Physics-informed neural networks for an optimal counterdiabatic quantum computation

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 025035 https://doi.org/10.1088/2632-2153/ad450f

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
22 November 2023

REVISED
17 March 2024

ACCEPTED FOR PUBLICATION
29 April 2024

PUBLISHED
9 May 2024

PAPER

Physics-informed neural networks for an optimal counterdiabatic quantum computation

Antonio Ferrer-Sánchez[1,2,*], **Carlos Flores-Garrigos**[1], **Carlos Hernani-Morales**[1,3], **José J Orquín-Marqués**[1], **Narendra N Hegade**[4], **Alejandro Gomez Cadavid**[4], **Iraitz Montalban**[4], **Enrique Solano**[4], **Yolanda Vives-Gilabert**[1] and **José D Martín-Guerrero**[1,2,3]

[1] IDAL, Electronic Engineering Department, ETSE-UV, University of Valencia, Avgda. Universitat s/n, 46100 Burjassot Valencia, Spain
[2] Valencian Graduate School and Research Network of Artificial Intelligence (ValgrAI), Valencia, Spain
[3] Quantum Spain, 46100 Burjassot, Valencia, Spain
[4] Kipu Quantum, Greifswalderstrasse 226, 10405 Berlin, Germany
[*] Author to whom any correspondence should be addressed.

E-mail: Antonio.Ferrer-Sanchez@uv.es and anfesan6@uv.es

Keywords: physics-informed neural networks, quantum computing, shortcuts to adiabaticity, counterdiabatic driving, Pauli operators

Abstract

A novel methodology that leverages physics-informed neural networks to optimize quantum circuits in systems with N_Q qubits by addressing the counterdiabatic (CD) protocol is introduced. The primary purpose is to employ physics-inspired deep learning techniques for accurately modeling the time evolution of various physical observables within quantum systems. To achieve this, we integrate essential physical information into an underlying neural network to effectively tackle the problem. Specifically, the imposition of the solution to meet the principle of least action, along with the hermiticity condition on all physical observables, among others, ensuring the acquisition of appropriate CD terms based on underlying physics. This approach provides a reliable alternative to previous methodologies relying on classical numerical approximations, eliminating their inherent constraints. The proposed method offers a versatile framework for optimizing physical observables relevant to the problem, such as the scheduling function, gauge potential, temporal evolution of energy levels, among others. This methodology has been successfully applied to 2-qubit representing H_2 molecule using the STO-3G basis, demonstrating the derivation of a desirable decomposition for non-adiabatic terms through a linear combination of Pauli operators. This attribute confers significant advantages for practical implementation within quantum computing algorithms.

1. Introduction

Quantum computing, fueled by remarkable achievements in applied quantum machine learning, quantum simulation, and circuit/system optimization, has become a vibrant field of research [1–8]. Of particular interest are optimization problems, prevalent in various sectors like medicine, economics, logistics, and more [9–14]. Quantum computing has emerged as a promising alternative to classical approaches due to its potential for increased efficiency and speed. Recent experimental advances have fueled industrial and commercial interest, with the goal of achieving 'quantum supremacy' [15]. This surge in interest has led to progress in various scientific fields, where contemporary quantum computers serve as proof of concept. However, it is crucial to note that extensive research is needed to fully understand the applicability of these quantum algorithms, especially in the noisy intermediate-scale quantum (NISQ) era, characterized by quantum processors with up to 1000 qubits [16].

Hybrid classical-quantum algorithms, such as the variational quantum Eigensolver (VQE) [17], leverage both NISQ devices and classical computers to distribute computational tasks efficiently in quantum computing. VQE focuses on finding the lowest energy quantum state by combining variational quantum circuits with a Hamiltonian operator. This approach extends to quantum machine learning, where classical

algorithms are adapted to quantum counterparts, harnessing quantum principles like superposition and entanglement. For instance, Grover's search algorithm [18] demonstrates quadratic speedup in solving problems like k-medians [19] and k-nearest neighbors [20, 21]. Meanwhile, the quantum approximate optimization algorithm [22, 23] offers an alternative for solving combinatorial optimization problems using shallow quantum circuits optimized through classical methods. Recent research efforts have dedicated substantial attention to applying these techniques to ground-state challenges in physical systems [24–26], demonstrating the ongoing efforts to broaden their applications in quantum optimization. In recent years, significant attention has been focused on adiabatic quantum optimization (AQO) methods [27, 28] with practical applications in physics and chemistry [29–31]. AQO algorithms begin by initializing a quantum system in a known Hamiltonian's ground state and slowly evolving it into one representing the problem to be solved, following the adiabatic theorem [32, 33]. However, practical implementation often requires speeding up adiabatic evolution, leading to various approaches like shortcuts to adiabaticity (STA) [34, 35], including fast-forward methods, invariant-based inverse engineering, and counterdiabatic (CD) protocols, which is the main focus of this study. Readers interested in more detailed developments on quantum optimization processes for CD protocols and direct applications are directed to works such as [36, 37] (as well as references therein).

CD protocols are designed to accelerate adiabatic evolution by incorporating non-adiabatic terms to nullify transition amplitudes between energy eigenstates of the original Hamiltonian [38]. This results in a modified Hamiltonian for practical applications,

$$\boldsymbol{\mathcal{H}}(t) := \boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t) + \boldsymbol{\mathcal{H}}_{\mathrm{CD}}(t). \tag{1}$$

In the context of quantum dynamics, the operator $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$ in (1) is customized to prepare the ground energy state initially. However, the primary challenge of CD protocols is accurately determining the operator $\boldsymbol{\mathcal{H}}_{\mathrm{CD}}(t)$, especially in complex many-body quantum systems [39]. Typically, a time-dependent external parameterization, $\boldsymbol{\lambda}(t)$, is introduced, to which the operators are linked [40]. Significant progress, such as the nested commutators (NC) methodology [41] and contributions like [23, 42], has simplified the computation of non-adiabatic terms by using commutators involving $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$ and its derivatives with respect to $\boldsymbol{\lambda}(t)$. However, as the complexity of the system and expansion order increase, so does the difficulty of obtaining these terms. These methodologies may also exhibit problem-specific characteristics, requiring alternative approaches for non-trivial physical scenarios. On the other hand, combinatorial and optimization problems are of significant interest across various industries and research domains. Adiabatic quantum computing (AQC) algorithms are used to solve this kind of problems and they are expected to outperform classical computers in the current NISQ era [43]. In this paradigm, we can prepare a Hamiltonian in an initial ground state, driving or mixing Hamiltonian, and evolve it towards a desired final operator, whose ground state encodes the solution to the underlying problem, or it can also be the solution itself. This initial or driving Hamiltonian operator should be easy to prepare and evolve. We shall commence by establishing the Hamiltonian operator associated with the adiabatic transition of the system, $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$, which is characterized by its energy eigenvalues, $E_n(t)$, and corresponding eigenstates, $|n(t)\rangle$, as determined by

$$\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)\,|n(t)\rangle = E_n(t)\,|n(t)\rangle. \tag{2}$$

The adiabatic theorem ensures that the energy of a quantum system will remain close to the initial state if the amount of time elapsed during the total evolution is sufficiently large for the transition process to take place in a smooth manner. In order to achieve this, the total Hamiltonian operator governing the dynamics in (1) needs to exhibit transition amplitudes between energy levels that are close (or even equal) to zero, thereby eliminating any undesired transition between eigenstates [44, 45]. Within the CD protocols this total operator is defined containing the non-adiabatic terms, which together with the external temporal parameterization encompasses the CD part of the evolution. As stated in the appendix B, this parameterization $\boldsymbol{\lambda}(t)$ can be seen as a set of various components. However, in order to be in concordance with previous and current literature in the field, this parameterization will be reduced to a scalar function $\lambda(t)$, commonly named as the 'scheduling function' of the evolution. Consequently, the resulting Hamiltonian for a $\mathrm{N_Q}$-qubit system undergoing a CD evolution can be written as follows in (3),

$$\boldsymbol{\mathcal{H}}(t) := \underbrace{(1-\lambda(t))\,\boldsymbol{\mathcal{H}}_{\mathrm{initial}} + \lambda(t)\,\boldsymbol{\mathcal{H}}_{\mathrm{final}}}_{\boldsymbol{\mathcal{H}}_{\mathrm{AD}}} + \frac{\mathrm{d}\lambda}{\mathrm{d}t}\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t). \tag{3}$$

The $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ that has been just introduced is known in the literature as adiabatic gauge potential (AGP), where the derivatives of the eigenstates of the system with respect to the external parameterization take the main role leaving apart the temporal ones (see appendix B for further details). It can be shown that this

operator satisfies the condition in (4), which is equivalent to minimizing the action $\mathcal{S}$ in (B.6) for the Hilbert–Schmidt operator $\boldsymbol{\mathcal{G}}_\lambda$. Therefore, this restriction can be seen as the Euler–Lagrange equations governing the system,

$$\left[i\frac{\partial \boldsymbol{\mathcal{H}}_{\mathrm{AD}}}{\partial \lambda} - \left[\boldsymbol{\mathcal{A}}_{\mathrm{CD}}, \boldsymbol{\mathcal{H}}_{\mathrm{AD}}\right], \boldsymbol{\mathcal{H}}_{\mathrm{AD}}\right] = 0. \tag{4}$$

This paper primarily focuses on finding the ground state of H_2 molecule using the STO-3G basis with varying bond distances, employing the 2-qubit initial and final Hamiltonians in time with coefficients dependent on bond distances, obtained via *Qiskit* [46]. In a different context, the Pauli matrices, encompassing $\sigma_X, \sigma_Y, \sigma_Z$, and the identity matrix $\boldsymbol{\mathcal{I}}$ (or σ_0), form an orthogonal basis for the Hilbert space of 2×2 Hermitian matrices. This basis expansion extends to N_Q qubits, allowing a comprehensive representation of Hermitian operators, aiding in quantum circuit analysis and quantum system manipulation [47, 48].

We begin with an easily preparable *driving* Hamiltonian, $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$, that is evolved through time to a certain endpoint, namely $\boldsymbol{\mathcal{H}}_{\mathrm{final}}$. The adiabatic evolution, considering non-adiabatic terms, is solved by incorporating the adiabatic operator into the total Hamiltonian operator. With this information and the introduction of the gauge potential $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}$, it is possible to establish the partial differential equation (PDE) in (3) as the driving equation, pursuing accelerated adiabatic temporal evolution while avoiding unexpected energy state transitions. This relation will be used for all systems presented in this manuscript, regardless of the number of qubits. However, in appendix C we try to show more explicitly the Hamiltonian matrices used for the H_2 use case presented in the Results section 3.

In order to solve this scenario trying to avoid any theoretical approximation, we propose an approach to obtain CD terms in CD protocols using the deep learning (DL)-based physics-informed neural-networks (PINNs) methodology. This approach directly utilizes DL to resolve the physics, obtaining the CD terms and the temporal parameterization $\lambda(t)$ via the neural network. To assess experimental feasibility, we decompose the non-adiabatic operator into tensor products of Pauli and identity matrices, providing a direct applicability assessment.

PINNs leverage neural networks for physical problem solving by incorporating physical knowledge. It is possible to utilize automatic differentiation to construct differential equations based on network outputs [49]. Minimization guided by a designated loss function adjusts trainable parameters to align with the physical framework. Physical constraints, initial and boundary conditions, among other considerations, are integrated into the total cost function as separate terms, each one denoted as $\mathcal{L}_i$ in (5) and constituting together the *a priori* knowledge available for the problem. Additionally, $\mathcal{L}_{\mathcal{F}}$ quantifies adherence to the system of PDEs, and the vector $(\omega_{\mathcal{F}}, \omega_i)$ represents the weights assigned to each term within the mixture. In general, these terms can be computed using metrics such as the mean squared error (MSE) or the mean absolute error (MAE), depending on the problem (for a further detailed background see appendix A),

$$\mathcal{L} := \omega_{\mathcal{F}}\mathcal{L}_{\mathcal{F}} + \sum_i \omega_i \mathcal{L}_i. \tag{5}$$

The main purpose of the CD driving protocols is to obtain the AGP, $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$, for a certain discrete time where the non-adiabatic terms are introduced into the circuit. This instant in time usually comes defined by a sigmoid-shaped $\lambda(t)$, thereby having the biggest value of its derivative at a some certain point in time. However, with PINNs what we do is to simulate the time continuously having then access to the entire evolution of the system. Moreover, by using DL techniques to compute the AGP and the scheduling, we also avoid considering approximations and numerical problems of instability. In addition, exact explicit diagonalization of the resulting Hamiltonian, $\boldsymbol{\mathcal{H}}(t)$, is no longer necessary since energies do not play any role in the loss function, as we will see in the following Methods section 2.

The remainder of this paper is structured as follows. First, section 'Methods' presents a detailed adaptation of the PINN methodology for our specific case, considering relevant physical factors. Afterwards, we present some general numerical results in section 'Results', subsection 'Numerical experiments'. In subsection 'Scheduling function, $\lambda(t)$' we derive the optimal method for characterizing the temporal parameterization that governs the CD process, while in subsection 'Temporal evolution of the $\boldsymbol{\mathcal{H}}(t)$ operator and the energy levels' we explicitly determine the eigenvalues of the final Hamiltonian operator with respect to time. Moreover, subsection '$\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ operator and its decomposition' presents the temporal evolution of the components of the CD operator while the scalability of the method is investigated in the 'Scalability' subsection. Finally, in the 'Conclusion' section, the main achieved results are discussed, outlining key areas for improvement in future research. At the end of the article, three appendices have been included where

concepts that have to do with the general methodology behind the PINN are discussed, as well as theoretical concepts behind the approach that go a little beyond the central idea presented in this manuscript.

2. Methods

The presented methodology uses a vanilla PINN without extra modifications in the architecture, thereby including all the inductive biases through the loss function. The neural part of the model will consist of a set of K fully connected layers which gives as output the set of physical variables of interest (the intrigued reader is directed to appendix A and references therein). The incorporation of physical knowledge through inductive biases is intended to ensure that the underlying physics governing the problem is adequately satisfied. To achieve this objective, it is essential to consider that the output of the underlying network in our methodology will consist of a set of variables denoted as

$$\boldsymbol{\mathcal{U}}_{\Theta}(t) := (\lambda, \boldsymbol{\mathcal{A}}_{\mathrm{CD}}, \boldsymbol{\mathcal{C}})_{\Theta}(t). \tag{6}$$

Henceforth, the utilization of the symbol 'Θ' to denote the network prediction will be omitted and presumed understood, except in cases where the terminology may potentially result in misconceptions. Here we introduce the scheduling function $\lambda \in \mathbb{R}$ and the AGP operator $\boldsymbol{\mathcal{A}}_{\mathrm{CD}} \in \mathcal{M}_{2^{\mathrm{N_Q}} \times 2^{\mathrm{N_Q}}}(\mathbb{C})$ associated with quantum evolution. The coefficients $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{4^{\mathrm{N_Q}}}$ represent the linear decomposition of $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}$, considering all possible tensor products of identity and Pauli matrices, $\{\boldsymbol{\mathcal{I}}, \sigma_{\mathrm{X}}, \sigma_{\mathrm{Y}}, \sigma_{\mathrm{Z}}\}$ (see [50] chapter 2.1). The size of $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}$ is $2^{\mathrm{N_Q}} \times 2^{\mathrm{N_Q}}$, where $\mathrm{N_Q}$ represents the number of qubits. All variables from the PINN depend on the input time, yielding solutions for various inference times. We optimize the neural network using the 'soft enforcement' technique whose general cost function is defined in (5). It is adapted to the Hamiltonian dynamics with loss functions for initial and final time conditions within the interval $t \in [t_{\min}, t_{\max}]$ ((7) and (8)),

$$\mathcal{L}_{\mathcal{IC}} := \omega_{\mathcal{IC},1} \left|\lambda(t_{\min})\right|^2 + \omega_{\mathcal{IC},2} \left|\boldsymbol{\mathcal{H}}(t_{\min}) - \boldsymbol{\mathcal{H}}_{\mathrm{initial}}\right|^2, \tag{7}$$

$$\mathcal{L}_{\mathcal{FC}} := \omega_{\mathcal{FC},1} \left|\lambda(t_{\max}) - 1\right|^2 + \omega_{\mathcal{FC},2} \left|\boldsymbol{\mathcal{H}}(t_{\max}) - \boldsymbol{\mathcal{H}}_{\mathrm{final}}\right|^2. \tag{8}$$

In the context provided, $(\omega_{\mathcal{IC},1}, \omega_{\mathcal{IC},2})$ and $(\omega_{\mathcal{FC},1}, \omega_{\mathcal{FC},2})$ represent weights for the total loss function, influenced by problem-specific factors. Moreover, $\mathrm{N}_{\mathcal{F}}$ represents the amount of data for the interior part of the sampling where the underlying PDEs are computed, i.e. $t \in (t_{\min}, t_{\max})$ (also known as 'collocation points' in the related literature). The scheduling function $\lambda(t)$ guides the evolution of the system with constraints: $\lambda(t_{\min}) = 0$ and $\lambda(t_{\max}) = 1$. The Hamiltonian operators at the beginning and end of the evolution, $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$ and $\boldsymbol{\mathcal{H}}_{\mathrm{final}}$, are computed using numerical methods in chemistry [46]. These matrices together with the values of the scheduling at those points will be the available beforehand information. Once the initial and final inductive biases are established, it is essential to clarify the physics governing intermediate time periods within the interval. With all these components, we can construct the Hamiltonian operator, as it has been shown in (3).

The Hamiltonian operators are well-known for their general form, involving complex numbers and the crucial property of Hermiticity, making them interpretable as physical observables with real energy state values. Therefore, it is imperative that our operators possess Hermitian properties. Additionally, our neural network must find the solution for the AGP that minimizes the physical action within the scenario,

$$\frac{\delta \mathcal{S}(\boldsymbol{\mathcal{A}}_{\mathrm{CD}})}{\delta \boldsymbol{\mathcal{A}}_{\mathrm{CD}}} = 0. \tag{9}$$

Ensuring optimal operator $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ within specific conditions, including local development, robustness, availability, and experimental accessibility, hinges on minimizing physical action through (9), as evidenced by research studies such as [44]. This minimization aligns with defining the term related to physical loss as written in (10). Additionally, the temporal rate of change of the scheduling function, $\mathrm{d}\lambda/\mathrm{d}t$, dictates the influence of the non-adiabatic components on the system evolution under the total Hamiltonian operator. When $\mathrm{d}\lambda/\mathrm{d}t = 0$, the conventional adiabatic Hamiltonian is regained, but excessive CD influence should be avoided. Thus, the derivative must remain small but non-zero, as introduced in (11).

By implementing this constraint within the network, what it is achieved in reality is a rapid quasiadiabatic (FAQUAD) methodology (see [51]), wherein we try to speed up the evolution while incorporating non-adiabatic terms, without deviating significantly from the original adiabatic framework. As a result, the network can effectively navigate the solution space, approximating the adiabatic scenario at each time step while introducing a non-zero gauge potential to facilitate practical acceleration of the process.

While this may appear contradictory, as discussed in the section 3, a natural compromise emerges upon imposition of this constraint,

$$\mathcal{L}_{\mathrm{LA}} := \frac{\omega_{\mathrm{LA}}}{\mathrm{N}_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left| \left[i\frac{\partial \boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)}{\partial \lambda} - \left[\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t), \boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)\right], \boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t) \right] \right|^2. \tag{10}$$

$$\mathcal{L}_{\mathrm{Ad}} := \frac{\omega_{\mathrm{Ad}}}{\mathrm{N}_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left| \frac{\mathrm{d}\lambda}{\mathrm{d}t} \right|^2. \tag{11}$$

On the other hand, it is known that $\{\mathcal{I}, \sigma_{\mathrm{X}}, \sigma_{\mathrm{Y}}, \sigma_{\mathrm{Z}}\} \in \mathcal{M}_{2\times 2}(\mathbb{C})$ constitute an orthogonal basis for the 2×2 Hermitian matrix Hilbert space. Consequently, we propose computing the operator $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}$ not only as a direct output of the network but also through tensor products of these Pauli matrices, generating a linear combination denoted as $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{4^{\mathrm{N_Q}}}$. Each coefficient in $\boldsymbol{\mathcal{C}}$ represents the relative magnitude of each term in the operator, enabling efficient simulations and facilitating quantum circuit-based analysis of physical systems, as outlined in (12), for a more accessible approach to quantum computation and analysis,

$$\boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{'}(t) := \sum_{i,j,\ldots,\mathrm{N_Q} \in \{0,\mathrm{X},\mathrm{Y},\mathrm{Z}\}} \boldsymbol{\mathcal{C}}_{i,j,\ldots,\mathrm{N_Q}}(t) \left(\sigma_i \otimes \sigma_j \otimes \ldots \otimes \sigma_{\mathrm{N_Q}}\right). \tag{12}$$

In this study, we utilize $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{'}(t)$ to denote non-adiabatic terms, distinct from the PINN output operator. We introduce an extra term in the neural network loss function (13), directly influencing the $\boldsymbol{\mathcal{C}}(t)$ coefficients and dynamically adapting them during training to construct the Gauge operator decomposition. Through this approach, our methodology can compute these scalar values not just at the initial and final moments but across the entire time span, as these scalars are inherently time-dependent functions,

$$\mathcal{L}_{\mathrm{C}} := \frac{\omega_{\mathrm{C}}}{\mathrm{N}_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left| \boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t) - \boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{'}(t) \right|^2. \tag{13}$$

Once all requisite components are specified using (10), (11) and (13), the loss term for our PINN (14) is established, linked solely to the underlying differential equations. The vector $(\omega_{\mathrm{LA}}, \omega_{\mathrm{Ad}}, \omega_{\mathrm{C}})$ denotes the weights for the combination process in constructing the final term. Equation (5) and predefined mixture weights in our loss terms simplify the formulation of the ultimate loss function (15), encompassing PDE losses (14) and temporal constraints at initial and final time steps (7) and (8). This defined loss function serves as the guiding metric for optimization, directing the neural network to minimize its objectives within our methodology,

$$\mathcal{L}_{\mathcal{F}} := \mathcal{L}_{\mathrm{LA}} + \mathcal{L}_{\mathrm{Ad}} + \mathcal{L}_{\mathrm{C}}. \tag{14}$$

$$\mathcal{L} := \mathcal{L}_{\mathcal{IC}} + \mathcal{L}_{\mathcal{FC}} + \mathcal{L}_{\mathcal{F}}. \tag{15}$$

The network needs to satisfy operator hermiticity. This could be achieved by including an additional term in (14) to minimize $|\boldsymbol{\mathcal{A}}_{\mathrm{CD}} - \boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{\dagger}|^2$. However, this restriction is redundant for the PINN. Being the coefficients $\boldsymbol{\mathcal{C}} \in \mathbb{R}^{N_{\mathrm{Q}}}$ defined real ($\boldsymbol{\mathcal{C}} = \boldsymbol{\mathcal{C}}^*$), (12) naturally ensures $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{'} - \boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{'\dagger} = 0$, without additional requirements. Therefore, (13) suffices for the neural network to ensure hermiticity of the operator $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$. As both $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$ and $\boldsymbol{\mathcal{H}}_{\mathrm{final}} \in \mathcal{M}_{2^{\mathrm{N_Q}}\times 2^{\mathrm{N_Q}}}(\mathbb{R})$, hermiticity of the complete Hamiltonian operator is guaranteed.

We succinctly describe our methodology, employing key elements such as a visual aid in the form of a diagram in figure 1 and a step-by-step algorithm outlined in algorithm 1. Our analysis centers around a single independent variable, the time, which operates within the defined interval of $[0, 1]$ for our illustrative use cases. The crucial step involves selecting an appropriate sampling method, and in our case, we opt for the Sobol sampling technique. This method has shown substantial promise in previous studies within the field [52], characterized by its use of powers of two for generating pseudo-random samples. This results in a more evenly distributed set of data points, minimizing overlap compared to purely random sampling methods. However, it is essential to acknowledge the inherent randomness within this approach.

Moving forward, the time domain generated serves as the input for the neural. The architecture consists of a sequence of fully connected layers, with the number of layers and neurons per layer (N_k) predetermined prior to training. While these hyperparameters are flexible and can be adjusted, we anticipate that a relatively conventional configuration, such as 6 or 7 layers with 30 to 40 neurons each, will be suitable for characterizing the CD driving problem. Nevertheless, we remain open to adapting these parameters based on the complexity of individual problems, recognizing that certain cases may necessitate variations in the number of layers and neurons to achieve optimal results.

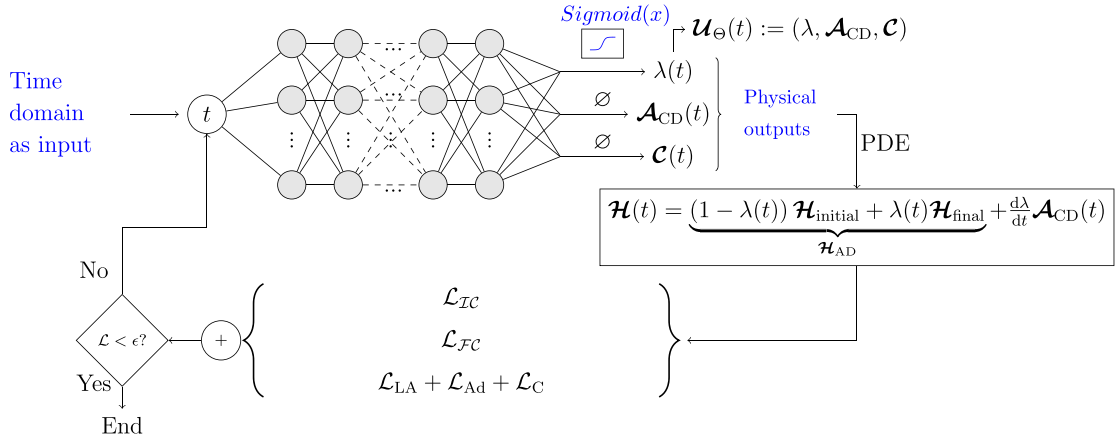

Figure 1. The methodology takes time as input, while the outputs consist of $\lambda(t)$, the scheduling function; $\mathcal{A}_{\mathrm{CD}}(t)$, representing counterdiabatic terms as separate components; and $\mathcal{C}(t)$, the coefficients expressing the decomposition of $\mathcal{A}_{\mathrm{CD}}(t)$ via Kronecker products involving identity and Pauli operators. The total Hamiltonian $\mathcal{H}(t)$ is computed while ensuring adherence to the necessary physical constraints. In each training step, the total loss $\mathcal{L}$ is computed by aggregating the different contributions, with training concluding upon reaching a certain amount of epochs or a predefined error threshold.

Algorithm 1. PINN algorithm for the counterdiabatic driving of a system of N_Q qubits.

Input: Physical domain: $t \in [t_{\min}, t_{\max}]$.

Output: Set of variables produced by the PINN comprising the scheduling function, the operator controlling the counterdiabatic terms, and the coefficients of its associated Pauli decomposition, denoted as $\mathcal{U}_\Theta(t) := (\lambda, \mathcal{A}_{\mathrm{CD}}, \mathcal{C})_\Theta$.

1 **Specify the training domain:** Generate the temporal interval t using *Sobol sequence* [53].

2 **Construct the underlying neural network** as a densely connected architecture consisting of K layers, with a specific number of neurons per layer. Initialize the trainable parameters of the network, denoted as $\Theta = \{\boldsymbol{W}^{(k)}, \boldsymbol{b}^{(k)}\}_{1 \leqslant k \leqslant K}$, using the Glorot initialization [54].

3 **By employing the principles of automatic differentiation**, compute the derivative $\mathrm{d}\lambda/\mathrm{d}t$ from the output and subsequently construct the operator $\mathcal{H}(t)$ in accordance with (3).

4 **Construct the initial and final losses in time** using the MSE metric as $\mathcal{L}_{\mathcal{IC}}$ and $\mathcal{L}_{\mathcal{FC}}$, ((7) and (8)), respectively.

5 **Derive the physical loss terms** for the differential equations $\mathcal{L}$, considering the initial and final specifications, the Principle of Least Action, adiabatic conditions, Pauli tensor products decomposition, and then express the aggregate physical loss accordingly.

$$\mathcal{L} := \mathcal{L}_{\mathcal{IC}} + \mathcal{L}_{\mathcal{FC}} + \mathcal{L}_{\mathcal{F}}.$$

Update the set of network variables by backpropagation, $\{\boldsymbol{W}^{(k)}, \boldsymbol{b}^{(k)}\}_{1 \leqslant k \leqslant K}$, thereby minimizing the total loss using a certain defined optimizer.

$$\Theta^* := \mathrm{argmin}\left(\mathcal{L}(\Theta)\right).$$

Repeat steps 3–6 for the required number of epochs until the convergence is as desired or a maximum number of iterations is reached.

The input domain and network construction have been established, allowing us to extract the output $\mathcal{U}_\Theta(t)$ representing physical variables ($\lambda \in \mathbb{R}$, $\mathcal{A}_{\mathrm{CD}} \in \mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$, $\mathcal{C} \in \mathbb{R}^{4^{N_Q}}$). We compute $\mathrm{d}\lambda/\mathrm{d}t$ using automatic differentiation to obtain the Hamiltonian operator $\mathcal{H}(t)$. We adhere to initial and final temporal constraints (7) and (8) and calculate the loss via the underlying PDEs (14) incorporating physical constraints (10), (11) and (13), adhering to the Principle of Least Action, adiabaticity, and the operator decomposition. We compute the final loss metric $\mathcal{L}$ (15) and update trainable variables through backpropagation to minimize the loss during optimization.

This methodology has been implemented for a 2-qubit representation to tackle the H_2 molecule problem in the STO-3G basis at varying bond distances as an illustrative use case. The Hamiltonian operators used for the evolution are obtained following the guidelines from [46], with a matrix size of 4×4 due to the representation. These operators are inherently real-valued, $\mathcal{H}_{\mathrm{initial}}, \mathcal{H}_{\mathrm{final}} \in \mathcal{M}_{4\times4}(\mathbb{R})$. Throughout the study, 500 000 epochs are utilized for training the PINN model, using the *PyTorch* module [55] and the *Adam*

optimizer [56] with a fixed learning rate of 10^{-5} in order to ensure convergence. Sobol sampling [53] is consistently employed for all examples, varying the number of collocation points. Additionally, the neural architecture for all experiments comprises six hidden layers, each with 30 neurons. For all the examples shown in this article, the mix weights considered for each of the individual loss terms are preset as

$$(\omega_{\mathcal{IC}}, \omega_{\mathcal{FC}}, \omega_{\mathrm{LA}}, \omega_{\mathrm{Ad}}, \omega_{\mathrm{C}}) = \left(10^3, 10^3, 10^2, 5 \times 10^{-1}, 2.5 \times 10^2\right). \tag{16}$$

The weights $\omega_{\mathcal{IC}}$ and $\omega_{\mathcal{FC}}$ in (7) and (8) are used to prioritize the satisfaction of initial and final conditions, crucial for our objectives, setting them significantly higher than ω_{LA}, ω_{Ad}, and ω_{C}. Emphasizing the minimization of the action, vital for identifying the appropriate operator $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$, we assign higher but still an order of magnitude lower weights. Notably, ω_{C} holds slightly higher importance, encompassing the hermiticity constraint, while adiabaticity is assigned the lowest weight to preserve the adiabatic theory's limit without dominating the overall metric.

3. Results

3.1. Numerical experiments

In this section, we present numerical results from our DL-based methodology, applied to solve the H_2 molecule problem in the STO-3G basis with varying bond distances, utilizing a 2-qubit representation. Figure 2 illustrates the physical loss functions used for neural network optimization. These loss curves vary between training processes due to factors like qubit count and molecular system specifics, particularly bond distances. In figure 2(a), $\mathcal{L}_{\mathcal{F}}$, $\mathcal{L}_{\mathcal{IC}}$, and $\mathcal{L}_{\mathcal{FC}}$ correspond to the cost function associated to the underlying PDEs and to the set of initial and final conditions in time, respectively. These are components of the total loss used for training, i.e. $\mathcal{L}$ in (15). The latter two quickly converge to values around 10^{-4} or lower, especially $\mathcal{L}_{\mathcal{FC}}$. Conversely, $\mathcal{L}_{\mathcal{F}}$ contains three terms with noticeable tension, as seen in figure 2(b). The term minimizing the principle of least action reaches the lowest value, ensuring a satisfactory minimization of action and a close adherence to the prescribed guidelines for the AGP. Additionally, there is a $\mathcal{L}_{\mathrm{Hermiticity}}$ term assessing the quadratic difference between $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ and $\boldsymbol{\mathcal{A}}^{\dagger}_{\mathrm{CD}}(t)$, confirming that minimizing $\mathcal{L}_{\mathrm{C}}$ in (13) enforces hermiticity due to real-valued coefficients $\boldsymbol{\mathcal{C}}(t)$ in the decomposition (12).

In the figure 2(b), the loss of recovery in the adiabatic theory ($\mathcal{L}_{\mathrm{Ad}}$) initially starts near zero and increases marginally before stabilizing around 10^0 throughout training, illustrating the limitation of the PINN in reducing the adiabatic speed ($\mathrm{d}\lambda/\mathrm{d}t$). This limitation arises from the complexity of the loss function, which includes terms unrelated to adiabaticity recovery, causing tension in addressing the physical problem. This rapid saturation of adiabatic recovery significantly impacts the elevated $\mathcal{L}_{\mathcal{F}}$ value in the left subfigure, emphasizing the need for an isolated term analysis. A closer examination of $\mathcal{L}_{\mathrm{Ad}}$ between iterations 30 000 and 50 000 reveals the transition of λ from sigmoidal to linear behavior with $\mathrm{d}\lambda/\mathrm{d}t = 1$, a topic further discussed in the following section.

For all exercises presented in the manuscript, the same GPU, an NVIDIA A100 with 40 GB of RAM, has been utilized to maintain consistency. All processes were executed on a cluster where four identical CPUs, along with the graphics card, were reserved for individual use during the entire training process, ensuring fidelity to the provided computational time data. In the case of the H_2 application shown in figure 2, the total training time for 500 000 iterations amounts to approximately 2 h and 30 min.

3.2. Scheduling function, $\lambda(t)$

Our methodology achieves three main outcomes: firstly, it extracts the time-dependent scheduling function λ; secondly, it retrieves the matrix components (i, j) of the AGP, $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$, accounting for the temporal variable, and, thirdly, it also obtains the components of its decomposition through time, $\boldsymbol{\mathcal{C}}(t)$, by employing a cost metric based on (14) with three distinct terms. Notably, the term $\mathcal{L}_{\mathrm{Ad}}$ enforces adherence of the function $\lambda(t)$ to the physical evolution, approximating the empirically validated adiabatic theory. Nevertheless, our PINN model gradually reduces its focus on optimizing $\mathcal{L}_{\mathrm{Ad}}$ after around 50 000 iterations, resulting in the scheduling function $\lambda(t)$ converging to a solution where its time derivative, representing the adiabatic velocity, remains approximately constant at 1 throughout time evolution.

In figure 3, we illustrate the evolution of $\lambda(t)$ and $\mathrm{d}\lambda/\mathrm{d}t$ during training for a 2-qubit system representing the H_2 molecule, emphasizing the persistence of a sigmoidal shape even after 15 000 and 25 000 epochs, a common representation in prior work [23, 42]. This sigmoidal shape visible in figure 3(b) is theoretically advantageous as it yields a 'bell curve' derivative, enabling enhanced adiabatic terms $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ at intermediate time points while diminishing at the start and end. Notably, this sigmoidal form emerges primarily in early training stages, aiding convergence. Consequently, the predicted optimal form of the λ function in figure 3(a) adheres to the condition $\lambda(t) = t$ for a complete training of 500 000 epochs. This achievement

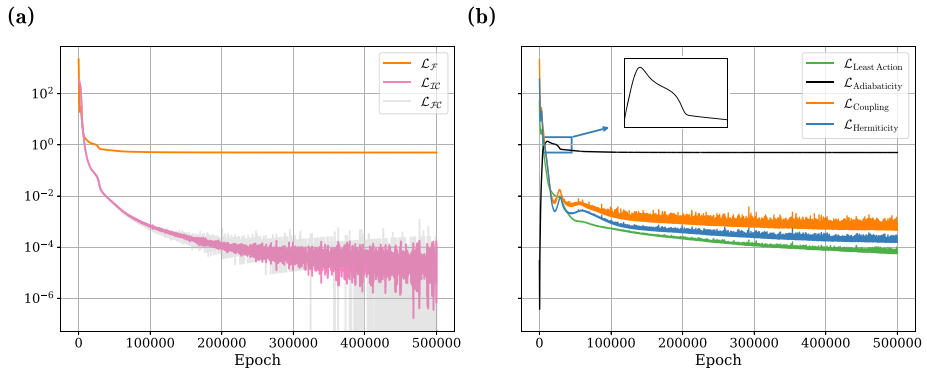

Figure 2. Analysis of the loss function during the training process of the H_2 use case for a bounding distance of $d = 1.0$ Å. (a) The dynamic changes of the components contributing to the total loss, $\mathcal{L}$, as defined in (15). (b) Each individual constituent of the collocation loss, $\mathcal{L}_{\mathcal{F}}$, is presented, corresponding to the different physical aspects under consideration. The loss term $\mathcal{L}_{\text{Hermiticity}}$ is included in the plot for illustrative purposes only since it remains unused throughout the training.

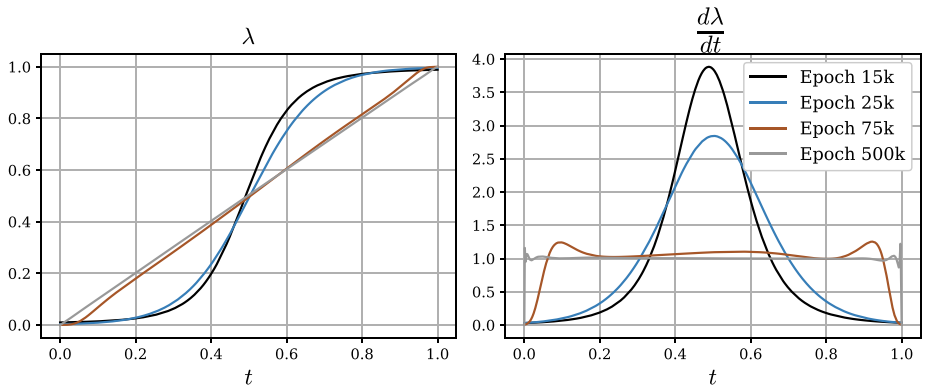

Figure 3. Analysis of the scheduling function λ (a) and its temporal derivative (b) for distinct iterations (epochs) of the training process. Observations reveal that during the initial stages of neural optimization, λ exhibits characteristics resembling a sigmoidal function. However, as the training advances, it converges towards a linear behavior ($d\lambda/dt \approx 1$).

reinstates the original concept of CD driving outlined in [38], eliminating the temporal parameterization of physical operators in favor of a specific scalar parameterization. Our aim is to reconcile the fully adiabatic theory with the inclusion of CD terms, leading the underlying neural network to converge to a function λ closely aligned with the temporal variable t transitioning from a differential equation with non-zero CD terms to recover adiabatic theory.

Drawing upon studies such as [51] and references therein, the loss term $\mathcal{L}_{\text{Ad}}$ can be interpreted as compelling the method to speed up the evolution as much as possible but without deviating substantially from the adiabatic limit. In that sense, the approach that we propose can be also seen as a 'fast-quasi adiabatic' (FAQUAD) protocol within the CD protocols. Consequently, the constructed PINN converges to $\lambda(t) = t$ after a few training iterations thereby leading to a constant velocity as the optimal solution. Therefore, a FAQUAD process emerges naturally with a small weight for the aforementioned term in the loss.

3.3. Temporal evolution of the $\mathcal{H}(t)$ operator and the energy levels

We can derive the total Hamiltonian operator $\boldsymbol{\mathcal{H}}(t)$ by minimizing the action of the system with respect to the gauge $\boldsymbol{\mathcal{A}}_{\text{CD}}(t)$. This operator belongs to $\mathcal{M}_{2^{N_Q}\times 2^{N_Q}}(\mathbb{C})$ for a 2-particle system, such as the H_2 use case, resulting in a 4×4 matrix. Figure 4 illustrates the time evolution of its components, showcasing the hermiticity property, where the real part exhibits symmetry and the imaginary part displays antisymmetry with values around $10^{-3} - 10^{-4}$. By ensuring hermiticity for $\boldsymbol{\mathcal{H}}(t)$, we can compute its instantaneous eigenvalues, providing insights into the energy levels of the system under investigation (figure 5). The findings reveal substantial fluctuations in the values across various time scales.

The operator $\boldsymbol{\mathcal{H}}(t)$, defined in (3), yields the eigenstates $|n(t)\rangle$ of $\boldsymbol{\mathcal{H}}_{\text{AD}}(t)$ over time, preventing transitions between energy levels $E_n(t)$. These 'moving eigenstates' can reveal energy levels for $\boldsymbol{\mathcal{H}}(t)$ during time evolution, compared to adiabatic transitions. We explored this for H_2 with bond distances $d \in \{1.0, 1.5, 2.0, 2.5\}$ Å in figure 5.

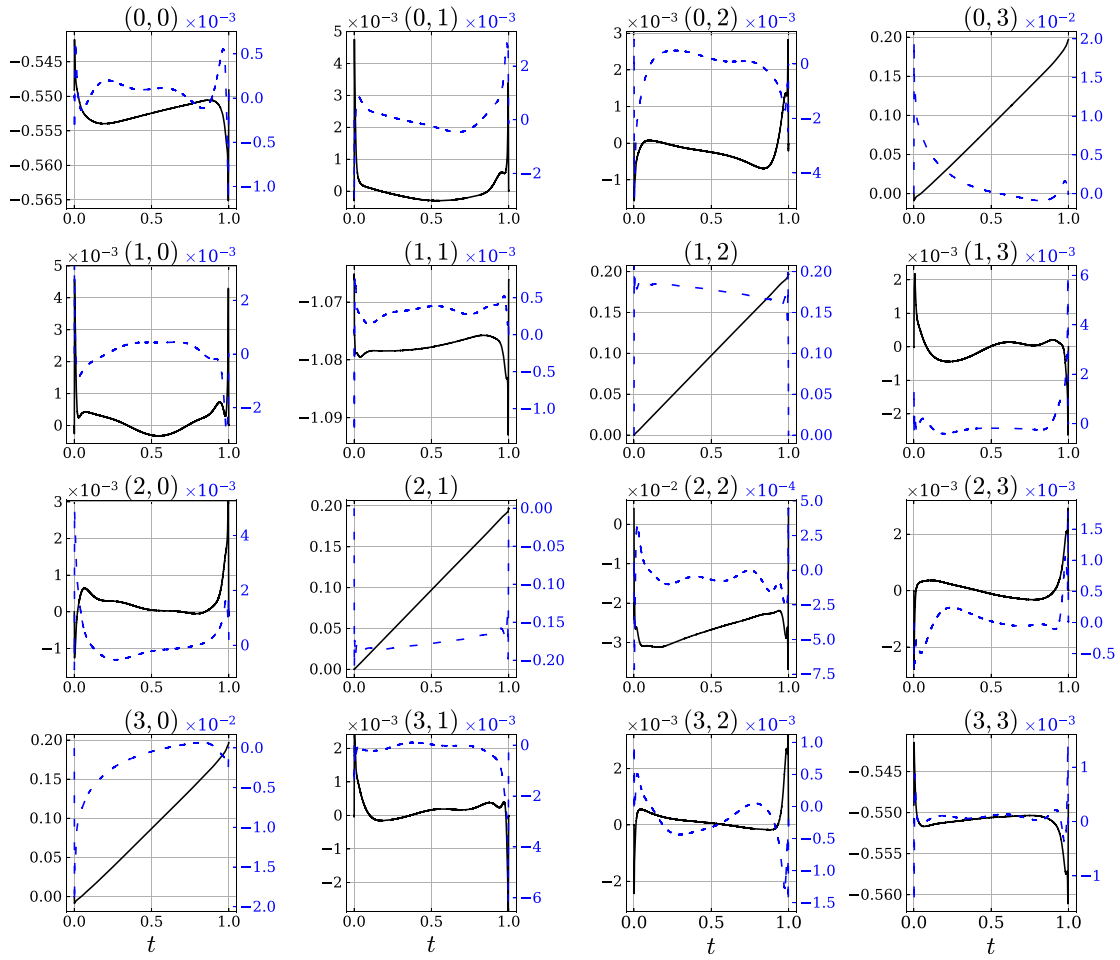

Figure 4. Evolution of the real and imaginary components of the Hamiltonian operator $\boldsymbol{\mathcal{H}}(t)$ for the H_2 molecule use case with a bond distance of $d = 1.0$ Å. Black and blue colors have been used for real and imaginary parts, respectively, using a dashed linestyle for the latter and a different scale for each one.

Regarding the real components in figure 5(a), it is observed that in fully adiabatic transitions, energy levels exhibit a nearly linear temporal evolution, converging as particle bonding increases. This proximity between energy levels increases the likelihood of transitions within the system. However, the CD protocol effectively addresses this issue by maintaining significant energy level separation throughout the entire evolution domain, particularly noticeable for the ground state energy (E_0) and the first excited level (E_1). At bound distances of 2.0 Å and 2.5 Å, these two energy levels remain notably distant, especially in the early stages of the evolution. This can be especially checked between the ground and first excited states, particularly when the system is initially prepared in the E_0 level. Figure 5(b) shows tiny imaginary eigenvalue parts (around 10^{-3}), negligible for analysis due to hermiticity. Adjusting ω_C can reinforce this (see Methods section). In the adiabatic case, since the initial and final Hamiltonians are real and due to the scalar nature of $\lambda(t)$, the energy levels are also purely real, as seen in figure 5(b).

Taking into account the outputs given by the PINN method, it is possible to simulate the evolution using *Qutip* [57] Master equation in order to ensure that a minimum state of energy at the end of the process is reached. This simulation is done using the complete Hamiltonian throughout the time evolution constructed using (3), along with the expected energy of the final Hamiltonian in time, $\boldsymbol{\mathcal{H}}_{\text{final}}$. Different inter-atomic bounding distances are considered. The results are detailed in table 1.

The expected value of the Hamiltonian operator can be understood as the energy of the system, which we desire to be as close as possible to the ground state. For the presented use case, the relative errors obtained (expressed in percentage) are negligible. This is a good indicator of the method's performance since, as stated

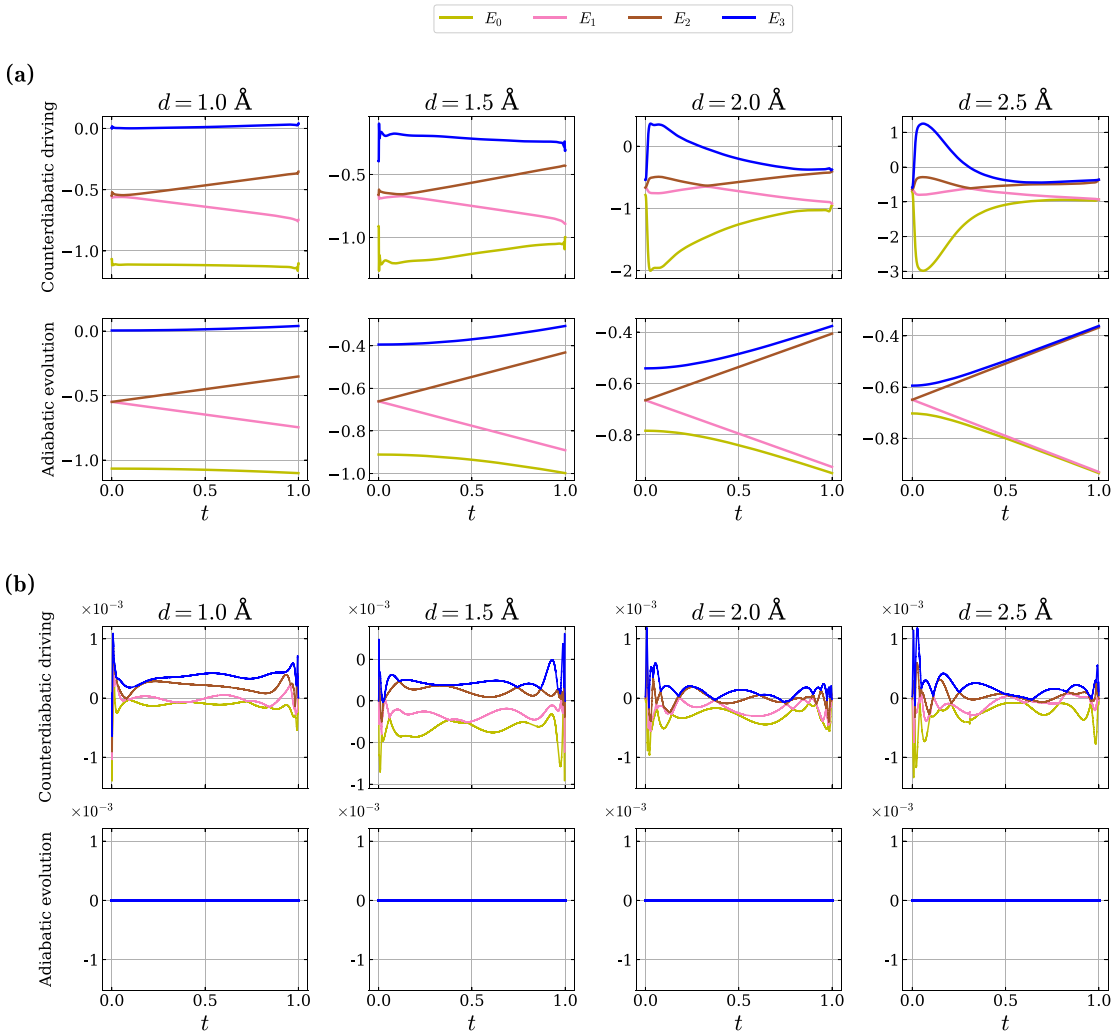

Figure 5. Temporal evolution of the real and imaginary components of the energy levels are shown in (a) and (b), respectively, both for a fully adiabatic evolution and the CD driving protocol. These correspond to the eigenvalues of the $\boldsymbol{\mathcal{H}}(t)$ operator describing the H_2 use case molecule. Diverse values of the interparticle bond distance (d) are used. Notice the difference in scales for the real components within bond distances. The energy unit used is the Hartree energy ($E_h = 4.359\,744\,722\,2071(85) \times 10^{18}$ J $= 27.211\,386\,245\,988(53)$ eV).

Table 1. Real components of the final energy levels achieved by the PINN for different inter-atomic distances of the H_2 system Hamiltonian operator. In the last two columns the computed expected value of predicted Hamiltonian at $t = 1$ using *Qutip* module, computed as $E_{\text{final}} = \langle H_{\text{final}} \rangle$, and its percentage relative error with respect to the predicted ground state energy, $\epsilon = 100 \times |(\text{Re}(E_0) - E_{\text{final}})/\text{Re}(E_0)|$, are shown.

d (Å)	$\text{Re}(E_0)$	$\text{Re}(E_1)$	$\text{Re}(E_2)$	$\text{Re}(E_3)$	E_{final}	ϵ (%)
1.0	−1.1012	−0.7459	−0.3523	0.0391	−1.1011	1×10^{-4}
1.5	−0.9981	−0.8906	−0.4315	−0.3072	−0.9981	2×10^{-4}
2.0	−0.9488	−0.9243	−0.4065	−0.3769	−0.9486	5×10^{-3}
2.5	−0.9360	−0.9316	−0.3672	−0.3611	−0.9347	1.5×10^{-1}

through the text, the main purpose of CD protocols is to accelerate the process trying to avoid transitions between energy states. Consequently, our method properly simulates the evolution considering non-adiabatic terms while leading to states that are in total concordance with the original adiabatic process.

Regarding the mentioned states, apart from the energies, it is also feasible to check the eigenstates of the system at the final step in time. For illustrative purposes only, the predicted states for the use case considering a bounding distance of $d = 1.0$ Å are written in (17). It is straightforward to see that the imaginary components are negligible by several orders of magnitude and that the orthogonality arises naturally,

$$|0\rangle = \begin{pmatrix} 0.707 \\ -0.707 + \mathcal{O}(10^{-5})j \\ \mathcal{O}(10^{-6}) + \mathcal{O}(10^{-5})j \\ \mathcal{O}(10^{-6}) + \mathcal{O}(10^{-5})j \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} \mathcal{O}(10^{-6}) + \mathcal{O}(10^{-6})j \\ \mathcal{O}(10^{-5}) + \mathcal{O}(10^{-6})j \\ 0.175 + \mathcal{O}(10^{-4})j \\ 0.984 \end{pmatrix},$$

$$|2\rangle = \begin{pmatrix} 0.707 + \mathcal{O}(10^{-5})j \\ 0.707 \\ \mathcal{O}(10^{-5}) + \mathcal{O}(10^{-5})j \\ \mathcal{O}(10^{-5}) + \mathcal{O}(10^{-6})j \end{pmatrix}, \quad |3\rangle = \begin{pmatrix} \mathcal{O}(10^{-6}) + \mathcal{O}(10^{-5})j \\ \mathcal{O}(10^{-6}) + \mathcal{O}(10^{-6})j \\ 0.984 \\ -0.175 + \mathcal{O}(10^{-4})j \end{pmatrix}. \tag{17}$$

3.4. $\mathcal{A}_{\mathrm{CD}}(t)$ operator and its decomposition

The methodology allows us to directly extract the temporal evolution of the AGP components, $\mathcal{A}_{\mathrm{CD}}(t)$, as illustrated in figure 6. Notably, the real part, $\mathrm{Re}(\mathcal{A}_{\mathrm{CD}})$, exhibits complete symmetry, while its counterpart, the imaginary part, $\mathrm{Im}(\mathcal{A}_{\mathrm{CD}})$, showcases full antisymmetry. This intrinsic property arises from the natural hermiticity of the operator, a quality indirectly imposed by the condition $\mathcal{L}_{\mathrm{C}}$ in (13) in conjunction with the fact that $\mathcal{C}(t) \in \mathbb{R}^{4^{\mathrm{N_Q}}}$. Our experimental focus, however, lies also in the dynamic acquisition of coefficients $\mathcal{C}(t)$ over time. These coefficients play a pivotal role in expressing the gauge potential as a linear combination of qubit interactions, a crucial aspect for effective implementation in a real quantum circuit. This decomposition is rigorously formulated in (12), encompassing all conceivable 2-qubit combinations within our current investigation.

Given the relatively modest size of our example system, it remains practical to explore all possible combinations of the Pauli matrices, with the number of these combinations scaling with $4^{\mathrm{N_Q}}$, where $\mathrm{N_Q}$ represents the number of qubits. Nevertheless, in certain experimental scenarios, it may be unnecessary to explore all combinations and interactions, allowing for particularized adaptations of the method.

In figure 7, we present the temporal evolution of the coefficients derived for the example system as a function of the bond distance (d) between the particles. The upper row illustrates the evolution itself, while the lower row displays a bar chart presenting the specific coefficients arranged in descending order based on their average values throughout the observed time interval. This visualization enables us to identify the most significant contributions in terms of the absolute value when implementing this system in a real quantum circuit. Notably, the two coefficients that exhibit the highest values are $\mathcal{C}_{\mathrm{XY}}$ and its symmetric counterpart $\mathcal{C}_{\mathrm{YX}}$. These findings align with previous literature that employed the NC methodology [23, 42]. Our approach naturally reveals these prominent contributions, facilitating an explicit understanding of their respective orders of magnitude. Additionally, there are less prominent contributions, such as $\mathcal{C}_{\mathrm{ZZ}}$, $\mathcal{C}_{\mathrm{XX}}$, and $\mathcal{C}_{\mathrm{II}}$, which warrant consideration as well.

It is noteworthy to note that, even though this decomposition may be useful in some scenarios, the fact of forcing the network to predict the coefficients is done particularly with illustrative purposes for this use case. In general, obtaining the full evolution of $\boldsymbol{A}_{\mathrm{CD}}(t)$ is more than enough, and its decomposition can also be obtained as a post-processing by a second network that would only rely on the loss term (13) in case of being necessary. From the other side, it is true that by predicting the gauge potential and its decomposition at once, the PINN becomes a multi-tasking model thereby converging faster and better to the solution considered optimal. However, the number of possible elements in the basis scales as $4^{\mathrm{N_Q}}$. Consequently, for a more realistic case where the number of qubits increases, the amount of outputs for the PINN will as well, thus increasing notably the training iteration time during which the gradients with respect to the losses need to be computed. This may lead to scalability problems when the number of qubits increases.

In these scenarios, our methodology may not be applied from such an academical point of view since considering all the possible interactions of the qubits within the system is, in general, not the most suitable approximation. From an experimental point of view it becomes impractical to account for all possible contributions in the decomposition of $\mathcal{A}_{\mathrm{CD}}(t)$, since the number of possible operations that can be done on the qubits is usually very limited. Consequently, a trade-off between purely theoretical outcomes, exemplified herein, and those of particular experimental significance may exist.

3.5. Scalability

In our study, we explored a 2-qubit system for simulating the H_2 molecule using the the STO-3G basis, highlighting the adaptability of our approach to different bases and qubit quantities. We focused on matrix dimensions, with $\mathcal{H}, \mathcal{A}_{\mathrm{CD}}, \mathcal{H}_{\mathrm{initial}}, \mathcal{H}_{\mathrm{final}}$ in $\mathcal{M}_{2^{\mathrm{N_Q}} \times 2^{\mathrm{N_Q}}}(\mathbb{C})$ and $\mathcal{C}(t)$ in $\mathbb{R}^{4^{\mathrm{N_Q}}}$. Our neural network architecture, formed by six layers of 30 neurons, drew from established practices in the literature, resulting in

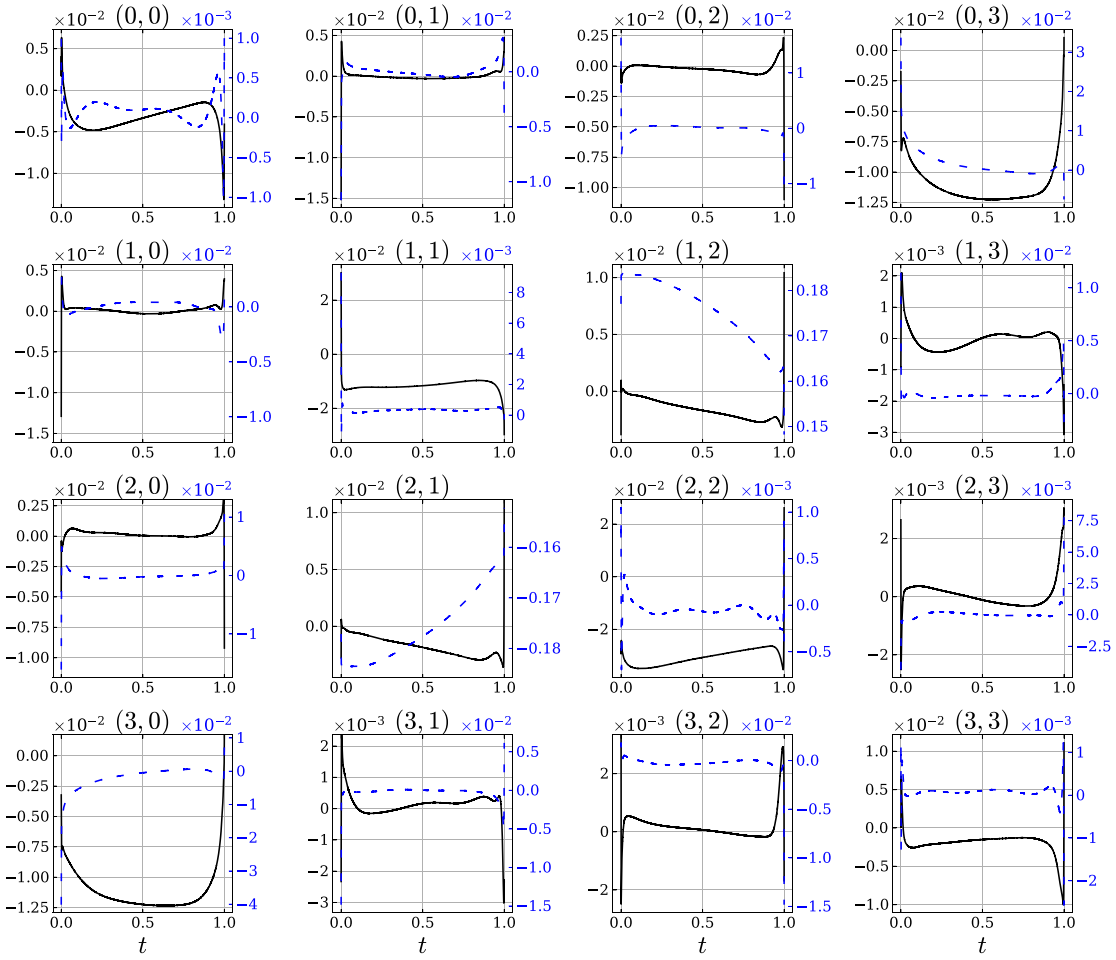

Figure 6. Time evolution of the real and imaginary components of the $\mathcal{A}_{\mathrm{CD}}(t)$ operator for the H_2 molecule use case using a bond distance of $d = 1.0$ Å, represented respectively in black and blue colors using two different scales. It is observed that the values exhibit variations across different orders of magnitude. Notably, the natural symmetry and antisymmetry arises naturally in these components over time due to the hermeticity of the operator.

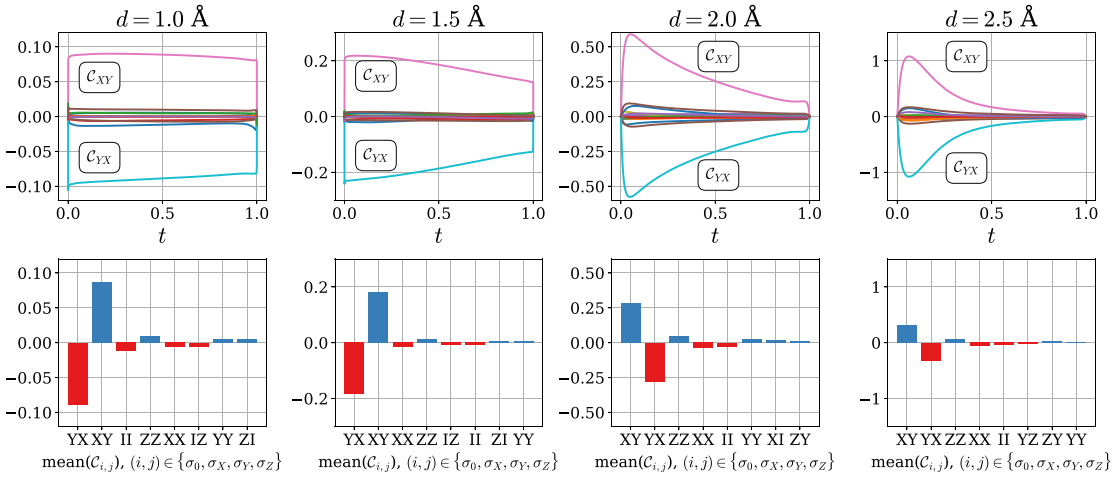

Figure 7. (a) Temporal evolutions of the coefficients $\mathcal{C}(t)$ resulting from the decomposition of the operator $\mathcal{A}_{\mathrm{CD}}(t)$ (12) applied to the H_2 of our system. (b) Bar chart illustrating the average values of each coefficient, arranged in descending order of magnitude. It is evident that both the coefficient $\mathcal{C}_{XY}$ and its symmetric counterpart $\mathcal{C}_{YX}$ exert the most substantial influence throughout the entire process, followed by $\mathcal{C}_{II}$, $\mathcal{C}_{ZZ}$ and $\mathcal{C}_{XX}$.

state-of-the-art outcomes [58–60]. The scalability of the architecture was acknowledged, but for our single differential equation-focused problem, it efficiently addressed the CD protocol, avoiding unnecessary computational overhead and potential convergence issues [61]. Expanding beyond H_2, we can apply this

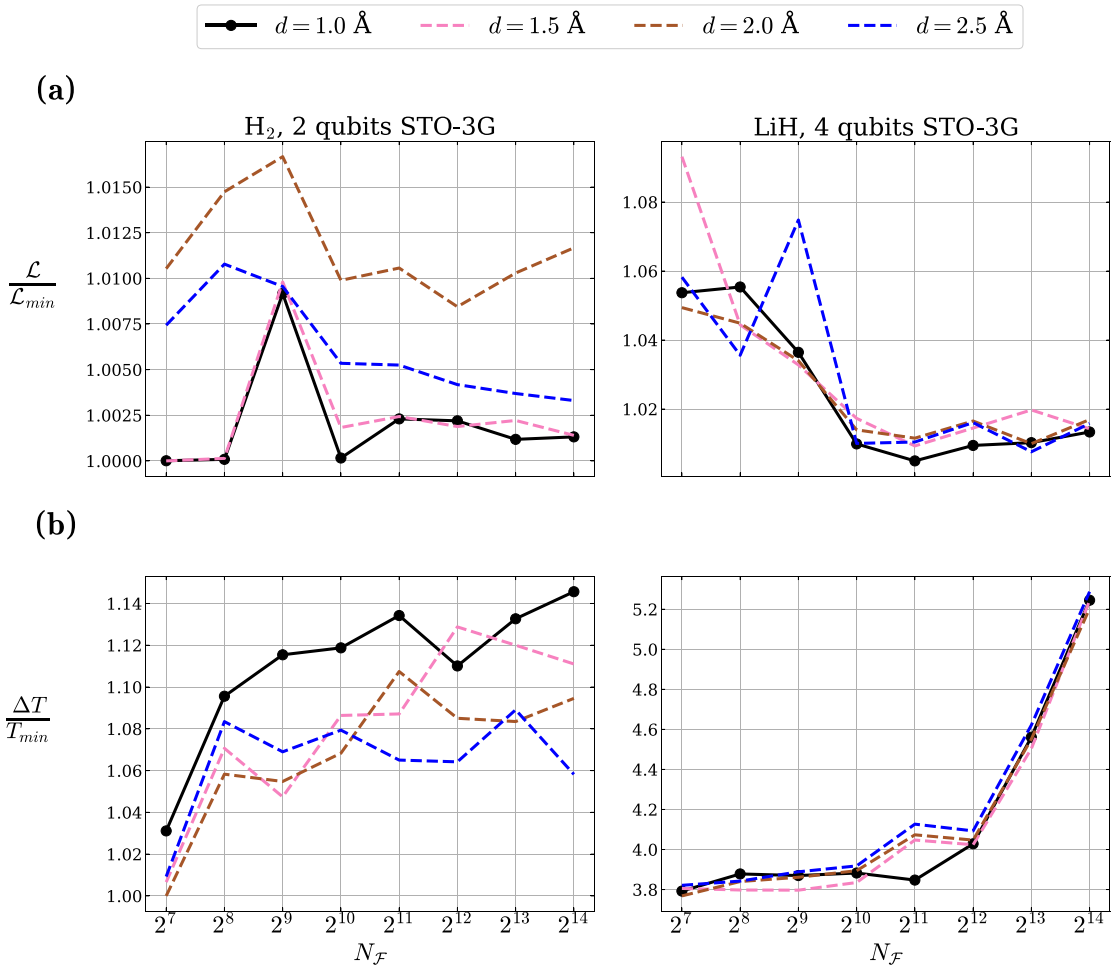

Figure 8. Scalability of the methodology for the H_2 and LiH molecules in the STO-3G basis. Various bond distances are considered, represented by distinct colors. A comparison between elapsed computing times and achieved losses with respect the minimum values ($T_{\text{min}} \approx 50$ min and $\mathcal{L}_{\text{min}} = 0.50$) are shown. All the experiments were trained for 500 000 epochs and conducted using an NVIDIA A100 40GB GPU. (a) Normalized total physical loss after completing the training process, plotted against the number of points $N_{\mathcal{F}}$ considered within the domain $t \in [0, T]$. (b) Elapsed time required to complete the entire training process, normalized to the minimum quantity.

methodology to molecules like LiH using 4 qubits, with Hamiltonian operators computed again using *Qiskit*. Scalability is an important consideration in evaluating our DL-based methodology. We emphasize the need to examine two pivotal factors: the number of qubits and the time interval points in the evaluation of the PDEs, i.e. $N_{\mathcal{F}}$, along with the GPU or hardware setup, which, for all computations, featured an NVIDIA A100 card with 40GB of available memory. Figure 8 presents a comprehensive analysis of the final physical loss for network training concerning the interval $[0, T]$. This analysis encompasses various bond distances for the H_2 and the LiH molecules in figure 8 aiding in comparisons, where in (a) and (b) the physical losses achieved and the elapsed computational times for both use cases are shown respectively.

The loss in figure 8(a) has been normalized with respect to $\mathcal{L}_{\text{min}}$, allowing for a comparison across various training sessions involving H_2 and LiH simulations. Notably, the LiH case exhibits slightly higher but comparable loss values to the H_2 case, with both using identical neural architectures of 6 layers, each having 30 neurons, and a uniform training regimen of 500 000 iterations. Even though it is possible to compare between physical losses of both use cases, one must have special care on the raw values of the initial and final Hamiltonian operators, since losses at the boundaries will be computed with respect to them. This is the main reason of LiH having bigger values for the losses in general, since these may present slightly different scales. Consequently, we aim to show the general decreasing tendency for the use cases with respect the amount of collocation points. Regarding the loss of H_2, it results to be more robust regarding the sampling leading to better performance with less data.

This consistent approach facilitates fair performance evaluations and meaningful comparisons between the simulations. On the other hand, to effectively quantify computational time during training sessions and

enable meaningful comparisons, it is essential to consider external factors like GPU selection, available CPU memory, among others. We present a comparative analysis in figure 8(b), evaluating time consumption (ΔT) normalized to the minimum value (T_{min}) for both molecules, where the CUDA Events API [62] has been employed to measure elapsed times. Increasing data points ($N_{\mathcal{F}}$) from 2^7 to 2^{14} for H_2 leads to an increase no bigger than approximately the 15% in elapsed training time, while minimal physical error for 2^7 and 2^{10} is achieved, suggesting that expanding the time domain sampling does not necessarily improve the performance of the model. With 2^{10} data points regarding the first use case, the PINN demonstrates significant capabilities in inference, extrapolation, and interpolation, without this entailing a substantial increase in computing time, thereby constituting a favorable trade-off between training time and performance.

Regarding the second use case, the LiH molecule represented using 4 qubits, there is a substantial increase in required time compared to the previous scenario. Concerning the physical loss, we can see a slight general increase for all the bond distances in comparison with the H_2 molecule, even though the loss diminishes smoothly with the amount of sampling, which is a behavior that we would expect to be important when the number of qubits increases, i.e. when the problem becomes more challenging. We see a prominent minimum of error for $N_{\mathcal{F}} = 2^{11}$, suggesting that more training iterations would be necessary for bigger samplings.

The largest disparities with the previous case arise concerning the elapsed computing time. The training for the LiH molecule takes almost a $\times 4$ factor for the lowest sampling, $N_{\mathcal{F}} = 2^7$, increasing with the sampling size specially from $N_{\mathcal{F}} = 2^{11}$ onwards. This is a good behavior since the lowest error is achieved for this size, thereby establishing 2^{10} and 2^{11} as suitable domain extents. At this point, it is necessary to take into account that our DL-based methodology extracts three outputs at the same time, encompassing the scheduling function and the $\boldsymbol{\mathcal{A}}_{\text{CD}}(t)$ operator together with its decomposition in terms of the Pauli basis. Each element of the mentioned basis corresponds to one possible operation that can be performed on the set of N_Q qubits under consideration, whose amount scales as 4^{N_Q}. Similarly, the matrix size of the operators scales as $2^{N_Q} \times 2^{N_Q}$, thus also leading to a total of 4^{N_Q} which will, in addition, depend on the physical time. This gets translated into getting a total of $1 + 2 \times 4^{N_Q}$ outputs from the neural network. For an increasing number of qubits, this number would represent an enormous amount of qubit interactions, which in reality is both impractical and unfeasible. Therefore, the limitations in scalability of the method rely more on the number of operations rather than on the amount of qubits.

From an experimental standpoint, the amount of possible operations is often very limited and usually involves scenarios where the qubits are arranged in a certain specific manner only allowing a subset of the totality of the interactions (e.g. [45, 63] where a certain ansatz for $\boldsymbol{\mathcal{A}}_{\text{CD}}$ is considered, or [64], where different graphs illustrating some connectivities are depicted). Consequently, trying to solve the entirety of the CD protocol in such situations would be impractical and unrealistic. The approach presented in this manuscript tries to tackle the shortcut-to-adiabaticity problem by dealing with everything at once, hence solving the situation from a more academic and methodological perspective. Nevertheless, given a certain previous insight about the arrangement of the qubits that are available in the laboratory, it is possible to determine a small subset of $\boldsymbol{\mathcal{C}}(t)$ coefficients. Once the experimental setup has been considered, our methodology can be adapted by properly modifying the loss function and solving the problem at hand.

4. Conclusions

This manuscript aims to enhance the utility of a deep learning (DL)-based PINNs technology to address the CD driving problem in analog quantum computation, departing from traditional numerical methods. Our proposed approach is a versatile, problem-agnostic solution encompassing the determination of CD terms and temporal parametrization, empirically validating the adiabatic theory. It offers a unified framework for understanding the underlying physics, including the theoretical Pauli matrix decomposition concerning the possible set of operations that can be done on the system. Moreover, CD driving enhances the separation between the ground state (E_0) and the first excited level (E_1) over a significant temporal range, as discussed in section 3.3. The proposed approach showcases its value when tackling problems from a methodological and academical standpoint. The amount of different allowed qubit interactions within an experiment can be a limiting aspect for a DL-based software to deal with all the information at once regarding the shortcut-to-adiabaticity scenario.

The methodology developed in this context has primarily been tested on systems with two and four qubits, as discussed in section 3.5. The model allows obtaining the scheduling function $\lambda(t)$ and the gauge potential $\boldsymbol{\mathcal{A}}_{\text{CD}}(t)$ together with its decomposition in terms of the Pauli and identity operators. These outputs are given using as input just the sampled time from a specific initial time to a particular ending point, being it usually defined as $t \in [0, T]$. Moreover, the partial differential equation (PDE) governing the system evolution from the initial to the final Hamiltonian operators (3) together with the initial and final

restrictions in time ((7) and (8)), they serve as beforehand information to train the network. These inductive biases are helped by the physical insights provided regarding the loss function for the collocation points, by means of the recovery of adiabatic theory (11), the minimization of the action (10) and the decomposition of the gauge potential (13).

In particular, regarding the former of the previous elements, by introducing that term we are forcing the network to give a solution where the CD velocity of the process, i.e. $\mathrm{d}\lambda/\mathrm{d}t$, is minimum. However, as it has been discussed, the network converges to $\lambda(t) \approx t$, thereby leading to a compromise that gets reflected by undoing the temporal parameterization initially presented through the scheduling function. This is in accordance with previous studies within the literature such as [51], where the so-called fast-quasi adiabatic dynamics (FAQUAD) were introduced. More recent studies apply these methodologies to solve experimental situations (check [65, 66] and references therein). In that sense, the DL-based approach presented throughout this article could also be seen as a FAQUAD procedure.

Our methodology tries to illustrate how a DL-based technique is able to solve the CD problem within the STA framework by giving all the needed information, in particular the $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ operator for all the time steps. This operator is predicted along with its decomposition considering all the possible operations and interactions the qubits may present. Forcing the network to learn both at once helps the method to achieve better performance and faster convergence since it becomes a multi-task model [67, 68]. However, for particular experimental scenarios it would be feasible to adapt and tailor the loss function regarding the collocation points, particularly if the way the qubits are arranged is known, i.e. the possible interactions between them are established beforehand. In these cases the set of coefficients $\boldsymbol{\mathcal{C}}(t)$ gets reduced to a small subset thereby allowing to construct the total Hamiltonian $\boldsymbol{\mathcal{H}}(t)$ straightforwardly.

These aspects link directly with the scalability aspect of the methodology. In figure 8 the elapsed computing times are depicted for the H_2 and LiH use cases. The quantity of outputs for the network goes as $1 + 2 \times 4^{\mathrm{N_Q}}$ with $\mathrm{N_Q}$ being the number of qubits (one scalar variable plus $4^{\mathrm{N_Q}}$ components of the gauge potential and $4^{\mathrm{N_Q}}$ coefficients for the decomposition). Consequently, it is sensible to expect learning to slow down or even reach a limit as this number increases. It is then interesting to resolve systems of relatively small sizes (two or four qubits) for illustrative purposes and to convey to the reader that DL-based approaches are a realistic alternative to previous techniques for solving the CD protocols. However, when moving to more relatistic scenarios, it becomes unfeasible to consider all the possible interactions between qubits, being thus necessary to adapt this methodology to a certain given ansatz. The usage of methods like NC [41] may help to shed some light on estimating which contributions may be the main ones and, once fixed, the resolution of the PINN can be tailored.

The proposed methodology introduces the physical information primarily through the loss function. Future improvements to this method may involve imposing temporal causal prerequisites [69] where PINNs have demonstrated to converge faster and better by gradually unlocking the network's visibility over time steps, thereby inducing a more pronounced decrease in the physical loss across the temporal domain. In addition, extra inductive biases concerning the neural architecture could also be introduced, specially in situations where symmetries arise. Reaching losses of approximately order 10^{-4} in figure 2 is a good indicator of the model's convergence. Nevertheless, there are opportunities for improvement and optimization within the methodology, encompassing aspects such as inductive biases, architectural enhancements, and even code refinement. With this aim in mind, our model tries to lay the groundwork for potential future studies.

In conclusion, our work has shown the suitability of PINN-based methods in optimizing AQC CD protocols. Despite significant progress, there are ample opportunities for improvement, particularly in exploring the impact of these results on digitized counterdiabatic quantum computing algorithms [70, 71]. These questions represent open avenues for future research, guiding our efforts to evolve and expand upon them.

Data availability statement

The only numerical data used in this study pertains to the initial and final Hamiltonian operators obtained through Qiskit and are stored as a compact HDF5 file, but not publicly available since these can be obtained freely using the aforementioned module. The corresponding author can provide access to the needed data upon a reasonable request. The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

The authors gratefully acknowledge the computer resources at Artemisa, funded by the European Union ERDF and Comunitat Valenciana as well as the technical support provided by the Instituto de Fisica

Corpuscular, IFIC (CSIC-UV). The authors express their sincere appreciation for the thoughtful attention provided by the Kipu Quantum team. A F S and J D M G are partially supported by the agreement funded by the European Union, between the Valencian Ministry of Innovation, Universities, Science and Digital Society, and the network of research centers in Artificial Intelligence (Valencian Foundation valgrAI). It has also been funded by the Valencian Government grant with Reference Number CIAICO/2021/184; the Spanish Ministry of Economic Affairs and Digital Transformation through the QUANTUM ENIA project call—Quantum Spain project, and the European Union through the Recovery, Transformation and Resilience Plan—NextGenerationEU within the framework of the Digital Spain 2025 Agenda.

Code availability statement

The underlying code for this study [and training/validation datasets] is not publicly available but may be made accessible to qualified researchers on reasonable request from the corresponding author.

Conflict of interest

The authors declare that a patent application, in which they are the inventors, was submitted in August 2023. There is no other conflict of interest to declare.

Appendix A. Theory behind PINNs

The approach employed in physics-informed neural networks (PINNs) methodologies [72] involves leveraging neural networks as tools for approximating functions and solving physical problems by fitting an underlying system of partial differential equations (PDEs). These neural networks incorporate physical knowledge through the usage of inductive biases. The biases are manifested from various aspects of the method, encompassing the design of the neural architecture, the formulation of appropriate cost functions (losses), among other characteristics that aim to ensure the convergence of the model. The algorithm leverages the automated differentiation capabilities found in current frameworks to construct differential equations based on the variables obtained as output. A minimization process is subsequently employed, guided by a suitable loss function thereby updating the trainable parameters of the architecture during the training process.

From a general perspective, the PINN methodology involves a neural network that presents a general architecture that can be also modified by incorporating some insights from a physical background. In our case, the structure consists of a set of multiple fully connected layers, with a total number of layers denoted as K including both the input and output layers. Each layer consists of a variable number of neurons, represented by N_k, characterized by a common output activation function written as σ_k. Even though more generality can be considered, for our method the same activation will remain consistent across all neurons within a layer. Let $\boldsymbol{W}^{(k)} \in \mathbb{R}^{\mathrm{N}_k \times \mathrm{N}_{k-1}}$ be the weight matrix connecting the $(k-t)$th and the kth layers, and $\boldsymbol{b}^{(k)} \in \mathbb{R}^{\mathrm{N}_k}$ be the bias vector for the kth one. It is feasible to define the output of that layer as $\boldsymbol{\mathcal{U}}_{\Theta}^{(k)}$ in (A.1) below,

$$\boldsymbol{\mathcal{U}}_{\Theta}^{(k)} = \sigma_k \left(\boldsymbol{W}^{(k)} \boldsymbol{\mathcal{U}}_{\Theta}^{(k-1)} + \boldsymbol{b}^{(k)} \right). \tag{A.1}$$

In terms of trainable parameters of the PINN, these are typically limited to those found in a conventional network, commonly denoted as $\Theta := \{\boldsymbol{W}^{(k)}, \boldsymbol{b}^{(k)}\}_{1 \leqslant k \leqslant K}$. Regarding the activation functions, in particular those employed at the output layer, they need to be suitable for the requirements of the specific physical problem at hand. For instance, certain physical variables may exhibit limitations within a defined range of values. In the context of our specific problem as defined in the appendix B, the sole independent variable is time. Considering (A.1) and the time interval $t \in [0, T]$, it is feasible to express the output ensemble of the architecture as (A.2),

$$\boldsymbol{\mathcal{U}}(t) \approx \boldsymbol{\mathcal{U}}_{\Theta}(t) = \sigma_K \left(\boldsymbol{\mathcal{U}}_{\Theta}^{(k)} \circ \sigma_{K-1} \circ \boldsymbol{\mathcal{U}}_{\Theta}^{(k-1)} \circ \ldots \circ \sigma_1 \circ \boldsymbol{\mathcal{U}}_{\Theta}^{(1)} \right)(t). \tag{A.2}$$

This expression showcases the mathematical representation of the approximation proposed by the neural network in a PINN methodology. It aims to closely resemble the actual physical solution following the training steps. In this context, the symbol '$\circ$' represents the composition operator. Recent research in the field asserts that PINNs enhance their performance by incorporating dynamic activation functions that vary with the learning process and are distinct for each neuron [73]. However, our focus lies primarily on establishing the definition of physical inductive biases within the network.

In a broader context, $\boldsymbol{\mathcal{U}}(t,\boldsymbol{x})$ would represent physical variables given by the PINN, constituting a system of PDEs over the spatial domain Ω and the time interval $[0,T]$ in general, yielding an operator as written in (A.3). Even though the initial time step may not be necessarily zero, we will be fixing it to that value for the sake of simplicity. As stated through the manuscript, in general the PINN is forced to fulfill some conditions according to beforehand available knowledge such as the initial information in time (A.4), some possible boundary conditions (A.5), or even additional information about experimental measurements. In addition, the constraint for the minimization of the underlying PDE is also necessary, all of this information being embedded together into the resulting loss function. The neural architecture outputs essential physical variables $\boldsymbol{\mathcal{U}}(t,\boldsymbol{x};\Theta)$, with Θ representing all trainable parameters of the network, updated during training to closely align with ground truth values, $\boldsymbol{\mathcal{U}}(t,\boldsymbol{x};\Theta) := \boldsymbol{\mathcal{U}}_\Theta(t,\boldsymbol{x}) \approx \boldsymbol{\mathcal{U}}(t,\boldsymbol{x})$,

$$\mathcal{F}\left(t,\boldsymbol{x};\frac{\partial\boldsymbol{\mathcal{U}}}{\partial t},\frac{\partial^2\boldsymbol{\mathcal{U}}}{\partial t^2},\ldots;\frac{\partial\boldsymbol{\mathcal{U}}}{\partial x_1},\ldots,\frac{\partial\boldsymbol{\mathcal{U}}}{\partial x_D};\frac{\partial^2\boldsymbol{\mathcal{U}}}{\partial x_1\partial x_1},\ldots,\frac{\partial^2\boldsymbol{\mathcal{U}}}{\partial x_1\partial x_D},\ldots\right)=0, \tag{A.3}$$

$$\mathcal{IC}\,(t,\boldsymbol{x})=0, \qquad (t,\boldsymbol{x})\in\{0\}\times\Omega. \tag{A.4}$$

$$\mathcal{B}(t,\boldsymbol{x})=0, \qquad (t,\boldsymbol{x})\in(0,T]\times\partial\Omega. \tag{A.5}$$

For instance, the constraints in equations (A.4) and (A.5) can be written as cost functions by employing a suitable difference measurement metric such as MSE or similar, representing the initial conditions in time and boundary restrictions within the spatial physical domain, respectively. In the specific scenario involving the counterdiabatic evolution of an N_Q-qubit system, we are bound by both initial and final temporal conditions, in addition to the internal collocation points, while the input consists only of the time variable (t). The expressions of these cost functions together with $\mathcal{L}_\mathcal{F}$ can be outlined as follows,

$$\mathcal{L}_{\mathcal{IC}} := \frac{1}{N_{\mathcal{IC}}}\sum_{\{0\}}\left|\boldsymbol{\mathcal{U}}_\Theta(0)-\boldsymbol{\mathcal{U}}(0)\right|^2. \tag{A.6}$$

$$\mathcal{L}_{\mathcal{FC}} := \frac{1}{N_{\mathcal{FC}}}\sum_{\{T\}}\left|\boldsymbol{\mathcal{U}}_\Theta(T)-\boldsymbol{\mathcal{U}}(T)\right|^2. \tag{A.7}$$

$$\mathcal{L}_{\mathcal{F}} := \frac{1}{N_{\mathcal{F}}}\sum_{(0,T)}\underbrace{\sum_{k}\left|\mathcal{R}_k(t;\Theta)\right|^2}_{|\mathcal{F}|^2}. \tag{A.8}$$

In equations (A.6)–(A.8), the set $(N_{\mathcal{IC}}, N_{\mathcal{FC}}, N_{\mathcal{F}})$ represents the amount of sampled points for each respective domain (initial/final conditions and collocation points). Additionally, $\mathcal{R}_k$ denotes the fulfillment of specific physical constraints imposed at the level of the PDEs. Through the usage of the PINN methodology, it becomes feasible to smoothly enforce the constraints by adjusting the underlying equations governing the behavior of the system, known as 'soft enforcement' in previous literature. However, an alternative approach leading state-of-the-art results for several problems denoted as 'hard enforcement' proposes forcing the network to fulfill the physical conditions from the onset by mathematically modifying its output [74]. This technique necessitates incorporating several constraints, but it entails establishing a specific dependence on the problem at hand. Other researchers achieve a certain level of independence concerning the set of weights in (5) through the application of the Augmented Lagrangian method [75] which involves the update of the multipliers during training in accordance with the degree of violation for each respective condition.

Appendix B. Physical background

A time-dependent Hamiltonian, as the one defined in (2), can generally lead to modifications in the quantum states that it governs. However, when these changes are sufficiently minute, analytically tractable and under controlled conditions, the adiabatic theorem ensures that the system, assuming non-degeneracy in energy, will preserve its proximity to the initial energy state throughout the temporal evolution. Therefore, it is feasible to formulate a Hamiltonian operator, denoted as $\boldsymbol{\mathcal{H}}(t)$, that exhibits a direct correlaction with $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$ and accurately reproduces the temporal progression of its eigenstates. It possesses transition amplitudes between energy levels that are precisely zero. This operator is designed to satisfy the Schrödinger equation in (B.1),

$$i\hbar\,\partial_t|\psi_n(t)\rangle = \boldsymbol{\mathcal{H}}(t)\,|\psi_n(t)\rangle. \tag{B.1}$$

In this expression, $|\psi_n\rangle$ can be defined in terms of the corresponding eigenstate of the operator in (2), $|n(t)\rangle$. These states represent each one of the n vectors driven by $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$ in a certain particular system. Through rigorous derivation, it is posible to obtain an analytical representation of the operator $\boldsymbol{\mathcal{H}}(t)$. For a detailed deducement, the reader is referred to [38, 76, 77]. This operator, as written in (B.2), governs the evolution of energy states within the framework of $\boldsymbol{\mathcal{H}}_{\mathrm{AD}}(t)$, ensuring the absence of transitions between them,

$$\boldsymbol{\mathcal{H}}(t) = \underbrace{\sum_n |n\rangle E_n \langle n|}_{\boldsymbol{\mathcal{H}}_{\mathrm{AD}}} + \underbrace{i\hbar \sum_n \left(|\partial_t n\rangle\langle n| - \langle n|\partial_t n\rangle |n\rangle\langle n|\right)}_{\boldsymbol{\mathcal{H}}_{\mathrm{CD}}}. \tag{B.2}$$

The second term, $\boldsymbol{\mathcal{H}}_{\mathrm{CD}}(t)$, can be interpreted as the Hamiltonian operator responsible for capturing the counterdiabatic effects during the evolution. These effects facilitate the acceleration of the system dynamics by introducing additional accessible degrees of freedom. This allows to reach the same results as the entire adiabatic evolution, eliminating any possible transition between eigenstates. These frameworks are well-known as counterdiabatic protocols, and expedite the adiabatic processes by introducing novel terms that offset any excitation that may arise during system acceleration. It is then possible to extend the deduction by introducing a set of time-dependent external parameters, denoted as $\boldsymbol{\lambda}(t)$, upon which all the operators will be dependent [40, 78]. Consequently, the temporal derivatives of the $|n\rangle$ states in (B.2) can be expressed as (B.3) below,

$$|\partial_t n\rangle = \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} |\boldsymbol{\nabla}_{\boldsymbol{\lambda}} n\rangle. \tag{B.3}$$

This expression allows us to rewrite the counterdiabatic operator $\boldsymbol{\mathcal{H}}_{\mathrm{CD}}(t)$ as follows in (B.4) in terms of the parameterization $\boldsymbol{\lambda}(t)$,

$$\boldsymbol{\mathcal{H}}_{\mathrm{CD}} = \frac{\mathrm{d}\boldsymbol{\lambda}}{\mathrm{d}t} \boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t). \tag{B.4}$$

This $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ is denoted as 'gauge potential' in the literature due to its nature. Moreover, it is straightforward to write this operator as in (B.5), where we have considered $\hbar = 1$ due to the use of natural units,

$$\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t) := i \sum_n \left(|\boldsymbol{\nabla}_{\lambda} n\rangle\langle n| - \langle n|\boldsymbol{\nabla}_{\lambda} n\rangle |n\rangle\langle n|\right). \tag{B.5}$$

Regarding the temporal parameterization, it is posible for it to encompasse a collection of values $\boldsymbol{\lambda} := (\lambda_1, \ldots, \lambda_N)$. However, in order to align with the latest literature in the field, we will confine ourselves to a single scalar function, $\lambda(t)$, which usually receives the name of scheduling function. This function carries out the counterdiabatic driving by terms of its temporal derivative. Indeed, it is evident that in the limit of $|\mathrm{d}\lambda/\mathrm{d}t| \to 0$, the counderdiabatic operator would not be present thereby the total Hamiltonian in (1) being simplified to the adiabatic operator, as anticipated during its construction. When extraoplating this understanding to contemporary digitized versions of algorithms within the field of quantum circuits, it is customary to initiate the process with a Hamiltonian that presents a ground state of energy that can be readily prepared in practical settings, denoted as $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$ through the text. This operator undergoes a gradual transformation over time governed by the $\lambda(t)$ function and its temporal derivative, eventually converging to the target Hamiltonian at the final time step, written as $\boldsymbol{\mathcal{H}}_{\mathrm{final}}$.

Regarding the gauge potential, $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}(t)$ (also written in the literature as $\boldsymbol{\mathcal{A}}_{\lambda}$), is defined as the adiabatic gauge potential (AGP). Obviously, this operator should fulfill the hermiticity condition in order to be interpreted as a physical observable. Our PINNs methodology will force the system to minimize the action $\mathcal{S}$ in (B.6) for the Hilbert-Schmidt operator $\boldsymbol{\mathcal{G}}_{\lambda}$. The Euler–Lagrange condition used as a loss function for the network, written in (4), is straightforward to derive by writing the adiabatic Hamiltonian in the moving frame as $\tilde{\boldsymbol{\mathcal{H}}}_{\mathrm{AD}} = U^{\dagger} \boldsymbol{\mathcal{H}}_{\mathrm{AD}} U$ and taking into account that $\left[i\partial_{\lambda} \tilde{\boldsymbol{\mathcal{H}}}_{\mathrm{AD}}, \tilde{\boldsymbol{\mathcal{H}}}_{\mathrm{AD}}\right] = 0$ (for a more detailed derivation the reader is addressed to [44] Methods section),

$$\mathcal{S} = \mathrm{Tr}\left[\boldsymbol{\mathcal{G}}_{\lambda}^2\right], \qquad \boldsymbol{\mathcal{G}}_{\lambda}(\boldsymbol{\mathcal{A}}_{\mathrm{CD}}) = \partial_{\lambda} \boldsymbol{\mathcal{H}}_{\mathrm{AD}} + i[\boldsymbol{\mathcal{A}}_{\mathrm{CD}}, \boldsymbol{\mathcal{H}}_{\mathrm{AD}}]. \tag{B.6}$$

Consequently, the Euler–Lagrange equation establishes a connection between the aforementioned external parameter $\lambda(t)$ and the instantaneous eigenstates. Finding the exact CD terms is not easy without

spectral information of the system. Therefore, these are usually approximated using the nested commutators (NC) approach [39] or through variational circuits [43],

$$\langle m|\boldsymbol{\mathcal{A}}_{\mathrm{CD}}|n\rangle = i\langle m|\partial_\lambda n\rangle = -i\frac{\langle m|\partial_\lambda \boldsymbol{\mathcal{H}}_{\mathrm{AD}}|n\rangle}{E_m - E_n}. \tag{B.7}$$

The main objective of CD driving processes is to obtain a representation for the gauge potential, $\boldsymbol{\mathcal{A}}_{\mathrm{CD}}$, by properly controlling the velocity of the overall process, $\mathrm{d}\lambda/\mathrm{d}t$, and preventing non-desired transitions between energy states. However, this approach becomes computationally unfeasible for larger systems due to the need for Hamiltonian diagonalization at every time step. Equation (B.7) highlights the complexity involved in determining this operator (interested reader is addressed to [38] for a detailed derivation). The need of exact diagonalization over time and the potential divergence of matrix elements due the energy differences $E_m - E_n$ leads to great complexity from a general perspective. To tackle these challenges, different methods such as the nested commutators approach offer an approximate way of computing the potential by some sort of a series expansion, as described in (B.8),

$$\boldsymbol{\mathcal{A}}_{\mathrm{CD}}^{(l)} = i\sum_{k=1}^{l}\alpha_k \underbrace{[\boldsymbol{\mathcal{H}}_{\mathrm{AD}},[\boldsymbol{\mathcal{H}}_{\mathrm{AD}},\ldots[\boldsymbol{\mathcal{H}}_{\mathrm{AD}},\partial_\lambda}_{2k-1}\boldsymbol{\mathcal{H}}_{\mathrm{AD}}]]]. \tag{B.8}$$

Appendix C. Quantum circuit design for the H_2 ground state problem

The H_2 molecule utilized as an illustrative use case for this manuscript can be described using a 2-qubit system through the full configuration interaction (FCI) Hamiltonian of (C.2). The set of Pauli matrices and the identity, $\{\boldsymbol{\mathcal{I}},\sigma_{\mathrm{X}},\sigma_{\mathrm{Y}},\sigma_{\mathrm{Z}}\}\in\mathcal{M}_{2\times 2}(\mathbb{C})$, have been used to define the following operators by means of the Pauli basis expansion. In practical terms, and regarding the use case that is proposed, the $\boldsymbol{\mathcal{H}}_{\mathrm{FCI}}$ operator can be understood as the Hamiltonian describing the system at the endpoint of the temporal evolution, i.e. $\boldsymbol{\mathcal{H}}_{\mathrm{final}}$ through the text. Similarly, the $\boldsymbol{\mathcal{H}}_{\mathrm{HF}}$ defined in (C.1) can be seen as the $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$ referring to the initial operator describing the states at the beginning of the evolution. Here, the symbol '$\otimes$' represents the Kronecker product of matrices, and the superscripts '$^{(1)}$', '$^{(2)}$' indicate the qubit to which each operator is addressed. The value of the coefficients accompanying each term, i.e. the sets $\{c_i\}$ and $\{p_i\}$, will vary with the bound distances considered through the numerical results, but the main structure defined below will remain consistent. Consequently, the $\boldsymbol{\mathcal{H}}_{\mathrm{final}}$ ($\boldsymbol{\mathcal{H}}_{\mathrm{FCI}}$) will have non-diagonal terms while $\boldsymbol{\mathcal{H}}_{\mathrm{initial}}$ ($\boldsymbol{\mathcal{H}}_{\mathrm{HF}}$) will not,

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{\mathrm{HF}} = {} & p_0\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)} + p_1\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\sigma_{\mathrm{Z}}^{(2)} + p_2\,\sigma_{\mathrm{Z}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)} \\ & + p_3\,\sigma_{\mathrm{Z}}^{(1)}\otimes\sigma_{\mathrm{Z}}^{(2)} + p_4\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)}.\end{aligned} \tag{C.1}$$

$$\begin{aligned}\boldsymbol{\mathcal{H}}_{\mathrm{FCI}} = {} & c_0\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)} + c_1\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\sigma_{\mathrm{Z}}^{(2)} + c_2\,\sigma_{\mathrm{Z}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)} \\ & + c_3\,\sigma_{\mathrm{Z}}^{(1)}\otimes\sigma_{\mathrm{Z}}^{(2)} + c_4\,\sigma_{\mathrm{X}}^{(1)}\otimes\sigma_{\mathrm{X}}^{(2)} + c_5\,\boldsymbol{\mathcal{I}}^{(1)}\otimes\boldsymbol{\mathcal{I}}^{(2)}.\end{aligned} \tag{C.2}$$

As detailed through the text, the system starts the evolution from a Hamiltonian that should be easy to prepare in experimental terms. For this H_2 molecule use case, the Hartree–Fock (HF) approach is used to approximate the ground state of the system using a single Slater determinant. Regarding the numerical computational part, the *Qiskit* module is employed. In a similar way that has been presented for this molecule, the initial and final Hamiltonian operators can also be derived for systems with a larger number of qubits.

ORCID iDs

Antonio Ferrer-Sánchez https://orcid.org/0000-0002-2305-5261
José D Martín-Guerrero https://orcid.org/0000-0001-9378-0285

References

[1] Adcock J *et al* 2015 Advances in quantum machine learning (arXiv:1512.02900)
[2] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
[3] Dunjko V, Taylor J M and Briegel H J 2016 Quantum-enhanced machine learning *Phys. Rev. Lett.* **117** 130501
[4] Wang R, Hernani-Morales C, Martín-Guerrero J D, Solano E and Albarrán-Arriagada F 2021 Quantum pattern recognition in photonic circuits *Quantum Sci. Technol.* **7** 015010
[5] Feynman R P 1982 Simulating physics with computers *Int. J. Theor. Phys.* **21** 467–88

[6] Georgescu I M, Ashhab S and Nori F 2014 Quantum simulation *Rev. Mod. Phys.* **86** 153–85
[7] Hogg T and Portnov D 2000 Quantum optimization *Inf. Sci.* **128** 181–97
[8] Niroula P, Shaydulin R, Yalovetzky R, Minssen P, Herman D, Hu S and Pistoia M 2022 Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer *Sci. Rep.* **12** 17171
[9] Prajapati J B *et al* 2023 *Quantum Machine Learning in Prediction of Breast Cancer* (Springer) pp 351–82
[10] Tang Y, Yan J, Hu G, Zhang B and Zhou J 2022 Recent progress and perspectives on quantum computing for finance *Serv. Oriented Comput. Appl.* **16** 227–9
[11] Weinberg S J, Sanches F, Ide T, Kamiya K and Correll R 2023 Supply chain logistics with quantum and classical annealing algorithms *Sci. Rep.* **13** 4770
[12] Ambainis A and Kokainis M 2017 Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games *Proc. 49th Annual ACM SIGACT Symp. on Theory of Computing* (STOC 2017) (Association for Computing Machinery) pp 989–1002
[13] Lloyd S and Weedbrook C 2018 Quantum generative adversarial learning *Phys. Rev. Lett.* **121** 040502
[14] Dey S, Bhattacharyya S and Maulik U 2016 New quantum inspired meta-heuristic techniques for multi-level colour image thresholding *Appl. Soft Comput.* **46** 677–702
[15] Arute F *et al* 2019 Quantum supremacy using a programmable superconducting processor *Nature* **574** 505–10
[16] Brooks M 2019 Beyond quantum supremacy: the hunt for useful quantum computers *Nature* **574** 19–21
[17] Peruzzo A *et al* 2014 A variational eigenvalue solver on a photonic quantum processor *Nat. Commun.* **5** 4213
[18] Grover L K 1997 Quantum mechanics helps in searching for a needle in a haystack *Phys. Rev. Lett.* **79** 325–8
[19] Aïmeur E, Brassard G and Gambs S 2013 Quantum speed-up for unsupervised learning *Mach. Learn.* **90** 261–87
[20] Wiebe N, Kapoor A and Svore K M 2015 Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning *Quantum Inf. Comput.* **15** 316–56
[21] Fel'dman E B and Zenchuk A I 2023 Nearest-neighbor approximation in one-excitation state evolution along spin-1/2 chain governed by XX-Hamiltonian *Phys. Lett.* A **457** 128572
[22] Farhi E, Goldstone J, Gutmann S and Zhou L 2022 The quantum approximate optimization algorithm and the Sherrington-Kirkpatrick model at infinite size *Quantum* **6** 759
[23] Chandarana P, Hegade N N, Paul K, Albarrán-Arriagada F, Solano E, del Campo A and Chen X 2022 Digitized-counterdiabatic quantum approximate optimization algorithm *Phys. Rev. Res.* **4** 013141
[24] Pagano G *et al* 2020 Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator *Proc. Natl Acad. Sci. USA* **117** 25396–401
[25] Lyu C, Xu X, Yung M-H and Bayat A 2023 Symmetry enhanced variational quantum spin eigensolver *Quantum* **7** 899
[26] Torta P, Mbeng G B, Baldassi C, Zecchina R and Santoro G E 2023 Quantum approximate optimization algorithm applied to the binary perceptron *Phys. Rev.* B **107** 094202
[27] Santoro G E and Tosatti E 2006 Optimization using quantum mechanics: quantum annealing through adiabatic evolution *J. Phys. A: Math. Gen.* **39** R393
[28] Albash T and Lidar D A 2018 Adiabatic quantum computation *Rev. Mod. Phys.* **90** 015002
[29] Farhi E, Goldstone J, Gutmann S, Lapan J, Lundgren A and Preda D 2001 A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem *Science* **292** 472–5
[30] Bapst V, Foini L, Krzakala F, Semerjian G and Zamponi F 2013 The quantum adiabatic algorithm applied to random optimization problems: the quantum spin glass perspective *Phys. Rep.* **523** 127–205
[31] Liu J, Spedalieri F, Yao K-T, Potok T, Schuman C, Young S, Patton R, Rose G and Chamka G 2018 Adiabatic quantum computation applied to deep learning networks *Entropy* **20** 380
[32] Born M and Fock V 1928 Beweis des adiabatensatzes *Z. Phys.* **51** 165–80
[33] Avron J E, Raveh A and Zur B 1988 Adiabatic quantum transport in multiply connected systems *Rev. Mod. Phys.* **60** 873–915
[34] Guéry-Odelin D *et al* 2019 Shortcuts to adiabaticity: concepts, methods and applications *Rev. Mod. Phys.* **91** 045001
[35] Takahashi K 2019 Hamiltonian engineering for adiabatic quantum computation: lessons from shortcuts to adiabaticity *J. Phys. Soc. Japan* **88** 061002
[36] Chandarana P, Vieites P S, Hegade N N, Solano E, Ban Y and Chen X 2023 Meta-learning digitized-counterdiabatic quantum optimization *Quantum Sci. Technol.* **8** 045007
[37] Romero S V, Chen X, Platero G and Ban Y 2024 Optimizing edge-state transfer in a Su-Schrieffer-Heeger chain via hybrid analog-digital strategies *Phys. Rev. Appl.* **21** 034033
[38] Berry M V 2009 Transitionless quantum driving *J. Phys. A: Math. Theor.* **42** 365303
[39] Hegade N N, Chandarana P, Paul K, Chen X, Albarrán-Arriagada F and Solano E 2022 Portfolio optimization with digitized counterdiabatic quantum algorithms *Phys. Rev. Res.* **4** 043204
[40] Jarzynski C 2013 Generating shortcuts to adiabaticity in quantum and classical dynamics *Phys. Rev.* A **88** 040101
[41] Claeys P W, Pandey M, Sels D and Polkovnikov A 2019 Floquet-engineering counterdiabatic protocols in quantum many-body systems *Phys. Rev. Lett.* **123** 090602
[42] Hegade N N, Chen X and Solano E 2022 Digitized counterdiabatic quantum optimization *Phys. Rev. Res.* **4** L042030
[43] Sun D, Chandarana P, Xin Z-H and Chen X 2022 Optimizing counterdiabaticity by variational quantum circuits *Phil. Trans. R. Soc.* A **380** 20210282
[44] Sels D and Polkovnikov A 2017 Minimizing irreversible losses in quantum systems by local counterdiabatic driving *Proc. Natl Acad. Sci. USA* **114** E3909–16
[45] Čepaitė I, Polkovnikov A, Daley A J and Duncan C W 2023 Counterdiabatic optimized local driving *PRX Quantum* **4** 010312
[46] Qiskit Contributors 2023 Qiskit: an open-source framework for quantum computing *Zenodo* https://doi.org/10.5281/zenodo.2573505
[47] Yanofsky N S and Mannucci M A 2008 *Quantum Computing for Computer Scientists* (Cambridge University Press)
[48] Rieffel E G and Polak W H 2011 *Quantum Computing: A Gentle Introduction* (MIT Press)
[49] Baydin A G, Pearlmutter B A, Radul A A and Siskind J M 2018 Automatic differentiation in machine learning: a survey *J. Mach. Learn. Res.* **18** 153 (available at: https://jmlr.org/papers/v18/17-468.html)
[50] Nielsen M A and Chuang I L 2010 *Quantum Computation And Quantum Information: 10th Anniversary Edition* (Cambridge University Press)
[51] Martínez-Garaot S, Ruschhaupt A, Gillet J, Busch T and Muga J G 2015 Fast quasiadiabatic dynamics *Phys. Rev.* A **92** 043406

[52] Wu C, Zhu M, Tan Q, Kartha Y and Lu L 2023 A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks *Comput. Methods Appl. Mech. Eng.* **403** 115671
[53] Sobol' I M 1967 On the distribution of points in a cube and the approximate evaluation of integrals *USSR Comput. Math. Math. Phys.* **7** 86–112
[54] Glorot X and Bengio Y 2010 Understanding the difficulty of training deep feedforward neural networks *Proc. 13th Int. Conf. on Artificial Intelligence and Statistics* (*Proc. Machine Learning Research* vol 9) ed Y W Teh and M Titterington (PMLR) pp 249–56
[55] Paszke A *et al* 2023 PyTorch (available at: https://pytorch.org)
[56] Kingma D P and Ba J 2014 Adam: a method for stochastic optimization (arXiv:1412.6980)
[57] Johansson J R, Nation P D and Nori F 2013 Qutip 2: a Python framework for the dynamics of open quantum systems *Comput. Phys. Commun.* **184** 1234–40
[58] Mao Z, Jagtap A D and Karniadakis G E 2020 Physics-informed neural networks for high-speed flows *Comput. Methods Appl. Mech. Eng.* **360** 112789
[59] Ferrer-Sánchez A, Martín-Guerrero J D, de Austri-Bazan R R, Torres-Forné A and Font J A 2024 Gradient-annihilated PINNs for solving Riemann problems: application to relativistic hydrodynamics *Comput. Methods Appl. Mech. Eng.* **424** 116906
[60] Yu J, Lu L, Meng X and Karniadakis G E 2022 Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems *Comput. Methods Appl. Mech. Eng.* **393** 114823
[61] Pantidis P, Eldababy H, Tagle C M and Mobasher M E 2023 Error convergence and engineering-guided hyperparameter search of PINNs: towards optimized I-FENN performance *Comput. Methods Appl. Mech. Eng.* **414** 116160
[62] NVIDIA Corporation 2024 *CUDA Events API* (NVIDIA Corporation) (available at: https://docs.nvidia.com/cuda/cuda-runtime-api/group__CUDART__EVENT.html)
[63] Barone F P, Kiss O, Grossi M, Vallecorsa S and Mandarino A 2024 Counterdiabatic optimized driving in quantum phase sensitive models *New J. Phys.* **26** 033031
[64] Barraza N, Barrios G A, Montalban I, Solano E and Albarrán-Arriagada F 2023 Variational coherent quantum annealing (arXiv:2310.02248)
[65] Martínez-Garaot S, Muga J G and Tseng S-Y 2017 Shortcuts to adiabaticity in optical waveguides using fast quasiadiabatic dynamics *Opt. Express* **25** 159–67
[66] Chung H-C and Tseng S-Y 2018 Ultrashort and broadband silicon polarization splitter-rotator using fast quasiadiabatic dynamics *Opt. Express* **26** 9655–65
[67] Baxter J 2000 A model of inductive bias learning *J. Artif. Intell. Res.* **12** 149–98
[68] Caruana R 1997 Multitask learning *Mach. Learn.* **28** 41–75
[69] Wang S, Sankaran S and Perdikaris P 2024 Respecting causality for training physics-informed neural networks *Comput. Methods Appl. Mech. Eng.* **421** 116813
[70] Hegade N N, Paul K, Ding Y, Sanz M, Albarrán-Arriagada F, Solano E and Chen X 2021 Shortcuts to adiabaticity in digitized adiabatic quantum computing *Phys. Rev. Appl.* **15** 024038
[71] Cadavid A G, Montalban I, Dalal A, Solano E and Hegade N N 2023 Efficient DCQO algorithm within the impulse regime for portfolio optimization (arXiv:2308.15475)
[72] Raissi M, Perdikaris P and Karniadakis G E 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations *J. Comput. Phys.* **378** 686–707
[73] Jagtap A D, Kawaguchi K and Karniadakis G E 2020 Adaptive activation functions accelerate convergence in deep and physics-informed neural networks *J. Comput. Phys.* **404** 109136
[74] Lu L, Pestourie R, Yao W, Wang Z, Verdugo F and Johnson S G 2021 Physics-informed neural networks with hard constraints for inverse design *SIAM J. Sci. Comput.* **43** B1105–32
[75] Son H, Cho S W and Hwang H J 2023 Enhanced physics-informed neural networks with augmented Lagrangian relaxation method (AL-PINNs) *Neurocomputing* **548** 126424
[76] Kato T 1950 On the adiabatic theorem of quantum mechanics *J. Phys. Soc. Japan* **5** 435
[77] Nenciu G 1980 On the adiabatic theorem of quantum mechanics *J. Phys. A: Math. Gen.* **13** L15
[78] Kolodrubetz M, Sels D, Mehta P and Polkovnikov A 2017 Geometry and non-adiabatic response in quantum and classical systems *Phys. Rep.* **697** 1–87