Solving the Time Dependent Schrödinger Equation using Machine Learning

Author: Elizabeth Birch Hardwick, ebirchha7@alumnes.ub.edu
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.

Advisor: Arnau Ríos Huguet, arnau.rios@ub.edu

Abstract: The time-dependent Schrödinger equation plays a central role in quantum physics, yet the methods used to solve it are typically computationally expensive. In this work, we use a Physics-Informed Neural Network approach to learn the dynamics of the quantum harmonic oscillator. Our model successfully reproduces the expected oscillatory motion of the coherent state and conserves energy with only very small deviations, with relative energy errors below 10^{-3}. The method achieves extremely low infidelities with respect to the analytical results, of the order of 10^{-5}. We also test the model on breathing mode dynamics, obtaining a low average infidelity of the order of 10^{-2} and a modest relative energy error around 10^{-2}. These results show that Physics-Informed Neural Networks can accurately learn and generalise solutions to the time-dependent Schrödinger equation, providing an efficient alternative to traditional solvers.

Keywords: Quantum physics, Time-dependent Schrödinger equation, Machine learning.
SDGs: This work is aligned with the UN Sustainable Development Goals 9, 4 and 13.

I. INTRODUCTION

Over the last few years we have seen a massive increase in the use of Artificial Intelligence, leading to applications in various fields in science to solve expensive computational tasks. In physics, one standout application are Physics-Informed Neural Networks (PINNs) [1], which solve partial differential equations (PDEs) whilst taking the physics of the problem at hand into account. There have been several successful applications of PINNs to problems such as the incompressible Navier-Stokes equations [2], the convection-diffusion equation [3], and wave propagation in acoustic media [4].

The time-dependent Schrödinger equation (TDSE) is fundamental in understanding the evolution of quantum systems and, while there are analytical solutions for a few simple cases [5], most physically relevant systems require numerical methods which come at a computational cost. PINNs have recently emerged as a powerful tool to address this issue [6].

In this project, we develop a PINN to solve the TDSE. The latter takes the form:

$$i\hbar \frac{\partial \psi}{\partial t} = \hat{H}\psi, \tag{1}$$

where $\hat{H}$ is the Hamiltonian operator which depends on the system that is being studied, ψ is the wavefunction, $\hbar$ is the reduced Planck constant, and t is time.

In Section II of this document we introduce the structure of PINNs. In Section III we explain the computational setup, including the network architecture and loss function design. In Section IVA we resolve the dynamics of the coherent state. Then, in Section IVB we analyse the impact of temporal resolution. In Section IVC we extend our results to the dynamics of the breathing mode. Finally, in Section V we present the conclusions and discuss future perspectives.

II. PHYSICS-INFORMED NEURAL NETWORKS

Artificial Neural Networks (ANNs) are computational models capable of approximating complex functions through a structured arrangement of interconnected layers. A single-layer ANN with one input (x) and one output ($f(x)$) is given by:

$$f(x) = \sum_{i=1}^{N_{\text{hid}}} W_i^{(2)}\, \sigma\left(W_i^{(1)} x + B_i\right). \tag{2}$$

where $W^{(1)}$, $W^{(2)}$, and B are the weights and bias, respectively, N_{hid} is the number of neurons in the hidden layer and σ is the activation function, i.e., a typically non-linear function applied elementwise to introduce nonlinearity in the network.

The *Universal Approximation Theorem* [7] states that, if N_{hid} is large enough, a single-layer ANN can approximate any continuous function. In order to best approximate the desired function we must find the correct weight and bias by training our neural network.

A single training iteration proceeds as follows: each input x from a chosen domain is first fed into the input layer. This data is then propagated forward through the network, layer by layer, and evaluated at each neuron using the affine transformation defined in Eq. (2). The final layer's output is used to compute a loss function, which quantifies how well the network satisfies the target constraint – in our case, the governing physical equation. The loss is then passed through an optimiser which backpropagates, i.e., calculates how the loss function changes with respect to each parameter and updates the weights and biases accordingly by an amount proportional to a learning rate.

Once the weights and biases are updated, the loss function is computed again and the process is repeated iteratively for a given number of epochs or until the network's prediction is converged.

In this project we will be using PINNs, which are a class of ANN that solve problems that deal with PDEs. To solve them we encode the constraints given by the PDE and its boundary conditions into the loss function.

III. COMPUTATIONAL SET-UP

For the dynamics we use the case of the quantum harmonic oscillator (QHO) as the analytical results are well known:

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + \frac{1}{2}m\omega^2 x^2, \tag{3}$$

where m is the mass of the particle and ω the trapping frequency. In what follows, we work in oscillator units of length and energy $x_{\text{ho}} = \sqrt{\frac{\hbar}{m\omega}}$, $E_{\text{ho}} = \hbar\omega$, respectively.

To ensure our PINN operates in $\mathbb{R}$, and to capture the correct physical dynamics, we represent the complex wavefunction using its real-valued logarithmic magnitude $\ln|\psi(x,t)|$ and phase $\theta(x,t)$. The input consists of the discrete spatial and temporal points $x_j \in [-4,4]$ and $t_k \in [0,2\pi]$, forming a spatial-temporal grid of shape $N_x \times N_t$. Then, the wavefunction is reconstructed as:

$$\psi(x,t) = e^{\ln|\psi(x,t)|} e^{i\theta(x,t)}. \tag{4}$$

In the discrete setting, we evaluate the wavefunction at points (x_j, t_k) and denote it as $\psi_{jk} \equiv \psi(x_j, t_k)$, with analogous expressions for the magnitude and phase.

Defining $\vec{r} = \begin{pmatrix} x \\ t \end{pmatrix}$, $\varphi_l(\vec{r}) = \begin{pmatrix} \ln|\psi(x,t)| \\ \arg(\psi(x,t)) \end{pmatrix}$ and $\varphi^{(0)}(\vec{r}) = \vec{r}$ we can express our two-input and two-output PINN with $\mathcal{M}-1$ hidden layers as:

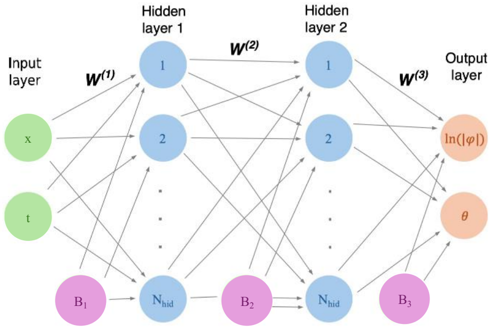

FIG. 1: Architecture of the neural network with 2 inputs (the spatial and temporal coordinates x and t) in green, 2 hidden layers with N_{hid} neurons each in blue and 2 outputs (the logarithm of the wavefunction's magnitude and its phase) in orange. We represent the bias vectors for each layer in purple.

$$\varphi_l^{(M)}(\vec{r}) = \sum_{k=1}^{N_{\text{hid}}^{(M)}} W_{lk}^{(M)}\,\sigma\left(\varphi_k^{(M-1)}(\vec{r})\right) + b_k^{(M)}, \tag{5}$$

where $W_{lk}^{(M)}$ and $b_i^{(M)}$ are elements of the weight matrix and bias vector of the layer M, respectively, $N_{\text{hid}}^{(M)}$ is the number of neurons in the layer M and σ is the activation function.

Our PINN loss function is defined as the squared norm of the residual of the TDSE:

$$\mathcal{L}_{\text{TDSE}} = \sum_{j=0}^{N_x}\sum_{k=0}^{N_t}\left| i\frac{\partial\psi_{jk}}{\partial t} - \hat{H}\psi_{jk}\right|^2. \tag{6}$$

To encode the physics into our PINN loss we add the initial and boundary conditions as Lagrangian multipliers, where $\mathcal{L}_{\text{edges}}$ is our boundary condition that imposes that the wavefunction at the edges goes to zero:

$$\mathcal{L} = \mathcal{L}_{\text{TDSE}} + \lambda_1\mathcal{L}_{\text{ini}} + \lambda_2\mathcal{L}_{\text{final}} + \lambda_3\mathcal{L}_{\text{edges}}. \tag{7}$$

An example evolution of the loss is illustrated in Fig. 2.

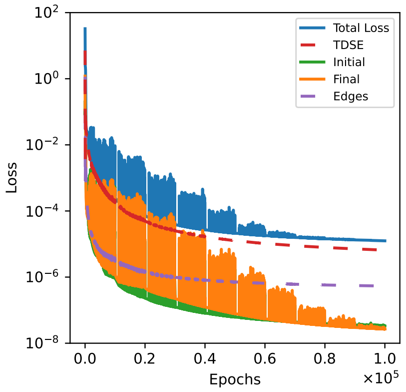

FIG. 2: Evolution of the total loss (blue) and its components during PINN training. The loss includes contributions from the TDSE residual (red dashed), initial (green) and final (orange) conditions, and boundary (purple dashed) constraints.

IV. RESULTS

A. Coherent state

The coherent state [8] corresponds to a special case of the QHO in which the centre of the analytic solution to the QHO – the Gaussian wavepacket – follows an oscillatory motion of amplitude $x(t=0) = x_0$:

$$\ln|\psi(x,t)| = -\frac{1}{4}\ln\pi - \frac{1}{2}\left(x - \langle x(t)\rangle\right)^2, \tag{8}$$

$$\theta(x,t) = \langle p(t)\rangle x - \frac{1}{2}\langle x(t)\rangle\langle p(t)\rangle - \frac{t}{2}. \tag{9}$$

where $\langle x(t)\rangle = x_0 \cos t$ and $\langle p(t)\rangle = -x_0 \sin t$ for $\langle p(t=0)\rangle = 0$.

We generate these states by setting the initial wavefunction as a Gaussian ground state of the QHO displaced by $x_0 = 0.5$:

$$\mathcal{L}_{\text{ini}} = \sum_{i=0}^{N_x} \left| \psi_{i,j=0} - \pi^{-1/4} \exp\left(-\frac{(x_i - x_0)^2}{2}\right) \right|^2 .$$

Since our Hamiltonian is periodic, for further stability, we also apply a final condition:

$$\mathcal{L}_{\text{final}} = \sum_{i=0}^{N_x} \left| \psi_{i,j=N_t} + \pi^{-1/4} \exp\left(-\frac{(x_i - x_0)^2}{2}\right) \right|^2 .$$

We trained the PINN using the architecture depicted in Fig. 1 with two hidden layers of $N_{\text{hid}} = 40$ neurons and loss weights $\lambda_1 = \lambda_2 = \lambda_3 = 10$. We used a `sigmoid` activation function between the first and second layers, and a `tanh` between the second and output layers. The training ran for 10^4 epochs using the Adam optimiser [9].

To speed up convergence, we applied a StepLR scheduler with initial learning rate $lr = 0.01$, halved every 10^3 epochs. Despite this, training took $t = 572.39$ minutes ≈ 9 hours for $N_t = N_x = 100$ on a single CPU.

To assess our PINN's performance, Fig. 3 shows the evolution of the kinetic and potential energy components, along with the expectation values $\langle x\rangle$ and $\langle p\rangle$. Solid lines indicate predictions and dashed lines analytic solutions.

In Fig. 3(a), the energy components oscillate sinusoidally out of phase. Predicted values closely follow the analytic curves, preserving the correct phase. In Fig. 3(b), $\langle x\rangle$ and $\langle p\rangle$ oscillate oppositely. The trajectories match the analytic evolution, capturing the correct coherent state behaviour.

Finally, we examine the wavefunction to verify coherent-state behaviour. In Fig. 4 we depict the real (a), imaginary (b), and squared absolute (c) components of the predicted wavefunction, along with the absolute error (d) compared to the exact solution, all plotted over position x and time t.

In Fig. 4(a), the real part oscillates spatially, switching from positive to negative halfway through the cycle. The imaginary part in Fig. 4(b) also oscillates but remains negative throughout. The squared absolute value in Fig. 4(c) retains a Gaussian shape, tracing periodic motion consistent with an initial displacement of 0.5 h.o. The absolute error plot in Fig. 4(d) confirms that the PINN prediction closely matches the analytical solution, with errors below 3×10^{-3} across the full domain.

These results show that the PINN captures both amplitude and phase evolution accurately, with minimal deviation from the exact solution. To further confirm this, we compute the average infidelity, defined as one minus the mean value of the overlap between the predicted wavefunction and the exact one at each time step t_i over the full evolution: $1 - \frac{1}{N_t} \sum_{i=0}^{N_t} \langle \phi^{\text{exact}}(t_i) | \phi_{\text{PINN}}(t_i) \rangle = 9.5 \times 10^{-6}$.

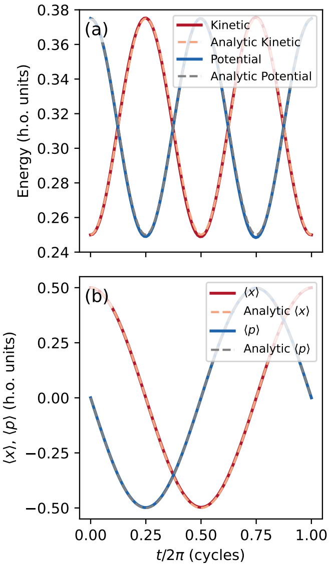

FIG. 3: (a): Temporal evolution of the kinetic (red) and potential (blue) energy components compared with analytic results (dashed). (b): Expectation values $\langle x\rangle$ (red) and $\langle p\rangle$ (blue) compared with analytic trajectories (dashed).

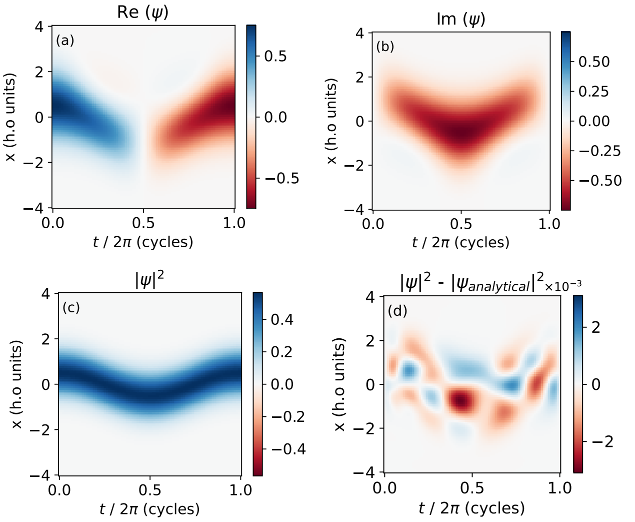

FIG. 4: (a): Real part of the wavefunction as a function of space and time. (b): Imaginary part of the wavefunction. (c): Squared absolute value of the wavefunction. (d): Absolute error between the predicted and exact solutions.

B. Temporal resolution

In order to understand how the temporal resolution affects the performance of our model, we trained several networks using different values of N_t. To ensure a consistent comparison and to test the generalisation ability of each model, we performed a post-training evaluation of all models on the same uniform grid of $N_t = 100$ points. This setup allows us to verify whether models trained on coarser grids can still produce accurate results at intermediate time steps where they were not explicitly trained on.

TABLE I: Summary of performance metrics for different temporal resolutions N_t used during training.

N_t	**Time (min)**	**Infidelity**	**Avg. Energy Err. (%)**
10	99.42	2.13×10^{-5}	0.24
25	144.23	6.1×10^{-6}	0.10
50	303.67	1.11×10^{-5}	0.11
75	402.50	9.5×10^{-6}	0.12
100	572.39	9.5×10^{-6}	0.13

The results are presented in Table I. All tested values of N_t reproduce the dynamics with very high accuracy, with average infidelities below 3×10^{-5} and relative energy errors below 0.24%. Optimal results are obtained for $N_t = 25$, where the average infidelity drops to 6.1×10^{-6} and the mean relative error falls below 0.11%. Beyond this point, increasing the resolution does not improve accuracy. However, reducing N_t to 10 doubles the relative error. This implies a limit to how coarse the temporal resolution can be before the accuracy is significantly affected.

To get a more complete picture, we also examined how the total energy evolves over time for each value of N_t. Fig. 5 shows the predicted total energy for different N_t, each represented by a distinct colour and line style. In all cases, the network conserves energy reasonably well, with a maximum deviation of under 0.45% for $N_t = 10$. This improves for $N_t \geq 25$, where the variation stays below 0.25%. The results for $N_t = 50$, 75, and 100 are practically indistinguishable at the plot's scale, suggesting a well-defined energy profile that the model approximates more closely as temporal resolution increases. We therefore conclude that any $N_t \geq 50$ provides sufficiently accurate and stable results, and further increasing the resolution leads to negligible improvement.

C. Breathing Mode Dynamics

To ensure that the network is not simply overfitting to the coherent state dynamics, we tested its performance on the breathing mode dynamics [10]. In this setup, we use a non-displaced initial condition corresponding to the ground state of a QHO with $\omega = 1$, but evolve it under a modified potential with reduced frequency $\omega = 0.5$. This induces width oscillations of the wavefunction while remaining centred at $x = 0$, producing a smooth breathing dynamic:

$$\ln|\psi(x,t)| = -\frac{1}{4}\ln\pi - \frac{1}{2}\ln s(t) - \frac{x^2}{2s(t)^2}, \tag{10}$$

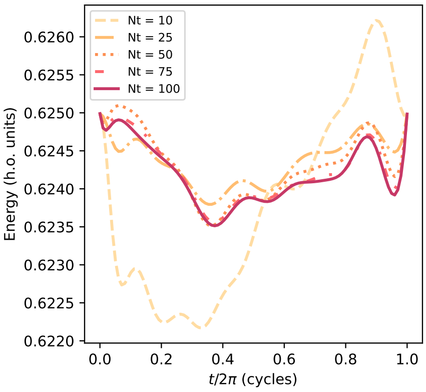

FIG. 5: Comparison of the predicted total energy over time for different values of N_t.

$$\theta(x,t) = -\frac{1}{2}\arctan\left(\omega\tan(\omega t)\right) - \frac{\omega t}{2}, \tag{11}$$

where $s(t) = \sqrt{\cos^2(\omega t) + \frac{1}{\omega^2}\sin^2(\omega t)}$.

This time, we have purposely omitted the final time condition to see whether the network can learn this more symmetric dynamic with fewer constraints provided.

To evaluate the breathing mode dynamics, we first examine the evolution of the energy components in Fig. 6. The total energy (purple) remains relatively stable with a slight downward trend, while the kinetic (red) and potential (blue) energies oscillate out of phase, as expected. Each component closely follows its analytical counterpart (dashed lines), though small deviations gradually increase over time. Despite that, the mean relative error in energy remains low at 1.07%.

Then in Fig. 7, we inspect the wavefunction itself. In Fig. 7(a), the squared absolute value of the wavefunction is plotted as a function of space and time. The profile remains Gaussian-shaped per time, periodically expanding and contracting in a symmetric breathing motion centered around $x_0 = 0$. This behaviour aligns with the expected dynamics. In Fig. 7(b) we depict the absolute error between the predicted and analytical wavefunctions, with most of the error at the end of the cycle. The error remains below 1×10^{-2} across the entire spatiotemporal range.

Although performance decreases towards the end of the cycle – consistent with the absence of a final condition – the average infidelity remains low at 3.2×10^{-2}, showing that the PINN still learns the essential dynamics accurately. This is slightly lower than in the coherent case with a final condition, but significantly better than the coherent case without one, where the network fails to reproduce the correct dynamics.

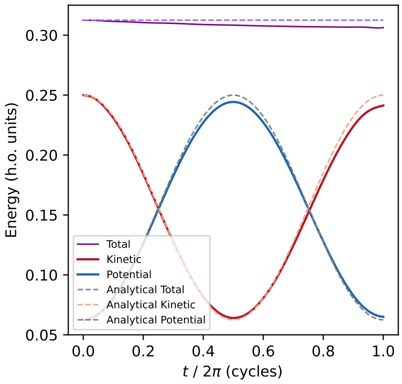

FIG. 6: Evolution of the total (purple), kinetic (red), and potential (blue) energy during breathing mode dynamics with their corresponding analytical results (dashed).

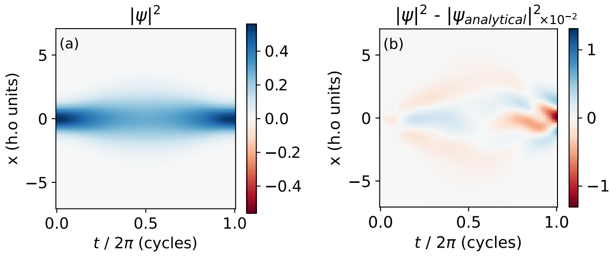

FIG. 7: (a): Squared absolute value of the wavefunction as a function of space and time. (b): Absolute error between the predicted and exact solutions.

V. CONCLUSIONS

We have developed a PINN that solves the TDSE for a QHO. We encoded the full PDE together with initial, final, and boundary conditions into the loss function, and trained our network using two hidden layers of 40 neurons each.

We have analysed the dynamics for the coherent state, showing that our model captures the evolution of expectation values $\langle x \rangle$, $\langle p \rangle$, and the wavefunction with high accuracy achieving an average infidelity of 9.5×10^{-6} with the analytic wavefunction and a mean relative energy error below 0.13%. The maximum energy deviation remains under 0.24%, demonstrating good energy conservation.

We conducted a detailed analysis of the effect of the temporal resolution N_t, finding that any value $N_t \geq 50$ provides sufficiently accurate results, with diminishing returns beyond this number.

To test generalisation further, we removed the final condition and studied the breathing mode. The network is able to reproduce the correct dynamics with an average infidelity of 3.2×10^{-2} and a relative energy error of 1.07%.

Overall, our results confirm that PINNs can effectively learn and generalise solutions of the TDSE which might offer a lower computational cost alternative to traditional solvers.

All the codes for the project can be found at [11].

Acknowledgments

I want to express gratitude to my advisor, Dr. Arnau Ríos, for all the help with this project. I also want to thank Javier Rozalén and Alejandro Romero for their useful comments during our meetings and my friends, family and partner for their support during the whole work.

---

[1] M. Raissi et al. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.*, 378:686–707, 2019.

[2] X. Jin et al. NSFnets (Navier–Stokes Flow nets): Physics-informed neural networks for the incompressible Navier–Stokes equations. *arXiv preprint arXiv:2003.06496*, 2020.

[3] A. Patel et al. Transformed physics-informed neural networks for the convection–diffusion equation. *arXiv preprint arXiv:2402.07671*, 2024.

[4] M. Rasht-Behesht et al. Physics-informed neural networks (pinns) for wave propagation and full waveform inversions. *J. Geophys. Res. Solid Earth*, 127(1):e2021JB023120, 2022.

[5] M. Fernández Guasti and H. Moya-Cessa. Solution of the Schrödinger equation for time-dependent 1D harmonic oscillators using the orthogonal functions invariant. *J. Phys. A*, 36(8):2069–2078, 2003.

[6] K. Shah et al. Physics-Informed Neural Networks as Solvers for the Time-Dependent Schrödinger Equation. *arXiv preprint arXiv:2210.12522*, 2022.

[7] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Math. Control Signals Syst.*, 2(4):303–314, 1989.

[8] C. Cohen-Tannoudji, B. Diu, and F. Laloë. *Quantum Mechanics, Volume 1: Basic Concepts, Tools, and Applications.* Wiley-VCH, 2019.

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

[10] H. R. Lewis and W. B. Riesenfeld. An exact quantum theory of the time-dependent harmonic oscillator. *J. Math. Phys.*, 10(8):1458–1473, 1969.

[11] E. Birch. TDSE_ML: Code repository for the project. `https://github.com/elizabethbirchhardwick/TDSE_ML.git`, 2025.

Resolent l'Equació de Schrödinger Depenent del Temps mitjançant Aprenentatge Automàtic.

Author: Elizabeth Birch Hardwick, ebirchha7@alumnes.ub.edu
Facultat de Física, Universitat de Barcelona, Diagonal 645, 08028 Barcelona, Spain.

Advisor: Arnau Ríos Huguet, arnau.rios@ub.edu

Resum: L'equació de Schrödinger dependent del temps juga un paper clau en la física quàntica, però els mètodes habituals per resoldre-la són computacionalment costosos. En aquest treball, utilitzem xarxes neuronals informades per la física (Physics-Informed Neural Networks) per aprendre la dinàmica de l'oscil·lador harmònic quàntic. El nostre model reprodueix amb èxit el moviment oscil·latori esperat de l'estat coherent i conserva l'energia amb desviacions molt petites, amb errors relatius d'energia per sota de 10^{-3}. El mètode aconsegueix valors d'infidelitat molt baixos respecte als resultats analítics, de l'ordre de 10^{-5}. També provem el model amb la dinàmica del mode de respiració, obtenint una infidelitat mitjana baixa de l'ordre de 10^{-2} i un error relatiu d'energia al voltant de 10^{-2}. Aquests resultats demostren que les xarxes neuronals informades per la física poden aprendre i generalitzar amb precisió solucions de l'equació de Schrödinger dependent del temps, oferint una alternativa eficient als mètodes tradicionals.

Paraules clau: Física quàntica, Equació de Schrödinger dependent del temps, Aprenentatge automàtic

ODSs: Aquest treball s'alinea amb els ODSs de l'ONU 9, 4 i 13.

Objectius de Desenvolupament Sostenible (ODSs o SDGs)

1. Fi de la es desigualtats		10. Reducció de les desigualtats	
2. Fam zero		11. Ciutats i comunitats sostenibles	
3. Salut i benestar		12. Consum i producció responsables	
4. Educació de qualitat	X	13. Acció climàtica	X
5. Igualtat de gènere		14. Vida submarina	
6. Aigua neta i sanejament		15. Vida terrestre	
7. Energia neta i sostenible		16. Pau, justícia i institucions sòlides	
8. Treball digne i creixement econòmic		17. Aliança pels objectius	
9. Indústria, innovació, infraestructures	X		

El contingut d'aquest TFG es relaciona amb l'ODS 9 (indústria, innovació i infraestructures), i en particular amb la fita 9.5, ja que contribueix a la recerca i desenvolupament tecnològic mitjançant l'ús de xarxes neuronals per resoldre equacions diferencials en física quàntica, millorant així les eines computacionals del camp. També es pot vincular amb l'ODS 4 (educació de qualitat), fita 4.4, per la seva contribució al desenvolupament de competències avançades en intel·ligència artificial i computació científica, fonamentals per a l'ocupació i la innovació. Finalment, es pot associar amb l'ODS 13 (acció pel clima), fita 13.3, ja que l'ús de mètodes més eficients pot reduir el consum de recursos computacionals, afavorint una recerca més sostenible des del punt de vista energètic.