Verifiable end-to-end delegated variational quantum algorithms

Matteo Inajetovic [1,*] Petros Wallden [2] and Anna Pappa [1]
[1]*Electrical Engineering and Computer Science Department, Technische Universität, Berlin, Germany*
[2]*Quantum Software Lab, School of Informatics, University of Edinburgh, Edinburgh, United Kingdom*

(Received 2 July 2025; accepted 9 May 2026; published 10 July 2026)

Variational quantum algorithms (VQAs) have emerged as promising candidates for solving complex optimization and machine learning tasks on near-term quantum hardware. However, executing quantum operations remains challenging for small-scale users because of several hardware constraints, making it desirable to delegate parts of the computation to more powerful quantum devices. In this work, we introduce a framework for delegated variational quantum algorithms, where a client with limited quantum capabilities delegates the execution of a VQA to a more powerful quantum server. In particular, we introduce a protocol that enables a client to delegate a variational quantum algorithm to a server while ensuring that the input, the output, and also the computation itself remain secret. Additionally, if the protocol does not abort, the client can be certain that the computation outcome is indeed correct. This work builds on the general verification protocol introduced by Fitzsimons and Kashefi [Phys. Rev. A **96**, 012303 (2017)], tailoring it to VQAs. Our approach first proposes a verifiable protocol for delegating the quantum computation at each optimization step of a VQA and then combines the iterative steps into an error-resilient optimization process that offers end-to-end verifiable algorithm execution. We also simulate the performance of our protocol tackling the transverse field Ising model. Our results demonstrate that secure delegation of variational quantum algorithms is a realistic solution for near-term quantum networks, paving the way for practical quantum cloud computing applications.

DOI: 10.1103/9yqy-8ch9

I. INTRODUCTION

Quantum computing is poised to revolutionize information processing by leveraging the principles of quantum mechanics to solve problems that are intractable for classical computers. With advancements in quantum hardware, software, and algorithms, we are witnessing rapid progress toward practical quantum computation. However, the realization of fault-tolerant, large-scale quantum computers remains a long-term goal. In the near term, noisy intermediate-scale quantum (NISQ) devices—quantum processors with tens to hundreds of qubits, but without robust error correction—represent the frontier of quantum technology. These NISQ devices offer promising opportunities for obtaining a quantum advantage in specific computational tasks, such as quantum simulation, optimization, and machine learning. Yet, their inherent noise and limited coherence times present significant challenges.

Variational quantum algorithms (VQAs) have emerged as a promising approach to leverage NISQ devices to solve complex computational problems [1]. They are hybrid quantum-classical algorithms, in which a quantum processor executes a circuit with adjustable parameters, and a classical optimizer iteratively updates these parameters to minimize a given cost function. By combining quantum circuits with classical optimization techniques, VQAs can efficiently explore high-dimensional solution spaces while mitigating the limitations of NISQ hardware.

As quantum processors become more accessible via cloud-based platforms, it is natural to consider the concept of delegated variational quantum algorithms (DVQAs). Such a process is envisioned to enable a classical client to delegate complex quantum computational tasks to a remote quantum server while maintaining control over the optimization process. Ensuring the privacy and correctness of delegated computations is crucial, particularly in scenarios where the client lacks direct access to the quantum hardware.

Most delegated quantum computation protocols are instantiated in the measurement-based quantum computing (MBQC) model, which naturally allows for blindness and verifiability of the computation [2–5]. Compared to the circuit model, MBQC inherently supports these features due to the one-way structure and the decoupling of resource state preparation from the actual (measurement-based) computation. Moreover, MBQC is particularly well suited to photonic implementations and quantum network architectures, further reinforcing its advantages for delegated quantum computation.

In the context of variational quantum algorithms, a direct application of the aforementioned established delegation methods is not feasible. Due to the hybrid nature of VQAs, which involves classical optimization and nondeterministic quantum computations, a tailored framework for delegating

*Contact author: m.inajetovic@tu-berlin.de

PROTOCOL 1. Verifiable step of the MB-DVQA.

1: **Inputs**: G and associated c coloring, $\theta = \{\theta_i\}_{i=1}^{N_P}$, N_s, ε_{th}.
2: **Protocol:**
(1) Suppose $n = d + t$ is the number of rounds, where $d = 2N_PN_s$ are the computation and t are the test rounds and the client chooses at random which round belongs in each set. For each round:
(a) If it is a test round, the client chooses a random graph coloring and sends the trap and dummy qubits to the server.
(b) If it is a computation round, the client sends the qubits associated with G.
(c) The server performs the blind measurement-based computation and returns the output.
(2) The client gathers and decodes the server's output:
(a) In case she detects a trap failure, she **aborts**.
(b) Otherwise, she **accepts**, obtaining $g(\theta)$ with a relative error up to ε_{th}.

such algorithms is required. This article explores the potential of measurement-based delegated variational quantum algorithms (MB-DVQAs). While previous studies have already provided the first investigations of MB-DVQAs [6–9], they focused on the shot-level verifiability of single iterations of the quantum variational procedure and therefore did not achieve verifiability for the entire optimization process. More specifically, given the number of shots required by VQAs, verifying every shot is not efficient.

We give an alternative protocol to achieve individual optimization step verifiability, and then propose a verifiable end-to-end MB-DVQA, encompassing both quantum and classical computations involved.

A. Overview of the contributions

(1) We show how to verify one step of an MB-DVQA (Protocol 1) by using a trapification scheme that alternates computation and test rounds [10], and prove that it succeeds with probability that grows exponentially with the number of total rounds (Theorem III 1).

(2) We propose a MB-DVQA using a custom gradient-descent optimizer (Protocol 2) that reruns highly attacked iterations without aborting the full optimization and show that it is verifiable (Theorem III 2).

(3) We benchmark the custom optimizer outlined in Protocol 2 through numerical simulations of a variational quantum eigensolver (VQE) [11] that finds the ground-state energy of the transverse field Ising Hamiltonian, incorporating simulated gradient perturbations that mimic adversarial attacks.

PROTOCOL 2. Verifiable MB-DVQA with gradient descent.

Inputs: Initial angles θ^1, maximum number of steps N_{iter}, a c coloring, N_s, ε_{th}, and learning rate α.
Protocol:
(1) For $k \in [1, \ldots, N_{\text{iter}}]$, run Protocol 1 to compute $\hat{g}(\theta^k)$:
(a) If it accepts: $\theta^{k+1} = \theta^k - \alpha\hat{g}(\theta^k)$.
(b) If it aborts: $\theta^{k+1} = \theta^k$ (i.e. rerun the same step).
(2) Set the final angles: $\theta^* \leftarrow \theta^{N_{\text{iter}}}$.
(3) If the optimization has not converged, **aborts**; otherwise, **accepts,** returning $\hat{f}^*$.

II. BACKGROUND

A central component in VQAs is a parametrized quantum circuit (PQC). This circuit (also called ansatz) is associated with a unitary $U(\theta)$, where θ is a vector of parametrized angles of dimension N_P. The main goal of a VQA is to minimize a cost function f represented as

$$f(\theta) = \text{Tr}[OU(\theta)\rho_{\text{in}}U^\dagger(\theta)],$$

where ρ_{in} is the initial state and O is the observable describing the problem to solve. In particular, O is decomposed as sum of N_o local Pauli operators P_i:

$$O = \sum_{i=1}^{N_o} O_i = \sum_{i=1}^{N_o} c_iP_i.$$

We will focus on gradient-based optimization, making use of the parameter-shift rule [12,13] to compute an approximation g of the gradient ∇f. For the resulting vector $g = (g_1(\theta), \ldots, g_{N_P}(\theta))$, it holds that for every element of the gradient vector the partial derivative is $\frac{\partial f}{\partial \theta} = \frac{1}{2}[f_+ - f_-]$, where

$$f_+ = f\left(\theta + \frac{\pi}{2}\right) \quad \text{and} \quad f_- = f\left(\theta - \frac{\pi}{2}\right).$$

This means that for a given PQC with N_P parameters, the total number of function evaluations per step is $2N_P$. Using a fixed amount of measurement shots per function evaluation N_s, we require a total number of $2N_PN_s$ shots to perform one step

Till recently, all proposals of VQEs were based on the circuit model of computation. In the last few years, variational quantum algorithms based on measurement-based quantum computation have also been proposed [14–20]. MBQC [21] differs significantly from the conventional gate-based approach; instead of executing quantum gates sequentially, MBQC begins with the preparation of a highly entangled state, and the computation is then carried out through a sequence of adaptive quantum measurements that depend on the outcomes of the previous measurements.

It has been shown that a client can securely and blindly delegate a quantum computation to a server by appropriately encrypting their input and the measurement angles of the MBQC pattern [3,22]. However, another crucial property in delegated quantum computing is verifiability: The client must be able to check whether the delegated computation has been executed correctly without malicious actions affecting the desired outcome. More specifically, the delegated protocol either outputs the desired computation or aborts. To ensure this property, it is sufficient to find an upper bound on the probability of failure, that is, the probability of having an incorrect quantum computation without detecting it by returning an abort signal. Initially proposed in Ref. [4], trapification allows one to detect dishonest behavior by modifying the graph G associated with the delegated computation, by inserting randomly prepared trap qubits in $|+_\theta\rangle = (|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$ and separating those from the rest of the computation by means of so-called dummy qubits, randomly chosen from $\{|0\rangle, |1\rangle\}$.

In a previous study of delegated bounded-error quantum polynomial-time computation [10], the proposed verification scheme interleaves d *computation rounds* and t *test rounds* (for a total of $n = d + t$ rounds), rather than inserting traps into the graph G associated with the desired computation. The latter approach results in an enlarged graph, whereas interleaving rounds maintain the original graph size. In the computation rounds, the client prepares and sends the qubits and the measurements associated with the desired computation unitary U (defined by graph G), while the test rounds instead are just made up of traps and dummy qubits. In particular, Ref. [10] makes use of a c coloring of G to place traps and dummy qubits. By having the same graph structure for both computation and test rounds, the server cannot distinguish them due to blindness inherited by previous works [3].

III. METHODS

Our goal is to provide an MB-DVQA that converges and is at the same time blind, secure, and verifiable. Since blindness and security are directly inherited from previous works on delegated MBQC [23], the only thing that remains to be addressed is verifiability. We first propose a verifiable protocol (Protocol 1) computing the gradients of one MB-DVQA step using the parameter-shift rule. In particular, we provide a bound (Lemma III 1) on the gradient's relative error in case of adversarial behavior, namely, when cost function evaluations are affected by a certain number of corrupted computation rounds. Subsequently, we make use of this result to come up with a trapification scheme based on Ref. [10], yielding an undetected highly corrupted outcome with a very low probability (Theorem III 1). We then exploit this protocol within the context of a complete MB-DVQA using gradient descent to minimize the cost function. Specifically, we modify the classical optimization part of the VQA by reperforming highly attacked steps detected by the step-level protocol and prove convergence and verifiability (Theorem III 2) for the full MB-DVQA protocol (Protocol 2).

A. Verifiability for one step of the MB-DVQA

Existing blind and verifiable MBQC protocols can in principle be directly applied to VQAs at shot-level computation. Nevertheless, this approach is impractical given the large number of shots required by the total number of function evaluations in the variational procedure. Furthermore, running a verification protocol for each shot and abort every corrupted one implies a heavy consumption of run-time resources. The robust VBQC protocol proposed by Ref. [10] allows one to delegate multiple shots at once and appears as a better candidate to delegate variational algorithms. However, this protocol, designed for deterministic BQP computations, does not account for the probabilistic, expectation-value-based cost functions required for variational parameter updates. Consequently, adapting this protocol to VQAs requires a tailored verification framework that aggregates measurements across shots and handles the multiple evaluations required per optimization step, while tolerating a small, noncritical fraction of corrupted shots. Therefore, the aim of Protocol 1 is to delegate the quantum computation required to compute the gradient of the cost function with a relative error bounded by a target value ε_{th}. This quantum computation consists of all the $2N_P$ evaluations of the cost function f, each performed N_s times.

To provide verifiability for this protocol and allow the client to check for the correctness of the computation, we adapt the trapification protocol proposed in Ref. [10]. In that work, majority vote was used to bound the probability of failure of the verification process for BQP computations. Here instead, due to the probabilistic nature of PQCs, we consider the computation outcome through the expectation value of observables, which are typically used to evaluate the cost function. We therefore execute $d = 2N_P N_s$ computation rounds and t test rounds shared across different quantum computations (i.e., distinct f evaluations). This allows us to reduce the amount of required test rounds compared to the case where each quantum computation is delegated separately.

In order to prove verifiability for our protocol, we first establish a connection between the relative error in the gradient approximation and the adversarial attacks, represented in our framework by means of the number of corrupted computation rounds δ.

Lemma III 1. We assume a DVQA step performing a parameter-shift rule to compute a gradient approximation g of ∇f with a global observable $O = \sum_i c_i P_i$. Using N_s shots per circuit evaluation, the gradient relative error $\varepsilon = \frac{\|g-\hat{g}\|_2}{\|g\|_2}$ and the number of corrupted computation rounds δ are related as follows:

$$\varepsilon \leqslant \frac{\sum_i |c_i|}{\epsilon_0 N_s}\delta, \tag{1}$$

where $\epsilon_0 > 0$ is a lower bound on the gradient norm, i.e., $\|g\|_2 \geqslant \epsilon_0$.

Proof. When we perform one execution of the PQC, the state collapses to an eigenstate of the measured observable. Therefore, the finite-shot estimator of the cost function, obtained by distributing N_s shots across the N_o Pauli terms, can be expressed as

$$f = \frac{N_o}{N_s}\sum_{i=1}^{N_o} c_i \sum_{k=1}^{N_s/N_o} \lambda_k^i.$$

Let us consider a single eigenvalue λ_k^i of the local Pauli P_i measured at the kth round. Due to attacks, in the worst-case scenario, we instead measure a corrupted eigenvalue equal to the most distant with respect to the correct λ_k^i. One can bound the difference between the correct and the worst-case corrupted eigenvalues (respectively, λ_k^i and $\hat{\lambda}_k^i$) as $\lambda_k^i - \hat{\lambda}_k^i \leqslant 2$, since Pauli operators have eigenvalues $+1$ or -1.

Due to the blindness property of our delegated protocol, we can assume that, on average, the δ' corrupted executions are uniformly distributed among the N_o Pauli observables, yielding an expected δ'/N_o corrupted measurements per local expectation value. Therefore, the absolute difference between the ideal function f and its corrupted version can be bounded as follows:

$$|f - \hat{f}| \leqslant \frac{N_o}{N_s}\sum_{j=1}^{N_o}\left|c_j \sum_{k=1}^{N_s/N_o}\left(\lambda_k^j - \hat{\lambda}_k^j\right)\right| \leqslant \frac{2\delta'}{N_s}\sum_{j=1}^{N_o}|c_j|.$$

For each component of the gradient vector obtained with the parameter-shift rule, we get

$$|g_i - \hat{g}_i| = \frac{1}{2}|(f_{i+} - \hat{f}_{i+}) - (f_{i-} - \hat{f}_{i-})| \leqslant \frac{(\delta_{i+} + \delta_{i-})}{N_s} \sum_{j=1}^{N_o} |c_j|.$$

The total number of corrupted computation rounds per step, distributed across the function evaluations required for all the components of the gradient vector, is expressed as $\delta = \sum_{i=1}^{N_P} \delta_i$, where $\delta_i = \delta_{i+} + \delta_{i-}$.

Given the relative error $\varepsilon = \frac{\|g-\hat{g}\|_2}{\|g\|_2}$, the numerator can be bounded using the component-wise absolute error bound:

$$|g_i - \hat{g}_i| \leqslant \frac{\sum_j |c_j|}{N_s}\delta_i,$$

$$\|g - \hat{g}\|_2 = \sqrt{\sum_{i=1}^{N_P} |g_i - \hat{g}_i|^2} \leqslant \frac{\sum_j |c_j|}{N_s} \sqrt{\sum_{i=1}^{N_P} \delta_i^2} \leqslant \frac{\sum_j |c_j|}{N_s}\delta,$$

where in the last step we used $\sqrt{\sum_i \delta_i^2} \leqslant \sum_i \delta_i$ when all $\delta_i \geqslant 0$. Assuming $\|g\|_2 \geqslant \epsilon_0 > 0$, we obtain

$$\varepsilon \leqslant \frac{\sum_j |c_j|}{\epsilon_0 N_s}\delta. \tag{2}$$

■

Given a desired target relative error ε_{th}, Lemma III 1 provides a bound on the tolerated corrupted computation rounds $\delta_{\max}$ affecting one step:

$$\delta \leqslant \delta_{\max} = \frac{\varepsilon_{\text{th}}\epsilon_0 N_s}{\sum_j |c_j|} = \frac{\varepsilon_{\text{th}}\epsilon_0 d}{2N_P \sum_j |c_j|} = wd, \tag{3}$$

where $w = \frac{\varepsilon_{\text{th}}\epsilon_0}{2N_P \sum_i |c_i|}$.

Now that we have a bound on δ, we can proceed with showing verifiability of a MB-DVQA step.

Theorem III 1. Let g denote the parameter-shift rule approximation of ∇f, and assume that $\|g\|_2 \geqslant \epsilon_0 > 0$. An MB-DVQA step following Protocol 1 computes g with relative error up to ε_{th} and probability of failure decreasing exponentially with the number of rounds n.

Proof. Given a total of m attacked rounds, let $X_d \approx \text{Binomial}(m, \frac{d}{n})$ be the random variable describing the probability of computation rounds being attacked. Also, let $X_t \approx \text{Binomial}((\frac{m}{n} - \epsilon_1)t, \frac{1}{c})$ be the random variable related to the number of attacks detected by the traps in the trapification scheme (with some $\epsilon_1 > 0$). Therefore, the probability of failure will be the probability of having more than $\delta_{\max}$ corrupted computation rounds and no detected attacks by any trap. This probability can be bounded as

$$\begin{aligned}\text{P(fail)} &\leqslant \max_{m\in[0,n]} \Pr\left[X_d \geqslant \delta_{\max} \wedge X_t < 1\right] \\ &\leqslant \max\left(\max_{m<nw} \Pr[X_d \geqslant \delta_{\max}], \max_{m\geqslant nw} \Pr[X_t < 1]\right).\end{aligned}$$

When $m \geqslant nw$, considering that computation and test rounds are chosen at random, the number of affected computation rounds $m\frac{d}{n}$ is higher than $\delta_{\max}$, leading to $\Pr[X_d > \delta_{\max}] = 1$. Therefore, we focus on the test rounds term to upper bound the probability of failure. Instead, when $m < nw$, we upper bound the term related to the computation rounds.

(1) Let us define $X' \approx \text{Binomial}(m, \frac{t}{n})$ to be the probability distribution describing the attacked test rounds. By the law of total probability, the probability to have no detected attacks in the test rounds is

$$\begin{aligned}\Pr(X_t < 1) &= \Pr[X_t = 0|X' \leqslant k]\Pr[X' \leqslant k] \\ &\quad + \Pr[X_t = 0|X' > k]\Pr[X' > k] \\ &\leqslant \Pr[X' \leqslant k] + \Pr[X_t = 0|X' > k],\end{aligned}$$

where $k = (\frac{m}{n} - \epsilon_1)t$.

Using Hoeffding's inequality and the fact that $(1 - 1/c)^k \leqslant e^{-k/c}$, we can rewrite the above probability (with $\tau = t/n$):

$$\Pr(X_t < 1) \leqslant \exp\left(\frac{-2\epsilon_1^2\tau^2 n^2}{m}\right) + \exp\left(-\left(\frac{m}{n} - \epsilon_1\right)\frac{\tau n}{c}\right).$$

Since $w \leqslant 1$ (m always such that $m \leqslant n$), we have $n \leqslant n^2/m \leqslant n/w$, and therefore $\exp(-2\epsilon_1^2\tau^2\frac{n^2}{m}) \leqslant \exp(-2\epsilon_1^2\tau^2 n)$, yielding

$$\Pr(X_t < 1) \leqslant \exp\left(-2\epsilon_1^2\tau^2 n\right) + \exp\left(-\frac{\tau n}{c}(w - \epsilon_1)\right).$$

Both terms, with $w > \epsilon_1$, decay exponentially with the number of copies n.

(2) When $m < nw$, using Hoeffding's inequality:

$$\Pr[X_d > \delta_{\max}] \leqslant \exp\left(-\frac{2d^2}{m}\left(\frac{m}{n} - w\right)^2\right).$$

Therefore, we have a bound on the probability of failure decaying exponentially with respect to the number of copies n and the number of test rounds t, since d is fixed. ■

So far, the verifiability of one step of the MB-DVQA has been shown. However, it might not be clear how to choose the threshold on the tolerated gradient relative error ε_{th}. In the next section, we will show how to appropriately set this bound to ensure both convergence of the optimization and verification of the full MB-DVQA.

B. Verifiable MB-DVQA with gradient descent

In this section, we explore what happens when we build a verifiable MB-DVQA protocol (Protocol 2) using Protocol 1. Given limited resources, namely, a total number of available rounds, the MB-DVQA protocol runs Protocol 1 for each step of the optimization. In case a step aborts, the protocol rejects the last gradient computation and reperforms the same step; otherwise, it accepts the outcome (with a relative error at most ε_{th}) and updates the variational parameters.

In the scenario, where a delegated VQA is executed with verifiability provided only at the step level, there is no guarantee regarding the final outcome of the variational optimization, namely, the final value of f. Although highly corrupted steps can be rerun, once the MB-DVQA is terminated, we cannot clearly state if the outcome matches the optimum of the cost function f^*. Therefore, such a protocol would have no practical means of knowing whether an execution is successful.

To address this gap, we introduce Protocol 2 that achieves verifiability by means of geometric convergence of the optimization procedure, as shown in Theorem III 2. This is done by properly tuning ε_{th} depending on the problem, a task made possible by the result achieved in the previous sections with Theorem III 1 and Lemma III 1. Given convergence guarantees, once the protocol terminates, the client either obtains the desired outcome f^* or receives an abort response, achieving verifiability.

Theorem III 2. Given a μ strongly convex cost function f with L-Lipschitz continuous gradient and a global optimum f^*, let g denote the parameter-shift rule approximation of ∇f and assume that $\|g\|_2 \geqslant \epsilon_0 > 0$. Protocol 2 ensures verifiability, provided that the learning rate α and error bound on the gradient ε_{th} satisfy the following conditions:

(1) $\alpha L < \frac{2(1-\varepsilon_{\text{th}})^2-\varepsilon_{\text{th}}^2}{(1-\varepsilon_{\text{th}})(1+\varepsilon_{\text{th}})^2}$,

(2) $\mu\alpha[2(1-\varepsilon_{\text{th}}) - \alpha L(1+\varepsilon_{\text{th}})^2 - \frac{\varepsilon_{\text{th}}^2}{1-\varepsilon_{\text{th}}}] < 1$.

Sketch of Proof. As explained in detail in Appendix 2, a VQA using stochastic gradient descent to optimize a function f can converge in the presence of adversarial attacks provided that the gradient relative error remains sufficiently bounded. If no such bound exists, multiple steps with large gradient errors may occur, potentially jeopardizing the optimization and leading to divergence. To prevent this and ensure convergence, Protocol 2 integrates Protocol 1, which detects steps where the relative error exceeds the threshold ε_{th}. Such steps are then reexecuted until the conditions of the theorem hold. These inequalities, derived in the full proof (Appendix 2), are related to standard conditions required for gradient-descent optimization, namely, having $0 < \gamma < 1$ such that for each optimization step k:

$$\mathbb{E}[f^{k+1}] - f^* \leqslant \gamma(\mathbb{E}[f^k] - f^*).$$

Notably, when $\varepsilon_{\text{th}} \to 0$, the inequalities above reduce to the standard gradient-descent conditions for convergence, having $\gamma = 1 - \mu\alpha\,(2 - \alpha L)$. This guarantees the average decrease of the function required for convergence and also bounds the final error neighborhood in the presence of shot noise, albeit at the cost of an overhead of steps due to repetitions. Finally, we associate convergence guarantees with verifiability: Since the client has access to classical gradient information at each step k, the protocol either outputs the correct result, namely, the optimal value of the cost function achieved due to convergence, or aborts. ■

While VQA landscapes are generally nonconvex, convexity-based analyses—common also in classical machine learning—can still provide useful insight, for example, when initialization or ansatz design places the optimizer in more regular regions of the landscape. In the above theorem, we adopt the strong convexity assumption as a simplified analytical setting to enable convergence analysis [24]. This assumption also guarantees that the lower bound on the gradient norm ϵ_0 in Lemma III 1 is sufficiently large and therefore provides a meaningful bound for the gradient relative error ε. Our assumptions do not cover scenarios where the cost landscape is affected by barren plateaus or local minima; addressing fully realistic, problem-specific landscapes and alternative optimization methods is left for future work.

IV. SIMULATIONS

We start by benchmarking the performance of Protocol 2, to demonstrate its effectiveness. We do this by trying to find the ground state of the transverse field Ising model (TFIM), a fundamental quantum spin model describing a system of interacting spins (typically on a lattice):

$$H = h\sum_i X_i + \sum_{\langle i,j\rangle} Z_i Z_j,$$

where $\langle i, j\rangle$ determines the edges of the lattice and h is the strength of the external transverse field. We express attacks as perturbations on the gradient vector for each step k (in the context of Protocol 2):

$$\hat{g}_i(\theta^k) = g_i(\theta^k) + \nu_i^k. \tag{4}$$

Each ν_i^k is drawn independently from a uniform distribution $\mathcal{U}(-\Delta, \Delta)$, where Δ is a parameter tuning the impact of the perturbation. The perturbation vector is defined as

$$\nu^k = \begin{cases} (\nu_1^k, \nu_2^k, \ldots, \nu_{N_P}^k), & \text{with probability } p, \\ (0, 0, \ldots, 0), & \text{with probability } 1-p. \end{cases}$$

The simulation of the computation rounds in the absence of quantum hardware noise is carried out using the Qrisp programming framework [25]. For now, we assume the existence of traps allowing for a negligible probability of failure and leave the simulation of the test rounds for later. In this way, we can simulate the trap detection by evaluating the relative error. Protocol 2 reperforms heavily attacked steps without aborting the full VQE and accepts steps that fall below the error threshold.

We show examples for two-dimensional TFIMs using $p = 0.7$ and an ansatz with N_L layers and n qubits; each layer U_l is given by

$$U_l(\theta_l) = \left(\prod_{i=1}^{n-1} \text{CNOT}_{i,i+1}\right)\left(\bigotimes_{i=1}^{n} R_y(\theta_l^i)\right).$$

From Fig. 1, we can see how a standard VQE using gradient descent diverges due to attacks, while the proposed approach, as expected, turns out to be robust and converges to the optimal solution with an overhead of steps with respect to the ideal attack-free case. Using shot-level verification would significantly increase the number of aborted shots over the full VQE run, extending the run-time. In contrast, our approach tolerates a reasonable fraction of corrupted shots, while enabling convergence within a reasonable time frame.

It is important to emphasize that our protocol does not depend on the aforementioned attack model. The latter has been designed to illustrate the performance of the proposed work. As reported in the previous section, Protocol 2 is general and robust, providing tools to adjust the number of traps to ensure convergence, independent of the attack model.

To complement the numerical analysis, we provide simulations of Protocol 1 tackling a two-qubit TFIM. In particular, we make use of Veriphix [26], a library for verifiable blind quantum computing based on the MBQC programming framework Graphix [27]. In Fig. 2, we show how the number of test rounds t can be tuned to detect highly corrupted steps depending on ε_{th}, the threshold value required by Protocol 2

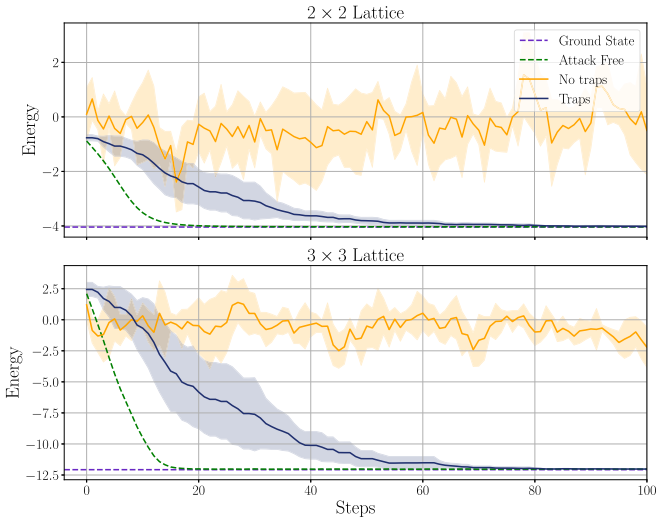

FIG. 1. Average performance over five runs comparing the proposed protocol and a standard VQE in the presence of the described attacks with $p = 0.7$. We tackle the following two-dimensional TFIM lattices: 2×2 and 3×3 with $h = 0.2$. The dashed green line (Attack Free) refers to an ideal attack-free scenario, the yellow line (No traps) is related to a nonverifiable delegated VQE in presence of attacks, and finally, the proposed protocol is shown in blue (Traps). We use $N_s = 1000$, $N_L = 2$ layers, and a learning rate $\alpha = 0.2$ for the optimization.

to ensure convergence. The Python code used to reproduce the simulations is openly available in Ref. [28].

V. CONCLUSION

This work offers valuable insights into the future of delegated quantum computation in the NISQ era and beyond, by proposing an end-to-end verifiable delegated variational quantum protocol. The theoretical results are further validated with simulations that tackle the transverse field Ising model. Although our work is currently based on gradient-descent optimization, it is readily adaptable to alternative optimization techniques. Specifically, Protocol 1 can be modified to accommodate different optimizers, leading to appropriately revised versions of Lemma III 1 and Theorem III 1. On the other hand, with regard to the verifiability of the overall VQA, an optimizer-specific theorem should be provided to guarantee convergence and, consequentially, verifiability. Furthermore, in our protocol, we assume a fixed relative error threshold for all the steps. In future work, the threshold across the steps could be varied depending on the current gradient value and accordingly the number of traps required for each step could be adjusted, thus reducing the total number of delegated computations. Another extension of this work could involve an improvement of the trapification procedure, for instance, exploiting the scheme based on stabilizer testing proposed in Ref. [29]. Finally, although circuit-based DVQA protocols have already been proposed (such as the one by Li *et al.* [30] based on quantum homomorphic encryption), they currently only provide blindness. It remains an open and interesting question how verification could be incorporated in such schemes, either through adaptations of our approach or via dedicated techniques like accreditation [31].

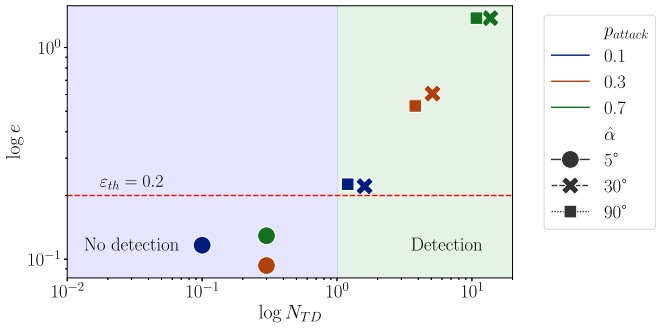

FIG. 2. Plot of relative error ε vs number of trap detections N_{TD} in one execution of Protocol 1 (in logarithmic scale). Here, we choose $d = 4000$ and $t = 50$. p_{attack} is the probability that a round is attacked and $\hat{\alpha}$ shows examples of a degree shift applied by an attacker on the measurement angles of the output qubits. The points represent average scores computed over ten runs for each setting. As expected, when the relative error exceeds the threshold ε_{th} (shown with the red dashed line), the attack is detected.

ACKNOWLEDGMENTS

M.I. and A.P. acknowledge support from the Federal Ministry for Economic Affairs and Climate Action (project Qompiler, Grant No. 01MQ22005A), the DFG (Emmy Noether programme, Grants No. 418294583 and No. SPP2514), the BMBF (project tubLANQ.0, Grant No. 16KISQ087K), and the Berlin Quantum Alliance. P.W. acknowledges support by EPSRC Grant No. EP/T001062/1.

DATA AVAILABILITY

The data that support the findings of this article are openly available [28]; embargo periods may apply.

APPENDIX

1. Useful lemmas

Lemma A 1. (Hoeffding's inequality). Given a binomial distribution $X \approx \text{Binomial}(n, p)$ an upper bound to the cumulative distribution function can be obtained for $k \leqslant np$:

$$\Pr(X \leqslant k) \leqslant \exp\left(-2n\left(p - \frac{k}{n}\right)^2\right).$$

Lemma A 2. (Lemma, page 870 in Ref. [32]). Given $f : \mathbb{R}^d \to \mathbb{R}$ with optimum f^* and μ strongly convex, it holds that

$$2\mu(f(x) - f^*) \leqslant \|\nabla f(x)\|^2, \forall x \in \mathbb{R}^d.$$

Lemma A 3. (Lemma 4.2 in Ref. [33]) The iterations of the stochastic gradient descent (with learning rate α) for an L-Lipschitz continuous gradient function f satisfy

$$\mathbb{E}[f(\theta^{k+1})] - f(\theta^k) \leqslant -\alpha \nabla f(\theta^k)^T \mathbb{E}[g(\theta^k)] + \frac{\alpha^2 L}{2} \mathbb{E}\left[\|g(\theta^k)\|_2^2\right],$$

where g is the estimation of ∇f.

2. Stochastic gradient-descent convergence with shot noise and attacks

Here, we provide convergence guarantees for a VQA using gradient descent in presence of adversarial attacks. In an ideal gradient-descent optimization, one has access to the real gradient value ∇f; therefore, the update rule for each optimization step k is

$$\theta^{k+1} = \theta^k - \alpha \nabla f(\theta^k).$$

Nevertheless, when running gradient descent as optimizer in a practical setting, the stochasticity of the estimation of ∇f needs to be taken into account. In particular, by executing a VQA with an NISQ device the stochasticity source is the so-called shot-noise, namely, the estimation error due to limited amount of measurements available for the computation of the expectation values. In this scenario, we rewrite the update rule as follows:

$$\theta^{k+1} = \theta^k - \alpha\, g(\theta^k),$$

where $g(\theta^k)$ is a stochastic (shot-noise) estimate of the true gradient $\nabla f(\theta^k)$. Here, we assume that $\mu_g = \mathbb{E}[g] = \nabla f$ and that the stochastic gradient estimator $g(\theta^k)$ has covariance $\mathrm{Cov}[g(\theta^k)] = \frac{\Sigma}{N_s}$; hence, the total variance of the gradient estimate is

$$\sigma_g^2 = \frac{\mathrm{Tr}[\Sigma]}{N_s} = \frac{1}{N_s}\sum_{i=1}^{N_P} \Sigma_{ii},$$

where $\Sigma_{ii} = \mathrm{Var}[g_i]$ denotes the variance of the ith component. By introducing in our analysis adversarial attacks, we introduce the "corrupted" gradient $\hat{g}$, having

$$\theta^{k+1} = \theta^k - \alpha \hat{g}(\theta^k).$$

Thanks to our protocol and Lemma III 1, we have

$$\|\hat{g}(\theta^k) - g(\theta^k)\|_2 \leqslant \varepsilon, \|g(\theta^k)\|_2.$$

Then, we define $\nu^k := \hat{g}(\theta^k) - g(\theta^k)$, so that $\|\nu^k\|_2 \leqslant \varepsilon \|g(\theta^k)\|_2$.

In order to investigate the convergence and apply Lemma A 3, we need to compute $\nabla f^T \mathbb{E}[\hat{g}]$ and $\mathbb{E}[\|\hat{g}\|_2^2]$.

By linearity of expectation and Cauchy-Schwarz,

$$\begin{aligned}
\mathbb{E}[\nabla f^\top \hat{g}] &= \nabla f^\top \mathbb{E}[g] + \nabla f^\top \mathbb{E}[\nu] \\
&= \|\nabla f\|_2^2 + \nabla f^\top \mathbb{E}[\nu] \\
&\geqslant \|\nabla f\|_2^2 - \|\nabla f\|_2 \|\mathbb{E}[\nu]\|_2 \\
&\geqslant \|\nabla f\|_2^2 - \|\nabla f\|_2\, \mathbb{E}\|\nu\|_2 \\
&\geqslant \|\nabla f\|_2^2 - \varepsilon \|\nabla f\|_2\, \mathbb{E}\|g\|_2 \\
&\geqslant \|\nabla f\|_2^2 - \varepsilon \|\nabla f\|_2 \sqrt{\mathbb{E}\|g\|_2^2} \\
&= \|\nabla f\|_2^2 - \varepsilon \|\nabla f\|_2 \sqrt{\|\nabla f\|_2^2 + \sigma_g^2} \\
&\geqslant (1-\varepsilon)\|\nabla f\|_2^2 - \varepsilon \|\nabla f\|_2 \sigma_g, \qquad \text{(A1)}
\end{aligned}$$

where we used (1) Jensen inequality $\|\mathbb{E}[\nu]\| \leqslant \mathbb{E}\|\nu\|$, (2) the error bound $\|\nu\|_2 \leqslant \varepsilon \|g\|_2$, (3) Cauchy-Schwarz $\mathbb{E}\|g\| \leqslant \sqrt{\mathbb{E}\|g\|^2}$, (4) $\mathbb{E}[\|g\|_2^2] = \|\nabla f\|_2^2 + \sigma_g^2$, and (5) $\sqrt{a^2+b^2} \leqslant a+b$ for $a, b \geqslant 0$.

$$\begin{aligned}
\mathbb{E}\big[\|\hat{g}\|_2^2\big] &= \mathbb{E}\big[\|g+\nu\|_2^2\big] \\
&= \mathbb{E}\big[\|g\|_2^2\big] + 2\mathbb{E}\big[g^\top \nu\big] + \mathbb{E}\big[\|\nu\|_2^2\big] \\
&\leqslant \mathbb{E}\big[\|g\|_2^2\big] + 2\varepsilon \mathbb{E}\big[\|g\|_2^2\big] + \varepsilon^2 \mathbb{E}\big[\|g\|_2^2\big] \\
&= (1+\varepsilon)^2 \mathbb{E}\big[\|g\|_2^2\big] \\
&= (1+\varepsilon)^2 \big(\|\nabla f\|_2^2 + \sigma_g^2\big). \qquad \text{(A2)}
\end{aligned}$$

Now, starting from Lemma A 3,

$$\begin{aligned}
\mathbb{E}[f(\theta^{k+1})] - f(\theta^k) &\leqslant -\alpha \mathbb{E}[\nabla f^\top \hat{g}] + \frac{\alpha^2 L}{2} \mathbb{E}\big[\|\hat{g}\|_2^2\big] \\
&\leqslant -\alpha(1-\varepsilon)\|\nabla f\|_2^2 + \alpha\varepsilon \|\nabla f\|_2 \sigma_g \\
&\quad + \frac{\alpha^2 L}{2}(1+\varepsilon)^2 \big(\|\nabla f\|_2^2 + \sigma_g^2\big) \\
&\leqslant -\frac{1}{2}\alpha[2(1-\varepsilon) - \alpha L (1+\varepsilon)^2]\|\nabla f\|_2^2 \\
&\quad + \alpha\varepsilon \|\nabla f\|_2 \sigma_g + \frac{\alpha^2 L}{2}(1+\varepsilon)^2 \sigma_g^2. \qquad \text{(A3)}
\end{aligned}$$

To handle the cross term $\alpha\varepsilon\|\nabla f\|_2\sigma_g$ and avoid dependence on the iterates, we apply the following inequality $ab \leqslant \frac{a^2}{2\lambda} + \frac{\lambda b^2}{2}$ [34] (valid for any $\lambda > 0$ and $a, b \geqslant 0$) obtaining

$$\alpha\varepsilon\|\nabla f\|_2\sigma_g \leqslant \frac{\alpha\varepsilon}{2\lambda}\|\nabla f\|_2^2 + \frac{\alpha\varepsilon\lambda}{2}\sigma_g^2. \qquad \text{(A4)}$$

Choosing $\lambda = \frac{1-\varepsilon}{\varepsilon}$ and substituting into the previous inequality gives

$$\begin{aligned}
&\mathbb{E}[f(\theta^{k+1})] - f(\theta^k) \\
&\quad \leqslant -\frac{\alpha}{2}\left[2(1-\varepsilon) - \alpha L(1+\varepsilon)^2 - \frac{\varepsilon^2}{1-\varepsilon}\right]\|\nabla f\|_2^2 \\
&\qquad + \left(\frac{\alpha(1-\varepsilon)}{2} + \frac{\alpha^2 L}{2}(1+\varepsilon)^2\right)\sigma_g^2. \qquad \text{(A5)}
\end{aligned}$$

The descent coefficient is positive whenever $\alpha L < \frac{2(1-\varepsilon)^2-\varepsilon^2}{(1-\varepsilon)(1+\varepsilon)^2}$, which is the condition stated in the main text. Applying Lemma A2, adding $f(\theta^k) - f^*$ to both terms, and applying the total expectation yields

$$\begin{aligned}
&\mathbb{E}[f(\theta^{k+1})] - f^* \\
&\quad \leqslant \underbrace{\left(1 - \mu\alpha\left[2(1-\varepsilon) - \alpha L(1+\varepsilon)^2 - \frac{\varepsilon^2}{1-\varepsilon}\right]\right)}_{\gamma} \\
&\qquad \times (\mathbb{E}[f(\theta^k)] - f^*) + \underbrace{\left(\frac{\alpha(1-\varepsilon)}{2} + \frac{\alpha^2 L}{2}(1+\varepsilon)^2\right)\sigma_g^2}_{B}, \qquad \text{(A6)}
\end{aligned}$$

where B is a constant independent of k.

Guarantees for convergence are met having $0 < \gamma < 1$, namely, when

$$\alpha L < \frac{2(1-\varepsilon)^2 - \varepsilon^2}{(1-\varepsilon)(1+\varepsilon)^2},$$
$$\mu\alpha\left[2(1-\varepsilon) - \alpha L(1+\varepsilon)^2 - \frac{\varepsilon^2}{1-\varepsilon}\right] < 1. \tag{A7}$$

When $\varepsilon \to 0$, these reduce to the standard gradient-descent conditions $\alpha L < 2$, recovering the original result. Protocol 2 allows to keep ε bounded and tunable to satisfy such requirements.

Without shot noise, namely, when $\sigma_g^2 \to 0$, strict convergence is achieved. Nevertheless, we cannot have a strict convergence due to the shot-noise variance, but rather a convergence to a neighborhood of f^*.

To define such a neighborhood, let us rearrange the inequality (A6) as

$$z_{k+1} \leqslant \gamma z_k + B,$$

where $z_k = \mathbb{E}[f(\theta^k)] - f^*$. When $k \to \infty$, then $z_\infty(1-\gamma) \leqslant B$, so the final error is bounded as

$$z_\infty \leqslant \frac{B}{1-\gamma} = \frac{\left(\frac{\alpha(1-\varepsilon)}{2} + \frac{\alpha^2 L}{2}(1+\varepsilon)^2\right)\sigma_g^2}{\mu\alpha\left[2(1-\varepsilon) - \frac{\varepsilon^2}{1-\varepsilon} - \alpha L(1+\varepsilon)^2\right]}.$$

The condition achieved for $\gamma < 1$ allows to have a nondivergent error z_∞. We can also derive how many iterations we need to achieve a final error ϵ. Assuming negligible σ_g^2 and having $\gamma = (1-\rho)$, we can write

$$\mathbb{E}[f(\theta^{k+1})] - f^* \leqslant (1-\rho)^k[\mathbb{E}[f(\theta^1)] - f^*] \leqslant \exp(-k\rho)[\mathbb{E}[f(\theta^1)] - f^*],$$

since $(1-\rho)^k \leqslant \exp(-k\rho)$. To have a final error $\mathbb{E}[f(\theta^{k+1})] - f^* \leqslant \epsilon$, only $O(\ln(1/\epsilon))$ iterations are required:

$$k \geqslant \frac{1}{\rho}\ln\left(\frac{\mathbb{E}[f(\theta^1)] - f^*}{\epsilon}\right).$$

It is important to notice that the iterations' scaling does not take into account the reperformed steps, considering instead only the iterations having a proper bound on the error variable.

Summarizing, to achieve convergence, having L and μ fixed by the problem, is it enough to set properly the learning rate α and the term ε. The latter can be modified by tuning the number of traps of our step-level protocol and Lemma III 1.

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K.Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[2] S. Barz, J. F. Fitzsimons, E. Kashefi, and P. Walther, Experimental verification of quantum computation, Nat. Phys. **9**, 727 (2013).

[3] A. Broadbent, J. Fitzsimons, and E. Kashefi, Universal blind quantum computation, in *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science* (American Physical Society, 2009), pp. 517–526.

[4] J. F. Fitzsimons and E. Kashefi, Unconditionally verifiable blind quantum computation, Phys. Rev. A **96**, 012303 (2017).

[5] A. Gheorghiu, T. Kapourniotis, and E. Kashefi, Verification of quantum computation: An overview of existing approaches, Theory Comput. Syst. **63**, 715 (2019).

[6] W. Li, S. Lu, and D.-L. Deng, Quantum federated learning through blind quantum computing, Sci. China Phys. Mech. Astron. **64**, 100312 (2021).

[7] Y. Shingu, Y. Takeuchi, S. Endo, S. Kawabata, S. Watabe, T. Nikuni, H. Hakoshima, and Y. Matsuzaki, Variational secure cloud quantum computing, Phys. Rev. A **105**, 022603 (2022).

[8] Y. Wang, J. Quan, and Q. Li, A delegated quantum approximate optimization algorithm, in *Proccedings of the 2022 14th International Conference on Wireless Communications and Signal Processing (WCSP)* (2022), pp. 804–808.

[9] J. Yang, B. Wang, J. Quan, and Q. Li, Verifiable cloud-based variational quantum algorithms, Opt. Commun. **578**, 131474 (2025).

[10] D. Leichtle, L. Music, E. Kashefi, and H. Ollivier, Verifying BQP computations on noisy devices with minimal overhead, PRX Quantum **2**, 040302 (2021).

[11] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[12] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[13] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[14] R. R. Ferguson, L. Dellantonio, K. Jansen, A. A. Balushi, W. Dür, and C. A. Muschik, A measurement-based variational quantum eigensolver, Phys. Rev. Lett. **126**, 220501 (2021).

[15] A. Majumder, M. Krumm, T. Radkohl, L. J. Fiderer, H. P. Nautrup, S. Jerbi, and H. J. Briegel, Variational measurement-based quantum computation for generative modeling, Phys. Rev. A **110**, 062616 (2024).

[16] F. K. Marqversen and N. T. Zinner, Applications and resource reductions in measurement-based variational quantum eigensolvers, Quantum Sci. Technol. **8**, 045001 (2023).

[17] M. Proietti, F. Cerocchi, and M. Dispenza, A native measurement-based QAOA algorithm, applied to the Max K-Cut problem, Phys. Rev. A **106**, 022437 (2022).

[18] Z. Qin, X. Li, Y. Zhou, S. Zhang, R. Li, C. Du, and Z. Xiao, Applicability of measurement-based quantum computation towards physically-driven variational quantum eigensolver, New J. Phys. **26**, 073040 (2024).

[19] A. Schroeder, M. Heller, and M. Gachechiladze, Deterministic ansätze for the measurement-based variational quantum eigensolver, arXiv:2312.13241 [quant-ph].

[20] T. Stollenwerk and S. Hadfield, Measurement-based quantum approximate optimization, in *Proccedings of the 2024 IEEE*

International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (2024), pp. 1115–1127.

[21] R. Raussendorf and H. J. Briegel, A one-way quantum computer, Phys. Rev. Lett. **86**, 5188 (2001).

[22] J. Fitzsimons, Private quantum computation: An introduction to blind quantum computing and related protocols, npj Quantum Inf. **3**, 23 (2017).

[23] V. Dunjko, J. F. Fitzsimons, C. Portmann, and R. Renner, *Composable Security of Delegated Quantum Computation* (Springer, Berlin, 2014), pp. 406–425.

[24] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Faehrmann, B. Meynard-Piganeau, and J. Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).

[25] R. Seidel, S. Bock, R. Zander, M. Petrič, N. Steinmann, N. Tcholtchev, and M. Hauswirth, Qrisp: A framework for compilable high-level programming of gate-based quantum computers, arXiv:2406.14792.

[26] S. Abdul Sater, M. Garnier, T. Martinez, and H. Ollivier, Veriphix, a Python library for the verification of quantum computation, Github, https://github.com/qat-inria/veriphix, Zenodo, https://doi.org/10.5281/zenodo.17435726.

[27] S. Sunami and M. Fukushima, Graphix: Optimizing and simulating measurement-based quantum computation on local-Clifford decorated graph, arXiv:2212.11975.

[28] M. Inajetovic, Code repository (2026), https://github.com/inajetovic/DelegatedVQA.

[29] T. Kapourniotis, E. Kashefi, D. Leichtle, L. Music, and H. Ollivier, Unifying quantum verification and error-detection: Theory and tools for optimisations, Quantum Sci. Technol. **9**, 035036 (2024).

[30] Q. Li, J. Quan, J. Shi, S. Zhang, and X. Li, Secure Delegated variational quantum algorithms, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems **43**, 3129 (2024).

[31] S. Ferracin, S. T. Merkel, D. McKay, and A. Datta, Experimental accreditation of outputs of noisy quantum computers, Phys. Rev. A **104**, 042603 (2021).

[32] B. Polyak, Gradient methods for the minimisation of functionals, USSR Comput. Math. Math. Phys. **3**, 864 (1963).

[33] L. Bottou, F. E. Curtis, and J. Nocedal, Optimization methods for large-scale machine learning, SIAM Rev. **60**, 223 (2018).

[34] From $0 \leqslant (a-b)^2$, it follows that $ab \leqslant \frac{a^2+b^2}{2}$. For any $\lambda > 0$, replacing with $a' = a/\sqrt{\lambda}$, $b' = b\sqrt{\lambda}$ yields $ab \leqslant \frac{a^2}{2\lambda} + \frac{\lambda b^2}{2}$.