

ARTICLE OPEN



Identifying optimal cycles in quantum thermal machines with reinforcement-learning

Paolo A. Erdman¹ and Frank Noé^{1,2,3,4}

The optimal control of open quantum systems is a challenging task but has a key role in improving existing quantum information processing technologies. We introduce a general framework based on reinforcement learning to discover optimal thermodynamic cycles that maximize the power of out-of-equilibrium quantum heat engines and refrigerators. We apply our method, based on the soft actor-critic algorithm, to three systems: a benchmark two-level system heat engine, where we find the optimal known cycle; an experimentally realistic refrigerator based on a superconducting qubit that generates coherence, where we find a non-intuitive control sequence that outperforms previous cycles proposed in literature; a heat engine based on a quantum harmonic oscillator, where we find a cycle with an elaborate structure that outperforms the optimized Otto cycle. We then evaluate the corresponding efficiency at maximum power.

npj Quantum Information (2022)8:1 | <https://doi.org/10.1038/s41534-021-00512-0>

INTRODUCTION

Thermal machines convert between thermal and mechanical energy in a controlled manner. Examples include heat engines such as steam and Otto engines, which extract useful work from a temperature difference, and refrigerators, that extract heat from a cold bath. Quantum thermal machines (QTM) perform thermodynamic cycles via nanoscale systems that can be as small as single particles or two-level quantum systems (qubits). Quantum heat engines and refrigerators could find applications in heat management at the nanoscale¹, or for on-chip active cooling^{2,3}.

Quantum thermodynamics is a rapidly growing research area that aims at the understanding, design, and optimization of QTMs⁴. An open fundamental question is whether quantum effects can boost the performance of QTMs^{2,4,5}. On the other hand, understanding how to optimally control the non-equilibrium dynamics of open quantum systems is a complicated task, which can improve existing quantum information processing technologies.

Nowadays it is possible to construct devices that behave as quantum systems with few degrees of freedom in platforms such as trapped ions^{6,7}, electron spins associated with nitrogen-vacancy centers⁸, circuit quantum electrodynamics⁹, and quantum dots¹⁰, and to control their state through time-dependent controls, such as electro-magnetic pulses or gate voltages. The heat flow across these systems has been measured^{11–14}, and recent experimental realizations of QTMs have been reported^{15–22}.

While the laws of thermodynamics pose universal constraints on the efficiency of thermal machines, regardless of their classical or quantum nature, they do not pose any restriction on the dynamics of the system, thus on the speed at which it operates. Therefore, it is crucial to study the power to discover the potential benefits of using QTMs. However, optimizing the power is a challenging task: having to operate in finite-time, the state can be driven far from equilibrium, requiring us to model the full dynamics of the quantum system. Furthermore, strategies are needed to identify optimally controlled cycles.

Power maximization of QTMs^{4,23} is generally carried out either in specific regimes or assuming a priori a specific shape of the control cycle. Within the slow-driving^{24–30} and fast-driving regime, general strategies have been recently derived^{31–33}. Beyond these regimes, common strategies consider specific cycle structures^{34–39}, such as the celebrated Otto cycle^{40–55}, and optimize specific aspects of the cycle. Shortcuts to adiabaticity^{56–63}, and variational optimization strategies have also been employed^{64–66}. The impact of quantum effects on the performance of QTMs is not straightforward. Several studies have found quantum advantages^{50,51,55,67–69}, while coherence-induced power losses were proven in linear response for small driving amplitudes⁷⁰ and in specific models^{42,48,54,64,71}.

In general, there is no guarantee that typical regimes and specific cycles considered in literature are optimal for power maximization. Overcoming this limitation may allow us to unlock quantum advantages in power extraction. This calls for the development of powerful search strategies to tackle power-maximization without relying on specific control sequences or assumptions.

In this manuscript, we propose a Reinforcement Learning (RL)⁷² based approach to optimize the performance of QTMs. Specifically, we use a generalization of soft actor-critic (SAC) methods^{73,74} for combined discrete and continuous actions, introduced in the context of robotics and video games^{75,76}, to discover thermodynamic cycles that deliver maximum power. RL has received a great deal of attention for its success at mastering complicated tasks beyond the human level such as playing video games^{77,78}, the board-game Go⁷⁹, and for robotic applications⁸⁰. RL has been recently used for accurate quantum state preparation^{81–84}, outperforming previous state-of-the-art methods^{85,86}, to minimize entropy production in closed quantum systems⁸⁷, for fault-tolerant quantum computation⁸⁸, and machine learning methods have been used for quantum thermometry⁸⁹.

Our RL-based scheme for power maximization of QTMs is generic in that it makes no assumptions on the shape of the control cycle. Rather, it starts from scratch, allowing the RL agent

¹Freie Universität Berlin, Department of Mathematics and Computer Science, Arnimallee 6, 14195 Berlin, Germany. ²Freie Universität Berlin, Department of Physics, Arnimallee 6, 14195 Berlin, Germany. ³Rice University, Department of Chemistry, Houston, TX 77005, USA. ⁴Microsoft Research, Cambridge, UK. ✉email: p.erdman@fu-berlin.de; frank.noe@fu-berlin.de

to arbitrarily couple or decouple the quantum system from any bath, and to arbitrarily manipulate the control parameter. We apply our approach to three paradigmatic systems that have been well studied in literature: (i) a benchmark heat engine based on a two-level system, where our approach automatically finds the known maximum power cycle⁹⁰. (ii) an experimentally realistic refrigerator based on a superconducting qubit coupled to resonant circuits⁴⁸ which generates coherence during its cycle. Our RL approach discovers a new and non-intuitive cycle that outperforms previous proposals^{48,54,62}. (iii) a heat engine based on a harmonic oscillator⁴², where we find a cycle with an elaborate structure that shares qualitative similarities with the Otto cycle, but which performs better thanks to additional features. We complement the study of these systems by analyzing the corresponding efficiency at maximum power.

The complexity and structure of these cycles demonstrate both the ability of the RL agent to choose actions based on a long-term return and the complicated non-analytic nature of optimal cycles. Our results also show that the celebrated Otto cycle is not in general optimal for power maximization. Furthermore, we show that the detrimental effect of coherence on the performance of QTMs⁷¹, specifically observed both in the superconducting qubit and harmonic oscillator cases, can be mitigated thanks to carefully crafted cycles.

RESULTS

Quantum thermal machines

We describe a QTM by a quantum system (QS), acting as a working medium, that can exchange heat with a hot (H) or cold (C) thermal bath characterized by inverse temperatures $\beta_H < \beta_C$ (Fig. 1). We can control the evolution of the QS and exchange work with it through a set of time-dependent control parameters $\mathbf{u}(t)$. In classical thermal machines, the working medium could be a gas in a cylinder, and $\mathbf{u}(t)$ could be the time-dependent position of the piston which influences the state of the gas and allows us to exchange energy. In a QTM, the working medium is a quantum system whose Hamiltonian can be parametrized by the control variables $\mathbf{u}(t)$ ⁹¹.

Here we study finite-time thermodynamics of QTMs within the Markovian regime using the commonly employed master equation^{92–95}. This approach describes the time-evolution of the reduced density matrix of the QS, $\hat{\rho}(t)$, under the assumption of weak system-bath interaction. Setting $\hbar = 1$, the master equation reads

$$\frac{\partial}{\partial t} \hat{\rho}(t) = -i[\hat{H}[\mathbf{u}(t)], \hat{\rho}(t)] + \sum_a \mathcal{D}_{\mathbf{u}(t), d(t)}^{(a)}[\hat{\rho}(t)], \quad (1)$$

where $\hat{H}[\mathbf{u}(t)]$ is the Hamiltonian of the QS which depends explicitly on time via the control parameters $\mathbf{u}(t)$, $[\cdot, \cdot]$ denotes the

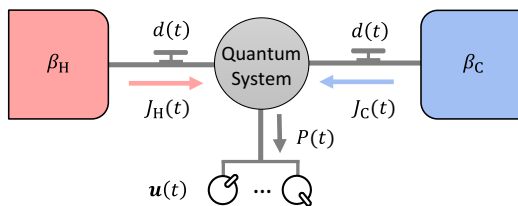


Fig. 1 Schematic representation of a quantum thermal machine. A hot (cold) bath at inverse temperature β_H (β_C), represented by the red (blue) box, can be coupled to the quantum system QS, gray circle, enabling a heat flux $J_H(t)$ ($J_C(t)$). The quantum system is controlled by an external agent through a set of control parameters $\mathbf{u}(t)$, which allow power exchange $P(t)$, and through a discrete control $d(t) = \{\text{Hot, Cold, None}\}$ which determines which bath (if any) is coupled to the QS.

commutator, and $\mathcal{D}_{\mathbf{u}(t), d(t)}^{(a)}[\cdot]$ describes the effect of the coupling between the QS and bath $a = H, C$. $d(t) = \{\text{Hot, Cold, None}\}$ is an additional discrete control parameter which allows us to choose which bath (if any) is coupled to the QS. We compute the extracted power $P(t)$ and the instantaneous heat flux $J_a(t)$ flowing out of bath a in the standard way²³ which guarantees the validity of the first law of thermodynamics $\partial U(t)/\partial t = -P(t) + \sum_a J_a(t)$, the internal energy being defined as $U = \text{Tr}[\hat{\rho}(t)\hat{H}[\mathbf{u}(t)]]$ (see “Methods” for details). The two main thermal machines we consider are the heat engine and the refrigerator. A heat engine is used to extract work, while a refrigerator is used to extract heat from the cold bath. Therefore, we define

$$P_{[E]}(t) \equiv \sum_{a=H,C} J_a(t), \quad P_{[R]}(t) \equiv J_C(t), \quad (2)$$

respectively as the instantaneous power of a heat engine E (since the total heat extracted coincides with the work if the internal energy difference is zero), and as the instantaneous cooling power of a refrigerator R.

Our goal is to determine the optimal driving, i.e. to determine the functions $\mathbf{u}(t)$ and $d(t)$ that maximize the average power in the long run. We thus define the following exponentially weighted average of the power

$$\langle P_{[v]} \rangle = \bar{\gamma} \int_0^\infty e^{-\bar{\gamma}t} P_{[v]}(t) dt, \quad (3)$$

where $\bar{\gamma}$ determines the timescale over which we average. While we do not enforce any periodic structure on the controls $\mathbf{u}(t)$ and $d(t)$, we expect the RL agent to automatically discover the optimality of periodically driving the QTM and the corresponding driving period. The intuition is the following: in the short term, we can maximize the power by taking advantage of the state preparation of the system, for example by extracting all the free energy from the system. However, the amount of work that can be extracted this way is bounded, while the work that can be extracted through cycles scales with the number of performed cycles, i.e. with time. Therefore in the long run, i.e. for small enough $\bar{\gamma}$, we expect the maximization of Eq. (3) to naturally discover thermodynamic cycles and to prevent the exploitation of transient effects. We confirm this hypothesis in all QTMs studied below.

Reinforcement learning for quantum thermal machines

We formulate the power optimization problem as a discounted, continuing RL task. As we will demonstrate, this approach is able to learn far-from-equilibrium finite-time thermodynamic cycles with high performance.

RL is a general framework that tackles optimization problems formulated in the following way. As schematically shown in Fig. 2, a computer agent (blue box) must learn to master some tasks by repeated interactions with some environment (gray box). Discretizing time in time steps Δt , we denote with $s_i \in \mathcal{S}$ the state of the environment at time $t = i\Delta t$, where \mathcal{S} is the state space. The agent must choose an action $a_i \in \mathcal{A}$ to perform on the environment based on its current *policy* (lower orange arrow). \mathcal{A} is the action space, and the policy $\pi(a_i|s_i)$ is a function that describes the probability distribution of choosing action a_i , given that the environment is in state s_i . The environment then evolves its state according to the chosen action and provides feedback back to the agent by returning the updated state s_{i+1} and a scalar quantity r_{i+1} known as the *reward* (upper orange arrow). This procedure is reiterated for a large number of time steps.

At every time step t_i , the aim of the agent is to use the feedback it receives from the environment to learn an optimal policy that maximizes, in expectation, the *return*, i.e. the total future reward it

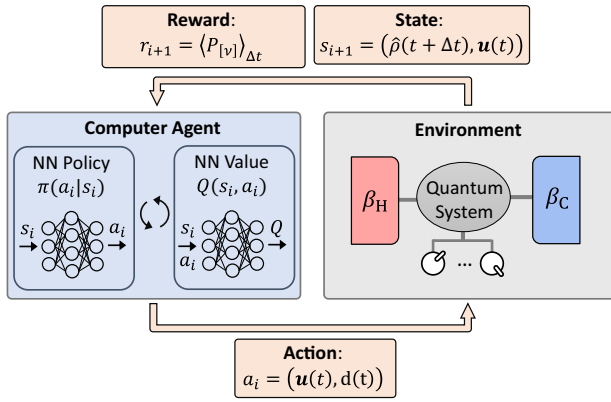


Fig. 2 Schematic representation of the learning process. A computer agent (blue box) chooses an action a_i at time step i based on the current state s_i of the environment (gray box) through the policy function $\pi(a_i|s_i)$. The action, which encodes the control $(\mathbf{u}(t), d(t))$, is passed to the environment (lower arrow) which evolves the quantum state s_i of the machine based on the action a_i , and computes the average power during the time step as a reward. The new state s_{i+1} and reward r_{i+1} are returned to the agent (upper arrow), which uses this information to improve $\pi(a|s)$ using the soft actor-critic algorithm, which learns also the value function $Q(s, a)$. Both $\pi(a|s)$ and $Q(s, a)$ are parameterized using fully connected neural networks (NNs). This process is reiterated until the convergence of the policy.

receives from the environment, defined as

$$r_{i+1} + \gamma r_{i+2} + \gamma^2 r_{i+3} + \dots = \sum_{k=0}^{\infty} \gamma^k r_{i+1+k}, \quad (4)$$

where $\gamma \in [0, 1)$ is the *discount factor* which determines how much we are interested in future rewards, as opposed to immediate rewards.

We now turn to the power maximization problem. Discretizing time in steps Δt , we search for protocols $\mathbf{u}(t)$ and $d(t)$ that are constant during each time step. As shown in Fig. 2, we choose as action space $\mathcal{A} = \{(\mathbf{u}, d) \mid \mathbf{u} \in \mathcal{U}, d \in \{\text{Hot, Cold, None}\}\}$, where \mathcal{U} is the continuous set of accessible controls, which can account for any experimental limitation, and d is the discrete action, motivated by typical thermodynamic cycles, which determines which bath (if any) is coupled to the QS. We choose the physical quantum states of the QS and the last chosen action as state space, i.e., $\mathcal{S} = \{(\hat{\rho}, \mathbf{u}) \mid \hat{\rho} \in \mathcal{D}, \mathbf{u} \in \mathcal{U}\}$, where $\mathcal{D} = \{\hat{\rho} \mid \hat{\rho} \geq 0, \text{Tr}[\hat{\rho}] = 1\}$ is the space of density matrices. Crucially, we choose as a reward

$$r_{i+1} = \langle P_{[v]} \rangle_{\Delta t} \equiv \frac{1}{\Delta t} \int_t^{t+\Delta t} P_{[v]}(\tau) d\tau, \quad (5)$$

which is the average power of the machine during the time interval $[t, t + \Delta t]$. Plugging Eq. (5) into Eq. (4), we see that the aim of the agent is to maximize the average power $\langle P_{[v]} \rangle$ introduced in Eq. (3), where $\bar{\gamma} = -\ln \gamma / \Delta t$ (see Methods for details). In the RL notation, γ sets the timescale for the power averaging, with $\gamma \rightarrow 1$ corresponding to long-term averaging.

Our agent has no prior knowledge of quantum dynamics, nor of thermodynamic cycles: the evolution of the state from $s_i = \hat{\rho}(t)$ to $s_{i+1} = \hat{\rho}(t + \Delta t)$ and the computation of the rewards r_i is performed by the environment. We learn the optimal policy employing the SAC method, which relies on learning also a value function $Q(s, a)$, generalized to a combination of discrete and continuous actions^{73–76}. In this approach, the policy function $\pi(a|s)$ plays the role of an “actor” that chooses the actions to perform, while the value function $Q(s, a)$ plays the role of a “critic” that

judges the choices made by the actor, thus providing feedback to improve the actor’s behavior. Both $\pi(a|s)$ and $Q(s, a)$ are parameterized using fully connected neural networks with two hidden layers, and they are determined by minimizing the loss functions in Eqs. (22) and (29) in Methods using the ADAM optimization algorithm⁹⁶. The gradient of the loss functions is computed off-policy, over a batch of past experience which is recorded and stored in a replay buffer, using backpropagation (see Methods for details).

Case studies

In this section, we prove the validity of our RL-based approach by applying it to three different systems, namely a heat engine based on a two-level system, a refrigerator based on a superconducting qubit, and a heat engine based on a quantum harmonic oscillator. While the results presented below were obtained performing a single training, in Methods we show that our RL approach reliably converges to solutions with nearly the same performance across multiple trainings.

We first benchmark our method on a two-level system for which the optimal control cycle is known:

$$\hat{H}[u(t)] = \frac{E_0 u(t)}{2} \hat{\sigma}_z, \quad (6)$$

where $u(t)$, which determines the energy gap of the two-level system, is our single control parameter, E_0 is a fixed energy scale and $\hat{\sigma}_z$ is a Pauli matrix. We consider the qubit to be coupled to fermionic baths with a flat density of states (for example, a single-level quantum dot tunnel-coupled to metallic leads), with thermalization timescales fixed by the rates Γ_a (see Methods for details).

Our results are shown in Fig. 3. Figure 3a shows the running average of the power $\langle P_{[E]} \rangle_{\gamma}(i) \equiv (1 - \gamma) \sum_{k=0}^i \gamma^k r_{i-k}$, i.e. an exponentially weighted average of the past rewards with weight γ , as a function of the time steps. Figure 3b shows the actions chosen by the agent, as a function of the steps, at three different moments during training highlighted by the black dots in Fig. 3a. The position of the segments corresponds to the chosen value of u , while the color represents the discrete action d (see legend). The optimal cycle learned at the end of the training is shown in Fig. 3c. Initially, the agent has no knowledge of the system, and its actions appear random (Fig. 3b, left), producing negative power $\langle P_{[E]} \rangle_{\gamma}$ (Fig. 3a). As expected, by performing random actions the agent is dissipating work into the heat baths, rather than extracting work. With increasing time, the agent gains experience and learns how to control the heat engine: $\langle P_{[E]} \rangle_{\gamma}$ increases, and structure appears in the chosen actions (Fig. 3b, center, and right). Eventually, the policy converges and $\langle P_{[E]} \rangle_{\gamma}$ saturates to a finite positive value.

The optimal cycle for this model was derived in ref.⁹⁰, and it corresponds to the exact structure discovered by the agent, i.e. a square wave alternating between the hot and the cold bath as fast as possible without spending any time disconnected from the baths. The black dashed line in Fig. 3a shows the corresponding power. Notably, although the frequency of the learned cycle is limited by the choice of Δt , and although the values of $u(t)$ found by the agent are slightly different respect to the ones predicted by ref.⁹⁰ (the difference is ≈ 0.05), the power it generates nearly coincides with the upper bound. This occurs because there is a manifold of near-optimal solutions.

We conclude the analysis of the two-level system heat engine by computing the *efficiency at maximum power* (EMP), i.e. the thermodynamic efficiency of the heat engine, defined as the ratio between the extracted work and the input heat, while the engine is operated at maximum power. We compare the EMP both to the Carnot efficiency $\eta_c = 1 - \beta_H/\beta_C$ and to the Curzon-Ahlborn efficiency $\eta_{ca} = 1 - \sqrt{\beta_H/\beta_C}$. Despite not being a fundamental

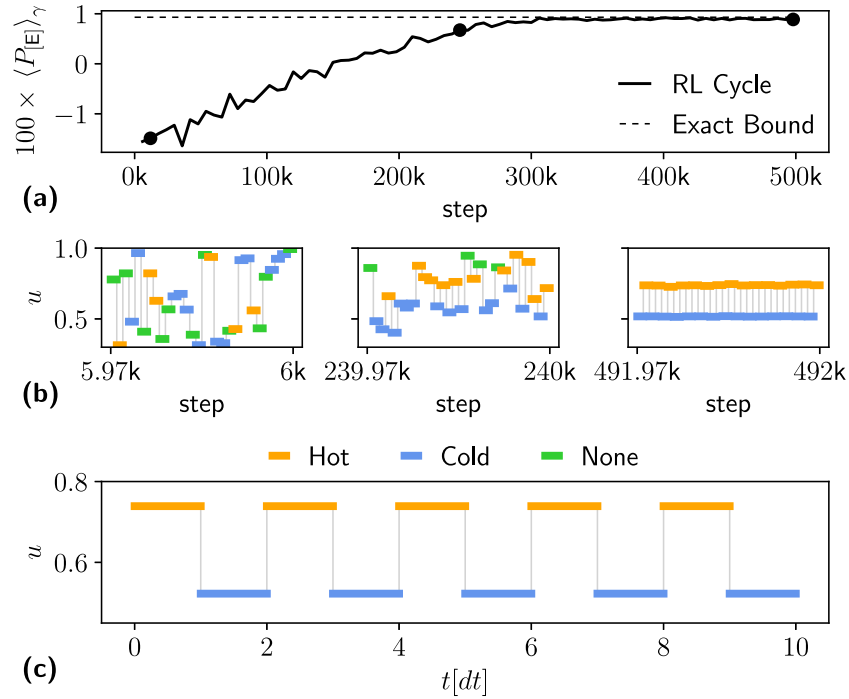


Fig. 3 Results of training the agent on the two-level heat engine model. **a** Running average of the extracted power $\langle P_{[E]} \rangle_{\gamma}$ as a function of the step during the whole training. The dashed line represents the theoretical upper bound derived in ref.⁹⁰ **b** Actions chosen by the agent, represented by the value of u , as a function of step, zoomed around the three black circles in panel (a). The color represents action d , i.e. the bath coupled to the QS. **c** Final cycle found by the agent as a function of time. The parameters used for training are: $\Gamma_H = \Gamma_C = 1$, $\beta_H = 1$, $\beta_C = 2$, $U = [0.3, 1]$, $E_0 = 2.5$, $\Delta t = 0.5$, and $\gamma = 0.995$.

upper bound, the latter has received considerable attention in the literature for its simplicity, for being an upper bound to the EMP in various specific models^{24,28,97,98}, and it has been derived by general arguments from linear irreversible thermodynamics⁹⁹. Interestingly, we find that the optimal cycle shown in Fig. 3c delivers a large EMP corresponding to 100% of η_{ca} , equivalent to 59% of η_c .

We now consider a refrigerator based on an experimentally realistic system, i.e. a superconducting qubit coupled to two resonant circuits which behave as heat baths⁴⁸. As opposed to the previous setup, here coherence between the instantaneous eigenstates is generated while driving the system, since $[\hat{H}(u_1), \hat{H}(u_2)] \neq 0$ for $u_1 \neq u_2$. This quantum effect was found to deter the power of this specific setup^{48,54,62} and of arbitrary systems in linear response and for small driving amplitudes⁷⁰.

As shown in refs.^{48,54,62}, the system Hamiltonian is given by

$$\hat{H}[u(t)] = -E_0[\Delta\hat{\sigma}_x + u(t)\hat{\sigma}_z], \quad (7)$$

where E_0 is a fixed energy scale, Δ characterizes the minimum gap of the system, and $u(t)$ is our control parameter. In this setup, the coupling to the bath is fixed, and cannot be controlled. However, the qubit is resonantly coupled to the baths at different energies. The u -dependent coupling strength to the C (H) bath is described by the function $\gamma_u^{(C)}$ ($\gamma_u^{(H)}$) that, as in ref.⁶², is peaked at $u = 0$ ($u = 1/2$) with a resonance width determined by the quality factor Q_C (Q_H) (see Methods for details).

Panels (a), (b), and (c) of Fig. 4 report the results in the same style as in Fig. 3, with the exception that all actions are black since there is no discrete choice to make, while Fig. 4d shows the coupling strength $\gamma_u^{(C)}$ (blue curve) and $\gamma_u^{(H)}$ (red curve) as a function of u (on the y-axis). The parameters were chosen as in Fig. 7 of ref.⁶². As previously, the agent begins with random actions in the first steps, and the corresponding running average cooling power $\langle P_{[R]} \rangle_{\gamma}$ is slightly negative, indicating that we are dissipating heat into the cold bath. As the agent gains experience, $\langle P_{[R]} \rangle_{\gamma}$

increases until it saturates to a final positive value obtained with the cycle shown in Fig. 4c (thick lines).

Interestingly, this setup was partially optimized in ref.⁶². In their analysis, the authors fix a smoothed trapezoidal cycle $u(t)$ (Fig. 4c, dashed line) which was shown to outperform a sine and a trapezoidal cycle⁴⁸. They find that the cooling power is positive for large cycle periods T , and tends to zero as $T \rightarrow \infty$. However, the cooling power becomes negative as $T \rightarrow 0$ because the detrimental effect of the generation of coherence increases with the speed of the cycle. As a consequence, there is an intermediate optimal value of T . They find that $\langle P_{[R]} \rangle \approx 2.3 \times 10^{-4}$ at this optimal choice (dashed line in Fig. 4a, c). Notably, our RL agent discovers a protocol with $\langle P_{[R]} \rangle \approx 10.8 \times 10^{-4}$ using the same system parameters. This improvement is due to the non-intuitive additional step visible in Fig. 4c. Indeed, both the trapezoidal and the agent's cycle spend time in resonance with the baths at $u = 0$ and $u = 0.5$. However, the agent identifies also a third point, around $u \approx 0.13$, where it spends $\approx 1/4$ of the total cycle time and where the system is essentially decoupled from the bath, thus undergoing unitary evolution (as can be seen by the small values of $\gamma_u^{(d)}$ corresponding to $u \approx 0.13$ in Fig. 4d). This additional feature allows us to roughly extract the same amount of heat per cycle, but 5 times faster. Interestingly, we verified that the trapezoidal cycle running at the same speed as the cycle found by the RL agent would yield negative power. As argued in^{48,54,62}, this power loss is attributed to the generation of coherence during the cycle, so we can interpret the power enhancement achieved by our cycle as mitigation of such detrimental effects. To confirm this, we rigorously quantify the generation of coherence in both cycles by computing the time average of the relative entropy of coherence¹⁰⁰. Indeed, we find that the trapezoidal cycle operated at the same speed as the cycle found by the RL agent generates almost twice as much coherence (see Methods for details).

We conclude the study of the qubit-based refrigerator by evaluating the *coefficient of performance* (COP) at maximum

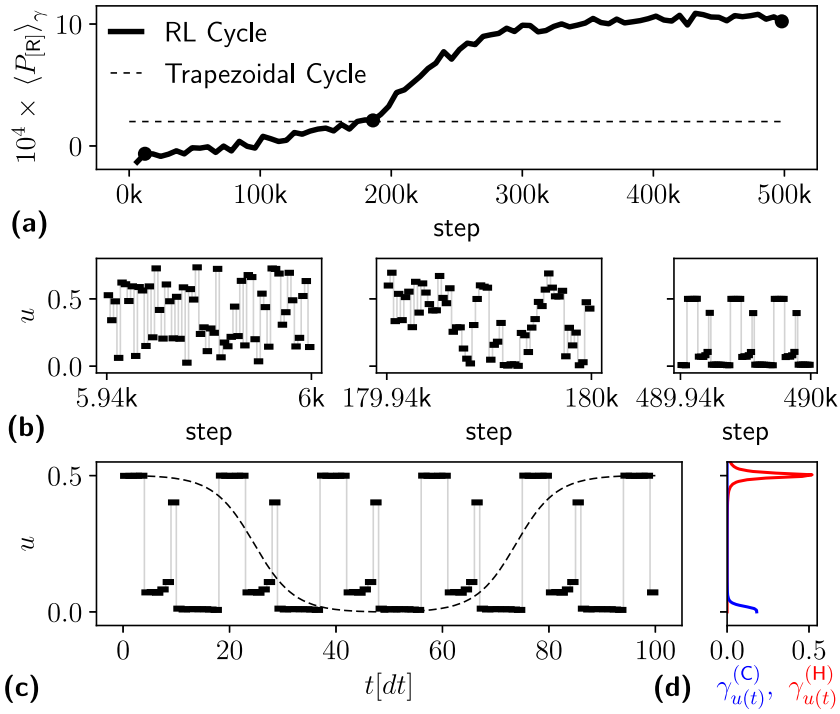


Fig. 4 Results of training the agent on the superconducting qubit refrigerator model. **a** Running average of the cooling power $\langle P_{\text{R}} \rangle_{\gamma}$ as a function of the step during the whole training. The dashed line represents the maximum cooling power found in ref. ⁶² by optimizing a smoothed trapezoidal cycle. **b** Actions chosen by the agent, represented by the value of u , as a function of step, zoomed around the three black circles in panel (a). **c** Final deterministic cycle found by the agent (thick discontinuous line) and smoothed trapezoidal cycle (thin dashed line) whose power is given by the dashed line in panel (a), as a function of time. **d** coupling strength $\gamma_{u(t)}^{(C)}$ (blue curve) and $\gamma_{u(t)}^{(H)}$ (red curve) as a function of u (on the y -axis). The parameters used for training are $g_{\text{H}} = g_{\text{C}} = 1$, $\beta_{\text{H}} = 10/3$, $\beta_{\text{C}} = 2\beta_{\text{H}}$, $Q_{\text{H}} = Q_{\text{C}} = 30$, $E_0 = 1$, $\Delta = 0.12$, $\omega_{\text{H}} = 1.03$, $\omega_{\text{C}} = 0.24$, $\mathcal{U} = [0, 0.75]$, $\Delta t = 0.98$, and $\gamma = 0.995$.

power, i.e. the ratio between the heat extracted from the cold bath and the input work, while the system is operated at maximum cooling power. We find that our cycle shown in Fig. 4c as a thick black line delivers a COP at maximum power that is 6% of $C_{\text{C}} = \beta_{\text{C}}^{-1}/(\beta_{\text{H}}^{-1} - \beta_{\text{C}}^{-1})$, which is Carnot's upper bound to the COP. While this may appear as a rather low value, we notice that the COP at a maximum power of a two-level system coupled to Fermionic or Bosonic baths is zero⁹⁰, and that for on-chip cooling applications, the aim is typically to maximize the cooling power regardless of the efficiency of such a process.

At last, we consider a heat engine based on a collection of non-interacting particles confined in a harmonic potential⁴². The Hamiltonian is given by

$$\hat{H}[u(t)] = \frac{1}{2m}\hat{p}^2 + \frac{1}{2}m(u(t)w_0)^2\hat{q}^2, \quad (8)$$

where m is the mass of the system, w_0 is a fixed frequency, and \hat{p} and \hat{q} are the momentum and position operators. The control parameter $u(t)$ allows us to change the frequency of the oscillator. As in the qubit heat engine case, we let the agent choose which bath (if any) to couple to the oscillator. Since $[\hat{H}(u_1), \hat{H}(u_2)] \neq 0$ for $u_1 \neq u_2$, also this system exhibits a power loss at finite driving speed^{42,71}. The coupling to the baths, characterized by the thermalization rates Γ_{α} , is modeled as in ref. ⁴² (see Methods for details).

The solid line in Fig. 5a, b shows $\langle P_{\text{E}} \rangle_{\gamma}$, as a function of the step, for the same system parameters (chosen as in the upper panel of Fig. 6 of ref. ⁴²), but setting respectively $\mathcal{U} = [0.5, 1]$ and $\mathcal{U} = [0.35, 1.5]$. In both cases, the power is negative for small steps, while it converges to a positive value as the agent gains experience. The corresponding final cycles learned by the agent are shown in Fig. 5c, d as thick lines. These cycles display a quite

elaborate structure which demonstrates the ability of the agent to perform planning for a long-time reward. Indeed, while the system is in contact with the cold bath, which happens roughly for 20-time steps, energy is flowing into the bath, thus producing negative rewards. Nonetheless, the agent discovers that this step, required by the second law of thermodynamics, is necessary for power extraction in the long run.

We now compare the cycle discovered by the agent to the well-known Otto cycle often considered in literature. The authors of ref. ⁴² study the power of this system by optimizing an Otto cycle, i.e. the cycle shown in Fig. 5c, d as a dashed line. The authors then fix the value of the control $u_{\text{H}} = 1$ ($u_{\text{C}} = 0.5$) while in contact with bath H (C), and maximize the power by tuning the duration of each of the four segments composing the cycle. The resulting optimized Otto cycle is the one shown as a dashed line in Fig. 5c, d, with corresponding power shown in Fig. 5a, b as a dashed line. As we can see, even constraining the choice of $u(t) \in [u_{\text{C}}, u_{\text{H}}] = [0.5, 1]$ to the values chosen in ref. ⁴², the agent discovers a cycle that outperforms the optimized Otto cycle (Fig. 5a). If we further allow the agent to modulate $u(t)$ in a larger interval, we find a cycle with an even larger power (Fig. 5b).

The protocol discovered by the agent in Fig. 5a resembles the Otto cycle (both are flat for some time at $u = u_{\text{C}}$ and $u = u_{\text{H}}$), yet there are additional features. First, the agent ramps up the control $u(t)$ between u_{C} and u_{H} in a non-linear fashion while in contact with a bath, rather than decoupling the system. Next, there are two strong discontinuities where the system is abruptly disconnected from the baths for a short time (Fig. 5c, green segments). These additional features turn out to enhance the power. At last, we notice that the cycle found by the agent in Fig. 5d is more regular than the one in Fig. 5c, and it further deviates from an Otto cycle. The crucial difference with respect to the Otto cycle,

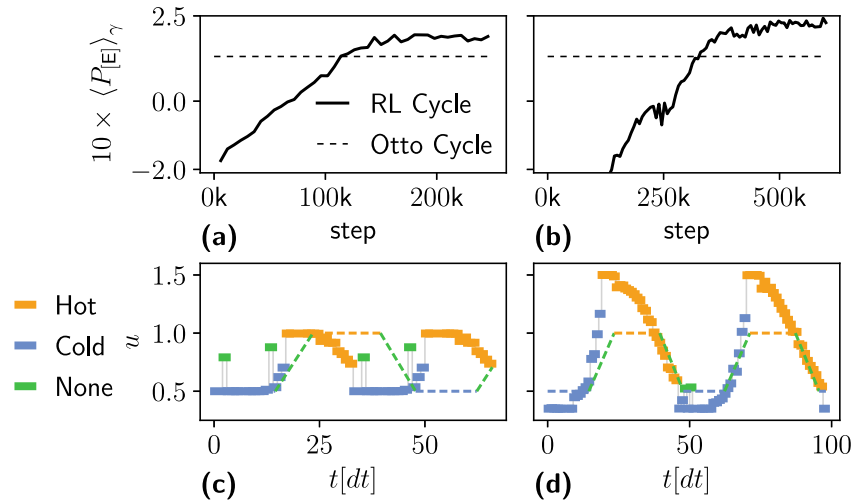


Fig. 5 Results of training the agent on the quantum harmonic oscillator heat engine model. **a, b** Running average of the extracted power $\langle P_{[E]} \rangle_{\gamma}$ as a function of the step during the whole training. The dashed line represents the maximum power found in ref. ⁴² optimizing the duration of each stroke of an Otto cycle (see “Methods” for details). **c, d** Actions chosen by the agent, represented by the value of u , as a function of time. The color represents action d , i.e. the bath coupled to S . The discontinuous thick line is the final cycle discovered by the agent, while the thin dashed line is the Otto cycle whose power is given by the dashed line in panels **(a, b)**. The parameters used for training are $\Gamma^{(H)} = \Gamma^{(C)} = 0.6$, $\beta_H = 0.2$, $\beta_C = 2$, $w_0 = 2$, $\Delta t = 0.2$, and $\gamma = 0.998$. In panels **a–c** we set $\mathcal{U} = [0.5, 1]$ to enable a fair comparison with the Otto cycle of ref. ⁴², while in **b–d** we set a larger control range, i.e., $\mathcal{U} = [0.35, 1.5]$.

therefore, seems to be the ramping up of $u(t)$ while in contact with the baths, which probably benefits from simultaneously exchanging heat and modulating $u(t)$, rather than doing them in two separate strokes.

As in the two-level case, we evaluate the EMP of the discovered cycles shown in Fig. 5c, d. In both cases the EMP is considerably high, corresponding respectively to 60% and 78% of η_{car} , equivalent to 46% and 59% of η_c . Interestingly, the cycle shown in Fig. 5d yields both a higher power and a larger EMP than the cycle displayed in Fig. 5c.

DISCUSSION

We introduced a general framework based on Reinforcement Learning to discover thermodynamic cycles that maximize the power of out-of-equilibrium quantum thermal machines, paving the way for more systematic use of machine learning in the field of quantum thermodynamics. Using state-of-the-art machine learning techniques, we applied our method to three different paradigmatic setups. Our method found the optimal known solution in the benchmark system, while in the other systems it discovered new unintuitive and elaborate cycles which outperform previously proposed cycles.

Our results show that the celebrated Otto cycle is not in general optimal for power extraction and that carefully crafted cycles can mitigate coherence-induced power losses^{48,54,70,71} without introducing additional controls as required by Shortcuts to Adiabaticity^{56–63}. As opposed to other optimal control techniques, such as the Pontryagin Minimum Principle^{64–66}, our RL-based approach has the following advantages: it does not require any analytic calculation; it can handle both continuous and discrete controls (such as the choice of the heat bath), and it can naturally find discontinuous and irregular protocols; it can be applied as-is to arbitrarily complicated setups and it could be used to find optimal protocols also in the presence of noise in the controls.

Future research directions include applications to multi-particle systems, where many-body advantages might be revealed, and a systematic study of the mitigation of coherence-induced power losses. Interesting extensions of our framework include investigating the strong system-bath coupling regime going beyond a master equation approach^{101–103}, optimizing additional

thermodynamic quantities, such as minimizing the fluctuations in the power output and developing of a scheme that can be applied directly to experimental setups that does not require the knowledge of the quantum state.

METHODS

Physical model

As discussed in the main text, we assume that the state evolves according to the Markovian master Eq. (1), which can be derived, also for non-adiabatic drivings⁹⁵, in the weak system-bath coupling regime performing the usual Born-Markov and secular approximation^{92–94} and neglecting the Lamb-shift contribution. We notice that since the RL agent produces piecewise constant protocols, we are not impacted by possible inaccuracies of the master equation subject to fast parameter driving¹⁰⁴, provided that Δt is not smaller than the bath timescale. Without loss of generality, the dissipators can be expressed as^{93,94}

$$\mathcal{D}_{\mathbf{u}(t), d(t)}^{(a)}[\hat{\rho}] = \lambda_a[d(t)] \sum_k \gamma_{k, \mathbf{u}(t)}^{(a)} \left(\hat{A}_{k, \mathbf{u}(t)}^{(a)} \hat{\rho} \hat{A}_{k, \mathbf{u}(t)}^{(a)\dagger} - \frac{1}{2} \hat{A}_{k, \mathbf{u}(t)}^{(a)\dagger} \hat{A}_{k, \mathbf{u}(t)}^{(a)} \hat{\rho} - \frac{1}{2} \hat{\rho} \hat{A}_{k, \mathbf{u}(t)}^{(a)\dagger} \hat{A}_{k, \mathbf{u}(t)}^{(a)} \right), \quad (9)$$

where $\lambda_a[d(t)] \in \{0, 1\}$ are functions that determine which bath is coupled the QS, $\hat{A}_{k, \mathbf{u}(t)}^{(a)}$ are the Lindblad operators, and $\gamma_{k, \mathbf{u}(t)}^{(a)}$ are the corresponding rates. In particular, $\lambda_H(\text{Hot}) = 1$, $\lambda_C(\text{Hot}) = 0$, while $\lambda_H(\text{Cold}) = 0$, $\lambda_C(\text{Cold}) = 1$, and $\lambda_H(\text{None}) = \lambda_C(\text{None}) = 0$. Notice that both the Lindblad operators and the rates can depend on time through the value of the control $\mathbf{u}(t)$. Their explicit form depends on the details of the system, i.e. on the Hamiltonian describing the dynamics of the overall system including the bath and the system-bath interaction. Below, we provide the explicit form of $\hat{A}_{k, \mathbf{u}(t)}^{(a)}$ and $\gamma_{k, \mathbf{u}(t)}^{(a)}$ used to model the three setups considered in the manuscript. We adopt the standard approach to compute the instantaneous power and heat currents²³

$$P(t) \equiv -\text{Tr} \left[\hat{\rho}(t) \frac{\partial}{\partial t} \hat{H}[\mathbf{u}(t)] \right], J_a(t) \equiv \text{Tr} \left[\mathcal{D}_{\mathbf{u}(t), d(t)}^{(a)}[\hat{\rho}(t)] \hat{H}[\mathbf{u}(t)] \right]. \quad (10)$$

In the two-level system heat engine, we consider the following Lindblad operators and corresponding rates (identifying $k = \pm$):

$$\hat{A}_{\pm, \mathbf{u}(t)}^{(a)} = \hat{\sigma}_{\pm}, \gamma_{\pm, \mathbf{u}(t)}^{(a)} = \Gamma_a f(\pm \beta_a u(t) E_0), \quad (11)$$

where $\hat{\sigma}_+$ and $\hat{\sigma}_-$ denote the raising and lowering operators, Γ_a is a constant rate which sets the thermalization timescale when the QS is coupled to bath a , and $f(x) = [1 + \exp(x)]^{-1}$ is the Fermi distribution. This choice can be derived, for example, when considering the qubit as a single-level quantum dot tunnel-coupled to metallic leads, with flat density of states, which act as heat baths^{105–108}.

In the superconducting qubit refrigerator, we employ the model first put forward in ref. 48, and further studied in refs. 54,62. In particular, we consider the following Lindblad operators and corresponding rates (identifying $k = \pm$):

$$\hat{A}_{+,u(t)}^{(a)} = -i|e_{u(t)}\rangle\langle g_{u(t)}|, \hat{A}_{-,u(t)}^{(a)} = +i|g_{u(t)}\rangle\langle e_{u(t)}|, \quad (12)$$

where $|g_{u(t)}\rangle$ and $|e_{u(t)}\rangle$ are, respectively, the instantaneous ground state and excited state of Eq. (7). The corresponding rates are given by $\gamma_{\pm,u(t)}^{(a)} = S_a[\pm \Delta\epsilon_{u(t)}]$, where $\Delta\epsilon_{u(t)}$ is the instantaneous energy gap of the system, and

$$S_a(\Delta\epsilon) = \frac{g_a}{2} \frac{1}{1 + Q_a^2(\Delta\epsilon/\omega_a - \omega_a/\Delta\epsilon)^2} \frac{\Delta\epsilon}{e^{\beta_a\Delta\epsilon} - 1} \quad (13)$$

is the noise power spectrum of bath a . Here ω_a , Q_a , and g_a are the base resonance frequency, quality factor, and coupling strength of the resonant circuit acting as bath $a = H, C$ (see refs. 48,62 for details). As in ref. 62, we choose $\omega_C = 2E_0\Delta$ and $\omega_H = 2E_0\sqrt{\Delta^2 + 1/4}$, such that the C (H) bath is in resonance with the qubit when $u = 0$ ($u = 1/2$). The width of the resonance is governed by Q_a . The total coupling strength to bath a , plotted in Fig. 4d, is quantified by

$$\gamma_{u(t)}^{(a)} \equiv \gamma_{+,u(t)}^{(a)} + \gamma_{-,u(t)}^{(a)}. \quad (14)$$

In the Harmonic oscillator heat engine, following ref. 42, we describe the coupling to the baths through the Lindblad operators $\hat{A}_{+,u(t)}^{(a)} = \hat{a}_{u(t)}^\dagger$, $\hat{A}_{-,u(t)}^{(a)} = \hat{a}_{u(t)}$ and corresponding rates $\gamma_{\pm,u(t)}^{(a)} = \Gamma_a n(\beta_a u(t)\omega_0)$, and $\gamma_{-,u(t)}^{(a)} = \Gamma_a [1 + n(\beta_a u(t)\omega_0)]$, where we identify $k = \pm$. $\hat{a}_{u(t)} = (1/\sqrt{2})\sqrt{m\omega_0 u(t)} \hat{q} + i/\sqrt{m\omega_0 u(t)} \hat{p}$ and $\hat{a}_{u(t)}^\dagger$ are respectively the (control dependent) lowering and raising operators, Γ_a is a constant rate setting the thermalization timescale of the system coupled to bath a , and $n(x) = [\exp(x) - 1]^{-1}$ is the Bose–Einstein distribution.

Reinforcement learning algorithm

As discussed in the Results section, the choice of the reward as in Eq. (5) guarantees that the aim of the RL agent is to maximize $\langle P_{[v]} \rangle$ introduced in Eq. (3). To be precise, plugging Eq. (5) into Eq. (4) gives $\langle P_{[v]} \rangle$ (up to an irrelevant constant prefactor) only in the limit of $\Delta t \rightarrow 0$. However, also for finite Δt , both quantities are time averages of the power, so they are equally valid definitions to describe a long-term power maximization.

We use a generalization of the SAC method, first developed for continuous actions^{73,74}, to handle a combination of discrete and continuous actions^{75,76}. We here present an overview of our implementation of SAC putting special emphasis on the differences with respect to the standard implementation. However, we refer to^{73–76} for additional details. Our method, implemented with PyTorch, is based on modifications and generalizations of the SAC implementation provided by Spinning Up from OpenAI¹⁰⁹. All code and data to reproduce the experiments are available online (see Data Availability and Code Availability sections).

The SAC algorithm is based on policy iteration, i.e. it consists of iterating multiple times over two steps: a policy evaluation step, and a policy improvement step. In the policy evaluation step, the value function of the current policy is (partially) learned, whereas in the policy improvement step a better policy is learned by making use of the value function. We now describe these steps more in detail.

In typical RL problems, the optimal policy $\pi^*(s|a)$ is defined as the policy that maximizes the expected reward defined in Eq. (4), i.e.:

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k r_{k+1} | s_0 = s \right], \quad (15)$$

where \mathbb{E}_{π} denotes the expectation value choosing actions according to the policy π . The initial state $s_0 = s$ is sampled from μ_{π} i.e. the steady-state distribution of states that are visited by π . In the SAC method, the balance between exploration and exploitation⁷² is achieved by introducing an

entropy-regularized maximization objective. In this setting, the optimal policy π^* is given by

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[\sum_{k=0}^{\infty} \gamma^k (r_{k+1} + \epsilon H(\pi(\cdot|s_k))) | s_0 = s \right], \quad (16)$$

where $\epsilon \geq 0$ (usually denoted with α in the RL literature) is a hyperparameter that balances the trade-off between exploration and exploitation, and

$$H(P) = \mathbb{E}_{x \sim P} [-\log P(x)] \quad (17)$$

is the entropy of the probability distribution P . Notice that we replaced the unknown state distribution μ_{π} with \mathcal{B} , which is a replay buffer populated during training by storing the observed one-step transitions $(s_k, a_k, r_{k+1}, s_{k+1})$.

We define the value function $Q^{\pi}(s, a)$ of a given policy π as

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi} \left[r_1 + \sum_{k=1}^{\infty} \gamma^k (r_{k+1} + \epsilon H(\pi(\cdot|s_k))) | s_0 = s, a_0 = a \right], \quad (18)$$

and its recursive Bellman equation reads

$$Q^{\pi}(s, a) = \mathbb{E}_{a_1 \sim \pi(\cdot|s_1)} [r_1 + \gamma(Q^{\pi}(s_1, a_1) - \epsilon \log \pi(a_1|s_1)) | s_0 = s, a_0 = a]. \quad (19)$$

We use a function approximator $Q_{\phi}(s, a)$ (e.g. a neural network) to describe the value function of the current policy, where ϕ represents a collection of learnable parameters.

Next, we describe the policy $\pi(a|s)$. Since we are dealing with a combination of discrete and continuous actions, we define $a = (u, d)$, where u is the continuous action and d is the discrete action (for simplicity, we describe the case of a single continuous action, though the generalization to multiple variables is straightforward). From now on, all functions of a are also to be considered as functions of u, d . We decompose the joint probability distribution as

$$\pi_{\theta}(u, d|s) = \pi_{\theta_0}(d|s) \cdot \pi_{\theta_{u,d}}(u|d, s), \quad (20)$$

where $\pi_{\theta_0}(d|s)$ is a function approximator for the marginal probability of taking discrete action d , which depends on learnable parameters θ_{θ_0} , and $\pi_{\theta_{u,d}}(u|d, s)$ is a parameterization of the conditional probability density of choosing action u , given action d , which depends on learnable parameters $\theta_{u,d}$ —one set for each discrete action d . We denote with θ the collection of all parameters θ_{θ_0} and $\theta_{u,d}$. Notice that this decomposition allows us to describe correlations between the discrete and the continuous action, which are crucial in our application. We further parameterize $\pi_{\theta_{u,d}}(u|d, s)$ as a squashed Gaussian policy, i.e. as the distribution of the variable

$$\tilde{u}_{\theta_{u,d}}(s, \xi) = u_a + \frac{u_b - u_a}{2} [1 + \tanh(\mu_{\theta_{u,d}}(s) + \sigma_{\theta_{u,d}}(s) \cdot \xi)], \quad (21)$$

$$\xi \sim \mathcal{N}(0, 1),$$

where $\mu_{\theta_{u,d}}(s)$ and $\sigma_{\theta_{u,d}}(s)$, representing respectively the mean and standard deviation of the Gaussian distribution, are function approximators which depend on the learnable parameters $\theta_{u,d}$. $\mathcal{N}(0, 1)$ is the normal distribution with zero mean and unit variance, and where we assume that $\mathcal{U} = [u_a, u_b]$.

We now describe the policy evaluation step. In the SAC algorithm, we learn two value functions $Q_{\phi_i}(s, a)$ described by the learnable parameters ϕ_i , for $i = 1, 2$. Since $Q_{\phi_i}(s, a)$ should satisfy the Bellman Eq. (19), we define the loss function for $Q_{\phi_i}(s, a)$ as the mean square difference between the left and right-hand side of Eq. (19), i.e.

$$L_Q(\phi_i) = \mathbb{E}_{(s,a,r,s') \sim \mathcal{B}} [(Q_{\phi_i}(s, a) - y(r, s'))^2], \quad (22)$$

where

$$y(r, s') = r + \gamma \mathbb{E}_{a' \sim \pi_{\theta}(\cdot|s')} \left[\min_{j=1,2} Q_{\phi_j}^{\text{target}}(s', a') - \epsilon \log \pi_{\theta}(a'|s') \right]. \quad (23)$$

Notice that in Eq. (23) we replaced Q_{ϕ_i} with $\min_{j=1,2} Q_{\phi_j}^{\text{target}}$, where ϕ_j^{target} , for $j = 1, 2$, are target parameters which are not updated when minimizing the loss function; instead, they are held fixed during backpropagation, and then they are updated according to polyak averaging, i.e.

$$\phi_j^{\text{target},i} \leftarrow \rho_{\text{polyak}} \phi_j^{\text{target},i} + (1 - \rho_{\text{polyak}}) \phi_i, \quad (24)$$

where ρ_{polyak} is a hyperparameter. This change was shown to improve learning^{73,74}. In order to evaluate the expectation value in Eq. (23), we use

the decomposition in Eq. (20) to write

$$\mathbb{E}_{d' \sim \pi_{\theta}(d'|s')} [\cdot] = \sum_{d'} \pi_{\theta_b}(d'|s') \mathbb{E}_{u' \sim \pi_{\theta_{u,d'}}(\cdot|d',s')}, \quad (25)$$

where we denote $d' = (u', d')$. Plugging Eq. (25) into Eq. (23), and approximating the expectation value over u' with a single sampled value yields

$$y(r, s') = r + \gamma \sum_{d'} \pi_{\theta_b}(d'|s') \cdot \left[\min_{j=1,2} Q_{\phi_{\text{target}}}^j(s', d', u') - \text{alog} \pi_{\theta}(d', u'|s') \right],$$

$$u' \sim \pi_{\theta_{u,d'}}(\cdot|d', s'). \quad (26)$$

We, therefore, perform a full average over the discrete action, and a single sampling of the continuous action.

We now turn to the policy improvement step. Given a policy $\pi_{\theta_{\text{old}}}$, ref. ⁷⁴ proves that $\pi_{\theta_{\text{new}}}$ is a better policy [respect to maximization in Eq. (16)] if we update the policy parameters according to

$$\theta_{\text{new}} = \arg \min_{\theta} D_{\text{KL}} \left(\pi_{\theta}(\cdot|s) \parallel \frac{\exp(Q^{\pi_{\theta_{\text{old}}}}(s, \cdot)/\alpha)}{Z^{\pi_{\theta_{\text{old}}}}} \right), \quad (27)$$

where s is any state, D_{KL} denotes the Kullback-Leibler divergence and $Z^{\pi_{\theta_{\text{old}}}}$ is the partition function of the exponential of the value function. Intuitively, this step is the equivalent of making the policy ϵ -greedy in the standard RL setting. The idea is to use the minimization in Eq. (27) to define a loss function to perform an update of θ . Noting that the partition function does not impact the gradient, multiplying the Kullback-Leibler divergence by α , and replacing $Q^{\pi_{\theta_{\text{old}}}}$ with $\min_j Q_{\phi_j}$, we define the loss function as

$$L_{\pi}(\theta) = \mathbb{E}_{\substack{s \sim \mathcal{B} \\ a \sim \pi_{\theta}(\cdot|s)}} \left[\text{alog} \pi_{\theta}(a|s) - \min_{j=1,2} Q_{\phi_j}(s, a) \right]. \quad (28)$$

In order to evaluate the expectation value in Eq. (28), we use the previous trick of averaging the discrete action and performing a single sample of the continuous action using ξ . Recalling Eq. (21), this yields

$$L_{\pi}(\theta) = \mathbb{E}_{s \sim \mathcal{B}} \left[\sum_{d'} \pi_{\theta_b}(d'|s) \left(\text{alog} \pi_{\theta}(d, \bar{u}_{\theta_{u,d}}(s, \xi)|s) - \min_{j=1,2} Q_{\phi_j}(s, d, \bar{u}_{\theta_{u,d}}(s, \xi)) \right) \right],$$

$$\xi \sim \mathcal{N}(0, 1). \quad (29)$$

To summarize, the SAC algorithm consists of repeating over and over a policy evaluation step, and a policy improvement step. The policy evaluation step consists of a single optimization step to minimize the loss functions $L_Q(\phi_i)$ (for $i=1,2$), given in Eq. (22), where $y(r, s')$ is computed using Eq. (26). The policy improvement step consists of a single optimization step to minimize the loss function $L_{\pi}(\theta)$ given in Eq. (29). In both loss functions, the expectation value over the states is approximated with a batch of experiences sampled randomly from the replay buffer \mathcal{B} .

Training details

We now provide the details of the algorithm used to learn the four specific cycles described in the main manuscript.

The value function $Q_{\phi}(s, u, d)$ is parameterized the following way. We use a fully connected neural network (NN), with two hidden layers, that take s and u as input (by stacking them into a single array), and output $|D|$ values, where $|D|$ is the number of discrete actions. The i_{th} output corresponds to $Q_{\phi}(s, u, d = d_i)$, where $\{d_i\}$ are the possible discrete actions. We use the ReLU activation function in all layers except for the output layer, where we apply the identity (since the value function can take arbitrary positive or negative values). The parameters ϕ correspond to the weights and biases of the whole network.

The policy $\pi_{\theta}(u, d|s) = \pi_{\theta_b}(d|s) \cdot \pi_{\theta_{u,d}}(u, d|s)$ is parameterized the following way. We use a fully connected NN, with two hidden layers, that takes s as input, and produces $3 \cdot |D|$ values, corresponding to

$$\left\{ \pi_{\theta_b}(d_i|s), \mu_{\theta_{u,d_i}}(s), \sigma_{\theta_{u,d_i}}(s) \right\}_{i=1, \dots, |D|}. \quad (30)$$

More specifically, we use the ReLU activation in all layers except for the output layer, where we use the identity. However, in order to enforce the normalization $\sum_i \pi_{\theta_b}(d_i|s) = 1$, we apply a soft-max to the corresponding outputs, and instead of outputting $\sigma_{\theta_{u,d_i}}(s)$, we output $\log(\sigma_{\theta_{u,d_i}}(s))$, which has no constraint on the sign.

In order to enforce sufficient exploration in the early stage of training, we do the following. For a fixed number of initial steps, we choose random

Table 1. Hyperparameters used in numerical calculations that are not reported in the figure captions.

Hyperparameter	Figs. 3, 4	Fig. 5a	Fig. 5b
Hidden layers	2	2	2
Hidden layer units	256	256	256
Initial random steps	5k	5k	10k
First update at step	1k	1k	1k
Batch size	256	256	256
learning rate	0.001	0.001	0.0005
ϵ_0	50	50	300
ϵ_{decay}	48k	24k	48k
n_{updates}	50	50	50
ρ_{polyak}	0.995	0.995	0.995
\mathcal{B} size	192k	192k	192k

actions sampling them uniformly within their range. Furthermore, for another fixed number of initial steps, we do not update the parameters to allow the replay buffer to have enough transitions. \mathcal{B} is a first-in-first-out buffer, of fixed dimension, from which batches of transitions are randomly sampled to update the NN parameters. After this initial phase, we repeat a policy evaluation and a policy improvement step n_{updates} times every n_{updates} steps. This way, the overall number of updates coincides with the number of actions performed on the environment. The optimization steps are performed using the ADAM optimizer with the standard values of β_1 and β_2 . To favor an exploratory behavior early in the training, and at the same time to end up with a policy that is approximately deterministic, we schedule ϵ . In particular, we vary it during each step according to

$$\epsilon(n_{\text{steps}}) = \epsilon_0 \exp(-n_{\text{steps}}/\epsilon_{\text{decay}}), \quad (31)$$

where n_{steps} is the current step number, and ϵ_0 and ϵ_{decay} are hyperparameters.

All hyperparameters used to produce the cycles in Figs. 3, 4, and 5 are provided in Table 1.

At last, we discuss the parameterization of the state s . As discussed in the manuscript, we use $s = (\hat{\rho}, u)$ as state. While u is passed to the NNs as-is, we now detail how we encoded $\hat{\rho}$ in the 3 systems studied in the manuscript.

In the two-level system heat engine, a closed equation of motion governing the evolution of $\rho(t) \equiv \text{Tr}[\hat{\rho}(t)\hat{\sigma}_+ \hat{\sigma}_-]$ can be derived from the Markovian master equation, and the instantaneous heat flux $J_a(t)$ can be expressed solely in terms of $\rho(t)$ (see ref. ⁹⁰ for details). Therefore, we use the single parameter $\rho(t) \in [0, 1]$ to encode $\hat{\rho}$.

In the superconducting qubit refrigerator, we encode $\hat{\rho}$ using the following three real parameters: $\langle e_{u(t)}|\hat{\rho}(t)|e_{u(t)}\rangle$, $\text{Re}[\langle g_{u(t)}|\hat{\rho}(t)|e_{u(t)}\rangle]$ and $\text{Im}[\langle g_{u(t)}|\hat{\rho}(t)|e_{u(t)}\rangle]$. These fully characterize a density matrix of a qubit.

In the Harmonic oscillator heat engine a closed set of equations of motion can be derived from the Markovian master equation for the following three quantities:

$$H(t) \equiv \text{Tr}[\hat{\rho}(t)\hat{H}[u(t)]],$$

$$L(t) \equiv \text{Tr}[\hat{\rho}(t)(\frac{1}{2m}\hat{p}^2 - \frac{1}{2}m(u(t)w_0)^2\hat{q}^2)], \quad (32)$$

$$D(t) \equiv \text{Tr}[\hat{\rho}(t)(\hat{q}\hat{p} + \hat{p}\hat{q})].$$

Furthermore, the heat flux $J_a(t)$ can be expressed solely in terms of these quantities (see ref. ⁴² for details). We thus use these three quantities to encode $\hat{\rho}$. More specifically, since they are not bounded in an obvious way by some system parameter, for numerical stability we use $\tilde{O}(t) \equiv \log(|O(t)| + \delta)$ and $sO(t) = \text{sign}(O(t))$ instead of $O = H, L, D$ to encode the state, where $\delta = 10^{-20}$ is a small parameter introduced to prevent numerical divergences. At last, we do not use $sH(t)$ since $H(t)$ is always a positive quantity (being the energy of the harmonic oscillator). Therefore, we encode $\hat{\rho}$ using the following 5 parameters: $(\tilde{H}, \tilde{L}, \tilde{D}, sL, sD)$.

Convergence of the RL approach

The training process presents some degree of stochasticity, such as the initial random steps, and the random sampling of a batch of experience

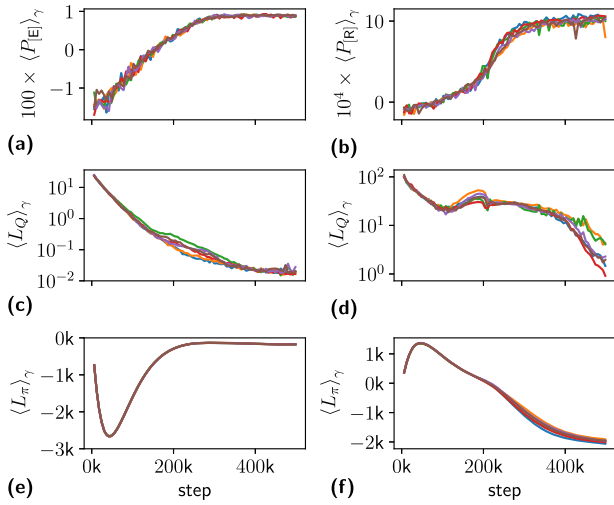


Fig. 6 Training curves. Training curves for six consecutive runs of the RL method in the two-level system heat engine case [(a, c, and e), corresponding to Fig. 3] and in the superconducting qubit refrigerator case [(b, d, and f), corresponding to Fig. 4]. The panels show, as a function of the training steps, the running average of the power (a, b), the running average of the loss function L_Q (c, d), and the running average of the loss function L_π (e, f). Each curve corresponds to a separate training.

from the replay buffer to compute an approximate gradient of the loss functions. We thus need to evaluate the reliability of our approach.

In Figs. 6 and 7, we show the training curves for six consecutive runs of our method applied to all four cases studied in the main text. More specifically, panels (a, c, and e) of Fig. 6 correspond to the training of the two-level system heat engine (considered in Fig. 3), while panels (b, d, and f) of Fig. 6 correspond to the training of the superconducting qubit refrigerator (considered in Fig. 4). Panels (a, c, and e) of Fig. 7 correspond to the training of the Harmonic oscillator in the interval considered in Fig. 5a, c, while panels (b, d, and f) of Fig. 7 correspond to the training of the Harmonic oscillator in the interval considered in Fig. 5b, d. Both Figs. 6 and 7 show, as a function of the step, the running average of the power [panels (a) and (b)], the running average $\langle L_Q \rangle_\gamma$ of the loss function L_Q [panels (c, d)] and the running average $\langle L_\pi \rangle_\gamma$ of the loss function L_π [panels (e) and (f)] computed on the batch of experience used to estimate the gradient of the corresponding loss function. Every curve corresponds to a separate training.

As we can see in Figs. 6 and 7, in all four cases the running average of the reward reliably converge to a solution yielding similar values of the power, and also the running averages of the loss functions display qualitatively similar behavior. We notice that while L_Q is the mean square of the Bellman error [see Eq. (22)], L_π is just a function whose gradient provides a better policy, so its value during training is not required to be a decreasing function.

Generation of coherence

In order to quantify the coherence generated in the instantaneous eigenbasis of the Hamiltonian in the refrigerator based on a superconducting qubit, we evaluated the time average of relative entropy of coherence¹⁰⁰, defined as

$$C(\hat{\rho}(t)) = S(\hat{\rho}_{\text{diag}}(t)) - S(\hat{\rho}(t)), \quad (33)$$

where $S(\hat{\rho}) = -\text{Tr}[\hat{\rho} \ln \hat{\rho}]$ is the Von Neumann entropy, and

$$\hat{\rho}_{\text{diag}}(t) = \langle g_{u(t)} | \hat{\rho}(t) | g_{u(t)} \rangle \cdot |g_{u(t)}\rangle \langle g_{u(t)}| + \langle e_{u(t)} | \hat{\rho}(t) | e_{u(t)} \rangle \cdot |e_{u(t)}\rangle \langle e_{u(t)}| \quad (34)$$

is the density matrix, in the instantaneous eigenbasis $|g_{u(t)}\rangle$ and $|e_{u(t)}\rangle$, with the off-diagonal terms canceled out. We find that that the time average of the relative entropy of coherence is ≈ 0.116 in the cycle found by the RL agent, while it is ≈ 0.194 applying the trapezoidal cycle with the same period.

Otto cycle comparison

In Fig. 5 we compare the performance of an Otto cycle, optimized as in the upper panel of Fig. 6 of ref. 42, with the cycle discovered by the RL agent.

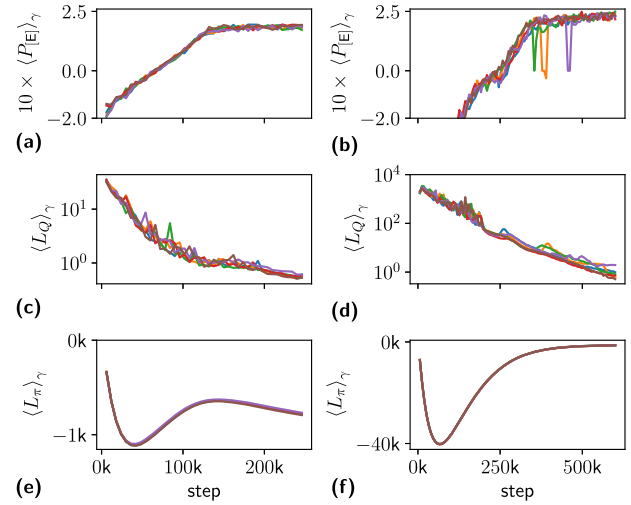


Fig. 7 Training curves. Training curves for six consecutive runs of the RL method in the harmonic oscillator heat engine. Panels (a), (c), and (e) correspond to the optimization carried out in the range considered in Fig. 5a, c, while panels (b), (d), and (f) correspond to the optimization carried out in Fig. 5b, d. The panels show, as a function of the training steps, the running average of the power [panels (a) and (b)], the running average of the loss function L_Q [panels (c) and (d)], and the running average of the loss function L_π [panels (e) and (f)]. Each curve corresponds to a separate training.

However, ref. 42 only provides the value of the power of the optimized Otto cycle, not the duration of the four strokes that produce such power. We, therefore, performed a grid search in the space of these four durations. After identifying the largest power, we ran the Netwon algorithm to further maximize the power. The final cycle we found is the one shown as dashed lines in Fig. 5c, d. The corresponding power, shown as a dashed line in Fig. 5a, b, nicely matches with ref. 42.

DATA AVAILABILITY

All raw data was generated with the accompanying code and is available in Figshare (<https://doi.org/10.6084/m9.figshare.16822054.v1>).

CODE AVAILABILITY

The code used to generate all results is available on GitHub (https://github.com/PaoloAE/paper_rl_thermal_machines).

Received: 31 August 2021; Accepted: 28 November 2021;

Published online: 10 January 2022

REFERENCES

- Fagas, G., Gammaitoni, L., Paul, D. & Berini, G. A. *ICT—Energy—Concepts Towards Zero Power Information and Communication Technology* (InTech, 2014).
- Pekola, J. P. Towards quantum thermodynamics in electronic circuits. *Nat. Phys.* **11**, 118 (2015).
- Giazotto, F., Heikkilä, T. T., Luukanen, A., Savin, A. M. & Pekola, J. P. Opportunities for mesoscopics in thermometry and refrigeration: physics and applications. *Rev. Mod. Phys.* **78**, 217 (2006).
- Binder, F., Correa, L., Gogolin, C., Anders, J. & Adesso, G. (eds) *Thermodynamics in the Quantum Regime: Fundamental Aspects and New Directions* (Springer International Publishing, 2019).
- Vinjanampathy, S. & Anders, J. Quantum thermodynamics. *Contemp. Phys.* **57**, 545 (2016).
- Friedenauer, H., Schmitz, H., Glueckert, J., Porras, D. & Schaetz, T. Simulating a quantum magnet with trapped ions. *Nat. Phys.* **4**, 757 (2008).
- Blatt, R. & Roos, C. Quantum simulations with trapped ions. *Nat. Phys.* **8**, 277 (2012).
- Childress, L. et al. Coherent dynamics of coupled electron and nuclear spin qubits in diamond. *Science* **314**, 281 (2006).

9. Wallraff, A. et al. Strong coupling of a single photon to a superconducting qubit using circuit quantum electrodynamics. *Nature* **431**, 162 (2004).
10. Petta, J. R. et al. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science* **309**, 2180 (2005).
11. Ronzani, A. et al. Tunable photonic heat transport in a quantum heat valve. *Nat. Phys.* **14**, 991 (2018).
12. Dutta, B. et al. Direct probe of the Seebeck coefficient in a kondo-correlated single-quantum-dot transistor. *Nano Lett.* **19**, 506 (2019).
13. Maillet, O., Subero, D., Peltonen, J. T., Golubev, D. S. & Pekola, J. P. Electric field control of radiative heat transfer in a superconducting circuit. *Nat. Commun.* **11**, 4326 (2020).
14. Senior, J. et al. Heat rectification via a superconducting artificial atom. *Commun. Phys.* **3**, 40 (2020).
15. Roßnagel, J. et al. A single-atom heat engine. *Science* **352**, 325 (2016).
16. Josefsson, M. et al. A quantum-dot heat engine operating close to the thermodynamic efficiency limits. *Nat. Nanotechnol.* **13**, 920 (2018).
17. Klatzow, J. et al. Experimental demonstration of quantum effects in the operation of microscopic heat engines. *Phys. Rev. Lett.* **122**, 110601 (2019).
18. von Lindenfels, D. et al. Spin heat engine coupled to a harmonic-oscillator flywheel. *Phys. Rev. Lett.* **123**, 080602 (2019).
19. Maslennikov, G. et al. Quantum absorption refrigerator with trapped ions. *Nat. Commun.* **10**, 202 (2019).
20. Peterson, J. P. S. et al. Experimental characterization of a spin quantum heat engine. *Phys. Rev. Lett.* **123**, 240601 (2019).
21. Prete, D. et al. Thermoelectric conversion at 30 k in InAs/InP nanowire quantum dots. *Nano Lett.* **19**, 3033 (2019).
22. Horne, N. V. et al. Single-atom energy-conversion device with a quantum load. *NPJ Quantum Inf.* **6**, 37 (2020).
23. Alicki, R. The quantum open system as a model of the heat engine. *J. Phys. A: Math. Gen.* **12**, L103 (1979).
24. Esposito, M., Kawai, R., Lindenberg, K. & den Broeck, C. V. Efficiency at maximum power of low-dissipation Carnot engines. *Phys. Rev. Lett.* **105**, 150603 (2010a).
25. Wang, J., He, J. & He, X. Performance analysis of a two-state quantum heat engine working with a single-mode radiation field in a cavity. *Phys. Rev. E* **84**, 041127 (2011).
26. Avron, J. E., Fraas, M., Graf, G. M. & Grech, P. Adiabatic theorems for generators of contracting evolutions. *Commun. Math. Phys.* **314**, 163 (2012).
27. Ludovico, M. F., Battista, F., von Oppen, F. & Arrachea, L. Adiabatic response and quantum thermoelectrics for ac-driven quantum systems. *Phys. Rev. B* **93**, 075136 (2016).
28. Cavina, V., Mari, A. & Giovannetti, V. Slow dynamics and thermodynamics of open quantum systems. *Phys. Rev. Lett.* **119**, 050601 (2017).
29. Abiuso, P. & Giovannetti, V. Non-Markov enhancement of maximum power for quantum thermal machines. *Phys. Rev. A* **99**, 052106 (2019).
30. Bhandari, B. et al. Geometric properties of adiabatic quantum thermal machines. *Phys. Rev. B* **102**, 155407 (2020).
31. Abiuso, P. & Perarnau-Llobet, M. Optimal cycles for low-dissipation heat engines. *Phys. Rev. Lett.* **124**, 110606 (2020).
32. Abiuso, P., Mille, H. J. D., Perarnau-Llobet, M. & Scandi, M. Geometric optimisation of quantum thermodynamic processes. *Entropy* **22**, 1076 (2020).
33. Cavina, V., Erdman, P. A., Abiuso, P., Tolomeo, L. & Giovannetti, V. Maximum power heat engines and refrigerators in the fast-driving regime. *Phys. Rev. A* **104**, 032226 (2021).
34. Arrachea, L., Moskalets, M. & Martin-Moreno, L. Heat production and energy balance in nanoscale engines driven by time-dependent fields. *Phys. Rev. B* **75**, 245420 (2007).
35. Esposito, M., Kawai, R., Lindenberg, K. & Van den Broeck, C. Quantum-dot Carnot engine at maximum power. *Phys. Rev. E* **81**, 041106 (2010b).
36. Juergens, S., Haupt, F., Moskalets, M. & Splettstoesser, J. Thermoelectric performance of a driven double quantum dot. *Phys. Rev. B* **87**, 245423 (2013).
37. Campisi, M., Pekola, J. & Fazio, R. Nonequilibrium fluctuations in quantum heat engines: theory, example, and possible solid state experiments. *New J. Phys.* **17**, 035012 (2015).
38. Dann, R. & Kosloff, R. Quantum signatures in the quantum Carnot cycle. *New J. Phys.* **22**, 013055 (2020).
39. Molitor, O. A. D. & Landi, G. T. Stroboscopic two-stroke quantum heat engines. *Phys. Rev. A* **102**, 042217 (2020).
40. Feldmann, T., Geva, E., Kosloff, R. & Salamon, P. Heat engines in finite time governed by master equations. *Am. J. Phys.* **64**, 485 (1996).
41. Feldmann, T. & Kosloff, R. Performance of discrete heat engines and heat pumps in finite time. *Phys. Rev. E* **61**, 4774 (2000).
42. Rezek, Y. & Kosloff, R. Irreversible performance of a quantum harmonic heat engine. *New J. Phys.* **8**, 83 (2006).
43. Quan, H., Liu, Y., Sun, C. & Nori, F. Quantum thermodynamic cycles and quantum heat engines. *Phys. Rev. E* **76**, 031105 (2007).
44. Abah, O. et al. Single-ion heat engine at maximum power. *Phys. Rev. Lett.* **109**, 203006 (2012).
45. Allahverdyan, A. E., Hovhannisyan, K. V., Melkikh, A. V. & Gevorgian, S. G. Carnot cycle at finite power: attainability of maximal efficiency. *Phys. Rev. Lett.* **111**, 050601 (2013).
46. Zhang, K., Bariani, F. & Meystre, P. Quantum optomechanical heat engine. *Phys. Rev. Lett.* **112**, 150602 (2014).
47. Campisi, M. & Fazio, R. The power of a critical heat engine. *Nat. Commun.* **7**, 11895 (2016).
48. Karimi, B. & Pekola, J. P. Otto refrigerator based on a superconducting qubit: classical and quantum performance. *Phys. Rev. B* **94**, 184503 (2016).
49. Kosloff, R. & Rezek, Y. The quantum harmonic Otto cycle. *Entropy* **19**, 136 (2017).
50. Watanabe, G., Venkatesh, B. P., Talkner, P. & del Campo, A. Quantum performance of thermal machines over many cycles. *Phys. Rev. Lett.* **118**, 050601 (2017).
51. Deffner, S. Efficiency of harmonic quantum Otto engines at maximal power. *Entropy* **20**, 875 (2018).
52. Gelbwaser-Klimovsky, D. et al. Single-atom heat machines enabled by energy quantization. *Phys. Rev. Lett.* **120**, 170601 (2018).
53. Chen, J., Sun, C. & Dong, H. Boosting the performance of quantum Otto heat engines. *Phys. Rev. E* **100**, 032144 (2019).
54. Pekola, J. P., Karimi, B., Thomas, G. & Averin, D. V. Supremacy of incoherent sudden cycles. *Phys. Rev. B* **100**, 085405 (2019).
55. Das, A. & Mukherjee, V. Quantum-enhanced finite-time Otto cycle. *Phys. Rev. B* **2**, 033083 (2020).
56. Berry, M. V. Transitionless quantum driving. *J. Phys. A: Math. Theor.* **42**, 365303 (2009).
57. Deng, J., Wang, Q.-h., Liu, Z., Hänggi, P. & Gong, J. Boosting work characteristics and overall heat-engine performance via shortcuts to adiabaticity: quantum and classical systems. *Phys. Rev. E* **88**, 062122 (2013).
58. Torrontegui, E. et al. Shortcuts to adiabaticity. *Adv. At., Mol., Opt. Phys.* **62**, 117 (2013).
59. del Campo, A., Gould, J. & Paternostro, M. More bang for your buck: superadiabatic quantum engines. *Sci. Rep.* **4**, 6208 (2014).
60. Çakmak, B. & Müstecaplıoğlu, O. E. Spin quantum heat engines with shortcuts to adiabaticity. *Phys. Rev. E* **99**, 032108 (2019).
61. Deng, S. et al. Superadiabatic quantum friction suppression in finite-time thermodynamics. *Sci. Adv.* **18**, eaar5909 (2018).
62. Funo, K. et al. Speeding up a quantum refrigerator via counterdiabatic driving. *Phys. Rev. B* **100**, 035407 (2019).
63. Villazon, T., Polkovnikov, A. & Chandran, A. Swift heat transfer by fast-forward driving in open quantum systems. *Phys. Rev. A* **100**, 012126 (2019).
64. Cavina, V., Mari, A., Carlini, A. & Giovannetti, V. Optimal thermodynamic control in open quantum systems. *Phys. Rev. A* **98**, 012139 (2018).
65. Suri, N., Binder, F. C., Muralidharan, B. & Vinjanampathy, S. Speeding up thermalisation via open quantum system variational optimisation. *Eur. Phys. J. Spec. Top.* **227**, 203 (2018).
66. Menczel, P., Pyhäranta, T., Flindt, C. & Brandner, K. Two-stroke optimization scheme for mesoscopic refrigerators. *Phys. Rev. B* **99**, 224306 (2019).
67. Scully, M. O., Chapin, K. R., Dorfman, K. E., Kim, M. B. & Svidzinsky, A. Quantum heat engine power can be increased by noise-induced coherence. *Proc. Natl. Acad. Sci. USA* **108**, 15097 (2011).
68. Uzdin, R., Levy, A. & Kosloff, R. Equivalence of quantum heat machines, and quantum-thermodynamic signatures. *Phys. Rev. X* **5**, 031044 (2015).
69. Jaramillo, J., Beau, M. & del Campo, A. Quantum supremacy of many-particle thermal machines. *New J. Phys.* **18**, 075019 (2016).
70. Brandner, K., Bauer, M. & Seifert, U. Universal coherence-induced power losses of quantum heat engines in linear response. *Phys. Rev. Lett.* **119**, 170602 (2017).
71. Kosloff, R. & Feldmann, T. Discrete four-stroke quantum heat engine exploring the origin of friction. *Phys. Rev. E* **65**, 055102 (2002).
72. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction* (MIT press, 2018).
73. Haarnoja, T., Zhou, A., Abbeel, P. & Levine, S. In *International Conference on Machine Learning* Vol. 80 (PMLR, 2018).
74. Haarnoja, T. et al. Soft actor-critic algorithms and applications. Preprint at <https://arxiv.org/abs/1812.05905> (2018).
75. Christodoulou, P. Soft actor-critic for discrete action settings. Preprint at <https://arxiv.org/abs/1910.07207> (2019).
76. Delalleau, O., Peter, M., Alonso, E. & Logut, A. Discrete and continuous action representation for practical RL in video games. Preprint at <https://arxiv.org/abs/1912.11077> (2019).
77. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529 (2015).
78. Vinyals, O. et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575**, 350 (2019).

79. Silver, D. et al. Mastering the game of go without human knowledge. *Nature* **550**, 354 (2017).
80. Haarnoja, T. et al. Learning to walk via deep reinforcement learning. Preprint at <https://arxiv.org/abs/1812.11103> (2018c).
81. Bukov, M. et al. Reinforcement learning in different phases of quantum control. *Phys. Rev. X* **8**, 031086 (2018).
82. An, Z. & Zhou, D. Deep reinforcement learning for quantum gate control. *EPL* **126**, 60002 (2019).
83. Dalgaard, M., Motzoi, F., Sørensen, J. J. & Sherson, J. Global optimization of quantum dynamics with AlphaZero deep exploration. *NPJ Quantum Inf.* **6**, 6 (2020).
84. Mackeprang, J., Dasari, D. B. R. & Wrachtrup, J. A reinforcement learning approach for quantum state engineering. *Quantum Mach. Intell.* **2**, 5 (2020).
85. Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *NPJ Quantum Inf.* **5**, 33 (2019).
86. Zhang, X.-M., Wei, Z., Asad, R., Yang, X.-C. & Wang, X. When does reinforcement learning stand out in quantum control? a comparative study on state preparation. *NPJ Quantum Inf.* **5**, 85 (2019).
87. Sgroi, P., Palma, G. M. & Paternostro, M. Reinforcement learning approach to nonequilibrium quantum thermodynamics. *Phys. Rev. Lett.* **126**, 020601 (2021).
88. Sweke, R., Kesselring, M. S., van Nieuwenburg, E. P. L. & Eisert, J. Reinforcement learning decoders for fault-tolerant quantum computation. *Mach. Learn.: Sci. Technol.* **2**, 025005 (2020).
89. Luiz, F. S., de Oliveira Junior, A., Fanchini, F. F. & Landi, G. T. Machine classification for probe based quantum thermometry. Preprint at <https://arxiv.org/abs/2107.04555> (2021).
90. Erdman, P. A., Cavina, V., Fazio, R., Taddei, F. & Giovannetti, V. Maximum power and corresponding efficiency for two-level heat engines and refrigerators: optimality of fast cycles. *New J. Phys.* **21**, 103049 (2019).
91. Lekscha, J., Wilming, H., Eisert, J. & Gallego, R. Quantum thermodynamics with local control. *Phys. Rev. E* **97**, 022142 (2018).
92. Gorini, V., Kossakowski, A. & Sudarshan, E. C. G. Completely positive dynamical semigroups of N level systems. *J. Math. Phys.* **17**, 821 (1976).
93. Lindblad, G. On the generators of quantum dynamical semigroups. *Commun. Math. Phys.* **48**, 119 (1976).
94. Breuer, H. & Petruccione, F. *The theory of open quantum systems* (Oxford University Press, 2002).
95. Yamaguchi, M., Yuge, T. & Ogawa, T. Markovian quantum master equation beyond adiabatic regime. *Phys. Rev. E* **95**, 012136 (2017).
96. Kingma, D. P. & Ba, J. Adam: a method for stochastic optimization. Preprint at <https://arxiv.org/abs/1412.6980> (2014).
97. Curzon, F. & Ahlborn, B. Efficiency of a Carnot engine at maximum power output. *Am. J. Phys.* **43**, 22 (1975).
98. Schmiedl, T. & Seifert, U. Efficiency at maximum power: an analytically solvable model for stochastic heat engines. *Europhys. Lett.* **81**, 20003 (2007).
99. den Broeck, C. V. Thermodynamic efficiency at maximum power. *Phys. Rev. Lett.* **95**, 190602 (2005).
100. Baumgratz, T., Cramer, M. & Plenio, M. B. Quantifying coherence. *Phys. Rev. Lett.* **113**, 140401 (2014).
101. Gallego, R., Riera, A. & Eisert, J. Thermal machines beyond the weak coupling regime. *New J. Phys.* **16**, 125009 (2014).
102. Gelbwaser-Klimovsky, D. & Aspuru-Guzik, A. Strongly coupled quantum heat machines. *J. Phys. Chem. Lett.* **6**, 3477 (2015).
103. Perarnau-Llobet, M., Wilming, H., Riera, A., Gallego, R. & Eisert, J. Strong coupling corrections in quantum thermodynamics. *Phys. Rev. Lett.* **120**, 120602 (2018).
104. Dann, R., Levy, A. & Kosloff, R. Time-dependent markovian quantum master equation. *Phys. Rev. A* **98**, 052129 (2018).
105. Beenakker, C. W. J. Theory of coulomb-blockade oscillations in the conductance of a quantum dot. *Phys. Rev. B* **44**, 1646 (1991).
106. Esposito, M., Lindenberg, K. & den Broeck, C. V. Thermoelectric efficiency at maximum power in a quantum dot. *Europhys. Lett.* **85**, 60010 (2009).
107. Nazarov, Y. V. & Banter, Y. M. *Quantum Transport* (Cambridge, 2009).
108. Erdman, P. A. et al. Thermoelectric properties of an interacting quantum dot based heat engine. *Phys. Rev. B* **95**, 245432 (2017).
109. Achiam, J. *Spinning Up in Deep Reinforcement Learning* <https://github.com/openai/spinningup> (2018).

ACKNOWLEDGEMENTS

We gratefully acknowledge funding by the BMBF (Berlin Institute for the Foundations of Learning and Data–BIFOLD), the European Research Commission (ERC CoG 772230), and the Berlin Mathematics Center MATH+ (AA1-6, AA2-8).

AUTHOR CONTRIBUTIONS

P.A.E. and F.N. designed the research and method. P.A.E. wrote the computer code and carried out the numerical calculations. P.A.E. and F.N. analyzed the data and wrote the manuscript.

FUNDING

Open Access funding enabled and organized by Projekt DEAL.

COMPETING INTERESTS

P.A.E. and F.N. are authors of a patent application containing aspects of this work (Application to the European Patent Office, file number: 21 191 966.7).

ADDITIONAL INFORMATION

Correspondence and requests for materials should be addressed to Paolo A. Erdman or Frank Noé.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2022