Quantum Science and Technology

Celebrating the 10th anniversary of Quantum Science and Technology
PURPOSE-LED PUBLISHING™

PAPER • **OPEN ACCESS**

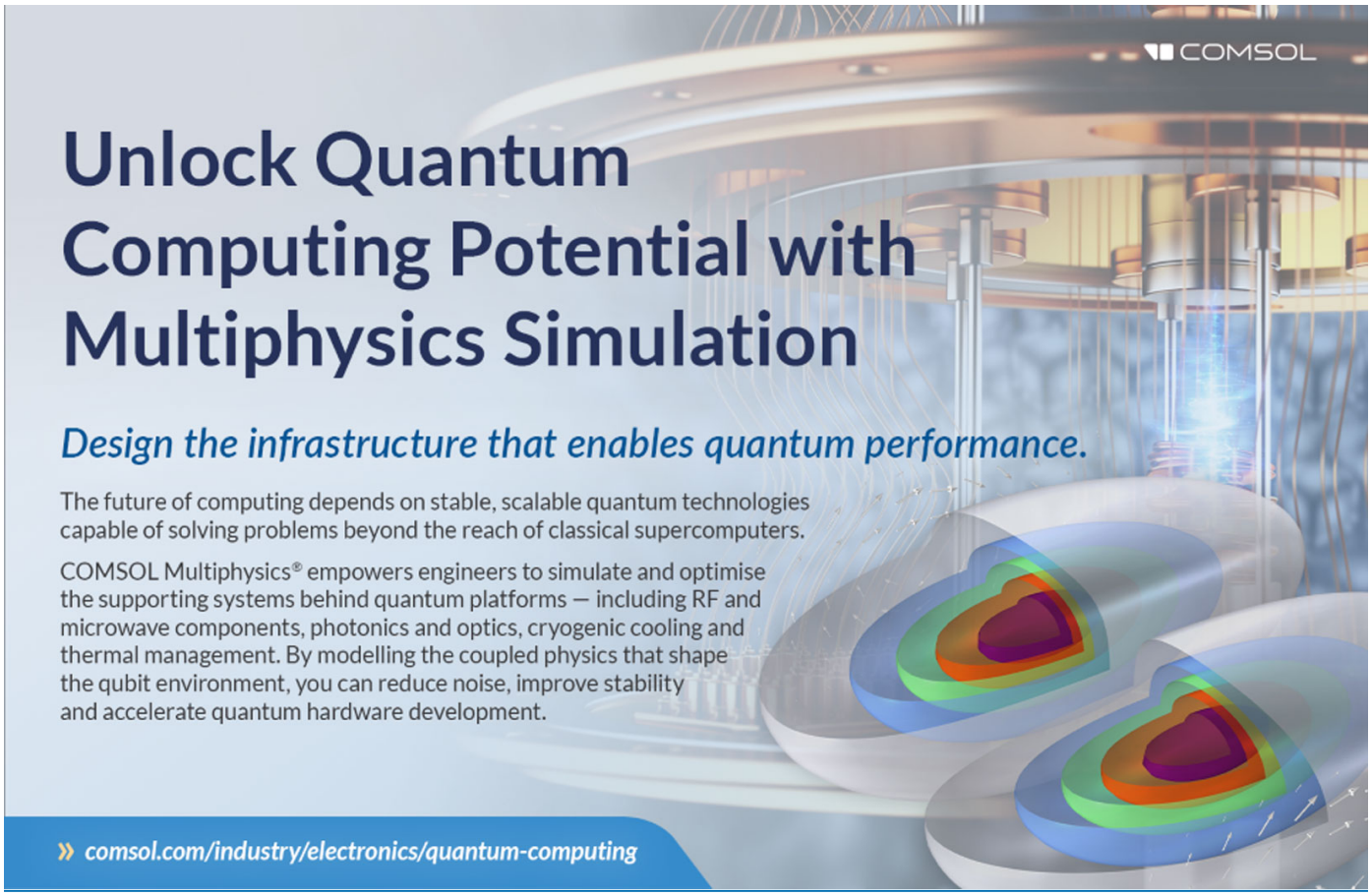
Operator learning for efficient quantum computation

To cite this article: Paul Over *et al* 2026 *Quantum Sci. Technol.* **11** 045097

View the article online for updates and enhancements.

You may also like

- Quantum mixed state compiling
Nic Ezzell, Elliott M Ball, Aliza U Siddiqui et al.
- Towards an efficient variational quantum algorithm for solving linear equations
WenShan Xu, Ri-Gui Zhou, YaoChong Li et al.
- PIQS: an efficient quantum subspace method for dynamical property estimation
Youle Wang, Wenbin Yu, Yanfeng Fan et al.



This content was downloaded from IP address 134.28.50.156 on 27/10/2026 at 08:31

Quantum Science and Technology

Check for updates

PAPER

Operator learning for efficient quantum computation

Paul Over[1,*], **Sergio Bengoechea**[1], **Leonardo Borello Busilacchi**[2], **Martin Kiffner**[2], **Thomas Rung**[1] **and Alexios A Michailidis**[2,*]

[1] Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Am Schwarzenberg-Campus 4, D-21073 Hamburg, Germany

[2] PlanQC GmbH, Münchener Str. 34, Garching, D-85748, Germany

[*] Authors to whom any correspondence should be addressed.

E-mail: paul.over@tuhh.de and alexios.michailidis@planqc.eu

OPEN ACCESS

RECEIVED
6 July 2026

REVISED
27 September 2026

ACCEPTED FOR PUBLICATION
13 October 2026

PUBLISHED
27 October 2026

Keywords: quantum operator learning, qubit connectivity, matrix encoding, quantum circuits synthesis

Original content from this work may be used under the terms of the Creative Commons Attribution 4.0 licence.

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Abstract

An efficient implementation of quantum algorithms is often hindered by the lack of efficient primitives for operators and state preparation. This limits both the ability of near-term quantum hardware to simulate complex problems and the potential of fault-tolerant algorithms to achieve practical quantum advantage. To address this, we propose a framework that maps discrete operators onto compact, structured quantum circuits. The resulting variational circuits can be tailored to the connectivity and long-range interaction of the target hardware. The learning process employs backpropagation together with a cost function that efficiently optimizes unitary operators and non-unitary—dense or sparse—operators using only a single ancilla qubit for block encoding. Additionally, we introduce a regularization term that reduces the approximation error. The approach is validated for both quantum mechanical and engineering applications. In the former case, we learn propagators that arise in native quantum problems—such as quantum simulation and quantum chemistry—and achieve improved resource scaling in comparison to standard Suzuki–Trotter expansions. In the latter case, we demonstrate the approach's ability to implement the second-order central finite difference approximation of the Laplace operator—relevant for solving partial differential equations—while improving upon current error metrics. The final example deals with learning a dense, non-unitary operator that arises in the analysis of inviscid potential flow around an airfoil. These results motivate the extension of the framework to a broader class of operator-synthesis problems beyond the prototypical engineering and quantum applications considered here.

1. Introduction

Over the past decades, numerical methods for scientific computing have achieved significant advances [1]. Nevertheless, in-depth analysis requires extensive high-performance computing resources, as classical algorithms exhibit inherent scaling limitations as the system size increases [2]. At the same time, the challenge of further reducing transistor sizes [3, 4] restricts performance improvements in classical hardware. In this context, quantum processing units (QPUs) offer a transformative alternative by leveraging the exponentially large vector spaces present in many-body quantum mechanics. QPUs enable novel computational paradigms in which quantum algorithms might achieve substantial improvements in runtime and memory complexity, while potentially offering superior energy efficiency compared to classical architectures [5].

Despite the great prospects of universal QPUs [6, 7], current hardware limitations prevent the observation of practical quantum advantage. Today's noisy intermediate-scale quantum (NISQ) devices suffer from finite coherence times, gate errors, and restricted qubit connectivity, which constrain achievable circuit depth and significantly increase the cost of multi-qubit operations [8–10]. These hardware constraints limit the complexity of circuits that can be executed reliably, making compact and noise-aware circuit design essential for near-term applications.

© 2026 The Author(s). Published by IOP Publishing Ltd

Even looking beyond NISQ, early fault-tolerant devices face a distinct but equally serious challenge: a disparity in clock speeds between classical processors and logical quantum qubits, typically estimated to span several orders of magnitude [11]. This means that quantum algorithms must deliver sufficient algorithmic speedup to compensate for this inherent slowdown, highlighting the importance of early-stage near-hardware algorithmic design. In practice, asymptotic quantum speedups are frequently offset by large constant prefactors arising from block-encoding overheads, quantum state preparation (QSP) costs, and measurement complexity [11]. In the worst case, the decomposition of generic multi-qubit unitaries and QSP requires resources that scale exponentially with the number of qubits [12], further eroding any theoretical advantage. Practical algorithms must therefore be carefully optimized and tailored to specific target platforms—whether superconducting [8], neutral atom [13], trapped ion [14], or photonic [15] hardware—highlighting the critical importance of early-stage algorithmic development and hardware-software co-design.

Quantum circuit synthesis (QCS) provides a unified framework for translating vectors and matrices into quantum gate sequences, encompassing two central tasks: QSP and the programming of operators. QSP concerns the efficient loading of classical data or quantum states into a quantum register, which is a prerequisite for many quantum algorithms. QSP circuits can be constructed through optimization [16], analytically [17], or deterministically using tensor networks (TNs) [18]. In particular, tensor trains (TTs) [19] -one-dimensional (1D) TNs—enable the efficient representation of weakly correlated data [20–23], which can then be translated into quantum gate sequences [24–29]. The programming of operators constitutes the other critical frontier in QCS, including, for example, analytical algorithms [17], or variational quantum compilation, which involves efficiently executable low-depth parameterized quantum circuits (PQCs) [30–36]. Along this route, [30] propose a hybrid partitioning scheme to distribute large quantum systems into subsystems, focusing on current hardware, where the recombination happens classically. Similarly, [31] advance the design of bricklayer circuits to minimize circuit depth in Hamiltonian simulations. With a particular focus on time evolution and Trotterization strategies, [32–36] also address simulation of Hamiltonians, whereby [36] employs machine-learning (ML) strategies to optimize circuit compilation tailored for two-dimensional (2D) chips.

As a more general alternative to PQCs, latest research determines the quantum gates directly [37–45]. This transformation process entails the additional challenge of ensuring the unitarity of the resulting gates, since—unlike when using PQCs—this is not inherently guaranteed. Such a solution based on TT is demonstrated in [37–39] and involves TT-sweeping strategies combined with polar decomposition, whereas [39] incorporate the adaptivity for 2D chips. Alternatively, other approaches [40–42] leverage TT-based Riemannian optimization to build upon the underlying manifold's structure of unitary operators. Notably, the former works demonstrate a viable alternative to Trotterization for resource optimization by effective compression using TTs [37–42]. In addition, [42] apply Riemannian optimization to engineering problems, thereby demonstrating the versatility and broad applicability of such approaches beyond quantum mechanics. In [42], non-unitary dynamics are embedded into larger unitaries within the optimization procedure; this represents an alternative to the block-encoding technique for handling non-unitary operators, as presented in [43].

Despite the efficient realization of shallow circuits by 1D TN models [42], their limited ability to capture the long-range correlations of the operator restricts their applicability in scientific computing or ML [28]. To address these limitations, [43] propose employing higher-dimensional TNs that better match the 2D connectivity of certain quantum hardware architectures. In this regard, [44, 45] advance QCS techniques for 2D quantum chips by employing higher-dimensional TNs, although the preparation time grows exponentially with system size [44]. Integrating gate-sets to Riemannian optimization is introduced by [46], who utilize brick-wall circuit layouts in conjunction with TTs for real 1D linear nearest neighbor (LNN) architectures. Notably, variational QCS for quantum hardware with higher-dimensional connectivity or long-range programmable interactions—often realized with neutral atoms [13] and trapped ions [14]—remains relatively unexplored, and the identification of scalable classical compression approaches is still an open problem.

Within this work, we explore QCS in a way that allows certain characteristics of quantum hardware to be taken into account during the algorithmic design, specifically by restricting the qubit support of each trainable gate to a user-specified connectivity list (see section 2.2). To this end, we develop a backpropagation-based operator learning algorithm that accommodates a given qubit connectivity via register permutations. The algorithm itself is thus hardware-agnostic, whereas each learned circuit is specific to the supplied topology (see section 2.2.1). The approach is demonstrated for 1D and 2D topologies in section 3 and can incorporate the chip layout as well as long-range and multi-body programmable interactions into the development of optimal quantum gates. Therefore, the current

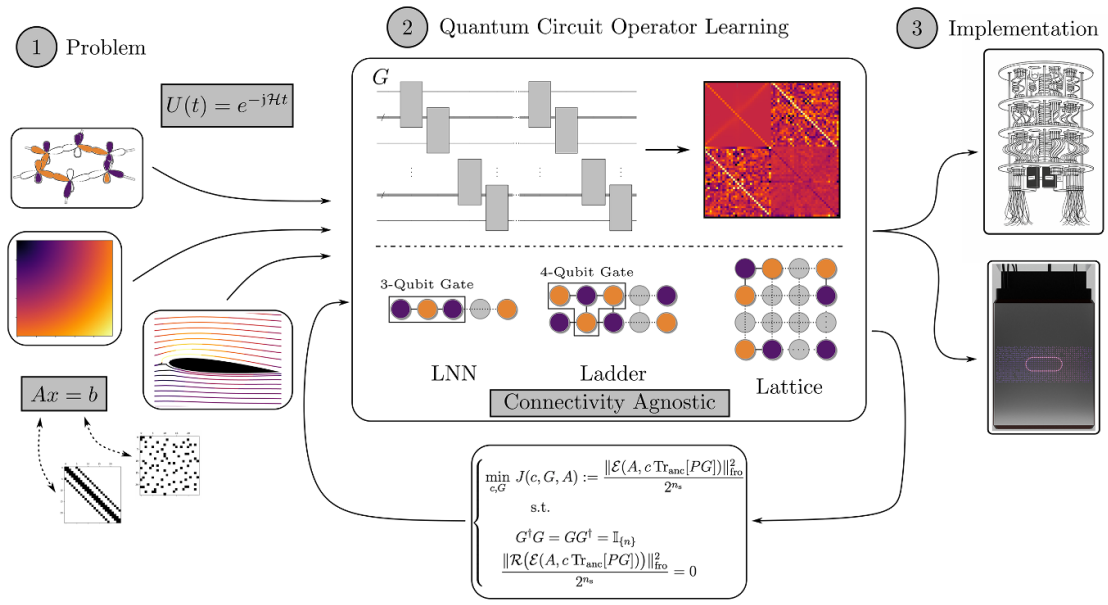


Figure 1. Illustration of the entire quantum operator learning approach.

implementation operates in the full operator space and does not rely on PQC or TT approaches. Non-unitary operators are block-encoded, similar to [42, 43], but only with one additional ancilla qubit. Gate sequences are optimized by combining an automatic differentiation-based steepest descent approach with a projection technique based on singular-value decomposition, as it has been introduced by [27]. This approach is inspired by the TN-based optimization algorithms from [47] and presents an alternative to Riemannian optimization [42]. To homogenize the accuracy across the resulting operators, we introduce an error-based regularization term into the learning procedure, which effectively reduces error outliers. The method can be extended to TN representations using standard techniques, e.g. [39], while preserving the optimization features of the full-space approach.

The article is organized as follows. The mathematical framework, the probabilistic features, and the computational strategy are described in sections 2.1 and 2.2. Section 2.2 also addresses the optimization strategy and analyzes the complexity of the entire framework. Section 3 presents four applications, from quantum mechanics, quantum chemistry, and engineering sciences. A summary and conclusions are presented in section 4. We denote arbitrary discrete vectors and matrices by Latin letters, and normalized vectors (quantum states) are indicated by the Dirac (bra-ket) notation. For representations of quantum systems, we employ the *big-endian* convention. The quantum applications are based on second quantization, whereas the engineering applications employ amplitude encoding.

2. Methodology

In the following, we present the mathematical and computational framework that forms the basis of the proposed methodology for mapping discrete, potentially non-square operators onto quantum hardware. Learned operators obtained within this framework can be employed in a variety of quantum computing tasks, including quantum signal processing [43], quantum linear solvers [48] as an alternative to Trotterized time-evolution circuits [49], incorporations into variational quantum algorithms [50–52], and fault-tolerant algorithmic frameworks [53–55]. To this end, we consider a generic problem and outline a strategy to derive a gate-level representation optimized for the target quantum hardware through efficient operator learning. An overview of the programming pipeline is illustrated in figure 1.

The synthesis begins with discretizing the target problem (see figure 1: ①), arising in application fields such as quantum mechanics, quantum chemistry, or engineering. The discretization yields an operator A, e.g. representing a linear system $Ax = b$ or a quantum time-evolution operator $A \approx U(t) = \mathrm{e}^{-j\mathcal{H}t}$. Since A is generally non-unitary, it is represented on quantum hardware via block-encoding into a larger unitary operator (see section 2.1). The encoding of A into a quantum circuit is strongly influenced by hardware features (see figure 1: ②), including chip architecture, qubit connectivity, gate size,

and decoherence time. Current devices typically feature one- or 2D qubit layouts with restricted connectivity, such as LNN, ladder, or square-lattice topologies.

The contribution presented here covers chip interconnection and layout, which are incorporated into the development of optimal quantum gates. These differences are illustrated by the example in figure 1, which depicts a 1D configuration (LNN) and a corresponding 3-qubit gate implementation of nearest neighbor interactions. Next to it, the ladder structure with a 4-qubit gate, relying also exclusively on nearest-neighbor interactions, exhibits comparable maximum information path lengths. In the case of an ideal, full inter-connectivity (all-to-all connectivity) [6], arbitrarily complex, i.e. arbitrarily large, qubit gates can be represented through local 2-qubit nearest-neighbor interactions. Notably, very few studies have addressed this aspect to date. Most studies are based on a simple 1D LNN configuration, which does not always reflect the reality of existing hardware. The method presented here aims to facilitate the development of optimal quantum gates for given hardware configurations. To this end, the study also investigates whether this significantly more flexible approach offers advantages in capturing correlations and structures.

In the following, connectivity generally relates to the topology, the chip inter-connection, as well as short- or long-range qubit interactions. The cost function J quantifies the approximation of the implemented operator G to the target operator A, which is approximated by a sequence of fixed-body qubit gates assembled as a sequential ansatz. The range of the gates can thereby vary from nearest neighbor to long-range interaction. In the final step (see figure 1: ③), the derived gate sequence is brought to the target hardware, e.g. being superconducting or ion-trap machines. Specific considerations of individual steps within the operator learning protocol follow in the next sections. A detailed description of the mathematical framework is given in section 2.1. The learning strategy, the sequential ansatz, and the optimization approach are part of the computational strategy, which is discussed and exploited along the algorithm in section 2.2.

2.1. Mathematical framework

We consider a problem described by a discrete operator $A \in \mathbb{C}^{N\times M} : N, M \in \mathbb{N}$, with an arbitrary structure. As a preprocessing step, the non-square operator is embedded into a larger square operator by setting $N = M = \max(N, M)$, where additional entries are filled with zeros. To implement the operator on a quantum device, at least $n_s = \lceil \log_2 N \rceil$ system qubits are required, so that the dimension of the qubit Hilbert space, 2^{n_s}, is at least as large as the operator dimension N. The number of qubits required depends on the encoding, and additional qubits are needed when the operator is not unitary ($A^\dagger A \neq \mathbb{I} \neq AA^\dagger$). More precisely, a non-unitary target operator A requires an embedding into a larger vector space, a procedure known as block encoding [10]. Block encoding is generally characterized by the normalization factor c, the number of ancillary qubits n_a, and an approximation error τ. The block-encoding on the full register $n = n_s + n_a$ is defined as

$$\left\| A - c\,\mathrm{Tr}_{\mathrm{anc}}\left[PG\right] \right\|_{\mathrm{fro}} \leqslant \tau\,, \quad P = |0^{n_a}\rangle\langle 0^{n_a}| \otimes \mathbb{I}_{\{n_s\}}\,, \tag{1}$$

where $G \in \mathbb{C}^{2^n \times 2^n}$ is a unitary operator, $\mathbb{I}_{\{n_s\}}$ is the identity matrix of size $2^{n_s} \times 2^{n_s}$, $\mathrm{Tr}_{\mathrm{anc}}[\cdot]$ is the partial trace over the ancilla sub-system and $\|\cdot\|_{\mathrm{fro}}$ indicates the Frobenius norm. The complete quantum register $|0^{n_a}\rangle \otimes |\psi\rangle$ consists of the ancilla system ($|0^{n_a}\rangle$) and the computational state ($|\psi\rangle$), and the target operator is encoded into the space prescribed by the projector P. Since G is unitary, its block $\mathrm{Tr}_{\mathrm{anc}}[PG]$ must be a contraction, i.e. its largest singular value cannot exceed unity. This condition is ensured by the normalization factor $c \geqslant c_{\mathrm{ideal}}$, where $c_{\mathrm{ideal}} = \sigma_{\max}(A) = \|A\|_2$ denotes the spectral norm of A, such that A/c_{ideal} has a largest singular value of exactly one. In the absence of approximation errors ($\tau = 0$), an exact encoding $\mathrm{Tr}_{\mathrm{anc}}[PG] = A/c$ is possible for any $c \geqslant c_{\mathrm{ideal}}$, with $c = c_{\mathrm{ideal}}$ being the smallest admissible choice. The normalization factor, however, suppresses the post-selection success probability since the approximation $A \approx c\,\mathrm{Tr}_{\mathrm{anc}}[PG]$ implies $\|\mathrm{Tr}_{\mathrm{anc}}[PG]\,|\psi\rangle\|^2 \propto c^{-2}$ [56], where $\|\cdot\|$ denotes the Euclidean norm of a vector. Therefore, a small normalization factor $c \approx c_{\mathrm{ideal}}$ is desirable, which motivates the stabilizing term introduced in equation (7). Since the success probability also depends on the state [56], we report its average over all input states, i.e.

$$p\left(PGP\right) = \frac{\|PGP\|_{\mathrm{fro}}^2}{2^{n_s}}. \tag{2}$$

If A is unitary, as in the case of a Hamiltonian simulation, the block encoding using ancillas is not required ($n_a = 0$), the projector P is the identity matrix $\mathbb{I}_{\{n\}}$, $c = c_{\mathrm{ideal}} = 1$, and $p(A) = p(G) = 1$.

In this work, we consider a single-qubit block encoding $n_a = 1$. We note that our procedure can be realized with any number of qubits, unlike other approaches [42, 43] which require a problem-dependent number of ancilla qubits. We assume that G is a sequence of parameterized quantum gates

expressed by a generic ansatz introduced in section 2.2. We aim to find solutions to equation (1) with small truncation error τ. For this reason, we define the residual for two matrices A, B as

$$\mathcal{E}(A,B) = A - B \tag{3}$$

and the relative error as

$$\varepsilon(A,B) = \frac{\|\mathcal{E}(A,B)\|_{\text{fro}}}{\|A\|_{\text{fro}}}. \tag{4}$$

Our approach optimizes only the average error. Norms that capture the worst-case error, such as the spectral norm, generally require iterative procedures within tensor-network implementations, whose efficiency and convergence cannot be guaranteed at the scale required for large systems. Nevertheless, appendix A discusses the worst-case error, as measured by the spectral norm, for the numerical results presented in this study.

The optimization problem for learning the target operator A is defined as

$$\begin{cases} \min\limits_{c,G} J(c,G,A) := \frac{\|\mathcal{E}(A,c\,\text{Tr}_{\text{anc}}[PG])\|_{\text{fro}}^2}{2^{n_s}} & (5a) \\ \quad \text{s.t.} & \\ \quad G^\dagger G = GG^\dagger = \mathbb{I}_{\{n\}} & (5b) \\ \quad \frac{\|\mathcal{R}(\mathcal{E}(A,c\,\text{Tr}_{\text{anc}}[PG]))\|_{\text{fro}}^2}{2^{n_s}} = 0\,, & (5c) \end{cases}$$

where 2^{n_s} realizes a normalization, making sure that the objective remains finite with increasing system sizes, and $\mathcal{R}$ is a variant of ℓ_2-regularization.

The purpose of regularization is to limit error outliers, maintaining constant error magnitudes throughout the optimization, or in other terms, that a non-zero residual $\mathcal{E}$ should be sufficiently smooth. This is achieved by penalizing the objective with a filter contribution evaluated for the entries of the matrix $\mathcal{E}$. The filter is defined as a five-point stencil on two dimensions with periodic boundary conditions, such that the regularization term defined at location ij is

$$\mathcal{R}(\mathcal{E})_{ij} = \left(\mathcal{E}_{i-1j} + \mathcal{E}_{ij-1} - 4\mathcal{E}_{ij} + \mathcal{E}_{i+1j} + \mathcal{E}_{ij+1}\right)_{ij}. \tag{6}$$

For implementation, the constrained problem (5) is reformulated as an unconstrained one. First, the evolution of a closed quantum system is unitary. Each gate in the circuit must therefore correspond to a unitary operator, which in turn ensures that the full circuit is unitary, as expressed in equation (5b). We enforce this unitary constraint on the individual gates by combining a relaxation strategy with a retraction onto the unitary manifold, as described in detail in section 2.2. Second, the regularization (5c) is handled by a penalty approach, incorporating the constant penalty term $\rho \in \mathbb{R}^+$ [57]. As a result, the unconstrained optimization problem achieves optimality with respect to its contributions, and is explicitly described by

$$\min_{c,G} \mathcal{J}(c,G,A,\rho,\mu) := \frac{\|\mathcal{E}(A,c\,\text{Tr}_{\text{anc}}[PG])\|_{\text{fro}}^2}{2^{n_s}} + \rho\frac{\|\mathcal{R}(\mathcal{E}(A,c\,\text{Tr}_{\text{anc}}[PG]))\|_{\text{fro}}^2}{2^{n_s}} + \mu c^2. \tag{7}$$

The additional term μc^2, with weight $\mu \in \mathbb{R}^+$, penalizes large normalization factors. Without this term, the optimization can converge to solutions with $c \gg c_{\text{ideal}}$. An example is presented in appendix B. For a suitably chosen μ, we did not observe a significant loss in approximation accuracy, while such solutions are avoided.

The solution of equation (7) is characterized by a stationary point, given by optimal controls c^*, G^*. At this point, the partial derivatives of $\mathcal{J}$ w.r.t the control vanish for all variations δc, and the admissible variations δG at G, viz.,

$$\frac{\partial \mathcal{J}}{\partial c}\delta c = 0 \;\forall \delta c, \quad (\nabla_{\text{R}}\mathcal{J}, \delta G) = 0 \;\forall \delta G \in T_G U(2^n), \tag{8}$$

with $(\cdot,\cdot)$ being the respective inner product of the underlying unitary manifold $U(2^n)$, and ∇_{R} the Riemann gradient, and $T_G U(2^n)$ is the tangent space at G [58, 59]. We employ a retraction based on the singular value decomposition (SVD) (see algorithm 2), enforcing the constraint in equation (5b). This system of first-order optimality conditions can be directly associated with each contribution to the

cost function. For example, the solution for c is obtained by imposing the first-order optimality condition associated with the quadratic cost function defined in equation (7) and solving equation (8) w.r.t c. This results in an analytical expression for the normalization constant, which is

$$c^* = \mathrm{Re}\left(\frac{\mathrm{Tr}_{\mathrm{sys}}\left[A^\dagger \mathrm{Tr}_{\mathrm{anc}}\left[PG\right]\right] + \rho\,\mathrm{Tr}_{\mathrm{sys}}\left[\mathcal{R}\left(A\right)^\dagger \mathcal{R}\left(\mathrm{Tr}_{\mathrm{anc}}\left[PG\right]\right)\right]}{2^{n_s}\mu + \|PGP\|_{\mathrm{fro}}^2 + \rho\left\|\mathcal{R}\left(\mathrm{Tr}_{\mathrm{anc}}\left[PG\right]\right)\right\|_{\mathrm{fro}}^2}\right), \tag{9}$$

where $\mathrm{Re}(\cdot)$ indicates the real-part. As evident from the equation above, the optimal c^* depends directly on G and can therefore be eliminated as a degree of freedom. The computational treatment of first-order optimality is further addressed in section 2.2.1. An investigation of the performance of the regularizer $\mathcal{R}$ is part of the numerical studies, presented in section 3.

2.2. Computational strategy

Building on the mathematical framework introduced above, this section presents the computational implementation of the proposed operator learning approach. We employ a hybrid strategy in which gate primitives are trained on classical hardware and the resulting operators are prepared for subsequent realization on quantum devices. Accordingly, the focus here is on the treatment of the classical optimization problem, with particular emphasis on the hierarchical optimization scheduler of section 2.2.1 and the algorithm's complexity specifications of section 2.2.2.

Since our algorithm is hardware-agnostic, we employ as a default ansatz a general circuit without constraints in the layout, illustrated in figure 2. Thereby, we ensure to fit any hardware into the same computing layout, which allows adapting to connectivity, long-range quantum gates, or other device-specific characteristics. Furthermore, we consider a standard n-qubit register $|q\rangle = |q_1\rangle \otimes \cdots \otimes |q_n\rangle \in \mathbb{C}^{2^n}$ and the trainable quantum circuit defined as

$$G = \prod_{j=1}^{l}\prod_{i=1}^{m^j} G_i^j, \tag{10}$$

where the gates are organized into l layers, each containing m^j gates. Each trainable gate G_i^j is a unitary operator, satisfying the unitary condition $(G_i^j)^\dagger G_i^j = \mathbb{I}$, where $\mathbb{I}$ is the respective identity matrix, and that acts non-trivially only on a subset of qubits $\mathcal{G}_i^j = \{g_1, \ldots, g_r\} \subseteq \{1, \ldots, n\}$.

More precisely, G_i^j is the canonical embedding into the full n-qubit Hilbert space of a local operator $\widetilde{G}_i^j$ acting on the r^j qubits indexed by $\mathcal{G}_i^j$, while acting as the identity on all remaining qubits. The gate G_i^j is often referred to as an r^j-body gate. A further classification of the gates is provided by their range, defined as the maximum distance between any two qubits in the support $\mathcal{G}_i^j$. This distance is measured with respect to the placement of the qubits in the underlying hardware architecture. Consequently, the gate sizes may range from two-qubit gates up to many-body gates acting on the entire register. To this end, the variables l, m^j and r^j are explicitly set with $\mathcal{G}_i^j$ prescribed by the user. Thereby, different connectivity patterns can be straightforwardly implemented to model, e.g. long-range interactions or two-dimensional lattice (2DL) structures. For example, the gate $G_1^1 = \widetilde{G}_1^1 \otimes \mathbb{I}_3 \otimes \ldots \otimes \mathbb{I}_n$ is illustrated in figure 2 where $\mathbb{I}_i \in \mathbb{R}^{2\times 2}$ is the identity matrix associated to qubit i. The gate G_1^1 acts non-trivially only on the first two qubits of the register, corresponding to the subset $\mathcal{G}_1^1 = \{1, 2\}$. Therefore, it is a two-body gate ($r^1 = 2$), and for a linear hardware layout following the qubit enumeration, it is a range–2 (or nearest-neighbor) gate. Instead, the last gate $G_{m^l}^l$ acts on the first and last two qubits of the register given by the subset $\mathcal{G}_{m^l}^l = \{1, n-1, n\}$, which makes it an r^l-qubit gate ($r^l = 3$), and for linear hardware, therefore a long-range interacting gate. The number of gates m^j in each layer and the number of qubit supports r^j, i.e. the number of qubits each gate acts on (assumed to be uniform within a layer) can be chosen independently for each layer. This allows the approach to be adapted to external constraints, such as the connectivity or native gate sizes of a given hardware platform.

In the following section, we describe how the trainable quantum circuit G is learned by optimizing the non-trivial component of each trainable gate.

2.2.1. Strategy

This section focuses on the optimization strategy by which the gates G—assembled as introduced in the previous section—are iteratively updated to minimize the objective $\mathcal{J}$. Rather than utilizing Riemannian

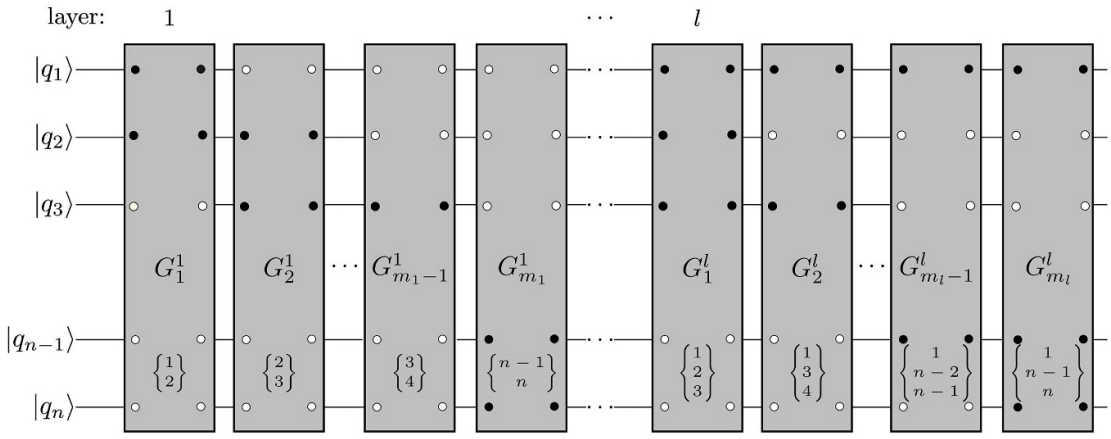


Figure 2. Illustration of an l-layer trainable quantum circuit ansatz G, see equation (10). For each gate, the subset of qubits on which it acts non-trivially, denoted by $\mathcal{G}_i^j$, is indicated by the closed vertices, or equivalently by its connectivity list illustrated with curly brackets. The first layer is composed of $m^1 = n-1$ nearest-neighbor two-qubit gates, whereas the lth layer consists of $m^l = n-2$ three-qubit gates, of which all but the first are long-range gates.

gradient [59], we adopt automatic differentiation [27] that can be naturally extended to TN representations [60]. To this end, the presented approach employs a gradient-based optimizer, where the derivatives are approximated by backpropagation through the computational graph—involving the computation of the environment tensor—as it is common in ML [61].

To reduce the computational cost, we perform the optimization at the level of the local gate primitives $\widetilde{G}_i^j$, rather than using their full Hilbert-space embeddings G_i^j. This avoids manipulating exponentially large matrices whenever only a few qubits are involved, leading to a substantial reduction in computational overhead (see section 2.2.2). For clarity of presentation, however, we formulate the algorithm below in terms of the embedded gates G_i^j. We refer the reader to [39] for the efficient gradient-evaluation procedure based on the non-embedded gates. This modification affects only the implementation of the gradient computation, and the structure of the remaining algorithm stays unchanged.

The update of the local gate primitives then follows gate-wise in the steepest descent direction as

$$\bar{G}_i^j = \widetilde{G}_i^j - \eta \frac{\partial \mathcal{J}}{\partial \widetilde{G}_i^j} \qquad \forall\, i \in \left[1, m^j\right],\, j \in [1, l]\,, \tag{11}$$

where $\bar{G}_i^j$ are non-unitary local latent gates, η is the learning rate (stepsize), and $\partial\mathcal{J}/\partial\widetilde{G}_i^j$ is the Euclidean derivative. Using the Euclidean update violates the unitary constraint in equation (5b), such that all $\bar{G}_i^j$ require to be corrected. This is done by a retraction, realized by performing an SVD on each latent gate primitive, viz.,

$$\mathrm{SVD}\left(\bar{G}_i^j\right) = U\Sigma V^\dagger \quad \rightarrow \quad \widetilde{G}_i^j = UV^\dagger\,, \tag{12}$$

where we replace the diagonal matrix of singular values Σ with the identity $\mathbb{I} \in \mathbb{R}^{2^{r^j}\times 2^{r^j}}$, retracting the latent gate back onto the unitary group $U(2^{r^j})$. This retraction is an approximation to the Riemannian exponential map [58]. Although this retraction of the operator back onto the manifold of the unitary group introduces an error, $\widetilde{G}_i^j$ is, by construction, the unitary matrix closest to $\bar{G}_i^j$ in the Frobenius norm [62]. More specifically, for a gate acting on r^j qubits, $\widetilde{G}_i^j \in U(2^{r^j})$. Since $\min_{G\in U(2^{r^j})} \|\mathrm{e}^{\mathrm{i}\phi}\bar{G} - G\|_{\mathrm{fro}}^2 = \min_{G\in U(2^{r^j})} \|\bar{G} - G\|_{\mathrm{fro}}^2$, the gate's global phase $\phi \in \mathbb{R}$ does not affect the retraction error. The approach can be understood as a polar decomposition and is inspired by the algorithm [47]. For QSP, this approach has been recently applied in [27], where [29] highlighted improvements over Riemannian descent methods, further motivating the protocol presented below.

With the corrected local gates, the operator programming protocol first canonically embeds the local gate primitives $\widetilde{G}_i^j$ (r^j-qubit gate) into the corresponding n-qubit Hilbert space, resulting in $\widehat{G}_i^j$. Secondly, a quantum permutation $S(\mathcal{G}_i^j)$ is applied to align with the given connectivity and hardware support $\mathcal{G}_i^j$, viz., $S(\mathcal{G}_i^j)^\dagger \widehat{G}_i^j S(\mathcal{G}_i^j) = G_i^j$. In a last step, the gate G results from equation (10), which allows to compute the normalization c and to finally evaluate the cost-function value $\mathcal{J}$.

Algorithm 1 Hierarchical optimization protocol.

Require: l, `tol`, `maxit`, $\mathcal{G}_i^j$
1: $m^j \leftarrow \mathcal{G}_i^j$ {Initialize gate counts}
2: **for** $j = 1$ to l **do**
3: **for** $i = 1$ to $\min_{k=1,\dots,j} m^k$ **do**
4: Optimize $(\widetilde{G}_i^1, \dots, \widetilde{G}_i^j)$ by executing algorithm 2
5: **end for**
6: Jointly optimize $\{\widetilde{G}_i^k : 1 \leqslant k \leqslant j,\ 1 \leqslant i \leqslant m^k\}$ by executing algorithm 2
7: **end for**
8: Train all gates $\{\widetilde{G}_i^j : 1 \leqslant j \leqslant l,\ 1 \leqslant i \leqslant m^j\}$ by executing algorithm 2

The optimization procedure is subordinate to a hierarchical optimization approach, which is based on prioritizing the training of the most significant qubits first. A differential relevance among individual qubits is particularly present using amplitude encoding, commonly employed for running scientific computing problems on QCs. The hierarchical optimization procedure is detailed in algorithm 1 and builds upon [63].

Algorithm 1 builds up the circuit layer by layer. For each layer j, the gates with index i are first optimized together with the gates of the same index in all preceding layers (see algorithm 1—line 4), before all gates of layers 1 to j are optimized jointly (see algorithm 1—line 6). Once the last layer has been added, all gates are trained in a final global optimization (see algorithm 1—line 8). Without the hierarchical stages, only the final global optimization is performed, see algorithm 1—line 8. Throughout, gates that have already been optimized but are not part of the current step are held fixed. To reduce computational effort, all gates that have not yet been optimized are excluded from the evaluation, so that the operator learning protocol starts with a one-gate quantum circuit. Within each call of algorithm 2 (see algorithm 1—lines 4, 6 and 8), the optimization is started for a user-defined maximum number of iterations `maxit` and is terminated based on a user-defined tolerance `tol` on the objective's derivative. The underlying optimization procedure is summarized in algorithm 2.

For the gradient-based optimization denoted in algorithm 2—line 16, we employ two algorithms, which are either the L-BFGS [57] or the ADAM [64] algorithm. The ADAM optimizer is particularly advantageous for enhancing barren plateau (BP) robustness, while problems that are more likely to be convex are addressed using the Quasi-Newton approach (L-BFGS), to reduce computational costs. In both cases, we incorporate informed initial guesses whenever available, otherwise, the method initializes the gates with identities. By omitting unused layers during backpropagation, we significantly reduce computational costs. In combination with hierarchical learning, we observed that L-BFGS achieves higher optimization accuracy and restricted our implementation to this optimizer. The following section (see section 2.2.2) discusses in detail the computational complexity and scalability of the proposed protocol.

To improve BP resilience, we incorporate multiple strategies, comprehensively reviewed in [65]. First, the hierarchical optimization algorithm presented in algorithm 1 reduces the optimization's sensitivity by avoiding sub-optimal local minima encountered in non-convex optimization landscapes. This advantage is elaborated upon in [28], which implements a sophisticated strategy to mitigate BPs and achieves improved accuracy. Second, the inclusion of developed regularization techniques shows promise in enhancing robustness against BP phenomena. Further options for resolving BP issues include optimized initial guesses, leveraging classical shadows [66], or using classically motivated surrogate models [67, 68] closely related to this work.

2.2.2. Complexity

This section analyzes the computational complexity of the proposed method. In particular, we focus on the dominant costs arising in algorithms 1 and 2, and their implications for the subsequent quantum hardware implementation. For comparison with the existing literature, we consider the worst case of a constant number of gates $m = \max_j m^j$ per layer and a constant gate size $r = \max_j r^j$ throughout the circuit, which yields an upper bound on the resources required for the general ansatz. Throughout, r is assumed to be as small as the desired expressivity permits, in particular sufficiently smaller than n, since the cost of implementing an r-qubit gate grows exponentially with r. Further, we assume the global optimization (see algorithm 1—line 8) as a worst-case scenario since this algorithmic step incurs the highest computational costs. Therefore, we concentrate on estimating the cost of algorithm 2.

Algorithm 2 Optimization procedure used for efficient operator learning.

REQUIRE ρ, μ, $i_{\max}$, $j_{\max}$, A, n_s, `tol`, `maxit`, $\mathcal{G}_i^j$
1: **if** $A^\dagger A = AA^\dagger = \mathbb{I}_{n_s}$ **then**
2: $n_a = 0$ and $P = \mathbb{I}_{n_s}$
3: **else**
4: $n_a = 1$ and $P = |0^{n_a}\rangle\langle 0^{n_a}| \otimes \mathbb{I}_{n_s}$
5: **end if**
6: $k = 0$
7: Assemble $S(\mathcal{G}_i^j)$ based on connectivity list $\mathcal{G}_i^j$
8: **while** $\|\partial\mathcal{J}/\partial G\| >$ tol and $k <$ maxit
9: Canonically embed $\widetilde{G}_i^j$ as $\widehat{G}_i^j$
10: Compute $G_i^j = S^\dagger(\mathcal{G}_i^j)\widehat{G}_i^j S(\mathcal{G}_i^j)$ {Apply connectivities}
11: Compute $B = \mathrm{Tr}_{\mathrm{anc}}[PG]$
12: Compute $c = \mathrm{Re}\Big(\big(\mathrm{Tr}_{\mathrm{sys}}[A^\dagger B] + \rho\,\mathrm{Tr}_{\mathrm{sys}}[\mathcal{R}(A)^\dagger\mathcal{R}(B)]\big)/\big(\mu 2^{n_s} + \|B\|_{\mathrm{fro}}^2 + \rho\|\mathcal{R}(B)\|_{\mathrm{fro}}^2\big)\Big)$
13: Evaluate $\mathcal{J} = \|\mathcal{E}(A, cB)\|_{\mathrm{fro}}^2/2^{n_s} + \rho\|\mathcal{R}\big(\mathcal{E}(A, cB)\big)\|_{\mathrm{fro}}^2/2^{n_s} + \mu c^2$
14: **for all** G_i^j, where $i \leqslant i_{\max}$ and $j \leqslant j_{\max}$ **do**
15: Compute required derivatives by backpropagation $\partial\mathcal{J}/\partial\widetilde{G}_i^j$
16: Apply Euclidean update $\bar{G}_i^j = \widetilde{G}_i^j - \eta\,\partial\mathcal{J}/\partial\widetilde{G}_i^j$ using a local gradient based approach
17: Enforce constraint (5b): $\mathrm{SVD}(\bar{G}_i^j) = U\Sigma V^\dagger \Rightarrow \widetilde{G}_i^j = UV^\dagger$ {Retract onto $U(2^r)$ manifold}
18: **end for**
19: $k = k + 1$
20: **end while**

In the operator encoding protocol, each local gate primitive belonging to the unitary group $U(2^r)$ is parametrized by a $2^r \times 2^r$ matrix with complex-valued entries. The corresponding control vector λ is constructed by concatenating the parameters of all gates, ordered layer-wise from top to bottom and, within each layer, from left to right, terminating at G_l^m. Since each matrix entry can be in general complex-valued, each entry is represented by two real parameters, i.e. $\lambda_i + \lambda_{i+1}\mathrm{j}$, with j being the imaginary unit. Consequently, the number of parameters per gate results in 2^{2r+1}. This number reduces by half when the target operator is restricted to real values. If we now assume l layers with m gates per layer and a complex-valued operator to be learned, the total number of degrees of freedom scales with $\mathcal{O}(lm2^{2r+1})$.

For evaluation of the cost function, these local gate primitives $\widetilde{G}_i^j$ are canonically embedded into the group $U(2^n)$ to perform the matrix multiplication given in equation (10). This multiplication is subject to the computational cost of $\mathcal{O}(2^{3n})$. If local primitives $\widetilde{G}_i^j$ are used instead of embedded gates G_i^j, the contraction cost can be improved to $\mathcal{O}(2^{2n}2^r)$. As this contraction is performed for l layers and m gates the contraction complexity is $\mathcal{C}_{\mathrm{cont.}} = \mathcal{O}(2^{2n}2^r lm)$.

Another central property of the optimization procedure described in algorithm 2—line 17 is the enforcement of the constraint (5b) by the retraction on the unitary manifold after each gradient-based update of the local gates $\widetilde{G}_i^j$. This retraction requires computing an SVD for each gate modification, with the computational effort for m gates and l layers scaling as $\mathcal{C}_{\mathrm{SVD}} = \mathcal{O}(2^{3r}lm)$. Accounting for all contributions during the optimization, the total cost per k iterations is

$$\mathcal{C}_{\mathrm{opt}} = k\,(\mathcal{C}_{\mathrm{SVD}} + \mathcal{C}_{\mathrm{cont.}}) = klm\,\mathcal{O}\left(2^{3r} + 2^{2n}2^r\right). \tag{13}$$

This computational effort is accumulated across the hierarchical optimization strategy (see algorithm 1). However, this increase is partially mitigated by reducing the degrees of freedom within early hierarchical steps, as discussed in section 2.2.1.

Equation (13) indicates an exponential scaling with the system size n and the gate size r. The scaling with respect to n is intrinsic to any classical approach that does not rely on TN. In TN implementations, the algorithm does not scale exponentially with the system size but instead polynomially with the complexity of the TN described by the auxiliary dimension [39]. However, the presented protocol can rely on small multi-qubit gates and therefore can meet the hardware requirements. Therefore, individual problems come with individual gate complexities, for which the computational resources are estimated by equation (13).

For estimating the gate complexity $\mathcal{C}$gate, we assume that decomposing a general r-qubit gate into CNOTs requires on the order of 4^{r-1} gates [69]. The total CNOT count of the circuit is then

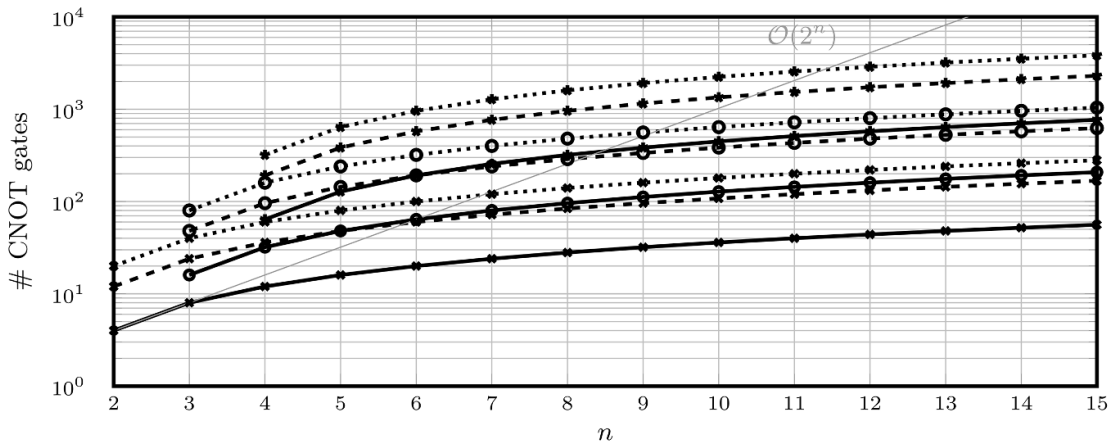


Figure 3. Gate complexity $\mathcal{C}_{\text{gate}}$ over number of qubits for layers $l = \{1\,(\text{solid}), 3\,(\text{dashed}), 5(\text{dotted})\}$ (line-type) and gate sizes $r = \{2\,(\times), 3\,(\circ), 4\,(\star)\}$ (symbol), see equation (14).

approximately

$$\mathcal{C}_{\text{gate}} \approx \sum_{i,j} 4^{r-1}\,\text{gates}, \tag{14}$$

which is shown in figure 3 for different values of l and r. For a fixed or at most polylogarithmically growing number of layers l, this count scales polylogarithmically with the operator dimension N. Note that in this figure, the combination of $l = 1\,(\text{solid})$ with $r = 2\,(\text{cross})$ is equivalent in gate resources to a Suzuki–Trotter (ST) first-order decomposition, see section 3.1.

Considering non-unitary operations, the approaches in [42, 43] require multiple ancilla qubits, depending on the auxiliary dimension of the employed TN approach. Alternatively, the Linear Combination of Unitaries (LCU) algorithm unitary decomposes a matrix A in κ unitaries employing $n_{\text{a}} = \lceil \log_2(\kappa) \rceil$ ancillas. Instead, the presented framework solely requires one ancilla qubit $n_{\text{a}} = 1$ for non-unitary embeddings. Despite differences in ancilla requirements, the success probability derived in equation (2) matches averaging the probability in [43], which shows an alignment with the LCU characteristics, as confirmed here.

3. Applications

This section demonstrates the capabilities of the presented operator learning protocol across four different applications, covering quantum and engineering problems. The learning process is shaped by specifications of the underlying quantum hardware, which in turn determine the final circuit architecture. To this end, key considerations within this demonstration include the availability of short- and long-range interaction in the quantum gates and the qubit connectivity defined by the chip architecture. First, the section 3.1 benchmarks our framework by learning propagators arising in native quantum problems such as the simulation of quantum many-body systems (see section 3.1.1) and quantum chemistry (see section 3.1.2). In addition, both cases cover comparisons to standard ST expansions. Second, engineering applications are specifically addressed in section 3.2, where we initially discuss the learning of a second-order finite difference approximation (see section 3.2.1) and then apply the learning protocol to a problem motivated by computational fluid dynamics (CFD) (see section 3.2.2). Note that the two engineering applications employ amplitude encoding together with the application of the regularization term given in equation (6).

To keep the benchmark comprehensive, we restrict ourselves to constrained layouts, namely 1D LNN and 2DL topologies. Instead of a simple line array topology (LNN), where inner qubits have two direct neighbors, the inner qubits in the 2DL chip have four neighbor interconnections (see figure 1). Additionally, the quantum chemistry application (see section 3.1.2) specifically addresses long-range interacting gates. For clarity, the hardware specifications for each case are defined individually in the corresponding section. The complexity and the accuracy of the framework are evaluated based on the relative error, see equation (4), and the probabilistic characteristics. Appendix A assesses the ratio of the spectral norm error $\|\mathcal{E}(A, c\,\text{Tr}_{\text{anc}}[PG])\|_2$ to the Frobenius norm error $\|\mathcal{E}(A, c\,\text{Tr}_{\text{anc}}[PG])\|_{\text{fro}}$ for the final

result of the following applications. Furthermore, all simulations have been performed on classical CPU hardware using PyTorch, assuming perfect qubit simulation.

3.1. Quantum applications

In this section, we benchmark our algorithm on the compression of the propagator (time-evolution operator), of the time-independent Schrödinger equation,

$$|\psi(t)\rangle = U(t)\,|\psi(0)\rangle\,, \qquad U(t) = \mathrm{e}^{-\mathrm{j}\mathcal{H}t}\,, \tag{15}$$

where $t \geqslant 0$ denotes time, $|\psi(t)\rangle \in \mathbb{C}^{2^n}$ is a quantum many-body state encoded on n qubits, and $\mathcal{H}: \mathbb{C}^{2^n} \to \mathbb{C}^{2^n}$ is a time-independent many-body Hamiltonian acting on the corresponding Hilbert space. Since the Hamiltonian is Hermitian, $\mathcal{H} = \mathcal{H}^\dagger$, the propagator $U(t)$ is unitary. Accordingly, the propagator $A \approx U(t)$ is also unitary (st. $A^\dagger A = AA^\dagger = \mathbb{I}$), and, therefore the operator does not require a block-encoding to be applied on quantum hardware. As this is valid for all the applications, the success probability stabilizer is $\mu = 0$. These applications are based on second quantization and not on amplitude encoding as they represent quantum many-body systems. Interactions are inherently local or decay rapidly with the distance between the qubits. In this case, we disable the regularizer in the cost function (see $\rho = 0$ in equation (7)).

Our benchmark problems are chosen to represent two important classes of applications: quantum many-body physics (see section 3.1.1) and a quantum chemistry application (see section 3.1.2). Although both paradigms are inherently quantum, they exhibit important structural differences. These include variations in the interaction range, the presence of additional symmetries, and distinct connectivity structures. As we discuss below, these differences lead to different requirements and strategies in the optimization procedure. In both cases, we compare our findings with Trotterized time evolution schemes, often employed to simulate Hamiltonian dynamics on digital quantum hardware.

3.1.1. Quantum simulation

To benchmark our approach in quantum simulations, we consider the 1D transverse-field Ising model with an additional longitudinal field, viz.

$$\mathcal{H} = -g_{zz}\sum_{i=1}^{n-1} Z_i Z_{i+1} - g_x \sum_{i=1}^{n} X_i - g_z \sum_{i=1}^{n} Z_i\,, \tag{16}$$

where n again denotes the number of qubits. The operators $Z = |0\rangle\langle 0| - |1\rangle\langle 1|$ and $X = |1\rangle\langle 0| + |0\rangle\langle 1|$ correspond to the Pauli-Z and Pauli-X operators acting on the ith qubit, respectively. In our numerical experiments, we consider system sizes $n = 4$–12 and set the coupling parameters to $g_{zz} = 1.0$, $g_x = 0.8$, and $g_z = 0.3$. These values are chosen to enable direct benchmarking against [42]. To set the time scale in our simulations, we normalize the Hamiltonian by its spectral norm density $\mathcal{N} = \|\mathcal{H}\|_2/n$, and replace $\mathcal{H}$ by $\mathcal{H}/\mathcal{N}$. This ensures that the rescaled Hamiltonian has unit spectral norm density $\|\mathcal{H}\|_2/n = 1$.

As benchmarks for compressing the propagator $U(t) = \mathrm{e}^{-\mathrm{j}\mathcal{H}t}$, we employ the first- and second-order ST [49] expansions

$$U_{\mathrm{ST}_1}(\delta t) = \prod_{i=1}^{n-1} \mathrm{e}^{-\mathrm{j}h_i\,\delta t}\,, \qquad U_{\mathrm{ST}_2}(\delta t) = \left(U_{\mathrm{ST}_1}(\delta t/2)\right)^{\mathrm{T}} U_{\mathrm{ST}_1}(\delta t/2)\,, \tag{17}$$

where $(\cdot)^{\mathrm{T}}$ denotes the matrix transpose. For a product of matrices, this reverses the ordering of the factors, i.e. $(AB)^{\mathrm{T}} = B^{\mathrm{T}}A^{\mathrm{T}}$, such that U_{ST_2} corresponds to applying the same sequence of exponentials in reverse order followed by the forward sequence. The approximation error satisfies $\varepsilon(U(\delta t), U_{\mathrm{ST}_m}(\delta t)) \propto \delta t^{m+1}$ and the Hamiltonian densities h_i are defined as

$$h_i = f_i \quad \forall i \in [2, n-2]\,,\, h_1 = f_1 - \frac{g_x}{2}X_1 - \frac{g_z}{2}Z_1\,,\, h_{n-1} = f_{n-1} - \frac{g_x}{2}X_n - \frac{g_z}{2}Z_n\,, \tag{18}$$

where

$$f_i = -g_{zz}Z_iZ_{i+1} - \frac{g_x}{2}\left(X_i + X_{i+1}\right) - \frac{g_z}{2}\left(Z_i + Z_{i+1}\right)\,. \tag{19}$$

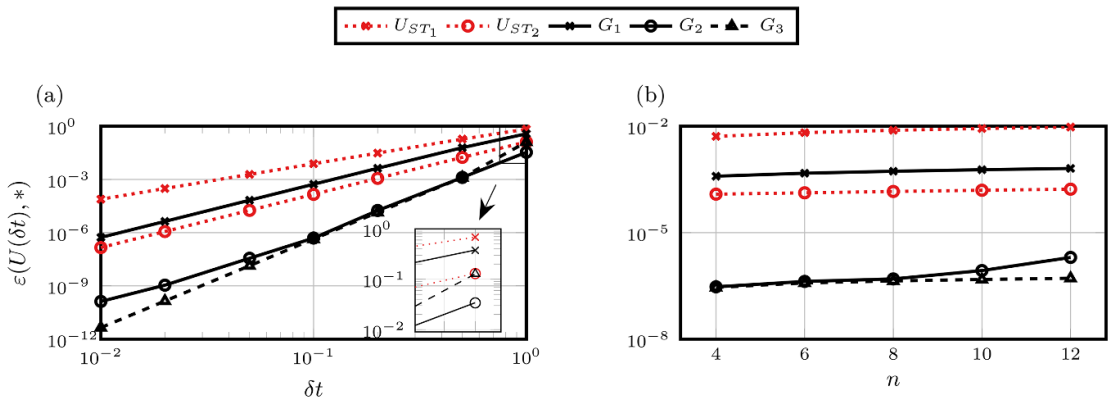


Figure 4. Application of the learning algorithm to multiple quantum many-body scenarios and comparison against commonly used Trotter decompositions in terms of timestep δt with constant system size $n = 8$ (a), system size n with constant timestep $\delta t = 0.1$ (b), and relative error ε. Same symbols denote identical quantum resources.

We aim to compare trainable quantum circuits with ST expansions over a range of time steps δt and different system sizes n. For this reason, we keep the quantum resources close to the Trotter's and pick three different trainable ansätze, viz.,

$$G_1 = \prod_{i=1}^{n-1} G_i, \quad \mathcal{G}_i = (i, i+1)\,, \quad G_2 = \prod_{j=1}^{2}\prod_{i=1}^{n-1} G_i^j, \quad \mathcal{G}_i^j = (i, i+1)\,, \\ G_3 = \prod_{i=1}^{n-2} G_i, \quad \mathcal{G}_i = (i, i+1, i+2)\,. \tag{20}$$

The ansatz G_1 mirrors the gate count and topology of the first-order expansion U_{ST_1}, see section 2.2.2. The ansatz G_2 has the same gate count as U_{ST_2} but different topology. Finally, the ansatz G_3 is a 3-body generalization of the ansatz G_1. All three circuits can be efficiently implemented by a 1D LNN hardware layout. The trainable circuits (G_1, G_3) are always initialized from the first-order expansion $U_{\mathrm{ST}_1}(\delta t)$. The circuit G_2 is similarly initialized as $(U_{\mathrm{ST}_1}(\delta t/2))^2$. Across all time steps and system sizes tested, this initialization strategy was found to perform better than initializing the gates as identity operators. Especially, for the $n = 12$ instance, we found it significantly more efficient to initialize the optimization from the gates obtained for the optimized $n = 10$ instance, rather than from U_{ST_1}. Specifically, the two additional qubits are inserted between the qubits at positions $n/2$ and $n/2+1$ of the previously optimized system. The two, three, or six $(l = 2)$ additional gates introduced by this enlargement are then initialized from the neighboring optimized gates of the smaller system.

We employ the learning strategy described in algorithm 1. We found that $\texttt{maxit} = 10$ steps of individual gate optimization and $\texttt{maxit} \sim 10^3$ global optimization steps provide adequate performance across all cases considered. The learning rate is set to a constant $\eta = 0.2$, and the optimization is terminated upon satisfying the convergence criterion $\|\partial\mathcal{J}/\partial G\| \leqslant \texttt{tol} = 10^{-13}$.

Figure 4(a) shows the dependence of the relative error $\varepsilon(U(\delta t), G_i)$, see equation (4), on the time step δt, and compares it with that of the ST expansions in equation (17) by keeping the system size constant at $n = 8$. The variational circuit G_1 yields an improved error scaling compared to the topologically equivalent first-order Trotter expansion U_{ST_1}. Specifically, $\varepsilon(U(\delta t), G_1)/\varepsilon(U(\delta t), U_{\mathrm{ST}_1}) \propto \delta t$, indicating an additional suppression by one power of δt. For small time steps, the error of G_1 is approximately 150 times smaller than that of U_{ST_1}. For G_2 and G_3, the improvement over U_{ST_2} is roughly three orders of magnitude for small time steps with fitted power-law scalings, $\varepsilon(U(\delta t), G_2) \propto \delta t^{4.26}$, $\varepsilon(U(\delta t), G_3) \propto \delta t^{5.14}$.

The approximate algebraic error scaling observed for the learned circuits suggests the existence of topologically equivalent Trotter formulas with the same order of accuracy. Indeed, the structure of the Ising Hamiltonian (see equation (16)) enables the construction of a second-order Trotter formula topologically equivalent to G_1. This formula is obtained by grouping the operators in the sequence $U_X(\delta t/2)U_Z(\delta t)U_X(\delta t/2)$, where U_X contains the Pauli-X terms and U_Z contains all terms composed of Pauli-Z operators. For G_2 and G_3, however, we did not identify Trotter formulas exhibiting the same algebraic error scaling.

Importantly, these results are not artifacts of fine-tuning on small systems. Figure 4(b) shows that, at fixed time step $\delta t = 0.1$, the error remains stable across all system sizes, $n = 4$–12. The only exception is G_2 at $n = 12$, where the optimization appears to become less effective. In general, our results show a remarkable improvement and are consistent with recent related work [39, 42], indicating that the proposed optimization can compress native quantum dynamics beyond conventional product-formula approximations.

In a direct comparison, the G_3 ansatz yields results approximately one order of magnitude better than those reported in [42]. A corresponding comparison for the G_1 and G_2 ansatz is not possible because of insufficient available data. Reference [42] additionally presents results for $r = r^j = 4, 5$-qubit gates, for which similar improvements are expected. Furthermore, it is found that the quality of the optimized circuit depends sensitively on the choice of initialization. For sufficiently small time steps δt, ST formulas provide a natural and effective initialization. For larger time steps and system sizes, however, this initialization becomes less reliable, and confirms the effect of improved initialization strategies.

This behavior is consistent with the Lieb–Robinson bounds [70] governing information propagation in locally interacting quantum systems. These bounds imply that information spreads at a finite velocity that remains constant in time. As a result, for evolution times $\delta t \sim \mathcal{O}(1)$, the unitary operator $U(\delta t)$ effectively entangles only qubits separated by a distance of order $\mathcal{O}(1)$. Consequently, the optimal gates are not expected to be significantly affected by the addition of new qubits, provided that the system size remains much larger than this distance. This confirms the insights found for the initialization strategy employed for jump-starting the larger systems.

3.1.2. Quantum chemistry

In this section, we consider quantum chemistry models, whose fundamental degrees of freedom are interacting fermionic modes rather than spins. To simulate intrinsically fermionic systems on a qubit-based QC, the fermions need to be mapped to qubits. Fermionic creation and annihilation operators obey anticommutation relations, mapping these Hamiltonians to qubits requires transformations such as Jordan–Wigner or Bravyi–Kitaev [71, 72], which can introduce parity strings and nonlocal Pauli operators. This additional nonlocal structure makes their digital simulation more demanding than the spin models considered in the previous section.

We consider the Pariser–Parr–Pople (PPP) model [73, 74] as a minimal semi-empirical description of the π-electron sector of butadiene [75], $CH_2 = CH - CH = CH_2$. In this approximation, the active space is restricted to the four carbon p_z orbitals, $N_{\mathrm{orb}} = 4$, participating in the conjugated π system. Each spatial orbital supports two spin states, so that the fermionic problem contains $N_{\mathrm{so}} = 2 \cdot N_{\mathrm{orb}} = 8$ spin orbitals.

The PPP Hamiltonian is

$$\mathcal{H}_{\mathrm{PPP}} = -\sum_{i=1}^{N_{\mathrm{orb}}-1} \sum_{\sigma \in \{\uparrow,\downarrow\}} t_i \left(c_{i\sigma}^{\dagger} c_{i+1\sigma} + c_{i+1\sigma}^{\dagger} c_{i\sigma} \right) + U \sum_{i=1}^{N_{\mathrm{orb}}} n_{i\uparrow} n_{i\downarrow} + \sum_{j=2}^{N_{\mathrm{orb}}} \sum_{i=1}^{j-1} V_{ij} \left(n_i - 1\right) \left(n_j - 1\right), \qquad (21)$$

where $i, j \in \{1, \ldots, N_{\mathrm{orb}}\}$ label the carbon sites along the conjugated chain and $\sigma, \tau \in \{\uparrow, \downarrow\}$ label the spin projections. The fermionic operators obey the canonical anticommutation relations $\{c_{i\sigma}, c_{j\tau}^{\dagger}\} = \delta_{ij}\delta_{\sigma\tau}$, with $\{c_{i\sigma}, c_{j\tau}\} = \{c_{i\sigma}^{\dagger}, c_{j\tau}^{\dagger}\} = 0$. We define the spin-resolved number operator $n_{i\sigma} = c_{i\sigma}^{\dagger} c_{i\sigma}$ and the total site occupation $n_i = n_{i\uparrow} + n_{i\downarrow}$.

The first term of equation (21) describes hopping between neighboring p_z orbitals, the second term is the on-site Coulomb repulsion, and the third term contains long-range Coulomb repulsion with a neutralizing background corresponding to one π electron per carbon atom. The Hamiltonian conserves the total fermion number $[\mathcal{H}_{\mathrm{PPP}}, N_f] = 0$, where $N_f = \sum_i^4 n_i$. Butadiene is described by the half-filling sector ($N_f = 4$) of the Hamiltonian.

For the numerical examples below, we adopt the standard PPP parameters for polyenes [76]: a uniform hopping parameter $t_i = 2.4$ eV and on-site interaction $U = 11.26$ eV. The long-range Coulomb matrix elements are chosen using the Ohno interpolation formula [77] as $V_{ij} = 14.397 \left(\left(\frac{14.397}{U} \right)^2 + d_{ij}^2 \right)^{-1/2}$ eV, where $d_{ij} = 1.4|i-j|$ Å—measured in Angstrom (Å) – is assumed for a uniform butadiene chain. We adopt the interleaved spin-orbital ordering

qubit p	1	2	3	4	5	6	7	8
spin orbital	$1\uparrow$	$1\downarrow$	$2\uparrow$	$2\downarrow$	$3\uparrow$	$3\downarrow$	$4\uparrow$	$4\downarrow$

,

to reduce the dimension of each degree of freedom to two and allow a mapping to a spin-qubit model. We then map the fermionic operators associated with the eight spin orbitals to eight qubits using the

Jordan–Wigner transformation [72], viz.,

$$c_{i\sigma} = \left(\prod_{\ell=1}^{p(i,\sigma)-1} Z_\ell\right) \frac{X_{p(i,\sigma)} + \mathrm{j} Y_{p(i,\sigma)}}{2}, \qquad c_{i\sigma}^\dagger = \left(\prod_{\ell=1}^{p(i,\sigma)-1} Z_\ell\right) \frac{X_{p(i,\sigma)} - \mathrm{j} Y_{p(i,\sigma)}}{2}. \tag{22}$$

Thus, the qubit PPP Hamiltonian for butadiene has the form

$$\mathcal{H}_{\mathrm{PPP}}^{\mathrm{qubit}} = E_{\mathrm{shift}} + \sum_{p=1}^{N_{\mathrm{so}}} h_p Z_p + \sum_{q=2}^{N_{\mathrm{so}}} \sum_{p=1}^{q-1} W_{pq} Z_p Z_q - \frac{1}{2} \sum_{p=1}^{N_{\mathrm{so}}-2} t_p \left(X_p Z_{p+1} X_{p+2} + Y_p Z_{p+1} Y_{p+2}\right), \tag{23}$$

where $E_{\mathrm{shift}} = U$ is a constant energy offset, $t_p = t_i = 2.4$ eV, $h_q = -U/4$ and

$$W_{pq} = \begin{cases} \dfrac{U}{4}, & p = (i,\uparrow),\ q = (i,\downarrow), \\ \dfrac{V_{ij}}{4}, & p = (i,\sigma),\ q = (j,\tau),\ i \neq j. \end{cases} \tag{24}$$

As a consequence of the interleaved ordering, the mapped Hamiltonian is at most three-local in the Pauli representation: The Coulomb sector is diagonal and two-local, while the hopping sector is three-local. This locality structure is important for digital simulation schemes in which native gates may act on non-neighboring qubits but are restricted to few-body interactions.

Due to the complexity of the PPP Hamiltonian, we perform only a first-order ST expansion. The decomposition is organized to reduce the number of elementary evolution blocks by combining terms that can be implemented on the same few-qubit support. In particular, the three-body hopping contributions are grouped with local and interaction terms that act on the same contiguous support, while the remaining long-range density-density interactions are implemented as separate two-body gates. The resulting first-order step is written as $U_{\mathrm{ST}_1}(\delta t) = U^3_{\mathrm{ST}_1}(\delta t) U^2_{\mathrm{ST}_1}(\delta t)$ and consists of 21 gates in total. The contribution $U^3_{\mathrm{ST}_1}$ contains six three-body gates associated with the hopping terms, and $U^2_{\mathrm{ST}_1}$ contains 15 two-body gates corresponding to the remaining density-density interactions, with the gate layout specified below.

We solve the problem described above, keeping the ST resources constant and the system aligned to $n = N_{\mathrm{so}} = 8$. For the learning, we employ $\texttt{maxit} = 10$ steps of individual gate optimization and $\texttt{maxit} \sim 10^3$ global optimization steps. The learning rate has been set to $\eta = 10^{-5}$ for short-range and $\eta = 10^{-3}$ for long-range interaction and the gradient criterion is set as $\|\partial \mathcal{J}/\partial G\| \leqslant \texttt{tol} = 10^{-15}$. This short- and long-range interacting ansatz, illustrated for $n = 8$ in figure 5(a), is given by

$$G = G_2 G_1, \quad \text{where}$$
$$G_1 = \prod_{j=1}^{1} \prod_{i=1}^{n-2} G_i^j, \quad \mathcal{G}_i^j = (i, i+1, i+2), \quad G_2 = \prod_{j=1}^{n-3} \prod_{i=1}^{n-2-j} G_i^j, \quad \mathcal{G}_i^j = (j, i+j+2). \tag{25}$$

Figure 5(b) compares the error of the optimized variational circuit, $\varepsilon(U(\delta t), G)$, with that of the first-order ST approximation, $\varepsilon(U(\delta t), U_{\mathrm{ST}_1}(\delta t))$, for several time steps δt. We initialize the learning procedure from the first-order ST expansion. We observe a consistent reduction in the error of approximately $\varepsilon_{\mathrm{ST}}/\varepsilon_{\mathrm{G}} - 1 \simeq 20\%$ across the full range of time steps considered.

This improvement is appreciable, although substantially smaller than the orders-of-magnitude reduction obtained for the Ising Hamiltonian in the previous section. We attribute this difference to the lower sparsity of the Trotterized gate generators in the PPP model, which restricts the ability of the learned circuit to improve significantly over the first-order ST initialization. We also tested extensions of the ansatz, including an additional three-body gate layer and extra two-body gates, but did not observe a significant improvement over the results reported here.

3.2. Engineering applications

In this section, we explore the application of the operator learning protocol to engineering problems. The first case concerns the accurate programming of the finite-difference operator associated with the centered second-order approximation of second derivatives, see section 3.2.1. This application is broadly relevant across computational science and engineering, as it underlies the solution of partial differential equations on quantum hardware.

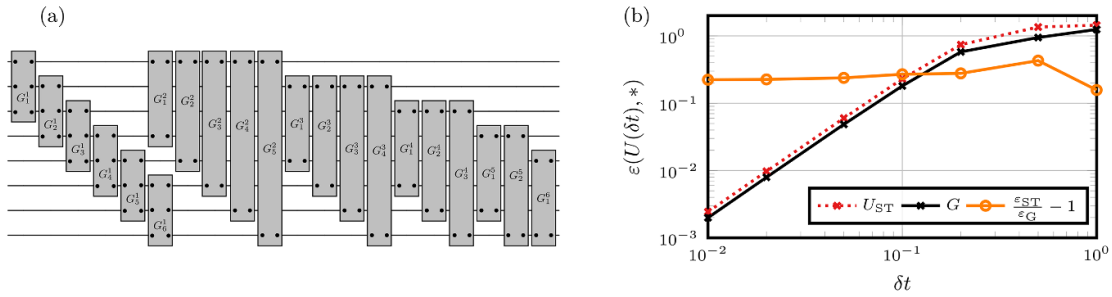


Figure 5. (a) Quantum circuit implementation of the butadiene PPP Hamiltonian, consisting of 21 nearest-neighbor and long-range interaction gates. (b) Time-step-induced errors as a function of the time-step size for the ST and learned-circuit propagators.

Secondly, in section 3.2.2, we focus on learning a dense operator that arises in CFD during the analysis of inviscid flow around a 2D airfoil, modeled using the Hess-Smith panel method [78]. Both applications employ amplitude encoding, whereby the value of the classical field at each grid point is encoded in the amplitude of the quantum state associated with the corresponding grid-point basis state. The operators are encoded using the *big-endian* notation. In this setting, local features of the classical field may result in long-range quantum correlations. Consequently, reducing local errors in the classical field often involves fine-tuning multiple local gates, making such errors challenging to correct. This provides a useful benchmark for the ℓ_2-regularization term in equation (6), controlled by ρ, which is actively used in both sections. Furthermore, the success probability stabilizer μ is chosen independently for each experiment.

Unlike the quantum mechanics applications discussed in section 3.1, the matrices encountered here are generally non-unitary and thus necessitate block encoding. Throughout the experiments, we assess the framework's complexity and accuracy by analyzing relative errors, success probabilities, and computational effort.

3.2.1. Finite approximation of second derivative

In the following, we consider the prototypical problem of implementing the discretized second spatial derivative, represented by the 1D Laplace operator. Without loss of generality, we assume a spatial equidistant 1D finite difference discretization, i.e. $x_{i+1} - x_i = \Delta x$, with 2^{n_s} supporting points i. For the approximation of the second spatial derivative of some generic scalar field $f(x)$, we employ a second-order accurate central difference scheme, viz.,

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f_{i-1} - 2f_i + f_{i+1}}{\Delta x^2}. \tag{26}$$

Furthermore, we assume homogeneous Dirichlet boundary conditions for the scalar field $f(x)$, which compiles the domain with non-cycling but closed boundary conditions. Rewriting equation (26) in matrix-vector form and including the boundary conditions, results in the matrix $A \in \mathbb{R}^{2^{n_s} \times 2^{n_s}}$ defined as

$$A = \begin{bmatrix} -2 & 1 & 0 & \dots & 0 \\ 1 & -2 & 1 & \ddots & \\ & & \ddots & & \\ & \ddots & 1 & -2 & 1 \\ 0 & \dots & 0 & 1 & -2 \end{bmatrix}. \tag{27}$$

Exploiting the tridiagonal Toeplitz structure, the operator A is efficiently represented by TTs with a resource scaling of $\mathcal{O}(n_s)$. In particular, it can be represented as a TT of n_s local tensors with auxiliary dimension $R(A) = 5$ for any grid size [79]. This finding [79] is applied for the initialization of the tensor quantum programming routine in [42]. In contrast, our work goes beyond this low-bond decomposition by leveraging correlations over all scales. Accordingly, it should be understood as an alternative learning approach (see section 1). We note that the operator in equation (27) cannot be written as a product or a sum of operators with local (range-$\mathcal{O}(1)$) operator densities. As a result, the exact encoding of the

operator requires multi-body gates, and finding an accurate approximation by limiting the gate size is therefore challenging.

The learned operator is assessed by the relative error and the success probability. Additionally, in this application, the system size and the qubit connectivity are varied. The base gate size r^j is estimated by assuming an auxiliary dimension of $R = 5$ [79]. Therefore, we employ four-qubit gates with a gate size $r = r^j = \lceil \log_2(R) \rceil + 1 = 4$, as in unitary tensor-network decompositions [42]. The number of gates per layer is set to $m^j = m = n - 3$, which yields overlapping gates. The number of layers has been chosen heuristically as $l = 4$, since additional layers did not reduce the cost function further. As the operator A is not unitary, we use $n_a = 1$ ancilla qubit for block encoding throughout. Furthermore, we employ a constant ansatz emulating an LNN-fashioned topology, given by the gate layout

$$G_{\mathrm{LNN}} = \prod_{j=1}^{4} \prod_{i=1}^{n-3} G_i^j, \quad \mathcal{G}_i^j = (i, i+1, i+2, i+3)\,. \tag{28}$$

We also employ a variant of 2DL topology for which the ansatz is inspired by a block encoding using quantum signal processing. Every gate in quantum signal processing (local rotation and signals) is applied to the ancilla qubit, a feature we incorporated in our ansatz,

$$G_{\mathrm{2DL}} = \prod_{j=1}^{4} \prod_{i=1}^{n-3} G_i^j, \quad \mathcal{G}_i^j = (1, i+1, i+2, i+3)\,. \tag{29}$$

This ansatz particularly addresses a 2D lattice chip topology, with qubit 1 located at the center and the remaining qubits arranged peripherally. For comparison, both ansätze use only nearest-neighbor four-qubit gates while enabling correlations to propagate over larger length scales across the 2D lattice. If the system size grows beyond ten qubits, the realization of G_{2DL} requires long-range programmable hardware, since all gates involve the same qubit in position 1. In the following, we assess their accuracy and probabilistic performance, and subsequently demonstrate the hierarchical training strategy together with the impact of regularization using the LNN ansatz.

a. Success probability and accuracy are first evaluated to characterize their scaling behavior with increasing system size n_s. The maximum number of iterations is set to $\texttt{maxit} \sim 10^3$, with the final training stage extended to $\texttt{maxit} \sim 10^4$ iterations. Throughout all experiments, the learning rate is kept constant at $\eta = 0.02$ and regularization is disabled ($\rho = 0$). The weight of the stabilizing term μ is adapted to the attainable accuracy, i.e. $\mu = 10^{-18}$ ($n_s = 4$), 10^{-15} ($n_s = 5$), and 10^{-5} ($n_s = 6$–9), such that it prevents $c \gg c_{\mathrm{ideal}}$ without limiting the accuracy, see equation (9) and appendix B. In addition to the iteration limit, convergence is determined by the gradient criterion $\|\partial \mathcal{J}/\partial G\| \leqslant \texttt{tol} = 10^{-9}$. To assess the sensitivity of the learning protocol to the underlying hardware connectivity—1D versus 2D chip topologies—the experiments compare the two ansätze (28) and (29). The corresponding results are presented in figure 6, where Subfigures (c) and (d) illustrate representative quantum circuit implementations for $n = 6$ qubits. We observe a highly accurate representation of the operator for both ansätze for system sizes $n_s < 6$ qubits, where crosses and circles denote the LNN and 2DL results, respectively. For both, the accuracy reaches machine precision, while the probability of success approximately converges to the operator's intrinsic probability of $p(A/\|A\|_2) \approx 0.3734$, see equation (2). As the system size increases, performance limitations emerge for both ansätze. The error increases to values on the order of 10^{-2}, yet remains approximately constant with further growth of the system size. Likewise, the success probability p exhibits only moderate variations and remains within the same order of magnitude. Overall, the results demonstrate improvements compared to literature [42], particularly visible for smaller system sizes. The enhancements are reflected not only in the reconstruction error (see equation (3)) but also, and more notably, in the achieved success probabilities.

Training the discretized second derivative operator A requires the representation of long-range interactions. For smaller system sizes, these interactions can be captured more accurately using a fixed arrangement of 4-qubit gates. As the system size increases, the gates span progressively smaller portions of the quantum register, thereby reducing the effective expressiveness of the ansatz. Together with potential BP issues during optimization, arising from the absence of an accurate initialization protocol, this provides a plausible explanation for the observed behavior. Comparing the two architectures, the 2DL ansatz outperforms the LNN approach. The differences can be attributed to the ansatz's ability to realize long-range interactions more effectively by coupling each gate to the first register qubit, i.e. the ancilla.

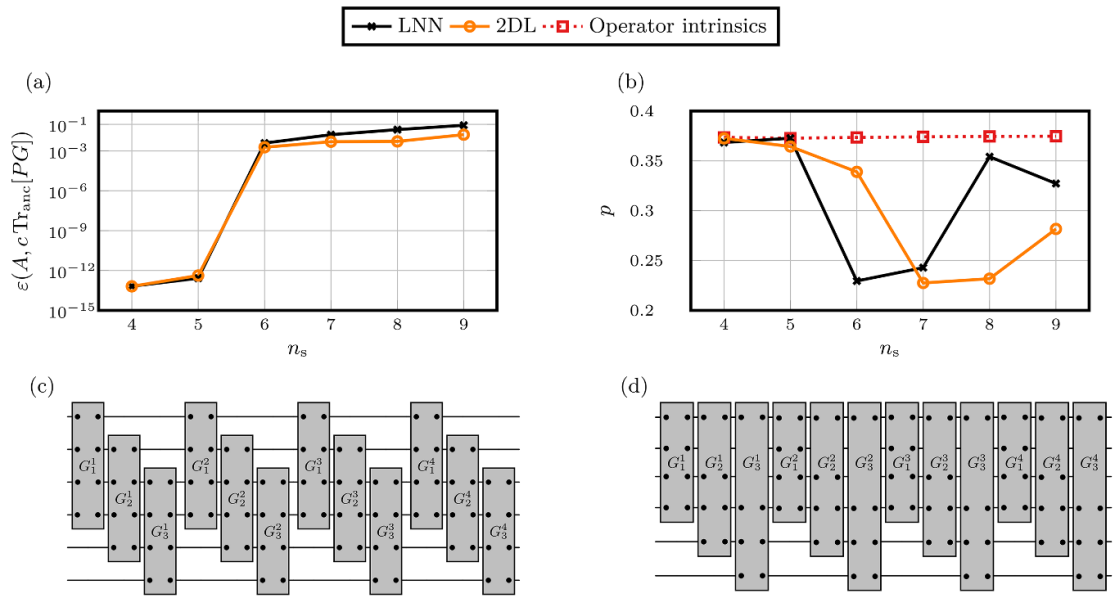


Figure 6. Operator learning of the discrete second derivative operator for varying system sizes n_s. The relative error $\varepsilon(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])$ and success probability p are shown in (a) and (b), respectively. The LNN and 2DL learnable circuit architectures used for $n = 6$ qubits are shown in (c) and (d).

b. Training strategy and regularization were introduced in section 2.2.1 and are now evaluated quantitatively. In the context of amplitude encoding, the ordering and structure of the gate sequence during optimization are expected to strongly influence the learned representation. Since the $n_s = 5$ qubit case already exhibits excellent agreement, with errors approaching machine precision (see figure 6), the following analysis focuses on the more challenging $n_s = 6$ qubit case.

To assess the influence of the optimization strategy, the matrix A for a system size of 2^6 is trained under three different configurations: A) without hierarchical optimization, B) with hierarchical optimization, and C) with hierarchical optimization and additional ℓ_2-regularization.

In experiment A, the hierarchical optimization is disabled, such that only a global optimization (see algorithm 1—line 8) is performed. In the other two experiments, B and C, the hierarchical optimization stages are performed analogously, each with $\texttt{maxit} \sim 10^3$ iterations. For experiment C, a regularization term with weight $\rho = 4 \times 10^{-8}$ is included, yielding an additional contribution to the objective function. For all configurations, the final optimization stage is executed with $\texttt{maxit} \sim 10^4$ iterations, and the convergence criterion on the derivative's norm is set to $\|\partial\mathcal{J}/\partial G\| \leqslant \texttt{tol} = 10^{-7}$. All experiments employ the same numerical stability parameter for the success probabilities, viz. $\mu = 10^{-6}$, as well as an identical initial learning rate of $\eta = 0.2$, which is adaptively modified within the L-BFGS optimization procedure. Compared with the system-size study above (see figure 6), the stabilizing penalty is relaxed from $\mu = 10^{-5}$ to $\mu = 10^{-6}$ to allow for a further gain in accuracy. The corresponding results are summarized in figure 7.

The upper row of figure 7 shows the target matrix A and the learned matrix G resulting from experiment C (see figure 7(b)), where the entries $i, j = 1 \ldots 64$ correspond to the encoded target operator A. The evolution of the objective function $\mathcal{J}$ for all three experiments is shown in figure 7(c), where experiment A terminates after approximately 2×10^3 iterations, whereas experiments B and C terminate after approximately 2×10^4 iterations.

Experiment A, which employs only global optimization, achieves a reduction of less than two orders of magnitude in the objective function. The relative error converges to $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{A}} \approx 0.2790$, while the resulting success probability $p_A \approx 0.4146$ remains only slightly above the operator's intrinsic probability of $p(A/\|A\|_2) \approx 0.3734$, see equation (2). The limited quality of the approximation is further illustrated in figure 7(d), where the absolute residual satisfies $|\mathcal{E}(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{A}}| \sim 10^{-1}$, see equation (3). In contrast, the advantages of the hierarchical optimization strategy become clearly visible in experiments B and C, as shown in figure 7(c). The discontinuous jumps in the objective function correspond to

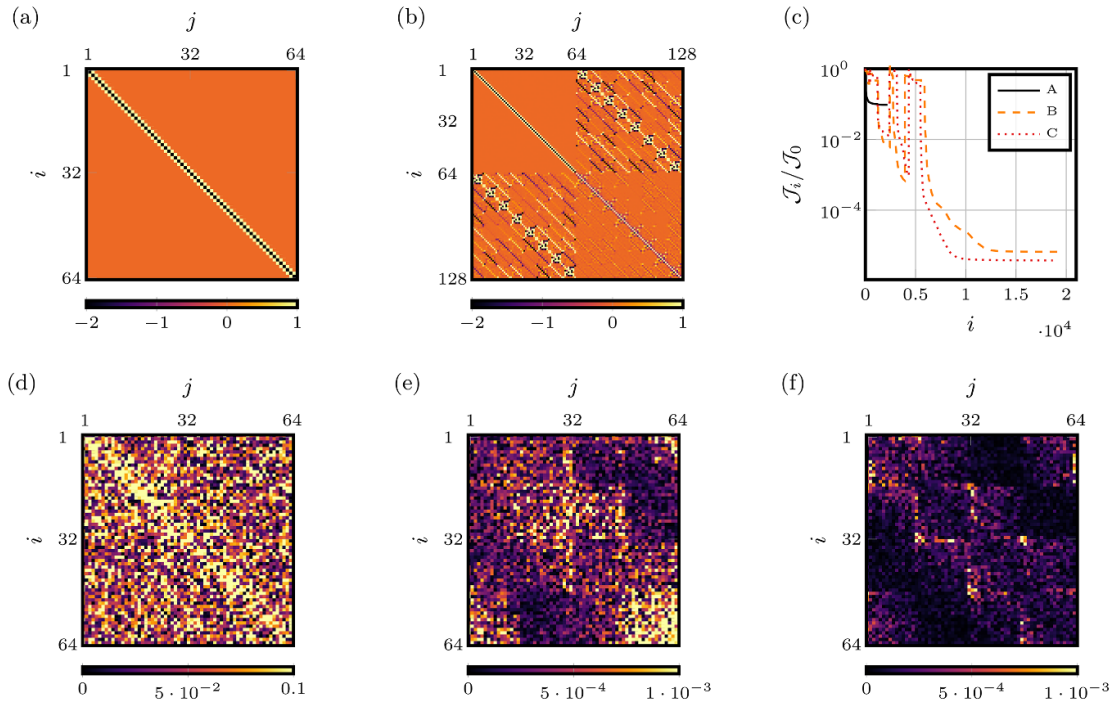


Figure 7. Application of the learning protocol to the discretized second derivative operator. The ground-truth operator A is shown in (a), and the reconstructed operator obtained from the learned block encoding G of experiment C is shown in (b). The reconstruction recovers the dominant diagonal structure of the target block $\mathrm{Tr}_{\mathrm{anc}}[PG]$ in the upper-left corner, while also inducing population in the remaining subblocks. Panel (c) shows the cost-function evolution for the different experiments. Panels (d)–(f) show the corresponding absolute residuals $|\mathcal{E}(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{A}}|$, $|\mathcal{E}(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{B}}|$, and $|\mathcal{E}(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{C}}|$, respectively. The results demonstrate that hierarchical learning with regularization suppresses spurious noise and enhances the stability of the recovered operator.

increases in the hierarchy level following the termination of algorithm 1—line 6. Both experiments converge to a stationary point, and the hierarchical optimization strategy allows a reduction of the objective function by approximately five orders of magnitude, corresponding to an improvement of approximately four orders of magnitude relative to experiment A. The converged relative errors for experiments B and C are $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{B}} \approx 1.72 \times 10^{-3}$, and $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{C}} \approx 7.17 \times 10^{-4}$, respectively. Their associated absolute residual is shown in figures 7(e) and (f). Experiment C not only achieves lower overall error levels but also demonstrates substantially suppressed outliers. This gain in accuracy is achieved without a loss in success probability. Experiment C achieves a success probability of $p_{\mathrm{C}} \approx 0.3628$, compared with $p_{\mathrm{B}} \approx 0.3597$ in experiment B. Hence, with the stabilizing term active, including the regularization term does not compromise the success probability.

3.2.2. Potential flow analysis

The final application deals with the steady flow around a 2D NACA 0012 airfoil. The airfoil is subject to a uniform free-stream velocity of U_∞ and set at an angle of attack (AoA) $\alpha = 5°$ as depicted in figure 8. The underlying physical model is based on an inviscid potential flow, and the numerical algorithm employs a Hess-Smith panel method [78].

The physical model assumes irrotational and solenoidal velocity fields and is typically associated with the limiting case of infinitely large Reynolds numbers, i.e. $Re = U_\infty \bar{c}/\nu \to \infty$, with $\bar{c}$ being the chord length of the airfoil, and ν the fluid's kinematic viscosity. Under these assumptions, the velocity vector can be expressed as $\vec{u} = \nabla\phi$, where ϕ is a scalar potential field satisfying the Laplace equation $\Delta\phi = 0$ as a direct consequence of the conservation of mass. The linearity of the Laplace equation permits the construction of flow configurations through the superposition of elementary solutions to $\Delta\phi = 0$, such as uniform flow as well as sources and free vortices of unknown strength [78]. The panel method represents the flow around arbitrary geometries as superpositions of such discrete elementary solutions. In this study, the potential ϕ is decomposed into contributions from the approaching uniform flow U_∞, sources, and a single vortex. The unknowns to be determined are the strengths of the sources and the vortex. These are determined from the kinematic flow condition [78] and the Kutta [80] condition, the former ensures that no flow passes through the body, while the latter guarantees that the flow leaves the

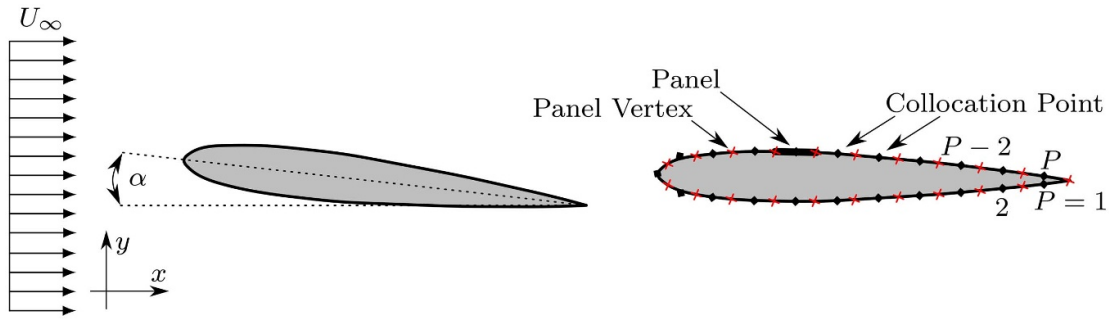


Figure 8. Illustration of the flow around a NACA 0012 airfoil subjected to an approach flow U_∞ at AoA α (left) and geometry discretization by P panels with collocation points in the center of each panel (right).

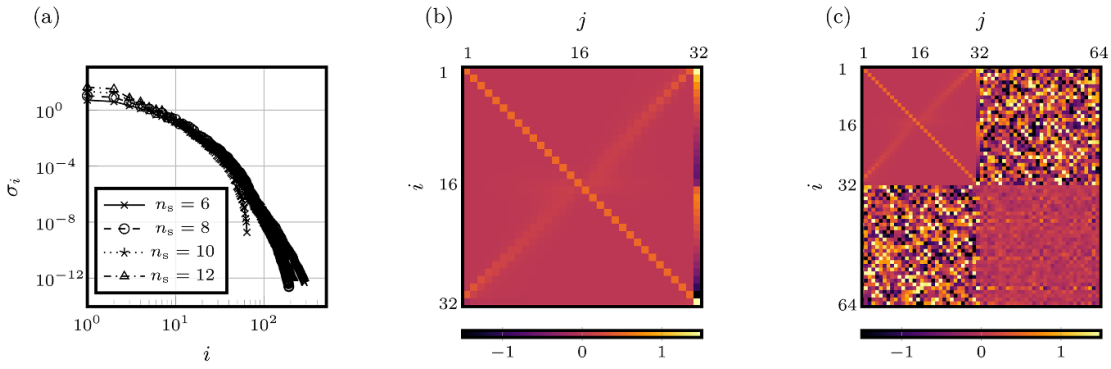


Figure 9. Sparsity and correlation measures of the potential flow operator $A \in \mathbb{R}^{P+1\times P+1}$. Subfigure (a) depicts the evolution of singular values σ_i of the central partition matrix $A^\sharp$ for multiple discretizations $P = 2^{n_s} - 1$. Subfigures (b), (c) show the target matrix A for a system size $P+1 = 32$ as well as the learned operator G in its block-encoded version (regularization disabled), with the approximation to A located in the upper-left corner.

trailing edge tangentially. For this purpose, the profile geometry is discretized into P panels, with a collocation point defined at the center of each panel where the kinematic flow condition must be satisfied, see figure 8. To satisfy the Kutta condition, the magnitudes of the tangential velocities at the upper and lower trailing-edge panels must match. This leads to a system of linear equations for $P+1$ unknowns (P sources q_j plus one vortex γ) for a geometry discretized by P panels, viz.

$$\sum_{j=1}^{P} A_{ij} q_j + A_{i(P+1)}\gamma = b_i \;:\; \forall i \in [1, P+1]\,. \tag{30}$$

Since the elementary solutions essentially correspond to global ansatz functions, the matrix A is fully populated—in contrast to matrices resulting from typical local ansatz functions.

To improve our understanding of the complexity of the matrix A, we study the singular value spectrum of a suitable partition of the matrix shown. Given $A \in \mathbb{C}^{2^{n_s}\times 2^{n_s}}$ where $2^{n_s} = P+1$, we bipartition the n_s-qubit Hilbert space into two subsystems of dimension $d = 2^{n_s/2}$. The matrix is first reshaped into a rank-four tensor $\mathcal{A}_{i_L,i_R,j_L,j_R} \in \mathbb{C}^{d\times d\times d\times d}$, where i_L, i_R label the row indices and j_L, j_R label the column indices of the two subsystems. We then realign the tensor by fusing the index pairs (i_L, j_L) and (i_R, j_R), yielding the matrix $A^\sharp_{(i_L,j_L),(i_R,j_R)} = \mathcal{A}_{i_L,i_R,j_L,j_R}$. The singular-value spectrum σ_i studied here is the spectrum of $A^\sharp$, visualized in figure 9(a). The method employed corresponds to the Schmidt decomposition or the realignment of A with respect to the bipartition $L|R$. For all investigated discretizations $P = 2^{n_s} - 1$ with $n_s \in \{6, 8, 10, 12\}$, a significant number of sufficiently large singular values appear. This highlights the complex structure of A and makes its exact encoding into a quantum circuit a challenge.

Figure 9(c) displays the learned matrix G for a discretization with $P = 31$ panels, where 16 panels are located on the upper side (suction side) and 15 panels on the lower side (pressure side) of the airfoil. The dense structure of the target matrix A is shown in figure 9(b) and requires $n_s = 5$ and $n_a = 1$ qubits for encoding.

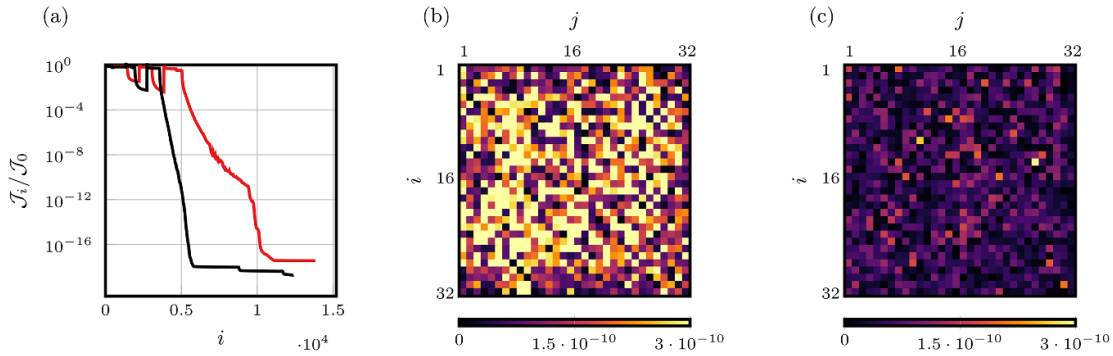


Figure 10. Results for the learning of an inviscid airfoil flow operator showing the evolution of the cost functional (a) – without $\rho = 0$ (red) and with $\rho = 10^{-6}$ (black) regularizer—and the absolute residual $|\mathcal{E}(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])|$ to the target matrix A without (b) and with regularization (c).

The optimization is performed using the L-BFGS optimizer setting stepsize $\eta = 0.02$, maximum number of iterations $\texttt{maxit} \sim 10^3$ for each of the hierarchical learning sequences in algorithm 1 and $\texttt{maxit} \sim 4 \times 10^3$ for the final training executed in algorithm 1—line 8. Furthermore, the convergence criterion is defined on the gate-wise derivative of the cost function, see equation (8), as $\|\partial\mathcal{J}/\partial G\| \leqslant \texttt{tol} = 10^{-15}$. Two experiments were conducted for demonstration purposes: one with regularization disabled ($\rho = 0$, experiment A) and one with regularization enabled ($\rho = 10^{-6}$, experiment B). In both cases, the stabilizer for the success probability was set to $\mu = 0.0$.

The quantum circuit is constructed out of $l = 4$ layers, where gate primitives are set to nearest-neighbor $r = r^j = 4$-qubit gates. This corresponds to the gate sequence for LNN-like chip topologies, which has been used in the previous experiment and is visualized in figure 6(c). Additional hardware topology constraints along the learning are not imposed, such that

$$G = \prod_{j=1}^{4} \prod_{i=1}^{n-3} G_i^j, \quad \mathcal{G}_i^j = (i, i+1, i+2, i+3)\,. \tag{31}$$

The learned operator G in its block-encoded version is shown in figure 9(c), in which the entries $i, j = 0 \ldots 31$ correspond to an approximation to the target matrix A with regularization disabled ($\rho = 0$). The convergence of the normalized cost function and the relative errors obtained with and without regularization are displayed in figure 10. The convergence history of the objective functional displays small fluctuations, which are due to the Wolfe-Conditions employed within the L-BFGS optimizer, and a greater descent in experiment B. Regardless of whether regularization is used, the cost function $\mathcal{J}$ decreases by more than 15 orders of magnitude. The use of amplitude encoding implies long-range qubit interactions, as the correlations of the target operator are not aligned with those of the qubit register. Despite this mismatch, the presented method achieves high accuracy in the learned operator with the relative error of $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{A}} \approx 1.78 \times 10^{-9}$ (regularization disabled) and $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])_{\mathrm{B}} \approx 4.10 \times 10^{-10}$ (regularization with $\rho = 10^{-6}$), see figures 10(b) and (c). These figures show that the inclusion of the regularization term provides an improvement, w.r.t. the maximum relative error. Furthermore, a substantial reduction of the outliers, which reduces the relative error $\varepsilon(A, c\mathrm{Tr}_{\mathrm{anc}}[PG])$, is observed in the results. For practical calculations, the averaged success probabilities outlined in equation (2) are relevant. These also improve as a result of regularization. Using a single ancilla qubit ($n_a = 1$), they stand at $p \approx 0.02766$ without regularization and $p \approx 0.04824$ with regularization. Generally, the experiments confirm the learning of dense and strongly correlated operators, and highlight potential to encode other structured dense operators arising in classical scientific computing.

4. Conclusion

In this work, we introduced a hardware-agnostic variational operator-learning framework for compiling discrete operators into compact quantum circuits. The method is designed to treat both unitary and non-unitary operators, and to incorporate hardware constraints directly at the circuit-construction level, including restricted qubit connectivity, long-range gates, and multi-qubit interaction patterns. To this end, the algorithm maps a target operator to a trained circuit, minimizing the normalized Frobenius

error between the two operators during training. The protocol integrates a backpropagation-based optimization strategy combined with manifold retraction capable of learning unitary operators directly, while non-unitary operators are learned through a cost-efficient block encoding. Unlike methods whose ancilla number scales with auxiliary dimension or LCU terms, the proposed block encoding uses only one global ancilla for non-unitary operators. Finally, the hierarchically executed learning process can incorporate a regularizer for smoothing residual errors in order to suppress localized outliers and improve the learned amplitude-encoded operators.

We quantitatively assess the generality of our programming protocol across a range of application domains, including quantum physics (many-body dynamics), quantum chemistry (butadiene molecule), and engineering (finite-difference approximations, potential flow analysis). These examples demonstrate that (a) the method yields compact circuit representations with higher accuracy compared to published reference techniques, (b) operator learning can serve as a practical programming layer between classical discretized models and quantum hardware, and (c) the protocol offers powerful capabilities for circuit compression, thereby effectively adapting resource consumption in quantum computations to specified constraints.

In conjunction with quantum applications, the proposed framework is evaluated using unitary time-evolution operators, where it outperforms standard Suzuki-Trotter decompositions and exhibits lower approximation errors in all experiments. The largest performance gains are observed for Hamiltonians exhibiting pronounced structure and locality, where the ansatz can exploit correlations that are not efficiently captured by fixed-order Trotter schemes. At the same time, the results indicate that circuit quality depends heavily on the initialization strategy, highlighting initialization as a crucial component.

For all engineering applications examined here, accuracy close to machine precision is achieved up to the 5-qubit instance, while requiring fewer ancilla qubits than TN-based constructions. The framework successfully learned both sparse, structured and non-sparse (dense), unstructured operators. This suggests that the approach is not restricted to sparse Hamiltonian operators and motivates its application to more general operators, for instance, those associated with the unstructured computational grids encountered in classical scientific computing. A principal limitation observed, particularly for the approximation of the Laplacian, is the lack of a scalable initialization strategy. Consequently, this remains an important aspect for future work.

The algorithm could be improved and extended in various ways. First, scalability beyond the current implementation in the full Hilbert space requires an extension based on TNs, see, e.g. [39]. Such a formulation would reduce the classical memory and contraction cost, at the price of restricting the method to operators that admit efficient TN compression. Second, the circuit ansatz could be made adaptive by adding gates according to gradient-based criteria, in the spirit of ADAPT-VQAs [81]. Related adaptive strategies have recently shown strong performance in quantum state learning [82], and could provide a systematic way to allocate circuit resources only where they are most effective. Third, versatility can be further enhanced by investigating retraction alternatives such as QR-decomposition-based approaches (Householder/Givens projections or Newton–Schulz iteration), and introducing an adaptive learning rate as well as derivative-dependent regularization terms. Finally, while qubit connectivity and device layout already enter the ansatz through the gate supports $\mathcal{G}_i^j$, the native gate set and the gate sizes r^j are currently fixed prior to the optimization. Including these choices directly into the optimization problem, e.g. by adjusting connectivity lists induced by a given qubit assignment, would extend the presented algorithm to incorporate non-trivial logical-to-physical qubit maps, which constitutes an important research direction. Beyond that, noise-aware optimization, in which the dissipative channels of the respective hardware noise model are incorporated directly into the optimization procedure, is a promising direction for future work.

In conclusion, the four test cases presented here demonstrate the applicability of our method to non-trivial operators arising in quantum computing and computational engineering. Extending it to a broader class of operator programming problems is a promising direction for future research.

Acknowledgments

This publication and the current work have received funding from the European Union's Horizon Europe research and innovation program (HORIZON-CL4-2021-DIGITAL-EMERGING-02-10) under Grant Agreement No. 101080085 QCFD.

Data availability statement

The data that support the findings of this study are openly available at the following URL: https://doi.org/10.25592/uhhfdm.18962 [83].

Appendix A. Spectral-to-Frobenius ratio of the residual

The relative error $\varepsilon(A,B)$, defined in equation (4) and used throughout this work to assess accuracy, is built on the Frobenius norm. This norm measures the average, and not the worst-case, deviation between the target A and the learned operator B. For example, a residual can have a small Frobenius norm while still acting almost maximally wrongly on one specific input state. For a quantitative demonstration, we take the Frobenius norm of the residual in equation (3), where the residual is $\mathcal{E}(A,B) = A - B$ with $B = c\,\mathrm{Tr}_{\mathrm{anc}}[PG]$. To this end, assume $A = \mathbb{I}$, $B = \mathbb{I} - |0\rangle\langle 0|$ of size $2^n \times 2^n$, for which the error is $\varepsilon(A,B) = 1/\sqrt{2^n}$ and obviously decreasing with rising system size. This issue gets set into relation by the norm inequality [84], viz.

$$\frac{\|\mathcal{E}(A,B)\|_{\mathrm{fro}}}{\sqrt{N}} \leqslant \|\mathcal{E}(A,B)\|_2 \leqslant \|\mathcal{E}(A,B)\|_{\mathrm{fro}}\,. \tag{A1}$$

Herein, the worst-case error is governed by the spectral norm

$$\|\mathcal{E}(A,B)\|_2 = \max_{\||\psi\rangle\|=1} \left\|\mathcal{E}(A,B)\,|\psi\rangle\right\|, \tag{A2}$$

where $\|\cdot\|$ denotes the Euclidean norm of a vector. Equation (A2) makes the dependence on the input state explicit, as the spectral norm is the error for the least favorable input state. The lower bound in equation (A1) is Frobenius-norm based and corresponds to the average over input states. Averaging over Haar-random input states gives [85]

$$\mathbb{E}_{\psi}\left[\left\|\mathcal{E}(A,B)\,|\psi\rangle\right\|^2\right] = \frac{\mathrm{Tr}\left(\mathcal{E}^{\dagger}(A,B)\,\mathcal{E}(A,B)\right)}{N} = \frac{\|\mathcal{E}(A,B)\|_{\mathrm{fro}}^2}{N}\,. \tag{A3}$$

Thus, $\|\mathcal{E}(A,B)\|_{\mathrm{fro}}/\sqrt{N}$ is the root-mean-square error over input states. Equation (A1) is therefore the statement that the worst case is never below, and at most $\sqrt{N}$ times above, the average measure.

As a consequence of the discussion above, we employ the ratio of the two measures to relate the average error to the worst-case scenario, viz.

$$\zeta(A,B) = \frac{\|\mathcal{E}(A,B)\|_2}{\|\mathcal{E}(A,B)\|_{\mathrm{fro}}/\sqrt{2^{n_s}}} = \frac{\sqrt{2^{n_s}}\,\|\mathcal{E}(A,B)\|_2}{\|\mathcal{E}(A,B)\|_{\mathrm{fro}}}\,. \tag{A4}$$

This ratio ζ measures how far the worst-case exceeds the average, and hence how uniformly the residual is spread over input states. For the sake of completeness, $\zeta = 1$ is reached when the singular values of $\mathcal{E}(A,B)$ are uniform, whereas $\zeta = \sqrt{N}$ when $\mathcal{E}(A,B)$ is rank one, which recovers the exemplary given situation above.

For clarity, we therefore report ζ for the optimal (converged) result of each application presented in section 3, within the next sections.

A.1. Quantum applications

Figure 11(a) shows $\zeta(U(\delta t), *)$ for the three trainable ansätze G_1, G_2, and G_3 as a function of the number of system qubits n, with the time step fixed at $\delta t = 0.1$. These are the same simulations shown in figure 4(b). Over the system sizes tested, ζ increases only weakly and appears to scale subexponentially with n, in contrast to the exponential scaling of its upper bound. This suggests that error concentration does not become more pronounced over the tested range.

Figure 11(b) shows $\zeta(U(\delta t), G)$ for the butadiene PPP propagator as a function of the time step δt, with $n = 8$ fixed. These are the same simulations shown in figure 5(b). Here, ζ remains approximately constant as δt varies, indicating no increase in error concentration over the time steps tested.

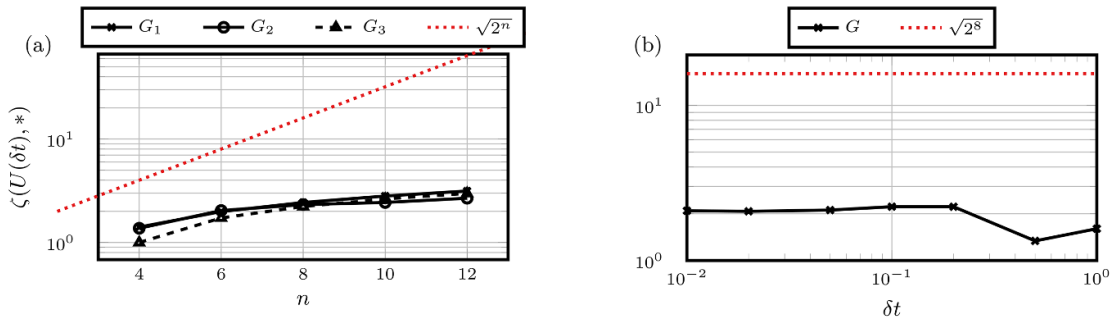
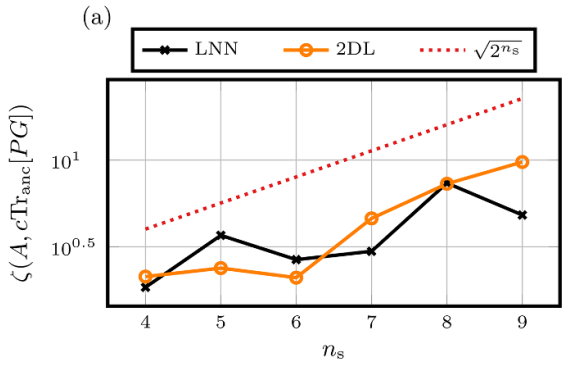


Figure 11. Spectral-to-Frobenius ratio $\zeta(U(\delta t), *)$ for the quantum applications, see equation (A4). Quantum Simulation results are shown in Subfigure (a), as a function of system size n at fixed $\delta t = 0.1$, for ansätze G_1, G_2, G_3, whereas results for the Quantum Chemistry application, given in subfigure (b), maintain a constant system size $n = 8$ demonstrating the error for a changing δt.



(b)

	$\rho = 0$	$\rho = 10^{-6}$
$\varepsilon \approx$	1.78×10^{-9}	4.10×10^{-10}
$\zeta \approx$	2.19	2.08

Figure 12. Spectral-to-Frobenius ratio $\zeta(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])$ for the engineering applications, see equation (A4). Simulation results for the discretized second-derivative operator are shown in Subfigure (a), as a function of system size n_s, for LNN and 2DL ansätze, whereas results for the potential flow analysis application with and without regularization are given in table (b).

A.2. Engineering applications

Figure 12(a) shows the spectral-to-Frobenius ratio $\zeta(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])$ for the discretized second-derivative operator. The ratio is plotted as a function of system size n_s for the LNN and 2DL ansätze, using the simulations shown in figure 6(a). Although the ratio is considerably smaller than its upper bound, the data are consistent with superpolynomial scaling. Oscillations in the numerical results, however, prevent a reliable fit. No regularization is used in these simulations ($\rho = 0$).

Figure 12(b) reports the spectral-to-Frobenius ratio for the panel-method operator without ($\rho = 0$) and with ($\rho = 10^{-6}$) regularization, see section 3.2.2. In this example, regularization reduces both the average relative error and ζ.

Overall, ζ provides a simple measure of error concentration. In most of the cases tested, minimizing the average error yields worst-case and typical errors that do not differ substantially.

Appendix B. Influence of the normalization factor

This appendix illustrates the role of the stabilizing term μc^2 in equation (7) for the discretized second-derivative operator of section 3.2.1 with $n_s = 6$. For this operator, $c_{\mathrm{ideal}} = \|A\|_2 \approx 3.998$. If the residual is small, $A \approx c\,\mathrm{Tr}_{\mathrm{anc}}[PG]$, the average success probability, equation (2), behaves as

$$p(PGP) \approx \frac{\|A\|_{\mathrm{fro}}^2}{2^{n_s} c^2} = p(A/\|A\|_2) \left(\frac{c_{\mathrm{ideal}}}{c}\right)^2, \tag{B1}$$

with the operator-intrinsic value $p(A/\|A\|_2) \approx 0.3734$, see figure 6(b). Any increase of c beyond c_{ideal} therefore reduces the success probability quadratically.

Table 1 compares four configurations, using the same optimization settings as in section 3.2.1: (A) no hierarchical optimization, (B_0) hierarchical optimization without the stabilizing term ($\mu = 0$), (B)

Table 1. Normalization factor c, relative error $\varepsilon(A, c\,\mathrm{Tr}_{\mathrm{anc}}[PG])$ and average success probability p for the discretized second-derivative operator with $n_s = 6$. The operator-intrinsic values are $c_{\mathrm{ideal}} \approx 3.998$ and $p(A/\|A\|_2) \approx 0.3734$.

Configuration	μ	ρ	$c \approx$	$\varepsilon \approx$	$p \approx$
A: no hierarchical	10^{-6}	0	3.64	2.79×10^{-1}	0.4146
B_0: hierarchical	0	0	96.74	2.07×10^{-3}	0.0006
B: hierarchical	10^{-6}	0	4.07	1.72×10^{-3}	0.3597
C: hierarchical + regularized	10^{-6}	4×10^{-8}	4.06	7.17×10^{-4}	0.3628

hierarchical optimization with $\mu = 10^{-6}$, and (C) hierarchical optimization with $\mu = 10^{-6}$ and regularization ($\rho = 4 \times 10^{-8}$). Configurations A, B, and C correspond to the experiments shown in figure 7.

For the inaccurate result of configuration A, c falls slightly below c_{ideal} and, consequently, p slightly exceeds the operator-intrinsic value. This is possible because $c\,\mathrm{Tr}_{\mathrm{anc}}[PG]$ is only an approximation to the discretized second-derivative operator, whereas the bound $c \geqslant c_{\mathrm{ideal}}$ applies to exact encodings ($\tau = 0$), see section 2.1. Without the stabilizing term (B_0), the optimization converges to a normalization factor $c \gg c_{\mathrm{ideal}}$ and, in accordance with equation (B1), to a correspondingly small success probability. Since the residual depends only on the product $c\,\mathrm{Tr}_{\mathrm{anc}}[PG]$, a smaller block combined with a larger c reproduces A equally well, and the cost function provides no incentive to prefer $c \approx c_{\mathrm{ideal}}$. With $\mu = 10^{-6}$ (B), c remains close to c_{ideal}, the success probability increases by almost three orders of magnitude, and the relative error is even slightly lower than for B_0. Experiment C achieves lower overall error levels, while its success probability, $p_C \approx 0.3628$, remains almost identical to $p_B \approx 0.3597$ of experiment B.

ORCID iDs

Paul Over 0000-0001-7436-5254
Sergio Bengoechea 0009-0001-8205-5878
Leonardo Borello Busilacchi 0009-0009-7755-382X
Martin Kiffner 0000-0002-8321-6768
Thomas Rung 0000-0002-3454-1804
Alexios A Michailidis 0000-0002-8443-1064

References

[1] Press W H, Teukolsky S A, Vetterling W T and Flannery B P 2007 *Numerical Recipes 3rd edn: The Art of Scientific Computing* (Cambridge University Press) (available at: www.cambridge.org/9780521880688)
[2] Slotnick J P, Khodadoust A, Alonso J, Darmofal D, Gropp W, Lurie E and Mavriplis D J 2014 CFD vision 2030 study: a path to revolutionary computational aerosciences *Technical Report* (available at: https://ntrs.nasa.gov/citations/20140003093)
[3] Khan H N, Hounshell D A and Fuchs E R H 2018 Science and research policy at the end of Moore's Law *Nat. Electron.* **1** 14
[4] Esmaeilzadeh H, Blem E, Amant R S, Sankaralingam K and Burger D 2011 Dark silicon and the end of multicore scaling *2011 38th Annual Int. Symp. on Computer Architecture (ISCA)* pp 365–76 (available at: https://ieeexplore.ieee.org/abstract/document/6307773)
[5] Givi P, Daley A J, Mavriplis D and Malik M 2020 Quantum speedup for aeroscience and engineering *AIAA J.* **58** 3715
[6] Bluvstein D *et al* 2026 A fault-tolerant neutral-atom architecture for universal quantum computation *Nature* **649** 39
[7] Campbell E T, Terhal B M and Vuillot C 2017 Roads towards fault-tolerant universal quantum computation *Nature* **549** 172
[8] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505
[9] Preskill J 2018 Quantum Computing in the NISQ era and beyond *Quantum* **2** 79
[10] Camps D and Van Beeumen R 2020 Approximate quantum circuit synthesis using block encodings *Phys. Rev. A* **102** 052411
[11] Hoefler T, Häner T and Troyer M 2023 Disentangling hype from practicality: On realistically achieving quantum advantage *Commun. ACM* **66** 82–87
[12] Möttönen M, Vartiainen J J, Bergholm V and Salomaa M M 2005 Transformation of quantum states using uniformly controlled rotations *Quantum Inf. Comput.* **5** 467
[13] Evered S J *et al* 2023 High-fidelity parallel entangling gates on a neutral-atom quantum computer *Nature* **622** 268
[14] Debnath S, Linke N M, Figgatt C, Landsman K A, Wright K and Monroe C 2016 Demonstration of a small programmable quantum computer with atomic qubits *Nature* **536** 63
[15] Wang J, Sciarrino F, Laing A and Thompson M G 2020 Integrated photonic quantum technologies *Nat. Photon.* **14** 273
[16] Shirakawa T, Ueda H and Yunoki S 2024 Automatic quantum circuit encoding of a given arbitrary quantum state *Phys. Rev. Res.* **6** 043008
[17] An D, Liu J-P and Lin L 2023 Linear combination of hamiltonian simulation for nonunitary dynamics with optimal state preparation cost *Phys. Rev. Lett.* **131** 150603
[18] Orús R 2019 Tensor networks for complex quantum systems *Nat. Rev. Phys.* **1** 538
[19] Oseledets I V 2011 Tensor-train decomposition *SIAM J. Sci. Comput.* **33** 2295
[20] Gourianov N, Lubasch M, Dolgov S, van den Berg Q Y, Babaee H, Givi P, Kiffner M and Jaksch D 2022 A quantum-inspired approach to exploit turbulence structures *Nat. Comput. Sci.* **2** 30

[21] Kiffner M and Jaksch D 2023 Tensor network reduced order models for wall-bounded flows *Phys. Rev. Fluids* **8** 124101
[22] van Hülst N-L, Siegl P, Over P, Bengoechea S, Hashizume T, Cecile M G, Rung T and Jaksch D 2026 Quantum-inspired tensor-network fractional-step method for incompressible flow in curvilinear coordinates *Comput. Phys. Commun.* **325** 110169
[23] Michailidis A A, Fenton C and Kiffner M 2025 Element-wise multiplication of tensor trains *SIAM J. Sci. Comput.* **47** B1158
[24] Lubasch M, Joo J, Moinier P, Kiffner M and Jaksch D 2020 Variational quantum algorithms for nonlinear problems *Phys. Rev. A* **101** 010301(R)
[25] Ran S-J 2020 Encoding of matrix product states into quantum circuits of one- and two-qubit gates *Phys. Rev. A* **101** 032310
[26] Lin S-H, Dilip R, Green A G, Smith A and Pollmann F 2021 Real- and imaginary-time evolution with compressed quantum circuits *PRX Quantum* **2** 010342
[27] Zhou P-F, Hong R and Ran S-J 2021 Automatically differentiable quantum circuit for many-qubit state preparation *Phys. Rev. A* **104** 042601
[28] Rudolph M S, Chen J, Miller J, Acharya A and Perdomo-Ortiz A 2023 Decomposition of matrix product states into shallow quantum circuits *Quantum Sci. Technol.* **9** 015012
[29] Szołdra T, Mukherjee R and Schmelcher P 2026 Scalable preparation of matrix product states with sequential and brick wall quantum circuits *Quantum Sci. Technol.* **11** 025056
[30] Barratt F, Dborin J, Bal M, Stojevic V, Pollmann F and Green A G 2021 Parallel quantum simulation of large systems on small NISQ computers *npj Quantum Inf.* **7** 79
[31] Tepaske M S J, Hahn D and Luitz D J 2023 Optimal compression of quantum many-body time evolution operators into brickwall circuits *SciPost Phys.* **14** 073
[32] Mansuroglu R, Eckstein T, Nützel L, Wilkinson S A and Hartmann M J 2023 Variational hamiltonian simulation for translational invariant systems via classical pre-processing *Quantum Sci. Technol.* **8** 025006
[33] Mc Keever C and Lubasch M 2023 Classically optimized Hamiltonian simulation *Phys. Rev. Res.* **5** 023146
[34] Mc Keever C and Lubasch M 2024 Towards adiabatic quantum computing using compressed quantum circuits *PRX Quantum* **5** 020362
[35] Robertson N, Akhriev A, Vala J and Zhuk S 2025 Approximate quantum compiling for quantum simulation: A tensor network based approach *ACM Trans. Quantum Comput.* **6** 1
[36] Zhang Y, Wiersema R, Carrasquilla J, Cincio L and Kim Y B 2026 Scalable quantum dynamics compilation via quantum machine learning *Phys. Rev. Res.* **8** 023128
[37] Causer L, Jung F, Mitra A, Pollmann F and Gammon-Smith A 2024 Scalable simulation of nonequilibrium quantum dynamics via classically optimized unitary circuits *Phys. Rev. Res.* **6** 033062
[38] Martin B A, Ayral T, Jamet F, Rančić M J and Simon P 2024 Combining matrix product states and noisy quantum computers for quantum simulation *Phys. Rev. A* **109** 062437
[39] Gibbs J and Cincio L 2025 Deep circuit compression for quantum dynamics via tensor networks *Quantum* **9** 1789
[40] Kotil A, Banerjee R, Huang Q and Mendl C B 2024 Riemannian quantum circuit optimization for Hamiltonian simulation *J. Phys. A: Math. Theor.* **57** 135303
[41] Le I N M, Sun S and Mendl C B 2025 Riemannian quantum circuit optimization based on matrix product operators *Quantum* **9** 1833
[42] Termanova A, Melnikov A, Mamenchikov E, Belokonev N, Dolgov S, Berezutskii A, Ellerbrock R, Mansell C and Perelshtein M R 2024 Tensor quantum programming *New J. Phys.* **26** 123019
[43] Nibbi M and Mendl C B 2024 Block encoding of matrix product operators *Phys. Rev. A* **110** 042427
[44] Wei Z-Y, Malz D and Cirac J I 2022 Sequential generation of projected entangled-pair states *Phys. Rev. Lett.* **128** 010607
[45] Wei Z-Y, Malz D and Cirac J I 2023 Efficient adiabatic preparation of tensor network states *Phys. Rev. Res.* **5** L022037
[46] Guo Y and Yang S 2025 Efficient quantum circuit compilation for near-term quantum advantage *EPJ Quantum Technol.* **12** 69
[47] Evenbly G and Vidal G 2009 Algorithms for entanglement renormalization *Phys. Rev. B* **79** 144108
[48] Harrow A W, Hassidim A and Lloyd S 2009 Quantum algorithm for linear systems of equations *Phys. Rev. Lett.* **103** 150502
[49] Suzuki M 1990 Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations *Phys. Lett. A* **146** 319
[50] Jaksch D, Givi P, Daley A J and Rung T 2023 Variational quantum algorithms for computational fluid dynamics *AIAA J.* **61** 1885
[51] Over P, Bengoechea S, Rung T, Clerici F, Scandurra L, de Villiers E and Jaksch D 2025 Boundary treatment for variational quantum simulations of partial differential equations on quantum computers *Comput. Fluids* **288** 106508
[52] Bengoechea S, Over P, Jaksch D and Rung T 2026 Toward variational quantum algorithms for generalized linear and nonlinear transport phenomena *AIAA J.* **64** 585
[53] Brearley P and Laizet S 2024 Quantum algorithm for solving the advection equation using Hamiltonian simulation *Phys. Rev. A* **110** 012430
[54] Over P, Bengoechea S, Brearley P, Laizet S and Rung T 2025 Quantum algorithm for the advection-diffusion equation by direct block encoding of the time-marching operator *Phys. Rev. A* **112** L010401
[55] Bengoechea S, Over P and Rung T 2026 Quantum time-marching algorithms for solving linear transport problems including boundary conditions *Int. J. Numer. Methods Eng.* **127** e70326
[56] Gilyén A, Su Y, Low G H and Wiebe N 2019 Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics *Proc. 51st Annual ACM SIGACT Symp. on Theory of Computing (STOC 2019)* (Association for Computing Machinery) pp 193–204
[57] Nocedal J and Wright S J 2006 *Numerical Optimization* 2nd edn (Springer) (https://doi.org/10.1007/978-0-387-40065-5)
[58] Absil P-A, Mahony R and Sepulchre R 2008 *Optimization Algorithms on Matrix Manifolds* (Princeton University Press) (https://doi.org/10.1515/9781400830244)
[59] Luchnikov I A, Krechetov M E and Filippov S N 2021 Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies *New J. Phys.* **23** 073006
[60] Liao H-J, Liu J-G, Wang L and Xiang T 2019 Differentiable programming tensor networks *Phys. Rev. X* **9** 031041
[61] Rumelhart D E, Hinton G E and Williams R J 1986 Learning representations by back-propagating errors *Nature* **323** 533
[62] Fan K and Hoffman A J 1955 Some metric inequalities in the space of matrices *Proc. Am. Math. Soc.* **6** 111
[63] Gharibyan H, Su V P and Tepanyan H 2024 Hierarchical learning for training large-scale variational quantum circuits *Int. Conf. on Machine Learning and Applications (ICMLA)* pp 1810–4
[64] Kingma D P and Ba J 2015 Adam: A method for stochastic optimization *3rd Int. Conf. on Learning Representations (ICLR)*

[65] Larocca M, Thanasilp S, Wang S, Sharma K, Biamonte J, Coles P J, Cincio L, McClean J R, Holmes Z and Cerezo M 2025 Barren plateaus in variational quantum computing *Nat. Rev. Phys.* **7** 174
[66] Sack S H, Medina R A, Michailidis A A, Kueng R and Serbyn M 2022 Avoiding barren plateaus using classical shadows *PRX Quantum* **3** 020365
[67] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 Barren plateaus in quantum neural network training landscapes *Nat. Commun.* **9** 4812
[68] Friedrich L and Maziero J 2022 Avoiding barren plateaus with classical deep neural networks *Phys. Rev. A* **106** 042433
[69] Shende V V, Markov I L and Bullock S S 2004 Minimal universal two-qubit controlled-NOT-based circuits *Phys. Rev. A* **69** 062321
[70] Lieb E H and Robinson D W 1972 The finite group velocity of quantum spin systems *Commun. Math. Phys.* **28** 251
[71] Bravyi S B and Kitaev A Y 2002 Fermionic quantum computation *Ann. Phys.* **298** 210
[72] Seeley J T, Richard M J and Love P J 2012 The Bravyi-Kitaev transformation for quantum computation of electronic structure *J. Chem. Phys.* **137** 224109
[73] Pariser R and Parr R G 1953 A semi-empirical theory of the electronic spectra and electronic structure of complex unsaturated molecules. I *J. Chem. Phys.* **21** 466
[74] Pople J A 1953 Electron interaction in unsaturated hydrocarbons *Trans. Faraday Soc.* **49** 1375
[75] Barford W 2005 *Electronic and Optical Properties of Conjugated Polymers* (Oxford University Press) (https://doi.org/10.1093/oso/9780198526803.001.0001)
[76] Chandross M and Mazumdar S 1997 Coulomb interactions and linear, nonlinear and triplet absorption in poly(para-phenylenevinylene) *Phys. Rev. B* **55** 1497
[77] Ohno K 1964 Some remarks on the Pariser-Parr-Pople method *Theor. Chim. Acta* **2** 219
[78] Hess J and Smith A 1967 Calculation of potential flow about arbitrary bodies *Prog. Aerosp. Sci.* **8** 1
[79] Kazeev V A and Khoromskij B N 2012 Low-rank explicit QTT representation of the Laplace operator and its inverse *SIAM J. Matrix Anal. Appl.* **33** 742
[80] Batchelor G K 2000 *An Introduction to Fluid Dynamics* (*Cambridge Mathematical Library*) (Cambridge University Press) (https://doi.org/10.1017/CBO9780511800955)
[81] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 An adaptive variational algorithm for exact molecular simulations on a quantum computer *Nat. Commun.* **10** 3007
[82] Jaderberg B, Pennington G, Marshall K V, Anderson L W, Agarwal A, Lindoy L P, Rungger I, Mensa S and Crain J 2026 Variational preparation of normal matrix product states on quantum computers *Phys. Rev. Res.* **8** 013081
[83] Over P, Bengoechea S, Borello B L, Kiffner M, Rung T and Michailidis A A 2026 Operator learning for efficient quantum computation (available at: https://doi.org/10.25592/uhhfdm.18962)
[84] Golub G H and Van Loan C F 2013 *Matrix Computations* 4th edn (Johns Hopkins University Press) (https://doi.org/10.1137/1.9781421407944)
[85] Mele A A 2024 Introduction to Haar Measure Tools in Quantum Information: A Beginner's Tutorial *Quantum* **8** 1340