

PAPER

Predicting the minimum control time of quantum protocols with artificial neural networks

To cite this article: Sofia Sevitz *et al* 2023 *Quantum Sci. Technol.* **8** 035023

View the [article online](#) for updates and enhancements.

You may also like

- [Glassy dynamics of a binary Voronoi fluid: a mode-coupling analysis](#)
C Ruscher, S Ciarella, C Luo et al.
- [Are polymers standard glass-forming systems? The role of intramolecular barriers on the glass-transition phenomena of glass-forming polymers](#)
J Colmenero
- [A comparative study of a class of mean field theories of the glass transition](#)
Indranil Saha, Manoj Kumar Nandi, Chandan Dasgupta et al.

Quantum Science and Technology



PAPER

Predicting the minimum control time of quantum protocols with artificial neural networks

RECEIVED
20 January 2023

REVISED
28 April 2023

ACCEPTED FOR PUBLICATION
15 May 2023

PUBLISHED
1 June 2023

Sofia Sevitz^{1,3,*} , Nicolás Mirkin¹  and Diego A Wisniacki^{1,2}

¹ Universidad de Buenos Aires, Facultad de Ciencias Exactas y Naturales, Departamento de Física, Buenos Aires, Argentina

² CONICET—Universidad de Buenos Aires, Instituto de Física de Buenos Aires (IFIBA), Buenos Aires, Argentina

³ Present affiliation: University of Potsdam, Institute of Physics Astronomy, Karl-Liebknecht-Str. 24-25, 14476 Potsdam, Germany.

* Author to whom any correspondence should be addressed.

E-mail: sofia.sevitz@uni-potsdam.de

Keywords: quantum control, machine learning, artificial neural networks

Abstract

Quantum control relies on the driving of quantum states without the loss of coherence, thus the leakage of quantum properties into the environment over time is a fundamental challenge. One work-around is to implement fast protocols, hence the Minimal Control Time (MCT) is of utmost importance. Here, we employ a machine learning network in order to estimate the MCT in a state transfer protocol. An unsupervised learning approach is considered by using a combination of an autoencoder network with the k -means clustering tool. The Landau–Zener (LZ) Hamiltonian is analyzed given that it has an analytical MCT and a distinctive topology change in the control landscape when the total evolution time is either under or over the MCT. We obtain that the network is able to not only produce an estimation of the MCT but also gains an understanding of the landscape's topologies. Similar results are found for the generalized LZ Hamiltonian while limitations to our very simple architecture were encountered.

1. Introduction

The current century promises an advent of revolutionary technologies based on quantum information and quantum computing [1, 2]. Quantum control is a crucial part in the advances of these fields given that it dictates how a system should be manipulated in order to achieve a target goal [3]. A possible approach is by applying optimal control techniques [4, 5]. This way the problem is reduced to finding a time dependent control field $\epsilon(t)$ that maximizes a certain objective function, $J[\epsilon(t)]$. This must be done while preserving the coherence of the system, thus the leakage of quantum information onto the environment over time is a fundamental challenge. An efficient way to work around the difficulty is to implement rapid controls, that is to say, to maximize $J[\epsilon(t)]$ in the fastest time possible [6].

The speed of these protocols are bounded by the minimum evolution time needed for a system to reach a target state driven by a time-dependent control field, namely Minimum Control Time (MCT), not to get confused with the quantum speed limit which is a lower bound dictated by quantum mechanics for a system to evolve from an initial to a final state [7–10]. Having an estimation of the MCT is of uttermost importance given that it governs the success of a control protocol in the presence of the detrimental effects from the environment. While conceptually easy, in practice it is a whole other story. One approach is by utilizing the quantum brachistochrone formalism, see [11, 12]. Another path is by analyzing the solutions of the optimal control scheme for different values of T , in other words, by analyzing the control landscapes [13].

Many efforts have been made in developing optimal control methods, especially from a numerical approach. In this framework, the studies of quantum control landscapes are a key factor. They are essentially the mapping of the objective function $J[\epsilon]$ on the field ϵ parameterized in time [14]. Their topologies are not trivial, for instance, local maximums or traps appear whenever the control problem holds a constraint [15]. On top of that, usually, optimal controls require very high parameterization dimensions that produce

landscapes beyond our intuitive understanding [16]. Thus, extracting useful information from them might be an unimaginable task.

Over the past years, various optimization algorithms have been developed such as gradient-ascent pulse engineering algorithm and chopped random-basis quantum optimization, see review in [4]. Recently, machine learning has been studied as an alternate optimization tool and in other quantum problems [17–25]. For example, in [26] a reinforcement learning algorithm has been proposed in order to find an optimal driving protocol for a state transition scheme. Another case is [27], where a supervised learning algorithm classifies randomness of a system in order to find an optimal control policy.

In this paper we use machine learning but not as an alternate optimization tool. We instead aspire to answer if an artificial neural network (ANN) is able to estimate the MCT just by training over the landscape of a system. On that note, we are also interested in the exploration if an ANN actually understands the topology of the landscape or merely detects that an extremity is reached. Our aspiration is to shed light on what ANN are able to tell us about physical insights within quantum control. This interpretability notion has been explored in other research fields see [28–31].

Here we implement an unsupervised ANN scheme is selected in order to avoid any external bias, thus any information obtained from the network is purely due to the information stored within the dataset. An autoencoder extracts the important features of the dataset by means of dimension reduction that are later classified with the use of the k -means clustering tool. As a first approach, the toy model Landau–Zener (LZ) Hamiltonian is analyzed given that it has an analytical MCT and a distinctive change in the landscape’s topologies when the total evolution time is either under or over the MCT. We obtain that the ANN is not only able to predict the MCT but also gains an understanding of the landscape’s topologies. We moved on to implement the network in the generalized LZ Hamiltonian where similar results are found beyond some limitations encountered within our simple scheme.

The paper is organized as follows. In section 2 we present the models used in this work, LZ Hamiltonian and its generalization, while giving an overview of the optimal control framework. The unsupervised neural network approach implemented is detailed in section 3. The results obtained for the LZ Hamiltonian and its generalization are given in section 4. Finally, in section 5 we communicate the conclusions and aspirations for future works.

2. Optimal control framework and models

Quantum control theory is the theoretical framework that studies the manipulation of quantum systems. A typical driving Hamiltonian is given in the following way,

$$H[\epsilon(t)] = H_0 + \epsilon(t)H_c, \quad (1)$$

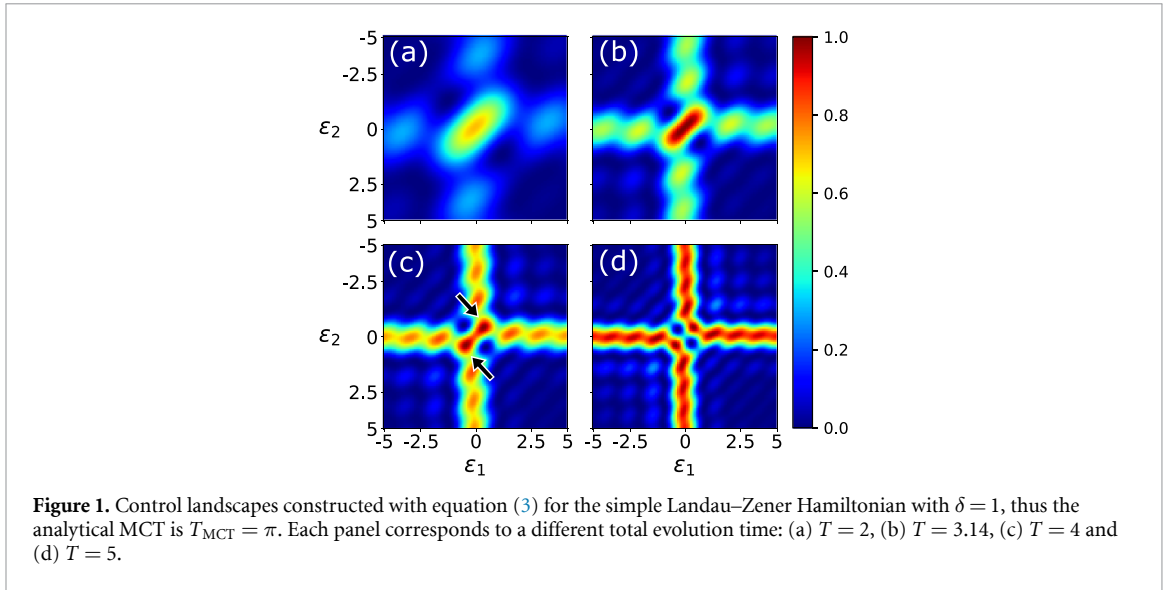
where H_0 is the isolated system one wants to manipulate, known as drift Hamiltonian, H_c the control Hamiltonian and $\epsilon(t)$ a time-dependent control field parameterized in time [32]. The evolution of the system is governed by the shaping of said field, also referred to as protocol. In this work, we consider the state transfer scheme; we seek to achieve a desired target state $|f\rangle$ from an initial state $|i\rangle$ after a given evolution time T . The protocol duration is divided into N_{ts} uniform intervals of $\Delta t = T/N_{ts}$ and a piecewise function is used to parameterize the control,

$$\epsilon(t) = \begin{cases} \epsilon_1 & \text{if } 0 < t \leq \Delta t \\ \vdots & \\ \epsilon_{N_{ts}} & \text{if } (N_{ts} - 1)\Delta t < t \leq T. \end{cases} \quad (2)$$

The measure used in this work to quantify how well a control protocol performs is the state fidelity, given by

$$F[\epsilon] = |\langle f|U_\epsilon(T)|i\rangle|^2 \in (0, 1). \quad (3)$$

Note that $U_\epsilon(T)$ is the evolution operator given by the Hamiltonian in equation (1) at time T , thus it is dependent of the protocol. A simple view of equation (3) discloses that $F = 1$ corresponds to an optimal control, the achieved state after a given evolution time is equal to the desired one.



In this work, we consider the paradigmatic LZ Hamiltonian where the drift and control Hamiltonian are the following

$$H_0 = \frac{\delta}{2}\sigma_x \quad \text{and} \quad H_c = \sigma_z. \quad (4)$$

Here σ_x and σ_z are the Pauli matrices, δ is referred to as the energy gap given that it measures the minimum separation between the energies of the system under no perturbation [32]. From now on we use the natural units such that $\hbar = 1$, thus the units of time T are $[1/\delta]$. This Hamiltonian is of particular interest within the field given that it describes a simple two level system that can be applied, for example, to a semiconductor quantum dot [33]. This model is a great starting point given that not only it has a closed solution for its minimum time [34] but also a rather simple landscape because that optimal solutions are reached with just a two-dimensional parameterization, $N_{\text{ts}} = 2$ [32].

The typical transfer scheme is to take the initial state $|0\rangle$ and the desired final state $|1\rangle$, in the σ_z base. In [34] it was demonstrated that the MCT for the scheme is given by

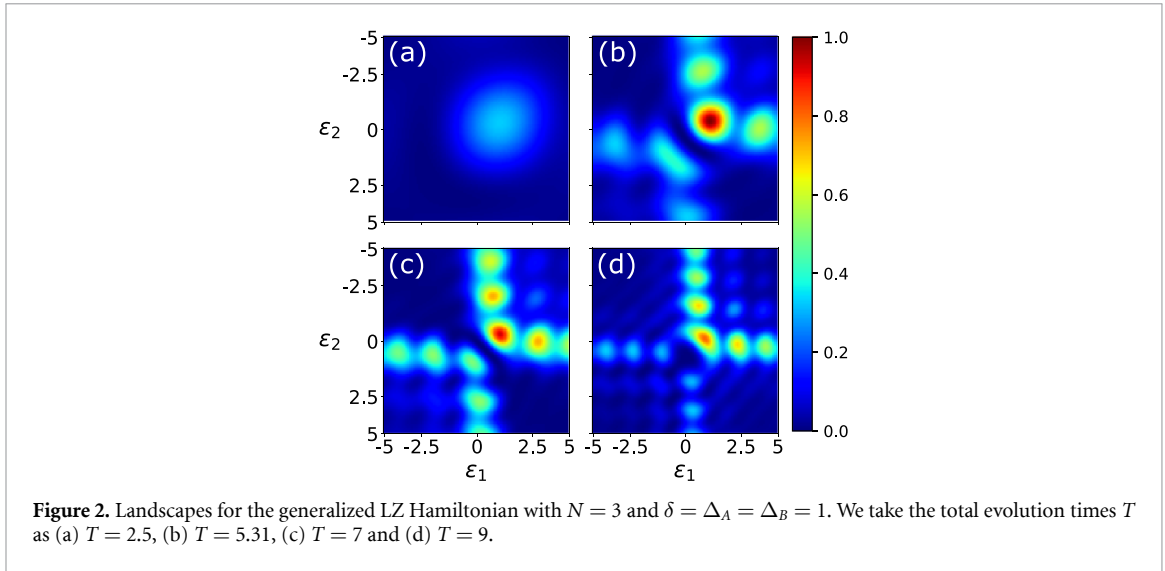
$$T_{\text{MCT}} = \frac{\pi}{\delta}. \quad (5)$$

Taking advantage that we can visualize optimal landscapes of these systems, in this work we start off concentrating in the simple case $N_{\text{ts}} = 2$. We parameterize the control field as a piecewise function in the following way

$$\epsilon(t) = \begin{cases} \epsilon_1 & \text{if } t \leq T/2 \\ \epsilon_2 & \text{if } t > T/2. \end{cases} \quad (6)$$

The controls (ϵ_1 and ϵ_2) are broken down into mesh points that conform the domain of the fidelity function forming the landscape. In figure 1 we show the mapping obtained for different evolution times: (a) $T < T_{\text{MCT}}$, (b) $T \simeq T_{\text{MCT}}$, (c) $T > T_{\text{MCT}}$ and (d) $T \gg T_{\text{MCT}}$.

When $T < T_{\text{MCT}}$ (figure 1(a)) the landscapes present a sub-optimal ($F < 1$) single global maximum at the origin. Whereas, when the system has reached the MCT (figure 1(b)), there is an optimal global maximum in the origin corresponding to $F = 1$. This indicates that an optimal control is achieved when taking the protocol $\epsilon(\epsilon_1, \epsilon_2) = 0$. Beyond this total evolution time, figure 1(c), interesting and complex topologies appear. The center maximum is splitted into two symmetrical structures (highlighted with black arrows) that oscillate in time. Also, many sub-optimal maximums form along the vertical and horizontal direction that transform into optimal maximums over time (see figure 1(d)). What is important to note from this system is that there is a distinctive difference in the landscape's topologies with $T < T_{\text{MCT}}$ and $T > T_{\text{MCT}}$.



The model of LZ can be generalized to an N -level system studied in [35]. In this work, we consider the case where $N = 3$ in which the driving Hamiltonian in the base $\{|0\rangle, |1\rangle, |2\rangle\}$ can be written as

$$H_0 = \frac{1}{2} \begin{pmatrix} 0 & \Delta_A & 0 \\ \Delta_A & 0 & \Delta_B \\ 0 & \Delta_B & -2\delta \end{pmatrix} H_c = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \quad (7)$$

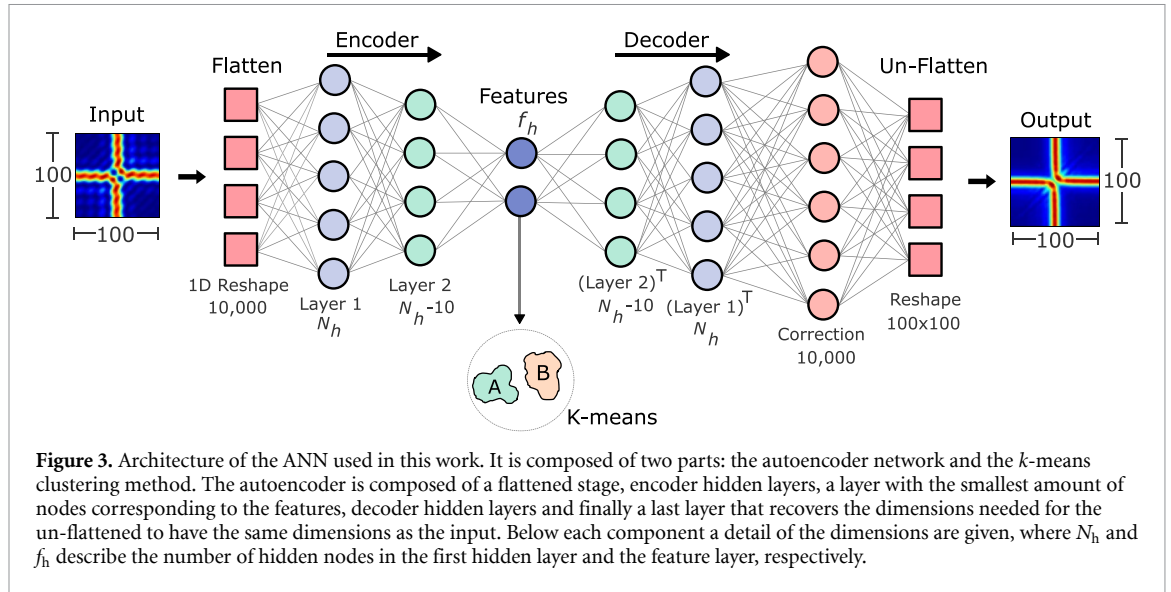
This Hamiltonian is commonly used in quantum optics given that it is suitable for describing a three-level system that leads to a STIRAP protocol when Δ_A and Δ_B are the controls [36]. However, in this work we take Δ_A , Δ_B and δ as fixed parameters. Under these conditions, the energy spectrum as a function of the control holds two avoided crossings, as opposed to the traditional LZ Hamiltonian that only has one. Although there is no closed analytical MCT solution in this case, we are able to obtain an empirical estimation from the control landscapes by taking the total evolution time of the landscape that reaches the highest fidelity within the dataset. This way, for $\delta = 1$, we define an empirical MCT, $T_{\text{MCT}} = 5.31$. Please refer to the appendix A for further details as to how the value was calculated.

The landscape topologies are very different for this system (figure 2). In particular, they do not hold symmetric topologies and the global maximum is shifted as a function of δ . More importantly, unlike LZ, this system does not exhibit a distinctive change in the landscape for when $T < T_{\text{MCT}}$ and $T > T_{\text{MCT}}$. For instance when $T \gg T_{\text{MCT}}$, the region of the landscape we are exploring does not hold an optimal control protocol. Thus, in this sense, we believe that this is a good system to test the generalization properties of our proposed ANN.

3. The unsupervised artificial neural network method

In this work, we would like to explore if an ANN can estimate the MCT by understanding a topology change within a landscape or simply detects that a pixel of the landscape obtains the value of ≈ 1 . Given that we do not want to add any external bias to the network and we usually do not know which is the real value of the MCT, we opt for a fully unsupervised approach by using a combination of an autoencoder network with k -means clustering. The architecture is illustrated in the figure 3. A similar implementation has been used in [37] under another context.

The autoencoder is a typical data reduction network. Its main task is to optimize the network such that the input layer is the same as the output layer while generating an effective representation of the data, commonly called features. Given that the network evaluates the loss function comparing its output to its input, the training is done in an unsupervised or self-supervised fashion [38]. The unsupervised k -means clustering method is then implemented over these extracted features and, by means of competition clustering, obtains a *natural* grouping of data. Note that we use an autoencoder instead of directly clustering over the control landscapes, this is because we wish to develop a network that detects subtle changes in the dataset and be represented in the features. Further details of the network are given in appendix B.



The input dataset is composed of a collection of control landscapes with different evolution times T . Given that the first layer of the network flattens the dataset into a 1-dimensional array, this enables us to generalize the architecture for any parameterization dimension (specifically $N_{ts} > 2$). The input data is randomly shuffled and separated using the holdout method, 70% of the data is used for training while the remaining 30% is reserved for validation. In order to avoid a bias training in the autoencoder and k -means, the training datasets are equally separated into two parts: autoencoder training dataset and k -means training dataset. For the same reason, the separation is also done for the validation dataset into autoencoder validation dataset and performance dataset. See appendix B for further information.

The data manipulation consists of three steps: (1) autoencoder training, (2) k -means unsupervised clustering training and (3) the performance measure. In the following, we will break down each of these steps. In step (1), the autoencoder is trained using backpropagation with the classical mean squared error (MSE) loss function. The training parameters are detailed in the appendix B. Once the autoencoder is optimized, we move on to step (2). The features of both the cluster training dataset and the performance dataset are extracted. The cluster training features are then used to train the k -means model. In this step we implement the elbow method to optimize the number of clusters and obtained two clusters, ideally we expect the classification of the landscapes to be corresponding to $T < T_{MCT}$ and $T > T_{MCT}$ (see appendix B). The performance features are to be used in step (3) in order to evaluate an overall performance of the network. Given that we are taking an unsupervised approach there are no *correct labels* to evaluate a performance measure, thus we appeal to an alteration of the confusion scheme presented in [39]. If the dataset one is trying to classify into two groups depends on a parameter that lies within a closed range, in our case the total evolution time T , then one can assume that there exists a critical point T' (within this range) that optimally classifies the dataset into the two groups. In order to find this T' , one can take an auxiliary critical point T_{aux} and create an auxiliary labeling of all data with parameters smaller than T_{aux} with label A and the others with label B. Next, by varying T_{aux} , the performance curve is analyzed. In this work the performance measure is the well used accuracy score. Although not the same scheme as presented in [39] we can still observe that the performance function with respect to the T_{aux} has an universal shape, with a maximum at the correct critical point T' .

Ensemble averaging is implemented to smooth the output performance function. This method consists in combining many networks in order to form a collection of networks. The ensembles are composed of different combinations of number of nodes in the first hidden layer $N_h \in [100, 110, 120, \dots, 190]$ and number of features extracted $f_h \in [10, 20, \dots, 40]$, this way we take a total of 40 architectures. The performances of each of the architectures are later averaged in order to obtain a single mean clustering accuracy.

4. Estimation of the Minimal Control Time

In this section we analyze the application of an ANN framework over the LZ Hamiltonian and its generalization presented in section 2. We commence with the landscapes from the LZ Hamiltonian, equation (4), with the parameterization $N_{ts} = 2$ and $\epsilon_{\{1,2\}} \in [-5, 5]$ with 100 points (see figure 1 for

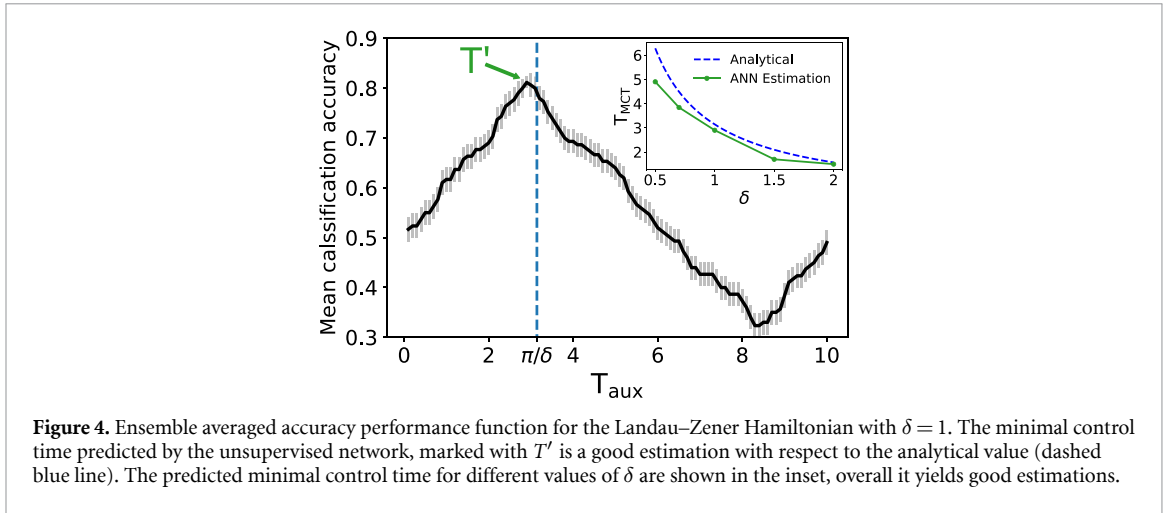


Figure 4. Ensemble averaged accuracy performance function for the Landau–Zener Hamiltonian with $\delta = 1$. The minimal control time predicted by the unsupervised network, marked with T' is a good estimation with respect to the analytical value (dashed blue line). The predicted minimal control time for different values of δ are shown in the inset, overall it yields good estimations.

reference). We chose a sweep of total evolution times T from 0.01 to 10 in steps of 0.01, which yields a dataset of 1000 landscapes (see appendix B for more details).

In figure 4 we show the smoothed output performance function taking the energy gap $\delta = 1$. The critical point is located slightly below the analytical MCT, with an estimated value of $T' = 2.9$ where the analytical MCT is $T_{\text{MCT}} = \pi$. In the inset of figure 4 we show the ANN estimation as a function of the energy gap δ . As we can see, the unsupervised network gives a good prediction of the MCT, that is to say, without any external guidance regarding the total evolution time of the landscapes, the network is able to identify a change within the time shuffled landscapes. We point out that in the main panel a minimum is formed near $T_{\text{aux}} = 8.1$, further investigation is given in the appendix C.

In the following we focus on understanding the underlining knowledge that the ANN gains over the datasets. We analyze the weights between the nodes in the first layer of the ANN and the pixels in the landscapes (see inset of figure 5(a) for reference). As mentioned in section 3, the collection of networks for ensemble averaging is composed of the different combinations of number of nodes in the first hidden layer and number of features extracted. Each one of these combinations form a specific architecture ν . Given an activation function φ , the output function of each node y_i^ν for a given architecture ν can be written as

$$y_i^\nu = \varphi \left(\sum_j \omega_{ij}^\nu x_j \right), \quad (8)$$

where x_j is the pixel in the position j with the landscape reshaped into a 1-dimensional array and the weight between the input x_j and the node y_i^ν is represented by ω_{ij}^ν . From equation (8) one can interpret the absolute value of each weight (i.e. $|\omega_{ij}^\nu|$) as a measure of strength of the connection between x_j and y_i^ν . Thus, the most important pixels for the network correspond to those that have a higher $|\omega_{ij}^\nu|$ compared to the rest. This interpretation is similar to the understanding of filters in convolutional neural networks [40].

Figure 5(a) is composed of 40 subplots stacked on top of each other (separated by black horizontal lines) corresponding to each architecture. Each subplot shows the absolute values of the weights for all the nodes against the pixels in the flattened landscape. For instance, the first subplot (marked with a purple arrow) contains the 190 node weights corresponding to the architecture with 40 features. The second subplot (red arrow) corresponds to the architecture with also 190 nodes but 30 features. This way all the 40 architectures are stacked on top of each other. In order to compare between architectures, we normalize over the maximum absolute weight of each architecture, $|w_{ij}^\nu| / \max_{kl} |w_{kl}^\nu|$. As explained, this plot gives an idea of the strength of the connection between the node i and the pixel j . The network shows a symmetric behavior with respect to pixel $x_j = 5000$. This symmetry is attributed to the symmetrical aspect of the landscapes of the LZ Hamiltonian (see figure 1). In the following, we select all the pixels x_j that satisfy the following condition: $\left\{ x_j \mid \frac{1}{N} \sum_{i,\nu} \left(|w_{ij}^\nu| / \max_{kl} |w_{kl}^\nu| \right) \geq 0.7 \right\}$, where N is the total amount of architectures. This selection identifies the regions of the landscapes where the network is paying the most attention. As an illustration, in figure 5(b) we take the landscape at $T = 4$ and highlight in black the region of interest to the network. The ANN detects

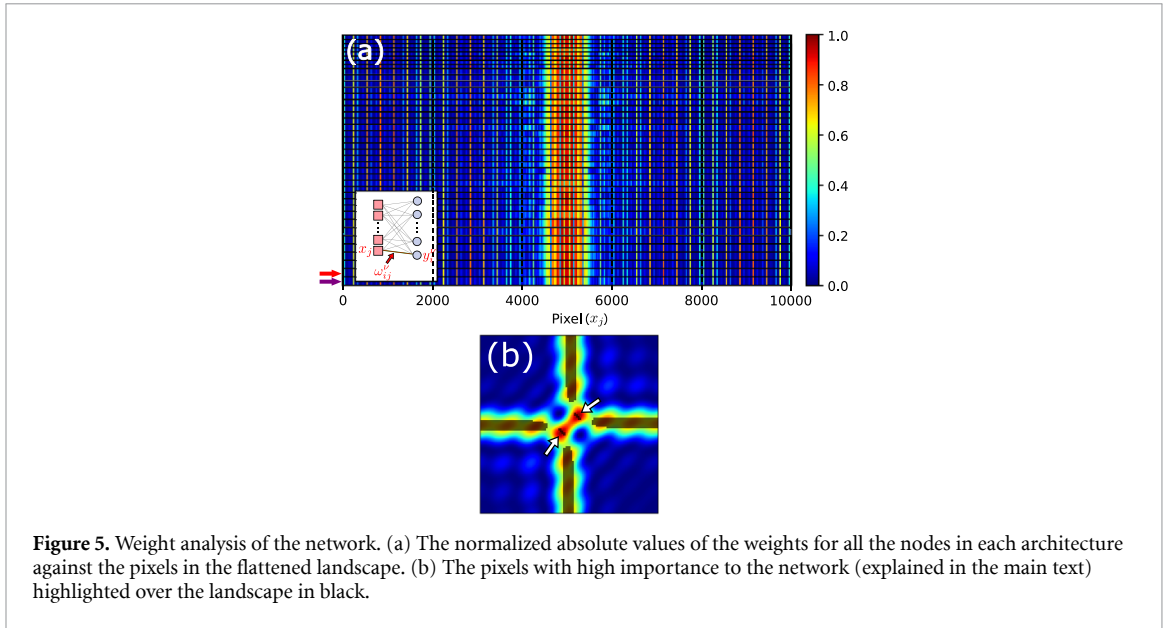
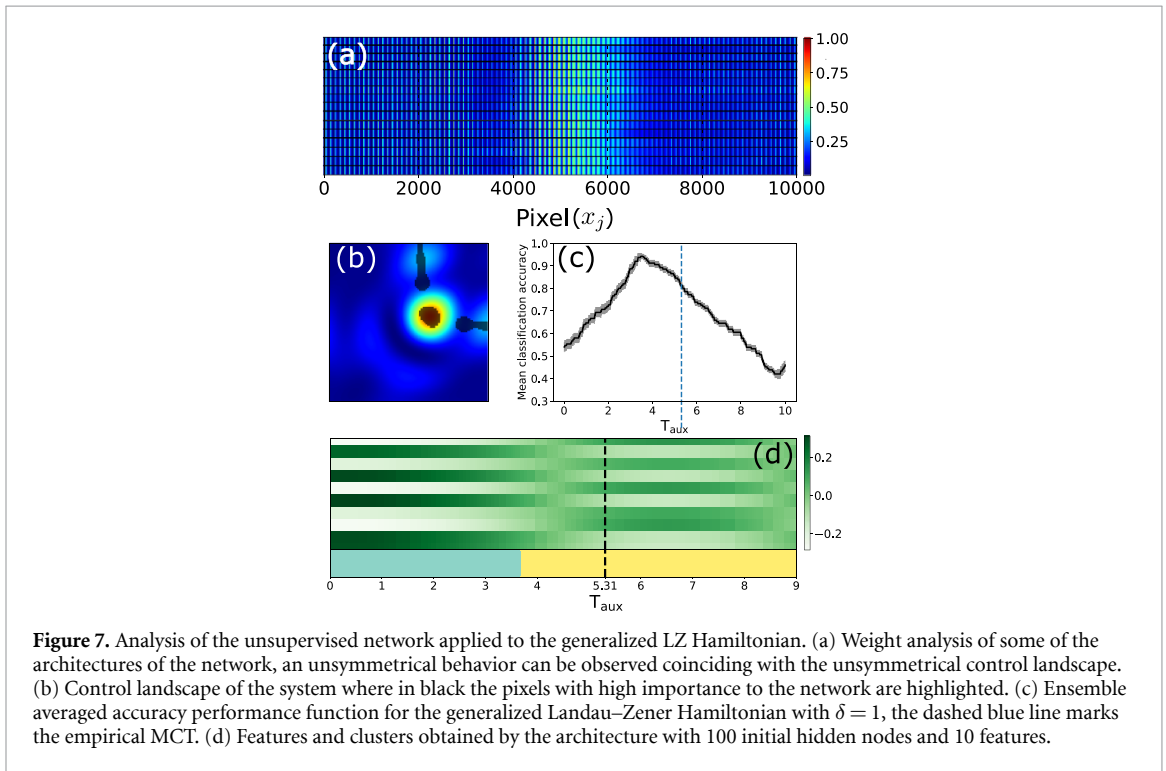
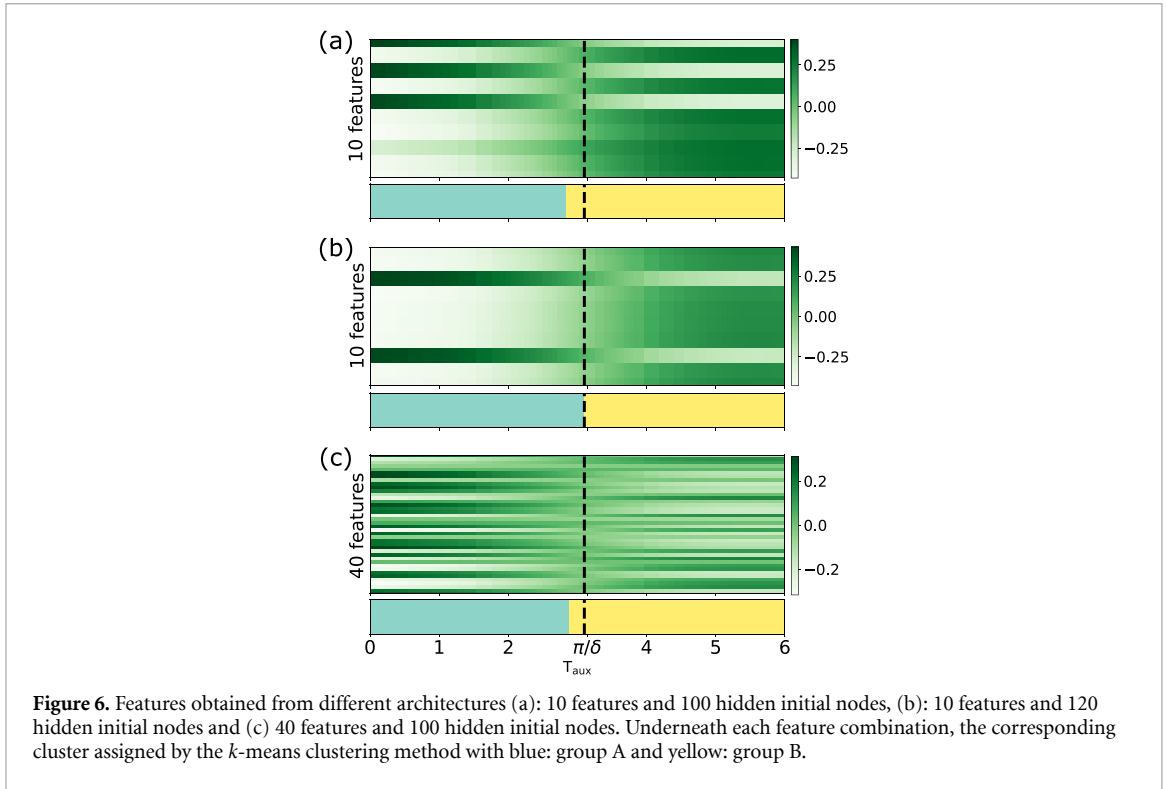


Figure 5. Weight analysis of the network. (a) The normalized absolute values of the weights for all the nodes in each architecture against the pixels in the flattened landscape. (b) The pixels with high importance to the network (explained in the main text) highlighted over the landscape in black.

the symmetrical structures that form when the optimal maximum is separated into two (white arrows in figure 5(b)). The network prioritizes the horizontal and vertical direction corresponding to the sub-optimal maximums that form after the total evolution time T reaches the MCT. This indicates that the ANN also understands that a change in the topology of the landscape corresponds to a change in an underlying parameter of the system, in this case T . It is an exciting result given that this ANN can give insights of landscapes that are difficult to analyze visually, for instance, taking landscapes with $N_{ts} > 2$. With this simple architecture we tested the LZ Hamiltonian with a $N_{ts} = 3$ parameterization. Given that it is computationally demanding to construct the landscapes the larger the N_{ts} , we took a broad discretization of the mesh points; $\epsilon_{\{1,2\}} \in [-5, 5]$ with 100 points and $\epsilon_3 \in [-5, -4, \dots, 5, 6]$ with 11 points. This yielded a predicted $T' = 4.34$, which we believe can be improved by taking a finer mesh discretization or adding complexity to our ANN. We note that this analysis was only possible because of the flatten stage of our ANN scheme, it would not have been possible if we chose, for instance, the widely used two dimensional convolutional neural networks.

Next we analyze the features obtained from different architectures with their corresponding cluster assigned by the unsupervised k -means clustering. Figures 6(a) and (b) exhibit the 10 features assigned to each time for two different architectures: 100 and 120 initial hidden nodes, respectively. As seen, the features extracted from the landscapes corresponding to a total evolution time $T < T_{MCT}$ are very different from those corresponding to $T > T_{MCT}$. Near the theoretical T_{MCT} (black dashed line) the features suffer a transition. We remind the reader that this is obtained with an unsupervised network and the landscapes are presented to the network in a shuffled time-order. This indicates that although we are not using feedback connections (typically used for time-dependent problems) our network still understands an underlying time order. The same was observed for all the architecture configurations, for instance, in figure 6(c) we exhibit the case with 40 features and 100 hidden nodes in the initial layer. An interesting study is the quest of a *good* set of hyperparameters (number of nodes, layers and features) such that the features have the physical meaning of the problem. Such exploration will be included in future works.

We now apply the unsupervised network to the generalized LZ Hamiltonian presented in section 2. In figure 7(a) the weight analysis of the network is presented. As mentioned in section 2 the landscapes of this control do not present symmetric topologies which is reflected in $|\omega_{ij}^y|$. As before, we would like to relate the most important regions of the landscape to the network, thus we take all the pixels x_j that satisfy the same condition used in the LZ Hamiltonian analysis but with a threshold of 0.5. The results are highlighted in black in the figure 7, where we take the total evolution time $T = 4$ as an example. In this case, the network identifies where the region control landscape reached the global maximum and, similar to before, it also pays attention to where the sub-optimal mountains are formed thus this gives us the belief that the network is able to understand the topologies. In section 2 we defined an empirical MCT $T_{MCT} = 5.31$ which is compared to the MCT predicted by the ANN in figure 7(c). Despite the fact that the prediction is lower than the empirical value (blue dashed line), we still were able to identify key factors of the networks understanding of the



landscapes where a region of interest as to where the MCT is located. We note that these results were only observed for the case of $\delta = 1$. For the other cases the network fails to estimate a MCT, we suspect that this is due to the simplicity of the network. We understand that there are more sophisticated ANN frameworks available, such as convolution neural networks or long short term memory networks which will probably

yield more accurate results but as this is one of the first works covering the idea, we wanted to investigate with the simplest case we could find.

5. Conclusion

The external driving of quantum systems is extremely vulnerable in the presence of environments due to decoherence. A work around to this challenge is to implement fast controls, thus the knowledge of the MCT is of much importance in order to obtain revolutionary advances in quantum technologies. Despite its importance, few methods have been developed both from an analytical and numerical approach. In this paper, we propose to employ machine learning techniques to estimate the MCT of a protocol while studying what an ANN is able to learn from a system as well as its limitations. In this sense, in order to not give the network any external bias, we tested a fully unsupervised ANN scheme composed of an autoencoder and a k -means clustering method. We analyzed the LZ Hamiltonian given that it has an analytical MCT and a distinctive change in the landscape's topologies when the total evolution time is under or over the MCT. We obtained that the network is able to not only produce an estimation of the MCT but also to gain an understanding of the landscape's topologies. We moved on to implement the network in the generalized LZ Hamiltonian where similar results were yielded and also found some limitations to our very simple architecture.

We believe this is the first approach into understand how and what the ANN does and ultimately shed light on its limitations. We would like to remark that here we generated an unsupervised network that is able to detect topologies changes in the control landscapes, even really subtle ones. As to the practicability of the implementation, we believe that in the low parameterization (coarse-grained modeling) the landscapes exhibit slight characteristics that the network is able to detect in order to generate a region of interest where the MCT is located. Once this region is defined, a more sophisticated method could be applied to refine the estimation. More work is still to be done in the future, specifically testing its limitations, modification of the architecture to enhance its generalization and interpretation and an analysis of the computational resources. We hope that our findings will serve as the foundations to further investigate and exploit the underlying abilities of the ANN.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

Acknowledgments

The authors would like to acknowledge the useful conversations with Emiliano Fortes. The work was partially supported by CONICET (PIP 11220200100568CO), UBACyT (20020170100234BA) and ANPCyT (PICT-2020-SERIEA-01082).

Appendix A. Empirical MCT of the generalized LZ Hamiltonian

In this section we give further details as to how we define an empirical MCT from the generalized LZ Hamiltonian defined by equation (7) in the main text.

Although there is no closed analytical MCT solution in this case, we are able to obtain an empirical estimation from the control landscapes by taking the total evolution time of the landscape that reaches the highest fidelity within the dataset. In figure 8 we plot the maximum fidelity achieved in each a landscape as a function of the total evolution time T for $\delta = 1$. This way an empirical MCT is defined by selecting T that holds the highest fidelity (marked with a vertical dashed line).

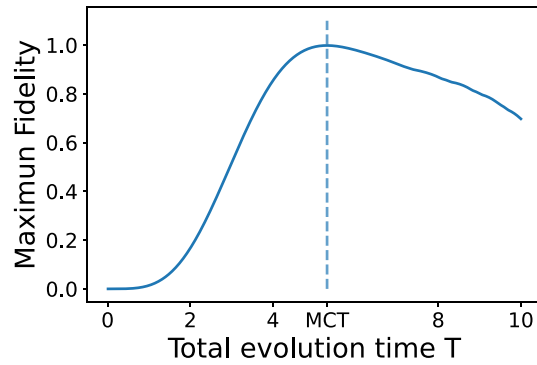


Figure 8. Maximum fidelity of the landscape corresponding to each total evolution time T for $\delta = 1$. In dotted lines, the empirical MCT is highlighted corresponding to $T_{\text{MCT}} = 5.31$.

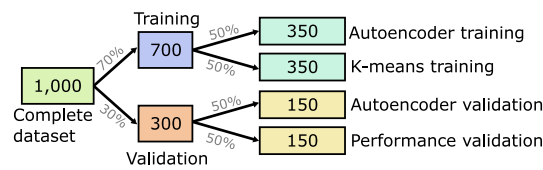


Figure 9. Illustration of the separation of the dataset for training and validation. The holdout method is implemented where 70% of the dataset is used for training and the remaining 30% is reserved for validation. Then the training datasets is equally separated into two parts: autoencoder training dataset and k -means training dataset. The same is done with the validation dataset separating into the autoencoder validation dataset and performance dataset.

Appendix B. Artificial neural network details

In this section we will give further details as to the creation of the landscapes used as the datasets as well as the ANN used in this work.

The control landscapes used as the datasets of the ANN were constructed using the QuTiP toolbox version 4.7.0 [41]. Each landscape corresponds to a certain total evolution time, $T \in [0.01, 0.02, 0.03, \dots, 10]$ yielding a dataset of 1000 components. The separation into training and validation subsets was done using the shuffled holdout method; 70% of the data is used for training while the remaining 30% is reserved for validation. In order to avoid a bias training in the autoencoder and k -means, the training datasets were equally separated into two parts: autoencoder training dataset and k -means training dataset. The same is done with the validation dataset separating into the autoencoder validation dataset and performance dataset. Figure 9 shows a visual illustration of the dataset separation to give clarity. Once the datasets are constructed, we move onto the construction of the ANN.

The autoencoder was developed using the Keras interface that runs over the Tensorflow platform version 2.6.0 [42]. The architecture (figure 3 in the main text) is composed of a flattened stage, encoder hidden layers, a layer with the smallest amount of nodes corresponding to the features, decoder hidden layers and finally a last layer that recovers the dimensions needed for the un-flattened to have the same dimensions as the input. The code allows for a deep network with many hidden layers, but for the purpose of this paper, we only required a shallow network with two encoder/decoder hidden layers. As described in the main text, ensemble averaging was implemented. For this, various architectures are to be constructed in order to form a collection of networks. This is done by building different combinations of number of nodes in the hidden layer $N_h \in [100, 110, 120, \dots, 190]$ and number of features extracted $f_h \in [10, 20, \dots, 40]$, which yields a total of 40 architectures. The non-linear hyperbolic tangent activation function is implemented over the encoder and decoder layers in order to generate complex solutions and not restrict ourselves to principal component analysis networks. In order to prevent overfitting while avoiding the underfitting regimen the classical L2-norm regularization term is implemented over the encoder/decoder layers with the hyperparameter $\alpha = 0.005$. We show that there is no overfitting by training the LZ Hamiltonian, equation (4), with $\delta = 0.7$ and predicting over another system, for instance taking $\delta = 0.5$ and $\delta = 1$ (figure 10(a)). The last layer has a linear activation function and its only purpose is to recover the dimensions needed for the un-flatten layer to reshape into same dimensions as the input landscape. The training is done with the Adam stochastic gradient optimizer with 100 epochs and a batch size of 32, implementing the MSE loss function and a learning rate of 0.001. With these configurations, a typical training and validation plot is shown in figure 10(b). In average, at

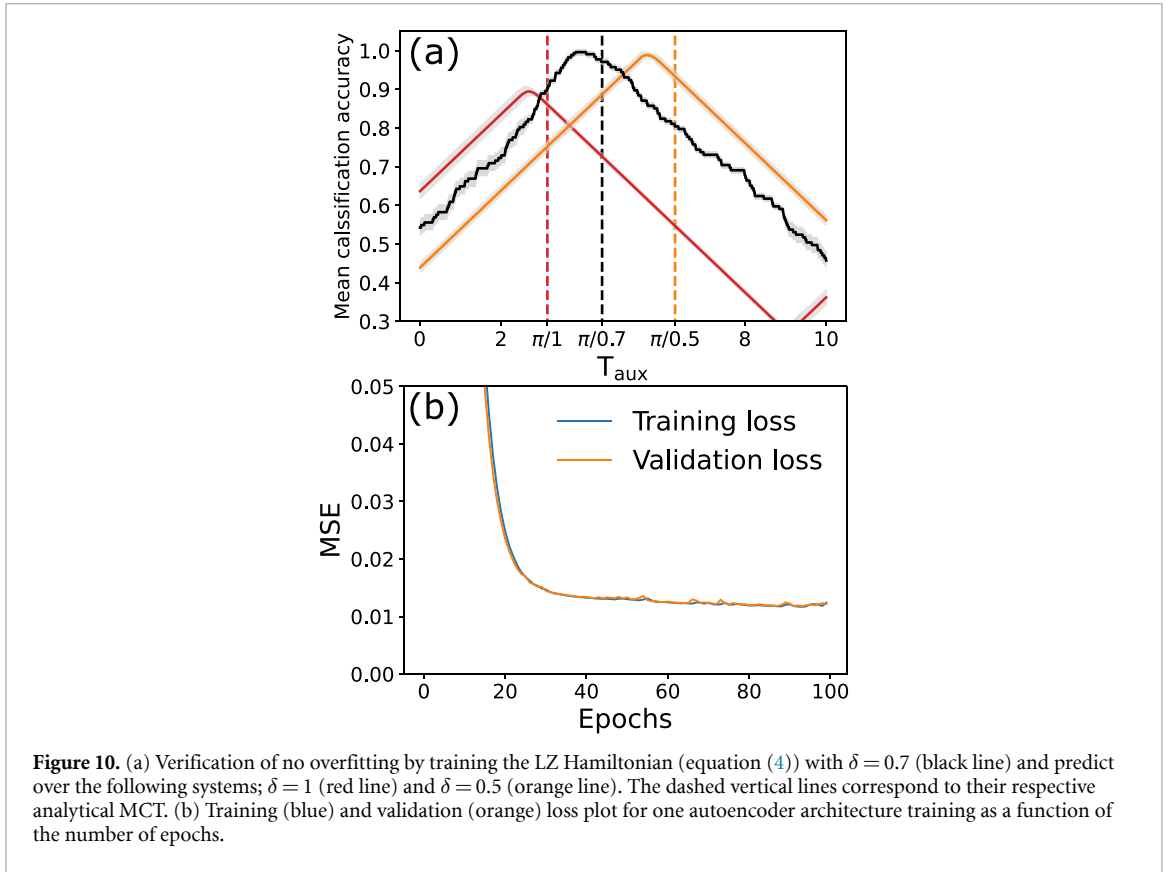


Figure 10. (a) Verification of no overfitting by training the LZ Hamiltonian (equation (4)) with $\delta = 0.7$ (black line) and predict over the following systems; $\delta = 1$ (red line) and $\delta = 0.5$ (orange line). The dashed vertical lines correspond to their respective analytical MCT. (b) Training (blue) and validation (orange) loss plot for one autoencoder architecture training as a function of the number of epochs.

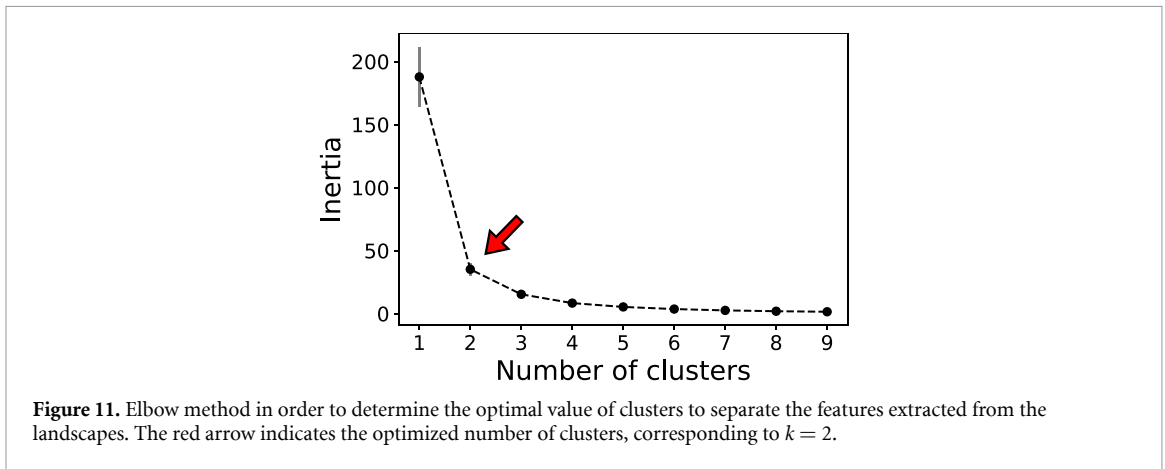


Figure 11. Elbow method in order to determine the optimal value of clusters to separate the features extracted from the landscapes. The red arrow indicates the optimized number of clusters, corresponding to $k = 2$.

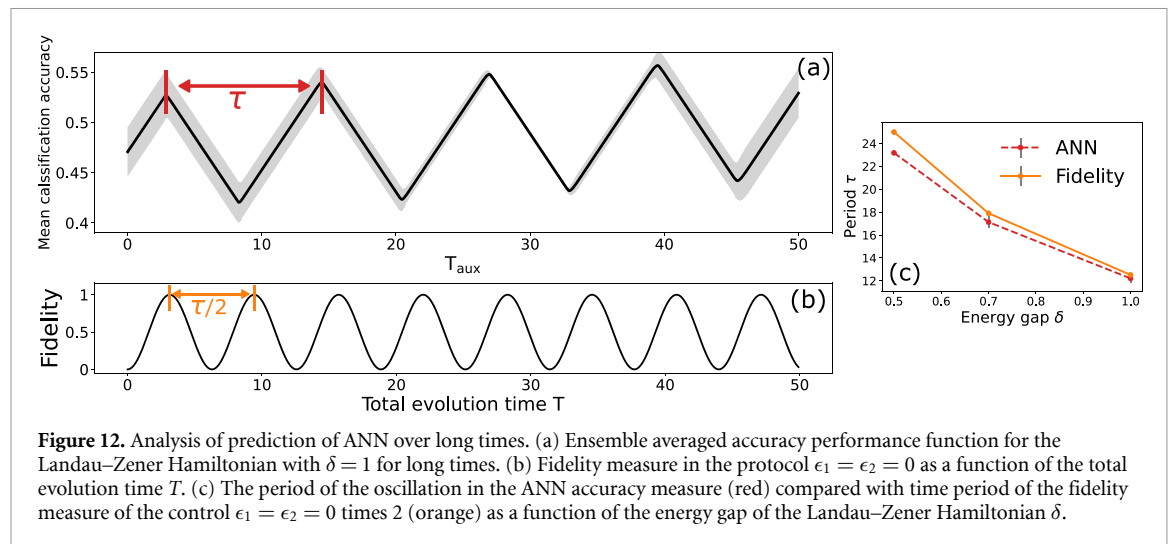
the end of the training the MSE loss is of order 10^{-2} , this value can be lowered to 10^{-4} by setting $\alpha = 0$. Next we move on to explain the k -means clustering method.

The K -means methods main function is to separate a dataset into k groups. In this work we use the algorithm given by Scikit-learn version 1.0.2 [43]. Here the unsupervised clustering is done by minimizing the inertia criterion.

The features obtained by using the ANN are clustered by the unsupervised k -means. Given that the algorithm requires the number of clusters to be specified beforehand and we did not want to force the ideal two clustering scheme, we implemented the elbow method. Figure 11 shows the average result (over all the 40 architectures) of the method implemented done over the training features extracted from the LZ Hamiltonian with an energy gap $\delta = 1$. As it can be seen 2 clusters is the optimized number of clusters, thus in all the networks we use the K -Means clustering into $k = 2$ groups.

Appendix C. Long times

In this section we give further details as to the minimum formed in the accuracy performance of figure 4 in the main text. For this, we train the ANN network with the LZ Hamiltonian (equation (4)), with the same



details as explained in the appendix B, and predict over a larger dataset that covers a total evolution time $T \in [0.01, 0.02, \dots, 49.9]$.

In the figure 12(a) we show the smoothed output performance accuracy achieved for a larger time scale. The accuracy presents an oscillatory behavior with period τ . In section 2, we noted that the optimal control of the LZ Hamiltonian in the MCT was located in the center, that is $\epsilon_1 = \epsilon_2 = 0$ where, for larger times, this center maximum splitted into two symmetrical structures oscillating in time. This can be observed by analyzing $F[\epsilon_1 = 0, \epsilon_2 = 0]$ (equation (3)) for different total evolution times, figure 12(b). Figure 12(c) compares the period obtained in the performance of the ANN (red) with the one obtained from the fidelity measure in the center of the control landscape (orange) times 2. As one can see, for the different energy gaps δ analyzed, they match pretty well. This displays that the network acquires more knowledge of landscapes than expected. Further analysis will be done in the future, given that it is outside the scope of this work.

ORCID iDs

Sofia Sevitz  <https://orcid.org/0000-0002-7135-4995>

Nicolás Mirkin  <https://orcid.org/0000-0002-7754-1505>

References

- [1] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge: Cambridge University Press)
- [2] Barends R *et al* 2016 Digitized adiabatic quantum computing with a superconducting circuit *Nature* **534** 222–6
- [3] Kelly J *et al* 2014 Optimal quantum control using randomized benchmarking *Phys. Rev. Lett.* **112** 240504
- [4] Koch C P, Boscain U, Calarco T, Dirr G, Filipp S, Glaser S J, Kosloff R, Montangero S, Schulte-Herbrüggen T, Sugny D and Wilhelm F K 2022 Quantum optimal control in quantum technologies. Strategic report on current status, visions and goals for research in Europe *EPJ Quantum Technol.* **9** 19
- [5] Peirce A P, Dahleh M A and Rabitz H 1988 Optimal control of quantum-mechanical systems: existence, numerical approximation, and applications *Phys. Rev. A* **37** 4950–64
- [6] D'Alessandro D 2007 *Introduction to Quantum Control and Dynamics* (Boca Raton, FL: Chapman and Hall/CRC)
- [7] Poggi P M 2019 Geometric quantum speed limits and short time accessibility to unitary operations *Phys. Rev. A* **99** 042116
- [8] Deffner S and Lutz E 2013 Quantum speed limit for non-Markovian dynamics *Phys. Rev. Lett.* **111** 010402
- [9] Taddei M M, Escher B M, Davidovich L and de Matos Filho R L 2013 Quantum speed limit for physical processes *Phys. Rev. Lett.* **110** 050402
- [10] Deffner S and Campbell S 2017 Quantum speed limits: from Heisenberg's uncertainty principle to optimal quantum control *J. Phys. A: Math. Theor.* **50** 453001
- [11] Wang X, Allegra M, Jacobs K, Lloyd S, Lupo C and Mohseni M 2015 Quantum brachistochrone curves as geodesics: obtaining accurate minimum-time protocols for the control of quantum systems *Phys. Rev. Lett.* **114** 170501
- [12] Yang J and del Campo A 2022 Minimum-time quantum control and the quantum brachistochrone equation (arXiv:2204.12792)
- [13] Moore T, Tibbetts K W, Brif C, Grace M D, Donovan A, Hocker D L, Ho T S, Wu R B and Rabitz H 2012 Exploring the tradeoff between fidelity and time optimal control of quantum unitary transformations *Phys. Rev. A* **86** 062309
- [14] Rabitz H A, Hsieh M M and Rosenthal C M 2004 Quantum optimally controlled transition landscapes *Science* **303** 1998–2001
- [15] Moore K W and Rabitz H 2012 Exploring constrained quantum control landscapes *J. Chem. Phys.* **137** 134113
- [16] Rach N, Müller M M, Calarco T and Montangero S 2015 Dressing the chopped-random-basis optimization: a bandwidth-limited access to the trap-free landscape *Phys. Rev. A* **92** 062343
- [17] Giannelli L, Sgroi P, Brown J, Paroanu G S, Paternostro M, Paladino E and Falsci G 2022 A tutorial on optimal control and reinforcement learning methods for quantum technologies *Phys. Lett. A* **434** 128054

- [18] Ding Y, Ban Y, Martín-Guerrero J D, Solano E, Casanova J and Chen X 2021 Breaking adiabatic quantum control with deep learning *Phys. Rev. A* **103** L040401
- [19] Zeng Y, Shen J, Hou S, Gebremariam T and Li C 2020 Quantum control based on machine learning in an open quantum system *Phys. Lett. A* **384** 126886
- [20] Dalgaard M, Motzoi F and Sherson J 2022 Predicting quantum dynamical cost landscapes with deep learning *Phys. Rev. A* **105** 012402
- [21] Niu M Y, Boixo S, Smelyanskiy V N and Neven H 2019 Universal quantum control through deep reinforcement learning *npj Quantum Inf.* **5** 33
- [22] Sauvage F and Mintert F 2022 Optimal control of families of quantum gates *Phys. Rev. Lett.* **129** 050507
- [23] Xiao T, Fan J and Zeng G 2022 Parameter estimation in quantum sensing based on deep reinforcement learning *npj Quantum Inf.* **8** 2
- [24] Huang H Y, Kueng R, Torlai G, Albert V V and Preskill J 2022 Provably efficient machine learning for quantum many-body problems *Science* **377** eabk3333
- [25] Xiao T, Huang J, Li H, Fan J and Zeng G 2022 Intelligent certification for quantum simulators via machine learning *npj Quantum Inf.* **8** 138
- [26] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev. X* **8** 031086
- [27] Huang T, Ban Y, Sherman E Y and Chen X 2022 Machine-learning assisted quantum control in a random environment *Phys. Rev. Appl.* **17** 024040
- [28] Ogure K 2022 Observing how deep neural networks understand physics through the energy spectrum of 1D quantum mechanics *Progr. Theor. Exp. Phys.* **2022** 113A01
- [29] Casert C, Viejra T, Nys J and Ryckebusch J 2019 Interpretable machine learning for inferring the phase boundaries in a nonequilibrium system *Phys. Rev. E* **99** 023304
- [30] Blücher S, Kades L, Pawłowski J M, Strodthoff N and Urban J M 2020 Towards novel insights in lattice field theory with explainable machine learning *Phys. Rev. D* **101** 094507
- [31] Grojean C, Paul A, Qian Z and Strümke I 2022 Lessons on interpretable machine learning from particle physics *Nat. Rev. Phys.* **4** 284–6
- [32] Larocca M, Poggi P M and Wisniacki D A 2018 Quantum control landscape for a two-level system near the quantum speed limit *J. Phys. A: Math. Theor.* **51** 385305
- [33] Murgida G, Wisniacki D and Tamborenea P 2009 Landau-Zener transitions in a semiconductor quantum dot *J. Mod. Opt.* **56** 799–804
- [34] Hegerfeldt G C 2014 High-speed driving of a two-level system *Phys. Rev. A* **90** 032110
- [35] Poggi P M, Lombardo F C and Wisniacki D A 2015 Time optimal control fields for quantum systems with multiple avoided crossings *Phys. Rev. A* **92** 053411
- [36] Vitanov N V, Rangelov A A, Shore B W and Bergmann K 2017 Stimulated raman adiabatic passage in physics, chemistry, and beyond *Rev. Mod. Phys.* **89** 015006
- [37] Mousavi S M, Zhu W, Ellsworth W and Beroza G 2019 Unsupervised clustering of seismic signals using deep convolutional autoencoders *IEEE Geosci. Remote Sens. Lett.* **16** 1693–7
- [38] Goodfellow I, Bengio Y and Courville A 2016 *Deep Learning* (Cambridge, MA: MIT Press)
- [39] van Nieuwenburg E P L, Liu Y H and Huber S D 2017 Learning phase transitions by confusion *Nat. Phys.* **13** 435–9
- [40] Zeiler M D and Fergus R 2013 Visualizing and understanding convolutional networks (arXiv:1311.2901)
- [41] Johansson J, Nation P and Nori F 2013 Qutip 2: a python framework for the dynamics of open quantum systems *Comput. Phys. Commun.* **184** 1234–40
- [42] Chollet F et al 2015 Keras (available at: <https://keras.io>)
- [43] Pedregosa F et al 2011 Scikit-learn: machine learning in python *J. Mach. Learn. Res.* **12** 2825–30 (available at: <https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>)