

Tensor Networks and the Renormalization Group

by

Markus Vili Antero Hauru

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Physics

Waterloo, Ontario, Canada, 2018

© Markus Vili Antero Hauru 2018

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

- External Examiner: Dr. Frank Pollmann
Associate Professor
Dept. of Physics, Technical University of Munich
- Supervisor(s): Dr. Guifré Vidal
Adjunct Professor
Dept. of Physics and Astronomy, University of Waterloo
Faculty
Perimeter Institute for Theoretical Physics
- Internal Member: Dr. K. Rajibul Islam
Assistant Professor
Dept. of Physics and Astronomy, University of Waterloo
- Internal-External Member: Dr. Achim Kempf
Professor
Dept. of Applied Mathematics, University of Waterloo
- Other Member(s): Dr. Robert Myers
Adjunct Professor
Dept. of Physics and Astronomy, University of Waterloo
Faculty
Perimeter Institute for Theoretical Physics

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

Chapter 2 of this thesis consists mostly of material published in Reference [52], co-authored with Clément Delcamp and Sebastian Mizera.

Chapter 3 of this thesis consists mostly of material published in Reference [53], co-authored with Glen Evenbly, Wen Wei Ho, Davide Gaiotto, and Guifré Vidal.

Chapter 4 of this thesis consists mostly of material published in Reference [55], co-authored with Guifré Vidal.

Abstract

Tensor networks are a class of methods for studying many-body systems. They give a geometrical description of the internal structure of many-body states, operators, and partition functions, that can be used to implement efficient algorithms to simulate them numerically. In this thesis, after providing a very brief overview of the field, we present new tensor network methods for three different use cases.

First, we present a new real-space renormalization group algorithm for tensor networks. Compared to existing methods, its advantages are its low computational cost, simplicity of implementation, and applicability to any network. We benchmark it on the 2D classical Ising model and find accuracy comparable with the best existing tensor network methods. Due to its simplicity and generalizability, we consider our algorithm to be an excellent candidate for implementation of real-space renormalization in higher dimensions, and discuss some of the details and the remaining challenges in 3D.

Second, we show how certain topological conformal defects of critical lattice theories can be represented as tensor networks, using the 2D classical Ising model as an example. Furthermore, we show how such tensor network descriptions, combined with a renormalization group algorithm, can be used to obtain accurate estimates of the universal properties of these defects. We also show how coarse-graining of defects can be applied to any conformal defect (i.e. not just topological ones), and yields a set of associated scaling dimensions.

Finally, we leave behind the focus on renormalization group methods, and present a method for spatially resolving the overlap between two tensor network states, to obtain localized information about the similarities and differences between them. For a given region, the similarity of two states in this region can be quantified by the Uhlmann fidelity of their reduced density matrices, and we show how such fidelities can be efficiently computed in many cases when the two states are represented as tensor networks. We demonstrate the usefulness of evaluating such subsystem fidelities with three example applications: studying local quenches, comparing critical and non-critical states, and quantifying convergence in tensor network simulations.

Acknowledgements

The work that has gone into this thesis has only been possible due to the help and support of many people around me. I take this space to briefly thank the most important among them.

Most importantly, I thank my supervisor Guifré Vidal for guiding me through the process of morphing from a student to a scientist: for all the physics and multilinear algebra, his willingness to lead me by the hand when I did not know what I was doing, and his trust in me when I wanted to try my own luck. I thank all my coauthors for their contributions to our work, a summary of which is this thesis, as well as for teaching me a lot of physics. I especially thank Clément Delcamp and Sebastian Mizera, with whom all the fun in Chapter 2 was had.

I thank the members of my supervisory committee and my PhD committee for providing periodic sanity checks and helpful conversations along the way, and for their feedback on this thesis.

I thank all the people of Perimeter Institute, from funders to faculty to facilities, for creating a vibrant and nurturing environment in which to work. Special thanks go out to the staff of the Bistro, as well as whoever decided that PI should have a sauna. I thank my friends in PI, including most of the graduate students and many postdocs and visitors, for creating a community of peers that is supportive, diverse, open, and amusing. I thank my friends outside of PI, most notably my roommates Andrew, Nick, and Adam, for distracting me, and for confirming most stereotypes of Canadians I have come across, including that of being friendly and welcoming. I thank my friends and loved ones in Europe for occasionally resetting my frame of reference, and reminding me of what matters.

I thank the tax payers of Canada for paying for my food, rent, and leisure, as well as for most of the research facilities and infrastructure that I have used over the last four years.

Table of Contents

List of Tables	xii
List of Figures	xiii
1 Introduction	1
1.1 Tensor networks	4
2 Graph Independent Local Truncations	8
2.1 Introduction	8
2.2 Background	11
2.2.1 Partition functions as tensor networks	11
2.2.2 Tensor network coarse-graining	13
2.3 Environment spectrum	17
2.4 Truncating bonds using the environment spectrum	19
2.4.1 Truncated singular value decomposition	20
2.4.2 Gilt: Graph independent local truncation	21
2.5 Gilt-TNR	24
2.6 Benchmark results	26
2.7 Gilt-TNR in 3D	31
2.8 Quantum states	35
2.9 Discussion	36

3	Topological defects	38
3.1	Introduction	38
3.1.1	Defects and transfer matrices	38
3.1.2	Structure of this chapter	41
3.1.3	Source code	42
3.2	Critical Ising model	42
3.2.1	Classical partition function	42
3.2.2	Quantum spin chain	43
3.2.3	Ising CFT	44
3.2.4	Extracting the universal data	45
3.3	Tensor networks	47
3.3.1	Tensor network representation	49
3.3.2	Coarse-graining	51
3.3.3	Numerical results	53
3.3.4	Multiscale Entanglement Renormalization Ansatz	55
3.4	Topological defects	55
3.5	Symmetry defect D_ϵ	57
3.5.1	Lattice representation	58
3.5.2	Tensor network representation	59
3.5.3	Coarse-graining	60
3.5.4	Numerical results	62
3.5.5	MERA	62
3.6	Duality defect D_σ	65
3.6.1	Lattice representation	65
3.6.2	Tensor network representation	66
3.6.3	Coarse-graining	69
3.6.4	Numerical results	71

3.6.5	MERA	72
3.7	Generic conformal defects	72
3.7.1	Family of defects for the Ising model	74
3.7.2	Numerical results	75
3.8	Discussion	75
3.8.1	Conformal defects in other models	75
3.8.2	Why Tensor Network Renormalization?	78
3.8.3	Alternative approach to extracting conformal data from a critical lattice model	80
3.8.4	Related work	81
4	Uhlmann fidelities for tensor networks	82
4.1	Introduction	82
4.2	Uhlmann fidelity of subsystems	85
4.3	Evaluating subsystem fidelities for tensor network states	87
4.3.1	Matrix Product States	87
4.3.2	Tree Tensor Networks	91
4.4	Applications	94
4.4.1	Local quench	94
4.4.2	Comparing states at different scales	96
4.4.3	Convergence of simulations	97
4.5	Conclusion	99
	References	101
	APPENDICES	111

A	Appendices for Chapter 2	112
A.1	Corner-double-line tensors	112
A.1.1	CDL and TRG	112
A.1.2	CDL and Gilt	113
A.1.3	CDL and Gilt-TNR	116
A.2	Choosing t'	116
A.3	Source code	119
B	Appendices for Chapter 3	120
B.1	Tensor network renormalization	120
B.1.1	Coarse-graining the D_ϵ defect	122
B.1.2	Coarse-graining the D_σ defect	126
B.2	Increasing the period of the conformal spins	129
B.3	Results for the 3-state Potts model	130
B.4	Number of degrees of freedom in Z_{D_σ}	132
C	Appendices for Chapter 4	133
C.1	Generic form of purifications	133
C.2	Solving the maximization	135

List of Tables

2.1	Conformal data of the Ising CFT, obtained with various tensor network coarse-graining methods.	30
3.1	Conformal data of the Ising CFT, obtained using Tensor Network Renormalization.	53
3.2	Primary operators of the Ising CFT with different topological defects. . .	57
3.3	Conformal data of the Ising CFT with a symmetry defect, obtained using Tensor Network Renormalization.	64
3.4	Conformal data of the Ising CFT with a duality defect, obtained using Tensor Network Renormalization.	72

List of Figures

2.1	Constructing a tensor network for a classical statistical physics model. . . .	13
2.2	The Tensor Renormalization Group algorithm.	15
2.3	The toy model of corner-double-line tensors.	16
2.4	The Gilt-TNR algorithm.	25
2.5	Error in free energy of the critical 2D classical Ising model at different bond dimensions, for TRG and for Gilt-TNR.	27
2.6	RG flows of coarse-grained tensors, for Tensor Renormalization Group and Gilt-TNR.	28
2.7	Comparing environment spectra in 2D and 3D.	34
2.8	Applying Gilt-TNR to a network with a boundary.	35
2.9	The tensor network created by applying Gilt-TNR to a network with a boundary.	37
3.1	Partition function and its transfer matrix on a torus, with and without a topological defect.	39
3.2	Conformal data of the Ising CFT, obtained using exact diagonalization. . . .	48
3.3	Partition function, transfer matrix, and translation operator as tensor networks.	49
3.4	The invariance of tensor A under the symmetry transformation V	51
3.5	Coarse-graining a tensor network partition function.	52
3.6	The coarse-grained transfer matrix and translation operator.	53
3.7	Conformal data of the Ising CFT, obtained using Tensor Network Renormalization.	54

3.8	Multiscale Entanglement Renormalization Ansatz on eight sites.	55
3.9	Tensor networks for describing the symmetry defect of the 1D quantum Ising model.	59
3.10	Conformal data of the Ising CFT with a symmetry defect, obtained using Tensor Network Renormalization.	61
3.11	Coarse-grained the tensor network for the symmetry defect of the Ising model.	62
3.12	Conformal data of the Ising CFT with a symmetry defect, obtained using Tensor Network Renormalization.	63
3.13	Multiscale Entanglement Renormalization Ansatz with a symmetry defect.	64
3.14	Tensor network for the sum of the identity and symmetry defects of the 1D quantum Ising model.	67
3.15	Tensor network for the duality defect of the 1D quantum Ising model, and related operators.	68
3.16	Conformal data of the Ising CFT with a duality defect, obtained using exact diagonalization.	70
3.17	Coarse-grained representation of the duality defect of the 1D quantum Ising model.	71
3.18	Conformal data of the Ising CFT with a duality defect, obtained using Tensor Network Renormalization.	73
3.19	Multiscale Entanglement Renormalization Ansatz for the duality defect of the 1D quantum Ising model.	74
3.20	Exact diagonalization results for a family of defects of the 1D quantum Ising model.	76
3.21	Tensor network renormalization results for a family of defects of the 1D quantum Ising model.	77
4.1	An example of subsystem fidelities for a local quench.	83
4.2	Subsystem fidelities for a local quench.	95
4.3	Window fidelities between a critical and off-critical state.	97
4.4	Studying convergence in bond dimension using window fidelities.	99

A.1	The Tensor Renormalization Group algorithm, as it applies to the corner-double-line model.	112
A.2	Gilt-TNR algorithm applied to the corner-double-line model.	117
B.1	An overview of Tensor Network Renormalization.	121
B.2	The \mathbb{Z}_2 invariance property of the w tensor (in yellow).	122
B.3	Coarse-graining the product of a transfer matrix and a translation.	123
B.4	Coarse-graining a symmetry defect of the 1D quantum Ising model.	124
B.5	Coarse-graining the product of a transfer matrix and a translation for the symmetry defect of the 1D quantum Ising model.	125
B.6	Coarse-graining a line defect of two tensors.	127
B.7	Coarse-graining the product of a transfer matrix and a translation for the duality defect of the 1D quantum Ising model.	128
B.8	An additional coarse-graining step to increase the period of the conformal spins.	129
B.9	Conformal data of the 3-state Potts model, obtained using Tensor Network Renormalization.	131

Chapter 1

Introduction

A major theme of modern physics is understanding the collective behavior of systems of many constituents. Simple interactions between microscopic degrees of freedom lead to complex and hard to predict emergent phenomena, when macroscopic numbers of them are present. This phenomenon of emergence permeates the modern scientific worldview, and is central to understanding systems from biology to economics. Closer to the topic of this thesis, it is the foundation of material physics and statistical physics. Various phases of matter from magnets to superconductors arise from local interactions between individual atoms or electrons, with transitions between phases happening abruptly when the interactions are slightly changed, or more energy is injected into the system. Emergence is a key concept in modern high energy physics too, from particle physics to quantum gravity, where recent ideas propose that gravity, or in fact spacetime itself, is emergent in nature.

A dominant paradigm in many-body physics since the 1970s has been that of the renormalization group. Starting from the early ideas of Kadanoff and others in the 60s [64, 65], and reaching full fruition in the work of Wilson [109], renormalization has since come to dominate the way physicists think of nature. It has put quantum field theories on a solid foundation as being intrinsically effective in nature, explained the curious similarity in the emergent behavior of entirely disparate systems, known as universality, and underlies most of modern statistical and high energy physics. The core philosophy of the renormalization group hinges on the observation, that even though physicists typically define theories on the microscopic level, the measurements we care about happen at far larger length scales. Thus it makes sense to clump together chunks of the microscopic degrees of freedom into a smaller number of coarse-grained degrees of freedom, and to describe the system in terms of these new, coarser variables. The interactions between the

coarse-grained degrees of freedom are determined from the microscopic interactions, but have lost some of the details in the transformation, such that only the long-range behavior of the system remains as it was. This allows one to move from one theory, valid at some range of length scales, to another one, that is valid only at the longer end of this range, but that is often simpler in nature. This creates a flow, known as the renormalization group flow, in the space of physical theories, usually thought of as a flow in the space of couplings in a Hamiltonian or Lagrangian. This flow is parametrized by length scale, and the usual question to ask is, given a microscopic theory, at long length scales, what kind of theory it flows to. Successfully answering this question means distilling the macroscopic physics out of the microscopic.

In quantum many-body physics in particular, a parallel theme has emerged since the 1990s, where systems are understood more through information theory. Instead of thinking in terms of order parameters, people coming to many-body physics from a quantum information background brought in the view point of entanglement, fidelities, and complexity. Whether one takes Wheeler’s radical “everything is information” stance, or a more pragmatic approach rooted in concrete observables, the notion of entanglement and other quantum information concepts are nowadays considered standard tools in understanding emergent quantum phenomena.

One might ask, what is so difficult about many-body systems in the first place, that all this talk about different perspectives to approach it is necessary. Without attempting to give a complete answer, let us comment on the issue from the point of view of numerical simulations. After all, only a tiny fraction of systems of interest yield themselves to exact analytic solutions, and in the generic case computational methods are necessary. Much of the difficulty in simulating many-body systems can be summarized by the statement, “the state space is just too big”. To be more concrete, consider a system of N classical spins, i.e. degrees of freedom that only have two possible states. This system as a whole has 2^N different configurations it can be in. Leaving aside scales of Avogadro’s number, $N = 10^{23}$, and considering just a measly $N = 100$ spins, the number of states is $2^{100} \approx 10^{30}$. If this does not yet seem imposing, consider making the spins quantum mechanical in nature, and the state space becomes a vector space with 2^N dimensions. This means that simply storing a single state of the 100 quantum spins would require storing 10^{30} coefficients. Any straight-forward approach to simulate such a system with any imaginable (classical) computer is clearly doomed to fail.

Tensor networks are a class methods for studying many-body systems, that draws most of its inspiration from the renormalization group and from quantum information theory. The core of its pitch is this: Even though the state space of a many-body system is huge, locality strongly restricts which parts of this space are relevant. Interactions are local in

spacetime, and consequently correlations between different degrees of freedom are typically restricted by locality too. In other words, even though the quantum state of N spins is a 2^N dimensional object, a tensor networker would say, “Let’s assume the states we are studying can be built out of local pieces, that may directly share information with their neighbours only. Not every state in the state space fits this mold, in fact only a vanishingly small fraction does, but because of locality, many physically interesting states probably do.” Likewise, for a system of classical statistical physics, when asked to compute a partition function that is a sum over 2^N terms, she would say, “Granted, there are 2^N terms, but we know that they are made up of independent contributions coming from local neighbourhoods, and I bet there is a way to use that to simplify things and speed them up.”

This turns out to be an immensely fruitful approach. By placing locality-informed constraints on the possibilities we consider, much of the interesting physics is still captured, but vast simplifications are allowed, where the 2^N parameters may be replaced by order $\mathcal{O}(N)$, $\mathcal{O}(\log N)$, or even $\mathcal{O}(1)$ parameters. Since their birth in the early 90s [27, 40, 83, 106, 107], tensor networks have come to dominate simulations of 1-dimensional quantum systems [11, 23, 66, 85, 86, 100, 101, 103, 108], and are a strong candidate for the state-of-the-art for many models in two dimensions [16, 17, 18, 24, 60, 91, 95, 98, 112]. Lately they have also been recognized as an efficient tool for studying classical lattice models, especially in two dimensions [3, 37, 47, 68, 111, 113]. In addition to being potent numerical tools, one of the key advantages of tensor networks is that their explicit geometrical structure provides plenty of intuition about the physics at play. Partially because of this, over the last 10 years tensor networks have seen a proliferation of applications to a range of subjects outside their original domain in condensed matter and statistical physics, including in holography [20, 22, 56, 78, 94], quantum field theories [41, 44, 48, 49, 59, 99], and even machine learning [6, 15, 74, 90].

In this thesis, we study various questions related to tensor networks, mostly concentrating on their relation to the renormalization group. Below, we give a basic, high-level introduction to what tensor networks are. In Chapter 2, we discuss the challenges of tensor network renormalization group algorithms, present a new algorithm that we call *Gilt*, demonstrate its usefulness, and discuss its merits as compared to other methods of the same nature. In Chapter 3 we show how one can describe conformal defects of critical lattice models in tensor networks, and use this description, together with tensor network coarse-graining methods, to extract conformal data of these defects to high accuracy. In Chapter 4 we leave behind the focus on renormalization, and show how one can spatially resolve overlaps between tensor network states using Uhlmann fidelities of their reduced density matrices, and present example cases where such fidelities are of interest or use.

1.1 Tensor networks

Tensor networks are a way of describing many-body systems, that is geometrically intuitive, and leads to efficient algorithms for simulating them. They can be utilized to describe both classical and quantum lattice systems. Starting from White’s Density Matrix Renormalization Group [106, 107] and its description in terms of Matrix Product States (MPS) [27, 83] in the 1990s, tensor networks have grown over the last 20 years to cover various kinds of systems and allow for answering many different types of questions, such as finding low energy subspaces of quantum systems, simulating quenches, and computing RG flows of classical and quantum lattice systems.

The core idea of tensor networks is the following. Both in classical and quantum physics, many quantities of interest are either vectors in or operators on a vector space, whose number of dimensions is exponential in the system size: The state space of a quantum many-body system of N qudits is of dimension d^N , and the sum over all configurations in a classical partition function can be seen as a trace in a d^N -dimensional space. However, the operators and vectors that we are interested in are often far from generic, but rather are restricted by locality. For instance, in low-energy states of quantum systems, most entanglement is between sites that are close to each other, leading to the area law of entanglement [29]. Tensor networks exploit this structure, by proposing that various operators and states in these d^N -dimensional vector spaces can be, at least approximately, built out of smaller pieces, namely various tensors, multiplied and contracted together in a pattern that is often easiest to write down as a network graph, and that mimics the locality-imposed structure of the object.

As an example, consider the 2^N dimensional state of a chain of N quantum spins:

$$|\psi\rangle = \sum_{i_1, \dots, i_N=1}^2 \psi_{i_1 \dots i_N} |i_1 \dots i_N\rangle. \quad (1.1)$$

The coefficients $\psi_{i_1 \dots i_N}$ can be thought of as a rank- N tensor in $\otimes_{i=1}^N \mathbb{C}^2$, and a tensor network description of it decomposes it into a contraction of many smaller tensors.¹ For example, a Matrix Product State (MPS) description of ψ consists of rank-3 tensors of dimensions $\chi \times \chi \times 2$, for some χ . There is a tensor $A_{i_n v_n v_{n+1}}^{(n)}$ for each spin n , and together they make up

$$\psi_{i_1 \dots i_N} = \sum_{v_1, \dots, v_N=1}^{\chi} A_{i_1 v_1 v_2}^{(1)} A_{i_2 v_2 v_3}^{(2)} \dots A_{i_N v_N v_1}^{(N)}. \quad (1.2)$$

¹Note that when we use the word tensor, we mean it in the multilinear algebra sense of a multilinear map, i.e. an array of numbers. This should not be confused with the usage common in field theory and gravity, where by “tensor” physicists often mean a tensor field, with specific covariance properties.

Expressions like the one on the right are cumbersome and hard to read, and we thus describe them instead in a graphical notation, where each tensor is represented as some shape, and each of its indices is a link, often called a leg or bond, sticking out of it. Contracted indices join two tensors, whereas uncontracted ones have one end dangling free. In this notation, the above equation can be written as

$$\psi = \begin{matrix} \text{---} \\ | \\ \text{---} \end{matrix} \begin{matrix} A^{(1)} & A^{(2)} & \dots & A^{(N)} \\ | & | & & | \\ i_1 & i_2 & & i_N \end{matrix} \quad (1.3)$$

Key to understanding the power and limitations of tensor networks is the dimension χ of the contracted indices, known as the bond dimension. It may be different for each of the contracted indices, but for simplicity we assume here that it is not. In the MPS decomposition above, we have described the 2^N -dimensional ψ with N tensors $A^{(n)}$, each of which has $2\chi^2$ elements, and thus in total we have used only $2\chi^2 N$ parameters. That such a description is possible for ψ , is an ansatz, the power of which depends on χ . For a large enough χ (exponential in N), any state ψ can be decomposed as an MPS. However, for a large class of states of interest, a much smaller χ is sufficient. For $\chi = 1$, the MPS ansatz can describe all product states, but nothing else. For $\chi = 2$, a small amount of entanglement is possible. Moreover, due to the contracted legs connecting nearest neighbours only, most $\chi = 2$ MPS states have entanglement only between near-by sites. With larger values of χ , more entanglement is made possible, but the locality structure is still retained. Ground states of gapped Hamiltonians typically obey the area law of entanglement, which for 1D systems means that the entanglement between left-right partitions may first grow as one increases the system size, but then saturates to a constant. This means that a finite bond dimension χ , that does not depend on N , is sufficient to write such states accurately as an MPS. Thanks to this, the computational cost of handling the MPS representation of ψ , and computing many of its properties such as correlation functions of local operators, scales only linearly in N . This is efficiency gained by relying on the special structure of low-energy states arising from locality, and it allows accurate study of spin chains of thousands of spins, or by assuming translation invariance, even infinite spin chains.

MPS was the first tensor network method proposed, and remains the most important one, as it is a superb tool for 1-dimensional quantum systems. However, many other tensor networks have since been created, for various other use cases. If, for instance, in the above example the N spins were organized in a 2-dimensional grid, a different tensor network, with nearest-neighbour tensors being contracted with each other in a 2-dimensional pattern, would be a more natural ansatz. Such higher dimensional generalizations of MPSes are

known as Tensor Product States (TPS) or Projected Entangled Pair States (PEPS) [73, 98]. Later in this thesis we will meet a few other networks with different structures, as well as networks used for describing classical partition functions.

Simply drawing a tensor network diagram does not yet guarantee that one has created a useful method. The structure of the network must respect the internal structure of the state or other object that it is describing, or otherwise the various bond dimensions in the network have a tendency to grow exponentially in system size. There must also be an efficient way to contract the network when evaluating observables or other quantities of interest, and there must be efficient ways of finding tensor network decompositions of a given form for an object of interest. The latter typically involves some optimization method, that may for instance minimize the energy within the set of tensor network states of a given structure, to find the ground state. A well-designed tensor network algorithm has a computational cost for all the necessary operations that is at most polynomial in the system size and the various bond dimensions involved. All tensor network algorithms are inherently immune to the sign-problem, that plagues Monte Carlo based approaches, and can thus equally well be used on bosons and fermions [19].

As mentioned in the previous section, in addition to providing an efficient method for simulating a given system, a good tensor network description tells us much about the qualitative nature of the physics at play. For instance, whenever a state can be described as an MPS of finite bond dimension, one immediately knows that this state has exponentially decaying two-point correlators at large distances, and can easily extract the leading exponent [85].² One also knows that the state must have at most area law entanglement, and can easily obtain the Schmidt decomposition (and thus the entanglement spectrum) for any left-right bipartition. Moreover, simply by looking at the graphical picture of the network, the structure of correlations is immediately obvious: Each contracted index of dimension χ can be roughly speaking seen as having the potential to carry $\log \chi$ bits of entanglement (or more generally, correlations) across it. For some networks other properties may be evident, such as for the Multiscale Entanglement Renormalization Ansatz (MERA) [103], which has an internal causal structure, and an interesting connection to the circuit complexity of states, among other properties.

The development of tensor network techniques has been guided by the renormalization group philosophy from the beginning, and many prominent tensor network methods implement various notions of scale transformations. Most relevant for this thesis are Tree

²An exception to this are states with long-range order, which may also be represented with MPSes. They too, however, are easy to identify given the MPS tensors, and one can observe the amount of long-range entanglement and extract the leading order decay in two-point correlators.

Tensor Networks (TTNs) [86] and the Multiscale Entanglement Renormalization Ansatz (MERA) [103], which apply to quantum systems formulated using operators on Hilbert spaces, and especially their Lagrangian equivalents, the Tensor Renormalization Group (TRG) [68] and Tensor Network Renormalization (TNR) [37], that can be applied to D -dimensional quantum systems or $D + 1$ -dimensional classical ones. They implement transformations, much in the spirit of Kadanoff’s original spin-blocking proposal, where each local patch of microscopic degrees of freedom is mapped to a single coarse-grained degree of freedom. The TTN and MERA networks do this for quantum systems, through applying local, finite-depth quantum circuits to the state space of the lattice, to separate the long distance physics from the short, and then project the short distance physics onto a product state, leaving only the coarse-grained degrees of freedom. TRG and TNR give an analogous treatment to networks that represent partition functions of classical statistical physics models, again locally deforming the physics to remove short-range details, and joining degrees of freedom together to form a coarse-grained model. These algorithms are of interest both for conceptually understanding renormalization and related concepts, such as criticality, conformal field theories and holographic dualities, and for the purpose of simulating many-body physics, especially critical phenomena, efficiently. TNR and TRG will be discussed in more detail in Chapter 2, where we propose a competing method for TNR, and in Chapter 3, where we show how methods like these can be used to study topological defects. Much more information about tensor networks in general can be found in the references, with the reviews of Orus [75] and Bridgeman & Chubb [7], as well as Glen Evenbly’s website [30], being good places to start.

Chapter 2

Graph Independent Local Truncations

2.1 Introduction

As mentioned in the introduction, tensor networks can be used to study both classical and quantum lattice systems. In this chapter we focus on the former case, where a tensor network representing a classical partition function ought to be contracted in an efficient way. The best-known algorithm for doing this is the Tensor Renormalization Group (TRG) [68]. Owing to its simplicity and efficiency, it has proven to be a potent tool for evaluating observables for 2D lattice models in statistical physics [13, 57, 69, 70, 115]. It is based on replacing tensors with their low-rank approximations using a truncated singular value decomposition (SVD), and contracting the tensors together in a way reminiscent of Kadanoff's spin blocking [64, 65]. Although TRG draws inspiration from Renormalization Group (RG) methods and performs a kind of coarse-graining transformation, it is well known that it does not properly implement an RG transformation [47, 67]: Some details of the UV physics survive the coarse-graining, and thus the RG flows produced by TRG are not the physically correct ones. We review TRG and its key properties in Sect. 2.2.

The issues of TRG were first addressed with the introduction of the Tensor Network Renormalization (TNR) algorithm [37]. It is a proper RG transformation, that yields the physically correct RG flows. Furthermore, it provides significantly more accurate observables than TRG at the same bond dimension, although at a higher computational cost. TNR has been studied in many contexts, including holography, topological defects, and conformal field theories [8, 9, 20, 21, 39, 53, 71]. Other algorithms have also been proposed, such as

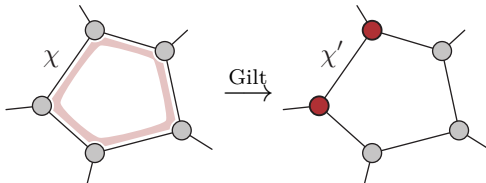
Loop-TNR [113] and TNR+ [3]. They all solve the problems of TRG in their own way, removing all UV details during the coarse-graining transformation.

However, both TNR and Loop-TNR have their shortcomings (TNR+ shares most of its features with Loop-TNR, so we leave it out of the comparison for now). Most importantly, generalizing them to other lattice types, in particular 3D lattices, is not easy. TNR is specifically designed for the square lattice, and although the philosophy is clear, applying it in some other context would require significantly redesigning the algorithm. Loop-TNR is somewhat easier to adapt to other lattices in 2D, but for it too, a generalization to 3D is far from being trivial. Although various schemes generalizing these algorithms to 3D can be constructed [51, 102], they seem to be characteristically plagued by unfeasibly high computational costs. In addition, both TNR and Loop-TNR are significantly more complicated to implement than TRG, since they replace a straightforward truncated SVD with iterative optimizations, that depend on an initial guess and may have issues with convergence and local minima of the cost function.

In this light, we ask whether there exists a simpler and easier-to-generalize way of performing real-space RG on tensor networks. The first question is, what exactly are the UV details, or local correlations, that TRG fails to renormalize properly? The usual answer is given using a toy model called corner-double-line (CDL) tensors, which we review in Sect. 2.2. Beyond CDL, the discussion regarding local correlations has remained on a purely qualitative level. In Sect. 2.3 we make it more concrete by introducing the *environment spectrum*, which gives a quantitative measure of what is meant by such local correlations.

Using this measure, the next natural question to ask is, can we use it to remove these local correlations from the network? After all, the problem with TRG is that a subset of such correlations, which correspond to short-range details and thus should be removed (or “integrated out” in the momentum-space RG terminology) in the RG transformation, remain in the network after the coarse-graining. In Sect. 2.4 we present a solution in the form of an algorithm for performing what we call *graph independent local truncations*, or *Gilts*. The Gilt procedure uses the environment spectrum to truncate a bond dimension of a single leg in a network, and in the process can remove local correlations in a neighborhood of this leg. It can be applied equally easily to any network or lattice, and only modifies the tensors next to the leg that is being truncated, leaving the network geometry intact. Using

the graphical notation, this can be summarized as



with bond dimensions $\chi' < \chi$.

Gilt provides a way of fixing the shortcoming of TRG with minimal changes: We can simply precede a TRG coarse-graining transformation with a step where Gilt is applied to deal with the UV details that TRG is unable to handle properly. This combination of Gilt and TRG, which we call *Gilt-TNR*, is a fast, simple and generalizable proper RG transformation for tensor networks. We discuss it in detail in Sect. 2.5.

In Sect. 2.6 we benchmark Gilt-TNR with the 2D classical Ising model. We confirm that Gilt-TNR leads to the correct scale-invariant tensors in the phases and at criticality, which explicitly demonstrates that it fixes the conceptual shortcomings of TRG by properly [37] implementing the philosophy of RG. We show that estimates for observables, such as free energy at criticality or spectrum of scaling dimensions, calculated with Gilt-TNR, are on par with the best ones obtained with TNR and Loop-TNR, and require only moderate computational effort.

In Sect. 2.7 we discuss applying Gilt-TNR in 3D. Its basic design is straightforward, and comes with a surprisingly low computational cost. However, we show evidence that significantly higher bond dimensions will be needed in 3D to achieve high quality physical results, and comment on the status 3D tensor network coarse-graining algorithms in general, and the implementation of Gilt-TNR on the cubical lattice in particular.

We briefly discuss the way Gilt-TNR can be used to represent ground and thermal states for quantum Hamiltonians in Sect. 2.8, before concluding this chapter in Sect. 2.9.

It should be noted that during the development of the Gilt procedure, another RG method called Tensor Network Skeletonization (TNS) [114] was published, which shares some of the features with Gilt-TNR. More specifically, TNS also focuses on truncating individual legs in a network, separates the coarse-graining step from the step removing local correlations, and can be applied to any network or lattice. However, unlike Gilt-TNR, TNS relies on an iterative optimization procedure, which is highly dependent on the initial guess and blind to the nature of the local correlations which it is trying to remove. Since Ref. [114] presents only limited numerical results, we cannot perform quantitative comparison between our methods.

This work is accompanied by ready-to-use source code implementing Gilt-TNR for the square lattice, which is freely available at arxiv.org/src/1709.07460v1/anc. It can be used to reproduce all the numerical results presented in Sect. 2.6. Another version of the source, which remains under development and includes an ongoing effort to produce an efficient implementation of Gilt-TNR on the cubical lattice, can be found at github.com/Gilt-TNR/Gilt-TNR. We discuss the source code in more detail in App. A.3.

2.2 Background

In this section we introduce in detail the tensor networks that we are dealing with in this chapter, namely ones describing classical partition functions and ground or thermal states of quantum Hamiltonians. We also review known coarse-graining algorithms for such networks. A reader familiar with tensor networks in this context may want to move to the next section.

2.2.1 Partition functions as tensor networks

Consider a square lattice with a classical configuration variable σ_i on each lattice site i . For a given configuration $\{\sigma\}$, assume the energy of the system is given by a Hamiltonian $H(\{\sigma\})$ that consists of only local terms. The Boltzmann weights are

$$W(\{\sigma\}) = e^{-\beta H(\{\sigma\})} \quad (2.1)$$

and their sum yields the partition function

$$Z = \sum_{\{\sigma\}} e^{-\beta H(\{\sigma\})} \quad (2.2)$$

with $\beta = \frac{1}{T}$ and T the temperature of the system.

Let us now derive a tensor network representation of the partition function (2.2). For concreteness, let us work with the 2D classical Ising model, which we later use as a benchmark model. The configuration variables are spins which can take two values $\sigma \in \{\uparrow, \downarrow\}$, and the Hamiltonian is

$$H(\{\sigma\}) = \sum_{\langle i,j \rangle} h(\sigma_i, \sigma_j) \quad (2.3)$$

where $h(\sigma_i, \sigma_j) = -1$ when $\sigma_i = \sigma_j$, and $+1$ otherwise, and $\langle i, j \rangle$ denotes nearest-neighbor vertices. To each pair of neighboring sites i and j , we can associate a local Boltzmann weight $W_{\sigma_i \sigma_j} = e^{-\beta h(\sigma_i, \sigma_j)}$, which can be written as the following matrix:

$$W = \begin{pmatrix} W_{\uparrow\uparrow} & W_{\uparrow\downarrow} \\ W_{\downarrow\uparrow} & W_{\downarrow\downarrow} \end{pmatrix} = \begin{pmatrix} e^{\beta} & e^{-\beta} \\ e^{-\beta} & e^{\beta} \end{pmatrix} =: \text{---}\bigcirc\text{---} . \quad (2.4)$$

In the last equality we have used the graphical tensor network notation, introduced in Chapter 1. As a reminder, in this notation each solid shape, such as the above circle, represents a tensor, and each leg (or link or bond) of it represents an index of the tensor. Connecting two legs means identifying them, and summing over them, i.e., performing a tensor contraction. W only has two legs which indicates that we are dealing with a rank-2 tensor, i.e., a matrix.

To obtain the partition function (2.2), we can take the tensor product of all the W 's between different nearest-neighbor pairs, and sum over the spins:

$$Z = \sum_{\{\sigma\}} \bigotimes_{\langle i, j \rangle} W_{\sigma_i \sigma_j} . \quad (2.5)$$

In the case of a 2×2 lattice, the above can be written as a tensor network as

$$Z = \text{---}\bigcirc\text{---} \text{---}\bigcirc\text{---} \text{---}\bigcirc\text{---} \text{---}\bigcirc\text{---} \quad \text{with} \quad \delta_{abcd} = \frac{a}{d} \frac{b}{c} . \quad (2.6)$$

The Kronecker δ fixes all the four indices to have the same value. Each δ_{abcd} tensor represents a spin that is summed over, and the W matrices connect nearest-neighbor spins. From this point on, we keep drawing such networks without explicitly showing the periodic boundary conditions, but they are always implicitly assumed.¹

To transform this into a network with one tensor per lattice site, we can decompose W as

$$W = MM^T \quad \text{with} \quad M = \begin{pmatrix} \sqrt{\cosh \beta} & \sqrt{\sinh \beta} \\ \sqrt{\cosh \beta} & -\sqrt{\sinh \beta} \end{pmatrix} \quad (2.7)$$

for which the graphical representation is

$$\text{---}\bigcirc\text{---} = \text{---}\bigcirc\text{---} . \quad (2.8)$$

¹Other kinds of boundary conditions are just as easy to incorporate into the tensor network description, as we will see in Chapter 3.

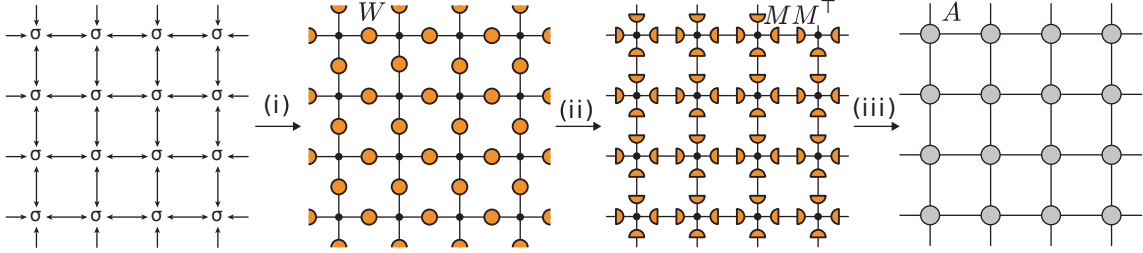


Figure 2.1: Graphical depiction of the construction of the tensor network associated to a given classical Hamiltonian. To each vertex of the square lattice is associated a configuration variable. We can associate (i) to each edge of the lattice a local Boltzmann weight W which is a function of its neighboring configuration variables. In step (ii), we decompose each W into two matrices M and M^\top . The contraction (iii) of such four matrices define the initial tensor A of the tensor network.

We then define the tensor

$$A_{ijkl} = \begin{array}{c} i \\ \circ \\ j \\ k \\ l \end{array} = \begin{array}{c} \text{---} \circ \text{---} \\ | \\ \text{---} \circ \text{---} \\ | \\ \text{---} \circ \text{---} \\ | \\ \text{---} \circ \text{---} \end{array} \quad (2.9)$$

$$= \sum_{a,b,c,d} \delta_{abcd} M_{ai} M_{bj} M_{ck} M_{dl} . \quad (2.10)$$

Using this A_{ijkl} , we can rewrite the partition function as the contraction of a homogeneous tensor network. In the case of a 4×4 lattice, these steps are summarized Fig. 2.1.

So far we have been focusing on the case of 2D classical statistical systems. However, the same kind of tensor networks arise when trying to represent ground or thermal states of quantum Hamiltonians H in 1D. More precisely, using a Suzuki–Trotter decomposition [92, 93] of the Hamiltonian H , the Euclidean path integral $e^{-\beta H}$ can be written as a network of the same form as the one in Fig. 2.1, but with open boundaries at the top and the bottom. For more details, see for instance Refs. [47, 61]. In this chapter we focus mostly on 2D classical models, but everything we do could also be applied to 1+1D quantum systems, as explained in Sect. 2.8.

2.2.2 Tensor network coarse-graining

In order to study a statistical model, such as the square lattice Ising model, we are interested in computing the partition function Z for a lattice of size $L \times L$, with L large, using the tensor network decomposition in Fig. 2.1. However, the computational cost of contracting

such a tensor network is exponential in L , and hence only small values of L can be accessed. Numerous tensor network algorithms exist for doing the contraction approximately, but in polynomial time. They often rely on the philosophy of the real-space renormalization group, where some local patch of tensors is contracted together to create a coarse-grained tensor, describing physics at a larger length scale. To keep the bond dimension from growing in this process, and thus avoiding the exponential growth of computation time, a local replacement is done, where a single tensor (or a patch of tensors) is replaced with others, lowering the bond dimension in the process. The fact that such a replacement can be done with only a small error relies on the requirement that some of the elements in the tensors only describe short-range physics, i.e., are irrelevant when moving to a coarser lattice. Such a tensor network coarse-graining can be iterated until the infrared length scale is reached, where the whole system to be studied consists of only a few sites. Such a process yields an RG flow in the space of tensors.

The first tensor network coarse-graining algorithm of this kind was TRG [68]. It is a highly successful algorithm, which is simple to implement and efficient at obtaining accurate observables for 2D classical lattice models or ground and thermal states of 1+1D quantum lattice models, especially in gapped phases. The TRG algorithm is summarized in Fig. 2.2, and further details can be found in the original paper [68]. At the heart of the algorithm is the step where tensors are replaced with their truncated singular value decomposition,² where the truncation ensures that the bond dimensions do not grow unmanageably high.³

In spite of its success as a numerical tool, it has been known for several years that TRG does not implement a proper RG transformation on the lattice [37]. The TRG coarse-graining transformation removes some short-range details from the tensors, but not all, and hence the coarse-grained tensors are polluted with details about UV physics. Because of this, the fixed point tensor reached at the end of the RG flow depends on non-universal features, such as the exact temperature. Why this happens is well illustrated by the so-called the corner-double-line (CDL) tensors, a toy model for extremely short-range physics [47, 67]. For instance, a four-valent CDL-tensor is obtained as the tensor product of four (arbitrarily chosen) matrices $M := \text{---}\circ\text{---}$ such that

$$A^{\text{CDL}} := \text{---}\circ\text{---} \quad (2.11)$$

²Singular value decomposition of a matrix M is given by $M = USV^\dagger$, where U and V are unitary matrices and S is a diagonal matrix. Its entries $S_{ij} = S_i\delta_{ij} \geq 0$ are referred to as singular values. The optimal [28] low-rank approximation of M is then given by $M_{ij} \approx \sum_{k=1}^\chi U_{ik}S_kV_{jk}^*$, where $\chi < \text{rank } M$ is the new rank.

³We discuss a variation of TRG known as HOTRG [110], based on higher-order singular value decomposition, in Sects. 2.7 and 2.8.

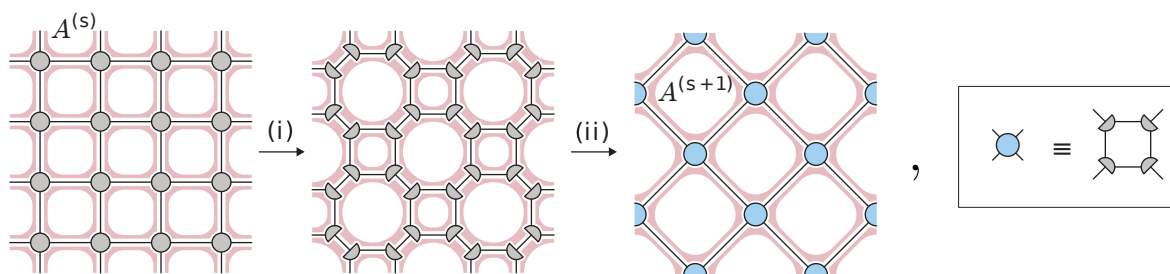


Figure 2.2: Single iteration of the TRG algorithm, producing coarse-grained tensor $A^{(s+1)}$ starting from tensor $A^{(s)}$. In step (i), the $A^{(s)}$ tensors of the homogeneous network are split via a truncated SVD along two different diagonals. In step (ii), sets of four tensors are contracted together to form $A^{(s+1)}$. This results in a new square lattice, rotated by 45° and with the lattice spacing multiplied $\sqrt{2}$. The same two steps can be repeated to return the lattice to the original orientation, now with the lattice spacing doubled. The red shading represents short-range correlations of the CDL-type, introduced in Fig. 2.3. Half of the red loops are captured in the contraction of $A^{(s+1)}$, but the rest remain, and become nearest-neighbor correlations of the coarse-grained tensors. This violates the principle of RG, that the coarse-grained description of physics should not include UV details. A detailed analysis of how CDL-tensors are a fixed point of the TRG transformation, to complement the more schematic picture here, is given in App. A.1.1.

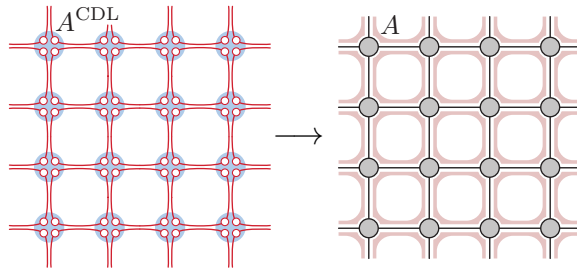


Figure 2.3: Lattice of A^{CDL} tensors forms closed loops within plaquettes. Any observable inserted on a leg would be highly correlated with other observables around the same plaquette, but entirely uncorrelated with anything further away. Thus the CDL model is a toy model for purely short-range physics, and having it flow to a trivial RG fixed point is a common test for tensor network RG algorithms. In this chapter we often accompany tensor network diagrams with the red shading shown here, that schematically represents loops of CDL-type correlations. We use it to illustrate how CDL-tensors behave under different algorithms. A more detailed and rigorous analysis of how the algorithms discussed in this chapter apply to CDL-tensors is given in App. A.1.

$$\begin{aligned}
 \text{and } A_{ijkl}^{\text{CDL}} &\equiv A_{(i_1 i_2)(j_1 j_2)(k_1 k_2)(l_1 l_2)}^{\text{CDL}} \\
 &= M_{i_1 j_2} M_{j_1 k_2} M_{k_1 l_2} M_{l_1 i_2}
 \end{aligned}
 \tag{2.12}$$

where the blue shading represents the A^{CDL} tensor whose internal internal structure is explicitly shown in red and white. Note that if the legs of A^{CDL} are of bond dimension χ , then the CDL-matrices M are $\sqrt{\chi} \times \sqrt{\chi}$. Fig. 2.3 displays a 4×4 square lattice of such CDL-tensors. Even though the CDL model is entirely trivial at length scales larger than the lattice spacing, it is an RG fixed point of the TRG transformation, as schematically illustrated in Fig. 2.2, and explained in detail in App. A.1.1.

The failure of TRG to produce proper RG flows is a conceptual shortcoming of the algorithm. Numerically, in 2D the remaining UV details in the tensors are a nuisance, that puts a strain on the bond dimension and thus makes computations somewhat slower or less accurate. This does not undermine usefulness of the algorithm though, and TRG remains a very potent tool in 2D. However, in higher dimensions, and to a lesser degree at critical points, the same problem emerges in a much worse form, and causes an exponential growth in the bond dimension. This is often called accumulation of local, or short-range, correlations [37]. Ultimately this issue stems from the area law of entanglement, which in gapped 1+1D quantum systems gives a constant contribution regardless of the length scale, but in higher dimensions grows as one considers larger and larger coarse-graining

blocks. This connection to the area law and the rampant growth of these UV remnants in 3D classical/2+1D quantum systems is discussed in Sect. 2.7.

To solve this shortcoming of TRG, several more advanced tensor network renormalization group algorithms have been designed. Tensor Network Renormalization or TNR [37], Loop-TNR [113] and TNR+ [3] introduce more complicated local replacements and optimizations, and manage to remove all short-range details during the RG transformation. This is exemplified by their treatment of the CDL-model, that is coarse-grained to trivial tensors of bond dimension 1. They yield proper RG flows with correct fixed point structures, and produce more accurate results than TRG when the same bond dimensions are used, but come with higher computational costs. In principle all of these algorithms generalize to higher dimensions as well. However, in practice, designing the details of the implementation in higher dimensions is far from trivial, and most importantly, the computational cost tends to be prohibitively high. Consequently, the only algorithm for which a concrete proposal for a generalization to higher dimensions exists is HOTRG [110], a variant of TRG. For the algorithms that deal with all local correlations and produce proper RG flows, no concrete proposals for generalizations to 3D have thus been put forth.

2.3 Environment spectrum

Numerous tensor network algorithms are based on performing local replacements in the network: A part of the network is replaced with something else, in such a way that the network as a whole is not affected. Moreover, typically the network that is kept invariant, which we call here T , is in fact a local neighborhood of the global network:



For instance, in TRG [68] a tensor is replaced by its truncated singular value decomposition, and in TNR [37] a plaquette of tensors is replaced with the same plaquette, but now surrounded by a number of isometric and unitary tensors.

Typically a small error is caused when performing such a replacement, so that the value of T remains only approximately the same. Since the purpose of most such replacements is to truncate, or lower the dimension of, a bond in the network, this error is usually called a

“truncation error”. Tensor network algorithms are therefore characterized by the kind of replacements they perform, as well as the optimization methods used in order to minimize the truncation error caused by such replacements.

Underlying all these algorithms is, however, the same question: What can we change about a local patch of a network, without affecting its neighborhood? We propose a general answer to this question. Consider a tensor network T (the neighborhood) and some subnetwork of it, R (the local patch):

$$T = \begin{array}{c} \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \end{array} = \begin{array}{c} \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \end{array} = \begin{array}{c} \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \quad \circ \end{array} . \quad (2.14)$$

Assume we want to make changes to R , without affecting T , and would like to know which changes are allowed. For this purpose, define E to be the network obtained by removing R from T ,

$$E = \begin{array}{c} \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \end{array} . \quad (2.15)$$

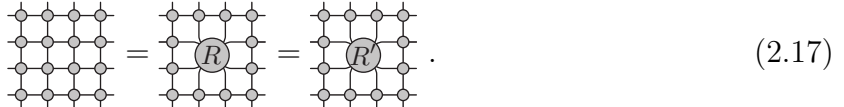
We call E the environment of R in T . Define \mathbb{V}_{ext} to be the vector space of all the external, open legs of T (the tensor product of the vector spaces of the individual legs), excluding any possible external legs in R , and \mathbb{V}_R to be the vector space of the legs that connect R to E . Now consider E as a linear map from \mathbb{V}_R to \mathbb{V}_{ext} . In other words, think of E as a matrix, where the legs with ingoing arrows in the following figure are grouped together to form one matrix index, and the legs with outgoing arrows are grouped together to form the other. Now perform a singular value decomposition of E as such a matrix, yielding $E = USV^\dagger$,

$$\begin{array}{c} \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \\ \circ \quad \circ \quad \circ \end{array} = \begin{array}{c} \uparrow \uparrow \uparrow \\ \uparrow \uparrow \uparrow \\ \uparrow \uparrow \uparrow \\ \downarrow \downarrow \downarrow \\ \downarrow \downarrow \downarrow \\ \downarrow \downarrow \downarrow \end{array} \stackrel{\text{svd}}{=} \begin{array}{c} \uparrow \uparrow \uparrow \\ \uparrow \uparrow \uparrow \\ \uparrow \uparrow \uparrow \\ \downarrow \downarrow \downarrow \\ \downarrow \downarrow \downarrow \\ \downarrow \downarrow \downarrow \end{array} . \quad (2.16)$$

We call the singular values S_i the *environment spectrum* of R with respect to T . It quantifies what is referred to as local correlations of the network: It tells us to what extent R can affect the external legs of T , and to what extent it only affects physics internal, or local, to T . Examples of environment spectra for physical models are shown in Sect. 2.7, in Fig. 2.7.

To clarify, consider a case where there are singular values in S that are equal to zero. The corresponding singular vectors in U span the kernel of E , i.e., the subspace of \mathbb{V}_R that

is mapped to the zero vector by E . Any components that R may have in this subspace are therefore irrelevant when R is contracted with E to form T . This means that as long as we replace R with something else, R' , so that $R - R'$ stays in this kernel, we know that



$$\text{Diagram} = \text{Diagram} = \text{Diagram} \cdot \quad (2.17)$$

Conversely, if $R - R'$ is not contained in the kernel of E , then we know that $T = ER \neq ER'$. Thus we have a full characterization of the local replacements of R in T that can be done without affecting T . In practice, the smallest singular values in S are usually only approximately zero and such a local replacement of R causes a small error.

In the case where every tensor in T represents Boltzmann weights for some Hamiltonian, each of the legs is some degree of freedom of the system. From this point of view, if there are small values in the environment spectrum, then some subspace of the configuration space \mathbb{V}_R is irrelevant for describing the physics on the external legs \mathbb{V}_{ext} . Thus some degrees of freedom can safely be discarded without affecting the physics as observed on the scale of T . Connections to renormalization group ideas can begin to be seen here, and are made clear later.

Although the above procedure to find the environment spectrum can in principle be applied to any subnetwork R , we find that its importance and usefulness are clearest when R is a single tensor, a set of legs, or even a single leg. The latter case is the focus of the next section.

2.4 Truncating bonds using the environment spectrum

In this section, we present how the environment spectrum can be used in order to define a general strategy to truncate bonds in a given network. First, we briefly show how the SVD decomposition is a special case of such a strategy. Then we consider a more general case, and show how the environment spectrum approach implements a structure preserving truncation [32, 114].

2.4.1 Truncated singular value decomposition

The step at the heart of the TRG algorithm consists in replacing a tensor by its truncated SVD. The SVD in question is

$$\text{---} \circ \text{---} \stackrel{\text{svd}}{=} \begin{array}{c} U \\ | \\ \text{---} \\ | \\ S \\ | \\ \text{---} \\ | \\ \tilde{V}^\dagger \end{array} . \quad (2.18)$$

An approximate decomposition is then obtained by truncating this SVD as $\tilde{U}\tilde{S}\tilde{V}^\dagger$. Here \tilde{S} is a diagonal matrix with the χ' largest singular values from S , and \tilde{U} and \tilde{V} contain the corresponding singular vectors.

This approximate decomposition can be expressed using the environment spectrum, by considering the tensor to be decomposed as the network T , and the left and top legs together as the subnetwork R :

$$T = \begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \quad \text{with} \quad R = \text{---} . \quad (2.19)$$

In other words, R is the tensor product of two identity matrices. The environment E is just T , since cutting away identity matrices from open legs does nothing, and the environment spectrum is just the singular value spectrum S . Furthermore, the truncation of the SVD can be rephrased as replacing R with the projector

$$R' = \begin{array}{c} \chi \\ \diagup \\ \text{---} \\ \diagdown \\ \chi' \\ \diagup \\ \chi \\ \diagdown \end{array} \equiv \tilde{U}\tilde{U}^\dagger . \quad (2.20)$$

This R' simply projects out the subspace to which the environment spectrum S gives the lowest weight. Replacing R by R' can be seen to be equivalent with the truncated SVD as follows.

$$\begin{array}{c} \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \stackrel{R \mapsto R'}{\approx} \begin{array}{c} \tilde{U} \\ | \\ \text{---} \\ | \\ \tilde{U}^\dagger \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \text{---} \end{array} \stackrel{(2.18)}{=} \begin{array}{c} \tilde{U} \\ | \\ \text{---} \\ | \\ \tilde{S} \\ | \\ \text{---} \\ | \\ \tilde{V}^\dagger \end{array} = \begin{array}{c} \tilde{U} \\ | \\ \text{---} \\ | \\ \text{---} \\ | \\ \tilde{V}^\dagger \end{array} . \quad (2.21)$$

In the same way, the truncated SVD of any tensor can be formulated in terms of replacing legs R with a projector R' , that projects out the vanishing part of the environment spectrum. This is of course only an unnecessarily complicated rephrasing of a well-known procedure. In the next section, we reveal the genuine usefulness of the environment spectrum as it can be used to truncate a single leg in a general setting.

2.4.2 Gilt: Graph independent local truncation

Consider again a tensor network T and a leg R in it that we want to truncate. As before, think of R as a subnetwork in the sense of Sect. 2.3. However, this time allow for R to be any leg in T , including any of the contracted, internal legs. Although T can be an arbitrary tensor network, for concreteness we focus on the case of a square plaquette:

$$T = \begin{array}{c} \diagup \quad R \quad \diagdown \\ | \quad \quad | \\ \diagdown \quad \quad \diagup \end{array} . \quad (2.22)$$

As we explain later, by truncating the leg R using the Gilt method, we can remove for instance a loop of CDL correlations within such a plaquette, and solve the issue TRG has with accumulating short-range correlations.

The environment for R in T is simply T with the leg R cut,

$$E = \begin{array}{c} \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \end{array} \equiv \begin{array}{c} \square \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \end{array} \quad (2.23)$$

and the singular value decomposition that yields the environment spectrum is the following:

$$E = \begin{array}{c} \square \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \end{array} \stackrel{\text{svd}}{=} \begin{array}{c} U \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \\ S \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \\ V^\dagger \end{array} . \quad (2.24)$$

Note that if the bond dimension of R is χ , then S has χ^2 elements in it.

As before, the spectrum S is telling us which part of the vector space of R is important only for physics strictly internal to the plaquette T . We use this information to truncate the leg R as follows.

Take the leg R and perform on it the following change of basis:

$$R = \text{---} = \int \text{---} = \begin{array}{c} U \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \\ U^\dagger \end{array} = \begin{array}{c} t \\ \diagup \quad \diagdown \\ | \quad | \\ \diagdown \quad \diagup \\ U^\dagger \end{array} \quad (2.25)$$

where we have defined the vector t_i as

$$t_i = \text{Tr } U_i \quad \text{with} \quad U_i = \begin{array}{c} \diagup \\ | \\ \diagdown \\ i \end{array} . \quad (2.26)$$

Here we have introduced the symbol $\text{---}\textcircled{i}$ for the vector $|i\rangle$ that has its i -th component be 1 and all the others 0.

We could of course do this change of basis with any unitary, since $UU^\dagger = \mathbb{1}$, but choosing the basis of the singular vectors of E lets us immediately see how we can modify R without causing a large error. To see this, observe that

$$T = \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} \stackrel{(2.25)}{=} \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} \stackrel{(2.24)}{=} \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} = \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l}. \quad (2.27)$$

At the last step, a pair of U and U^\dagger have cancelled, and we see that the environment spectrum S , which is a diagonal matrix, is directly multiplying the elements of t . Thus if we assume that out of the χ^2 elements in S only the first D are non-zero, then changing any of the first D elements of t would result in a significant change in T . However, changing t_i for $i = D + 1, \dots, \chi^2$ has no effect on T . In other words, we can replace the original leg R , which was just the identity matrix, with a matrix

$$R' = \text{---}\textcircled{i} = \text{---}\textcircled{i} U^\dagger \quad (2.28)$$

and as long as $t'_i = t_i$ for $i = 1, \dots, D$, we know that

$$\text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} \approx \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} = T. \quad (2.29)$$

Here the approximation in the first equation arises from the fact that in reality smallest elements in S are only approximately zero. The remaining elements t'_i , $i = D + 1, \dots, \chi^2$ we are free to choose as we wish, because they provide weights for those contributions in R' that are in the kernel of E .

To truncate the leg we are working on, we would like to use this freedom in t' to make the matrix R' have as low rank χ' as possible. We could then singular value decompose R' with only χ' singular values, and multiply the parts of the decomposition into the neighboring tensors in the environment:

$$\text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} \stackrel{(2.29)}{\approx} \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} \stackrel{\text{svd}}{=} \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l} = \text{---}\textcircled{i} \text{---}\textcircled{j} \text{---}\textcircled{k} \text{---}\textcircled{l}. \quad (2.30)$$

This way we would have performed a structure preserving truncation of the leg R from dimension χ down to χ' .

So how do we choose t' to minimize the rank of R' ? Several approaches for this optimization are possible, and unfortunately we do not know of a general algorithm to make an optimal choice. After trying several approaches, we have settled to using a way that (a) is fast, (b) is optimal according to a cost function that indirectly favors choices of R' that have low rank, (c) produces good results when used in a renormalization group algorithm.

We explain the details of the cost function and how to optimize t' for it in App. A.2. The final result is

$$t'_i = t_i \frac{S_i^2}{\epsilon^2 + S_i^2}. \quad (2.31)$$

Here ϵ is a parameter that sets the scale in the environment spectrum S , such that any values smaller than ϵ are considered to be close enough to zero to allow changing the corresponding t'_i . The value of ϵ is chosen by the user. The larger it is, the more the algorithm will truncate, but causing a larger truncation error.

In summary, we have have designed an algorithm to truncate a leg R in a given environment E , that we call *Gilt*. It consists of the following steps:

1. Singular value decompose the environment E to obtain the unitary U and the environment spectrum S .⁴
2. Compute the traces $t_i = \text{Tr } U_i$.
3. Set the vector t' as in (2.31).
4. Compute the matrix $R' = \sum_{i=1}^{\chi^2} t'_i U_i^\dagger$.
5. Singular value decompose R' as $R' = usv^\dagger = (u\sqrt{s})(\sqrt{s}v^\dagger)$, and multiply the matrices $u\sqrt{s}$ and $\sqrt{s}v^\dagger$ into the neighboring tensors as in (2.30). The rank of this singular value decomposition determines the new bond dimension χ' .

A pictorial summary of this algorithm is in (2.30).

Often applying Gilt once does not yet lead to a significant reduction in the bond dimension. However, it can be applied repeatedly on the same leg, and this procedure quickly converges so that further attempts to truncate yield $R' = R$, meaning no further progress

⁴Since we only need U and S , instead of singular value decomposing $E = USV^\dagger$, we can eigenvalue decompose the Hermitian EE^\dagger as $EE^\dagger = US^2U^\dagger$. This is computationally much cheaper, and reduces the cost of doing this for the square plaquette to $\mathcal{O}(\chi^6)$.

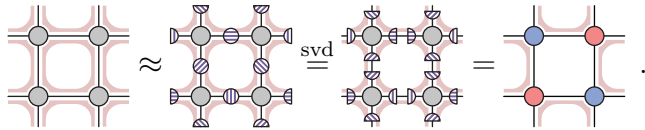
is possible. These repeated iterations do not significantly increase the computational cost, since the most time-consuming part, the SVD of the environment, needs to be performed only once.⁵

Finally, we come back to the CDL toy model for local correlations. If there is a CDL-loop within the plaquette T , then by truncating R using Gilt we can cut this loop, and by applying such a truncation to all the legs around the plaquette we can completely remove it. The details of how this happens when Gilt is applied to CDL-tensors is explained in App. A.1.2. That discussion also clarifies why several iterations of Gilt on the same leg are often required.

2.5 Gilt-TNR

The Gilt algorithm described in the previous section can be used in various ways as a part of different tensor network schemes. Here we use it to fix the problem TRG had (see Sect. 2.2) with accumulating short-range correlations.

Recall that the issue with TRG was that it only properly dealt with local correlations around every other plaquette (see Fig. 2.2). This can be easily fixed by preceding each TRG step with a step where Gilt is applied to the problematic plaquettes. Matrices R' can be created on all the legs surrounding a plaquette, using this plaquette as the neighborhood T . Note that these matrices need to be created and applied in serial, not parallel, since each one modifies the environment for the others. They truncate away any details internal to the plaquette, by modifying the tensors at the corners:



We call this combination of Gilt and TRG *Gilt-TNR*. One complete iteration of it is shown in Fig. 2.4. It can be seen to properly remove all short-range details, and as is proven by the results shown in Sect. 2.6, Gilt-TNR is indeed a proper RG transformation, with the correct structure of universal fixed points. Here, the red shadings illustrate how the removal of UV details happens, but a more rigorous discussion of how Gilt-TNR deals with the

⁵See the source code for details.

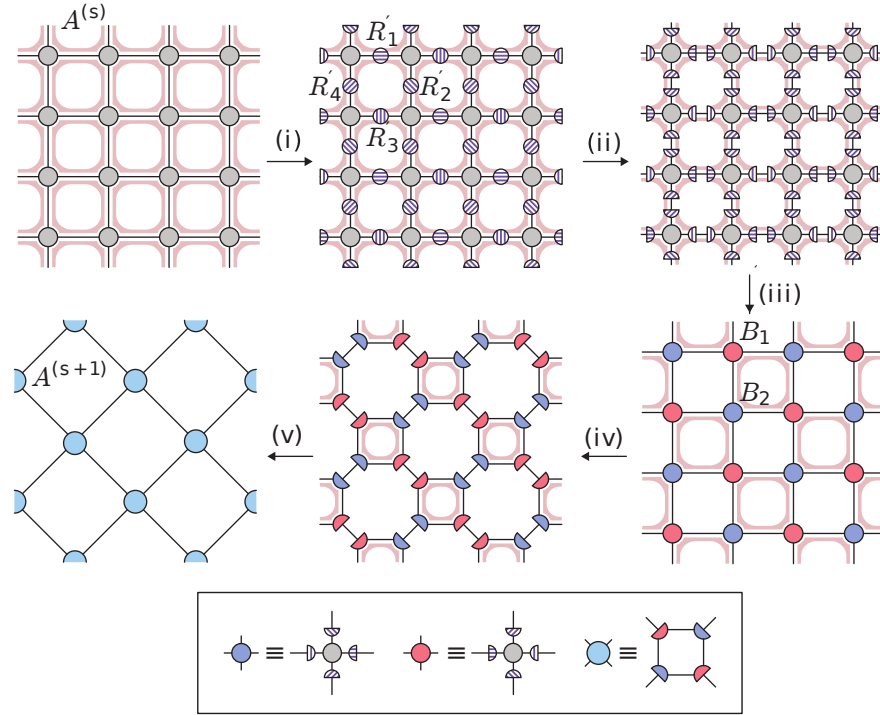


Figure 2.4: Single iteration of Gilt-TNR. In step (i), four matrices R'_1, \dots, R'_4 are inserted between neighboring tensors, using Gilt. These matrices are factorized in step (ii) via an SVD, and the pieces are absorbed into the neighboring tensors in step (iii). This results in a checkerboard network with two kinds of tensors which then undergoes a regular TRG iteration, depicted in steps (iv) and (v). It shows in particular that the TRG iteration restores the homogeneity of the network. As before, the red shading represents short-range correlations which behave like CDL-loops. By the end of the coarse-graining step all such correlations have been removed. Note that in step (iii) short-range correlations are only removed from around every other plaquette, since these were the plaquettes that were used as the neighborhood T when creating R'_1, \dots, R'_4 .

CDL-model can be found in App. A.1.3. Note that this combination of performing local replacements on single legs with TRG is of the same form as the one proposed in Ref. [114].

By fixing TRG's issue with short-range details, Gilt-TNR provides another way of solving the same problem that TNR, Loop-TNR and TNR+ solve. As shown in Sect. 2.6, Gilt-TNR can produce results competitive with the best achieved with these other algorithms. The leading order in its computational cost is $\mathcal{O}(\chi^6)$, the same as for TNR and Loop-TNR. The

bottle necks are the truncated SVD in TRG⁶ and the singular value decomposition that yields the environment spectrum of a plaquette. Subtleties and caveats exist in comparing the performance of different algorithms, that are discussed in Sect. 2.6.

We consider the main advantages of Gilt-TNR over the other algorithms to be its simplicity and generalizability: Implementing Gilt-TNR requires only adding a relatively simple, additional step to TRG, and a minimal working implementation takes a mere hundred lines of code (see App. A.3). Unlike any other proper tensor network RG algorithm, Gilt-TNR does not require an iterative optimization procedure, which would require an initial guess and could suffer from varying speed of convergence or getting stuck in local minima. Moreover, applying Gilt-TNR to lattices other than the square lattice is a matter of simply changing the neighborhood T that is used for the Gilt step (on a hexagonal lattice for instance, T would naturally consist of a single hexagon), and choosing a way of putting tensors together to move to the next length scale. This is in stark contrast especially to TNR, where adapting it to different lattices requires significant redesigning of the algorithm. Moreover, the generalization of Gilt to a cubical lattice in 3D (as well as many other lattices) has a remarkably low computational cost, only slightly more expensive than HOTRG. We discuss the case of a cubical lattice in Sect. 2.7.

Note that, because the truncation procedure which removes short-range correlations preserves the graph and is independent of the coarse-graining step, there are many other possible ways of combining Gilt and TRG, or other tensor network coarse-graining algorithms. They would presumably also yield proper RG flows. Some possibilities include applying Gilt to all plaquettes instead of just half of them, or considering larger neighborhoods, such as ones consisting of two neighboring plaquettes, in the Gilt procedure. Our method of optimizing for the R' matrices could also be combined with that of the TNS algorithm [114]. We have chosen the implementation here because it is simple, faster than some other options, and yields accurate results.

2.6 Benchmark results

To illustrate the efficiency of the RG algorithm we explained in the previous section, we present benchmark results for the 2D classical Ising model.

In Fig. 2.5, we show the error in the free energy at the critical point as a function of the bond dimension χ , comparing plain TRG and our Gilt-TNR algorithm. Running times of the two algorithms are also compared. Note that the TRG results have been obtained with

⁶ Note that a $\mathcal{O}(\chi^5)$ implementation of TRG is also possible [26].

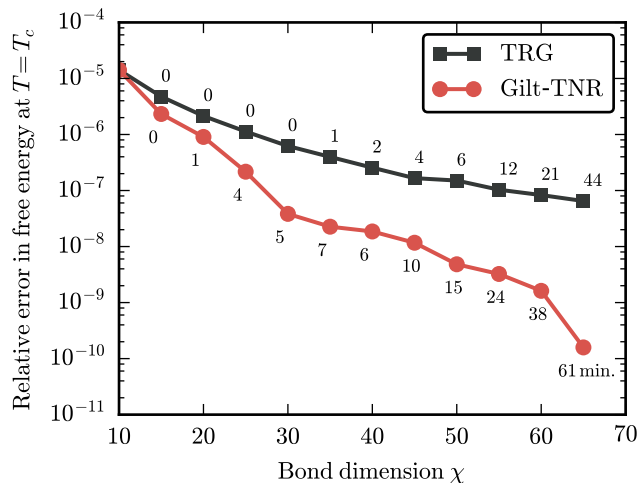


Figure 2.5: Error in free energy of the critical 2D classical Ising model at different bond dimensions, for TRG and for Gilt-TNR. The numbers next to the data points are total running times in minutes, for the simulation consisting of 25 iterations of the algorithm. The exact running times of course depend on hardware and implementation details, but worth noting is the relatively small difference between the TRG and Gilt-TNR algorithms. Even though adding Gilt into the algorithm slows it down a bit, this is more than compensated for in the quality of the results. For the Gilt-TNR results shown here, the parameter ϵ has been chosen to be $8 \cdot 10^{-7}$. Note that this is not the optimal choice of ϵ for this whole range of χ . Instead, one should vary ϵ as one varies χ , making it smaller as χ grows. It is only for simplicity of presentation that we have chosen to stick to a single value of ϵ that performs well over the whole range of χ 's shown.

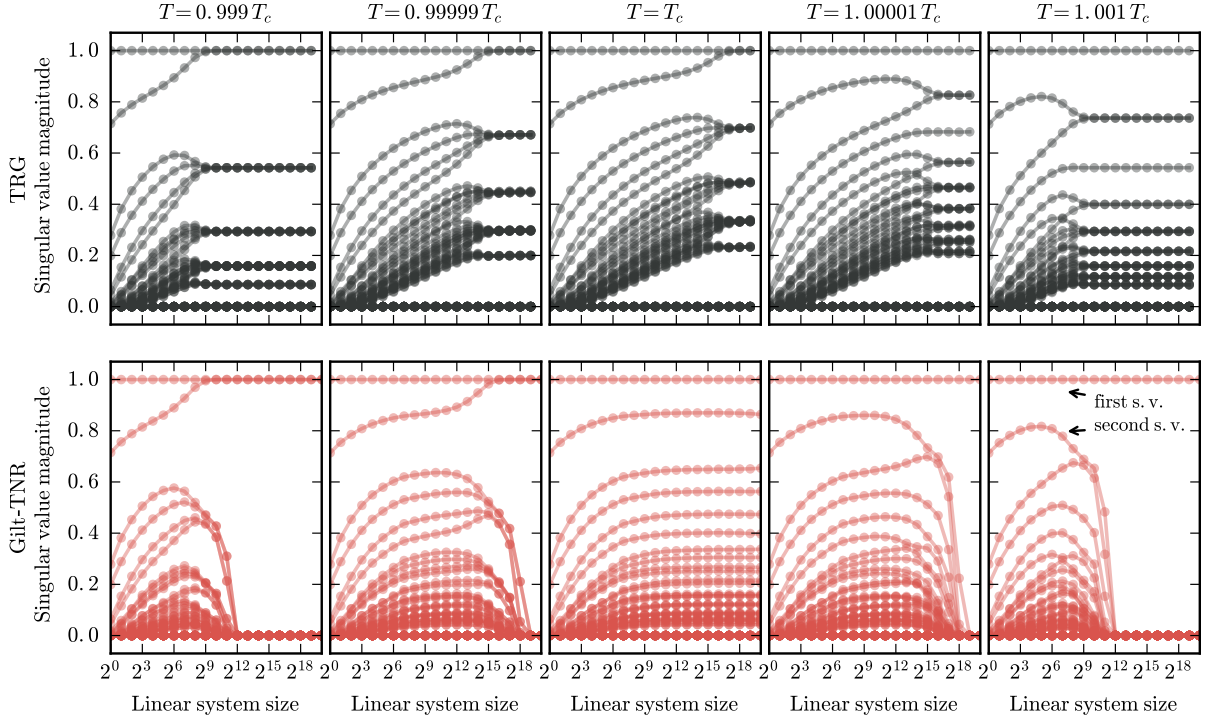


Figure 2.6: RG flow of the coarse-grained tensors, illustrated for TRG (top row) and Gilt-TNR (bottom row) for five different temperatures. The horizontal axis is the linear system size, or in other words the number of RG transformations applied. At each system size, the data points are the 60 largest singular values of the coarse-grained tensor, with the same decomposition as that shown in (2.18). Thus each of the lines follows the development of one of the singular values along the RG flow. These singular values provide a rough, basis independent characterization of the structure of the tensor. Note how, for TRG, the spectrum is different at every temperature, even at the end of RG flow, when a fixed point has been reached. In contrast, for Gilt-TNR, on both sides of the critical point the RG flow ends in a trivial fixed point characteristic of that phase, with either one or two dominant singular values. At the critical point a complex fixed point structure is found, that comes from the CFT. This critical fixed point is maintained over several orders of magnitude in linear system size. These results were obtained with $\chi = 110$ for both TRG and Gilt-TNR, and $\epsilon = 5 \cdot 10^{-8}$ for Gilt-TNR.

the same code, by simply turning off the Gilt algorithm. In these results, Gilt is seen to improve accuracy by up to three orders of magnitude for the same bond dimension χ , with

only a moderate increase in running time. The results, which are all achievable in a couple of hours on a laptop, reach down to a relative error of 10^{-10} , which is comparable with the best results achieved with other tensor network algorithms [37, 113].

At this point, let us remark on comparing Gilt-TNR to other algorithms in the literature. First of all, since Gilt-TNR builds on top of TRG, a fair comparison can be made by simply switching on and off the additional Gilt performed in between coarse-graining steps of TRG. In this setting, we find that Gilt-TNR consistently outperforms TRG by a large margin in terms of the accuracy of physical observables.

A much more interesting comparison, however, would be to other algorithms that implement proper RG transformations, such as TNR [37] and Loop-TNR [113]. Although their asymptotic computational complexity is the same as that of Gilt-TNR, namely $\mathcal{O}(\chi^6)$, actual computational times can vary drastically, as both TNR and Loop-TNR include iterative optimization procedures, where thousands of iterations may be necessary to reach convergence. No such optimization is necessary for Gilt-TNR, which, for the same bond dimension, makes it significantly faster to run in practice. However, at the same bond dimension the other two algorithms produce more accurate results, which exemplifies the usual trade-off between speed and accuracy.⁷ Since a robust comparison of Gilt-TNR to these algorithms would depend on a specific implementation of each scheme and vary from machine to machine, we attempt no such benchmark.⁸ Instead, we present published data for TNR [37] and Loop-TNR [113] alongside our results to demonstrate that we obtain results of comparable accuracy with modest computational effort.⁹

In Tab. 2.1 we show the first few scaling dimensions of the Ising CFT, obtained by diagonalizing a transfer matrix on a cylinder/torus [10, 47], contrasted with the same numbers obtained with other algorithms. All the more advanced algorithms, that produce correct RG flows, clearly outperform TRG. Between them, similar quality of results can be achieved, with the above issues preventing fair comparison beyond this statement.

In Fig. 2.6 we show how the Gilt-TNR algorithm produces physically correct RG flows in the tensors. Shown there are the singular value spectra of the coarse-grained tensors, as they develop through repeated applications of the RG transformation. Five different

⁷Note that when we quote the bond dimension χ for Gilt-TNR, this refers to the bond dimension in the TRG step of the algorithm. This dimension is further reduced by Gilt. The bond dimension Gilt truncates to is determined dynamically by the threshold ϵ , but as an example, in the run that produces the Gilt-TNR results in Fig. 2.6 at criticality, Gilt typically truncates the bond dimension from 110 to around 30.

⁸As a qualitative comparison, our implementation of TNR achieves similar results to Gilt-TNR in running time of the same order.

⁹We used the Mammouth Parallèle 2 nodes of the Calcul Québec cluster with 24 Opteron cores and 32GB of RAM.

Exact	TRG $\chi = 120$	TNR $\chi = 24$	Loop-TNR $\chi = 24$	Gilt-TNR $\chi = 120$
0.125	0.124993	0.1250004	0.12500011	0.12500015
1	1.0002	1.00009	1.000006	1.00002
1.125	1.1255	1.12492	1.124994	1.12504
1.125	1.1255	1.12510	1.125005	1.12506
2	2.002	1.9992	1.9997	2.0002
2	2.002	1.99986	2.0002	2.0002
2	2.003	2.00006	2.0003	2.0003
2	2.002	2.0017	2.0013	2.0004

Table 2.1: First few scaling dimensions of the Ising CFT, as obtained by diagonalizing a transfer matrix on a cylinder/torus [47]. In all these cases the cylinder consists of two coarse-grained sites, but the amount of coarse-graining varies. In the Gilt-TNR results a linear system size of 2^8 sites has been used, and ϵ was chosen to be $4 \cdot 10^{-9}$. We are able to reach similar quality as with TNR and Loop-TNR, with moderate computational effort (the simulation in question finished in a little less than 12 hours on the machines we use, cf. footnote 9).

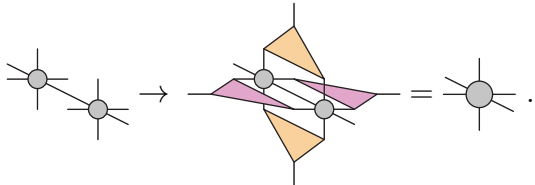
temperatures are used, and for Gilt-TNR, one can see how on both sides of the critical point the tensors flow to a simple fixed point structure with either one dominant singular value (in the high temperature, disordered phase) or two dominant singular values (in the low temperature, symmetry-breaking phase). These fixed points are the same within a phase, regardless of the exact temperature, although flowing into them takes longer (requires “zooming out” further) as one gets closer to the critical point. Such behaviour is compatible with having a second-order phase transition. At the critical point a more complex fixed point is reached, which arises from the rich structure of the conformally invariant theory. For comparison, similar spectra for TRG are shown, and there the fixed point at the end of the RG flow shows non-universal characteristics, dependent on the temperature.

In all of these results, \mathbb{Z}_2 symmetry preserving tensors, as described in Refs. [87, 88], have been used to speed up the computations. We have also used the algorithm from Ref. [81] to find efficient contraction sequences of tensor networks we use.

2.7 Gilt-TNR in 3D

Let us now consider a cubical lattice with a classical configuration variable at each site, and a nearest-neighbor Hamiltonian. Applying the same procedure as for 2D systems described in Sect. 2.2, we obtain a tensor network representation of the classical partition function of a 3D classical lattice model. The idea of applying the philosophy of RG to implement efficient algorithms to contract these networks is also equally valid in higher dimensions. However, due to the larger number of legs of the tensors and more complicated connectivity of the network, the computational cost in higher dimensions is starkly higher than in 2D.

The only computationally viable algorithm, that we are aware of, for contracting networks on a cubical lattice is the Higher-Order Tensor Renormalization Group, or HOTRG [110]. It is a variant of the TRG algorithm and is based on repeated truncated SVDs, which together amount to what is known as a higher-order SVD, hence the name. One iteration of the HOTRG algorithm consist of performing three coarse-graining steps, each one being along a different spatial direction. One such coarse-graining consists of contracting two neighboring tensors via four isometries, which are found with a higher-order SVD:


(2.32)

The cost of contracting the network above is $\mathcal{O}(\chi^{11})$, with χ the bond dimension. Recall that the leading order for coarse-graining a 2D network using TRG is $\mathcal{O}(\chi^6)$, which illustrates the increase of computational cost in higher dimensions. Furthermore, as for TRG in 2D, HOTRG removes some, but not all, short-range details during the coarse-graining.

However, the problem of some UV details “leaking” into the coarse-grained tensors is far more serious in 3D than it was in 2D. This is essentially a consequence of the area law of entanglement. For 2+1D quantum states, this law states that a block of size $L \times L$ has an amount of local entanglement between it and the rest of the lattice that is proportional to L (note that in 2D this amount is a constant, instead). This local entanglement translates for classical systems into the kind of local correlations discussed in Sect. 2.2. As one keeps coarse-graining, and L grows, if these local correlations are not properly removed, they accumulate, forcing either an exponential growth in the bond dimension or an explosion of the truncation error.

We explained in Sect. 2.2 that in 2D this mechanism can be understood using the CDL toy model. Similarly, we can introduce a generalization of the CDL-model to 3D in order

to appreciate the accumulation of local correlations under coarse-graining. In 2D, the CDL-tensors consist of the tensor product of four matrices, one for each corner. When organized in a square, they give rise to a loop of correlations. The 3D generalization has instead three-valent tensors $M_{ijk} = \text{---}\text{---}\text{---}$, one for each 3D corner. When organized in a cube, these tensors give rise to a closed network of correlations within the cube, that we illustrate below with a sphere.



Repeating the analysis of how CDL-tensors behave under the TRG transformation in 2D (see App. A.1.1), but using the above 3D generalization of CDL and the HOTRG coarse-graining, one can see that in 3D this generalized CDL is no longer a fixed point of TRG-like algorithms, but the local correlations keep accumulating over the RG flow. Thus the failure of TRG-like algorithms to clean these UV details up during the coarse-graining is in 3D no longer only a conceptual issue, but a serious numerical obstacle. This makes the need for a proper RG algorithm even more dire.

However, as we mentioned in Sect. 2.1, so far no concrete proposals to generalize TNR, Loop-TNR or any other proper RG algorithm to 3D has existed. In principle the idea is clear, but putting together the details of the algorithm is highly non-trivial, and most schemes have computational costs that are unfeasibly high. For context, we can keep in mind the connection between 3D classical systems and 2+1D quantum systems, and conclude that the quest for a proper RG algorithm for 3D classical systems is comparable to designing a 2D MERA scheme. For 2D MERAs, the most economical implementation for the square lattice [35] has computational cost that grows as $\mathcal{O}(\chi^{16})$, which is already a great improvement over previous, more straight-forward generalizations of MERA to 2D, which scales as $\mathcal{O}(\chi^{28})$ [35]. This illustrates the difficulty of keeping the computational cost at bay in 3D/2+1D.

Gilt-TNR was specifically designed with the goal of having it generalize trivially to any network, including a cubical lattice. Just like in 2D, where we simply “cleaned up” the problematic local correlations using Gilt and then applied the usual TRG procedure, on the cubical lattice we can combine Gilt with HOTRG. The neighborhood T for Gilt, which

in 2D was a plaquette, should now be a cube of neighboring tensors:

$$T = \begin{array}{c} \textcircled{R} \\ \textcircled{} \\ \textcircled{} \\ \textcircled{} \end{array} \quad \text{and} \quad E = \begin{array}{c} \textcircled{} \\ \textcircled{} \\ \textcircled{} \\ \textcircled{} \end{array}. \quad (2.34)$$

This is because a cube is the smallest local unit that can hold within it local correlations of a general form, such as those in (2.33). The computational cost of applying Gilt to the environment in (2.34) is $\mathcal{O}(\chi^{12})$ when implemented straightforwardly, and avenues for reducing it further exist. A face of the cube can also be used as a neighborhood when applying Gilt, especially as a preliminary step, as it should already allow to remove some types of local correlations that HOTRG cannot deal with. The computational cost of such a step is only $\mathcal{O}(\chi^8)$.

In many ways Gilt-TNR is thus a promising candidate for bringing proper RG transformations to 3D tensor networks. However, significant challenges are still in sight. Most importantly, it seems that in 3D higher bond dimensions are required to reach the same level of accuracy, compared to 2D. We argue this based on differences in the environment spectra in 2D and in 3D, which are exemplified in Fig. 2.7. First, observe how the 3D spectra decay more slowly, and remain well above the 2D spectra. Recall that having small values in the environment spectrum is what gives us the freedom to perform a truncation. Thus truncating in 3D with a small error seems much harder. Second, note how the spectra change when we increase the bond dimension χ . In 2D, new values are mainly added at the end of spectrum, and with each increase in χ the tail of the spectrum sinks significantly lower. In 3D, in contrast, the spectra grow much more horizontally, with new values appearing at many scales. This indicates that with the low values of χ that can be reached, in 3D HOTRG is still truncating away very significant parts of the tensors, that describe relatively long-range physics. This signals a severe need for higher bond dimensions for HOTRG. This conclusion is further supported by the relatively strong oscillations in physical observables as bond dimension is increased for HOTRG, as shown in Refs. [50, 110].

In summary, we have strong reasons to believe that Gilt-TNR, as described above, should be able to perform proper RG transformations on 3D tensor networks. In addition, its computational cost is considerably low, when compared to other algorithms with similar aims, such as 2D MERA. However, in 3D, higher bond dimensions will be necessary to reach high accuracy physical observables, which makes optimizing both the asymptotic cost and the implementation details of any algorithm a high priority. As of the writing of this thesis, we are working on an implementation of Gilt-TNR on the cubical lattice. The source code is freely available, as described in App. A.3.

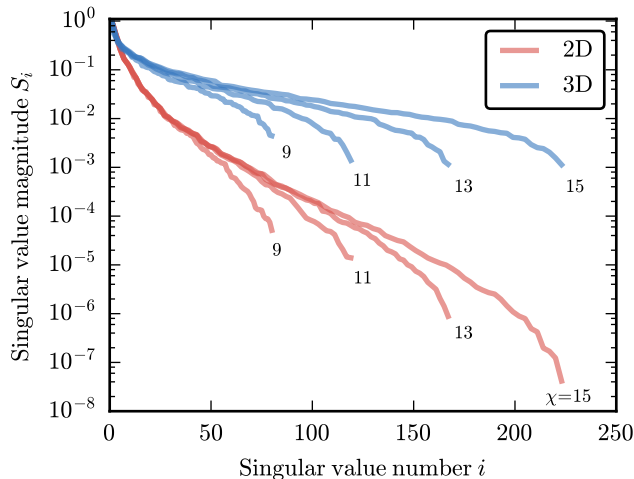


Figure 2.7: Typical environment spectra of a single leg with respect to a square plaquette in 2D (bottom four spectra) and a cube in 3D (top four spectra), each labeled with the corresponding bond dimension χ . Recall that in each spectrum, the large values correspond to parts of the vector space of the leg that are relevant for physics outside the plaquette or the cube, whereas small values signify contributions relevant only for short-range details. The spectra in 3D can be seen to decay much more slowly, indicating that larger bond dimensions are necessary, before truncations with a small error are possible. The behavior of the spectra as χ is increased, is also somewhat different in 2D and 3D. In 2D the longer spectra have more values mainly at the bottom-end, whereas in 3D new values appear almost throughout the whole spectrum. The example spectra shown here are for the Ising model, from systems that have been coarse-grained thrice. Many other choices of system sizes would yield qualitatively similar results, and the same overall difference between 2D and 3D can also be seen with the 3-state Potts model.

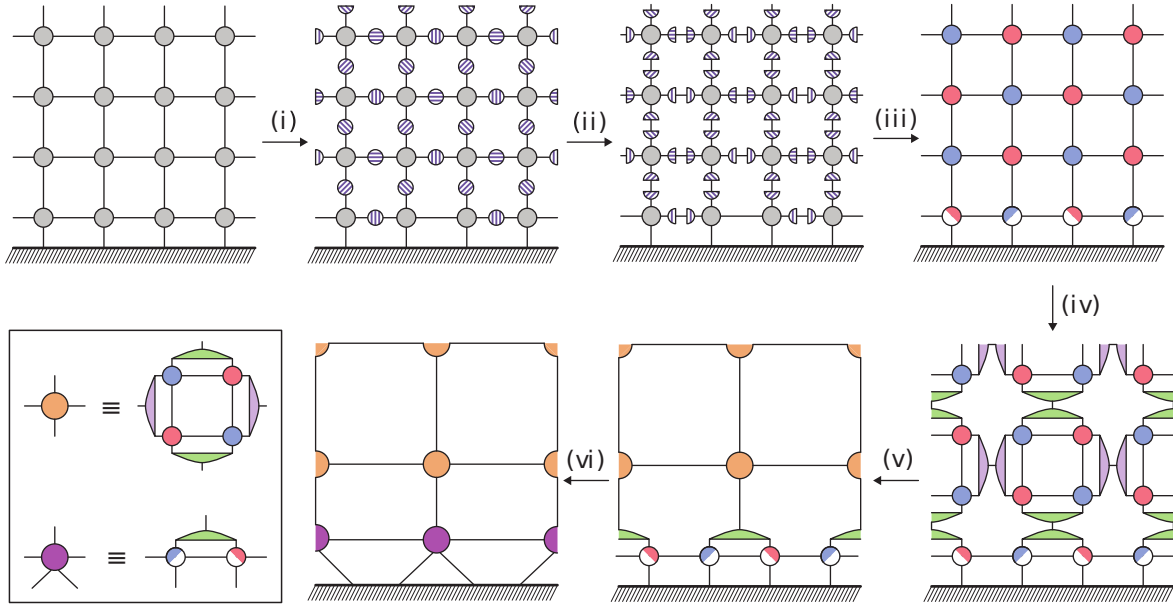


Figure 2.8: Top left panel displays the tensor network representation of the quantum state $|\psi\rangle$ on an infinite lattice covering the half-plane. The hatched strip represents an open boundary, where the boundary indices are uncontracted. In steps (i) to (v), we apply one full step of the Gilt-TNR algorithm leaving open indices untouched. Note that in the coarse-graining steps (iv) and (v) we have chosen to use a slightly different, but qualitatively equivalent, coarse-graining procedure, which more resembles 2D HOTRG [110]. A final contraction is performed in step (vi), resulting in a new homogeneous network with an additional strip of tensors. By iterating this procedure we can obtain a representation for $|\psi\rangle$ as a network of the form shown in Fig. 2.9.

2.8 Quantum states

As mentioned in Sect. 2.2, the kind of tensor networks we have been considering can be used either to represent partition functions of classical systems or ground and thermal states of quantum Hamiltonians. In this section, we explore in more detail the latter scenario.

Given a quantum Hamiltonian H , we can obtain a tensor network representation of the Euclidean path integral $e^{-\beta H}$ using a Suzuki–Trotter decomposition [92, 93]. The result of this procedure is a 2D tensor network which extends both in the space direction and the Euclidean time direction, with the height of the network being proportional to β [47, 61]. The difference between the tensor network representation of the Euclidean path integral $e^{-\beta H}$ and the tensor network representation of a classical partition function (see Fig. 2.1) is

that instead of tracing over the upper and lower boundaries, they are left uncontracted, and represent the indices of the quantum state.

As first explained in Ref. [38], when applied to the tensor network representation of the Euclidean path integral restricted to the upper-half plane, a proper renormalization scheme yields an efficient, approximate representation for the corresponding quantum ground state. The resulting network is organized in layers, each describing a length scale in the state, hence representing an RG flow in the space of wave functions. Using a horizontally infinite strip of width β , instead of the half-plane, yields a representation of a thermal state.

The most common example of such a procedure is using TNR to create a Multiscale Entanglement Renormalization Ansatz (MERA) [38] network. Since our Gilt-TNR algorithm produces a proper RG flow for classical systems, it can also yield efficient representations of quantum states. As shown in Fig. 2.8, when applying Gilt-TNR to a network representing the Euclidean path integral for a ground state $|\psi\rangle$, the RG transformation creates a layer of five-valent tensors at the boundary, that describe the short-range properties of $|\psi\rangle$, while the longer-range properties are stored in the usual coarse-grained tensors. By iterating this procedure, we obtain a representation for $|\psi\rangle$ as shown in Fig. 2.9, where the physical features of the state are organized in layers corresponding to length scale. Unlike MERA, it does not have unitarity and isometricity constraints, and thus no strict causal cones.

As mentioned before, Gilt-TNR algorithm bears a resemblance to TNS [114], with the important difference being how the truncating matrices are created. Therefore, it leads to the same kind of networks for quantum states as considered in Ref. [114]. Furthermore, the network we obtain after iterating the Gilt-TNR algorithm (Fig. 2.9) is of the same form as the ones discussed in Ref. [4] in the context of coarse-graining of transfer matrices using a Matrix Product State (MPS) representation. However, these networks come with an additional isometricity condition, which is absent in our case. In the future, we hope to study further the potential of such networks as representations of quantum systems.

2.9 Discussion

We propose a novel approach to truncating bonds in an arbitrary tensor network, based on measuring the environment spectrum of a leg in relation to its neighborhood. We call this method Gilt, which stands for graph independent local truncation. It works by inserting matrices on the legs to be truncated, and does not modify the graph of the network. Furthermore, in the process of truncating, local correlations within the neighborhood of the leg are removed. Together with a simple coarse-graining procedure such as TRG, this yields

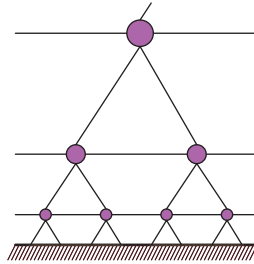


Figure 2.9: Approximate representation of a ground state of a quantum Hamiltonian H obtained by iterating the Gilt-TNR algorithm on a network for the Euclidean path integral, while leaving the indices ending at the open boundary untouched.

a proper RG transformation on tensor networks. Overall this new approach stands out due to its simplicity and flexibility. In particular, thanks to its graph independence, it could be used to implement real-space RG in higher dimensions, and is thus a suitable candidate for the study of 3D classical partition functions or 2+1D Euclidean path integrals. We apply the Gilt-TNR algorithm to the 2D classical Ising model and obtain results of comparable accuracy with other available algorithms.

It is also worth noting, that although here we have concentrated on applying Gilt in the context of coarse-graining algorithms, it is a generic method to truncate legs in any tensor network, and could have many other uses as well. Possible applications include optimizing various tensor network ansätze, such as a PEPS or a periodic MPS, and speeding up the contraction of various networks, such as expectation values of PEPS states.

Chapter 3

Topological defects

3.1 Introduction

A conformal defect is a universality class of critical behavior at the junction of two critical systems. Relevant examples include point impurities, interfaces and boundary phenomena in critical 1D quantum systems, as well as line defects, interfaces and boundaries in critical 2D classical systems [2, 25, 58, 77, 82]. A *topological* conformal defect in a conformal field theory (CFT) is a particular type of conformal defect that can be deformed without affecting the value of correlators as long as it is not taken across a field insertion [82]. It can be regarded as defining a form of twisted boundary conditions for that CFT.

The goal of this chapter is to investigate the use of tensor network techniques to describe topological conformal defects in microscopic lattice models. For simplicity, we analyze the two-dimensional critical Ising model, working mostly with the 2D classical statistical spin system, but also repeatedly connecting to the (1+1)D quantum spin chain. The critical Ising model turns out to have two non-trivial topological conformal defects: a symmetry defect D_ϵ and a duality defect D_σ [80]. The symmetry defect D_ϵ relates to the global \mathbb{Z}_2 spin-flip symmetry of the Ising model and implements antiperiodic boundary conditions, whereas the duality defect D_σ relates to the Kramers-Wannier self-duality of the critical Ising model and can be thought of as implementing some form of twisted boundary conditions.

3.1.1 Defects and transfer matrices

Consider the statistical partition function Z of the critical 2D classical Ising model on a square lattice with periodic boundary conditions (that is, on a torus), made of $m \times n$ sites.

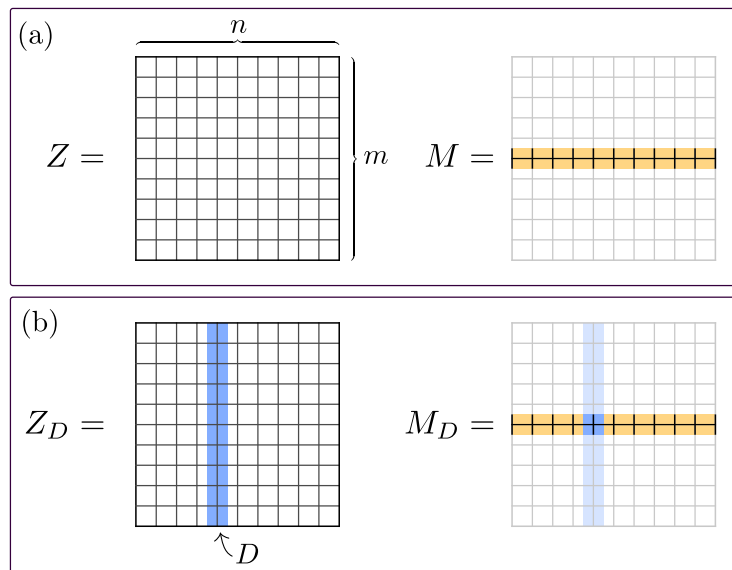


Figure 3.1: (a) Partition function Z on a square lattice made of $n \times m$ sites with periodic boundary conditions in both directions (a torus), and the corresponding transfer matrix M , such that $Z = \text{Tr}(M^m)$. (b) Partition function Z_D on the same torus, where the defect D implements some form of boundary conditions, and the generalized transfer matrix M_D , such that $Z_D = \text{Tr}((Z_D)^m)$.

By the operator-state correspondence of CFT [25], this partition function is expressed in terms of the scaling dimensions Δ_α and conformal spins s_α of some specific set of scaling operators ϕ_α , namely, those in the conformal towers of the three local primary fields of the Ising CFT: the identity, energy-density and spin primaries. We can extract $\{\Delta_\alpha, s_\alpha\}$ from the spectrum of eigenvalues of a transfer matrix M for the partition function Z , fulfilling $Z = \text{Tr}(M^m)$ [see Fig. 3.1(a)].

In the presence of a topological conformal defect D , the modified partition function Z_D is expressed now in terms of the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_D$ of other conformal towers, corresponding to a different set of primary fields. We can again extract $\{\Delta_\alpha, s_\alpha\}_D$ from the spectrum of eigenvalues of a modified transfer matrix M_D for the partition function Z_D , fulfilling $Z_D = \text{Tr}((M_D)^m)$ [see Fig. 3.1(b)].

In this chapter of the thesis, we build tensor network representations of the transfer matrices M_{D_ϵ} and M_{D_σ} for the two topological conformal defects D_ϵ and D_σ of the critical Ising model and explain how to extract accurate estimates of the corresponding scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_{D_\epsilon}$ and $\{\Delta_\alpha, s_\alpha\}_{D_\sigma}$, which we regard as a charac-

terization of these defects. These estimates are obtained by first coarse-graining and then diagonalizing the transfer matrices M_{D_ϵ} and M_{D_σ} .

We emphasize that on a sufficiently small $m \times n$ torus, say in the range $n \sim 10 - 20$, one can already diagonalize the transfer matrix M_D for defect D , and thus obtain numerical estimates of $\{\Delta_\alpha, s_\alpha\}_D$, by using exact diagonalization techniques. Why do we then need to use tensor networks? As we will review, these estimates for $\{\Delta_\alpha, s_\alpha\}_D$ are affected by non-universal, finite-size corrections, which diminish with growing n . The merit of tensor network techniques is then merely that, through proper coarse-graining, they allow us to consider a much larger n than exact diagonalization techniques, thus producing more accurate numerical estimates. However, the coarse-graining of the tensor network introduces truncation errors, which must be kept under check and effectively limit the size n that can be reliably considered.

In the absence of defects, the use of tensor networks to obtain more accurate estimates of the conformal data $\{\Delta_\alpha, s_\alpha\}$ by diagonalizing a transfer matrix M for the partition function Z was proposed and demonstrated in Ref. [47]. Here we generalize the proposal of Ref. [47] to the presence of defects, focusing on topological conformal defects for concreteness. This requires several new steps, which we list here. The first two steps, which apply to generic line defects on a 2D classical partition function (equivalent to a point defect in a 1D quantum model), are: (i) encoding of the defect D as a 1D tensor network, which upon insertion in the 2D tensor network for the clean partition function Z produces a tensor network for the defect partition function Z_D , as well as a tensor network for the corresponding defect transfer matrix M_D ; (ii) coarse-graining of the partition function Z_D / transfer matrix M_D . Diagonalization of the coarse-grained transfer matrix M_D will yield the scaling dimensions Δ_α associated to defect D . Steps (i) and (ii) apply to a generic type of conformal defect. Topological conformal defects are special in that we can extract additional conformal data, namely, the conformal spins s_α , by following two additional new steps: (iii) identification of a local unitary transformation that moves the location of the topological defect, so as to be able to define a generalized translation operator T_D that commutes with the transfer matrix M_D ; (iv) coarse-graining of the generalized translation operator T_D and diagonalization of the product $T_D M_D$ of coarse-grained versions of the translation operator T_D and the transfer matrix M_D .

The main result of this chapter of the thesis is the proposal of the steps (i) and (iv) described above to extract accurate estimates of the $\{\Delta_\alpha, s_\alpha\}_D$ associated to a topological conformal defect, together with a thorough demonstration of the approach for the symmetry defect D_ϵ and duality defect D_σ of the critical Ising model. We also propose steps (i)-(ii) to extract accurate estimates of the Δ_α associated to a generic (i.e. non-topological) conformal defect D , which we demonstrate with specific examples.

3.1.2 Structure of this chapter

Sections 3.2–3.4 are mostly devoted to discussing background material, whereas Secs. 3.5–3.8 contain our main results.

In Sec. 3.2, we review the Ising model in the absence of a defect. This includes the 2D classical Ising model on the square lattice, the 1D quantum Ising model, and the Ising CFT that effectively describes the previous two lattice models at criticality. We introduce the partition function Z and transfer matrix M , and relate the eigenvalue spectrum of M to the conformal data $\{\Delta_\alpha, s_\alpha\}$ of the three local primary fields of the Ising model. In Sec. 3.3, we review the use of tensor networks to represent the partition function Z and transfer matrix M , and of coarse-graining algorithms for tensor networks, which allow us to obtain estimates of $\{\Delta_\alpha, s_\alpha\}$ with smaller finite-size errors than is possible with exact diagonalization. In Sec. 3.4, we review the properties of the topological defects D_ϵ and D_σ of the critical Ising model, including their associated field content and their fusion rules.

Sections 3.5 and 3.6 analyze the symmetry defect D_ϵ and the duality defect D_σ , respectively. On the lattice, the topological character of a defect appears to be related to the existence of a local unitary transformation that changes the location of the defect. We identify such local unitary transformations for D_ϵ and D_σ , which also allow us to fuse these defects and confirm the expected fusion rules. We express the partition functions Z_{D_ϵ} and Z_{D_σ} of the critical Ising model in terms of transfer matrices M_{D_ϵ} and M_{D_σ} , and propose generalized translation operators T_{D_ϵ} and T_{D_σ} , which commute with M_{D_ϵ} and M_{D_σ} , respectively. Working with a tensor network representation of all the above objects, we then consider a coarse-graining transformation for the products $M_{D_\epsilon}T_{D_\epsilon}$ and $M_{D_\sigma}T_{D_\sigma}$, whose eigenvalue spectra yield the conformal data $\{\Delta_\alpha, s_\alpha\}_{D_\epsilon}$ and $\{\Delta_\alpha, s_\alpha\}_{D_\sigma}$.

It turns out that, from a tensor network perspective, the two topological conformal defects of the Ising model are very different. Indeed, as we will see in Sec. 3.5, the symmetry defect D_ϵ can be readily incorporated into a tensor network by employing \mathbb{Z}_2 -symmetric tensors [88], in which case D_ϵ is represented by a simple unitary matrix V acting on a bond index. This implies, in particular, that the defect transfer matrix M_{D_ϵ} and translation operator T_{D_ϵ} are obtained from the clean transfer matrix M and translation operator T by insertions of V . This also applies to the coarse-grained version of M_{D_ϵ} and T_{D_ϵ} , and we conclude that one can extract the new set of $\{\Delta_\alpha, s_\alpha\}_{D_\epsilon}$ with remarkably little effort by recycling the coarse-grained tensor networks used for the critical Ising model without a defect. On the other hand, we will see in Sec. 3.6 that representing the duality defect D_σ requires using different tensors altogether (as does a generic conformal defect), which one needs to explicitly coarse-grain.

In Sec. 3.7 we then briefly discuss the case of a generic (i.e. non-topological) conformal

defect D , for which we can also extract the scaling dimensions $\{\Delta_\alpha\}_D$ by diagonalizing a transfer matrix M_D for the defect partition function Z_D , and demonstrate the performance of the approach for a known continuous family of conformal defects of the critical Ising model.

Section 3.8 concludes this chapter with a discussion of the present tensor network approach, including its extension to defects in other critical lattice models, a discussion of different coarse-graining transformations one can use, and a comparison to other tensor network approaches (based on explicitly realizing scale invariance of the tensor network under coarse-graining) that can also be used to extract conformal data. Finally, several appendices contain technical discussions, as well as a study of the \mathbb{Z}_3 symmetry defects of the three-level Potts model.

3.1.3 Source code

The numerical results we present were obtained using a Python 3 implementation of the algorithms we describe. The source code is available at <https://arxiv.org/src/1512.03846/anc>, licensed under the MIT License, a permissive free software license. It can be used to reproduce our results and as the ultimate reference on details of the algorithms we describe here.

3.2 Critical Ising model

In this section, we review the critical Ising model on the lattice, both the 2D classical partition function and the 1D quantum spin chain. We also review their continuum limit, the Ising conformal field theory. Universal properties of the phase transition (conformal data) can be numerically estimated using exact diagonalization. The accuracy of these numerical estimates is limited by non-universal, finite-size corrections.

3.2.1 Classical partition function

The classical Ising model is defined by its Hamiltonian $K = -\sum_{\langle i,j \rangle} \sigma_i \sigma_j$, where i and j label sites on the lattice and σ_i is a classical spin variable on site i that can take the values ± 1 . $\sum_{\langle i,j \rangle}$ is a sum over nearest-neighbor pairs of sites. The partition function at inverse

temperature β is

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

where $\sum_{\{\sigma\}}$ is a sum over all the spin configurations. Here, we consider that the spins inhabit the sites of a square lattice with periodic boundaries in both directions (a torus). The Ising Hamiltonian K is invariant under a simultaneous flip of all the spins so that if we map $\sigma_i \mapsto -\sigma_i$ for all sites i the Hamiltonian remains unchanged: $K \mapsto K$. Flipping a spin twice recovers the original configuration, so the spin-flip symmetry is a global, internal \mathbb{Z}_2 symmetry. This symmetry is preserved at high temperatures (low β) and spontaneously broken at low temperatures (large β). At $\beta = \frac{1}{2} \log(1 + \sqrt{2})$ there is a second-order phase transition that separates the low temperature symmetry-breaking, ordered phase from the high temperature disordered phase.

The Ising model on the square lattice also has a order-disorder duality called the Kramers-Wannier duality which states that a low temperature Ising model is equivalent to a high temperature model on the dual lattice. At the critical point, the model is self-dual under this duality map. [5]

On a torus made of $m \times n$ spins we can write down a transfer matrix M such that $Z = \text{Tr}(M^m)$. The transfer matrix M (which is made of $2^n \times 2^n$ entries) is associated to a row of n spins. Many questions about the model can be answered in terms of the eigenvalue spectrum of M . We omit here the explicit form of M , which we will later build using tensor networks.

3.2.2 Quantum spin chain

The one-dimensional quantum Ising model is defined by the Hamiltonian

$$H(h) = - \left(\sum_{i=1}^n \sigma_i^z \sigma_{i+1}^z + h \sum_{i=1}^n \sigma_i^x \right). \quad (3.2)$$

On every site i there is a quantum spin and σ^z and σ^x are Pauli matrices. The parameter h is the strength of a transverse field.

The Hamiltonian is symmetric under a global spin-flip or, in other words,

$$\Sigma^x H (\Sigma^x)^\dagger = H, \quad (3.3)$$

where $\Sigma^x = \otimes_{i=1}^n \sigma_i^x$, and there are again two phases: a symmetry-breaking phase for small transverse magnetic field, and a disordered phase for large magnetic field, with a continuous quantum phase transition at $h = 1$.

The Kramers-Wannier duality for the quantum model equates it with another spin chain living on the dual lattice with a similar Hamiltonian but with external field [46] $\tilde{h} = \frac{1}{h}$. As in the classical model, the spin chain is Kramers-Wannier self-dual at the critical point.

The partition function of the quantum model for inverse temperature β_Q is $Z_Q = \text{Tr} \left(e^{-\beta_Q H} \right)$. A transfer matrix for this partition function can be written as $M_Q \equiv e^{-\frac{\beta_Q}{m} H}$, so that $Z_Q = \text{Tr} \left((M_Q)^m \right)$. Through the standard classical-quantum mapping, M_Q also corresponds to a transfer matrix for the partition function of a classical two-dimensional Ising model [84]. However, this classical dual is not the isotropic Ising model that we discussed above, but an extremely anisotropic one with very different couplings in different directions. The anisotropic and isotropic classical models are, nevertheless, in the same universality class, and hence the universal properties of the quantum and the classical Ising models are the same. To extract these universal properties we will mostly concentrate on studying the classical model.

3.2.3 Ising CFT

As we have mentioned above, the classical square lattice Ising model has a critical point at $\beta = \frac{1}{2} \log(1 + \sqrt{2})$ and the Ising spin chain has a quantum critical point at $h = 1$. The continuum limits of both of these critical points are described by the Ising CFT. The universal properties of the phase transition are captured by the conformal data of this CFT.

Consider the torus formed by parameterizing the two coordinates (x, y) of the plane by a complex variable $w \equiv x + iy$ and by identifying the points $w, w + 2\pi$ and $w + 2\pi\tau$, for $\tau = \tau_1 + i\tau_2$ a complex *modular parameter* that defines the shape of the torus. A purely imaginary modular parameter $\tau = i\tau_2$ produces a torus consisting of a rectangle with periodic boundaries.

On a torus defined by a complex modular parameter τ , the partition function of a CFT is

$$\begin{aligned} Z_{\text{CFT}} &= \text{Tr} \left(e^{-2\pi\tau_2(L_0 + \bar{L}_0 - \frac{c}{12})} e^{2\pi i\tau_1(L_0 - \bar{L}_0)} \right) \\ &= \text{Tr} \left(e^{-2\pi\tau_2 H_{\text{CFT}}} e^{2\pi i\tau_1 P} \right). \end{aligned} \quad (3.4)$$

Here L_0 and \bar{L}_0 are the Virasoro generators and $H_{\text{CFT}} = L_0 + \bar{L}_0 - \frac{c}{12}$ and $P = L_0 - \bar{L}_0$

are the Hamiltonian and momentum operators that generate translations in the directions $\text{Im}(w)$ and $\text{Re}(w)$, respectively, while c is the central charge. [45]

The scaling operators ϕ_α of the CFT are eigenoperators of dilations on an infinite plane. The operator-state correspondence identifies them with states $|\phi_\alpha\rangle$ that are the eigenstates of L_0 and \bar{L}_0 : $L_0|\phi_\alpha\rangle = h_\alpha|\phi_\alpha\rangle$ and $\bar{L}_0|\phi_\alpha\rangle = \bar{h}_\alpha|\phi_\alpha\rangle$. h_α and \bar{h}_α are known as the holomorphic and antiholomorphic conformal dimensions of ϕ_α . In terms of the eigenvalues of L_0 and \bar{L}_0 we can rewrite the partition function as

$$Z_{\text{CFT}} = \sum_{\alpha} e^{-2\pi\tau_2(h_\alpha + \bar{h}_\alpha - \frac{c}{12}) + 2\pi i\tau_1(h_\alpha - \bar{h}_\alpha)}, \quad (3.5)$$

$$= \sum_{\alpha} e^{-2\pi\tau_2(\Delta_\alpha - \frac{c}{12}) + 2\pi i\tau_1 s_\alpha}, \quad (3.6)$$

where $\Delta_\alpha = h_\alpha + \bar{h}_\alpha$ and $s_\alpha = h_\alpha - \bar{h}_\alpha$ are known as the scaling dimension and conformal spin of ϕ_α , respectively. [45] The scaling operators come in conformal towers built up from the primary operators. If h_p and \bar{h}_p are the conformal dimensions of a primary operator, then the scaling operators in its conformal tower have conformal dimensions of the form $h = h_p + k$ and $\bar{h} = \bar{h}_p + l$, where $k, l \in \mathbb{N}$. [45]

The Ising CFT is a conformal field theory of central charge $c = \frac{1}{2}$. For a unitary $c = \frac{1}{2}$ CFT the conformal dimensions h and \bar{h} of the primaries can take the values $0, \frac{1}{2}$ and $\frac{1}{16}$. [25] However, not all the possible combinations of these values of h and \bar{h} are realized as local primary operators in the CFT. The Ising CFT only includes the three ‘‘diagonal’’ primary operators that have $h = \bar{h}$. They are called the identity $\mathbb{1}$ for $(0, 0)$, the energy density ϵ for $(\frac{1}{2}, \frac{1}{2})$ and the spin σ for $(\frac{1}{16}, \frac{1}{16})$. Because of the \mathbb{Z}_2 symmetry of the model, the conformal towers come with a parity (a \mathbb{Z}_2 charge). This parity is $+1$ for $\mathbb{1}$ and ϵ , and -1 for σ .

We will see later in Sec. 3.4 that the non-diagonal combinations of h and \bar{h} are relevant to the discussion of topological conformal defects of the Ising CFT.

3.2.4 Extracting the universal data

For a critical, classical lattice model that has a CFT as its continuum limit, the partition function on an $m \times n$ torus (corresponding to $\tau = i\frac{m}{n}$) can be written as [10]

$$Z = \sum_{\alpha} e^{2\pi\frac{m}{n}(\frac{c}{12} - \Delta_\alpha) + mnf + \mathcal{O}(\frac{m}{n^\gamma})}, \quad \gamma > 1. \quad (3.7)$$

The sum is again over scaling operators. Equation (3.7) only differs from Eq. (3.5) in two non-universal terms. One non-universal term is the free energy term mnf , where f is the

free energy per site at the thermodynamic limit. The second is the subleading finite-size corrections $\mathcal{O}\left(\frac{m}{n^\gamma}\right)$, which become negligible for a large torus. The transfer matrix M corresponding to a row of n sites of the lattice so that $Z = \text{Tr}(M^m)$ then has eigenvalues [10]

$$\lambda_\alpha = e^{2\pi\frac{1}{n}\left(\frac{c}{12}-\Delta_\alpha\right)+nf+\mathcal{O}\left(\frac{1}{n^\gamma}\right)}. \quad (3.8)$$

More generally, if the transfer matrix M corresponds instead to l rows of n sites, so that now $Z = \text{Tr}\left(M^{\frac{m}{l}}\right)$, then its eigenvalues are [10]

$$\lambda_\alpha = e^{2\pi\frac{l}{n}\left(\frac{c}{12}-\Delta_\alpha\right)+lnf+\mathcal{O}\left(\frac{1}{n^\gamma}\right)}. \quad (3.9)$$

Thus, if we manage to diagonalize a transfer matrix for large n , the subleading, non-universal corrections will become negligible, whereas we can extract f by varying l and n while keeping $\frac{l}{n}$ fixed. We can then rescale M by e^{lnf} (or equivalently we can rescale the Boltzmann weights $e^{\beta\sigma_i\sigma_j}$ in the partition function Z) to get rid of the free energy term in the eigenvalue spectrum. From now on we will always assume the transfer matrix M has been rescaled in this way, so that its spectrum is

$$\lambda_\alpha \approx e^{2\pi\frac{l}{n}\left(\frac{c}{12}-\Delta_\alpha\right)}. \quad (3.10)$$

The equality is approximate because we have left out the subleading, non-universal, finite-size terms. Each of the eigenvalues λ_α then tells us the value of $\frac{c}{12} - \Delta_\alpha$ for one of the scaling operators ϕ_α . The scaling dimension $\Delta_{\mathbb{1}}$ of the identity operator is always 0 and for a unitary CFT this is the smallest scaling dimension possible. Thus we can obtain c from the largest eigenvalue λ_0 and the rest of the λ_α 's give us the rest of the scaling dimensions.¹

For a critical quantum spin chain of n spins the spectrum of the Hamiltonian is [10]

$$E_\alpha = a + b \left[2\pi\frac{1}{n} \left(\frac{c}{12} - \Delta_\alpha \right) + nf + \mathcal{O}\left(\frac{1}{n^\gamma}\right) \right] \quad (3.11)$$

where a and b are non-universal constants and $\gamma > 1$ as before. We could thus extract all the same critical data by diagonalizing the quantum Hamiltonian H instead of the transfer matrix M of the classical partition function Z . Here we choose, however, to work mostly with the transfer matrix.

¹In a finite system M only has a finite number of eigenvalues whereas there is an infinite number of scaling operators. However, we do observe that at least the largest λ_α 's correspond to the scaling operators with smallest Δ_α , see Fig. 3.2.

For a translationally invariant lattice model, such as the Ising model, the transfer matrix M commutes with the translation operator $T = e^{\frac{2\pi i}{n}P}$ that implements a discrete translation by one lattice site. Each eigenstate of T , with eigenvalue $e^{\frac{2\pi i}{n}p_\alpha}$, has well-defined momentum p_α . As discussed above, the momentum operator is $P = L_0 - \bar{L}_0$ and thus the momentum p_α corresponds to a conformal spin $s_\alpha = h - \bar{h}$. Hence we can diagonalize T and M simultaneously to obtain both the scaling dimensions Δ_α and the conformal spins s_α for the scaling operators ϕ_α . In fact, we can get away with even less work by diagonalizing only the product $T \cdot M$, which corresponds to the transfer matrix on a torus with a modular parameter τ that has real part $\tau_1 = \text{Re}(\tau) = 1/n$ and imaginary part $\tau_2 = \text{Im}(\tau) = l/n$. The eigenvalues of $T \cdot M$ are the products of the eigenvalues of T and M ,

$$\tilde{\lambda}_\alpha = \lambda_\alpha \cdot e^{\frac{2\pi i}{n}p_\alpha} \approx e^{2\pi \frac{l}{n}(\frac{c}{12} - \Delta_\alpha) + \frac{2\pi i}{n}s_\alpha}, \quad (3.12)$$

where we have again scaled away the free energy contribution and ignored the subleading finite-size corrections. The real part $\text{Re}(\log \tilde{\lambda}_\alpha) = 2\pi \frac{l}{n}(\frac{c}{12} - \Delta_\alpha)$ then yields the scaling dimension Δ_α and the imaginary part $\text{Im}(\log \tilde{\lambda}_\alpha) = \frac{2\pi i}{n}s_\alpha$ the conformal spin s_α . Note that because of the periodicity of $e^{\frac{2\pi i}{n}s_\alpha}$, the spin can only be determined modulo n , a point we will come back to later.

Thus to obtain scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}$ of the CFT numerically we can construct $T \cdot M$ for a finite but sufficiently large system and diagonalize it using an exact diagonalization algorithm. Results obtained in this manner are shown in Fig. 3.2. They clearly show the structure of the conformal towers coming out correctly and the accuracy of the estimates of Δ_α for the operators with lowest scaling dimensions is remarkably good. However, for operators with higher scaling dimensions the numerical estimates start to deteriorate significantly. This is due to the subleading finite-size corrections, which are still large at the system size $n = 18$ that we used here.

Unfortunately if we try to push for larger systems the computations quickly become prohibitively expensive because the dimension of the transfer matrix grows as 2^n , and the cost of exact diagonalization grows as the third power of this dimension. To diminish the effect of these non-universal corrections, we can describe the system using tensor networks and apply tensor network coarse-graining algorithms to reach large system sizes, as we review in the next section.

3.3 Tensor networks

In this section, we first review how tensor networks can be used to express the partition function Z and its transfer matrix M . We then describe how a coarse-graining algorithm

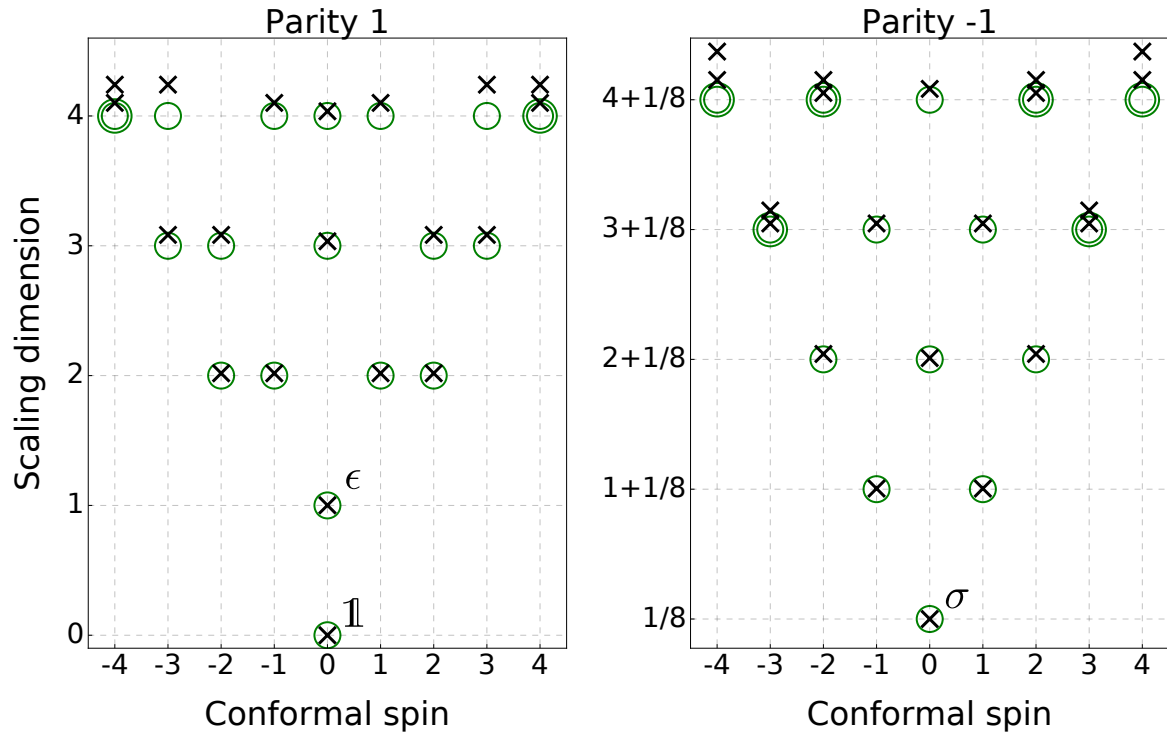


Figure 3.2: The scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first scaling operators of the Ising CFT obtained from exact diagonalization of a transfer matrix of $n = 18$ sites. The scaling operators are divided by their parity, i.e. their eigenvalue under the \mathbb{Z}_2 symmetry operator Σ^x that commutes with the transfer matrix. The crosses mark the numerical values that can be compared with the circles that are centered at the exact values. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair. The primary fields *identity* $\mathbb{1}$, *spin* σ and *energy density* ϵ appear at the basis of their three conformal towers.

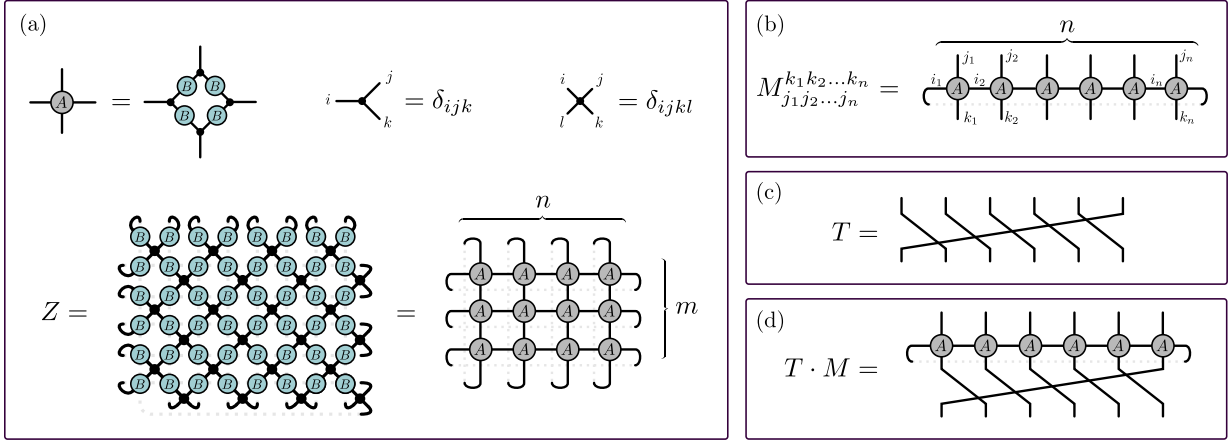


Figure 3.3: (a) The partition function Z of a classical two dimensional lattice model on a torus as a tensor network, first using the Boltzmann weights $B_{ij} = e^{-\beta E_{ij}}$ and then in terms of the tensor $A_{ijkl} = B_{ij}B_{jk}B_{kl}B_{li}$. We call the network on the right $Z_{n,m}(A)$. δ_{ijk} and δ_{ijkl} are three- and four-way Kronecker deltas that fix all their indices to have the same value. (b) The transfer matrix M as a tensor network. (c) The one-site translation operator T . (d) The translation operator composed with the transfer matrix.

for tensor networks can be used to analyze larger systems than with exact diagonalization, as first proposed and demonstrated in Ref. [47]. By diagonalizing a transfer matrix M corresponding to a large number n of sites we can reduce very significantly the errors, due to finite-size corrections, in the estimates of scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}$. This comes at the price of introducing truncation errors.

3.3.1 Tensor network representation

In terms of the Boltzmann weights $B_{ij} = e^{\beta \sigma_i \sigma_j}$ the partition function is

$$Z = \sum_{\{\sigma\}} \prod_{\langle i,j \rangle} B_{ij}, \quad (3.13)$$

with the sum and the product being over all spin configurations and all nearest-neighbor pairs, respectively. Periodic boundary conditions in both directions are again assumed. As explained in Chapter 2, such a partition function can be written as a tensor network in a natural way. In this chapter we will use a slightly different, but equivalent of expressing Z as a network, shown in Fig. 3.3(a) at the bottom. The first network in Fig. 3.3(a) is a

straight-forward translation of Eq. (3.13) into the graphical tensor network notation. In it for every spin there is a four-index Kronecker delta δ_{ijkl} . Each of the four indices connects to one of the neighboring spins through the matrix B . On the right this is then rewritten in terms of tensor $A_{ijkl} = B_{ij}B_{jk}B_{kl}B_{li}$ that encodes the interactions around a plaquette of spins. Every index of A corresponds to one spin. We will be working with this latter network, which we denote $Z_{n,m}(A)$. Notice that here m and n label the number of rows and columns of tensors A , not of spins, with each tensor accounting for two spins. However, all the expressions in the preceding section can be seen to remain valid due to the isotropy of the original spin model. Note that the construction from Section 2.2 could equally well be used, and the reasons for different choices in the two chapters are circumstantial.

From Fig. 3.3(a) it is clear that we can write $Z = \text{Tr}(M^m)$ where the transfer matrix is as in Fig. 3.3(b), or in other words

$$M_{j_1 j_2 \dots j_n}^{k_1 k_2 \dots k_n} = \sum_{i_1, i_2, \dots, i_n} \prod_{\alpha=1}^n A_{i_\alpha j_\alpha i_{\alpha+1} k_\alpha}. \quad (3.14)$$

Here all the i_α indices are summed over and i_{n+1} is identified with i_1 . When writing $\text{Tr}(M^m)$ we have interpreted M as a linear map from \mathbb{V}_j to \mathbb{V}_k , where \mathbb{V}_j (respectively \mathbb{V}_k) is the tensor product of the vector spaces of the indices j_α (respectively k_α).

Implementing a lattice translation in the network is straight-forward, and shown in Fig. 3.3(c). In Fig. 3.3(d) is the operator $T \cdot M$, which we want to diagonalize in order to extract universal data of a phase transition (see previous section).

It should be noted here that when we translate between network diagrams and equations, our convention is that reading an equation from left to right corresponds to reading a diagram from either left to right or bottom to top, but never right to left or top to bottom.

The \mathbb{Z}_2 symmetry of the Ising model plays an important role in the tensor network representation, as we will see later when we consider a system with a topological defect. For a model with a global internal symmetry, the symmetry can be made manifest in the tensors themselves. This is covered in length in Refs. [87, 88, 89]. For the present discussion it suffices to know that for the Ising model, we use tensors that fulfill the identity in Fig. 3.4, namely, tensors that are left unchanged if we apply a spin-flip matrix V on each of the indices. The spin-flip matrix V here is nothing but σ^x , but we call it V for consistency with the case where it acts on a coarse-grained index. As explained in Appendix B.1, in general V is some unitary representation of the non-trivial element of the symmetry group \mathbb{Z}_2 . In other words it is a unitary matrix such that $V^2 = \mathbb{1}$. We call a tensor that obeys the invariance property of Fig. 3.4 a \mathbb{Z}_2 invariant tensor. The vector space attached to each index of a \mathbb{Z}_2 invariant tensor can be understood as the direct sum of two subspaces,

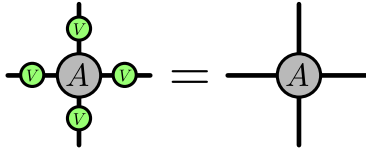


Figure 3.4: The invariance of tensor A under the symmetry transformation V .

one for each parity \pm (i.e. each \mathbb{Z}_2 charge). In this way, we can attach a parity to each eigenvalue/vector of the transfer matrix and, by extension, to the corresponding scaling operators ϕ_α (discussed in the previous section).

3.3.2 Coarse-graining

In Chapter 2 we discussed in length coarse-graining methods for tensor networks, and here, too, we shall use one. For the purposes of this chapter, the exact details of the method are not of qualitative importance. Rather we can consider in general a transformation that maps a network like $Z_{n,m}(A)$ [see Fig. 3.3(a)] to a smaller network $Z_{n',m'}(A')$ that is of the same form but consists of tensors A' . Each A' corresponds to a small region of the original network $Z_{n,m}(A)$ and describes longer length scale features of the system. For concreteness we consider a coarse-graining where each A' corresponds to four of the original tensors A and $n' = \frac{n}{2}$ and $m' = \frac{m}{2}$, i.e. the coarse-graining has scaled the linear size of the system by a factor of $\frac{1}{2}$. Ideally the coarse-grained and the original network would contract to exactly the same value, the partition function, but in practice this is true only up to truncation errors incurred in the coarse-graining.

Any such map produces a sequence of coarse-grained tensors that we call $A^{(s)}$, with $A^{(0)} \equiv A$ and s labeling the number of coarse-grainings. Each tensor network $Z_{\frac{n}{2^s}, \frac{m}{2^s}}(A^{(s)})$ is an approximate representation of the original network $Z_{n,m}(A^{(0)})$. This is illustrated in Fig. 3.5. As first proposed and demonstrated in Ref. [47], we can then use the coarse-grained tensors $A^{(s)}$ to produce a transfer matrix M representing many spins and extract $\{\Delta_\alpha, s_\alpha\}$ with smaller finite-size corrections. We emphasize that although Ref. [47] described this approach in the context of a particular coarse-graining scheme, namely the Tensor Entanglement-Filtering Renormalization (TEFR) method, it can be used with any coarse-graining scheme that accurately preserves the partition function Z or transfer matrix M .

As always with tensor networks, a key role is played by the bond dimension χ of the indices of $A^{(s)}$, which controls both the computational cost and the truncation errors. For

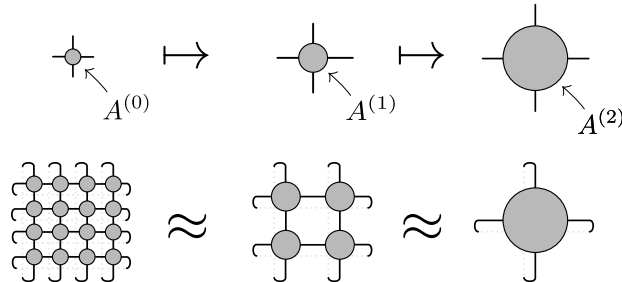


Figure 3.5: Repeating a coarse-graining produces a series of tensors $A^{(s)}$ and corresponding networks that all contract to approximately the same value. We think of each $A^{(s)}$ as representing a local patch of the system at a different length scale. With a $2 \times 2 \mapsto 1$ coarse-graining like the one we consider, a network $Z_{2^k, 2^k}(A)$ can be coarse-grained to a single tensor in k steps.

a fixed χ , i.e. for fixed computational cost per coarse-graining step, the best numerical estimates are obtained by applying a number of iterations s such that the finite-size corrections and truncation errors are of the same magnitude, and their cumulative effect on the results is at a minimum.

The specific coarse-graining scheme that we use in this chapter of the thesis is Vidal & Evenbly’s Tensor Network Renormalization (TNR) [37]. It is based on inserting approximate partitions of unity into the network, consisting of isometric and unitary tensors, that can be optimized to minimize the truncation error. We will not explain the details of the algorithm, but refer the reader to Refs. [31] and [37]. However, an outline of the algorithm can be found in Appendix B.1. Similar results to the ones obtained in this chapter with TNR could equally well be obtained (with varying computational cost and accuracy) using any coarse-graining that acts sufficiently locally, such as the simpler Tensor Renormalization Group algorithm [68] or the Gilt-TNR algorithm of Chapter 2. This point is elaborated further in the discussion section.

Applying a $2 \times 2 \mapsto 1$ coarse-graining transformation s times, a transfer matrix of $2^s \times (2^s \cdot n_s)$ tensors A can be coarse-grained down to the transfer matrix $M^{(s)}$ in Fig. 3.6, consisting of a row of n_s tensors $A^{(s)}$. The computational cost thus scales logarithmically in system size. Interpreted as a matrix, $M^{(s)}$ has dimensions $\chi^{n_s} \times \chi^{n_s}$, and can be diagonalized for sufficiently small values of n_s and χ . We diagonalize $M^{(s)}$ simultaneously with a translation operator $T^{(s)}$, also shown in Fig. 3.6. $T^{(s)}$ is a translation by 2^s sites in the original system and its eigenvalues yield the conformal spins modulo n_s , as explained in Appendix B.1. In Appendix B.2 we show how to perform a final coarse-graining step on

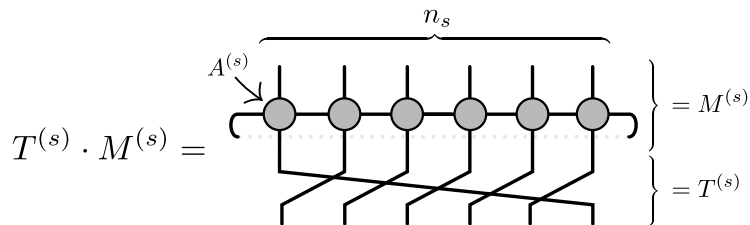


Figure 3.6: The coarse-grained transfer matrix and translation operator.

Primary	(h, \bar{h})	Δ_{TNR}	Δ_{exact}	s_{TNR}	s_{exact}
$\mathbb{1}$	$(0, 0)$	–	0	0	0
ϵ	$(1/2, 1/2)$	1.000256	1	0	0
σ	$(1/16, 1/16)$	0.125109	0.125	0	0

Table 3.1: The scaling dimensions and conformal spins of the primary fields of the Ising CFT obtained using TNR, contrasted with the exact values. No numerical value for the scaling dimension of the identity operator is provided because we extract the central charge c by assuming that $\Delta_{\mathbb{1}} = 0$ exactly. The central charge we get is $c = 0.500091$ whereas the exact one is $c = \frac{1}{2}$. The conformal spins we obtain are exactly zero without any numerical errors, because we know that the possible eigenvalues of the translation operator for a four site system are ± 1 and $\pm i$, which yields the possible conformal spins $-1, 0, 1$ and 2 .

the composite operator $T^{(s)} \cdot M^{(s)}$ to raise the periodicity of the conformal spins to $2n_s$.

3.3.3 Numerical results

In Fig. 3.7, we show scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}$ obtained by diagonalizing a transfer matrix that has been coarse-grained using TNR. To obtain the scaling dimensions we have diagonalized a transfer matrix of 4 tensors $A^{(s)}$ for $s = 7$ (corresponding to $2 \times 4 \times 2^7 \times 2^7 = 2^{17} \approx 130\,000$ spins) using bond dimensions $\chi' = 14$ and $\chi = 28$ (the TNR scheme we use has two relevant bond dimensions, see Appendix B.1). For the conformal spins a slightly larger system was used, as explained in Appendix B.2. The numerical results are in excellent agreement with the exact values even higher up in the conformal towers, in contrast with the exact diagonalization results in Fig. 3.2. Table 3.1 shows a comparison of the numerical values to the exact ones for the primary fields. For the central charge we obtain $c = 0.500091$ where the exact value would be $\frac{1}{2}$.

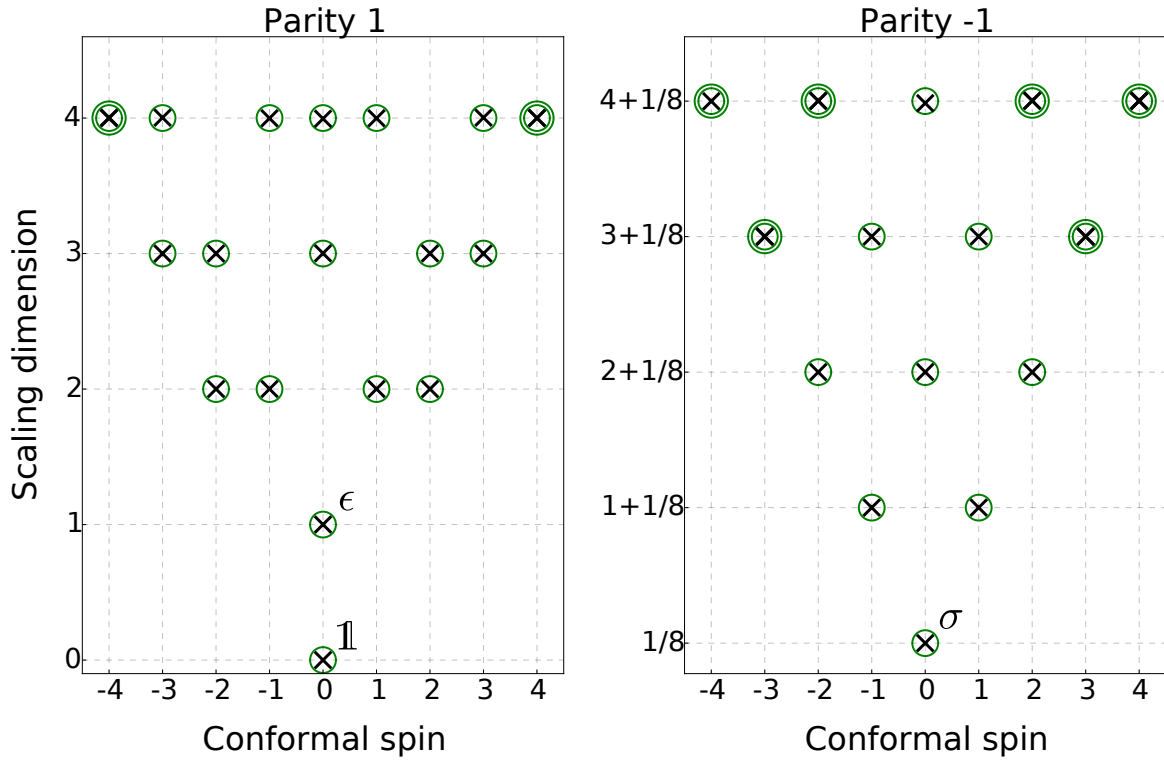


Figure 3.7: By coarse-graining and diagonalizing a transfer matrix for the classical square lattice Ising model we have obtained scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first few scaling operators with lowest dimensions in the Ising CFT. The crosses mark the numerical values that should be compared with the circles that are centered at the exact values. The scaling operators are divided according to their \mathbb{Z}_2 charge, that is their parity under a global spin-flip. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair. Although it is not clear from the figure, these degeneracies also come out correctly.

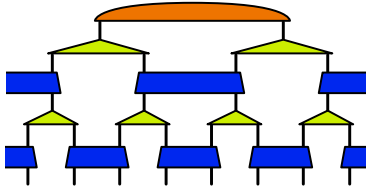


Figure 3.8: A MERA for a state of eight spins in a system with periodic boundaries. Such a network for the ground state of a spin chain can be obtained by applying TNR to a tensor network describing the Euclidean path integral of the quantum Hamiltonian [38]. The tensors in the network are the same unitaries and isometries as used in Appendix B.1. Even though our notation does not reflect it, the unitaries and isometries on different layers generally differ from each other.

3.3.4 Multiscale Entanglement Renormalization Ansatz

We conclude this section by recalling that, as explained in Ref. [38], if we apply TNR to a tensor network representing the Euclidean path integral $e^{-\beta Q^H}$ of a quantum Hamiltonian H , we obtain a Multiscale Entanglement Renormalization Ansatz (MERA) for the ground state of H . Such a MERA, shown in Fig. 3.8, is built of disentanglers and isometries that are produced during the coarse-graining of the Euclidean path integral. We have observed empirically that the partition function of the 2D classical Ising model that we study is also the Euclidean path integral of the 1D quantum model. It then follows that the disentanglers and isometries produced during the coarse-graining of the classical partition function Z can be put together into a MERA that represents the ground state of the Hamiltonian H for the 1D quantum Ising model. This observation will extend to the case of topological defects, discussed in Secs. 3.5 and 3.6.

3.4 Topological defects

In this section, we review the topological conformal defects (often referred to simply as topological defects) of the Ising CFT. On a torus these defects can be thought of as different boundary conditions, and their presence modifies the operator content of the partition function. The Ising CFT has two different non-trivial topological defects, which we introduce in this section and whose lattice realization will be analyzed in the next two sections.

As stated earlier in Eq. (3.4), the partition function of a CFT on a torus can be written

as

$$Z_{\text{CFT}} = \text{Tr} \left(e^{-2\pi\tau_2 H_{\text{CFT}}} e^{2\pi i\tau_1 P} \right). \quad (3.15)$$

We consider now a twisted partition function Z_D of the form²

$$Z_D = \text{Tr} \left(D e^{-2\pi\tau_2 H_{\text{CFT}}} e^{2\pi i\tau_1 P} \right). \quad (3.16)$$

D is the twist operator, which can be thought of as implementing a special type of boundary condition on the torus. If D commutes with all the generators of the Virasoro algebra it is called a topological conformal defect (topological defect for simplicity). A twist operator can be seen as a line defect that loops around the torus. If the defect is topological the loop can be freely deformed without affecting correlation functions in the system as long as the defect is not moved across a field insertion. The conformality of the defect also means that it is invariant under scale transformations. [43, 80]

The twisted partition function Z_D for a topological defect can be written as a sum of terms corresponding to scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_D$, similarly as for the non-twisted Z in Eq. (3.5). However, the $\{\Delta_\alpha, s_\alpha\}_D$ present in the sum are in general different from those of the non-twisted Z . [43, 80]

For the Ising CFT, all possible topological defects can be built as linear combinations of three defects which we denote $D_{\mathbb{1}}$, D_ϵ and D_σ . They are known as the *simple* defects of the Ising CFT and are related to the same irreducible representations of the Virasoro algebra as the primary fields $\mathbb{1}$, ϵ and σ , hence the names. [42, 80] The $D_{\mathbb{1}}$ defect is the trivial defect or rather the lack of any defect where the twist operator is just the identity. The partition function for it, $Z_{D_{\mathbb{1}}} = Z$, has been the topic of the last two sections. D_ϵ is also known as the (\mathbb{Z}_2) symmetry defect and D_σ is called the (Kramers-Wannier) duality defect.

The operators present in the twisted partition functions Z_{D_ϵ} and Z_{D_σ} come organized in conformal towers built on top of primary operators, each of which is identified with conformal dimensions (h, \bar{h}) that may take values $0, \frac{1}{2}$ and $\frac{1}{16}$, just like for the operators of the non-twisted $Z_{D_{\mathbb{1}}}$. The combinations of h and \bar{h} that are present in each partition function are shown in Table 3.2. Note that together the three partition functions include all the possible pairs (h, \bar{h}) . [80]

Consider bringing two defects next to each other so that they effectively behave as one defect. This gives rise to fusion rules for topological defects. The fusion rules of the

²We have dropped here the previous distinction between the partition function Z of a lattice model and Z_{CFT} of the CFT: Here by Z_D we mean a field theory partition function, whereas later we will use the same symbol to refer to its lattice realization. Context should make clear which one we are referring to.

Z_{D_1}	0	$\frac{1}{2}$	$\frac{1}{16}$	Z_{D_ϵ}	0	$\frac{1}{2}$	$\frac{1}{16}$	Z_{D_σ}	0	$\frac{1}{2}$	$\frac{1}{16}$
0	$\mathbb{1}$			0	ψ			0	X		
$\frac{1}{2}$	ϵ			$\frac{1}{2}$	$\bar{\psi}$			$\frac{1}{2}$	Y		
$\frac{1}{16}$	σ			$\frac{1}{16}$	μ			$\frac{1}{16}$	\bar{X}	\bar{Y}	

Table 3.2: The primary operators with conformal dimensions (h, \bar{h}) included in the Ising partition functions with different defects in them. The horizontal axis is h , the vertical one is \bar{h} . The operators in Z_{D_1} and Z_{D_ϵ} have established names shown in the table, with the Z_{D_1} ones already familiar to us from earlier in the chapter. The ones included in Z_{D_σ} we denote X, \bar{X}, Y and \bar{Y} in the absence of more a established convention.

topological defects of the Ising model are the same as the fusion rules of the primary operators, namely

$$D_\epsilon \times D_\epsilon = D_1 \tag{3.17}$$

$$D_\sigma \times D_\epsilon = D_\sigma \tag{3.18}$$

$$D_\sigma \times D_\sigma = D_1 + D_\epsilon. \tag{3.19}$$

The next two sections are devoted to analyzing the two non-trivial defects D_ϵ and D_σ on the lattice. Similarly as we have done in the two preceding sections for the trivial defect D_1 , we will first study the realization of each defect in the quantum and classical lattice models, then discuss how to represent them using tensor networks, and finally study how to coarse-grain these networks. We will then diagonalize transfer matrices for the twisted partition functions and extract the scaling dimensions and conformal spins of the scaling operators. In the process we will also end up discussing how these defects can be moved around and how their fusion rules manifest in the lattice models.

3.5 Symmetry defect D_ϵ

In this section, we review how to realize, on the lattice, the symmetry defect D_ϵ of the Ising CFT in the classical and quantum Ising models and present a way of implementing D_ϵ in a tensor network. We discuss how to coarse-grain this tensor network representation and use it to numerically evaluate the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_{D_\epsilon}$ of the operators in the twisted partition function Z_{D_ϵ} .

3.5.1 Lattice representation

The symmetry defect D_ϵ is directly related to the \mathbb{Z}_2 spin-flip symmetry of the Ising model. Consider the classical Ising model on a square lattice and draw a closed loop around a connected region of spins. If one flips the spins encircled by the loop, then most of the terms $-\sigma_i\sigma_j$ in the Hamiltonian are unaffected, but along the loop there is a string of nearest-neighbor pairs where one of the spins is flipped and the other one is not and their couplings become $\sigma_i\sigma_j$. This string is the classical lattice realization of the D_ϵ defect.

We are interested in a partition function Z_{D_ϵ} where such a defect forms a non-contractible loop around a torus. Z_{D_ϵ} is said to have antiperiodic boundary conditions due to how most of the spins are coupled ferromagnetically but along the boundary the coupling is antiferromagnetic.

Analogously, in the quantum spin chain the D_ϵ defect is realized by changing the sign of one of the nearest-neighbor terms to obtain

$$H_{D_\epsilon} = - \left(\sum_{i=1}^{n-1} \sigma_i^z \sigma_{i+1}^z - \sigma_n^z \sigma_1^z + \sum_{i=1}^n \sigma_i^x \right). \quad (3.20)$$

Let us concentrate on the quantum case for a moment and consider moving and fusing D_ϵ defects. In the Hamiltonian 3.20 the defect is located on the coupling between spins n and 1. If we conjugate the Hamiltonian with σ_n^x , we effectively move the defect by one site:

$$\sigma_n^x H_{D_\epsilon} \sigma_n^x = - \left(\sum_{i=1}^{n-2} \sigma_i^z \sigma_{i+1}^z + \sum_{i=1}^n \sigma_i^x \right) \quad (3.21)$$

$$- \sigma_{n-1}^z \sigma_n^z + \sigma_n^z \sigma_1^z \Big). \quad (3.22)$$

σ_n^x is unitary and Hermitian and conjugating by it does not affect the spectrum of the Hamiltonian. Using σ^x to move defects one can easily check that taking a Hamiltonian with two D_ϵ defects and moving them to the same site gives the usual Ising Hamiltonian H . This is the fusion rule $D_\epsilon \times D_\epsilon = D_{\mathbb{1}}$.

Similarly in the classical model a D_ϵ defect can be moved by flipping a spin that is next to it. This is because flipping a spin changes all its couplings from ferromagnetic to antiferromagnetic or vice versa. Such a spin-flip is only a matter of relabeling a degree of freedom and does not affect the partition function. Now consider a system with two D_ϵ defects parallel to each other and move one of them until there is only a string of

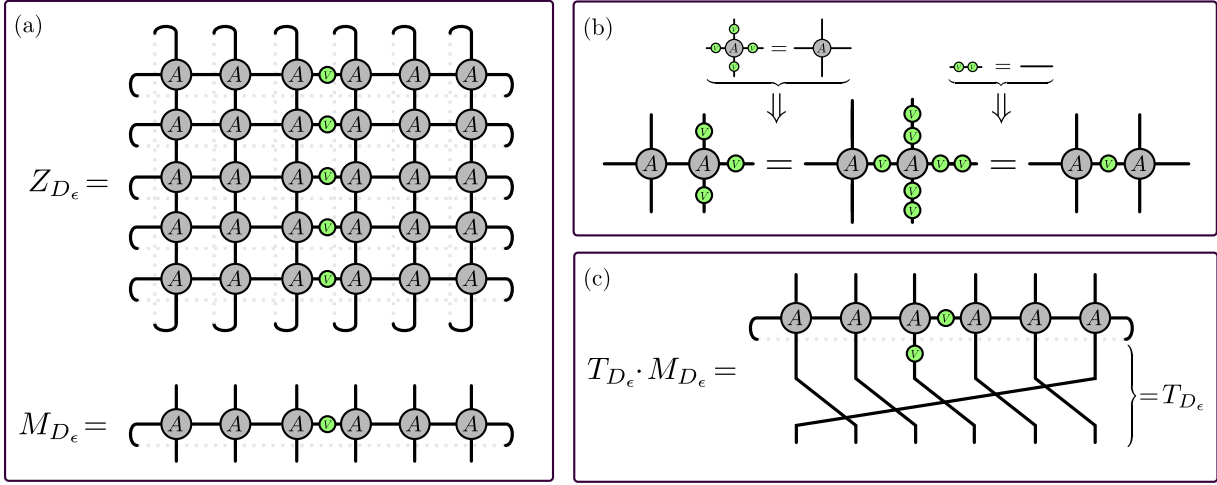


Figure 3.9: (a) The tensor networks for the partition function Z_{D_ϵ} and its transfer matrix M_{D_ϵ} , that fulfills $Z_{D_\epsilon} = \text{Tr}(M_{D_\epsilon}^m)$. V is the Pauli matrix σ^x that flips a spin. (b) The invariance property of A and the fact that $V^2 = \mathbb{1}$ imply that conjugating the tensor next to the defect from above and below with V moves the defect by one site. (c) The operator $T_{D_\epsilon} \cdot M_{D_\epsilon}$ whose diagonalization produces the conformal data $\{\Delta_\alpha, s_\alpha\}_{D_\epsilon}$ for Z_{D_ϵ} . Here, T_{D_ϵ} is the translation operator that commutes with the transfer matrix M_{D_ϵ} .

spins between the two defects. Flipping the spins between the defects changes all the antiferromagnetic couplings to ferromagnetic and we are left with the usual Ising model — a D_1 defect.

3.5.2 Tensor network representation

In a tensor network $Z_{n,m}(A)$ that represents a partition function Z_{D_1} [see Fig. 3.3(a)], every bond corresponds to one classical spin. If we multiply one of the legs i of a tensor A_{ijkl} with a spin-flip matrix V this makes the corresponding spin i couple antiferromagnetically to its neighbors on one side. Thus, the network for Z_{D_ϵ} is as depicted in Fig. 3.9(a), where the defect lives on a string of bonds. The same figure shows the transfer matrix M_{D_ϵ} for this twisted partition function.

Just as in the system without a defect, by diagonalizing M_{D_ϵ} we can extract the scaling dimensions of the operators in Z_{D_ϵ} . For the conformal spins we would need to diagonalize M_{D_ϵ} simultaneously with the translation operator. However, the usual lattice translation T

does not commute with M_{D_ϵ} because the translation moves the defect by one lattice site. We can move the defect back to where it was by conjugating the tensor A that is next to the defect with V from above and below, as shown in Fig. 3.9(b).

Thus the operator $T_{D_\epsilon} = VT$ commutes with M_{D_ϵ} . We call T_{D_ϵ} the generalized translation operator for the D_ϵ defect. It is the notion of translation under which the partition function with a D_ϵ defect is translation invariant. [12, 105] $T_{D_\epsilon} \cdot M_{D_\epsilon}$, shown in Fig. 3.9(c), has a spectrum of the form in Eq. (3.12) from which the conformal spins and scaling dimensions of the operators in Z_{D_ϵ} can be obtained. Figure 3.10 shows estimates for $\{\Delta_\alpha, s_\alpha\}$ obtained numerically by diagonalizing a transfer matrix consisting of $n = 18$ tensors. The structure of the bottom of the conformal towers can be clearly recognized, but the accuracy of the estimates deteriorates quickly as we look at larger scaling dimensions. We shall see below that again increasing the system size by using a coarse-graining algorithm will greatly improve these estimates.

3.5.3 Coarse-graining

Coarse-graining the symmetry defect is trivial when we use a coarse-graining scheme that is based on using \mathbb{Z}_2 invariant tensors: By utilizing the invariance of all the tensors involved we can move the string of spin-flip matrices V to the next scale without any additional numerical effort. In other words, as is shown in Fig. 3.11, a network $Z_{n,m}(A)$ that has V matrices on a string of bonds coarse-grains into a network $Z_{\frac{n}{2}, \frac{m}{2}}(A')$ with matrices V' on a string of bonds. A' is the same tensor that we obtain when coarse-graining a network without a defect. V' 's are the spin-flip matrices of the bonds of the coarse-grained network, i.e. the unitary representations of the non-trivial element of \mathbb{Z}_2 under which A' is \mathbb{Z}_2 invariant. Similarly, the generalized translation operator T_{D_ϵ} coarse-grains into a translation operator T'_{D_ϵ} at next scale with a V' on one of the legs. How, exactly, this happens when using TNR is explained in Appendix B.1. However, this also holds for other coarse-graining schemes, provided that \mathbb{Z}_2 symmetric tensors are used.

The coarse-graining of Z_{D_ϵ} or M_{D_ϵ} can be iterated. We then diagonalize an operator $T_{D_\epsilon} \cdot M_{D_\epsilon}$ similar to that in Fig. 3.9(c), but with $A^{(s)}$'s and $V^{(s)}$'s instead of A 's and V 's. The form of $V^{(s)}$ is determined by the \mathbb{Z}_2 invariance property of $A^{(s)}$. Notice that if we had already coarse-grained an equivalent network without defects, then we already have the tensors $A^{(s)}$ and $V^{(s)}$ and the only additional computational work required is the diagonalization of $T_{D_\epsilon} \cdot M_{D_\epsilon}$.

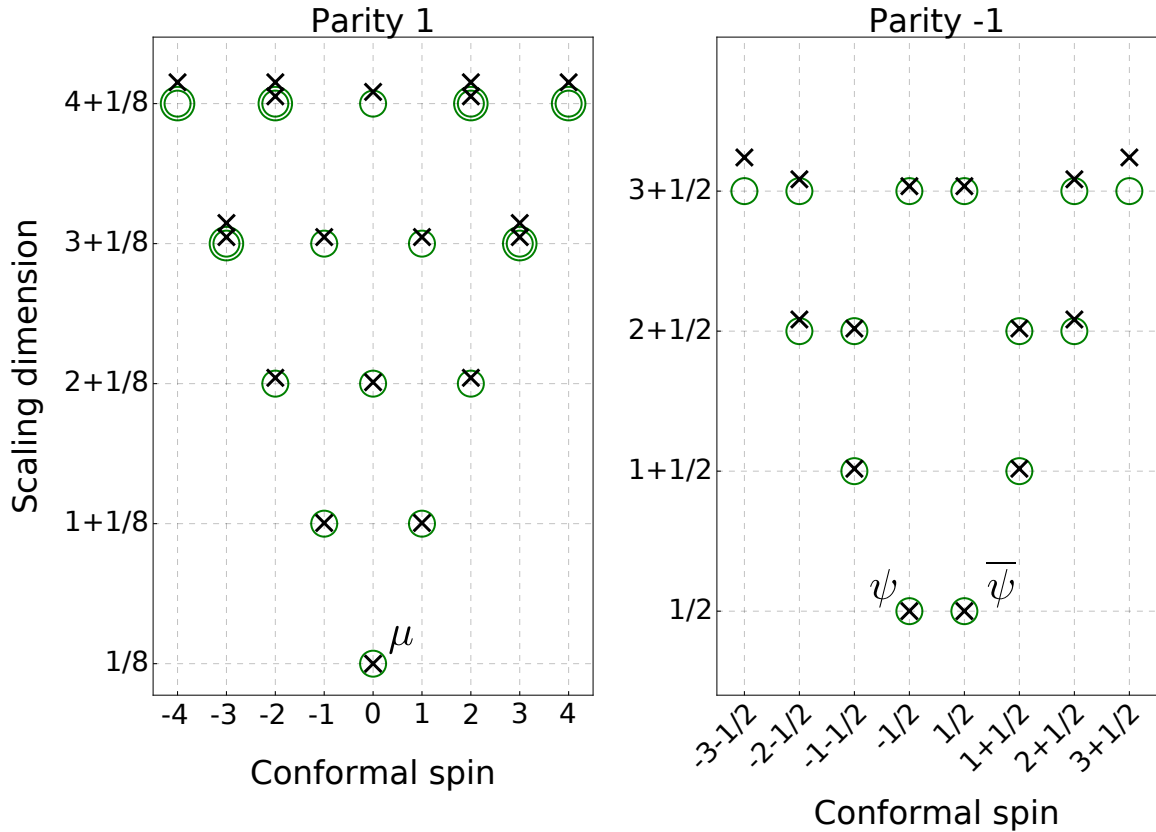


Figure 3.10: The scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first scaling operators in Z_{D_ϵ} obtained from exact diagonalization of a transfer matrix of $n = 18$ sites. The scaling operators are again divided by their \mathbb{Z}_2 charge. The crosses mark the numerical values that can be compared with the circles that are centered at the exact values. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair.

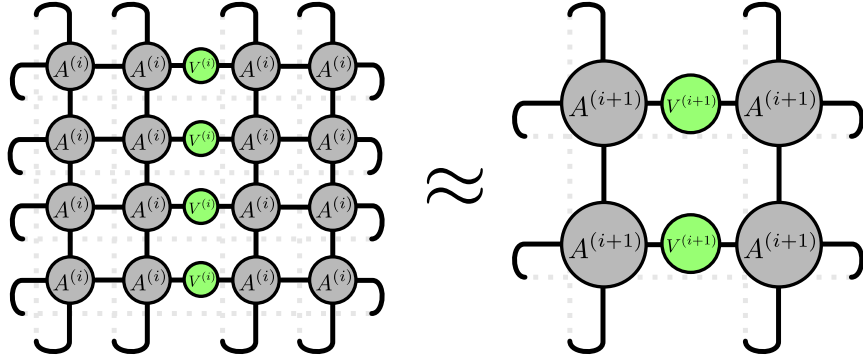


Figure 3.11: Coarse-graining a D_ϵ defect produces a similar defect at the next scale. $A^{(i)}$ and $A^{(i+1)}$ are the same tensors that we obtain when coarse-graining a system without a defect and $V^{(i)}$ and $V^{(i+1)}$ are the spin-flip matrices of the bonds they are on.

3.5.4 Numerical results

As mentioned earlier in Sec. 3.4, the primary fields present in Z_{D_ϵ} are μ , ψ and $\bar{\psi}$ with the conformal dimensions $(\frac{1}{16}, \frac{1}{16})$, $(\frac{1}{2}, 0)$ and $(0, \frac{1}{2})$ [80]. μ has parity $+1$, ψ and $\bar{\psi}$ have parity -1 . Numerical results for the scaling dimensions and conformal spins of these primaries obtained with the TNR method are shown in Table 3.3. Similar values for some of the first descendants are shown in Fig. 3.12. The results are in excellent agreement with the exact results even higher up in the conformal towers. The parameters used to obtain these results are the same as for the D_1 case: A transfer matrix of $2^7 \times (4 \times 2^7)$ tensors $A^{(0)}$ (corresponding to $2 \times 4 \times 2^7 \times 2^7 = 2^{17}$ spins) coarse-grained with bond dimensions $\chi' = 14$ and $\chi = 28$ for the scaling dimensions and a transfer matrix twice as wide and with one additional coarse-graining step for the conformal spins, as explained in Appendix B.2.

3.5.5 MERA

By using TNR to coarse-grain the tensor network for Z_{D_ϵ} we get a MERA for the ground state of the quantum model with a D_ϵ defect. As shown in Fig. 3.13 it is like the defectless MERA, but with a spin-flip matrix $V^{(s)}$ at every layer and a different top tensor. Note that this is a special case of an impurity MERA, discussed in Ref. [36].

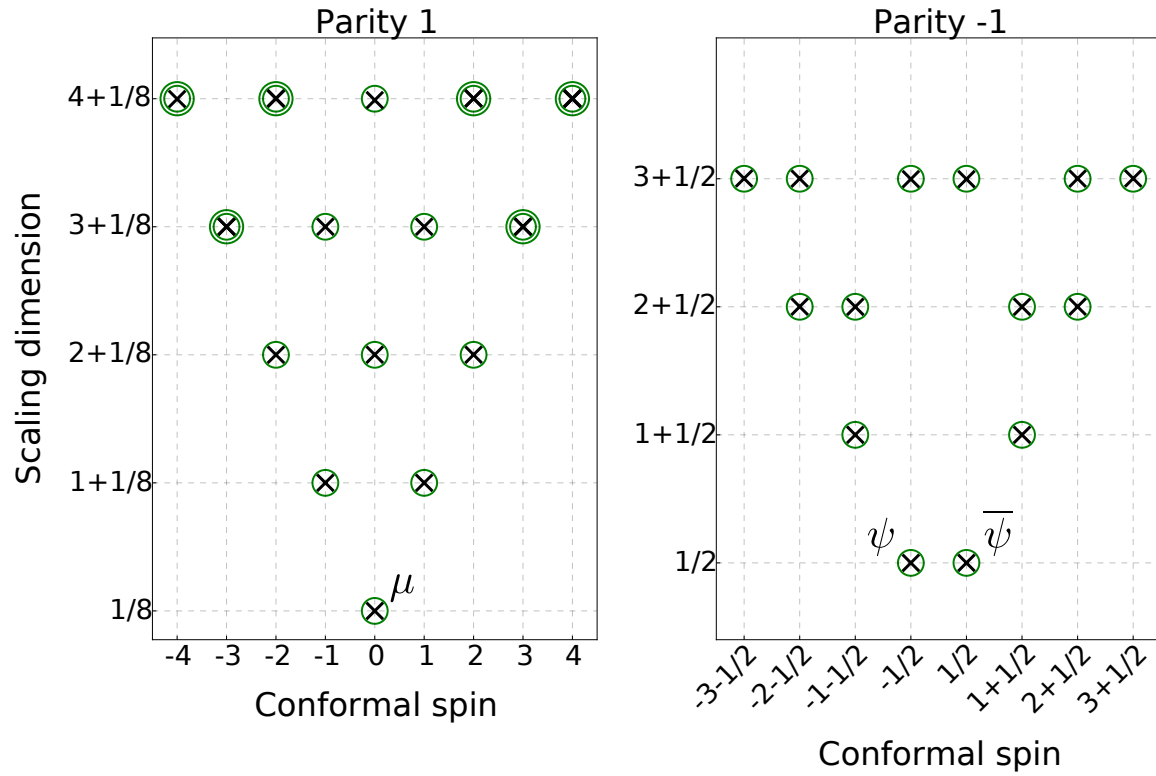


Figure 3.12: The scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first scaling operators of the two-dimensional classical Ising model with a D_ϵ defect as obtained with TNR. The crosses mark the numerical values and the circles mark the exact values. The scaling operators are divided according to their \mathbb{Z}_2 charge, that is their parity under a global spin-flip. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair.

Primary	(h, \bar{h})	Δ_{TNR}	Δ_{exact}	s_{TNR}	s_{exact}
μ	$(1/16, 1/16)$	0.1249287	0.125	10^{-16}	0
ψ	$(1/2, 0)$	0.5000704	0.5	0.4999847	0.5
$\bar{\psi}$	$(0, 1/2)$	0.5000704	0.5	-0.4999847	-0.5

Table 3.3: The scaling dimensions Δ and conformal spins s for the primaries of Z_{D_ϵ} as obtained with TNR compared with the exact values. Note that one could easily see that the conformal spins must be half-integers in the parity -1 sector and integers in the parity $+1$ sector by observing that $(T_{D_\epsilon})^n = \Sigma^x$, the \mathbb{Z}_2 symmetry operator. However, we choose to present the numerical values for the conformal spins, including the small numerical errors, to demonstrate the accuracy of our method.

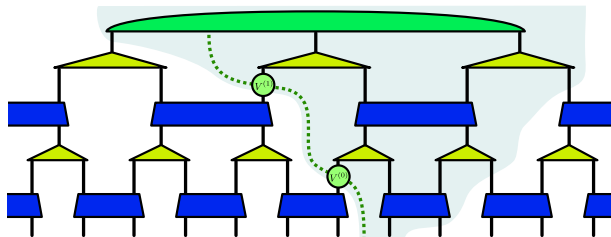


Figure 3.13: A MERA for the ground state of the quantum spin chain that has a symmetry defect in it, produced by coarse-graining the tensor network for Z_{D_ϵ} with TNR. Compared to the MERA without a defect [Fig. 3.8] the only differences here are the spin-flip matrices and the different top tensor. Like the V 's, the new top tensor can also be obtained without additional numerical work. The dotted green line traces the path of the defect through the coarse-graining. Note how at every layer of the MERA the defect is situated between two sites. We could have chosen the defect to take different paths through the network by choosing slightly different coarse-graining schemes (see Appendix B.1 for details on the scheme we use). Shown in pale blue is the causal cone of the site that is situated next to the defect. The dotted green line of the defect follows the edge of this causal cone. This is a special case of the more general situation where a defect may affect the tensors within, and only within, its causal cone. We shall encounter the more general situation in the next section, specifically in Fig. 3.19.

3.6 Duality defect D_σ

In this section, we first review the realization, on a quantum spin chain, of the Kramers-Wannier duality defect D_σ of the Ising CFT, and how one can move the defect with a local unitary operator. We then construct a tensor network representation of the duality defect D_σ for the classical partition function. We discuss how the defect can be coarse-grained following a strategy that is common to any type of line defect since, in contrast to the symmetry defect D_ϵ , we are not able to incorporate the duality defect D_σ into the bond index. Finally, we present numerical results for the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_{D_\sigma}$ of the scaling operators in the twisted partition function Z_{D_σ} .

3.6.1 Lattice representation

The duality defect D_σ is related to the Kramers-Wannier self-duality of the critical Ising model in a manner similar to how the symmetry defect D_ϵ is related to the \mathbb{Z}_2 symmetry of the model. To the best of our knowledge, its explicit realization on the classical partition function was not known (although Ref. [14] reports related work). For the quantum spin chain, the D_σ defect is realized in the Hamiltonian [46]

$$H_{D_\sigma} = - \left(\sum_{i=1}^{n-1} \sigma_i^z \sigma_{i+1}^z + \sum_{i=1}^{n-1} \sigma_i^x + \sigma_n^y \sigma_1^z \right). \quad (3.23)$$

Note that, in addition to the new term involving σ^y , the one-site term σ^x is missing from the n th site. We say that in this Hamiltonian the defect is on site n .

As before for the D_ϵ defect, we need a way to move the D_σ defect from one site to the next. The quantum Ising model can be mapped to a theory of free Majorana fermions with a Jordan-Wigner transformation. In the Majorana fermion picture the D_σ defect corresponds to one fermion missing from the chain. There it is then clear what moving the defect means. Translating this back to the spin chain language gives the two-site unitary operator [46] (acting here on sites 1 and n)

$$U_{D_\sigma} = \left[\left(R_z^{\frac{\pi}{4}} \right)_n \otimes \left(R_y^{\frac{\pi}{4}} R_x^{\frac{\pi}{4}} \right)_1 \right] \text{CZ}_{1,n}. \quad (3.24)$$

Here $R_a^\alpha = e^{i\alpha\sigma^a} = \mathbb{1} \cos(\alpha) + i\sigma^a \sin(\alpha)$ with σ^a 's being the Pauli matrices and $\text{CZ}_{1,n}$ is a controlled- Z gate $|0\rangle\langle 0|_n \otimes \mathbb{1}_1 + |1\rangle\langle 1|_n \otimes \sigma_1^z$. Which site is considered the control qubit for CZ does not matter because CZ is symmetric under swapping of the two sites. U_{D_σ} moves

the defect in the sense that

$$U_{D_\sigma} H_{D_\sigma} U_{D_\sigma}^\dagger = - \left(\sum_{i=2}^n \sigma_i^z \sigma_{i+1}^z + \sum_{i=2}^n \sigma_i^x + \sigma_1^y \sigma_2^z \right), \quad (3.25)$$

which is like H_{D_σ} but with the defect now on site 1. We will be referring to U_{D_σ} , H_{D_σ} and $U_{D_\sigma}^\dagger$ without mentioning which site the defect is on and which sites U_{D_σ} operates on as this should be clear from the context.

Next, we briefly investigate how the fusion rules come about in the quantum Hamiltonian. We take a Hamiltonian that has a D_ϵ defect on one site and a D_σ defect on another. By moving either one (or both) of the defects by conjugating with U_{D_σ} or V we can bring both of the two defects to site n . The resulting Hamiltonian is

$$H_{D_\epsilon \times D_\sigma} = - \left(\sum_{i=1}^{n-1} \sigma_i^z \sigma_{i+1}^z + \sum_{i=1}^{n-1} \sigma_i^x - \sigma_n^y \sigma_1^z \right). \quad (3.26)$$

This is related to H_{D_σ} by conjugation with the unitary σ_n^z . Therefore it is the same defect D_σ up to a local change of basis, thus demonstrating the fusion rule $D_\epsilon \times D_\sigma = D_\sigma$.

Next consider a Hamiltonian with two D_σ defects on different sites, such as

$$- \left(\sum_{i \neq 1,4} \sigma_i^z \sigma_{i+1}^z + \sum_{i \neq 1,4} \sigma_i^x + \sigma_1^y \sigma_2^z + \sigma_4^y \sigma_5^z \right). \quad (3.27)$$

If we use conjugation by U_{D_σ} to bring the defect on site 1 to site 4 we get the Hamiltonian

$$H_{D_\sigma \times D_\sigma} = - \left(\sum_{i \neq 3,4} \sigma_i^z \sigma_{i+1}^z + \sum_{i \neq 4} \sigma_i^x + \sigma_3^z \sigma_4^x \sigma_5^z \right). \quad (3.28)$$

This is like the usual Ising Hamiltonian of $n - 1$ spins but now with an extra spin (the 4th one) that is otherwise decoupled, but controls the coupling between spins 3 and 5. $H_{D_\sigma \times D_\sigma}$ is invariant under $\sigma_4^x H_{D_\sigma \times D_\sigma} \sigma_4^x$ and we can decompose it into two sectors according to the parity of the 4th spin. In the $+1$ sector the coupling between spins 3 and 5 is ferromagnetic and we have the usual Ising chain of $n - 1$ spins. In the parity -1 sector the coupling is antiferromagnetic and we get the Hamiltonian for the Ising model with D_ϵ defect. This is the sense in which $D_\sigma \times D_\sigma = D_1 + D_\epsilon$ in the quantum Ising model.

3.6.2 Tensor network representation

Since we do not know how to represent the duality defect D_σ in the classical Ising model, a priori we also do not know how to insert it into a tensor network $Z_{n,m}(A)$ [see Fig. 3.3(a)].

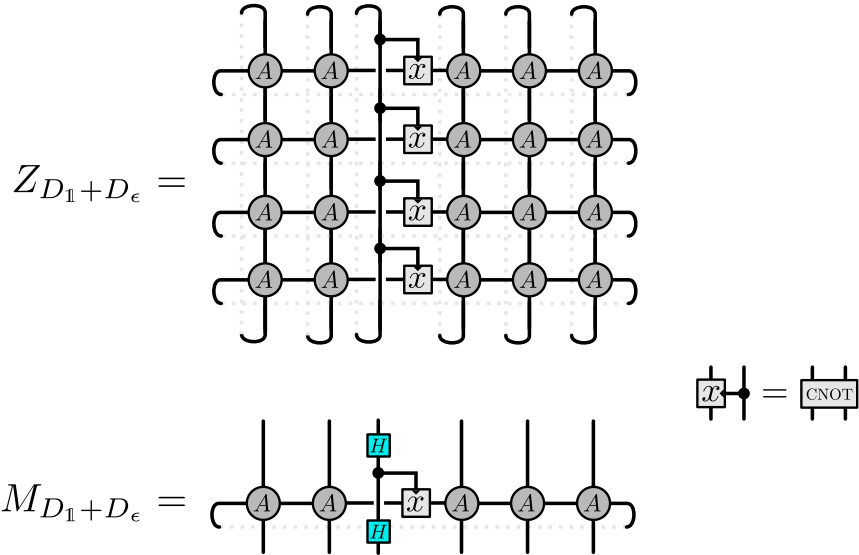


Figure 3.14: The partition function $Z_{D_1 + D_\epsilon}$ and its transfer matrix $M_{D_1 + D_\epsilon}$ that is the classical equivalent of $H_{D_\sigma \times D_\sigma}$. CNOT is a controlled-NOT gate $|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes \sigma^x$ with the dot marking the control qubit. $H = \frac{1}{2}(\sigma^x + \sigma^z)$ is the Hadamard matrix that transforms between the spin basis and the parity eigenbasis. Because $H^2 = \mathbb{1}$ the presence of the Hadamards in the transfer matrix does not affect the partition function but it ensures the transfer matrix is spin-flip invariant.

However, because of the duality between the classical transfer matrix and the quantum Hamiltonian, one would expect that if we had a transfer matrix M_{D_σ} with a D_σ line defect looping through it, conjugating M_{D_σ} with the unitary U_{D_σ} would move the defect. We can make use of this intuition and the fusion rule $D_\sigma \times D_\sigma = D_1 + D_\epsilon$ to construct a tensor for D_σ . As a reminder, in Fig. 3.15(a) we have written the operator U_{D_σ} using the tensor network notation. In the quantum Hamiltonian $H_{D_\sigma \times D_\sigma}$, having two duality defects D_σ on site k appeared as the term $\sigma_{k-1}^z \sigma_k^x \sigma_{k+1}^z$. The tensor network analog of this is an auxiliary spin in the network that, when written in the parity eigenbasis, controls the coupling along a defect line. The partition function $Z_{D_1 + D_\epsilon}$ and the transfer matrix $M_{D_1 + D_\epsilon}$ that implement such defect are shown in Fig. 3.14.

We now have a tensor representing two duality defects D_σ on the same site: the CNOT with the two Hadamard gates. We can move one of the defects away by conjugating with U_{D_σ} , thus obtaining the tensor for a single D_σ defect, as shown in Fig. 3.15(b). The explicit form of the tensor we obtain is shown in Fig. 3.15(c). We have omitted the calculation deriving this form as it is a long and uninformative exercise in using the properties of Pauli

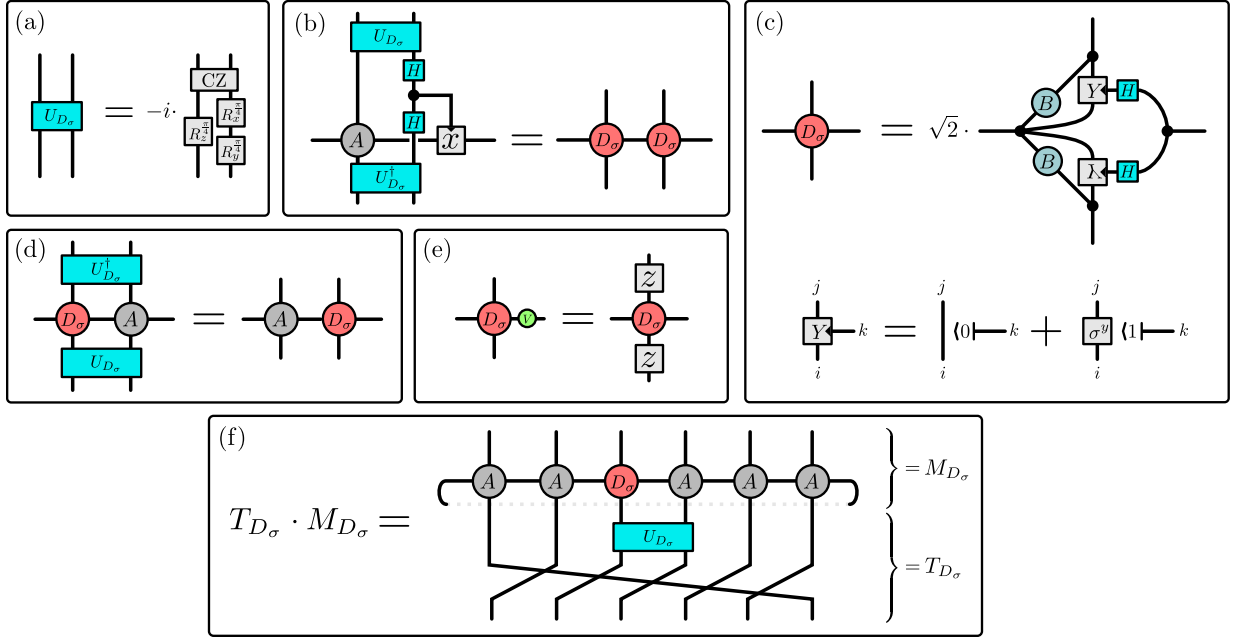


Figure 3.15: (a) The unitary that moves the D_σ defect by one site. Here $R_a^\alpha = e^{i\alpha\sigma^a} = \mathbb{1} \cos(\alpha) + i\sigma^a \sin(\alpha)$ with σ^a 's being the Pauli matrices. CZ denotes a controlled-Z gate $|0\rangle\langle 0| \otimes \mathbb{1} + |1\rangle\langle 1| \otimes \sigma^z$. Compared to the U_{D_σ} in Eq. (3.24) this one has an additional factor of $-i$. We are in fact free to multiply U_{D_σ} with an arbitrary phase, because to move a defect we multiply it with U_{D_σ} and $U_{D_\sigma}^\dagger$ and the phases cancel. Because of this phase freedom in U_{D_σ} , when extracting the conformal spins from the eigenvalues of the translation operator [see Eq. (3.12)] the conformal spins are only determined up to an additive constant. The phase has been chosen here so that the conformal spins come out correctly, which relies on us knowing at least one of the exact conformal spins of the operators in Z_{D_σ} . (b) Taking the tensor network equivalent of having two D_σ defects on the same site and moving one of them away with U_{D_σ} yields two copies of the same tensor contracted with each other. We call this tensor D_σ and identify it as the tensor representing a single duality defect. Its explicit form is shown in (c). (c) The D_σ tensor obtained using the procedure in (b). $B_{ij} = e^{\beta\sigma_i\sigma_j}$ are the Boltzmann weights of the Ising model, H is the Hadamard operator $H = \frac{1}{\sqrt{2}}(\sigma^z + \sigma^x)$, and the dots are Kronecker δ 's that fix all their indices to have the same value. Note how the tensor marked with a Y , defined at the bottom as $\mathbb{1} \otimes \langle 0| + \sigma^y \otimes \langle 1|$, is almost like a controlled- Y gate. λ denotes the same tensor but with the indices i and j transposed. (d) The D_σ defect as represented by the tensor in (c) can be unitarily moved by conjugating with U_{D_σ} . (e) The fusion rule $D_\sigma \times D_\epsilon = D_\sigma$ as it manifests in the defect tensors. z denotes the σ^z Pauli matrix. (f) M_{D_σ} composed with the translation operator T_{D_σ} that commutes with it.

matrices.

The form of the D_σ tensor shown in Fig. 3.15(c) is physically intuitive though: The Kramers-Wannier duality maps the Ising model to an equivalent model on the dual lattice (with a degree of freedom in every plaquette). Recall how the D_ϵ defect can be seen as separating two parts of the system, one of which has been transformed with a spin-flip and the other one has not. Similarly the D_σ defect is the boundary separating a part of the system that has been mapped with the Kramers-Wannier duality from the part that has not. At this boundary we would expect spins on the dual lattice side [in Fig. 3.15(c) on the right] to represent domain walls between spins of the original lattice [left, top and bottom in Fig. 3.15(c)]. The tensor $Y_{ijk} = \mathbb{1}_{ij} \delta_{0,k} + \sigma_{ij}^y \delta_{1,k}$ does exactly that: $Y_{ijk} \neq 0$ if and only if $k = 0$ and $i = j$ (no domain wall between spins i and j) or $k = 1$ and $i \neq j$ (a domain wall between spins i and j). The index k of Y_{ijk} (marked in the figure with a small arrowhead) thus directly represents a domain wall and it is related to the free index on the right by a simple Hadamard rotation. The B matrices meanwhile provide the usual Ising couplings between the spins of the original lattice.

The tensor for the duality defect D_σ fulfills the property that it moves by one lattice site under conjugation by U_{D_σ} , as illustrated in Fig. 3.15(d). We can also observe the fusion rule $D_\sigma \times D_\epsilon = D_\sigma$ by multiplying the D_σ tensor with V as in Fig. 3.15(e). The result is the same tensor D_σ multiplied by two Pauli matrices σ^z from above and below. This represents the same defect, because the Pauli matrices only provide a local change of basis.

Thus we propose the transfer matrix M_{D_σ} for the twisted partition function Z_{D_σ} to be the one in Fig. 3.15(f). The validity of this choice is ultimately confirmed by the numerical results shown below. As with the symmetry defect D_ϵ , the usual lattice translation T moves the duality defect D_σ and we need to build a generalized translation operator $T_{D_\sigma} = U_{D_\sigma} T$ that commutes with M_{D_σ} . This, too, is shown in Fig. 3.15(f).

In Fig. 3.16, we show scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_{D_\sigma}$ for the operators in Z_{D_σ} , obtained by diagonalizing $T_{D_\sigma} \cdot M_{D_\sigma}$ for 18 sites. The results reproduce the expected conformal towers, confirming that our choice of tensor D_σ indeed represents the duality defect. Again, the accuracy of the numerical estimates quickly deteriorates with increasing scaling dimensions. Next, we will discuss how to coarse-grain in the presence of a D_σ defect, which will let us reach larger system sizes and more accurate results.

3.6.3 Coarse-graining

Coarse-graining the symmetry defect D_ϵ was particularly simple because the global \mathbb{Z}_2 symmetry is explicitly realized in the individual tensors of the tensor network. This is no

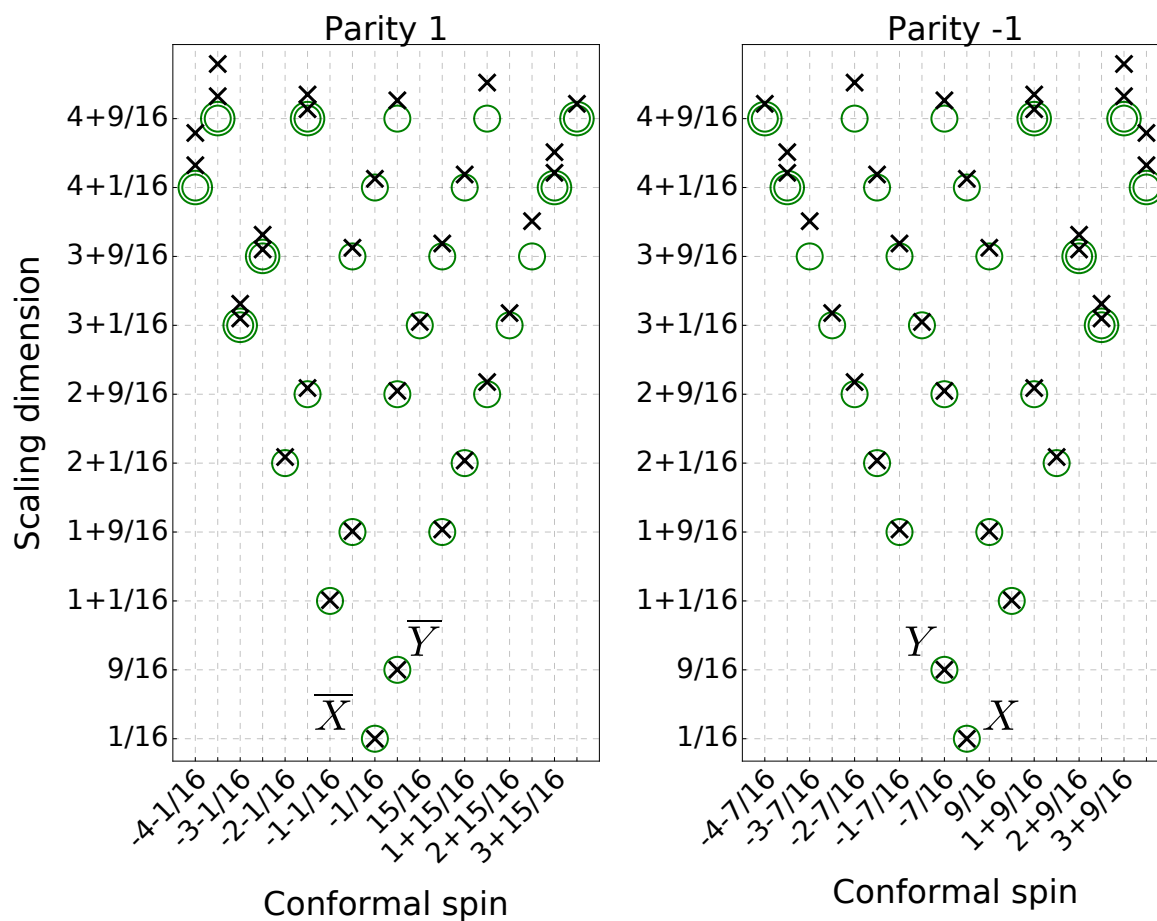


Figure 3.16: The scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first scaling operators in Z_{D_σ} obtained from exact diagonalization of a transfer matrix of $n = 18$ sites. The scaling operators are again divided by their \mathbb{Z}_2 charge. The crosses mark the numerical values that can be compared with the circles that are centered at the exact values. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair.

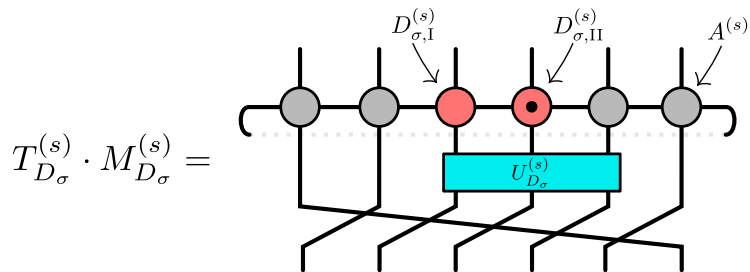


Figure 3.17: The coarse-grained transfer matrix for the D_σ defect composed with the generalized translation $T_{D_\sigma}^{(s)}$.

longer the case for the Kramers-Wannier self-duality. As a result, we need to coarse-grain the tensors for D_σ as we would coarse-grain any other line of impurity tensors (representing a generic conformal defect), that is, without being able to exploit that they correspond to a topological defect. For a sufficiently local coarse-graining scheme, such as TNR [37] (but also TRG [68] and its generalizations, see Sec. 3.8), a line defect is coarse-grained into a line defect at the next scale. The details of how we coarse-grain the line defect using TNR are explained in Appendix B.1. The same appendix also shows how the translation operator T_{D_σ} is coarse-grained in this process.

Under TNR, the duality defect D_σ is coarse-grained into a line defect with a width of two tensors. The coarse-grained transfer matrix $M_{D_\sigma}^{(s)}$ and translation operator $T_{D_\sigma}^{(s)}$ are thus as shown in Fig. 3.17. The operator $T_{D_\sigma}^{(s)} \cdot M_{D_\sigma}^{(s)}$ can then be exactly diagonalized and its eigenvalue spectrum yields the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_{D_\sigma}$ of the scaling operators with lowest dimensions in Z_{D_σ} . There is, however, a small technical subtlety in how to extract the conformal data from the spectrum that is discussed in Appendix B.4.

3.6.4 Numerical results

As discussed earlier, the primaries present in Z_{D_σ} are the ones with the conformal dimensions $(\frac{1}{2}, \frac{1}{16})$, $(0, \frac{1}{16})$, $(\frac{1}{16}, \frac{1}{2})$ and $(\frac{1}{16}, 0)$. The first two have parity +1, the latter two -1. Numerical results for the scaling dimensions and conformal spins of these primaries obtained with the TNR method are shown in Table 3.4. Similar values for some of the first descendants are shown in Fig. 3.18. The results are again in excellent agreement with the exact results even higher up in the conformal towers. These results were obtained by coarse-graining a transfer matrix of $2^5 \times (4 \times 2^5)$ $A^{(0)}$ tensors using TNR with bond dimensions $\chi = 22$ and

Primary	(h, \bar{h})	Δ_{TNR}	Δ_{exact}	s_{TNR}	s_{exact}
X	$(1/16, 0)$	0.0626656	0.0625	0.0624974	0.0625
\bar{X}	$(0, 1/16)$	0.0626656	0.0625	-0.0624974	-0.0625
Y	$(1/16, 1/2)$	0.5627685	0.5625	-0.4374828	-0.4375
\bar{Y}	$(1/2, 1/16)$	0.5627685	0.5625	0.4374828	0.4375

Table 3.4: The scaling dimensions Δ and conformal spins s for the primaries of Z_{D_σ} as obtained with TNR compared with the exact values. Note that, as with D_ϵ , one can analytically deduce the possible values of the conformal spins in the two parity sectors by observing that $(T_{D_\sigma})^{2n-1} = \frac{1}{\sqrt{2}}(\mathbb{1} + i\Sigma^x)$, where Σ^x is the global spin-flip operator. However, we choose to present the numerical values for the conformal spins, including the small numerical errors, to demonstrate the accuracy of our method.

$\chi' = 11$. Again, a slightly larger system was used for obtaining the conformal spins, see Appendix B.2.

3.6.5 MERA

As was the case with the $D_{\mathbb{1}}$ and D_ϵ defects, coarse-graining the network for Z_{D_σ} with TNR produces a MERA for the ground state of the spin chain with D_σ defect. This MERA is shown in Fig. 3.19. As the figure shows, only the tensors within the causal cone of the defect are different from the ones in the $D_{\mathbb{1}}$ MERA in Fig. 3.8. Such a MERA is known as an impurity MERA [36]. It should be noted that unlike for a general impurity MERA the impurity tensors in Fig. 3.19 can be moved with the unitary operator U_{D_σ} .

3.7 Generic conformal defects

This chapter is devoted to the study of topological conformal defects using tensor network techniques. As discussed in the introduction, the main difference between a generic (i.e. non-topological) conformal defect and a topological conformal defect is that in the latter case there is a local unitary transformation that moves the location of the defect. This allowed us to define a generalized translation operator T_D whose eigenvalues yield the conformal spins s_α associated with the defect D .

For a non-topological defect, the absence of a local unitary transformation that moves the defect implies that we can no longer define the translation operator T_D , and therefore we

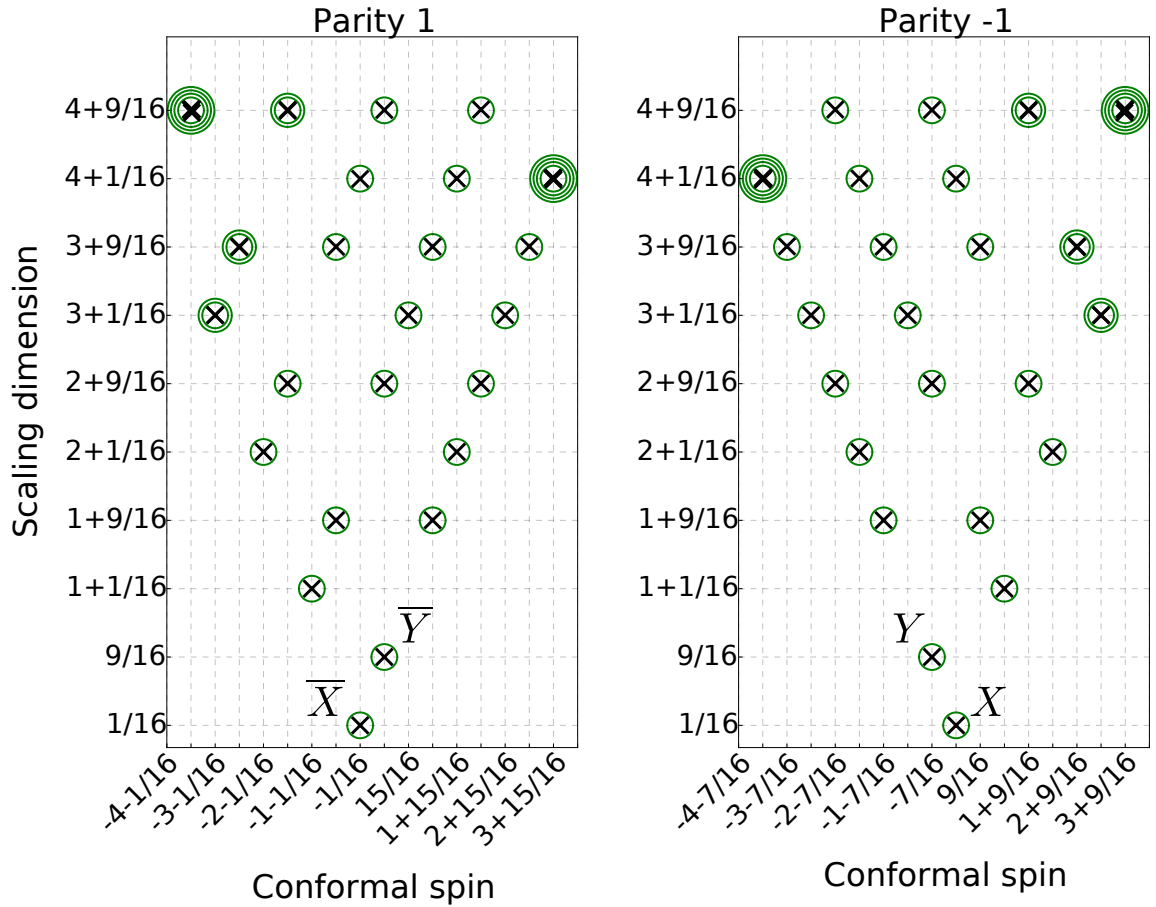


Figure 3.18: The scaling dimensions (vertical axis) and conformal spins (horizontal axis) of the first scaling operators of the two-dimensional classical Ising model with a D_σ defect as obtained with TNR. The crosses mark the numerical values that can be compared with the circles that are centered at the exact values. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair. The keen-eyed reader may notice that high up in the conformal towers some of the momenta differ from those shown in Fig. 3.16. This is because the periodicity of the momenta in these results is lower, so that for instance the conformal spins $-4 + \frac{7}{16}$ and $4 + \frac{7}{16}$ are indistinguishable.

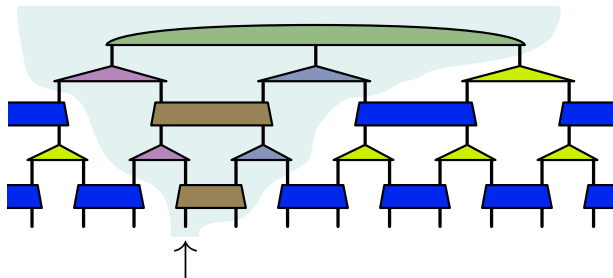


Figure 3.19: A MERA for the ground state of the quantum spin chain that has a duality defect in it, produced by coarse-graining Z_{D_σ} with TNR. The tensors in this MERA are the same ones that are used in coarse-graining the D_σ defect in Fig. B.6. Even though our notation does not reflect it, the unitaries and isometries on different layers generally differ from each other. The arrow marks the position of the defect and the pale blue region is the causal cone of the defect site.

cannot extract conformal spins. However, we can still build a tensor network representation of a non-topological defect D , and thus of the corresponding partition function Z_D on a torus and its transfer matrix M_D . In addition, we can still coarse-grain and diagonalize the transfer matrix M_D corresponding to a number n of sites much larger than what is accessible with exact diagonalization. The spectrum of M_D for a non-topological defect is no longer given by Eq. (3.10), but it still provides information about the universal properties of the system.

3.7.1 Family of defects for the Ising model

As a simple example, we consider a continuous family of conformal defects D_γ of the critical Ising model [2, 77] where the coupling between spins across the defect is proportional to a real number $\gamma \in [0, 1]$. The choice $\gamma = 0$ corresponds to no coupling across the line and thus to open boundary conditions, whereas $\gamma = 1$ corresponds to periodic boundary conditions, and thus no defect. Ignoring again finite-size corrections of higher order in n^{-1} , the eigenvalues $\lambda_\alpha(\gamma)$ of M_{D_γ} can be expressed as

$$\frac{\lambda_\alpha(\gamma)}{\lambda_0(\gamma)} = e^{-\frac{2\pi}{n} \Delta_\alpha(\gamma)}. \quad (3.29)$$

At $\gamma = 1$, Δ_α are the scaling dimensions of the Ising CFT. For general γ , Δ_α can be predicted analytically using a description in terms of fermionic operators. This will be

discussed in detail in future work (Ref. [54]). The result is that Δ_α behave linearly in $\theta = \tan^{-1} \left(\frac{1-\gamma}{1+\gamma} \right)$.

3.7.2 Numerical results

Figure 3.20 shows estimates for Δ_α as functions of θ , obtained by diagonalizing M_{D_γ} for a system of size $n = 18$, using exact diagonalization. The results are compared with the analytic values, which appear as lines. Figure 3.21 shows similar estimates obtained by diagonalizing a coarse-grained transfer matrix $M_{D_\gamma}^{(s)}$, effectively reaching a system size of roughly 4000 spins. Here we have used the same coarse-graining strategy as for the duality defect D_σ in Sec. 3.6, with bond dimension $\chi = 22$ and $\chi' = 11$. Again, the results obtained using coarse-graining show greater accuracy with respect to the analytic predictions, but even exact diagonalization gets many of the qualitative features of the spectrum correct.

3.8 Discussion

In this chapter, we have explained how to compute accurate numerical estimates of the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_D$ associated to a topological conformal defect D , or the scaling dimensions $\{\Delta_\alpha\}_D$ associated to a generic (i.e., non-topological) conformal defect D . For simplicity, we have focused on the topological conformal defects of the critical Ising model, namely the symmetry defect D_ϵ and duality defect D_σ , and have briefly considered also a family of non-topological, conformal defects. In order to improve significantly on the numerical estimates readily available through exact diagonalization, we have used a particular coarse-graining transformation: Tensor Network Renormalization (TNR). Our numerical results clearly demonstrated that tensor network techniques are a useful tool to characterize conformal defects in critical lattice models.

We conclude with a short discussion on applying this approach to other lattice models, on using other coarse-graining schemes for the same purpose, and on an alternative, more direct route to extracting conformal data from critical lattice models based on building a lattice version of the scaling operators of the theory.

3.8.1 Conformal defects in other models

The tensor network approach described in this chapter can be applied to line defects on any critical 2D classical partition function on the lattice (or point defects on any critical

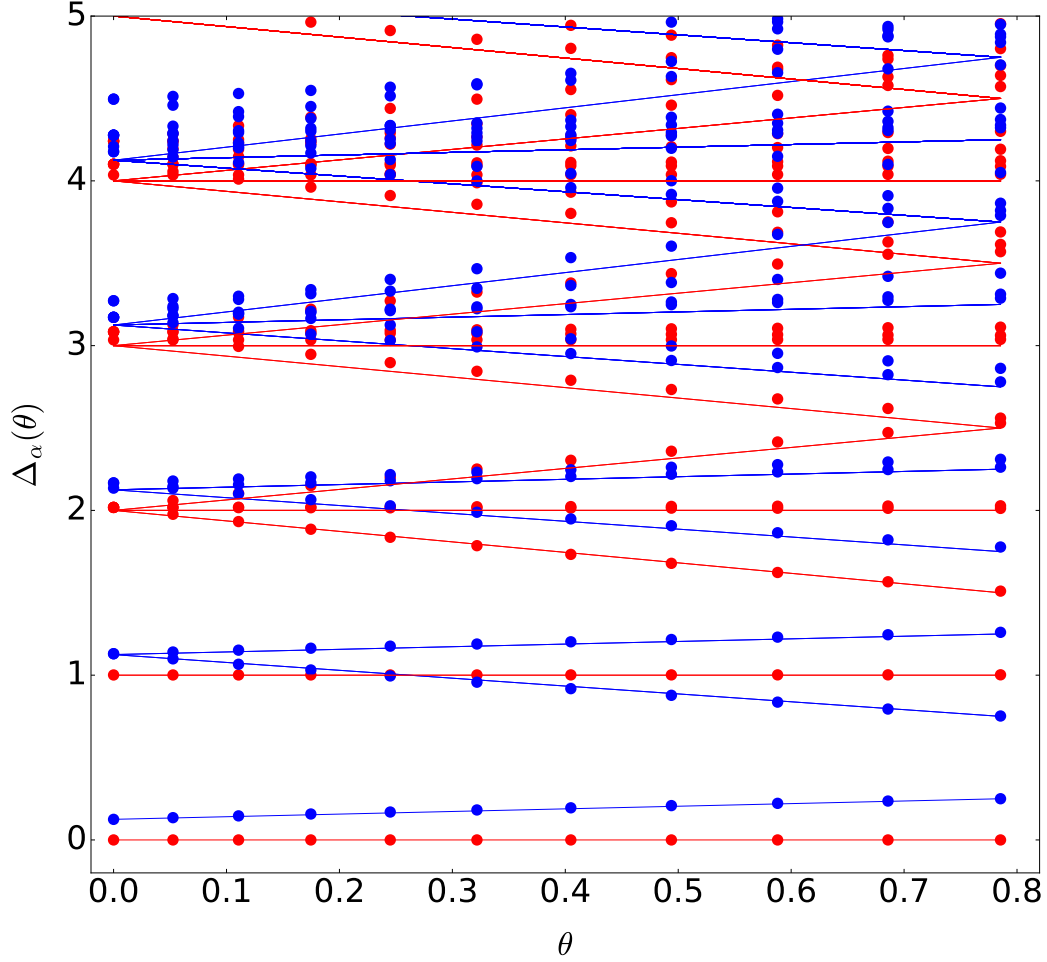


Figure 3.20: Numerical estimates for $\Delta_\alpha(\theta) = -\frac{n}{2\pi} \log\left(\frac{\lambda_\alpha(\theta)}{\lambda_0(\theta)}\right)$ obtained using exact diagonalization on a system of size $n = 18$. Here $\lambda_\alpha(\theta)$ are the eigenvalues of M_{D_γ} and $\theta = \tan^{-1}\left(\frac{1-\gamma}{1+\gamma}\right)$. The lines are the analytic predictions coming from free fermion calculations [54]. The blue and red colors mark the parity odd and parity even sectors, respectively. At the extreme left at $\theta = 0$ is the case of periodic boundary conditions, where the Δ_α are the scaling dimensions of the Ising CFT. At the extreme right at $\theta = \frac{\pi}{4}$ is the case of open boundary conditions. The numerical results clearly show qualitative agreement with the analytic values, but accuracy deteriorates significantly after the first few Δ_α 's.

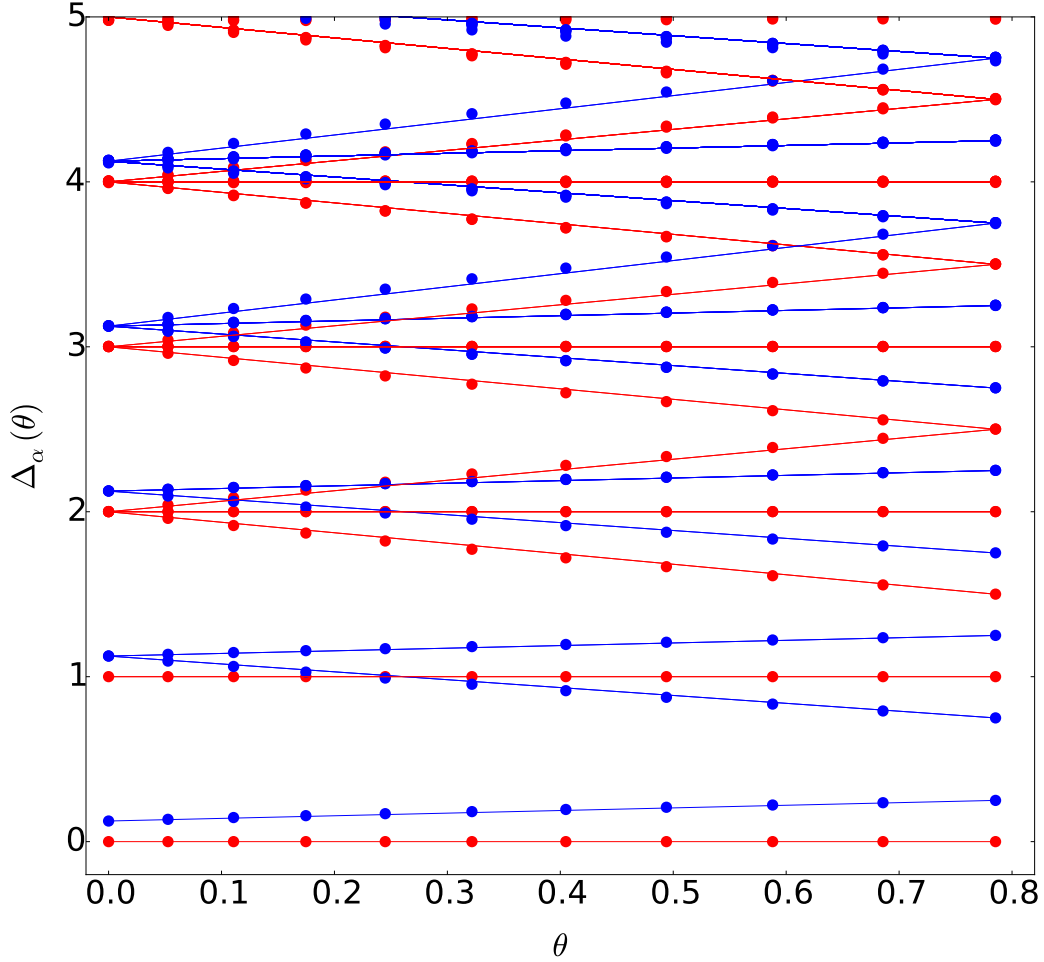


Figure 3.21: Numerical estimates for $\Delta_\alpha(\theta) = -\frac{n}{2\pi} \log \left(\frac{\lambda_\alpha(\theta)}{\lambda_0(\theta)} \right)$ obtained by coarse-graining and then diagonalizing a transfer matrix for a system of roughly 4000 spins. $\lambda_\alpha(\theta)$ are the eigenvalues of M_{D_γ} and $\theta = \tan^{-1} \left(\frac{1-\gamma}{1+\gamma} \right)$. As in Fig. 3.20, $\theta = 0$ corresponds to periodic boundary conditions, $\theta = \frac{\pi}{4}$ corresponds to open boundary conditions and the lines are the analytic predictions coming from free fermion calculations [54]. The blue and red colors mark the parity odd and parity even sectors, respectively. The numerical and analytic values are seen to agree to a high precision, owing to the fact that the coarse-graining has significantly reduced finite-size effects.

1D quantum lattice system). Let us recall what the requirements of the approach are.

The scaling dimensions $\{\Delta_\alpha\}_D$ associated to a conformal defect D can be extracted from the eigenvalues of the transfer matrix M_D for the partition function Z_D that includes that defect. Thus, we only need to have a lattice representation of the line defect D , from which to build Z_D and M_D .

In addition, for a topological conformal defect D , the associated conformal spins $\{s_\alpha\}_D$ can be extracted from the eigenvalues of a generalized translation operator T_D , built by composing a one-site translation with the application of the local unitary transformation that moves the defect back to its initial position, as explained in Secs. 3.5 and 3.6 for the symmetry defect D_ϵ and duality defect D_σ of the critical Ising model. Thus, in this case we also need to have a lattice representation of the local unitary transformation that moves the location of the defect D .

Symmetry defects, associated to a global internal symmetry group \mathcal{G} , are a type of topological conformal defect that is particularly easy to deal with within the tensor network formalism. Indeed, as we did in the case of the \mathbb{Z}_2 group of the Ising model, for each group element $g \in \mathcal{G}$ we can create a line defect D_g as a form of twisted boundary conditions by inserting copies of a unitary representation V_g along a vertical line of bond indices. This is remarkably simple when using \mathcal{G} -symmetric tensors to represent the tensor network. In this case, each tensor index is labeled by irreducible representations of \mathcal{G} , and V_g acts diagonally on the irreducible representations by placing a different complex phase on each of them. As in the Ising model, the position of the line defect can be moved by a local unitary transformation that acts on single sites. Thus, we have all the required elements to extract both scaling dimensions and conformal spins for any defect arising from an internal symmetry. This is illustrated in Appendix B.3 by presenting results for the symmetry defects of the 3-state Potts model.

Duality defects are in general more difficult to characterize, but a lattice representation is known in several models (see Ref. [14]). Finding a unitary transformation that moves the duality defect ought to also be possible after a case-by-case analysis. Then, we would be able to use the tools described in this chapter.

3.8.2 Why Tensor Network Renormalization?

In this chapter, we have employed a particular choice of coarse-graining transformation, namely, the Tensor Network Renormalization (TNR) scheme, to coarse-grain the tensor network representation of the generalized transfer matrix M_D and translation operator T_D corresponding to a topological defect D , in order to extract accurate estimates of the

associated scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_D$. However, we could have used many other coarse-graining schemes.

Indeed, any coarse-graining transformation that accurately preserves the spectra of M_D and T_D will allow us to extract the conformal data. Examples of such coarse-graining transformations include Tensor Renormalization Group (TRG) [68], Higher-Order Tensor Renormalization Group (HOTRG) [110], Tensor Entanglement-Filtering Renormalization (TEFR) [47], the Second Renormalization Group (SRG) [111], Higher-Order Second Renormalization Group (HOSRG) [110], Loop-TNR [113], TNR+ [3], and the Gilt-TNR algorithm of the previous chapter.

TRG is the simplest option, and already produces a very significant gain of accuracy with respect to exact diagonalization, since much larger systems can be considered for an equivalent computational cost, without introducing significant truncation errors, so that the estimates for $\{\Delta_\alpha, s_\alpha\}_D$ are less affected by non-universal, finite-size corrections.

The accuracy of TRG can be further increased, for the same bond dimension and similar computational cost, in a number of ways. Several improved algorithms, such as SRG and HOSRG, are based on computing an environment that accounts for the rest of the tensor network during the truncation step of the coarse-graining. The use of a global environment has the important advantage that it leads to a better truncation of bond indices, and thus to more accurate results, compared to an equivalent scheme that does not employ the environment (for instance, SRG compared to TRG, or HOSRG compared to HOTRG).

However, in the context of studying defects, the use of a global environment also has a second, less favorable implication. Since in order to truncate a given bond index we use a cost function that is aware of the whole tensor network, the resulting coarse-grained tensors will notice the presence of the defect even when the defect is away from those tensors (recall that in a critical systems correlations decay as a power-law with the distance). Consider a tensor network for the partition function Z_D with line defect D , where the line defect is initially characterized by a column of tensors that is inserted into the tensor network for the partition function Z in the absence of the defect, as we have done in this work. Then under a coarse-graining transformation such as SRG or HOSRG, the coarse-grained tensor network will consist of a collection of different tensors that depend on their distance to the defect. In other words, the representation of the defect will spread throughout the whole tensor network, instead of remaining contained in a (single or double) column of tensors, as was the case here. As a matter of fact, for topological defects this can be prevented through a careful analysis, since in some sense these defects can be made invisible to neighboring tensors (since their location can be changed through local unitary transformations). For generic conformal defects, however, one needs to consider a mixed strategy where the global

environment is used in order to coarse-grain the partition function Z in the absence of a defect, as well as the defect tensors in Z_D , but is not used in order to coarse-grain the rest of tensors in Z_D , which are recycled from Z .

A direct comparison of computational resources required by several of these approaches has shown that TNR provides significantly more accurate estimates than the methods that do not remove all short-range correlations, such as (HO)TRG and (HO)SRG, when evaluating $\{\Delta_\alpha, s_\alpha\}$ from the transfer matrix M and translation operator T in the absence of a defect, see e.g. Ref. [37]. The ultimate reason is that TNR provides a much more accurate description of the partition function when using tensors with the same bond dimension. We expect the same to be true for the estimation of $\{\Delta_\alpha, s_\alpha\}_D$ from the transfer matrix M_D and translation operator T_D in the presence of a topological defect D (or just $\{\Delta_\alpha\}_D$ from M_D in the presence of a generic conformal defect D). As shown in Chapter 2, other proper RG algorithms, such as Gilt-TNR, Loop-TNR, and TNR+, perform on par with TNR in the case clean partition functions, and we expect that this would be the case in the presence of a defect as well.

3.8.3 Alternative approach to extracting conformal data from a critical lattice model

In this chapter, we have extracted the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}_D$ associated to a topological conformal defect D by diagonalizing the transfer matrix M_D and translation operator T_D of a corresponding partition function Z_D on the lattice. This approach is based on generalizing, to the case of a line defect, the observation of Ref. [47] that the operator-state correspondence of a CFT allows us to extract the scaling dimensions and conformal spins $\{\Delta_\alpha, s_\alpha\}$ of local scaling operators of the underlying CFT by diagonalizing the transfer matrix M and translation operator T of the clean partition function Z_D on the lattice. Indeed, as discussed in Sec. 3.2, the operator-state correspondence relates the scaling operators ϕ_α of the theory with the energy and momentum eigenvectors $|\alpha\rangle$ of the Hamiltonian H and momentum P operators of the same theory (where the transfer matrix M and the translation operator T can be thought of as the exponentials of H and P , respectively), allowing the extraction of $\{\Delta_\alpha, s_\alpha\}$ directly from (a properly normalized version of) the spectra of energies and momenta $\{E_\alpha, p_\alpha\}$.

An alternative, more direct way of extracting $\{\Delta_\alpha, s_\alpha\}$ from a lattice system is also possible, by identifying a lattice version of the corresponding scaling operators ϕ_α , and studying their transformation properties under changes of scale and rotations. This alternative approach was recently made possible by the introduction of the Tensor Network

Renormalization (TNR) [37, 39]. The key of the approach is that, through the use of disentanglers that eliminate short-range correlations / entanglement from the coarse-grained partition function Z , at criticality it is possible to explicitly realize scale invariance: the tensor network before and after coarse-graining is expressed in terms of the same critical fixed-point tensor. As explained in Ref. [39], it is then possible to build a transfer matrix R *in scale*, representing a lattice version of the dilation operator of the CFT, whose eigenvectors correspond to a lattice version of the scaling operators ϕ_α , while the eigenvalues are the exponential of the scaling dimensions Δ_α (conformal spins s_α are also extracted by analysis of two-point correlators). Similar constructions are also possible with other algorithms that realize proper RG transformations.

This direct approach is computationally more challenging, since it requires ensuring that scale invariance is explicitly realized during the coarse-graining, before building and diagonalizing the scale transfer matrix R . However, it also has some remarkable advantages. On the one hand, it appears to provide even more accurate results for $\{\Delta_\alpha, s_\alpha\}$ than the diagonalization of the space-time transfer matrix M discussed in this chapter, see Appendix in Ref. [39]. Even more important is the fact that, using the explicit lattice representation of the scaling operators ϕ_α obtained from the scale transfer matrix R , we can study the fusion of two such operators into a third one, thus yielding the operator product expansion (OPE) coefficients of the CFT, which can not be obtained from the space-time transfer matrix M . These possibilities extend to the presence of defects, as demonstrated in Ref. [39] for the symmetry defect D_ϵ of the Ising model.

Finally, we also emphasize that in the context of quantum spin systems, conformal defects have already been studied using MERA. In that case, scale invariance was explicitly used to extract the scaling dimensions Δ_α attached to a conformal defect that represented an impurity, an open boundary or an interface between two critical systems, see Refs. [33] and [36].

3.8.4 Related work

A few weeks after the work presented in this chapter was originally posted on the arXiv, Aasen, Mong, and Fendley posted a paper with independent, closely related work on topological defects of the classical square-lattice Ising model [1]. While the emphasis here is on the use of tensor network methods, Ref. [1] is centered on constructing analytical models. Thus, the two nicely complement each other.

Chapter 4

Uhlmann fidelities for tensor networks

4.1 Introduction

Given two, pure many-body quantum states $|\psi\rangle$ and $|\phi\rangle$, their similarity can be quantified by their fidelity $|\langle\psi|\phi\rangle|$. It is intuitively clear, however, that there is more to say: One can discuss the similarity of the two states with regard to certain parts of the system, and make statements such as “the two states are similar at short length scales, but not at long length scales”, or “the two states only significantly differ from each other in this particular region”. This intuition is quantified by the Uhlmann fidelity of the reduced states of $|\psi\rangle$ and $|\phi\rangle$ on the subsystems in question, which for two density matrices ρ and σ is defined as [72, 96]

$$F(\rho, \sigma) = \text{Tr} \sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}. \quad (4.1)$$

When ρ and σ are reduced density matrices arising from pure states restricted to a subsystem, we call such fidelities subsystem fidelities.

As an example of a situation where subsystem fidelities are useful quantities to evaluate, consider the following experiment: Take the ground state $|E_0\rangle$ of a many-body system, disturb it locally with an operator O , that could for example flip a single spin, and let the state evolve for some time t . The result of this local quench is an evolved state $|\psi(t)\rangle = e^{itH}O|E_0\rangle$. In the top half of Fig 4.1, we can see the progression of such a quench in the Ising model, as measured by the magnetization. The effect of the disturbance can be seen propagating out ballistically. One might wonder, how the time evolved state $|\psi(t)\rangle$

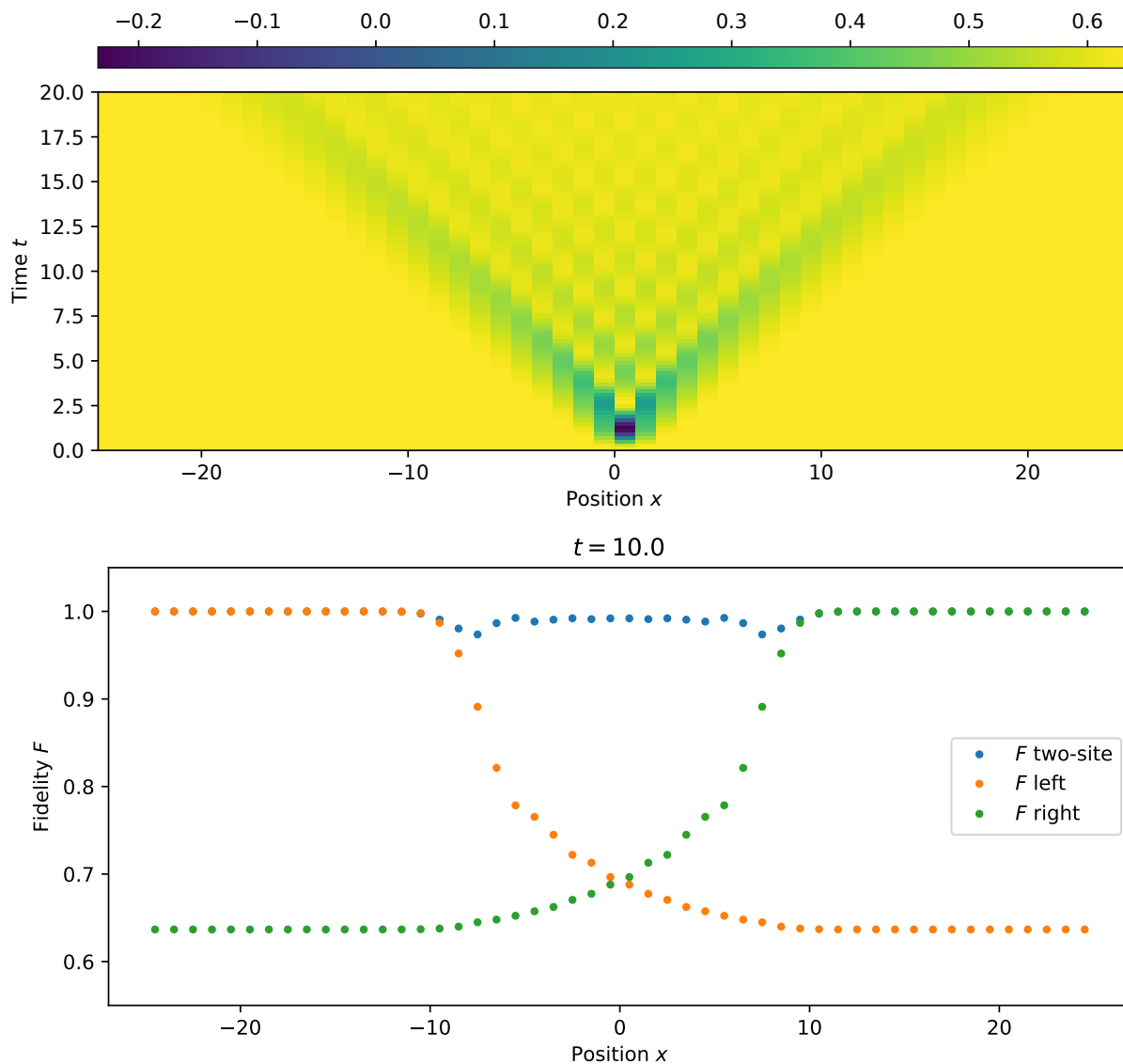


Figure 4.1: Above, the expectation value of the Pauli σ^z operator, on the locally quenched state $|\psi(t)\rangle$, as a function of position x and time t , obtained using a Matrix Product State simulation. The model in question is the critical 1D Ising model, and the quench consisted of perturbing the ground state with σ^z . This plot serves to merely illustrate the progression of the quench. Below, subsystem fidelities between the ground state $|E_0\rangle$ and the quenched state $|\psi(t = 10)\rangle$, as functions of position x . The three different fidelities plotted are the half-system Uhlmann fidelities to the left and to the right of x , and the two-site fidelity at x .

is different from the ground state $|E_0\rangle$. It is natural to guess that far away from the local disturbance $|\psi(t)\rangle$ still looks like the ground state, but closer by, the effect of the disturbance has evolved and spread. This question can be answered by resolving the overlap between $|\psi(t)\rangle$ and $|E_0\rangle$ in space, using subsystem fidelities. As an example of what such a resolution may look like, in the bottom half of Fig. 4.1 we show three different types of subsystem fidelities between $|\psi(t)\rangle$ and $|E_0\rangle$ for the case of the critical 1D Ising model: One is the fidelity between two-site reduced density matrices, positioned at different places, showing local differences. The two others are fidelities between reduced density matrices on either the left or right half of the system, where the left-right bipartition is with respect to different points on the lattice. From the profiles of these fidelities across the spin chain one can clearly see the spread of the disturbance, with the difference between the ground state $|E_0\rangle$ and the quenched state $|\psi(t)\rangle$ being the most prominent at the ballistic front, where the effect of the disturbance is propagating outwards.

Note that to be able to do this comparison between the quenched state and the ground state, we did not need to specify any observables to use as probes, nor any other further information about the system. Like other quantum information concepts that are nowadays used to analyze many-body systems, such as entanglement entropies, fidelities are entirely agnostic about the nature of the physics in the system, or even the degrees of freedom in question. Moreover, they are a more sensitive probe than any single observable, in the sense that for any observable to differ between two states, their reduced density matrices on the support of the observable must be different.

To be able to compute subsystem fidelities for many-body states, two main obstacles need to be overcome: We need a way to efficiently represent many-body states in an exponentially large state space, and second, given representations of two pure states, we need to be able to compute their fidelity on a subsystem of interest. We will use tensor network states to overcome these obstacles. In Sect. 4.2 we discuss Uhlmann fidelities in detail, and in particular how they can be formulated in terms of purifications of the reduced density matrices, while avoiding constructing the reduced density matrices themselves. Then, in Sect. 4.3, we turn our attention to tensor network states, which we can use to efficiently represent low-entanglement states of many-body systems, and for many choices of subsystems, also purifications of their reduced density matrices. We concentrate on Matrix Product States and Tree Tensor Networks, and show in detail how fidelities between two such tensor network states can be evaluated for certain choices of subsystems, at the same leading order computational cost as producing the states. In Sect. 4.4 we return to the above example of a local quench to discuss it in more detail, and present two other applications of subsystem fidelities: resolving the difference between a critical and an off-critical state as a function of scale, and quantifying convergence and the effects of limited bond dimension

in tensor network simulations. Finally, we conclude in Sect. 4.5.

Python 3 source code that implements the Matrix Product State algorithms for evaluating subsystem fidelities described in Sect. 4.3.1 and produces the results shown in Sect. 4.4, is available in the ancillary files at arxiv.org/src/1807.01640.

4.2 Uhlmann fidelity of subsystems

Let $|\psi\rangle$ and $|\phi\rangle$ be two states of the same lattice system. Consider some part of this lattice, call it M , and its complement M^c , and suppose we want to compare $|\psi\rangle$ and $|\phi\rangle$ on M . For this purpose, the natural objects to consider are the reduced density matrices $\rho = \text{Tr}_{M^c} |\psi\rangle\langle\psi|$ and $\sigma = \text{Tr}_{M^c} |\phi\rangle\langle\phi|$, and their similarity can be quantified by their Uhlmann fidelity

$$F(\rho, \sigma) = \text{Tr} \sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}. \quad (4.2)$$

The Uhlmann fidelity (4.2) is usually considered to be the most natural generalization of the overlap $|\langle\psi|\phi\rangle|$ of pure states to mixed states [72]. It fulfills Jozsa's axioms for fidelities [63], meaning that it

- is symmetric between ρ and σ
- ranges from 0 to 1, and is 1 if and only if $\rho = \sigma$,
- is invariant under unitary transformations of the state space of M ,
- reduces to $|\langle\phi|\rho|\phi\rangle|$ if $\sigma = |\phi\rangle\langle\phi|$ is pure.

Instead of trying to evaluate Eq. (4.2) directly, we will make use of Uhlmann's theorem [72]. It states that for any reduced density matrices ρ and σ , the Uhlmann fidelity can equivalently be defined as

$$F(\rho, \sigma) = \max_{|\varphi_\rho\rangle, |\varphi_\sigma\rangle} |\langle\varphi_\rho|\varphi_\sigma\rangle|, \quad (4.3)$$

where $|\varphi_\rho\rangle$ and $|\varphi_\sigma\rangle$ are purifications of ρ and σ , and the maximum is taken over all possible purifications. As an aside, note at this point, that Uhlmann's theorem makes it obvious that when ρ and σ are reduced density matrices arising from pure states restricted to a subsystem M , then the Uhlmann fidelity is monotonic in M , in that if one increases M to include more of the system, the fidelity must decrease.

To use Eq. (4.3) to evaluate Uhlmann fidelities, we need to construct the generic form of purifications of ρ and σ . A priori this may seem like a daunting task. However, concentrating

for the moment on ρ , assume that we have access to a decomposition of the form $\rho = XX^\dagger$, with some matrix X . This may for instance arise from being able to compute the Schmidt decomposition of $|\psi\rangle$ between M and M^c , or from some other structure of the state we have access to. Then, as we review in App. C.1, all purifications φ_ρ of ρ , when viewed as matrices between M and the ancilla¹, can be written in the form

$$\varphi_\rho = XW_\rho, \quad (4.4)$$

with W_ρ being some isometric matrix, meaning it fulfills $W_\rho W_\rho^\dagger = \mathbb{1}$. Given this, if $\rho = XX^\dagger$ and $\sigma = YY^\dagger$, we can write the Uhlmann fidelity between them as

$$F(\rho, \sigma) = \max_{|\varphi_\rho\rangle, |\varphi_\sigma\rangle} |\langle \varphi_\rho | \varphi_\sigma \rangle| = \max_{\varphi_\rho, \varphi_\sigma} |\text{Tr}[\varphi_\rho \varphi_\sigma^\dagger]| = \max_{W_\rho, W_\sigma} |\text{Tr}[XW_\rho W_\sigma^\dagger Y^\dagger]|, \quad (4.5)$$

where the last maximum is over all isometries W_ρ and W_σ . As we show in App. C.2, Eq. (4.5) can be further simplified to

$$F(\rho, \sigma) = \max_W |\text{Tr}[XWY^\dagger]|, \quad (4.6)$$

where W is again an isometry. Note that the dimensions of W are determined by the dimensions of X and Y , and whether W 's isometricity means $WW^\dagger = \mathbb{1}$ or $W^\dagger W = \mathbb{1}$, depends on these dimensions. Furthermore, the solution to the maximization problem of Eq. (4.6) is straight-forward (see App. C.2), and given by $W = VU^\dagger$, where $Y^\dagger X = USV^\dagger$ is the singular value decomposition (SVD) of $Y^\dagger X$. This yields for the Uhlmann fidelity, using the cyclicity of trace,

$$F(\rho, \sigma) = \max_W |\text{Tr}[WY^\dagger X]| = |\text{Tr}[VU^\dagger USV^\dagger]| = |\text{Tr} S| = \|Y^\dagger X\|_{\text{tr}}, \quad (4.7)$$

with $\|\cdot\|_{\text{tr}}$ being the trace norm.

Thus we conclude that to be able to compute the fidelity between ρ and σ , all we need is decompositions of the form $\rho = XX^\dagger$ and $\sigma = YY^\dagger$, in such a format that calculating the trace norm of $Y^\dagger X$ is computationally feasible. As we shall see in the next section, when the states $|\psi\rangle$ and $|\phi\rangle$ are described as tensor networks, this is often possible.

¹Throughout this chapter we often consider bipartite states $|\varphi\rangle \in \mathcal{H}_1 \otimes \mathcal{H}_2$ as matrices $\varphi : \mathcal{H}_1 \rightarrow \mathcal{H}_2$. Switching between the two is simply a question of changing between the space \mathcal{H}_2 and its dual, or in other words, the defining relation between $|\varphi\rangle$ and φ is that $\langle i|\varphi|j\rangle = [\langle i| \otimes \langle j|]|\varphi\rangle$ for all $|i\rangle \in \mathcal{H}_1$ and $|j\rangle \in \mathcal{H}_2$.

4.3 Evaluating subsystem fidelities for tensor network states

In this section, we describe how many subsystem fidelities can be easily evaluated for states that are described as Matrix Product States (MPSes) or Tree Tensor Networks (TTNs), using Uhlmann’s theorem as explained in Sect. 4.2.

4.3.1 Matrix Product States

Here we show how to evaluate subsystem fidelities between two MPSes, for two different choices of the subsystem: the left (or right) side of a system partitioned at some point x , and a finite window between two points x_0 and x_1 .

Left-right bipartitions

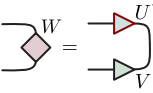

Let $|\psi\rangle$ and $|\phi\rangle$ be two Matrix Product States, given by the MPS tensors $A^{(n)}$ and $B^{(n)}$, where n labels lattice sites. Let x be a point of the lattice, that divides it into a left (L) and a right (R) half (not necessarily of the same size). Using $|\psi\rangle$ as an example, this can be expressed graphically as

$$|\psi\rangle = \cdots \underbrace{\text{---} \overset{A^{(1)}}{\circ} \text{---} \overset{A^{(2)}}{\circ} \text{---} \circ \text{---}}_L \text{---} \overset{x}{\vdots} \text{---} \underbrace{\circ \text{---} \circ \text{---} \circ \text{---}}_R \cdots \quad (4.8)$$

We now ask what is the Uhlmann fidelity between the reduced density matrices $\rho_L = \text{Tr}_R |\psi\rangle\langle\psi|$ and $\sigma_L = \text{Tr}_R |\phi\rangle\langle\phi|$, that describe the left half of the system. Here the MPSes may be finite with open boundaries, semi-infinite, or infinite. We concentrate on the left half and call $\rho_L = \rho$ and $\sigma_L = \sigma$, but the right half can be treated the same way.

Let us concentrate on finding a decomposition $\rho = XX^\dagger$, as discussed in Sect. 4.2. Given an MPS like the one in Eq. (4.8), one can follow a standard procedure [76, 79] to gauge transform it, i.e. to insert partitions of the identity on the contracted indices, to put it into the canonical form

$$|\psi\rangle = \cdots \text{---} \underbrace{\square \text{---} \circ \text{---} \square \text{---} \circ \text{---}}_{S^{(1)}} \text{---} \underbrace{\square \text{---} \circ \text{---} \square \text{---} \circ \text{---}}_{S^{(2)}} \text{---} \cdots \quad (4.9)$$

as  = , which then yields

$$F(\rho, \sigma) = \text{Tr } S = \left\| \begin{array}{c} \cdots \text{---} \square \text{---} \circ \text{---} \square \text{---} \circ \text{---} \square \text{---} \circ \\ \cdots \text{---} \square \text{---} \circ \text{---} \square \text{---} \circ \text{---} \square \text{---} \circ \end{array} \right\|_{\text{tr}}. \quad (4.17)$$

where we have bent the external legs of the matrix for which to trace norm is taken, for readability. If the two MPSes have bond dimension χ_ψ and χ_ϕ , then this matrix is of dimensions $\chi_\psi \times \chi_\phi$.

We thus arrive at the following algorithm to evaluate the Uhlmann fidelity in Eq. (4.2):

1. Gauge transform the MPSes for $|\psi\rangle$ and $|\phi\rangle$ into the canonical form, shown in Eqs. (4.9) and (4.10).²
2. Construct the matrix of Eq. (4.17).
3. Evaluate the trace norm of this matrix. This norm is the Uhlmann fidelity $F(\rho, \sigma)$.

The computational time cost of this procedure scales as $\mathcal{O}(\chi^3)$ for an MPS of bond dimension χ , which is the same as the scaling of other typical MPS operations.

Windows

Consider now the same setup as before, of two MPSes $|\psi\rangle$ and $|\phi\rangle$, but this time assume we want to evaluate the fidelity of their reduced density matrices ρ and σ , not on half the system, but on a finite window in the middle. We call this window M , and say that it is between two half-integer points on the lattice, x_0 and x_1 . We denote the parts of the lattice to the left and the right of M by L and R :

$$|\psi\rangle = \cdots \text{---} \underbrace{\circ \text{---} \circ}_{L} \text{---} \underbrace{\circ \text{---} \circ \text{---} \circ}_{M} \text{---} \underbrace{\circ \text{---} \circ}_{R} \cdots. \quad (4.18)$$

x_0 x_1

²In fact, transforming the whole MPS into the canonical form is not necessary. It is sufficient to only gauge transform the bond at x , such that it labels the Schmidt values and vectors of the left-right bipartition at x .

that for a periodic MPS, $\mathcal{O}(\chi^6)$ is the usual leading order cost [117].) This is because the isometry W is a $\chi^2 \times \chi^2$ matrix⁴, that connects both ends of the region M . W is answering the question “How large can the overlap of the two states be, if outside of M they are allowed to match each other perfectly?”, and it is answering it in a way that allows the two, disconnected ends of the system, L and R , to conspire with each other. One natural question to ask is, whether it is necessary for the left and the right ends to be correlated to maximize this overlap. This can be answered by doing the maximization of Eq. (4.23), but with the restriction that W is a tensor product of two disjoint isometries at the two ends:

$$F_d(\rho, \sigma) = \max_{W_L, W_R} \left| \begin{array}{c} \text{Diagram of a tensor network for } F_d(\rho, \sigma) \end{array} \right|, \quad (4.25)$$

with W_L and W_R isometric. The optimal choice of W_L and W_R is no longer a single straight-forward SVD, since choice of one affects the other. However, the optimization can be done by holding one of W_L and W_R fixed while optimizing the other as in Eq. (4.15), and repeating this procedure, alternating between W_L and W_R until convergence is reached. Each iteration can be done in $\mathcal{O}(\chi^3)$ time, and we find that the process usually converges in just a few iterations.

This new measure of fidelity over M from Eq. (4.25), which we call F_d or *disjoint fidelity*, can be compared to the usual Uhlmann fidelity F . First, note that since the product $W_L \otimes W_R$ is a valid choice for the isometry W in Eq (4.23), F_d is a strict lower bound for F . Second, if the region M is larger than the correlation lengths of the MPSes, then $F_d \approx F$, as one end of M is essentially uncorrelated from the other. F_d can be seen either as a cheap and conservative approximation to F , or as a separate notion of fidelity, that forbids collusion between disjoint parts of M^c in the purification.

4.3.2 Tree Tensor Networks

In this section, we concentrate on Tree Tensor Networks, or TTNs. Like MPSes, they too can be used to represent low entanglement states of 1D lattice systems. They naturally support entanglement structures that resemble a tree, and implement a notion of coarse-graining for lattice systems. More background about TTNs can be found for instance in Refs. [86, 95, 104].

⁴Or more generally $\chi_\psi^2 \times \chi_\phi^2$, if the two MPSes have different bond dimensions.

A TTN is a tensor network of the following form:


(4.26)

The open indices at the bottom are the physical lattice sites, and for simplicity of discussion we assume that all the contracted indices have bond dimension χ . The tensors in a TTN are constrained to be isometric in the sense that

$$\begin{array}{c} \triangle \\ \hline \triangle \end{array} = \mathbb{1}. \quad (4.27)$$

Written in the traditional linear algebra notation, if w , of dimensions $\chi \times \chi^2$, is the tensor of the TTN, then the isometricity condition is $w w^\dagger = \mathbb{1}$.

As with MPSes, certain subsystem fidelities are more natural and efficient to compute for TTNs than others. The characterizing criterion is, how many indices need to be cut to be able to separate a given subsystem from its complement. For MPSes, left-right bipartitions can be done by cutting only one index, and thus evaluating subsystem fidelities for them was simple and computationally cheap. Similarly for a TTN, the subsystems that can be separated from the rest by cutting a single leg allow for computing the fidelity with the lowest effort and computational cost. These subsystems are finite windows of size 2^n , that correspond to branches of the tree. This means every single-site subsystem, every second contiguous two-site block, every fourth contiguous four-site block, etc. Below are shown some examples of such subsystems, underlined in red, together with the single-leg cuts that separate them from their complements.


(4.28)

As an example, let us show how to compute the subsystem fidelity between two TTN states on the subsystem marked above as M . Call the state in Eq. (4.28) $|\psi\rangle$, and the reduced density matrix $\rho = \text{Tr}_{M^c} |\psi\rangle\langle\psi|$. Using the isometricity condition (4.27), it is easy to see that


(4.29)

where again red boundaries on tensors mark complex conjugation. Eq. (4.29) is already of the form $\rho = XX^\dagger$ that we need, but X has a very large number of columns, namely χ^n , with n being the number of vertical legs passing through the middle of the diagram, in this case 3. To improve the situation, we contract the middle part of the diagram in Eq. (4.29) and then decompose it:

The diagram shows the decomposition of a round matrix ρ . On the left, ρ is represented as a round matrix with four legs extending from its center. The top two legs are red, and the bottom two are black. This is equal to a vertical chain of two tensors. The top tensor is X^\dagger , represented by a blue circle with four legs (two red, two black). The bottom tensor is X , represented by a blue circle with four legs (two red, two black). The equation is labeled (4.30).

At the final step, the decomposition of the round matrix in the middle uses its positive semidefiniteness.

Eq. (4.30) is of the $\rho = XX^\dagger$ form, but with X now having only χ columns, which makes it computationally manageable. From this point on we can invoke Eqs. (4.5) and (4.6) as we did with MPSes, and arrive at the following expression for the Uhlmann fidelity of $\rho = \text{Tr}_{M^c} |\psi\rangle\langle\psi|$ and $\sigma = \text{Tr}_{M^c} |\phi\rangle\langle\phi|$:

The diagram shows the expression for the Uhlmann fidelity $F(\rho, \sigma)$. It is defined as the maximum over an isometry W of the trace norm of a tensor network. The tensor network consists of two vertical chains of tensors. The left chain has three blue tensors (from $|\psi\rangle$) and one red tensor (from $|\phi\rangle$). The right chain has one red tensor (from $|\psi\rangle$) and three blue tensors (from $|\phi\rangle$). A diamond-shaped tensor W connects the two chains. The entire expression is labeled (4.31).

Here, again, W is an isometry, and the green tensors are the tensors from the TTN $|\phi\rangle$, whereas the blue ones are from $|\psi\rangle$.

Constructing the matrix in Eq. (4.31) and evaluating its trace norm can be done in $\mathcal{O}(\chi^4 \log_2 L)$ time, with L being the system size and χ the bond dimension⁵. Since most TTN operations necessary to optimize such a state or evaluate observables from it scale as $\mathcal{O}(\chi^4 \log_2 L)$ or worse, evaluating these subsystem fidelities is never the bottleneck of the computation. Although we presented here how to evaluate fidelities for the subsystem M from Eq. (4.28), the same procedure applies to any subsystem that can be separated by a single cut.

As in the case of MPSes, fidelities for other subsystems can also be evaluated, although typically at higher computationally cost. Similar notions of disjoint fidelity as the one in

⁵We assume again for simplicity that $|\psi\rangle$ and $|\phi\rangle$ have the same bond dimension. This need not be the case.

Sect. 4.3.1 can also be defined, by restricting the maximization in Uhlmann’s theorem to purifications that limit correlations between disjoint parts of $M^{\mathcal{L}}$. We omit the general analysis due to its complexity, but specific choices of M can easily be analysed case-by-case.

Here we have concentrated on TTNs as they are most commonly used in many-body physics, with the isometricity constraint (4.27). Consider now relaxing the isometricity condition, and furthermore allowing the graph of contractions for the tensor network to be any tree, as opposed to the binary trees of fixed depth discussed above. This larger class of tensor networks is exactly that of acyclic graphs, meaning networks that have no closed loops. Again we can consider subsystems that can be separated by cutting a single index in the network, and the above analysis requires slight modifications, but the result remains the same: Subsystem fidelities for these subsystems, between two states that have the same tree-graph of contractions, can be evaluated efficiently and easily. The computational cost scales with a power of the bond dimension χ , that is the same as for most operations needed for the tensor networks in question (for instance, for a ternary tree, most basic operations scale as $\mathcal{O}(\chi^5)$ in χ). For more details on how to implement this for a generic tree, see Ref. [86].

4.4 Applications

The ability to evaluate Uhlmann fidelities for subsystems allows us to spatially resolve the overlaps of pure states. In this section we give some example applications of where this is useful. When a benchmark model is needed, we use the 1D Ising model with a transverse field:

$$H_{\text{Ising}} = -\frac{1}{2} \sum_i \left(\sigma_i^x \sigma_{i+1}^x + h \sigma_i^z - \frac{4}{2\pi} \mathbb{1} \right). \quad (4.32)$$

The external magnetic field h , chosen to be $h \geq 0$, distinguishes two phases, a symmetry breaking one for $h < 1$ and a disordered one for $h > 1$, which are separated by a critical point at $h = 1$. Note that the normalization in the Hamiltonian (4.32) is slightly different from what was used in the previous chapters: It has been chosen such that the ground state energy is 0 and the slope of the dispersion relation at low energies is 1.

All the results presented here are computed using MPS simulations.

4.4.1 Local quench

Consider a Hamiltonian H and its ground state $|E_0\rangle$. We may ask what happens in the time-evolution after a local quench, where the state is perturbed with some local operator

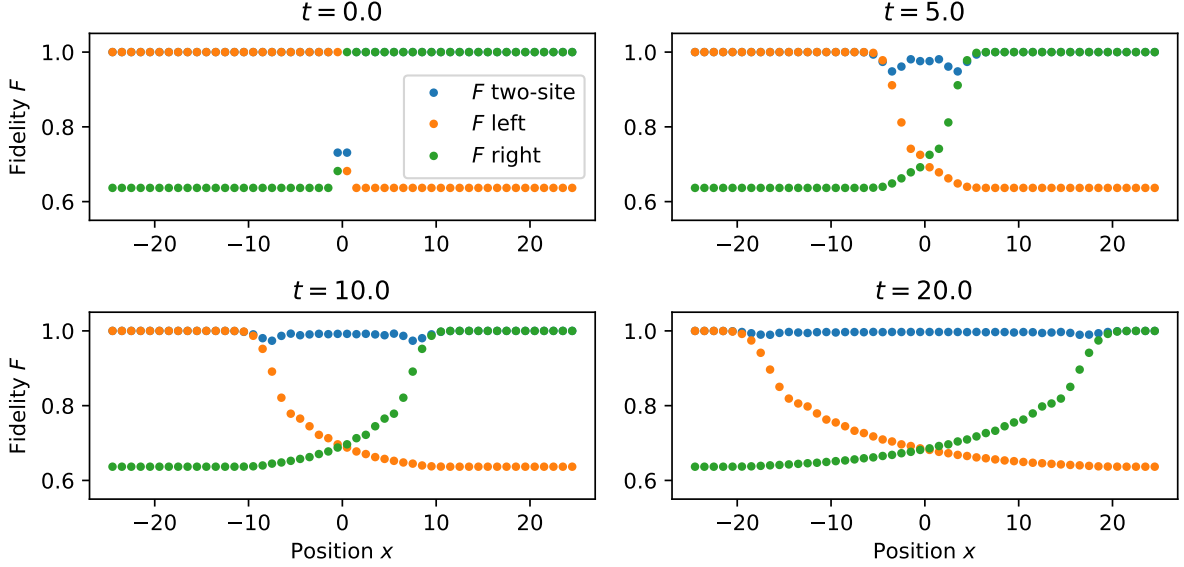


Figure 4.2: Subsystem fidelities between the ground state $|E_0\rangle$ of the infinite, critical Ising model, and the locally quenched state $e^{itH_{\text{Ising}}} \sigma_0^z |E_0\rangle$, as functions of position x , at various times t after the quench. The three different fidelities plotted are the half-system Uhlmann fidelities to the left and to the right of x , and the two-site fidelity at x .

O_x at site x , and then time evolved by time t , to reach the state $|\psi(t)\rangle = e^{itH} O_x |E_0\rangle$. Presumably the effect of the perturbation has spread to a region around x , and we may ask for instance, where is most of the perturbation “located”, and has some part of the system returned to its original state. These questions can be answered by computing subsystem fidelities between $|\psi(t)\rangle$ and the unperturbed $|E_0\rangle$.

To illustrate the idea, we perturb the ground state of the infinite, critical Ising model, represented as an MPS, with the Pauli σ^z operator, and time evolve to obtain $|\psi(t)\rangle = e^{itH_{\text{Ising}}} \sigma_0^z |E_0\rangle$, where we have chosen to call the position of the σ^z insertion the origin. For various positions x , ranging from $x \ll t$ to $t \ll x$, we then compute three different fidelities between $|\psi(t)\rangle$ and $|E_0\rangle$: The window fidelity for a two-site window around x ; the half-system fidelity for the system left of x ; and a similar half-system fidelity but for the right. These fidelities, evaluated at various times t , are shown in Fig. 4.2, and one of them, for $t = 10$, was already used as an example in the introduction.

Several observations can be made from these results. To start off, as a sanity check, it is good notice that the half-system fidelities start from 1, since far away from the

perturbation its effect is not seen, then decay monotonously as the size of the subsystem that they are computed on increases, and finally asymptote to the expectation value $|\langle\psi(t)|E_0\rangle| = |\langle E_0|\sigma^z|E_0\rangle|$. Next, note that with the normalization of the Hamiltonian chosen in Eq. (4.32), the ballistic front propagates at speed 1, and correspondingly we see that the time evolved state $|\psi(t)\rangle$ differs from the unperturbed state $|E_0\rangle$ most strongly at the fronts $x \approx t$ and $x \approx -t$. In the region $-t \ll x \ll t$, where the propagation of the perturbation has already “passed by”, the two-site fidelity reports that the state mostly looks like the ground state, but the half-system fidelities keep decreasing. Finally, notice an interesting asymmetry in the behavior of the half-system fidelities: They show a sharp decline when they meet the first ballistic front, but the final dip down to the asymptotic value at the second front is only a very small one.

4.4.2 Comparing states at different scales

Another instance where spatially resolving the overlap between two states is of interest are cases where the states are translation invariant, and similar at some length scales, while different at others. One such circumstance is comparing ground states of the same model at slightly different values of the couplings. Such ground state fidelities are useful to explore many-body physics, including first order and continuous phase transitions, as discussed in Ref. [116]. As an example, we again consider the Ising Hamiltonian (4.32) on an infinite system. We take its ground states at the critical point $h = 1.0$ and slightly in the disordered phase at $h = 1.01$, and compare their fidelities over finite windows of varying sizes.

Results are shown in Fig. 4.3. The most immediately visible feature is the disagreement between the Uhlmann fidelity (dotted blue line) and the disjoint fidelity (solid blue line). This is a consequence of the long-ranged nature of the critical state, where correlations persist over all length scales. This means that the optimization for the isometry W in Eq. (4.23) benefits from being able to bridge the two ends of the window, compared to the disjoint fidelity that forbids this. The two fidelities get closer to each other at larger distances.

Let us now concentrate on the Uhlmann fidelity, as it is a more sensitive measure of the similarity of the two states. Intuition based on the renormalization group would suggest that the almost-critical state at $h = 1.01$ should look mostly like the critical one at short length scales, and then significantly differ at long length scales. This behavior can be seen in the Uhlmann fidelity (dotted blue line) in Fig. 4.3: The slope of the curve starts mostly flat, and then dips down around the correlation length of the off-critical state, marked with the grey vertical line, before settling into a steady exponential decay. This feature

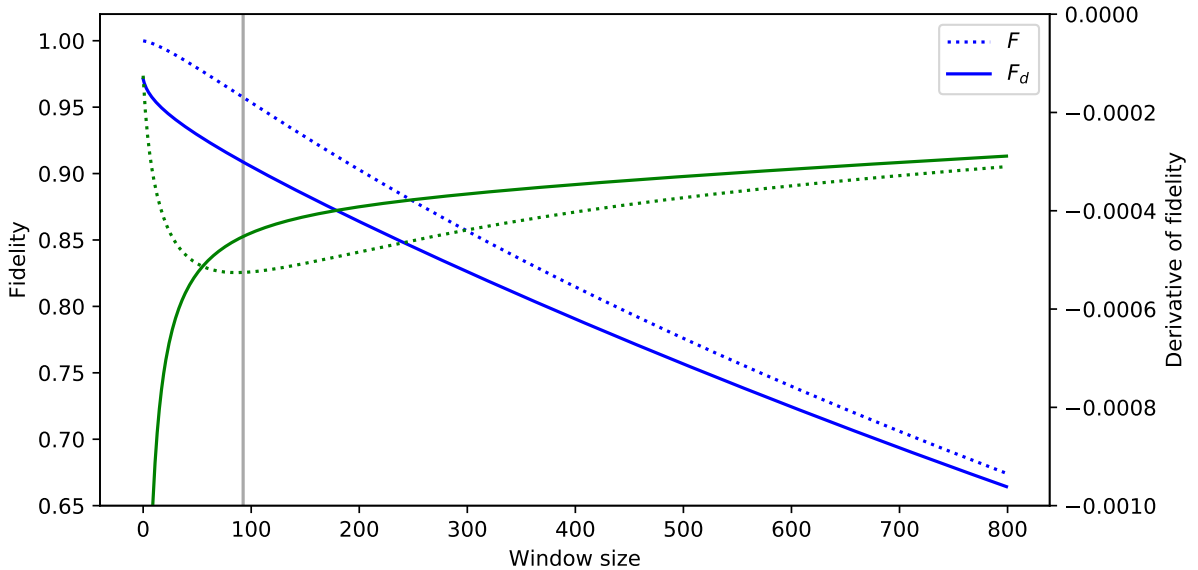


Figure 4.3: Window fidelities between the ground states of the infinite, critical Ising model at the critical point $h = 1.0$ and slightly in the disordered phase $h = 1.01$, as a function of the size of the window. The blue lines are the fidelities, dotted line for the Uhlmann fidelity F and solid line for the disjoint fidelity F_d , with their axis on the left. The green lines are the discrete derivatives of the blue lines, with their axis on the right. The vertical grey line marks the correlation length of the $h = 1.01$ ground state.

is easier to see in the derivative $\frac{\partial F}{\partial |M|}$, which is plotted with the green dotted line, and shows a minimum close to the correlation length. In other words, adding one more site to the window M causes the largest change in the fidelity when the size of M is close to the correlation length, demonstrating that the difference between the critical and the off-critical states is the most pronounced at these length scales. In results not shown here, we observe that the minimum of the derivative follows the correlation length of the off-critical state for a wide range of values for h .

4.4.3 Convergence of simulations

As with most numerical methods, tensor network algorithms typically require iterative optimizations, and have parameters that control the level of approximations, namely the bond dimensions. When simulating a given system, one needs to then ask, has the

optimization converged, and were the bond dimensions used large enough to faithfully describe the physics. In this section, we demonstrate using fidelities F and F_d to answer these kinds of questions, using convergence in bond dimension as our example.

Consider two MPSes $|\psi\rangle$ and $|\phi\rangle$, with different bond dimensions χ_ψ and χ_ϕ , that have both been optimized to minimize the energy for a given Hamiltonian H .⁶ Especially if H is critical or nearly critical, we may worry that the bond dimensions we have chosen may not be sufficient to accurately represent the ground state of H . For critical systems, we in fact know that no finite bond dimension is sufficient to describe the long distance physics correctly, but we would still hope that for distances shorter than the effective correlation lengths of $|\psi\rangle$ and $|\phi\rangle$ (imposed by the finite bond dimension), the MPSes would approximate the ground state well. To test whether our hopes are fulfilled or our bond dimensions are too small, we can compute F or F_d of $|\psi\rangle$ and $|\phi\rangle$, for finite windows of various sizes: If for a window of size $|M|$, the subsystem fidelity of $|\psi\rangle$ and $|\phi\rangle$ is far from one, the simulations can not be trusted to faithfully represent the physics at distances of order $|M|$.

In Fig. 4.4 we benchmark this idea, using again the Ising model. On its vertical axis is $1 - \text{fidelity}$, where fidelity is the Uhlmann fidelity F for the dotted lines and the disjoint fidelity F_d for the solid lines. Each line in the figure is the fidelity between two MPS approximations to the same state, at bond dimensions 10 and 20.

The green lines show fidelities at $h = 1.05$. Both the Uhlmann fidelity and the disjoint fidelity remain very close to one, which signals that bond dimension 10 is probably already sufficient for accurately describing the state, at least up to distances of 300 sites. At short distances the disjoint fidelity significantly underestimates the Uhlmann fidelity, but it is always a lower bound for it, and for high bond dimensions would be much faster to compute.

The blue lines show a similar comparison, but now at the critical point $h = 1.0$. The more entangled nature of the ground state makes it harder for the MPS to faithfully represent the state, which shows as a large difference between the $\chi_\psi = 10$ and $\chi_\phi = 20$ states, calling any long-range properties evaluated from these MPSes into question. The disjoint fidelity is seen to much more grossly underestimate the Uhlmann fidelity at short distances, due to the long-range correlations in the state. Finally, note that at short distances the Uhlmann fidelity remains quite large, which means that up to a distance of a few dozen sites, the state already has converged in bond dimension to a reasonable accuracy.

⁶Many different optimization algorithms could be used, we choose here to use imaginary time evolution, implemented using a Matrix Product Operator representation of $e^{-\tau H}$.

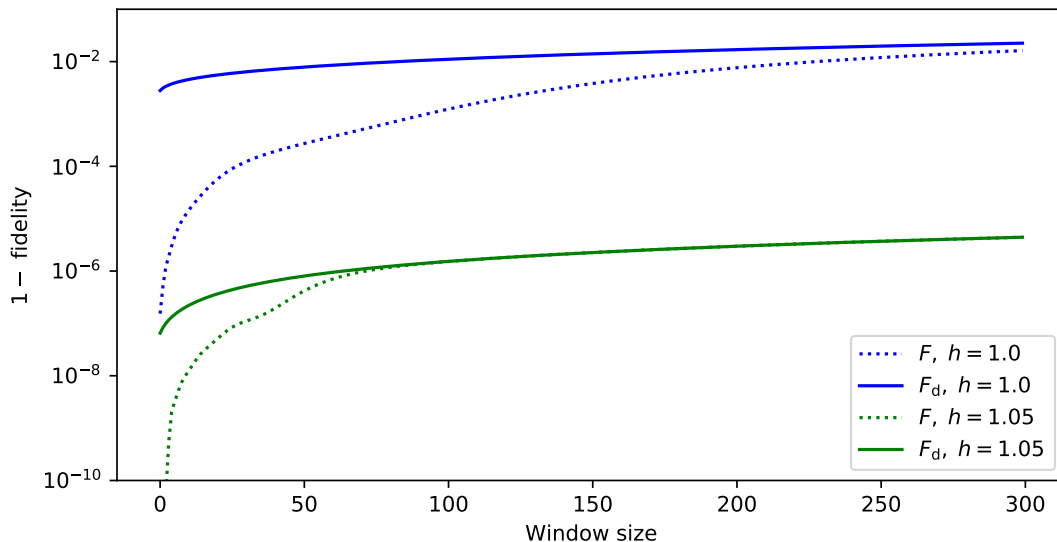


Figure 4.4: Window fidelities of pairs of MPSes, representing the same ground state, but with different bond dimensions 10 and 20, as functions of the window size $|M|$. In the optimal case these fidelities would be exactly one, so the vertical axis is the difference $1 - \text{fidelity}$, on a logarithmic scale. Dotted lines mark the Uhlmann fidelities F , solid lines are disjoint fidelities F_d . In both cases the model is the infinite Ising model, with the blue lines being for ground states of the $h = 1.0$ critical Hamiltonian, and the green ones for $h = 1.05$.

Although we have used subsystem fidelities to analyze convergence in bond dimension, similar comparisons could be made for instance between states at different stages of an optimization procedure, or at different temperatures.

4.5 Conclusion

In this chapter we explain how to compute subsystem fidelities for many-body systems using tensor network states. Such fidelities give a spatial characterization of differences between two states, that is agnostic about the nature of the degrees of freedom or the interactions. We demonstrate their usefulness with example applications: We study a local quench, resolve in scale the difference between a critical and an off-critical state, and use

similarity between states as a measure of convergence in a simulation.

Other applications, not discussed here, are also possible. For instance, one could study the effect of an impurity in the Hamiltonian, by comparing low-energy states with and without the purity. One could also study the bipartite entanglement between two halves of a system, and characterize it beyond the entanglement spectrum, by resolving the orthogonality of the Schmidt vectors: The Schmidt vectors are by definition orthogonal to each other, but their subsystems fidelities may decay in different ways as functions of the size of the subsystem, informing us of how different parts of the system contribute to the entanglement. We leave these, and possible other applications, for future study.

In Sect. 4.3 we discussed how to evaluate subsystem fidelities for Matrix Product States and Tree Tensor Networks. The reason we concentrated on these two network types is that they allow for separating certain subsystems from their complements by cutting only one or two indices. This means that the corresponding reduced density matrices have small-rank decompositions of the form $\rho = XX^\dagger$, which allowed us to evaluate Uhlmann fidelities at relatively low cost. It is worth pointing out, that for some other networks, such as MERA or PEPS, this is not the case. For instance, in a PEPS, separating an $L \times L$ region requires cutting $n \sim \mathcal{O}(L)$ indices, and the cost of computing Uhlmann fidelities grows exponentially in n .

Finally, note that both MPS and TTN are useful ansätze for 2-dimensional systems too [16, 24, 60, 91, 95, 112], and the methods we describe can be applied in that context as well.

References

- [1] D. Aasen, R. S. K. Mong, and P. Fendley. Topological defects on the lattice: I. The Ising model. *J. Phys. A: Math. Theor.*, 49:354001, Sept. 2016, 1601.07185.
- [2] I. Affleck. Quantum Impurity Problems in Condensed Matter Physics. Sept. 2008, 0809.3474.
- [3] M. Bal, M. Mariën, J. Haegeman, and F. Verstraete. Renormalization group flows of Hamiltonians using tensor networks. *Phys. Rev. Lett.*, 118:250602, 2017, 1703.00365.
- [4] M. Bal, M. M. Rams, V. Zauner, J. Haegeman, and F. Verstraete. Matrix product state renormalization. *Phys. Rev. B*, 94(20):205122, 2016, 1509.01522.
- [5] R. J. Baxter. *Exactly Solved Models in Statistical Mechanics*. Academic Press, London, 1982.
- [6] J. A. Bengua, H. N. Phien, and H. D. Tuan. Optimal Feature Extraction and Classification of Tensors via Matrix Product State Decomposition. In *2015 IEEE International Congress on Big Data*, pages 669–672, June 2015, 1503.00516.
- [7] J. C. Bridgeman and C. T. Chubb. Hand-waving and interpretive dance: an introductory course on tensor networks. *Journal of Physics A Mathematical General*, 50(22):223001, June 2017, 1603.03039.
- [8] P. Caputa, N. Kundu, M. Miyaji, T. Takayanagi, and K. Watanabe. Anti-de Sitter Space from Optimization of Path Integrals in Conformal Field Theories. *Phys. Rev. Lett.*, 119(7):071602, 2017, 1703.00456.
- [9] P. Caputa, N. Kundu, M. Miyaji, T. Takayanagi, and K. Watanabe. Liouville Action as Path-Integral Complexity: From Continuous Tensor Networks to AdS/CFT. 2017, 1706.07056.

- [10] J. L. Cardy. Operator content of two-dimensional conformally invariant theories. *Nuclear Physics B*, 270:186–204, 1986.
- [11] G. K.-L. Chan and M. Head-Gordon. Highly correlated calculations with a polynomial cost algorithm: A study of the density matrix renormalization group. *J. Chem. Phys.*, 116(11):4462–4476, 2002.
- [12] P. Chason. Toroidal boundary conditions in n-state quantum chains. *Journal of Physics A: Mathematical and General*, 22(13):2495, 1989.
- [13] Q. N. Chen, M. P. Qin, J. Chen, Z. C. Wei, H. H. Zhao, B. Normand, and T. Xiang. Partial Order and Finite-Temperature Phase Transitions in Potts Models on Irregular Lattices. *Phys. Rev. Lett.*, 107(16):165701, Oct. 2011, 1105.5030.
- [14] C. H. O. Chui, C. Mercat, W. Orrick, and P. A. Pearce. Integrable Lattice Realizations of Conformal Twisted Boundary Conditions. *Physics Letters B*, 517(3-4):429–435, Oct. 2001, hep-th/0106182.
- [15] A. Cichocki. Era of Big Data Processing: A New Approach via Tensor Networks and Tensor Decompositions. 2014, 1403.2048.
- [16] L. Cincio and G. Vidal. Characterizing topological order by studying the ground states of an infinite cylinder. *Phys. Rev. Lett.*, 110:067208, Feb 2013, 1208.2623.
- [17] P. Corboz, M. Lajkó, A. M. Läuchli, K. Penc, and F. Mila. Spin-Orbital Quantum Liquid on the Honeycomb Lattice. *Phys. Rev. X*, 2(4):041013, Oct. 2012, 1207.6029.
- [18] P. Corboz, T. M. Rice, and M. Troyer. Competing States in the t-J Model: Uniform d-Wave State versus Stripe State. *Phys. Rev. Lett.*, 113(4):046402, July 2014, 1402.2859.
- [19] P. Corboz and G. Vidal. Fermionic multi-scale entanglement renormalization ansatz. *Phys. Rev. B*, 80(16), Oct. 2009, 0907.3184.
- [20] B. Czech. Einstein’s Equations from Varying Complexity. 2017, 1706.00965.
- [21] B. Czech, G. Evenbly, L. Lamprou, S. McCandlish, X.-L. Qi, J. Sully, and G. Vidal. Tensor network quotient takes the vacuum to the thermal state. *Phys. Rev.*, B94(8):085101, 2016, 1510.07637.
- [22] B. Czech, L. Lamprou, S. McCandlish, and J. Sully. Tensor Networks from Kinematic Space. *JHEP*, 07:100, 2016, 1512.01548.

- [23] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces. *J. Stat. Mech.: Theory Exp.*, 4:04005, Apr. 2004, cond-mat/0403313.
- [24] S. Depenbrock, I. P. McCulloch, and U. Schollwöck. Nature of the spin-liquid ground state of the $S = 1/2$ Heisenberg model on the kagome lattice. *Phys. Rev. Lett.*, 109:067201, Aug 2012, 1205.4858.
- [25] P. di Francesco, P. Mathieu, and D. Sénéchal. *Conformal Field Theory*. Graduate Texts in Contemporary Physics. Springer, New York, 1997.
- [26] B. Dittrich, S. Mizera, and S. Steinhaus. Decorated tensor network renormalization for lattice gauge theories and spin foam models. *New J. Phys.*, 18(5):053009, 2016, 1409.2407.
- [27] J. Dukelsky, M. A. Martín-Delgado, T. Nishino, and G. Sierra. Equivalence of the variational matrix product method and the density matrix renormalization group applied to spin chains. *EPL (Europhysics Letters)*, 43:457–462, Aug. 1998, cond-mat/9710310.
- [28] C. Eckart and G. Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, Sep 1936.
- [29] J. Eisert, M. Cramer, and M. B. Plenio. Area laws for the entanglement entropy - a review. *Reviews of Modern Physics*, 82(1):277–306, Feb. 2010, 0808.3773.
- [30] G. Evenbly. tensors.net. [Online; accessed 2018-08-21].
- [31] G. Evenbly. Algorithms for tensor network renormalization. *Phys. Rev. B*, 95:045117, Jan 2017, 1509.07484.
- [32] G. Evenbly. Implicitly disentangled renormalization. 2017, 1707.05770.
- [33] G. Evenbly, R. N. C. Pfeifer, V. Pico, S. Iblisdir, L. Tagliacozzo, I. P. McCulloch, and G. Vidal. Boundary quantum critical phenomena with entanglement renormalization. *Phys. Rev. B*, 82(16):161107, Oct. 2010, 0912.1642.
- [34] G. Evenbly and G. Vidal. Algorithms for entanglement renormalization. *Phys. Rev. B*, 79(14):144108, Apr. 2009, 0707.1454.
- [35] G. Evenbly and G. Vidal. Entanglement Renormalization in Two Spatial Dimensions. *Phys. Rev. Lett.*, 102(18):180406, May 2009, 0811.0879.

- [36] G. Evenbly and G. Vidal. Algorithms for entanglement renormalization: boundaries, impurities and interfaces. *Journal of Statistical Physics*, 157(4-5):931–978, Dec. 2014, 1312.0303.
- [37] G. Evenbly and G. Vidal. Tensor Network Renormalization. *Phys. Rev. Lett.*, 115:180405, Oct 2015, 1412.0732.
- [38] G. Evenbly and G. Vidal. Tensor network renormalization yields the multi-scale entanglement renormalization ansatz. *Phys. Rev. Lett.*, 115(20):200401, Nov. 2015, 1502.05385.
- [39] G. Evenbly and G. Vidal. Local Scale Transformations on the Lattice with Tensor Network Renormalization. *Phys. Rev. Lett.*, 116(4):040401, Jan. 2016, 1510.00689.
- [40] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Commun. Math. Phys.*, 144(3):443–490, Mar 1992.
- [41] A. Franco-Rubio and G. Vidal. Entanglement and correlations in the continuous multi-scale renormalization ansatz. 2017, 1706.02841.
- [42] J. Fröhlich, J. Fuchs, I. Runkel, and C. Schweigert. Kramers-Wannier duality from conformal defects. *Phys. Rev. Lett.*, 93(7):070601, Aug. 2004, cond-mat/0404051.
- [43] J. Fröhlich, J. Fuchs, I. Runkel, and C. Schweigert. Duality and defects in rational conformal field theory. *Nuclear Physics B*, 763(3):354–430, Feb. 2007, hep-th/0607247.
- [44] M. Ganahl, J. Rincón, and G. Vidal. Continuous Matrix Product States for Quantum Fields: An Energy Minimization Algorithm. *Phys. Rev. Lett.*, 118(22):220402, June 2017, 1611.03779.
- [45] P. Ginsparg. Applied Conformal Field Theory. *arXiv:hep-th/9108028*, Nov. 1988.
- [46] U. Grimm and G. M. Schuetz. The spin-1/2 XXZ Heisenberg chain, the quantum algebra $U_q[sl(2)]$, and duality transformations for minimal models. *Journal of Statistical Physics*, 71(5-6):923–966, June 1993, hep-th/0111083.
- [47] Z.-C. Gu and X.-G. Wen. Tensor-Entanglement-Filtering Renormalization Approach and Symmetry Protected Topological Order. *Phys. Rev. B*, 80(15):155131, Oct. 2009, 0903.1069.
- [48] J. Haegeman, J. I. Cirac, T. J. Osborne, and F. Verstraete. Calculus of continuous matrix product states. *Phys. Rev. B.*, 88(8):085118, Aug. 2013, 1211.3935.

- [49] J. Haegeman, T. J. Osborne, H. Verschelde, and F. Verstraete. Entanglement Renormalization for Quantum Fields in Real Space. *Phys. Rev. Lett.*, 110(10):100402, 2013, 1102.5524.
- [50] M. Hauru. Comment on “Coarse-graining renormalization by higher-order singular value decomposition”. in preparation.
- [51] M. Hauru, C. Delcamp, and S. Mizera. unpublished.
- [52] M. Hauru, C. Delcamp, and S. Mizera. Renormalization of tensor networks using graph-independent local truncations. *Phys. Rev. B*, 97(4):045111, Jan. 2018, 1709.07460.
- [53] M. Hauru, G. Evenbly, W. W. Ho, D. Gaiotto, and G. Vidal. Topological conformal defects with tensor networks. *Phys. Rev. B*, 94(11):115125, Sept. 2016, 1512.03846.
- [54] M. Hauru, D. Gaiotto, G. Vidal, et al. In preparation.
- [55] M. Hauru and G. Vidal. Uhlmann fidelities from tensor networks. July 2018, 1807.01640.
- [56] P. Hayden, S. Nezami, X.-L. Qi, N. Thomas, M. Walter, and Z. Yang. Holographic duality from random tensor networks. *JHEP*, 11:009, 2016, 1601.01694.
- [57] H. He, H. Moradi, and X.-G. Wen. Modular matrices as topological order parameter by a gauge-symmetry-preserved tensor renormalization approach. *Phys. Rev. B*, 90(20):205114, Nov. 2014, 1401.5557.
- [58] M. Henkel. *Conformal Invariance and Critical Phenomena*. Springer, Berlin, Heidelberg, 1999.
- [59] Q. Hu and G. Vidal. Spacetime Symmetries and Conformal Data in the Continuous Multiscale Entanglement Renormalization Ansatz. *Phys. Rev. Lett.*, 119(1):010603, 2017, 1703.04798.
- [60] H.-C. Jiang, Z. Wang, and L. Balents. Identifying topological order by entanglement entropy. *Nature Physics*, 8:902–905, Dec. 2012, 1205.4289.
- [61] H. C. Jiang, Z. Y. Weng, and T. Xiang. Accurate Determination of Tensor Network State of Quantum Lattice Models in Two Dimensions. *Phys. Rev. Lett.*, 101(9):090603, Aug. 2008, 0806.3719.

- [62] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. [Online; accessed 2017-09-19].
- [63] R. Jozsa. Fidelity for mixed quantum states. *Journal of Modern Optics*, 41(12):2315–2323, 1994, <https://doi.org/10.1080/09500349414552171>.
- [64] L. P. Kadanoff. Variational Principles and Approximate Renormalization Group Calculations. *Phys. Rev. Lett.*, 34:1005–1008, Apr 1975.
- [65] L. P. Kadanoff, A. Houghton, and M. C. Yalabik. Variational approximations for renormalization group transformations. *J. Stat. Phys.*, 14(2):171–203, Feb 1976.
- [66] A. Klümper, A. Schadschneider, and J. Zittartz. Matrix Product Ground States for One-Dimensional Spin-1 Quantum Antiferromagnets. *Europhys. Lett.*, 24(4):293, 1993, cond-mat/9307028.
- [67] M. Levin. Real space renormalization group and the emergence of topological order. [Talk at the IPAM workshop “Topological Quantum Computing”, March 2007.](#)
- [68] M. Levin and C. P. Nave. Tensor renormalization group approach to 2D classical lattice models. *Phys. Rev. Lett.*, 99(12):120601, Sept. 2007, cond-mat/0611687.
- [69] W. Li, S.-S. Gong, Y. Zhao, S.-J. Ran, S. Gao, and G. Su. Phase transitions and thermodynamics of the two-dimensional Ising model on a distorted kagome lattice. *Phys. Rev. B.*, 82(13):134434, Oct. 2010, 1010.5297.
- [70] J.-W. Mei, J.-Y. Chen, H. He, and X.-G. Wen. Gapped spin liquid with Z_2 topological order for the kagome Heisenberg model. *Phys. Rev. B*, 95(23):235107, June 2017, 1606.09639.
- [71] M. Miyaji, T. Takayanagi, and K. Watanabe. From path integrals to tensor networks for the AdS /CFT correspondence. *Phys. Rev. D*, 95(6):066004, Mar. 2017, 1609.04645.
- [72] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.
- [73] Y. Nishio, N. Maeshima, A. Gendiar, and T. Nishino. Tensor Product Variational Formulation for Quantum Systems. Jan. 2004, cond-mat/0401115.
- [74] A. Novikov, D. Podoprikin, A. Osokin, and D. P. Vetrov. Tensorizing Neural Networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 442–450. Curran Associates, Inc., 2015, 1509.06569.

- [75] R. Orús. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals Phys.*, 349:117–158, 2014, 1306.2164.
- [76] R. Orús and G. Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Phys. Rev. B*, 78(15):155117, Oct. 2008, 0711.3960.
- [77] M. Oshikawa and I. Affleck. Boundary conformal field theory approach to the two-dimensional critical Ising model with a defect line. *Nuclear Physics B*, 495(3):533–582, June 1997, cond-mat/9612187.
- [78] F. Pastawski, B. Yoshida, D. Harlow, and J. Preskill. Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *JHEP*, 06:149, 2015, 1503.06237.
- [79] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix Product State Representations. *Quantum Information and Computation*, 7(5&6):401–430, 2007, quant-ph/0608197.
- [80] V. B. Petkova and J.-B. Zuber. Generalised twisted partition functions. *Physics Letters B*, 504(1-2):157–164, Apr. 2001, hep-th/0011021.
- [81] R. N. C. Pfeifer, J. Haegeman, and F. Verstraete. Faster identification of optimal contraction sequences for tensor networks. *Phys. Rev. E.*, 90(3):033315, Sept. 2014, 1304.6112.
- [82] T. Quella, I. Runkel, and G. M. T. Watts. Reflection and Transmission for Conformal Defects. *Journal of High Energy Physics*, 2007(04):095–095, Apr. 2007, hep-th/0611296.
- [83] S. Rommer and S. Östlund. Class of ansatz wave functions for one-dimensional spin systems and their relation to the density matrix renormalization group. *Phys. Rev. B*, 55:2164–2181, Jan. 1997, cond-mat/9606213.
- [84] S. Sachdev. *Quantum phase transitions*. Cambridge University Press, Cambridge; New York, 2011.
- [85] U. Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, Jan. 2011, 1008.3477.
- [86] Y. Shi, L. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Phys. Rev. A*, 74(2), Aug. 2006, quant-ph/0511070.

- [87] S. Singh, R. N. C. Pfeifer, and G. Vidal. Tensor network decompositions in the presence of a global symmetry. *Phys. Rev. A*, 82(5):050301, Nov. 2010, 0907.2994.
- [88] S. Singh, R. N. C. Pfeifer, and G. Vidal. Tensor network states and algorithms in the presence of a global $U(1)$ symmetry. *Phys. Rev. B*, 83(11):115125, Mar. 2011, 1008.4774.
- [89] S. Singh and G. Vidal. Tensor network states and algorithms in the presence of a global $SU(2)$ symmetry. *Phys. Rev. B*, 86(19):195114, Nov. 2012, 1208.3919.
- [90] E. Stoudenmire and D. J. Schwab. Supervised Learning with Tensor Networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 4799–4807. Curran Associates, Inc., 2016, 1605.05775.
- [91] E. M. Stoudenmire and S. R. White. Studying Two Dimensional Systems With the Density Matrix Renormalization Group. *Annu. Rev. of Condens. Matter Phys.*, 3(1):111–128, 2012, 1105.1374.
- [92] M. Suzuki. Fractal decomposition of exponential operators with applications to many-body theories and Monte Carlo simulations. *Phys. Lett. A*, 146:319–323, June 1990.
- [93] M. Suzuki. General theory of fractal path integrals with applications to many-body theories and statistical physics. *J. Math. Phys.*, 32:400–407, Feb. 1991.
- [94] B. Swingle. Entanglement Renormalization and Holography. *Phys. Rev. D*, 86(6), Sept. 2012, 0905.1317.
- [95] L. Tagliacozzo, G. Evenbly, and G. Vidal. Simulation of two-dimensional quantum systems using a tree tensor network that exploits the entropic area law. *Phys. Rev. B*, 80(23):235127, Dec. 2009, 0903.5017.
- [96] A. Uhlmann. The “transition probability” in the state space of a $*$ -algebra. *Reports on Mathematical Physics*, 9(2):273 – 279, 1976.
- [97] S. van der Walt, S. C. Colbert, and G. Varoquaux. The NumPy Array: A Structure for Efficient Numerical Computation. *Comp. Sci. Eng.*, 13(2):22–30, March 2011, 1102.1523.
- [98] F. Verstraete and J. I. Cirac. Renormalization algorithms for quantum-many body systems in two and higher dimensions. 2004, cond-mat/0407066.

- [99] F. Verstraete and J. I. Cirac. Continuous Matrix Product States for Quantum Fields. *Phys. Rev. Lett.*, 104:190405, 2010, 1002.1824.
- [100] F. Verstraete, V. Murg, and J. I. Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in Physics*, 57:143–224, Mar. 2008, 0907.2796.
- [101] F. Verstraete, D. Porras, and J. I. Cirac. Density Matrix Renormalization Group and Periodic Boundary Conditions: A Quantum Information Perspective. *Phys. Rev. Lett.*, 93:227205, Nov 2004, cond-mat/0404706.
- [102] G. Vidal. personal communication.
- [103] G. Vidal. A class of quantum many-body states that can be efficiently simulated. *Phys. Rev. Lett.*, 101(11):110501, Sept. 2008, quant-ph/0610099.
- [104] G. Vidal. Entanglement Renormalization: an introduction. *Understanding Quantum Phase Transitions. Series: Condensed Matter Physics, ISBN: 978-1-4398-0251-9. CRC Press, Edited by Lincoln Carr*, pages 115–138, 2010, 0912.1651.
- [105] G. von Gehlen and V. Rittenberg. The Ashkin-Teller quantum chain and conformal invariance. *Journal of Physics A: Mathematical and General*, 20(1):227, 1987.
- [106] S. R. White. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69(19):2863–2866, Nov. 1992.
- [107] S. R. White. Density-matrix algorithms for quantum renormalization groups. *Phys. Rev. B*, 48:10345–10356, Oct 1993.
- [108] S. R. White and A. E. Feiguin. Real-Time Evolution Using the Density Matrix Renormalization Group. *Phys. Rev. Lett.*, 93(7):076401, Aug. 2004, cond-mat/0403310.
- [109] K. G. Wilson. Renormalization Group and Critical Phenomena. I. Renormalization Group and the Kadanoff Scaling Picture. *Phys. Rev. B*, 4(9):3174–3183, Nov. 1971.
- [110] 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(4):045139, July 2012, 1201.1144.
- [111] 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(16):160601, Oct. 2009, 0809.0182.

- [112] S. Yan, D. A. Huse, and S. R. White. Spin-Liquid Ground State of the $S = 1/2$ Kagome Heisenberg Antiferromagnet. *Science*, 332:1173, June 2011, 1011.6114.
- [113] S. Yang, Z.-C. Gu, and X.-G. Wen. Loop Optimization for Tensor Network Renormalization. *Phys. Rev. Lett.*, 118:110504, Mar 2017, 1512.04938.
- [114] L. Ying. Tensor Network Skeletonization. 2016, 1607.00050.
- [115] J. F. Yu, Z. Y. Xie, Y. Meurice, Y. Liu, A. Denbleyker, H. Zou, M. P. Qin, J. Chen, and T. Xiang. Tensor renormalization group study of classical XY model on the square lattice. *Phys. Rev.*, E89(1):013308, 2014, 1309.4963.
- [116] H.-Q. Zhou, R. Orús, and G. Vidal. Ground State Fidelity from Tensor Network Representations. *Phys. Rev. Lett.*, 100(8):080601, Feb. 2008, 0709.4596.
- [117] Y. Zou, A. Milsted, and G. Vidal. Conformal data and renormalization group flow in critical quantum spin chains using periodic uniform matrix product states. Oct. 2017, 1710.05397.

APPENDICES

A

Appendices for Chapter 2

A.1 Corner-double-line tensors

In this appendix we explain in detail how the TRG, Gilt and Gilt-TNR algorithms handle the CDL toy model introduced in Refs. [47, 67] and illustrated in Fig. 2.3.

A.1.1 CDL and TRG

The progression of the TRG algorithm when applied to CDL-tensors is shown in Fig. A.1. Note how the CDL-model, even though it represents extremely local physics, which is trivial at length scales larger than the lattice spacing, is a fixed point of the TRG coarse-graining. This indicates that TRG does not properly implement the philosophy of RG.

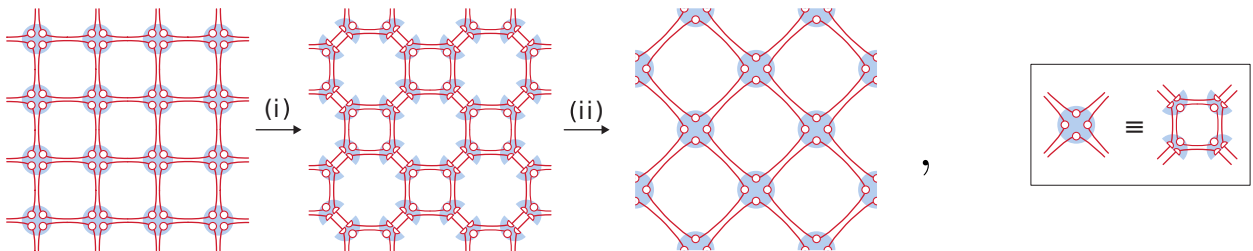


Figure A.1: The TRG algorithm, as it applies to the CDL-model. In step (i) the CDL-tensors are split with an SVD. In step (ii) the pieces are contracted together, as shown in the inset, to form the new coarse-grained tensors, which are of the CDL-form as well.

A.1.2 CDL and Gilt

Consider the following plaquette of CDL-tensors:

$$\begin{aligned}
 T &= \begin{array}{c} \text{Diagram 1} \\ \text{Diagram 2} \end{array} \equiv \begin{array}{c} \text{Diagram 3} \\ \text{Diagram 4} \end{array} \quad (\text{A.1}) \\
 \text{with } M &= \text{Diagram 5} \equiv \text{Diagram 6} .
 \end{aligned}$$

We would like to truncate the leg R using Gilt, as explained in Sect. 2.4.2. For simplicity, we assume that the CDL-matrices Diagram 5 are normalized so that their Frobenius norm is 1, although this may not hold for M .

The first observation to make, is that the SVD which yields the environment spectrum can be written as

$$\begin{aligned}
 E &= \begin{array}{c} \text{Diagram 7} \\ \text{Diagram 8} \end{array} \stackrel{\text{svd}}{\equiv} \begin{array}{c} \text{Diagram 9} \\ \text{Diagram 10} \end{array} \quad (\text{A.2}) \\
 \text{with } \text{Diagram 5} &\equiv \frac{\text{Diagram 5}}{\|\text{Diagram 5}\|} = \frac{M}{\|M\|} .
 \end{aligned}$$

Notice how the CDL-line which goes from one external leg to another and passes through R , goes through the environment spectrum S , but the loop that is internal to T does not, and is instead captured in U . For clarity, let us write down U and S explicitly:

$$\begin{aligned}
 U_i &= \begin{array}{c} \text{Diagram 11} \\ \text{Diagram 12} \end{array} \equiv \begin{array}{c} \text{Diagram 13} \\ \text{Diagram 14} \end{array} \\
 &= |j\rangle\langle k| \otimes \frac{M}{\|M\|} \quad \text{for } i = 1, \dots, \chi , \quad (\text{A.3})
 \end{aligned}$$

$$S_i = \begin{cases} \|M\| & \text{for } i = 1, \dots, \chi, \\ 0 & \text{for } i = \chi + 1, \dots, \chi^2 . \end{cases} \quad (\text{A.4})$$

The integers j and k range from 1 to $\sqrt{\chi}$, and are such that $i = \sqrt{\chi}(k - 1) + j$. For $i > \chi$, U_i are some matrices orthogonal to the other U_i 's, that play no role in this discussion.

Using the choice (2.31), we get

$$t'_i = t_i \frac{S_i^2}{\epsilon^2 + S_i^2} \approx \begin{cases} t_i & \text{for } i = 1, \dots, \chi, \\ 0 & \text{for } i = \chi + 1, \dots, \chi^2, \end{cases} \quad (\text{A.5})$$

where the approximation becomes sharp when ϵ is small compared to $\|M\|$. Recall that

$$\begin{aligned} t_i = \text{Tr } U_i &= \text{Diagram} = \text{Diagram} \\ &= \begin{cases} \frac{\text{Tr } M}{\|M\|} & \text{when } k = j, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \quad (\text{A.6})$$

Based on this, the matrix R' that shall replace R is

$$\begin{aligned} R' &= \sum_{i=1}^{\chi^2} t'_i U_i^\dagger \equiv \text{Diagram} = \text{Diagram} U^\dagger = \text{Diagram} \\ &= \frac{\text{Tr } M}{\|M\|^2} \cdot (\mathbb{1} \otimes M^\dagger). \end{aligned} \quad (\text{A.7})$$

At this point it is useful to stop and reflect on what we have shown. If Gilt is applied once to a plaquette of CDL-tensors, where the CDL-matrix on the closed loop is M , then the R' matrix that is introduced on one of the legs is $\frac{\text{Tr } M}{\|M\|^2} \cdot (\mathbb{1} \otimes M^\dagger)$.

Now recall that the next step in Gilt is decomposing R' as $R' = usv^\dagger$. If the SVD of M is $M = x\varsigma y^\dagger$, then clearly

$$u = (\mathbb{1} \otimes y), \quad (\text{A.8})$$

$$s = \frac{\text{Tr } M}{\|M\|^2} \cdot (\mathbb{1} \otimes \varsigma), \quad (\text{A.9})$$

$$v = (\mathbb{1} \otimes x^\dagger). \quad (\text{A.10})$$

Presenting the whole procedure graphically,

$$\begin{aligned} \text{Diagram} &= \text{Diagram} \\ &= \frac{\text{Tr } M}{\|M\|^2} \text{Diagram} = \frac{\text{Tr } M}{\|M\|^2} \text{Diagram}. \end{aligned} \quad (\text{A.11})$$

Hence by applying Gilt once, we have replaced the matrix M on the closed CDL-loop with $M' = \text{Tr}[M] \frac{\varsigma^2}{\|M\|^2} = \text{Tr}[M] \frac{\varsigma^2}{\text{Tr}[\varsigma^2]}$, where ς is the diagonal matrix with singular values of M . The CDL-lines between the external legs have not been affected, and no truncation error has been caused. By applying the same procedure repeatedly n times, through a simple recursion argument we find that the matrix on the closed CDL-loop becomes

$$M^{(n)} = \text{Tr}[M] \frac{\varsigma^{2n}}{\text{Tr}[\varsigma^{2n}]} . \quad (\text{A.12})$$

As n grows, assuming the largest singular value of M is not degenerate, this very quickly approaches

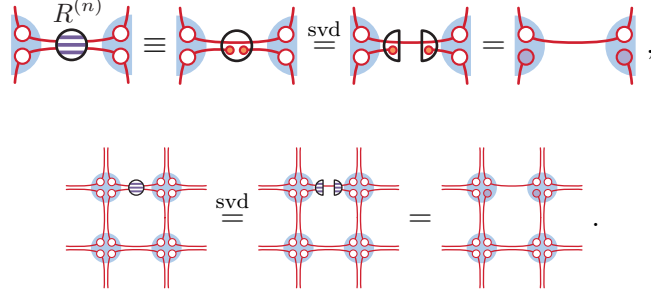
$$M^{(\infty)} = \text{Tr}[M] |0\rangle\langle 0| = \begin{pmatrix} \text{Tr}[M] & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \mathbf{0} & \\ 0 & & & \end{pmatrix} . \quad (\text{A.13})$$

Thus, by applying Gilt repeatedly, the CDL-loop can be truncated down to a bond dimension of 1, and the value it contracts to, $\text{Tr}[M]$, is stored as a scalar factor on the tensors around the loop. This concludes the proof that the Gilt method can remove CDL-type local correlations.

The only caveat above is the assumption that the dominant singular value of M is not degenerate. If it is, with some degeneracy D , then the CDL-loop is truncated down to a bond dimension of D . In working with physical models, we have not encountered this situation, nor would we expect to, since any, even small, breaking of the degeneracy would quickly get blown up by the double exponential ς^{2n} . Note also, that the failure to deal with CDL-matrices with exact degeneracy is not a shortcoming of the Gilt procedure as a whole, but a blind spot of the way we optimize for t' . Indeed, for any CDL-matrix, a choice of t' can be manually designed such that the CDL-loop is truncated away in a single step. However, the more generic approach of (2.31) for choosing t' that we use is necessary when dealing with physical models.

Above, we applied Gilt to create the matrix R' , decomposed it and absorbed the pieces of the SVD into the environment. Then this procedure was repeated. Instead of repeatedly absorbing the matrices R' into the environment, we can also combine all of them together to form a matrix $R^{(n)}$, and insert this matrix into the environment. We give details of this simple procedure in the source code. For sufficiently large n , $R^{(n)}$ has rank 1, and inserting it in the original plaquette truncates the CDL-loop in one go. This operation can

be graphically represented as



The same plaquette can be used as an environment for creating an $R^{(n)}$ matrix one-by-one on all the four legs, resulting in a complete removal of the CDL-loop:

(A.14)

For clarity of notation we have neglected to show the accumulated scalar factor $\text{Tr}[M]$.

A.1.3 CDL and Gilt-TNR

As explained in Sect. 2.5, combining Gilt with TRG leads to a proper RG transformation, called Gilt-TNR. In Fig. A.2 we show how Gilt-TNR handles the CDL-model. The analysis is simply a matter of combining what we learned in the previous two sections about how TRG and Gilt apply to CDL. Notice how at the last step the coarse-grained tensors have been reduced to bond dimension 1, i.e., scalars. This is the physically correct RG fixed point for the CDL-model.

A.2 Choosing t'

The rank of R' as a function of the coefficients t'_i in the sum $R' = \sum_{i=1}^{\chi^2} t'_i U_i^\dagger$ is a complicated cost function to optimize for. Therefore we instead use a cost function which is easier to optimize and favors similar choices of R' . This cost function is $C_{\text{norm}} = \|R'\|^2 = \text{Tr}[R' R'^\dagger] = \sum_i s_i^2$. Here, s_i are the singular values of R' (not to be confused with the environment spectrum S). Since each contribution from a singular value is positive, bringing any of the

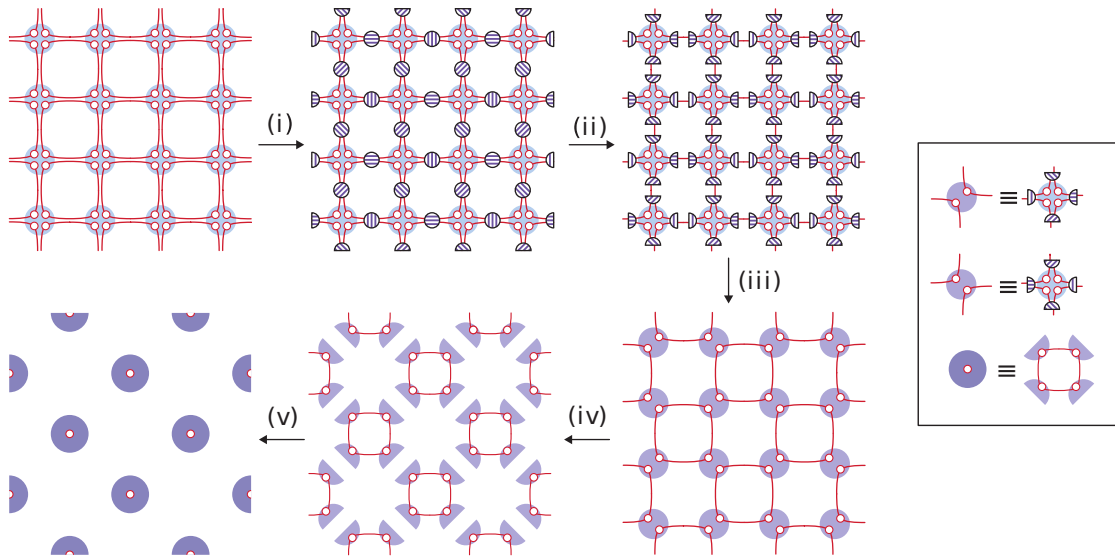


Figure A.2: Gilt-TNR algorithm applied to the CDL-model. In steps (i) – (iii), the $R^{(n)}$ matrices are inserted on all the legs, using every second plaquette as the environment, as explained in Fig. 2.4. This results in the removal of half of the CDL-loops. Note that truncating each of these loops results in a multiplicative scalar factor $\text{Tr}[M]$, which we omit from the figure. In step (iv) and (v) TRG is applied to the remaining tensors, which disposes of the remaining CDL-loops, yielding trivial tensors of bond dimension 1.

s_i close to zero, i.e., reducing the rank of R' , would bring the cost function C_{norm} down. The “blind spot” for this cost function are situations where one singular value could be brought down at the cost of making the others significantly larger. Such solutions would be heavily penalized by C_{norm} , although they could be a viable way of minimizing the rank of R' . It seems, however, that such solutions are not typically needed when truncating bonds in a network representing a partition function, as proven by the quality of the results we show in Sect. 2.6.

Using $R' = \sum_{i=1}^{\chi^2} t'_i U_i^\dagger$ and the unitarity of U , we see further that $C_{\text{norm}} = \|R'\|^2 = \text{Tr}[R'R'^\dagger] = \sum_{i=1}^{\chi^2} |t'_i|^2$. So clearly the elements t'_i that we are free to choose should be chosen to be as small as possible. However, the picture where some singular values in the environment spectrum S are exactly zero, and thus the corresponding t'_i 's can be chosen with complete freedom, is of course a simplification. In reality, the singular values are only approximately zero, and a small error is caused when replacing t with t' . This error can be quantified as

$$\begin{aligned}
C_{\text{error}} &= \left\| \begin{array}{c} \text{---} \text{---} \\ \text{---} \text{---} \\ \text{---} \text{---} \\ \text{---} \text{---} \end{array} - \begin{array}{c} \text{---} \text{---} \\ \text{---} \text{---} \\ \text{---} \text{---} \\ \text{---} \text{---} \end{array} \right\|^2 = \left\| \begin{array}{c} t' \\ | \\ S \\ | \\ \text{---} \end{array} - \begin{array}{c} t \\ | \\ S \\ | \\ \text{---} \end{array} \right\|^2 \\
&= \left\| \begin{array}{c} t' \\ | \\ S \\ | \\ \text{---} \end{array} - \begin{array}{c} t \\ | \\ S \\ | \\ \text{---} \end{array} \right\|^2 = \sum_{i=1}^{\chi^2} |t'_i - t_i|^2 S_i^2. \tag{A.15}
\end{aligned}$$

The final cost function we want to use is a weighted sum of this error term and the above-described C_{norm} term, weighted with a small constant $\epsilon^2 > 0$:

$$\begin{aligned}
C_{\text{total}} &= C_{\text{error}} + \epsilon^2 C_{\text{norm}} \\
&= \sum_{i=1}^{\chi^2} |t'_i - t_i|^2 S_i^2 + \epsilon^2 |t'_i|^2 \tag{A.16}
\end{aligned}$$

which is minimized by choosing

$$t'_i = t_i \frac{S_i^2}{\epsilon^2 + S_i^2}. \tag{A.17}$$

This is the choice of t' that we use.

The final remaining question is how to choose the coefficient ϵ , which sets the balance between the term favoring a low-rank R' and the one measuring the truncation error. The

factor $\frac{S_i^2}{\epsilon^2 + S_i^2}$ which weighs t_i in (A.17) is close to 0 when $S_i \ll \epsilon$ and close to 1 when $S_i \gg \epsilon$. Thus ϵ sets a scale below which the values in the environment spectrum are considered small enough, so that we can safely change the corresponding coefficient t'_i . Typically we find a choice of ϵ in the range 10^{-6} to 10^{-8} suitable, depending on the model and the bond dimension. The larger ϵ , the more the algorithm will truncate (the smaller χ' will be), causing a larger error.

Note that this means we cannot choose a bond dimension χ' to truncate to, but only set the threshold ϵ . Compared to other truncation algorithms, such as the truncated SVD, where the bond dimension is chosen directly, this feature has good and bad sides. The negative side is that, when this truncation is used as a part of a larger program, the computational time is harder to predict, since we do not know the bond dimensions before running the algorithm. A few values of ϵ need to be tried to see how the network in question responds. The bright side is that the algorithm naturally adapts to the ease of the problem: For the same ϵ , the truncation errors caused tend to be roughly comparable, and the bond dimension χ' then adjusts automatically according to whether the leg and the environment in question allow for easy squeezing or not.

A.3 Source code

The work in Chapter 2 is supplemented with ready-to-use source code implementing Gilt-TNR for the square lattice. It can be found at arxiv.org/src/1709.07460v1/anc. The code is written in Python 3, and makes extensive use of the NumPy and SciPy libraries [62, 97]. It can be used to reproduce all the results shown in Sect. 2.6. For details on how to do this, see the README file that comes with the code.

In addition, a different version of the code is available on GitHub at github.com/Gilt-TNR/Gilt-TNR. For this version, no permanence is guaranteed, and it remains under active development for the time being. In addition to the well-tested square lattice implementation, the GitHub repository also includes an unfinished implementation of the Gilt-TNR algorithm on the cubical lattice, that we are currently working on. We welcome contributions for the development of the code, as well as invitations to collaborate on projects that would use our code. All the source code, both on arXiv and on GitHub, is licensed under the permissive free software MIT license.

B

Appendices for Chapter 3

B.1 Tensor network renormalization

This appendix gives a quick overview of the Tensor Network Renormalization (TNR) scheme that we use and shows how to adapt it to coarse-graining the topological defects D_ϵ and D_σ of the Ising model. It also shows how the translation operators T , T_{D_ϵ} and T_{D_σ} are coarse-grained in the process.

TNR is a coarse-graining transformation for tensor networks that is based on inserting approximate partitions of unity into the network and optimizing them to minimize the truncation error. Like Gilt-TNR of Chapter 2, it removes all short-range correlations during the coarse-graining and thus manages to realize a proper renormalization group flow with the right fixed point structure. Full details of the algorithm can be found in Refs. [37] and [31] and will not be repeated here. As a summary and a reminder, Fig. B.1 shows the progression of a TNR coarse-graining step. It shows how a network $Z_{n,m}(A^{(i)})$ is coarse-grained into a new network $Z_{\frac{n}{2},\frac{m}{2}}(A^{(i+1)})$, where each tensor $A^{(i+1)}$ corresponds to four $A^{(i)}$'s. χ and χ' mark the dimensions of the bonds. They control the accuracy of the approximations done in the algorithm but also the computational cost: Higher bond dimensions mean smaller truncation errors but require more memory and computation time.

All the tensors used in coarse-graining the Ising model are \mathbb{Z}_2 invariant in the following sense: For every bond in the network there exists a unitary matrix representation V of the non-trivial element of \mathbb{Z}_2 , called the spin-flip matrix of that bond, such that for any tensor t in the network, multiplying all the legs of t with the appropriate spin-flip matrix gives back

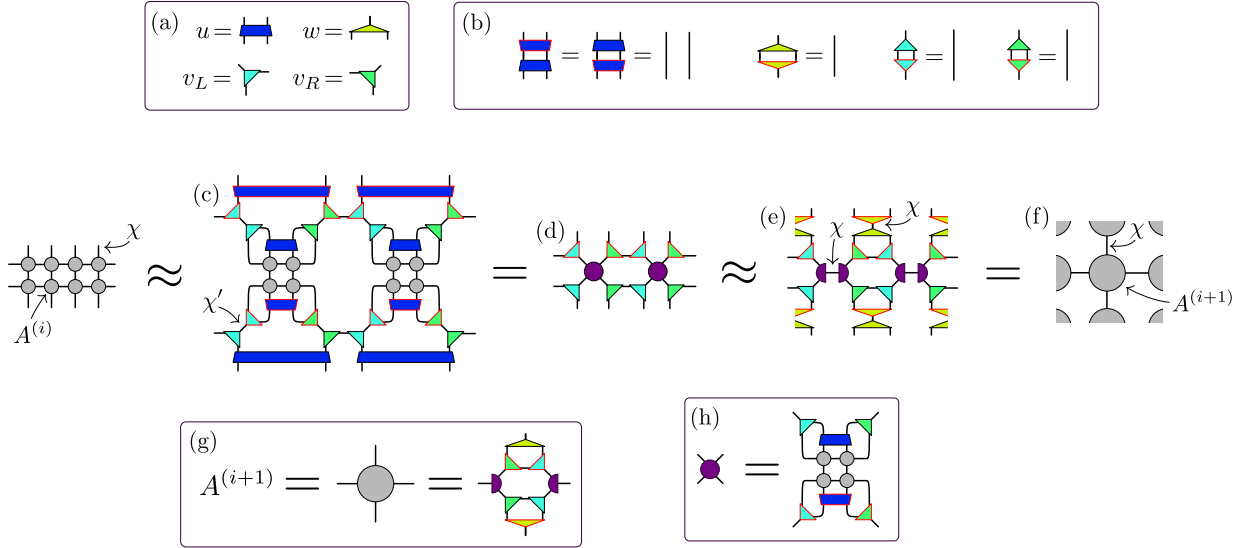


Figure B.1: An overview of the Tensor Network Renormalization procedure that we use. The network is assumed to repeat as exactly similar in all directions. χ and χ' are the (maximum allowed) dimensions of the bonds they point at and all similar bonds in the network. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. (a) The tensors used in TNR. (b) The u tensor is unitary whereas v_L , v_R and w are isometric. (c) Insertion of u 's, v_L 's and v_R 's as approximate partitions of unity. The fact that v_L and v_R are not unitary introduces a truncation error. u , v_L and v_R are optimized to minimize this error. (d) Some of the tensors have been contracted together to the one defined in (h). Because the network repeats itself in all directions the u 's and u^\dagger 's at the edges of the network have disappeared: they cancel with their counterparts in the next, similar blocks of tensors above and below the one shown. (e) Another approximate partition of unity has been inserted into the network as uw^\dagger and the purple tensor has been split into two using a truncated singular value decomposition, as is done in Levin & Nave's Tensor Renormalization Group [68]. (f) Tensors in (e) are contracted to form the new network of tensors $A^{(i+1)}$, defined in (g).

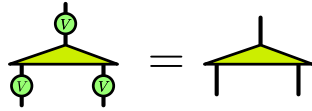


Figure B.2: The \mathbb{Z}_2 invariance property of the w tensor (in yellow). V is the spin-flip matrix of the index it is on. Note that all the three V 's in this figure may in fact be different matrices, although our naming convention does not reflect this. Similar properties hold for all the tensors involved in coarse-graining the Ising model.

t again. The spin-flip matrices for the different bonds are in general different, but we use V or $V^{(i)}$ to denote all of them: The bond they are on defines the correct V unambiguously. Figure B.2 illustrates, as an example, the \mathbb{Z}_2 invariance of the w tensor. Information on how such symmetry properties can be implemented and how to make computational use of them can be found in Refs. [87, 88, 89].

If we coarse-grain a transfer matrix multiplied with a lattice translation by two sites, the translation is coarse-grained into translation by one site at the next scale, as shown in Fig. B.3. More generally, a translation by an even number of sites n is coarse-grained into a translation by $\frac{n}{2}$ sites at the next scale. Because of this, the translation operator $T^{(s)}$ [Fig. 3.6, back in Sec. 3.3] is a translation by 2^s sites at the original scale. Thus the eigenvalues of $T^{(s)}$ are $\left(e^{\frac{2\pi i}{2^s n_s} s \alpha}\right)^{2^s} = e^{\frac{2\pi i}{n_s} s \alpha}$ and the conformal spins are determined modulo n_s . To obtain the conformal spins with more possible values we can do an additional coarse-graining step on $T^{(s)} \cdot M^{(s)}$, as explained in Appendix B.2.

B.1.1 Coarse-graining the D_ϵ defect

The TNR coarse-graining explained above can accommodate for a D_ϵ defect without any changes. As explained in Sec. 3.5, a D_ϵ defect in a tensor network is realized by having spin-flip matrices on a string of bonds. Figure B.4 shows how, using the \mathbb{Z}_2 invariance property of all the tensors in the network, such a defect coarse-grains into a similar string of spin-flip matrices at the next scale. Hence, we can coarse-grain as if there was no defect, and at any scale i we can insert a D_ϵ defect into the system as a string of spin-flip matrices $V^{(i)}$, that are the representations of the non-trivial element of \mathbb{Z}_2 under which the matrix $A^{(i)}$ is \mathbb{Z}_2 invariant.

In Sec. 3.5, we introduced the generalized translation operator T_{D_ϵ} that is the proper notion of translation for a system with a D_ϵ defect. Figure B.5 shows how the generalized two-site translation operator is coarse-grained into a generalized one-site translation at the

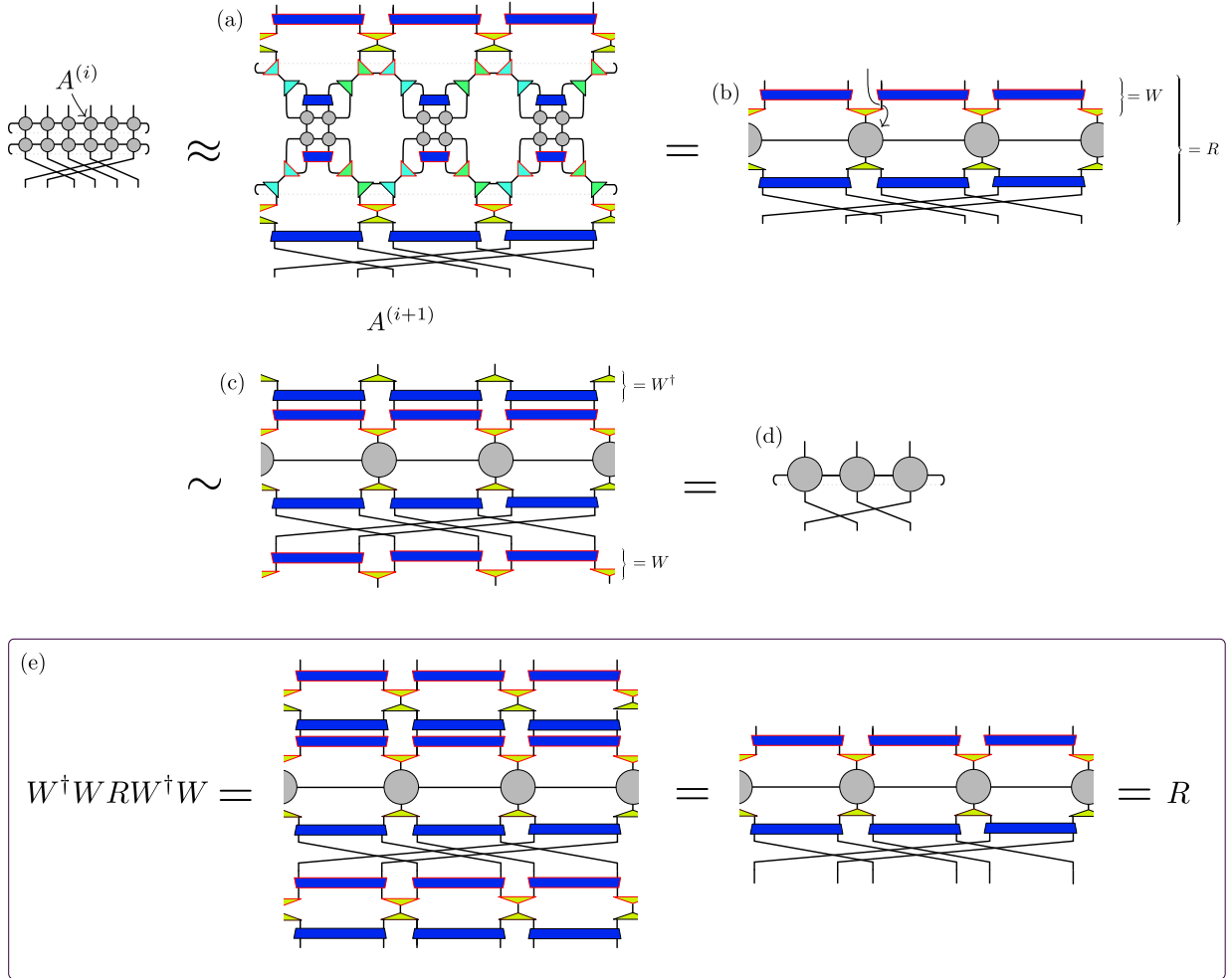


Figure B.3: One step of coarse-graining for the operator $T^{(i)} \cdot M^{(i)}$. The lattice translation by two sites gets coarse-grained into a translation by one site at the next scale. Periodic boundaries in the horizontal direction are assumed even when they are not explicitly shown. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. (a) The first step of the TNR algorithm has been applied to the tensors $A^{(i)}$. (b) A number of tensors are contracted to form $A^{(i+1)}$. We also define the operators W and R . (c) The operator R has been conjugated with W . \sim denotes that the operators in (b) and (c) have the spectrum, or in other words that WRW^\dagger has the same spectrum as R . This is true because, although W is isometric and not unitary, it acts like a unitary on M , as shown in (e). (d) The unitaries and isometries cancel and we are left with the coarse-grained transfer matrix and translation operator.

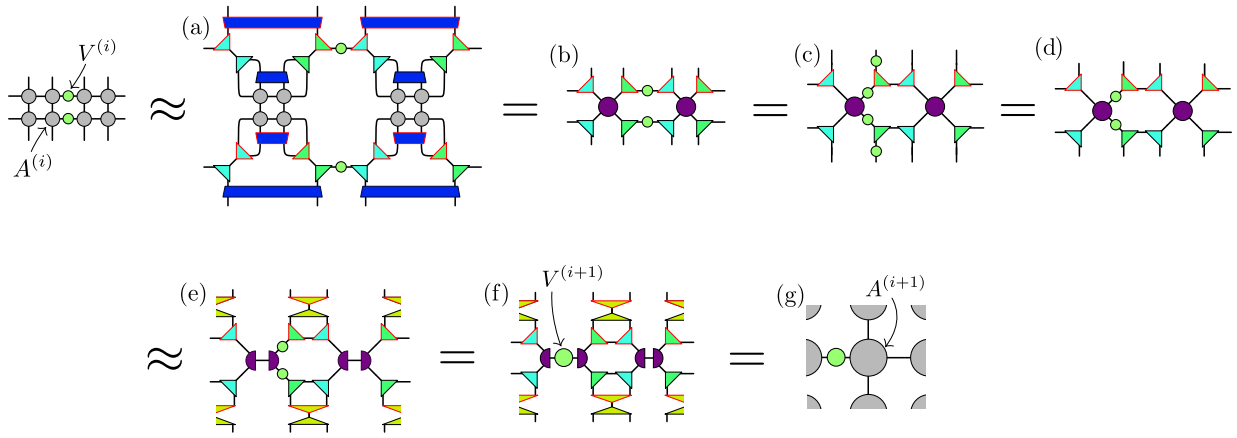


Figure B.4: Coarse-graining a system with a D_ϵ defect in it. By repeatedly using the \mathbb{Z}_2 invariance of the different tensors and the fact that $(V^{(i)})^2 = \mathbb{1}$ we can take the string of spin-flip matrices $V^{(i)}$ “through” the TNR procedure to the next scale. In the figure, the network is assumed to repeat as exactly similar in all directions. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. Steps (a), (b), (e) and (g) are the same steps as taken in the usual TNR algorithm [see Fig. B.1]. At steps (c) and (f) we have moved the spin-flip matrices to different legs using the \mathbb{Z}_2 invariance of the tensors and the fact that $(V^{(i)})^2 = \mathbb{1}$. At step (d) the spin-flip matrices at the top and the bottom have canceled with similar matrices coming from the parts of the network above and below the one shown here.

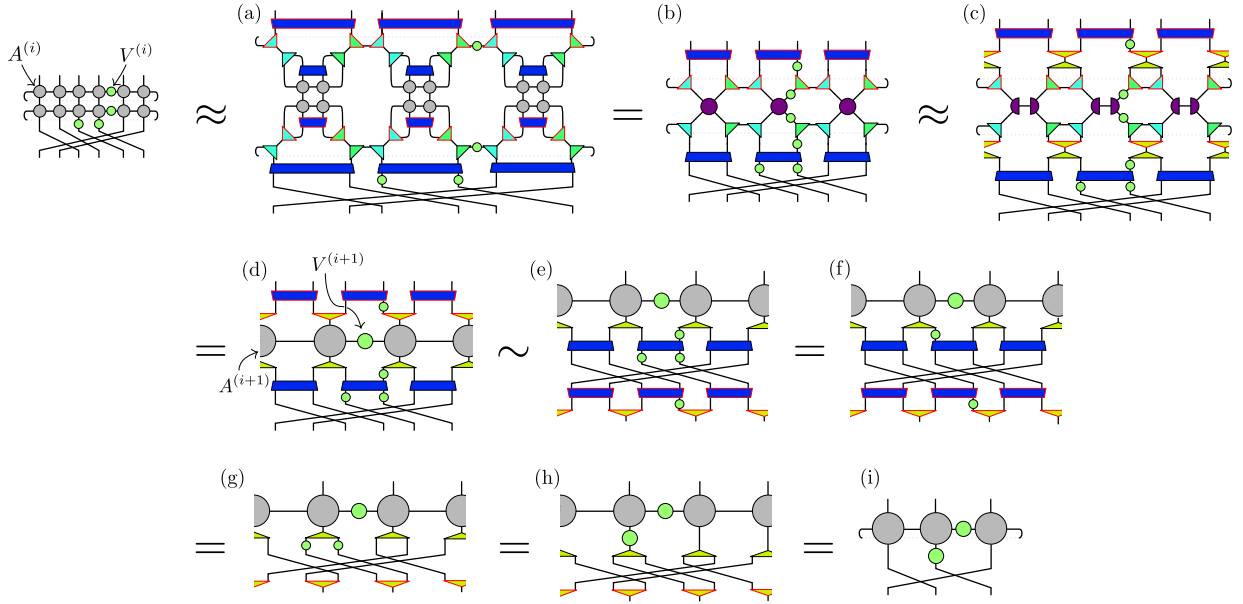


Figure B.5: One step of coarse-graining for the operator $T_{D_\epsilon}^{(i)} \cdot M_{D_\epsilon}^{(i)}$. The generalized translation by two sites gets coarse-grained into a generalized translation by one site at the next scale. Periodic boundaries in the horizontal direction are again assumed even when they are not explicitly shown. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. Steps (a-d) are a repetition of what was done in Fig. B.4, with some of the intermediate steps omitted. \sim again denotes that the two operators in (d) and (e) have the same spectrum. That they indeed do have the same spectrum is not obvious, but can be shown with an argument exactly like the one used in Fig. B.3(e). We have omitted this argument for the sake of brevity. In steps (f) and (h) we use the \mathbb{Z}_2 invariance of u and w to move the spin-flip matrices in the network. In steps (g) and (i) the u and w tensors cancel with their complex conjugates because of their unitarity and isometricity.

next scale. The procedure is the same as what we did with the usual lattice translation in Fig. B.3, except for the spin-flip matrices that get transferred to the next scale. Again, this generalizes to a generalized translation by an even number of sites n that is coarse-grained into a generalized translation by $\frac{n}{2}$ sites at the next scale. Thus we know how to construct the generalized translation operator $T_{D_\epsilon}^{(i)}$ at any scale with no additional numerical work needed.

B.1.2 Coarse-graining the D_σ defect

Coarse-graining the symmetry defect D_ϵ is easy because of the \mathbb{Z}_2 invariance of the tensors in the TNR procedure. The Kramers-Wannier symmetry of the critical Ising model is not similarly explicitly realized in the individual tensors and does not help in coarse-graining the D_σ defect. Instead we treat the string of D_σ tensors as we would treat any other string of impurity tensors. Coarse-graining such a string can be done using a variation of the usual TNR scheme. The only change is that the unitaries and isometries that are in the vicinity of the D_σ tensors need to be optimized for their respective environments [31]. The tensors elsewhere in the network are the same ones used in coarse-graining D_1 . This modified scheme is shown in Fig. B.6.

In Fig. B.6, we depict the D_σ defect as a two-tensor-wide string. This is because regardless of whether we start with a two-tensor or one-tensor-wide string it is coarse-grained into a two-tensor-wide string at the next scale.¹ For the first coarse-graining step we can simply choose $D_{\sigma, I}^{(0)} = D_\sigma$ and $D_{\sigma, II}^{(0)} = A$ and at all later scales the defect will consist of two-tensor-wide string.

The generalized translation operator $T_{D_\sigma}^{(i)}$ is coarse-grained as in Fig. B.7. The principle is the same as for $T^{(i)}$ and $T_{D_\epsilon}^{(i)}$, but implementation is not quite as simple because the defect-moving unitary $U_{D_\sigma}^{(i)}$ needs to be contracted with what is essentially a MERA ascending superoperator [34] to get the $U_{D_\sigma}^{(i+1)}$ at the next scale. The computational time to perform this coarse-graining of $U_{D_\sigma}^{(i)}$ scales asymptotically as $\mathcal{O}(\chi^{10})$, whereas all the other steps in the TNR procedure (with or without a defect) are at most $\mathcal{O}(\chi^7)$ [31], making this step the computational bottleneck.

The argument in Fig. B.7 can be generalized to show that a generalized translation by an even number of sites n is coarse-grained to a generalized translation by $\frac{n}{2}$ sites at the next scale, with the unitary $U^{(i+1)}$ being the one defined in Fig. B.7(i). Analogously to how a one-tensor string of D_σ 's becomes a two-tensor string under coarse-graining, the two-site operator U_{D_σ} coarse-grains into a three-site operator $U_{D_\sigma}^{(1)}$. Because of this in Fig. B.7 $U_{D_\sigma}^{(i)}$ is shown as a three-site operator that is then coarse-grained into a three-site operator $U_{D_\sigma}^{(i+1)}$.

¹This is related to the causal cone of the binary MERA that is two sites wide for certain operators [38, 103].

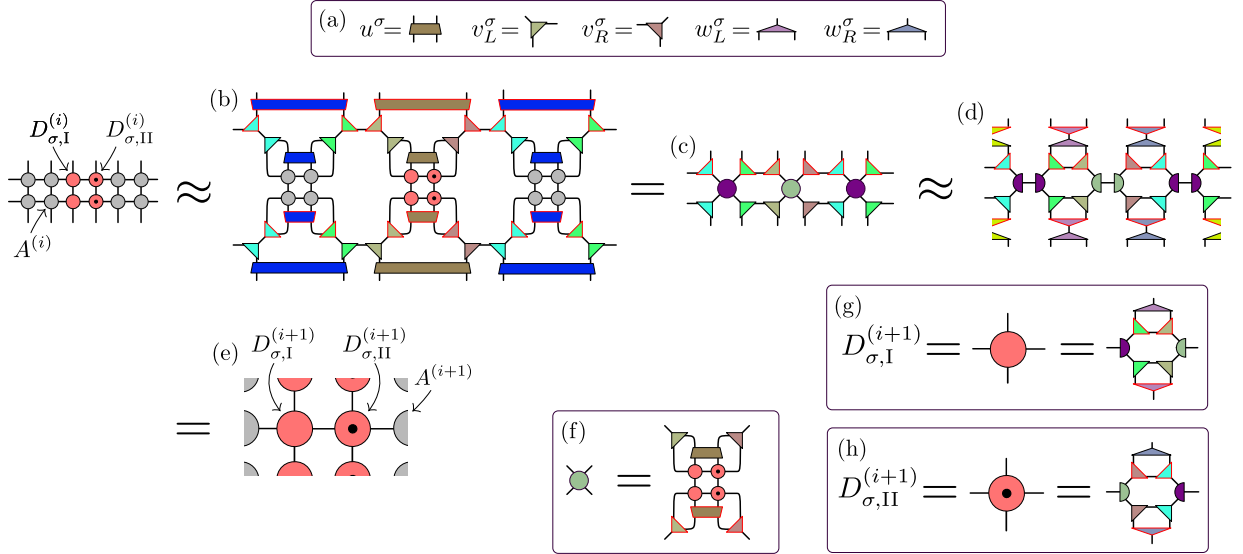


Figure B.6: The TNR procedure to coarse-grain a line defect of two tensors. The network is assumed to repeat as similar above and below the part shown, where as on the left and the right it is assumed to consist of tensors $A^{(i)}$. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. (a) The new tensors used in coarse-graining the defect. They have the same unitarity and isometricity properties as their counterparts in the usual scheme [see Fig. B.1]. (b – d, f) The steps of the usual TNR algorithm are repeated for the new isometries and unitaries that coarse-grain the defect. The isometries and unitaries are optimized to again minimize truncation errors, but for the environments that include the $D_{\sigma,I}^{(i)}$ and $D_{\sigma,II}^{(i)}$ tensors. (e) The different tensors are contracted together to form $D_{\sigma,I}^{(i+1)}$ and $D_{\sigma,II}^{(i+1)}$, defined in (g) and (h), that represent the defect at next length scale.

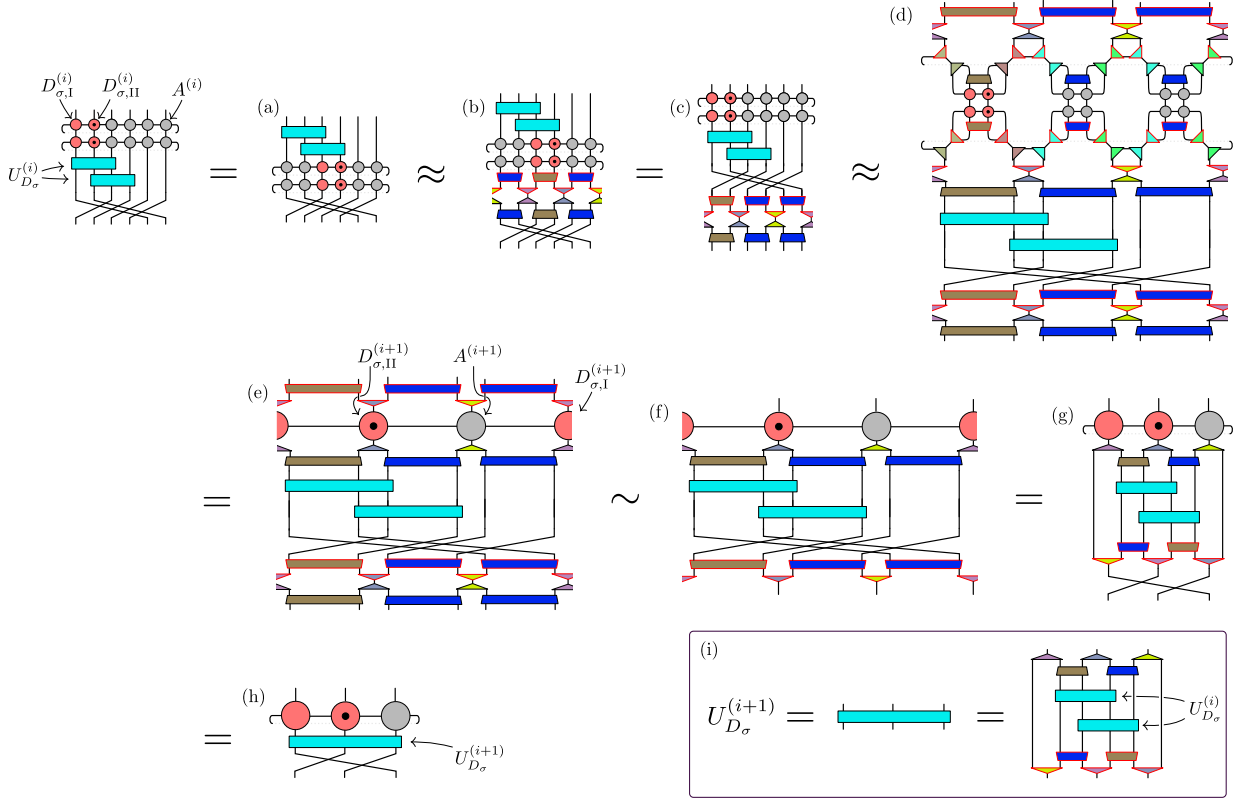


Figure B.7: One step of coarse-graining for the operator $T_{D_\sigma}^{(i)} \cdot M_{D_\sigma}^{(i)}$. The generalized translation by two sites gets coarse-grained into a generalized translation by one site at the next scale. Periodic boundaries in the horizontal direction are assumed even when they are not explicitly shown. All the tensors used in these diagrams are familiar from Fig. B.6 and Fig. B.1 and we have skipped a few intermediate steps that are just a repetition of what is done in those figures. (a) Because $U_{D_\sigma}^{(i)}$ moves the D_σ defect we can move the unitaries to the other side of the transfer matrix as shown. (b) Unitaries and isometries have been inserted into the network. This is like the insertion in the first step of the TNR procedure in Fig. B.6, but without the v isometries and only on one side of the transfer matrix. Thus the truncation error introduced by this insertion is at most the truncation error introduced in the TNR step, because less projections are performed. (c) Using again the property that $U_{D_\sigma}^{(i)}$ moves the defect allows us to move the unitaries back to where they were originally. (d) The first step of the TNR procedure in Fig. B.6 has been applied to the tensors of the transfer matrix. (e) Some of the tensors have been contracted to form $A^{(i+1)}$, $D_{\sigma,I}^{(i+1)}$ and $D_{\sigma,II}^{(i+1)}$, as in Fig. B.4(e). Again, \sim denotes that the operators in (e) and (f) have the same spectrum. This can be shown with an argument exactly analogous to the argument in Fig. B.3(e). (g) Pair of unitaries uu^\dagger have canceled and the graph of the network has been reorganized to make it easier to read. (h) The isometries and unitaries have been contracted to form $U_{D_\sigma}^{(i+1)}$, defined in (i). Note that the definition of $U_{D_\sigma}^{(i+1)}$ is nothing but a MERA ascending superoperator acting on two $U_{D_\sigma}^{(i)}$'s.

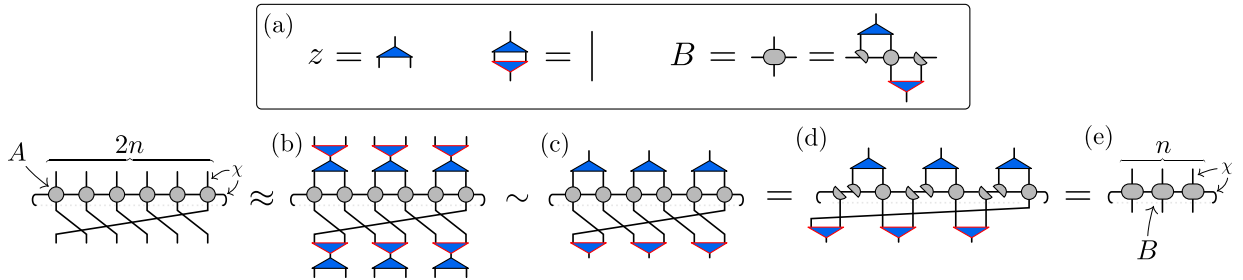


Figure B.8: The additional coarse-graining step to increase the period of the conformal spins. The tensors with red borders are complex conjugates of the ones with the same shape but with black borders. (a) z is a coarse-graining isometry, with all of its legs having dimension χ . B is the tensor obtained at the end of the coarse-graining. (b) zz^\dagger pairs have been inserted in the network as approximate partitions of unity. They can be optimized the same way the TNR isometries are optimized, see Ref. [31]. (c) \sim denotes that the operators in (b) and (c) have the same spectrum. This can be shown with an argument exactly analogous to the argument in Fig. B.3(e). (d) The A tensors have been split into two using a truncated singular value decomposition, as in Levin & Nave’s Tensor Renormalization Group [68]. (e) Some of the tensors have been contracted together into B , defined in (a). The bond dimension of B is the same as the bond dimension of the original tensor A .

B.2 Increasing the period of the conformal spins

The method we present in Sec. 3.3 and Appendix B.1 yields conformal spins modulo n , where n is the number of tensors in the transfer matrix. This appendix describes an additional coarse-graining step that can be taken to increase the period to $2n$. We explain how to do this for a model with no defect, but the method can easily be adapted to accommodate for the presence of a defect.

To get conformal spins with a period of $2n$ we want to diagonalize the operator $T \cdot M$ of $2n$ tensors. Assume, however, that we can numerically only afford to diagonalize a transfer matrix of n tensors, i.e. a matrix with dimensions $\chi^n \times \chi^n$. We can use an additional layer of isometries to coarse-grain $T \cdot M$ down to n tensors (without changing the bond dimension). The scheme we use to do this is shown in Fig. B.8.

This additional coarse-graining does not utilize unitaries the same way TNR does and thus does not remove all short-range correlations. Because of this it causes a truncation error that is relatively large. In practice we find that doing a single such coarse-graining does not qualitatively affect the results: Some accuracy is lost in the scaling dimensions but the conformal spins come out correctly to high accuracy. This lets us match the results

obtained with and without the additional coarse-graining and pick the best of both worlds: Use the additional coarse-graining to get the values for the conformal spins and for the scaling dimensions use values obtained without the additional coarse-graining.

B.3 Results for the 3-state Potts model

In this appendix, we present results for the critical 3-state Potts model on the square lattice. It is defined by the partition function

$$Z = \sum_{\{s\}} \prod_{\langle i,j \rangle} e^{\beta \delta_{s_i, s_j}}. \quad (\text{B.1})$$

s_i are the local degrees of freedom which take three values, say 0, 1 and 2, and δ_{s_i, s_j} is a Kronecker δ that is 1 only if neighboring degrees of freedom are in the same state and 0 otherwise. The sum and the product are over all configurations and all nearest-neighbor pairs, respectively. The model has a critical point at $\beta = \log(\sqrt{3} + 1)$ and the continuum limit at criticality is described by a $c = \frac{4}{5}$ CFT.

If we permute the three different values of s_i , with the same permutation at every site, the Boltzmann weights remain unchanged. Thus the 3-state Potts model has a global internal S_3 symmetry, S_3 being the symmetric group for three elements. For every element of S_3 , of which there are 6, there is topological defect for the CFT. We will concentrate here on three elements $\mathbb{1}$, a and a^2 that form the \mathbb{Z}_3 subgroup of S_3 . This is because \mathbb{Z}_3 is Abelian, and manipulating symmetry preserving tensors is computationally much less intensive for Abelian symmetries than for non-Abelian symmetries.

We call the defects related to these three group elements $D_{\mathbb{1}}$, D_a and D_{a^2} and the corresponding twisted partition functions $Z_{D_{\mathbb{1}}}$, Z_{D_a} and $Z_{D_{a^2}}$. As discussed in Sec. 3.8, the tensor network methods used for the D_ϵ defect of the Ising model can be generalized to these symmetry defects of the Potts model. Hence we can extract the conformal spins and scaling dimensions of the scaling operators present in the partition functions $Z_{D_{\mathbb{1}}}$, Z_{D_a} and $Z_{D_{a^2}}$.

The numerical results we obtain are shown in Fig. B.9. Because of the \mathbb{Z}_3 symmetry each scaling operator comes with a \mathbb{Z}_3 charge, and we have organized the results by this charge. The agreement with the exact values is again excellent. To obtain these results we coarse-grained and diagonalized a transfer matrix of $2^6 \times 4 \times 2^6$ $A^{(0)}$ tensors ($\approx 32\,000$ of the original degrees of freedom). The bond dimensions used in coarse-graining were $\chi = 30$ and $\chi' = 15$. For the conformal spins, a slightly larger system was used, as explained in Appendix B.2.

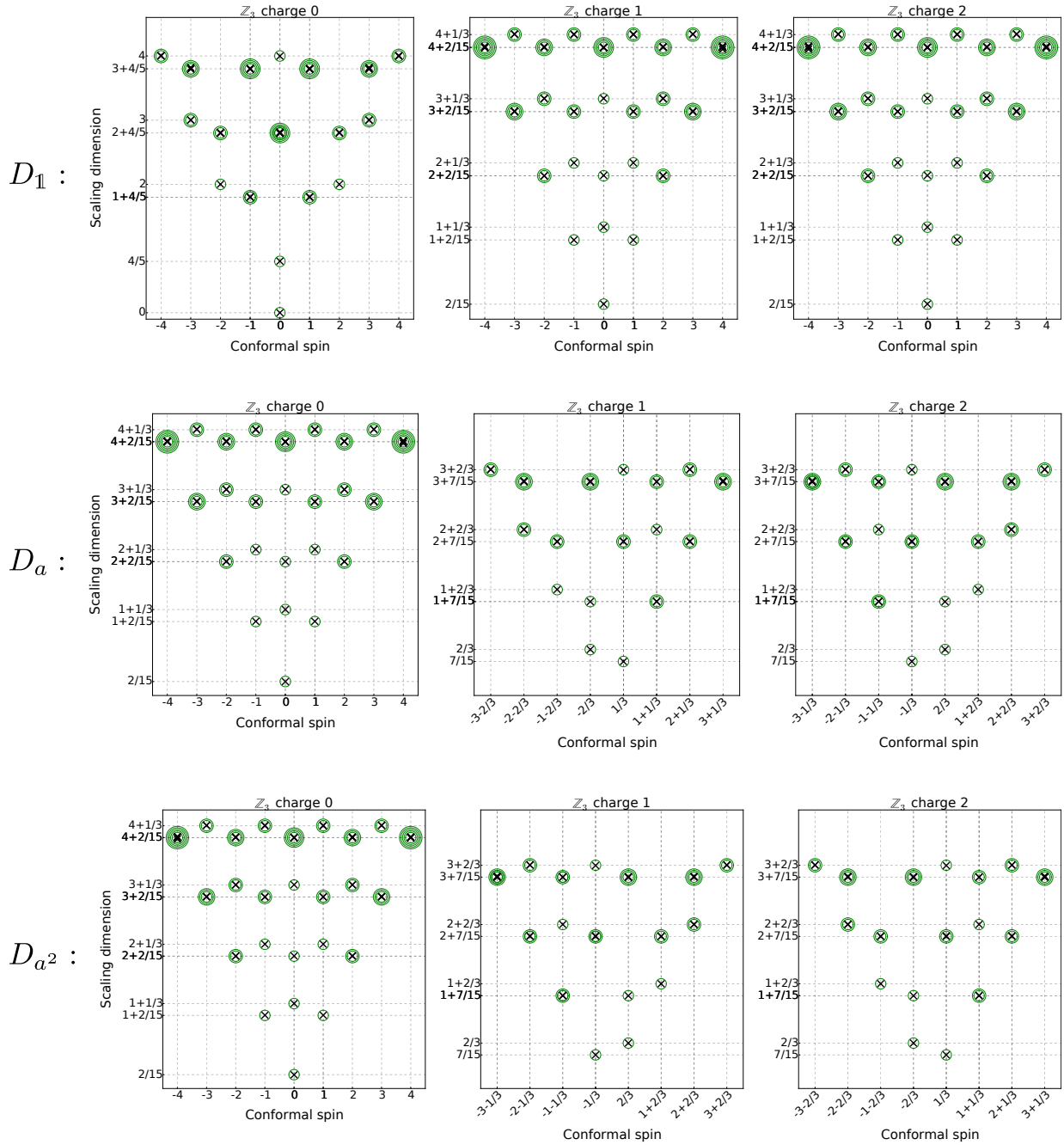


Figure B.9: The scaling dimensions and conformal spins of the first scaling operators of the square lattice 3-state Potts model with various defects as obtained with TNR. Every row of three plots includes the results for one of the defects we study: at the top the trivial defect D_1 , in the middle the defect D_a and at the bottom the defect D_{a^2} . Crosses mark the numerical values, circles mark the exact values. Several concentric circles denote the degeneracy N_α of that $(\Delta_\alpha, s_\alpha)$ pair. Although it is not clear from the figure, these degeneracies also come out correctly.

B.4 Number of degrees of freedom in Z_{D_σ}

In this appendix, we explain the correct way to count the degrees of freedom in a system with a D_σ defect. This affects how we normalize the transfer matrix to extract conformal data.

We build the transfer matrix $M_{D_\sigma}^{(s)}$ from the tensors $A^{(s)}$, $D_{\sigma, \text{I}}^{(s)}$, and $D_{\sigma, \text{II}}^{(s)}$ and diagonalize it. We use the dependence of $M_{D_\sigma}^{(s)}$ on the system size to determine the free energy term lnf in the spectrum, as explained in Sec. 3.2, and then normalize this term away. ln is the number of spins included in the transfer matrix. For the usual Ising model network $Z_{n,m}(A)$ every bond corresponds to a spin and there are twice as many spins as there are $A^{(0)}$ tensors. However, the bonds on the right side of the D_σ tensor are special: they correspond to only half a spin. This needs to be taken into account when determining f .

That there is half a spin missing can be seen from the way the fusion rules manifest in the lattice models, or straight from the explicit form of the D_σ tensor. Perhaps the clearest way is to use the Jordan-Wigner transformation to map the quantum spin chain of n spins into a chain of $2n$ Majorana fermions. There, the D_σ defect is realized as one Majorana mode missing from the chain.

C

Appendices for Chapter 4

C.1 Generic form of purifications

In this appendix, we prove the following theorem, that characterizes all purifications of a given density matrix.

Theorem 1. *Let ρ be a density matrix on the χ -dimensional state space \mathcal{H} . Let X be a $\chi \times \chi_X$ matrix such that $\rho = XX^\dagger$. Let $|\varphi\rangle \in \mathcal{H} \otimes \mathcal{H}_{anc.}$ be a candidate for being a purification of ρ . $\mathcal{H}_{anc.}$ is an ancilla space of dimension $\chi_\varphi \geq \chi_X$. Let φ be a $\chi \times \chi_\varphi$ matrix dual to the state $|\varphi\rangle$, in the sense that $\langle i|\varphi|j\rangle = [\langle i| \otimes \langle j|]|\varphi\rangle$ for all $|i\rangle \in \mathcal{H}$ and $|j\rangle \in \mathcal{H}_{anc.}$. Then $|\varphi\rangle$ is a purification of ρ if and only if there exists a matrix W of dimensions $\chi_X \times \chi_\varphi$, such that*

- W is isometric in the sense that $WW^\dagger = \mathbb{1}$,
- $\varphi = XW$.

Proof. First, assume $\varphi = XW$, with W being isometric. Then

$$\text{Tr}_{anc.} |\varphi\rangle\langle\varphi| = \varphi\varphi^\dagger = XWW^\dagger X^\dagger = XX^\dagger = \rho, \quad (\text{C.1})$$

and thus $|\varphi\rangle$ is a purification of ρ .

To prove the inverse statement, assume $|\varphi\rangle$ is a purification of ρ . Let $X = U_X S_X V_X^\dagger$ and $\varphi = U_\varphi S_\varphi V_\varphi^\dagger$ be the singular value decompositions of X and φ . Then

$$\rho = XX^\dagger = U_X S_X S_X^\dagger U_X^\dagger \quad (\text{C.2})$$

and

$$\rho = \text{Tr}_{\text{anc.}} |\varphi\rangle\langle\varphi| = \varphi\varphi^\dagger = U_\varphi S_\varphi S_\varphi^\dagger U_\varphi^\dagger. \quad (\text{C.3})$$

$S_X S_X^\dagger$ and $S_\varphi S_\varphi^\dagger$ are square and diagonal, and U_X and U_φ are unitary, and thus the above are both eigenvalue decompositions of ρ . Furthermore, we can choose S_X and S_φ to have the singular values ordered by magnitude, which then makes $S_X S_X^\dagger$ and $S_\varphi S_\varphi^\dagger$ be ordered by magnitude. Since the eigenvalue decomposition of a Hermitian matrix is unique up to unitaries that commute with the diagonal matrix of eigenvalues, we then know that $S_\varphi S_\varphi^\dagger = S_X S_X^\dagger = S^2$ and $U_\varphi = U_X u$, where u commutes with S^2 .

S_φ and S_X are in general non-square, and their right-most dimensions, χ_φ and χ_X , may be different. However, since they both square to S^2 in the above sense, we know that the values on their diagonals must be the same. Thus the only difference between them is that S_φ may be padded with columns of zeros compared to S_X (keep in mind that $\chi_\varphi \geq \chi_X$). We can in fact write $S_\varphi = S_X E$, with E (for embedding) being the $\chi_X \times \chi_\varphi$ matrix

$$E = \begin{bmatrix} \mathbb{1}_{\chi_X \times \chi_X} & \bar{0} \end{bmatrix}, \quad (\text{C.4})$$

where $\bar{0}$ is zero matrix of dimensions $\chi_X \times (\chi_\varphi - \chi_X)$.

For u , the fact that it commutes with S^2 implies that there exist unitary matrices u_X and u_φ that fulfill $u S_X = S_X u_X$ and $u S_\varphi = S_\varphi u_\varphi$. They can be constructed by either dropping rows and columns of u corresponding to the null space, or taking the direct sum $u \oplus \mathbb{1}$ with an identity matrix of a suitable dimension, depending on whether χ is larger or smaller than χ_X and χ_φ .

With the above technicalities out of the way, let us finish the proof. Choose $W = V_X u_X E V_\varphi^\dagger$. Then

$$XW = U_X S_X V_X^\dagger V_X u_X E V_\varphi^\dagger = U_X S_X u_X E V_\varphi^\dagger = U_X u S_X E V_\varphi^\dagger = U_\varphi S_\varphi V_\varphi^\dagger = \varphi. \quad (\text{C.5})$$

Thus we have found a W such that $\varphi = XW$. To conclude the proof, we only need to observe that

$$WW^\dagger = V_X u_X E V_\varphi^\dagger V_\varphi E^\dagger u_X^\dagger V_X^\dagger = V_X u_X E E^\dagger u_X^\dagger V_X^\dagger = V_X u_X u^\dagger V_X^\dagger = V_X V_X^\dagger = \mathbb{1}, \quad (\text{C.6})$$

to see that W is isometric. \square

In the above, we have shown that if $\rho = XX^\dagger$, then any purification of ρ can be written as XW for some isometry W . The only caveat here is the assumption that the ancilla space

of the purification has a dimension $\chi_\varphi \geq \chi_X$. This, however, can be easily circumvented by embedding any purification that uses a smaller ancilla, into a larger space. Thus we conclude that Theorem 1 is a full characterization of all purifications of a density matrix of the form $\rho = XX^\dagger$.

C.2 Solving the maximization

In this appendix, we show that for any matrix M ,

$$\max_{W_1, W_2} \left| \text{Tr}[W_1 M W_2^\dagger] \right| = \max_W |\text{Tr}[W M]| = \|M\|_{\text{tr}}, \quad (\text{C.7})$$

where W_1 and W_2 are isometric matrices in the sense that $W_1^\dagger W_1 = \mathbb{1}$ and $W_2^\dagger W_2 = \mathbb{1}$. W is also constrained to be isometric, and if we assume, with no loss of generality, that M is of dimensions $\chi_1 \times \chi_2$ with $\chi_1 \leq \chi_2$, then the isometricity condition on W is $W^\dagger W = \mathbb{1}$.

To get started, singular value decompose M as $M = U S V^\dagger$. Since U and V are unitary, we can always redefine $W_1 U \mapsto W_1$ and $W_2 V \mapsto W_2$ without affecting the isometricity of W_1 or W_2 . Thus,

$$\max_{W_1, W_2} \left| \text{Tr}[W_1 M W_2^\dagger] \right| = \max_{W_1, W_2} \left| \text{Tr}[W_1 S W_2^\dagger] \right|. \quad (\text{C.8})$$

We can further rewrite

$$\left| \text{Tr}[W_1 S W_2^\dagger] \right| = \left| \text{Tr}[S W_2^\dagger W_1] \right| = \left| \sum_{i=1}^{\text{rank } M} \langle W_{2,i} | W_{1,i} \rangle S_i \right|, \quad (\text{C.9})$$

where S_i are the singular values of M , and $|W_{1,i}\rangle$ and $|W_{2,i}\rangle$ are the i^{th} columns of W_1 and W_2 . The isometricity of the W 's translates into

$$\langle W_{1,i} | W_{1,j} \rangle = \delta_{ij} \quad \& \quad \langle W_{2,i} | W_{2,j} \rangle = \delta_{ij}. \quad (\text{C.10})$$

Given that $|W_{1,i}\rangle$ and $|W_{2,i}\rangle$ are normalized, and that S_i are real and non-negative, it is clear that the best one can possibly hope to do in maximizing

$$\left| \sum_{i=1}^{\text{rank } M} \langle W_{2,i} | W_{1,i} \rangle S_i \right|, \quad (\text{C.11})$$

is to have $\langle W_{2,i} | W_{1,i} \rangle = 1$ for all i . This can be achieved by choosing $|W_{1,i}\rangle = |W_{2,i}\rangle$ for the first $\text{rank } M$ columns. The remaining columns can be anything orthogonal to the first

rank M ones, as they do not contribute, and similarly, what exactly $|W_{1,i}\rangle$ are chosen to be makes no difference. With this choice we see that

$$\max_{W_1, W_2} \left| \text{Tr}[W_1 M W_2^\dagger] \right| = \sum_{i=1}^{\text{rank } M} S_i = \|M\|_{\text{tr}}, \quad (\text{C.12})$$

which proves the second part of Eq. (C.7).

In what may seem like bizarrely over-complicated way of reexpressing the above result, we finally point out that with a logic very similar to the one above, one can easily show that

$$\max_W |\text{Tr}[W M]| = \max_W |\text{Tr}[W S]| = \max_W \left| \sum_{i=1}^{\text{rank } M} W_{ii} S_i \right| = \sum_{i=1}^{\text{rank } M} S_i = \|M\|_{\text{tr}}. \quad (\text{C.13})$$

The usefulness of this first part of Eq. (C.7) can be found in the main text, where it is used to justify the definition of the quantity we call *disjoint fidelity*.