

SCHOOL OF NATURAL SCIENCES —  
PHYSICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Physics

**Efficient Path-Integral Computation of  
Correlation Functions for Perturbed  
Ternary Unitary Circuits**

Lion Frangoulis

SCHOOL OF NATURAL SCIENCES —  
PHYSICS

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis in Physics

**Efficient Path-Integral Computation of  
Correlation Functions for Perturbed  
Ternary Unitary Circuits**

**Effiziente Pfadintegral Berechnung von  
Korrelationsfunktionen in gestörten ternär  
unitären Schaltkreisen**

Author: Lion Frangoulis  
Supervisor: Prof. Dr. rer. nat. Frank Pollmann  
Advisor: Prof. Dr. Christian Mendl, Richard Milbradt  
Submission Date: 14.07.2023

# Abstract

Time evolution of two dimensional quantum lattices are subject to the "curse of dimensionality" and therefore numerical simulations are unable to reach large number of time steps for even intermediate sized lattices. In this thesis a number of new methods already applied on 1+1 dimensional networks of (almost) dual unitary gates [18] are combined with a new class of rank 8 tensors, so called ternary unitary gates [25], to grant new insights in 2+1 dimensional systems. While these gates allow for exact computation of spatiotemporal correlation functions, non trivial results are limited to a very small region of the network and the class of such gates is very limited. The combination of these methods allows for the approximation of non trivial correlation functions within the entire light pyramid and a larger number of gates, while allowing the computation at arbitrary high times and lattice sizes. While the original 1+1D work still had exponentially scaling cost, a new method of evaluation of this approximation is shown here, that has polynomial runtime and memory requirements and therefore allows for much larger number of time steps. Additionally, the conditions necessary for the approximation error to be bound were further refined to be verifiable, even in this higher dimensional setting. This new method is verified under a toy model using random longitudinal fields, and generically generated gates. Additionally, the question of an optimal contraction order for the original tensor network is investigated, in order to verify the numeric results with the exact solutions.

# Zusammenfassung

Die exakte Zeitentwicklung zweidimensionaler Quantensysteme unterliegt dem sogenannten "Curse of dimensionality", weshalb numerische Simulationen dieser auf weniger Zeitschritte beschränkt sind. In dieser Arbeit werden eine Reihe von bekannten Methoden, welche bereits erfolgreich in 1+1 dimensional System von gestörten dual unitary Gates angewandt wurden [18], kombiniert mit einer neuen Klasse von Rang 8 Tensoren, so genannten ternary unitary gates [25]. Während die exakten Korrelationsfunktionen für diese Art von Gattern bereits effizient gelöst wurde, sind nicht triviale Resultate auf sehr spezifische Gitterpositionen beschränkt, und die Generalität dieser Klasse ist ebenfalls stark beschränkt. Die Verallgemeinerung der Störungstheorie von 1+1 auf 2+1 dimensionale Systeme, erlaubt eine effiziente Näherung der Korrelationsfunktion für gestörte ternary unitaries, welche in nicht trivialen Korrelationsfunktionen innerhalb der gesamten Lichtpyramide resultiert, und eine größere Menge an Gattern erlaubt. Die originale Methode, welche weiterhin exponentiell in der Zeit wachsende Kosten aufwies, wurde weiter verbessert hin zu einer kubischen Skalierung des Rechenaufwandes. Zusätzlich wurde die Bedingung, unter welcher die Approximation hält, auf 2+1 Dimensionen erweitert und abgewandelt, um den Rechenaufwand zu reduzieren. Die resultierende Methode wurde numerisch verifiziert, zunächst unter Zuhilfenahme eines durch zufällige Magnetfelder reduzierten Systems, und letztendlich für unmodifizierte ternary unitaries. Die optimale Kontraktionsreihenfolge des exakten Tensornetzes wird ebenfalls besprochen und implementiert, um den Näherungsfehler numerisch zu überprüfen.

# Contents

<b>Abstract</b>	<b>ii</b>
<b>Zusammenfassung</b>	<b>iii</b>
<b>1. Introduction</b>	<b>1</b>
<b>2. Problem Setting</b>	<b>2</b>
2.1. Lattice structure and correlation functions . . . . .	2
2.2. Folded picture . . . . .	3
2.3. Ternary unitary gates . . . . .	7
2.4. Gate generation . . . . .	8
<b>3. Strategy and Conditions</b>	<b>11</b>
3.1. Path sum representation . . . . .	11
3.2. Skeleton Paths simplification . . . . .	12
3.2.1. Exact cases . . . . .	14
3.3. Conditions for bound approximation error . . . . .	14
3.3.1. Condition 1: planar propagators . . . . .	17
3.3.2. Condition 2: Linear propagators . . . . .	21
3.3.3. Simplified case with path-picture . . . . .	23
<b>4. Calculation</b>	<b>25</b>
4.1. Reduced Gates . . . . .	25
4.1.1. Analytic order expansion approach . . . . .	25
4.1.2. Numeric layer approach . . . . .	27
4.2. Generic Gates . . . . .	30
4.3. Exact solutions for benchmarking . . . . .	32
4.4. Weakness and Mitigation . . . . .	33
<b>5. Results</b>	<b>36</b>
5.1. Reduced Gates . . . . .	36
5.1.1. Correlation for small times . . . . .	36
5.1.2. Behaviour for larger times . . . . .	38

*Contents*

---

5.1.3. Perturbation strength . . . . .	41
5.1.4. Order Expansion . . . . .	43
5.1.5. Improved Results . . . . .	45
5.2. Generic Gates . . . . .	47
5.2.1. Stacked gates . . . . .	47
5.2.2. Crossed gates . . . . .	49
<b>6. Application outlook: Verification on a quantum computer</b>	<b>51</b>
<b>7. Conclusion and outlook</b>	<b>53</b>
<b>A. Numerical gate parameters</b>	<b>56</b>
<b>B. Additional Order Analysis</b>	<b>59</b>
<b>Bibliography</b>	<b>62</b>

# 1. Introduction

The simulation of quantum many-body systems with local interactions is a vital process of understanding the dynamics of systems and Interaction models [2], where physical implementations are hard to realize or can only be partly analyzed [32]. Especially in situations, where finite size effects can change the results drastically [8], efficient simulation methods for large system sizes are highly sought after. Especially the combination of real-time evolution and systems of two or more dimensions typically leads to exponentially high computational cost [11, 16]. However, in specific systems, the calculation of two-point correlation function can be done efficiently, even though the full time evolution would be exponentially expensive [5, 6].

The simulation of two-point, time-evolved correlation functions of local observables can already give vital information about various macroscopical quantities, especially when used in the context of linear response theory [1, 21]. While there are certain systems with analytic solutions [13], these are far between and oftentimes of limited generality. Recently, a new class of operators, called "dual unitary" [28], has been defined. These gates, behaving unitarily in both time and spatial direction, allow the construction of brickwork-type quantum circuits, that result in exactly solvable two-point correlation functions [4]. This specific class of gates has sparked several new developments [9, 15, 36], and have been further generalized to different types of networks [10, 14, 24], with one of the most recent development being the perturbation around this dual-unitary point [18], leading to efficient evaluation methods of correlation functions in 1+1 dimensions while allowing for a broader class of gates to be applied.

With the generalization of this concept to ternary unitary gates [25], a class of gates showing unitary behaviour in time and both spatial dimensions, the exact calculation of correlation functions for infinite states in 2+1 dimensions has been accomplished. From this, further research is needed, to generalize the results achieved for dual unitary circuits into higher dimensions. In this thesis, an efficient calculation algorithm for two-point correlation functions for gates, that are perturbed around the ternary unitary point, is shown and numerically verified.

## 2. Problem Setting

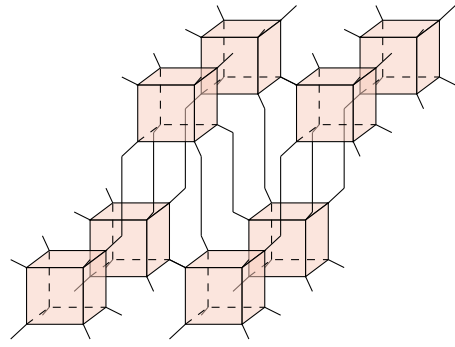
### 2.1. Lattice structure and correlation functions

The goal of this thesis is to generalize the path sum formula for two-point correlation functions from 1+1D [18] to 2+1D. In the original work, the correlation function was shown as a sum over all possible paths, and further refined to only incorporate so-called "skeleton" paths, which radically reduced computational complexity.

For this, we consider a two-dimensional system of qudits with side length  $L$ , arranged in a tetragonal lattice. Each site is therefore representing a local quantum system with Hilbert space  $\mathcal{H} \simeq \mathbb{C}^d$ , described by a basis of orthonormal state  $\mathcal{B} = \{|j\rangle, j = 0, \dots, d-1\}$ . The time evolution of an initial state is governed by an operator  $\mathcal{U}$ , which applies one discrete time step on the system, and consists of the local rank 8 tensor  $U_{x,y}$  applied in a shifted manner:

$$\mathcal{U} = \bigotimes_{x,y \text{ odd}} U_{x,y} \bigotimes_{x,y \text{ even}} U_{x,y}, \quad (2.1)$$

with  $U_{x,y}$  acting on sites  $(x,y)$ ,  $(x+1,y)$ ,  $(x,y+1)$  and  $(x+1,y+1)$ . Additionally, by assuming translation invariance, namely  $U_{x,y} = U$  with the local gate  $U, u_{ij,kl}^{m,n,o,p} \in \mathbb{C}$ , this can be described diagrammatically as



(2.2)



---

## 2. Problem Setting

---

with the gates  $U$  and  $U^\dagger$  being defined as:

$$\begin{array}{c}
 \begin{array}{cc}
 \begin{array}{c}
 \begin{array}{c}
 \text{2} \quad \text{3} \\
 \diagdown \quad \diagup \\
 \text{0} \quad \text{1} \\
 \text{---} \quad \text{---} \\
 \diagup \quad \diagdown \\
 \text{4} \quad \text{5} \\
 \text{---} \quad \text{---} \\
 \text{6} \quad \text{7}
 \end{array} \\
 U = \begin{array}{c} \text{---} \end{array} \\
 \begin{array}{c}
 \text{2} \quad \text{3} \\
 \diagdown \quad \diagup \\
 \text{0} \quad \text{1} \\
 \text{---} \quad \text{---} \\
 \diagup \quad \diagdown \\
 \text{4} \quad \text{5} \\
 \text{---} \quad \text{---} \\
 \text{6} \quad \text{7}
 \end{array} \\
 U^\dagger = \begin{array}{c} \text{---} \end{array}
 \end{array}
 \end{array}
 \tag{2.3}$$

Additionally, these gates are assumed to fulfill the unitarity condition:

$$\begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array}
 \end{array}
 \end{array}
 \tag{2.4}$$

Additionally, we assume spatial periodic boundary conditions, and we consider the system to be in the infinite-temperature limit, resulting in tracing over the top and bottom legs, essentially assuming periodic boundary conditions in time, denoted by:

$$\begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \dots \\
 \begin{array}{c}
 \begin{array}{c}
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array} \\
 \begin{array}{c}
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---} \\
 \diagdown \quad \diagup \\
 \text{---} \\
 \diagup \quad \diagdown \\
 \text{---}
 \end{array}
 \end{array}
 \end{array}
 \tag{2.5}$$

### 2.2. Folded picture

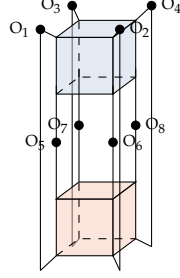
Further on, we adopt the so-called "folded" representation of the tensor network [3], by combining two rank 8  $d$ -dimensional tensors  $U$  and  $U^\dagger$  to one rank 8 ( $d^2$ )-dimensional tensor  $W \in \mathbb{R}^{d^8, d^8}$ . This trick has been used in several successful applications [26, 6, 30], and allows us to represent this new gate in the basis of different operators

## 2. Problem Setting

---

$\mathcal{B}_f = \{|\circ\rangle, |\bullet_j\rangle, j = 1, \dots, d^2 - 1\}$  being applied to the individual legs:

$$\langle O_1, O_2, O_3, O_4 | W | O_5, O_6, O_7, O_8 \rangle = \text{tr}[(O_1 \otimes O_2 \otimes O_3 \otimes O_4) U^\dagger (O_5 \otimes O_6 \otimes O_7 \otimes O_8) U]. \quad (2.6)$$



(2.7)

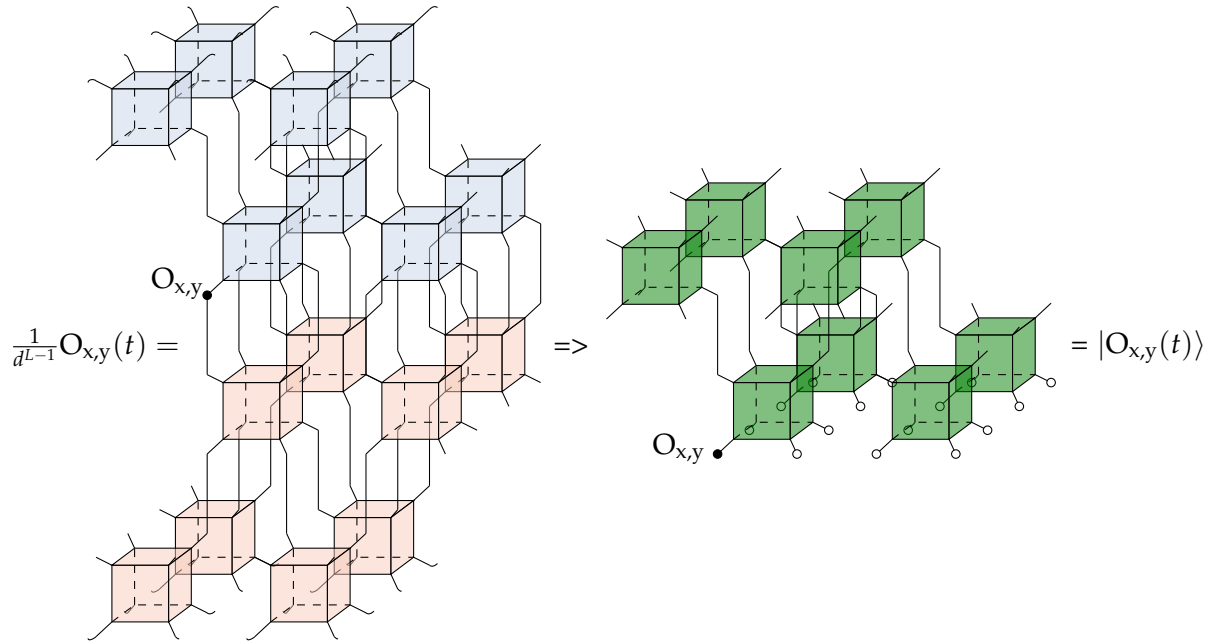
It is straightforward to see if either the top four or bottom four operators are the identity operator, the unitarity of the original gate  $U$  leads to the gates contracting. The unitarity is therefore conserved in the relation:

$$W |\text{Id}, \text{Id}, \text{Id}, \text{Id}\rangle = \text{[Green cube with legs]} = \text{[Four vertical lines]} = |\text{Id}, \text{Id}, \text{Id}, \text{Id}\rangle \quad (2.8)$$

Furthermore, we are in the limit of infinite temperature states, which translates to applying the identity state on both the top and bottom sides of the network before the folding. The representation of a time-evolved operator  $O$  acting on this initial state on site  $(x, y)$  is as follows (for clarity, only one time step and only a small number of tensors is shown):

## 2. Problem Setting

---

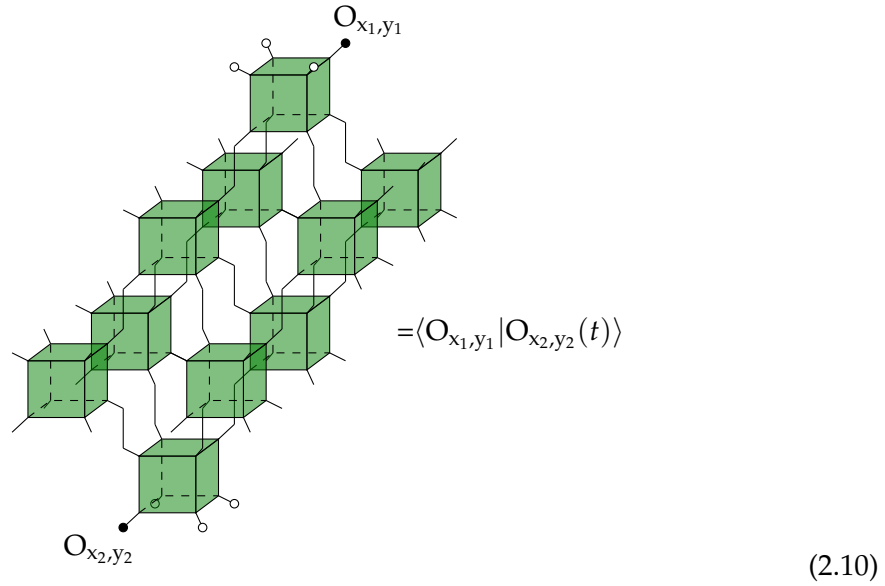


(2.9)

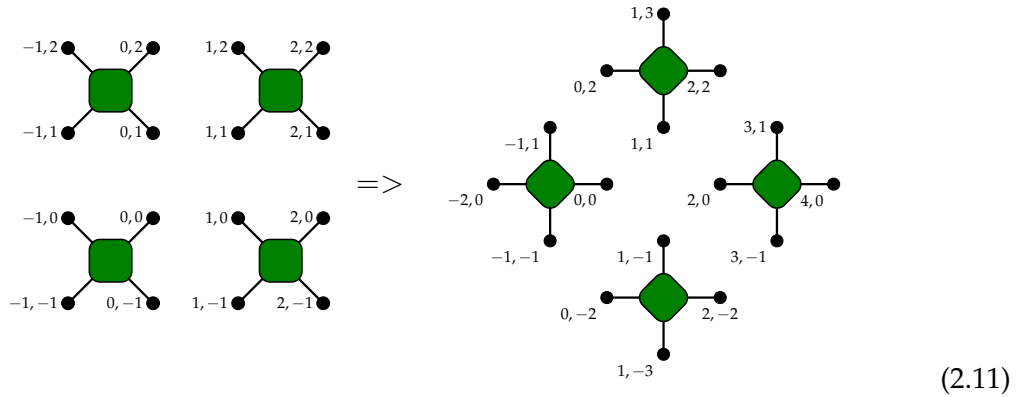
To get the correlation function  $\langle O_{x_1, y_1}^1 | O_{x_2, y_2}^2 \rangle$ , one needs to apply the second operator on the corresponding position at the top of the diagram, and Id operators everywhere else. For simplicity, we can assume the position of the bottom operator from now on to be  $(0,0)$ , since any other position corresponds to a shift in the coordinates of the top operators, or a rotation of the system, and denote the correlation function  $\langle O_{0,0}^1 | O_{x_2, y_2}^2 \rangle$  as  $C(x, y, T)^{O^1, O^2}$ . Assuming now, that we are in the thermodynamic limit of  $L - |x| > T$  and  $L - |y| > T$ , and using the Unitality relation, this leads to a simplified structure of a (possibly shifted) double pyramid:

## 2. Problem Setting

---



To allow for an easier description of the directions, the whole network gets turned by  $45^\circ$  around the time axis:



It's important to note, that this is a top-down view on one slice in time, where each of the green tensors has 8 legs, 4 upwards and 4 downwards, that overlap in this view. The coordinates of the individual sites change from  $(x,y)$  to  $(x+y,y-x)$ . In this picture, each tensor leg goes either up, down, left or right, and applying one tensor always goes one step forward in time. Other than in the 1+1 dimensional case, where time and space directions were combined, here we combine the two space dimensions, and time is untouched, still describing the number of operations that have to be applied.

### 2.3. Ternary unitary gates

A special case are so-called ternary unitary gates, which expand the unitarity relation into all 3 directions and are from here marked in orange. This threefold unitarity relation can also be directly translated into the folded picture:

$$(2.12)$$

$$(2.13)$$

It has been shown, that this kind of gate only produces finite correlation functions for operators directly on the light beam of the first operator. This result can be explained intuitively in the folded picture:

$$\langle O_{x_1, y_1} | O_{x_2, y_2}(t) \rangle = \dots = \dots = \dots$$

$$(2.14)$$

The Identity operators are again, for visibility, not shown everywhere but only at the top and bottom, and where the ternary unitarity relation is used. By repeatedly applying it, the entire network starts collapsing from the left and right, until only traces over the operators are left, which are zero for all traceless operators. The only case where this is not the case is if there are no gates that have 4 identities applied on any sideward face, i.e. if the second operator is exactly on the firsts lightray, in which case the correlation function is easily calculated by  $\langle O_{x',y'} | O_{x,y}(t) \rangle = \langle O_{x',y'} | W^T | O_{x,y}(0) \rangle$ . While this result allows for very fast calculation of these functions, it is of limited interest. Instead, we now want to further expand this on systems that have non-trivial correlations inside the light pyramid, by looking at two different ways of breaking this relation:

1. **low density, unit strength** The system mainly consists of ternary unitary gates, with only individual sites being exchanged for non-ternary unitary ones. There is no need for a regular pattern, it is only assumed that the distance between two perturbed sites is large. This case is mainly interesting for the argument of the later approximations.
2. **unit density, low strength** The system consists only of non-ternary unitary, however, the parameter that describes the strength of the perturbation on each gate is assumed to be small. This case shows non-trivial correlation functions inside the entire light pyramid.

## 2.4. Gate generation

### General Gates

While the generation of general ternary unitary gates is still under investigation, there are ways of generating a subset of them from dual unitaries. From now we focus on the smallest possible set of gates, with dimension 2 on each leg, where a general unitary form for rank 4 tensors (two site gates) is known [19, 37]:

$$U = e^{i\phi} (u_1 \otimes u_2) V[J] (u_3^\dagger \otimes u_4^\dagger), \quad (2.15)$$

with

$$V[J_1, J_2, J_3] = \exp[i(J_1 \sigma^x \otimes \sigma^x + J_2 \sigma^y \otimes \sigma^y + J_3 \sigma^z \otimes \sigma^z)] \quad (2.16)$$

$$u_i = e^{i(\alpha_i/2)\sigma^z} e^{i(\beta_i/2)\sigma^y} e^{i(\gamma_i/2)\sigma^x}. \quad (2.17)$$

These gates, originally having 16 parameters, are dual unitary for  $J_1=J_2=\frac{\pi}{2}$ . By instead choosing  $J_1=J_2=\frac{\pi}{2} + \eta$ , we get a parameter controlling the strength of the perturbation around the dual unitary and later ternary unitary point. These gates can be combined in

two different fashions to generate (almost) ternary unitary gates. Either by generating 4 such dual unitaries, and "stacking" them to each connect 2 neighboring sites, or using only 2 of these and connecting 2 diagonally neighboring sites each, as seen in figure 2.1, creating a "crossed" structure. For the crossed structure, the two dual unitaries never connect, and the number of parameters of the folded gate is quite straightforward. With each gate having 14 parameters, but the phase canceling out in the folded picture, we gain a total of 26 parameters plus the parameter for breaking the ternary unitarity  $\eta$ . For the stacked structure, 4 dual unitaries are originally constructed, however, several single site gates cancel out, as seen in figure 2.1. By only allowing 3 rotational gates within two site gates, this leads to a final number of 40 free parameters. We further denote the folded gate with a perturbation  $\eta$  as  $W_\eta$ , with the exception of  $W_{\eta=0} = W_{\text{tu}}$  being the ternary unitary folded gate.

### Reduced Gates

Following the original 1+1D idea, a reduced case is introduced to investigate the validity of the generalization to 2+1D, as well as allowing for a more intuitive description of the methods used later. The corresponding gates are based on the more general, stacked, exactly ternary unitary gate  $U$  described earlier ( $J_1=J_2=\frac{\pi}{2}$ ):

1. A magnetic field of variable strength is applied on each site, which corresponds to the gate  $U$  being modified as

$$\tilde{U} = (e^{\phi_1\sigma_z} \otimes e^{\phi_2\sigma_z} \otimes e^{\phi_3\sigma_z} \otimes e^{\phi_4\sigma_z}) \cdot U \cdot (e^{\phi_5\sigma_z} \otimes e^{\phi_6\sigma_z} \otimes e^{\phi_7\sigma_z} \otimes e^{\phi_8\sigma_z}) \quad (2.18)$$

2. The correlation function is averaged with respect to each magnetic field strength individually. This corresponds to applying the single site Haar-measure of the  $U(1)$  group and results in the projection of the folded gate  $W_{\text{tu}}$  to the subspace spanned by the Identity and Sz operator, leading to the reduced folded gate  $w$ .

Intuitively, this can be understood as only the Identity and Sz operators surviving the average, and all other elements of the folded gate to be zero. This folded reduced gate has dimension 2 on each leg, acting non trivially only in the basis  $\mathcal{B} = \{|\circ\rangle, |\bullet\rangle\}$ , and therefore allows for benchmarking on larger system sizes, and an on/off interpretation on the legs, either having the Identity or Sz operator applied, leading to only one correlation function being non-zero:  $C(x, y, T)^{\sigma_z, \sigma_z} = C(x, y, T)$ . As shown later, while this Haar measure does simplify the problem massively, it does not result in trivial results for the correlation functions. Of this reduced folded gate, only 32 elements are needed to be zero for the ternary unitarity condition to hold. Instead of modifying the parameters  $J_1, J_2$  as in the general case, we add a perturbation  $\epsilon \cdot w_{\text{pert}}$  after folding

## 2. Problem Setting

---

the gate, giving more precise control over the strength of the perturbation. The gate  $w_{\text{pert}}$  has non-zero elements only for exactly those 32 elements. Additionally, due to the commutation relation  $[e^{i(\alpha/2)\sigma^z}, \sigma^z] = 0$ , the outer single site  $z$  rotations also vanish, leading to only 36 parameters.

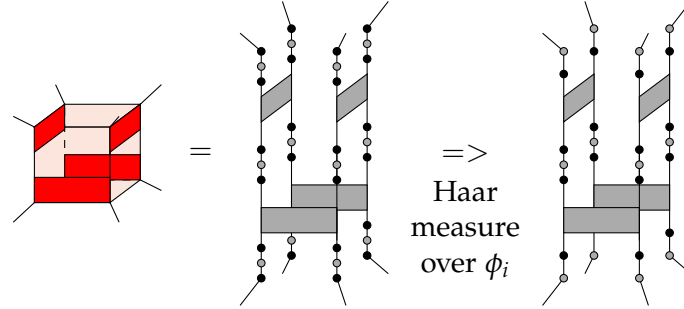


Figure 2.1.: Left: Construction of stacked ternary-unitary gate  $U$  out of dual-unitary ones.

Middle: Individual one- and two-site operators for general ternary-unitary gates, with the  $\sigma_z$  rotation gates being marked in black and  $\sigma_y$  rotations in grey.

Right: Construction for reduced gates, with the outer  $8 e^{i(\alpha_i/2)\sigma^z}$  operations vanishing in the average.

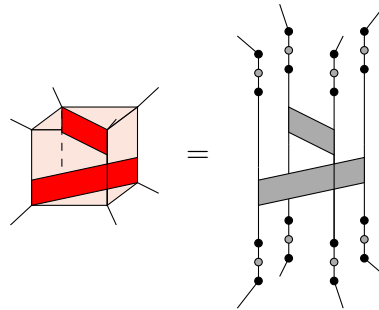


Figure 2.2.: Left: Construction of crossed ternary-unitary gate  $U$  out of dual-unitary ones.

Right: Individual one- and two-site operators for general ternary-unitary gates.



## 3. Strategy and Conditions

### 3.1. Path sum representation

From now on we focus on the minimal version by looking at two-dimensional wires and the reduced gates. This means that at each leg of the reduced gate  $w$  only 2 states can be applied: The identity or Sz state. The correlation function  $C(x,y,T)$  can then be calculated by inserting the identity in form of  $|\text{Sz}\rangle \langle \text{Sz}| + |\text{Id}\rangle \langle \text{Id}| = |\bullet\rangle \langle \bullet| + |\circ\rangle \langle \circ|$  at each leg. This returns a large number of  $2^{V(x,y,T)}$  terms, where  $V(x,y,T)$  is the volume of the light pyramid spanned by the operators. However, a significant portion of these have zero contribution. In this representation each element of the folded gate stands for one piece, connecting Sz operators on the tensor legs with each other and the correlation function is simply the sum over all paths. These pieces can be grouped into different types:

bare propagator    turn    curve    merge    split    complex

(3.1)

To rule out all the paths that have zero contribution, a set of "rules" can be formed for these paths:

1. The two projectors in the inserted identity have no overlap => "Sz legs" must connect to "Sz legs"
2. Unitality leads to zeros in the first row and column => no turns in time (zero Sz states connecting upwards) are allowed
3. Expansion in  $\epsilon$  => all turns, curves and other elements that have zero Sz states on one sideways face are suppressed with  $\epsilon$

Intuitively, this simply means we only need to look at paths that connect the two applied operators, without having any dead ends or turns in time:

$$C(4,2,2) = \text{[Diagram 1]} + \text{[Diagram 2]} + \text{[Diagram 3]} + \dots \quad (3.2)$$

### 3.2. Skeleton Paths simplification

When summing over all these paths, one essentially does the contraction of the network sequentially, leading to the same result, with worse runtime but less memory required. In order to improve the actual runtime, we further improve this analogously to the 1+1D case. We therefore approximate the result with so-called "skeleton paths", meaning all paths with constant width of one. For these paths, only 16 elements are relevant, according to their in- and output directions in the rotated picture:

output/input	$\ell$	r	b	t
r	$b_{1,1}$	$\epsilon \cdot b_{2,1}$	$\epsilon \cdot b_{3,1}$	$\epsilon \cdot b_{4,1}$
$\ell$	$\epsilon \cdot b_{1,2}$	$b_{2,2}$	$\epsilon \cdot b_{3,2}$	$\epsilon \cdot b_{4,2}$
t	$\epsilon \cdot b_{1,3}$	$\epsilon \cdot b_{2,3}$	$b_{3,3}$	$\epsilon \cdot b_{4,3}$
b	$\epsilon \cdot b_{1,4}$	$\epsilon \cdot b_{2,4}$	$\epsilon \cdot b_{3,4}$	$b_{4,4}$

Table 3.1.: Skeleton path elements of reduced gates to allow for an easier description of skeleton paths. For ternary unitary gates that are perturbed with an additional gate, all turns and curves are suppressed by a factor of  $\epsilon$

The 4 bare propagators are on the diagonal and are the dominant part, while the 4 turn and 8 curve weights are each suppressed by  $\epsilon$  and set to zero for the pure ternary unitary case. While this can be exact in a handful of cases, it is a good approximation for a larger number of cases, namely whenever the more complex paths are suppressed due to smaller merge or split weights. While we keep the exact conditions for this

### 3. Strategy and Conditions

---

to hold for a later section, the intuitive explanation for the low density, unit strength is as follows: When constructing the network, only a handful of gates will be the non-ternary unitary kind. The network can now be separated into multiple layers, parallel to one of the faces of the pyramid. While the paths can still take a large number of complex shapes, they are limited to straight propagation between layers involving perturbed gates. Within those layers, several different starts can be constructed. Under the assumption, that the density is low and we are far away from the opposite face of the pyramid (in this case large  $x'$ ), the distance between two perturbed layers becomes large. One can now construct the operator of one such layer, and calculate its eigenvectors and eigenvalues, and only the eigenvector with the highest eigenvalue will give the dominant contribution. To visualize this, the skeleton contribution to the correlation function for such a locally disturbed ternary unitary network with  $y'=0$  can be expressed as:

$$C(4,2,2) = \text{Diagram 1} = \text{Diagram 2} + \text{Diagram 3} \tag{3.3}$$

The diagram shows the equation  $C(4,2,2) = \text{Diagram 1} = \text{Diagram 2} + \text{Diagram 3}$ . Diagram 1 is a single path of 6 cubes: a green cube at the bottom left, an orange cube above it, a green cube to the right, another orange cube above that, and a final green cube at the top right. Diagram 2 is a path of 4 cubes: a green cube at the bottom left, an orange cube above it, a green cube to the right, and a final green cube at the top right. Diagram 3 is a path of 3 cubes: a green cube at the bottom left, an orange cube above it, and a final green cube at the top right. Each cube is a 3D representation with dashed lines for hidden edges and a small black dot at one vertex.

This figure is simplifying the underlying concepts drastically to keep the visualization clearly structured. For larger systems, the number of skeleton and non skeleton paths grows exponentially, and an important assumption is that the distance between the perturbed gates is large.

The argument for the unit density, low strength case is highly similar: For each order  $n$  in the perturbation expansion,  $n$  turns are scattered across the entire path. The majority of terms will still have large distances between these, and the argument still holds, assuming large distances between the first and second operator. However, the calculation becomes even more complicated: Instead of having fixed points in the network, where turns/curves are allowed, they can be anywhere, but in each order  $n$  of  $\epsilon$  only  $n$  of them are allowed. The question of how to evaluate these large numbers of terms is addressed later.

### 3.2.1. Exact cases

While not necessarily of physical interest, there are cases where the skeleton results are exact. These cases can still be useful for understanding the basic idea, as well as verifying the correctness of the skeleton computation compared to a full contraction of the network. These can be put into two groups:

1. **Reduced curves:** If the ternary unitary elements are not all zero, but in each of the two directions (up/down or left/right) there is only one non-zero curve or turn element, this allows the correlation function to be non-trivial on a number of coordinates within the light pyramid, but not all. This can be explained by the conditions for allowed paths: In order to have no open ends, only 3 different direction changes are allowed for each direction. By having zero weights for all 3 of these turns in 2 of the directions, the paths are "locked in" after a maximum of 2 turns, not allowing the formation of any complex paths.
2. **No merge or split weights:** If either all merge or all split weights are zero. While this is a very straightforward reason as to why only skeleton diagrams can contribute, this interestingly is a case independent of the ternary unitary attribute.

### 3.3. Conditions for bound approximation error

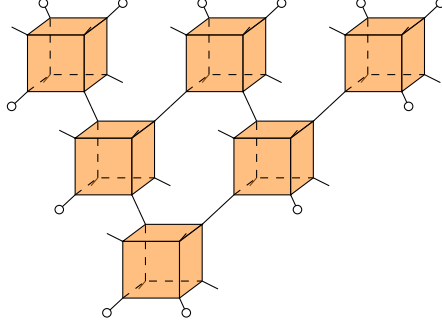
The skeleton path approximation does not always give the dominant contribution to the correlation function, in fact, multiple specific requirements must be fulfilled for this to be the case. In the case of a low density of perturbed gates, one can assume several layers of operators between each perturbed gate. These layers are parallel to one of the light pyramid's faces, and are applied repeatedly to gain the correlation functions. Assuming a large distance between two of these layers, only the eigenvector with leading eigenvalue will give a dominant contribution. We therefore group the eigenvectors into two groups: Eigenvector of support 1, i.e. of the form  $|\circ\dots\circ\bullet\circ\dots\circ\rangle$ , that have a non-identity operator acting on only one single site, and those of support greater than one, acting on at least two sites.

Furthermore, the exact structure of these transfer operators must be defined. The faces

### 3. Strategy and Conditions

---

of the pyramid are triangular networks of folded gates:



(3.4)

However, opposite sides have opposite orientations, and each layer in between changes from one shape to the other. In order to guarantee all kinds of these shapes, this triangle gets completed to a square lattice of folded gates. Additionally, these layers are applied in a shifted manner, meaning that a state acting on two sites on one gate in one layer, outputs on two sites on two different gates on the next. Keeping this in mind, and furthermore requiring all paths to start in one common location and converge together again, we can limit our investigation of the transfer operator to parallel paths, meaning reducing the two open directions to one by applying the identity state on one input and output leg. The resulting operator with side lengths  $j$  and  $k$  is further regarded to as  $A_{j,k}^{l,n} \in \mathbb{R}^{d^{j \times k}, d^{j \times k}}$ , with the upper index  $l$  and  $n$  identifying the states applied on the right and left side in a binary representation  $\{l_1, l_2, \dots\}_2$ , is defined as:

$$A_{j,k}^{l,n} := \sum_{m_1, \dots, m_j=1}^{d^k} A_{1,k}^{l,m_1} \otimes A_{1,k}^{n_1,m_2} \otimes \dots \otimes A_{1,k}^{m_j,n} \quad (3.5)$$

$$A_{1,k}^{l,n} := \sum_{m_1, \dots, m_k=1}^d W_{l_1,0,n_1}^{l_2,m_1,n_2} \otimes W_{l_3,0,n_3}^{l_4,m_2,n_4} \otimes \dots \quad (3.6)$$

$$W_{l_1,0,n_1}^{l_2,m_1,n_2} := \langle l_1 |_0 \langle 0 |_2 \langle n_1 |_4 W |l_2 \rangle_4 |m_1 \rangle_5 |n_2 \rangle_7 \quad (3.7)$$

### 3. Strategy and Conditions

$$A_{j,k}^{00} = \text{[Diagram 1]} \quad A_{j,k}^{3,9} = \text{[Diagram 2]} \quad (3.8)$$

With 4 different orientations of the operator, and 2 propagation directions each, there are 8 operators to be investigated in total,  $A_{j,k}^{00} - H_{j,k}^{00}$ .

By defining the projectors of the subspaces of the pure identity state and the subspaces of eigenvectors of support 1

$$P_{j,k}^0 := |\circ\rangle^{\otimes j \cdot k} \langle \circ|^{\otimes j \cdot k} \quad (3.9)$$

$$P_{j,k}^{x,y} := |\circ\rangle^{\otimes y \cdot k+x} |\bullet\rangle |\circ\rangle^{\otimes j \cdot k - y \cdot k+x} \langle \circ|^{\otimes y \cdot k+x} \langle \bullet| \langle \circ|^{\otimes j \cdot k - y \cdot k+x} \quad (3.10)$$

we can conclude

*Property 1 a):* The operator  $A_{j,k}^{00}$  in the reduced case takes the following block diagonal form:

$$A_{j,k}^{00} = P_{j,k}^0 + b_{3,3} \sum_{x=1, y=1}^{x=j, y=k} P_{j,k}^{x,y} + r_{j,k}, \quad (3.11)$$

*Property 1 b):* The operator  $A_{j,k}^{00}$  in the general case takes the following block diagonal form:

$$A_{j,k}^{00} = P_{j,k}^0 + \sum_{x=1, y=1, l=1}^{x=j, y=k, l=3} \lambda_l P_{j,k}^{x,y,l} + r_{j,k}, \quad (3.12)$$

with  $\lambda_l$  and  $v_l$  being the eigenvalues of eigenvectors of the gate  $\langle \circ \circ \circ | W | \circ \circ \circ \rangle$  and

$$P_{j,k}^{x,y,l} := |\circ\rangle^{\otimes y \cdot k+x} |v_l\rangle |\circ\rangle^{\otimes j \cdot k - y \cdot k+x} \langle \circ|^{\otimes y \cdot k+x} \langle v_l| \langle \circ|^{\otimes j \cdot k - y \cdot k+x} \quad (3.13)$$

This property intuitively splits the reduced transfer operator into 3 parts, the identity projector, the projector of the subspace of eigenvectors of support 1, and the rest acting

only on eigenvectors of support more than 1. Therefore, we can conclude

*Property 2:* Under the assumption, that

1. The minimal distance  $\mu_1$  between two layers of perturbed gates is large
2. The norm of the rest term  $|r_{j,k}|$  can be bound by some value  $r < b_{3,3}$  or in the general case  $r < \max(\lambda_l)$

We can substitute  $A_{j,k}^{00}$  with  $P_{j,k}^0 + b_{3,3} \sum_{x=1, y=1}^{x=j, y=k} P_{j,k}^{x,y}$ , and the relative error  $R(x,y,T) = \left| \frac{C(x,y,T) - C_{sk}(x,y,T)}{C(x,y,T)} \right|$  is bound by  $\mathcal{O}\left(\left(\frac{r}{b_{3,3}}\right)^{\mu_1}\right)$  or in the general case  $\mathcal{O}\left(\left(\frac{r}{\max(\lambda_l)}\right)^{\mu_1}\right)$ . The same reasoning works for each of the other 7 planar operators  $B_{j,k}^{00} - H_{j,k}^{00}$  and their respective spacing between perturbed gates. This substitution is equivalent to only taking skeleton diagrams into consideration.

The straightforward way of binding this norm, constructing the operator  $A_{j,k}^{00}$  from the folded gate  $W$ , removing the eigenvectors of support zero and one, and calculating the leftover norm or eigenspectrum, is unfortunately not viable. Due to the 2D structure, the resulting operator with side lengths  $j$  is a  $2^{j^2} \times 2^{j^2}$  matrix in the reduced case ( $4^{j^2} \times 4^{j^2}$  in the general case), resulting in a runtime of  $\mathcal{O}(2^{j^2})$ , and even for a side length of just 4, already 32 GB of memory are required assuming 64-bit double precision floating point numbers. Therefore, a more efficient method is needed.

### 3.3.1. Condition 1: planar propagators

*Property 3:* The operator  $A_{j,k}^{00}$  also takes the following alternative block diagonal form:

$$A_{j,k}^{00} = P_{j,k}^0 + \sum_{m=1, x=1}^{m=2^k, x=j} \Lambda_m P_{j,k}^{x,m} + R_{j,k}, \quad (3.14)$$

where  $\Lambda_m$  and  $V_m$  are the eigenvalues and eigenvectors of  $A_{1,k}^{00}$ , and

$$P_{j,k}^{x,m} := |\circ\rangle^{\otimes x} |V_m\rangle |\circ\rangle^{\otimes k-2} \langle \circ|^{\otimes x} \langle V_m| \langle \circ|^{\otimes k-2}. \quad (3.15)$$

This divides the operator  $A_{j,k}^{00}$  into 3 parts: The pure identity eigenstate, a number of complex eigenstates that are limited to a single vertical slice of the operator, and a rest that is acting on an orthogonal subspace. With this, we can bind the eigenvalues of all eigenvectors that are acting on more than one slice by binding the norm of the last term  $|R_{j,k}| = \tau_{j,k}$ , and requiring it to be smaller than the bare propagator weight with the recursive condition:

### 3. Strategy and Conditions

---

*Condition 1:*

$$\begin{aligned}\tau_{j,k} &\leq \max(\tau_{j-1,k}, |R_{1,k}^{00}| \cdot |\tau_{j,k}^*| + \sum_m^{2^k} |A_{1,k}^{0m}| \cdot |A_{j-1,k}^{m0}|) \leq b_{3,3} \\ \tau_{j,k}^* &\leq \max(\tau_{j-1,k}^*, |R_{1,k}^{00}| \cdot |A_{j-1,k}^{00}| + \sum_m^{2^k} |A_{1,k}^{0m}| \cdot |A_{j-1,k}^{m0}|)\end{aligned}\tag{3.16}$$

*Proof:* To bind the norm, we first recursively define the planar propagator  $A_{j,k}^{00}$  as followed:

$$A_{j,k}^{00} = A_{1,k}^{00} \otimes A_{j-1,k}^{00} + \sum_{m=1}^{2^k} A_{1,k}^{0m} \otimes A_{j-1,k}^{m0}\tag{3.17}$$

This corresponds to inserting the identity in the shape of  $\sum_{m=0}^{2^k} |m\rangle \langle m|$ , where the left term is simply the inserted identity state. The right turn includes a linear slice and another planar structure with reduced length, where  $m$  is the inserted state in binary notation. The resulting planer operator with arbitrary states applied to its left side can be separated as well:

$$A_{j,k}^{ln} = A_{1,k}^{l0} \otimes A_{j-1,k}^{0n} + \sum_{m=1}^{2^k} A_{1,k}^{lm} \otimes A_{j-1,k}^{mn}\tag{3.18}$$

Additionally, the shape of some of those slices can be shown to be:

$$A_{1,k}^{00} = \begin{pmatrix} 1 & 0 \\ 0 & R_{1,k}^{00} \end{pmatrix} \quad A_{1,k}^{0m} = \begin{pmatrix} 0 & 0 \\ 0 & R_{1,k}^{0m} \end{pmatrix} \quad A_{1,k}^{m0} = \begin{pmatrix} 0 & 0 \\ 0 & R_{1,k}^{m0} \end{pmatrix}\tag{3.19}$$

It is now important to differentiate between the two terms on the right side:

1. The state with the inserted identity has no "connections" between the two parts, both can however contain whatever projectors limited to their subspace. The only parts that contribute to the first two terms of equation 3.18 are the identity projector in either of the two structures or both. Intuitively, this means if no operators are applied in the slice, one simply needs to look on the right side, if no operators are applied on the right side, the left side falls into the subspace of the left two terms of equation 3.18.
2. The states with inserted operators in between definitely have operators applied on both sides and therefore fall into the subspace of the rest term defined in equation 3.18.



### 3. Strategy and Conditions

---

By using these separations into subspaces, and in a second step applying the triangle inequality on the expansion of the propagator, the norm of  $r_{j,k}$  can be bound by:

$$\begin{aligned}
\tau_{j,k} &:= |R_{j,k}| = \max(\tau_{j-1,k}, |R_{1,k}^{00} \otimes (A_{j-1,k}^{00} - P_{j-1,k}^{00}) + \sum_m^{2^k} A_{1,k}^{0m} \otimes A_{j-1,k}^{m0}|) \\
&\leq \max(\tau_{j-1,k}, |R_{1,k}^{00}| \cdot |(A_{j-1,k}^{00} - P_{j-1,k}^{00})| + \sum_m^{2^k} |A_{1,k}^{0m}| \cdot |A_{j-1,k}^{m0}|) \quad (3.20) \\
&= \max(\tau_{j-1,k}, |R_{1,k}^{00}| \cdot |\tau_{j,k}^*| + \sum_m^{2^k} |A_{1,k}^{0m}| \cdot |A_{j-1,k}^{m0}|),
\end{aligned}$$

which involves the new norm  $\tau_{j,k}^*$ , defined as the norm of the shortened operator without the identity projector. Using the same inequality again leads to

$$\begin{aligned}
\tau_{j,k}^* &:= |(A_{j-1,k}^{00} - P_{j-1,k}^{00})| = \max(\tau_{j-1,k}^*, |R_{1,k}^{00} \otimes A_{j-1,k}^{00} + \sum_m^{2^k} A_{1,k}^{0m} \otimes A_{j-1,k}^{m0}|) \\
&\leq \max(\tau_{j-1,k}^*, |R_{1,k}^{00}| \cdot |A_{j-1,k}^{00}| + \sum_m^{2^k} |A_{1,k}^{0m}| \cdot |A_{j-1,k}^{m0}|) \quad (3.21)
\end{aligned}$$

It is clear to see that  $\tau_{1,k} = 0$  and  $\tau_{1,k}^* = |R_{1,k}^{00}|$ , which allows to recursively solve for an upper bound of  $\tau_{j,k}$  to limit the highest eigenvalue of any eigenvector of  $A_{j-1,k}^{00}$  that acts on more than one slice, concluding the proof.

While the direct approach required the norm or eigenvalue decomposition of an operator of size  $2^{j \cdot k} \times 2^{j \cdot k}$ , this requires the norm of  $2^j$  operators of size  $2^j \times 2^j$ , drastically reducing the amount of memory and time required.

The implementation of this can be described in five steps:

1. Construct chains of operators for all  $2^j$  possible states applied on each side (It is crucial to apply the states on each gate before combining them to the chain, to reduce computational overhead)
2. Calculate the norm of each chain
3. Calculate the norm of blocks of length up to  $j-1$ , as required for the calculation of  $\tau_{j,k}$
4. Start recursively calculating  $\tau_{j,k}^*, \tau_{j-1,k}^*, \dots$
5. Calculate recursively  $\tau_{j,k}, \tau_{j-1,k}, \dots$

### 3. Strategy and Conditions

---

It should be noted, that this method can be used for both reduced and generic gates. The only difference is, in the generic gate, the parameter to compare the rest norm to is not simply the weight of the Sz propagator, but the highest eigenvalue of the gate:

$$\begin{array}{c} \text{Diagram of a 3D gate with 8 input/output nodes} \end{array} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & b_{4,4}^{1,1} & b_{4,4}^{1,2} & b_{4,4}^{1,3} \\ 0 & b_{4,4}^{2,1} & b_{4,4}^{2,2} & b_{4,4}^{2,3} \\ 0 & b_{4,4}^{3,1} & b_{4,4}^{3,2} & b_{4,4}^{3,3} \end{pmatrix} \quad (3.22)$$

However, this can only be done for a finite-sized layer. While this does not give a rigorous condition, under which the skeleton paths will be dominant for arbitrary-sized layers, when analyzing the norm of the rest for various sizes, as done in figure 3.1, strong evidence is shown that there are gates for which this holds even for arbitrarily large structures. Assuming this, one can expect to see 3 different classes of gates:

1. Gates, where the skeleton diagrams dominate for arbitrarily large structures. In this case, the bound should quickly stop growing with size and stay below the bare propagator.
2. Gates, where there is an eigenvector with support larger than one, with the highest eigenvalue, yet only a limited number of those. The bound grows slowly, but surpasses the bare propagator eigenvalue.
3. Gates, where some of the split and merge weights are high enough, that eigenvalues of vectors with large support keep growing, resulting in the bound to grow rapidly.

In figure 3.1, the upper bound for the rest norm for each of those 8 layers of growing size is shown for gates 1-4. For the first 3 gates, the condition holds, leading to an almost immediate stop of growth. For the last gate, this condition is broken and the rest norm grows rapidly. It is important to note, that while the bound can grow arbitrarily large, the norm itself is limited to values smaller than one.

### 3. Strategy and Conditions

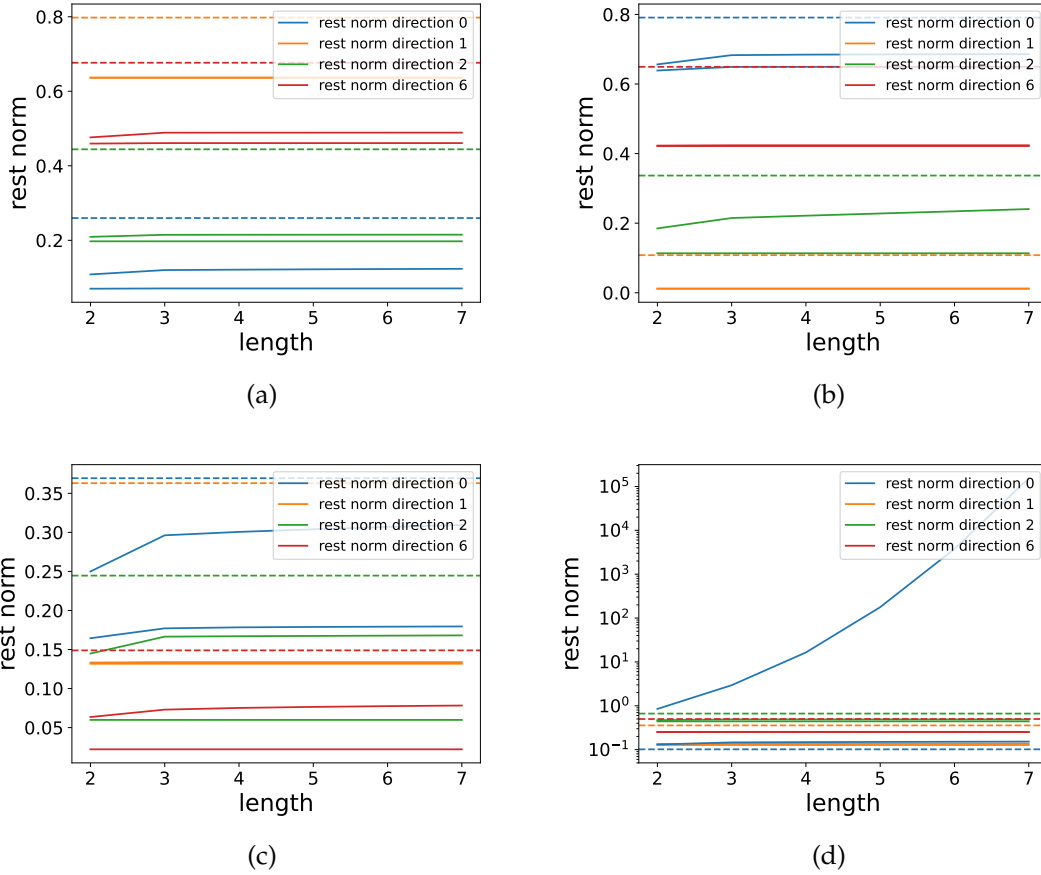


Figure 3.1.: Growth of the upper bound of the rest norm  $\tau_{j,k}$  as described in 3.20 for gates 1-4 with regard to the side length of the operator layer. As expected, for gates 1-3 the bound stagnates quickly and stays under the corresponding eigenvalue of support 1 (shown in dashed lines). For gate 4, the terms of the bound add up rapidly, meaning the correlation of this gate is not captured by the skeleton approximation.

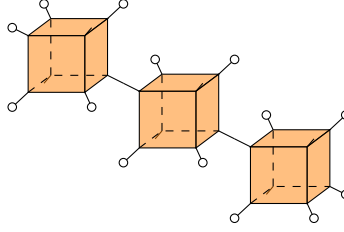
#### 3.3.2. Condition 2: Linear propagators

What is left to find is a bound of the eigenvalues of support larger than 1, that are limited to one slice of the propagator operator. For this, linear chains of the gates must be constructed, and all but on propagation directions are reduced with identity states

### 3. Strategy and Conditions

---

applied:



(3.23)

This propagator  $A_{1,k}^{\circ\circ}$ , with  $n$  being the length of the chain,  $i$  and  $j$  determining the direction of the structure and the open legs, and the upper index describing the states applied at the end, can for all directions be split apart into orthogonal subspaces again: *Property 4:* In the reduced case, the propagator  $A_{1,k}^{\circ\circ}$  takes the following block diagonal form:

$$A_{1,k}^{\circ\circ} = P_{1,k}^0 + b_{3,3} \sum_{y=1}^{y=k} P_{1,k}^{x,y} + r_{1,k}, \quad (3.24)$$

which is merely the reduction of condition 1 for a grid of width 1.

This propagator of length  $n$  can be described by propagators of length  $n-1$  with the following relation:

$$P_{n,i,j}^{\circ\circ} = P_{n-1,i,j}^{\circ\circ} \otimes \begin{pmatrix} 1 & 0 \\ 0 & a \end{pmatrix} + P_{n-1,i,j}^{\circ\bullet} \otimes \begin{pmatrix} 0 & 0 \\ 0 & b \end{pmatrix} \quad (3.25)$$

$$P_{n,i,j}^{\circ\bullet} = P_{n-1,i,j}^{\circ\bullet} \otimes \begin{pmatrix} c & d \\ e & g \end{pmatrix} + P_{n-1,i,j}^{\circ\circ} \otimes \begin{pmatrix} 0 & 0 \\ 0 & f \end{pmatrix} \quad (3.26)$$

This is, in fact, the exact same structure as in the 1+1D case. Additionally, the elements of the tensor were renamed to recover the same nomenclature as used in the 1+1D case ( $b_{3,3} = a$ ) and depend on both the direction of the propagator and the open direction. With this, one can derive

*Condition 2 a):*

$$|r_{1,k}| \leq a^2 + \frac{|bf|}{1-a} \quad \text{for} \quad a^2 + \frac{|bf|}{1-a} < |a| \quad (3.27)$$

This allows us to quickly check, whether condition 2 holds for a reduced gate, by applying this condition on the 12 possible linear propagators.

For generic gates, this recursive relation becomes more complicated, since each propagator of length  $n$  needs 4 propagators of length  $n-1$  to be accurately described. Instead, the Condition there is simply

*Condition 2 b):* The next to leading eigenvalue of  $A_{1,k}^{\circ\circ}$  has eigenvectors of support one and has a finite gap to the first eigenvalue with eigenvector of support higher than one. In order to investigate this property for a given length  $k$ , there is a number of ways:

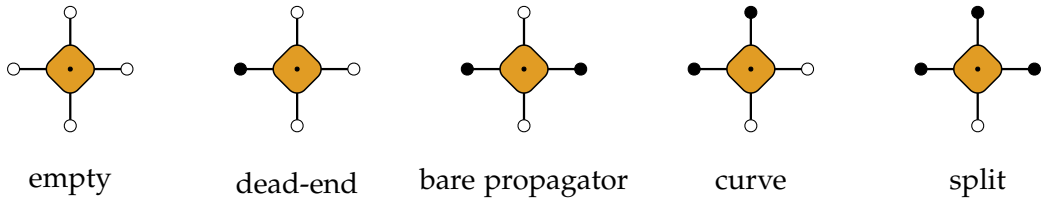
1. Evaluate the eigendecomposition of  $A_{1,k}^{\circ\circ}$  and compare the dominant eigenvectors.
2. Compute the norm of the operator  $A_{1,k}^{\circ\circ} - P_{1,k}^0 - \sum_{x=1,y=k,l=3} \lambda_l P_{1,k}^{x,y,l}$ , which acts as an upper bound of all eigenvalues with eigenvector of support more than one.
3. Use the power method to calculate the dominant eigenvector.

This shows a similar result as shown in figure 3.1, namely either rapidly growing eigenvalues of support higher than one, or barely any growth.

This leads to the final result:

*Property 5:* If both Condition 1 and either Condition 2a) or 2b) are fulfilled (depending on the kind of gate), the rest norm is bound to be smaller than the bare propagator/highest eigenvalue with eigenvector of support 1, and the error of the skeleton diagram approximation is bound by  $\mathcal{O}((\frac{r}{b_{3,3}})^{\mu_1})$  or respectively in the general case  $\mathcal{O}((\frac{r}{\max(\lambda_l)})^{\mu_1})$ .

### 3.3.3. Simplified case with path-picture



(3.28)

The total propagator layer can also be described as the sum over all different kinds of paths through it, however with far fewer restrictions. To visualize this, one can imagine it as a 2D layer of tensors, each connecting to 4 neighbors, and a connection orthogonal to the drawing layer. With the addition of these two extra legs, all kinds of elements are allowed: Dead ends, lines, all 4 curves, splits and merges. With all curves being allowed, there is no clear time direction anymore, so the differentiation between splits and merges is redundant. With the triangle equation, one can get an upper bound of the norm of this propagator by drawing out all different paths, and by summing over the individual paths. To get the weights, however, one must instead take the norm of the matrices generated by applying the 4 states inside the layer. With the addition of dead ends, this leads to a massive amount of contributions. While the sum over all these paths does give an upper bound for the norm of the rest, this bound is not very tight and rarely helpful. To further refine and construct a toy model for this picture, to help understand how these complex eigenvectors can have relevant eigenvalues even though all individual elements are small, it can be reduced by assuming all merge and

### 3. *Strategy and Conditions*

---

splits weights within one layer to be zero. This leads to only the bare propagators and dead ends still existing. This leads to 8 of the terms in condition 1 being automatically fulfilled, and condition 2 now completely vanishes. This situation can be interpreted as followed:

The paths are not allowed to split within one of the 8 layer-propagators defined earlier. However, they can split along the 2 main diagonals of the system, and with a combination of curves and multiple of these splits and merges, there are still a number of complex thickened paths the full contraction considers. When manually constructing such gates, and calculating their correlation function and relative error, the results showed still a complex non-trivial correlation function and finite errors within the entire light pyramid.

## 4. Calculation

In order to calculate the skeleton correlation function, an efficient evaluation of all skeleton paths must be found. We first focus on the case of the reduced gate  $w$ , for which we show two different approaches of evaluation, each with its own advantages. For readability, we focus the discussion here on the case of density one, but small perturbation strength  $\epsilon$ , however, both approaches can be applied to the small density case as well. For the generic gate  $W$ , the simple path sums picture breaks down, which leads to a more complex evaluation and only one viable approach.

### 4.1. Reduced Gates

#### 4.1.1. Analytic order expansion approach

The total number of skeleton diagrams grows exponentially with  $T$ , and its individual terms are difficult to structure. We begin by constructing all possible strings of length  $T$  of turn/propagator/curve weights as described in 3.1. The maximum number of curves/turns is restricted by the width  $b$  of the resulting light pyramid. We can order these by their number, position and direction of curves/turns:

$$C(x', y', T) = \sum_{n=0}^w \sum_{l_1, \dots, l_n} \sum_{d_1, \dots, d_{n-1}} [b_{d_1, d_1}^{l_1} b_{d_1, d_2} b_{d_2, d_2}^{l_2} \dots b_{d_{n-1}, d_n} b_{d_n, d_n}^{l_n} \cdot \epsilon^n] \quad (4.1)$$

with the following constraints acting on  $l_1, \dots, l_n$  and  $d_1, \dots, d_{n-1}$ :

$$\sum_n \delta_{d_n, 1} \cdot l_n - \delta_{d_n, 2} \cdot l_n = x', \quad \sum_n \delta_{d_n, 3} \cdot l_n - \delta_{d_n, 4} \cdot l_n = y', \quad \forall i, j < n_{\max} : d_i \neq d_j \quad (4.2)$$

In order to gain an analytic solution to evaluate this, we want to develop the correlation function in orders of  $\epsilon$  and cut off at some order  $n_{\max}$ :

$$\begin{aligned} C(x', y', T) &= \sum_{n=0}^{n_{\max}} \sum_{l_1, \dots, l_n} \sum_{d_1, \dots, d_{n-1}} [b_{d_1, d_1}^{l_1} b_{d_1, d_2} b_{d_2, d_2}^{l_2} \dots b_{d_{n-1}, d_n} b_{d_n, d_n}^{l_n} \cdot \epsilon^n] + R_{n_{\max}}(x', y', T) \\ &= C_{n_{\max}}(x', y', T) + R_{n_{\max}}(x', y', T) \end{aligned} \quad (4.3)$$

#### 4. Calculation

---

To reduce the number of terms and gain an explicit sum instead of a large sum with certain conditions applied, all terms with the same curves and turns (without disregarding their order) are grouped together, and evaluated by a number of combinatorial factors and a system of linear equations:

$$C(x', y', T)_{n_{\max}} = \sum_{n=0}^{n_{\max}} \sum_{\substack{j_1=1 \dots j_{n-1}=1 \\ j_k \neq j_{k+1}}}^4 \sum_l \epsilon^n \cdot b_{j_0, j_1} \dots b_{j_{n-1}, j_n} \cdot b_{1,1}^\alpha \cdot b_{2,2}^\beta \cdot b_{3,3}^\gamma \cdot b_{4,4}^\delta \cdot \binom{\alpha + n_r}{n_r} \binom{\beta + n_l - 1}{n_l - 1} \binom{\gamma + n_t - 1}{n_t - 1} \binom{\delta + n_b - 1}{n_b - 1} \quad (4.4)$$

This n-th order approximation is the result of the following reordering steps:

1. construct all combinations of curves with length n that have the right input and output direction (depending on the x and y coordinate of operators). For matching directions this returns  $\frac{3}{4}(3^{n-1} - (-1)^{n-1})$  in n-th order, and  $\frac{1}{4}(3^n - (-1)^n)$  for non matching ones.
2. multiply each term by  $b_{1,1}^\alpha \cdot b_{2,2}^\beta \cdot b_{3,3}^\gamma \cdot b_{4,4}^\delta$  with  $d_x$  and  $d_y$  being the relative distance after the curves have been applied:
  - a)  $\alpha - \beta = \frac{d_x}{2}$
  - b)  $\gamma - \delta = \frac{d_y}{2}$
  - c)  $\alpha + \beta + \gamma + \delta = T - n$
  - d) If any of the 4 propagators are not allowed by the curve structure, set its exponent to zero. Otherwise, if all 4 are allowed (starting from n=3) add a sum over  $l = \alpha + \beta$
3. multiply each term by  $\binom{\alpha + n_r}{n_r} \binom{\beta + n_l - 1}{n_l - 1} \binom{\gamma + n_t - 1}{n_t - 1} \binom{\delta + n_b - 1}{n_b - 1}$ , with:

$$a) \binom{n}{k} = \begin{cases} \frac{n!}{k!(n-k)!} & \text{if } 0 \leq k \leq n \\ 1 & \text{if } k=n \\ 0 & \text{else} \end{cases}$$

- b)  $n_r$  is the number of turns outputting to the right etc.

Given the exponentially growing number of terms in each order, it is only viable to use the first couple of orders. On the other hand, similar to the 1+1D case needing the first two terms to cover the whole light cone, in 2+1D the first 4 orders are necessary to cover the entire light pyramid, which is why we show these explicitly for even x and y:



## 4. Calculation

---

1. 0th order:

$$a) \delta_{x,T} \delta_{y,T} b_{1,1}^T$$

2. 2nd order:

$$a) \delta_{x,y} b_{1,2} b_{2,1} \cdot b_{1,1}^{0.5(T-2+\frac{x'}{2})} \cdot b_{1,1}^{0.5(T-2-\frac{x'}{2})} \binom{0.5(T+\frac{x'}{2})}{1}$$

$$b) \delta_{y,T} b_{1,3} b_{3,1} \cdot b_{1,1}^{\frac{x'-2}{2}} \cdot b_{3,3}^{\frac{y'-2}{2}} \binom{0.5 \cdot x'}{1}$$

$$c) \delta_{x,T} b_{1,4} b_{4,1} \cdot b_{1,1}^{\frac{x'-2}{2}} \cdot b_{4,4}^{\frac{-y'-2}{2}} \binom{0.5 \cdot x'}{1}$$

3. 3rd order:

$$a) b_{1,2} b_{2,3} b_{3,1} \cdot b_{1,1}^{0.5(T-3+\frac{x'}{2})} \cdot b_{2,2}^{0.5(T-3-\frac{x'}{2})} \cdot b_{3,3}^{\frac{y'-2}{2}} \binom{0.5(T-1+\frac{x'}{2})}{1}$$

$$b) b_{1,2} b_{2,4} b_{4,1} \cdot b_{1,1}^{0.5(T-3+\frac{x'}{2})} \cdot b_{2,2}^{0.5(T-3-\frac{x'}{2})} \cdot b_{4,4}^{\frac{-y'-2}{2}} \binom{0.5(T-1+\frac{x'}{2})}{1}$$

$$c) b_{1,3} b_{3,2} b_{2,1} \cdot b_{1,1}^{0.5(T-3+\frac{x'}{2})} \cdot b_{2,2}^{0.5(T-3-\frac{x'}{2})} \cdot b_{3,3}^{\frac{y'-2}{2}} \binom{0.5(T-1+\frac{x'}{2})}{1}$$

$$d) b_{1,3} b_{3,4} b_{4,1} \cdot b_{1,1}^{\frac{x'-2}{2}} \cdot b_{3,3}^{0.5(T-4+\frac{y'}{2})} \cdot b_{4,4}^{0.5(T-4-\frac{y'}{2})} \binom{0.5 \cdot x'}{1}$$

$$e) b_{1,4} b_{4,2} b_{2,1} \cdot b_{1,1}^{0.5(T-3+\frac{x'}{2})} \cdot b_{2,2}^{0.5(T-3-\frac{x'}{2})} \cdot b_{4,4}^{\frac{-y'-2}{2}} \binom{0.5(T-1+\frac{x'}{2})}{1}$$

$$f) b_{1,4} b_{4,3} b_{3,1} \cdot b_{1,1}^{\frac{x'-2}{2}} \cdot b_{3,3}^{0.5(T-4+\frac{y'}{2})} \cdot b_{4,4}^{0.5(T-4-\frac{y'}{2})} \binom{0.5 \cdot x'}{1}$$

This approach has two advantages. First, if one is interested in correlation functions for a specific set of locations (x, y, T), but different gates, that all fulfill the necessary conditions and have small  $\epsilon$ , one only needs to solve for  $\alpha, \dots, \delta$  on each term once, and can simply plug in different parameters. Secondly, this analytical approach allows for derivatives with respect to the gates parameters, and in extension for the original Hamiltonians parameters, if one is interested in these. However, if one is interested in one specific gate on a large number of locations, one needs to solve for  $\alpha, \dots, \delta$  on each position individually. Additionally, the number of terms one needs to consider grows exponentially with  $n_{\max}$ , so this is only viable for small  $\epsilon$ , which is already a requirement for the skeleton approximation in the first place.

### 4.1.2. Numeric layer approach

As a second, numerical approach, a type of dynamic programming can be used to numerically evaluate the skeleton correlation function by calculating them layer by layer. This can best be described by an example with a small number of timesteps T: The skeleton correlation at T=2 only ever contains one term, consisting only of two

#### 4. Calculation

---

weights depending on the input and output directions, e.g.  $C_S(0,0,2) = \epsilon^2 \cdot b_{1,2} \cdot b_{2,1}$ ,  $C_S(1,1,1) = \epsilon \cdot b_{1,1} \cdot b_{1,2}$ . The correlation of positions on the third layer already contains up to 4 terms:  $C_S(0,0,3) = \epsilon^3 \cdot b_{1,2} \cdot b_{2,1} \cdot b_{1,2} + \epsilon^3 \cdot b_{1,3} \cdot b_{3,4} \cdot b_{4,2} + \epsilon^3 \cdot b_{1,4} \cdot b_{4,3} \cdot b_{3,2} + \epsilon \cdot b_{1,1} \cdot b_{1,2} \cdot b_{2,2}$ . This can however be rewritten as  $C_S(0,0,3) = C_S(0,0,2) \cdot \epsilon \cdot b_{1,2} + C_S(0,1,2) \cdot \epsilon \cdot b_{4,2} + C_S(1,0,2) \cdot \epsilon \cdot b_{3,2} + C_S(1,1,2) \cdot b_{2,2}$ .

This rephrasing allows us to describe the correlation function at any point  $(x, y, T)$  by only using the correlation function of the 4 positions below, relying on the fact that skeleton paths are only acting on one site at each time steps, allowing one to ignore all other positions. The skeleton correlation function  $C_S(x, y, T)$  on tensor leg  $k$  therefore only depends on the 4 results from the last layer, with their relative position depending on which output leg the wanted position is, and the 4 weights outputting to leg  $k$ :

$$C_S(x, y, T) = C_S(\tilde{x}, \tilde{y}, T - 1) \cdot a_{l,0} + C_S(\tilde{x} + 1, \tilde{y}, T - 1) \cdot a_{l,1} + C_S(\tilde{x}, \tilde{y} + 1, T - 1) \cdot a_{l,2} + C_S(\tilde{x} + 1, \tilde{y} + 1, T - 1) \cdot a_{l,3}, \quad (4.5)$$

with  $l = x \bmod 2 + 2 \cdot (y \bmod 2)$  defining the output leg, which is periodic in  $x$  and  $y$  with period 2,  $\tilde{x} = x \bmod 2$ ,  $\tilde{y} = y \bmod 2$  defining the 4 relative positions that can lead to a result in the next layer at  $(x, y, T)$ , and  $a_{l,m}$  being the matrix element connecting the legs  $l$  and  $m$  as defined in 2.3. The implementation of this recursive method is quite straightforward:

#### 4. Calculation

---



---

##### Algorithm 1: SkeletonCorrelationLayerByLayer

---

**Input** : An array  $[w_{i,j}], i,j=1,\dots,4$ , with each element being the float containing the weight from leg  $i$  to leg  $j$ , maximum time  $T$

**Output**: Skeleton correlation function for  $t=1,\dots,T$

```

1 firstLayer=[[1,0],[0,0]]
2 correlations=[firstLayer]
3 for  $T$  in range( $T$ ) do
4     | nextLayer=GetNextLayer(gate,  $T$ , correlations[-1])
      | correlations.append(nextLayer)
5 end for
6 return correlations

```

---



---

##### Algorithm 2: GetNextLayer

---

**Input** : An array  $[w_{i,j}], i,j=1,\dots,4$ , with each element being the float containing the weight from leg  $i$  to leg  $j$ , an array  $[c_{k,l}]$  containing last layers results, int  $T$

**Output**: Skeleton correlation function for  $t=T$

```

1 correlations=np.zeros([2T,2T])
2 for  $x$  in range(2T) do
3     | for  $y$  in range(2T) do
4         | out= $x\%2+2(y\%2)$ 
5         | for  $in$  in range(4) do
6             | correlations[ $x$ ][ $y$ ]+= $w_{in,out} \cdot c_{k',l'}$ 
7         | end for
8     | end for
9 end for
10 return correlations

```

---

By building up from the bottom to the top, this allows for an efficient calculation of the skeleton paths for entire layers at once. With a linear number of layers, quadratic number of sites in each layer, and constant cost per site independent of the position, this results in a complexity of  $\mathcal{O}(T^3)$  as shown in figure 4.1. The runtime is shown for different values of  $T$  for both the so far outlined "simple" solution, as well as the in section 4.4 explained improved version, including the preparation of parameters, but excluding the generation of the gate since that depends on the type of gate used. While for small  $T$ , the cubic nature is not quite visible yet, it later is clearly visible, especially in the magnified graph. While the runtime advantage over the exact contraction is quite visible already (about 1000s for 120 time steps), it is important to keep in mind the two other factors that increase this advantage even further. This simulation requires only a

minimal amount of memory, since no large vectors are constructed, and calculates the correlation for the entire light pyramid up to  $T$  in this time, instead of only a single point.

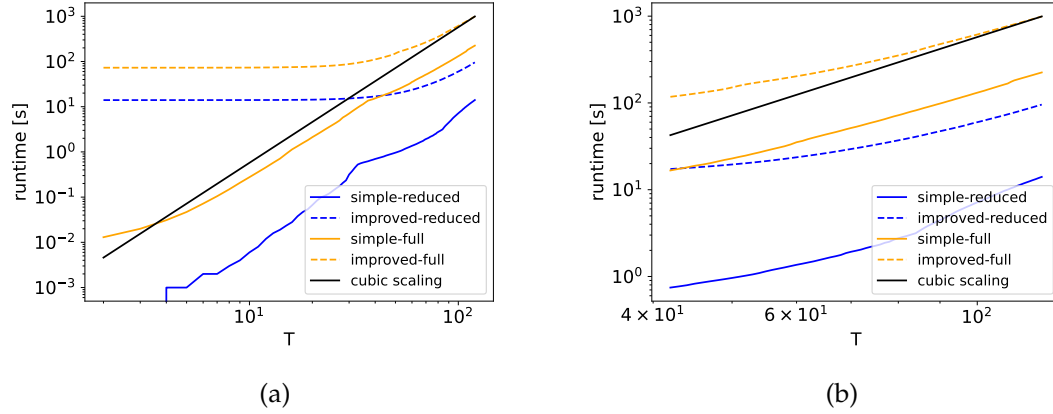


Figure 4.1.: Runtime growth with the number of timesteps for reduced gates (blue) and general gates (orange). The improved runtime (shown as dashed) lines show a significant initial cost due to the initial contractions. All 4 show an asymptotic cubic scaling for large time steps, visible in the magnified plot (b).

## 4.2. Generic Gates

In the case of Generic Gates things become more complicated, since 3 different operators can be applied at each position. However, the path sum interpretation can still be used by adjusting it: Instead of summing over simple paths, one needs to sum over all paths with different "colors" at each point in time, with the colors standing for one of the 3 spin operators. This increases the number of possible paths by a factor of  $3^T$ , and since the order of colors matters, the order expansion approach becomes inefficient. However, the layer-by-layer approach still works with a slight modification, instead of using 4 results from the previous layer, one now needs to sum over the 3 possible applied operators on the last 4 positions:

$$C_S(x, y, T)^{O_1, O_2} = \sum_{k'=0, O=1}^{k'=3, O=3} C_S(x', y', T)^{O', O_2} \cdot a_{k, k'}^{O_1, O}, \quad (4.6)$$

#### 4. Calculation

---

with  $a_{k,k'}^{O_1,O}$  being the element of the folded gate  $W$  with operator  $O_1$  applied on the top leg  $k$ , and operator  $O$  applied to the bottom leg  $k'$ . The implementation of this works very similarly to the reduced case. Before starting the calculation, one must preprocess the folded gate  $W$  by changing it into the  $3 \times 3$  matrices corresponding to one of the 3 operators applied on one input and output leg. After this step, one merely needs to adjust the dimensions of the entries and the multiplication with a matrix product:

---

#### Algorithm 3: SkeletonCorrelationLayerByLayerFull

---

**Input** : An array  $[w_{i,j}], i,j=1,\dots,4$ , with each element being the  $3 \times 3$  submatrix containing the weights from leg  $i$  to leg  $j$ , maximum time  $T$ , integer inputOp

**Output**: Skeleton correlation function for  $t=1,\dots,T$

```

1 firstLayer=[[0,0,0],[0,0,0]],[[0,0,0],[0,0,0]]
2 firstLayer[0,0,inputOp]=1
3 correlations=[firstLayer]
4 for T in range(T) do
5     | nextLayer=GetNextLayer(gate, T, correlations[-1])
6     | correlations.append(nextLayerFull)
7 end for
8 return correlations

```

---



---

#### Algorithm 4: GetNextLayerFull

---

**Input** : An array  $[w_{i,j}], i,j=1,\dots,4$ , with each element being the  $3 \times 3$  submatrix containing the weights from leg  $i$  to leg  $j$ , an array  $[c_{k,l}]$  containing last layers results as 3 element vectors, int  $T$

**Output**: Skeleton correlation function for  $t=T$

```

1 correlations=np.zeros([2T,2T])
2 for x in range(2T) do
3     | for y in range(2T) do
4         | out=x%2+2(y%2)
5         | for in in range(4) do
6             | correlations[x][y]+=win,out · ck,l
7         | end for
8     | end for
9 end for
10 return correlations

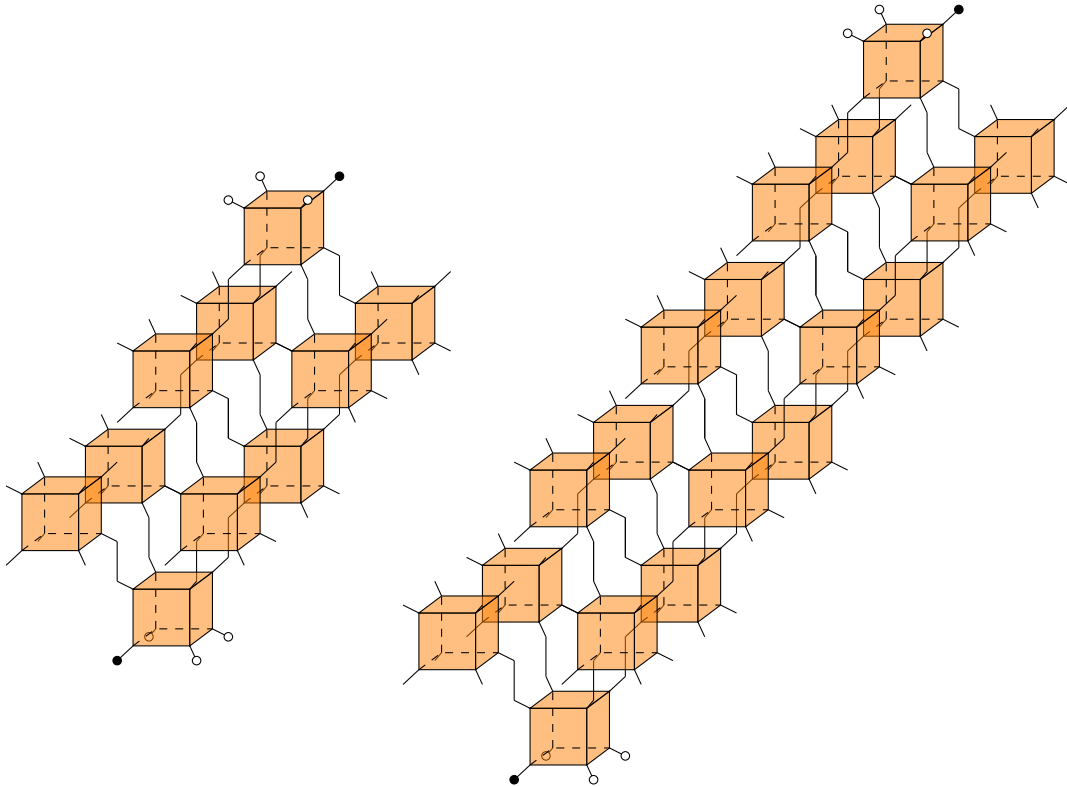
```

---

While this is in fact more computationally expensive, due to the matrix multiplications required, it only adjusts the prefactor of the cost, not the cubic scaling itself. The required memory remains negligible.

### 4.3. Exact solutions for benchmarking

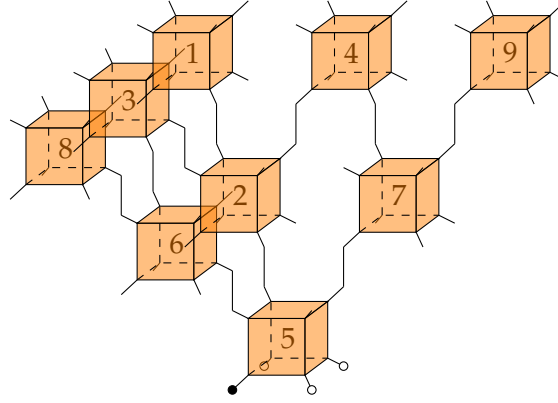
In order to numerically verify the approximation exact results are necessary. While these can be generated by contracting the folded network, the required time and memory grow exponentially, and the question of contraction order is complicated on its own. In order to reach later times we limit our coordinates to positions on a line parallel to the light ray, meaning that for each value of  $T$ , we chose  $x$  and  $y$  as  $T - 2 \cdot b$ ,  $b$  being the width of the largest layers in the resulting corridor. This leads to structures similar to the following:



(4.7)

These corridors have the advantage that the largest tensor size is constant for corridors of different lengths  $T$  but the same width  $b$ , and the runtime scales linearly with corridor length. With a number of  $\frac{1}{3}b(b+1)(2b+1) + b^2 \cdot (T-2b)$  gates (and a number of  $4T^2$  states applied to those gates, whose contraction is trivial so they are not of importance here), exhaustive search is not applicable, and even most greedy search algorithms fail to find efficient contraction orders for step numbers used in the comparisons of around 50 layers applied [33], resulting in multiple hundred gates. In order to contract these efficiently, one needs to deconstruct them into "pyramid double-faces", meaning two

connected faces of the light pyramid opposite of the growth direction of the corridor, resulting for  $b = 3$  in the following structure, rotated around for better visibility:



After contracting the first such layer, the next is constructed and contracted onto the last layer by starting at the middle top corner, and going outwards while prioritizing lower points, as indicated by the numbers. The reason for this rather unintuitive contraction order is, while one such layer has a fixed number of open legs, during the contraction of the next layer onto the last, this number can jump drastically if the wrong order is used. If simply contracting the next layer from the bottom up, the leg number temporarily increases by 7 for  $b = 3$ , leading to an increase of memory required by a factor of  $d^7$ . In this specific order however, the maximum tensor size is limited to the dimension of one such double face, specifically  $d^{b \cdot (2b-1)}$ . The actual network contractions were done by using a modified version of [23] and the `opt einsum` library [34].

#### 4.4. Weakness and Mitigation

As mentioned, the skeleton path approximation is based on the assumption, that the distance between curves is large. For the low density, unit strength case this is a given, but for the reverse case, it is assumed that at each given order in  $\epsilon$ , the number of constellations where the turns are close to each other is small compared to the number of cases where they are far apart, which are then again sufficiently suppressed. The terms omitted can now be categorized into different kinds of structures:

1. **low height** The first kind of structures, that expand into non-skeleton paths only for a smaller number of time steps, before recombining again. The width of these terms is naturally limited by the height, a kind of "sub-pyramid" of a smaller scale.

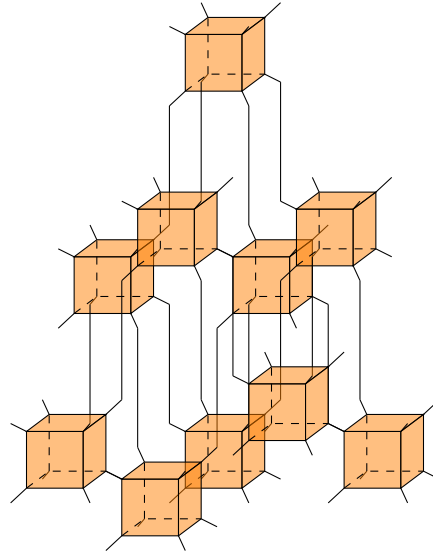
#### 4. Calculation

---

2. **low width, high height** The second kind of structure can follow a non-skeleton path for longer times, but never spreads far. It is essentially multiple paths parallel to each other, merging later in time again.
3. **high height, high width** The last kind is the most unhandy to calculate, large times and width require a complete contraction of the entire pyramid to account for these.

The last kind is impossible to account for without cost-intensive contractions, while the second kind follows a kind of corridor structure that was already used to get exact comparison values, which would allow for it to be calculated with limited amounts of memory, yet the number of terms would still lead to long computation times. However, both these kinds fortunately are exactly, what conditions 1&2 guarantee to suppress, long times of width larger than 1. This leaves the first type as the main problem, short "blips" in time where the paths widen, and then quickly recombine. Fortunately, there is a way to account for these, without having to contract the network for each position in the pyramid. To explain this, the concrete example of the smallest possible blip is chosen, with height  $T_s=3$ , meaning that the skeleton path widens, continues for one time step as a thickened path, and recombines again.

First, the weight for each of those parts of a path must be calculated. While for the skeleton paths the height of each part was 1, and the weights were simply the elements of the tensor, we now need to contract the  $T=3$  pyramid for all possible operators applied at the top and bottom, where the path can split up in between, but is only on a single leg at the top and bottom, resulting in the weights  $A_{k',k,x_t,y_t}$ , where  $k$  and  $k'$  describe the input and output leg and  $x_t, y_t$  describe the relative tensor position.



(4.9)



#### 4. Calculation

---

This introduces an initial cost of 80 (720 for the non-reduced gates, due to the 3 operators applicable at each position) contractions for  $T'=3$  networks, but does not affect the scaling of the algorithm. Furthermore, one now needs to redefine the recursive relation for the correlation at a point  $(x, y, T)$  as follows: This introduces an initial cost of 80 contractions for  $T'=3$  networks,

$$C_S(x, y, T) = \sum_{x_t=-1, y_t=-1}^{x_t=1, y_t=1} \sum_{k'=0}^{k'=3} C_S(x', y', T-3) \cdot A_{k', k, x_t, y_t} \quad (4.10)$$

This means, that one now needs the correlation function of the last and third last layer, to calculate the next one, resulting in a similar recursive simulation as before. When modifying the simulation, however, one needs to be careful, if the original part is kept, to not count the skeleton paths twice now. While the cubic runtime is conserved by this, the cost of each position is increased, since more past positions must be taken into account. To generate the correlation function at a time  $T$ , for a folded gate with leg dimension  $d$ , and including complex paths of height  $T_s$ , a resulting complexity for  $T$  of  $\mathcal{O}(d^2 \cdot T_s^2 \cdot T^3)$  is achieved, however with an exponential in  $T_s$  initial cost. For  $T_s = T$ , this recovers the complete contraction of the network with exponential in  $T$  cost.

## 5. Results

### 5.1. Reduced Gates

#### 5.1.1. Correlation for small times

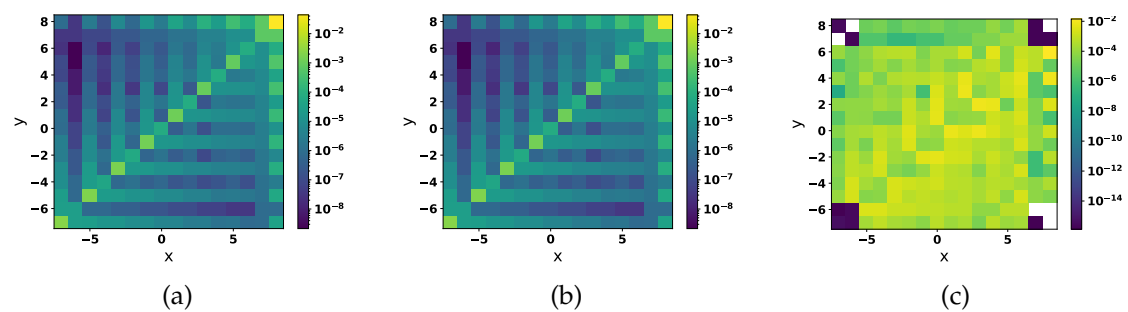


Figure 5.1.: Exact results (a), skeleton results (b), and relative error (c) of gate 1 for all top operator positions at a constant time  $T=8$  and a perturbation strength of  $\epsilon=0.01$ . The overall structure is completely preserved, with error barely reaching 1%.

As a first test, the correlation was computed for a full time layer for  $T=8$ , where exact results were still reachable. A full comparison between the skeleton and exact results for the reduced gate 1 with  $\epsilon=0.01$  are seen in figure 5.1, as well as the corresponding errors. The shape of the resulting correlation is remarkably similar to the approximated results, with errors of less than 1%. While the maximum correlation in this regime is conserved to the clearly visible light ray, the entire light pyramid displays a non-zero correlation, making it much more interesting than the pure ternary unitary case. To check for the complexity this kind of gate can produce, the same procedure was done for reduced gates 2 and 3 with the same perturbation strength in figure 5.2. While all gates share the strong light ray, the correlation in the rest of the light pyramid behaves very differently, further showing the interest of this subclass of gates. In particular for gate 3, the correlation is visible to be stronger along two of the other edges of the light pyramid than on the original light ray.

## 5. Results

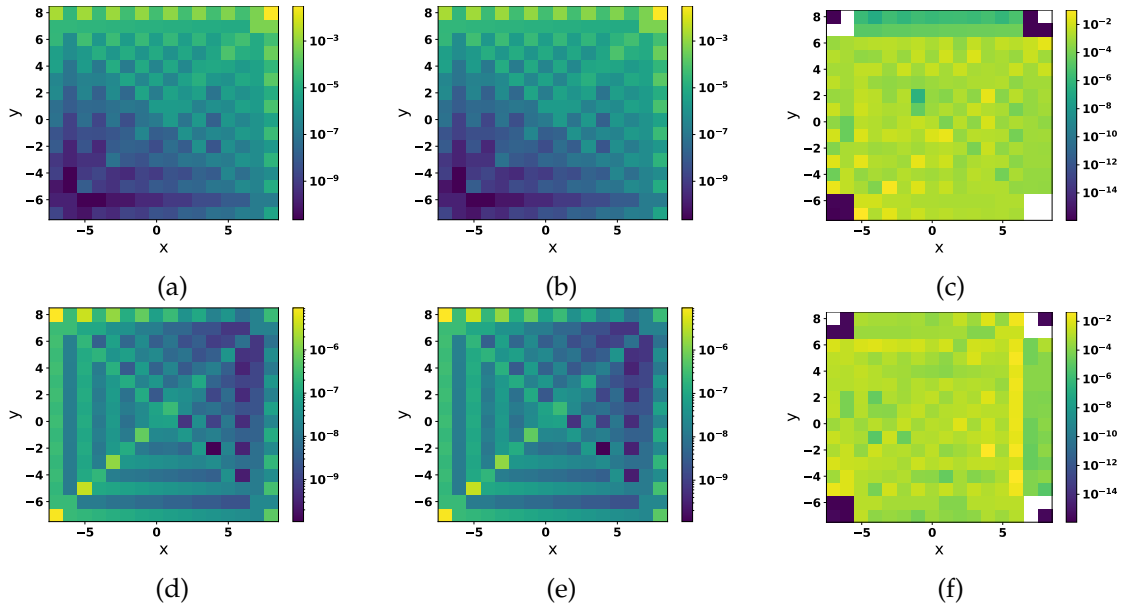


Figure 5.2.: Exact results (left), skeleton results (middle) and relative error (right) of gate 2 (top) and 3 (bottom) for all top operator positions for a constant time  $T=8$  and perturbation strength  $\epsilon = 0.01$ . Again, the correlation structure is conserved to a high detail for all tested gates. Additionally, the distribution is very different for the individual gates, showing how much more versatile the almost ternary unitary gate class is compared to the exactly ternary unitary one.

## 5.1.2. Behaviour for larger times

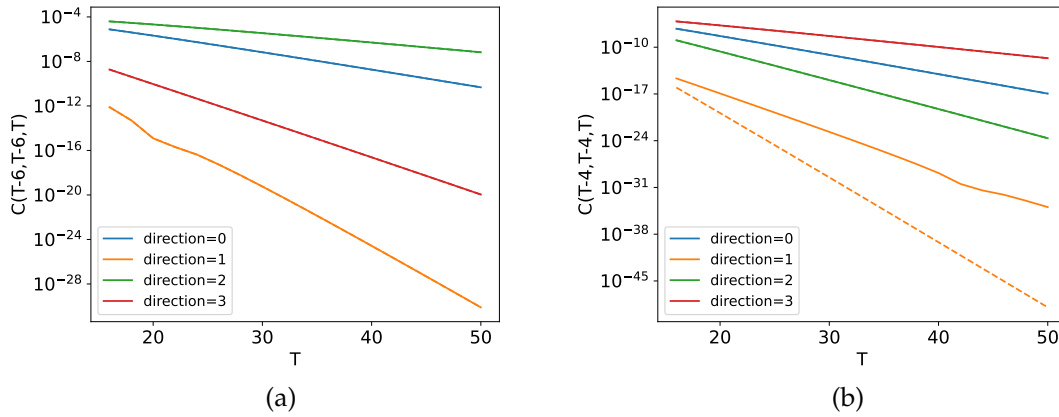


Figure 5.3.: Exact (solid curves) and skeleton (dotted curves) correlation results for gates 1&4 and a constant perturbation strength  $\epsilon = 0.01$ . The correlation is compared for various numbers of timesteps  $T$ , yet always with a constant distance to the lightray as indicated in the plots. The error of the approximation is not visible on the scale of the plot for gates fulfilling conditions 1&2, gate 4 however does not fulfill them and the error along direction 4 is clearly visible.

To check the validity of the approximation for larger times, both the skeleton correlation and exact solution were calculated along a parallel line to the light-ray for different times. The resulting correlations can be seen in figure 5.3. For each gate, the simulation was run in all four directions by rotating the gate after folding it, which corresponds to going into a different direction and applying it on the opposite leg of the first tensor applied.

For the different distances to the light ray available, the error between the exact and skeleton results is undetectable on the correlation plot, which is why the relative error is shown as well in figure 5.4. While the error behaves somewhat erratic for small times, it becomes more stable for larger times and larger distances. Especially interesting is the fact that for gate 1, for distances of 4 or 6, the error of 2 different directions converge, which was not seen in 1+1D. This is due to a key difference: For the propagation in one direction, two different propagator planes can be repeatedly applied, but also each plane can be applied in two different directions. If one such plane has a highly dominant eigenvector, yet still less relevant than the bare propagator, this main error source is shared for two different directions.

Gate 4 does not fulfill the requirements described in section 3.3, however, it was only broken along one direction. This is reproduced by the error in that direction being of the order of 1, the skeleton diagrams fail to approximate the results at all, however, the other directions aren't affected at all. This leads to the conclusion, that even when one of the conditions in section 3.3 is broken, it only affects the areas where the corresponding propagator layer is applied many times. It would be of interest, how the middle area of the light pyramid, where all 4 propagators are of equal relevance, behaves in this case. However, the memory and time required to get exact solutions in that area are massive and beyond the scope of this thesis.

## 5. Results

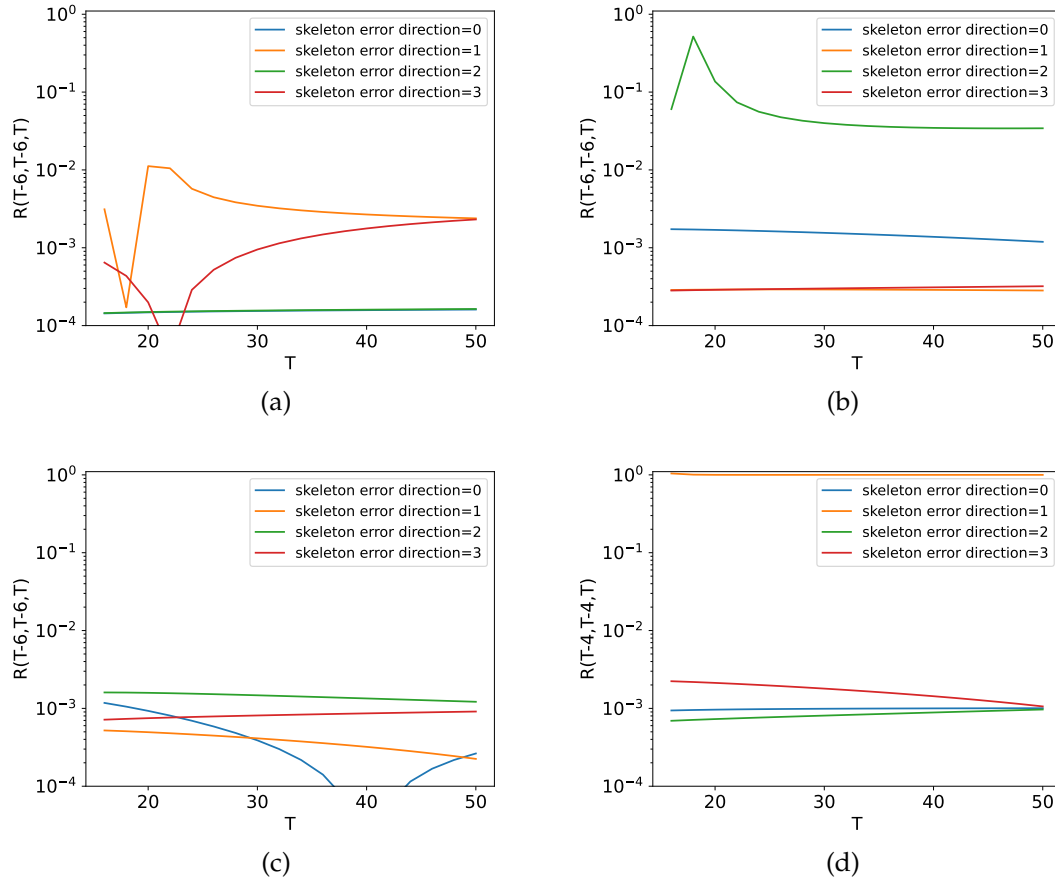


Figure 5.4.: Relative error between exact results and skeleton approximation for gates 1-4 and a constant perturbation strength  $\epsilon = 0.01$ , along a parallel to the light rays as with the distance indicated in the plots. The error stays approximately constant for different timesteps since the expected behavior of decaying error requires a higher distance from the edge of the light pyramid. For the non-suitable gate 4, the error is constantly of the order of one, allowing us to immediately identify gates not being suitable.

## 5.1.3. Perturbation strength

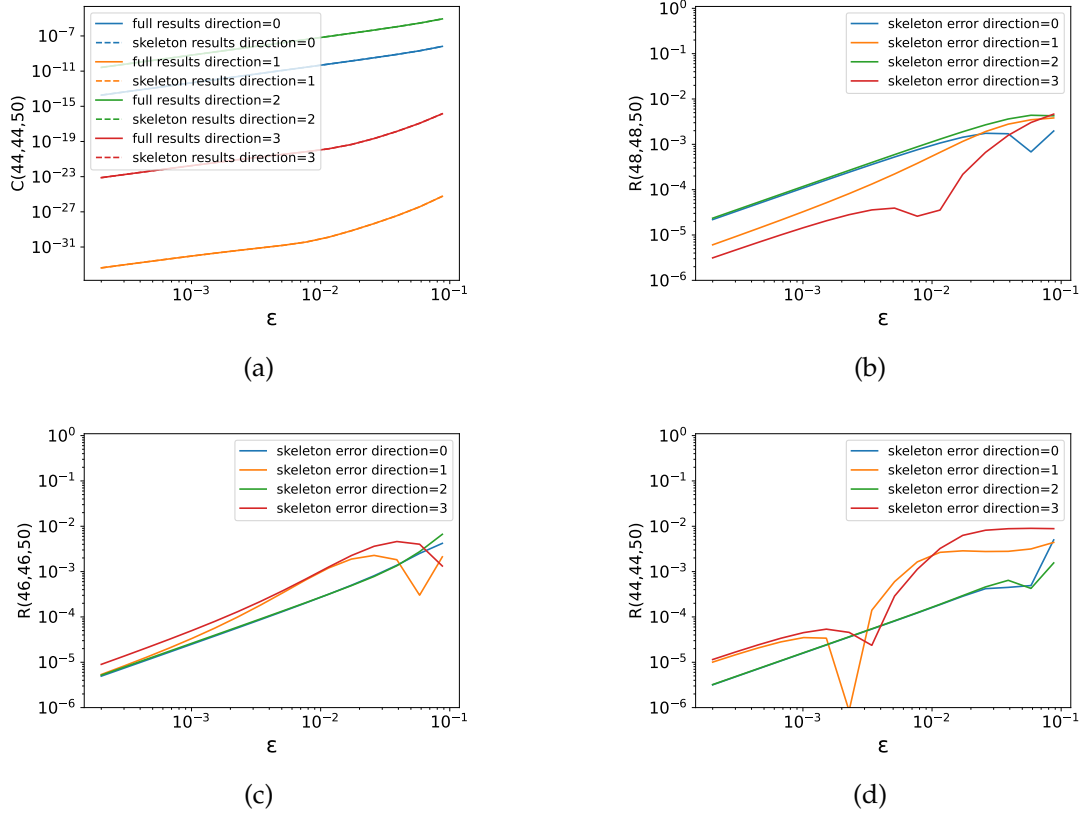


Figure 5.5.: Correlation for the reduced gate 1 at maximum reached distance (a) for varying perturbation strength  $\epsilon$ , and the relative error of the same gate for different distance (b-d). The approximation error is not visible on the scale of the Correlation plot, even for relatively large perturbation strengths. For  $\epsilon$  going to zero, the error also vanished as expected.

To analyze the behavior of the approximation for different strengths of the perturbation strength  $\epsilon$ , the correlation is shown for gate 1, as well as the relative error for different distances to the light ray. Again, in the correlation plots, the difference is not visible, even for relatively large perturbation strengths of  $\epsilon=0.1$ . Interestingly, even for small corridor widths, the relative error goes to zero for small perturbation strengths. There are significant spikes in the error, most likely due to the time not being high enough ( $T=50$ ) and certain relatively dominant complex paths canceling each other out for cer-

## 5. Results

tain perturbation strengths. The further one goes in time, the better the approximation works, however, the exponential decay leads to increasingly small correlations for very high times. To further prove the stability and convergence of the error for small  $\epsilon$ , the relative error for the maximum reached corridor width is shown in 5.6 for gates 2 and 3, as well as the correlation and error for gate 4. As expected, for the direction that does not fulfill the condition, the approximation fails to reproduce the results, and the error does not converge to zero for small  $\epsilon$ , reproducing the 1+1D expectation.

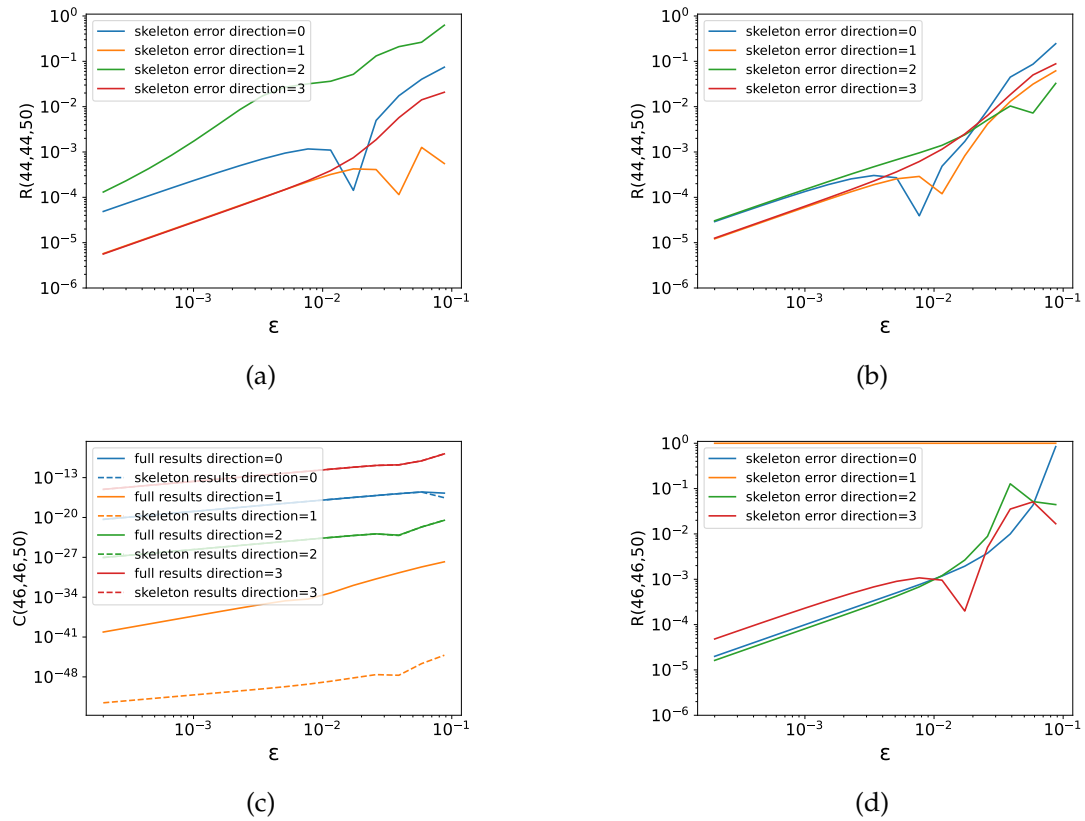


Figure 5.6.: Top: Relative error of the skeleton approximation for gates 2&3. As expected, the relative error goes to zero for small perturbation strengths. Bottom: Correlation (left) and relative error (right) for the non-suitable gate 4. The error is clearly visible on the scale of the correlation itself, leading to a constant relative error of one. Even for vanishing perturbation strengths, the error stays constant and does not vanish.



#### 5.1.4. Order Expansion

Since two different methods to calculate the skeleton correlation were developed, we now want to compare the second, less precise method of expanding in different orders of  $\epsilon$  and generating an analytical solution to this to the original result. Since the number of terms in each order grows exponentially, we only take into account third and fourth order terms, compared to the full skeleton result. Instead of comparing these to the exact solution, we compare them to the skeleton results, allowing us to investigate the behavior inside the entire light pyramid, instead of just the border areas. For this, 4 points were chosen for the same time  $T=50$  with even coordinates for both operators:

1.  $x = 44, y = 44$  The point already used in the last chapters, marking the largest corridor reached so far.
2.  $x = 40, y = 40$  Another point on the parallel to the light ray, however unreachable with exact contractions.
3.  $x = 0, y = 2$  A point off of any of the four main diagonals, close to the center for a maximum size of the pyramid, therefore leading to the highest number of orders possible.
4.  $x = -10, y = 24$  A random point, chosen far off the four main diagonals, yet still far enough inside the pyramid to offer high order terms.

For gate 1, the resulting relative error with respect to the full skeleton diagrams is shown in figure 5.7 for the different positions. Additionally, the individual graphs were fitted to check the resulting order of the error for small  $\epsilon$ . It is quite straightforward to see, that the resulting error order is dependent on the operator position, specifically being higher on the parallel to the light ray. This is due to the fact, that not all orders have terms on each position, due to the constraints of the number of turns/curves. Furthermore, the stability of the error is highly dependent on both the gate and the position of the operator. To understand this, one has to think about what a small perturbation  $\epsilon$  means here, or rather small compared to what. For each gate, there are 4 propagator weights, which are the parameters  $\epsilon$  must be small compared to, which results in good approximation for very small  $\epsilon$  in all cases. However, if one of the 4 propagator weights is significantly smaller than the other 3, an edge case can occur: In certain areas, where some or all of the other 3 weights are mainly applied, the third and fourth order approximation still works well, while in other places, where the low propagator weight is the relevant parameter, higher order terms become more relevant. Therefore, while the results for very small perturbations are still reliable, the numerical layer-by-layer method should be preferred, unless derivatives are needed, in which

case, due to the short runtime of both methods, a comparison with the full skeleton results should always be done as well.

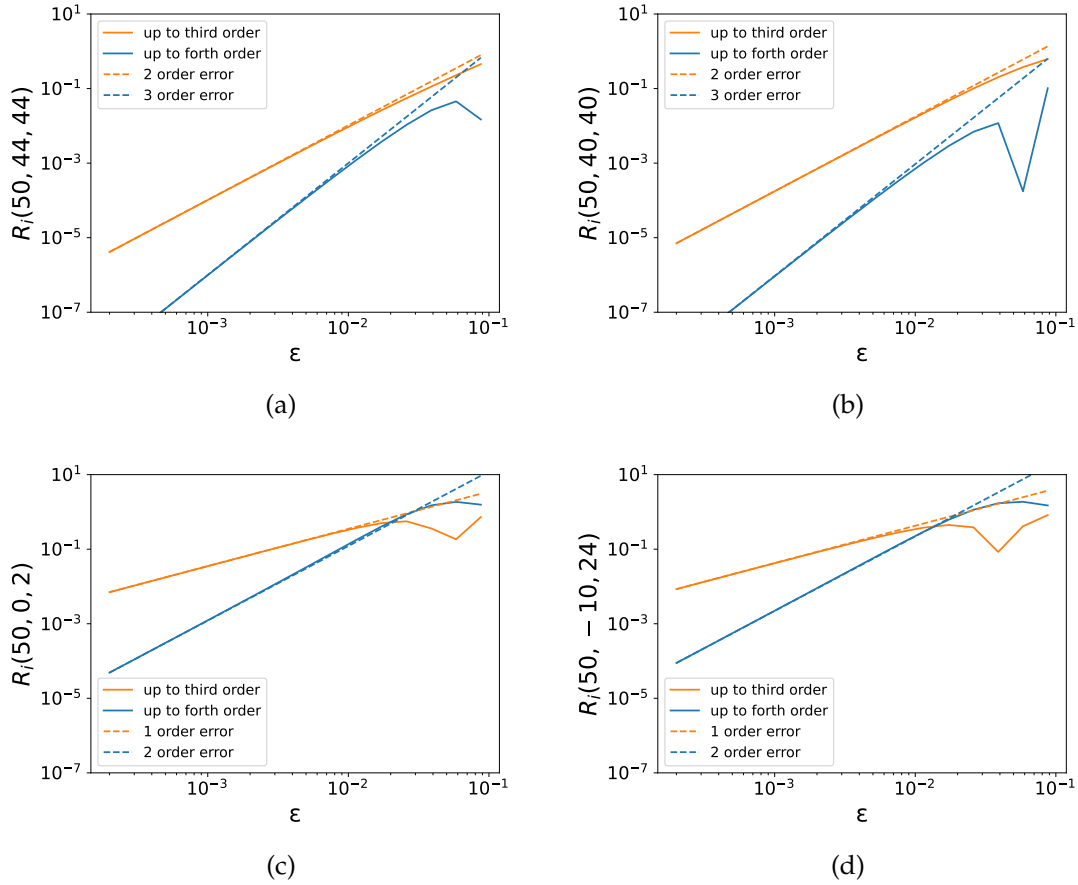


Figure 5.7.: Limited order skeleton results for gate 1. The individual colors show the relative order of the approximation up to third and fourth order, each being compared to a polynomial fit to show the order of the error for small  $\epsilon$ , which is position dependent. The chosen positions are: Within a small corridor used for the last sections (a), further inside the light pyramid where exact results aren't available (b), close to the middle of the pyramid off the main diagonal where the pyramid volume is maximal (c), far off the main diagonals (d).

### 5.1.5. Improved Results

As explained in section 4.4, the approximation for the unit density case is based on the assumption, that the distance between the perturbed gates is large. However, paths that split and then merge very quickly again, but stay skeleton paths for the rest of the time, are allowed. With the improved version, these kinds of paths can also be taken into account, without requiring large amounts of memory. To further investigate, how big the improvement of this is, the results for the improved algorithms with the same parameters as in section 5.1.3 are shown in figure 5.8. While the behavior is again a little erratic for higher perturbation strengths, it generally leads to an improvement of the relative error of up to one order of magnitude. Especially interesting is, that while the improved results can sometimes produce their own spikes of low errors, when the original skeleton results showed one, the improved version shows a similar spike, for lower  $\epsilon$ . This can sometimes lead to slightly higher errors for specific parameters, when the original results are in such a spike, while the improved ones have already left it. These results only take into consideration complex structures of height up to three time steps, which is still a very manageable amount of memory and computation time. However, it serves as a proof of concept, that this method allows to fine tune the simulation precision, according to what kind of resources and time are available. However, as seen in the result for gate 4, this slight improvement is not able to give good results for gates not fulfilling conditions 1&2, since those have dominant eigenvectors of support larger than 1, which require complex structures of constant width, but maximum height.

## 5. Results

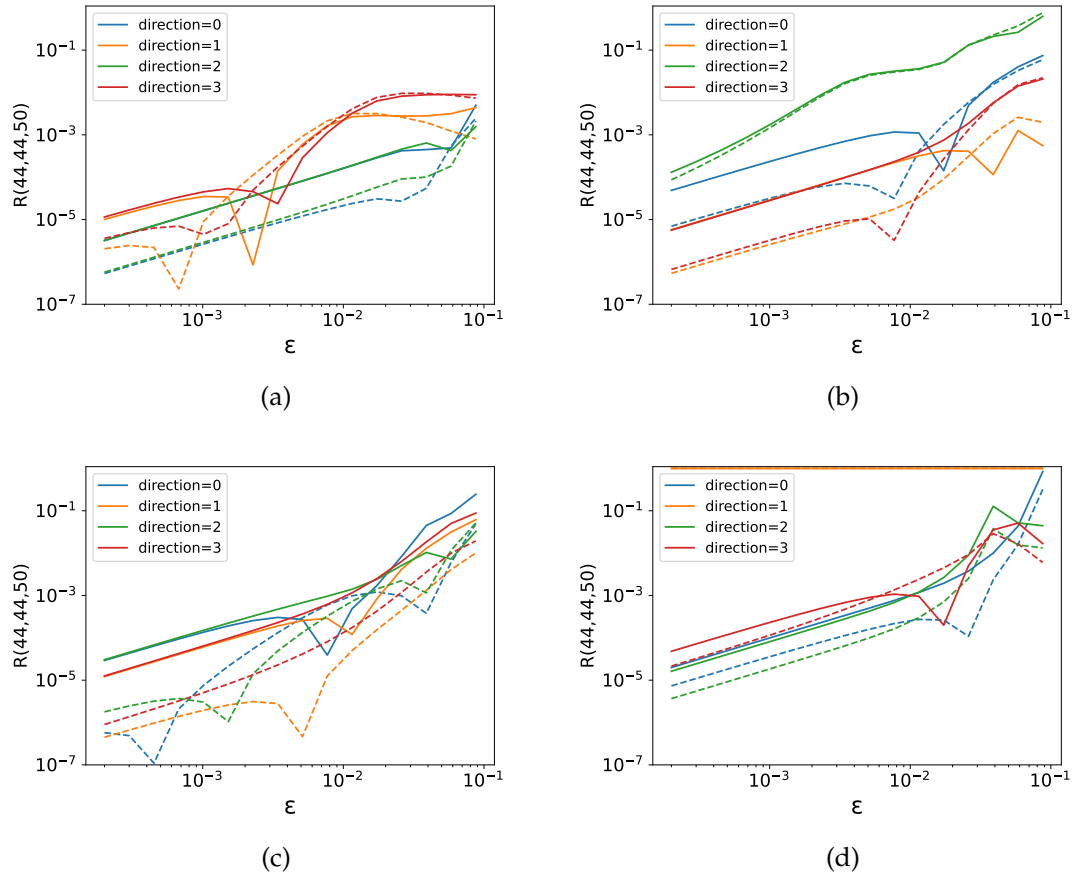


Figure 5.8.: Comparison of the original skeleton error (solid lines) and the improved error (dashed lines) for gates 1-4. The actual improvement ranges from almost none to a factor of 10, depending on the dominant error source: Either subleading eigenvalues can lead to large paths that contribute to the error, or small splits and merges can do the same, depending on the gate. Only the second kind is captured by the improved algorithm. Gate 4 shows that, if conditions 1&2 are not fulfilled, the improved algorithm does not capture the results either, as expected.

## 5.2. Generic Gates

### 5.2.1. Stacked gates

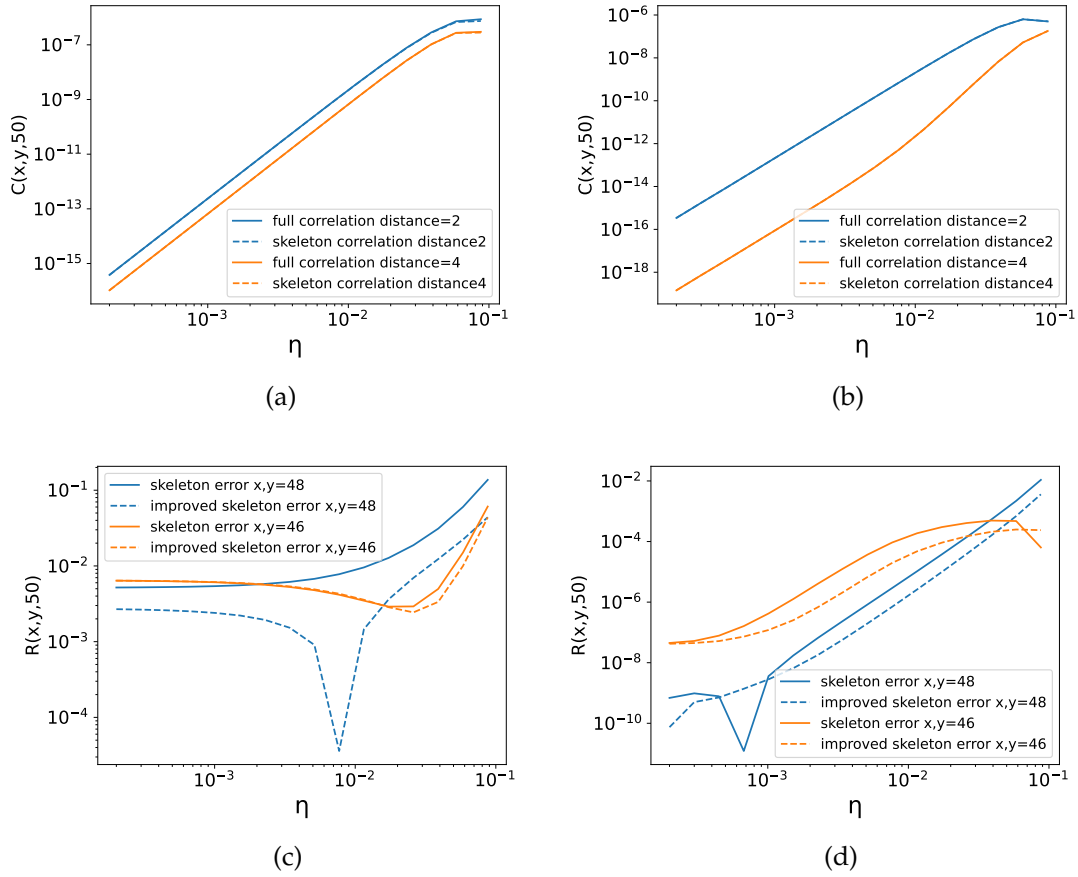


Figure 5.9.: Correlation (top) and relative error (bottom) for symmetrically stacked gates 5 (left) and 6 (right) as defined in A. For each, the colors correspond to different distances from the light cone. The error stays small even for relatively high values of  $\eta$ , especially gate 6, where the error is multiple orders of magnitude smaller than for the rest of the tested gates.

The described methods and improvements are now applied to the full case, without the randomized magnetic field. The first kind is the stacked structure, as described in section 2.4, and essentially implements nearest-neighbor interactions. The folded gate  $W$  has dimension 4 on each leg, resulting in significantly higher memory and calculation

requirements for the exact comparison results. For this reason, only corridors of width 2 and 3 are compared, for symmetric gates 5&6 with their parameters shown in A. The rotational symmetry of these gates allows us to only investigate one direction, with the results shown in figure 5.9. Two things become clear right away: The skeleton approximation is still able to reproduce highly accurate results, however, the change in relative error by using the improved version is significantly lower. This is likely due to a different error source: While the eigenvalues of support larger than 1 were significantly smaller than the bare propagator weight for reduced gates, short complex paths were by far the biggest source of error, whereas now, there are more eigenvalues almost as large as the bare propagator eigenvalue, which is also generally higher since it not simply the weight of the Sz state applied on each side, but the highest eigenvalue of the submatrix. While it may seem counter-intuitive, a higher propagator eigenvalue is not necessarily ideal. To understand this, one can take a look at the norm of the operator acting on states acting on two neighboring gates. This norm consists of two parallel lines ( $a^2$ ) and some additional term for an H-like structure (two parallel lines connected within the layer). The difference between this and the bare propagator  $a$  is maximized by maximizing  $a - a^2$ , namely for  $a=0.5$ . Therefore, the main error source in this case appears to be a large number of subleading eigenvectors, resulting in a relevant contribution. This error source however becomes smaller for large  $T$  and most importantly, larger corridor widths, which unfortunately aren't available for comparison. Additionally, the relative error for gate 6 is incredibly small. This is merely a result of the randomized gate generation, and should not be seen as the expected error range.

## 5.2.2. Crossed gates

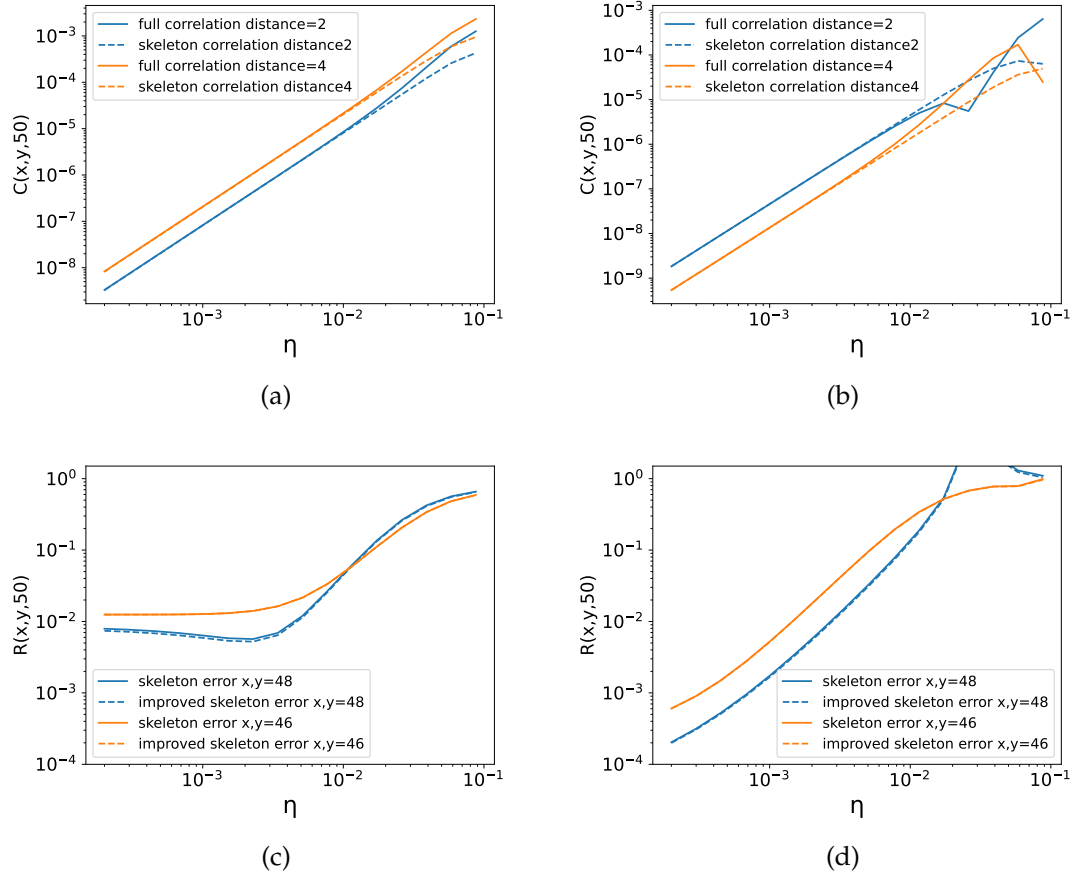


Figure 5.10.: Correlation (top) and relative error (bottom) for symmetric crossed gates 7 (left) and 8 (right) as defined in A. For each, the colors correspond to different distances from the light cone. While the error grows significantly faster, the skeleton approximation still manages to highly accurately approximate the correlation function for small  $\eta$ . The improved error is shown on the right as dashed lines, that are barely different from the original error, due to the absence of turn weights.

Applying the approximation to gates 7&8, generated by the crossed structure of section 2.4, which incorporates next nearest neighbor interactions instead, again with rotational symmetries used to reduce computational cost. The results are shown in figure 5.10,

and display a slightly different picture. While the error is still small for small  $\eta$ , it grows much quicker, which is simply a product of different split and merge weights compared to the example gates for the stacked structure. However, the improved method appears to give almost no reduction in error at all. This is due to the structure of the gate, which does not allow for curves at all, only straights and turns. These elements are however crucial for the small expansions and reductions the improved method incorporates, which leads to the conclusion that these are barely relevant for the error in this case, which is again dominated by the subleading eigenvalues. While the error is higher in these examples, the comparison still clearly shows that for both proposed structures, the skeleton approximation is able to approximate the exact results with high accuracy for small  $\eta$ , as intended.



## 6. Application outlook: Verification on a quantum computer

Now that the stability of perturbations around the ternary unitary point is verified, and the efficient calculation schemes are in place, we want to briefly touch upon possible applications on a quantum computer. While the size of NISQ era quantum computers is ever increasing [29], and numerous new kinds of quantum algorithms have been discovered, the difficulty of error correction and noise benchmarking is still ongoing [31]. While methods for one or two-qubit benchmarking, such as randomized benchmarking [17], are available, benchmarking of larger systems are usually based on implementing whole algorithms and verifying their results [31]. This leads to the potential application of the skeleton approximation for perturbed dual-/ternary-unitary gates: While we perturbed the ternary unitary attribute, the unitarity is never touched upon and allows for the direct implementation on a quantum computer, while the classical verification can be done efficiently. However, there is a number of problems to be addressed before one can implement it:

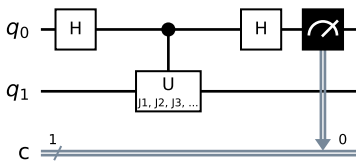
### Initial state preparation

This work was based on the infinite temperature initial state, or essentially tracing over the time direction after the time evolution. This non pure state can not be prepared by unitary operations alone, and therefore requires additional methods. In order to gain this initial state, ancillary qubits can be used. By applying Hadamard gates on each of the computational qubits  $q_i$ , and CNOT gates between the computational qubits  $q_i$  and the ancillary qubits  $q'_i$ , one can prepare the initial Bell state  $|\Theta^+\rangle = |0\rangle_{q_i} |0\rangle_{q'_i} + |1\rangle_{q_i} |1\rangle_{q'_i}$  [27]. By taking the partial trace over the ancillary system  $Tr_{q'_i}(|\Theta^+\rangle \langle\Theta^+|) = \mathbb{1}$ , we can prepare the main system in the maximally mixed state, as required. Alternatively, one could statistically sample this trace by preparing the system in random initial states, and averaging over the individual results.

### Measurement

Calculating the quantity  $\langle O_{x',y'}^1 | O_{x,y}^2(t) \rangle$  for any initial state is not an easy task, even less so for the infinite temperature one. The naive approach, of evolving the initial

state  $|\Psi\rangle$  forward in time before applying the second operator  $|\Psi'\rangle = O_{x,y}^2 U^t |\Psi\rangle$ , and doing the same with the same initial state, but applying the first operator before the time evolution  $|\Psi''\rangle = U^t O_{x,y}^2 |\Psi\rangle$ , one can calculate the contribution to the correlation function of this particular initial state by taking the overlap of the two evolved states, which in returns requires the full state tomography of each state vector. Instead, one can employ a Hadamard test [20] to estimate the the expectation value of a unitary  $U'$ . By defining the entire circuit, including the forward and backward time evolution and both operators, as  $U' = O_{x,y}^2 U^t O_{x',y'}^1 U^{t\dagger}$ , we can evaluate  $\text{Re}\langle\Psi|U'|\Psi\rangle$  by evaluating the quantum circuit



### Number of required qubits

The main issue of simulating correlation functions in 2+1D is the number of required qubit. For  $T$  timesteps, we require a number of  $T^2$  gates in the first time step, resulting in a required  $4 \cdot T^2$  main qubits, or  $8 \cdot T^2 + 1$  total qubits when including the ancillary qubits and the extra qubit for the Hadamard test. This means, that in order to even reach 10 timesteps, the required number of 801 qubits is not viable. However, the efficient numerical evaluation of section 4.2 can also be used in 1+1D systems, and significantly boosts the classical speed over the original exponentially expensive approach [18]. The required qubit number of  $2 \cdot T$  allows for the simulation of large enough number of timesteps for the skeleton error to be suppressed sufficiently for an error benchmarking of the quantum computer used. The additional advantage is, while both the general perturbed dual unitary in 1+1D and the stacked structure 2.1 in 2+1D systems only require nearest neighbour interactions, the interpretation of a 2-dimensional lattice as a string of qubits may require, depending on the interaction layout of the qubits, gates acting on far apart qubits.

## 7. Conclusion and outlook

First, let us summarize the results of this thesis: We started by recapitulating the basic ideas and results of ternary unitary gates and their, while efficiently solvable, rather simple correlation patterns. We transferred these gates and their conditions into the folded picture 2.9, and introduced the idea of ternary unitality 2.8. After covering the current state of generation for this class of gates, we introduced the basic ideas of [18] by introducing the interpretation of correlations functions as path sums, and focusing the discussion on paths with constant width of 1, so-called "skeleton paths". After classifying the individual elements of the folded gate into different "shapes", the condition of when the skeleton path approximation results in a bounded error had to be heavily modified: The shape of the operator in 2+1D is a planar network of gates, resulting in rapidly growing operator dimensions, which was solved by constructing the recursive conditions 1 (3.16) and 2 (3.27).

The next question was how to evaluate these skeleton diagrams in the more complex 2+1 dimensional setting. While in the 1+1D case, an analytic solution of the skeleton contribution for arbitrarily high orders of the perturbation was found for the reduced gates, the general case still required the contraction of an exponentially fast growing number of terms. We introduced a finite order analytical approximation of the skeleton diagrams 4.4 for the reduced gates, but most importantly developed a numeric algorithm to evaluate all orders of skeleton diagrams, for both the reduced and general setting, that scales polynomially with  $T$  and can be adjusted to arbitrary dimensions. This method allowed further improvements to the skeleton approximation, allowing us to account for short splittings of the paths. This new approximation method, while having a high initial computational cost, still scales polynomially with  $T$  and can be arbitrarily adjusted for higher precision at a higher runtime cost. We finally did the numerical simulations for several different gates and did an elaborate verification of the results by comparing them with exact contractions of the tensor network. In order to get these exact results, a network structure was identified that allows for a large number of timesteps while keeping the memory requirements constant, and the question of optimal network contraction order was discussed. Finally, a potential application of this model is briefly touched upon, as a way of verifying and benchmarking quantum computer results with this efficient evaluation of quantum time evolution.

While these results give a solid foundation for the usage of the new class of ternary unitary operators, a whole number of possible paths for further research have opened up. Firstly, one could focus on generalizing the method furthermore: So far, only a subset of ternary unitary gates was investigated, since no general description of all ternary unitary gates is found yet, therefore one further extension of the field would be to try to construct such a general parametrization from the parametrization of general unitary operators [12, 35]. Additionally, the used initial state of a thermally mixed state is, while useful for this work, rather limiting its use cases. The generalization to other solvable states described in [25] would increase the field of possible applications. The gates applied in this thesis were specifically kept as general as possible, and generated with randomized parameters to allow for an impartial testing of the method. However, since this has successfully been done, fine-tuning of the parameters could allow for the generation of gates where this method works particularly well, in order to use it to benchmark other methods that allow the computation of long-time correlation functions. The application to specific already studied Hamiltonians with special attributes, such as conserved charges [22] or certain types of transport [7], is also a necessary next step to verify, whether the method is able to capture the right physics behind such systems.

Additionally, the comparison to quantum computer results can be further expanded upon. While it was only briefly touched upon in this thesis, the results of the time evolution can also be computed on a physical quantum computer and compared. The required number of qubits is quite high for two-dimensional systems, however, the modification to one-dimensional systems is minuscule. This would allow us to verify larger quantum circuits with small classical computational power needed. Before doing the physical implementation, different noise models and their effect on the resulting correlation function could be modelled.

Finally, we categorized the error sources into 3 different categories: low height, low width, high height, and lastly high height, high width. While the last kind seems impossible to handle without contracting large networks, and the first kind is already accounted for in the improved algorithm, the second kind was barely touched upon, however, the algorithm can in fact be adjusted to incorporate these terms: By contracting tensor network of a certain height  $T'$ , with states acting on 2 bottom and 2 top sites, one can calculate the weights of such paths of width 2. By initially calculating all such weights in addition to the weights of small splittings that converge again within  $T'$ , the initial cost would be again increased, however by adjusting the recursive part one can take paths into consideration that have a width of 2 for arbitrarily long times, without having to contract any networks of height higher than  $T'$ . While the cubic runtime

## 7. *Conclusion and outlook*

---

would still be preserved, the prefactor and more importantly the initial contraction cost would be increased noticeably.

## A. Numerical gate parameters

In order to allow the verification of the results, the parameters used for the individual gates 1-8 are shown in the following. The parameters of the reduced gates 1-4 are split into the 4 individual dual unitaries, that make up the total ternary unitary. As mentioned earlier, there 4 one qubit gates between the two qubit gates combine to only three of them.

A. Numerical gate parameters

---

Table A.1.: Parameters of the dual unitary gates used to generate the suitable reduced gates 1-3 used in the numerical simulations. The other parameters are set to  $J_1 = J_2 = \frac{\pi}{2}$ .

Gate 1	1	2	3	4
sub-gatee 1, $\beta_i$	2.528048	3.284062	3.005824	3.489408
sub-gatee 1, $\gamma_i$	3.414195	4.780848	4.475981	3.893577
sub-gatee 2, $\beta_i$	2.097164	1.374768	0.413486	6.175558
sub-gatee 2, $\gamma_i$	0.803341	2.024007	0.445747	1.412271
sub-gatee 3, $\beta_i$	6.187363	0.180529	2.2097	2.393635
sub-gatee 3, $\gamma_i$	4.80136	5.898679	2.00867	2.717326
sub-gatee 4, $\beta_i$	0.431593	3.79281	4.998742	0.201113
sub-gatee 4, $\gamma_i$	2.861663	4.964801	6.211535	3.669157
$J_3$	2.704715	6.118913	2.170599	4.010173
Gate 2	1	2	3	4
sub-gate 1, $\beta_i$	2.465789	3.883337	2.588233	0.015487
sub-gate 1, $\gamma_i$	5.554538	5.560289	1.88753	3.704452
sub-gate 1, $\beta_i$	1.851934	1.809145	5.167708	3.934424
sub-gate 1, $\gamma_i$	0.694152	0.003323	5.919805	0.889075
sub-gate 1, $\beta_i$	2.692985	5.207199	4.510396	0.749123
sub-gate 1, $\gamma_i$	3.747192	0.815283	0.487161	5.222617
sub-gate 1, $\beta_i$	3.697565	4.860755	4.120799	3.504175
sub-gate 1, $\gamma_i$	1.119961	1.511631	3.179635	2.492827
$J_3$	1.050167	0.408881	5.465021	3.44303
Gate 3	1	2	3	4
sub-gate 1, $\beta_i$	3.400222	0.182299	4.610277	2.481544
sub-gate 1, $\gamma_i$	5.039411	1.598575	0.357419	5.445314
sub-gate 1, $\beta_i$	0.481686	5.298137	5.334042	6.103895
sub-gate 1, $\gamma_i$	2.421395	5.997226	2.800782	4.208004
sub-gate 1, $\beta_i$	1.64811	0.03223	3.413042	2.98826
sub-gate 1, $\gamma_i$	3.998454	6.146238	5.709296	5.71866
sub-gate 1, $\beta_i$	5.988128	2.588377	5.435063	4.223414
sub-gate 1, $\gamma_i$	3.950778	1.731387	5.634429	1.299937
$J_3$	4.185168	1.986093	1.872411	1.13672

A. Numerical gate parameters

---

Table A.2.: Parameters of the dual unitary gates used to generate the not suitable reduced gates 4 used in the numerical simulations. The other parameters are set to  $J_1 = J_2 = \frac{\pi}{2}$ .

Gate 4	1	2	3	4
sub-gate 1, $\beta_i$	5.603175	6.054872	2.409234	4.974555
sub-gate 1, $\gamma_i$	3.323145	3.569129	5.815695	0.446333
sub-gate 1, $\beta_i$	4.889303	5.466448	6.14884	5.021261
sub-gate 1, $\gamma_i$	2.89956	4.904209	0.74314	4.020742
sub-gate 1, $\beta_i$	2.605398	1.662252	4.864654	2.866077
sub-gate 1, $\gamma_i$	3.571576	0.11806	3.880718	3.845911
sub-gate 1, $\beta_i$	2.258855	2.745953	4.383346	0.378408
sub-gate 1, $\gamma_i$	4.189419	4.213742	1.321873	0.810068
$J_3$	2.749442	5.231505	3.27887	4.284003

Table A.3.: Parameters of the perturbed dual unitary gate used for the construction of the symmetric stacked gates 5&6 and symmetric crossed gates 7&8. The gate is used 4/2 times to construct one perturbed ternary unitary gate. The other parameters are set to  $J_1 = J_2 = \frac{\pi}{2} + \eta$ .

Gate	$\alpha_1$	$\beta_1$	$\gamma_2$	$\alpha_2$	$\beta_2$	$\gamma_1$	$J_3$
gate 5	5.610006	0.676427	3.738822	5.610006	0.676427	3.738822	0.261988
gate 6	4.846351	3.132212	1.412439	4.846351	3.132212	1.412439	4.704874
gate 7	5.610006	3.738822	3.328941	2.085891	3.738822	3.328941	0.676427
gate 8	0.479459	3.383469	3.148633	4.900374	3.383469	3.148633	6.144889





## B. Additional Order Analysis

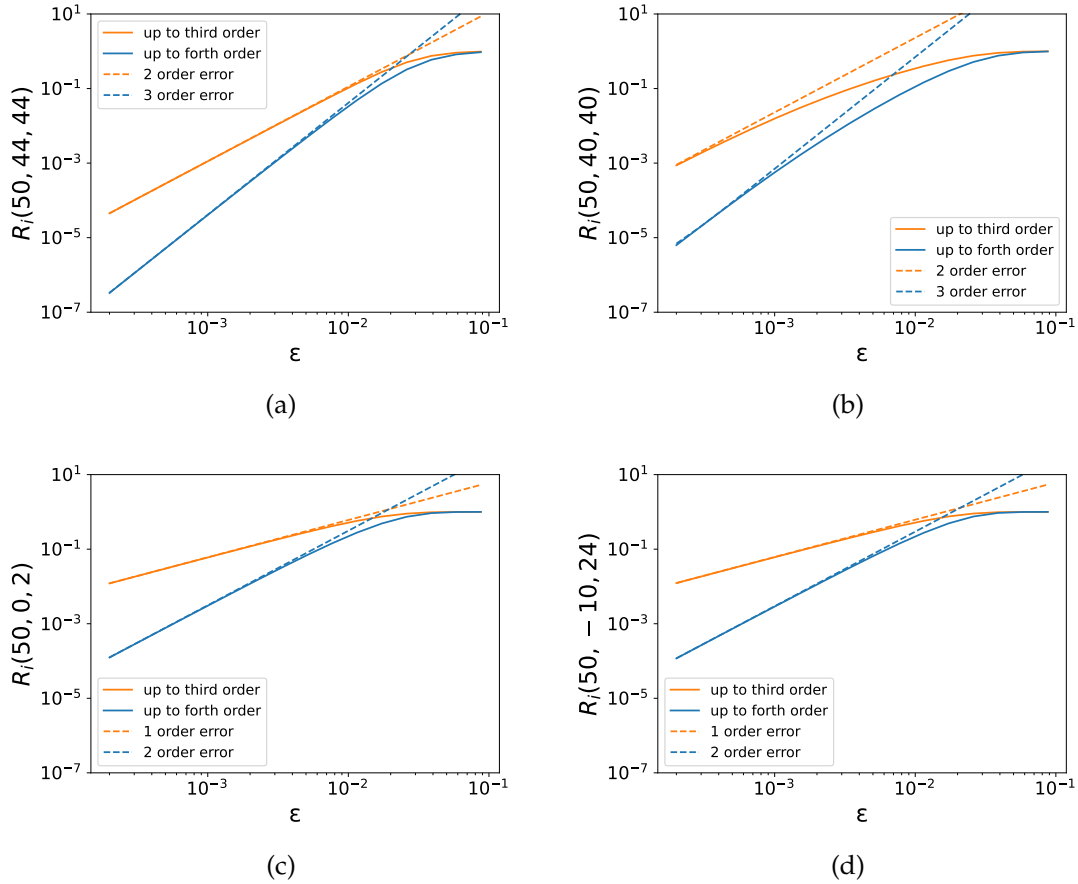


Figure B.1.: Limited order skeleton results for gate 2. The individual colours show the relative order of the approximation up to third and fourth order, each being compared to a polynomial fit to show the order of the error for small  $\epsilon$ , which is position dependant. The chosen positions are from left to right: within a small corridor used for the last sections, further inside the light pyramid, where exact results aren't available, close to the middle of the pyramid off the main diagonal, where the pyramid volume is maximal, far off the main the diagonals.

B. Additional Order Analysis

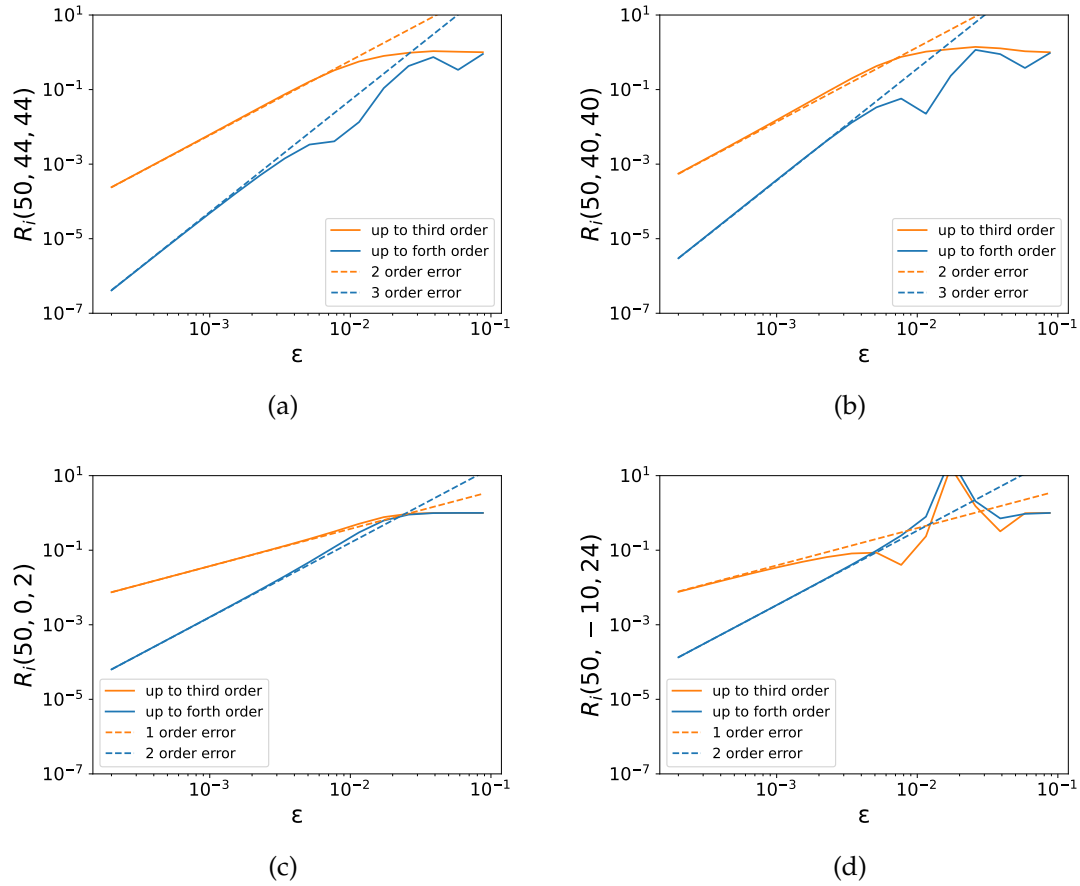


Figure B.2.: Limited order skeleton results for gate 3. The individual colours show the relative order of the approximation up to third and fourth order, each being compared to a polynomial fit to show the order of the error for small  $\epsilon$ , which is position dependent. The chosen positions are from left to right: within a small corridor used for the last sections, further inside the light pyramid, where exact results aren't available, close to the middle of the pyramid off the main diagonal, where the pyramid volume is maximal, far off the main the diagonals.

# Bibliography

- [1] A. Altland and B. D. Simons. *Condensed matter field theory*. Cambridge University Press, 2010.
- [2] P. Attard. “Chapter 11 Non-Equilibrium Computer Simulation Algorithms.” In: *Non-equilibrium Thermodynamics and Statistical Mechanics: Foundations and Applications*. Oxford University Press, Oct. 2012. ISBN: 9780199662760. DOI: 10.1093/acprof:oso/9780199662760.003.0011. eprint: [https://academic.oup.com/book/0/chapter/156339289/chapter-ag-pdf/44949154/book\\\_9437\\\_section\\\_156339289.ag.pdf](https://academic.oup.com/book/0/chapter/156339289/chapter-ag-pdf/44949154/book\_9437\_section\_156339289.ag.pdf).
- [3] M. C. Bañuls, M. B. Hastings, F. Verstraete, and J. I. Cirac. “Matrix Product States for Dynamical Simulation of Infinite Chains.” In: *Physical Review Letters* 102.24 (June 2009). DOI: 10.1103/physrevlett.102.240603.
- [4] B. Bertini, P. Kos, and T. Ž. Prosen. “Exact Correlation Functions for Dual-Unitary Lattice Models in 1 + 1 Dimensions.” In: *Phys. Rev. Lett.* 123 (21 Nov. 2019), p. 210601. DOI: 10.1103/PhysRevLett.123.210601.
- [5] B. Bertini, P. Kos, and T. Prosen. “Operator Entanglement in Local Quantum Circuits I: Chaotic Dual-Unitary Circuits.” In: *SciPost Phys.* 8 (2020), p. 067. DOI: 10.21468/SciPostPhys.8.4.067.
- [6] B. Bertini and L. Piroli. “Scrambling in random unitary circuits: Exact results.” In: *Phys. Rev. B* 102 (6 Aug. 2020), p. 064305. DOI: 10.1103/PhysRevB.102.064305.
- [7] A. Biella, M. Collura, D. Rossini, A. D. Luca, and L. Mazza. “Ballistic transport and boundary resistances in inhomogeneous quantum spin chains.” In: *Nature Communications* 10.1 (Oct. 2019). DOI: 10.1038/s41467-019-12784-4.
- [8] K. Binder. “Finite size effects on phase transitions.” In: *Ferroelectrics* 73.1 (1987), pp. 43–67. DOI: 10.1080/00150198708227908. eprint: <https://doi.org/10.1080/00150198708227908>.
- [9] P. W. Claeys and A. Lamacraft. “Ergodic and Nonergodic Dual-Unitary Quantum Circuits with Arbitrary Local Hilbert Space Dimension.” In: *Phys. Rev. Lett.* 126 (10 Mar. 2021), p. 100603. DOI: 10.1103/PhysRevLett.126.100603.

- [10] P. W. Claeys, A. Lamacraft, and J. Vicary. *From dual-unitary to biunitary: a 2-categorical model for exactly-solvable many-body quantum dynamics*. 2023. arXiv: 2302.07280 [quant-ph].
- [11] A. J. Daley, C. Kollath, U. Schollwöck, and G. Vidal. “Time-dependent density-matrix renormalization-group using adaptive effective Hilbert spaces.” In: *Journal of Statistical Mechanics: Theory and Experiment* 2004.04 (Apr. 2004), P04005. DOI: 10.1088/1742-5468/2004/04/P04005.
- [12] P. Dita. “Parametrisation of unitary matrices.” In: *Journal of Physics A: Mathematical and General* 15.11 (Nov. 1982), p. 3465. DOI: 10.1088/0305-4470/15/11/023.
- [13] I. Fuentes-Schuller and P. Barberis-Blostein. “A family of many-body models which are exactly solvable analytically.” In: *Journal of Physics A: Mathematical and Theoretical* 40.27 (June 2007), F601. DOI: 10.1088/1751-8113/40/27/F04.
- [14] C. Jonay, V. Khemani, and M. Ippoliti. “Triunitary quantum circuits.” In: *Phys. Rev. Res.* 3 (4 Oct. 2021), p. 043046. DOI: 10.1103/PhysRevResearch.3.043046.
- [15] Y. Kasim and T. Prosen. “Dual unitary circuits in random geometries.” In: *Journal of Physics A: Mathematical and Theoretical* 56.2 (Jan. 2023), p. 025003. DOI: 10.1088/1751-8121/acb1e0.
- [16] D. Kennes and C. Karrasch. “Extending the range of real time density matrix renormalization group simulations.” In: *Computer Physics Communications* 200 (2016), pp. 37–43. ISSN: 0010-4655. DOI: <https://doi.org/10.1016/j.cpc.2015.10.019>.
- [17] E. Knill, D. Leibfried, R. Reichle, J. Britton, R. B. Blakestad, J. D. Jost, C. Langer, R. Ozeri, S. Seidelin, and D. J. Wineland. “Randomized benchmarking of quantum gates.” In: *Physical Review A* 77.1 (Jan. 2008). DOI: 10.1103/physreva.77.012307.
- [18] P. Kos, B. Bertini, and T. ž. Prosen. “Correlations in Perturbed Dual-Unitary Circuits: Efficient Path-Integral Formula.” In: *Phys. Rev. X* 11 (1 Feb. 2021), p. 011022. DOI: 10.1103/PhysRevX.11.011022.
- [19] B. Kraus and J. I. Cirac. “Optimal creation of entanglement using a two-qubit gate.” In: *Phys. Rev. A* 63 (6 May 2001), p. 062309. DOI: 10.1103/PhysRevA.63.062309.
- [20] A. Luongo. *Quantum computing and quantum algorithms*. 2023. URL: <https://quantumalgorithms.org/chapter-intro.html#hadamard-test> (visited on 06/27/2023).
- [21] G. D. Mahan. *Condensed Matter in a Nutshell*. Princeton University Press, 2010.

- [22] K. Maruyoshi, T. Okuda, J. W. Pedersen, R. Suzuki, M. Yamazaki, and Y. Yoshida. “Conserved charges in the quantum simulation of integrable spin chains.” In: *Journal of Physics A: Mathematical and Theoretical* 56.16 (Mar. 2023), p. 165301. DOI: 10.1088/1751-8121/acc369.
- [23] C. B. Mendl and R. M. Milbradt. *Ternary Unitary Networks*. [https://github.com/cmendl/ternary\\_unitary\\_circuits](https://github.com/cmendl/ternary_unitary_circuits). 2021.
- [24] M. Mestyán, B. Pozsgay, and I. M. Wanless. *Multi-directional unitarity and maximal entanglement in spatially symmetric quantum states*. 2022. arXiv: 2210.13017 [quant-ph].
- [25] R. M. Milbradt, L. Scheller, C. Aßmus, and C. B. Mendl. “Ternary Unitary Quantum Lattice Models and Circuits in 2 + 1 Dimensions.” In: *Phys. Rev. Lett.* 130 (9 Mar. 2023), p. 090601. DOI: 10.1103/PhysRevLett.130.090601.
- [26] A. Müller-Hermes, J. I. Cirac, and M. C. Bañuls. “Tensor network techniques for the computation of dynamical observables in one-dimensional quantum spin systems.” In: *New Journal of Physics* 14.7 (July 2012), p. 075003. DOI: 10.1088/1367-2630/14/7/075003.
- [27] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010. DOI: 10.1017/CB09780511976667.
- [28] L. Piroli, B. Bertini, J. I. Cirac, and T. Ž. Prosen. “Exact dynamics in dual-unitary quantum circuits.” In: *Phys. Rev. B* 101 (9 Mar. 2020), p. 094304. DOI: 10.1103/PhysRevB.101.094304.
- [29] J. Preskill. “Quantum Computing in the NISQ era and beyond.” In: *Quantum* 2 (Aug. 2018), p. 79. ISSN: 2521-327X. DOI: 10.22331/q-2018-08-06-79.
- [30] T. Rakovszky, C. W. von Keyserlingk, and F. Pollmann. *Dissipation-assisted operator evolution method for capturing hydrodynamic transport*. 2020. arXiv: 2004.05177 [cond-mat.str-el].
- [31] S. Resch and U. R. Karpuzcu. “Benchmarking Quantum Computers and the Impact of Quantum Noise.” In: *ACM Comput. Surv.* 54.7 (July 2021). ISSN: 0360-0300. DOI: 10.1145/3464420.
- [32] A. Rupe, V. V. Vesselinov, and J. P. Crutchfield. “Nonequilibrium statistical mechanics and optimal prediction of partially-observed complex systems.” In: *New Journal of Physics* 24.10 (Oct. 2022), p. 103033. DOI: 10.1088/1367-2630/ac95b7.

- [33] F. Schindler and A. S. Jermyn. "Algorithms for tensor network contraction ordering." In: *Machine Learning: Science and Technology* 1.3 (July 2020), p. 035001. DOI: 10.1088/2632-2153/ab94c5.
- [34] D. Smith and J. Gray. "opt\_einsum - A Python package for optimizing contraction order for einsum-like expressions." In: *Journal of Open Source Software* 3 (June 2018), p. 753. DOI: 10.21105/joss.00753.
- [35] C. Spengler, M. Huber, and B. C. Hiesmayr. "A composite parameterization of unitary groups, density matrices and subspaces." In: *Journal of Physics A: Mathematical and Theoretical* 43.38 (Aug. 2010), p. 385306. DOI: 10.1088/1751-8113/43/38/385306.
- [36] R. Suzuki, K. Mitarai, and K. Fujii. "Computational power of one- and two-dimensional dual-unitary quantum circuits." In: *Quantum* 6 (Jan. 2022), p. 631. ISSN: 2521-327X. DOI: 10.22331/q-2022-01-24-631.
- [37] F. Vatan and C. Williams. "Optimal quantum circuits for general two-qubit gates." In: *Phys. Rev. A* 69 (3 Mar. 2004), p. 032315. DOI: 10.1103/PhysRevA.69.032315.