Thèse n. 11 052

# Exact Sampling for Classical and Quantum Many-Body Systems

Présentée le 13 novembre 2024

Faculté des sciences de base
Laboratoire de sciences quantiques numériques
Programme doctoral en physique

pour l'obtention du grade de Docteur ès Sciences

par

**Dian WU**

Acceptée sur proposition du jury

Prof. F. Mila, président du jury
Prof. G. Carleo, directeur de thèse
Prof. F. Krzakala, rapporteur
Prof. G. Biroli, rapporteur
Prof. D. Poletti, rapporteur

2024

“Nature imitates Art.”
— Oscar Wilde, *The Decay of Lying*, 1889

*To my parents.*

# Abstract

Many-body systems at low temperature have revealed non-trivial phases of materials, such as spin liquids, which have found applications in the evolving fields of superconductivity, nanoelectronics, and quantum computing. Their exponentially large state spaces generally forbid us from obtaining their exact solutions and motivate us to develop sampling methods instead. The Markov chain Monte Carlo (MCMC) method is the traditional way to sample from complicated many-body distributions, but it suffers from issues such as autocorrelation time and critical slowing down. Exact sampling methods with variational ansatzes are preferred to overcome these issues, but it had been difficult to construct ansatzes that are both expressive enough to capture the rich physical phenomena and efficient enough to sample.

The recent development of neural networks sheds new light on this area. In this thesis, we present autoregressive neural networks (ARNNs) as a powerful family of variational ansatzes that share the same framework as modern large language models. They can be modularly constructed with higher expressiveness than previous ansatzes while supporting exact sampling and efficient evaluation.

For classical systems, we present the sparse two-body (TwoBo) ansatz to incorporate the sparsity of the physical system in ARNN. It substantially improves the computational efficiency and the convergence of variational optimization compared to the conventional dense ARNN, without loss of expressiveness. Then we present the sampling method of neural cluster updates with symmetries (NCUS), which finds a balance between the local updates of MCMC and the global updates of conventional exact sampling methods. It removes the bias in variational approximation and substantially reduces the autocorrelation time.

For quantum systems, we present the tensor-RNN ansatz, which combines the strengths of tensor networks (TNs) and recurrent neural networks (RNNs). It captures the desired analytical properties of entanglement entropy and spatial correlations in two-dimensional systems, and produces systematically improved accuracy as the computational budget increases. These advantages are supported by both theoretical and numerical evidence. Lastly, we present the results of VarBench, an extensive project to benchmark the performance of variational methods on quantum many-body systems, which witnesses the development of this field and identifies certain targets for the demonstration of quantum advantage in the future. The metric used in these benchmarks is the V-score, a universal quantity to measure the accuracy of any variational approximation, as well as the hardness of simulating any Hamiltonian. These techniques together enable more accurate approximation of systems with larger size and higher complexity.

# Résumé

À basse température, les systèmes à plusieurs corps ont révélé des phases non triviales de matériaux, tels que les liquides de spin, qui ont trouvé des applications dans les domaines de la supraconductivité, la nanoélectronique et l'informatique quantique, en constante évolution. Leurs espaces d'états de taille exponentielle nous empêchent généralement d'obtenir des solutions exactes et nous incitent en conséquence à développer des méthodes d'échantillonnage. La méthode de Monte-Carlo par chaîne de Markov (MCMC) est le moyen traditionnel d'échantillonner des distributions complexes de systèmes à plusieurs corps, mais elle est affectée de problèmes tels que le temps d'autocorrélation et le phénomène de ralentissement critique. Les méthodes d'échantillonnage exact appliquées à des ansatzes variationnels sont à privilégier afin de surmonter ces difficultés. Cependant, la conception de tels ansatzes, à la fois suffisamment expressifs pour capturer une phénoménologie riche et à l'échantillonnage suffisamment efficace, est longtemps restée un défi.

Le développement récent des réseaux neuronaux apporte un éclairage nouveau sur ce domaine. Dans cette thèse, nous présentons les réseaux neuronaux autorégressifs (ARNN) comme une famille puissante d'ansatzes variationnels, qui s'insère dans le même cadre que les grands modèles de langage modernes. Ils peuvent être construits de manière modulaire avec une plus grande expressivité que les ansatzes précédemment utilisés, tout en permettant un échantillonnage exact et une évaluation efficace.

Pour les systèmes classiques, nous présentons l'ansatz sparse à deux corps (TwoBo) pour incorporer dans l'ARNN la structure faiblement dense du système physique. Ceci améliore considérablement l'efficacité numérique et la convergence de l'optimisation variationnelle par rapport à l'ARNN dense conventionnel, sans perte d'expressivité. Par la suite, nous présentons la méthode neurale d'échantillonnage par mise à jour de clusters avec symétries (NCUS), qui concilie les mises à jour locales du MCMC et les mises à jour globales des méthodes conventionnelles d'échantillonnage exact. Cela supprime le biais de l'approximation variationnelle et réduit considérablement le temps d'autocorrélation.

Pour les systèmes quantiques, nous présentons l'ansatz tenseur-RNN, qui combine les atouts des réseaux tensoriels (TN) et des réseaux neuronaux récurrents (RNN). Il capture les propriétés analytiques souhaitées de l'entropie d'enchevêtrement et des corrélations spatiales dans les systèmes bidimensionnels, et produit une précision systématiquement accrue au fur et à mesure que le budget de calcul est augmenté. Ces avantages sont étayés par des preuves théoriques et numériques. Enfin, nous présentons les résultats de VarBench, un vaste projet visant à comparer les performances des méthodes variationnelles sur les systèmes quantiques

à plusieurs corps et qui témoigne du développement de ce domaine tout en identifiant certains objectifs pour la démonstration de l'avantage quantique à l'avenir. La métrique utilisée dans ces benchmarks est le V-score, une quantité universelle permettant de mesurer l'exactitude de toute approximation variationnelle, ainsi que le degré de difficulté de la simulation pour tout hamiltonien. Dans l'ensemble, ces techniques permettent une approximation plus exacte de systèmes de plus grande dimension et de plus grande complexité.

# Publications

The following publications are covered in this thesis:

- D. Wu, R. Rossi, and G. Carleo. "Unbiased Monte Carlo cluster updates with autoregressive neural networks". In: *Phys. Rev. Res.* 3.4 (2021), p. L042024. DOI: 10.1103/PhysRevResearch.3.L042024
- F. Vicentini, D. Hofmann, A. Szabó, D. Wu, C. Roth, C. Giuliani, G. Pescia, J. Nys, V. Vargas-Calderon, N. Astrakhantsev, and G. Carleo. "NetKet 3: Machine learning toolbox for many-body quantum systems". In: *SciPost Physics Codebases* (2022), p. 7. DOI: 10.21468/SciPostPhysCodeb.7
- D. Wu, R. Rossi, F. Vicentini, and G. Carleo. "From tensor-network quantum states to tensorial recurrent neural networks". In: *Phys. Rev. Res.* 5.3 (2023), p. L032001. DOI: 10.1103/PhysRevResearch.5.L032001
- I. Biazzo, D. Wu, and G. Carleo. "Sparse autoregressive neural networks for classical spin systems". In: *Mach. Learn.: Sci. Technol.* 5.2 (2024), p. 025074. DOI: 10.1088/2632-2153/ad5783
- D. Wu, R. Rossi, F. Vicentini, N. Astrakhantsev, F. Becca, X. Cao, J. Carrasquilla, F. Ferrari, A. Georges, M. Hibat-Allah, et al. "Variational benchmarks for quantum many-body problems". In: *Science* 386.6719 (2024), pp. 296–301. DOI: 10.1126/science.adg9774

The following research has been conducted during the thesis, but is not explicitly covered:

- D. Wu, F. Yang, and G. Carleo. "Unveiling nonmagnetic phase and many-body entanglement in two-dimensional random quantum magnets $Sr_2CuTe_{1-x}W_xO_6$". In: *arXiv Preprint* (2024). arXiv: 2407.05917

# Acknowledgements

First and foremost, I would like to thank my thesis director Giuseppe Carleo, who has always been a pioneering figure in the field of neural quantum states and guided me through my four years of PhD study. This journey would have been impossible to complete without the generous help of my postdoc colleagues — Riccardo Rossi, who can always ask questions; Filippo Vicentini, who knows NetKet and typography; Indaco Biazzo, who knows replica symmetry breaking; Fan Yang, who knows solid-state theory; and the valuable experiences from Jannes Nys, Friederike Metz, and Zakari Denis. I am equally thankful to the joyful discussions with my PhD colleagues and office mates — Stefano Barison, Julien Gacon, Gabriel Pescia, Imelda Romero, Alessandro Sinibaldi, Gian Gentinetta, Linda Mauron, David Linteau, Douglas Hendry, Sara Santos, and particularly — Clemens Giuliani, who is a bit less quiet than me and has helped me a lot with the hardware problems of my workstation in the office. Moreover, I am thankful to Svetlana Mashkina, who has kindly handled all the secretary works for us.

I would also like to thank the collaborators I have met, and more researchers from whom I have learned. Attila Szabó, Christopher Roth, and Damian Hofmann have given me great help in the development of NetKet. Juan Carrasquilla, Mohamed Hibat-Allah, and Or Sharir have shown me different routes in the search space of neural quantum states. Federico Becca has led me a thoughtful discussion on resonating valence bonds states, and Luciano Viteritti has provided me the latest insights on vision transformers. Tom Westerhout's lattice-symmetries is the fastest exact diagonalization library I have seen. It is written in Haskell, and it is hard for one to write a program with ever a bug in Haskell. I am also thankful to the friends I have made in EPFL — Jin Jiang, Yifei Guan, and Longdi Zhu, who have helped me solve the problem that the dishes in Chinese restaurants are too much for one person.

Finally, I would like to thank my parents, with whom I spend most time on video calling rather than text messaging. I am thankful as well to the indie game dev group Lunatic Works and the indie AI dev group ChatGal, who have fulfilled my spare time and have proven that friendship is not hindered by geographical distance.

Dian Wu
13 Nov. 2024, in Lausanne

# Contents

**Abstract (English/Français)** **i**

**Publications** **v**

**Acknowledgements** **vii**

**List of Figures** **xiii**

**I Introduction** **1**

**1 Background** **3**
1.1 Phases in many-body systems . . . . . 3
1.2 Computational approach . . . . . 4
1.3 Neural networks . . . . . 4
1.4 Outline of the thesis . . . . . 5

**2 Spin models of many-body systems** **9**
2.1 Classical systems . . . . . 9
2.1.1 Ising model . . . . . 9
2.1.2 Observables . . . . . 10
2.1.3 Geometrical frustration and residual entropy . . . . . 11
2.1.4 Spin glass . . . . . 12
2.1.5 Associative memory . . . . . 13
2.1.6 Frustrated plaquette model (FPM) . . . . . 14
2.2 Quantum systems . . . . . 15
2.2.1 Heisenberg model . . . . . 15
2.2.2 Transverse-field Ising model (TFIM) . . . . . 16

**II Computational studies of classical systems** **17**

**3 Markov chain Monte Carlo (MCMC)** **19**
3.1 Monte Carlo method . . . . . 19
3.2 Exact sampling . . . . . 20
3.3 Markov chain sampling . . . . . 21

3.4 Autocorrelation time . . . 22
3.4.1 Autocorrelation in Markov chain . . . 22
3.4.2 Integrated autocorrelation time . . . 23
3.4.3 Effective sample size . . . 24
3.5 Critical slowing down . . . 25
3.6 Burn-in stage and Gelman–Rubin statistic . . . 26
3.7 Other sampling methods . . . 28
3.7.1 Importance sampling . . . 28
3.7.2 MCMC importance sampling . . . 28
3.7.3 Cluster update . . . 29
3.8 Mode collapse . . . 30

**4 Variational method** **33**
4.1 Variational free energy . . . 33
4.2 Minimization of variational free energy . . . 34
4.2.1 Gradient descent optimizers . . . 34
4.2.2 Second-order optimizers . . . 35
4.3 Gradient estimator . . . 36
4.4 Comparison between MCMC and variational method . . . 37
4.5 Variational ansatzes for classical spin systems . . . 38
4.5.1 Naive mean-field ansatz . . . 38
4.5.2 Bethe ansatz . . . 41
4.6 Neural network ansatzes . . . 43
4.6.1 Layers in neural networks . . . 43
4.6.2 Symmetries in neural networks: convolutional and pooling layers . . . 44
4.6.3 Neural network as variational ansatz . . . 46
4.6.4 Machine learning models other than neural networks . . . 46

**5 Autoregressive neural networks (ARNNs)** **49**
5.1 Autoregressive factorization of joint probability . . . 49
5.2 Dense and convolutional ARNNs . . . 50
5.2.1 Numerical results . . . 52
5.2.2 Remarks on network sizes . . . 55
5.3 Sparse two-body ARNN (TwoBo) . . . 56
5.3.1 TwoBo architecture from Boltzmann distribution . . . 56
5.3.2 Sparsity of intermediate variables . . . 57
5.3.3 Numerical results . . . 58
5.3.4 Conclusion . . . 62
5.4 ARNN in importance sampling . . . 62
5.4.1 Neural cluster updates with symmetries (NCUS) . . . 63
5.4.2 Numerical results . . . 64
5.4.3 Conclusion . . . 66

**III Computational studies of quantum systems** **69**

**6 Unbiased evaluation of ground state** **71**
6.1 Imaginary time evolution (ITE) . . . 71
6.2 Exact diagonalization (ED) . . . 73
6.3 Quantum Monte Carlo (QMC) . . . 73
6.4 Sign problem . . . 75

**7 Variational Monte Carlo (VMC)** **77**
7.1 Variational energy . . . 77
7.2 Energy variance . . . 78
7.3 Complex gradient . . . 79
7.4 Stochastic reconfiguration (SR) . . . 81
7.5 Variational ansatzes for quantum spin systems . . . 84
7.5.1 Jastrow ansatz . . . 84
7.5.2 Gutzwiller projected states . . . 84
7.5.3 Neural quantum states (NQSs) . . . 86
7.6 Amplitude and phase of quantum ansatz . . . 87

**8 Tensor networks (TNs)** **89**
8.1 Tensor diagram notations . . . 90
8.2 Matrix product state (MPS) . . . 90
8.3 Contraction of MPS . . . 91
8.4 Matrix product operator (MPO) . . . 92
8.5 Density matrix renormalization group (DMRG) . . . 93
8.6 TNs with higher-dimensional geometries . . . 95

**9 Tensor-RNN: bridge between tensor networks and neural networks** **99**
9.1 Vanilla MPS-RNN . . . 100
9.2 1D MPS-RNN . . . 103
9.3 2D MPS-RNN . . . 104
9.4 Tensor-RNN . . . 105
9.5 Hierarchical initialization . . . 106
9.6 Numerical results . . . 106
9.6.1 Variational energy . . . 106
9.6.2 Spatial correlations . . . 108
9.6.3 Magnitudes of tensor, matrix, and vector terms . . . 108
9.7 Conclusion . . . 110

**10 VarBench: variational benchmarks for quantum many-body systems** **111**
10.1 V-score: universal metric for variational accuracy . . . 111
10.2 Hamiltonians and variational methods in the benchmark . . . 113
10.3 Benchmark results . . . 116
10.4 Hardness of Hamiltonian . . . 116

10.5 Conclusion . . . . . 118

**11 Conclusion** **119**
11.1 Main results . . . . . 119
11.2 Outlook . . . . . 120

**A Energy gradient in VMC** **123**

**B Details of the SR method** **125**

**Bibliography** **127**

**Curriculum vitae** **153**

# List of Figures


2.1 Frustrated interactions in Ising model . . . . . 11
2.2 Phase diagram of frustrated plaquette model (FPM) . . . . . 14

3.1 Energy of Ising model from independent and autocorrelated samples . . . . . 22
3.2 Autocorrelation in energy for Ising model at different temperatures . . . . . 23
3.3 Distribution of energy for Ising model at different temperatures . . . . . 24
3.4 Sample of Ising model at different temperatures . . . . . 25
3.5 Convergence of multiple Markov chains . . . . . 26
3.6 Distribution and sampling of magnetization for Ising model . . . . . 30
3.7 Distribution of energy for FPM with different sizes and temperatures . . . . . 31

4.1 Graphical solution to Curie–Weiss self-consistent equation . . . . . 40
4.2 Architecture of feedforward neural network (FFNN) . . . . . 43
4.3 Commonly used activation functions in neural networks . . . . . 45

5.1 Architectures of autoregressive neural networks (ARNNs) . . . . . 51
5.2 ARNN results of Ising model on square and triangular lattices . . . . . 52
5.3 ARNN results of Hopfield model . . . . . 54
5.4 ARNN results of Sherrington–Kirkpatrick (SK) model and inverse SK problem . 55
5.5 Architecture of the sparse two-body ARNN (TwoBo) . . . . . 57
5.6 TwoBo results of Edwards–Anderson (EA) model . . . . . 59
5.7 Convergence of TwoBo variational free energy in training . . . . . 60
5.8 TwoBo variational free energy vs. annealing speed . . . . . 60
5.9 Number of parameters vs. system size for TwoBo and MADE . . . . . 61
5.10 Variational free energy vs. number of parameters for TwoBo and RNN . . . . . 61
5.11 Procedure of neural cluster updates with symmetries (NCUS) . . . . . 63
5.12 NCUS results of Ising model . . . . . 64
5.13 NCUS results of FPM . . . . . 65
5.14 Choices of the distribution $P_{\text{cluster}}(k)$ in NCUS . . . . . 66

8.1 Tensor diagram notations . . . . . 89
8.2 Matrix product state (MPS) and matrix product operator (MPO) . . . . . 90
8.3 Sequential contraction of MPS . . . . . 91
8.4 Density matrix renormalization group (DMRG) . . . . . 94

8.5 Orders of MPS on 2D lattice . . . . . 96
8.6 Projected entangled-pair state (PEPS) . . . . . 96

9.1 Tensor diagram for conditional probability in MPS . . . . . 101
9.2 Tensor diagrams for an AR sampling step of MPS . . . . . 102
9.3 Computational graph for vanilla MPS-RNN . . . . . 102
9.4 Tensor diagram for memory update of 2D MPS-RNN . . . . . 104
9.5 Computational graph for 2D MPS-RNN . . . . . 105
9.6 Tensor diagram for memory update of tensor-RNN . . . . . 105
9.7 Variational energy vs. bond dimension for tensor-RNN . . . . . 107
9.8 Variational energy vs. number of parameters for tensor-RNN . . . . . 107
9.9 Spin correlations in MPS-RNN . . . . . 108
9.10 Contributions of tensor, matrix, and vector terms in tensor-RNN . . . . . 109

10.1 Overview of VarBench results . . . . . 115
10.2 Comparison of V-score and energy relative error . . . . . 116
10.3 V-scores of different variational methods . . . . . 117
10.4 V-scores of different Hamiltonians . . . . . 118

# Introduction Part I

# 1 Background

## 1.1 Phases in many-body systems

The phases of matter are among the oldest and most mysterious physical phenomena. The solid, liquid, and gas phases are just elementary-school-level examples of the boarder classification of phases. Another widely known example is that magnets lose their magnetization when heated, which can be described as a transition from the ferromagnetic phase to the paramagnetic one. In the Landau theory of phase transitions [7], different phases are defined by their symmetries, and a phase transition occurs with a breaking of symmetry and is characterized by a corresponding order parameter. For example, the transition from the paramagnetic phase to the ferromagnetic phase breaks the spatial reflection symmetry, and magnetization serves as an order parameter.

This simple theory worked well until the discovery of the fractional quantum Hall effect [8], where the conductivity of a two-dimensional electron gas exhibits quantized values as the external magnetic field changes, and the difference between these values can be smaller than the minimal difference predicted by the previous theory of energy level quantization. Perhaps more surprisingly, researchers attempted to distinguish the states with different conductivities into different phases, but only found that they all have the same symmetry and cannot be characterized by any local order parameter. Later, it was established that they can be characterized by global properties of the system, known as topological orders [9]. Since then, topological phases of matter with exotic properties have found applications in various evolving fields of physics, including superconductivity [10], nanoelectronics [11], and quantum computing [12].

A common kind of models for studying topological phases are the quantum spin liquid states [13–15]. They are characterized by the absence of any magnetic order, and they possess long-range correlations to produce the topological orders. A particular example is the resonating valence bonds (RVB) state [16], which is defined by the superposition of all possible parings of electrons in the system. Spin liquids usually appear as the ground states of many-body systems with competing interactions, known as the frustration. These systems have been realized in experiments, such as the triangular lattice of $YbMgGaO_4$ [17] and the kagome lattice of herbertsmithite [18]. Meanwhile, the theoretical understanding of spin liquids is also under active development, such as the Kitaev model [19] on the hexagonal

lattice with anisotropic interactions, which has been solved exactly. In addition to theoretical and experimental approaches, computational methods have been another line of research to predict the properties of these systems.

## 1.2 Computational approach

Since their invention, electronic computers have been executing one of the earliest algorithms in computational physics — the Monte Carlo method [20], for those secret nuclear researches in World War II. This approach replaces the analytical integrals and summations in physical models by random sampling and obtains the results with the power of computers. Over the decades, our computational power has grown by orders of magnitude, and algorithms that once required hours on a supercomputer can be finished in minutes on a laptop today. However, the mere increase in computational power is nothing close to the computational complexity of many-body systems.

This complexity comes from the exponentially large state spaces. For example, a spin-$\frac{1}{2}$ particle has two possible states — up and down, then a system of $N$ spins has $2^N$ possible states, which causes the so-called curse of dimensionality in statistics. The statistical properties of the system are given by summations over all the states, which would require exponentially long times if evaluated exactly, while in the viewpoint of computational complexity we only consider an algorithm to be practical if it finishes in a polynomially long time. Therefore, rather than exact methods, we can only develop approximation methods to estimate these properties. The Monte Carlo is such a method known for avoiding the curse of dimensionality, where the variance of the estimator always decreases as the number of samples increases, regardless of the state space dimension.

In various schemes of Monte Carlo, we still need to deal with the complexity of the physical system apart from the dimensionality. The variational Monte Carlo (VMC) [21] is one of such schemes to study the ground states of quantum many-body systems, upon which this thesis mainly develops. It approximates the ground state wave function by a trial function, commonly referred to as the “ansatz”, and optimizes the ansatz according to the sampled local energies of the physical system. Its difficulty lies in constructing an ansatz that can accurately approximate the target wave function, which can be a highly oscillating function in the exponentially high-dimensional state space, while evaluating and sampling from the ansatz in polynomial time. Traditional ansatzes usually fulfill these requirements with the mathematical properties of determinants [22, 23], and it had been a long and difficult search to construct more powerful ansatzes from physical arguments.

## 1.3 Neural networks

The development of neural networks in the past decade has shed new light on this area. Neural networks, in the context of modern machine learning, are a diverse class of mathematical functions that are historically inspired by the functionality of biological neural networks [24, 25]. A striking difference from previous machine learning approaches is that neural networks

typically have numerous parameters, where modern large models have reached hundreds of billions of parameters [26]. In theory, they are universal approximators of any well-behaved functions if optimized w.r.t. the parameters [27]. Although the ideal optimization of so many parameters is intractable in practice, they can still be far more expressive than traditional function approximators [28]. On the other hand, the large number of parameters makes them more computationally expensive than traditional approaches, and they are usually executed on graphics processing units (GPUs) and other specialized hardware [29]. The optimization of these parameters is known as "training" or "learning", highlighting the paradigmatic difference from traditional optimization problems with few variables.

Since their breakthrough on image classification problems [30], the neural networks with deep multilayer architectures have opened a new era of machine learning studies, known as "deep learning" [31]. They have dominated state-of-the-art results in all machine learning domains and led to the growth of the data-driven paradigm in vast scientific fields [32], where prominent examples include the modeling of languages [26], images [33], and strategic games [34]. These successful results are conjectured to stem from the high efficiency of neural networks to represent complicated probability distributions in exponentially high-dimensional spaces — probabilities of valid speeches in the space of many words, probabilities of recognizable images in the space of many pixels, and probabilities of winning moves in the space of many game board states. Therefore, they have also been introduced as promising tools in many-body physics, where such distributions have become the main difficulty.

## 1.4 Outline of the thesis

This thesis is written in a mostly self-contained way, assuming that the reader has basic backgrounds in statistics and quantum mechanics. It is organized as follows:

- Chapter 2 provides the definitions of the classical and quantum many-body systems studied in this thesis, including the classical Ising model and its variants, the quantum Heisenberg model, and the transverse-field Ising model (TFIM). It discusses the brief histories and the motivations to study them, and the issues of geometrical frustration and random interactions that create disordered states of the systems. These systems exhibit rich and exotic physical phenomena, which pose challenge to the computational studies we will present.

- Chapter 3 starts the discussion of computational studies on classical many-body systems. It reviews the Markov chain Monte Carlo (MCMC) method, which is the most traditional and widely used method to stochastically estimate observables under complicated many-body distributions of physical systems. It discusses the issues of autocorrelation time, critical slowing down, burn-in stage before equilibrium, and mode collapse, which will reoccur in the upcoming discussions on all other methods based on MCMC, as well as in quantum systems. Before introducing the Markov chain sampling, the concept of exact sampling is defined with simple examples, which will be the focus of this thesis.

- Chapter 4 reviews the variational method to approximate the Boltzmann distributions of

classical many-body systems. It gives a detailed discussion on the optimization methods and the gradient estimators for the variational free energy, as well as a comparison with MCMC. Then it lists some commonly used variational ansatzes, including the naive mean-field ansatz, the Bethe ansatz, and neural network ansatzes, with a self-contained formulation of neural networks. This variational method has close resemblance to the variational Monte Carlo method for quantum systems in Chapter 7.

- Chapter 5 introduces the autoregressive neural networks (ARNNs). Using the power of neural networks, they enable practical applications of exact sampling for complicated many-body distributions. It presents the basic architectures of ARNN from Ref. [35], the sparse two-body ARNN (TwoBo) from Ref. [4] to substantially improve the efficiency using the sparsity structure of the physical system, and the neural cluster updates with symmetries (NCUS) from Ref. [1] to remove the bias from the variational approximation, with numerical results demonstrating their accuracy and efficiency.

- Chapter 6 moves the discussion to computational studies on quantum many-body systems. It reviews the unbiased methods to evaluate the ground state, including the theoretical formulation of imaginary time evolution (ITE), the exact diagonalization (ED) method derived from the former, and the quantum Monte Carlo (QMC) methods, in particular the path integral Monte Carlo (PIMC). It discusses the impact of the energy gap size on the convergence speeds of these methods, and the notorious sign problem, which will reoccur in Chapter 7. The results of ED and QMC are used as baselines for other numerical studies, including the benchmarks in Chapter 10.

- Chapter 7 reviews the variational Monte Carlo (VMC) method, which is widely used in quantum problems because of higher efficiency than the unbiased QMC schemes. It skips the formulation similar to the classical variational method in Chapter 4, and mentions the relation between the variance and the accuracy of the variational energy, which will be the fundament of the benchmarks in Chapter 10. It emphasizes the particular difficulty of complex gradients for wave functions, and presents the stochastic reconfiguration (SR) method to help the optimization in complicated energy landscapes of quantum systems. Then it lists some commonly used variational ansatzes, including the Jastrow ansatz, the Gutzwiller projected states, and the neural quantum states (NQSs), with a discussion on parameterizing and optimizing the phase of the complex-valued wave function.

- Chapter 8 reviews the tensor network (TN) method, which is an alternative to VMC for the variational approximation of quantum states. TNs have well-understood physical properties, such as entanglement entropy and spatial correlations, as well as the advantage of exact contraction if the computational cost allows. In particular, it presents the matrix product state (MPS) and the matrix product operator (MPO) architectures in 1D, as well as the density matrix renormalization group (DMRG) algorithm to optimize MPS, with brief introductions to architectures in 2D and higher.

- Chapter 9 introduces the tensor-RNN from Ref. [3], a variational ansatz combining the strengths of TN in Chapter 8 and NQS in Chapter 7. This ansatz supports exact sampling and efficient evaluation, captures the desired analytical properties of entanglement entropy and spatial correlations in 2D, and produces systematically improved accuracy as the computational budget increases. These advantages are supported by both theoretical and numerical evidence.

- Chapter 10 introduces the VarBench project from Ref. [5]. It first proposes the V-score, a universal metric based on the variance of variational energy, which measures the accuracy of any variational approximation, as well as the hardness of simulating any Hamiltonian. This score is used in an extensive collection of benchmarks for many kinds of variational methods and quantum many-body systems, which identifies certain hard Hamiltonians that are not yet well solved, and can be the targets to demonstrate the quantum advantage in the future.

- Chapter 11 concludes this thesis, summarizes the main results that improve the accuracy and the efficiency of computational studies in both classical and quantum many-body systems, and poses directions and open questions for further research.

# 2 Spin models of many-body systems

After the general introduction in the previous chapter, we now provide the formal definitions of the physical systems studied in this thesis, along with the motivations to study them. We mainly focus on systems of spin-$\frac{1}{2}$ particles placed on lattices, and most results are generalizable to higher spins, fermions, and bosons. Specifically, we focus on classical systems in thermal equilibrium and quantum systems in ground states.

## 2.1 Classical systems

### 2.1.1 Ising model

The early studies of the Ising model [36, 37] predate the modern formulation of quantum physics. It was proposed to describe the properties of ferromagnets using spins $\boldsymbol{s} = (s_1, s_2, \ldots, s_N)$ on a lattice of $N$ sites, where each spin $s_i$ can take the values $\pm 1$, and the Hamiltonian function for the interactions between the spins is

$$H(\boldsymbol{s}) = J \sum_{\langle i,j \rangle} s_i s_j, \tag{2.1}$$

where $\langle i, j \rangle$ runs over nearest neighbors on the lattice. We omit the external fields for simplicity and assume that $J = 1$ unless otherwise stated. When the system is in thermal equilibrium at the temperature $T$, or more conventionally the inverse temperature $\beta = \frac{1}{T}$, it is described by an ensemble of spin configurations following the Boltzmann distribution

$$p_{\mathrm{B}}(\boldsymbol{s}) = \frac{\mathrm{e}^{-\beta H(\boldsymbol{s})}}{Z}, \tag{2.2}$$

$$Z = \sum_{\boldsymbol{s}} \mathrm{e}^{-\beta H(\boldsymbol{s})}, \tag{2.3}$$

where $Z$ is called the partition function.

While the Curie–Weiss mean-field theory [38] predicted a second-order phase transition at the critical point $\beta_{\mathrm{c}} = \frac{1}{2D}$, where $D$ is the spatial dimension, the analytical solutions established that there is no phase transition in one dimension (1D) [36], a second-order phase transition in the thermodynamic limit on two-dimensional (2D) grids at $\beta_{\mathrm{c}} = \frac{1}{2}\ln\left(1+\sqrt{2}\right) \approx 0.44$ [39], as

well as second-order phase transitions on four- and higher-dimensional grids where the mean-field theory is asymptotically exact [40]. However, no analytical solution in three dimensions (3D) is known yet [41].

Although the above formulation of spins is overly simplified from the viewpoint of quantum physics, the Ising model was revived in the studies of various quantum systems, as we will see in Section 2.2.2, notably being the simplest model for quantum phase transitions [42]. The classical formulation is still relevant when the quantum hopping effect described by the transverse field is negligible. With the presence of the transverse field, it has also been shown that any quantum Ising model in $D$ dimensions can be mapped to a classical one in $D+1$ dimensions in the same universality class [43], which opens a practical way to study these quantum systems. Moreover, the same mathematical form appears in the tight-binding theory of electronic bands [44].

### 2.1.2 Observables

Given the distribution of the configurations $p(\boldsymbol{s})$, we can obtain the macroscopic expectation value of any physical observable $O(\boldsymbol{s})$ by

$$\bar{O} = \sum_{\boldsymbol{s}} p(\boldsymbol{s}) O(\boldsymbol{s}). \tag{2.4}$$

The observables of common interest include the energy $E$, the heat capacity $\mathcal{C}_V$, the magnetization $M$, the magnetic susceptibility $\chi_M$, and the spin correlation $C(d)$ as a function of the distance $d$. We note that $\mathcal{C}_V$ and $\chi_M$ can be expressed from the fluctuations of $E$ and $M$ respectively, or alternatively from the derivatives of the partition function. It is also feasible to evaluate the entropy

$$S = -\sum_{\boldsymbol{s}} p(\boldsymbol{s}) \ln p(\boldsymbol{s}) \tag{2.5}$$

and the free energy

$$F = -T \ln Z = E - TS, \tag{2.6}$$

which require to evaluate the partition function itself in $\ln p(\boldsymbol{s})$, and lead to extra difficulty compared to the former observables. Moreover, the ground state and its energy

$$\boldsymbol{s}_0 = \arg\min_{\boldsymbol{s}} H(\boldsymbol{s}), \tag{2.7}$$

$$E_0 = \lim_{T \to 0} E = H(\boldsymbol{s}_0) \tag{2.8}$$

are of interest in the case of non-trivial lattice geometries and interactions, which has its root in combinatorial optimization problems arisen from mathematics, computer sciences, and even social sciences, as we will discuss in Section 2.1.4.

The analytical solutions to these observables are attainable only with some simple lattice geometries and interactions. Many 1D systems are solvable using the transfer matrix method [45], with a few results in 2D [46–48]. In the remaining cases, which include rich and unknown physical phenomena, numerical methods are required to study them.

However, the efficiency of numerical methods is impeded by the sheer high dimension of the configuration space. The summation in Eq. (2.4) runs over $2^N$ possible configurations of the system in total, which is exponentially large and quickly becomes intractable as $N$ increases. As of this writing, the largest supercomputers in the world have peak computational power on the order of $10^{18}$ floating-point operations per second (1 exaFLOPS) [49], which can sum over only 81 spins in the "sane" time budget of a month, even if the summation is unrealistically optimized to take one floating-point operation for each configuration, and other costs such as data storage and transfer are ignored. This means that we can only resort to approximation methods to compute the above expectation in practice, which we will discuss throughout this thesis.

### 2.1.3 Geometrical frustration and residual entropy

A difficulty in solving the Ising model comes from the lattice geometry, as shown in Fig. 2.1. If the system is ferromagnetic (FM) with $J < 0$ in Eq. (2.1), where every pair of interacting spins tend to align in the same direction to produce lower energy, then the ground state is intuitively obtained when all spins are aligned. In contrast, if the system is antiferromagnetic (AFM) with $J > 0$, and the lattice contains odd-length cycles, then there is no longer an intuitive way to obtain the ground state, and even a simple system of three spins has as many as six degenerate ground states. This phenomenon is known as geometrical frustration.

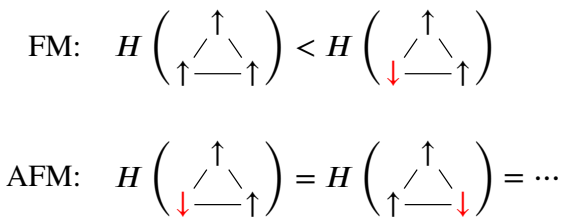



Figure 2.1: The non-frustrated FM and the frustrated AFM interactions for the Ising model on a triangle of three spins.

On a triangular lattice of $N$ spins, the number of degenerate ground states $N_{\mathrm{GS}}$ increases exponentially with $N$, which poses challenge for analytical and numerical methods to accurately evaluate the observables, or merely count the order of magnitude of $N_{\mathrm{GS}}$. In addition to the triangular lattice [50], other types of frustrated lattices also appear in the modeling of materials, such as the kagome lattice [51] and the pyrochlore (tetrahedral) lattice [52]. These disordered ground states can lead to the spin liquid phase in the quantum description of these materials, which is an active topic of research.

Counting $N_{\mathrm{GS}}$ is equivalent to computing the entropy at zero temperature, where only the ground states contribute to the entropy:

$$\lim_{\beta\to\infty} S = \ln N_{\mathrm{GS}}. \tag{2.9}$$

In the thermodynamic limit $N \to \infty$, we assume

$$N_{\mathrm{GS}} = A\mathrm{e}^{cN}, \tag{2.10}$$

where $A$ and $c$ are coefficients to be determined. Then we have

$$\lim_{\beta\to\infty} \frac{S}{N} = \frac{\ln A}{N} + c, \tag{2.11}$$

which converges to $c$ as $N \to \infty$, and $c$ is known as the residual entropy [53, 54]. The existence of a non-zero residual entropy indicates the exponentially large number of degenerate ground states. For the relatively simple case of Ising model on the 2D triangular lattice, the analytical result has been derived [55–57]:

$$\lim_{N\to\infty} \lim_{\beta\to\infty} \frac{S}{N} = \frac{2}{\pi} \int_0^{\frac{\pi}{3}} \ln(2\cos\omega)\mathrm{d}\omega \approx 0.323. \tag{2.12}$$

This residual entropy will be studied in Section 5.2.1. It is also worth mentioning that the boundary conditions of finite-sized systems can substantially affect the observables in the presence of frustration.

### 2.1.4 Spin glass

In addition to geometrical frustration, another cause of disorder in Ising models is that the interactions $J_{ij}$ between the spins can be random. In this case, the Hamiltonian function is written as

$$H(\boldsymbol{s}) = \sum_{ij} J_{ij} s_i s_j. \tag{2.13}$$

This kind of systems are known as spin glasses [58, 59], because of the analogy to the positional disorder in conventional amorphous glasses. Their random interactions can produce numerous metastable states, making it difficult to investigate the properties around the true ground state. Even with the optimization power of nature, spin glass materials actually have exceptionally long relaxation times before reaching equilibrium.

In particular, the Sherrington–Kirkpatrick (SK) model [60] is an early example of spin glasses. It is defined for $N$ spins that interact with each other, and each interaction strength $J_{ij}$ is sampled from a Gaussian distribution with zero mean and a variance of $\frac{1}{N}$. The Edwards–Anderson (EA) model [61] is another example, which is defined on a lattice with nearest-neighbor interactions:

$$H(\boldsymbol{s}) = J \sum_{\langle i,j \rangle} J_{ij} s_i s_j. \tag{2.14}$$

The interaction strength $J_{ij}$ is conventionally sampled from a Gaussian distribution, and in Section 5.3.3 we will study a variant with binary interactions, where each $J_{ij}$ randomly takes $\pm 1$. In the limit of large system size with the average over random interactions, they can be studied using the replica method [62], while cases with finite system sizes and specific interactions require more detailed studies.

In addition to modeling a variety of non-uniform alloys and compounds [63], spin glasses also appear in the vast fields of mathematics and computer science. Finding the ground state of a spin glass model with specific interactions is equivalent to several combinatorial optimization problems that are known to be hard, such as the quadratic unconstrained binary optimization (QUBO), the maximum cut of graphs (max-cut), and the boolean satisfiability with three literals (3-SAT) [64]. In particular, while it has been proven that finding the ground state only has polynomial time complexity on 2D lattices without external fields, it is NP-hard on 2D lattices with external fields and on 3D lattices [65]. Moreover, computing the partition function is #P-hard on general graphs [66–68]. Both NP-hard and #P-hard problems are at least as hard as NP-complete problems, which cannot be solved in polynomial time unless P = NP. This means that if we aim to find the ground state exactly and provably in the general case, then any smart algorithm is unlikely to be substantially faster than the brute-force enumeration. Therefore, we are further motivated to develop non-deterministic and heuristic approximation methods for this kind of problem.

### 2.1.5 Associative memory

Ising models also manifest in the studies of associative memory, in the sense that we can encode data in its parameters and retrieve the data by sampling from it. This kind of model has its roots in both computer science and neuroscience [69]. An example is the Hopfield model [24, 70], which we will study in Section 5.2.1. It is defined by the fully connected Ising model in Eq. (2.13) with the interactions

$$J_{ij} = -\frac{1}{N}\sum_{p=1}^{n_{\text{pat}}} \xi_i^{(p)} \xi_j^{(p)}, \tag{2.15}$$

where $\{\boldsymbol{\xi}^{(1)}, \boldsymbol{\xi}^{(2)}, \ldots, \boldsymbol{\xi}^{(n_{\text{pat}})}\}$ are some special spin configurations, known as the "patterns", and $n_{\text{pat}}$ is the number of patterns. We usually consider orthogonal patterns:

$$\left\langle \boldsymbol{\xi}^{(p)} \middle| \boldsymbol{\xi}^{(q)} \right\rangle = 0 \quad \forall p, q, \tag{2.16}$$

where the overlap between two configurations is defined by

$$\left\langle \boldsymbol{s} \middle| \boldsymbol{s}' \right\rangle = \frac{1}{N}\sum_i s_i s_i'. \tag{2.17}$$

Then the model has $2n_{\text{pat}}$ degenerate ground states: $\boldsymbol{s}_0 = \pm\boldsymbol{\xi}^{(p)}$, $H(\boldsymbol{s}_0) = -N$. Therefore, we say that the model "memorizes" or "stores" the patterns.

We note that for any configuration $\boldsymbol{s}$ and two orthogonal patterns $\boldsymbol{\xi}^{(1)}$ and $\boldsymbol{\xi}^{(2)}$, we have

$$\left| \left\langle \boldsymbol{s} \middle| \boldsymbol{\xi}^{(1)} \right\rangle \right| + \left| \left\langle \boldsymbol{s} \middle| \boldsymbol{\xi}^{(2)} \right\rangle \right| \leq 1, \tag{2.18}$$

which will be seen in Fig. 5.3.

### 2.1.6 Frustrated plaquette model (FPM)

While the previous models only have second-order phase transitions, one way to produce first-order phase transitions is to include interactions between more spins. We have constructed a frustrated plaquette model (FPM) as an example of first-order phase transition, which we will study in Section 5.4.2.

The FPM is defined on an $L \times L$ lattice of the spins $s_{x,y}$, with nearest-neighbor $J_1$, third-nearest-neighbor $J_3$, and plaquette $K$ interactions:

$$\begin{aligned} H(\boldsymbol{s}) = \quad & J_1 \sum_{i,j=1}^{L} s_{i,j}(s_{i+1,j} + s_{i,j+1}) \\ & + J_3 \sum_{i,j=1}^{L} s_{i,j}(s_{i+2,j} + s_{i,j+2}) \\ & + K \sum_{i,j=1}^{L} s_{i,j}\, s_{i+1,j}\, s_{i,j+1}\, s_{i+1,j+1}. \end{aligned} \tag{2.19}$$

When $J_1 = J_3 = -1$, its phase diagram with varying $K$ and $T$ is shown in Fig. 2.2. In addition to the paramagnetic (PM) phase at high $T$, it has a ferromagnetic (FM) phase at low $T$ and low $K$, and a ferrimagnetic (fM) phase at low $T$ and high $K$. In the fM phase, the ground state breaks the spin-inversion symmetry and the translational symmetry by one site, which is the result of competing interactions $J_1$, $J_3$, and $K$.

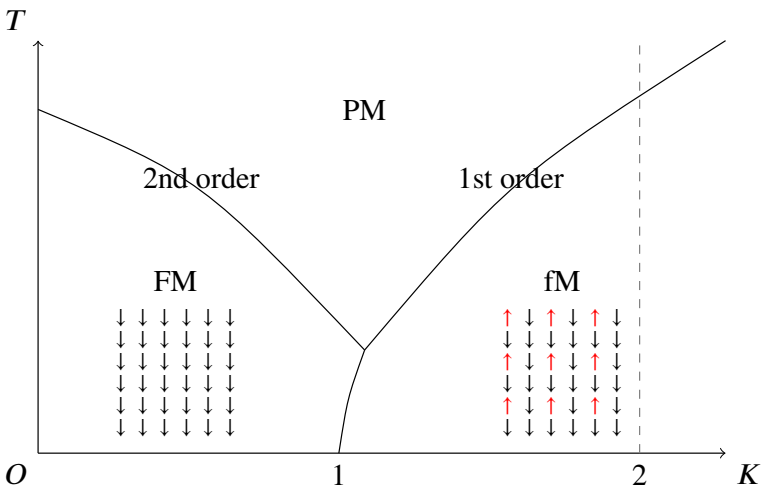



Figure 2.2: Sketch of the phase diagram for the FPM with varying plaquette interaction strength $K$ and temperature $T$, at $J_1 = J_3 = -1$. Examples of the ground state in the ferromagnetic (FM) and the ferrimagnetic (fM) phases are shown respectively. The case of $K = 2$ as indicated by the dashed vertical line is studied in Ref. [1], with a first-order phase transition between the fM and the paramagnetic (PM) phases. This figure is reproduced from Fig. 3 (a) in Ref. [1].

Although not covered by our research, it is worth mentioning that the Wishart planted ensemble [71, 72] has been proposed to be a more controllable example of first-order phase

transition, as well as a more modern example of associative memory.

## 2.2 Quantum systems

### 2.2.1 Heisenberg model

The Heisenberg model [73] is a common description of the magnetic properties of materials in quantum physics. It assumes that the electrons are localized on the lattice, so the energy levels are only determined by their spin interactions rather than positional interactions. The Hamiltonian is written as

$$\hat{H} = J \sum_{\langle i,j \rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j, \tag{2.20}$$

where the Pauli operators for spin-$\frac{1}{2}$ particles are represented by

$$\hat{\sigma}^x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \hat{\sigma}^y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \hat{\sigma}^z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.21}$$

and $\hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j = \hat{\sigma}_i^x \hat{\sigma}_j^x + \hat{\sigma}_i^y \hat{\sigma}_j^y + \hat{\sigma}_i^z \hat{\sigma}_j^z$ is the dot product of two 3D vectors of Pauli operators. It can also be derived from the Hubbard model with half filling of electrons and small hopping strength, where in the ground state each site is occupied by an electron, and the electron exchanges are given by the second-order perturbation of hopping [74].

The ground state of the system is determined by

$$\left|\psi_0\right\rangle = \arg\min_{|\psi\rangle} \left\langle \psi \middle| \hat{H} \middle| \psi \right\rangle, \tag{2.22}$$

which contains the complete information about the system at zero temperature, and is usually the foundation for further studying the excitations or the thermal properties of the system. In contrast to the classical case, even if the quantum system is in the ground state and is non-degenerate, it can still be a superposition of exponentially many basis states $|\boldsymbol{s}\rangle$. Given a state $\left|\psi\right\rangle$, we can obtain the expectation value of any physical observable $\hat{O}$ by

$$\bar{O} = \left\langle \psi \middle| \hat{O} \middle| \psi \right\rangle = \sum_{\boldsymbol{s},\boldsymbol{s}'} \psi^*(\boldsymbol{s}) O(\boldsymbol{s}, \boldsymbol{s}') \psi(\boldsymbol{s}), \tag{2.23}$$

where $\psi(\boldsymbol{s}) = \left\langle \boldsymbol{s} \middle| \psi \right\rangle$ and $O(\boldsymbol{s}, \boldsymbol{s}') = \left\langle \boldsymbol{s} \middle| \hat{O} \middle| \boldsymbol{s}' \right\rangle$. As in the classical case, the summation in Eq. (2.23) also contains exponentially many terms and is intractable. Worse still, to exactly compute the ground state vector in Eq. (2.22) in general, one needs the exact diagonalization (ED) of the $2^N \times 2^N$ Hamiltonian matrix, which takes even higher exponential time [75]. Therefore, we again need approximation methods to study the system in practice.

The issue of frustration discussed in Section 2.1.3 also occurs in the quantum case. On frustrated lattices such as the triangular lattice [17], the kagome lattice [18], and the pyrochlore lattice [76], the Heisenberg model can exhibit spin liquid phases. Another model commonly believed to exhibit the spin liquid phase is the $J_1$-$J_2$ model, which is defined as the Heisenberg

model with next-nearest-neighbor interactions [77–80].

### 2.2.2 Transverse-field Ising model (TFIM)

The transverse-field Ising model (TFIM) is formulated as the classical Ising model in Section 2.1.1 with a transverse field $\Gamma$ describing the quantum hopping effect. Its Hamiltonian is written as

$$\hat{H} = J \sum_{\langle i,j \rangle} \hat{\sigma}_i^z \hat{\sigma}_j^z + \Gamma \sum_i \hat{\sigma}_i^x. \tag{2.24}$$

Its usages in the early days of solid-state physics include a special case of anisotropic Heisenberg model [81], and a phenomenological model with fictitious spins for the order-disorder transition of the ferroelectric crystal $KH_2PO_4$ [82]. With more research on the classification of phase transitions, it is known to be the simplest model to demonstrate the quantum phase transition at zero temperature, which is driven by quantum fluctuations rather than thermal fluctuations [42, 83]. This actually describes the phase transitions of several magnetic materials, such as $LiHOF_4$ [84] and $CoNb_2O_6$ [85].

The TFIM in 1D has been analytically solved with the critical point $\Gamma_c = 1$. The analytical solution in 2D is still unknown, which has the same difficulty as the classical 3D Ising model due to the mapping between them, while numerical methods have estimated $\Gamma_c \approx 3.04438(2)$ [86].

# Part II

# Computational studies of classical systems

# 3 Markov chain Monte Carlo (MCMC)

As seen from the previous part, many-body systems exhibit a rich variety of physical phenomena emerged from statistics of microscopic configurations. Their exponentially large configuration spaces prohibit us from evaluating observables by exact enumerations, and motivate us to use numerical approximation methods in the studies of their properties. In this part, we focus on such techniques for classical systems, which are relatively simpler than quantum ones. We first review the traditional method of Markov chain Monte Carlo (MCMC) and the variational method, list some commonly used ansatzes to approximate the Boltzmann distribution, and discuss their limitations. Then we introduce the autoregressive models with the advantage of exact sampling, and combine the strengths of the exact sampling and the Markov chain sampling to remove the bias in the variational method.

## 3.1 Monte Carlo method

When studying an observable of a classical many-body system in Eq. (2.4), we have seen that it consists of a summation over all configurations of the system, which costs an exponentially large amount of computation if performed by an exact enumeration. Instead, to obtain the result within a practical computation budget, a general method is to generate many random samples of configuration $\{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(M)}\}$ following the target distribution $p(\boldsymbol{s})$, where $M$ is the sample size, then estimate the $p$-weighted summation over all configurations by the uniform average over only the samples:

$$\mathbb{E}_{\mathrm{MC}}[O] = \frac{1}{M}\sum_{i=1}^{M} O\left(\boldsymbol{s}^{(i)}\right). \tag{3.1}$$

This method is named Monte Carlo, after the casino that would later become famous to every computational physicist [20, 87]. It is proven that [88]

$$\mathrm{Var}_{\mathrm{MC}}[O] = \frac{1}{M}\mathrm{Var}[O], \tag{3.2}$$

where the sample variance of the estimator $\mathrm{Var}_{\mathrm{MC}}[O] = \mathbb{E}_{\mathrm{MC}}\left[(O - \mathbb{E}_{\mathrm{MC}}[O])^2\right]$ and the true variance of the observable $\mathrm{Var}[O] = \overline{(O-\bar{O})^2}$. Therefore, the sample variance vanishes and the estimator converges to the true value as $M$ increases, given that the true variance is finite, and

all samples are independently drawn from the true distribution.

## 3.2 Exact sampling

The Monte Carlo method requires randomly generating independent samples from a given distribution, but this is not always straightforward. In computational physics, the so-called "random numbers" are produced by pseudorandom number generators (PRNGs) [89], which are designed to generate statistically random values from the univariate uniform distribution $\mathcal{U}$ over an interval, such as $[0,1)$. In some simple cases, we can transform every sample from the PRNG to the target distribution in a deterministic way: Univariate distributions can be sampled by a change of variable, also known as reparameterization. For example, the exponential distribution $p(x) = \mathrm{e}^{-x}$, $x \in [0,+\infty)$ can be sampled by $x = -\ln u$, where $u$ is randomly generated from $\mathcal{U}[0,1)$, as the probability density function (PDF) satisfies

$$p(x) = q(u)\left|\frac{\partial u}{\partial x}\right| = 1 \cdot \left|\frac{\partial}{\partial x}\mathrm{e}^{-x}\right| = \mathrm{e}^{-x}. \tag{3.3}$$

Multivariate distributions can be sampled by first generating multiple independent values from $\mathcal{U}[0,1)$ and then changing the variables. For example, the 2D unit Gaussian distribution $p(x,y) = \frac{1}{2\pi}\mathrm{e}^{-\frac{1}{2}(x^2+y^2)}$ can be sampled by the Box–Muller transform [90]:

$$p(x,y) = q(u)q(v)\left\|\begin{matrix} \frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} \\ \frac{\partial v}{\partial x} & \frac{\partial v}{\partial y} \end{matrix}\right\|, \tag{3.4}$$

$$x = r\cos\theta, \quad y = r\sin\theta, \tag{3.5}$$

$$r = \sqrt{-2\ln u}, \quad \theta = 2\pi v, \tag{3.6}$$

where $u$ and $v$ are independently sampled from $\mathcal{U}[0,1)$, and $\|\cdot\|$ denotes the absolute value of the determinant of the matrix. Discrete distributions can be obtained by discretizing continuous distributions. For example, the Bernoulli distribution $p(x) = (1-p_1)\delta_{x,0} + p_1\delta_{x,1}$ with the parameter $p_1 \in [0,1]$ can be sampled by Algorithm 1. In this thesis, we refer to this kind of sampling methods as exact sampling, also known as direct sampling or perfect sampling.

**Algorithm 1** Discrete Bernoulli distribution from continuous uniform distribution.

```
Sample u from U[0,1)
if u < p_1 then
    Output x = 0, p(x) = 1 − p_1
else
    Output x = 1, p(x) = p_1
end if
```

## 3.3 Markov chain sampling

In the case of Boltzmann distributions for many-body systems with complicated energy landscapes, an analytical transformation for exact sampling is generally impossible to find, and we can only resort to indirect methods for sampling these distributions. A common method is to construct a Markov chain of samples, whose equilibrium distribution converges to the target. Therefore, this sampling method is known as the Markov chain Monte Carlo (MCMC) [91, 92]. For spin systems, such a Markov chain is usually constructed using the Metropolis–Hastings algorithm [93]. In each sampling step, we propose to update the current configuration $\boldsymbol{s}$ to a new one $\boldsymbol{s}'$ with the proposal distribution $g(\boldsymbol{s}' \mid \boldsymbol{s})$, where $\sum_{\boldsymbol{s}'} g(\boldsymbol{s}' \mid \boldsymbol{s}) = 1$ for all $\boldsymbol{s}$. Then we accept $\boldsymbol{s}'$ to become the current configuration by an acceptance probability[1] defined by

$$A(\boldsymbol{s} \to \boldsymbol{s}') = \min\left(1, \frac{p(\boldsymbol{s}')g(\boldsymbol{s} \mid \boldsymbol{s}')}{p(\boldsymbol{s})g(\boldsymbol{s}' \mid \boldsymbol{s})}\right). \tag{3.7}$$

Otherwise, we reject it and rollback to $\boldsymbol{s}$. It can be checked that the target distribution $p(\boldsymbol{s})$ is exactly an equilibrium distribution of the Markov transition matrix:

$$M_{\boldsymbol{s}_j \boldsymbol{s}_i} = A(\boldsymbol{s}_i \to \boldsymbol{s}_j) g(\boldsymbol{s}_j \mid \boldsymbol{s}_i), \tag{3.8}$$

$$\sum_{\boldsymbol{s}_i} M_{\boldsymbol{s}_j \boldsymbol{s}_i} p(\boldsymbol{s}_i) = p(\boldsymbol{s}_j). \tag{3.9}$$

If we want to sample the target distribution using a single Markov chain, i.e., the target distribution is the only equilibrium distribution, then the proposals must be ergodic, which allows the initial configuration to transition to all configurations in a finite number of samples. The simplest method to propose an update is to randomly select a spin and flip it, therefore $g(\boldsymbol{s}' \mid \boldsymbol{s}) = g(\boldsymbol{s} \mid \boldsymbol{s}')$ for all $\boldsymbol{s}$ and $\boldsymbol{s}'$, and Eq. (3.7) simplifies to

$$A(\boldsymbol{s} \to \boldsymbol{s}') = \min\left(1, \frac{p(\boldsymbol{s}')}{p(\boldsymbol{s})}\right). \tag{3.10}$$

More advanced choices of $g(\boldsymbol{s}' \mid \boldsymbol{s})$ will be discussed in Section 3.7.

When $p(\boldsymbol{s})$ is the Boltzmann distribution in Eq. (2.2), the acceptance probability further simplifies to

$$A(\boldsymbol{s} \to \boldsymbol{s}') = \min\left(1, \mathrm{e}^{\beta\left(H(\boldsymbol{s}) - H(\boldsymbol{s}')\right)}\right). \tag{3.11}$$

It avoids computing the partition function $Z$, which consists of a summation over exponentially many configurations. In the case of locally interacting systems, such as the Ising model in Eq. (2.13) where each site only interacts with a fixed number of neighbors, the energy difference of flipping a spin $s_i$ simplifies to

$$H(\boldsymbol{s}) - H(\boldsymbol{s}') = 2s_i \left( \sum_{j \text{ that interacts with } i} J_{ij} s_j \right), \tag{3.12}$$

[1] The acceptance probability depends on the configurations $\boldsymbol{s}$ and $\boldsymbol{s}'$, not to be confused with the acceptance rate defined over the whole Markov chain.

so the time to compute the acceptance probability, and therefore the total time of a sampling step to propose and accept a sample, only depends on the number of neighbors rather than the whole system size. This advantage makes MCMC a feasible method for studying large many-body systems.

After collecting many samples from the Markov chain, we estimate the observables using Eq. (3.1). Care should be taken when generating these samples, as we discuss the caveats in the following.

## 3.4 Autocorrelation time

### 3.4.1 Autocorrelation in Markov chain

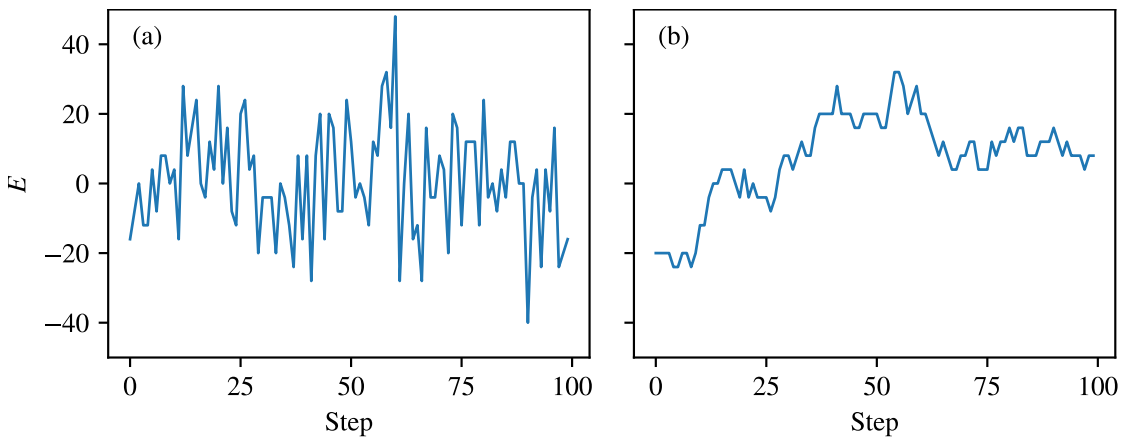



Figure 3.1: Energies of 100 samples for the Ising model on the $10 \times 10$ square lattice. (a) The samples are independently generated from the uniform distribution. (b) The samples are generated by single-spin updates with $\beta = 0$, thus have autocorrelation.

Although MCMC allows us to estimate the observables from complicated distributions where an exact sampling method cannot be easily constructed, it breaks the assumption of Eq. (3.1) that the samples should be independent. In each sampling step, we propose a new configuration $\boldsymbol{s}'$ that depends on the current configuration $\boldsymbol{s}$, such as flipping a single spin. Moreover, we may reject $\boldsymbol{s}'$ and continue using $\boldsymbol{s}$ to evaluate the observable for this step. The visual difference between a sequence of independent samples and a Markov chain is shown in Fig. 3.1. Quantitatively, the samples in the Markov chain $\{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(M)}\}$ produce autocorrelation in the observable values [94]:

$$C_{O,M}(t) = \left( \frac{1}{M-t} \sum_{i=1}^{M-t} O\left(\boldsymbol{s}^{(i)}\right) O\left(\boldsymbol{s}^{(i+t)}\right) \right) - \mathbb{E}^2_{\mathrm{MCMC}}[O], \tag{3.13}$$

where $\mathbb{E}_{\mathrm{MCMC}}[O]$ has the same form as Eq. (3.1), but uses the samples from the Markov chain.

We mainly consider the large sample size limit

$$C_O(t) = \lim_{M\to\infty} C_{O,M}(t). \tag{3.14}$$

The correlated samples cause $C_O(t) \neq 0$ when $t > 0$. For example, Fig. 3.2 shows the autocorrelation in energy $C_E(t)$ of the $10 \times 10$ Ising model. As the number of steps $t$ between two samples increases, the average autocorrelation between the two samples decays to zero. At the critical point $\beta_c = \frac{1}{2}\ln\left(1+\sqrt{2}\right) \approx 0.44$ [39], the autocorrelation decays particularly slowly, and we can assume that a sample is approximately independent of the original one only after as many as $10^4$ steps, which renders the MCMC sampling inefficient.

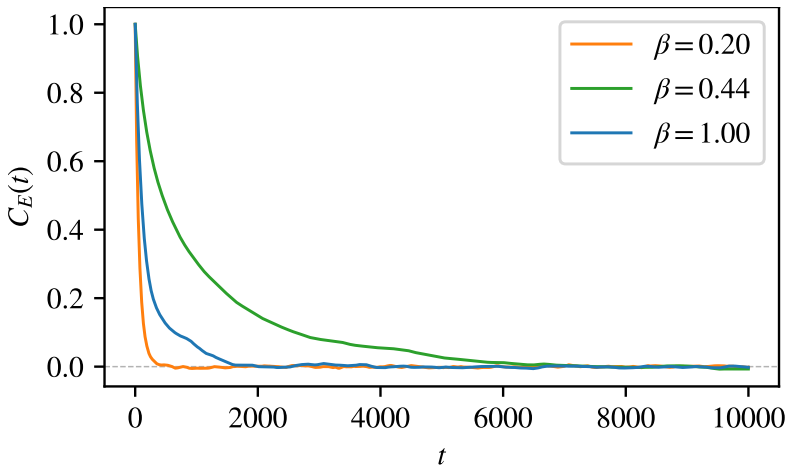



Figure 3.2: Autocorrelation in energy $C_E(t)$ of the Ising model on the $10 \times 10$ square lattice at different inverse temperatures $\beta$. The samples are generated by single-spin updates, and the autocorrelation is estimated using $10^7$ samples.

### 3.4.2 Integrated autocorrelation time

The integrated autocorrelation time [95, 96], or "autocorrelation time" for short, is a metric for MCMC to assess the loss in sample diversity due to autocorrelation, defined by

$$\tau = \sum_{t=1}^{\infty} r_O(t), \tag{3.15}$$

$$r_O(t) = \frac{C_O(t)}{C_O(0)}, \tag{3.16}$$

where $r_O(t)$ is the normalized autocorrelation. The value of $\tau$ depends on the complexity of the target distribution and the method to propose new configurations. It is also known as the relaxation time in analytical studies of spin dynamics. When estimating it in practice, $C_O(t)$ is replaced by $C_{O,M}(t)$ with a sufficiently long chain length $M$, and we need to verify $M \gg \tau$ after obtaining $\tau$. The summation $\sum_{t=1}^{\infty}$ is also replaced by a finite summation $\sum_{t=1}^{t_{\text{cutoff}}}$, where $t_{\text{cutoff}}$ is the first step when $C_{O,M}(t)$ crosses zero [1]. By the Wiener–Khinchin theorem [97], it can be

efficiently computed using the fast Fourier transform (FFT). Other methods of estimating $\tau$ include binning analysis [98], and fitting the exponential decay of $C_O(t)$ [99]. These methods should produce consistent results when the chain length $M$ is sufficiently large.

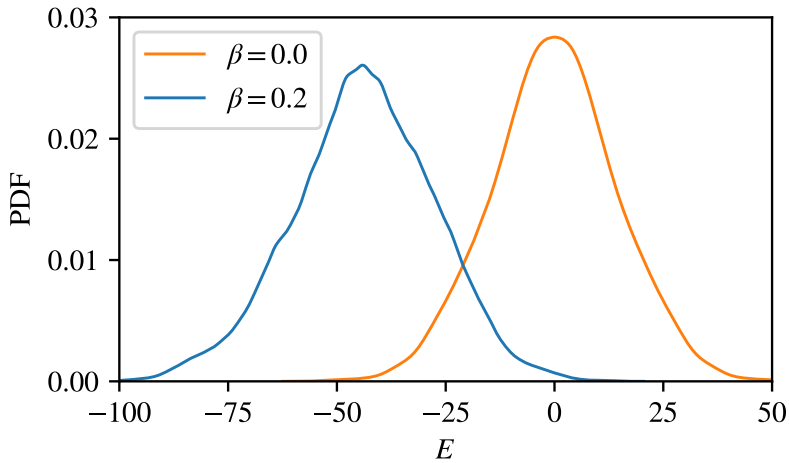



Figure 3.3: Distribution of energy $E$ for the Ising model on the $10 \times 10$ square lattice at inverse temperatures $\beta = 0$ (uniform distribution) and 0.2 (Boltzmann distribution with moderately high temperature). The typical energies from the two distributions differ by approximately 50, and for larger systems and lower temperatures the difference will be even larger, which leads to exponentially low acceptance probability. The probability density function (PDF) of the distribution is estimated from the samples using Silverman's method [100].

Intuitively, the autocorrelation time takes two factors into account that affect the sample diversity: the correlation between the proposed configuration and the current one, and the acceptance probability. For single-spin updates, the successive samples are almost fully correlated, which leads to high autocorrelation in the observable values. On the other extreme, if we do global updates, i.e., randomly generate a proposed configuration $\boldsymbol{s}'$ from the uniform distribution of all configurations in each sampling step, then it is fully uncorrelated with the current one. However, the energy $H(\boldsymbol{s}')$ from the uniform distribution is almost always much higher than the current energy from the target distribution, as shown in Fig. 3.3, which makes the acceptance probability exponentially low, and the collected samples are still almost fully correlated. Between the two extremes of single-spin updates and global updates, a variety of cluster updates have been proposed to balance these two factors, which we will discuss in Section 3.7.3.

### 3.4.3 Effective sample size

The loss in sample diversity is quantitatively reflected in the variance of the Monte Carlo estimator in Eq. (3.2). When the Markov chain is in equilibrium, in the presence of autocorrelation, the variance increases to

$$\mathrm{Var}_{\mathrm{MCMC}}[O] = \frac{1}{M_{\mathrm{eff}\,\tau}} \mathrm{Var}[O], \tag{3.17}$$

where the effective sample size

$$M_{\mathrm{eff}\,\tau} = \frac{1}{2\tau+1} M \tag{3.18}$$

is less than the original sample size $M$.

## 3.5 Critical slowing down

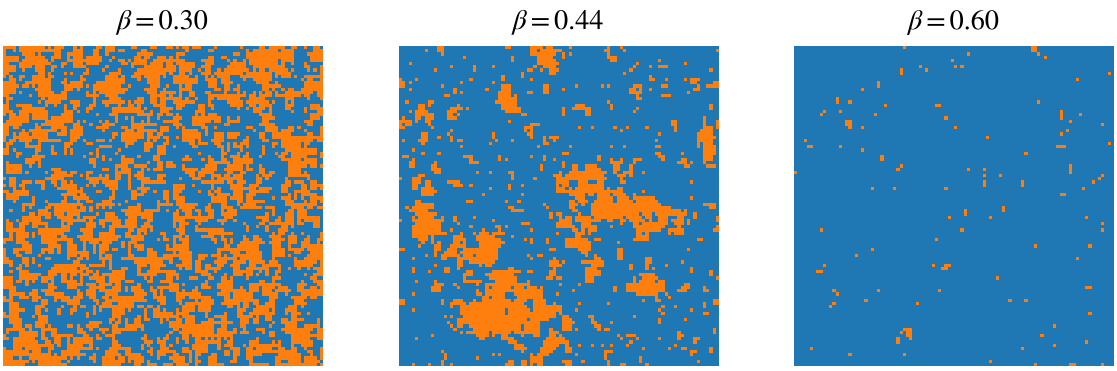



Figure 3.4: A typical sample of the Ising model on the $100 \times 100$ square lattice at different inverse temperatures. The blue and the orange plaquettes denote spin up and down respectively. Around the critical point $\beta_c \approx 0.44$, the clusters of spins are particularly visible, which pose difficulty to MCMC with single-spin updates.

In many-body systems, the autocorrelation time is particularly dependent on physical parameters, such as the inverse temperature $\beta$. It is a long-standing issue with MCMC that the autocorrelation time substantially increases when the system is near its critical point, known as the critical slowing down [101, 102]. Intuitively, the collective dynamics of the spins form clusters in the system that separate from each other by domain walls, as shown in Fig. 3.4. Their average diameter is roughly the correlation length $\xi$ of the system. In the case of single-spin update, the Markov chain can be interpreted as a random walk in the configuration space, where it moves from one configuration to an adjacent one with a spin flipped. As a corollary, the time $\tau$ to walk the distance $\xi$ scales by $\tau \sim \xi^2$, regardless of the spatial dimension. Detailed analysis from lattice field theory has shown that $\tau \sim \xi^z$, where $z$ is the dynamical critical exponent, and $z \approx 2$ in many cases [103]. At the critical point, $\xi$ diverges until it is bounded by the system size [104], therefore the critical slowing down occurs.

From the perspective of statistical complexity [105], the Boltzmann distribution at the critical temperature is difficult to sample because it has both low entropy and low uniformity, leading to high complexity. It is between the high-temperature distribution with low uniformity but high entropy, and the low-temperature distribution with low entropy but high uniformity. This phenomenon presents a challenge to studies of phase transitions using MCMC and can be alleviated by cluster updates.

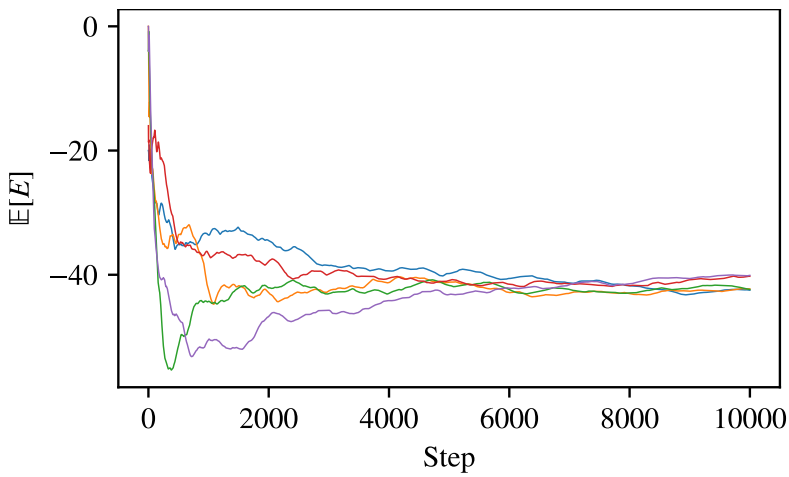



Figure 3.5: Convergence of five different Markov chains, shown with different colors, for the Ising model on the $10 \times 10$ square lattice at inverse temperature $\beta = 0.2$. The $y$-axis is the estimated energy using the samples up to the current step. Although the chains start from different random initializations, they eventually reach the equilibrium distribution and produce consistent estimations of the energy after sufficiently many steps.

## 3.6 Burn-in stage and Gelman–Rubin statistic

The autocorrelation time $\tau$ is a crucial metric for the efficiency of an MCMC algorithm when the Markov chain is already in equilibrium. However, at the beginning of MCMC, the configuration is usually initialized "randomly", which is from the uniform distribution of all configurations. The Markov chain can take many steps to transition from the initial configuration to a typical one in the target distribution, and these steps are called the burn-in or warm-up stage. The untypical samples in this stage introduce bias in the estimation of observables, unless we have exponentially many samples in total to compensate the exponentially low probabilities of untypical samples. To avoid this bias, we discard these samples and start collecting actual samples after this stage.

In theory, the length of the burn-in stage is at most on the order of magnitude of $\tau$, which lets the new configuration decorrelate with the initial one, but in practice we can estimate $\tau$ only if we already know that the burn-in stage has ended. Therefore, we need another metric to estimate the length of the burn-in stage before collecting samples from the equilibrium distribution, which can only be derived from the dynamical properties of the Markov chain.

The Gelman–Rubin (GR) statistic [106, 107] can be used to identify whether the Markov chain has reached equilibrium. To compute the GR statistic, we generate multiple independent Markov chains starting from different random initial configurations, as shown in Fig. 3.5, which is already a common practice to utilize the parallelization capability of modern computers [108]. The samples are denoted by $\boldsymbol{s}^{(j,m)}$, where $j \in [1, J]$ identifies the chain and $m \in [1, M]$ identifies the sampling step, and the corresponding observable values are denoted

by $O_{j,m} = O\left(\boldsymbol{s}^{(j,m)}\right)$. Then we have the mean values

$$\bar{O}_j = \frac{1}{M}\sum_{m=1}^{M} O_{j,m}, \tag{3.19}$$

$$\bar{O}_* = \frac{1}{J}\sum_{j=1}^{J} \bar{O}_j, \tag{3.20}$$

and the variances

$$s_j^2 = \frac{1}{M-1}\sum_{m=1}^{M} (O_{j,m} - \bar{O}_j)^2, \tag{3.21}$$

$$s_*^2 = \frac{1}{J}\sum_{j=1}^{J} s_j^2, \tag{3.22}$$

where $\bar{O}_j$ is the sample mean of the observable over a chain, $\bar{O}_*$ is the sample mean of all chains, $s_j^2$ is the sample variance of a chain, and $s_*^2$ is the mean of these sample variances.

Due to the presence of autocorrelation, each $s_j^2$ underestimates the true variance of the target distribution $\sigma^2$. A correction to this bias is estimated by comparing the means of different chains:

$$\frac{B}{M} = \frac{1}{J-1}\sum_{j=1}^{J} (\bar{O}_j - \bar{O}_*)^2. \tag{3.23}$$

Now the estimated variance is

$$\sigma_*^2 = \frac{M-1}{M} s_*^2 + \frac{B}{M}. \tag{3.24}$$

The GR statistic is defined to normalize the above correction to the estimated variance:

$$R = \sqrt{\frac{\sigma_*^2}{s_*^2}}. \tag{3.25}$$

It is also related to the effective sample size by

$$R \approx \sqrt{1 + \frac{M}{M_{\text{eff}}}}. \tag{3.26}$$

If we have enough random Markov chains from different initial configurations, which cover all regions of interest in the configuration space, and converge to similar distributions with mean observable values close to each other, then we can assume that these Markov chains have reached equilibrium. It was originally proposed in Ref. [106] to use a threshold $R < \delta$, and a common choice is $\delta = 1.1$. According to Ref. [107], practical choices of $\delta$ range from 1.003 to 1.3, and a more stable estimation of the true variance can be obtained by dividing each chain into batches and analyzing batch-wise variances.

## 3.7 Other sampling methods

### 3.7.1 Importance sampling

Apart from exact sampling and MCMC sampling, there are other sampling methods to estimate the observable in Eq. (2.4). In principle, we can generate samples $\{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(M)}\}$ from any distribution $q(\boldsymbol{s})$, then estimate the observable by

$$\mathbb{E}_{\text{imp}}[O] = \frac{\sum_{i=1}^{M} w\left(\boldsymbol{s}^{(i)}\right) O\left(\boldsymbol{s}^{(i)}\right)}{\sum_{i=1}^{M} w\left(\boldsymbol{s}^{(i)}\right)}, \tag{3.27}$$

$$w\left(\boldsymbol{s}^{(i)}\right) = \frac{p\left(\boldsymbol{s}^{(i)}\right)}{q\left(\boldsymbol{s}^{(i)}\right)}, \tag{3.28}$$

given that $q(\boldsymbol{s})$ has a wider support than $p(\boldsymbol{s})$, i.e., $q(\boldsymbol{s}) > 0$ for all $\boldsymbol{s}$ such that $p(\boldsymbol{s}) > 0$. The probabilities $p(\boldsymbol{s})$ and $q(\boldsymbol{s})$ can be unnormalized, as an overall coefficient in $w(\boldsymbol{s})$ will cancel out in the numerator and the denominator of $\mathbb{E}_{\text{imp}}[O]$. This method is called importance sampling in statistics [109, 110], and the distribution $q(\boldsymbol{s})$ is called an ansatz[2] in the context of statistical physics. The ansatz should be close to $p(\boldsymbol{s})$, and easier to sample than $p(\boldsymbol{s})$. Some choices of the ansatz for importance sampling will be discussed in Section 4.5.1. The variance of this estimator is

$$\text{Var}_{\text{imp}}[O] = \frac{1}{M_{\text{eff}\,w}} \text{Var}[O], \tag{3.29}$$

where the effective sample size

$$M_{\text{eff}\,w} = \frac{\left(\sum_{i=1}^{M} w\left(\boldsymbol{s}^{(i)}\right)\right)^2}{\sum_{i=1}^{M} w^2\left(\boldsymbol{s}^{(i)}\right)}. \tag{3.30}$$

The variance reaches its minimum when $q(\boldsymbol{s}) = p(\boldsymbol{s})$, which is in accordance with the intuition that $q(\boldsymbol{s})$ should be as close to $p(\boldsymbol{s})$ as possible.

### 3.7.2 MCMC importance sampling

In addition to reweighing the samples from $q(\boldsymbol{s})$ using the target distribution $p(\boldsymbol{s})$, we can also use the Metropolis–Hastings algorithm in Eq. (3.7) to reject the samples, which we refer to as MCMC importance sampling [111, 112]. In each sampling step, we propose a new configuration $\boldsymbol{s}'$ from the distribution $q(\boldsymbol{s}')$, which is independent of the current configuration $\boldsymbol{s}$, i.e., the proposal distribution $g(\boldsymbol{s}' \mid \boldsymbol{s}) = q(\boldsymbol{s}')$. Then the acceptance probability becomes

$$A(\boldsymbol{s} \to \boldsymbol{s}') = \min\left(1, \frac{p(\boldsymbol{s}')q(\boldsymbol{s})}{p(\boldsymbol{s})q(\boldsymbol{s}')}\right), \tag{3.31}$$

[2] The word "ansatz" is borrowed from German. Its plural form is "Ansätze" in German, but the adapted form "ansatzes" is also used in the English literature.

which is close to 1 if $q(\boldsymbol{s})$ is close to $p(\boldsymbol{s})$. While increasing the acceptance probability, this method also reduces the correlation between $\boldsymbol{s}$ and $\boldsymbol{s}'$ compared to the single-spin update, which leads to an overall reduction in the autocorrelation time. An empirical comparison of importance sampling and MCMC importance sampling will be presented in Section 5.4.1. More advanced methods have been proposed to construct the proposal distribution $g(\boldsymbol{s}' \mid \boldsymbol{s})$ depending on the current configuration $\boldsymbol{s}$, such as Langevin Monte Carlo [113] and Hamiltonian Monte Carlo [114].

### 3.7.3 Cluster update

A variety of cluster update algorithms can also be discussed in the framework of MCMC importance sampling. They originally arose from studies of the percolation transition in statistical physics and graph theory, or the series expansion of the partition function [115–117]. A prominent example is the Wolff algorithm [118] for Ising models with two-body interactions. In each sampling step, it constructs and flips a cluster of spins according to Algorithm 2. The resulting proposal probability satisfies $g(\boldsymbol{s}' \mid \boldsymbol{s}) \propto p_{\mathrm{B}}(\boldsymbol{s}')$, so the acceptance probability is always 1, which makes the Wolff cluster update far more efficient than the single-spin update when applicable.

**Algorithm 2** Wolff cluster update algorithm to construct and flip a cluster of spins, in the MCMC sampling of the Boltzmann distribution of Ising models with two-body interactions. $\mathcal{C}$ is the cluster, $\mathcal{B}$ is a set of spins on the boundary of the cluster, $\beta$ is the inverse temperature, $\partial i$ denotes the neighbors of the site $i$, and `rand()` generates a random number from the uniform distribution over $[0, 1)$.

```
Input the current configuration s, interactions J
Randomly choose a site i, C ← {i}, B ← {i}
while B ≠ ∅ do
    i ← any site in B
    for j ∈ ∂i do
        if j ∉ C and rand() > e^{2βJ_ij s_i s_j} then
            C ← C ∪ {j}, B ← B ∪ {j}
        end if
    end for
    B ← B \ {i}
end while
for i ∈ C do
    s_i ← −s_i
end for
Output the new configuration s′ ← s
```

## 3.8 Mode collapse

Even if we collect effectively independent samples according to the autocorrelation time and the Gelman–Rubin statistic, it can still be questionable whether those samples actually represent the entire target distribution. The target distribution can have multiple modes, where a "mode" is defined to be a cluster of configurations $\{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \ldots\}$ with high target probabilities $p(\boldsymbol{s}_i)$ and high Markov transition probabilities $M_{\boldsymbol{s}_j \boldsymbol{s}_i}$ in Eq. (3.8) between each other. The paths of configurations to transition from one mode to another always encounter energy barriers whose heights are proportional to the system size, so the probability of visiting different modes is exponentially low. A possible failure of MCMC is the mode collapse, where the Markov chain is seemingly in equilibrium, but actually it is only in "equilibrium" within a single mode, and it takes an impractically long time to visit another mode, which leads to bias in the estimation of observables.

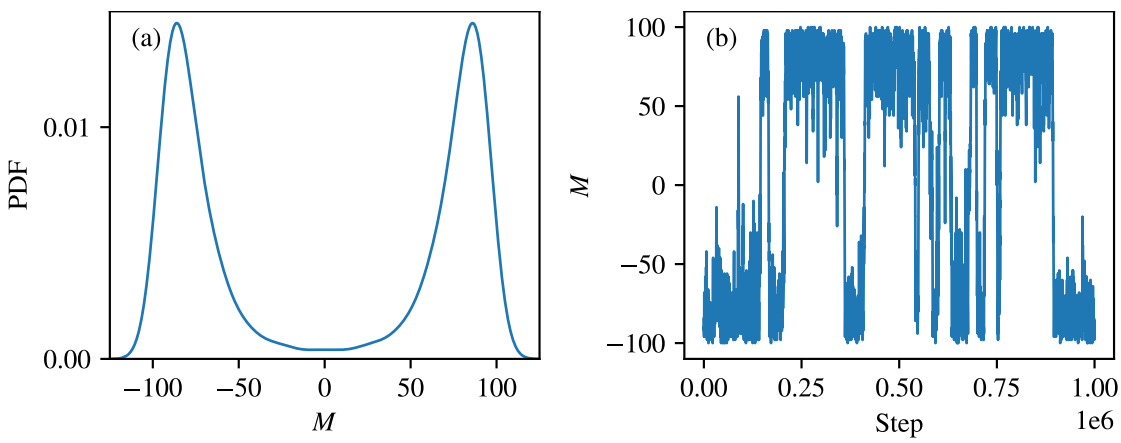



Figure 3.6: (a) Distribution of magnetization $M$ for the Ising model on the $10 \times 10$ square lattice at inverse temperature $\beta = 0.44$. The two modes characterized by the magnetization are clearly visible. The energy barrier between the two modes will be even higher as the system size increases, leading to exponentially low probability of transitioning from one mode to another. The probability density function (PDF) of the distribution is estimated from the samples using Silverman's method [100].
(b) Magnetizations of samples generated by single-spin updates. The Markov chain can stay in a mode for as long as $10^5$ steps before transitioning to another mode.

Intuitively, a mode is a peak of the target distribution in the configuration space. Although the exponentially high-dimensional configuration space cannot be directly visualized, we can characterize different modes by appropriate observables, such as the energy or the magnetization, and display the modes as peaks on the histogram of the observable. When mode collapse occurs, the estimated observable characterizing the uncovered modes becomes biased. A typical example consists of the two modes around the two degenerate ground states $\boldsymbol{s} = (+1, +1, +1, \ldots)$ and $\boldsymbol{s} = (-1, -1, -1, \ldots)$ in the ferromagnetic Ising model, as shown in Fig. 3.6. In this case, even if the estimated energy is still close to the true value in a single mode, the magnetization becomes biased.

The mode collapse in the simple case above can be eliminated by applying the spin flipping symmetry to the generated samples. Other symmetries of the Hamiltonian, such as translations, rotations, and reflections, are also useful to alleviate the mode collapse and improve the accuracy of the estimated observables. Cluster updates are another approach to alleviate this problem, as they can propose large updates to the configuration to transition through the energy barriers that block local updates. The problem becomes more challenging in frustrated models, as discussed in Section 2.1.3, where the number of degenerate ground states can be exponentially large and cannot be eliminated by a polynomially large symmetry group [53–55].

The effect of mode collapse is also demonstrated in the context of importance sampling in Eq. (3.27). Assume that the actual distribution of the samples $q(\boldsymbol{s})$ covers only a single mode and is exponentially suppressed in the uncovered modes, where $p(\boldsymbol{s}) > 0$ but $q(\boldsymbol{s}) \to 0$. Then we have pathologically high $w(\boldsymbol{s})$ in the uncovered modes, which cause the variance in Eq. (3.29) to diverge. These configurations are identified as exponentially suppressed configurations (ESCs) in Ref. [1]. Even if an estimated observable is still close to the true value in a single mode, the diverging variance indicates that such an estimation is unreliable.

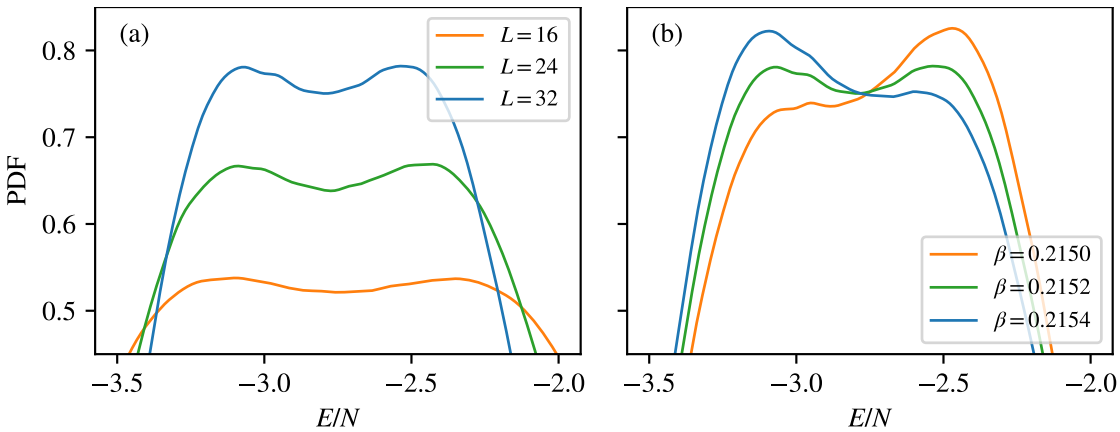



Figure 3.7: Distribution of the energy per site $E/N$ for the FPM in Section 2.1.6. (a) The results with different system sizes, at their respective "phase transition" temperatures where the two peaks have the same area. (b) The results at different temperatures, with the system size $L = 32$. The probability density function (PDF) of the distribution is estimated from the samples using Silverman's method [100]. The panel (a) is reproduced from Fig. 5 in Ref. [1].

A particular consequence of mode collapse is the difficulty of MCMC sampling around first-order phase transitions of many-body systems [119]. The first-order phase transition is characterized by a discontinuity in the energy. As an example, we investigate the transition between the ferrimagnetic (fM) and the paramagnetic (PM) phases in the 2D frustrated plaquette model (FPM) in Section 2.1.6, with $N = L \times L$ spins. When we plot its distribution of energies in Fig. 3.7, we can see two peaks appearing around the transition temperature. As shown in panel (a), the depth of the valley between the two peaks increases as the system size $L$ increases, which indicates an increasing energy barrier that blocks local updates. Compared with the peak on the left, the one on the right contains more excited states, but each has

a lower probability, so the areas of the two peaks are comparable. When the temperature increases, the low-energy peak shrinks its area while the high-energy one grows, as shown in panel (b).

The first-order phase transition appears when the two peaks happen to have the same area. In the thermodynamic limit, the probability distribution is exponentially concentrated at the largest peak, so a change of the largest peak causes a discontinuity in the mean energy. Around the phase transition temperature, it is difficult for the system to reach thermal equilibrium, and the system can stay in a metastable state or can be described by a superposition of multiple classical states. This two-peak distribution proposes a challenge to MCMC with local updates, and cluster updates are a viable approach to this problem, as we will discuss in Section 5.4.1.

# 4 Variational method

The MCMC method in the previous chapter is widely used to compute various observables of many-body systems. However, a limitation is that it is agnostic of the partition function $Z$ in Eq. (2.2), therefore it cannot compute quantities that require $Z$ or the normalized probability $p_{\mathrm{B}}(\boldsymbol{s})$, including the free energy and the entropy. Variants of MCMC have been proposed to compute $Z$, such as the Wang–Landau algorithm [120, 121]. However, they also involve other issues such as the discretization of the energy histogram [122], which limit their use in practical computations.

A different approach is to track the probability distribution in the framework of variational approximation and statistical inference [25, 123], which also has deep roots in statistical physics, traced back to mean-field theories [124, 125]. We define a trial distribution $q(\boldsymbol{s})$, or the "ansatz" in the context of statistical physics, to approximate the target distribution $p(\boldsymbol{s})$. The form of the ansatz is chosen so that the normalized probability $q(\boldsymbol{s})$ can be efficiently computed given a configuration $\boldsymbol{s}$, and the distribution can be easier to sample than $p(\boldsymbol{s})$, or even be analytically summed over in Eq. (2.4). Some choices of the ansatz for variational inference will be discussed in Section 4.5.1.

## 4.1 Variational free energy

To make a good approximation, we want the ansatz $q(\boldsymbol{s})$ to be as close to the target $p(\boldsymbol{s})$ as possible. A natural metric of the distance between two distributions is the Kullback–Leibler (KL) divergence [126]:

$$D_{\mathrm{KL}}(q \,||\, p) = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \ln \frac{q(\boldsymbol{s})}{p(\boldsymbol{s})}. \tag{4.1}$$

When $p(\boldsymbol{s})$ is the Boltzmann distribution, we have

$$D_{\mathrm{KL}}(q \,||\, p_{\mathrm{B}}) = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left( \ln q(\boldsymbol{s}) - \ln \frac{\mathrm{e}^{-\beta H(\boldsymbol{s})}}{Z} \right) \tag{4.2}$$

$$= \beta(F_q - F), \tag{4.3}$$

where $F = -\frac{1}{\beta}\ln Z$ is the true free energy, and we define the variational free energy

$$F_q = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) F_{\text{loc}}(\boldsymbol{s}), \tag{4.4}$$

and the local free energy

$$F_{\text{loc}}(\boldsymbol{s}) = \frac{1}{\beta}\ln q(\boldsymbol{s}) + H(\boldsymbol{s}). \tag{4.5}$$

Similar to how we have approximated Eq. (2.4) by Eq. (3.1), the variational free energy in Eq. (4.4) is also a weighted sum over the distribution $q(\boldsymbol{s})$ with exponentially many terms, and we can estimate it by Monte Carlo sampling:

$$\mathbb{E}_{\text{MC}}[F_q] = \frac{1}{M}\sum_{i=1}^{M} F_{\text{loc}}\left(\boldsymbol{s}^{(i)}\right), \quad \boldsymbol{s}^{(i)} \sim q. \tag{4.6}$$

This method has close resemblance to the variational Monte Carlo (VMC) method for quantum systems, which we will discuss in the next part, although the term VMC is rarely used in the context of classical systems. It is worth noting that variational inference requires one to evaluate the normalized probability $q(\boldsymbol{s})$, in contrast to MCMC in the previous chapter where the normalization constant cancels out in the derivation of the Markov chain, as in Eq. (3.7). Therefore, it is important to choose an ansatz $q(\boldsymbol{s})$ whose probability is tractable.

## 4.2 Minimization of variational free energy

The ansatz $q(\boldsymbol{s})$ is usually controlled by some parameters $\theta$, which allows us to minimize the KL divergence by varying $\theta$. In Eq. (4.3), the term $F$ is independent of $\theta$, so we only need to minimize $F_q$. As the KL divergence is non-negative, the lower bond of $F_q$ among all possible ansatzes is exactly the true free energy $F$, which is desired to compute in many cases. This property is known as the Bogolubov inequality [127]. A common practice is to minimize $\beta F_q$ rather than $F_q$ itself, which avoids the singularity of $\frac{1}{\beta}$ in Eq. (4.4) when $\beta \to 0$.

### 4.2.1 Gradient descent optimizers

As $F_q$ is generally a multivariate non-linear function of $\theta$, its optimization is usually performed by gradient descent (GD)-based algorithms [128]. In the naive setting of GD, in each optimization step $t$ we update $\theta$ by

$$\theta_{t+1} \leftarrow \theta_t - \gamma\, \nabla F_q\big|_{\theta=\theta_t}, \tag{4.7}$$

where $\nabla F_q = \frac{\partial F_q}{\partial \theta}$ is the gradient vector of the target function, and $\gamma$ is a small positive number called the learning rate. A common challenge in GD is to choose an appropriate magnitude of $\gamma$, such that the optimization is neither unstable nor too slow to converge [129]. Another challenge is to converge to the global minimum of the target function, rather than to be trapped by local minima and saddle points. It is particularly important in physical problems

to find the global minimum accurately. In contrast to typical machine learning problems, such as computer vision, where one only seeks a "good-looking" solution that simulates the inaccurate nature of human vision, in physics there is great interest in obtaining quantitative properties to high accuracies. It is even more crucial to identify qualitative properties about the global minimum, such as symmetries of the ground state. This demand poses unique challenges to the optimization algorithms we use [130, 131].

When estimating the gradient in Eq. (4.7), we usually generate a set of random samples $\{\boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(M)}\}$ in each sampling step, rather than sum over the exponentially large space of configurations. This practice is known as the stochastic gradient descent (SGD) [132, 133], and the sample size $M$ is also known as the "batch" size in machine learning. In the continuous-time limit, SGD is equivalent to exact gradient flow with an additional Brownian noise [134], and the variance of the noise scales by $O\left(\frac{1}{M}\right)$. This Brownian motion is also equivalent to a simulated annealing, whose equilibrium distribution is a Boltzmann distribution[1] in the space of the parameter $\theta$. In other words, this noise helps the optimizer escape from local minima and saddle points, given an appropriate ratio of the learning rate to the batch size, which explains the successful application of SGD in various non-convex problems.

Many advanced variants of SGD have been proposed in the recent trend of machine learning [135]. In particular, the Adam optimizer [136] has become the most popular among them. It estimates the appropriate learning rate for each parameter by the second moment[2] of the gradient, and uses the first moment to improve convergence, which largely reduces manual labor to tune the learning rate. Numerous experiments have demonstrated its broad applicability to machine learning problems, although its convergence properties are still debatable in physical problems that require high accuracy.

### 4.2.2 Second-order optimizers

We refer to GD as a first-order optimizer[3], because it approximates the target function with only the first-order term in the Taylor expansion, which is the gradient. There are also various second-order optimizers that approximate the target function with a quadratic function, and utilize information about the Hessian matrix to precondition the gradient, i.e., to modify the direction and the magnitude of the gradient in order to reduce the condition number of the optimization problem and accelerate the convergence. The second-order optimizers date back to the Newton's method, which updates $\theta$ by

$$\theta_{t+1} \leftarrow \theta_t - \gamma\, \boldsymbol{H}^{-1} \nabla F_q \big|_{\theta=\theta_t}, \tag{4.8}$$

where $\boldsymbol{H}$ is the Hessian matrix of the target function defined by $H_{ij} = \frac{\partial^2 F_q}{\partial\theta_i \partial\theta_j}$. In this case, the learning rate $\gamma$ can be close to 1 if the target function is close to the quadratic approximation. However, this method can easily become unstable with highly non-linear target functions,

[1] Not to be confused with the physical Boltzmann distribution in the space of the configurations $\boldsymbol{s}$.

[2] It is the second moment of the norm of each gradient component in the case of complex-valued gradients.

[3] Not to be confused with the order of the convergence speed, which is related to the order in the definition of the optimizer, but not always equivalent.

and an interpolation between the first- and the second-order updates is required to enhance stability, such as the Levenberg–Marquardt algorithm [137]. Moreover, directly storing $\boldsymbol{H}$ requires $O(n_\mathrm{p}^2)$ space, where $n_\mathrm{p}$ is the number of parameters, and directly inverting it requires $O(n_\mathrm{p}^3)$ computation, both of which are impractical with large numbers of parameters in neural networks. Therefore, modern methods approximate $\boldsymbol{H}^{-1}$ with low-rank updates and line searches, such as the conjugate gradient method [138] and the limited-memory Broyden–Fletcher–Goldfarb–Shanno (L-BFGS) algorithm [139], or precondition it in subspaces [140]. Even with these improvements, the use of second-order optimizers is still hardly practical in large-scale machine learning problems, let alone higher-order ones.

When the target function is a KL divergence, a special technique is to replace $\boldsymbol{H}$ in Eq. (4.8) by the Fisher information matrix

$$\boldsymbol{S} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \nabla \ln q(\boldsymbol{s}) \left(\nabla \ln q(\boldsymbol{s})\right)^\top . \tag{4.9}$$

While in principle the gradient can be preconditioned by the inverse of any positive definite matrix, the Fisher information matrix is particularly useful, because it is a natural metric in the function space where the KL divergence measures the distance [141]. This technique is known as natural gradient descent (NGD) [142], and has led to another line of research to approximate it, such as the Kronecker factorization (K-FAC) [143] and online approximations [144, 145]. We will give more discussion on NGD in the context of quantum problems in Section 7.4, where first-order optimizers produce unsatisfactory results and NGD is considered necessary. Moreover, local and sequential updates can be applied to avoid large matrix inversions [146], which shares the spirit with the neuron-wise and layer-wise training before the modern large-scale training was possible [147].

## 4.3 Gradient estimator

The gradient $\nabla F_q$ in Eq. (4.7) is derived by automatic differentiation (AD) [148] in modern software[4], which can be more accurate and efficient than traditional methods of finite difference and more prone to errors than manually deriving and implementing the gradient. Care should be taken that Eq. (4.4) involves the distribution $q(\boldsymbol{s})$, which is a function of $\theta$, but this term no longer exists after replacing Eq. (4.4) by Eq. (4.6), and we cannot take the gradient of this term in the usual way after sampling the configurations in Eq. (4.6). Therefore, we rewrite the gradient of Eq. (4.4) as

$$\frac{\partial F_q}{\partial \theta} = \sum_{\boldsymbol{s}} \left( \frac{\partial q(\boldsymbol{s})}{\partial \theta} F_\mathrm{loc}(\boldsymbol{s}) + q(\boldsymbol{s}) \frac{\partial F_\mathrm{loc}(\boldsymbol{s})}{\partial \theta} \right) \tag{4.10}$$

$$= \sum_{\boldsymbol{s}} \left( q(\boldsymbol{s}) \frac{\partial \ln q(\boldsymbol{s})}{\partial \theta} F_\mathrm{loc}(\boldsymbol{s}) + q(\boldsymbol{s}) \frac{1}{\beta} \frac{\partial \ln q(\boldsymbol{s})}{\partial \theta} \right). \tag{4.11}$$

[4] Specifically, the technique to derive the gradient of a scalar target function w.r.t. many parameters is named backpropagation, or reverse mode AD.

The second term of Eq. (4.11) becomes

$$\frac{1}{\beta}\sum_{\boldsymbol{s}} q(\boldsymbol{s})\frac{\partial \ln q(\boldsymbol{s})}{\partial\theta} \tag{4.12}$$

$$= \frac{1}{\beta}\sum_{\boldsymbol{s}} \frac{\partial q(\boldsymbol{s})}{\partial\theta} \tag{4.13}$$

$$= \frac{1}{\beta}\frac{\partial}{\partial\theta}\sum_{\boldsymbol{s}} q(\boldsymbol{s}). \tag{4.14}$$

As we have the normalization $\sum_{\boldsymbol{s}} q(\boldsymbol{s}) = 1$, this term is always zero. The remaining first term of Eq. (4.11) can be again estimated by Monte Carlo sampling:

$$\mathbb{E}_{\mathrm{MC}}\left[\frac{\partial F_q}{\partial\theta}\right] = \frac{1}{M}\sum_{i=1}^{M} F_{\mathrm{loc}}\left(\boldsymbol{s}^{(i)}\right)\frac{\partial \ln q\left(\boldsymbol{s}^{(i)}\right)}{\partial\theta}, \quad \boldsymbol{s}^{(i)} \sim q. \tag{4.15}$$

This method is called the REINFORCE gradient [149] or the policy gradient [150] in the context of machine learning, and the gradient of the log-probability $\frac{\partial \ln q(\boldsymbol{s})}{\partial\theta}$ is called the score function [151–153]. This kind of differentiation through discrete variables and stochastic sampling of distributions is a recent interest in AD frameworks [154–156].

Moreover, a common practice is to shift the terms in the above gradient estimator to have zero means:

$$\mathbb{E}_{\mathrm{MC}}\left[\frac{\partial F_q}{\partial\theta}\right] = \frac{1}{M}\sum_{i=1}^{M}\left(F_{\mathrm{loc}}\left(\boldsymbol{s}^{(i)}\right) - \tilde{F}\right)\left(\frac{\partial \ln q\left(\boldsymbol{s}^{(i)}\right)}{\partial\theta} - \tilde{\boldsymbol{D}}\right), \quad \boldsymbol{s}^{(i)} \sim q, \tag{4.16}$$

$$\tilde{F} = \mathbb{E}_{\mathrm{MC}}[F_q], \tag{4.17}$$

$$\tilde{\boldsymbol{D}} = \mathbb{E}_{\mathrm{MC}}\left[\frac{\partial \ln q}{\partial\theta}\right], \tag{4.18}$$

and we do not take the gradient of $\tilde{F}$ and $\tilde{\boldsymbol{D}}$ w.r.t. $\theta$. Although $\tilde{F}$ and $\tilde{\boldsymbol{D}}$ do not affect the expectation of the estimator, they reduce the variance of the estimator in many practical cases and thus improve the convergence of SGD. This technique falls into a broader family of control variate methods to reduce variances [157, 158].

## 4.4 Comparison between MCMC and variational method

Besides being able to estimate observables that require the partition function, another feature of variational inference in contrast to MCMC is that it always produces an upper bound of the true free energy, as long as the statistical error of Monte Carlo sampling is sufficiently small. This leads to an intuitive way to compare the accuracies of two variational methods: a lower variational free energy indicates a better approximation of the target distribution. However, this also means a systematic bias in the estimated energy. Other estimated observables can contain biases with unknown directions, and a lower variational free energy does not always imply that other estimated observables are more accurate.

Moreover, there is no straightforward way to improve the accuracy of variational inference

as the computation budget grows, whether by increasing the number of variational parameters or optimization steps. In contrast, we can always improve the accuracy or the probability of covering all modes in MCMC by increasing the number of samples. Therefore, to obtain unbiased and accurate estimations of observables, we may use a variational ansatz with MCMC importance sampling in Eq. (3.31).

In classical statistical physics, certain variational ansatzes have been constructed for specific physical systems, such as variants of the Heisenberg model [159–161]. However, before neural networks became popular, it had been difficult to construct ansatzes that are generalizable to more complicated systems and expressive enough to capture the rich variety of their physical properties. In the following, we discuss some commonly used ansatzes for classical spin systems. They may be easier to sample than the target Boltzmann distribution, thus can be used with MCMC importance sampling in Eq. (3.31); or they may support efficient evaluation of the normalized probability, which is crucial for variational inference in Eq. (4.4).

## 4.5 Variational ansatzes for classical spin systems

### 4.5.1 Naive mean-field ansatz

In the mean-field approach, we construct an approximated Hamiltonian in which all spins are independent. The original interactions between the spins are represented by the "mean fields" interacting with the spins, which are not summed over when evaluating the observables. An early practice of this approach was applied to the Curie–Weiss model [38], which is able to describe the phase transition of ferromagnets. Here, we review its derivation and reformulate it in the framework of variational inference.

The Hamiltonian in Eq. (2.1) can be written as the sum of each spin $s_i$ in its local magnetic field $h_{\mathrm{loc}\,i}$ produced by adjacent spins:

$$H(\boldsymbol{s}) = J \sum_{\langle i,j \rangle} s_i s_j = \frac{1}{2} J \sum_i h_{\mathrm{loc}\,i} s_i, \tag{4.19}$$

$$h_{\mathrm{loc}\,i} = \sum_{j \in \partial i} s_j, \tag{4.20}$$

where $\partial i$ denotes the set of nearest neighbors of the site $i$. To find an ansatz whose free energy can be analytically studied, we start from replacing $h_{\mathrm{loc}\,i}$ by an approximate mean field produced by all spins:

$$h_{\mathrm{loc}\,i} \approx h_{\mathrm{MF}} = \frac{2d}{N} \sum_i s_i, \tag{4.21}$$

where $d$ is the average number of interactions per site, and each site has $2d$ nearest neighbors. Then the locally interacting model becomes a globally interacting mean-field model

$$H(\boldsymbol{s}) \approx H_{\mathrm{MF}}(\boldsymbol{s}) = \frac{dJ}{N} \sum_{ij} s_i s_j. \tag{4.22}$$

Next, we split each spin variable into $s_i = m + \Delta s_i$, where $m$ is the mean magnetization,

$\Delta s_i$ is the fluctuation around the mean, and we assume $|\Delta s_i| \ll |m|$. Then Eq. (4.22) becomes

$$H_{\mathrm{MF}}(\boldsymbol{s}) = \frac{dJ}{N}\sum_{ij}(m+\Delta s_i)(m+\Delta s_j) \tag{4.23}$$

$$= dJm\sum_i(m+2\Delta s_i)+\mathcal{O}(\Delta s^2) \tag{4.24}$$

$$= dJm\sum_i(2s_i-m)+\mathcal{O}(\Delta s^2). \tag{4.25}$$

Ignoring the higher-order terms $\mathcal{O}(\Delta s^2)$, the mean-field model can be written in a non-interacting form:

$$H_{\mathrm{MF}}(\boldsymbol{s}) = \sum_i H_{\mathrm{MF}\,i}(s_i), \tag{4.26}$$

$$H_{\mathrm{MF}\,i}(s_i) = dJm(2s_i-m). \tag{4.27}$$

Thus we can perform the summation of the partition function:

$$Z_{\mathrm{MF}} = \left(\sum_s \mathrm{e}^{-\beta H_{\mathrm{MF}\,i}(s)}\right)^N = \left(\mathrm{e}^{d\beta Jm^2}\cdot 2\cosh\big(2d\beta Jm\big)\right)^N. \tag{4.28}$$

The free energy is

$$F_{\mathrm{MF}} = -\frac{1}{\beta}\ln Z_{\mathrm{MF}} = -\frac{N}{\beta}\left(d\beta Jm^2+\ln\big(2\cosh\big(2d\beta Jm\big)\big)\right). \tag{4.29}$$

At this point, the mean-field free energy $F_{\mathrm{MF}}$ is derived from the approximated Hamiltonian in Eq. (4.22), and we have no guarantee that it is an upper bound of the true free energy derived from the original Hamiltonian in Eq. (2.1). However, Eq. (4.3) still motivates us to minimize $F_{\mathrm{MF}}$, and the tunable parameter $\theta$ is $m$ here. When $F_{\mathrm{MF}}$ reaches its minimum, we have the self-consistent equation

$$\frac{\partial F_{\mathrm{MF}}}{\partial m} = 0 \implies m+\tanh\big(2d\beta Jm\big) = 0. \tag{4.30}$$

As shown in Fig. 4.1, non-zero solutions to Eq. (4.30) appear only when $2d\beta J < -1$, where $J < 0$ and the critical temperature $T_{\mathrm{c}} = \frac{1}{\beta_{\mathrm{c}}} = 2d(-J)$. Therefore, at low temperatures $T < T_c$, the system is ferromagnetic with spontaneous magnetization as the non-zero solutions of Eq. (4.30); otherwise, at high temperatures $T > T_c$, the system is paramagnetic without spontaneous magnetization.

The non-interacting Hamiltonian Eq. (4.27) also implies that the joint probability $p(\boldsymbol{s})$ can be factorized into univariate probabilities:

$$p_{\mathrm{MF}}(\boldsymbol{s}) = \prod_i p_{\mathrm{MF}\,i}(s_i), \tag{4.31}$$

$$p_{\mathrm{MF}\,i}(s_i) = \frac{\mathrm{e}^{-\beta H_{\mathrm{MF}\,i}(s_i)}}{\sum_s \mathrm{e}^{-\beta H_{\mathrm{MF}\,i}(s)}}. \tag{4.32}$$

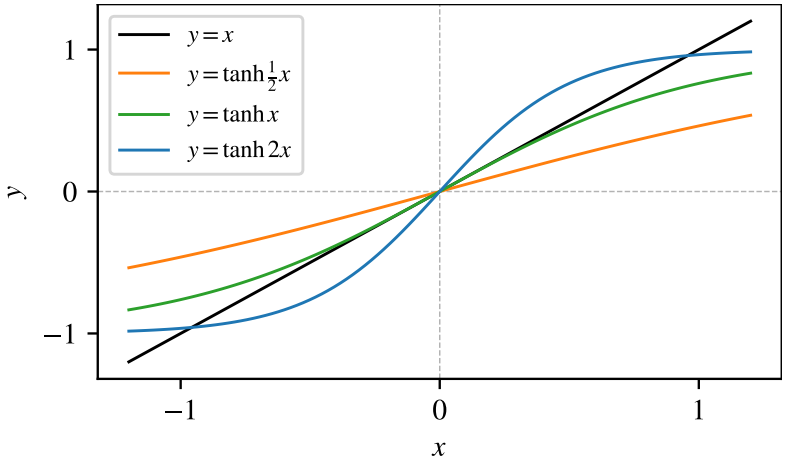



Figure 4.1: Graphical solution to the Curie–Weiss self-consistent equation in Eq. (4.30), where $x = -m$. The equation $x = \tanh ax$ has non-zero solutions only when $a = 2d\beta(-J) > 1$.

Simplifying $p_{\mathrm{MF}}(s_i)$ with Eq. (4.30), we have

$$p_\uparrow = p_{\mathrm{MF}\,i}(s_i = +1) = \frac{1+m}{2}, \quad p_\downarrow = p_{\mathrm{MF}\,i}(s_i = -1) = \frac{1-m}{2}. \tag{4.33}$$

The probability distribution $p_{\mathrm{MF}}(\boldsymbol{s})$ can serve as a variational ansatz with a single tunable parameter $m$, which we refer to as the naive mean-field (NMF) ansatz.

Next, we use the NMF ansatz and the original Hamiltonian in Eq. (2.1) to evaluate the variational energy. The energy can be directly evaluated by

$$E_{\mathrm{MF}} = \sum_{\boldsymbol{s}} p_{\mathrm{MF}}(\boldsymbol{s}) H(\boldsymbol{s}) \tag{4.34}$$

$$= NdJ(p_\uparrow p_\uparrow - p_\uparrow p_\downarrow - p_\downarrow p_\uparrow + p_\downarrow p_\downarrow) \tag{4.35}$$

$$= NdJm^2, \tag{4.36}$$

and the entropy is

$$S_{\mathrm{MF}} = -\sum_{\boldsymbol{s}} p_{\mathrm{MF}}(\boldsymbol{s}) \ln p_{\mathrm{MF}}(\boldsymbol{s}) \tag{4.37}$$

$$= -N(p_\uparrow \ln p_\uparrow + p_\downarrow \ln p_\downarrow). \tag{4.38}$$

Under this ansatz, the free energy $F_{\mathrm{MF}} = E_{\mathrm{MF}} - \frac{1}{\beta} S_{\mathrm{MF}}$ is the same as Eq. (4.29) derived from the approximated Hamiltonian. It is indeed an upper bound of the true free energy, as it is now derived from the variational ansatz on the original Hamiltonian. For more complicated models on non-regular graphs and with more than two-body interactions, where a self-consistent equation like Eq. (4.30) cannot be easily constructed, we can still apply the NMF ansatz to obtain a preliminary variational approximation.

### 4.5.2 Bethe ansatz

Starting from an analytical solution to the 1D quantum Heisenberg model [162], the Bethe ansatz has become a gross term for methods to exactly solve or approximate partition functions on lattices and other graphs [46, 48, 163, 164]. It is also deeply related to belief propagation, which is a message passing algorithm used in Bayesian inference [165, 166], tensor networks [167, 168], and the recent trend of graph machine learning [169]. Unlike the NMF ansatz which approximates the original model using only uniform global interactions, the Bethe ansatz utilizes the local geometry to more accurately represent interactions between nearest neighbors.

In general, the Bethe ansatz is applicable to any probability distribution defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of sites and $\mathcal{E}$ is the set of edges, as long as the probability of each configuration can be factorized into two-body interactions:

$$p(\boldsymbol{s}) = \frac{1}{Z} \prod_{(i,j)\in\mathcal{E}} f_{ij}(s_i, s_j). \tag{4.39}$$

For the Ising model in Eq. (2.1), we have

$$f_{ij}(s_i, s_j) = \mathrm{e}^{-\beta J s_i s_j}. \tag{4.40}$$

To study the properties of an edge or a site while marginalizing the rest of the graph, we define the auxiliary probability with an edge $(ij)$ removed from the graph:

$$\mu^{(ij)}(\boldsymbol{s}) = \frac{1}{Z^{(ij)}} \prod_{(k,l)\in\mathcal{E}\setminus(ij)} f_{kl}(s_k, s_l), \tag{4.41}$$

and with all edges on the site $i$ removed from the graph:

$$\mu^{(i)}(\boldsymbol{s}) = \frac{1}{Z^{(i)}} \prod_{(k,l)\in\mathcal{E}, i\notin\{k,l\}} f_{kl}(s_k, s_l), \tag{4.42}$$

where $Z^{(ij)}$ and $Z^{(i)}$ are corresponding normalization constants. We also evaluate the marginal probabilities

$$\mu_i^{(ij)}(s_i) = \sum_{\boldsymbol{s}_{\mathcal{V}\setminus i}} \mu^{(ij)}(s_i, \boldsymbol{s}_{\mathcal{V}\setminus i}), \tag{4.43}$$

$$\mu_{ij}^{(ij)}(s_i, s_j) = \sum_{\boldsymbol{s}_{\mathcal{V}\setminus\{i,j\}}} \mu^{(ij)}(s_i, s_j, \boldsymbol{s}_{\mathcal{V}\setminus\{i,j\}}), \tag{4.44}$$

$$\mu_{\partial i}^{(i)}(\boldsymbol{s}_{\partial i}) = \sum_{\boldsymbol{s}_{\mathcal{V}\setminus\partial i}} \mu^{(i)}(\boldsymbol{s}_{\partial i}, \boldsymbol{s}_{\mathcal{V}\setminus\partial i}), \tag{4.45}$$

where $\partial i = \{j \mid (i,j) \in \mathcal{E}\}$ denotes the neighbors of the site $i$. The univariate marginal probability with an edge removed is denoted by the "message":

$$\mu_{i\to j}(s_i) = \mu_i^{(ij)}(s_i). \tag{4.46}$$

After removing the edge $(i, j)$, there is no interaction between $i$ and $j$ after marginalizing other

sites, so we have

$$\mu_{ij}^{(ij)}(s_i, s_j) = \mu_{i\to j}(s_i)\mu_{j\to i}(s_j). \tag{4.47}$$

Then we write down the relation between $\mu^{(ij)}(\boldsymbol{s})$ and $\mu^{(i)}(\boldsymbol{s})$. Temporarily ignoring the normalization constants, we can marginalize $\mu^{(ij)}(\boldsymbol{s})$ by first marginalizing $\mu^{(i)}(\boldsymbol{s})$ and then evaluating the remaining interactions between $i$ and $\partial i \setminus j$:

$$\mu_{ij}^{(ij)}(s_i, s_j) \propto \sum_{\boldsymbol{s}_{\partial i\setminus j}} \mu_{\partial i}^{(i)}(s_j, \boldsymbol{s}_{\partial i\setminus j}) \prod_{k\in\partial i\setminus j} f_{ik}(s_i, s_k). \tag{4.48}$$

It is proposed to approximate the environment to the site $i$ by the product of messages:

$$\mu_{\partial i}^{(i)}(\boldsymbol{s}_{\partial i}) \approx \prod_{j\in\partial i} \mu_{j\to i}(s_j). \tag{4.49}$$

Now we have the self-consistent equations

$$\mu_{i\to j}(s_i) \propto \sum_{\boldsymbol{s}_{\partial i\setminus j}} \prod_{k\in\partial i\setminus j} \mu_{k\to i}(s_k) \prod_{k\in\partial i\setminus j} f_{ik}(s_i, s_k) \tag{4.50}$$

$$= \prod_{k\in\partial i\setminus j} \sum_{s_k} f_{ik}(s_i, s_k)\mu_{k\to i}(s_k), \tag{4.51}$$

from which we can solve the $4Nd$ variables $\mu_{i\to j}(s_i)$. After recovering the normalization constants in Eq. (4.51), they can be solved by iterative algorithms if the convergence conditions are satisfied [170], or by gradient-based algorithms.

Given the messages, we can obtain the two-body marginals of the original distribution

$$\mu_{ij}(s_i, s_j) = \frac{f_{ij}(s_i, s_j)\mu_{i\to j}(s_i)\mu_{j\to i}(s_j)}{\sum_{s_i', s_j'} f_{ij}(s_i', s_j')\mu_{i\to j}(s_i')\mu_{j\to i}(s_j')}, \tag{4.52}$$

and estimate the energy and other observables with at most two-body interactions. For the Ising model in Eq. (2.1), we have

$$E_{\text{Bethe}} = J \sum_{\langle i,j\rangle} \mu_{ij}(s_i, s_j) s_i s_j. \tag{4.53}$$

We can also construct a Bethe entropy from the edge terms and the site terms:

$$S_{\text{Bethe}} = S_{\mathcal{E}} + S_{\mathcal{V}}, \tag{4.54}$$

$$S_{\mathcal{E}} = -\sum_{(i,j)\in\mathcal{E}} \ln \sum_{s_i, s_j} f_{ij}(s_i, s_j)\mu_{i\to j}(s_i)\mu_{j\to i}(s_j), \tag{4.55}$$

$$S_{\mathcal{V}} = \sum_i \ln \sum_{s_i} \prod_{j\in\partial i} \sum_{s_j} f_{ij}(s_i, s_j)\mu_{j\to i}(s_j), \tag{4.56}$$

and the Bethe free energy $F_{\text{Bethe}} = E_{\text{Bethe}} - \frac{1}{\beta} S_{\text{Bethe}}$.

When the graph is a tree, including a 1D chain, there actually exists a probability distribution $p_{\text{Bethe}}(\boldsymbol{s})$ that produces the marginals $\mu_{ij}(s_i, s_j)$ and the energy $E_{\text{Bethe}}$, and its entropy

is exactly $S_{\text{Bethe}}$, so the Bethe ansatz can serve as a variational ansatz. However, for general graphs, $S_{\text{Bethe}}$ no longer equals the entropy of $p_{\text{Bethe}}(\boldsymbol{s})$, so $F_{\text{Bethe}}$ is not guaranteed to be an upper bound of the true free energy. Nevertheless, it is a close approximation in many cases. Similar techniques have been generalized to systems beyond two-body interactions, known as cluster variation methods [171].

## 4.6 Neural network ansatzes

### 4.6.1 Layers in neural networks

As introduced in Section 1.3, neural networks in modern machine learning are composed of a common set of simple functions, known as "layers", and the composition can be arbitrarily sophisticated. The basic architecture among them is the feedforward neural network (FFNN), which is defined by a sequential composition of alternating linear layers and non-linear activation layers:

$$\boldsymbol{y} = (\text{Lin}_1 \circ \text{Act}_1 \circ \text{Lin}_2 \circ \text{Act}_2 \circ \cdots \circ \text{Lin}_{n_\text{l}} \circ \text{Act}_{n_\text{l}})(\boldsymbol{x}), \tag{4.57}$$

where $\boldsymbol{x}$ is the inputs to the network, $\boldsymbol{y}$ is the outputs, $n_\text{l}$ is the number of layers, and $\circ$ denotes function composition. This architecture is sketched in Fig. 4.2.

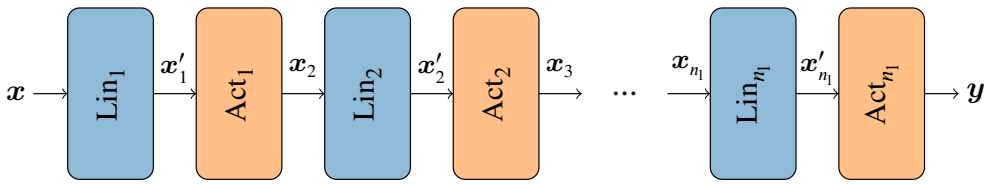



Figure 4.2: Architecture of FFNN, which is a sequential composition of alternating linear layers $\text{Lin}_i$ and non-linear activation layers $\text{Act}_i$. Each layer contains its own parameters. The values $\boldsymbol{x}_i$ and $\boldsymbol{x}'_i$ are outputs from the previous layer, which become inputs to the next layer.

The linear[5] layer is defined by

$$\text{Lin}_i(\boldsymbol{x}_i) = \boldsymbol{W}_i \boldsymbol{x}_i + \boldsymbol{b}_i, \tag{4.58}$$

where $\boldsymbol{W}_i$ is called the weight and $\boldsymbol{b}_i$ the bias, which are the parameters of this layer. At this point, the parameters are not shared between layers. Assuming $\boldsymbol{x}_i$ is a vector of size $n_{\text{in}\,i}$, we can choose an output size $n_{\text{out}\,i}$, then $\boldsymbol{W}_i$ is inferred to be a matrix of size $n_{\text{out}\,i} \times n_{\text{in}\,i}$, $\boldsymbol{b}_i$ is a vector of size $n_{\text{out}\,i}$, and the input size of the next linear layer $n_{\text{in}\,i+1}$ equals $n_{\text{out}\,i}$. In practice, the inputs $\boldsymbol{x}_i$ have an additional batch dimension, which denotes the evaluation of the same network on different input samples, but it is omitted in our discussion for simplicity. The sizes of the intermediate outputs are known as the "hidden sizes", and the hyperparameters of the network include the number of layers and the hidden sizes. These hyperparameters, in contrast to the parameters $\boldsymbol{W}_i$ and $\boldsymbol{b}_i$, define the network architecture and mainly determine

[5] It is technically an affine transformation rather than a linear transformation of $\boldsymbol{x}_i$, because $\boldsymbol{b}_i$ is included, but the name "linear layer" is commonly used in neural networks.

its expressiveness.

The activation layer, also known as the activation function, is an element-wise function of the input vector. A great variety of activation functions have been proposed, each with its own claimed advantages, but most are proven in practice to have only a minuscule impact on the expressiveness of the whole network, apart from the major difference in the output domain [172]. The simplest choice is the rectified linear unit (ReLU):

$$\mathrm{ReLU}(x) = \begin{cases} x, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{4.59}$$

It has been used since the early research on neural networks, and is still efficient today despite its simplicity.

A modern choice is the Gaussian error linear unit (GELU) [173]:

$$\mathrm{GELU}(x) = x\,\Phi(x), \tag{4.60}$$

where $\Phi(x)$ is the cumulative distribution function (CDF) of the unit Gaussian distribution. While having the same asymptotic behaviors at $x \to \pm\infty$ as ReLU, it is derived from the stochastic regularization of the network and helps to stabilize the evaluation and the training of the network. It is the default choice in the NetKet software package [2, 174], and is used in most of the results reported in this thesis.

In particular, the activation function in the last layer can be the identity function if the outputs are unconstrained in $\mathbb{R}$, or specifically constructed to constrain the outputs in an interval. In the latter case, a common choice is the sigmoid function:

$$\mathrm{Sigmoid}(x) = \frac{1}{1 + \mathrm{e}^{-x}}, \tag{4.61}$$

which maps from $\mathbb{R}$ to $(0, 1)$, or to another interval with an additional linear transform. The shapes of ReLU, GELU, and sigmoid are shown in Fig. 4.3.

### 4.6.2 Symmetries in neural networks: convolutional and pooling layers

Many-body systems usually have symmetries and locality, which also arise in machine learning of natural images. To take these advantages, the linear layers can be specialized into convolutional layers. Ignoring the layer index, a convolutional layer is denoted by $\boldsymbol{y} = \mathrm{Conv}(\boldsymbol{x})$, and in the 1D case it is defined by

$$y_{i,c} = \left( \sum_{k=1}^{n_\mathrm{k}} \sum_{c'=1}^{n_{\mathrm{c\,in}}} W_{k,c,c'} x_{i+k-1,c'} \right) + b_{i,c}, \tag{4.62}$$

where $\boldsymbol{x}$ is an array of size $n_\mathrm{s} \times n_{\mathrm{c\,in}}$, $\boldsymbol{y}$ is an array of size $n_\mathrm{s} \times n_{\mathrm{c\,out}}$, and we assume that the spatial dimension $n_\mathrm{s}$ is the same in the inputs and the outputs for simplicity. Both the inputs and the outputs have another dimension of "channels", also known as "features", which determines the size of the network. The convolutional layer acts as if it were a linear layer

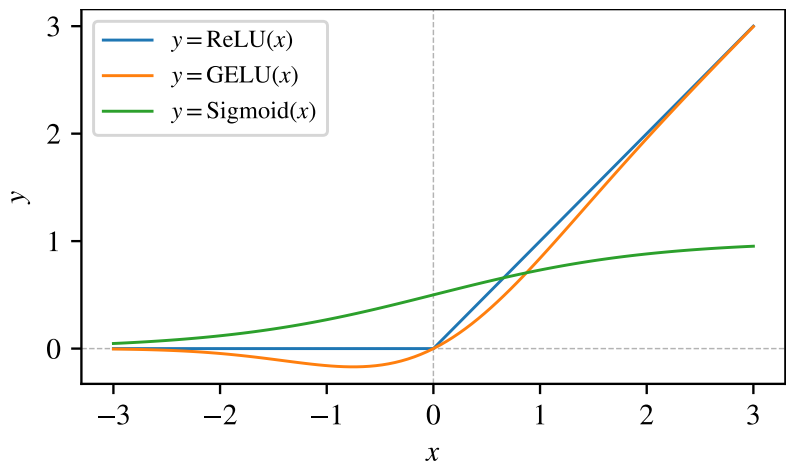



Figure 4.3: Commonly used activation functions ReLU, GELU, and sigmoid in neural networks. Both ReLU and GELU have asymptotic behaviors $y \to 0$ when $x \to -\infty$ and $y \to x$ when $x \to +\infty$, while GELU has more subtle non-linearity around $x = 0$. In contrast, the output of sigmoid is bounded in $(0, 1)$.

on the channel dimension, mapping from $n_{\mathrm{c\,in}}$ channels to $n_{\mathrm{c\,out}}$ channels. It performs a convolution only on the spatial dimension, rather than on the channel dimension. The weights $\boldsymbol{W}$ are an array of size $n_{\mathrm{k}} \times n_{\mathrm{c\,out}} \times n_{\mathrm{c\,in}}$, where $n_{\mathrm{k}}$ is the convolution kernel size, and the biases $\boldsymbol{b}$ are an array of size $n_{\mathrm{k}} \times n_{\mathrm{c\,out}}$. Higher-dimensional convolutional layers can be defined similarly. In contrast to the convolutional layer, the general linear layer is also known as the dense layer. The FFNN with convolutional layers is known as the convolutional neural network (CNN).

In Eq. (4.62), care should be taken when accessing the out-of-boundary values $x_{i,c}$ where $i > n_{\mathrm{s}}$. For physical systems with open boundary conditions, the common practice is to pad with zeros: $x_{i,c} = 0$ [175]; while for periodic boundary conditions, the convolution can be correspondingly periodic: $x_{i,c} = x_{i-n_{\mathrm{s}},c}$. In the latter case, the convolutional layer is translational equivariant in the spatial dimension, which means a translation of the inputs leads to the same translation of the outputs:

$$f(T_d \boldsymbol{x}) = T_d f(\boldsymbol{x}), \tag{4.63}$$

where $T_d$ denotes the translation by $d$ steps: $T_d(x_1, \ldots, x_{n_{\mathrm{s}}}) = (x_{1+d}, x_{2+d}, \ldots, x_{n_{\mathrm{s}}}, x_1, x_2, \ldots, x_d)$, and $f$ is a neural network composed of convolutional and activation layers. Beyond translational symmetry, this equivariance can be generalized to more symmetry groups, such as rotations and reflections [176, 177].

There is also a kind of special linear layers known as pooling layers[6], which are invariant under the symmetry operations:

$$f(T_d \boldsymbol{x}) = f(\boldsymbol{x}), \tag{4.64}$$

where $f$ is a neural network composed of convolutional, activation, and pooling layers. For the purpose of this thesis, we may use the sum pooling as the last linear layer, which sums

[6] Pooling layers can also be non-linear, such as the max pooling, which is not used in this thesis.

over the vector input to obtain a scalar output:

$$\mathrm{Pool}(\boldsymbol{x}) = \sum_{i=1}^{n_\mathrm{s}} x_i. \tag{4.65}$$

### 4.6.3 Neural network as variational ansatz

After introducing the various layers as building blocks, we proceed to apply a neural network as the ansatz for a many-body system of $N$ spins. One may naively define

$$q_\theta(\boldsymbol{s}) = \frac{1}{Z}\mathrm{NN}(\boldsymbol{s}), \tag{4.66}$$

$$Z = \sum_{\boldsymbol{s}} \mathrm{NN}(\boldsymbol{s}), \tag{4.67}$$

where NN is a neural network with input size $N$ and a scalar output, and $\theta$ denotes all parameters in it. However, the normalization constant $Z$ is an intractable sum of exponentially many configurations, and this distribution is not easier to sample than the Boltzmann distribution; therefore, it cannot be directly used for MCMC importance sampling in Eq. (3.31), or for variational inference in Eq. (4.4).

Some early attempts have been proposed to choose a simple ansatz that allows efficient cluster updates, such as a two-body interacting Hamiltonian [178] or a restricted Boltzmann machine (RBM) [179], and use it in Eq. (3.31). As the normalization constant of the ansatz is still intractable, and thus the variational inference is inapplicable, the parameters of the ansatz are trained in a supervised learning approach. One first generates some samples from the target distribution using traditional MCMC, and then optimizes the energies of these samples evaluated by the ansatz towards those evaluated by the original Hamiltonian. After training, more samples can be generated from the ansatz using efficient cluster updates. However, the quality of the samples used in the training limits the accuracy of this approach.

Another approach is to specifically construct a neural network with tractable normalized probability. A prominent family of such an approach is the autoregressive models, which we will discuss in Chapter 5. Moreover, a notable family of neural networks with this property is the normalizing flows [180, 181], which reparameterize the target distribution into a simple one using the same principle as Eq. (3.4). They are mostly used for continuous variables, which are beyond the scope of this thesis.

### 4.6.4 Machine learning models other than neural networks

Apart from neural networks, tensor networks [182] with chain or tree geometries also support tractable normalized probability, which we will discuss in Chapter 8. However, they are more naturally defined and frequently used as variational ansatzes in quantum systems, while in classical systems they are usually used to approximate the partition function as a generalization of the transfer matrix to higher dimensions [54, 183–185].

A lesser known family of machine learning models that share this property are the sum-product networks [186], which define complicated multivariate distributions using directed

graphs. They have been applied to various fields of machine learning [187], and they receive renewed attention with the recently proposed Kolmogorov–Arnold networks (KANs) [188], but their usage is still not as prevalent as neural networks due to limited support in hardware and software [189].

# 5 Autoregressive neural networks (ARNNs)

In the previous chapter, we have seen the power of variational inference to approximate the free energy of intractable many-body systems. A crucial requirement for this method is a variational ansatz that supports efficient evaluation of normalized probability and preferably has the rich expressiveness of neural networks. The search for this kind of variational ansatz has led to the development of autoregressive (AR) models, which have yielded the most prominent achievements in modern machine learning, and also found broad applications in many-body physics.

## 5.1 Autoregressive factorization of joint probability

We start from noticing that any joint probability $q(\boldsymbol{s})$ can be factorized into a product of conditional probabilities:

$$q(\boldsymbol{s}) = \prod_i q_i(s_i \mid \boldsymbol{s}_{<i}) \tag{5.1}$$

$$= q_1(s_1) q_2(s_2 \mid s_1) q_3(s_3 \mid s_1, s_2) \cdots q_N(s_N \mid s_1, s_2, \ldots, s_{N-1}), \tag{5.2}$$

where we still assume that $\boldsymbol{s}$ is a vector of $N$ spins, each can take the values $\pm 1$, and $\boldsymbol{s}_{<i} = (s_1, s_2, \ldots, s_{i-1})$ denotes all variables before the $i$-th. This relation between random variables is called an AR model, which is a generalization from the traditional concept of AR in time series, where each random variable only has a linear dependency on the previous ones. In our case, the dependency can be arbitrarily non-linear.

In principle, if the joint probability is known, one can compute the conditional probabilities by definition:

$$q_i(s_i \mid \boldsymbol{s}_{<i}) = \frac{\sum_{\boldsymbol{s}_{>i}} q(\boldsymbol{s})}{\sum_{\boldsymbol{s}_{\geq i}} q(\boldsymbol{s})}, \tag{5.3}$$

which involves summations over exponentially many terms. However, neither do we know the joint probability in practice nor can we exactly perform the summations. We generally parameterize the conditional probabilities by arbitrarily expressive neural networks, therefore the name ARNN, and use variational inference to optimize them towards the target distribution. The requirement of normalization in variational inference is particularly easy to fulfill under this factorization because as long as each conditional probability is normalized: $\sum_{s_i} q_i(s_i \mid$

$\boldsymbol{s}_{<i}) = 1$, then the joint probability in Eq. (5.1) will automatically be normalized. For binary variables, each conditional probability is naturally modeled by a Bernoulli distribution $\mathcal{B}(s_i \mid \hat{s}_i) = (1-\hat{s}_i)\delta_{s_i,-1} + \hat{s}_i\delta_{s_i,+1}$, which is always normalized and easy to sample, and the parameter $\hat{s}_i$ can have an arbitrarily sophisticated dependency on the previous variables $\boldsymbol{s}_{<i}$.

The joint probability of an AR model supports exact sampling, because we can sample the variables sequentially, following their order defined by the AR model. We first generate the value of $s_1$ from the independent distribution $q_1(s_1)$, then obtain the distribution $q_2(s_2 \mid s_1)$ and generate the value of $s_2$, then obtain the distribution $q_3(s_3 \mid s_1, s_2)$ and so forth, until all variables are generated. This procedure is also known as ancestral sampling. It involves $N$ sequential evaluations of the neural network, which can be a bottleneck in the time complexity of variational inference. On the other hand, when evaluating $q(\boldsymbol{s})$ given all variables $\boldsymbol{s}$, all conditional probabilities can be computed in parallel, which is comparatively not a concern in time complexity.

ARNN is an excellent ansatz for variational inference in Eq. (4.4), because it supports exact sampling and efficient evaluation of normalized probability, while having the rich expressiveness of neural networks. In the following, we introduce some neural network architectures to parameterize the conditional probabilities, each with its own use cases.

## 5.2 Dense and convolutional ARNNs

The most general way to parameterize conditional probabilities is a dense feedforward neural network (FFNN) as in Eq. (4.57), which takes the spins $\boldsymbol{s}$ as inputs and outputs the parameters $\hat{\boldsymbol{s}}$, then the conditional probabilities are modeled by the Bernoulli distributions $\mathcal{B}(s_i \mid \hat{s}_i)$. The network needs to fulfill the AR property: each output $\hat{s}_i$ can only depend on the previous inputs $\boldsymbol{s}_{<i}$. For a single dense layer, this is achieved by applying a triangular mask on the weight matrix in Eq. (4.58):

$$\text{MaskedLin}(\boldsymbol{x}) = (\boldsymbol{M} \odot \boldsymbol{W})\boldsymbol{x} + \boldsymbol{b}, \tag{5.4}$$

$$M_{ij} = \begin{cases} 1, & i > j \\ 0, & i \leq j \end{cases}, \tag{5.5}$$

where $\odot$ denotes the element-wise multiplication. This masked linear layer is illustrated in Fig. 5.1 (a). Multiple linear layers can be stacked to improve the expressiveness of the network, as shown in Fig. 5.1 (b), where the whole network fulfills the AR property as long as only one layer masks out the "straight right" connections $M_{ii}$. This architecture is known as the masked autoencoder for distribution estimation (MADE) [190].

Convolutional layers can also be applied to utilize translational symmetry and locality of the physical system, with the same mask $\boldsymbol{M}$ to ensure AR property. However, unlike a single convolutional FFNN, the joint probability $q(\boldsymbol{s})$ of a convolutional ARNN is not translational invariant as in Eq. (4.64). The ARNN always assigns an artificial order of the spins to factorize the joint probability, thus breaking the translational symmetry. Instead, the purpose of convolutions in ARNN should be understood as enforcing the translational invariance of certain

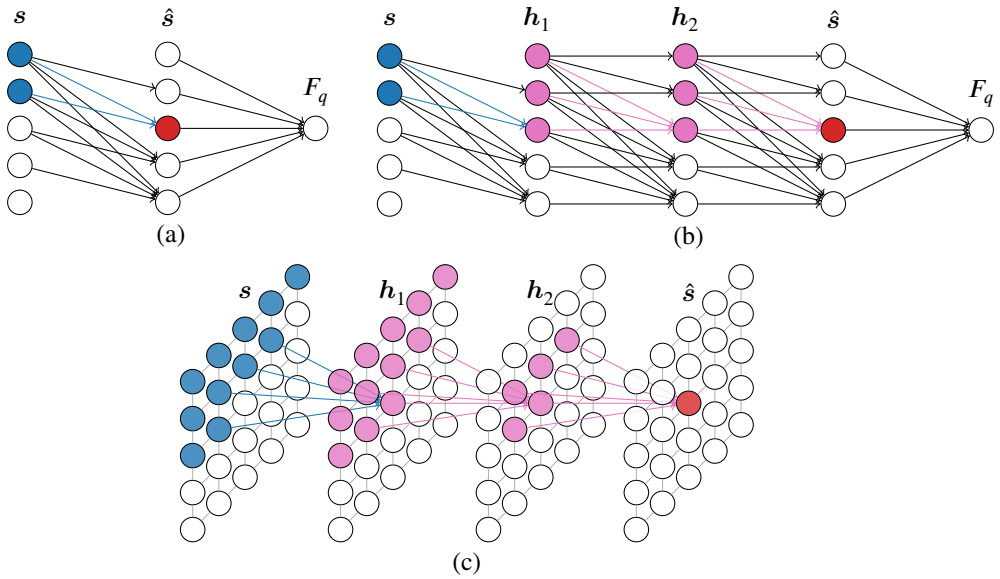



Figure 5.1: Architectures of dense and convolutional ARNNs for variational inference. The spins $\boldsymbol{s}$ are the inputs to the network. The values $\hat{\boldsymbol{s}}$ are the outputs from the network, which become parameters of the Bernoulli distributions. The values $\boldsymbol{h}_i$ are intermediate results in the network. The variational free energy $F_q$ is given by Eq. (4.4), which depends on both the spins $\boldsymbol{s}$ and the parameters $\hat{\boldsymbol{s}}$. The colored sites denote the receptive field of a site in $\hat{\boldsymbol{s}}$, i.e., the input and the intermediate sites that an output site depends on. (a) The network has only one layer, which is densely connected, and masked to ensure the AR property. The connections that are not masked out are illustrated as the arrows. (b) The network has three masked dense layers. (c) The network has three masked convolutional layers on a 2D lattice. In each layer, only the connections to one output site are shown for clarity, which depict the shape of the convolution kernel. This figure is reproduced from Fig. 1 in Ref. [35].

rules to determine the effect of the local environment on a spin.

An early application of 1D convolutional ARNN is the modeling of audios, known as WaveNet [191]. The 2D convolutional ARNN has been applied similarly to the modeling of images, known as PixelCNN [192], whose shape of the 2D convolutional kernel is illustrated in Fig. 5.1 (c). These ARNN architectures have also been introduced to the modeling of many-body systems, originally under the name of variational autoregressive networks (VANs) [35], and achieved superior results compared to traditional variational ansatzes.

In addition to dense and convolutional ones, another widely used family of ARNNs is the recurrent neural networks (RNNs), which incorporate the translational symmetry of the physical system in another way. They maintain the AR property by updating of some hidden variables $\boldsymbol{h}^{(i)}$, also known as "memories", at each AR step $i$:

$$\boldsymbol{h}^{(i)} = f\left(s_{i-1}, \boldsymbol{h}^{(i-1)}\right), \tag{5.6}$$

$$\hat{s}_i = g\left(\boldsymbol{h}^{(i)}\right), \tag{5.7}$$

where $f$ and $g$ can be arbitrarily sophisticated neural networks, and their parameters are usually shared between sites. Due to the sharing of parameters, RNNs can achieve satisfactory results with fewer parameters than dense networks. More variants of RNNs, including hybrids with CNNs and tensor networks, are also used in practice [192, 193]. In particular, the popular large language models in modern machine learning have essentially inherited the principles of RNN [26]. It is also worth mentioning that ARNNs with hierarchical architectures [194, 195] and representations in frequency or wavelet spaces [196, 197] have been proposed to enhance translational symmetry and scalability.

### 5.2.1 Numerical results

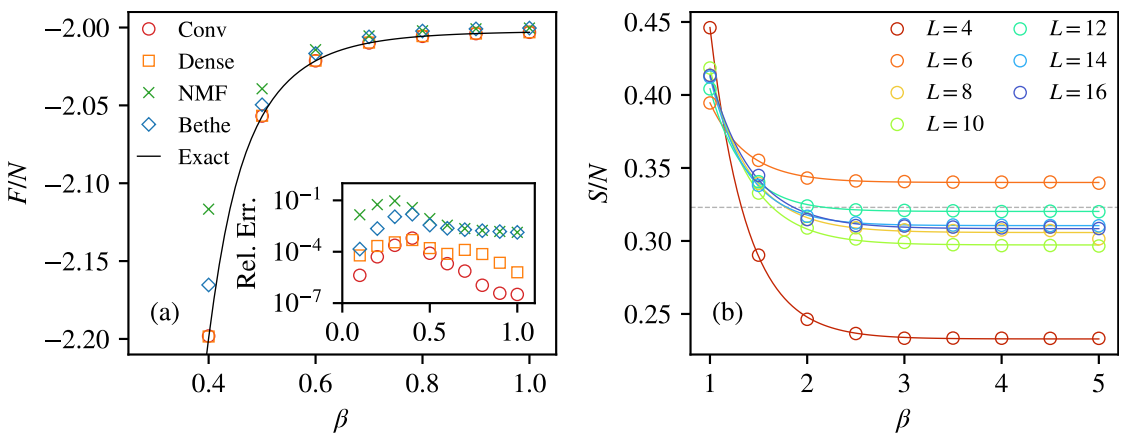



Figure 5.2: (a) Free energy per site $F/N$ of the ferromagnetic (FM) Ising model on the $16 \times 16$ square lattice with periodic boundary conditions (PBC), at varying inverse temperature $\beta$, given by a dense ARNN and a convolutional ARNN, compared to the exact result [39] and the traditional methods of naive mean-field (NMF) and Bethe ansatzes. The inset shows the relative error of the numerical results compared to the exact result.
(b) Entropy per site $S/N$ of the frustrated antiferromagnetic (AFM) Ising model on triangular lattices of different sizes $N = L \times L$ with PBC, given by a convolutional ARNN. The curves are least-squares fittings of $S/N = ae^{-b\beta} + c$, where $a, b, c$ are parameters for fitting, and $c$ is the residual entropy when $\beta \to \infty$. The horizontal dashed line indicates the exact result $\lim_{N\to\infty} \lim_{\beta\to\infty} S/N \approx 0.323$ [55–57]. Note that the curves for $L = 8, 14, 16$ are almost overlapped. This figure is reproduced from Fig. 2 in Ref. [35].

#### Square Ising model

We first demonstrate the performance of ARNN on the well-studied ferromagnetic (FM) Ising model on the $10 \times 10$ square lattice with periodic boundary conditions (PBC), where the analytical result [39] and the traditional methods of naive mean-field (NMF) ansatz in Section 4.5.1 and Bethe ansatz in Section 4.5.2 are available for comparison. A dense ARNN and a convolutional ARNN are trained on the Boltzmann distribution of this Hamiltonian, whose hyperparameters, such as the number of layers, the number of channels, and the

convolution kernel size, are selected to produce the lowest variational free energies within a reasonable computation budget. The networks are trained from scratch at each inverse temperature $\beta = 0.1, 0.2, \ldots, 1$. Figure 5.2 (a) shows the results of free energy, where we can clearly see that the ARNNs produce superior results compared to the traditional ansatzes, whose relative errors are lower by orders of magnitude.

In particular, the results from the convolutional ARNN are generally more accurate than those from the dense one with a similar number of parameters, thanks to the utilization of the translational symmetry and the locality of the physical system. This comparison justifies the use of convolutional layers in ARNN, even though the overall joint probability is not translational invariant. However, this advantage diminishes around the critical point $\beta_c = \frac{1}{2}\ln\left(1+\sqrt{2}\right) \approx 0.44$, where the correlation length of the physical system diverges until it is bounded by the system size, as discussed in Section 3.4. The convolutional neural network can capture these long-range correlations only if the size of its receptive field is at least comparable to the correlation length.

### Triangular Ising model

Then we apply the convolutional ARNN to the more complicated problem of antiferromagnetic (AFM) Ising model on the triangular lattice with PBC, which is geometrically frustrated. As discussed in Section 2.1.3, its frustration leads to exponentially many degenerate ground states and a non-zero residual entropy, which poses challenge for both MCMC and traditional variational ansatzes.

In Fig. 5.2 (b), the ARNN produces entropies that exponentially decay to a constant as $\beta \to \infty$ for each system size $L$, and the results with large $L$ are close to the analytical result with $L \to \infty$. These results demonstrate the rich expressiveness of ARNN, which has the potential to accurately approximate exponentially complex target distributions using polynomial computation time and number of parameters.

### Hopfield model

An experiment to qualitatively study the ability of ARNN to capture multiple ground states is conducted on the Hopfield model. As discussed in Section 2.1.5, it is a model of associative memory where one can store and retrieve multiple patterns. We construct a Hopfield model with $N = 100$ spins and $n_{\text{pat}} = 2$ random orthogonal patterns, and train a dense ARNN that consists of only a single layer. Then we generate $10^4$ samples from the ARNN, and compute their probabilities $q(\boldsymbol{s})$ given by the ARNN, as well as their overlaps $O_1(\boldsymbol{s}) = \langle \boldsymbol{\xi}^{(1)} | \boldsymbol{s} \rangle$, $O_2(\boldsymbol{s}) = \langle \boldsymbol{\xi}^{(2)} | \boldsymbol{s} \rangle$ with the two patterns, respectively.

Figure 5.3 shows that the ARNN successfully samples the memorized patterns with exponentially higher probabilities than other states, in the retrieval phase with high $\beta$. This indicates the capability of ARNN to avoid mode collapse and capture all the modes separated by exponentially high energy barriers in this Hamiltonian.

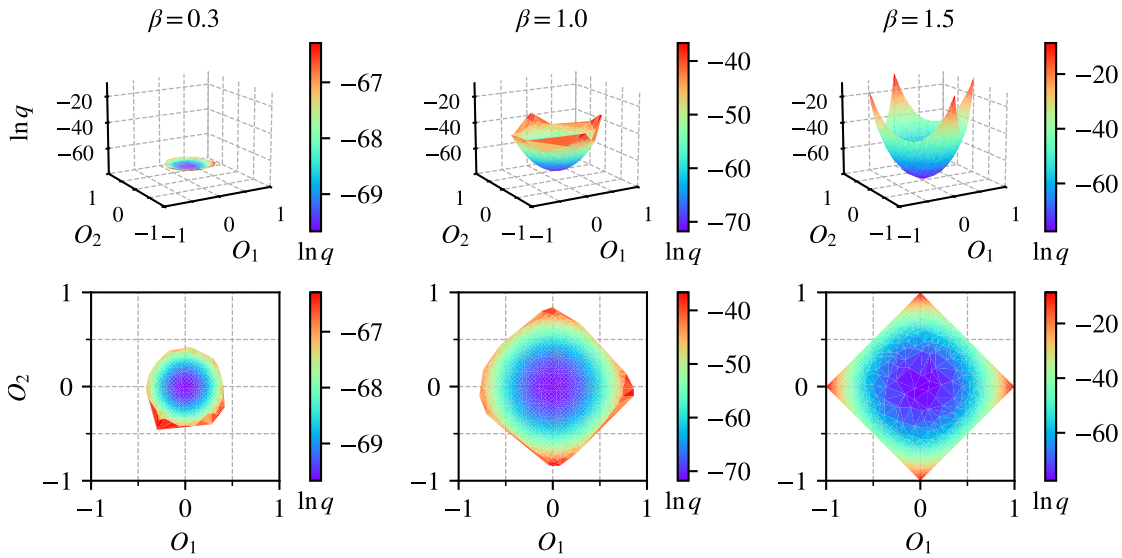



Figure 5.3: Log-probability $\ln q$ of samples from a dense ARNN, for the Hopfield model with $N = 100$ spins and $n_{\text{pat}} = 2$ orthogonal patterns. The top row shows the 3D view of the log-probability surfaces, and the bottom row shows the same results in the 2D view from top. The columns are results with different inverse temperatures $\beta = 0.3, 1.0, 1.5$. The $O_1$ and $O_2$ axes denote the overlaps between each sample and the two patterns respectively, as defined in Eq. (2.17). The four ground states have $(O_1, O_2) = (+1, 0)$, $(-1, 0)$, $(0, +1)$, $(0, -1)$ respectively. This figure is reproduced from Fig. 3 in Ref. [35].

### Sherrington–Kirkpatrick (SK) model

We also have the experiment on the Sherrington–Kirkpatrick (SK) model in Section 2.1.4, which has frustrated interactions with even higher complexity than the previous models. As a proof of concept, we generate a random instance of the SK model with $N = 20$ spins and compute its properties by exact enumeration. Then we train a dense ARNN with a single layer. Figure 5.4 (a) shows that the ARNN still produces a substantially more accurate approximation than traditional ansatzes for this Hamiltonian, especially in the glassy phase with high $\beta$ where the iterative solution of the Bethe ansatz fails to converge.

### Inverse SK problem

In addition to the ordinary problem of estimating the observables given the Hamiltonian, ARNNs can also be used in the inverse SK problem, where we estimate the unknown interactions $\{J^*_{ij}\}$ given the correlations $\{C^*_{ij}\}$. Because each estimated correlation $\mathbb{E}_{\text{MC}}[C_{ij}]$ as in Eq. (3.1) is a differentiable function of $\{J_{ij}\}$, we can directly use stochastic gradient descent to minimize the least-square loss

$$L(\{J_{ij}\}) = \frac{1}{N^2} \sum_{ij} \left( \mathbb{E}_{\text{MC}}[C_{ij}] - C^*_{ij} \right)^2 . \tag{5.8}$$

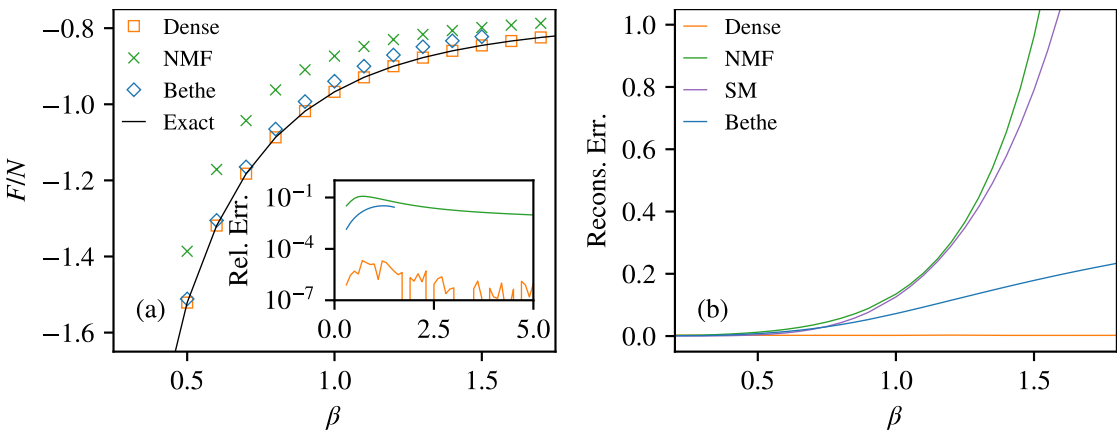



Figure 5.4: (a) Free energy per site $F/N$ for a random instance of the SK model with $N = 20$ spins, at varying inverse temperature $\beta$, given by a dense ARNN, compared to the exact result by enumeration, the NMF ansatz, and the Bethe ansatz. The inset shows the relative error of the numerical results compared to the exact result. The iterative solution of the Bethe ansatz converges only when $\beta \leq 1.5$.
(b) Reconstruction error of the inverse SK problem with $N = 20$ spins, given by a dense ARNN, compared to the NMF ansatz [198], the Sessak–Monasson (SM) expansion [199], and the Bethe ansatz [200]. This figure is reproduced from Fig. 4 in Ref. [35].

In this problem, we use an ARNN with two dense layers to estimate $C_{ij}$. The abilities of different methods to estimate the interactions is assessed by the reconstruction error

$$\text{Recons. Err.} = \frac{1}{N^2} \sum_{ij} \left( J_{ij} - J^*_{ij} \right)^2, \tag{5.9}$$

where $J_{ij}$ are the estimated values and $J^*_{ij}$ are the true values. The results are shown in Fig. 5.4 (b), where the ARNN again outperforms traditional methods by a large margin.

### 5.2.2 Remarks on network sizes

It is worth discussing the sizes of the ARNNs used in the above experiments. For fully connected cases of the Hopfield model, the SK model, and the inverse SK problem, we use small ARNNs containing only $O(N^2)$ parameters, which take minutes to train on a high-end GPU as of this writing, and they produce substantially better results than the Bethe ansatzes with the same order of parameters. However, for the Ising models on the $16 \times 16$ square and triangular lattices, the ARNNs with the optimal variational free energies we have found contain as many as $O(10^6)$ parameters, while the Bethe ansatzes contain only $O(10^3)$. The large number of parameters also leads to substantially more computation time to evaluate the observables, since the training and the sampling of such a large neural network typically take hours on a high-end GPU, while the NMF ansatz and the Bethe ansatz can be solved almost instantly. Therefore, many use cases of neural networks to replace traditional ansatzes appear only in the regime where we seek for much higher accuracy at the cost of correspondingly more

computation. The comparison of neural networks and traditional methods under the same computation budget will be discussed in Section 5.4.

Apart from training a new network with each system size and temperature, it has also been proposed to directly reuse or fine-tune a trained network with different system sizes and temperatures [201–203]. This line of research opens a promising way to numerically compute the renormalization group (RG) flow, and extrapolate the critical temperature to the thermodynamic limit [204].

## 5.3 Sparse two-body ARNN (TwoBo)

In the previous section, we have seen that the simple ARNN with dense architecture approximates many-body systems to much higher accuracy than traditional ansatzes, but also introduces much more trainable parameters and requires longer computation time. The convolutional ARNN reaches even higher accuracy with approximately the same number of parameters, as it utilizes the translational symmetry and the locality of the physical system. In this section, we continue to search for a more sophisticated ARNN architecture that further embodies knowledge of the physical system, particularly the sparsity, to reduce the number of trainable parameters and the computation time without losing the expressiveness.

An approach has been proposed in Ref. [205] to convert any many-body system on a sparse graph into an equivalent system on a smaller dense graph using the feedback vertex set. Meanwhile, it has been noticed in Ref. [206] that the dense ARNN architecture involves a lot of redundant computation that can be eliminated given the physical properties of the Boltzmann distribution and the locally interacting Hamiltonian. In the following, we present a different approach proposed in Ref. [4], which utilizes these physical properties to improve the efficiency of ARNN.

### 5.3.1 TwoBo architecture from Boltzmann distribution

We start from directly factorizing the many-body Boltzmann distribution $p_{\mathrm{B}}(\boldsymbol{s})$ into AR conditional distributions in Eq. (5.1):

$$p_{\mathrm{B}i}(s_i \mid \boldsymbol{s}_{<i}) = \frac{\sum_{\boldsymbol{s}_{>i}} p_{\mathrm{B}}(\boldsymbol{s})}{\sum_{\boldsymbol{s}_{\geq i}} p_{\mathrm{B}}(\boldsymbol{s})} = \frac{\sum_{\boldsymbol{s}_{>i}} \mathrm{e}^{-\beta H(\boldsymbol{s})}}{\sum_{\boldsymbol{s}_{\geq i}} \mathrm{e}^{-\beta H(\boldsymbol{s})}}, \tag{5.10}$$

and simplifying the result. When the Hamiltonian has only two-body interactions, such as the Ising model in Eq. (2.13), it has been derived in Ref. [207] that the conditional probability takes the form:

$$p_{\mathrm{B}i}(s_i = +1 \mid \boldsymbol{s}_{<i}) = S\left(2\beta\xi_{ii} + \rho_{\mathrm{B}i}\left(\{\xi_{il} \mid l > i\}\right)\right), \tag{5.11}$$

$$\xi_{il} = \sum_{k<i} J_{kl} s_k, \tag{5.12}$$

in the notations of this thesis, and omitting the external fields for simplicity. Here, $S$ is the sigmoid function in Eq. (4.61), and $\rho_{\mathrm{B}i}$ is a function depending on $\boldsymbol{J}$ and $\beta$. The function $\rho_{\mathrm{B}i}$

is different for each index $i$, and it only takes the intermediate variables $\xi_{il}$ with $l > i$ as the inputs. Although the exact evaluation of $\rho_{\mathrm{B}i}$ can take exponential time, it has been shown in Ref. [207] that $\rho_{\mathrm{B}i}$ can be simplified for specific physical systems, such as an analytical expression for the Curie–Weiss model and a replica symmetric breaking (RSB) approximation for the SK model.

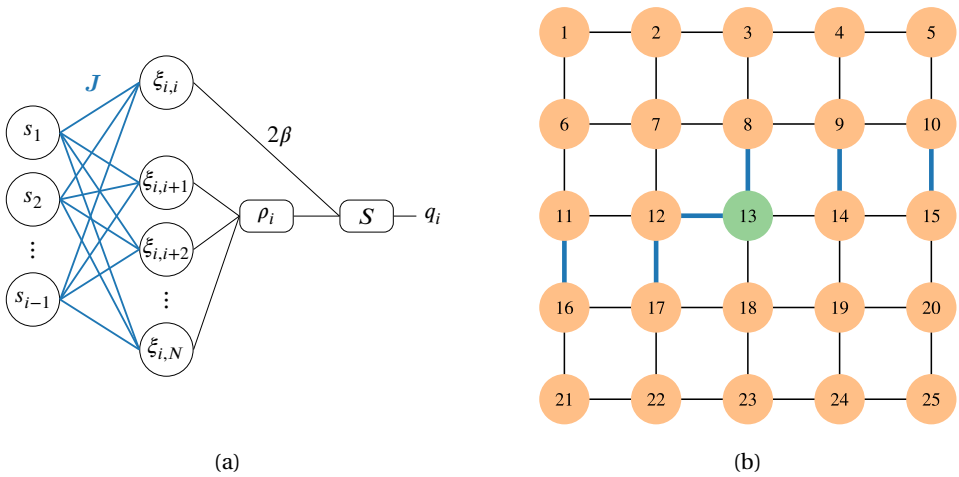



Figure 5.5: (a) Sketch of the TwoBo ARNN architecture to compute a conditional probability $q_i(s_i = +1 \mid \boldsymbol{s}_{<i})$. The weights of the first layer are directly taken from the interaction matrix $\boldsymbol{J}$ and fixed. The function $\rho_i$ is parameterized by a dense layer, whose inputs are the variables $\{\xi_{il} \mid l > i\}$ after the first layer. The last activation function $S$ is the sigmoid function in Eq. (4.61), and there is a skip connection from $\xi_{ii}$ to $S$ with the fixed weight $2\beta$.
(b) Calculating the conditional probability $q_i(s_i \mid \boldsymbol{s}_{<i})$ with $i = 13$ on a 2D grid of $N = 25$ spins. The edges between the spins $\boldsymbol{s}_{<i}$ and $\boldsymbol{s}_{\geq i}$ are highlighted in blue, and only these edges are used when computing the intermediate variable $\xi_{il}$. Therefore, among the variables $\{\xi_{il} \mid l > i\}$, only those with $l < i + L$ are kept as inputs to $\rho_i$, and the others are ensured to be zero, which enhances the sparsity of the TwoBo network. This figure is reproduced from Fig. 1 in Ref. [4].

For the purpose of this thesis, we approximate $\rho_{\mathrm{B}i}$ using a neural network $\rho_i$ with trainable parameters, and we have surprisingly found that using a single dense layer is enough to obtain results with satisfactory accuracies in our numerical experiments. In addition, we interpret the linear transformation from the input spins $s_i$ to the intermediate variables $\xi_{il}$ in Eq. (5.12) as another linear layer without bias, whose weights are taken directly from $\boldsymbol{J}$ and are not trained. Therefore, Eq. (5.11) becomes an ARNN ansatz $q_i(s_i \mid \boldsymbol{s}_{<i})$ as shown in Fig. 5.5 (a). This architecture is colloquially named TwoBo because the authors were tired of the ever-growing lengths of acronyms.

### 5.3.2 Sparsity of intermediate variables

Incorporating the interaction matrix $\boldsymbol{J}$ into the parameters of TwoBo leads to the opportunity to reduce the computation time and the number of trainable parameters using the sparsity of

the physical system. In general, the first layer in Eq. (5.12) can be evaluated by a matrix-vector multiplication $\boldsymbol{Js}$ while storing cumulative sums, which takes $O(N^2)$ time, where $N$ is the number of spins. When the physical system is sparse and the number of non-zero entries in $\boldsymbol{J}$ only scales by $O(N)$, the evaluation takes only $O(N)$ time as well.

For each function $\rho_i$, there are $N-i$ input variables in general and a scalar output, which introduces $O(N)$ parameters and $O(N)$ computation time when parameterized by a single dense layer. Fortunately, when the physical system has a local geometry, many of the input variables are ensured to be zero. For example, on a 2D grid with $N = L^2$ spins as shown in Fig. 5.5 (b), each variable $\xi_{il}$ is non-zero only when $i < l < i + L$, therefore the number of input variables to $\rho_i$, the number of parameters in it, and the time to evaluate it are reduced from $O(L^2)$ to $O(L)$. Taking into account all the functions $\{\rho_i \mid 1 \le i \le N\}$, they take $O(LN) = O(L^3)$ time to evaluate in total, which dominates over the $O(N)$ time of the first layer. Those functions also have $O(L^3)$ parameters in total. Therefore, the time to evaluate the TwoBo ansatz and the number of trainable parameters in it scale both by $O(L^3)$, which is polynomially lower than $O(L^4)$ of the dense ARNN (MADE) in Section 5.2. Similarly, the time and the space are reduced from $O(L^6)$ to $O(L^5)$ on 3D grids.

### 5.3.3 Numerical results

The performance of TwoBo is demonstrated in numerical experiments using the Edwards–Anderson (EA) model with binary interactions on 2D and 3D grids with PBC, as well as random regular graphs (RRGs). As discussed in Section 2.1.4, the EA model is generally more difficult to solve than the simple Ising model in Section 5.2, and the 3D EA model is particularly challenging from the perspective of computational complexity. Moreover, the RRG demonstrates the performance of TwoBo on systems without regular geometry. We use RRGs with degree $d = 3$, and sample them from the uniform distribution of all RRGs with the given $N$ and $d$ using the Steger–Wormald algorithm [208]. Each reported result is averaged over 10 random instances of the Hamiltonian, and the error bar shows the standard error over the instances.

The results from TwoBo are compared with those from MADE. We use the MADE with only one layer, and it still has more trainable parameters than the TwoBo with the same system size. On the 2D grids, another comparison is made with the RNN architecture in Ref. [72], which has outperformed several dense and convolutional ARNN architectures in the regime of numerous parameters. In this thesis, we compare with it mainly in the regime of few parameters, using only four memory units and a comparable number of trainable parameters to TwoBo, while the comparison using more parameters will be shown in Fig. 5.10.

An annealing procedure is performed to compute the free energies at different inverse temperatures $\beta \in (0, 3]$, and to help the optimization converge to the desired global minimum at high $\beta$. We equally divide the range of $\beta$ into $N_{\text{anneal}} = 60$ annealing steps. At the first $\beta = 0.05$, we perform $N_{\text{warm}} = 500$ optimization steps as a warm-up. After that, in each annealing step, we perform $N_{\text{opt}} = 200$ optimization steps while keeping $\beta$ unchanged. In the next annealing step, we continue to optimize the ansatz with a different value of $\beta$. All the architectures of TwoBo, MADE, and RNN are trained using this procedure.

### Variational free energy and ground state energy

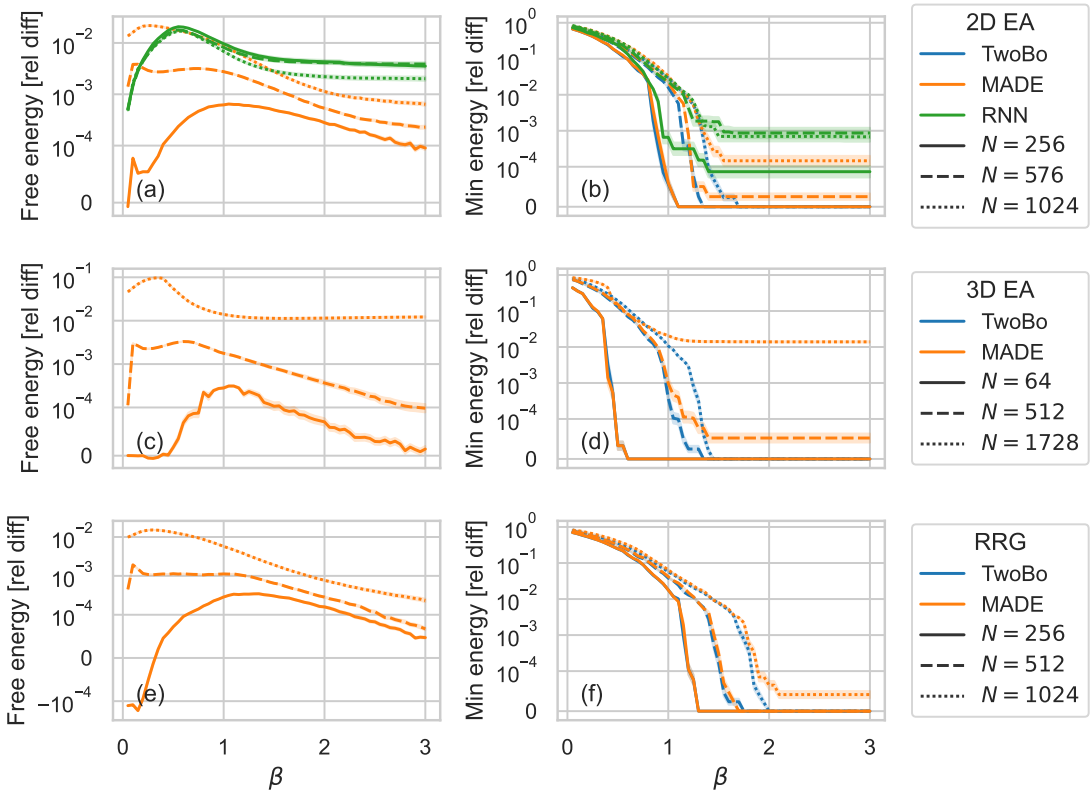



Figure 5.6: Variational free energy (left column) and minimum energy of sampled configurations (right column) at varying inverse temperature $\beta$, for the EA model on 2D grids (top row), 3D grids (middle row), and RRGs (bottom row) of different sizes. The performance of TwoBo, MADE, and RNN (only in 2D) architectures are compared. The variational free energy is shown as the relative difference from TwoBo at the same $\beta$, and the minimum energy is relative from the one found by TwoBo at $\beta = 3$. The $y$-axis uses the logarithmic scale when the absolute value is greater than $10^{-4}$. This figure is reproduced from Fig. 2 in Ref. [4].

As shown in the left column of Fig. 5.6, TwoBo gives a lower variational free energy than MADE and RNN with a comparable number of trainable parameters, on almost all tested geometries, system sizes, and temperatures. TwoBo is only outperformed by MADE in the simple cases of small system size and low $\beta$. This demonstrates the accuracy of TwoBo in approximating the Boltzmann distribution, even with disordered and frustrated Hamiltonians, and this advantage becomes more substantial at larger system sizes.

The right column of Fig. 5.6 shows the minimum energy found by the ansatzes in the annealing procedure, which is of more interest in combinatorial optimization problems. TwoBo is also able to find energies lower than those of MADE and RNN, especially at large system sizes. In addition to variational methods, we also use the exact solvers in McGroundstate [209] and the heuristic solvers in MQLib [210] to solve the same systems. Compared to these solvers,

TwoBo gives equal or lower energies with comparable computation time in all tested cases, which provides evidence that the sophisticated heuristics of neural networks outperform traditional methods in optimization problems with many variables and complicated interactions.

### Convergence of training

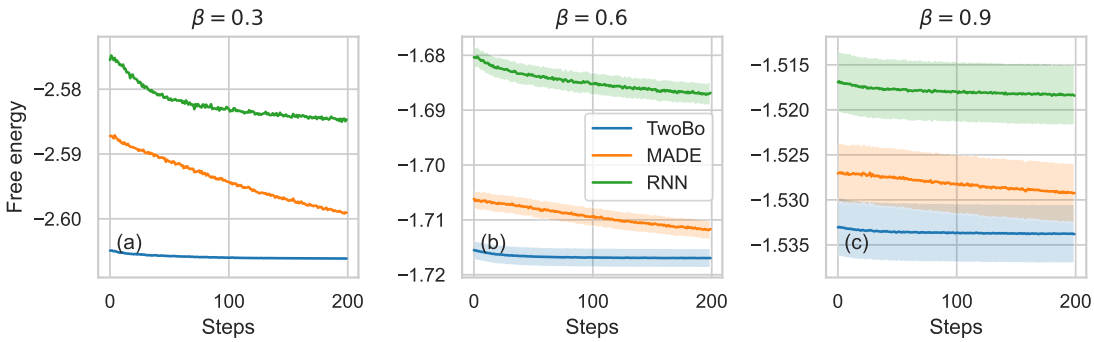



Figure 5.7: Convergence of variational free energy during the optimization steps in an annealing step, at different inverse temperatures $\beta$, on the 2D EA model with $N = 576$. This figure is reproduced from Fig. 3 in Ref. [4].

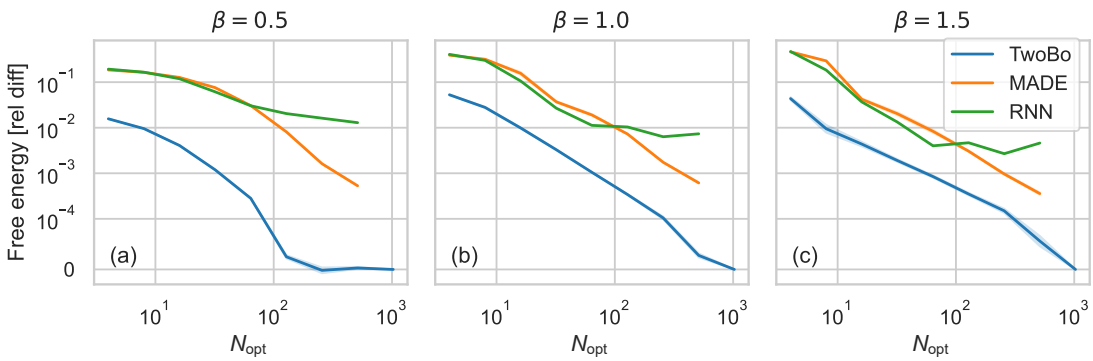



Figure 5.8: Variational free energy as a function of the annealing speed, at different inverse temperatures $\beta$, on the 2D EA model with $N = 576$. The annealing speed is controlled by the number of optimization steps $N_{\text{opt}} = 4, 8, 16, \ldots, 1024$ in each annealing step. The variational free energy is shown as the relative difference from TwoBo at the same $\beta$ with $N_{\text{opt}} = 1024$. The $y$-axis uses the logarithmic scale when the value is greater than $10^{-4}$. This figure is reproduced from Fig. 4 in Ref. [4].

In addition to accuracy after training, TwoBo also has the advantage of fast convergence during training. As shown in Fig. 5.7, in each annealing step with a fixed $\beta$, TwoBo always starts from a lower variational energy and reaches convergence in fewer optimization steps. This advantage can be attributed to the incorporation of the interactions $\boldsymbol{J}$ in the parameters, which makes the initialization of the ansatz $q(\boldsymbol{s})$ physically feasible, i.e., closer to the target distribution $p_{\text{B}}(\boldsymbol{s})$ than a neural network with only random initialization.

This fast convergence also allows us to reduce the total computation time of the annealing procedure using a higher annealing speed. The annealing speed is controlled by the number of optimization steps $N_{\mathrm{opt}}$ in each annealing step. As shown in Fig. 5.8, the variational free energy systematically decreases as $N_{\mathrm{opt}}$ increases, presumably in a power law. In particular, TwoBo reaches the same variational free energy with almost an order of magnitude smaller $N_{\mathrm{opt}}$ than MADE and RNN, which demonstrates its high efficiency in training. Therefore, we use only $N_{\mathrm{opt}} = 200$ to achieve satisfactory results in Fig. 5.6, and each training can finish in an hour on a high-end GPU.

**Parameter efficiency**

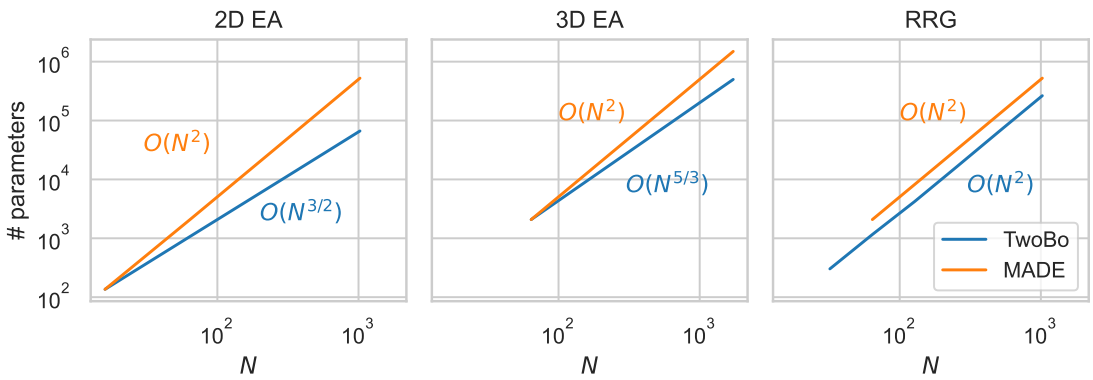



Figure 5.9: Number of trainable parameters in TwoBo and MADE as a function of the system size $N$, on 2D grids, 3D grids, and RRGs. This figure is reproduced from Fig. S1 in the supplementary material for Ref. [4].

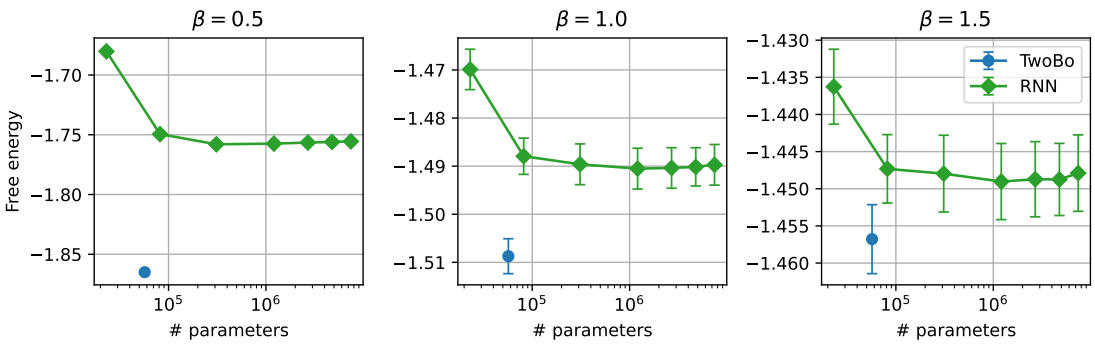



Figure 5.10: Variational free energy of TwoBo compared to RNN with different numbers of memory units $(2, 4, 8, 16, 24, 32, 40)$, on the 2D EA model with $N = 576$. This figure is reproduced from Fig. S3 in the supplementary material for Ref. [4].

Lastly, we investigate the empirical scaling of the number of trainable parameters in TwoBo with the system size. As shown in Fig. 5.9, TwoBo has polynomially fewer parameters than MADE on 2D and 3D grids. However, for the RRGs without a local geometry, TwoBo no longer

has polynomially fewer parameters, but only fewer by a constant coefficient of approximately 2. These results are consistent with the theoretical analysis in Section 5.3.2.

This advantage is more clearly demonstrated when we compare with RNN in the regime of numerous parameters. As shown in Fig. 5.10, the TwoBo with few parameters gives a lower variational free energy than the RNN with a comparable number of parameters, and even if we enlarge the RNN by orders of magnitude, the variational free energy does not systematically decrease. These results emphasize the importance of the lightweight architecture with the physical knowledge in TwoBo, which can be more efficient than scaling up a general neural network architecture with increasing computation.

### 5.3.4 Conclusion

In conclusion, the TwoBo architecture incorporates the knowledge of the Boltzmann distribution and the sparse two-body interacting Hamiltonian into the neural network design, therefore achieves more accurate free energy estimation, faster training speed, and polynomially higher parameter efficiency compared to previous ARNN architectures. It opens a promising way to scale up the variational inference computation to larger systems and extract more physically interpretable results from neural networks.

## 5.4 ARNN in importance sampling

In the previous sections, we have seen the advantages of ARNN over previous ansatzes in variational inference. Meanwhile, as discussed in Section 4.4, when unbiased estimations of observables rather than an upper bound of the free energy is desired, we can also use ARNN with the importance sampling in Eq. (3.27) and the MCMC importance sampling in Eq. (3.31).

Soon after Ref. [35] introduced ARNN in statistical physics problems, Ref. [211] has proposed to use ARNN in the importance sampling and the MCMC importance sampling. However, Ref. [212] has questioned the efficiency of these sampling methods on physical systems with complicated probability landscapes, such as the Potts model and the random graph coloring problem in the random first-order transition (RFOT) universality class. Moreover, Ref. [99] has noticed that a reliable analysis of the autocorrelation time in Eq. (3.15) is largely overlooked in the studies of these sampling methods, which is crucial to measuring the efficiency of MCMC methods. Fortunately, as discussed in Section 3.8, cluster update methods and symmetry operations can help to overcome the problem of mode collapse in first-order phase transitions, while reducing the autocorrelation time as well.

In the following, we present a sampling method proposed in Ref. [1], which constructs cluster updates using the AR property of the ansatz, and applies the symmetries of the physical system, to achieve the unbiased sampling with high efficiency even if the physical system exhibits a first-order phase transition.

### 5.4.1 Neural cluster updates with symmetries (NCUS)

**Algorithm 3** A sampling step in NCUS for a system of $N$ spins on a square lattice with translation, $D_4$ lattice reflection, and $\mathbb{Z}_2$ spin flipping symmetries.

1: Input the current configuration $\boldsymbol{s}$
2: Sample an integer $k \in \{1, \ldots, N\}$ from the distribution $P_{\text{cluster}}(k)$
3: Sample the last $k$ spins and propose the configuration $\boldsymbol{s}'$
4: Accept $\boldsymbol{s} \leftarrow \boldsymbol{s}'$ with the probability in Eq. (5.13)
5: Translate $\boldsymbol{s}$ by a random displacement
6: Reflect $\boldsymbol{s}$ along the $x$-axis, the $y$-axis, and the diagonal, each with 50% probability
7: Reflect $\boldsymbol{s}$ along the $z$-axis (flip all spins) with 50% probability
8: Output $\boldsymbol{s}$ as a sample in the Markov chain

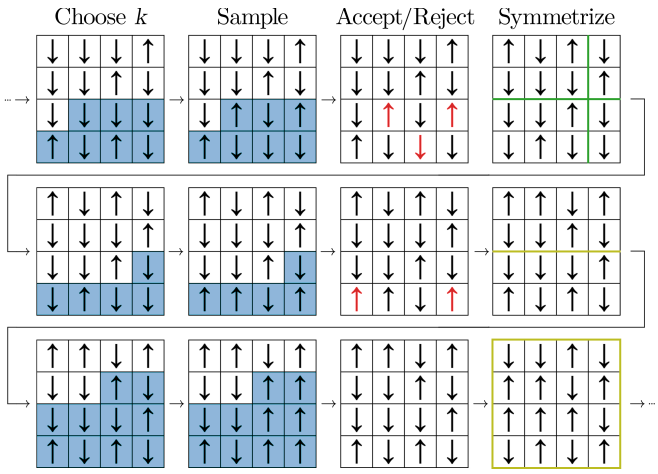



Figure 5.11: Examples of three steps of NCUS applied to a system on the $4 \times 4$ square lattice. The operations in several lines of Algorithm 3 are visualized by colors. In line 2, the last $k$ spins chosen are highlighted in blue. In line 3, some of them are flipped in the proposal. Then in line 4, if the proposal is actually accepted, the flipped spins are shown in red. In line 5, the original borders of the lattice before the translation are shown in green. In line 6, the plane of reflection is shown in yellow, and a reflection along the $z$-axis (across the $xy$-plane) is indicated by yellow borders around the lattice. This figure is reproduced from Fig. 1 in Ref. [1].

During the MCMC importance sampling procedure in Eq. (3.31), we propose new configurations $\boldsymbol{s}'$ from the ansatz $q(\boldsymbol{s})$. When the ansatz is factorized into the AR conditional distributions in Eq. (5.1), we notice that it is possible to avoid generating all $N$ spins from the beginning, but only the last $k$ spins. In this way, the acceptance in Eq. (3.31) becomes

$$A(\boldsymbol{s} \to \boldsymbol{s}') = \min\left(1, \frac{p(\boldsymbol{s}')}{p(\boldsymbol{s})} \prod_{i=N-k+1}^{N} \frac{q_i(s_i \mid \boldsymbol{s}_{<i})}{q_i(s'_i \mid \boldsymbol{s}'_{<i})}\right), \tag{5.13}$$

which is usually close to 1 when $k$ is small. With suitable values of $k$, we can enjoy the

efficiency of cluster updates, which overcome both the inability of local updates to transition through energy barriers and the low acceptance probability of global updates. We refer to the distribution of $k$ as $P_{\text{cluster}}(k)$, and it can be the uniform distribution over $\{1, \ldots, N\}$ without loss of generality. Comparisons between the choices of $P_{\text{cluster}}(k)$ will be presented in Fig. 5.14.

The construction of the above clusters introduces an obvious bias that the spins closer to the end of the autoregressive order will be more frequently flipped. To flip every spin with the same frequency on average, we randomly apply the symmetry operations of the Hamiltonian to the samples in the Markov chain. These operations act as global updates that transition through local energy barriers, and they are never rejected because they conserve the energy and thus have an acceptance probability of 1. Combining the cluster updates and the symmetry operations, we refer to the resulting algorithm as neural cluster updates with symmetries (NCUS). This algorithm is described in Algorithm 3 and visualized in Fig. 5.11.

### 5.4.2 Numerical results

#### Ising model

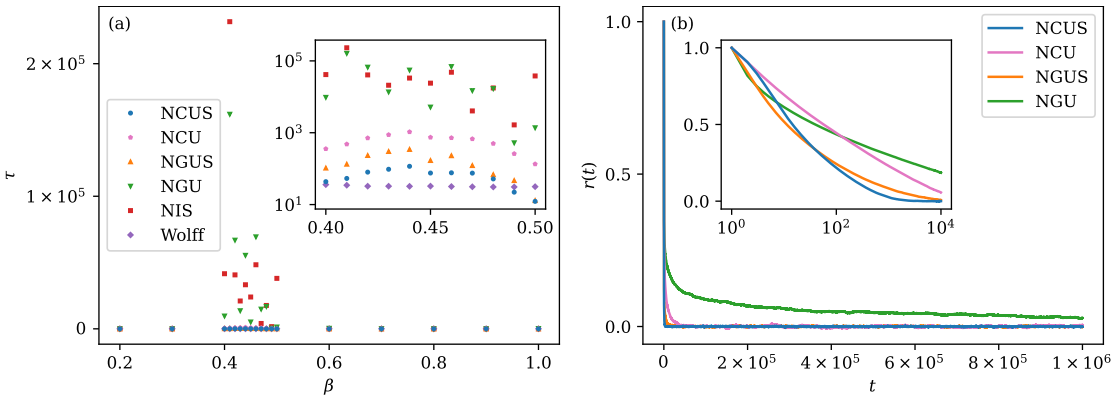



Figure 5.12: (a) Autocorrelation time $\tau$ for the Ising model on the $16 \times 16$ square lattice, at varying inverse temperature $\beta$, given by the sampling methods presented in this section. The inset focuses on their behaviors around the critical point and uses the logarithmic scale on the $y$-axis.
(b) Normalized autocorrelation $r(t)$ at $\beta = 0.44$ around the critical point. The inset uses the logarithmic scale on the $x$-axis to focus on their behaviors at small $t$. This figure is reproduced from Fig. 2 in Ref. [1].

To demonstrate the performance of NCUS, we first conduct numerical experiments using the Ising model on the square lattice with PBC, which has translation, $D_4$ lattice reflection, and $\mathbb{Z}_2$ spin flipping symmetries. We compare NCUS with Eq. (3.27), referred to as neural importance sampling (NIS) here, as well as Eq. (3.31), referred to as neural global updates (NGU) here. As ablation studies, we also consider neural global updates with symmetries (NGUS) and neural cluster updates (NCU) without symmetries. In addition, the well-established Wolff cluster update algorithm [118] is taken for comparison.

In all the neural sampling methods, we use a same neural network trained using the

usual variational method. The network is a convolutional ARNN with approximately $4 \times 10^3$ parameters, which is much smaller compared to the networks used in Section 5.2, as we now emphasize efficient sampling and correction to the non-ideally trained ansatz $q(\boldsymbol{s})$. After the sampling, the mean free energies given by all the methods are apparently close to each other, so we focus on analyzing the autocorrelation times of the free energy as defined in Eq. (3.15), which determine the error bars. For NIS, although it does not produce a Markov chain, we still use the effective sample size in Eqs. (3.18) and (3.30) to define an effective autocorrelation time for comparison.

As shown in Fig. 5.12 (a), around the critical point, NCUS produces autocorrelation times lower by orders of magnitude than the previous neural sampling methods, while the autocorrelation times of NGU and NIS are pathologically high. A closer look at the normalized autocorrelation functions in Fig. 5.12 (b) confirms these results, where the samples from NCUS quickly decorrelate in less than $10^3$ steps, while those from NGU cannot decorrelate after as many as $10^5$ steps. Moreover, the autocorrelation times of NCUS are comparable to those of the Wolff algorithm, although the latter is specifically designed for the Ising model. These results illustrate the advantage of cluster updates and symmetry operations in alleviating the problem of critical slowing down, as discussed in Section 3.5.

**Frustrated plaquette model (FPM)**

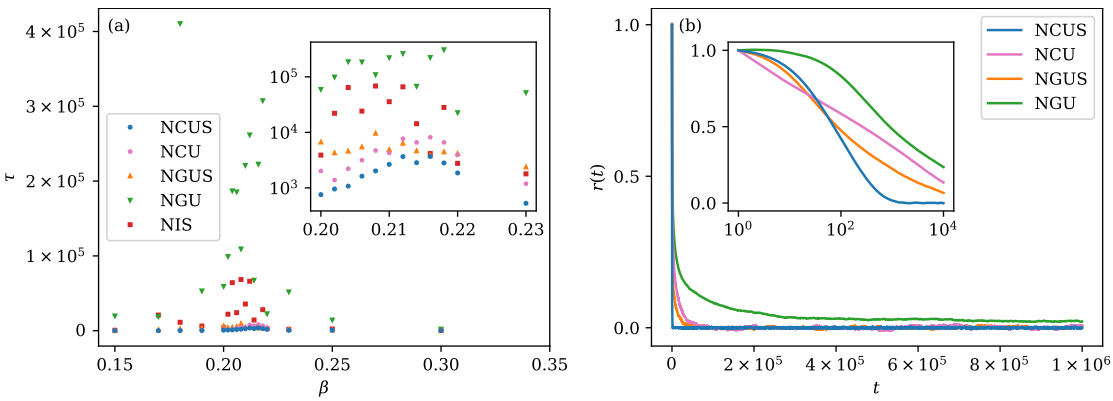



Figure 5.13: (a) Autocorrelation time $\tau$ for the FPM on the $32 \times 32$ square lattice, at varying inverse temperature $\beta$, given by the sampling methods in this section. The inset focuses on their behaviors around the critical point and uses the logarithmic scale on the $y$-axis.
(b) Normalized autocorrelation $r(t)$ at $\beta = 0.2$ around the critical point. The inset uses the logarithmic scale on the $x$-axis to focus on their behaviors at small $t$. This figure is reproduced from Fig. 4 in Ref. [1].

Next, we apply the sampling methods to the frustrated plaquette model (FPM) discussed in Section 2.1.6, which is more difficult than the Ising model due to the first-order phase transition induced by the frustrated $J_3$ interactions and the plaquette interactions. Moreover, we increase the system size from $16 \times 16$ to $32 \times 32$. This time, the Wolff algorithm is no longer applicable as it is designed only for systems with two-body interactions. In contrast, the neural

sampling methods are general enough to handle the plaquette interactions.

As shown in Fig. 5.13, the results have a trend similar to those for the Ising model around the critical point. The autocorrelation times of NCUS can be as high as $O(10^3)$, but it is still practical to obtain reliable estimations of the free energy with the error bars derived from them, and they are lower by orders of magnitude than the results from the previous neural sampling methods. These results provide evidence that cluster updates and symmetry operations help to alleviate the problem of mode collapse with the presence of first-order phase transitions aforementioned in Section 3.8.

#### Choices of the distribution $P_{\text{cluster}}(k)$

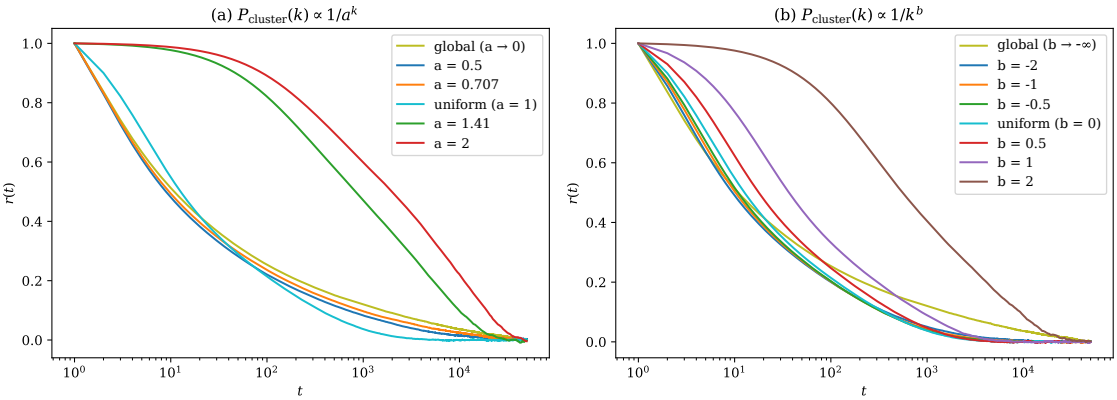



Figure 5.14: Normalized autocorrelation $r(t)$ given by NCUS on the $16 \times 16$ Ising model at inverse temperature $\beta = 0.44$, with different choices of $P_{\text{cluster}}(k)$ from exponential and power distributions. This figure is reproduced from Fig. S1 in the supplemental material for Ref. [1].

The efficiency of NCUS can depend on the distribution $P_{\text{cluster}}(k)$ of the number of spins to generate. In the above results, we have arbitrarily chosen the uniform distribution. Empirical comparisons between different choices of exponential distributions $P_{\text{cluster}}(k) \propto 1/a^k$ and power distributions $P_{\text{cluster}}(k) \propto 1/k^b$, where $a$ and $b$ are adjustable parameters, are shown in Fig. 5.14, respectively. The uniform distribution gives an autocorrelation time comparable or lower to all other distributions we have tested. Moreover, it also outperforms the more naive method of always using the same number, i.e., $P_{\text{cluster}}(k) = \delta(k - k')$ for all $k' \in \{1, \ldots, N\}$. It remains an open question to reveal the relation between the optimal $P_{\text{cluster}}(k)$ and the properties of the target distribution $p(\boldsymbol{s})$ as well as the ansatz $q(\boldsymbol{s})$, using theoretical analysis on the Markov chain transitions.

### 5.4.3 Conclusion

In conclusion, the NCUS algorithm utilizes the cluster updates constructed by the AR property of the ansatz, as well as the symmetries of the physical system, to produce substantially lower autocorrelation times than the conventional global update method around the critical point, and more reliable estimations of observables with lower error bars in consequence. It is worth

mentioning that the recently developed AR models with random orders [213–215] may be used to construct clusters with more general shapes.

We look forward to more research combining the strengths of the techniques presented in this section, including neural networks as highly expressive heuristics for complicated multivariate distributions, autoregressive models with exact sampling and normalized probability evaluation, physical properties such as symmetries and sparsity incorporated into network architectures, and cluster updates that alleviate the problems of critical slowing down and mode collapse, which will eventually lead to solutions for physical systems with unprecedented size and complexity.

# Part III

# Computational studies of quantum systems

# 6 Unbiased evaluation of ground state

The previous part has let us see that exact sampling methods, particularly autoregressive neural networks (ARNNs), achieve higher accuracy and efficiency in approximating classical many-body systems. Now we move on to investigate quantum systems, which are generally known to be more intricate than their classical counterparts. Even though we only focus on their ground states, rather than finite-temperature ensembles or dynamics, they involve peculiar issues that do not exist in the classical world, such as vanishing energy gaps, sign structures, and quantum entanglements.

In this part, we first review the traditional methods of exact diagonalization (ED), path integral Monte Carlo (PIMC), variational Monte Carlo (VMC), and tensor networks (TNs). They not only share some common issues with Markov chain Monte Carlo (MCMC) and variational methods discussed in the previous part but are also affected by peculiar issues in quantum systems. Then we introduce a variational ansatz named tensor-RNN, which combines the strengths of TN and recurrent neural networks (RNNs) to shed new light on both the analytical and the numerical aspects of quantum systems. Lastly, we present the results of VarBench, an extensive project to benchmark the performance of different variational methods on quantum many-body systems These results witness the recent development of this field, including exact sampling methods.

## 6.1 Imaginary time evolution (ITE)

The ground state vector $|\psi_0\rangle$ provides the complete information of a quantum many-body system in its ground state, which allows obtaining all its observables. We start with a straightforward analytical method to evaluate $|\psi_0\rangle$, using the basic properties of eigenstates. We apply the exponential of the system's Hamiltonian $\hat{H}$, with a scaling parameter $\tau$, on an arbitrary initial state $|\psi\rangle$, and obtain

$$|\psi_\tau\rangle = \mathrm{e}^{-\tau\hat{H}}|\psi\rangle, \tag{6.1}$$

where we temporarily ignore the normalization of the states. Because of the orthogonality and the completeness of the eigenstates, the initial state can be decomposed into a linear combination of them:

$$|\psi\rangle = \sum_i c_i |\psi_i\rangle, \tag{6.2}$$

where $|\psi_i\rangle$ is the $i$-th lowest eigenstate, and $c_i$ is the corresponding coefficient that can be determined by projection. Under the action of the Hamiltonian, each eigenstate is scaled by its energy $E_i$:

$$\hat{H}|\psi_i\rangle = E_i|\psi_i\rangle. \tag{6.3}$$

The exponential of an operator is defined by the Taylor expansion:

$$\mathrm{e}^{-\tau\hat{H}} = \sum_{j=0}^{\infty} \frac{1}{j!}(-\tau\hat{H})^j, \tag{6.4}$$

assuming it converges. Therefore, Eq. (6.1) becomes

$$|\psi_\tau\rangle = \sum_i c_i \mathrm{e}^{-\tau E_i}|\psi_i\rangle. \tag{6.5}$$

In the limit $\tau \to \infty$, assuming $E_0 < 0$ with a reasonable energy zero point, the exponential $\mathrm{e}^{-\tau E_0}$ grows faster than all other terms, so the resulting state $|\psi_\tau\rangle$ is dominated by the ground state $|\psi_0\rangle$. This is true regardless of the coefficients $c_i$, as long as $c_0 \neq 0$, i.e., the initial state is not orthogonal to the ground state, which is usually fulfilled if the initial state is chosen randomly. In the case of degenerate ground states, this method produces an arbitrary one in the subspace spanned by them, depending on the choice of the initial state.

Recovering the normalization, we have

$$|\psi_0\rangle = \lim_{\tau\to\infty} \frac{|\psi_\tau\rangle}{\sqrt{\langle\psi_\tau|\psi_\tau\rangle}}. \tag{6.6}$$

This method is known as the imaginary time evolution (ITE) [216], because it has a similar form to the conventional notation of time evolution under the Schrödinger equation:

$$|\psi(t)\rangle = \mathrm{e}^{-\mathrm{i}\hat{H}t}|\psi\rangle, \tag{6.7}$$

which becomes Eq. (6.1) if we let the time be an imaginary number $t = -\mathrm{i}\tau$.

The asymptotic convergence speed of the limit in Eq. (6.6) depends on the energy gap $\Delta E = E_1 - E_0$ between the ground state $|\psi_0\rangle$ and the first excited state $|\psi_1\rangle$, as we can see from the ratio of their overlaps with $|\psi_\tau\rangle$:

$$\frac{\langle\psi_1|\psi_\tau\rangle}{\langle\psi_0|\psi_\tau\rangle} \propto \mathrm{e}^{-\tau\Delta E}. \tag{6.8}$$

It is also related to the condition number of the matrix $\boldsymbol{H} - E_0\boldsymbol{I}$:

$$\kappa = \frac{E_\mathrm{M} - E_0}{E_1 - E_0}, \tag{6.9}$$

where $E_\mathrm{M}$ is the highest energy level of the system, and a higher $\kappa$ indicates that it is more difficult to accurately find the null space of $\boldsymbol{H} - E_0\boldsymbol{I}$, which is equivalent to finding the ground

state of the physical system. This is a root cause of various numerical difficulties in gapless systems, including systems in disordered or critical regimes, such as spin liquids.

## 6.2 Exact diagonalization (ED)

When numerically implementing ITE, we discretize the limit $\tau \to \infty$ into an iterative scheme, and do the normalization in each iteration to ensure numerical stability:

$$\left|\psi'_k\right\rangle = \mathrm{e}^{-\Delta\tau\hat{H}}\left|\psi_k\right\rangle, \tag{6.10}$$

$$\left|\psi_{k+1}\right\rangle = \frac{\left|\psi'_k\right\rangle}{\sqrt{\left\langle\psi'_k|\psi'_k\right\rangle}}. \tag{6.11}$$

Meanwhile, we can only keep the first order in the Taylor expansion of $\mathrm{e}^{-\Delta\tau\hat{H}}$, therefore Eq. (6.4) is approximated by

$$\mathrm{e}^{-\Delta\tau\hat{H}} \approx 1 - \Delta\tau\hat{H}. \tag{6.12}$$

Given that the time step $\Delta\tau$ is small enough and the number of iterations $k$ is large enough, $\left|\psi_k\right\rangle$ converges to the desired $\left|\psi_0\right\rangle$. This method is known as power iteration [217]. It falls in a broader family of iterative methods to compute the lowest one or few eigenvalues and eigenvectors of a matrix, where the Lanczos algorithm [218] is particularly well known, followed by various algorithmic improvements and software implementations [219, 220]. The convergence speeds of these algorithms also depend on the energy gap $\Delta E$, as shown in Eq. (6.8).

In the context of quantum physics, this kind of method is known as exact diagonalization (ED) [75], because they do not involve any approximation or stochastic estimation in the obtained states. However, we can only perform ED on small systems, because the storage of the entire state vector and the multiplication with the Hamiltonian become impractical on larger systems due to the exponentially high dimension of the Hilbert space aforementioned in Section 2.2. As of this writing, the largest ED computation has been performed on 50 spin-$\frac{1}{2}$ particles [221], which utilizes the symmetries of the physical system to reduce the dimension of the Hilbert space, as well as implements sophisticated parallelization and storage strategies on the supercomputer.

Despite its exponential computational complexity, ED is still a necessary method to provide initial insights at small system sizes when investigating an unfamiliar quantum model, and it serves as the ground truth to test the correctness of more advanced computational techniques.

## 6.3 Quantum Monte Carlo (QMC)

On larger systems where the exact evaluation of the ground state is impractical, we can use stochastic methods to approximate it, which fit into the framework of Monte Carlo estimation aforementioned in Section 3.1. Quantum Monte Carlo (QMC) has become a gross term referring to these methods.

In the following, we present a QMC scheme directly following the above derivation of ITE.

Temporarily ignoring the normalization, we discretize each component of the ground state vector $\langle \boldsymbol{s}|\psi_0\rangle$ with

$$\langle \boldsymbol{s}|\psi_0\rangle = \lim_{\tau\to\infty} \langle \boldsymbol{s}|\mathrm{e}^{-\tau\hat{H}}|\psi\rangle \tag{6.13}$$

$$= \lim_{N_\tau\to\infty} \langle \boldsymbol{s}|\left(\mathrm{e}^{-\Delta\tau\hat{H}}\right)^{N_\tau}|\psi\rangle \tag{6.14}$$

$$= \lim_{N_\tau\to\infty} \sum_{\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{N_\tau}} \langle \boldsymbol{s}|\mathrm{e}^{-\Delta\tau\hat{H}}|\boldsymbol{s}_1\rangle \langle \boldsymbol{s}_1|\mathrm{e}^{-\Delta\tau\hat{H}}|\boldsymbol{s}_2\rangle \cdots \langle \boldsymbol{s}_{N_\tau-1}|\mathrm{e}^{-\Delta\tau\hat{H}}|\boldsymbol{s}_{N_\tau}\rangle \langle \boldsymbol{s}_{N_\tau}|\psi\rangle, \tag{6.15}$$

where the auxiliary configurations $\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{N_\tau}$ are inserted using the completeness of basis states. They are used to define the short-time propagator

$$G(\boldsymbol{s},\boldsymbol{s}') = \langle \boldsymbol{s}|\mathrm{e}^{-\Delta\tau\hat{H}}|\boldsymbol{s}'\rangle. \tag{6.16}$$

As long as $\Delta\tau$ is small, $G(\boldsymbol{s},\boldsymbol{s}')$ can be computed with high accuracy, where the Trotter–Suzuki decomposition [222] is usually applied.

Using this discretization to estimate the energy, and recovering the normalization, we have

$$E = \lim_{\tau\to\infty} \frac{\langle \psi|\mathrm{e}^{-\tau\hat{H}}\hat{H}\,\mathrm{e}^{-\tau\hat{H}}|\psi\rangle}{\langle \psi|\mathrm{e}^{-\tau\hat{H}}\mathrm{e}^{-\tau\hat{H}}|\psi\rangle} \tag{6.17}$$

$$= \lim_{\tau\to\infty} \frac{\langle \psi|\mathrm{e}^{-2\tau\hat{H}}\hat{H}|\psi\rangle}{\langle \psi|\mathrm{e}^{-2\tau\hat{H}}|\psi\rangle} \tag{6.18}$$

$$= \lim_{\tau\to\infty} \frac{\sum_{\boldsymbol{s},\boldsymbol{s}'} \langle \psi|\mathrm{e}^{-2\tau\hat{H}}|\boldsymbol{s}\rangle \langle \boldsymbol{s}|\hat{H}|\boldsymbol{s}'\rangle \langle \boldsymbol{s}'|\psi\rangle}{\langle \psi|\mathrm{e}^{-2\tau\hat{H}}|\psi\rangle} \tag{6.19}$$

$$= \lim_{\tau\to\infty} \frac{\sum_{\boldsymbol{s}} \langle \psi|\mathrm{e}^{-2\tau\hat{H}}|\boldsymbol{s}\rangle\langle \boldsymbol{s}|\psi\rangle \sum_{\boldsymbol{s}'} \langle \boldsymbol{s}|\hat{H}|\boldsymbol{s}'\rangle \frac{\langle \boldsymbol{s}'|\psi\rangle}{\langle \boldsymbol{s}|\psi\rangle}}{\langle \psi|\mathrm{e}^{-2\tau\hat{H}}|\psi\rangle} \tag{6.20}$$

$$= \lim_{N_\tau\to\infty} \sum_{\mathfrak{s}} \Pi(\mathfrak{s}) E_{\mathrm{loc}}(\boldsymbol{s}_{N_\tau}), \tag{6.21}$$

where $\mathfrak{s} = (\boldsymbol{s}_0,\boldsymbol{s}_1,\ldots,\boldsymbol{s}_{N_\tau})$ denotes all the auxiliary configurations, also known as the integration path, and

$$\Pi(\mathfrak{s}) = \frac{\tilde{\Pi}(\mathfrak{s})}{\sum'_{\mathfrak{s}} \tilde{\Pi}(\mathfrak{s}')}, \tag{6.22}$$

$$\tilde{\Pi}(\mathfrak{s}) = \psi^*(\boldsymbol{s}_0)G(\boldsymbol{s}_0,\boldsymbol{s}_1)G(\boldsymbol{s}_1,\boldsymbol{s}_2)\cdots G(\boldsymbol{s}_{N_\tau-1},\boldsymbol{s}_{N_\tau})\psi(\boldsymbol{s}_{N_\tau}), \tag{6.23}$$

$$E_{\mathrm{loc}}(\boldsymbol{s}) = \sum_{\boldsymbol{s}'} H(\boldsymbol{s},\boldsymbol{s}')\frac{\psi(\boldsymbol{s}')}{\psi(\boldsymbol{s})}. \tag{6.24}$$

Although the local energy $E_{\mathrm{loc}}(\boldsymbol{s})$ appears to be a sum over exponentially many configurations, in most physical systems of interest, the Hamiltonian is indeed locally interacting, i.e., it is a sum of polynomially many terms, and each term is a product of operators on few sites, usually connected in a regular geometry. Therefore, the non-zero entries $H(\boldsymbol{s},\boldsymbol{s}')$ are only

polynomially many and can be efficiently evaluated according to the sparsity structure of $\hat{H}$.

The resulting Eq. (6.21) has a form similar to the weighted summation in Eq. (2.4). If the weight $\Pi(\mathfrak{s}) \geq 0$ for all $\mathfrak{s}$, we can interpret it as a probability distribution and estimate Eq. (6.21) using the Monte Carlo summation in Eq. (3.1). This method requires samples of $\mathfrak{s}$ generated from $\Pi(\mathfrak{s}) \geq 0$, and each sample contains $N(N_\tau + 1)$ scalar variables, where $N$ is the system size. This is an unbiased estimator of $E$ only when $N_\tau \to \infty$, and in practice we usually need a large $N_\tau$ such that $N_\tau \Delta\tau \gg 1$, which leads to much higher computational cost than Monte Carlo methods for the corresponding classical system. Evaluation of Eq. (6.23) not only requires accurately and efficiently computing the propagator $G(\boldsymbol{s}, \boldsymbol{s}')$, but also the initial wave function $\psi(\boldsymbol{s})$, and a good choice of $|\psi\rangle$ that is close to the ground state can substantially reduce the magnitude of $N_\tau$ needed. In addition, we note that Eq. (6.18) holds not only when estimating the energy, but also any observable that commutes with $\hat{H}$.

This method is commonly referred to as the path integral Monte Carlo (PIMC) [223, 224], and a particular method to sample $\mathfrak{s}$ is known as the reptation Monte Carlo [225]. For some physical systems, this kind of QMC computations have successfully achieved numerically exact results [226, 227], which can serve as the ground truth to test the correctness of other methods even if the ED result is unavailable.

## 6.4 Sign problem

The above discussion reveals a peculiar caveat in quantum systems: The distribution of a classical ensemble always has $p(\boldsymbol{s}) \geq 0$, but the weight in Eq. (6.21) may not satisfy $\Pi(\mathfrak{s}) \geq 0$, and the Monte Carlo estimator is invalid in this case, which is known as the sign problem [228]. In spin systems, the sign problem is usually a result of frustrated interactions between the particles [229]. It is even more prevalent in fermionic systems, where the wave function frequently changes its sign because of the anti-commutation of the particles.

A quick remedy is to factor out the sign from the weight:

$$E = \lim_{N_\tau \to \infty} \sum_{\mathfrak{s}} \Pi'(\mathfrak{s}) E'_{\text{loc}}(\mathfrak{s}), \tag{6.25}$$

$$\Pi'(\mathfrak{s}) = \frac{|\tilde{\Pi}(\mathfrak{s})|}{\sum'_{\mathfrak{s}} |\tilde{\Pi}(\mathfrak{s}')|}, \tag{6.26}$$

$$E'_{\text{loc}}(\mathfrak{s}) = \operatorname{sign}\left(\tilde{\Pi}(\mathfrak{s})\right) E_{\text{loc}}(\boldsymbol{s}_{N_\tau}). \tag{6.27}$$

However, the intrinsic complexity of the wave function does not simply disappear. Unlike the classical local free energy $F_{\text{loc}}(\boldsymbol{s})$ in Eq. (4.5), which is usually far below zero when $s$ is around the ground state, the local energy $E'_{\text{loc}}(\mathfrak{s})$ in Eq. (6.27) can frequently change its sign as the integration path $\mathfrak{s}$ changes. In Eq. (6.25), there can be a cancellation of many positive and negative terms, which leads to a small mean value and a large variance of the estimator, and prevents the result from achieving high accuracy with reliable error estimation.

As a well-studied case, if a Hamiltonian is stoquastic, i.e., all its off-diagonal entries are non-positive [230], then it is guaranteed to be free of the sign problem, i.e., $\psi_0(\boldsymbol{s}) \geq 0$ for all $\boldsymbol{s}$. For example, the ferromagnetic (FM) Heisenberg model on a 1D chain is stoquastic, while the

antiferromagnetic (AFM) one is not. Fortunately, the sign problem of a Hamiltonian can be removed with a change of basis in some cases. In the previous example, an AFM Heisenberg model on a non-frustrated graph can be converted to an FM one by applying the Marshall sign rule [231]. However, such a transformation is not available if the sign problem arises from the intrinsic complexity of the wave function, rather than the apparent choice of the basis [232].

Various other schemes of QMC have been proposed to alleviate the sign problem, such as diffusion Monte Carlo (DMC) [233–235] and auxiliary-field quantum Monte Carlo (AFQMC, also known as the determinant QMC) [236, 237], which are beyond the scope of this thesis. In the next chapter, we present a detailed review of another QMC scheme, namely variational Monte Carlo (VMC), upon which this thesis mainly develops.

# 7 Variational Monte Carlo (VMC)

The PIMC method in the previous chapter allows us to obtain an unbiased stochastic approximation of the ground state properties for quantum many-body problems, in the same manner as the MCMC method in Chapter 3. Apart from that approach, another QMC scheme has been proposed to obtain a variational approximation of the target system, which bears the name variational Monte Carlo (VMC) [21, 238] and resembles the variational method in Chapter 4. Unlike classical systems where MCMC suffices to obtain accurate results in many cases, in quantum systems the sheer large amount of computation required by unbiased methods has motivated a wide usage of variational methods, as we will see in the following.

## 7.1 Variational energy

The ground state of a quantum system in Eq. (2.22) is already defined by the variational principle, and we only need to rewrite it in a form suitable for Monte Carlo summation. Let $\psi(\boldsymbol{s})$ be the variational ansatz, which is not necessarily normalized, then its energy can be written as

$$E_\psi = \frac{\langle\psi|\hat{H}|\psi\rangle}{\langle\psi|\psi\rangle} \tag{7.1}$$

$$= \frac{\sum_{\boldsymbol{s},\boldsymbol{s}'}\langle\psi|\boldsymbol{s}\rangle\langle\boldsymbol{s}|\hat{H}|\boldsymbol{s}'\rangle\langle\boldsymbol{s}'|\psi\rangle}{\langle\psi|\psi\rangle} \tag{7.2}$$

$$= \frac{\sum_{\boldsymbol{s}}\langle\psi|\boldsymbol{s}\rangle\langle\boldsymbol{s}|\psi\rangle\sum_{\boldsymbol{s}'}\langle\boldsymbol{s}|\hat{H}|\boldsymbol{s}'\rangle\frac{\langle\boldsymbol{s}'|\psi\rangle}{\langle\boldsymbol{s}|\psi\rangle}}{\langle\psi|\psi\rangle} \tag{7.3}$$

$$= \sum_{\boldsymbol{s}} q(\boldsymbol{s})E_{\mathrm{loc}}(\boldsymbol{s}), \tag{7.4}$$

where the probability

$$q(\boldsymbol{s}) = \frac{|\psi(\boldsymbol{s})|^2}{\langle\psi|\psi\rangle}, \tag{7.5}$$

and the local energy

$$E_{\mathrm{loc}}(\boldsymbol{s}) = \sum_{\boldsymbol{s}'} H(\boldsymbol{s},\boldsymbol{s}')\frac{\psi(\boldsymbol{s}')}{\psi(\boldsymbol{s})}. \tag{7.6}$$

The variational energy in Eq. (7.4) has a similar form to the variational free energy in Eq. (4.4). In the same manner discussed there, we can generate samples of the configuration $\boldsymbol{s}$ from $q(\boldsymbol{s})$, estimate $E_\psi$ using the Monte Carlo summation, then minimize it w.r.t. the parameters in $\psi(\boldsymbol{s})$ using gradient-based optimizers, and a lower $E_\psi$ indicates a better ansatz to approximate the ground state. It is worth mentioning that the state $|\psi\rangle$ obtained from variational optimization can be further improved by iterative eigenvector solvers in Section 6.2 [79, 239], or used as the initial state for other QMC schemes in Section 6.3.

When the Hamiltonian is non-stoquastic, unlike the weight $\Pi(\mathfrak{s})$ for PIMC in Eq. (6.21), in VMC we always have $q(\boldsymbol{s}) \geq 0$. However, the sign problem can still occur in the sense that $E_{\text{loc}}(\boldsymbol{s})$ can be either positive or negative, which further causes large noise in the estimated gradient and impedes the optimization.

## 7.2 Energy variance

Besides the energy itself, its variance $\text{Var}E_\psi = \langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2$ under the state $|\psi\rangle$ can also be estimated by Monte Carlo sampling, where the first term can be written as

$$\frac{\langle \psi | \hat{H}^2 | \psi \rangle}{\langle \psi | \psi \rangle} = \frac{\sum_{\boldsymbol{s},\boldsymbol{s}',\boldsymbol{s}''} \langle \psi | \boldsymbol{s}' \rangle \langle \boldsymbol{s}' | \hat{H} | \boldsymbol{s} \rangle \langle \boldsymbol{s} | \hat{H} | \boldsymbol{s}'' \rangle \langle \boldsymbol{s}'' | \psi \rangle}{\langle \psi | \psi \rangle} \tag{7.7}$$

$$= \frac{\sum_{\boldsymbol{s}} \langle \psi | \boldsymbol{s} \rangle \langle \boldsymbol{s} | \psi \rangle \left( \sum_{\boldsymbol{s}'} \frac{\langle \psi | \boldsymbol{s}' \rangle}{\langle \psi | \boldsymbol{s} \rangle} \langle \boldsymbol{s}' | \hat{H} | \boldsymbol{s} \rangle \right) \left( \sum_{\boldsymbol{s}''} \langle \boldsymbol{s} | \hat{H} | \boldsymbol{s}'' \rangle \frac{\langle \boldsymbol{s}'' | \psi \rangle}{\langle \boldsymbol{s} | \psi \rangle} \right)}{\langle \psi | \psi \rangle} \tag{7.8}$$

(Assuming $\langle \boldsymbol{s} | \psi \rangle \neq 0$ for all $\boldsymbol{s}$)

$$= \sum_{\boldsymbol{s}} q(\boldsymbol{s}) |E_{\text{loc}}(\boldsymbol{s})|^2. \tag{7.9}$$

Therefore, the variance of the energy happens to be the same as the variance of the local energy in Eq. (7.4). This variance has the noteworthy property that it reaches zero when the variational state $|\psi\rangle$ reaches the ground state $|\psi_0\rangle$, which is useful to indicate the progress of the variational optimization. For a simple example, in a two-state system $|\psi\rangle = \sqrt{\lambda}|\psi_0\rangle + \sqrt{1-\lambda}|\psi_1\rangle$, where $\lambda \in [0,1]$ is a tunable parameter, we have

$$\text{Var}E_\psi = \lambda(1-\lambda)\Delta E^2, \tag{7.10}$$

where $\Delta E = E_1 - E_0$ is the energy gap, and we can see $\text{Var}E_\psi = 0$ when $\lambda = 1$. However, the variance is also zero if $|\psi\rangle$ is trapped in any excited state, as we can see $\text{Var}E_\psi = 0$ when $\lambda = 0$ in Eq. (7.10). This will be discussed in more depth in Chapter 10.

Caution should be taken that we have assumed $\langle \boldsymbol{s} | \psi \rangle \neq 0$ for all $\boldsymbol{s}$ when inserting $\langle \boldsymbol{s} | \psi \rangle$ into the denominators in Eq. (7.8). If this assumption is not fulfilled, then the summation in Eq. (7.9) will only contain the terms with $\langle \boldsymbol{s} | \psi \rangle \neq 0$, and the variance of the local energy will be a biased estimator of the true variance, as pointed out in Ref. [240]. In practice, when some $\langle \boldsymbol{s} | \psi \rangle$ are exponentially suppressed and numerically close to zero, they can cause high variance of the variance estimator, which means that the estimated variance and therefore the convergence of the optimization is unreliable. For example, if the ansatz is pathologically

concentrated such that we can only sample a single configuration $\boldsymbol{s}$ from it, then the local energy $E_{\text{loc}}(\boldsymbol{s})$ always has the same value, and the estimated $\text{Var}E_\psi$ is zero. This problem is similar to the mode collapse aforementioned in Section 3.8, and in the quantum case it can occur even though we are using the variational approximation rather than the unbiased approximation. Note that this problem does not occur when estimating the energy, because we have $\lim_{\psi(\boldsymbol{s})\to 0} q(\boldsymbol{s})E_{\text{loc}}(\boldsymbol{s}) = 0$, while it occurs in the variance, because $\psi(\boldsymbol{s})$ cancels out in the numerator and the denominator in $q(\boldsymbol{s})|E_{\text{loc}}(\boldsymbol{s})|^2$.

## 7.3 Complex gradient

Similar to Eq. (4.15), the gradient of $E_\psi$ also needs to be derived in a form suitable for Monte Carlo summation. In the general case where the Hamiltonian $\hat{H}$, the wave function $\psi(\boldsymbol{s})$, and the parameters $\theta$ can be complex valued, we have

$$\frac{\partial E_\psi}{\partial \theta} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left( \left(E_{\text{loc}}(\boldsymbol{s}) - E_\psi\right)^* \frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta} + \left(E_{\text{loc}}(\boldsymbol{s}) - E_\psi\right) \frac{\partial \ln \psi^*(\boldsymbol{s})}{\partial \theta} \right), \tag{7.11}$$

whose derivation is presented in Appendix A. In the simple case where $\hat{H}$, $\psi(\boldsymbol{s})$, and $\theta$ are all real-valued, it simplifies to

$$\frac{\partial E_\psi}{\partial \theta} = 2\sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left(E_{\text{loc}}(\boldsymbol{s}) - E_\psi\right) \boldsymbol{D}(\boldsymbol{s}), \tag{7.12}$$

where $\boldsymbol{D}(\boldsymbol{s}) = \frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta}$ denotes the gradient vector of $\ln \psi(\boldsymbol{s})$. To reduce the variance when estimating the gradient using Monte Carlo sampling, in the same manner as Eq. (4.16), we shift $\boldsymbol{D}(\boldsymbol{s})$ to have zero mean, without changing the expectation of the gradient:

$$\frac{\partial E_\psi}{\partial \theta} = 2\sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left(E_{\text{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\boldsymbol{D}(\boldsymbol{s}) - \bar{\boldsymbol{D}}\right), \tag{7.13}$$

$$\bar{\boldsymbol{D}} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \boldsymbol{D}(\boldsymbol{s}). \tag{7.14}$$

A peculiar nature of quantum systems is that $\hat{H}$, $\psi(\boldsymbol{s})$, and $\theta$ can actually be complex valued, which has caused substantial confusion because there is no universally accepted and applicable definition for the gradient of a complex-valued function w.r.t. complex parameters. For example, the usual definition in complex analysis requires that $\lim_{\Delta z\to 0} \frac{f(z+\Delta z)-f(z)}{\Delta z}$ exists for all directions of $\Delta z$ on the complex plane [241], which is usually unfulfilled for complicated functions such as neural networks [242]. Even if this condition is fulfilled, it usually implies that $f(z)$ is holomorphic and has singular points[1], which impedes the numerical stability of the variational optimization.

Fortunately, the variational energy $E_\psi$ is guaranteed to be real when the summation in Eq. (7.4) is performed exactly, which allows us to define a "split" gradient for the purpose of the

[1] Including the singular point at infinity, i.e., $|f(z)| \to \infty$ as $|z| \to \infty$, which is usually undesired for localized wave functions.

gradient descent (GD) optimizer in Eq. (4.7). Assuming $\theta = \theta_{\mathrm{R}} + \mathrm{i}\theta_{\mathrm{I}}$, we compute the gradient of the real function $E_\psi$ w.r.t. the real parameters $\theta_{\mathrm{R}}$ and $\theta_{\mathrm{I}}$ respectively, then combine them by

$$\frac{\partial E_\psi}{\partial \theta} = \frac{\partial E_\psi}{\partial \theta_{\mathrm{R}}} + \mathrm{i}\frac{\partial E_\psi}{\partial \theta_{\mathrm{I}}}, \tag{7.15}$$

which yields the correct direction of optimization when substituted into Eq. (4.7). Using this definition, the gradient in Eq. (7.11) becomes

$$\begin{aligned}\frac{\partial E_\psi}{\partial \theta} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \Bigg( & \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right)^* \left(\frac{\partial}{\partial \theta_{\mathrm{R}}} + \mathrm{i}\frac{\partial}{\partial \theta_{\mathrm{I}}}\right) \ln \psi(\boldsymbol{s}) \\ & + \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial}{\partial \theta_{\mathrm{R}}} + \mathrm{i}\frac{\partial}{\partial \theta_{\mathrm{I}}}\right) \ln \psi^*(\boldsymbol{s}) \Bigg).\end{aligned} \tag{7.16}$$

Because $\frac{\partial f^*(x)}{\partial x} = \left(\frac{\partial f(x)}{\partial x}\right)^*$ when $x$ is real, we have

$$\begin{aligned}\frac{\partial E_\psi}{\partial \theta} = & \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left( \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right)^* \frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{R}}} + \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{R}}}\right)^* \right) \\ & + \mathrm{i} \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left( \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right)^* \frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{I}}} + \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{I}}}\right)^* \right) \qquad (7.17) \\ = & 2 \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left( \mathrm{Re}\left( \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{R}}}\right)^* \right) + \mathrm{i}\,\mathrm{Re}\left( \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{I}}}\right)^* \right) \right). \qquad (7.18)\end{aligned}$$

In addition, we can reduce its variance in the same manner as Eq. (7.13).

The same gradient, except differing by an overall coefficient of 2 because of the convention, can be derived from the viewpoint of the Wirtinger calculus [243], also known as the $\mathbb{CR}$-calculus [244]. In this theory, any complex-valued function can be written as $f(z, z^*)$, which is holomorphic in $z$ and $z^*$ respectively, and we can take the gradient $\frac{\partial f}{\partial z}$ and the co-gradient $\frac{\partial f}{\partial z^*}$ while treating the other variable as constant. It has been proven that

$$\frac{\partial f}{\partial z} = \frac{1}{2}\left(\frac{\partial f}{\partial x} - \mathrm{i}\frac{\partial f}{\partial y}\right), \quad \frac{\partial f}{\partial z^*} = \frac{1}{2}\left(\frac{\partial f}{\partial x} + \mathrm{i}\frac{\partial f}{\partial y}\right), \tag{7.19}$$

where $z = x + \mathrm{i}y$. For the purpose of GD, we substitute the co-gradient $\frac{\partial E_\psi}{\partial \theta^*}$ into Eq. (4.7), which is the same as Eq. (7.18) except without the coefficient of 2. Automatic differentiation software can use the Wirtinger calculus with the chain rule of derivatives to compute $\frac{\partial \ln \psi}{\partial \theta^*}$ [245], which usually takes less computation time than computing $\frac{\partial \ln \psi}{\partial \theta_{\mathrm{R}}}$ and $\frac{\partial \ln \psi}{\partial \theta_{\mathrm{I}}}$ separately, but care should be taken that different software can have different conventions for the sign of the imaginary part and the overall coefficient.

It is worth discussing the Re notation in Eq. (7.18). The gradients $\frac{\partial E_\psi}{\partial \theta_{\mathrm{R}}}$ and $\frac{\partial E_\psi}{\partial \theta_{\mathrm{I}}}$ are both real when evaluated exactly, because they are gradients of a real-valued function w.r.t. real parameters. Therefore, the sums $\sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{R}}}\right)^*$ and $\sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left(E_{\mathrm{loc}}(\boldsymbol{s}) - E_\psi\right) \left(\frac{\partial \ln \psi(\boldsymbol{s})}{\partial \theta_{\mathrm{I}}}\right)^*$ are also real when evaluated exactly. However, when using Monte Carlo sampling, the estimators $\mathbb{E}_{\mathrm{MC}}\left[\left(E_{\mathrm{loc}} - E_\psi\right) \left(\frac{\partial \ln \psi}{\partial \theta_{\mathrm{R}}}\right)^*\right]$ and $\mathbb{E}_{\mathrm{MC}}\left[\left(E_{\mathrm{loc}} - E_\psi\right) \left(\frac{\partial \ln \psi}{\partial \theta_{\mathrm{I}}}\right)^*\right]$ can have imaginary parts because

of the discrepancy of samples, which are discarded in Eq. (7.18) before outputting the gradient and updating the parameters in Eq. (4.7).

In some implementations of VMC, as well as stochastic gradient descent (SGD) in other complex-valued optimization problems, the Re notation is ignored either deliberately or unintentionally, and the imaginary parts of the estimated $\frac{\partial E_\psi}{\partial \theta_\mathrm{R}}$ and $\frac{\partial E_\psi}{\partial \theta_\mathrm{I}}$ are mixed in the updates to $\theta_\mathrm{I}$ and $\theta_\mathrm{R}$, respectively. This additional noise in the gradient may affect the convergence of SGD, as discussed in Section 4.2.1. To the author's knowledge, there is no strong evidence that either implementation consistently outperforms the other. However, an occasionally occurring mistake is to take the real parts before multiplying, i.e., $\mathrm{Re}\left(E_\mathrm{loc} - E_\psi\right)\mathrm{Re}\left(\frac{\partial \ln \psi}{\partial \theta_\mathrm{R}}\right)$, which changes the result even if the summation over $\boldsymbol{s}$ is performed exactly. We refer to Ref. [242] as a recent survey on complex-valued neural networks, including their optimization, in various fields of machine learning.

## 7.4 Stochastic reconfiguration (SR)

Beyond the simple GD optimizer, a commonly used optimization method in VMC is stochastic reconfiguration (SR)[2]. It is naturally derived as an approximation of the imaginary time evolution (ITE) in Section 6.1, and is similar to the natural gradient descent (NGD) in Section 4.2.2.

We consider the ITE of the state $|\psi\rangle$ with a small step $\Delta\tau$:

$$|\psi_{\Delta\tau}\rangle = \mathrm{e}^{-\Delta\tau \hat{H}}|\psi\rangle = |\psi\rangle - \Delta\tau \hat{H}|\psi\rangle + O(\Delta\tau^2), \tag{7.20}$$

and project it onto a Hilbert subspace generated by the states $\{|\psi\rangle, |\partial_1\psi\rangle, |\partial_2\psi\rangle, \ldots, |\partial_{n_\mathrm{p}}\psi\rangle\}$, where we denote $|\partial_i\psi\rangle = \frac{\partial}{\partial\theta_i}|\psi\rangle$, and $n_\mathrm{p}$ is the number of parameters. This space contains all the states that the ansatz can reach by varying the parameters, up to the local linear approximation. To make the projection, we try to write $|\psi_{\Delta\tau}\rangle$ as a linear combination of these basis states:

$$|\psi_{\Delta\tau}\rangle = \alpha|\psi\rangle + \sum_i \Delta\theta_i |\partial_i\psi\rangle, \tag{7.21}$$

where $\{\Delta\theta_i\}$ will be used as the updates to the parameters, and $\alpha$ represents the change in the magnitude of $|\psi\rangle$. The dimension of the subspace $n_\mathrm{p}+1$ is usually much less than that of the whole Hilbert space $2^N$, so Eq. (7.21) does not hold exactly, but we require it to hold when multiplying both sides with each basis state:

$$\langle\psi|\psi_{\Delta\tau}\rangle = \alpha\langle\psi|\psi\rangle + \sum_i \Delta\theta_i \langle\psi|\partial_i\psi\rangle, \tag{7.22}$$

$$\langle\partial_j\psi|\psi_{\Delta\tau}\rangle = \alpha\langle\partial_j\psi|\psi\rangle + \sum_i \Delta\theta_i \langle\partial_j\psi|\partial_i\psi\rangle. \tag{7.23}$$

[2]The name "stochastic reconfiguration" is also used in diffusion Monte Carlo (DMC), where it refers to the technique to reconfigure the number of walkers [246]. Both names originate from the earlier work on Green's function Monte Carlo [247].

Cancelling $\alpha$, and ignoring $O(\Delta\tau^2)$ terms, the equations above simplify to

$$\boldsymbol{S}\Delta\theta = -\frac{\Delta\tau}{2}\boldsymbol{f}, \tag{7.24}$$

where $\boldsymbol{f}$ is the gradient in Eq. (7.12), commonly known as the "force" in the context of VMC, and $\boldsymbol{S}$ is an $n_\mathrm{p} \times n_\mathrm{p}$ matrix define by

$$S_{ij} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \left(D_i(\boldsymbol{s}) - \bar{D}_i\right)^* \left(D_j(\boldsymbol{s}) - \bar{D}_j\right). \tag{7.25}$$

The details on the derivation of Eq. (7.24) are presented in Appendix B. It has the same form in the real and the complex cases, as long as the same definition of the complex gradient is used in $\boldsymbol{S}$ and $\boldsymbol{f}$. In the same form as the preconditioned GD in Eq. (4.8), the parameters are updated by

$$\theta_{t+1} \leftarrow \theta_t - \gamma\, \boldsymbol{S}^{-1}\boldsymbol{f}\big|_{\theta=\theta_t}, \tag{7.26}$$

where $\gamma = \frac{\Delta\tau}{2}$ is the learning rate. Therefore, the SR algorithm is also known as the quantum natural gradient descent [248]. Alternatively, it can be derived from the viewpoint of the Fubini–Study metric [249], which replaces the classical case of the KL divergence in Eq. (4.2), and $\boldsymbol{S}$ is called the quantum geometric tensor (QGT) [250], which generalizes the Fisher information matrix in Eq. (4.9) into the quantum case.

In practice, we usually need to estimate $\boldsymbol{S}$ along with $\boldsymbol{f}$ using Monte Carlo sampling:

$$\theta_{t+1} \leftarrow \theta_t - \gamma\, \tilde{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{f}}\big|_{\theta=\theta_t}, \tag{7.27}$$

$$\tilde{\boldsymbol{S}} = \frac{1}{M}\sum_{i=1}^{M}\left(\boldsymbol{D}\big(\boldsymbol{s}^{(i)}\big) - \tilde{\boldsymbol{D}}\right)^*\left(\boldsymbol{D}\big(\boldsymbol{s}^{(i)}\big) - \tilde{\boldsymbol{D}}\right)^\top, \tag{7.28}$$

$$\tilde{\boldsymbol{f}} = \frac{2}{M}\sum_{i=1}^{M}\left(E_\mathrm{loc}\big(\boldsymbol{s}^{(i)}\big) - \tilde{E}\right)\left(\boldsymbol{D}\big(\boldsymbol{s}^{(i)}\big) - \tilde{\boldsymbol{D}}\right)^*, \tag{7.29}$$

$$\tilde{E} = \frac{1}{M}\sum_{i=1}^{M} E_\mathrm{loc}\big(\boldsymbol{s}^{(i)}\big), \tag{7.30}$$

$$\tilde{\boldsymbol{D}} = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{D}\big(\boldsymbol{s}^{(i)}\big), \quad \boldsymbol{s}^{(i)} \sim q, \tag{7.31}$$

where $M$ is the number of samples, and usually $M < n_\mathrm{p}$. In this case, a problem occurs that each term $\boldsymbol{D}(\boldsymbol{s})^*\boldsymbol{D}(\boldsymbol{s})^\top$ is a rank-1 matrix, so $\tilde{\boldsymbol{S}}$ has a rank of at most $M$, and it cannot be inverted in Eq. (7.26). A common remedy is to add a diagonal term to $\tilde{\boldsymbol{S}}$ before inverting it:

$$\theta_{t+1} \leftarrow \theta_t - \gamma\, (\tilde{\boldsymbol{S}} + \lambda\boldsymbol{I})^{-1}\tilde{\boldsymbol{f}}\big|_{\theta=\theta_t}. \tag{7.32}$$

It is technically known as the Tikhonov regularization [251] and colloquially referred to as the "diagonal shift", and it also has resemblance to the Levenberg–Marquardt algorithm discussed in Section 4.2.2 as an interpolation between the first- and the second-order updates. In the extreme case that $\lambda \to \infty$ while $\frac{\gamma}{\lambda}$ is fixed, Eq. (7.32) degenerates to the simple GD in Eq. (4.7). In addition to the learning rate $\gamma$, the diagonal shift $\lambda$ is another crucial hyperparameter

in VMC that usually requires manual tuning. We may use a large $\lambda$ at the beginning of the optimization to ensure stability, and decrease it as the optimization approaches the minimum.

Recently, a different approach to invert $\tilde{\boldsymbol{S}}$ has been proposed in Refs. [130, 252]. The estimated update to the parameters can be expressed as

$$\theta_{t+1} \leftarrow \theta_t - \gamma \left(\boldsymbol{X}\boldsymbol{X}^\dagger + \lambda \boldsymbol{I}\right)^{-1} \boldsymbol{X}\tilde{\boldsymbol{f}}\Big|_{\theta=\theta_t}, \tag{7.33}$$

which has the same form as the common definition of the pseudoinverse of a singular matrix, where $\boldsymbol{X}$ is a $M \times n_\mathrm{P}$ matrix defined by

$$X_{ij} = D_j\left(\boldsymbol{s}^{(i)}\right) - \tilde{D}_j. \tag{7.34}$$

This method is named MinSR in Ref. [130], and it is gradually becoming the default implementation of SR when $M < n_\mathrm{p}$. Note that even if $\boldsymbol{X}\boldsymbol{X}^\dagger$ is full rank, the diagonal shift $\lambda$ is usually still needed to improve stability, because the limited number of samples can cause substantial noise in $\tilde{\boldsymbol{S}}$. It is worth mentioning that previously the large size of $\tilde{\boldsymbol{S}}$ requires us to use iterative solvers for the linear system in Eq. (7.24), which avoid storing all the $n_\mathrm{p} \times n_\mathrm{p}$ entries in $\tilde{\boldsymbol{S}}$. As the size of the matrix to reduce is substantially smaller in MinSR, we can use exact solvers instead, which enable more efficient parallelization on GPUs, and have fewer hyperparameters prone to manual tuning.

However, $\boldsymbol{X}\boldsymbol{X}^\dagger$ can still be singular if the expressiveness of the ansatz is limited, which is indicated by the degeneracy in the basis vectors of the locally linearized variational space in Eq. (7.21). Worse still, the magnitude of $\psi(\boldsymbol{s})$ can be exponentially suppressed for many $\boldsymbol{s}$, which causes numerically zero eigenvalues in $\boldsymbol{S}$, and the number of numerically non-zero eigenvalues can be less than $M$ when the wave function is highly concentrated. These zero or near-zero eigenvalues correspond to zero or near-zero entries in $\boldsymbol{X}\tilde{\boldsymbol{f}}$ in principle, but the estimated linear system can contain noise that leads to a pathological solution of $\tilde{\boldsymbol{S}}^{-1}\tilde{\boldsymbol{f}}$. Therefore, extra care should be taken when inverting $\tilde{\boldsymbol{S}}$. In addition to diagonal shift, it has been proposed to truncate the near-zero eigenvalues before the pseudoinverse, either with a fixed threshold or using the signal-to-noise ratio (SNR) of $\tilde{\boldsymbol{S}}$ [130, 253]. Alternatively, Ref. [254] has proposed a running estimation of $\tilde{\boldsymbol{S}}^{-1}$ over multiple optimization steps, which increases the effective sample size compared to inverting it in each step.

It has been shown that SR is necessary to obtain correct solutions to non-stoquastic systems in many cases [255–257], and the simple SGD or its variants such as Adam fail to converge to the ground state. This success is attributed to the incorporation of the local geometry in SR, which smoothens the highly non-convex energy landscape. In the ideal case where the estimations of $\boldsymbol{f}$ and $\boldsymbol{S}$ are exact, the ansatz has sufficiently high expressiveness, and the learning rate $\gamma$ is sufficiently small, then SR is equivalent to the exact ITE and always converges to the ground state after sufficiently many steps, as long as the initial state is valid.

It is also commonly believed that SR should not be used together with optimizers employing adaptive learning rates, such as Adam, because they make SR deviate from the ideal convergence path of ITE. On the other hand, multistep solvers for ordinary differential equations (ODEs), such as the Heun and the Runge–Kutta schemes, have been proposed to replace

the single-step update in Eq. (7.26) [257]. They improve the order of approximation to ITE and are more commonly used in literature on dynamics [253], although the limited accuracy in estimating $\boldsymbol{S}$ still impedes their performance.

After the above introduction to the general framework of VMC, in the following we discuss some common choices of the variational ansatz for quantum spin-$\frac{1}{2}$ systems that are of interest in this thesis.

## 7.5 Variational ansatzes for quantum spin systems

### 7.5.1 Jastrow ansatz

The Jastrow ansatz [258] was originally proposed to model the interactions between particles depending on their relative distances. In the context of spin systems, it is a quadratic form of the input configuration, without the normalization [259]:

$$\ln \psi_{\mathrm{J}}(\boldsymbol{s}) = \frac{1}{2}\boldsymbol{s}^{\top}\boldsymbol{W}^{\mathrm{J}}\boldsymbol{s}, \tag{7.35}$$

where $\boldsymbol{W}^{\mathrm{J}}$ is an $N \times N$ matrix of tunable parameters. Despite its simplicity, it is capable of modeling two-spin correlations and providing some initial insights into the system. The symmetries of the system can be imposed in the structure of $\boldsymbol{W}^{\mathrm{J}}$.

### 7.5.2 Gutzwiller projected states

Quantum many-body systems can exhibit a plethora of collective and emergent behaviors beyond two-spin correlations, such as spinons and fractional excitations in spin liquids. Although we mainly focus on spin systems in this thesis, an approach to study these systems is to transform the Hamiltonian of spins to an equivalent Hamiltonian of fermions, then it is possible to construct highly entangled states using the anti-commutation relations of fermions, which produce the necessary many-body correlations to model the aforementioned intricate behaviors, while can be evaluated in polynomial time. These introduced fermions are known as partons, namely the Abrikosov fermions [260] in the following case. The transformation is given by

$$\hat{\boldsymbol{S}}_i = \frac{1}{2}\sum_{\alpha\beta} \hat{c}^{\dagger}_{i\alpha}\hat{\boldsymbol{\sigma}}_{\alpha\beta}\hat{c}_{i\beta}, \tag{7.36}$$

where $\hat{\boldsymbol{S}}_i = \left(\hat{S}^x_i, \hat{S}^y_i, \hat{S}^z_i\right)$ is a vector of the spin operators at the site $i$, $\hat{\boldsymbol{\sigma}} = \left(\hat{\sigma}^x_i, \hat{\sigma}^y_i, \hat{\sigma}^z_i\right)$ is a vector of the Pauli matrices, while $\hat{c}^{\dagger}_{i\alpha}$ and $\hat{c}_{i\beta}$ are the fermionic creation and annihilation operators respectively at the site $i$. The indices $\alpha$ and $\beta$ can take the values $\uparrow$ and $\downarrow$ for the fermions, and are also used as the indices of the Pauli matrices. The valid states of the original spin Hamiltonian pose the constraint $\langle \hat{n}_i \rangle = 1$ for all sites $i$, where $\hat{n}_i = \sum_{\alpha} \hat{c}^{\dagger}_{i\alpha}\hat{c}_{i\alpha}$ is the number operator. To satisfy this constraint, the Gutzwiller projector [261]

$$\hat{P}_{\mathrm{G}} = \prod_i \hat{n}_i(2 - \hat{n}_i) \tag{7.37}$$

is applied to any variational state for the fermionic Hamiltonian, and this kind of variational states are commonly referred to as the Gutzwiller projected states.

A variety of variational ansatzes originally developed for fermionic systems can be applied to spin systems in this approach. To start with, the Slater determinant [22] is a general description of the ground state of non-interacting fermionic Hamiltonians, which can be represented using the creation operators, without the normalization [262]:

$$\left|\psi_{\mathrm{S}}\right\rangle = \prod_{i=1}^{2N}\left(\sum_{j=1}^{2N} W_{ij}^{\mathrm{S}}\hat{c}_j^{\dagger}\right)|0\rangle\,, \tag{7.38}$$

where $|0\rangle$ is the fermionic vacuum state, and $\boldsymbol{W}^{\mathrm{S}}$ is a $2N\times 2N$ matrix of variational parameters. The indices $i$ and $j$ run over both the spin value and the site, i.e., $i=(1\uparrow),\ldots,(N\uparrow),(1\downarrow),\ldots,(N,\downarrow)$. In VMC, we need to sample the fermionic configuration $\boldsymbol{n}=(n_{1\uparrow},\ldots,n_{N\uparrow},n_{1\downarrow},\ldots,n_{N\downarrow})$, and evaluate the wave function values of the samples. Following the anti-commutation relations of fermions, the wave function value is given by a determinant:

$$\left\langle \boldsymbol{n}\middle|\psi_{\mathrm{S}}\right\rangle = \langle 0|\,\hat{c}_{x_N}\cdots\hat{c}_{x_1}\left(\sum_p(-1)^p\prod_{k=1}^{N}W_{x_k,x_{p(k)}}^{\mathrm{S}}\right)\hat{c}_{x_1}^{\dagger}\cdots\hat{c}_{x_N}^{\dagger}\,|0\rangle \tag{7.39}$$

$$=\det\begin{pmatrix} W_{x_1x_1}^{\mathrm{S}} & \cdots & W_{x_1x_N}^{\mathrm{S}}\\ \vdots & \ddots & \vdots\\ W_{x_Nx_1}^{\mathrm{S}} & \cdots & W_{x_Nx_N}^{\mathrm{S}}\end{pmatrix}, \tag{7.40}$$

where $\boldsymbol{x}=(x_1,\ldots,x_N)$ is a vector of indices where a fermion exists, and there must be $N$ fermions in the $2N$ possible values of $i$, because of the constraint $\langle\hat{n}_i\rangle=1$. The permutation $p$ runs over all possible permutations of $(1,\ldots,N)$. Before VMC optimization, the value of $\boldsymbol{W}^{\mathrm{S}}$ can be initialized as the unitary matrix that diagonalizes the quadratic part of the Hamiltonian, and it can still be a sensible initial value in some cases even if the Hamiltonian contains interactions between sites [263–265].

For Hamiltonians with electron pairing interactions as in the Bardeen–Cooper–Schrieffer (BCS) theory, a more precise description is the pair-product (PP) state, also known as the geminal wave function [23, 266–268]:

$$\left|\psi_{\mathrm{PP}}\right\rangle = \left(\sum_{i,j=1}^{2N} W_{ij}^{\mathrm{PP}}\hat{c}_i^{\dagger}\hat{c}_j^{\dagger}\right)^{\frac{N}{2}}|0\rangle\,, \tag{7.41}$$

where $\boldsymbol{W}^{\mathrm{PP}}$ is a $2N\times 2N$ matrix of variational parameters. The wave function value is given by a Pfaffian [269], which is a property similar to the determinant but defined specifically for skew-symmetric matrices. Both the determinant and the Pfaffian can be evaluated in $O(N^3)$ time. When we only need the ratio of two probabilities, e.g., for the MCMC sampling in Eq. (3.7), it can be evaluated more efficiently without the whole determinant or Pfaffian [270].

Based on the above ansatzes, which are constructed from the analytical descriptions of the ground states of simple Hamiltonians, we can multiply more terms on them, commonly

referred to as correlators, with more variational parameters to improve their expressiveness. A simple example is the Jastrow correlator:

$$\hat{C}_{\mathrm{J}} = \exp\left(\frac{1}{2}\sum_{ij} W^{\mathrm{J}}_{ij}\hat{S}_i\hat{S}_j\right), \tag{7.42}$$

which is used in the well-known Slater–Jastrow ansatz:

$$\left|\psi_{\mathrm{SJ}}\right\rangle = \hat{C}_{\mathrm{J}}\left|\psi_{\mathrm{S}}\right\rangle. \tag{7.43}$$

The restricted Boltzmann machine (RBM) to be discussed in Section 7.5.3 can also be used as a correlator, which gives higher accuracy than the traditional Slater–Jastrow ansatz in modern VMC computations, especially for supposed spin liquids with the presence of disorder [271–273]. We refer to Ref. [270] as a self-contained introduction to VMC on fermionic systems, and to Ref. [274] for details of applying this approach on spin systems.

### 7.5.3 Neural quantum states (NQSs)

The neural networks discussed in Section 4.6 have also been proposed as variational ansatzes for quantum many-body systems, commonly known as neural quantum states (NQSs). A simple yet effective architecture is the restricted Boltzmann machine (RBM), which is essentially a two-layer neural network [255]:

$$\ln\psi_{\mathrm{RBM}}(\boldsymbol{s}) = \sum_i x_i + \sum_j a_j s_j, \tag{7.44}$$

$$x_i = \ln 2\cosh\left(b_i + \sum_j W_{ij}s_j\right). \tag{7.45}$$

Its variational parameters include $\boldsymbol{W}, \boldsymbol{a}, \boldsymbol{b}$, where $\boldsymbol{W}$ is an $\alpha N \times N$ matrix, $\alpha$ is a hyperparameter controlling the size of the network, $\boldsymbol{a}$ is a vector known as the visible bias, and $\boldsymbol{b}$ is another vector known as the hidden bias. When $\alpha = 1$, the RBM has approximately the same number of parameters as the Jastrow ansatz and achieves a higher accuracy than the latter on many physical systems [5]. It has a deep physical background rooted in the statistical properties of disordered spin systems [24, 60], and its simple architecture has motivated studies to interpret the learned parameters [275, 276]. Despite its simplicity, it is still continuously used in recent VMC computations [271, 273].

A growing variety of NQS architectures have emerged in recent literature, including deeper multilayer architectures to improve the expressiveness [277, 278], as well as convolutional neural networks (CNNs) [256, 279] and group equivariant neural networks [176, 177] to implement symmetries of physical systems. In particular, Ref. [280] has proposed an architecture named ClebschTree to incorporate the non-abelian SU(2) symmetry into NQS. Meanwhile, autoregressive neural networks (ARNNs) [281] and recurrent neural networks (RNNs) [282, 283] have been used to efficiently approximate complicated many-body distributions while supporting exact sampling, as discussed in Chapter 5. The advancement in language modeling

has also motivated the use of the transformer architecture in NQS [176, 284].

A recent breakthrough has been made by the vision transformer (ViT) [285–287], which greatly improves the results on the long-standing difficult problem of learning the sign structures in disordered systems, as we will discuss in Section 7.6. Compared to the original ViT in vision-language modeling, the ViT used as NQS does not employ the autoregressive architecture, and many non-linear activation functions are removed [288]. Its success can be attributed to the linear transformations arranged in a hierarchical scheme.

## 7.6 Amplitude and phase of quantum ansatz

In the above discussion on quantum ansatzes, we did not emphasize that they are complex valued. In fact, we tend to avoid the intricacy of complex-valued functions aforementioned in Section 7.3, especially when working with deep neural networks. A common practice is to parameterize the amplitude and the phase[3] of the wave function ansatz respectively using real-valued functions:

$$\psi(\boldsymbol{s}) = \sqrt{q(\boldsymbol{s})}\, \mathrm{e}^{\mathrm{i}\phi(\boldsymbol{s})}, \tag{7.46}$$

where $q(\boldsymbol{s})$ can be interpreted as a probability distribution. It is more common to separate the amplitude and the phase, rather than the real and the imaginary parts, because we frequently work with the log-gradient in Eq. (7.11), and the ratio of two probabilities in MCMC sampling as in Eq. (3.7). When sampling the configurations $\boldsymbol{s}$, we only need to evaluate $q(\boldsymbol{s})$, and not $\phi(\boldsymbol{s})$. In particular, when parameterizing $q(\boldsymbol{s})$ using an AR model as in Eq. (5.1), there is no need for $\phi(\boldsymbol{s})$ to also fulfill the AR property.

This separation of the amplitude and the phase has been applied in a lot of literature on NQS [268, 282, 289]. In particular, Ref. [257] provides empirical evidence that an ansatz with this separation achieves higher accuracy than the corresponding ansatz with the complex parameterization, and Ref. [290] even shows that the complex parameterization is outperformed by only optimizing the amplitude while fixing the phase according to the Marshall sign rule. On the other hand, there is also a lot of literature using the complex parameterization to achieve high accuracies on various systems [283, 285, 291], which typically use deeper networks and more parameters than the formers. Moreover, it remains an open question whether there is benefit in sharing a part of the parameters between the amplitude and the phase networks.

Although optimizing the amplitude can already be a difficult problem [257, 292], optimizing the phase for non-stoquastic Hamiltonians is more intricate [293, 294]. The ground state $\psi_0(\boldsymbol{s})$ must be real if the Hamiltonian is Hermitian, but its sign can have a complicated dependence on $\boldsymbol{s}$, which poses challenge to the generalization ability of the neural network, i.e., predicting $\psi_0(\boldsymbol{s})$ of unseen configurations from the sampled configurations in training, even if the sampling is unbiased. This intricacy also comes from the fact that the signs have to be optimized either in the complex plane, or using discrete and combinatorial optimization methods beyond gradient descent (GD).

[3] In this section we refer to the phase of complex number, not the phase of matter.

It has been practically shown in many cases that applying the Marshall sign rule is necessary for obtaining the correct sign structure [255–257, 291], and it has been believed that it is more difficult for deeper neural networks to obtain the sign structure [294]. Only with the recent breakthrough of ViT, it has been achieved to learn the sign structures in disordered systems by GD without manually applying the Marshall sign rule [285, 286].

To ease the optimization of the sign by GD, Ref. [294] proposes a phasor sum as the last layer of the phase network $\phi(\boldsymbol{s})$:

$$\phi(\boldsymbol{s}) = \arg \sum_i \exp\left(b + \sum_i w_i \phi_i\right), \tag{7.47}$$

where $\{\phi_i\}$ are the outputs of the previous layer, $\{w_i\}$ and $b$ are trainable parameters, and in numerical experiments it outperforms the usual sum

$$\phi(\boldsymbol{s}) = b + \sum_i w_i \phi_i. \tag{7.48}$$

It is also helpful to train the phase and the amplitude separately in a two-stage training procedure [294], or alternatively train them for more stages in an "alternating learning" procedure [268]. To obtain the sign structure in the disordered regime, we can use the known sign structure from the neighboring regime as the initialization in an "hysteresis optimization" procedure [268], or fine-tune only a part of the network after such initialization [203].

Beyond GD optimization in the usual setting of VMC, the signs can also be optimized in the setting of supervised learning, which uses a cross-entropy loss with known results [292], e.g., from exact diagonalization (ED) or highly accurate QMC on small systems, and then generalize to larger systems. It remains an open question whether this method can be used in unsupervised learning, possibly with modern estimators to relax the discrete signs into continuous and trainable variables [295–298]. Alternatively, the signs can be obtained by combinatorial optimization [299] or evolutionary algorithm [300].

Although these methods have successful results on many physical systems, the understanding is still limited on the relation between the effectiveness of these methods and the properties of the physical system as well as the neural network. We also remark that even if we know the true sign structure, the sign problem aforementioned in Section 6.4 can still occur in the difficulty of reliably estimating the energy in Eq. (7.4) with a highly oscillating $E_{\text{loc}}(\boldsymbol{s})$.

# 8 Tensor networks (TNs)

Quantum many-body systems are naturally represented in spaces of high-rank tensors, and the demand to decompose them into smaller tensors has arisen in studies of their entanglement properties [301]. The diagrammatic notation of tensors, namely Penrose graphical notations [302], have greatly eased the construction and reasoning of complicated contractions with multiple tensors, known as tensor networks (TNs) [182, 303]. Their use has been widely promoted in vast fields of physics besides condensed matters, including quantum computing [304, 305], high-energy physics [306, 307], and quantum gravity [308–311].

TNs can also be used as variational ansatzes, which enable exact and efficient summation over exponentially many configurations, instead of Monte Carlo sampling. Compared to neural networks, TNs are usually more controllable and systematically improvable, in the sense that they have few hyperparameters prone to manual tuning, and by tuning these hyperparameters, we can continuously improve the accuracy of variational approximation at the cost of more computation, which may allow extrapolating the variational results to the limit of infinite expressiveness.

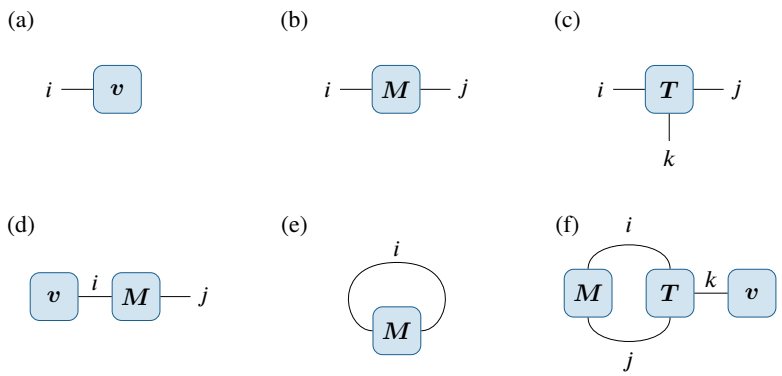



Figure 8.1: Examples of tensor diagram notations. (a) A vector $\boldsymbol{v}$ with an index $i$. (b) A matrix $\boldsymbol{M}$ with two indices $i, j$. (c) A tensor $\boldsymbol{T}$ with three indices $i, j, k$. (d) The matrix-vector multiplication $\sum_i v_i M_{ij}$, where $j$ is left as a free index. (e) The matrix trace $\sum_i M_{ii}$. (f) The tensor contraction $\sum_{i,j,k} T_{ijk} M_{ij} v_k$.

## 8.1 Tensor diagram notations

For the purpose of this thesis, we think of tensors as multidimensional arrays, and we do not emphasize their invariance under symmetry operations. In diagrammatic notation, each tensor is a node in the diagram, and its indices are denoted by "legs" stretching out from the node[1]. The contraction between two indices is indicated by connecting the two legs. Some examples of tensor diagrams are shown in Fig. 8.1.

In principle, when the tensor is not symmetric in any two indices, we need to specify which of its legs corresponds to which index, e.g., whether Fig. 8.1 (f) represents $\sum_{i,j,k} T_{ijk} M_{ij} v_k$ or $\sum_{i,j,k} T_{ijk} M_{ji} v_k$. But we usually omit this specification for brevity when it can be inferred from the context. In addition, we omit the names of the indices when not needed.

## 8.2 Matrix product state (MPS)

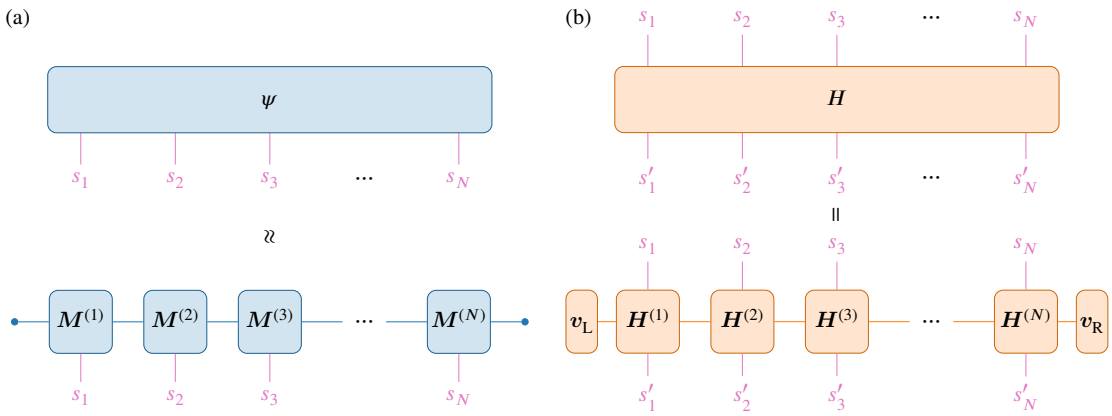



Figure 8.2: (a) A quantum many-body wave function $\psi(s_1, \ldots, s_N)$ approximated by a matrix product state (MPS). The pink legs can take spin values, the blue legs have the MPS bond dimension $\chi$, and the solid dot denotes summing over an index.
(b) A Hamiltonian $H(s_1, \ldots, s_N; s'_1, \ldots, s'_N)$ represented by a matrix product operator (MPO). The orange legs have the MPO bond dimension $\chi_H$.

For a quantum many-body system of $N$ spin-$\frac{1}{2}$ particles, the wave function $\psi(\boldsymbol{s})$ lives in a Hilbert space of $2^N$ dimension, as defined in Section 2.2. We can view $\psi(\boldsymbol{s})$ as a tensor with $N$ indices $s_1, s_2, \ldots, s_N$, and each index can take the spin values, which we denote by $\{\uparrow, \downarrow\}$, and also represent as $\{0, 1\}$ when using 0-based numeric array indexing. This tensor is visualized at the top of Fig. 8.2 (a).

The matrix product state (MPS) is a TN ansatz to approximately decompose $\psi(\boldsymbol{s})$ into a product of rank-3 tensors[2], also known as the tensor train decomposition [312]. It is formally

[1] Some literature distinguishes between covariant and contravariant indices, but in this thesis we omit that and allow contracting any two indices as long as they are defined in the same space.

[2] Although they are rank-3 tensors, for historical reasons the ansatz is still named "matrix" product state.

defined by

$$\psi(\boldsymbol{s}) = \sum_{a_0,\ldots,a_N=1}^{\chi} \prod_{i=1}^{N} M^{(i)}_{s_i;a_i,a_{i-1}}, \tag{8.1}$$

and visualized at the bottom of Fig. 8.2 (a). Here, $\chi$ is called the bond dimension, which is the most important hyperparameter of the MPS to control its overall size and expressiveness, and each bond index $a_i$ runs in $1,\ldots,\chi$. Different sites can have different $\chi$, as we will discuss in Section 8.5, but we simply use a typical value of $\chi$ when analyzing the computational complexity of MPS. Therefore, the MPS contains $2N\chi^2$ variational parameters, which is an exponential reduction from the $2^N$-dimensional Hilbert space.

Note that we have imposed the open boundary conditions (OBC) in the MPS, where the indices $a_0$ and $a_N$ at the two ends are simply summed over, which are equivalent to contracting with two vectors of ones, denoted by the solid dots in Fig. 8.2. These summations also make some parameters in $\boldsymbol{M}^{(1)}$ and $\boldsymbol{M}^{(N)}$ redundant. The OBC and the finite system size rule out translational symmetry, so we do not share parameters $\boldsymbol{M}^{(i)}$ between different sites $i$. There are also methods to study the limit of infinite system size, where $\boldsymbol{M}^{(i)}$ obeys the translational symmetry [313].

## 8.3 Contraction of MPS

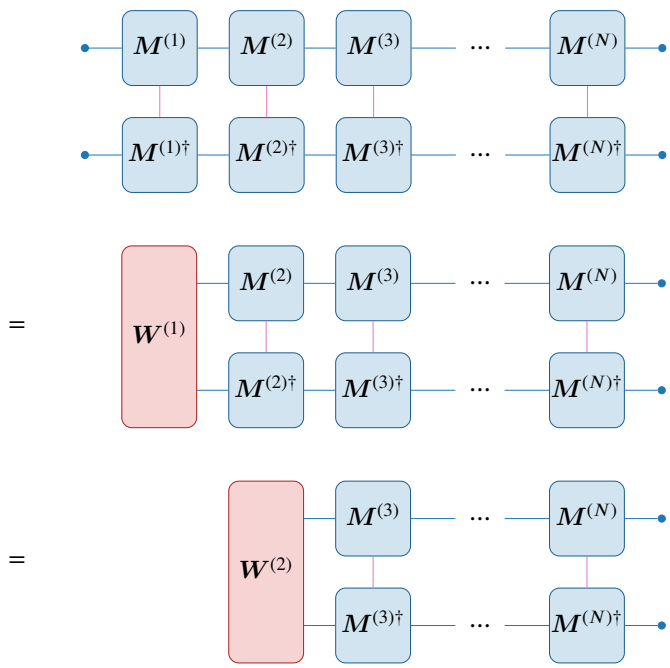



Figure 8.3: Sequential contraction of an MPS and its conjugate, where $\boldsymbol{W}^{(i)}$ is the intermediate result after the $i$-th step. In each step, we contract $\boldsymbol{W}^{(i-1)}$ and $\boldsymbol{M}^{(i)}$, then contract the result with $\boldsymbol{M}^{(i)\dagger}$ to obtain $\boldsymbol{W}^{(i)}$, which takes $O(\chi^3)$ time and $O(\chi^2)$ space.

In addition to the parameter efficiency, another advantage of MPS is that we can exactly and efficiently compute the normalization constant $\langle\psi|\psi\rangle$ by contracting the MPS with the

conjugate of itself. As shown in Fig. 8.3, this contraction is performed site-by-site. In each step, we contract the intermediate result $\boldsymbol{W}^{(i-1)}$ and a tensor $\boldsymbol{M}^{(i)}$ to obtain

$$V^{(i)}_{s;a_i,a'_{i-1}} = \sum_{a_{i-1}} M^{(i)}_{s;a_i,a_{i-1}} W^{(i-1)}_{a_{i-1},a'_{i-1}}, \tag{8.2}$$

which takes $O(\chi^3)$ time. Then we contract $\boldsymbol{V}^{(i)}$ and $\boldsymbol{M}^{(i)\dagger}$ to obtain

$$W^{(i)}_{a_i,a'_i} = \sum_{s,a'_{i-1}} M^{(i)\dagger}_{s;a'_i,a'_{i-1}} V^{(i)}_{s;a_i,a'_{i-1}}, \tag{8.3}$$

which also takes $O(\chi^3)$ time. Therefore, the contraction of $N$ sites takes $O(N\chi^3)$ time, which is also an exponential reduction from naively summing over the $2^N$ configurations. In addition, we can see that if we impose periodic boundary conditions (PBC) rather than OBC, i.e., connect the solid dots at the two ends of the MPS, then the intermediate result $\boldsymbol{W}^{(i-1)}$ has to maintain four indices of size $\chi$, therefore the time complexity to contract the above TN increases to $O(N\chi^4)$, while the dynamic space required by the intermediate result increases even more rapidly from $O(\chi^2)$ to $O(\chi^4)$.

It is worth noting that one may normalize and canonize the MPS, such that

$$\sum_{s,a_{i-1}} M^{(i)\dagger}_{s;a'_i,a_{i-1}} M^{(i)}_{s;a_i,a_{i-1}} = \delta_{a_i,a'_i}, \tag{8.4}$$

which simplifies the above contraction and avoids the numerical problems of exponentially exploding or vanishing intermediate results. However, when used in VMC as we will discuss in Chapter 9, this is usually only needed when initializing the parameters, rather than in each VMC step.

## 8.4 Matrix product operator (MPO)

When using MPS to study the ground state of a Hamiltonian, we need to decompose the latter into a form similar to the former, namely the matrix product operator (MPO), as visualized in Fig. 8.2 (b). The MPO also has a bond dimension $\chi_H$, and we usually set $\chi \gg \chi_H$ in large-scale computations. Unlike MPS, the boundary conditions of MPO are not simply imposed by summing over the indices, but contracting with specific vectors depending on the Hamiltonian, as indicated by $\boldsymbol{v}_{\mathrm{L}}$ and $\boldsymbol{v}_{\mathrm{R}}$ in Fig. 8.2 (b).

In the same manner discussed in Section 8.3, the expectation of the Hamiltonian $\langle\psi|\hat{H}|\psi\rangle$ without normalization can be exactly contracted in $O(\chi^3)$ time and $O(\chi^2)$ space. Other observables besides the Hamiltonian can also be measured in this manner, which is especially efficient if the operator acts only on one or a few sites.

Algorithms have been proposed to exactly or approximately construct an MPO from a locally interacting operator [314–316]. As a concrete example, the MPO exactly equivalent to

the Hamiltonian of the transverse-field Ising model in Eq. (2.24) is given by

$$\boldsymbol{H}^{(i)} = \begin{pmatrix} \hat{I} & 0 & 0 \\ \hat{\sigma}^z & 0 & 0 \\ \Gamma\hat{\sigma}^x & J\hat{\sigma}^z & \hat{I} \end{pmatrix}, \tag{8.5}$$

$$\boldsymbol{v}_{\mathrm{L}} = \begin{pmatrix} 0 & 0 & \hat{I} \end{pmatrix}, \tag{8.6}$$

$$\boldsymbol{v}_{\mathrm{R}} = \begin{pmatrix} \hat{I} & 0 & 0 \end{pmatrix}^{\top}, \tag{8.7}$$

where the matrix and the vectors have bond indices, while the operators have spin indices. Substitute this definition into Fig. 8.2 (b) and contract the TN, then the original Hamiltonian is recovered.

Before modern numerical approaches were possible, MPS with small $\chi$ has been successfully applied in analytical solutions to ground states of many one-dimensional systems, including the renowned Affleck–Kennedy–Lieb–Tasaki (AKLT) model of the spin-1 Heisenberg chain [317–319], and the Majumdar–Gosh model with the addition of next-nearest-neighbor frustrated interaction [320]. It has been proven that if we cut an MPS into two parts, then the entanglement entropy $S$ between them scales at most by $S = O(\ln\chi)$, therefore MPS is suitable for representing states with low entanglement [321]. Moreover, the spatial correlation of any single-site operator

$$C(d) = \frac{1}{N-d}\sum_{i=1}^{N-d}\langle\hat{O}_i\hat{O}_{i+d}\rangle - \langle\hat{O}_i\rangle\langle\hat{O}_{i+d}\rangle \tag{8.8}$$

decays exponentially: $C(d) \sim \mathrm{e}^{-\frac{d}{\xi}}$, where $\xi$ is the correlation length. This is exactly the case for the ground states of 1D gapped systems, so it is possible for MPS to exactly represent them.

## 8.5 Density matrix renormalization group (DMRG)

Along with the analytical studies, the numerical approach to optimize MPS for variational approximation of ground states developed at the same time. As the name suggests, the density matrix renormalization group (DMRG) algorithm [301, 322] originated from the theory of renormalization group, but for the purpose of this thesis we can simply think of it as an iterative series of two-site updates.

In each step, we choose a site $\boldsymbol{M}^{(i)}$ and its neighbor $\boldsymbol{M}^{(i+1)}$, and contract $\langle\psi|\hat{H}|\psi\rangle$ and $\langle\psi|\psi\rangle$ respectively, apart from the two sites to be updated, and obtain the environment tensors $\boldsymbol{H}_{\mathrm{env}}$ and $\boldsymbol{I}_{\mathrm{env}}$, as shown in Fig. 8.4 (a) and (b). Then we view $\langle\psi|\hat{H}|\psi\rangle$ and $\langle\psi|\psi\rangle$ as $4\chi^2 \times 4\chi^2$ matrices, and solve the optimization problem

$$\arg\min_{\boldsymbol{x}} \frac{\boldsymbol{x}^{\dagger}\boldsymbol{H}_{\mathrm{env}}\boldsymbol{x}}{\boldsymbol{x}^{\dagger}\boldsymbol{I}_{\mathrm{env}}\boldsymbol{x}}, \tag{8.9}$$

which is equivalent to the generalized eigenvector problem

$$\boldsymbol{H}_{\mathrm{env}}\boldsymbol{x} = \lambda\boldsymbol{I}_{\mathrm{env}}\boldsymbol{x} \tag{8.10}$$

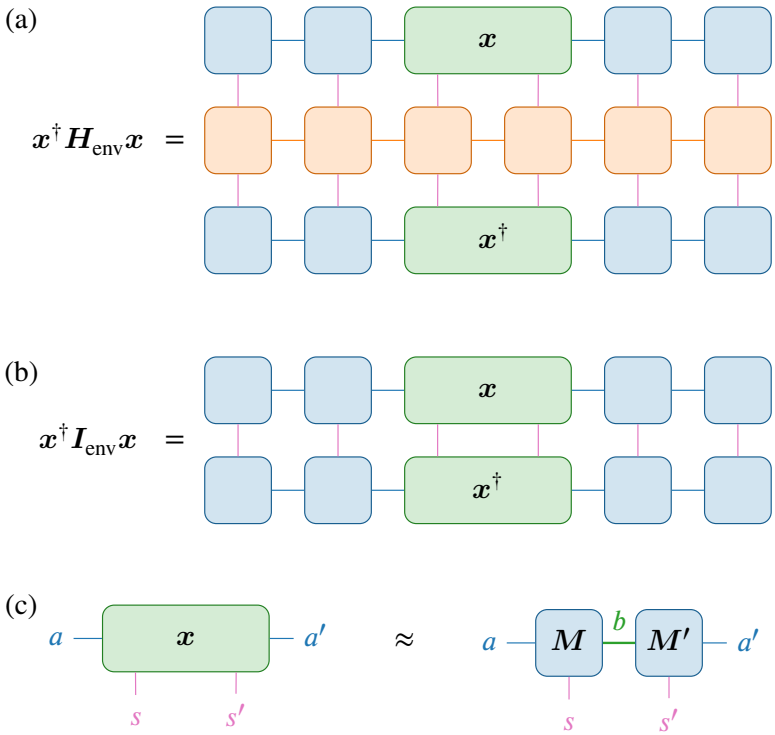



Figure 8.4: Tensor diagrams used in DMRG. We show six sites as an example, and omit the vectors on the boundaries for clarity.
(a) Obtaining the environment tensor $\boldsymbol{H}_{\text{env}}$ by contracting $\langle\psi|\hat{H}|\psi\rangle$ apart from the two sites $\boldsymbol{x}$ to be updated.
(b) Obtaining the environment tensor $\boldsymbol{I}_{\text{env}}$ by contracting $\langle\psi|\psi\rangle$ apart from $\boldsymbol{x}$.
(c) Decomposing the updated $\boldsymbol{x}$ into two sites, where the thick green index $b$ can have a dimension $\chi'$ different from the original $\chi$.

with the smallest eigenvalue $\lambda$, and can be solved in $O(\chi^4)$ time and space.

After obtaining $\boldsymbol{x}$, it needs to be decomposed into the updated sites $\boldsymbol{M}^{(i)}$ and $\boldsymbol{M}^{(i+1)}$, as shown in Fig. 8.4 (c). We can view $\boldsymbol{x}$ as a $2\chi \times 2\chi$ matrix, whose rank is at most $2\chi$, so in general the dimension of the index $b$ after the decomposition can be as large as $2\chi$. If we want $\boldsymbol{M}^{(i)}$ and $\boldsymbol{M}^{(i+1)}$ to have the same size as before, we can use singular value decomposition (SVD) to obtain the best approximation of them, where we only keep the first $\chi$ singular values. Alternatively, we can control the new bond dimension $\chi'$ between $\boldsymbol{M}^{(i)}$ and $\boldsymbol{M}^{(i+1)}$ by discarding singular values below a threshold. Therefore, the size of the ansatz is dynamically adapted during the DMRG procedure, which depends on the amount of entanglement in the physical system.

The above step of DMRG can only update the parameters of two sites. We usually apply $N-1$ steps on all sites sequentially and call it a "sweep". After sufficiently many sweeps, the parameters of all sites will converge to the best approximation. This algorithm is also known as alternating least squares (ALS) in other fields of optimization problems [323, 324]. As a second-order optimization algorithm, it can have a faster convergence speed than the gradient

descent (GD) optimizer, and avoid some local minima and saddle points that the latter is prone to. The extensions to second-order optimizers discussed in Section 4.2.2 can also be applied to DMRG, including the damping to improve stability and the low-rank updates to reduce memory usage.

It should be noted that the above MPS also supports evaluating $\psi(\boldsymbol{s})$ given a configuration $\boldsymbol{s}$, i.e., specifying a spin value at each spin index and contracting the remaining TN, which takes $O(\chi^3)$ time and $O(\chi^2)$ space. In principle, MPS can be optimized by GD in the framework of VMC, instead of DMRG, which is especially feasible when PBC are imposed and the computational complexity of DMRG is polynomially higher [325] than in the OBC case. The gradients through tensor contractions and SVD truncations can be obtained by automatic differentiation [326, 327]. Moreover, MPS supports exact sampling in the autoregressive form [328–330], which we will discuss in more depth in Chapter 9.

In addition to DMRG, another optimization algorithm that arises from the physical properties of MPS is the time-evolving block decimation (TEBD) algorithm [331], which decomposes the Hamiltonian into one- or two-site blocks and applies them to the MPS sequentially. It can be more efficient than DMRG for systems with PBC or higher-dimensional geometries, as it avoids contracting the environment tensors [332].

## 8.6 TNs with higher-dimensional geometries

In the previous sections, we have seen that MPS can be analytical or numerically exact solutions to the ground states of many 1D and quasi-1D systems, and it only requires polynomial time and space to exactly evaluate observables. However, it is no longer as efficient for systems on 2D or higher-dimensional geometries.

The applicability of MPS can be examined by the area law of entanglement entropy [183, 333–335]: For gapped and locally interacting systems, when we cut out a region from it, the entanglement entropy $S$ between the region and the environment is asymptotically proportional to the area of the boundary, rather than the volume of the region. For 1D systems, the boundary consists of only two sites, so we have $S = O(1)$, which can be captured by an MPS if its bond dimension $\chi$ is large enough.

For a 2D system with $N$ sites, we have $S = O(L)$, where $L = O(N^{\frac{1}{2}})$ is the linear size of the system. When applying an MPS to it, we need to arrange it in a 1D order of the sites, and some possible choices are sketched in Fig. 8.5. Regardless of how we choose the order, we can always cut the MPS into two parts, i.e., divide the system into $\boldsymbol{s}_{\leq k} = \{s_i \mid 1 \leq i \leq k\}$ and $\boldsymbol{s}_{>k} = \{s_i \mid k < i \leq N\}$, where the index $i$ follows the order of the MPS, and the bond between the sites $k$ and $k+1$ is cut. Therefore, the entanglement entropy between $\boldsymbol{s}_{\leq k}$ and $\boldsymbol{s}_{>k}$ produced by the MPS is $S = O(\ln \chi)$, and we can obtain the expected area law only if the bond dimension $\chi = O(\mathrm{e}^L)$, which is considered intractable when $L$ is large.

The lack of entanglement in MPS also leads to the incapability of capturing long-range correlations. The correlations in MPS decay by $C(d) \sim \mathrm{e}^{-\frac{d}{\xi}}$, where $r$ is the distance in the 1D order. However, the 1D order breaks the locality of the 2D distance. For example, when we use the zigzag order as in Fig. 8.5 (a), every pair of vertically adjacent sites has spatial distance 1,

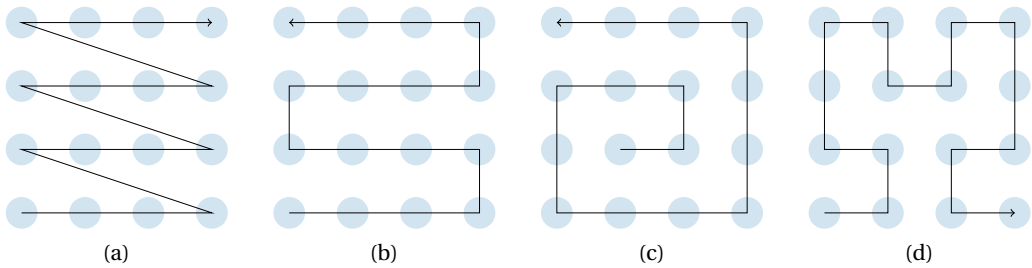



Figure 8.5: Different orders of MPS on 2D lattices, where a $4 \times 4$ lattice is shown as an example: (a) zigzag order, (b) snake order, (c) spiral order, (d) Hilbert space-filling curve [336].

but their distance in the 1D order is as large as $L$, making it difficult for the MPS to capture their correlation. One may attempt to arrange the MPS in a more complicated order with better locality, such as the Hilbert space-filling curve[3] [336] in Fig. 8.5 (d), but it has been proven that no 1D order is substantially better than the most naive zigzag order in preserving the distances of all pairs of sites on the 2D lattice [337].

These difficulties motivate us to propose ansatzes with 2D geometries natively for 2D systems, and a relatively simple and prevalent choice is the projected entangled-pair state (PEPS) [321, 338]. Although it originated from the representation of two-site entanglements, in the context of variational ansatzes, it usually refers to the most intuitive generalization of MPS to 2D, as shown in Fig. 8.6. This ansatz has $2N\chi^4$ variational parameters, which is polynomially more than MPS.

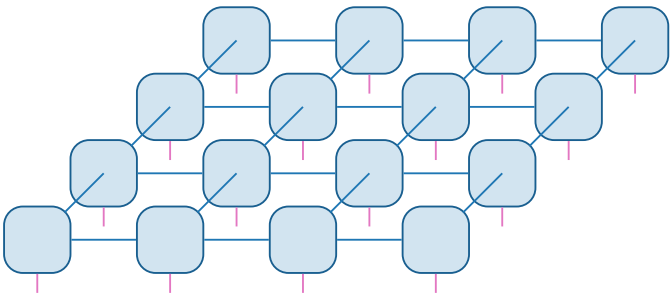


Figure 8.6: PEPS as the generalization of MPS to 2D, on a $4 \times 4$ lattice as an example. On each site, there is a tensor of size $2 \times \chi \times \chi \times \chi \times \chi$, which is contracted with four neighboring sites and has a free index of spin. The vectors on the boundaries are omitted for clarity.

When we cut out a region with a linear size of $O(L)$ from a PEPS, there are always $O(L)$ bonds across the boundary, and we can view the entire boundary as a thick bond of size $\chi^{O(L)}$. Therefore, the entanglement entropy produced by the MPS is $S = O(L \ln \chi)$, which can achieve the area law. It has also been shown by construction that PEPS can exactly represent some states with the area law, such as the resonating valence bond (RVB) state [339] to describe spin liquids and the toric code [12, 340] useful in quantum computing. Moreover, PEPS can

[3] Not related to the Hilbert space.

produce algebraically[4] decaying correlations [183], which is a property of 2D critical systems.

However, the main drawback of PEPS is that we cannot exactly contract $\langle\psi|\psi\rangle$ or evaluate $\psi(\boldsymbol{s})$ in polynomial time and space. It is impossible to sequentially contract a PEPS and keep a fixed number of free indices in the intermediate result, as in Fig. 8.3, while a growing number of free indices leads to exponentially growing time and space complexities. It has been proven that contracting a general PEPS belongs to the #P-hard class of computational complexity [341], which is considered intractable for large systems.

A practical approach is to approximately contract the PEPS and use SVD to truncate the intermediate result, such that the time and the space complexities remain polynomial. When optimizing a PEPS using DMRG, the environment tensors can be contracted in this approach, which has been successfully applied to systems with more than 1000 spins [342–344]. In addition to regular lattices, this approach can also contract more complicated TNs on unstructured graphs [345–347], which is useful in disordered systems with random interactions, as well as quantum circuits. Alternatively, TNs contracted in this approach can be optimized by GD in the framework of VMC [348–350].

The difficulty of contracting PEPS is caused by loops in the 2D grid, and other TNs with loop-free and tree-like architectures have been proposed [351–354]. Some of them can be exactly contracted, at the cost of reduced expressiveness. A renowned architecture is the multiscale entanglement renormalization ansatz (MERA) [355, 356], which can produce algebraically decaying correlations, but not the 2D area law, in polynomial computation time and space. Numerically, they have been optimized to higher accuracies than MPS ever achieved on 2D systems.

[4] Here “algebraic” means decaying in a power law, as opposed to exponentially decaying.

# 9 Tensor-RNN: bridge between tensor networks and neural networks

Comparing tensor network (TN) quantum states in Chapter 8 and neural quantum states (NQSs) in Section 7.5.3, we can see that TNs have a deeper background in physics, as their representations of the ground states of some important physical systems are found by construction, and their analytical properties such as entanglement entropy and spatial correlations are proven. They are also more controllable, in the sense that we can systematically improve the accuracy of the variational approximation by increasing their bond dimensions. Their relatively simple multilinear architectures are suitable for second-order optimizations, which have better convergence properties towards the ground state than the first-order optimizations usually used for neural networks. However, the 1D matrix product state (MPS) is not expressive enough for 2D and higher-dimensional systems, and the 2D projected entangled-pair state (PEPS) cannot be efficiently contracted.

In contrast, NQSs are proven to have the expressiveness of universal approximation, and they are practically successful in representing complicated ground states, such as spin liquids, with high accuracy and efficiency. However, their interpretability and controllability are worse than TNs. Their analytical properties, the relation between the accuracy of variational approximation and their hyperparameters, such as the numbers of layers and channels, as well as the optimization methods to fully exploit their expressiveness, are all under research.

Since the initial proposal of NQSs, there has been ongoing research on their ability to represent certain classes of quantum states [277, 278, 357]. Exact mappings have been constructed between TNs and NQSs, especially restricted Boltzmann machines (RBMs) [358, 359] with relatively simple architectures in NQSs. Meanwhile, mapping from TNs to recurrent neural networks (RNNs) has been studied from the viewpoint of recurrent arithmetic circuits [360, 361], suggesting that the extra expressiveness of RNNs over TNs can be attributed to their reuse of information between sites. There have also been attempts to incorporate multilinear contractions of TNs into NQSs in various ways, which can produce greater expressiveness than usual linear layers [72, 362, 363].

In the following, we present a series of NQS architectures, namely MPS-RNN and tensor-RNN [3], which combine the strengths of TNs and RNNs. They can achieve higher numerical accuracy than MPS and PEPS on 2D systems, show the desired analytical properties of entanglement entropy and spatial correlations, and get systematically improved by increasing the bond dimension, which is the only hyperparameter for their architectures. Moreover,

they support exact sampling and efficient evaluation, leading to the reduction of practical computation time compared to PEPS.

## 9.1 Vanilla MPS-RNN

We start from the aforementioned definition of an MPS with bond dimension $\chi$, on $N$ sites of spin-$\frac{1}{2}$ particles:

$$\psi(\boldsymbol{s}) = \sum_{a_0,\ldots,a_N=1}^{\chi} \prod_{i=1}^{N} M^{(i)}_{s_i;a_i,a_{i-1}}. \tag{8.1}$$

Note that we use 1-based indexing for the spatial and the bond dimensions, which is consistent with the notations in previous chapters, as opposed to the 0-based indexing in Ref. [3], which can be more convenient for software implementation. The many-body probability $q(\boldsymbol{s}) = |\psi(\boldsymbol{s})|^2$ can be factorized into autoregressive (AR) conditional probabilities [328–330]:

$$q(\boldsymbol{s}) = \prod_{i=1}^{N} q_i(s_i \mid \boldsymbol{s}_{<i}), \tag{5.1}$$

which are suitable for exact sampling. Each conditional probability is formally written as

$$q_i(s_i \mid \boldsymbol{s}_{<i}) = \frac{\tilde{q}_i(s_i \mid \boldsymbol{s}_{<i})}{\sum_s \tilde{q}_i(s \mid \boldsymbol{s}_{<i})}, \tag{9.1}$$

$$\tilde{q}_i(s_i \mid \boldsymbol{s}_{<i}) = \boldsymbol{h}^{(i)\dagger}(s_i, \boldsymbol{s}_{<i})\boldsymbol{\gamma}^{(i)}\boldsymbol{h}^{(i)}(s_i, \boldsymbol{s}_{<i}), \tag{9.2}$$

$$h^{(i)}_{a_i}(\boldsymbol{s}_{\leq i}) = \sum_{a_1,\ldots,a_{i-1}} \prod_{j=1}^{i} M^{(j)}_{s_j;a_j,a_{j-1}}, \tag{9.3}$$

$$\gamma^{(i)}_{b_i,a_i} = \sum_{\substack{a_{i+1},\ldots,a_V \\ b_{i+1},\ldots,b_V \\ s_{i+1},\ldots,s_V}} \prod_{j=i+1}^{V} M^{(j)*}_{s_j;b_j,b_{j-1}} M^{(j)}_{s_j;a_j,a_{j-1}}, \tag{9.4}$$

where $\boldsymbol{h}^{(i)}$ is a vector of size $\chi$, denoting the partially contracted sites up to the site $i$, and depending on the spins $\boldsymbol{s}_{\leq i}$, while $\boldsymbol{\gamma}^{(i)}$ is a $\chi \times \chi$ matrix, denoting the partially contracted sites and their conjugates after the site $i$.

The tensor diagram for the unnormalized conditional probability in Eq. (9.2) is shown in Fig. 9.1. Note that the solid dot with $r$ indices denotes the rank-$r$ Kronecker tensor

$$\delta_{i_1,\ldots,i_r} = \begin{cases} 1, & i_1 = \cdots = i_r \\ 0, & \text{otherwise} \end{cases}, \tag{9.5}$$

and when $r = 1$ it simply sums over the index. As edge cases, we have

$$h^{(0)}_a = 1 \quad \forall a, \tag{9.6}$$

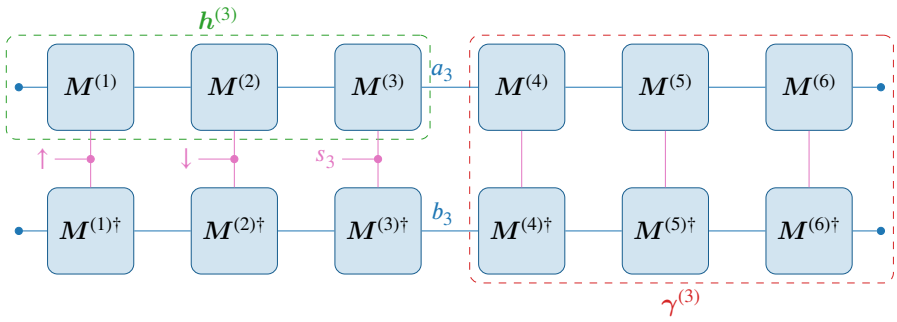



Figure 9.1: Tensor diagram for the unnormalized conditional probability $\tilde{q}_3(s_3 \mid s_1 = \uparrow, s_2 = \downarrow)$ as in Eq. (9.2). An MPS with six sites is shown as an example. This figure is reproduced from Fig. S1 in the supplemental material for Ref. [3].

which is just the top leftmost solid dot, and

$$\gamma^{(N)}_{b,a} = 1 \quad \forall a, b, \tag{9.7}$$

which is the direct product of the rightmost two solid dots.

The MPS can be optimized using VMC with AR sampling. At the beginning of each optimization step of VMC, we precompute $\boldsymbol{\gamma}^{(N-1)}, \ldots, \boldsymbol{\gamma}^{(1)}$ sequentially during a usual contraction of $\langle\psi|\psi\rangle$, which are independent of the input spins. Then in each AR sampling step $i$, we update $\boldsymbol{h}^{(i)}$ by

$$\boldsymbol{h}^{(i)}(s_i) = \boldsymbol{M}^{(i)}_{s_i} \boldsymbol{h}^{(i-1)}, \tag{9.8}$$

which can be interpreted as updating the hidden memory in an RNN as in Eq. (5.6). Note that the spin $s_i$ is a free variable at this point, while all previous spins $\boldsymbol{s}_{<i}$ that are input to $\boldsymbol{h}^{(i)}$ are already determined. Also, unlike conventional RNN, in the MPS we do not share parameters $\boldsymbol{M}^{(i)}$ between different sites $i$. After obtaining $\boldsymbol{h}^{(i)}$, we compute the unnormalized conditional probability

$$\tilde{q}_i(s_i \mid \boldsymbol{s}_{<i}) = \boldsymbol{h}^{(i)\dagger}(s_i) \boldsymbol{\gamma}^{(i)} \boldsymbol{h}^{(i)}(s_i), \tag{9.9}$$

and normalize it if needed, which can be interpreted as the output layer of the RNN as in Eq. (5.7). The tensor diagrams for this AR sampling step are shown in Fig. 9.2.

Therefore, the MPS is exactly mapped to an RNN architecture with memory size $\chi$, linear updates of the memory in Eq. (9.8), and a bilinear output layer in Eq. (9.9). To avoid precomputing $\{\boldsymbol{\gamma}^{(i)}\}$ from $\{\boldsymbol{M}^{(i)}\}$, and make a generalization from the MPS, we treat $\{\boldsymbol{\gamma}^{(i)}\}$ as variational parameters in addition to $\{\boldsymbol{M}^{(i)}\}$, where each $\{\boldsymbol{M}^{(i)}\}$ and $\{\boldsymbol{\gamma}^{(i)}\}$ is independent of each other. Therefore, the RNN architecture has greater expressiveness than the MPS, but not every trained RNN can be mapped back to an MPS. In the following, we refer to this architecture as the vanilla MPS-RNN.

Besides the probability $q(\boldsymbol{s})$, the complete definition of the wave function also includes the phase $\phi(\boldsymbol{s})$:

$$\psi(\boldsymbol{s}) = \sqrt{q(\boldsymbol{s})} e^{i\phi(\boldsymbol{s})}. \tag{7.46}$$

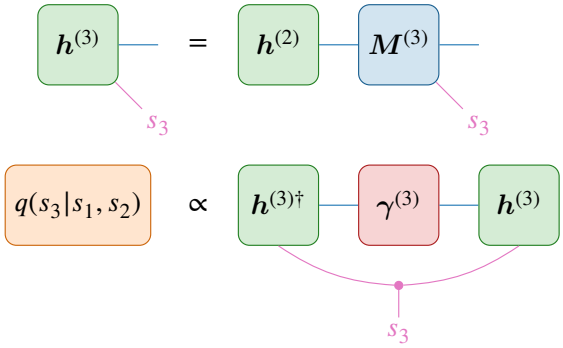



Figure 9.2: Tensor diagrams for updating the memory $\boldsymbol{h}^{(i)}$ as in Eq. (9.8), and computing the conditional probability $q_i(s_i \mid \boldsymbol{s}_{<i})$ as in Eq. (9.9), in an AR sampling step of MPS. This figure is reproduced from Fig. 1 (b) in Ref. [3].

After the AR sampling of the vanilla MPS-RNN, $\phi(\boldsymbol{s})$ is obtained by

$$\phi(\boldsymbol{s}) = \arg \sum_a h_a^{(N)}. \tag{9.10}$$

The computational graph for the vanilla MPS-RNN to evaluate $\psi(\boldsymbol{s})$ is shown in Fig. 9.3.

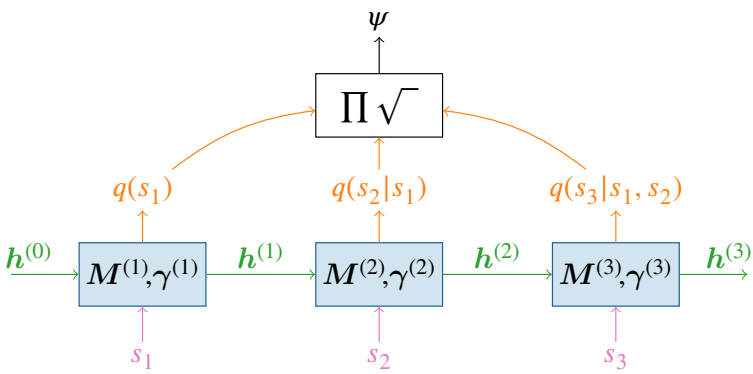



Figure 9.3: Computational graph for the vanilla MPS-RNN to evaluate the wave function $\psi(\boldsymbol{s})$, where three sites are shown as an example. Every site $i$ has the variational parameters $\boldsymbol{M}^{(i)}$ and $\boldsymbol{\gamma}^{(i)}$, takes the spin $s_i$ and the previous memory $\boldsymbol{h}^{(i-1)}$ as inputs, and outputs the conditional probability $q(s_i \mid \boldsymbol{s}_{<i})$ and the updated memory $\boldsymbol{h}^{(i)}$. The phase part is omitted for simplicity. This figure is reproduced from Fig. 1 (a) in Ref. [3].

The vanilla MPS-RNN takes $O(N\chi^2)$ time and space to sample the configuration $\boldsymbol{s}$ and evaluate the wave function $\psi(\boldsymbol{s})$. It is worth noting that MPS can exactly represent the ground states of 1D gapped systems, and every MPS can be exactly mapped to a vanilla MPS-RNN, so the non-linear activation functions in the update rules of conventional RNNs are unnecessary to represent these states.

## 9.2 1D MPS-RNN

We make a few modifications to the vanilla MPS-RNN to reduce its computation time and improve its training stability, while also slightly increasing its expressiveness. Each matrix $\boldsymbol{\gamma}^{(i)}$ in Eq. (9.4) is positive semidefinite, so we diagonalize it by

$$\boldsymbol{\gamma}^{(i)} = \boldsymbol{U}^{(i)} \operatorname{diag}\left(\boldsymbol{\eta}^{(i)}\right) \boldsymbol{U}^{(i)\dagger}, \tag{9.11}$$

where $\boldsymbol{U}^{(i)}$ is a unitary matrix, and $\boldsymbol{\eta}^{(i)}$ is a real and positive vector, which is further parameterized by the real vector $\ln \boldsymbol{\eta}^{(i)}$. In the following, we absorb $\boldsymbol{U}^{(i)}$ into the definition of $\boldsymbol{M}^{(i)}$, and also the resulting $\boldsymbol{h}^{(i)}$:

$$\boldsymbol{M}^{(i)} \leftarrow \boldsymbol{U}^{(i)} \boldsymbol{M}^{(i)} \boldsymbol{U}^{(i-1)\dagger}, \tag{9.12}$$

$$\boldsymbol{h}^{(i)} \leftarrow \boldsymbol{U}^{(i)} \boldsymbol{h}^{(i)}. \tag{9.13}$$

Therefore, the probability in Eq. (9.2) is simplified to

$$\tilde{q}_i(s_i \mid \boldsymbol{s}_{<i}) = \sum_s \eta_s^{(i)} \left| h_s^{(i)} \right|^2, \tag{9.14}$$

where the number of parameters and the computation time are reduced without loss of expressiveness.

Then we generalize the update rule Eq. (9.8) by adding a term $\boldsymbol{v}^{(i)}$, which is a vector of size $\chi$ and depends on the spin $s_i$, i.e., it is a $2 \times \chi$ tensor when we view $s_i$ as a free variable. Then the update rule becomes

$$\tilde{\boldsymbol{h}}^{(i)}(s_i) = \boldsymbol{M}_{s_i}^{(i)} \boldsymbol{h}^{(i-1)} + \boldsymbol{v}_{s_i}^{(i)}. \tag{9.15}$$

The expressiveness of the architecture with the term $\boldsymbol{v}^{(i)}$ and with bond dimension $\chi$ is not greater than that without the term $\boldsymbol{v}^{(i)}$ and with bond dimension $\chi + 1$, because $\boldsymbol{v}^{(i)}$ can be absorbed into $\boldsymbol{M}^{(i)}$ as an extra row. Moreover, we normalize the intermediate result after each AR sampling step:

$$\boldsymbol{h}^{(i)}(s_i) = \frac{\tilde{\boldsymbol{h}}^{(i)}(s_i)}{\sqrt{\sum_s \tilde{\boldsymbol{h}}^{(i)\dagger}(s) \tilde{\boldsymbol{h}}^{(i)}(s)}}, \tag{9.16}$$

which acts similarly to the sigmoid function in conventional RNN architectures to bound the intermediate result and avoid gradient explosion [364]. When analyzing a trained ansatz, we can always absorb the normalization into $\boldsymbol{M}^{(i)}$.

We also generalize the phase in Eq. (9.10) to a sum of "local phases":

$$\phi(\boldsymbol{s}) = \sum_i \phi^{(i)}(\boldsymbol{s}_{\leq i}), \tag{9.17}$$

$$\phi^{(i)}(\boldsymbol{s}_{\leq i}) = \arg\left(\boldsymbol{w}_{s_i}^{(i)\top} \boldsymbol{h}^{(i)}(\boldsymbol{s}_{\leq i}) + c_{s_i}^{(i)}\right), \tag{9.18}$$

where each local phase $\phi^{(i)}(\boldsymbol{s}_{\leq i})$ is obtained from a linear transformation of the memory $\boldsymbol{h}^{(i)}(\boldsymbol{s}_{\leq i})$, whose parameters $\boldsymbol{w}^{(i)}$ and $c^{(i)}$ depend on the spin $s_i$. Therefore, the gradient

information from every site, rather than only the last site, can more directly affect the phase during training. It should be noted that the phasor sum in Eq. (7.47) can be an alternative to the simple sum in Eq. (9.17).

The RNN architecture defined by the update rule in Eq. (9.15), the output layer in Eq. (9.14), and the phase in Eq. (9.18) is named 1D MPS-RNN. It is more suitable for practical computations than the vanilla one, in terms of computation time and numerical stability in training.

## 9.3 2D MPS-RNN

Next, we investigate the generalization of the MPS-RNN to 2D lattices, which should ideally capture the area law of entanglement entropy and the algebraically decaying correlations in the 2D systems of interest, while still supporting efficient evaluation in polynomial time and space, as well as AR sampling. An intuitive attempt is to include the memories from two previous neighboring sites as inputs for the update rule in Eq. (9.15):

$$\tilde{\boldsymbol{h}}^{(x,y)}(s_{x,y}) = \boldsymbol{M}^{(x,y)}_{\mathrm{y};s_{x,y}}\boldsymbol{h}^{(x,y-1)} + \boldsymbol{M}^{(x,y)}_{\mathrm{x};s_{x,y}}\boldsymbol{h}^{(x-1,y)} + \boldsymbol{v}^{(x,y)}_{s_{x,y}}, \tag{9.19}$$

where we use the 2D indexing $(x, y)$ instead of the 1D indexing $i$ for the sites. In all the results reported, we use the snake order as in Fig. 8.5 (b), and for simplicity we assume $y$ is odd in Eq. (9.19), so $x$ increases in the snake order. At each site, there are two matrices $\boldsymbol{M}^{(x,y)}_{\mathrm{x}}$ and $\boldsymbol{M}^{(x,y)}_{\mathrm{y}}$ to transform the two input memory vectors. The tensor diagram for this update rule is shown in Fig. 9.4. The RNN architecture defined by Eq. (9.19) and the previous Eqs. (9.14) and (9.18) is named 2D MPS-RNN, and its computational graph to evaluate the wave function is shown in Fig. 9.5.

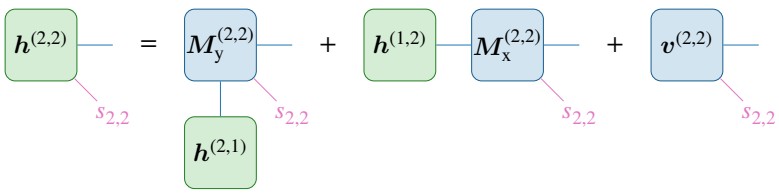



Figure 9.4: Tensor diagram for updating the memory $\boldsymbol{h}^{(2,2)}$ of the 2D MPS-RNN as in Eq. (9.19). The memories from the two previous neighboring sites $\boldsymbol{h}^{(1,2)}$ and $\boldsymbol{h}^{(2,1)}$ are taken as inputs.

With the added vertical connections, the time and the space complexities to sample the configuration $\boldsymbol{s}$ and evaluate the wave function $\psi(\boldsymbol{s})$ do not change asymptotically compared to the 1D case. The 2D MPS-RNN is more suitable for capturing the isotropic correlations in 2D systems, because the correlations between vertically neighboring sites no longer need to propagate $O(L)$ steps with exponential decay, which we will show in Section 9.6.2.

However, the 2D MPS-RNN does not have exponentially greater expressiveness than the 1D MPS-RNN to asymptotically produce the area law of entanglement entropy. It has been proven that every 2D MPS-RNN with bond dimension $\chi$ can be exactly represented by a 1D MPS-RNN with bond dimension $L\chi$, because each memory $\boldsymbol{h}^{(x,y)}$ is a linear combination of the memories $\{\boldsymbol{h}^{(x,y-1)} \mid 1 \le x \le L\}$ in the previous row.

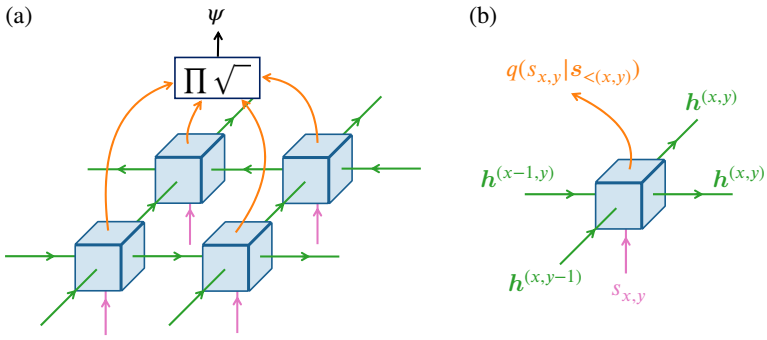



Figure 9.5: Computational graph for the 2D MPS-RNN to evaluate the wave function $\psi(\boldsymbol{s})$. In (a), four sites arranged in the snake order are shown as an example, and the phase part is omitted for simplicity. In (b), we focus on one site at $(x, y)$, assuming $x$ increases in the snake order. The site has the variational parameters $\boldsymbol{M}_{\mathrm{x}}^{(x,y)}$, $\boldsymbol{M}_{\mathrm{y}}^{(x,y)}$, and $\boldsymbol{\eta}^{(x,y)}$, takes the spin $s_{x,y}$ and the memories $\boldsymbol{h}^{(x-1,y)}$ and $\boldsymbol{h}^{(x,y-1)}$ as inputs, and outputs the conditional probability $q(s_{x,y} \mid \boldsymbol{s}_{<(x,y)})$ and the updated memory $\boldsymbol{h}^{(x,y)}$. The updated memory will be used in two future neighboring sites. This figure is reproduced from Fig. 2 in Ref. [3].

## 9.4 Tensor-RNN

We continue to look for other update rules that are expressive enough to produce the area law. Inspired by the higher-rank tensors in PEPS, we add a tensor term to the update rule in Eq. (9.19), then it can be written in components as

$$\begin{aligned}\tilde{h}_a^{(x,y)}(s_{x,y}) = & \sum_{b,c} T^{(x,y)}_{s_{x,y};a,b,c} h_b^{(x-1,y)} h_c^{(x,y-1)} \\ & + \sum_b M^{(x,y)}_{\mathrm{y};s_{x,y};a,b} h_b^{(x,y-1)} + \sum_b M^{(x,y)}_{\mathrm{x};s_{x,y};a,b} h_b^{(x-1,y)} \\ & + v^{(x,y)}_{s_{x,y};b},\end{aligned} \tag{9.20}$$

where $\boldsymbol{T}^{(x,y)}$ is a $\chi \times \chi \times \chi$ tensor and depends on the spin $s_{x,y}$, and the tensor diagram for this update rule is shown in Fig. 9.6. The RNN architecture defined by Eq. (9.20) and the previous Eqs. (9.14) and (9.18) is named tensor-RNN. Its computational graph is the same as Fig. 9.5, except that there are the additional variational parameters $\boldsymbol{T}^{(x,y)}$ at each site $(x, y)$.

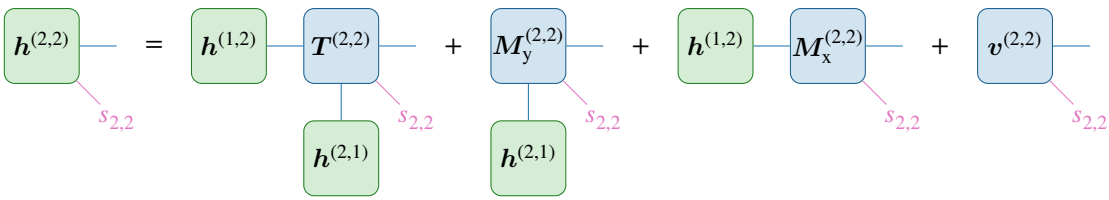



Figure 9.6: Tensor diagram for updating the memory $\boldsymbol{h}^{(2,2)}$ of the tensor-RNN as in Eq. (9.20). This figure is reproduced from Fig. 3 in Ref. [3].

The tensor-RNN takes $O(N\chi^3)$ time and space to sample the configuration $\boldsymbol{s}$ and evaluate

the wave function $\psi(\boldsymbol{s})$, which avoids the exponential computational complexity of PEPS. It can actually produce the area law, as proved by construction in Section S5 of the supplemental material for Ref. [3]. This is also supported by numerical evidence in systems more common than that artificial construction, which we will show in Section 9.6.3.

When the $O(\chi^3)$ time and space complexities introduced by the tensor term $\boldsymbol{T}$ are undesired for large systems, we can further parameterize it by

$$T_{a,b,c} = \sum_{a',b',c'=1}^{\chi'} K_{a',b',c'} U_{\mathrm{o};a,a'} U_{\mathrm{x};b,b'} U_{\mathrm{y};c,c'}, \tag{9.21}$$

which is known as the Tucker decomposition [365], where $\boldsymbol{K}$ is a smaller "kernel" tensor of size $\chi' \times \chi' \times \chi'$, $\boldsymbol{U}_{\mathrm{x}}$, $\boldsymbol{U}_{\mathrm{y}}$, and $\boldsymbol{U}_{\mathrm{o}}$ are $\chi \times \chi'$ matrices to transform the $x$-directional input, the $y$-directional input, and the output respectively, and we omit the spatial and the spin indices for clarity. The parameter $\chi'$ controls the balance between the expressiveness and the computational cost of the decomposed tensors. In the results reported, we choose $\chi' = \chi^{\frac{2}{3}}$ rounded up to an integer, so the time and the space complexities of contracting $\boldsymbol{T}$ are kept to $O(\chi^2)$. The resulting architecture is called the compressed tensor-RNN.

## 9.5 Hierarchical initialization

The ansatzes we have introduced form a hierarchy: MPS $\subsetneq$ 1D MPS-RNN $\subsetneq$ 2D MPS-RNN $\subsetneq$ compressed tensor-RNN $\subsetneq$ tensor-RNN, where each architecture can exactly represent any architecture to its left with the same bond dimension. This property can be utilized in the training of ansatzes. We first optimize an MPS using DMRG, then convert it to a 1D MPS-RNN, which represents the same wave function but has more potential expressiveness, and continue to train all its parameters. When the 1D MPS-RNN is sufficiently trained, we convert it to a 2D MPS-RNN and repeat the procedure, until we obtain the tensor-RNN.

This procedure is referred to as hierarchical initialization. It provides the initial parameters at reasonably low energies for each ansatz and prevents them from getting stuck in local minima and saddle points with high energies. For non-stoquastic Hamiltonians, it also provides reasonable sign structures from DMRG, which can be hard to obtain by gradient-based optimizers, as discussed in Section 7.6. This procedure is a particular advantage of our approach with exact mappings, which cannot be performed in other hybrid architectures of TN and RNN [72, 362, 363].

## 9.6 Numerical results

### 9.6.1 Variational energy

The performance of the above ansatzes is demonstrated in numerical experiments using the antiferromagnetic Heisenberg model (AFHM) and the transverse-field Ising model (TFIM) on 2D square and triangular lattices with open boundary conditions (OBC), as defined in Section 2.2.

We first investigate the variational energy as a function of the bond dimension which is the only hyperparameter in the architectures of these ansatzes and controls their overall sizes, as shown in Fig. 9.7. We can clearly see that the 2D ansatzes reach the same error as the 1D ones with bond dimensions less by one to two orders of magnitude, which is attributed to the vertical connections. Moreover, the error can be systematically reduced as the bond dimension increases. For the tensor-RNN, the reduction of the error particularly exhibits a power law, which can be fitted by a straight line on the log-log plot, and on the square lattice it eventually outperforms the best result of PEPS [326] to the author's knowledge.

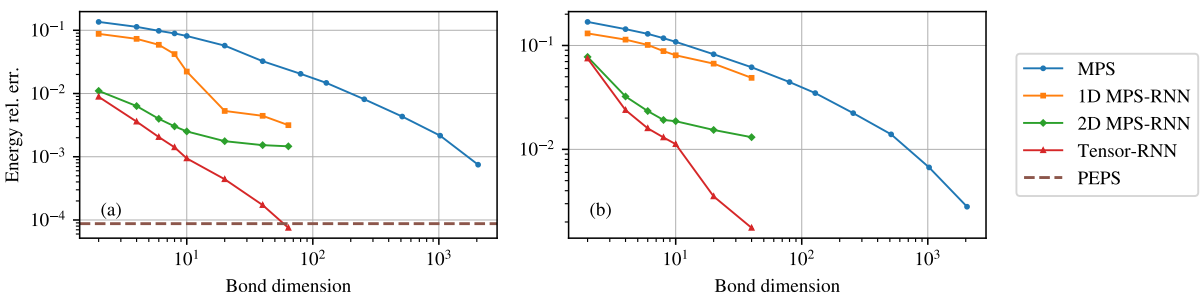



Figure 9.7: Variational energy as a function of the bond dimension $\chi$, for the AFHM on (a) square and (b) triangular lattice with $10 \times 10$ sites and open boundary conditions (OBC). The comparison is made between MPS (optimized by DMRG) and the ansatzes in this chapter (optimized by VMC with hierarchical initialization). The results are shown as the relative error from (a) QMC [226] and (b) DMRG with $\chi = 4096$. For the square lattice, the result of PEPS with $\chi = 10$ [326] is also taken into comparison. This figure is reproduced from Fig. 4 in Ref. [3].

We also present the plots of the variational energy as a function of the number of parameters in Fig. 9.8, which better represents the computational costs of different ansatzes than the bond dimension. From this perspective, the 2D ansatzes can reach the same error as the 1D ones with parameters fewer by more than two orders of magnitude, and the error still decays by the power law.

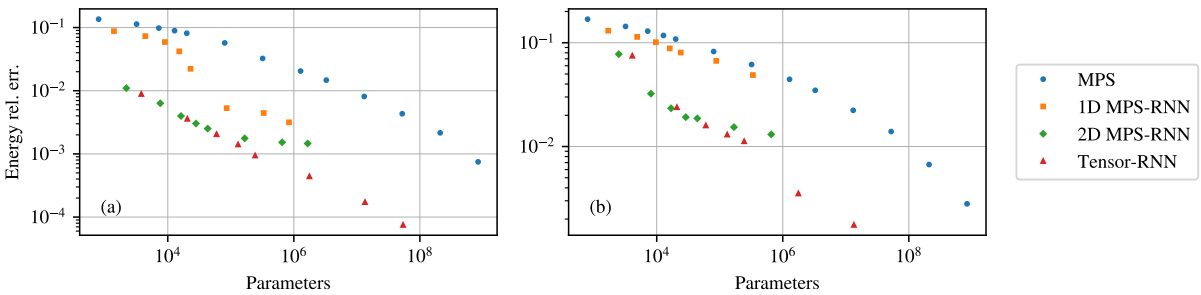



Figure 9.8: Variational energy as a function of the number of parameters, in the same setting as Fig. 9.7. This figure is reproduced from Fig. S3 in the supplemental material for Ref. [3].

The tensor-RNN has better parameter efficiency than the original MPS on both the square and the triangular lattices. However, we remark that the relative error on the triangular lattice

is higher than that on the square lattice by an order of magnitude for all ansatzes in the plot, which means the frustrated sign structure is still a challenge for the tensor-RNN.

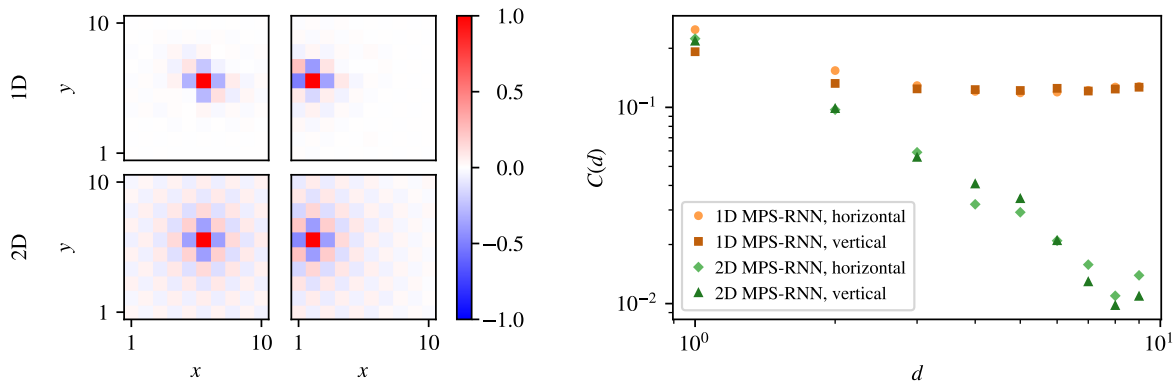
### 9.6.2 Spatial correlations

Next, we examine the ability of our ansatzes to capture spatial correlations in 2D systems. A qualitative visualization of the correlations with a certain spin is shown in Fig. 9.9 (a) for the AFHM. The 2D MPS-RNN successfully produces isotropic and long-range correlations, as expected in Section 9.3, while the 1D one fails to do so.

To demonstrate the particularly interesting case of algebraically decaying correlations in 2D critical systems, we use the TFIM at its critical point $\Gamma = 3$, and increase the lattice size to $20 \times 20$ to better investigate the asymptotic behavior. As shown in Fig. 9.9 (b), the 2D MPS-RNN correctly captures the algebraically decaying correlations in both the horizontal and the vertical directions, which can be fitted by a straight line in the log-log plot, while the 1D one can only produce constant correlations in the long range.

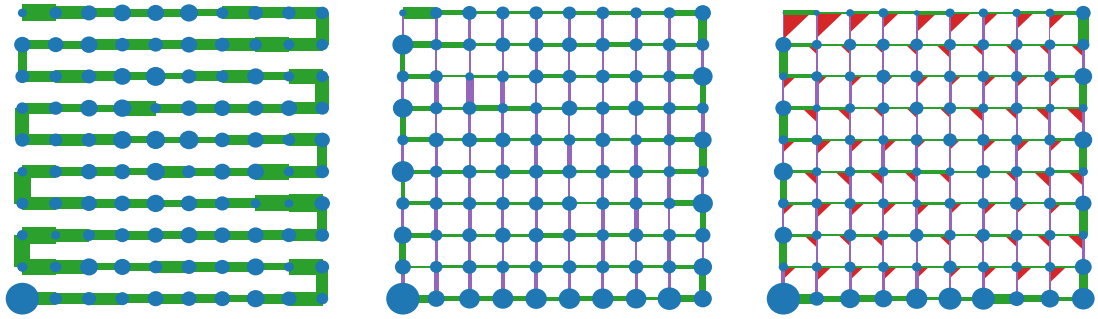
### 9.6.3 Magnitudes of tensor, matrix, and vector terms

When evaluating the above ansatzes, the memory $\boldsymbol{h}^{(x,y)}$ at each site can consist of contributions from the tensor terms, the matrix terms, and the vector term, and it is worth investigating their respective magnitudes. Considering how $\boldsymbol{h}^{(x,y)}$ is used in Eq. (9.14) to compute the




(a) $10 \times 10$ AFHM (b) $20 \times 20$ TFIM

Figure 9.9: (a) Correlations of all spins $s_{x,y}$ with the spin $s_{5,5}$ (left column) and $s_{1,5}$ (right column) for the AFHM on the $10 \times 10$ square lattice with OBC, given by the 1D MPS-RNN (top row) and the 2D MPS-RNN (bottom row) with bond dimension $\chi = 10$.
(b) Correlation $C(d)$ as a function of the distance $d$ in Eq. (8.8) for the TFIM at its critical point $\Gamma = 3$, on the $20 \times 20$ square lattice with OBC, in horizontal and vertical directions, given by the 1D MPS-RNN and the 2D MPS-RNN. This figure is reproduced from Fig. S4 in the supplemental material for Ref. [3].



(a) 1D MPS-RNN (b) 2D MPS-RNN (c) Tensor-RNN

Figure 9.10: Contributions of the tensor term $\tilde{T}^{(x,y)}$ (areas of red triangles), the matrix terms $\tilde{M}_{\mathrm{x}}^{(x,y)}$, $\tilde{M}_{\mathrm{y}}^{(x,y)}$ (areas of rectangles), and the vector term $\tilde{v}^{(x,y)}$ (areas of blue circles) to the memory $\boldsymbol{h}^{(x,y)}$ at each site, in (a) 1D MPS-RNN, (b) 2D MPS-RNN, and (c) tensor-RNN with bond dimension $\chi = 10$, trained for the AFHM on the $10 \times 10$ square lattice with OBC. In the matrix terms, the green rectangles are along the snake order, and the purple ones are not. This figure is reproduced from Fig. S5 in the supplemental material for Ref. [3].

probability, we define the 2-norms weighted by $\boldsymbol{\eta}^{(x,y)}$:

$$\|T\|^{(x,y)}(\boldsymbol{s}) = \sqrt{\sum_a \eta_a^{(x,y)} \left| \sum_{b,c} T^{(x,y)}_{s_{x,y};a,b,c} h_b^{(x-1,y)} h_c^{(x,y-1)} \right|^2}, \tag{9.22}$$

$$\|M\|_{\mathrm{x}}^{(x,y)}(\boldsymbol{s}) = \sqrt{\sum_a \eta_a^{(x,y)} \left| \sum_{b} M^{(x,y)}_{\mathrm{x};s_{x,y};a,b} h_b^{(x-1,y)} \right|^2}, \tag{9.23}$$

$$\|M\|_{\mathrm{y}}^{(x,y)}(\boldsymbol{s}) = \sqrt{\sum_a \eta_a^{(x,y)} \left| \sum_{b} M^{(x,y)}_{\mathrm{y};s_{x,y};a,b} h_b^{(x,y-1)} \right|^2}, \tag{9.24}$$

$$\|v\|^{(x,y)}(\boldsymbol{s}) = \sqrt{\sum_a \eta_a^{(x,y)} \left| v^{(x,y)}_{s_{x,y};b} \right|^2}, \tag{9.25}$$

assuming $x$ increases in the snake order. As they are functions of the spin configuration $\boldsymbol{s}$, we further define the averages

$$\overline{\|T\|}^{(x,y)} = \sum_{\boldsymbol{s}} q(\boldsymbol{s}) \|T\|^{(x,y)}(\boldsymbol{s}), \tag{9.26}$$

and similarly $\overline{\|M\|}_{\mathrm{x}}^{(x,y)}$, $\overline{\|M\|}_{\mathrm{y}}^{(x,y)}$, and $\overline{\|v\|}^{(x,y)}$, which are all estimated by Monte Carlo sampling. To ease the visualization, we normalize them at each site:

$$\tilde{T}^{(x,y)} = \frac{\overline{\|T\|}^{(x,y)}}{\overline{\|T\|}^{(x,y)} + \overline{\|M\|}_{\mathrm{x}}^{(x,y)} + \overline{\|M\|}_{\mathrm{y}}^{(x,y)} + \overline{\|v\|}^{(x,y)}}, \tag{9.27}$$

and similarly $\tilde{M}_{\mathrm{x}}^{(x,y)}$, $\tilde{M}_{\mathrm{y}}^{(x,y)}$, and $\tilde{v}^{(x,y)}$.

The results are visualized in Fig. 9.10. In the 1D MPS RNN, the matrix terms are particularly large near the ends of rows because they need to transfer all information about the current

row to the next row. As for the case of 2D MPS RNN, most matrix terms have the same order of magnitude regardless of their directions. In the tensor-RNN, the interesting phenomenon is that the tensor terms are particularly large only in the last row, which are essential to produce the area law of entanglement entropy, as expected in Section 9.4. In contrast, if there is a large tensor term on every site, then a volume law would be produced.

## 9.7 Conclusion

We have presented the tensor-RNN ansatz, which is a hybrid architecture between TN and RNN taking advantage of both. Compared to MPS, which can only produce $O(\ln \chi)$ entanglement entropy and exponentially decaying correlations, there is both analytical and numerical evidence for tensor-RNN to produce the 2D area law of entanglement entropy, as well as numerical evidence for algebraically decaying correlations in 2D critical systems. Compared to PEPS, which has an exponentially high computational cost, tensor-RNN supports efficient evaluation in polynomial time, as well as exact sampling in the AR order. As a result, tensor-RNN has higher parameter efficiency than MPS, and also achieves better accuracy of variational approximation than PEPS in a practical computation time.

On the other hand, compared to conventional RNN architectures, which are difficult to analytically study and adaptively scale up, the simple multilinear architecture of tensor-RNN allows us to investigate its analytical properties, and systematically improve its accuracy by increasing the bond dimension. The exact mapping from MPS to tensor-RNN leads to the hierarchical initialization, which makes it possible to optimize tensor-RNN with numerous parameters. It remains an open question to exactly construct more 2D states of interest using tensor-RNN, such as the RVB state and the toric code, or to provide analytical evidence for the algebraically decaying correlations.

# 10 VarBench: variational benchmarks for quantum many-body systems

The previous chapters have introduced multiple variational methods for quantum many-body systems, including variational Monte Carlo (VMC) in Chapter 7 and tensor networks (TNs) in Chapter 8, each with its own appropriate use cases, as well as trade-off between accuracy and computational cost. When different methods are applied to the same physical system, there is a simple criterion to assess their performance: A lower variational energy indicates that the method approximates the true ground state more accurately. Besides that, it is also attempting to compare the accuracies of one method on different physical systems, or compare the universal applicability of two methods on multiple systems. Therefore, we require a metric that can be compared both between methods and between systems.

In the following, we present the metric proposed in Ref. [5], namely V-score. It is used in the VarBench project, an extensive collection of benchmarks for the accuracies of variational methods on a diverse variety of quantum many-body systems. Moreover, the V-score can measure the hardness of simulating a physical system, which identifies certain hard systems that are not yet solved by available computational techniques.

## 10.1 V-score: universal metric for variational accuracy

It has been widely observed that when a variational state is sufficiently close to the ground state, the variance of the variational energy $\mathrm{Var}E = \langle \hat{H}^2 \rangle - \langle \hat{H} \rangle^2$ scales linearly with the difference between the variational energy $E$ and the ground state energy $E_0$ [366–368]:

$$E - E_0 = A\mathrm{Var}E + O\left((\mathrm{Var}E)^2\right), \tag{10.1}$$

where $A$ is a coefficient depending on the properties of the Hamiltonian $\hat{H}$ and the variational state $|\psi\rangle$. For example, in a simple two-state system as discussed in Eq. (7.10), we have

$$A = \frac{1}{E_1 - E_0}. \tag{10.2}$$

For general systems with more energy levels, Ref. [369] has given more rigorous conditions of "sufficiently close" to hold Eq. (10.1), which are usually satisfied in computational studies. It has been common practice to linearly extrapolate $E$ as a function of $\mathrm{Var}E$ and obtain an

estimation of $E_0$ at $\mathrm{Var}E = 0$. Therefore, even though we do not know the true ground state $E_0$, the variance can indicate the accuracy of approximating the given Hamiltonian, and we seek a metric proportional to it.

The metric should be a dimensionless number, so we normalize the variance by the variational energy and obtain $\frac{\mathrm{Var}E}{E^2}$. Moreover, it should not scale with the system size $N$. The energy $E$ scales linearly with $N$, as it is an extensive quantity for all physical systems of our interest. The variance $\mathrm{Var}E$ also scales linearly with $N$ if the Hamiltonian satisfies the following clustering property [370–372]: The Hamiltonian can be written as a sum of $N$ terms labeled by the site index $i$,

$$\hat{H} = \sum_{i=1}^{N} \hat{h}_i, \tag{10.3}$$

and the correlations between these terms decay quickly enough with the spatial distance of the sites,

$$\left| \langle \hat{h}_i \hat{h}_j \rangle - \langle \hat{h}_i \rangle \langle \hat{h}_j \rangle \right| \leq \frac{A_{\mathrm{corr}}}{d^{D+\epsilon}(i,j)}, \tag{10.4}$$

where $A_{\mathrm{corr}}$ is a constant for all $i, j$, $d(i,j)$ is the distance between the sites $i, j$, $D$ is the spatial dimension, and $\epsilon$ is a small positive number for all $i, j$. All physical systems of our interest have this clustering property, and we note that for non-critical systems the correlations decay exponentially with the distance. Therefore, we scale the variance by $N$ and obtain $\frac{N\mathrm{Var}E}{E^2}$, which is asymptotically independent of $N$.

The remaining issue is that we can arbitrarily shift the Hamiltonian by a constant: $\hat{H} \leftarrow \hat{H} + C$, which does not change any physical property of the system, and our metric should be invariant under this shift. To achieve this, we define a zero point of energy $E_\infty$, and replace $E$ by $E - E_\infty$. For spin and fermionic systems with a finite number of energy levels per site, $E_\infty$ is defined to be the expected energy of a random state in the Hilbert space, and we are sure that a variational state after optimization typically has a lower energy than a random state. It is equivalent to the expected energy of a thermal state at infinite temperature, which can be derived from the trace of the Hamiltonian:

$$E_\infty = \frac{\mathrm{tr}\,\hat{H}}{\dim \mathcal{H}}, \tag{10.5}$$

where $\dim \mathcal{H}$ is the dimension of the Hilbert, or the Hilbert subspace for fermionic systems when the number of fermions is fixed. It can be analytically derived for simple Hamiltonians or numerically evaluated using Monte Carlo sampling. However, it remains an open question to define $E_\infty$ for bosonic systems because there are infinitely many energy levels per site. We may define it with a cutoff of physically relevant energy levels or a mean-field energy, but there is not yet an intuitive definition that is consistent with spin and fermionic systems.

The resulting definition of the metric for variational accuracy, namely V-score, is

$$\text{V-score} = \frac{N\mathrm{Var}E}{(E - E_\infty)^2}. \tag{10.6}$$

It is dimensionless, asymptotically independent of the system size, and invariant to the energy

shift. Moreover, it can be efficiently evaluated without knowing the ground state energy. Substituting Eqs. (10.1) and (10.6) into the energy relative error $\frac{E-E_0}{|E_0-E_\infty|}$, we can obtain the linear relation

$$\frac{E-E_0}{|E_0-E_\infty|} = B\,\text{V-score} + O(\text{V-score}^2), \tag{10.7}$$

$$B = \frac{A(E-E_\infty)^2}{N|E_0-E_\infty|}, \tag{10.8}$$

where $A$ has the dimension of $E^{-1}$ and is asymptotically independent of $N$. For any Hamiltonian with the clustering property and any variational state converging to the ground state, the magnitude of $B$ will not be far from 1. Therefore, the V-score can be a universal metric to predict the energy relative error in variational approximation, which will be visualized in Fig. 10.2.

It is worth discussing the possibilities for the V-score to underestimate or overestimate the energy relative error. An intuitive case of the underestimation is when the variational state converges to an excited state, as Var$E$ under any excited state is zero. This situation can be detected by applying the symmetries of the ground state or repeating the optimization with different initial parameters. Another case is the mode collapse discussed in Section 7.2, where we estimate Var$E$ using $|E_{\text{loc}}(\boldsymbol{s})|^2$, and it can be detected by actually evaluating the matrix elements of $\hat{H}^2$.

If the V-score overestimates the energy relative error, an upper bound of their ratio is proven that

$$\frac{\text{V-score}}{(E-E_0)/(E_\infty-E_0)} \le N\frac{(E_\infty-E_0)(E_{\text{M}}-E)}{(E_\infty-E)^2}, \tag{10.9}$$

where $E_{\text{M}}$ is the highest energy level of the system, and the variational state can be an arbitrary linear combination of the eigenstates. This is a loose upper bound as it scales linearly with $N$, and it remains an open question to find a tighter upper bound under more assumptions, such as the variational state being close to the ground state.

## 10.2 Hamiltonians and variational methods in the benchmark

To demonstrate the universal applicability of V-score, in the VarBench project we have collected more than 400 numerical studies with many kinds of variational methods on spin and fermionic Hamiltonians. The Hamiltonians mainly include the following:

1. The spin-$\frac{1}{2}$ transverse-field Ising model (TFIM)

$$\hat{H} = J\sum_{\langle i,j\rangle}\hat{\sigma}_i^z\hat{\sigma}_j^z + \Gamma\sum_i\hat{\sigma}_i^x, \tag{2.24}$$

   where $\langle i,j\rangle$ runs over nearest neighbors, $\Gamma$ is the transverse field strength, $\hat{\sigma}^x, \hat{\sigma}^y, \hat{\sigma}^z$ are the Pauli matrices, and we always set $J = 1$.

2. The spin-$\frac{1}{2}$ Heisenberg model
$$\hat{H} = J \sum_{\langle i,j \rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j, \tag{2.20}$$
where $\hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j = \hat{\sigma}_i^x \hat{\sigma}_j^x + \hat{\sigma}_i^y \hat{\sigma}_j^y + \hat{\sigma}_i^z \hat{\sigma}_j^z$.
3. The spin-$\frac{1}{2}$ $J_1$-$J_2$ model
$$\hat{H} = J_1 \sum_{\langle i,j \rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j + J_2 \sum_{\langle\langle i,j \rangle\rangle} \hat{\boldsymbol{\sigma}}_i \cdot \hat{\boldsymbol{\sigma}}_j, \tag{10.10}$$
where $\langle\langle i, j \rangle\rangle$ runs over next-nearest neighbors (diagonals if the lattice is a 2D grid), $J_2$ is the next-nearest neighbor interaction strength, and we always set $J_1 = 1$.
4. The spinless fermionic $t$-$V$ model
$$\hat{H} = -t \sum_{\langle i,j \rangle} \left( \hat{c}_i^\dagger \hat{c}_j + \hat{c}_j^\dagger \hat{c}_i \right) + V \sum_{\langle i,j \rangle} \hat{n}_i \hat{n}_j, \tag{10.11}$$
where $V$ is the Coulomb repulsive interaction strength, $\hat{c}_i^\dagger$ and $\hat{c}_i$ are the fermionic creation and annihilation operators respectively, $\hat{n}_i = \hat{c}_i^\dagger \hat{c}_i$ is the number operator, the number of fermions is fixed to $N_\mathrm{f}$, and we always set $t = 1$.
5. The spin-$\frac{1}{2}$ fermionic Hubbard model
$$\hat{H} = -t \sum_{\langle i,j \rangle, \sigma} \left( \hat{c}_{i\sigma}^\dagger \hat{c}_{j\sigma} + \hat{c}_{j\sigma}^\dagger \hat{c}_{i\sigma} \right) + U \sum_i \hat{n}_{i\uparrow} \hat{n}_{i\downarrow}, \tag{10.12}$$
where $U$ is the on-site interaction strength, the numbers of fermions are fixed to $N_\uparrow$ and $N_\downarrow$, and we only consider the case of $N_\uparrow = N_\downarrow$.

These Hamiltonians can be defined on all kinds of lattice geometries, including the non-frustrated 1D chains and 2D grids, the frustrated 2D triangular [17, 50], kagome [18], and shuriken [373, 374] lattices, as well as 3D pyrochlore lattices [76, 268]. The boundary conditions of these lattices also affect the results. In addition, we include Anderson impurity models, which have applications in fields such as quantum embedding and nanoelectronic devices [287, 375–378].

The variational methods benchmarked are categorized by:

1. VMC with physically motivated ansatzes (PMAs) in Sections 7.5.1 and 7.5.2.
2. VMC with neural quantum states (NQSs) in Section 7.5.3.
3. TNs in Chapter 8, most of which are matrix product states (MPSs) optimized using the density matrix renormalization group (DMRG).
4. Parameterized quantum circuits (PQCs) [305, 379] optimized using variational quantum eigensolver (VQE) [380–382].
5. Variational auxiliary-field quantum Monte Carlo (VAFQMC) [383] for the Hubbard model.

An overview of the complete benchmark results is visualized in Fig. 10.1.

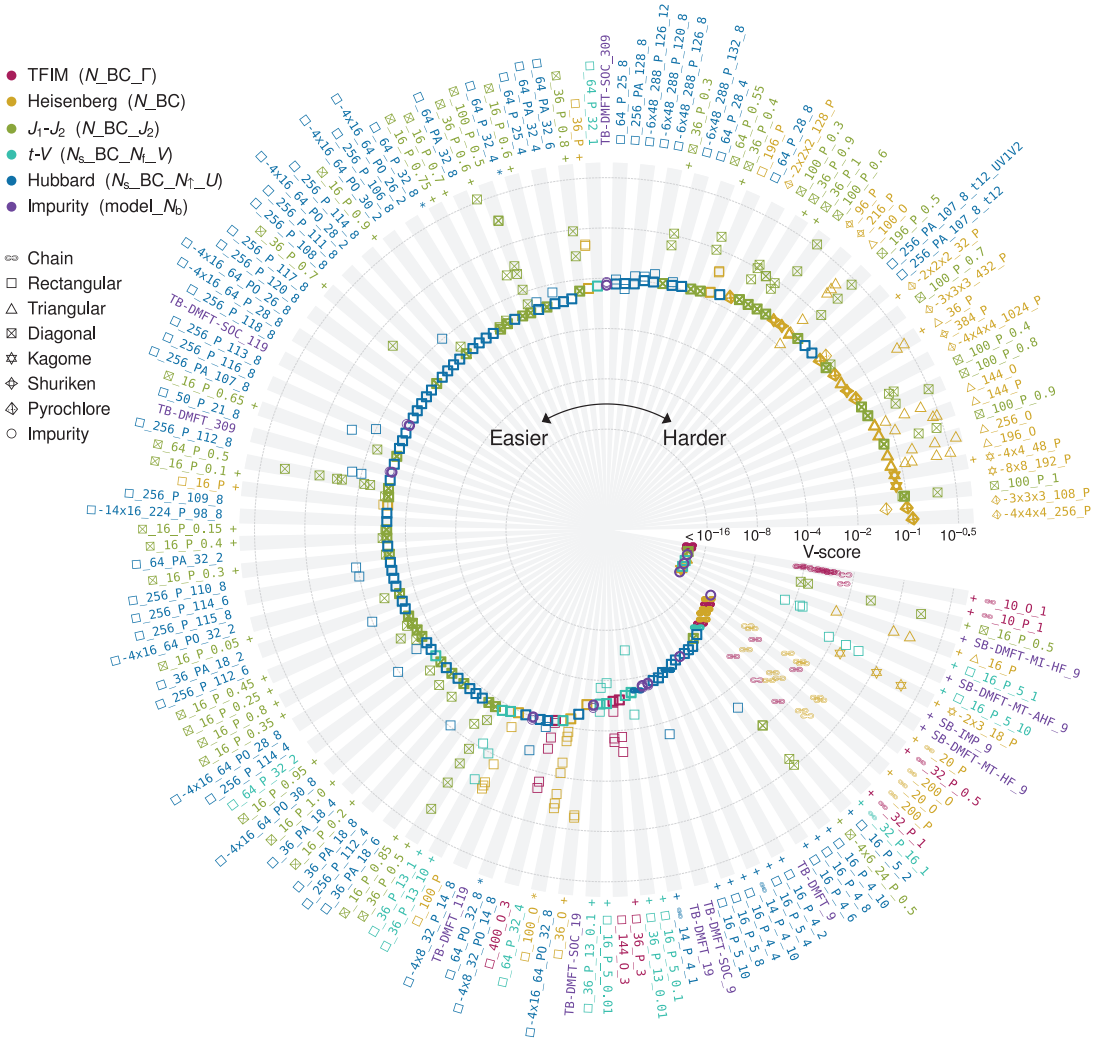



Figure 10.1: Overview of the complete benchmark results. Each slice in the radial direction contains different variational methods for a given Hamiltonian. The bold marker indicates the method with the lowest variational energy (rather than the lowest V-score), which defines the V-score for the Hamiltonian and determines its clockwise ranking. The radial axis uses the doubly log scale to show the results across a wide range of V-score magnitude. The results with V-scores less than $10^{-16}$ are indistinguishable from the exact solutions due to limited numerical precision. The label outside each slice is the name of the Hamiltonian in the dataset, which describes the lattice geometry, the lattice size $N_s$, the boundary conditions (BC), and the Hamiltonian parameters such as the interaction strength ($\Gamma$, $J_2$, $V$, $U$) and the number of particles ($N_f$, $N_\uparrow$, $N_b$). The BC can be O (open), P (periodic), A (anti-periodic) on each dimension. The '+' marker near the label indicates that an ED solution is available for that Hamiltonian, and the '$*$' marker indicates a numerically exact QMC solution. This figure is reproduced from Fig. 3 in Ref. [5].

## 10.3 Benchmark results

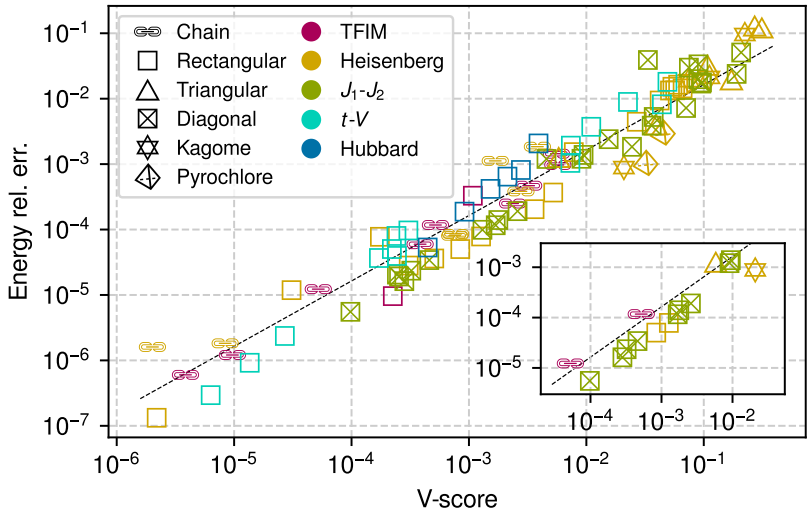



Figure 10.2: Comparison of V-score and energy relative error for systems whose exact solutions from ED or QMC are available. The black dashed line is fitted by least-squares regression with the slope fixed to 1. The inset focuses on PQC results simulated on classical hardware without noise. This figure is reproduced from Fig. 2 in Ref. [5].

We first compare the V-scores with the known energy relative errors, for small systems whose ground state energies are obtained by exact diagonalization (ED), and simple systems for which quantum Monte Carlo (QMC) methods produce numerically exact results. As shown in Fig. 10.2, the results in the log-log plot can be fitted with high confidence using least-squares regression where the slope is fixed at 1. This validates their linear relation in Eq. (10.7), where the magnitude of the coefficient $B$ does not change substantially between Hamiltonians and variational methods. Therefore, we can use the V-score to predict the energy relative error, at least up to the order of magnitude, for systems whose ground state energies cannot be practically obtained in other ways.

Then we select some Hamiltonians of interest and compare different kinds of variational method on them. It is generally agreed that for 1D systems, MPSs can solve them with high accuracy, and the more complicated NQSs and PQCs are unnecessary, which is shown in Fig. 10.3 (a). In contrast, for the 2D $J_1$-$J_2$ model with frustrated interactions, especially in the supposed spin liquid phase [77–80, 256, 273, 279], the most accurate records are produced by NQSs. Therefore, the V-score can clearly demonstrate the appropriate use cases of different variational methods.

## 10.4 Hardness of Hamiltonian

The V-score can assess not only the accuracy of any variational method but also the hardness to simulate any Hamiltonian. If we have applied all available methods but still cannot obtain

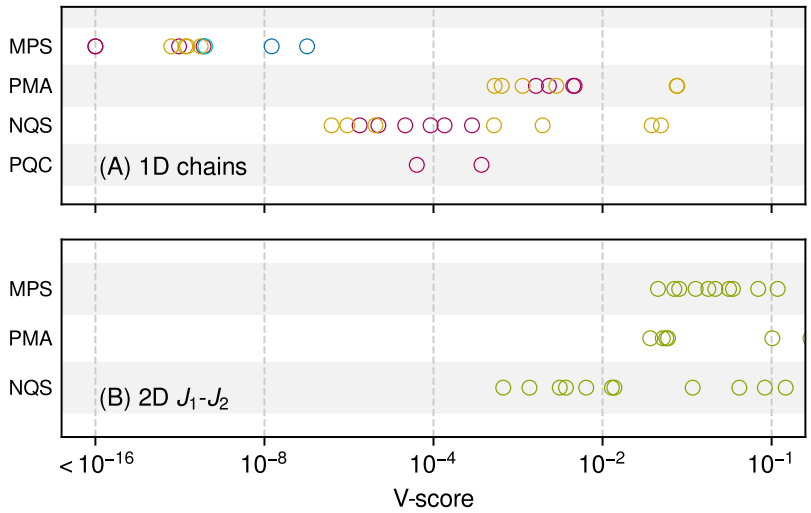



Figure 10.3: V-scores of different variational methods, including matrix product state (MPS), physically motivated ansatz (PMA), neural quantum state (NQS), and parameterized quantum circuit (PQC), for (a) all Hamiltonians on 1D chains and (b) the $J_1$-$J_2$ model on 2D grids. The $x$-axis uses the doubly log scale to show the results across a wide range of V-score magnitude.

a low V-score, then we consider the Hamiltonian to be hard, so the V-score of the best method (with the lowest variational energy, rather than the lowest V-score) is practically defined to be the V-score of the Hamiltonian.

The V-scores of the Hamiltonians benchmarked are plotted in Fig. 10.4. The easy Hamiltonians include the stoquastic Ising and Heisenberg models on the non-frustrated chains and grids, as well as small-sized systems, which are already well solved with V-scores lower than $10^{-4}$. In contrast, the hard Hamiltonians include the non-stoquastic $J_1$-$J_2$ and Hubbard models, as well as the frustrated lattices, whose V-scores remain higher than $10^{-2}$.

These hard Hamiltonians can serve as the aims for the development of more advanced computational methods, particularly quantum computers. In quantum computing theory, random circuit sampling [384–386] is a traditional task to demonstrate the quantum advantage, but currently developed quantum computing platforms actually do not outperform large-scale classical computers on this task [387–390]. Meanwhile, the demonstration of quantum advantage on tasks with more practical impact is still under active research [391]. The hard Hamiltonians identified by the V-score are exactly a kind of tasks promising for quantum computers to solve [392], with practical impact in physics.

In the inset of Fig. 10.2, we provide the initial insights of small-scale PQCs simulated on classical hardware without noise, where their V-scores reliably predict their energy relative errors. While larger-scale simulations of PQCs and noisy intermediate-scale quantum (NISQ) platforms are in active development, the V-score will be a useful metric to track their progress.

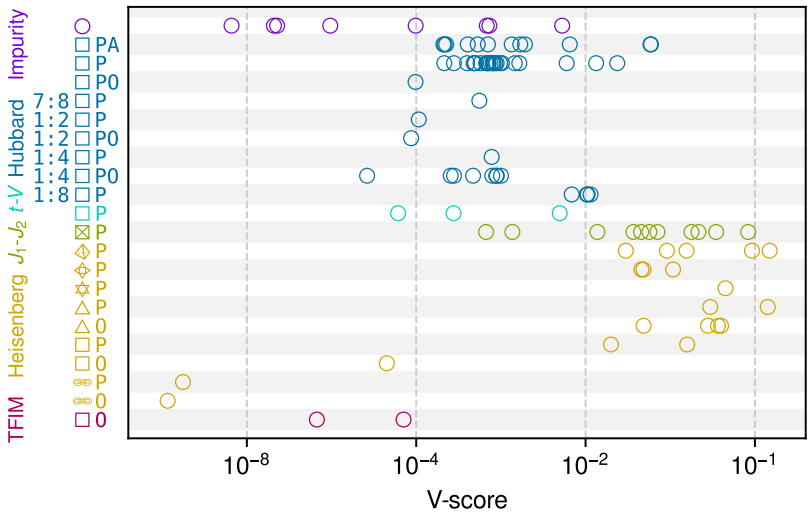



Figure 10.4: V-scores of different Hamiltonians, classified by Hamiltonian types and lattice geometries. Only the Hamiltonians without ED results are shown. The $x$-axis uses the doubly log scale to show the results across a wide range of V-score magnitude. The ratios to the left of the lattice icons are aspect ratios of rectangular lattices. The letters to the right are boundary conditions, which can be O (open), P (periodic), A (anti-periodic) on each dimension. This figure is reproduced from Fig. 4 in Ref. [5].

## 10.5 Conclusion

In conclusion, the V-score is a universal metric for the accuracy of variational approximation, which can be compared both between variational methods and between Hamiltonians. It also measures the hardness of simulating any Hamiltonian and identifies the hard ones that are not yet solved by any available method. These Hamiltonians are the targets for the future development of computational methods, especially the demonstration of quantum advantage. The generalization of the V-score to bosonic and molecular systems is still under investigation, which would further extend its universality. It would also be interesting to study the behavior of the V-score with the change of Hamiltonian parameters such as $J_2$, and possibly discover some universal critical exponents.

The VarBench project is an extensive collection of benchmarks for many kinds of variational methods and quantum many-body systems. Besides witnessing the previous development of methods, it also provides a framework to test new methods on a collection of Hamiltonians, which helps to thoroughly and orthogonally assess their performance. We plan to continuously update it with the development of new methods and computational power, especially experimental results of quantum computing platforms, and possibly set up a website like PubChem and PapersWithCode to ease the collaboration of the community. A topic not covered by the VarBench project is the computational costs of all these methods, and it would be another huge challenge to systematically measure them and quantify their relation to the accuracies of the methods.

# 11 Conclusion

## 11.1 Main results

In this thesis, we have applied exact sampling with ARNN to both classical and quantum many-body systems. For classical systems, in Section 5.2 we have shown that ARNN obtains substantially more accurate variational free energy than traditional NMF and Bethe ansatzes on the square Ising model, estimates the residual entropy for the triangular Ising model, and retrieves the memorized patterns for the Hopfield model. It also outperforms NMF and Bethe ansatzes on the SK model and the inverse SK problem, showing its potential to capture exponentially many modes of the complicated many-body distribution in polynomial amount of computation. The globally interacting systems only require small dense ARNNs with $O(N^2)$ parameters, while larger ARNNs have better accuracies for the Ising models on the lattices. The convolutional ARNN has a better parameter efficiency than the dense one on the lattices because of the incorporated translational symmetry and locality of the physical system, and it needs a sufficiently large receptive field to capture the long-range correlations around the critical point.

In Section 5.3 we have presented the TwoBo ansatz, which incorporates the knowledge of the Boltzmann distribution and the sparse two-body interacting Hamiltonian into the neural network design. When applied to the EA model with binary interactions on 2D grids, 3D grids, and RRGs, it achieves more accurate variational free energy and ground state energy, substantially faster convergence in training, and polynomially higher parameter efficiency on the regular grids, compared to the dense ARNN and the previously proposed RNN.

In Section 5.4 we have demonstrated the NCUS sampling method, which consists of cluster updates constructed by the AR property of the ansatz and symmetry operations of the physical system. When applied to the Ising model with the second-order phase transition and the FPM with the first-order phase transition, it produces autocorrelation times lower than the conventional global update method using the same ARNN, and comparable to the Wolff cluster update method specially designed for the Ising model. This advantage is particularly substantial around the critical point, leading to more reliable estimations of observables with lower error bars.

For quantum systems, in Chapter 9 we have developed the tensor-RNN ansatz with both the advantages of TN and RNN. Compared to MPS, which can only produce $O(\ln \chi)$ entangle-

ment entropy and exponentially decaying correlations, there is both analytical and numerical evidence for tensor-RNN to produce the 2D area law of entanglement entropy, as well as numerical evidence for algebraically decaying correlations in 2D critical systems. Compared to PEPS, which has an exponentially high computational cost, tensor-RNN supports efficient evaluation of the wave function in polynomial time, as well as exact sampling. As a result, it has higher parameter efficiency than MPS, and also achieves better accuracy of variational approximation than PEPS in a practical computation time. Meanwhile, compared to conventional RNN architectures which are difficult to analytically study and adaptively scale up, the physical properties of tensor-RNN have been analytically investigated, and its accuracy can be systematically improved by increasing the bond dimension. The exact mapping from MPS to tensor-RNN leads to the hierarchical initialization, which makes it possible to optimize tensor-RNN with numerous parameters.

Lastly, in Chapter 10 we have presented the VarBench project, which is an extensive collection of benchmarks for many kinds of variational methods and quantum many-body systems in more than 400 numerical studies. The metric used in these benchmarks is the V-score, a universal quantity to measure the accuracy of any variational approximation, as well as the hardness of simulating any Hamiltonian, which is derived from the relation between the variance and the accuracy of the variational energy. Using this metric, we have identified certain hard Hamiltonians that have not yet been solved by any available method, which can be the targets to demonstrate the quantum advantage in the future.

## 11.2 Outlook

A large topic that has been deliberately omitted in this thesis is the symmetries in the ansatz. Although we can always implement the symmetries "out" of the ansatz using a mixture of symmetrized copies of the ansatz, commonly known as the quantum number projection [267], it introduces a factor of $|G|$ on the computation time, where $|G|$ is the size of the symmetry group. The more feasible approach is to implement the symmetries "in" the architecture of the ansatz, such as CNNs and group equivariant neural networks, and even the SU(2) ClebschTree [280]. However, a common criticism of ARNN is that it breaks the symmetries as long as it artificially defines an AR order. It can be promising to implement ARNNs with random orders [213–215], hierarchical architectures [194, 195], or inputs in frequency or wavelet spaces [196, 197], especially considering the recent theory that the diffusion models, being the state-of-the-art models for 2D images, can be formulated as AR models in the frequency space. Meanwhile, the constraint of fixed total magnetization or fixed number of fermions can already be implemented in ARNNs as described in Appendix D.2 of Ref. [282].

Another issue concerning the title of this thesis is whether we can find clear evidence that exact sampling outperforms MCMC sampling in real use cases of variational optimization. Although we have been using ARNN intuitively with AR sampling since the early studies, there are results showing that ARNN with AR sampling does not significantly outperform the same ARNN with MCMC sampling in terms of accuracy and computation time. Ref. [257] even provides evidence that the full-basis simulation does not produce substantially more

accurate variational energy than the Monte Carlo sampling for a small quantum system. Moreover, despite the recent achievement of ViT in modeling natural images, it seems that the autoregressive architecture of ViT is unnecessary for physical systems [285, 288]. Although we can show the advantage of exact sampling in toy models such as Fig. 3.6, it is unclear how much the discrepancy between the ansatz distribution and its MCMC samples affects variational optimization, and there have been studies of the convergence properties of SGD with autocorrelated samples from Markov chains [393]. A future step of research is to show the advantage of exact sampling in controlled experiments on systems with more complicated energy landscapes.

Speaking of ViT, it motivates us to introduce more techniques that are proven effective in machine learning practices into the solution of physical systems. A promising technique is the positional encoding [394], which allows convolutional ARNNs to know its position in the system. Methods for scaling the positional encoding have been developed to enable longer context windows in language models [395, 396], which may be helpful in scaling physical systems to the thermodynamic limit.

# A Energy gradient in VMC

In this appendix, we present the derivation of the gradient in Eq. (7.11), which is used to minimize the variational energy in VMC. For clarity, we use $s$ to denote the configuration $\boldsymbol{s}$ in the main text and $H_{ss'}$ to denote the Hamiltonian entry $H(\boldsymbol{s}, \boldsymbol{s}')$. The variational energy from Eq. (7.4) is

$$E = \frac{\sum_{s,s'} \psi^*(s') H_{s's} \psi(s)}{\sum_{s''} \psi^*(s'')\psi(s'')} = \sum_s q(s) E_{\mathrm{loc}}(s), \tag{A.1}$$

where

$$q(s) = \frac{\psi^*(s)\psi(s)}{\langle \psi | \psi \rangle}, \tag{A.2}$$

$$E_{\mathrm{loc}}(s) = \sum_{s'} H_{ss'} \frac{\psi(s')}{\psi(s)}. \tag{A.3}$$

The gradient is

$$\begin{aligned}
\partial_\theta E = & \frac{\partial_\theta \left( \sum_{s,s'} \psi^*(s') H_{s's} \psi(s) \right)}{\sum_{s''} \psi^*(s'')\psi(s'')} \\
& - \frac{\left( \sum_{s,s'} \psi^*(s') H_{s's} \psi(s) \right) \partial_\theta \left( \sum_{s''} \psi^*(s'')\psi(s'') \right)}{\left( \sum_{s''} \psi^*(s'')\psi(s'') \right)^2} \qquad \text{(A.4)} \\
= & \frac{\sum_{s,s'} H_{s's} \partial_\theta \left( \psi^*(s')\psi(s) \right)}{\sum_{s''} \psi^*(s'')\psi(s'')} \\
& - E \frac{\sum_{s''} \partial_\theta \left( \psi^*(s'')\psi(s'') \right)}{\sum_{s''} \psi^*(s'')\psi(s'')} \qquad \text{(A.5)} \\
= & \frac{1}{\langle \psi | \psi \rangle} \sum_s \left( \left( \sum_{s'} H_{s's} \partial_\theta \left( \psi^*(s')\psi(s) \right) \right) - E \partial_\theta \left( \psi^*(s)\psi(s) \right) \right) \qquad \text{(A.6)}
\end{aligned}$$

(Note that $\frac{1}{\langle \psi | \psi \rangle} \sum_s \partial_\theta \left( \psi^*(s)\psi(s) \right) \neq \partial_\theta \frac{\sum_s \psi^*(s)\psi(s)}{\langle \psi | \psi \rangle}$, because there is no constraint that $\psi$ is normalized)

(Cont'd)

$$
\begin{aligned}
&= \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H_{s's}\psi^*(s')\partial_\theta\psi(s)\right) - E\psi^*(s)\partial_\theta\psi(s)\right) \\
&\quad + \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H_{s's}\psi(s)\partial_\theta\psi^*(s')\right) - E\psi(s)\partial_\theta\psi^*(s)\right) \qquad \text{(A.7)} \\
&= \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H^*_{ss'}\psi^*(s')\partial_\theta\psi(s)\right) - E^*\psi^*(s)\partial_\theta\psi(s)\right)
\end{aligned}
$$

(We used $H_{s's} = H^*_{ss'}$, $E = E^*$)

$$
+ \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H_{ss'}\psi(s')\partial_\theta\psi^*(s)\right) - E\psi(s)\partial_\theta\psi^*(s)\right) \qquad \text{(A.8)}
$$

(We renamed $s \leftarrow s'$, $s' \leftarrow s$ in $\sum_{ss'}$)

$$
\begin{aligned}
&= \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H^*_{ss'}\frac{\psi^*(s')}{\psi^*(s)}\psi^*(s)\psi(s)\frac{\partial_\theta\psi(s)}{\psi(s)}\right) - E^*\psi^*(s)\psi(s)\frac{\partial_\theta\psi(s)}{\psi(s)}\right) \\
&\quad + \frac{1}{\langle\psi|\psi\rangle}\sum_s\left(\left(\sum_{s'} H_{ss'}\frac{\psi(s')}{\psi(s)}\psi^*(s)\psi(s)\frac{\partial_\theta\psi^*(s)}{\psi^*(s)}\right) - E\psi^*(s)\psi(s)\frac{\partial_\theta\psi^*(s)}{\psi^*(s)}\right) \qquad \text{(A.9)} \\
&= \sum_s q(s)\left((E_{\text{loc}}(s) - E)^*\,\partial_\theta \ln\psi(s) + (E_{\text{loc}}(s) - E)\,\partial_\theta \ln\psi^*(s)\right). \qquad \text{(A.10)}
\end{aligned}
$$

# B Details of the SR method

In this appendix, we present the derivation of the SR formula in Eq. (7.24). Substituting Eq. (7.21) into Eqs. (7.22) and (7.23), and ignoring $O(\Delta\tau^2)$ terms, we have

$$\langle\psi|\psi\rangle - \Delta\tau\,\langle\psi|\hat{H}|\psi\rangle = \alpha\,\langle\psi|\psi\rangle + \sum_i \Delta\theta_i\,\langle\psi|\partial_i\psi\rangle\,, \tag{B.1}$$

$$\langle\partial_j\psi|\psi\rangle - \Delta\tau\,\langle\partial_j\psi|\hat{H}|\psi\rangle = \alpha\,\langle\partial_j\psi|\psi\rangle + \sum_i \Delta\theta_i\,\langle\partial_j\psi|\partial_i\psi\rangle\,. \tag{B.2}$$

We define the operator $\hat{D}_i\,|\psi\rangle = \mathrm{diag}(\partial_i \ln\psi)\,|\psi\rangle$, and its adjoint $\langle\psi|\,\hat{D}_i^\dagger = \langle\psi|\,\mathrm{diag}(\partial_i \ln\psi^*)$. It can be used as

$$\frac{\langle\psi|\partial_i\psi\rangle}{\langle\psi|\psi\rangle} = \frac{\sum_{\boldsymbol{s}}\psi^*(\boldsymbol{s})\partial_i\psi(\boldsymbol{s})}{\langle\psi|\psi\rangle} \tag{B.3}$$

$$= \frac{\sum_{\boldsymbol{s}}\psi^*(\boldsymbol{s})\psi(\boldsymbol{s})\partial_i\ln\psi(\boldsymbol{s})}{\langle\psi|\psi\rangle} \tag{B.4}$$

$$= \langle\hat{D}_i\rangle\,. \tag{B.5}$$

Normalizing Eqs. (B.1) and (B.2) by $\langle\psi|\psi\rangle$, we have

$$1 - \Delta\tau\langle\hat{H}\rangle = \alpha + \sum_i \Delta\theta_i\langle\hat{D}_i\rangle, \tag{B.6}$$

$$\langle\hat{D}_j^\dagger\rangle - \Delta\tau\langle\hat{D}_j^\dagger\hat{H}\rangle = \alpha\,\langle\hat{D}_j^\dagger\rangle + \sum_i \Delta\theta_i\,\langle\hat{D}_j^\dagger\hat{D}_i\rangle. \tag{B.7}$$

From Eq. (B.6), we obtain

$$\alpha = 1 - \Delta\tau\,\langle\hat{H}\rangle - \sum_i \Delta\theta_i\,\langle\hat{D}_i\rangle. \tag{B.8}$$

Substituting it into Eq. (B.7),

$$\langle\hat{D}_j^\dagger\rangle - \Delta\tau\langle\hat{D}_j^\dagger\hat{H}\rangle = \langle\hat{D}_j^\dagger\rangle - \Delta\tau\langle\hat{D}_j^\dagger\rangle\langle\hat{H}\rangle + \sum_i \Delta\theta_i\left(\langle\hat{D}_j^\dagger\hat{D}_i\rangle - \langle\hat{D}_j^\dagger\rangle\langle\hat{D}_i\rangle\right) \tag{B.9}$$

$$\Longrightarrow\ -\Delta\tau\left(\langle\hat{D}_j^\dagger\hat{H}\rangle - \langle\hat{D}_j^\dagger\rangle\langle\hat{H}\rangle\right) = \sum_i \Delta\theta_i\left(\langle\hat{D}_j^\dagger\hat{D}_i\rangle - \langle\hat{D}_j^\dagger\rangle\langle\hat{D}_i\rangle\right). \tag{B.10}$$

Using the notations in the main text, we obtain

$$\boldsymbol{S}\Delta\theta = -\frac{\Delta\tau}{2}\boldsymbol{f}, \tag{B.11}$$

where

$$S_{ij} = \langle \hat{D}_i^\dagger \hat{D}_j \rangle - \langle \hat{D}_i^\dagger \rangle \langle \hat{D}_j \rangle, \tag{B.12}$$

$$f_i = 2\left( \langle \hat{D}_i^\dagger \hat{H} \rangle - \langle \hat{D}_i^\dagger \rangle \langle \hat{H} \rangle \right), \tag{B.13}$$

$\Delta\theta$ is a vector of the updates to all the parameters, and $\boldsymbol{f}$ is the gradient vector in Eq. (7.12) in the case of real-valued functions.

# Bibliography


[1] D. Wu, R. Rossi, and G. Carleo. "Unbiased Monte Carlo cluster updates with autoregressive neural networks". In: *Phys. Rev. Res.* 3.4 (2021). DOI: 10.1103/PhysRevResearch.3.L042024.

[2] F. Vicentini et al. "NetKet 3: Machine learning toolbox for many-body quantum systems". In: *SciPost Physics Codebases* (2022). DOI: 10.21468/SciPostPhysCodeb.7.

[3] D. Wu et al. "From tensor-network quantum states to tensorial recurrent neural networks". In: *Phys. Rev. Res.* 5.3 (2023). DOI: 10.1103/PhysRevResearch.5.L032001.

[4] I. Biazzo, D. Wu, and G. Carleo. "Sparse autoregressive neural networks for classical spin systems". In: *Mach. Learn.: Sci. Technol.* 5.2 (2024). DOI: 10.1088/2632-2153/ad5783.

[5] D. Wu et al. "Variational benchmarks for quantum many-body problems". In: *Science* 386.6719 (2024). DOI: 10.1126/science.adg9774.

[6] D. Wu, F. Yang, and G. Carleo. "Unveiling nonmagnetic phase and many-body entanglement in two-dimensional random quantum magnets $Sr_2CuTe_{1-x}W_xO_6$". In: *arXiv Preprint* (2024). arXiv: 2407.05917.

[7] L. D. Landau. "On the theory of phase transitions". In: *Zh. Eksp. Teor. Fiz.* 7 (1937).

[8] D. C. Tsui, H. L. Stormer, and A. C. Gossard. "Two-dimensional magnetotransport in the extreme quantum limit". In: *Phys. Rev. Lett.* 48.22 (1982). DOI: 10.1103/PhysRevLett.48.1559.

[9] X.-G. Wen and Q. Niu. "Ground-state degeneracy of the fractional quantum Hall states in the presence of a random potential and on high-genus Riemann surfaces". In: *Phys. Rev. B* 41.13 (1990). DOI: 10.1103/PhysRevB.41.9377.

[10] T. H. Hansson, V. Oganesyan, and S. L. Sondhi. "Superconductors are topologically ordered". In: *Ann. Phys.* 313.2 (2004). DOI: 10.1016/j.aop.2004.05.006.

[11] M. J. Gilbert. "Topological electronics". In: *Commun. Phys.* 4 (2021). DOI: 10.1038/s42005-021-00569-5.

[12] A. Y. Kitaev. "Fault-tolerant quantum computation by anyons". In: *Ann. Phys.* 303.1 (2003). DOI: 10.1016/S0003-4916(02)00018-0.

[13] L. Balents. "Spin liquids in frustrated magnets". In: *Nature* 464.7286 (2010). DOI: 10.1038/nature08917.

[14] G. Misguich. "Quantum spin liquids and fractionalization". In: *Introduction to Frustrated Magnetism*. 2011. ISBN: 978-3-642-10588-3.

[15] F. Mila. "Frustrated spin systems". In: *Many-Body Physics: From Kondo to Hubbard*. 2015. Chap. 7. ISBN: 978-3-95806-074-6.

[16] P. W. Anderson. "Resonating valence bonds: A new kind of insulator?" In: *Mater. Res. Bull.* 8.2 (1973). DOI: 10.1016/0025-5408(73)90167-0.

[17] Y. Li et al. "Rare-earth triangular lattice spin liquid: A single-crystal study of $YbMgGaO_4$". In: *Phys. Rev. Lett.* 115.16 (2015). DOI: 10.1103/PhysRevLett.115.167203.

[18] M. R. Norman. "Colloquium: Herbertsmithite and the search for the quantum spin liquid". In: *Rev. Mod. Phys.* 88.4 (2016). DOI: 10.1103/RevModPhys.88.041002.

[19] A. Y. Kitaev. "Anyons in an exactly solved model and beyond". In: *Ann. Phys.* 321.1 (2006). DOI: 10.1016/j.aop.2005.10.005.

[20] N. Metropolis and S. Ulam. "The Monte Carlo method". In: *J. Am. Stat. Assoc.* 44.247 (1949). DOI: 10.1080/01621459.1949.10483310.

[21] P. O. J. Scherer. "Computational Physics: Simulation of Classical and Quantum Systems". Springer, 2017. Chap. 24. ISBN: 978-3-319-61088-7.

[22] J. C. Slater. "The theory of complex spectra". In: *Phys. Rev.* 34.10 (1929). DOI: 10.1103/PhysRev.34.1293.

[23] J. P. Bouchaud, A. Georges, and C. Lhuillier. "Pair wave functions for strongly correlated fermions and their determinantal representation". In: *J. Phys. France* 49.4 (1988). DOI: 10.1051/jphys:01988004904055300.

[24] J. J. Hopfield. "Neural networks and physical systems with emergent collective computational abilities". In: *Proc. Natl. Acad. Sci.* 79.8 (1982). DOI: 10.1073/pnas.79.8.2554.

[25] D. J. C. MacKay. "Information Theory, Inference and Learning Algorithms". Cambridge University Press, 2003. Chap. 33. ISBN: 978-0-521-64298-9.

[26] T. Brown et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020.

[27] K. Hornik, M. Stinchcombe, and H. White. "Multilayer feedforward networks are universal approximators". In: *Neural Netw.* 2.5 (1989). DOI: 10.1016/0893-6080(89)90020-8.

[28] E. D. Sontag. "VC dimension of neural networks". In: *Neural Networks and Machine Learning*. 1998. ISBN: 978-3-540-64928-1.

[29] Y. Chen et al. "A survey of accelerator architectures for deep neural networks". In: *Engineering* 6.3 (2020). DOI: 10.1016/j.eng.2020.01.007.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 25. 2012.

[31] I. Goodfellow, Y. Bengio, and A. Courville. "Deep Learning". MIT Press, 2016. ISBN: 978-0-262-03561-3.

[32] F. J. Montáns et al. “Data-driven modeling and learning in science and engineering”. In: *C. R. Mécanique* 347.11 (2019). DOI: 10.1016/j.crme.2019.11.009.

[33] R. Rombach et al. “High-resolution image synthesis with latent diffusion models”. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 38. 2022. DOI: 10.1109/CVPR52688.2022.01042.

[34] D. Silver et al. “Mastering the game of Go with deep neural networks and tree search”. In: *Nature* 529.7587 (2016). DOI: 10.1038/nature16961.

[35] D. Wu, L. Wang, and P. Zhang. “Solving statistical mechanics using variational autoregressive networks”. In: *Phys. Rev. Lett.* 122.8 (2019). DOI: 10.1103/PhysRevLett.122.080602.

[36] E. Ising. “Beitrag zur Theorie des Ferromagnetismus”. In: *Z. Phys.* 31.1 (1925). DOI: 10.1007/BF02980577.

[37] M. Niss. “History of the Lenz–Ising model 1950–1965: From irrelevance to relevance”. In: *Arch. Hist. Exact Sci.* 63.3 (2008). DOI: 10.1007/s00407-008-0039-5.

[38] P. Weiss. “L’hypothèse du champ moléculaire et la propriété ferromagnétique”. In: *J. Phys. Theor. Appl.* 6.1 (1907). DOI: 10.1051/jphystap:019070060066100.

[39] L. Onsager. “Crystal statistics. I. A two-dimensional model with an order-disorder transition”. In: *Phys. Rev.* 65.3–4 (1944). DOI: 10.1103/PhysRev.65.117.

[40] J. Als-Nielsen and R. J. Birgeneau. “Mean field theory, the Ginzburg criterion, and marginal dimensionality of phase transitions”. In: *Am. J. Phys.* 45.6 (1977). DOI: 10.1119/1.11019.

[41] G. M. Viswanathan et al. “What does it take to solve the 3D Ising model? Minimal necessary conditions for a valid solution”. In: *Entropy* 24.11 (2022). DOI: 10.3390/e24111665.

[42] S. Sachdev. “Quantum Phase Transitions”. Cambridge University Press, 2001. ISBN: 978-0-521-00454-1.

[43] J. A. Hertz. “Quantum critical phenomena”. In: *Phys. Rev. B* 14.3 (1976). DOI: 10.1103/PhysRevB.14.1165.

[44] G. Tréglia, B. Legrand, and F. Ducastelle. “Segregation and ordering at surfaces of transition metal alloys: The tight-binding Ising model”. In: *Europhys. Lett.* 7.7 (1988). DOI: 10.1209/0295-5075/7/7/001.

[45] P. M. Chaikin and T. C. Lubensky. “Principles of Condensed Matter Physics”. Cambridge University Press, 1995. Chap. 5. ISBN: 978-0-521-43224-5.

[46] R. J. Baxter. “Solvable models in statistical mechanics, from Onsager onward”. In: *J. Stat. Phys.* 78 (1995). DOI: 10.1007/BF02183337.

[47] N. H. March and G. G. N. Angilella. “Exactly Solvable Models in Many-Body Theory”. World Scientific, 2016. ISBN: 978-981-3140-14-1.

[48] F. Caravelli. "Some exactly solvable and tunable frustrated spin models". In: *Physica A* 594 (2022). DOI: 10.1016/j.physa.2022.127007.

[49] P. M. Kogge and W. J. Dally. "Frontier vs the exascale report: Why so long? And are we really there yet?" In: *IEEE/ACM International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*. Vol. 13. 2022. DOI: 10.1109/PMBS56514.2022.00008.

[50] C. Liu, C.-J. Huang, and G. Chen. "Intrinsic quantum Ising model on a triangular lattice magnet $TmMgGaO_4$". In: *Phys. Rev. Res.* 2.4 (2020). DOI: 10.1103/PhysRevResearch.2.043013.

[51] M. Wolf and K. D. Schotte. "Ising model with competing next-nearest-neighbour interactions on the kagome lattice". In: *J. Phys. A* 21.9 (1988). DOI: 10.1088/0305-4470/21/9/032.

[52] R. Siddharthan et al. "Ising pyrochlore magnets: Low-temperature properties, 'ice rules,' and beyond". In: *Phys. Rev. Lett.* 83.9 (1999). DOI: 10.1103/PhysRevLett.83.1854.

[53] M. Mambrini, J. Trébosc, and F. Mila. "Residual entropy and spin gap in a one-dimensional frustrated antiferromagnet". In: *Phys. Rev. B* 59.21 (1999). DOI: 10.1103/PhysRevB.59.13806.

[54] L. Vanderstraeten, B. Vanhecke, and F. Verstraete. "Residual entropies for three-dimensional frustrated spin systems with tensor networks". In: *Phys. Rev. E* 98.4 (2018). DOI: 10.1103/PhysRevE.98.042145.

[55] G. H. Wannier. "Antiferromagnetism. The triangular Ising net". In: *Phys. Rev.* 79.2 (1950). DOI: 10.1103/PhysRev.79.357.

[56] G. H. Wannier. "Errata: Antiferromagnetism. The triangular Ising net". In: *Phys. Rev. B* 7.11 (1973). DOI: 10.1103/PhysRevB.7.5017.

[57] R. M. F. Houtappel. "Order-disorder in hexagonal lattices". In: *Physica* 16.5 (1950). DOI: 10.1016/0031-8914(50)90130-3.

[58] K. H. Fischer and J. A. Hertz. "Spin Glasses". Cambridge University Press, 1993. ISBN: 978-0-521-34296-4.

[59] H. Nishimori. "Statistical Physics of Spin Glasses and Information Processing: An Introduction". Clarendon Press, 2001. ISBN: 978-0-19-850940-0.

[60] D. Sherrington and S. Kirkpatrick. "Solvable model of a spin-glass". In: *Phys. Rev. Lett.* 35.26 (1975). DOI: 10.1103/PhysRevLett.35.1792.

[61] S. F. Edwards and P. W. Anderson. "Theory of spin glasses". In: *J. Phys. F* 5.5 (1975). DOI: 10.1088/0305-4608/5/5/017.

[62] G. Parisi. "Infinite number of order parameters for spin-glasses". In: *Phys. Rev. Lett.* 43.23 (1979). DOI: 10.1103/PhysRevLett.43.1754.

[63] J. A. Mydosh. "Spin glasses: Redux: An updated experimental/materials survey". In: *Rep. Prog. Phys.* 78.5 (2015). DOI: 10.1088/0034-4885/78/5/052501.

[64] R. M. Karp. "Reducibility among combinatorial problems". In: *Complexity of Computer Computations*. 1972. DOI: 10.1007/978-1-4684-2001-2_9.

[65] F. Barahona. "On the computational complexity of Ising spin glass models". In: *J. Phys. A* 15.10 (1982). DOI: 10.1088/0305-4470/15/10/028.

[66] A. Galanis, D. Štefankovič, and E. Vigoda. "Inapproximability of the partition function for the antiferromagnetic Ising and hard-core models". In: *Comb. Probab. Comput.* 25.4 (2016). DOI: 10.1017/S0963548315000401.

[67] B. Fefferman, M. Foss-Feig, and A. V. Gorshkov. "Exact sampling hardness of Ising spin models". In: *Phys. Rev. A* 96.3 (2017). DOI: 10.1103/PhysRevA.96.032324.

[68] H. Peters and G. Regts. "Location of zeros for the partition function of the Ising model on bounded degree graphs". In: *J. Lond. Math. Soc.* 101.2 (2020). DOI: 10.1112/jlms.12286.

[69] G. A. Carpenter. "Neural network models for pattern recognition and associative memory". In: *Neural Netw.* 2.4 (1989). DOI: 10.1016/0893-6080(89)90035-X.

[70] D. J. Amit, H. Gutfreund, and H. Sompolinsky. "Spin-glass models of neural networks". In: *Phys. Rev. A* 32.2 (1985). DOI: 10.1103/PhysRevA.32.1007.

[71] F. Hamze et al. "Wishart planted ensemble: A tunably rugged pairwise Ising model with a first-order phase transition". In: *Phys. Rev. E* 101.5 (2020). DOI: 10.1103/PhysRevE.101.052102.

[72] M. Hibat-Allah et al. "Variational neural annealing". In: *Nature Mach. Intell.* 3.11 (2021). DOI: 10.1038/s42256-021-00401-3.

[73] W. Heisenberg. "Zur Theorie des Ferromagnetismus". In: *Z. Phys.* 49.9 (1928). DOI: 10.1007/BF01328601.

[74] C. L. Cleveland and R. Medina. "Obtaining a Heisenberg Hamiltonian from the Hubbard model". In: *Am. J. Phys.* 44.1 (1976). DOI: 10.1119/1.10537.

[75] A. Weiße and H. Fehske. "Exact diagonalization techniques". In: *Computational Many-Particle Physics*. 2008. ISBN: 978-3-540-74685-0.

[76] R. Moessner and J. T. Chalker. "Properties of a classical spin liquid: The Heisenberg pyrochlore antiferromagnet". In: *Phys. Rev. Lett.* 80.13 (1998). DOI: 10.1103/PhysRevLett.80.2929.

[77] E. Dagotto and A. Moreo. "Phase diagram of the frustrated spin-$\frac{1}{2}$ Heisenberg antiferromagnet in two dimensions". In: *Phys. Rev. Lett.* 63.19 (1989). DOI: 10.1103/PhysRevLett.63.2148.

[78] H. J. Schulz, T. A. L. Ziman, and D. Poilblanc. "Magnetic order and disorder in the frustrated quantum Heisenberg antiferromagnet in two dimensions". In: *J. Phys. I* 6.5 (1996). DOI: 10.1051/jp1:1996236.

[79] W.-J. Hu et al. "Direct evidence for a gapless $Z_2$ spin liquid by frustrating Néel antiferromagnetism". In: *Phys. Rev. B* 88.6 (2013). DOI: 10.1103/PhysRevB.88.060402.

[80] W.-Y. Liu et al. "Gapless spin liquid ground state of the spin-1/2 $J_1$-$J_2$ Heisenberg model on square lattices". In: *Phys. Rev. B* 98.24 (2018). DOI: 10.1103/PhysRevB.98.241109.

[81] S. Katsura. "Statistical mechanics of the anisotropic linear Heisenberg model". In: *Phys. Rev.* 127.5 (1962). DOI: 10.1103/PhysRev.127.1508.

[82] P. G. de Gennes. "Collective motions of hydrogen bonds". In: *Solid State Commun.* 1.6 (1963). DOI: 10.1016/0038-1098(63)90212-6.

[83] S. Suzuki, J.-i. Inoue, and B. K. Chakrabarti. "Quantum Ising Phases and Transitions in Transverse Ising Models". Springer, 2013. ISBN: 978-3-642-33038-4.

[84] D. Bitko, T. F. Rosenbaum, and G. Aeppli. "Quantum critical behavior for a model magnet". In: *Phys. Rev. Lett.* 77.5 (1996). DOI: 10.1103/PhysRevLett.77.940.

[85] R. Coldea et al. "Quantum criticality in an Ising chain: Experimental evidence for emergent $E_8$ symmetry". In: *Science* 327.5962 (2010). DOI: 10.1126/science.1180085.

[86] H. W. J. Blöte and Y. Deng. "Cluster Monte Carlo simulation of the transverse Ising model". In: *Phys. Rev. E* 66.6 (2002). DOI: 10.1103/PhysRevE.66.066110.

[87] D. P. Landau and K. Binder. "A Guide to Monte Carlo Simulations in Statistical Physics". Cambridge University Press, 2021. Chap. 1. DOI: 978-1-108-49014-6.

[88] W. Feller. "An extention of the law of the iterated logarithm to variables without variance". In: *J. Math. Mech.* 18.4 (1968). DOI: 10.1512/iumj.1969.18.18027.

[89] W. H. Press et al. "Numerical Recipes". Cambridge University Press, 2007. Chap. 7. ISBN: 978-0-521-88068-8.

[90] G. E. P. Box and M. E. Muller. "A note on the generation of random normal deviates". In: *Ann. Math. Stat.* 29.2 (1958). DOI: 10.1214/aoms/1177706645.

[91] A. E. Gelfand and A. F. M. Smith. "Sampling-based approaches to calculating marginal densities". In: *J. Am. Stat. Assoc.* 85.410 (1990). DOI: 10.1080/01621459.1990.10476213.

[92] C. P. Robert and W. Changye. "Markov chain Monte Carlo methods, a survey with some frequent misunderstandings". In: *arXiv Preprint* (2020). arXiv: 2001.06249.

[93] W. K. Hastings. "Monte Carlo sampling methods using Markov chains and their applications". In: *Biometrika* 57.1 (1970). DOI: 10.2307/2334940.

[94] H. Müller-Krumbhaar and K. Binder. "Dynamic properties of the Monte Carlo method in statistical mechanics". In: *J. Stat. Phys.* 8 (1973). DOI: 10.1007/BF01008440.

[95] V. Ambegaokar and M. Troyer. "Estimating errors reliably in Monte Carlo simulations of the Ehrenfest model". In: *Am. J. Phys.* 78.2 (2010). DOI: 10.1119/1.3247985.

[96] J. Goodman and J. Weare. "Ensemble samplers with affine invariance". In: *Commun. Appl. Math. Comput. Sci.* 5.1 (2010). DOI: 10.2140/camcos.2010.5.65.

[97] N. Wiener. "Generalized harmonic analysis". In: *Acta Math.* 55.1 (1930). DOI: 10.1007/bf02546511.

[98] M. Wallerberger. "Efficient estimation of autocorrelation spectra". In: *arXiv Preprint* (2018). arXiv: 1810.05079.

[99] P. Białas, P. Korcyl, and T. Stebel. "Analysis of autocorrelation times in neural Markov chain Monte Carlo simulations". In: *Phys. Rev. E* 107.1 (2023). DOI: 10.1103/PhysRevE.107.015303.

[100] B. W. Silverman. "Density Estimation for Statistics and Data Analysis". Chapman & Hall, 1986. Chap. 3. ISBN: 978-0-412-24620-3.

[101] J. Goodman and A. D. Sokal. "Multigrid Monte Carlo method. Conceptual foundations". In: *Phys. Rev. D* 40.6 (1989). DOI: 10.1103/PhysRevD.40.2035.

[102] U. Wolff. "Critical slowing down". In: *Nucl. Phys. B Proc. Suppl.* 17 (1990). DOI: 10.1016/0920-5632(90)90224-I.

[103] P. C. Hohenberg and B. I. Halperin. "Theory of dynamic critical phenomena". In: *Rev. Mod. Phys.* 49.3 (1977). DOI: 10.1103/RevModPhys.49.435.

[104] E. Lubetzky and A. Sly. "Critical Ising on the square lattice mixes in polynomial time". In: *Commun. Math. Phys.* 313 (2012). DOI: 10.1007/s00220-012-1460-9.

[105] R. Lopez-Ruiz, H. L. Mancini, and X. Calbet. "A statistical measure of complexity". In: *Phys. Lett. A* 209.5–6 (1995). DOI: 10.1016/0375-9601(95)00867-5.

[106] A. Gelman and D. B. Rubin. "Inference from iterative simulation using multiple sequences". In: *Stat. Sci.* 7.4 (1992). DOI: 10.1214/ss/1177011136.

[107] D. Vats and C. Knudson. "Revisiting the Gelman–Rubin diagnostic". In: *Stat. Sci.* 36.4 (2021). DOI: 10.1214/20-STS812.

[108] A. Lee et al. "On the utility of graphics cards to perform massively parallel simulation of advanced Monte Carlo methods". In: *J. Comput. Graph. Stat.* 19.4 (2010). DOI: 10.1198/jcgs.2010.10039.

[109] T. Kloek and H. K. van Dijk. "Bayesian estimates of equation system parameters: An application of integration by Monte Carlo". In: *Econometrica* 46.1 (1978). DOI: 10.2307/1913641.

[110] M. F. Bugallo et al. "Adaptive importance sampling: The past, the present, and the future". In: *IEEE Signal Process. Mag.* 34.4 (2017). DOI: 10.1109/MSP.2017.2699226.

[111] R. Liesenfeld and J.-F. Richard. "Improving MCMC, using efficient importance sampling". In: *Comput. Stat. Data Anal.* 53.2 (2008). DOI: 10.1016/j.csda.2008.07.028.

[112] I. Schuster and I. Klebanov. "Markov chain importance sampling — a highly efficient estimator for MCMC". In: *J. Comput. Graph. Stat.* 30.2 (2020). DOI: 10.1080/10618600.2020.1826953.

[113] P. J. Rossky, J. D. Doll, and H. L. Friedman. "Brownian dynamics as smart Monte Carlo simulation". In: *J. Chem. Phys.* 69.10 (1978). DOI: 10.1063/1.436415.

[114] S. Duane et al. "Hybrid Monte Carlo". In: *Phys. Lett. B* 195.2 (1987). DOI: 10.1016/0370-2693(87)91197-X.

[115] C. M. Fortuin and P. W. Kasteleyn. "On the random-cluster model: I. Introduction and relation to other models". In: *Physica* 57.4 (1972). DOI: 10.1016/0031-8914(72)90045-6.

[116] P. W. Leung and C. L. Henley. "Percolation properties of the Wolff clusters in planar triangular spin models". In: *Phys. Rev. B* 43.1 (1991). DOI: 10.1103/PhysRevB.43.752.

[117] H. G. Evertz, G. Lana, and M. Marcu. "Cluster algorithm for vertex models". In: *Phys. Rev. Lett.* 70.7 (1993). DOI: 10.1103/PhysRevLett.70.875.

[118] U. Wolff. "Collective Monte Carlo updating for spin systems". In: *Phys. Rev. Lett.* 62.4 (1989). DOI: 10.1103/PhysRevLett.62.361.

[119] K. Binder. "Theory of first-order phase transitions". In: *Rep. Prog. Phys.* 50.7 (1987). DOI: 10.1088/0034-4885/50/7/001.

[120] F. Wang and D. P. Landau. "Efficient, multiple-range random walk algorithm to calculate the density of states". In: *Phys. Rev. Lett.* 86.10 (2001). DOI: 10.1103/PhysRevLett.86.2050.

[121] D. P. Landau and K. Binder. "A Guide to Monte Carlo Simulations in Statistical Physics". Cambridge University Press, 2021. Chap. 5. DOI: 978-1-108-49014-6.

[122] R. E. Belardinelli and V. D. Pereyra. "Wang–Landau algorithm: A theoretical analysis of the saturation of the error". In: *J. Chem. Phys.* 127.18 (2007). DOI: 10.1063/1.2803061.

[123] M. I. Jordan et al. "An introduction to variational methods for graphical models". In: *Mach. Learn.* 37 (1999). DOI: 10.1023/A:1007665907178.

[124] P. M. Chaikin and T. C. Lubensky. "Principles of Condensed Matter Physics". Cambridge University Press, 1995. Chap. 4. ISBN: 978-0-521-43224-5.

[125] L. Zdeborová and F. Krzakala. "Statistical physics of inference: Thresholds and algorithms". In: *Adv. Phys.* 65.5 (2016). DOI: 10.1080/00018732.2016.1211393.

[126] S. Kullback and R. A. Leibler. "On information and sufficiency". In: *Ann. Math. Stat.* 22.1 (1951). DOI: 10.1214/aoms/1177729694.

[127] N. N. Bogolubov Jr. "On model dynamical systems in statistical mechanics". In: *Physica* 32.5 (1966). DOI: 10.1016/0031-8914(66)90024-3.

[128] H. B. Curry. "The method of steepest descent for non-linear minimization problems". In: *Q. Appl. Math.* 2.3 (1944). DOI: 10.1090/qam/10667.

[129] S. Boyd and L. Vandenberghe. "Convex Optimization". Cambridge University Press, 2004. ISBN: 978-0-521-83378-3.

[130] A. Chen and M. Heyl. "Efficient optimization of deep neural quantum states toward machine precision". In: *arXiv Preprint* (2023). arXiv: 2302.01941.

[131] E. J. Michaud, Z. Liu, and M. Tegmark. "Precision machine learning". In: *Entropy* 25.1 (2023). DOI: 10.3390/e25010175.

[132] H. Robbins and S. Monro. "A stochastic approximation method". In: *Ann. Math. Stat.* 22.3 (1951). DOI: 10.1214/aoms/1177729586.

[133] L. Bottou. "On-line learning and stochastic approximations". In: *On-Line Learning in Neural Networks*. 1998. ISBN: 978-0-521-65263-6.

[134] W. Hu et al. "On the diffusion approximation of nonconvex stochastic gradient descent". In: *arXiv Preprint* (2017). arXiv: 1705.07562.

[135] R. Kashyap. "A survey of deep learning optimizers — first and second order methods". In: *arXiv Preprint* (2022). arXiv: 2211.15596.

[136] D. P. Kingma and J. Ba. "Adam: A method for stochastic optimization". In: *International Conference on Learning Representations (ICLR)*. Vol. 3. 2015. arXiv: 1412.6980.

[137] K. Levenberg. "A method for the solution of certain non-linear problems in least squares". In: *Q. Appl. Math.* 2.2 (1944). DOI: 10.1090/qam/10666.

[138] R. Fletcher and C. M. Reeves. "Function minimization by conjugate gradients". In: *Comput. J* 7.2 (1964). DOI: 10.1093/comjnl/7.2.149.

[139] D. C. Liu and J. Nocedal. "On the limited memory BFGS method for large scale optimization". In: *Math. Program.* 45.1 (1989). DOI: 10.1007/BF01589116.

[140] V. Gupta, T. Koren, and Y. Singer. "Shampoo: Preconditioned stochastic tensor optimization". In: *The 35th International Conference on Machine Learning (ICML)*. Vol. 80. 2018.

[141] J. Martens. "New insights and perspectives on the natural gradient method". In: *J. Mach. Learn. Res.* 21.146 (2020).

[142] S.-i. Amari. "Natural gradient works efficiently in learning". In: *Neural Comput.* 10.2 (1998). DOI: 10.1162/089976698300017746.

[143] J. Martens and R. Grosse. "Optimizing neural networks with Kronecker-factored approximate curvature". In: *The 32nd International Conference on Machine Learning (ICML)*. Vol. 37. 2015.

[144] N. Roux, P.-A. Manzagol, and Y. Bengio. "Topmoumoute online natural gradient algorithm". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 20. 2007.

[145] Y. Ollivier. "True asymptotic natural gradient optimization". In: *arXiv Preprint* (2017). arXiv: 1712.08449.

[146] W. Zhang et al. "Ground state search by local and sequential updates of neural network quantum states". In: *Phys. Rev. B* 107.16 (2023). DOI: 10.1103/PhysRevB.107.165149.

[147] Y. Bengio et al. "Greedy layer-wise training of deep networks". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 19. 2006.

[148] A. G. Baydin et al. "Automatic differentiation in machine learning: A survey". In: *J. Mach. Learn. Res.* 18.153 (2018).

[149] R. J. Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Mach. Learn.* 8 (1992). DOI: 10.1007/BF00992696.

[150] R. S. Sutton et al. "Policy gradient methods for reinforcement learning with function approximation". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 12. 1999.

[151] R. A. Fisher. "The detection of linkage with 'dominant' abnormalities". In: *Ann. Eugen.* 6.2 (1935). DOI: 10.1111/j.1469-1809.1935.tb02227.x.

[152] A. Hyvärinen and P. Dayan. "Estimation of non-normalized statistical models by score matching." In: *J. Mach. Learn. Res.* 6.4 (2005).

[153] S. Mohamed et al. "Monte Carlo gradient estimation in machine learning". In: *J. Mach. Learn. Res.* 21.132 (2020).

[154] E. Krieken, J. Tomczak, and A. ten Teije. "Storchastic: A framework for general stochastic automatic differentiation". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 34. 2021.

[155] G. Arya et al. "Automatic differentiation of programs with discrete randomness". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022.

[156] G. Catumba, A. Ramos, and B. Zaldivar. "Stochastic automatic differentiation for Monte Carlo processes". In: *arXiv Preprint* (2023). arXiv: 2307.15406.

[157] R. Ranganath, S. Gerrish, and D. Blei. "Black box variational inference". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 33. 2014.

[158] R. Wan et al. "Neural control variates for Monte Carlo variance reduction". In: *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. Vol. 11907. 2019. DOI: 10.1007/978-3-030-46147-8_32.

[159] C. Tsallis. "Classical statistical approach to anisotropic two dimensional XY-model". In: *Nuovo Cim. B* 34.2 (1976). DOI: 10.1007/BF02728617.

[160] L. M. Castro, A. S. T. Pires, and J. A. Plascak. "Free-energy variational approach for the classical anisotropic XY model in a crystal field". In: *Phys. Rev. B* 75.2 (2007). DOI: 10.1103/PhysRevB.75.024410.

[161] D. C. Carvalho, L. M. Castro, and J. A. Plascak. "A variational approach for the classical anisotropic Heisenberg model in a crystal field". In: *Physica A* 391.4 (2012). DOI: 10.1016/j.physa.2011.04.037.

[162] H. Bethe. "Zur Theorie der Metalle: I. Eigenwerte und Eigenfunktionen der linearen Atomkette". In: *Z. Phys.* 71.3 (1931). DOI: 10.1007/BF01341708.

[163] P. D. Gujrati. "Bethe or Bethe-like lattice calculations are more reliable than conventional mean-field calculations". In: *Phys. Rev. Lett.* 74.5 (1995). DOI: 10.1103/PhysRevLett.74.809.

[164] M. Mézard and G. Parisi. "The Bethe lattice spin glass revisited". In: *Eur. Phys. J. B* 20.2 (2001). DOI: 10.1007/PL00011099.

[165] J. S. Yedidia, W. T. Freeman, and Y. Weiss. "Understanding belief propagation and its generalizations". In: *Exploring Artificial Intelligence in the New Millennium*. 2003. ISBN: 978-1-55860-811-5.

[166] S. Ikeda, T. Tanaka, and S.-i. Amari. "Stochastic reasoning, free energy, and information geometry". In: *Neural Comput.* 16.9 (2004). DOI: 10.1162/0899766041336477.

[167] R. Alkabetz and I. Arad. "Tensor networks contraction and the belief propagation algorithm". In: *Phys. Rev. Res.* 3.2 (2021). DOI: 10.1103/PhysRevResearch.3.023073.

[168] C. Guo, D. Poletti, and I. Arad. "Block belief propagation algorithm for two-dimensional tensor networks". In: *Phys. Rev. B* 108.12 (2023). DOI: 10.1103/PhysRevB.108.125111.

[169] J. Kuck et al. "Belief propagation neural networks". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 33. 2020.

[170] J. M. Mooij and H. J. Kappen. "Sufficient conditions for convergence of the sum–product algorithm". In: *IEEE Trans. Inf. Theory* 53.12 (2007). DOI: 10.1109/TIT.2007.909166.

[171] A. Pelizzola. "Cluster variation method in statistical physics and probabilistic graphical models". In: *J. Phys. A* 38.33 (2005). DOI: 10.1088/0305-4470/38/33/R01.

[172] V. Kunc and J. Kléma. "Three decades of activations: A comprehensive survey of 400 activation functions for neural networks". In: *arXiv Preprint* (2024). arXiv: 2402.09092.

[173] D. Hendrycks and K. Gimpel. "Gaussian error linear units (GELUs)". In: *arXiv Preprint* (2016). arXiv: 1606.08415.

[174] G. Carleo et al. "NetKet: A machine learning toolkit for many-body quantum systems". In: *SoftwareX* (2019). DOI: 10.1016/j.softx.2019.100311.

[175] M. A. Islam, S. Jia, and N. D. B. Bruce. "How much position information do convolutional neural networks encode?" In: *International Conference on Learning Representations (ICLR)*. Vol. 8. 2020.

[176] D. Luo et al. "Gauge equivariant neural networks for quantum lattice gauge theories". In: *Phys. Rev. Lett.* 127.27 (2021). DOI: 10.1103/PhysRevLett.127.276402.

[177] C. Roth and A. H. MacDonald. "Group convolutional neural networks improve quantum state accuracy". In: *arXiv Preprint* (2021). arXiv: 2104.05085.

[178] J. Liu et al. "Self-learning Monte Carlo method". In: *Phys. Rev. B* 95.4 (2017). DOI: 10.1103/PhysRevB.95.041101.

[179] L. Huang and L. Wang. "Accelerated Monte Carlo simulations with restricted Boltzmann machines". In: *Phys. Rev. B* 95.3 (2017). DOI: 10.1103/PhysRevB.95.035105.

[180] J. Song, S. Zhao, and S. Ermon. "A-NICE-MC: Adversarial training for MCMC". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 30. 2017.

[181] T. Müller et al. "Neural importance sampling". In: *ACM Trans. Graphics (ToG)* 38.145 (5 2019). DOI: 10.1145/3341156.

[182] J. C. Bridgeman and C. T. Chubb. "Hand-waving and interpretive dance: An introductory course on tensor networks". In: *J. Phys. A* 50.22 (2017). DOI: 10.1088/1751-8121/aa6dc3.

[183] F. Verstraete et al. "Criticality, the area law, and the computational power of projected entangled pair states". In: *Phys. Rev. Lett.* 96.22 (2006). DOI: 10.1103/PhysRevLett.96.220601.

[184] B. Vanhecke et al. "Solving frustrated Ising models using tensor networks". In: *Phys. Rev. Res.* 3.1 (2021). DOI: 10.1103/PhysRevResearch.3.013041.

[185] B. Lanthier, J. Côté, and S. Kourtis. "Tensor networks for $p$-spin models". In: *arXiv Preprint* (2024). arXiv: 2405.08106.

[186] H. Poon and P. Domingos. "Sum-product networks: A new deep architecture". In: *IEEE International Conference on Computer Vision Workshops (ICCV)*. Vol. 13. 2011. DOI: 10.1109/ICCVW.2011.6130310.

[187] R. Sánchez-Cauce, I. París, and F. J. Díez. "Sum-product networks: A survey". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.7 (2022). DOI: 10.1109/TPAMI.2021.3061898.

[188] Z. Liu et al. "KAN: Kolmogorov–Arnold networks". In: *arXiv Preprint* (2024). arXiv: 2404.19756.

[189] P. Barham and M. Isard. "Machine learning systems are stuck in a rut". In: *Workshop on Hot Topics in Operating Systems (HotOS)*. 2019. DOI: 10.1145/3317550.3321441.

[190] M. Germain et al. "MADE: Masked autoencoder for distribution estimation". In: *The 32nd International Conference on Machine Learning (ICML)*. Vol. 37. 2015.

[191] A. van den Oord et al. "WaveNet: A generative model for raw audio". In: *ISCA Speech Synthesis Workshop*. Vol. 9. 2016.

[192] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu. "Pixel recurrent neural networks". In: *The 33nd International Conference on Machine Learning (ICML)*. Vol. 48. 2016.

[193] S. A. Khandoker, J. M. Abedin, and M. Hibat-Allah. "Supplementing recurrent neural networks with annealing to solve combinatorial optimization problems". In: *Mach. Learn.: Sci. Technol.* 4.1 (2023). DOI: 10.1088/2632-2153/acb895.

[194] P. Białas, P. Korcyl, and T. Stebel. "Hierarchical autoregressive neural networks for statistical systems". In: *Comput. Phys. Commun.* 281 (2022). DOI: 10.1016/j.cpc.2022.108502.

[195] K. Tian et al. "Visual autoregressive modeling: Scalable image generation via next-scale prediction". In: *arXiv Preprint* (2024). arXiv: 2404.02905.

[196] C. Nash et al. "Generating images with sparse representations". In: *The 38th International Conference on Machine Learning (ICML)*. Vol. 139. 2021.

[197] W. Mattar et al. "Wavelets are all you need for autoregressive image generation". In: *arXiv Preprint* (2024). arXiv: 2406.19997.

[198] Y. Roudi, J. Tyrcha, and J. A. Hertz. "Ising model for neural data: Model quality and approximate methods for extracting functional connectivity". In: *Phys. Rev. E* 79.5 (2009). DOI: 10.1103/PhysRevE.79.051915.

[199] V. Sessak and R. Monasson. "Small-correlation expansions for the inverse Ising problem". In: *J. Phys. A* 42.5 (2009). DOI: 10.1088/1751-8113/42/5/055001.

[200] F. Ricci-Tersenghi. "The Bethe approximation for solving the inverse Ising problem: A comparison with other inference methods". In: *J. Stat. Mech.: Theory Exp.* 2012.08 (2012). DOI: 10.1088/1742-5468/2012/08/P08015.

[201] S. Efthymiou, M. J. S. Beach, and R. G. Melko. "Super-resolving the Ising model with convolutional neural networks". In: *Phys. Rev. B* 99.7 (2019). DOI: 10.1103/PhysRevB.99.075113.

[202] K. Mills et al. "Extensive deep neural networks for transferring small scale learning to large scale systems". In: *Chem. Sci.* 10.15 (2019). DOI: 10.1039/C8SC04578J.

[203] R. Rende et al. "Fine-tuning neural network quantum states". In: *arXiv Preprint* (2024). arXiv: 2403.07795.

[204] D. Ron, R. H. Swendsen, and A. Brandt. "Inverse Monte Carlo renormalization group transformations for critical phenomena". In: *Phys. Rev. Lett.* 89.27 (2002). DOI: 10.1103/PhysRevLett.89.275701.

[205] F. Pan et al. "Solving statistical mechanics on sparse graphs with feedback-set variational autoregressive networks". In: *Phys. Rev. E* 103.1 (2021). DOI: 10.1103/PhysRevE.103.012103.

[206] D. Pr. "Analysis of Ising model using neural networks". MA thesis. 2021.

[207] I. Biazzo. "The autoregressive neural network architecture of the Boltzmann distribution of pairwise interacting spins systems". In: *Commun. Phys.* 6 (2023). DOI: 10.1038/s42005-023-01416-5.

[208] A. Steger and N. C. Wormald. "Generating random regular graphs quickly". In: *Comb. Probab. Comput.* 8.4 (1999). DOI: 10.1145/780542.780576.

[209] J. Charfreitag et al. "McSparse: Exact solutions of sparse maximum cut and sparse unconstrained binary quadratic optimization problems". In: *Symposium on Algorithm Engineering and Experiments (ALENEX)*. 2022. DOI: 10.1137/1.9781611977042.5.

[210] I. Dunning, S. Gupta, and J. Silberholz. "What works best when? A systematic evaluation of heuristics for Max-Cut and QUBO". In: *INFORMS J. Comput.* 30.3 (2018). DOI: 10.1287/ijoc.2017.0798.

[211] K. A. Nicoli et al. "Asymptotically unbiased estimation of physical observables with neural samplers". In: *Phys. Rev. E* 101.2 (2020). DOI: 10.1103/PhysRevE.101.023304.

[212] S. Ciarella et al. "Machine-learning-assisted Monte Carlo fails at sampling computationally hard problems". In: *Mach. Learn.: Sci. Technol.* 4.1 (2023). DOI: 10.1088/2632-2153/acbe91.

[213] E. Hoogeboom et al. "Autoregressive diffusion models". In: *International Conference on Learning Representations (ICLR)*. Vol. 10. 2022.

[214] A. Pannatier, E. Courdier, and F. Fleuret. "$\sigma$-GPTs: A new approach to autoregressive models". In: *European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD)*. Vol. 14947. 2024. DOI: 10.1007/978-3-031-70368-3_9.

[215] Q. Yu et al. "Randomized autoregressive visual generation". In: *arXiv Preprint* (2024). arXiv: 2411.00776.

[216] A. Goldberg and J. L. Schwartz. "Integration of the Schrödinger equation in imaginary time. I". In: *J. Comput. Phys.* 1.3 (1967). DOI: 10.1016/0021-9991(67)90049-6.

[217] R. von Mises and H. Pollaczek-Geiringer. "Praktische Verfahren der Gleichungsauflösung". In: *Z. Angew. Math. Mech.* 9.2 (1929). DOI: 10.1002/zamm.19290090206.

[218] C. Lanczos. "An iteration method for the solution of the eigenvalue problem of linear differential and integral operators". In: *J. Res. Natl. Bur. Stand.* 45.4 (1950). DOI: 10.6028/jres.045.026.

[219] A. V. Knyazev. "Toward the optimal preconditioned eigensolver: Locally optimal block preconditioned conjugate gradient method". In: *SIAM J. Sci. Comput.* 23.2 (2001). DOI: 10.1137/S1064827500366124.

[220] A. Stathopoulos and J. R. McCombs. "PRIMME: Preconditioned iterative multimethod eigensolver — methods and software description". In: *ACM Trans. Math. Softw.* 37.2 (2010). DOI: 10.1145/1731022.1731031.

[221] A. Wietek and A. M. Läuchli. "Sublattice coding algorithm and distributed memory parallelization for large-scale exact diagonalizations of quantum many-body systems". In: *Phys. Rev. E* 98.3 (2018). DOI: 10.1103/PhysRevE.98.033309.

[222] M. Suzuki. "Generalized Trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems". In: *Commun. Math. Phys.* 51 (1976). DOI: 10.1007/BF01609348.

[223] J. A. Barker. "A quantum-statistical Monte Carlo method; path integrals with boundary conditions". In: *J. Chem. Phys.* 70.6 (1979). DOI: 10.1063/1.437829.

[224] H. De Raedt and A. Lagendijk. "Monte Carlo simulation of quantum statistical lattice models". In: *Phys. Rep.* 127.4 (1985). DOI: 10.1016/0370-1573(85)90044-4.

[225] S. Baroni and S. Moroni. "Reptation quantum Monte Carlo: A method for unbiased ground-state averages and imaginary-time correlations". In: *Phys. Rev. Lett.* 82.24 (1999). DOI: 10.1103/PhysRevLett.82.4745.

[226] A. W. Sandvik. "Finite-size scaling of the ground-state parameters of the two-dimensional heisenberg model". In: *Phys. Rev. B* 56.18 (1997). DOI: 10.1103/PhysRevB.56.11678.

[227] S. Todo and K. Kato. "Cluster algorithms for general-$S$ quantum spin systems". In: *Phys. Rev. Lett.* 87.4 (2001). DOI: 10.1103/PhysRevLett.87.047203.

[228] E. Y. Loh Jr. et al. "Sign problem in the numerical simulation of many-electron systems". In: *Phys. Rev. B* 41.13 (1990). DOI: 10.1103/PhysRevB.41.9301.

[229] P. Henelius and A. W. Sandvik. "Sign problem in Monte Carlo simulations of frustrated quantum spin systems". In: *Phys. Rev. B* 62.2 (2000). DOI: 10.1103/PhysRevB.62.1102.

[230] S. Bravyi et al. "The complexity of stoquastic local Hamiltonian problems". In: *Quantum Inf. Comput.* 8.5 (2008). DOI: 10.5555/2011772.2011773.

[231] W. Marshall. "Antiferromagnetism". In: *Proc. R. Soc. Lond. A* 232.1188 (1955). DOI: 10.1098/rspa.1955.0200.

[232] M. Troyer and U.-J. Wiese. "Computational complexity and fundamental limitations to fermionic quantum Monte Carlo simulations". In: *Phys. Rev. Lett.* 94.17 (2005). DOI: 10.1103/PhysRevLett.94.170201.

[233] M. H. Kalos, D. Levesque, and L. Verlet. "Helium at zero temperature with hard-sphere and other forces". In: *Phys. Rev. A* 9.5 (1974). DOI: 10.1103/PhysRevA.9.2178.

[234] P. J. Reynolds et al. "Fixed-node quantum Monte Carlo for molecules". In: *J. Chem. Phys.* 77.11 (1982). DOI: 10.1063/1.443766.

[235] F. Becca and S. Sorella. "Quantum Monte Carlo Approaches for Correlated Systems". Cambridge University Press, 2017. Chap. 10. ISBN: 978-1-107-12993-1.

[236] R. Blankenbecler, D. J. Scalapino, and R. L. Sugar. "Monte Carlo calculations of coupled boson-fermion systems. I". In: *Phys. Rev. D* 24.8 (1981). DOI: 10.1103/PhysRevD.24.2278.

[237] S. Zhang and H. Krakauer. "Quantum Monte Carlo method using phase-free random walks with Slater determinants". In: *Phys. Rev. Lett.* 90.13 (2003). DOI: 10.1103/PhysRevLett.90.136401.

[238] S. Sorella. "Wave function optimization in the variational Monte Carlo method". In: *Phys. Rev. B* 71.24 (2005). DOI: 10.1103/PhysRevB.71.241103.

[239] H. Chen et al. "Systematic improvement of neural network quantum states using Lanczos". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 35. 2022.

[240] A. Sinibaldi et al. "Unbiasing time-dependent variational Monte Carlo by projected quantum evolution". In: *Quantum* 7 (2023). DOI: 10.22331/q-2023-10-10-1131.

[241] W. Rudin. "Real and Complex Analysis". McGraw Hill, 1986. Chap. 10. ISBN: 978-0-07-100276-9.

[242] J. Bassey, L. Qian, and X. Li. "A survey of complex-valued neural networks". In: *arXiv Preprint* (2021). arXiv: 2101.12249.

[243] W. Wirtinger. "Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen". In: *Math. Ann.* 97.1 (1927). DOI: 10.1007/BF01447872.

[244] K. Kreutz-Delgado. "The complex gradient operator and the $\mathbb{CR}$-calculus". In: *arXiv Preprint* (2009). arXiv: 0906.4835.

[245] N. Krämer. "A tutorial on automatic differentiation with complex numbers". In: *arXiv Preprint* (2024). arXiv: 2409.06752.

[246] R. Assaraf, M. Caffarel, and A. Khelif. "Diffusion Monte Carlo methods with a fixed number of walkers". In: *Phys. Rev. E* 61.4 (2000). DOI: 10.1103/PhysRevE.61.4566.

[247] S. Sorella. "Green function Monte Carlo with stochastic reconfiguration". In: *Phys. Rev. Lett.* 80.20 (1998). DOI: 10.1103/PhysRevLett.80.4558.

[248] J. Stokes et al. "Quantum natural gradient". In: *Quantum* 4 (2020). DOI: 10.22331/q-2020-05-25-269.

[249] E. Study. "Kürzeste Wege im komplexen Gebiet". In: *Math. Ann.* 60.3 (1905). DOI: 10.1007/BF01457616.

[250] M. V. Berry. "The quantum phase, five years after". In: *Geometric Phases in Physics.* 1989. ISBN: 978-9971-5-0599-8.

[251] A. N. Tikhonov. "On the stability of inverse problems". In: *Proc. USSR Acad. Sci.* 39.5 (1943).

[252] R. Rende et al. "A simple linear algebra identity to optimize large-scale neural network quantum states". In: *Commun. Phys.* 7 (2024). DOI: 10.1038/s42005-024-01732-4.

[253] M. Schmitt and M. Heyl. "Quantum many-body dynamics in two dimensions with artificial neural networks". In: *Phys. Rev. Lett.* 125.10 (2020). DOI: 10.1103/PhysRevLett.125.100503.

[254] G. Goldshlager, N. Abrahamsen, and L. Lin. "A Kaczmarz-inspired approach to accelerate the optimization of neural network wavefunctions". In: *J. Comput. Phys.* 516.1 (2024). DOI: 10.1016/j.jcp.2024.113351.

[255] G. Carleo and M. Troyer. "Solving the quantum many-body problem with artificial neural networks". In: *Science* 355.6325 (2017). DOI: 10.1126/science.aag2302.

[256] K. Choo, T. Neupert, and G. Carleo. "Two-dimensional frustrated $J_1$-$J_2$ model studied with neural network quantum states". In: *Phys. Rev. B* 100.12 (2019). DOI: 10.1103/PhysRevB.100.125124.

[257] M. Bukov, M. Schmitt, and M. Dupont. "Learning the ground state of a non-stoquastic quantum Hamiltonian in a rugged neural network landscape". In: *SciPost Physics* 10.6 (2021). DOI: 10.21468/SciPostPhys.10.6.147.

[258] R. Jastrow. "Many-body problem with strong forces". In: *Phys. Rev.* 98.5 (1955). DOI: 10.1103/PhysRev.98.1479.

[259] D. A. Huse and V. Elser. "Simple variational wave functions for two-dimensional Heisenberg spin-1/2 antiferromagnets". In: *Phys. Rev. Lett.* 60.24 (1988). DOI: 10.1103/PhysRevLett.60.2531.

[260] A. A. Abrikosov. "Electron scattering on magnetic impurities in metals and anomalous resistivity effects". In: *Physics Physique Fizika* 2.1 (1965). DOI: 10.1103/PhysicsPhysiqueFizika.2.5.

[261] M. C. Gutzwiller. "Effect of correlation on the ferromagnetism of transition metals". In: *Phys. Rev. Lett.* 10.5 (1963). DOI: 10.1103/PhysRevLett.10.159.

[262] J. Toulouse and C. J. Umrigar. "Optimization of quantum monte carlo wave functions by energy minimization". In: *J. Chem. Phys.* 126.8 (2007). DOI: 10.1063/1.2437215.

[263] H. Yokoyama and H. Shiba. "Variational Monte-Carlo studies of Hubbard model. I". In: *J. Phys. Soc. Jpn.* 56.4 (1987). DOI: 10.1143/JPSJ.56.1490.

[264] C. Gros, R. Joynt, and T. M. Rice. "Antiferromagnetic correlations in almost-localized Fermi liquids". In: *Phys. Rev. B* 36.1 (1987). DOI: 10.1103/PhysRevB.36.381.

[265] F. Ferrari, F. Becca, and R. Valentí. "Charge-density waves in kagome-lattice extended Hubbard models at the van Hove filling". In: *Phys. Rev. B* 106.8 (2022). DOI: 10.1103/PhysRevB.106.L081107.

[266] C. Gros. "Superconductivity in correlated wave functions". In: *Phys. Rev. B* 38.1 (1988). DOI: 10.1103/PhysRevB.38.931.

[267] D. Tahara and M. Imada. "Variational Monte Carlo method combined with quantum-number projection and multi-variable optimization". In: *J. Phys. Soc. Jpn.* 77.11 (2008). DOI: 10.1143/JPSJ.77.114701.

[268] N. Astrakhantsev et al. "Broken-symmetry ground states of the Heisenberg model on the pyrochlore lattice". In: *Phys. Rev. X* 11.4 (2021). DOI: 10.1103/PhysRevX.11.041021.

[269] A. Cayley. "Sur les déterminants gauches". In: *Journal für die reine und angewandte Mathematik* 38 (1849). DOI: 10.1515/crll.1849.38.93.

[270] F. Becca and S. Sorella. "Quantum Monte Carlo Approaches for Correlated Systems". Cambridge University Press, 2017. Chap. 5. ISBN: 978-1-107-12993-1.

[271] Y. Nomura et al. "Restricted Boltzmann machine learning for solving strongly correlated quantum systems". In: *Phys. Rev. B* 96.20 (2017). DOI: 10.1103/PhysRevB.96.205152.

[272] F. Ferrari, F. Becca, and J. Carrasquilla. "Neural Gutzwiller-projected variational wave functions". In: *Phys. Rev. B* 100.12 (2019). DOI: 10.1103/PhysRevB.100.125131.

[273] Y. Nomura and M. Imada. "Dirac-type nodal spin liquid revealed by refined quantum many-body solver using neural-network wave function, correlation ratio, and level spectroscopy". In: *Phys. Rev. X* 11.3 (2021). DOI: 10.1103/PhysRevX.11.031034.

[274] F. Ferrari. "Static and dynamical properties of frustrated spin models". PhD thesis. 2019.

[275] B. Abdollahi and O. Nasraoui. "Explainable restricted Boltzmann machines for collaborative filtering". In: *arXiv Preprint* (2016). arXiv: 1606.07129.

[276] J. Fernandez-de-Cossio-Diaz, S. Cocco, and R. Monasson. "Disentangling representations in restricted Boltzmann machines without adversaries". In: *Phys. Rev. X* 13.2 (2023). DOI: 10.1103/PhysRevX.13.021003.

[277] G. Carleo, Y. Nomura, and M. Imada. "Constructing exact representations of quantum many-body systems with deep neural networks". In: *Nat. Commun.* 9 (2018). DOI: 10.1038/s41467-018-07520-3.

[278] O. Sharir, A. Shashua, and G. Carleo. "Neural tensor contractions and the expressive power of deep neural quantum states". In: *Phys. Rev. B* 106.20 (2022). DOI: 10.1103/PhysRevB.106.205136.

[279] X. Liang et al. "Solving frustrated quantum many-particle models with convolutional neural networks". In: *Phys. Rev. B* 98.10 (2018). DOI: 10.1103/PhysRevB.98.104426.

[280] T. Vieijra and J. Nys. "Many-body quantum states with exact conservation of non-abelian and lattice symmetries through variational Monte Carlo". In: *Phys. Rev. B* 104.4 (2021). DOI: 10.1103/PhysRevB.104.045123.

[281] O. Sharir et al. "Deep autoregressive models for the efficient variational simulation of many-body quantum systems". In: *Phys. Rev. Lett.* 124.2 (2020). DOI: 10.1103/PhysRevLett.124.020503.

[282] M. Hibat-Allah et al. "Recurrent neural network wave functions". In: *Phys. Rev. Res.* 2.2 (2020). DOI: 10.1103/PhysRevResearch.2.023358.

[283] C. Roth. "Iterative retraining of quantum spin models using recurrent neural networks". In: *arXiv Preprint* (2020). arXiv: 2003.06228.

[284] Y.-H. Zhang and M. Di Ventra. "Transformer quantum state: A multipurpose model for quantum many-body problems". In: *Phys. Rev. B* 107.7 (2023). DOI: 10.1103/PhysRevB.107.075147.

[285] L. L. Viteritti, R. Rende, and F. Becca. "Transformer variational wave functions for frustrated quantum spin systems". In: *Phys. Rev. Lett.* 130.23 (2023). DOI: 10.1103/PhysRevLett.130.236401.

[286] L. L. Viteritti et al. "Transformer wave function for the Shastry–Sutherland model: Emergence of a spin-liquid phase". In: *arXiv Preprint* (2023). arXiv: 2311.16889.

[287] X. Cao, Z. Zhong, and Y. Lu. "Vision transformer neural quantum states for impurity models". In: *arXiv Preprint* (2024). arXiv: 2408.13050.

[288] R. Rende and L. L. Viteritti. "Are queries and keys always relevant? A case study on transformer wave functions". In: *arXiv Preprint* (2024). arXiv: 2405.18874.

[289] G. Torlai et al. "Neural-network quantum state tomography". In: *Nat. Phys.* 14.5 (2018). DOI: 10.1038/s41567-018-0048-5.

[290] J.-Q. Wang et al. "Variational optimization of the amplitude of neural-network quantum many-body ground states". In: *Phys. Rev. B* 109.24 (2024). DOI: 10.1103/PhysRevB.109.245120.

[291] M. Li et al. "Bridging the gap between deep learning and frustrated quantum spin system for extreme-scale simulations on new generation of Sunway supercomputer". In: *IEEE Trans. Parallel Distrib. Syst.* 33.11 (2022). DOI: 10.1109/TPDS.2022.3145163.

[292] C.-Y. Park and M. J. Kastoryano. "Expressive power of complex-valued restricted Boltzmann machines for solving nonstoquastic Hamiltonians". In: *Phys. Rev. B* 106.13 (2022). DOI: 10.1103/PhysRevB.106.134437.

[293] T. Westerhout et al. "Generalization properties of neural network approximations to frustrated magnet ground states". In: *Nat. Commun.* 11 (2020). DOI: 10.1038/s41467-020-15402-w.

[294] A. Szabó and C. Castelnovo. "Neural network wave functions and the sign problem". In: *Phys. Rev. Res.* 2.3 (2020). DOI: 10.1103/PhysRevResearch.2.033075.

[295] C. J. Maddison, A. Mnih, and Y. W. Teh. "The concrete distribution: A continuous relaxation of discrete random variables". In: *International Conference on Learning Representations (ICLR)*. Vol. 5. 2017.

[296] E. Jang, S. Gu, and B. Poole. "Categorical reparameterization with Gumbel-softmax". In: *International Conference on Learning Representations (ICLR)*. Vol. 5. 2017.

[297] G. Tucker et al. "REBAR: Low-variance, unbiased gradient estimates for discrete latent variable models". In: *Advances in Neural Information Processing Systems (NIPS)*. Vol. 30. 2017.

[298] W. Grathwohl et al. "Backpropagation through the void: Optimizing control variates for black-box gradient estimation". In: *International Conference on Learning Representations (ICLR)*. Vol. 6. 2018.

[299] T. Westerhout, M. I. Katsnelson, and A. A. Bagrov. "Many-body quantum sign structures as non-glassy Ising models". In: *Commun. Phys.* 6 (2023). DOI: 10.1038/s42005-023-01388-6.

[300] A. Chen et al. "Neural network evolution strategy for solving quantum sign structures". In: *Phys. Rev. Res.* 4.2 (2022). DOI: 10.1103/PhysRevResearch.4.L022026.

[301] S. R. White. "Density matrix formulation for quantum renormalization groups". In: *Phys. Rev. Lett.* 69.19 (1992). DOI: 10.1103/PhysRevLett.69.2863.

[302] R. Penrose et al. "Applications of negative dimensional tensors". In: *Combinatorial Mathematics and Its Applications.* 1971. ISBN: 978-0-12-743350-9.

[303] R. Orús. "A practical introduction to tensor networks: Matrix product states and projected entangled pair states". In: *Ann. Phys.* 349 (2014). DOI: 10.1016/j.aop.2014.06.013.

[304] M. A. Nielsen and I. L. Chuang. "Quantum Computation and Quantum Information". Cambridge University Press, 2010. Chap. 4. ISBN: 978-1-107-00217-3.

[305] M. Cerezo et al. "Variational quantum algorithms". In: *Nat. Rev. Phys.* 3.9 (2021). DOI: 10.1038/s42254-021-00348-9.

[306] M. C. Bañuls et al. "Tensor networks and their use for lattice gauge theories". In: *The 36th Annual International Symposium on Lattice Field Theory (LATTICE2018)*. Vol. 334. 2019. DOI: 10.22323/1.334.0022.

[307] M. C. Bañuls and K. Cichy. "Review on novel methods for lattice gauge theories". In: *Rep. Prog. Phys.* 83.2 (2020). DOI: 10.1088/1361-6633/ab6311.

[308] A. Perez. "The spin-foam approach to quantum gravity". In: *Living Rev. Relativ.* 16.3 (2013). DOI: 10.12942/lrr-2013-3.

[309] Y.-Z. You, X.-L. Qi, and C. Xu. "Entanglement holographic mapping of many-body localized system by spectrum bifurcation renormalization group". In: *Phys. Rev. B* 93.10 (2016). DOI: 10.1103/PhysRevB.93.104205.

[310] P. Hayden et al. "Holographic duality from random tensor networks". In: *J. High Energy Phys.* 2016.11 (2016). DOI: 10.1007/JHEP11(2016)009.

[311] M. Asaduzzaman, S. Catterall, and J. Unmuth-Yockey. "Tensor network formulation of two-dimensional gravity". In: *Phys. Rev. D* 102.5 (2020). DOI: 10.1103/PhysRevD.102.054510.

[312] I. V. Oseledets. "Tensor-train decomposition". In: *SIAM J. Sci. Comput.* 33.5 (2011). DOI: 10.1137/090752286.

[313] I. P. McCulloch. "Infinite size density matrix renormalization group, revisited". In: *arXiv Preprint* (2008). arXiv: 0804.2509.

[314] G. M. Crosswhite and D. Bacon. "Finite automata for caching in matrix product algorithms". In: *Phys. Rev. A* 78.1 (2008). DOI: 10.1103/PhysRevA.78.012356.

[315] C. Hubig, I. P. McCulloch, and U. Schollwöck. "Generic construction of efficient matrix product operators". In: *Phys. Rev. B* 95.3 (2017). DOI: 10.1103/PhysRevB.95.035129.

[316] S. Paeckel, T. Köhler, and S. R. Manmana. "Automated construction of $U(1)$-invariant matrix-product operators from graph representations". In: *SciPost Physics* 3.5 (2017). DOI: 10.21468/SciPostPhys.3.5.035.

[317] I. Affleck et al. "Rigorous results on valence-bond ground states in antiferromagnets". In: *Phys. Rev. Lett.* 59.7 (1987). DOI: 10.1103/PhysRevLett.59.799.

[318] A. Klumper, A. Schadschneider, and J. Zittartz. "Equivalence and solution of anisotropic spin-1 models and generalized $t$-$J$ fermion models in one dimension". In: *J. Phys. A* 24.16 (1991). DOI: 10.1088/0305-4470/24/16/012.

[319] M. Fannes, B. Nachtergaele, and R. F. Werner. "Finitely correlated states on quantum spin chains". In: *Commun. Math. Phys.* 144 (1992). DOI: 10.1007/BF02099178.

[320] C. K. Majumdar and D. K. Ghosh. "On next-nearest-neighbor interaction in linear chain. I". In: *J. Math. Phys.* 10.8 (1969). DOI: 10.1063/1.1664978.

[321] F. Verstraete, V. Murg, and J. I. Cirac. "Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems". In: *Adv. Phys.* 57.2 (2008). DOI: 10.1080/14789940801912366.

[322] U. Schollwöck. "The density-matrix renormalization group". In: *Rev. Mod. Phys.* 77.1 (2005). DOI: 10.1103/RevModPhys.77.259.

[323] P. Comon, X. Luciani, and A. L. F. de Almeida. "Tensor decompositions, alternating least squares and other tales". In: *J. Chemom.* 23.7–8 (2009). DOI: 10.1002/cem.1236.

[324] S. V. Dolgov and D. V. Savostyanov. "Corrected one-site density matrix renormalization group and alternating minimal energy algorithm". In: *European Conference on Numerical Mathematics and Advanced Applications (ENUMATH)*. Vol. 10. 2013. DOI: 10.1007/978-3-319-10705-9_33.

[325] A. W. Sandvik and G. Vidal. "Variational quantum Monte Carlo simulations with tensor-network states". In: *Phys. Rev. Lett.* 99.22 (2007). DOI: 10.1103/PhysRevLett.99.220602.

[326] W.-Y. Liu et al. "Gradient optimization of finite projected entangled pair states". In: *Phys. Rev. B* 95.19 (2017). DOI: 10.1103/PhysRevB.95.195154.

[327] H.-J. Liao et al. "Differentiable programming tensor networks". In: *Phys. Rev. X* 9.3 (2019). DOI: 10.1103/PhysRevX.9.031041.

[328] A. J. Ferris and G. Vidal. "Perfect sampling with unitary tensor networks". In: *Phys. Rev. B* 85.16 (2012). DOI: 10.1103/PhysRevB.85.165146.

[329] Z.-Y. Han et al. "Unsupervised generative modeling using matrix product states". In: *Phys. Rev. X* 8.3 (2018). DOI: 10.1103/PhysRevX.8.031012.

[330] Z.-Y. Wei, D. Malz, and J. I. Cirac. "Sequential generation of projected entangled-pair states". In: *Phys. Rev. Lett.* 128.1 (2022). DOI: 10.1103/PhysRevLett.128.010607.

[331] G. Vidal. "Efficient classical simulation of slightly entangled quantum computations". In: *Phys. Rev. Lett.* 91.14 (2003). DOI: 10.1103/PhysRevLett.91.147902.

[332] H.-C. Jiang, Z.-Y. Weng, and T. Xiang. "Accurate determination of tensor network state of quantum lattice models in two dimensions". In: *Phys. Rev. Lett.* 101.9 (2008). DOI: 10.1103/PhysRevLett.101.090603.

[333] M. Srednicki. "Entropy and area". In: *Phys. Rev. Lett.* 71.5 (1993). DOI: 10.1103/PhysRevLett.71.666.

[334] M. B. Hastings. "An area law for one-dimensional quantum systems". In: *J. Stat. Mech.: Theory Exp.* 2007.08 (2007). DOI: 10.1088/1742-5468/2007/08/P08024.

[335] J. Eisert, M. Cramer, and M. B. Plenio. "Colloquium: Area laws for the entanglement entropy". In: *Rev. Mod. Phys.* 82.1 (2010). DOI: 10.1103/RevModPhys.82.277.

[336] D. Hilbert. "Über die stetige Abbildung einer Linie auf ein Flächenstück". In: *Math. Ann.* 38 (1891). DOI: 10.1007/BF01199431.

[337] P. Xu and S. Tirthapura. "A lower bound on proximity preservation by space filling curves". In: *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Vol. 26. 2012. DOI: 10.1109/IPDPS.2012.118.

[338] F. Verstraete and J. I. Cirac. "Renormalization algorithms for quantum-many body systems in two and higher dimensions". In: *arXiv Preprint* (2004). arXiv: cond-mat/0407066.

[339] N. Schuch et al. "Resonating valence bond states in the PEPS formalism". In: *Phys. Rev. B* 86.11 (2012). DOI: 10.1103/PhysRevB.86.115108.

[340] N. Schuch, J. I. Cirac, and D. Pérez-García. "PEPS as ground states: Degeneracy and topology". In: *Ann. Phys.* 325.10 (2010). DOI: 10.1016/j.aop.2010.05.008.

[341] N. Schuch et al. "Computational complexity of projected entangled pair states". In: *Phys. Rev. Lett.* 98.14 (2007). DOI: 10.1103/PhysRevLett.98.140506.

[342] E. M. Stoudenmire and S. R. White. "Studying two-dimensional systems with the density matrix renormalization group". In: *Annu. Rev. Condens. Matter Phys.* 3.1 (2012). DOI: 10.1146/annurev-conmatphys-020911-125018.

[343] M. Lubasch, J. I. Cirac, and M.-C. Banuls. "Algorithms for finite projected entangled pair states". In: *Phys. Rev. B* 90.6 (2014). DOI: 10.1103/PhysRevB.90.064425.

[344] L. Vanderstraeten et al. "Variational methods for contracting projected entangled-pair states". In: *Phys. Rev. B* 105.19 (2022). DOI: 10.1103/PhysRevB.105.195140.

[345] A. Jermyn. "Automatic contraction of unstructured tensor networks". In: *SciPost Physics* 8.1 (2020). DOI: 10.21468/SciPostPhys.8.1.005.

[346] F. Pan et al. "Contracting arbitrary tensor networks: General approximate algorithm and applications in graphical models and quantum circuit simulations". In: *Phys. Rev. Lett.* 125.6 (2020). DOI: 10.1103/PhysRevLett.125.060503.

[347] S. Sahu and B. Swingle. "Efficient tensor network simulation of quantum many-body physics on sparse graphs". In: *arXiv Preprint* (2022). arXiv: 2206.04701.

[348] N. Schuch et al. "Simulation of quantum many-body systems with strings of operators and Monte Carlo tensor contractions". In: *Phys. Rev. Lett.* 100.4 (2008). DOI: 10.1103/PhysRevLett.100.040501.

[349] F. Mezzacapo et al. "Ground-state properties of quantum many-body systems: Entangled-plaquette states and variational Monte Carlo". In: *New J. Phys.* 11.8 (2009). DOI: 10.1088/1367-2630/11/8/083026.

[350] L. Vanderstraeten et al. "Gradient methods for variational optimization of projected entangled-pair states". In: *Phys. Rev. B* 94.15 (2016). DOI: 10.1103/PhysRevB.94.155123.

[351] Y.-Y. Shi, L.-M. Duan, and G. Vidal. "Classical simulation of quantum many-body systems with a tree tensor network". In: *Phys. Rev. A* 74.2 (2006). DOI: 10.1103/PhysRevA.74.022320.

[352] P. Silvi et al. "Homogeneous binary trees as ground states of quantum critical Hamiltonians". In: *Phys. Rev. A* 81.6 (2010). DOI: 10.1103/PhysRevA.81.062335.

[353] S. Cheng et al. "Tree tensor networks for generative modeling". In: *Phys. Rev. B* 99.15 (2019). DOI: 10.1103/PhysRevB.99.155131.

[354] T. Felser, S. Notarnicola, and S. Montangero. "Efficient tensor network ansatz for high-dimensional quantum many-body problems". In: *Phys. Rev. Lett.* 126.17 (2021). DOI: 10.1103/PhysRevLett.126.170603.

[355] G. Vidal. "Entanglement renormalization". In: *Phys. Rev. Lett.* 99.22 (2007). DOI: 10.1103/PhysRevLett.99.220405.

[356] X. Qian and M. Qin. "From tree tensor network to multiscale entanglement renormalization ansatz". In: *Phys. Rev. B* 105.20 (2022). DOI: 10.1103/PhysRevB.105.205102.

[357] X. Gao and L.-M. Duan. "Efficient representation of quantum many-body states with deep neural networks". In: *Nat. Commun.* 8 (2017). DOI: 10.1038/s41467-017-00705-2.

[358] I. Glasser et al. "Neural-network quantum states, string-bond states, and chiral topological states". In: *Phys. Rev. X* 8.1 (2018). DOI: 10.1103/PhysRevX.8.011006.

[359] J. Chen et al. "Equivalence of restricted Boltzmann machines and tensor network states". In: *Phys. Rev. B* 97.8 (2018). DOI: 10.1103/PhysRevB.97.085104.

[360] Y. Levine et al. "On the long-term memory of deep recurrent networks". In: *arXiv Preprint* (2017). arXiv: 1710.09431.

[361] Y. Levine et al. "Quantum entanglement in deep learning architectures". In: *Phys. Rev. Lett.* 122.6 (2019). DOI: 10.1103/PhysRevLett.122.065301.

[362] M. Hibat-Allah, R. G. Melko, and J. Carrasquilla. "Supplementing recurrent neural network wave functions with symmetry and annealing to improve accuracy". In: *arXiv Preprint* (2022). arXiv: 2207.14314.

[363] Z. Chen et al. "ANTN: Bridging autoregressive neural networks and tensor networks for quantum many-body simulation". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 36. 2023.

[364] R. Pascanu, T. Mikolov, and Y. Bengio. "On the difficulty of training recurrent neural networks". In: *The 30th International Conference on Machine Learning (ICML)*. Vol. 28. 2013.

[365] L. R. Tucker. "Some mathematical notes on three-mode factor analysis". In: *Psychometrika* 31.3 (1966). DOI: 10.1007/BF02289464.

[366] Y. Kwon, D. M. Ceperley, and R. M. Martin. "Effects of three-body and backflow correlations in the two-dimensional electron gas". In: *Phys. Rev. B* 48.16 (1993). DOI: 10.1103/PhysRevB.48.12037.

[367] M. Imada and T. Kashima. "Path-integral renormalization group method for numerical study of strongly correlated electron systems". In: *J. Phys. Soc. Jpn.* 69.9 (2000). DOI: 10.1143/JPSJ.69.2723.

[368] S. Sorella. "Generalized Lanczos algorithm for variatonal quantum Monte Carlo". In: *Phys. Rev. B* 64.2 (2001). DOI: 10.1103/PhysRevB.64.024512.

[369] M. Taddei et al. "Iterative backflow renormalization procedure for many-body ground-state wave functions of strongly interacting normal Fermi liquids". In: *Phys. Rev. B* 91.11 (2015). DOI: 10.1103/PhysRevB.91.115106.

[370] Y. M. Park and H. J. Yoo. "Uniqueness and clustering properties of Gibbs states for classical and quantum unbounded spin systems". In: *J. Stat. Phys.* 80 (1995). DOI: 10.1007/BF02178359.

[371] B. Nachtergaele and R. Sims. "Lieb–Robinson bounds and the exponential clustering theorem". In: *Commun. Math. Phys.* 265 (2006). DOI: 10.1007/s00220-006-1556-1.

[372] M. B. Hastings and T. Koma. "Spectral gap and exponential decay of correlations". In: *Commun. Math. Phys.* 265 (2006). DOI: 10.1007/s00220-006-0030-4.

[373] R. Siddharthan and A. Georges. "Square kagome quantum antiferromagnet and the eight-vertex model". In: *Phys. Rev. B* 65.1 (2001). DOI: 10.1103/PhysRevB.65.014417.

[374] N. Astrakhantsev et al. "Pinwheel valence bond crystal ground state of the spin-$\frac{1}{2}$ Heisenberg antiferromagnet on the shuriken lattice". In: *Phys. Rev. B* 104.22 (2021). DOI: 10.1103/PhysRevB.104.L220408.

[375] P. W. Anderson. "Localized magnetic states in metals". In: *Phys. Rev.* 124.1 (1961). DOI: 10.1103/PhysRev.124.41.

[376] J. Kanamori. "Electron correlation and ferromagnetism of transition metals". In: *Prog. Theor. Phys.* 30.3 (1963). DOI: 10.1143/PTP.30.275.

[377] Y. Lu et al. "Natural-orbital impurity solver and projection approach for Green's functions". In: *Phys. Rev. B* 100.11 (2019). DOI: 10.1103/PhysRevB.100.115134.

[378] X. Cao et al. "Tree tensor-network real-time multiorbital impurity solver: Spin-orbit coupling and correlation functions in $Sr_2RuO_4$". In: *Phys. Rev. B* 104.11 (2021). DOI: 10.1103/PhysRevB.104.115119.

[379] D. Wecker, M. B. Hastings, and M. Troyer. "Progress towards practical quantum variational algorithms". In: *Phys. Rev. A* 92.4 (2015). DOI: 10.1103/PhysRevA.92.042303.

[380] A. Peruzzo et al. "A variational eigenvalue solver on a photonic quantum processor". In: *Nat. Commun.* 5 (2014). DOI: 10.1038/ncomms5213.

[381] K. Seki, T. Shirakawa, and S. Yunoki. "Symmetry-adapted variational quantum eigensolver". In: *Phys. Rev. A* 101.5 (2020). DOI: 10.1103/PhysRevA.101.052340.

[382] N. Astrakhantsev et al. "Phenomenological theory of variational quantum ground-state preparation". In: *Phys. Rev. Res.* 5.3 (2023). DOI: 10.1103/PhysRevResearch.5.033225.

[383] S. Sorella. "The phase diagram of the Hubbard model by variational auxiliary field quantum Monte Carlo". In: *arXiv Preprint* (2021). arXiv: 2101.07045.

[384] A. P. Lund, M. J. Bremner, and T. C. Ralph. "Quantum sampling problems, BosonSampling and quantum supremacy". In: *npj Quantum Inf.* 3 (2017). DOI: 10.1038/s41534-017-0018-2.

[385] S. Boixo et al. "Characterizing quantum supremacy in near-term devices". In: *Nat. Phys.* 14.6 (2018). DOI: 10.1038/s41567-018-0124-x.

[386] F. Arute et al. "Quantum supremacy using a programmable superconducting processor". In: *Nature* 574.7779 (2019). DOI: 10.1038/s41586-019-1666-5.

[387] F. Pan, K. Chen, and P. Zhang. "Solving the sampling problem of the Sycamore quantum circuits". In: *Phys. Rev. Lett.* 129.9 (2022). DOI: 10.1103/PhysRevLett.129.090502.

[388] D. Aharonov et al. "A polynomial-time classical algorithm for noisy random circuit sampling". In: *ACM Symposium on Theory of Computing (SOTC)*. Vol. 55. 2023. DOI: 10.1145/3564246.3585234.

[389] X. Gao et al. "Limitations of linear cross-entropy as a measure for quantum advantage". In: *PRX Quantum* 5.1 (2024). DOI: 10.1103/PRXQuantum.5.010334.

[390] X.-H. Zhao et al. "Leapfrogging Sycamore: Harnessing 1432 GPUs for 7× faster quantum random circuit sampling". In: *Natl. Sci. Rev.* (2024). DOI: 10.1093/nsr/nwae317.

[391] A. J. Daley et al. "Practical quantum advantage in quantum simulation". In: *Nature* 607.7920 (2022). DOI: 10.1038/s41586-022-04940-6.

[392] R. P. Feynman. "Simulating physics with computers". In: *Int. J. Theor. Phys.* 21.6–7 (1982). DOI: 10.1007/BF02650179.

[393] T. Sun, Y. Sun, and W. Yin. "On Markov chain gradient descent". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 31. 2018.

[394] G. Ke, D. He, and T.-Y. Liu. "Rethinking positional encoding in language pre-training". In: *International Conference on Learning Representations (ICLR)*. Vol. 9. 2021.

[395] X. Liu et al. "Scaling laws of RoPE-based extrapolation". In: *International Conference on Learning Representations (ICLR)*. Vol. 12. 2024.

[396] B. Peng et al. "Yarn: Efficient context window extension of large language models". In: *International Conference on Learning Representations (ICLR)*. Vol. 12. 2024.

# Dian Wu (吴 典)

PhD in Physics

ORCID: 0000-0003-3888-5003
Primary Email: wdphy16@pku.edu.cn
Google Scholar: vvcfMgkAAAAJ
GitHub: wdphy16


## Research Interests

Computational physics; Many-body systems; Disordered and frustrated magnets; Variational Monte Carlo; Numerical optimization; Machine learning

## Education

- PhD: Computational Quantum Science Laboratory, École polytechnique fédérale de Lausanne (EPFL), supervised by Giuseppe Carleo — Oct. 2020 – Nov. 2024
- Summer research: Condensed Matter Physics Group, University of California San Diego (UCSD), supervised by Yi-Zhuang You — July 2018
- Undergraduate research: Institute of Physics, Chinese Academy of Sciences, supervised by Lei Wang — Sept. 2017 – July 2020
- BSc: School of Physics, Peking University — Sept. 2016 – July 2020
- Early life in Hangzhou, China

## Publications

**Variational Monte Carlo using autoregressive neural networks for many-body systems**

- I. Biazzo, D. Wu, G. Carleo, "Sparse autoregressive neural networks for classical spin systems", Mach. Learn.: Sci. Technol. **5**, 025074 (2024)
- D. Wu, R. Rossi, G. Carleo, "Unbiased Monte Carlo cluster updates with autoregressive neural networks", Phys. Rev. Res. **3**, L042024 (2021)
- D. Wu, L. Wang, P. Zhang, "Solving statistical mechanics using variational autoregressive networks", Phys. Rev. Lett. **122**, 080602 (2019)

**Other computational studies including exact diagonalization and tensor networks**

- D. Wu, R. Rossi, F. Vicentini, *et al.*, "Variational benchmarks for quantum many-body problems", Science **386**, 296 (2024)
- D. Wu, F. Yang, G. Carleo, "Unveiling nonmagnetic phase and many-body entanglement in two-dimensional random quantum magnets $Sr_2CuTe_{1-x}W_xO_6$", arXiv:2407.05917
- D. Wu, R. Rossi, F. Vicentini, G. Carleo, "From tensor-network quantum states to tensorial recurrent neural networks", Phys. Rev. Res. **5**, L032001 (2023)
- F. Vicentini, D. Hofmann, A. Szabó, D. Wu, *et al.*, "NetKet 3: Machine learning toolbox for many-body quantum systems", SciPost Phys. Codeb. **7** (2022)

**Collaboration on machine learning in other fields**

- Z. Zhong, J. An, D. Wu, *et al.*, "A machine learning strategy for enhancing the strength and toughness in metal matrix composites", Int. J. Mech. Sci. **281**, 109550 (2024)
- H.-Y. Hu, D. Wu, Y.-Z. You, B. Olshausen, Y. Chen, "RG-Flow: A hierarchical and explainable flow model based on renormalization group and sparse prior", Mach. Learn.: Sci. Technol. **3**, 035009 (2022)

## Teaching experience

Teaching assistant for: Computational physics; Quantum physics; Computational quantum physics

## Skills

- Programming languages: Python, Julia, C/C++ (daily use); Haskell, Fortran, Mathematica, MATLAB (academic experience); JavaScript, C#, Rust, Lisp, Prolog... (it is not hard to learn a new language after knowing some programming language theory)
- Software frameworks: JAX, PyTorch, TensorFlow, NetKet, ITensor, Qiskit
- Experienced in Linux and high performance computing (HPC) clusters
- Amateur interest in compiler optimization and symbolic computation
- Amateur experience of training and fine-tuning large language and image models

Last update: 2024-11-13