QUANTUM MEMORY: DESIGN AND OPTIMIZATION

Xiaotong Ni

Vollständiger Abdruck der von der Fakultät für Physik der Technischen Universität Mnchen zur Erlangung des akademischen Grades eines Doktors der Naturwissenschaften genehmigten Dissertation.

Vorsitzender:	Prof. Dr. A. Holleitner
Prüfer der Dissertation:	1. Prof. Dr. I. Cirac
	2. Prof. Dr. A. Ibarra

Die Dissertation wurde am 06.12.2016 bei der Technischen Universität München eingereicht und durch die Fakultät für Physik am 28.02.2017 angenommen.

Abstract

This thesis contributes to the design and optimization of quantum memories, which are crucial components of many quantum information processing tasks.

In the first part of the thesis, we introduce a generalization of the Pauli stabilizer formalism (PSF). The PSF is the main tool for constructing quantum error correction codes as well as many other applications. We show that our generalized formalism includes genuinely different codes compared to the PSF while maintaining the tractability, i.e. many properties of the codes can be computed efficiently.

In the second part, we investigate the feasibility of preparing ground states of topological Hamiltonians through adiabatic evolutions, which can be considered as a way to initialize topological quantum memories. Compared to other preparation methods, it requires less control on the microscopic level, and it is more robust against possible perturbations in the Hamiltonians. Through a numerical study of systems with a small number of sites, we show that adiabatic evolutions generally lead to ground states of several different topological models. We also observe that the prepared states display a certain stability with respect to the change of initial Hamiltonians, which is then partially explained with the use of perturbation theory.

In the last part, we show that it is possible to use machine learning to optimize dynamical decoupling sequences for quantum memories. More concretely, we use recurrent neural networks to model the structure of "good" sequences and then generate possibly better sequences using the learned structure. By iterating this routine, we are eventually able to find sequences with better or similar performance compared to those constructed by humans.

Zusammenfassung

Diese Doktorarbeit leistet einen Beitrag zum Design und zur Optimierung von Quantenspeichern. Diese sind zentrale Bestandteile vieler Anwendungen in der Quanteninformationstheorie.

Im ersten Teil der Arbeit präsentieren wir eine Verallgemeinerung des Pauli-Stabilisator-Formalismus (PSF). Mit Hilfe des PSF können unter anderem Quantenfehlerkorrekturcodes konstruiert werden. Wir zeigen dass unser verallgemeinerter Formalismus grundsätzlich andere Codes im Vergleich zum PSF hervorbringt und gleichzeitig die Handhabbarkeit aufrecht erhalten wird, d.h. dass viele Codes effizient berechnet werden können.

Im zweiten Teil untersuchen wir inwiefern es möglich ist Grundzustnde von topologischen Hamiltonians durch adiabatische Entwicklung zu generieren. Dies kann als Methode zur Initialisierung von topologischen Quantenspeichern gesehen werden. Im Vergleich zu anderen Herangehensweisen bentigt dies weniger Kontrolle auf dem mikroskopischen Level und ist stabiler im Hinblick auf Pertubationen im Hamiltonian. Durch eine numerische Analyse von kleinen Systemen zeigen wir, dass die adiabatische Entwicklung generell zu Grundzuständen unterschiedlicher topologischer Modelle führt. Wir beobachten auch, dass die geniererten Zustände eine gewisse Stabilität im Hinblick auf Änderungen in den Ausgangshamiltonians aufzeigen. Dies beschreiben wir auch mit Hilfe der Perturbationstheorie.

Im letzten Teil zeigen wir, dass es möglich ist maschinelles Lernen zu nutzen um dynamische Sequenzen zur Entkoppelung von Quantenspeichern zu optimieren. Genauer gesagt werden periodische neuronale Netze verwendet um die Struktur von guten Sequenzen zu modellieren und um dann mglicherweise bessere Sequenzen durch die erlernte Struktur zu generieren. Durch Wiederholung dieser Routine, findet man schlielich Sequenzen mit besserer oder ähnlicher Leistung wie solche, die durch Menschen entwickelt wurden.

Publication

The thesis is based on the following publications and preprints

1. Xiaotong Ni, Oliver Buerschaper, and Maarten Van den Nest. *A non-commuting stabilizer formalism.* Journal of Mathematical Physics, 56(5):052201, 2015.

2. Xiaotong Ni, Fernando Pastawski, Beni Yoshida, and Robert Koenig. *Preparing topologically ordered states by hamiltonian interpolation.* New J. Phys. 18 093027, 2016.

3. Moritz August and Xiaotong Ni. *Using recurrent neural networks to optimize dynamical decoupling for quantum memory.* arXiv preprint arXiv:1604.00279, 2016.

Contents

Chapter 1

Introduction

Computation and communication devices exist almost as long as human civilizations: they range from abaci and beacons in the early day to electronic computers and optical fibers in the modern society. While the media and underlying mechanics have changed a lot, they all process and transmit classical information. This profound observation eventually led to the formulation of classical computation and communication theory, in particular by Turing [Tur36] and Shannon [Sha48]. One interesting consequence is that, for decades, it was believed that whether a computational problem can be solved efficiently or not is independent of the underlying computational models and physical systems (see chapter 4 of [Aar13]).

During the last few decades, it was realized that quantum mechanics could be helpful in many information processing tasks. Notable examples include Shor's factoring algorithm [Sho99] and quantum key distribution [BB84] in the early phase, as well as quantum metrology [GLM06] and quantum machine learning [LMR13] later. For the first time, researchers broke the above paradigm by realizing that information can be quantum in a useful way. Besides being a new theory, quantum information also started to change the way how scientists understand problems in various fields of physics such as condensed matter and black holes (e.g. [KP06, ADH15]). While the use of quantum phenomena promises advantages in several areas, an often necessary condition is the ability to keep the quantum states coherent. For instance, it has been shown in [DiV95, EdMFD11, AGJO$^+$15] that the existence of noise can nullify the quantum advantages. The need of fault-tolerance is even more apparent for quantum computing, as a quantum algorithm may require accurate execution of hundreds of unitary gates.

A quantum memory is designed to achieve the simplest task of fault-tolerance: output the same state as the initial input state after a delay. While it is easy to store classical information for centuries, we are still unable to store an unknown qubit (i.e. a 2-dimensional quantum state) for hours. This is the case partially because it is hard to find a quantum system which can be controlled but at the same time sufficiently isolated from the environment in order to suppress decoherence. A storage cycle of a quantum memory usually consists of a subset of three steps: encoding, maintenance and decoding.

- **Encoding:** It is natural that the input states are transformed so that they fit better to the structure of the quantum memory, e.g. when the memory has a decoherence-free subspace [LW03]. The encoding step is necessary when using quantum error correction codes (QECC) [NC00], as the main idea is to encode input states into a larger Hilbert space so that a small number of errors cannot map one encoded state to another (which is the same for classical error correction codes). However, encoding is not required for techniques like dynamical decoupling [VKL99].

- **Maintenance:** After a possible encoding step, it is often necessary that one maintains an active control of the system over the storage time. As the majority time of a storage cycle is the in maintenance phase, it is a deciding factor for the success of a quantum memory. In the case of QECC, the goal of the maintenance is to prevent errors from accumulating, since the QECC is only able to correct a limited number of errors. This can be achieved once the error rate of quantum gates (i.e. unitary operations) is below a certain threshold, which is

still beyond today's experimental capacities. As a result, there is significant motivation to search for quantum memories that do not require an active maintenance phase.

- **Decoding:** Reversing the encoding step so that the output states can be the same as the inputs.

Currently, using QECC is the most popular way of achieving fault-tolerance. It is done by encoding a Hilbert space into a subspace of a higher dimensional one. The subspace is chosen in a way such that if only a few defects happen among the components of the system (e.g. individual spins of a spin system), then the defects can be detected and corrected. To achieve this task, one needs a way to specify these subspaces which consist of highly entangled quantum states, and to study their properties. This is not easy since a general N-particle quantum state requires exponentially many coefficients for its description.

The Pauli stabilizer formalism (PSF) [Got97] has become the main tool to describe QECC for 2-level systems, since it efficiently deals with the difficulties mentioned above. In more detail, a subspace is specified as the common eigenspace of a set of commuting operators, where each operator is a tensor product of Pauli matrices. The properties of such subspaces are well-understood, in the sense that there are efficient algorithms to compute their dimension, to construct logical operators supported on them and the quantum circuits for encoding/decoding, to compute entanglement properties, etc. In addition, the PSF is also used to build toy models for topologically ordered systems [Kit03] (and to some degree the AdS/CFT correspondence [PYHP15]), and is used as the main tool for measurement-based quantum computation [RBB03]. The wide use of the PSF is mainly due to the fact that it is one of the few versatile Hamiltonian classes (if not the only) that one can work with pen and paper, rather than relying on numerical tools. From a physical point of view, it might be argued that the simplicity of the PSF trivially comes from the commutativity of the individual terms in the Hamiltonians. However, it is known that computing the ground space of a commuting Hamiltonian is hard [BV05]. Thus, the details in the algebraic structure of the PSF are important, and it is not straightforward to enlarge the PSF class without ruining its features.

In the **first part** of this thesis, we will introduce a generalization of the PSF, while keeping most of the nice properties mentioned above. We still describe subspaces by tensor products of matrices, but the matrices are chosen from a larger group that contains the Pauli group. We show that, under moderate conditions, the tasks mentioned above (constructing a basis for the subspace, etc) can again be solved efficiently. An interesting aspect of our generalization is the connection to topological order. It is known that the PSF can only describe the toric code in 2D [BDCP12], while we show that our formalism includes more models. Thus, it is interesting to analyze how the slight generalization leads to this change and to use it to construct new models.

Another approach to protect quantum information and doing quantum computation is by utilizing physical systems with topological order [Kit03]. The main idea is that for each low energy level in these systems, there is a degenerate subspace, where the splitting between them is exponentially small with respect to the system size. Thus, if one suppresses the system-environment interaction and keeps the quantum states in one of the degenerate subspaces, the quantum information can be stored safely, despite possible perturbations in the Hamiltonians. Compared to using QECC and doing active error correction, this approach aims at very different types of physical systems. For example, it is reasonable to assume that we will be able to build certain 2D materials that have anyonic excitations and control them[1], while we will not be able to control these systems on an atomic/molecular level for active QECC. It is then natural to use the topological approach. While it is very demanding to prevent thermal noise (couplings to the environment) from corrupting quantum information stored in the topological memory only by increasing the system size, it is possible that with certain well engineered materials the storage time can be long enough for all practical purposes. Moreover, it is still an open question whether there is a 3D topological Hamiltonian which, in the presence of thermal noise, allows a storage time longer than the logarithm of the system size. Thus, in principle there could exist some materials which protect

[1] It is widely believed the fractional quantum Hall effect experiments have non-abelian anyonic excitations.

quantum information on a hardware level and save us from the stringent requirement of active QECC. However, a necessary step for topological quantum memories and computers is initializing the system in the ground space (or a subspace corresponding to a certain chosen low energy level). Several approaches have been studied [DKLP02, DKP14, BBK$^+$13, LMGH15]. Nevertheless, as we mentioned earlier, the topological approach is mostly suitable for physical systems where we do not have access to the individual degrees of freedoms or do not have the ability to control the Hamiltonian precisely. Thus, it is desirable to have a preparation procedure that satisfies these requirements.

The **second part** of this thesis is about adiabatic preparation of topological states. More concretely, we start with a product state and the corresponding Hamiltonian, and linearly interpolate towards the final topological Hamiltonian. This approach was first studied in [HL08] for preparing toric code ground states. Besides satisfying the requirements mentioned in the previous paragraph (no need to control the system at a microscopic level), the adiabatic preparation also has the advantage of being robust to the perturbation of the Hamiltonian during the interpolation. The only requirement is that the temperature stays low compared to the energy gap, and the total evolution time is short enough so that the system does not thermalize. In particular, it likely allows one to prepare states inside the ground space of the final perturbed Hamiltonian, which is crucial for enabling the topological protection, because the ground space is not robust against perturbations. On the other hand, although this process seems to be straightforward, several questions still need to be understood. The first one is that since there is a phase transition between the initial and final Hamiltonian in the thermodynamic limit, the energy gap above the ground state will close at the critical point. Thus, the adiabatic theorem cannot be directly applied, and it is unclear how in general this process could succeed. Another question is what kind of final states will be prepared by this process. While this is not a question particularly interesting in physics, it matters a lot for quantum information processing. For example, it is known that the ability to prepare certain "magic states" would drastically lower the requirement for fault-tolerant quantum computation [BK05]. In our work, we attempt to address these two questions for systems with small size, with both numerical simulation and theoretical analysis. We find that the final ground space can be reached for most initial Hamiltonians, and interestingly the prepared final states concentrate around a discrete set of states. A perturbation argument is given to explain the mechanism of this phenomenon.

While it seems that there is agreement on several high level architectures of achieving fault-tolerance, we still need to apply them to concrete experimental setups. This task can be demanding, since there are many parameters to be fine tuned. It is natural to automatize this task for which many different algorithms have already been invented. However, often the optimization has the form of a local search, e.g. gradient descent [MSG$^+$11] or the Nelder-Mead simplex method [KBC$^+$14]. While local searches are very effective for suitable tasks, it is desirable to understand the structure of good control parameters and use the learned structure to suggest new parameters. As we want to automatically solve harder and harder problems, the algorithms need to exploit the structures that underly these problems. For example, it is extremely hard to believe that for the factoring problem Shor's algorithm can be found through a local search. Rather, the search procedure has to utilize existing knowledge in an organic way. While achieving this kind of "human" search procedure is still infeasible to date, it does not stop us to find problems with suitable complexity and apply algorithms which exploit their structures. This direction is particularly interesting, since the recent development in machine learning makes it easier to find structure in a dataset with readily available software.

In the **third part** of this thesis, we will apply the approach of machine learning to find good dynamical decoupling (DD) sequences. DD is a relatively mature technique to suppress errors in a quantum system. Compared to QECC, it has the advantage of being an open-loop control protocol and does not bring overheads (i.e. every physical qubit is a logical one). As a result, DD has been demonstrated in several experiments where it increased the coherence time by orders of magnitude [BUV$^+$09, dLWR$^+$10, SAS11]. While existing DD sequences already show good performance, it is still important to optimize them, especially if we want to perform the optimization directly for some specific experimental setups. Another reason we choose to apply

machine learning to the DD problem is because the structure of DD sequences naturally fits to some well studied models. In particular, we will use the recurrent neural network for our task, which is widely used in modeling natural language, handwriting, etc. It is conceivable that the connection we introduce will bring progress from the field of machine learning to optimal quantum control.

To sum up, in this thesis, we studied multiple aspects of quantum memory, including a formalism that describes quantum error correction codes, a procedure of initializing topological quantum memory, and machine learning methods for optimizing the performance.

Chapter 2

A non-commuting stabilizer formalism

This chapter is based on [NBVdN15].

In this chapter, we propose a generalization of the Pauli stabilizer formalism (PSF) while keeping its tractability. More concretely, we consider a set of operators which are tensor products of single-qubit (2-dimensional) matrices, and use them to describe a subspace that is the common eigenspace of them. The single-qubit matrices belong to an extension of the Pauli group. We show that under certain mild conditions, we can efficiently construct a basis for such subspaces and can compute entanglement properties of the states in that basis. Moreover, we demonstrate that the generalized formalism is able to describe a larger number of topological models then the PSF, thus establishing a meaningful distinction between the generalized and the original formalism.

2.1 Introduction

In this chapter, we will focus on the description of quantum error correcting codes. As mentioned earlier, often the preliminary of doing quantum error correction is describing the code subspaces. This is an non-trivial task because a general quantum state of N particles requires exponentially many parameters to describe. One valid strategy to deal with this problem is to study subclasses of states that may be described with considerably less parameters, while maintaining a sufficiently rich structure to allow for nontrivial phenomena. The Pauli stabilizer formalism (PSF) is one such class and it is a widely used tool throughout the development of quantum information [Got97], in particular for constructing quantum error correcting codes. In the PSF, a quantum state or subspace is described in terms of a group of operators that leave the state invariant. Such groups consist of Pauli operators and are called Pauli stabilizer groups. An n-qubit Pauli operator is a tensor product $g = g^{(1)} \otimes \cdots \otimes g^{(n)}$ where each $g^{(i)}$ belongs to the single-qubit Pauli group, i.e. the group generated by the Pauli matrices X and Z and the diagonal matrix iI. Since every stabilizer group is fully determined by a small set of generators, the PSF offers an efficient means to describe a subclass of quantum states and gain insight into their properties. States of interest include the cluster states [RBB03], GHZ states [GHSZ90] and the toric code [Kit03]; these are entangled states which appear in the contexts of e.g. measurement based quantum computation [RBB03] and topological phases. As an illustration, the 3-qubit GHZ state $|\psi\rangle = |000\rangle + |111\rangle$ is the unique common eigenstate (up to some constants) of

$$A = \{X_1 \otimes X_2 \otimes X_3, Z_1 \otimes Z_2, Z_2 \otimes Z_3\}$$

with eigenvalue 1, where $X_i(Z_i)$ are the corresponding Pauli matrices on the qubit i. Thus, we say $|\psi\rangle$ is a Pauli-stabilizer state which is stabilized by the set A. If we remove the operator $X_1 \otimes X_2 \otimes X_3$ from A, the result

$$A' = \{Z_1 \otimes Z_2, Z_2 \otimes Z_3\}$$

will stabilize the subspace spanned by $\{|000\rangle, |111\rangle\}$. It then serves as a very basic error correcting code that can correct a single X error on one of the three qubits. For example, if an X error happened on the first qubit, it will cause the eigenvalue of $Z_1 \otimes Z_2$ to flip while leave the eigenvalue of $Z_2 \otimes Z_3$ to stay unchanged. From these information, we can deduce the position of the X error and apply the correction accordingly. With the quantum error-correction condition [KL97] and more sophisticated codes, we know that the errors from typical error models can be corrected when the error rate is low enough.

Considering the importance of the PSF in quantum error correction as well as in other branches of quantum information theory, it is natural to ask whether we can extend this framework and describe a larger class of states, while keeping as much as possible both a transparent mathematical description and computational efficiency. In this chapter, we provide a generalization of the PSF. In our setting, we allow for stabilizer operators which are tensor product operators $g^{(1)} \otimes \cdots \otimes g^{(n)}$ where each $g^{(i)}$ belongs to the group generated by the matrices X, $S \coloneqq \sqrt{Z}$ and $\sqrt{\mathrm{i}}I$. Our setting contains the PSF because $S^2 = Z$. Similar to the PSF, we consider states that are invariant under the action of such generalized stabilizer operators. The resulting stabilizer formalism is called here the *XS-stabilizer formalism*. It is a subclass of the monomial stabilizer formalism introduced recently in [VdN11]. Interestingly, the XS-stabilizer formalism allows for non-Abelian stabilizer groups, whereas it is well known that stabilizer groups in the PSF must be Abelian.

Even though the definition of the XS-stabilizer formalism is close to that of the original PSF, these frameworks differ in several ways. In particular, the XS-stabilizer formalism is considerably richer than the PSF, and we will encounter several manifestations of this. At the same time, the XS-stabilizer formalism keeps many favorable features of the PSF. For example, XS-stabilizer groups have a simple structure and are easy to manipulate, and there exists a close relation between the stabilizer generators of an XS-stabilizer state/code and the associated Hamiltonian. Moreover, we will show that (under a mild restriction of the XS-stabilizers) many quantities of interest can be computed efficiently, such as expectation values of local observables, code degeneracy and logical operators. However, in most cases we found that efficient algorithms could not be obtained by straightforwardly extending methods from the PSF, and new techniques needed to be developed.

The purpose of this chapter is to introduce the XS-stabilizer formalism, to provide examples of XS-stabilizer states and codes that are not covered by the PSF and to initiate a systematic development of the XS-stabilizer framework. In particular, we discuss several properties related to the structure of XS-stabilizer states and codes, their entanglement, their efficient generation by means of quantum circuits and their efficient simulation with classical algorithms. A detailed statement of our results is given in section 2.3. Here we briefly highlight two aspects.

First, we consider the potential of the XS-stabilizer formalism to describe topological phases (a more detailed introduction to several topological models can be found in the next chapter). This is motivated by recent works on classifying quantum phases within the PSF [Yos11, BDCP12] and finding a self-correcting quantum memory. In particular, Haah constructed a novel Pauli stabilizer code for a 3D lattice in [Haa11] which does not have string like operators. This is a necessary condition for self-correcting, and also an evidence that the PSF is an useful tool for constructing new physical models. In the present chapter we show that the XS-stabilizer formalism can describe 2D topological phases beyond the PSF and, surprisingly, some of these harbour non-Abelian anyons. Specific examples of models covered by the XS-stabilizer formalism are the doubled semion model [LW05] and, more generally, the twisted quantum double models for the groups $\mathbb{Z}_2^k$ [HWW13, Bue14].

Second, we study entanglement in the XS-stabilizer formalism. Entanglement has always been the defining feature of quantum information. It says that the correlation between physical particles can be more complicated than classical probability distributions (with respect to multiple choices of measurements). It is then realized that the richer structure would allow certain protocols to only exist in quantum information, such as quantum cryptography. Entanglement is also a necessary condition for exponential speed-up of a quantum algorithm, as a quantum computation with no entanglement can be efficiently simulated on a classical computer. Thus, it is interesting to understand the nature of entanglement for XS-stabilizer states. We note that various entanglement properties of Pauli stabilizer states have been studied extensively in the past decade [HEB04,

FCY+04]. While the bipartite entanglement structure is very well understood, less is known about the multipartite scenario. For example, recently in Ref. [LMRW13] the entropy inequalities for Pauli stabilizer states were studied. Here we will show that, for any bipartition, we can always map any XS-stabilizer state into a Pauli stabilizer state locally, which means their bipartite entanglement is identical. This implies in particular that all reduced density operators of an XS-stabilizer state are projectors and each single qubit is either fully entangled with the rest of the system or fully disentangled from it. In contrast, the XS-stabilizer formalism is genuinely richer than the PSF when viewed through the lens of multipartite entanglement. For example, we will show that there exist XS-stabilizer states that cannot be mapped onto any Pauli stabilizer state under local unitary operations. Thus there seems to be a complex and intriguing relation between the entanglement properties of Pauli and XS-stabilizer states.

We also mention other works that, similar in spirit to the present chapter, aim at extending the PSF. These include: Ref. [HCDB07] which introduced the family of weighted graph states as generalizations of graph and stabilizer states; Ref. [KK09] where the family of locally maximally entanglable (LME) states were considered (which in turn generalize weighted graph states); Ref. [RHBM13] where hypergraph states were considered. The XS-stabilizer formalism differs from the aforementioned state families in that its starting point is the representation of states by their stabilizer operators. We have not yet investigated the potential interrelations between these classes, but it would be interesting to understand this in more detail.

Outline of the chapter. In section 2.2 we introduce the basic notions of XS-stabilizer states and codes, while in section 2.3 we give a summary of the results presented in this chapter. The following sections are dedicated to developing the technical arguments.

2.2 The XS-stabilizer formalism

In this section we introduce the basic notions of XS-stabilizer states and codes and we provide several examples.

2.2.1 Definition

First we briefly recall the standard Pauli stabilizer formalism. Let X, Y and Z be the standard Pauli matrices. The single-qubit Pauli group is $\langle \mathrm{i}I, X, Z\rangle$. For a system consisting of n qubits we use X_j, Y_j and Z_j to represent the Pauli matrices on the j-th qubit. An operator g on n qubits is a Pauli operator if it has the form $g = g^{(1)} \otimes \cdots \otimes g^{(n)}$ where each $g^{(i)}$ belongs to the single-qubit Pauli group. Every n-qubit Pauli operator can be written as

$$g = \mathrm{i}^s X^{a_1} Z^{b_1} \otimes \cdots \otimes X^{a_n} Z^{b_n}$$

where $s \in \{0, \ldots, 3\}$, $a_j \in \{0, 1\}$ and $b_j \in \{0, 1\}$. We say an n-qubit quantum state $|\psi\rangle \neq 0$ is stabilized by a set of Pauli operators $\{g_j\}$ if

$$g_j |\psi\rangle = |\psi\rangle \qquad \text{for all } j.$$

The operators g_j are called *stabilizer operators* of $|\psi\rangle$.

In this chapter, we generalize the Pauli stabilizer formalism by allowing more general stabilizer operators. Instead of the single-qubit Pauli group, we start from the larger group $\mathcal{P}^S \coloneqq \langle \alpha I, X, S\rangle$ where $\alpha = \mathrm{e}^{\mathrm{i}\pi/4}$ and $S = \mathrm{diag}(1, \mathrm{i})$. Note that the latter group, which we call the *Pauli-S group*, contains the single-qubit Pauli group since $S^2 = Z$. We then consider stabilizer operators $g = g^{(1)} \otimes \cdots \otimes g^{(n)}$ where each $g^{(i)}$ is an element of $\mathcal{P}^S$. It is easy to show that every such operator can be written as

$$g = \alpha^s X^{a_1} S^{b_1} \otimes \cdots \otimes X^{a_n} S^{b_n} =: \alpha^s X(\vec{a})\, S(\vec{b}) \tag{2.1}$$

where $s \in \{0, \ldots, 7\}$, $a_j \in \{0, 1\}$ and $b_j \in \{0, \ldots, 3\}$. Here we also defined $X(\vec{a}) \coloneqq X^{a_1} \otimes \cdots \otimes X^{a_n}$ for $\vec{a} = (a_1, \ldots, a_n)$ and similarly $S(\vec{b})$ and $Z(\vec{c})$. These are called *X-type*, *S-type* and *Z-type operators* respectively.

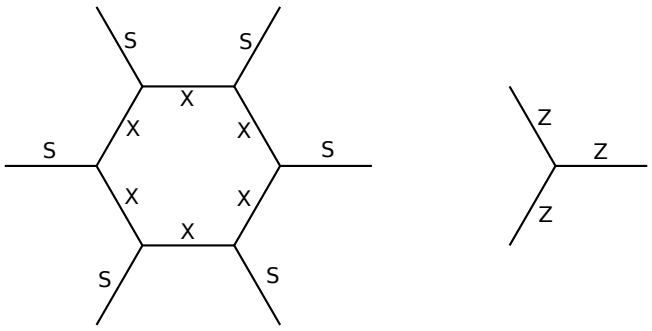


Figure 2.1: In the doubled semion model, the qubits are on the edges of a honeycomb lattice. The ground space of the Hamiltonian can be equivalently described by the two types of XS-stabilizers in the above figure. The left one is associated to each face of the lattice and the right one is associated to each vertex.

For a set $\{g_1, \ldots, g_m\}$ of such operators we consider the group $G = \langle g_1, \ldots, g_m \rangle$, and we say a state $|\psi\rangle \neq 0$ is stabilized by G if we have $g|\psi\rangle = |\psi\rangle$ for every $g \in G$. Whenever such a state exists we call G an *XS-stabilizer group*. The space $\mathcal{L}_G$ of all states stabilized by G is referred to as the *XS-stabilizer code* associated with G. A state which is uniquely stabilized by G is called an *XS-stabilizer state*.

Thus the XS-stabilizer formalism is a generalization of the Pauli stabilizer formalism. Perhaps the most striking difference is that XS-stabilizer states/codes may have a *non*-Abelian XS-stabilizer group G – while Pauli stabilizer groups must always be Abelian. We will see examples of this in the next section.

2.2.2 Examples

Here we give several examples of XS-stabilizer states and codes and highlight how their properties differ from the standard Pauli stabilizer formalism.

A first simple example of an XS-stabilizer state is the 6-qubit state $|\psi\rangle$ stabilized by the (non-commuting) operators

$$\begin{aligned} g_1 &= X \otimes S^3 \otimes S^3 \otimes S \otimes X \otimes X, \\ g_2 &= S^3 \otimes X \otimes S^3 \otimes X \otimes S \otimes X, \\ g_3 &= S^3 \otimes S^3 \otimes X \otimes X \otimes X \otimes S. \end{aligned}$$

Explicitly, $|\psi\rangle$ is given by

$$|\psi\rangle = \sum_{x_j=0}^{1} (-1)^{x_1 x_2 x_3} |x_1, x_2, x_3, x_1 \oplus x_2, x_2 \oplus x_3, x_3 \oplus x_1\rangle. \tag{2.2}$$

It is straightforward to show that $|\psi\rangle$ is the unique (up to a global phase) state stabilized by g_1, g_2 and g_3. Note that in this example 3 stabilizer operators suffice to uniquely determine the 6-qubit state $|\psi\rangle$. This is different from the Pauli stabilizer formalism, where 6 stabilizers would be necessary (being equal to the number of qubits). Notice also that $|\psi\rangle$ contains amplitudes of the form $(-1)^{c(x)}$ where $c(x)$ is a cubic polynomial of the bit string $x = (x_1, x_2, x_3)$. This shows that $|\psi\rangle$ cannot be a Pauli stabilizer state, since the latter cannot have such cubic amplitudes [DDM03]. This example thus shows that the XS-stabilizer formalism covers a strictly larger set of states than the Pauli stabilizer formalism. What is more, we will show (cf. section 2.10.2) that the state $|\psi\rangle$ is not equivalent to any Pauli stabilizer state even if arbitrary local basis changes are allowed. Thus, $|\psi\rangle$ belongs to a different local unitary equivalence class than any Pauli stabilizer state.

A second example is the *doubled semion model* which belongs to the family of string-net models [LW05]. It is defined on a honeycomb lattice with one qubit per edge and has two types of

	Pauli	Regular XS	General XS
Commuting stabilizer operators	yes	no	no
Commuting parent Hamiltonian	yes	yes	yes
Complexity of stabilizer problem	P	P	NP-complete
Non-Abelian anyons in 2D	no	yes	yes

Table 2.1: Summary of the properties

stabilizer operators[1] which are shown in Figure 2.1. Let g_s and g_p be the stabilizer operators corresponding to the vertex s and the face p respectively. Then the ground space of the doubled semion model consists of all states $|\psi\rangle$ satisfying $g_s|\psi\rangle = g_p|\psi\rangle = |\psi\rangle$ for all s and p. The doubled semion model is closely related to the toric code which is a Pauli stabilizer code. The Pauli stabilizer operators of the toric code are obtained from the XS-stabilizer operators of the doubled semion model by replacing all occurrences of S with I. This is no coincidence since both the doubled semion model and the toric code are *twisted quantum double models* for the group $\mathbb{Z}_2$ [HWW13, Bue14]. In spite of this similarity it is known that both models represent different topological phases [LW05]. Thus, the XS-stabilizer formalism allows one to describe states with genuinely different topological properties compared to any state arising in the Pauli stabilizer formalism [Yos11, BDCP12]. In fact, we can use XS-stabilizers to describe other, more complex, twisted quantum double models as well, as we will show in section 2.A. Some of these even support non-Abelian anyons.

The third example is related to magic state distillation. In [BK05] the authors consider a 15 qubit code $\mathrm{CSS}(Z, \mathcal{L}_2; XS, \mathcal{L}_1)$, where $\mathcal{L}_1$ and $\mathcal{L}_2$ are punctured Reed-Muller codes of order one and two, respectively. Roughly speaking, this quantum code is built from two types of generators. One type has the form $Z \otimes \cdots \otimes Z$ acting on some of the qubits, while the other type has the form $XS \otimes \cdots \otimes XS$. Surprisingly, this 15 qubit XS-stabilizer code has the same code subspace as the Pauli stabilizer code $\mathrm{CSS}(Z, \mathcal{L}_2; X, \mathcal{L}_1)$ which is obtained by replacing every S operator with an identity matrix. From this example we can see that having S in the stabilizer operators does not necessarily mean an XS-stabilizer group and a Pauli stabilizer group stabilize different spaces.

2.3 Main results

2.3.1 Commuting parent Hamiltonian

Even though an XS-stabilizer group $G = \langle g_1, \ldots, g_m \rangle$ is non-Abelian in general, we will show that there always exists a Hamiltonian $H' = \sum_j h_j$ with mutually commuting projectors h_j whose ground state space coincides with the space stabilized by G (section 2.5). If the generators of G satisfy some locality condition (e.g. they are k-local on some lattice), then the h_j will satisfy the same locality condition (up to a constant factor). This means that general properties of ground states of commuting Hamiltonians apply to XS-stabilizer states. For example, every state uniquely stabilized by a set of local XS-stabilizers defined on a D-dimensional lattice satisfies the area law [WVHC08], and for local XS-stabilizers on a 2D lattice, we can find string like logical operators [BPT10].

While the ground state spaces of H' and the non-commuting Hamiltonian $H = \sum_{j=1}^{m}(g_j + g_j^\dagger)$ are identical, the latter may have a completely different spectrum. This may turn out important for the purpose of quantum error correction.

2.3.2 Computational complexity of finding stabilized states

In the Pauli stabilizer formalism, it is always computationally easy to determine whether, for a given set of stabilizer operators, there exists a common stabilized state. However, we will prove

[1] The local single-qubit basis used in [LW05] is different from ours.

that the same question is NP-*complete* for XS-stabilizers (see section 2.7). More precisely, we consider the problem XS-Stabilizer defined as follows: given a set of XS-stabilizer operators $\{g_1, \ldots, g_m\}$, the task is to decide whether there exists a state $|\psi\rangle \neq 0$ stabilized by every g_j. The NP-hardness part of the XS-Stabilizer problem is proved via a reduction from the Positive 1-in-3-Sat problem. In order to show that the problem is in NP, we use tools developed for analyzing monomial stabilizers, as introduced in [VdN11].

The NP-hardness of the XS-Stabilizer problem partially stems from the fact that the group $G = \langle g_1, \ldots, g_m \rangle$ may contain *diagonal* operators which have one or more S operators in their tensor product representation (2.1). In order to render the XS-Stabilizer problem tractable, we impose a (mild) restriction on the group G and demand that every diagonal operator in G can be written as a tensor product of I and Z, i.e. no diagonal operator in G may contain an S operator. We call such a group G *regular*. We will show that, for every regular G, the existence of a state stabilized by G can then be checked efficiently (section 2.8).

Finally, we will show that in fact every XS-stabilizer *state* affords a regular stabilizer group (although finding it may be computationally hard), i.e. the condition of regularity does not restrict the set of states that can be described by the XS-stabilizer formalism (Section 2.12). In contrast, the stabilizer group of an XS-stabilizer *code* cannot always be chosen to be regular.

2.3.3 Entanglement

Given an XS-stabilizer state $|\psi\rangle$ with associated XS-stabilizer group G, we show how to compute the entanglement entropy for any bipartition (A, B) (section 2.10). This is achieved by showing that $|\psi\rangle$ can always be transformed into a Pauli stabilizer state $|\phi_{A,B}\rangle$ (which depends on the bipartition in question) by applying a unitary $U_A \otimes U_B$, where U_A and U_B each only act on the qubits in each party. Since an algorithm to compute the entanglement entropy of Pauli stabilizer states is known, this yields an algorithm to compute this quantity for the original XS-stabilizer state $|\psi\rangle$ since the unitary $U_A \otimes U_B$ does not change the entanglement. Our overall algorithm is efficient (i.e. runs in polynomial time in the number of qubits) for all regular XS-stabilizer groups (cf. also section 2.11). It is worth noting that our method of computing the entanglement entropy uses a very different technique compared to the one typically used for studying the entanglement entropy of Pauli stabilizer states (for example, the methods in [LMRW13]).

The fact that $|\psi\rangle = U_A \otimes U_B |\phi_{A,B}\rangle$ for any bipartition (A, B) implies in particular that any reduced density matrix of $|\psi\rangle$ is a projector since this is the case for all Pauli stabilizer states [HDE+05]. Consequently, all α-Rnyi entanglement entropies of an XS-stabilizer state coincide with the logarithm of the Schmidt rank.

We also formulate the following open problem: for every XS-stabilizer state $|\psi\rangle$, does there exist a single Pauli stabilizer state $|\phi\rangle$ with the same Schmidt rank as $|\psi\rangle$ for *every* bipartition? For example, it would be interesting to know whether the inequalities in [LMRW13] hold for XS-stabilizer states.

As far as multipartite entanglement is concerned, we finally show that the 6-qubit XS-stabilizer state (2.2) is *not* equivalent to any Pauli stabilizer state even if arbitrary local basis changes are allowed.

2.3.4 Efficient algorithms

In section 2.11 we show that several basic tasks can be solved efficiently for an XS-stabilizer state $|\psi\rangle$, provided its regular XS-stabilizer group is known:

1. Compute the entanglement entropy for any bipartition.
2. Compute the expectation value of any local observable.
3. Prepare $|\psi\rangle$ on a quantum computer with a poly-size quantum circuit.
4. Compute the function $f(x)$ in the standard basis expansion

$$|\psi\rangle = \sum_x f(x) |x\rangle.$$

Moreover, we can efficiently construct a basis $\{|\psi_1\rangle, \ldots, |\psi_d\rangle\}$ for any XS-stabilizer code with a *regular* XS-stabilizer group. In particular, we can efficiently compute the degeneracy d of the code. For each $|\psi_j\rangle$ we can again solve all the above tasks efficiently. Finally, we can also efficiently compute logical operators.

The algorithms given in section 2.11 depend heavily on the technical results for XS-stabilizer states and codes given in section 2.9.2, where we characterize several structural properties of these states and codes.

2.4 Basic group theory

In this section we introduce some further basic notions, discuss basic manipulations of XS-stabilizer operators and describe some important subsets and subgroups of XS-stabilizer groups.

2.4.1 Pauli-S group

Let us write

$$[g,h] \coloneqq ghg^{-1}h^{-1}$$

for the commutator of any two group elements g and h. In the following we always assume the elements of a set $\{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$ to be given in the standard form

$$g_j = \alpha^{s_j} X(\vec{a}_j)\, S(\vec{b}_j). \tag{2.3}$$

Lemma 2.4.1 (Commutators)**.**

$$[g_1, g_2] = \bigotimes_{k=1}^{n} (-1)^{a_{1k}a_{2k}(b_{1k}+b_{2k})}\, (\mathrm{i}Z_k)^{a_{1k}b_{2k}-a_{2k}b_{1k}}.$$

Proof. It suffices to prove this for $\mathcal{P}^S$ and $\alpha^s = 1$. So let $g_j = X^{a_j} S^{b_j}$. Then

$$g_2 g_1 = (-1)^{a_1 a_2 (b_1+b_2)}\, (\mathrm{i}Z)^{a_2 b_1 - a_1 b_2} g_1 g_2$$

where we used $S^b X^a = (-\mathrm{i}Z)^{ab} X^a S^b$. The claim for the tensor product group $\mathcal{P}_n^S$ follows from applying the above to each component. □

Lemma 2.4.2 (Squares)**.** *Let $g = \alpha^s X^a S^b \in \mathcal{P}^S$. Then*

$$g^2 = \mathrm{i}^{s+ab}\, Z^{(a+1)b}.$$

Lemma 2.4.3 (Multiplication)**.** *There exists $\vec{b}'$ such that*

$$g_1 g_2 \propto X(\vec{a}_1 \oplus \vec{a}_2)\, S(\vec{b}').$$

2.4.2 Important subgroups

For any group $G \subset \mathcal{P}_n^S$ there are two important subgroups.

Definition 2.4.4. *The group*

$$G_D \coloneqq G \cap \langle \alpha I, S_1, \ldots, S_n \rangle = G \cap \left\{ \alpha^s S(\vec{b}) \;\middle|\; s \in \{0,\ldots,7\}, \vec{b} \in \{0,\ldots 3\}^n \right\}$$

is called the diagonal subgroup *and*

$$G_Z \coloneqq G \cap \langle \alpha I, Z_1, \ldots, Z_n \rangle = G \cap \left\{ \alpha^s Z(\vec{c}) \;\middle|\; s \in \{0,\ldots,7\}, \vec{c} \in \{0,1\}^n \right\} \tag{2.4}$$

is called the Z-subgroup.

In other words, the diagonal subgroup G_D contains all elements of G which are diagonal matrices in the computational basis. These are precisely the elements which do not contain any X operators in their tensor product representation (2.1). The Z-subgroup G_Z consists of all Z-type operators. In particular, all commutators and squares of elements in G are contained in G_Z, as can be seen from Lemmas 2.4.1 and 2.4.2.

If G is an XS-stabilizer group, then all its elements must have an eigenvalue 1. Clearly, its Z-subgroup G_Z must then be contained in $\langle \pm Z_1, \ldots, \pm Z_n \rangle \setminus \{-I\}$, otherwise G_Z (and thus G) may contain elements which lack the eigenvalue 1, as is evident from (2.4). In particular, G cannot contain $-I$. This implies that G_Z lies in the centre $Z(G)$ of G. Indeed, every $\mathcal{Z} \in G_Z$ either commutes or anticommutes with all elements of G, however, $[\mathcal{Z}, g] = -I \in G$ for some $g \in G$ would give a contradiction. Furthermore one can easily see from the above that all elements of G_Z have an order of at most 2, thus we conclude that $g^4 = I$ for all $g \in G$ since $g^2 \in G_Z$. We have just proved

Proposition 2.4.5. *Every XS-stabilizer group G satisfies*

1. $-I \notin G$,
2. $G_Z \subset \langle \pm Z_1, \ldots, \pm Z_n \rangle \setminus \{-I\} = \{(-1)^s Z(\vec{c})\} \setminus \{-I\}$,
3. $G_Z \subset Z(G)$,
4. $g^4 = I$ *for all* $g \in G$.

2.4.3 Admissible generating sets

Typically it is computationally hard to check the above necessary conditions for the *entire* group G. Instead, we focus on a small set of generators which fully determine G, like in the Pauli stabilizer formalism. We are interested in finding necessary conditions for such a set to generate an XS-stabilizer group.

While we can build arbitrary words from the generators, of course, commutators and squares of generators will play a distinguished role in this article.

Definition 2.4.6. *Let $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$. Then*

$$\mathcal{C}_\mathcal{S} \coloneqq \{[g_j, g_k] \mid g_j, g_k \in \mathcal{S} \wedge j \neq k\},$$
$$\mathcal{Q}_\mathcal{S} \coloneqq \{g_j^2 \mid g_j \in \mathcal{S}\}.$$

Definition 2.4.7. *A set $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$ is called an* admissible generating set *if*

1. *every g_j has an eigenvalue* 1,
2. *every $[g_j, g_k]$ has an eigenvalue* 1,
3. $[[g_j, g_k], g_l] = I$,
4. $[g_j^2, g_k] = I$.

Clearly, if $G = \langle \mathcal{S} \rangle$ is an XS-stabilizer group, then $\mathcal{S}$ must be an admissible generating set by Proposition 2.4.5 (and the discussion preceding it). The converse is not true: there exist admissible generating sets $\mathcal{S}$ for which $\langle \mathcal{S} \rangle$ is *not* an XS-stabilizer group.

Note that the properties in the above definition are independent in the sense that the first k properties do not imply the next one. It can be checked in $\mathrm{poly}(n, m)$ time whether a given generating set $\mathcal{S}$ is admissible.

We then have the following lemma:

Lemma 2.4.8 (Relative standard form)**.** *If $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$ is an admissible generating set, then the elements of $G = \langle \mathcal{S} \rangle$ are given by*

$$\mathcal{Z}\, g(\vec{x}) \coloneqq \mathcal{Z} g_1^{x_1} \cdots g_m^{x_m} \tag{2.5}$$

where $\vec{x} \in \mathbb{Z}_2^m$ and $\mathcal{Z} \in \langle \mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S} \rangle \subset G_Z$.

Furthermore, for two elements $h = \mathcal{Z}\, g(\vec{x})$ and $h' = \mathcal{Z}' g(\vec{x}')$ we have

$$hh' = \mathcal{Z}'' g(\vec{x} \oplus \vec{x}').$$

Proof. Let $h = g_{\beta_1} g_{\beta_2} \cdots g_{\beta_p} \in G$ an arbitrary word in the generators $\mathcal{S}$. We will show how to reduce it to the form (2.5). Suppose $\beta_{j-1} > \beta_j$ for some j. Since $g_{\beta_{j-1}} g_{\beta_j} = \mathcal{Z} g_{\beta_j} g_{\beta_{j-1}}$ for some $\mathcal{Z} \in \mathcal{C}_\mathcal{S}$ we can reorder the generators locally and move any commutator $\mathcal{Z}$ to the left. (Since $\mathcal{S}$ is admissible, $\mathcal{Z}$ commutes with all generators.) Repeating this procedure we arrive at $h = \mathcal{Z} g_1^{x_1} \cdots g_m^{x_m}$ for some $\mathcal{Z} \in \langle \mathcal{C}_\mathcal{S} \rangle$, where the exponents x_j may still be arbitrary integers. We can restrict them to $\{0, 1\}$ by extracting squares of generators and moving them to the left. We obtain $h = \mathcal{Z}\mathcal{Z}' g_1^{x_1} \cdots g_m^{x_m}$ for some $\mathcal{Z}' \in \langle \mathcal{Q}_\mathcal{S} \rangle$ which proves the first claim. The second claim follows easily from a similar argument. □

The diagonal subgroup G_D will play an important role in the formalism. Here we give a method to compute the generators of the diagonal subgroup G_D efficiently.

Lemma 2.4.9. *If $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$ is an admissible generating set and $G = \langle \mathcal{S} \rangle$, then a generating set of G_D can be found in* $\mathrm{poly}(n, m)$ *time.*

Proof. We see from Lemma 2.4.8 that G_D is generated by $\mathcal{C}_\mathcal{S}$, $\mathcal{Q}_\mathcal{S}$ and those elements $g(\vec{x})$ which are diagonal. Hence we only need to find a generating set for the latter. Assume that the generators of G are given in the standard form (2.3) and define the $n \times m$ matrix

$$A \coloneqq [\vec{a}_1 \quad \ldots \quad \vec{a}_m]$$

whose columns are the bit strings $\vec{a}_j$. It follows from Lemma 2.4.3 that $g(\vec{x}) \propto X(A\vec{x})\, S(\vec{b}')$ for some $\vec{b}'$. This implies that $g(\vec{x})$ is a diagonal operator if and only if $A\vec{x} = 0$ over $\mathbb{Z}_2$. Denote a basis of the solution space of this linear system by $\{\vec{u}_i\}$. Such a basis can be computed efficiently. Notice that by Lemma 2.4.8 we have $g(\vec{u}_i \oplus \vec{u}_j) = \mathcal{Z}\, g(\vec{u}_i)\, g(\vec{u}_j)$ for any two basis vectors $\vec{u}_i$ and $\vec{u}_j$ and some $\mathcal{Z} \in \langle \mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S} \rangle$. This implies that all diagonal elements $g(\vec{x})$ can be generated by $\mathcal{C}_\mathcal{S}$, $\mathcal{Q}_\mathcal{S}$ and $\{g(\vec{u}_i)\}$, and so can G_D. Finally we note that the length of this generating set is $\mathrm{poly}(m, n)$. □

2.5 Commuting parent Hamiltonian

In this section we show that the space stabilized by $\{g_j\}$ can also be described by the ground space of a set of commuting Hamiltonians. In fact, the Hamiltonians are monomial.

Let $G = \langle \mathcal{S} \rangle$ be an XS-stabilizer group with the generators $\mathcal{S} = \{g_1, \ldots, g_m\}$ and the corresponding code $\mathcal{L}_G$. While it is straightforward to turn each generator into a Hermitian projector onto its stabilized subspace, these projectors will *not* commute with each other in general. Perhaps surprisingly, we can still construct a commuting parent Hamiltonian for $\mathcal{L}_G$ by judiciously choosing a subset of G such that a) this subset yields a commuting Hamiltonian with the *larger* ground state space $\mathcal{L} \supset \mathcal{L}_G$, and b) all generators mutually commute when restricted to $\mathcal{L}$. We will call $\mathcal{L}$ the *gauge-invariant subspace* in the following.

We claim that the subset $\mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S} \subset G$ precisely fits this strategy. First, let us define $P_g \coloneqq (I+g)/2$ for arbitrary $g \in G$. It is easy to see that all $P_\mathcal{Z}$ with $\mathcal{Z} \in \mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S}$ are Hermitian projectors which commute with each other and all elements of G. We may define the gauge-invariant subspace as the image of the Hermitian projector $P \coloneqq \prod_\mathcal{Z} P_\mathcal{Z}$ which commutes with all $P_\mathcal{Z}$ and all elements of G by construction. Moreover, note that

$$P\mathcal{Z} = P, \tag{2.6}$$

in other words, the gauge-invariant subspace "absorbs" commutators and squares of generators. Second, it is easy to check that all PP_{g_j} with $g_j \in \mathcal{S}$ are Hermitian projectors which mutually commute. Indeed, they are projectors since $(PP_{g_j})^2 = (P^2 + 2P^2 g_j + P^2 g_j^2)/4 = PP_{g_j}$ where we used (2.6). Moreover, they are Hermitian since $(Pg_j)^\dagger = g_j^3 P = Pg_j$ where we used Proposition 2.4.5 and (2.6). Finally, they commute with each other because

$$Pg_k Pg_j = Pg_k g_j = P\mathcal{Z} g_j g_k = Pg_j g_k = Pg_j Pg_k$$

for some $\mathcal{Z} \in \mathcal{C}_\mathcal{S}$ which is absorbed by virtue of (2.6).

We can now define the commuting Hamiltonian associated with G (and $\mathcal{S}$) by

$$H_{G,\mathcal{S}} \coloneqq \sum_{\mathcal{Z}} (I - P_\mathcal{Z}) + \sum_{g_j \in \mathcal{S}} (I - PP_{g_j}).$$

It remains to show that the space annihilated by $H_{G,\mathcal{S}}$ is precisely the XS-stabilizer code $\mathcal{L}_G$. It is easy to see that a state $|\psi\rangle$ has zero energy if it is stabilized by G. Conversely, if $|\psi\rangle$ has zero energy then $P_\mathcal{Z}|\psi\rangle = |\psi\rangle$ and $PP_{g_j}|\psi\rangle = |\psi\rangle$ follow directly. The former condition actually implies $P|\psi\rangle = |\psi\rangle$, hence the latter turns into $P_{g_j}|\psi\rangle = |\psi\rangle$ from which we deduce $g_j|\psi\rangle = |\psi\rangle$.

Remark 2.5.1 (Locality)**.** *It is not hard to see the above construction of a commuting Hamiltonian can be modified to preserve the locality of g_j. Assume g_j is local on a d-dimension lattice. Then by construction, $P_\mathcal{Z}$ are also local. Thus the only nonlocal terms in the Hamiltonian are PP_{g_j}, and below we show how to make a modification such that they become local. We say g_k is a neighbour of g_j if g_j and g_k act on some common qubits, and we denote that by $k \in n(j)$ (we also set $j \in n(j)$ for our purpose). It is easy to check that if we replace the PP_{g_j} terms in the Hamiltonian by*

$$\Big(\prod_{k \in n(j)} P_{jk}\Big) P_{g_j},$$

the Hamiltonian is still commuting, while it is now local on the lattice.

Remark 2.5.2 (Quantum error correcting code)**.** *We can use XS-stabilizer codes $\mathcal{L}_G$ for quantum error correction. Here it is important that error syndromes can be measured* simultaneously *which seems impossible if the XS-stabilizer group G is non-Abelian. Yet we can exploit the commuting stabilizers constructed above and extract the error syndromes in two rounds. First we measure the syndromes of the mutually commuting stabilizers in the subset $\mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S}$ and correct as necessary. We are now guaranteed to be in the gauge-invariant subspace where the original generators $\{g_j\}$ commute. We can thus measure their syndromes simultaneously in the second round.*

2.6 Concepts from the monomial matrix formalism

In this subsection we introduce some definitions and theorems from [VdN11] that will be useful later.

In [VdN11], we consider a group $G = \langle U_1, \ldots, U_m \rangle$, where each U_j is a unitary monomial operator, i.e.

$$U_j = P_j D_j$$

where P_j is a permutation matrix and D_j is a diagonal unitary matrix. Define $\mathfrak{P}$ to be the permutation group generated by P_j. The goal of [VdN11] is to study the space of states that satisfy

$$U_j|\psi\rangle = |\psi\rangle \qquad \text{for every } j = 1, \ldots, m.$$

Given a computational basis state $|x\rangle$, following [VdN11] we define the orbit $\mathcal{O}_x$ to be

$$\mathcal{O}_x = \{|y\rangle \mid \exists P \in \mathfrak{P} \colon P|x\rangle = |y\rangle\}.$$

We also define G_x to be the subgroup of all $U \in G$ that have $|x\rangle$ as an eigenvector. Then we have the following theorem

Theorem 2.6.1. *Consider a group $G = \langle U_1, \ldots, U_m \rangle$ of monomial unitary matrices.*

(a) There exists a state $|\psi\rangle \neq 0$ stabilized by G if and only if there exists a computational basis state $|x\rangle$ such that

$$U|x\rangle = |x\rangle \qquad \text{for all } U \in G_x. \tag{2.7}$$

(b) For every computational basis state $|x\rangle$ satisfying (2.7), there exists a state $|\psi_x\rangle$ stabilized by G, which is of the form

$$|\psi_x\rangle = \frac{1}{\sqrt{|\mathcal{O}_x|}} \sum_{|y\rangle \in \mathcal{O}_x} f(y)|y\rangle,$$

where $|f(y)| = 1$ for all $y \in \mathcal{O}_x$. Moreover, there exists a subset $\{|x_1\rangle, \ldots, |x_d\rangle\}$ (each satisfying (2.7)) such that

- *the orbits $\mathcal{O}_{x_i}$ are mutually disjoint;*
- *the set of all x satisfying (2.7) is precisely $\mathcal{O}_{x_1} \cup \cdots \cup \mathcal{O}_{x_d}$;*
- *$\{|\psi_{x_1}\rangle, \ldots, |\psi_{x_d}\rangle\}$ is a basis of the space stabilized by G. In particular, d is the dimension of this space.*

2.7 Computational complexity of the XS-stabilizer problem

Here we address the computational complexity of determining whether a subgroup G of the Pauli-S group, specified in terms of a generating set, *is* an XS-stabilizer group, i.e. whether there exists a quantum state $|\psi\rangle \neq 0$ that is stabilized by G. More precisely, the problem can be formulated as

Problem XS-Stabilizer.

Input A list of $s_j \in \{0, \ldots, 7\}$, $\vec{a}_j \in \{0,1\}^n$ and $\vec{b}_j \in \{0, \ldots, 3\}^n$ where $j = 1, \ldots, m$, which describe a set $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$.

Output If there exists a quantum state $|\psi\rangle \neq 0$ such that $g_j|\psi\rangle = |\psi\rangle$ for every j then output YES; otherwise output NO.

To show an efficient algorithm exists for a computational problem, it suffices to write down the particular algorithm. However, to show that there is no efficient algorithm for a certain problem is a much harder task. This is often done by demonstrating that the computational problem belongs to a hard complexity class, such as NP-hard. Roughly speaking, if one problem in NP-hard has an efficient algorithm, then we will immediately find efficient algorithms for hundreds of other hard problems. Given that researchers have spent decades trying without success, it is likely the problems in NP-hard do not have efficient algorithms.

We will show that the above computational problem related to XS-stabilizer likely does not have an efficient algorithm by proving the following theorem:

Theorem 2.7.1. *The* XS-Stabilizer *problem is* NP*-hard.*

Proof. We will show this via a reduction from the Positive 1-in-3-Sat problem which is NP-complete [Sch78]. The Positive 1-in-3-Sat problem is to determine whether a set of logical clauses in n Boolean variables can be satisfied simultaneously or not. Each clause has three variables exactly one of which must be satisfied. We may express such a clause C_j as

$$x_{w_{j1}} + x_{w_{j2}} + x_{w_{j3}} = 1 \tag{2.8}$$

for variables $x_{w_{jk}} \in \{0,1\}$ and $1 \leq w_{jk} \leq n$.

We construct a corresponding instance of the XS-Stabilizer problem by encoding each clause C_j in a generator

$$g_j = \mathrm{i}^3 S_{w_{j1}} S_{w_{j2}} S_{w_{j3}} \in \mathcal{P}_n^S.$$

Since all g_j are diagonal this stabilizer problem is equivalent to determining whether there exists a computational basis state $|x_1, \ldots, x_n\rangle$ stabilized by every g_j, i.e. whether all equations

$$\mathrm{i}^{3+x_{w_{j1}}+x_{w_{j2}}+x_{w_{j3}}} = 1 \tag{2.9}$$

have a common solution. Since (2.8) and (2.9) are equivalent we have shown that the XS-Stabilizer problem is at least as hard as the Positive 1-in-3-Sat problem. □

The NP-hardness of the XS-Stabilizer problem is in sharp contrast with the corresponding problem in the Pauli stabilizer formalism, which is known to be in P (in other words, there is an efficient algorithm which has running time scaling polynomially with respect to the size of inputs).

Next we show that the XS-Stabilizer problem is in NP, which means there is an efficient classical proof allowing to verify whether a group is indeed an XS-stabilizer group. This is in contrast to many other problems involving quantum mechanics that belongs to the (likely) harder computational class QMA-hard.

Theorem 2.7.2. *The* XS-Stabilizer *problem is in* NP.

Proof. We first determine if $\mathcal{S}$ is an admissible generating set, which can be done efficiently. If it is not, $\langle\mathcal{S}\rangle$ cannot be an XS-stabilizer group, hence we output NO. If $\mathcal{S}$ is found to be an admissible generating set, we proceed with the group $G = \langle\mathcal{S}\rangle$ as follows. Given a computational basis state $|x\rangle$, recall the definition of the set G_x in Section 2.6. Note that every XS-operator $\alpha^s X(\vec{a})\, S(\vec{b})$ maps $|x\rangle$ to $\lambda|x \oplus \vec{a}\rangle$ for some complex phase λ. This implies that $G_x = G_D$ for every x. Then, by Theorem 2.6.1(a), to check whether G is an XS-stabilizer group, we only need to check whether there is a computational basis state $|z\rangle$ stabilized by G_D. Note that a generating set $\{D_1, \ldots, D_r\}$ of G_D can be computed efficiently owing to Lemma 2.4.9. Furthermore, $|z\rangle$ is stabilized by G_D if and only if it is stabilized by every generator D_j. Summarizing, we find that G is an XS-stabilizer group iff there exists a computational basis state $|z\rangle$ satisfying $D_j|z\rangle = |z\rangle$ for all j. So if G is an XS-stabilizer group, a classical string z satisfying these conditions will serve as a proof since the equations $D_j|z\rangle = |z\rangle$ can be verified efficiently. This shows that the XS-Stabilizer problem is in NP. □

2.8 Regular XS-stabilizer groups

The results in section 2.7 imply that working with the XS-stabilizer formalism is computationally hard in general. In order to recover tractability, we may impose certain restrictions on the type of stabilizers we can have. In particular, we have the following theorem.

Theorem 2.8.1. *Let $\mathcal{S} = \{g_1, \ldots, g_m\} \subset \mathcal{P}_n^S$ and $G = \langle\mathcal{S}\rangle$. If $G_D = G_Z$ then the* XS-Stabilizer *problem is in* P. *Moreover, this condition can be checked efficiently.*

Proof. As in the proof of Theorem 2.7.2, we only need to consider admissible generating sets, and we need to decide whether there exists a computational basis state stabilized by G_D. Since Lemma 2.4.9 states that G_D is generated by $\{D_1, \ldots, D_r\} = \mathcal{C}_\mathcal{S} \cup \mathcal{Q}_\mathcal{S} \cup \{g(\vec{u}_i)\}$ with $r = \mathrm{poly}(m, n)$ we can efficiently check whether $\{g(\vec{u}_i)\} \subset G_Z$ or some $g(\vec{u}_i)$ contains an S operator, i.e. whether $G_D = G_Z$ or not. Now let us assume that $G_D = G_Z$. By Proposition 2.4.5 we immediately deduce that $D_j = (-1)^{s_j} Z(\vec{c}_j)$ for some $s_j \in \{0, 1\}$ and $\vec{c}_j \in \{0, 1\}^n$. Then a computational basis state $|\vec{z}\rangle$ stabilized by G_D is equivalent to a nontrivial common solution of the equations $\vec{c}_j \cdot \vec{z} = s_j$. These are polynomially many linear equations in n variables over $\mathbb{Z}_2$ and can hence be solved in polynomial time. This proves that the restricted XS-Stabilizer problem is indeed in P. □

Motivated by this result, we call any XS-stabilizer group G with $G_D = G_Z$ a *regular XS-stabilizer group.*

Remark 2.8.2. *We want to mention that diagonal elements containing S operators play a crucial role in certain examples, which we will discuss in section 2.12. However, as will be proved in theorem 2.12.1, we can construct a basis of the space stabilized by a general XS-stabilizer group such that each basis state is described by a regular XS-stabilizer group.*

Here we give a sufficient condition for an XS-stabilizer group to be regular.

Lemma 2.8.3. *If an XS-stabilizer group $G \subset \mathcal{P}_n^S$ has a generating set $\mathcal{S} = \{g_1, \ldots, g_t, g_{t+1}, \ldots, g_m\}$ where*

$$g_j = \begin{cases} \alpha^{s_j} X(\vec{a}_j)\, S(\vec{b}_j) & \text{if } j \leq t, \\ (-1)^{s_j} Z(\vec{c}_j) & \text{else} \end{cases} \tag{2.10}$$

and $\{\vec{a}_1, \ldots, \vec{a}_t\}$ are linearly independent over $\mathbb{Z}_2$, then it is regular.

Proof. We see from Lemma 2.4.8 that G_D is generated by $\mathcal{C}_\mathcal{S}$, $\mathcal{Q}_\mathcal{S}$ and those elements $g(\vec{x}) = g_1^{x_1} \cdots g_m^{x_m}$ which are diagonal. In order to show that $G_D = G_Z$ we only need to show that all diagonal elements $g(\vec{x})$ are Z-type operators. It follows from Lemma 2.4.3 that $g(\vec{x}) \propto X(A\vec{x})\, S(\vec{b}')$ for the $n \times m$ matrix

$$A = [\vec{a}_1 \quad \ldots \quad \vec{a}_t \quad 0 \quad \ldots \quad 0]$$

and some $\vec{b}'$. This implies that $g(\vec{x})$ is a diagonal operator if and only if $A\vec{x} = 0$ over $\mathbb{Z}_2$. Since the bit strings $\vec{a}_j$ are linearly independent this can only be true if $x_j = 0$ for all $j \leq t$. Thus every diagonal element $g(\vec{x}) = g_{t+1}^{x_{t+1}} \cdots g_m^{x_m}$ is indeed a Z-type operator. □

Theorem 2.8.4 (Normal form)**.** *Every XS-stabilizer group $G \subset \mathcal{P}_n^S$ has a generating set $\mathcal{S} = \{g_1, \ldots, g_t, g_{t+1}, \ldots, g_m\}$ where*

$$g_j = \begin{cases} \alpha^{s_j} X(\vec{e}_j, \vec{w}_j)\, S(\vec{b}_j) & \text{if } j \leq t, \\ \mathrm{i}^{s_j} S(\vec{b}_j) & \text{else} \end{cases}$$

for the canonical basis vectors $\vec{e}_j \in \mathbb{Z}_2^t$ and some $\vec{w}_j \in \mathbb{Z}_2^{n-t}$. Furthermore, $G_D = \langle g_{t+1}, \ldots, g_m \rangle$.

Any other generating set $\mathcal{S}' = \{h_1, \ldots, h_l\}$ of G can be reduced to $\mathcal{S}$ in $\mathrm{poly}(n, l)$ *time and the length of $\mathcal{S}$ is* $\mathrm{poly}(n, l)$.

Proof. We will prove this by explicitly reducing $\mathcal{S}' = \{h_1, \ldots, h_l\}$. By Lemma 2.4.9 we can efficiently find a generating set $\{D_1, \ldots, D_r\}$ of G_D from $\mathcal{S}'$. Next we can extend it to a generating set of G by adding a minimal subset of $\mathcal{S}'$ which can be found efficiently. Indeed, suppose we added all generators $h_j = \alpha^{s_j} X(\vec{a}_j)\, S(\vec{b}_j)$. If $\{\vec{a}_1, \ldots, \vec{a}_l\}$ are linearly dependent over $\mathbb{Z}_2$, we may assume that $\vec{a}_l = y_1\vec{a}_1 \oplus \cdots \oplus y_{l-1}\vec{a}_{l-1}$ for some $y_j \in \mathbb{Z}_2$ without loss of generality. Then Lemma 2.4.3 implies $h_1^{y_1} \cdots h_{l-1}^{y_{l-1}} = D h_l$ for some diagonal element $D \subset \mathcal{P}_n^S$. It is not difficult to see that actually $D \in G$ and hence $D \in G_D$. This means that the generator $h_l \in \langle D_1, \ldots, D_r, h_1, \ldots, h_{l-1} \rangle$ is redundant. So we can find a minimal subset of $\mathcal{S}'$ by finding bit strings $\vec{a}_j$ which form a basis of the linear space $\langle \vec{a}_1, \ldots, \vec{a}_l \rangle$. Note that this only involves Gaussian elimination over $\mathbb{Z}_2$. Now let $\{h_1, \ldots, h_t\}$ be the desired subset of $\mathcal{S}'$ after relabeling the generators h_j, so that $G = \langle h_1, \ldots, h_t, D_1, \ldots, D_r \rangle$.

We can arrange the bit strings $\vec{a}_1$, , $\vec{a}_t$ in the $n \times t$ matrix

$$A \coloneqq [\vec{a}_1 \quad \ldots \quad \vec{a}_t].$$

Since the $\vec{a}_j$ are linearly independent, by Gaussian elimination and suitable permutation of columns, we can efficiently transform A into

$$PAR = \begin{bmatrix} \mathbb{1}_t \\ W \end{bmatrix}$$

for some $n \times n$ permutation matrix P, $t \times t$ invertible matrix R and $(n-t) \times t$ matrix W. By relabeling the qubits according to the permutation defined by P and by multiplying the generators h_1, , h_t according to the transformation R, we obtain an equivalent set of generators $\{g_1, \ldots, g_t\}$ such that $g_j = \alpha^{s'_j} X(\vec{e}_j, \vec{w}_j)\, S(\vec{b}'_j)$ for some s'_j and $\vec{b}'_j$. Here $\vec{w}_j$ denotes the j-th column of W.

Finally note that $D_j = \mathrm{i}^{s'_j} S(\vec{b}'_j)$ for some s'_j and $\vec{b}'_j$ since G is an XS-stabilizer group. We conclude that $\mathcal{S} = \{g_1, \ldots, g_t, D_1, \ldots, D_r\}$ is the desired generating set of G. □

Corollary 2.8.5. *Every regular XS-stabilizer group $G \subset \mathcal{P}_n^S$ has a generating set $\mathcal{S} = \{g_1, \ldots, g_t, g_{t+1}, \ldots, g_m\}$ where*

$$g_j = \begin{cases} \alpha^{s_j} X(\vec{e}_j, \vec{w}_j)\, S(\vec{b}_j) & \text{if } j \leq t, \\ (-1)^{s_j} Z(\vec{c}_j) & \text{else} \end{cases}$$

for the canonical basis vectors $\vec{e}_j \in \mathbb{Z}_2^t$ and some $\vec{w}_j \in \mathbb{Z}_2^{n-t}$. Furthermore, $G_D = \langle g_{t+1}, \ldots, g_m \rangle$.

Unless stated otherwise, we will always work with regular XS-stabilizer groups from this point on.

2.9 Constructing a basis of a regular XS-stabilizer code

The goal of this section is to construct a basis for a regular XS-stabilizer code. For each state of the basis, we will give an explicit form of its expansion in the computational basis. To achieve this (in sections 2.9.2, 2.9.3 and 2.9.4), we will first introduce some preliminary material on quadratic and cubic functions (in section 2.9.1).

2.9.1 Quadratic and cubic functions

In this chapter we will often deal with functions of the form $i^{\sum x_j x_k}$ or $(-1)^{\sum x_j x_k x_l}$ with $x_j \in \{0,1\}$. Here we list some properties of such functions that will become useful later. First we have the following lemma.

Lemma 2.9.1 (Exponentials of parities)**.** *Let $x_1, \ , \ x_n \in \{0,1\}$. Then*

$$\alpha^{x_1 \oplus \cdots \oplus x_n} = \alpha^{\sum_j x_j}\, \mathrm{i}^{-\sum_{j<k} x_j x_k}\, (-1)^{\sum_{j<k<l} x_j x_k x_l}$$

and

$$i^{x_1 \oplus \cdots \oplus x_n} = \mathrm{i}^{\sum_j x_j}\, (-1)^{\sum_{j<k} x_j x_k}.$$

Proof. The first equation is proved in [BH12]. The second equation follows by similar arguments. □

All exponents in this Lemma are homogeneous polynomials of degree at most 3. Below we will often only be interested in whether a given exponent is a quadratic or cubic polynomial, but not in its concrete form. We will therefore use $l(x)$, $q(x)$ and $c(x)$ to represent arbitrary linear, quadratic and cubic polynomials in $\mathbb{Z}[x_1, \ldots, x_n]$ respectively. (These polynomials need not be homogeneous.) Using this notation, the Lemma can be summarized as

$$\alpha^{x_1 \oplus \cdots \oplus x_n} = \alpha^{l(x)}\, \mathrm{i}^{q(x)}\, (-1)^{c(x)},$$
$$\mathrm{i}^{x_1 \oplus \cdots \oplus x_n} = i^{l(x)}\, (-1)^{q(x)}.$$

Let $\mathcal{F}$ denote the class of all functions $f\colon \{0,1\}^n \to \mathbb{C}$ having the form

$$f(x) = \alpha^{l(x)}\, \mathrm{i}^{q(x)}\, (-1)^{c(x)} \qquad \text{for all } x \in \{0,1\}^n.$$

Note that $\mathcal{F}$ is closed under multiplication, i.e. if f and g belong to this class, then so does fg.

Remark 2.9.2. *Linear phases $\alpha^{l(x)}$ are generated by $\{\alpha^{x_j}\}$ via multiplication since $\alpha^{l(x)} = \prod_{j=1}^n (\alpha^{x_j})^{\lambda_j}$ for a linear polynomial $l(x) = \sum_{j=1}^n \lambda_j x_j$. By the same token, $\{\mathrm{i}^{x_j}, \mathrm{i}^{x_j x_k}\}$ generate all quadratic phases $\mathrm{i}^{q(x)}$ and $\{(-1)^{x_j}, (-1)^{x_j x_k}, (-1)^{x_j x_k x_l}\}$ all cubic phases $(-1)^{c(x)}$. This implies*

$$\mathcal{F} = \langle \alpha^{x_j}, \mathrm{i}^{x_j x_k}, (-1)^{x_j x_k x_l} \rangle.$$

□

Lemma 2.9.3 (Covariance)**.** *Let Q be a linear map in the vector space $\mathbb{Z}_2^n = \{0,1\}^n$. If a function f belongs to $\mathcal{F}$ then so does $f \circ Q$.*

Proof. Let $x = (x_1, \ldots, x_n) \in \mathbb{Z}_2^n$ and $y = Q(x)$, so each $y_j = \bigoplus_{k=1}^n Q_{jk}x_k$ is the parity of some substring of x. By Remark 2.9.2 it is enough to show that the phases α^{y_j}, $\mathrm{i}^{y_j y_k}$ and $(-1)^{y_j y_k y_l}$ belong to $\mathcal{F}$ since any $f(y)$ equals some product of them.

It is immediate from Lemma 2.9.1 that every $\alpha^{y_j} \in \mathcal{F}$. Furthermore, Lemma 2.9.1 implies

$$\mathrm{i}^{y_j y_k} = \left(\mathrm{i}^{l(x)}(-1)^{q(x)}\right)^{y_k} = \mathrm{i}^{l(x)\,y_k}(-1)^{q(x)\,y_k} = (\mathrm{i}^{y_k})^{l(x)}(-1)^{c(x)}$$

for some linear, quadratic and cubic polynomials l, q and c respectively. Invoking Lemma 2.9.1 once more we see that $(\mathrm{i}^{y_k})^{l(x)} \in \mathcal{F}$, thus $\mathrm{i}^{y_j y_k} \in \mathcal{F}$. Finally, it is easy to check that $(-1)^{y_j y_k y_l} = (-1)^{c'(x)}$ for some cubic polynomial c', which shows $(-1)^{y_j y_k y_l} \in \mathcal{F}$. □

2.9.2 Constructing a basis

Consider an n-qubit XS-stabilizer code $\mathcal{H}_G$ with regular stabilizer group $G = \langle g_1, \ldots, g_m\rangle$. Without loss of generality, we may assume that the generators g_j have the form given in Corollary 2.8.5. We will construct a basis for this XS-stabilizer code by applying the monomial matrix method outlined in section 2.6.

Denote the $(n-t) \times t$ matrix $W \coloneqq [\vec{c}_1|\cdots|\vec{c}_t]$ (as in the proof of Theorem 2.8.4). Define V to be the linear subspace of $\mathbb{Z}_2^n$ consisting of all couples (x, Wx) with $x \in \mathbb{Z}_2^t$. A basis of this space is given by the vectors $(\vec{e}_i, \vec{c}_i)$ where $\vec{e}_i$ is the i-th canonical basis vector in $\mathbb{Z}_2^t$. Furthermore consider the set V_D of those n-bit strings z satisfying $D|z\rangle = |z\rangle$ for all $D \in G_D$. This coincides with the set of all z satisfying $g_j|z\rangle = |z\rangle$ for all $j = t+1, \ldots, m$, since these g_j generate the diagonal subgroup by Corollary 2.8.5. Since each of these g_j has the form $(-1)^{s_j}Z(\vec{b}_j)$, V_D is the set of all z satisfying $\vec{b}_j^T z = s_j$ for all $j = t+1, \ldots, m$. The set V_D is thus an affine subspace of $\mathbb{Z}_2^n$. A basis of V_D can be computed efficiently.

Note that every Pauli-S operator $g = i^s X(\vec{a})S(\vec{b})$ is a monomial unitary matrix, where $X(\vec{a})$ is the corresponding permutation matrix and $i^s S(\vec{b})$ the corresponding diagonal matrix (recall subsection 2.6). It follows that the permutation group $\mathfrak{P}$ associated with G is generated by the operators $X_j X(\vec{c}_j)$. Recalling the definition of the space V, this implies that $\mathfrak{P} = \{X(v) \mid v \in V\}$. Furthermore, the orbit of a computational basis state $|x\rangle$ is the coset of V containing x i.e. $\mathcal{O}_x = x + V$. To see this, note that

$$\{X(v)|x\rangle \mid v \in V\} = \{|x+v\rangle \mid v \in V\}.$$

Applying theorem 2.6.1(b), we conclude that there exist orbits $\mathcal{O}_1$, , $\mathcal{O}_d$ such that

$$V_D = \mathcal{O}_1 \cup \cdots \cup \mathcal{O}_d$$

and d coincides with the dimension of the XS-stabilizer code. Note that we can efficiently compute d: each orbit has size $|V|$ and thus $d|V| = |V_D|$; since both $|V|$ and $|V_D|$ can be computed efficiently (as we know bases for both of these spaces), we can efficiently compute d. Note that d is a power of two, since both $|V|$ and $|V_D|$ are powers of two. Finally, a set of strings $\vec{\lambda}_1, \ldots, \vec{\lambda}_d \in V_D$ such that $\mathcal{O}_i = \vec{\lambda}_i + V$ can be computed in $\mathrm{poly}(n, m, d)$ time.

As a corollary of the above discussion, we also note:

Lemma 2.9.4. *The XS-stabilizer code $\mathcal{H}_G$ is one-dimensional, i.e. it is an XS-stabilizer state, iff $|V| = |V_D|$.*

By theorem 2.6.1, for each vector $\vec{\lambda} \in V_D$ there exists a state

$$|\psi\rangle = \sum_{x \in \mathbb{Z}_2^t} g(x)\,|x + \vec{\lambda}_1, Wx + \vec{\lambda}_2\rangle. \tag{2.11}$$

stabilized by G, where $g(x)$ is some function that satisfies $|g(x)| = |V|^{-\frac{1}{2}}$ for all x and where we have partitioned $\vec{\lambda} = (\vec{\lambda}_1, \vec{\lambda}_2)$ with $\vec{\lambda}_1$ representing the first t components and $\vec{\lambda}_2$ the last $n-t$ components of λ. By performing the substitution $x \mapsto x + \vec{\lambda}_1$ and denoting $f(x) \coloneqq g(x + \vec{\lambda}_1)$ and $\vec{\mu} \coloneqq \vec{\lambda}_2 + W\vec{\lambda}_1$, we find

$$|\psi\rangle = \sum_{x \in \mathbb{Z}_2^t} f(x)\,|x, Wx + \vec{\mu}\rangle. \tag{2.12}$$

Lemma 2.9.5. *Suppose the state $|\psi\rangle$ of the form (2.11) is stabilized by a regular XS-stabilizer group G. Then based on the generators g_j of G and the string $\vec{\lambda}$, the following data in equation (2.12) can be computed efficiently: (i) the matrix W and the string $\vec{\mu}$; (ii) the function $f(x)$. Moreover, we can efficiently find a complete set of stabilizers that uniquely stabilize $|\psi\rangle$.*

Proof. That the matrix W and the string $\vec{\mu}$ can be computed efficiently was shown in the argument above the lemma. In order to prove that the function $f(x)$ can be computed efficiently, we note that $g_j|\psi\rangle = |\psi\rangle$ for all $1 \leq j \leq t$ and thus

$$g_t^{x_t} \cdots g_1^{x_1} |\psi\rangle = |\psi\rangle.$$

Comparing the terms with $|x\rangle$ on both sides, we have

$$g_t^{x_t} \cdots g_1^{x_1} |0, \vec{\mu}\rangle = f(x)\,|x, Wx + \vec{\mu}\rangle. \tag{2.13}$$

This shows that $f(x)$ can be computed by computing the phase which appears when computing $g_t^{x_t} \cdots g_1^{x_1}|0, \vec{\mu}\rangle$. Finally, we construct a complete stabilizer of $|\psi\rangle$, showing that this state is an XS-stabilizer state.

We work with the representation (2.11) of $|\psi\rangle$, which implies that $|\psi\rangle$ has the form

$$|\psi\rangle = \sum_{v \in V + \vec{\lambda}} \alpha_v |v\rangle \tag{2.14}$$

(for some coefficients α_v) where we recall the definition of $V \subset \mathbb{Z}_2^n$, which is the linear subspace of all pairs (x, Wx). The orthogonal complement of V is the $(n-t)$-dimensional space of all pairs $(W^T y, y)$ with $y \in \mathbb{Z}_2^{n-t}$. A basis $\{\vec{z}_1, \ldots, \vec{z}_{n-t}\}$ of the latter space can be computed efficiently. It follows that any string $v \in \mathbb{Z}_2^n$ belongs to $V + \vec{\lambda}$ if and only if $\vec{z}_j^T v = \vec{z}_j^T \vec{\lambda}$. Define operators

$$D_j \coloneqq (-1)^{\vec{z}_j{}^T \vec{\lambda}} Z(\vec{z}_j) \qquad j = 1, \ldots, n-t.$$

Then for any string $v \in \mathbb{Z}_2^n$, we have

$$v \in V + \vec{\lambda} \quad \Leftrightarrow \quad D_j\,|v\rangle = |v\rangle \text{ for every } j = 1, \ldots, n-t.$$

We now supplement the initial generators $\{g_1, \ldots, g_m\}$ of G with the operators D_j. Denote the resulting set by $\mathcal{S}'$ and let G' denote the group generated by $\mathcal{S}'$. Clearly, G' is regular since we supplemented the initial group G, which was regular, with Z-type operators. We now claim that G' has $|\psi\rangle$ as its unique stabilized state. First, combining (2.14) with (2.7) shows that $|\psi\rangle$ is stabilized by every operator D_j. This shows that $|\psi\rangle$ is stabilized by G'. Second, let G'_D be the diagonal subgroup of G'. Let U be the set of all $u \in \mathbb{Z}_2^n$ satisfying $D|u\rangle = |u\rangle$ for all $D \in G'_D$. According to the argument above lemma 2.9.4, U is the disjoint union of cosets of V; the number of such cosets is the dimension of the code stabilized by G'. We show that in fact $U = V + \vec{\lambda}$, implying that this dimension is one, so that $|\psi\rangle$ is the unique stabilized state. Since each D_j belongs to G'_D, every $u \in U$ must satisfy $D_j|u\rangle = |u\rangle$ for all $j = 1, \ldots, n-t$. With (2.7) this shows that $U \subseteq V + \vec{\lambda}$. Furthermore, since $D|\psi\rangle = |\psi\rangle$ for every $D \in G'_D$ and since $|\psi\rangle$ has the form (2.14), it follows from that every $u \in V + \vec{\lambda}$ satisfies $D|u\rangle = |u\rangle$ for all $D \in G'_D$. This shows that $V + \vec{\lambda} \subseteq U$ and thus $V + \vec{\lambda} = U$. □

Next we determine the form of the function f in more detail. This will be e.g. useful for constructing a quantum circuit to generate $|\psi\rangle$ (see section 2.11). For simplicity of notation, we rescale the state $|\psi\rangle$ by multiplying it with a suitable constant so that we can assume $f(0) = 1$. We will show that then f belongs to the class $\mathcal{F}$ introduced in section 2.9.1. We recall the identity (2.13). To show that $f \in \mathcal{F}$, we will compute the left hand side of the equation, and we will do this by using induction on k. As for the trivial step of the induction, if $k = 1$, it is easy to see that $f \in \mathcal{F}$. Set $x[k] \coloneqq (x_1, \ldots, x_k, 0, \ldots, 0)$, and assume that

$$g_k^{x_k} \cdots g_1^{x_1} |0, \vec{\mu}\rangle = f(x[k])\,|x[k], Wx[k] + \vec{\mu}\rangle$$

with $f \in \mathcal{F}$. If we apply an S operator to the l-th qubit of $|x[k]\rangle$, we simply obtain the phase i^{x_l}. Second, the j-th bit of $Wx[k] + \vec{\mu}$ is

$$W_{j1}x_1 \oplus \cdots \oplus W_{jk}x_k \oplus \mu_j.$$

Therefore, if we apply an S operator on the corresponding qubit, we will obtain the phase

$$\mathrm{i}^{W_{j1}x_1 \oplus \cdots \oplus W_{jk}x_k \oplus \mu_j} = \mathrm{i}^{W_{j1}x_1 + \cdots + W_{jk}x_k + \mu_j} (-1)^{q(x[k])},$$

for some quadratic polynomial $q(x[k])$ (which contains all possible products of $W_{jl}x_l$ and λ_j, where we have applied lemma 2.9.1. It is then easy to check that

$$g_{k+1}^{x_{k+1}} |x[k], Wx[k] + \vec{\mu}\rangle = \alpha^{l(x[k+1])} \mathrm{i}^{q(x[k+1])} (-1)^{c(x[k+1])} |x[k+1], Wx[k+1] + \vec{\mu}\rangle,$$

where l, q and c are linear, quadratic and cubic polynomials in x_1, , x_{k+1} respectively. So we can conclude that

$$g_t^{x_{k+1}} \cdots g_1^{x_1} |0, \vec{\mu}\rangle = f(x[k+1]) |x[k+1], Wx[k+1] + \vec{\mu}\rangle$$

for some $f \in \mathcal{F}$.

In the argument above, for any j we can also check that in the phase $f(x) = \alpha^{l(x)} i^{q(x)} (-1)^{c(x)}$, the part that depends on both μ_j and x is

$$\mathrm{i}^{l(x)\mu_j} (-1)^{q(x)\mu_j} \tag{2.15}$$

This will be useful for finding the logical operators in section 2.9.3.

Summarizing, we have shown:

Theorem 2.9.6. *Every regular XS-stabilizer state on n qubits has the form*

$$|\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{x \in \mathbb{Z}_2^t} f(x) |x, Wx + \vec{\mu}\rangle \qquad \text{with } f(x) = \alpha^{l(x)} \mathrm{i}^{q(x)} (-1)^{c(x)}. \tag{2.16}$$

up to a permutation of the qubits. Moreover, the polynomials $l(x)$, $q(x)$ and $c(x)$ can be computed efficiently.

Below in theorem 2.12.1 we will show that every XS-stabilizer state affords a regular stabilizer group, so that the above theorem in fact applies to all XS-stabilizer states.

It is interesting to compare theorem 2.9.6 with a similar result for Pauli stabilizer states [DDM03]: every Pauli stabilizer state also has the form (2.16), but where

$$f(x) = \mathrm{i}^{l(x)} (-1)^{q(x)}, \tag{2.17}$$

i.e. there are no cubic terms $(-1)^{c(x)}$, no quadratic terms $i^{q(x)}$ and no linear terms $\alpha^{l(x)}$. It is also known that every state (2.16) with f having the form (2.17) is a valid Pauli stabilizer state. A similar statement does not hold for XS-stabilizer states i.e. not all functions $f \in \mathcal{F}$ are allowed. We will revisit this property in section 2.9.4.

2.9.3 Logical operators

Logical operators play a very important role in understanding the Pauli stabilizer formalism and performing fault tolerant quantum computation. They represent the X and Z operators on the encoded qubits. Since it is not clear yet how to define logical operators for an XS-stabilizer group, in this section we will just construct a set of operators that preserve the code space $\mathcal{H}_G$ (which we assume to have dimension 2^s) and obey the same commutation relations as $\{X_j, Z_j\}$ on s qubits.

We already know there is a set of the basis of a regular XS-stabilizer code $\mathcal{H}_G$ which has the form

$$|\psi_j\rangle = \sum_{x \in \mathbb{Z}_2^t} f_j(x) |x, Wx + \vec{\mu}_j\rangle.$$

Note that $\vec{\mu}_j$ can be viewed as elements of the quotient space V_D/V (for definition of V_D and V see section 2.9.2), which is an affine subspace. So we know that up to some permutation of qubits, each $\vec{\mu}_j$ has the form

$$(y_1, \ldots, y_s, \Lambda y + \vec{\lambda}).$$

Here Λ and $\vec{\lambda}$ can be found by Gaussian elimination, and s is the dimension of the space V_D/V. So for each μ_j, we can find a corresponding y. Assume we do not need to do permutation of qubits in the above step, the vector $|x, Wx + \vec{\mu}_j\rangle$ becomes

$$|x, Wx + (y, \Lambda y + \vec{\lambda})\rangle.$$

For simplicity of notation, we will assume $\vec{\lambda} = 0$. The case $\vec{\lambda} \neq 0$ can be dealt with similarly by the following procedure. We will show how to find operators $\bar{Z}_k$ and $\bar{X}_k$ for $k = 1, \ldots, s$, such that they act on the state $|\psi(y)\rangle \equiv |\psi_j\rangle$ by the following

$$\begin{aligned} \bar{Z}_k |\psi(y)\rangle &= (-1)^{y_k} |\psi(y)\rangle \\ \bar{X}_k |\psi(y)\rangle &= |\psi(y + \vec{e}_k)\rangle, \end{aligned}$$

where $\vec{e}_k$ is the k-th canonical basis vector. It is then easy to see within the space spanned by $\{|\psi_j\rangle\}$, we have $\bar{X}_j \bar{Z}_k = (-1)^{\delta(j,k)} \bar{Z}_k \bar{X}_j$, where $\delta(j,k)$ is the Kronecker delta function. The $\bar{Z}_k$ can be found straightforwardly. For example, we can construct $\bar{Z}_k$ by noticing

$$(-1)^{y_k} = (-1)^{\sum_l W_{kl} x_l + y_k} \prod_l (-1)^{W_{kl} x_l},$$

where W_{kl} is the matrix element of W. In the r.h.s of the above equation, the term $(-1)^{\sum_l W_{kl} x_l + y_k}$ can be achieved by applying Z on the k-th qubit in the block $|Wx + (y, \Lambda y)\rangle$, and the terms $(-1)^{W_{kl} x_l}$ can be obtained by applying Z on the corresponding qubits in the block $|x\rangle$.

The construction of $\bar{X}_k$ is more involved and different from the one used in the Pauli stabilizer formalism. To do the map from $|\psi(y)\rangle$ to $|\psi(y + \vec{e}_k)\rangle$, we need to flip y_k and the corresponding qubits of Λy in the block $|Wx + (y, \Lambda y)\rangle$. This can be done by an X-type operator, which we will denote as $\bar{X}'$. However, we also need to change the coefficient functions $f_j(x)$ correspondingly. By changing y to $y + \vec{e}_k$, we will change $\vec{\mu}_j$ to $\vec{\mu}_{j'} = \mu_j + (\vec{e}_k, \Lambda \vec{e}_k)$. Assume that $\bar{X}' D$ achieves the task

$$\bar{X}' D \sum_{x \in \mathbb{Z}_2^t} f_j(x) |x, Wx + \vec{\mu}_j\rangle = \sum_{x \in \mathbb{Z}_2^t} f_{j'}(x) |x, Wx + \vec{\mu}_{j'}\rangle.$$

Then D can be found by noticing that $f_j(x)$ depends on $\vec{\mu}_j$ by the relation (2.15) (note that in relation (2.15), μ_j is the j-th coordinate of $\vec{\mu}$). We can compute straightforwardly that

$$\frac{f_{j'}(x)}{f_j(x)} = \mathrm{i}^{l(x)} (-1)^{q(x)} \prod_h (-1)^{\mu_{jh}\, l(x)},$$

where μ_{jh} is the h-th coordinate of $\vec{\mu}_j$. The terms $\mathrm{i}^{l(x)} (-1)^{q(x)}$ can be obtained by including S and CZ on the corresponding qubits in D. The terms $(-1)^{\mu_{jh}\, l(x)}$ can also be obtained by Z and CZ by noticing μ_{jh} is the parity of some qubits in $|x, Wx + \vec{\mu}_j\rangle$. Thus we have found $\bar{X} = \bar{X}' D$.

2.9.4 A stronger characterization

Consider an XS-stabilizer state $|\psi\rangle$. We have shown that $|\psi\rangle$ has the form given in theorem 2.9.6. However, not all covariant phases $f \in \mathcal{F}$ are valid amplitudes. Here we characterize precisely the subclass of valid amplitudes.

First we note that the group $\mathcal{P}_n^S$ is closed under conjugation by X, $\sqrt{S}$ and CZ, hence we can make two assumptions about the state:

1. We can assume $\vec{\mu} = 0$ in theorem 2.9.6, since we can apply X to the corresponding qubits and update the stabilizers by conjugation.

2. We can restrict ourselves to studying covariant phases of the form

$$f(x) = \mathrm{i}^{\sum_{j<k} \zeta_{jk} x_j x_k} (-1)^{\sum_{j<k<l} \zeta_{jkl} x_j x_k x_l} \tag{2.18}$$

where ζ_{jk} and ζ_{jkl} take values in $\mathbb{Z}_2$. Indeed, if f is a valid amplitude, then so is

$$f(x)\, \alpha^{l(x)}\, \mathrm{i}^{l'(x)}\, (-1)^{q(x)}$$

for all linear polynomials l, l' and quadratic polynomials q. After all, we can always generate these additional phases by applying suitable combinations of the gates $\sqrt{S}$ and CZ to the state $|\psi\rangle$.

We will treat this class of f as a vector space V_1 over $\mathbb{Z}_2$, with a natural basis given by the functions $i^{x_j x_k}$ and $(-1)^{x_j x_k x_l}$. Similarly, we consider the set of all functions of the form

$$(-1)^{\sum_{j<k} \eta_{jk} x_j x_k}$$

which can also be viewed as a vector space V_2 over $\mathbb{Z}_2$ with basis functions $(-1)^{x_j x_k}$. We define a set of linear mappings $\{F_h \mid h = 1, \ldots, t\}$ from V_1 to V_2 by the rules

$$F_h\, (-1)^{x_j x_k x_l} = \begin{cases} (-1)^{x_k x_l} & \text{if } h = j, \\ (-1)^{x_j x_l} & \text{if } h = k, \\ (-1)^{x_j x_k} & \text{if } h = l, \\ 1 & \text{otherwise,} \end{cases} \tag{2.19}$$

and

$$F_h\, \mathrm{i}^{x_j x_k} = \begin{cases} (-1)^{x_j x_k} & \text{if } h = j \text{ or } h = k, \\ 1 & \text{otherwise.} \end{cases} \tag{2.20}$$

Recall the matrix W that appears in equation (2.12). Let $\vec{w}_j = (w_{j1}, \ldots, w_{jt})$ denote the j-th row of W, for every $j = 1, \ldots, n - t$. We define the quadratic functions $\gamma_j \in V_2$ by

$$\gamma_j(x) = (-1)^{\sum_{k<l} w_{jk} w_{jl} x_k x_l}. \tag{2.21}$$

This definition stems from the fact that when we apply the S gate on the single-qubit standard basis state described by $|w_{j1} x_1 \oplus \cdots \oplus w_{jt} x_t\rangle$, we obtain the phase

$$\mathrm{i}^{w_{j1} x_1 \oplus \cdots \oplus w_{jt} x_t} = \mathrm{i}^{\sum_k w_{jk} x_k}\, \gamma_j(x), \tag{2.22}$$

where we have used lemma 2.9.1. Then we set

$$\Gamma = \mathrm{span}\{\gamma_j\}. \tag{2.23}$$

We will prove the following theorem

Theorem 2.9.7. *Consider any state of the form*

$$|\psi\rangle = \frac{1}{\sqrt{2^t}} \sum_{x \in \mathbb{Z}_2^t} f(x)\, |x, Wx\rangle$$

with $f \in V_1$. If f satisfies

$$F_h(f) \in \Gamma \qquad \textit{for all } 1 \le h \le t \tag{2.24}$$

then $|\psi\rangle$ is an XS-stabilizer state.

Proof. Assuming that f satisfies condition (2.24), we will show how to construct a set of XS-stabilizers that uniquely stabilize $|\psi\rangle$. Consider XS-operators

$$g_j = X_j X(\vec{a}_j) S(\vec{b}_j), \qquad 1 \leq j \leq t,$$

where X_j denotes the Pauli matrix X acting on the j-th qubit, where the $X(\vec{a}_j)$ are X-type operators that only act on qubits $t+1$ to n with $\vec{a}_j$ the j-th column of the matrix W. The strings $\vec{b}_j$ are at the moment unspecified. Furthermore, define

$$g_j = Z(\vec{c}_j) \qquad t+1 \leq j \leq n$$

where $\{\vec{c}_{t+1}, \ldots, \vec{c}_n\} \subseteq \mathbb{Z}_2^n$ form a basis of the orthogonal complement of the subspace

$$V = \{(x, Wx) \mid x \in \mathbb{Z}_2^t\} \subseteq \mathbb{Z}_2^n.$$

For every $y, z \in \mathbb{Z}_2^n$ we have

$$Z(z)\,|y\rangle = (-1)^{z^T y}\,|y\rangle.$$

This implies that $g_j|\psi\rangle = |\psi\rangle$ for every $j = t+1, \ldots, n$. Next we show that, for a suitable choice of $\vec{b}_j$, the operator g_j stabilizes $|\psi\rangle$ for every $j = 1, \ldots, t$. This last condition is equivalent to $X_j X(\vec{a}_j)|\psi\rangle = S(\vec{b}_j)|\psi\rangle$. Note that

$$X_j X(\vec{a}_j)\,|\psi\rangle = \sum_x f(x)\,|x + e_j, W(x+e_j)\rangle = \sum_x f(x+e_j)\,|x, Wx\rangle$$

since $\vec{a}_j$ is the j-th column of W and thus $\vec{a}_j = We_j$. Thus, the condition $g_j|\psi\rangle = |\psi\rangle$ is equivalent to

$$f(x+e_j)\,|x, Wx\rangle = S(\vec{b}_j)\,f(x)\,|x, Wx\rangle.$$

By using the fact that f has the form (2.24) and by applying the definition of F_j, it is easy to check that we have

$$\frac{f(x+e_j)}{f(x)} = \mathrm{i}^{l_j(x)}\,(F_j \circ f)(x).$$

for some linear function l_j. Summarizing so far, we find that $g_j|\psi\rangle = |\psi\rangle$ if and only if

$$S(\vec{b}_j)\,|x, Wx\rangle = \mathrm{i}^{l_j(x)}\,F_j(f)\,|x, Wx\rangle. \tag{2.25}$$

Since, by assumption, we have

$$F_j(f) \in \Gamma,$$

we can find a vector $\vec{b}'_j \in \mathbb{Z}_2^{n-t}$, such that

$$\prod_{1 \leq k \leq n-t} \gamma_k^{b'_{jk}} = F_j(f).$$

This in turn means that (2.25) is equivalent to

$$S(\vec{b}_j)\,|x, Wx\rangle = \mathrm{i}^{l_j(x)} \prod_{1 \leq k \leq n-t} \gamma_k(x)^{b'_{jk}} |x, Wx\rangle. \tag{2.26}$$

We now claim that a string $\vec{b}_j$ satisfying this condition exists. To see this, first recall (2.22) and the surrounding discussion, which implies that

$$S(\vec{b}'_j)\,|Wx\rangle = \mathrm{i}^{l'_j(x)} \prod_{1 \leq k \leq n-t} \gamma_k(x)^{b'_{jk}} |Wx\rangle$$

for some linear function l'_j. Second, there exists a string $b''_j \in \{0,1,2,3\}^t$ such that $S(\vec{b}''_j)|x\rangle = i^{l_j(x) - l'_j(x)}|x\rangle$. This shows that

$$S(\vec{b}_j) \coloneqq S(\vec{b}''_j) \otimes S(\vec{b}'_j)$$

satisfies the condition (2.26). We have shown that the operators g_j $(j = 1, \ldots, n)$ stabilize $|\psi\rangle$.

Finally we show that $|\psi\rangle$ is uniquely stabilized by these operators. Let G be the group generated by $\{g_1, \ldots, g_n\}$. Since the $\vec{c}_k$ form a basis of $V^\perp$, any string $v \in \mathbb{Z}_2^n$ belongs to V if and only if $\vec{z}_j^T v = 0$. Thus we have

$$v \in V \quad \Leftrightarrow \quad g_j |v\rangle = |v\rangle \text{ for every } j = t+1, \ldots, n. \tag{2.27}$$

Let G_D be the diagonal subgroup of G. Let U be the set of all $u \in \mathbb{Z}_2^n$ satisfying $D|u\rangle = |u\rangle$ for all $D \in G_D$. According to the argument above lemma 2.9.4, U is the disjoint union of cosets of V; the number of such cosets is the dimension of the code stabilized by G. We show that in fact $U = V$, implying that this dimension is one, so that $|\psi\rangle$ is the unique stabilized state. Since each g_j $(j = t+1, \ldots, n)$ belongs to G_D, every $u \in U$ must satisfy $g_j|u\rangle = |u\rangle$ for all $j = t+1, \ldots, n$. With (2.27) this shows that $U \subseteq V$. Furthermore, since $D|\psi\rangle = |\psi\rangle$ for every $D \in G_D$ and since $|\psi\rangle$ has the form

$$|\psi\rangle = \sum_{v \in V} \alpha_v |v\rangle$$

for some coefficients α_v, it follows from that every $v \in V$ satisfies $D|v\rangle = |v\rangle$ for all $D \in G'_D$. This shows that $V \subseteq U$ and thus $V = U$. □

Remark 2.9.8. *Since F_h are linear transformations from V_1 to V_2, and Γ is a linear subspace, the condition $F_h(f) \in \Gamma$ can be written as linear equations.*

Note that there will be multiple ways to write down the same XS-stabilizer state. Let us consider the following example with five qubits:

$$|\psi\rangle = \sum_x f(x) \, |x_1, x_2, x_3, x_1 \oplus x_2, x_2 \oplus x_3\rangle.$$

Equally, we can set $(x'_1, x'_2, x'_3) = (x_1, x_1 \oplus x_2, x_2 \oplus x_3)$, and the state becomes

$$|\psi\rangle = \sum_{x'} f'(x') \, |x'_1, x'_1 \oplus x'_2, x'_1 \oplus x'_2 \oplus x'_3, x'_2, x'_3\rangle.$$

We can define Γ' in the same way as we defined Γ. We will show that if $F_{x_j} f(x) \in \Gamma$, then $F_{x'_j} f'(x') \in \Gamma'$.

More formally, we consider the state $|\psi\rangle = \sum_x f(x)|x, Wx\rangle$. For an invertible matrix R over $\mathbb{Z}_2$ and $x' = R^{-1}x$, we have

$$\begin{aligned} |\psi\rangle &= \sum_x f(x) \, |RR^{-1}x, WRR^{-1}x\rangle \\ &= \sum_{x'} f(Rx') \, |Rx', WRx'\rangle \\ &\equiv \sum_{x'} f'(x') \, |W'x'\rangle. \end{aligned}$$

In the above equation, we can change the summation from over x to x' because R is invertible. Then we define Γ' from W' in the same way as (2.21) and (2.23). We have the following theorem

Theorem 2.9.9. *Let $|\psi\rangle$ be an XS-stabilizer state in the form given in theorem 2.9.6. Then f satisfies the condition* (2.24)*. What is more, for any invertible matrix R over $\mathbb{Z}_2$, the function f' defined by $f'(x') \coloneqq f(Rx')$ also satisfies*

$$F_{x'_j} f' \in \Gamma'.$$

Proof. First, we note that we have slightly abused the notation here, since f' is in general not a function in V_1 (which is defined in section 2.9.4). However, we can simply ignore the terms

$\alpha^{l(x')}(-1)^{q(x')}$ in f', again by the reasoning in section 2.9.4. After the transformation $x' = R^{-1}x$, the state can be written as

$$|\psi\rangle = \sum_{x'} f'(x')\,|Rx', WRx'\rangle.$$

Note that if we apply $g_1' = \prod_{j=1}^{t} g_j^{R_{j1}}$ on the component $f'(x')\,|Rx', WRx'\rangle$, we would have

$$g_1'\, f'(x')\,|Rx', WRx'\rangle = f'(x' + \vec{e}_1)\,|R(x' + \vec{e}_1), WR(x' + \vec{e}_1)\rangle,$$

where $\vec{e}_1$ is the first canonical basis vector. By the same reasoning we used in the proof of theorem 2.9.7, we know that

$$\frac{f'(x' + \vec{e}_1)}{f'(x')} = \mathrm{i}^{l(x')}\, F_{x_1'}[f'](x'),$$

while at the same time

$$g_1'\,|Rx', WRx'\rangle = \mathrm{i}^{l'(x')}\, h(x)\,|R(x' + \vec{e}_1), WR(x' + \vec{e}_1)\rangle,$$

where $h(x) \in \Gamma'$. Thus $F_{x_1'}[f'] \in \Gamma'$. Similarly we can show $F_{x_j'}[f'] \in \Gamma'$. □

To illustrate how theorem 2.9.7 works, we give two examples here, which demonstrate extreme cases. First consider the state

$$|\psi\rangle = \sum_{x} f(x)\,|x_1, x_2, \ldots, x_n\rangle.$$

By definition, Γ is a trivial vector space. It is then straightforward to check that $f(x)$ has to be of the form $f(x) = i^{l(x)}(-1)^{q(x)}$, and thus $|\psi\rangle$ is a Pauli stabilizer state. On the other hand, consider the state

$$|\psi\rangle = \sum_{x} f(x)\,|x_1, \ldots, x_t\rangle \bigotimes_{j<k\leq t} |x_j \oplus x_k\rangle.$$

It is easy to check Γ is the full vector space V_2. Thus the condition (2.24) becomes trivial, which means $f(x)$ can be an arbitrary function in $\mathcal{F}$.

2.10 Entanglement

2.10.1 Bipartite entanglement

In this section we study the bipartite entanglement in XS-stabilizer states. We consider an n-qubit XS-stabilizer state $|\psi\rangle$ (with regular stabilizer group) in the form given in theorem 2.9.6. Thus we have

$$|\psi\rangle = \sum_{x\in\mathbb{Z}_2^t} f(x)\,|x, Wx\rangle = \sum_{x} f(x)\,|W'x\rangle$$

where we have denoted

$$W' \coloneqq \begin{bmatrix} \mathbb{1}_t \\ W \end{bmatrix}$$

which is an $n \times t$ matrix. The function f belongs to $\mathcal{F}$ and can be evaluated efficiently owing to theorem 2.9.6.

Let (A, B) be a bipartition of the qubits. For convenience we call the two parties Alice and Bob. We will show that there exists a diagonal operation $D_A \otimes D_B$ mapping the state $|\psi\rangle$ to a Pauli stabilizer state.

There exist permutation matrices P_A and P_B acting in A and B, respectively, and an invertible matrix R of dimension t, such that

$$\begin{bmatrix} P_A & \\ & P_B \end{bmatrix} W'R^{-1} = \left[\begin{array}{cc} \mathbb{1}_k & 0 \\ M & 0 \\ \hline 0 & \mathbb{1}_{t-k} \\ C_1 & C_2 \end{array}\right]$$

for some k and some M, C_1 and C_2 (and where, in the r.h.s, the upper block refers to the qubits in A and the lower block refers to the qubits in B). We carry out a change of variables $x \mapsto x' = Rx$. Furthermore we denote $x' = (u, v)$ where u denotes the first k bits of x'. Also, we set $f'(x') \coloneqq f(x)$. We can then write the state in the form

$$|\psi\rangle = \sum_{x'=(u,v)} f'(x') \, |u, Mu\rangle_A \otimes |v, C_1 u + C_2 v\rangle_B,$$

where u ranges over $\mathbb{Z}_2^k$ and v ranges over $\mathbb{Z}_2^{t-k}$. Let r denote the rank of C_1. Notice that there exists a full-rank matrix D_1 (its rank being r) and an invertible matrix T such that $C_1 = [D_1 \mid 0]T$ where 0 is the zero matrix of appropriate dimensions. This means there is a further change of variables from u to $w = Tu$ such that only the first r bits of w appear on Bob's side. In a more explicit way, we can rewrite the computational basis states appearing on Bob's side as

$$|v, D_1(w_1, \ldots, w_r) + C_2 v\rangle. \tag{2.28}$$

And it is easy to see that the form of the state on Alice's side still does not involve v, since it can be written as:

$$|Tw, MTw\rangle.$$

We write $x'' = (w, v)$ and $f''(x'') \coloneqq f'(x')$. Note that the function f'' is related to the function f by a linear change of variables. Hence, owing to theorem 2.9.9, f'' must also satisfy condition (2.24). We will use this condition to gain insight in the form of f''. We will only consider the non-trivial part $h(x'')$ of $f''(x'')$, as defined in (2.18). The reason we only focus on h is because we can obtain all linear terms $\alpha^{l(x'')}$ by acting locally in each party, and we can simply ignore $(-1)^{q(x'')}$ because these terms are all allowed for a Pauli stabilizer state (recall that our goal is to map $|\psi\rangle$ to a Pauli stabilizer state by means of an operation $D_A \otimes D_B$). We claim h only depends on w_1, , w_r and v. We prove this fact by contradiction. For example, assume that h contains a term $(-1)^{w_{r+1} v_a v_b}$. Then by equations (2.19) and (2.20), we know

$$(F_a \circ h)(w, v) = (-1)^{w_{r+1} v_b + q(w,v)},$$

where $q(w, v)$ is a quadratic function that does not contain the term $w_{r+1} v_b$. By theorem 2.9.7, we know that

$$F_a(h) \in \Gamma.$$

However, by observing (2.28), we notice that none of the functions γ_j (which are defined in (2.21)) can contain $(-1)^{w_{r+1} v_b}$. This implies that

$$F_a(f'') \notin \Gamma,$$

which leads to the contradiction. Similarly we can show that there are no $i^{w_j v_b}$ terms in h for $j \geq r + 1$.

We have shown that the function h only depends on w_1, , w_r and v. This implies that, by acting with a suitable diagonal unitary operation within Bob's side, we can remove the corresponding phase in the state $|\psi\rangle$; the resulting state is a Pauli stabilizer state. To see how these phases can be removed locally, we argue as follows. The standard basis kets on Bob's side have the form (2.28) with D_1 full rank. Thus there exists a (full rank) matrix E such that $ED_1 w = w$ for every $w = (w_1, \ldots, w_r)$. Let U_1 be the unitary operation which implements E

$$U_1 \colon |v, D_1(w_1, \ldots, w_r) + C_2 v\rangle \mapsto |v, (w_1, \ldots, w_r) + EC_2 v\rangle$$

After applying U_1, we apply the operation U_2 defined by

$$U_2 \colon |v, (w_1, \ldots, w_r) + EC_2 v\rangle \mapsto |v, (w_1, \ldots, w_r)\rangle$$

Note that both U_1 and U_2 can be realized as circuits of CNOT gates. Then we apply the diagonal operation D defined by

$$D \colon |v, (w_1, \ldots, w_r)\rangle \mapsto h(w_1, \ldots, w_r, v)^{-1} |v, (w_1, \ldots, w_r)\rangle.$$

Since h can be evaluated efficiently (this follows from the fact that f can be evaluated efficiently), the operation D can be implemented efficiently. Finally, we apply $U_2^\dagger$ followed by $U_1^\dagger$, yielding

$$h(w_1, \ldots, w_r, v) \, |D_1(w_1, \ldots, w_r) + C_2 v\rangle.$$

This total procedure $D_B \coloneqq U_1^\dagger U_2^\dagger D U_2 U_1$ thus allows us to multiply each ket (2.28) with the inverse of $h(w_1, \ldots, w_r, v)$, thereby "canceling out" the function h. Note that D_B is a diagonal operation, since D is diagonal and since $U_2 U_1$ is a permutation of the standard basis.

We have shown:

Theorem 2.10.1. *Let $|\psi\rangle$ be an XS-stabilizer state and (A, B) a bipartition of its qubits. Then there exists a Pauli stabilizer state $|\phi_{A,B}\rangle$ and diagonal operators D_A and D_B such that*

$$|\psi\rangle = D_A \otimes D_B \, |\phi_{A,B}\rangle.$$

A description of $|\phi_{A,B}\rangle$ can be computed efficiently.

Corollary 2.10.2. *The von Neumann entanglement entropy of $|\psi\rangle$ w.r.t. (A, B) can be computed efficiently.*

Corollary 2.10.3. *Let ρ_A be the reduced density operator of $|\psi\rangle$ for the qubits in A. Then ρ_A is proportional to a projector i.e. all nonzero eigenvalues of ρ_A coincide.*

The first corollary holds since the entanglement entropy of Pauli stabilizer states can be computed efficiently [FCY+04]. The second corollary holds since reduced density operators of Pauli stabilizer states are proportional to projectors [HDE+05].

The state $|\phi_{A,B}\rangle$ generally depends on the bipartition (A, B). It would be interesting to understand whether Theorem 2.10.1 can be made independent of that:

Problem. *For every XS-stabilizer state $|\psi\rangle$, does there exist a single Pauli stabilizer state $|\phi\rangle$ such that for* every *bipartition (A, B) we have*

$$|\psi\rangle = U_A \otimes U_B \, |\phi\rangle$$

for some local unitaries U_A and U_B?

2.10.2 LU-inequivalence of XS- and Pauli stabilizer states

Here we show that the XS-stabilizer state (2.2) is not LU-equivalent to any Pauli stabilizer state, i.e. there does not exist any Pauli stabilizer state $|\phi\rangle$ satisfying $|\psi\rangle = U \, |\phi\rangle$ for any $U \coloneqq U_1 \otimes \cdots \otimes U_6 \in \mathrm{U}(2)^{\otimes 6}$. This result demonstrates that there exist XS-stabilizer states whose multipartite entanglement (w.r.t LU-equivalence) is genuinely different from that of any Pauli stabilizer state.

In order to prove the claim, we consider the classification of 6-qubit stabilizer (graph) states under LU-equivalence as given in Fig. 4 and Table II of Ref. [HEB04]. In the latter figure, 11 distinct LU-equivalence classes are shown to exist for fully entangled 6-qubit stabilizer states; the classes are labeled from 9 to 19. A representative of each class is given in Fig 4. Furthermore each class is uniquely characterized by its list of Schmidt ranks, i.e. the Schmidt ranks for all possible bipartitions of the system. In table II it is shown that, for any of the classes 9–17, there is at least one bipartition of the form (two qubits – rest) for which the state is not maximally entangled. This shows that the XS-stabilizer state $|\psi\rangle$ cannot be LU-equivalent to any of the states in the classes 9–17, since $|\psi\rangle$ is maximally entangled for all such bipartitions. Furthermore, for class 19, the entanglement is maximal for all bipartitions of the form (3 qubits – rest); since this is not the case for $|\psi\rangle$, the latter cannot be LU-equivalent to any state in class 19. This leaves class 18. Consider the state

$$|\phi\rangle = \sum_{x_j=0}^{1} |x_1, x_2, x_3, x_1 \oplus x_2, x_2 \oplus x_3, x_3 \oplus x_1\rangle$$

which is a Pauli stabilizer state. By direct computation of all Schmidt ranks, one verifies that this state belongs to class 18. We prove by contradiction that $|\psi\rangle$ is not LU-equivalent to $|\phi\rangle$. First, it is straightforward to show that for both $|\phi\rangle$ and $|\psi\rangle$, the 3-qubit reduced density matrix of the 1st, 2nd and 4th qubits is

$$\begin{aligned}\rho_{124} &= \frac{1}{4}(|000\rangle\langle 000| + |011\rangle\langle 011| + |101\rangle\langle 101| + |110\rangle\langle 110|) \\ &= \frac{1}{8}(I + Z_1 \otimes Z_2 \otimes Z_4).\end{aligned}$$

If there is a local unitary transformation U from $|\phi\rangle$ to $|\psi\rangle$, then $U_1 \otimes U_2 \otimes U_4$ must leave ρ_{124} unchanged. This implies that $U_1 \otimes U_2 \otimes U_4$ must leave $Z_1 \otimes Z_2 \otimes Z_4$ unchanged, so that $U_1 Z_1 U_1^\dagger \propto Z_1$ and similarly for U_2 and U_4. This implies that U_1, U_2 and U_4 must have the form D_j or $D_j X$ for some diagonal matrix D_j. Analogously, we can show that the same holds for all other j. Thus $U = D\,X(\vec{a})$ for some diagonal operator $D \coloneqq D_1 \otimes \cdots \otimes D_6$ and some $\vec{a} \in \mathbb{Z}_2^6$. Note that $|\psi\rangle$ and $|\phi\rangle$ have the form

$$|\phi\rangle = \sum_{v \in V} |v\rangle \qquad \text{and} \qquad |\psi\rangle = \sum_{v \in V} \alpha_v |v\rangle,$$

for some linear subspace $V \subseteq \mathbb{Z}_2^6$ and real coefficients α_v. Then

$$D\,X(\vec{a})\,|\phi\rangle = \sum_{v \in V} \beta_v |v + \vec{a}\rangle$$

for some coefficients β_v. Thus $U|\phi\rangle = |\psi\rangle$ implies that $V = V + \vec{a}$. This shows that $\vec{a} \in V$. But then $X(\vec{a})|\phi\rangle = |\phi\rangle$. The identity $U|\phi\rangle = |\psi\rangle$ thus implies that $D|\phi\rangle = |\psi\rangle$. It is straightforward to verify that this cannot be true. We have thus shown that $|\psi\rangle$ and $|\phi\rangle$ are not LU-equivalent. In conclusion, $|\psi\rangle$ does not belong to any LU-equivalence class of Pauli stabilizer states.

2.11 Efficient algorithms

In this section we will give a list of problems that can be solved with efficient classical algorithms for regular XS-stabilizer states (codes). We consider an arbitrary n-qubit regular XS-stabilizer stabilizer code $\mathcal{H}_G$ specified in terms of a generating set of m stabilizers in the standard form given in Corollary 2.8.5. Then the following holds:

1. The **degeneracy** d of the code can be computed in $\text{poly}(n, m)$ time (recall section 2.9.2).

2. An efficient algorithm exists to **determine** d **basis states** $|\psi_1\rangle$, , $|\psi_d\rangle$, each of which is an XS-stabilizer state with regular stabilizer group and each state having the form

$$|\psi_i\rangle = \sum_{x \in \mathbb{Z}_2^t} f_i(x)\,|x, Wx + \vec{\mu}_i\rangle \qquad \text{with } f_i \in \mathcal{F}.$$

   The matrix W (which is the same for all $|\psi_i\rangle$) can be computed in $\text{poly}(n, m)$ time. The list $\{\vec{\mu}_1, , \vec{\mu}_d\}$ can be computed in $\text{poly}(n, m, d)$ time. Given a specific $\vec{\mu}_i$, a complete generating set of stabilizer operators having $|\psi_i\rangle$ as unique stabilized state can be computed in $\text{poly}(n, m)$ time. Furthermore, given $\vec{\mu}_i$, the function $x \mapsto f_i(x)$ can be computed in $\text{poly}(n, m)$ time as well. See section 2.9.2.

3. The **logical operators** of $\mathcal{H}_G$ can be computed in $\text{poly}(n, m, d)$ time. See section 2.9.3.

4. The **commuting Hamiltonian** described in section 2.5 can be computed in $\text{poly}(n, m)$ time.

On input of $\vec{\mu}_i$ the following holds in addition:

5. The von Neumann **entanglement entropy** of any $|\psi_i\rangle$ with regular stabilizer group can be computed, for any bipartition, in $\mathrm{poly}(n, m)$ time. This claim holds since we have shown in section 2.10 how to efficiently compute the description of a Pauli stabilizer state with the same entanglement as $|\psi_i\rangle$; furthermore an efficient algorithm to compute the von Neumann entanglement entropy of Pauli stabilizer states is known [FCY+04].

6. A $\mathrm{poly}(n)$ size **quantum circuit** to generate any $|\psi_i\rangle$ can be computed for in $\mathrm{poly}(n, m)$ time. This circuit can always be chosen to be a Clifford circuit followed by a circuit composed of the diagonal gates $\mathrm{CC}Z$ (controlled-$\mathrm{C}Z$), $\mathrm{C}S := \mathrm{diag}(1, 1, 1, \mathrm{i})$ (controlled-S) and T. To see this, we recall theorem 2.9.6. This implies that the state $|\psi_i\rangle$ can be prepared as follows:

   - Using a Clifford circuit $\mathcal{C}_1$, prepare the state $\sum |x, Wx + \vec{\mu}_i\rangle$. In fact, this can be done using a circuit composed of Hadamard, X and CNOT gates.
   - Since the function f_i belongs to the class $\mathcal{F}$, it has the form

   $$\alpha^{l(x)}\, \mathrm{i}^{q(x)}\, (-1)^{c(x)}.$$

   Note that the we have the following gate actions on the standard basis:

   $$\begin{aligned} T\colon |x\rangle &\mapsto \alpha^x |x\rangle, \\ S\colon |x\rangle &\mapsto \mathrm{i}^x |x\rangle, \\ \mathrm{C}S\colon |x, y\rangle &\mapsto \mathrm{i}^{xy} |x\rangle, \\ \mathrm{C}Z\colon |x, y\rangle &\mapsto (-1)^{xy} |x, y\rangle, \\ \mathrm{CC}Z\colon |x, y, z\rangle &\mapsto (-1)^{xyz} |x, y, z\rangle. \end{aligned}$$

   Therefore, the phase $f_i(x)$ can be generated by first applying a suitable circuit $\mathcal{C}_2$ of Clifford gates $\mathrm{C}Z$ and S to generate the quadratic part of $c(x)$ and the linear part of $q(x)$, and by subsequently applying a suitable circuit $\mathcal{U}$ composed of the (non-Clifford) gates T, $\mathrm{C}S$ and $\mathrm{CC}Z$ to generate $l(x)$, the quadratic part of $q(x)$ and the cubic part of $c(x)$, respectively. Since the function f_i can be computed efficiently, the descriptions of $\mathcal{C}_2$ and $\mathcal{U}$ can be computed efficiently. The overall circuit is $\mathcal{U}\mathcal{C}_2\mathcal{C}_1$.

7. Given any $|\psi_i\rangle$ and Pauli operator P, we can **compute the expectation value** $\langle\psi_i|P|\psi_i\rangle$ in $\mathrm{poly}(n, m)$ time. This implies in particular that the expectation of any local observable (i.e. an observable acting on a subset of qubits of constant size) can be computed efficiently as well, since every such observable can be written as a sum of $\mathrm{poly}(n)$ Pauli observables. To see that $\langle\psi_i|P|\psi_i\rangle$ can be computed efficiently, recall from point 6 above that $|\psi_i\rangle$ can be decomposed as $|\psi_i\rangle = \mathcal{U}|\psi_i'\rangle$ where $\mathcal{U}$ is a circuit composed of T, $\mathrm{C}S$ and $\mathrm{CC}Z$, and where $|\psi_i'\rangle = \mathcal{C}_2\mathcal{C}_1|0\rangle$ is a Pauli stabilizer state. Then

   $$\langle\psi_i|P|\psi_i\rangle = \langle\psi_i'|\mathcal{U}^\dagger P \mathcal{U}|\psi_i'\rangle.$$

   Its is easily verified that $\mathcal{U}^\dagger P \mathcal{U} =: \mathcal{C}''$ is a Clifford operation, for every circuit $\mathcal{U}$ composed of T, $\mathrm{C}S$ and $\mathrm{CC}Z$ (for example $TXT \propto S$). Thus we have

   $$\langle\psi_i|P|\psi_i\rangle = \langle\psi_i'|\mathcal{C}''|\psi_i'\rangle.$$

   Recall that $|\psi_i'\rangle = \mathcal{C}_2\mathcal{C}_1|\psi_i\rangle$, we know that

   $$\langle\psi_i|P|\psi_i\rangle = \langle 0|\mathcal{C}'''|0\rangle,$$

   where $\mathcal{C}''' = \mathcal{C}_1^\dagger\mathcal{C}_2^\dagger\mathcal{C}\mathcal{C}_2\mathcal{C}_1$. Note that $\langle 0|\mathcal{C}'''|0\rangle$ is simply the coefficient of the basis $|0\rangle$ in the Pauli stabilizer state $\mathcal{C}'''|0\rangle$, which can be computed efficiently according to [VDN10].

2.12 Non-regular XS-stabilizer groups

Though we have tried to avoid non-regular XS-stabilizer groups due to the computational hardness, there are situations where they appear naturally. For example, let us look at (2.31) through (2.32) in the appendix. They describe a code space $\mathcal{H}_G$ that is equivalent to the ground space of the twisted quantum double model $\mathrm{D}^\omega(\mathbb{Z}_2\times\mathbb{Z}_2\times\mathbb{Z}_2)$ by a local unitary circuit (as defined in [CGW10]). These stabilizer operators have an interesting property: if they are on an infinite lattice or a lattice with open boundary, then they generate a regular XS-stabilizer group. On the other hand, for example, if they are on a torus, the group they generated will not be regular. This is related to the fact that this model has a ground state degeneracy of 22 when it is on a torus, which cannot be the degeneracy of a regular XS-stabilizer code. It is also known that this twisted quantum double model support non-Abelian anyons, which has shown to be impossible for Pauli stabilizer codes on 2D.

Given the existence of interesting non-regular XS-stabilizer groups, we want to make a few comments about which results in this chapter still hold for non-regular groups. First, we have the following theorem:

Theorem 2.12.1. *Every XS-stabilizer state has a* regular *XS-stabilizer group which uniquely stabilizes it.*

Proof. Let G be the initial (generally non-regular) stabilizer group of $|\psi\rangle$. We show that G can be replaced with a regular stabilizer group. Let $\{D_1,\dots,D_r\}$ be generators of the diagonal subgroup of G. Extend this set to a generating set of G, say $\mathcal{S}=\{D_1,\dots,D_r,g_1,\dots,g_k\}$. Here each g_j has the form $g_j=i^{s_j}X(\vec{a}_j)S(\vec{b}_j)$. We can assume g_j is non-diagonal, and $\vec{a}_j$ are linearly independent of each other. Since if this is not the case, we can use the procedure in Theorem 2.8.4 to transform g_j to satisfy this condition. Notice that g_j are all monomial unitary matrix. It follows that the permutation group $\mathfrak{P}$ associated with G is generated by the operators $X(\vec{a}_j)$. Let $V\subseteq\mathbb{Z}_2^n$ (where n denotes the number of qubits) be the linear span of the $\vec{a}_j$. Then $\mathfrak{P}=\{X(v)\mid v\in V\}$. Furthermore, the orbit of a computational basis state $|x\rangle$ is the coset of V containing x i.e. $\mathcal{O}_x=x+V$.

Consider the set V_D of those n-bit strings z satisfying $D|z\rangle=|z\rangle$ for all $D\in G_D$. Furthermore, recall from the proof of theorem 2.7.2 that $G_x=G_D$ for every x. Applying theorem 2.6.1(b) and using that the dimension of the space stabilized by G is 1 (since $|\psi\rangle$ is an XS-stabilizer state) we conclude that $V_D=x+V$ for some x, and that $|\psi\rangle$ must have the form

$$|\psi\rangle=\sum_{v\in V}f(v)\,|v+x\rangle.$$

Now define Z-type operators h_k of the form $h_k(-1)^{s_k}Z(\vec{b}_k)$ $(k=1,\dots,q)$, where s_k and $\vec{b}_k$ are chosen such that V_D coincides with the set of all z satisfying $\vec{b}_k^T z=s_k$ for all k. This means $h_k|z\rangle=|z\rangle$ for all $z\in x+V$ and in turn $h_k|\psi\rangle=|\psi\rangle$. It follows that $|\psi\rangle$ is stabilized by $\mathcal{S}':=\{h_1,\dots,h_q,g_1,\dots,g_k\}$. Finally, by lemma 2.8.3, we know the group G' is regular, and by the argument in lemma 2.9.5, we know $|\psi\rangle$ is uniquely stabilized by G'. □

Now consider the procedure in lemma 2.9.5. It is easy to see even if the group G is non-regular, as long as we have a $\vec{\lambda}_j\in V_D$, we can still find the $|\psi_j\rangle$ corresponds to $\vec{\lambda}_j$. By theorem 2.6.1, there is a set of $\{\vec{\lambda}_j\}$ such that the corresponding $|\psi_j\rangle$ form a basis for the space stabilized by G. Again by the procedure in lemma 2.9.5, we know G can be expanded to uniquely stabilize each $|\psi_j\rangle$. Thus by theorem 2.12.1 we know $|\psi_j\rangle$ is a regular XS-stabilizer state. This means although it is (computationally) hard to find $\vec{\lambda}_j$, the basis $|\psi_j\rangle$ for the code space still satisfies all the properties we proved, including the form of the phases $f(x)$ and the bipartite entanglement. The construction of commuting Hamiltonian also does not require the stabilizer group to be regular.

On the other hand, for non-regular stabilizer groups, there is no general formula for the degeneracy. We also cannot find logical operators that have a similar form as the ones in section 2.9.3, since the degeneracy of $\mathcal{H}_G$ is not necessarily 2^k.

2.13 Open questions

In this section we will summarize a few interesting questions about the XS-stabilizer formalism, some of which have already been mentioned in the text.

The group structure While the tractability of XS-stabilizer states $|\psi_j\rangle$ is closely related to the fact that each XS-stabilizer group G is a rather particular finite group, the properties of $|\psi_j\rangle$ are not. It would be interesting to establish some direct link between the group G and the states $|\psi_j\rangle$ (e.g. a relation between the reduced density matrix ρ and G).

Properties of entanglement As we mentioned in section 2.10, it is not known whether for any XS-stabilizer state $|\psi\rangle$ there exists a single Pauli stabilizer state $|\varphi\rangle$ that has the same von Neumann entropy across *all* bipartitions. It would also be interesting to know to what extent the inequalities described in [LMRW13] hold for XS-stabilizer states.

Logical operators and transversal gates We have shown how to construct $\bar{Z}_j$ and $\bar{X}_j$ operators in section 2.9.3. The $\bar{Z}_j$ are transversal gates by definition. While we showed that the $\bar{X}_j$ operators include X, S, and CZ in general, it is possible that for many codes the $\bar{X}_j$ only contain X and S. In particular, S and CZ are interchangeable in some cases. For example, consider the state

$$|\psi\rangle = \sum_{x_1,x_2} |x_1, x_2, x_1 \oplus x_2\rangle.$$

It is easy to check that

$$\mathrm{C}Z_{12}\,|\psi\rangle = S_1^3 S_2^3 S_3\,|\psi\rangle.$$

Thus it would be interesting to know when a certain XS-stabilizer code has transversal $\bar{X}_j$ operators, and possibly some other transversal gates.

Quantum phases Understanding topological phases is an extremely important but also very hard task. Compared to general local Hamiltonians, the Hamiltonians generated by local Pauli stabilizer codes are much easier to analyze. Thus the Pauli stabilizer formalism has proved a gateway both to studying the behaviour of topological phases and to constructing new models. It is then natural to ask whether we can classify all topological phases described by XS-stabilizer codes or whether we can construct new models in 2D and 3D.

Non-regular XS-stabilizer As we have shown in this chapter, it is in general computationally hard to study the states stabilized by non-regular XS-stabilizer groups. Restricting to regular groups is sufficient to circumvent this problem, but not necessary. It is thus desirable to find the necessary conditions under which the XS-stabilizer problem will become efficient. For example, it is not clear whether the XS-stabilizer problem is still hard if the number of S-type operators in the generators of the diagonal subgroup is constant.

Appendix

2.A Twisted quantum double models

We study the twisted quantum double models $\mathrm{D}^\omega(\mathbb{Z}_2^n)$ with the groups $\mathbb{Z}_2^n$ and twists $\omega \in H^3\big(\mathbb{Z}_2^n, U(1)\big)$ on a triangular lattice. Although every such group is Abelian, for certain n and ω the twisted quantum double model $\mathrm{D}^\omega(\mathbb{Z}_2^n)$ will harbour *non*-Abelian anyons as excitations.

Without loss of generality we choose the branching structure shown in Figure 2.A.1 for the triangular lattice. Each lattice edge i carries a Hilbert space with basis $\{|x_i\rangle \mid x_i \in \mathbb{Z}_2^n\}$. By abuse of notation $|x_i\rangle$ is either the state of an actual qubit if $n = 1$ or the state of a qudit if $n > 1$. In the latter case we write elements $t = (t_1, \ldots, t_n) \in \mathbb{Z}_2^n$ as binary strings over the alphabet $\{0, 1\}$ and accordingly expand the qudit state $|x_i\rangle = |x_{i,1}, \ldots, x_{i,n}\rangle$ in terms of qubit states $|x_{i,\sigma}\rangle$ where i denotes the position on the lattice and σ the "layer". Furthermore we write group multiplication in $\mathbb{Z}_2^n$ additively.

The Hamiltonian is given as a sum of commuting projectors:

$$H = -\sum_s A^\omega(s) - \sum_p B(p).$$

Each operator $B(p)$ is associated with a triangle p of the lattice and reads

$$B(p) = \delta(x_i + x_j + x_k)\,|x_i, x_j, x_k\rangle\langle x_i, x_j, x_k|$$

where i, j and k denote the edges of p. It enforces a flat connection on the triangle p in the ground state subspace. The operator $A^\omega(s)$ associated with a vertex s is defined by

$$A^\omega(s) = \frac{1}{2^n} \sum_{t \in \mathbb{Z}_2^n} A_t^\omega(s).$$

If s is the central vertex of Figure 2.A.1 the individual terms are given by

$$A_t^\omega(s) = \sum_{x_i \in \mathbb{Z}_2^n} f_t^\omega(x)\,|x_1 + t, \ldots, x_6 + t\rangle\langle x_1, \ldots, x_6| \otimes |x_7, \ldots, x_{12}\rangle\langle x_7, \ldots, x_{12}|$$

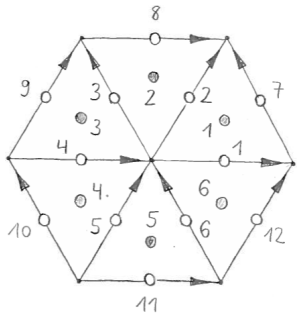


Figure 2.A.1: A branching structure on the triangular lattice. White circles denote qudits $|x_i\rangle$, grey circles denote possible ancilla qudits $|y_p\rangle$.

with the phases[2]

$$f_t^\omega(x) = \frac{\omega(t, x_4, x_{10})\omega(x_3 + t, t, x_4)\omega(x_8, x_3 + t, t)}{\omega(t, x_6, x_{11})\omega(x_1 + t, t, x_6)\omega(x_7, x_1 + t, t)} = \pm 1.$$

Note that each ω couples two distinct qudit variables x_i and x_j which always belong to some triangle. Also, these phases enjoy the property

$$f_{tt'}^\omega(x) = f_t^\omega(x_1 + t', \ldots, x_6 + t', x_7, \ldots, x_{12})\, f_{t'}^\omega(x)$$

which implies $A_t^\omega(s)\, A_{t'}^\omega(s) = A_{tt'}^\omega(s)$. The phases arising from a product of 3-cocycles ω and ω' factorize as

$$f_t^{\omega\omega'}(x) = f_t^\omega(x)\, f_t^{\omega'}(x) \tag{2.29}$$

because $(\omega\omega')(a, b, c) = \omega(a, b, c)\omega'(a, b, c)$ is the multiplication of 3-cocycles.

Since $x_i + x_j + x_k = 0$ in $\mathbb{Z}_2^n$ is equivalent to $x_{i,\sigma} \oplus x_{j,\sigma} \oplus x_{k,\sigma} = 0$ for all layers σ we can describe the common $+1$ eigenspace of all triangle operators $B(p)$ as the subspace stabilized by

$$Z_{i,\sigma} Z_{j,\sigma} Z_{k,\sigma}$$

for all edges i, j and k forming a triangle and all layers σ. This subspace is exactly the gauge-invariant subspace of Section 2.5. In order to describe the ground state subspace of the complete Hamiltonian it suffices to add the stabilizers $A_t^\omega(s)$ for all vertices s and all generators t of $\mathbb{Z}_2^n$. While a vertex operator $A_t^\omega(s)$ itself may not belong to the Pauli-S group we will find an equivalent stabilizer $\mathcal{A}_t^\omega(s) \in \mathcal{P}^S$ which coincides with $A_t^\omega(s)$ on the gauge-invariant subspace.

2.A.1 $\mathbb{Z}_2$

The third cohomology group $H^3\big(\mathbb{Z}_2, U(1)\big) \simeq \mathbb{Z}_2$ is generated by

$$\omega(a, b, c) = (-1)^{abc}.$$

It is well known that all twisted quantum double models for the group $\mathbb{Z}_2$ support Abelian anyons only.

For this ω we obtain the phases

$$f_1^\omega(x) = (-1)^{x_1x_6 + x_1x_7 + x_3x_4 + x_3x_8 + x_4x_{10} + x_6x_{11}}\, (-1)^{x_4 + x_6 + x_7 + x_8}.$$

The phases with linear exponent can always be generated by applying Z. On the gauge-invariant subspace we can also generate all quadratic phases $(-1)^{x_ix_j}$ by applying suitable powers of S because the edges i and j always belong to some triangle. Denoting the third edge of the triangle by k we can indeed get $(-1)^{x_ix_j}$ from $S_i^3 S_j^3 S_k$ because $\mathrm{i}^{x_k} = \mathrm{i}^{x_i \oplus x_j} = \mathrm{i}^{x_i + x_j}\, (-1)^{x_ix_j}$ holds by Lemma 2.9.1. Hence the operator

$$X_1 \cdots X_6 Z_1 Z_2 Z_3 Z_5 S_7 S_8 S_9 S_{10}^\dagger S_{11}^\dagger S_{12}$$

coincides with $A_1^\omega(s)$ on the gauge-invariant subspace. We can recover a more symmetric expression by multiplying with Z-type stabilizers and obtain

$$\mathcal{A}_1^\omega(s) = X_1 \cdots X_6 Z_1 \cdots Z_6 S_7 \cdots S_{12}.$$

This is the same stabilizer as the one in the doubled semion model [LW05] up to conjugation by $S_1 \cdots S_6$. The subspace stabilized by all $\mathcal{A}_1^\omega(s)$ and Z-type stabilizers is thus equivalent to the ground state subspace of the doubled semion model up to local unitaries.

Now for a given lattice, we can define g_j with $j \leq t$ to be $\mathcal{A}_1^\omega(s)$ on each vertex s, and the rest of g_j to be the operator $B(p)$. One thing needs to be taken care of is when the lattice periodic

[2] Note that these phases do not explicitly depend on the values x_2, x_5, x_9 and x_{12}. This may change if one fixes a different branching structure on the triangular lattice.

boundary condition (e.g. torus), g_j will no long be in the standard form as we defined in (2.10), since we have

$$\prod_{j\leq t} X(\vec{a}_j) = I.$$

To check that in this case the stabilizer group is still regular, we only need to check the product

$$\prod_{j\leq t} g_j \tag{2.30}$$

is a Z-type operator. We notice that by lemma (2.4.1), we can exchange X and S in the product (2.30) with the only price being introducing new Z operators into the product. Thus as long as for each j, the S_j operator (S on the jth qubit) appears even number of times in the product, we know the product will be a Z-type operator. And this can be readily checked. With a straightforward but more involved calculation, we can show that the product (2.30) is satisfied by the gauge-invariant subspace, or in other words, the product can be generated by $\{B(p)\}$.

2.A.2 $\mathbb{Z}_2 \times \mathbb{Z}_2$

The third cohomology group $H^3\big(\mathbb{Z}_2 \times \mathbb{Z}_2, U(1)\big) \simeq \mathbb{Z}_2^3$ is generated by

$$\begin{aligned}
\omega_1(a,b,c) &= (-1)^{a_1 b_1 c_1},\\
\omega_2(a,b,c) &= (-1)^{a_2 b_2 c_2},\\
\omega_3(a,b,c) &= (-1)^{a_1 b_2 c_2}.
\end{aligned}$$

It is known that all twisted quantum double models for the group $\mathbb{Z}_2 \times \mathbb{Z}_2$ support Abelian anyons only.

It is not difficult to see that the 3-cocycles ω_1 and ω_2 do not lead to anything qualitatively new compared to the case $\mathbb{Z}_2$.[3]

The 3-cocycle ω_3 is much more interesting. We obtain the phases

$$\begin{aligned}
f^{\omega_3}_{(1,0)}(x) &= (-1)^{x_{4,2}x_{10,2}+x_{6,2}x_{11,2}},\\
f^{\omega_3}_{(0,1)}(x) &= (-1)^{x_{1,1}x_{6,2}+x_{3,1}x_{4,2}+x_{1,2}x_{7,1}+x_{3,2}x_{8,1}}\,(-1)^{x_{7,1}+x_{8,1}}.
\end{aligned}$$

Clearly, the phases associated with $(1,0)$ are confined to layer 2 and we can apply the methods of 2.A.1. This results in

$$\mathcal{A}^{\omega_3}_{(1,0)}(s) = X_{1,1}\cdots X_{6,1}S^3_{4,2}Z_{5,2}S^3_{6,2}S^3_{10,2}S^3_{11,2}.$$

However, the quadratic phases $(-1)^{x_{i,1}x_{j,2}}$ arising from $(0,1)$ are of a different kind. Although all pairs of edges i and j continue to belong to some triangle we can no longer exploit the flat connection since the qubits reside on different layers. Instead we introduce the ancilla qubits

$$|y_p\rangle = |x_{i,1} \oplus x_{j,2}\rangle$$

for $(p,i,j) \in \{(1,7,1),(2,8,3),(3,3,4),(4,4,10),(5,6,11),(6,1,6)\}$ and these may be associated with the triangles of the lattice as shown in Figure 2.A.1. Clearly, the above coupling can be enforced by additional Z-type stabilizers. We will write $\tilde{O}_p$ for an operator O acting on the ancilla qubit y_p in the triangle p. On the gauge-invariant subspace coupled to the ancilla layer we then have

$$\mathcal{A}^{\omega_3}_{(0,1)}(s) = X_{1,2}\cdots X_{6,2}S^3_{1,1}S^3_{1,2}S^3_{3,1}S^3_{3,2}S^3_{4,2}S^3_{6,2}S_{7,1}S_{8,1}\tilde{S}_1\tilde{S}_2\tilde{S}_3\tilde{S}_6.$$

Similar to 2.A.1, we can also compute the additional diagonal operators when we have a lattice with periodic boundary condition. Notice that $\mathcal{A}^{\omega_3}_{(1,0)}(s)$ ($\mathcal{A}^{\omega_3}_{(0,1)}(s)$) commute with each other for any two vertices. It is then straightforward to check the multiplication of all $\mathcal{A}^{\omega_3}_{(1,0)}(s)$ is identity, and $\mathcal{A}^{\omega_3}_{(0,1)}(s)$ can be generated by $B(p)$.

[3]Indeed, for ω_1 the phases $f^{\omega_1}_{(1,0)}$ are confined to layer 1 where we can apply the methods of 2.A.1. In contrast, the other generator $(0,1)$ yields trivial phases only so that $\mathcal{A}^{\omega_1}_{(0,1)} = A^{\omega_1}_{(0,1)}$ is an X-type element confined to layer 2.

2.A.3 $\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2$

The third cohomology group $H^3\big(\mathbb{Z}_2 \times \mathbb{Z}_2 \times \mathbb{Z}_2, U(1)\big) \simeq \mathbb{Z}_2^7$ is generated by

$$\begin{aligned}
\omega_1(a,b,c) &= (-1)^{a_1 b_1 c_1},\\
\omega_2(a,b,c) &= (-1)^{a_2 b_2 c_2},\\
\omega_3(a,b,c) &= (-1)^{a_3 b_3 c_3},\\
\omega_4(a,b,c) &= (-1)^{a_1 b_2 c_2},\\
\omega_5(a,b,c) &= (-1)^{a_1 b_3 c_3},\\
\omega_6(a,b,c) &= (-1)^{a_2 b_3 c_3},\\
\omega_7(a,b,c) &= (-1)^{a_1 b_2 c_3}.
\end{aligned}$$

It turns out that the twisted quantum double models $\mathrm{D}^\omega(\mathbb{Z}_2^3)$ support non-Abelian anyons if and only if the twist ω contains ω_7 [dWP97].

Again, the 3-cocycles ω_1, , ω_6 lead to situations which qualitatively resemble the cases $\mathbb{Z}_2$ and $\mathbb{Z}_2 \times \mathbb{Z}_2$.

Now the 3-cocycle ω_7 leads to truly interesting results. We obtain the phases

$$\begin{aligned}
f^{\omega_7}_{(1,0,0)}(x) &= (-1)^{x_{4,2}x_{10,3}+x_{6,2}x_{11,3}},\\
f^{\omega_7}_{(0,1,0)}(x) &= (-1)^{x_{1,1}x_{6,3}+x_{3,1}x_{4,3}},\\
f^{\omega_7}_{(0,0,1)}(x) &= (-1)^{x_{1,2}x_{7,1}+x_{3,2}x_{8,1}}.
\end{aligned}$$

Let us introduce the ancilla qubits

$$\begin{aligned}
|y_{p,1}\rangle &= |x_{i,1} \oplus x_{j,2}\rangle,\\
|y_{p,2}\rangle &= |x_{i,1} \oplus x_{j,3}\rangle,\\
|y_{p,3}\rangle &= |x_{i,2} \oplus x_{j,3}\rangle
\end{aligned}$$

for positions $(p,i,j) \in \{(1,7,1),(2,8,3),(3,3,4),(4,4,10),(5,6,11),(6,1,6)\}$. This coupling can again be enforced by additional Z-type stabilizers. We can then write

$$\mathcal{A}^{\omega_7}_{(1,0,0)}(s) = X_{1,1}\cdots X_{6,1} S^3_{4,2} S^3_{10,3} S^3_{6,2} S^3_{11,3} \tilde{S}_{4,3} \tilde{S}_{5,3}, \tag{2.31}$$

$$\mathcal{A}^{\omega_7}_{(0,1,0)}(s) = X_{1,2}\cdots X_{6,2} S^3_{1,1} S^3_{3,1} S^3_{4,3} S^3_{6,3} \tilde{S}_{3,2} \tilde{S}_{6,2},$$

$$\mathcal{A}^{\omega_7}_{(0,0,1)}(s) = X_{1,3}\cdots X_{6,3} S^3_{1,2} S^3_{3,2} S^3_{7,1} S^3_{8,1} \tilde{S}_{1,1} \tilde{S}_{2,1}. \tag{2.32}$$

For a given $(j,k,l) \in \{(0,0,1),(0,1,0),(0,0,1)\}$, again $\mathcal{A}^{\omega_7}_{(j,k,l)}(s)$ commute with each other for different s. Thus it is easy to compute the product $\prod_s \mathcal{A}^{\omega_7}_{(j,k,l)}(s)$ for a lattice with periodic boundary condition. However, in this case, the product would be some tensor product that contains S operators. Thus the stabilizer group G for this model on a torus is not a regular XS-stabilizer group, which is different from the previous two models that are based on $\mathbb{Z}_2$ and $\mathbb{Z}_2 \times \mathbb{Z}_2$. However, on a 2D lattice with suitable boundary the stabilizer group G *is* regular and the unique ground state continues to support non-Abelian anyons since these excitations can be created locally.

2.A.4 $\mathbb{Z}_2^n$

In general, the third cohomology group $H^3\big(\mathbb{Z}_2^n, U(1)\big)$ is generated by the following types of generators [dWP97]:

$$\begin{aligned}
\omega_i(a,b,c) &= (-1)^{a_i b_i c_i},\\
\omega_{ij}(a,b,c) &= (-1)^{a_i b_j c_j},\\
\omega_{ijk}(a,b,c) &= (-1)^{a_i b_j c_k}.
\end{aligned}$$

Here i, j and k denote distinct factors (layers) of the direct product group $\mathbb{Z}_2^n$. We have shown above how the phases f_t^ω for each such generator ω can be expressed within the XS-stabilizer formalism by coupling ancilla qubits to the original ones as necessary.

This clearly extends to arbitrary elements of the third cohomology group. Suppose we want to obtain the phases associated with the 3-cocycle $\omega\omega'$ where ω and ω' are any of the above generators. From (2.29) we see that we can construct these phases independently for ω and ω'. This shows that we can describe the ground state subspaces of arbitrary twisted quantum double models $\mathrm{D}^\omega(\mathbb{Z}_2^n)$ with our XS-stabilizer formalism.

Chapter 3

Adiabatic preparation of topological states

This Chapter is based on [NPYK16].

In this chapter we study the preparation of topologically ordered states by interpolating between an initial Hamiltonian with a unique product ground state and a Hamiltonian with a topologically degenerate ground state space. It can be used to initialize a topological quantum memory/computer. This approach of initialization can be done without access to individual degrees of freedom, and without knowing the form of perturbation in the final Hamiltonian. Thus, it may be the preferred initialization method for certain physical systems. By numerically simulating the dynamics mentioned above for small systems, we observe a certain stability of the prepared state as a function of the initial Hamiltonian, which is interesting in the context of fault-tolerant quantum computation, as certain "magic states" can reduce the requirement of achieving fault-tolerance. For small systems or long interpolation times, we argue that the resulting state can be identified by computing suitable effective Hamiltonians. For effective anyon models, this analysis singles out the relevant physical processes and extends the study of the splitting of the topological degeneracy by Bonderson [Bon09]. We illustrate our findings using Kitaevs Majorana chain, effective anyon chains, the toric code and Levin-Wen string-net models.

3.1 Introduction

In this chapter, we are going to explore a very different approach to quantum error correction, which is by making a connection to the topologically ordered phases of matter. This approach was first explored in the seminal work of Kitaev [Kit03]. One of their most attractive features is their ground space degeneracy: it provides a natural quantum error-correcting code for encoding and manipulating information. Remarkably, the ground space degeneracy is approximately preserved in the presence of weak static Hamiltonian perturbations [BHM10, BH11, MZ13]. This feature suppresses the uncontrolled accumulation of relative phases between code states, and thus helps to overcome decoherence. The initial hope was that topological codes such as the toric code can passively protect the quantum information stored in it, once we can realize the toric code Hamiltonian in a material. However, it is realized later that the toric code cannot prevent the transition between code states at non-zero temperature passively. Many ideas have been proposed to make 2D topological codes self-correcting[1], but they all have certain shortcomings. Therefore, a realistic expectation is that if we can make the coupling between a topological quantum memory and its environment low enough, we can possibly store the information there for a decent amount of time, albeit not forever (i.e. storage time scales exponentially with respect to system size).

To use topologically ordered systems as quantum memories and for fault-tolerant quantum

[1]There is no clear definition of "self-correcting". However, it is expected that a self-correcting memory should be able to store information with time $T \sim L^\alpha$ at low temperature, where L is the system size.

computation, concrete procedures for the preparation of specific ground states are required. Such mechanisms depend on the model Hamiltonian which is being realized as well as on the particular experimental realization. Early work [DKLP02] discussed the use of explicit unitary encoding circuits for the toric code. This consideration is natural for systems where we have full access to unitary gates over the underlying degrees of freedom. We may call this the *bottom-up approach* to quantum computing: here one proceeds by building and characterizing individual components before assembling them into larger structures. An example are arrays of superconducting qubits [BKM$^+$14a, CGM$^+$14, CMS$^+$15]. Other proposed procedures for state preparation in this approach involve engineered dissipation [DKP14, BBK$^+$13], measurement-based preparation [LMGH15] or the PEPS preparing algorithm in [STV$^+$13]. However, achieving the control requirements for experimentally performing such procedures is quite challenging. They require either a) independently applying complex sequences of gates on each of the elementary constituents b) precisely engineering a dissipative evolution, or c) performing an extensive set of local measurements and associated non-local classical data processing to determine and execute a suitable unitary correction operation. Imperfections in the implementation of such protocols pose a severe problem, especially in cases where the preparation time is extensive [BHV06, KP14].

In fact, these procedures achieve more than is strictly necessary for quantum computation: any ground state can be prepared in this fashion. That is, they constitute *encoders*, realizing an isometry from a number of unencoded logical qubits to the ground space of the target Hamiltonian. We may ask if the task of preparing topologically ordered state becomes easier if the goal is to prepare specific states instead of encoding arbitrary states. In particular, we may ask this question in the *top-down approach* to quantum computing, where the quantum information is encoded in the ground space of a given condensed matter Hamiltonian. An example are Majorana wires [MZF$^+$12, NPDL$^+$14] or fractional quantum Hall substrates [VYPW11]. Indeed, a fairly standard approach to preparing ground states of a Hamiltonian is to cool the system by weakly coupling it with a thermal bath at a temperature significantly lower than the Hamiltonian gap. Under appropriate ergodicity conditions, this leads to convergence to a state mainly supported on the ground space. Unfortunately, when using natural equilibration processes, convergence may be slow, and the resulting prepared state is generally a (logical) mixed state unsuitable for computation.

A natural alternative method for preparing ground states of a given Hamiltonian is adiabatic evolution: here one initializes the system in an easy-to-prepare state (e.g., a product state), which is the unique ground state of a certain initial Hamiltonian (e.g., describing a uniform field). Subsequently, the Hamiltonian of the system is gradually changed (by tuning external control parameters in a time-dependent fashion) until the target Hamiltonian is reached. If this time-dependent change of the Hamiltonian is "slow enough", i.e., satisfies a certain adiabaticity condition (see Section 3.2), the state of the system will closely follow the trajectory of instantaneous ground states. The resulting state then is guaranteed to be mainly supported on the ground space of the target Hamiltonian, as desired.

Adiabatic preparation has some distinct advantages compared to e.g., encoding using a unitary circuit. For example, in contrast to the latter, adiabatic evolution guarantees that the final state is indeed a ground state of the *actual* Hamiltonian describing the system, independently of potential imperfections in the realization of the ideal Hamiltonians. In contrast, a unitary encoding circuit is designed to encode into the ground space of an ideal model Hamiltonian, and will therefore generally not prepare exact ground states of the actual physical system (which only approximate the model Hamiltonian). Such an encoding into the ideal ground space may lead to a negligible quantum memory time in the presence of an unknown perturbation [PKSC10]; this is because ideal and non-ideal (perturbed) ground states may differ significantly (this phenomenon is referred to as Anderson's orthogonality catastrophe [And67]). Adiabatic evolution, on the other hand, elegantly sidesteps these issues.

The fact that adiabatic evolution can follow the actual ground state of a system Hamiltonian makes it a natural candidate for achieving the task of topological code state preparation. An additional attractive feature is that its experimental requirements are rather modest: while some time-dependent control is required, this can be local, and additionally translation-invariant.

Namely, the number of external control parameters required does not scale with the system size or code distance.

Summary and outlook

Motivated by these observations, we consider the general problem of preparing topologically ordered states by what we refer to as *Hamiltonian interpolation*. We will use this terminology instead of "adiabatic evolution" since in some cases, it makes sense to consider scenarios where adiabaticity guarantees cannot be given. For concreteness, we consider a time-dependent Hamiltonian $H(t)$ which monotonically sweeps over the path

$$H(t) = (1 - t/T) \cdot H_{\mathsf{triv}} + t/T \cdot H_{\mathsf{top}} \qquad t \in [0, T] \ , \tag{3.1}$$

i.e., we assume that the interpolation is linear in time and takes overall time[2] T. Guided by experimental considerations, we focus on the translation-invariant case: here the Hamiltonians $H(t)$ are translation-invariant throughout the evolution. More precisely, we consider the process of interpolating between a Hamiltonian H_{triv} with unique ground state $\Psi(0) = \varphi^{\otimes L}$ and a Hamiltonian H_{top} with topologically degenerate ground space (which is separated from the remainder of the spectrum by a constant gap): the state $\Psi(t)$ of the system at time $t \in [0, T]$ satisfies the equation of motion

$$\frac{\partial \Psi(t)}{\partial t} = -iH(t)\Psi(t) \ , \qquad \Psi(0) = \varphi^{\otimes L} \ . \tag{3.2}$$

Generally, we consider families of Hamiltonians (or models) parametrized by a system size L; throughout, we will assume that L is the number of single particles, e.g., the number of qubits (or sites) in a lattice with Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes L}$. The dimension of the ground space of H_{top} will be assumed to be independent of the system size.

Our goal is to characterize the set of states which are preparable by such Hamiltonian interpolations starting from various product states, i.e., by choosing different initial Hamiltonians H_{triv}. To each choice $\Psi(0) = \varphi^{\otimes L}$ of product state we associate a normalized initial trivial Hamiltonian $H_{\mathsf{triv}} := -\sum_j P_\varphi^{(j)}$ which fully specifies the interpolating path of Eq. (3.1), with $P_\varphi^{(j)} = |\varphi\rangle\langle\varphi|$ being the single particle projector onto the state φ at site j.

In the limit $T \to \infty$, one may think of this procedure as associating an encoded (logical) state $\iota(\varphi)$ to any single-particle state φ. However, some caveats are in order: first, the global phase of the state $\iota(\varphi)$ cannot be defined in a consistent manner in the limit $T \to \infty$, and is therefore not fixed. Second, the final state in the evolution (3.2) does not need to be supported entirely on the ground space of H_{top} because of non-adiaticity errors, i.e., it is not a logical (encoded) state itself. To obtain a logical state, we should think of $\iota(\varphi)$ as the final state projected onto the ground space of H_{top}. Up to these caveats, our goal is essentially to characterize the image of the association $\iota : \varphi \mapsto \iota(\varphi)$, as well as its continuity properties. We will also define an analogous map ι_T associated to fixed evolution time T and study it numerically by simulating the corresponding Schrödinger equation (3.2) on a classical computer.

While there is a priori no obvious relationship between the final states $\iota_T(\varphi)$, $\iota_T(\varphi')$ resulting from different initial (product) states $\varphi^{\otimes L}, \varphi'^{\otimes L}$, we numerically find that the image of ι_T is concentrated around a particular discrete family of encoded states. In particular, we observe for small system sizes that the preparation enjoys a certain stability property: variations in the initial Hamiltonian do not significantly affect the final state. We support this through analytic arguments, computing effective Hamiltonians associated to perturbations around H_{top} which address the large T limit. This also allows us to provide a partial prediction of which states $\iota(\varphi)$ may be obtained through such a preparation process. We find that under certain general conditions, $\iota(\varphi)$ belongs to a certain finite family of preferred states which depend on the final Hamiltonian H_{top}. As we will argue, there is a natural relation between the corresponding states $\iota(\varphi)$ for different system

[2] We remark that in some cases, using a non-linear monotone 'schedule' $\vartheta : [0, T] \to [0, 1]$ with $\vartheta(0) = 0$, $\vartheta(T) = 1$ and smooth derivatives may be advantageous (see Discussion in Section 3.2). However, for most of our considerations, the simple linear interpolation (3.1) is sufficient.

sizes: they encode the same logical state if corresponding logical operators are chosen (amounting to a choice of basis of the ground space).

Characterizing the set $\{\iota(\varphi)\}_\varphi$ of states preparable using this kind of Hamiltonian interpolation is important for quantum computation because certain encoded states (referred to as "magic states") can be used as a resource for universal computation [BK05]. This chapter provides insight into this question for 'small' systems, which we deem experimentally relevant. Indeed, there is a promising degree of robustness for the Hamiltonian interpolation to prepare certain (stabilizer) states. However, a similar preparation of magic states seems to require imposing additional symmetries which will in general not be robust. We exemplify our considerations using various concrete models, including Kitaev's Majorana chain [Kit01] (for which we can provide an exact solution), effective anyon chains (related to the so-called golden chain [FTL$^+$07] and the description used by Bonderson [Bon09]), as well as the toric code [Kit03] and Levin-Wen string-net models [LW05] (for which we simulate the time-evolution for small systems, for both the doubled semion and the doubled Fibonacci model).

Prior work

The problem of preparing topologically ordered states by adiabatic interpolation has been considered before by Hamma and Lidar [HL08]. Indeed, their contribution is one of the main motivations for our study. They study an adiabatic evolution where a Hamiltonian having a trivial product ground state is interpolated into a toric code Hamiltonian having a four-fold degenerate ground state space. They found that while the gap for such an evolution must forcibly close, this may happen through second order phase transitions. Correspondingly, the closing of the gap is only polynomial in the system size. This allows an efficient polynomial-time Hamiltonian interpolation to succeed at accurately preparing certain ground states. We revisit this case in Section 3.2.1 and give further examples of this phenomenon. The authors of [HZHL08] also observed the stability of the encoded states with respect to perturbations in the preparation process.

Bonderson [Bon09] considered the problem of characterizing the lowest order degeneracy splitting in topologically ordered models. Degeneracy lifting can be associated to tunneling of anyonic charges, part of which may be predicted by the universal algebraic structure of the anyon model. Our conclusions associated to Sections 3.5 and 3.6 can be seen as supporting this perspective.

Beyond small systems

In general, the case of larger systems (i.e., the thermodynamic limit) requires a detailed understanding of the quantum phase transitions [Sac11] occurring when interpolating between H_{triv} and H_{top}. Taking the thermodynamic limit while making T scale as a polynomial of the system size raises a number of subtle points. A major technical difficulty is that existing adiabatic theorems do not apply, since at the phase transition gaps associated to either of the relevant phases close. This is alleviated by scaling the interpolation time T with the system size and splitting the adiabatic evolution into two regimes, the second of which can be treated using degenerate adiabatic perturbation theory [RO10, RO12, RO14]. However, such a methodology still does not yield complete information about the dynamical effects of crossing a phase boundary.

More generally, it is natural to conjecture that interpolation between different phases yields only a discrete number of distinct states corresponding to a discrete set of continuous phase transitions in the thermodynamic limit. Such a conjecture links the problem of Hamiltonian interpolation to that of classifying phase transitions between topological phases. It can be motivated by the fact that only a discrete set of possible condensate-induced continuous phase transitions is predicted to exist in the thermodynamic limit [BS09, BSS11].

3.2 Adiabaticity and ground states

The first basic question arising in this context is whether the evolution (3.2) yields a state $\Psi(T)$ close to the ground space of H_{top}. The adiabatic theorem in its multiple forms (see e.g., [Teu03])

provides *sufficient* conditions for this to hold: These theorems guarantee that given a Hamiltonian path $\{H(t)\}_{0\leq t\leq T}$ satisfying certain smoothness and gap assumptions, initial eigenstates evolve into approximate instantaneous eigenstates under an evolution of the form (3.2). The latter assumptions are usually of the following kind:

(i) **Uniform gap:** There is a uniform lower bound $\Delta(t) \geq \Delta > 0$ on the spectral gap of $H(t)$ for all $t \in [0, T]$. The relevant spectral gap $\Delta(t)$ is the energy difference between the ground space $P_0(t)\mathcal{H}$ of the instantaneous Hamiltonian $H(t)$ and the rest of its spectrum. Here and below, we denote by $P_0(t)$ the spectral projection onto the ground space [3] of $H(t)$.

(ii) **Smoothness:** There are constants $c_1, \ldots, c_M$ such that the M first derivatives of $H(t)$ are uniformly bounded in operator norm, i.e., for all $j = 1, \ldots, M$, we have

$$\big\| \frac{d^j}{dt^j} H(t) \big\| \leq c_j \qquad \text{for all } t \in [0, T] \ . \tag{3.3}$$

The simplest version of such a theorem is:

Theorem 3.2.1. *Given a state $\Psi(0)$ such that $P_0(0)\Psi(0) = \Psi(0)$ and a uniformly gapped Hamiltonian path $H(t)$ for $t \in [0, T]$ given by Eq.* (3.1)*, the state $\Psi(T)$ resulting from the evolution* (3.2) *satisfies*

$$\|\Psi(T) - P_0(T)\Psi(T)\| = O(1/T) \ .$$

In other words, in the adiabatic limit of large times T, the state $\Psi(T)$ belongs to the instantaneous eigenspace $P_0(T)\mathcal{H}$ and its distance from the eigenspace is $O(1/T)$.

This version is sufficient to support our analytical conclusions qualitatively. For a quantitative analysis of non-adiabaticity errors, we perform numerical simulations. Improved versions of the adiabatic theorem (see [GMC15, LRH09]) provide tighter analytical error estimates for general interpolation schedules at the cost of involving higher order derivatives of the Hamiltonian path $H(t)$ (see Eq.‘(3.3)), but do not change our main conclusions.

Several facts prevent us from directly applying such an adiabatic theorem to our evolution (3.1) under consideration.

Topological ground space degeneracy. Most notably, the gap assumption (i) is not satisfied if we study ground spaces: we generally consider the case where $H(0) = H_{\mathsf{triv}}$ has a unique ground state, whereas the final Hamiltonian $H(T) = H_{\mathsf{top}}$ is topologically ordered and has a degenerate ground space (in fact, this degeneracy is exact and independent of the system size for the models we consider). This means that if $P_0(t)$ is the projection onto the ground space of $H(t)$, there is no uniform lower bound on the gap $\Delta(t)$.

We will address this issue by restricting our attention to times $t \in [0, \kappa T]$, where $\kappa \approx 1$ is chosen such that $H(\kappa T)$ has a non-vanishing gap but still is “inside the topological phase”. We will illustrate in specific examples how $\Psi(T)$ can indeed be recovered by taking the limit $\kappa \to 1$.

We emphasize that the expression “inside the phase” is physically not well-defined at this point since we are considering a Hamiltonian of a fixed size. Computationally, we take it to mean that the Hamiltonian can be analyzed by a convergent perturbation theory expansion starting from the unperturbed Hamiltonian H_{top}. The resulting lifting of the ground space degeneracy of H_{top} will be discussed in more detail in Section 3.3.

Dependence on the system size. A second potential obstacle for the use of the adiabatic theorem is the dependence on the system size L (where e.g., L is the number of qubits). This dependence enters in the operator norms (3.3), which are extensive in L – this would lead to polynomial dependence of T on L even if e.g., the gap were constant (uniformly bounded).

[3] More generally, $P_0(t)$ may be the sum of the spectral projections of $H(t)$ with eigenvalues in a given interval, which is separated by a gap $\Delta(t)$ from the rest of the spectrum.

More importantly, the system size enters in the gap $\Delta(t)$: in the topological phase, the gap (i.e., the splitting of the topological degeneracy of H_{top}) is exponentially small in L for constant-strength local perturbations to H_{top}, as shown for the models considered here by Bravyi, Hastings and Michalakis [BHM10]. Thus a naïve application of the adiabatic theorem only yields a guarantee on the ground space overlap of the final state if the evolution time is exponentially large in L. This is clearly undesirable for large systems; one may try to prepare systems faster (i.e., more efficiently) but would need alternate arguments to ensure that the final state indeed belongs to the ground space of H_{top}.

For these reasons, we restrict our attention to the following two special cases of the Hamiltonian interpolation (3.1):

- *Symmetry-protected preparation*: if there is a set of observables commuting with both H_{triv} and H_{top}, these will represent conserved quantities throughout the Hamiltonian interpolation. If the initial state is an eigenstate of such observables, one may restrict the Hilbert space to the relevant eigenvalue, possibly resolving the topological degeneracy and guaranteeing a uniform gap. This observation was first used in [HL08] in the context of the toric code: for this model, such a restriction allows mapping the problem to a transverse field Ising model, where the gap closes polynomialy with the system size. We identify important cases satisfying this condition. While this provides the most robust preparation scheme, the resulting encoded states are somewhat restricted (see Section 3.2.1).

- *Small systems:* For systems of relatively small (constant) size L , the adiabatic theorem can be applied as all involved quantities are essentially constant. In other words, although 'long' interpolation times are needed to reach ground states of H_{top} (indeed, these may depend exponentially on L), these may still be reasonable experimentally. The consideration of small system is motivated by current experimental efforts to realize surface codes [KBF$^+$15]: they are usually restricted to a small number of qubits, and this is the scenario we are considering here (see Section 3.2.2).

Obtaining a detailed understanding of the general large L limiting behaviour (i.e., the thermodynamic limit) of the interpolation process (3.1) is beyond the scope of this thesis.

3.2.1 Symmetry-protected preparation

Under particular circumstances, the existence of conserved quantities permits applying the adiabatic theorem while evading the technical obstacle posed by a vanishing gap in the context of topological order. Such a case was considered by Hamma and Lidar [HL08], who showed that certain ground states of the toric code can be prepared efficiently. We can formalize sufficient conditions in the following general way (which then is applicable to a variety of models, as we discuss below).

Observation 3.2.2. *Consider the interpolation process* (3.1) *in a Hilbert space* $\mathcal{H}$*. Let* $P_0(T)$ *be the projection onto the ground space* $P_0(T)\mathcal{H}$ *of* $H(T) = H_{\mathsf{top}}$*. Suppose that* $Q = Q^2$ *is a projection such that*

(i) Q *is a conserved quantity:* $[Q, H_{\mathsf{top}}] = [Q, H_{\mathsf{triv}}] = 0$*.*

(ii) The initial state $\Psi(0)$ *is the ground state of* H_{triv}*, i.e.,* $P_0(0)\Psi(0) = \Psi(0)$ *and satisfies* $Q\Psi(0) = \Psi(0)$*.*

(iii) The final ground space has support on $QP_0(T)\mathcal{H} \neq 0$

(iv) The restriction $QH(t)$ *of* $H(t)$ *to* $Q\mathcal{H}$ *has gap* $\Delta(t)$ *which is bounded by a constant* Δ *uniformly in* t*, i.e.,* $\Delta(t) \geq \Delta$ *for all* $t \in [0, T]$*.*

Then $Q\Psi(t) = \Psi(t)$*, and the adiabatic theorem can be applied with lower bound* Δ *on the gap, yielding* $\|\Psi(T) - P_0(T)\Psi(T)\| \leq O(1/T)$*.*

The proof of this statement is a straightforward application of the adiabatic theorem (Theorem 3.2.1) to the Hamiltonians QH_{triv} and QH_{top} in the restricted subspace $Q\mathcal{H}$. In the following sections, we will apply Observation 3.2.2 to various systems. It not only guarantees that the ground space is reached, but also gives us information about the specific state prepared in a degenerate ground space.

As an example of the situation discussed in Observation 3.2.2, we discuss the case of fermionic parity conservation in Section 3.4. This symmetry is naturally present in fermionic systems. We expect our arguments to extend to more general topologically ordered Hamiltonians with additional symmetries. It is well-known that imposing global symmetries on top of topological Hamiltonians provides interesting classes of systems. Such symmetries can exchange anyonic excitations, and their classification as well as the construction of associated defect lines in topological Hamiltonians is a topic of ongoing research [BSW11, KK12, BJQ13]. The latter problem is intimately related to the realization (see e.g., [BMD09, Bom15]) of transversal logical gates, which leads to similar classification problems [BK13, BBK$^+$14, Yos15b, Yos15a]. Thus we expect that there is a close connection between adiabatically preparable states and transversally implementable logical gates. Indeed, a starting point for establishing such a connection could be the consideration of interpolation processes respecting symmetries realized by transversal logical gates.

For later reference, we also briefly discuss a situation involving conserved quantities which – in contrast to Observation 3.2.2 – project onto excited states of the final Hamiltonian. In this case, starting with certain eigenstates of the corresponding symmetry operator Q, the ground space cannot be reached:

Observation 3.2.3. *Assume that $Q, H_{\mathsf{triv}}, H_{\mathsf{top}}, \Psi(0)$ obey properties* (i),(ii) *and* (iv) *of Observation 3.2.2. If the ground space $P_0(T)\mathcal{H}$ of H_{top} satisfies $QP_0(T)\mathcal{H} = 0$ (i.e., is orthogonal to the image of Q), then the Hamiltonian interpolation cannot reach the ground space of H_{top}, i.e., $\langle \Psi(T), P_0(T)\Psi(T)\rangle = \Omega(1)$.*

The proof of this observation is trivial since Q is a conserved quantity of the Schrödinger evolution. Physically, the assumptions imply the occurrence of a level-crossing where the energy gap exactly vanishes and eigenvalue of Q restricted to the ground space changes. We will encounter this scenario in the case of the toric code on a honeycomb lattice, see Section 3.7.3.

3.2.2 Small-system case

In a more general scenario, there may not be a conserved quantity as in Observation 3.2.2. Even assuming that the ground space is reached by the interpolation process (3.1), it is a priori unclear which of the ground states is prepared. Here we address this question.

As remarked earlier, we focus on systems of a constant size L, and assume that the preparation time T is large compared to L. Generically, the Hamiltonians $H(t)$ are then non-degenerate (except at the endpoint, $t \approx T$, where $H(t)$ approaches H_{top}). Without fine tuning, we may expect that there are no exact level crossings in the spectrum of $H(t)$ along the path $t \mapsto H(t)$ (say for some times $t \in [0, \kappa T]$, $\kappa \approx 1$). For sufficiently large overall evolution times T, we may apply the adiabatic theorem to conclude that the state of the system follows the (unique) instantaneous ground state (up to a constant error). Since our focus is on small systems, we will henceforth assume that this is indeed the case, and summarily refer to this as the *adiabaticity assumption*. Again, we emphasize that this is a priori only reasonable for small systems.

Under the adiabaticity assumption, we can conclude that the prepared state $\Psi(T)$ roughly coincides with the state obtained by computing the (unique) ground state ψ_κ of $H(\kappa T)$, and taking the limit $\kappa \to 1$. In what follows, we adopt this computational prescription for identifying prepared states. Indeed, this approach yields states that match our numerical simulation, and provides the correct answer for certain exactly solvable cases. Furthermore, the computation of the states ψ_κ (in the limit $\kappa \to 1$) also clarifies the physical mechanisms responsible for the observed stability property of preparation: we can relate the prepared states to certain linear combination of string-operators (Wilson-loops), whose coefficients depend on the geometry (length) of these loops, as well as the amplitudes of certain local particle creation/annihilation and tunneling processes.

Since $H(\kappa T)$ for $\kappa \approx 1$ is close to the topologically ordered Hamiltonian H_{top}, it is natural to use ground states (or logical operators) of the latter as a reference to express the instantaneous states ψ_κ. Indeed, the problem essentially reduces to a system described by H_{top}, with an additional perturbation given by a scalar multiple of H_{triv}. Such a local perturbation generically splits the topological degeneracy of the ground space. The basic mechanism responsible for this splitting for topologically ordered systems has been investigated by Bonderson [Bon09], who quantified the degeneracy splitting in terms of local anyon-processes. We seek to identify low-energy ground states: this amounts to considering the effective low-energy dynamics (see Section 3.3). This will provide valuable information concerning the set $\{\iota(\varphi)\}$.

3.3 Effective Hamiltonians

As discussed in Section 3.2.2, for small systems (and sufficiently large times T), the state $\Psi(\kappa T)$ in the interpolation process (3.1) should coincide with the ground state of the instantaneous Hamiltonian $H(\kappa T)$. For $\kappa \approx 1$, the latter is a perturbed version of the Hamiltonian H_{top}, where the perturbation is a scalar multiple of H_{triv}. That is, up to rescaling by an overall constant, we are concerned with a Hamiltonian of the form

$$H_0 + \epsilon V \tag{3.4}$$

where $H_0 = H_{\mathsf{top}}$ is the target Hamitonian and $V = H_{\mathsf{triv}}$ is the perturbation. To compute the ground state of a Hamiltonian of the form (3.4), we use *effective Hamiltonians*. These provide a description of the system in terms of effective low-energy degrees of freedom.

3.3.1 Low-energy degrees of freedom

Let us denote by P_0 the projection onto the degenerate ground space of H_0. Since H_0 is assumed to have a constant gap, a perturbation of the form (3.4) effectively preserves the low-energy subspace $P_0\mathcal{H}$ for small $\epsilon > 0$, and generates a dynamics on this subspace according to an *effective Hamiltonian* $H_{\mathsf{eff}}(\epsilon)$. We will discuss natural definitions of this effective Hamiltonian in Section 3.3.3. For the purpose of this section, it suffices to mention that it is entirely supported on the ground space of H_0, i.e., $H_{\mathsf{eff}}(\epsilon) = P_0 H_{\mathsf{eff}}(\epsilon) P_0$. As such, it has spectral decomposition

$$H_{\mathsf{eff}}(\epsilon) = \sum_{k=0}^{K-1} E_k^{\mathsf{eff}}(\epsilon)\Pi_k^{\mathsf{eff}}(\epsilon) \ , \tag{3.5}$$

where $E_0^{\mathsf{eff}} < E_1^{\mathsf{eff}} < \ldots$, and where $\Pi_k^{\mathsf{eff}}(\epsilon) = \Pi_k^{\mathsf{eff}}(\epsilon) P_0$ are commuting projections onto subspaces of the ground space $P_0\mathcal{H}$ of H_0. (Generally, we expect $H_{\mathsf{eff}}(\epsilon)$ to be non-degenerate such that $K = \dim P_0\mathcal{H}$.) In particular, the effective Hamiltonian (3.5) gives rise to an orthogonal decomposition of the ground space $P_0\mathcal{H}$ by projections $\{\Pi_k^{\mathsf{eff}}(\epsilon)\}_{k=0}^{K-1}$. States in $\Pi_0^{\mathsf{eff}}(\epsilon)\mathcal{H}$ are distinguished by having minimal energy. We can take the limiting projections as the perturbation strength goes to 0, setting

$$\Pi_k^{\mathsf{eff}}(0) = \lim_{\epsilon \to 0} \Pi_k^{\mathsf{eff}}(\epsilon) \qquad \text{for } k = 0, \ldots, K-1 \ .$$

In particular, the effective Hamiltonian $H_{\mathsf{eff}}(\epsilon)$ has ground space $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ in the limit $\epsilon \to 0$. Studying $H_{\mathsf{eff}}(\epsilon)$, and, in particular, the space $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ appears to be of independent interest, as it determines how perturbations affect the topologically ordered ground space beyond spectral considerations as in [Bon09].

3.3.2 Hamiltonian interpolation and effective Hamiltonians

The connection to the interpolation process (3.1) is then given by the following conjecture. It is motivated by the discussion in Section 3.2.2 and deals with the case where there are no conserved quantities (unlike, e.g., in the case of the Majorana chain, as discussed in Section 3.4).

Conjecture 1. *Under suitable adiabaticity assumptions (see Section 3.2.2) the projection of the final state $\Psi(T)$ onto the ground space of H_{top} belongs to $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ (up to negligible errors[4]), i.e., it is a ground state of the effective Hamiltonian $H_{\mathsf{eff}}(\epsilon)$ in the limit $\epsilon \to 0$.*

In addition to the arguments in Section 3.2.2, we provide evidence for this conjecture by explicit examples, where we illustrate how $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ can be computed analytically. We also verify that Conjecture 1 correctly determines the final states by numerically studying the evolution (3.1).

We remark that the statement of Conjecture 1 severly constrains the states that can be prepared by Hamiltonian interpolation in the large T limit: we will argue that the space $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ has a certain robustness with respect to the choice of the initial Hamiltonian H_{triv}. In fact, the space $\Pi_0^{\mathsf{eff}}(0)\mathcal{H}$ is typically 1-dimensional and spanned by a single vector φ_0. Furthermore, this vector φ_0 typically belongs to a finite family $\mathcal{A} \subset P_0\mathcal{H}$ of states defined solely by H_{top}. In particular, under Conjecture 1, the dependence of the final state $\Psi(T)$ on the Hamiltonian H_{triv} is very limited: the choice of H_{triv} only determines which of the states in $\mathcal{A}$ is prepared. We numerically verify that the resulting target states $\Psi(T)$ indeed belong to the finite family $\mathcal{A}$ of states obtained analytically.

3.3.3 Perturbative effective Hamiltonians

As discussed in Section 3.3.2, we obtain distinguished final ground states by computation of suitable effective Hamiltonians $H_{\mathsf{eff}}(\epsilon)$, approximating the action of $H_0+\epsilon V$ on the ground space $P_0\mathcal{H}$ of H_0. In many cases of interest, computing this effective Hamiltonian (whose definition for the Schrieffer-Wolff-case we present in Appendix 3.A.1) exactly is infeasible (The effective Hamiltonian for the Majorana chain (see Section 3.4) is an exception.).

Instead, we seek a perturbative expansion

$$H_{\mathsf{eff}}^{(n)} = \sum_{n=0}^{\infty} \epsilon^n X_n$$

in terms of powers of the perturbation strength ϵ. This is particularly natural as we are interested in the limit $\epsilon \to 0$ anyway (see Conjecture 1). Furthermore, it turns out that such perturbative expansions provide insight into the physical mechanisms underlying the 'selection' of particular ground states.

We remark that there are several different methods for obtaining low-energy effective Hamiltonians. The *Schrieffer-Wolff method* [SW66, BDL11] provides a unitary U such that $H_{\mathsf{eff}} = U(H_0+\epsilon V)U^\dagger$ preserves $P_0\mathcal{H}$ and can be regarded as an effective Hamiltonian. One systematically obtains a series expansion

$$S = \sum_{n=1}^{\infty} \epsilon^n S_n \qquad \text{where } S_n^\dagger = -S_n$$

for the anti-Hermitian generator S of $U = e^S$; this then naturally gives rise to an order-by-order expansion

$$H_{\mathsf{eff}}^{(n)} = H_0P_0 + \epsilon P_0 V P_0 + \sum_{q=2}^{n} \epsilon^q H_{\mathsf{eff},q} \ . \tag{3.6}$$

of the effective Hamiltonian, where P_0 is the projection onto the ground space $P_0\mathcal{H}$ of H_0 (explicit expressions are given in Appendix 3.A.2).

Using the Schrieffer-Wolff method has several distinct advantages, including the fact that

(i) the resulting effective Hamiltonian H_{eff}, as well as the terms $H_{\mathsf{eff}}^{(n)}$ are Hermitian, and hence have a clear physical interpretation. This is not the case e.g., for the Bloch expansion [Blo58].

[4]By negligible, we mean that the errors can be made to approach zero as T is increased.

(ii) There is no need to address certain self-consistency conditions arising e.g., when using the Dyson equation and corresponding self-energy methods [ABD75, FW03]

We point out that the series resulting by taking the limit $n \to \infty$ in (3.6) has the usual convergence issues encountered in many-body physics: convergence is guaranteed only if $\|\epsilon V\| \leq \Delta$, where Δ is the gap of H_0. For a many-body system with extensive Hilbert space (e.g., L spins), the norm $\|V\| = \Omega(L)$ is extensive while the gap $\Delta = O(1)$ is constant, leading to convergence only in a regime where $\epsilon = O(1/L)$. In this respect, the Schrieffer-Wolff method does not provide direct advantages compared to other methods. As we are considering the limit $\epsilon \to 0$, this is not an issue (also, for small systems as those considered in our numerics, we do not have such issues either).

We point out, however, that the results obtained by Bravyi et al. [BDL11] suggest that considering partial sums of the form (3.6) is meaningful even in cases in which the usual convergence guarantees are not given: indeed, [BDL11, Theorem 3] shows that the ground state energies of $H_{\mathsf{eff}}^{(n)}$ and $H_0 + \epsilon V$ are approximately equal for suitable choices of ϵ and n. Another key feature of the Schrieffer-Wolff method is the fact that the effective Hamiltonians $H_{\mathsf{eff}}^{(n)}$ are essentially local (for low orders n) when the method is applied to certain many-body systems, see [BDL11]. We will not need the corresponding results here, however.

Unfortunately, computing the Schrieffer-Wolff Hamiltonian $H_{\mathsf{eff}}^{(n)}$ generally involves a large amount of combinatorics (see [BDL11] for a diagrammatic formalism for this purpose). In this respect, other methods may appear to be somewhat more accessible. Let us mention in particular the method involving the Dyson equation (and the so-called 'self-energy' operator), which was used e.g., in [Kit06, Section 5.1] to compute 4-th order effective Hamiltonians. This leads to remarkably simple expressions of the form

$$P_0(VG)^{n-1}VP_0 \tag{3.7}$$

for the n-th order term effective Hamiltonian, where $G = G(E_0)$ is the resolvent operator

$$G(z) = (I - P_0)(zI - H_0)^{-1}(I - P_0) \tag{3.8}$$

evaluated at the ground state energy E_0 of H_0. In general, though, the expression (3.7) only coincides with the Schrieffer-Wolff-method (that is, (3.6)) up to the lowest non-trivial order.

3.3.4 Perturbative effective Hamiltonians for topological order

Here we identify simple conditions under which the Schrieffer-Wolff Hamiltonian of lowest non-trivial order has the simple structure (3.7). We will see that these conditions are satisfied for the systems we are interested in. In other words, for our purposes, the self-energy methods and the Schrieffer-Wolff method are equivalent. While establishing this statement (see Theorem 3.3.2 below) requires some work, this result vastly simplifies the subsequent analysis of concrete systems.

The condition we need is closely related to quantum error correction [KL97]. In fact, this condition has been identified as one of the requirements for topological quantum order (TQO-1) in Ref. [BHM10]. To motivate it, consider the case where $P_0\mathcal{H}$ is an error-correcting code of distance L. Then all operators T acting on less than L particles[5] have trivial action on the code space, i.e., for such T, the operator P_0TP_0 is proportional to P_0 (which we will write as $P_0TP_0 \in \mathbb{C}P_0$). In particular, this means that if V is a Hermitian linear combination of single-particle operators, then $P_0V^nP_0 \in \mathbb{C}P_0$ for all $n < L$. The condition we need is a refinement of this error-correction criterion that incorporates energies (using the resolvent):

Definition 3.3.1. *We say that the pair (H_0, V)* satisfies the topological order condition with parameter L *if L is the smallest interger such that for all $n < L$, we have*

$$P_0VZ_1VZ_2\cdots Z_{n-1}VP_0 \in \mathbb{C}P_0 \tag{3.9}$$

for all $Z_j \in \{P_0, Q_0\} \cup \{G^m \mid m \in \mathbb{N}\}$. Here P_0 is the ground space projection of H_0, $Q_0 = I - P_0$ is the projection onto the orthogonal complement, and $G = G(E_0)$ is the resolvent (3.8) *(supported on $Q_0\mathcal{H}$).*

[5] By particle we mean a physical constituent qubit or qudit degree of freedom.

We remark that this definition is easily verified in the systems we consider: if excitations in the system are local, the resolvent operators and projection in a product of the form (3.9) can be replaced by local operators, and condition (3.9) essentially reduces to a standard error correction condition for operators with local support.

Assuming this definition, we then have the following result:

Theorem 3.3.2. *Suppose that (H_0, V) satisfies the topological order condition with parameter L. Then the n-th order Schrieffer-Wolff effective Hamiltonian satisfies*

$$H_{\mathsf{eff}}^{(n)} \in \mathbb{C}P_0 \qquad \text{for all } n < L \ ,$$

i.e., the effective Hamiltonian is trivial for these orders, and

$$H_{\mathsf{eff}}^{(L)} = P_0(VG)^{L-1}VP_0 + \mathbb{C}P_0 \ .$$

We give the proof of this statement in Appendix 3.A.

3.4 The Majorana chain

In this section, we apply our general results to Kitaev's Majorana chain. We describe the model in Section 3.4.1. In Section 3.4.2, we argue that the interpolation process (3.2) is an instance of symmetry-protected preparation; this allows us to identify the resulting final state. We also observe that the effective Hamiltonian is essentially given by a 'string'-operator F, which happens to be the fermionic parity operator in this case. That is, up to a global energy shift, we have

$$H_{\mathsf{eff}} \approx f \cdot F$$

for a certain constant f depending on the choice of perturbation.

3.4.1 The model

Here we consider the case where H_{top} is Kitaev's Majorana chain [Kit01], a system of spinless electrons confined to a line of L sites. In terms of $2L$ Majorana operators $\{c_p\}_{p=1}^{2L}$ satisfying the anticommutation relations

$$\{c_p, c_q\} = 2\delta_{p,q} \cdot I$$

as well as $c_p^2 = I$, $c_p^\dagger = c_p$, the Hamiltonian has the form

$$H_{\mathsf{top}} = \frac{i}{2}\sum_{j=1}^{L-1} c_{2j}c_{2j+1} \ . \tag{3.10}$$

Without loss of generality, we have chosen the normalization such that elementary excitations have unit energy. The Hamiltonian has a two-fold degenerate ground space. The Majorana operators c_1 and c_{2L} correspond to a complex boundary mode, and combine to form a Dirac fermion

$$a = \frac{1}{2}(c_1 + ic_{2L}) \tag{3.11}$$

which commutes with the Hamiltonian. The operator $a^\dagger a$ hence provides a natural occupation number basis $\{|g_\sigma\rangle\}_{\sigma\in\{0,1\}}$ for the ground space $P_0\mathcal{H}$ defined (up to arbitrary phases) by

$$a^\dagger a|g_\sigma\rangle = \sigma|g_\sigma\rangle \qquad \text{for } \sigma \in \{0,1\} \ .$$

As a side remark, note that the states $|g_0\rangle$ and $|g_1\rangle$ cannot be used directly to encode a qubit. This is because they have even and odd fermionic parity, respectively, and thus belong to different superselection sectors. In other words, coherent superposition between different parity sectors

are nonphysical. This issue can be circumvented by using another fermion or a second chain, see [BK12]. Since the conclusions of the following discussion will be unchanged, we will neglect this detail for simplicity.

We remark that the Hamiltonian H_{top} of Eq. (3.10) belongs to a one-parameter family of extensively studied and well-understood quantum spin Hamiltonians. Indeed, the Jordan-Wigner transform of the Hamiltonian (with $g \in \mathbb{R}$ an arbitrary parameter)

$$H_{I,g} = \frac{i}{2} \sum_{j=1}^{L-1} c_{2j} c_{2j+1} - \frac{gi}{2} \sum_{j=1}^{L} c_{2j-1} c_{2j}. \tag{3.12}$$

is the transverse field Ising model

$$H'_{I,g} = -\frac{1}{2} \sum_{j=1}^{L-1} X_j X_{j+1} + \frac{g}{2} \sum_{j=1}^{L} Z_j$$

where X_j and Z_j are the spin 1/2 Pauli matrices acting on qubit j, $j = 1, \ldots, L$. This transformation allows analytically calculating the complete spectrum of the translation invariant chain for both periodic and open boundary conditions [Pfe70].

The Hamiltonian $H'_{I,g}$ has a quantum phase transition at $g = 1$, for which the lowest energy modes in the periodic chain have an energy scaling as $1/L$. The open boundary case has been popularized by Kitaev as the Majorana chain and has a unique low energy mode a (see Eq. (3.11)) which has zero energy for $g = 0$ and for finite $0 < g < 1$, becomes a dressed mode with exponentially small energy (in L) and which is exponentially localized at the boundaries.

3.4.2 State preparation by interpolation

The second term in (3.12) may be taken to be the initial Hamiltonian H_{triv} for the interpolation process. More generally, to prepare ground states of H_{top}, we may assume that our initial Hamiltonian is a quadratic Hamiltonian with a unique ground state. That is, H_{triv} is of the form

$$H_{\mathsf{triv}} = \frac{i}{4} \sum_{p,q=1}^{2L} \mathbf{V}_{p,q} c_p c_q \ ,$$

where $\mathbf{V}$ is a real antisymmetric $2L \times 2L$ matrix. We will assume that it is bounded and local (with *range* r) in the sense that

$$\|\mathbf{V}\| \leq 1 \qquad \text{and} \qquad \mathbf{V}_{p,q} = 0 \text{ if } |p - q| > r \ ,$$

where $\|\cdot\|$ denotes the operator norm. As shown in [BK12, Theorem 1], the Hamiltonian $H_{\mathsf{top}} + \epsilon H_{\mathsf{triv}}$ has two lowest energy states with exponentially small energy difference, and this lowest-energy space remains separated from the rest of the spectrum by a constant gap for a fixed (constant) perturbation strength $\epsilon > 0$. Estimates on the gap along the complete path $H(t)$ are, to the best of our knowledge, not known in this more general situation.

Let us assume that $\Psi(0)$ is the unique ground state of H_{triv} and consider the linear interpolation (3.2). The corresponding process is an instance of the symmetry-protected preparation, i.e., Observation 3.2.2 applies in this case. Indeed, the *fermionic parity operator*

$$F = \prod_{j=1}^{L} (-i) c_{2j-1} c_{2j} \ , \tag{3.13}$$

commutes with both H_{triv} and H_{top}. Therefore, the initial ground state $\Psi(0)$ lies either in the even-parity sector, i.e., $F\Psi(0) = \Psi(0)$, or in the odd-parity sector ($F\Psi(0) = -\Psi(0)$). (Even parity is usually assumed by convention, since the fermionic normal modes used to describe the system are chosen to have positive energy.) In any case, the ± 1 eigenvalue of the initial ground state with respect to F will persist throughout the full interpolation. This fixes the final state:

Lemma 3.4.1. *Under suitable adiabaticity assumptions (see Observation 3.2.2), the resulting state in the evolution* (3.2) *is (up to a phase) given by the ground state* $|g_0\rangle$ *or* $|g_1\rangle$*, depending on whether the initial ground state* $\Psi(0)$ *lies in the even- or odd-parity sector.*

In particular, if $H_{\mathsf{triv}} = -\frac{gi}{2}\sum_{j=1}^{L} c_{2j-1}c_{2j}$ is given by the second term in (3.12), we can apply the results of [Pfe70]: the gap at the phase transition is associated with the lowest energy mode (which is not protected by symmetry) and is given by $\lambda_2(H'_{I,g=1}) = 2\sin[\pi/(2L+1)]$. In other words, it is linearly decreasing in the system size L. Therefore, the total evolution time T only needs to grow polynomially in the system size L for Hamiltonian interpolation to accurately follow the ground state space at the phase transition. We conclude that translation-invariant Hamiltonian interpolation allows preparing the state $|g_0\rangle$ in a time T polynomial in the system size L and the desired approximation accuracy.

To achieve efficient preparation through Hamiltonian interpolation, one issue that must be taken into account is the effect of disorder (possibly in the form of a random site-dependent chemical potential). In the case where the system is already in the topologically ordered phase, a small amount of Hamiltonian disorder can enhance the zero temperature memory time of the Majorana chain Hamiltonian [BK12]. This 1D Anderson localization effect [And58], while boosting memory times, was also found to hinder the convergence to the topological ground space through Hamiltonian interpolation. Indeed, in [CFS07] it was found that the residual energy density $[E_{\mathrm{res}}(T)/L]_{\mathrm{av}} \propto 1/\ln^{3.4}(T)$ averaged over disorder realizations decreases only polylogarithmically with the Hamiltonian interpolation time. Such a slow convergence of the energy density indicates that in the presence of disorder, the time T required to accurately reach the ground space scales exponentially with the system size L. For this reason, translation-invariance (i.e., no disorder) is required for an efficient preparation, and this may be challenging in practice.

We emphasize that according to Lemma 3.4.1, the prepared state is largely independent of the choice of the initial Hamiltonian H_{triv} (amounting to a different choice of $\mathbf{V}$): we do not obtain a continuum of final states. As we will see below, this stability property appears in a similar form in other models. The parity operator (3.13), which should be thought of as a string-operator connecting the two ends of the wire, plays a particular role – it is essentially the effective Hamiltonian which determines the prepared ground state.

Indeed, the Schrieffer-Wolff-effective Hamiltonian can be computed *exactly* in this case, yielding

$$H_{\mathsf{eff}}(\epsilon) = \frac{E_0(\epsilon)}{2} I - \frac{\Delta(\epsilon)}{2} F \ , \tag{3.14}$$

where $E_0(\epsilon)$ is the ground state energy of $H_{\mathsf{top}} + \epsilon H_{\mathsf{triv}}$, and $\Delta(\epsilon) = E_1(\epsilon) - E_0(\epsilon)$ is the gap. Expression (3.14) can be computed based on the variational expression (3.55) for the Schrieffer-Wolff transformation, using the fact that the ground space is two-dimensional and spanned by two states belonging to the even- and odd-parity sector, respectively. Note that the form (3.14) can also be deduced (without the exact constants) from the easily verified fact (see e.g., Eq. (3.54)) that the Schrieffer-Wolff unitary U commutes with the fermionic parity operator F, and thus the same is true for $H_{\mathsf{eff}}(\epsilon)$. This expression illustrates that Conjecture 1 does not directly apply in the context of preserved quantities, as explained in Section 3.3.2: rather, it is necessary to know the parity of the initial state $\Psi(0)$ to identify the resulting final state $\Psi(T)$ in the interpolation process.

3.5 General anyon chains

In this section, we generalize the considerations related to the Majorana chain to more general anyonic systems. Specifically, we consider a 1-dimensional lattice of anyons with periodic boundary conditions. This choice retains many features from the Majorana chain such as locally conserved charges and topological degeneracy yet further elucidates some of the general properties involved in the perturbative lifting of the topological degeneracy.

In Section 3.5.1, we review the description of effective models for topologically ordered systems. A key feature of these models is the existence of a family $\{F_a\}_a$ of string-operators indexed by particle labels. Physically, the operators F_a correspond to the process of creating a particle-antiparticle

pair $(a, \bar{a})$, tunneling along the 1-dimensional (periodic) lattice, and subsequent fusion of the pair to the vacuum (see Section 3.5.1). These operators play a fundamental role in distinguishing different ground states.

In Section 3.5.2, we derive our main result concerning these models. We consider local translation-invariant perturbations to the Hamiltonian of such a model, and show that the effective Hamiltonian is a linear combination of string-operators, i.e.,

$$H_{\mathsf{eff}} \approx \sum_a f_a F_a \tag{3.15}$$

up to an irrelevant global energy shift. The coefficients $\{f_a\}_a$ are determined by the perturbation. They can be expressed in terms of a certain sum of diagrams, as we explain below. While not essential for our argument, translation-invariance allows us to simplify the parameter dependence when expressing the coefficients f_a and may also be important for avoiding the proliferation of small gaps.

We emphasize that the effective Hamiltonian has the form (3.15) independently of the choice of perturbation. The operators $\{F_a\}_a$ are mutually commuting, and thus have a distinguished simultaneous eigenbasis (we give explicit expressions for the latter in Section 3.5.1). The effective Hamiltonian (3.15) is therefore diagonal in a fixed basis irrespective of the considered perturbation. Together with the general reasoning for Conjecture 1, this suggests that Hamiltonian interpolation can only prepare a discrete family of different ground states in these anyonic systems.

In Section 3.6, we consider two-dimensional topologically ordered systems and find effective Hamiltonians analogous to (3.15). We will also show numerically that Hamiltonian interpolation indeed prepares corresponding ground states.

3.5.1 Background on anyon chains

The models we consider here describe effective degrees of freedom of a topologically ordered system. Concretely, we consider one-dimensional chains with periodic boundary conditions, where anyonic excitations may be created/destroyed on L sites, and may hop between neighboring sites. Topologically (that is, the language of topological quantum field theory), the system can be thought of as a torus with L punctures aligned along one fundamental cycle. Physically, this means that excitations are confined to move exclusively along this cycle (we will consider more general models in section 3.6). A well-known example of such a model is the Fibonacci golden chain [FTL$^+$07]. Variational methods for their study were developed in [PCB$^+$10, KB10], which also provide a detailed introduction to the necessary formalism. In this section, we establish notation for anyon models and review minimal background to make the rest of the paper self-contained.

Algebraic data of anyon models: modular tensor categories

Let us briefly describe the algebraic data defining an anyon model. The underlying mathematical object is a tensor category. This specifies among other things:

(i) A finite set of particle labels $\mathbb{A} = \{1, a, \ldots\}$ together with an involution $a \mapsto \bar{a}$ (called particle-anti-particle exchange/charge conjugation). There is a distinguished particle $1 = \bar{1}$ called the trivial or vacuum particle.

(ii) A collection of integers N_{ab}^c indexed by particle labels, specifying the so-called *fusion multiplicities* (as well as the fusion rules). For simplicity, we will only consider the multiplicity-free case, where $N_{ab}^c \in \{0, 1\}$ (this captures many models of interest). In this case, we will write $N_{ab}^c = \delta_{ab\bar{c}}$.

(iii) A 6-index tensor $F : \mathbb{A}^6 \to \mathbb{C}$ (indexed by particle labels) F_{cdf}^{abe} which is unitary with respect to the rightmost two indices (e, f) and can be interpreted as a change of basis for fusion trees.

(iv) A positive scalar d_a for every particle label a, called the *quantum dimension.*

(v) A unitary, symmetric matrix S_{ij} indexed by particle labels such that $S_{\bar{i}j} = \overline{S_{ij}}$.

(vi) A *topological phase* $e^{i\theta_j}$, $\theta_j \in \mathbb{R}$, associated with each particle j. We usually collect these into a diagonal matrix $T = \mathsf{diag}(\{e^{i\theta_j}\}_j)$; the latter describes the action of a twist in the mapping class group representation associated with the torus (see Section 3.6.2).

A list of the algebraic equations satisfied by these objects can be found e.g., in [LW05] (also see [NSS+08, LW05, Kit06, Wan10] for more details). Explicit examples of such tensor categories can also be found in [LW05], some of which we discuss in Section 3.6.3.

Here we mention just a few which will be important in what follows: the fusion rules δ_{ijk} are symmetric under permutations of (i, j, k). They satisfy

$$\sum_m \delta_{ij\bar{m}}\delta_{mk\bar{\ell}} = \sum_m \delta_{jk\bar{m}}\delta_{im\bar{\ell}}$$

which expresses the fact that fusion (as explained below) is associative, as well as

$$\delta_{i\bar{j}1} = \delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise .} \end{cases} \tag{3.16}$$

Some of the entries of the tensor F are determined by the fusion rules and the quantum dimensions, that is,

$$F^{i\bar{i}1}_{jjk} = \sqrt{\frac{d_k}{d_i d_j}}\delta_{ijk} \ . \tag{3.17}$$

Another important property is the Verlinde formula

$$\delta_{bc\bar{d}} = N^d_{bc} = \sum_a \frac{S_{ba}S_{ca}S_{\bar{d}a}}{S_{1a}} \ , \tag{3.18}$$

which is often summarized by stating that S "diagonalizes the fusion rules".

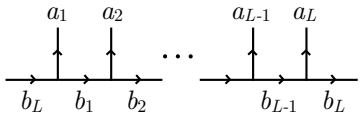
The Hilbert space

The Hilbert space of a one-dimensional periodic chain of L anyons is the space associated by a TQFT to a torus with punctures. It has the form

$$\mathcal{H} \cong \bigoplus_{\substack{a_1,\dots,a_L \\ b_0,\dots,b_L}} V^{a_1 b_1}_{b_0} \otimes V^{a_2 b_2}_{b_1} \otimes \cdots \otimes V^{a_L b_L}_{b_{L-1}} \ ,$$

where the indices a_j, b_k are particle labels, V^{ab}_c are the associated finite-dimensional fusion spaces and we identify $b_0 = b_L$. The latter have dimension $\dim V^{ab}_c = N^c_{ab}$. Again, we will focus on the multiplicity-free case where $N^c_{ab} = \delta_{ab\bar{c}} \in \{0, 1\}$. In this case, we can give an orthonormal basis $\{|\vec{a}, \vec{b}\rangle\}_{(\vec{a},\vec{b})}$ of $\mathcal{H}$ in terms of 'fusion-tree' diagrams, i.e,

$$|\vec{a}, \vec{b}\rangle = \frac{1}{(\prod_j d_{a_j})^{1/4}}$$

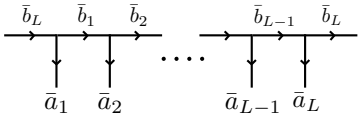


(3.19)

where $\vec{a} = (a_1, \dots, a_L)$ and $\vec{b} = (b_1, \dots, b_L)$ have to satisfy the fusion rules at each vertex, i.e., $\dim V^{a_j b_j}_{b_{j-1}} = \delta_{a_j b_j \bar{b}_{j-1}} = 1$ for all $j = 1, \dots, L$.

The prefactor in the definition of the state (3.19) involves the quantum dimensions of the particles, and is chosen in such a way that $\{|\vec{a}, \vec{b}\rangle\}$ is an orthonormal basis with respect to the inner product defined in terms of the isotopy-invariant calculus of diagrams: the adjoint of $|\vec{a}, \vec{b}\rangle$ is represented as

$$\langle\vec{a}, \vec{b}| = \frac{1}{(\prod_j d_{a_j})^{1/4}}$$

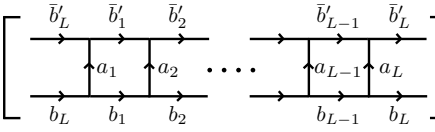


.

Inner products and diagramatic reduction rules

Inner products are evaluated by composing diagrams and then reducing, i.e.,

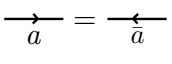
$$\langle \vec{a}', \vec{b}' | \vec{a}, \vec{b} \rangle = (\prod_j d_{a_j})^{-1/2} \prod_{j=1}^{L} \delta_{a_j, a'_j}$$



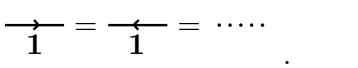
(3.20)

where $[\cdot]_{\mathsf{vac}}$ is the coefficient of the empty diagram when reducing. Reduction is defined in terms of certain local moves. These include

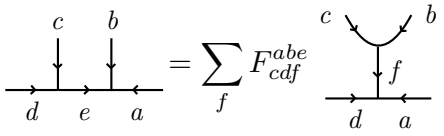
(i) reversal of arrows (together particle-antiparticle involution $a \mapsto \bar{a}$)



(ii) (arbitrary) insertions/removals of lines labeled by the trivial particle 1. Since $\bar{1} = 1$, such lines are not directed, and will often be represented by dotted lines or omitted altogether,



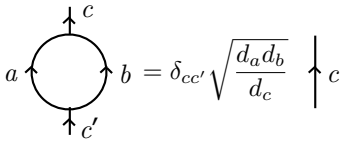
(iii) application of the F-matrix in the form



(3.21)

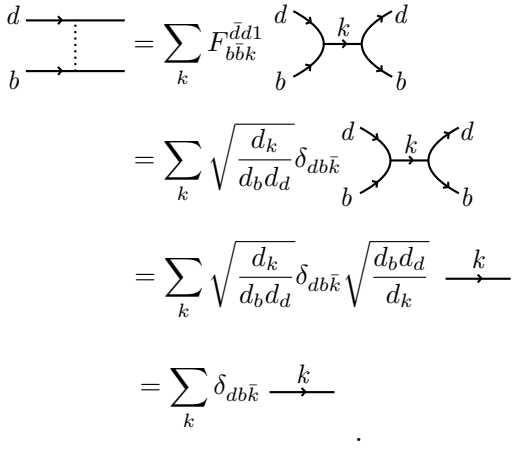
which leads to a formal linear combination of diagrams where subgraphs are replaced locally by the figure on the rhs.

(iv) removal of "bubbles" by the substitution rule



(3.22)

These reduction moves can be applied iteratively in arbitrary order to yield superpositions of diagrams. An important example of this computation is the following:



(3.23)

The series of steps first makes use of an F-move (3.21), followed by Eq. (3.17) as well as (3.22). Together with property (3.16) and evaluation of the inner product (3.20), this particular calculation shows that the flux-eigenstates (3.27) are mutually orthogonal. We refer to [LW05] for more details.

Local operators

Operators are also defined by diagrams, and are applied to vectors/multiplied by stacking (attaching) diagrams on top of the latter. Expressions vanish unless all attachment points have identical direction and labels. Here we concentrate on 1- and 2-local operators, although the generalization is straightforward (see [KB10, Bon09]).

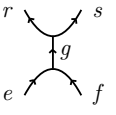
A single-site operator $\hat{H}$ is determined by coefficients $\{\epsilon_a\}_a$ and represented at

$\hat{H} = \sum_a \epsilon_a$.

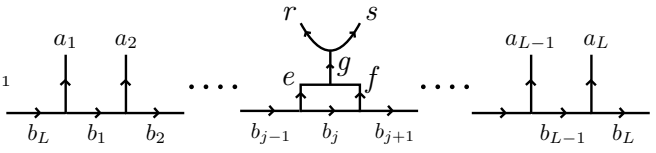
It acts diagonally in the fusion tree basis, i.e., writing H_j for the operator $\hat{H}$ applied to site j, we have

$$H_j|\vec{a},\vec{b}\rangle = \epsilon_{a_j}|\vec{a},\vec{b}\rangle \ .$$

A two-site operator $\hat{V}$ acting on two neighboring sites is determined by a tensor $\{\alpha^{rs}_{efg}\}_{r,s,e,f,g}$ (where the labels have to satisfy appropriate fusion rules) via the linear combinations of diagrams

$\hat{V} = \sum_{\substack{e,f,g\\ r,s}} \alpha^{rs}_{efg}$  . (3.24)

When applied to sites j and $j+1$ it acts as

$V_{j,j+1}|\vec{a},\vec{b}\rangle = \sum_{\substack{e,f,g\\ r,s}} \alpha^{rs}_{efg}\delta_{e,a_j}\delta_{f,a_{j+1}}$  ,

where the rhs. specifies a vector in $\mathcal{H}$ in terms of the reduction rules. It will be convenient in the following to distinguish between linear combinations of the form (3.24) and operators which are scalar multiplies of a single diagram (i.e., with only one non-zero coefficient α^{rs}_{efg}). We call the latter kind of two-site operator *elementary*.

We can classify the terms appearing in (3.24) according to the different physical processes they represent: in particular, we have pair creation- and annihilation operators

$\hat{V}^{\mathsf{C}}(a) =$  and $\hat{V}^{\mathsf{A}}(a) = (\hat{V}^{\mathsf{C}}(a))^\dagger =$  ,

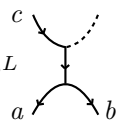
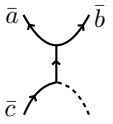
simultaneous annihilation- and creation operators

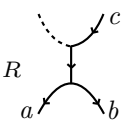
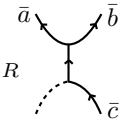
$$\hat{V}^{\mathsf{CA}}(a,b) = \hat{V}^{\mathsf{C}}(a)\hat{V}^{\mathsf{A}}(b)$$

left- and right-moving 'propagation' terms

$\hat{V}^{\mathsf{L}}(a) =$  and $\hat{V}^{\mathsf{R}}(a) = (\hat{V}^{\mathsf{L}}(a))^\dagger =$ 

as well as more general fusion operators such as e.g.,

$\hat{V}_{a,b,c} = \sum_{a,b,c} \Phi^c_{ab,L}$  $+\bar{\Phi}^c_{ab,L}$ 

$+\Phi^c_{ab,R}$  $+\bar{\Phi}^c_{ab,R}$  ,

(We are intentionally writing down a linear combination here.) Note that a general operator of the form (3.24) also involves braiding processes since  can be resolved to diagrams of the form  using the R-matrix (another object specified by the tensor category). We will consider composite processes composed of such two-local operators in Section 3.5.1.

Ground states of anyonic chains

We will consider translation-invariant Hamiltonians $H_0 = \sum_j \hat{H}_j$ with local terms of the form

$$\hat{H} = \sum_a \epsilon_a \, \uparrow a \qquad \text{with } \epsilon_a > 0 \text{ for } a \neq 1 \text{ and } \epsilon_1 = 0 \; . \tag{3.25}$$

Such a Hamiltonian H_0 corresponds to an on-site potential for anyonic excitations, where a particle of type a has associated energy ϵ_a independently of the site j. We denote the projection onto the ground space of this Hamiltonian by P_0. This is the space

$$P_0\mathcal{H} = \mathsf{span}\{|\vec{1}, b\cdot\vec{1}\rangle \mid b \text{ particle label}\} \tag{3.26}$$

where $\vec{1} = (1,\ldots,1)$ and $b\cdot\vec{1} = (b,\ldots,b)$. In other words, the ground space of H_0 is degenerate, with degeneracy equal to the number of particle labels.

It will be convenient to use the basis $\{|b\rangle\}_b$ of the ground space consisting of the 'flux' eigenstates

$$|b\rangle = |\vec{1}, b\cdot\vec{1}\rangle \; . \tag{3.27}$$

In addition, we can define a dual basis $\{|b'\rangle\}_b$ of the ground space using the S-matrix. The two bases are related by

$$|a'\rangle = \sum_b \overline{S_{ba}}|b\rangle \tag{3.28}$$

for all particle labels a, b.

As we discuss in Section 3.6.3, in the case of two-dimensional systems, the dual basis (3.28) is simply the basis of flux eigenstates with respect to a 'conjugate' cycle. While this interpretation does not directly apply in this 1-dimensional context, the basis $\{|a'\rangle\}_a$ is nevertheless well-defined and important (see Eq. (3.30)).

Non-local string-operators

In the following, certain non-local operators, so-called *string-operators*, will play a special role. Strictly speaking, these are only defined on the subspace (3.26). However, we will see in Section 3.5.2 that they arise naturally from certain non-local operators.

The string-operators $\{F_a\}_a$ are indexed by particle labels a. In terms of the basis (3.27) of the ground space $P_0\mathcal{H}$ of H_0, the action of F_a is given in terms of the fusion rules as

$$F_a|b\rangle = \sum_c N_{ab}^c |c\rangle = \sum_c \delta_{ab\bar{c}}|c\rangle \; . \tag{3.29}$$

[6] The operator F_a has the interpretation of creating a particle-antiparticle pair $(a, \bar{a})$, moving one around the torus, and then fusing to vacuum. For later reference, we show that every string-operator F_a is diagonal in the dual basis $\{|a'\rangle\}$. Explicitly, we have

$$F_b P_0 = \sum_a \frac{S_{ba}}{S_{1a}} |a'\rangle\langle a'| \; . \tag{3.30}$$

[6] In fact, the operators $\{F_a\}_a$ form a representation of the Verlinde algebra, although we will not use this fact here.

Proof. We first expand P_0 into its span and F_b according to eq. (3.29), followed by an expansion of N_{bc}^d through the Verlinde formula (3.18). Finally, we use the unitarity and symmetry of S to transform bra and ket factors into the dual basis given by Eq. (3.28)

$$F_b P_0 = \sum_{c,d} N_{bc}^d |d\rangle\langle c| = \sum_a \frac{S_{ba}}{S_{1a}} \sum_{c,d} S_{ca} S_{\bar{d}a} |d\rangle\langle c| = \sum_a \frac{S_{ba}}{S_{1a}} |a'\rangle\langle a'| \ .$$

□

Products of local operators and their logical action

Operators preserving the ground space $P_0\mathcal{H}$ (cf. (3.27)) are called *logical operators*. As discussed in Section 3.5.1, string-operators $\{F_a\}$ are an example of such logical operators. Clearly, because they can simultaneously be diagonalized (cf. (3.30)), they do not generate the full algebra of logical operators. Nevertheless, they span the set of logical operators that are generated by geometrically local physical processes preserving the space $P_0\mathcal{H}$.

That is, if $O = \sum_j \prod_k V_{j,k}$ is a linear combinations of products of local operators $V_{j,k}$, then its restriction to the ground space is of the form

$$P_0 O P_0 = \sum_a o_a F_a \ , \tag{3.31}$$

i.e., it is a linear combination of string operators (with some coefficients o_a). Eq. (3.31) can be interpreted as an emergent superselection rule for topological charge, which can be seen as the generalization of the parity superselection observed for the Majorana chain. It follows directly from the diagrammatic formalism for local operators.

To illustrate this point (and motivate the following computation), let us consider three examples of such operators, shown in Figures 3.1a, 3.1c and 3.1b.

$O_1 = \hat{V}^{\mathsf{A}}_{j-1,j}(a)\hat{V}^{\mathsf{L}}_{j+1,j+2}(a)\hat{V}^{\mathsf{R}}_{j+1,j+2}(a)\hat{V}^{\mathsf{C}}(a)_{j,j+1}$**:** This processes has trivial action on the ground space: it is entirely local. It has action $P_0 O_1 P_0 = d_a P_0$, where the proportionality constant d_a results from Eq. (3.22).

$O_2 = \hat{V}^{\mathsf{L}}_{j-1,j}(\bar{a})\hat{V}^{\mathsf{R}}_{j,j+1}(a)\hat{V}^{\mathsf{C}}_{j,j+1}(a)$**:** This process creates a particles anti-particle pair $(a, \bar{a})$ and further separates these particles. Since the operator maps ground states to excited states, we have $P_0 O_3 P_0 = 0$.

$O_3 = \hat{V}^{\mathsf{A}}(\bar{a})_{N,1}\hat{V}^{\mathsf{R}}(a)_{N-1,N} \ldots \hat{V}^{\mathsf{R}}(a)_{3,4}\hat{V}^{\mathsf{R}}(a)_{2,3}\hat{V}^{\mathsf{C}}(a)_{1,2}$**:** This process involves the creation of a pair of particles $(a, \bar{a})$, with subsequent propagation and annihilation. Its logical action is $P_0 O_2 P_0 = F_a$ is given by the string-operator F_a, by a computation similar to that of (3.23).

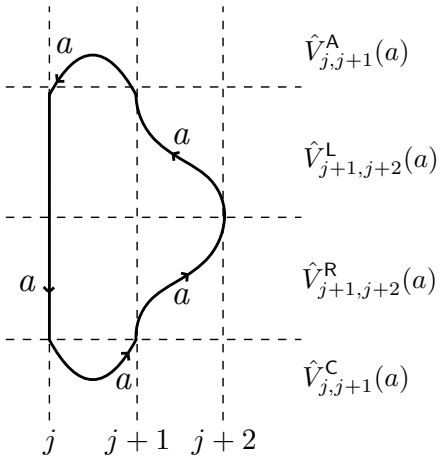
3.5.2 Perturbation theory for an effective anyon model

In this section, we consider a 1-dimensional translation-invariant system of anyons described by the Hamiltonian H_0 introduced in (3.25). We further consider a translation-invariant two-local perturbation $V = \sum_j \hat{V}_{j,j+1}$ with local terms $\hat{V}_{j,j+1}$ of the form (3.24) given by

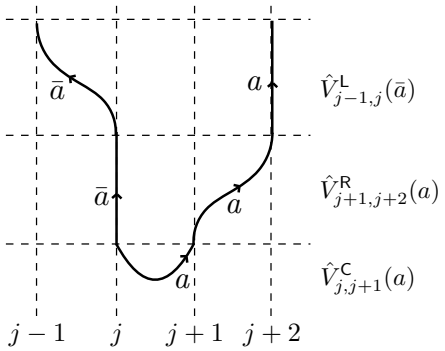
$$\hat{V} = \sum_a \left(\gamma_a V^C(a) + \overline{\gamma_a} V^A(a)\right) + \sum_a \left(\tau_a V^L(a) + \overline{\tau_a} V^R(a)\right) + V_R \ , \tag{3.32}$$

where V_R collects all other two-anyon processes (it will turn out that in lowest order perturbation theory, only creation and propagation are relevant). The choice of complex conjugate pairs of parameters ensures that the perturbation is self-adjoint. We may think of γ_a as the 'creation ampitude', τ_a as the 'propagation amplitude', and ϵ_a as the energy of particle a.

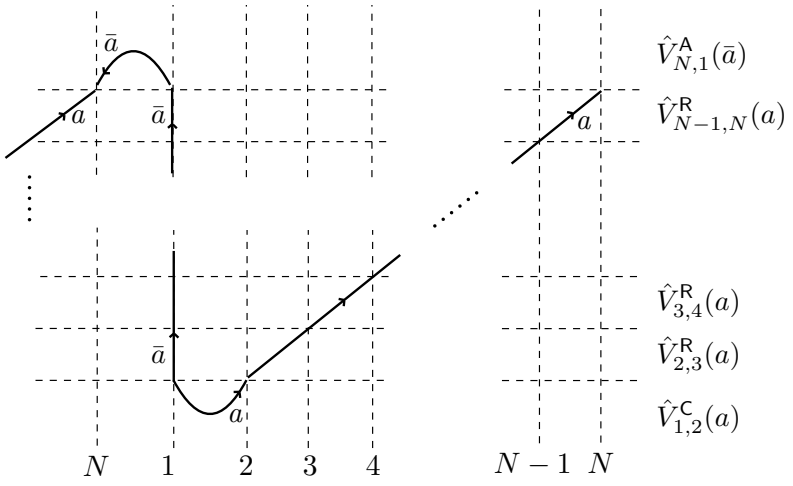
We now compute the form of the effective Schrieffer-Wolff-Hamiltonian. Our main result is the following:



(a) The operator $O_1 = \hat{V}^{\mathsf{A}}_{j,j+1}(a)\hat{V}^{\mathsf{L}}_{j+1,j+2}(a)\hat{V}^{\mathsf{R}}_{j+1,j+2}(a)\hat{V}^{\mathsf{C}}(a)_{j,j+1}$ corresponds to a process where a particle pair $(a, \bar{a})$ is created, there is some propagation, and the particles fuse subsequently. This has trivial action on the ground space, i.e., $P_0 O P_0 = d_a P_0$ is proportional to the identity.



(b) The process described by the operator $O_2 = \hat{V}^{\mathsf{L}}_{j-1,j}(\bar{a})\hat{V}^{\mathsf{R}}_{j,j+1}(a)\hat{V}^{\mathsf{C}}_{j,j+1}(a)$ leaves behind excitations, hence $P_0 O_3 P_0 = 0$.



(c) The operator $O_3 = V^{\mathsf{A}}(\bar{a})_{N,1} \cdots V^{\mathsf{R}}(a)_{N-1,N} .... V^{\mathsf{R}}(a)_{3,4} V^{\mathsf{R}}(a)_{2,3} V^{\mathsf{C}}(a)_{1,2}$ corresponds to a process where a pair $(a, \bar{a})$ of particles is created, and they propagate all the way around the chain before annihilating. Its action on the ground space is given by the string-operator $P_0 O_2 P_0 = F_a$.

Figure 3.1: This figure illustrates different processes in the diagrammatic formalism. Each process corresponds to an operator and is a product of elementary processes (diagrams). Ground space matrix elements vanish if the process leaves behind excitations (corresponding to endpoints of open strings).

Lemma 3.5.1 (Effective Hamiltonians for 1-dimensional anyon chains)**.** *Consider $H_0 + \epsilon V$, with the perturbation V as described. Let P_0 be the projection onto the ground space of H_0. Then the L-th order effective Hamiltonian has the form*

$$H_{\mathsf{eff}}^{(L)}(\epsilon) = \sum_a f_L(\epsilon_a, \gamma_a, \tau_a) F_a + cP_0 \ , \tag{3.33}$$

for some constant $c \in \mathbb{R}$, and some function f_L which is independent of the particle label a and is a homogeneous polynomial of degree L in γ_a and τ_a.

Clearly, the form Eq. (3.33) of the effective Hamiltonian is consistent with the topological superselection rule (3.31). However, Eq. (3.33) provides additional information: for example, the coefficient of the string-operator F_a only depends on the energy ϵ_a of anyon a, as well as its creation/annihilation (γ_a respectively $\overline{\gamma_a}$) and propagation (τ_a) amplitudes. There is no dependence on particles distinct from a (and corresponding braiding processes). Such terms only enter in higher orders of the perturbative series. This can be thought of as a rigorous derivation of the tunneling amplitude for a particle in the weak perturbation limit. We note that due to f_L being homogeneous of degree L, the dominant tunneling process will be highly sensitive to the perturbation strengths associated to different anyon labels a for large system sizes L. In the absence of a symmetry or fine tuning, it should be possible to order the terms $f_L(\epsilon_a, \gamma_a, \tau_a)$ by absolute value, with different orders of magnitude being expected for each term (see Section 3.6.1 for further discussion).

Proof. It is easy to check that the conditions of Theorem 3.3.2 are satisfied with L equal to the length of the chain. Indeed, $(L-1)$-local terms have trivial action on the ground space as discussed in Section 3.5.1. It thus suffices to consider expressions of the form

$$P_0 (VG)^{L-1} V P_0$$

involving L factors of V. Inserting the definition (3.32) of V, and diagrammatically expanding each term as in Section 3.5.1, we are left with a linear combination of terms of the form

$$P_0 V_{\alpha_1} G V_{\alpha_2} G V_{\alpha_3} \cdots G V_{\alpha_L} P_0 \ ,$$

where V_{α_j} is a local operator given by an elementary (two-anyon) diagram (not a linear combination). Since such operators V_{α_j} map eigenstates of H_0 to eigenstates, and the energies of excited states reached from the ground space by applying such operators is independent of the ground state considered, each operator G merely adds a scalar, i.e., we have

$$P_0 V_{\alpha_1} G V_{\alpha_2} G V_{\alpha_3} \cdots G V_{\alpha_L} P_0 = \theta(V_{\alpha_1}, \ldots, V_{\alpha_L}) \cdot P_0 V_{\alpha_1} V_{\alpha_2} V_{\alpha_3} \cdots V_{\alpha_L} P_0$$

for some constant θ depending on the perturbations $\{V_{\alpha_j}\}$. But the rhs. of this equation is a product of local operators as considered in Section 3.5.1. According to the expression (3.31), this is a linear combination of string-operators, i.e.,

$$P_0 V_{\alpha_1} V_{\alpha_2} V_{\alpha_3} \cdots V_{\alpha_L} P_0 = \sum_a o_a F_a \ .$$

Furthermore, since each V_{α_j} is an elementary two-local operator, and we consider only products of length L, the only terms $P_0 V_{\alpha_1} V_{\alpha_2} V_{\alpha_3} \cdots V_{\alpha_L} P_0$ that have non-trivial action on the ground space are those associated with processes where a single particle (say of type a) winds around the whole chain. We will call such a process *topologically non-trivial*. Its action on the ground space is given by a single string-operator F_a.

In summary (rearranging the sum), we conclude that the L-th order effective Hamiltonian has the form (3.33), where the coefficient $f_L(\epsilon_a, \gamma_a, \tau_a)$ has the form

$$f_L(\epsilon_a, \gamma_a, \tau_a) = \sum_{(V_{\alpha_1}, \ldots, V_{\alpha_L}) \in \Theta_a} \theta(V_{\alpha_1}, \ldots, V_{\alpha_L}) \nu(V_{\alpha_1}, \ldots, V_{\alpha_L}) \ ,$$

and where the sum is over the set

$$\Theta_a = \{(V_{\alpha_1}, \ldots, V_{\alpha_L}) \mid P_0 V_{\alpha_1} \cdots V_{\alpha_L} P_0 \in \mathbb{C} P_0\}$$

of all length-L-topologically non-trivial processes (consisting of elementary terms) involving particle a. The coefficient $\nu(V_{\alpha_1}, \ldots, V_{\alpha_L})$ is defined by $P_0 V_{\alpha_1} \cdots V_{\alpha_L} P_0 = \nu(V_{\alpha_1}, \ldots, V_{\alpha_L}) F_a$. Furthermore, $\nu(V_{\alpha_1}, \ldots, V_{\alpha_L})$ can only be non-zero when all L operators V_{α_j} are either pair creation/anihilation or hopping terms involving the particle a. This implies the claim. □

3.6 2D topological quantum field theories

As discussed in Section 3.4, adding a local perturbation to a Majorana chain leads to an effective Hamiltonian given by the parity (string)-operator. Similarly, in the case of a general anyon chain (discussed in Section 3.5), the effective Hamiltonian is a linear combination of string-operators F_a, associated with different particle labels a. Here we generalize these considerations to arbitrary systems described by a 2-dimensional topological quantum field theory (TQFT) and subsequently specialize to microscopic models, including the toric code and the Levin-Wen string-net models [LW05].

Briefly, a TQFT associates a "ground space" $\mathcal{H}_\Sigma$ to a two-dimensional surface Σ – this is e.g., the ground space of a microscopic model of spins embedded in Σ with geometrically local interactions given by some Hamiltonian H_0 (see Section 3.6.3). In other words, $\mathcal{H}_\Sigma \subset \mathcal{H}_{phys,\Sigma}$ is generally a subspace of a certain space $\mathcal{H}_{phys,\Sigma}$ of physical degrees of freedom embedded in Σ. The system has localized excitations (anyons) with (generally) non-abelian exchange statistics. In particular, there are well-defined physical processes involving creation, propagation, braiding and annihilation of anyons, with associated operators as in the case of 1-dimensional anyon chains (see Section 3.5). Contrary to the latter, however, the particles are not constrained to move along a 1-dimensional chain only, but may move arbitrarily on the surface Σ. Nevertheless, the description of these processes is analogous to the case of spin chains, except for the addition of an extra spatial dimension. For example, this means that local operators acting on a region $\mathcal{R} \subset \Sigma$ are now represented by a linear combination of string-nets (directed trivalent graphs with labels satisfying the fusion rules) embedded in $\mathcal{R} \times [0, 1]$. We refer to e.g., [FKLW03] for more examples of this representation.

As before, there are distinguished ground-space-to-ground-space (or "vacuum-to-vacuum") processes which play a fundamental role. These are processes where a particle-anti-particle pair $(a, \bar{a})$ is created, and the particles fuse after some propagation (tunneling), i.e., after tracing out a closed loop C on Σ. Non-trivial logical operators must necessarily include topologically non-trivial loops C on Σ in their support (the spatial region in which they are physically realized). In particular, for any such loop C, there is a collection $\{F_a(C)\}_a$ of string-operators associated with different particle labels. More precisely, a loop is a map $C : [0, 1] \to \Sigma$ satisfying $C(0) = C(1)$. Reversing direction of the loop gives a new loop $\bar{C}(t) := C(1 - t)$, and this is equivalent to interchanging particle- and antiparticle labels: we have the identity $F_a(C) = F_{\bar{a}}(\bar{C})$. In Section 3.6.2, we state some general properties of the string-operators $\{F_a(C)\}_a$, and, in particular, explain how to express them in suitable bases of the ground space.

3.6.1 Perturbation theory for Hamiltonians corresponding to a TQFT

In general, the anyon model associated with a TQFT is emergent from a microscopic spin Hamiltonian H_0. The anyon notion of site, as discussed in Section 3.5, does not necessarily coincide with the spin notion of site associated with the microscopic spin model. Nevertheless, the following statements are true:

(i) any non-trivial logical operator must include at least one non-contractible loop in its support.

(ii) given a perturbation V consisting of geometrically local operators, there exists some minimum integer L such that H_0, V satisfy the topologically ordered condition with parameter L.

In general, the value of L will depend on the length of the shortest non-contractible loop(s), and the resulting effective Hamiltonian will be of the form

$$H_{\mathsf{eff}}^{(L)}(\epsilon) = \epsilon^L \sum_{a,C:|C|=L} f_L(a,C)F_a(C) + c(\epsilon)P_0 \ , \tag{3.34}$$

where the dependence on H_0 and the coefficients in V has been left implicit. The sum is over all non-trivial loops C of length L (where length is defined in terms of the spin model), as well as all particle labels a.

Computing the coefficients $\{f_L(a,C)\}$ may be challenging in general. Here we discuss a special case, where anyon processes associated with a single particle a (respectively its antiparticle $\bar{a}$) are dominant (compared to processes involving other particles). That is, let us assume that we have a translation-invariant perturbation V of the form

$$V = \sum_{(j,j')} \left(\hat{V}_{j,j'}^{(1)} + \eta V_{j,j'}^{(2)} \right) \ ,$$

where the sum is over all pairs (j,j') of nearest-neighbor (anyonic) sites, and $\hat{V}_{j,j'}^{(1)} = \hat{V}^{(1)}$ and $\hat{V}_{j,j'}^{(2)} = \hat{V}^{(2)}$ are both 1- and 2-local operators on the same anyon site lattice – this is a straightforward generalization of anyon chains to 2D. Our specialization consists in the assumption that all local creation, propagation and annihilation processes constituting the operator $\hat{V}_{j,j'}^{(1)} = \hat{V}^{(1)}$ only correspond to a single anyon type a (and $\bar{a}$), and that these processes are dominant in the sense that the remaining terms satisfy $\|\eta\hat{V}^{(2)}\| \ll \|\hat{V}^{(1)}\|$. In the limit $\eta \to 0$, perturbation theory in this model only involves the particles $(a,\bar{a})$.

Assuming that the shortest non-contractible loops have length L in this anyonic lattice, we claim that

$$H_{\mathsf{eff}}^{(L)}(\epsilon) = \epsilon^L \left(\sum_{C:|C|=L} f_L(a,C)F_a(C) + \eta^L G_{\mathsf{eff}}^{(L)} \right) + c(\epsilon)P_0 \ , \tag{3.35}$$

where $G_{\mathsf{eff}}^{(L)}$ is an effective Hamiltonian with the same form as $H_{\mathsf{eff}}^{(L)}(\epsilon)$, but only contains string operators $F_b(C)$ with $b \neq a$. The reason is that in order to generate a string operator $F_b(C)$ in L steps (i.e., at L-th order in perturbation theory), we need to apply local operators corresponding to anyon b L times, as discussed in Lemma 3.5.1. Such local operators can only be found in ηV_2, therefore we obtain the coefficient η^L of $G_{\mathsf{eff}}^{(L)}$. Thus if we fix the system size and slowly increase η from 0, the (relative) change of the total effective Hamiltonian is exponentially small with respect to L. This implies that the ground state of the effective Hamiltonian is stable when η is in a neighbourhood of 0. We will see in Section 3.7 that the final states of Hamiltonian interpolation are indeed stable in some regions of initial Hamiltonians. The above discussion can be viewed as a partial explanation[7] for this phenomenon.

3.6.2 String-operators, flux bases and the mapping class group

In the following, we explain how to compute effective Hamiltonians of the form (3.35) in the case where the perturbation is isotropic, resulting in identical coefficients $f_L(a,C) = f_L(a,C')$ for all loops C of identical length. This will be guaranteed by symmetries. We give explicit examples in Section 3.7.

For this purpose, we need a more detailed description of the action of string-operators on the ground space. Consider a fixed (directed) loop $C : [0,1] \to \Sigma$ embedded in the surface Σ. The process of creating a particle-anti-particle pair $(a,\bar{a})$, then propagating a along C, and subsequently fusing with $\bar{a}$ defines an operator $F_a(C)$ which preserves the ground space $\mathcal{H}_\Sigma$. The family of operators $\{F_a(C)\}_a$ is mutually commuting and defines a representation of the Verlinde algebra. It

[7] Note that in the cases we consider in Section 3.7, $\hat{V}^{(1)}$ and $\hat{V}^{(2)}$ often do not live on the same anyon site lattice.

is sometimes convenient to consider the associated (images of the) idempotents, which are explicitly given by (as a consequence of the Verlinde formula (3.18))

$$P_a(C) = S_{1a} \sum_b \overline{S_{ba}} F_b(C) \ .$$

The operators $P_a(C)$ are mutually orthogonal projections $P_a(C)P_b(C) = \delta_{ab}P_a(C)$. The inverse relationship (using the unitarity of S) reads

$$F_b(C) = \sum_a \frac{S_{ba}}{S_{1a}} P_a(C) \tag{3.36}$$

and is the generalization of (3.30): indeed, specializing to the case where Σ is the torus (this will be our main example of interest), and C is a fundamental loop, the operators $P_a(C)$ are rank-one projections (when restricted to the ground space), and determine (up to phases) an orthonormal basis of $\mathcal{B}_C = \{|a_C\rangle\}_a$ of $\mathcal{H}_\Sigma$ by $P_a(C) = |a_C\rangle\langle a_C|$. In physics language, the state $|a_C\rangle$ has "flux a" through the loop C. (More generally, one may define "fusion-tree" basis for higher-genus surfaces Σ by considering certain collections of loops and the associated idempotents, see e.g., [KKR10]. However, we will focus on the torus for simplicity.)

Consider now a pair of distinct loops C and C'. Both families $\{F_a(C)\}_a$ and $\{F_a(C')\}_a$ of operators act on the ground space, and it is natural to ask how they are related. There is a simple relationship between these operators if $C' = \vartheta(C)$ is the image of C under an element $\vartheta : \Sigma \to \Sigma$ of the mapping class group MCG_Σ of Σ (i.e., the group of orientation-preserving diffeomorphisms of the surface): The TQFT defines a projective unitary representation $V : \mathsf{MCG}_\Sigma \to \mathsf{U}(\mathcal{H}_\Sigma)$ of this group on $\mathcal{H}_\Sigma$, and we have

$$F_a(C') = V(\vartheta)F_a(C)V(\vartheta)^\dagger \quad \text{for all } a \text{ if } C' = \vartheta(C) \ .$$

In general, while the topology of the manifold is invariant under the mapping class group, the specific lattice realization may not be. For this reason, if we desire to lift the representation V to the full Hilbert space $\mathcal{H}_\Sigma \supset \mathcal{H}_{phys,\Sigma}$, such that the resulting projective unitary representation preserves the microscopic Hamiltonian H_0 under conjugation, we may need to restrict to a finite subgroup of the mapping class group MCG_Σ. If the lattice has sufficient symmetry, such as for translation-invariant square or rhombic lattices, one may exploit these symmetries to make further conclusions about the resulting effective Hamiltonians.

String-operators and the mapping class group for the torus

For the torus, the mapping class group MCG_Σ is the group $SL(2,\mathbb{Z})$. To specify how a group element maps the torus to itself, it is convenient to parametrize the latter as follows: we fix complex numbers (e_1, e_2) and identify points z in the complex plane according to

$$z \equiv z + n_1 e_1 + n_2 e_2 \qquad \text{for } n_1, n_2 \in \mathbb{Z} \ .$$

In other words, (e_1, e_2) defines a lattice in $\mathbb{C}$, whose unit cell is the torus (with opposite sides identified). A group element $A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \in SL(2,\mathbb{Z})$ then defines parameters (e_1', e_2') by

$$\begin{aligned} e_1' &= ae_1 + be_2 \\ e_2' &= ce_1 + de_2 \ , \end{aligned}$$

which a priori appear to be associated with a new torus. However, the constraint that $A \in SL(2,\mathbb{Z})$ ensures that (e_1', e_2') and (e_1, e_2) both define the same lattice, and this therefore defines a map from the torus to itself: The action of A is given by $\alpha e_1 + \beta e_2 \mapsto \alpha e_1' + \beta e_2'$ for $\alpha, \beta \in \mathbb{R}$, i.e., it is simply a linear map determined by A.

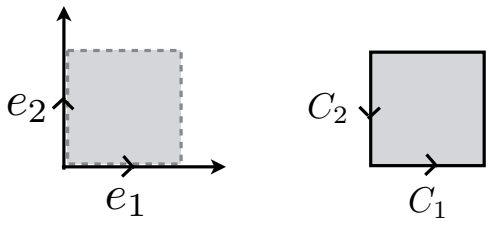


Figure 3.1: Minimal loops on the square torus

The group $SL(2,\mathbb{Z}) = \langle t, s\rangle$ is generated by the two elements

$$t = \text{Dehn twist} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \qquad \text{and} \qquad \pi/2 \text{ rotation} \quad s = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \tag{3.37}$$

which are equivalent to the Möbius transformations $\tau \mapsto \tau + 1$ and $\tau \mapsto -1/\tau$. Clearly, t fixes e_1 and hence the loop $C : t \mapsto C(t) = te_1$, $t \in [0,1]$ on the torus (this loop is one of the fundamental cycles). The matrices representing the unitaries $V(t)$ and $V(s)$ in the basis $\mathcal{B}_C = \{|a_C\rangle\}_a$ of $\mathcal{H}_\Sigma$ (where $|a_C\rangle$ is an eigenstate of $P_a(C) = |a_C\rangle\langle a_C|$) are denoted T and S, respectively. These matrices are given by the modular tensor category: T is a diagonal matrix with $T_{aa} = e^{i\theta_a}$ (where θ_a is the topological phase of particle a), whereas S is the usual S-matrix. This defines the mapping class group representation on the Hilbert space $\mathcal{H}_\Sigma$ associated with the torus Σ.

In the following, we compute explicit relationships between string-operators of minimal length. We consider two cases: a square torus and a rhombic torus. This allows us to express terms such as those appearing in Eq. (3.34) in a fixed basis.

Square torus. Here we have

$$e_1 = 1 \qquad \text{and} \qquad e_2 = i \ .$$

There are (up to translations) two loops of minimal length,

$$\begin{aligned} C_1(t) &= te_1 \\ C_2(t) &= (1-t)e_2, \end{aligned}$$

which may be traversed in either of two directions namely for $t \in [0,1]$, see Fig. 3.1. Since $se_1 = -e_2$ and $se_2 = e_1$, we conclude that

$$C_2(t) = s(C_1(t)) \qquad \overline{C_1}(t) = s^2(C_1(t)) \qquad \overline{C_2}(t) = s^3(C_1(t)) \qquad C_1(t) = s^4(C_1(t))$$

In particular, expressed in the basis $\mathcal{B}_{\mathcal{C}_1}$, we have

$$\sum_{j=1,2} \big(F_a(C_j) + F_a(\overline{C_j})\big) = \sum_{j=0}^{3} S^j F_a(C_1) S^{-j}. \tag{3.38}$$

Thus, when the lattice and Hamiltonian H_0 obey a $\pi/2$ rotation symmetry, the effective perturbation Hamiltonian will be proportional to (3.38). This is the case for the toric code on a square lattice.

Rhombic torus. We set

$$e_1 = 1 \qquad \text{and} \qquad e_2 = \cos(2\pi/6) + i\sin(2\pi/6) \ .$$

Minimal loops of interest are shown in Fig. 3.2 and can be defined as

$$\begin{aligned} C_1(t) &= te_1 \\ C_2(t) &= e_1 + t(e_2 - e_1) \\ C_3(t) &= (1-t)e_2 \ . \end{aligned}$$

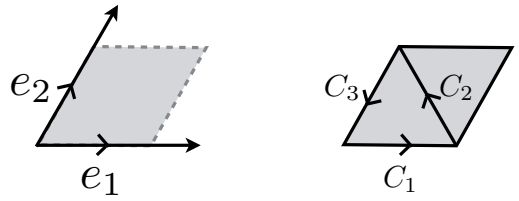


Figure 3.2: Minimal loops on the rhombic torus

for $t \in [0,1]$. Observe that these can be related by a $\pi/3$ rotation u (if we use the periodicity of the lattice), i.e.,

$$\begin{aligned} \overline{C_3}(t) &= u(C_1(t)) & C_2(t) &= u^2(C_1(t)) & \overline{C_1}(t) &= u^3(C_1(t)) \\ C_3(t) &= u^3(C_1(t)) & \overline{C_2}(t) &= u^5(C_1(t)) & C_1(t) &= u^6(C_1(t)). \end{aligned}$$

Since such a rotation u maps e_1, e_2 to

$$\begin{aligned} e_1' &= e_2 \\ e_2' &= e_2 - e_1 \ , \end{aligned}$$

it is realized by the element $u = \begin{pmatrix} 0 & 1 \\ -1 & 1 \end{pmatrix} \in SL(2,\mathbb{Z})$, which decomposes into the generators (3.37) as $u = ts^3ts$. We conclude that, expressed in the basis $\mathcal{B}_{\mathcal{C}_1}$, we have

$$\sum_{j=1}^{3} \left(F_a(C_j) + F_a(\overline{C_j}) \right) = \sum_{j=0}^{5} U^j F_a(C_1) U^{-j} \quad \text{where} \quad U = TS^3TS. \tag{3.39}$$

Again, if the lattice and Hamiltonian H_0 are invariant under a $\pi/3$ rotation, we may conclude that the effective perturbation Hamiltonian will have the form (3.39). This is the case for the Levin-Wen model on a honeycomb lattice embedded in a rhombic torus (see also Section 3.7.2).

3.6.3 Microscopic models

The purpose of this section is two-fold: First, we briefly review the construction of the microscopic models we use in our numerical experiments in Section 3.7: these include the toric code (see Section 3.6.3) as well as the doubled semion and the doubled Fibonacci model, both instantiations of the Levin-Wen construction (see Section 3.6.3). Second, we define single-qubit operators in these models and discuss their action on quasi-particle excitations (i.e., anyons). This translation of local terms in the microscopic spin Hamiltonian into operators in the effective anyon models is necessary to apply the perturbative arguments presented in Section 3.6.1. We will use these local terms to define translation-invariant perturbations (respectively trivial initial Hamiltonians) in Section 3.7).

The toric code

Kitaev's toric code [Kit03] is arguably the simplest exactly solvable model which supports anyons. It can be defined on a variety of lattices, including square and honeycomb lattices. Here we will introduce the Hamiltonian corresponding to honeycomb lattice. On each edge of the lattice resides a qubit. The Hamiltonian consists of two parts and takes the form

$$H_{\mathsf{top}} = -\sum_v A_v - \sum_p B_p \ , \tag{3.40}$$

where $B_p = X^{\otimes 6}$ is the tensor product of Pauli-X operators on the six edges of the plaquette p, and $A_v = Z^{\otimes 3}$ is the tensor product of Pauli-Z operators on the three edges connected to the vertex v.

Note that in terms of its anyonic content, the toric code is described by the double of $\mathbb{Z}_2$; hence a model with the same type of topological order could be obtained following the prescription given by Levin and Wen (see Section 3.6.3). Here we are not following this route, but instead exploit that this has the structure of a quantum double (see [Kit03]). The resulting construction, given by (3.40), results in a simpler plaquette term B_p as opposed to the Levin-Wen construction.

The anyonic excitations supported by the toric code are labeled by $\{\mathbf{1}, \boldsymbol{e}, \boldsymbol{m}, \boldsymbol{\epsilon}\}$. The $\boldsymbol{e}$ anyon or electric excitation corresponds to vertex term excitations. The $\boldsymbol{m}$ anyon or magnetic excitations correspond to plaquete term excitations. Finally, the $\boldsymbol{\epsilon}$ anyon corresponds to an excitation on both plaquete and vertex and has the exchange statistics of a fermion. We can write down the string operators $F_a(C)$ for a closed loop C on the lattice explicitly (see [Kit03]). Without loss of generality, we can set $F_{\boldsymbol{e}}(C) = P_0 \bigotimes_{i\in C} X_i P_0$ and $F_{\boldsymbol{m}}(C) = P_0 \bigotimes_{i\in D} Z_i P_0$, where D is a closed loop on the dual lattice corresponding to C. Finally, the operator $F_{\boldsymbol{\epsilon}}(C) = F_{\boldsymbol{e}}(C) \times F_{\boldsymbol{m}}(C)$ can be written as a product of $F_{\boldsymbol{e}}(C)$ and $F_{\boldsymbol{m}}(C)$, since $\boldsymbol{e}$ and $\boldsymbol{m}$ always fuse to $\boldsymbol{\epsilon}$. With respect to the ordering $(\mathbf{1}, \boldsymbol{e}, \boldsymbol{m}, \boldsymbol{\epsilon})$ of the anyons, the S- and T-matrices described in Section 3.5.1 are given by

$$T = \mathsf{diag}(1, 1, 1, -1) \quad S = 1/2 \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \tag{3.41}$$

for the toric code.

Local spin operators. A natural basis of (Hermitian) operators on a single qubit is given by the Pauli operators. For the toric code, each of these operators has a natural interpretation in terms of the underlying anyon model.

Consider for example a single-qubit Z-operator. The "anyonic lattice" associated with $\boldsymbol{m}$-anyons is the dual lattice (i.e., these anyons 'live' on plaquettes), and a single-qubit Z-operator acts by either creating or annihilating a $(\boldsymbol{m}, \bar{\boldsymbol{m}}) = (\boldsymbol{m}, \boldsymbol{m})$ on the neighboring plaquettes, or propagating an existing $\boldsymbol{m}$ from one plaquette to the other. That is, in the terminology of Section 3.5.1, a Z-operator acts as a local term

$$Z \qquad \longleftrightarrow \quad \hat{V}^{\mathsf{C}}(\boldsymbol{m}) + \hat{V}^{\mathsf{A}}(\boldsymbol{m}) + \hat{V}^{\mathsf{L}}(\boldsymbol{m}) + \hat{V}^{\mathsf{R}}(\boldsymbol{m}) \tag{3.42}$$

in the effective anyon model. An analogous identity holds for X, which is associated with $\boldsymbol{e}$-anyons: the latter live on vertices of the spin lattice. Finally, Y-operators act on $\boldsymbol{\epsilon}$-anyons in the same manner; these anyons live on 'supersites', consisting of a plaquette and and an adjacent vertex.

Short introduction to the Levin-Wen model

Levin and Wen [LW05] define a family of frustration-free commuting Hamiltonian with topologically ordered ground space and localized anyonic excitations. Their construction is based on interpreting the state of spins residing on the edges of a trivalent lattice (such as a honeycomb lattice) as configurations of string-nets.

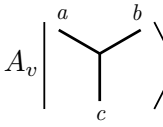
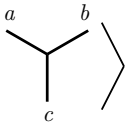
To specify a string-net model, we need algebraic data associated with an anyon model as described in Section 3.5.1. This specifies, in particular, a set of anyon labels $\mathcal{F} = \{a_i\}$, associated fusion rules, as well as S- and F-matrices. The Levin-Wen model then associates a qudit to each edge of the lattice, where the local dimension of each spin corresponds to the number of anyon labels in $\mathcal{F}$. One chooses an orthonormal basis $\{|a\rangle\}_{a\in\mathcal{F}} \subset \mathbb{C}^{|\mathcal{F}|}$ indexed by anyon labels; in the following, we usually simply write a instead of $|a\rangle$ to specify a state of a spin in the microscopic

model. The Levin-Wen spin Hamiltonian can be divided into two parts,

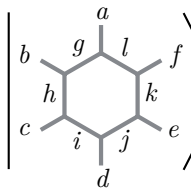
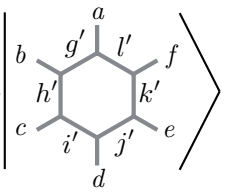
$$H_{\mathsf{top}} = -\sum_v A_v - \sum_p B_p \ , \tag{3.43}$$

where each B_p is a projector acting on the 12 edges around a plaquette p, and each A_v is a projector acting on the 3 edges around a vertex v. In particular, we can construct the spin Hamiltonian for the doubled semion and the doubled Fibonacci models in this way by choosing different initial data.

As long as all the particles in the underlying model $\mathcal{F}$ are their own antiparticles (i.e., the involution $a \mapsto \bar{a}$ is the identity), it is not necessary to assign an orientation to each edge of the lattice. This affords us an important simplification, which is justified for the models under consideration: these only have a single non-trivial anyon label, which is itself its own antiparticle (recall that the trivial label satisfies $\bar{1} = 1$). With this simplification, which we will use throughout the remainder of this paper, the vertex operator A_v can be written as

A_v  $= \delta_{abc}$ 

where $\delta_{abc} = 1$ if a and b can fuse to c and $\delta_{abc} = 0$ otherwise. The plaquette operator B_p is more complicated compared to A_v. We will give its form without further explanation

B_p  $= \sum_{\substack{s,g',h' \\ i',j',k',l'}} \frac{d_s}{D^2} F^{agl}_{sl'g'} F^{bhg}_{sg'h'} F^{cih}_{sh'i'} F^{dji}_{si'j'} F^{ekj}_{sj'k'} F^{flk}_{sk'l'}$ 

,

where d_s is the quantum dimension of the anyon label s, and $D = \sqrt{\sum_j d_j^2}$ is the total quantum dimension.

Having specified the spin Hamiltonian, we stress that the anyon labels $\mathcal{F}$ used in this construction should not be confused with the anyon labels $\mathrm{D}(\mathcal{F})$ describing the local excitations in the resulting Hamiltonian (3.43). The latter can be described as 'pairs' of anyons from $\mathcal{F}$, i.e., $\mathrm{D}(\mathcal{F}) = \{(a_i, a_j)\}_{a_i, a_j \in \mathcal{F}}$. Their fusion, twist and braiding properties are described by the double of the original theory. The $S_{\mathrm{D}(\mathcal{F})}$- and $F_{\mathrm{D}(\mathcal{F})}$- matrices of $\mathrm{D}(\mathcal{F})$ can be obtained from the S- and T-matrix associated with $\mathcal{F}$ (see [LW05]). String operators $F_{a_i,a_j}(C)$ acting on the spin lattice have also been explicitly constructed in [LW05]

Below, we present some of the specifics of two models constructed in this way: the doubled semion and doubled Fibonacci model. In addition to Kitaev's toric codes $\mathrm{D}(\mathbb{Z}_2)$, these are the only models defined on two labels (i.e., with microscopic qubit degrees of freedom).

The doubled semion model

The underlying string-net model of the doubled semion model only consists of one non-trivial label $\mathbf{s}$ and the trivial label $\mathbf{1}$. To specify the spin Hamiltonian, we have $d_{\boldsymbol{s}} = 1$, and $\delta_{abc} = 1$ if and only if an even number of a, b, c are $\boldsymbol{s}$. The F-matrix is given by $F^{\boldsymbol{ss1}}_{\boldsymbol{ss1}} = -1$ and otherwise F^{abc}_{def} is 0 or 1 depending on whether (a, b, c, d, e, f) is a legal configuration (see [Kit06] for more detailed explanation). As we explained above, to construct a spin Hamiltonian, we put a qubit on each edge of the lattice with orthonormal basis $|\mathbf{1}\rangle, |\boldsymbol{s}\rangle$. The spin Hamiltonian obtained this way is similar to the toric code and it also supports Abelian anyons. The excitations of the spin model can be labeled by $\mathrm{D}(\mathcal{F}) = \{(\mathbf{1}, \mathbf{1}), (\mathbf{1}, \boldsymbol{s}), (\boldsymbol{s}, \mathbf{1}), (\boldsymbol{s}, \boldsymbol{s})\}$, which is the quantum double of $\mathcal{F} = \{\mathbf{1}, \boldsymbol{s}\}$. With

respect to the given ordering of anyons, the S- and T-matrices of these excitations are given by

$$S = 1/2 \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix} \qquad T = \mathsf{diag}(1, i, -i, 1) \tag{3.44}$$

Local operators. Identifying $|\mathbf{1}\rangle$ with the standard basis state $|0\rangle$ and $|\boldsymbol{s}\rangle$ with $|1\rangle$, we can again use Pauli operators to parametrize single-spin Hamiltonian terms.

Here we will discuss the effect of single qubit operators X and Z on the ground states of the resulting topologically ordered Hamiltonian. The goal is to interpret single spin operators in terms of effective anyon creation, annihilation and hopping operators.

When Z-operator is applied to an edge of the system in a ground state, only the neighboring plaquete projectors B_p will become excited. More specifically, a pair of $(\boldsymbol{s}, \boldsymbol{s})$ anyons are created if none were present. Since $(\boldsymbol{s}, \boldsymbol{s})$ is an abelian anyon, in fact a boson, and is the anti-particle of itself, a Z operator could also move an $(\boldsymbol{s}, \boldsymbol{s})$ anyon or annihilate two such particles if they are already present. Thus we conclude that single-qubit Z-operators have a similar action as in the toric code (cf. (3.42)), with $\boldsymbol{s}$ playing the role of the anyon $\boldsymbol{m}$.

When an X operator is applied on edge of the system in a ground state, it excites the two neighboring vertex terms A_v (in the sense that the state is no longer a $+1$-eigenstate any longer). Since the plaquete terms B_p are only defined within the subspace stabilized by A_v, the four plaquette terms B_p terms around the edge also become excited. It is unclear how to provide a full interpretation of X operators in terms of an effective anyon language. In order to provide this, a full interpretation of the spin Hilbert space and its operators in the effective anyonic language is required; such a description is currently not known.

In summary, this situation is quite different from the case of the toric code, where X and Z are dual to each other.

The doubled Fibonacci

Again, the underlying string-net model of doubled Fibonacci contains only one non-trivial label $\boldsymbol{\tau}$, with quantum dimension $d_{\boldsymbol{\tau}} = \varphi$, where $\varphi = \frac{1+\sqrt{5}}{2}$. The fusion rules are given by $\delta_{abc} = 0$ if only one of the a, b, c is $\boldsymbol{\tau}$, and otherwise $\delta_{abc} = 1$. Non-trivial values of F are

$$F^{\boldsymbol{\tau}\boldsymbol{\tau}\mathbf{1}}_{\boldsymbol{\tau}\boldsymbol{\tau}\mathbf{1}} = \varphi^{-1},\ F^{\boldsymbol{\tau}\boldsymbol{\tau}\mathbf{1}}_{\boldsymbol{\tau}\boldsymbol{\tau}\boldsymbol{\tau}} = \varphi^{-1/2}$$
$$F^{\boldsymbol{\tau}\boldsymbol{\tau}\boldsymbol{\tau}}_{\boldsymbol{\tau}\boldsymbol{\tau}\mathbf{1}} = \varphi^{-1/2},\ F^{\boldsymbol{\tau}\boldsymbol{\tau}\boldsymbol{\tau}}_{\boldsymbol{\tau}\boldsymbol{\tau}\boldsymbol{\tau}} = -\varphi^{-1},$$

and otherwise F^{abc}_{def} is either 0 or 1 depending on whether (a, b, c, d, e, f) is a legal configuration.

Many aspects of the doubled Fibonacci spin Hamiltonian are similar to the doubled semion model:

- There is one qubit on each edge, with orthonormal basis states associated with the anyon labels $\mathcal{F} = \{\mathbf{1}, \boldsymbol{\tau}\}$.
- The anyons supported by the spin Hamiltonian carry labels $\mathrm{D}(\mathcal{F}) = \{(\mathbf{1}, \mathbf{1}), (\mathbf{1}, \boldsymbol{\tau}), (\boldsymbol{\tau}, \mathbf{1}), (\boldsymbol{\tau}, \boldsymbol{\tau})\}$.

With respect to the given ordering of anyons, the S- and T-matrices are given by

$$S = \begin{pmatrix} 1 & \varphi & \varphi & \varphi^2 \\ \varphi & -1 & \varphi^2 & -\varphi \\ \varphi & \varphi^2 & -1 & -\varphi \\ \varphi^2 & -\varphi & -\varphi & 1 \end{pmatrix} /(1+\varphi^2) \qquad T = \mathsf{diag}(1, e^{-4\pi/5}, e^{4\pi/5}, 1)\ . \tag{3.45}$$

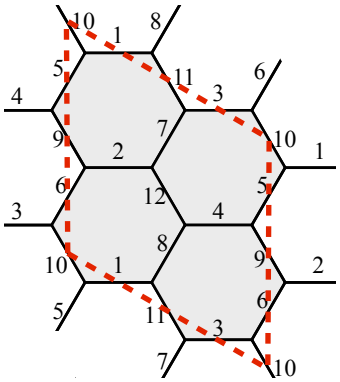


Figure 3.1: The 12-qubit-torus we use for numerical simulation (qubits are numbered 1 to 12. It is a rhombic torus and we can identify three minimal loops $\{1,2\},\{5,7\},\{9,11\}$ (and their inverses) which are related by $\pi/3$ rotations.

A substantial difference to the doubled semion model is that the non-trivial anyons supported by the model are non-abelian. One manifestation of this fact we encounter concerns the $(\boldsymbol{\tau},\boldsymbol{\tau})$-anyon:

- While $(\boldsymbol{\tau},\boldsymbol{\tau})$ is its own anti-particle, it is not an abelian particle so in general two $(\boldsymbol{\tau},\boldsymbol{\tau})$ particles will not necessarily annihilate with each other. In other words, the dimension of the subspace carrying two localized $(\boldsymbol{\tau},\boldsymbol{\tau})$ charges is larger than the dimension of the charge-free subspace.

- Two intersecting string operators $F_{(\boldsymbol{\tau},\boldsymbol{\tau})}(C_1)$ and $F_{(\boldsymbol{\tau},\boldsymbol{\tau})}(C_2)$ corresponding to the $(\boldsymbol{\tau},\boldsymbol{\tau})$ particle do not commute with each other.

Neither of these properties holds for the $(\boldsymbol{s},\boldsymbol{s})$-anyon in the case of the doubled semion model.

Local operators. Similarly, as before, we identify $|\mathbf{1}\rangle$ with the standard basis state $|0\rangle$ and $|\boldsymbol{\tau}\rangle$ with $|1\rangle$, enabling us to express single-qubit operators in terms of the standard Pauli operators.

Again, we want to consider the effect of single qubit operators in terms of anyons. This is generally rather tricky, but for single-qubit Z-operators, we can obtain partial information from an analysis presented in appendix 3.B: Let $|\psi\rangle$ be a ground state. Then $Z|\psi\rangle = \frac{1}{\sqrt{5}}|\psi\rangle + \frac{4}{5}|\varphi\rangle$, where $|\varphi\rangle$ is a ψ-dependent excited state with a pair of $(\boldsymbol{\tau},\boldsymbol{\tau})$ on the plaquettes next to the edge Z acts on. Thus the resulting state after application of a single Z operator has support both on the excited and as well as the ground subspace. Again, this is in contrast to the doubled semion model, where a single-qubit Z operator applied to the ground space always results in an excited eigenstate of the Hamiltonian.

3.7 Numerics

In this section, we present results obtained by numerically simulating Hamiltonian interpolation for small systems. Specifically, we consider three topologically ordered systems on the 12-qubit honeycomb lattice of Fig 3.1: the toric code, the doubled semion and the doubled Fibonacci Levin-Wen models. That is, the target Hamiltonian H_{top} is given either by (3.40) (with stabilizer plaquette- and vertex-operators A_v and B_p) in the toric code case, and expression (3.43) specified in Section 3.6.3 (with projection operators A_v and B_p) for the doubled semion and the doubled Fibonacci case. As initial Hamiltonian H_{triv}, we choose certain translation-invariant Hamiltonians consisting of single-qubit Pauli-X, Pauli-Y and Pauli-Z operators (see Sections 3.6.3 and 3.6.3 for their definition and a discussion of the effect of these operators in the two Levin-Wen models.) For concreteness and ease of visualization, we will consider the following families of such Hamiltonians:

the one-parameter family

$$H_{\mathsf{triv}}(\theta) = \cos\theta \sum_j Z_j + \sin\theta \sum_j X_j \tag{3.46}$$

where $\theta \in [0, 2\pi[$, and two two-parameter families of the form

$$H^{\pm}_{\mathsf{triv}}(a,b) = a\sum_j X_j + b\sum_j Y_j \pm (1-a^2-b^2)^{1/2}\sum_j Z_j \ , \tag{3.47}$$

where $(a,b) \in \mathbb{R}^2$ belongs to the unit disc, $a^2+b^2 \leq 1$. (In some instances, we will permute the roles of X, Y and Z, and use an additional superscript to indicate this.)

For different parameter choices θ respectively (a,b), we study Hamiltonian interpolation (i.e., the evolution (3.2)) along the linear interpolation path $H(t)$ (cf. (3.1)) with a total evolution time T. In order to numerically simulate the evolution under the time-dependent Schrödinger equation, we perform a time-dependent Trotter expansion using the approximation

$$\mathcal{T}\exp\left(i\int_0^t H(s)ds\right) \approx \prod_{j=1}^{\lfloor T/\Delta t\rfloor} e^{iH(j\cdot\Delta T)\Delta t} \quad \text{and} \quad e^{iH(t)\Delta t} \approx e^{i\frac{(T-t)}{T}H_{\mathsf{triv}}\Delta t}e^{i\frac{t}{T}H_{\mathsf{top}}\Delta t} \ . \tag{3.48}$$

Unless otherwise specified, the time discretization is taken to be $\Delta t = 0.1$.

3.7.1 Quantities of interest and summary of observations

Recall that our initial state $\Psi(0) = \varphi^{\otimes 12}$ is the unique 12-qubit ground state of the chosen trivial Hamiltonian H_{triv}. We are interested in the states $\Psi(t)$ along the evolution, and, in particular, the final state $\Psi(T)$ for a total evolution time T. For notational convenience, we will write $\Psi_\theta(t)$, respectively $\Psi^{\pm}_{a,b}(t)$ to indicate which of the initial Hamiltonians H_{triv} is considered (cf. (3.46) and (3.47)). We consider the following two aspects:

(Non)-adiabaticity: We investigate whether the state $\Psi(t)$ follows the instantanenous ground space along the evolution (3.2). We quantify this using the *adiabaticity error*, which we define (for a fixed total evolution time T, which we suppress in the notation) as

$$\epsilon_{\text{adia}}(t) := 1 - |\langle\Psi(t)|P_0(t)|\Psi(t)\rangle|^2 \qquad \text{for } 0 \leq t \leq T \ , \tag{3.49}$$

where $P_0(t)$ is the projection onto the ground space of $H(t)$ (note that except for $t = T$, where $P_0(T)$ projects onto the degenerate ground space of H_{top}, this is generally a rank-one projection). The function $t \mapsto \epsilon_{\text{adia}}(t)$ quantifies the overlap with the instantaneous ground state of $H(t)$ along the Hamiltonian interpolation $t \mapsto H(t)$, and hence directly reflects adiabaticity.

Ultimately, we are interested in whether the evolution reaches a ground state of H_{top}. This is measured by the expression $\epsilon_{\text{adia}}(T)$, which quantifies the deviation of the final state $\Psi(T)$ from the ground space of H_{top}. Clearly, the quantity $\epsilon_{\text{adia}}(T)$ depends on the choice of initial Hamiltonian H_{triv} (i.e., the parameters θ respectively (a,b)) and the total evolution time T. For sufficiently large choices of the latter, we expect the adiabaticity assumption underlying Conjecture 1 to be satisfied, and this is directly quantifiable by means of the adiabaticity error. We will also discuss situations where, as discussed in Observation 3.2.3, symmetries prevent reaching the ground space of H_{top} as reflected in a value of $\epsilon_{\text{adia}}(T)$ close to 1.

Logical state: assuming the ground space of H_{top} is reached (as quantified by $\epsilon_{\text{adia}}(T)$), we will identify the logical state $\Psi(T)$ and investigate its stability under perturbations of the the initial Hamiltonian H_{triv} (i.e., changes of the parameters θ respectively (a,b)). For this purpose, we employ the following measures:

- We argue (see Section 3.7.2) that symmetries constrain the projection of the resulting state $\Psi(T)$ onto the ground space of H_{top} to a two-dimensional subspace (see Section 3.7.2). For the toric code, the state is then fully determined by the expectation values $\langle \bar{X} \rangle_{\Psi(T)}$, $\langle \bar{Z} \rangle_{\Psi(T)}$ of two logical operators $\bar{X}$ and $\bar{Z}$. To investigate stability properties of the prepared state, we can therefore consider $(\langle \bar{X} \rangle_{\Psi(T)}, \langle \bar{Z} \rangle_{\Psi(T)})$ as a function of parameters of the initial Hamiltonian.
- for the Levin-Wen models, we proceed as follows: we pick a suitable reference state $|\psi_R\rangle \in (\mathbb{C}^2)^{\otimes 12}$ in the ground space of H_{top}, and then study how the overlap $|\langle \Psi^{\pm}_{a,b}(T) | \psi_R \rangle|^2$ changes as the parameters (a, b) of the initial Hamiltonian are varied. In particular, if we fix a pair (a_0, b_0) and choose $|\psi_R\rangle$ as the normalized projection of the state $|\Psi^{\pm}_{a_0,b_0}(T)\rangle$ onto the ground space of H_{top}, this allows us to study the stability of the prepared state $|\Psi^{\pm}_{a,b}(T)\rangle$ as a function of the Hamiltonian parameters (a, b) in the neighborhood of (a_0, b_0).

  According to the reasoning in Section 3.3.2 (see Conjecture 1), the specific target state $|\psi^{\mathrm{ref}}_{a_0,b_0}\rangle$ chosen in this way should correspond to the ground state of $H_{\mathsf{top}} + \epsilon H^{\pm}(a_0, b_0)$ in the limit $\epsilon \to 0$ of infinitesimally small perturbations (or, more precisely, the corresponding effective Hamiltonian). Furthermore, according to the reasoning in Section 3.6.1, the family of effective Hamiltonians associated with $H_{\mathsf{top}} + \epsilon H^{\pm}(a, b)$ has a very specific form. This should give rise to a certain stability of the ground space as a function of the parameters (a, b).

  To support this reasoning, we numerically compute the (exact) ground state $|\psi^{\mathrm{pert}}_{a,b}\rangle$ of $H_{\mathsf{top}} + \epsilon H^{\pm}(a, b)$ for the choice $\epsilon = 0.001$ (as a proxy for the effective Hamiltonian), and study the overlap $|\langle \psi^{\mathrm{pert}}_{a,b} | \psi^{\mathrm{ref}}_{a_0,b_0} \rangle|^2$ as a function of the parameters (a, b) in the neighborhood of (a_0, b_0).

The results of our numerical experiments support the following two observations:

- Hamiltonian interpolation is generically able to prepare approximate ground states of these topological models for sufficiently long total evolution times T.
- Specific final state(s) show a certain degree of stability with respect to changes in the initial Hamiltonian. The theoretical reasoning based on perturbation theory presented in Section 3.6 provides a partial explanation of this phenomenon.

3.7.2 A symmetry of the 12-qubit rhombic torus

As discussed in Section 3.6.3, the ground space of H_{top} on a torus is 4-dimensional for the toric code, the doubled semion- and the Fibonacci model. In this section, we argue that adiabatic interpolation starting from a translation-invariant Hamiltonian (as considered here) yields states belonging to a two-dimensional subspace of this ground space, thus providing a simplification.

Consider again the 12-qubit rhombic torus illustrated in Fig. 3.1. A $\pi/3$ rotation permuting the physical qubits according to

$$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12) \mapsto (5, 7, 8, 6, 9, 12, 11, 10, 2, 4, 1, 3)$$

defines a unitary $U_{\pi/3}$ on $\mathbb{C}^{\otimes 12}$. Because of translation-invariance, this is a symmetry of the trivial Hamiltonian, $U_{\pi/3} H_{\mathsf{triv}} U^{\dagger}_{\pi/3} = H_{\mathsf{triv}}$, and it can easily be verified that for the models considered here, the unitary $U_{\pi/3}$ also commutes with H_{top}. Because of the product form of the initial state $\Psi(0)$, it thus follows that $U_{\pi/3}\Psi(t) = \Psi(t)$ along the whole trajectory $t \mapsto \Psi(t)$ of adiabatic interpolation. In particular, the projection of the final state $\Psi(T)$ onto the ground space of H_{top} is supported on the $+1$-eigenspaces space of $U_{\pi/3}$.

As discussed in Section 3.6.2, a $\pi/3$-rotation of the rhombic torus corresponds to the modular transformation ts^3ts. Since $U_{\pi/3}$ realizes this transformations, its restriction to the ground space of H_{top} can be computed from the T and S-matrices. That is, expressed in the flux bases discussed in Section 3.6.3, the action of $U_{\pi/3}$ on the ground space is given by the matrix TS^3TS, where

(S, T) are given by (3.41) for the toric code, as well as (3.44) and (3.45) for the doubled semion and Fibonacci models, respectively. The specific form of TS^3TS or its eigenvectors is not particularly elucidating, but may be computed explicitly.

Importantly, the $+1$ eigenspace of TS^3TS is two-dimensional for the toric code, the doubled semion and the Fibonacci models. (In the case of the toric code, it can be verified that this eigenspace is contained in the logical symmetric subspace. The latter is the subspace invariant under swapping the two logical qubits in the standard computational basis.) As a result, the projection of the state $\Psi(T)$ onto the ground space of H_{top} belongs to a known two-dimensional subspace which can be explicitly computed. This means that we may characterize the resulting state in terms of a restricted reduced set of logical observables, a fact we will exploit in Section 3.7.3.

3.7.3 The toric code

As discussed in Section 3.6.3, for the toric code on the honeycomb lattice (see Fig. 3.1), the Hamiltonian of the model is $H_{\mathsf{top}} = -(\sum_p B_p + \sum_v A_v)$, where $B_p = X^{\otimes 6}$ is a tensor product of Pauli-X operators on the six edges of the plaquette p, and $A_v = Z^{\otimes 3}$ is a tensor product of Pauli-Z operators on the three edges incident on the vertex v. We point out that the toric code on a honeycomb lattice has several differences compared to a toric code on a square lattice (which is often considered in the literature). Assuming that both lattices are defined with periodic boundary conditions,

(i) there are twice as many vertices compared to plaquettes on a honeycomb lattice (as opposed to the same number on a square lattice)

(ii) the vertex terms $A_v = Z^{\otimes 3}$ of the Hamiltonian have odd weights (as opposed to even weight for the square lattice)

(iii) the weight of a logical minimal $\bar{X}$-string operator (i.e. the number of spins it acts on) is roughly twice as large compared to the corresponding minimal $\bar{Z}$-string operator on the dual lattice (as opposed to the square lattice, where both operators have the same weight). For the 12-qubit code of Fig. 3.1, an example of such a pair $(\bar{X}, \bar{Z})$ of lowest-weight logical operators is given below in Eq. (3.51).

Properties (i) and (ii) imply that the usual symmetries $X \leftrightarrow Z$ and $Z \leftrightarrow -Z$ of the toric code on the square lattice are not present in this case. The absence of these symmetries is reflected in our simulations. Property (iii) also directly affects the final state, as can be seen by the perturbative reasoning of Section 3.6.1: $\bar{Z}$-string operators appear in lower order in perturbation theory compared to $\bar{X}$-string operators.

(Non)-adiabaticity. We first present the adiabaticity error $\epsilon_{\text{adia}}(T)$ for the Hamiltonian $H_{\mathsf{triv}}(\theta)$ given by (3.46) (for different values of θ) as a function of the total evolution time T. Fig. 3.2 illlustrates the result. It shows that for sufficiently long total evolution times T, the Hamiltonian interpolation reaches the ground space of the toric code when the initial Hamiltonian is $H_{\mathsf{triv}}(\theta = \pi) = -\sum_i Z_i$; this is also the case for $\theta \in \{\pi/4, \pi/2, 3\pi/4\}$.

However, if the initial Hamiltonian is $H_{\mathsf{triv}}(\theta = 0) = \sum_i Z_i$, then the final state $\Psi(T)$ is far from the ground space of the toric code Hamiltonian H_{top}. This phenomenon has a simple explanation along the lines of Observation 3.2.3. Indeed, if $\theta = 0$, then every vertex terms $A_v = Z^{\otimes 3}$ commutes with both H_{triv} as well as H_{top} (and thus all intermediate Hamiltonians $H(t)$). In particular, the expectation value of the vertex terms remains constant throughout the whole evolution, and this leads to an adiabaticity error $\epsilon_{\text{adia}}(T)$ of 1 in the case of $H_{\mathsf{triv}}(\theta = 0) = \sum_i Z_i$.

In Figs. 3.3a, 3.3b, we consider neighborhoods of Hamiltonians of the form (cf. (3.47))

$$H^+_{\mathsf{triv}}(a, b) \qquad \text{around} \qquad H^+_{\mathsf{triv}}(0, 0) = H_{\mathsf{triv}}(\theta = 0) = \sum_j Z_j \qquad \text{and}$$

$$H^-_{\mathsf{triv}}(a, b) \qquad \text{around} \qquad H^-_{\mathsf{triv}}(0, 0) = H_{\mathsf{triv}}(\theta = \pi) = -\sum_j Z_j \ .$$

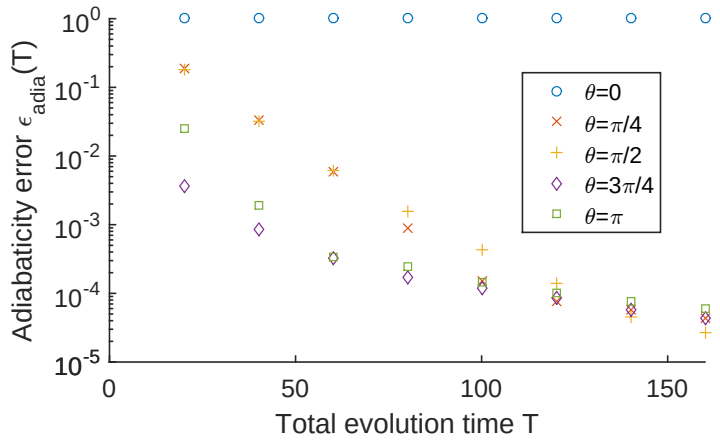


Figure 3.2: This figure gives the adiabaticity error $\epsilon_{\text{adia}}(T) = 1 - \langle\Psi(T)|P_0(T)|\Psi(T)\rangle$ (cf. (3.49)) as a function of the total evolution time T and the initial Hamiltonian chosen. For the latter, we consider the one-parameter family $H_{\text{triv}}(\theta)$ given by (3.46). For $\theta = 0$, the adiabatic evolution is not able to reach the final ground space because initially $\langle A_v\rangle = -1$ for every vertex operator $A_v = Z^{\otimes 3}$, and this quantity is conserved during the evolution. This is a feature of the honeycomb lattice because the vertex terms A_v have odd weights. For other values of θ, the ground space is reached for sufficiently large total evolution times T.

The initial Hamiltonians $H_{\text{triv}}(\theta = 0)$ and $H_{\text{triv}}(\theta = \pi)$ correspond to the center points in Fig. 3.3a and 3.3b, respectively.

- In the first case (Fig. 3.3a), we observe that for all initial Hamiltonians of the form $H^{+}_{\text{triv}}(a, b)$ in a small neighborhood of $H^{+}_{\text{triv}}(0, 0)$, the adiabaticity error $\epsilon_{\text{adia}}(T)$ is also large, but drops off quickly outside that neighborhood. This is consistent with the relevant level crossing(s) being avoided by introducing generic perturbations to the initial Hamiltonian.

- In contrast, almost all initial Hamiltonians in the family $H^{-}_{\text{triv}}(a, b)$ (around the initial Hamiltonian $H^{-}_{\text{triv}}(0, 0)$) lead to a small adiabaticity error $\epsilon_{\text{adia}}(T)$ (Fig. 3.3b), demonstrating the stability of the adiabatic preparation.

In a similar vein, Fig. 3.3c illustrates the non-adiabaticity for the family of Hamiltonian

$$H^{-X}_{\text{triv}}(a, b) = -(1 - a^2 - b^2)^{1/2} \sum_j X_j + b \sum_j Y_j + a \sum_j Z_j \ . \tag{3.50}$$

The family $H^{+X}_{\text{triv}}(a, b)$ (defined with a positive square root) would behave exactly the same due to the symmetry $+X \leftrightarrow -X$.

Logical state. For the 12-qubit rhombic toric code (Fig. 3.1), logical observables associated with the two encoded logical qubits can be chosen as

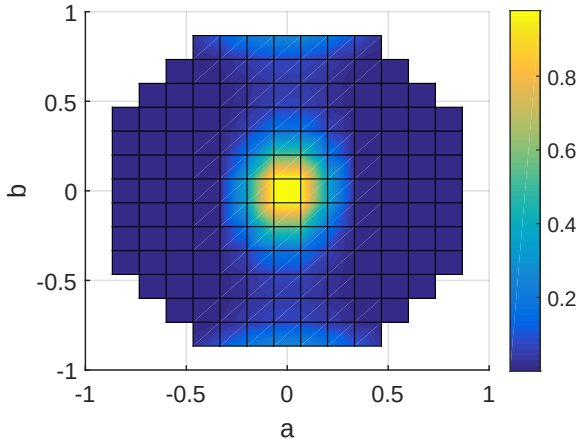
$$\begin{array}{rcc} \bar{X}_1 &=& X_7X_8X_{11}X_{12} \\ \bar{Z}_1 &=& Z_{10}Z_{12} \end{array} \quad \text{and} \quad \begin{array}{rcc} \bar{X}_2 &=& X_4X_0X_2X_{12} \\ \bar{Z}_2 &=& Z_1Z_2 \end{array} \ .$$

Because of the symmetry (3.7.2), however, these are not independent for a state $\Psi(T)$ (or more precisely, its projection $P_0(T)\Psi(T)$) prepared by Hamiltonian interpolation from a product state: their expectation values satisfy the identities

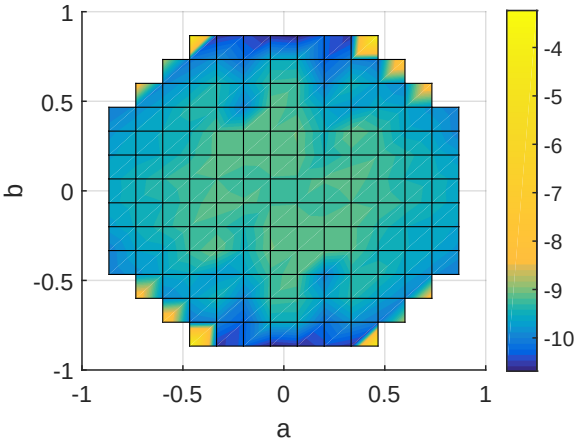
$$\langle\bar{Z}_1\rangle = \langle\bar{Z}_2\rangle \qquad \text{and} \qquad \langle\bar{X}_1\rangle = \langle\bar{X}_2\rangle \ .$$

We will hence use the two (commuting) logical operators

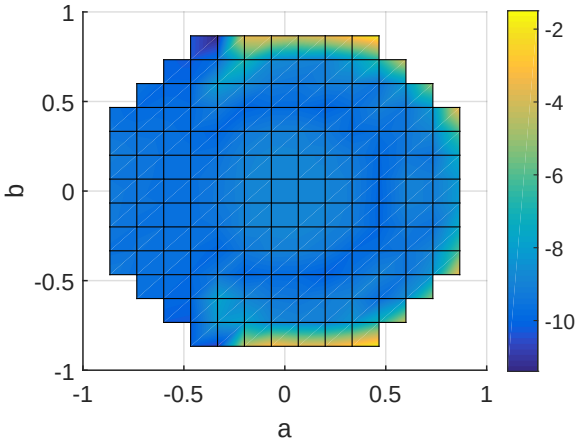
$$\bar{X} = \bar{X}_1 = X_7X_8X_{11}X_{12} \qquad \text{and} \qquad \bar{Z} = \bar{Z}_2 = Z_1Z_2 \tag{3.51}$$



(a) The adiabaticity error $\epsilon_{\text{adia}}(T)$ in the neighborhood around $H^{+}_{\text{triv}}(0,0) = \sum_i Z_i$ for different Hamiltonians $H^{+}_{\text{triv}}(a,b)$. As explained, the evolution cannot reach the ground space of the toric code around $(a,b) = (0,0)$ because the expectation values of plaquette-operators are preserved.



(b) The logarithm $\ln \epsilon_{\text{adia}}(T)$ of the adiabaticity error in the neighborhood around $H^{-}_{\text{triv}}(0,0) = -\sum_i Z_i$ for different Hamiltonians $H^{-}_{\text{triv}}(a,b)$. Here we use a log-scale because the variation in values is small. The ground space of the toric code Hamiltonian H_{top} is reached for almost the entire parameter region.



(c) The logarithm of adiabaticity error $\ln \epsilon_{\text{adia}}(T)$ in the neighborhood around $H^{-X}_{\text{triv}}(0,0) = -\sum_j X_j$ for different Hamiltonians $H^{-X}_{\text{triv}}(a,b)$. Note that the resulting figure would look identical for the Hamiltonians $H^{+X}_{\text{triv}}(a,b)$ because of the $-X \leftrightarrow +X$ symmetry.

Figure 3.3: The adiabaticity error $\epsilon_{\text{adia}}(T) = 1 - \langle \Psi(T)|P_0(T)|\Psi(T)\rangle$, measuring how well the final state $\Psi(T)$ overlaps with the ground space of the toric code. All three figures are for a total evolution time $T = 120$. In Fig. 3.3a, we consider the family of initial Hamiltonians $H^{+}_{\text{triv}}(a,b)$ in the neighborhood of $H^{+}_{\text{triv}}(0,0) = H_{\text{triv}}(\theta = 0) = \sum_j Z_j$. In contrast, Fig. 3.3b illustrates different choices of initial Hamiltonians $H^{-}_{\text{triv}}(a,b)$ around $H^{-}_{\text{triv}}(0,0) = H_{\text{triv}}(\theta = \pi) = -\sum_j Z_j$. The values $(a,b) \subset \mathbb{R}^2$ are restricted to the unit disc $a^2+b^2 \leq 1$; the center points of the two figures correspond respectively to $\theta = 0$ and $\theta = \pi$ in Fig. 3.2. Finally, Fig. 3.3c gives the non-adiabaticity error for initial Hamiltonians of the form $H^{-X}_{\text{triv}}(a,b)$ (as defined in Eq. (3.50)).

to describe the obtained logical state.

In Fig. 3.4, we plot the expectation values of $\bar{Z}$ and $\bar{X}$ in the final state $\Psi(T)$ for initial Hamiltonians of the form (cf. (3.47) and (3.50))

$$H^{-}_{\mathsf{triv}}(a,b) \qquad \text{around} \qquad H^{-}_{\mathsf{triv}}(0,0) = -\sum_j Z_j$$

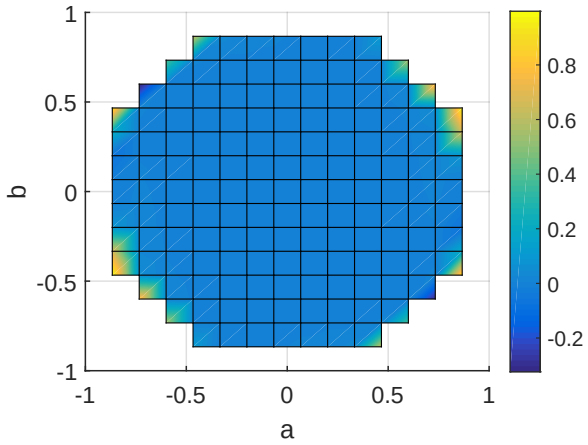
$$H^{-X}_{\mathsf{triv}}(a,b) \qquad \text{around} \qquad H^{-X}_{\mathsf{triv}}(0,0) = -\sum_j X_j$$

We again discuss the center points in more detail. It is worth noting that the single-qubit $\{Z_i\}$ operators correspond to the local creation, hopping and annihilation of $\boldsymbol{m}$ anyons situated on plaquettes, whereas the operators $\{X_i\}$ are associated with creation, hopping and annihilation of $\boldsymbol{e}$ anyons situated on vertices. In particular, this means that the initial Hamiltonians associated with the center points in the two figures each generate processes involving only either type of anyon.

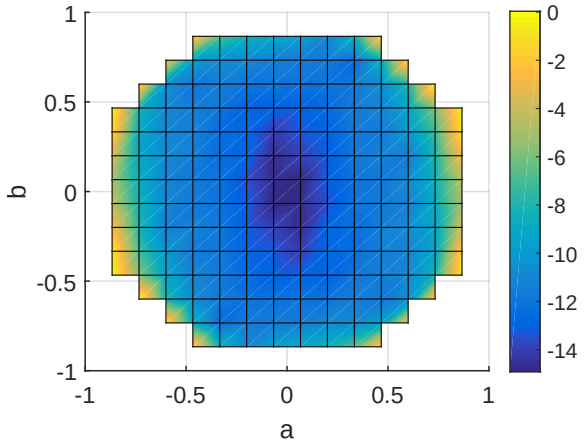
- For $H^{-}_{\mathsf{triv}}(0,0) = -\sum_i Z_i$, we know that $\langle \bar{Z} \rangle = 1$ during the entire evolution because $\bar{Z}$ commutes with the Hamiltonians $H(t)$, and the initial ground state $\Psi(0)$ is a $+1$ eigenstate of $\bar{Z}$. In Figs. 3.4a and 3.4b, we can see that there is a large region of initial Hamiltonians $H^{-}_{\mathsf{triv}}(a,b)$ around $H^{-}_{\mathsf{triv}}(0,0) = -\sum_i Z_i$ which lead to approximately the same final state.

- On the other hand, as shown in Figs. 3.4c and 3.4d, the stable region of Hamiltonians $H^{-X}_{\mathsf{triv}}(a,b)$ around the initial Hamiltonian $H^{-X}_{\mathsf{triv}}(0,0) = -\sum_i X_i$ is much smaller. This is due to the fact that the operator $\bar{X}$ appears in higher order perturbation expansion compared to $\bar{Z}$, and the evolution time T is taken to be quite long. Given sufficiently large total evolution time T, in the neighborhood of $H^{-X}_{\mathsf{triv}}(0,0) = -\sum_i X_i$, the lower order term $\bar{Z}$ in the effective Hamiltonian will dominate the term $\bar{X}$ associated with $V = -\sum_i X_i$.

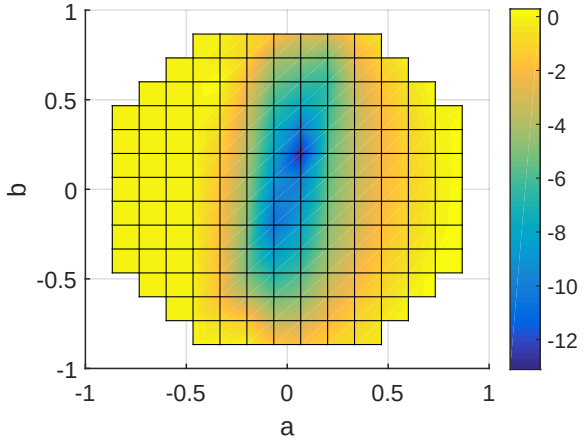
However, in both cases considered in Fig. 3.4, we observe that one of two specific logical states is prepared with great precision within a significant fraction of the initial Hamiltonian parameter space.



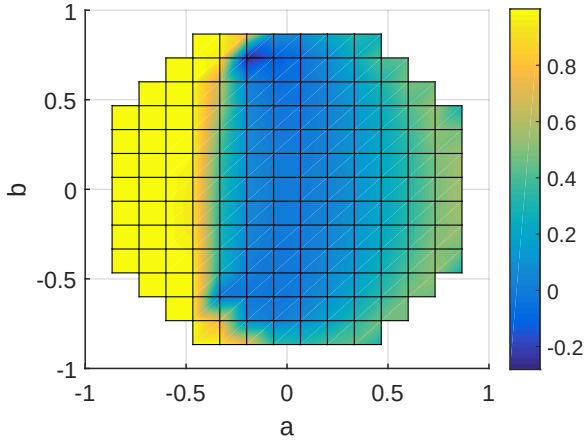
(a) The expectation value $\langle \bar{X} \rangle$ of the final state $\Psi(T)$, for initial Hamiltonians $H^{-}_{\text{triv}}(a,b)$ in the neighborhood of $H^{-}_{\text{triv}}(0,0) = -\sum_j Z_j$. Note that, as illustrated in Fig. 3.3b, the ground space of the toric code is reached for the whole parameter range; hence these values, together with the expectation values shown in Fig. 3.4b uniquely determine the state $\Psi(T)$.



(b) The quantity $\ln(1 - \langle \bar{Z} \rangle)$ for initial Hamiltonians $H^{-}_{\text{triv}}(a,b)$ (we plot the logarithm because the variation is small) as in Fig. 3.4a.



(c) The quantity $\ln(1 - \langle \bar{X} \rangle)$ for initial Hamiltonians $H^{-X}_{\text{triv}}(a,b)$ in the neighborhood of $H^{-X}_{\text{triv}}(0,0) = -\sum_j X_j$. The corresponding adiabaticity error is shown in Fig. 3.3c.



(d) The quantity $\langle \bar{Z} \rangle$ for initial Hamiltonians $H^{-X}_{\text{triv}}(a,b)$ as in Fig. 3.4c.

Figure 3.4: These figures illustrate the expectation values $\langle \bar{X} \rangle$ and $\langle \bar{Z} \rangle$ of string-operators (cf. (3.51)) of the final state $\Psi(T)$, for different choices of the initial Hamiltonian. The total evolution time is $T = 120$.

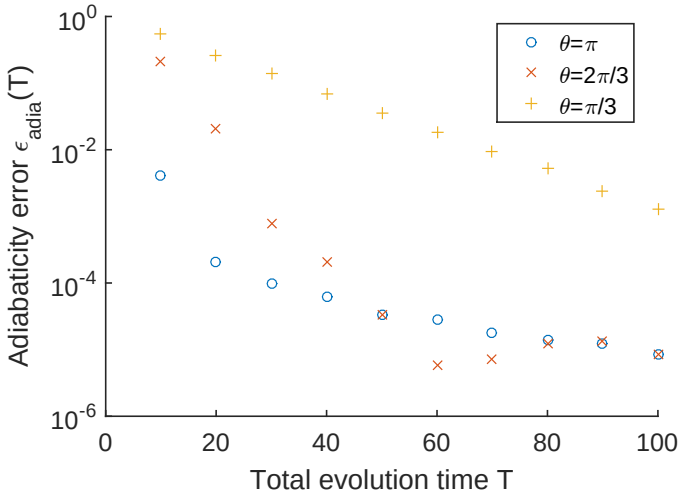


Figure 3.5: The adiabaticity error $\epsilon_{\text{adia}}(T)$ for the doubled semion model as a function of the total evolution time T. Initial Hamiltonians $H_{\text{triv}}(\theta)$ with $\theta \in \{\pi/3, 2\pi/3, \pi\}$ are considered.

3.7.4 The doubled semion model

In this section, we present our numerical results for Hamiltonian interpolation in the case of the doubled semion model (see Section 3.6.3).

(Non)-adiabaticity. We first consider the total evolution time T necessary to reach the final ground space of H_{top}, for different initial Hamiltonians H_{triv}. Specifically, Fig. 3.5 shows the adiabaticity error $\epsilon_{\text{adia}}(T)$ (cf. (3.49)) as a function of the total evolution time T for the three initial Hamiltonians $H_{\text{triv}}(\theta)$, $\theta \in \{\pi, \pi/3, 2\pi/3\}$ (cf. (3.46)). The case of $\theta = 0$, corresponding to the initial Hamiltonian $H_{\text{triv}}(0) = \sum_j Z_j$ is not shown in Fig. 3.5 since the situation is the same as in the toric code: No overlap with the ground space of H_{top} is achieved because the vertex-operators $A_v = Z^{\otimes 3}$ are conserved quantities with $\langle A_v \rangle = -1$.

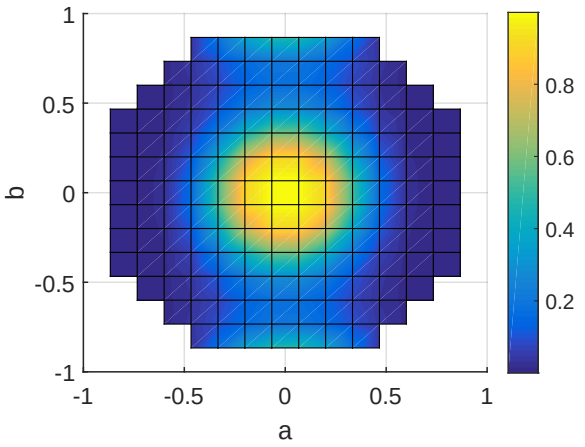
In Fig. 3.6a, we plot the adiabaticity error $\epsilon_{\text{adia}}(T)$ with initial Hamiltonian among the family of Hamiltonians $H^+_{\text{triv}}(a, b)$ in the vicinity of $H^+_{\text{triv}}(0, 0) = \sum_i Z_i$. Similarly, Fig. 3.6b provides the adiabaticity error for initial Hamiltonians $H^-_{\text{triv}}(a, b)$ in the vicinity of $H^-_{\text{triv}}(0, 0) = -\sum_i Z_i$.

Logical state. To explore the stability of the resulting final state, we consider the family of initial Hamiltonians $H^{\pm}_{\text{triv}}(a, b)$ and compute the overlap $|\langle \Psi^{\pm}_{a,b}(T) | \psi_R \rangle|^2$ of the resulting final state $\Psi^{\pm}_{a,b}(T)$ with a suitably chosen reference state ψ_R. We choose the latter as follows: ψ_R is the result of projecting the final state $\Psi^-_{0,0}(T)$ of the Hamiltonian interpolation, starting from the initial Hamiltonian $H^-_{\text{triv}}(0, 0) = -\sum_i Z_i$ onto the ground space of the doubled semion model H_{top} and normalizing, i.e.,

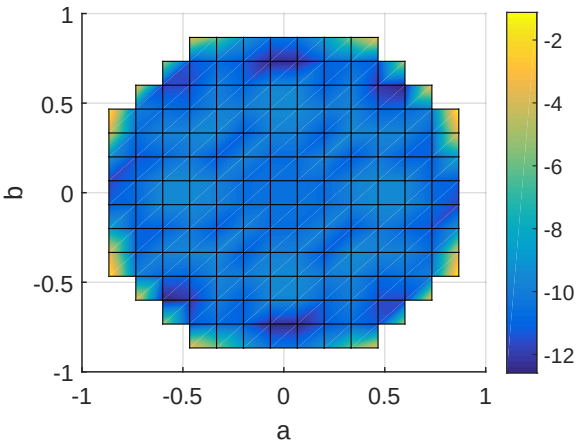
$$\psi_R = \frac{P_0 \Psi^-_{0,0}(T)}{\| P_0 \Psi^-_{0,0}(T) \|} \,. \tag{3.52}$$

We briefly remark that the state ψ_R is uniquely determined (up to a phase) as the unique simultaneous $+1$-eigenvector of TS^3TS (see Section 3.7.2) and the string operator $\bar{Z} = Z_1 Z_2$ (which is the string-operator $F_{(s,s)}(C)$ for the associated loop C when acting on the ground space of H_{top}): indeed, the latter operator commutes with both $H^-_{\text{triv}}(0, 0)$ and H_{top}. We also point out that, similarly to the toric code, the local Z_i-operators correspond to a combination of pair creation, hopping and pair annihilation of $(\boldsymbol{s}, \boldsymbol{s})$ anyons.

The preparation stability of the reference state ψ_R with respect to the initial Hamiltonians $H^{\pm}_{\text{triv}}(a, b)$ with negative and positive Z field component is illustrated in Fig. 3.7. For negative Z field (Fig. 3.7b) the resulting state $\Psi^-_{a,b}(T)$ has large overlap with the reference state ψ_R



(a) The adiabaticity error $\epsilon_{\text{adia}}(T)$ for different Hamiltonians $H^{+}_{\text{triv}}(a,b)$ in the vicinity of $H^{+}_{\text{triv}}(0,0) = \sum_j Z_j$. The adiabaticity error is maximal for the latter because of conserved quantities; however, it decays rapidly outside this center region. This situation is analogous to Fig. 3.3a for the toric code.



(b) The logarithmic adiabaticity error $\ln \epsilon_{\text{adia}}(T)$ among the family of Hamiltonians $H^{-}_{\text{triv}}(a,b)$ around $H^{-}_{\text{triv}}(0,0) = -\sum_j Z_j$.

Figure 3.6: The adiabaticity error $\epsilon_{\text{adia}}(T)$ for different initial Hamiltonians H_{triv} and the doubled semion model as H_{top}. In both cases, the total evolution time is $T = 120$.

for almost the entire parameter range. Even when starting from initial Hamiltonians with positive Z field component (Fig. 3.7a), where the final state does not have a large overlap with the topological ground space (see Fig. 3.6a), the ground space contribution comes almost exclusively from the reference state. Thus, for doubled semion model, we identify a single stable final state ψ_R corresponding to the initial Hamiltonian $H_{\text{triv}} = -\sum_i Z_i$.

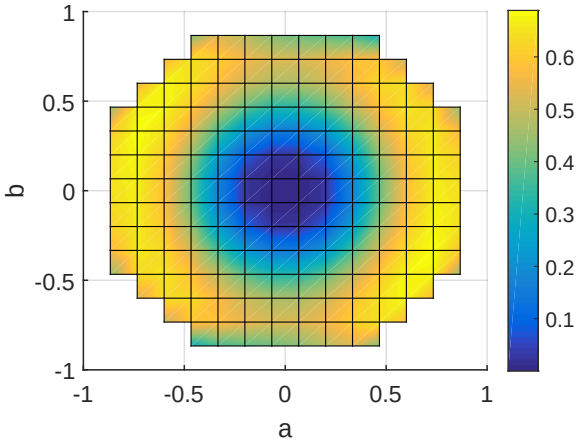
3.7.5 The doubled Fibonacci model

As our last case study of Hamiltonian interpolation, we consider the doubled Fibonacci model described in Section 3.6.3.

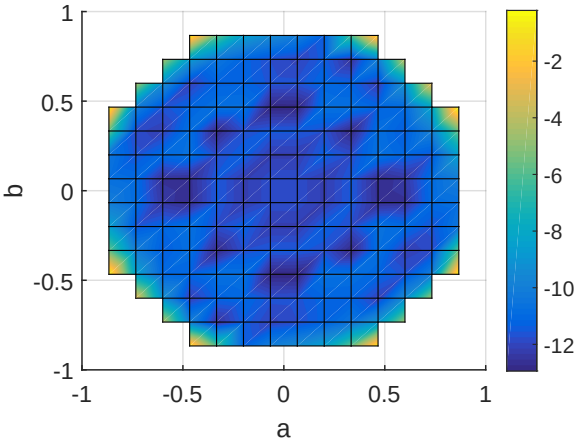
(Non)-adiabaticity. Fig. 3.8 shows the adiabaticity error $\epsilon_{\text{adia}}(T)$ as a function of the total evolution time T for the initial Hamiltonians $H^{\pm}_{\text{triv}} = \pm\sum_j Z_j$. Note that to achieve the same error, the total evolution time T needs to be much longer compared to the toric code and the doubled semion models. It also illustrates that an error of around $\epsilon_{\text{adia}}(T) \approx 10^{-3}$ is obtained for $T = 320$: the final state $\Psi(T)$ overlaps well with the ground space of H_{top}.

In Fig. 3.9, we consider the non-adiabaticity $t \mapsto \epsilon_{\text{adia}}(t)$ along the evolution, again for the initial Hamiltonians $H^{\pm}_{\text{triv}} = \pm\sum_j Z_j$. In particular, Fig. 3.9a, which is for a total evolution time of $T = 320$, we see that the deviation of the state $\Psi(t)$ from the instantaneous ground state of $H(t)$ can be much larger (compared to the non-adiabaticity $\epsilon_{\text{adia}}(T)$) along the evolution, even when approaching the end of Hamiltonian interpolation: we have $\epsilon_{\text{adia}}(t) \gtrsim 10^{-2}$ for $t \approx 280$. The fact that the ground space of the final Hamiltonian H_{top} is reached nevertheless at time $t \approx T$ is essentially due to the exact degeneracy in the final Hamiltonian H_{top}: In fact, the system is in a state which has a large overlap with the subspace of 'low energy' (corresponding to the 4-fold degenerate subspace of H_{top}) along the trajectory, but not necessarily with the unique instantaneous ground state of $H(t)$ for $t < T$. For $t = T$, the state has a large overlap with the ground space of H_{top} since the latter is higher-dimensional.

This illustrates that the adiabaticity error $t \mapsto \epsilon_{\text{adia}}(t)$ along the evolution (i.e., for $t < T$) does not provide sufficient information to conclude that the ground space of H_{top} is reached at the end of the evolution. Due to the small energy splitting within the topological "phase" it is more fruitful



(a) The overlap $|\langle\Psi^{+}_{a,b}(T)|\psi_R\rangle|^2$ for initial Hamiltonians $H^{+}_{\mathsf{triv}}(a,b)$ around $H^{+}_{\mathsf{triv}}(0,0) = \sum_i Z_i$. We observe that outside the center region (where the ground space of H_{top} is not reached, see Fig. 3.6a), the prepared state $\Psi^{+}_{a,b}(T)$ is not too far from the reference state ψ_R. Note that definition of the latter does not correspond to any Hamiltonian in this plot, but rather the centerpoint of Fig. 3.7b.



(b) The quantity $\ln(1 - |\langle\Psi^{-}_{a,b}(T)|\psi_R\rangle|^2)$ for initial Hamiltonians $H^{-}_{\mathsf{triv}}(a,b)$ around $H^{-}_{\mathsf{triv}}(0,0) = -\sum_i Z_i$. We plot the logarithm of this quantity because the variation is small. As illustrated, the resulting state is close to the reference state ψ_R throughout most of the parameter region. Observe that, while ψ_R corresponds to the center point in this figure, it still deviates from $\Psi^{-}_{a,b}$ since the latter has support outside the ground space of H_{top} (cf. Fig. 3.6b).

Figure 3.7: The overlaps $|\langle\Psi^{\pm}_{a,b}(T)|\psi_R\rangle|^2$ between the final states $\Psi^{\pm}_{a,b}(T)$ of Hamiltonian interpolation and the reference state ψ_R (cf. (3.52)). Observe that the same reference state is used in both figures even though ψ_R is naturally associated with the centerpoint in Fig. 3.7b. The total evolution time is $T = 120$ in both cases. Comparing with Figs. 3.6a and 3.6b, we conclude that throughout the region where the ground space of H_{top} is reached, approximately same state is prepared.

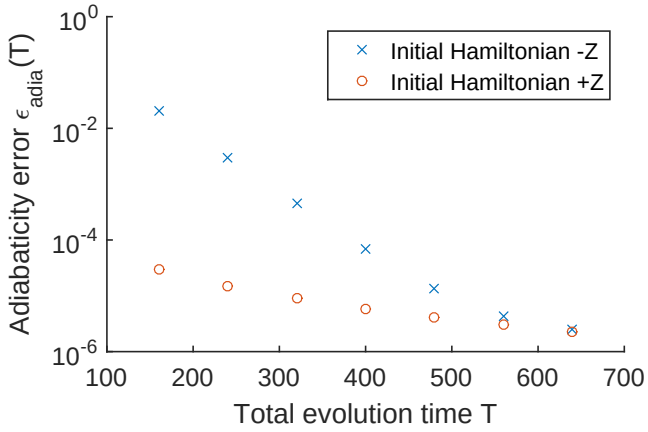


Figure 3.8: The adiabaticity error $\epsilon_{\text{adia}}(T)$ with respect to different total evolution times T for the Fibonacci model. The initial Hamiltonian H_{triv} is either $H^{+}_{\text{triv}}(0,0) = \sum_i Z_i$ or $H^{-}_{\text{triv}}(0,0) = -\sum_i Z_i$. Note that for this choice of initial Hamiltonians, the vertex terms A_v are conserved quantities (as for example in the toric code). Since both $|1\rangle^{\otimes 3}$ and $|\tau\rangle^{\otimes 3}$ are in the ground space of A_v, both signs of the pure Z field lead to a Hamiltonian interpolation which invariantly remains in the ground space of A_v. In other words, the adiabaticity error stems from the plaquette terms. Other fields are computationally more costly, since they lift the block decomposition of the interpolating Hamiltonians $H(t)$ induced by the conserved vertex terms, reducing the sparsity of the unitary evolution.

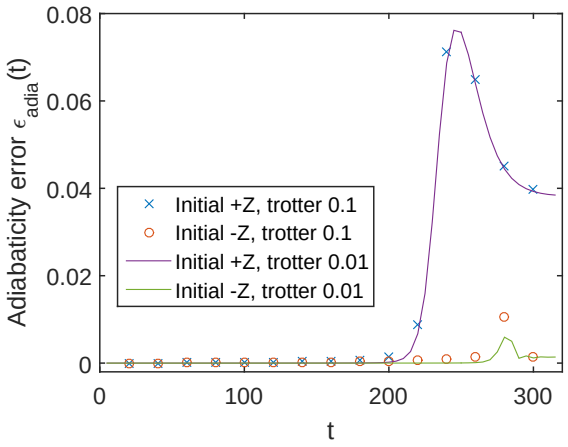
to view the part of the interpolation close to $t \approx T$ in terms of degenerate adiabatic perturbation theory [RO14] instead of the traditional adiabatic theorem.

Fig. 3.9a also shows that for $T = 320$, changing the Trotter time steps Δt (cf. (3.48)) from $\Delta t = 0.1$ to $\Delta t = 0.01$ does not significantly change the behavior, particularly for the initial Hamiltonian $\sum_i Z_i$. On the other hand, by increasing the Hamiltonian interpolation time for $H_{\text{triv}} = \sum_i Z_i$ to $T = 1280$, as in Fig. 3.9b, we see the evolution closely follows the instantaneous ground state. The discrepancy can be seen as a "lag" or delay of the evolved state and the instantaneous ground state and is largest at the "phase transition", $H(t) \approx 1/4 H_{\text{top}} + 3/4 \sum_i Z_i$, where the gap closes.

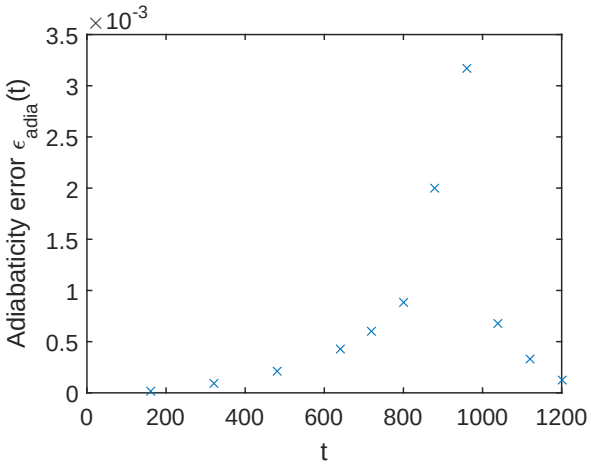
Logical state. Fig. 3.10 provides information about the final state $\Psi^{\pm}_{a,b}(T)$ of Hamiltonian interpolation, for the family of initial Hamiltonians $H^{\pm}_{\text{triv}}(a,b)$ (cf. (3.50)). Again, the figure gives the overlap with a single reference state ψ_R. Similarly as before, we choose the latter as the final state of Hamiltonian interpolation, starting with initial Hamiltonian $H^{-}_{\text{triv}}(0,0) = -\sum_i Z_i$, and subsequently projected into the ground space and normalized (cf. (3.52)).

We observe significant overlap of the final state with the reference state ψ_R for the whole parameter range for the initial Hamiltonians $H^{-}_{\text{triv}}(a,b)$ (Fig. 3.10b). In contrast, for the initial Hamiltonians $H^{+}_{\text{triv}}(a,b)$, the final state depends strongly on the choice of parameters (a,b) (Fig. 3.10a).

To relate this to the discussion in Section 3.6 (respectively Conjecture 1), let us first consider the centerpoint of Fig. 3.10b associated with the initial Hamiltonian $H^{-}_{\text{triv}}(0,0) = -\sum_j Z_j$. These terms correspond to a combination of local pair creation, hopping and pair annihilation of $(\boldsymbol{\tau}, \boldsymbol{\tau})$ anyons, as explained in Appendix 3.B. The effective Hamiltonian can be computed at this point based on expression (3.35) and the S- and T-matrices given in Eq. (3.45). The result is given numerically in Eq. (3.100) in the appendix. Computing the ground state ψ_{eff} of this effective Hamiltonian, we observe that with respect to the projections $\{P_{1,1}, P_{\tau,\tau}, P_{1,\tau}, P_{\tau,1}\}$, the expectation values of the reference state ψ_R and ψ_{eff} are similar,



(a) The total evolution time is $T = 320$. As explained in the text, the fact that the overlap with the instantaneous ground state towards the end of the evolution is small does not prevent the system from reaching the degenerate ground space of the final Hamiltonian H_{top} (see 3.8). Changing the Trotter discretization step from $\Delta t = 0.1$ to $\Delta t = 0.01$ does not significantly change the behavior.



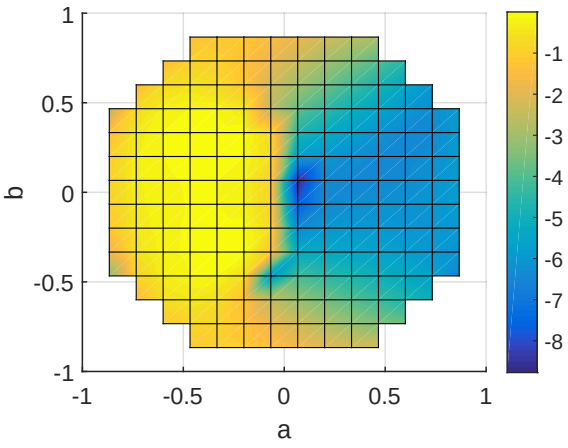
(b) For the initial Hamiltonian $H_{\text{triv}} = \sum_i Z_i$ and a total evolution time $T = 1280$, the system closely follows the instantaneous ground state of $H(t)$.

Figure 3.9: The overlap of the state $\Psi(t)$ at time t with the instantaneous ground space of $H(t)$, as expressed by the adiabaticity error $t \mapsto \epsilon_{\text{adia}}(t)$ along the evolution. The initial Hamiltonian is either $H^{+}_{\text{triv}}(0,0) = \sum_i Z_i$ or $H^{-}_{\text{triv}}(0,0) = -\sum_i Z_i$, and the final Hamiltonian H_{top} is the doubled Fibonacci model.

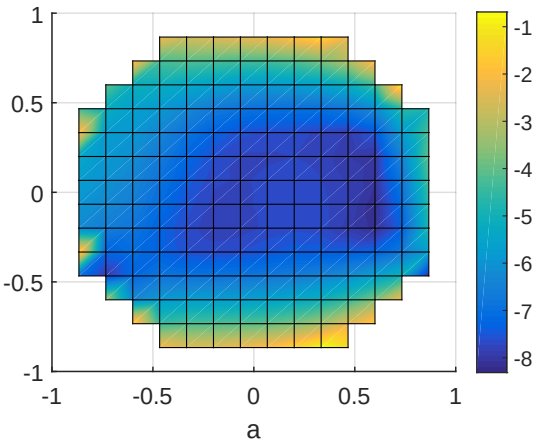
	$P_{1,1}$	$P_{\tau,\tau}$	$P_{1,\tau}$	$P_{\tau,1}$
ψ_R	0.5096	0.4838	0.0033	0.0033
ψ_{eff}	0.5125	0.4804	0.0036	0.0036

Moving away from the center point in Fig. 3.10b, we compute the overlaps of the reference state ψ_R with the ground states $\psi^{\pm}_{\text{pert}}(a,b)$ of perturbed Hamiltonians of the form $H^{\pm}_{\text{pert}}(a,b) = H_{\text{top}} \pm 0.001 H^{\pm}_{\text{triv}}(a,b)$, as illustrated in Fig. 3.11 (the latter providing an approximate notion of effective Hamiltonians). The figure illustrates that these perturbed states have, as expected, a certain degree of stability with respect to the parameters (a,b). Comparison with Fig. 3.10 thus points to a certain discrepancy between the behavior of perturbed states and states obtained by Hamiltonian interpolation: Fig. 3.10a shows high sensitivity of the final state to initial parameters (a,b) (which is absent in the perturbative prediction), whereas Fig. 3.10b shows that the final state is close to the reference state ψ_R throughout (as opposed to the perturbative prediction, where this is not the case along the boundary). To rule out that this discrepancy stems from an insufficiently large choice of the total evolution time T, we also show that different choices of the total evolution time T do not significantly affect the overlap with the reference state along the line $b = 0$, see Fig. 3.12.

In summary, we conclude that while for a large parameter range of initial parameters the reference state ψ_R is indeed reached, the stability property is less pronounced than for the toric code and the doubled semion models. In addition, a naïve comparison with ground states of perturbed Hamiltonians suggests that the description via effective Hamiltonians does not capture all relevant features. We conjecture that higher orders in perturbation theory are needed to provide more information in the case of the Fibonacci model: the state may be "locked" in eigenstates of such higher-order Hamiltonians before the lowest order effective Hamiltonian dominates.

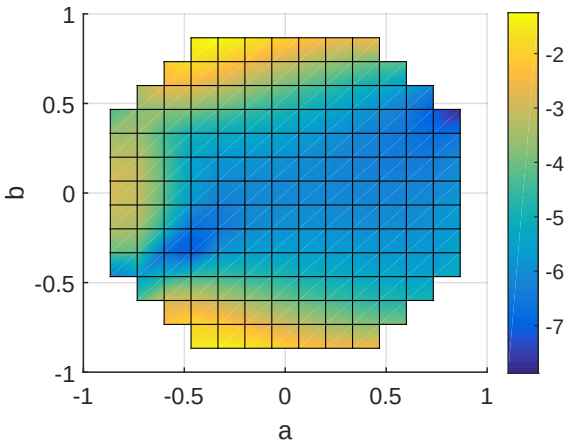


(a) The quantity $\ln(1-|\langle\Psi^+_{a,b}(T)|\psi_R\rangle|^2)$ for initial Hamiltonian of the form $H^+_{\mathsf{triv}}(a,b)$ around $H^+_{\mathsf{triv}}(0,0) = \sum_j Z_j$. For the whole range of parameters (a,b), the adiabaticity error is small, $\epsilon_{\text{adia}}(T) \leq 10^{-4}$. The reference state ψ_R corresponds to the center of Fig. 3.10b (up to projection onto the ground space of H_{top} and normalization). The figure illlustrates that the final state $\Psi^+_{a,b}(T)$ has non-trivial overlap with the reference state in the region $a > 0$, but is very sensitive to the choice of parameters (a,b), especially around $(a,b) = (0,0)$.

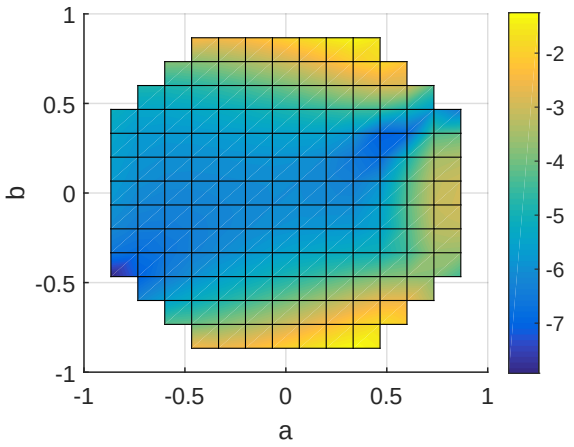


(b) The quantity $\ln(1-|\langle\Psi^-_{a,b}(T)|\psi_R\rangle|^2)$, for initial Hamiltonians $H^-_{\mathsf{triv}}(a,b)$ around $H^-(0,0) = -\sum_j Z_j$. For the whole range of parameters (a,b), the adiabaticity error is small, $\epsilon_{\text{adia}}(T) \leq 0.005$ apart from the point on the boundary of the plot. The Hamiltonian interpolation reaches the reference state ψ_R essentially for the full parameter range.

Figure 3.10: These figures show the overlap between the final states $\Psi^\pm_{a,b}$ of Hamiltonian interpolation and the reference state ψ_R. This is for the family $H^\pm_{\mathsf{triv}}(a,b)$ of initial Hamiltonians and the double Fibonacci model H_{top} as the final Hamiltonian. The reference state ψ_R is chosen in both figures as in (3.52) (corresponding to the center point in Fig. 3.10b). The total evolution time is $T = 320$ in both cases.



(a) The quantity $\ln(1 - |\langle\psi^{+}_{\mathsf{pert}}(a,b)|\psi_R\rangle|^2)$ for Hamiltonians $H^{+}_{\mathsf{pert}}(a,b)$.



(b) The quantity $\ln(1 - |\langle\psi^{-}_{\mathsf{pert}}(a,b)|\psi_R\rangle|^2)$ for Hamiltonians $H^{-}_{\mathsf{pert}}(a,b)$. The reference state ψ_R has overlap $|\langle\psi^{-}_{\mathsf{pert}}(0,0)|\psi_R\rangle|^2 \approx 0.9976$ with the ground state of the perturbed Hamiltonian $H^{-}_{\mathsf{pert}}(0,0) = H_{\mathsf{top}} - 0.001\sum_j Z_j$.

Figure 3.11: To compare with the perturbative prediction, these figures give the overlap between the reference state ψ_R and the ground state $\psi_{\mathsf{pert}}(a,b)$ of the perturbed Hamiltonians $H^{\pm}_{\mathsf{pert}}(a,b) = H_{\mathsf{top}} \pm 0.001 H^{+}_{\mathsf{triv}}(a,b)$.

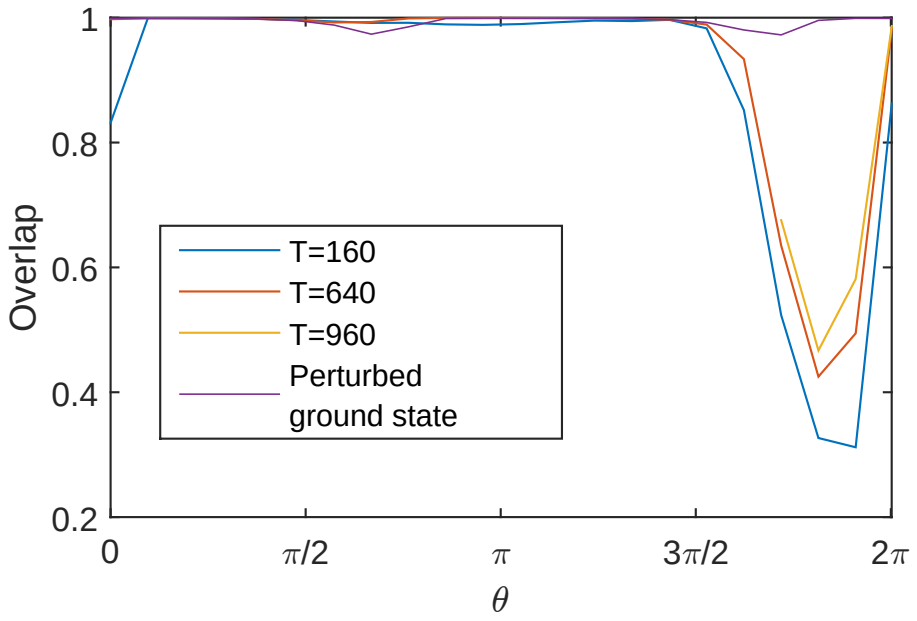


Figure 3.12: This figure shows the overlap $|\langle\Psi_{\mathsf{pert}}(\theta)|\psi_R\rangle|^2$ for initial Hamiltonians $H_{\mathsf{triv}}(\theta) = \sum_j \cos\theta Z_j + \sin\theta X_j$ along the line of the horizontal axis in Fig. 3.11a, for different values of the total evolution time T. We only compute $T = 960$ on the rightmost region to show that increasing the total evolution time do not significantly change the final states. It also gives the corresponding overlap $|\langle\Psi_{\mathsf{pert}}(\theta)|\psi_R\rangle|^2$ between the ground state of $H_{\mathsf{pert}}(\theta)$ and the reference state. The figure illustrates that increasing the evolution time T does not significantly alter the overlap with the reference state.

Appendix

3.A Equivalence of the self-energy- and Schrieffer-Wolff methods for topological order

As discussed in Section 3.3.4, here we show that at lowest non-trivial order, the expressions obtained from the self-energy-method and the Schrieffer-Wolff method coincide if the Hamiltonian and perturbation satisfies a certain topological order condition.

We begin with a review of the exact Schrieffer-Wolff transformation (Section 3.A.1), as well as the expressions resulting from the Schrieffer-Wolff perturbative expansion (Section 3.A.2). In Section 3.A.3, we present some preliminary computations. In Section 3.A.4, we introduce the topological order constraint and establish our main result.

3.A.1 Exact-Schrieffer-Wolff transformation

As mentioned in Section 3.3, the Schrieffer-Wolff method provides a unitary U such that

$$H_{\mathsf{eff}} = U(H_0 + \epsilon V)U^\dagger \tag{3.53}$$

preserves the ground space $P_0\mathcal{H}$ of H_0, and can be considered as an effective Hamiltonian. The definition of the unitary is as follows: let P be the projection onto the ground space of the perturbed Hamiltonian $H_0 + \varepsilon V$. Defining the reflections

$$R_{P_0} = 2P_0 - I$$
$$R_P = 2P - I$$

the (exact) Schrieffer-Wolff transformation is defined by the "direct rotation"

$$U = \sqrt{R_{P_0} R_P}\ , \tag{3.54}$$

where the square root is defined with a branch cut along the negative real axis. The effective Hamiltonian is then given by

$$H_{\mathsf{eff}}(\epsilon) = P_0 U(H_0 + \varepsilon V)U^\dagger P_0\ .$$

A variational characterization (see [BDL11]) of the unitary U (instead of (3.54)) is often more useful (e.g., for computing the effective Hamiltonian in the case of a two-dimensional ground space, such as for the Majorana chain): we have

$$U = \arg\min \left\{ \|I - U\|_2 \;\middle|\; U \text{ unitary and } UPU^\dagger = P_0 \right\}\ , \tag{3.55}$$

where $\|A\|_2 = \sqrt{\mathrm{tr}(A^\dagger A)}$ is the Frobenius norm.

3.A.2 The perturbative SW expansion

Since the transforming unitary (3.54), as well as expression (3.53), are difficult to compute in general, a standard approach is to derive systematic series in the parameter ϵ (the perturbation strength). In this section, we summarize the expressions for this explicit perturbative expansion

of the Schrieffer-Wolff effective Hamiltonian obtained in [BDL11]. The perturbation is split into diagonal and off-diagonal parts according to

$$V_{\rm d} = P_0 V P_0 + Q_0 V Q_0 =: \mathcal{D}(V) \tag{3.56}$$
$$V_{\rm od} = P_0 V Q_0 + Q_0 V P_0 =: \mathcal{O}(V) \ . \tag{3.57}$$

where P_0 is the projection onto the ground space of H_0, and $Q_0 = I - P_0$ the projection onto the orthogonal complement. Assuming that $\{|i\rangle\}_i$ is the eigenbasis of H_0 with energies $H|i\rangle = E_i|i\rangle$, one introduces the superoperator

$$\mathcal{L}(X) = \sum_{i,j} \frac{\langle i|\mathcal{O}(X)|j\rangle}{E_i - E_j} |i\rangle\langle j| \ .$$

Then the operators S_j are defined recursively as

$$\begin{aligned} S_1 &= \mathcal{L}(V_{\rm od}) \\ S_2 &= -\mathcal{L}(\mathsf{Ad}_{V_{\rm d}}(S_1)) \\ S_n &= -\mathcal{L}(\mathsf{Ad}_{V_{\rm d}}(S_{n-1})) + \sum_{j\geq 1} a_{2j} \mathcal{L}(\hat{S}^{2j}(V_{\rm od})_{n-1}) \ , \end{aligned} \tag{3.58}$$

where

$$\hat{S}^k(V_{\rm od})_m = \sum_{\substack{n_1,\ldots,n_k\geq 1 \\ \sum_{r=1}^k n_r = m}} \mathsf{Ad}_{S_{n_1}} \cdots \mathsf{Ad}_{S_{n_k}}(V_{\rm od}) \ , \tag{3.59}$$

and where $\mathsf{Ad}_S(X) = [S, X]$. The constants are $a_m = \frac{2^m \beta_m}{m!}$, where β_m is the m-th Bernoulli number. Observe that

$$\hat{S}^k(V_{\rm od})_m = 0 \qquad \text{for } k > m \ .$$

The q-th order term in the expansion (3.6) is

$$H_{\mathsf{eff},q} = \sum_{1\leq j\leq \lfloor q/2 \rfloor} b_{2j-1} P_0 \hat{S}^{2j-1}(V_{\rm od})_{q-1} P_0 \ , \tag{3.60}$$

where $b_{2n-1} = \frac{2(2^n-1)\beta_{2n}}{(2n)!}$.

Since our main goal is to apply the perturbation theory to topologically ordered (spin) systems, we can try to utilize their properties. In particular, one defining property of such systems is that, if an operator is supported on a topological trivial region, then it acts trivially inside the ground space. A common non-trivial operation in the ground space corresponds to the virtual process of tunneling an anyon around the torus. This property will allow us to simplify the computation when we want to compute the lowest order effective Hamiltonian. In the following subsections, we will show that although S_n is defined recursively based on $S_1, \ldots, S_{n-1}$, only the first term $-\mathcal{L}(\mathsf{Ad}_{V_{\rm d}}(S_{n-1}))$ on the rhs of (3.58) would contribute to the lowest order effective Hamiltonian. The intuition behind this claim is that the other term $\sum_{j\geq 1} a_{2j} \mathcal{L}(\hat{S}^{2j}(V_{\rm od})_{n-1})$ corresponds to virtual processes which go through the ground space $\to$ excited space $\to$ ground space cycle multiple times (larger than one). It is intuitive that such virtual processes would not happen when we want to consider the lowest order perturbation.

3.A.3 Some preparatory definitions and properties

Let

$$G(z) = (zI - H_0)^{-1}$$

be the resolvent of the unperturbed Hamiltonian H_0. Let E_0 be the ground space energy of H_0. We set

$$G = G(E_0) = G(E_0)Q_0 = Q_0 G(E_0) Q_0 \ ,$$

i.e., the inverse is taken on the image of Q_0. Then $\mathcal{L}$ can be written as

$$\mathcal{L}(X) = P_0 X G - G X P_0 \ . \tag{3.61}$$

To organize the terms appearing in the perturbative Schrieffer-Wolff expansion, it will be convenient to introduce the following subspaces of operators.

Definition 3.A.1. *For each $n \in \mathbb{N}$, let $\Gamma(n)$ be the linear span of operators of the form*

$$Z_0 V Z_1 V Z_2 \cdots Z_{n-1} V Z_n \ , \tag{3.62}$$

where for each $j = 0, \ldots, n$, the operator Z_j is either one of the projections P_0 or Q_0, or a positive power of G, i.e., $Z_j \in \{P_0, Q_0\} \cup \{G^m \ | m \in \mathbb{N}\}$.

Let $\Gamma^\star(n) \subset \Gamma(n)$ the span of operators of the form (3.62) *which additionally satisfy the condition*

$$Z_0 Z_n = Z_n Z_0 = 0 \ ,$$

i.e., Z_0 and Z_n are orthogonal.

For later reference, we remark that operators in $\Gamma^\star(n)$ a linear combinations of certain terms which are off-diagonal with respect the ground space of H_0 (and its orthogonal complement). In particular, any product of an even number of these operators is diagonal.

The first observation is that the summands the effective Hamiltonian (3.60) have this particular form.

Lemma 3.A.2. *We have*

$$V_{od} \in \Gamma^\star(1) \tag{3.63}$$

and

$$S_n \in \Gamma^\star(n) \text{ for every } n \in \mathbb{N} \tag{3.64}$$

Furthermore,

$$\hat{S}^k (V_{od})_m \in \Gamma(m+1) \tag{3.65}$$

for all k, m.

Proof. The definition of $\Gamma(n)$ immediately implies that

$$XY \in \Gamma(n_1 + n_2) \qquad \text{for } X \in \Gamma(n_1) \text{ and } Y \in \Gamma(n_2) \ . \tag{3.66}$$

Thus

$$\mathsf{Ad}_X(Y) \in \Gamma(n_1 + n_2) \qquad \text{for } X \in \Gamma(n_1) \text{ and } Y \in \Gamma(n_2) \ .$$

Furthermore, inspecting the Definitions (3.57) and (3.56), we immediately verify that

$$V_{\mathrm{od}} \in \Gamma(1) \qquad \text{and} \qquad V_{\mathrm{d}} \in \Gamma(1) \tag{3.67}$$

Similarly, (3.63) follows directly from the definitions.

We first argue that

$$S_1 \in \Gamma(1) \qquad \text{and} \qquad S_2 \in \Gamma(2) \ . \tag{3.68}$$

Inserting the definition of $V_{\rm od}$ and $\mathcal{L}$ (that is, (3.61)), we have

$$\begin{aligned} S_1 &= \mathcal{L}(P_0VQ_0 + Q_0VP_0) \\ &= P_0(P_0VQ_0 + Q_0VP_0)G - G(P_0VQ_0 + Q_0VP_0)P_0 \\ &= P_0VG - GVP_0 \ , \end{aligned} \tag{3.69}$$

where we used the fact $GQ_0 = Q_0G = G$ and that Q_0, P_0 are orthogonal projections. This proves the claim (3.64) for $n = 1$ and, in particular, shows that $S_1 \in \Gamma(1)$.

Similarly, for $n = 2$, using the definition of $V_{\rm d}$, a straightforward calculation (using (3.69)) gives

$$\mathsf{Ad}_{V_{\rm d}}(S_1) = (P_0VP_0VG - Q_0VGVP_0) + h.c.$$

(where $h.c.$ denotes the Hermitian conjugate of the previous expression) and thus with (3.61)

$$S_2 = (P_0VP_0VG^2 + GVGVP_0) - h.c. \ .$$

We conclude that (3.64) holds $n = 2$ and, in particular, $S_2 \in \Gamma(2)$, as claimed (Eq. (3.68)).

With (3.67) and (3.68), we can use the composition law (3.66) to show inductively that

$$S_n \in \Gamma(n) \qquad \text{for all } n \in \mathbb{N} \ . \tag{3.70}$$

Indeed, (3.70) holds for $n = 1, 2$. Furthermore, assuming $S_m \in \Gamma(m)$ for all $m \leq n-1$, we can apply (3.66) and (3.67) to the Definition (3.59) of $\hat{S}^{2j}(V_{\rm od})_{n-1}$, obtaining

$$\hat{S}^{2j}(V_{\rm od})_{n-1} \in \Gamma(n) \qquad \text{and} \qquad \mathsf{Ad}_{V_d}(S_{n-1}) \in \Gamma(n) \ .$$

Thus (3.70) follows by definition (3.58) of S_n, the easily verified fact (cf. (3.61)) that $\mathcal{L}(\Gamma(n)) \subset \Gamma(n)$, and linearity.

Finally, observe that (3.61) also implies

$$\mathcal{L}(\Gamma(n)) \subset \Gamma^\star(n) \ , \tag{3.71}$$

hence (3.64) follows with (3.70).

The claim (3.65) is then immediate from the composition law (3.66), as well as (3.70) and (3.67). □

3.A.4 Topological-order constraint

In the following, we will assume that

$$P_0\Gamma(n)P_0 \subset \mathbb{C}P_0 \qquad \text{for all } n < L \ . \tag{3.80}$$

which amounts to saying that (H_0, V) satisfies the topological order condition with parameter L (see Definition 3.3.1). In Section 3.A.4, we argue that this implies that the effective Hamiltonian is trivial (i.e., proportional to P_0) for all orders $n < L$. In Section 3.A.4, we then compute the non-trivial contribution of lowest order.

Triviality of effective Hamiltonian at orders $n < L$

A simple consequence of Definition 3.A.1 then is the following.

Lemma 3.A.3. *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then for any $2k$-tuple of integers $n_1, \ldots, n_{2k} \in \mathbb{N}$ with*

$$\sum_{j=1}^{2k} n_j < L \ ,$$

and all $T_{n_j} \in \Gamma^\star(n_j)$, $j = 1, \ldots, 2k$, we have

$$\begin{aligned} T_{n_1} \cdots T_{n_{2k}} P_0 &\in \mathbb{C}P_0 \\ P_0 T_{n_1} \cdots T_{n_{2k}} &\in \mathbb{C}P_0 \ . \end{aligned}$$

Proof. It is easy to check that because of property (3.66), the expression $T_{n_1}\cdots T_{n_{2k}}P_0$ is contained in $P_0\Gamma(n)P_0$, where $n=\sum_{j=1}^{2k} n_j$. The claim follows immediately. The argument for $P_0T_{n_1}\cdots T_{n_{2k}}$ is identical. □

Lemma 3.A.4. *Assume that $P_0\Gamma(n)P_0\subset\mathbb{C}P_0$ for all $n<L$. Then*

$$\begin{aligned} &P_0\hat{S}^{2j-1}(V_{od})_{n-1}P_0\in\mathbb{C}P_0 \qquad \text{for all } j \text{ and all } n<L\ . \\ &P_0\hat{S}^{2j-1}(V_{od})_{L-1}P_0\in\mathbb{C}P_0 \qquad \text{for all } j>1\ . \end{aligned} \tag{3.72}$$

Lemma (3.A.4) suffices to show that the n-th order effective Hamiltonian $H_{\mathsf{eff}}^{(n)}$ is trivial (i.e., proportional to P_0) for any order $n<L$ (see Theorem 3.3.2 below).

Proof. The claim (3.72) is an immediate consequence of the assumption since $\hat{S}^{2j-1}(V_{\mathrm{od}})_{n-1}\in\Gamma(n)$ according to (3.65) of Lemma 3.A.2.

For $j>1$, we use the definition

$$\hat{S}^{2j-1}(V_{\mathrm{od}})_{L-1}=\sum_{\substack{n_1,\dots,n_{2j-1}\geq 1\\ \sum_{r=1}^{2j-1}n_r=L-1}}\mathsf{Ad}_{S_{n_1}}\cdots\mathsf{Ad}_{S_{n_{2j-1}}}(V_{\mathrm{od}})\ .$$

First summing over n_1 (using the linearity of $\mathsf{Ad}_{S_{n_1}}$), we obtain

$$\begin{aligned} \hat{S}^{2j-1}(V_{\mathrm{od}})_{n-1}&=\sum_{n_1\geq 1}\mathsf{Ad}_{S_{n_1}}(Y_{n_1}) \quad \text{where}\\ Y_{n_1}&=\sum_{\substack{n_2,\dots,n_{2j-1}\geq 1\\ \sum_{r=2}^{2j-1}n_r=L-1-n_1}}\mathsf{Ad}_{S_{n_2}}\cdots\mathsf{Ad}_{S_{n_{2j-1}}}(V_{\mathrm{od}}) \end{aligned}$$

Observe that Y_{n_1} is a linear combination of products $T_1\cdots T_{2j-1}$ of an odd number $2j-1$ of elements $\{T_r\}_{r=1}^{2j-1}$, where $(T_1,\dots,T_{2j-1})$ is a permutation of $(S_{n_2},\dots,S_{n_{2j-1}},V_{\mathrm{od}})$. By linearity, it suffices to show that $P_0\mathsf{Ad}_{S_{n_1}}(T_1\cdots T_{2j-1})P_0\in\mathbb{C}P_0$ for such a product.

We will argue that

$$T_1\cdots T_{2j-1}P_0=TP_0 \qquad \text{for some } T\in\Gamma(m) \qquad \text{with } m<L-n_1 \text{ and} \tag{3.73}$$

$$P_0T_1\cdots T_{2j-1}=P_0T' \qquad \text{for some } T'\in\Gamma(m') \qquad \text{with } m'<L-n_1\ . \tag{3.74}$$

This implies the claim since

$$\begin{aligned} P_0\mathsf{Ad}_{S_{n_1}}(T_1\cdots T_{2j-1})P_0&=P_0S_{n_1}T_1\cdots T_{2j-1}P_0-P_0T_1\cdots T_{2j-1}S_{n_1}P_0\\ &=P_0S_{n_1}TP_0-P_0T'S_{n_1}P_0\\ &\in\mathbb{C}P_0 \end{aligned}$$

where we used that $S_{n_1}T\in\Gamma(n_1+m)$ and $T'S_{n_1}\in\Gamma(n_1+m')$, $n_1+m<L$, $n_1+m'<L$ and our assumption in the last step.

To prove (3.73) (the proof of (3.74) is analogous and omitted here), we use that $S_{n_j}\in\Gamma^\star(n_j)$ and $V_{\mathrm{od}}\in\Gamma^\star(1)$ according to Lemma 3.A.2. In other words, there are numbers $m_1,\dots,m_{2j-1}\geq 1$ with $\sum_{r=1}^{2j-1}m_r=1+\sum_{r=2}^{2j-1}n_r=L-n_1<L$ such that $T_r\in\Gamma^\star(m_r)$ for $r=1,\dots,2j-1$. In particular, with Lemma 3.A.3, we conclude that

$$\begin{aligned} T_1\cdots T_{2j-1}P_0&=T_1(T_2\cdots T_{2j-1})P_0\\ &\in\mathbb{C}T_1P_0\ . \end{aligned}$$

Since $m_1=L-n_1-\sum_{r=2}^{2j-1}m_r<L-n_1$, the claim (3.73) follows. □

Computation of the first non-trivial contribution

Lemma (3.A.4) also implies that the first (potentially) non-trivial term is of order L, and given by $P_0\hat{S}^1(V_{\mathrm{od}})_{L-1}P_0$. Computing this term requires some effort.

Let us define the superoperator $\mathcal{V}_d = -\mathcal{L}\circ\mathsf{Ad}_{V_{\mathrm{d}}}$, that is,

$$\mathcal{V}_d(X) = \mathcal{L}(XV_{\mathrm{d}} - V_{\mathrm{d}}X)$$

For later reference, we note that this operator satisfies

$$\mathcal{V}_d(\Gamma^\star(n)) \subset \Gamma^\star(n+1)\ . \tag{3.75}$$

as an immediate consequence of (3.71).

We also define the operators

$$B_n = \sum_{j\geq 1} a_j \mathcal{L}(\hat{S}^{2j}(V_{\mathrm{od}})_{n-1}) \tag{3.76}$$

Then we can rewrite the recursive definition (3.58) of the operators S_n as

$$\begin{aligned} S_1 &= \mathcal{L}(V_{\mathrm{od}}) \\ S_n &= \mathcal{V}_d(S_{n-1}) + B_n = A_n + B_n \qquad \text{for } n\geq 2\ , \end{aligned}$$

where we also introduced

$$A_n = \mathcal{V}_d(S_{n-1}) \qquad \text{for } n\geq 2\ . \tag{3.77}$$

Similarly to Lemma 3.A.4, we can show the following:

Lemma 3.A.5. *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then for any*

$$Y = \begin{cases} Z_0V_dZ_1V_dZ_2\cdots Z_{m-1}V_dZ_m & \text{for } m>0 \\ Z_0 & \text{for } m=0 \end{cases}$$

where $Z_j \in \{P_0, Q_0\}\cup\{G^k\ |k\in\mathbb{N}\}$, we have

$$P_0B_\ell Y V_{od}P_0 \in \mathbb{C}P_0 \qquad \text{and} \qquad P_0V_{od}YB_\ell P_0 \in \mathbb{C}P_0 \tag{3.78}$$

for all ℓ, m satisfying $\ell + m - 1 < L$.

Proof. By definition (3.76), B_ℓ is a linear combination of terms of the form $\mathcal{L}(\hat{S}^{2j}(V_{\mathrm{od}})_{\ell-1})$ with $j\geq 1$, which in turn (cf. (3.59)) is a linear combination of expressions of the form

$$\mathcal{L}\left(\mathsf{Ad}_{S_{n_1}}\cdots\mathsf{Ad}_{S_{n_{2j}}}(V_{\mathrm{od}})\right) \qquad \text{where} \qquad \sum_{r=1}^{2j} n_r = \ell - 1\ .$$

It hence suffices to show that

$$P_0\mathcal{L}\left(\mathsf{Ad}_{S_{n_1}}\cdots\mathsf{Ad}_{S_{n_{2j}}}(V_{\mathrm{od}})\right) YV_{\mathrm{od}}P_0 \in \mathbb{C}P_0\ . \tag{3.79}$$

(The proof of the second statement in (3.78) is identical and omitted here.)

By definition of $\mathcal{L}$, the claim is true if $Z_0 = P_0$, since in this case the lhs. vanishes as $P_0V_{\mathrm{od}}P_0 = 0$. Furthermore, for general $Z_0 \in \{Q_0\}\cup\{G^k \mid k\in\mathbb{N}\}$, the claim (3.79) follows if we can show that

$$P_0(\mathsf{Ad}_{S_{n_1}}\cdots\mathsf{Ad}_{S_{n_{2j}}}(V_{\mathrm{od}}))YV_{\mathrm{od}}P_0 \in \mathbb{C}P_0\ ,$$

i.e., we can omit $\mathcal{L}$ from these considerations. This follows by inserting the expression (3.61) for $\mathcal{L}$.

Observe that $\mathsf{Ad}_{S_{n_1}} \cdots \mathsf{Ad}_{S_{n_{2j}}}(V_{\rm od})$ is a linear combination of products $T_1 \cdots T_{2j+1}$ of $2j+1$ operators $\{T_r\}_{r=1}^{2j+1}$, where $(T_1, \ldots, T_{2j+1})$ is a permutation of $(S_{n_1}, \ldots, S_{n_{2j}}, V_{\rm od})$. That is, it suffices to show that for each such $2j+1$-tuple of elements $\{T_r\}_{r=1}^{2j+1}$, we have

$$P_0 T_1 \cdots T_{2j} T_{2j+1} Y V_{\rm od} P_0 \in \mathbb{C} P_0 \ . \tag{3.80}$$

By Lemma 3.A.2, $T_r \in \Gamma^*(m_r)$ for some integers $m_r \geq 1$ satisfying $\sum_{r=1}^{2j+1} m_r = 1 + \sum_{r=1}^{2r} n_r = 1 + \ell - 1$. This implies (by our assumption $\ell + m - 1 < L$) that

$$\sum_{r=1}^{2j} m_r = \ell - 1 < L \ , \tag{3.81}$$

and thus $P_0 T_1 \cdots T_{2j} \in \mathbb{C} P_0$ according to Lemma 3.A.3. We conclude that

$$\begin{aligned} P_0 T_1 \cdots T_{2j} T_{2j+1} Y V_{\rm od} P_0 &\in \mathbb{C} P_0 T_{2j+1} Y V_{\rm od} P_0 \\ &\in \mathbb{C} P_0 \Gamma(m_{2j+1} + m + 1) P_0 \ . \end{aligned}$$

But by (3.81) and the because $j \geq 1$, we have

$$\begin{aligned} m_{2j+1} + (m+1) &= (\ell - 1 - \sum_{r=1}^{2j} m_r) + (m+1) \\ &\leq (\ell - 1 - 2j) + (m+1) \leq \ell + m - 2 < L \end{aligned}$$

by assumption on ℓ, m and L, hence (3.80) follows from the assumption (3.80). □

Lemma 3.A.6. *Suppose that $P_0 \Gamma(n) P_0 \subset \mathbb{C} P_0$ for all $n < L$. Then for any ℓ, m satisfying $\ell + m - 1 < L$, we have*

$$P_0 \mathcal{V}_d^{\circ m}(B_\ell) V_{od} P_0 \in \mathbb{C} P_0 \qquad \textit{and} \qquad P_0 V_{od} \mathcal{V}_d^{\circ m}(B_\ell) P_0 \in \mathbb{C} P_0 \tag{3.82}$$

In particular, for every $q < L$, we have

$$P_0 \mathsf{Ad}_{\mathcal{V}_d^{\circ k+1}(B_{q-k})}(V_{od}) P_0 \in \mathbb{C} P_0 \tag{3.83}$$

for all $k = 0, \ldots, q-2$. Furthermore,

$$P_0 \mathsf{Ad}_{B_\ell}(V_{od}) P_0 \in \mathbb{C} P_0 \qquad \textit{for all } \ell \leq L \ . \tag{3.84}$$

Proof. By definition of $\mathcal{V}_d$, the expression $\mathcal{V}_d^{\circ m}(B_\ell)$ is a linear combination of terms of the form

$$A^L B_\ell A^R \qquad \text{where} \qquad \begin{aligned} A^L &= Z_0 V_{\rm d} Z_1 \cdots Z_{r-1} V_{\rm d} Z_r \\ A^R &= Z_{r+1} V_{\rm d} Z_{r+2} \cdots Z_m V_{\rm d} Z_{m+1} \end{aligned}$$

and each $Z_j \in \{P_0, Q_0\} \cup \{G^m \,|m \in \mathbb{N}\}$. Since A^L only involves diagonal operators and the number of factors $V_{\rm d}$ is equal to $r < L$, we have $P_0 A^L = P_0 A^L P_0 \in P_0 \Gamma(r) P_0 \in \mathbb{C} P_0$. In particular,

$$P_0 (A^L B_\ell A^R) V_{\rm od} P_0 \in \mathbb{C} P_0 B_\ell A^R V_{\rm od} P_0 \ .$$

But

$$P_0 B_\ell A^R V_{\rm od} P_0 \in \mathbb{C} P_0 \ ,$$

where we applied Lemma 3.A.5 with $Y = A^R$ (note that A^R involves $m - r$ factors $V_{\rm d}$, and $\ell + (m-r) - 1 < L$ by assumption). We conclude that

$$P_0 (A^L B_\ell A^R) V_{\rm od} P_0 \in \mathbb{C} P_0 \ ,$$

and since $P_0\mathcal{V}_d^{\circ m}(B_\ell)V_{\rm od}P_0$ is a linear combination of such terms, the first identity in (3.82) follows. The second identity is shown in an analogous manner.

The claim (3.83) follows by setting $m = k+1$ and $\ell = q-k$, and observing that $\ell+m-1 = q < L$.

Finally, consider the claim (3.84). We have

$$P_0B_\ell V_{\rm od}P_0 = P_0B_\ell Q_0V_{\rm od}P_0 \in \mathbb{C}P_0$$
$$P_0V_{\rm od}B_\ell P_0 = P_0V_{\rm od}Q_0B_\ell P_0$$

for all ℓ with $\ell - 1 < L$ by Lemma 3.A.5, hence the claim follows. □

Lemma 3.A.7. *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then*

$$P_0\mathsf{Ad}_{\mathcal{V}_d(A_q)}(V_{od})P_0 \in P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ q}(\mathcal{L}(V_{od}))}(V_{od})P_0 + \mathbb{C}P_0 \ .$$

for all $q < L$.

Proof. We will show that for $k = 1, \ldots, q-2$, we have the identity

$$P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ k}(A_{q+1-k})}(V_{\rm od})P_0 \in P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ k+1}(A_{q-k})}(V_{\rm od})P_0 + \mathbb{C}P_0 \ . \tag{3.85}$$

(Notice that the expression on the rhs. is obtained from the lhs by substituting $k+1$ for k.) Iteratively applying this implies

$$P_0\mathsf{Ad}_{\mathcal{V}_d(A_q)}(V_{\rm od})P_0 \in P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ q-1}(A_2)}(V_{\rm od})P_0 + \mathbb{C}P_0 \ ,$$

from which the claim follows since $A_2 = \mathcal{V}_d(\mathcal{L}(V_{\rm od}))$.

To prove (3.85), observe that by definition (3.77) of A_n, we have by linearity of $\mathcal{V}_d$

$$\mathcal{V}_d^{\circ k}(A_{q+1-k}) = \mathcal{V}_d^{\circ k+1}(S_{q-k}) = \mathcal{V}_d^{\circ k+1}(A_{q-k}) + \mathcal{V}_d^{\circ k+1}(B_{q-k}) \ .$$

By linearity of the map $X \mapsto P_0\mathsf{Ad}_X(V_{\rm od})P_0$, it thus suffices to show that

$$P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ k+1}(B_{q-k})}(V_{\rm od})P_0 \in \mathbb{C}P_0$$

for all $k = 1, \ldots, q-2$. This follows from (3.83) of Lemma 3.A.6. □

Lemma 3.A.8. *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then*

$$P_0\hat{S}^1(V_{od})_{n-1}P_0 = P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ n-2}(\mathcal{L}(V_{od}))}(V_{od})P_0 + \mathbb{C}P_0 \qquad \textit{for all } n < L+2 \ . \tag{3.86}$$

Proof. By definition (3.59) and the linearity of $\mathsf{Ad}_\cdot$, we have

$$\hat{S}^1(V_{\rm od})_{n-1} = \mathsf{Ad}_{S_{n-1}}(V_{\rm od}) = \mathsf{Ad}_{A_{n-1}}(V_{\rm od}) + \mathsf{Ad}_{B_{n-1}}(V_{\rm od}) \ .$$

But by definition of A_{n-1}, we have if $n-2 < L$

$$\begin{aligned} P_0\mathsf{Ad}_{A_{n-1}}(V_{\rm od})P_0 &= P_0\mathsf{Ad}_{\mathcal{V}_d(S_{n-2})}(V_{\rm od})P_0 \\ &= P_0\mathsf{Ad}_{\mathcal{V}_d(A_{n-2})}(V_{\rm od})P_0 + P_0\mathsf{Ad}_{\mathcal{V}_d(B_{n-2})}(V_{\rm od})P_0 \\ &\in P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ n-2}(\mathcal{L}(V_{\rm od}))}(V_{\rm od})P_0 + \mathbb{C}P_0 \end{aligned}$$

where we again used the linearity of the involved operations in the second step and Lemma 3.A.6 and Lemma 3.A.7 in the last step (with $k = 0$ and $q = n-2$).

Similarly, we have (again by Lemma 3.A.6) if $n-2 < L$.

$$P_0\mathsf{Ad}_{B_{n-1}}(V_{\rm od})P_0 = \mathbb{C}P_0 \ .$$

The claim (3.86) follows. □

Lemma 3.A.9. *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then*

$$P_0\hat{S}^1(V_{od})_{L-1}P_0 = 2P_0VGVG\cdots GVP_0 \ ,$$

where there are L factors V on the rhs.

Proof. We will first show inductively for $k = 1, \ldots, n-2$ that

$$\mathcal{V}_d^{\circ k}(\mathcal{L}(V_{\rm od})) = -((GV)^{k+1}P_0 - h.c.) + T_k \qquad \text{for some } T_k \in \Gamma^\star(k) \ . \tag{3.87}$$

By straightforward computation, we have

$$\begin{aligned}
\mathcal{L}(V_{\rm od}) &= P_0V_{\rm od}G - h.c.\\
[\mathcal{L}(V_{\rm od}), V_{\rm d}] &= -GV_{\rm od}P_0V_{\rm d} + V_{\rm d}GV_{\rm od}P_0 + h.c.\\
\mathcal{V}_d(\mathcal{L}(V_{\rm od})) &= -(GV_{\rm d}GV_{\rm od}P_0 - h.c.) + T_1,
\end{aligned}$$

where $T_1 = G^2V_{\rm od}P_0V_{\rm d}P_0 - h.c.$. By assumption, $P_0V_{\rm d}P_0 = P_0V_{\rm d}P_0 \in \mathbb{C}P_0$. Thus $T_1 \in \Gamma^*(1)$, and the claim (3.87) is verified for $k = 1$ (since $GV_{\rm d}GV_{\rm od}P_0 = GVGVP_0$).

Now assume that (3.87) holds for some $k \leq n-1$. We will show that it is also valid for k replaced by $k+1$. With the assumption, we have

$$\begin{aligned}
\mathcal{V}_d^{\circ k+1}(\mathcal{L}(V_{\rm od})) &= \mathcal{V}_d\left(\mathcal{V}_d^{\circ k}(\mathcal{L}(V_{\rm od}))\right)\\
&= -\mathcal{V}_d\left((GV)^{k+1}P_0 - h.c.\right) + \mathcal{V}_d(T_k)
\end{aligned}$$

But

$$\begin{aligned}
\mathcal{V}_d((GV)^{k+1}P_0) &= \mathcal{L}((GV)^{k+1}P_0V_{\rm d} - V_{\rm d}(GV)^{k+1}P_0)\\
&= -G(GV)^{k+1}P_0V_{\rm d}P_0 + GV_{\rm d}(GV)^{k+1}P_0\\
&= -G(GV)^{k+1}P_0VP_0 + (GV)^{k+2}P_0
\end{aligned}$$

and by doing a similar computation for the Hermitian conjugate we find

$$\begin{aligned}
\mathcal{V}_d^{\circ k+1}(\mathcal{L}(V_{\rm od})) &= -\left((GV)^{k+2}P_0 - h.c.\right) + T_{k+1} \qquad \text{where}\\
T_{k+1} &= \left(G(GV)^{k+1}P_0VP_0 - h.c.\right) + \mathcal{V}_d(T_k) \ .
\end{aligned}$$

We claim that $T_{k+1} \in \Gamma^\star(k+1)$. Indeed, by assumption we have $P_0VP_0 \in P_0\Gamma(1)P_0 \subset \mathbb{C}P_0$, hence $G(GV)^{k+1}P_0VP_0 \in \mathbb{C}G(GV)^{k+1}P_0 \subset \Gamma^\star(k+1)$ and the same reasoning applies to the Hermitian conjugate. Furthermore, for $T_k \in \Gamma^\star(k)$, we have $\mathcal{V}_d(T_k) \in \Gamma^\star(k+1)$ by (3.75).

This concludes the proof of (3.87), which we now apply with $k = L-2$ to get

$$\begin{aligned}
P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))}(V_{\rm od})P_0 &= P_0\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))V_{\rm od}P_0 - P_0V_{\rm od}\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))P_0\\
&= P_0(VG)^{L-1}V_{\rm od}P_0 + P_0V_{\rm od}(GV)^{L-1}P_0\\
&\quad + P_0T_{L-2}V_{\rm od}P_0 - P_0V_{\rm od}T_{L-2}P_0
\end{aligned}$$

Since $P_0T_{L-2}V_{\rm od}P_0$ and $P_0V_{\rm od}T_{L-2}P_0$ are elements of $P_0\Gamma(L-1)P_0$, we conclude that

$$\begin{aligned}
P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))}(V_{\rm od})P_0 &= P_0\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))V_{\rm od}P_0 - P_0V_{\rm od}\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))P_0\\
&= P_0((VG)^{L-1}V + V(GV)^{L-1})P_0 + \mathbb{C}P
\end{aligned}$$

Finally, with the expression obtained by Lemma 3.A.8 (with $n = L$), we get

$$\begin{aligned}
P_0\hat{S}^1(V_{\rm od})_{L-1}P_0 &= P_0\mathsf{Ad}_{\mathcal{V}_d^{\circ L-2}(\mathcal{L}(V_{\rm od}))}(V_{\rm od})P_0\\
&= 2P_0(VG)^{L-1}VP_0 + \mathbb{C}P \ ,
\end{aligned}$$

as claimed.

□

Equivalence of self-energy method and Schrieffer-Wolff transformation

With Lemma 3.A.6, Lemma 3.A.7 and Lemma 3.A.9, we now have the expressions necessary to obtain effective Hamiltonians.

Theorem 3.A.10 (Theorem 3.3.2 in the main text)**.** *Suppose that $P_0\Gamma(n)P_0 \subset \mathbb{C}P_0$ for all $n < L$. Then the n-th order Schrieffer-Wolff effective Hamiltonian satisfies*

$$H_{\mathsf{eff}}^{(n)} \in \mathbb{C}P_0 \qquad \text{for all } n < L \ , \tag{3.88}$$

i.e., the effective Hamiltonian is trivial for these orders, and

$$H_{\mathsf{eff}}^{(L)} = 2b_1 P_0 (VG)^{L-1} V P_0 + \mathbb{C}P_0 \ , \tag{3.89}$$

and where there are L factors V involved.

Proof. Consider the definition (3.60) of the n-th order term $H_{\mathsf{eff},n}$ in the expansion (3.6): we have

$$H_{\mathsf{eff},n} = \sum_{1 \le j \le \lfloor n/2 \rfloor} b_{2j-1} P_0 \hat{S}^{2j-1} (V_{\mathrm{od}})_{n-1} P_0 \ .$$

For $n < L$, each term $P_0 \hat{S}^{2j-1} (V_{\mathrm{od}})_{n-1} P_0$ is proportional to P_0 (see (3.72) of Lemma 3.A.4), hence the claim (3.88) follows.

On the other hand, for $n = L$, we have

$$P_0 \hat{S}^{2j-1} (V_{\mathrm{od}})_{L-1} P_0 \begin{cases} \in \mathbb{C}P_0 & \text{if } j > 1 \\ P_0 VGVGV \cdots GV P_0 & \text{if } j = 1 \end{cases}$$

according to Lemma 3.A.4 and Lemma 3.A.9, hence (3.89) follows. □

3.B On a class of single-qudit operators in the Levin-Wen model

In this appendix, we consider the action of certain single-qudit operators and discuss how they affect states in the Levin-Wen model. For simplicity, we will restrict our attention to models where each particle satisfies $\bar{a} = a$, i.e., is its own antiparticle. Similar local operators were previously considered (for example, in [BSS11]). We introduce the operators in Section 3.B.1 and compute the associated effective Hamiltonians in Section 3.B.2

3.B.1 Definition and algebraic properties of certain local operators

Recall that for each qudit in the Levin-Wen model, there is an orthonormal basis $\{|a\rangle\}_{a \in \mathcal{F}}$ indexed by particle labels. For each particle $a \in \mathcal{F}$, we define an operator acting diagonally in the orthonormal basis as

$$O_a |b\rangle = \frac{S_{ab}}{S_{1b}} |b\rangle \qquad \text{for all } b \in \mathcal{F} \ . \tag{3.90}$$

As an example, consider the Pauli-Z operator defined in Section 3.6.3 for the doubled semion model. Because the S-matrix of the semion model is given by (see e.g., [Sch13, Section 2.4])

$$S = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

with respect to the (ordered) basis $\{|\mathbf{1}\rangle, |\boldsymbol{s}\rangle\}$, the operator $O_{\boldsymbol{s}}$ takes the form

$$O_{\boldsymbol{s}} = \mathsf{diag}(1, -1) = Z \tag{3.91}$$

according to (3.90).

As another example, we can use the fact that the Fibonacci model has S-matrix (with respect to the basis $\{|\mathbf{1}\rangle, |\boldsymbol{\tau}\rangle\}$)

$$S = \frac{1}{\sqrt{1+\varphi^2}} \begin{bmatrix} 1 & \varphi \\ \varphi & -1 \end{bmatrix}$$

to obtain

$$O_{\boldsymbol{\tau}} = \mathsf{diag}(\varphi, -1/\varphi) \ .$$

Therefore, the Pauli-Z-operator in the doubled Fibonacci model takes the form

$$Z = \frac{\varphi}{\varphi+2} \left(-I + 2O_{\boldsymbol{\tau}}\right) \ , \tag{3.92}$$

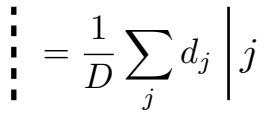
where I is the identity matrix.

We will write $O_a^{(e)} = O_a$ for the operator O_a applied to the qudit on the edge e of the lattice. To analyze the action of such an operator $O_a^{(e)}$ on ground states of the Levin-Wen model, we used the "fattened honeycomb" description of (superpositions) of string-nets: this gives a compact representation of the action of certain operators (see the appendix of [LW05]), as well as a representation of ground states (see [KKR10]). In this picture, states of the many-spin system are expressed as superpositions of string-nets (ribbon-graphs) embedded in a surface where each plaquette is punctured. Coefficients in the computational basis of the qudits can be obtained by a process of "reduction to the lattice", i.e., the application of F-moves, removal of bubbles etc. similar to the discussion in Section 3.5. Importantly, the order of reduction does not play a role in obtaining these coefficients as a result of MacLane's theorem (see the appendix of [Kit06]). Note, however, that this diagrammatic formalism only makes sense in the subspace

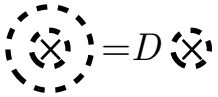
$$\mathcal{H}_{\text{valid}} = \{|\psi\rangle \mid A_v|\psi\rangle = |\psi\rangle \text{ for all vertices } v\}$$

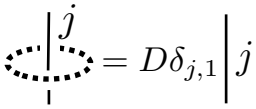
spanned by valid string-net configurations, since otherwise reduction is not well-defined.

This provides a significant simplification for certain computations. For example, application of a plaquette operator B_p corresponds – in this terminology – to the insertion of a "vacuum loop" times a factor $1/D$. The latter is itself a superposition of strings, where each string of particle type j carries a coefficient $\frac{d_j}{D}$. We will represent such vacuum strings by dotted lines below:

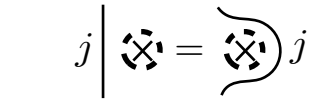


Crucial properties of this superposition are (see [KKR10, Lemma A.1])





and the pulling-through rule



Similarly, a single-qudit operator $O_a^{(e)}$ of the form (3.90) can be expressed in this language, and takes the form of adding a "ring" around a line: we have

$O_a^{(e)}|b\rangle =$ 

.

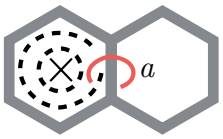
(The color is only used to emphasize the application of the operator, but is otherwise of no significance.)

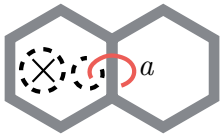
Lemma 3.B.1. *Let $a \neq 1$, and let $O_a^{(e)}$ be an operator of the form (3.90) acting an an edge e of the qudit lattice. Let p, p' be the two plaquettes adjacent to the edge e, and let $B_p, B_{p'}$ be the associated operators. Then for any $|\psi\rangle \in \mathcal{H}_{valid}$, we have*

$$B_p|\psi\rangle = |\psi\rangle \quad \Rightarrow \quad B_p(O_a^{(e)}|\psi\rangle) = 0$$
$$B_{p'}|\psi\rangle = |\psi\rangle \quad \Rightarrow \quad B_{p'}(O_a^{(e)}|\psi\rangle) = 0$$

For example, for any ground state $|\psi\rangle$ of the Levin-Wen model H_{top}, $O_a^{(e)}|\psi\rangle$ is an eigenstate of H_{top} with energy 2. Furthermore, for any ground state $|\psi\rangle$, and any edges $e_1, \ldots, e_n$ which (pairwise) do not belong to the same plaquette, the state $O_a^{(e_1)} \cdots O_a^{(e_n)}|\psi\rangle$ is an eigenstate (with energy $2n$) of H_{top}. The case where the edges belong to the same plaquette will be discussed below in Lemma 3.B.2.

Proof. For concreteness, consider the plaquette operator B_p "on the left" of the edge (the argument for the other operator is identical). Because $|\psi\rangle$ is a ground state, we have $B_p|\psi\rangle = |\psi\rangle$. Using the graphical calculus (assuming that the state $|\psi\rangle$ is expressed as a string-net embedded in the gray lattice), we obtain

$B_p O_a^{(e)} B_p|\psi\rangle = \frac{1}{D^2}$ 

$= \frac{1}{D^2}$ 

$= \frac{1}{D}\delta_{a,1}$ 

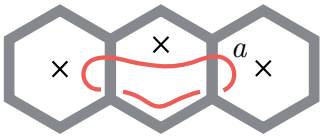
$= \delta_{a,1}|\psi\rangle$

□

Lemma 3.B.2. *Let $e_1 \neq e_2$ be two edges lying on the same plaquette p, and let us assume that they lie on opposite sides of the plaquette p (this assumption is for concreteness only and can be dropped). Let $O_a^{(e_1)}$ and $O_a^{(e_2)}$ be the associated single-qudit operators (with $a \neq 1$). Then for all $|\psi\rangle \in \mathcal{H}_{valid}$, we have*

$$B_p O_a^{(e_1)} O_a^{(e_2)} B_p|\psi\rangle = \frac{d_a}{D} B_p O_a^{(e_1 e_2)} B_p|\psi\rangle \ ,$$

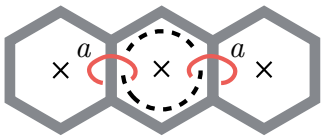
where the operator $O_a^{(e_1 e_2)}$ is defined by

$O_a^{(e_1 e_2)}|\psi\rangle =$ 

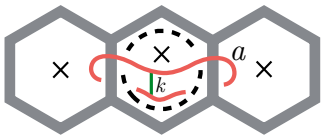
in the diagrammatic formalism. In other words, $O_a^{(e_1 e_2)}$ adds a single loop of type a around the edges e_1, e_2.

Proof. Let $|\psi\rangle \in \mathcal{H}_{\text{valid}}$. Then we have by a similar computation as before

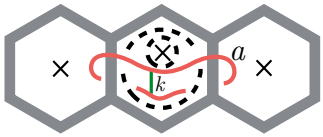
$$B_p(O_a^{(e_1)}O_a^{(e_2)})B_p|\psi\rangle = B_p\frac{1}{D}$$



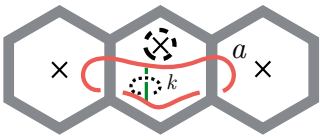
$$= B_p\frac{1}{D}\sum_k F^{aa1}_{aak}$$



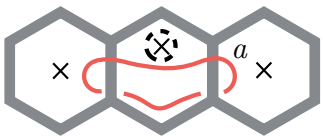
$$= \frac{1}{D^2}\sum_k F^{aa1}_{aak}$$



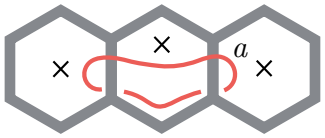
$$= \frac{1}{D^2}\sum_k F^{aa1}_{aak}$$



$$= \frac{1}{D^2}\sum_k F^{aa1}_{aak} D\delta_{k1}$$



$$= B_p\frac{1}{d_a}$$



$$= \frac{1}{d_a}B_p O_a^{(e_1 e_2)}B_p|\psi\rangle\ ,$$

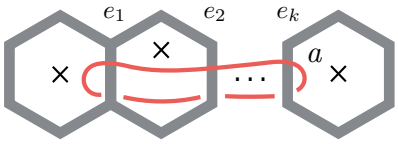
as claimed. □

Clearly, the reasoning of Lemma 3.B.2 can be applied inductively to longer sequences of products $O_a^{(e_1)}O_a^{(e_2)}\cdots O_a^{(e_k)}$ if the edges $\{e_1,\ldots,e_k\}$ correspond to a path on the dual lattice, giving rise to certain operators $O_a^{(e_1\cdots e_k)}$ with a nice graphical representation: we have for example

$$P_0 O_a^{(e_1)}O_a^{(e_2)}\cdots O_a^{(e_k)}P_0 = c\cdot P_0 O_a^{(e_1\cdots e_k)}P_0$$

for some constant c, where P_0 is the projection onto the ground space of the Levin-Wen model and where $O_a^{(e_1\cdots e_k)}$ is the operator given in the diagrammatic formalism as

$$O_a^{(e_1\cdots e_k)}|\psi\rangle =$$

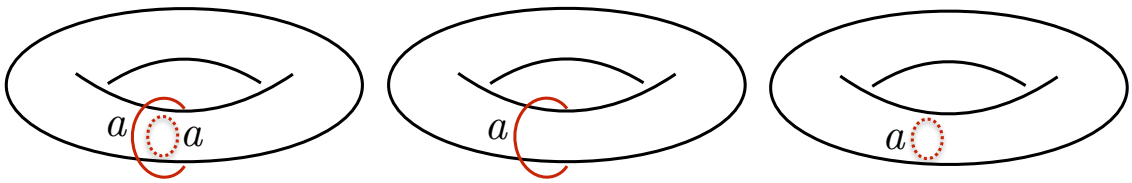


(3.93)

Using this fact, we can relate certain products of operators to the string-operator $F_{(a,a)}(C)$ associated with the (doubled) anyon (a,a). That is, assume that the edges $\{e_1,\ldots,e_L\}$ cover a topologically non-trivial loop C on the (dual) lattice (e.g., $\{10,12\}$ in the 12-qudit torus of Fig. 3.1). Then we have

$$P_0(O_a^{(e_1)}O_a^{(e_2)}\cdots O_a^{(e_L)})P_0 = c\cdot F_{a,a}(C) \tag{3.94}$$

for some constant c. This follows by comparing (3.93) with the graphical representation of the string-operators of the doubled model as discussed in [LW05], see Fig. 3.B.1. Note also that by the topological order condition, operators of the form $P_0 O_a^{(e_1)}O_a^{(e_2)}\cdots O_a^{(e_k)}P_0$ are proportional to P_0 if $k < L$.



(a) $F_{(a,a)}(C)$ (b) $F_{(1,a)}(C)$ (c) $F_{(a,1)}(C)$

Figure 3.B.1: The graphical representation of certain anyonic string-operators in the doubled model. The dashed line is inside the torus.

3.B.2 Effective Hamiltonians for translation-invariant perturbation

According to (3.91) and (3.92), a translation-invariant perturbation of the form $V = \sum_j Z_j$ for the doubled semion or Fibonacci models (as considered in Section 3.7) is, up to a global energy shift and a proportionality constant, equivalent to a perturbation of the form

$$V = \sum_e O_a^{(e)} \ , \tag{3.95}$$

where $a \neq 1$ and the sum is over all edges e of the lattice (Here $a = \boldsymbol{s}$ in the doubled semion model and $a = \boldsymbol{\tau}$ in the Fibonacci model). We show the following:

Lemma 3.B.3. *For the perturbation* (3.95) *to the Levin-Wen model H_0, the L-th order effective Hamiltonian is given by*

$$H_{\mathsf{eff}}^{(L)} = c_1 \left(\sum_C F_{(a,a)}(C) \right) + c_2 P_0 \ , \tag{3.96}$$

where c_1 and c_2 are constants, and the sum is over all topologically non-trivial loops C of length L.

Proof. According to Theorem 3.3.2, the L-th order effective Hamiltonian is proportional to

$$P_0 (VG)^{L-1} V P_0 = \sum_{e_1,\ldots,e_L} P_0 O_a^{(e_1)} G O_a^{(e_2)} G \cdots G O_a^{(e_L)} P_0$$

up to an energy shift. By the topological order constraint, the only summands on the rhs. which can have a non-trivial action on the ground space are those associated with edges $\{e_1, \ldots, e_L\}$ constituting a non-trivial loop C on the (dual) lattice. Note that for such a collection of edges, every plaquette p has at most two edges $e_j, e_k \in \{e_1, \ldots, e_L\}$ as its sides, a fact we will use below. Our claim follows if we show that for any such collection of edges, we have

$$P_0 O_a^{(e_1)} G O_a^{(e_2)} G \cdots G O_a^{(e_L)} P_0 = c F_{(a,a)}(C) \tag{3.97}$$

for some constant c.

We show (3.97) by showing that the resolvent operators G only contribute a global factor; the claim then follows from (3.94). The reason is that the local operators $O_a^{(e_\ell)}$ create localized excitations, and these cannot be removed unless operators acting on the edges of neighboring plaquettes are applied. Thus a process as the one on the lhs. (3.94) is equivalent to one which goes through a sequence of eigenstates of the unperturbed Hamiltonian H_0.

The proof of this statement is a bit more involved since operators $O_a^{(e_\ell)}$ can also create superpositions of excited and ground states. We proceed inductively. Let us set

$$\begin{array}{rclcrcl}
\Lambda_1 &=& P_0 & & \Gamma_1 &=& O_a^{(e_1)} G O_a^{(e_2)} G O_a^{(e_3)} G \cdots G O_a^{(e_L)} P_0 \\
\Lambda_2 &=& P_0 O_a^{(e_1)} G & & \Gamma_2 &=& O_a^{(e_2)} G O_a^{(e_3)} G \cdots G O_a^{(e_L)} P_0 \\
\Lambda_k &=& P_0 O_a^{(e_1)} G O_a^{(e_2)} \cdots O_a^{(e_{k-1})} G & & \Gamma_k &=& O_a^{(e_k)} G O_a^{(e_{k+1})} G \cdots G O_a^{(e_L)} P_0 \text{ for } k = 3, \ldots, L-1 \\
\Lambda_L &=& P_0 O_a^{(e_1)} G O_a^{(e_2)} \cdots O_a^{(e_{L-1})} G & & \Gamma_L &=& O_a^{(e_L)} P_0 \ .
\end{array}$$

such that

$$P_0 O_a^{(e_1)} G O_a^{(e_2)} G \cdots G O_a^{(e_L)} P_0 = \Lambda_k \Gamma_k \qquad \text{for } k = 1, \ldots, L-1 \ . \tag{3.98}$$

Let $|\psi\rangle$ be a ground state of the Levin-Wen model H_0. We claim that for every $k = 1, \ldots, L-1$, there is a set of plaquettes $\mathcal{P}_k$ and a constant c_k (independent of the chosen ground state) such that

(i) $\Lambda_k \Gamma_k |\psi\rangle = c_k \cdot \Lambda_k \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle$.

(ii) The (unnormalized) state $\left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle$ is an eigenstate of H_0. Its energy ϵ_k is independent of the state $|\psi\rangle$.

(iii) The set $\mathcal{P}_k$ only contains plaquettes which have two edges in common with $\{e_k, \ldots, e_L\}$.

Note that for $k = 1$, this implies $P_0 O_a^{(e_1)} G O_a^{(e_2)} G \cdots G O_a^{(e_L)} P_0 = c_1 \cdot P_0 O_a^{(e_1)} O_a^{(e_2)} \cdots O_a^{(e_L)} P_0$ because $P_0 B_p = P_0$, and the claim (3.97) follows with (3.94)

Properties (i), (ii) hold for $k = L$, with $\mathcal{P}_L = \emptyset$ and $\epsilon_L = 2$: we have for any ground state $|\psi\rangle$

$$\Gamma_L |\psi\rangle = O_a^{(e_L)} |\psi\rangle$$

and this is an eigenstate of H_0 with energy 2 according to Lemma 3.B.1.

Assume now that (i), (ii) hold for some $k \in \{2, L\}$. Then we have according to (3.98)

$$\begin{aligned}
\Lambda_{k-1} \Gamma_{k-1} |\psi\rangle &= \Lambda_k \Gamma_k |\psi\rangle \\
&= c_k \Lambda_k \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle \\
&= c_k (\Lambda_{k-1} O_a^{(e_{k-1})} G) \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle \\
&= c_{k-1} \cdot \Lambda_{k-1} O_a^{(e_{k-1})} \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle \ ,
\end{aligned}$$

where

$$c_{k-1} = \begin{cases} \frac{c_k}{E_0 - \epsilon_k} & \text{if } \epsilon_k > E_0 \\ 0 & \text{otherwise} \end{cases}$$

It hence suffices to show that for some choice of plaquettes $\mathcal{P}_{k-1}$, we have

(a) $\Lambda_{k-1} O_a^{(e_{k-1})} \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle = \Lambda_{k-1} \left(\prod_{p \in \mathcal{P}_{k-1}} B_p \right) O_a^{(e_{k-1})} \cdots O_a^{(e_L)} |\psi\rangle$

(b) $\left(\prod_{p \in \mathcal{P}_{k-1}} B_p \right) O_a^{(e_{k-1})} \cdots O_a^{(e_L)} |\psi\rangle$ is an eigenstate of H_0 with energy ϵ_{k-1} (independent of $|\psi\rangle$).

(c) that the set $\mathcal{P}_{k-1}$ only contains plaquettes sharing two edges with $\{e_{k-1}, \ldots, e_L\}$.

By assumption (iii) and the particular choice of $\{e_1, \ldots, e_L\}$, none of the plaquettes $p \in \mathcal{P}_k$ contains the edge e_{k-1}. Therefore, we can commute the operator $O_a^{(e_{k-1})}$ through, getting

$$O_a^{(e_{k-1})} \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle = \left(\prod_{p \in \mathcal{P}_k} B_p \right) O_a^{(e_{k-1})} O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle \tag{3.99}$$

We then consider two cases:

- If e_{k-1} does not lie on the same plaquette as any of the edges $\{e_k, \dots, e_L\}$, then application of $O_a^{(e_{k-1})}$ creates a pair of excitations according to Lemma 3.B.1 and the state (3.99) is an eigenstate of H_0 with energy $\epsilon_{k-1} = \epsilon_k + 2 > E_0$. In particular, setting $\mathcal{P}_{k-1} = \mathcal{P}_k$, properties (a)–(c) follow.

- If there is an edge e_ℓ, $\ell \geq k$ such that e_{k-1} and e_ℓ belong to the same plaquette $\tilde{p}$, then the state (3.99) is a superposition of states with $B_{\tilde{p}}$ excited/not excited, that is, we have

$$|\varphi\rangle = \left(\prod_{p\in\mathcal{P}_k} B_p\right) O_a^{(e_{k-1})} O_a^{(e_k)} \cdots O_a^{(e_L)} |\psi\rangle = (I - B_{\tilde{p}})|\varphi\rangle + B_{\tilde{p}}|\varphi\rangle \ .$$

  However, an excitation at $\tilde{p}$ cannot disappear by applying the operators $O_a^{(e_1)}, \dots, O_a^{(e_{k-2})}$ since these do not share an edge with $\tilde{p}$, hence $\Lambda_{k-1}(I - B_{\tilde{p}})|\varphi\rangle = 0$ (recall that $\Lambda_{k-1} = P_0\Lambda_{k-1}$ includes a projection onto the ground space). Thus setting $\mathcal{P}_{k-1} = \mathcal{P}_k \cup \{\tilde{p}\}$, we can verify that (a)–(c) indeed are satisfied. (The case where there are two such plaquettes $\tilde{p}$ can be treated analogously.)

□

Let us compute the effective Hamiltonian (3.96) for the case of the rhombic torus, or more specifically, the lattice we use in the numerical simulation, Fig. 3.1. It has three inequivalent weight-2 loops: $\{10, 12\}, \{1, 2\}, \{5, 7\}$. Follow the recipe in Section 3.6.2, respectively Section 3.7.2, these three loops are related by a 120° rotation. The corresponding unitary transformation for this rotation is given by the product of matrices $A = TS$ when expressed in the flux basis discussed in Section 3.6.2 (for the doubled Fibonacci model, the latter two matrices are given by (3.45)). Similarly, we can express the action of $F_{(a,a)}(C)$ in this basis using (3.36), getting a matrix F. By (3.39), the effective Hamiltonian for the perturbation $-\epsilon \sum_j Z_j$ is then proportional to (when expressed in the same basis)

$$H_{\mathsf{eff}} \sim -(F + A^{-1}FA + A^{-2}FA^2).$$

Note that the overall sign of the effective Hamiltonian is not specified in (3.34), but can be determined to be negative here by explicit calculation. For example, substituting in the S matrix (Eq. (3.45)) of the doubled Fibonacci model, we have $F = \mathsf{diag}(\varphi + 1, -1, -1, \varphi + 1)$ for the Fibonacci model. It is then straightforward to obtain the ground state of H_{eff}, which is

$$0.715|(1,1)\rangle + (0.019 - 0.057i)|(\tau,1)\rangle + (0.019 + 0.057i)|(1,\tau)\rangle + 0.693|(\tau,\tau)\rangle \ , \quad (3.100)$$

where $|(a,b)\rangle$ is a flux basis vector, i.e., the image of $P_{(a,b)}(C)$ (see Section 3.6.2) up to some phase.

Chapter 4

Optimizing dynamical decoupling with RNN

This Chapter is based on [AN16].

In this chapter, we study the possibility of using tools from machine learning to improve the performance of quantum memories. In particular, we utilize generative models which are based on recurrent neural networks to optimize dynamical decoupling (DD) sequences. DD is a less demanding technique (compared to QECC) to suppress the errors of certain noise models, and has already demonstrated in multiple experiments that it can indeed prolong the coherence time. With the numerical simulation of a spin interacting with its environment as a toy quantum memory, we show that neural networks are able to learn the structure of the good sequences from an initially random database and use it to guess potential new ones. By repeatedly updating the database, eventually the neural networks output DD sequences with performance better than that of the well-known DD families (for a specific noise model). Since our algorithm does not rely on knowing the noise model or the concrete experimental setup a priori, it is likely able to adapt to different settings and find optimized sequences specifically for them.

4.1 Introduction

As mentioned multiple times in this thesis, a major challenge of quantum information processing (e.g. quantum computation and communication) is to preserve the coherence of quantum states. While in principle we can build a fault-tolerant quantum memory or universal quantum computer once the error rate of the device is below a certain threshold, it is still beyond nowadays experimental capacity to build a decent size one. A less explored area is the optimization of implementing the fault-tolerant protocol on a concrete experimental setting. This is often a tedious problem, due to the amount of details in the real devices, and the fact that the architectures of both experimental devices and theoretical protocols are still rapidly changing. Thus, an attractive approach is to automatize this optimization task. Apart from convenience, it is conceivable that given enough time, the experimental devices together with classical computers can explore a larger parameter region compared to the often small number of possibilities considered by human.

While we are talking about the optimization of experimental devices, it might be worthwhile to look at the recent progress of machine learning. For example, in the work [MKS+13] it is shown that the computer can learn to play many relatively simple games at a human or even superhuman level. On the other hand, controlling a quantum memory is not that different from playing a game. In general, it can be abstracted as inputting a (time-dependent) series of parameters to the devices, and at the mean time possibly getting a (time-dependent) series of output from them, with a final score usually being the fidelity of the stored state. However, the recent progress of machine learning is to a large degree about the ability to processing visual information, which is not an element of the above "quantum memory" game. But rather, the increasingly versatile

machine learning computer libraries and fast computer hardware allow us to utilize a larger amount of data compared to before, as well as trying more approaches to solve the same problem. One particularly popular machine learning model at this moment is the artificial neural network (ANN), mostly because it is the state of art model to handle visual related data. Nevertheless, the key idea of ANNs is to be a family of functions/programs that are continous with respect to some parameters, which makes it applicable to the optimization problems we have in mind.

Automatically optimizing parameters in real (or numerical simulations of) experiments is not a new idea. For example, it has been applied to optimizing the pulse shape of a laser, the parameters of Hamiltonians to achieve certain unitary operations or parameters of dynamical decoupling and cold atom experiments. Most works that attempt to obtain optimal parameters use genetic algorithms [JR92, HARSS15, GCM$^+$13, PS16] (and to some degree [KMF$^+$16]) or local searches such as gradient descent [CWW13, DCQ$^+$14, PS16, BPB15, MSG$^+$11] and the Nelder-Mead simplex method [BUV$^+$09, DCM11, KBC$^+$14]. It is argued that by using these optimization methods directly on the experiments, we can avoid the hardness of modelling the imperfect control and the system-environment interaction. However, one possible weakness of these optimization methods is that they generate new trials only by looking at a fixed number of previous ones, and often they need to restart once they reach a local minimum. Thus, in the long run, they do not fully utilize all the data generated by the experiments, especially for those experiments that operates on a very short time scale.

In this chapter, we propose an orthogonal approach, where we try to mimic the structure of good parameters by building a model that approximates the probability distribution of these parameters. After an initial optimization, this model can then be used to efficiently generate new possible trials and can be continuously updated based on new data. In particular, we choose this model to be a variant of the recurrent neural network (RNN), which makes our approach very similar to the way in which natural languages or handwriting are currently modelled. This ansatz enables us to exploit the models and insights developed by the machine learning community and possibly translate further progress there into the field of quantum control. Possible advantages include the ability to utilize large amount of data with complex RNNs, as well as the easiness of training them due to mature computer libraries. It is worth pointing out that the machine learning part of this chapter is purely classical; only the (classical) data are related to quantum time evolution. Among the previous work, the approach in [WEvdH$^+$15] is the most similar to ours, as they attempt to build a model from the data and utilize the model to perform optimization. (Classical) machine learning is also used in [CFC$^+$14, OTB15, TGB15] to characterize the error models in quantum error correction and to react accordingly.

To demonstrate the feasibility of using our method to help optimizing quantum memory, we consider the problem of automatically learning and optimizing dynamical decoupling sequences (almost) without using any prior knowledge. Dynamical decoupling (DD) [VKL99] is a technique which combats certain noise by applying a sequence of unitary operations on the system (see [SAS11, QL13] for a review). It has a less stringent requirement compared to general error correction protocols, which allows it to be demonstrated in experiments [BUV$^+$09, dLWR$^+$10, SAS11] in contrast to other methods. Moreover, DD-sequences have a relatively simple structure, and therefore it is conceivable that a learning algorithm could eventually sample them without the need of using heavy mathematics. This is an important point, as machine learning is still not generally applicable to solve problems that need large amount of prior knowledge. In order to solve a problem where heavy mathematics is needed, such algorithms need either to

- develop all necessary mathematical theorems from the scratch, or
- translate the needed knowledge from mathematics books into formal languages that computers can recognize.

Both approaches are not achievable nowadays.

4.2 Background

4.2.1 Dynamical decoupling

The majority of dynamical decoupling schemes are designed for error models where the system-environment interaction can be described by a Hamiltonian. We will use $\mathcal{H}_S$ and $\mathcal{H}_B$ to denote the Hilbert space of the system and environment (often called bath), respectively. The difference between system and environment is that the former represents the part of the Hilbert space we can apply the Hamiltonian on and in which we store quantum information. The total noise Hamiltonian is

$$H_0 = H_S \otimes I_B + I_S \otimes H_B + H_{SB}.$$

Without intervention, in general H_0 would eventually destroy the quantum states we store on $\mathcal{H}_S$. To suppress this noise, we could apply a time dependent Hamiltonian $H_C(t)$ to the system, which makes the total Hamiltonian $H(t) = H_0 + H_C(t)$. In the ideal case, we can control the $H_C(t)$ perfectly and reach very high strength (i.e. norm of the Hamiltonian), which allows the ideal pulse

$$V(t) = O\delta(t - t_0).$$

It applies a unitary operator e^{-iO} to the system for an infinitely small duration (we set $\hbar = 1$ in this work). A very simple DD-scheme for a qubit (a two level system S) is the XY_4 sequence: it applies pulses of the Pauli-matrices X and Y alternatingly with equal time interval τ_d in between. A complete cycle consists of four pulses $XYXY$, thus the total time period of a cycle is $T_c = 4\tau_d$. In the limit of $\tau_d \to 0$, the qubit can be stored for an arbitrarily long time. The intuition behind DD-sequences is the average Hamiltonian theory. Let $U_C(t) = \mathcal{T}\exp\{-i\int_0^t dt' H_C(t')\}$ be the total unitary applied by $H_C(t')$ up to time t. In the interaction picture defined by $U_C(t)$, the dynamics is governed by the Hamiltonian $\tilde{H}(t) = U_C^\dagger(t) H_0 U_C(t)$. If the time interval τ_d between pulses is much smaller than the time scale defined by the norm of $\|H_0\|$, it is reasonable to consider the average of $\tilde{H}(t)$ within a cycle. The zeroth-order average Hamiltonian in T_c (with respect to τ_d) is

$$\bar{H}^{(0)} = \frac{1}{T_c}\int_0^{T_c} dt' U_C^\dagger(t) H_0 U_C(t) \ .$$

For the XY_4 sequences introduced above, it is easy to compute $\bar{H}^{(0)} = \frac{1}{4}\sum_{\sigma\in\{I,X,Y,Z\}} \sigma H_0 \sigma$. Since the mapping $O \to \sum_{\sigma\in\{I,X,Y,Z\}} \sigma O \sigma$ maps any 2×2 matrix to 0, by linearity we know $\bar{H}^{(0)} = 0$.

Here we are going to list several classes of DD-sequences. We will first explain how to concatenate two sequences, as most long DD-sequences are constructed in this manner. Given two DD-sequences $A = P_1 \cdots P_m$ and $B = Q_1 \cdots Q_n$, the concatenated sequence $A[B]$ is

$$A[B] = (P_1Q_1)Q_2 \cdots Q_n (P_2Q_1)Q_2 \cdots Q_n \cdots (P_mQ_1)Q_2 \cdots Q_n$$

As an example, when we concatenate the length-2 and length-4 sequences XX and $XYXY$, we obtain $IYXYIYXY$.

We will use P_i to represent any Pauli matrix X, Y or Z, and for $i \neq j$, $P_i \neq P_j$. The families of DD-sequences can then be listed as the following:

DD4 are length-4 sequences $P_1P_2P_1P_2$.

DD8 are length-8 sequences $IP_2P_1P_2IP_2P_1P_2$.

EDD8 are length-8 sequences $P_1P_2P_1P_2P_2P_1P_2P_1$

CDD16 are length-16 concatenated sequences $DD4[DD4]$

CDD32 are length-32 concatenated sequences $DD4[DD8]$ and $DD8[DD4]$

CDD64 are length-64 concatenated sequences $DD4[CDD16]$ and $DD8[DD8]$

Longer DD-sequences can again be obtained by the concatenation of the ones listed above, and in the ideal situation they provide better and better protection against the noise. However, with realistic experimental capability, the performance usually saturates at a certain concatenation-level. Since at this moment we are only optimizing short DD-sequences, the listed ones are sufficient to provide a baseline for our purpose. One important family we did not include here is the "Knill DD" (KDD) [RHC10], because it requires the use of non-Pauli gates.

However, we cannot expect these requirements to be met in all real world experiments. The two major imperfections that are often studied are the flip-angle errors and the finite duration of the pulses. Flip-angle errors arise from not being able to control the strength and time duration of $H_C(t)$ perfectly, thus the intended pulse $V(t) = O\delta(t)$ becomes $V(t) = (1 \pm \epsilon)O\delta(t)$. And since zero-width pulses $O\delta(t)$ are experimentally impossible, we must consider finite-width pulses which approximate the ideal ones. In this paper, we will only consider the imperfection of finite-width pulses. However, it is straightforward to apply our algorithm to pulses with flip-angle errors.

4.2.2 Measure of performance

There are multiple ways to quantify the performance of DD-sequences. In practice, we choose different measures to suit the intended applications. Here we use the same measure as in [QL13], which has the advantage of being (initial) state-independent and having a closed formula for numerical simulation:

$$D(U, I) = \sqrt{1 - \frac{1}{d_S d_B} \|\mathsf{Tr}_S(U)\|_{\mathsf{Tr}}}$$

where U represents the full evolution operator generated by $H(t)$, d_S and d_B are the dimensions of the system and environment Hilbert space $\mathcal{H}_S$ and $\mathcal{H}_B$, respectively. $\|X\|_{\mathsf{Tr}} = \mathsf{Tr}(\sqrt{X^\dagger X})$ is the trace-norm, and $\mathsf{Tr}_S(\cdot)$ is the partial trace over $\mathcal{H}_S$. The smaller $D(U, I)$ is, the better the system preserved its quantum state after the time evolution. For example, the ideal evolution $U = I_S \otimes U_B$ has the corresponding $D(U, I) = 0$.

In experiments, it is very hard to evaluate $D(U, I)$, as we often do not have access to the bath's degree of freedom. Instead, the performance of DD-sequences is often gauged by doing process tomography for the whole time duration where DD is applied [dLWR+10, SÁS12b]. Although it is a different measure compared to our choice above, the optimization procedure can still be applied as it does not rely on the concrete form of the measure. Moreover, for solid state implementations such as superconducting qubits or quantum dots, a typical run of initialization, applying DD-sequences and measurements can be done on the time scale of 1 ms or much faster. Thus, it is realistic that on the time scale of days we can gather a large dataset of DD-sequences and their performance, which is needed for our algorithm.

4.2.3 Recurrent Neural Networks

Sequential models are widely used in machine learning for problems with a natural sequential structure, e.g. speech and handwriting recognition, protein secondary structure prediction, etc. For dynamical decoupling, not only do we apply the gates sequentially in the time domain, but also the longer DD sequences are often formed by repetition or concatenation of the short ones. Moreover, once the quantum information of the system is completely mixed into the environment, it is hard to retrieve it again by DD. Thus, an educated guess is that the performance of a DD-sequence largely depends on the short subsequences of it, which can be modelled well by the sequential models.

Since our goal is not simply to approximate the distribution of good dynamical decoupling sequences by learning their structure but to sample from the learned distribution to efficiently generate new good sequences, we will further restrict ourselves to the class of generative sequential models. Overall, these models try to solve the following problem: given $\{x_i\}_{i<t}$, approximate the conditional probability $p(x_t|x_{t-1}, \ldots, x_1)$. As a simple example, we can estimate the conditional

Algorithm 1: Optimization Algorithm

Input : Number of initial models to train: n, Number of models to keep: k, Percentage of data to keep: p, Set of possible topologies: $\mathcal{M}$, Size of data: d

```
D ← generateRandomData(d) ;
D, ⟨ς_s⟩ ← keepBestData(D,p) ;
M ← trainRandomModels(n, D, 𝓜) ;
M ← keepBestKModels(M,k) ;
while ⟨ς_s⟩ not converged do
    M ← trainBestModels(D) ;
    D ← generateDataFromModels(M, d) ;
    D, ⟨ς_s⟩ ← keepBestData(D,p) ;
end
```

Output: $\langle \varsigma_s \rangle, D, M$

probability $p(x_t|x_{t-1})$ from a certain data set, and use it to generate new sequences[1]. For more sophisticated problems (e.g. natural language or handwriting), it is not enough to only consider the nearest neighbour correlations as simple models like Markov-chains of order one do.

The long short-term memory (LSTM) network, a variation of the recurrent neural network (RNN), is a state-of-the-art technique for modelling longer correlations [Gra13] and is comparably easy to train. The core idea of RNNs is that the network maintains an internal state in which it encodes information from previous time steps. This allows the model to, at least theoretically, incorporate all previous time steps into the output for a given time. Some RNNs have even been shown to be Turing-complete [Pol87]. In practice, however, RNNs often can only model relatively short sequences correctly due to an inherently unstable optimization process. This is where LSTMs improve over normal RNNs, as they allow for training of much longer sequences in a stable manner. Furthermore, LSTMs, like all ANNs, are based on matrix multiplication and the element-wise application of simple non-linear functions. This makes them especially efficient to evaluate.

From the machine learning perspective, we treat the problem at hand as a supervised learning problem where we provide the model with examples that it is to reproduce according to some error measure. We do not use the framework of reinforcement learning since the performance of a DD-sequence could only be measured after the whole sequence is applied. In fact, to some degree, our approach is to some degree similar to reinforcement learning, while the use of RNNs and provide supervised answers in the middle of the sequences allow us to handle this high dimension optimization problem effectively. Similarly, playing the board game Go can be easily formulated as a pure reinforcement learning problem, yet it is only solved efficiently with the help of human playing data and supervised learning. A short introduction to machine learning, LSTMs and their terminology can be found in the appendix and in [Nie15] while a more exhaustive discussion can be found in [GSK+15, GBC16].

4.3 Algorithm

The algorithm presented in this section is designed with the goal in mind to encode little prior knowledge about the problem into it, in order to make it generally applicable to different imperfections in the experiment. Following this idea, the method is agnostic towards the nature of the considered gates, the noise model and the measure of performance. To implement this, the algorithm assumes that

- the individual gates are represented by a unique integer number such that every sequence $s \in \mathcal{G}^{\otimes L_s}$ with $\mathcal{G}$ denoting the set of the unique identifiers and L_s being the length of s.

[1] This idea can be at least dated back to Shannon [Sha48], where this model generated "English sentences" like "ON IE ANTSOUTINYS ARE T INCTORE ST BE S DEAMY ACHIN D ILONASIVE TUCOOWE AT...."

- it is provided with a function $f(s)$ to compute the score ς_s of a given sequence s, taking into account the noise model.

The optimization problem we want to solve is

$$\min_s f(s) = \min_s \varsigma_s.$$

By assumption, we have no information about f but can efficiently evaluate it. We do furthermore assume the set of good sequences to exhibit common structural properties that can be learned well by a machine learning model. So, we propose to solve it indirectly by training a generative model $m \in \mathcal{M}$ to approximate the distribution of good sequences, $\mathcal{M}$ being the set of possible models. That means we assume $s_t \sim p_m(s_{t-1}, \ldots, s_1)$ with s_t being the gate at time t and p_m denoting the distribution learned by m. Then, we want to find an optimal m that ideally learns a meaningful representation of the structure of good sequences. As explained in the background section, we choose the type of model to be the LSTM. We now tackle this surrogate problem by alternatingly solving

$$\max_{m \in \mathcal{M}} \mathcal{L}(m|T),$$

where $\mathcal{L}$ denotes the likelihood and T the training data, and then sampling sequences from the model m to generate a new T consisting of better solutions. The algorithm hence consists of two nested optimization loops, where the inner loop fits a number of LSTMs to the current data while the outer loop uses the output of the inner loop to generate new training data. This scheme of alternatingly fixing the data to optimize the models and consecutively fixing the models to optimize the data resembles the probabilistic model building genetic algorithm [PGL02] and to some extent the expectation-maximization algorithm [DLR77]. The method is shown in Algorithm 1. Partial justification of this heuristic algorithm is given in the appendix 4.C. However, it is easy to see that the algorithm will not always find the global optimum. For example, it is conceivable that for certain problems the second to the 100-th best solutions share no common structure with the first one. In that case, it would be unlikely for the machine learning approach to find the optimal one. There is however likely no universal method to bypass this obstruction, as unless we know the best sequences already, it is impossible to verify that they exhibit some structure similar to the training sets. This obstruction seems natural since many optimization problems are believed to be computationally hard. Thus, we should not assume be able to solve them by the above routine.

We will now explain the most important aspects of the algorithm in more detail:

Choice of LSTMs The data we want to generate in our application is of sequential nature. This makes employing LSTMs an obvious choice as they pose one of the most powerful models available today for sequential data. Furthermore, the known well-performing families of DD sequences are constructed by nested concatenations of shorter sequences and hence show strong local correlations as well as global structure. LSTMs and especially models consisting of multiple layers of LSTMs are known to perform very well on such data and should therefore be able to learn and reproduce this multi-scale structure better than simpler and shallow models.

Generation of the initial training data The size d and the quality, i.e. the percentage p of the initial data to be kept, are the parameters which we can specify. The data are then generated by sampling a gate from the uniform distribution over all gates for each time-step. The average score of the initial data can then be used as a baseline to compare against in case no other reference value is available. We would like to point out that in the application considered here, an alternative way to generate the initial data might be to use the models trained on shorter sequences. This approach could lead to an initial data set with much higher average score, but at the price of introducing the bias from the previously trained RNNs.

Training of the LSTMs To reduce the chance of ending up in a bad local optimum, for each training set several different architectures of LSTMs are trained (see 4.D.2 for detailed description of LSTMs). These models are independently sampled $\mathcal{M}$. More precisely, for the first generation of models, we sample a larger set of n models from $\mathcal{M}$ and train them. We then select the best k models and reuse them for all following generations. While it might introduce some bias to the optimization, this measure drastically reduces the number of models that need to be trained in total. The training problem is defined by assuming a multinoulli distribution over the gates of each time step and minimizing the corresponding negative log-likelihood $-\sum_t \delta_{s_t,i} \log p_{m,i}(s_{t-1}, \ldots, s_1)$, where i is the index of the correct next gate, $p_{m,i}$ is its predicted probability computed by the LSTM m and $\delta_{s_t,i} = 1$ iff $s_t = i$. This error measure is also known as the *cross-entropy*. To avoid overfitting, we use a version of early stopping where we monitor the average score $\langle \varsigma_s \rangle_{p_m}$ of sequences generated by m and stop training when $\langle \varsigma_s \rangle_{p_m}$ stops improving. We employ the optimizer Adam [KB15] for robust stochastic optimization.

Selecting the best models As we employ early stop based on the average score $\langle \varsigma_s \rangle_{p_m}$, we also rank every trained model m according to this measure. One could argue that ranking the models with respect to their best scores would be a more natural choice. This however might favour models that actually produce bad sequences but have generated a few good sequences only by chance. Using $\langle \varsigma_s \rangle_{p_m}$ is hence a more robust criterion. It would of course be possible to also consider other modes of the p_m, like the variance or the skewness. These properties could be used to assess the ability of a model to generate diverse and good sequences. We find however that the models in our experiments are able to generate new and diverse sequences, thus we only use the average score as benchmark for selecting models.

Generation of the new training data The selected models are used to generate d new training data by sampling from p_m. This is done by sampling s_t from $p_i(s_{t-1}, \ldots, s_1)$ beginning with a random initialization for $t = 1$ and then using s_{t-1} as input for time step t. We combine the generated sequences with the previous training sets, remove any duplicates, and order the sequences by their scores. We then choose the best p percent for the next iteration of the optimization. This procedure ensures a monotonic improvement of the training data. Note that all selected models contribute equally many data to strengthen the diversity of the new training data. A possible extension would be to apply weighting of the models according to some properties of their learned distributions. Note though that ordering the generated sequences by their score is already a form of implicit weighting of the models.

4.4 Numerical Results

4.4.1 Noise model and the control Hamiltonian

Throughout the paper, we will use the same noise model as in [QL13]. We consider a 1-qubit system and a 4-qubit bath, namely $\mathsf{dim}(\mathcal{H}_\mathcal{S}) = 2$ and $\mathsf{dim}(\mathcal{H}_\mathcal{B}) = 16$. The small dimension of the bath is for faster numerical simulation, and there is no reason for us to think that our algorithm would only work for a small bath as the size of the bath enters the algorithm only via the score-computation function. The total noise Hamiltonian consists of (at most) 3-body interactions between the system and bath-qubits with random strength:

$$H_0 = \sum_{\mu \in \{I,X,Y,Z\}} \sigma^\mu \otimes B_\mu \ , \tag{4.1}$$

where σ^μ is summed over Pauli-matrices on the system-qubit. And B_μ is given by

$$B_\mu = \sum_{i \neq j} \sum_{\alpha,\beta} c^\mu_{\alpha\beta} \left(\sigma^\alpha_i \otimes \sigma^\beta_j \right) \ ,$$

where i, j is summed over indices of the bath qubits, and $\sigma_i^{\alpha(\beta)}$ is the Pauli-matrix on qubit i of the bath. We consider the scenario where the system-bath interaction is much stronger than the pure bath terms. More precisely, we set $c^{\mu}_{\alpha\beta} \approx 1000 c^{I}_{\alpha\beta}$ for $\mu \in \{X, Y, Z\}$. Apart from this constraint, the absolute values $|c^{\mu}_{\alpha\beta}|$ are chosen randomly from a range $[a, b]$, where we set $b \approx 3a$ to avoid too many terms vanishing in (4.1). The result Hamiltonian has a 2-norm $\|H_0\| = 20.4$.

For the control Hamiltonian, we consider the less explored scenario where the pulse shape have finite width but no switch time between them (100% duty cycle). In other words, the control Hamiltonian is piecewise constant

$$H_C(t) = H_k \ , \quad \text{for } k\tau_d \leq t < (k+1)\tau_d \ ,$$

where τ_d is a small time period with respect to the norm of H_0, and $e^{-iH_k\tau_d} \in \{I, X, Y, Z\}$. This is a good toy model for experimental settings whose DD-performance is mainly limited by the strength of the control Hamiltonian, but not the speed of shifting between Hamiltonians. Since this regime is less explored in theoretical studies, it is an interesting scenario to explore via machine learning. Another restriction we put on $H_C(t)$ is

$$H_C(t) = -H_C(T - t) \ ,$$

where T is the total evolution time. This condition ensures $U_C(T) = \mathcal{T} \exp\{-i \int_0^T dt' H_C(t')\} = I$, and it allows us to apply the same code on the setting where the system has more than one qubit. It is known that this family of symmetric Hamiltonians can remove the first order terms of τ_d in the average Hamiltonian[VKL99, SÁS12a]. So strictly speaking, this should be counted as prior knowledge. However, when we compare the known DD-sequences with the numerically found ones, we also use the symmetric version of the known DD-sequences. Thus, we perform the comparison on equal terms.

In the following, we present the results of a number of experiments we have conducted to evaluate the performance of our method. We consider sequences consisting of 32, 64 and 128 gates for varying values of τ_d. This translates to having to optimize the distribution of the first 16, 32 and 64 gates respectively. To compute ς_s, we use the figure of merit D as defined in Section 4.2.1. Thus, a lower score is better. For $\mathcal{M}$, we consider models with two or three stacked LSTM-layers followed by a final softmax layer. The layers comprise 20 to 200 units where layers closer to the input have a higher number of units. We allow for peephole connections and linear projections of the output of every LSTM-layer to a lower number dimensions [GSK+15]. The optimization parameters are also randomly sampled from sets of reasonable values. We choose the step rate to be in $\{10^{-1}, 10^{-2}\}$ and the batch size to take values in $\{200, 500, 1000\}$. The parameters specific to the Adam optimizer β_1, β_2 and ϵ, we sample from $\{0.2, 0.7, 0.9\}$, $\{0.9, 0.99, 0.999\}$ and $\{10^{-8}, 10^{-5}\}$ respectively. We perform a truncation of the gradients to 32 time steps in order to counter instabilities in the optimization (see 4.D.3). As we have stated above, we also employ early stopping in the sense that, for every optimization of a model, we keep the parameters that generate the sequences with the best average score. The algorithm was run until either the best known score was beat or the scores converged, depending on the goal of the respective experiment. We will now briefly list the concrete experiment settings and discuss the results.

Exp. E1: Length 32 In this first experiment, we considered sequences of 32 gates with $\tau_d = 0.002$. We let the algorithm train $n = 30$ models initially and set the number of models to be kept k to 5. We combined the data generated by the LSTMs with the previous training set after each generation, and chose the best 10% as the new training data, consisting of 10,000 sequences for each generation. We let every model train for 100 epochs.

Exp. E2: Length 64 In our next experiment, we tackled a more difficult scenario with 64 gates and a larger $\tau_d = 0.004$. We set $n = 50$ and $k = 5$. Again, we used the best 10% of both generated and previous data as new training data which consists in total 10,000 sequences for each training set.

Table 4.1: A comparison of the results obtained in experiments E1, E2, E3 and E4 to the best theoretically derived DD families. For each experiment, the average and best score of the last training data and the average score of the best model of the last generation are shown. They are compared to random sequences and the two DD classes that yield the best average and overall best score respectively. The best results are printed bold.

(a) Experiment E2

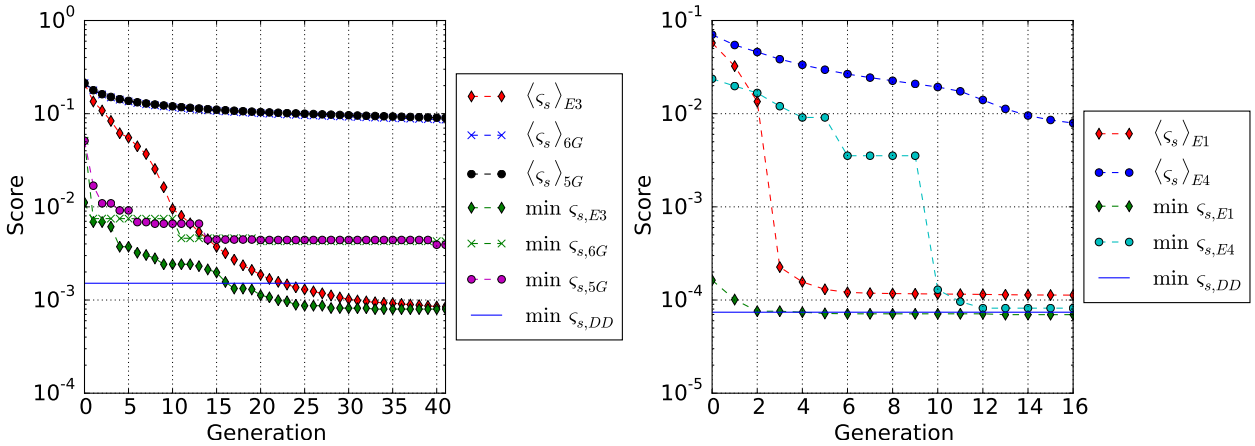
Sequences	$\langle \varsigma_s \rangle$	$\min \varsigma_s$
EDD8	0.002398	0.002112
CDD32	0.053250	0.000803
Last training set E2	**0.000712**	**0.000381**
Best model E2	0.016692	-
Random	0.341667	-

(b) Experiment E3

Sequences	$\langle \varsigma_s \rangle$	$\min \varsigma_s$
EDD8	0.004793	0.004222
CDD64	0.031547	0.001514
Last training set E3	**0.000827**	**0.000798**
Best model E3	0.029341	-
Random	0.44918	-

(c) Experiments E1 and E4

Sequences	$\langle \varsigma_s \rangle$	$\min \varsigma_s$
EDD8	0.000151	0.000133
CDD16	0.010699	0.000074
Last training set E1	**0.000112**	**0.000070**
Last training set E4	0.007178	0.000082
Best model E1	0.003089	-
Random	0.125371	-



(a) Experiment E3 (b) Experiment E1 and E4

Figure 4.1: Two figures showing the convergence of the algorithm a) in E3 compared to the case where LSTMs are replaced by 5/6-gram models and b) in E1 comparted to E4 as both consider the same problem setting. In a) it is clearly visible that LSTMs outperform the n-gram models while b) reflects the physical knowledge that the Pauli unitaries are a better choice than random gates. As a reference, we show the score of the best DD sequence obtained from the known DD classes.

Exp. E3: Length 128 In the third experiment we tried our method on even longer sequences of 128 gates with τ_d again being 0.004. Due to the very large sequence space, we set the size of the training sets to 20,000, again using the best 10% of sequences generated by the selected models and the previous training set. The number of epochs was increased to 200. We set $n = 30$ and $k = 5$. Here, we let the algorithm run until both average and best score converged to examine its behaviour in long runs.

Exp. E4: Length 32 with Random Gates Finally, we tested the performance of Algorithm 1 in the case where we replaced the Pauli gates $\{I, X, Y, Z\}$ with ten randomly chosen gates. More precisely, we chose each gate g_j to be a randomly generated single two-dimensional unitary operator with eigenvalues 1 and -1, i.e. $g_j = U_j^\dagger X U_j$, where U_j is a random unitary. All other parameters were kept as in experiment E1.

In the Tables 4.1b, 4.1a and 4.1c, we compare the last training data and the best model of the last generation of E1-E4 against the two DD families that achieve the best average and minimal scores for the given experiment respectively. We also plot the convergence of the training data of E3 and E1 with E4 in the Figures 4.1a and 4.1b respectively. In general, the results for E1, E2 and E3 clearly show that our method outperforms DD, achieving a better minimal score of the generated data in a moderate number of iterations and with a relatively small set of models. The results of E4 will be discussed below. These findings indicate that our method converges to good local optima and that the models are able to learn a meaningful internal representation of the sequences that allows for efficient sampling of good sequences. There is however a noticeable gap between the scores of the training data and the models. A possible remedy for this could be an increase of the training data size or an adjustment of the model parameters in later stages of the optimization to account for the change in the structure of the data.

To assess the importance of LSTMs for the performance of our algorithm, in experiment E3, we also ran a different version of our method where we replaced the LSTMs by simple 5/6-gram models, which only model and generate sequences based on local correlations (see Appendix 4.A.2 for the definition). The convergence plots in Figure 4.1a show that LSTMs are indeed superior to the simpler models. They are able to improve the average and best scores faster and ultimately let the algorithm converge to a better local optimum. This advantage most likely results from the fact that the LSTM-models are able to leverage information about longer range correlations in the data. These results hence justify our choice of LSTMs as machine learning model to optimize DD-sequences.

We also compared the results of experiments E1 and E4 to examine the importance of using the Pauli group as the gate set. Figure 4.1b shows that while for E1 the average score quickly becomes very good and the best score exceeds the best known result after a few generations, in E4 the average score of the data improves much slower and remains significantly worst than that of E1. Although the best score exhibits a much stronger improvement, it eventually converges to a value slightly worse than that of the best theoretical DD-sequence and the one found in E1. This is expected since with the Pauli group we can achieve first-order decoupling with DD sequences of length 4, which is the shortest. On the other hand, with random unitaries, in general it will take much longer sequences to have approximate first-order decoupling, during which the system and environment can become fairly entangled.

Another interesting aspect to note is the rather strong improvement of the average scores occurring in E3 and E1 between generations 8 to 10 and 2 to 3, respectively. These jumps can be explained by the known existence of several strictly separate regimes in sequence space that differ strongly in their performance. The results indicate that our algorithm is able to iteratively improve the learned distributions to eventually capture the regime of very good sequences.

In order to verify that sampling the initial training data from the distributions learned for shorter sequences is a viable alternative to uniform sampling, we let the best model obtained in E2 generate an initial data set for the problem setting of E3. The obtained data was found to have an average score of 0.037175, which is about one order of magnitude better than the average of the initial training data generated by uniform sampling.

4.5 Conclusion

We have introduced a novel method for optimizing dynamical decoupling sequences, which differs from previous work by the ability to utilize much larger datasets generated during the optimization. Its ability to efficiently generate large sets of good sequences could be used along with other optimization methods to cover their weaknesses or to perform statistical analysis of these sequences. We show that for certain imperfect control Hamiltonians, our method is able to outperform (almost all) known DD-sequences. The little prior knowledge about DD we use is (1) choosing Pauli operators as pulses in the sequences (see experiment E4 and its discussion), (2) choosing specific lengths for the DD-sequences and (3) enforcing the reversal symmetry, as discussed in section 4.4.1. However, we do not need to initialize the dataset in a specific way as in the Appendix C.5.a of [QL13], which actually contains a certain amount of prior knowledge of DD. Also, our method does not fundamentally rely on the prior knowledge stated above. It is conceivable that the use of this prior knowledge can be lifted, at the price of a possibly much slower optimization procedure. For example, as mentioned in [SAS11] , the KDD scheme helps to further increase the performance of CDD-sequences in some experiments. Thus, an interesting question is when given the freedom of applying non-Pauli gates and choosing variable lengths of the sequences, whether our algorithm could discover a similar strategy. Thus, a possible direction of future research is to see how we can minimize the slow-down when not incorporating any prior knowledge and whether we can obtain good DD-sequences with non-Pauli pulses.

While we have applied the algorithm to the case of quantum memory and compared it to dynamical decoupling, it is of general nature. It can in principle be applied to every problem where the optimization of a sequence of gates with respect to some well-defined figure of merit is desired and where it is feasible to evaluate this performance measure for larger numbers of sequences. However, due to the nature of the underlying machine learning model, good results will likely only be obtained for problems whose solution depends strongly on local correlations in the sequences.

Appendix

4.A Analysis

4.A.1 Local correlations of DD sequences

As we suggested earlier, the reason we use RNNs as the probabilistic model is that the performance of dynamical decoupling sequences heavily depends on their local correlations. To illustrate this fact, we can count the frequency of length-2 (3) subsequences from the training set of the 30th generation in Experiment 3. We can then compare these statistics to the ones of the sequences generated by the LSTM, which is trained based on the training set. We can see indeed the percentages match very well. To get more detail about local correlations, we could also count the frequency of length-3 subsequences (see table 4.A.2). Note that since the table is based on the datasets in the late stage of the optimization, the distribution of the subsequences are already very polarized. However, we observe the same behavior (the percentages matches well) in other experiments at different stages of the optimization as well.

However, RNNs do not only take into account local correlations, as we show in Figure 4.1 that they perform better compared to the n-gram models, which we will introduce in the next subsection.

4.A.2 n-gram models

n-grams are the simplest sequential models that treat the sequences as stationary Markov chains with order $n-1$. Operationally, given a set of sequences, we first estimate the conditional probability distribution

$$p_{x_n,x_{n-1}\cdots x_1} = \mathsf{Pr}(X_t = x_n | X_{t-1} = x_{n-1}, \ldots X_{t-n+1} = x_1).$$

Note that we assume the conditional probability is independent of t (hence stationary Markov chain). The estimation is done by counting over the whole set of sequences. The generation of new sequences based on the conditional probability $p_{x_n,x_{n-1}\cdots x_1}$ is straightforward, as we can repeatedly sample from it based on the previous $n-1$ items. This behavior is different compared to the RNNs', which have memory units that can store information for arbitrary long time in theory.

Previous \ Next gate	I	X	Y	Z
I	0.00% (0.00%)	0.04% (0.08%)	0.15% (0.68%)	0.02% (0.08%)
X	0.05% (0.22%)	5.38% (5.04%)	30.53% (30.47%)	1.39% (1.26%)
Y	0.07% (0.20%)	30.17% (30.47%)	18.40% (18.61%)	5.84% (5.50%)
Z	0.01% (0.02%)	1.90% (1.68%)	5.75% (5.42%)	0.30% (0.27%)

Table 4.A.1: The frequency of length-2 subsequences, from the training set and the set generated by the trained LSTM (given in parentheses) at the generation 30 of Experiment 3. The total number of subsequences is around 1.2 million

Second \ Last gate	I	X	Y	Z
I	0.00% (0.00%)	0.02% (0.05%)	0.12% (0.55%)	0.00% (0.01%)
X	0.00% (0.00%)	1.40% (1.22%)	11.99% (11.52%)	0.32% (0.32%)
Y	0.15% (0.47%)	44.79% (45.09%)	33.39% (33.54%)	4.11% (3.85%)
Z	0.01% (0.01%)	2.38% (2.14%)	1.05% (0.98%)	0.28% (0.26%)

Table 4.A.2: The frequency of length-3 subsequences started with the gate X, from the training set and the set generated by the trained LSTM (given in parentheses) at the generation 30 of Experiment 3. The total numbers of the subsequences started with X are around 450 thousands.

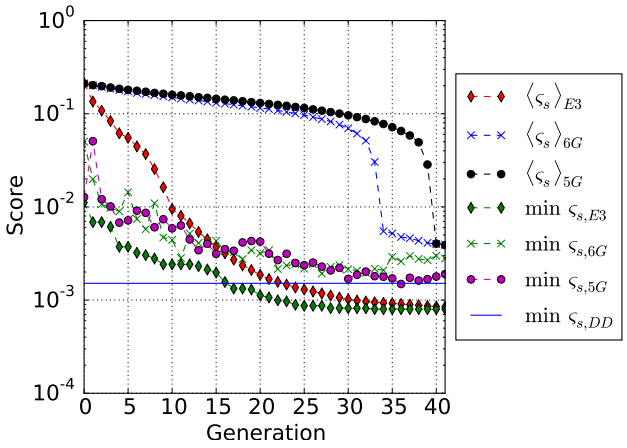


Figure 4.A.1: Experiment 3 and 5/6-gram without data reusage. Otherwise, the experiments are done in the same way as in Figure 4.1a.

4.A.3 Optimization without reusing data from previous training sets

During the optimization processes in the main text, we always reuse the data from previous training sets, in the sense that we first add the new sequences generated by the models to the training sets and then delete the worst sequences. An interesting question is what will happen if we generate new training sets completely from the trained models. In Figure 4.A.1, we plot the counterpart of Figure 4.1a with this modification (as well as not deleting duplicated sequences from the training set). We can see that for the LSTMs experiment, the final minimum score gets slightly worse, which is 0.000874. However, the 5/6-gram experiments actually performs better when not reusing data. While it seems counterintuitive, this can be possibly explained by the fact that in the case of reused data with unique sequences the higher diversity of the data might make it harder for the models to find local correlations which then in turn slows down the optimization. There is other interesting information contained in the plot. For example, we can see the minimum scores almost always decrease, which implies that the LSTMs are able to learn new information about good sequences in most generations.

4.A.4 Performance of the obtained sequences with a larger heat bath

In the main text, all the numerical simulations are done on a randomly generated noise Hamiltonian with the dimension of the bath being $\mathsf{dim}(\mathcal{H}_\mathcal{B}) = 16$. The small dimension of the bath is used in order to have a fast simulation. Here, we test the performance of some obtained sequences from the experiment 2, in the presence of a larger bath with $\mathsf{dim}(\mathcal{H}_\mathcal{B}) = 128$. Apart from the change of dimension, the Hamiltonian H_0 is again randomly generated according to the description in 4.4.1, which has a 2-norm $\|H_0\| = 24.0$. We then computed the scores of the top 500 DD sequences in the last generation of Experiment 2. The results are shown in Table 4.A.3. While the best

Sequences	$\langle\varsigma\rangle$	$\min\varsigma$
EDD8	0.002781	0.002203
CDD32	0.053753	0.000432
Top 500 sequences	0.001081	0.000626

Table 4.A.3: A comparison between the scores of the top 500 DD sequences in the last generation of Experiment 2 and some DD families for the larger bath $\mathsf{dim}(\mathcal{H}_\mathcal{B}) = 128$. The best score of the 500 sequences is worse than best score of CDD32. However, it is clear that on average, the obtained sequences still work fairly well.

score of the obtained sequences is worse than best score of CDD32, it is clear that on average, the obtained sequences still work fairly well. This also suggests that our algorithm is potentially capable of adapting to the particular noise Hamiltonian, as the learned sequences outperform known DD-families in Experiment 2.

4.B Best Sequences

We list here the best sequences we found in Experiment 1,2 and 3 from the numerical results section. We denote the identity by I, X, Y, Z refer to the respective Pauli-matrices. Note that we show only the first half of the complete sequence as the second one is just the first half reversed.
Experiment 1 X, Y, X, Z, X, Y, X, Z, Z, X, Y, X, Z, X, Y, X
Experiment 2 Z, Z, X, Z, Z, Z, X, Z, Z, X, Z, X, X, X, Z, X, X, X, Z, X, X, Z, X, X, X, Z, X, Z, Z, X, Z, Z
Experiment 3 Z, X, Z, Z, Y, X, Y, Z, Y, X, Y, X, Y, Y, X, Y, Y, Y, Y, X, Y, Y, Y, X, Y, Y, X, Y, X, Y, X, Y, Y, Z, X, Z, Y, Z, X, Z, Y, X, Y, X, X, Y, X, Y, X, Y, X, Y, Y, X, Y, Y, Y, X, Y, X, X, Y, X, X

4.C Comparison of optimization algorithms

In this section, we will give a comparison between several optimization algorithms applied to black-box problems. In other words, the algorithm needs to optimize (minimize) the objective function f only by looking at the values of $f(x)$ (without knowing the concrete formula of it). We are going to look at the following types of algorithms:

- Gradient-based algorithms (when we can access the gradient of f), e.g. Newton's method, variants of gradient descent.
- Metropolis-Hasting algorithms and its variants, e.g. simulated annealing
- Genetic algorithm and its variants, e.g. probabilistic model building genetic algorithm (PM-BGA).

The performance of an optimization algorithm depends heavily on the class of the problems it is applied to. (This fact is remotely related to the "no free lunch theorem for optimization"). Thus in the following, we will use different objective functions to illustrate the strong and weak points of those algorithms.

4.C.1 Gradient based algorithms

To understand the idea of these algorithms, it is enough to consider $f : \mathbb{R} \to \mathbb{R}$ defined on a single variable. The simplest gradient descent for finding the minimum of f is the following iterative algorithm: starting from a random number x_0 and successively computing $x_{n+1} = x_n - \alpha f'(x_n)$. Gradient based algorithms perform well on functions with non-vanishing gradients almost

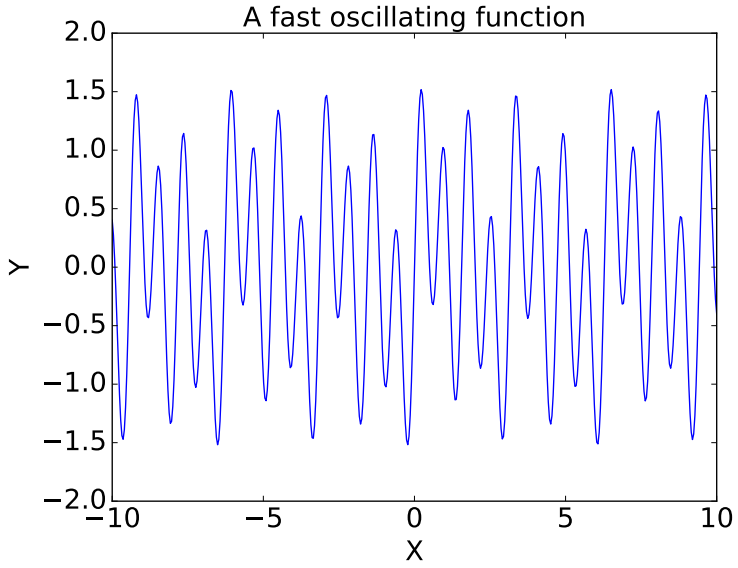


Figure 4.C.1: The plot of function (4.2).

everywhere and very few local minima, and likely have a poor performance otherwise. For example, the above algorithm would perform very well on a simple function $f(x) = x^2$, but much worse on the following fast oscillating function

$$f(x) = \sin(8x) + 0.5\sin(4x) + 0.3\sin(2x) + 0.1\sin(x) \tag{4.2}$$

We plot the above function in Figure 4.C.1. It is easy to see we can construct $f(x) = \sum_{i=1}^{N} a_i \sin(2^i x)$ such that the chance of finding the global minimum is arbitrarily small.

4.C.2 Simulated annealing

Simulated annealing (SA) and its variants stem from the Metropolis-Hastings algorithm. The main idea is constructing a family of probability distribution $p(x, T)$ based on the values of the objective function $f(x)$, with the requirement $p(x, 0) > 0$ only when x is a global minimum of f. Then we repeatedly sample from $p(x, T)$ while slowly decreasing T. In practice, simulated annealing is also an iterative algorithm, i.e. it chooses x_{n+1} based on x_n. Since SA uses the Metropolis-Hastings algorithm as a subroutine, there is a non-zero chance to choose x_{n+1} such that $f(x_{n+1}) > f(x_n)$. So in principle, SA could escape from local minima, which is an advantage compared to gradient descent. SA also works for functions with discrete variables. As a trade-off, it is likely to be slower compared to gradient descent when f has very few local minima. Moreover, while SA has the mechanism to escape from local minima, in practice it could work poorly on functions with many local minima and high barriers between them, e.g. the Function (4.2).

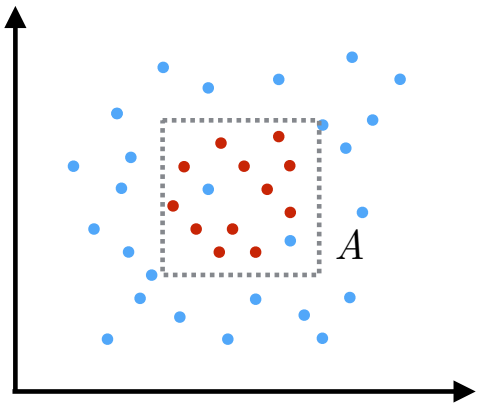
4.C.3 Genetic algorithms and beyond

In this subsection we will assume f has the form $f : \mathbb{R}^N \to \mathbb{R}$. A common feature in all versions of genetic algorithms (GA) is that they maintain a population of solutions $\{\vec{x}_i, 1 \leq i \leq M\}$, where $\vec{x}_i = (x_{i1}, \ldots, x_{iN})$. For the first generation, a number of $M' > M$ solutions is randomly generated, then we pick the $\vec{x}_i$ with the M smallest $f(\vec{x}_i)$ as the population. To generate new potential solutions for new generations, several different operations are introduced. In the original genetic algorithm, the two such operations are crossover and mutation. The effect of the mutation operation on a solution $\vec{x}$ is

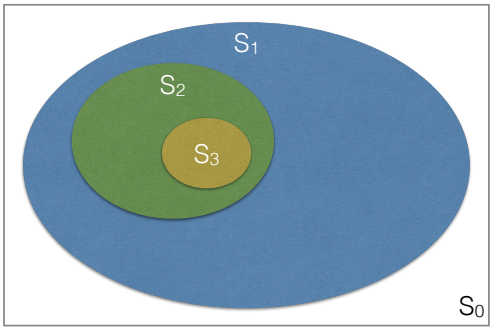
$$(x_1, \ldots, x_j, \ldots, x_N) \to (x_1, \ldots, x'_j, \ldots, x_N) \ ,$$

where x'_j is a random number. The crossover operation acts on two solutions $\vec{x}$ and $\vec{y}$

$$(\vec{x}, \vec{y}) \to (x_1, \ldots, x_j, y_{j+1}, \ldots, y_N) \ ,$$



(a) Correct hypothesis allows us to sample from a smaller region. (Red points correspond to smaller $f(x, y)$)



(b) Concatenating the operation performed in Figure (a) allows us to sample from sets S_i with better and better solutions.

Figure 4.C.2: These two figures can be viewed as an outline of our algorithm. Figure (a) demonstrates that if we can model the distribution correctly, then we will be able to sample from good solutions more efficiently. Figure (b) illustrates the idea of concatenating the step performed in Figure (a) in order to achieve an exponential speedup compared to random search.

where the position j is picked randomly. Then we can use these two operations to generate M'' new test solutions from the first generation, combine them with the M old solutions and pick the top M solutions as the population of the second generation. Later generations can be obtained by repeating these steps.

To illustrate the advantage of the (original) genetic algorithm, we can consider the following objective function f

$$f(\vec{x}) = \sum_j f_j(x_j) \ .$$

In this case, if $f(\vec{x})$ is (relatively) small, then either $\sum_{j=1}^{k} f_j(x_j)$ or $\sum_{j=k+1}^{N} f_j(x_j)$ is (relatively) small. Thus the crossover operations serve as non-local jumps, while the mutation operations help to find local minimum. However, in general, it is not clear for what kind of function f the inclusion of the crossover operations could provide an advantage. It is easy to construct counter-examples such that the crossover operations deteriorate the performance, such as

$$f(\vec{x}) = f(\vec{x}_a, \vec{x}_b) = \|\vec{x}_a - \vec{x}_b\| \ ,$$

where $\vec{x}_a, \vec{x}_b$ has equal dimension, and $\|\cdot\|$ is the Euclidean norm. Clearly, in most cases, the crossover of two good solutions will only produce inferior new solutions.

It turns out that the most important feature of genetic algorithms is the use of a population. In comparison, other optimization methods we mentioned previously only keep track of the last test solution. If we are willing to believe that good solutions of the function f have a certain structure (thus partially dropping the black-box requirement of f), it is possible that we can identify this structure from the solutions in the population, and then generate new test solutions. This idea has led to the so-called probabilistic model building genetic algorithm (PMBGA) and its variants [PGL02, Pel05]. The optimization algorithm we introduced in the main text is also closely related to this idea.

Instead of going through the details of these algorithms, we will explain the idea using a simple example, as illustrated in Fig 4.C.2. Suppose that we want to minimize a function $f(x, y)$ with two variables which defined on a finite region of $\mathbf{R}^2$, and prior knowledge of f allows us to make the hypothesis h that all points $\{(x, y)\}$ with values $f(x, y) < M$ live in a certain region A (e.g. the rectangular in Fig 4.C.2a). By sampling random points from the domain of the function, we can verify or refute the hypothesis h. For simplicity, we assume h is satisfied for all sampled points

and N of them is inside the region, then the opposite hypothesis "an α fraction of points $\{(x, y)\}$ with values $f(x, y) < M$ live outside the region A" will give the observed data a likelihood of $(1 - \alpha)^N$. Thus, we can just optimize f over the region A by ignoring a very small fraction of the good solutions. It is easy to see that we can iterate this process, as long as we can formulate a small number of hypothesis such that one of them will describe the good solutions correctly. Our algorithm in the main text resembles this toy example. However, for functions in high dimension and sophisticated generative models such as RNNs, it is hard to give a mathematical justification like in the above example.

It is natural to concatenate the above process (see Figure 4.C.2b). Let S_0 be the domain of f, and S_1 be the points in region A. By sampling enough points from S_1, we might be able to build a model and sample from a even smaller set S_2 with the good solutions (e.g. find a region $B \subset A$). This way we will introduce a series of sets $\{S_i\}_{i \leq K}$ that we can sample from. Assuming the order of these subsets satisfies $|S_{i+1}| < \frac{1}{2}|S_i|$, then in the ideal scenario the above iterative algorithm would provide an exponential speedup with respect to K. However, it is worth pointing out that automatically building a model from a data set is, in general, a difficult task (if possible at all).

As another concrete example, we can consider the objective function (4.2) and a routine which looks for the periodicity of the data and then generates new test solutions accordingly. After we go through multiple generations, it is likely that the population would converge to the correct periodic subset that has the minimum $f(x)$.

4.C.4 Summary

As seen in the discussion above, each of these optimization methods has its strong and weak points. Thus different methods are chosen depending on the prior knowledge we have on the concrete problems. It should be emphasized that we should not consider these methods as in a pure competition; instead, they can be used in complement with each other. For example, stochastic gradient Langevin dynamics (SGLD) [WT11] can be viewed as a combination of gradient descent and annealing, and in [PH06], it is mentioned that inclusion of the deterministic hill climber (discrete version of gradient descent) can lead to a substantial speedup in the PMBGA.

4.D Machine Learning

This section will give a brief overview over the subfield of machine learning known as supervised learning and introduce a model for time-series data, known as Recurrent Neural Networks (RNN). Furthermore, some aspects of the optimization of this class of models will be elaborated on.

4.D.1 Supervised Learning

The field of machine learning can be divided into three main subfields: supervised learning, unsupervised learning and reinforcement learning. These branches differ from each other by the way in which the respective models obtain information about the utility of their generated outputs.

In the case of supervised learning, it is assumed that for every input that a model shall be trained on, a "supervisor" provides a target, corresponding to the desired output of the model for the given input. These pairs of inputs and desired outputs are then used to make the model learn the general mapping between input and output.

More formally and from a Bayesian perspective, one assumes to have a dataset D of size N, consisting of several tuples of i.i.d. observations $x \in \mathbb{C}^l$ and corresponding targets $y \in \mathbb{C}^k$, such that

$$D = \{(x_i, y_i)|_{i=1}^N\}$$

where x_i and y_i are instances of two random variables X and Y respectively. These random variables are assumed to be distributed according to some unknown probability distribution p_{Gen}, the so-called data-generating distribution,

$$X, Y \sim p_{Gen}(X, Y).$$

The goal of any supervised learning method now is to approximate the conditional distribution $p_{Gen}(Y|X)$ in a way that allows for evaluation in some new observation $x_* \notin \{x_i\}|_{i=1}^N$. Since p_{Gen} is not available, one resorts to fitting the empirical distribution p_{Emp} given by D as surrogate problem.

A typical way of deriving a concrete optimization-problem from this is to make an assumption regarding the form of p_{Gen} and treating the model at hand as a distribution $p_M(Y|X,\Theta)$ of this kind, parametrized by the parameters of the model Θ that are also often called the *weights* of the model. Now, the fitting of the model can be perceived as a maximum-likelihood problem and hence the supervised learning problem can be formulated as

$$\max_\Theta \mathcal{L}(\Theta|D) = \max_\Theta \prod_i p_M(y_i|x_i,\Theta),$$

making use of the i.i.d.-assumption. A commonly employed trick to obtain a more benign optimization problem is to instead optimize the *negative log-likelihood*. As the logarithm is a monotonic function, this transformation does not change the location of the optimum in the error landscape, but turns the product of probabilities into a sum over the tuples in D. This step then yields a minimization problem, given by

$$\min_\Theta -\frac{1}{N}\sum_i \log p_M(y_i|x_i,\Theta)$$

which is called *empirical risk minimization* (ERM). These statements of the problem can now be tackled with the optimization methods appropriate for the given model. In the case of the RNN, gradient-based optimization is the state-of-the-art approach and will be explained in Section 4.D.3.

While it is obvious that fitting a model with respect to p_{Emp} is identical to fitting it to p_{Gen} as long as every tuple in D is only considered once, this is not necessarily true anymore when considering each tuple multiple times. This however is needed by many models in order to fit their parameters to a satisfying degree. In order to prevent the model from learning characteristics of the empirical distribution that are not present in the data-generating distribution, a phenomenon commonly known as *over-fitting*, often some form of regularization is applied. This may be done by punishing too large parameter values, stopping the training after performance starts to decrease on some hold-out data set or by averaging over multiple models. Note that in the Bayesian picture some penalty-terms can be perceived as the logarithm of a prior distribution over Θ, hence turning the optimization problem into finding the *maximum a-posteriori* parameters.

4.D.2 Recurrent Neural Networks

In this section, the Recurrent Neural Network model will be introduced. We will start with an introduction of the standard version of the model and based upon this, explain the advanced version of the model employed in this work in a second step.

The Standard RNN Model

In many areas of application, the data can be perceived as, often non-Markovian, discrete time-series data, such that an observation $x_t \in \mathbb{R}^l$ at some time t depends on the previous observations $x_{t-1},\dots,x_1$ or with respect to the framework introduced above,

$$X_t \sim p(X_t|X_{t-1},\dots,X_1).$$

While Markov Chains have been the state-of-the-art approach for this kind of data during the last decades, with the recent rise of artificial neural networks, RNNs [WZ89, Wer90] have also gained momentum and are now generally considered to be the most potent method.

A RNN is defined by the two non-linear maps $s_t : \mathbb{R}^l \to \mathbb{R}^h$ and $o_t : \mathbb{R}^h \to \mathbb{R}^o$ given by

$$\begin{aligned} s_t &= f_s(Ux_t + Ws_{t-1} + b_s) \\ o_t &= f_o(Vs_t + b_o), \end{aligned}$$

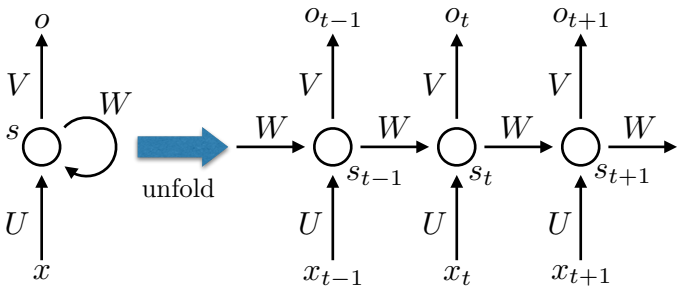


Figure 4.D.1: The standard model of a Recurrent Neural Network shown for three time-steps.

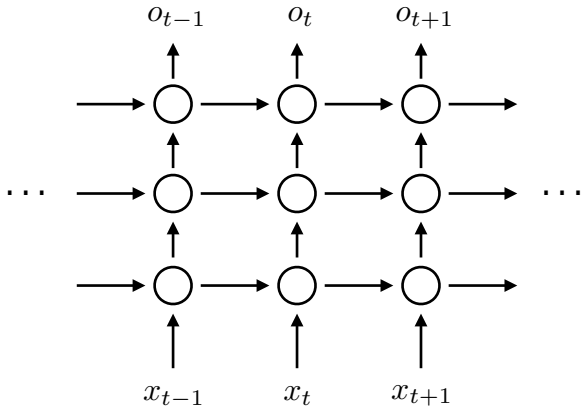


Figure 4.D.2: An illustration of an RNN with 3 hidden layers.

where $U \in \mathbb{R}^{h\times l}$, $W \in \mathbb{R}^{h\times h}$, $V \in \mathbb{R}^{o\times h}$, $b_s \in \mathbb{R}^{1\times h}$, $b_o \in \mathbb{R}^{1\times o}$ and the trainable parameters of the models are constituted by $\Theta = \{U, V, W, b_s, b_o\}$. The non-linear function f_s is often chosen to be *tanh*, the rectifier-function given by

$$rect(x) = \max(0, x)$$

or the sigmoid-function given by

$$sigm(x) = \frac{1}{1+e^{-x}}.$$

The function f_o must be chosen according to the distribution that is to be approximated by the model. For the case of a multinoulli-distribution as assumed in this work, the corresponding function would be the $softmax$, defined as

$$softmax(x)_j = \frac{e^{x_j}}{\sum_k e^{x_k}},$$

the superscripts in this case denoting the single elements of the vector x.

The intuition behind this simple model is that it combines its information about the input at a given time step with a memory of the previous inputs, referred to as *state* of the network. The precise nature of this combination and the state depends on the weight matrices U and V and the bias-vector b_s. The combined information is then used as input of the chosen non-linear function f_h to generate the next state. From this state, the output o_t is then computed as defined by W, b_o and f_o. The effect of an RNN acting on the sequence $\{x_t\}$ is illustrated in Figure 4.D.1.

From the above explanation, it is clear that the power of the model depends strongly on the size of the hidden state h. It should however also be noted that another effective way of increasing the expressive power of an RNN is to construct a composition of multiple functions of the form of s_t, see Figure 4.D.2. In the machine learning terminology, the respective functions are called the *layers* of an artificial neural network and the number of composed functions is referred to as the depth of a network. The layers between the input and the output are referred to as *hidden layers*. The common intuitive reasoning behind stacking multiple layers is that it will allow the

network to learn a hierarchy of concepts, called *features*, from the initial input data. Thereby, the features are assumed to be of increasing complexity with every layer, as they are based on a linear combination of the features learned by the layer below. Apart from this intuitive reasoning, also more rigorous work on the benefits of using at least one hidden layer between input and output can be found in the literature[BL07, HSW89, Hor91]. This ansatz of increasing the power of neural network models via deepening their architecture is publicly known as *Deep Learning* and has led to a drastic increase in success of machine learning methods during the last decade. However, having a composition of many state-computing functions of similar size can slow down the optimization process. This is why, when forming such a composition, each pair of functions is often connected via a simple linear projection from the space of the state of the earlier function onto some lower-dimensional space that is then used by the following function. Note that while all the above claims seem natural and lead to a good enough performance for our paper, more benchmarking is needed to really confirm them.

Now, in the case of supervised learning, one assumes to be in possession of a set of time-series $x_1, \ldots, x_n$ that shall be used to let the RNN learn to predict series of this kind. The natural way of doing this is to define the pairs $(x_i, y_i) \coloneqq (x_t, x_{t+1})$. While in principle the model is capable of taking into account all previous time steps, in practice it shows that optimization is only feasible for a relatively short number of steps. This is mainly due to the fact that the gradients that are needed to optimize the parameters of an RNN tend to grow to infinity or zero for higher numbers of steps. This will be discussed more in-depth below.

Long Short-Term Memory Networks

In order to improve upon the standard RNN, Hochreiter et. al. introduced the Long Short-Term Memory network (LSTM) [HS97], which provides a different way of computing the state of an RNN. Hence the following set of equations can be perceived as a replacement for s_t from the previous section. The main advantage of the approach is that it drastically mitigates the problem of unstable gradients by construction. It is defined by the following set of equations,

$$\begin{aligned} i_t &= sigm(U^i x_t + W^i s_{t-1} + b^i) \\ f_t &= sigm(U^f x_t + W^f s_{t-1} + b^f) \\ o_t &= sigm(U^o x_t + W^o s_{t-1} + b^o) \\ \tilde{c}_t &= tanh(U^{\tilde{c}} x_t + W^{\tilde{c}} s_{t-1} + b^{\tilde{c}}) \\ c_t &= c_{t-1} * f_t + \tilde{c}_t * i_t \\ s_t &= tanh(c_t) * o_t \end{aligned} \tag{4.3}$$

where again x_t is the input at time step t, s_{t-1} is the previous state of the network and c_t is the state of the cell. $U^i, U^f, U^o, U^{\tilde{c}} \in \mathbb{R}^{h \times l}$, while $W^i, W^f, W^o, W^{\tilde{c}} \in \mathbb{R}^{h \times h}$, $b^i, b^f, b^o, b^{\tilde{c}} \in \mathbb{R}^{1 \times h}$ and $*$ denotes the element-wise multiplication.

As it can be seen from the equations, the way in which an LSTM computes the state is a bit more involved. If needed, it may however just be treated as a black box and can be stacked just in the same manner as it was described for the plain RNN model. The general idea of an LSTM is to give the model a higher degree of control over the information that is propagated from one time step to the next. This is achieved by making use of so-called *gates* that control the information flow to and from the network and cell state. These gates, by taking into account the previous state and the new input, output vectors of values in $[0, 1]$ that determine how much information they let through. In the equations given above, i_t is called the *input gate*, f_t is referred to as the *forget gate* and o_t denotes the *output gate*. Now, the mechanism works as follows:

- For a given time step t, the new input and previous network state are processed by $\tilde{c}_t$ like for the standard RNN and the output values are squashed to the interval $[-1, 1]$ to yield candidate values for the next cell state.
- The input gate i_t determines how to manipulate the information flow from the candidate cell state. Likewise, the forget gate f_t determines how to affect the information flow form the old

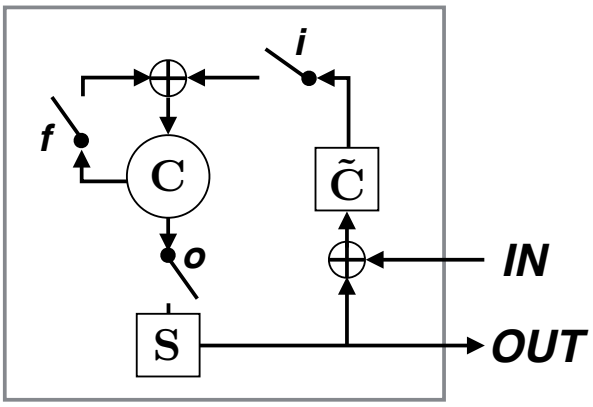


Figure 4.D.3: The Long Short-Term Memory model illustrated in a schematic way. In addition to the diagram above, the input gate **i**, the forget gate **f** and the output gate **o** all depend on the current input x_t and previous state of the network s_{t-1}, as described in (4.3).

cell state. The gated previous cell state and the gated input are then added to form the new cell state c_t.

- Finally, the output gate o_t determines what to output from the new cell state. The new cell state is then also projected onto the interval $[-1, 1]$ and put through the output gate to become the network state.

The whole process is shown in Figure 4.D.3.

Naturally, there exists a plethora of possibilities to adapt the normal LSTM as explained above. One important enhancement is commonly referred to as *peepholes*, which allows the gates to incorporate the cell state via an extra term in the sum, in addition to the input and the network state. One other popular possibility introduced in [SSB14] is the use of projection layers between different time steps of LSTM. In this case, we replace s_{t-1} by r_{t-1} in the equations for i_t, f_t, o_t and $\tilde{c}_t$ and add the simple equation

$$r_t = W^p s_t$$

where $W^p \in \mathcal{R}^{k \times h}$ is the projection matrix. In this work, we have made use of both of these extensions of the normal LSTM. For an exhaustive overview over the known variants of the LSTM, we refer the interested reader to [GSK+15].

4.D.3 Optimization of RNNs

As the optimization problem described in the beginning of this section can not be solved analytically for the models considered in this work, gradient-based approaches have established themselves as the state of the art. However, in the case of fitting the parameters of neural network models, three main restrictions need to be accounted for:

1. The number of parameters for neural network models easily exceeds 100,000 and can for larger architectures go up to several tens or even hundreds of millions. Hence, computing the Hessian (or its inverse) explicitly is not tractable and so, one is limited to first-order or approximative second-order methods.

2. As the error function that is minimized is only a surrogate error function, its global optimum is not necessarily the optimum of the error function one actually wants to minimize.

3. For many real-world data sets, computing the gradient of the complete sum of the error function over all samples is not feasible. Hence, the sum is normally split up into smaller parts called *mini batches* and these batches are looped over. A complete loop over D is then called an *epoch*.

These restrictions have led to the rise of an own subfield of machine learning that is concerned with the parallelization of gradient computations in the mini batch case, the approximation of second-order information and the formal justification for the splitting up of the error function. All of the currently available methods are nevertheless extensions of the simplest method for gradient-based optimization known as *steepest gradient descent*: At iteration i in the loop over the batches, the parameters Θ are updated according to

$$\Theta_{i+1} = \Theta_i - \gamma \frac{\partial \mathcal{E}}{\partial \Theta_i}$$

where $\mathcal{E}(D, \Theta)$ is the respective error function and γ is called the *step rate*. The most straight forward natural adaption is to make γ depend on the iteration and slowly decrease it over time, following the intuition that smaller steps are beneficial the closer one gets to the respective optimum. In addition to that, many methods employ some kind of momentum term [SMDH13] or try to approximate second order information and scale the gradient accordingly [TH12, KB15].

Besides this, the size of the batches also has an influence on the performance of the respective optimization method. In the extreme case where each batch only consists of one sample, the gradient descent method is known to converge almost surely to an optimum under certain constraints [Saa09]. As picking individual samples for optimization can be perceived as sampling from the empirical distribution to approximate the overall gradient, this method is called *stochastic gradient descent* (SGD). Using single data points however is computationally inefficient and by definition leads to heavily oscillating optimization, so it is common practice to resort to larger batches. Following the ERM-interpretation, batches B consisting of S_B samples are often used to compute an approximation of the mean gradient over D given by

$$\langle \frac{\partial \mathcal{E}}{\partial \Theta_i} \rangle_D \approx \langle \frac{\partial \mathcal{E}}{\partial \Theta_i} \rangle_B = \frac{1}{S_B} \sum_{(x,y) \in B} \frac{\partial \mathcal{E}_{(x,y)}}{\partial \Theta_i}$$

where obviously

$$\lim_{|B| \to |D|} \langle \frac{\partial \mathcal{E}}{\partial \Theta_i} \rangle_B = \langle \frac{\partial \mathcal{E}}{\partial \Theta_i} \rangle_D.$$

This interpretation is used, e. g. by the recently proposed algorithm *Adam* which has been shown to yield very good local optima while being very robust with respect to noisy gradients and needing comparatively little adjustment of its parameters. We have employed Adam for fitting the models used in this work.

While the approach to optimizing artificial neural networks is well established, this does not change the fact that the optimization problems posed by them are inherently difficult. It is well known that the error landscape becomes less smooth the more layers one adds to a network. This results in error surfaces with large planes where $\frac{\partial \mathcal{E}}{\partial \Theta} \approx 0$ that are followed by short but very steep cliffs. If the step rate is not adapted correctly, the optimization procedure is very likely to get stuck in one these planes or saddle points and to jump away from an optimum in the vicinity of Θ if evaluated on one of the cliffs. The phenomena of the frequent occurrence of very large or very small gradients are referred to in the literature as the *exploding gradient* or *vanishing gradient* problem respectively. To get a better understanding of why these problems exist, it is instructive to examine how the gradients for a given model are obtained.

As has been explained above, multi-layer neural network models are a composition of non-linear functions $\mathbb{R}^{i_k} \to \mathbb{R}^{o_k} : x_{k+1} = f_k(W_k x_k + b_k)$, where W_k is the weight-matrix, b_k the bias-vector, x_0 the input data and x_K the final output of the network. From this definition it is clear that $o_k = i_{k+1}$. For convenience, we define $y_k \equiv W_k x_k + b_k$. In order to obtain the gradient for a specific W_k or b_k one must obviously make use of the chain rule, such that

$$\frac{\partial \mathcal{E}}{\partial W_k} = \frac{\partial \mathcal{E}}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial y_k} \frac{\partial y_k}{\partial W_k} = \frac{\partial \mathcal{E}}{\partial x_K} \left(\prod_{j=k+1}^{K-1} \frac{\partial x_{j+1}}{\partial x_j} \right) \frac{\partial x_{k+1}}{\partial y_k} \frac{\partial y_k}{\partial W_k}$$

and

$$\frac{\partial \mathcal{E}}{\partial b_k} = \frac{\partial \mathcal{E}}{\partial x_{k+1}} \frac{\partial x_{k+1}}{\partial y_k} \frac{\partial y_k}{\partial b_k} = \frac{\partial \mathcal{E}}{\partial x_K} \left(\prod_{j=k+1}^{K-1} \frac{\partial x_{j+1}}{\partial x_j} \right) \frac{\partial x_{k+1}}{\partial y_k} \frac{\partial y_k}{\partial b_k}$$

where $\frac{\partial}{\partial W_k}$ is the shortcut of doing the derivative element-wise:

$$\left[\frac{\partial}{\partial W_k} \right]_{ab} = \frac{\partial}{\partial [W_k]_{ab}}$$

The same convention applies to $\frac{\partial}{\partial b_k}$. As $\frac{\partial}{\partial W_k}$ and $\frac{\partial}{\partial b_k}$ depend on all the gradients of the later layers, this formulation yields an efficient method of computing the gradients for all layers by starting with the uppermost layer and then descending in the network, always reusing the gradients already computed. Together with the fact that many of the commonly used non-linearities have an easy closed-form expression of the first derivative, this allows for fully automatic computation of the gradients as it is done in every major deep learning framework. This dynamic programming method of computing the gradients is known in the literature as *Back-Propagation*. The vanishing (exploding) gradient problem arises because of the product $\prod_{j=k+1}^{K-1} \frac{\partial x_{j+1}}{\partial x_j}$ in the above equations. For example, if one of the $\frac{\partial x_{j+1}}{\partial x_j} \approx 0$ in the product, then likely we have $\frac{\partial \mathcal{E}}{\partial W_k} \approx 0$, which leads to an ineffective gradient descent. Similarly, if many of the terms $\frac{\partial x_{j+1}}{\partial x_j}$ have large norms, then there is a possibility that $\frac{\partial \mathcal{E}}{\partial W_k}$ becomes too large, which often causes the optimization method to jump out of a local optimum.

In the case of an RNN as defined in Section 4.D.2, the above generic equations for the derivative become a little more involved, as in addition to the term for possibly multiple stacked layers, a term accounting for states of previous times has to be added. Nevertheless, at the heart of the problem, it is still about computing derivatives of composite functions. This slightly more involved back-propagation method is known as *Back-Propagation through Time* and can also be fully automatized. Similar to the multi-layer neural network models mentioned above, the gradient computation of RNNs also has these instability issues. As can be seen from Figure 4.D.1, the same matrix W is used in all time step of an RNN. Thus, a tiny change of W could affect the output o_t drastically when the time step t gets big. In other words, the derivative of the error function $\mathcal{E}$ with respect to W could again become very large or very small in certain situations. To deal with this issue, we could truncate the number of time steps during the computation, as described in Figure 4.D.4. More discussion on this topic can be found in Section 3.2 of [Gra13].

4.E Technical Aspects

For the implementation of this work, we have made use of Python with the numerical libraries NumPy, SciPy and TensorFlow [VDWCV11, JOP14, AAB+15]. All experiments were run on single workstations with up to 8 threads. The runtime of the experiments varied, depending on the optimization parameters, from a few hours to several days.

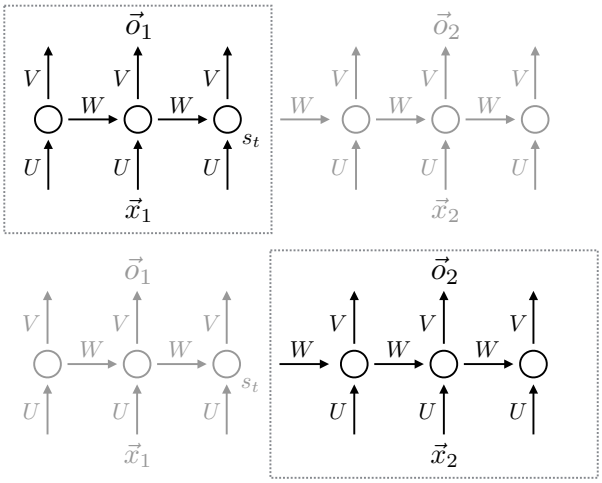


Figure 4.D.4: An illustration of how we truncate the gradient computation for long sequences. Here we divide the sequences into two halves. As the first step, we compute the gradient of the error function $\mathcal{E}(\vec{x}_1, \vec{o}_1)$ with respect to the parameters U, V, W, while ignoring the other half of the network. In the second step, we compute the gradient of $\mathcal{E}(\vec{x}_2, \vec{o}_2)$, while treating the final state of the network s_t of the first half as a constant. The final gradients are approximated by the sums of these two constituents. Thus, we are able to avoid the instability of computing gradients, but still capture the correlation between two halves, since we feed the final network state s_t into the second half.

Chapter 5

Outlook

In this last chapter, we will briefly discuss the future direction of quantum memories and how the works presented in this thesis can help the progress.

At the moment, a large amount of experimental effort is spent on solid-state implementation of the 2D surface code with active control. Apart from a relatively low fidelity threshold of the surface code, this approach is favored also because the technology of putting many qubits on a 2D chip can be very useful later for building a quantum computer. As the theory of the surface code is well-studied, the difficulty lies in how to scale up the number of qubits without decreasing the fidelity of the gates. For example, when we put many qubits on a chip, some unwanted coupling between them can arise (aka crosstalk) [BKM$^+$14b]. As the coupling is of unitary nature, dynamical decoupling can suppress it when we want to preserve quantum states or implement gates [KV09]. It is conceivable that our machine learning methods can find good DD sequences for this scenario.

It is possible that in the near future, the surface code with more than 50 qubits can be implemented that can store qubits for long duration. However, this will not immediately lead to a working quantum computer. The first problem is that certain logical gates are hard to implement for the surface code. The recently developed 2D gauge color code [BC15] has solved this problem but at the cost of introducing more qubits and more coupling between them. This brings on the second problem, which is that too many (physical) qubits are needed to have a non-trivial quantum computer. It is reasonable to assume that a logical qubit requires around 30 physical qubits (for surface code), and then we need at least 30 logical qubits to do a computation which cannot be simulated easily on a classical computer. Thus, it is still a long way until this goal is achieved, and unforeseeable obstacles can emerge during the progress. With this borne in mind, it is important that we keep researching alternative error correction codes or different approaches to quantum computing such as topological quantum computers.

The generalized stabilizer formalism proposed by us certainly enlarges the class of error correction codes that can be studied. For example, we can try to study subsystem codes within our formalism, as subsystem codes are proven useful at having more protected local gates and reducing the number of interactions needed between qubits. Another possible use of our formalism is to construct and study topologically ordered systems in three or higher dimensions. Although 2D topologically ordered systems are well-understood and all constructed examples support anyons which can be moved via string operators, less is known for higher dimensions. Intuitively, the generalization of string operators to 3D are membrane operators, and thus the excitations in 3D correspond to string and membrane operators. However, it has been shown in [Haa11] that certain 3D models cannot be understood this way, which suggests more studies need to be done. Due to the difficulty of classifying all phases, it is natural that we restrict ourselves to a subclass of all possible Hamiltonians, for which our generalized formalism is a good choice.

Another alternative approach to quantum computing is topological quantum computing. Our work on preparing ground states of topological Hamiltonians can be viewed as a way to initialize these computers. A natural question to study in this direction is generalizing our result to systems of larger size. To achieve this, we need a better understanding on how the gap at the critical point scales with system size for different models, which is also of interest for the study of phase transitions. The perturbation method we use needs to be updated to handle larger system size as

well.

In summary, the work presented in this thesis is relevant for the future research and development of quantum memories and thus quantum computing in general.

Chapter 6

Acknowledgements

First, I want to thank Ignacio Cirac for giving me the chance to work in a group of very intelligent and friendly people, and for giving me the freedom to explore topics I am interested in. While there are not many discussions, some of his opinions have changed the direction of my research substantially. I wish the works I have done are worthy of the trust.

I also want to thank Maarten van den Nest not only for bringing me into the world of research but also for being a very nice person, which I hope I could one day become.

I would like to acknowledge many people for offering me help on research during my Ph.D., including (but not limited to) Oliver Buerschaper, Fernando Pastawski, Robert König, Geza Giedke, Moritz August, and Beni Yoshida. I am glad I have the chance to learn from these people who excel in their own fields.

One of the most enjoyable thing during my Ph.D. is to make friends with many intelligent and warm-hearted colleagues. I want to thank Juan Bermejo Vega, Vanessa Paulisch, and Andras Molnar for their continuous support, which helps me through some low mood periods. Vanessa also translated the German version of the abstract in the beginning of the thesis. Alexander Müller-Hermes will be remembered for many awesome riddles and nerdy knowledge. Nayeli Rodriguez Briones was once a nice office mate and later guided me through Waterloo and Innsbruck. One of my motivations for doing Ph.D. in quantum information is to understand the mysterious nature of quantum mechanics. In that aspect, I appreciate the many interesting insights brought by Johannes Kofler, as well as his help with the writing of this thesis. Tao shi is an amazing physicist and a wonderful story teller. I am grateful for the knowledge he taught me and at the same time for many earth-shattering gossips. I also want to thank Anna Hackenbroich, Carlos Navarrete-Benlloch, Mari Carmen Bañuls, Michele Burrello, Yue Chang, , Stefan Kühn, Hyungwog Kim, Thorsten Wahl, Eliška Greplová, Johannes Knörzer and all my other coworkers and friends at the MPQ theory division. Special thanks to Veronika Lechner and Andrea Kluth for invaluable support with any administrative matters.

During these years, I am truly amazed by the beauty of Alps, and want to thank Shaowei Chou, Wu Xing, Wang Yu for teaching me climbing and snowboarding.

To my parents for their support from the distance. I also want to thank Haiyan for the company and the cheerful spirit she brings. My landlady Ursula (and her dog Dana) offered me a lot of help during my stay in Germany, for which I am very grateful. The amazing Chinese restaurant "Honghong" in Garching

Bibliography

[AAB+15] Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous systems. *Software available from tensorflow. org*, 2015.

[Aar13] Scott Aaronson. *Quantum computing since Democritus*. Cambridge University Press, 2013.

[ABD75] A.A. Abrikosov, D.E. Brown, and I.E. Dzyaloshinsky. *Quantum Field Theoretical Methods in Statistical Physics*. Dover, New York, 1975.

[ADH15] Ahmed Almheiri, Xi Dong, and Daniel Harlow. Bulk locality and quantum error correction in ads/cft. *Journal of High Energy Physics*, 4(2015):1–34, 2015.

[AGJO+15] Srinivasan Arunachalam, Vlad Gheorghiu, Tomas Jochym-OConnor, Michele Mosca, and Priyaa Varshinee Srinivasan. On the robustness of bucket brigade quantum ram. *New Journal of Physics*, 17(12):123010, 2015.

[AN16] Moritz August and Xiaotong Ni. Using recurrent neural networks to optimize dynamical decoupling for quantum memory. *arXiv preprint arXiv:1604.00279*, 2016.

[And58] P. W. Anderson. Absence of Diffusion in Certain Random Lattices. *Physical Review*, 109(5):1492–1505, mar 1958.

[And67] P. W. Anderson. Infrared Catastrophe in Fermi Gases with Local Scattering Potentials. *Physical Review Letters*, 18(24):1049, jun 1967.

[BB84] Charles H Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *International Conference on Computers, Systems & Signal Processing, Bangalore, India, Dec 9-12, 1984*, pages 175–179, 1984.

[BBK+13] CE Bardyn, MA Baranov, CV Kraus, E Rico, A İmamoğlu, P Zoller, and S Diehl. Topology by dissipation. *New Journal of Physics*, 15(8):085001, 2013.

[BBK+14] Michael E. Beverland, Oliver Buerschaper, Robert König, Fernando Pastawski, John Preskill, and Sumit Sijher. Protected gates for topological quantum field theories. arXiv:1409.3898, 2014.

[BC15] Sergey Bravyi and Andrew Cross. Doubled color codes. *arXiv preprint arXiv:1509.03239*, 2015.

[BDCP12] H Bombin, Guillaume Duclos-Cianci, and David Poulin. Universal topological phase of two-dimensional stabilizer codes. *New Journal of Physics*, 14(7):073048, 2012.

[BDL11] Sergey Bravyi, David P. DiVincenzo, and Daniel Loss. Schrieffer–wolff transformation for quantum many-body systems. *Annals of Physics*, 326(10):2793 – 2826, 2011.

[BH11] Sergey Bravyi and Matthew B Hastings. A short proof of stability of topological order under local perturbations. *Communications in mathematical physics*, 307(3):609–627, 2011.

[BH12] Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Physical Review A*, 86(5):052329, 2012.

[BHM10] Sergey Bravyi, Matthew B Hastings, and Spyridon Michalakis. Topological quantum order: stability under local perturbations. *Journal of Mathematical Physics*, 51(9):093512, 2010.

[BHV06] S. Bravyi, M. B. Hastings, and F. Verstraete. Lieb-robinson bounds and the generation of correlations and topological quantum order. *Phys. Rev. Lett.*, 97:050401, Jul 2006.

[BJQ13] Maissam Barkeshli, Chao-Ming Jian, and Xiao-Liang Qi. Classification of topological defects in abelian topological states. *Phys. Rev. B*, 88(24):241103–, 12 2013.

[BK05] Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Physical Review A*, 71(2):022316, 2005.

[BK12] Sergey Bravyi and Robert König. Disorder-assisted error correction in majorana chains. *Communications in Mathematical Physics*, 316(3):641–692, 2012.

[BK13] Sergey Bravyi and Robert König. Classification of topologically protected gates for local stabilizer codes. *Phys. Rev. Lett.*, 110:170503, Apr 2013.

[BKM+14a] R Barends, J Kelly, A Megrant, A Veitia, D Sank, E Jeffrey, T C White, J Mutus, A G Fowler, B Campbell, Y Chen, Z Chen, B Chiaro, A Dunsworth, C Neill, P O'Malley, P Roushan, A Vainsencher, J Wenner, A N Korotkov, A N Cleland, and John M Martinis. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–3, apr 2014.

[BKM+14b] R Barends, J Kelly, A Megrant, A Veitia, D Sank, E Jeffrey, TC White, J Mutus, AG Fowler, B Campbell, et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014.

[BL07] Yoshua Bengio and Yann LeCun. Scaling learning algorithms towards ai. *Large-scale kernel machines*, 34(5), 2007.

[Blo58] Claude Bloch. Sur la théorie des perturbations des états liés. *Nuclear Physics*, 6:329–347, 1958.

[BMD09] H. Bombin and M. A. Martin-Delgado. Quantum measurements and gates by code deformation. *Journal of Physics A: Mathematical and Theoretical*, 42(9):095302, 2009.

[Bom15] H. Bombin. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, 2015.

[Bon09] Parsa Bonderson. Splitting the Topological Degeneracy of Non-Abelian Anyons. *Physical Review Letters*, 103(11):110403, 2009.

[BPB15] Leonardo Banchi, Nicola Pancotti, and Sougato Bose. Quantum gate learning in engineered qubit networks: Toffoli gate with always-on interactions. page 5, sep 2015.

[BPT10] Sergey Bravyi, David Poulin, and Barbara Terhal. Tradeoffs for reliable quantum information storage in 2d systems. *Physical review letters*, 104(5):050503, 2010.

[BS09] FA Bais and JK Slingerland. Condensate-induced transitions between topologically ordered phases. *Physical Review B*, 79(4):045316, 2009.

[BSS11] FJ Burnell, Steven H Simon, and JK Slingerland. Condensation of achiral simple currents in topological lattice models: Hamiltonian study of topological symmetry breaking. *Physical Review B*, 84(12):125434, 2011.

[BSW11] Salman Beigi, Peter W. Shor, and Daniel Whalen. The quantum double model with boundary: Condensations and symmetries. *Commun. Math. Phys.*, 306(3):663–694, 2011.

[Bue14] Oliver Buerschaper. Twisted injectivity in projected entangled pair states and the classification of quantum phases. *Annals of Physics*, 351:447–476, 2014.

[BUV+09] Michael J Biercuk, Hermann Uys, Aaron P VanDevender, Nobuyasu Shiga, Wayne M Itano, and John J Bollinger. Optimized dynamical decoupling in a model quantum memory. *Nature*, 458(7241):996–1000, apr 2009.

[BV05] S. Bravyi and M. Vyalyi. Commutative version of the k-local Hamiltonian problem and common eigenspace problem. *Quantum Inf. and Comp.*, 5(3):187–215, December 2005.

[CFC+14] Joshua Combes, Christopher Ferrie, Chris Cesare, Markus Tiersch, G. J. Milburn, Hans J. Briegel, and Carlton M. Caves. In-situ characterization of quantum devices with error correction. may 2014.

[CFS07] Tommaso Caneva, Rosario Fazio, and Giuseppe E. Santoro. Adiabatic quantum dynamics of a random Ising chain across its quantum critical point. *Physical Review B*, 76(14):144427, oct 2007.

[CGM+14] Jerry M Chow, Jay M Gambetta, Easwar Magesan, David W Abraham, Andrew W Cross, B R Johnson, Nicholas A Masluk, Colm A Ryan, John A Smolin, Srikanth J Srinivasan, and M Steffen. Implementing a strand of a scalable fault-tolerant quantum computing fabric. *Nature communications*, 5:4015, jan 2014.

[CGW10] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Local unitary transformation, long-range quantum entanglement, wave function renormalization, and topological order. *Physical review b*, 82(15):155138, 2010.

[CMS+15] A.D. Córcoles, Easwar Magesan, Srikanth J. Srinivasan, Andrew W. Cross, M. Steffen, Jay M. Gambetta, and Jerry M. Chow. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature Communications*, 6:6979, apr 2015.

[CWW13] Chunlin Chen, Lin-Cheng Wang, and Yuanlong Wang. Closed-loop and robust control of quantum systems. *The Scientific World Journal*, 2013:869285, 2013.

[DCM11] Patrick Doria, Tommaso Calarco, and Simone Montangero. Optimal control technique for many-body quantum dynamics. *Physical review letters*, 106(19):190501, may 2011.

[DCQ+14] Daoyi Dong, Chunlin Chen, Bo Qi, Ian R Petersen, and Franco Nori. Robust manipulation of superconducting qubits in the presence of fluctuations. *Scientific Reports*, 5:7873, aug 2014.

[DDM03] Jeroen Dehaene and Bart De Moor. The clifford group, stabilizer states, and linear and quadratic operations over GF(2). *Phys. Rev. A*, 68:042318, April 2003.

[DiV95] David P DiVincenzo. Quantum computation. *Science*, 270(5234):255, 1995.

[DKLP02] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002.

[DKP14] John Dengis, Robert König, and Fernando Pastawski. An optimal dissipative encoder for the toric code. *New Journal of Physics*, 16(1):013023, 2014.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[dLWR+10] G. de Lange, Z H Wang, D Ristè, V V Dobrovitski, and R Hanson. Universal dynamical decoupling of a single solid-state spin from a spin bath. *Science (New York, N.Y.)*, 330(6000):60–63, 2010.

[dWP97] Mark de Wild Propitius. (spontaneously broken) abelian chern-simons theories. *Nuclear Physics B*, 489(1):297–359, 1997.

[EdMFD11] BM Escher, RL de Matos Filho, and L Davidovich. General framework for estimating the ultimate precision limit in noisy quantum-enhanced metrology. *Nature Physics*, 7(5):406–411, 2011.

[FCY+04] David Fattal, Toby S Cubitt, Yoshihisa Yamamoto, Sergey Bravyi, and Isaac L Chuang. Entanglement in the stabilizer formalism. *arXiv preprint quant-ph/0406168*, 2004.

[FKLW03] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. Topological quantum computation. *Bulletin of the American Mathematical Society*, 40(1):31–38, 2003.

[FTL+07] Adrian Feiguin, Simon Trebst, Andreas WW Ludwig, Matthias Troyer, Alexei Kitaev, Zhenghan Wang, and Michael H Freedman. Interacting anyons in topological quantum liquids: The golden chain. *Physical Review Letters*, 98(16):160409, 2007.

[FW03] Alexander L Fetter and John Dirk Walecka. *Quantum theory of many-particle systems*. Courier Corporation, 2003.

[GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016.

[GCM+13] I. Geisel, K. Cordes, J. Mahnke, S. Jollenbeck, J. Ostermann, J. Arlt, W. Ertmer, and C. Klempt. Evolutionary optimization of an experimental apparatus. *Applied Physics Letters*, 102(21):214105, may 2013.

[GHSZ90] Daniel M Greenberger, Michael A Horne, Abner Shimony, and Anton Zeilinger. Bells theorem without inequalities. *Am. J. Phys*, 58(12):1131–1143, 1990.

[GLM06] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum metrology. *Physical review letters*, 96(1):010401, 2006.

[GMC15] Yimin Ge, András Molnár, and J Ignacio Cirac. Rapid adiabatic preparation of injective peps and gibbs states, 2015. arXiv:1508.00570.

[Got97] Daniel Gottesman. Stabilizer Codes and Quantum Error Correction. May 1997.

[Gra13] Alex Graves. Generating Sequences With Recurrent Neural Networks. *arXiv preprint arXiv:1308.0850*, aug 2013.

[GSK+15] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

[Haa11] Jeongwan Haah. Local stabilizer codes in three dimensions without string logical operators. *Physical Review A*, 83(4):042330, 2011.

[HARSS15] U. Las Heras, U. Alvarez-Rodriguez, E. Solano, and M. Sanz. Genetic Algorithms for Digital Quantum Simulations. dec 2015.

[HCDB07] Lorenz Hartmann, J Calsamiglia, W Dür, and HJ Briegel. Weighted graph states and applications to spin chains, lattices and gases. *Journal of Physics B: Atomic, Molecular and Optical Physics*, 40(9):S1, 2007.

[HDE+05] Marc Hein, Wolfgang Dür, Jens Eisert, Robert Raussendorf, M Nest, and H-J Briegel. Entanglement in graph states and its applications. *Proceedings of the International School of Physics "Enrico Fermi" on "Quantum Computers, Algorithms and Chaos"*, 2005.

[HEB04] Marc Hein, Jens Eisert, and Hans J Briegel. Multiparty entanglement in graph states. *Physical Review A*, 69(6):062311, 2004.

[HL08] Alioscia Hamma and Daniel A Lidar. Adiabatic preparation of topological order. *Physical Review Letters*, 100(3):030502, 2008.

[Hor91] Kurt Hornik. Approximation capabilities of multilayer feedforward networks. *Neural networks*, 4(2):251–257, 1991.

[HS97] Sepp Hochreiter and Juergen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[HSW89] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[HWW13] Yuting Hu, Yidun Wan, and Yong-Shi Wu. Twisted quantum double model of topological phases in two dimensions. *Physical Review B*, 87(12):125114, 2013.

[HZHL08] A Hamma, W Zhang, S Haas, and DA Lidar. Entanglement, fidelity, and topological entropy in a quantum phase transition to topological order. *Physical Review B*, 77(15):155111, 2008.

[JOP14] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[JR92] RS Judson and H Rabitz. Teaching lasers to control molecules. *Physical review letters*, 68(10):1500–1503, mar 1992.

[KB10] Robert König and Ersen Bilgin. Anyonic entanglement renormalization. *Physical Review B*, 82(12):125118, 2010.

[KB15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representation*, 2015.

[KBC+14] J. Kelly, R. Barends, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, I. C. Hoi, E. Jeffrey, A. Megrant, J. Mutus, C. Neill, P. J J O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, A. N. Cleland, and John M. Martinis. Optimal quantum control using randomized benchmarking. *Physical Review Letters*, 112(24):1–5, 2014.

[KBF+15] J Kelly, R Barends, AG Fowler, A Megrant, E Jeffrey, TC White, D Sank, JY Mutus, B Campbell, Yu Chen, et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015.

[Kit01] A Yu Kitaev. Unpaired Majorana fermions in quantum wires. *Physics-Uspekhi*, 44(10S):131, 2001.

[Kit03] A Yu Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.

[Kit06] Alexei Kitaev. Anyons in an exactly solved model and beyond. *Annals of Physics*, 321(1):2–111, 2006.

[KK09] Caroline Kruszynska and Barbara Kraus. Local entanglability and multipartite entanglement. *Physical Review A*, 79(5):052304, 2009.

[KK12] Alexei Kitaev and Liang Kong. Models for gapped boundaries and domain walls. *Commun. Math. Phys.*, 313(2):351–373, 2012.

[KKR10] Robert König, Greg Kuperberg, and Ben W Reichardt. Quantum computation with Turaev–Viro codes. *Annals of Physics*, 325(12):2707–2749, 2010.

[KL97] Emanuel Knill and Raymond Laflamme. Theory of quantum error-correcting codes. *Physical Review A*, 55(2):900–911, feb 1997.

[KMF+16] Mario Krenn, Mehul Malik, Robert Fickler, Radek Lapkiewicz, and Anton Zeilinger. Automated Search for new Quantum Experiments. *Physical Review Letters*, 116(9):090405, mar 2016.

[KP06] Alexei Kitaev and John Preskill. Topological entanglement entropy. *Physical review letters*, 96(11):110404, 2006.

[KP14] Robert König and Fernando Pastawski. Generating topological order: no speedup by dissipation. *Physical Review B*, 90(4):045101, 2014.

[KV09] Kaveh Khodjasteh and Lorenza Viola. Dynamically error-corrected gates for universal quantum computation. *Physical review letters*, 102(8):080501, 2009.

[LMGH15] Justyna Lodyga, Pawel Mazurek, Andrzej Grudka, and Michal Horodecki. Simple scheme for encoding and decoding a qubit in unknown state for various topological codes. *Scientific reports*, 5:8975, jan 2015.

[LMR13] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.

[LMRW13] Noah Linden, František Matúš, Mary Beth Ruskai, and Andreas Winter. The quantum entropy cone of stabiliser states. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography*, page 270, 2013.

[LRH09] Daniel A Lidar, Ali T Rezakhani, and Alioscia Hamma. Adiabatic approximation with exponential accuracy for many-body systems and quantum computation. *Journal of Mathematical Physics*, 50(10):102106, 2009.

[LW03] Daniel A Lidar and K Birgitta Whaley. Decoherence-free subspaces and subsystems. In *Irreversible Quantum Dynamics*, pages 83–120. Springer, 2003.

[LW05] Michael A Levin and Xiao-Gang Wen. String-net condensation: A physical mechanism for topological phases. *Physical Review B*, 71(4):045110, 2005.

[MKS+13] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[MSG+11] S. Machnes, U. Sander, SJ Glaser, P de Fouquieres, A Gruslys, S Schirmer, and T Schulte-Herbrüggen. Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework. *Physical Review A*, 84(2):022305, 2011.

[MZ13] Spyridon Michalakis and Justyna P. Zwolak. Stability of Frustration-Free Hamiltonians. *Communications in Mathematical Physics*, 322(2):277–302, jul 2013.

[MZF+12] V Mourik, K Zuo, S M Frolov, S R Plissard, E P A M Bakkers, and L P Kouwenhoven. Signatures of Majorana fermions in hybrid superconductor-semiconductor nanowire devices. *Science (New York, N.Y.)*, 336(6084):1003–7, may 2012.

[NBVdN15] Xiaotong Ni, Oliver Buerschaper, and Maarten Van den Nest. A non-commuting stabilizer formalism. *Journal of Mathematical Physics*, 56(5):052201, 2015.

[NC00] Michael A. Nielsen and Isaac L. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.

[Nie15] Michael Nielsen. *Neural network and deep learning*. Determination Press, 2015.

[NPDL+14] Stevan Nadj-Perge, Ilya K Drozdov, Jian Li, Hua Chen, Sangjun Jeon, Jungpil Seo, Allan H MacDonald, B Andrei Bernevig, and Ali Yazdani. Topological matter. Observation of Majorana fermions in ferromagnetic atomic chains on a superconductor. *Science (New York, N.Y.)*, 346(6209):602–7, oct 2014.

[NPYK16] Xiaotong Ni, Fernando Pastawski, Beni Yoshida, and Robert Koenig. Preparing topologically ordered states by hamiltonian interpolation. *New Journal of Physics*, 18(093027), 2016.

[NSS+08] Chetan Nayak, Steven H Simon, Ady Stern, Michael Freedman, and Sankar Das Sarma. Non-abelian anyons and topological quantum computation. *Reviews of Modern Physics*, 80(3):1083, 2008.

[OTB15] Davide Orsucci, Markus Tiersch, and Hans J. Briegel. Estimation of coherent error sources from stabilizer measurements. dec 2015.

[PCB+10] R. N. C. Pfeifer, P. Corboz, O. Buerschaper, M. Aguado, M. Troyer, and G. Vidal. Simulation of anyons with tensor network algorithms. *Physical Review B*, 82(11):115126, sep 2010.

[Pel05] Martin Pelikan. Bayesian optimization algorithm. In *Hierarchical Bayesian optimization algorithm*, pages 31–48. Springer, 2005.

[Pfe70] Pierre Pfeuty. The one-dimensional Ising model with a transverse field. *Annals of Physics*, 57(1):79–90, mar 1970.

[PGL02] Martin Pelikan, David E Goldberg, and Fernando G Lobo. A survey of optimization by building and using probabilistic models. *Computational optimization and applications*, 21(1):5–20, 2002.

[PH06] Martin Pelikan and Alexander K Hartmann. Searching for ground states of ising spin glasses with hierarchical boa and cluster exact approximation. In *Scalable Optimization via Probabilistic Modeling*, pages 333–349. Springer, 2006.

[PKSC10] Fernando Pastawski, Alastair Kay, Norbert Schuch, and Ignacio Cirac. Limitations of Passive Protection of Quantum Information. *Quantum Information and Computation*, 10(7&8):0580–0618, nov 2010.

[Pol87] Jordan B Pollack. *On connectionist models of natural language processing*. Computing Research Laboratory, New Mexico State University, 1987.

[PS16] Lukasz Pawela and Przemysaw Sadowski. Various methods of optimizing control pulses for quantum systems with decoherence. *Quantum Information Processing*, jan 2016.

[PYHP15] Fernando Pastawski, Beni Yoshida, Daniel Harlow, and John Preskill. Holographic quantum error-correcting codes: toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6):1, 2015.

[QL13] Gregory Quiroz and Daniel Lidar. Optimized dynamical decoupling via genetic algorithms. *Physical Review A - Atomic, Molecular, and Optical Physics*, 88(5), 2013.

[RBB03] Robert Raussendorf, Daniel E Browne, and Hans J Briegel. Measurement-based quantum computation on cluster states. *Physical review A*, 68(2):022312, 2003.

[RHBM13] Matteo Rossi, M Huber, D Bruß, and C Macchiavello. Quantum hypergraph states. *New Journal of Physics*, 15(11):113022, 2013.

[RHC10] C. A. Ryan, J. S. Hodges, and D. G. Cory. Robust decoupling techniques to extend quantum coherence in diamond. *Physical Review Letters*, 105(20):1–4, 2010.

[RO10] Gustavo Rigolin and Gerardo Ortiz. Adiabatic Perturbation Theory and Geometric Phases for Degenerate Systems. *Physical Review Letters*, 104(17):170406, apr 2010.

[RO12] Gustavo Rigolin and Gerardo Ortiz. Adiabatic theorem for quantum systems with spectral degeneracy. *Physical Review A*, 85(6):062111, jun 2012.

[RO14] Gustavo Rigolin and Gerardo Ortiz. Degenerate adiabatic perturbation theory: Foundations and applications. *Physical Review A*, 90(2):022104, aug 2014.

[Saa09] David Saad. *On-line learning in neural networks*, volume 17. Cambridge University Press, 2009.

[Sac11] Subir Sachdev. *Quantum Phase Transitions*. Cambridge University Press, 2011.

[SAS11] Alexandre M Souza, Gonzalo A Alvarez, and Dieter Suter. Robust dynamical decoupling for quantum computing and quantum memory. *Physical review letters*, 106(24):240501, jun 2011.

[SÁS12a] Alexandre M. Souza, Gonzalo A. Álvarez, and Dieter Suter. Effects of time-reversal symmetry in dynamical decoupling. *Physical Review A*, 85(3):032306, mar 2012.

[SÁS12b] Alexandre M Souza, Gonzalo A Álvarez, and Dieter Suter. Robust dynamical decoupling. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 370(1976):4748–69, oct 2012.

[Sch78] Thomas J Schaefer. The complexity of satisfiability problems. In *Proceedings of the tenth annual ACM symposium on Theory of computing*, pages 216–226. ACM, 1978.

[Sch13] Marc Daniel Schulz. *Topological phase transitions driven by non-Abelian anyons*. PhD thesis, Technischen Universität Dortmund, 2013.

[Sha48] CE Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[Sho99] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[SMDH13] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1139–1147, 2013.

[SSB14] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[STV+13] Martin Schwarz, Kristan Temme, Frank Verstraete, David Perez-Garcia, and Toby S Cubitt. Preparing topological projected entangled pair states on a quantum computer. *Physical Review A*, 88(3):032321, 2013.

[SW66] J. R. Schrieffer and P. A. Wolff. Relation between the anderson and kondo hamiltonians. *Phys. Rev.*, 149:491–492, Sep 1966.

[Teu03] Stefan Teufel. *Adiabatic perturbation theory in quantum dynamics*. Springer-Verlag Berlin Heidelberg, 2003.

[TGB15] M. Tiersch, E J Ganahl, and H J Briegel. Adaptive quantum computation in changing environments using projective simulation. *Scientific reports*, 5:12874, 2015.

[TH12] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4:2, 2012.

[Tur36] Alan Mathison Turing. On computable numbers, with an application to the entscheidungsproblem. *J. of Math*, 58(345-363):5, 1936.

[VDN10] Maarten Van Den Nes. Classical simulation of quantum computation, the gottesman-knill theorem, and slightly beyond. *Quantum Information & Computation*, 10(3):258–271, 2010.

[VdN11] Maarten Van den Nest. A monomial matrix formalism to describe quantum many-body states. *New Journal of Physics*, 13(12):123004, 2011.

[VDWCV11] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[VKL99] Lorenza Viola, Emanuel Knill, and Seth Lloyd. Dynamical Decoupling of Open Quantum Systems. *Physical Review Letters*, 82(12):2417–2421, mar 1999.

[VYPW11] Vivek Venkatachalam, Amir Yacoby, Loren Pfeiffer, and Ken West. Local charge of the $\nu = 5/2$ fractional quantum Hall state. *Nature*, 469(7329):185–8, jan 2011.

[Wan10] Zhengang Wang. *Topological quantum computation*, volume Regional Conference Series in Mathematics/Conference Board of the mathematical Sciences, no. 112. American Mathematical Society, 2010.

[Wer90] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[WEvdH+15] P. B. Wigley, P. J. Everitt, A. van den Hengel, J. W. Bastian, M. A. Sooriyabandara, G. D. McDonald, K. S. Hardman, C. D. Quinlivan, P. Manju, C. C. N. Kuhn, I. R. Petersen, A. Luiten, J. J. Hope, N. P. Robins, and M. R. Hush. Fast machine-learning online optimization of ultra-cold-atom experiments. page 6, jul 2015.

[WT11] Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 681–688, 2011.

[WVHC08] Michael M Wolf, Frank Verstraete, Matthew B Hastings, and J Ignacio Cirac. Area laws in quantum systems: mutual information and correlations. *Physical review letters*, 100(7):070502, 2008.

[WZ89] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

[Yos11] Beni Yoshida. Classification of quantum phases and topology of logical operators in an exactly solved model of quantum codes. *Annals of Physics*, 326(1):15–95, 2011.

[Yos15a] Beni Yoshida. Gapped boundaries, group cohomology and fault-tolerant logical gates. arXiv:1509.03626, 2015.

[Yos15b] Beni Yoshida. Topological color code and symmetry-protected topological phases. *Phys. Rev. B*, 91(24):245131–, 06 2015.