Machine Learning Approach to Optimization Problems in Nonequilibrium Thermodynamics

by

XU, Rongxing

August 2022

A thesis submitted to the

Graduate School of Science and Technology

Keio University

in partial fulfilment of the requirements for the

degree of

Ph.D in Science

Acknowledgments

Thank you to my supervisor, Prof. Keiji Saito, for your patience, guidance, and support. I have benefited greatly from your wealth of knowledge. I am extremely grateful that you took me on as a student and continued to believe and support me over the years.

Thank you to my committee members, Prof. Youhei Fujitani, Prof. Keiya Shirahama, and Prof. Hiroshi Watanabe. Your thoughtful, detailed feedback on my thesis has been critical to me.

Thank you to the financial support and the working chance from the Junior Research Assistant program of RIKEN AIP.

Thank you to Dr. Tomokata Kuwahara for all of the assistance you have provided. I would not have a chance to be one of the members of the mathematical science team of RIKEN AIP.

Thank you to the leader of the mathematical science team, Prof. Kenichi Bannai, for your kind assistance and the presentation chances you have provided.

Thank you to Dr. Tan Van Vu for your brilliant ideas and passionate discussion on our project. I learned a lot from you in our discussions.

Thank you to my parents, Yong Xu and Tinghong Zhou, for your endless support. You have always stood behind me. Thank you for all of your love and for constantly receiving my negative emotions all the time.

Thank you to my girlfriend, Zongzeng Yu, for your tender and constantly accompanying me during the Ph.D. period. I would not be such firm when I was facing the problems I met these years. I am very sorry for not seeing you because of the COVID.

Thank you to my roommates in Okurayama dormitory. I received unforgettable and true friendship because of you. You made me no longer homesick and helped me learn Japanese and deal with daily life stuff. You shared almost all my happiness.

Abstract

This thesis aims to develop systematic methods to solve optimization problems in nonequilibrium thermodynamics by machine learning approaches. Solving optimization problems in nonequilibrium thermodynamics is a vital routine to help us to understand statistical physics. However, some problems such as heat engine controlling and shortcuts between two states become difficult to solve under general conditions. New powerful tools are required. Recently, machine learning as a newly developed technique has shown its wide variety of applications in computer science, medicine, and physics in recent decades. Thanks to its generality and relations with optimization problem, machine learning has the potential to be a systematic method for the aforementioned optimization problems in physics.

In this thesis, we focus on applications of machine learning approaches to the problem of finding shortcuts between two equilibrium states with minimum entropy production and the problem of finding the optimal cycle in microscopic heat engines. For the first topic, we propose a systematic method based on the reinforcement learning to find the optimal protocol for minimizing the total entropy production between two equilibrium states of open systems at the same temperature in a given fixed time. We successfully apply our method to the classical and quantum two-level systems. For the second topic, we propose systematic methods based on the universal approximation theorem of neural networks to find an optimal heat engine cycle with some objective function. We apply the method to heat engines with both slow and fast-driving parameters. Besides, we also apply the genetic algorithm to the slow-driving case under some constraints. Optimal cycles in the classical overdamped Brownian heat engine and single-qubit heat engine are successfully found by our method.

Table of Contents

Acknowledgments **1**

Abstract **2**

Table of Contents **3**

Chapter 1:

Introduction **1**

1.1 Background 1

1.2 Motivation 4

1.3 Structure of the thesis 6

Chapter 2:

Machine learning approaches **7**

2.1 Background 7

2.1.1 Development of machine learning 7

2.1.2 Machine learning and physics 9

2.2 Machine learning approaches 13

2.2.1 Supervised learning 13

2.2.2 Reinforcement learning 18

2.3 Chapter summary 27

Chapter 3:

Finding shortcuts between thermodynamic states with minimum entropy production 28

3.1 Setup . . . 28
3.2 Method . . . 30
3.3 Model: the two-level system . . . 34
3.3.1 The dynamics . . . 34
3.3.2 Analytical solution in the classical case . . . 35
3.4 Results and discussions . . . 39
3.4.1 Classical case . . . 39
3.4.2 Quantum case . . . 41
3.5 Chapter summary . . . 43

Chapter 4:

Finding Optimal Cycles in Microscopic Heat Engines 48

4.1 Heat engine with driving . . . 48
4.2 Main goals . . . 53
4.3 Models . . . 54
4.3.1 Classical overdamped Brownian particle heat engine . . . 54
4.3.2 Single-qubit heat engine . . . 55
4.4 Method: genetic algorithm . . . 59
4.4.1 Parametrization . . . 59
4.4.2 Genetic algorithm . . . 62
4.4.3 Example to check the GA . . . 64
4.5 Results and discussions 1 . . . 66
4.5.1 Brownian heat engine . . . 68
4.5.2 Single-qubit heat engine . . . 69
4.6 Method: machine learning . . . 71

4.6.1 Setup for the near quasi-static case 73
4.6.2 Setup for the finite-time case 75
4.7 Results and discussions 2 77
4.7.1 The near quasi-static case 77
4.7.2 The finite-time case 80
4.8 Chapter summary 83

Chapter 5:

Conclusion and Perspectives **84**

Appendix A: Details of the reinforcement learning method for Chapter 2 **86**

Appendix B: Calculation of the upper bound of the final state in the classical two-level system **88**

Appendix C: Adiabatic response theory on classical overdamped Brownian particle heat engine **93**

Appendix D: Adiabatic response theory on single-qubit heat engine **97**

Appendix E: Details for the genetic method **102**

Appendix F: Detail for the machine learning method in Chapter 4 **104**

Bibliography **106**

Chapter1

Introduction

1.1 Background

Thermodynamics is established to deal with fundamental problems of heat, work, and temperature and their relations to energy and entropy of macroscopic systems consisting of many microscopic particles. These relations are well studied and summarized in four laws of thermodynamics. Remarkably, the second law of thermodynamics formulates the direction of a natural process, i.e., an adiabatic system will spontaneously relax to the thermodynamic equilibrium where the system has the maximum entropy. Thermodynamic equilibrium is an axiomatic concept, meaning that no macroscopic changes occur in the system. A thermodynamic equilibrium state can be determined via seeking the extremum of a thermodynamic potential which is often specified by the surroundings of the system. Two equilibrium states are continuously connected via reversible processes, along which the system evolves extremely slowly to maintain the thermodynamic equilibrium without dissipation. Extending the conclusion to the cyclic processes, one obtains a fundamental limit on the efficiency of heat-to-work conversion of heat engines and refrigerators, i.e, the Carnot theorem. These thermodynamic properties in equilibrium can be explained from the microscopic view by the equilibrium statistical mechanics, which states that the Boltzmann factor gives the probability of finding a system in any specific microstate.

Though equilibrium thermodynamics and statistical mechanics are well-established for thermodynamic equilibrium, they do not refer to time and thus to dynamics, which leads to various nonequilibrium phenomena. Therefore, a theory to interpret these time-dependent phenomena is required, and hence, nonequilibrium thermodynamics (also known as thermodynamics of irreversible processes) is developed [1–5]. Different from the thermodynamic equilibrium, nonequilibrium systems are generally inhomogeneous. These inhomogeneous in parts will be spontaneously erased, which is the origin of the irreversibility. The irreversible process always accompanies the energy dissipation, transportation currents, fluctuations, and so on. As thermodynamic equilibrium is the ideal concept under the thermodynamic limit, it is important to understand nonequilibrium behavior for the practical guidance [6].

Unlike equilibrium thermodynamics, nonequilibrium thermodynamics is more mathematically complex, and its physical interpretations are more comprehensive and general. Near the equilibrium regime, where the thermodynamic fluxes are linearly dependent on thermodynamic forces, linear irreversible thermodynamics is developed and widely applied to transport phenomena. Beyond this regime, many branches, such as rational thermodynamics, thermodynamics of internal variables, extended irreversible thermodynamics, etc., under different assumptions were proposed to try to explain specific nonequilibrium phenomena [1–5]. In recent decades, fresh approaches have revealed general laws applicable to nonequilibrium systems, including the fluctuation theorem [7], nonequilibrium work theorem [8, 9] which become particularly relevant for microscopic systems with fluctuations and refer to distribution functions of thermodynamic quantities such as heat, work or entropy production. This promotes the development of stochastic thermodynamics, which links classical thermodynamics and kinetics. In this framework, quantities known from classical thermodynamics are defined along individual trajectories where they become accessible to experimental or numerical measurements. Stochastic thermodynamics provides a robust framework in nonequilibrium thermodynamics focusing on the work, heat, and entropy production [10]. However, these theoretical developments still have limitations, leading to the fact that nonequilibrium

thermodynamics needs further development.

In particular, one of the interesting and important topics in nonequilibrium is the finite-time thermodynamic processes. As is well-known, finite-time thermodynamic processes are irreversible, accompanied by dissipation. How to control or balance the relationship between dissipation and the duration time of the process is of great importance for practical applications. This is also the equivalent question in the trade-off relation between power and efficiency in heat engines. This question inspires the development of finite-time thermodynamics from 1970s [11, 12]. Combining with stochastic thermodynamics, a series of studies discovered quantitative trade-off relations in microscopic heat engines [13–16]. Then optimization problems related to these trade-off relations, such as shortcuts between two states in a finite-time irreversible process and optimal cycles that simultaneously maximizes the power and efficiency of microscopic heat engines, are raised. These optimization problems provide not only an essential practical guide in our daily life but also a better understanding of nonequilibrium thermodynamics. However, current theory and results can only be applied and solved on a few ideal models. The lack of efficient computation methods is still a big problem in nonequilibrium thermodynamics. New mathematical and technical tools are needed.

Recently, the development of machine learning has made it widely apply to various fields, including computer science, biology, and chemistry. Rooted in the statistical learning theory, optimization theory, and optimal control theory, machine learning has many similarities with physics. These similarities open a possible gate to promote each other between machine learning and physics. On the one hand, benefited from its excellent and efficient performance in extracting features, identifying features, and classifying complex data sets [17–19], the applications of machine learning in physics such as solving quantum many-body impurity problems [20, 21],classified phase transitions [22–24], manipulating quantum states, finding physical laws from raw data, etc., have been successful. On the other hand, the lack of a theoretical basis of the deep learning is also a big problem in machine learning. Nevertheless,

existing theories in physics, mainly statistical and quantum theories, offer machine learning the possibility to complete its theories. For example, the Hopfield network is inspired by the Ising model [25], the Boltzmann machine developed under the influence of the Boltzmann distribution [26], and the quantum support vector machine algorithm combined with quantum algorithms [27]. The combination between traditional physics methods and machine learning will surely be the trend and direction of a new era of continued development of physics in the 21st century.

1.2 Motivation

In this background, in this thesis, as the example of combinations of physics and machine learning, we propose two different methods based on machine learning respectively for two kinds of optimization problems in nonequilibrium thermodynamics.

The first problem is the acceleration of the transitions between two thermodynamic states. Accelerating transitions between two states are studied commonly in many physical situations. In pure quantum systems, problems that aim to find fast protocols for transitions between ground states [28–31] are studied, known as shortcuts to adiabaticity. The central idea and technique have been successfully applied and extended in quantum computing [32–34], atomic physics [35–37] and statistical physics, especially improving the performance of microscopic heat engines [38–41]. In thermodynamics, accelerating thermodynamic transformations is also a fundamental problem. It includes the problems of thermalization to obtain an equilibrium distribution from an initial state [42–45], the acceleration of isothermal processes [46–48], and so on. As regularized by the laws of thermodynamics, extra time-dependent perturbation is necessary to accelerate the transitions between thermodynamic states. To this end, additional energy dissipation is inevitably generated [47, 49]. In practice, it is important to understand and control the associated dissipative energy, which can be used to improve the performance of thermodynamic cycles of microscopic heat en-

gines. From these backgrounds, it is highly desired to develop a systematic method to find an optimal protocol that satisfies (i) accelerating the transitions between two states and (ii) minimizing the total dissipation.

The second problem is finding the optimal cycle related to the power-efficiency trade-off relation of microscopic heat engines. Recently, microscopic heat engines have been studied experimentally and theoretically. In experiments, microscopic scale heat engines are proved possible in reality using Brownian particles [50, 51], single-qubit systems [52], and atomic-scale systems [53]. These are crucial to understanding the role of thermal and quantum fluctuations in small systems and the practical control of thermodynamic states. On the theoretical side, one of the important subjects is to figure out the robust relations between thermodynamic efficiency and output power. Universal aspects of the efficiency at the maximum power [54–56] are important in fundamental physics and practical for our daily life. Motivated by many realistic heat engines, the universal relations between efficiency and power have been discussed for classical systems [57], linear response regimes [16], and near quasi-static cycle [58]. To further understand heat engines, another crucial problem is a methodology for controlling or optimizing thermodynamic states. Schmidle and Seifert have developed a method to find an optimization protocol to transform one particle distribution into another [59]. Later, the operating time required to achieve desired distribution has been intensively studied in terms of the speed limit [60, 61]. However, a systematic method to find an optimal cycle in a heat engine has not been seriously discussed. The problem is whether there is a systematic approach to finding a heat engine cycle to optimize some target function within a fixed available parameter region. Difficulties in this problem lie in that parameters in the heat engine cycles are continuous, and the cycles are usually restricted by the condition that no intersects are allowed. The analytical approach is complicated even for the most simple model. Hence, we aim to propose a systematic numerical approach to finding the optimal cycle of the heat engine. We focus on heat engines with external parameters driving near the quasi-static and finite-time case.

1.3 Structure of the thesis

The structure of the thesis is as follows.

In Chapter 2, we look back at the development of the machine learning approaches and their relationships and applications to physical problems. We also review the basic machine learning approaches, including supervised learning and reinforcement learning (RL).

In Chapter 3, we turn to solve the problem of finding the shortcuts between two thermodynamic states with minimum entropy production. We first illustrate our basic setup and the main purpose. Then, the application of the reinforcement learning approach is illustrated in detail to our problem. We then explain the dynamics of the two-level system and provide the theoretical analysis of our problem in the classical case. Finally, the results of our RL method are discussed.

In Chapter 4, we first briefly review the theory of heat engines with driving parameters and illustrated our main purposes based on this framework. We then illustrate the numerical method based on the genetic algorithm and its application to the near quasi-static case. The numerical results are discussed then. For the finite-time case, we propose a general framework based on the machine learning techniques to overcome some drawbacks of the method based on the genetic algorithm.

In Chapter 5, we summarize this thesis's major contributions and discuss our method's drawbacks. We then discuss the future works.

Chapter2

Machine learning approaches

2.1 Background

Machine learning (ML) focuses on the problems of how to build a computer to complete tasks through experience. It has now developed into a comprehensive theory of various disciplines such as statistics, neurobiology, computer graphics, and optimization theory, covering almost all aspects. One of the important goals is to understand and extract the essence of human intelligence and implement it into machines. The accuracy, efficiency, and versatility in processing big data make ML widely used in various fields. In this section, we introduce the development of machine learning and then briefly describe the latest progress in applying machine learning to physics.

2.1.1 Development of machine learning

The concept of ML was born from Artificial Intelligence (AI), which implies an intelligent agent interacting with its environment to learn and take actions to maximize its chance of achieving its goals. In AI, ML tends to develop computer algorithms that learn and improve its performance from experiences for some tasks. The formal appearance of AI was from the workshop at Dartmouth College in 1956, while the concept of ML was first introduced in 1959 by the American scientist Arthur Samuel [62].

Before the formal concept was proposed, research on ML had been studied. In 1949, Hebb proposed a general ML method called perceptron based on the idea of neurology where the brain is constructed by a series of connecting neurons, trying to find the relationship between nodes and nodes in the network through this theory [63]. This rough idea is the origin of the neural networks now. Then, Rosenblatt improved Hebb's perceptron model in 1957, making this concept widely accepted. Rosenblatt believed that a perceptron can be a simple model for briefly describing the basic properties of intelligent objects [64], which is the first artificial neural network (ANN). In 1960, Widrow applied gradient optimization to training perceptrons, namely the least-squares method. Widrow's perceptron is similar in structure and function to Rosenblatt's but has a better approach for linearly separable problems [65]. Later, Minsky formally proposed the famous linear inseparability problem and thought that the neural network approach would not work on this problem [66]. With the unavoidable difficulty of linear inseparability problem of the perceptron and limitations on computers' calculation ability, the field of ML almost stopped for a decade.

Until around 1980, Rumelhart, Hinton, and others independently applied the back-propagation (BP) algorithm to the multi-layer perceptron, which solved Minsky's linear inseparability problem [67]. New algorithms and computer developments have also removed many obstacles to developing experiments in ML, which helps ML be reborn. Then, a variety of machine learning algorithms are starting to emerge. The famous iterative dichotomiser 3 (ID3) algorithm was proposed in this period, which vigorously promoted the research of decision tree algorithm [68]. In 1995, Vapnik et al. proposed the well-known Support Vector Machine (SVM). Benefited from the complete convex optimization theory, generalization boundary theory, and flexible selection of kernel functions, SVM performed better and more efficiently than neural networks in many problems [69]. Also, a robust neural network called long short-term memory (LSTM) architecture was proposed in 1997, which aimed to deal with the problems of time series prediction [70]. Around 2000, ANN and SVM each divided half of the field of ML. Then, the introduction of ensemble learning algorithms, such as

Boosting algorithm [71], random forest algorithm [72], and so on, also injected new vitality into ML.

In addition, computer performance was further improved during this period. In 2006, Hinton et al. proposed the concept of deep learning. He proposed that through layer-by-layer learning, the difficulty of initial value selection in the process of multi-hidden layer network training could be solved so that it has a better effect on classification problems [73]. The emergence of deep learning networks has rapidly promoted the development of machine learning and its applications. In 2016, the AlphaGo artificial neural network designed and developed by Google based on reinforcement learning and Monte-Carlo tree-searching defeated the famous Go master, proving that artificial intelligence is possible [74]. Also, ML shows its power in biology. In 2018, a new deep learning architecture that can predict the protein structure was proposed [75]. Today, many efficient ML approaches have been applied to many fields such as medical diagnosis, face recognition, autonomous driving, natural language translation, etc.

However, although deep learning has been proven to be an effective tool in the application, the reasons behind it are still unclear theoretically. For example, why is the deeper the hidden layer of the deep neural network, the better? What kind of structure can have better learning efficiency and learning effect? The back-propagation algorithm is contrary to neurobiological research. Does it mean that the back-propagation algorithm has its flaws? What is the better way to do this instead? These problems are still to be solved. This thesis does not focus on these unsolved fundamental ML problems. However, we try to connect current ML approaches and problems in physics, especially in statistical physics.

2.1.2 Machine learning and physics

It is well-known that modern ML approaches are developed on the basis statistics and optimization theory. With this background, we believe that there will be many interesting new findings when combining ML approaches and physics. Generally speaking, the intersection

of ML and physics can be roughly divided into the following two types: the first is to use the concepts and theories of physics to understand the ML algorithm itself and develop new ML algorithms; the second is to use existing ML algorithms to deal with physical problems.

Similarities between ML and physics, especially statistical physics and condensed matter theory provide ready-made theories and models for ML, such as tensor networks, mean-field theory, Monte-Carlo method, quantum computing, etc. As early as 1982, using ideas from physics to improve and create new ML algorithms has been noticed. At that time, Hopfield built a recurrent neural network that borrowed the idea of the Ising model. Analogous to the spin state of the Ising model, he designed the state of the neuron as a binary state and naturally derived the energy function of the network consistent with the Hamiltonian of the Ising model. This work cleverly combined statistical physics theory and machine learning methods. The Hopfield network provides a model for understanding human memory [25]. Then in 1985, based on the Hopfield network, by introducing a statistical distribution function in the changes of neuron states, Hinton et al. designed the famous Boltzmann machine [26]. As a powerful unsupervised learning method, it is widely used in feature learning, dimensionality reduction analysis, etc. With the development of deep learning in the following decades, researchers are curious to ask why deep learning methods can have such good performance? Currently, the deep learning algorithm is regarded as a black box for people. People are trying to uncover the secrets of deep learning methods from the perspective of physics. In 2014, Mehta et al. proposed that the renormalization group method in statistical physics is related to the fully connected neural network algorithm; they believe that both ML and renormalization methods can be regarded as coarse-grained data to filter out redundant information to extract important features [76].

Furthermore, as the concept of the quantum computer and quantum algorithm is proposed and realized, quantum mechanics' influence can also help design various new ML algorithms. In general, the improvement of ML by quantum mechanics can be mainly divided into two aspects. One is that quantum computing devices can perform better and

faster calculations of ML algorithms than ordinary computers; the other is to use quantum mechanics theory to improve ML algorithms. As we introduced before, the ultimate goal of ML is to predict the future by using existing experiences, which means that we have to encounter vast amounts of data. Quantum mechanics can provide these storing and transmitting ways of this information. For example, Anguita et al. successfully applied quantum computing to the SVM algorithm [69]. Then, Rebentrost et al. proposed the quantum SVM algorithm, which is optimized quickly in the training and testing speed of the SVM than its traditional algorithm. They also found that the quantum ML algorithm can provide a faster method than the traditional k-means clustering method in cluster problems [77]. Quantum mechanics can accelerate the calculation in ML problems with big data, but it also helps a lot in optimization problems. Based on this, the quantum adiabatic algorithm is derived. Denchev successfully applied quantum annealing to the improvement of the Boosting algorithm [78]. Dulny and Kim et al. tested a similar approach on natural language processing, linearly separable problems [79]. Besides, the tensor networks, widely used in quantum circuits, quantum protocols, and renormalization, can also be applied to ML. For example, Anandkumar studied the problem of parameter estimation in latent variable models such as the Gaussian Mixture model, Hidden Markov model, Hidden Dirichlet distribution model, etc., and found the tensor network structure in their low-order moments [80]. In short, physical theory can provide ML with mature research models, methods, and ideas from another different viewpoint, which play important roles in understanding and improving ML algorithms. Currently, the intrinsic connection between ML algorithms and physical theory is still unclear, which is also one of future research focuses.

On the other hand, the efficiency, accuracy, and stability of ML in dealing with big data problems can also be used to solve physical problems. In the field of materials, Brockherde et al. used ML methods to approximate the density functional. This work bypassed the traditional Kohn-Sham equation approach and also generalized from one-dimensional systems to three-dimensional systems. ML has also been attempted to deal with quantum

many-body problems [81]. The work of Carleo et al. in Science attempts to encode the high-dimensional many-body problem wave function into a neural network. By optimizing the neural network, the ground state of the system can be found efficiently [82]. This work successfully avoided the calculation obstacles to finding the ground state of a quantum many-body system, promoting the development of research in quantum many-body systems. Besides, applying deep learning techniques, especially convolutional neural networks in image recognition, has brought new ideas for phase transition recognition in statistical physics. Wang et al. bypassed complex physical theories, directly started from the data, used the principal component analysis method, and found that the high-temperature phase and low-temperature phase of the Ising model phase transition can be directly separated. They also found that the dimensionality reduction projection in ML is consistent with the calculation of the order parameter for solving the Ising model [24]. The two approaches have created a precedent for using only observed data to study physics, which is an idea that has not appeared in the field of physics itself before. Then, the applications of ML on phase transitions are attractive. One of them is the work of Carrasquilla et al. which applied a supervised learning method using single hidden layer neural networks and convolutional neural networks for the Ising model, which reflects the powerful feature extraction ability of neural network [22]. The other one starts from unsupervised learning and believes that as long as the phase transition system under study has decisive distinguishing characteristics of the two phases, the two phases can be clustered by combining the ideas of neural networks and error labels to predict the phase transition point accurately. The success of this method in the topological phase transition model and the classical Ising phase transition model provides a new solution for some problems that are difficult to determine the phase transition point [23]. Recently, inspired by the successful examples in phase transition problems, a more abstract method called physics informed neural networks (PINN) has been proposed. Generally speaking, PINN is one kind of architecture to solve general nonlinear partial differential equations automatically. Depending on different physical problems, PINN can be trained to solve the

problem by providing the laws of physics described by differential equations. Because it can summarize the rule from raw data, PINN has witnessed much attraction recently [83]. While supervised learning techniques are widely applied to physics problems with big data, reinforcement learning approaches are powerful in dealing with control problems. Nowadays, using reinforcement learning to find the optimal protocol in quantum systems or to manipulate the state to the designated state has become more and more popular in physics research [84, 85].

In summary, the mutual penetration and integration of ML and physical ideas will inevitably greatly promote the development of both parties. This is also an emerging research topic nowadays.

2.2 Machine learning approaches

From the mainstream viewpoint, ML approaches are mainly divided into three main fields: supervised learning (SL), unsupervised learning (USL), and reinforcement learning (RL) (see Figure 2.1). Specifically, the primary mechanism of RL is different from SL and USL. In this thesis, we only focus on the SL and RL approaches.

2.2.1 Supervised learning

2.2.1.1 General framework

We usually use knowledge and conditions to summarize and make assumptions and predictions about unknown problems in scientific research. SL is designed to use this naive fact.

Specifically, in SL, a ML algorithm is proposed to learn a function $g : X \to Y$ which maps an feature vector x_i in the input space X to an label vector y_i in the output space Y by the given training set $\{(x_1, y_1), (x_2, y_2), ..., (x_i, y_i), ..., (x_N, y_N)\}$. All possible functions of g constructs the space G, called hypothesis space. Statistical learning theory establishes

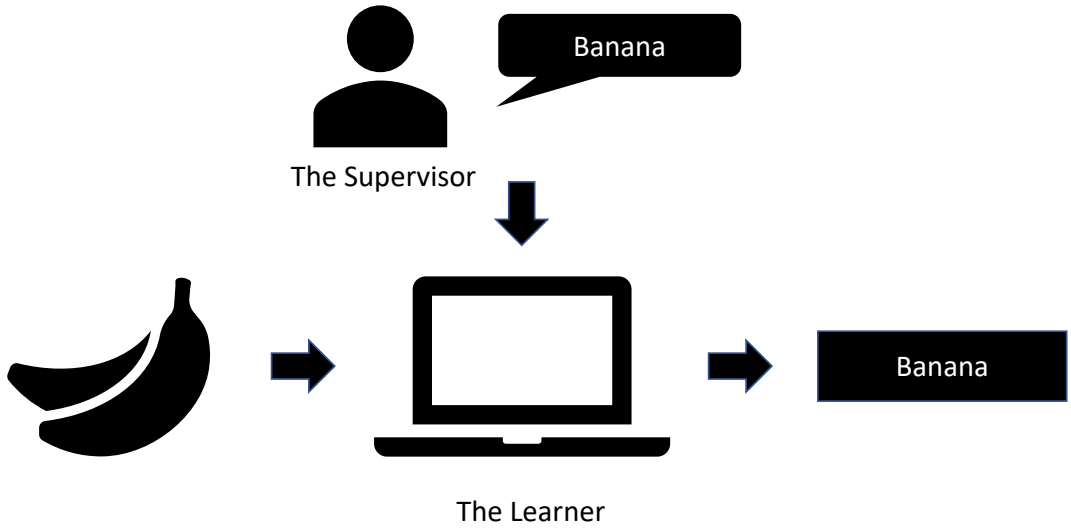

Figure 2.1: The basic structure of supervised learning. The learner learns this data and gives an output. In this process, if there is a supervisor with the external prior knowledge to guide the learning process, it is supervised learning. Without such guidance, the learner learns knowledge from the input samples by itself. This is unsupervised learning in this case.
.

the connection between the SL problem and the probability model. The problem of learning function g can be transformed to learn the conditional probability of y given x, $P(y|x)$, using the finite-number training set. In this case, the function $g(x)$ takes the form of the conditional probability $P(y|x)$.

To evaluate how well a function g in G fits the training set, a non-negative function $l : Y \times Y \rightarrow \mathbb{R}^{\geq 0}$, called a loss function, is needed to be defined, which measures the difference between the given output y_i and the predicted output $\hat{y}_i$. Then, by minimizing the averaged loss over the training set, i.e., empirical risk

$$R_{\text{emp}}(g) = \frac{1}{N} \sum_{i=1}^{N} l(y_i, g(x_i)) \,, \tag{2.1}$$

one can find the best g in the hypothesis space G. However, in most cases, when there

are many functions or the training set is not enough, minimizing $R_{\text{emp}}(g)$ results in poor generalization. The learning machine tends to memorize the data instead of finding the correct map, which leads to overfitting.

Hence, structural risk R_{struc} is introduced to avoid overfitting. Structural risk is often some regularization penalty function, which follows the Occam's Razor rule: keeping the simplest function and discarding complicated functions. The regularization method can avoid overfitting by penalizing the regression coefficients of high value. By minimizing $R_{\text{emp}}(g)$ and R_{struc} simultaneously, the optimal g can be founded. We also call the risk-minimizing process using training data the training process.

After training, the learning machine can predict the label $\hat{y}_i$ according to the corresponding feature x_i. One can prepare a testing data set similar to the training set, but the labels cannot be provided to the learning machine next. According to the loss function among the testing set, the training performance can be evaluated. Generally, choosing different ML algorithms can improve the performance for different problems. Multi-layer ANNs are considered a general framework for various problems because of their high efficiency and accuracy. In this thesis, we apply ANNs as our framework to deal with our problems.

2.2.1.2 Artificial neural networks

The idea of the artificial neural network is taken from the principle of activation and inhibition of neurons in biology. As shown in Figure 2.2, a neuron can be regarded as an information processing unit that processes multiple-input information and then gives a single output. For a single neuron, the processing of the information of input $\boldsymbol{x}$ is mainly carried out through the linear transformation by the weight matrix $\boldsymbol{w}$ and the bias $\boldsymbol{b}$ and the nonlinear transformation by the nonlinear activation function f:

$$\begin{aligned} \boldsymbol{h} &= \boldsymbol{w}\boldsymbol{x} + \boldsymbol{b}\,, \\ \hat{\boldsymbol{y}} &= f(\boldsymbol{h})\,. \end{aligned} \tag{2.2}$$

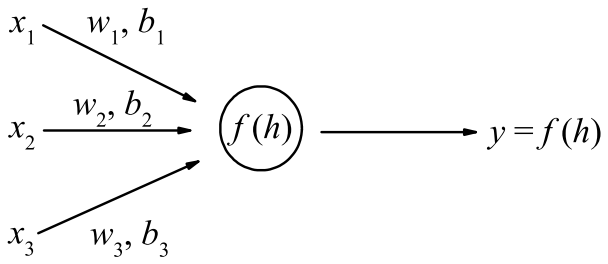

Figure 2.2: The schematic graph for a single neuron.

Here, $\boldsymbol{w}$ and $\boldsymbol{b}$ are real-numbered. In addition, the activation function f is often chosen to be nonlinear to deal with nonlinear separable problems. Practically, activation functions can strongly affect the performance of the learning algorithm. Thus, choosing the most suitable activation function is one of the most critical and challenging tasks when designing an SL algorithm. Here are several widely-applied candidates:

$$\begin{aligned} \text{Rectified Linear Unit (ReLU)}: &\quad f(x) = \max(0, x); \\ \text{Logistic function}: &\quad f(x) = \frac{1}{1+e^{-x}}; \\ \text{Hyperbolic tangent function}: &\quad f(x) = \tanh(x); \\ \text{Snake function}: &\quad f(x) = x + \frac{\sin^2 ax}{x} \ (a > 0)\,. \end{aligned} \tag{2.3}$$

We call this kind of single-neuron network perceptron. By optimizing its weights $\boldsymbol{w}$ and biases $\boldsymbol{b}$, a perceptron can be trained to accomplish some simple learning goals.

As the computation power of one single perceptron is quite limited for many practical problems, multi-layer neural networks, consisting of multiple layers of perceptrons connected in a feed-forward way (also known as feed-forward neural networks), are constructed in general (see Figure 2.3). The universal approximation theorem ensures that any continuous functions that map some input real numbers to some output real numbers can be approx-

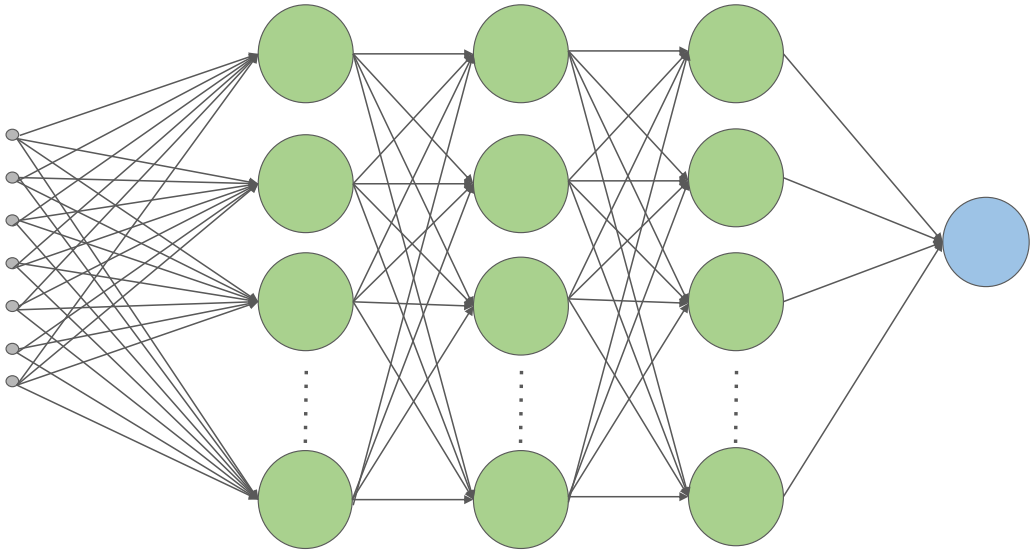

Figure 2.3: The framework of a multi-layer neural networks.

imated closely by arbitrary multi-layer neural networks [86, 87]. Because of this general property, multi-layer neural networks have become the most popular framework for realizing ML algorithms.

Next, to optimize parameters in an ANN, we need to i) define a loss function and ii) find an efficient algorithm to minimize the loss function. For convenience, we respectively denote $\boldsymbol{w}$ and $\boldsymbol{b}$ to the weights and biases for an arbitrary ANN. As we mentioned in the preceding section, the risk function $J(\boldsymbol{w}, \boldsymbol{b})$ for an ANN consists of the empirical risk and structural risk: $J(\boldsymbol{w}, \boldsymbol{b}) = R_{\text{emp}}(\boldsymbol{w}, \boldsymbol{b}) + R_{\text{struc}}(\boldsymbol{w}, \boldsymbol{b})$. Here, the empirical risk is often chosen to be the quadratic function, while the structural risk is generally the L^2-regularization:

$$J(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{2N} \sum_{i=1}^{N} (y_i - g(x_i, \boldsymbol{w}, \boldsymbol{b}))^2 + \frac{\zeta}{2N} \|\boldsymbol{w}\|^2 . \tag{2.4}$$

Here $\|\circ\|$ denotes the L^2 norm (Frobenius norm), and the non-negative constant ζ controls the intensity of the regularization. The quadratic loss function is chosen because of its nice analytical property: continuous, smooth and differentiated at every point. For the structural risk, under the assumption that an ANN has less weights makes simpler models, reducing the

values of weights matrices $\boldsymbol{w}$ can avoid overfitting. Hence, L^2 regularization is introduced. When $J(\boldsymbol{w}, \boldsymbol{b})$ is well-defined, designing a SL algorithm is equivalent to the optimization problem: finding the optimal $\boldsymbol{w}$ and $\boldsymbol{b}$ to minimize $J(\boldsymbol{w}, \boldsymbol{b})$.

To this end, numerical optimization algorithms, including random search algorithms such as genetic algorithms and gradient-oriented algorithms such as Newton's method, can be applied to solve such an optimization problem. Considering the calculating efficiency, in machine learning, gradient-oriented, i.e., gradient descent methods are widely applied. Specifically, for the feed-forward NN architecture, the BP algorithm is believed to be the most popular and efficient method to minimize the risk function.

Furthermore, one can also notice that ANNs are not only applied to design an SL algorithm but also it is a nice approximator of an arbitrary function essentially. According to the requirements of the problem, an ANN can be used to find some optimal function by designing a suitable loss function. In Chapter 3 and Chapter 4, we will show the application of this property of ANNs.

2.2.2 Reinforcement learning

Reinforcement learning (RL) is the third basic machine learning method, which is about learning the optimal behavior in a given environment to maximize a numerical reward signal. Unlike supervised learning, the learner is not under the supervision of some correct answers but must discover behaviors according to the interactions with the environment by trying them. In this section, we review the basic concepts and methods in RL.

2.2.2.1 The problem of RL

The problem of RL borrows from the optimal control theory of incompletely-known Markov decision processes (MDPs), which are a classical framework of sequential decision making. In MDPs, we call the learner or decision-maker the agent which interacts with the environment. When learning, the agent continually interacts with the environment and selects actions.

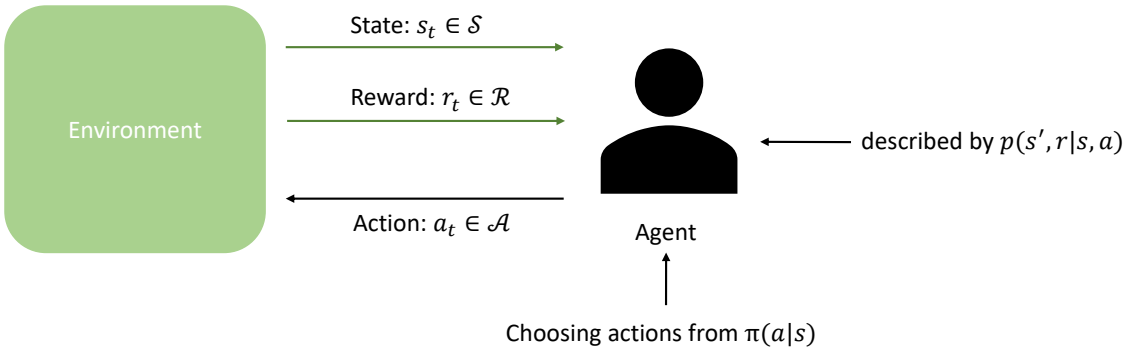

Figure 2.4: The basic structure of reinforcement learning.

The environment responds to these actions, updates to some new environment, and provides reward signals to the agent. Here reward signals are often specific numerical values that the agent aims to maximize by choosing suitable actions. Figure 2.4 shows the basic structure of reinforcement learning.

Specifically, the agent-environment interactions can be divided into a series of discrete steps, $t = 0, 1, 2, ...$ At the j-th step, the agent observes the environment, receives some representation of the environment's state s_j, and selects an action a_j according to s_j. After executing a_j, the agent receives a reward r_j from the environment and interacts with the environment, making the current state transform to another state s_{j+1}. The whole process forms a sequence τ:

$$\tau = \{s_1, a_1, r_1, s_2, a_2, r_2, ...\}. \tag{2.5}$$

All states, actions, and rewards in the sequence construct the sets of states $\mathcal{S}$, actions $\mathcal{A}$ and rewards $\mathcal{R}$ respectively. For the cases where the time is finite, the sets $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ all have a finite number of elements. In finite MDPs, the sequence of the agent-environment interactions is divided into repeated subsequences called episodes. Each episode is terminated at the terminal time and reset to a designated starting state. Besides, the return G_j of at

step j is defined as the sum of the reward from j to the terminal time.

$$G_j \equiv \sum_{k=0}^{N-j} r_{j+k} \,. \tag{2.6}$$

In a MDP, the dynamics of the environment which determines the subsequent state s' and the reward r according to the current state s and action a is described by the probability:

$$p(s', r|s, a) \equiv \Pr\{s_{j+1} = s', r_j = r|s_j = s, a_j = a\} \,, \tag{2.7}$$

for all $s', s \in \mathcal{S}, r \in \mathcal{R}$, and $a \in \mathcal{A}$. Naturally, the function p is normalized for all $s \in \mathcal{S}$, $a \in \mathcal{A}$:

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r|s, a) = 1 \,. \tag{2.8}$$

Moreover, the property of MDP requires that the probability of each possible value of s_{j+1} and r_j only depends on the state and action in the previous step state, i.e., s_j and a_j. The state must include all information about the past interactions between the agent and the environment.

As we illustrated before, the agent's goal is to choose optimal actions to maximize the total reward it receives. Hence, any RL problem follows the following reward hypothesis: all of what we mean by goals and purposes can be well thought of as the maximization of the expected value of the cumulative sum of a received scalar signal (reward). Besides, the agent's rule to choose an action according to the current state is called policy. The policy π maps the states to the probabilities of selecting each possible action:

$$\pi(a|s) \equiv \Pr\{a_j = a|s_j = s\} \,. \tag{2.9}$$

With the preparation above, the problem of RL can be formally defined as: finding the

optimal π^* to maximize the expected discounted return of the starting time, i.e.,

$$\pi^* = \arg\max_{\pi} \mathbb{E}_{\tau\sim\pi,p}\left[\sum_{k=1}^{N} r_k\right] . \tag{2.10}$$

Here $\mathbb{E}_{\tau\sim\pi,p}[\cdot]$ denotes the expected value of a random trajectory given the policy π and the dynamics of the environment p.

Under the framework of MDP, many goal-directed problems can be well-solved in flexible ways. For example, states can be determined to try to satisfy the Markov property. Actions can be not only scalars but also multi-dimensional vectors. Also, combined with ANNs, the forms of states, actions, and the policy can have various choices.

2.2.2.2 The solution of an RL problem

The methods for solving RL problems are called RL methods. According to the different quantities or functions to optimize, RL methods are divided into two primary methods: the value-based methods and the policy-based methods.

2.2.2.2.1 The Bellman equation Before we introduce both methods, we first review the optimal control theory, which plays an important role when designing a RL method. In optimal control theory, value functions are introduced to describe how good the state or action that the agent stays or selects is in a given state. The state-value function $v_\pi(s)$ for a policy π denotes the expected return starting from state s following π, i.e., are defined by

$$v_\pi(s) \equiv \mathbb{E}_{\tau\sim\pi,p}\left[G_j | s_j = s\right] , \text{ for all } s \in \mathcal{S} . \tag{2.11}$$

The action-value function $q_\pi(s,a)$ represents the expected return starting from s, taking the action a and following π:

$$q_\pi(s,a) \equiv \mathbb{E}_{\tau\sim\pi,p}\left[G_j | s_j = s, a_j = a\right] , \text{ for all } s \in \mathcal{S} . \tag{2.12}$$

The state-value function can be obtained by averaging all action-value functions of each possible a,

$$v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s)\mathbb{E}_{\tau\sim\pi,p}\left[G_j|s_j = s, a_j = a\right] = \sum_{a\in\mathcal{A}} \pi(a|s)q_\pi(s,a)\,,\ \text{for all } s\in\mathcal{S}\,. \tag{2.13}$$

One can notice a relationship between the value of a state and the values of its successor states in the definition of $v_\pi(s)$, i.e., for an arbitrary π and s, we have

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_{\tau\sim\pi,p}\left[G_j|s_j = s\right] \\ &= \mathbb{E}_{\tau\sim\pi,p}\left[r_j + G_{j+1}|s_j = s\right] \\ &= \sum_a \pi(a|s)\sum_{s'}\sum_r p(s',r|s,a)\left[r + \mathbb{E}_{\tau\sim\pi,p}\left[G_{j+1}|s_{j+1} = s'\right]\right] \\ &= \sum_a \pi(a|s)\sum_{s',r} p(s',r|s,a)\left[r + v_\pi(s')\right]\,. \end{aligned} \tag{2.14}$$

The last equality is the Bellman equation for v_π. Similarly, for q_π, the Bellman equation is

$$q_\pi(s,a) = \sum_{s',r} p(s',r|s,a)\left[r + q_\pi(s',a)\right]\,. \tag{2.15}$$

Value functions can be used to compare two policies. We think that the policy π is better or equal to π', i.e., $\pi \geq \pi'$, if and only if $v_\pi(s) \geq v_{\pi'}(s)$ for all $s \in \mathcal{S}$. The optimal policy is the one that is better or equal to all policies, denoted as π^*. Thus, the corresponding optimal state-value function is defined as,

$$v_*(s) \equiv \max_\pi v_\pi(s)\,,\ \text{for all } s\in\mathcal{S}\,. \tag{2.16}$$

Also, we have the similar definition for optimal action-value function, i.e.,

$$q_*(s,a) \equiv \max_\pi q_\pi(s,a) = \mathbb{E}_{\tau\sim\pi,p}\left[r_j + v_*(s_{j+1})|s_j = s, a_j = a\right]\,,\ \text{for all } s\in\mathcal{S} \text{ and } a\in\mathcal{A}\,. \tag{2.17}$$

Like the Bellman equation for value functions for an arbitrary policy, we also have the similar relationships for the optimal policy, i.e.,

$$\begin{aligned} v_*(s) &= \max_{a\in\mathcal{A}} q_*(s,a) \\ &= \max_{a\in\mathcal{A}} \mathbb{E}_{\tau\sim\pi,p}\left[r_j + v_*(s_{j+1})|s_j = s, a_j = a\right] \\ &= \max_{a\in\mathcal{A}} \sum_{s',r} p(s',r|s,a)\left[r + v_*(s')\right] , \end{aligned} \tag{2.18}$$

and

$$\begin{aligned} q_*(s,a) &= \mathbb{E}_{\tau\sim\pi,p}\left[r_j + \max_{a'} q_*(s_{j+1},a')|s_j = s, a_j = a\right] \\ &= \sum_{s',r} p(s',r|s,a)\left[r + \max_{a'} q_*(s',a')\right] . \end{aligned} \tag{2.19}$$

Eq. (2.18) and Eq. (2.19) are respectively two forms of Bellman optimality equation for $v_*(s)$ and $q_*(s,a)$.

2.2.2.2.2 Value-based method The Bellman optimality equation provides a straightforward way to solve the RL problem. According to the definition of the optimal policy, the policy π is the optimal policy π^* if and only if the state-value function $v_\pi(s')$ of the successor state s' from an arbitrary state s equals to $v_*(s')$. The Bellman optimality equation provides the relationship between the optimal value function of state s and its successive state s'. Hence, the basic structure of a value-based method has the following three steps: i) Calculating the state-value function $v(s)$ for each state $s \in \mathcal{S}$; ii) According to the Bellman optimality equation Eq. (2.18), using the value function $v(s')$ each successor state s' of s to estimate the state-value function of s, $V(s)$; iii) Calculating the difference between $V(s)$ and $v(s)$: $\Delta = |v(s) - V(s)|$. iv) Replacing $v(s)$ by $V(s)$ and repeating step ii) and iii) until Δ

becomes almost zero. Then the optimal policy can be expressed in a deterministic way, i.e.,

$$\pi^*(s) = \arg\max_a \sum_{s',r} p(s', r|s, a) \left[r + V(s') \right] . \tag{2.20}$$

This is also known as dynamic programming in optimal control theory.

However, one can notice that the dynamic programming method requires one has to look ahead at all possibilities of s, a, and r under the condition that the dynamics of the environment ($p(s', r|s, a)$) is wholly known. In practise, because the set of s, a, and r are usually huge, it is complicated to look ahead all of them can calculate the corresponding value functions. Moreover, in many cases, the dynamics of the environment are unknown. These two difficulties make explicitly solving the Bellman optimality equation impractical.

Instead, methods that approximately estimate the value functions in the Bellman equation are considered. The basic idea is to use an incremental method to estimate the expected value by sampling methods. For example, by looking ahead to one episode, the Monte-Carlo method approximates the state-value functions by the returns from the sampling episodes. The exact state-value function is then calculated by the incremental method,

$$V(s_t) \leftarrow V(s_t) + \alpha(G_t - V(s_t)) , \tag{2.21}$$

where α is the incremental factor which controls the convergent speed. More efficiently, one can look ahead only one step and using the return of this step to approximate the state-value function, known as temporal-difference method, i.e.,

$$V(s_t) \leftarrow V(s_t) + \alpha(r_t + V(s_{t+1}) - V(s_t)) . \tag{2.22}$$

Applying Eq. (2.21) and (2.22) to the basic structure of the value-based method, one can solve the RL problem in principle. Practically, to save the computation resources, one can apply ANNs to approximate the value functions by parametrizing $V(s)$ by the parameter $\boldsymbol{w}$:

$V(s, \boldsymbol{w})$. Once the optimal $\boldsymbol{w}$ is obtained, the optimal policy can be determined directly. This generalization makes the RL to be more powerful than traditional optimal control theory.

2.2.2.2.3 Policy-based method Recall that the RL problem (2.10) requires us to find the optimal policy. In value-based methods, we try to learn the values and then select actions based on value functions to obtain the optimal policy. The optimal policy can be obtained more directly without considering the value functions.

If the policy's parameter vector is denoted by $\boldsymbol{\theta} \in \mathbb{R}^{d'}$, the parametrized policy $\pi(a|s, \boldsymbol{\theta})$ represents the probability that the action a is chosen given the state s with parameter $\boldsymbol{\theta}$. As long as the optimal $\boldsymbol{\theta}$ is found, the policy can be optimized. The problem (2.10) provides the way to optimize $\boldsymbol{\theta}$ via maximizing the objective function $L(\boldsymbol{\theta})$. Here, $L(\boldsymbol{\theta})$ is known as the state-value function of the initial state

$$L(\boldsymbol{\theta}) = v_\pi(s_1). \tag{2.23}$$

Thus, the parameter $\boldsymbol{\theta}$ is updated by the gradient ascent method:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \nabla L(\boldsymbol{\theta}). \tag{2.24}$$

According to the definition of $L(\boldsymbol{\theta})$, one can derive the relationship between $\nabla L(\boldsymbol{\theta})$ and

$\pi(a|s,\boldsymbol{\theta})$, known as the policy gradient theorem:

$$\begin{aligned}\nabla L(\boldsymbol{\theta}) &= \nabla v_\pi(s_1) \\ &= \nabla \mathbb{E}_{\tau\sim\pi,p}\left[G_1(\tau)\right] \\ &= \nabla p(\tau,\boldsymbol{\theta})\left[G_1(\tau)\right] \\ &= \sum_\tau G_1(\tau) p(\tau,\boldsymbol{\theta}) \nabla \ln p(\tau,\boldsymbol{\theta}) \\ &= \mathbb{E}_{\tau\sim\pi,p}\left[G_1(\tau)\nabla \ln \prod_{t=1}^{N} \pi(a_t|s_t,\boldsymbol{\theta})) p(s_{t+1}|s_t,a_t)\right] \\ &\propto \mathbb{E}_{\tau\sim\pi,p}\left[G_1(\tau)\sum_t \nabla \ln \pi(a_t|s_t,\boldsymbol{\theta})\right]\end{aligned} \tag{2.25}$$

However, notice that calculating the gradient $\nabla L(\boldsymbol{\theta})$ requires the dynamics of the environment ($p(s',r|s,a)$). It is inefficient to calculate in many cases in practical. Thus, one considers replacing the gradient ascent method by the stochastic ascent method

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\widehat{\nabla L(\boldsymbol{\theta})}\,, \tag{2.26}$$

where $\widehat{\nabla L(\boldsymbol{\theta})}$ is the expectation of the sample gradient to replace the actual gradient. The sampled gradient need only be proportional to the gradient because any constant of proportionality can be absorbed into the learning rate α. Because the policy-based method improves the policy directly by the policy gradient $\pi(a|s,\boldsymbol{\theta})$, it is also called policy gradient method.

Hence, Eq. (2.25) provides a simple structure to solve the RL problem: i) Generate a sequence τ; ii) According to sequence τ, calculate $[G_1(\tau)\sum_t \nabla \ln \pi(a_t|s_t,\boldsymbol{\theta})]$; iii) Update the parameter $\boldsymbol{\theta}$ by $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha G_1(\tau)\sum_t \nabla \ln \pi(a_t|s_t,\boldsymbol{\theta})$; iv) Repeat step i) to iii) until convergence.

Compared with the value-based methods, policy gradient methods provides practical ways to deal with continuous action spaces with an infinite number of actions. This can be

realized by applying parametrized distribution functions over $\mathbb{R}$ to the parametrized policy $\pi(a|s, \boldsymbol{\theta})$. Moreover, by choosing different sampling methods (for example, Monte-Carlo method, the temporal difference (TD) method, etc.) to evaluate the actual gradient, the policy gradient provides a variety of methods to make the computation efficient [88].

2.3 Chapter summary

This chapter first reviewed the development of the machine learning approaches and the combination of physics and machine learning. We then reviewed the basic machine learning approaches, including supervised learning and reinforcement learning. In the section on supervised learning, we briefly introduced the basic learning theory. We then gave a simple structure of the artificial neural networks widely applied in machine learning problems. In the section on reinforcement learning, we reviewed how to construct a reinforcement learning problem from the optimal control theory and how to solve the reinforcement learning problem.

Chapter3

Finding shortcuts between thermodynamic states with minimum entropy production

In this chapter, we aim to reach the equilibrium distribution of some system parameter $\Lambda_{\rm f}$ at the final time $t_{\rm f}$ when the initial state is an equilibrium state with a parameter $\Lambda_{\rm i}$ with the fixed period τ in a dissipative dynamics (See Figure 3.1 as the schematic graph). We do not control the coupling strength between the system and reservoirs since it is usually fixed in standard physical setups. Instead, we control only system parameters: $\Lambda_{\rm i} \to \Lambda_{\rm f}$, so that the total dissipation becomes minimum for the path. We propose the policy gradient method combined with artificial neural networks to achieve the goal [89].

3.1 Setup

The system we consider has a Hamiltonian $H(\Lambda)$. Here, Λ is a set of time-dependent controllable parameters, and these parameters can vary continuously. This system is coupled with the heat bath with the constant temperature T. See Figure 3.1 as a schematic graph.

Then, the following transition process is considered. The system initially stays at some

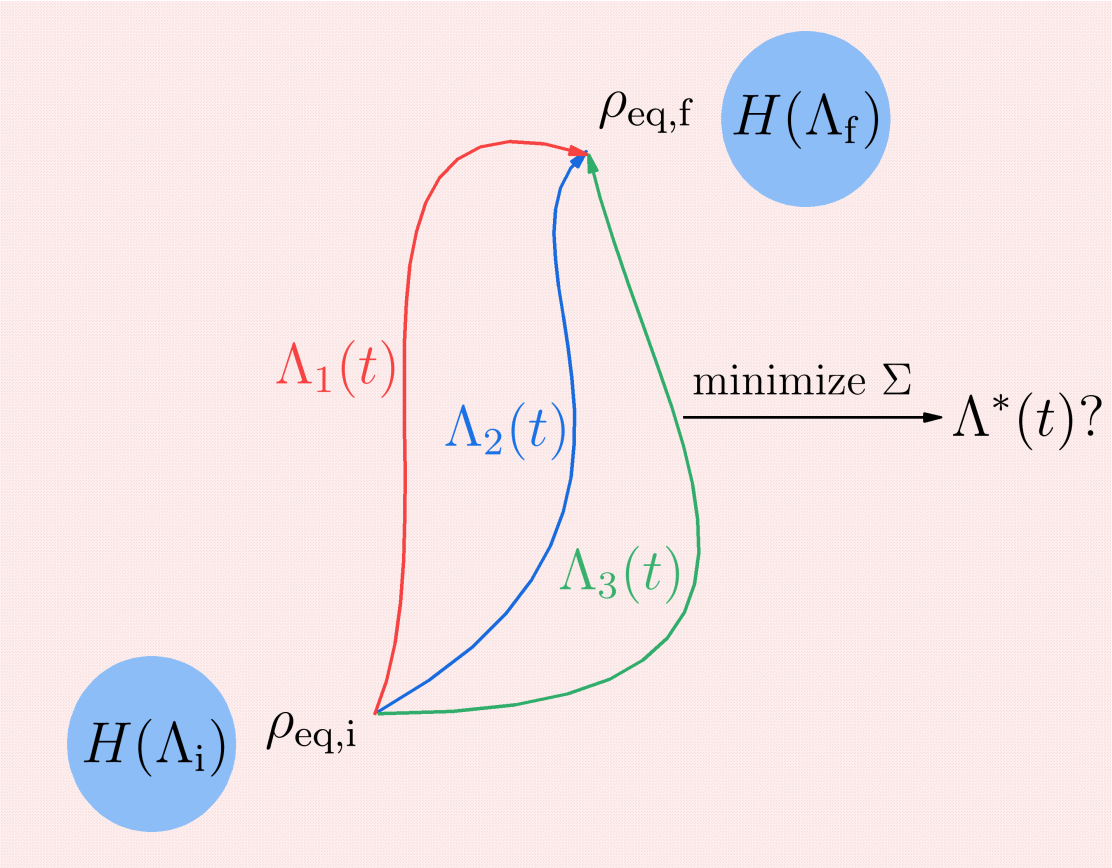

Figure 3.1: The schematic graph of the shortcut between two equilibrium states. The system (blue circle) evolves in a thermal environment (pink background) with a constant temperature T in a given fixed period. An optimal protocol $\Lambda^*(t)$, which minimizes the total entropy production Σ of the system and the environment during the transition between two equilibrium states $\rho_{\mathrm{eq,i}}$ and $\rho_{\mathrm{eq,f}}$, is required to be obtained.

instantaneous equilibrium state $\rho_{\mathrm{eq,i}}$ specified by the parameter $\Lambda(t_{\mathrm{i}})$,

$$\rho_{\mathrm{eq,i}} = \frac{e^{-\beta H(\Lambda_{\mathrm{i}})}}{Z_{\mathrm{i}}}, \quad Z_{\mathrm{i}} = \mathrm{Tr}\left[e^{-\beta H(\Lambda_{\mathrm{i}})}\right], \tag{3.1}$$

where β is the inverse temperature. In all our calculations, the Boltzmann constant is set to be unity. For simplicity, we denote $\Lambda(t_{\mathrm{i}})$ ($\Lambda(t_{\mathrm{f}})$) Λ_{i} (Λ_{f}) and $\rho_{\mathrm{eq}}(\Lambda_{\mathrm{i}})$ ($\rho_{\mathrm{eq}}(\Lambda_{\mathrm{f}})$) $\rho_{\mathrm{eq,i}}$ ($\rho_{\mathrm{eq,f}}$). Then, we control the system by changing the parameter Λ from the initial equilibrium state $\rho_{\mathrm{eq,i}}$ at the time t_{i} to another equilibrium state $\rho_{\mathrm{eq,f}}$ at the time t_{f}. We fix the equilibrium

state and the equilibrium state.

Our main purpose here is to find an optimal path $\Lambda^*(t)$ that minimizes the total entropy production Σ of the system and the environment during the whole transition process, i.e.,

$$\begin{aligned} \Lambda^*(t) &= \arg\min_{\Lambda(t)} \Sigma = \int_{t_\mathrm{i}}^{t_\mathrm{f}} \mathrm{d}t\, \dot{\Sigma} \\ \text{such that} \quad & \Delta d = d(\rho(t_\mathrm{f}), \rho_{\mathrm{eq,f}}) = 0\,, \\ & \tau = t_\mathrm{f} - t_\mathrm{i} : \text{fixed} \end{aligned} \tag{3.2}$$

where $\rho(t_\mathrm{f})$ is the final state at time t_f and $d(\rho_1, \rho_2)$ measures the distance between two different states, where we use trace distance (L_1 norm) for quantum (classical) systems. In general, it is difficult to solve this type of optimization problem exactly. We here employ the reinforcement learning (RL) techniques in subsequent sections.

3.2 Method

In this section, we explain how to apply the RL method to our problem (3.2). Following the basic framework of the RL methods, we first transform our problem to the form of standard RL problem (2.10) and then apply the RL methods to solve the problem.

We assume that there is an agent which can observe the interaction between our system and the thermal environment and control the evolution of the system by changing the parameter $\Lambda(t)$. The control process is executed during the designated fixed time τ ($= t_\mathrm{f} - t_\mathrm{i}$). At time t, the agent observes the system and obtains its state s_t, which is specified by time-dependent density matrix $\rho(t)$ of the system, which makes sense because $\rho(t)$ represents all information of the system at time t. Hence, the set of state $\mathcal{S}$ is constructed by all possible states of the system,

$$\mathcal{S} = \{s_t = \rho(t)\}\,. \tag{3.3}$$

The agent chooses the suitable action a_t given that the state is s_t at time t. Naturally, the

action set $\mathcal{A}$ is defined as the collection of all possible controllable parameter vectors $\Lambda(t)$, i.e.,

$$\mathcal{A} = \{a_t = \Lambda(t)\}\,. \tag{3.4}$$

For our optimization problem (3.2), because the next state is determined once the current state and the action have been determined, the transition is always deterministic. To this end, the dynamics of the environment is totally determined by the equation of motion with respect to the density matrix $\rho(t)$. For specific systems the exact expression of the dynamics is different, and we will discuss this in the next section.

As we illustrated in Chapter 2, designing a suitable quantity as the reward can determine direction of evolution of the MDP and finally affects the performance of the RL method. In our problem (3.2), noticing that we have two major goals, minimization of the entropy production and reaching the final state, need to be simultaneously accomplished, the reward is designed as a linear combination of these two goals as

$$\begin{aligned} &\mathcal{R} = \{r_t = (1-\zeta)r_{1,t} + \zeta r_{2,t}\}\,, \\ &\text{where}\ \ r_{1,t} = -\dot{\Sigma}_j \mathrm{d}t, \ \ \text{and}\ \ r_{2,t} = -\delta_{t_\mathrm{f},t}\Delta d\,. \end{aligned} \tag{3.5}$$

Here, $\delta_{t_\mathrm{f},t}$ is Kronecker delta, and ζ is an arbitrary constant. In our following discussions, we choose ζ to be 0.9 as the optimal parameter. The quantity Δd is a distance between the final state and the target state at time t_f, which is defined in Eq. (3.2).

Now, we can re-state our problem into the RL version: finding the optimal policy π^* to maximize the expected return of the state $s_{t_{,\mathrm{i}}}$, i.e.,

$$\pi^* = \arg\min_{\pi} \mathbb{E}\left[\int_{t_\mathrm{i}}^{t_\mathrm{f}} \mathrm{d}t\, r_t\right], \tag{3.6}$$

where the notation $\mathbb{E}[\cdot]$ represents the expectation over the policy π and the dynamics of the system. For the convenience of numerical calculations, we approximate (3.6) by dividing the period τ into N equal intervals.

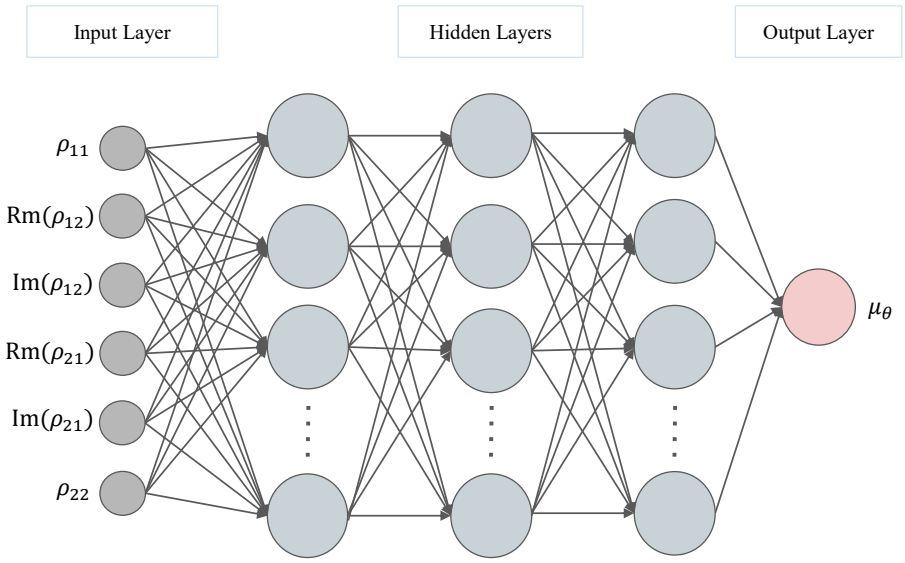

Figure 3.2: The schematic graph of the architecture of the neural networks to approximate $\mu_\theta(s)$.

As the action a_t is continuous and can be any real number for the given state s_t, the policy gradient method is considered as the most suitable method in this case. Hence, we parametrize the policy function $\pi(a_t|s_t)$ with the parameter vector $\boldsymbol{\theta}$ which represents the weights and biases in the neural networks we use, i.e., $\pi(a_t|s_t, \boldsymbol{\theta})$. We choose the following parametrized multivariate normal distribution to be the policy function:

$$\begin{aligned}\pi(a_t|s_t, \boldsymbol{\theta}) =& \frac{1}{(2\pi)^{d/2}|\Sigma_a|^{1/2}} \\ & \times \exp\left[-\frac{1}{2}(a_t - \mu_{\boldsymbol{\theta}}(s_t))^T \Sigma_a^{-1}(a_t - \mu_{\boldsymbol{\theta}}(s_t))\right],\end{aligned} \tag{3.7}$$

where $\boldsymbol{\theta}$ is the parameter vector and Σ_a is the covariance matrix of the action a. Here, the parametrized expectation value of a over the distribution $(\mu_\theta(s))$ is represented by the ANN shown in Figure 3.2. See details in Appendix A.

According to the dimension d of action a, Eq. (3.7) has different forms, which is specified by the model. Then, the aim of the RL is transformed into finding the optimal $\boldsymbol{\theta}$ that can

maximize the total reward.

The policy gradient theorem indicates the update rule [88]:

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha G_j \nabla \ln \pi(a_j|s_j, \boldsymbol{\theta}) \,, \tag{3.8}$$

where the notation G_j is the expected return of the j-th step:

$$G_j = \sum_{k=j}^{N} r_k \quad \text{for} \quad j \in [1, N] \,. \tag{3.9}$$

Combining (3.7), (3.8) and (3.9), we can then derive the update rule of the parameter $\boldsymbol{\theta}$ of this neural network. After enough number of episodes, $\boldsymbol{\theta}$ will be optimized, and the optimal actions can be taken according to (3.7). There are two major reasons to choose the parametrized multivariate normal distribution as our policy. First, one can notice that the log-likelihood function for the policy $\pi(a_j|s_j, \boldsymbol{\theta})$ becomes the quadratic form when the multivariate normal distribution (3.7) is applied. Mathematically, the quadratic log-likelihood function is simpler to deal with because of its symmetry: the error above the target causes the same loss as the same magnitude below the target. Second, the covariance matrix Σ_a of the multivariate normal distribution (3.7) controls the performance of the RL method. If Σ_a is too large, the method is hard to converge because the controllable becomes totally random numbers. However, if Σ_a is too small, our calculation will stuck in some local minimum, because we lost our exploration ability in this case. By choosing a suitable Σ_a, the multivariate normal distribution (3.7) enables us to control multiple parameters simultaneously.

3.3 Model: the two-level system

3.3.1 The dynamics

We use the two-level system coupling with the environment with the constant temperature T as our example model for performing our main purpose. The Hamiltonian of the two-level system is

$$H(\Lambda_t) = \frac{1}{2}\left(\varepsilon_t \sigma_x + \lambda_t \sigma_z\right) , \tag{3.10}$$

where σ_x and σ_z are respectively the x and z components of the Pauli matrices. The x-component of Zeeman term gives the quantum coherence effect for the eigenbasis for the z-component of the Zeeman term. Both ε and λ change continuously in time. Hence, the controllable parameter $\Lambda_t = (\varepsilon_t, \lambda_t)$ has the vector notation. In this thesis, we set $\hbar$ to be unity.

The dynamics of this open system is described by the Gorini-Kossakowski-Lindblad-Sudarshan (GKLS) equation [90, 91], which stands for the dissipative dynamics of the state ρ_t of the system:

$$\partial_t \rho_t = \mathbb{L}_t \rho_t \, , \tag{3.11}$$

$$\begin{aligned} \mathbb{L}_t X = & -i\left[H_{\lambda_t}, X\right] \\ & - \frac{1}{2}\sum_{\sigma=\pm}\left(\left[V_t^{\sigma} X, V_t^{\sigma\dagger}\right] + \left[V_t^{\sigma}, X V_t^{\sigma\dagger}\right]\right) , \end{aligned} \tag{3.12}$$

where $\mathbb{L}_t$ is a time-evolution generator including the dissipative dynamics. The operator V_t^{σ} is the quantum jump operator given by

$$V_t^{\pm} = \sqrt{\frac{1}{1+e^{\mp 2\beta\Omega_t}}}|e_{\mp}\rangle\langle e_{\pm}| \, . \tag{3.13}$$

Here Ω_t $(= \sqrt{\lambda_t^2 + \varepsilon_t^2}/2)$ is the generalized Rabi frequency. The states $|e_+\rangle$ and $|e_-\rangle$ is the

excited and ground state of the corresponding energies Ω_t and $-\Omega_t$, respectively. Note that this expression satisfies the quantum detailed balance condition, which guarantees that the steady-state of the generator $\mathbb{L}_t$ is the instantaneous equilibrium distribution. The quantum master equation Eq. (3.12) is derived under the assumption that the relaxation time in the bath is much faster than the time scale of the system. Besides, we measure the the distance between two states by the trace distance [92]:

$$d(\rho_1, \rho_2) = \mathrm{Tr}[|\rho_1 - \rho_2|]\,. \tag{3.14}$$

Note that the total entropy production rate of the system and the environmentis given by [93]

$$\dot{\Sigma}_t = -\mathrm{Tr}\left[(\partial_t \rho_t)\ln \rho_t\right] + \mathrm{Tr}\left[(\partial_t \rho_t)\ln \rho_{\mathrm{eq},t}\right]\,, \tag{3.15}$$

where $\rho_{\mathrm{eq},t}$ is an instantaneous equilibrium distribution at time t.

3.3.2 Analytical solution in the classical case

While it is generically difficult to analytically solve the optimization problem (3.2) in the general case, one can access the solution analytically when $\varepsilon = 0$, since the system's dynamics are reduced to a classical probabilistic process, and the problem is reduced to a much simpler optimization problem using the calculus of variations.

The classical probabilistic process for this case is simply given as

$$\begin{pmatrix} \dot{p}_- \\ \dot{p}_+ \end{pmatrix} = \begin{pmatrix} -(1-\omega_t) & \omega_t \\ 1-\omega_t & -\omega_t \end{pmatrix} \begin{pmatrix} p_- \\ p_+ \end{pmatrix}\,, \tag{3.16}$$

where p_- and p_+ are the probabilities of the eigenstates $|e_-\rangle$ and $|e_+\rangle$, respectively. The function ω_t here is defined as the instantaneous equilibrium state of $|e_-\rangle$,

$$\omega_t = \frac{1}{e^{-\beta\lambda_t}+1}\,. \tag{3.17}$$

Using the normalization $p_+ = 1 - p_-$, we obtain the following single equation:

$$\dot{p}_- = -p_- + \omega_t \,. \tag{3.18}$$

For the classical probabilistic process obeying $\dot{p}_i = \sum_j W_{ij} p_j$, the entropy production rate can be written in the form [94]

$$\dot{\Sigma} = \frac{1}{2} \sum_{i \neq j} (W_{ji} p_i - W_{ij} p_j) \ln \frac{W_{ji} p_i}{W_{ij} p_j} \,, \tag{3.19}$$

which is equivalent to the expression Eq. (3.15). Using this expression, the entropy production rate can be explicitly written as

$$\dot{\Sigma} = \dot{p}_- \ln \frac{(1 - p_-)(p_- + \dot{p}_-)}{p_-[1 - (p_- + \dot{p}_-)]} \,. \tag{3.20}$$

Then, the standard Euler-Lagrange method can be applied to minimize the entropy production. By defining the Lagrangian $\mathcal{L}(p_-, \dot{p}_-) := \dot{\Sigma}$, we can find the extremum by the Euler-Lagrange equation

$$\frac{\mathrm{d}}{\mathrm{d}t} \left(\frac{\partial \mathcal{L}(p_-, \dot{p}_-)}{\partial \dot{p}_-} \right) - \frac{\partial \mathcal{L}(p_-, \dot{p}_-)}{\partial p_-} = 0 \,. \tag{3.21}$$

After simplification, it becomes a second-order differential equation about t with two boundary conditions,

$$\begin{aligned} & 2\ddot{p}_-[1 - (p_- + \dot{p}_-)](p_- + \dot{p}_-) - \\ & \dot{p}_-(\dot{p}_- + \ddot{p}_-)[1 - 2(p_- + \dot{p}_-)] = 0 \,, \quad \text{with} \\ & p_-(t_\mathrm{i}) = \frac{1}{e^{-\beta \lambda_\mathrm{i}} + 1} \quad \text{and} \quad p_-(t_\mathrm{f}) = \frac{1}{e^{-\beta \lambda_\mathrm{f}} + 1} \,. \end{aligned} \tag{3.22}$$

Here the boundary condition is satisfied because the requirement of our problem which needs the states at the initial and final time are fixed equilibrium states.

For the finite time τ, Eq. (3.22) cannot provide a real solution of p_- if the distance

between the initial state and the final state is too large. It is worthwhile to provide the upper bound of the distance here. For simplifying the calculation, we fix the initial state $p_-(t_\mathrm{i})$ to be 0.5 (i.e., $\lambda_\mathrm{i} = 0.0$), and thus we only need to determine the upper bound of $p_-(t_\mathrm{f})$. In the following derivation, we use p, $p_{\mathrm{eq,i}}$ and $p_{\mathrm{eq,f}}$ to represent p_-, $p_-(t_\mathrm{i})$ and $p_-(t_\mathrm{f})$, respectively.

We start from the integration form of Eq. (3.21):

$$\mathcal{L}(p, \dot{p}) - \dot{p}\frac{\partial \mathcal{L}}{\partial \dot{p}} = -K\,, \tag{3.23}$$

where K is a constant because the Lagrangian L does not explicitly depend on time t. After simplification, we have,

$$\frac{\dot{p}^2}{(p+\dot{p})[1-(p+\dot{p})]} = K\,. \tag{3.24}$$

Noticing that the term $(p+\dot{p})/[1-(p+\dot{p})]$ in Eq. (3.20) must be positive because the entropy production rate $\dot{\Sigma}$ is a real value, we find that K is non-negative according to Eq. (3.24). Thus, Eq. (3.24) is a quadratic equation. The solution of it can be expressed as,

$$\dot{p} = \frac{(1-2p)K \pm \sqrt{\Delta}}{2(K+1)}\,, \tag{3.25}$$

where $\Delta = K^2 + 4Kp(1-p)$.

In our setup, the controllable parameter λ_t is finite and non-negative, which ensures that the instantaneous equilibrium state $p_\mathrm{eq}(\lambda_t)$ is in the range of $[0.5, 1)$. To satisfy this, we take the "+" branch of Eq. (3.25). Then, the constant K can be calculated by integrating both sides of Eq.(3.25):

$$F(p_{\mathrm{eq,f}}, K) - F(p_{\mathrm{eq,i}}, K) = \tau\,, \tag{3.26}$$

where

$$\begin{aligned} F(p,K) &= \int \mathrm{d}p\, \frac{2(K+1)}{(1-2p)K+\sqrt{\Delta}} \\ &= -\ln(1-p) + \frac{1}{2}\ln\left[\frac{2(1-p)+K+\sqrt{\Delta}}{2p+K+\sqrt{\Delta}}\right] \\ &\quad + \frac{1}{\sqrt{K}}\arctan\left[\frac{(1-2p)\sqrt{K}}{\sqrt{\Delta}}\right]. \end{aligned} \tag{3.27}$$

Noticing that the first derivative of $p_{\mathrm{eq,f}}$ with respect to K is non-negative (See the detail in Appendix B), we deduce that $p_{\mathrm{eq,f}}$ monotonically increases as K increases. When K approaches to infinity, we can calculate the upper bound of $p_{\mathrm{eq,f}}$:

$$\begin{aligned} & \lim_{K\to+\infty} F(p_{\mathrm{eq,f}},K) - \lim_{K\to+\infty} F(p_{\mathrm{eq,i}},K) = \tau \\ \Rightarrow\quad & \ln(1-p_{\mathrm{eq,i}}) - \ln(1-p_{\mathrm{eq,f}}) = \tau \\ \Rightarrow\quad & p_{\mathrm{eq,f}} = 1-(1-p_{\mathrm{eq,i}})e^{-\tau}\,. \end{aligned} \tag{3.28}$$

In our calculation, we set $p_{\mathrm{eq,i}} = 0.5$ and $\tau = 1$. Hence, we have $\max(p_{\mathrm{eq,f}}) = 1 - 1/2e \approx 0.82$. The corresponding maximum value of λ_{f} is then calculated by

$$\max\lambda_{\mathrm{f}} = -\ln\left[\frac{1}{p_{\mathrm{eq,f}}} - 1\right] \approx 1.49\,. \tag{3.29}$$

Choosing the suitable value of λ_{f} according to (3.29) and solving the equation (3.21) numerically, one can obtain the theoretical optimal path that minimizes the entropy production during the transition between two equilibrium states in the classical case. These results will be used to check the performance of our RL method. Please see Appendix A for the details of the calculation. We also remark that the trace distance (3.14) is reduced to L_1-norm distance in the classical case:

$$d(\rho_1,\rho_2) = \sum_{\sigma=\pm} |p_{1,\sigma} - p_{2,\sigma}|\,. \tag{3.30}$$

For the cases that ε continuously changes with time, we will show that our RL method can

also find an optimal path in the subsequent section.

3.4 Results and discussions

In this section, we present the numerical results of our problem (3.2) obtained by the above RL method, and we show that it works well for our problem. Both classical and quantum cases are considered.

For the classical case, the time-dependent controllable parameter has only one component, i.e., $\Lambda_t = \lambda_t$ (ε is always 0), which leads to that the policy function $\pi(a|s, \boldsymbol{\theta})$ is chosen as an one-dimensional normal distribution (Eq.(3.7)). The numerical solution of Eq. (3.21) is used as the theoretical result. We compare results obtained by the RL method with the theoretical one to establish a benchmark for our RL method through the classical case. For the quantum case, as two controllable parameters are considered (i.e., $\Lambda_t = (\varepsilon_t, \lambda_t)$), Eq. (3.7) has to be a two-dimensional normal distribution. We consider two different situations here, i.e., the situations of $(\varepsilon(t_\mathrm{i}), \varepsilon(t_\mathrm{f})) = (0.0, 0.0)$ and $(0.5, 0.5)$ (denoted as Quantum 1 and Quantum 2 respectively for simplicity in the following descriptions).

For both classical and quantum cases, we fix the initial equilibrium state by setting λ_i to be 0.0. By changing the value of λ_f and fixing the time duration τ $(= t_\mathrm{f} - t_\mathrm{i})$ to be 1.0, we show optimal paths for different final states. The results for the classical case are shown in Figure 3.3 and for the quantum cases are shown in the Figures 3.4 and 3.5.

3.4.1 Classical case

We first discuss the results for the classical case. Figure 3.3 (a) shows the convergence of the cumulative reward after a sufficient number of episodes. When the total reward is maximized, we first check the distance Δd between the final state obtained by our RL method and the target state, which is specified by the given λ_f. If Δd approximately equals 0, we regard that the corresponding target state can be reached from the initial state with the minimum

entropy production. Indeed, if the parameter λ_{f} becomes large, one may not be able to reach $\rho_{\mathrm{eq,f}}$ within a given time τ. Thus, there must be maximum $\rho_{\mathrm{eq,f}}$ which separates the reachable and unreachable target states. This phenomenon is shown in Figure 3.3 (b-1) and the turning point of $\rho_{\mathrm{eq,f}}$ is around 1.4. This result obtained by the RL method corresponds well with our theoretical analysis in (3.29). Simultaneously, we notice that the minimum total entropy production Σ_{min} monotonically increases with λ_{f} until it reaches around this turning point in Figure 3.3. This makes sense because the dissipated energy will increase as the target state goes further and further to the initial state under the condition that the evolution time τ is fixed. In other words, there is a trade-off between the entropy production and the distance between the initial and target state: the system has to produce dissipated energy into the environment to reach the target state until the target state is too far to be reachable. In our example, when $\lambda_{\mathrm{f}} \gtrsim 1.4$, no matter how we control λ_t to increase the entropy production, target states cannot be reached. Thus, the plateau in Figure 2 (b-2) shows that the RL method can only control the system to the state that produces the dissipated energy with maximum entropy production bounded by the evolution time τ. This trade-off phenomenon reminds us of the speed limit in classical systems, which is reported in Ref. [60].

Thus, protocols of target states that $\lambda_{\mathrm{f}} \lesssim 1.4$ are exactly the optimal protocols that we intended to find in the problem (3.2). We show the evolution of p_- for each λ_{f} under the control of the optimal protocol in Figure 3.3 (c). One can see that p_- continuously varies with time from the fixed initial state to the designed target state.

The corresponding optimal protocols denoted by $\lambda^*(t)$ are shown in Figure 3.3 (d). One might expect that $\lambda^*(t)$ connecting the given value at initial and final time are smooth, but we notice that sudden jumps appear at the initial and final time of $\lambda^*(t)$. These jumps are also noticed when finding the optimal transition between two equilibrium states with minimum work in both classical Brownian particle model and the single-qubit model [95–98]. Mathematically, the phenomenon of the appearances of these sudden jumps of $\lambda^*(t)$ is caused by the solution of Eq. (3.18). At the initial (final) time, the system stays in

equilibrium state $\omega_{\rm i}$ ($\omega_{\rm f}$) around the initial (final) time, but the corresponding first derivative $\dot{p}^*_-$ is non-zero when we are about to start (ready to finish) to control the system. This non-zero $\dot{p}^*_-$ at the boundary causes the jump at the initial (final) time. Physically, we know that equilibrium states are connected through a reversible change of parameters without entropy production (i.e., through a quasi-static change of parameters). This suggests that a finite speed of parameter change does not smoothly link two different equilibrium states. To resolve this incompatibility (i.e., imposing finite-speed parameters in the present setup and the requirement of quasi-static parameter change to minimize entropy production to link two equilibrium states), the state has to be in nonequilibrium. Once the states become nonequilibrium, a finite-speed protocol can link different nonequilibrium states. This is a possible interpretation.

The accuracy of our RL method can be checked by the solution of Eq. (3.25). As shown in Figure 3.3 (b-2) and (c), one can notice that the RL can quantitatively matches the theoretical results, which means our RL method is successful in this situation. In addition, we cannot obtain the theoretical results for $\lambda_{\rm f} \gtrsim 1.4$ because Eq. (3.22) has no real solution. Intriguingly, at $\lambda_{\rm f}$ larger than 1.4 where the system cannot reach the target state, our method can still provide the results which are as close as possible to the target state. These results should be the same for every unreachable state. Hence, the similarity between the result of $\lambda_{\rm f} = 1.4$ and $\lambda_{\rm f} = 1.8$ in Figure 3.3 (c) supports that our numerical calculation is reliable.

3.4.2 Quantum case

Using the results in classical case as our benchmark of the RL method, we turn to the quantum case where it is difficult to obtain a theoretical results like above. In the quantum case, we need to optimize off-diagonal elements in the density matrix, while in the classical case, only the diagonal elements are optimized. To this end, we introduce an additional controllable parameter ε_t to control the off-diagonal elements. Thus, the controllable parameter is a two-dimensional vector in the quantum case, i.e., $\Lambda_t = (\varepsilon_t, \lambda_t)$.

In Figure 3.4, we show the results for the case $(\varepsilon_{\mathrm{i}}, \varepsilon_{\mathrm{f}}) = (0.0, 0.0)$ (Quantum 1), where the RL method gives similar results with the classical case. One can notice that the turning point between reachable and unreachable states also around 1.4, the increasing behavior of the minimum total entropy production in Figure 3.4 is similar to the classical case, and the optimal ε^* keeps to be zero during the whole driving period τ in Figure 3.4 (d-1). We mention that compared to the classical case, the controllable parameter Λ of Quantum 1 has another degree of freedom, i.e., ε is not fixed to be zero but can vary with time. This setup for Quantum 1 suggests that if an optimal protocol of Quantum 1 is found, it should be better than or equal to that of the classical case. Our RL method gives almost the same value of the minimum entropy production between these two cases (shown in Figure 3.6), which means that there is no better protocol of Quantum 1 than the one of the classical case. Also, the equivalence of the optimal paths of the Quantum 1 and the classical case provides another piece of evidence that our RL method is reliable.

Moreover, the performance of the optimal paths found by the RL method is shown in Figure 3.4 (c). Here we show the evolution of the expected value of σ_z for different λ_{f}. Similar to the result of the classical case (Figure 3.3 (c)), $\langle \sigma_z \rangle$ continuously varies with time and can reach the corresponding reachable target state. Similar to the classical case, our RL method can also provide the results which are as close as possible to the target state for the unreachable cases (see results for $\lambda_{\mathrm{f}} = 1.8$ in Figure 3.4).

Now we turn to the pure quantum situation where we set $(\varepsilon_{\mathrm{i}}, \varepsilon_{\mathrm{f}}) = (0.5, 0.5)$ (Quantum 2). In this situation, as shown in Figure 3.5 (b-1), the turning point moves to approximately 1.8 because the fixed initial state is changed when ε_{i} becomes 0.5. The target state for each λ_{f} is no longer the same as that in the classical case, which should be calculated by Eq. (3.1). The expected values of σ_x and σ_z are applied to check the performance of our RL method, and they can also reach the target values respectively, as shown in Figure 3.5 (c-1) and (c-2). For the results at λ_{f} smaller than 1.8, the distance Δd is almost 0. These results for Quantum 2 support that our RL method has the potential to find the optimal protocol

in the quantum case.

3.5 Chapter summary

This chapter proposes an RL method based on policy gradient techniques to find the optimal protocol for the controllable parameter that minimizes the total entropy production during the finite-time transition between two equilibrium states. Both classical and quantum two-level systems are considered. The optimal protocols for each system are successfully obtained by our method.

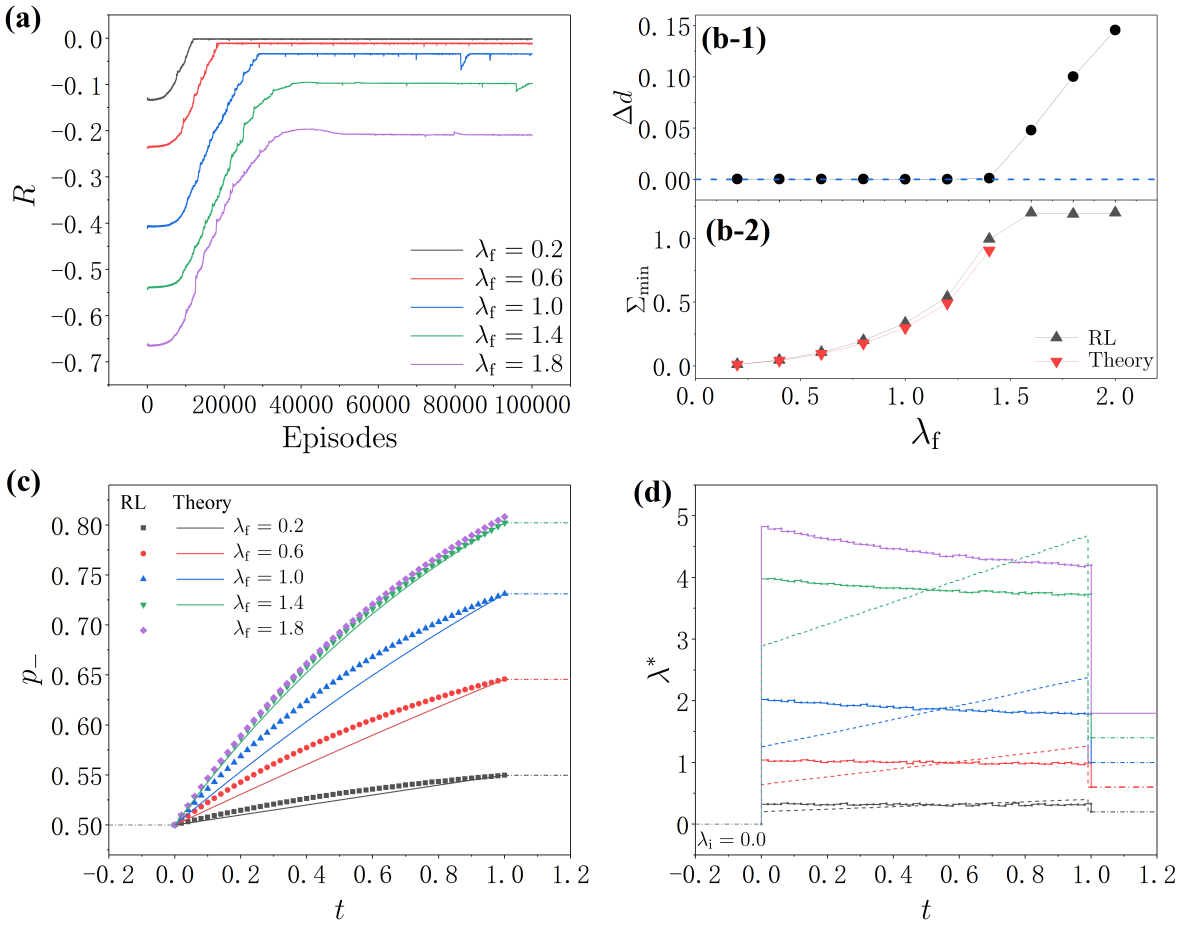

Figure 3.3: Results for finding the optimal path that minimizes the entropy production during the transition between two equilibrium states for the classical two-level system. (a) shows the total rewards in the RL setups increase and converge to a constant as the number of episodes increases from 0 to 10^5. (b) shows how the distance Δd between the target state and the final state obtained by the RL method ((b-1)), and how the corresponding minimum entropy production $\Sigma_{\min}$ ((b-2)) behaves when given the different target values of $\lambda_{\rm f}$. Red down-triangles here are the theoretical results obtained by numerically solving Eq. (3.22). (c) provides the optimal evolution of the probability of staying at the state $|e_-\rangle$ for the given different target value of $\lambda_{\rm f}$ denoted by different symbols. (d) shows the optimal protocols (stair lines) of the controllable parameter λ_t for different $\lambda_{\rm f}$s by different symbols. In (c) and (d), theoretical results are also provided for comparison denoted by solid lines with corresponding colors, and the dash-dot lines with corresponding colors denote the target states.

.

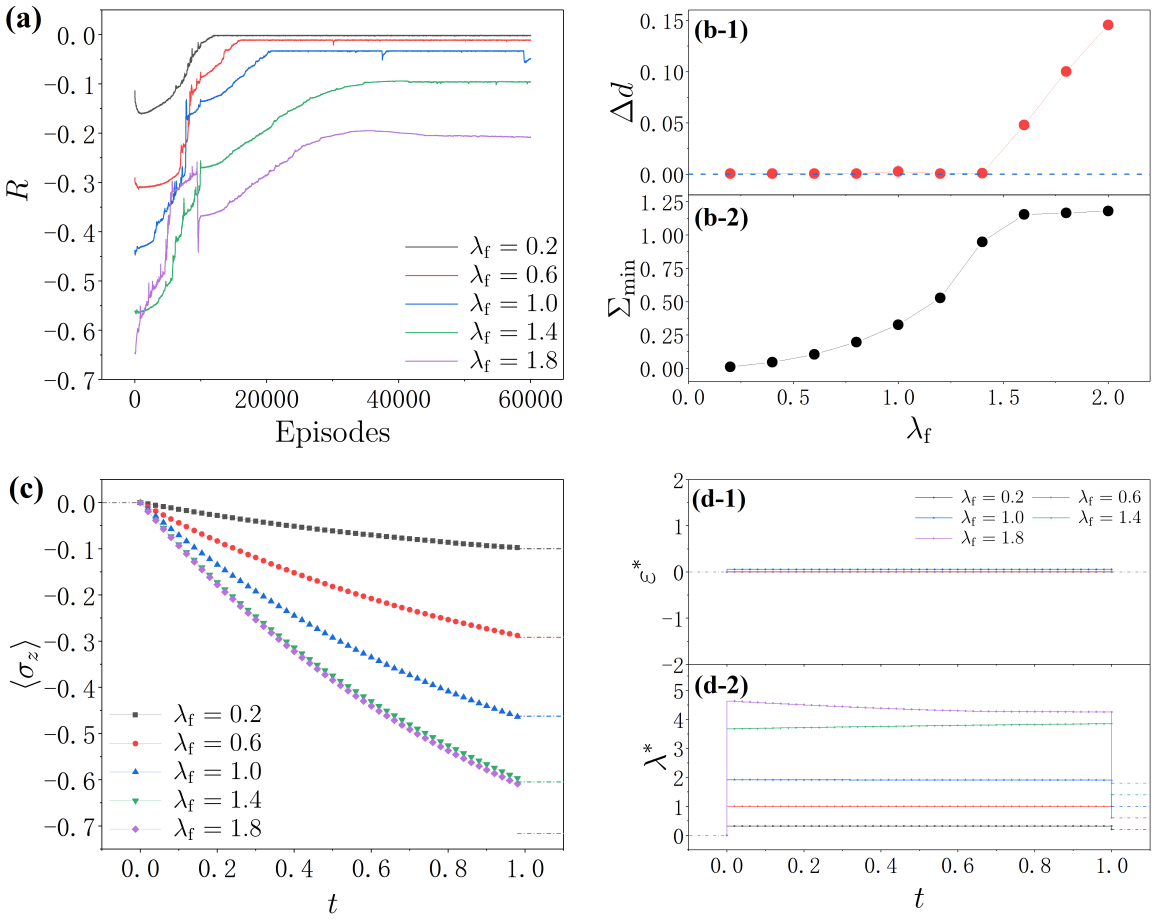

Figure 3.4: Results for finding the optimal path that minimizes the entropy production during the transition between two equilibrium points for the quantum two-level system where $(\varepsilon(t_{\rm i}), \varepsilon(t_{\rm f})) = (0.0, 0.0)$. (a) shows the total rewards in RL setups increase and converge to a constant as the number of episodes increases from 0 to 6×10^4. (b) shows how the distance Δd between the target state and the final state obtained by the RL method ((b-1)), and how the corresponding minimum entropy production $\Sigma_{\rm min}$ ((b-2)) behaves when given the different target value $\lambda_{\rm f}$. Blue dash line in (b-1) denotes for the value $\Delta d = 0$. (c) provides the optimal evolution of the expected value of σ_z for the given different target value of $\lambda_{\rm f}$ denoted by different symbols. (d-1) and (d-2) respectively shows the optimal controllable parameter ε^* and λ^* (stair lines) for different λ_f by different symbols. Gray lines in (c) and (d) denote the corresponding quantities at the initial state, while color lines corresponding to RL results denote the corresponding quantities at the target states.

.

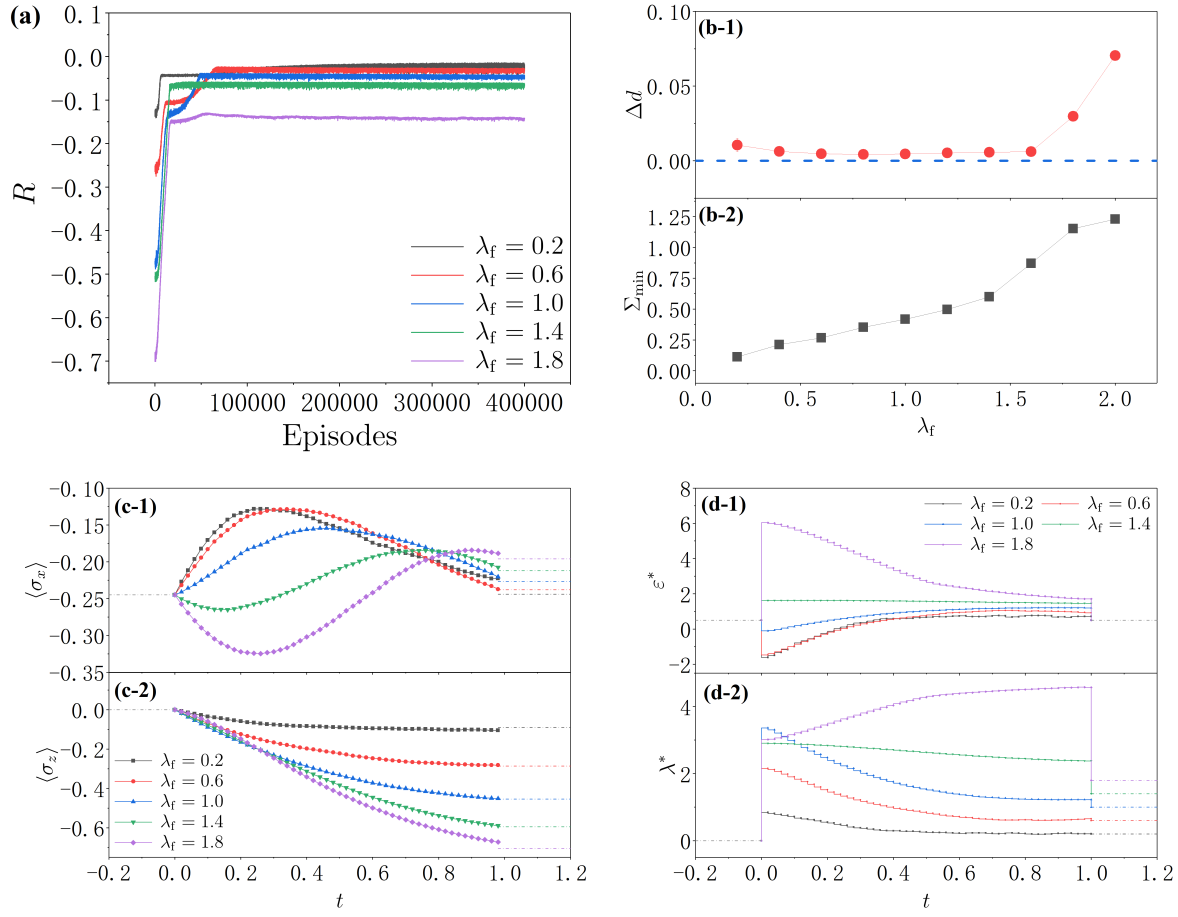

Figure 3.5: Results for finding the optimal path that minimizes the entropy production during the transition between two equilibrium points for the quantum two-level system where $(\varepsilon(t_{\rm i}), \varepsilon(t_{\rm f})) = (0.5, 0.5)$. (a) shows the total rewards in RL setups increase and converge to a constant as the number of episodes increases from 0 to 4×10^5. (b) shows how the distance Δd between the target state and the final state obtained by the RL method ((b-1)), and how the corresponding minimum entropy production $\Sigma_{\rm min}$ ((b-2)) behaves when given the different target value $\lambda_{\rm f}$. Blue dash line in (b-1) denotes for the value $\Delta d = 0$. (c-1) and (c-2) respectively provides the optimal evolution of expected value of σ_x and σ_z for the given different target value of $\lambda_{\rm f}$ denoted by different symbols. (d-1) and (d-2) respectively shows the optimal controllable parameter ε^* and λ^* (stair lines) for different $\lambda_{\rm f}$s by different symbols. Gray lines in (c) and (d) denote the corresponding quantities at the initial state, while color lines corresponding to RL results denote the corresponding quantities at the target states.

.

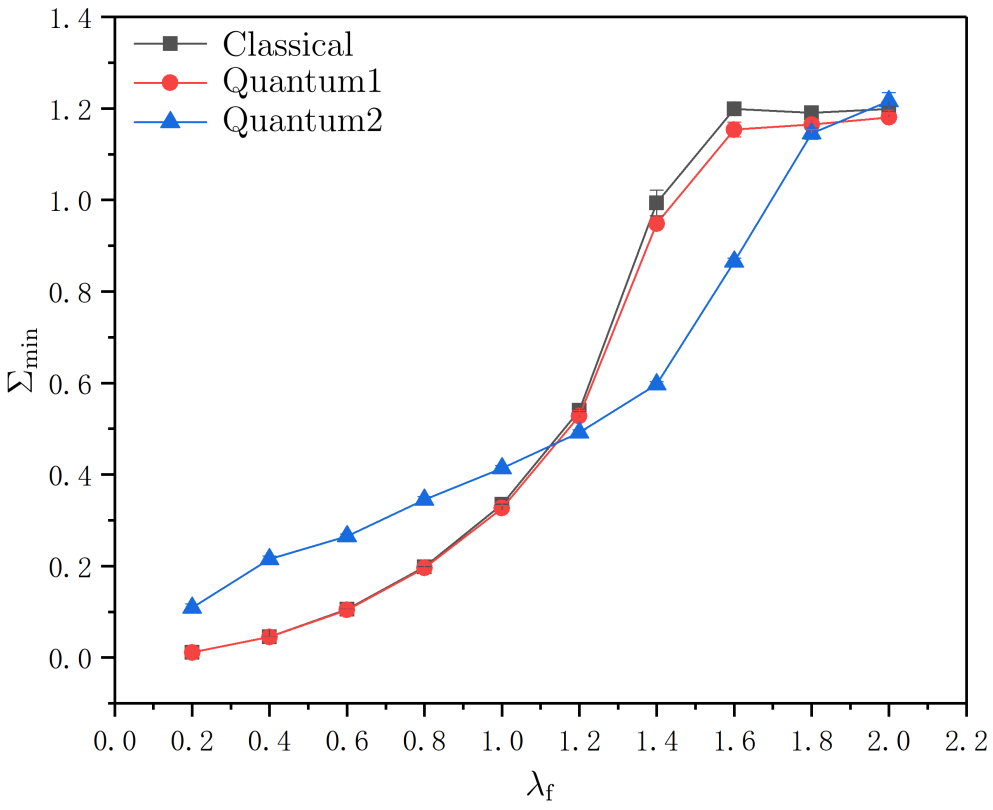

Figure 3.6: The comparison of minimum entropy production Σ_{min} provided by the RL method among the classical case, Quantum 1 and Quantum 2.

.

Chapter4

Finding Optimal Cycles in Microscopic Heat Engines

In this chapter, we aim to propose a systematic numerical approach to find the heat engine cycles in both near quasi-static and finite time cases.

For the near quasi-static case, we apply both the machine learning technique and the genetic algorithm (GA). Specifically, we apply ANNs as function approximators to represent the cycle and use the standard gradient descent method to find the optimal solution. The GA is also considered to find the optimal cycle. For the finite-time case, machine learning is applied for convenience. In this case, we have to simultaneously deal with the optimal cycle problem and evolve the system according to its dynamics. By developing the numerical methods for both goals, we will show that the method works well and obtain a physical insight for optimizing the cycles.

4.1 Heat engine with driving

To describe the main purpose of our goal and apply the numerical method clearly, we first review the framework of microscopic heat engine driving by external parameters.

As shown in Figure 4.1 (a), we consider a microscopic heat engine which consists of a working medium with the Hamiltonian $H(\lambda)$ and a heat bath with the temperature T. Here,

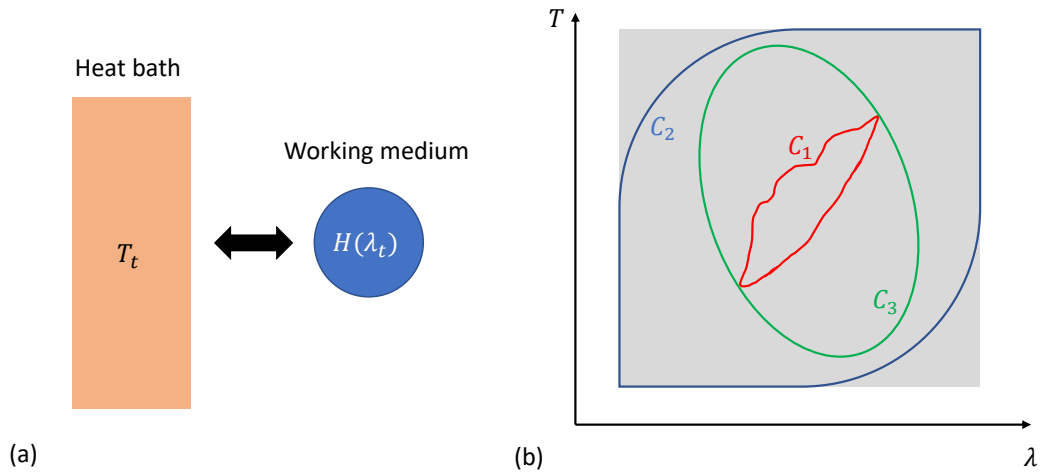

Figure 4.1: Schematic graphs. (a) The schematic graph of microscopic heat engine. It is a heat engine that consists of a working medium with controllable parameter λ interacting with a heat bath with the controllable temperature T. (b) The schematic graph of optimal cyclic protocol C in the designated regime (gray). The limitation for the near-adiabatic case is that two corners of the regime must on C as shown in (b). The cycle runs along the direction of positive output work.

λ is a time-dependent control parameter for the working medium, and the temperature also changes continuously in time. Both λ and T vary continuously and return to the original values after the period τ as shown in Figure 4.1 (b). The heat engine cycle forms a closed path C in the (λ, T)-plane.

Under the assumption that the relaxation time in the bath is much faster than the time scale of the system, the dynamics of the system is described by the time-dependent Markovian master equation for the density matrix ρ_t:

$$\partial_t \rho_t = \mathbb{L}_t \rho_t \,. \tag{4.1}$$

Eq. (4.1) takes the form of Fokker-Planck equation [99] for the classical case and the GKLS equation for the quantum case [90, 91, 100].

Following the standard framework of stochastic thermodynamics, the average work done

by the working medium to the outer environment is written as follows

$$W = \int_0^\tau dt f_t^\lambda \dot{\lambda}\,, \tag{4.2}$$

$$f_t^\lambda \equiv -\mathrm{Tr}\left[\rho_t \partial_\lambda H(\lambda)\right]\,, \tag{4.3}$$

where τ is the period of the cycle. Throughout this section, the Boltzmann constant is set to be unity. In addition, the effective heat uptake over the cycle is defined as

$$U = -\int_0^\tau dt f_t^T \dot{T}\,, \tag{4.4}$$

$$f_t^T \equiv -\mathrm{Tr}\left[\rho_t \ln \rho_t\right]\,. \tag{4.5}$$

The functions f_t^λ and f_t^T are thermodynamic forces, which respectively play roles of the instantaneous generalized pressure and entropy of the working system.

When the heat engine cycle contains continuously varying temperatures, the conventional thermodynamic efficiency is not convenient to describe the property of the system. Hence, instead, Ref.[58] introduced a more general definition on efficiency. Using the non-negativity of the entropy production rate and cyclic property, one can show the inequality $U - W \geq 0$. From this property, new efficiency η is defined as

$$\eta \equiv W/U \leq 1\,. \tag{4.6}$$

One can notice that η is 1 for reversible heat engines.

Besides, one can find the relationship between newly defined efficiency η and the Carnot efficiency in the engine between two heat bath with different constant temperature. Here we denote the temperature two heat baths are T_h and T_c respectively and set $T_h > T_c$. In a thermodynamic cycle, the working medium absorbs heat Q_h from the hot bath, produces the output work W and release heat Q_c to the cool bath. Hence, the traditional efficiency is defined by $\eta' = W/Q_h$ and the Carnot efficiency is $\eta_c = 1 - T_c/T_h$. If the entropy

change of the working medium during the hot phase is ΔS, according to the second law of thermodynamics ($\Delta S - Q_h/T_h \geq 0$), we have

$$\begin{aligned} U &= (T_h - T_c)\Delta S \\ &\geq (T_h - T_c)Q_h/T_h \\ &= Q_h\eta_c\,. \end{aligned} \tag{4.7}$$

Therefore, we find the relationship between the newly defined efficiency η and the conventional efficiency η':

$$\begin{aligned} \eta &= W/U \\ &\leq W/(Q_h\eta_c) \quad = \eta'/\eta_c\,. \end{aligned} \tag{4.8}$$

Near the quasi-static regime where the controllable parameters change very slowly, the periodic state ρ_t can be expanded from the quasi-static limit $1/\tau \to 0$ as $\rho_t \approx \rho_{eq,t} + \delta\rho_t + \delta(1/\tau^2)$, where $\rho_{eq,t}$ is the instantaneous equilibrium distribution, i.e, $\mathbb{L}_t\rho_{eq,t} = 0$. The first order $\delta\rho_t$ is solved through the equation,

$$\partial_t\rho_{eq,t} = \mathbb{L}_t\delta\rho_t\,. \tag{4.9}$$

Substituting the expressions into the thermodynamic forces, one obtains the following linear response form,

$$\begin{pmatrix} f_t^T \\ f_t^\lambda \end{pmatrix} = \begin{pmatrix} f_{eq,t}^T \\ f_{eq,t}^\lambda \end{pmatrix} + \mathcal{R}\begin{pmatrix} \dot{T} \\ \dot{\lambda} \end{pmatrix} = \begin{pmatrix} f_{eq,t}^T \\ f_{eq,t}^\lambda \end{pmatrix} + \begin{pmatrix} \mathcal{R}^{TT} & \mathcal{R}^{T\lambda} \\ \mathcal{R}^{\lambda T} & \mathcal{R}^{\lambda\lambda} \end{pmatrix}\begin{pmatrix} \dot{T} \\ \dot{\lambda} \end{pmatrix}\,, \tag{4.10}$$

where $f_{eq,t}^T$ and $f_{eq,t}^\lambda$ are thermodynamic forces of instantaneous equilibrium state. The matrix $\mathcal{R}$ is the adiabatic response matrix, which stands for the response against the perturbation from the instantaneous equilibrium state. The expression of the matrix elements in $\mathcal{R}$ depends on the choice of dynamics $\mathbb{L}_t$. Using this adiabatic expansion, it is shown in Ref.[58],

the power P and efficiency η has the following trade-off relation

$$P \leq (1-\eta)\left(\mathcal{W}/\mathcal{L}\right)^2 , \tag{4.11}$$

where $\mathcal{W}$ is the quasi-static work for given cycle C, and the quantity $\mathcal{L}$ is the thermodynamic length defined as

$$\mathcal{L} \equiv \oint_C \sqrt{g^{\mu\nu} d\mu d\nu}\,, \quad \nu, \mu = T \text{ or } \lambda\,, \tag{4.12}$$

$$g^{\mu\nu} = -(\mathcal{R}^{\mu\nu} + \mathcal{R}^{\nu\mu})/2\,. \tag{4.13}$$

The metric $g^{\mu\nu}$ is originally defined in the expansion of the function $U - W = \int_0^\tau dt g^{\mu\nu}\dot{\mu}\dot{\nu}$. The thermodynamic length is thought to be a measure of the minimal dissipation along the fixed contour C. The upper bound of power is linearly decreasing as the efficiency increases and its decreasing speed is determined by $\mathcal{W}/\mathcal{L}$. The quantity $\mathcal{W}/\mathcal{L}$ is purely geometrical quantity dependent solely on the given contour C.

Moreover, Eq. (4.11) provides a universal optimization principle that Eq. (4.11) becomes an equality if the path C is parameterized in terms of its thermodynamic length. To this end, t has to be replaced with the speed function ϕ_t:

$$t = \tau \int_0^{\phi_t} \mathrm{d}s \sqrt{g^{\mu_s\nu_s}\dot{\mu_s}\dot{\nu_s}}/\mathcal{L}\,. \tag{4.14}$$

The speed function is determined by the given cycle C and we can obtain the geometric maximum of the efficiency η according to the speed function. Numerically, the speed function ϕ_t can be determined by the point-to-point inversion and interpolation method for a given cycle C.

4.2 Main goals

It is known that the relation (4.11) is a tight bound for the power [58]. Once the cycle is determined, the upper bound can be obtained by changing the speed of protocol along the given contour C. It means that when $\mathcal{W}/\mathcal{L}$ is extremely large for a certain cycle, high efficiency and large power output of the heat engine can be obtained simultaneously using such cycle.

Hence, our first goal in this section is to consider a numerical method to find the optimal cyclic protocol C^* in the (λ, T)-plane to maximize $\mathcal{W}/\mathcal{L}$, i.e.,

$$C^* = \arg\max_C \mathcal{W}/\mathcal{L} \tag{4.15}$$

within the available parameter regime in the $\lambda - T$ plane.

Furthermore, we are also curious about the optimal cycle in the finite-time case where the controllable parameters do not vary slowly but within a time period τ. The second goal is to find the optimal cyclic protocol C^*_τ in the (λ, T)-plane to maximize $P/(1-\eta)$

$$C^*_\tau = \arg\max_C P/(1-\eta) \tag{4.16}$$

within the available parameter regime in the $\lambda - T$ plane. The optimal cycle that maximizes $P/(1-\eta)$ at finite time τ can be used to compare with the one that maximizes $\mathcal{W}/\mathcal{L}$ to show a approximating behavior as τ becomes large.

4.3 Models

4.3.1 Classical overdamped Brownian particle heat engine

We apply the overdamped Brownian particle heat engine as our one example. The working system here is an one-dimensional classical overdamped Brownian particle in the time-dependent quadratic potential: $V(x, \lambda_t) = \lambda_t x^2/2$. The temperature T_t of the system is also continuously controlled in time. Then the dynamics of the system is described by the Fokker-Planck equation:

$$\partial_t \rho_t(x) = \mathbb{L}_t \rho_t(x)\,, \tag{4.17}$$

$$\mathbb{L}_t = \lambda_t \partial_x x + T_t \partial_x^2\,, \tag{4.18}$$

where $\rho_t(x)$ is the distribution of the particle of working medium at the position x and time t. The instantaneous equilibrium state of the distribution is given by

$$\rho_{eq,t}(x) = \sqrt{\omega/\pi} e^{-\omega x^2}\,, \tag{4.19}$$

where $\omega = \lambda/2T$. To solve Eq. (4.17) in the slow driving case near quasi-static motion, the distribution $\rho_t(x)$ is expanded up to the first order. The first order of the distribution is explicitly given as

$$\delta\rho_t(x) = \frac{\dot{\omega}}{4\omega\lambda}\rho_{eq}(2\omega x^2 - 1)\,. \tag{4.20}$$

The derivation of this expression is explained in the Appendix C. Inserting the expression (4.20) to the thermodynamic forces (4.4) and (4.5), the quasi-static generalized pressure $f^{\lambda}_{eq,t}$

and the adiabatic response matrix $\mathcal{R}$ in (4.10) are given as follows

$$\mathcal{R} = \begin{pmatrix} \mathcal{R}^{TT} & \mathcal{R}^{T\lambda} \\ \mathcal{R}^{\lambda T} & \mathcal{R}^{\lambda\lambda} \end{pmatrix} = \frac{1}{4\lambda^2} \begin{pmatrix} -T/\lambda & 1 \\ 1 & -\lambda/T \end{pmatrix} , \tag{4.21}$$

$$f^{\lambda}_{eq,t} = -\frac{T}{2\lambda} . \tag{4.22}$$

From these expressions, one can arrive at the expression of the quasi-static work and thermodynamic length for the given cycle C

$$\mathcal{W} = -\oint_C \frac{T}{2\lambda} d\lambda , \tag{4.23}$$

$$\mathcal{L} = \oint_C \sqrt{\frac{T}{4\lambda^3}} \left| d\lambda - \frac{\lambda}{T} dT \right| . \tag{4.24}$$

4.3.2 Single-qubit heat engine

Another example we consider is the single-qubit heat engine. This model is also employed in Ref. [58] in a different context. The heat engine here consists of a qubit surrounded by a thermal environment with time-dependent controllable temperature T_t. The Hamiltonian of the qubit is also time-dependent by a controllable parameter λ_t

$$H_{\lambda_t} = -\frac{\hbar\Omega}{2}(\varepsilon\sigma_x + \sqrt{\lambda_t^2 - \varepsilon^2}\sigma_z) . \tag{4.25}$$

Here, σ_x and σ_z are Pauli matrices, $\hbar\Omega$ denotes overall energy scale. Both $\hbar$ and Ω are set to be unity. The term ε causes the quantum coherence and hence, the dynamics degenerate to classical once we set $\varepsilon = 0$. We regularize $\varepsilon \geq 0$ and $\lambda_t \geq \varepsilon$. The Hamiltonian (4.25) can

be decomposed by its normalized eigenvectors:

$$\begin{aligned} H(\lambda) &= \frac{\hbar\Omega\lambda}{2}(|e_\lambda^+\rangle\langle e_\lambda^+| - |e_\lambda^-\rangle\langle e_\lambda^-|) \\ \text{with eigenvectors}: \ |e_\lambda^+\rangle &= \begin{pmatrix} \sin[\theta_\lambda/2] \\ -\cos[\theta_\lambda/2] \end{pmatrix} \text{ and } |e_\lambda^-\rangle = \begin{pmatrix} \cos[\theta_\lambda/2] \\ \sin[\theta_\lambda/2] \end{pmatrix}, \\ \text{where } \sin[\theta_\lambda/2] &\equiv \frac{\varepsilon}{\sqrt{2\lambda^2 + 2\lambda\sqrt{\lambda^2-\varepsilon^2}}}. \end{aligned} \tag{4.26}$$

The equation of motion with respect to the density matrix ρ_t of the qubit is described by the GKLS equation, where the operator $\mathbb{L}_t$ satisfies,

$$\mathbb{L}_t X = -\frac{i}{\hbar}[H_\lambda, X] + \sum_\sigma \left([V^\sigma X, V^{\sigma\dagger}] + [V^\sigma, XV^{\sigma\dagger}]\right). \tag{4.27}$$

Here the operator V^σ is the jump operator satisfying the quantum detailed balance condition, $[H_\lambda, V^\pm] = \pm\hbar\Omega\lambda V^\pm$, and $\mathrm{Tr}\left[V^\pm V^{\pm\dagger}\right] = \pm\Gamma\Omega\lambda/\left[1-\exp\left[\mp\hbar\Omega\lambda/T\right]\right]$:

$$\begin{aligned} V^\pm &= \frac{1}{2}\sqrt{\frac{\pm\Gamma\Omega\lambda}{1-\exp\left[\mp\hbar\Omega\lambda/T\right]}}(J_x \pm iJ_y), \\ &\text{where} \\ J_x &\equiv |e_\lambda^+\rangle\langle e_\lambda^-| + |e_\lambda^-\rangle\langle e_\lambda^+|, \\ J_y &\equiv i|e_\lambda^-\rangle\langle e_\lambda^+| - i|e_\lambda^+\rangle\langle e_\lambda^-|, \\ J_z &\equiv |e_\lambda^+\rangle\langle e_\lambda^+| - |e_\lambda^-\rangle\langle e_\lambda^-|. \end{aligned} \tag{4.28}$$

The constant Γ is the jumping rate and set to be unity in our numerical calculations.

4.3.2.1 The finite-time case

To calculate the output work W and the heat uptake U from the heat bath, one needs to evolve the system into periodic state ρ_t. To simplify the calculation of the periodic state ρ_t,

we parametrize the periodic state ρ_t in the form:

$$\rho_t \equiv \frac{\mathbb{1}}{2} + \varrho_x J_x + \varrho_y J_y + \varrho_z J_z \,. \tag{4.29}$$

Plugging this ansatz into the master equation (4.1) and (4.27) yields the generalized Bloch equations

$$\frac{\partial}{\partial t}\begin{pmatrix} \varrho_x \\ \varrho_y \\ \varrho_z \end{pmatrix} = \begin{pmatrix} -k_\lambda^+ & -\Omega\lambda & -\theta'_\lambda \dot{\lambda} \\ \Omega\lambda & -k_\lambda^+ & 0 \\ \theta'_\lambda \dot{\lambda} & 0 & -2k_\lambda^+ \end{pmatrix}\begin{pmatrix} \varrho_x \\ \varrho_y \\ \varrho_z \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ k_\lambda^- \end{pmatrix} . \tag{4.30}$$

where the parameters are

$$k_\lambda^\pm \equiv \Gamma\Omega\lambda \frac{1 \pm \exp\left[\hbar\Omega\lambda/T\right]}{\exp\left[\hbar\Omega\lambda/T\right]-1} \text{ and } \theta'_\lambda \equiv \frac{\varepsilon}{\lambda\sqrt{\lambda^2-\varepsilon^2}} \,. \tag{4.31}$$

For a given specific driving protocol C, the differential equation (4.30) can be solved. Then the output work W and the heat uptake U can be determined by

$$\begin{aligned} W &= -\hbar\Omega \int_0^\tau \mathrm{d}t \left(\lambda\theta'_\lambda \varrho_x + \varrho_z\right)\dot{\lambda} \quad \text{and} \\ U &= \frac{1}{2}\int_0^\tau \mathrm{d}t \left(4\varrho_t \mathrm{arctanh}[2\varrho_t] + \ln\left[\frac{1}{4} - \varrho_t^2\right]\right)\dot{T}, \\ &\text{where} \quad \varrho_t \equiv \sqrt{\varrho_x^2 + \varrho_y^2 + \varrho_z^2}. \end{aligned} \tag{4.32}$$

4.3.2.2 The near quasi-static case

From the adiabatic approximation, the quasi-static work $\mathcal{W}$ is given as

$$\mathcal{W} = \frac{\hbar\Omega}{2}\oint_C \tanh\left[\hbar\Omega\lambda/2T\right] d\lambda \,, \tag{4.33}$$

We can calculate the thermodynamic length by Eq. (4.12) and the adiabatic response matrix

$$\begin{aligned}\mathcal{R} &= \mathcal{D}+\mathcal{C} = \begin{pmatrix} \mathcal{D}^{TT} & \mathcal{D}^{T\lambda} \\ \mathcal{D}^{\lambda T} & \mathcal{D}^{\lambda\lambda} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{C}^{\lambda\lambda} \end{pmatrix} \\ &= \frac{R}{T^2}\begin{pmatrix} -\lambda/T & 1 \\ 1 & -T/\lambda \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{C}^{\lambda\lambda} \end{pmatrix}, \end{aligned} \tag{4.34}$$

where,

$$\begin{aligned} R &= \frac{\hbar^2\Omega}{8\Gamma}\frac{\tanh\left[\hbar\Omega\lambda/2T\right]}{\cosh^2\left[\hbar\Omega\lambda/2T\right]} \\ \text{and} \qquad \mathcal{C}^{\lambda\lambda} &= -\frac{\hbar\varepsilon^2}{2\lambda^2(\lambda^2-\varepsilon^2)}\frac{\Gamma}{\Gamma^2\coth^2\left[\hbar\Omega\lambda/2T\right]+1}\,. \end{aligned} \tag{4.35}$$

Especially for $\varepsilon = 0$, the thermodynamic length is given by $\mathcal{L}_d$:

$$\mathcal{L}_d = \oint_C \sqrt{-g_d^{\mu\nu}d\mu d\nu} = \oint_C \sqrt{\frac{R}{4\lambda T}}\left|d\lambda - \frac{\lambda}{T}dT\right|. \tag{4.36}$$

When $\varepsilon \neq 0$, we obtain different expressions containing the quantum effect. See [58] and Appendix D for the details.

It is interesting that $\mathcal{L}_d$ shows the similar expression to $\mathcal{L}$ of the classical Brownian heat engine (Eq. (4.24)). When thermodynamic length takes its minimum value 0, the absolute terms in both Eq. (4.24) and Eq. (4.36) are zero and give the differential equation,

$$d\lambda = \frac{\lambda}{T}dT\,, \tag{4.37}$$

One can solve this equation directly and find that λ is linearly proportional to T, i.e., $\lambda = \kappa T$, where κ is a time-independent constant. In this situation, the corresponding quasi-static work is also zero.

4.4 Method: genetic algorithm

In this section, we apply the genetic algorithm to deal with our first goal [101].

Here we have two constraints on this problem. First, concerning the available parameter regime, for simplicity, we fix the two points as shown in Figure 4.2 (a), and the available parameter regime is assumed to be inside the square given determined by the two points (the gray shaded regime). Although this setup is simple, there are infinite numbers of candidates of cycles even in this limited regime, and hence the optimization is highly nontrivial. Second, to avoid the trivial solution (4.37), we put a constraint $\mathcal{W} \geq \mathcal{W}_0$ on the adiabatic work to avoid zero work production.

We first transform the problem (4.15) to an optimization problem by parametrization and then apply the genetic algorithm to solve the optimization problem.

4.4.1 Parametrization

The constraints we impose for the cycle are listed as follows. (i) The cycle should be a simple closed curve, which means that the curve has no intersections. Otherwise, the protocol can be regarded as several sub-cycles. (ii) As explained in the section 4.2, we impose the constraint that the cycle must pass through the diagonal points A and B of the given square region D (see Figure 4.1 (b)). By imposing special constraints, we can make sure the curve we find is a simple closed curve. (iii) We also impose the condition that the quasi-static work along the cycle must be larger than a given constant, i.e., $\mathcal{W} \geq \mathcal{W}_0$.

Concerning the constraint (ii), the available regime D is mathematically described as

$$D = \{(\lambda, T) | \lambda \in [1.0, 2.0], T \in [1.0, 2.0]\} . \tag{4.38}$$

Because controllable parameters, such as the temperature T and strength of potential or shifting energy λ, are usually finite.

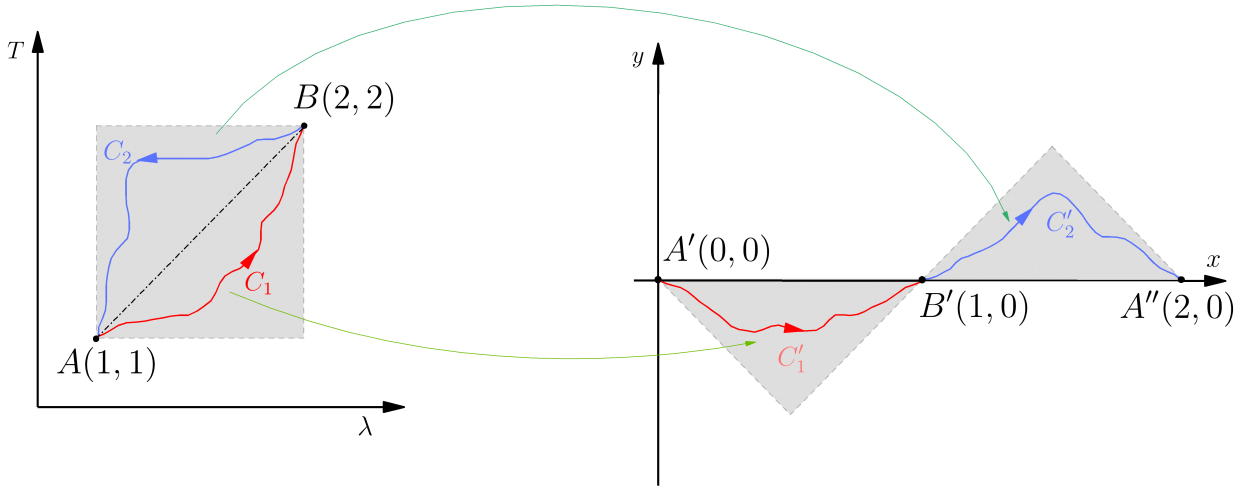

Figure 4.2: Schematic of the parametrization. We consider the cycle that passes two points A and B inside the gray regime. The original cycle on $T-\lambda$ plane is mapped onto the $x-y$ plane as in the right figure.

Based on the above constraints on the cycle, we employ the following parametrizing method. See the schematic graph in Figure 4.2. Assume that the optimal cycle C^* must path through two corners of the available square regime D: $A(1,1)$ and $B(2,2)$ which divide the path into two parts, $C_1 : A \to B$ and $C_2 : B \to A$. We then map the closed curve C^* into an unclosed curve C' by the following coordinate transformations (See Figure 4.2, again). We first translate coordinates so that A can overlap with the origin, and rotate them $\pi/4$ counterclockwise to ensure that B lies in the horizontal axis. Next, we fold the part C_2 into C_2' centered on B. Thus, A is folded to the other side of B along the horizontal axis. After zooming, the intersects between the new curve C' and the horizontal axis are $A'(0,0)$, $B'(1,0)$ and $A''(2,0)$. Then, λ and T can be expressed by the new coordinates x and y, i.e.,

$$\begin{cases} \lambda = x - y + 1 \\ T = x + y + 1 \end{cases} \quad \text{for} \qquad C_1', \tag{4.39}$$

and

$$\begin{cases} \lambda = 3 - x - y \\ T = 3 - x + y \end{cases} \text{for} \qquad C_2' \,, \tag{4.40}$$

where C_1' and C_2' respectively corresponds C_1 and C_2 in original curve. We can express them respectively by functions of x, $g_1(x)$ and $g_2(x)$. The total quasi-static work and thermodynamic length can then be expressed as the summations of the quasi-static works and thermodynamic lengths for C_1 and C_2 respectively.

Thus, our optimization problem becomes finding the optimal $g_1(x)$ and $g_2(x)$ to maximize $\mathcal{W}/\mathcal{L}$, i.e.,

$$g_1^*, g_2^* = \arg\max_{\mathrm{g_1,g_2}} \frac{\mathcal{W}_1 + \mathcal{W}_2}{\mathcal{L}_1 + \mathcal{L}_2} \,, \tag{4.41}$$

where $\mathcal{W}_m$ and $\mathcal{L}_m$ $(m = 1, 2)$ are the quasi-static works and the thermodynamic lengths for the contour C_m, respectively.

In addition, we parametrize the functions $g_m(x)$ $(m = 1, 2)$ with the Fourier representation:

$$g_m(x; \boldsymbol{b}^{(m)}) \;\; = \textstyle\sum_{j=1}^{n} b_j^{(m)} \sin(j\pi x) \,, \;\; m = 1 \,, 2 \,. \tag{4.42}$$

Finding optimal path now reduces to finding the Fourier components, and hence the corresponding $\mathcal{W}_m$ and $\mathcal{L}_m$ is respectively parametrized by $\boldsymbol{b}^{(m)}$ as $\mathcal{W}_m(\boldsymbol{b}^{(m)})$ and $\mathcal{L}_m(\boldsymbol{b}^{(m)})$.

Finally, we write the constraints (i), (ii) and (iii) in terms of the functions $g_m(x; \boldsymbol{b}^{(m)})$. Note that without loss of generality, we choose the condition that $g_1(x) \leq 0$ and $g_2 \geq 0$ to

satisfy the Constraint (i), combining the Constraint (ii),

$$\begin{aligned} g_1(x;\boldsymbol{b}^{(1)}) &\geq -x \quad \text{for } x \in [0, 0.5), \\ g_1(x;\boldsymbol{b}^{(1)}) &\geq x-1 \text{ for } x \in [0.5, 1), \\ g_2(x;\boldsymbol{b}^{(2)}) &\leq x-1 \text{ for } x \in [1, 1.5), \\ g_2(x;\boldsymbol{b}^{(2)}) &\leq 2-x \text{ for } x \in [1.5, 2]. \end{aligned}$$

The expressions for the constraints can be written as follows

$$\begin{aligned} &M_1(x;\boldsymbol{b}^{(1)}) = g_1(x;\boldsymbol{b}^{(1)}) \leq 0 && x \in [0,1) \\ &M_2(x;\boldsymbol{b}^{(2)}) = -g_2(x;\boldsymbol{b}^{(2)}) \leq 0 && x \in [1,2] \\ &M_3(x;\boldsymbol{b}^{(1)}) = -g_1(x;\boldsymbol{b}^{(1)}) - x \leq 0 && x \in [0,0.5) \\ &M_4(x;\boldsymbol{b}^{(1)}) = -g_1(x;\boldsymbol{b}^{(1)}) + x - 1 \leq 0 && x \in [0.5,1) \\ &M_5(x;\boldsymbol{b}^{(2)}) = g_2(x;\boldsymbol{b}^{(2)}) - x + 1 \leq 0 && x \in [1,1.5) \\ &M_6(x;\boldsymbol{b}^{(2)}) = g_2(x;\boldsymbol{b}^{(2)}) + x - 2 \leq 0 && x \in [1.5,2] \\ &\mathcal{W}(\boldsymbol{b}^{(m)}) = \textstyle\sum_{m=1}^{2} \mathcal{W}_m(\boldsymbol{b}^{(m)}) \geq \mathcal{W}_0\,. \end{aligned} \tag{4.43}$$

Consequently, our optimization problem (4.15) is reduced to finding the optimal parameters $\boldsymbol{b}_*^{(m)}$ under constraints (4.43) to maximize $\mathcal{W}/\mathcal{L}$, i.e.,

$$\boldsymbol{b}_*^{(m)} = \arg\max_{\boldsymbol{b}^{(m)}} \frac{\mathcal{W}(\boldsymbol{b}^{(m)})}{\mathcal{L}(\boldsymbol{b}^{(m)})}, \quad \text{with the condition (4.43)} \tag{4.44}$$

4.4.2 Genetic algorithm

Here we consider the genetic algorithm with penal functions to solve the optimization problem (4.43). The GA has been widely developed and applied to various optimization problems in computer science [102–106]. Because of its problem independence and random-searching technique, the GA can maximally avoid getting trapped in local extremums, be generalized to other related problems, and be succinctly performed when dealing with the problem with

complex constraints [103–106]. The convergence can be checked through a relevant error estimation. The GA is the stochastic optimization method via simulating the natural selection [102], which is originally inspired by the process of natural selection and biological evolution based on genetic mechanism. Focusing on all individuals in a population, the GA randomly searches the optimal solution in coded parameter space. This gives its well global searching ability. To establish the GA on our problem, the following five parts are needed to be carefully considered, including parameter-coding, initialization, design of fitness function and genetic operations (selection, crossover, and mutation), and settings of control parameters.

First, we consider the parameter-coding process. For our problem, the parameter vector to be optimized is $\boldsymbol{b}^{(m)}$. Thus, one can uniquely specify some imaginary creature by the values of $\boldsymbol{b}^{(m)}$. Similar to the gene sequence in the real world, if one regards $\boldsymbol{b}^{(m)}$ as the unique character of the imaginary creature, its binary notation vector $\hat{\boldsymbol{b}}^{(m)}$ then can be regarded as the corresponding unique gene sequence. The length of the binary notation can be adjusted by a factitiously determined precision Δ.

Second, in order to simulate the natural selection process, a population which consists of N arbitrary imaginary creature is needed. Initially, it is a set of randomly-selected parameter vectors, i.e., $G = \{\hat{\boldsymbol{b}}_1^{(m)}, \hat{\boldsymbol{b}}_2^{(m)}, ..., \hat{\boldsymbol{b}}_N^{(m)}\}$.

Third, natural selection process selects the creature according to its fitness, which is described by the fitness function in the GA. Basically, the fitness function is often set to be the target function. For our problem, considering the constraints (4.43), we transform the problem (4.44) into the following one without constraints with the help of penal function method [104–106],

$$b_*^{(m)} = \arg\min_{\boldsymbol{b}^{(m)}} \quad F(\boldsymbol{b}^{(m)}) = -\frac{\mathcal{W}(\boldsymbol{b}^{(m)})}{\mathcal{L}(\boldsymbol{b}^{(m)})} + P(\boldsymbol{b}^{(m)}). \tag{4.45}$$

Here, $P(\boldsymbol{b}^{(m)})$ is the penal function,

$$P(x;\boldsymbol{b}^{(m)}) = \sigma \sum_{k=1}^{n_c} \sum_{i=1}^{6} \left|\max(0, M_i(x_k;\boldsymbol{b}^{(m)})\right| + \sigma \left|\max(0, \mathcal{W}(\boldsymbol{b}^{(m)}))\right|$$

by equally dividing the range $[0, 2]$ of x with n_c points. The symbol σ represents a large penal constant. Hence, the fitness function in our problem is set to be $F(\boldsymbol{b}^{(m)})$. In the selection step, we calculate $F(\boldsymbol{b}^{(m)})$ for every creature in the population G. If some creature has a low value of fitness, it may die out with a larger probability than others. We select N_s creatures in G to survive. Please see Appendix E for details.

Moreover, in order to adapt to the environment, these N_s surviving creatures must reproduce their children. Parents are equally randomly selected in the surviving creatures and a pair of parents can only reproduce one child. By gene mutation and gene crossover process, genes in children can be optimized. In situations considered in this Chapter, two random sites of each dimension of the binary notation vector $\hat{\boldsymbol{b}}^{(m)}$ are selected to be the crossover points on both parents' gene sequences, and each fragment divided by the selected sites has a probability to be inherited by their child. In the gene mutation process, random sites of each dimension of the binary notation vector $\hat{\boldsymbol{b}}_1^{(m)}$ has a probability to change to its opposite value (0 or 1) only.

Thus, the remaining N_s parents and the new $N - N_s$ child make up the new population. After enough such iterations, the process of natural selection ensures that the fitness of each individual in the population can be optimized.

4.4.3 Example to check the GA

To demonstrate that the GA can correctly find the optimal solution for our problem, we present an analytically solvable problem in this section. The problem is to find the optimal protocol γ^* in the parameter plane $\lambda - T$ from $A(1, 1)$ to $B(2, 2)$ that minimizes the minimum thermodynamic length $\mathcal{L}$. We consider this problem for the classical overdamped Brownian

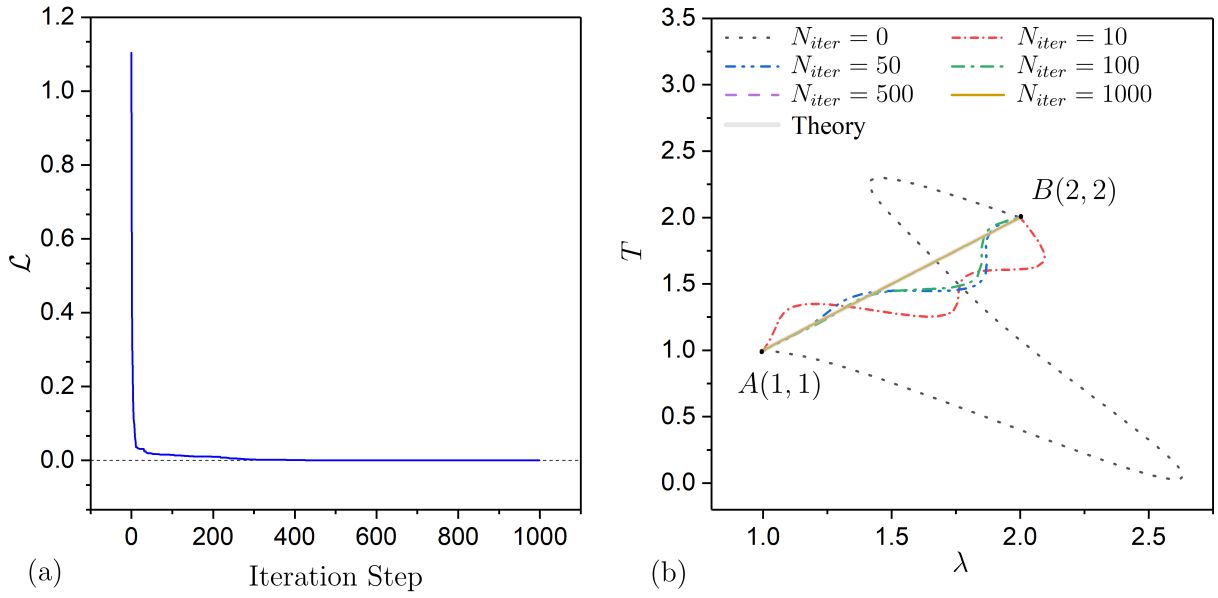

Figure 4.3: Numerical results of optimal protocol that minimize the thermodynamic length of classical overdamped Brownian particle heat engine. (a) Relationship between objective function $\mathcal{L}$ and iteration step of Genetic Algorithm. They show that $\mathcal{L}$ decreases and converge to 0 as iteration step increases from 0 to 1000. (b) Optimal protocols for different iteration steps. The gray line is the theoretical result $T = \lambda$. We set the Fourier bases to be 5 here.

particle, i.e.,

$$\gamma^* = \arg \min_{\gamma: A \to B} \mathcal{L}. \tag{4.46}$$

As discussed in Section 4.2, we can theoretically find the optimal path γ^*. The solution is a simple line $T = \lambda$, where the corresponding minimum thermodynamic length $\mathcal{L}^* = 0$. Numerically, we deal with this problem by employing the parametrization method and the GA above. Instead of considering the whole cycle, we only need to parametrize half of it (i.e., C_1) like the form of Eq. (4.43). Besides, we do not have any constraints on γ here. Results are shown in Figure 4.3. One can notice that as the iteration steps increases, the protocol gradually converges to the theoretical solution (Figure 4.3 (a)) and the target function $\mathcal{L}$ converges to the theoretical prediction 0 (Figure 4.3 (b)). It indicates that our method with the GA is reliable.

4.5 Results and discussions 1

In this section, we present numerical results obtained by the above method, using the target function F in (4.45). We maximize the ratio $\mathcal{W}/\mathcal{L}$ for the models that we have introduced in the model section, i.e., the classical Brownian heat engine, the classical and quantum single-qubit heat engines. See detailed setup for the GA in Appendix E. As shown below, large $\mathcal{W}$ requires a large area of the cycle, while small $\mathcal{L}$ needs a small area. Hence, there is a sort of trade-off to maximize the ratio $\mathcal{W}/\mathcal{L}$. This competition leads to several nontrivial behaviors on the heat engine cycle in the available squared region. We runs our programs in sufficient steps to ensure our results convergent. Figure 4.4 shows the convergence for each examples.

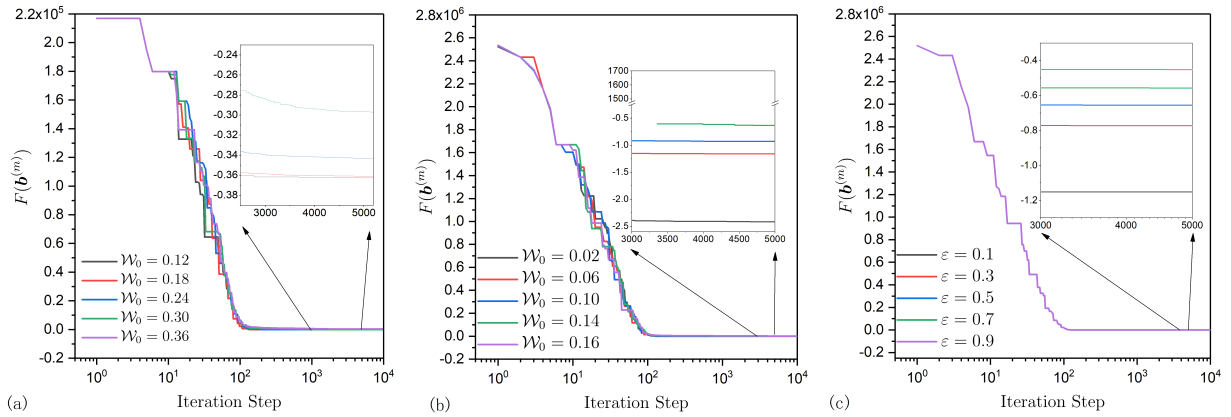

Figure 4.4: (a), (b) and (c) respectively shows that objective function $F(\boldsymbol{b}^{(m)})$ converges to a constant as iteration step increases for the Brownian heat engine, single-qubit heat engine in the quasi-classical case ($\varepsilon = 0.0$) and that in the quantum case ($\varepsilon \neq 0.0$). In (a) and (b), lines of different colors represent the results for different values of lower bound $\mathcal{W}_0$ of quasi-static work. In (c), results for different εs are represented in different colors. Details are shown in the insets. Here, we use the same random seed in the numerical calculation as an initial state, and hence, different curves overlap for different parameters at small iteration steps.

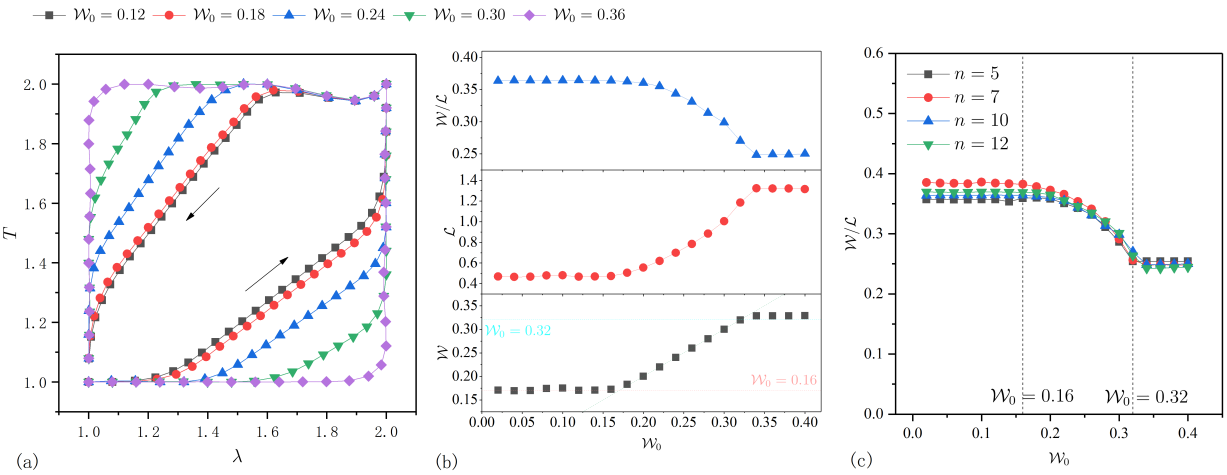

Figure 4.5: Numerical results of optimal thermodynamic cycle for classical overdamped Brownian particle heat engine. (a) shows that how optimal cycles behave in the parameter plane $T - \lambda$. The directions of cycles are counterclockwise along which the quasi-static work is positive. (b) is the relation of optimal quantities versus $\mathcal{W}_0$. Quasi-static work $\mathcal{W}$, thermodynamic length $\mathcal{L}$ and the ratio $\mathcal{W}/\mathcal{L}$ is presented by the blue, red and black line, respectively. (c) indicates the effect of the number of Fourier bases on $\mathcal{W}/\mathcal{L}$.

4.5.1 Brownian heat engine

We first show the results for the Brownian heat engine. In this case, we set a finite constraint on the quasi-static work, i.e., $\mathcal{W} \geq \mathcal{W}_0$ (> 0). Results are shown in Figure 4.5. Here, we use the Fourier modes $n = 10$.

Once target functions converge, optimal thermodynamic cycles can be obtained, which are shown in Figure 4.5 (a). We also show the saturated values of $\mathcal{W}/\mathcal{L}$, $\mathcal{L}$, and $\mathcal{W}$ as a function of constraint $\mathcal{W}_0$ in Figure 4.5 (b). As shown in Figure 4.5 (a), by imposing the constraint on the quasi-static work $\mathcal{W}$, one can see that when we are increasing the constraint $\mathcal{W}_0$, the optimal cycle approaches to the boundary of D which is the allowable maximum cycle in the domain. This supports the intuition that a large work $\mathcal{W}$ requires a large area.

The work $\mathcal{W}$ reaches its maximum value (~ 0.32) shown in Figure 4.5 (b). The available domain D plays the role of an upper bound on the quasi-static work. To produce the maximum work and maximum $\mathcal{W}/\mathcal{L}$ simultaneously, the heat engine is required to be operated inside the boundary of D. Particularly, if $\mathcal{W}_0$ exceeds the upper limit (~ 0.32), i.e., the constraint $\mathcal{W} \geq \mathcal{W}_0$ is no longer applicable. In this case, the term $\sigma \left|\max(0, \mathcal{W}(\boldsymbol{b}^{(m)})\right|$ in the penal function (4.46) can never be avoided. Our algorithm can still minimize the target function F and reduce the penal term above as much as possible. The consequence of this trend is that the quasi-static work can still be maximized to the upper limit, but the corresponding optimal cycle will no longer change. This is why the cycles overlap with each other when $\mathcal{W}_0 > 0.32$ in Figure 4.5 (a).

By imposing a large constraint $\mathcal{W}_0$ as above, one can see that a large area in the cycle is generally necessary for large adiabatic work $\mathcal{W}$. On the other hand, we also need to note that a small area in the cycle is required for a small thermodynamic length $\mathcal{L}$. Hence, in general, there is a sort of trade-off between obtaining large $\mathcal{W}$ and reducing $\mathcal{L}$. Because the adiabatic work, $\mathcal{W}$, measures an amplitude of power when τ is fixed and $\mathcal{L}$ is a measure of minimal dissipation along the fixed cycle, the trade-off between $\mathcal{W}$ and $\mathcal{L}$ is exactly another

representation of the power-efficiency trade-off of the heat engine. We are decreasing the constraint $\mathcal{W}_0$, the effects arising from this trade-off become significant. Let us discuss the regime of $\mathcal{W}_0 < 0.32$ in the bottom figure of Figure 4.5 (b). The figure shows that in this regime, the quasi-static work along the optimal cycle tends to equal $\mathcal{W}_0$. It is also equivalent to that our algorithm tries to obtain a small area of the engine cycle, which leads to small values of $\mathcal{L}$. Hence, the numerical results non-trivially show that to obtain the maximizing the rate $\mathcal{W}/\mathcal{L}$, obtaining small adiabatic work $\mathcal{W}$ is a good strategy. This also implies that the constraint on $\mathcal{W}$ limits the maximum value of $\mathcal{W}/\mathcal{L}$ in this regime. We also notice that the corresponding optimal cycle also gradually shrinks from the boundary (the cases $\mathcal{W}_0 = 0.18, 0.24$ and 0.30) in Figure 4.5 (b). We believe that this behavior appears as a result of nontrivial trade-off properties to $\mathcal{W}/\mathcal{L}$.

Until $\mathcal{W}_0 \leq 0.17$, flat behavior appear again in the graph of $\mathcal{W}$, $\mathcal{L}$ and $\mathcal{W}/\mathcal{L}$. These platforms indicate that there will be no optimal cycle better than that of $\mathcal{W} = 0.17$, i.e., the cycle we obtain here is the global optimal cycle that can maximize $\mathcal{W}/\mathcal{L}$ in domain D.

We now recall that the value $\mathcal{W}/\mathcal{L}$ determines whether the large power and large efficiency can be obtained simultaneously, as shown in the general theory (4.11). As shown in Figure 4.5 (b), the maximum value of $\mathcal{W}/\mathcal{L}$ is about 0.40, which is finite, and hence large power and large efficiency cannot be obtained simultaneously in this system. However, we stress that based on our approach, one can systematically find the optimal cycle for the Brownian heat engine.

4.5.2 Single-qubit heat engine

We next show that our numerical scheme also works well in the single-qubit heat engines. In this case, we can consider the effect of the quantum coherence ε in the dynamics. Hence, we consider two cases, i.e., the quasi-classical case and the quantum case. In the quasi-classical case, we set the quantum coherence ε to zero in the Hamiltonian (4.25), and we set a finite constraint on the adiabatic work. In the quantum case, we set a finite value for the quantum

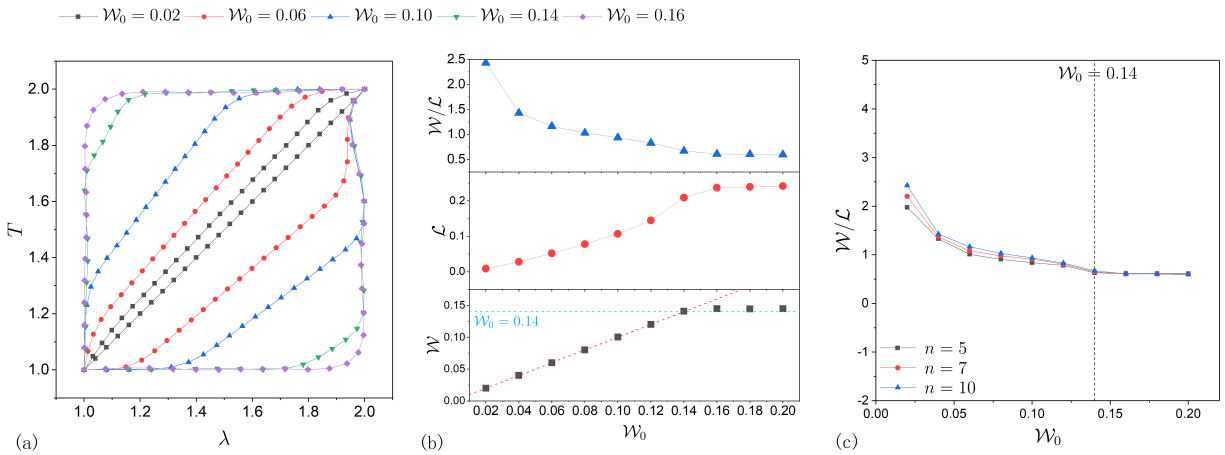

Figure 4.6: Numerical results of optimal thermodynamic cycle for quantum single-qubit heat engine in quasi-classical limit ($\epsilon = 0$). (a) shows that how optimal cycles behave in the parameter plane $T - \lambda$. The directions of cycles are counterclockwise along which the quasi-static work is positive. (b) is the relation of optimal quantities versus $\mathcal{W}_0$. Quasi-static work $\mathcal{W}$, thermodynamic length $\mathcal{L}$ and the ratio $\mathcal{W}/\mathcal{L}$ is presented by the blue, red and black line, respectively. (c) indicates the effect of the number of Fourier bases on $\mathcal{W}/\mathcal{L}$.

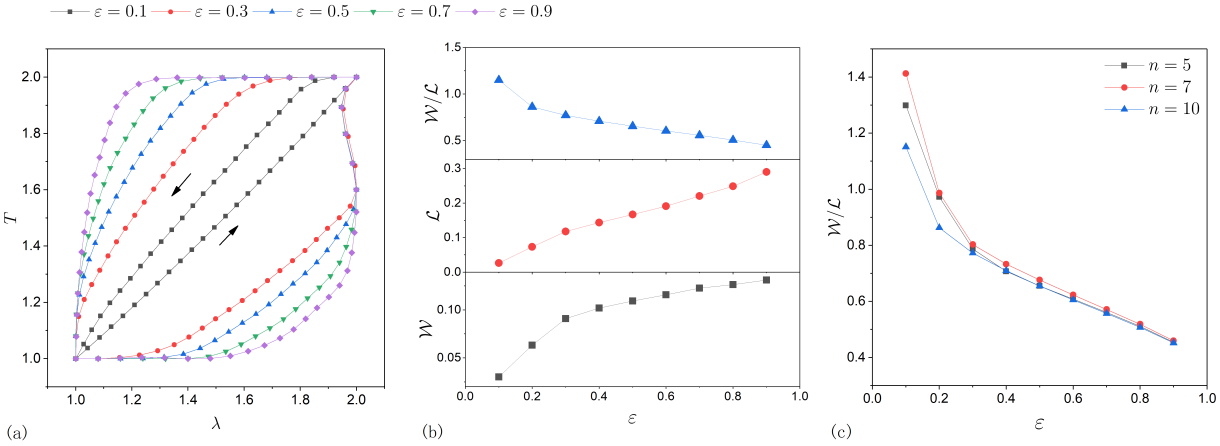

Figure 4.7: Numerical results of optimal cyclic protocols for quantum single-qubit heat engine in the quantum case. (a) shows that how optimal cycles behave in the parameter plane $T - \lambda$. The directions of cycles are counterclockwise along which the quasi-static work is positive. (b) is the relation of optimal quantities versus ϵ. Quasi-static work $\mathcal{W}$, thermodynamic length $\mathcal{L}$ and the ratio $\mathcal{W}/\mathcal{L}$ is presented by the blue, red and black line, respectively. (c) indicates the effect of the number of Fourier bases on $\mathcal{W}/\mathcal{L}$.

coherence ε, and we do not impose a constraint on the adiabatic work, i.e., $\mathcal{W}_0 = 0$.

We first discuss the quasi-classical case $\varepsilon = 0$. The results are shown in Figure 4.6. We use the Fourier mode $n = 10$, which does not cause significant errors, as can be seen in Figure 4.6 (c). This indicates that our numerical scheme works well even in discrete level heat engines. Compared to the Brownian heat engine, the global optimal protocol cannot be found when decreasing the lower bound $\mathcal{W}_0$ of the adiabatic work. Instead, within the allowable range of error, we did not observe a large value of $\mathcal{W}/\mathcal{L}$ as $\mathcal{W}_0$ asymptotes to zero, as shown in Figure 4.6 (b). We deduce that the difference between Figure 4.5 (b) and Figure 4.6 (b) arises from the difference in models: the Brownian heat engine has a continuous and infinite-number of energy levels, while the energy levels in single-qubit heat engine are discrete and finite. Our numerical calculation shows that the value of $\mathcal{W}/\mathcal{L}$ is always finite, which implies that the large power and large efficiency cannot be obtained simultaneously, similarly to the Brownian heat engine.

We next look at the quantum case, i.e., $\varepsilon \neq 0$. We can see that our method works well as shown in Figures 4.7. As indicated in Figure 4.7, the quantum coherence factor ε automatically provides a constraint on $\mathcal{W}$ when maximizing $\mathcal{W}/\mathcal{L}$, which reduces the performance of the heat engine. This is consistent with Brander and Saito's conclusion for the example cycle they introduced [58]. For the available λ-T plane, our numerical algorithm clarifies the best heat engine cycle even in the quantum case.

4.6 Method: machine learning

We have obtained a nice performance of our numerical method based on the GA so far, but the disadvantages of the method are obvious. First, the setup above can only deal with the problems with geometrical quantities (for example, the quantities $\mathcal{W}$ and $\mathcal{L}$). However, it cannot deal with the problems which have time-dependent quantities (for example, the state ρ_t). Thanks to the quasi-static assumption, the quantity $\mathcal{W}/\mathcal{L}$ is only depend on the

geometric property of the cycle and not related to the time-dependent property of the driving parameters. Second, in our setup, we notice that the discontinuity on the point A and B of the cycle may induce numerical errors when calculating the $\mathcal{W}$ and $\mathcal{L}$. Third, we constrain the quasi-static work to avoid trivial solutions in our setup. However, we also wish to avoid too many constraints for a better solution. Keeping these three disadvantages in mind, in this section, we propose a general method based on ANNs. Our method can avoid these disadvantages and be used to achieve our two goals.

The basic idea of our framework is to use an ANN to parametrize a series of cycles, and the optimal cycle can be obtained by optimizing these parameters so that the objective function in (4.15) and (4.16) can be maximized.

The difficulty is that how to describe a thermodynamic cycle. For simplicity, we put a constrain on the cycle like above: the cycle should be a simple closed curve. Hence, for an arbitrary point O surrounded by the cycle C in the plane of (λ, T), we apply the polar coordinate (θ, r) to describe the cycle C as shown in the Figure 4.8 (a). Here the axis r represents the Euclidean's distance from the center O to an arbitrary point $A : (\lambda, T)$ on the cycle, and θ is the included angle between the vector $\overrightarrow{OA}$ and the λ-axis. The coordinate transformation from (θ, r) to (λ, T) is

$$\begin{cases} \lambda = \lambda_0 + r\cos\theta \\ T = T_0 + r\sin\theta \end{cases}, \tag{4.47}$$

where the coordinate of the point O is (λ_0, T_0).

We impose the following condition for r of the cycle C for simplicity. i) The distance r is non-negative. ii) The distance r is a single-valued function with respect to θ. iii) The function $r(\theta)$ is a periodic function with period 2π, i.e., $r(\theta) = r(\theta + 2\pi)$. iv) The cycle C is limited in the square domain D:

$$D = \{(\lambda, T) | \lambda \in [\lambda_m, \lambda_m + 2a] \text{ and } T \in [T_m, T_m + 2a]\} \tag{4.48}$$

The first two condition ensures the cycle C does not have intersections, and the third condition ensures the curve represented by r is a cycle. The last condition is to avoid infinite values of the parameters.

With the above conditions, one can parametrize the function $r(\theta)$ by a ANN, and by optimizing these parameters, the problems (4.15) and (4.16) can be solved. Practically, the problems (4.15) and (4.16) are slightly different with each other.

4.6.1 Setup for the near quasi-static case

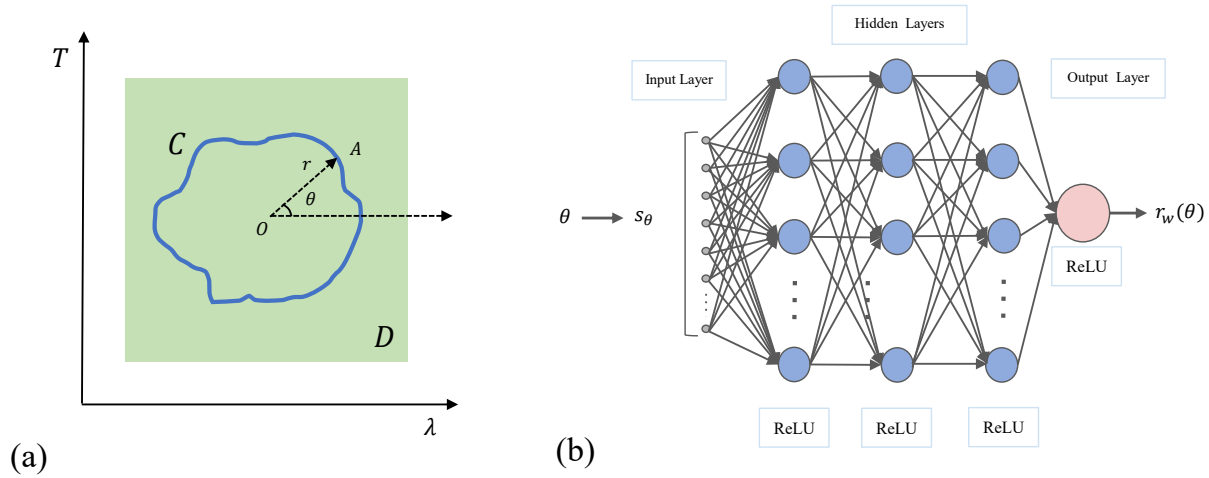

Figure 4.8: (a) Schematic graph of the thermodynamic cycle for the near-adiabatic case. (b) The architecture of the neural network for the near-adiabatic case.

.

Here, we first deal with the problem (4.15). We parametrize $r(\theta)$ to $r_{\boldsymbol{w}}(\theta)$ by the ANN shown in the Figure 4.8 (b).

One needs to notice that the architecture of the ANN should be carefully designed so that the output of the ANN satisfies the periodic condition. We consider the strategy that we transform the input θ to a vector $\boldsymbol{s}_\theta$ with M number of Fourier orders:

$$\boldsymbol{s}_\theta = (1, \cos\theta, \cos 2\theta, ..., \cos M\theta, \sin\theta, \sin 2\theta, ..., \sin M\theta)\,. \tag{4.49}$$

Here the number M is a non-negative integer. Because each element of $\boldsymbol{s}_\theta$ are the function of θ with period 2π, if we input these into the ANN, the output $r_{\boldsymbol{w}}(\theta)$ can also have a period 2π. The architecture of the ANN is shown in Figure 4.8 (b).

To optimize the parameters $\boldsymbol{w}$ of the ANN, we define the loss function $L(\boldsymbol{w})$ according to the objective function in (4.15) with penal terms which limits the cycle C is in the given domain D, i.e.,

$$\begin{aligned} L(\boldsymbol{w}) = &- \frac{\mathcal{W}}{\mathcal{L}} \\ &+ \zeta \sum_i \max(0, -r^i_{\boldsymbol{w}}) \\ &+ \zeta \sum_i \max(0, T_m - T^i_{\boldsymbol{w}}) + \zeta \sum_i \max(0, T^i_{\boldsymbol{w}} - T_m - 2a) \\ &+ \zeta \sum_i \max(0, \lambda_m - \lambda^i_{\boldsymbol{w}}) + \zeta \sum_i \max(0, \lambda^i_{\boldsymbol{w}} - \lambda_m - 2a)\,. \end{aligned} \tag{4.50}$$

Here the first line is our objective function. The second line is the constraint on $r_w(\theta)$. The third and the forth line are the constraint given by the domain D. The constant ζ is the penal constant which varies from different problems. We choose it to be unity for its nice performance and efficiency in numerical calculation. When proceeding numerical calculation, we divide the range of θ, i.e., $[\theta_r, \theta_r + 2\pi]$ into N intervals with length $2\pi/N$, and hence, we have the discrete $\theta_i = \theta_r + 2\pi i/N$. The term $r^i_{\boldsymbol{w}}$ is the discrete form of $r_{\boldsymbol{w}}(\theta)$, and $T^i_{\boldsymbol{w}}$ and $\lambda^i_{\boldsymbol{w}}$ can be calculated by the coordinate transformation (4.47).

Now, the problem (4.15) is converted to a standard machine learning problem that aims to minimize the loss function $L(w)$. It can be solved by the standard gradient descent method. Moreover, to efficiently optimize the ANN, we randomly choose θ_r according to the Gaussian distribution for every training step. Randomly choosing θ_r can avoid to stuck in the local minimum of the cost function and thus can converge faster than using the same θ_r all the time. After the loss function is converged, the optimal cycle C^* can finally be obtained.

4.6.2 Setup for the finite-time case

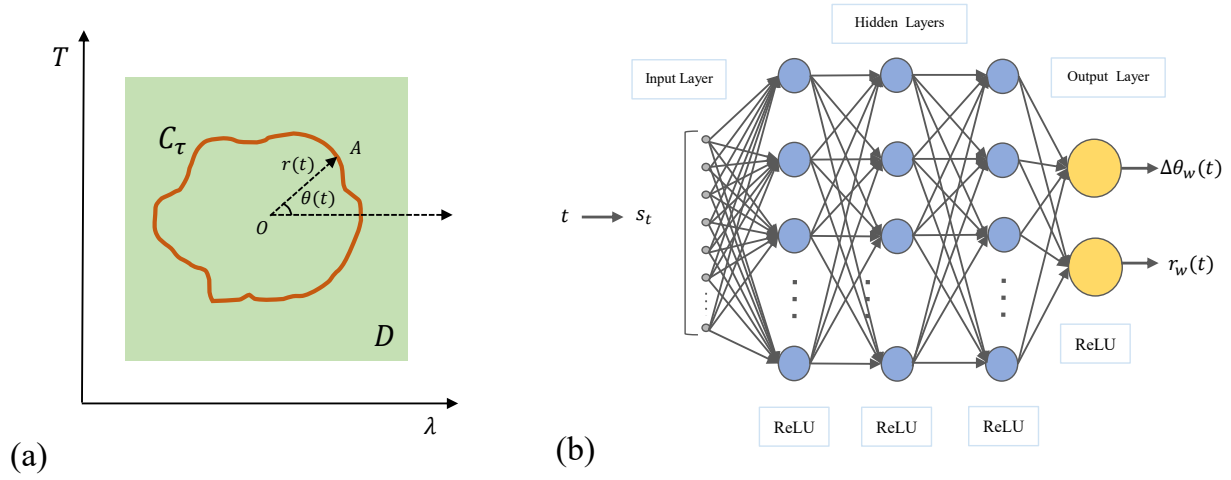

Figure 4.9: (a) Schematic graph of the thermodynamic cycle for the finite-time case. (b) The architecture of the neural network for the finite-time case.

.

Recall that we aim to find the optimal cycle C_t^* to maximize the quantity $P/(1-\eta)$ in the given squared domain D here, as shown in Figure 4.9 (a). The method to find the optimal cycle in the finite case is similar to the one for the near-adiabatic case. The significant differences are illustrated below.

First, the input of the ANN becomes the time t, which leads the output of the ANN $r_{\boldsymbol{w}}(t)$ has a period τ. To satisfy this condition, the input vector $\boldsymbol{s}_t$ in (4.49) becomes:

$$\boldsymbol{s}_t = (1, \cos\frac{2\pi}{\tau}t, \cos\frac{2\pi}{\tau}2t, ..., \cos\frac{2\pi}{\tau}Mt, \sin\frac{2\pi}{\tau}t, \sin\frac{2\pi}{\tau}2t, ..., \sin\frac{2\pi}{\tau}Mt)\,. \tag{4.51}$$

Second, different from the near-adiabatic case, the angle θ is no-longer a constant but a function of t. Meanwhile, to satisfy the cyclic condition, $\theta(t)$ should monotonically increase as t increases in the range of $[t_r, t_r+\tau]$, where t_r is an arbitrary initial time. Moreover, the

distance between the initial and final $\theta(t)$ in one period should be 2π. Hence, $\theta(t)$ satisfies:

$$\theta(t+\tau) - \theta(t) = 2\pi \tag{4.52}$$

Combining the first and the second modification, we renew the ANN shown in Figure 4.9 (b). Different from the previous architecture, it has two outputs: one for the parametrized radius $r_{\boldsymbol{w}}(t)$, the other for the parametrized fluctuation of the angle $\Delta\theta_{\boldsymbol{w}}(t)$. The parametrized $\theta(t)$ can be obtained by

$$\theta_{\boldsymbol{w}}(t) = \Delta\theta_{\boldsymbol{w}}(t) + \frac{2\pi}{\tau}t\,. \tag{4.53}$$

Once the output $(\Delta\theta_w(t), r_w(t))$ is obtained, the corresponding parameters $\lambda_w(t)$ and $T_w(t)$ can be calculated by:

$$\begin{cases} \lambda_w = \lambda_0 + r_w(t)\cos\left[\theta_w(t)\right] \\ T_w = T_0 + r_w(t)\sin\left[\theta_w(t)\right] \end{cases}. \tag{4.54}$$

Thus, the loss function for the finite-time case can be written as,

$$\begin{aligned} L(w) = &-\frac{P}{1-\eta} \\ &+ \zeta\sum_i \max(0, -r_w^i) \\ &+ \zeta\sum_i \max(0, T_m - T_w^i) + \zeta\sum_i \max(0, T_w^i - T_m - 2a) \\ &+ \zeta\sum_i \max(0, \lambda_m - \lambda_w^i) + \zeta\sum_i \max(0, \lambda_w^i - \lambda_m - 2a) \\ &+ \zeta\sum_i \max(0, \theta_i - \theta_{i+1}). \end{aligned} \tag{4.55}$$

Third, because P and η are not geometric quantity, to calculate them, one have to evolve the system to the periodic state ρ_t with the equation of motion (4.30) and (4.31) according to the output of the ANN $(\theta_{\rm w}(t), r_{\rm w}(t))$ for each training steps.

Similar to the near quasi-static case, the problem in the finite-time case can also be solved by the standard gradient descent method.

4.7 Results and discussions 2

This section shows the results of our numerical methods based on ML for the near quasi-static case and the finite-time case of the single-qubit heat engine. We set $\lambda_m = 1.0$ and $T_m = 1.0$. Then the half-side-length a of the region D is set to be 0.5 so that the available regime is D is a square with side length 1. Because $\lambda \geq \lambda_m$, the maximum ε should be less than λ_m to make sure the energy of the system is real number, i.e., $\varepsilon \in [0.0, 1.0]$. In the finite-time case, we numerically solve the differential Eq. (4.30) with the 4th-order Runge-Kutta algorithm. We choose the Fourier order M to be 16 in the following calculations, which is sufficient for our problem (details are shown in Appendix F).

4.7.1 The near quasi-static case

We first investigate the optimal cycles in the near quasi-static case under different values of the quantum coherence factor ε. Figure 4.10 shows the convergence of the loss function of our ML method for different ε. After sufficient steps, the penal terms vanish, and we finally obtain the optimal cycles, which are shown in Figure 4.11. One can notice that as ε increases, the area enclosed by the cycle increases. Simultaneously, the corresponding value of $\mathcal{W}/\mathcal{L}$ decreases ($\mathcal{W}/\mathcal{L} = 0.355$, 0.267 and 0.202 for $\varepsilon = 0.1$, 0.5 and 0.9 respectively). This phenomenon is also noticed in our results with GA method, which is regarded as the trade-off between $\mathcal{W}$ and $\mathcal{L}$. Specifically, we notice that the cycle for large ε (where the quantum effect is strong) cannot path the corner $(1.0, 1.0)$ (lower bound of λ restricted by D) and shifts to the inner part of D. The interpretation of this phenomenon is obvious. Notice that denominator of the quantum contribution $C^{\lambda\lambda}$ of the adiabatic response matrix $\mathcal{R}$ (See Eq. (4.35)) has the term $(\lambda^2 - \varepsilon^2)$, which leads to that $C^{\lambda\lambda}$ will be very large when λ and ε are comparable with each other. This detail on the optimal cycle cannot be shown from our setup with the GA method, which proves that the machine learning method is more general than our GA setup. Once we obtain the results in the near quasi-static case, we can

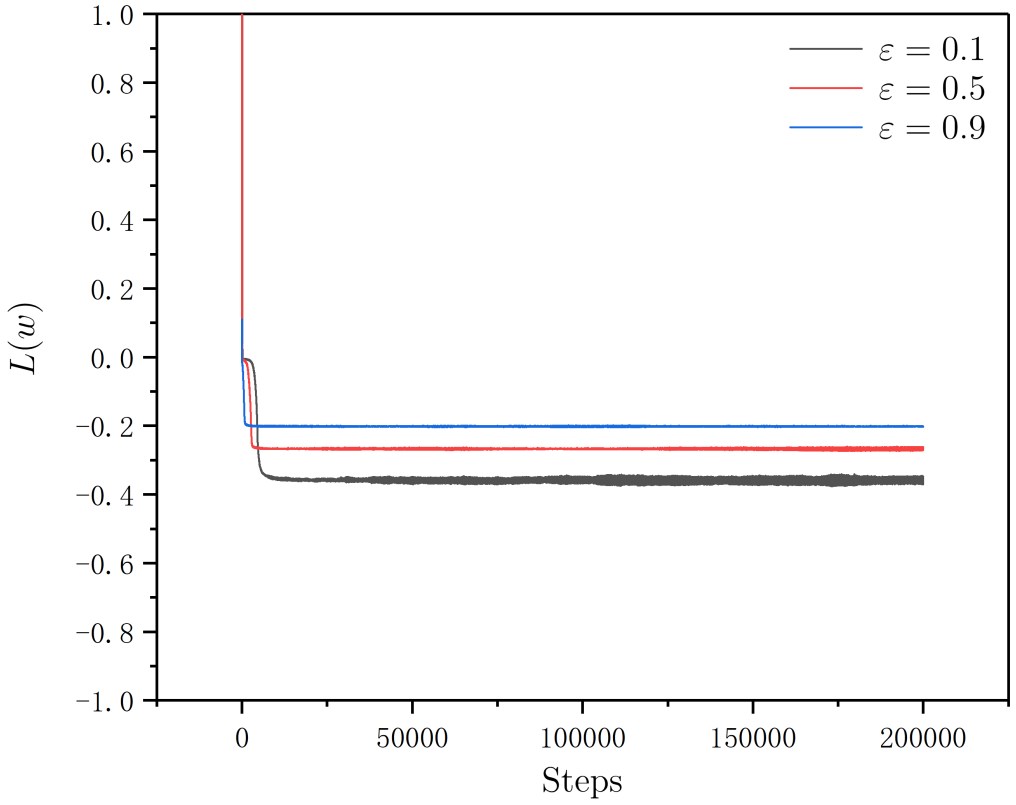

Figure 4.10: The dependence of the loss function $L(w)$ on the number of training steps for the near quasi-static case of the two-level system. Here, the black, red and blue lines respectively represents the results for $\varepsilon = 0.1$, 0.5 and 0.9.

.

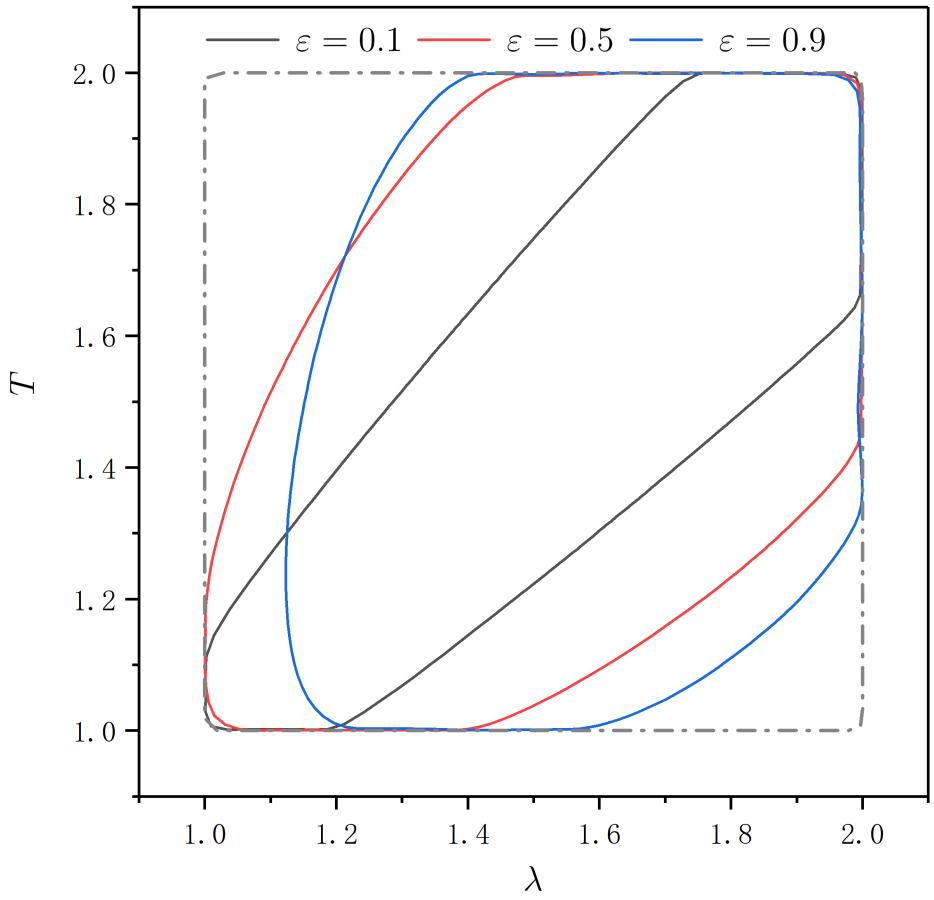

Figure 4.11: The optimal cycles obtained by the machine learning method for the near quasi-static case of the two-level system. Here, the black, red and blue lines respectively represents the results for $\varepsilon = 0.1$, 0.5 and 0.9. The directions of these cycles are counter-clockwise, which ensures the output work W is non-negative.

.

use it as a benchmark for the finite-time case when the period τ is large.

4.7.2 The finite-time case

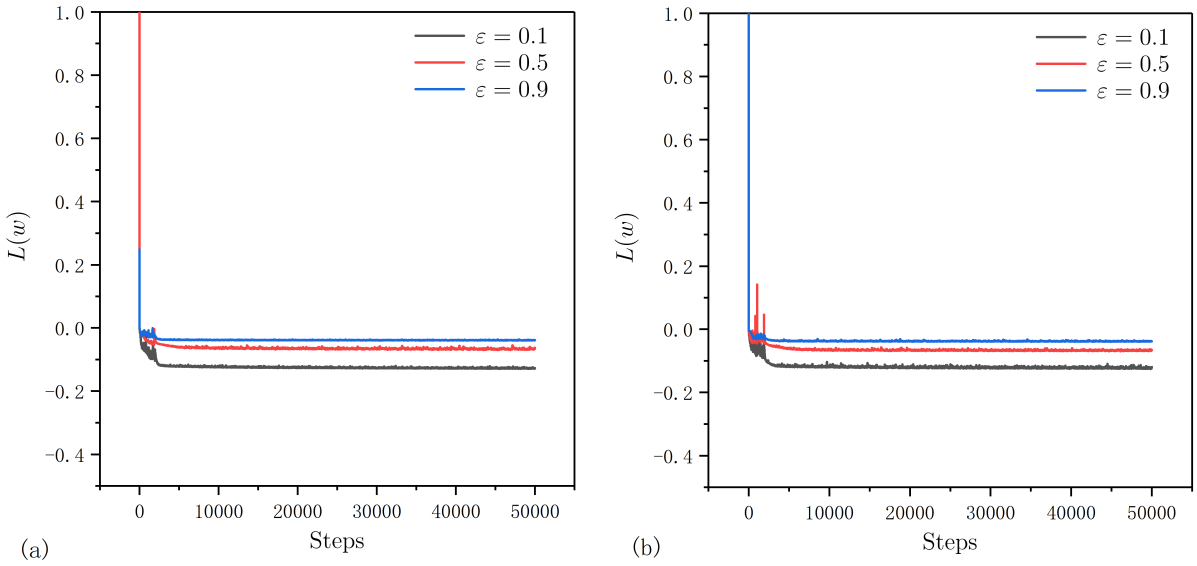

Figure 4.12: The loss function versus the training steps for the finite-time case. (a) and (b) respectively show the results for $\tau = 6$ and 20. Here, the black, red and blue lines respectively represent the results for $\varepsilon = 0.1$, 0.5, and 0.9. The penal terms for all cases are zero after convergence.

.

In this section, we investigate the optimal cycles in the finite-time case under different values of the quantum coherence factor ε and different τ. We first check the convergence of the calculation, shown in Figure 4.12. After sufficient numerical calculations, we obtain the quantity $P/(1-\eta)$, the output power P and the efficiency η for different ε and τ, shown in Figure 4.13. The optimal cycles are also found and shown in Figure 4.14.

We first fix the period τ and investigate the influence of quantum coherence. Similar to the quantity $\mathcal{W}/\mathcal{L}$ of the near quasi-static case, the quantity $P/(1-\eta)$ for the fixed τ also decreases as ε increases. At the same time, the area enclosed by the cycle increases as ε becomes large shown in Figure 4.14. This makes sense because, as shown in Figure 4.13, for a small ε, we have a high efficiency η while the output power P does not change too much.

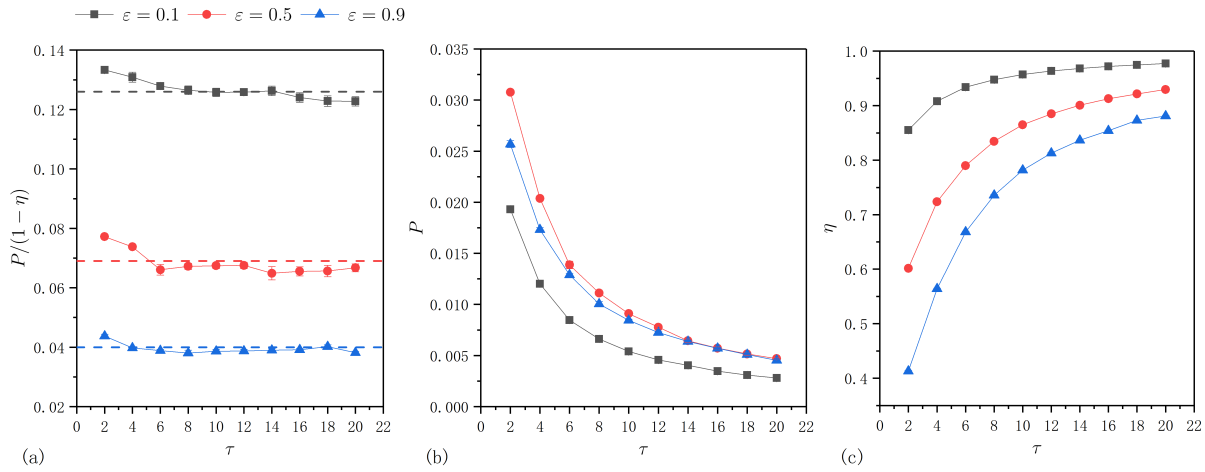

Figure 4.13: The optimal quantities versus the period τ of the cycle for the finite-time case. Here (a), (b), and (c) respectively show the results of the objective function $P/(1-\eta)$, the power P and the efficiency η. Here, the black, red and blue lines respectively represent the results for $\varepsilon = 0.1$, 0.5, and 0.9. In (a), the corresponding dash lines are the results of the near quasi-static case for different εs.

.

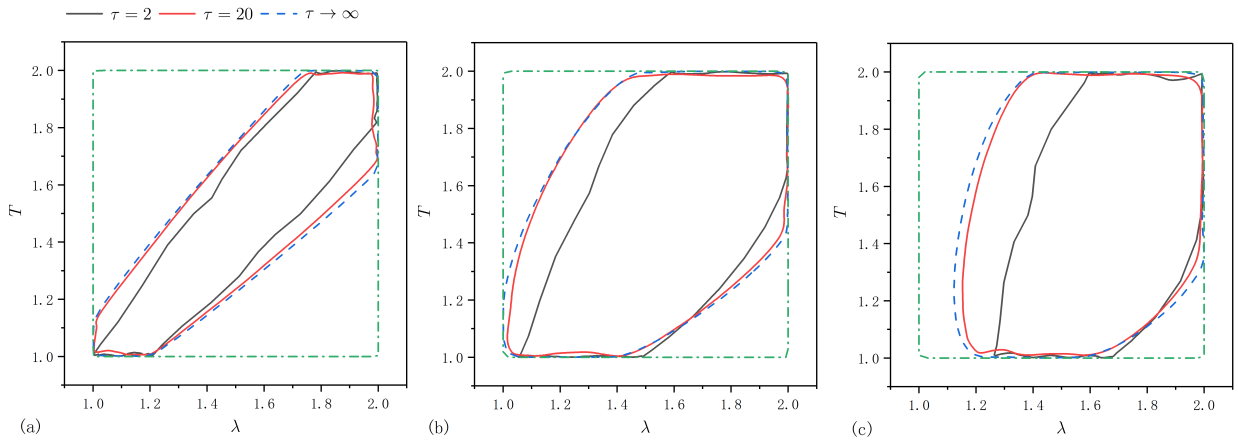

Figure 4.14: The optimal cycles of the single-qubit heat engine at different period τ. Here (a), (b) and (c) respectively show the results for $\varepsilon = 0.1$, 0.5 and 0.9. Here, the dash line in each graph represents the corresponding optimal cycle in the near quasi-static case. Dash-dot lines are the boundary of the given region D.

.

Hence, strong quantum coherence also reduces the performance of the heat engine in the finite-time case.

We then study how the results changes with the period τ. Figure 4.14 suggests that the output power decreases to 0 and the efficiency η approach 1 as τ approaches infinity. However, the ratio $P/(1-\eta)$ approaches a constant corresponding to the near quasi-static case. This behavior quantitatively shows how the driving speed $1/\tau$ affects the heat engine's efficiency and power. This also shows that high efficiency and large output power cannot coexist in this single-qubit heat engine. The cycles we found in Figure 4.14 are optimal for the heat engine at a given period τ considering the power-efficiency trade-off.

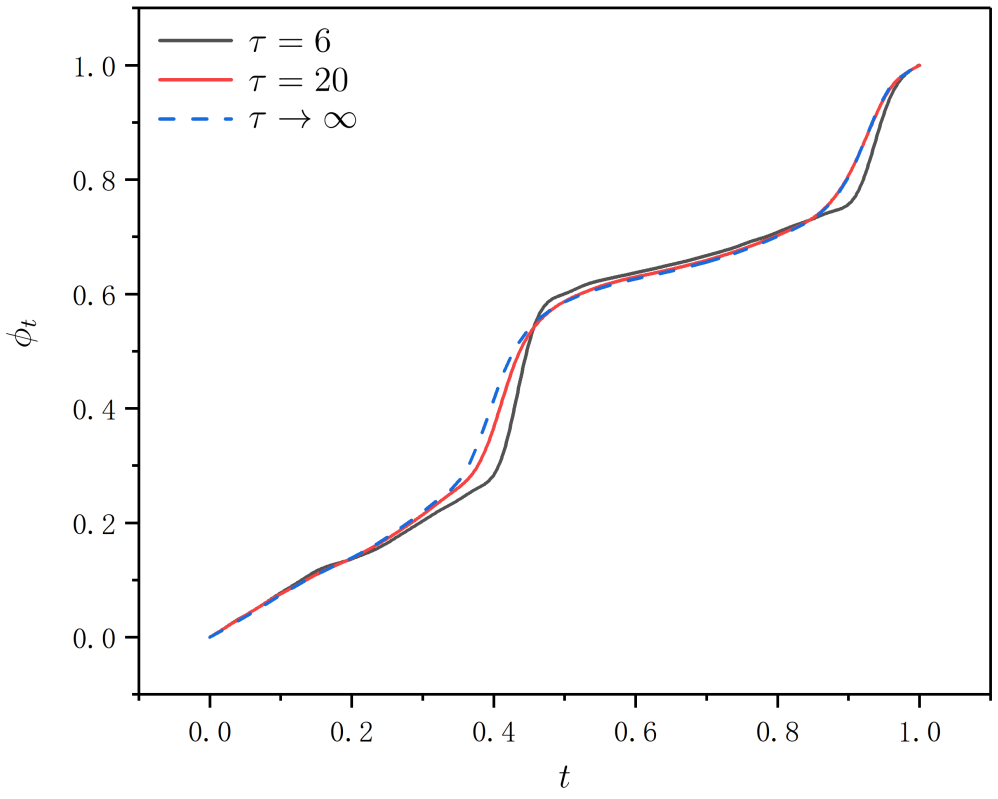

Figure 4.15: The speed function of the optimal cycle of the single-qubit heat engine for $\varepsilon = 0.5$. Here, the dash line in each graph represents the corresponding speed function in the near quasi-static case.

.

In particular, we discuss the speed function for both cases, and the results are shown in Figure 4.15. Here, the speed function for the finite-time case is exactly the time-dependent angle function $\phi_t = \theta_w(t)/2\pi$. As τ becomes large, the finite-time case's speed function gradually approaches the near quasi-static case. As ϕ_t in the quasi-static case is theoretically given by the equality of Eq. (4.11), we use it as the benchmark, which suggests that not only our ML method can find the optimal cycle but also provides the correct speed function.

4.8 Chapter summary

This chapter first reviewed the adiabatic response theory of microscopic heat engines. We described the definition of the optimal thermodynamic cycle. This chapter's two primary goals are then illustrated: finding the optimal thermodynamic cycles in the near quasi-static and finite-time cases. We then propose two numerical optimization methods to deal with these two problems: the method based on the GA and the method based on the ML. Two models driven by slow temperature and slow external forces in a designated domain are considered for the GA method. The optimal thermodynamic cycle for both models under a classical, quasi-classical, and quantum case are successfully obtained by our method. We use the ANN as a function approximator for the ML method to approximate the optimal cycle. It is more general than the GA method and can be applied to both problems. These numerical results successfully find the optimal cycles that simultaneously maximize the output power and efficiency. However, our results show that large power and high efficiency cannot be obtained simultaneously.

Chapter5

Conclusion and Perspectives

This thesis investigates the applications of machine learning approaches to optimization problems in nonequilibrium thermodynamics. Benefiting from the generalization property of the machine learning approaches, we successfully provide two general numerical methods for two different kinds of optimization problems.

First, we study the acceleration of the transitions between thermodynamic states by the reinforcement learning approach. Both classical and quantum two-level systems are considered as our example models. For the classical case, by comparing the results of the reinforcement learning approach with theoretical results, we show that our numerical method is reliable for this optimization problem. When searching for these optimal protocols, we find that: i) Jumps at the initial and final time on the protocol of controllable systems appear because of the particular type of variational differential equations. While it has been clarified in classical cases, theoretically, it is still worthwhile to clarify the jumps in quantum cases. ii) Accelerating transitions between two equilibrium states require extra dissipated energy, increasing entropy production. There is an upper bound on the entropy production beyond which the target state cannot reach a given finite period. This phenomenon is related to the speed limit in classical and quantum systems. While the speed limit is not a tight bound, a tight bound is further investigated. iii) Our results show that the quantum system needs to produce more energy to overcome the quantum effect and reach the target state than the classical two-level system.

Second, we propose two numerical optimization methods based on the genetic algorithm and machine learning approach to find the optimal thermodynamic cycle of a microscopic heat engine. The cycle is optimal because it can simultaneously maximize the heat engine's power and efficiency. The main difficulty we face is how to represent the cycle in a restricted domain. We consider a particular parametrization setup and apply the genetic algorithm to solve it successfully under the adiabatic response regime. However, this method cannot deal with optimal cycles in the finite-time case. We apply the machine learning approach by using artificial neural networks as a function approximator. Under this framework, optimal cycles in both two cases are found. We find that the optimal cycle can maximize the power and the efficiency simultaneously, but large power and high efficiency cannot coexist.

For further works, numerical methods proposed in this thesis can be applied to many other similar problems. For the first topic we discussed, the advantage of our reinforcement learning method is that it can be generalized to other optimal-protocol-seeking problems, especially shortcuts between equilibrium states with maximizing or minimizing some other physical quantities. We believe our method may also be applied to find the optimal protocols of transitions between two nonequilibrium steady states. Also, this method can be applied to discuss the problems of finite-time Landauer's principle, which focuses on the minimum energy cost of erasing a state in finite-time [107]. Moreover, our method is proved to be successful in Markovian dynamics. We wish to extend the framework for the systems with non-Markovian dynamics [108]. Our method can provide a general framework for the second topic to find optimal cycles in different problems. One of the relevant perspectives is finding the power-efficiency trade-off relations and optimal thermodynamic cycles in non-Markovian microscopic heat engines.

AppendixA

Details of the reinforcement learning method for Chapter 2

We provide here some technical details and hyperparameters used in Chapter 2 [89].

The input of our neural network is a vector including the elements of the density matrix for the quantum case, while the input vector for the classical case is a two dimensional vector with p = (p_-, p_+). Our neural network has 3 hidden layers and each layer contains 100 neurons with Rectified Linear Unit activation function. We set the activation function of the final layer to be linear layer to avoid the disappearance of gradient when calculating. Neurons in our neural network are fully-connected. The optimal parameters in our neural network can be obtained by back-propagation method. Here, we use the Adam optimizer [109] for its fast and stable performance. The learning rate α is set to be 10^{-5} for all three examples in this work. The covariance matrix Σ of the parametrized policy is a scalar and we set it to be 0.01 for the classical case. It is a two-dimensional matrix,

$$\Sigma_a = \begin{pmatrix} \sigma_a & 0 \\ 0 & \sigma_a \end{pmatrix} \tag{A.1}$$

for the quantum case. Especially, We set $\sigma_a = 0.01$ for Quantum 1 and $\sigma_a = 0.1$ for Quantum 2.

We also subtracted a baseline to the expected return for each step of the RL episodes, i.e., $G_j \leftarrow G_j - w$, where w is initially a random number and can be renewed by $\Delta w = \alpha_w G_j$

in each step of the RL episodes with the learning rate $\alpha_w = 10^{-4}$. This method usually leads to better convergence properties [88]. We think the optimal solution is obtained when the averaging total reward in 10^4 episodes almost does not change (< 0.01). We average the corresponding quantities in the last 10^4 steps to avoid the influences from the fluctuations when calculating. The number of time intervals N for all three examples is set to be 50, and we assume that the relaxation time in the reservoir is much shorter than the typical time scale of the system's parameter change, which ensures validation of the GKLS equation.

All calculations of neural networks are based on Pytorch tools [110]. The evolution of quantum states and the numerical calculation of the GKLS equation are calculated by QuTip toolbox in Python [111].

AppendixB

Calculation of the upper bound of the final state in the classical two-level system

We provide the details of the calculation of the upper bound of final state $p_{\mathrm{eq,f}}$ in the classical two-level system [101].

As illustrated in the main text, we start from the integration form of Euler-Lagrange equation:

$$\mathcal{L}(p,\dot{p}) - \dot{p}\frac{\partial \mathcal{L}}{\partial \dot{p}} = -K\,. \tag{B.1}$$

Here, according to the definition of entropy production, we have

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \dot{p}} =& \frac{\partial}{\partial \dot{p}}\left\{\dot{p}\ln\frac{(1-p)(p+\dot{p})}{p[1-(p+\dot{p})]}\right\} \\
=& \ln\frac{(1-p)(p+\dot{p})}{p[1-(p+\dot{p})]} \\
&+ \dot{p}\cdot\frac{p[1-(p+\dot{p})]}{(1-p)(p+\dot{p})}\cdot\frac{(1-p)p[1-(p+\dot{p})]-(1-p)(p+\dot{p})(-p)}{\{p[1-(p+\dot{p})]\}^2} \\
=& \ln\frac{(1-p)(p+\dot{p})}{p[1-(p+\dot{p})]} + \dot{p}\cdot\frac{1}{(1-p)(p+\dot{p})}\cdot\frac{(1-p)p}{p[1-(p+\dot{p})]} \\
=& \ln\frac{(1-p)(p+\dot{p})}{p[1-(p+\dot{p})]} + \frac{\dot{p}}{(p+\dot{p})[1-(p+\dot{p})]}\,.
\end{aligned} \tag{B.2}$$

Plugging (B.2) into (B.1), we obtain

$$\frac{\dot{p}^2}{(p+\dot{p})[1-(p+\dot{p})]} = K\,. \tag{B.3}$$

Because the entropy production rate $\dot{\Sigma}$ is a real number, the term $\frac{(p+\dot{p})}{1-(p+\dot{p})}$ must be a positive value. Hence, $K \geq 0$.

After simplification, we have

$$(K+1)\dot{p}^2 + K(2p-1)\dot{p} - Kp(1-p) = 0\,. \tag{B.4}$$

The solution can be expressed as,

$$\dot{p} = \frac{(1-2p)K \pm \sqrt{\Delta}}{2(K+1)}\,, \tag{B.5}$$

where $\Delta = K^2 + 4Kp(1-p)$.

Now we explain the physical meaning of K. By inserting the master equation (**??**) into Eq. (B.3), we obtain

$$\begin{aligned} &\frac{(p-\omega_t)^2}{\omega_t(1-\omega_t)} = K \\ \Rightarrow\quad &p = \omega_t(1 \pm \sqrt{\frac{1-\omega_t}{\omega_t}}\sqrt{K})\,. \end{aligned} \tag{B.6}$$

Because ω_t is the instantaneous equilibrium state, we notice that the system approaches to the quasistatic limit as K approaches to 0. We think that K is a quantity that measures how far the state of the system varies from the quasistatic limit. In our setup, we limit $\lambda(t) \geq 0$. Hence, we have the relation $p_{\mathrm{eq,f}} > p_{\mathrm{eq,i}} \geq 0.5$. Thus, the equation (B.5) and (B.6) takes the "+" branch. The exact value of K can be calculated by integrating both sides of Eq. (B.1):

$$\begin{aligned} &\int_{p_{\mathrm{eq,i}}}^{p_{\mathrm{eq,f}}} \mathrm{d}p\, \frac{2(K+1)}{(1-2p)K + \sqrt{\Delta}} = \tau \\ \Rightarrow\quad &F(p_{\mathrm{eq,f}}, K) - F(p_{\mathrm{eq,i}}, K) = \tau\,, \end{aligned} \tag{B.7}$$

where

$$\begin{aligned} F(p,K) =& \int \mathrm{d}p \, \frac{2(K+1)}{(1-2p)K+\sqrt{\Delta}} \\ =& -\ln(1-p) + \frac{1}{2}\ln\left[\frac{2(1-p)+K+\sqrt{\Delta}}{2p+K+\sqrt{\Delta}}\right] + \frac{1}{\sqrt{K}}\arctan\left[\frac{(1-2p)\sqrt{K}}{\sqrt{\Delta}}\right]. \end{aligned} \tag{B.8}$$

The value of K is determined once $p_{\mathrm{eq,i}}$, $p_{\mathrm{eq,f}}$ and τ are fixed.

To find the maximum $p_{\mathrm{eq,f}}$ when given $p_{\mathrm{eq,i}}$ and τ, we first calculate the derivatives of K and $p_{\mathrm{eq,f}}$.

We define $G(p_{\mathrm{eq,f}}, K | p_{\mathrm{eq,i}}, \tau) = F(p_{\mathrm{eq,f}}, K) - F(p_{\mathrm{eq,i}}, K) - \tau$. The differential of G is calculated by

$$\mathrm{d}G = \frac{\partial G}{\partial p_{\mathrm{eq,f}}}\mathrm{d}p_{\mathrm{eq,f}} + \frac{\partial G}{\partial K}\mathrm{d}K \tag{B.9}$$

Here, we have

$$\frac{\partial G}{\partial p_{\mathrm{eq,f}}} = \frac{\partial F}{\partial p_{\mathrm{eq,f}}} = \frac{2(K+1)}{(1-2p_{\mathrm{eq,f}})K+\sqrt{\Delta_{\mathrm{f}}}}, \tag{B.10}$$

and

$$\begin{aligned} \frac{\partial G}{\partial K} =& \frac{\partial}{\partial K}\int_{p_{\mathrm{eq,i}}}^{p_{\mathrm{eq,f}}} \mathrm{d}p \, F(p,K) \\ =& \int_{p_{\mathrm{eq,i}}}^{p_{\mathrm{eq,f}}} \mathrm{d}p \, \frac{\partial F}{\partial K} \\ =& \int_{p_{\mathrm{eq,i}}}^{p_{\mathrm{eq,f}}} \mathrm{d}p \, \frac{\partial}{\partial K}\left[\frac{2(K+1)}{(1-2p)K+\sqrt{\Delta}}\right] \\ =& -\int_{p_{\mathrm{eq,i}}}^{p_{\mathrm{eq,f}}} \mathrm{d}p \, \frac{1}{K\sqrt{\Delta}} \\ =& \frac{1}{2K\sqrt{K}}\left\{\arctan\left[\frac{(2p_{\mathrm{eq,i}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{i}}}}\right] - \arctan\left[\frac{(2p_{\mathrm{eq,f}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{f}}}}\right]\right\}. \end{aligned} \tag{B.11}$$

Hence, the derivative of $p_{\mathrm{eq,f}}$ with respect to K is then obtained,

$$\begin{aligned}\frac{\mathrm{d}p_{\mathrm{eq,f}}}{\mathrm{d}K} =& -\frac{\frac{\partial G}{\partial K}}{\frac{\partial G}{\partial p_{\mathrm{eq,f}}}} \\ =& \frac{(1-2p_{\mathrm{eq,f}})K+\sqrt{\Delta_{\mathrm{f}}}}{4K\sqrt{K}(K+1)} \\ & \times\left\{\arctan\left[\frac{(2p_{\mathrm{eq,f}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{f}}}}\right]-\arctan\left[\frac{(2p_{\mathrm{eq,i}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{i}}}}\right]\right\}.\end{aligned} \tag{B.12}$$

Because

$$\begin{aligned}(1-2p_{\mathrm{eq,f}})K+\sqrt{\Delta_{\mathrm{f}}} =& \frac{(\sqrt{\Delta_{\mathrm{f}}}-(2p_{\mathrm{eq,f}}-1)K)(\sqrt{\Delta_{\mathrm{f}}}+(2p_{\mathrm{eq,f}}-1)K)}{\sqrt{\Delta_{\mathrm{f}}}+(2p_{\mathrm{eq,f}}-1)K} \\ =& \frac{\Delta_{\mathrm{f}}-(2p_{\mathrm{eq,f}}-1)^2K^2}{\sqrt{\Delta_{\mathrm{f}}}+(2p_{\mathrm{eq,f}}-1)K} \\ =& \frac{4p_{\mathrm{eq,f}}(1-p_{\mathrm{eq,f}})K(K+1)}{\sqrt{\Delta_{\mathrm{f}}}+(2p_{\mathrm{eq,f}}-1)K}\geq 0\,,\end{aligned} \tag{B.13}$$

and

$$\begin{aligned}&\frac{(2p_{\mathrm{eq,f}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{f}}}}-\frac{(2p_{\mathrm{eq,i}}-1)\sqrt{K}}{\sqrt{\Delta_{\mathrm{i}}}} \\ =&\sqrt{\frac{k}{\Delta_{\mathrm{i}}\Delta_{\mathrm{f}}}}\left[(2p_{\mathrm{eq,f}}-1)\Delta_{\mathrm{i}}-(2p_{\mathrm{eq,i}}-1)\Delta_{\mathrm{f}}\right] \\ =&\sqrt{\frac{k}{\Delta_{\mathrm{i}}\Delta_{\mathrm{f}}}}\frac{[(2p_{\mathrm{eq,f}}-1)\Delta_{\mathrm{i}}-(2p_{\mathrm{eq,i}}-1)\Delta_{\mathrm{f}}][(2p_{\mathrm{eq,f}}-1)\Delta_{\mathrm{i}}+(2p_{\mathrm{eq,i}}-1)\Delta_{\mathrm{f}}]}{[(2p_{\mathrm{eq,f}}-1)\Delta_{\mathrm{i}}+(2p_{\mathrm{eq,i}}-1)\Delta_{\mathrm{f}}]} \\ =&\sqrt{\frac{k}{\Delta_{\mathrm{i}}\Delta_{\mathrm{f}}}}\frac{[4p_{\mathrm{eq,i}}(1-p_{\mathrm{eq,i}})-4p_{\mathrm{eq,f}}(1-p_{\mathrm{eq,f}})](K^2+1)}{[(2p_{\mathrm{eq,f}}-1)\Delta_{\mathrm{i}}+(2p_{\mathrm{eq,i}}-1)\Delta_{\mathrm{f}}]}\geq 0\,,\end{aligned} \tag{B.14}$$

the derivative of $p_{\mathrm{eq,f}}$ with respect to K is non-negative. Thus, $p_{\mathrm{eq,f}}$ monotonically increases as K increases. When K approaches to ∞, we have the upper bound of $p_{\mathrm{eq,f}}$

$$\begin{aligned}&\lim_{K\to\infty}F(p_{\mathrm{eq,f}},K)-\lim_{K\to\infty}F(p_{\mathrm{eq,i}},K)=\tau \\ \Rightarrow\quad &\ln(1-p_{\mathrm{eq,i}})-\ln(1-p_{\mathrm{eq,f}})=\tau \\ \Rightarrow\quad &p_{\mathrm{eq,f}}=1-(1-p_{\mathrm{eq,i}})e^{-\tau}\end{aligned} \tag{B.15}$$

In our setup, we set $p_{\mathrm{eq,i}} = 0.5$ and $\tau = 1$. Hence, $\max\left(p_{\mathrm{eq,f}}\right) = 1 - 1/2e \approx 0.82$. The corresponding turning point of λ_{f} is

$$\lambda_{\mathrm{f}} = -\ln\left[\frac{1}{p_{\mathrm{eq,f}}} - 1\right] \approx 1.49\,. \tag{B.16}$$

The results given by our RL method is very close to the theoretical one, where $max(p_{\mathrm{eq,f}}) \approx 0.80$ and $\lambda_{\mathrm{f}} \approx 1.38$.

AppendixC

Adiabatic response theory on classical overdamped Brownian particle heat engine

We introduce details on the classical overdamped Brownian particle heat engine [101]. Specifically, we show how its finite-time thermodynamics can be analyzed under the adiabatic response regime.

As illustrated in the main text, the probability distribution $\rho(x,t)$ to find the Brownian particle at position x at time t is governed by the corresponding Fokker-Planck equation.

Our goal is to find the first two orders of adiabatic expansion on the generalized pressure f_t^λ and the entropy f_t^T. Consider the quasi-adiabatic expansion of $\rho(x,t)$ (by the order of speed $1/\tau$),

$$\rho(x,t) = \rho_{eq} + \delta\rho + o(1/\tau^2) \approx \rho_{eq} + \delta\rho \tag{C.1}$$

Supposing that we denote $\omega = \lambda/2T$, we obtain ρ_{eq}, the instantaneous equilibrium state,

$$\rho_{eq}(x,t) = \sqrt{\frac{\omega}{\pi}} e^{-\omega x^2}. \tag{C.2}$$

Thus, we have this relationship by the normalization property of ρ,

$$\int_{-\infty}^{+\infty} dx \delta\rho = 0 \tag{C.3}$$

Inserting Eq. (C.1) into the master equation and eliminating the high-order small term, we obtain

$$\frac{\partial \rho_{eq}}{\partial t} = \mathbb{L}_t \delta\rho, \tag{C.4}$$

To solve Eq. (C.4), we do the following Fourier transformation on both sides,

$$\widetilde{\delta\rho}(k,t) = \mathcal{F}[\delta\rho(x,t)] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{+\infty} dx e^{-ikx} \delta\rho(x,t). \tag{C.5}$$

Then, Eq. (C.4) becomes,

$$\mathcal{F}\left[\frac{\partial \rho_{eq}}{\partial t}\right] = -\lambda(t) k \frac{\partial \widetilde{\delta\rho}(k,t)}{\partial k} - T(t) k^2 \widetilde{\delta\rho}(k,t), \tag{C.6}$$

where

$$\begin{aligned} \mathcal{F}\left[\frac{\partial \rho_{eq}}{\partial t}\right] &= \mathcal{F}\left[\dot{\omega}\sqrt{\frac{\omega}{\pi}} e^{-\omega x^2}\left(\frac{1}{2\omega} - x^2\right)\right] \\ &= \frac{\dot{\omega}}{4\sqrt{2\pi}\omega^2} k^2 e^{-\frac{k^2}{4\omega}} \end{aligned} \tag{C.7}$$

Thus, we obtain a differential equation of first-order of k of $\delta\rho(k,t)$,

$$\frac{\partial \widetilde{\delta\rho}(k,t)}{\partial k} + \frac{k}{2w}\widetilde{\delta\rho}(k,t) + \frac{\dot{\omega}}{4\sqrt{2\pi}\omega^2\lambda} k e^{-\frac{k^2}{4\omega}} = 0. \tag{C.8}$$

We can directly solve it,

$$\begin{aligned} \widetilde{\delta\rho}(k,t) &= C e^{-\int dk \frac{k}{2\omega}} \\ &\quad - e^{-\int dk \frac{k}{2\omega}} \int dk \frac{\dot{\omega}}{4\sqrt{2\pi}\omega^2\lambda} k e^{-\frac{k^2}{4\omega}} e^{\int dk \frac{k}{2\omega}} \\ &= C e^{-\frac{k^2}{4\omega}} - \frac{\dot{\omega}}{8\sqrt{2\pi}\omega^2\lambda} k^2 e^{-\frac{k^2}{4\omega}}. \end{aligned} \tag{C.9}$$

By using inverse Fourier Transform on $\widetilde{\delta\rho}(k,t)$, we obtain,

$$\begin{aligned}\delta\rho(x,t) &= \mathcal{F}^{-1}[\widetilde{\delta\rho}(k,t)] \\ &= C\sqrt{2\omega}e^{-\omega x^2} + \frac{\dot{\omega}}{4\omega\lambda}\sqrt{\frac{\omega}{\pi}}e^{-\omega x^2}(2\omega x^2-1)\end{aligned} \tag{C.10}$$

Using property (C.3) of $\delta\rho(x,t)$, we can determine $C = 0$. Therefore, first-order of adiabatic expansion of $\rho(x,t)$ is,

$$\delta\rho(x,t) = \frac{\dot{\omega}}{4\omega\lambda}\rho_{eq}(2\omega x^2-1) \tag{C.11}$$

Then, the generalized pressure f_t^λ and the entropy f_t^T can be calculated. The equilibrium part is,

$$\begin{aligned}f_{eq,t}^\lambda &= \langle -\frac{\partial H}{\partial\lambda}\rangle = -\langle x^2\rangle/2 = -\frac{T}{2\lambda} \\ f_{eq,t}^T &= -\langle \ln\rho(x,t)\rangle = \frac{1}{2}\left(1+\ln\frac{2\pi T}{\lambda}\right)\end{aligned} \tag{C.12}$$

Therefore, we have

$$\begin{aligned}f_t^\lambda &= \langle -\frac{\partial H}{\partial\lambda}\rangle \approx -\frac{1}{2}\int dx(\rho_{eq}+\delta\rho)x^2 \\ &= f_{eq,t}^\lambda - \frac{T}{4\lambda^3}\dot{\lambda} + \frac{1}{4\lambda^2}\dot{T}\end{aligned} \tag{C.13}$$

$$\begin{aligned}f_t^T &= -\langle\ln\rho\rangle \approx \int dx(\rho_{eq}+\delta\rho)\ln\rho_{eq} \\ &= f_{eq,t}^T + \frac{1}{4\lambda^2}\dot{\lambda} - \frac{1}{4\lambda T}\dot{T}.\end{aligned} \tag{C.14}$$

Thus, the adiabatic response matrix can be obtained,

$$\mathcal{R} = \begin{pmatrix}\mathcal{R}^{\lambda\lambda} & \mathcal{R}^{\lambda T} \\ \mathcal{R}^{T\lambda} & \mathcal{R}^{TT}\end{pmatrix} = \frac{1}{4\lambda^2}\begin{pmatrix}-T/\lambda & 1 \\ 1 & -\lambda/T\end{pmatrix} \tag{C.15}$$

Then, the corresponding adiabatic work $\mathcal{W}$ is expressed as

$$\mathcal{W} = \oint_c d\lambda f_{eq,t}^\lambda = -\int_0^{\Delta t} dt\frac{T}{2\lambda}\dot{\lambda} \tag{C.16}$$

and the thermodynamic length $\mathcal{L}$ is denoted by

$$\begin{aligned}\mathcal{L} &= \oint_C \sqrt{-[\mathcal{R}^{\lambda\lambda} d\lambda d\lambda + (\mathcal{R}^{\lambda T} + \mathcal{R}^{T\lambda}) d\lambda dT + \mathcal{R}^{TT} dT dT]} \\ &= \oint_C \sqrt{\frac{T}{4\lambda^3}} \left| d\lambda - \frac{\lambda}{T} dT \right|\end{aligned} \tag{C.17}$$

AppendixD

Adiabatic response theory on single-qubit heat engine

In this section, we derive the expressions of thermodynamic length and quasi-static work of single-qubit heat engine under the framework of adiabatic response theory, which has been discussed in the Ref. [58] and [101]

As illustrated before, the time evolution of the density matrix ρ_t of the working medium is described by the Lindblad equation,

$$\partial_t \rho_t = \mathbb{L}_t \rho_t \,. \tag{D.1}$$

Here $\mathbb{L}_t$ is the Lindblad generator and it is composed of two parts: the unitary evolution term of the bare system and the influence of the environment, i.e.,

$$\mathbb{L}_t X \equiv (-i\mathrm{H}_t + \mathrm{D}_t)X \,, \tag{D.2}$$

where

$$\begin{aligned} &\mathrm{H}_t X \equiv [H, X]/\hbar \quad \text{and} \\ &\mathrm{D}_t X \equiv \sum_\sigma \left([V^\sigma X, V^{\sigma\dagger}] + [V^\sigma, X V^{\sigma\dagger}]\right) \,. \end{aligned} \tag{D.3}$$

According to the result of Ref. [58], the state ρ_t can be determined by an adiabatic pertur-

bation theory. By inserting the ansatz

$$\rho_t = \rho_{eq,t} + \delta\rho_t \tag{D.4}$$

into (D.3), after simplification, the adiabatic response coefficients can be expressed by

$$\mathcal{R}^{\mu\nu} = -\frac{1}{T}\int_0^\infty ds \left\langle \left(F_t^\nu - f_{eq,t}^\nu\right) \middle| \exp\left[(i\mathrm{H}_t + \mathrm{D}_t^\ddagger)s\right] F_t^\mu \right\rangle . \tag{D.5}$$

Here $\delta\rho_t$ is the first order of $1/\tau$,

$$\delta\rho_t = \int_0^1 dx \rho_{eq,t}^{1-x} \xi_t \rho_{eq,t}^x , \tag{D.6}$$

and $\langle X|Y\rangle$ represents the scalar product between to superoperator X and Y

$$\langle X|Y\rangle \equiv \int_0^1 dx \mathrm{Tr}\left[\rho_{eq,t}^{1-x} X^\dagger \rho_{eq,t}^x Y^\dagger\right] . \tag{D.7}$$

And the corresponding instantaneous thermodynamic forces are expressed as

$$\begin{aligned} F_t^\lambda &\equiv -\partial_\lambda H_\lambda \\ F_t^T &\equiv -\ln \rho_{eq,t} . \end{aligned} \tag{D.8}$$

Based on the above derivation, we can obtain the geometric properties of the single-qubit heat engine.

Firstly, we notice that the Hamiltonian of the two-level system can be represented by the eigenstates $|\phi^+\rangle$ and $|\phi^-\rangle$ of the energy,

$$H_\lambda = \frac{\hbar\Omega\lambda}{2}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) , \tag{D.9}$$

where

$$|\phi^+\rangle = \begin{pmatrix} \sin[\theta/2] \\ -\cos[\theta/2] \end{pmatrix} \text{ and } |\phi^-\rangle = \begin{pmatrix} \cos[\theta/2] \\ \sin[\theta/2] \end{pmatrix} \text{ with}$$
$$\tan[\theta/2] = \frac{\varepsilon}{\sqrt{\lambda^2 - \epsilon^2} + \lambda}\,. \tag{D.10}$$

Thus, according to the quantum detailed balance condition, ($[H_\lambda, V^\pm] = \pm\hbar\Omega\lambda V^\pm$, and $\mathrm{Tr}\left[V^\pm V^{\pm\dagger}\right] = \pm\Gamma\Omega\lambda/\left[1 - \exp\left[\mp\hbar\Omega\lambda/T\right]\right]$), the expression of jump operator $V^\pm$ can be obtained,

$$V^\pm = \sqrt{\frac{\pm\Gamma\Omega\lambda}{1 - \exp\left[\mp\hbar\Omega\lambda/T\right]}}|\phi^\pm\rangle\langle\phi^\mp|\,. \tag{D.11}$$

As the consequence of quantum detailed balance condition, superoperators H_t and $\mathrm{D}_t^\ddagger$ (the adjoint of D_t with respect to Hilbert-Schmidt scalar product) share the normalized eigenvectors M^j with real eigenvalues ω^j and non-positive κ^j respectively,

$$\mathrm{H}_t M^j = \omega^j M^j\,,\ \mathrm{D}_t^\ddagger M^j = \kappa^j M^j\,,\ \langle M^j|M^k\rangle = \delta_{jk}\,. \tag{D.12}$$

$\kappa^0 = 0$ and it is the maximum eigenvalue, which corresponds to the unique stationary eigenvector $M^0 = \mathbf{1}$.

These the normalized eigenvectors can be calculated as,

$$\begin{aligned} M^0 &= \mathbf{1}\,, \\ M^z &= \frac{2}{\hbar\Omega\lambda}\cosh\left[\hbar\Omega\lambda/2T\right]H_\lambda + \sinh\left[\hbar\Omega\lambda/2T\right]\mathbf{1} \quad \text{and} \\ M^\pm &= \sqrt{\frac{\hbar\Omega\lambda}{T\tanh\left[\hbar\Omega\lambda/2T\right]}}|\phi^\pm\rangle\langle\phi^\mp|\,. \end{aligned} \tag{D.13}$$

with corresponding eigenvalues,

$$\begin{aligned}
&\kappa^0 = 0, \quad \kappa^z = -2\Gamma\Omega\lambda \coth\left[\hbar\Omega\lambda/2T\right], \quad \kappa^{\pm} = \kappa^z/2\,, \\
&\omega^0 = \omega^z = 0, \quad \omega^{\pm} = \pm\Omega\lambda\,.
\end{aligned} \tag{D.14}$$

Hence, variances of thermodynamic forces can be calculated. For simplicity, we first calculate the generalized pressure. Notice that

$$\begin{aligned}
F_t^{\lambda} =& -\partial_\lambda H_\lambda \\
=& -\frac{\hbar\Omega}{2}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) - \frac{\hbar\Omega\lambda}{2}\frac{\partial}{\partial\lambda}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) \\
=& -\frac{\hbar\Omega}{2}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) - \frac{\hbar\Omega\lambda}{2} \\
& \left(\frac{\partial|\phi^+\rangle}{\partial\lambda}\langle\phi^+| + |\phi^+\rangle\frac{\partial\langle\phi^+|}{\partial\lambda} - \frac{\partial|\phi^-\rangle}{\partial\lambda}\langle\phi^-| - |\phi^-\rangle\frac{\partial\langle\phi^-|}{\partial\lambda}\right) \\
=& -\frac{\hbar\Omega}{2}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) - \frac{\hbar\Omega\lambda}{2}\frac{\partial\theta}{\partial\lambda}\left(|\phi^+\rangle\langle\phi^-| + |\phi^-\rangle\langle\phi^+|\right) \\
=& -\frac{\hbar\Omega}{2}\left(|\phi^+\rangle\langle\phi^+| - |\phi^-\rangle\langle\phi^-|\right) + \frac{\hbar\Omega}{2}\frac{\varepsilon}{\sqrt{\lambda^2-\varepsilon^2}}\left(|\phi^+\rangle\langle\phi^-| + |\phi^-\rangle\langle\phi^+|\right) \\
=& \left(-\frac{\hbar\Omega}{2\cosh\left[\hbar\Omega\lambda/2T\right]}M^z + \frac{\hbar\Omega\tanh\left[\hbar\Omega\lambda/2T\right]}{2}\mathbf{1}\right) + \\
& \frac{\varepsilon\sqrt{\hbar\Omega\lambda T\tanh\left[\hbar\Omega\lambda/2T\right]}}{2\lambda\sqrt{\lambda^2-\varepsilon^2}}\left(M^+ + M^-\right)\,.
\end{aligned} \tag{D.15}$$

And the generalized pressure at instantaneous equilibrium state can be directly calculated from according to the Helmholtz free energy $F = -T\ln Z$, i.e.,

$$f_{eq,t}^{\lambda} = -\partial_\lambda F = \frac{\hbar\Omega\tanh\left[\hbar\Omega\lambda/2T\right]}{2}\,. \tag{D.16}$$

Thus, combining (D.15) and (D.16), we obtain the variance of the generalized pressure

$$F_t^{\lambda} - f_{eq,t}^{\lambda} = -\frac{\hbar\Omega}{2\cosh\left[\hbar\Omega\lambda/2T\right]}M^z + \frac{\varepsilon\sqrt{\hbar\Omega\lambda T\tanh\left[\hbar\Omega\lambda/2T\right]}}{2\lambda\sqrt{\lambda^2-\varepsilon^2}}\left(M^+ + M^-\right)\,. \tag{D.17}$$

For the entropy, because

$$\begin{aligned} F_t^T =& -\ln \rho_{eq,t} \\ =& \frac{\hbar\Omega\lambda}{2T\cosh\left[\hbar\Omega\lambda/2T\right]} M^z - \frac{\hbar\Omega\lambda\tanh\left[\hbar\Omega\lambda/2T\right]}{2T} + \ln\left[2\cosh\left[\hbar\Omega\lambda/2T\right]\right] , \end{aligned} \tag{D.18}$$

and the entropy at instantaneous equilibrium state

$$\begin{aligned} f_{eq,t}^T =& -\partial_T F \\ =& \ln\left[2\cosh\left[\hbar\Omega\lambda/2T\right]\right] - \frac{\hbar\Omega\lambda\tanh\left[\hbar\Omega\lambda/2T\right]}{2T} , \end{aligned} \tag{D.19}$$

one can derive the variance of entropy as

$$F_{eq,t}^T - f_{eq,t}^T = \frac{\hbar\Omega\lambda}{2T\cosh\left[\hbar\Omega\lambda/2T\right]} M^z . \tag{D.20}$$

Inserting the above variances of thermodynamic forces into (D.5), after integrating over s, one can obtain the adiabatic response matrix

$$\begin{aligned} \mathcal{R} \quad &= \mathcal{D} + \mathcal{C} = \begin{pmatrix} \mathcal{D}^{TT} & \mathcal{D}^{T\lambda} \\ \mathcal{D}^{\lambda T} & \mathcal{D}^{\lambda\lambda} \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{C}^{\lambda\lambda} \end{pmatrix} \\ &= \frac{R}{T^2} \begin{pmatrix} -\lambda/T & 1 \\ 1 & -T/\lambda \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & \mathcal{C}^{\lambda\lambda} \end{pmatrix} , \end{aligned} \tag{D.21}$$

where,

$$R \quad = \frac{\hbar^2\Omega}{8\Gamma} \frac{\tanh\left[\hbar\Omega\lambda/2T\right]}{\cosh^2\left[\hbar\Omega\lambda/2T\right]} \tag{D.22}$$

$$\text{and} \qquad \mathcal{C}^{\lambda\lambda} = -\frac{\hbar\varepsilon^2}{2\lambda^2(\lambda^2-\varepsilon^2)} \frac{\Gamma}{\Gamma^2\coth^2\left[\hbar\Omega\lambda/2T\right]+1} . \tag{D.23}$$

AppendixE

Details for the genetic method

We first describe the basic principles of the penalty method [101].

The penalty method is applied to deal with the optimization problem with constraints in computer science [104–106]. Note that one can transform any optimization problems with constrains into the minimization problem below,

$$\begin{aligned} \min_{\boldsymbol{x}} \quad & f(\boldsymbol{x}) \\ \text{s.t.} \quad & g_i(\boldsymbol{x}) \leq 0 \qquad i = 1, 2, ...n; \\ & h_j(\boldsymbol{x}) = 0 \qquad j = 1, 2, ...m. \end{aligned} \tag{E.1}$$

Here $g_i(\boldsymbol{x})$ and $h_j(\boldsymbol{x})$ is the i-th inequality and the j-th equality constraint respectively.

The main idea to deal with this kind of problem is to construct an auxiliary function by the objective and constraint functions so that the original problem can be transformed into one with no constraints. Current methods provide a bunch of theories and algorithms for such auxiliary functions, but the penalty method is one of the most straightforward and efficient tools in numerical calculations.

The basic starting point of this method is that the auxiliary function $F(\boldsymbol{x})$ can be divide into the original objective function $f(\boldsymbol{x})$ and an additional penalty function $P(\boldsymbol{x})$. The latter is based on the constraint functions. Then, the problem (E.1) becomes,

$$\min_{\boldsymbol{x}} F(\boldsymbol{x}) = f(\boldsymbol{x}) + P(\boldsymbol{x}), \tag{E.2}$$

where

$$P(\boldsymbol{x}) = \sigma \left(\sum_{i=1}^{n} |\max(0, g_i(\boldsymbol{x}))| + \sum_{j=1}^{m} |h_i(\boldsymbol{x})| \right). \tag{E.3}$$

σ is a large penal constant.

If $\boldsymbol{x}$ is in the feasible region determined by the constraints, $F(\boldsymbol{x}) = f(\boldsymbol{x})$, i.e., $P(\boldsymbol{x}) = 0$. Otherwise, $P(\boldsymbol{x})$ will be a sufficiently large number to "punish" the current $\boldsymbol{x}$ and let it closer to the feasible region in the next iteration.

We then provide the details of the hyperparameters and numerical setups.

In our numerical calculation, we set the range $[0, 2]$ of x has been divided into n_c parts. The penalty constant here σ is set to be 10^5. In the parameter-coding step, the precision Δ is designed to be 10^{-7}. The size of the initial population N equals 1000 here. In the selection step, the Roulette wheel method is applied. By normalizing the fitness function of each individual into a positive number, the surviving probability can be calculated from the normalized fitness function divided by the sum of all. Then, 500 of individuals are selected to be parents and start to reproduce 500 child. In the gene crossover step, two random sites are selected to be the crossover points on both parents' chromosomes and each fragment divided by the selected sites has 50% probability to be inherited by the child. Next, in the gene mutation step, each site on the chromosome of each offspring is required to be randomly mutated at a small constant probability p_m which is 10^{-3} in our calculations. Therefore, $N_s = 500$ parents and their 500 child compose the new population and follow the algorithm iteratively. The iteration time is set to be 10^4 here. Values of all hyperparameters mentioned in this section are examined to be sufficient for convergence.

AppendixF

Detail for the machine learning method in Chapter 4

Our neural network has 3 hidden layers and each layer contains 128 neurons with the snake function, i.e., $f(x) = \sin^2(ax)/a$ ($a = 0.5$), which is proved to be efficient in fitting periodic functions [112]. Neurons in our neural network are fully-connected. The optimal parameters in our neural network can be obtained by back-propagation method. Here, we use the Adam optimizer [109] for its fast and stable performance. The learning rate for all cases are changing with the training steps. The starting learning rate is set to be 2×10^{-4}. Then it decreases a magnitude for every 10^4 steps. We discuss the influence of the location of the center and the number of Fourier bases M to our results. We use the near quasi-static case as our example and it can extends to the finite-time case.

We first discuss the influence of the location of the center. The limitation of our setup in Section 4.7.1 is that we cannot choose the origin out of the area enclosed by the optimal cycle. This problem arises: what if the cycle under the setup is a local optimal cycle? To answer the question, we choose several different origins with different locations and show the results in Figure F.1. If one chooses the location out of the optimal cycle, the cycle we finally obtained is no longer optimized (see the result using the origin A_1). Also, if the cycle is optimal, the origin can be many points in the area enclosed by the cycle (see the results using the origin A_0 and A_1). Hence, we show our setup is correct to find the optimal results instead of the local maximums.

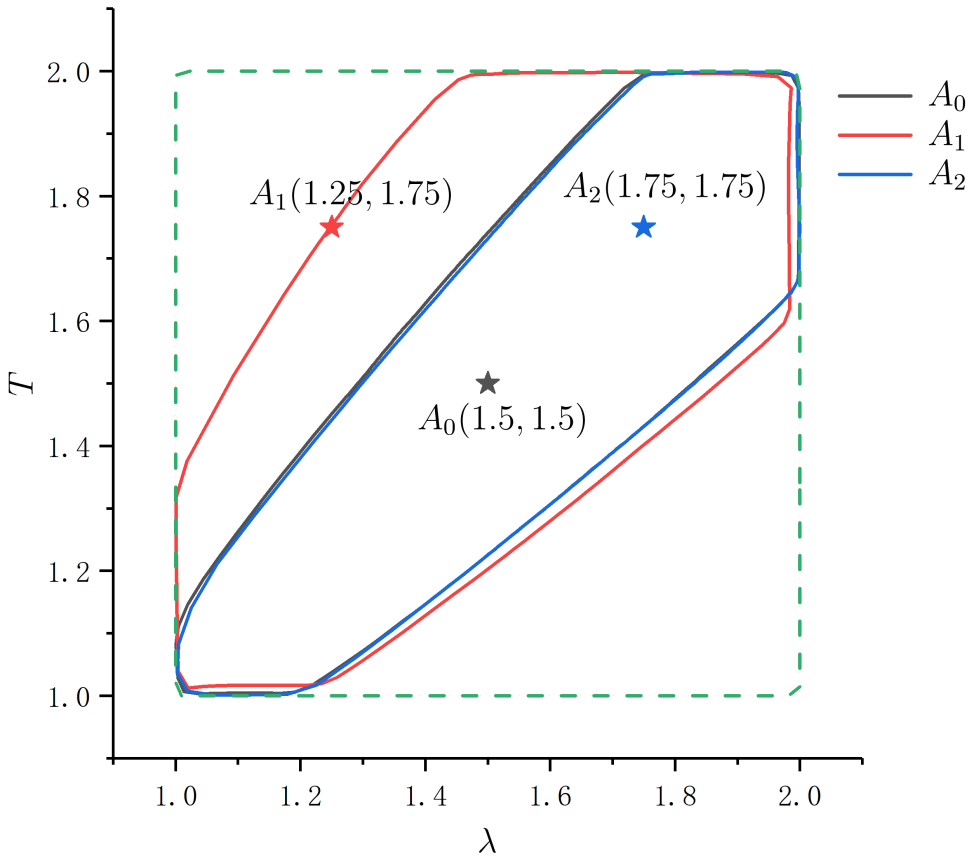

Figure F.1: The optimal thermodynamic cycles that maximize $\mathcal{W}/\mathcal{L}$ in the square boundary D for different the origins A_0, A_1 and A_2. The domain D is given by $(\lambda_m, T_m, a) = (1.0, 1.0, 0.5)$. Here the quantum coherence factor $\varepsilon = 0.1$. The number of Fourier modes $M = 8$. We notice that the value of $\mathcal{W}/\mathcal{L}$ for A_2 is smaller than those for A_0 and A_1.

.

We then discuss the influence of the number of Fourier orders M. As shown in the Figure F.2, we notice that when M is larger than 8, there is almost no difference among different M. Hence, for the calculation efficiency, we choose $M = 16$ for all examples in this thesis.

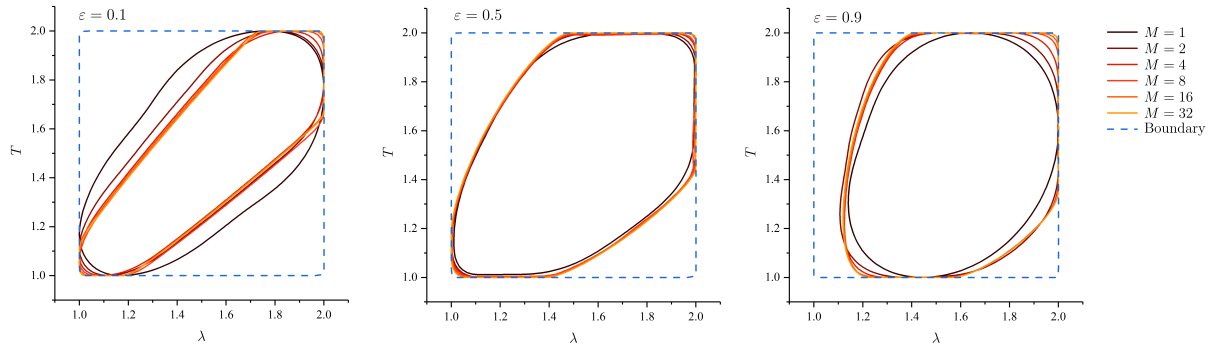

Figure F.2: The optimal thermodynamic cycles that maximize $\mathcal{W}/\mathcal{L}$ in the square boundary D for different number of Fourier modes. From left to right panel, the optimal cycles are shown for $\varepsilon = 0.1$, 0.5 and 0.9 respectively.

.

Bibliography

[1] John William Strutt. "Some general theorems relating to vibrations". In: *Proceedings of the London Mathematical Society* 1.1 (1871), pp. 357–368.

[2] Lars Onsager. "Reciprocal relations in irreversible processes. I." In: *Physical review* 37.4 (1931), p. 405.

[3] Donald G Miller. "Thermodynamics of Irreversible Processes. The Experimental Verification of the Onsager Reciprocal Relations." In: *Chemical Reviews* 60.1 (1960), pp. 15–37.

[4] Ilya Prigogine. "Time, structure, and fluctuations". In: *Science* 201.4358 (1978), pp. 777–785.

[5] Dilip Kondepudi and Ilya Prigogine. *Modern thermodynamics: from heat engines to dissipative structures*. John Wiley & Sons, 2014.

[6] David Jou, José Casas-Vázquez, and Georgy Lebon. "Extended irreversible thermodynamics". In: *Extended Irreversible Thermodynamics*. Springer, 1996, pp. 41–74.

[7] Denis J Evans and Debra J Searles. "The fluctuation theorem". In: *Advances in Physics* 51.7 (2002), pp. 1529–1585.

[8] Gavin E Crooks. "Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences". In: *Physical Review E* 60.3 (1999), p. 2721.

[9] Chris Jarzynski. "Nonequilibrium work theorem for a system strongly coupled to a thermal environment". In: *Journal of Statistical Mechanics: Theory and Experiment* 2004.09 (2004), P09005.

[10] Udo Seifert. "Stochastic thermodynamics, fluctuation theorems and molecular machines". In: *Reports on progress in physics* 75.12 (2012), p. 126001.

[11] Frank Weinhold. "Metric geometry of equilibrium thermodynamics". In: *The Journal of Chemical Physics* 63.6 (1975), pp. 2479–2483.

[12] Bjarne Andresen. "Current trends in finite-time thermodynamics". In: *Angewandte Chemie International Edition* 50.12 (2011), pp. 2690–2704.

[13] Kay Brandner, Keiji Saito, and Udo Seifert. "Strong bounds on Onsager coefficients and efficiency for three-terminal thermoelectric transport in a magnetic field". In: *Physical review letters* 110.7 (2013), p. 070603.

[14] Karel Proesmans and Christian Van den Broeck. “Onsager coefficients in periodically driven systems”. In: *Physical review letters* 115.9 (2015), p. 090601.

[15] Karel Proesmans, Bart Cleuren, and Christian Van den Broeck. “Power-efficiency-dissipation relations in linear thermodynamics”. In: *Physical review letters* 116.22 (2016), p. 220601.

[16] Naoto Shiraishi, Keiji Saito, and Hal Tasaki. “Universal trade-off relation between power and efficiency for heat engines”. In: *Physical review letters* 117.19 (2016), p. 190601.

[17] George Stockman and Linda G. Shapiro. *Computer Vision*. Prentice Hall, 2001, pp. 69–73.

[18] Alexander Toshev and Christian Szegedy. “DeepPose: Human Pose Estimation via Deep Neural Networks”. In: (2014), pp. 1653–1660.

[19] M. I. Jordan and T. M. Mitchell. “Machine learning: Trends, perspectives, and prospects”. In: *Science* 349.6245 (2015), pp. 255–260.

[20] LouisFranҫois Arsenault et al. “Machine learning for many-body physics: The case of the Anderson impurity model”. In: *Phys. Rev. B* 90.15 (2014).

[21] Giuseppe Carleo and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. In: *Science* 355.6325 (2017), pp. 602–606.

[22] Juan Carrasquilla and Roger G. Melko. “Machine learning phases of matter”. In: *Nature Physics* (2016).

[23] Evert P. L. van Nieuwenburg, Ye Hua Liu, and Sebastian D. Huber. “Learning phase transitions by confusion”. In: *Nature Physics* (2016).

[24] Lei Wang. “Discovering Phase Transitions with Unsupervised Learning”. In: 94.19 (2016).

[25] John J Hopfield. “Neural networks and physical systems with emergent collective computational abilities”. In: *Proceedings of the national academy of sciences* 79.8 (1982), pp. 2554–2558.

[26] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. “A learning algorithm for Boltzmann machines”. In: *Cognitive science* 9.1 (1985), pp. 147–169.

[27] D Anguita et al. “Quantum optimization for training support vector machines.” In: *Neural Networks the Official Journal of the International Neural Network Society* 16.5-6 (2003), p. 763.

[28] Michael Victor Berry. "Transitionless quantum driving". In: *Journal of Physics A: Mathematical and Theoretical* 42.36 (2009), p. 365303.

[29] Erik Torrontegui et al. "Shortcuts to adiabaticity". In: *Advances in atomic, molecular, and optical physics* 62 (2013), pp. 117–169.

[30] David Guéry-Odelin et al. "Shortcuts to adiabaticity: Concepts, methods, and applications". In: *Reviews of Modern Physics* 91.4 (2019), p. 045001.

[31] Adolfo Del Campo and Kihwan Kim. "Focus on shortcuts to adiabaticity". In: *New Journal of Physics* 21.5 (2019), p. 050201.

[32] Narendra N Hegade et al. "Shortcuts to adiabaticity in digitized adiabatic quantum computing". In: *Physical Review Applied* 15.2 (2021), p. 024038.

[33] Alan C Santos and Marcelo S Sarandy. "Superadiabatic controlled evolutions and universal quantum computation". In: *Scientific reports* 5.1 (2015), pp. 1–10.

[34] Kazutaka Takahashi. "Hamiltonian engineering for adiabatic quantum computation: Lessons from shortcuts to adiabaticity". In: *Journal of the Physical Society of Japan* 88.6 (2019), p. 061002.

[35] E Torrontegui et al. "Fast atomic transport without vibrational heating". In: *Physical Review A* 83.1 (2011), p. 013415.

[36] Xi Chen et al. "Fast optimal frictionless atom cooling in harmonic traps: Shortcut to adiabaticity". In: *Physical review letters* 104.6 (2010), p. 063002.

[37] Yan-Xiong Du et al. "Experimental realization of stimulated Raman shortcut-to-adiabatic passage with cold atoms". In: *Nature communications* 7.1 (2016), pp. 1–7.

[38] Andreas Hartmann et al. "Many-body quantum heat engines with shortcuts to adiabaticity". In: *Physical Review Research* 2.2 (2020), p. 023145.

[39] Jiawen Deng et al. "Boosting work characteristics and overall heat-engine performance via shortcuts to adiabaticity: Quantum and classical systems". In: *Physical Review E* 88.6 (2013), p. 062122.

[40] Mathieu Beau, Juan Jaramillo, and Adolfo Del Campo. "Scaling-up quantum heat engines efficiently via shortcuts to adiabaticity". In: *Entropy* 18.5 (2016), p. 168.

[41] Ander Tobalina, Ion Lizuain, and J Gonzalo Muga. "Vanishing efficiency of a speeded-up ion-in-Paul-trap Otto engine". In: *EPL (Europhysics Letters)* 127.2 (2019), p. 20005.

[42] Jun Jing et al. "Inverse engineering control in open quantum systems". In: *Physical Review A* 88.5 (2013), p. 053422.

[43] Victor Mukherjee et al. "Speeding up and slowing down the relaxation of a qubit by optimal control". In: *Physical Review A* 88.6 (2013), p. 062326.

[44] Roie Dann, Ander Tobalina, and Ronnie Kosloff. "Shortcut to equilibration of an open quantum system". In: *Physical review letters* 122.25 (2019), p. 250402.

[45] Roie Dann, Ander Tobalina, and Ronnie Kosloff. "Fast route to equilibration". In: *Physical Review A* 101.5 (2020), p. 052102.

[46] Geng Li, HT Quan, and ZC Tu. "Shortcuts to isothermality and nonequilibrium work relations". In: *Physical Review E* 96.1 (2017), p. 012144.

[47] John AC Albay et al. "Thermodynamic cost of a shortcuts-to-isothermal transport of a Brownian particle". In: *Physical Review Research* 1.3 (2019), p. 033122.

[48] John AC Albay, Pik-Yin Lai, and Yonggun Jun. "Realization of finite-rate isothermal compression and expansion using optical feedback trap". In: *Applied Physics Letters* 116.10 (2020), p. 103706.

[49] Nicola Pancotti et al. "Speed-ups to isothermality: Enhanced quantum thermal machines through control of the system-bath coupling". In: *Physical Review X* 10.3 (2020), p. 031015.

[50] Ignacio A Martínez et al. "Adiabatic Processes Realized With A Trapped Brownian Particle". In: *Physical Review Letters* 114.12 (2015), p. 120601.

[51] Ignacio A Martínez et al. "Brownian Carnot Engine". In: *Nature Physics* 12.1 (2016), pp. 67–70.

[52] Alberto Ronzani et al. "Tunable photonic heat transport in a quantum heat valve". In: *Nature Physics* 14.10 (2018), pp. 991–995.

[53] Johannes Roßnagel et al. "A Single-atom Heat Engine". In: *Science* 352.6283 (2016), pp. 325–329.

[54] Frank L Curzon and B Ahlborn. "Efficiency of a Carnot engine at maximum power output". In: *American Journal of Physics* 43.1 (1975), pp. 22–24.

[55] Christian Van den Broeck. "Thermodynamic efficiency at maximum power". In: *Physical review letters* 95.19 (2005), p. 190602.

[56] Massimiliano Esposito, Katja Lindenberg, and Christian Van den Broeck. "Universality of efficiency at maximum power". In: *Physical review letters* 102.13 (2009), p. 130602.

[57] Kay Brandner, Keiji Saito, and Udo Seifert. "Thermodynamics of micro-and nanosystems driven by periodic temperature variations". In: *Physical Review X* 5.3 (2015), p. 031019.

[58] Kay Brandner and Keiji Saito. "Thermodynamic geometry of microscopic heat engines". In: *Physical Review Letters* 124.4 (2020), p. 040602.

[59] Tim Schmiedl and Udo Seifert. "Optimal finite-time processes in stochastic thermodynamics". In: *Physical review letters* 98.10 (2007), p. 108301.

[60] Naoto Shiraishi, Ken Funo, and Keiji Saito. "Speed limit for classical stochastic processes". In: *Physical review letters* 121.7 (2018), p. 070601.

[61] Ken Funo, Naoto Shiraishi, and Keiji Saito. "Speed limit for open quantum systems". In: *New Journal of Physics* 21.1 (2019), p. 013006.

[62] Arthur L Samuel. "Some studies in machine learning using the game of checkers". In: *IBM Journal of research and development* 44.1.2 (2000), pp. 206–226.

[63] Donald Olding Hebb. *The Organization of Behavior: A Psychological Theory*. Wiley New York, 1949.

[64] F Rosenblatt. "The perceptron: a probabilistic model for information storage and organization in the brain". In: *Psychological Review* 65.6 (1958), pp. 386–408.

[65] B Widrow. "Adaptive switching circuits". In: *Ire Wescon Conv.rec* 4 (1960), pp. 126–134.

[66] Marvin Minsky and Seymour Papert. "An introduction to computational geometry". In: *Cambridge tiass., HIT* 479 (1969), p. 480.

[67] D. E Rumelhart, G. E Hinton, and R. J Williams. "Learning internal representations by error propagation". In: *Nature* 323.6088 (1986), pp. 533–536.

[68] Paul E. Utgoff. "Incremental Induction of Decision Trees". In: *Machine Learning* 4.2 (1989), pp. 161–186.

[69] Corinna Cortes and Vladimir Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297.

[70] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[71] Michael J. Kearns and Leslie G. Valiant. "Cryptographic limitations on learning Boolean formulae and finite automata". In: *Journal of the Acm* 41.1 (1994), pp. 433–444.

[72] Tin Kam Ho. "Random decision forests". In: *International Conference on Document Analysis and Recognition*. 1995, p. 278.

[73] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7 (2006), pp. 1527–1554.

[74] David Silver et al. "Mastering the game of Go with deep neural networks and tree search". In: *nature* 529.7587 (2016), pp. 484–489.

[75] Andrew W Senior et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710.

[76] Pankaj Mehta and David J. Schwab. "An exact mapping between the Variational Renormalization Group and Deep Learning". In: *Computer Science* (2014).

[77] P Rebentrost, M Mohseni, and S Lloyd. "Quantum support vector machine for big data classification." In: *Physical Review Letters* 113.13 (2013), p. 130503.

[78] Vasil S Denchev et al. "Robust Classification with Adiabatic Quantum Optimization". In: *Logistics Technology* (2012).

[79] Joseph Dulny Iii and Michael Kim. "Developing Quantum Annealer Driven Data Discovery". In: (2016).

[80] Anima Anandkumar et al. "Tensor Decompositions for Learning Latent Variable Models (A Survey for ALT)". In: *International Conference on Algorithmic Learning Theory*. 2015, pp. 19–38.

[81] Felix Brockherde et al. "Bypassing the Kohn-Sham equations with machine learning". In: *Nature Communications* 8.1 (2017), p. 872.

[82] G Carleo and M Troyer. "Solving the quantum many-body problem with artificial neural networks." In: *Science* 355.6325 (2016), pp. 602–606.

[83] Shengze Cai et al. "Physics-informed neural networks (PINNs) for fluid mechanics: A review". In: *Acta Mechanica Sinica* (2022), pp. 1–12.

[84] Pierpaolo Sgroi, G Massimo Palma, and Mauro Paternostro. "Reinforcement learning approach to nonequilibrium quantum thermodynamics". In: *Physical Review Letters* 126.2 (2021), p. 020601.

[85] Marin Bukov et al. "Reinforcement learning in different phases of quantum control". In: *Physical Review X* 8.3 (2018), p. 031086.

[86] Kurt Hornik. "Approximation capabilities of multilayer feedforward networks". In: *Neural networks* 4.2 (1991), pp. 251–257.

[87] Zhou Lu et al. "The expressive power of neural networks: A view from the width". In: *Advances in neural information processing systems* 30 (2017).

[88] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[89] Rongxing Xu. "Reinforcement learning approach to shortcuts between thermodynamic states with minimum entropy production". In: *Physical Review E* 105.5 (2022), p. 054123.

[90] Goran Lindblad. "On the generators of quantum dynamical semigroups". In: *Communications in Mathematical Physics* 48.2 (1976), pp. 119–130.

[91] Vittorio Gorini, Andrzej Kossakowski, and Ennackal Chandy George Sudarshan. "Completely positive dynamical semigroups of N-level systems". In: *Journal of Mathematical Physics* 17.5 (1976), pp. 821–825.

[92] Michael A Nielsen and Isaac Chuang. *Quantum computation and quantum information*. 2002.

[93] Sebastian Deffner and Eric Lutz. "Nonequilibrium entropy production for open quantum systems". In: *Physical review letters* 107.14 (2011), p. 140404.

[94] Giuliano Benenti et al. "Fundamental aspects of steady-state conversion of heat to work at the nanoscale". In: *Physics Reports* 694 (2017), pp. 1–124.

[95] Yehuda B Band, Oded Kafri, and Peter Salamon. "Finite time thermodynamics: Optimal expansion of a heated working fluid". In: *Journal of Applied Physics* 53.1 (1982), pp. 8–28.

[96] Tim Schmiedl and Udo Seifert. "Efficiency at maximum power: An analytically solvable model for stochastic heat engines". In: *EPL (Europhysics Letters)* 81.2 (2007), p. 20003.

[97] Massimiliano Esposito et al. "Finite-time thermodynamics for a single-level quantum dot". In: *EPL (Europhysics Letters)* 89.2 (2010), p. 20003.

[98] Massimiliano Esposito et al. "Quantum-dot Carnot engine at maximum power". In: *Physical review E* 81.4 (2010), p. 041106.

[99] Adriaan Daniël Fokker. "Die mittlere Energie rotierender elektrischer Dipole im Strahlungsfeld". In: *Annalen der Physik* 348.5 (1914), pp. 810–820.

[100] Andrzej Kossakowski. "On quantum statistical mechanics of non-Hamiltonian systems". In: *Reports on Mathematical Physics* 3.4 (1972), pp. 247–274.

[101] Rongxing Xu. "A numerical method to find the optimal thermodynamic cycle in microscopic heat engine". In: *Journal of Statistical Physics* 184.3 (2021), pp. 1–23.

[102] Darrell Whitley. "A genetic algorithm tutorial". In: *Statistics and Computing* 4.2 (1994), pp. 65–85.

[103] Zbigniew Michalewicz and Cezary Z Janikow. "Handling constraints in genetic algorithms." In: *Icga*. 1991, pp. 151–157.

[104] Özgür Yeniay. "Penalty function methods for constrained optimization with genetic algorithms". In: *Mathematical and computational Applications* 10.1 (2005), pp. 45–56.

[105] Abdollah Homaifar, Charlene X Qi, and Steven H Lai. "Constrained optimization via genetic algorithms". In: *Simulation* 62.4 (1994), pp. 242–253.

[106] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[107] Tan Van Vu and Keiji Saito. "Finite-Time Quantum Landauer Principle and Quantum Coherence". In: *Physical Review Letters* 128.1 (2022), p. 010602.

[108] Yoshitaka Tanimura and Ryogo Kubo. "Time evolution of a quantum system in contact with a nearly Gaussian-Markoffian noise bath". In: *Journal of the Physical Society of Japan* 58.1 (1989), pp. 101–114.

[109] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[110] Adam Paszke et al. "Pytorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019), pp. 8026–8037.

[111] J Robert Johansson, Paul D Nation, and Franco Nori. "QuTiP: An open-source Python framework for the dynamics of open quantum systems". In: *Computer Physics Communications* 183.8 (2012), pp. 1760–1772.

[112] Liu Ziyin, Tilman Hartwig, and Masahito Ueda. "Neural networks fail to learn periodic functions and how to fix it". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1583–1594.