopt an equal split, with $|\mathcal{D}^{\text{tr}}| = |\mathcal{D}|/2$ training examples and $|\mathcal{D}^{\text{cal}}| = |\mathcal{D}|/2$ calibration examples, unless specified otherwise. For the rest of this subsection, we write the data used for training as $\mathcal{D}^{\text{tr}}$, with the understanding that for naïve schemes, this set is intended to be the overall set $\mathcal{D}$. We focus on the more general supervised learning setting, for which each training data example is a pair of input x^{tr} and output y^{tr}, since for unsupervised learning it is sufficient to remove the input x^{tr}.

Given a training data set $\mathcal{D}^{\text{tr}}$, the PQC model parameter vector $\theta_{\mathcal{D}^{\text{tr}}}$ is trained using Adam [60] based on gradients evaluated via automatic differentiation in PyTorch [61] on a classical simulator. When training on a quantum hardware device, the parameter-shift rule [27] is utilized to update the parameters of the PQC. In the rest of this subsection, we detail the loss functions used for the training of deterministic predictors (8) and implicit probabilistic models (13).

1) Training PQCs as Deterministic Predictors

When the PQC is used as a deterministic predictor, the prediction is given by $\hat{y} = \langle O \rangle_{\rho(x|\theta)}$ as in (8). For this case, we adopt the standard *quadratic loss*, yielding the empirical risk minimization problem

$$\theta_{\mathcal{D}^{\text{tr}}} = \arg\min_{\theta} \sum_{i=1}^{|\mathcal{D}^{\text{tr}}|} (y^{\text{tr}}[i] - \langle O \rangle_{\rho(x^{\text{tr}}[i]|\theta)})^2. \quad (42)$$

2) Training PQCs as Implicit Probabilistic Models

When the PQC is used as an implicit probabilistic predictor, with distribution $p(\hat{\mathbf{y}} = o_j|x,\theta) = \text{Tr}(\Pi_j \rho(x|\theta))$ as in (13), we adopt the cross-entropy loss. To this end, we first quantize the true label y of an example (x, y) so that the quantization levels match the possible values $\{o_j\}_{j=1}^{N'}$ obtained from the measurement of the given observable $O = \sum_{j=1}^{N'} o_j \Pi_j$. Then, the empirical risk minimization problem tackled during training is written as

$$\theta_{\mathcal{D}^{\text{tr}}} = \arg\min_{\theta} \sum_{i=1}^{|\mathcal{D}^{\text{tr}}|} (-\log \text{Tr}(\Pi_{j_i} \rho(x^{\text{tr}}[i]|\theta))), \quad (43)$$

where $j_i \in \{1, ..., N'\}$ is the index of the eigenvalue o_{j_i} that is closest (in Euclidean distance) to the i-th training output $y^{\text{tr}}[i]$.

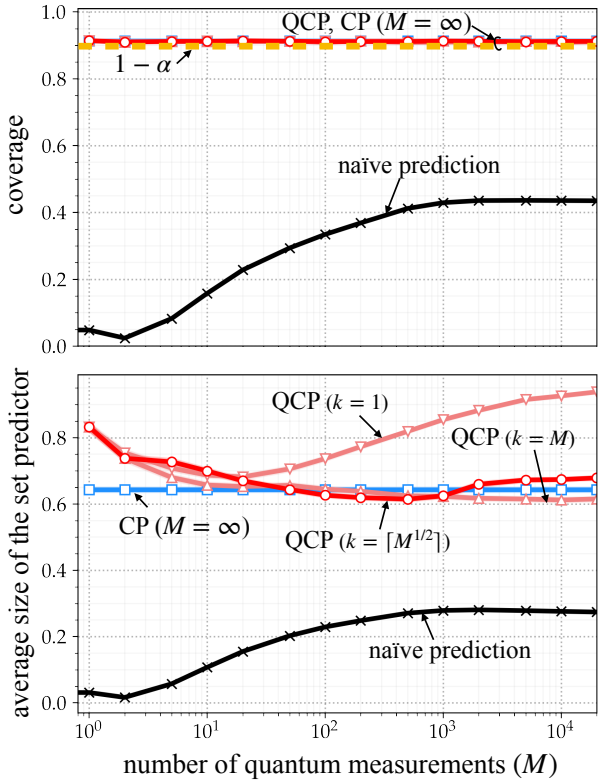


FIGURE 11. Density estimation with a weakly bimodal ground-truth Gaussian distribution: Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 20$ available training samples. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values.

VII. EXPERIMENTAL RESULTS

This section describes experimental results for both the unsupervised learning and supervised learning settings introduced in the previous section.

A. UNSUPERVISED LEARNING: DENSITY LEARNING

As explained in the previous section, we compare the performance of deterministic CP (Sec. II), which requires an arbitrarily large number of shots, with the naïve predictor (29), and with the proposed QCP scheme (Algorithm 1), in terms of their coverage probability and of the average size of the predicted set. These metrics are evaluated using $K = 1000$ experiments as the averages $\hat{\mathbf{P}}^{\text{supp}} = \frac{1}{K}\sum_{k=1}^{K} \int_{y \in \Gamma(\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})} p^*(y) \text{d}y$ and $\frac{1}{K}\sum_{k=1}^{K} |\Gamma(\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})|$ for deterministic CP; and as the averages $\hat{\mathbf{P}}_M^{\text{supp}} = \frac{1}{K}\sum_{k=1}^{K} \int_{y \in \mathbf{\Gamma}_M(\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})} p^*(y) \text{d}y$ and $\frac{1}{K}\sum_{k=1}^{K} |\mathbf{\Gamma}_M(\boldsymbol{\mathcal{D}}_k^{\text{cal}}, \theta_{\mathcal{D}^{\text{tr}}})|$ for QCP.

1) A Visual Comparison

Fig. 7 presents a visual comparison of the predicted sets produced by the different techniques with $|\mathcal{D}| = 20$ examples, assuming $M = 200$ measurements from the PQC. Specifically, panel (v) in the figure displays examples of predicted sets obtained with the training data and with the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

realizations of the calibration data set shown in panel (*i*) for $1-\alpha = 0.9$. Deterministic CP is observed to fail when faced with a bimodal ground-truth distribution (shown in panel (*i*)). This is because it makes the underlying design assumption of a single-modal likelihood, as implied by the conventional choice of the quadratic loss (42). As a result, deterministic CP produces inefficient set predictors. In contrast, a naïve set predictor tends to underestimate the coverage, since the samples produced by the trained PQC are excessively concentrated around the modes of the two Gaussians as seen in panels (*ii*)-(*iv*).

Unlike these baselines, the proposed QCP scheme provably meets the desired coverage level of $1-\alpha = 0.9$ (see Theorem 3). Furthermore, as observed in the last two columns of panel (*v*), it does so by producing efficient predicted sets, as long as the value of the parameter k used in the proposed QCP set predictor in (21) is properly chosen. In this regard, it is observed that setting $k = 1$ as in the original PCP work [23] does not provide good performance in terms of efficiency of the set predictor in the presence of quantum hardware noise (see second row and third column of panel (*v*)), while setting $k = \lceil M^{1/2} \rceil = 15$ ensures efficient set predictors. This result suggests that increasing the value of k enhances the robustness of the k-NN density estimator in (18), with $k = 1$ yielding an excessive sensitivity to randomness due to shot and quantum noise.

2) Quantitative Results with a Noiseless Simulator

We now provide numerical evaluations of the coverage probability and of the average size of the set predictor as a function of the number of shots M produced by the PQC. As in the example above, we also show the performance of deterministic CP, which assumes $M = \infty$. We assume a data set of $|\mathcal{D}| = 20$ data points, a target miscoverage level $\alpha = 0.1$.

We start by assuming a *weakly* bi-modal ground-truth distribution $p^*(y)$, obtained as the mixture of Gaussians with similar means given their standard deviation. We specifically set $p^*(y) = \frac{1}{2}\mathcal{N}(-0.15, 0.1^2) + \frac{1}{2}\mathcal{N}(0.15, 0.1^2)$. Fig. 11 shows coverage and average size of the set predictor as a function of number of quantum measurements M. While CP-based approaches provably provide well-calibrated support estimators, the naïve predictor (29) fails to cover the $1-\alpha$ of the mass of the ground-truth distribution $p^*(y)$.

In terms of coverage, since the weakly bimodal distribution at hand can be well approximated by a unimodal Gaussian distribution, deterministic CP – labelled as "CP" in the figure – performs well. While using a finite number of samples, QCP performs comparably to CP as long as k is suitably chosen as $k = \lceil M^{1/2} \rceil$ and M is not too small. This is because, as illustrated in Fig. 7-(*ii*), the trained PQC produces samples also in low-density regions of the ground-truth distribution $p^*(y)$, and the probability of obtaining one or more of such samples increases with the number of samples, M.

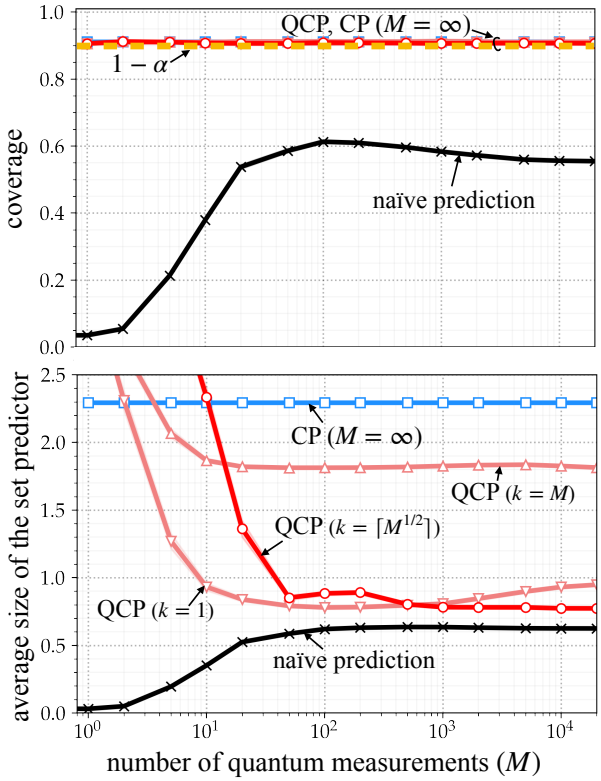


FIGURE 12. Density estimation with a strongly bimodal ground-truth Gaussian distribution: Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 20$ available training samples. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values.

We now consider a ground-truth Gaussian distribution that presents a *stronger* bimodality. To this end, we set $p^*(y) = \frac{1}{2}\mathcal{N}(-0.75, 0.1^2) + \frac{1}{2}\mathcal{N}(0.75, 0.1^2)$, and the results are illustrated in Fig. 12. While the conclusions in terms of coverage remain the same as in the previous example, comparisons in terms of average predicted set size reveal remarkably different behaviors of the considered predictors. In particular, despite its requirements in terms of number of shots, deterministic CP significantly underperforms QCP due to the underlying assumption of unimodality of the likelihood function. Using QCP with either $k = 1$ or $k = \lceil M^{1/2} \rceil$ yields a reduction by a factor of 3 in the size of the predicted set.

3) Quantitative Results with Quantum Hardware Noise

Finally, we provide a quantitative performance comparison based on results obtained on the `imbq_quito` NISQ device with or without M3 quantum error mitigation (QEM) [50]. As discussed in the previous section, for this case, training was done using a classical simulator, and the quantum computer was used solely for testing, i.e., to produce the density support estimate. Additional experimental results with PQC trained on `imbq_quito` NISQ device can be found in Appendix C. For reference, for this experiment, we show the performance of deterministic CP by using the empirical average of the measurements, i.e., $\frac{1}{M}\sum_{m=1}^{M} \hat{\mathbf{y}}^m$, in lieu of

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

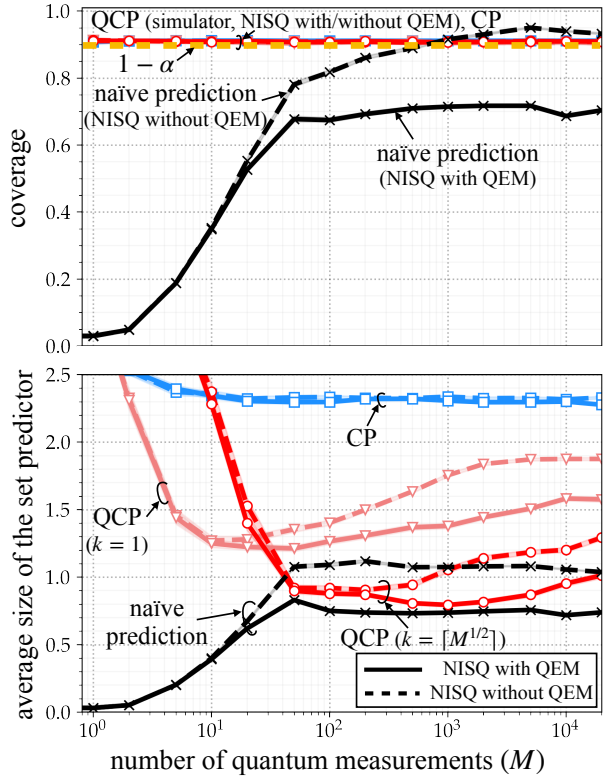


FIGURE 13. Density estimation with a strongly bimodal ground-truth Gaussian distribution: Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 20$ available training samples. Training is done on a classical simulator, while testing is implemented on `imbq_quito` NISQ device, with or without M3 QEM [50]. The shaded areas correspond to confidence intervals covering 95% of the realized values.

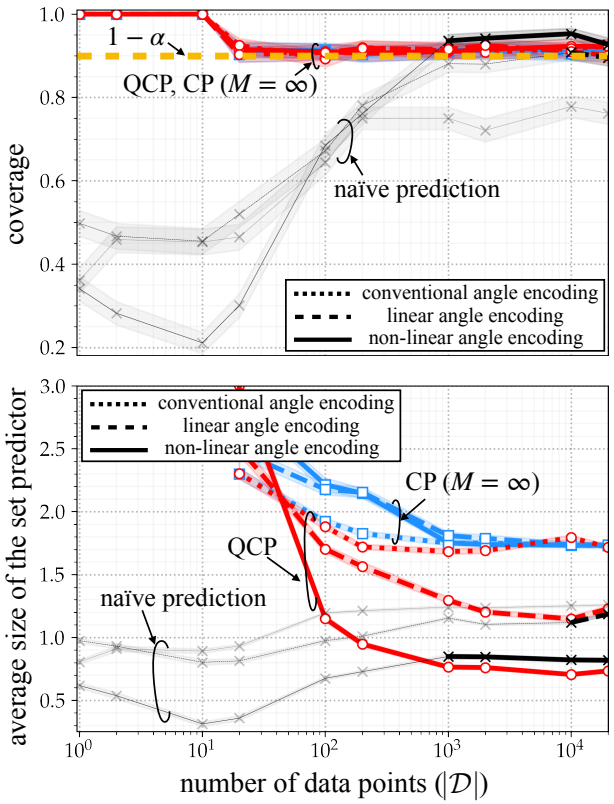


FIGURE 14. Regression for mixture of two sinusoidal functions (34): Coverage and average size of the set predictors as a function of the number $|\mathcal{D}|$ of available data points given $M = 1000$ quantum measurements. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 1000$ experiments, and the transparent lines are used for set predictors that do not meet the coverage level $1 - \alpha = 0.9$.

the true expectation $\hat{y} = \langle O \rangle_{\rho(x|\theta_{\mathcal{D}^{\text{tr}}})}$. This allows to report results for CP that depend on the number of shots, M.

Fig. 13 validates the conclusion reported in Sec. IV that QCP is provably well calibrated despite the presence of quantum hardware noise. In contrast, the naïve predictor is not well calibrated, even in the absence of quantum hardware noise (see Fig. 12). In this regard, the naïve predictor is observed to benefit from quantum hardware noise, achieving validity with a sufficiently large number of measurements $M \geq 10^3$. This can be understoood as a consequence of the fact that quantum hardware noise tends to produce samples that cover a wider range of output values [50], making it easier to cover a fraction $1 - \alpha$ of the mass of the original density $p^*(y)$ using the predictor (29).

In terms of average size of the prediction set, QCP with $k = 1$ is seen to be particularly sensitive to quantum hardware noise, as also anticipated with the examples in Fig. 7. In this case, increasing the number of shots can be deleterious as more noise is injected in the estimate of the predicted set. In contrast, with $k = \lceil M^{1/2} \rceil$, the QCP set predictor provides a significantly more robust performance to quantum hardware noise, even in the presence of QEM. Moreover, QEM is observed to improve the informativeness of the QCP set predictor by enhancing the quality of the underlying probabilistic predictor via the mitigation of quantum noise (see, e.g., [62]).

B. SUPERVISED LEARNING: REGRESSION

In this subsection, we turn to the supervised learning problem with ground-truth distribution (34). As done for unsupervised learning, we compare QCP (Sec. IV) against deterministic CP – in the ideal case of an infinite shots (Sec. II) – as well as against the naïve predictor (36) by evaluating coverage probability and average size of the set predictors using $K = 500$ experiments as discussed in Sec. V.

1) A Visual Comparison

Fig. 9 presents a visual comparison of the different set predictors assuming a data set of $|\mathcal{D}| = 100$ data points with $M = 100$ shots. We adopt learned non-linear angle encoding as described in Sec. VI-B2 with neural network composed of two hidden layers, each with 10 neurons having ELU activations. Panel (ii) in the figure depicts examples of predicted sets obtained with the training data and with the realizations of the calibration data set shown in panel (i) for $1 - \alpha = 0.9$. The naïve set predictor is observed to underestimate the support of the distribution. This can be

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

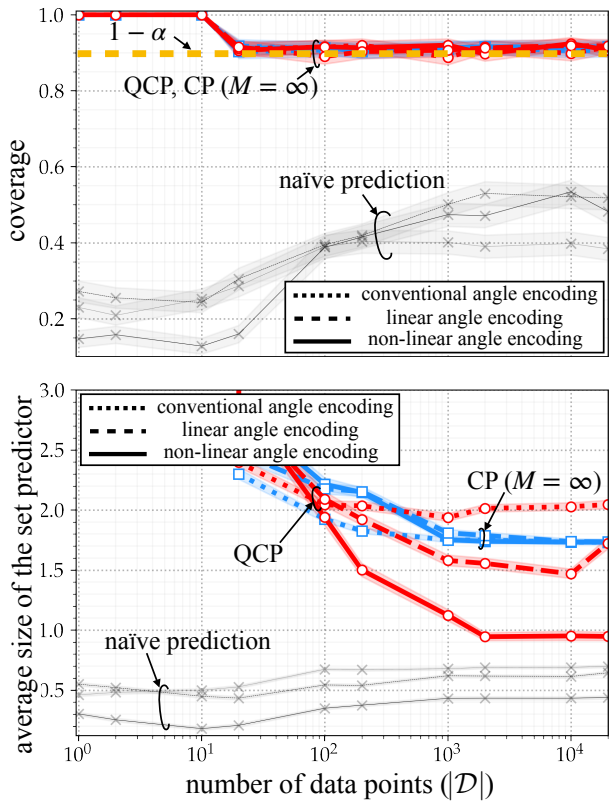


FIGURE 15. Regression for mixture of two sinusoidal functions (34): Coverage and average size of the set predictors as a function of the number $|\mathcal{D}|$ of available training samples given $M = 20$ quantum measurements. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 1000$ experiments, and the transparent lines are used for set predictors that do not meet the coverage level $1 - \alpha = 0.9$.

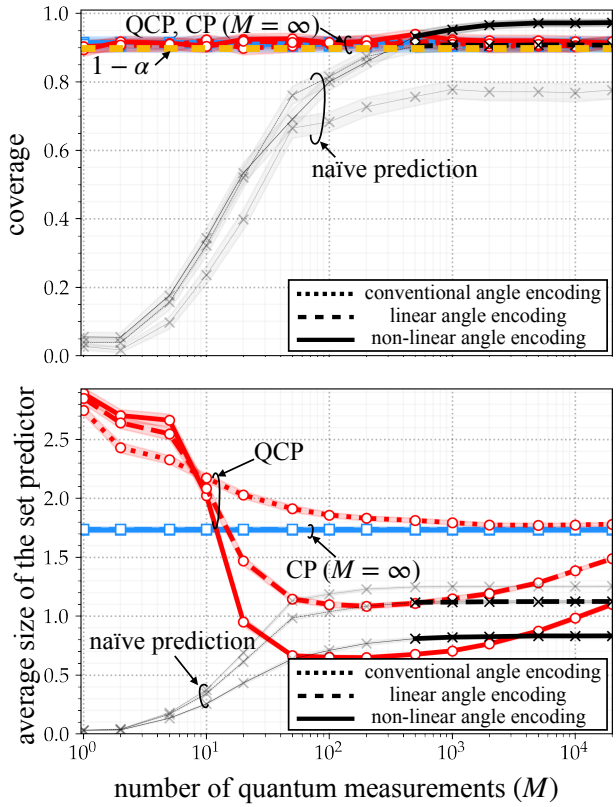


FIGURE 16. Regression for mixture of two sinusoidal functions (34): Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 10^4$ available training samples. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 1000$ experiments, and the transparent lines are used for set predictors that do not meet the coverage level $1 - \alpha = 0.9$.

interpreted in light of the typical overconfidence of trained predictors [1], [2], which causes the naïve predictor (36) to concentrate on a smaller subset of values as compared to the desired support, at level $1 - \alpha = 0.9$, of the ground-truth distribution. In contrast, QCP provably satisfies the predetermined coverage level of $1 - \alpha = 0.9$. Furthermore, as seen in the last column of panel (*ii*), it also produces efficient predicted sets with the choice $k = \lceil M^{1/2} \rceil$ in the set predictor (21).

2) Quantitative Results with a Noiseless Simulator

We now provide numerical evaluations of the coverage probability and of the average size of the set predictor as a function of the number $|\mathcal{D}|$ of data points for different angle encoding strategies as described in Sec. VI-B2. For learning non-linear angle encoding, we adopt the same architecture described above. For CP set predictors, if $|\mathcal{D}| \leq 20$, we split the data set $\mathcal{D}$ in equal parts for training and calibration; while, when $|\mathcal{D}| > 20$, we fix the number of calibration examples to $|\mathcal{D}^{\text{cal}}| = 10$ to use all the remaining data points for training. We assume a target miscoverage level $\alpha = 0.1$, and the reported values of coverage and average size of the set predictor are averaged over $K = 1000$ experiments as defined in Sec. V.

In Fig. 14 and Fig. 15, we show the coverage rate and average set size as a function of the number of data points, $|\mathcal{D}|$, with a large and small numbers of shots, namely $M = 1000$ and $M = 20$, respectively. In the first case, with a larger M, given enough training examples, here, for $|\mathcal{D}| \geq 10^4$, naïve prediction yields a well-calibrated set predictor that achieves $1 - \alpha$ coverage. This is the case when adopting either linear [56] or non-linear angle encoding. In contrast, naïve prediction fails to meet the coverage requirements for smaller data set size; and also for the smaller value of M irrespective of the data set size.

In line with their theoretical properties, deterministic CP and QCP are guaranteed to provide coverage at the desired level $1 - \alpha$, irrespective of data availability and number of shots. However, despite the use of an arbitrarily large number of shots, deterministic CP produces larger prediction set sizes than QCP. As in the case of unsupervised learning studied in the previous subsection, this is caused by the unimodality of the likelihood function assumed by deterministic CP. As an example, given $|\mathcal{D}| = 2000$ with non-linear angle encoding, when $M = 1000$, QCP yields set predictors with average size 0.76, while the average size of deterministic CP predictors

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

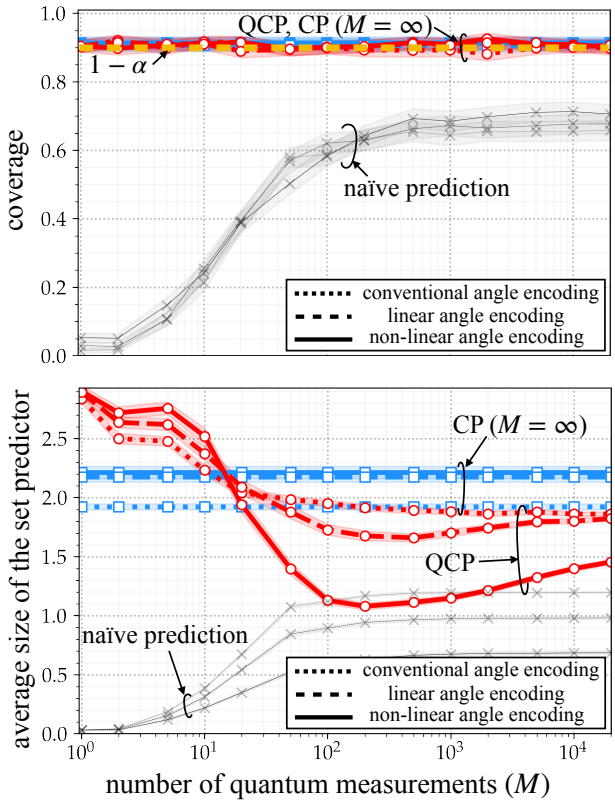


FIGURE 17. Regression for mixture of two sinusoidal functions (34): Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 100$ available training samples. Training and testing are done on a classical simulator. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 1000$ experiments, and the transparent lines are used for set predictors that do not meet the coverage level $1 - \alpha = 0.9$.

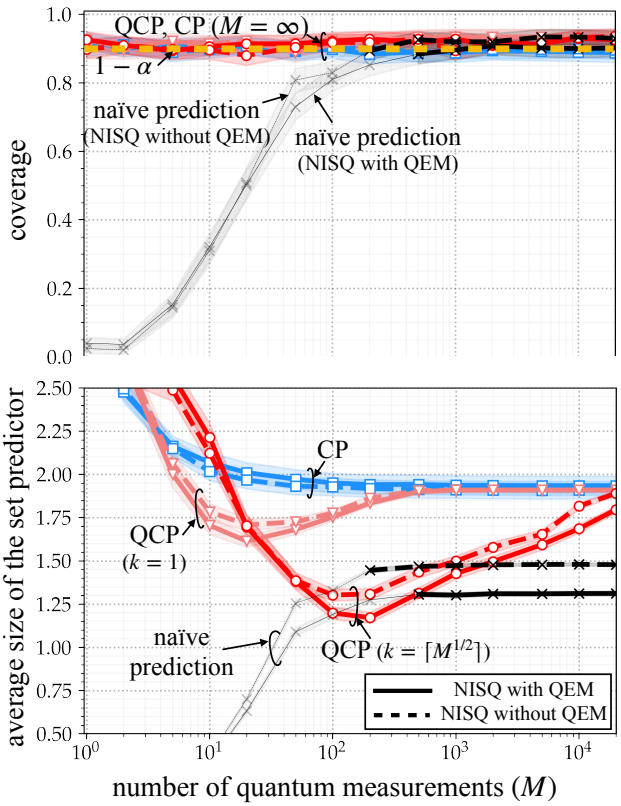


FIGURE 18. Regression for mixture of two sinusoidal functions (34): Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 100$ available training samples. Training is done on a classical simulator, while testing is implemented on `imbq_quito` NISQ device, with or without M3 QEM [50]. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 500$ experiments, and the transparent lines are used for set predictors that do not meet the coverage level $1 - \alpha = 0.9$.

is 1.74; and with $M = 20$ we obtain set size 1.74 with deterministic CP and 0.94 for QCP.

We now further investigate the impact of the number M of shots by focusing on regimes with abundant data, i.e., with $|\mathcal{D}| = 10^4$, and with limited data, i.e., $|\mathcal{D}| = 100$ in Fig. 16 and Fig. 17, respectively. In a manner that parallels the discussion in the previous paragraph on the role of the data set size, if $|\mathcal{D}|$ is sufficiently large, naïve set prediction achieves the desired coverage level as long as the number of shots is also sufficiently large, here $M \geq 500$ (Fig. 16). In contrast, CP schemes attain calibration in all regimes, with QCP significantly outperforming deterministic CP when M is not too small. As an example, given $M = 200$ with non-linear angle encoding, when $|\mathcal{D}| = 10^4$, QCP yields set predictors with average size 0.65, while the average size of deterministic CP predictors is 1.73; and with $|\mathcal{D}| = 100$ we obtain set size 2.21 with deterministic CP and 1.08 for QCP. Following the discussion in the previous subsection, an increase in the number of shots M does not necessarily translate in more efficient set predictors, as it becomes more likely for the PQC to draw outliers that cover low-density areas of the ground-truth distribution.

3) Quantitative Results with Quantum Hardware Noise

Moreover, we present a quantitative performance comparison based on results obtained on the `imbq_quito` NISQ device with or without M3 QEM. As discussed in the previous section, for this case, training was done using a classical simulator, and the quantum computer was used solely for testing, i.e., to produce the set prediction given a test input. In Fig. 18, we investigate the performance metrics as a function of number of quantum measurements M. QCP with parameter $k = \lceil M^{1/2} \rceil$ is confirmed to provide the best performance within a suitable range of values of M, significantly outperforming deterministic CP, even in the presence of quantum noise, with or without QEM.

C. QUANTUM DATA CLASSIFICATION

Finally, to demonstrate the validity of QCP when applied to quantum data, we consider the problem of classifying quantum states as described in Sec. IV-E. In particular, we aim at classifying $C = 10$ density matrices of size 16×16, assuming a uniform label probability $p(y) = 1/C$ for all $y \in \{1, ..., 10\}$. The density matrix $\rho(y)$ for each class y is expressed as the Gibbs state $\rho(y) = e^{-H(y)/T}/\text{Tr}(e^{-H(y)/T})$

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

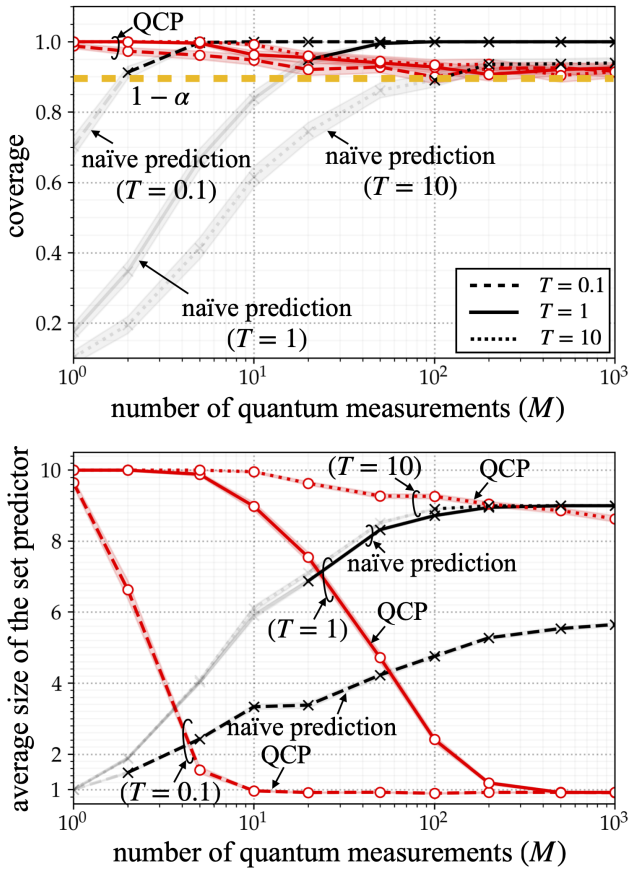


FIGURE 19. Quantum data classification: Coverage and average size of the set predictors as a function of the number M of quantum measurements given $|\mathcal{D}^{\text{cal}}| = 10$ calibration samples. The ten possible density matrices to be classified are generated as $\rho(y) = e^{-H(y)/T}/\text{Tr}(e^{-H(y)/T})$ with temperature $T > 0$, where the Hamiltonian matrices $H(y)$ are independently generated so as to ensure a sparsity level of 0.2 at temperature $T = 1$ as in [63]. Pretty good measurements detector is adopted [45]. The shaded areas correspond to confidence intervals covering 95% of the realized values. The results are averaged over $K = 1000$ experiments, and transparent lines are used to highlight regimes in which the set predictors do not meet the coverage level $1 - \alpha = 0.9$.

with temperature $T > 0$, where the Hamiltonian matrices $H(y)$ are independently generated so as to ensure a sparsity level of 0.2 at temperature $T = 1$ as in [63]. We adopt a pretty good measurement detector [45], [46] as the pre-designed POVM $\mathcal{P}$. We note that we could have also adopted PQC-based detectors designed using a number of copies of the training states in a manner similar to [24], since the validity condition (23) holds for any fixed detector.

In Fig. 19, we plot coverage and average size of the set predictor as a function of number of measurements M for QCP and for the naïve predictor. Note that an increased temperature T makes the classification problem more challenging since all the C density matrices $\{\rho(y)\}_{y=1}^{C}$ become increasingly close to the maximally mixed state. As per our theoretical results, QCP always achieves coverage no smaller than the predetermined level $1 - \alpha = 0.9$, while the naïve prediction fails to achieve validity unless it is supplied a sufficiently large number of quantum measurements M, i.e., $M \geq 100$ for temperature $T = 10$. In the regime of many shots, i.e., for $M \geq 100$, although both naïve and QCP set predictors are valid, the naïve set predictor tends to be extremely conservative, yielding predicted set that include 9 out of 10 labels for $T = 1$, while the set predictions output by QCP include on average a single class.

VIII. CONCLUSIONS

In this article, we have proposed a general methodology for quantum machine learning, referred to as *quantum conformal prediction* (QCP), that formally quantifies the uncertainty of the decisions made outside the training set. QCP provides "error bars" with coverage guarantees that do not rely on the amount of training data, on the number of shots, on the ansatz, on the training algorithm, and on the presence of quantum hardware noise. We have shown that a proper design of the scoring function used by QCP can significantly reduce the size of the error bars especially in the presence of quantum hardware noise, with or without quantum error mitigation. Experimental results confirmed that error bars produced by QCP contains the true target with desired coverage level, with up to three times smaller predictive set sizes as compared to existing baselines. Furthermore, it is concluded that, when quantum models are augmented with QCP, it is generally advantageous not to average over the shots, as typically done in the literature. Rather, treating the shots as separate samples allows QCP to obtain more informative error bars.

Future directions for research include the generalization of the QCP framework to more general form of risk control beyond coverage [64]–[66] and the investigation of conditional coverage guarantees [48], [67]. Furthermore, extensions to problems such as quantum tomography and unsupervised learning, as well as to settings with distributional shifts between calibration and test data [8], [68], [69], are interesting open problems.

APPENDIX A. QUANTUM GENERALIZATION BOUNDS

In this appendix, we provide some details on the generalization bounds plotted in Fig. 1-(*iii*). As discussed in Sec. I, the figure reports the generalization bounds derived in [6] as a function of the size of training data, $|\mathcal{D}^{\text{tr}}|$, for PQCs with $T = 1, 2, 5, 10$ trainable local quantum gates.

Let us fix a loss function $l(x, y|\theta)$ of the form $l(x, y|\theta) = \text{Tr}(O_y^{\text{loss}}\rho(x|\theta))$ for any example (x, y), where O_y is the *loss observable* for target variable value y. We recall that $\rho(x|\theta)$ is the density matrix that describes the state produced by the PQC with model parameters θ. As an example, choosing the loss observable as $O_y^{\text{loss}} = \sum_j \mathbb{1}(o_j \neq y)\Pi_j$, with $\Pi_j = |j\rangle\langle j|$, ensures that the loss function $l(x, y|\theta)$ measures the *probability of error*, i.e., the probability that the measurement output of the PQC is not equal to the label y.

We are interested in bounding the *generalization error*, which is given by the difference between the *population loss* $\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(x,y)}[l(\mathbf{x}, \mathbf{y}|\theta)]$, evaluated with respect to the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

ground-truth, unknown, distribution $p(x,y)$ and the training loss $1/|\mathcal{D}^{\text{tr}}|\sum_{(x,y)\in\mathcal{D}^{\text{tr}}} l(x,y|\theta)$. Reference [6] showed that the following bound holds with probability at least $1-\delta$ over the choice of training data set $\mathcal{D}^{\text{tr}}$ [6, Theorem 6]

$$\begin{aligned}\mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p(x,y)}[l(\mathbf{x},\mathbf{y}|\theta)]-\frac{1}{|\mathcal{D}^{\text{tr}}|}\sum_{(x,y)\in\mathcal{D}^{\text{tr}}} l(x,y|\theta) \qquad (44)\\ \leq \frac{24C_{\text{loss}}}{\sqrt{|\mathcal{D}^{\text{tr}}|}}\sqrt{512T}\Bigg(\frac{1}{2}\sqrt{\log(6T)}+\frac{1}{2}\sqrt{\log 2}\\ -\frac{\sqrt{\pi}}{2}\text{erf}(\sqrt{\log 2})+\frac{\sqrt{\pi}}{2}\Bigg)\\ +3C_{\text{loss}}\sqrt{\frac{2\log(2/\delta)}{|\mathcal{D}^{\text{tr}}|}},\end{aligned}$$

where T is the number of trainable gates in the PQC and C_{loss} is the maximum spectral norm of the loss observables O_y^{loss}. The error function in (44) is defined as $\text{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)\text{d}t$. Fig. 1-(*iii*) plots this bound for different values of $|\mathcal{D}^{\text{tr}}|$ and T by assuming the probability of error loss described above, which has $C_{\text{loss}}=1$. We choose $\delta=0.1$, but changes in δ have a negligible impact on the bound. We refer to Sec. I for discussions on the bound.

APPENDIX B. PROOFS OF THEOREMS 1-3

In order to prove Theorems 1-3 at once, we unify the expression for the scoring function as $t(z|\boldsymbol{\nu})$, where $\boldsymbol{\nu}$ is a context variable. This way, for deterministic CP (Sec. II-B), we have $t(z|\boldsymbol{\nu}) = s(z|\theta_{\mathcal{D}^{\text{tr}}})$ with the context being deterministic; for PCP, we have $t(z|\boldsymbol{\nu}) = s(z|\hat{\mathbf{y}}^{1:M})$ as in (10), with random variable $\boldsymbol{\nu}=\hat{\mathbf{y}}^{1:M}$ generated i.i.d. from the classical model $p(y|x,\theta_{\mathcal{D}^{\text{tr}}})$; and for QCP (Sec. IV-D), we have $t(z|\boldsymbol{\nu}) = s(z|\hat{\mathbf{y}}^{1:M})$ with random variable $\boldsymbol{\nu}=\hat{\mathbf{y}}^{1:M}$ generated by the PQC with trained model $\theta_{\mathcal{D}^{\text{tr}}}$ following either (13) or (14).

With this notation, the validity conditions proved by Theorems 1-3 can be expressed as

$$\begin{aligned}&\Pr(\mathbf{y}\in\Gamma(\mathbf{x}|\boldsymbol{\mathcal{D}}^{\text{cal}},\theta_{\mathcal{D}^{\text{tr}}}))\\ &=\Pr(t(\mathbf{z}[N+1]|\boldsymbol{\nu}[N+1])\leq Q_{1-\alpha}(\{t(\mathbf{z}[i]|\boldsymbol{\nu}[i])\}_{i=1}^N))\\ &\geq 1-\alpha, \qquad (45)\end{aligned}$$

where $t(\mathbf{z}[N+1]|\boldsymbol{\nu}[N+1])$ is the score for the test data $\mathbf{z}$. We recall that, given a set of real numbers $\{s[1],...,s[N],\infty\}$, the notation $Q_{1-\alpha}(\{s[i]\}_{i=1}^N)$ represents the $\lceil(1-\alpha)(N+1)\rceil$-th smallest value in the set. To prove the inequality (45), we introduce the following lemmas.

Lemma 1 (Exchangeability lemma)**.** *Assume that* $\mathbf{z}[1],...,\mathbf{z}[N+1]\in\mathcal{Z}$ *are exchangeable random variables. Furthermore, assume that the joint distribution of random variables* $(\mathbf{z}[1],\boldsymbol{\nu}[1]),...,(\mathbf{z}[N+1],\boldsymbol{\nu}[N+1])$ *can be written as*

$$\begin{aligned}&p((z[1],\nu[1]),...,(z[N+1],\nu[N+1]))\\ &=\prod_{i=1}^{N+1}p(\nu[i]|z[i])p(z[1],...,z[N+1]) \qquad (46)\end{aligned}$$

for some conditional distribution $p(\nu[i]|z[i])$ *that does not depend on* $i=1,...,N+1$*. Then, for any real-valued function* $t(z|\nu)$*, the random variables*

$$t(\mathbf{z}[1]|\boldsymbol{\nu}[1]),...,t(\mathbf{z}[N+1]|\boldsymbol{\nu}[N+1]) \qquad (47)$$

are exchangeable.

Proof. The result follows directly from the permutation-invariance of the distribution (46). □

Lemma 2 (Quantile lemma [8])**.** *If* $\mathbf{s}[1],...,\mathbf{s}[N],\mathbf{s}[N+1]$ *are exchangeable random variables, then for any* $\alpha\in(0,1)$*, the following inequality holds*

$$\Pr(\mathbf{s}[N+1]\leq Q_{1-\alpha}(\{\mathbf{s}[i]\}_{i=1}^N))\geq 1-\alpha. \qquad (48)$$

Proof. Defining $Q^*_{1-\alpha}(\{\mathbf{s}[i]\}_{i=1}^{N+1})$ as the $\lceil(1-\alpha)(N+1)\rceil$-th smallest value in the set $\{\mathbf{s}[1],...,\mathbf{s}[N],\mathbf{s}[N+1]\}$, we have the inequality

$$\Pr(\mathbf{s}[N+1]\leq Q^*_{1-\alpha}(\{\mathbf{s}[i]\}_{i=1}^{N+1}))\geq 1-\alpha, \qquad (49)$$

by the exchangeability of the random variables [70]. Furthermore, we have the following equivalence [8, Sec. A.1]

$$\begin{aligned}&\mathbf{s}[N+1]>Q_{1-\alpha}(\{\mathbf{s}[i]\}_{i=1}^N)\\ \Leftrightarrow&\mathbf{s}[N+1]>Q^*_{1-\alpha}(\{\mathbf{s}[i]\}_{i=1}^{N+1}), \qquad (50)\end{aligned}$$

which can be readily checked by noting that $\mathbf{s}[N+1]$ cannot be strictly larger than itself or of ∞.

□

Combining Lemma 1 and Lemma 2 for the random variables $\mathbf{s}[i]=t(\mathbf{z}[i]|\boldsymbol{\nu}[i])$, we obtain the desired condition (45).

APPENDIX C. ADDITIONAL EXPERIMENTS

A. IMPACT OF THE CHOICE OF PARAMETER k IN THE QCP SCORING FUNCTION (21)

To elaborate on the impact of the choice of parameter k in the k-NN density estimator adopted by QCP, we plot in Fig. 20 the coverage and average size of the QCP set predictor for the density learning problem (see Sec. VI-A) as a function of k given availability of $M=1000$ measurements. Recall that $k=1$ corresponds to the choice of scoring function assumed in PCP [23], while the selection $k=\lceil\sqrt{M}\rceil$ ensures consistency of the k-NN density estimator as summarized in Sec. IV-D. From Fig. 20 we conclude that the proposed scoring function (19) with the theoretically motivated choice $k=\lceil\sqrt{M}\rceil$ achieves nearly minimal average predicted set size, decreasing the set size by 57.25% as compared to the case $k=1$.

B. QCP WITH PQC TRAINED IN THE PRESENCE OF QUANTUM HARDWARE NOISE

In order to further verify that the reliability guarantees of QCP in Theorem 3 hold irrespective of the quality of the trained PQC, even trained in the presence of quantum hardware noise, in Fig. 21, we plot the coverage and average size of the set predictors for the problem of density estimation

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

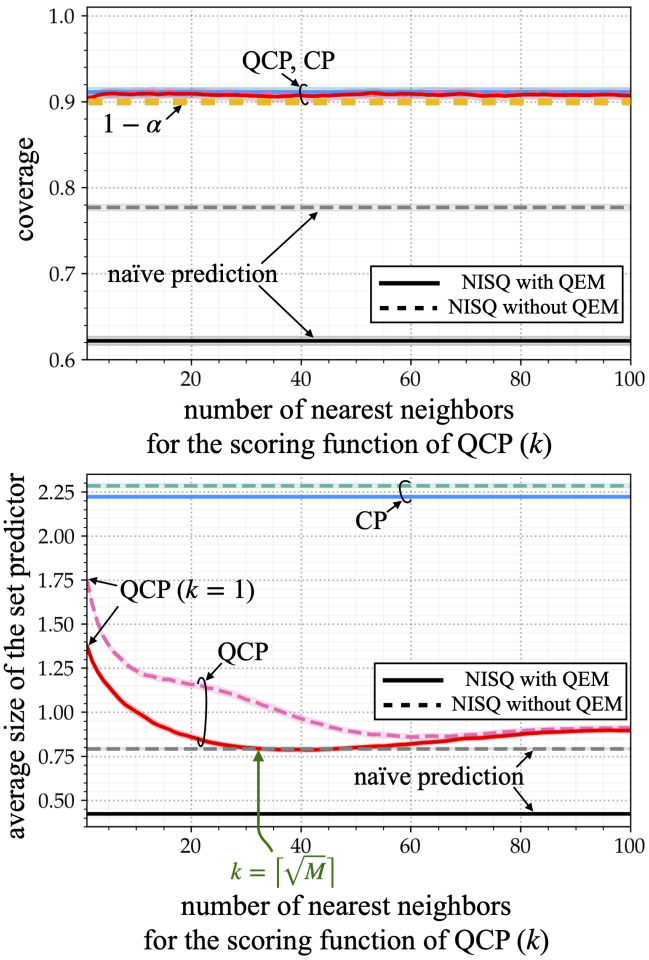


FIGURE 20. Density estimation with a strongly bimodal ground-truth Gaussian distribution: Coverage and average size of the set predictors as a function of k given $|\mathcal{D}| = 20$ available training samples. Training is done on a classical simulator, while testing is implemented on `imbq_quito` NISQ device, with or without M3 QEM [50]. The shaded areas correspond to confidence intervals covering 95% of the realized values.

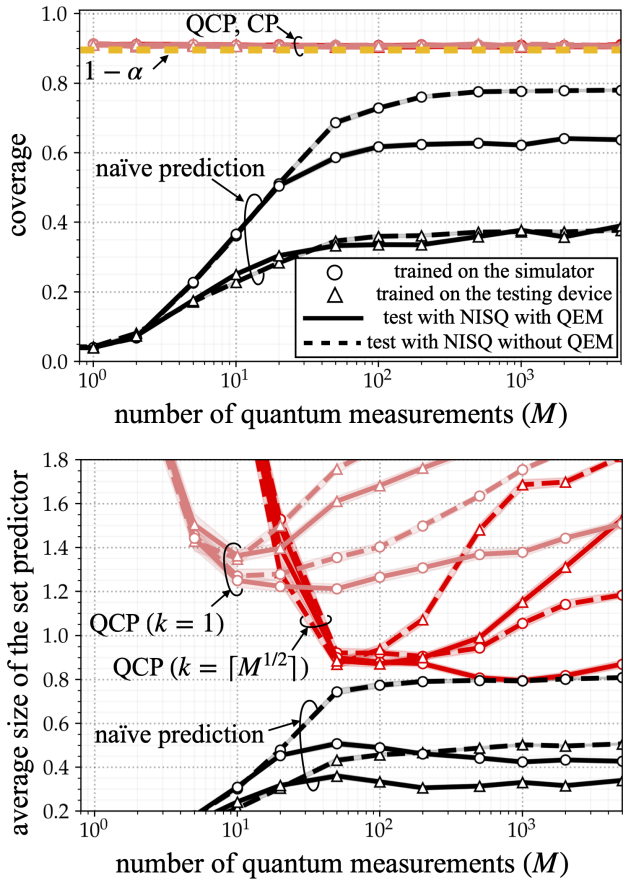


FIGURE 21. Density estimation with a strongly bimodal ground-truth Gaussian distribution: Coverage and average size of the set predictors as a function of number M of quantum measurements given $|\mathcal{D}| = 20$ available training samples. Training is done either on a classical simulator or on a `imbq_quito` NISQ device with or without M3 QEM [50]. When trained on the quantum device, the same QEM strategy is applied during the testing phase. The shaded areas correspond to confidence intervals covering 95% of the realized values.

using a PQC trained on `imbq_quito` NISQ device with or without QEM. The other settings are same as in Fig. 7. QCP is observed to guarantee reliability also for the model trained on the quantum computer. This is in contrast to the naïve set predictor, which only covers 40% of the support, falling far short of the target coverage level 90%.

REFERENCES

[1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[2] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*. Springer, 2021.

[3] O. Simeone, "An introduction to quantum machine learning for engineers," *Foundations and Trends® in Signal Processing*, vol. 16, no. 1-2, pp. 1–223, 2022.

[4] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[5] O. Simeone, *Machine Learning for Engineers*. Cambridge University Press, 2022.

[6] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, "Generalization in quantum machine learning from few training data," *Nature Communications*, vol. 13, no. 1, p. 4919, 2022.

[7] V. Vovk, A. Gammerman, and G. Shafer, *Algorithmic Learning in a Random World*. Springer Nature, 2022.

[8] R. J. Tibshirani, R. Foygel Barber, E. Candes, and A. Ramdas, "Conformal prediction under covariate shift," in *Proc. of Adv. in Neural Inf. Processing Sys. (NIPS)*, 2019.

[9] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Predictive inference with the jackknife+," *The Annals of Statistics*, vol. 49, no. 1, pp. 486–507, 2021.

[10] A. N. Angelopoulos and S. Bates, "A gentle introduction to conformal prediction and distribution-free uncertainty quantification," *arXiv preprint arXiv:2107.07511*, 2021.

[11] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *arXiv preprint arXiv:2210.00921*, 2022.

[12] L. Banchi, J. Pereira, and S. Pirandola, "Generalization in quantum machine learning: A quantum information standpoint," *PRX Quantum*, vol. 2, no. 4, p. 040321, 2021.

[13] S. T. Jose and O. Simeone, "Transfer learning in quantum parametric classifiers: An information-theoretic generalization analysis," *arXiv preprint arXiv:2201.06297*, 2022.

[14] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, "The power of quantum neural networks," *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.

[15] M. Weber, A. Anand, A. Cervera-Lierta, J. S. Kottmann, T. H. Kyaw, B. Li, A. Aspuru-Guzik, C. Zhang, and Z. Zhao, "Toward reliability in the NISQ

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

era: Robust interval guarantee for quantum measurements on approximate states," *Physical Review Research*, vol. 4, no. 3, p. 033217, 2022.

[16] D. A. McAllester, "PAC-Bayesian model averaging," in *Proc. of Annual Conf. Computational Learning Theory (COLT)*, July 1999, pp. 164–170.

[17] B. Guedj, "A primer on PAC-Bayesian learning," *arXiv preprint arXiv:1901.05353*, 2019.

[18] D. Russo and J. Zou, "Controlling bias in adaptive data analysis using information theory," in *Proc. of Artificial Intelligence and Statistics (AISTATS)*, May 2016, pp. 1232–1240.

[19] A. Xu and M. Raginsky, "Information-theoretic analysis of generalization capability of learning algorithms," in *Proc. of Adv. in Neural Inf. Processing Sys. (NIPS)*, Dec. 2017, pp. 2524–2533.

[20] S. Duffield, M. Benedetti, and M. Rosenkranz, "Bayesian learning of parameterised quantum circuits," *Machine Learning: Science and Technology*, vol. 4, no. 2, 2023.

[21] Y. Du, Z. Tu, B. Wu, X. Yuan, and D. Tao, "Theory of quantum generative learning models with maximum mean discrepancy," *arXiv preprint arXiv:2205.04730*, 2022.

[22] K. Gili, M. Mauri, and A. Perdomo-Ortiz, "Evaluating generalization in classical and quantum generative models," *arXiv preprint arXiv:2201.08770*, 2022.

[23] Z. Wang, R. Gao, M. Yin, M. Zhou, and D. M. Blei, "Probabilistic conformal prediction using conditional random samples," *arXiv preprint arXiv:2206.06584*, 2022.

[24] Y. Deville and A. Deville, "New single-preparation methods for unsupervised quantum machine learning problems," *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–24, 2021.

[25] C. M. Caves, C. A. Fuchs, and R. Schack, "Unknown quantum states: The quantum de Finetti representation," *Journal of Mathematical Physics*, vol. 43, no. 9, pp. 4537–4559, 2002.

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

[27] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Physical Review A*, vol. 99, no. 3, p. 032331, 2019.

[28] W. Lavrijsen, A. Tudor, J. Müller, C. Iancu, and W. De Jong, "Classical optimizers for noisy intermediate-scale quantum devices," in *Proc 2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 2020, pp. 267–277.

[29] A. Rad, A. Seif, and N. M. Linke, "Surviving the barren plateau in variational quantum circuits with bayesian learning initialization," *arXiv preprint arXiv:2203.02464*, 2022.

[30] E. Knill, "Quantum computing with realistically noisy devices," *Nature*, vol. 434, no. 7029, pp. 39–44, 2005.

[31] A. W. Harrow and M. A. Nielsen, "Robustness of quantum gates in the presence of noise," *Physical Review A*, vol. 68, no. 1, p. 012308, 2003.

[32] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles, "Noise resilience of variational quantum compiling," *New Journal of Physics*, vol. 22, no. 4, p. 043006, 2020.

[33] A. He, B. Nachman, W. A. de Jong, and C. W. Bauer, "Zero-noise extrapolation for quantum-gate error mitigation with identity insertions," *Physical Review A*, vol. 102, no. 1, p. 012426, 2020.

[34] J. Chow, L. DiCarlo, J. Gambetta, A. Nunnenkamp, L. S. Bishop, L. Frunzio, M. Devoret, S. Girvin, and R. Schoelkopf, "Detecting highly entangled states with a joint qubit readout," *Physical Review A*, vol. 81, no. 6, p. 062325, 2010.

[35] S. Bravyi, S. Sheldon, A. Kandala, D. C. Mckay, and J. M. Gambetta, "Mitigating measurement errors in multiqubit experiments," *Physical Review A*, vol. 103, no. 4, p. 042605, 2021.

[36] E. Van Den Berg, Z. K. Minev, and K. Temme, "Model-free readout-error mitigation for quantum expectation values," *Physical Review A*, vol. 105, no. 3, p. 032620, 2022.

[37] L. Schwarz and S. van Enk, "Detecting the drift of quantum sources: not the de Finetti theorem," *Physical Review Letters*, vol. 106, no. 18, p. 180501, 2011.

[38] W. Chen, K.-J. Chun, and R. F. Barber, "Discretized conformal prediction for efficient distribution-free inference," *Stat*, vol. 7, no. 1, p. e173, 2018.

[39] E. Parzen, "On estimation of a probability density function and mode," *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[40] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[41] D. O. Loftsgaarden, C. P. Quesenberry *et al.*, "A nonparametric estimate of a multivariate density function," *The Annals of Mathematical Statistics*, vol. 36, no. 3, pp. 1049–1051, 1965.

[42] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[43] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics." *Journal of Machine Learning Research*, vol. 13, no. 2, 2012.

[44] G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," *The Journal of Machine Learning Research*, vol. 22, no. 1, pp. 2617–2680, 2021.

[45] P. Hausladen and W. K. Wootters, "A 'pretty good'measurement for distinguishing quantum states," *Journal of Modern Optics*, vol. 41, no. 12, pp. 2385–2390, 1994.

[46] S. Gambs, "Quantum classification," *arXiv preprint arXiv:0809.0444*, 2008.

[47] J. Lei, M. G'Sell, A. Rinaldo, R. J. Tibshirani, and L. Wasserman, "Distribution-free predictive inference for regression," *Journal of the American Statistical Association*, vol. 113, no. 523, pp. 1094–1111, 2018.

[48] V. Vovk, "Conditional validity of inductive conformal predictors," in *Proc. Asian Conference on Machine Learning*. PMLR, 2012, pp. 475–490.

[49] G. Jowett, "The relationship between the binomial and F distributions," *Journal of the Royal Statistical Society. Series D (The Statistician)*, vol. 13, no. 1, pp. 55–57, 1963.

[50] P. D. Nation, H. Kang, N. Sundaresan, and J. M. Gambetta, "Scalable mitigation of measurement errors on quantum computers," *PRX Quantum*, vol. 2, no. 4, p. 040326, 2021.

[51] J. Lei, J. Robins, and L. Wasserman, "Efficient nonparametric conformal prediction regions," *arXiv preprint arXiv:1111.1418*, 2011.

[52] C. Zoufal, A. Lucchi, and S. Woerner, "Quantum generative adversarial networks for learning and loading random distributions," *npj Quantum Information*, vol. 5, no. 1, pp. 1–9, 2019.

[53] A. Masegosa, "Learning under model misspecification: Applications to variational and ensemble methods," in *Proc. of Adv. in Neural Inf. Processing Sys. (NIPS)*, 2020.

[54] W. R. Morningstar, A. Alemi, and J. V. Dillon, "PAC^m Bayes: Narrowing the empirical risk gap in the misspecified bayesian regime," in *Proc. of Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022.

[55] M. Zecchin, S. Park, O. Simeone, M. Kountouris, and D. Gesbert, "Robust PAC^m: Training ensemble models under model misspecification and outliers," *arXiv preprint arXiv:2203.01859*, 2022.

[56] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.

[57] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Physical Review A*, vol. 103, no. 3, p. 032430, 2021.

[58] M. Kordzanganeh, P. Sekatski, L. Fedichkin, and A. Melnikov, "An exponentially-growing family of universal quantum circuits," *Machine Learning: Science and Technology*, vol. 4, 2023.

[59] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural Networks*, vol. 4, no. 2, pp. 251–257, 1991.

[60] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[61] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Proc. of Adv. in Neural Inf. Processing Sys. (NIPS)*, 2019.

[62] G. S. Dhillon, G. Deligiannidis, and T. Rainforth, "On the expected size of conformal prediction sets," *arXiv preprint arXiv:2306.07254*, 2023.

[63] S. Ahmed, C. S. Munoz, F. Nori, and A. F. Kockum, "Classification and reconstruction of optical quantum states with deep neural networks," *Physical Review Research*, vol. 3, no. 3, p. 033278, 2021.

[64] S. Bates, A. Angelopoulos, L. Lei, J. Malik, and M. Jordan, "Distribution-free, risk-controlling prediction sets," *Journal of the ACM (JACM)*, vol. 68, no. 6, pp. 1–34, 2021.

[65] A. N. Angelopoulos, S. Bates, E. J. Candès, M. I. Jordan, and L. Lei, "Learn then test: Calibrating predictive algorithms to achieve risk control," *arXiv preprint arXiv:2110.01052*, 2021.

[66] A. N. Angelopoulos, S. Bates, A. Fisch, L. Lei, and T. Schuster, "Conformal risk control," *arXiv preprint arXiv:2208.02814*, 2022.

[67] M. Bian and R. F. Barber, "Training-conditional coverage for distribution-free predictive inference," *arXiv preprint arXiv:2205.03647*, 2022.

[68] M. Cauchois, S. Gupta, A. Ali, and J. C. Duchi, "Robust validation: Confident predictions even when distributions shift," *arXiv preprint arXiv:2008.04267*, 2020.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4

This article has been accepted for publication in IEEE Transactions on Quantum Engineering. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TQE.2023.3333224

[69] R. F. Barber, E. J. Candes, A. Ramdas, and R. J. Tibshirani, "Conformal prediction beyond exchangeability," *arXiv preprint arXiv:2202.13415*, 2022.

[70] A. K. Kuchibhotla, "Exchangeability, conformal prediction, and rank tests," *arXiv preprint arXiv:2005.06095*, 2020.

SANGWOO PARK (Member, IEEE) received his B.Sc. degree in physics in 2014; M.S.E and Ph.D. degrees in electrical engineering in 2016 and 2020, all from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea. From April to December 2019, he was a visiting Ph.D. student with King's Communications, Learning and Information Processing lab (KCLIP), King's College London (KCL), United Kingdom (UK). He is currently a research associate with KCLIP, KCL, UK. His research interests lie in practical, reliable AI and its application to engineering systems including wireless communications and quantum information processing.

OSVALDO SIMEONE (Fellow, IEEE) is a Professor of Information Engineering. He co-directs the Centre for Intelligent Information Processing Systems within the Department of Engineering of King's College London, where he also runs the King's Communications, Learning and Information Processing lab. He received an M.Sc. degree (with honors) and a Ph.D. degree in information engineering from Politecnico di Milano, Milan, Italy, in 2001 and 2005, respectively. From 2006 to 2017, he was a faculty member of the Electrical and Computer Engineering (ECE) Department at New Jersey Institute of Technology (NJIT), where he was affiliated with the Center for Wireless Information Processing (CWiP). His research interests include information theory, machine learning, wireless communications, neuromorphic computing, and quantum machine learning. Dr Simeone is a co-recipient of the 2022 IEEE Communications Society Outstanding Paper Award, the 2021 IEEE Vehicular Technology Society Jack Neubauer Memorial Award, the 2019 IEEE Communication Society Best Tutorial Paper Award, the 2018 IEEE Signal Processing Best Paper Award, the 2017 JCN Best Paper Award, the 2015 IEEE Communication Society Best Tutorial Paper Award and of the Best Paper Awards of IEEE SPAWC 2007 and IEEE WRECOM 2007. He was awarded an Open Fellowship by the EPSRC in 2022 and a Consolidator grant by the European Research Council (ERC) in 2016. His research has been also supported by the U.S. National Science Foundation, the European Commission, the European Research Council, the Vienna Science and Technology Fund, the European Space Agency, as well as by a number of industrial collaborations including with Intel Labs and InterDigital. He was the Chair of the Signal Processing for Communications and Networking Technical Committee of the IEEE Signal Processing Society in 2022, as well as of the UK & Ireland Chapter of the IEEE Information Theory Society from 2017 to 2022. He was a Distinguished Lecturer of the IEEE Communications Society in 2021 and 2022, and he was a Distinguished Lecturer of the IEEE Information Theory Society in 2017 and 2018. Prof. Simeone is the author of the textbook "Machine Learning for Engineers" published by Cambridge University Press, four monographs, two edited books, and more than 200 research journal and magazine papers. He is a Fellow of the IET, EPSRC, and IEEE.

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4