A STUDY OF SPIN GLASSES AND THEIR APPLICATIONS

A Dissertation

by

CHAO FANG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Helmut G. Katzgraber
Co-Chair of Committee,	Winfried Teizer
Committee Members,	Stephen A. Fulling
	Joseph H. Ross
Head of Department,	Grigory V. Rogachev

August 2020

Major Subject: Physics

ABSTRACT

Spin glasses are disordered magnetic systems with frustration. The extremely complicated energy landscapes built from the frustration gives spin glasses a new transition between the paramagnetic phase and spin-glass phase. This new phenomenon triggers scientists' enthusiasm for deeply understanding the physical foundations and mathematical description for disordered systems. Also the use of numerical tools and statistical methods opens a window into an overlapping area between computational physics and information science, which makes an interdisciplinary study necessary and helpful.

This research consists of two themes. The first theme includes the study on the universality of a diluted spin-glass model, the development of algorithms by introducing new tools of artificial intelligence and extending the current algorithms for more general use. First, I demonstrate that convolutional neural networks is a very powerful tool for detecting spin-glass phase with appropriate training. Next, I show that isoenergetic cluster moves can only be effective for graphs with low connectivities due to the relatively small percolation thresholds in topologies. To conclude the first theme, I introduce the research on how the critical exponent of the correction to the correlation length does not change with disorder, which supports the strong universality scenario. The second theme includes the study of the Boolean satisfiability problem through the use of Ising spin form. It show that Metropolis algorithm combined with the Parallel tempering algorithm has an obvious advantage over the current state-of-art algorithms on Boolean maximum satisfiability problem. Then, I introduce how I improve the efficiency of the Boolean satisfiability problem-based membership filter by turning the Boolean satisfiability instance into a not-all-equal form.

DEDICATION

To my parents

ACKNOWLEDGMENTS

First and foremost I would like to express my deepest gratitude to my Ph.D. advisor, Dr. Helmut G. Katzgraber for his guidance, support, and patience. It has been an honor to be Dr. Katzgraber's Ph.D. student at Texas A&M University. Dr. Katzgraber's passion in physics and every single detail in research have always be my motivation to thrive for the best in my research. Dr. Katzgraber also tough me how to communicate and collaborate with other scientists. I appreciate all the time, funding and other effort that Dr. Katzgraber has contributed to make my graduate study productive and meaningful.

I would like to thank the members of my advisory committee, Dr. Stephen A. Fulling, Dr. Joseph H. Ross, and Dr. Winfried Teizer for carefully evaluating my dissertation and providing useful comments. I would like to thank Zheng Zhu, Andrew Ochoa, Amin Barzegar, Katja Biswas, Dilina Perera, Darryl Jacob and Wenlong Wang for being great collaborators and friends. Without their help I could have not finished my Ph.D. program. I also would like to thank Oliver Melchert and Humberto Munoz for offering fruitful discussions.

It is also my pleasure to meet the great professors, staff and graduate students at Texas A&M University. A lot thanks to Dr. Alexy Belyanin, Dr. John C. Hardy, Dr. Donald Naugle, Dr. Ralf Rapp, Dr. Grigory Rogachev and Sherree Kessler for their help during my graduate study.

Last but not least, my parents have been supporting me unconditionally all the time. I would like to give them my deepest gratitude and will thank them for the rest of my life.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supported by a dissertation committee consisting of Dr. Helmut G. Katzgraber, Joseph H. Ross and Winfried Teizer of the Department of Physics and Astronomy and Dr. Stephen A. Fulling of the Department of Mathematics.

A careful reading of this dissertation was performed by Darryl C. Jacob.

The algorithm presented in Section 6 was co-developed with Zheng Zhu. The data analysis in Section 3 was collaborated with Amin Barzeger. The comparison data in Section 4 was provided by Dilina Perera and Zheng Zhu.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

Graduate study was supported in part by the National Science Foundation (Grant No. DMR-1151387). Additional research efforts were supported in part by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via MIT Lincoln Laboratory Air Force Contract No. FA8721-05-C-0002. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

I would also like to acknowledge the Texas A&M Department of Physics and Astronomy. Computational resources were provided by Texas A&M High Performance Research Computing on their Ada and Terra clusters.

NOMENCLATURE

E	Internal energy
U	internal energy per spin
m	Magnetization per spin
N	Number of variables or spins
T	Temperature
β	Inverse temprature 1/T
$\mathcal{Z}$	Partition function
S_i	Ising spin variable
d	Dimension
L	Linear dimension
$\mathcal{H}$	Hamiltonian
J_{ij}	Interaction between spin variable S_i and S_j
M	Magnetization
T_c	Critical temperature
$\langle \cdots \rangle$	Thermal average
$\left[\cdots \right]$	Disorder average
g_m	Binder cumulant
q	Configurational overlap
q_l	Link-overlap
PT	Parallel Tempering
MCS	Monte Carlo Sweeps

MCMC	Markov-chain Monte Carlo
ICM	Isoenergetic Cluster Move
RICM	Restricted Isoenergetic Cluster Move
SAT	Boolean satisfiability problem
MAX-SAT	Boolean maximum satisfiability problem
NAE-SAT	Not all equal satisfiability problem
NAE-SAT filter	NAE-SAT based membership filter
SAT filter	SAT based membership filter
V_4	Unnormalized Binder ratio
NP	Nondeterministic polynomial
$\mathcal{O}$	Observable
k_B	Boltzmann constant
P	Distribution
p	Probability
$\mathcal{P}$	Transition rate
ξ	Correlation length
χ	Susceptibility
τ_{auto}	Autocorrelation time
C	Special heat
PPR	Projection pursuit regression
ANN	Artificial neural networks
FFS	Finite size scaling
CNN	Convolutional neural networks
CNF	Conjunctive normal form

TABLE OF CONTENTS

Page

ABSTRACT ii

DEDICATION iii

ACKNOWLEDGMENTS iv

CONTRIBUTORS AND FUNDING SOURCES v

NOMENCLATURE vi

TABLE OF CONTENTS viii

LIST OF FIGURES x

LIST OF TABLES xii

1. INTRODUCTION 1

2. PRELIMINARIES 6

    2.1 Spin glass 6
    2.2 Phase transitions 8
    2.3 Critical phenomena 10
    2.4 Algorithms for spin glasses 12
        2.4.1 Metroplis algorithm 13
        2.4.2 Parallel tempering 14
        2.4.3 Houdayer cluster algorithm 15
    2.5 Optimization 16

3. DETECTING CRITICAL TEMPERATURE OF A DISORDERED SYSTEM WITH CONVOLUTIONAL NEURAL NETWORKS AND THE IMPACT OF POISONED TRAINING 19

    3.1 Motivation 19
    3.2 Statistical learning and convolutional neural networks 21
    3.3 Previous research: application of fully connected neural networks on ferromagnetic Ising model 23
    3.4 Model 28

3.5 Implementation of convolutional neural networks 29
3.6 Results from a model trained by well-prepared data 33
3.7 Results from a model trained by poisoned data..... 34
3.8 Summary 36

4. RESTRICTED CLUSTER MOVE ON HIGHLY CONNECTED TOPOLOGIES ... 38

4.1 Motivation 38
4.2 Equilibration..... 40
4.3 Isoenergetic cluster move 42
4.4 Restricted cluster move with detailed balance 44
4.5 Restricted cluster move without detailed balance 47
4.6 Summary 48

5. UNIVERSALITY OF TWO-DIMENSIONAL BOND DILUTED ISING MODEL . 50

5.1 Motivation 50
5.2 Model, Observations and Algorithms 52
5.3 Finite size scaling analysis and numerical results 55
5.4 Summary 60

6. APPLICATION OF SPIN GLASSES: BOOLEAN SATISFIABILITY PROBLEM AND BOOLEAN SATISFIABILITY BASED MEMBERSHIP FILTER..... 63

6.1 Application of metropolis algorithm with parallel tempering on Boolean satisfiability problem 63
6.1.1 Boolean satisfiability problem 65
6.1.2 Mapping..... 67
6.1.3 Implementation of metropolis algorithm with parallel tempering on Boolean satisfiability problem 69
6.1.4 Results on maximum Boolean satisfiability problem 71
6.2 Application on not-all-equal-satisfiability-based set membership filter 72
6.2.1 Set membership filter..... 73
6.2.2 Reminder — probabilistic Bloom filters 77
6.2.3 Satisfiability-based set membership filter 78
6.2.4 Summary..... 85

7. CONCLUSION AND OUTLOOK..... 87

REFERENCES 91

LIST OF FIGURES

FIGURE Page

3.1 A schematic of the fully connected neural network used in our simulation 24

3.2 Parameters from different hidden layers in two-dimensional model 26

3.3 Parameters from output layer in two-dimensional model 27

3.4 Classification probabilities from different system sizes and prediction of phase transition of a bimodal distribution by a convolutional neural network trained by the data from Gaussian distribution combined with finite size scaling..... 35

3.5 Classification probabilities for different system sizes of bimodal distribution with mixed labels by probability of 1% 36

3.6 Classification probabilities for different system sizes of bimodal distribution predicted by a convolutional neural networks trained with data from system that is not is equilibrium 37

4.1 Ratios of the cluster sizes over temperatures of the systems in both two-dimensional and three-dimensional with $L = 8$ for Edward-Anderson model with Gaussian disorder..... 44

4.2 Result of simulation performed on three-dimensional Edward-Anderson model of size $L = 8$ with Gaussian distribution 45

4.3 Cluster size and acceptance for RICM on three-dimensional Edward-Anderson model with Gaussian distribution 46

4.4 RICM without the detailed balance under the flipping ratio of 0.167 48

4.5 Comparison on performances of both RICM and ICM on instances produced in the way presented in Ref. [1] 49

5.1 Fitting Eq. (5.10) to the data from Monte Carlo simulation..... 57

5.2 Cumulant ratio V_4 as a function of system size L 60

5.3 Fitting Eq. (5.10) to data from $p = 0.97$ at each trial temperature..... 61

5.4 Fitting Eq. (5.10) to data from $p = 0.99$ at each trial temperature 62

6.1 Time to solution (CPU time) in seconds of our algorithm and CCLS 74

6.2 Time to solution (CPU time) in seconds of our algorithm and DistUP 75

6.3 Time to solution (CPU time) in seconds of our algorithm and Dist1 76

6.4 Time to solution (CPU time) in seconds for our algorithm and CCLS as a function of system size N for random unweighted MAX-SAT instances 77

6.5 Efficiency ξ_{NAE} as a function of false-positive rate p_{NAE} for a k-NAE-SAT filter 83

6.6 Efficiency ξ_{NAE} as a function of false-positive rate p_{NAE} for a k-NAE-SAT filter 84

6.7 FPR p as a function of the number of solutions s using one, or two-hash functions in `MurmurHash3` [2] 85

6.8 Query time as a function of the number of used instances s for a NAE-SAT filter with $k = 5$, $m = 2^{17}$, and $n = 16282$ 86

LIST OF TABLES

TABLE Page

3.1 Parameters for training samples 32

3.2 Parameter for prediction samples..... 32

3.3 Parameters and Architecture for convolutional neural networks..... 33

5.1 Simulation data of V_4 for pure Ising model from each size 57

5.2 Simulation data of V_4 for diluted Ising model with bond concentration of 0.99 from each size..... 59

5.3 Simulation data of V_4 for diluted Ising model with bond concentration of 0.97 from each size..... 59

6.1 Parameters for the different experiments in unweighted MAX-SAT, partial MAX-SAT, weighted partial MAX-SAT 73

6.2 Build time in seconds, memory size in bytes and false-positive rate (FPR) in percent for the k-SAT filter studied in Ref. [3] 82

6.3 Build time in seconds, memory size in bytes and false-positive rate (FPR) in percent for the k-NAE-SAT filter case study using *borealis* 83

1. INTRODUCTION

Spin glasses are models that describe magnetic materials with atomic spins not aligned in a regular pattern because of the co-existence of ferromagnetic and antiferromagnetic bonds [4]. Spin glasses originally offer a simplified dilute solution to the "cusp" in susceptibility that was found in experiments [5]. In the theoretical study of spin glasses, we usually assign a probability distribution to the interaction J_{ij} between each spin pair (ij) to mimic the randomness of the positions of the atoms since it is impossible to simulate the random positions of all the atoms in real materials that usually have countless atoms. As a result, unlike the Ising model, spin orientation in spin glasses is unlikely to be uniform in space because the interactions between spins can randomly be either ferromagnetic and antiferromagnetic. Even at low temperatures, the patterns of the spin configurations are disordered. Over time the spin-glass phase is formed by the "frozen" disordered spin orientations that break ergodicity, which is different from the paramagnetic phase in which ergodicity is preserved during the thermodynamic process. Although from the perspective of material science, spin glasses are not particularly special since they are simply magnetic materials with disorders, in statistical physics spin glasses invoke large interest not only in theoretical studies but also in industrial applications, especially in optimization problems. First of all, in a spin-glass system, a thermodynamic phase transition (usually second-order) can take place starting from a paramagnetic state at a relatively high temperature, which makes the study of spin glasses appealing because the transition from paramagnetic to a new phase implies the existence of a new magnetic state that can be completely different from ferromagnetic or antiferromagnetic matter. A deep understanding of how this disordered glass state differs from the ordered state and how this transition happens poses an important question in condensed matter physics and statistical physics [6]. Secondly, the applications of spin glasses are important in multiple

fields including biology, computer science, neuroscience, and mathematics because it offers a simple playground to conduct different studies in. Since the time when numerical simulation became the primary tool for studying the thermodynamic properties of spin-glass systems, various efficient Monte Carlo algorithms have been adapted to different scenarios. These simulation techniques play significant roles in the scientific research of spin glasses as well as in a wide range of problems that are related to spin glasses, such as optimization problems [7, 8], quantum computing [9, 10, 11, 12, 13], and associative memory [14, 15]. For example, in Boolean variable optimization area spin glasses offer a brand new approach to find the optimal solutions by mapping the original problem into a spin-glass Hamiltonian, which allows physics to study the optimizations. [4, 16, 8, 17, 18].

As mentioned in the previous paragraph, spin glasses have introduced new physical phenomena into condensed matter physics, and Monte Carlo simulation is the most used tool in this area. In the study of disordered systems, the numerical effort required to make systems reach thermodynamic equilibrium is significant because of the existence of complicated free energy landscapes caused by the quenched disorder. Also frustration in spin-glass instances can lead to problems such as metastability, new forms of symmetry breaking, and even some non-equilibration phenomena like memory and rejuvenation effects that are not seen in ordered systems [4, 18]. Although scientists have already developed various efficient Monte Carlo algorithms such as parallel tempering [19], simulated annealing [20], Wang-Landau algorithm [21] and population annealing [22], these algorithms are still not capable of filling the fast-growing demands in larger system sizes and more precise estimations of thermodynamic quantities. In general, to help push research forward there are two approaches to develop new algorithms. First of all, we can modify existing algorithms to make them adapt to new situations. For example, parallel tempering is a very powerful algorithm for dealing with metastability because its design of exchanging replicas between high and low temperatures can effectively avoid the system from getting stuck in local minima of the complex en-

ergy landscape during simulations. But when the system temperature is low so that it is very likely for the system to have a large local cluster, parallel tempering is not very efficient. In addition to parallel tempering, a new cluster move based on the Houdayer cluster move helps improve the performance in topologies with low percolation thresholds [23]. Furthermore, we can continue upgrading this algorithm to make it work for the highly connected graphs. Besides, we can also adapt algorithms from outside of physics to improve the efficiency of calculation in order to save computational resources. For example, to simulate a medium-size three-dimensional Edward-Anderson model with bimodal distribution, usually we need 20000 instances to cover the disorder and run 300000 Monte Carlo sweep on each instance [24], which is already a heavy burden for computing not to mention that much larger system sizes are needed if we target on more accurate results. In situations like this, we can borrow ideas from other disciplines such as machine learning using statistical approaches to reduce the simulation load and process a large number of data from simulations. Recently machine learning algorithms, in particular deep learning algorithms, have been widely used in condensed matter physics in detecting the phase transitions of physical systems [25, 26, 27]. It has already been shown that well designed convolutional neural networks are capable of recognizing the difference between ferromagnetic states and antiferromagnetic states with high accuracy in an ordered Ising model if the networks are trained by the same set of data in advance [27]. This is an intuitive idea that can lead real applications of machine learning in disordered systems. Usually for an ordered system like the Ising model, traditional Monte Carlo simulation is efficient enough, so it is not desirable to develop new approaches. Also, it is usually impossible to know the phase transition temperature in advance when we are working on unknown systems. Inspired by previous work [27, 25], a proposed systematic machine learning approach with the ability of precisely detecting phase transitions of disordered systems without knowing any information in advance can be a promising tool for the research of disordered systems. We have obtained constructive results int this area, which

will be introduced in Chapter. 3.

The value of spin glasses is in its contributions to physics and real-world applications. In complexity theory, there is a class of problems defined as P problems [7], which can be solved by polynomial-time decision algorithms. Another class, Nondeterministic Polynomials (NP) [28, 29, 30, 16], can only be checked for the correctness of a given solution in polynomial time. NP-complete problems is an important subclass of NP problems, which can be introduced as follows: if A is NP-complete then any solution to A can be mapped in polynomial time as a solution to any of the NP problems. Mathematically, the decision form of the Ising spin model is NP-complete [31, 16] while spin glasses are known as NP-hard problems [32], which is a class of optimization problems such as finding a ground in an Ising-spin formation. Naturally, we consider that if there are connections between spin glasses and any other NP-problems, we might be able to use algorithms for spin glasses to solve these optimization problems. In fact, the connections between statistical physics and NP problems have already caught scientists' eyes [33, 17] and some algorithms developed for spin glasses, like simulated annealing, have already been used as approximate algorithms in classical computers [16]. More than that, the ruggedness of the energy landscapes of spin glasses has already been used to explain the hardness of the NP-problems through the connections between them [34]. Inspired by these previous works, we apply the Metropolis algorithm combined with the parallel tempering algorithm on the Boolean satisfiability (SAT) problem by mapping the original problem into a Hamiltonian in Ising spin form. During simulating the thermodynamics governed by the Hamiltonian, we record the ground states we have found, then map these ground states back to the optimal solutions for the original problem. This application is very successful and inspiring. I will discuss this project in Chapter. 6.

This dissertation can be separated into two main parts. The first part one is about the research of spin glasses, including its physical properties and development of algorithms used to study them. After an introduction of spin glasses, I discuss the application of machine

learning in detecting the phase transition in a three-dimensional Edward-Anderson model with different disorders. Next, I will discuss my research on extending the isoenergetic cluster moves (ICM) to a restricted cluster moves (RICM), followed by the study of the universality of the two-dimensional bond-diluted Ising model. The second part is the application of spin glasses in solving the Boolean satisfiability (SAT) problem, which includes further exploration of the applications on the SAT filter. In conclusion I summarize this work.

2. PRELIMINARIES

2.1 Spin glass

Let us start with the Ising model, which is named after the physicist Ernst Ising [35, 36, 37]. Ising model is a simple mathematical model describing many-body interactions between spins that can only take binary values: $S = \pm 1$. In physics, the Ising spin S can represent the microscopic magnetic moment is either pointing up or pointing down. Usually, an Ising spin system has many spins that are distributed in a background topology, which we call lattice while there is an interaction between some specific spin pair, whose strength can be described by a bond J. To label the sites where the spins sit we introduce subscripts. For example, if we have N spins then we can label spins as S_i with $i \in \{1, ..., N\}$. In this way, we assign an interaction to a spin pair by using the bond between them: $-J$. Then the interaction energy is $-J$ when $S_i = S_j$ and J when $S_i = -S_j$. Thus, if $J > 0$ then two spins tend to point to the same direction because lower energy is more stable than high energy, and in this case the positive bonds can lead to a macroscopic magnetism, so the $J > 0$ is referred to as ferromagnetic interaction while $J < 0$ has an opposite situation and is referred to as antiferromagnetic interaction. Besides, in some cases, a spin has its energy or Zeeman energy in magnetism $-hS_i$. Then the total energy of an Ising model can be described by the following Hamiltonian:

$$\mathcal{H} = -J \sum_{\langle ij \rangle} S_i S_j - h \sum_{i=1}^{N} S_i, \; ij = 1, ..., N, \tag{2.1}$$

since the interaction in the Ising model is constant we do not need subscript for J and the summation goes over all the spin pairs without repeat. The choice of the set for bonds depends on the problem we are looking at. For example, if we have two-dimensional lattice

with all the spins sitting on sites with regular intervals and the interactions are limited to the nearest spins, then in Eq. (2.1) the first summation only runs over the nearest neighbor bonds. And the thermodynamic average $\langle \cdots \rangle$ of a physical observable $\mathcal{O}$ can be computed by the trace over partition function $\mathcal{Z}$ using Gibbs-Boltzmann distribution $\mathcal{S}$:

$$\langle \mathcal{O} \rangle = \sum_{\mathcal{S}} \mathcal{O}(\mathcal{S}) \frac{1}{\mathcal{Z}} \exp\left[-\beta \mathcal{H}(\mathcal{S})\right] , \tag{2.2}$$

with $\beta = 1/k_B T$.

Ising models and spin glasses have some similarities such as both are using binary variables and the interactions can be described by bonds, but Spin glasses are different from Ising model especially in the choice of bonds. As we mentioned in introduction, each bond in spin-glass model obeys a certain distribution, which means that both ferromagnetic and antiferromagnetic interactions can coexist in the same spin-glass model. We suppose that the Hamiltonian without field is expressed as:

$$\mathcal{H} = -\sum_{\langle ij \rangle} J_{ij} S_i S_j, \; ij = 1, ..., N, \tag{2.3}$$

here the summationgoes over all the spin pairs $\langle ij \rangle$ without repeat and J_{ij} can be considered as Independent and identically distributed random variables. Usually the randomness of J_{ij} depends on the specific problem. Similar to Ising model, the set of bonds are different for different models. For example the often used Edward-Anderson model (EA) [38] has the nearest neighbor interactions so that the summation in Eq. (2.3) is only performed in nearest neighbor spin pairs. For the randomness of J_{ij}, the often used distributions $P(J_{ij})$ include Gaussian distribution and random $\pm J$ distribution. Their explicit forms are:

$$P(J_{ij}) = \frac{1}{\sqrt{2\pi J^2}} \exp - \frac{(J_{ij} - J_0)^2}{2J^2} \tag{2.4}$$

$$P(J_{ij}) = p\delta(J_{ij} - J) + (1 - p)\delta(J_{ij} + J), \tag{2.5}$$

where Eq. (2.4) is a Gaussian distribution with mean J_0 and variance J^2 while in Eq. (2.5) J_{ij} is either J with probability p or $-J$ with probability $(1 - p)$. To compute a physical observable, in addition to thermodynamic average $\langle \cdots \rangle$ a disorder average $\big[\cdots \big]$ is required to compute the effect of randomness of interactions, which can be realized by instances with different bonds from the same distribution. For example, the often used configurational overlap between two replicas in a spin-glass model can be written in the form:

$$q = \Big[\langle \sum_{i}^{N} S_i^{\alpha} S_i^{\beta} \rangle \Big], \tag{2.6}$$

where α and β are two replicas in the same system and the average are over thermodynamics $\langle \cdots \rangle$ and disorder $\big[\cdots \big]$.

2.2 Phase transitions

Phase transition is defined as transformation from one state to another in the same thermodynamic system. Generally, a phase is a kind of macroscopic physical state with uniform properties that can be described by a certain set of parameters such as volume, pressure, temperature, etc. Some of these properties have abnormal changes when the phase transition is happening. For example, water and ice are two different states of the same chemical substance. When the temperature drops from above $0°C$ to below $0°C$, water becomes ice, and if the water is heated up to the boiling point $100°C$, water becomes vapor. A similar transition happens in a two-dimensional Ising model whose magnetization suddenly disappears as the temperature exceeds the critical point. Usually, phase transition involves symmetry breaking. We can use some of these physical quantities to quantitatively measure the degree of order across the boundaries during the transition, and these quantities are called order parameters. For example, in the two-dimensional Ising model, the order parameter magneti-

zation changes from a paramagnetic phase to a ferromagnetic phase in the transition, and this phase transition can be seen as the Isotropic symmetry being broken when the temperature drops through the critical temperature.

Although almost all the phase transitions can usually be described as sudden changes in some physical quantities, it can be conventionally divided into two subcategories based on the types of physical quantities that have singularities in the transition [39, 40], which are called first-order phase transition and continues phase transition. This categorization can be based on the behavior of the derivative of the free energy. For example in the first-order phase transition, the first-order derivative of the free energy has a discontinuity, and latent heat is transferred from one phase to another. A typical example of first-order phase transition is the transition from ice to water, in which we can see that during the transition the temperature of the mixture of ice-water remains but heat is flowing from outside into the substance. In continuous phase transition, although there is no latent heat and discontinuity of the first-order derivative of free energy, the second- or higher-order derivative of free energy has divergence. These continue phase transitions are usually characterized by the divergence in susceptibility and a power-law decay in correlations in the vicinity of the critical temperature. An example of the continuous phase transition is the two-dimensional Ising model whose magnetic susceptibility has divergence at the critical temperature. To compute the critical temperature, analytic solutions can only be derived in limited cases such as one-dimensional and two-dimensional Ising models [41, 35]. In principle, any physical quantities can be calculated using Gibbs-Boltzmann Eq. (2.2), but in most situations, there are a large number of spins so that it is almost impossible to calculate the 2^N terms in the partition function. Therefore, approximation methods are used in these situations like mean-filed theory are often used. The idea of the mean-field theory is to approximate the local physical effect by only considering the mean values without the microscopic fluctuation [42]. Although mean-field theory can be used for some spin systems like Sherrington-Kirkpatrick (SK) model [43],

in most of the case mean-field theory can not give qualitative approximations especially for low dimensional systems because mean-field theory tends to treat the spins independently, but only for models with high connectivities the interactions with neighbors can cancel each other so that interactions can be neglected.

2.3 Critical phenomena

Continuous phase transitions always involve abnormal behaviors around the critical points where different phases coexist. These behaviors are called critical phenomena. Let us again take the magnetic materials as an example to explain how we approach the critical phenomena. Suppose that the temperature is originally above the critical point T_c, so the magnetization is zero because spins do not spontaneously point to the same direction at a high temperature $T > T_c$. Then we decrease the temperature letting it approach T_c from above, which makes spins tend to have a similar direction in a relatively large region when T is close to T_c. The scale of these clustered spin will grow into a macroscopic size once T is in the vicinity of T_c. In this case, some physical quantities such as correlation length (ξ) and susceptibility (χ) show singularities. And the degree of singularity of physical quantities near critical point can be described by critical exponents (α, β, γ,δ,...), and experiments show that physical quantities that have singularities will have power-law behaviors as functions of control parameters such as temperature near critical point [44]. Again, we use a magnetic system (Ising model) to demonstrate these power-law relations:

$$\chi \sim t^{-\gamma}, \tag{2.7}$$

$$M \sim |t|^{-\beta}, \tag{2.8}$$

$$\xi \sim |t|^{-\nu}, \tag{2.9}$$

$$C \sim |t|^{-\alpha}, \tag{2.10}$$

where χ is the susceptibility, C is special heat, M is magnetization, ξ is correlation length and t is the reduced temperature that is dimensionless: $t = \frac{(T-T_c)}{T_c}$. Besides, there actually exist additional weaker singularities and regular terms. Therefore, a more accurate expression should include these terms. We take ξ as an example to write the terms:

$$\xi = A|t|^{-\gamma} + B|t|^{-\gamma+1} + ... + const + t + t^2 + ... \tag{2.11}$$

And additionally these Ising critical exponents obey the following relations:

$$\begin{aligned} \alpha + 2\beta + \gamma &= 2 \\ \nu d &= 2 - \alpha, \end{aligned} \tag{2.12}$$

here d is the dimension of the system. Although the degree of freedom of the critical exponents is reduced to 2 by the relation in Eq. (2.12), it is still difficult to compute the critical exponents for systems with infinite size. We introduce scaling hypotheses, which can be used to compute the critical exponents [45]. Again we use Ising model to explain how it works. According to renormalization group theory [46], the finite-size magnetization of a d-dimensional Ising system has a asymptotical form for large system size L:

$$\langle M_L(t) \rangle \sim L^{\beta/\nu} \tilde{M}(tL1/\nu), \tag{2.13}$$

here $\tilde{M}$ is an unknown scaling function, and other quantities have the similar relations. In Eq. (2.13), if we evaluate the function exactly at $T = T_c$ it is only the function of L, therefore in large-L limit we expect the data from different system sizes cross at $T = T_c$, provide we give correct β and ν. But in practice it is difficult to precisely compute these critical

exponents without knowing the true values for the β and ν. Instead, a more practical method would be using a dimensionless quantities such as Binder cumulant or Binder ratio [47]:

$$g_m = \frac{1}{2}\left[3 - \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2}\right] \sim \tilde{G}[L^{1/\nu}(T - T_c)] \,. \tag{2.14}$$

Since Binder ratio is a dimensionless quantity, the pre-factor in Eq. (2.13) will cancel out, therefore asymptotically data from different sizes will have a uniform curve with correct critical exponents. By using this dimensionless quantity the T_c can be determined as well as critical exponents. If two models have the same independent critical exponents, they belong to the same universality class, which is a concept of modern theory of critical phenomena predicted by renormalization group near critical points. Models belonging to the same universality class have the same scaling function and the thermodynamic properties near critical point only depends on dimensionality and symmetry.

2.4 Algorithms for spin glasses

As we mention in Section. 2.1, we need to average the quantity over the states weighted by the Gibbs-Boltzmann distribution Eq. (2.2) to computer a physics quantities, which is difficult for a system with N spins because a complete calculation will have to go over total 2^N different spin configurations. Indeed if we are interested in the thermodynamic limit, the summation is over an infinite number of states which is computationally impossible. Although exact solutions have been found for some models like two-dimensional Ising, it has not been proved possible to analytically solve the majority of the modes of interest. Also, approximate methods like the mean-field theory can only deal with systems with weak interactions, and no general approximate approaches that are suitable for general systems have already been found yet. Therefore, it is necessary and desirable to develop numerical algorithms.

The often used numerical methods in statistical physics is Monte Carlo simulation. The

basic idea behind Monte Carlo is using Markov-chain to directly simulate the thermodynamic procedure, which is a random fluctuation of the system from one state to another over time. Although for past decades, scientists have already developed various efficient Monte Carlo algorithms that can be used in different scenarios, research topics such as critical temperatures, critical exponents and seeking ground states require notoriously large calculation load that can not be completely fulfilled by modern computing facilities. So more powerful algorithms are needed to tackle the computational limitations. This section will give a review of the algorithms used in spin glasses.

2.4.1 Metroplis algorithm

Metropolis algorithm is a beautifully simple and widely used Markov-chain Monte Carlo algorithm introduced by Nicolas Metropolis and his co-workers [48]. The idea behind Metropolis algorithm is to choose the selection a probability $g(\mu \to \nu)$ and an acceptance rate $A(\mu \to \nu)$ so that the transition rate $\mathcal{P}(\mu \to \nu)$ for each states pair fulfill detailed balance:

$$\frac{\mathcal{P}(\mu \to \nu)}{\mathcal{P}(\nu \to \mu)} = \frac{g(\mu \to \nu)A(\mu \to \nu)}{g(\nu \to \mu)A(\nu \to \mu)} = \exp[-\beta(E_\nu - E_\mu)]. \tag{2.15}$$

Assume that we have overall N spins, then for each state μ there are 2^N states that state μ can transfer to and the selection for each ν is equally $\frac{1}{2^N}$. Therefore, Eq. (2.15) becomes:

$$\frac{\mathcal{P}(\mu \to \nu)}{\mathcal{P}(\nu \to \mu)} = \frac{g(\mu \to \nu)A(\mu \to \nu)}{g(\nu \to \mu)A(\nu \to \mu)} = \frac{A(\mu \to \nu)}{A(\nu \to \mu)} = \exp[-\beta(E_\nu - E_\mu)]. \tag{2.16}$$

For the acceptance rate, in order to satisfy Eq. (2.15) a simple choice would be:

$$A(\mu \to \nu) = A_0 \exp[\frac{1}{2}\beta(E_\nu - E_\mu)]. \tag{2.17}$$

Since the constant can be canceled in Eq. (2.15), any value would work for it except that it is not greater than one because it is a probability. Therefore the simplest but workable choice

for $A(\mu \to \nu)$ would be:

$$A(\mu \to \nu) = \begin{cases} \exp[-\beta(E_\nu - E_\mu)] \; E_\nu - E_\mu > 0 \\ \\ 1 \text{ otherwise.} \end{cases} \tag{2.18}$$

It is the Eq. (2.18) that makes it a Metropolis algorithm, which was first invented by Metropolis and his co-workers, and any algorithms that use probability Eq. (2.18) can be said to be Metropolis algorithm. The Metropolis algorithm we introduced here is a single-spin flip algorithm. For Ising models with ferromagnetic interactions, i.e., $J_{ij} = 1$, this works quite well. But when disorder is added, Metropolis is insufficient because the energy landscape can be very complicated with many metastable states. When temperature is low, the dynamics of spin system can be very slow because the acceptance probabilities (Eq. (2.18)) are often exponentially small. Therefore, for complex systems like spin glasses we need better ideas from different perspectives.

2.4.2 Parallel tempering

Parallel tempering is a replica-exchange Markov-chain Monte Carlo algorithm first developed by Hukushima [49]. The basic idea is to swap the replicas from high temperatures and lower temperatures to improve the thermodynamic procedure. As an additional move to Metropolis algorithms or other Monte Carlo algorithms, parallel tempering can efficiently overcome metastable states. In principle, N_T replicas of the system are randomly initialized, and each of these replicas is performed at a range of temperatures $\{T_1, T_2, ..., T_{N_T}\}$. To swap the states between replicas, configurations at different temperatures are exchanged based on the Metropolis criterion, which is

$$\mathcal{P}\left(E_i, T_i \to E_{i+1}, T_{i+1}\right) = \min\{1, \exp\left[\Delta E \Delta\beta\right]\}\,, \tag{2.19}$$

where $\Delta\beta = 1/T_{i+1} - 1/T_i$ is the difference between the inverse temperatures and $\Delta E = E_{i+1} - E_i$ is the difference in the energy of the two replicas. By swapping the configurations between different configurations at high temperatures available to the simulations at low temperatures, and vice versa. This method results in a very robust ensemble that can sample both low and high-temperature configurations.

For parallel tempering, the choice of the temperature set has a significant effect on the performance [50]. Large temperature gaps between replicas will result in low acceptance of swap because replicas with large temperature gaps tend to have fewer states in common and from Eq. (2.19) we can see that the acceptance rate will be low if ΔE is large. If the temperature gaps are small, large number of CPU hours are needed [51]. By performing parallel tempering with other algorithms like Metropolis algorithms, the low-temperature problem can be efficiently resolved for intermediate size systems because the swap between replicas with high and low temperatures can help systems get out of metastable states.

2.4.3 Houdayer cluster algorithm

As mentioned before, pure Metropolis could face the problem of getting stuck in metastable states and parallel tempering can help get the system out of the local minimums. Another tough problem in Monte Carlo is called "critical slowing down", which means that the auto-correlation time is given by:

$$\tau \sim \eta^z, \tag{2.20}$$

with $z \sim 2$. Because when the temperature is at the critical point the correlation length grows as large as system size, then τ will diverge too, which means the system will suffer a severe critical slowing down. In this situation, single spin flipping Monte Carlo simulations do not work well, therefore a cluster update is required.

Houdayer cluster move is a global update algorithm that originally was designed for two-dimensional Edward-Anderson model, and was proved to be able to significantly improve the

thermodynamic procedure so that it is possible to reach much lower temperatures and much large system sizes than as ever been reached [52]. To perform the algorithm, first, we need to introduce two replicas with local overlap denoted by:

$$q_i = S_i^1 S_i^2, \tag{2.21}$$

which defines two domains in the lattice, one with $q_i = -1$ and the other one with $q_i = 1$. The clusters are defined as the connected parts of this domain. By "connected" it means two spins having a bond J_{ij} between them. One cluster move includes that first we randomly pick a site with $q_i = -1$, then find the whole cluster that includes this site and flip the spin on each site in both replicas. By doing this cluster move we can see that in each replica a new configuration is created and the sum of total energies of two replicas stays the same. This property is an important feature to keep the detailed balance for a reject-free cluster move:

$$\mathcal{P}_{\mathcal{C}_1 \to \mathcal{C}_2} = \min(1, \exp[(\beta_2 - \beta_1)(H_2 - H_1)]), \tag{2.22}$$

here $\mathcal{C}$ represents the configuration of the replica, $\beta = 1/kT$ and $\mathcal{H}$ denotes the energy of the replica and the subscript denotes the index of the replica. In this way, the Houdayer cluster move is a reject-free multi-spin flipping Monte Carlo move that keeps the detailed balance. Only Houdayer cluster move is not ergodic, therefore for simulations on thermodynamic systems, a simple way to achieve ergodicity is to add another Markov-chain Monte Carlo such as Metropolis algorithm.

2.5 Optimization

Optimization problems are very common in daily life. For example, when driving to a destination we always want to find the shortest route that can b considered as a "optimal solution" for the trip. Sometimes we need to find the "optimal solution" amongst all the

options that are constrained in a certain range. For example, when driving to work we only get to pick these routes that are suitable for your vehicle. Mathematically, optimization can be described as the problem of finding the best solution from a feasible solutions [53]. In general it can be defined as follows:

$$\begin{aligned} &\min_x f(x), \\ &g_i(x) \leq 0,\ i = 1, ..., m, \end{aligned} \tag{2.23}$$

where $g_i(x)$ is called " constrains". To solve this type of problem we can introduce "Lagrange multipliers" [54], which re-expresses Eq. (2.23) as follows:

$$L_P = f(x) + \sum_{i=1}^{m} \alpha_i g_i(x), i = 1, ..., m, \tag{2.24}$$

where we minimize w.r.t x, α_i, and α_i is a Lagrange multiplier. Based on the properties of function $f(x)$, there are many different methods to find the optimal solution. For example, if $f(x)$ is quadratic and $g_i(x)$ is linear then the problem is convex, which means it has unique optimal solution [55, 56, 57]. In general we can use Karush-Kuhn-Rucker (KKT) [58] approach to solve it.

Similarly, in physics, each system can be described by a Hamiltonian. This function fully describes the behavior of the said system, from a simple ball tossed across a courtyard to a magnetic material or a quantum system. The Hamiltonian describes the dynamics and evaluating the Hamiltonian of a physical system for a given set of parameters and variables results in the energy. In the simplest Ising representation, for a system with N magnetic moments (variables), 2^N possible arrangements of the variables are allowed. Because the number of arrangements grows exponentially fast with the size of the input N, statistical physics, where *ensembles* are studied, is the tool of choice. The study of magnetic systems is thus closely related to constraint optimization problems (COP) [8], because the magnetic

moments can be identified with the variables of a COP and the Hamiltonian with the set constraint-optimization formula. In physics, a system is in its ground-state equilibrium when the energy of a Hamiltonian system is minimal, i.e., the ground state of a magnetic system corresponds directly to the minimum of a COP cost function. The relation between the physical model and optimization problem can be described in Table. tab:opt.

Physics terminology		Optimization terminology
Hamiltonian	$\longrightarrow$	cost function
Ising spin	$\longrightarrow$	variable
spin flip	$\longrightarrow$	variable update
ground state	$\longrightarrow$	minimizing configuration (solution)
Monte Carlo	$\longrightarrow$	stochastic search
cluster	$\longrightarrow$	connected component
temperature	$\longrightarrow$	parameter that influences variable flips

3. DETECTING CRITICAL TEMPERATURE OF A DISORDERED SYSTEM WITH CONVOLUTIONAL NEURAL NETWORKS AND THE IMPACT OF POISONED TRAINING

Machine learning algorithms are statistical methods for learning information from data, which has been studied for decades. Recently machine learning became a hot area for both theoretical research and practical applications due to fast developments in computer science. The field encompasses many methods such as lasso [59, 60] and sparse regression [61, 62], classification and regression trees [63, 64, 65], and boosting and support vector machines [66, 67, 68, 69, 70]. Neural networks [71, 72] are one of the most versatile and powerful tools. They have been well developed and widely used from image recognition to self-driving cars [73, 74]. Recently machine learning algorithms, especially neural network-related algorithms have been applied in studying statistical and condensed matter physics [25, 26, 27, 75, 76, 77, 78, 79]. The advantage of neural network, especially convolutional neural networks [55, 80, 81] is the ability of learning patterns of various graphs. For instance, in [27] the authors use the configurations generated from a two-dimensional Ising model to train a convolutional neural network, then use this network to detect the phase transition of the two-dimensional Ising model. This suggests that the convolutional neural network algorithm might have the ability to detect the phase transition of a system with Boolean variables like the Ising model.

3.1 Motivation

We apply the neural network algorithm on spin-glass systems [4, 18, 82] with disordered magnetic moments aligned in special patterns. Due to the existence of frustration, Monte Carlo simulation usually takes a huge number of CPU hours to reach thermal equilibrium for large system sizes [83]. While Ref. [84] already shows the potential utility of machine

learning and works around the issues imposed by different observables typically measured in spin-glass simulations, here we develop a more reliable and systematic usage of convolutional neural networks with finite size scaling [85, 86] to detect the phase transition of spin-glass systems. We note that similar to the Binder ratio [4, 87, 83] classification probability has finite size effect as well. Furthermore, finite size scaling can be used to extract the critical temperature and critical exponents from the classification probabilities of systems with different sizes.

More importantly, in Refs. [88, 89, 90] the authors show that data poisoning is very crucial in machine learning. For instance, attackers can deliberately influence the training dataset to manipulate the results of a predictive model [91, 92], which means that the results from machine learning algorithms can sensitively rely on the inputs. In our case, the poisoned data can come from the training data by Monte Carlo with incorrect parameters. For example, when we generate training data, we might either mislabel the training data, which can lead to a misclassification on the following prediction stage or use the training data from a system in a bad equilibrium. To well evaluate how poisoned data impact the prediction, we conduct experiments from different angles to demonstrate the importance of training data.

In experiments, we implement neural network architectures using the Tensorflow [93] library in python, which is a widely used deep learning [55, 94, 95, 96] tool developed by Google. First, we calculate the classification probabilities from different system sizes. Then we extract the phase transition temperatures and the corresponding exponent indices using finite size scaling. The results match the ones from Monte Carlo [83] very well. Furthermore, we generate sets of poisoned data from different perspectives to train the neural network, then estimate how the results are affected by the poisoning.

3.2 Statistical learning and convolutional neural networks

Statistical learning, also referred to as machine learning, uses data to approximate and predict trends in a model. Specifically, there are two categories of machine learning algorithms: supervised learning and unsupervised learning [56]. For supervised learning, each data point includes a predictor and a response variable: (x_i, y_i), where x_i could be a vector of p features: $x_i = (x_i^1, x_i^2, ..., x_i^p)$. Depending on whether the response variable y_i is quantitative or qualitative, supervised learning can be subdivided into two subcategories: regression and classification. For regression, the outputs belong to $\mathbb{R}^n, n = 1, 2, ...$, and the goal is to predict the conditional mean, $f(x_i) = E(y_i|x_i)$, which can be proved to have the minimum prediction error as a prediction function [56]. For classification, the outputs are categories, $\hat{G}_i$, and the goal is to assign each input to its corresponding category. Typically, we would pick the category that maximizes the conditional probability, $P(y_i = \hat{G}_i|x_i)$, a method used in the *Bayes classifier* [56]. For the scope of this dissertation, we focus on supervised learning.

First let us look at a simplified model called projection pursuit regression (PPR). Let us assume that we have an input vector X with p components, and a target Y. Let $\omega_m (m = 1, 2, ..., M)$ be unit p-vectors of unknown parameters (M is the number of parameters). The PPR model has the form:

$$f(X) = \sum_{m=1}^{M} g_m(\omega_m^T X). \tag{3.1}$$

By forming nonlinear functions from linear combinations of predictors, we are generating a large class of models. PPR can be stated as having the "general" form of these models. Based on PPR model we can further introduce a toy model, which is also called a "single hidden layer back-propagation network". For a classification problem with K categories, we begin with K units, where the kth unit models the probability that a data point belongs to

class k. There are K target measurements $Y_k(k = 1, ..., K)$ with each serving as Boolean variable for the kth class. The target Y_k is modeled as a function of linear combinations of Z_m, where, as shown in Eq. (3.2), Z_m is created from linear combinations of input predictors:

$$\begin{aligned} Z_m &= \sigma(\alpha_{0m} + \alpha_m^T X),\ m = 1, ..M, \\ T_k &= \beta_{0k} + \beta_k^T Z,\ k = 1, ...K, \\ f_k(X) &= g_k(T),\ k = 1, ..., K, \end{aligned} \tag{3.2}$$

where $Z = (Z_1, Z_2, ..., Z_M)$, and $T = (T_1, T_2, ..., T_K)$. There are numerous options for the activation function $\sigma(v)$. The standard choice would be:

$$\sigma(v) = 1/(1 + e^{-v}), \tag{3.3}$$

because we have finer control over the function's transition from linear to non-linear. But in modern deep learning implementation, *Rectifier Linear Unit(ReLU)* is a better choice because it could avoid the function from saturating [55]:

$$\sigma(x) = \begin{cases} x & \text{if} \quad x > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

The output function $g_k(T)$ allows a final transformation of the output vector T and is usually the *softmax* function, which can make final result in the region of $(0, 1]$:

$$g_k = \frac{e^{T_k}}{\sum_{l=1}^{K} e^{T_l}}. \tag{3.5}$$

The units responsible for computing derived features Z_m are called *hidden units* because they are not directly observed. In real applications, the network typically has more than one

intermediate layer, especially in deep learning models. Since this method can theoretically model any function, the network can be very complex.

For convolutional neural networks (CNN) local convolution rather than global linear transformation is used before nonlinear operations are applied. Local convolutions take into account connections between neighbors, yielding serious benefits if our input is a graph with strong local connections. Assume that our data is a two-dimensional grid of pixels, then if we use $Z^p_{i,j,k}$ to denote data from the pth layer and $K^p_{i,l,m,n}$ to denote the kernel for convolution which provides the connection from the pth layer to the $(p+1)$th layer, we have:

$$Z^{p+1}_{i,j,k} = \sigma(\sum_{l,m,n} Z^p_{l,j+m-1,k+n-1} K_{i,l,m,n}), \tag{3.6}$$

where i is the index for channels in the pth layer while l is the index for the channels in the $p+1$ layer. j and k are indices for rows and columns while m and n are the indices for the convolution.

3.3 Previous research: application of fully connected neural networks on ferromagnetic Ising model

The Hamiltonian of a two-dimensional Ising model in the absence of an external magnetic field is given by [25]

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \sigma^z_i \sigma^z_j \tag{3.7}$$

where $J = \pm 1$ is the coupling constant for ferromagnetic and antiferromagnetic system respectively, and $\sigma^z_i = \pm 1$ is the Ising spin on site i. For a lattice with length L, the number of spins is $N = L^2$ and the number of states is up to 2^L. Using the Monte Carlo method, spin configuration weighted by Boltzmann distribution can be prepared at any temperature on a specific lattice structure. Various observables, such as magnetization and specific heat, can be calculated from these samples to study the thermodynamic properties of the system.

The existence of a phase transition between low-temperature ferromagnetic phase and high-temperature paramagnetic phase [41, 97] makes it possible to classify the two types of phase by using supervised machine learning instead of thermodynamic observables [25].

To set up the machine learning model, the first thing is to prepare training and testing data. We implement a standard Monte Carlo simulation with C to thermalize spin configurations of lattice size $L = 8, 16, 32$, from $T = 1.0$ to $T = 3.5$ with a step $\Delta T = 0.05$, and label the configurations below $T = 2.25$ as low-temperature phase ("1") and those above $T = 2.30$ as high-temperature phase ("0"). Two hundred independent samples are generated at each temperature, so the total sample size is 10200. The data is ready to use after being split into 70% training set and 30% testing set.

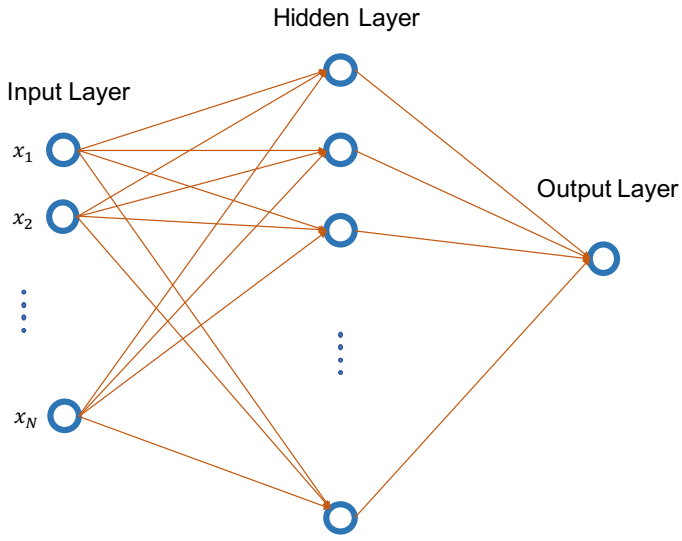

Figure 3.1: A schematic of the fully connected neural network used in our simulation. Data representing spin configurations are fed into the model through the input layer and learned by the hidden layer. The output layer return values indicating how likely an input state can be classified as in the low-temperature phase.

We construct a fully connected neural network with one hidden layer of 100 neurons and an output layer with only one node (Fig. 3.1). The arguments of the hidden layer neurons are

given by

$$\boldsymbol{z_1} = \boldsymbol{W}_1 \boldsymbol{x_1} + \boldsymbol{b}_1 \tag{3.8}$$

where $\boldsymbol{W_1}$ is the weight matrix of size $(100, N)$, $\boldsymbol{b_1}$ is the bias vector of size $(100, 1)$, and $\boldsymbol{x_1} = (\sigma_1^z, \sigma_2^z, \ldots, \sigma_N^z)^{\mathrm{T}}$ is the spin configurations. In our neural network, the activation functions of the hidden layer and output layer are set to be sigmoid function, which can be written as

$$S(x) = \frac{1}{1 + e^{-x}} \tag{3.9}$$

It follows that the arguments of the output layer can be found in a similar manner

$$\boldsymbol{z_2} = \boldsymbol{W}_2 \boldsymbol{x_2} + \boldsymbol{b}_2 \tag{3.10}$$

where $\boldsymbol{W_2}$ and $\boldsymbol{b_2}$ are the weights and bias respectively, and $\boldsymbol{x}_2 = S(\boldsymbol{z_1})$ is the output of the hidden layer. The final result is given by

$$y = S(\boldsymbol{z}_2) \tag{3.11}$$

which is an indicator of how likely a particular spin configuration is in low-temperature phase. The neural network is implemented by using TensorFlow [98] with Keras library. To prevent overfitting, we use the binary cross-entropy cost function with L2 regularization in the training.

A wide range of spin configurations on a square lattice at temperatures below and above T_c are prepared by Monte Carlo simulation, without labeling which phase they are in, to evaluate the performance of our neural network. The evaluation data sets have never been fed into the model during training and testing. A thousand evaluation samples are generated at each temperature to do statistics. As shown in Fig. 3.2(a), our neural network can correctly identify the phase of a given spin configuration at temperature T from the evaluation

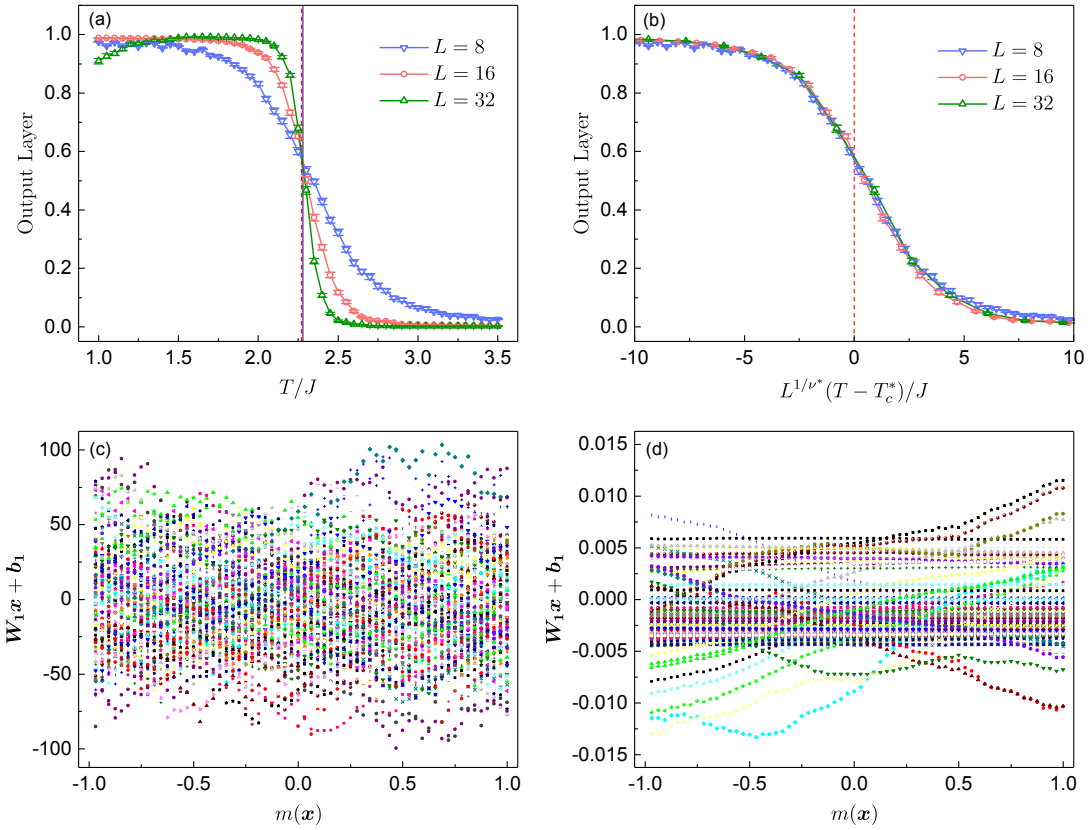

Figure 3.2: Parameters from different hidden layers in two-dimensional model. (a) The output layer averaged over a set of evaluation data as a function of T/J, for square lattice of size $L = 8, 16, 32$. The vertical dash line marks the critical temperature in thermodynamic limit, $T_c/J = 2/\ln(1+\sqrt{2})$, and the vertical solid line is the estimated location of the crossover point for systems of different size. (b) Finite-size scaling of the averaged output layer as a function of $L^{1/\nu^*}(T - T_c^*)/J$, where $\nu^* = 0.98$ and $T_c^*/J = 2.273$. Data from lattice size $L = 8, 16, 32$ collapse to one curve. (c) and (d) Hidden layer arguments as a function of the magnetization of manually generated spin configurations with linear increasing magnetization, before and after training, respectively.

data set. Finite-size scaling with estimated $\nu^* = 0.98$ and $T_c^*/J = 2.273$ illustrates the collapse of data for different lattice size, which is a characteristic feature of second-order phase transition (Fig. 3.2(b)). The estimated T_c^*/J, and ν^* are close to the theoretical value $T_c/J = 2/\ln(1+\sqrt{2})$ and $\nu = 1$. This result also suggests the predictions made by our

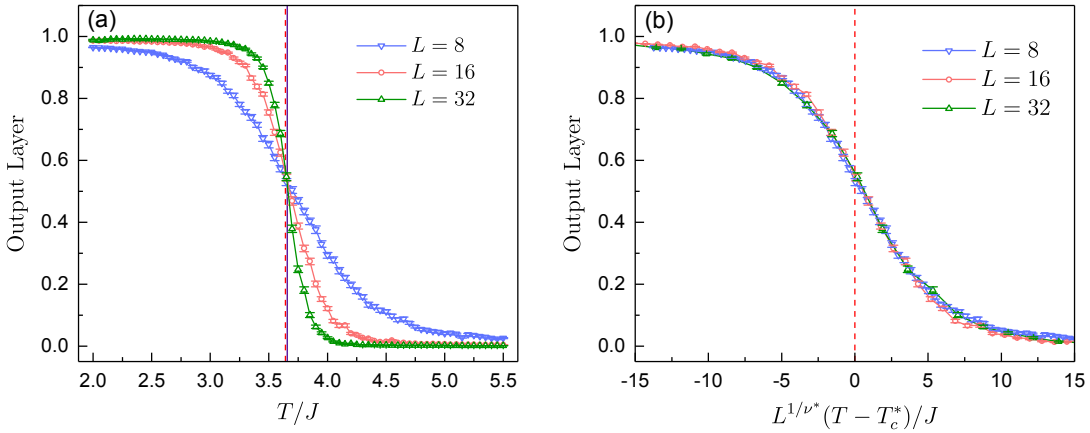

Figure 3.3: Parameters from output layer in two-dimensional model. (a) The output layer averaged over a set of evaluation data as a function of T/J, for triangular lattice of size $L = 8, 16, 32$. The vertical dash line marks the critical temperature in thermodynamic limit, $T_c/J = 4/\ln 3$, and the vertical solid line is the estimated location of the crossover point for systems of different size. (b) Finite-size scaling of the averaged output layer as a function of $L^{1/\nu^*}(T - T_c^*)/J$, where $\nu^* = 0.98$ and $T_c^*/J = 3.645$. Data from lattice size $L = 8, 16, 32$ collapse to one curve.

model are physically correct.

To understand what happens in the hidden layer and why the machine can learn to classify different phases, we manually generate spin configurations $\boldsymbol{x}$ with linear increasing magnetizations and plot all the components of hidden layer argument given by Eq. (3.8) as a function of the magnetization $m(\boldsymbol{x})$. At the beginning of training, the weight and bias are randomly initialized from a normal distribution with $\mu = 0$ and $\sigma = 1$, so there is no explicit correlation between $\boldsymbol{W_1 x} + \boldsymbol{b_1}$ and $m(\boldsymbol{x})$ (Fig. 3.2(c)). After the training is completed, the elements of $\boldsymbol{W_1}$ and $\boldsymbol{b_1}$ are adjusted so that some components of $\boldsymbol{W_1 x} + \boldsymbol{b_1}$ become constants and the others are approximately linear functions of $m(\boldsymbol{x})$ (Fig. 3.2(d)). These results indicate that, in the hidden layer, the model can learn that magnetization is the "rule" that connects different spin configurations to different phases.

A more powerful aspect of the neural networks is the ability to solve problems beyond its

original design [25]. To demonstrate this, we evaluate our model, already trained by square-lattice spin configurations, by feeding into the evaluation data set produced by Monte Carlo simulation onto triangular-lattice Ising spins. As shown in Fig. 3.3(a), our model is able to classify the phases of triangular lattice Ising spins, and data collapse in finite-size scaling with $\nu^* = 0.98$ and $T_c^*/J = 3.645$ also indicates the existence of a second-order phase transition (Fig. 3.3(b)). The estimated T_c^*/J and ν^* is close to the exact value $T_c = 4/\ln 3$ and $\nu = 1$, and are consistent with the values reported in Ref. [25].

We have shown that the artificial neural networks can be used to classify phases of two-dimensional ferromagnetic Ising model and identify phase transitions. The model trained by square-lattice Ising spins can be directly applied to the case of a triangular lattice. We expect the model could make more precise predictions by extending to a deep neural network (three or more hidden layers) [99] or a convolutional neural network [100].

3.4 Model

The model we study for disordered systems is the three dimensional Edward-Anderson Ising glass model [5] without field, which is described by Hamiltonian

$$\mathcal{H} = -\sum_{\langle ij \rangle} J_{ij} s_i s_j, \tag{3.12}$$

where each J_{ij} is a random variable generated from a certain probability distribution. Here, couplings J_{ij} come from two distributions, one from a Gaussian distribution with zero mean and unit variance and the other one from a bimodal distribution. In this work, we demonstrate how to use a convolutional neural network to detect the phase transition of a three-dimensional Edward-Anderson model with a bimodal distribution by using the network trained with the data from a Gaussian distribution.

3.5 Implementation of convolutional neural networks

In [27] authors successfully use convolutional neural networks to detect the phase transition of two dimensional Ising model with the spin configurations as input. They also show that the impact of convolution in the network is very important compared to the pure neural networks, which can be explained by the fact that the interactions between spins are from the nearest neighbors, therefore the local convolutional filters help more in extracting information about local structures of the spin patterns than just a global linear function as used in pure neural networks. For the same reason, in our model, we use convolutional neural networks as well. But in our model, we use convolutional overlaps as input instead of spin configurations. The reason is that in spin glasses the order parameter is the configurational overlap [101, 83] between two replicas, which is defined as follows:

$$q_{\alpha\beta} = \sum_{i}^{N} S_i^{\alpha} S_i^{\beta}, \tag{3.13}$$

where α and β denote two independent replicas of the same system. Since the configurational overlaps (3.13) include the information about phases, we expect that different phases have different overlap patterns similar to grid-like graphs. Therefore, in the region of a specific phase, it is reasonable to believe that the probability (classification probability) for the convolutional neural network to identify the phase correctly should be larger than 50%. Only at the temperatures that are close to the phase transition point does the classification probability approach 50%. Based on this hypothesis, our task is to identify the temperature at which the classification probability is 50%. This temperature is our estimation of the phase transition point.

In order to quantitatively process the classification probabilities from the simulations, We define a function $p(T, L)$ to represent the classification probabilities, which is a function of temperature t and system size L. In prediction, for each system size we average over 500

samples to calculate the classification probability at each t. Also similar to [25, 26, 27], we label all the prediction samples "0", so that we can expect $p(T, L)$ to be a monotonous function of t at a fixed L. The reason is as follows: usually we expect the lowest probability (0.5) happens around the "confusing area", which is the vicinity of the phase transition temperature, and the closer the temperature is to the phase transition temperature the more "confused" the network would be in term of prediction. Therefore we always expect a "U" shaped pattern for the classification probability as a function of T with lowest point being at the phase transition temperature. But here we mislabel all the samples from category "1", so that half of the "U" shape turns to the mirror image of the other half, which means the whole function is a monotonic function of T. From the scaling hypothesis of the theory of critical phenomena, we expect $p(T, L)$ to have the following behavior in the vicinity of the critical temperature T_{c}:

$$\langle p(T, L) \rangle = \tilde{F}\left[L^{1/\nu_{\mathrm{ml}}}\left(T - T_{\mathrm{c}}\right)\right], \tag{3.14}$$

in which the average is over disorder realizations. Note that the critical exponents ν_{ml} are different from the ones calculated from physical quantities. Due to the limited system sizes that we have studies, finite size scaling (FFS) must be used in order to reliably calculate the critical parameters at the thermodynamic limit. Assuming that we are close enough to the critical temperature T_{c}, the scaling function $\tilde{F}$ in Eq. (3.14) can be expanded to a third order polynomial in $x = L^{1/\nu_{\mathrm{ml}}}\left(T - T_{\mathrm{c}}\right)$.

$$\langle p(T, L) \rangle \sim p_0 + p_1 x + p_2 x^2 + p_3 x^3. \tag{3.15}$$

First we evaluate ν_{ml} by noting that to the leading order in x, the derivative of $\langle p(T, L) \rangle$ in

Eq. (3.15) with respect to temperature has the following form:

$$\frac{d\langle p(T,L)\rangle}{dT} \sim L^{1/\nu_{\rm ml}} \left[p_1 + 2p_2 L^{1/\nu_{\rm ml}} \left(T - T_{\rm c}\right) + 3p_3 L^{2/\nu_{\rm ml}} \left(T - T_{\rm c}\right)^2\right]. \tag{3.16}$$

Therefore, the extremum point of $\frac{d\langle p(T,L)\rangle}{dT}$ will scale as

$$\frac{d\langle p(T,L)\rangle}{dT}|_{T=T^*} \sim L^{1/\nu_{\rm ml}}. \tag{3.17}$$

A linear fit in logarithmic scale will then give the value of $\nu_{\rm ml}$ which is then used to estimate $T_{\rm c}$. To do so we turn back to Eq. (3.15) where we realize that the coefficient of the linear term with $L^{1/\nu_{\rm ml}}$ as the independent variable is proportional to $(T - T_{\rm c})$ which will flip sign at $T = T_{\rm c}$. Alternatively, we can vary $T_{\rm c}$ until the data for all system sizes collapse to a common third order polynomial curve. This is true because the scaling function $\tilde{F}$ as a function of $L^{1/\nu_{\rm ml}} \left(T - T_{\rm c}\right)$ is universal. The error bars can be computed using the bootstrap method.

For training, we use the configurational overlaps from the three-dimensional Edward-Anderson model with Gaussian distribution using Monte Carlo simulation. The details of the parameters in Monte Carlo are listed in Table. 3.1 and the parameters for Monte Carlo simulations that generate the configurational overlaps from bimodal distribution are listed in Table. 3.2.

In order to cover the disorder, we use the same amount of instances as in Monte Carlo simulation [24] with 100 configurational overlaps from each temperature in each instance. Since we already know the phase transition temperature of Gaussian distribution is 0.94 [102], similar to the method used in Ref. [25, 26, 27] for training data we label the convolutional overlaps from temperature above 0.94 "1" and the ones from below 0.94 "0". For

Table 3.1: Parameters for training samples. L is the size of system, N_{sa} is the number of samples, N_{sw} is the number of Monte Carlo sweeps for each of replicas for as a single sample, T_{min} and T_{max} are lowest and highest temperatures simulated, N_T is the temperature numbers used in parallel tempering method for each system size L, and N_{con} is the number of configurational overlaps for each temperature in each instance.

L	N_{sa}	N_{sw}	T_{min}	T_{max}	N_T	N_{con}
8	20000	50000	0.8	1.21	20	100
10	10000	40000	0.8	1.21	20	100
12	20000	655360	0.8	1.21	20	100
14	10000	1050000	0.8	1.21	20	100
16	5000	1050000	0.8	1.21	20	100

Table 3.2: Parameter for prediction samples. L is the size of system, N_{sa} is the number of samples, N_{sw} is the number of Monte Carlo sweeps for each of replicas for as single sample, T_{min} and T_{min} are lowest and highest temperatures simulated, N_T is the temperature numbers used in parallel tempering method for each system size L, and N_{con} is the number of configurational overlaps for each temperature in each instance.

L	N_{sa}	N_{sw}	T_{min}	T_{max}	N_T	N_{con}
8	15000	80000	1.05	1.25	12	500
10	10000	300000	1.05	1.25	12	500
12	4000	300000	1.05	1.25	12	500
14	4000	1280000	1.05	1.25	12	500
16	4000	1280000	1.05	1.25	12	500

the architecture of the convolutional neural network, the parameters are listed in Table. 3.3. We inherit the structure with single layer from the previous work in Ref. [27], and all the parameters are determined by extra validation sample sets, which are also generated from Monte Carlo simulations.

Most importantly, in our work, the prediction samples are from a bimodal distribution which is a different model whose phase transition temperate is not known previously. And this design can show the ability of convolutional neural networks of detecting the common features of different phases in different spin-glass models. We also use $4000 \sim 10000$ in-

Table 3.3: Parameters and Architecture for convolutional neural networks.

Number of Layers	1
Channels in each layer	5
Filter size	3x3x3
stride	2
Activation function	ReLU
Optimizer	AdamOptimizer(10^{-4})
Batch size	1000
Iteration	10000
Software	Python with TensorFlow
Hardware	Lenovo x86 HPC Cluster with single dual-GPU Tesla K80 accelerator and 128 GB of RAM

stances to cover the disorder of a bimodal distribution, which is about 30% of the samples used in Monte Carlo [24]. For each system size, the temperatures range from 1.05 to ~ 1.25, which is in the vicinity of the phase transition temperature. Similar to the other physical quantities, we further use finite size scaling to precisely extract the phase transition temperature, and the details about the finite size scaling we use are described in the next section.

3.6 Results from a model trained by well-prepared data

First of all, we show that a well-trained neural network can give a very precise prediction of transition, which has the same accuracy as the Monte Carlo method [24]. Fig. 3.4 shows the result of neural networks trained with well-prepared data from Gaussian distribution predicting the phase transition of a bimodal distribution. As presented in panel (a) of Fig. 3.4 there are five sets of data each of which represents the average classification probability at each temperature for each size. All the data sets smoothly cross in the close region of the transition point. In panels (b)-(d) we show the estimates of $\nu_{\rm ml}$ and $T_{\rm c}$ using the techniques developed in Section. 3.5.

We want to emphasize that in this experiment, we use fewer instances than in Monte Carlo, but still obtain the same accuracy. It is promising because most studies on spin glasses

require a huge number of samples, while in this work not only does the same accuracy is achieved, but also the sample size is reduced by $\sim 2/3$.

3.7 Results from a model trained by poisoned data

Although the previous section shows that the prediction from the convolutional neural networks can be very precise and efficient, we still need to test how the poisoned data impact the final prediction because in experiments or even in simulations error happens normally. In this simulation, first, we randomly mix the classification labels of the training sample by the probability of 1%, which is already a very small portion since we use 100 sample from each temperature from each instance and in average only 1 sample is mislabeled at each temperature in each instance. Then we train the network by optimized parameters and use the same samples in the prediction stage.

Compared to Fig. 3.4, Fig. 3.5 shows no clear signal of phase transition because there is not a common crossing point at a finite temperature. This means that even by poisoning a very small portion of the training data, the result can be chaos. The reason is that: neural networks have a large number of parameters, and also the networks have hierarchic structures, which make the errors easily amplified in propagation [103, 104]. In our case since we mixed the labels, even by a small portion, the difference between patterns from the spin-glass phase and the paramagnetic phase becomes ambiguous, which is why in Fig. 3.5 the trend of each curve is very different from the one in Fig. 3.4.

Furthermore, we test the impact of poisoned training from a non-equilibrium system. This is similar to what happens in Monte Carlo simulation, in which when you have data from systems that are not in good equilibrium, there is no way to have a good scaling result. Fig. 3.6 shows the prediction result from the simulation that uses the training data from a system that is not in equilibrium. In this simulation, we run 50% Monte Carlo sweeps as needed to reach an equilibrium on each Gaussian instance for training samples. Then we

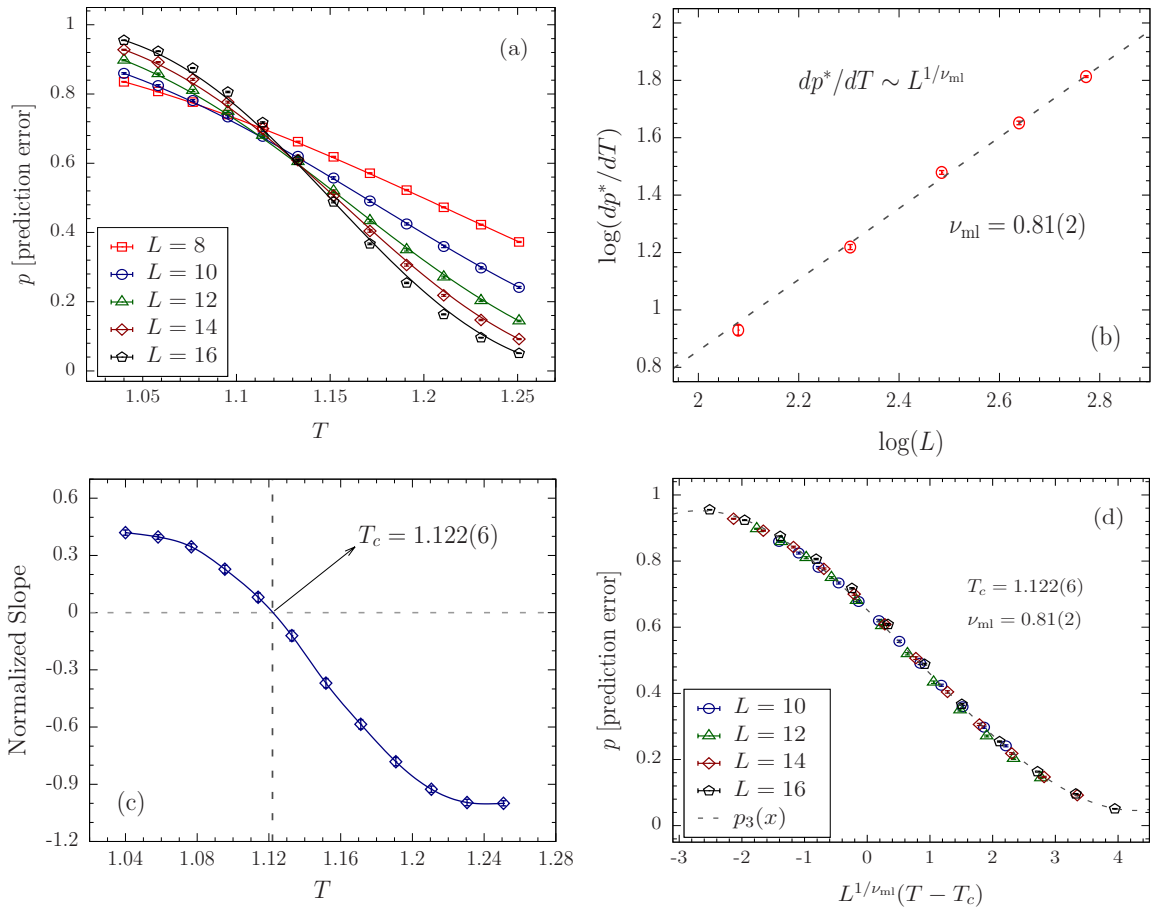

Figure 3.4: Classification probabilities from different system sizes and prediction of phase transition of a bimodal distribution by a convolutional neural network trained by the data from Gaussian distribution combined with finite size scaling. In panel (a) there five different classification probabilities for five different system sizes at different temperatures near the phase transition temperature. Different data sets cross at $T_{\rm c} \sim 1.122$. In panel (b) we measure $\nu_{\rm ml}$ by performing a linear fit in logarithmic scale using the extremum points of the derivative of the prediction error with respect to temperature. Panel (c) shows the estimate of the critical temperature $T_{\rm c}$ using the coefficient of linear term in Eq. 3.15 (normalized to 1) with $L^{1/\nu_{\rm ml}}$ as the independent variable. The vertical dashed line shows the temperature where the slope vanishes which corresponds to $T_{\rm c}$. Finally in panel (d) we have plotted the data for all system sizes as a third-order polynomial in $x = L^{1/\nu_{\rm ml}}(T - T_{\rm c})$ using the previously estimated value of $\nu_{\rm ml}$ and $T_{\rm c}$. We observe that the points collapse onto a universal curve indicating that the estimates are accurate.

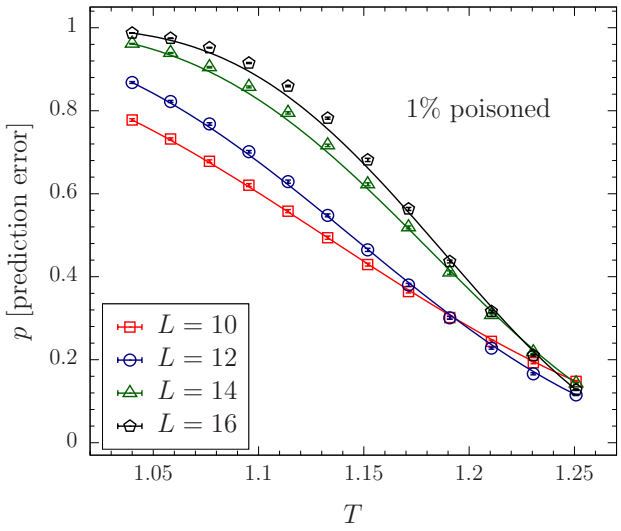

Figure 3.5: Classification probabilities for different system sizes of bimodal distribution with mixed labels by probability of 1%.

follow the same steps as the previous runs. Fig. 3.6 shows that with training by data from non-equilibrium systems the prediction result shows no consistent crossing point between data from different system sizes.

3.8 Summary

We have implemented and evaluated a systematic way to detect the phase transition of a spin-glass system using convolutional neural networks. Our method is nontrivial in two ways: 1. training model and target model are different. This method has the potential to make it possible to study a system that is hard to be simulated by using the information from a relatively easy model. 2.the result is quantitatively precise compared to the traditional Monte Carlo simulation while $\sim 1/3$ samples are used, which has the potential to benefit large system size simulations.

More importantly, we test the impact of poisoned training from different perspectives, which normally happen in research. We test the impact of both mixed training data and non-

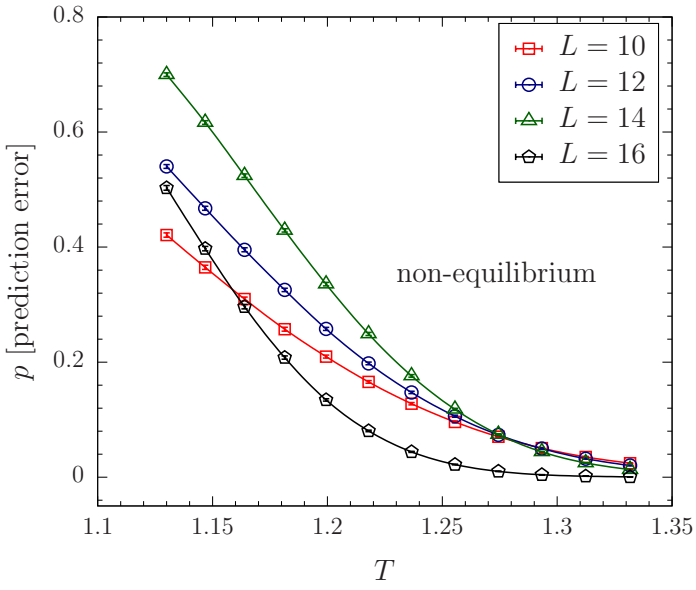

Figure 3.6: Classification probabilities for different system sizes of bimodal distribution predicted by a convolutional neural networks trained with data from system that is not is equilibrium.

equilibrium data. Neural networks usually have a large number of parameters, which does not only mean a powerful ability to learn but also sensitivity to the training data [105]. This is why the training is very important in this statistic learning model. To obtain a reliable and robust result, we need to pay attention to the quality of the training data, which determines the quality of the prediction.

Our method is intuitive. We believe that more advanced algorithms can further improve the ability to detect phase transitions, especially by using much fewer samples and delivering a more precise prediction. In this work, we demonstrate the possibility of neural networks working between two different models which are expected to belong to the same universality class [106, 24]. In the future, models from different universality classes can be tested for the ability of neural networks working on disordered systems.

4. RESTRICTED CLUSTER MOVE ON HIGHLY CONNECTED TOPOLOGIES

4.1 Motivation

The lack of deep understanding of disordered systems such as spin glasses mainly comes from the difficulty of analytically studying the systems beyond mean-field theory and the massive computational hours required by numerical simulations. Therefore efficient algorithms such as parallel tempering [107] would make it reasonable to deeply study disordered systems. However, every algorithm has its own "comfort zone", beyond which we need other algorithms to deal with new situations. One of the toughest situations for simulation disordered systems is called "critical slowing down" [50], which means that at the temperatures close to the critical point, the autocorrelation time is defined by:

$$\tau_{auto} \sim \xi^z, \tag{4.1}$$

with $z > 1$ and typically around 2. According to Eq. (2.9) correlation length ξ diverges at phase critical point, therefore ξ diverges too. This means that at a temperature close to the critical point the time it takes to reach equilibrium is indefinite. In the case of Ferromagnetic systems, cluster algorithms such as Swendsen-Wang [19] and Woff [108] resolve this problem by flipping the cluster that includes the correlated spins, it works very well for Ferromagnetic systems with any dimension. For example, in Wolff algorithm we first select a random spin, then around this spin we gradually add new spins that are parallel to the initial spin with probability $1 - \exp(-2J/kT)$, which guarantees the detailed balance, then flip all the spins that have been added into the cluster. This cluster move randomizes the spin configurations and updates multiple spins, which can effectively reduce the autocorrelation time. But it does not work well when the temperature is extremely low because the in that case almost all the spins align and the acceptance probability for flipping is high, therefore the

algorithm just flip the entire spin configurations without making new states. Also for large temperatures, the spins are pointing to random directions, so the cluster size is very small, which means there is no big difference between "cluster move" and "single spin flipping". Thus this algorithm works best right at the critical point.

Similarly Houdayer developed a cluster algorithm [52] based on configurational overlaps, which does not reflect the spin correlations in the spin system. As we mentioned in Section. 2.4.3, Houdayer cluster move builds a cluster based on the configurational overlaps $q_i = S_i^1 S_i^2$. In this case, the cluster can speed up the equilibration when the cluster is not extended to the size of the entire system (percolation) or the cluster is not comparable to a single spin. Therefore, to make the cluster move works well, we still need to avoid the situations with either small or large clusters. In the previous work [109], based on Houdayer cluster move a rejection-free cluster algorithm - isoenergetic cluster moves (ICM) is developed and it works extremely well on spin systems on topologies with low connectivities by accelerating the equilibration by several orders of magnitudes. One of the most interesting properties of ICM is that the total energy between two replicas is unchanged, which is of importance because any Monte Carlo algorithm used as a solver to find ground states of Boolean variable Hamiltonian can benefit by adding the ICM. Because the ICM keeps the total energy while generating new spin configurations, it has the potential to help the system get through the energy barriers, which is a fact to significantly improve the efficiency of the energy landscape traversal. Therefore for Boolean variable optimization problems, ICM can be a very efficient solver. But when the dimension is high the connections between variables are intensive, which causes cluster move to lose its power due to small percolation threshold [109]. This limit restricts our research on problems with high dimensions, which are very common in both academic and industrial research. For example, SAT instances around threshold are highly connected if you turn these instances into Ising Hamiltonian. Additionally, using Ising spin glass instances with a dimension larger than 3 ICM hardly helps

because of the intensive connection between spins. To update ICM to fit in a solver suitable for any connection, we used different ways to cut the cluster sizes for spin flipping to try to bring back the efficiency of cluster moves.

4.2 Equilibration

In Monte Carlo simulations the goal is to simulate the thermodynamic process of the systems so that we can obtain the physical quantities by averaging over samples recorded during the process. But before the measurement, we need to make that the system has reached the equilibrium, which turns out to one of the most important but difficult tasks in simulations. Since any thermodynamic system can be considered as a subsystem of a bigger closed system (a system not interacting with external word), and the interaction between the system considered and the rest of the bigger system can be described by a heat bath which keeps exchanging energy and interacting with the system considered and allows the system to reach a final equilibrium. Because of the complexity of the interactions, the system will hit every possible state multiple times during a sufficiently long period. Let $\Delta p \Delta q$ denotes an infinitesimal volume that represents a state in phase space of the system, then in a sufficiently long time, the trajectory of the system will pass the small volume many times. Also if we use Δt to be the accumulated time that the system has stayed in a state that is represented by volume $\Delta p \Delta q$, when the total time T approach infinite, then the ratio $\Delta t/T$ reach a limit [110]:

$$w = \lim_{T \to \infty} \Delta t/T. \tag{4.2}$$

A common explanation about this ratio is the probability that the system is observed to be hitting the state represented by the given volume of $\Delta p \Delta q$. If a system is in a state such that to a high degree of accuracy all the physical quantities are equal to the mean value (w in Eq. (4.2)), we say that the system is in equilibrium.

Based on the previous theoretical description, during Mote Carlo we always hope to find

a point after which all the physical quantities will have a stable mean value. A simple and intuitive way is to record all the measurements of all the physical quantities, then compare their mean values at different time points to decide which point can be the starting point of equilibrium. This method is computationally not workable since even for a system of small size it is impossible to record all the measurements for tens of thousands of Monte Carlo sweeps. But for systems without disorders like the Ising model, we can usually measure several physical quantities to test the equilibrium because for simple systems it is safe to say that to a very accuracy system is in a good equilibrium once these physical quantities like energy and magnetism have stable mean values. But the story is different for systems with disorders like spin glasses. Usually, it is very hard to determine if a system has already reached equilibrium because the existence of disorder and frustration, different quantities will have different equilibration time, sometimes the difference is significant.

Although it is usually a tough task to determine the equilibrium for systems with disorder, for spin glasses with Gaussian disorder we have a strong criterion for testing the equilibrium. In a spin system the internal energy per spin is defined as follows:

$$U = -\frac{1}{N}\sum_{i,j}[\langle J_{ij}S_iS_j\rangle]_{\text{av}}, \tag{4.3}$$

where $\langle\cdots\rangle_{\text{av}}$ denotes average over Monte Carlo sweeps and $[\cdots]_{\text{av}}$ denotes average over disorder. If we take a partial integral over J_{ij} we can obtain U from average link overlap [111]:

$$U(q_\ell) = -\left[\left\langle \frac{z}{2}\frac{1-q_\ell}{T}\right\rangle\right], \tag{4.4}$$

here q_ℓ is the link overlap

$$q_\ell = \frac{1}{Nd}\sum_{i,j} s_i^\alpha s_j^\alpha s_i^\beta s_i^\beta, \tag{4.5}$$

z being the number of neighbors per spin and d being the dimension of the system. Therefore

we can define a quantity Δ as a criterion to test the equilibrium of systems with Gaussian disorder:

$$\Delta = [\langle U(q_\ell)\rangle - \langle U\rangle]. \tag{4.6}$$

When $\Delta \to 0$, the bulk of the disorder instances are thermalized [112]. Usually this method gives a reliable estimation of equilibrium for spin-glass system with Gaussian disorder, so we can use this as a criterion to test the efficiency of an algorithm. In [23], this criterion is used to test how the Isoenergetic cluster move helps improve the efficiency on different graphs compared to pure parallel tempering algorithm. And we also use this criterion as one of the methods to test the efficiency of algorithms in my research.

4.3 Isoenergetic cluster move

Isoenergetic cluster move (ICM) is a reject-free cluster move algorithm based on Houdayer cluster move, and it can be used to any dimensional graphs with low connectivities ($< 50\%$). Similar to Houdayer Cluster move two replicas for each temperature are required so that clusters can be built based on the overlaps between two of these replicas. Because the cluster move along is not ergodic we need to combine this move with one of these Markov-chain Monte Carlo algorithms such as parallel tempering and simulated annealing. The algorithm is described as follows [23]: First we perform one or multiple simple Monte Carlo sweeps, then we build the largest cluster based on Houdayer cluster move which is described in Section. 2.4.3. Then if the cluster size is large, say bigger than 50% of the entire system we change the spins in one of the replicas, after which the total energy of two replicas stay the same due to the spin-reversal symmetry and also the cluster size will be reduced to be smaller than $N/2$. After that, we perform the Houdayer cluster mover on all the temperatures to update multiple spins. Optionally we can add any moves before or after the cluster move based on the properties of the problem in hand so that we take advantage of the cluster move while keeping the features of the original algorithm.

The most important condition for ICM to work is that the topology of the graph should have a high percolation threshold, which has been numerically studied in Ref. [23]. The reason is that when the percolation of the graph is small it is relatively easy for the algorithm to build a cluster that has a similar size as the entire system, which means that flipping spins in this cluster are almost equivalent to flipping all the spins. Because there are replicas that have spins pointing to opposite directions flipping spins in a cluster with system size is no different from swapping the two replicas, so that there are no new states generated by this cluster move. In Ref. [23] it studies the comparison on the time to reach equilibrium amongst three different graphs. It shows that the ICM works extremely well on graphs with dimensions around two like two-dimensional Edward-Anderson model and chimera graph, but ICM does not work well on graphs with relatively high dimension. The reason explained in Ref. [23] is that for either two-dimensional Edward-Anderson model or chimera, the percolation threshold is larger than 50% so that the largest cluster size is always lower the threshold, meaning that the cluster will never grow to the size that is comparable to the entire system. But graphs like three-dimensional Edward-Anderson model has relatively small percolation threshold, therefore the largest connected component can easily be the cluster that connects almost the entire system. For this reason, for a graph with low percolation threshold ICM is equivalent to flipping the whole spins, which does not create new configurations.

We conjecture that if we stop the cluster from growing at some cutoff ratio (cluster size/total spin number), ICM might still work well because the inefficiency comes from flipping a large-sized cluster so that the cluster move becomes nothing but swapping two replicas. Because In the previous work [109], it shows that two-dimensional regular lattice is the ideal topology for ICM, we assume that the optimal cutoff ratio could be similar to the ratio as in two-dimensional topology. Besides if we focus on optimization problems we can give up on the detailed balance when performing the cluster move to improve the acceptance of the cluster move.

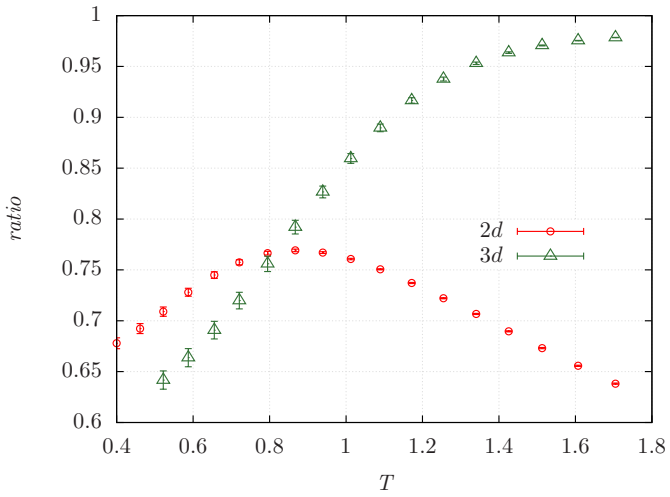

Figure 4.1: Ratios of the cluster sizes over temperatures of the systems in both two-dimensional and three-dimensional with $L = 8$ for Edward-Anderson model with Gaussian disorder. Here the y-axis is the ratio of the size of the cluster grown in ICM over the number of potential cluster members, which is the total number of the sites with overlap equal -1 (In ICM the largest number of potential cluster members is set to be $N/2$).

4.4 Restricted cluster move with detailed balance

As we mentioned before, ICM works much better on a two-dimensional graph than on a three-dimensional graph because of a relatively high percolation threshold. We simulate the two-dimensional Edward-Anderson model and three-dimensional Edward-Anderson model with Gaussian disorder to calculate the ratio of the size of the largest cluster over the size of all the potential cluster members. From Fig. 4.1 we can see that for a three-dimensional model when $T \gtrsim J = 1$ the ratio is $\gtrsim 0.9$, which is much larger than the largest ratio in two-dimensional. For temperatures < 1 the ratio seems to be in a reasonable range, so the cluster move might be able to work in this range. But the T_c for the three-dimensional Gaussian model is ~ 1, therefore the potential cluster size is either very small or very large. Because we already limit the cluster size to be smaller than $N/2$, for temperatures < 1, the cluster size is very small, which means the gain is limited. Unfortunately due to the reasons

we just mentioned ICM does not work for the three-dimensional model. Here we propose to modify the ICM as follows based on the maximum cluster ratio in the two-dimensional model (~ 0.72 in Fig. 4.1):

Algorithm 1: RICM with detailed balance

1 Perform one Monte Carlo sweep (N Metropolis updates) in each replica;

2 Count the number of spins that can potentially be the cluster members N_p;

3 Build a cluster as we do in Houdyer cluster move, but when the cluster size reaches the $0.72 * N_p$ stop growing the cluster;

4 Flip the spins in the cluster built in the last step based on the probability $\exp[\Delta H \beta]$, ΔHbeing the total energy change due to flipping the spins and β being $1/T$;

5 Perform one parallel tempering update for a pair of neighboring temperatures;

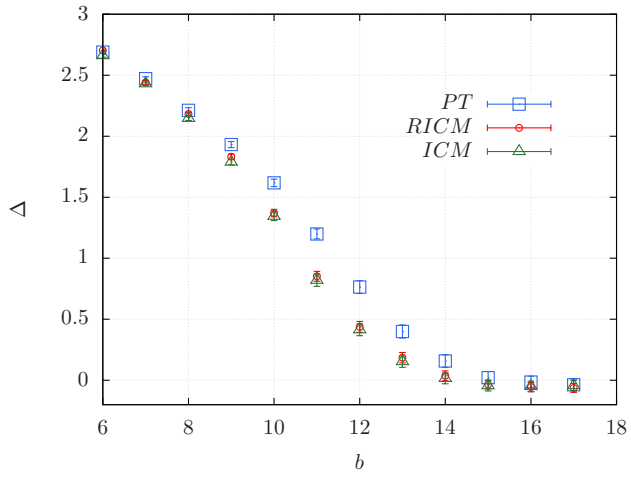

Figure 4.2: Result of simulation performed on three-dimensional Edward-Anderson model of size $L = 8$ with Gaussian distribution. X axis is the Monte Carlo sweeps in power of 2. Y axis is the difference of linkoverlaps calculated in two ways defined in Eq. (4.6). The simulation is performed on three-dimensional Edward-Anderson model of size $L = 8$ with Gaussian distribution. The maximum cluster ratio is selected to be $0.5 * 0.72 = 0.36$, which follows the two-dimensional result in Fig. 4.1. There is almost no improvement from RICM compared with ICM and PT.

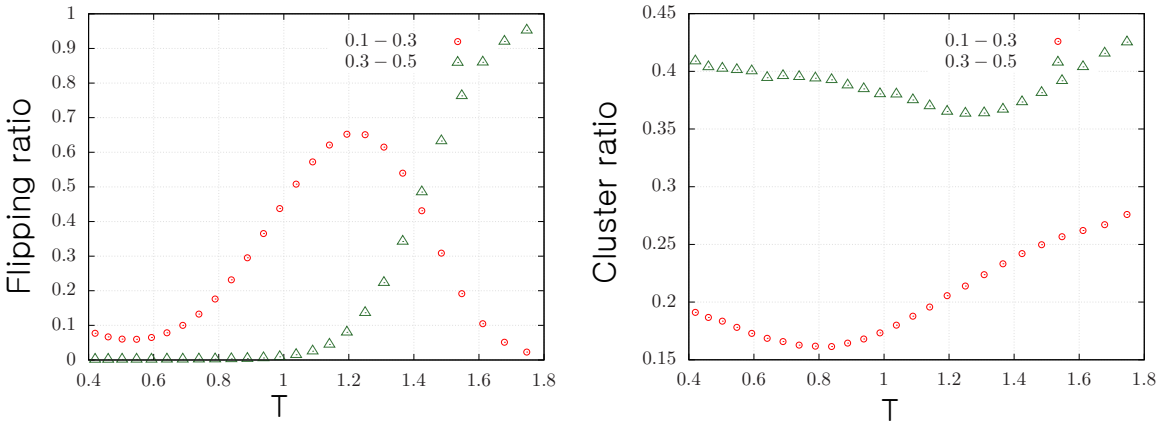

Figure 4.3: Cluster size and acceptance for RICM on three-dimensional Edward-Anderson model with Gaussian distribution. Left panel: acceptance probabilities for RCIM in two regions. For small cluster size (red) the acceptance probability is relatively high, while the acceptance probability is relatively low for large cluster size (green). This means that with the detailed balance either larger cluster size or small size does not allow RCIM to flip spins efficiently. Right panel: The cluster ratios of RCIM in two regions. Green one represents the RCIM with cluster restricted between 10% - 30%, while red one represents the RCIM with cluster restricted between 30% - 50%.

We apply this restricted cluster move (RICM) on both a two-dimensional and a three-dimensional model and the result is showed in Fig. 4.2, which is similar to what is studied in Ref. [23]. To find out if the cluster size is the reason why RICM does not work for three-dimensional model, we do two sets simulations with different ratios of cluster size over the entire system size, which is equivalent to what we described in Algorithm 1 but with easier implementations. The results are shown in Fig. 4.3. We can that when the cluster size is large ($0.3 \sim 0.5$) the acceptance probability is low while the acceptance is high for small cluster size ($0.1 \sim 0.3$), which causes the accepted cluster size to be always small. The reason is that for ICM since it is reject-free all the spins in the cluster will be flipped as long as the cluster has been built but for RICM, the flipping probability is:

$$p = \exp[\Delta H \beta]. \tag{4.7}$$

which takes on small values for a big cluster since a big cluster causes the energy change ΔH to be big while small cluster size allows relatively large flipping probability. Again the overall effect is a small accepted cluster of spins is flipped, which does not give many gains. Based on our simulation we can have a conclusion that for a cluster move that is not rejected-free, as long as the percolation of the graph is small the trade-off between detailed balance and cluster size makes the cluster move inefficient.

4.5 Restricted cluster move without detailed balance

As we mentioned before Monte Carlo algorithms can be used as optimization algorithms to find ground states which are the optimal solutions for the original problems. In this case, we can give up the detailed balance on these algorithms if necessary to improve the efficiency. Coincidently detailed balance is one of the obstacles that make cluster move not to perform well on highly connected graphs. Therefore we try to give up the detailed balance on the cluster move so that we have more freedom to select the cluster sizes. We modify the RICM algorithm as follows:

Algorithm 2: RICM without detailed balance

6 Perform one Monte Carlo sweep (N Metropolis updates) in each replica;

7 Build a cluster as we do in Houdyer cluster move, but when the cluster size reaches a certain size as we designed preciously stop growing the cluster;

8 Flip the spins in the cluster built in the last step;

9 Perform one parallel tempering update for a pair of neighboring temperatures;

Here we abandon the detailed balance, but to keep the algorithm ergodic, we keep the Markov-chain Monte Carlo. In Fig. 4.4 we can see that the modified RICM has the advantage over ICM, but the final value of Δ is not zero, which means the system could be still in equilibrium but the ensemble could not be canonical [110]. Furthermore, we use the planted instances with high connectivities to test if the modified RICM can be used as a solver for highly connected graphs. The metric we use for this test is time to solution (TTS), which is

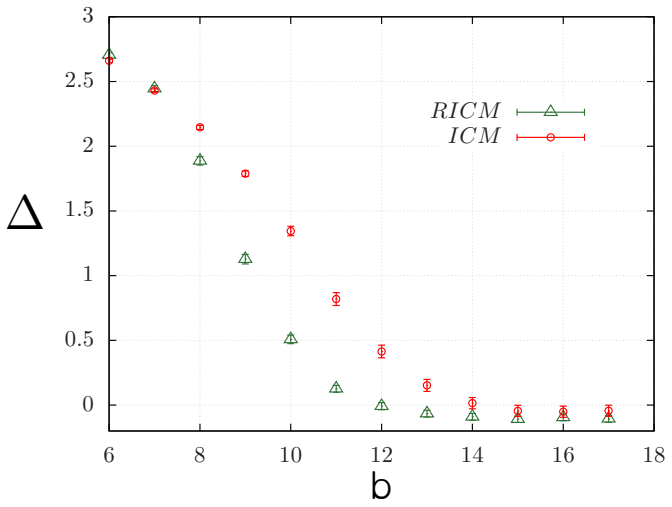

Figure 4.4: RICM without detailed balance under the flipping ratio of 0.167. It shows that Δ drops faster on the system with RICM than ICM. Although the system with RICM does not obey detailed balance, we still consider the Δ as a criterion that measures if the system reaches equilibrium but the final distribution is not Boltzmann distribution.

the time a certain algorithm needs to find a ground state. Fig. 4.5 shows that modified RICM performs even worse than ICM on these highly connected instances. The top panel shows than in terms of Monte Carlo sweeps ICM takes less than modified RICM, which means that regardless of implementation ICM is more efficient as an algorithm. Also in terms of the wall-clock time, ICM still completely outperforms modified RICM, which is presented on the right panel in Fig. 4.4.

4.6 Summary

We systematically considered several ways to extend the working zone of ICM to highly connected graphs. We found that as far as detailed balance plays an important role in the implementation, percolation threshold is the key factor to determine the efficiency for a algorithm that is not reject-free because the acceptance probability for the detailed balance keep the size of the accepted cluster size in a relatively low level, which makes no gain for the cluster move. Although abandoning detailed balance can be a breakpoint to make a clus-

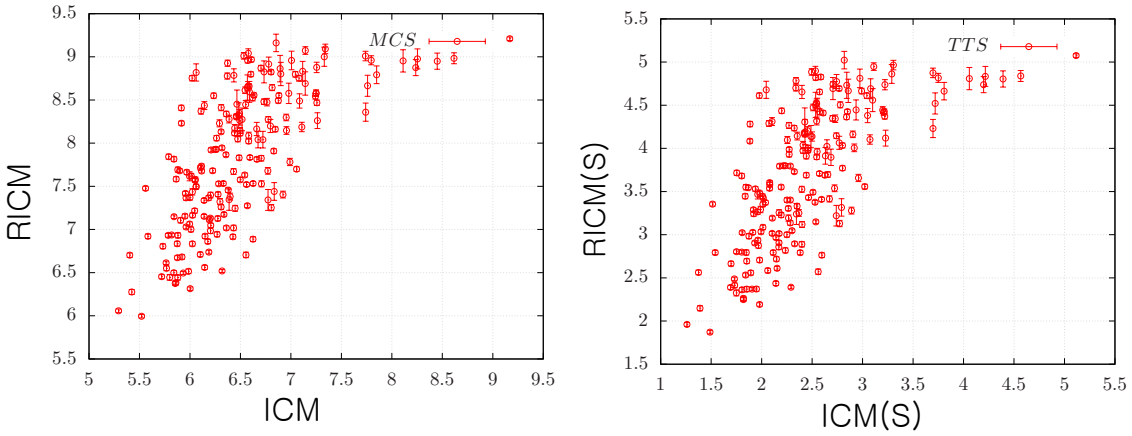

Figure 4.5: Comparison on performances of both RICM and ICM on instances produced in the way presented in Ref. [1]. Both Monte Carlo and Time to Solution (TTS) are on log scale. Left panel: Comparison between RICM and ICM on MCS needed to find the ground states for each solver. It shows that ICM takes less MCS to find ground states for almost all the 200 instances. Right panel: Comparison between RICM and ICM on the TTS. It shows that ICM takes less time to find ground states for almost all the 200 instances. All the error bars are calculated over 200 funs for each instance. The conclusion is that ICM outperform RICM.

ter move more versatile, the system might end up in a different distribution that has a lower probability of hitting ground states. So far cluster move works the best for models that have low connectivities, but as we saw in Fig. 4.1 if we can find a specific temperature region in which the ratios of the cluster sizes are $30\% \sim 40\%$ the cluster move can still be efficient.

5. UNIVERSALITY OF TWO-DIMENSIONAL BOND DILUTED ISING MODEL

In this project, I study the universality in the two-dimensional Ising model with bond dilution by Monte Carlo simulation. To avoid the effect of the small clusters with zero-dimension, we consider only the largest cluster in geometry. We mainly study the behavior of the critical exponent of the correction to the correlation length by simulating the Binder cumulant in different bond concentration. The numerical calculation and finite size scaling analysis suggest that the critical exponent of the correction to the correlation length is independent of the dilution, which supports the so-called strong universality scenario.

5.1 Motivation

Critical behavior in topologies with a disorder is of great importance in the research of phase transition in complex systems. Universality, as one the three pillars for modern theory of critical phenomena was first brought up by Kadanoff at 1970 [113] and gives a way to categorize different critical systems into relatively few "universality classes" (two systems with the same values of local critical exponents and scaling functions belong to the same universality class, which only depends on some global features such as dimensionality and symmetry instead of the details of local interactions [24, 113]). A modern explanation of universality comes from Wilson's renormalization group theory [114, 115, 116, 117] that a T_c can be explained as a fixed point of a transformed Hamiltonian of the system, and renormalization group also provides a calculation tool for numerical studies of critical exponents. Besides, for these two-dimensional models, the critical behavior can also be analytically studied by conformal field theory [118, 119], which has already helped exactly determine the critical exponents for many two-dimensional models [120, 118, 121, 122]. Although these analytical and numerical methods have helped statistical physicists made enormous progress on determining critical exponents for various universality classes [123, 124, 125],

there are few works on systems with dilution. As a result, attention has been drawn to the area of diluted systems, which belong to the category of disordered systems.

The earliest research in the area of impure systems could be tracked to 1974 when Harris criterion was introduced for the first time [126]. The Harris criterion can be briefly described as follows: if the specific heat critical exponent α is positive, disorder would affect both the critical component and universality class, and if α is negative, the disorder will affect neither the critical component nor universality class. However, some recent works report that the Harris criterion is violated in the Voronoi Lattice [127]. For example, the Ising model in three-dimensional Voronoi lattice has the same critical behavior as the system with complete geometry [127]. Besides, the two-dimensional three-state Potts model also violates Harris criterion in such a lattice [128]. Therefore, it is still a debate on how critical behaviors depend on the disordered structure. One of the most interesting and relevant problems is the criticality in the disordered structure in the two-dimensional Ising model, of which the complete structure has zero value for α so that the issue is not inclusive of the Harris criterion. Recently, for this situation, physicists have two different points of view. One is that this behavior follows the weak universality scenario, meaning that disorder will very slowly change critical exponents. The other one is the strong universality scenario meaning that critical behavior is independent of the disorder. Recent works [129, 130], in which both Monte Carlo simulation and renormalization group methods are used, have supported strong universality scenario, but also found the existence of logarithmic corrections to the critical behavior. For example, in Ref. [129] a two-dimensional Ising model with site dilution is studied by simulating the universal distribution function. The critical behavior is further characterized by an effective temperature with scaling form as:

$$T_L = T_c + AL^{-(1+\theta)/\nu}, \tag{5.1}$$

where T_L is an estimated critical temperature for diluted system with size L, which is determined by the scaled distribution function, ν is the correlation length critical exponent, which does not vary as disorder involves, θ is the correction exponent depending on the strength of disorder and A is a pre-factor. In Ref. [129] the site dilution is realized by randomly canceling a certain number of sites according to the site concentration. Results show that the correction term θ in Eq. (5.1) changes over the concentration of the sites. A very interesting phenomenon is that in this model even with a very small change in concentration, correction term varies a lot, especially for the case of concentration dropping from 1.0 to 0.99 while θ changing from ~ 3.0 to ~ 1.8, which is much more significant than the change for the same concentration difference at other site concentrations. It would be natural to speculate that this phenomenon might be caused by the zero-dimensional clusters in the diluted system, which are formed by sites that are not connected to the major connected component. Also in calculating T_L, it is still difficult to quantitatively and precisely identify two identical scaled distributions represented in two different histograms. We suspect that these two facts can have an impact on the precision of the estimation on θ. Although numerical evidence delivered in Ref. [129] seems to support the strong universality scenario, the existence of different logarithmic corrections makes the debate about the universality of diluted systems continue. Therefore studying the diluted two-dimensional Ising model from a different perspective is necessary.

5.2 Model, Observations and Algorithms

The hamiltonian of the Ising model is given by:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} S_i S_j, \tag{5.2}$$

where J is a constant, the summation over i and j goes over all the adjacent spin pairs and $S_i = \pm 1$. Periodic boundary condition is applied in our model. Similar to what has been

done for site dilution in [131], for bond dilution we introduce a variable ϵ_{ij} for each bond to control the bond dilution:

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} \epsilon_{ij} S_i S_j, \tag{5.3}$$

where ϵ_{ij} takes value 1 by probability of p, which is the bond concentration.

To implement the dilution, we need to first decide the concentration p, then generate some instances to cover the disorder. Considering that we only need the largest connected cluster in the diluted system, we use the breadth-first search (BFS) algorithm to efficiently pick out the largest cluster:

Algorithm 3: Find the largest connected cluster for an diluted instance

Initialize a set (st) to store the sites that have already been visited and a vector (v) to record the indexes of the spins in the largest connected component so far;
for $i = 1$ *to* $i =$ *spin number* $- 1$ **do**
    if i *is in* st **then**
        continue;
    end
    Initialize a local vector lv for storing cluster member connected to i and a queue q to perform the BFS search;
    Initialize q with i;
    while q *is not empty* **do**
        for d*: members in* q **do**
            Push all the spins connected to d that are not in st into q and lv;
            Add d into st;
            Pop out d from q and push d into st;
        end
    end
    if *size of* $lv >$ *size of* v **then**
        $v = lv$;
    end
end
Reindex all the spins in v;

In simulation we measure Binder cumulant (or Binder ratio) [47], given by:

$$g_m = \frac{1}{2}\left(3 - \frac{\left[\langle q^4\rangle_T\right]_{av}}{\left[\langle q^2\rangle_T\right]^2_{av}}\right), \tag{5.4}$$

where $\langle \cdots \rangle$ represents a thermal average, $\big[\cdots \big]$ is a disorder average over all the instances and q is the spin overlap defined as follows:

$$q = \frac{1}{N} \sum_{i=1}^{N} S_i^{\alpha} S_i^{\beta}, \tag{5.5}$$

here α and β are two replicas of the system with same bonds and disorder. Binder cumulant is a dimensionless quantity, which has simple scaling format [132]. But in Eq. (5.4) the normalized Binder cumulant is in the range $g \in \big[0, 1\big]$ near T_c, especially below T_c, so that it is hard to "splay out", we use the unnormalized binder ratio instead [124]:

$$V_4 \equiv \frac{\langle m^4 \rangle}{\langle m^2 \rangle^2}, \tag{5.6}$$

which we will describe in details in Sec.5.3.

Considering that our simulations are performed near T_c, critical slowing down happens [50]. So we use a hybrid algorithm where each Monte Carlo sweep includes a Metropolis algorithm followed by the Parallel tempering algorithm [49] and Wolff algorithm [108]. By using this combination, the equilibrium for our model is easy to be reached after $\sim 10^4$ Monte Carlo sweeps, after which we usually need $\sim 10^9$ more Monte Carlo sweeps to record data points to obtain an accuracy around 10^{-4} for V_4. For each concentration, the system size L goes from 20 to 70 with each L having 7 temperatures.

5.3 Finite size scaling analysis and numerical results

According to the scaling ansatz [133, 134], the cumulant ratio V_4 has the scaling form [46, 24]:

$$V_4(T, L) = \tilde{V}(A_V t L^{1/\nu}, B_V L^{-\Delta}), \tag{5.7}$$

where $\tilde{V}$ is an universal function, ν is the critical exponent of the correlation length, Δ is the critical exponent of the correction of the correlation length [135, 136], $t = \frac{T - T_c}{T_c}$ is the

reduced temperature and A_V and B_V are two universal constants. When the system size L is large enough, the correction $L^{-\Delta}$ can be neglected. The scaling formula in Eq. (5.7) is reduced to:

$$V_4(T, L) = \tilde{V}_r(tL^{1/\nu}), \tag{5.8}$$

which is usually the formula to be used to determine the critical exponents [24]. The scaling relation in Eq. (5.7) has been well established both analytically and numerically and already been proved that the critical exponent is $\nu = 1$ [135, 136]. Our goal is to focus on the universality issue on the second part in Eq. (5.7), which is related to the correction of critical exponent Δ. We aim to further determine the value of correction of critical exponent Δ for both pure Ising model and diluted system so that the two universality scenarios can be investigated from a different angle from what has been done in Ref. [129] .

Eq. (5.7) can be expanded in powers of $L^{-\Delta}$ as follows:

$$V_4(T, L) = V_0(tL^{1/\nu}) + V_1(tL^{1/\nu})L^{-\Delta} + O(L^{-2\Delta}), \tag{5.9}$$

where V_0 and V_1 are coefficients depending on $tL^{1/\nu}$. At $T = T_c$, Eq. (5.9) can be reduced to:

$$V_4(T_c, L) = V_0(0) + V_1(0)L^{-\Delta} = V_\infty + BL^{-\Delta}, \tag{5.10}$$

here $V_\infty = V_0(0)$ is an universal value for V_4 when L is infinity large at critical temperature and $B = V_1(0)$. This is the scaling formula we will be focused on in the this work. We will see that in order to estimate Δ we need to know T_c, which is not a problem for pure Ising but needs a bit more work for diluted systems that will be described later.

First, we consider the pure Ising model. Since the exact T_c is exactly known, we can directly fit the Eq. (5.10) to the data from simulations. The data from the simulation is showed in Table. 5.1. Fig. 5.1 shows a good agreement between the equation and the simulated data.

Table 5.1: Simulation data of V_4 for pure Ising model from each size. T is the temperature of system, the errors are calculated from 400 instances using bootstrap resampling, for each instance $10^4 \sim 10^5$ Monte Carlo sweeps is run for equilibration and 10^9 data points are recorded.

T	20	30	40	50	60	70
2.2686	1.16530(7)	1.16538(8)	1.16516(1)	1.16462(8)	1.16389(7)	1.16359(12)
2.2688	1.16548(8)	1.16583(10)	1.16598(7)	1.16557(7)	1.16538(9)	1.16493(13)
2.2690	1.16590(7))	1.16662(10)	1.16658(11)	1.16672(8)	1.16662(9)	1.16651(10)
2.2692	1.16636(8)	1.16721(9)	1.16736(9)	1.16747(10)	1.16785(8)	1.16801(11)
2.2694	1.16691(10)	1.16778(8)	1.16830(10)	1.16862(10)	1.16917(11)	1.16928(11)
2.2696	1.16723(7)	1.16832(10)	.16898(10)	1.16968(10)	1.17028(14)	1.17082(10)
2.2698	1.16756(8)	1, 16904(7)	1.16995(11)	1.17054(10)	1.17144(9)	1.17236(14)

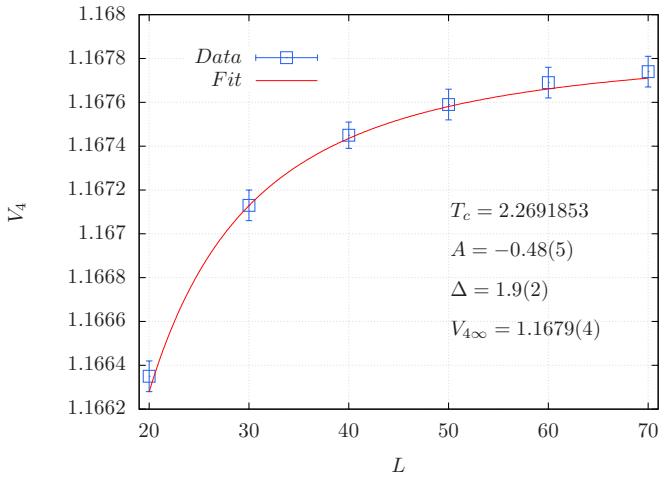

Figure 5.1: Fitting Eq. (5.10) to the data from Monte Carlo simulation. Blue data points are original data from Monte Carlo simulation and the error bars are calculated by resampling from 400 sets of runs with bootstraps. The estimated Δ is $1.96(10)$ and the estimation of $\mathrm{V}_{4\infty}$ is $1.1679(4)$ agree well with results in Ref. [124] within error bars.

The correction Δ and the $V_{4\infty}$ extracted from the fit agree very well with the results calculated in Ref. [124]. For diluted systems, we consider the two bond dilution rates $p = 0.99$ and $p = 0.97$. These two rates are quite close to 1.00 to ensure that the value of p is far from the percolation critical point so that we can avoid the impact of another disorder caused by

the existence of multiple sizable clusters, which is not we are interested in. Compared to pure model, the most difficult part in implementation is that for pure Ising model we already know the exact solutions [137, 37] so we only need to consider Δ for the scaling, while for a diluted model because we do not know the T_c we have to deal with the T_c as well. However, we can still freeze the freedom for T_c by using different trial temperatures to narrow down the error bars for the estimations of Δ. Indeed Eq. (5.10) only works in a very small vicinity of T_c, therefore for data from temperatures that are not within the error bars, there is no good fit for Eq. (5.10).

To determine the trial temperatures we look for the crossing point for the data from different sizes because according to Eq. (5.8) the leading team shows a universal behavior at T_c for different sizes, which means that all the data will cross at T_c [50, 24]. Following this principle, we can continue to narrow down the interval of confidence to a certain size that is precise enough. In our simulation, we first use the finite size scaling to find the crossing point, then use a regression function to find the trial temperatures near the crossing point. The data from simulations for $p = 0.97$ and $p = 0.99$ are showed in Table. 5.2 and Table. 5.3. Fig. 5.2 shows the results of the trial temperatures, which have been narrowed down to the three temperatures in the region that crossing happens.

After determining the trial temperatures we fit Eq. (5.10) to the data from each of these trial temperatures, which is demonstrated in Fig. 5.3 for $p = 0.97$ and Fig. 5.4 for $p = 0.99$. This approach is reliable because it gives the error in the worst case directly based on the original data from simulations. For example in our case, by a series attempts for trial temperatures, we have narrowed down to three temperatures which means that the interval of confidence is the gap between the highest trial temperature and the lowest trial temperature. This non-linear fit is similar to the linear fit in Ref. [138]. The reason why we use non-linear fit is that the propagation errors that come from taking logarithm on Eq. (5.10) can be large for large L's, which is also the reason why we simulate system sizes up to 70.

Table 5.2: Simulation data of V_4 for diluted Ising model with bond concentration of 0.99 from each size. T is the temperature of system, the errors are calculated from 400 instances using bootstrap resampling, $10^4 \sim 10^5$ Monte Carlo sweeps is run for equilibration and 10^9 data points are recorded.

T	20	30	40	50	60	70
2.23750	1.16244(8)	1.16180(6)	1.16057(7)	1.15923(7)	1.15783(4)	1.15656(5)
2.23830	1.16040(5)	1.16419(7)	1.16372(6)	1.16316(5)	1.16248(4)	1.16187(4)
2.23895	1.16532(5)	1.16610(6)	1.16635(7)	1.16641(5)	1.16639(4)	1.16649(5)
2.23899	1.16540(5)	1.16622(6)	1.16651(7)	1.16666(7)	1.16663(4)	1.16677(5)
2.23902	1.16546(5)	1.16632(6)	1.16663(7)	1.16677(7)	1.16682(4)	1.16698(5)
2.23950	1.16647(5)	1.16782(7)	1.16857(7)	1.16921(7)	1.16976(4)	1.70430(5)
2.24000	1.16749(5)	1.16937(7)	1.17063(7)	1.17178(7)	1.17286(5)	1.17406(5)

Table 5.3: Simulation data of V_4 for diluted Ising model with bond concentration of 0.97 from each size. T is the temperature of system, the errors are calculated from 400 instances using bootstrap resampling, for each instance $10^4 \sim 10^5$ Monte Carlo sweeps is run for equilibration and 10^9 data points are recorded.

T	20	30	40	50	60	70
2.1765	1600(2)	1.1589(2)	1.1573(2)	1.1555(2)	1.1538(1)	1.1520(1)
2.1775	1.1620(2)	1.1619(2)	1.1612(2)	1.1603(2)	1.1595(1)	1.1586(1))
2.1784	1.1637(2)	1.1644(2)	1.1646(2)	1.1645(2)	1.1645(1)	1.1644(1)
2.1785	1.1640(2)	1.1649(2)	1.1652(2)	1.1652(2)	1.1653(1)	1.1654(1)
2.1786	1.1641(2)	1.1651(2)	1.1655(2)	1.1655(2)	1.1658(1)	1.1659(1)
2.1795	1.1660(2)	1.1678(2)	1.1691(2)	1.1702(2)	1.1714(1)	1.7250(1)
2.1800	1.1670(2)	1.1694(2)	1.1713(2)	1.1728(2)	1.1745(1)	1.1761(2)

For $p = 0.97$, the estimated T_c is $2.17850(14)$, Δ is 1.97(18) and $V_{4\infty}$ is 1.1655(9), which are showed in Fig. 5.3. The results for $p = 0.99$ are listed in Fig. 5.4, which are $T_c = 2.23899(4)$, $\Delta = 1.96(10)$ and $V_{4\infty} = 1.1699(3)$. Within error bars the Δs from both $P = 0.97$ and $p = 0.99$ are in a very agreement with the result for pure Ising model, which is showed in Fig. 5.1, which is supported by renormalization group theory arguments that the critical exponent of the correction should be the same for the same universality class.

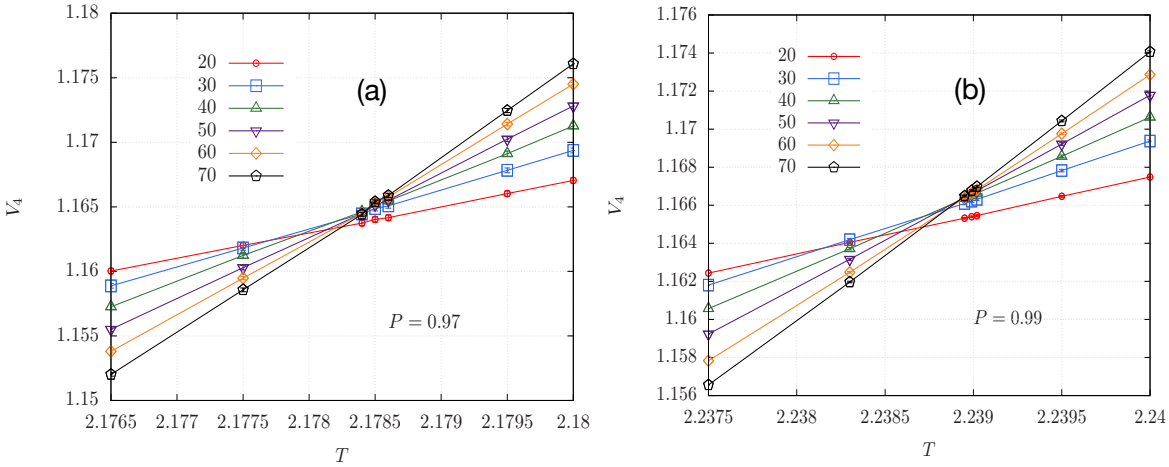

Figure 5.2: Cumulant ratio V_4 as a function of system size L. (a): Data for ratio $p = 0.99$. The crossing happens in range from the 3rd to the 5th temperatures (trial temperatures). (b): Data for ratio $p = 0.99$. The crossing happens in range from the 3rd to the 5th temperatures (trial temperatures). These data points at trial temperatures are generated based on a finite size scaling on the original data. First use finite scaling to find the estimated T_c. Then in the vicinity of T_c fit Eq. (5.10) to the data until a the regression can not be held inside error bars, record the up and lower limit of T_c.

And this is also a strong evidence to support strong universality class scenario for disordered system, which agrees with the conclusion in Ref. [129]. Also we found a slight disagreement in $V_{4\infty}$ that by introducing the dilution the $V_{4\infty}$ changes from $1.1679(4)$ to $1.1655(4)$, which shows a consistent trend of decreasing over disorder.

5.4 Summary

Numerically, we have studied the critical exponent Δ of the correction to the correlation length in diluted Ising models with different bond concentration. The results show that within error bars the correlation keeps the same when introducing disorders by changing the bond concentration. This supports the strong universality scenario, which agrees with what has been found in the Ising model with site dilution [129]. Importantly, we do not find the Δ slowly changing, which is different from what has been reported in Ref. [129]. Compared to the implementation of dilution in Ref. [129], we suspect that a logarithmic correction might

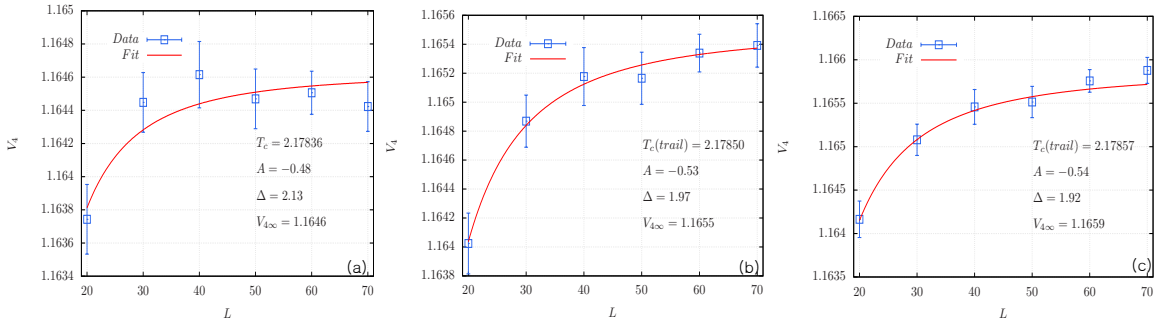

Figure 5.3: Fitting Eq. (5.10) to data from $p = 0.97$ at each trial temperature. (a): fitting result for trial temperature 2.17836. For the second and 6th data point the fit can barely keep within the error bars. This identify the lower boundary of the validity of fitting Eq. (5.10). For temperatures that lower than this temperature there will never be a good fit then can not be a estimation of T_c. (b)l: fitting result for trial temperature 2.17850. All the data points stay within the error bars. This is a good estimation of T_c. (c): fitting result for trial temperature 2.17857. For the 6th data point the fit can barely keep within the error bars. This identify the high boundary of the validity of fitting Eq. (5.10). For temperatures that higher than this temperature there will never be a good fit then can not be a estimation of T_c. The estimations of T_c, Δ and $V_{4\infty}$ including the error bars can be read from these three trial temperatures: $T_c = 2.17850(14)$, $\Delta = 1.97(18)$, and $V_{4\infty} = 1.1655(9)$.

be caused by the impact of zero-dimensional clusters in diluted models. We also notice that $V_{4\infty}$ depends on P, which is controversial to the traditional acknowledgment that the $V_{4\infty}$ should be universal for the same universality class. This puzzling paradox seems not a single case [139], which requires further study. Although the asymptotical behavior of $V_{4\infty}$ seems not to completely agree with what we expect, the current results on the critical exponent of the correction to the correlation length tend to support the strong universality scenario.

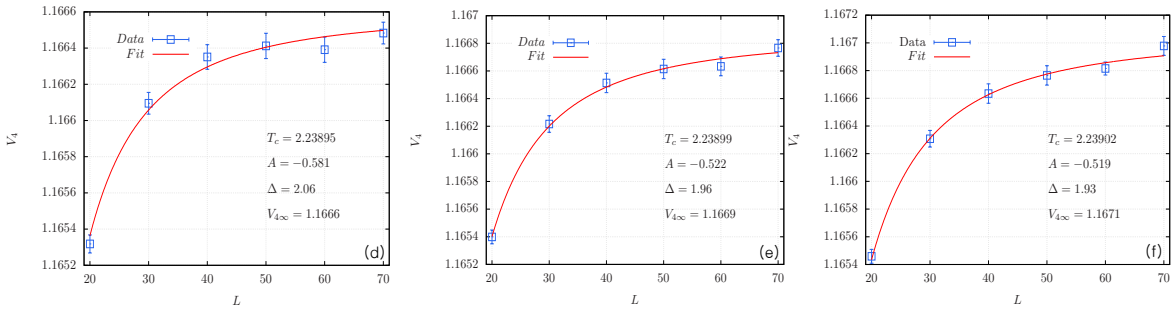

Figure 5.4: Fitting Eq. (5.10) to data from $p = 0.99$ at each trial temperature. (d) fitting result for trial temperature 2.17836. For the second and 6th data point the fit can barely keep within the error bars. This identify the lower boundary of the validity of fitting Eq. (5.10). For temperatures that lower than this temperature there will never be a good fit then can not be a estimation of T_c. (e): fitting result for trial temperature 2.17850. All the data points stay within the error bars. This is a good estimation of T_c. (f): fitting result for trial temperature 2.17857. For the 6th data point the fit can barely keep within the error bars. This identify the high boundary of the validity of fitting Eq. (5.10). For temperatures that higher than this temperature there will never be a good fit then can not be a estimation of T_c. The estimations of T_c, Δ and $V_{4\infty}$ including the error bars can be read from these three trial temperatures: $T_c = 2.23899(4)$, $\Delta = 1.96(10)$, and $V_{4\infty} = 1.1669(3)$.

6. APPLICATION OF SPIN GLASSES: BOOLEAN SATISFIABILITY PROBLEM AND BOOLEAN SATISFIABILITY BASED MEMBERSHIP FILTER *

Optimization problems with Boolean variables that fall into the nondeterministic polynomial (NP) class when cast as decision problems are of fundamental importance in computer science, mathematics, physics, and industrial applications. Most notably, solving Boolean satisfiability problems, which are related to spin-glass-like Hamiltonians in physics, remains a difficult numerical task. As such, there has been great interest in designing efficient heuristics to solve these computationally difficult problems. Inspired by parallel tempering Monte Carlo developed for Ising spin glasses, we present a generalized global searching optimization heuristic that can be applied to different NP-complete problems with Boolean variables. The global searching algorithm allows for a wide-spread sampling of phase space, thus considerably speeding up optimization. More than that a new type of membership filter that is based on the Boolean satisfiability (SAT) problem has been proposed and studied in Ref. [3]. It shows that this new type of filter (SAT filter) is a promising candidate by offering high efficiency in the usage of storage. Base on this proposal in Ref. [3], we try to use a new solver that is based on the Ising spin form and a modified form of SAT filter to improve the performance of SAT filer.

6.1 Application of metropolis algorithm with parallel tempering on Boolean satisfiability problem

In complexity theory, there is a class of problems defined as P problems [7], which can be solved by polynomial-time decision algorithms. Another class, Nondeterministic Polynomials (NP) [28, 29, 30, 16], can only be checked for the correctness of a given solution in

*Part of this section is reprinted with permission from "*borealis* - A generalized global update algorithm for Boolean optimization problems" by Zhu, Z., Fang, C., & Katzgraber, H.G., 2020. Optimization Letters, https://doi.org/10.1007/s11590-020-01570-7, Copyright [2020] by Springer Nature.

polynomial time. NP-complete problems are an important subclass of NP problems, which can be introduced as follows: if A is NP-complete then any solution to A can be mapped in polynomial time as a solution to any of the NP problems. Therefore, if we can prove that "some P problem is NP-complete." then all the NP problems are polynomially solvable. In the category of NP, the Boolean satisfiability (SAT) problem is a very important problem due to its useful applications in industry. One of the most important applications of SAT problems is SAT filter [3], a highly efficient set membership filter that can be used in areas like web indexing [140] and virus detection [141]. Additional applications of the SAT include Combinational Equivalence Checking, Automatic Test-Pattern Generation, and Model Checking. Because these NP problems have many direct or indirect applications in industry, efficient solvers for NP problems are in demand, which has already boosted the development of many efficient solvers in computer science. For example, *Walksat*, developed by Selman et al. [142], is a classic local search solver for random SAT instances whose working zone is ratios lower than the threshold. Another solver, *Dimetheus* [143], is famous for its ability to work around the instances with ratios higher than thresholds. Although we already have many efficient solvers, the market is still hungry for more efficient solvers because harder problems have been discovered and are waiting to be solved. Fortunately, we can develop new algorithms for general NP problems from the perspective of physics, using the Ising model [35]. This method is strongly supported by the fact that It has been proven that the Ising model * is NP-hard [16], meaning that all NP-complete problems (and all the NP problems by proxy) can be reduced to Ising models. More than that, many efficient algorithms have already been developed [50] and widely used in the research area of Ising spin models. Also, it has been proven that many NP problems, including SAT, MVC, and Traveling Salesman problems (TSP), can be mapped to the Ising Models with reasonable effort [17], which makes physics a promising approach to solve these NP problems. One of my research areas

*In this thesis, we refer to the Ising model as the model that could have both positive and negative for bonds. This fits one of the mathematical definition of spin glass.

is to find efficient ways to apply these physics algorithms to solve SAT, and further develop these algorithms for industrial applications.

6.1.1 Boolean satisfiability problem

An Boolean satisfiability problem (SAT) problem involves determining if there is a set of values for its Boolean variables (either "False (0)" or "True (1)") such that the given Boolean formula can be satisfied. If the goal is to satisfy the maximum number of clauses instead of the entire formula, the problem is called MAX-SAT. The Boolean formula is built from variables, operators AND (conjunction, $\wedge$), OR (disjunction, $\vee$), NOT (negation, $\neg$), and parentheses. A formula is said to be "satisfiable" if it can be made True by assigning appropriate logical values (i.e. True, False) to its variables. Any formula is built by the block of clauses, which is a combination of literals connected by operators. A literal is either a variable, called positive literal, or the negation of a variable, called a negative literal. A clause is a disjunction of literals (or a single literal). A formula is in conjunctive normal form (CNF) if it is a conjunction of clauses (or a single clause):

$$\begin{aligned} \text{CNF} &: C_1 \wedge C_2 \cdots \wedge C_m \ , \\ \text{Clause} &: C_m = l_{i,1} \wedge \cdots \vee l_{i,k_i} \ , \\ l_{i,k_1} &\text{ is a literal. It could be either } x_k \text{ or } \neg x_k \ . \end{aligned} \tag{6.1}$$

If each clause includes exactly k literals and all clauses are randomly, uniformly and with replacement drawn, it is called random k-SAT [144]. Random k-SAT instances exhibit quite regular behavior in terms of the clauses-to-variables ratio. Specifically, given a random k-SAT instance $\mathcal{S}$, the ratio α determines with high probability the satisfiability of $\mathcal{S}$ [145]. Given a fixed k, there exists a number α_k such that whenever $\alpha_{\mathcal{S}} < \alpha_k$, $\mathcal{S}$ is almost certainly satisfiable, and whenever $\alpha_{\mathcal{S}} > \alpha_k$, $\mathcal{S}$ is almost certainly not satisfiable. In Ref. [146] it is found that $\alpha_k = 2k \ln 2 O(k)$, and the experimental values can be found in Ref. [3].

SAT also has many variants. Not-All-Equal-SAT (NAE-SAT) is one of them. The difference between random SAT and NAE-SAT exists in the constraints to the solutions. As we introduced above, a solution to a random SAT CNF is an assignment for all the clauses to be satisfied. Therefore, to satisfy a single clause, at least one literal should be "True", which means the only unsatisfied situation for a single clause is all the literals unsatisfied. But in NAE-SAT, a clause with all the literals satisfied is considered as "unsatisfied". This feature causes the solutions of NAE-SAT (NAE-solutions) to have large hamming distances so that the solutions can be considered uncorrelated [147]. Considering that uncorrelated solutions guarantee high efficiency for an SAT filter, NAE-SAT can improve the performance of SAT filters. The maximum satisfiability problem (MAX-SAT) is also an NP optimization problem of determining a set of Boolean variables $\{x_1, ..., x_N\}$ that maximize the number of satisfied clauses $\{C_1, ..., C_N\}$ in a conjunctive normal form defined in Eq. (6.1).

Many optimization problems in the NP complexity class can be solved by local search (LS) heuristics. These types of algorithms start from a candidate solution and then iteratively moves to a neighboring solution with random or greedy moves of single Boolean variables. However, either the greedy single-variable dynamics are quickly trapped in local minima of the cost function, or exhaustively explores plateaus in the landscape where no local moves can decrease the cost in a reasonable amount of time. To escape this single-move traps, randomizing moves can be performed at the cost of additional computational time. Paradigmatic examples of (stochastic) local search algorithms have evolved from algorithms such as GSAT and WalkSAT [148] for the maximum satisfiability problem, NuMVC [149] for minimum vertex covers, as well as simulated annealing and 2-opt algorithms [150, 151] for the traveling salesman problem. For spin glasses, methods such as extremal optimization [152], local genetic algorithms [153] or the cluster-exact approximation method [7, 8] have been successful in tackling problems with up to approximately 2^{12} variables. In contrast to these local search algorithms that rely on updating variables in a certain area of

the configuration space, when trapped in a local minimum are restarted from a new initial configuration, global update algorithms target in traversal in the whole solution space. As we mentioned parallel tempering algorithm is capable of avoiding the system from getting trapped in local minima, and because it keeps the ergodic property parallel tempering can be an ideal candidate for a global searching algorithm for the ground states. Fortunately, there is a close relationship between the statistical physics of Ising spin glasses and a wide variety of Boolean NP problems [16]. Mathematically, because the decision form of the Ising spin glass model is NP-complete [32], there exists a polynomial-time mapping to any other NP-complete problem with Boolean variables [29]. In this way, the original problem can be transferred to the problem of finding the ground states of the corresponding Ising spin form. we experiment with an SAT solver [154] that efficiently traverses the solution space, thus generating typically uncorrelated solutions. *borealis* — a method that works extremely well to solve both weighted and unweighted MAX-SAT problems — is based on parallel tempering Monte Carlo, a standard workhorse in the study of frustrated magnetic systems in statistical physics. The idea is to randomly propose variable changes using a simple Monte Carlo method. In addition to the local updates, the system is replicated at multiple temperatures [49, 132, 155]. Swaps between temperatures are allowed, therefore allowing the system to relax out of local minima and more efficiently sample the solution space. *borealis* is typically not faster than highly-tuned SAT solvers. However, it is a generic method that works relatively well for many SAT-type problems and can produce easily uncorrelated solutions.

6.1.2 Mapping

As we mentioned, for SAT problem the clauses-to-variables ratio α determine if the instances are solvable, which in the language of statistics there are at least two different phases in the instance spaces that are separated by the "order parameter" α. The detailed results can be obtained by looking at the SAT problem in the perspective of statistical physics.

3-SAT (SAT with exact 3 literals in each clause) is the simplest NP model in SAT problem [7], and the existence of a phase transition can provide a heuristic example of applying tools developed in statistical physics of disordered system. The clauses can be uniquely represented by:

$$c_{\mu,i} = \begin{cases} +1 \quad if \quad x_i \in C_\mu \,, \\ -1 \quad if \quad \overline{x_i} \in C_\mu \,, \\ 0 \qquad else \,. \end{cases} \tag{6.2}$$

and the total hamiltonian counts the number of unsatisfied clauses,

$$\mathcal{H} = \sum_{\mu=1}^{\alpha N} \delta_{-k,\sum_i c_{\mu,i}\mathcal{S}_i} \,. \tag{6.3}$$

In this way, 3-SAT has the following Hamiltonian:

$$\mathcal{H} = \frac{\alpha}{8}N - \sum_{i=1}^{N} H_i S_i - \sum_{i\leq j} T_{ij} S_i S_j - \sum_{i\leq j\leq k} J_{ijk} S_i S_j S_k \,. \tag{6.4}$$

For the 3-SAT model, the Hamiltonian has teams up to order 3, which perspectively represent random local fields, two-spin interactions and three-spin interactions of the corresponding disordered statistical model:

$$\begin{aligned} H_i &= \frac{1}{8}\sum_{\mu} c_{\mu,i} \,, \\ T_{ij} &= -\frac{1}{8}\sum_{\mu} c_{\mu,i}c_{\mu,j} \,, \\ J_{ijk} &= \frac{1}{8}\sum_{\mu} c_{\mu,i}c_{\mu,j}c_{\mu,k} \,. \end{aligned} \tag{6.5}$$

This teams can be extended to any k-SAT problem that results in interactions with up to k spins in the corresponding statistical model.

Base on the mapping, the ground states of the Hamiltonian for the statistical model correspond to the optimal solutions for the original Boolean variable instance. In the satisfiable phase of the SAT problem, the ground-state energy is zero while in the unsatisfiable phase the energy of ground states is positive, which means that there is no solution to satisfy all clauses simultaneously. Therefore, if we can determine the ground-state energy as a function of α, we can identify the SAT/UNSAT threshold by using statistical physics. In Gibbs-Boltzmann distribution, each spin configuration is assigned a weight $\exp[1\beta\mathcal{H}]$ with $\beta = 1/T$, which means when β approaches ∞, the states will be more and more concentrated in ground-states. When β reaches the zero-temperature limit $\beta \to \infty$, only ground states maintain. Therefore, for SAT problems we are only interested in the zero-temperature thermodynamic properties of the corresponding statistical models. The ground-state properties in 3-SAT have been studied based on replica-symmetric approximation [8, 156], and the threshold of SAT/UNSAT for general k-SAT is given by [157]:

$$\alpha_c(k) = (2^k - 0.5)\ln 2 - 0.5 + \mathcal{O}(2^{-k}). \tag{6.6}$$

Although the current analytical and numerical tools can give some approximate results, it is still hard for these local algorithms like RandomWlkSAT to get into a deep area - clustering phase, which is insider the SAT phase including a phase with solutions concentrated in a large cluster and a phase with solutions spreading in different small clusters [8].

6.1.3 Implementation of metropolis algorithm with parallel tempering on Boolean satisfiability problem

Here we use the parallel tempering algorithm combined with the Metropolis algorithm as an optimizer to solve the SAT problem, which is to find the ground states of the Hamiltonian

for the corresponding statistical model. As we mentioned, the challenge is from the complex energy landscapes and the current solvers are not able to execute global searching tasks due to the existence of local minima. The parallel tempering algorithm, which is designed to overcome the obstacles from energy landscapes, has already had its success in studying disordered systems like spin glasses, therefore it is a strong candidate for the optimizer to tackle the SAT problem in the perspective of statistical physics. It is described by the pseudo code in (Algorithm 4) (we call it "Borealis"):

Algorithm 4: $Borealis$

```
Input: SAT instance;
Initialize systems with random truth assignments;
for MCS = 1 to maxMCS do
    Metropolis update;
    Parallel tempering update;
    Keep track of lowest energy E_min of all systems;
end
Return E_min;
```

Here are some comments for this implementation. First of all, for the algorithm to be able to work on it, we need to use Eq. (6.3) to build the Hamiltonian for the Ising spin form. Then we set the maximum number of Monte Carlo steps, in which we assume that up to a certain number of Monte Carlo steps all the configuration space has been visited. Inside the Monte Carlo loop, one step of Metropolis plus one step of parallel tempering count as one Monte Carlo step, and the Metropolis algorithm, it is unnecessary to use the whole Hamiltonian to update the configuration. The reason is as follows: assume we are working on a k-SAT instance we only need to pay attention to these clauses that are about to change from "satisfied" to "unsatisfied" or verse Versa. Therefore, for the update, we count the number of unsatisfied clauses with $k-1$ unsatisfied literals that include the variable we

are currently considering, then update this number. In the implementation, we use a two-dimensional array to store this "$k-1$ unsatisfied" clauses with the first index indicating the variable and the second index indicating the position of the corresponding position for the clause. Also, the most interesting ratio is alway around the transition ratio α_c, which means the instances are still close to "solvable", therefore the clauses with $k-1$ unsatisfied literals are way less than other types of clauses. We compared this update method and the whole update method, and this method can save 20% time in terms of finding the ground states for the instances below and close to the threshold [154].

6.1.4 Results on maximum Boolean satisfiability problem

We evaluate our algorithm on a broad range of benchmarks, including unweighted MAX-SAT, partial MAX-SAT, weighted partial MAX-SAT, NAE-MAX-SAT, weighted XOR-MAX-SAT. The MAX-SAT instances comprise the most wide-spread benchmark, including random instances from the Tenth MAX-SAT Evaluation in 2015 [158]. In general, physics-based algorithms perform less well for highly-structured problems. Therefore, in this study, we focus on random-instance benchmarks. To perform a scaling analysis we use the `makewff` generator [159] with minor modifications to generate random MAX-k-SAT ($M/N = 30$) instances and weighted XOR-MAX-2SAT instances ($M/N = 1$) with certain clause-to-variable ratios.

Also we have compared our algorithm to four local search solvers: CCLS, DistUP, Dist1. CCLS combines a configuration checking strategy with a random walk and has won four categories in the incomplete track of the 2015 MAX-SAT Evaluation. Dist is a local search algorithm with a clause weighting scheme and variable selection strategy. It has won the weighted partial random MAX-SAT incomplete track of the 2015 MAX-SAT Evaluation. DistUP combines an assigning procedure PrioUP with the solver Dist and has won the partial random MAX-SAT incomplete track of the 2015 MAX-SAT Evaluation.

We first compare the performance of our algorithm to CCLS, DistUP, and Dist1 from the Tenth MAX-SAT Evaluation (2015) [160, 161]. Simulation parameters used in the experiments with our algorithm are shown in Table. 6.1. We note that the algorithm is typically relatively robust to parameter selection, as long as enough temperatures are used. While this might seem as overhead at first, the speedup obtained by a too large set of temperatures scales faster than the linear increase in effort due to additional temperatures and is typically exponential. It is, however, important to select $maxMCS$ carefully to ensure the solution space is traversed efficiently.

Fig. 6.1, 6.2 and 6.3 show the time to solution (TTS) [162] of our algorithm and CCLS, DistUP and Dist1 as a function category index for unweighted, partial and weighted partial random MAX-SAT instances in Tenth MAX-SAT Evaluation (2015), respectively. Our algorithm finds solutions for all instances and significantly outperforms CCLS, DistUP and Dist1 in most categories. In the partial and weighted partial MAX-SAT benchmark instances, PT greatly benefits from weighting schemes which lower energy barriers without distorting the original solution space. Fig. 6.4 demonstrates that our algorithm scales better than CCLS with large k and ratio M/N.

6.2 Application on not-all-equal-satisfiability-based set membership filter

SAT problem has many important applications. One such application is an SAT filter, a novel application of SAT to the set membership problem, which is applied in many areas, especially in computer engineering. Probabilistic membership filters are a type of data structure designed to quickly verify whether an element of a large data set belongs to a subset of the data. While false negatives are not possible, false positives are. Therefore, the main goal of any good probabilistic membership filter is to have a small false-positive rate while being memory efficient and fast to query. Although Bloom filters are fast to construct, their memory efficiency is bounded by a strict theoretical upper bound. Weaver *et al.* in-

troduced random satisfiability-based filters that significantly improved the efficiency of the probabilistic filters, however, at the cost of solving a complex random satisfiability (SAT) formula when constructing the filter. Here we present an improved SAT filter approach with a focus on reducing the filter building times, as well as query times. Our approach is based on using not-all-equal (NAE) SAT formulas to build the filters, solving these via a mapping to random SAT using traditionally-fast random SAT solvers, as well as bit packing and the reduction of the number of hash functions. Paired with fast hardware, NAE-SAT filters could result in enterprise-size applications.

6.2.1 Set membership filter

The *set membership problem* is ubiquitous. It appears in many industrial [140], computer science [163, 140], and security applications, and finds applicability across many fields of research. The problem is simple to pose: Given a pool of subjects D, and a set of interest $Y \subseteq D$, determine if an element $x \in D$ belongs to Y. In real-world applications the subset Y

Table 6.1: Parameters for the different experiments in unweighted MAX-SAT, partial MAX-SAT, weighted partial MAX-SAT. For each instance category simulated, we perform $maxMCS$ Monte Carlo sweeps for each of the $2N_T$ copies of the system. $T_{\min}$ [$T_{\max}$] is the lowest [highest] temperature simulated, and N_T is the total number of temperatures used in the parallel tempering Monte Carlo method. Reprinted with permission from [154].

Track	category	$T_{\min}$	$T_{\max}$	$N_T(N_c)$	$maxMCS$
unweighted	all	0.05	1.23	25	30000
partial	min2sat	0.10	0.50	80	30000
partial	min3sat	0.10	0.50	80	30000
partial	pmax2sat	0.10	2.05	40	30000
partial	pmax3sat	0.10	2.05	40	30000
weighted partial	abrame	0.10	39.00	40	30000
weighted partial	wmax2sat	0.10	39.00	40	30000
weighted partial	wmax3sat	0.10	39.00	40	30000
weighted partial	wpmax2sat	0.10	23.50	40	30000
weighted partial	wpmax3sat	0.10	7.90	40	30000

is finite, however, it can be very large. Therefore, determining if x is a member of the subset Y can be a computationally expensive task. Some technical terms about set membership problem are explained as follows:

- Let D be a particular domain, from which the element being tested for and the set being tested against will be drawn. Some examples of such a domain may be the collection of all the words in the English dictionary, the collection of all the travelers in an airport, or the collection of all the stars in the sky. Usually, the domain is very large, potentially infinite.

- Then the set membership problem is the following: given an element $x \in D$ and a set of interest $Y \subseteq D$, determine if $x \in Y$, here Y is finite, but potentially very large. If we take the dictionary for an example, D should be the set of all words in the

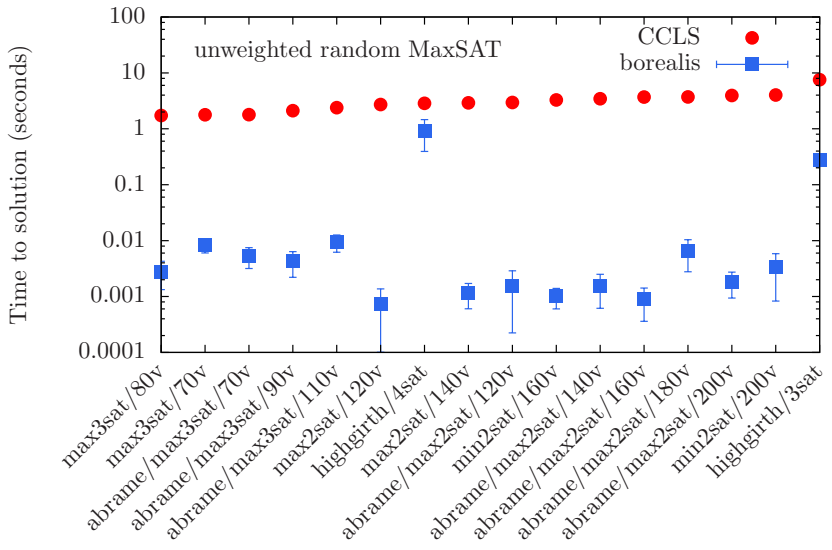

Figure 6.1: Time to solution (CPU time) in seconds of our algorithm and CCLS. The horizontal axis lists the different categories for the unweighted random MAX-SAT track in the Tenth MAX-SAT Evaluation (2015). The time is averaged over all instances in each instance category and error bars for our algorithm are computed via a bootstrap analysis. Our algorithm significantly outperforms CCLS in all categories except for the high-girth instances where it is still faster. Reprinted with permission from [154].

dictionary and Y could be the set of all verbs. Then set membership problem could be to check if a given string of letters is a word defined in Y.

- If the elements of D have a simple representation and also $|Y|$ is small, a very simple approach to set membership testing can be taken: compare the given element x against every member A.

A simple example is the following: Let D be all travelers crossing a country's border in a year and Y be a terrorist watch list. The set membership problem is then to determine if a randomly-screened traveler $x \in D$ is also a member of the watch list $Y \subseteq D$. For a country with few travelers $|Y|$ crossing the border, this task is easily accomplished by listing all members of Y and testing if x is one of them. However, for a large country where millions of travelers across the border each year, verifying that $x \in Y$ can be a time-consuming task. It is therefore desirable to develop a *filter* that quickly verifies if a particular traveler x is on

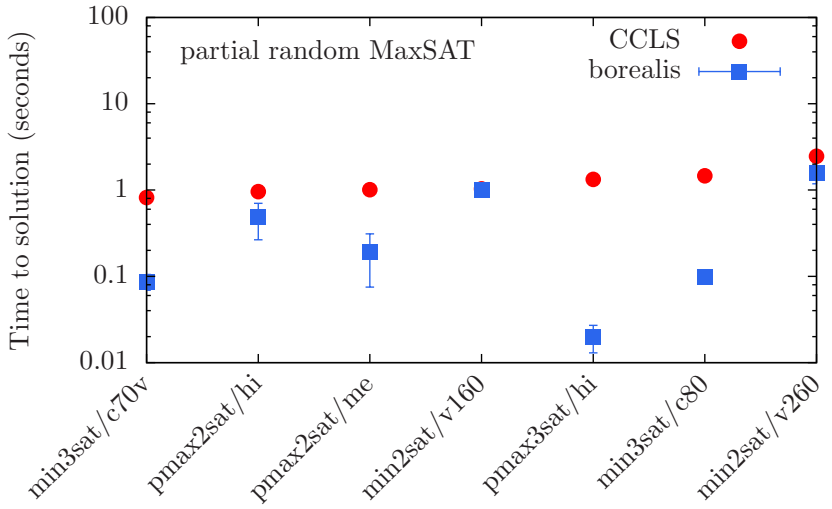

Figure 6.2: Time to solution (CPU time) in seconds of our algorithm and DistUP. The horizontal axis lists the different categories for the partial random MAX-SAT track in the 2015 Tenth MAX-SAT Evaluation. Our algorithm significantly outperforms DistUP in all categories except for the MIN-2SAT instances where the performance is comparable (overlapping symbols). Reprinted with permission from [154].

the watch list Y. In turn, membership filters are not Boolean. If an element $x \in D$ is sent through a filter, it will return either *maybe* or *no*. While *no* here is a definite no, *maybe* is interpreted as a possible presence of $x \in Y$. This means that membership filters have a finite false-positive rate. However, the storage needed to store the filter is considerably smaller than the storage needed to store the set Y. Furthermore, the query time is (ideally) faster than exhaustively searching for x in the set Y. Within the traveler example, this would mean that when travelers are screened using a probabilistic membership filter, a query returning *no* means $x \notin Y$. However, should this not be the case, then x would be sent to secondary screening.

An ideal probabilistic set membership filter should, therefore, be fast, have a small memory footprint, and a low false-positive rate. Traditional workhorses are Bloom filters [164]. These are fast and easy to implement. However, there is an information theoretical up-

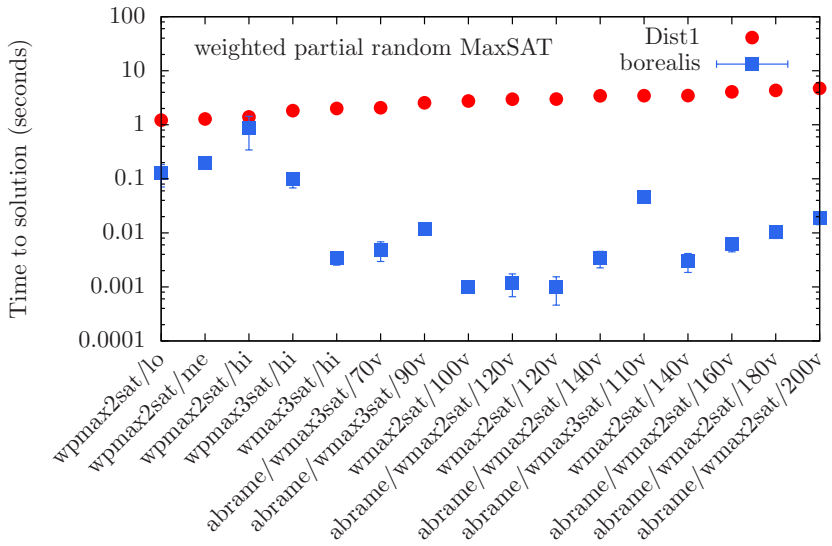

Figure 6.3: Time to solution (CPU time) in seconds of our algorithm and Dist1. The horizontal axis lists the different categories for the weighted partial random MAX-SAT track in the Tenth MAX-SAT Evaluation (2015). Our algorithm significantly outperforms Dist1 in all categories except for the weighted partial MAX-2SAT instances with medium clauses to variables ratio where the performance is comparable. Reprinted with permission from [154].

per bound on their memory efficiency. More recently, Weaver *etal.* [3], introduced random satisfiability-based membership filters that significantly improved efficiency. At the core of the filter lies the solution of a complex satisfiability (SAT) formula [165] needed to construct the actual filter. In this work, we present a variation of Weaver *et al.*'s SAT filter approach with a focus on improving the filter building, as well as query times. Our approach is based on using not-all-equal (NAE) SAT formulas to build the filters, as well as bit packing and reduction of the hash functions to reduce the query times. Also, NAE-SAT filters have excellent memory efficiency and are fast, therefore ideally suited for deployments on large-scale applications.

6.2.2 Reminder — probabilistic Bloom filters

Probably the widest used probabilistic membership filters are Bloom filters [164]. Let D be any set and $Y \subseteq D$ with $m = |Y|$. We assume that the available memory for the filter B_Y is n bits. Select a hash function h: $D \to \mathbb{Z}$ that maps the elements in D to the

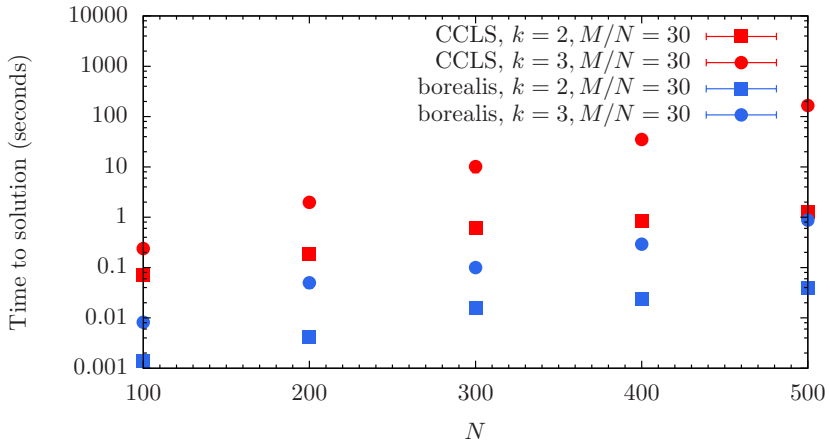

Figure 6.4: Time to solution (CPU time) in seconds for our algorithm and CCLS as a function of system size N for random unweighted MAX-SAT instances. The time is averaged over 100 instances for a given system size N. Error bars are computed using a bootstrap analysis and are smaller than the symbols. Our algorithm outperforms CCLS for all system sizes and scales better with large k and ratio $M/N = 30$. Reprinted with permission from [154].

range [0:n) uniformly and randomly. After having chosen the hash function, the bits of B_Y must be initialized to 0. Then, for an element in $y \in Y$, we set the bit at $h(y)$ to 1, i.e., $B_Y[h(Y)] \equiv 1$. To store all $y \in Y$, store all elements of Y one by one. In most implementations there are multiple hash functions h_1, h_2, ... h_k. In that case $h_1(y)$, $h_2(y)$, ... $h_k(y)$ must be computed first. Once that is completed, the bits of B_Y are all set to 1 at the respective locations (for a pseudo-code version of the full Bloom filter algorithm, see, for example, Ref. [3]). To query the filter with an element $x \in D$, simply verify that the bits at all the locations $h_1(x)$, $h_2(x)$, ... $h_k(x)$ are set to 1. Only when the bits at those locations are 1, will the filter return a *maybe*, otherwise the filter returns a definite *no*. In the latter case $x \notin Y$. While Bloom filters are relatively fast, there is a information-theoretical limit to their memory efficiency [3]. Therefore, fast probabilistic membership filters with a better memory footprint are desirable.

6.2.3 Satisfiability-based set membership filter

There are two ways to build an SAT filter. Here we only discuss the single-instance filter. For details on how to build a filter with more than one instance see Ref. [3]. The following steps are needed to build a probabilistic membership filter based on SAT formulas:

- **Build a CNF** — We use a set of hash functions h_1, h_2, ..., h_k to create a random clause C_Y with k literals for each $y \in Y$ with $|Y| = m$. This means there are m clauses, which make a random k-SAT instance (CNF) X_Y. During the process, it is important to ensure that all literals are different in each clause. Furthermore, the clause-to-variable ratio m/n should not be too high such that the CNF is not in the unsatisfied regime.

- **Find solutions to the CNF** – Once the CNF has been constructed, multiple uncorrelated solutions are needed to construct a good filter. It is important to use SAT formulas for which efficient solvers are known to speed up this step of the filter building process.

- **Store the filter** — several (ideally uncorrelated) solutions, these are stored in an array as the filter. Note that the storage requirements for the filter are considerably smaller than the original data.

Note that SAT filters do not allow insertions after they have been built, i.e., adding a new element to Y will require a new filter to be constructed.

To query the filter, the first step is to generate a clause C_x from the elements $x \in D$ using the hash functions used to generate the random clause in the first step of the filter construction process. Then one has to verify if C_x is satisfied by the filter, i.e., *all* solutions stored. If so, the filter returns *maybe*. However, if C_x is not satisfied by any of the solutions, the filter returns a definite *no*. A filter is characterized by its false-positive rate, which should be as low as possible, its memory efficiency, as well as ideally short build and query times.

Probabilistic membership filters have a finite false positive rate (FPR). This means that for an element x that is not in the set of interest Y the filter might still return a *maybe* result. The FPR for SAT filters is equivalent to the probability that a random k-SAT clause can be satisfied by a specific solution. For a random k-SAT clause, the probability that the clause can definitely not pass one regular solution is 2^{-k}. Therefore, for a single solution to the CNF, the FPR is $(1 - 2^{-k})$. This can be improved, by using $s > 1$ solutions to the CNF, i.e., for s solutions

$$p_{\mathrm{SAT}} = (1 - 2^{-k})^s. \tag{6.7}$$

Using $s > 1$ solutions reduces the FPR, but increases both query times and storage requirements by a factor s. Note that if the solutions are correlated, then the FPR might not be reduced by increasing the number of solutions. Therefore, it is imperative to use a SAT solver that produces as uncorrelated solutions as possible (i.e., with a large hamming distance). We note that for the special case of building SAT filters with NAE-SAT formulas

Eq. (6.7) changes to

$$p_{\mathrm{NAE}} = (1 - 2^{-k+1})^s. \tag{6.8}$$

The memory efficiency of a probabilistic membership filter is defined as the number of filter bits required per keyword item. As introduced in [3], the memory efficiency ξ for a SAT filter is given by

$$\xi = \frac{-\log_2 p}{n/m}. \tag{6.9}$$

Here, n is the number of memory bits and $m = |Y|$. For a SAT filter that uses s solutions, one needs sn memory bits and therefore the efficiency is given by

$$\xi_{\mathrm{SAT}} = \frac{-\log_2 p_{\mathrm{SAT}}}{sn/m} = \frac{-\log_2(1 - 2^{-k})}{n/m}. \tag{6.10}$$

Again, for the special case that uses NAE-SAT solutions, Eq. (6.10) changes to

$$\xi_{\mathrm{NAE}} = \frac{-\log_2(1 - 2^{-k+1})}{n/m}. \tag{6.11}$$

In Ref. [3], it can be rigorously shown that SAT filters can achieve a theoretical efficiency of 1 (i.e., 100%). This is to be contrasted to Bloom filters that have an information-theoretical upper bound for the efficiency, namely $\xi_{\mathrm{Bloom}} \leq \log 2 \approx 0.693$. This is the main reason why SAT filters are more desirable than Bloom filters.

In Ref. [3] it is shown that while query times for SAT filters are short, they are still larger than for Bloom filters. Furthermore, the construction of an SAT filter requires multiple uncorrelated solutions to a CNF. While efficient SAT solvers exist, some tend to produce correlated solutions, which thus means that there is potentially a large overhead in finding as uncorrelated solutions (i.e., with a large hamming distance) as possible. In what follows we demonstrate how filters designed using NAE-SAT formulas have considerably shorter build, as well as query times. Furthermore, we show how an NAE-SAT CNF can be reformulated

into an SAT formula such that one can take advantage of state-of-the-art random k-SAT solvers. Finally, we show that only one hash function is needed, thus optimally speeding up build and query times.

As we know, a low FPR is of utmost importance for probabilistic membership filters. In the case of SAT filters, the theoretical FPR [Eq. (6.7)] can only be reached if the solutions to the underlying SAT formula are *uncorrelated*. Because the solutions are from a *single* SAT formula, the probability that these are correlated is high unless a typically large effort to find enough uncorrelated solutions is performed. Here we work around this bottleneck by replacing standard SAT formulas with NAE-SAT formulas, because the solutions to NAE-SAT problems have large Hamming distances (typically around 50% of the number of variables) and are therefore far less correlated [147]. In our approach, we randomly select NAE-SAT solutions to build the filter. Because we never know which solution in the pair was selected, we can state statistically that the *average* hamming distances would be around 50% of the number of variables by construction. This saves considerable time when building the probabilistic filter. From now on, unless otherwise specified, we use NAE-SAT solutions.

We first analyze the performance of NAE-SAT filters using traditional random k-SAT solvers and then show results using *borealis*. Note that we use similar parameters as used in Ref. [3] to be able to perform a direct comparison between the results of Ref. [3] on traditional SAT filters and our NAE-SAT implementation shown here. To be able to use *borealis* for NAE-SAT instances, we have modified the original NAE-SAT CNF into an SAT CNF such that by construction all solutions satisfy the NAE requirement. This is accomplished by adding a penalty clause to rule out the all-satisfied assignments to each clause in the original CNF. In the penalty clause, all literals are complementary to the original clause. Then, as long as the solver finds a solution, the solution is an NAE solution to the original CNF. For

example,

$$(x_3 \vee x_{18} \vee \overline{x_{12}} \vee x_5) \rightarrow (x_3 \vee x_{18} \vee \overline{x_{12}} \vee x_5) \wedge (\overline{x_3} \vee \overline{x_{18}} \vee x_{12} \vee \overline{x_5}). \tag{6.12}$$

The term on the right now satisfies the NAE constraint and can be handled by SAT solvers, including*borealis*. Because *borealis* [154] is designed to tackle statistical physics problems, we need to cast the CNF of the NAE-SAT formula as a physical Hamiltonian (cost function). We use the number of unsatisfied clauses as a simple cost function for the n Boolean variables in the CNF. If a cost of 0 is found (in physics, the ground state energy), the configuration represents a valid variable assignment to the CNF. Details on the algorithm and its implementation can be found in Ref. [154].

Table 6.2: Build time in seconds, memory size in bytes and false-positive rate (FPR) in percent for the k-SAT filter studied in Ref. [3]. The average hamming distance is at least 50%. Simulations were performed on a 2009 MacBook Pro with a 3.06 GHz processor. s represents the number of SAT solutions used to build the filter.

Filter size	Build time (s)	Size (bytes)	FPR (%)
$k = 4$, $s = 22$	20802	44748	24.20
$k = 5$, $s = 44$	610	44000	24.86
$k = 6$, $s = 88$	643	44144	25.09

Fig. 6.5 shows the memory efficiency ξ_{NAE} vs the FPR p_{NAE}. Using formulas with $m/n = 10.1$ and increasing the filter size m eventually has little effect on the efficiency. However, for increasing m the FPR p_{NAE} can be reduced to arbitrarily-low values; here below 10^{-5}. The solid horizontal (green) line represents the optimal bound which can easily be achieved with little numerical effort. Note that the solutions were generated using Dimetheus [143] because the ratio is close to the threshold.

Table. 6.3 lists our results for the same instances studied in Table. 6.2, which shows that NAE-SAT filters is still considerably faster than k-SAT filters. We do note, however, that *borealis* works reasonably well for a broad range of α values unlike traditional SAT solvers

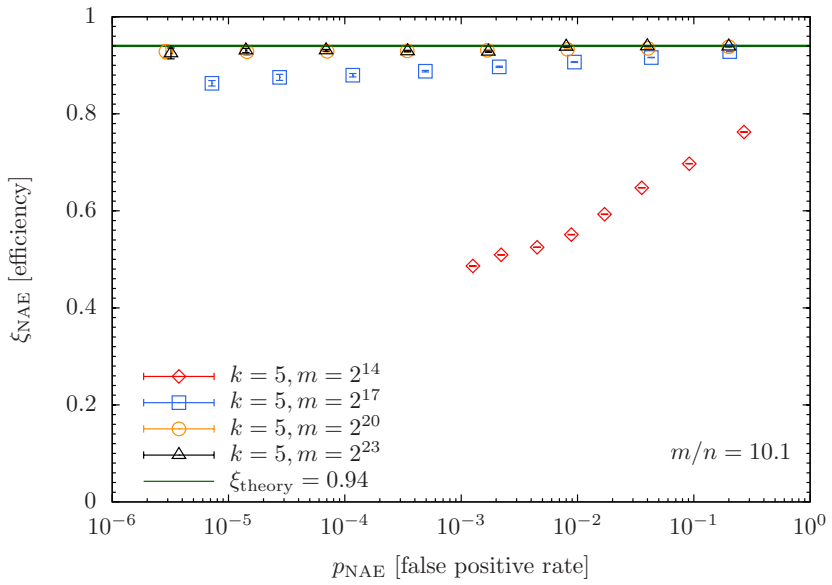

Figure 6.5: Efficiency ξ_{NAE} as a function of false-positive rate p_{NAE} for a k-NAE-SAT filter. As m increases for fixed k, the efficiency approaches the theoretical bound (solid horizontal line). Note that very low FPRs can be obtained. Instances generated with Dimetheus.

Table 6.3: Build time in seconds, memory size in bytes and false-positive rate (FPR) in percent for the k-NAE-SAT filter case study using *borealis*. By design, the average Hamming distance is around 50%. Simulations were performed on a 2013 MacBook Pro with a 2.60 GHz processor. s represents the number of NAE-SAT instances used to build the filter.

Filter size	Build time(s)	Size(bytes)	FPR(%)
$k = 4$, $s = 11$	6.2	44748	23.00
$k = 5$, $s = 22$	11.0	44000	24.45
$k = 6$, $s = 44$	17.8	44144	25.10

that are tuned to specific regimes of α values. Fig. 6.6 shows the efficiency ξ vs FPR p for a fixed filter size $m = 2^{14}$ and different values of $\alpha = m/n$ ($k = 5$). *borealis* performs reasonably well for a broad range of α values with $8 \lesssim \alpha \lesssim 19 < \alpha_c \approx 21.11$. There is a decrease of the efficiency ξ for small FPRs. This is because m is relatively small and therefore the number of uncorrelated solutions is accordingly small. As such, finding a set s of many uncorrelated solutions is difficult. This problem is referred to as *finite-size effect* in physics and is easily alleviated by increasing the filter size. For large n/s the finite-size effects become negligible.

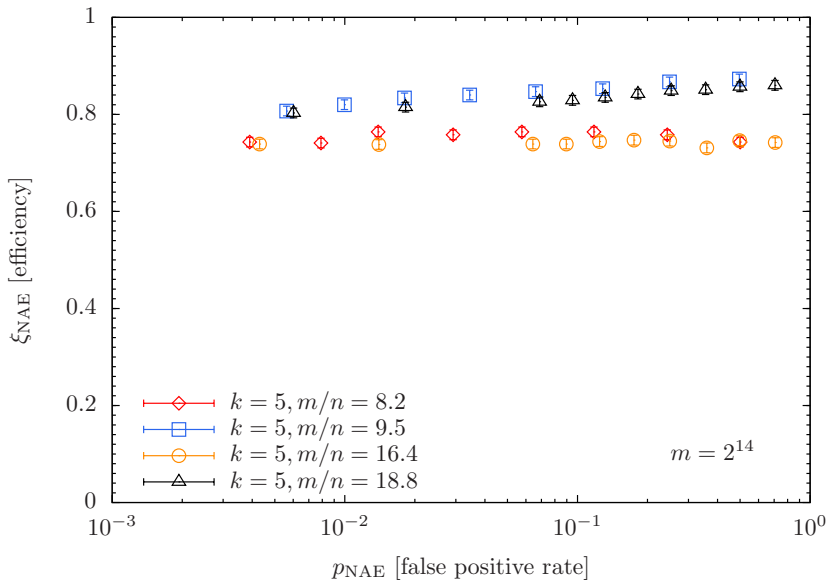

Figure 6.6: Efficiency ξ_{NAE} as a function of false-positive rate p_{NAE} for a k-NAE-SAT filter. Data computed with *borealis* for different values of $\alpha = m/n$. Finite-size effects (see text) are visible due to the small filter size used. However, *borealis* is an efficient solver for a broad range of α values.

The last step for the filter is the query, To speed up query times for the NAE-SAT filter we use bit packing. Given the 64-bit architecture of the benchmark machine, this means that 64 solutions can be handled in parallel. For the benchmarks, we use a single-core 2013

MacBook Pro with a 2.60GHz processor and 8Gb RAM. We query 2^{17} 64-bit strings. The hash function used is `MurmurHash3` [2]. Note that we deviate from the approach used in Ref. [3], because, as Fig. 6.7 shows, the difference between the FPR using one or two hashes is negligible. We did simply change the seed in `MurmurHash3` to achieve these results. This is yet another advantage of our implementation that speeds up query times.

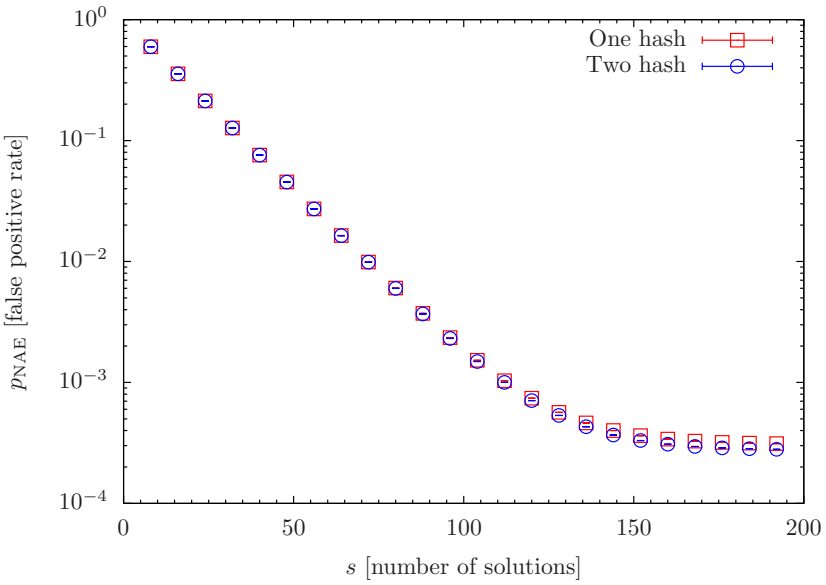

Figure 6.7: FPR p as a function of the number of solutions s using one, or two-hash functions in `MurmurHash3` [2]. The difference between both approaches is $\sim 10^{-5}$, i.e., negligible.

Fig. 6.8 shows that the query times are approximately similar for different values of s, as long as there are more than a certain number of solutions. The jump in the data might be due to buffering issues in the bit packing.

6.2.4 Summary

By using NAE-SAT problems for the construction of SAT filters as first mentioned in Ref. [3], filter build times can be considerably reduced. Using NAE-SAT formulas to build

the filters has the advantage that, by design, the solutions tend to be uncorrelated. Furthermore, we show how the NAE-SAT constraint can be accommodated into a random SAT formula such that standard SAT solvers can be used. Query times in our implementation are reduced via bit packing and the use of a single hash function. Finally, by using physics-inspired algorithms such as *borealis* the filter construction can be further parallelized and improved further because the algorithm efficiently traverses the solution space.

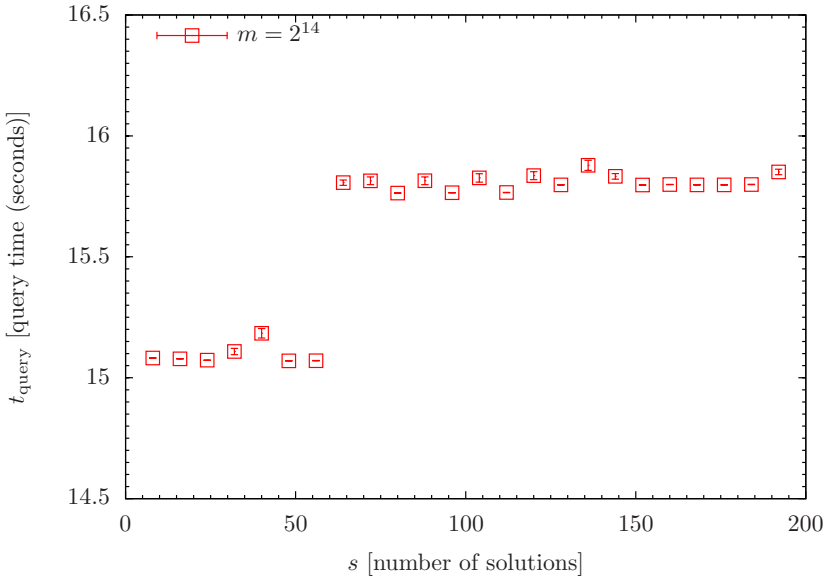

Figure 6.8: Query time as a function of the number of used instances s for a NAE-SAT filter with $k = 5$, $m = 2^{17}$, and $n = 16282$. Queried are 2^{17} 64-bit strings. Time t is measured in seconds.

7. CONCLUSION AND OUTLOOK

In this dissertation, I introduced my interdisciplinary study of algorithms that are applied to and derived from spin glasses along with the physical properties of Ising spin models. The motivation is two-folded. On one hand, the fact that the Ising spin form has strong connections with research areas such as combinational optimization [8] and complexity theory [166] gives the motivation to deeply study the physical properties such as critical behavior, which is closely related to some NP-hard problems such as Boolean satisfiability problem [166, 3]. Also, it has already been proved that the Ising spin form is NP-complete [16], therefore Ising spin form can be a potential playground to build solvers for these NP problems [33, 167, 166] by applying the well-developed algorithms in spin glasses to these real-world problems. On the other hand, the fast development of the software in computer science, especially in the algorithms for artificial intelligence [168, 169], can generally offer new algorithms for problems in spin glasses, especially for these classification problems such as detecting phase transitions [25, 26, 27].

One of these most successful applications of artificial intelligence to condensed matter physics is using convolutional neural networks to detect phase transitions. In Chapter 3, I demonstrated a systematic approach to quantitatively detect the phase transition of the three-dimensional Edward-Anderson model with Bimodal disorder using Gaussian disorder. We successfully find the phase transition temperature at ~ 1.122, which agrees with the result obtained from Monte Carlo [83]. In this work, making a training model. Using different models in training and prediction is practical because this approach can be a tool to study new systems based on the data from well-known systems. More than that, I also showed the importance of training. It shows that either training data from a non-equilibrium system or a small portion of mixture between two classes ($\sim 1\%$) can give an unreliable prediction on

the phase transition temperature. For future research, I think there are two directions can be further studied. The first one is to test if this approach still works for two models belonging to different universal classes, which is also a task to test if convolutional neural networks are capable of recognizing the phases between different types of disorders. This extention can not only help us study the difference between the phases of models from different universality classes through the parameters of the networks but also help us test the limit of neural networks. The second direction is to develop more sophisticated convolutional networks to gain more accurate estimation of phase transition in spin glasses, which can be a promising candidate to replace Monte Carlo since we already showed that fewer instances are needed in this new approach for results with the same accuracy.

In Chapter 4, I systematically explored the extension of ICM to topologies with high connectivities. In previous work [109], the ICM move is proved to work very well on graphs with low connectivities because a large cluster reduces the efficiency of the multi-spin update procedure proposed by ICM. Inspired by the study on ICM, first I proposed to cut the cluster to a size that is similar to the two-dimensional model, in which ICM works very well. But a further simulation showed that as long as the detailed balance exists, the trade-off between cluster size and the acceptance probability could cause the size of the final accepted cluster to be relatively small so that the cluster update is no different from the single-spin update. However, the detailed balance is not necessary for optimization problems, which inspired me to modify the ICM by cutting the cluster size without performing the detailed balance. Although this reject-free cluster update is similar to ICM, the total energy changes. We found that this cutting could equilibrate the system faster than ICM, but the final distribution is probably not Gibbs-Boltzmann distribution. To find out if this modified ICM can bring speed-up for the highly connected graphs, I simulated the time to solution (TTS) on a planted model [1] for both ICM and modified ICM. The results show that ICM has an advantage over modified ICM on both Monte Carlo sweeps and wall clock time. For future research, I think

it is worthy to look for a local reject-free cluster update that remains the total energy, which could be a similar cluster move with ICM except that it is local so that it can avoid the trouble of flipping the entire replica.

Chapter 5 is dedicated to the study of the Ising model with bond dilution. This work is inspired by [129], which shows that the two-dimensional Ising model with site dilution agrees with the strong universality scenario with logarithmic correction. In our work, we realized the disorder by implementing the bond dilution in the two-dimensional Ising model. For the implementation of the dilution, we eliminated all the zero-dimensional clusters to avoid the impact on the estimation of the critical exponents. We found that the critical exponent of the correction to the correlation length does not vary with the disorder, which supports the strong universality scenario. But a different asymptotical behavior of $V_{4\infty}$ was found. In the future, we still need to investigate the reason for the existence of different $V_{4\infty}$, which still put the debate between strong and weak universality scenarios in the air.

Chapter 6.1 is my research on an intuitive case of solving the SAT problem using the Ising spin form. First of all, we map the SAT instance to a Hamiltonian of an Ising spin model, which is a one-on-one mapping without overhead [166]. Then we use the Metropolis algorithm combined with the parallel tempering algorithm to simulate the thermodynamic procedure to find the ground states of the Hamiltonian, which can be mapped back to the solutions for the original SAT instance. This approach is global because thermodynamics of the Ising spin model is ergodic. It works extremely well on the MAX-SAT problem because not like the SAT problem whose solutions always have specific symmetry solutions, MAX-SAT have more degree of freedom so that they are distributed globally in the configuration space. Based on the Metropolis algorithm and parallel tempering algorithm, a solver is developed generally for SAT and SAT related problems, which shows excellent performance on MAX-SAT instances. This solver is further adapted to help solve the instances from the SAT filter. By modifying the SAT filter to the NAE form, we developed a Not-all-equal-

SAT filter (NAE-SAT filter) that can naturally generate instances whose solutions have large hamming distance, which can significantly reduce the false positives for the filter. Combined with the solver's ability to globally seek solutions, the NAE-SAT filter can have both high efficiency and low false positives, which makes NAE-SAT filter a promising candidate to compete with the traditional Bloom filter [170]. For future work, I think we can improve the performance of the solver by introducing parallel computing such as multi-threads, which can be very efficient for parallel tempering algorithm since the replicas in parallel tempering are independent of each other. Other than that because usually the instances from the SAT problem are highly connected so that global cluster move like ICM does not work well, we can introduce the local cluster update to speed up the thermodynamics as mentioned in the last paragraph.

To sum up, this body of work shows a connection between spin glasses and computer science in terms of algorithms. First I demonstrated that Monte Carlo simulation is not the only reliable method to study disordered systems. We can also study disordered systems using artificial intelligence like machine learning by looking into the macroscopic properties of the graph of configurations. On the algorithm side, I showed that a global multi-spin cluster update can not be effective for the highly connected graph as long as the detailed balance is considered. It tells that future work could be focused on looking for a local reject-free cluster update that can avoid the trade-off obstacle caused by detailed balance. Secondly, I studied the approach of solving the SAT problem using the Ising spin form. The success of the Metropolis algorithm combined with the parallel tempering algorithm on the SAT problem shows the power of global algorithms from spin glasses on optimization. We already know that the Ising spin form is NP-complete and the mapping is one-on-one without any overhead, therefore by mapping to the Ising spin form, statistical physics can provide a new platform for studying NP-problems. For example, we can use phase transitions in the Ising spin form to study the transitions of the hardness of SAT instances. .

REFERENCES

[1] F. Hamze, D. C. Jacob, A. J. Ochoa, D. Perera, W. Wang, and H. G. Katzgraber, "From near to eternity: Spin-glass planting, tiling puzzles, and constraint-satisfaction problems," *Phys. Rev. E*, vol. 97, p. 043303, Apr 2018.

[2] A. Appleby. https://github.com/aappleby/smhasher/tree/master/src.

[3] S. A. Weaver, K. J. Ray, V. W. Marek, A. J. Mayer, and A. K. Walker, "Satisfiability-based set membership filters," *Journal on Satisfiability, Boolean Modeling and Computation (JSAT)*, vol. 8, p. 129, 2014.

[4] K. Binder and A. P. Young, "Spin Glasses: Experimental Facts, Theoretical Concepts and Open Questions," *Rev. Mod. Phys.*, vol. 58, p. 801, 1986.

[5] S. F. Edwards and P. W. Anderson, "Theory of spin glasses," *J. Phys. F: Met. Phys.*, vol. 5, p. 965, 1975.

[6] H. Kawamura and T. Taniguchi, *Chapter 1 - Spin Glasses*, vol. 24 of *Handbook of Magnetic Materials*. Elsevier, 2015.

[7] A. K. Hartmann and H. Rieger, *Optimization Algorithms in Physics*. Berlin: Wiley-VCH, 2001.

[8] A. K. Hartmann and H. Rieger, *New Optimization Algorithms in Physics*. Berlin: Wiley-VCH, 2004.

[9] S. Boixo, T. Albash, F. M. Spedalieri, N. Chancellor, and D. A. Lidar, "Experimental signature of programmable quantum annealing," *Nat. Commun.*, vol. 4, p. 2067, 2013.

[10] H. G. Katzgraber, F. Hamze, and R. S. Andrist, "Glassy Chimeras Could Be Blind to Quantum Speedup: Designing Better Benchmarks for Quantum Annealing Machines," *Phys. Rev. X*, vol. 4, p. 021008, 2014.

[11] Z. Zhu, A. J. Ochoa, F. Hamze, S. Schnabel, and H. G. Katzgraber, "Best-case performance of quantum annealers on native spin-glass benchmarks: How chaos can affect success probabilities," *Phys. Rev. A*, vol. 93, p. 012317, 2016.

[12] G. Santoro, E. Martoňák, R. Tosatti, and R. Car, "Theory of quantum annealing of an Ising spin glass," *Science*, vol. 295, p. 2427, 2002.

[13] D. A. Lidar and O. Biham, "Simulating ising spin glasses on a quantum computer," *Phys. Rev. E*, vol. 56, pp. 3661–3681, Sep 1997.

[14] D. J. Amit, H. Gutfreund, and H. Sompolinsky, "Storing infinite numbers of patterns in a spin-glass model of neural networks," *Phys. Rev. Lett.*, vol. 55, pp. 1530–1533, Sep 1985.

[15] S.-I. Amari and K. Maginu, "Statistical neurodynamics of associative memory," *Neural Networks*, vol. 1, no. 1, pp. 63 – 73, 1988.

[16] lucas:14, "Ising formulations of many NP problems," *Front. Physics*, vol. 12, p. 5, 2014.

[17] A. K. Hartmann, "Calculation of Partition Functions by Measuring Component Distributions," *Phys. Rev. Lett.*, vol. 94, p. 050601, 2005.

[18] A. P. Young, ed., *Spin Glasses and Random Fields*. Singapore: World Scientific, 1998.

[19] R. H. Swendsen and J.-S. Wang, "Replica Monte Carlo simulation of spin-glasses," *Phys. Rev. Lett.*, vol. 57, p. 2607, 1986.

[20] E. Marinari and G. Parisi, "Simulated tempering: A new Monte Carlo scheme," *Europhys. Lett.*, vol. 19, p. 451, 1992.

[21] F. Wang and D. P. Landau, "An efficient, multiple-range random walk algorithm to calculate the density of states," *Phys. Rev. Lett.*, vol. 86, p. 2050, 2001.

[22] W. Wang, J. Machta, and H. G. Katzgraber, "Chaos in spin glasses revealed through thermal boundary conditions," *Phys. Rev. B*, vol. 92, p. 094410, 2015.

[23] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, "Efficient Cluster Algorithm for Spin Glasses in Any Space Dimension," *Phys. Rev. Lett.*, vol. 115, p. 077201, 2015.

[24] H. G. Katzgraber *et al.*, "Universality in three-dimensional Ising spin glasses: A Monte Carlo study," *Phys. Rev. B*, vol. 73, p. 224432, 2006.

[25] J. Carrasquilla and R. G. Melko, "Machine learning phases of matter," *Nature Physics*, vol. 13, pp. 431–414, 2017.

[26] J. Carrasquilla, K. Ch'ng, R. G. Melko, and E. Khatami, "Machine learning phases of strongly correlated fermions," *Phys.Rev X*, vol. 7, 2017.

[27] A. Tanaka and A. Tomiya, "Detection of phase transition via convolutional neural network," *J. Phys. Soc. Jap.*, vol. 86, no. 6, p. 063001, 2017.

[28] R. E. Ladner, "On the Structure of Polynomial Time Reducibility," *J. ACM*, vol. 22, p. 155, 1975.

[29] R. M. Karp, *Complexity of Computer Computations*, ch. Reducibility among Combinatorial Problems, p. 85. New York: Plenum, 1972.

[30] S. A. Cook, "The Complexity of Theorem-proving Procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, (New York, NY, USA), p. 151, ACM, 1971.

[31] S. Istrail, "Statistical mechanics, three-dimensionality and np-completeness: I. universality of intracatability for the partition function of the ising model across non-planar surfaces (extended abstract)," in *STOC*, 2000.

[32] F. Barahona, "On the computational complexity of Ising spin glass models," *J. Phys. A*, vol. 15, p. 3241, 1982.

[33] Y. Fu and P. W. Anderson, "Application of statistical mechanics to NP-complete problems in combinatorial optimisation," *Journal of Physics A: Mathematical and General*, vol. 19, pp. 1605–1620, jun 1986.

[34] M. Mezard and A. Montanari, *Information, Physics, and Computation*. New York, NY, USA: Oxford University Press, Inc., 2009.

[35] E. Ising, "Beitrag zur Theorie des Ferromagnetismus," *Z. Phys.*, vol. 31, p. 253, 1925.

[36] S. G. BRUSH, "History of the lenz-ising model," *Rev. Mod. Phys.*, vol. 39, pp. 883–893, Oct 1967.

[37] K. Huang, *Statistical Mechanics*. New York: Wiley, 1987.

[38] S. F. Edwards and P. W. Anderson, "Theory of spin glasses," *Journal of Physics F Metal Physics*, vol. 5, pp. 965–974, May 1975.

[39] G. Jaeger, "The ehrenfest classification of phase transitions: Introduction and evolution," *Archive for History of Exact Sciences*, vol. 53, pp. 51–81, 05 1998.

[40] H. Nishimori, *Statistical Physics of Spin Glasses and Information Processing: An Introduction*. New York: Oxford University Press, 2001.

[41] L. Onsager, "Crystal Statistics. I. A Two-Dimensional Model with an Order-Disorder Transition," *Phys. Rev.*, vol. 65, p. 117, 1944.

[42] L. D. Landau *Phys. Z. Sowjetunion*, vol. 11, p. 26, 1937.

[43] D. Panchenko, "The sherrington-kirkpatrick model: An overview," *Journal of Statistical Physics*, vol. 149, pp. 362–383, Oct 2012.

[44] H. Nishimori and G. Ortiz, "Elements of phase transitions and critical phenomena," 01 2011.

[45] M. E. Fisher, "The theory of equilibrium critical phenomena," *Reports on Progress in Physics*, vol. 30, pp. 615–730, jul 1967.

[46] V. Privman, ed., *Finite Size Scaling and Numerical Simulation of Statistical Systems*. Singapore: World Scientific, 1990.

[47] K. Binder, "Critical properties from Monte Carlo coarse graining and renormalization," *Phys. Rev. Lett.*, vol. 47, p. 693, 1981.

[48] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys.*, vol. 21, p. 1087, 1953.

[49] K. Hukushima and K. Nemoto, "Exchange Monte Carlo method and application to spin glass simulations," *J. Phys. Soc. Jpn.*, vol. 65, p. 1604, 1996.

[50] H. G. Katzgraber, "Introduction to Monte Carlo Methods." (arXiv:0905.1629), 2009.

[51] N. Rathore, M. Chopra, and J. J. de Pablo, "Optimal allocation of replicas in parallel tempering simulations," *J. Chem. Phys.*, vol. 122, p. 024111, 2005.

[52] J. Houdayer, "A cluster Monte Carlo algorithm for 2-dimensional spin glasses," *Eur. Phys. J. B.*, vol. 22, p. 479, 2001.

[53] Wikipedia contributors, "Optimization problem — Wikipedia, the free encyclopedia," 2019. [Online; accessed 5-October-2019].

[54] Wikipedia contributors, "Lagrange multiplier — Wikipedia, the free encyclopedia," 2019. [Online; accessed 5-October-2019].

[55] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[56] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 ed., 2009.

[57] Wikipedia contributors, "Convex optimization — Wikipedia, the free encyclopedia," 2019. [Online; accessed 5-October-2019].

[58] Wikipedia contributors, "Karush−kuhn−tucker conditions — Wikipedia, the free encyclopedia," 2019. [Online; accessed 5-October-2019].

[59] F. Santosa and W. W. Symes, "Linear inversion of band-limited reflection seismograms," *SIAM J. Sci. Stat. Comput.*, vol. 7, pp. 1307–1330, Oct. 1986.

[60] R. Tibshirani, "Regression shrinkage and selection via the lasso," *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B*, vol. 58, pp. 267–288, 1994.

[61] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *Trans. Sig. Proc.*, vol. 58, pp. 5262–5276, Oct. 2010.

[62] J. Quinonero Candela and C. Rasmussen, "A unifying view of sparse approximate gaussian process regression," *Journal of Machine Learning Research*, vol. 6, pp. 1935–1959, Dec. 2005.

[63] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2nd ed., 2014.

[64] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge University Press, 2014.

[65] D. Mehta and V. Raghavan, "Decision tree approximations of boolean functions," *Theor. Comput. Sci.*, vol. 270, pp. 609–623, Jan. 2002.

[66] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Application in R* . Springer Press, 2013.

[67] C. wei Hsu, C. chung Chang, and C. jen Lin, "A practical guide to support vector classification," 2010.

[68] J. C. Platt, "Advances in kernel methods," ch. Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pp. 185–208, Cambridge, MA, USA: MIT Press, 1999.

[69] A. Widodo and B.-S. Yang, "Support vector machine in machine condition monitoring and fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 21, pp. 2560–2574, 08 2007.

[70] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *Proceedings of the 10th European Conference on Machine Learning*, ECML'98, (Berlin, Heidelberg), pp. 137–142, Springer-Verlag, 1998.

[71] Y. LeCun and Y. Bengio, "The handbook of brain theory and neural networks," ch. Convolutional Networks for Images, Speech, and Time Series, pp. 255–258, Cambridge, MA, USA: MIT Press, 1998.

[72] W. Zhang, K. Itoh, J. Tanida, and Y. Ichioka, "Parallel distributed processing model with local space-invariant interconnections and its optical architecture," *Appl. Opt.*, vol. 29, pp. 4790–4797, Nov 1990.

[73] C. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer-Verlag, 2006.

[74] Y. Tian, S. Pei, K ann Jana, and B. Ray, "DeepTest: Automated Testing of Deep-Neural-Network-driven Autonomous Cars," *CoRR*, vol. abs/1708.08559, 2017.

[75] S. J. Wetzel and M. Scherzer, "Machine learning of explicit order parameters: From the Ising model to SU(2) lattice gauge theory," *Physical Review B*, vol. 96, p. 184410, Nov. 2017.

[76] L.-F. Arsenault, A. Lopez-Bezanilla, O. A. von Lilienfeld, and A. J. Millis, "Machine learning for many-body physics: The case of the Anderson impurity model," *Physical Review B*, vol. 90, p. 155136, Oct. 2014.

[77] C. Deans, L. D. Griffin, L. Marmugi, and F. Renzoni, "Machine Learning Based Localization and Classification with Atomic Magnetometers," *Physical Review Letters*, vol. 120, p. 033204, Jan. 2018.

[78] P. Mehta and D. J. Schwab, "An exact mapping between the Variational Renormalization Group and Deep Learning," *arXiv:1410.3831 [cond-mat, stat]*, Oct. 2014.

[79] A. G. Kusne, T. Gao, A. Mehta, L. Ke, M. C. Nguyen, K.-M. Ho, V. Antropov, C.-Z. Wang, M. J. Kramer, C. Long, and I. Takeuchi, "On-the-fly machine-learning for high-throughput experiments: search for rare-earth-free permanent magnets," *Scientific Reports*, vol. 4, p. 6367, Sept. 2014.

[80] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[81] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1746–1751, 2014.

[82] H. Nishimori and K. Nemoto, "Duality and Multicritical Point of Two-Dimensional Spin Glasses," *J. Phys. Soc. Jpn.*, vol. 71, p. 1198, 2002.

[83] H. G. Katzgraber, M. Körner, and A. P. Young, "Universality in three-dimensional Ising spin glasses: A Monte Carlo study," *Phys. Rev. B*, vol. 73, p. 224432, 2006.

[84] H. Munoz-Bauza, F. Hamze, and H. G. Katzgraber, "Learning to find order in disorder," *arXiv:1903.06993v1*, 2019.

[85] K. Binder, M. Nauenberg, V. Privman, and A. P. Young, "Finite-size tests of hyperscaling," *Phys. Rev. B*, vol. 31, p. 1498, 1985.

[86] O. Melchert, "autoscale.py - a program for automatic finite-size scaling analyses: A user's guide," 10 2009.

[87] K. Binder and J. D. Reger, "Theory of orientational glasses Models, concepts, simulations," *Adv. Phys.*, vol. 41, p. 547, 1992.

[88] M. Jagielski, A. Oprea, B. Biggio, C. Liu, and B. Nita-Rotaru, C andLi, "Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning ," *CoRR*, vol. abs/1804.00308, 2018.

[89] R. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," *AAAI*, 2016.

[90] Y. Shi, T. Erpek, Y. E. Sagduyu, and J. H. Li, "Spectrum data poisoning with adversarial deep learning," *CoRR*, vol. abs/1901.09247, 2019.

[91] B. Nelson, F. Barreno, F. J. Chi, A. D. Joseph, B. I. Rubinstein, U. Saini, C. Sutton, Tygar.J., and K. Xia, "Exploiting machine learning to subvert your spam filter.," *In Proc. First USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET,*, 2008.

[92] A. Newell, L. Potharaju, L. Xiang, and C. Nita-Rotaru, "On the practicality of integrity attacks on document-level sentiment analysis," *In Proc. Workshop on Artificial Intelligence and Security, AISec, 2014*, 2014.

[93] M. Abadi and et al, "Tensorflow: A system for large-scale machine learning," pp. 265–283, 2016.

[94] Y. Bengio, A. C. Courville, and P. Vincent, "Unsupervised feature learning and deep learning: A review and new perspectives," *CoRR*, vol. abs/1206.5538, 2012.

[95] D. C. Ciresan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," *CoRR*, vol. abs/1202.2745, 2012.

[96] J. Schmidhuber, "Deep learning in neural networks: An overview," *CoRR*, vol. abs/1404.7828, 2014.

[97] G. F. Newell, "Crystal Statistics of a Two-Dimensional Triangular Ising Lattice," *Physical Review*, vol. 79, pp. 876–882, Sept. 1950.

[98] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," *arXiv:1603.04467 [cs]*, Mar. 2016.

[99] A. Morningstar and R. G. Melko, "Deep Learning the Ising Model Near Criticality," Aug. 2017.

[100] A. Tanaka and A. Tomiya, "Detection of phase transition via convolutional neural network," *Journal of the Physical Society of Japan*, vol. 86, p. 063001, June 2017.

[101] H. G. Katzgraber, "Monte Carlo simulations of vector spin glasses at low temperatures," *Computer Physics Communications*, vol. 147, p. 439, 2002.

[102] C. W. Liu, A. Polkovnikov, A. W. Sandvik, and A. P. Young, "Universal dynamic scaling in three-dimensional ising spin glasses," *Physical Review E*, vol. 92, 11 2014.

[103] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Neurocomputing: Foundations of research," ch. Learning Representations by Back-propagating Errors, pp. 696–699, Cambridge, MA, USA: MIT Press, 1988.

[104] R. Hecht-Nielsen, "Neural networks for perception (vol. 2)," ch. Theory of the Backpropagation Neural Network, pp. 65–93, Orlando, FL, USA: Harcourt Brace & Co., 1992.

[105] R. Novak, Y. Bahri, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein, "Sensitivity and generalization in neural networks: an empirical study," in *International Conference on Learning Representations*, 2018.

[106] T. Nakamura, S.-i. Endoh, and T. Yamamoto, "Weak universality of spin-glass transitions in three-dimensional $\pm J$ models," *J. Phys. A*, vol. 36, p. 10895, 2003.

[107] K. Hukushima *et al.*, "Exchange Monte Carlo dynamics in the sk model," *J. Phys. Soc. Jpn.*, vol. 67, p. 12, 1998.

[108] U. Wolff, "Collective Monte Carlo updating for spin systems," *Phys. Rev. Lett.*, vol. 62, p. 361, 1989.

[109] Z. Zhu, A. J. Ochoa, and H. G. Katzgraber, "Efficient Cluster Algorithm for Spin Glasses in Any Space Dimension," 2015. (cond-mat/1501.05630).

[110] L. Landau and E. Lifshitz, *Statistical Physics*. No. v. 5, Elsevier Science, 2013.

[111] H. G. Katzgraber *et al.*, "Monte Carlo simulations of spin glasses at low temperatures," *Phys. Rev. B*, vol. 63, p. 184422, 2001.

[112] B. Yucesoy, J. Machta, and H. G. Katzgraber, "Correlations between the dynamics of parallel tempering and the free-energy landscape in spin glasses," *Phys. Rev. E*, vol. 87, p. 012104, 2013.

[113] H. E. Stanley, *Scaling, Universality, and Renormalization: Three Pillars of Modern Critical Phenomena*, pp. 601–616. New York, NY: Springer New York, 1999.

[114] K. G. Wilson and J. B. Kogut, "The Renormalization group and the epsilon expansion," *Phys. Rept.*, vol. 12, pp. 75–200, 1974.

[115] K. G. Wilson, "The renormalization group: Critical phenomena and the kondo problem," *Rev. Mod. Phys.*, vol. 47, pp. 773–840, Oct 1975.

[116] L. Y. Chen, N. Goldenfeld, and Y. Oono, "Renormalization group theory for global asymptotic analysis," *Phys. Rev. Lett.*, vol. 73, pp. 1311–1315, Sep 1994.

[117] J. Cardy, *Scaling and Renormalization in Statistical Physics*. Cambridge: Cambridge University Press, 1996.

[118] C. Itzykson, H. Saleur, and J. Zuber, *Conformal Invariance and Applications to Statistical Mechanics*. World Scientific, 1998.

[119] P. Di Francesco, P. Mathieu, and D. Senechal, *Conformal Field Theory*. Graduate Texts in Contemporary Physics, New York: Springer-Verlag, 1997.

[120] T. T. Wu, B. M. McCoy, C. A. Tracy, and E. Barouch, "Spin-spin correlation functions for the two-dimensional ising model: Exact theory in the scaling region," *Phys. Rev. B*, vol. 13, pp. 316–374, Jan 1976.

[121] J. L. Cardy, "Central charge and universal combinations of amplitudes in two-dimensional theories away from criticality," *Phys. Rev. Lett.*, vol. 60, pp. 2709–2711, Jun 1988.

[122] J. L. Cardy and H. Saleur, "Universal distance ratios for two-dimensional polymers," *Journal of Physics A: Mathematical and General*, vol. 22, pp. L601–L604, jul 1989.

[123] C. Domb and J. L. Lebowitz, eds., *PHASE TRANSITIONS AND CRITICAL PHENOMENA. VOL. 11*. 1987.

[124] J. Salas and A. D. Sokal, "Universal amplitude ratios in the critical two-dimensional ising model on a torus," *Journal of Statistical Physics*, vol. 98, pp. 551–588, Feb 2000.

[125] J. Zinn-Justin, "Quantum field theory and critical phenomena," *Int. Ser. Monogr. Phys.*, vol. 113, pp. 1–1054, 2002.

[126] A. B. Harris, "Effect of random defects on the critical behaviour of Ising models," *J. Phys. C*, vol. 7, p. 1671, 1974.

[127] Janke, W. and R. Villanoca, "Three-dimensional 3-state Potts model revisited with new techniques," *Phys. Rev. B*, vol. 66, p. 134208, 2002.

[128] B. H and V. T, "Phase Transitions on Random Lattices: How Random is Topological Disorder?," *Phys.Rev.Lett.*, vol. 113, 2014.

[129] P. Martins and J. Plascak, "Universality class of the two-dimensional site-diluted Ising model," *Phys. Rev. E*, vol. 76, 2007.

[130] M. Hasenbusch, A. Pelissetto, and E. Vicari, "The critical behavior of three-dimensional Ising glass models," *Phys. Rev. B*, vol. 78, p. 214205, 2008.

[131] H. G. Ballesteros, L. A. Fernández, V. Martín-Mayor, A. Muñoz Sudupe, G. Parisi, and J. J. Ruiz-Lorenzo, "Critical exponents of the three-dimensional diluted ising model," *Phys. Rev. B*, vol. 58, pp. 2740–2747, Aug 1998.

[132] H. G. Katzgraber, S. Trebst, D. A. Huse, and M. Troyer, "Feedback-optimized parallel tempering Monte Carlo," *J. Stat. Mech.*, vol. P03018, 2006.

[133] S. Wiseman and E. Domany, "Finite-Size Scaling and Lack of Self-Averaging in Critical Disordered Systems," *Phys. Rev. Lett.*, vol. 81, p. 22, 1998.

[134] J. L. Pichard and G. Sarma, "Finite-size scaling approach to anderson localisation. II. quantitative analysis and new results," *Journal of Physics C: Solid State Physics*, vol. 14, pp. L617–L625, jul 1981.

[135] J. Salas and A. D. Sokal, "Universal Amplitude Ratios in the Critical Two-Dimensional Ising Model on a Torus." (cond-mat/9904038v1), 1999.

[136] A. D. Sokal, *Monte Carlo Methods in Statistical Mechanics: Foundations and New Algorithms (Lecture notes, Cours de Troisieme Cycle de la Physique en Suisse Romande)*. 1989.

[137] R. Baxter, *Exactly Solved Models in Statistical Mechanics*. London: Academic Press, 1982.

[138] A. Barzegar, J. C. Andresen, M. Schechter, and H. G. Katzgraber, "Numerical observation of a glassy phase in the three-dimensional coulomb glass," *Phys. Rev. B*, vol. 100, p. 104418, Sep 2019.

[139] H. W. J. Blöte and Y. Deng, "Cluster monte carlo simulation of the transverse ising model," *Phys. Rev. E*, vol. 66, p. 066110, Dec 2002.

[140] S. Tarcoma, C. E. Rothenberg, and E. Lagerspetz, "Theory and practice of Bloom filters for distributed systerms," *Communications Surveys & Tutorials*, vol. 14(1), p. 131, 2012.

[141] v. Radvilavicius, L. Marozas and Cenys.A, "Overview of real-time antivirus scanning engines," *Journal of Engineering Science and Technology Review.*, vol. 5(1), pages = 63âĂŞ71, 2012.

[142] B. Selman and H. Kautz, "Planning as satisfiability," *European Conference on Artificial Intelligence*, pp. 359–363, 1992.

[143] O. Gableske. https://www.gableske.net/dimetheus.

[144] J. Franco and M. Paull, "Probabilistic analysis of the Davis Putnam procedure for solving the satisfiability problem. ," *Discrete Applied Mathematics*, vol. 5(1), p. 77âĂŞ87, 1983.

[145] D. Achlioptas, "Random satisfiability," *Handbook of Satisfiability, 185 of Frontiers in Artificial Intelligence and Applications. IOS Press*, p. 245, 2009.

[146] Y. Achlioptas, D. Peres, "The threshold for random k-SAT is 2klog2âĹŠO(k).," *Journal of the American Mathematical Society*, vol. 17(4), pp. 947–973, 2004.

[147] A. Coja-Oghlan and K. Panagiotou, "Going after the k-SAT threshold," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 705–714, ACM, 2013.

[148] B. Selman, H. Kautz, and B. Cohen *et al.*, "Local search strategies for satisfiability testing," *Cliques, coloring, and satisfiability: Second DIMACS implementation challenge*, vol. 26, p. 521, 1993.

[149] S. Cai, K. Su, C. Luo, and A. Sattar, "NuMVC: An efficient local search algorithm for minimum vertex cover," *Journal of Artificial Intelligence Research*, p. 687, 2013.

[150] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, p. 671, 1983.

[151] D. S. Johnson and L. A. McGeoch, "The traveling salesman problem: A case study in local optimization," *Local search in combinatorial optimization*, vol. 1, p. 215, 1997.

[152] S. Boettcher and A. G. Percus, "Optimization with Extremal Dynamics," *Phys. Rev. Lett.*, vol. 86, p. 5211, 2001.

[153] K. F. Pal, "Genetic algorithm with local optimization," *Biol. Cybern.*, vol. 73, p. 335, 1995.

[154] Z. Zhu, C. Fang, and H. G. Katzgraber *Optimization Letters*, 2020.

[155] D. J. Earl and M. W. Deem, "Parallel Tempering: Theory, Applications, and New Perspectives," *Phys. Chem. Chem. Phys.*, vol. 7, p. 3910, 2005.

[156] H. Rieger, *Frustrated systems: Ground state properties via combinatorial optimization*, vol. 501, p. 122. 1998.

[157] I. Grigorenko and M. Garcia, "An evolutionary algorithm to calculate the ground state of a quantum system," *Physica A: Statistical Mechanics and its Applications*, vol. 284, no. 1, pp. 131 – 139, 2000.

[158] http://www.maxsat.udl.cat/15/.

[159] https://www.cs.rochester.edu/u/kautz/walksat/.

[160] We use the executables from the 2015 SAT competition available for download the the SAT competition website.

[161] http://www.maxsat.udl.cat/15/solvers/.

[162] By "TSS" we mean the needed to find an optimal solution, i.e., the time cost to find a solution.

[163] A. Broder and M. Mitzenmacher, "Network application of Bloom filters: A survey," *Internet Mathematics*, vol. 1(4), p. 485, 2004.

[164] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Comm. ACM*, vol. 13(7), p. 422, 1970.

[165] A. Biere, M. Heule, H. Maaren, and T. Walsh, *Handbook of Satisfiability,* ***185*** *of Frontiers in Artificial Intelligence and Application*. Amsterdam, The Netherlands: IOS press, 2009.

[166] A. Hartmann, *Phase Transitions in Combinatorial Optimization Problems - Basics, Algorithms and Statistical Mechanics*. Wiley-VCH, 2005.

[167] M. Mezard, G. Parisi, and M. Virasoro, *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*. World Scientific Lecture Notes in Physics, World Scientific Publishing Company, 1987.

[168] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Upper Saddle River, NJ, USA: Prentice Hall Press, 3rd ed., 2009.

[169] T. Ellman, "Explanation-based learning: A survey of programs and perspectives," *ACM Comput. Surv.*, vol. 21, pp. 163–221, June 1989.

[170] C. Fang, Z. Zhu, and H. G. Katzgraber, “Nae-sat-based probabilistic membership filters,” *CoRR*, vol. abs/1801.06232, 2018.