


PHYSICS-INFORMED NEURAL NETWORKS FOR AN OPTIMAL COUNTERDIABATIC QUANTUM COMPUTATION

A PREPRINT

 **Antonio Ferrer-Sánchez**^{1,2}
Antonio.Ferrer-Sanchez@uv.es
Corresponding author

 **Carlos Flores-Garrigos**¹
Carlos.Flores-Garrigos@uv.es

 **Carlos Hernani-Morales**^{1,3}
Carlos.Hernani@uv.es

 **José J. Orquín-Marqués**¹
jose.j.orquin@uv.es


 **Narendra N. Hegade**⁴

 **Alejandro Gomez Cadavid**⁴

 **Iraitz Montalban**⁴

 **Enrique Solano**⁴

 **Yolanda Vives-Gilbert**¹
yolanda.vives@uv.es

 **José D. Martín-Guerrero**^{1,2,3}
jose.d.martin@uv.es

September 11, 2023

ABSTRACT

We introduce a novel methodology that leverages the strength of Physics-Informed Neural Networks (PINNs) to address the counterdiabatic (CD) protocol in the optimization of quantum circuits comprised of systems with N_Q qubits. The primary objective is to utilize physics-inspired deep learning techniques to accurately solve the time evolution of the different physical observables within the quantum system. To accomplish this objective, we embed the necessary physical information into an underlying neural network to effectively tackle the problem. In particular, we impose the hermiticity condition on all physical observables and make use of the principle of least action, guaranteeing the acquisition of the most appropriate counterdiabatic terms based on the underlying physics. The proposed approach offers a dependable alternative to address the CD driving problem, free from the constraints typically encountered in previous methodologies relying on classical numerical approximations. Our method provides a general framework to obtain optimal results from the physical observables relevant to the problem, including the external parameterization in time known as scheduling function, the gauge potential or operator involving the non-adiabatic terms, as well as the temporal evolution of the energy levels of the system, among others. The main applications of this methodology have been the H_2 and LiH molecules, represented by a 2-qubit and 4-qubit systems employing the STO-3G basis. The presented results demonstrate the successful derivation of a desirable decomposition for the non-adiabatic terms, achieved through a linear combination utilizing Pauli operators. This attribute confers significant advantages to its practical implementation within quantum computing algorithms.

Keywords Neural networks, counterdiabatic driving, Quantum computing, Quantum information, PINNs

¹IDAL, Electronic Engineering Department, ETSE-UV, University of Valencia, Avda. Universitat s/n, 46100 Burjassot, Valencia, Spain.

²Valencian Graduate School and Research Network of Artificial Intelligence (ValgrAI), Spain.

³Quantum Spain, 46100 Burjassot, Valencia, Spain.

⁴Kipu Quantum, Greifswalderstrasse 226, 10405 Berlin, Germany.

1 Introduction

Quantum computing has emerged as a dynamic and vibrant domain of research within the scientific community, primarily driven by notable achievements and advancements in applied quantum machine learning [1–4], quantum simulation [5, 6], and optimization of circuits and systems [7, 8]. Optimization problems have garnered particular attention, given their pervasive presence in various domains, including medicine [9], economics [10], logistics [11], and numerous others [12–14]. Classical approaches to solving these problems from an industrial perspective often face challenges in terms of efficiency and speed, thereby motivating the exploration of quantum computing as a promising alternative. The escalating interest in these methods can be attributed primarily to recent experimental advancements. This surge in interest is particularly noticeable from an industrial and commercial perspective. Consequently, there is growing anticipation that both conventional computers and quantum computing, in general, could yield significant advantages, eventually achieving a state known as “quantum supremacy” [15]. This potential advantage and progress have, in turn, spurred developments in various scientific domains, wherein contemporary quantum computers serve as proof of concept. However, it is essential to underscore that the applicability of these quantum algorithms warrants extensive research and investigation, particularly in the current state of quantum computing, which is commonly referred to as the Noisy Intermediate-Scale Quantum (NISQ) era [16] whose defining characteristic is the utilization of quantum processors with capacities of up to 1000 qubits.

Hybrid classical-quantum algorithms leverage NISQ devices while offloading a portion of their computational workload onto classical devices, offering considerable potential for practical applications in the field of quantum computing. A prominent example worth highlighting is the Variational Quantum Eigensolver (VQE) [17]. The primary objective of VQE is to determine the lowest energy quantum state through hybrid optimization, utilizing a designated Hamiltonian operator in conjunction with variational quantum circuits. The realm of quantum machine learning also falls under the purview of these algorithms, seeking to adapt classical algorithms to their quantum counterparts to enhance and expedite computations by capitalizing on the principles of quantum superposition, entanglement, and interference. Within this domain, one can identify supervised classification algorithms like binary classification and those based on Grover’s search algorithm [18]. Notably, Grover’s algorithm has demonstrated quadratic acceleration in solving problems such as k -medians [19] or k -nearest neighbors [20, 21]. On the other hand, an alternative methodology that has significantly progressed in the literature of this field and has laid the foundation for numerous studies is the Quantum Approximate Optimization Algorithm (QAOA) [22, 23]. This approach presents a valuable alternative for tackling combinatorial optimization problems using shallow quantum circuits through classical optimization of the associated parameters. In recent literature, substantial endeavors have been dedicated to employing these methodologies to solve the ground-state challenges of physical systems [24–26], reflecting the ongoing efforts to enhance and adapt these techniques for broader applications in quantum optimization.

In recent years, significant attention and interest have been directed towards the development of adiabatic quantum optimization (AQO) methodologies for confronting optimization problems [27, 28] with direct practical implementations in the branches of physics and chemistry [29–31]. These algorithms begin by initializing a quantum system in the ground state of a known Hamiltonian. The system’s Hamiltonian is then slowly evolved into one that represents the problem to be solved, with its ground state encoding the solution. Leveraging the adiabatic theorem [32, 33], it becomes feasible to ensure that the quantum system remains in its instantaneous state of lowest energy, provided the evolution of the Hamiltonian is carried out in a sufficiently slow and gradual manner and within a sufficiently extended period of time. Nevertheless, implementing slow adiabatic evolution at the experimental level is typically not feasible, necessitating the development of methodologies that accelerate these processes. In pursuit of this objective, recent scientific literature puts forth various approaches based on Shortcuts To Adiabaticity (STA) [34, 35]. These methodologies encompass diverse techniques, including fast-forward methods [36, 37], invariant-based inverse engineering [38, 39], and counterdiabatic (CD) protocols [40, 41]. Despite the noticeable progress in the first two methodologies, this study primarily centers around the CD protocols. These techniques are specifically designed to speed up the adiabatic evolution process from an initial Hamiltonian to a final Hamiltonian. This is achieved by incorporating non-adiabatic terms following Equation (1), which effectively nullify the transition amplitudes between any energy eigenstate of the original Hamiltonian [42]. Consequently, the quantum system undergoes an accelerated adiabatic evolution in practical applications. The resulting Hamiltonian by including the CD term is given by

$$\mathcal{H}(t) := \mathcal{H}_{\text{AD}}(t) + \mathcal{H}_{\text{CD}}(t). \quad (1)$$

The operator designated as $\mathcal{H}_{\text{AD}}(t)$ in (1) will be tailored to facilitate the preparation of its ground energy state at the initial time of the evolution. Nevertheless, the main challenge of the CD protocols lies in the accurate and plausible determination of the operator encompassing the non-adiabatic terms of the process, denoted here as $\mathcal{H}_{\text{CD}}(t)$. In general, the computation and acquisition of this operator are exceedingly intricate tasks, particularly when dealing with many-body quantum systems [43]. As a customary approach in related literature, a time-dependent external

parameterization, denoted as $\lambda(t)$, is introduced to which the operators [44] are dependent (explained in detail in Section 2.2). Efforts have been directed towards a method for the approximate determination of this operator, leading to significant progress, such as the development of the Nested Commutator (NC) methodology [45]. This advancement has led to recent contributions, exemplified by [23, 46]. Within this framework, the computation of these terms is simplified into a series of commutators involving $\mathcal{H}_{AD}(t)$ and its derivatives concerning the mentioned external parameterization $\lambda(t)$. As a result, the non-adiabatic terms of these protocols are obtained approximately through an expansion in orders, where the complexity of obtaining them rises accordingly with the number of particles in the system and the order considered in the expansion. Nevertheless, these methodologies may exhibit problem-dependent characteristics, as their escalating complexity in non-trivial physical scenarios might necessitate the adoption of alternative perspectives to approach the issue at hand.

In a distinct domain, within the rapid progression of the computer science field, the realm of Deep Learning (DL) has achieved prominence as a highly potent instrument for constructing diverse models, owing to its direct applicability in domains such as image processing [47], natural language generation and processing [48], time series prediction and classification [49], among a plethora of other possibilities. Present-day technologies encompass Recurrent Neural Networks (RNNs) [50], Long Short-Term Memory architectures (LSTMs) [51], the well-known transformers [52], sparse and submanifolds convolutional networks [53] utilized for images with limited information load, among other advanced techniques. The Physics-Informed Neural Networks (PINNs) methodology has emerged as a highly intriguing DL application within the realm of physical sciences since its first appearances in the literature [54, 55]. This approach aims to address specific problems by employing neural networks as powerful universal approximators for systems of Partial Differential Equations (PDEs) that govern the physics describing the evolution. Due to their remarkable potential as numerical solvers, PINNs have garnered considerable attention and established themselves as a viable alternative to conventional numerical solving algorithms. Extensive efforts have been undertaken in diverse branches of physics to apply this methodology and its adaptations. These fields encompass classical hydrodynamics [56], relativistic hydrodynamics [57], electrodynamics [58], chemistry [59], and many others. PINNs demonstrate their utility wherever differential equations are employed to describe the underlying physics of a given scenario. Their ability to unravel complex physical phenomena and offer numerical solutions has positioned them as promising tools for tackling intricate problems across various scientific domains. Consequently, the motivation to employ this methodology for addressing the challenge of CD protocols in quantum systems arises organically. The investigation of potential applications of PINNs in quantum systems and circuits becomes a natural course of study.

In this paper, we present an innovative approach for designing the counterdiabatic terms in CD protocols. Our proposed method lays on the utilization of PINNs, thereby representing a direct application of DL. We introduce a PINN-based methodology without supplementary alterations, enabling it to effectively resolve the underlying physics of the problem. Through the neural network, we obtain both the counterdiabatic terms and the temporal parameterization $\lambda(t)$, as expounded in Section 2.2. Additionally, we directly explore the experimental feasibility by decomposing the non-adiabatic operator into tensor products of the set of Pauli and identity matrices. This approach offers a comprehensive and direct means to address the experimental applicability of the method.

The rest of the paper is structured as follows: Section 2 provides an introduction to the foundational theoretical framework concerning the operation of baseline PINNs. It further presents a comprehensive exposition of the theoretical framework under consideration, encompassing CD protocols and the specific problem that serves as the motivation for our research. Additionally, a thorough literature review of prior work in this domain is presented. Section 3 delves into a meticulous presentation of the adapted PINN methodology, particularized to address our specific case, while taking into account all pertinent physical factors that the network must conform to. In Section 4, we present notable outcomes obtained through our methodology and juxtapose them with previous findings in the field. Furthermore, we conduct comparisons and explore the scalability of the proposed methodology. Finally, Section 5 serves as a concluding segment, summarizing the principal conclusions drawn from our research and offering insights into potential avenues for future investigations.

2 General Concepts

2.1 Physics-Informed Neural Networks

The fundamental approach employed in PINNs methodologies [55] involves leveraging neural networks as powerful tools for approximating functions and solving physical problems by fitting sets of differential equations, known as PDEs. PINNs derive their name from the fact that they incorporate physical knowledge through the incorporation of inductive biases. These biases are manifested in various aspects of the methodology, including the design of the underlying neural network architecture, the formulation of appropriate cost functions (losses), and other characteristics that aim to ensure or enhance the convergence of the neural model. The underlying algorithm of these networks leverages the

automated differentiation capabilities found in contemporary frameworks [60] to construct differential equations based on the output variables obtained from the network. These variables are essential for computing the specific problem at hand. By performing calculations, a minimization process is subsequently employed, guided by a designated loss function, to update the trainable parameters of the architecture. Consequently, this adjustment aligns the network with the requirements of the physical framework.

Taking a broad perspective while maintaining generality, let us denote by $\mathbf{U} := \mathbf{U}(t, \mathbf{x})$ a collection of physical variables that serve as the output of the neural network. These variables, along with their derivatives, are the components of a system of PDEs defined within a domain of interest Ω over a specific time interval $[0, T]$. Consequently, it is possible to write the following definition:

$$\mathcal{F} \left(t, \mathbf{x}; \frac{\partial \mathbf{U}}{\partial t}, \frac{\partial^2 \mathbf{U}}{\partial t^2}, \dots; \frac{\partial \mathbf{U}}{\partial x_1}, \dots, \frac{\partial \mathbf{U}}{\partial x_D}; \frac{\partial^2 \mathbf{U}}{\partial x_1 \partial x_1}, \dots, \frac{\partial^2 \mathbf{U}}{\partial x_1 \partial x_D}, \dots \right) = 0, \quad \mathbf{x} = (x_1, \dots, x_D) \in \Omega, \quad t \in [0, T], \quad (2)$$

where D corresponds to the spatial dimension of the problem and $\Omega \subset \mathbb{R}^D$. The operator \mathcal{F} defined in Equation (2) can be conceptualized as the comprehensive collection of physical constraints inherent to the system, which must be fulfilled in order to satisfy the underlying PDEs. It is worth noting that, in addition to these constraints, supplementary limitations can be established, such as the initial conditions that dictate the evolution of the system, or potential boundary conditions that may influence the behavior of the system at the spatial boundaries of the domain, namely,

$$\mathcal{IC}(t, \mathbf{x}) = 0, \quad (t, \mathbf{x}) \in \{0\} \times \Omega. \quad (3)$$

$$\mathcal{B}(t, \mathbf{x}) = 0, \quad (t, \mathbf{x}) \in (0, T] \times \partial\Omega. \quad (4)$$

In addition to the aforementioned conditions, other factors can be taken into consideration, such as imposing additional constraints on the final time (final conditions), or at specific points of significant physical significance within the spatiotemporal framework. Furthermore, if there are actual experimental measurements available for a subset of the domain, they can also be incorporated. Consequently, each of these physical conditions represents a segment of a priori knowledge regarding the physical scenario that can be integrated into the cost function as separate terms (referred to as ‘‘soft enforcement’’), as denoted with \mathcal{L}_i in Equation (5):

$$\mathcal{L} := \omega_{\mathcal{F}} \mathcal{L}_{\mathcal{F}} + \sum_i \omega_i \mathcal{L}_i, \quad (5)$$

where $\mathcal{L}_{\mathcal{F}}$ corresponds to the metric pertaining to the underlying system of PDEs, while the collection $(\omega_{\mathcal{F}}, \omega_i, \dots)$ represents the weights assigned to each term within the mixture. The neural architecture employed in this methodology yields a set of essential physical variables, denoted as $\mathbf{U}(t, \mathbf{x}; \Theta) := \mathbf{U}_{\Theta}(t, \mathbf{x})$, where Θ encompasses all the trainable parameters of the network that are updated during the training process. Consequently, the output aims to closely align with the corresponding real-world values:

$$\mathbf{U}_{\Theta}(t, \mathbf{x}) \approx \mathbf{U}(t, \mathbf{x}).$$

The constraints expressed in (3) and (4) can be transformed into cost functions by employing a suitable difference measurement metric, such as *Mean Squared Error* (MSE) or similar approaches. The determination of these cost functions together with $\mathcal{L}_{\mathcal{F}}$ can be outlined as follows.

$$\mathcal{L}_{\mathcal{IC}} := \frac{1}{N_{\mathcal{IC}}} \sum_{\{0\} \times \Omega} |\mathbf{U}_{\Theta}(0, \mathbf{x}) - \mathbf{U}(0, \mathbf{x})|^2, \quad (6)$$

$$\mathcal{L}_{\mathcal{B}} := \frac{1}{N_{\mathcal{B}}} \sum_{(0, T] \times \partial\Omega} |\mathbf{U}_{\Theta}(t, \mathbf{x} \in \partial\Omega) - \mathbf{U}(t, \mathbf{x} \in \partial\Omega)|^2, \quad (7)$$

$$\mathcal{L}_{\mathcal{F}} := \frac{1}{N_{\mathcal{F}}} \sum_{(0, T] \times \Omega} \underbrace{\sum_k |\mathcal{R}_k(t, \mathbf{x}; \Theta)|^2}_{|\mathcal{F}|^2}. \quad (8)$$

Here, the set (N_{IC}, N_B, N_F) represents the number of points in the sample for each respective domain considered. Additionally, \mathcal{R}_k denotes the fulfillment of specific physical constraints imposed at the level of the PDE. Through the utilization of the fundamental methodology employed in PINN models, it becomes feasible to smoothly enforce the imposed constraints by adjusting the PDEs associated with the given problem. Nonetheless, alternative approaches, such as the ‘‘hard enforcement’’ method [61], propose compelling the neural network to enforce predetermined constraints from the onset of training by modifying the output of the network. 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 $(\omega_{\mathcal{R}}, \omega_i, \dots)$ in (5) through the application of the Augmented Lagrangian method [62]. This technique involves updating multipliers with corresponding names during training in accordance with the degree of violation for each respective condition.

2.2 Physical background

Combinatorial and optimization problems are of great interest in many industry and research fields [23, 46]. 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 [63, 64]. 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 [43, 46]. 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, $\mathcal{H}_{AD}(t)$, which is characterized by its energy eigenvalues, $E_n(t)$, and corresponding eigenstates, $|n(t)\rangle$, as determined by:

$$\mathcal{H}_{AD}(t) |n(t)\rangle = E_n(t) |n(t)\rangle. \quad (9)$$

A time-dependent Hamiltonian, as the one defined in (9), generally could lead to modifications in the quantum states it governs. Nevertheless, when these changes are sufficiently minute, analytically tractable, and under controlled conditions, the adiabatic theorem [32] ensures that the system, assuming non-degeneracy in energy, will preserve its proximity to the initial energy state throughout its temporal evolution. Considering the aforementioned perspective, it is consistently feasible to formulate a Hamiltonian operator, denoted as $\mathcal{H}(t)$, that exhibits a direct correlation with $\mathcal{H}_{AD}(t)$ and accurately reproduces the temporal progression of its eigenstates. In other words, it possesses transition amplitudes between energy levels that are precisely zero [42]. This operator will be designed to satisfy the following Schrödinger equation:

$$i\hbar \partial_t |\psi_n(t)\rangle = \mathcal{H}(t) |\psi_n(t)\rangle,$$

where $|\psi_n(t)\rangle$ can be defined in terms of the corresponding eigenstate of the operator in (9), $|n(t)\rangle$. These states represent each one of the n states driven by $\mathcal{H}_{AD}(t)$ in a certain particular system. Through rigorous derivation and computation, one can obtain a plausible analytical representation for the operator $\mathcal{H}(t)$. For a detailed deduction, the interested reader is encouraged to refer to the works of [42, 65, 66]. This operator, defined in (10), effectively governs the evolution of energy states within the framework of $\mathcal{H}_{AD}(t)$, ensuring the absence of transitions between them.

$$\mathcal{H}(t) = \underbrace{\sum_n |n\rangle E_n \langle n|}_{\mathcal{H}_{AD}} + i\hbar \underbrace{\sum_n (|\partial_t n\rangle \langle n| - \langle n| \partial_t n\rangle |n\rangle \langle n|)}_{\mathcal{H}_{CD}} := \mathcal{H}_{AD}(t) + \mathcal{H}_{CD}(t). \quad (10)$$

The Hamiltonian operator $\mathcal{H}_{CD}(t)$ corresponding to the second term in (10), can be interpreted as the operator responsible for capturing the counterdiabatic effects during system evolution. These effects facilitate the acceleration of the underlying system dynamics by introducing additional accessible degrees of freedom allowing to reach the same results as the entire adiabatic evolution of the system and eliminating any possible transition between eigenstates [64, 67], as previously stated. These frameworks, which are the well-known as counterdiabatic Protocols, effectively expedite the adiabatic processes by introducing novel terms that precisely offset any excitations that may arise during system acceleration. These have been recently developed as part of the STA methods. With this in mind, it is feasible to extend the theoretical deduction by incorporating a set of time-dependent external parameters, denoted as $\lambda(t)$, upon which all our operators will be dependent [44]. By doing so, when retrieving Equation (10), it follows that the temporal derivatives of the $|n\rangle$ states can be written as follows,

$$|\partial_t n\rangle = \frac{d\lambda}{dt} |\nabla_{\lambda} n\rangle. \quad (11)$$

Equation (11) allows us to redefine the operator $\mathcal{H}(t)$ as

$$\mathcal{H}(t) := \mathcal{H}_{\text{AD}}(t) + \underbrace{\frac{d\lambda}{dt} \mathcal{A}_{\text{CD}}(t)}_{\mathcal{H}_{\text{CD}}}, \quad \mathcal{A}_{\text{CD}}(t) := i\hbar \sum_n (|\nabla_{\lambda} n\rangle \langle n| - \langle n|\nabla_{\lambda} n\rangle |n\rangle \langle n|).^1 \quad (12)$$

In general, this parameterization could encompass a collection of values $\lambda := (\lambda_1, \dots, \lambda_N)$. However, to align with the latest literature in the field, we will confine ourselves to a single scalar function. Consequently, $\lambda(t)$, which usually receives the name of *scheduling function* in the literature, carries out the counterdiabatic driving by terms of its temporal derivative. Indeed, it is evident that in the limit of zero velocity, $|\frac{d\lambda}{dt}| \rightarrow 0$, the Hamiltonian in Equation (12) simplifies to the adiabatic operator, as anticipated during its construction. When extrapolating this understanding to contemporary digitized versions of algorithms within the realm of quantum circuits [43], it is customary to initiate the process with a Hamiltonian which presents a ground state of energy that can be readily prepared in practical settings, denoted as $\mathcal{H}_{\text{initial}}$. Under adiabatic conditions, it is imperative for this operator to undergo a gradual transformation over time, eventually converging to the target Hamiltonian corresponding to the specific problem under investigation, written as $\mathcal{H}_{\text{problem}}$.

$$\mathcal{H}_{\text{AD}}(t) := (1 - \lambda(t)) \mathcal{H}_{\text{initial}} + \lambda(t) \mathcal{H}_{\text{problem}}. \quad (13)$$

In Equation (13), the scheduling function can once again be identified. Within the time interval $t \in [t_{\text{min}}, t_{\text{max}}]$, it is necessary for $\lambda(t)$ to fulfill the conditions $\lambda(t_{\text{min}}) = 0$ and $\lambda(t_{\text{max}}) = 1$ based on its functional definition, which enables the interpolation of the process from the initial Hamiltonian to the desired endpoint of the process.

On the other hand, the $\mathcal{A}_{\text{CD}}(t)$ operator (also written in the literature as \mathcal{A}_{λ}) is defined as the Adiabatic Gauge Potential. Obviously, this operator should fulfill the hermiticity condition, i.e., $\mathcal{A}_{\text{CD}} = \mathcal{A}_{\text{CD}}^{\dagger}$, in order to be interpreted as a physical observable. It is also easy to show that the potential $\mathcal{A}_{\text{CD}}(t)$ satisfies the condition (15), which is equivalent to minimizing the action \mathcal{S} defined in Equation (14) for the Hilbert-Schmidt operator \mathcal{G}_{λ} [44, 68]. Consequently, Equation (15) can be understood as the Euler-Lagrange equation resulting from the minimization of the physical action.

$$\mathcal{S} = \text{Tr} [\mathcal{G}_{\lambda}^2], \quad \mathcal{G}_{\lambda}(\mathcal{A}_{\text{CD}}) = \partial_{\lambda} \mathcal{H}_{\text{AD}} + \frac{i}{\hbar} [\mathcal{A}_{\text{CD}}, \mathcal{H}_{\text{AD}}]. \quad (14)$$

$$\left[i\hbar \frac{\partial \mathcal{H}_{\text{AD}}}{\partial \lambda} - [\mathcal{A}_{\text{CD}}, \mathcal{H}_{\text{AD}}], \mathcal{H}_{\text{AD}} \right] = 0.^2 \quad (15)$$

This term should establish a connection between the aforementioned external parameter $\lambda(t)$ and instantaneous eigenstates. Finding this exact CD terms is not easy without spectral information of the system and they are usually approximated using the NC approach [23, 43, 46] or through Variational Circuits [63]. This lead to a set of possible local CD terms and different techniques that have been developed to determine which of them are the most adequate for the particular problem. Nonetheless, these methods even though they are of great interest and may constitute the state-of-the-art approaches, they are still approximations. Consequently, there could be other relevant terms and aspects that are not considered within these methodologies.

2.3 Quantum circuit design for the H_2 ground state problem

The main numerical application of this paper will be to find the ground state of the H_2 molecule in the STO-3G basis assuming different bond distances where, in particular, this STO-3G basis corresponds to a minimal set that uses three Gaussian functions to approximate the atomic orbitals [69]. This can be described with the 2-qubit Full Configuration Interaction (FCI) Hamiltonian in Equation (16), where the value of the coefficients vary with the bond distances. This Hamiltonian has been obtained using the well-know *Qiskit* module [70]. In a different domain, it is well-known that the Pauli matrices comprise a set of Hermitian and unitary matrices, namely $\{\sigma_X, \sigma_Y, \sigma_Z\} \in \mathcal{M}_{2 \times 2}(\mathbb{C})$, which are highly suitable for representing quantum operators and physical observables. These matrices, together with the identity matrix \mathcal{I} (also called σ_0), form an orthogonal basis of the Hilbert space of 2×2 Hermitian matrices. Similarly, when dealing

¹While the operators in this formulation exhibit time dependence through λ , for the sake of notation simplicity, we will denote the time dependency explicitly.

²To provide clarity and consistency, from now on we will exclusively employ Planck units, wherein the reduced Planck constant, denoted as \hbar , is established as unity (i.e., $\hbar = 1$).

with a system of N_Q qubits, the matrix space of Hermitian $\mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$ matrices can be decomposed by means of tensor products involving the aforementioned set (the interested reader is referred to [71] Chapter 2 and [72] Chapter 3). This procedure, referred to as the Pauli basis expansion, enables us to achieve a comprehensive representation of any Hermitian operator on a system comprising a certain amount of qubits. From the perspective of quantum circuits, this approach offers a structured and concise depiction of quantum operations and physical observables, thereby facilitating the analysis and manipulation of quantum systems.

$$\mathcal{H}_{\text{FCI}} = c_0 \mathcal{I}^{(1)} \otimes \mathcal{I}^{(2)} + c_1 \mathcal{I}^{(1)} \otimes \sigma_Z^{(1)} + c_2 \sigma_Z^{(1)} \otimes \mathcal{I}^{(2)} + c_3 \sigma_Z^{(1)} \otimes \sigma_Z^{(1)} + c_4 \sigma_X^{(1)} \otimes \sigma_X^{(1)} + c_5 \mathcal{I}^{(1)} \otimes \mathcal{I}^{(2)}. \quad (16)$$

This corresponds to $\mathcal{H}_{\text{problem}}$ in Equation (13). In this context, the numeric superscripts enclosed in parentheses mean that the written operator pertain to the specific qubit under consideration thereby resulting in distinct matrices associated with each one. The symbol \otimes denotes the Kronecker product. Furthermore, the first and last coefficients are written separately with the same operator since they correspond to different atoms in the molecule, even though they have the same interaction. The FCI Hamiltonian in Equation (16) can also be written in matrix form Equation (17).

$$\mathcal{H}_{\text{FCI}} = \begin{bmatrix} c_0 + c_1 + c_2 + c_3 + c_5 & 0 & 0 & c_4 \\ 0 & c_0 - c_1 + c_2 - c_3 + c_5 & c_4 & 0 \\ 0 & c_4 & c_0 + c_1 - c_2 - c_3 + c_5 & 0 \\ c_4 & 0 & 0 & c_0 - c_1 - c_2 + c_3 + c_5 \end{bmatrix}. \quad (17)$$

As detailed before, we start from a *driving* Hamiltonian that should be easy to prepare. For this particular problem, the Hartee-Fock (HF) approximation Hamiltonian (18) is used, with its matrix form written in Equation (19). It is easy to see that both the FCI and the HF Hamiltonian are real-valued since they are composed only with the identity matrix, \mathcal{I} , and the Pauli matrices σ_X and σ_Z .

$$\mathcal{H}_{\text{HF}} = p_0 \mathcal{I}^{(1)} \otimes \mathcal{I}^{(2)} + p_1 \mathcal{I}^{(1)} \otimes \sigma_Z^{(1)} + p_2 \sigma_Z^{(1)} \otimes \mathcal{I}^{(2)} + p_3 \sigma_Z^{(1)} \otimes \sigma_Z^{(1)} + p_4 \mathcal{I}^{(1)} \otimes \mathcal{I}^{(2)}. \quad (18)$$

$$\mathcal{H}_{\text{HF}} = \begin{bmatrix} p_0 + p_1 + p_2 + p_3 + p_4 & 0 & 0 & 0 \\ 0 & p_0 - p_1 + p_2 - p_3 + p_4 & 0 & 0 \\ 0 & 0 & p_0 + p_1 - p_2 - p_3 + p_4 & 0 \\ 0 & 0 & 0 & p_0 - p_1 - p_2 + p_3 + p_4 \end{bmatrix}. \quad (19)$$

The HF method approximates the ground state of the system using a single Slater determinant. The contributions of both the initial and final Hamiltonian operators in the adiabatic evolution can be determined using advanced numerical algorithms [70]. As stated before, the primary objective is to solve the adiabatic temporal evolution while considering non-adiabatic terms. The goal is to accelerate this evolution without allowing unexpected transitions between energy states. By incorporating the adiabatic operator defined in Equation (13) into the total Hamiltonian operator described in Equation (12), and considering that the initial and final Hamiltonians are represented as shown in Equations (16-19) with $\mathcal{H}_{\text{initial}} := \mathcal{H}_{\text{HF}}$ and $\mathcal{H}_{\text{final}} := \mathcal{H}_{\text{FCI}} = \mathcal{H}_{\text{problem}}$, we can establish the following PDE,

$$\mathcal{H}(t) := (1 - \lambda(t)) \mathcal{H}_{\text{HF}} + \lambda(t) \mathcal{H}_{\text{problem}} + \frac{d\lambda}{dt} \mathcal{A}_{\text{CD}}(t). \quad (20)$$

2.4 Related work

The primary objective of CD driving processes is to attain a gauge potential, $\mathcal{A}_{\text{CD}}(t)$, by expediting the adiabatic process through a temporal velocity parameter, specifically the time derivative of the scheduling function, $\frac{d\lambda}{dt}$. The aforementioned operator possesses the property that the product $\frac{d\lambda}{dt} \mathcal{A}_{\text{CD}}$ in Equation (20) fully mitigates the occurrence of non-adiabatic transitions that would otherwise manifest in a specific eigenstate $|n(t)\rangle$ under the total $\mathcal{H}(t)$, which has undergone evolution from the initial state $|n(t=0)\rangle$ of the control Hamiltonian, $\mathcal{H}_{\text{AD}}(t)$ [73]. Therefore, the aim of the related methodologies is to ensure precise compensation of the adiabatic terms in the evolutionary dynamics within the moving frame [40]. This potential, comprising the non-adiabatic compensations, represents an operator that is typically intricate to derive. While an exact numerical construction is possible for certain systems with a small number of particles, resolving it for systems with numerous bodies (a substantial number of qubits) becomes exceedingly

challenging, if not impossible, due to the requirement of diagonalizing the resulting Hamiltonian observable across the entire Hilbert space. Furthermore, \mathcal{A}_{CD} often entails the emergence of highly intricate and non-local couplings, thus limiting its implementation to specific cases [74–76]. Part of the research focused on a robust obtaining of the potential \mathcal{A}_{CD} has led to works such as [77] where fast-forward methods are presented through which it is possible to obtain an objective wave function in a shorter time. This investigation is conducted within the field of microscopic quantum mechanics, aiming to delve into the macroscopic domain primarily governed by the Schrödinger equation. These inquiries are expounded upon in references such as [36, 78]. Despite these notable progressions, there presently exists no viable approach to extend these studies to the context of many-body systems. Consequently, these methodologies are not applicable to complex problem sets. On the other hand, several recent studies such as [63] have suggested alternative approaches for addressing the aforementioned issues. The authors suggest to employ Variational Circuits (VC) in conjunction with classical optimizers to optimize the choice of CD terms, with these being treated as trainable parameters, being tested on the specific case of an Ising model of interactions with nearby neighbors and carrying out a comparison with some of the state-of-the-art QAOA methodologies [23, 79, 80]. Some assumptions are made regarding the form of the function $\lambda(t)$ thereby predefining a temporal parameterization, which may exhibit certain dependencies on the problem under investigation.

Returning to the latest approaches to derive $\mathcal{A}_{\text{CD}}(t)$ as described in the existing literature, based on its definition in Equation (12) and assuming that the parameterization is completely determined by the function $\lambda(t)$, by differentiation of Equation (9) [42], it is straightforward to arrive at

$$\langle m | \mathcal{A}_{\text{CD}} | n \rangle = i \langle m | \partial_\lambda | n \rangle = -i \frac{\langle m | \partial_\lambda \mathcal{H}_{\text{AD}} | n \rangle}{E_m - E_n}. \quad (21)$$

However, it can be readily observed from Equation (21) that determining the operator $\mathcal{A}_{\text{CD}}(t)$ can become highly intricate and computationally expensive, especially when dealing with many-body problems. This potential necessitates exact diagonalization of the operator $\mathcal{H}_{\text{AD}}(t)$ over time and, additionally, the difference in energies $E_m - E_n$ can introduce complications, leading to divergent and mathematically ill-defined matrix elements. In an attempt to address these challenges, the authors in [45] suggest employing the so-called method of NC for approximating the representation of $\mathcal{A}_{\text{CD}}(t)$.

$$\mathcal{A}_{\text{CD}}^{(l)} = i \sum_{k=1}^l \alpha_k \underbrace{[\mathcal{H}_{\text{AD}}, [\mathcal{H}_{\text{AD}}, \dots [\mathcal{H}_{\text{AD}}, \partial_\lambda \mathcal{H}_{\text{AD}}]]]}_{2k-1}. \quad (22)$$

Equation (22) presents a methodology for obtaining an approximate numerical approach for the operator \mathcal{A}_{CD} . In this expression, l represents the order of the expansion, and the set of coefficients $\{\alpha_k\}_{k=1}^l$ can be determined by minimizing the action in Equation (14) specifically tailored for the l -th order. For further details and a comprehensive demonstration, interested readers are referred to the reference, where it is demonstrated that the exact potential \mathcal{A}_{CD} is recovered in the limit as l tends towards infinity. Notwithstanding its utilization as an approximation limited to an expansion order l , this ansatz has proven to be a methodological framework for attaining cutting-edge outcomes in the field of research [23, 43, 46, 81, 82]. In the absence of an analytical solution to the problem of the CD protocol, the most compelling alternative at our disposal is to employ techniques rooted in DL, which possess the capacity to comprehend and acquire knowledge of the fundamental physics governing the problem.

3 Methodology

3.1 PINN framework

Our approach involves employing a Physics-Informed Neural Network (PINN) that incorporates a neural network structure comprising multiple fully connected dense layers, with the total number of layers denoted as K . This includes both the input and output layers. Each layer consists of a variable number of neurons, represented by N_k , and is characterized by a common output activation function denoted as σ_k . This activation function remains consistent across all neurons within a given layer after it is specified. Let $\mathbf{W}^{(k)} \in \mathbb{R}^{N_k \times N_{k-1}}$ be the weight matrix connecting the $(k-1)$ -th layer to the k -th layer, and $\mathbf{b}^{(k)} \in \mathbb{R}^{N_k}$ be the bias vector for the k -th layer. Consequently, the output of the k -th layer, denoted as $\mathbf{u}_\Theta^{(k)}$, can be expressed as the application of the activation function σ_k to the weighted sum of the output of the previous layer, followed by the addition of the biases, as denoted in Equation (23):

$$\mathbf{u}_\Theta^{(k)} = \sigma_k \left(\mathbf{W}^{(k)} \mathbf{u}_\Theta^{(k-1)} + \mathbf{b}^{(k)} \right) \quad (23)$$

In this manner, contingent upon the specific problem under consideration, additional network parameters beyond weights and biases may be taken into account. Nevertheless, if only these two sets are considered, the trainable variables would be limited to those typically found in a conventional network, denoted as $\Theta := \{\mathbf{W}^{(k)}, \mathbf{b}^{(k)}\}_{1 \leq k \leq K}$. Regarding the activation functions, particularly those employed at the output layer, they need to be tailored to suit the requirements of the specific physical problem at hand. For instance, certain physical variables may exhibit limitations within a defined range of values. An example of such a case is the scheduling function as described in (20), which is constrained to the interval $[0, 1]$ by definition, or the velocity of a specific fluid within relativistic scenarios which is subject to an upper limit that cannot surpass the speed of light [57]. In the context of our specific problem, the sole independent variable is time, thereby allowing us to exclude spatial dimensions as inputs to the neural model. Consequently, by considering (23) and the time interval $t \in [0, T]$, we can express the ensemble of output variables of the architecture in the following manner:

$$\mathbf{u}(t) \approx \mathbf{u}_\Theta(t) = \sigma_K \left(\mathbf{u}_\Theta^{(K)} \circ \sigma_{K-1} \circ \mathbf{u}_\Theta^{(K-1)} \circ \dots \circ \sigma_1 \circ \mathbf{u}_\Theta^{(1)} \right) (t) \quad (24)$$

Equation (24) showcases the mathematical representation of the approximation proposed by the underlying neural network in our methodology. This approach aims to closely resemble the actual physical solution following the completion of the training process. 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 training process and are distinct for each neuron [83]. However, our focus lies primarily on establishing the definition of physical inductive biases within the network. Consequently, each layer k will be associated with an activation function denoted as σ_k , which uniformly affects the output tensor of that particular layer.

3.2 Inductive biases and optimization

Our approach involves employing a methodology based on PINNs, which allows us to incorporate strong inductive biases and a priori knowledge into the neural network. This incorporation 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

$$\mathbf{u}_\Theta(t) := (\lambda, \mathcal{A}_{\text{CD}}, \mathcal{C})_\Theta^3. \quad (25)$$

Here, $\lambda \in \mathbb{R}$ denotes the scheduling function, while $\mathcal{A}_{\text{CD}} \in \mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$ represents the counterdiabatic terms of the evolution, and $\mathcal{C} \in \mathbb{R}^{4^{N_Q}}$ stands for the set of coefficients in which these counterdiabatic terms can be linearly decomposed attending to all the possible tensors that come from the Kronecker products of the possible combinations according to both the number of qubits considered and the set of identity and Pauli matrices, $\{\mathbf{I}, \sigma_X, \sigma_Y, \sigma_Z\}$ (see [84] Chapter 2.1). In general, \mathcal{A}_{CD} is an operator composed of complex terms and will be of size $2^{N_Q} \times 2^{N_Q}$ with N_Q being the number of qubits into consideration. Moreover, notwithstanding that the dependency may not always be explicitly stated, all variables emerging from the PINN are contingent upon the input time. Hence, the network yields solutions for each of the considered inference times.

The underlying neural network will be optimized using the so-called ‘‘soft enforcement’’ technique, as introduced in equations (6,7,8). We will adapt this methodology to our specific scenario of Hamiltonian dynamics, where the global cost function can be decomposed into multiple terms. Specifically, we will address the initial and final time conditions for an input time interval $t \in [t_{\min}, t_{\max}]$ as described in Equations (26) and (27), respectively.

$$\mathcal{L}_{\text{IC}} := \frac{\omega_{\text{IC},1}}{N_{\text{IC}}} \sum_{\{t_{\min}\}} |\lambda(t_{\min})|^2 + \frac{\omega_{\text{IC},2}}{N_{\text{IC}}} \sum_{\{t_{\min}\}} |\mathcal{H}(t_{\min}) - \mathcal{H}_{\text{HF}}|^2, \quad (26)$$

$$\mathcal{L}_{\text{FC}} := \frac{\omega_{\text{FC},1}}{N_{\text{FC}}} \sum_{\{t_{\max}\}} |\lambda(t_{\max}) - 1|^2 + \frac{\omega_{\text{FC},2}}{N_{\text{FC}}} \sum_{\{t_{\max}\}} |\mathcal{H}(t_{\max}) - \mathcal{H}_{\text{problem}}|^2. \quad (27)$$

In the definitions above, $(\omega_{\text{IC},1}, \omega_{\text{IC},2})$ and $(\omega_{\text{FC},1}, \omega_{\text{FC},2})$ are the weights of the mixture in the calculation of the total loss function and whose values will depend on the knowledge applied as well as on the problem being treated, while N_{IC} and N_{FC} represent the number of sample points at the initial and final instants, respectively. Regarding the

³Henceforth, 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.

scheduling function of the problem, denoted as $\lambda(t)$, this function shall delineate the progression of physical states subsequent to the introduction of counterdiabatic terms as stipulated in Equation (20). At the initial time instant, we shall impose a condition where $\lambda(t_{\min})$ equals zero, i.e., $\lambda(t_{\min}) = 0$. Similarly, at the terminal time instant, we would prescribe that $\lambda(t_{\max})$ assumes a value of one, i.e., $\lambda(t_{\max}) = 1$, as per its formal definition [46]. The aforementioned conditions outlined correspond to the physical limitations that are inherently necessary to be satisfied within our specific scenario. At the initial moment, denoted as $t = t_{\min}$, we enforce the scheduling function to be zero, while ensuring that the resulting Hamiltonian operator (20) is equivalent to the one obtained through the Hartree-Fock method (see [85] Chapter 2.2), defined as $\mathcal{H}(t_{\min}) = \mathcal{H}_{\text{HF}}$. Additionally, by incorporating counterdiabatic terms, our intention is to accelerate the adiabatic transition and reduce the computational complexity of the underlying circuits [23, 43]. This, however, does not impede our knowledge of the final Hamiltonian operator, denoted as $\mathcal{H}_{\text{problem}}$, which can be computed via advanced numerical methods in the chemistry field [70]. These conditions, combined with the requirement that the scheduling function must be equal to one at the conclusion of the evolution, collectively constitute the final conditions mandated by the methodology.

After establishing the initial and final inductive biases explicitly, it becomes crucial to elucidate the physics governing the intermediate time periods within the interval. Upon acquiring all the aforementioned components, we possess the necessary elements to construct the complete Hamiltonian operator for the given problem, as delineated in Equation (20). However, for the purpose of coherence, we reiterate the expression here to avoid any disruption in the logical progression.

$$\mathcal{H}(t) = \mathcal{H}_{\text{AD}}(t) + \frac{d\lambda}{dt} \mathcal{A}_{\text{CD}}(t), \quad \text{with} \quad \mathcal{H}_{\text{AD}}(t) := (1 - \lambda(t)) \mathcal{H}_{\text{HF}} + \lambda(t) \mathcal{H}_{\text{problem}}. \quad (28)$$

As it is well-known, the Hamiltonian operators in their general form are operators comprising complex numbers, which necessitate adherence to the property of Hermiticity. By satisfying this requirement, the operators can be appropriately interpreted as physical observables, enabling extraction of values from energy states that are purely real. Consequently, it is obvious that our operators should possess Hermitian properties. Furthermore, it is imperative to impose the condition that the neural network yields the solution for the Gauge potential (counterdiabatic terms) that achieves the utmost reduction in the physical action within the given scenario. This entails selecting the solution that results in achieving

$$\frac{\delta \mathcal{S}(\mathcal{A}_{\text{CD}})}{\delta \mathcal{A}_{\text{CD}}} = 0. \quad (29)$$

The minimization of the physical action represents a crucial requirement for ensuring that the employed methodology yields an optimal operator $\mathcal{A}_{\text{CD}}(t)$ under specific conditions, encompassing local temporary development, robustness, availability, and experimental accessibility, among other factors. Research studies, such as [64] and related literature, demonstrate the impact of the action in recovering the Euler-Lagrange Equation (15). Consequently, demanding the neural network to minimize the action is entirely equivalent to defining the term associated with the physical loss, as described in (30). Moreover, it is well-established that the temporal rate of change of the scheduling function, denoted as $\frac{d\lambda}{dt} := \dot{\lambda}$, represents the velocity or rate at which the non-adiabatic components drive the evolution of the system in the presence of the total Hamiltonian operator. Consequently, when the derivative approaches zero, i.e., $\dot{\lambda} = 0$, the conventional adiabatic Hamiltonian is recovered. However, it is undesirable for the counterdiabatic terms to greatly surpass the adiabatic counterparts, as their purpose is to expedite the process without exerting dominant influence. Hence, it is essential that the time derivative of λ remains small, yet not entirely nullified. This particular information must be communicated to the PINN through Equation (31).

$$\mathcal{L}_{\text{Least Action}} := \frac{\omega_{\text{Action}}}{N_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left[i \frac{\partial \mathcal{H}_{\text{AD}}(t)}{\partial \lambda} - [\mathcal{A}_{\text{CD}}(t), \mathcal{H}_{\text{AD}}(t)], \mathcal{H}_{\text{AD}}(t) \right]. \quad (30)$$

$$\mathcal{L}_{\text{Adiabaticity}} := \frac{\omega_{\text{Ad}}}{N_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left| \frac{d\lambda}{dt} \right|^2. \quad (31)$$

As mentioned in Section 2.3, the set $\{\sigma_0, \sigma_X, \sigma_Y, \sigma_Z\} \in \mathcal{M}_{2 \times 2}(\mathbb{C})$ form an orthogonal basis of the Hilbert space of 2×2 Hermitian matrices. Consequently, it is both logical and well-founded to seek the computation of the operator \mathcal{A}_{CD} as a composite of tensor products involving the previously introduced Pauli matrices for a certain system of qubits. By doing so, it is possible to construct a linear combination of tensor products involving these matrices, yielding a set of

coefficients denoted as $\mathcal{C} \in \mathbb{R}^{4^{N_Q}}$. Each element within this set represents the relative magnitude of each term within the operator. Therefore, the decomposition expressed in Equation (32) would enable us to perform efficient simulations and facilitate a more accessible analysis of physical systems through the utilization of quantum circuit models.

$$\mathcal{A}'_{\text{CD}}(t) := \sum_{i,j,\dots,N_Q \in \{0,X,Y,Z\}} \mathcal{C}_{i,j,\dots,N_Q}(t) (\sigma_i \otimes \sigma_j \otimes \dots \otimes \sigma_{N_Q}). \quad (32)$$

In this study, we employ $\mathcal{A}'_{\text{CD}}(t)$ to represent the non-adiabatic terms. This notation serves to distinguish this expansion, which takes the form of a linear combination, from the operator that serves as an output of the PINN. To achieve this objective, it is necessary to introduce an additional term into the loss function of the neural network, which directly affects the set of coefficients denoted as $\mathcal{C}(t)$, as shown in Equation (33). Consequently, these terms are dynamically adjusted during the training process in order to construct the decomposition of the Gauge operator. By employing the specified procedure and adhering to the prescribed requirement, our methodology exhibits the capability to yield these scalar quantities not only at the initial and final moments but also throughout the entire temporal interval. This is attributable to the fact that these scalars, in a general sense, are functions contingent upon time.

$$\mathcal{L}_{\text{Coupling}} := \frac{\omega_{\text{Coupling}}}{N_{\mathcal{F}}} \sum_{(t_{\min}, t_{\max})} \left| \mathcal{A}_{\text{CD}}(t) - \mathcal{A}'_{\text{CD}}(t) \right|^2. \quad (33)$$

Once all the requisite components have been specified according to Equations (30), (31), and (32), it becomes feasible to establish the loss term for our PINN, Equation (34), which is solely linked to the underlying differential equations. In this context, the vector $(\omega_{\text{Action}}, \omega_{\text{Adiabaticity}}, \omega_{\text{Coupling}})$ denotes the weights employed in the combination process when constructing the resultant term. Thus, by considering Equation (5) and acknowledging the pre-established mixture weights within our loss terms, the formulation of the final loss function, Equation (35), becomes straightforward. This incorporates the loss associated with the PDEs outlined in Equation (34), as well as the temporal constraints at the initial and final temporal steps, as indicated in (26) and (27). Consequently, the defined loss function serves as the physical metric that guides the optimization process, dictating the objective for the neural network within our methodology to minimize.

$$\mathcal{L}_{\mathcal{F}} := \mathcal{L}_{\text{Least Action}} + \mathcal{L}_{\text{Adiabaticity}} + \mathcal{L}_{\text{Coupling}}. \quad (34)$$

$$\mathcal{L} := \mathcal{L}_{\text{IC}} + \mathcal{L}_{\mathcal{F}C} + \mathcal{L}_{\mathcal{F}}. \quad (35)$$

In addition, the network has not explicitly been required to satisfy the necessary condition of operators hermiticity even though it can be achieved by including an additional term in Equation (34) that minimizes the difference $\left| \mathcal{A}_{\text{CD}} - \mathcal{A}'_{\text{CD}} \right|^2$. Nonetheless, such a restriction is not obligatory and would be redundant for the PINN. If the coefficients $\mathcal{C} \in \mathbb{R}^{N_Q}$ are defined as real, i.e., $\mathcal{C} = \mathcal{C}^*$, then it is evident from the decomposition of \mathcal{A}'_{CD} in Equation (32) that we recover $\mathcal{A}'_{\text{CD}} - \mathcal{A}'_{\text{CD}}^\dagger = 0$ naturally, without necessitating any additional requirements. Therefore, the physical condition expressed in Equation (33) is more than sufficient for the neural network to ensure the hermiticity of the operator \mathcal{A}_{CD} , and since $\mathcal{H}_{\text{HF}}, \mathcal{H}_{\text{problem}} \in \mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{R})$ [23], the hermiticity of the complete Hamiltonian operator is also ensured.

After considering all the relevant factors, a comprehensive description of our methodology can be provided. In order to accomplish this, we will utilize a visual aid in the form of a diagram (Figure 1), and a step-by-step algorithm outlined in Algorithm 1. The initial step involves identifying the independent variable(s) that will influence our outcomes. In our particular case, we only have the temporal variable, denoted as t , defined within the interval $[t_{\min}, t_{\max}]$, commonly specified as $[t_{\min}, t_{\max}] = [0, 1]$. Consequently, we need to select an appropriate sampling method. Various methods are available, including uniform sampling with equidistant points, random sampling, Latin hypercube sampling [86, 87], and Sobol sampling [88]. The choice of method is somewhat arbitrary, with greater impact when considering fewer time points, as its significance diminishes as the sample size approaches infinity. In particular, the Sobol sampling has exhibited significant advancements in prior inquiries within the relevant body of literature [89]. In our investigation, we have employed this approach, which entails generating pseudo-random samples using powers of two. This technique results in a more homogeneous distribution of points within the space, with reduced overlap compared to completely random sampling. Nonetheless, it is important to acknowledge the inherent randomness involved in this approach. After generating the time domain, it serves as the input to the network, which will be constructed as a sequence of fully connected K dense layers, encompassing both the input and output layers. The number of layers and the number of

neurons in each layer (N_k) are hyperparameters that will be predetermined prior to training. While these parameters can be adjusted, it is anticipated that a relatively conventional configuration, such as 6 or 7 layers with 30 to 40 neurons each, will suffice for characterizing the counterdiabatic driving problem. Nevertheless, the complexity of individual problems may necessitate varying numbers of layers and/or neurons.

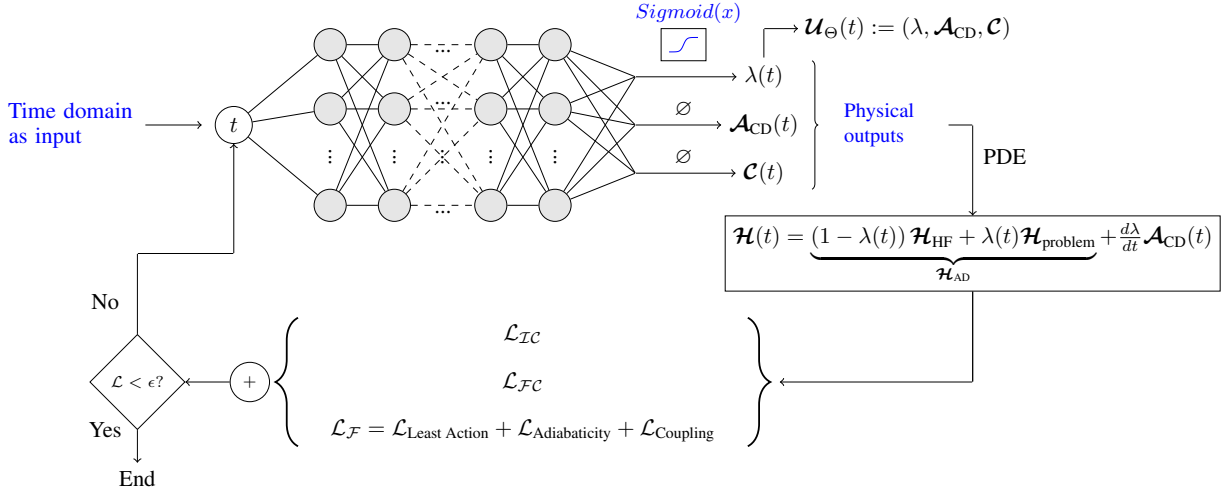


Figure 1: The methodology follows a general procedure where time, represented by the variable t , is the only independent physical variable considered, incorporated into our network as a tensor. The output of the methodology comprises three elements: the scalar variable of the scheduling function denoted as $\lambda(t)$; the operator representing the counterdiabatic terms of the process, expressed as $\mathcal{A}_{\text{CD}}(t)$, with each of its components treated as an independent output; and the coefficients $\mathcal{C}(t)$ denote the general decomposition coefficients of the operator \mathcal{A}_{CD} expressed as a linear combination of tensors resulting from all possible Kronecker products formed between the set of operators comprising the identity matrix and the Pauli operators. Subsequently, the computations required to construct the total Hamiltonian, denoted as $\mathcal{H}(t)$, are carried out. Various physical constraints are imposed during these computations, including inductive biases. These constraints adhere to the principle of minimum action, satisfy the initial and final conditions, and ensure the hermiticity of the physical operators among other specifications. At each step of the training process, the total loss is calculated by aggregating the contributions from all imposed conditions, denoted as \mathcal{L} . This calculation continues until a specific training period is reached or until a predetermined error threshold is achieved.

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 $\mathbf{U}_{\Theta}(t) := (\lambda, \mathcal{A}_{\text{CD}}, \mathcal{C})_{\Theta}$.

- 1 **Specify the training domain:** Generate the temporal interval t using *Sobol sequence* [88].
- 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 $\Theta = \{\mathbf{W}^{(k)}, \mathbf{b}^{(k)}\}_{1 \leq k \leq K}$, using the Glorot initialization method as described in the reference [90].
- 3 **By employing the principles of automatic differentiation**, compute the derivative $\frac{d\lambda}{dt}$ from the output and subsequently construct the operator $\mathcal{H}(t)$ in accordance with the expression (20).
- 4 **Construct the initial and final losses in time** using the MSE metric as \mathcal{L}_{IC} (26) and \mathcal{L}_{FC} (27), respectively.
- 5 **Derive the comprehensive set of physical loss terms** associated with the differential equations, \mathcal{L} , incorporating considerations for the Principle of Least Action, adherence to adiabatic conditions, the decomposition in terms of Pauli tensor products and the hermiticity of the physical operators as a direct consequence. Subsequently, formulate the aggregate physical loss as follows:

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

- 6 **Update the set of network variables** by backpropagation, $\{\mathbf{W}^{(k)}, \mathbf{b}^{(k)}\}_{1 \leq k \leq K}$, that minimizes the total loss using a certain defined optimizer:

$$\Theta^* := \arg \min (\mathcal{L}(\Theta)).$$

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

With the establishment of the input domain and the construction of the network, we are now capable of extracting the output and interpreting it as the tensor of physical variables denoted by $\mathbf{U}_{\Theta}(t) = (\lambda, \mathcal{A}_{\text{CD}}, \mathcal{C})_{\Theta}$, where $\lambda \in \mathbb{R}$,

$\mathcal{A}_{\text{CD}} \in \mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$, and $\mathcal{C} \in \mathbb{R}^{4^{N_Q}}$. Subsequently, the derivative $\frac{d\lambda}{dt}$ can be straightforwardly computed using automatic differentiation, getting the Hamiltonian operator $\mathcal{H}(t)$ as described in (20). Furthermore, the initial and final temporal constraints, defined in equations (26) and (27), are necessary. These constraints are accompanied by the calculation of the loss associated with the underlying PDEs stated in Equation (34), which incorporates the physical constraints outlined in equations (30), (31), and (33). These physical restrictions represent the fulfillment of the Principle of Least Action, the agreement with the adiabaticity, and the decomposition of the operator $\mathcal{A}_{\text{CD}}(t)$ as previously explained. By combining all these components, the final loss metric \mathcal{L} given in Equation (35) is computed, and the set of trainable variables Θ is updated via backpropagation, minimizing the loss through an optimization process.

4 Numerical experiments and results

In this section, various numerical results obtained through our DL-based methodology are presented, as outlined in Section 3. This approach has been applied to address the H_2 molecule problem within the STO-3G basis, considering different bond distances between the particles and utilizing a 2-qubit representation. The initial and final Hamiltonian operators used for the evolution, as denoted by the PDE in (20), are listed in Table 1 and were configured following the guidelines in [70]. Since our numerical example involves a 2-qubit representation, these operators will possess a matrix size of 4×4 . Furthermore, it is pertinent to note that these operators are real-valued by definition, i.e., $\mathcal{H}_{\text{HF}}, \mathcal{H}_{\text{problem}} \in \mathcal{M}_{4 \times 4}(\mathbb{R})$. In a general context, and unless explicitly stated otherwise, all conducted trainings comprised a total of 500,000 epochs (iterations) dedicated to updating the Θ parameters of the used PINN. The training procedure utilized the *PyTorch* module [91], employing the *Adam* optimizer [92] with a fixed learning rate of 10^{-5} . This learning rate was chosen to be sufficiently small, ensuring convergence without excessive numerical noise. Throughout all examples, the sampling method used has been the Sobol sampling [88] as stated in Section 3. The number of sampling points at the initial and final time instances of the evolution, denoted as N_{IC} and N_{FC} respectively, has remained constant at 2^{13} . However, the number of points in the inner part of the interval, which is linked to the underlying PDE and represented as $N_{\mathcal{F}}$, was subject to variation and examination in the experiments. Moreover, if not otherwise specified, the neural architecture used for all the examples consists of six hidden layers with 30 neurons each.

Table 1: Numerical configurations for the initial Hamiltonian operator, denoted as \mathcal{H}_{HF} , and the final Hamiltonian operator, denoted as $\mathcal{H}_{\text{problem}}$, obtained using the quantum computing library *Qiskit* [70]. These configurations correspond to the evolution of the molecule H_2 in the STO-3G basis and are represented by their respective matrix descriptions given in Equations (19) and (17), respectively.

	\mathcal{H}_{HF}				$\mathcal{H}_{\text{problem}}$			
$d = 1.0 \text{ \AA}$	-0.5490812	0	0	0	-0.5490812	0	0	0.19679058
	0	-1.0661087	0	0	0	-1.0661087	0.19679058	0
	0	0	0.00400595	0	0	0.19679058	0.00400595	0
	0	0	0	-0.5490812	0.19679058	0	0	-0.5490812
$d = 1.5 \text{ \AA}$	-0.6610488	0	0	0	-0.6610488	0	0	0.22953594
	0	-0.91087353	0	0	0	-0.91087353	0.22953594	0
	0	0	-0.3944683	0	0	0.22953594	-0.3944683	0
	0	0	0	-0.6610488	0.22953594	0	0	-0.6610488
$d = 2.0 \text{ \AA}$	-0.66539884	0	0	0	-0.66539884	0	0	0.25913846
	0	-0.7837927	0	0	0	-0.7837927	0.25913846	0
	0	0	-0.5412806	0	0	0.25913846	-0.5412806	0
	0	0	0	-0.66539884	0.25913846	0	0	-0.66539884
$d = 2.5 \text{ \AA}$	-0.649429	0	0	0	-0.649429	0	0	0.28221005
	0	-0.7029436	0	0	0	-0.7029436	0.28221005	0
	0	0	-0.5944048	0	0	0.28221005	-0.5944048	0
	0	0	0	-0.649429	0.28221005	0	0	-0.649429

To illustrate initial general results, Figure 2 presents the physical loss functions employed for optimizing the neural network, as defined in Section 3. It is essential to emphasize that each training process would exhibit in general distinctive loss curves, influenced by factors such as the number of qubits used for representation and the specific molecular system being studied, including the considered bond distances between molecules, among other features. However, the figure showcases the cost function for the specific scenario under investigation: the H_2 molecule, represented by a 2-qubit system in the STO-3G basis, with specifications outlined in Section 2.3. The left subfigure displays the three constituents comprising the total loss \mathcal{L} (35), namely, $\mathcal{L}_{\mathcal{F}}$, \mathcal{L}_{IC} , and \mathcal{L}_{FC} , corresponding to the

residual conditions of the PDE, the initial conditions over time, and the final conditions, respectively. The latter two diminish rapidly, converging to magnitudes on the order of 10^{-4} or even lower, especially evident for $\mathcal{L}_{\mathcal{FC}}$, where the error is so minute that any variation induces substantial noise on a logarithmic scale. Conversely, the residual loss $\mathcal{L}_{\mathcal{F}}$ (34) encompasses three internal terms, generating discernible tension among them, which impedes its reduction to such small orders of magnitude. This becomes more evident in the right subfigure, which provides a detailed breakdown of these terms. It is evident that the loss term responsible for minimizing the Principle of Least Action or, equivalently, the Euler-Lagrange equations (30), attains the lowest value, thus ensuring a highly satisfactory minimization of the action and, consequently, the attainment of a potential gauge $\mathcal{A}_{\text{CD}}(t)$ that adheres closely to its prescribed guidelines. Additionally, the graphical representation includes a loss term denoted as $\mathcal{L}_{\text{Hermiticity}}$, which assesses the quadratic discrepancy between $\mathcal{A}_{\text{CD}}(t)$ and its corresponding operator, defined in a similar way to how the other terms are. However, this factor is not incorporated in the total loss; instead, it serves as a representation allowing us to confirm that minimizing the loss $\mathcal{L}_{\text{Coupling}}$ (33) through the decomposition of the potential gauge directly enforces its hermiticity, given the strict real-valued nature of the coefficients $\mathcal{C}(t)$ involved in the decomposition (32). Thus, for all the examples shown in this article, the mix weights considered for each of the individual loss terms have been preset as:

$$(\omega_{\mathcal{IC}}, \omega_{\mathcal{FC}}, \omega_{\text{Action}}, \omega_{\text{Ad}}, \omega_{\text{Coupling}}) = (10^3, 10^3, 10^2, 5 \times 10^{-1}, 2.5 \times 10^2), \quad (36)$$

where $\omega_{\mathcal{IC},1} = \omega_{\mathcal{IC},2} = \omega_{\mathcal{IC}}$ and $\omega_{\mathcal{FC},1} = \omega_{\mathcal{FC},2} = \omega_{\mathcal{FC}}$ in Equations (26) and (27). These weights play a crucial role in determining the relative emphasis that the neural network places on each term, essentially representing their respective priorities. For our specific objectives, ensuring prompt and robust satisfaction of the initial and final conditions in time is of crucial importance. Consequently, we set $\omega_{\mathcal{IC}}$ and $\omega_{\mathcal{FC}}$ to significantly larger values than ω_{Action} , ω_{Ad} , and ω_{Coupling} . Following the same thoughts, the minimization of the action is also essential in our methodology as it leads us to identify the most appropriate operator $\mathcal{A}_{\text{CD}}(t)$ for the system [64], and we express it through its linear combination (32). Consequently, these two weights will be considerably higher, yet an order of magnitude lower than the first two. Particularly, ω_{Coupling} is considered slightly higher, as this condition inherently encompasses the constraint on the hermiticity of the operator. Lastly, the condition of adiabaticity will be assigned the lowest weight, reflecting our intent to recover the limit of the adiabatic theory without permitting it to dominate the overall metric.

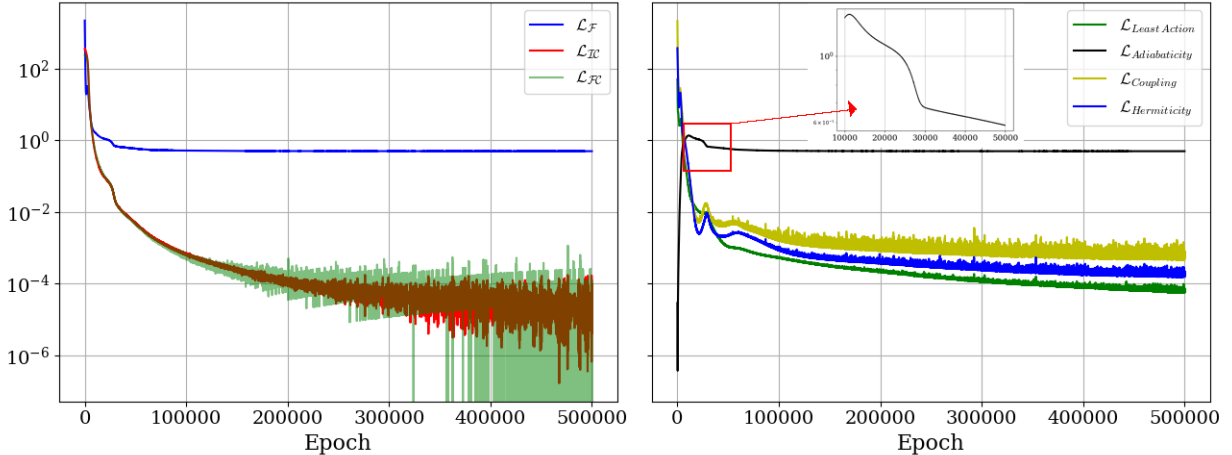


Figure 2: Analysis of the evolution of the loss function during the training process. On the left we illustrate the dynamic changes of the components contributing to the total loss, \mathcal{L} , as defined in Equation (35). On the right side of the graph, each individual constituent of the loss, $\mathcal{L}_{\mathcal{F}}$, is presented, corresponding to the different physical aspects under consideration. It is important to note that the loss term $\mathcal{L}_{\text{Hermiticity}}$ is included in the plot, although it remains undefined and unused throughout the training. This term quantifies the discrepancy between \mathcal{A}_{CD} and its adjoint operator but is solely provided for visualization purposes, tracking its reduction relative to $\mathcal{L}_{\text{Coupling}}$ due to their mathematical equivalence.

We could continue our analysis of the right-hand subfigure by specifically putting our attention on the curve that corresponds to the loss of recovery of the adiabatic theory, denoted as $\mathcal{L}_{\text{Adiabaticity}}$ (31) and represented in a solid black curve. This loss exhibits a marginal decline, commencing from a value proximal to zero at the onset of the training process. Subsequently, it rises to approximately 10^2 before descending to around 10^0 and eventually maintaining a constant level of magnitude for the duration of the evolution. The physical reason behind this phenomenon is clear: while the theory necessitates a decrease in the adiabatic speed, $\frac{d\lambda}{dt}$, the PINN fails to recognize the appropriateness of taking it to values lower than what is observed in our results. This is primarily attributed to the presence of multiple

terms within the loss function, not solely related to adiabaticity recovery, thereby creating inherent tension in defining the physical problem as a whole. Consequently, the solution predicted by the methodology does not significantly benefit from further reductions in the adiabatic speed. The fast saturation of adiabatic recovery is the main cause of the considerably elevated value for $\mathcal{L}_{\mathcal{F}}$ observed in the left subfigure, underscoring the importance of isolating and analyzing each term separately. Thus, a zoom has been applied to $\mathcal{L}_{\text{Adiabaticity}}$ between 30,000 and 50,000 iterations, a range during which the λ function transitions from a sigmoidal to a linear behavior. Consequently, it assumes a temporal derivative of 1, i.e., $\frac{d\lambda}{dt} = 1$. A more comprehensive discussion on this topic will be presented in the following section. It is important to note that unless otherwise specified, all the results shown in this section have been carried out for the system formed by 2 qubits representing the H_2 molecule in the STO-3G basis, following the guidelines of Section 2.3.

4.1 Scheduling function, $\lambda(t)$

The output of our methodology is triple. Firstly, it enables the retrieval of the scheduling function λ as a time-dependent variable. Secondly, it facilitates the extraction of the matrix components (i, j) of the potential gauge $\mathcal{A}_{\text{CD}}(t)$, considering also the temporal variable, and it also enables the obtaining of the $\mathcal{C}(t)$ components of its decomposition through time. This achievement is made possible by the utilization of a cost metric defined on the underlying PDE comprising three distinct terms, as stated in Equation (34). Each of these terms compels the neural network to update one of the outputs while adhering to the imposed physical constraints. In particular, the term $\mathcal{L}_{\text{Adiabaticity}}$ plays a crucial role in enforcing the adherence of the function $\lambda(t)$ to the physical evolution, thereby recovering, in the most accurate approximation, the adiabatic theory that has been empirically demonstrated to hold true. Among all the terms taken into account within the total cost function, the neural network exhibits the lowest attention towards fulfilling the requirement of recovering the fully adiabatic theory, $\mathcal{L}_{\text{Adiabaticity}}$. As illustrated in Figure 2, our PINN ceases to actively optimize this particular term after approximately 50,000 iterations. Consequently, the scheduling function $\lambda(t)$ converges to an optimal solution wherein its time derivative, representing the adiabatic velocity, maintains an approximately constant value of 1 throughout the temporal evolution. As a consequence, the optimal form of the λ function predicted for the system is the one that adheres to the condition $\lambda(t) = t$ for the evolution, as governed by the counterdiabatic differential Equation (20).

This phenomenon is exemplified in Figure 3, where we show the values of $\lambda(t)$ and its temporal derivative in the left and right subfigures, respectively, for different training steps in both cases concerning the 2-qubit system representing the H_2 molecule. As observed, even after 15,000 and 25,000 epochs, the function maintains a sigmoidal shape, a widely employed representation in the existing literature [23, 46]. This sigmoidal form is considered appropriate from a theoretical perspective due to its derivative taking the form of a ‘‘bell curve’’, facilitating the adiabatic terms \mathcal{A}_{CD} to present higher values at intermediate values during time evolution while effectively being turned off at the initial and temporal instants. It should be noted, however, that this sigmoidal shape for λ emerges predominantly during the early phases of the training thereby helping in the driving of the convergence of the methodology towards a more optimal solution. From a theoretical point of view, our ultimate goal is to recover the fully adiabatic theory while incorporating the non-zero presence of counterdiabatic terms. Consequently, our neural network converges towards a function λ that precisely matches the temporal variable t . This outcome signifies the restoration of the original formulation of counterdiabatic driving, as explained in [42], thereby undoing the temporal parameterization of physical operators through a specific set of parameters $\lambda(t)$, which, in our case, corresponds to a single scalar parameterization.

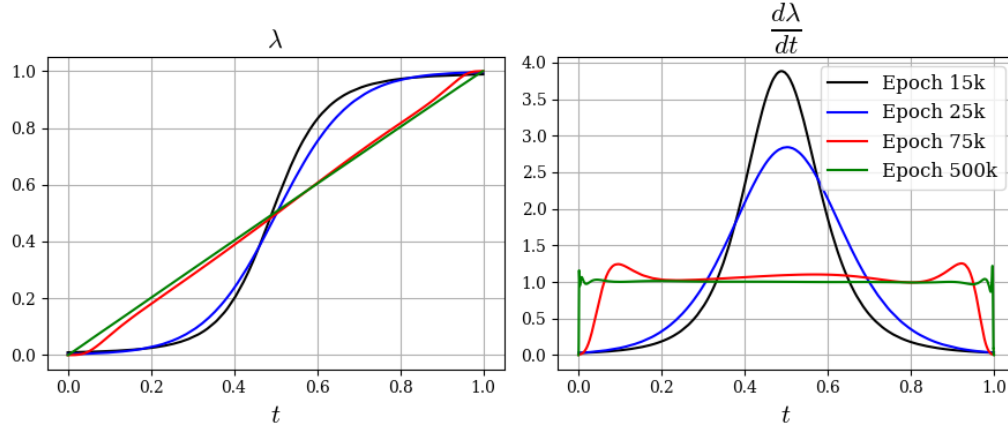


Figure 3: Analysis of the scheduling function λ and its temporal derivative for distinct iterations (epochs) of the training process. Observations reveal that during the initial stages of neural optimization, λ exhibits characteristics resembling a sigmoidal function. However, as the training advances,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

Through this procedure, our objective is to begin with a differential equation containing non-zero counterdiabatic terms and then strive to recover the adiabatic theory in the limiting case. By doing so, we can obtain all the necessary results in accordance with the theory of adiabaticity. In Figure 4, we present the profiles of $\lambda(t)$ and its time derivative for the counterdiabatic (CD) protocol from Equation (20) on the left, while on the right subfigure, we depict the same results but without considering the presence of counterdiabatic terms, i.e., working directly with the adiabatic theory (13). It is evident that for both cases we obtain $\lambda(t) = t$, except for some numerical noise at the edges of the time interval, which is directly related to the nature of the automatic differentiation process [60, 93, 94]. Notably, during the initial stages of training, the neural network capitalizes on the sigmoid-shaped $\lambda(t)$, while simultaneously adjusting other physical conditions. This aids the network in achieving a more optimal solution in recovering the adiabatic theory.

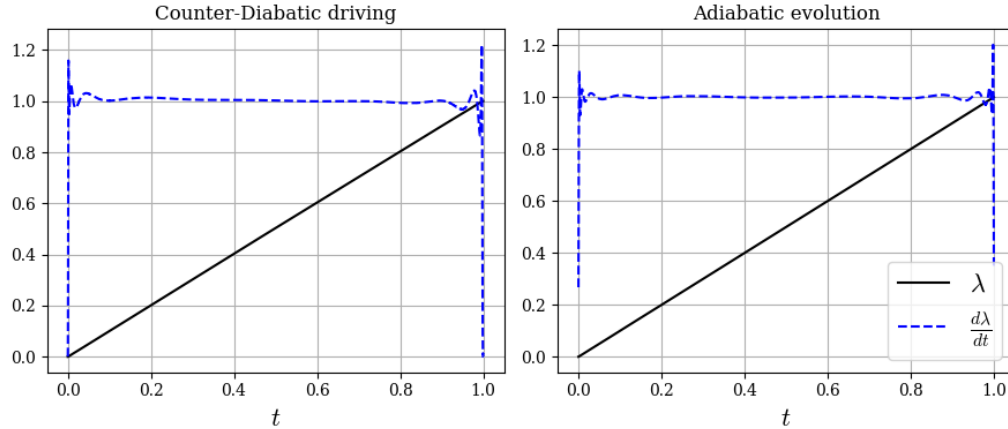


Figure 4: Evolution over time of the *scheduling function* $\lambda(t)$ (in solid black) along with its derivative (in dashed blue) predicted by our methodology for the H_2 molecule in the STO-3G basis set using the CD protocol on the left. On the right, the same is depicted for a fully adiabatic evolution according to expression (13).

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

From the three outputs obtained from the network (see diagram in Figure 1), all the necessary information can be derived. As an initial step, considering the potential gauge $\mathcal{A}_{\text{CD}}(t)$ that minimizes the physical action of the system, the total Hamiltonian operator $\mathcal{H}(t)$ can be computed. The time evolution of all its components is illustrated in Figure 5. Notably, $\mathcal{H} \in \mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$, where N_Q represents the number of qubits, implying that for a 2-particle system, both $\mathcal{H}(t)$ and the remaining operators will have a matrix size of 4×4 . In both depictions, the inherent hermiticity of the observable is evident; the real part of the components exhibits symmetry with respect to the diagonal, while the

imaginary part displays complete antisymmetry, leading to diagonal elements being as close to zero as possible (around the order of $10^{-3} \sim 10^{-4}$). This condition can be further reinforced by increasing its relative weight, ω_{Coupling} . In the mentioned figure, the real and imaginary parts of distinct components of the operator are depicted within the same graph, distinguished by black and blue colors, respectively, and with individual scales. By considering the fulfillment of hermiticity for $\mathcal{H}(t)$, we can now extract its instantaneous eigenvalues, thereby obtaining information about the energy levels of the 2-qubit physical system representing the H_2 molecule with which we are currently conducting our investigation.

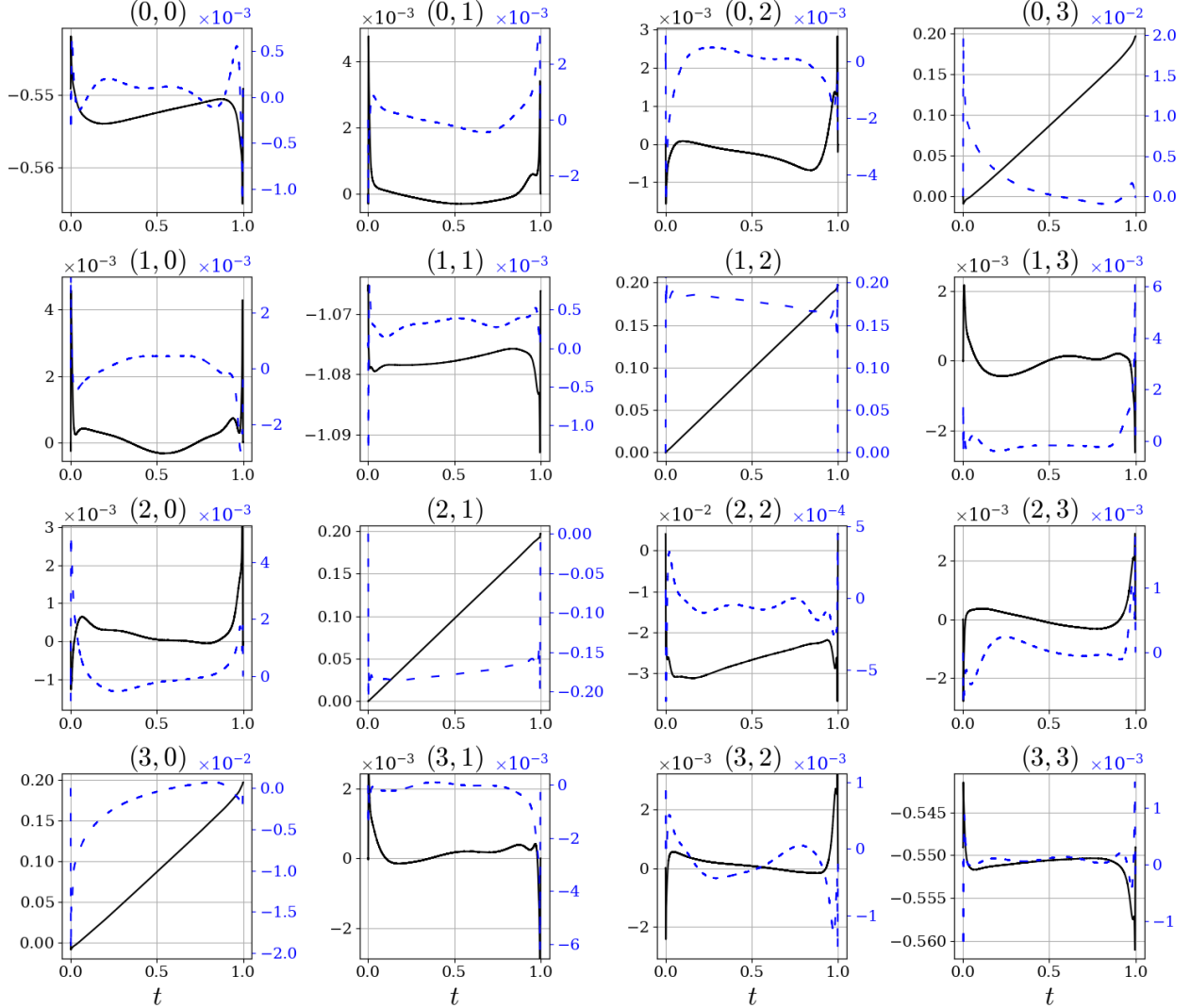


Figure 5: The evolution of the real and imaginary components of the Hamiltonian operator $\mathcal{H}(t)$ for the H_2 molecule is examined using a bond distance of $d = 1.0 \text{ \AA}$. Black and blue colors have been used for real and imaginary parts, respectively, using dashed lines for the latter and a different scale for each one. The findings reveal substantial fluctuations in the values across various time scales. Notably, the natural symmetry and antisymmetry for the real and imaginary components, respectively, arise due to the hermiticity of the operator.

The operator $\mathcal{H}(t)$, as defined in Equation (20), is specifically designed to yield the eigenstates $|n(t)\rangle$ of $\mathcal{H}_{\text{AD}}(t)$ exactly over time. It ensures that no transitions between energy levels, $E_n(t)$, are possible [42]. This property holds for all eigenstates of $\mathcal{H}_{\text{AD}}(t)$, allowing us to interpret the set of states $|n(t)\rangle$ as “moving eigenstates” of the total operator $\mathcal{H}(t)$. Consequently, we can extract the energy levels corresponding to $\mathcal{H}(t)$ throughout the entire time evolution and compare them to the energy levels obtained under the assumption of a completely adiabatic transition, i.e., considering $\frac{d\lambda}{dt} = 0$. Figures 6 and 7 depict the real and imaginary components, respectively, corresponding to two distinct scenarios considered. The first one employs the CD protocol and is shown in the top row, while the second scenario involves

a totally adiabatic transition and is displayed in the bottom row. These investigations were conducted for the H_2 molecule. To broaden the scope of our study, we examined different bond distance values between particles, specifically $d \in \{1.0, 1.5, 2.0, 2.5\}$ Å. The numerical values of the corresponding initial (\mathcal{H}_{HF}) and final ($\mathcal{H}_{\text{problem}}$) Hamiltonian operators are described in Table 1. Notably, Figure 7 reveals that the imaginary part of the eigenvalues (energies) obtained is on the order of 10^{-3} , indicating their proximity to zero. As such, these values have negligible influence on observable analyses. However, it is essential to recognize that this outcome directly arises from the hermiticity of the physical observable. Furthermore, by adjusting the weight ω_{Coupling} , it is possible to further fortify the enforcement of this property. Moreover, in view of the formulation of the Hamiltonian operator under the entirely adiabatic scenario, \mathcal{H}_{AD} (13), and the consideration of the initial and final operators as detailed in Table 1, it is important to note that the scalar nature of the function $\lambda(t)$ ensures the absence of complex numbers. As a result, the imaginary component of the energy levels in the completely adiabatic case is strictly zero, as evidenced in the bottom row of Figure 7.

Regarding the real component of the energies, which is of primary interest, it is observed that in the case of a fully adiabatic transition, these energies demonstrate nearly linear temporal evolution, with diminishing separation as particle bonding increases. The close proximity of energy levels leads to an unfavorable outcome wherein transitions between them become more likely for the system. However, such a challenge is addressed in the CD protocol, wherein energy levels tend to be notably more separated throughout the entire evolution domain. This phenomenon becomes more noticeable for the energy ground state, E_0 , and the first excited level, E_1 . Notably, when the bond distance (d) is set at 2.0 Å and 2.5 Å, it is evident that these two energy levels remain substantially distant, especially at the initial stages of the evolution of the system. This occurrence is highly desirable from an experimental perspective, as it aims to minimize the probability of transitions between energy levels, especially between the ground and the first excited state, given that the system is initially prepared in the E_0 level.

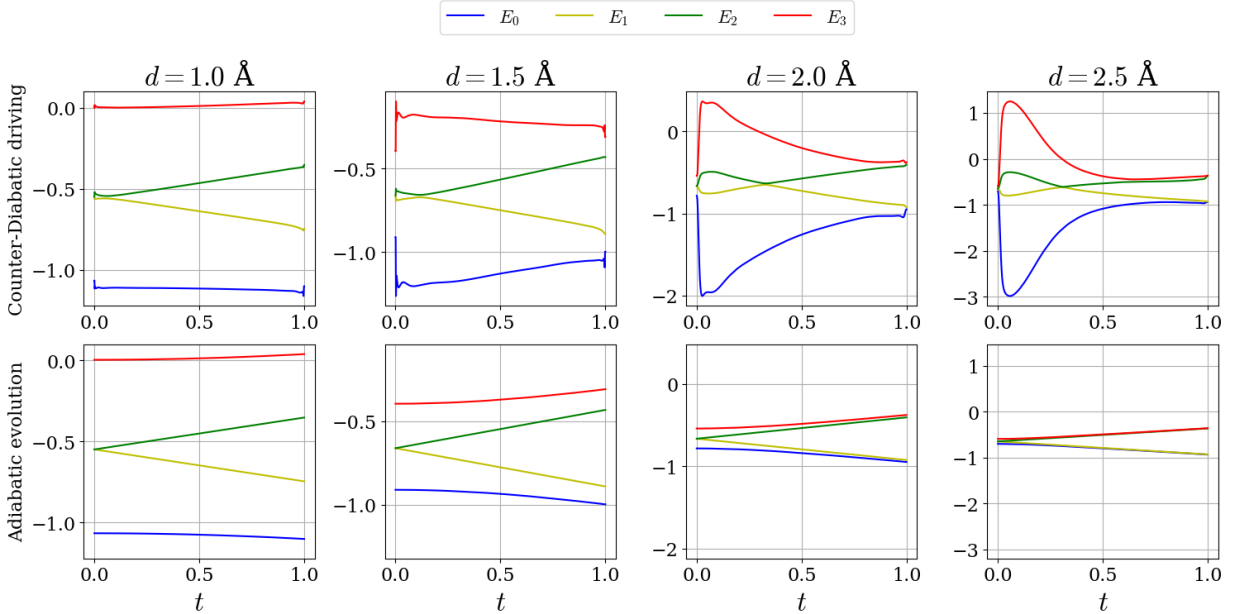


Figure 6: Temporal evolution of the real component of the instantaneous energy levels, namely the eigenvalues $E_n(t)$, describing the molecule H_2 within a system of 2 qubits utilizing the STO-3G basis. These computations are conducted for diverse values of the interparticle bond distance, d . A comparative analysis of the energy levels is presented, showing the results obtained from the CD protocol (20) in the top row, juxtaposed against the levels obtained from the same methodology but with a fully adiabatic transition (13) in the bottom row. It is noteworthy that the energy levels demonstrate a tendency to exhibit greater separation under the CD protocol, a phenomenon that becomes particularly pronounced at $d = 2.0$ Å and $d = 2.5$ Å.

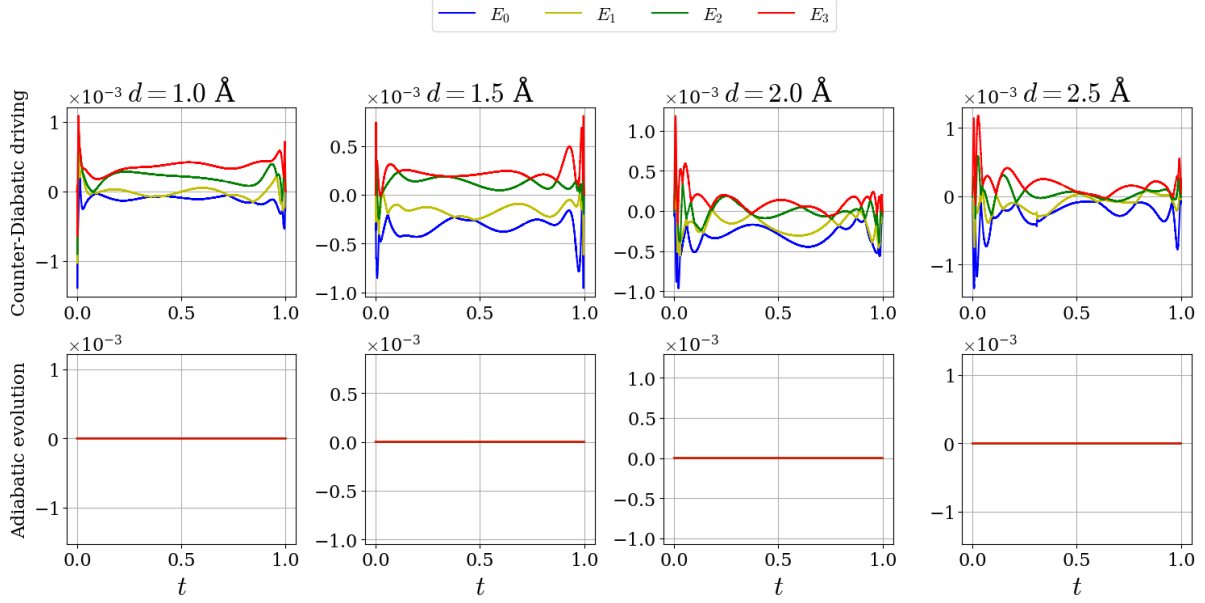


Figure 7: Time-dependent variation of the imaginary component of the eigenvalues $E_n(t)$ investigated for the molecular system H_2 , represented by a 2-qubit configuration in the STO-3G basis. The computational analysis encompasses two scenarios: one employing the CD protocol in the top row, and the other considering a purely adiabatic transition in the bottom row. In the former case, the imaginary components exhibit magnitudes on the order of 10^{-3} , whereas in the latter case, these are precisely zero, as dictated by the definition of the underlying PDE (13).

4.3 $\mathcal{A}_{\text{CD}}(t)$ operator and its decomposition

Our methodology enables us to directly obtain the components of the potential gauge, denoted as $\mathcal{A}_{\text{CD}}(t)$. Consequently, we are capable of visualizing the temporal evolution of both its real and imaginary components. Analogous to the presentation of $\mathcal{H}(t)$ in Figure 5 above, we display the temporal evolution of the corresponding operator \mathcal{A}_{CD} in Figure 8 of this section. We differentiate their respective real and imaginary components on two distinct scales. Observing the plot, it becomes evident that as a physical observable from an experimental standpoint, the real part of the operator, $\text{Re}(\mathcal{A}_{\text{CD}})$, exhibits complete symmetry, while its imaginary part, $\text{Im}(\mathcal{A}_{\text{CD}})$, is fully antisymmetric. This property comes directly from the natural hermiticity of the operator, which, in turn, is imposed indirectly by the condition $\mathcal{L}_{\text{Coupling}}$ (33), together with $\mathcal{C}(t) \in \mathbb{R}^{4^{N_Q}}$.

While exploring the components of $\mathcal{A}_{\text{CD}}(t)$ holds theoretical significance, our primary focus from an experimental point of view lies in obtaining the set of coefficients $\mathcal{C}(t)$ over time. These coefficients play a crucial role as they enable the gauge potential to be expressed as a linear combination of all the possible combinations and interactions between the qubits of the system, thereby allowing a better implementation in a real quantum circuit [23, 46]. The theoretical formulation of this decomposition is represented by Equation (32), wherein the potential is expressed using the required Kronecker products. This formulation takes into account all possible combinations for the 2-qubit system under current investigation. Given the relatively small size of our example system, it remains feasible to consider all the possible combinations of the Pauli matrices as the number of these combinations scales with 4^{N_Q} , being N_Q the number of qubits. Nevertheless, in certain experimental scenarios, it may be unnecessary to explore all combinations and interactions. In such cases, specific specializations and additional requirements can be applied, guided by the methodology we present.

In Figure 9, we present the temporal evolution of the coefficients derived from the decomposition of the H_2 system on the STO-3G basis, 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 denoted as \mathcal{C}_{XY} and its symmetric counterpart \mathcal{C}_{YX} . These findings align with previous literature that employed the NC methodology [23, 46]. 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 C_{ZZ} , C_{XX} , and C_{II} , which warrant consideration as well.

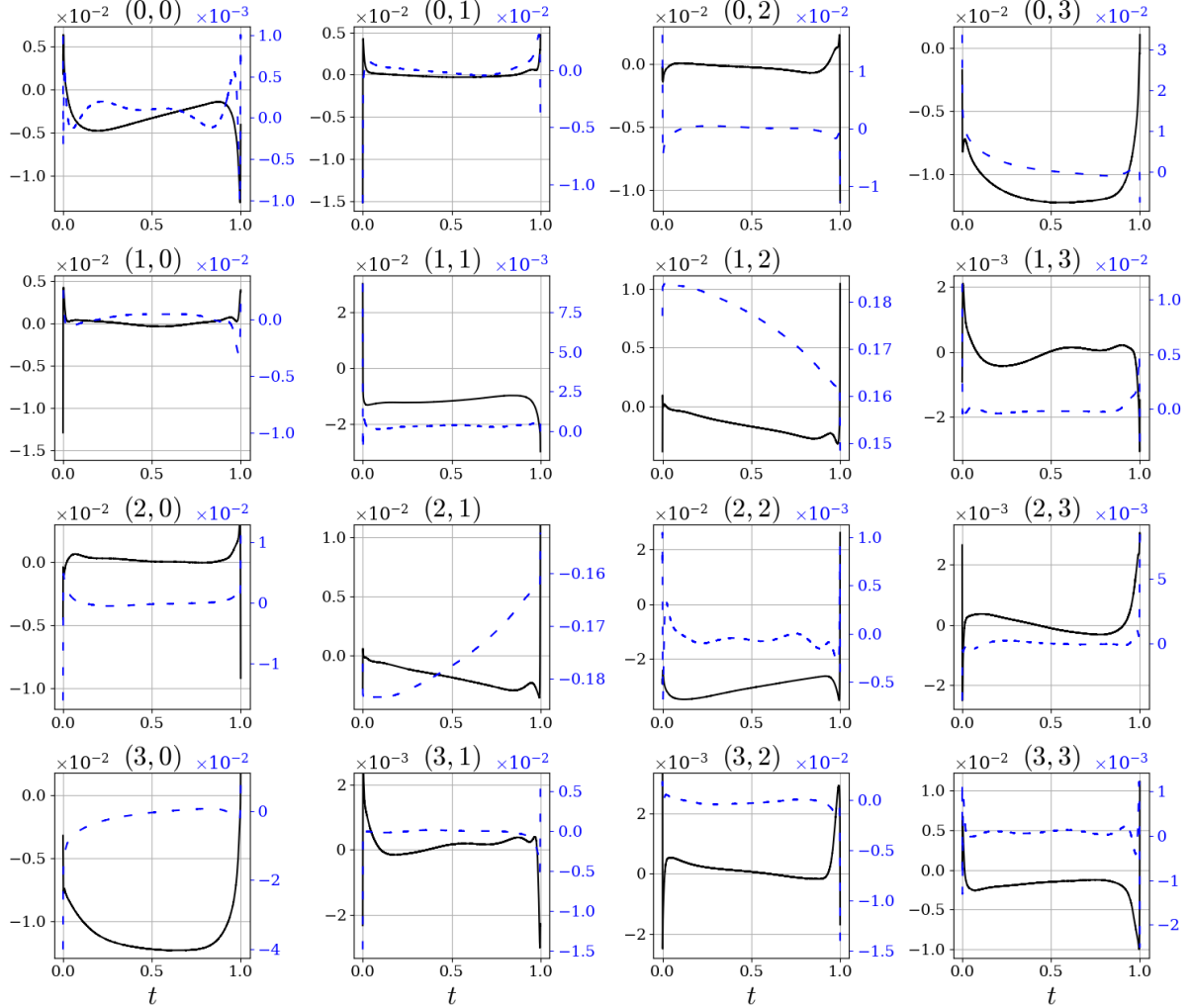


Figure 8: Time evolution of the real and imaginary components of the gauge potential $\mathcal{A}_{CD}(t)$ for the H_2 molecule using the STO-3G basis and a bond distance of $d = 1.0 \text{ \AA}$, represented respectively in black and blue colors using two different scales. It is observed that the values exhibit variations across different orders of magnitude. Notably, the natural symmetry and antisymmetry arises naturally in these components over time due to the hermiticity of the operator.

The determination of these coefficients is universally applicable, i.e., it emerges as an outcome of implementing our methodology for any system, irrespective of the number of qubits considered. Nevertheless, as previously commented in the preceding section, certain specific instances may arise in experimental scenarios where it becomes impractical to account for all possible contributions in the decomposition of $\mathcal{A}_{CD}(t)$. In such circumstances, it would be interesting to adapt the method and restrict the output $\mathcal{C}(t)$ of the neural network to a subset of the entire set, which becomes particularly relevant when dealing with an increasing number of qubits. Consequently, a trade-off between purely theoretical outcomes, exemplified herein, and those of particular experimental significance may always exist.

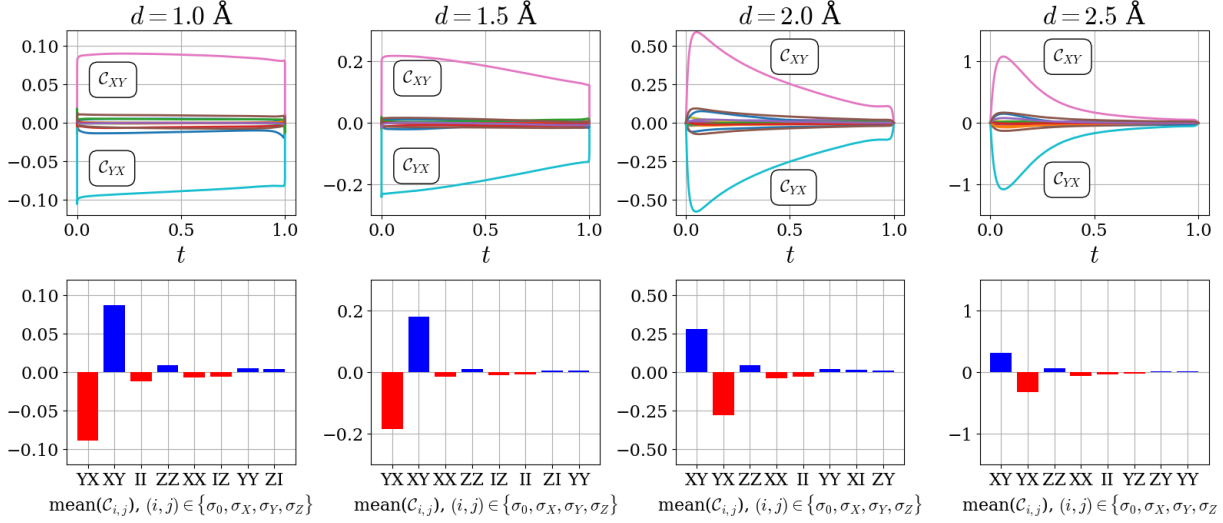


Figure 9: In the upper row, we present the temporal evolutions of the coefficients $\mathcal{C}(t)$ resulting from the decomposition of the operator $\mathcal{A}_{\text{CD}}(t)$ (32) applied to the H_2 molecule utilizing a 2-qubit configuration in the STO-3G basis. In the lower row, a bar chart illustrates the average values of each coefficient, arranged in descending order of magnitude. It is evident that both the coefficient C_{XY} and its symmetric counterpart C_{YX} exert the most substantial influence throughout the entire process, followed by C_{II} , C_{ZZ} and C_{XX} .

4.4 Scalability

So far, we have conducted a study on a 2-qubit system representing the H_2 molecule in the STO-3G basis. However, it is straightforward and easily feasible to modify the base or adjust the number of qubits used in our methodology. This process primarily involves examining the matrix dimensions of the respective network outputs. Specifically, the matrices \mathcal{H} , \mathcal{A}_{CD} , \mathcal{H}_{HF} , $\mathcal{H}_{\text{problem}}$ are elements of the matrix space $\mathcal{M}_{2^{N_Q} \times 2^{N_Q}}(\mathbb{C})$, while $\mathcal{C}(t)$ belongs to $\mathbb{R}^{4^{N_Q}}$. Consequently, the number of components of the variables obtained may increase with the number of qubits, but it is important to note that the neural architecture for all calculations has consistently comprised six internal layers, each containing 30 neurons. This architectural choice is widely documented in the literature and has resulted in state-of-the-art outcomes across various domains of research [56, 57, 95].

In general, the scale and arrangement of the neural architecture of the PINN will depend on the specific problem being addressed, meaning that it is more or less directly related to the difficulty presented by the underlying PDEs. In our case, we are dealing with a single differential equation, which essentially defines $\mathcal{H}(t)$ in Equation (20) whereby involving only one derivative function. Consequently, our chosen neural network architecture efficiently addresses the problem of the CD protocol, as increasing the number of trainable weights usually does not get translated into better results. Moreover, such an increase could even negatively impact computation time and model convergence during the backpropagation process [96]. Apart from the H_2 molecule, others can also be considered on the STO-3G basis such as lithium hydride, LiH , which can be represented using 4 qubits. The initial and final Hamiltonian operators of the process can be computed again using [70].

Discussing scalability within these methodologies holds substantial importance as it elucidates the extent to which an approach can be experimentally applicable. Undoubtedly, our DL-based methodology provides a wealth of information, as we have shown throughout the entire paper. However, when addressing the scalability of the method, we need an examination of two key factors: the number of qubits and the quantity of points encompassed within the time interval ($N_{\mathcal{F}}$), in conjunction with the graphics processing unit (GPU) or hardware employed, as a whole. All computations considered employed an NVIDIA A16 card with a memory capacity of 16 GB. Consequently, we present, in Figure 10 (top row), a comprehensive analysis of the final physical loss (or final total residual) which marks the conclusion of network training concerning the number of points within the interval $(t_{\text{min}}, t_{\text{max}})$ denoted as $N_{\mathcal{F}}$. This analysis encompasses diverse bond distances for the H_2 molecule (left) and a singular value of d for the LiH molecule (right), solely for the purpose of facilitating comparisons.

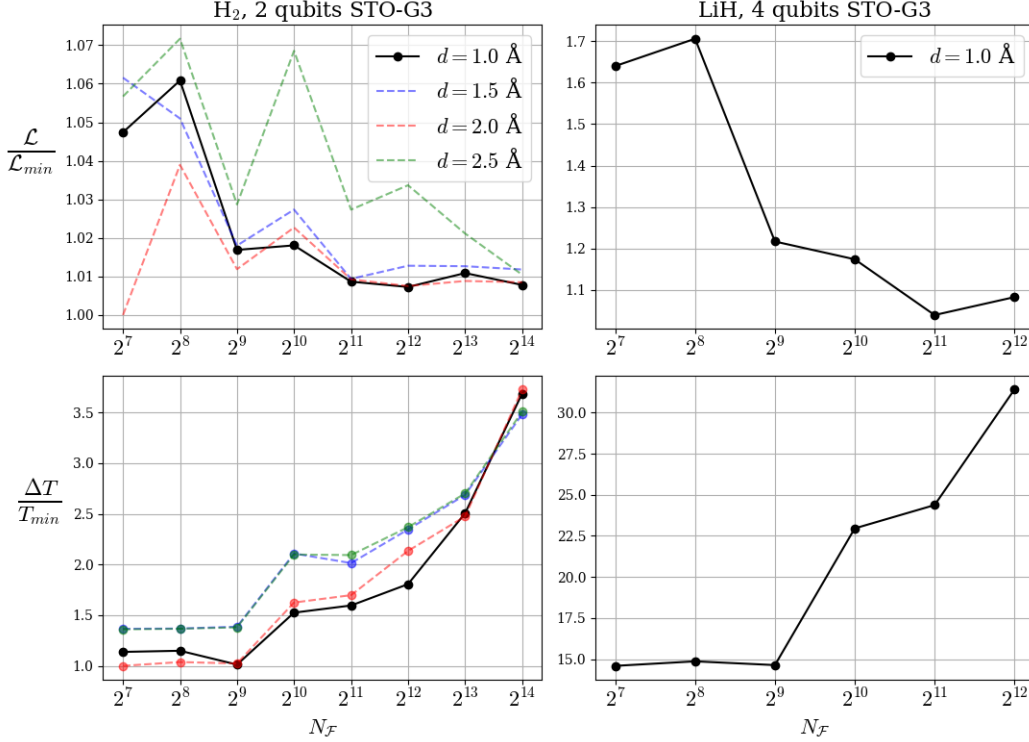


Figure 10: Graphical investigation of the scalability of our methodology for the H_2 molecule in the STO-3G basis. Various bond distances of the hydrogen atoms are considered, represented by distinct colors as indicated in the legend to the right. The top side of the graph illustrates the total physical loss \mathcal{L} (35) after completing the training process, plotted against the number of points $N_{\mathcal{F}}$ considered within the domain $t \in (t_{\min}, t_{\max})$. On the bottom side of the graph, we present the time required to complete the entire training process, normalized to the minimum time. We conducted these experiments using an NVIDIA A16 GPU.

The physical loss depicted in the top row of the figure has been normalized with respect to the minimum value observed for that particular magnitude across all training sessions considered in this section, encompassing both the H_2 and LiH simulations. This normalization, denoted as \mathcal{L}_{\min} , allows us to represent the loss in a manner independent of specific training instances. An analysis of the results reveals that the general loss for the LiH case is marginally higher, although the values are within the same order of magnitude as the H_2 case. It is worth noting that both simulations employed an identical neural architecture comprising 6 internal layers, each containing 30 neurons. Furthermore, a common training regimen of 500,000 iterations (epochs) was applied to all cases. Consequently, a longer training duration would likely result in reduced final physical errors, owing to the common training approach. The consistency in architecture and training duration enables us to draw meaningful comparisons between both simulations and enables us to fairly evaluate their respective final performances.

On the other side, for the purpose of effectively quantifying the computational time involved in various training sessions and facilitating a meaningful comparison, it is imperative to consider external factors that could impact the simulations. These factors include the specific GPU utilized, available free memory in the CPU (as some calculations are delegated to the CPU), among others. In the lower section of the figure, we present a comparative analysis between the two molecules concerning the time consumed (ΔT), which is also normalized with respect to the minimum value obtained for this metric, denoted as T_{\min} . The results demonstrate that an increase in the number of data points, $N_{\mathcal{F}}$, generally leads to a multiplication of approximately 3.5 times in the compute time consumed, when transitioning from 2^7 to 2^{14} data points, for the H_2 molecule. However, it is crucial to highlight that with only $N_{\mathcal{F}} = 2^{11}$ the final physical error \mathcal{L} obtained is minimal. This observation indicates that augmenting the sampled time domain does not necessarily enhance the performance of the model. In other words, with this particular number of data points in the training domain, the PINN exhibits significant capabilities in making inferences, extrapolating information to new temporal instances, and interpolating. Consequently, the adoption of 2^{11} data points implies a multiplicative factor of slightly more than 1.5 in comparison to the minimum elapsed computation time and it can be regarded as a favorable trade-off between training time and performance.

Concerning the training of the LiH molecule which involves 4 qubits, a significant increase in the required time is observed compared to the case of H_2 with 2 qubits. This discrepancy arises due to the consideration that the latter necessitates a decomposition of the potential gauge $\mathcal{A}_{CD}(t)$ comprising 16 possible combinations, each represented by a 4×4 matrix. However, in the case of the lithium hydride, there are 256 possible combinations in total, each represented by a matrix size of 16×16 . Consequently, both hardware memory and training time experience a considerable surge from one case study to another. It is important to note, however, that this increase in resources is essential for extracting all possible information from the problem. This includes the scheduling function, all components of the potential gauge at each time instant, as well as the instantaneous values of each coefficient of the decomposition. In practical situations, theoretical analysis often focuses on a subset of all the possible interactions of the qubits constituting the system. By reducing the number of interactions from 256 to a more manageable figure, the problem becomes more amenable to study. Under these circumstances, the primary contributors to the memory and computing time consumption are both $\mathcal{L}_{\text{Coupling}}$ (33) and $\mathcal{L}_{\text{Least Action}}$ (30). The former involves simultaneous manipulation of numerous matrices, while the latter involves performing two commutators. Moreover, both terms span $N_{\mathcal{F}}$ defined points in time, which further adds to the computational complexity.

5 Discussion and conclusions

In this study we have shown that deep learning methodologies such as Physics-Informed Neural Networks (PINNs) can be used to tackle the problem of counterdiabatic (CD) driving for analog quantum computation, deviating from conventional numerical methodologies [45]. Our proposed method is a generalized problem-independent methodology that offers a comprehensive solution, encompassing the determination of counterdiabatic terms and the temporal parametrization which empirically demonstrates that the adiabatic theory holds true. The suggested approach provides an unified and effective way to resolve the underlying physics of the problem while also giving the theoretical Pauli matrix decomposition needed to be implemented experimentally. Furthermore, using the CD approach allows to get a greater separation between the ground state, E_0 , and the first excited level, E_1 , throughout a significant portion of the temporal progression. This is a desirable property from an experimental perspective, as mentioned in Section 4.2.

Several questions may emerge from these findings. Firstly, an exploration of the computational capacity regarding the maximum qubit count achievable through PINN-based simulation is desirable, i.e., scalability. Secondly, there is an opportunity to enhance our methodology by integrating recent PINN advancements, including the incorporation of causality as an additional constraint and enabling the network to autonomously learn activation functions. Lastly, the restrictions of the theoretical Pauli matrix decomposition to encompass hardware-specific limitations introduce the prospect of improving the operational performance of analog quantum computers within experimental contexts.

Currently, all methodologies formulated within this context have been exclusively applied to systems comprising two and four qubits, as discussed in Section 4.4. Indeed, our study reveals that the training duration of a PINN for a 4-qubit system is approximately 15 times greater than that required for the 2-qubit counterpart. However, it is imperative to acknowledge that, empirically, the possible permutations of gates and qubit interconnections are usually restricted. Hence, despite the potential exponential increase in training time for an N-qubit system, the imposed experimental limitations render it feasible to effectively train the methodology for a substantial quantity of qubits.

Aside from reducing the problem, it is also possible to improve the overall performance of our proposal by forcing it to respect the causality principle [97] further to restrict the time evolution of our Hamiltonian operators. Using this approach is sufficient to achieve substantial improvements in terms of physical accuracy. Therefore, it is feasible to reduce the number of temporal points used during the training process without altering the network’s performance. Moreover, implementing dynamically trainable activation functions may help improve performance and convergence time [83].

Furthermore, the physical losses presented in Figure 2 are around 10^{-4} , thus underscoring the imperative to comprehend the attainable precision of a base PINN in the context of CD protocols. Enhancing the presented methodology could entail the imposition of temporal causal prerequisites and the optimization of the neural architecture. The incorporation of mixture coefficients (36) within the loss function is a predetermined selection based on inductive physical biases, thereby specifying greater weights for the initial and final loss components to steer the progression of the system. Other constituents associated with physical conditions encompass hyperparameters that can be selected through iterative experimentation. The said coefficients are amenable to alteration during the training process, potentially employing techniques such as the Augmented Lagrangian approach [62], which adapts them according to the deviation from each physical condition. Consequently, the presented approach offers opportunities for enhancing the achievements and mitigating physical losses, improving, if feasible, the robustness of the model from a physical perspective.

In conclusion, our work has shown that PINNs-based approaches are a promising methodology that can be effectively used to optimize CD protocols for adiabatic quantum computing. However, despite the substantial advancements

achieved within this context, it is evident that ample opportunities for enhancement exist. In particular, it would be worth studying the impact of these results in digitized-counterdiabatic quantum computing (DCQC) algorithms [98, 99]. The aforementioned questions stand as open paths for our future research, aiming to evolve and elaborate upon them.

Acknowledgements

The authors express their sincere appreciation for the thoughtful attention provided by the Kipu Quantum team. AFS and JDMG 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.

References

- [1] Jeremy Adcock, Euan Allen, Matt Day, Stefan Frick, Janna J Hinchliff, Mack Johnson, Sam Morley-Short, Sam Pallister, Alasdair Price, and Stasja Stanisic. Advances in quantum machine learning. *arXiv*, 1512.02900, 2015.
- [2] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [3] Vedran Dunjko, Jacob M. Taylor, and Hans J. Briegel. Quantum-enhanced machine learning. *Phys. Rev. Lett.*, 117:130501, 2016.
- [4] Rui Wang, Carlos Hernani-Morales, José D Martín-Guerrero, Enrique Solano, and Francisco Albarrán-Arriagada. Quantum pattern recognition in photonic circuits. *Quantum Science and Technology*, 7(1):015010, 2021.
- [5] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982.
- [6] I. M. Georgescu, S. Ashhab, and Franco Nori. Quantum simulation. *Reviews of Modern Physics*, 86(1):153–185, 2014.
- [7] Tad Hogg and Dmitriy Portnov. Quantum optimization. *Information Sciences*, 128(3):181–197, 2000.
- [8] Pranav Niroula, Rustem Shaydulin, and Ryan Yalovetzky, et al. Constrained quantum optimization for extractive summarization on a trapped-ion quantum computer. *Scientific Reports*, 12:17171, 2022.
- [9] Jigna B. Prajapati, Himanshu Paliwal, Bhupendra G. Prajapati, Surovi Saikia, and Rajiv Pandey. *Quantum Machine Learning in Prediction of Breast Cancer*, pages 351–382. Springer Nature Singapore, Singapore, 2023.
- [10] Yehui Tang, Junchi Yan, Guoqiang Hu, Baohua Zhang, and Jinzan Zhou. Recent progress and perspectives on quantum computing for finance. *Service Oriented Computing and Applications*, 16(4):227–229, 2022.
- [11] Sean J. Weinberg, Fabio Sanches, Takanori Ide, Kazumitsu Kamiya, and Randall Correll. Supply chain logistics with quantum and classical annealing algorithms. *Scientific Reports*, 13(1), 2023. Cited by: 2; All Open Access, Gold Open Access, Green Open Access.
- [12] Andris Ambainis and Martins Kokainis. Quantum algorithm for tree size estimation, with applications to backtracking and 2-player games. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 989–1002, New York, NY, USA, 2017. Association for Computing Machinery.
- [13] Seth Lloyd and Christian Weedbrook. Quantum generative adversarial learning. *Phys. Rev. Lett.*, 121:040502, 2018.
- [14] Sandip Dey, Siddhartha Bhattacharyya, and Ujjwal Maulik. New quantum inspired meta-heuristic techniques for multi-level colour image thresholding. *Applied Soft Computing Journal*, 46:677 – 702, 2016. Cited by: 50.
- [15] Frank Arute, Kunal Arya, and Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574:505–510, 2019.
- [16] Michael Brooks. Beyond quantum supremacy: The hunt for useful quantum computers. *Nature*, 574(7776):19–21, 2019-10-03.
- [17] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, 2014.

- [18] Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79:325–328, 1997.
- [19] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013.
- [20] Nathan Wiebe, Ashish Kapoor, and Krysta M. Svore. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Info. Comput.*, 15(3–4):316–356, mar 2015.
- [21] E.B. Fel’dman and A.I. Zenchuk. Nearest-neighbor approximation in one-excitation state evolution along spin-1/2 chain governed by xx-hamiltonian. *Physics Letters A*, 457:128572, 2023.
- [22] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Leo Zhou. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. *Quantum*, 6:759, 2022.
- [23] P. Chandarana, N. N. Hegade, K. Paul, F. Albarrán-Arriagada, E. Solano, A. del Campo, and Xi Chen. Digitized-counterdiabatic quantum approximate optimization algorithm. *Phys. Rev. Res.*, 4:013141, 2022.
- [24] Guido Pagano, Aniruddha Bapat, et al. Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator. *Proceedings of the National Academy of Sciences of the United States of America*, 117(41):25396–25401, 2020.
- [25] Chufan Lyu, Xusheng Xu, Man-Hong Yung, and Abolfazl Bayat. Symmetry enhanced variational quantum spin eigensolver. *Quantum*, 7:899, 2023.
- [26] Pietro Torta, Glen B. Mbeng, Carlo Baldassi, Riccardo Zecchina, and Giuseppe E. Santoro. Quantum approximate optimization algorithm applied to the binary perceptron. *Phys. Rev. B*, 107:094202, 2023.
- [27] Giuseppe E. Santoro and Erio Tosatti. Optimization using quantum mechanics: quantum annealing through adiabatic evolution. *Journal of Physics A: Mathematical and General*, 39(36):R393, 2006.
- [28] Tameem Albash and Daniel A. Lidar. Adiabatic quantum computation. *Rev. Mod. Phys.*, 90:015002, 2018.
- [29] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Dan Preda. A quantum adiabatic evolution algorithm applied to random instances of an NP-complete problem. *Science*, 292(5516):472–475, 2001. PMID: 11313487.
- [30] V. Bapst, L. Foini, F. Krzakala, G. Semerjian, and F. Zamponi. The quantum adiabatic algorithm applied to random optimization problems: The quantum spin glass perspective. *Physics Reports*, 523(3):127–205, 2013. The Quantum Adiabatic Algorithm Applied to Random Optimization Problems: The Quantum Spin Glass Perspective.
- [31] Jeremy Liu, Federico M. Spedalieri, Ke-Thia Yao, Thomas E. Potok, Catherine Schuman, Steven Young, Robert Patton, Garrett S. Rose, and Gangotree Chamka. Adiabatic quantum computation applied to deep learning networks. *Entropy*, 20(5), 2018.
- [32] M. Born and V. Fock. Beweis des adiabatensatzes. *Zeitschrift für Physik*, 51(3):165–180, 1928.
- [33] J. E. Avron, A. Raveh, and B. Zur. Adiabatic quantum transport in multiply connected systems. *Rev. Mod. Phys.*, 60:873–915, 1988.
- [34] D. Guéry-Odelin, A. Ruschhaupt, A. Kiely, E. Torrontegui, S. Martínez-Garaot, and J. G. Muga. Shortcuts to adiabaticity: Concepts, methods, and applications. *Rev. Mod. Phys.*, 91:045001, 2019.
- [35] Kazutaka Takahashi. Hamiltonian Engineering for Adiabatic Quantum Computation: Lessons from Shortcuts to Adiabaticity. *Journal of the Physical Society of Japan*, 88(6):061002, 2019.
- [36] Shumpei Masuda and Katsuhiko Nakamura. Fast-forward of adiabatic dynamics in quantum mechanics. *Proc. R. Soc. A.*, 466:1135–1154, 2010.
- [37] Iwan Setiawan, Bobby Eka Gunara, Sanat Avazbaev, and Katsuhiko Nakamura. Fast-forward approach to adiabatic quantum dynamics of regular spin clusters: Nature of geometry-dependent driving interactions. *Phys. Rev. A*, 99:062116, 2019.
- [38] Xi Chen, E. Torrontegui, and J. G. Muga. Lewis-riesenfeld invariants and transitionless quantum driving. *Phys. Rev. A*, 83:062116, 2011.
- [39] Xue-Ke Song, Hao Zhang, Qing Ai, Jing Qiu, and Fu-Guo Deng. Shortcuts to adiabatic holonomic quantum computation in decoherence-free subspace with transitionless quantum driving algorithm. *New Journal of Physics*, 18:023001, 2016.
- [40] Mustafa Demirplak and Stuart A. Rice. Assisted Adiabatic Passage Revisited. *Journal of Physical Chemistry B*, 109(14):6838–6844, 2005.

- [41] Takuya Hatomura. Shortcuts to Adiabaticity in the Infinite-Range Ising Model by Mean-Field Counter-Diabatic Driving. *Journal of the Physical Society of Japan*, 86(9):094002, 2017.
- [42] M V Berry. Transitionless quantum driving. *Journal of Physics A: Mathematical and Theoretical*, 42(36):365303, 2009.
- [43] N. N. Hegade, P. Chandarana, K. Paul, Xi Chen, F. Albarrán-Arriagada, and E. Solano. Portfolio optimization with digitized counterdiabatic quantum algorithms. *Phys. Rev. Res.*, 4:043204, 2022.
- [44] Christopher Jarzynski. Generating shortcuts to adiabaticity in quantum and classical dynamics. *Phys. Rev. A*, 88:040101, 2013.
- [45] Pieter W. Claeys, Mohit Pandey, Dries Sels, and Anatoli Polkovnikov. Floquet-engineering counterdiabatic protocols in quantum many-body systems. *Phys. Rev. Lett.*, 123:090602, 2019.
- [46] Narendra N. Hegade, Xi Chen, and Enrique Solano. Digitized counterdiabatic quantum optimization. *Phys. Rev. Res.*, 4:L042030, 2022.
- [47] Diego Andina, Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018:7068349, 2018.
- [48] Ronan Collobert, Jason Weston, Léon Bottou, et al. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493 – 2537, 2011. Cited by: 5478.
- [49] Hassan Ismail Fawaz, Germain Forestier, Jonathan Weber, Lhassane Idoumghar, and Pierre-Alain Muller. Deep learning for time series classification: a review. *Data Mining and Knowledge Discovery*, 33(4):917–963, 2019.
- [50] Robin M. Schmidt. Recurrent Neural Networks (RNNs): A gentle Introduction and Overview. 2019.
- [51] Sepp Hochreiter and Jürgen Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [53] Benjamin Graham and Laurens van der Maaten. Submanifold sparse convolutional networks, 2017.
- [54] Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. 2017.
- [55] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [56] Zhiping Mao, Ameya D. Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.
- [57] Antonio Ferrer-Sánchez, José D. Martín-Guerrero, Roberto Ruiz de Austri, Alejandro Torres-Forné, and José A. Font. Gradient-annihilated pinns for solving riemann problems: Application to relativistic hydrodynamics, 2023.
- [58] Jianxin Xie and Bing Yao. Physics-constrained deep active learning for spatiotemporal modeling of cardiac electrodynamics. *Computers in Biology and Medicine*, 146:105586, 2022.
- [59] Weiqi Ji, Weilun Qiu, Zhiyu Shi, Shaowu Pan, and Sili Deng. Stiff-pinn: Physics-informed neural network for stiff chemical kinetics. *The Journal of Physical Chemistry A*, 125(36):8098–8106, 2021. PMID: 34463510.
- [60] Atilim Gunes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. 2015.
- [61] Lu Lu, Raphaël Pestourie, Wenjie Yao, Zhicheng Wang, Francesc Verdugo, and Steven G. Johnson. Physics-Informed Neural Networks with Hard Constraints for Inverse Design. *SIAM Journal on Scientific Computing*, 43(6):B1105–B1132, 2021.
- [62] Hwijae Son, Sung Woong Cho, and Hyung Ju Hwang. Enhanced physics-informed neural networks with augmented lagrangian relaxation method (al-pinns). *Neurocomputing*, 548:126424, 2023.
- [63] Dan Sun, Pranav Chandarana, Zi-Hua Xin, and Xi Chen. Optimizing counterdiabaticity by variational quantum circuits. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 380(2239), 2022.

- [64] Dries Sels and Anatoli Polkovnikov. Minimizing irreversible losses in quantum systems by local counterdiabatic driving. *Proceedings of the National Academy of Sciences of the United States of America*, 114(20):E3909–E3916, 2017.
- [65] Tosio Kato. On the Adiabatic Theorem of Quantum Mechanics. *Journal of the Physical Society of Japan*, 5(6):435, 1950.
- [66] G Nenciu. On the adiabatic theorem of quantum mechanics. *Journal of Physics A: Mathematical and General*, 13(2):L15, 1980.
- [67] Ieva Čepaitė, Anatoli Polkovnikov, Andrew J. Daley, and Callum W. Duncan. Counterdiabatic optimized local driving. *PRX Quantum*, 4:010312, 2023.
- [68] Michael Kolodrubetz, Dries Sels, Pankaj Mehta, and Anatoli Polkovnikov. Geometry and non-adiabatic response in quantum and classical systems. *Physics Reports*, 697:1–87, 2017. Geometry and non-adiabatic response in quantum and classical systems.
- [69] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, 1982.
- [70] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.
- [71] Noson S. Yanofsky and Mirco A. Mannucci. *Quantum Computing for Computer Scientists*. Cambridge University Press, 2008.
- [72] Eleanor G. Rieffel and Wolfgang H. Polak. *Quantum Computing: A Gentle Introduction*. The MIT Press, 2011.
- [73] Sebastian Deffner, Christopher Jarzynski, and Adolfo del Campo. Classical and quantum shortcuts to adiabaticity for scale-invariant driving. *Phys. Rev. X*, 4:021013, 2014.
- [74] Analia Zwick, Gonzalo A Álvarez, Guy Bensky, and Gershon Kurizki. Optimized dynamical control of state transfer through noisy spin chains. *New Journal of Physics*, 16(6):065021, 2014.
- [75] Ryuji Takahashi and Naoto Nagaosa. Berry curvature in magnon-phonon hybrid systems. *Phys. Rev. Lett.*, 117:217205, 2016.
- [76] Pengpeng Diao, Shujin Deng, Fang Li, Shi Yu, Aurélia Chenu, Adolfo del Campo, and Haibin Wu. Shortcuts to adiabaticity in fermi gases. *New Journal of Physics*, 20(10):105004, 2018.
- [77] Shumpei Masuda and Katsuhiko Nakamura. Fast-forward problem in quantum mechanics. *Phys. Rev. A*, 78:062108, 2008.
- [78] E. Torrontegui, S. Martínez-Garaot, A. Ruschhaupt, and J. G. Muga. Shortcuts to adiabaticity: Fast-forward approach. *Phys. Rev. A*, 86:013601, 2012.
- [79] J. Wurtz and P.J. Love. Counter-diabaticity and the quantum approximate optimization algorithm. *Quantum*, 6:635, 2022.
- [80] Yahui Chai, Yong-Jian Han, Yu-Chun Wu, Ye Li, Menghan Dou, and Guo-Ping Guo. Shortcuts to the quantum approximate optimization algorithm. *Phys. Rev. A*, 105:042415, 2022.
- [81] Narendra N. Hegade, Koushik Paul, F. Albarrán-Arriagada, Xi Chen, and Enrique Solano. Digitized adiabatic quantum factorization. *Phys. Rev. A*, 104:L050403, 2021.
- [82] Pranav Chandarana, Narendra N. Hegade, Iraitz Montalban, Enrique Solano, and Xi Chen. Digitized counterdiabatic quantum algorithm for protein folding. *Phys. Rev. Appl.*, 20:014024, 2023.
- [83] Ameya D. Jagtap, Kenji Kawaguchi, and George Em Karniadakis. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. *Journal of Computational Physics*, 404:109136, 2020.
- [84] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [85] Attila Szabo and Neil S. Ostlund. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory*. Dover Publications, 1996.
- [86] Wei-Liem Loh. On latin hypercube sampling. *The Annals of Statistics*, 24(5):2058–2080, 1996.
- [87] Richard J. Beckman and Michael D. McKay. Monte carlo estimation under different distributions using the same simulation. *Technometrics*, 29(2):153–160, 1987.
- [88] I.M Sobol’. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86–112, 1967.

- [89] Chenxi Wu, Min Zhu, Qinyang Tan, Yadhu Kartha, and Lu Lu. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 403:115671, 2023.
- [90] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterton, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 2010. PMLR.
- [91] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, H. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. PyTorch. <https://pytorch.org>, 2023.
- [92] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv:1412.6980*, 2014.
- [93] Ward Cheney and David Kincaid. *Numerical Mathematics and Computing*. Cengage Learning, 6th edition, 2007.
- [94] James Brown and Ruel Churchill. *Fourier Series and Boundary Value Problems*. McGraw-Hill, 8th edition, 2011.
- [95] Jeremy Yu, Lu Lu, Xuhui Meng, and George Em Karniadakis. Gradient-enhanced physics-informed neural networks for forward and inverse pde problems. *Computer Methods in Applied Mechanics and Engineering*, 393:114823, 2022.
- [96] Panos Pantidis, Habiba Eldababy, Christopher Miguel Tagle, and Mostafa E. Mobasher. Error convergence and engineering-guided hyperparameter search of pinns: Towards optimized i-fenn performance. *Computer Methods in Applied Mechanics and Engineering*, 414:116160, 2023.
- [97] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks, 2022.
- [98] Narendra N Hegade, Koushik Paul, Yongcheng Ding, Mikel Sanz, Francisco Albarrán-Arriagada, Enrique Solano, and Xi Chen. Shortcuts to adiabaticity in digitized adiabatic quantum computing. *Physical Review Applied*, 15(2):024038, 2021.
- [99] Alejandro Gomez Cadavid, Iraitz Montalban, Archismita Dalal, Enrique Solano, and Narendra N Hegade. Efficient dcqo algorithm within the impulse regime for portfolio optimization. *arXiv preprint arXiv:2308.15475*, 2023.