

Master's Thesis

修士論文

Renormalization group prescription in
tensor-network language

(テンソルネットワーク表現に
よる繰り込み群の方法)

A Thesis Submitted for Degree of Master of Science
August 2021

令和3年8月修士（理学）申請

Department of Physics, Graduate School of Science,
The University of Tokyo

東京大学大学院理学系研究科物理学専攻

Xinliang Lyu

呂 欣亮

Statement of Contributions

This thesis is adapted from the material published in Ref. [40], co-authored with Xu and Kawashima. Kawashima initiated this research project and proposed the idea of realizing the textbook RG prescription in tensor-network language. I decided what specific techniques to employ and finished most of the numerical analysis. Xu helped with the implementation of the GILT and the usage of automatic differentiation.

Renormalization group prescription in tensor-network language

by

Xinliang Lyu

Submitted to the Department of Physics
on August 12, 2021, in partial fulfillment of the
requirements for the degree of
Master of Science

Abstract

This thesis introduces how to perform the renormalization group (RG) prescription in the modern tensor-network language to extract scaling dimensions of a critical system—write down the tensor RG equation, identify a fixed-point tensor, linearize the RG equation around this fixed point, and calculate scaling dimensions from the eigenvalues of the resulting linearized RG equation.

A traditional RG transformation is based on the Hamiltonian (or energy) description of a statistical system; the RG equation is a map from the old Hamiltonian to the coarse-grained Hamiltonian. The real-space RG transformations—the decimation for example—are intuitive and have a clear physical picture. Nevertheless, in the textbook example of a real-space RG transformation, Migdal-Kadanoff bond moving scheme, the approximations are uncontrollable, so the estimation of the scaling dimensions cannot be improved systematically.

Recently, ideas from quantum information have stimulated novel types of real-space RG transformations based on a tensor-network representation of a statistical system. The coarse graining of this modern tensor-network-based RG resembles the conventional block-spin method, inheriting the clear physical picture, and is even more straightforward to define than the Hamiltonian-based description. The advantage of the tensor-network-based RG over the Migdal-Kadanoff idea is that the approximations of the former can be controlled by an integer called the bond dimension.

If the traditional RG prescription for extracting scaling dimensions work in the tensor-network-based RG method, we then have a kind of real-space RG that is both intuitive and quantitatively accurate. The main result of the thesis is to show that the above “if” statement is true, by proposing a general framework of the tensor-network-based RG prescription and a concrete numerical realization of the general framework, benchmarked in the context of the two-dimensional Ising model.

Thesis Supervisor: Naoki Kawashima

Acknowledgments

This thesis has only been possible due to the help of my supervisor, Naoki Kawashima, who suggested this promising idea of realizing the textbook renormalization group prescription in tensor-network language when I started my Master's course two years ago. Working with him is a pleasant and rewarding journey that teaches me how to keep calm outside the comfort zone full of mysteries and then try one's luck to enlarge the territory of our knowledge.

I am also very grateful for the support of two of our group members: Satoshi Morita and Shumpei Iino, who provided lots of useful suggestions and technical support about the implementation of various tensor-network-based RG techniques. Besides, I want to thank RuQing G. Xu for the discussions about automatic differentiation, my second supervisor Takeo Kato for regular feedback about my research progress, and all the students, postdocs, and faculty members of the Kawashima group. The suggestions from one of my Thesis Defense Committee Members, Hirokazu Tsunetsugu, are very enlightening and stimulating. I thank Georgetown Professor Jim Freericks, who gave me my first research opportunity in physics, making possible the start of my Master's course here at the University of Tokyo.

I thank all members of the University of Tokyo who I have met and talked with, especially the peer students with whom I have private seminars regularly, including Weiguang Cao, Shih-Wen Hor, and Yuan Yao. I thank all my friends who have enriched my daily life here in Tokyo, including my old friend from undergraduate time, Xiaofei Yan, and my new friends from the dormitory (Kashiwa-no-ha International Village), Zehao Li, Tianyi Zhu, and ChuTong (Tetra) Shen. I thank all my old friends and family members back in China.

Finally, I want to thank the financial support of the Global Science Graduate Course (GSGC) program of the University of Tokyo. The numerical computations were performed on computers at the Supercomputer Center, the Institute for Solid State Physics (ISSP), the University of Tokyo.

Contents

1	Introduction	17
1.1	Conventional RG in Hamiltonian-based language	17
1.2	Monte Carlo RG	19
1.3	Tensor-network-based RG	20
1.4	Does the canonical RG prescription work in tensor-network language?	22
1.5	Outline of this thesis	24
2	Real-space Renormalization Group	25
2.1	A simple example: decimation of the one-dimensional Ising model . .	26
2.1.1	Coarse graining by decimation	26
2.1.2	Fixed points and linearized RG equations	29
2.2	General framework	32
2.3	Decimation of the two-dimensional Ising model	34
2.3.1	Decimation in $d = 2$ generates new interactions	34
2.3.2	The Migdal-Kadanoff bond moving approximation	36
2.3.3	The linearized RG equation and divergence of the correlation length	38
3	Tensor-network Renormalization Group	41
3.1	From a lattice with statistical variables to a tensor-network model . .	41
3.2	Coarse graining of a tensor-network model	44
3.2.1	A naïve approach	44

3.2.2	An approximation scheme: the higher-order tensor renormalization group (HOTRG)	46
3.3	Systematic improvement by including more interactions	50
4	Renormalization Group Prescription in Tensor-network Language	53
4.1	Heuristic argument	54
4.2	Connection with the Hamiltonian-based approach	55
4.2.1	Decimation of the 1D Ising model in tensor-network language	56
4.2.2	Formal argument	59
4.3	Two caveats	62
4.3.1	Local correlations	63
4.3.2	Gauge redundancy	65
5	A Numerical Realization	67
5.1	Filtering out local correlations: a HOTRG-like scheme	67
5.2	Gauge fixing	72
5.2.1	A general proposal	72
5.2.2	A special property of the proposed HOTRG-like scheme	73
5.2.3	Sign ambiguities	74
5.3	Overall procedure for the numerical realization	74
5.4	Benchmark results	75
5.4.1	Analysis of the RG flow	75
5.4.2	Scaling dimensions	78
5.4.3	Remarks	80
6	Summary and Discussions	83
A	Optimization forms of eigenvalue problems	85
B	Proof regarding gauge fixing	87

List of Figures

1-1	How an RG transformation is defined. μ is the probability measure for a system with Hamiltonian H , usually with the form $\mu \propto e^{-H}$. A real-space RG transformation is first defined as a map $\mu \rightarrow \mu'$. In Hamiltonian-based language, upon imposing that the coarser probability distribution has the form $\mu' \propto e^{-H'}$, we determine the coarser Hamiltonian H' , thus inducing the Hamiltonian-based RG equation $\mathcal{T}^{\text{old}} : H \rightarrow H'$. The last step might be ill-defined. In tensor-network-based RG, the Hamiltonian description of the system is not necessary, and we can work with the map $\mathcal{T} : \mu \rightarrow \mu'$ between probability distributions directly.	19
2-1	The decimation of the 1D Ising model, where the linear dimension of the system is larger by a factor of $b = 2$. The black dots are spin variables. We trace out all the spins on even sites $\sigma_2, \sigma_4, \dots$ in the partition function; the remaining spins $\sigma_1, \sigma_3, \dots$ are renamed $\sigma'_1, \sigma'_2, \dots$ to become new spin variables at larger scale of length.	27
2-2	The decimation of the 2D Ising model. The black dots are spin variables. The spins that are marked with a cross sign like σ_0 are decimated, which will mediate new interactions including second-neighbor terms like $K'_2\sigma_1\sigma_3$ and a four-spin term $K'_4\sigma_1\sigma_2\sigma_3\sigma_4$. The lattice constant is $b = \sqrt{2}$ times as the original one.	35

2-3	The bond moving trick for the Migdal-Kadanoff RG with $d = 2$ and $b = 2$. (a) Move the bonds not connected to the spins that will be retained after decimation. (b) Perform the decimation. Summing over spins like σ_2, σ_4 will only contribute to the overall multiplicative constant before the partition function, while summing over spins like σ_0 will mediate interactions between σ_1, σ_3 in a same way as the $d = 1$ case shown in Fig. 2-1.	37
2-4	The roots of this function are the fixed points of Eq. (2.27).	38
3-1	Map spin variables to tensors. The blue dots are where the spin variables locate, forming a square lattice slanted by 45° . The larger green circles are tensors encoding the probability distribution of the configurations of the four surrounding spin variables sitting on four legs of the tensor; the tensors form another square lattice.	42
3-2	The physical picture of the block tensor transformation $A \rightarrow A_c$ in Eq. (3.5). Part of the spins shared by two tensors A are summed over according to Eq. (3.5). The black squares are larger after the decimation.	45
3-3	Relative error of free energy per spin g_{est} estimated at bond dimensions $\chi = 4, 8, 12, 16, 20$ using the HOTRG. T_c, g are exact values of the critical temperature and free energy per spin of the 2D Ising model according to Onsager's solution.	51
4-1	In the tensor network language, the decimation in Fig. 2-1 is nothing but a matrix multiplication of two transfer matrices to form a coarse-grained matrix $A_c = AA$	56
4-2	Relation between the two linearized RG equations in Hamiltonian and tensor-network languages. It is reminiscent of Fig. 1-1. Provided that a Hamiltonian description exists for a tensor-network coarse-graining scheme, such that $A_{(i)} = f_{(i)}(\mathbf{K}), (A_c)_{(i)} = f_{(i)}(\mathbf{K}')$, the linearized RG equation in tensor-network language \mathcal{R} and that in Hamiltonian-based language \mathcal{R}^{old} are the same linear transformation in two representations.	62

4-3 Physical original of the CDL structure. (a) The physical picture of a block-tensor transformation. The old tensor A is surrounded by four spin variables, while the coarser tensor A_c is surrounded by eight spin variables. (b) At a later stage of the block-tensor transformation, many original spin variables sit on the sides of the square, with all spins in the middle are traced out during the coarse graining. If the system is in its high-temperature phase, the spin variables on one edge must be uncorrelated with those on another edge, except for the spin variables around the four corners, which remain to be very close to each other. The correlations between spin variables around the corner can be described by a matrix C , the direct product of four copies of which gives rise to a CDL tensor A^{CDL} 64

4-4 Schematic RG flows of the 2D Ising model, one is what we expect and the other is what the TRG (or any TRG-like scheme) generates. The starting points of the RG flows form the blue dashed line, representing the lattice model at different temperatures. RG flows are drawn using solid lines with arrows. (a) The tensor RG flows we expect for the 2D Ising model. There are three fixed points in the tensor space, one high-temperature fixed point indicated by $T = \infty$, one low-temperature fixed point indicated by $T = 0$, and one critical fixed point $T = T_c$. (b) The tensor RG flows the TRG (or any TRG-like scheme) generate. Since the local correlations at the original lattice scale are carried all the way to larger scales, different starting points, even belonging to the same phase, will flow to different fixed points with the CDL structure. The critical fixed point might disappear at bond dimensions $\chi > 8$, as is indicated in the numerical evidence by Hinczewski and Berker [27]. 65

5-1	Truncation a single bond of the plaquette consisting of tensor A using the GILT. Four copies of C matrices are unknown inner structure of 4-leg tensor forming the plaquette. They are drawn explicitly to make the demonstration clearer. In the first step, a low-rank matrix Q is inserted into a bond. The splitting of the Q into two pieces using singular value decomposition cuts the leg of corner matrix C . Finally, two pieces of the matrix Q are absorbed into the two neighboring 4-leg tensors. The original GILT paper [26] presents a way to determine the low-rank matrix Q . A brief introduction is provided in Appendix B.	69
5-2	How to choose the plaquette and where to insert the low-rank matrices to filter out the problematic local correlations for the HOTRG.	70
5-3	The RG flows of the tensor norms $\ A^{(n)}\ $ at temperatures near the estimated critical temperature $T_c^{[\chi]}$. Different markers represent different deviations $ \Delta T $ from $T_c^{[\chi]}$. Blue solid lines are for $\Delta T < 0$ and black dashed lines for $\Delta T > 0$. (a) For the proposed HOTRG-like scheme with $\chi = 30, \epsilon_{\text{gilt}} = 6 \times 10^{-6}$, two trivial fixed points are isolated and the critical fixed point can be reached. It corresponds to the schematic RG flows in Fig. 4-4(a). (b) For the plain HOTRG with $\chi = 12$, we have fixed lines and there is no exhibition of a critical fixed point. It corresponds to the schematic RG flows in Fig. 4-4(b).	77
5-4	The RG flows of (a) singular values defined in Eq. (5.16) and (b) the difference between the normalized tensors, $\ \mathcal{A}^{(n+1)} - \mathcal{A}^{(n)}\ $ with sign fixing and (c) without, all at the estimated critical temperature $T_c^{[30]}$, generated by the proposed HOTRG-like scheme with $\chi = 30, \epsilon_{\text{gilt}} = 6 \times 10^{-6}$	79
5-5	The scaling dimensions of the 2D Ising model from the canonical RG prescription using the proposed HOTRG-like scheme with $\chi = 30, \epsilon_{\text{gilt}} = 6 \times 10^{-6}$. Dashed lines are the exact values.	80

List of Tables

5.1	The scaling dimensions for the relevant and marginal operators of the 2D Ising model at criticality from the canonical RG prescription and from the transfer matrix method à la Gu and Wen [24], both using the proposed HOTRG-like scheme with $\chi = 30$, $\epsilon_{\text{gilt}} = 6 \times 10^{-6}$ at RG step $n = 22$	81
-----	---	----

Chapter 1

Introduction

The renormalization group (RG) is a powerful technique for studying physical systems where fluctuations in all scales of length are important [61]. In statistical mechanics, the most famous example is critical phenomena. Take water as an example, imagine the temperature and pressure are increased along the liquid-vapor coexistence line, when we reach a temperature of 374°C and pressure of 218 atm, the liquid-vapor phase-separation boundary starts vanishing and the density of water fluctuates at all scales of length. The density fluctuations with the order of hundreds of nanometers will scatter visible light strongly. The water looks milky due to such fluctuations. The RG can be applied to critical phenomena like this, being able to make predictions that agree with experiment results.

1.1 Conventional RG in Hamiltonian-based language

Conventional RG schemes, such as ϵ -expansion [62] and block-spin methods [28, 29, 44, 30, 47], are based on Hamiltonian description of a physical system. We look for a map from the Hamiltonian describing the system at a short scale of length to an effective Hamiltonian describing the system at a longer one, such that the partition function is unchanged [4]. The map is known as an RG equation. A fixed point of the RG equation corresponds to a conformal field theory (CFT) [49, 46]. A critical system is described by a fixed point. Universal properties like scaling dimensions of the critical

system can be extracted from the behavior of the RG equation near the corresponding fixed point. The critical exponents, closely related to the scaling dimensions, are physical observables that can be compared with experiment results. In this thesis, the procedure of extracting critical exponents from the linearized behavior of the RG equation around a critical fixed point is referred to as *canonical RG prescription*. Besides giving us physical observables, this canonical RG prescription also helps us understand universality in critical phenomena.

In principle, the exact RG equation is a transformation from an infinite-dimensional space to another infinite-dimensional space [61, 58]. For a practical implementation of the RG idea, approximations are necessary. Some approximation schemes, like Migdal-Kadanoff bond moving approximation [44, 43, 30] (see Section 2.3.2 for details), are easy to implement but their results cannot be improved systematically. To find a controllable approximation, people have developed perturbative expansions based on small parameters related to, for example, the spatial dimension (ϵ expansion [62]), internal freedom of the statistical variable ($1/n$ expansion [6]), and ranges of interactions (Niemeijer-van Leeuwen cumulant approximation [47]). However, these expansions are nontrivial to calculate and often yield asymptotic series, which are non-convergent and should be treated by summation techniques in general.

There is an even more annoying and subtle problem about the Hamiltonian-based real-space RG transformations, though might be less familiar [23, 55]. A real-space RG transformation needs to be first defined as a map from a probability distribution of a local configuration of old spins to that of coarser spins, after which we then determine effective Hamiltonian corresponding to the probability distribution of coarser spins (see the decimation of the 1D Ising model in Section 2.1 for a concrete example). However, in the thermodynamic limit, we may not be able to find the effective Hamiltonian given the probability distribution of coarser spins; this means that the real-space RG transformation sometimes even *does not exist* (see Fig. 1-1) in the Hamiltonian-based language! This potential pathologies limit the practical implementation of the real-space RG idea in the conventional Hamiltonian space.

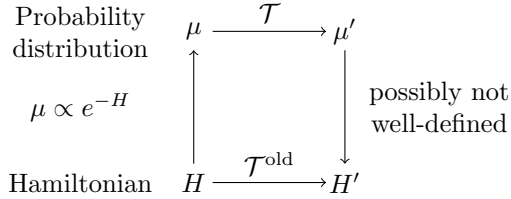


Figure 1-1: How an RG transformation is defined. μ is the probability measure for a system with Hamiltonian H , usually with the form $\mu \propto e^{-H}$. A real-space RG transformation is first defined as a map $\mu \rightarrow \mu'$. In Hamiltonian-based language, upon imposing that the coarser probability distribution has the form $\mu' \propto e^{-H'}$, we determine the coarser Hamiltonian H' , thus inducing the Hamiltonian-based RG equation $\mathcal{T}^{\text{old}} : H \rightarrow H'$. The last step might be ill-defined. In tensor-network-based RG, the Hamiltonian description of the system is not necessary, and we can work with the map $\mathcal{T} : \mu \rightarrow \mu'$ between probability distributions directly.

1.2 Monte Carlo RG

Besides implementing the RG transformation deterministically, it is also possible to apply the Monte Carlo (MC) sampling to the above Hamiltonian-based RG idea [41, 53]. After reaching equilibrium, an MC simulation samples snapshots of configurations following the usual thermal distribution of a given Hamiltonian. From the sequence of snapshots, correlation functions can be calculated. A real-space RG transformation now can be applied to each sample configuration to generate a coarser configuration. For example, a block-spin transformation according to the majority rule can be applied to each 3×3 block of original spins for the 2D Ising model. The coarser snapshots of configurations correspond to what we would get if an MC simulation had been applied to sample the coarser Hamiltonian obtained using a deterministic RG transformation. If we apply the MC simulation to the original Hamiltonian at criticality, we can use the two sequences of the original and coarser samples to calculate the linearized RG equation near the fixed-point Hamiltonian, from which the scaling dimensions can be extracted. Recent developments in the machine learning community and information theory [34, 36] provide a novel way to learn an optimal real-space RG transformation for generating the coarser configurations to keep the interactions in the renormalization Hamiltonian as local as possible. For example, the neural Monte Carlo RG (MCRG) [9] provides better estimations of the scaling dimensions of the 3D

Ising model than the majority rule does.

1.3 Tensor-network-based RG

Novel types of RG schemes based on tensor-network language have been booming recently, initially inspired by the development of quantum information. They are versatile numerical RG methods; different physical models correspond to different initial tensors, while the tensor-network RG equation shares a universal form. The idea is to *skip the Hamiltonian description altogether, focusing instead on the probability distribution of local configurations directly*; the RG equation is a map from a tensor encapsulating the probability distribution of local configurations at a short length scale to a coarser tensor at a longer one (see Fig. 1-1). It is Levin and Nave who proposed the first numerical implementation of this tensor-network RG idea, tensor renormalization group (TRG), in 2007. The TRG is designed for two-dimensional (2D) classical systems (or $(1 + 1)$ -dimensional quantum systems), and has an excellent performance in calculations of free energy, with its approximation controlled by an integer χ , called the *bond dimension*. The larger the value of χ , the more time the computation takes, the better the approximation becomes, and the more accurate the estimation of free energy is. Later, people realized that even better approximation can be achieved by considering a larger environment [64, 67, 63, 45]. With the help of these tensor-network RG methods like the TRG, we can easily calculate the free energy of, for example, the 2D Ising model, with an error of order 10^{-7} within one minute in a desktop computer (see Section 3.3 for details). The accuracy improves exponentially as χ increases, while the computation cost only grows polynomially. Generalizations to three-dimensional (3D) classical systems (or $(2 + 1)$ -dimensional quantum systems) are also successful [63, 1, 31], the earliest of which is higher-order tensor renormalization group (HOTRG) [63] proposed in 2012.

Advantages of tensor-network-based RG and a missing piece Why do we care about developing these novel RG schemes in tensor-network language, instead of

sticking with the well-developed Hamiltonian-based RG? Tensor-network-based RG is a different perspective to understand and implement the RG idea. It has the following *advantages compared with the Hamiltonian-based RG*. First of all, the growing range of interactions in Hamiltonian language (see Section 2.3 for a short review) is replaced by the growing of the number of states (or the bond dimension) of the tensor legs, while the interaction range in the tensor-network picture remains short-ranged, mediated by nearest-neighbor tensors (see Eqs. (3.5), (3.6) and Fig. 3-2). In addition, the map from an old tensor A to a coarser tensor A_c is explicit, while the RG equation in Hamiltonian language is complicated by the nonlinear exponential and logarithmic transformations (see Fig. 1-1). Due to these two advantages, it is now possible to establish rigorous theorems about the behavior of an *exact* tensor-network-based RG equation near the high-temperature fixed-point tensor located in an infinite-dimensional Hilbert space [33].

As a numerical method, the above-mentioned two advantages make it easier to improve the estimations of the tensor-network-based RG systematically. It also has advantages in quantum systems and classical frustrated systems, where the complex Boltzmann weight and long relaxation time make MC simulations in MCRG quite nontrivial.

Finally, the coarse-graining step is easier to grasp in this modern tensor-network language (see Chapter 3). Pedagogically, it can supplement the conventional example of the decimation for one-dimensional (1D) Ising model (see Section 2.1 and Section 4.2.1) and serve as a generalization of the Migdal-Kadanoff bond moving approximations in higher dimensions (see Section 2.3.2 and Section 3.2.2). A tensor-network RG equation and its linearized version can be much simpler than its Hamiltonian counterpart.

However, these tensor-network RG techniques still miss an important aspect as real-space RG transformations: the canonical RG prescription for extracting universal properties of a critical system.

1.4 Does the canonical RG prescription work in tensor-network language?

The unprecedented success of the tensor-network-based RG is mainly demonstrated by its accurate estimation of free energy. However, the free energy of a system is often not a direct physical observable. In order to get physical observables that can be compared to experiment, we need to take derivatives of free energy with respect to various parameters like temperature and external magnetic field. This is a possible way to proceed in these tensor-network-based RG methods [63, 27, 39], but may not be the best one.

The advantage of the RG idea is its canonical prescription for extracting physical observables by studying the behavior of the RG equation near a critical fixed point: it avoids dealing with the divergence directly but can tell us how physical observables diverge in a clever way. Therefore, we want to ask whether the canonical RG prescription works in these tensor-network RG schemes.

Early attempts Several attempts were made to carry out the canonical RG prescription in tensor-network language. In 2008, just one year after Levin and Nave proposed the TRG, Hinczewski and Berker [27] first analyzed the tensor RG flows generated by the TRG. What they discovered is interesting. By fine-tuning the temperature of the 2D Ising model on a triangular lattice, they found that for bond dimensions $\chi \leq 8$, the tensor would flow to the critical fixed point before flowing away to the high- and low-temperature trivial fixed points. When they increased the bond dimensions to $\chi = 12$ and further, the tensor RG equation *ceases to* exhibit a critical fixed point. This means that the canonical RG prescription is limited to $\chi \leq 8$ in any tensor RG scheme that generates RG flows similar to the TRG. People have tried the RG prescription at $\chi = 2, 3, 4$ in the context of the 2D Ising model, and the estimated scaling dimension of the energy density operator has an accuracy similar to the old potential moving tricks [27, 2, 42], and that of the spin operator is more than a factor of 2 larger than the exact value [15]

Peculiar RG flows generated by early tensor-network RG schemes and the possible solutions The problem of the RG flow noticed by Hinczewski and Berker might be unfamiliar since it seldom occurs in the examples of the RG discussed in textbooks. Actually, Wilson, one of the architects of the RG idea, did notice that there is no guarantee for the RG transformations to exhibit fixed points, and called it “a serious problem with the renormalization-group transformations” in his 1982 Nobel prize lecture [61].

In the last ten years, many tricks have been proposed to rectify the tensor RG flows generated by the TRG. Following Levin’s suggestion [38, 37], Gu and Wen identified the cause of the peculiar tensor RG flows to be a type of tensors with a corner double-line (CDL) structure, representing local correlations. Since CDL tensors are fixed points of the TRG transformation, the local correlations at the original lattice scale will be carried to the larger scales. By filtering out the tensors with CDL structure, several advanced tensor-network RG techniques [24, 18, 16, 65, 3] are able to produce the critical fixed-point tensor of the 2D Ising model at a general bond dimension.

Since a system at the critical fixed point is described by a CFT, the tensor network formed by the critical fixed-point tensor of the 2D Ising model represents the partition function of the 2D Ising CFT. According to a well-known 2D CFT theorem [8], Gu and Wen [24] pointed out that the scaling dimensions of the CFT can be extracted from the eigenvalues of the system’s transfer matrix constructed from the fixed-point tensor. This method is simple to apply to 2D critical systems and thus becomes the bread-and-butter tool to extracting scaling dimensions in the tensor-network RG techniques. Later, Evenbly and Vidal used the tensor network renormalization (TNR) [18, 16] to implement local scale transformation that maps a plane to a cylinder [19]; the spectrum of eigenvalues of a transfer matrix on the cylinder gives scaling dimensions. These methods provide two main routines to extracting scaling dimensions in tensor-network RG schemes, while the canonical RG prescription has never been followed up.

1.5 Outline of this thesis

In this thesis, we provide the missing piece of carrying out, in the general prescription, the RG in tensor-network language at a general bond dimension. In Chapter 2, we use the decimation of the 1D Ising model, an exact RG transformation, to demonstrate the general RG idea and the RG prescription in the conventional Hamiltonian-based setting, followed by a short review of the general framework of the Hamiltonian-based RG. We also supplement the 1D Ising example with the decimation of the 2D Ising model, showing the necessity of approximations in a general RG transformation; the Migdal-Kadanoff bond moving approximation is introduced. The idea of modern tensor-network RG idea is demonstrated in the context of the 2D Ising model in Chapter 3, using the higher-order tensor renormalization group (HOTRG) as the approximation scheme, which is closely related to the Migdal-Kadanoff bond moving approximation. After the above review of previous study, the *key results of the thesis* are presented in Chapter 4 and Chapter 5. Specifically, in Chapter 4, we define the *canonical RG prescription in tensor-network language* first heuristically using the RG equation of the HOTRG, and then formally in a general way. Two caveats are noted for any practical implementation of the canonical RG prescription in tensor-network language. In Chapter 5, we provide *a practical numerical realization of the prescription* and the proposed method is benchmarked in the context of the 2D Ising model. We summarize in Chapter 6.

The success of the RG prescription in the tensor-network language demonstrated in this thesis offers a better understanding of the nature of these tensor-network RG techniques as real-space RG transformations. The method might be relevant in three dimensions, where Gu and Wen’s method is inapplicable and Evenbly and Vidal’s local-scaling-transformation idea is nontrivial to implement. Pedagogically, it provides a better way to introduce the *full* RG idea, because both the RG equation and the linearized RG equation in tensor-network language have simple and clear pictorial representation, and more importantly, the results can be improved systematically by increasing the bond dimension.

Chapter 2

Real-space Renormalization Group

The renormalization group (RG) is a framework for asking the question: how does a physical system look like when some of its microscopic information that we are not interested in is thrown away? This way of thinking is powerful for studying physical systems where fluctuations in all scales of length are important, including critical phenomena, elementary particle physics, and turbulent fluid flow. In this thesis, we focus on the application of the RG framework to critical phenomena, and the above-mentioned process of throwing away information that we are not interested in is referred to as coarse graining.

The RG idea can be most easily grasped in the method of real-space RG when applied to classical lattice spin systems. In this chapter, we will use the simplest possible example, the decimation of the one-dimensional (1D) Ising model, to demonstrate the essence of the RG—how is coarse graining defined and what does the RG flows generated by the coarse graining process tell us about the spin system. We will then extract the general aspects of the RG from this simple example and summarize the general framework of the traditional RG prescription. The exposition in this chapter follow Kardar’s [32] and Cardy’s [7] textbooks, and the review article [15] closely.

For a detailed treatment of momentum-space RG in the context of field theory, see Kardar’s textbook [32]. Besides, in Shankar’s recent textbook [50], he clarified many subtle points that tend to confuse people a lot. For the historic background of how the RG framework was developed, see Wilson’s 1982 Nobel prize lecture [61].

2.1 A simple example: decimation of the one-dimensional Ising model

The real-space RG transformation of the Ising model in 1D can be made exact through decimation. We use this example to introduce some central concepts of the RG analysis: coarse graining, RG equations, fixed points, and linearized RG equations.

The partition function of the 1D Ising model is

$$Z_{1D} = \sum_{\{\sigma_j\}} \exp \left[\sum_{i=1}^N \mathcal{H}(\sigma_i, \sigma_{i+1}) \right], \quad (2.1)$$

where σ_i is the spin variable located at the i -th lattice point, taking values ± 1 , and the local interactions in \mathcal{H} (we will be sloppy and also call it Hamiltonian or energy of the system, although they may differ by an overall multiplication constant) involve the nearest-neighbor term at most,

$$\mathcal{H}(\sigma_1, \sigma_2) = g + \frac{h}{2}(\sigma_1 + \sigma_2) + K\sigma_1\sigma_2. \quad (2.2)$$

Here, \mathcal{H} is parametrized by three *coupling constants*, g, h, K , organized according to different interaction terms $1, \sigma_1, \sigma_2, \sigma_1\sigma_2$. Physically, h is related to the external magnetic field and K specifies the tendency of how much the nearest-neighbor spins want to align with each other, while g is an overall additional constant that do not affect the probability distribution of physical configurations but is useful for the RG analysis as will be seen later.

2.1.1 Coarse graining by decimation

Fig. 2-1 shows how the decimation is defined. It is realized by summing over all the even-numbered spins in the partition function, followed by a renumbering of the remaining odd-numbered spins. We denote $\sigma'_i = \sigma_{2i-1}$, $s_i = \sigma_{2i}$ and trace out all

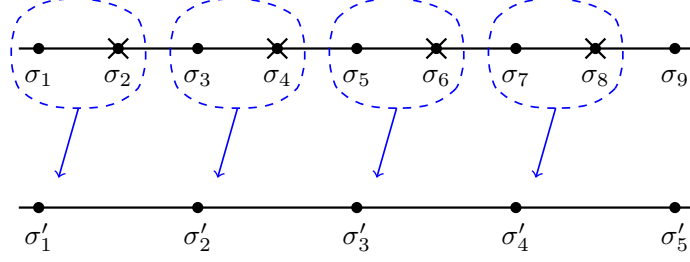


Figure 2-1: The decimation of the 1D Ising model, where the linear dimension of the system is larger by a factor of $b = 2$. The black dots are spin variables. We trace out all the spins on even sites $\sigma_2, \sigma_4, \dots$ in the partition function; the remaining spins $\sigma_1, \sigma_3, \dots$ are renamed $\sigma'_1, \sigma'_2, \dots$ to become new spin variables at larger scale of length.

s -spins in the partition function in Eq. (2.1) to have

$$\begin{aligned}
 Z_{1D} &= \sum_{\{\sigma'_j\}} \sum_{\{s_j\}} \exp \left[\sum_i^{N/2} [\mathcal{H}(\sigma'_i, s_i) + \mathcal{H}(s_i, \sigma'_{i+1})] \right] \\
 &= \sum_{\{\sigma'_j\}} \prod_i^{N/2} \left(\sum_{s_i=\pm 1} \exp [\mathcal{H}(\sigma'_i, s_i) + \mathcal{H}(s_i, \sigma'_{i+1})] \right), \quad (2.3)
 \end{aligned}$$

from which we can define the term in the large parenthesis as an effective local interaction

$$\exp [\mathcal{H}'(\sigma'_1, \sigma'_2)] = \sum_{s=\pm 1} \exp [\mathcal{H}(\sigma'_1, s) + \mathcal{H}(s, \sigma'_2)], \quad (2.4)$$

where the effective local interaction \mathcal{H}' has the same form as the old one in Eq. (2.2) but with new coupling constants g', h', K' ,

$$\mathcal{H}'(\sigma'_1, \sigma'_2) = g' + \frac{h'}{2} (\sigma'_1 + \sigma'_2) + K' \sigma'_1 \sigma'_2. \quad (2.5)$$

Replace the large parenthesis in Eq. (2.3) with the left-hand side of Eq. (2.4), so we see that the partition function can be fully described by the coarser σ' -spins,

$$Z_{1D} = \sum_{\{\sigma'_j\}} \exp \left[\sum_{i=1}^{N/2} \mathcal{H}'(\sigma'_i, \sigma'_{i+1}) \right]. \quad (2.6)$$

Equations (2.2), (2.4) and (2.5) together define the RG equation that maps the old coupling constants (g, h, K) to the new coupling constants (g', h', K') . The explicit expression of the RG equation is clearer if we work with proxy variables of coupling constants $(x, y, z), (x', y', z')$ defined as

$$\begin{aligned} x &= e^K, & y &= e^h, & z &= e^g, \\ x' &= e^{K'}, & y' &= e^{h'}, & z' &= e^{g'}. \end{aligned} \quad (2.7)$$

By examining all of the four possible configurations generated by setting $\sigma'_1, \sigma'_2 = \pm 1$ in Eq. (2.4), we get four conditions that must be satisfied between (x', y', z') and (x, y, z) ,

$$\begin{aligned} \text{for } \sigma'_1 = 1, \sigma'_2 = 1 : & \quad x'y'z' = z^2y(x^2y + x^{-2}y^{-1}), \\ \text{for } \sigma'_1 = -1, \sigma'_2 = -1 : & \quad x'(y')^{-1}z' = z^2y^{-1}(x^{-2}y + x^2y^{-1}), \\ \text{for } \sigma'_1 = 1, \sigma'_2 = -1 : & \quad (x')^{-1}z' = z^2(y + y^{-1}), \\ \text{for } \sigma'_1 = -1, \sigma'_2 = 1 : & \quad (x')^{-1}z' = z^2(y + y^{-1}). \end{aligned} \quad (2.8)$$

Since the last two equations are identical, we can solve for three renormalized x', y', z' in terms of the old x, y, z using the remaining three equations,

$$\begin{aligned} (x')^4 &= (x^2y + x^{-2}y^{-1})(x^{-2}y + x^2y^{-1})(y + y^{-1})^{-2}, \\ (y')^2 &= y^2(x^2y + x^{-2}y^{-1})(x^{-2}y + x^2y^{-1})^{-1}, \\ (z')^4 &= z^8(x^2y + x^{-2}y^{-1})(x^{-2}y + x^2y^{-1})(y + y^{-1})^2. \end{aligned} \quad (2.9)$$

The RG equation in Eq. (2.9) is the *key result* of the above coarse-graining process realized by decimation. We can transform all the proxy variables in Eq. (2.9) back to the coupling constants according to Eq. (2.7) to get a map $(g, h, K) \rightarrow (g', h', K')$, but the resultant expression of the RG equation would be very cumbersome and does not bring new insights; we will work with the RG equation in the form of Eq. (2.9) from now on.

Let us pause at this point and discuss the physical picture of the above analysis. During the decimation of the even-numbered spins in Fig. 2-1, one coarser spin σ'_i

represents two original spins $\sigma_{2i-1}, \sigma_{2i}$. Fix the index to $i = 1$, there are four original configurations $(\sigma_1, \sigma_2) = (+1, +1), (+1, -1), (-1, +1), (-1, -1)$ corresponding to two coarser configurations $\sigma'_1 = +1, -1$. The decimation relates the coarser configurations to the original ones according to $\sigma'_1 = +1 \iff (\sigma_1, \sigma_2) = (+1, +1), (+1, -1)$ and $\sigma'_1 = -1 \iff (\sigma_1, \sigma_2) = (-1, +1), (-1, -1)$. In terms of coarser spin σ'_i , the effective local interaction \mathcal{H}' is parametrized with new coupling constants g', h', K' as are defined in Eq. (2.5) and their relation to old ones is specified in Eq. (2.4), with the explicit form shown in the RG equation in Eq. (2.9). The RG equation defined by a given coarse graining process tells us *how the physics at larger scales emerges from lattice-scale physics*.

Mathematically, the process from microscopic descriptions to macroscopic observations is represented by an RG flow in the coupling-constant space with coordinates (g, h, K) , or equivalently, (x, y, z) , by applying the RG equation iteratively. The large-length-scale physics in thermodynamic limit is thus described by various fixed points of the RG equation, where the coupling constants stay the same under coarse graining.

2.1.2 Fixed points and linearized RG equations

Fixed points Different fixed points of the RG equation corresponds to different large-length-scale physics or phases of matter. Let us discuss the fixed points of the RG equation in Eq. (2.9). First notice that x', y' only depends on x, y but not z , which is reasonable from the physical meaning of the coupling constants; z is a proxy variable of g , an overall additional constant, which should not affect any nontrivial coupling like the external magnetic field or nearest-neighbor interaction. Next, for simplicity, let us focus on the $y = 1$ subspace, corresponding to zero magnetic field $h = 0$ (for $h \neq 0$ see Kardar's book [32]). The second expression in Eq. (2.9) tells us the coarser $y' = 1 = y$; the algebra works out correctly since the energy of the old system is invariable under spin-flip transformation $\sigma_i \rightarrow -\sigma_i$, and the RG transformation should not generate terms that break this symmetry. After the restriction $y = 1$, the first

expression in Eq. (2.9) becomes

$$(x')^2 = \frac{x^2 + x^{-2}}{2} \text{ or } e^{2K'} = \cosh 2K, \quad (2.10a)$$

which can be written as

$$\tanh K' = \tanh^2 K. \quad (2.10b)$$

This equation has two fixed points:

$$\begin{aligned} \text{high-temperature fixed point: } & \tanh K^* = 0, \quad K^* = 0, \quad T^* = \infty, \\ \text{low-temperature fixed point: } & \tanh K^* = 1, \quad K^* = \infty, \quad T^* = 0, \end{aligned} \quad (2.11)$$

where $T \propto 1/K$ is the temperature.

Linearized RG equations The behavior of the RG equation around a given fixed point reveals the property of the phase of the matter corresponding to this fixed point. When the fixed point corresponds to a critical system, the linearized RG equation around this fixed point tells us how various physical observables, like the correlation length, diverge, but the linearized RG equation itself is well-behaved without any divergence. Wilson's paper [59] used an analogy to explain how a continuous RG equation near a critical fixed point gives rise to divergence behavior of various physical observables. Later in Section 2.3.3, we will demonstrate how to extract the divergence behavior of the correlation length of the 2D Ising model at criticality from the linearized RG equation. Here, although the 1D Ising model does not have an interesting critical fixed point, we can still use it to demonstrate how to linearize the RG equation around a fixed point.

Let us focus on the high-temperature fixed point, where $K^* = h^* = 0$ or $x^* = y^* = 1$. Plug these values into the last express in Eq. (2.9), we have $z^* = 1/2$ or $g^* = \log(1/2)$.

Consider small perturbations around this fixed point and set

$$g = \log(1/2) + \delta g, \quad h = 0 + \delta h, \quad K = 0 + \delta K. \quad (2.12)$$

After the decimation, the coarse g', h', K' should also be close to the fixed point,

$$g' = \log(1/2) + \delta g', \quad h' = 0 + \delta h', \quad K' = 0 + \delta K'. \quad (2.13)$$

Use Eq. (2.7) to convert g, h, K in Eq. (2.12) to x, y, z , plug them into the right hand side of Eq. (2.9), and only keep the linear terms. A straightforward but tedious algebra gives

$$\begin{aligned} \delta g' &= 2 \times \delta g + \text{higher-order terms}, \\ \delta h' &= 1 \times \delta h + \text{higher-order terms}, \\ \delta K' &= 0 \times \delta K + \text{higher-order terms}. \end{aligned} \quad (2.14a)$$

It can also be put into a matrix form,

$$\begin{pmatrix} \delta g' \\ \delta h' \\ \delta K' \end{pmatrix} = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta g \\ \delta h \\ \delta K \end{pmatrix}. \quad (2.14b)$$

Equation (2.14) is the linearized RG equation near the high-temperature fixed point of the 1D Ising model; it can be represented by a 3-by-3 matrix, already in its diagonal form with eigenvalues 2, 1, 0 for $\delta g, \delta h, \delta K$ respectively. In general, *it is the eigenvalues of a linearized RG equation that are related directly with the divergence behavior of physical observables in a critical system* (not for this particular example, of course, since the system is not at criticality).

2.2 General framework

It is a nice point to generalize the RG analysis of the 1D Ising model and discuss the general framework of the traditional Hamiltonian-based RG, which will also be referred to as the *canonical RG prescription in Hamiltonian space* in this thesis. The discussion here follows Refs [7, 15] closely.

Let us say we have a classical system with spin variables $\sigma \in \{+1, -1\}$ on a lattice. The Hamiltonian (or energy) of the system is local and can be parameterized by a set of coupling constants $\mathbf{K} = \{K_j\}$, each of which couples to a possible short-ranged interaction term $s_j(\mathbf{r})$,

$$\mathcal{H} = \sum_{\mathbf{r}} \sum_j K_j s_j(\mathbf{r}). \quad (2.15)$$

For example, if K_1 is the magnetic field, $s_1(\mathbf{r}) = \sigma(\mathbf{r})$ is the spin variable at lattice point \mathbf{r} ; if K_2 is the nearest neighbor interaction along x direction, $s_2(\mathbf{r}) = \sigma(\mathbf{r})\sigma(\mathbf{r} + a\hat{\mathbf{e}}_x)$, where $\hat{\mathbf{e}}_x$ is the unit vector along x direction and a is the lattice constant. A conventional RG transformation maps the old Hamiltonian \mathcal{H} to a new one \mathcal{H}' with the same form as Eq. (2.15) but characterized by a set of new coupling constants $\mathbf{K}' = \{K'_j\}$. The map $\mathcal{H} \xrightarrow{\text{RG}} \mathcal{H}'$ is then parametrized explicitly as the transformation from the old coupling constants to the new ones,

$$\mathbf{K}' = \mathcal{T}^{\text{old}}(\mathbf{K}). \quad (2.16)$$

The RG transformation is constructed in such a way as to keep the partition function of the system invariant in principle, or to keep the change of the partition function as small as possible in practice. Furthermore, the RG equation should exhibit a fixed-point Hamiltonian \mathcal{H}^* parameterized by coupling constants \mathbf{K}^* , such that \mathbf{K}^* remains unchanged under the RG transformation,

$$\mathbf{K}^* = \mathcal{T}^{\text{old}}(\mathbf{K}^*). \quad (2.17)$$

The linearized RG equation around \mathbf{K}^* tells us the behavior of the RG equation in the vicinity of this fixed point. It can be determined by following the procedure below. We study how the points near \mathbf{K}^* change under the RG transformation. To this end, we plug $\mathbf{K} = \mathbf{K}^* + \delta\mathbf{K}$ into the RG equation in Eq. (2.16). The coarser coupling constants are $\mathbf{K}' = \mathbf{K}^* + \delta\mathbf{K}'$ by continuity of an RG transformation. The linearized RG equation around \mathbf{K}^* is a matrix \mathcal{R}^{old} telling us how $\delta\mathbf{K}'$ is related to $\delta\mathbf{K}$,

$$\delta K'_i = \sum_j \mathcal{R}_{ij}^{\text{old}} \delta K_j. \quad (2.18)$$

Since the eigenvalues of the linearized RG equation are related to physical observables, we usually want to diagonalize \mathcal{R}^{old} . Let us say it has right and left eigenvectors $\{\psi^\alpha\}, \{\phi^\alpha\}$ with the same set of eigenvalues $\{\lambda^\alpha\}$,

$$\sum_j \mathcal{R}_{ij}^{\text{old}} \psi_j^\alpha = \lambda^\alpha \psi_i^\alpha \quad \text{and} \quad \sum_i \phi_i^\alpha \mathcal{R}_{ij}^{\text{old}} = \lambda^\alpha \phi_j^\alpha. \quad (2.19)$$

The left eigenvector ϕ^α tells us a certain linear combination of δK_i like

$$h^\alpha = \sum_i \phi_i^\alpha \delta K_i, \quad (2.20)$$

transforms in a simple way under RG; we usually call h^α scaling fields. Similarly, a certain linear combination of interaction terms $s_j(\mathbf{r})$ according to the right eigenvector ψ^α are known as scaling operators

$$o^\alpha(\mathbf{r}) = \sum_j s_j(\mathbf{r}) \psi_j^\alpha. \quad (2.21)$$

The RG transformation with rescaling factor b for a system in dimension d is represented by a diagonal matrix in the basis specified by the scaling fields and the scaling operators

$$(h^\alpha)' = b^{d-x_\alpha} h^\alpha \quad \text{and} \quad (o^\alpha)' = b^{x_\alpha} o^\alpha, \quad (2.22)$$

where x_α are the scaling dimensions of the scaling operators $o^\alpha(\mathbf{r})$. Using Eqs (2.18)

to (2.20) and (2.22), we can find the relation between the scaling dimensions $\{x_\alpha\}$ and the eigenvalues $\{\lambda^\alpha\}$ of the linearized RG equation,

$$b^{d-x_\alpha} = \lambda^\alpha. \quad (2.23)$$

The scaling dimension x_α of the scaling operator o^α tells us the divergent behavior of the physical observables related to this operator.

In summary, in the conventional approach, we first find a fixed point of the RG equation and determine the linearized RG equation \mathcal{R}^{old} at this fixed point. Then, we find the eigenvalues λ^α of \mathcal{R}^{old} and calculate the scaling dimensions according to Eq. (2.23). Various critical exponents can be calculated from these scaling dimensions if desired.

2.3 Decimation of the two-dimensional Ising model

It is very unusual that the decimation of the 1D Ising model is exact. In this section, we use decimation of the two-dimensional (2D) Ising model to demonstrate the difficulty of the real-space RG encountered in higher dimensions and introduce a simple trick to deal with this difficulty.

2.3.1 Decimation in $d = 2$ generates new interactions

Consider the 2D Ising model on the square lattice, initially with only nearest-neighbor interaction. The decimation is performed as per Fig. 2-2. We concentrate on the five spins $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4$ shown in the figure, where σ_0 is decimated and mediates the effective interactions between $\sigma_1, \sigma_2, \sigma_3, \sigma_4$. We are tempted to assume that there is still only nearest-neighbor terms in the effective interactions due to our success in $d = 1$,

$$\sum_{\sigma_0=\pm 1} e^{K\sigma_0(\sigma_1+\sigma_2+\sigma_3+\sigma_4)} = e^{\frac{1}{2}K'(\sigma_1\sigma_2+\sigma_2\sigma_3+\sigma_3\sigma_4+\sigma_4\sigma_1)} e^{g'}, \quad (2.24)$$

where the $1/2$ factor in front of K' is because σ_0 only mediates half of the interaction between, say, σ_1 and σ_2 , with the other half contributed by the decimation of the spin σ' inside the dashed lines shown in the figure. However, we can see that Eq. (2.24) cannot be true based on a symmetry argument. The left-hand side of Eq. (2.24) is $2 \cosh[K(\sigma_1 + \sigma_2 + \sigma_3 + \sigma_4)]$, where four spins appear in a symmetric way; it is invariant under the permutation $\sigma_1 \leftrightarrow \sigma_2$. The right-hand side of Eq. (2.24) does not respect this symmetry. Therefore, second-neighbor terms like $K_2\sigma_1\sigma_3$ must arise. It turns out that there is also a four-spin term $K_4\sigma_1\sigma_2\sigma_3\sigma_4$. As a result, we should change Eq. (2.24) to become

$$\sum_{\sigma_0=\pm 1} e^{K\sigma_0(\sigma_1+\sigma_2+\sigma_3+\sigma_4)} = e^{\frac{1}{2}K'(\sigma_1\sigma_2+\sigma_2\sigma_3+\sigma_3\sigma_4+\sigma_4\sigma_1)} e^{K'_2(\sigma_1\sigma_3+\sigma_2\sigma_4)+K'_4\sigma_1\sigma_2\sigma_3\sigma_4} e^{g'}. \quad (2.25)$$

By examining any three independent spin configurations not related by the overall spin-flip transformation, we can solve for the coarser coupling constants in terms of

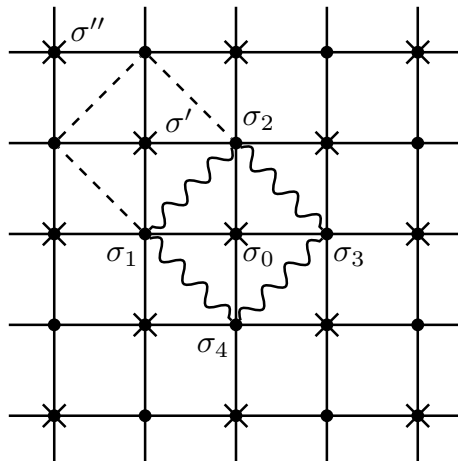


Figure 2-2: The decimation of the 2D Ising model. The black dots are spin variables. The spins that are marked with a cross sign like σ_0 are decimated, which will mediate new interactions including second-neighbor terms like $K'_2\sigma_1\sigma_3$ and a four-spin term $K'_4\sigma_1\sigma_2\sigma_3\sigma_4$. The lattice constant is $b = \sqrt{2}$ times as the original one.

the original ones,

$$K' = \frac{1}{4} \ln \cosh 4K, \quad (2.26a)$$

$$K'_2 = \frac{1}{8} \ln \cosh 4K, \quad (2.26b)$$

$$K'_4 = \frac{1}{8} \ln \cosh 4K - \frac{1}{2} \ln \cosh 2K. \quad (2.26c)$$

Although we start with only the nearest-neighbor interaction, the decimation in $d = 2$ generates extra second-neighbor and four-spin terms, making subsequent decimation steps less trivial. To see the problem these newly-generated interactions cause in subsequent coarse graining, imagine that we rotate and rescale the system after the above decimation so it looks the same as before, and then repeat the decimation. Let us examine Fig. 2-2 again, but this time with interactions K', K'_2, K'_4 among the spins in a plaquette like the one containing $\sigma_1, \sigma', \sigma_2, \sigma_0$. During this second coarse-graining step, we can no longer treat the local patch containing $\sigma_0, \sigma_1, \sigma_2, \sigma_3, \sigma_4$ separately from the rest system due to the second-neighbor interaction K'_2 between σ_0 and σ' . Instead, the sum over σ_0 is related to the sum over σ' , which is further related to more spins outside like σ'' . We have to make approximations to make the subsequent decimation steps well-defined.

2.3.2 The Migdal-Kadanoff bond moving approximation

One way to keep the interactions from growing is to move the problematic interactions to non-problematic ones by hand, often known as the Migdal-Kadanoff bond moving approximation [44, 43, 30], before the decimation. Consider the same lattice as the one in Fig. 2-2, but this time, for an RG increasing the lattice distance by a factor of $b = 2$ instead of $b = \sqrt{2}$. As is shown in Fig. 2-3, the bonds original at the dashed lines are moved to the upper or right bonds, enhancing the interactions there. Then, the unwanted spins are summed over. Since the spins like σ_2, σ_4 are decoupled from other spins, the paths of generating longer interactions are cut off; the decimation thus only generated nearest-neighbor interactions at most, just like the $d = 1$ case

shown in Fig. 2-1. After the bond moving trick, the nearest-neighbor interaction is increased from K to $2K$. After the decimation, the coarser interaction K' is given by Eq. (2.10),

$$\tanh K' = \tanh^2(2 \times K) = \frac{4 \tanh^2 K}{(1 + \tanh^2 K)^2}. \quad (2.27)$$

This trick of moving the bonds before decimation can be easily generated to higher dimensions. For a hypercubic lattice in d -dimensions, the bonds retained are strengthened by a factor of 2^{d-1} , $K \rightarrow 2^{d-1}K$; the subsequent decimation will then give the RG equation,

$$\tanh K' = \tanh^2(2^{d-1}K). \quad (2.28)$$

However, the approximation errors during the Migdal-Kadanoff bond moving step are difficult to control; the reliability of its results is thus questionable. Its performance

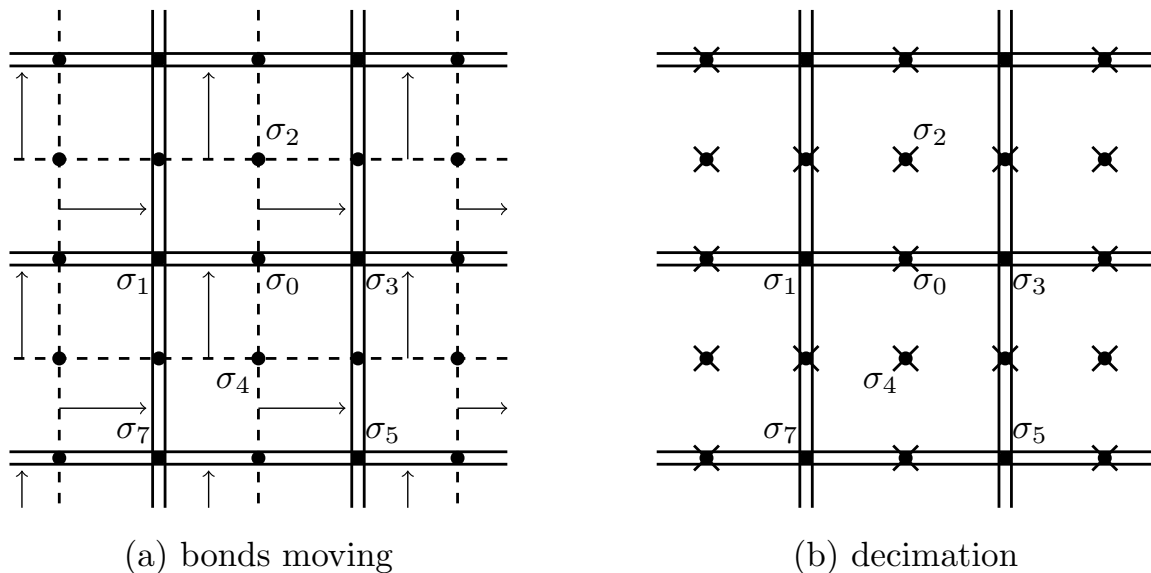


Figure 2-3: The bond moving trick for the Migdal-Kadanoff RG with $d = 2$ and $b = 2$. (a) Move the bonds not connected to the spins that will be retained after decimation. (b) Perform the decimation. Summing over spins like σ_2, σ_4 will only contribute to the overall multiplicative constant before the partition function, while summing over spins like σ_0 will mediate interactions between σ_1, σ_3 in a same way as the $d = 1$ case shown in Fig. 2-1.

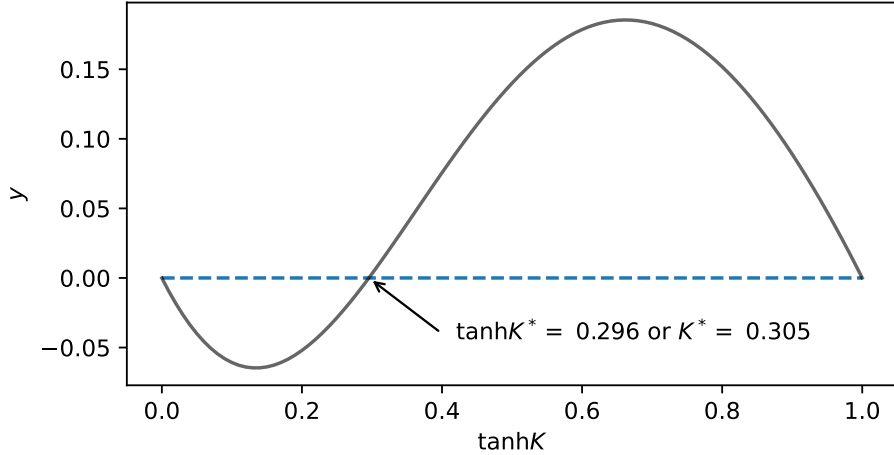


Figure 2-4: The roots of this function are the fixed points of Eq. (2.27).

on the Ising models in $d = 1, 2, 3$ indicates the results become worse in higher dimensions [32].

2.3.3 The linearized RG equation and divergence of the correlation length

To find the fixed points of the RG equation (2.27), we plot the function $y = f(\tanh K) = 4 \tanh^2 K / (1 + \tanh^2 K)^2 - \tanh K$ from $\tanh K = 0$ to 1, and the roots of the function are fixed points. As is shown in Fig 2-4, besides high-temperature fixed point $\tanh K^* = 0$ and low-temperature fixed point $\tanh K^* = 1$, the RG equation in Eq. (2.27) also has a non-trivial fixed point $\tanh K^* \approx 0.29560$ or $K^* \approx 0.305$, compared with the exact critical value $K_c \approx 0.441$. Since the RG equation (2.27) only involves one coupling constant, the linearized RG equation is simply the derivative of K' with respect to K evaluated at K^* ,

$$\delta K' = \left. \frac{\partial K'}{\partial K} \right|_{K^*} \delta K = 2 \tanh(4K^*) \delta K. \quad (2.29)$$

At the critical fixed point, $K^* = 0.305$, the linearized RG equation (2.29) tells us

$$\delta K' = 1.6786 \times \delta K, \quad (2.30)$$

which pushes the small perturbation away from the fixed point since $1.6786 > 1$.

Divergence of dimensionless correlation length Let us denote $\xi(K)$ as the dimensionless correlation length measured in lattice units; the physical correlation length is $\xi(K)$ multiplied by the original lattice constant. After the above $b = 2$ decimation, the coarser dimensionless correlation length

$$\xi(K') = \frac{1}{2}\xi(K), \quad (2.31)$$

because the coarser lattice constant is twice as big as the old one, while the physical correlation length is the same after the coarse graining. Right at the fixed point, $K' = K = K^*$, so according to Eq. (2.31), $\xi(K^*) = 0$ or ∞ . The critical fixed point is featured by a divergent correlation length, following a power law as a function of $\delta K = K - K^*$,

$$\xi(\delta K) = c_0|\delta K|^{-\nu}, \quad (2.32)$$

where $\nu > 0$ is known as a critical exponent, describing the divergent behavior of a physical observable. Use Eq. (2.32) in Eq. (2.31) and use the relation between $\delta K'$ and δK in Eq. (2.29) to have

$$\begin{aligned} |\delta K'|^{-\nu} &= \frac{1}{2}|\delta K|^{-\nu}, \\ \xrightarrow{\text{Eq. (2.30)}} 1.6786^{-\nu}|\delta K|^{-\nu} &= \frac{1}{2}|\delta K|^{-\nu}, \end{aligned} \quad (2.33)$$

from which we have

$$\nu = \frac{\ln(2)}{\ln(1.6786)} = 1.34. \quad (2.34)$$

The exact value given by Onsager's solution is $\nu = 1$. We use this example to demonstrate a general principle: various critical exponents are related directly with the eigenvalues of the linearized RG equation around a critical fixed point.

Chapter 3

Tensor-network Renormalization Group

The Migdal-Kadanoff bond moving approximation discussed in Section 2.3.2 is uncontrollable. One way to proceed is to construct a perturbative expansion like Niemeijer-van Leeuwen cumulant approximation or Wilson's ϵ expansion, discussed in many textbooks. Here we want to introduce a modern RG method, tensor-network RG; they are versatile numerical schemes whose approximations are controlled by an integer, χ , called the *bond dimension*.

3.1 From a lattice with statistical variables to a tensor-network model

Tensor-network RG is to apply the RG idea to a set of models called tensor-network models. It can be shown that the partition function of a classical statistical model or the space-time path integral of a quantum spin system can be rewritten as tensor-network models [24, 38]. In this chapter, we use the square lattice two-dimensional Ising model as a concrete example to demonstrate how to find a tensor-network representation of a partition function. There are at least two ways to construct the tensor-network model. The method introduced by Hauru, Delcamp and Mizera [26] is

more general and can be trivially adapted to apply to the 3D classical Ising model [63] and other models like the ϕ^4 theory in 2D [13]. However, we will use the other method introduced in Refs [18, 38], which is more straightforward and whose physical meaning is more transparent.

Consider classical spin variables living on a square lattice shown in Fig. 3-1. The square lattice is sketched with blue dashed lines and the blue dots are where spin variables sit. The partition function is

$$Z = \sum_{\{\sigma(\mathbf{r})\}} e^{K \sum_{\langle i,j \rangle} \sigma_i \sigma_j}, \quad (3.1)$$

where σ_i is the shorthand for the spin variable $\sigma(\mathbf{r}_i)$ located at lattice point \mathbf{r}_i and can take values ± 1 ; $K = J/k_B T$, where $J > 0$ is the ferromagnetic interaction and T is temperature. In this thesis, we measure temperature in units of J/k_B so it becomes a dimensionless number.

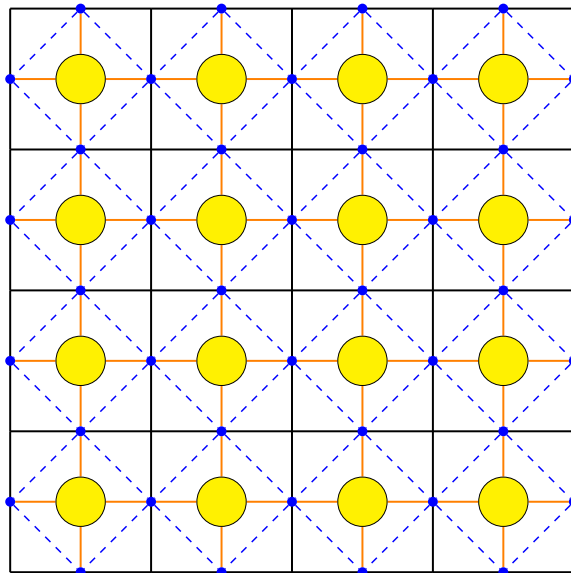
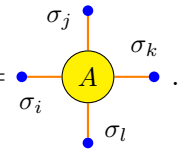


Figure 3-1: Map spin variables to tensors. The blue dots are where the spin variables locate, forming a square lattice slanted by 45° . The larger green circles are tensors encoding the probability distribution of the configurations of the four surrounding spin variables sitting on four legs of the tensor; the tensors form another square lattice.

Now, imagine dividing this region into square blocks using the black solid lines in Fig. 3-1, such that each side of a black square holds a spin variable. Finally, put a

four-leg tensor (green circles with four green legs) at the center of each black square, with its four legs connected with the four spin variables on the sides of the black square. The graph we get in green is the tensor network, where the four-leg tensors connected with each other to form a new square lattice. To determine the value of the four-leg tensors, we simply divide the summation on the exponential in Eq. (3.1) into groups where pairs of spin variables are brought together if they belong to the same black square. We define the local Boltzmann weight for pairs of spins on the same black square as a 4-leg tensor

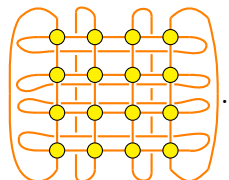
$$A_{\sigma_i \sigma_j \sigma_k \sigma_l} \equiv e^{K(\sigma_i \sigma_j + \sigma_j \sigma_k + \sigma_k \sigma_l + \sigma_l \sigma_i)} = \text{Diagram} \quad (3.2)$$


Each index of this tensor can take two values ± 1 and we say the *bond dimension* of a leg of this tensor is $\chi = 2$. It is now possible to rewrite the partition function of the 2D Ising model in Eq. (3.1) as the tensor product of N copies of A , with all their indices summed over (Fig. 3-1)

$$Z = \sum_{\{\sigma(\mathbf{r})\}} \bigotimes_{x=1}^N A_{\sigma_{i(x)} \sigma_{j(x)} \sigma_{k(x)} \sigma_{l(x)}} \equiv \text{tTr} \left(\bigotimes_{x=1}^N A \right), \quad (3.3)$$

where the last equal sign defines the tTr symbol. The index x is to distinguish different copies of A .

As a concrete example, for a finite system in Fig. 3-1 with periodic boundary condition, the total $4 \times 8 = 32$ spin variables are mapped into $4 \times 4 = 16$ tensors A . If we adopt the *summation convention that a leg is summed over whenever it is shared by two tensors*, the partition function in Eq. (3.1) will have a simple pictorial description

$$Z_{4 \times 4} = \text{Diagram} \quad (3.4)$$


Eq. (3.4) above provides a tensor network representation of the partition function in Eq. (3.1). It is the starting point of all the tensor-network renormalization group methods. Notice that the number of tensors in the tensor network is half of the number of spin variables. In addition, if the lattice constant of the original spin system is $a/\sqrt{2}$ (that is, the length of a side of a blue dashed square in Fig. 3-1), the distance of the nearest two tensors in the resultant tensor network is $a/\sqrt{2} \times \sqrt{2} = a$. Compared with the original picture of spin variables, the tensor-network picture encodes more information about the probability distribution of local spin configurations into a tensor, the legs of which represent the original spin variables.

3.2 Coarse graining of a tensor-network model

The coarse graining of a tensor-network model is similar to the conventional block-spin methods. The transformation is usually defined by grouping several old variables to form a new variable. The system consisting of the new variables will have a different length scale (in particular, a different lattice constant) but its partition function should be the same as (or a good approximation of) the partition function of the system consisting of the old variables. Interestingly, the coarse graining in the tensor representation is easier to define and more transparent. Let's focus on a concrete example whose partition function is defined in Eq. (3.4).

3.2.1 A naïve approach

The system originally consists of sixteen A tensors. The simplest way to define an RG transformation is to block a square of four tensors by contracting legs between them and group every two legs on the same side. Call the coarser tensor A_c ,

$$\text{Diagram of } A_c \equiv \text{Diagram of } A \text{ block} \quad (3.5)$$

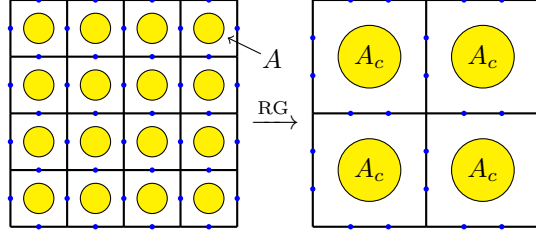


Figure 3-2: The physical picture of the block tensor transformation $A \rightarrow A_c$ in Eq. (3.5). Part of the spins shared by two tensors A are summed over according to Eq. (3.5). The black squares are larger after the decimation.

Then, the partition function can be fully described by copies of the coarser tensor A_c after the coarse graining

$$Z_{4 \times 4} = \text{[Diagram of a 4x4 grid of tensors A with orange and blue lines connecting them]} \stackrel{\text{RG}}{=} \text{[Diagram of a 2x2 grid of tensors A_c with orange lines connecting them]}, \quad (3.6)$$

so the coarse graining defined by the above block-tensor transformation is exact.

We can draw the original spin variables explicitly in the tensor-network figure to see what is happening to the spin variables under the block-tensor transformation. To avoid too many lines clustering together, we only keep spin variables, tensors and black solid squares in Fig. 3-1. The big picture for the block-tensor transformation is shown schematically in Fig. 3-2. The block-spin transformation resembles the conventional decimation. The contracting of the tensor network on the right-hand side of Eq. (3.5) represents the decimation of spin variables sitting on the legs contracted. We end up with four bigger black squares, each with a A_c in the middle and two spin variables sitting on any of its four sides. The length scale is multiplied by $b = 2$. This block-tensor transformation in Eq. (3.5) then defines a map from the old tensor A to

the coarser tensor A_c ,

$$\text{---} \circlearrowleft A \text{---} \xrightarrow[\text{graining}]{\text{coarse}} \text{---} \circlearrowleft A_c \text{---} . \quad (3.7)$$

Since the tensors A, A_c encode the probability distributions μ, μ' of the original and coarser spin configurations respectively, equation (3.7) provides a first example of the map $T : \mu \rightarrow \mu'$ in Fig. 1-1. This coarse graining is exact, but the bond dimension grows fast, which is expected based on the example of the decimation of the 2D Ising model shown in Section 2.3. The initial tensor has bond dimension $\chi = 2$. After n RG transformations, the bond dimension grows to 2^{2^n} . It can be easily shown that the computation cost will grow exponentially in the original lattice size. A clever approximation technique is necessary to prevent the bond dimension from growing as the RG transformation goes.

In the conventional RG approaches, we avoid generating a Hamiltonian with infinitely many interaction terms by only keeping a finite number of the couplings that are important quantitatively [61, 58]. This statement was confirmed by Wilson in his 1975 numerical RG calculations of the 2D Ising model using Kadanoff's decimation approach [60]. This observation suggests that we should be able to find a good approximation of a local patch of the tensors, such as the square of four A tensors in Eq. (3.5). We will introduce such an approximation scheme in the next subsection.

3.2.2 An approximation scheme: the higher-order tensor renormalization group (HOTRG)

The higher-order tensor renormalization group (HOTRG) [63] is the earliest tensor-network RG technique that can be easily applied to three-dimensional classical statistical models. It is simple to understand, easy to implement, and straightforward for the RG prescription in tensor-network language, which is the main purpose of this thesis.

As has been explained at the end of the previous subsection, we want to approximate the square of four A tensors on the right-hand side of Eq. (3.5) to prevent the growing of the bond dimensions. In the original treatment of the HOTRG [63], the approximation was explained in terms of higher-order singular value decompositions [10, 11, 12]. However, we will present the HOTRG from a different perspective inspired by a broader class of local approximation methods called projective truncations [18]. Let's imagine we first contract two A tensors on the left in Eq. (3.5). A projection operator $P = ww^\dagger$ is inserted and acts on, say, two left legs whose bond dimensions are both assumed to be χ , and we hope this patch of two A tensors after projection gives a good approximation to the original patch,

$$(3.8)$$

where w is an isometric tensor to be determined and w^\dagger its hermitian conjugate. The isometry w is a linear mapping: $\mathbb{V}_{\tilde{\chi}} \rightarrow \mathbb{V}_{\chi} \otimes \mathbb{V}_{\chi}$, where \mathbb{V}_{χ} denotes a χ -dimensional vector space, and the isometry satisfies $w^\dagger w = \mathbb{1}$. In a more physicist-oriented language, the isometry w is nothing but a collection of $\tilde{\chi}$ orthonormal ket vectors with dimensionality χ^2 in a given representation. We draw it as a gray triangle to mimic Dirac's bra-ket notation. If we fix the leg connecting the vertex angle of the gray triangle to be k -th index, we get a ket vector $|w_k\rangle$. The projection operator is $P = \sum_{k=1}^{\tilde{\chi}} |w_k\rangle\langle w_k|$. If $\tilde{\chi} = \chi^2$, we have a complete orthonormal set and the projection operator becomes the identity operator. We will see later that to prevent the growing of the bond dimensions, we should choose $\tilde{\chi} \leq \chi$. We usually choose $\tilde{\chi} = \chi = \chi_{\max}$ in this thesis, so the bond dimension remains the same value χ_{\max} after coarse graining (more precisely, in first few RG steps, the bond dimension will grow from 2 to χ_{\max} for case of the 2D Ising model).

Now, suppose we have found the isometry w that gives a good approximation in Eq. (3.8), then we use this approximation to replace all pairs of A tensors in the tensor

network representation of the partition function in Eq. (3.4) to get

$$\begin{aligned}
 Z_{4 \times 4} &\stackrel{(3.8)}{\approx} \text{Diagram 1} \\
 &= \text{Diagram 2}, \tag{3.9}
 \end{aligned}$$

where during the second step, we contract two A tensors and w, w^\dagger in the dashed circle to get a coarser tensor A' ,

$$\text{Diagram 3} \equiv \text{Diagram 4}. \tag{3.10}$$

Notice in the approximation step in Eq. (3.9), we use the periodic boundary condition to move the two leftmost w tensors to the right. Equation (3.10) defines a coarse graining $A \rightarrow A'$ in the vertical direction. Apply a similar projective truncation on two A' tensors put side by side using another isometry v , we will get a coarse graining $A' \rightarrow A_c$ in the horizontal direction. The total coarse graining $A \rightarrow A_c$ is a combination of $A \rightarrow A', A' \rightarrow A_c$,

$$\text{Diagram 5} \equiv \text{Diagram 6}. \tag{3.11}$$

The legs of tensor A_c have the bond dimensions $\tilde{\chi} = \chi = \chi_{\max}$, so the HOTRG implements an approximate RG transformation which prevents the bond dimensions from growing. The computational costs of the tensor contraction in Eq. (3.11) are $O(\chi_{\max}^7)$.

Determining the isometric tensors The final question is how to determine the isometry w , so that we have a good approximation in Eq. (3.8). The approximation error is usually quantified by the Frobenius norm of the difference of two tensors. To simplify the notation, we combine the two left legs of the patch of two A tensors on right-hand side of Eq. (3.8) as one group, and combine its remaining four legs as the other group; treat the patch as a matrix M ,

$$M = \begin{array}{c} \chi \\ \chi \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right] M \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \equiv \begin{array}{c} \chi \\ \chi \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right] \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} A \\ A \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} . \quad (3.12)$$

The square of the approximation error ϵ of Eq. (3.8) is

$$\begin{aligned} \epsilon^2 &= \|M - ww^\dagger M\|^2 \\ &= \text{Tr} \left((M - ww^\dagger M) (M - ww^\dagger M)^\dagger \right) \\ &= \text{Tr} (MM^\dagger) - \text{Tr} (w^\dagger MM^\dagger w) , \end{aligned} \quad (3.13a)$$

where in the last step, we expand the two parentheses, apply the cyclic property of trace, and use the property of an isometry, $w^\dagger w = \mathbb{1}$. Notice the first term $\text{Tr} (MM^\dagger)$ is a constant since matrix M is given. The result can be put pictorially as

$$\epsilon^2 = \text{Const.} - \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right] M \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \left[\begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \right] M^\dagger \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \begin{array}{c} w^\dagger \\ w \end{array} . \quad (3.13b)$$

Since the square of any number is non-negative, we have $\text{Tr} (w^\dagger MM^\dagger w) \leq \text{Tr} (MM^\dagger)$. It follows that ϵ in Eq. (3.13) is minimized when $\text{Tr} (w^\dagger MM^\dagger w)$ is maximized. This is a

well-known optimization problem that can be converted into an eigenvalue problem [21]. In Appendix A, we will use the method of Lagrangian multiplier to prove this statement. The optimal isometry w is a collection of $\tilde{\chi}$ eigenvectors of the positive semi-definite matrix MM^\dagger corresponding to the first $\tilde{\chi}$ largest eigenvalues. For example, the ket vector $|w_k\rangle$ corresponding to the column vector we get when fixing the leg connecting the vertex angle of w (see, for example, the figure in Eq. (3.13b)) should satisfy

$$MM^\dagger|w_k\rangle = \lambda_k|w_k\rangle, \quad (3.14)$$

where λ_k is the k -th largest eigenvalues of MM^\dagger . The square of the approximation error ϵ^2 in Eq. (3.13) is the sum of $\chi^2 - \tilde{\chi}$ smallest eigenvalues we throw away during the approximation.

3.3 Systematic improvement by including more interactions

From the way how the isometric tensors are determined discussed in Section 3.2.2, we know that the approximation error ϵ^2 becomes smaller if more eigenvectors of MM^\dagger are kept, corresponding to larger bond dimension $\tilde{\chi}$. Therefore, we expect that *increasing the bond dimension $\tilde{\chi}$ can improve the final results of the HOTRG systematically*. This is very reasonable. Recall the physical picture of the block-tensor transformation in Fig. 3-2 and the meaning of an isometric tensor described below Eq. (3.8), larger bond dimension $\tilde{\chi}$ means keeping more spin configurations, which in turn indicates that the 4-leg coarser tensor A_c in Eq. (3.11) encodes more kinds of local interactions.

We demonstrate this claim numerically by comparing the free energy of the 2D Ising model estimated using the HOTRG with Onsager's exact result. We start with the initial tensor $A^{(0)} \equiv A$ shown in Eq. (3.2), which corresponds to two spin variables. By applying the RG equation of the HOTRG shown in Eq. (3.11), the number of copies of the initial tensor is increased by a factor of 4. After n iterations according

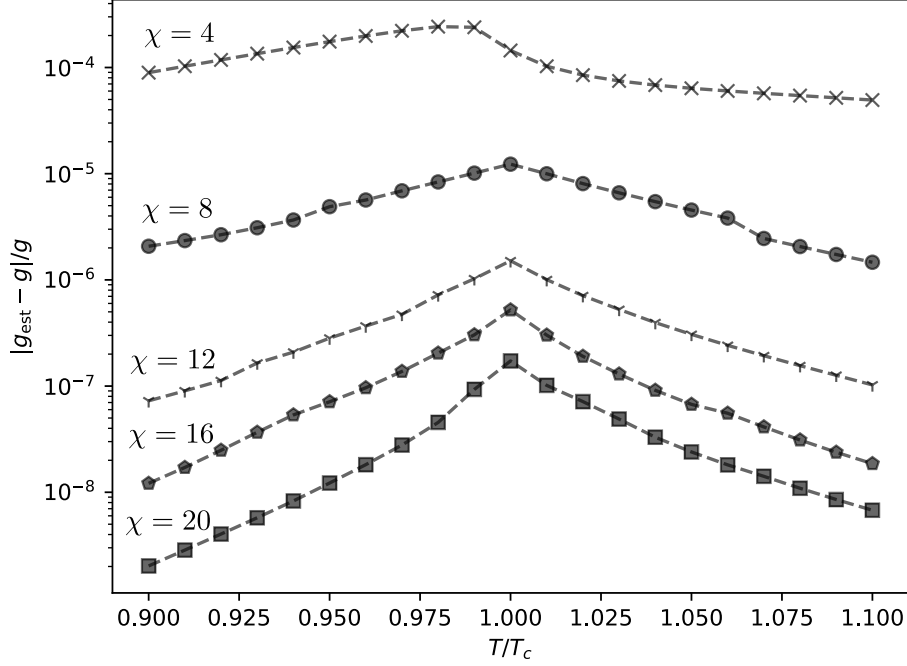


Figure 3-3: Relative error of free energy per spin g_{est} estimated at bond dimensions $\chi = 4, 8, 12, 16, 20$ using the HOTRG. T_c, g are exact values of the critical temperature and free energy per spin of the 2D Ising model according to Onsager's solution.

to the RG equation, the coarser tensor $A^{(n)}$ corresponds to $\mathcal{N} = 2 \times 4^n$ original spin variables. The partition function $Z_{\mathcal{N}}$ of these \mathcal{N} original spin variables with only the nearest-neighbor interaction and a periodic boundary condition is given by tracing out the opposite legs of the 4-leg tensor $A^{(n)}$,

$$Z_{\mathcal{N}} = \text{Tr} \left[A^{(n)} \right]. \quad (3.15)$$

The estimated free energy per spin g_{est} is thus,

$$g_{\text{est}} = \frac{\ln(Z_{\mathcal{N}})}{\mathcal{N}}, \quad (3.16)$$

which can be compared with the exact solution g . The relative errors $|g_{\text{est}} - g|/g$ at different temperature T/T_c are plotted in Fig. 3-3, where T_c is the exact value of the critical temperature. The relative errors decay exponentially as the bond dimension

increases, while the computation costs only grow polynomially like $O(\chi^7)$. It is quite amazing that 21 iterations of the HOTRG (corresponding to roughly 10^{13} spins) at bond dimension $\chi = 20$ can produce estimations of the free energy of the 2D Ising model with an error of order 10^{-7} within one minute in a desktop computer!

Chapter 4

Renormalization Group Prescription in Tensor-network Language

The great success of the tensor-network RG in producing accurate free energy demonstrated at the end of Chapter 3 is unprecedented. As long as the partition function of the physical model in hand admits a tensor-network representation, the resultant initial tensor can be fed into the tensor-network RG equation, like the one of the 2D HOTRG shown in Eq. (3.11), recursively, and out spit is an accurate estimation of the free energy. Tensor-network RG schemes are thus both versatile and systematically improvable. Later developments of these techniques improve the accuracy of the approximation during the leg squeezing step even further by considering a larger environment [64, 67, 45] or manipulating the entanglement structure of the network [24, 18, 16, 65, 3].

Nevertheless, free energy cannot be compared directly with experiment results. For example, if the physical problem under consideration is a critical system, various critical exponents, like the ν in Eq. (2.32), are physical observables of great interest. A question arises naturally: Can we apply the RG prescription of the conventional Hamiltonian-based RG in a similar manner in the context of the modern tensor-network-based RG methods? If the answer is positive, we then have a kind of versatile numerical RG methods in real space, being able to produce systematically-improvable estimations of critical exponents directly. In this chapter, we propose a definition of

the *canonical RG prescription in tensor-network language* and provide a theoretical justification for the definition by considering its connection with the conventional Hamiltonian-based way. In the next chapter, the validity of the definition will be checked in the context of the two-dimensional Ising model, using an HOTRG-like scheme.

4.1 Heuristic argument

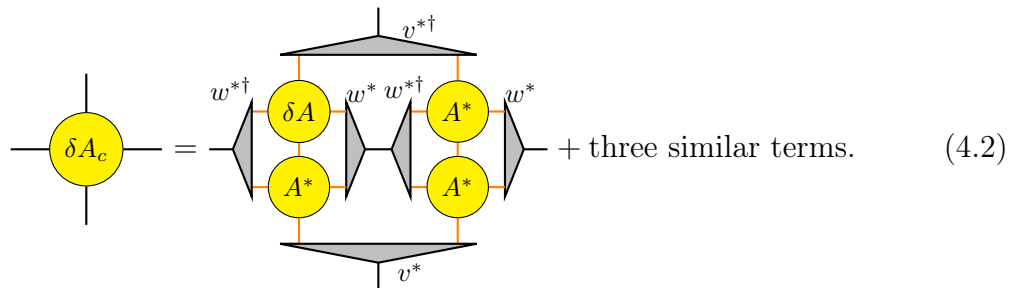
A heuristic proposal of the RG prescription in tensor-network language is manifest if we make an analogy to the conventional approach. As is shown in the example of the 1D Ising model in Section 2.1, there are two steps after having an RG equation in hand: *identify fixed points and linearize the RG equation around the fixed points*. The critical exponents (or equivalently, scaling dimensions) are obtained from the eigenvalues of the linearized RG equation around the corresponding critical fixed point. For concreteness, let us mimic this conventional approach using the RG equation of the HOTRG in Eq. (3.11) in the context of the 2D Ising model.

Step 1: Identify the critical fixed-point tensor A^* By fine-tuning the inverse temperature K in the initial tensor shown in Eq. (3.2), it is expected to reach the vicinity of the critical fixed-point tensor of the 2D Ising model by applying the RG equation iteratively, such that

The diagram shows a tensor A^* on the left, represented as a yellow circle with a vertical line extending upwards and downwards. This is equated to a contraction of two A^* tensors. The two A^* tensors are arranged in a 2x2 grid. The top-left and bottom-right corners are A^* tensors, and the top-right and bottom-left corners are A^* tensors. The top and bottom edges of the grid are connected by gray trapezoidal shapes labeled $v^{*\dagger}$ (top) and v^* (bottom). The left and right edges of the grid are connected by gray trapezoidal shapes labeled $w^{*\dagger}$ (left) and w^* (right). The equation is labeled (4.1) on the right.

where we add an asterisk on every tensor to indicate that the above equation is at a fixed point; it does *not* mean complex conjugate.

Step 2: Linearize the RG equation around the fixed-point tensor Consider small perturbation around this fixed-point tensor $A = A^* + \delta A$, plug into the right-hand side of Eq. (3.11) and collect all four first-order terms to have the linearized RG equation,



By diagonalizing the linear transformation defined by Eq. (4.2), critical exponents (or scaling dimensions) can be extracted.

This heuristic proposal is almost correct, except for two caveats: local correlations and gauge redundancy. We will discuss these two caveats in detail in Section 4.3, and propose a practical way to deal with them in Chapter 5. After the two caveats are handled, the only change is a *modification of the isometric tensors w, v* by incorporating the tensors coming from the procedures to address the caveats (see Eq. (5.15) about this modification). However, let us first discuss why this definition of the RG prescription in tensor-network language is reasonable by unveiling its connection with the well-established Hamiltonian-based approach.

4.2 Connection with the Hamiltonian-based approach

In this section, we will see the value of the tensor in the modern tensor-network RG plays the same role as the coupling constants in Hamiltonian-based RG. Mathematically, this means the RG equation in tensor-network language in principle represents the same linear transformation as the RG equation in Hamiltonian-based language does, but expressed in a different set of basis. We demonstrate this point by first working out the RG prescription of the 1D Ising model in tensor-network language, followed by a general argument inspired by this concrete example, where a formal

definition of the *canonical RG prescription in tensor-network language* is provided.

4.2.1 Decimation of the 1D Ising model in tensor-network language

We have used decimation of the 1D Ising model in Section 2.1 to demonstrate the idea of real-space RG, where we focus on finding a map from an old Hamiltonian to a coarser one. We see that the decimation for the 1D Ising model is an exact real-space RG transformation. In this section, we will see that, even better, the decimation has a natural tensor network representation. This makes the Ising model in 1D a nice example to see the relation of the canonical RG prescription between the tensor-network and the conventional Hamiltonian-based approaches.

We start by defining the tensor A sitting on the bond connecting two spins shown in Fig. 4-1 as (also look at Fig. 2-1)

$$A_{\sigma_1\sigma_2} = \exp[\mathcal{H}(\sigma_1, \sigma_2)]. \quad (4.3a)$$

Plug in the expression for \mathcal{H} in Eq. (2.2) and remember that $\sigma_1, \sigma_2 = \pm 1$, we have

$$A = \begin{pmatrix} \exp(g + h + K) & \exp(g - K) \\ \exp(g - K) & \exp(g - h + K) \end{pmatrix}, \quad (4.3b)$$

which is the familiar transfer matrix. We can use the tensor A to rewrite the partition

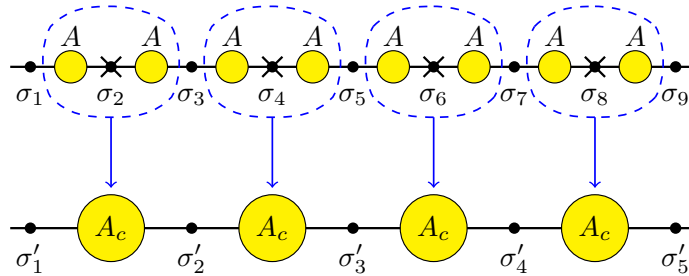


Figure 4-1: In the tensor network language, the decimation in Fig. 2-1 is nothing but a matrix multiplication of two transfer matrices to form a coarse-grained matrix $A_c = AA$.

function of the 1D Ising model in Eq. (2.1) as a tensor network shown in Fig. 4-1,

$$Z_{1D} = \sum_{\{\sigma_j\}} \bigotimes_{i=1}^N A_{\sigma_i \sigma_{i+1}}. \quad (4.4)$$

In the decimation, the spins on even sites $\sigma_2, \sigma_4, \dots$ are summed over in the partition function. In the tensor-network language, the spins become legs (or indices) of the tensor, so the decimation becomes a multiplication of every two tensors,

$$A_c = AA. \quad (4.5)$$

The partition function can now be expressed using the coarser tensor A_c ,

$$Z_{1D} = \sum_{\{\sigma'_j\}} \bigotimes_{i=1}^{N/2} (A_c)_{\sigma'_i \sigma'_{i+1}}. \quad (4.6)$$

Equation (4.5) is the RG equation in tensor-network language. If we examine both sides of Eq. (4.5) component by component, we will restore Eq. (2.8) in Chapter 2. What we do in the conventional RG prescription is to impose that A_c should have the same form as A in Eq. (4.3) but with new coupling constants g', h', K' . However, we can treat Eq. (4.5) as a legitimate RG equation, a map from old A to coarser A_c , on its own, without relying on the coupling constants description. The usual RG idea then asks us to find fixed-point tensor A^* of the tensor RG equation such that $A^* = A^*A^*$ and linearize the RG equation around this fixed-point tensor. We will know from previous analysis of the 1D Ising model that there is a high-temperature fixed point $g^* = \log(1/2), h^* = 0, K^* = 0$, corresponding to the fixed-point tensor,

$$A^* = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}. \quad (4.7)$$

Notice that the constant term g in the Hamiltonian becomes the overall multiplicative constant $\exp(g)$ of the tensor A in Eq. (4.3) and that the fixed-point g^* is adjusted to have a $1/2$ to make sure that we have $A^* = A^*A^*$, not just $A^* \propto A^*A^*$. It is

straightforward to find the linearized RG equation around A^* ,

$$\delta A_c = \delta A A^* + A^* \delta A = I \delta A A^* + A^* \delta A I, \quad (4.8)$$

where in the last equal sign, we add two identity matrices. We can find the expression of the linearized RG equation \mathcal{R} immediately by writing Eq. (4.8) in its component form, $(\delta A_c)_{ab} = \sum_{\alpha,\beta} I_{a\alpha} (\delta A)_{\alpha\beta} (A^*)_{\beta b} + (A^*)_{a\alpha} (\delta A)_{\alpha\beta} I_{\beta b}$, so we have

$$\mathcal{R}_{(ab)(\alpha\beta)} = \frac{(\delta A_c)_{ab}}{(\delta A)_{\alpha\beta}} = I_{a\alpha} (A^*)_{\beta b} + (A^*)_{a\alpha} I_{\beta b}, \quad (4.9)$$

where the index $(ab), (\alpha\beta)$ of \mathcal{R} means a group of two indices a, b and α, β , with the following order convention,

$$(11) \rightarrow 1, (12) \rightarrow 2, (21) \rightarrow 3, (22) \rightarrow 4. \quad (4.10)$$

The matrix \mathcal{R} takes the following value

$$\mathcal{R} = \begin{pmatrix} 1 & 1/2 & 1/2 & 0 \\ 1/2 & 1 & 0 & 1/2 \\ 1/2 & 0 & 1 & 1/2 \\ 0 & 1/2 & 1/2 & 1 \end{pmatrix}. \quad (4.11)$$

This matrix \mathcal{R} symmetric, with eigenvalues and eigenvectors: $\lambda_1 = 2, \mathbf{v}_1 = (1, 1, 1, 1)^T$; $\lambda_2 = 1, \mathbf{v}_2 = (1, 0, 0, -1)^T$; $\lambda_3 = 1, \mathbf{v}_3 = (0, 1, -1, 0)^T$ and $\lambda_4 = 0, \mathbf{v}_4 = (1, -1, -1, 1)^T$. The eigenvalues are the same as what we get in Eq. (2.14).

Notice how simple the canonical RG prescription is in tensor-network language! The main advantage lies in the simplicity of the tensor RG equation in Eq. (4.5). Since it is just a matrix multiplication, the linearization is straightforward. This advantage remains in higher dimensions, as can be seen in Eqs. (4.1) and (4.2) for the RG equation and the linearized version of the 2D HOTRG. By comparison, in the Hamiltonian approach, the linearized RG equation *has to* be expressed as a linear

map between old and coarser perturbation of the *coupling constants* around the fixed point. The nonlinear relation between the coupling constants and the Boltzmann weight makes the linearization process less trivial.

The relation between the tensor A and the coupling constants in Eq. (4.3b) reveals how the canonical RG prescription in tensor-network language is related to the conventional one in Hamiltonian-based language. We want to find the relation between δA and the coupling constants $\delta g, \delta h, \delta K$. To this end, plug $g = \log(1/2) + \delta g, h = \delta h, K = \delta K$ into the right hand side of Eq. (4.3b), Taylor expand, and only keep the linear terms,

$$\begin{aligned}
A_p &= A^* + \frac{1}{2}\delta g \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2}\delta h \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\
&+ \frac{1}{2}\delta K \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} + \text{higher-order terms} .
\end{aligned} \tag{4.12}$$

We can read off $\delta A = A_p - A^*$ as

$$\delta A = \frac{1}{2}\delta g \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \frac{1}{2}\delta h \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} + \frac{1}{2}\delta K \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix} . \tag{4.13}$$

Recall the order convention in Eq. (4.10), we see the correspondence $\mathbf{v}_1 \leftrightarrow \delta g, \mathbf{v}_2 \leftrightarrow \delta h$ and $\mathbf{v}_4 \leftrightarrow \delta K$. Compare these results with Eq. (2.14), we notice that the linearized RG equations in tensor-network and Hamiltonian-based language are indeed the same linear transformation expressed in different bases.

4.2.2 Formal argument

The decimation of the 1D Ising model in tensor-network language provides a prototype of the canonical RG prescription in tensor-network language and also demonstrates the relation of the canonical RG prescription in two languages. We are now ready to give a general definition of the *canonical RG prescription in tensor-network language*.

In the tensor-network RG approach, we skip the Hamiltonian description of the

system. Instead, we use a tensor network made of copies of tensor A to represent the partition function Z of the system. The tensor RG equation is a map from the tensor A to the coarser tensor A_c , like the processes that are shown in Eq. (3.5) and Fig. 4-1. We claim that the components of the tensor A can be thought of as some proxies of the coupling constants \mathbf{K} based on the previous 1D Ising model example (this claim was also hinted in Ref. [24]).

Definition 4.2.1 (Canonical RG prescription in tensor-network language). Given a d -dimensional tensor-network model consisting of copies of tensor A and a *legitimate* tensor-network RG equation $A_c = \mathcal{T}(A)$ with scaling factor b , the canonical RG prescription is the following procedure: 1) identify a fixed-point tensor A^* satisfying $A^* = \mathcal{T}(A^*)$; 2) linearize the tensor RG equation around A^* to have $(\delta A_c)_{(i)} = \sum_j \mathcal{R}_{(i)(j)} \delta A_{(j)}$; 3) extract the scaling dimension $\{x_\alpha\}$ from the eigenvalues $\{\lambda^\alpha\}$ of the matrix \mathcal{R} according to $b^{d-x_\alpha} = \lambda^\alpha$.

Remark 1. Besides keeping the change of the partition function as small as possible, a *legitimate* tensor-network RG equation should exhibit fixed-point tensors and have its gauge redundancy properly fixed. See Section 4.3 for the detailed discussions about this.

Remark 2. Since the overall multiplicative constant of the fixed-point tensor A^* does not affect physics, the fixed-point equation generally looks like $A^* \propto \mathcal{T}(A^*)$. However, it is always possible to adjust the multiplicative constant to make sure that the fixed-point tensor should satisfy $A^* = \mathcal{T}(A^*)$. See Eqs. (4.3) and (4.7) for the case of the 1D Ising model as a concrete example and Appendix 2 in Ref. [24] for a more detailed discussion.

Let us discuss why this definition is reasonable by working out its formal connection with the Hamiltonian-based approach. However, keep in mind that the canonical RG prescription does not rely on the Hamiltonian description and can be carried out on its own without thinking about coupling constants at all. To start, remember

that the partition function in Hamiltonian-based language can be mapped into a tensor-network model, as is shown in Section 3.1; each element of the initial tensor A is the probability weight of a given local spin configuration and depends on the coupling constants,

$$A_{(i)} = f_{(i)}(\mathbf{K}), \quad (4.14)$$

where all legs of A are group together to form a single index, $A_{(i)} \equiv A_{i_1 i_2 i_3 i_4}$. We perform the coarse graining according to a given RG equation. Each element of the coarser A_c are thus functions of \mathbf{K} but with different functional forms determined by the RG equation,

$$(A_c)_{(i)} = (f_c)_{(i)}(\mathbf{K}). \quad (4.15)$$

In the conventional Hamiltonian-based approach, we impose the condition that each element of the coarser tensor A_c should have the same functional form (usually the form of the Boltzmann weight) as that of A , but with different coupling constants \mathbf{K}' ,

$$f_{(i)}(\mathbf{K}') = (A_c)_{(i)} = (f_c)_{(i)}(\mathbf{K}), \forall(i). \quad (4.16)$$

If such \mathbf{K}' exists, the RG transformation will have both the tensor-network and the Hamiltonian representations. In tensor-network language, $A \xrightarrow{\text{RG}} A_c$, and in Hamiltonian-based language $K \xrightarrow{\text{RG}} K'$; the two languages are related to each other according to Eqs. (4.14) and (4.16). At the fixed point,

$$A^* \xrightarrow{\text{RG}} A^* \text{ or } A^* = \mathcal{T}(A^*). \quad (4.17)$$

Take the total derivative of tensors A and A_c in Eqs. (4.14) and (4.16) and set

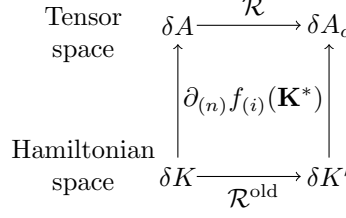


Figure 4-2: Relation between the two linearized RG equations in Hamiltonian and tensor-network languages. It is reminiscent of Fig. 1-1. Provided that a Hamiltonian description exists for a tensor-network coarse-graining scheme, such that $A_{(i)} = f_{(i)}(\mathbf{K})$, $(A_c)_{(i)} = f_{(i)}(\mathbf{K}')$, the linearized RG equation in tensor-network language \mathcal{R} and that in Hamiltonian-based language \mathcal{R}^{old} are the same linear transformation in two representations.

$$\mathbf{K} = \mathbf{K}' = \mathbf{K}^*,$$

$$\delta A_{(i)} = \sum_n (\partial_{(n)} f_{(i)}) \Big|_{\mathbf{K}=\mathbf{K}^*} \delta K_n, \quad (4.18)$$

$$(\delta A_c)_{(i)} = \sum_n (\partial_{(n)} f_{(i)}) \Big|_{\mathbf{K}'=\mathbf{K}^*} \delta K'_n. \quad (4.19)$$

The translation between the RG prescription in two languages is specified in Eqs. (4.18) and (4.19), with $\partial_{(n)} f_{(i)}$ evaluated at \mathbf{K}^* being the change of basis matrix. Under this transformation, the linearized RG equation in Eq. (2.18) becomes

$$(\delta A_c)_{(i)} = \sum_j \mathcal{R}_{(i)(j)} \delta A_{(j)}, \quad (4.20)$$

which is the linearized RG equation in tensor-network language. We have shown that the RG equations in two languages are two different representations of the same linear transformation, so they must have the same eigenvalue spectrum (see Fig. 4-2).

4.3 Two caveats

In the formal definition of the canonical RG prescription in tensor-network language in Section 4.2.2, the tensor-network RG equation should be legitimate. In this section, we explain the word “legitimate”. A legitimate tensor-network RG equation should exhibit fixed-point tensors satisfying Eq. (4.17) (or Eq. (4.1) for the HOTRG). There

are two major obstacles preventing us from achieving a legitimate tensor-network RG: local correlations and gauge redundancy in tensor-network language.

4.3.1 Local correlations

This feature was discovered at the birth of the tensor-network RG idea. When proposing the tensor renormalization group (TRG), the first realization of the tensor-network RG, Levin and Nave found there are *too many* fixed points for the RG equation of the TRG. Specifically, this means that two different initial tensors both belonging to the high-temperature phase will flow to two different fixed points, instead of one [37]. Levin and Nave’s claim was supported by the numerical evidence shown by Hinczewski and Berker [27], who also discovered that the critical fixed-point tensor seems to disappear at larger bond dimensions. The numerical evidence indicates that the TRG has difficulty filtering out microscopic details (like the temperature of the system), so physical at the original lattice scale is carried all the way to larger scales, which is not a desirable feature of a good RG transformation. The local correlations retaining in the tensor make identification of fixed-point tensors more difficult than the conventional Hamiltonian approach.

We can understand the origin of the problem of local correlations by reexamining what would happen at a later stage of a block-tensor transformation described in Eq. (3.6) and Fig. 3-2. For the convenience of reading, we show the physical picture of the block-tensor transformation again in Fig. 4-3. Let us say we start with a system at its high-temperature phase. After a step of the block-tensor transformation, 16 smaller black squares turn into 4 bigger squares, with more spin variables sitting on each side of each larger square. At a later stage of the RG iterations, the fixed-point tensor corresponds to a black square that is very large. To determine the structure of the fixed-point tensor, let us consider the following physical argument. The spin variables on different sides must be uncorrelated since they are very far away from each other, except for the spin variables around the four corners. Around a corner, the spin variables are always closed to each other and thus are correlated. The local correlations are represented by a matrix C in Fig. 4-3, encoding physical at the original

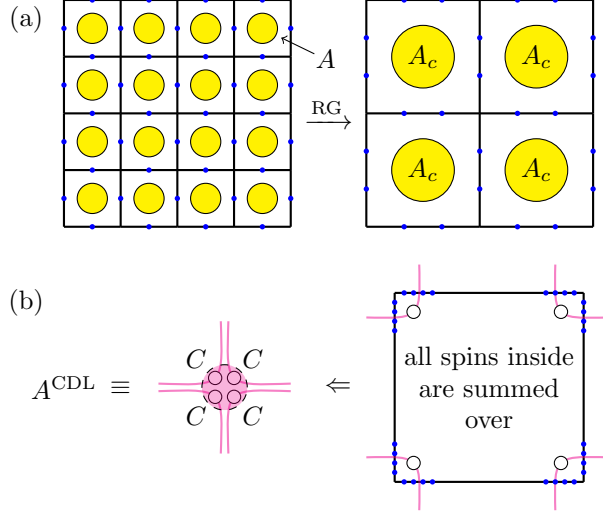


Figure 4-3: Physical original of the CDL structure. (a) The physical picture of a block-tensor transformation. The old tensor A is surrounded by four spin variables, while the coarser tensor A_c is surrounded by eight spin variables. (b) At a later stage of the block-tensor transformation, many original spin variables sit on the sides of the square, with all spins in the middle are traced out during the coarse graining. If the system is in its high-temperature phase, the spin variables on one edge must be uncorrelated with those on another edge, except for the spin variables around the four corners, which remain to be very close to each other. The correlations between spin variables around the corner can be described by a matrix C , the direct product of four copies of which gives rise to a CDL tensor A^{CDL} .

lattice scale. Any two groups of the spin variables on two different corners are far away, and thus are uncorrelated; the fixed-point tensor should be a direct product of four copies of C . We call a tensor with such structure a corner double-line (CDL) tensor.

It has been shown that a tensor with the CDL structure is an exact fixed-point tensor of the TRG [37, 24, 18, 26] and the HOTRG [54]. This fact indicates that the lattice-scale physics will be kept under the coarse graining according to the TRG or the HOTRG. As a concrete example, imagine we start with two initial tensors at temperature $T_1 > T_2 > T_c$. The local correlations around the corner are different for the two tensors, captured by two different matrices $C_1 \neq C_2$. Though the two tensors belong to the high-temperature phase, they eventually flow to two different CDL tensors, one consisting of C_1 and the other C_2 . Therefore, instead of a single high-temperature fixed point, the tensor RG flow generated by the TRG (or the

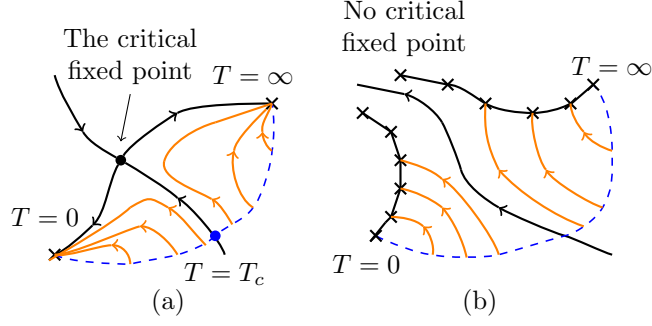


Figure 4-4: Schematic RG flows of the 2D Ising model, one is what we expect and the other is what the TRG (or any TRG-like scheme) generates. The starting points of the RG flows form the blue dashed line, representing the lattice model at different temperatures. RG flows are drawn using solid lines with arrows. (a) The tensor RG flows we expect for the 2D Ising model. There are three fixed points in the tensor space, one high-temperature fixed point indicated by $T = \infty$, one low-temperature fixed point indicated by $T = 0$, and one critical fixed point $T = T_c$. (b) The tensor RG flows the TRG (or any TRG-like scheme) generate. Since the local correlations at the original lattice scale are carried all the way to larger scales, different starting points, even belonging to the same phase, will flow to different fixed points with the CDL structure. The critical fixed point might disappear at bond dimensions $\chi > 8$, as is indicated in the numerical evidence by Hinczewski and Berker [27].

HOTRG) will exhibit a fixed line shown in Fig. 4-4(b). At criticality, Hinczewski and Berker’s calculations have shown that the critical fixed point disappears at bond dimensions $\chi > 8$ for the TRG. By comparison, Fig. 4-4(a) shows the RG flow we expect. We will introduce a way to collapse all the tensors with the CDL structure to a single tensor with bond dimension $\chi = 1$ for the HOTRG in Chapter 5.

4.3.2 Gauge redundancy

The second obstacle that makes the identification of the fixed-point tensors difficult is the gauge redundancy of the tensor-network representation of a partition function. The partition function is invariant under a certain transformation of the local tensor, gauge transformation. For example, for the tensor-network representation of the partition function in Fig. 3-1, we are free to transform the local tensor A in the following way,

$$\tilde{A}_{ijkl} = \sum_{\substack{m,n \\ p,q}} A_{mnpq} (S_x)_{im} (S_y)_{jn} (S_x^{-1})_{pk} (S_y^{-1})_{ql}, \quad (4.21a)$$

or pictorially as

(4.21b)

where S_x, S_y are some invertible matrices. It is easy to see that such transformation leaves the partition function invariant. We can treat Eq. (4.21) as a equivalence relation that defines a equivalence class $[A]$.

In fact, the tensor-network RG techniques feature the exploiting of this gauge redundancy of tensor-network representations. For the HOTRG described in Section 3.2.2, we use eigenvectors defined in Eq. (3.14) as a set of good basis vectors, since their importance is reflected by the corresponding eigenvalues. Several unimportant basis vectors are thrown away and a change of horizontal basis of a patch of two copies of A is performed in Eq. (3.10) using w^\dagger as S_x to define the coarse graining in the vertical direction.

However, the gauge redundancy makes the identification of a fixed point more difficult than the conventional Hamiltonian-based approach. The tensor RG equation can take the fixed-point tensor in the equivalence class $[A^*]$ from one representation to another representation,

$$A^* \xrightarrow[\text{graining}]{\text{coarse}} \tilde{A}^*. \quad (4.22)$$

A legitimate tensor-network RG equation should have its gauge redundancy fixed by choosing a preferred set of basis to achieve a manifestly fixed-point tensor satisfying $A^* \xrightarrow{\text{RG}} A^*$. We will show a way to fix the gauge redundancy in Chapter 5.

Chapter 5

A Numerical Realization

In this chapter, we propose a practical numerical realization of the canonical RG prescription in tensor-network language defined in Chapter 4, using the HOTRG as the tensor coarse-graining scheme. The key for a numerical realization is to deal with the two caveats, local correlations and gauge redundancy, discussed in Section 4.3, for the HOTRG.

5.1 Filtering out local correlations: a HOTRG-like scheme

We use a technique called the graph-independent local truncation (GILT) [26] to solve the problem of local correlations for the HOTRG. It should be mentioned that there are many other techniques to filter out the local correlations [24, 18, 16, 3, 65, 25, 20, 66, 35]. We choose to combine the HOTRG with the GILT in this thesis because of its conceptual simplicity, ease in implementation, and more importantly, the convenience for the further generalization to 3D systems.

Detect the local correlations in a tensor network containing loops The defining feature of the GILT is that it can in principle be applied to any tensor-network coarse-graining scheme, making it very flexible. The first coarse-graining pattern of the GILT is the TRG, with their collaboration designed in the original GILT paper [26].

People have shown that this combination is capable of generating correct tensor RG flows in the context of the 2D Ising model [26] and the 2D ϕ^4 theory [13]. The GILT detects the local correlations in a tensor A by looking at the property of a loop-like tensor network consisting of copies of A . For example, a possibility is a plaquette consisting of four copies of A shown in Fig. 5-1, where the C matrix is drawn explicit for a clear demonstration, and it is in fact unknown inner structure of the tensor A . When four copies of A forming a plaquette, copies of C matrix hold their hands to form a loop, representing a single number. We know, a priori, that if we know how to determine this number represented by this loop, we can distribute this number to tensor A as an overall multiplication constant; the resultant tensor will have a smaller bond dimension. The GILT provides a way to simplify this loop redundancy form by four copies of C . The idea is to insert a low-rank matrix Q into a bond we wish to truncate. The plaquette after the insertion should give a good approximation of the initial one. The low-rank matrix Q is constructed in such a way that its singular value decomposition cuts the legs of matrix C . We can repeat the procedure on other three legs and filter out the local correlations inside this plaquette completely. However, since in practice, every insertion of a low-rank matrix Q comes with an approximation error, we should only perform the necessary truncations of legs. To this end, let us discuss why the HOTRG has difficulty filtering out local correlations.

Design GILTs for the HOTRG Let us study how the CDL tensors in Fig. 4-3 transform under the HOTRG. The corner matrix C can be assumed to be symmetric considering its physical origin. It can be further put into its diagonal form. In order to find the isometric tensor for the CDL tensors, we refer to Eqs. (3.10), (3.12) and (3.13b), which says the isometry is a collection of the eigenvectors of

$$\begin{array}{c} \text{---} \square M \square \text{---} \\ \text{---} \square M^\dagger \square \text{---} \end{array} \propto \begin{array}{c} \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \\ \text{---} \circ \text{---} \end{array} \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} C \begin{array}{c} \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} . \quad (5.1)$$

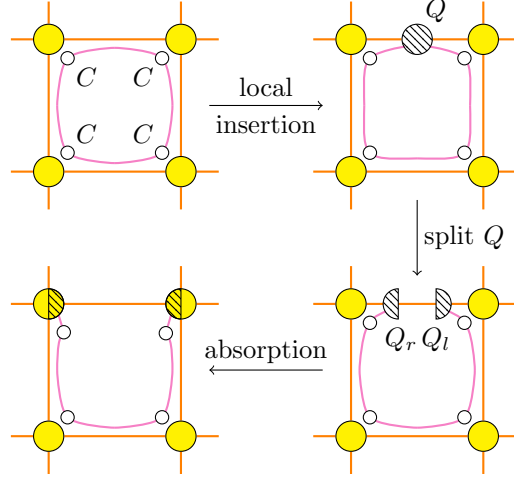


Figure 5-1: Truncation a single bond of the plaquette consisting of tensor A using the GILT. Four copies of C matrices are unknown inner structure of 4-leg tensor forming the plaquette. They are drawn explicitly to make the demonstration clearer. In the first step, a low-rank matrix Q is inserted into a bond. The splitting of the Q into two pieces using singular value decomposition cuts the leg of corner matrix C . Finally, two pieces of the matrix Q are absorbed into the two neighboring 4-leg tensors. The original GILT paper [26] presents a way to determine the low-rank matrix Q . A brief introduction is provided in Appendix B.

Let us say C is a $\sqrt{\chi} \times \sqrt{\chi}$ matrix, so MM^\dagger is a $\chi^2 \times \chi^2$ matrix. Equation (5.1) shows the rank of MM^\dagger is χ . The eigenvectors of this MM^\dagger are easy to write down. Since C is already in its diagonal form, we do nothing about the two outer matrices, $\text{---} \circ \text{---}$ and $\text{---} \circ \text{---}$. The inner matrix, $\text{---} \circ \text{---}$, is the outer product of two χ -dimensional vectors, both constructed from two C matrices, so the inner matrix has rank 1. The only eigenvector of the inner matrix with non-zero eigenvalues is the vector in the outer product, properly normalized,

$$\text{---} \circ \text{---} \equiv \text{---} \circ \text{---} / \sqrt{\text{---} \circ \text{---}}. \quad (5.2)$$

Therefore, the isometry w for the HOTRG coarse graining of two A^{CDL} tensors in the vertical direction is

$$\text{---} \circ \text{---} \equiv \text{---} \circ \text{---} \cdot w. \quad (5.3)$$

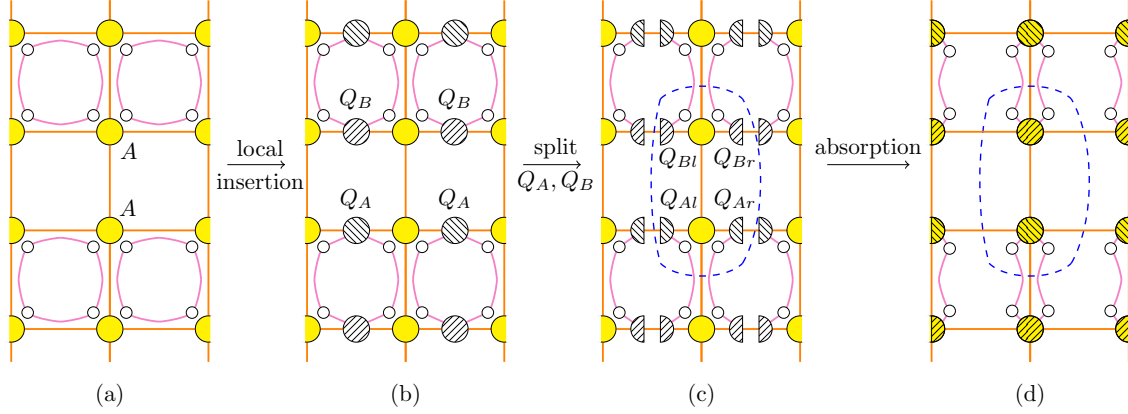


Figure 5-2: How to choose the plaquette and where to insert the low-rank matrices to filter out the problematic local correlations for the HOTRG.

Use this isometry to squeeze the legs of A^{CDL} in the manner as is shown in Eq. (3.10), we see that A^{CDL} is a fixed-point tensor of the HOTRG,

$$\text{Diagram} \propto \text{Diagram} \quad (5.4)$$

Equations (5.3) and (5.4) tells us that the isometric tensor of the HOTRG can only detect and trace out the inner copies of C matrices, but cannot filter out the outer copies. Therefore, we should use the GILT to truncate the legs of outer copies of the matrix C . Let say we want to block the two copies of A indicated in Fig. (5-2)(a). Based on the reason explained before, we should choose the plaquettes where the loops of local correlations are drawn explicitly as the target of the GILT. Two low-rank matrices Q_A, Q_B are then inserted into the bonds to truncate the legs of unwanted C matrices. Finally, the ordinary HOTRG is applied to the patch of tensors in the dashed circle in Fig. 5-2. The coarse graining in the vertical direction becomes

$$\text{Diagram} \equiv \text{Diagram} \quad (5.5)$$

It is clear that the coarse graining defined in Eq. (5.5) can filter out all four horizontal legs of copies of C matrix; two of four are removed by the GILT, and other two are traced out by the HOTRG isometric tensor. Repeat this process in the horizontal direction to achieve the full coarse graining (we refrain from calling it an RG equation because the gauge redundancy has not been taken care of yet),

$$(5.6)$$

Notice if the tensors coming from the GILT are absorbed into the HOTRG isometric tensors, the coarse graining has the same form as the HOTRG in Eq. (3.11), except a modification of the triangular tensors. Since the computation costs of the GILT are $O(\chi^6)$, smaller than $O(\chi^7)$ for the HOTRG, the overall computational costs grow like $O(\chi^7)$.

Since the RG transformation defined in Eq. (5.6) will simplify a CDL tensor to a single number,

$$(5.7)$$

the problem of local correlations is solved in the proposed HOTRG-like scheme. The fixed line represented by tensors with the CDL structure shown in Fig. 4-4(b) will collapse into a single fixed point.

5.2 Gauge fixing

In this section, we propose a way to fix the gauge redundancy and point out a special property of the HOTRG-like scheme proposed in the previous section, reducing the gauge redundancy to sign ambiguities for real tensors with spatial reflection symmetries.

5.2.1 A general proposal

If the physical model in hand has a global internal symmetry, the symmetry can be exploited to fix part of the gauge redundancy, because the global symmetry can be imposed into the tensor-network representation of the model and the symmetry can be preserved exactly during a tensor-network RG transformation. For example, if \mathbb{Z}_2 symmetry of the 2D Ising model is imposed, half of the gauge redundancy will be fixed, since we are always in the basis where states in the even sector transform trivially, while those in the odd sector is multiplied by -1 under the spin-flip transformation.

The remaining gauge (except for sign or phase ambiguities) in the degenerate sectors of A can be fixed by the following procedure. For the simplicity of the demonstration, we explicitly show the procedure only in the horizontal direction; the vertical direction can be done in the same manner. We first contract two vertical legs of A to get a matrix N_x ,

$$\begin{array}{c} \text{---} \end{array} \textcircled{N_x} \text{---} = \begin{array}{c} \text{---} \end{array} \textcircled{A} \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{---} \quad (5.8)$$

We then find its eigenvalue decomposition,

$$\begin{array}{c} \text{---} \end{array} \textcircled{N_x} \text{---} = \begin{array}{c} \text{---} \end{array} \text{---} \text{---} \text{---} \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{---} \quad (5.9)$$

where λ is the diagonal matrix encoding eigenvalues. To fix the gauge redundancy in the horizontal direction, we act the invertible matrix W_x and its inverse on the

horizontal legs of the tensor A ,

$$\text{Diagram showing the horizontal gauge fixing of a tensor } A. \quad (5.10)$$

In Appendix B, we show why this procedure defines a preferred set of basis.

5.2.2 A special property of the proposed HOTRG-like scheme

It turns out that the general gauge fixing procedure is not necessary for the HOTRG-like scheme if the physical system under consideration has reflection symmetries like the 2D Ising model. The scheme itself has its own preferred set of basis, collapsing the general gauge redundancy into sign ambiguities (or phase ambiguities for tensors with complex components). We focus on real tensors in this section, but the generalization to complex tensors is straightforward.

We will show in Appendix B that, if two real tensors A, \tilde{A} connected with each other by the gauge transformation defined in Eq. (4.21), where the S_x, S_y is further restricted to be orthogonal matrices, the coarser tensors generated the HOTRG-like scheme, A_c and \tilde{A}_c , are equal up to sign ambiguities,

$$\left(\tilde{A}_c\right)_{ijkl} = (A_c)_{ijkl} (d_x)_i (d_y)_j (d_x)_k (d_y)_l, \quad (5.11)$$

where d_x, d_y are vectors with components ± 1 . Let us assume that the sign ambiguities have already been taken care of, the tensor RG equation ensures

$$\mathcal{T}(A) = \mathcal{T}(\tilde{A}). \quad (5.12)$$

Equation (5.12) says that the whole equivalence class $[A]$ will be mapped into the same coarser tensor A_c after the coarse graining! Apply this special property to the equivalence class of a fixed-point tensor $[A^*]$, we realized that we can start with an arbitrary representation \tilde{A}^* , the tensor RG equation will bring it to the unique basis

automatically to have $A^* = \mathcal{T}(\tilde{A}^*)$, which is a manifestly-fixed-point tensor,

$$\mathcal{T}(\tilde{A}^*) = \mathcal{T}\left(\mathcal{T}(\tilde{A}^*)\right) \equiv A^*. \quad (5.13)$$

5.2.3 Sign ambiguities

The sign ambiguities between two tensors can be fixed by comparing the sign of their non-vanishing components. As an simple example, upon making sure that $(A_c)_{i111} \neq 0, \forall i$, examine these components in Eq. (5.11) to have

$$(\tilde{A}_c)_{i111} = (A_c)_{i111}(d_x)_i. \quad (5.14)$$

The sign difference of two tensors determines $(d_x)_i$. For a symmetric tensor, we can the above trick in each degenerate sector of the tensor. We provide an implementation of the sign fixing for \mathbb{Z}_2 symmetric tensors in the source code ¹.

5.3 Overall procedure for the numerical realization

Let us pause at this point to summarize the proposed numerical realization of the canonical RG prescription in tensor-network language. For simplicity, we only discuss real tensors here.

Step 1: Determine pieces of low-rank matrices in the GILTs according to Fig. 5-2 to filter out the local correlations for the subsequent HOTRG.

Step 2: Apply the HOTRG in the vertical direction to the patch of tensors shown in Fig. 5-2.

Step 3: Repeat the first steps in the horizontal directions to obtain the coarser tensor A_c . The pictorial representation of the coarse graining from A to A_c is shown in Eq. (5.6).

¹The source code of this paper can be found at github.com/brucelyu/tensorRGflow.

Step 4: Fixed the sign ambiguities of A_c by comparing it with the input tensor A according to Eq. (5.14); notice that A_c and A should be properly normalized. The end result is to apply d_x, d_y on horizontal and vertical legs of A_c .

Step 5: Define the modified tensor \tilde{w}, \tilde{v} by absorbing into w, v pieces of low-rank matrices coming from the GILTs in the right hand side of Eq. (5.6) and sign fixing diagonal matrices d_x, d_y . As an example, \tilde{w} is define as

$$\tilde{w} = \begin{matrix} Q_{Br} \\ Q_{Ar} \end{matrix} w d_x \quad (5.15)$$

After this modification, the linearized RG equation is Eq. (4.2) where all the isometric tensors are replaced with their tilde version.

Step 6: Diagonalize the linearized RG equation to obtain critical exponents (or scaling dimensions).

5.4 Benchmark results

In this section, we use the two-dimensional (2D) classical Ising model to demonstrate how to carry out the canonical RG prescription in tensor-network language using the proposed HOTRG-like scheme. The source code can be used to reproduce the numerical analysis below ².

5.4.1 Analysis of the RG flow

Pull out the tensor norm after each RG iteration The partition function under consideration is given in Eq. (3.1), with its tensor-network representation in Fig. 3-1. For convenience, the initial tensor in Eq. (3.2) is denoted as $A^{(0)}$ here. During the RG iteration, the magnitude of the tensor will in general grow (or decay) rapidly, causing trouble numerically; this concern can be taken care of by pulling out the Frobenius

²The source code of this paper can be found at github.com/brucelyu/tensorRGflow.

norm of the tensor before feeding it into the tensor RG equation. For example, at the first iteration, we perform $A^{(0)} = \|A^{(0)}\| \mathcal{A}^{(0)}$, to define a normalized tensor $\mathcal{A}^{(0)}$, which will be fed into the RG equation in Eq. (5.6). The subsequent iteration is similar. At the n -th iteration, the output is $A^{(n)} = \|A^{(n)}\| \mathcal{A}^{(n)}$. The RG flow of $\|A^{(n)}\|$ can be easily visualized.

The RG flows of the tensor norm We use the RG flows of $\|A^{(n)}\|$ to demonstrate the proposed HOTRG-like scheme generates what we expect, as is shown in Fig. 4-4(a). Several RG flows starting at temperatures close to the critical temperature are shown in Fig. 5-3. Let us first discuss RG flows generated by the proposed HOTRG-like scheme, shown in Fig. 5-3(a), at bond dimension $\chi = 30$ and the hyper-parameter of the GILT $\epsilon_{\text{gilt}} = 6 \times 10^{-6}$. Different curves are generated at different temperatures deviations from the estimated critical temperature $T_c^{[\chi=30]}$. For example, the $\Delta T = +10^{-3}$ curve is the RG flow starting from the initial tensor at temperature $T_c^{[30]} + 10^{-3}$. As the deviation $|\Delta T|$ from the estimated critical temperature becomes smaller and smaller, the RG flow tends to stay longer near the critical fixed-point tensor, before flowing away to two trivial fixed points. The estimated critical temperature can be determined to very high precision using the RG flow of $\|A^{(n)}\|$. The relative difference of the estimation from the exact value, $|T_c^{[30]} - T_c|$, is of order 10^{-6} .

By comparison, we also show the RG flow generated by the HOTRG at bond dimension $\chi = 12$ in Fig. 5-3(b)³. As has been shown in the schematic RG flow in Fig. 4-4(b), the RG flow generated by the HOTRG has lots of trivial fixed points, and the critical fixed-point tensor does not appear.

The RG flows of the singular values of the normalized tensor The RG flow of a single indicator such as $\|A^{(n)}\|$ gives us evidence that we have reached a critical fixed-point tensor. More convincing it would be if more information about the tensor

³In principle, we can choose $\chi = 30$ here. In practice, however, our calculations show that the problem of local correlations in the HOTRG becomes worse at larger bond dimensions (see Ref. [27] for a similar observation in the context of the TRG) and that $\chi = 12$ is enough to demonstrate this problem.

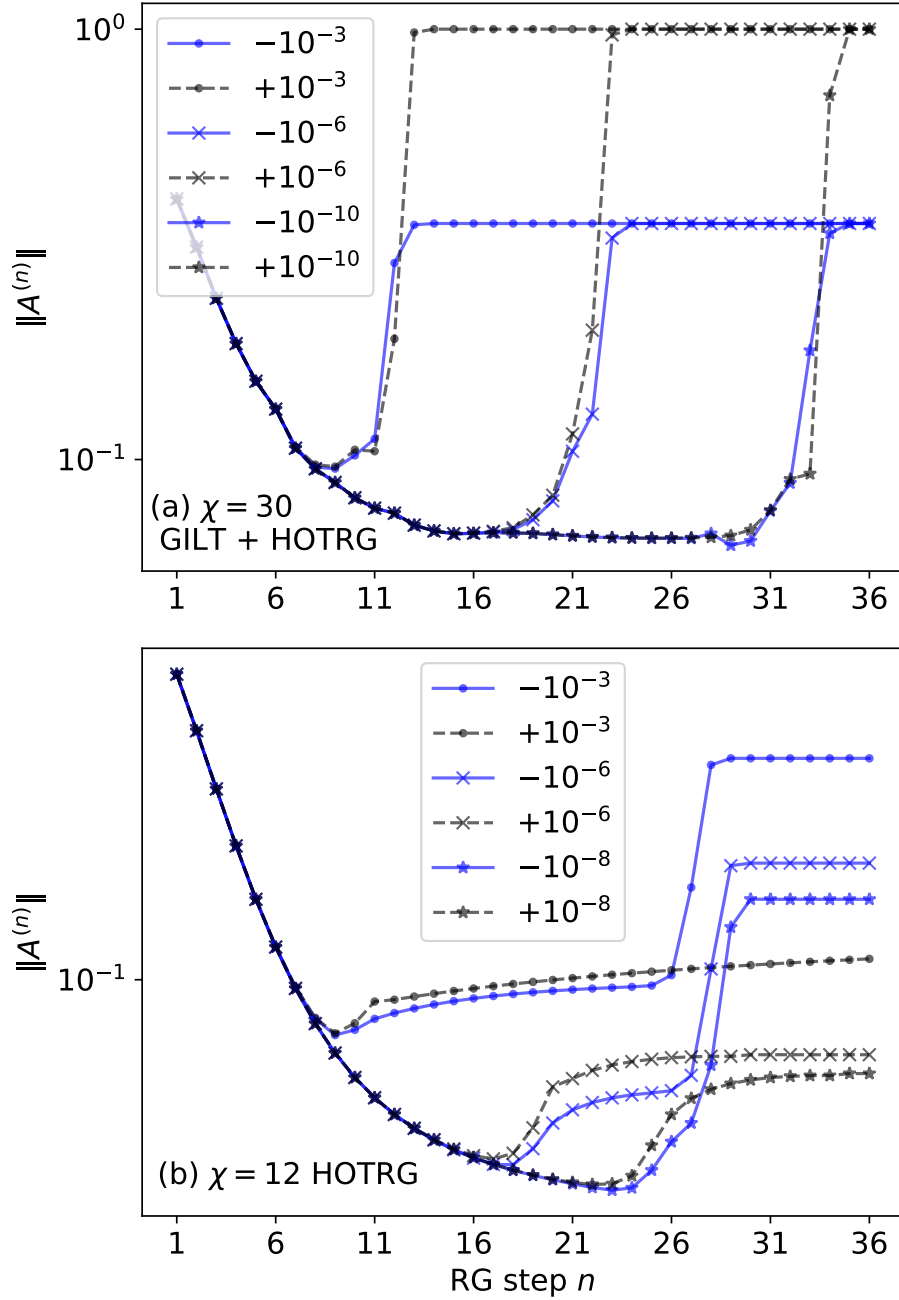


Figure 5-3: The RG flows of the tensor norms $\|A^{(n)}\|$ at temperatures near the estimated critical temperature $T_c^{[\chi]}$. Different markers represent different deviations $|\Delta T|$ from $T_c^{[\chi]}$. Blue solid lines are for $\Delta T < 0$ and black dashed lines for $\Delta T > 0$. (a) For the proposed HOTRG-like scheme with $\chi = 30$, $\epsilon_{\text{gilt}} = 6 \times 10^{-6}$, two trivial fixed points are isolated and the critical fixed point can be reached. It corresponds to the schematic RG flows in Fig. 4-4(a). (b) For the plain HOTRG with $\chi = 12$, we have fixed lines and there is no exhibition of a critical fixed point. It corresponds to the schematic RG flows in Fig. 4-4(b).

RG flows is in hand. One choice is the singular values $s^{(n)}$ of tensors $\mathcal{A}^{(n)}$ defined as

$$\mathcal{A}^{(n)} \stackrel{\text{svd}}{=} U s^{(n)} V^\dagger, \quad (5.16)$$

which is another gauge-independent indicator of the tensor RG flow apart from the Frobenius norm. Fig. 5-4(a) gives us more confidence that a critical fix-point tensor is reached.

After adding a sign-fixing step, we can achieve a manifestly-fixed-point tensor A^* . The difference between the normalized tensor at adjacent RG steps $\|\mathcal{A}^{(n+1)} - \mathcal{A}^{(n)}\|$ is shown in Fig. 5-4(b). From RG step $n = 14$, the difference starts to decay, reaches its minimal value of order 10^{-2} at $n = 23$, and increases when the tensor flows away from this unstable fixed point. If the sign is not fixed, as is shown in Fig. 5-4(c), the RG transformation will scatter the tensor among different representations of the fixed-point equivalence class.

5.4.2 Scaling dimensions

The linearized tensor RG equation \mathcal{R} with a form similar to Eq. (4.2) can be generated conveniently using the automatic differentiation implemented in JAX [5]. We linearize the RG equation at RG steps $n = 14, 15, \dots, 28$, and extract scaling dimensions $\{x_\alpha\}$ from the eigenvalues $\{\lambda^\alpha\}$ of \mathcal{R} according to $2^{2-x_\alpha} = \lambda^\alpha$. The first few smallest scaling dimensions are shown in Fig. 5-5, where the exact values [14] are indicated using dashed lines. We get correct estimations of scaling dimensions up to 2.125. The estimations at RG step $n = 14$ and 28 are unreliable since $\|\mathcal{A}^{(n+1)} - \mathcal{A}^{(n)}\|$ is of order 1 (see Fig. 5-4(b)). Comparison between scaling dimensions in Fig. 5-5 and difference of adjacent normalized tensors in Fig. 5-4(b) indicates that the canonical RG prescription in tensor-network language gives reliable estimations of scaling dimensions as long as the values of $\|\mathcal{A}^{(n+1)} - \mathcal{A}^{(n)}\|$ have order of or smaller than 10^{-1} .

The estimations of all scaling dimensions ≤ 2 at RG step $n = 22$ are shown in Table 5.1. The results obtained by the bread-and-butter Gu and Wen's method [24]

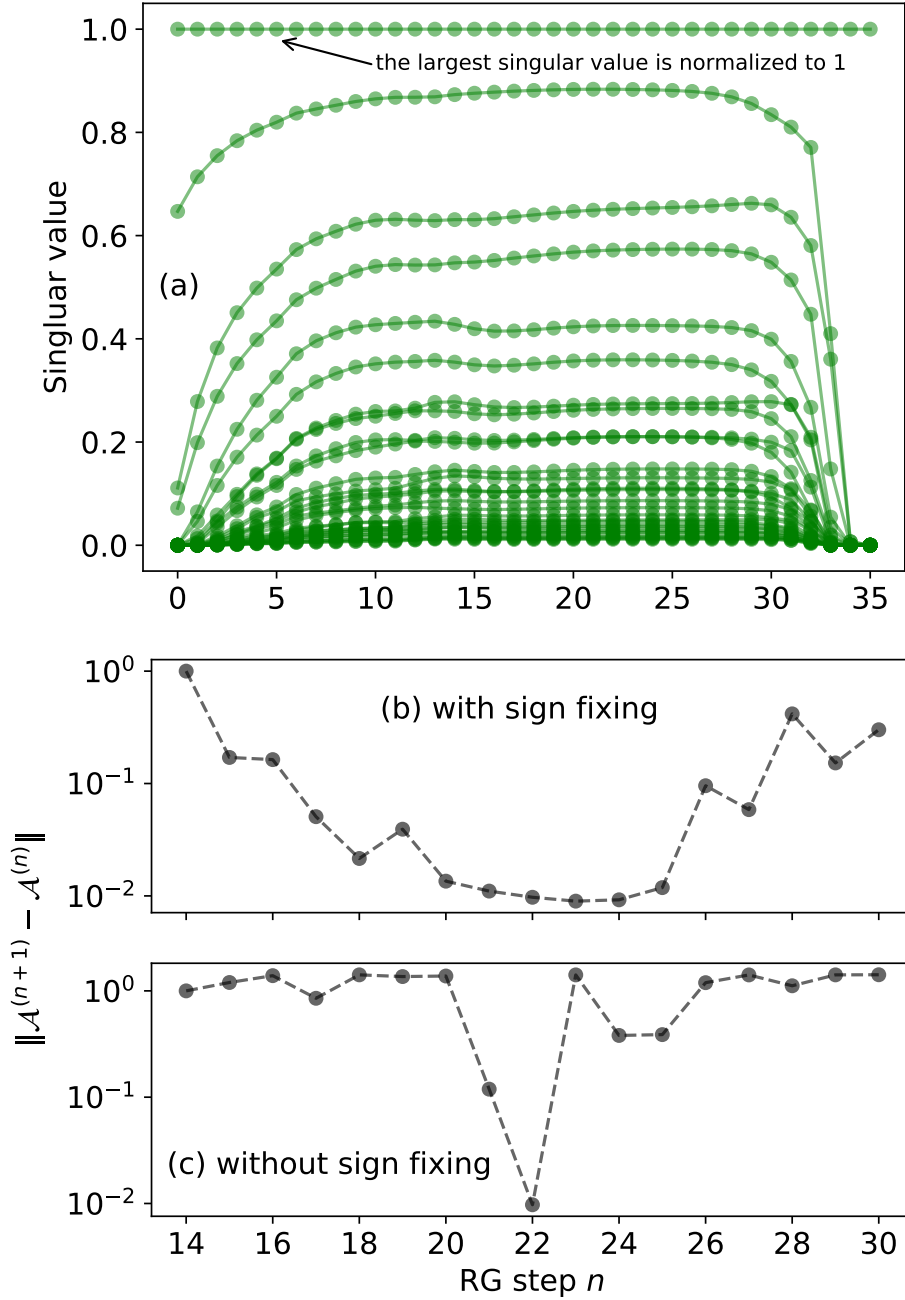


Figure 5-4: The RG flows of (a) singular values defined in Eq. (5.16) and (b) the difference between the normalized tensors, $\|\mathcal{A}^{(n+1)} - \mathcal{A}^{(n)}\|$ with sign fixing and (c) without, all at the estimated critical temperature $T_c^{[30]}$, generated by the proposed HOTRG-like scheme with $\chi = 30$, $\epsilon_{\text{gilt}} = 6 \times 10^{-6}$.

are also shown (the transfer matrix is constructed using two copies of the critical fixed-point tensor). The canonical RG prescription estimates scaling dimensions with similar accuracy as the bread-and-butter method.

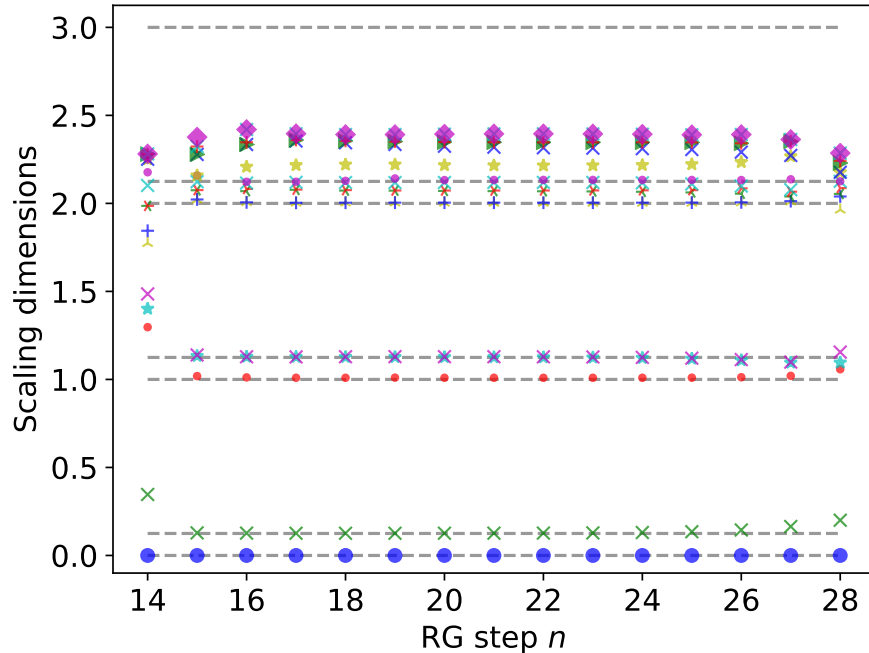


Figure 5-5: The scaling dimensions of the 2D Ising model from the canonical RG prescription using the proposed HOTRG-like scheme with $\chi = 30$, $\epsilon_{\text{gilt}} = 6 \times 10^{-6}$. Dashed lines are the exact values.

5.4.3 Remarks

We end this chapter with a few remarks on the above calculations. Firstly the \mathbb{Z}_2 is preserved exactly in the RG transformation by imposing this symmetry on the tensors [51, 52]. There are three reasons. The first consideration is to prevent the generations of any \mathbb{Z}_2 -odd fields. Otherwise, the low-temperature fixed point becomes unstable and tends to flow to the high-temperature fixed point eventually due to numerical errors. The second reason is to fix part of the gauge redundancy (see Sec. 5.2). The third reason is to make the computation costs lower since the effective bond dimension becomes smaller when symmetric tensors are employed.

The second remark is the systematic improvement of the estimations as we increase the bond dimension. At a fixed bond dimension χ , the hyper-parameter of the GILT ϵ_{gilt} can be tuned in the following way. Since the GILT process gives rise to additional approximation errors, ϵ_{gilt} should be chosen as small as possible since it controls the approximation errors of the GILT. However, it cannot be too small,

Table 5.1: The scaling dimensions for the relevant and marginal operators of the 2D Ising model at criticality from the canonical RG prescription and from the transfer matrix method à la Gu and Wen [24], both using the proposed HOTRG-like scheme with $\chi = 30, \epsilon_{\text{gilt}} = 6 \times 10^{-6}$ at RG step $n = 22$.

Exact	0.125	1	1.125	1.125	2	2	2	2
RG pres.	0.127	1.009	1.125	1.128	2.002	2.004	2.068	2.073
Trans. mat.	0.125	1.002	1.128	1.128	2.014	2.014	2.016	2.016

or otherwise the problem of local correlations arises and there would be no critical fixed point. In our calculations, the best ϵ_{gilt} at bond dimensions $\chi = 10, 20, 30$ are $6 \times 10^{-4}, 6 \times 10^{-5}, 6 \times 10^{-6}$. The estimated scaling dimensions converge to the exact results in this process.

The final remark is that the method proposed in this chapter is only one possible numerical realization of the canonical RG prescription in tensor-network language defined in Chapter 4. The purpose is to show that the convention RG prescription really works in this modern RG idea and to prepare for its further applications to 3D systems, where the bread-and-butter method is inapplicable.

Chapter 6

Summary and Discussions

In this thesis, we show the canonical RG prescription works in modern tensor-network language. Pedagogically, it provides a new way to introduce the RG idea. The intuitive aspect of the tensor-network RG as a real-space RG makes the physical picture easier to grasp than the momentum-space RG implementation. Besides, the estimated scaling dimensions, as physical observables that can be compared directly with experiment results, can be improved systematically by increasing the bond dimension.

Compared with the state-of-art methods to extract scaling dimensions in tensor-network RG ideas, the canonical RG prescription is from a pure RG perspective and does not rely on any nontrivial RG arguments explicitly. This makes the RG prescription more advantageous in 3D systems. In future work, we plan to generalize the HOTRG-like scheme proposed in this thesis to 3D and use the canonical RG prescription to extract scaling dimensions of 3D critical systems, where there are few practical tensor-network-based methods ¹ to extract scaling dimensions efficiently. It would also be very interesting to study how to implement the canonical RG prescription in the context of other 3D tensor-network RG techniques with lower computational costs than the HOTRG, for example, the anisotropic tensor renormalization group [1]

¹From the perspective of a real-space RG transformation for quantum systems [56], the scale-invariant multiscale entanglement renormalization ansatz (MERA) [57] can be used to build a scaling superoperator [48]. The scaling dimensions are obtained from the eigenvalues of the scaling superoperator [22]. However, the computation costs of the MERA for 2+1D quantum systems grow as $O(\chi^{16})$ [17], much higher than $O(\chi^{11})$ for the 3D HOTRG.

or renormalization group on a triad network [\[31\]](#).

Appendix A

Optimization forms of eigenvalue problems

In Sec. 3.2.2, it is claimed that the optimization problem in Eq. (3.13) can be converted into an eigenvalue problem. This is a well-known result. We will provide a short proof here to make this thesis self-contained.

Given a Hermitian positive semi-definite matrix H , find the optimal matrix w such that $\text{Tr}(w^\dagger H w)$ is maximized, provided that the matrix w satisfies $w^\dagger w = \mathbb{1}$. This constraint means that $(w^\dagger w)_{ij} = \sum_\alpha w_{\alpha i}^* w_{\alpha j}$ is 0 for $i \neq j$ and is 1 for $i = j$. The first condition says that the column vectors of w are orthogonal to each other, and the second says that they are normalized. We first impose the normalization constraint and construct the Lagrangian,

$$\mathcal{L} = \sum_{\alpha\beta i} H_{\alpha\beta} w_{\alpha i}^* w_{\beta i} - \sum_i \lambda_i \left(\sum_\alpha w_{\alpha i}^* w_{\alpha i} - 1 \right), \quad (\text{A.1})$$

where λ_i are the Lagrange multipliers. Set the derivative of \mathcal{L} with respect to $w_{\alpha i}^*$ to zero, we get the following eigenvalue problem for H ,

$$\sum_\beta H_{\alpha\beta} w_{\beta i} = \lambda_i w_{\alpha i}, \quad (\text{A.2})$$

which means the columns of w are the eigenvectors of H . Then, the quantity that we

want to maximize becomes $\text{Tr}(w^\dagger H w) = \sum_i \lambda_i$. Finally, we impose the orthogonality condition of w , so the columns of w should be the linearly independent eigenvectors of H corresponding to the first several largest eigenvalues. This completes our proof.

Appendix B

Proof regarding gauge fixing

For the general gauge fixing scheme Let first show the gauge fixing procedure in Eqs. (5.8) to (5.10) can collapse the gauge freedom to phase ambiguities. The goal is to show that if two tensors A, \tilde{A} are fed into the procedure, the general gauge transformation in horizontal legs becomes a sign ambiguity. After the contraction of two vertical legs, the two matrices N_x, \tilde{N}_x have the following relation

$$\text{---} \circlearrowleft \tilde{N}_x \text{---} = \text{---} \boxed{S_x} \circlearrowleft N_x \boxed{S_x^{-1}} \text{---}. \quad (\text{B.1})$$

Assume that the eigenvalues of N_x are all distinct, the matrix W_x in the eigenvalue decomposition (see Eq. (5.9)) is related to \tilde{W}_x through

$$\text{---} \triangleleft \tilde{W}_x \text{---} = \text{---} \boxed{S_x} \triangleleft W_x \text{---} \circ d_x, \quad (\text{B.2})$$

where d_x coming from phase ambiguities of eigenvectors has its diagonal entries to be pure phases. After the horizontal gauge fixing, the tensor \tilde{A} becomes

$$\text{---} \circlearrowleft \tilde{A} \text{---} \xrightarrow[\text{gauge fixing}]{\text{horizontal}} \text{---} \circ d_x \triangleleft W_x^{-1} \boxed{S_y} \circlearrowleft A \triangleleft W_x \text{---} \circ d_x \text{---}. \quad (\text{B.3})$$

Comparison between Eq. (5.10) and Eq. (B.3) shows the gauge freedom S_x in horizontal direction now collapse into phase ambiguities d_x .

For the special property of the proposed HOTRG-like scheme Next, we show the special property of the proposed HOTRG scheme in Eq. (5.11). For simplicity of demonstration, we focus on real tensors; the generalization to general complex tensors is straightforward. The S_x, S_y matrices are further restricted to be orthogonal,

The diagram shows a central yellow circle labeled \tilde{A} with four orange lines extending from its center. This is equated to a central yellow circle labeled A with four orange lines extending from its center. The top and bottom lines of A are enclosed in pink rounded rectangles labeled S_y and S_y^T respectively. The left and right lines of A are enclosed in pink rounded rectangles labeled S_x and S_x^T respectively.

$$\tilde{A} = S_x A S_x^T S_y S_y^T. \quad (\text{B.4})$$

Such orthogonal matrices are sufficient if the representations of the equivalence class $[A]$ have spatial reflection symmetries [16],

$$A_{kjil} = \sum_{j'l'} (O_y)_{jj'} (O_y)_{ll'} A_{i'j'k'l'} \quad (\text{B.5a})$$

and

$$A_{ilkj} = \sum_{i'k'} (O_x)_{ii'} (O_x)_{kk'} A_{i'j'k'l}, \quad (\text{B.5b})$$

where O_x, O_y are orthogonal matrices, also with $O_x^2 = O_y^2 = \mathbb{1}$, and the legs' order convention is as per Eq. (3.2). The tensor has such symmetries if the physical system under consideration has spatial reflection symmetries, and the tensor-network RG scheme preserves such symmetries, which is true for the proposed HOTRG-like scheme¹ applied to the 2D Ising model. Furthermore, in the preferred basis of the proposed HOTRG-like scheme, O_x, O_y become diagonal, with their diagonal entries ± 1 .

Let study how two tensors A, \tilde{A} transforms under the coarse graining defined in Eq. (5.5). The goal is to show that two coarser tensors A', \tilde{A}' have the following

¹This is because the pieces of low-rank matrices and the isometric tensors in the RG equation of the HOTRG-like scheme in Eq. (5.6) will inherit the reflection symmetries of the input tensor A .

relation

$$\text{Diagram of } \bar{A}' = \text{Diagram of } A' \text{ with } S_y, S_y^T, d_x, \text{ (B.6)}$$

where d_x is diagonal with entries ± 1 . It follows that the full tensor coarse graining in Eq. (5.6) collapses general gauge freedom into sign ambiguities.

To this end, it is necessary to give a brief introduction about how the GILT constructs the low-rank matrix Q in Fig. 5-1. The environment E is defined, followed by its singular value decomposition,

$$E \equiv \text{Diagram of } E \text{ (grid)} \stackrel{\text{svd}}{=} \text{Diagram of } U, s, V^\dagger \text{ (B.7)}$$

where we refrain from drawing the unknown C matrices in the plaquette. Here, we treat E as a linear map from the vector space of all the legs with ingoing arrows to that of all the legs with outgoing arrows. The matrix Q is constructed from the tensor U and singular value spectrum s in the following fashion. Define a vector t using U ,

$$\text{Diagram of } t \equiv \text{Diagram of } U \text{ (B.8)}$$

Then, the vector t is truncated according to the singular value spectrum s ,

$$t'_i = t_i \frac{s_i^2}{s_i^2 + \epsilon_{\text{gilt}}^2}, \quad \text{(B.9)}$$

where ϵ_{gilt} is the hyper-parameter of the GILT, controlling the approximation error of the above soft truncation. The original GILT paper [26] explains why this soft truncation works. The low-rank matrix Q is constructed from the tensor U^\dagger and the

truncated vector t' as

$$\text{---} \textcircled{Q} \text{---} \equiv \text{---} \textcircled{t'} \text{---} U^\dagger . \quad (\text{B.10})$$

We are all prepared for proving Eq. (B.6). We must determine how various tensors with and without tilde are related. First, it is easy to see the singular value spectrum remains invariant under the gauge transformation, but the tensor U transforms like

$$\text{---} \textcircled{\tilde{U}} \text{---} = \text{---} \textcircled{U} \text{---} \begin{matrix} S_x^T \\ S_x \end{matrix} . \quad (\text{B.11})$$

The vector \tilde{t} is thus invariant, which further means $\tilde{t}'_i = t'_i$ since the tilde version of the right hand side of Eq. (B.9) is the same as the original version. Finally, Eq. (B.10) gives

$$\begin{aligned} \text{---} \textcircled{\tilde{Q}_A} \text{---} &\stackrel{(\text{B.10})}{=} \text{---} \textcircled{\tilde{t}'} \text{---} \tilde{U}^\dagger = \text{---} \textcircled{U^\dagger} \text{---} \begin{matrix} S_x \\ S_x^T \end{matrix} \\ &= \begin{matrix} S_x \\ S_x^T \end{matrix} \text{---} \textcircled{Q_A} \text{---} \begin{matrix} S_x^T \\ S_x \end{matrix} . \end{aligned} \quad (\text{B.12})$$

We see the low-rank matrix Q transforms covariantly under the gauge transformation. If the singular values of Q_A *do not have degeneracy*, after the singular value decomposition of Q_A , we have Q_{Ar}, Q_{Al} transform like (the sign ambiguities coming from singular value decomposition would contribute to d_x in Eq. (B.6), but we do not draw them explicitly)

$$\text{---} \textcircled{\tilde{Q}_{Ar}} \text{---} = \begin{matrix} S_x \\ S_x^T \end{matrix} \text{---} \textcircled{Q_{Ar}} \text{---} \quad \text{and} \quad \text{---} \textcircled{\tilde{Q}_{Al}} \text{---} = \text{---} \textcircled{Q_{Al}} \text{---} \begin{matrix} S_x^T \\ S_x \end{matrix} . \quad (\text{B.13})$$

Equation (B.13) indicates that all the S_x, S_x^T matrices acting on the four horizontal legs of the local patch in Eq. (5.5) will be canceled since the low-rank matrices transform covariantly. We see that the GILT has a preferred basis.

However, there is a final twist. Our calculations show that the low-rank matrices are

projection operators, which are highly degenerated. As a result, the gauge redundancy in the degenerate subspace will leak out. Luckily, the subsequent HOTRG favors the basis where the positive semi-definite matrix MM^\dagger in Eq. (3.13) is diagonal (the sign ambiguities coming from eigenvalue decomposition of MM^\dagger would similarly kick in here and contribute to d_x in Eq. (B.6)), and there is no apparent degeneracy in the eigenvalue spectrum of MM^\dagger based on our numerical results.

Bibliography

- [1] Daiki Adachi, Tsuyoshi Okubo, and Synge Todo. Anisotropic tensor renormalization group. *Phys. Rev. B*, 102:054432, Aug 2020.
- [2] KEN-ICHI Aoki, Tamao Kobayashi, and Hiroshi Tomita. Domain wall renormalization group analysis of two-dimensional ising model. *International Journal of Modern Physics B*, 23(18):3739–3751, 2009.
- [3] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete. Renormalization group flows of hamiltonians using tensor networks. *Phys. Rev. Lett.*, 118:250602, Jun 2017.
- [4] Thomas L. Bell and Kenneth G. Wilson. Nonlinear renormalization groups. *Phys. Rev. B*, 10:3935–3944, Nov 1974.
- [5] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.
- [6] E. Brézin and D. J. Wallace. Critical behavior of a classical heisenberg ferromagnet with many degrees of freedom. *Phys. Rev. B*, 7:1967–1974, Mar 1973.
- [7] John Cardy. *Scaling and Renormalization in Statistical Physics*. Cambridge Lecture Notes in Physics. Cambridge University Press, 1996.
- [8] John L. Cardy. Operator content of two-dimensional conformally invariant theories. *Nuclear Physics B*, 270:186 – 204, 1986.
- [9] Jui-Hui Chung and Ying-Jer Kao. Neural monte carlo renormalization group, 2020.
- [10] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1253–1278, 2000.
- [11] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. On the best rank-1 and rank- (R_1, R_2, \dots, R_N) approximation of higher-order tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000.

- [12] Vin de Silva and Lek-Heng Lim. Tensor rank and the ill-posedness of the best low-rank approximation problem. *SIAM Journal on Matrix Analysis and Applications*, 30(3):1084–1127, 2008.
- [13] Clement Delcamp and Antoine Tilloy. Computing the renormalization group flow of two-dimensional ϕ^4 theory with tensor networks. *Phys. Rev. Research*, 2:033278, Aug 2020.
- [14] Philippe Di Francesco, Pierre Mathieu, and David Sénéchal. *Minimal Models I*. Springer New York, New York, NY, 1997.
- [15] Efi Efrati, Zhe Wang, Amy Kolan, and Leo P. Kadanoff. Real-space renormalization in statistical mechanics. *Rev. Mod. Phys.*, 86:647–667, May 2014.
- [16] G. Evenbly. Algorithms for tensor network renormalization. *Phys. Rev. B*, 95:045117, Jan 2017.
- [17] G. Evenbly and G. Vidal. Entanglement renormalization in two spatial dimensions. *Phys. Rev. Lett.*, 102:180406, May 2009.
- [18] G. Evenbly and G. Vidal. Tensor network renormalization. *Phys. Rev. Lett.*, 115:180405, Oct 2015.
- [19] G. Evenbly and G. Vidal. Local scale transformations on the lattice with tensor network renormalization. *Phys. Rev. Lett.*, 116:040401, Jan 2016.
- [20] Glen Evenbly. Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops. *Phys. Rev. B*, 98:085155, Aug 2018.
- [21] Benyamin Ghojogh, Fakhri Karray, and Mark Crowley. Eigenvalue and generalized eigenvalue problems: Tutorial, 2019.
- [22] V. Giovannetti, S. Montangero, and Rosario Fazio. Quantum multiscale entanglement renormalization ansatz channels. *Phys. Rev. Lett.*, 101:180503, Oct 2008.
- [23] Robert B. Griffiths. Mathematical properties of renormalization-group transformations. *Physica A: Statistical Mechanics and its Applications*, 106(1):59–69, 1981.
- [24] Zheng-Cheng Gu and Xiao-Gang Wen. Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order. *Phys. Rev. B*, 80:155131, Oct 2009.
- [25] Kenji Harada. Entanglement branching operator. *Phys. Rev. B*, 97:045124, Jan 2018.
- [26] Markus Hauru, Clement Delcamp, and Sebastian Mizera. Renormalization of tensor networks using graph-independent local truncations. *Phys. Rev. B*, 97:045111, Jan 2018.

- [27] Michael Hinczewski and A. Nihat Berker. High-precision thermodynamic and critical properties from tensor renormalization-group flows. *Phys. Rev. E*, 77:011104, Jan 2008.
- [28] Leo P. Kadanoff. Scaling laws for ising models near T_c . *Physics Physique Fizika*, 2:263–272, Jun 1966.
- [29] Leo P. Kadanoff. Variational principles and approximate renormalization group calculations. *Phys. Rev. Lett.*, 34:1005–1008, Apr 1975.
- [30] Leo P Kadanoff. Notes on migdal’s recursion formulas. *Annals of Physics*, 100(1):359 – 394, 1976.
- [31] Daisuke Kadoh and Katsumasa Nakayama. Renormalization group on a triad network, 2019.
- [32] Mehran Kardar. *Statistical Physics of Fields*. Cambridge University Press, 2007.
- [33] Tom Kennedy and Slava Rychkov. Tensor rg approach to high-temperature fixed point, 2021.
- [34] Maciej Koch-Janusz and Zohar Ringel. Mutual information, neural networks and the renormalization group. *Nature Physics*, 14(6):578–582, 2018.
- [35] Hyun-Yong Lee and Naoki Kawashima. Tensor-ring decomposition with index-splitting. *Journal of the Physical Society of Japan*, 89(5):054003, 2020.
- [36] Patrick M. Lenggenhager, Doruk Efe Gökmen, Zohar Ringel, Sebastian D. Huber, and Maciej Koch-Janusz. Optimal renormalization group transformation from information theory. *Phys. Rev. X*, 10:011037, Feb 2020.
- [37] Michael Levin. Real space renormalization group and the emergence of topological order. Talk at the IPAM workshop “Topological Quantum Computing”, Mar 2007.
- [38] Michael Levin and Cody P. Nave. Tensor renormalization group approach to two-dimensional classical lattice models. *Phys. Rev. Lett.*, 99:120601, Sep 2007.
- [39] Hai-Jun Liao, Jin-Guo Liu, Lei Wang, and Tao Xiang. Differentiable programming tensor networks. *Phys. Rev. X*, 9:031041, Sep 2019.
- [40] Xinliang Lyu, RuQing G. Xu, and Naoki Kawashima. Scaling dimensions from linearized tensor renormalization group transformations. *Phys. Rev. Research*, 3:023048, Apr 2021.
- [41] Shang-keng Ma. Renormalization group by monte carlo methods. *Phys. Rev. Lett.*, 37:461–464, Aug 1976.
- [42] Y. Meurice. Accurate exponents from approximate tensor renormalizations. *Phys. Rev. B*, 87:064422, Feb 2013.

- [43] AA Migdal. Phase transitions in gauge and spin-lattice systems. *Zh. Eksp. Teor. Fiz*, 69:1457, 1975.
- [44] AA Migdal. Recursion equations in gauge field theories 1975 sov. phys. *JETP*, 42(413):743, 1975.
- [45] Satoshi Morita and Naoki Kawashima. Global optimization of tensor renormalization group using the corner transfer matrix. *Phys. Rev. B*, 103:045131, Jan 2021.
- [46] Yu Nakayama. Scale invariance vs conformal invariance. *Physics Reports*, 569:1 – 93, 2015. Scale invariance vs conformal invariance.
- [47] Th. Niemeijer and J. M. J. van Leeuwen. Wilson theory for spin systems on a triangular lattice. *Phys. Rev. Lett.*, 31:1411–1414, Dec 1973.
- [48] Robert N. C. Pfeifer, Glen Evenbly, and Guifré Vidal. Entanglement renormalization, scale invariance, and quantum criticality. *Phys. Rev. A*, 79:040301, Apr 2009.
- [49] Joseph Polchinski. Scale and conformal invariance in quantum field theory. *Nuclear Physics B*, 303(2):226 – 236, 1988.
- [50] Ramamurti Shankar. *Quantum Field Theory and Condensed Matter: An Introduction*. Cambridge University Press, 2017.
- [51] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifré Vidal. Tensor network decompositions in the presence of a global symmetry. *Phys. Rev. A*, 82:050301, Nov 2010.
- [52] Sukhwinder Singh, Robert N. C. Pfeifer, and Guifre Vidal. Tensor network states and algorithms in the presence of a global $u(1)$ symmetry. *Phys. Rev. B*, 83:115125, Mar 2011.
- [53] Robert H. Swendsen. Monte carlo renormalization group. *Phys. Rev. Lett.*, 42:859–861, Apr 1979.
- [54] Hiroshi Ueda, Kouichi Okunishi, and Tomotoshi Nishino. Doubling of entanglement spectrum in tensor renormalization group. *Phys. Rev. B*, 89:075116, Feb 2014.
- [55] A.C.D van Enter, Roberto Fernández, and Alan D. Sokal. Regularity properties and pathologies of position-space renormalization-group transformations: Scope and limitations of gibbsian theory. *Journal of Statistical Physics*, 72:879, Sep 1993.
- [56] G. Vidal. Entanglement renormalization. *Phys. Rev. Lett.*, 99:220405, Nov 2007.
- [57] G. Vidal. Class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.*, 101:110501, Sep 2008.

- [58] Kenneth G. Wilson. Model of coupling-constant renormalization. *Phys. Rev. D*, 2:1438–1472, Oct 1970.
- [59] Kenneth G. Wilson. Renormalization group and critical phenomena. i. renormalization group and the kadanoff scaling picture. *Phys. Rev. B*, 4:3174–3183, Nov 1971.
- [60] Kenneth G. Wilson. The renormalization group: Critical phenomena and the kondo problem. *Rev. Mod. Phys.*, 47:773–840, Oct 1975.
- [61] Kenneth G. Wilson. The renormalization group and critical phenomena. *Rev. Mod. Phys.*, 55:583–600, Jul 1983.
- [62] Kenneth G. Wilson and Michael E. Fisher. Critical exponents in 3.99 dimensions. *Phys. Rev. Lett.*, 28:240–243, Jan 1972.
- [63] Z. Y. Xie, J. Chen, M. P. Qin, J. W. Zhu, L. P. Yang, and T. Xiang. Coarse-graining renormalization by higher-order singular value decomposition. *Phys. Rev. B*, 86:045139, Jul 2012.
- [64] Z. Y. Xie, H. C. Jiang, Q. N. Chen, Z. Y. Weng, and T. Xiang. Second renormalization of tensor-network states. *Phys. Rev. Lett.*, 103:160601, Oct 2009.
- [65] Shuo Yang, Zheng-Cheng Gu, and Xiao-Gang Wen. Loop optimization for tensor network renormalization. *Phys. Rev. Lett.*, 118:110504, Mar 2017.
- [66] Lexing Ying. Tensor network skeletonization. *Multiscale Modeling & Simulation*, 15(4):1423–1447, 2017.
- [67] H. H. Zhao, Z. Y. Xie, Q. N. Chen, Z. C. Wei, J. W. Cai, and T. Xiang. Renormalization of tensor-network states. *Phys. Rev. B*, 81:174411, May 2010.