

Physics-informed neural networks for quantum control

Ariel Norambuena,^{1,*} Marios Mattheakis,² Francisco J. González,³ and Raúl Coto^{3,4,†}

¹*Centro de Optica e Información Cuántica, Universidad Mayor, camino la Piramide 5750, Huechuraba, Santiago, Chile*

²*John A. Paulson School of Engineering and Applied Sciences, Harvard University, Cambridge, Massachusetts 02138, USA*

³*Centro de Investigación DAiTA Lab, Facultad de Estudios Interdisciplinarios, Universidad Mayor, Chile*

⁴*Department of Physics, Florida International University, Miami, Florida 33199, USA*

(Dated: December 11, 2023)

Quantum control is a ubiquitous research field that has enabled physicists to delve into the dynamics and features of quantum systems, delivering powerful applications for various atomic, optical, mechanical, and solid-state systems. In recent years, traditional control techniques based on optimization processes have been translated into efficient artificial intelligence algorithms. Here, we introduce a computational method for optimal quantum control problems via physics-informed neural networks (PINNs). We apply our methodology to open quantum systems by efficiently solving the state-to-state transfer problem with high probabilities, short-time evolution, and using low-energy consumption controls. Furthermore, we illustrate the flexibility of PINNs to solve the same problem under changes in physical parameters and initial conditions, showing advantages in comparison with standard control techniques.

Optimal Quantum Control (QC) is crucial to exploit all the advantages of quantum systems, ranging from entangled states preparation and quantum registers to quantum sensing. Nowadays, QC can be achieved by means of controllable dissipative dynamics [1, 2], measurement-induced backaction [3–5], Lyapunov control [6, 7], optimal pulse sequences [8], and differentiable programming [9, 10]. These QC techniques serve multiple purposes including state preservation, state-to-state transfer [11], dynamical decoupling in open systems [12–14] and trajectory tracking [15, 16]. Furthermore, we have witnessed powerful applications across multiple platforms, including atomic systems [17, 18], light-matter systems [19, 20], solid-state devices [21, 22], trapped ions [23], among others. Dynamical QC stems from a time-dependent Hamiltonian that steers the dynamics [24], and it is subjected to several constraints like laser power, inhomogeneous frequency broadening, and relaxation processes, to name a few. Therefore, finding the optimal sequence for QC is highly cumbersome and generally depends on the system.

Complex computational calculations are at the forefront of numerical methods to tackle down simulation of quantum systems. For instance, a parametrization of quantum states in terms of neural networks has enabled the approximation of many-body wavefunctions in closed quantum systems [25–27], and it has also been extended to approach the density operator in open dynamics (dissipative) [28–33]. Along these ideas, other models have focused on hybrid implementations [34–36], probabilistic formulations based on positive operator-valued measure [37, 38], or data-driven model via time-averaged generators [39]. Overall, estimating the dynamics of open quantum systems is a challenging problem. Here, machine learning provides versatile and promising algorithms to expand our alternatives towards completing this task [40–46]. However, combining time evolution and QC with artificial intelligence, being solved

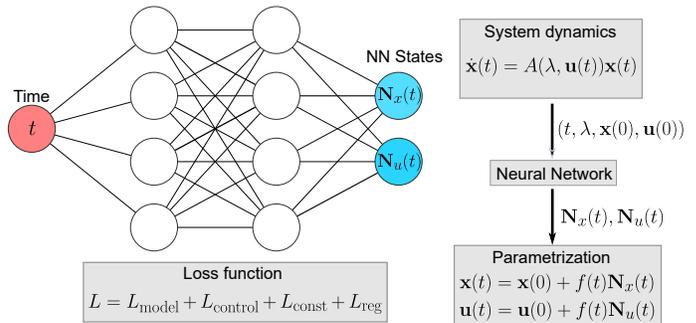


FIG. 1. PINNs architecture for solving quantum control problems. Quantum evolution can be translated into a dynamical system $\dot{\mathbf{x}}(t) = A(\lambda, \mathbf{u}(t))\mathbf{x}(t)$, where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the state and control vectors, respectively, and λ are the system parameters. The input data (red circle) is given by the discrete time vector t , and the outputs of the neural network (NN) are $\mathbf{N}_x(t)$ and $\mathbf{N}_u(t)$ (blue circle). By minimizing the loss function L the NN discover $\mathbf{N}_{x,u}(t)$ for the parameterized solutions $\mathbf{x}(t)$ and $\mathbf{u}(t)$.

within a single deep learning method still needs to be explored.

Neural Networks (NNs) are commonly trained with data allowing them to learn the dynamics of quantum systems. However, NNs that preserve the underlying physical laws without preliminary data would have practical advantages. Hence, physics-informed neural networks (PINNs) have been introduced as a new artificial intelligence paradigm that only requires the model itself [47, 48]. This is a general physics-informed machine learning framework that has been applied to solve high-dimensional partial differential equations [49, 50], many-body quantum systems [51, 52] and quantum fields [53], inverse problems using sparse and noisy data [54], and to discover underlying physics hidden in data structures [55]. Since PINNs are coded using phys-

ical laws, they can be applied to any quantum evolution where the model is well known [56–62].

In this letter, we introduce a novel PINN architecture to find optimal control functions in open quantum systems. This is a data-free inverse modeling deep learning approach with a target dynamical behavior instead of data. Our approach suggests smooth control functions for driving quantum states to a pre-selected target state.

Let us consider the following n -dimensional non-autonomous dynamical system:

$$\dot{\mathbf{x}} = A(\lambda, \mathbf{u}(t))\mathbf{x}(t), \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{u}(0) = \mathbf{u}_0, \quad (1)$$

where $\mathbf{x}(t) = (x_1, \dots, x_n)^T \in \mathbb{R}^n$, $\mathbf{u}(t) = (u_1, \dots, u_m)^T \in \mathbb{R}^m$, and $\lambda = (\lambda_1, \dots, \lambda_s)^T \in \mathbb{R}^s$ are the state, control, and parameter vectors (with $n, m, s \geq 1$ and $m \leq n^2$), respectively. Here, $A(\lambda, \mathbf{u}(t))$ is a real $n \times n$ dynamical matrix that depends on $\mathbf{u}(t)$ (control) and λ (parameters).

Given λ and $\mathbf{x}(t)$ satisfying Eq. (1), we can apply machine learning to discover an optimal control vector $\mathbf{u}(t)$ such that the system evolves from $\mathbf{x}(0)$ to some desired target state \mathbf{x}_d . Techniques based on optimal control [63], Lyapunov control theory [6] or linear control theory are based on optimization rules to find a suitable control vector $\mathbf{u}(t)$. Therefore, the main idea is to construct a PINN that minimizes a loss function to achieve optimal quantum control.

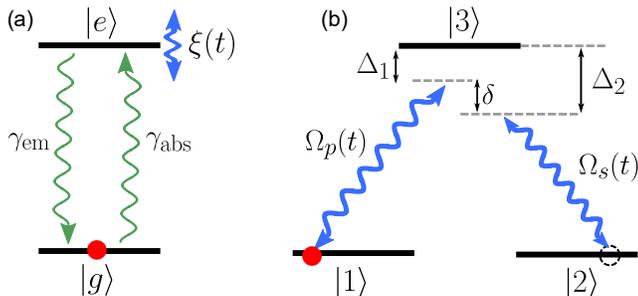


FIG. 2. (a) Open two-level system controlled by a time-dependent modulation $\xi(t)$. Losses are included through absorption (γ_{abs}) and emission (γ_{em}) processes. (b) Three-level Λ system controlled by two driving fields $\Omega_p(t)$ and $\Omega_s(t)$. We include pure dephasing rates γ_i ($i = 1, 2, 3$) acting on each state $|i\rangle$. One- ($\Delta_{1,2}$) and two-photon (δ) detunings are considered in our simulations.

A feed forward NN is a powerful universal approximator for any vector function $F : \mathbb{R}^p \mapsto \mathbb{R}^q$ ($r, q \geq 1$) (Universal Approximation Theorem) [64]. Let us consider the NN architecture illustrated in Fig. 1 as a new paradigm to solve quantum control problems. We use an equally distributed time array $t = (t_1, \dots, t_M)$ as the input to the NN, with M representing the dimension of the sample points. PINNs do not require a structured mesh; thus, t_i can be arbitrarily discretized. The NN consists of multiple hidden layers with activation function $\sin(\cdot)$ for the hidden neurons. This choice of activation has been shown to improve PINNs’ performance in solving nonlinear dynamical

systems [69] and high-dimensional partial differential equations [50]. The outputs of the NN are the solutions $\mathbf{N}_x(t) \in \mathbb{R}^n$ and $\mathbf{N}_u(t) \in \mathbb{R}^m$. We construct a neural state and control vectors that identically satisfy the initial conditions by using a hard constraint $\mathbf{x}(t) = \mathbf{x}(0) + f(t)\mathbf{N}_x(t)$ and $\mathbf{u}(t) = \mathbf{u}(0) + f(t)\mathbf{N}_u(t)$, where $f(t) = 1 - e^{-t}$ is a function satisfying $f(0) = 0$ [69]. This hard constraint approach avoids numerical errors in the initial conditions. The network parameters, weights and biases, are randomly initialized, and then, they are optimized by minimizing a physics-informed loss function defined by

$$L = L_{\text{model}} + L_{\text{control}} + L_{\text{const}} + L_{\text{reg}}. \quad (2)$$

The component L_{model} describes the system dynamics:

$$L_{\text{model}} = \sum_{i=1}^M \|\dot{\mathbf{x}}(t_i) - A(\lambda, \mathbf{u}(t_i))\mathbf{x}(t_i)\|^2, \quad (3)$$

with $\|\cdot\|$ representing a Euclidean distance. The time derivatives in the neural solutions are computed using the automatic differentiation method provided by PyTorch package [66]. By minimizing the above functional, the state vector will approximately satisfy the system dynamics and, thus, the underlying physics. The second term on the right-hand side of Eq. (2) represents the control, that reads

$$L_{\text{control}} = \eta \sum_{i=1}^M \|\mathbf{x}(t_i) - \mathbf{x}_d\|^2, \quad 0 \leq \eta \leq 1, \quad (4)$$

where the factor η regulates the relevance of the control condition compared to the leading model component L_{model} . Note that \mathbf{x}_d could be a constant (regulation) or time-dependent (trajectory tracking) vector depending on the control scheme. The term L_{const} could take into account additional physical constraints for the state/control vector, respectively, such as probability conservation or holonomic constraints of the form $H(\mathbf{x}, \mathbf{u}, t) = 0$. Finally, L_{reg} is a standard regularization loss term that encourages the network parameters to take relatively small values avoiding overfitting. We remark that imposing initial conditions into the loss function (soft constraint) is also possible, as illustrated in Ref. [47]. The comparison between soft and hard constraints are discussed in Ref. [67]. The effect of overfitting will be the prediction of a too complex $\mathbf{u}(t)$, which might be practically unfeasible for designing a real control. We introduce L_{reg} as a l_2 -norm of the network weights $L_{\text{reg}} = \chi \sum_i w_i^2$, where χ is the regularization parameter.

The minimization of the loss function given in Eq. (2) yields NN predictions that obey the underlying physics and suggest optimal control functions. For the training (minimization of Eq. (2)), we employ Adam optimizer [77]. Moreover, the points t_i are randomly perturbed during training iteration— this method has been shown to improve the training and the neural predictions [49, 69]. To

highlight the method and keep the presentation elegant, we focused on low-dimensional Hilbert space examples. In the Supplemental Material, we demonstrate that the proposed PINN can successfully deal with larger systems.

We consider a two-level system (TLS) as a proof-of-principle example to illustrate the use of PINNs for QC. We address the problem of generating Gibbs (mixed) states of the form $\rho_{\text{Gibbs}} = Z^{-1} \sum_j e^{-\beta E_j} |j\rangle\langle j|$ with $Z = \text{Tr}(e^{-\beta H})$ (partition function) and $\beta = (k_B T)^{-1}$ (inverse temperature). The preparation of mixed states is relevant for simulating high-temperature superconductivity in variational quantum circuits [78]. In addition, QC of two-level systems is also relevant in the context of pulse reverse engineering [79], feedback control [3], optimal control theory [63], and controllable quantum dissipative dynamics [80]. Quantum transitions can be written in terms of the operators $\sigma_{ij} = |i\rangle\langle j|$ ($i, j = e, g$) being $|e\rangle$ ($|g\rangle$) the excited (ground) state. Let us consider the following Hamiltonian with a phase damping control

$$H(t) = \omega_z \sigma_z + \omega_x \sigma_x + \xi(t) \sigma_{ee}, \quad (5)$$

with $\omega_{x,z}$ representing system parameters and $\xi(t)$ describing the unknown control field. Here, $\sigma_z = \sigma_{ee} - \sigma_{gg}$ and $\sigma_x = \sigma_{eg} + \sigma_{ge}$. Here, $\xi(t)$ is the control used to generate Gibbs states. To train our PINN we use the Markovian master equation $\dot{\rho} = -i[H(t), \rho] + \sum_{i=1,2} \gamma_i (L_i \rho L_i^\dagger - (1/2)\{L_i^\dagger L_i, \rho\})$, with $[\cdot, \cdot]$ ($\{\cdot, \cdot\}$) representing the commutator (anticommutator). The amplitude damping channel is described by absorption ($L_1 = \sigma_{eg}$) and emission ($L_2 = \sigma_{ge}$) processes with rates $\gamma_1 = \gamma_{\text{abs}}$ and $\gamma_2 = \gamma_{\text{em}}$, respectively, see Fig. 2-(a). In what follows, we use $\omega_z = 2$, $\omega_x = 1$, $\gamma_{\text{abs}} = 0.1$, $\gamma_{\text{em}} = 0.3$, and $\xi(0) = 0$. For $\xi(t) = 0$, we get the steady-state (ss) $\rho^{\text{ss}} = 0.2775|e\rangle\langle e| + 0.7225|g\rangle\langle g| + [(-0.1106 + i0.0083)|g\rangle\langle e| + c.c.]$. Thus, we use $\xi(t)$ to drive the system to another ss, say, $\rho_d = (1/2)(|e\rangle\langle e| + |g\rangle\langle g|)$. Hence, we introduce the real state vector $\mathbf{x}(t) = (x_1, x_2, x_3, x_4)^T = (\rho_{gg}, \rho_{ee}, \text{Re}(\rho_{eg}), \text{Im}(\rho_{eg}))^T$, where $\rho_{ij} = \langle i|\rho(t)|j\rangle$ are the elements of the density matrix. The dynamics can be written as $\dot{\mathbf{x}} = A(\lambda, \mathbf{u}(t))\mathbf{x}(t)$, with

$$A(\lambda, \mathbf{u}(t)) = \begin{pmatrix} -\gamma_{\text{abs}} & \gamma_{\text{em}} & 0 & -2\omega_x \\ \gamma_{\text{abs}} & -\gamma_{\text{em}} & 0 & 2\omega_x \\ 0 & 0 & -\Gamma & -2\omega_z - \xi(t) \\ \omega_x & -\omega_x & 2\omega_z + \xi(t) & -\Gamma \end{pmatrix}, \quad (6)$$

where $\lambda = (w_x, w_z, \gamma_{\text{abs}}, \gamma_{\text{em}})$ is the set of parameters, $\mathbf{u}(t) = \xi(t)$ is the control vector that needs to be discovered, and $\Gamma = (1/2)(\gamma_1 + \gamma_2)$ is the effective dephasing rate. We remark that all results concerning the time evolution are simulated from a traditional ODE solver using the control obtained from the PINN. In Fig. 3, we plot the dynamics using the PINN's prediction for the control $\xi(t)$ (inset). The PINN discovers an optimal Gibbs state preservation with fidelity $F(\rho(t), \rho_d) = [\text{Tr}([\rho^{1/2}(t)\rho_d\rho^{1/2}(t)]^{1/2})]^2 = 0.99$ (for $t \geq 20$), and the steady-state approaches to ρ_d within an

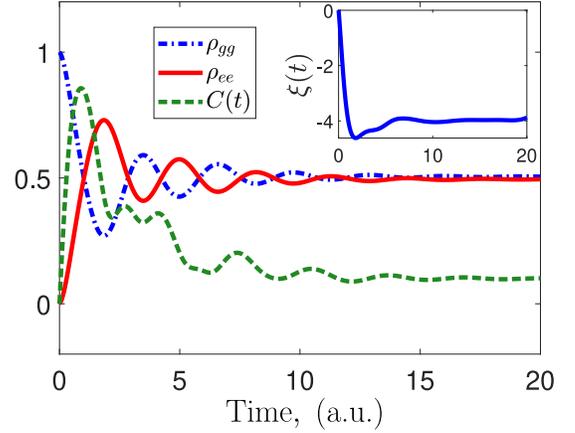


FIG. 3. Evolution of populations $\rho_{gg}(t)$, $\rho_{ee}(t)$ and coherence $C(t)$ using the control function $\xi(t)$ predicted by the PINN. The architecture of the NN consists of 4 hidden layers of 200 neurons, it is trained for 4×10^4 epochs with a learning rate 10^{-4} , $\chi = 10^{-3}$, and $\eta = 1$.

error of 1% for each component of the density matrix [67]. We remark that our result outperforms the analytically optimized solution that finds $\rho_{gg}^{\text{ss}} = 0.549$, $\rho_{ee}^{\text{ss}} = 0.4510$, $\text{Re}(\rho_{eg}) = 0$, and $\text{Im}(\rho_{eg}) = 0.049$, for a constant control $\xi^{\text{ss}} = -4$ (see [67] for further details). The latter explains the asymptotic behavior for $\xi(t)$ predicted by the PINN. Moreover, we note in Fig. 3 that the quantum coherence $C(t) = 2|\rho_{eg}(t)|$ is highly activated during the transient dynamics in order to generate an equally distributed mixed state, but it asymptotically reaches $C \approx 0.1014$.

We now focus on a Λ -configuration with two control fields (see Fig. 2-(b)), a platform for studying electromagnetically induced transparency [81, 82], coherent population trapping [83, 84], and adiabatic population transfer [85]. The latter has been dubbed Stimulated Raman Adiabatic Passage (STIRAP) [86]. Let's begin with the system Hamiltonian $H = \sum_i E_i \sigma_{ii} + H_c(t)$, where E_i stands for the eigenenergies, $\sigma_{ij} = |i\rangle\langle j|$, and $H_c(t)$ is the control Hamiltonian. In a multi-rotating frame and after the rotating wave approximation, the dynamics of the three-level system is governed by ($\hbar = 1$):

$$H(t) = \delta \sigma_{22} + \Delta_1 \sigma_{33} + \left(\frac{\Omega_p(t)}{2} \sigma_{31} + \frac{\Omega_s(t)}{2} \sigma_{32} + h.c. \right), \quad (7)$$

where $\Delta_1 = E_3 - E_1 - \omega_p$ and $\Delta_2 = E_3 - E_2 - \omega_s$ are the one-photon detunings that originate from off-resonant driving fields with frequencies ω_p and ω_s , while $\delta = \Delta_1 - \Delta_2$ is the two-photon detuning. Here, $\Omega_p(t)$ and $\Omega_s(t)$ are the control fields to be found. A similar Hamiltonian can be obtained from the interaction of a Nitrogen-Vacancy center with a Carbon-13 nuclear spin [87, 88]. We aim to find the optimal control pulses ($\Omega_p(t)$ and $\Omega_s(t)$) to transfer population from state $|1\rangle$ to state $|2\rangle$ via the lossy intermediary state $|3\rangle$. Our goal is to train a PINN that completes the task reaching high fidelity, with (i) minimizing the popu-

lation in the state $|3\rangle$, and (ii) minimizing the pulse area. We train our model using the Markovian master equation ($\hbar = 1$)

$$\dot{\rho} = -i[H(t), \rho] + \sum_{i=1}^3 \gamma_i (2\sigma_{ii}\rho\sigma_{ii} - \sigma_{ii}\rho - \rho\sigma_{ii}), \quad (8)$$

where $\gamma_i > 0$ are dephasing rates. We set $\gamma_3 = 0.14$ and $\gamma_1 = \gamma_2 = 10^{-3}$ to account for larger dissipation in the excited state. We use the real vector $\mathbf{z} = (\rho_{11}, \rho_{22}, \rho_{33}, \text{Re}[\rho_{12}], \text{Im}[\rho_{12}], \text{Re}[\rho_{13}], \text{Im}[\rho_{13}], \text{Re}[\rho_{23}], \text{Im}[\rho_{23}])^T$ to write the dynamics (details are given in [67]).

In Fig. 4-(a), we show the population evolution and the predicted NN solutions for the control fields $\Omega_{s,p}(t)$. Note that our PINN successfully delivers a population transfer with smooth pulses. Furthermore, it attempts to implement a counterintuitive sequence, turning on the Stoke pulse Ω_s (red-solid) at the same time that the pump field Ω_p (blue-dashed)—for a genuine counterintuitive sequence like STIRAP, Stokes pulse precedes the pump pulse. This is remarkable, considering that the PINN does not know QC theory or the relevance of following a dark state evolution. It is worthwhile noticing that the Stoke pulse shown in the inset of Fig. 4-(a) triggers the $|2\rangle \leftrightarrow |3\rangle$ transition, which in STIRAP serves the purpose of preparing a dark state [88], since initially, all the population is in state $|1\rangle$.

For completeness, we consider a more challenging initial state given by $\rho(0) = \sigma_{11}/2 + \sigma_{22}/2 + \epsilon(\sigma_{12} + \sigma_{21})/2$. For $\epsilon = 0$, we end up with a fully mixed state (without quantum coherence), while $\epsilon = 1$ provides a balanced coherent state. Our PINN can handle this new task without changing the network's architecture, showing that PINNs provide a general and adaptive framework for inverse design (standard methods are not designed for this task). In Figs. 4-(b) and -(c), we show the population transfer and the corresponding pulse sequences for $\epsilon = 1$ and $\epsilon = 0.7$, respectively. Note that the PINN updates the pulses to deliver good polarizations.

For a thorough benchmarking, we consider other control methods besides STIRAP [85, 86], such as Stimulated Raman Exact Passage (STIREP) [89], Inverse Engineering [90, 91] and Modified Superadiabatic Transitionless Driving (MOD-SATD) [21, 92]. For detailed calculations of these pulses, see [67]. In Table II we show a comparison. One can observe that PINN itself speeds up the population transfer with a high fidelity and using a small amount of energy. The transfer time t_f is defined as the time required to reach the highest probability in the target state (more details in Ref. [67]). The predicted control functions have the smallest area \mathcal{A} compared to the other methods. We remark that the regularization L_{reg} penalizes the fields for being too large and provides smooth functions. Thus, we can control the amplitudes of the fields and the pulse area to achieve a less power-consuming transfer. Another important advantage of our protocol is the robustness against changes over initial training parameters. It is known that

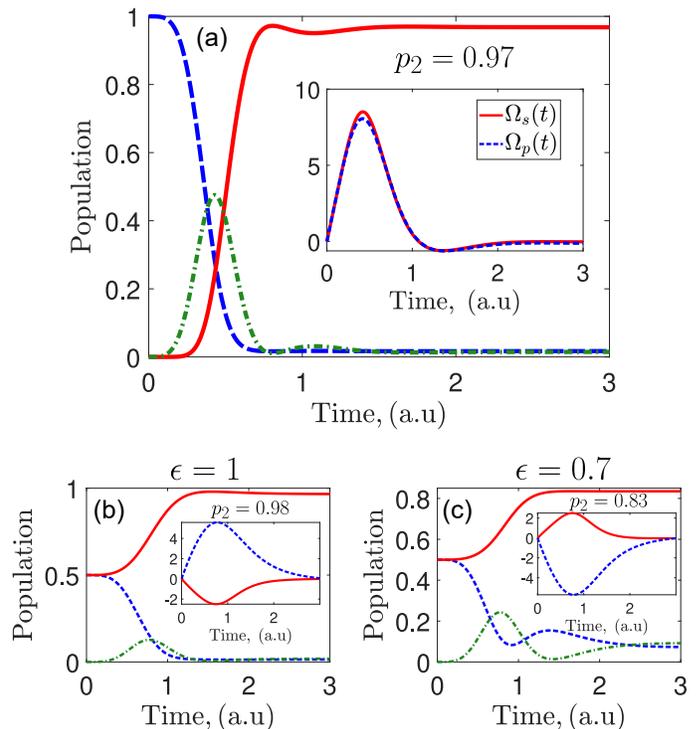


FIG. 4. (a) Population dynamics for the Λ -system using $\Omega_{s,p}(t)$ predicted by the PINN. (b) The PINN allows us to polarize the system starting from a coherent ($\epsilon = 1$) and quasi-thermal state ($\epsilon = 0.7$). The architecture of the NN includes 5 hidden layers of 150 neurons and it is optimized over 2×10^4 training epochs with learning rate 8×10^{-3} , $\eta = 0.2$ and $\chi = 2.8 \times 10^{-3}$ [67].

STIRAP deteriorates when increasing δ [93, 94]. Furthermore, the optimization for the other sequences with $\delta \neq 0$ is non trivial, and there is not much literature about it—to our best knowledge. To compare the robustness of the earlier discovered pulse sequences, we also report in Table II the population transfer in the presence of two-photon detuning $\delta/2\pi = 0.2$. We stress that no training or further optimizations have been made to account for the new δ . Therefore, based on Table II, we conclude that: i) PINNs can reach high fidelities in a short time under low energy consumption, and ii) PINNs are very robust when initial training parameters are changed, delivering better results in comparison with standard methods. In [67], we show that PINNs can be easily trained to counteract the adversary effect of δ .

Finally, we extend our calculations to a four-level system and show that our PINN performs well against cross-talk to the newly added state, and also we check that our protocol deal with larger systems [67].

In this letter, we introduced a physics-informed neural network to find control functions in open quantum systems. We demonstrated a data-free deep learning approach that jointly solves the open dynamics of quantum systems and the inverse design of control functions. First, we applied this formalism to prepare a Gibbs state in a two-level sys-

TABLE I. The Table shows the population $p_2 = \text{Tr}[\rho\sigma_{22}]$, pulse area $\mathcal{A} = \int_0^{t_f} dt \sqrt{|\Omega_p(t)|^2 + |\Omega_s(t)|^2}$ and transfer time t_f (in arbitrary units). In parenthesis, we report the values with $\Delta_1/2\pi = 0.2$ and $\delta/2\pi = 0.2$, the one- and two-photon detuning, respectively.

	PINN	STIRAP	STIREP	Inv. Eng.	MOD-SATD
p_2	0.97(0.93)	0.98(0.88)	0.98(0.91)	0.97(0.79)	0.98(0.89)
\mathcal{A}	7.3	128.6	53.3	19.8	50.0
t_f	2.0	35	9.4	3.0	13

tem. Second, we applied it to state-to-state transfer in a three-level system. We found that the PINN provides a flexible method that adapts to different parameters, initial states, noise channels, and power consumption requirements. We hope that PINNs will be very attractive for problems such as adiabatic quantum computing, quantum gates, state preservation, manipulation in high-dimensional Hilbert spaces, initialization of entangled states, and time-dependent induced behavior in many-body systems.

A.N. and R.C acknowledge the financial support from the projects Fondecyt Iniciación #11220266 and #11180143, respectively. F.J.G. acknowledges support from Universidad Mayor through the Doctoral fellowship.

* ariel.norambuena@umayor.cl

† raul.coto@protonmail.com

[1] P. Jamonneau, G. Hétet, A. Dréau, J.-F. Roch, and V. Jacques, Phys. Rev. Lett. 116, 043603 (2016).
[2] Y. Lin, J. Gaebler, F. Reiter, et al., Nature 504, 415–418 (2013).
[3] A. M. Brańczyk, P. E. M. F. Mendonca, A. Gilchrist, A. C. Doherty, and S. D. Bartlett, Phys. Rev. A 75, 012329 (2007).
[4] M. S. Blok, C. Bonato, M. L. Markham, D. J. Twitchen, V. V. Dobrovitski, and R. Hanson, Nat. Phys. 10, 189 (2014).
[5] V. Montenegro, R. Coto, V. Eremeev, and M. Orszag, Phys. Rev. A 96, 053851 (2017).
[6] A. Bacciotti and L. Rosier, *Liapunov Functions and Stability in Control Theory* (Springer, London, 2001), Second Edition.
[7] Y.-Q. Wang, Y. Wang, X. Zhao, et al. Quantum Inf Process 20, 404 (2021).
[8] J. Cui, R. van Bijnen, T. Pohl, S. Montangero and T. Calarco, Quantum Sci. Technol. 2, 035006 (2017).
[9] F. Schäfer, M. Kloc, C. Bruder and N. Lörch, Mach. Learn.: Sci. Technol. 1, 035009 (2020).
[10] L. Coopmans, D. Luo, G. Kells, B. K. Clark and J. Carrasquilla, PRX Quantum 2, 020332 (2021).
[11] S. Cong, *Control of Quantum Systems: Theory and Methods* (Wiley, Singapore, 2014), First Edition.
[12] L. Viola, E. Knill, and S. Lloyd, Phys. Rev. Lett. 82, 2417 (1999).
[13] L. Viola and E. Knill, Phys. Rev. Lett. 94, 060502 (2005).
[14] Xing-Long Zhen, Fei-Hao Zhang, Guanru Feng, Hang Li,

and Gui-Lu Long Phys. Rev. A 93, 022304 (2016).
[15] J. Liu and S. Cong, 2011 9th IEEE International Conference on Control and Automation (ICCA), 318-323 (2011).
[16] S. L. Wu and W. Ma, Phys. Rev. A 105, 012204 (2022).
[17] Y.-X. Du, Z.-T. Liang, W. Huang, H. Yan, and S.-L. Zhu, Phys. Rev. A 90, 023821 (2014).
[18] M. Anderlini, P. J. Lee, B. L. Brown, J. Sebby-Strabley, W. D. Phillips, and J. V. Porto, Nature 448, 452 (2007).
[19] G. Calajo, M. J. A. Schuetz, H. Pichler, M. D. Lukin, P. Schneeweiss, J. Volz, and P. Rabl, Phys. Rev. A 99, 053852 (2019).
[20] D. Tancara, A. Norambuena, R. Peña, G. Romero, F. Torres, and R. Coto, Phys. Rev. A 103, 053708 (2021).
[21] B. B. Zhou, A. Baksic, H. Ribeiro, C. G. Yale, F. J. Heremans, P. C. Jerger, A. Auer, G. Burkard, A. A. Clerk, and D. D. Awschalom, Nature Phys. 13, 330–334 (2017).
[22] Jiazhao Tian, Tianyi Du, Yu Liu, Haibin Liu, Fangzhou Jin, Ressa S. Said, and Jianming Cai Phys. Rev. A 100, 012110 (2019).
[23] D. Leibfried, R. Blatt, C. Monroe, and D. Wineland, Rev. Mod. Phys. 75, 281 (2003).
[24] C. Altafini and F. Ticozzi, IEEE Transactions on Automatic Control 57, 1898-1917 (2012)
[25] G. Carleo and M. Troyer, Science 355, 602 (2017).
[26] Z. Cai and J. Liu, Phys. Rev. B 97, 035116 (2018).
[27] K. Choo, G. Carleo, N. Regnault, and T. Neupert, Phys. Rev. Lett. 121, 167204 (2018).
[28] G. Torlai and R. G. Melko, Phys. Rev. Lett. 120, 240503 (2018).
[29] A. Nagy and V. Savona, Phys. Rev. Lett. 122, 250501 (2019).
[30] M. J. Hartmann and G. Carleo, Phys. Rev. Lett. 122, 250502 (2019).
[31] F. Vicentini, A. Biella, N. Regnault, and C. Ciuti, Phys. Rev. Lett. 122, 250503 (2019).
[32] N. Yoshioka and R. Hamazaki, Phys. Rev. B 99, 214306 (2019).
[33] J. Carrasquilla and G. Torlai, PRX Quantum 2, 040201 (2021).
[34] Rouven Koch and Jose L. Lado, Phys. Rev. Research 3, 033102 (2021).
[35] B. Gardas, M. M. Rams, and J. Dziarmaga, Phys. Rev. B 98, 184304 (2018).
[36] Z. Liu, L.-M. Duan, and D.-L. Deng, Phys. Rev. Research 4, 013097 (2022).
[37] M. Reh, M. Schmitt, and M Garttner, Phys. Rev. Lett. 127, 230501 (2021).
[38] D. Luo, Z. Chen, J. Carrasquilla, and B. K. Clark, Phys. Rev. Lett. 128, 090501 (2022).
[39] P. P. Mazza, D. Zietlow, F. Carollo, S. Andergassen, G. Martius, and I. Lesanovsky, Phys. Rev. Research 3, 023084 (2021).
[40] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, npj Quantum Inf 5, 33 (2019).
[41] Tangyou Huang, Yue Ban, E. Ya. Sherman, and Xi Chen, Phys. Rev. Applied 17, 024040 (2022).
[42] D. Castaldo, M. Rosa, and S. Corni, Phys. Rev. A 103, 022613 (2021).
[43] Thomas Fosel, Petru Tighineanu, Talitha Weiss, and Florian Marquardt, Phys. Rev. X 8, 031084 (2018).
[44] Y. X. Zeng, J. Shen, S. C. Hou, T. Gebremariam, C. Li, Physics Letters A 384, 126886 (2020).
[45] J. Brown, P. Sgroi, L. Giannelli, G. S. Paraoanu, E. Paladino, G. Falci, M. Paternostro, and A. Ferraro, New J.

- Phys. 23, 093035 (2021).
- [46] V. V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. H. Devoret, Phys. Rev. X 12, 011059 (2022).
- [47] M. Raissi, P. Perdikaris, G.E. Karniadakis, Journal of Computational Physics 378, 686-707 (2019).
- [48] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, Nat Rev Phys 3, 422-440 (2021).
- [49] J. A. Sirignano, K. Spiliopoulos, Journal of Computational Physics 375, 1339-1364 (2018).
- [50] S. Zeng, Z. Zhang, Q. Zou, Journal of Computational Physics 463, 111232 (2022).
- [51] D. Pfau, J. S. Spencer, A. G. D. G. Matthews, W. M. C. Foulkes, Phys. Rev. Research 2, 033429 (2020).
- [52] Z. Zhu, M. Mattheakis, W. Pan, E. Kaxiras, Phys. Rev. Research (accepted, 2023).
- [53] J. M. Martyn, K. Najafi, and D. Luo, Phys. Rev. Lett. 131, 081601 (2023).
- [54] M. Angeli, G. Neofotistos, M. Mattheakis, E. Kaxiras, Chaos, Solitons & Fractals 154, 111621, 2021.
- [55] S. A. Desai, M. Mattheakis, D. Sondak, P. Protopapas, and S. J. Roberts Phys. Rev. E 104, 034312 (2021).
- [56] X. Fang, K. Kruse, T. Lu, and J. Wang, Rev. Mod. Phys. 91, 045004 (2019).
- [57] C. Letellier, I. Sendiña-Nadal, E. Bianco-Martinez, and M. S. Baptista, Sci Rep 8, 3785 (2018).
- [58] A. Hannachi, Phys. Rev. E 60, 429 (1999).
- [59] A. Tacchella, D. Mazzilli, and L. A. Pietronero, Nature Phys 14, 861-865 (2018).
- [60] S. Denisov, J. Klafter, and M. Urbakh, Phys. Rev. E 66, 046203 (2002).
- [61] D. Harnack, E. Laminski, M. Schunemann, and K. R. Pawelzik, Phys. Rev. Lett. 119, 098301 (2017).
- [62] U. Gungordu and J. P. Kestner, Phys. Rev. Research 4, 023155 (2022).
- [63] W. H. Fleming and R. W. Rishel, *Deterministic and Stochastic Optimal Control* (Springer-Verlag, New York, 1975).
- [64] K. Hornik, Neural Networks 4, 251-257 (1991).
- [65] M. Mattheakis, D. Sondak, A. S. Dogra, P. Protopapas, Phys. Rev. E 105, 065305 (2022).
- [66] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, A. Lerer, NeurIPS (2017).
- [67] See Supplemental Material for more details and applications of the proposed method based on PINNs, which includes Refs. [68-75].
- [68] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, NeurIPS (2017).
- [69] M. Mattheakis, D. Sondak, A. S. Dogra, P. Protopapas, Phys. Rev. E 105, 065305 (2022).
- [70] Z. Zhu, M. Mattheakis, W. Pan, and E. Kaxiras, Phys. Rev. Research 5, 043084 (2023).
- [71] D. Guéry-Odelin, A. Ruschhaupt, A. Kiely, E. Torrontegui, S. Martínez-Garaot, J. G. Muga, Rev. Mod. Phys. 91, 045001 (2019).
- [72] Luigi Giannelli and Ennio Arimondo, Phys. Rev. A, 89, 033419 (2014).
- [73] X. Chen, I. Lizuain, A. Ruschhaupt, D. Guéry-Odelin, and J. G. Muga, Phys. Rev. Lett. 105, 123003 (2010).
- [74] X. Chen, E. Torrontegui, and J. G. Muga, Phys. Rev. A 83, 062116 (2011).
- [75] H. Sun, N. Xu, S. Fan, and M. Liu, Annals of Physics 418, 168200 (2020).
- [76] A. Kiely and A. Ruschhaupt, J. Phys. B: At. Mol. Opt. Phys. 47 115501 (2014).
- [77] D. P. Kingma, J. Ba, CORR (2014)
- [78] R. Sagastizabal, S. P. Premaratne, B.A Klaver, et al. npj Quantum Inf 7, 130 (2021).
- [79] D. Ran, Wu-Jiang Shan, Zhi-Cheng Shi, Zhen-Biao Yang, J. Song, and Y. Xia, Phys. Rev. A 101, 023822 (2020).
- [80] W. Wu and Ze-Zhou Zhang, Sci Rep 11, 7188 (2021).
- [81] K.-J. Boller, A. Imamoglu, and S. E. Harris, Phys. Rev. Lett. 66, 2593 (1991).
- [82] M. Fleischhauer, A. Imamoglu, and J. P. Marangos, Rev. Mod. Phys. 77, 633 (2005).
- [83] E. Arimondo and G. Orriols, Lett. al Nuovo Cim. 17, 333-338 (1976).
- [84] I. V. Jyotsna and G. S. Agarwal, Phys. Rev. A, 52, 3147 (1995).
- [85] J. R. Kuklinski, U. Gaubatz, F. T. Hioe, and K. Bergmann, Phys. Rev. A 40, 6741 (1989).
- [86] K. Bergmann, H. Theuer, and B. W. Shore, Rev. Mod. Phys. 70, 1003 (1998).
- [87] F. J. González and R. Coto, Quantum Sci. Technol. 7, 025015 (2022).
- [88] R. Coto, V. Jacques, G. Hétet, and J. R. Maze, Phys. Rev. B 96, 085420 (2017).
- [89] X. Laforgue, G. Dridi, and S. Guerin, Phys. Rev. A 105, 032807 (2022).
- [90] X. Chen and J. G. Muga, Phys. Rev. A 86, 033405 (2012).
- [91] Y.-Z. Lai, J.-Q. Liang, H. J. W. Müller-Kirsten, and J.-G. Zhou, Phys. Rev. A 53, 3691 (1996).
- [92] A. Baksic, H. Ribeiro, and A. A. Clerk, Phys. Rev. Lett. 116, 230503 (2016).
- [93] V.I. Romanenko and L.P. Yatsenko, Optics Comm. 140, 231 (1997).
- [94] N. V. Vitanov, A. A. Rangelov, B. W. Shore, and K. Bergmann, Rev. Mod. Phys. 89, 015006 (2017).

**SUPPLEMENTAL MATERIAL FOR
“PHYSICS-INFORMED NEURAL NETWORKS
FOR QUANTUM CONTROL”**

**BACKGROUND OF PHYSICS-INFORMED
NEURAL NETWORKS**

Physical laws are usually given in the form of differential equations. Neural Networks (NNs), on the other hand, provide closed-form solutions allowing the computation of analytical derivatives practically by using a technique called automatic differentiation [68]. Having the NN’s predictions and the derivatives in an analytical form, a physical law can directly be encoded in the loss function, that is, when the loss function converges to zero, the differential equation is approximately satisfied. These machine learning models are usually called physics-based or physics-informed NN (PINNs). For example, driven dynamical systems are described by ordinary differential equations of the general form:

$$D(t, \mathbf{x}, \dot{\mathbf{x}}, f(t)) = 0, \quad (9)$$

where t is the independent time variable, and $\mathbf{x}(t)$ is a vector that denotes the dependent variables subjected to certain initial conditions $\mathbf{x}(0) = \mathbf{x}_0$. In Eq. (9), D is an arbitrary function of t , $\mathbf{x}(t)$, the first time derivative $\dot{\mathbf{x}}$, and of the forcing time-dependent function $f(t)$. Similarly to the systems investigated in this study, Eq. (9) describes a non-autonomous system.

As shown in Fig. 1 of the main manuscript, we considered that the solutions \mathbf{x} are parameterized by a NN where t is the input to the network, and each output corresponds to a different variable of \mathbf{x} . Hence, solving the differential equations of Eq. (9) is reduced to an optimization problem of the form:

$$\arg \min_{\text{NN parameters}} \left(\sum_{i=0}^M \left(D(t_i, \mathbf{x}_i, \dot{\mathbf{x}}_i, f(t_i)) \right)^2 \right), \quad (10)$$

where $\mathbf{x}_i = \mathbf{x}(t_i)$, and $\dot{\mathbf{x}}_i$ is analytically computed using automatic differentiation. The summation in Eq. (10) is over all the time points M . It defines a loss function which minimization yields NN parameters (weights and biases) that construct a neural solution \mathbf{x} that approximately solves the differential equation (9). The initial conditions can be identically satisfied through a hard-constraint or approximately satisfied using a soft-constraint. In the main manuscript (in the text and in Fig. 1), it is explained how the initial conditions are identically satisfied through a specific parametrization. Later in Supplemental Material, we explain the soft-constraint approach and compare both in terms of the NN’s performance. Subsequently, during the optimization (training) phase the NN is encouraged to make predictions that satisfy the underlying differential equations; thus, the predictions are approximate solutions to the physical problem. Although data are not required in

this process, when they are available can be considered in an extra term in the loss function— usually in the form of a mean square error [47]. An important advantage of PINNs over conventional numerical methods is that any known information (such as symmetries, part of the solution, conservation laws, etc) can potentially be embedded in the loss function as a soft-constraint, improving the training and the accuracy of NN’s solutions [69].

It has been shown that NNs can learn very compact and rich representations, making PINNs a powerful tool for learning complex solutions, like high-dimensional many-body wavefunctions [25–27, 70]. Moreover, PINNs scale well in high dimensions, suffering less than conventional numerical methods from the curse of dimensionality [49, 50]. In the study, we introduce a novel physics-based architecture in the lines of PINNs. Our model can discover not only the solutions \mathbf{x} of Eq. (9) but also the forcing function when a target final state is given. This target state plays the role of data and is satisfied through an extra loss function (Eq. (4) in the main manuscript). Additional constraints, like minimizing the amplitude of the external force (the case of this work that is discussed in the main manuscript), can be introduced as extra terms in the loss function.

**CONTROL AND DYNAMICAL ANALYSIS IN A
TWO-LEVEL SYSTEM**

For the two-level system, we use the following loss function

$$L = \|\dot{\mathbf{x}} - A(\lambda, \mathbf{u}(t))\mathbf{x}\|^2 + \|\mathbf{x} - \mathbf{x}_d\|^2 + L_{\text{reg}}, \quad (11)$$

where $\mathbf{x}_d = (1/2, 1/2, 0, 0)^T$ is the desired target state vector which correspond to ρ_d . The first, second, and third terms on the right-hand side of the above equation imposes the Markovian master equation, the state preservation problem ($\rho(t) \rightarrow \rho_d$), and the constraint for off-diagonal elements ($\rho_{eg} \rightarrow 0$), respectively. From the equation of motion $\dot{\mathbf{x}} = A\mathbf{x}$ given in the section *Two-level system of the main text*, we found:

$$\dot{x}_1 = \gamma_e - 2\Gamma x_1 - 2\omega_x x_4, \quad (12)$$

$$\dot{x}_3 = -\Gamma x_3 - (2\omega_z + \xi(t)) x_4, \quad (13)$$

$$\dot{x}_4 = -\omega_x + 2\omega_x x_1 + (2\omega_z + \xi(t)) x_3 - \Gamma x_4, \quad (14)$$

where we have used $x_1 + x_2 = 1$ ($\text{Tr}(\rho) = 1$) and $\Gamma = (\gamma_{\text{abs}} + \gamma_{\text{em}})/2$. First, we can analyze the steady state which implies that solutions satisfy $\dot{x}_i = 0$ ($i = 1, 2, 3$). In what follow, we demonstrate that the steady-state solution for the density matrix can be analytically solved in terms of the $\xi^{\text{ss}} = \lim_{t \rightarrow \infty} \xi(t)$. Then, by numerically computing $\min_{\xi^{\text{ss}}} (1 - F(\rho(\xi), \sigma))$ ($F = 1$ for a good control problem) we can compare the theoretical and neural network predictions for the steady-state. From numerical calculations

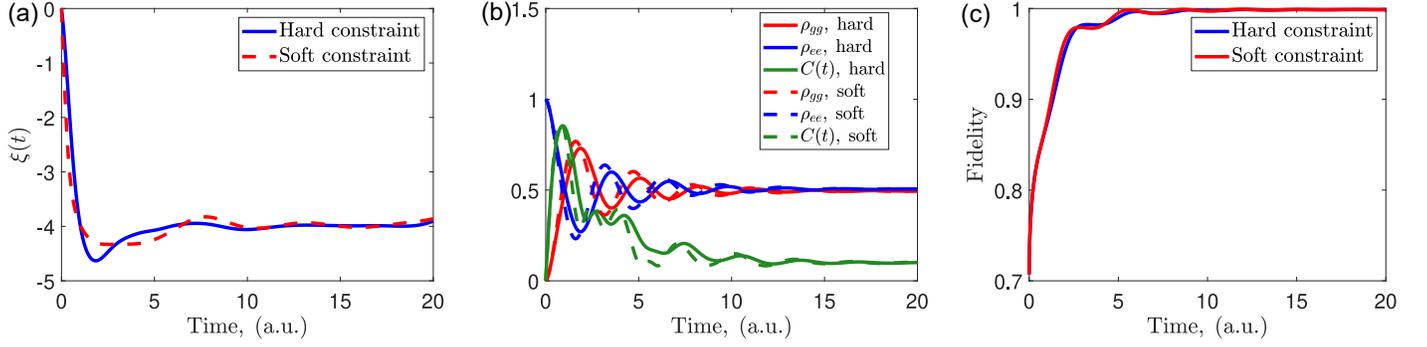


FIG. 5. Comparison between soft ($\lambda = 1$) and hard constraints used for initial conditions in terms of: (a) control function $\xi(t)$, (b) dynamics, and (c) quantum fidelity. We used the same set of parameters presented in the main text for the simulations.

using the PINN we found that $\xi(t)$ converges to a stable value, then we can asseverate that the steady state for $\xi(t)$ exist. Let us define x_i^{ss} and ξ^{ss} as the steady-state solutions, from equations (12)-(14), we obtain:

$$\begin{pmatrix} 2\Gamma & 0 & 2\omega_x \\ 0 & \Gamma & 2\omega_z + \xi^{ss} \\ 2\omega_x & 2\omega_z + \xi^{ss} & -\Gamma \end{pmatrix} \begin{pmatrix} x_1^{ss} \\ x_3^{ss} \\ x_4^{ss} \end{pmatrix} = \begin{pmatrix} \gamma_e \\ 0 \\ \omega_x \end{pmatrix}. \quad (15)$$

By solving the above linear system, we get the following analytical solutions for the steady-states in terms of ξ^{ss} :

$$x_1^{ss} = \frac{\gamma_{em}}{\gamma_{em} + \gamma_{abs}} + \frac{4w_x^2(\gamma_{abs} - \gamma_{em})}{(\gamma_{abs} + \gamma_{em}\Delta(\xi^{ss}))}, \quad (16)$$

$$x_3^{ss} = \frac{4w_x(\gamma_{abs} + \gamma_{em})(2w_z + \xi^{ss})}{(\gamma_{abs} - \gamma_{em})\Delta(\xi^{ss})}, \quad (17)$$

$$x_4^{ss} = -\frac{2w_x(\gamma_{abs} - \gamma_{em})}{\Delta(\xi^{ss})}, \quad (18)$$

$$\Delta(\xi^{ss}) = 4\Gamma^2 + 8w_x^2 + 16w_z^2 + 16w_z\xi^{ss} + 4(\xi^{ss})^2. \quad (19)$$

For the particular case $w_x = 1$, $w_z = 2$, $\gamma_{abs} = 0.1$, and $\gamma_{em} = 0.3$ it follow that

$$x_1^{ss} = \frac{3}{4} - \frac{25}{2\Delta(\xi^{ss})}, \quad (20)$$

$$x_3^{ss} = -\frac{25(\xi^{ss} + 4)}{\Delta(\xi^{ss})}, \quad (21)$$

$$x_4^{ss} = -\frac{5}{2\Delta(\xi^{ss})}, \quad (22)$$

$$\Delta(\xi^{ss}) = 451 + 200\xi^{ss} + 25(\xi^{ss})^2. \quad (23)$$

Therefore, since the target state is defined as $\rho_d = (1/2)(|e\rangle\langle e| + |g\rangle\langle g|)$, or equivalently $x_d = (1/2, 1/2, 0, 0)^T$, we note that in order to obtain $x_3^{ss} = 0$ it is necessary to have $\xi^{ss} = -4$, as is shown in the inset of Figure 4 in the main text. In order to corroborate this previous observations rigorously, we impose that the steady state must minimize the function $f(\xi) = 1 - F(\rho(\xi), \sigma)$,

where $\sigma = (1/2)(|e\rangle\langle e| + |g\rangle\langle g|)$ is the target state, $F(\rho(t), \sigma) = [\text{Tr}([\rho^{1/2}(t)\sigma\rho^{1/2}(t)]^{1/2})]^2$ is the quantum fidelity, and $\rho(\xi^{ss})$ is the steady state density matrix for ξ^{ss} , and is given by

$$\rho(\xi^{ss}) = \begin{pmatrix} x_1^{ss} & x_3^{ss} + ix_4^{ss} \\ x_3^{ss} - ix_4^{ss} & x_2^{ss} \end{pmatrix}, \quad (24)$$

where $x_2^{ss} = 1 - x_1^{ss}$. By employing a nonlinear optimization package of MATLAB we minimize $f(\xi) = 1 - F(\rho(\xi), \sigma)$ over all possible values of ξ^{ss} , and we obtain

$$\xi^{ss} = -4, \quad \text{and} \quad F(\rho(-4), \sigma) = 0.9988. \quad (25)$$

By setting the value $\xi^{ss} = -4$, we obtain the best performance for the steady state solution in terms of the quantum fidelity ($F \approx 1$), which implies that $x_1^{ss} = 0.549$, $x_3^{ss} = 0$, and $x_4^{ss} = 0.049$.

INITIAL CONDITIONS THROUGH HARD AND SOFT CONSTRAINTS

In this section we compare two different approaches to incorporate initial conditions in our neural network architecture. In general terms, we have two possibilities to impose initial conditions, namely, i) use the parametric solution $\mathbf{x}(t) = \mathbf{x}(0) + f(t)\mathbf{N}_x(t)$ and $\mathbf{u}(t) = \mathbf{u}(0) + f(t)\mathbf{N}_u(t)$ with $f(0) = 0$ (**hard constraint**) or ii) impose initial conditions into the loss function (**soft constraint**). To implement the soft constraint (without using the parametrization), we introduce the following loss function

$$L_{\text{soft}} = L_{\text{model}} + L_{\text{control}} + L_{\text{const}} + L_{\text{reg}} + L_{\text{ic}} = L + \lambda L_{\text{ic}}, \quad (26)$$

where λ controls the soft constraint contribution, $L = L_{\text{model}} + L_{\text{control}} + L_{\text{const}} + L_{\text{reg}}$ is the same loss function presented in the main text, where we only added the term L_{ic} to take into account the initial conditions in the loss

function. In the soft constraint approach, the term L_{ic} is defined as:

$$L_{ic} = [||\mathbf{x}(t_1) - \mathbf{x}(0)||^2 + ||\mathbf{u}(t_1) - \mathbf{u}(0)||^2], \quad (27)$$

where $\mathbf{x}(t)$ and $\mathbf{u}(t)$ are the vector and control functions, respectively, and t_1 is the initial time. To make a further comparison, we now solve the two-level system using these two different approaches (hard and soft). In Fig. 5 we plot a comparison between the soft (for $\lambda = 1$) and hard constraints for the neural network solution of the two-level system presented in the main text. In Fig. 5(a), we observe that both control functions $\xi(t)$ are very similar leading a temporal behavior shown in Fig. 5(b). We note that both methods work equally well in terms of quantum fidelity, see Fig. 5(c). However, by changing the parameter λ we observe that the convergence of the soft loss function decreases when λ increases, as shown in Fig. 6.

ENERGY EFFICIENCY OF THE CONTROL SCHEME

Let us consider a general Markovian open quantum system described by the master equation ($\hbar = 1$)

$$\dot{\rho} = -i[H_0 + H_c(t), \rho] + \mathcal{L}(\rho), \quad (28)$$

where H_0 and $H_c(t)$ are the bare and control Hamiltonians, respectively. Here, $\mathcal{L}(\rho)$ describes the loss term in the Lindblad master equation. Then, the average energy of the system can be defined as $\langle E \rangle = \text{Tr}(H(t)\rho)$, with $H(t) = H_0 + H_c(t)$. The average power can be written as

$$\begin{aligned} \frac{dU}{dt} &= \text{Tr}(\dot{H}_c(t)\rho(t)) + \text{Tr}(H(t)\dot{\rho}) \\ &= \dot{W}(t) + \dot{Q}. \end{aligned} \quad (29)$$

The term $\dot{W} = \text{Tr}(\dot{H}_c(t)\rho(t))$ is the power done by the system and $\dot{Q} = \text{Tr}(H(t)\dot{\rho})$ is the rate of heat induced by the reservoir of the two-level system. Thus, by direct integration we obtain the expression for the work $W(t)$ and heat $Q(t)$:

$$W(t) = \int_0^t \text{Tr}(\dot{H}_c(\tau)\rho(\tau)) d\tau, \quad (30)$$

$$Q(t) = \int_0^t \text{Tr}(H(\tau)\dot{\rho}) d\tau. \quad (31)$$

The time-dependent Hamiltonian competes with the heat flow induced by the environment. Therefore, we define $\eta = |W(t)/Q(t)|$ as the fraction of work done over the system compared to the internal heat that enters the system.

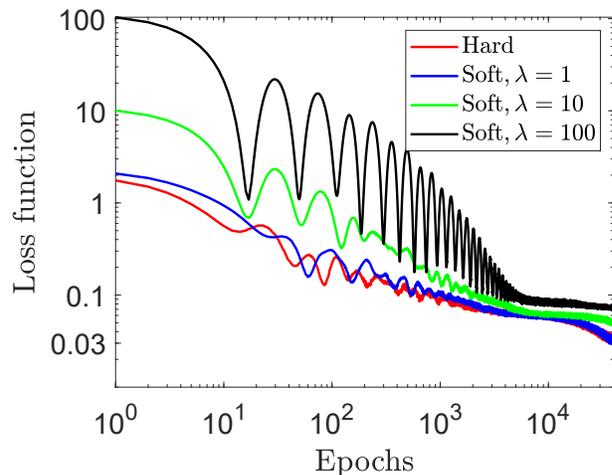


FIG. 6. Comparison between soft and hard constraints in terms of the convergence of the loss function.

For the particular case of the two-level system presented in the main text, the work is given by

$$W(t) = \int_0^t \dot{\xi}(\tau)\rho_{ee}(\tau) d\tau, \quad (32)$$

where $\xi(t)$ is the control field. In Figure 7(a) we plot the ratio the quantities $Q(t)$ and $W(t)$ for the two-level system. Also, in Figure 7(b) we show the temporal behavior of $\eta(t) = |W(t)/Q(t)|$ for the same parameters used in Figure. 3 of the main text. We note that we must apply a large quantity of work to overcome the effect of the internal heat. However, as time increases, the ratio η converges to 1, showing that the applied work compensates for undesired heat effects. Consequently, the system reaches a stable and free-losses state, which is required to preserve the state over time.

Now, we shall elaborate on a measure to quantify the efficiency of the control protocol in terms of energy considerations. By analyzing Figure 7(b), we note that an efficient quantum control solution can be recognized as the one that minimizes the colored area for the curve $\eta - 1$ for $\eta > 1$. In other words, if we control a quantum system without inverting a large quantity of energy, then $W_{\text{app}} \approx Q$, and therefore $\eta > 1$ only in a short time domain. Let us define the integral, $I_1 = \int_{\eta > 1} (\eta(\tau) - 1) d\tau$ as a measure of the region where $\eta > 1$. In the best scenario, $I_1 = 0$, implying that the energy efficiency will be equal to one. In addition, the total area of the curve η will be represented by $I_2 = \int \eta(\tau) d\tau$. Based on these observations, we define the energy efficiency over the whole process as $\text{Eff} = 1 - I_1/I_2$, or equivalently

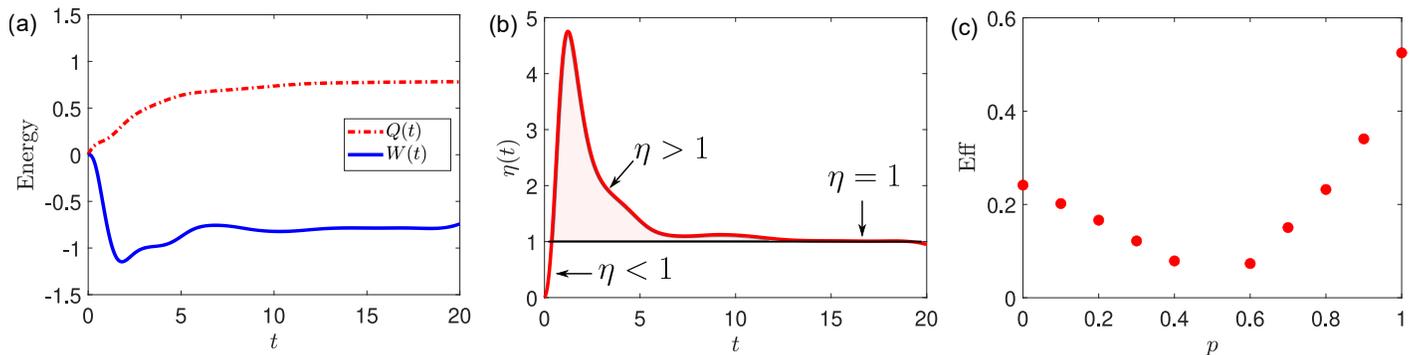


FIG. 7. (a) Heat and work as a function of time for the two-level system. (b) Parameter $\eta = |Q(t)/W(t)|$ as a function of time. (c) Efficiency parameter defined in Eq. (33) for parametrized initial states $|\Psi(0)\rangle = p|g\rangle\langle g| + (1-p)|e\rangle\langle e|$, where $0 \leq p \leq 1$.

$$\text{Eff} = 1 - \frac{\int_{\eta > 1} (\eta(\tau) - 1) d\tau}{\int \eta(\tau) d\tau}, \quad 0 \leq \text{Eff} \leq 1. \quad (33)$$

Since of $I_1 < I_2$ (by construction) and $I_1 = 0$ is the minimum value, it follow that $0 \leq \text{Ef} \leq 1$. Using this definition we have that $\text{Eff} = 0$ ($\text{Eff} = 1$) is the worst (best) scenario. For the two-level system, we obtain that $\text{Ef} = 0.5249$. Now, if we consider the initial state parametrization $\rho(0) = p|g\rangle\langle g| + (1-p)|e\rangle\langle e|$, we can calculate the energy efficiency in terms of the mixing parameter $0 \leq p \leq 1$. In Figure 7(c), we plot the efficiency of the PINN for $p = \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$. Note that the efficiency is not reported for $p = 0.5$ since in such a case, the initial state is equal to the target state.

CONTROL AND DYNAMICAL ANALYSIS IN A THREE-LEVEL SYSTEM

As it was stated in the main text, our PINN only handles real-valued functions. Therefore, we rewrite the master equation using the state vector $\vec{z} = (\rho_{11}, \rho_{22}, \rho_{33}, \text{Re}[\rho_{12}], \text{Im}[\rho_{12}], \text{Re}[\rho_{13}], \text{Im}[\rho_{13}], \text{Re}[\rho_{23}], \text{Im}[\rho_{23}])^T$. Hence, the dynamical equations reads,

$$\begin{aligned} 0 &= \dot{z}_1 + \Omega_p(t)z_7, \\ 0 &= \dot{z}_2 + \Omega_s(t)z_9, \\ 0 &= \dot{z}_3 - \Omega_p(t)z_7 - \Omega_s(t)z_9, \\ 0 &= \dot{z}_4 + \delta z_5 + (\gamma_1 + \gamma_2)z_4 + \frac{\Omega_p(t)}{2}z_9 - \frac{\Omega_s(t)}{2}z_7, \\ 0 &= \dot{z}_5 - \delta z_4 + (\gamma_1 + \gamma_2)z_5 + \frac{\Omega_p(t)}{2}z_8 - \frac{\Omega_s(t)}{2}z_6, \\ 0 &= \dot{z}_6 + \Delta_1 z_7 + (\gamma_1 + \gamma_3)z_6 + \frac{\Omega_s(t)}{2}z_5, \\ 0 &= \dot{z}_7 - \Delta_1 z_6 + (\gamma_1 + \gamma_3)z_7 + \frac{\Omega_p(t)}{2}z_3 - \frac{\Omega_p(t)}{2}z_1 \\ &\quad - \frac{\Omega_s(t)}{2}z_4, \\ 0 &= \dot{z}_8 - \delta z_9 + \Delta_1 z_9 + (\gamma_2 + \gamma_3)z_8 - \frac{\Omega_p(t)}{2}z_5, \\ 0 &= \dot{z}_9 + \delta z_8 - \Delta_1 z_8 + (\gamma_2 + \gamma_3)z_9 - \frac{\Omega_p(t)}{2}z_4 - \frac{\Omega_s(t)}{2}z_2 \\ &\quad + \frac{\Omega_s(t)}{2}z_3. \end{aligned} \quad (34)$$

We can define the loss function that accounts for the model as $L_{\text{model}} = \sum_{i=1}^9 \|L_i\|^2$, where L_i are given by the right-hand side in Eq. (34). The control loss function comes from the target state, such that

$$L_{\text{control}} = \eta \sum_{i=1}^M \|z_2(t_i) - 1\|^2. \quad (35)$$

It is well-known that the master equation in the Lindblad form ensures that the density matrix will be Hermitian, positive, semi-definite, and $\text{Tr}[\rho] = 1$. However, we found that enforcing the last constraint helps the PINN to find the best solution. Therefore, we add the following term to the loss function

$$L_{\text{const}} = \eta_c \sum_{i=1}^M (\|z_1(t_i)\|^2 + \|z_3(t_i)\|^2), \quad (36)$$

which together with L_{control} enforces $\text{Tr}[\rho] = 1$. Finally, we added a l_2 -norm regularization $L_{\text{reg}} = \chi \sum_i \Omega_i^2$ that penalizes the control fields being too large and delivers smooth functions. Henceforth, the PINN tries to minimize the overall loss function:

$$L = L_{\text{model}} + L_{\text{control}} + L_{\text{const}} + L_{\text{reg}}, \quad (37)$$

with a learning rate that amounts to 8×10^{-3} . The weights in L_{control} , L_{const} and L_{reg} ($\eta = 0.2$, $\eta_c = 0.1$ and $\chi = 2.8 \times 10^{-3}$, respectively) regulate the relevance of the control, constraint, and smoothness conditions comparing to the leading loss component L_{model} . These hyperparameters were adjusted manually. In Figure 8 we show the Loss function of the training process corresponding to Figure 4(a) in the main text, where we also show the convergence of L_{model} , L_{control} and L_{const} .

We now focus on the transfer time. We numerically observe that the PINN reduces the time for having a successful population transfer as compared to the other sequences, see Table I in the main text. This particular feature of the PINN naturally shows up when enforcing the control and constraining loss functions upon each time point (t_i). In other words, the PINN itself minimizes the transfer time. To realize this, we compute the PINN with only the control and constraint loss functions enforced by the last thirty points. We found that the population transfer slows down, and the probability slightly decreases. Nevertheless, the PINN can still handle this scenario if we randomly sample the time grid for each epoch and then pick thirty points (or even ten) to enforce the control and constrain loss functions. We remark that this is only possible because PINNs do not require a structured mesh; thus, t_i can be arbitrarily discretized.

PULSE SEQUENCES FOR CONTROLLING A Λ -SYSTEM.

In this section, we detail some control protocols that exploit adiabaticity, shortcuts to adiabaticity, or inverse engineering. We aim to efficiently transfer population from an initially polarized state ($|1\rangle$) to a target state ($|2\rangle$) via an intermediary state ($|3\rangle$). To achieve this state-to-state transfer, two radiation fields are used to induce transitions $|1\rangle \leftrightarrow |3\rangle$ and $|2\rangle \leftrightarrow |3\rangle$, as shown in Fig. 2(b). In what follows, we provide a detailed derivation of each of these methods— some of them already addressed by us in Ref. [87] and included here to make the present paper self-contained. We remark that the optimization can be improved by increasing the control fields for most of these methods and our PINN. However, we restrict them to $\Omega_{s,p} \leq 10$ to fairly compare the methods. In Fig. 9 we compare all methods for the population transfer between the states $|1\rangle$ and $|2\rangle$.

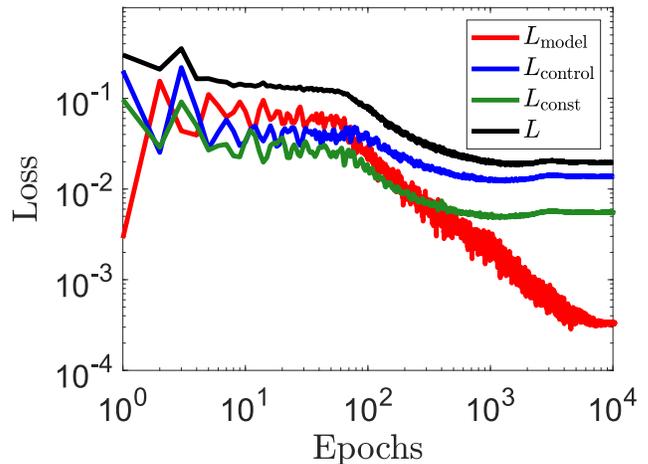


FIG. 8. Convergence of different components of the loss function for the three-level system defined in Eq. (37).

Stimulated Raman Adiabatic Passage (STIRAP)

First, we consider a well-known protocol for adiabatic transfer dubbed Stimulated Raman Adiabatic Passage (STIRAP) [85, 86, 94]. Under the rotating wave approximation, the Hamiltonian for this system is given by,

$$H(t) = \delta\sigma_{22} + \Delta_1\sigma_{33} + \left(\frac{\Omega_p(t)}{2}\sigma_{31} + \frac{\Omega_s(t)}{2}\sigma_{32} + h.c. \right). \quad (38)$$

Hereafter, we set to zero the two-photon detuning $\delta = 0$ ($\Delta_1 = \Delta_2 = \Delta$). The eigenstates of the Hamiltonian are:

$$\begin{aligned} |\Phi_+\rangle &= \sin\theta \sin\phi |1\rangle + \cos\phi |3\rangle + \cos\theta \sin\phi |2\rangle, \\ |\Phi_-\rangle &= \sin\theta \cos\phi |1\rangle - \sin\phi |3\rangle + \cos\theta \cos\phi |2\rangle, \\ |\Phi_d\rangle &= \cos\theta |1\rangle - \sin\theta |2\rangle, \end{aligned} \quad (39)$$

with instantaneous eigenvalues $E_d = 0$ and $E_{\pm}(t) = \Delta/2 \pm (\Delta^2 + \Omega_p^2(t) + \Omega_s^2(t))^{1/2}/2$. The mixing angles are defined through the relations

$$\tan\theta(t) = \frac{\Omega_p(t)}{\Omega_s(t)}, \quad \tan 2\phi(t) = \frac{\sqrt{\Omega_p^2(t) + \Omega_s^2(t)}}{\Delta}. \quad (40)$$

We note that the dark state ($|\Phi_d\rangle$) has no contribution from the excited state $|3\rangle$, so the population transfer from state $|1\rangle$ to state $|2\rangle$ is driven by the variation of the mixing angle $\theta(t)$. The latter implies that the Rabi frequencies $\Omega_p(t)$ and $\Omega_s(t)$ must be correlated. First, we note that $|\Phi_d\rangle$ coincides with $|1\rangle$ when $\theta(t) = 0$, which is obtained from $\Omega_p(t)/\Omega_s(t) \rightarrow 0$. Second, the population transfer is completed when $|\Phi_d\rangle$ coincides with $|2\rangle$ ($\theta(t) = \pi/2$), that is obtained from $\Omega_p(t)/\Omega_s(t) \rightarrow \infty$. Hence, the population transfer is attained with a counterintuitive pulse order, i.e. $\Omega_s(t)$ precedes $\Omega_p(t)$.

As mentioned above, the STIRAP protocol follows an adiabatic evolution, usually slower than superadiabatic

control protocols. Therefore, we now focus on three different modifications to STIRAP that use superadiabatic corrections [71–73, 75], inverse engineering [90, 91] and counteradiabatic approach [92].

Inverse Engineering

Inverse engineering is one of the protocols that achieves high fidelity in shorter times. The main goal is to design the optimal control pulses $\Omega_p(t)$ and $\Omega_s(t)$. To determine these control fields, an invariant operator $I(t)$ is used, which satisfies the equation [90, 91],

$$\frac{\partial I(t)}{\partial t} + \frac{1}{i\hbar}[I(t), H_0(t)] = 0. \quad (41)$$

The condition $\Delta_1 = 0$ yields the following operator ($\hbar = 1$),

$$I(t) = \frac{\Omega_0}{2} \begin{pmatrix} 0 & \Xi(t) & \Upsilon(t) \\ \Xi(t) & 0 & \Xi(t) \\ \Upsilon^*(t) & \Xi(t) & 0 \end{pmatrix}, \quad (42)$$

with $\Xi(t) = \cos \gamma(t) \sin \beta(t)$ and $\Upsilon(t) = -i \sin \gamma(t)$. The time-dependent auxiliary parameters $\gamma(t)$ and $\beta(t)$ satisfy the following equations [74],

$$\begin{aligned} \frac{d\gamma(t)}{dt} &= \frac{1}{2}(\Omega_p(t) \cos \beta(t) - \Omega_s(t) \sin \beta(t)), \\ \frac{d\beta(t)}{dt} &= \frac{1}{2} \tan \gamma(t)(\Omega_s(t) \cos \beta(t) + \Omega_p(t) \sin \beta(t)). \end{aligned} \quad (43)$$

From the above equations one can find the optimal control fields as

$$\begin{aligned} \Omega_s(t) &= 2\left(\frac{d\beta(t)}{dt} \cot \gamma(t) \cos \beta(t) - \frac{d\gamma(t)}{dt} \sin \beta(t)\right), \\ \Omega_p(t) &= 2\left(\frac{d\beta(t)}{dt} \cot \gamma(t) \sin \beta(t) + \frac{d\gamma(t)}{dt} \cos \beta(t)\right). \end{aligned} \quad (44)$$

Henceforward, we shall consider different Ansatzs for the auxiliary parameters $\gamma(t)$ and $\beta(t)$.

Ansatz 1: For simplicity, we choose $\gamma(t) = \epsilon$ and $\beta(t) = \pi t/2t_f$ [90], that leads to

$$\begin{aligned} \Omega_s(t) &= \frac{\pi}{t_f} \cot \epsilon \cos \left(\frac{\pi t}{2t_f}\right), \\ \Omega_p(t) &= \frac{\pi}{t_f} \cot \epsilon \sin \left(\frac{\pi t}{2t_f}\right). \end{aligned} \quad (45)$$

For the numerical calculations we set $\epsilon = 0.05$ and $t_f = 10$.

Ansatz 2: We choose the polynomial solution introduced in Ref. [90],

$$\beta(t) = \sum_{j=0}^3 b_j t^j, \quad \gamma(t) = \sum_{j=0}^4 a_j t^j, \quad (46)$$

where the coefficients b_j and a_j are determined from the initial conditions

$$\begin{aligned} \gamma(0) &= \epsilon, & \dot{\gamma}(0) &= 0, & \gamma(t_f) &= \epsilon, & \dot{\gamma}(t_f) &= 0, & \gamma(t_f/2) &= \delta, \\ \beta(0) &= 0, & \dot{\beta}(0) &= 0, & \beta(t_f) &= \frac{\pi}{2}, & \dot{\beta}(t_f) &= 0. \end{aligned} \quad (47)$$

To obtain an optimal and fast population transfer we set $\epsilon = 0.02$, $\delta = \pi/10$ and $t_f = 3$. We show our results for this ansatz in Table I in the main text.

Ansatz 3: We choose the solution proposed in Ref. [76], that achieves successful population transfer in the presence of unwanted transitions. The parameters are,

$$\begin{aligned} \gamma(t) &= -\frac{8(\pi - 2d_0)}{T^4} t^4 + \frac{2(7\pi - 16d_0 + T)}{T^3} t^3 \\ &\quad - \frac{5\pi - 16d_0 + 3T}{T^2} t^2 + t, \\ \beta(t) &= -\pi \left(\frac{t}{T}\right)^3 + \frac{1}{2}(2\pi d_1 + 3\pi) \left(\frac{t}{T}\right)^2 \\ &\quad + \left(-\frac{1}{2}(2\pi d_1 + 3\pi + \frac{3\pi}{2})\right) \frac{t}{T} + d_1 \sin \left(\frac{\pi t}{T}\right) \end{aligned} \quad (48)$$

where $d_0 = 1.8$, $d_1 = 0.1$ and $T = 1$. While this ansatz works fine, we did not include it in Table I due to the large control field, which we could not decrease below 10 with a successful transfer rate.

Stimulated Raman Exact Passage (STIREP)

Recently, a new protocol combines inverse engineering and optimization methods [89]. Here, the control pulses are determined by trajectories $\tilde{\phi}$ and read,

$$\begin{aligned} \tilde{\Omega}_s &= \Omega_s(t)/2\dot{\eta} = \cos \tilde{\phi} \sin \tilde{\theta} - \dot{\tilde{\phi}} \cos \tilde{\theta}, \\ \tilde{\Omega}_p &= \Omega_p(t)/2\dot{\eta} = -\cos \tilde{\phi} \cos \tilde{\theta} - \dot{\tilde{\phi}} \sin \tilde{\theta}, \end{aligned} \quad (50)$$

with $\tilde{\phi} = \phi[\eta(t)]$. The trajectory $\tilde{\phi}$ is obtained by applying robust inverse optimization (RIO) [89], which optimizes a cost function represented by the total area of the two pulses and is given by $\mathcal{A}_t = 2 \int_{\eta_i}^{\eta_f} |\dot{\eta}| \sqrt{\dot{\tilde{\phi}}^2 + \cos^2 \tilde{\phi}} d\eta$. By using the Euler-Lagrange equations and the Lagrange Multipliers method we obtain the following differential equations for the trajectory,

$$\begin{aligned} \dot{y}_1 &= \dot{\tilde{\phi}}^\pm = y_2, \\ \dot{y}_2 &= \ddot{\tilde{\phi}}^\pm = -(2y_2^2 + \cos^2 \tilde{\phi}^\pm) \tan \tilde{\phi}^\pm \\ \pm(\lambda_0 \sec \tilde{\phi}^\pm \lambda_1 \sin \eta - \lambda_2 \cos \eta) (y_2^2 + \cos^2 \tilde{\phi}^\pm)^{3/2} &= 0. \end{aligned} \quad (51)$$

For the initial condition $\dot{\tilde{\phi}} = 0$ we end up with the Lagrange Multipliers $\lambda_0 = 0.394$, $\lambda_1 = -0.064$ and $\lambda_2 = 0.283$.

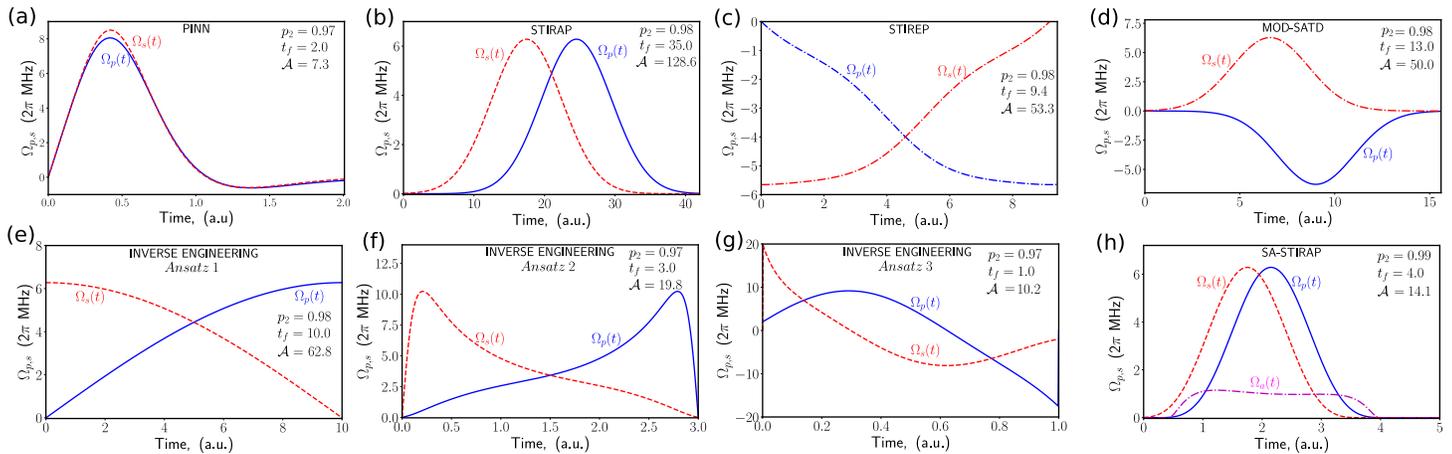


FIG. 9. Comparing all methods for population transfer from state $|1\rangle$ to $|2\rangle$.

Modified Superadiabatic Transitionless Driving (MOD-SATD)

The MOD-SATD [92] protocol is a different alternative, which bypasses the adiabatic condition while counteracting the effect of the loss of adiabaticity. The fields in this protocol are parameterized according to [92]

$$\begin{aligned}\Omega_p(t) &= -\Omega'(t) \sin \theta'(t), \\ \Omega_s(t) &= \Omega'(t) \cos \theta'(t),\end{aligned}\quad (52)$$

where $\theta'(t) = \theta(t) - \arctan[g_x(t)/(\Omega(t) + g_z(t))]$, $\Omega'(t) = \sqrt{[\Omega(t) + g_z(t)]^2 + g_x^2(t)}$, $\mu(t) = -\arctan[\theta/(\Omega(t) + g(t)/\sigma_m)]$, $g_x(t) = \dot{\mu}$, $g_z(t) = -\Omega(t) - \dot{\theta}/\tan(\mu)$, $g(t) = A/\cosh(\zeta t)$ with $A = 1/40$, $\zeta = 9/(10\sigma_m)$ and $\sigma_m = 2.0 \mu\text{s}$. For the Gaussian fields considered in STIRAP one finds that [92]

$$\begin{aligned}\theta(t) &= \arctan[\exp(t_d t/\sigma^2)], \\ \Omega(t) &= \Omega_0 \exp\left(-\frac{t^2 + t_d^2/4}{2\sigma^2}\right) \sqrt{2 \cosh\left(\frac{t_d t}{\sigma^2}\right)},\end{aligned}\quad (53)$$

with $t_d = 6/5\sigma$ and $\Omega_0/2\pi = 1 \text{ MHz}$.

Superadiabatic STIRAP (SA-STIRAP)

For completeness, we also show the SA-STIRAP sequence. However, we do not consider it in the main text (Table I) because it relies on a pulse connecting states $|1\rangle$ and $|2\rangle$, which we have not allowed for our PINN.

The superadiabatic approximation requires an external control Hamiltonian $H_c(t)$ [72, 73, 75], such that

$$H_{sa}(t) = H(t) + H_c(t),\quad (54)$$

where $H(t)$ is the original Hamiltonian in Eq. (38). The superadiabatic correction reads ($\hbar = 1$),

$$\begin{aligned}H_c(t) &= i \sum_{n=\pm,d} [|\partial_t \Phi_n(t)\rangle \langle \Phi_n(t)| \\ &\quad - \langle \Phi_n(t)| \partial_t \Phi_n(t)\rangle |\Phi_n(t)\rangle \langle \Phi_n(t)|],\end{aligned}\quad (55)$$

where $|\Phi_n\rangle$ are the eigenstates $H(t)$, as given in Eq. (39). After straightforward calculations we obtain the superadiabatic correction

$$H_c(t) = \frac{1}{2} (i\Omega_a(t)\sigma_{21} - i\Omega_a^*(t)\sigma_{12}),\quad (56)$$

with $\Omega_a(t) \equiv 2d\theta(t)/dt$, and $\theta(t)$ given in Eq. (40). Instead of Gaussian fields (as we used in the STIRAP protocol), here we use $\Omega_p(t) = \Omega_0 \sin^4(\pi(t - \tau)/T)$ and $\Omega_s(t) = \Omega_0 \sin^4(\pi t/T)$, with $\Omega_0/2\pi = 1$, $\tau = 0.1T$ and $T = 4$.

OPTIMIZATION WITH TWO-PHOTON DETUNING

The one-photon detuning does not have a detrimental role in the transfer protocol. For instance, it is known that it does not affect STIRAP in the adiabatic limit [94] (We performed numerical simulations and observed no difference with the one-photon detuning.). In contrast, the two-photon detuning (δ) hurts the success in the population transfer [93, 94]. Back to STIRAP, neither the dark state nor the null eigenvalue are available, forcing variations in the sequence (or a total departure from it) to achieve a successful transfer. Our PINN, on the other hand, can be easily trained to counteract this effect. It only requires setting the new value of δ before the training process, and then the PINN automatically updates the weights in the optimization and delivers a population transfer with a final probability $p_2 = 0.94$, see Fig. 10. Furthermore, it is worth noticing that the pulse sequence found by the PINN does not experience a significant change. This means the sequence is robust without a subsequent optimization ($p_2 = 0.93$ in the main text). It is important to note that along the manuscript we have fixed the regularization parameter $\chi = 2.8 \times 10^{-3}$. However, decreasing it down to $\chi = 1 \times 10^{-3}$ will increase the control fields and deliver a population transfer with $p_2 = 0.97$.

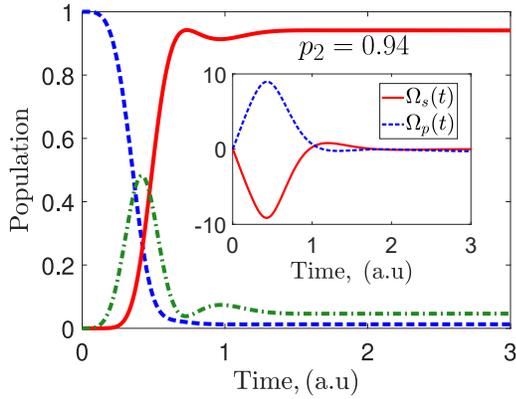


FIG. 10. Training the PINN considering one- ($\Delta_1/2\pi = 0.2$) and two-photon ($\delta/2\pi = 0.2$) detuning can easily improve the population transfer from $p_2 = 0.93$ to $p_2 = 0.94$.

FOUR-LEVEL SYSTEM

The optimization of control pulses is, in general, system-dependent. This means that once we have the optimal pulse sequence for a Λ -system, the extension to a four-level system is not straightforward. For instance, the new system involves more non-linear equations and the possibility of cross-talk to the newly added state. Henceforth, any analytical approach for optimization becomes a hard task. In this section we show that our PINN can handle a four-level system without modification to the architecture of the network. Then, we feed the PINN with the new set of differential equations, and we add a new constraint in L_{const} (36) for the population in the fourth state $\|z_4\|^2$. Our goal is to train a PINN that succeeds in transferring population from state $|1\rangle$ to state $|2\rangle$, with, (i) minimizing the population in the lossy intermediary state ($|3\rangle$), (ii) minimizing the pulse area, and (iii) minimizing cross-talk to state $|4\rangle$. Note that the last condition guarantees selectivity and it was not required for the three-level system.

To begin with, we consider the Hamiltonian of the system in a multi-rotating frame and in the eigenstate basis ($\hbar = 1$),

$$\hat{H} = \delta\sigma_{22} + \Delta_2\sigma_{33} + \Delta_4\sigma_{44} + \frac{\Omega_p(t)}{2}(\sigma_{13} + \sigma_{14}) + \frac{\Omega_s(t)}{2}(\sigma_{23} - \sigma_{24}) + h.c., \quad (57)$$

where $\Omega_p(t)$ and $\Omega_s(t)$ are the Rabi frequencies of the control fields and σ_{ij} the ladder operators. The one-photon detunings are $\Delta_2 = (E_3 - E_2 - \omega_s)$, $\Delta_3 = (E_3 - E_1 - \omega_p)$, $\Delta_4 = (E_4 - E_1 - \omega_p)$ and the two-photon detuning is $\delta_2 = (\Delta_3 - \Delta_2)$. E_i are the eigenenergies and ω_p and ω_s are the frequency of the control fields. This Hamiltonian can be obtained from the interaction of a Nitrogen-Vacancy center with a Carbon-13 nuclear spin [87, 88]. In Fig. 11 we show the results for our population transfer method.

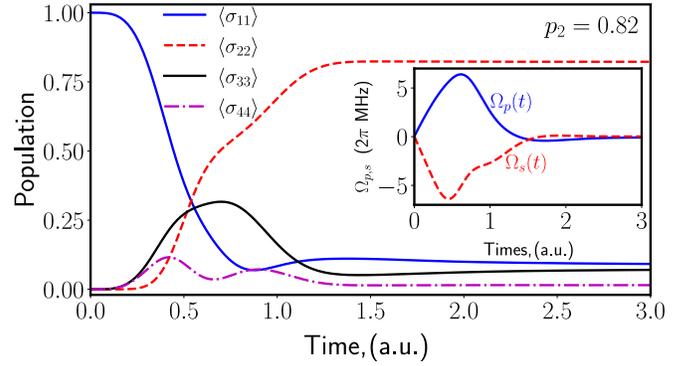


FIG. 11. PINN delivers a successful population transfer method that can be scaled to a four-level system without changing the network's architecture. We set $\Delta_2 = \Delta_3 = \delta = 0$ and $\Delta_4 = 6.79$.

LARGE QUANTUM SYSTEMS

Non-interacting qubits

Let us consider the following Hamiltonians

$$H_0 = -\frac{\omega_x}{2} \sum_{j=1}^N \sigma_{x,j}, \quad H_p = \frac{\omega_z}{2} \sum_{j=0}^N (1 - \sigma_{z,j}). \quad (58)$$

where $\sigma_{x,j}$ and $\sigma_{z,j}$ are the Pauli matrices for $S = 1/2$ of the j th qubit. Here, N is the number of qubits, and 2^N is the dimension of the Hilbert space. All qubits are identical with $\omega_x, \omega_z > 0$. Now, the target of this control problem is to minimize the expectation value of the final Hamiltonian. For that reason, we define

$$L_{\text{control}} = \langle H_p \rangle = \langle \Psi(t) | H_p | \Psi(t) \rangle, \quad (59)$$

Let us consider that $u_1(t) = g_0(t)$ and $u_2 = g_p(t)$ are two control functions that defines a real control vector $\mathbf{u}(t) = (u_1(t), u_2(t))^T$. The whole system satisfies the Schrödinger equation $i\hbar\partial_t|\Psi(t)\rangle = H(t)|\Psi(t)\rangle$, where $|\Psi(t)\rangle$ is the wavefunction. By using the decomposition into real and imaginary parts, *i.e.*, $|\Psi(t)\rangle = |\Psi(t)\rangle^R + i|\Psi(t)\rangle^I$ and $H(t) = H^R(t) + iH^I(t)$ ($H = H^\dagger$), we get the following non-autonomous dynamical system ($\hbar = 1$)

$$\dot{\mathbf{x}} = A(t)\mathbf{x}(t), \quad A(t) = \begin{pmatrix} H^I(t) & H^R(t) \\ -H^R(t) & H^I(t) \end{pmatrix}, \quad (60)$$

where $\mathbf{x}(t) = (|\Psi(t)\rangle^R, |\Psi(t)\rangle^I)^T \in \mathbb{R}^{2^{N+1}}$ is the real state vector required for the artificial neural network, and $A(t)$ is the dynamical matrix that depends on the control vector. Thus, we can apply our methodology to the dynamical system and find optimal control functions to reach the ground state of the final Hamiltonian.

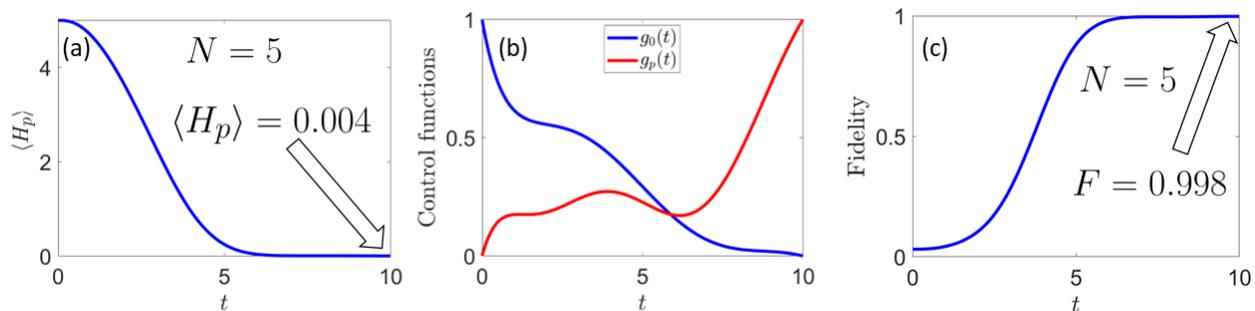


FIG. 12. (a) Expectation value $\langle H_p \rangle$ as a function of time. (b) Control functions obtained from our PINN approach. (c) Fidelity of the protocol.

For illustration, let's consider the case of five qubits ($N = 5$), where the dynamical system is highly non-trivial since we have 64 coupled equations of motion for the real vector $\mathbf{x}(t)$ ($2^{5+1} = 64$) or equivalently $2^5 = 32$ energy levels. For the simulation, we can fix $\omega_x = \omega_z = 1$, and the simulation time is chosen as $T = 10/\omega_z = 10$. Also, we choose 100 neurons and three hidden layers for the neural network architecture. In figure 12, we plot the solution of this control problem for five qubits and two independent control functions $g_0(t)$ and $g_p(t)$. We observe that the efficiency of the PINN approach in terms of the final fidelity amounts to 99.8%, demonstrating that our approach works even in larger quantum systems without interaction. At the simulation time T , our PINN yields an expectation value $\langle H_p \rangle$ that amounts to 0.004, which is close to the theoretical value $\min_{|\Psi\rangle} \langle H_p \rangle = 0$.

Interacting qubits

Let us consider the following Hamiltonians

$$H_0 = -\frac{\omega_x}{2} \sum_{j=1}^N \sigma_{x,j}, \quad (61)$$

$$H_p = \frac{\omega_f}{2} \sum_{j=0}^N (1 - \sigma_{z,j}) - J \sum_{i=1}^{N-1} \sigma_{z,i} \sigma_{z,i+1} \quad (62)$$

For the simulation, we consider the same quantum control problem explained in the previous section but using $N = 3$ qubits ($2^3 = 8$ energy levels). In this case, we use $\omega_x = \omega_f = 1$ and $J = \omega_f/4$ with a simulation time $T = 10/\omega_f = 10$. Also, we use the constraint $g_0(t) = 1 - g_p(t)$ to solve this problem.

In figure 13, we plot the solution of this control problem for three interacting qubits and two control functions $g_0(t)$ and $g_p(t)$ with the constraint $g_0(t) = 1 - g_p(t)$. We observe that the efficiency of the PINN approach in terms of the final fidelity amounts to 99.1%. Moreover, our PINN scheme reaches an expectation $\langle H_p \rangle$ value of -0.4839 , close to the theoretical value $\min_{|\Psi\rangle} \langle H_p \rangle = -0.5$. These results demonstrate that our approach performs well in larger quantum systems with interactions.

SUMMARY OF PINNS PARAMETERS

To summarize our results concerning the application of PINNs for quantum control, in Table II, we show the relevant information about how we implemented our method for different systems. In particular, we show the value for the hidden layers, epochs, neurons, control (η) and regularization (χ) weights, and learning rate (λ).

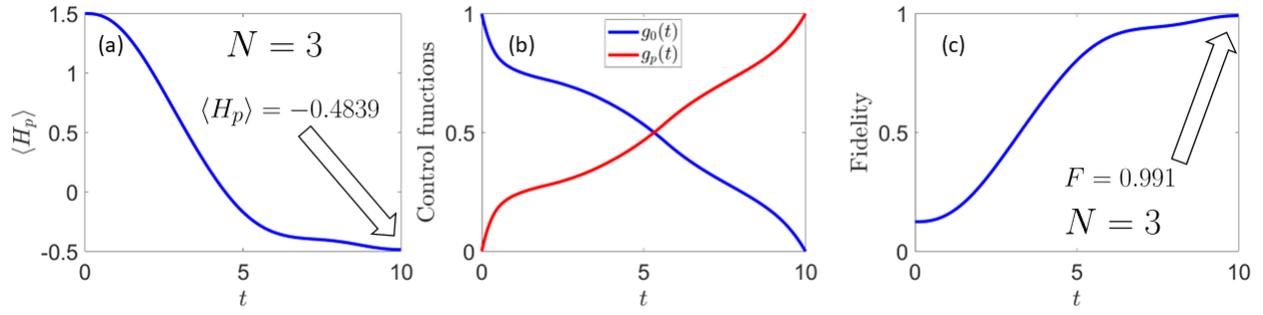
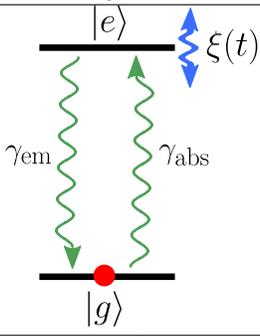
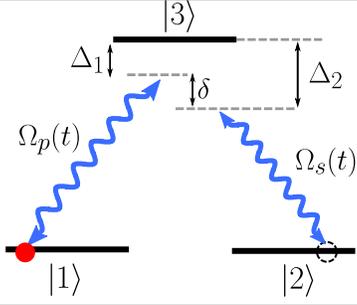
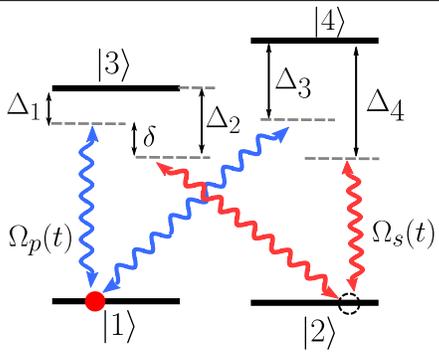
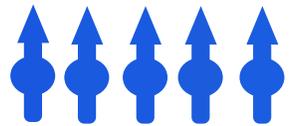
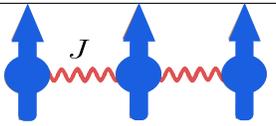


FIG. 13. (a) Expectation value $\langle H_p \rangle$ as a function of time. (b) Control functions obtained from our PINN approach using $g_0(t) = 1 - g_p(t)$. (c) Fidelity of the protocol.

TABLE II. The Table displays relevant information about the PINNs. Our PINNs are feed-forward neural networks with hard constraints on the initial conditions. η and χ stand for the control and regularization weights, respectively. λ accounts for the learning rate.

System	Hidden Layers	Epochs	Neurons	η	χ	λ
	4	4×10^4	200	0.1	10^{-3}	10^{-4}
	5	2×10^4	150	0.2	2.8×10^{-3}	8×10^{-3}
	5	2×10^4	150	0.2	2.8×10^{-3}	8×10^{-3}
	1	10^5	3	0.05	10^{-5}	10^{-3}
	4	5×10^5	100	0.01	10^{-4}	10^{-3}