Engineering quantum current states with machine learning

Tobias Haug,1 Rainer Dumke,1,2,3 Leong-Chuan Kwek,1,3,4,5 Christian Miniatura,3,1,6,7,8,9 and Luigi Amico1,3,10,11,12

1Centre for Quantum Technologies, National University of Singapore, 3 Science Drive 2, Singapore 117543, Singapore
2Division of Physics and Applied Physics, Nanyang Technological University, 21 Nanyang Link, Singapore 637371, Singapore
3MajuLab, CNRS-UCA-SU-NUS-NTU International Joint Research Unit, Singapore
4School of Electrical and Electronic Engineering, 50 Nanyang Ave., Singapore 637553
5National Institute of Education, Nanyang Technological University, 1 Nanyang Walk, Singapore 637616, Singapore
6Department of Physics, National University of Singapore, 2 Science Drive 3, Singapore 117542, Singapore
7School of Physical and Mathematical Sciences, Nanyang Technological University, 637371 Singapore, Singapore
8Yale-NUS College, 16 College Avenue West, Singapore 138527, Singapore
9Université Côte d’Azur, CNRS, Institut de Physique de Nice, 1361 route des Lucioles, 06560 Valbonne, France
10Dipartimento di Fisica e Astronomia, Via S. Sofia 64, 95127 Catania, Italy
11CNR-MATIS-IMM & INFN-Sezione di Catania, Via S. Sofia 64, 95127 Catania, Italy
12LANEF ‘Chaire d’excellence’, Université Grenoble-Alpes & CNRS, F-38000 Grenoble, France

(Dated: November 22, 2019)

The design, accurate preparation and manipulation of quantum states are essential operational tasks at the heart of quantum technologies. Nowadays, physical parameters of quantum devices and networks can be controlled with unprecedented accuracy and flexibility. However, the generation of well-controlled current states is still a nagging bottleneck, especially when different circuit elements are integrated together. In this work, we show how machine learning can effectively address this challenge and outperform the current existing methods. To this end, we exploit deep reinforcement learning to prepare prescribed quantum current states within a short time scale and with a high fidelity. To highlight our method, we show how to engineer bosonic persistent currents in ring circuits as they are key ingredients in different quantum technology devices. With our approach, quantum current states characterized by a single winding number or entangled currents with two winding numbers can be prepared superseding the existing protocols. In addition, we generated quantum states entangling a larger set of different winding numbers. Our deep reinforcement learning scheme provides solutions for known challenges in quantum technology and opens new avenues for the control of quantum devices.

With the advent of quantum technologies, new forms of quantum circuit have emerged. The exact architecture and circuit performance depend on the specific physical implementation and the type of ‘quantum fluid’ operating in the quantum network. Examples range from electronic and superconducting circuits [1] based on charged matter-wave on nanolithography to photonic circuits employing photons in fiberoptics [2], or atomtronic circuits involving neutral matterwaves of cold atoms on optically generated structures with micrometric resolution [3, 4]. Most of the available quantum technologies exploit an enhanced control on the physical properties of the quantum fluid operating in the circuit like particle-particle interaction or the statistical properties of the particles (fermions/bosons). In addition, it is possible with the latest achievements in the field to adjust and reconfigure dynamically the spatial features of the circuit at the local scale [5–9]. Finally, quantum circuits with increasingly complex architecture and hybrid systems in which different technologies are interfaced are at a mature stage of technology readiness level [10–13]. In most of such situations, it is necessary to achieve and control dynamical current states with high fidelity.

In this work, we employ machine learning to approach the problem of generating current states in quantum circuits. Neural networks have been very successful in optimizing problems with complex parameter landscapes. Machine learning with deep reinforcement learning is a powerful tool to engineer dynamics in quantum systems [14–18], that can outperform other methods [19]. Deep learning relies on representing a highly complex function (e.g. the quality of the driving protocol) with a neural network, and optimize it using observable data (e.g. measurement outcomes). This general method is well suited to be applied directly to optimize experiments with different kind of circuits. Here, we demonstrate this approach to prepare quantum current states describing the flow of coherent matter-wave in closed circuits: persistent currents [20]. Persistent currents are a direct manifestation of the phenomenon of quantum coherence and are therefore of central interest in fundamental aspects of many-body physics like superfluidity, superconductivity and mesoscopic physics [21]. At the same time, such concepts play a vital role for important technological applications such as in Superconducting Quantum Interference Devices (SQUID’s) [22] where they act as quantum sensors for the accurate measurement of magnetic fields, or of rotations as in the case of cold atom counterpart of SQUID devices [23, 24]. Persistent currents have been the object of intense studies in different contexts of quantum technologies like cold atoms [25–27], superconducting circuits [28], optical cavities [29], opto-mechanical cavities [30] and tailored reservoirs [31]. While charged or neutral matter-wave persistent currents have been obtained in simplified situations, protocols achieving such states in more general settings are still missing. We consider quantum systems that can be mapped onto ring-shaped circuits comprising L local units that we call sites. These systems encompass most of the general features and challenging aspects for the generation of
Quantum currents in closed circuits. We propose to create current states by locally driving the circuit in a suitable way. If the system can be driven by changing few control parameters, state engineering can be carried out through optimal control theory [32]. For increasingly large number of parameters, however, the circuit driving cannot be handled with standard means. In this work, we employ deep reinforcement learning to implement current state engineering by driving each lattice site of the ring circuit independently. With our approach, we demonstrate that persistent currents with specific winding number can be imparted, on a timescale that is much shorter than other known protocols. Additionally, we can create entangled current states of up to three winding numbers, for which there is no known protocol. These protocols are generated by training the neural network with experimental observables only.

Local drive of Bose-Hubbard ring circuits. As sketched in Fig.1b, our model system is a ring circuit comprising $L$ sites, a natural architecture to consider to generate persistent currents. $N_p \geq 2$ interacting scalar bosonic particles are filling the ring lattice and can hop between nearest-neighbor sites $j$ and $j+1$ with an amplitude $J$. The ring lattice can be locally driven by externally varying in time each on-site potential $P_j(t)$. Such a many-body quantum system is described by the Bose-Hubbard Hamiltonian

$$\mathcal{H}_{BH} = \sum_{j=1}^{L} \left[ -J (\hat{a}^\dagger_j \hat{a}_{j+1} + \hat{a}^\dagger_{j+1} \hat{a}_j) + P_j(t) \hat{n}_j + \frac{U}{2} \hat{n}_j (\hat{n}_j - 1) \right]. \quad (1)$$

Here $\hat{a}_j$, $\hat{a}^\dagger_j$ and $\hat{n}_j = \hat{a}^\dagger_j \hat{a}_j$ are the usual bosonic creation, annihilation and number operators on site $j$. They satisfy the bosonic commutation relation $[\hat{a}_j, \hat{a}^\dagger_k] = \delta_{jk}$. Periodic boundary conditions are fixed by imposing $\hat{a}^\dagger_{L+1} = \hat{a}^\dagger_1$. Finally $U$ is the on-site inter-particle interaction.

In the limit of a large average number of particles per site $N_s = N_p/L \gg 1$, the Bose-Hubbard Hamiltonian effectively reduces to the so-called quantum phase model (QPM)

$$\mathcal{H}_{QP} = \sum_{j=1}^{L} \left[ -2J_E \cos(\phi_j - \phi_{j+1}) + P_j(t) \hat{\phi}_j + \frac{U}{2} \hat{\phi}_j^2 \right]. \quad (2)$$

where $J_E = JN_s$, $\hat{\phi}_j = \hat{\phi}_j - N_s$ is the on-site particle number fluctuations and $\phi_j$ the phase operators [33, 34]. The operators satisfy the commutation relations $[\hat{\phi}_j, \hat{\phi}_k] = i\hbar \delta_{jk}$. Hamiltonians (1) and (2) describe a wide class of different physical quantum systems ranging from 1d arrays of Josephson junctions and qubits [35] to atomtronic circuits, hence their theoretical importance in quantum technologies.

Quantum current states. In a coherent quantum circuit the current states in the ring are quantized as the phase along a closed path can only change by integer multiples of $2\pi$. We describe these winding numbers $k$ by defining the single-particle winding state $|k\rangle = \hat{b}_k^\dagger |\text{vac}\rangle$, where $|\text{vac}\rangle$ denotes the vacuum state and $\hat{b}_k^\dagger = \frac{1}{\sqrt{2\pi}} \sum_n e^{i2\pi kn/L} \hat{a}^\dagger_n$ the quasi-momentum creation operator (see Methods). In a ring system, the quasi-momentum corresponds to the winding number. We choose $\Omega = \{k_1, k_2, \ldots, k_{N_C}\}$ as a set of $N_C$ winding numbers that we want to prepare in a superposition state. In the simplest instance, the many-body state is the tensor product state of $N_p$ uncorrelated particles in the same quantum current state:

$$|PS\rangle = \left( \sum_{k=\Omega}^{N_s} \frac{1}{\sqrt{N_C}} |k\rangle \right)^{\otimes N_p}. \quad (3)$$

However, because of the interaction, quantum correlations can develop between the different particles and the many-body state is generically entangled. These states are beyond any classical description and are notoriously difficult to generate. In the following, we consider entangled quantum current states of the form

$$|ES\rangle = \frac{1}{\sqrt{N_C}} \sum_{k=\Omega}^{N_s} |k\rangle \otimes \cdots \otimes |k\rangle$$

consisting of $N_C$ winding numbers. Important examples of entangled current states that we will specifically consider in the present work are the NOON-state...
(\(N_C = 2\)) and the W-state (\(N_C = 3\)). To distinguish W from PS, we refer to the following certification measure for \(N_C \geq 2\)

\[
W_\text{PS} = \frac{N_C^2}{N_p N_c^2} \prod_{k \in \Omega} \langle \Psi | \hat{n}_k \hat{b}_k | \Psi \rangle, \tag{3}
\]

where \(\hat{n}_k = \hat{b}_k^\dagger \hat{b}_k\). It is easy to check that for entangled states \(W_\text{ES} = 1\) irrespective of \(N_p\) and \(N_c\), while for product states \(W_\text{PS} = \left(\frac{N_p + N_c - 1}{N_p N_c}\right)^{N_c} \leq \frac{9}{16}\), the upper bound being obtained for \(N_p = N_c = 2\). Indeed, Eq. (3) behaves similarly to the fidelity \(F = |\langle \Psi | \text{ES} \rangle|^2\) and gives similar results for the state design (see Supplemental Material and [36]). In contrast to the fidelity \(F\) though, we note that \(W_\text{PS}\) is related to the density operator and therefore is an observable. In a cold atom setting, for example, \(W_\text{PS}\) can be accessed by measuring the number of particles in a specific momentum mode, which can be achieved by time-of-flight measurements [37] (see Methods).

These current states can be generated with the following algorithm: The driving protocol of total time \(T\) is discretized into \(N_T\) timesteps of equal length \(\Delta t = T/N_T\). The potential at each lattice site can be chosen freely within a range \(|P_j| < P_{\text{max}}\). Within each timestep, the system evolves under constant parameters and we assume that the potential parameters change instantaneously between two timesteps. We learn the driving protocol via a deep Q-network [38]. We adapt our method with the actor-critic method, and revised it to learn over a continuous action space. Our method is using Proximal Policy optimization [39] and the implementation is based on Tensorflow [40]. A sketch of the neural network including a general explanation is shown in Fig. 1, while the details to the algorithm are presented in the Methods section. Here, we note that our scheme relies on a model-free optimization algorithm: The learning algorithm does not make any assumptions about the specific system or knows about quantum mechanics. As such, the algorithm can be supplied with experimental data (in our case the potential parameters and measurement observables) to optimize the experiment directly.

**Stirring a localized barrier potential.** Transforming the non-rotating ground state to a specific rotating state requires...
perturbing the state in a manner that explicitly breaks time-reversal symmetry. This has been implemented in cold atoms settings [24, 25, 41]. Here, we adapt this stirring protocol to a ring lattice: We move a potential barrier initially localized at one site of height $P_{B}$ to the next site with a fixed rate $v$, in a time $t_{\text{jump}} = 1/v$. We target generating current states with one winding number, starting from the ground state of the system. We find that a high fidelity is reached by driving the ring for a time of about $T = 20/J$ for different particle numbers $N_{p}$ (rightmost curves in Fig.2a). Consistently with previous results, we find that the first rotational state is created best by stirring with about a speed of half the desired atom velocity (see Fig.2d, [42]). The actual implementation of the barrier protocol implies a trade-off between achieving high fidelity $F$ and short protocol time $T$: With increasing $P_{B}$ the maximal achievable fidelity decreases, however it is reached in a shorter time $T$ (see Fig.2c). To shorten the protocol run time without sacrificing fidelity, more complex protocols are required.

**Local control of the circuit.** To achieve a higher control over the dynamics of the quantum system, we define the Full Control Protocol (FCP) in which each lattice site is driven individually. In Fig.2a, we compare stirring and FCP to generate current state with winding number $k = 1$. The FCP reaches a better fidelity than the barrier driving protocol requiring only half the time or less. An example protocol that optimizes the state generation is shown in Fig.1c. Next, we study a ring with many cold atoms. We employ the QPM Eq.(2) to describe the limit of many atoms and intermediate interaction. We show our results in the black curve in Fig.2a and in Fig.2b. The fidelity is initially non-zero as the ground state of the QPM has a broad winding number distribution and finite overlap with the target state. To create a state with high fidelity, a sufficient protocol time $T$ is required. After that, further increase in $T$ does not enhance fidelity anymore.

To go beyond quantum currents states composed of a single winding number, we employ FCP to engineer entangled superposition of winding numbers. This way, we demonstrate the preparation of entangled superposition states of different currents for up to three winding numbers for which no driving protocols are known so far to our knowledge (see Fig.3a). The fidelity improves over protocol time $T$ and reaches eventually a plateau. The number of time steps is important to generate current states. With increasing number of steps, higher fidelity is reached. For interacting systems, we achieve best results for $N_{T} \geq 4$ for (Fig.3b). Sufficient interaction $U$ between particles is necessary to generate entangled currents. We find that there is an optimal interaction $U \approx J$, for which we achieve best fidelity (Fig.3c). All states we considered show similar scal-
ing with interaction energy, however the certification measure is lower for states of higher winding number. The certification measure increases with longer protocol time $T$ until it reaches a plateau (see Fig. 3d). Increasing $T$ beyond does not improve the state generation. For single-particle superpositions, the threshold time to reach the plateau is about $T \approx 3/J$.

For interacting system with a few particles, we find that longer times $T$ are required.

Discussion. In this work we demonstrated how to use machine learning for the efficient generation of currents in closed quantum circuits. The essential features of this problem are captured by a ring consisting of lumped elements which we schematize as lattice sites. We introduced the Full Control Protocol (FCP), where all sites of the lattice are driven individually. The FCP scheme defines a challenging problem with a large numbers of parameters to be optimized. We solve this problem with deep reinforcement learning. A neuronal network, running on a classical computer, generates the sequence of potentials to be applied on the quantum system. Then, we measure the fidelity or certification measure and optimize the neuronal network with it. Running this procedure repeatedly will improve the machine-generated protocol. Our algorithm can optimize large parameter spaces (we demonstrate up to 108 free parameters). The algorithm is agnostic of the underlying physical system as a kind of hybrid quantum-classical optimizer [43–45]. It can be directly applied to systems like cold atoms, superconducting arrays or photonic waveguides, relying only on readily available experimental observables. As a demonstration, we applied our algorithm to fundamental bosonic models (Bose-Hubbard and quantum phase model). Our protocol can generate current states consisting of a single winding number more than twice as fast and with higher fidelity compared to the standard stirring scheme used so far by the cold atom community - Fig. 2a. We demonstrated entangled superpositions of currents in Fig. 3. For the certification of these states, we devised the measure $W_F$ that is benchmarked by comparing it with the fidelity with the entangled state (see Supplemental Material). Here, we use $W_F$ instead of the fidelity $F$ itself, since $W_F$ is a convenient observable that can be measured in experiments and allows our scheme to be directly applied to optimize actual experimental setups in quantum technology. Besides current states with a single winding number and NOON states, we realize entangled current states of W-type involving three winding number. In rotating cold atoms, the latter states entangle three angular momentum states. To our knowledge, no known protocols for such states have been devised so far. We find that the complexity of the driving protocol (protocol time and number of time steps) depends on the number of particles: Non-interacting systems can generate currents much faster and with simpler protocols compared to the interacting many-body system. Both small and large number of particles regimes (through Bose-Hubbard and quantum phase dynamics respectively) were explored.

Our results are of direct relevance in different contexts of quantum technology like cold-atoms and superconducting circuits. Our approach can be extended to other quantum many-body systems and readily applied to prepare current states in more complex circuit geometries and hybrid quantum networks. In particular, the preparation of current states and entanglement is very relevant for quantum-enhanced sensing [46]. Finally, it would be of major interest to compare our deep reinforcement learning based method with other optimization methods like GRAPE or CRAB [19].

Acknowledgments

We thank D. Basko, M. Holzmann, D. Hutchinson and A. Minguzzi for discussions. The Grenoble LANEF framework (ANR-10-LABX-51-01) is acknowledged for its support with mutualized infrastructure. The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg).

Data Availability

The data that support the findings of this study are available from the corresponding author upon reasonable request.

[1] R. Barends, L. Lamata, J. Kelly, L. García-Álvarez, A. Fowler, A. Megrant, E. Jeffrey, T. White, D. Sank, J. Mutus, et al., Nat. Comm. 6, 7654 (2015).
[2] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O’Brien, Science 320, 646 (2008).
[3] R. Dumke, Z. Lu, J. Close, N. Robins, A. Weis, M. Mukherjee, G. Birkl, C. Hufnagel, L. Amico, M. G. Boshier, et al., J. Opt. 18, 093001 (2016).
[4] L. Amico, G. Birkl, M. Boshier, and L.-C. Kwek, New J. Phys. 19, 020201 (2017).
[5] G. Gauthier, I. Lenton, N. M. Parry, M. Baker, M. Davis, H. Rubinsztein-Dunlop, and T. Neely, Optica 3, 1136 (2016).
[6] J. Liang, J. Rudolph N. Kohn, M. F. Becker, and D. J. Heinzen, Appl. Opt. 48, 1955 (2009).
[7] C. Muldoon, L. Brandt, J. Dong, D. Stuart, E. Brainis, M. Himsworth, and A. Kuhn, New Journal of Physics 14, 073051 (2012).
[8] K. Henderson, C. Ryu, C. MacCormick, and M. G. Boshier, New J. Phys. 11, 043030 (2009).
[9] H. Rubinsztein-Dunlop, A. Forbes, M. V. Berry, M. R. Dennis, D. L. Andrews, M. Mansuripur, C. Denz, C. Alpmann, P. Banzer, T. Bauer, et al., Journal of Optics 19, 013001 (2016).
[10] G. Kurizki, P. Bertet, Y. Kubo, K. Mølmer, D. Petrosyan, P. Rabl, and J. Schmiedmayer, Proceedings of the National Academy of Sciences 112, 3866 (2015).
[11] A. Aspuru-Guzik and P. Walther, Nat. Phys. 8, 285 (2012).
[12] A. A. Houck, H. E. Türeci, and J. Koch, Nat. Phys. 8, 292 (2012).
[13] I. Bloch, J. Dalibard, and S. Nascimbene, Nat. Phys. 8, 267 (2012).
Winding number states

To define our current states, we transform the ring Hamiltonian Eq. 1 of $L$ sites with $U = 0$ and $P_f = 0$ by Fourier transforming the operators into the quasi-momentum basis

$$\mathcal{H}_{\text{FT}} = \sum_{k=0}^{L-1} -2J \cos \left( \frac{2\pi k}{L} \right) \hat{a}_k^\dagger \hat{a}_k$$

where $\hat{b}_k = \hat{b}_{-k}^\dagger$, with $\hat{b}_j^\dagger = \frac{1}{\sqrt{L}} \sum_n e^{2\pi in/(n+L)}$. As the wavefunction around the ring is continuous, the wavefunction must be the same after going once around the ring. Thus, we demand $\exp(\frac{i2\pi k}{L} n) = \exp(\frac{i2\pi k}{L}(n+L))$, which is only fulfilled if $k$ is an integer number, which describes how often the phase of the wavefunction winds by $2\pi$ around the ring. The state with winding number $k$ for a single particle is defined as $\hat{b}_k^\dagger |\text{vac}\rangle = |k\rangle$, where $\text{vac}$ denotes the vacuum state. Many-body states are generated as tensor products of particles with winding number $k$.

Quantum phase model fidelity

To define the current state for the quantum phase model (QPM), we use the QPM with an applied artificial magnetic
field $\Phi_M$

$$H_{QP}(\Phi_M) = \sum_{j=1}^{L} \left[ -2J N_j \cos(\phi_j - \phi_{j+1} - \Phi_M) + P_j(t) \hat{Q}_j + \frac{U}{2} \hat{Q}^2_j \right]. \quad (5)$$

To define the target state that carries a current, we refer to the ground state with a winding number distribution that is centered around the winding number $k$. The ground state $|\Psi(\Phi_M)\rangle$ winding number distribution is centered around the winding number $k$. We define the fidelity as $\langle \Psi|\Psi(\Phi_M)\rangle^2$.

**Measuring entangled current**

To characterize the entangled current states, we define the certification measure Eq.3, which is a product of expectation values of observables. Experimentally, one is required to measure the square of the particle-number operator $\langle \hat{n}_k^2 \rangle$ of the winding number mode $k$. For cold atoms condensates, this measure can be determined from time-of-flight measurements, where the prepared state is expanded in free space [37].

For superconducting circuits, the expectation value of the square of the particle number of a specific winding number $k$ can be derived from the expectation value of fourth order correlators between different qubits. We find

$$\hat{n}_k^2 = \frac{1}{L^2} \sum_{n,m,r,s} e^{2\pi i (n+m-r-s)/L} \hat{a}^\dagger_n \hat{a}^\dagger_m \hat{a}_r \hat{a}_s,$$

where the correlators can be derived by Fourier transforming the annihilation and creation operators of the operator.

**Deep reinforcement learning**

Here, we describe our machine learning algorithm in more detail. A detailed figure on the neuronal network structure and pseudo-code is found in the Supplemental Material. We learn the driving protocol via a deep Q-learning network [38], utilizing the actor-critic method acting on a continuous action space. Our method is using Proximal Policy optimization[39] and the implementation is based on Tensorflow [40]. The quantum system is controlled by an agent, that depending on the state $s_t$ of the system acts with an action $a_t$ using the probabilistic policy $\pi(a_t|s_t)$. The idea of Q-learning is to find the Q-function $Q(s_t, a_t)$ that estimates the future reward that is paid out at the end the full protocol with this policy. The goal is to learn a policy that can realize long-term rewards over smaller short-term gains. The optimal Q-function is determined by the Bellman equation

$$Q(s_t, a_t, \pi) = \mathbb{E} \left[ r_t + \gamma Q(s_{t+1}, a_{t+1}, \pi) \right] = \mathbb{E} \left[ r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \ldots \right]$$

where $\mathbb{E}[]$ indicates sampling over many instances. $\gamma \leq 1$ is a discount factor that weighs future rewards against immediate rewards. The input to the neural network are the Hamilton parameters at previous timesteps and it outputs the parameters for the policy $\pi(a_t|s_t, \mu, \sigma)$, where the actions are sampled from a normal distribution with mean value $\mu$ and width $\sigma$. $\mu$ is determined by the neural network and $\sigma$ is optimized as a global variable and decreases during the optimization procedure. We constrain the possible output values for the potential by mapping values outside of the constraint to the maximally allowed value. Proximal policy optimization is based on the actor-critic method. The idea is to have two neural networks: A policy network and a value network. The policy network (actor) decides on the next action by determining the parameters of the policy. The value based network (critic) evaluates the taken action on how well it solves the task and estimates the future expected reward. It is used as an input to train the policy network. The two networks are trained at the same time using Adam[47]. Better performance can be achieved if the Q-function is split into two parts[48]:

$$Q(s_t, a_t) = A(s_t, a_t) + V(s_t),$$

where $A(s_t, a_t)$ is the advantage function and $V(s_t)$ the value function. $V(s_t)$ gives the expected future reward averaged over the possible actions according to the policy. This is the output of the critic network. $A(s_t, a_t)$ gives the improvement in reward for action $a_t$ compared to the mean of all choices. We estimate the Q-function from the value function with $Q(s_t, a_t) = r_t + \gamma V(s_{t+1})$, where $r_t$ is the reward given out under action $a_t$ and $V(s_{t+1})$ is the value function for the next timestep. We then minimize the square of the difference of the value function of the network and the predicted reward in the next timestep $L_V(\theta) = \mathbb{E}_i \left[ (V_i(s_t) - y_i)^2 \right]$, where $\theta$ are the network parameters, $y_i = r_t + V_{t+1}$ is the calculated reward of the next timestep. The advantage function $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ tells us how good a certain action $a_t$ is compared to other possible actions. Using above estimation of the Q-function, the advantage function can be approximated. The advantage function is the input to train the policy network (the actor). Following the idea of proximal policy optimization [39], the goal is to maximize

$$L_p(\theta) = \mathbb{E}_i \left[ \frac{\pi_{\theta}(s_t, a_t)}{\pi_{\text{old}}(s_t, a_t)} A(s_t, a_t) \right], \quad (6)$$

where $\theta$ are the network parameters and $\theta_{\text{old}}$ are the network parameters of a previous instance. Maximizing $L_p(\theta)$ for the network parameters $\theta$ over many sampled instances guides the distribution $\pi_{\theta}(s_t, a_t)$ such that it returns actions $a_t$ with maximal advantage. However, the ratio

$$b_i(\theta) = \frac{\pi_{\theta}(s_t, a_t)}{\pi_{\text{old}}(s_t, a_t)}$$

can acquire excessive large values, causing too large changes in the policy in every training step and making convergence difficult. It was proposed to use a clipped ratio [39]

$$L_p(\theta) = \mathbb{E}_i \left[ \min \{ b_i(\theta) A(s_t, a_t), \text{clip}(r_i(\theta), 1 - \epsilon, 1 + \epsilon) A(s_t, a_t) \} \right],$$

where $\text{clip}(x, a, b)$ is a function that returns $a$ if $x \leq a$ and $b$ if $x \geq b$. The clipping prevents excessive large ratios from dominating the update. The clipped ratio can be rewritten as

$$L_p(\theta) = \mathbb{E}_i \left[ \min \{ b_i(\theta) A(s_t, a_t), A(s_t, a_t) \} \right] - \mathbb{E}_i \left[ \min \{ b_i(\theta) A(s_t, a_t), 0 \} \right]$$

This form is useful because it separates the optimistic and pessimistic updates. The optimistic update is the first term, which encourages the agent to explore new actions, while the pessimistic update is the second term, which discourages the agent from taking actions that are likely to be worse than the current best action.
such that the update at each step stays in reasonable bounds. We use $\epsilon = 0.1$. We optimize the neural network over many epochs $N_E$. For our results, we show the best protocol that was achieved during the optimization process. We update the network by randomly sampling $N_{\text{train}}$ past iterations from a memory (replay Buffer $B$) that stores the last $N_{\text{memory}}$ epochs. To reduce premature convergence, we add the entropy of the normal distributions of the policy to the loss function $L_N(\theta) = E_p[\frac{1}{2}\sigma \ln(2\pi\sigma)]$. This contribution slow down optimization to avoid convergence to a local minimum. The final loss function to optimize is $L(\theta) = L_P - cV + cSL$, where $cV$ and $cS$ are hyperparameters. We find $cV = 0.02$ and $cS = 0.5$ as good choices. A sketch of our neural network is shown in Fig.1. The protocol solves the Schrödinger equation for a total time $T$ in Fig.1. The protocol solves the Schrödinger equation for $N$ parameters at different sites $m$ of in total $L$ sites. For one epoch, the system runs the network $N_T$ times. Input are the potentials used at previous timesteps $t_n$, and it returns the parameters to be used for the $n+1$ timestep. The input vector has length $(L+1)N_T$; it lists the parameters and the corresponding times $t_k$ used up to current timestep $t_k = t_1 ... t_n$. The network propagates through two hidden layers of fully connected neurons of size $N_H$ with ReLu activation functions. The output layer has size $L+1$ and uses a linear activation function. For the value function (critic), the output of the last hidden layer is collected to a single node, that represents the value function $V(s_k)$. For the policy (actor), $L$ outputs determine the mean values of the normal distribution that generates the potential at the next timestep $t_{n+1}$ of the protocol. The neuronal network is trained with the loss function after calculating the full time evolution to time $T$ and measuring all the rewards.

For the actual implementation, we choose the following parameters: learning rate with Adam $\alpha = 0.0002$, $N_H = 200$ neurons in the hidden layer, training over $N_E = 120000$ epochs, training with a randomly sampled batch size 500, a replay buffer $B$ of $N_{\text{train}} = 500N_T$ previous results.

A: Fidelity with entangled state

In the main text, we discuss entangled current states using the certification measure $W_E$. This measure is closely related to the fidelity with the entangled state $|ES\rangle$. In Fig.4, we show results on the fidelity for the same parameters as in the main text in Fig.3. We find that the fidelity behaves similar to the certification measure. The certification measure is always zero for the initial state (e.g. seen for datapoints with $T = 0$), however the fidelity can be non-zero. This is because the initial state has in some cases a finite overlap with the target entangled state. In contrast, the certification measure is constructed that this initial overlap does not affect it and it is zero.

B: Supporting numerical results

Here, we present further data to support our findings. To solve the quantum phase model numerically, we restrict the fluctuations around the mean particle number to $\Delta \bar{Q}_m$. In Fig.5, we show the fidelity for $\Delta \bar{Q}_m = 4$ and $L = 5$.

In the main text, we demonstrate the generation of a state with winding number $k = 1$ using the barrier protocol. The same protocol can also generate superposition states of $k = 0$ and $k = 1$. To generate the entangled state, the barrier is rotated at the same speed as for the $k = 1$ case with shorter time $T$. In Fig.6, we study the dynamics of creating entangled superposition states and compare the FCP against driving a barrier.

C: Deep reinforcement learning

The details on our deep reinforcement learning algorithm are given in the method section of the main text. Here, we show an explanatory figure (Fig. 7) to illustrate the learning algorithm and the structure of the neuronal network. A pseudo-code of the algorithm is given in Fig.8.

D: Optimization runtime

The machine learning algorithm starts with a randomly initialized neuronal network, that generates the driving sequence. By running repeatedly, the network generates better driving sequences. Finally, we plot the best driving sequence found during the training episode. Here, we show results on the training procedure. The fidelity achieved increases during the training over many episodes. In Fig.9, we show the fidelities during the training procedure for the FCP protocols that were used for Fig.2a of the main text.

E: Statistics

Our goal is to optimize a high-parameter space driving protocol. In general, the optimization landscape is complex, with many local minima. We run the machine learning algorithm several times, and look at the convergence of the certification measure. As the algorithm is non-deterministic and not guaranteed to converge to the global minimum, each run can yield different end results. In Fig.10, we show the minimal and maximal certification measure achieved for 20 runs to create entangled states. For reaching $\Omega = [0, 1]$ we see only a small variation between minimal and maximal achieved certification measure –Fig.10a. Thus, in one run of the algorithm we can be sure that a very good solution is found. However, we see a significant spread in certification measure results for higher particle number and more complex entangled states, e.g. $\Omega = [-1, 1]$ and $N_p = 3$ particles –Fig.10b. Thus, for this parameter set to find the best result, several runs have to
FIG. 4. Comparison fidelity with entangled state $F = |\langle \Psi | ES \rangle|^2$ (for index 1) and certification measure $W_\Psi$ (for index 2) for $L = 12$ sites for various parameters. Same parameters as in Fig.3 of main text. We optimize for equal weight entangled states of $N_C$ winding number $k$ of type $|ES\rangle = \frac{1}{\sqrt{N_C}} \sum_{k} |k\rangle^\otimes N_p$ of a set of winding number $\Omega = \{k_1, k_2, \ldots k_{N_C}\}$. 
a) Varying time $T$ to generate different entangled states for $N_p = 2$ particles, $N_T = 6$ timesteps and $U = J$
b) For varying timesteps $N_T$ to reach state $\Omega = \{0, 1\}$ for $N_p = 2$ particles.

c) Interaction $U$ dependence for different types of states for $N_T = 6$ timesteps and protocol time $T = 9/J$.
d) Total protocol time $T$ for $N_T = 4$ timesteps to generate entangled superposition state of winding number $\Omega = \{0, 1\}$. 
performed. This implies that the complexity and difficulty of the optimization problem to generate entangled states is highly dependent on the parameters of the problem. We took care to check that the variance of the solutions is within reasonable bounds for our results.

**F: Experimental considerations**

For a cold atom implementation, the driving of the ring lattice can create excitations. Within the Bose-Hubbard approximation, only the first Bloch band is considered. It is assumed that higher Bloch bands are far-detuned in energy and thus do not contribute. In most experiments, the energy gap between the Bloch bands within harmonic approximation in the lattice well is given by $E_{\text{lattice}} = 2 \sqrt{V_0 E_R}$, where $V_0$ is the potential energy of a sinusoidal confinement and $E_R$ is the recoil energy [49]. For typical $V_0 = 10E_R$, we find $E_{\text{lattice}} = 6.3E_R$. The nearest-neighbor coupling $J$ can be approximated as $J/E_R = \frac{4}{\sqrt{\pi}} \left( \frac{V_0}{E_R} \right)^{3/4} \exp \left( -2 \sqrt{\frac{V_0}{E_R}} \right) \approx 0.02$ [37].

Thus, the energy separation between the first and second Bloch band is $\Delta E = E_{\text{lattice}}/J \approx 315$. From first order perturbation theory, we know that the overlap with higher-order states scales as $P/\Delta E$, where $P_{\text{max}}$ is the strength of the perturbation. The perturbation of the potential that we apply is on the order of $P_{\text{max}} = J$, which is much smaller than the energy gap. Thus, we can safely ignore excitations to higher Bloch bands.

For the quantum phase model, the nearest-neighbor coupling strength scales as $J = JN_p/L$. To justify the one Bloch band approximation, $J$ has to be much less than the energy gap between the Bloch bands. Thus, $J = JN_p/L \ll E_{\text{lattice}}$.

For cold atoms, the control parameter to change the local potential $P_j$ is the potential $V_0$. In our protocol, we change $P_j(t)$ and thus $V_0$ in time on the order of the nearest-neighbor coupling $J$. However, $J$ is actually function of $V_0$ and thus may change due to the driving. However, as $V_0 \gg J$, changing $V_0$ on the order of $J$ has a negligible effect on $J$.

The potential has to be controlled by experimental control. We assume a step-wise control of the potential, with sharp changes in the potential. We apply steps in our experiments that change the potential on the order of $J$, at timescales of $J$. We detail how this be realized in experiment. In the case of cold atoms, the potential is generated by laser pulses. Light-shaping techniques can modify the potential with a frequency of about 20kHz [5]. The relevant timescales of the experiment are on the order of $J$, which are far smaller. For superconducting circuits, the potential is controlled by microwave pulses that modifies the circuit potential. The circuit can be modulated on the order of 35MHz, while the nearest-neighbor couplings are far slower with $J \approx 4$MHz [28]. Thus, we conclude that the driving parameters can feasible be modulated on the time scales we consider.
FIG. 6. Generation of (entangled) superposition states of zero and one rotational quantum \( \Omega = \{0, 1\} \) in a ring lattice with \( L = 12 \) sites. Evolution of fidelity \( F \) (see Eq. 3) during driving. We compare two different protocols: A barrier localized at a single site moving at constant speed (right curves in a) or fully controlling the potential (FCP) of every lattice individually (left curves of a). c) Minimal time \( T_{\text{min}} \) required to create rotational states above a threshold fidelity \( F_{\text{min}} = 0.95 \) for \( N_p = 1 \), else \( F_{\text{min}} = 0.8 \) for different values of barrier amplitude. We find best rotation speed of barrier is at \( v \approx 0.5 J \). c) best protocol for rotating barrier. Curve shows barrier position over time d) best protocol for full control over lattice potentials for a protocol of two timesteps \( n_t \). Barrier and full control protocols shown calculated for \( N_p = 1 \) particles.

FIG. 7. Neural network to optimize protocols to generate quantum states. Deep learning relies on representing a highly complex function (e.g. the quality of the driving protocol) with a neural network, and optimize it using observable data (e.g. measurement outcomes). The quantum system is a lattice ring with \( L \) sites where particles can hop between neighboring sites with strength \( J \). Each site \( m \) has a local potential \( P_m(t) \) that can be modulated in discrete timesteps \( t_n \). The neural network controls the evolution of the quantum system by adjusting \( P_m(t) \) and optimizes the parameters over many runs. The neural network performs step-wise evolution of the quantum in \( N_T \) discrete time steps \( t_n \) over total runtime \( T \). It uses the chosen potentials of previous time steps as an input (state \( s(t_n) \)), and returns the potentials to be chosen at the next step (action \( a(t_n) \)) by sampling them from a Gaussian distribution. The training is performed by using a measure for the quantum state (reward \( r(t_n) \)).
Randomly initialize critic \( V(s;\theta) \) and actor \( \mu(s;\theta) \) with weights \( \theta \)

Initialize replay buffer \( B \)

for episode=1, \( N_E \) do
  Input initial state \( s_1 \)
  for \( t = 1, N_T \) do
    Sample action \( a_t = \mu(s_t;\theta) \) from probability distribution
    Execute action \( a_t \), receive reward \( r_t \) and next state \( s_{t+1} \)
    Sample random batch of \( N_{train} \) transitions \( (s_t, a_t, r_t, s_{t+1}) \) from \( B \)
    Set \( y_t = r_t + \gamma V(s_{t+1}) \)
    Update critic by minimizing loss
    \[
    L = \frac{1}{M} \sum_i (y_i - V(s_i;\theta))^2
    \]
    Calculate advantage function
    \[
    A(s_t, a_t) = Q(s_t, a_t) - V(s_t) = r_t + \gamma V(s_{t+1}) - V(s_t)
    \]
    Calculate probability ratio \( r_p = \frac{P(a_t|\mu(s_t;\theta))}{P(a_t|\mu(s_t;\theta^{old})} \) of current policy \( \mu(s;\theta) \) and previous policy \( \mu(s;\theta^{old}) \)
    function clip \( (r_p, c) \) clips \( r_p \) between \( 1 - c < r_p < 1 + c \)
    Update actor policy with clipped loss
    \[
    L = \min \left[ r_p A(s_t, a_t), A(s_t, a_t) \text{clip}(r_p, c) \right]
    \]
  end for
  Store \( (s_t, a_t, r_t, s_{t+1}) \) \( \forall t \) in replay buffer \( B \)
end for

FIG. 8. Pseudo-code for our proximal policy optimization algorithm to generate quantum states.

![Graphs](image1.png)

FIG. 9. Optimization of the FCP protocol by the neuronal network over the number of epochs (number of protocol runs). We show exemplary data that generated the protocol shown in Fig.2a of the main text. The dots indicate fidelity achieved during a particular run, while the red line is the moving average over the results. a) \( N_p = 1 \) b) \( N_p = 2 \) c) \( N_p = 3 \) d) quantum phase model.
FIG. 10. Statistics (minimum, maximum and average certification measure $W_\Psi$) over 20 repeated runs of the algorithm for different parameter sets a) different particle numbers $N_p$ for $N_T = 4$, $T = 8/J$, $U = J$, $\Omega = \{0, 1\}$ b) different protocol steps $N_T$ for a total protocol length of $T = 10/J$, $\Omega = \{-1, 1\}$, $L = 12$ sites for $N_p = 3$ particles and $U = J$. Driving with local potential $|P| < P_{\text{max}} = J$. 