

Beyond Penrose tensor diagrams with the ZX calculus:

Applications to quantum computing, quantum machine learning,
condensed matter physics, and quantum gravity

Quanlong Wang

Richard D. P. East
Boldizsár Poór

Razin A. Shaikh
Bob Coecke

Lia Yeh

Overview. We introduce the *Spin-ZX calculus*, a diagrammatic language for $SU(2)$ representation theory obtained by embedding Penrose/Yutsis spin-network diagrams into the mixed-dimensional ZX calculus [13, 10, 15, 11]. The point is not only representational. The embedding places standard spin-network manipulations inside a compositional rewrite system, so identities and calculations can be derived by local diagrammatic rewrites with a clear denotational semantics [13, 3]. The paper develops this framework and demonstrates it in four settings: permutational quantum computing (PQC), AKLT/symmetry-protected topological (SPT) states, $SU(2)$ -equivariant quantum machine learning (QML) ansätze, and loop quantum gravity (LQG) volume calculations [6, 1, 2, 4, 12, 13].

The core construction is the representation of spin- j spaces as symmetrised subspaces of $(\mathbb{C}^2)^{\otimes 2j}$, together with explicit Spin-ZX generators for symmetrisers, couplings, and related structures [13]. This gives a single graphical language in which Clebsch-Gordan structure, Wigner matrices, invariant tensors, and spin Hamiltonians can be treated uniformly [13].

A rewrite calculus for spin networks. The foundational part of the paper formulates Spin-ZX as a fragment of the mixed-dimensional ZX calculus and fixes its semantics by explicit interpretation into finite-dimensional Hilbert spaces [13, 11]. In particular, spin-theoretic objects (symmetrisers, coupling tensors, invariant states) are represented as typed diagrams rather than informal tensor pictures [13]. This is what allows calculations to be carried out by a common set of local rewrite rules rather than by separate domain-specific symbolic manipulations.

A central ingredient is the explicit symmetriser diagram, which realises the spin- j irrep inside $2j$ spin- $\frac{1}{2}$ wires [13]. The paper derives a compact family of symmetriser identities—projector, invariance, capping/cupping, sliding, and looping rules—which serve as the basic rewrite library used throughout the applications [13]. These semantically nontrivial but diagrammatically simple identities are the main technical reason later examples can be treated in a uniform way.

Another generator of the Spin-ZX calculus is the Wigner $3j$ symbol, that is, the invariant tensor given by the renormalised, symmetry-optimised Choi state of the Clebsch-Gordan transformation connecting three $SU(2)$ irreducible representations. What makes the Spin-ZX calculus a more native representation than the similarly shaped Yutsis diagrams, is that the Clebsch-Gordan conditions arise intrinsically from the compositional action of the Z and X spiders in the mixed-dimensional ZX calculus, emerging as a natural and structural feature of the framework itself rather than as externally imposed constraints. No further conditions are needed to be externally imposed other than the diagram alone, to recover the faithful interpretation of Yutsis diagrams.

Application I: permutational quantum computing. For PQC, the paper shows how coupling-tree basis states can be expressed as Spin-ZX diagrams and how transition amplitudes become closed diagrams that evaluate to scalars under the rewrite system [6, 13]. This recasts standard recoupling calculations (usually phrased with Racah moves and $6j$ -symbol identities) in a compositional graphical language [8, 7, 13].

The QPL-relevant point is methodological. The framework does not only *draw* PQC amplitudes as tensor networks; it provides a typed rewrite calculus in which the recoupling logic itself is represented and simplified graphically. The paper’s low-qubit examples make this translation explicit and show how closed-diagram evaluation reproduces the expected amplitudes [13].

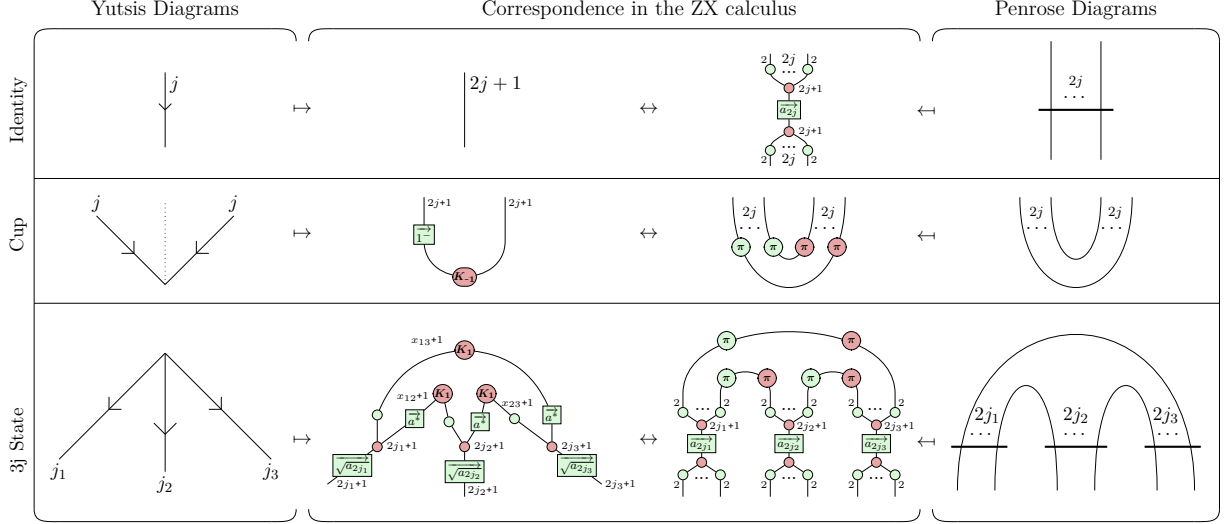


Figure 1: Generators of the Spin-ZX calculus, and their correspondence with Penrose and Yutsis diagrams from source paper Figure 1. The non-identity Spin-ZX diagram in the first row is the *symmetriser*, i.e. the projection onto the symmetric subspace invariant under permutation. The Wigner 3j symbol (bottom row) is the invariant tensor of the Clebsch-Gordan transformation.

Application II: AKLT/SPT states. The condensed-matter section revisits AKLT-state constructions in a spin-native graphical language [1, 2, 5, 13]. Compared with qubit-first diagrammatic treatments, this presentation makes the role of spin projectors and symmetrisers explicit and avoids repeated encoding/decoding overhead [5, 13].

The paper treats the 1D AKLT chain—spin-1 sites built from paired spin- $\frac{1}{2}$ systems with nearest-neighbour singlets—along with the 2D honeycomb AKLT setting presented in Figure 3 [2, 13]. The honeycomb construction is expressed and simplified in the Spin-ZX language to recover the expected graph structure. These derivations build on earlier qubit ZX calculus-based AKLT analyses [5], but now exhibit the elegance of working in a representation-theoretically native formalism able to accommodate spin- j irreps with wires of dimension $2j + 1$ [13]. This is a useful example of the same rewrite toolkit handling both circuit-like and many-body constructions.

Application III: $SU(2)$ -equivariant QML and diagrammatic gradients. The QML section studies the $SU(2)$ -equivariant two-qubit gate introduced in [4], which acts by a phase on the spin-0 sector in $\frac{1}{2} \otimes \frac{1}{2} \cong 0 \oplus 1$. Using the two-qubit Schur transform, the gate is represented compactly in Spin-ZX and used as a local building block in a brick-wall ansatz [4, 13].

The paper then shows that expectation values and parameter derivatives are expressible in the same graphical language [14, 13]. Notably, the gradient-variance object used in barren-plateau analyses is written as a single Spin-ZX diagram, connecting symmetry-aware QML ansätze with existing ZX-based differentiable-circuit methods [16, 14, 13]. This is one of the clearest QPL-facing aspects of the paper: the calculus supports not only state and process identities, but also derivative-level analysis.

Application IV: loop quantum gravity volume operator. The LQG section represents geometric operators on spin networks in the calculus [7, 9, 12, 13]. The area operator relates to $\sqrt{\vec{J}^2}$, while the more-involved volume operator depends on the triple product $\vec{J}_1 \cdot (\vec{J}_2 \times \vec{J}_3)$ [12, 13].

The paper isolates the core volume term and gives an explicit Spin-ZX construction using dimension splitters and controlled Pauli operations [13]. It then performs a minimal-volume eigenvalue/eigenvector calculation inside the graphical formalism, recovering the standard smallest nonzero volume (up to conventional choices of normalisation and absolute-value conventions) [13]. This is a substantive test of the formalism: the diagrams are used for an operator-eigenproblem computation, not only for rewriting state preparations or circuit fragments.

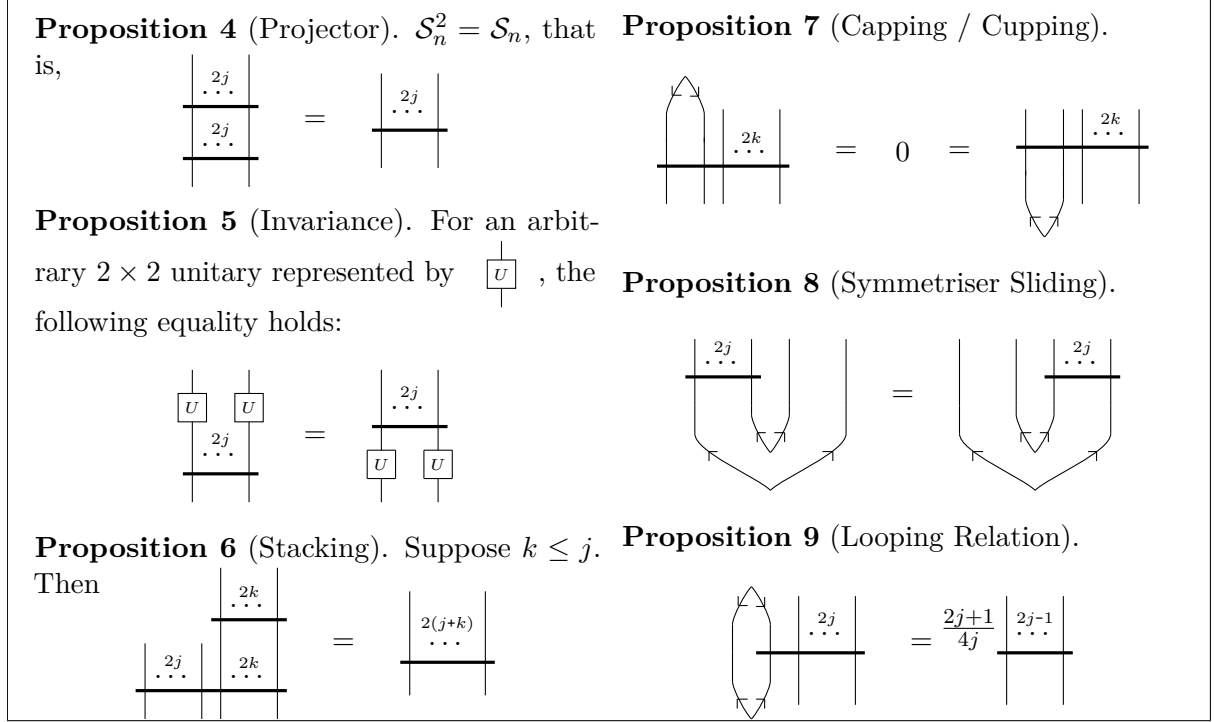


Figure 2: A compact rewrite “library” for symmetrisers from source paper Figure 2.

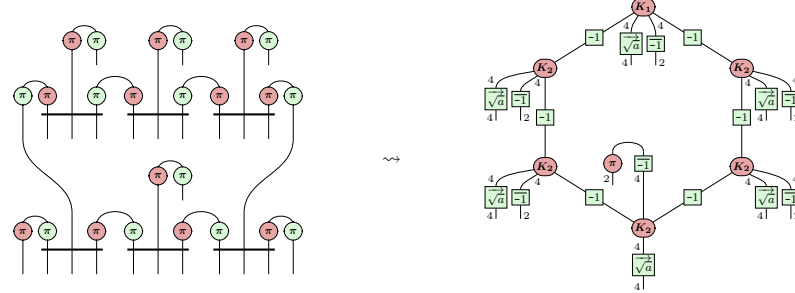


Figure 3: A feature of the diagrammatic representation of states is that it is simple to represent higher dimensional AKLT states which are formed from n -qubit symmetrisers acting on one end of n singlets projected to spin- $\frac{n}{2}$. Through Spin-ZX calculus rewrites transforming from Penrose- to Yutsis-style diagram, the representation is simplified to get the Honeycomb shape [13, Eq. 66].

Relevance for QPL and outlook. From a QPL perspective, the paper contributes a typed graphical language for a large and important class of symmetry-based calculations. The main technical ingredients are: (i) a semantic embedding into finite-dimensional quantum theory, (ii) a reusable rewrite theory centred on symmetrisers and spin couplings, and (iii) worked applications across quantum algorithms, many-body physics, QML, and quantum gravity [13]. The value is not only unification at the level of notation, but reuse of the same compositional proof style across domains that are usually treated separately. The completeness of the mixed-dimensional ZX calculus guarantees that all true equalities of Spin-ZX diagrams’ interpretations imply their provability wholly within the calculus. As such, it can be asserted for any equivalence of Penrose and Yutsis diagrams that the calculus’s tools suffice to prove so.

The paper also points to natural follow-up work: extensions beyond $SU(2)$, tighter connections to quantum-information constructions such as decoherence-free subspaces, and proof-theoretic questions about the Spin-ZX fragment itself (normal forms, automated rewriting, and completeness/faithfulness for useful sublanguages) [13, 11]. These are well aligned with QPL interests, since they concern both the expressivity and the algorithmic tractability of the graphical calculus.

Companion paper. This extended abstract summarises the results of [13].

References

- [1] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Physical Review Letters*, 59(7):799–802, 1987.
- [2] Ian Affleck, Tom Kennedy, Elliott H. Lieb, and Hal Tasaki. Valence bond ground states in isotropic quantum antiferromagnets. *Communications in Mathematical Physics*, 115(3):477–528, 1988.
- [3] Bob Coecke and Ross Duncan. Interacting quantum observables: Categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.
- [4] Richard D. P. East, Guillermo Alonso-Linaje, and Chae-Yeun Park. All you need is spin: $Su(2)$ equivariant variational quantum circuits based on spin networks. *arXiv preprint arXiv:2309.07250*, 2023.
- [5] Richard D. P. East, John van de Wetering, Nicholas Chancellor, and Adolfo G. Grushin. Aklt-states as zx-diagrams: Diagrammatic reasoning for quantum states. *PRX Quantum*, 3(1):010302, 2022.
- [6] Stephen P. Jordan. Permutational quantum computing. *Quantum Information and Computation*, 10(5):470–497, 2010.
- [7] Seth A. Major. A spin network primer. *American Journal of Physics*, 67(11):972–980, 1999.
- [8] Ilkka Mäkinen. Introduction to $su(2)$ recoupling theory and graphical methods for loop quantum gravity. *arXiv preprint arXiv:1910.06821*, 2019.
- [9] Pierre Martin-Dussaud. A primer of group theory for loop quantum gravity and spin-foams. *General Relativity and Gravitation*, 51(9):110, 2019.
- [10] Roger Penrose. Applications of negative dimensional tensors. In *Combinatorial Mathematics and its Applications*, pages 221–244. 1971.
- [11] Boldizsár Poór, Razin A. Shaikh, and Quanlong Wang. Zx-calculus is complete for finite-dimensional hilbert spaces. *arXiv preprint arXiv:2405.10896*, 2024.
- [12] Carlo Rovelli and Francesca Vidotto. *Covariant Loop Quantum Gravity: An Elementary Introduction to Quantum Gravity and Spinfoam Theory*. Cambridge University Press, Cambridge, 2014.
- [13] Quanlong Wang, Richard D. P. East, Razin A. Shaikh, Lia Yeh, Boldizsár Poór, and Bob Coecke. Beyond penrose tensor diagrams with the zx calculus: Applications to quantum computing, quantum machine learning, condensed matter physics, and quantum gravity. *arXiv preprint arXiv:2511.06012*, 2025.
- [14] Quanlong Wang, Richie Yeung, and Mark Koch. Differentiating and integrating zx diagrams with applications to quantum machine learning. *Quantum*, 8:1491, 2024.
- [15] Adolfas P. Yutsis, I. B. Levinson, and V. V. Vanagas. *Mathematical Apparatus of the Theory of Angular Momentum*. Israel Program for Scientific Translations, Jerusalem, 1962.
- [16] Chen Zhao and Xiao-Shan Gao. Analyzing the barren plateau phenomenon in training quantum neural networks with the zx-calculus. *Quantum*, 5:466, 2021.

Beyond Penrose tensor diagrams with the ZX calculus:

Applications to quantum computing, quantum machine learning, condensed matter
physics, and quantum gravity

Quanlong Wang¹ Richard D. P. East² Razin A. Shaikh^{1,3} Lia Yeh^{1,3,4}
Boldizsár Poór^{1,3} Bob Coecke^{1,5}

¹Quantinuum, Oxford, United Kingdom

²Haiqu Inc., San Francisco, USA

³University of Oxford, United Kingdom

⁴University of Cambridge, United Kingdom

⁵Perimeter Institute for Theoretical Physics, Waterloo, Canada

We introduce the *Spin-ZX calculus* as an elevation of Penrose’s diagrams and associated binor calculus to the level of a formal diagrammatic language. The power of doing so is illustrated by the variety of scientific areas we apply it to: permutational quantum computing, quantum machine learning, condensed matter physics, and quantum gravity. Respectively, we analyse permutational computing transition amplitudes, evaluate barren plateaus for $SU(2)$ symmetric ansätze, study properties of AKLT states, and derive the minimum quantised volume in loop quantum gravity.

Our starting point is the mixed-dimensional ZX calculus, a purely diagrammatic language that has been proven to be complete for finite-dimensional Hilbert spaces. That is, any equation that can be derived in the Hilbert space formalism, can also be derived in the mixed-dimensional ZX calculus. We embed the Spin-ZX calculus inside the mixed-dimensional ZX calculus, rendering it a quantum information flavoured diagrammatic language for the quantum theory of angular momentum, i.e. $SU(2)$ representation theory. We diagrammatically derive the fundamental spin coupling objects — such as Clebsch-Gordan coefficients, symmetrising mappings between qubits and spin spaces, and spin Hamiltonians — under this embedding.

Our results establish the Spin-ZX calculus as a powerful tool for representing and computing with $SU(2)$ systems graphically, offering new insights into foundational relationships and paving the way for new diagrammatic algorithms for theoretical physics.

1 Introduction

The quantum theory of angular momentum, familiar to mathematicians as the representation theory of $SU(2)$ and its Lie algebra, studies the quantum property of spins and their coupling. It is of great importance in a broad range of research areas such as quantum chemistry, condensed matter physics, particle physics, quantum computing and quantum gravity. However, calculations involving the coupling of spins quickly become unwieldy using algebraic methods. This has led to the development of diagrammatic methods to represent and evaluate complex expressions — containing symmetrisers and Clebsch-Gordan coefficients — in an intuitive way.

Graphical tensor notation was introduced by Roger Penrose mainly as a calculational tool for areas of physics that work with multilinear functions or tensors [42]. While Yutsis was

studying quantum chemistry, he developed a graphical notation for $SU(2)$, i.e. spin, to aid his calculations. These two roots later found a unification in Penrose’s binor calculus — a graphical tensors network notation designed for $SU(2)$. Due to their specificity, they remained a niche technical tools for many years.

Inspired by the works of Penrose, new diagrammatic languages were also being developed in computer science to represent information flows in programming language semantics [29, 26]. Their mathematical status was elucidated in the 90’s by Joyal and Street [31]. To unify these, in around 2004, Categorical Quantum Mechanics (CQM) aimed to provide a new formalism for quantum theory [1]. This led to the development of the ZX calculus [11], a specialisation of CQM, which is now widely used in quantum information and quantum computing.

ZX calculus [11, 12] offers a diagrammatic approach to reasoning about qubit quantum systems. Any quantum computation between qubits, or more generally, any linear map between qubits, can be represented as a ZX-diagram, a type of tensor network created from a small number of generating elements. The ZX calculus is a sound, universal, and complete language for qubit linear algebra [41, 54]. This means that any proof it derives is correct, it can represent any linear map, and we can prove any equality between linear maps using only the rules of the calculus, making it as powerful as the Hilbert space formalism. Recent works have extended the ZX calculus beyond qubits to higher dimensional quantum systems in both finite [44, 46, 56, 45] and infinite [51, 40, 8] dimensional Hilbert spaces.

This approach to quantum mechanics is philosophically different from the usual physicists’ analysis of quantum systems that has come before; it is born from the theoretical computer science perspective focusing on processes and their compositions, specifically formalised through applied category theory [2]. The calculus has found practical use in a number of areas across quantum information, with results in measurement-based quantum computation [17, 20, 34], topological quantum computation [15, 16, 25, 27], quantum error correction [53, 10, 28, 32, 47, 48], quantum chemistry [18, 50], and quantum circuit compilation and optimisation [5, 14, 19, 33].

In this paper we bring together Penrose’s binor calculus and ZX calculus, presenting a formal diagrammatic language for the analysis of spin. The extension of diagrammatic language techniques from qubits to spin has been investigated previously [21, 23, 24]. This paper can be seen as a completion of the work initiated there to provide a compositional framework for quantum algorithms beyond qubits, through the development of a calculus specific to physical symmetries, largely spin systems. The underlying aim of this research program is to create a bridge between quantum algorithms based upon qubits and qudits as an engineering resource, and algorithms based on the representation theory of spin as a fundamental physical symmetry. The concrete medium term ambition is the construction of a formal diagrammatic language for the full standard model. Given the model itself is founded on the representations of the three groups $U(1) \times SU(2) \times SU(3)$, this paper amounts to a formal diagrammatic language for the second of these.

1.1 Whistle-stop pictorial tour

Let us first give a details-light diagram-heavy guide to the paper that will be fleshed out in the main text. In this work, we introduce the Spin-ZX calculus — a diagrammatic formalism that unifies $SU(2)$ recoupling theory and quantum informational methods by embedding Penrose’s classical diagrams as a fragment of the mixed-dimensional ZX calculus. Mixed-dimensional ZX is made up of elements:

- *Z spider*,

$$\begin{array}{c} d_1 \quad \dots \quad d_n \\ \diagdown \quad \diagup \\ \boxed{\vec{a}} \\ \diagup \quad \diagdown \\ d_{n+1} \quad \dots \quad d_{n+m} \end{array} \xrightarrow{[\![\cdot]\!]} \sum_{j=0}^{\min\{d_i\}_i - 1} a_j |j, \dots, j\rangle \langle j, \dots, j|,$$

where $a_0 := 1$ and $\vec{a} = (a_1, \dots, a_{\min\{d_i\}_i - 1})$.

- *X spider*,

$$\begin{array}{c} n \\ \vdots \\ \text{---} \\ \text{---} \\ \vdots \\ m \end{array} \xrightarrow{[\![\cdot]\!]} \sum_{\substack{i_1 + \dots + i_m + j \\ \equiv j_1 + \dots + j_n \pmod{d}}} |i_1, \dots, i_m\rangle \langle j_1, \dots, j_n|.$$

where $j \in \mathbb{Z}$. Two X spiders labelled j and j' are equal by definition if $j \equiv j' \pmod{d}$.

We use these to construct the generators of Spin-ZX calculus which, as explained further below, are composed of Yutsis diagrams [58] which can be seen as spin-invariant components that can be composed to form Penrose's spin networks [42].

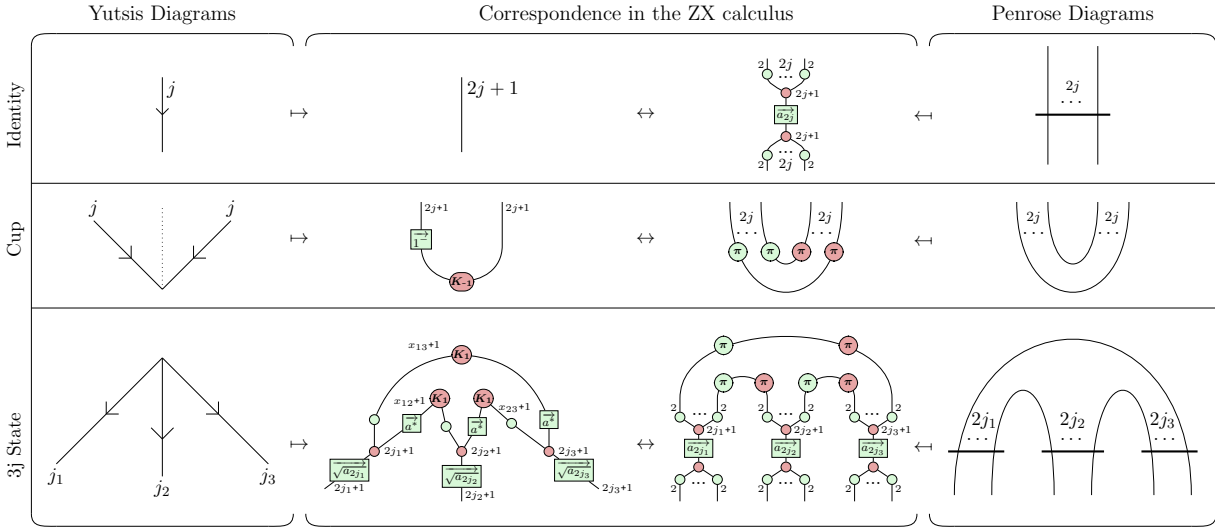


Figure 1: Generators of the Spin-ZX calculus. Here, $j, j_1, j_2, j_3 \in \mathbb{N}/2$, $x_{lm} = j_l + j_m - j_n$, where $\vec{a}_n = \left(\frac{1}{\binom{n}{1}}, \dots, \frac{1}{\binom{n}{k}}, \dots, \frac{1}{\binom{n}{n}}\right)$, $\vec{1} = ((-1)^1, \dots, (-1)^{2j})$, and $\frac{\vec{1}}{a} = \left((-1)^1 \binom{x_{k\ell}}{1}, \dots, (-1)^{x_{k\ell}} \binom{x_{k\ell}}{x_{k\ell}}\right)$.

In Figure 1, we can see the following:

- The *Yutsis diagram* (on the left);
- The *Penrose diagram* (on the right); and
- The *ZX diagrams* (in the middle), which are their translations into our modern diagrammatic language.

In the left half of Figure 1, each row has the following *interpretation* in terms of standard SU(2) representation theory:

- The *identity* operator on the spin- j space is $\sum_m |j; m\rangle \langle j; m|$;
- The *cup*, which represents the map from an input to an output wire, or a spin- j Hilbert space to its dual, is $\sqrt{2j+1} \sum_m (-1)^{j+m+1} \langle j; m, j; -m|$; and
- The *3j-state*, which is equivalent¹ to the Clebsch-Gordan map that tells us how two spins combine, is $\sum_{m_1, m_2, m_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} |j_1; m_1, j_2; m_2, j_3; m_3\rangle$.

The ZX diagrams faithfully satisfy the correspondence between the left and right halves of Figure 1, by natively representing the following isometry map to the symmetrised subspace of $(\mathbb{C}^2)^{\otimes n}$ which defines this correspondence:

Composing this isometry and its adjoint yields the projection onto the symmetrised subspace, called the *symmetriser*:

On the left, we can see Penrose's notation for the symmetriser, which builds a spin- j space by symmetrising the $2j$ spin- $\frac{1}{2}$ wires. On the right, we have the ZX calculus representation of the symmetriser. This enables us to exactly represent the decomposition of higher spins into the symmetric subspace of multiple copies of spin- $\frac{1}{2}$.

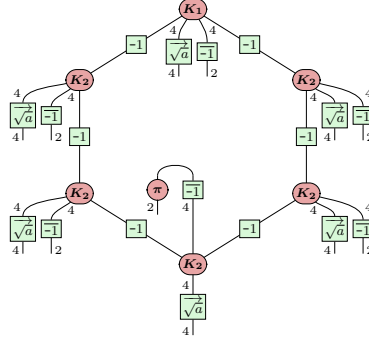
As a diagrammatic calculus, it also comes with a collection of rewrite rules given in Section 3.

With the calculus outlined we apply it in a number of different settings. In quantum computing theory we tackle the permutational quantum computing (PQC) model:

¹This equivalence is up to the 3j-state being symmetric with respect to the spins. Rather than a map from two of the spins to a third, it represents how all three spins combine to spin-0.

We represent the relevant basis states of binary trees of coupling spins and show how ZX rewrites allow us to derive expectation values of permutation operators directly from the structure of the network.

In condensed matter physics, we tackle symmetry-protected topological phases in 1D and 2D greatly simplifying previous diagrammatic computations on these systems Ref.[24]. For instance, the diagram



depicts the 2D AKLT Hexagon state.

We will show how the unitary spin-network gates of Ref.[23] can be compactly written in the form:

$$\boxed{V(\theta)} = \boxed{S_2} \boxed{P_2(\theta)} \boxed{S_2^\dagger} = \text{diagram with } \Lambda \text{ box and } \theta \text{ circle}$$

where the Λ box is shorthand explained in the text below. These diagrams are then integrated into previously derived diagrammatic QML calculations for use in gradient computations.

Finally, we will turn to Loop quantum gravity where we will diagrammatically calculate the minimal volume of a chunk of discretised space as $\frac{-\sqrt{3}}{4}$:

$$\text{Diagram 1} = \frac{-\sqrt{3}}{4} \text{Diagram 2} \quad (1)$$

Diagram 1 is a complex network of nodes and edges. It includes nodes labeled K_1 , K_{-1} , and K_{-2} . There are green boxes with values like $-1, -1, i\sqrt{3}$ and $-1, 0$. Edges are labeled with values like $\frac{i}{8}$, $\frac{1-i}{4\sqrt{3}}$, and $\frac{1-i\sqrt{3}}{4}$. Diagram 2 is a simplified version of Diagram 1, showing the same structure but with a different set of labels and values.

1.2 Paper structure

In concrete terms what is presented in this paper is as follows. We provide an introduction to how mixed-dimensional ZX calculus can be extended to allow us to focus on $SU(2)$ representations rather than qubits. We then explicitly introduce the Spin-ZX calculus with its generators and rewrite rules showing diagrammatically how the symmetrisation operator allows us to define different representations and connect them to qubits, detailing its rewrite properties and demonstrating diagrammatically that it has the expected behaviour. We then show how the diagrammatic description of spin-coupling via 3jm symbols or Clebsch-Gordan coefficients

(these being isomorphic, with the first being invariant states and the second being the equivalent equivariant maps) can be formulated as mixed-dim ZX diagrams embedding them in the broader calculus. We then use the symmetriser-based rewrites to derive all the properties of these symbols diagrammatically. Given the embedding into mixed-dimensional ZX, we show how this allows us to formally express related mathematical objects in the algebra of $\mathfrak{su}(2)$ in a form that is related to but distinct from the Spin-ZX calculus itself.

To showcase the broad range of applications for these diagrams and to offer a starting point to other researchers, we then apply these techniques to a number of areas. Firstly, we start in the realm of quantum computing itself and derive permutational quantum computing (PQC) transition amplitudes entirely diagrammatically. This turns calculations usually requiring the use of more specialist Racah transformations, into the (diagrammatic) reduction of a tensor network. Next, we look to condensed matter physics, where we simplify and extend more cumbersome diagrammatic calculations for the analysis of 1D and 2D AKLT states previously attempted with qubit-only calculi [24]. Following this, we turn to calculations in quantum machine learning, demonstrating how parametrised quantum circuits designed for spin systems can be analysed, and providing as an example how to calculate the variance of the $SU(2)$ equivariant ansatz seen in Ref. [22]. Finally, we apply our techniques to quantum gravity, demonstrating the ability to simplify the diagrammatic representation of the volume operator of loop quantum gravity, and computing the minimal quantised volume via an eigenvalue calculation performed entirely diagrammatically which simplifies and extends the results seen in Ref. [21].

The general principle is that the Spin-ZX calculus streamlines the representation and manipulation of $SU(2)$ calculations, central to various domains including permutational quantum computing, loop quantum gravity, quantum machine learning, and condensed matter physics. This not only elucidates these calculations but gives a new perspective on the core relationships within $SU(2)$ representation theory and offers a formal diagrammatic language with which to analyse and apply them.

2 ZX calculus for finite-dimensional quantum theory

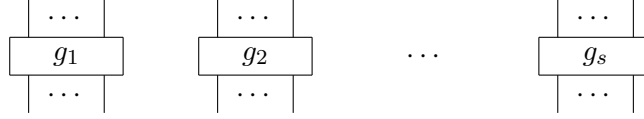
In this section, we give an introduction to the general notion of a graphical calculus, and specifically, outline the parts of the mixed-dimensional ZX calculus [45] that we will need to outline the fragment that corresponds to the Spin-ZX calculus, which can be used to streamline the representation and manipulation of $SU(2)$ calculations.

2.1 A general graphical calculus

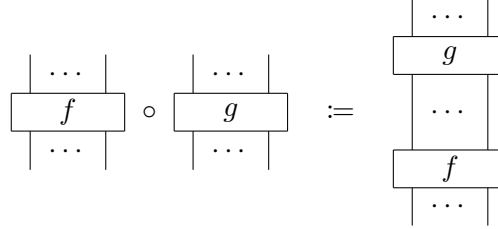
A graphical calculus is a mathematical language for expressing and reasoning about processes using graphical representations² (i.e. diagrams) whilst maintaining the rigour of an explicit interpretation. By working with a graphical representation, we obtain higher expressivity, enabling a cleaner presentation of the underlying structure than if we were to rely solely on conventionally used symbolic notation. This section explains the basic idea of a general graphical calculus; for more details, a thorough introduction to the topic is presented in [13].

²Note here we are not using representation in the manner of representation theory, unfortunately, these two meanings do appear in this paper though context should indicate which is meant.

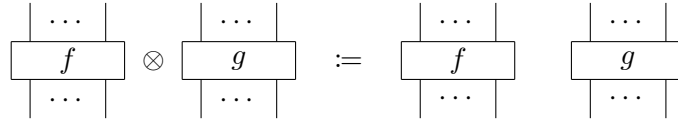
A graphical calculus uses diagrams composed of boxes and wires, which can be manipulated according to a set of equalities called graphical rewrite rules. The basic building blocks of a graphical language are its *generators*:



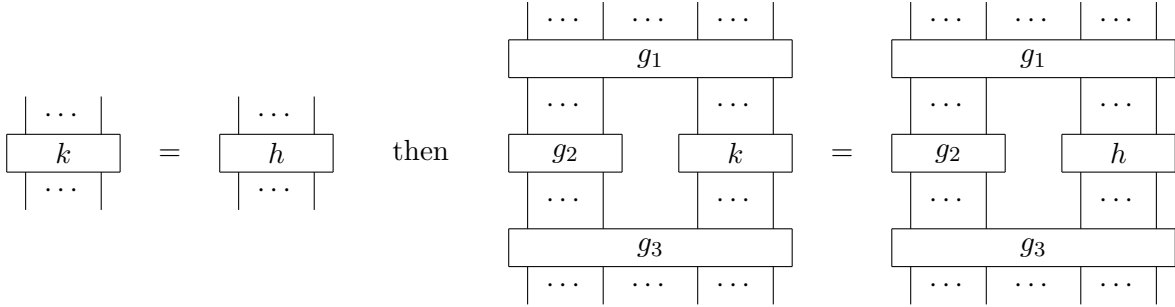
With these generators, we can construct larger diagrams by *composing* them sequentially or in parallel. *Sequential composition* (denoted by $\circ$) of two diagrams is given by connecting the outputs of one diagram to the inputs of the other:



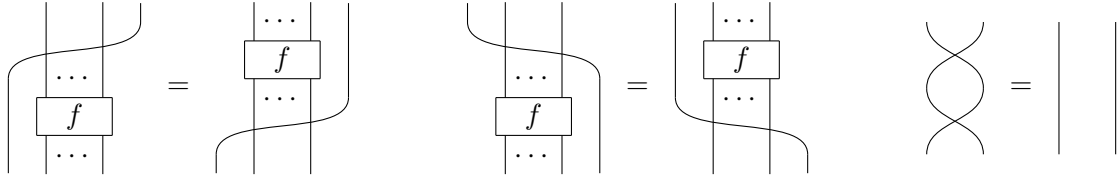
Parallel composition (denoted by $\otimes$) of two diagrams is given by placing two diagrams next to each other:



In addition to composition, we can also reason about diagrams using graphical reasoning. This is done by replacing sub-diagrams within a complex diagram according to a set of *graphical rewrite rules*, that is, if



In particular, a lot of well-known graphical calculi, including the graphical calculus used in this paper, obey the following structural rules that allow us to move diagrams around freely:



Lastly, to relate a graphical calculus to different mathematical systems, one needs to give an interpretation to diagrams such that sequential and parallel compositions are preserved. In our case, the interpretation of diagrams is to C-linear maps, sequential composition is matrix multiplication, and parallel composition is the tensor product.

2.2 Finite-dimensional ZX calculus

We begin with an introduction to the generators of the finite-dimensional ZX calculus as it is presented in [46], describing their interpretation as linear maps, and presenting useful notations. Following this, we introduce the generators whose wires have mixed dimensions as given in [55], including the swap, and the mixed-dimensional Z-spider. The reason we turn to mixed-dimensional ZX at all instead of relying on variants of the standard ZX calculus is that representation theory involves the representation of the same group in different dimensional Hilbert spaces and studies how they interact. This naturally suggests that the most fitting framework to study this would be within a calculus-like mixed-dimensional ZX that is specifically tailored to this.

Note that all the ZX diagrams presented in this paper are read from top to bottom. Furthermore, when working with mixed dimensions, each wire is labelled with its dimension, and only wires with the same dimension can be connected. Since a wire labelled with dimension 1 is just an empty diagram, throughout this paper each wire label will represent an integer strictly bigger than 1.

2.2.1 Generators and their interpretation

We introduce the generators of the qudit ZX calculus and their standard interpretation as given in [45]. We assume that each wire is labelled as a fixed positive integer d , thus for convenience, we ignore all the labels in this subsection. The generators of the qudit ZXW calculus together with their standard interpretation $\llbracket \cdot \rrbracket$ are:

- *Z spider*,

$$\begin{array}{c} d_1 \quad \dots \quad d_n \\ \diagdown \quad \diagup \\ \boxed{\vec{d}} \\ \diagup \quad \diagdown \\ d_{n+1} \quad \dots \quad d_{n+m} \end{array} \xmapsto{\llbracket \cdot \rrbracket} \sum_{j=0}^{\min\{d_i\}_i - 1} a_j |j, \dots, j\rangle \langle j, \dots, j|,$$

where $a_0 := 1$ and $\vec{d} = (a_1, \dots, a_{\min\{d_i\}_i - 1})$.

- *X spider*,

$$\begin{array}{c} n \\ \diagdown \quad \diagup \\ \textcircled{K_j} \\ \diagup \quad \diagdown \\ m \end{array} \xmapsto{\llbracket \cdot \rrbracket} \sum_{\substack{i_1 + \dots + i_m + j \\ \equiv j_1 + \dots + j_n \pmod{d}}} |i_1, \dots, i_m\rangle \langle j_1, \dots, j_n|.$$

where $j \in \mathbb{Z}$. Two X spiders labelled j and j' are equal by definition if $j \equiv j' \pmod{d}$. If $0 \leq j < d$, then X-spiders labelled j that have a single input or output correspond to a computational basis costates/states as follows:

$$\begin{array}{c} | \\ \textcircled{K_j} \end{array} \xmapsto{\llbracket \cdot \rrbracket} \langle j| \quad \begin{array}{c} \textcircled{K_j} \\ | \end{array} \xmapsto{\llbracket \cdot \rrbracket} |d - j\rangle$$

- The *Hadamard box*,

$$\begin{array}{c} | \\ \boxed{H} \end{array} \xmapsto{\llbracket \cdot \rrbracket} \frac{1}{\sqrt{d}} \sum_{k,j=0}^{d-1} \omega^{jk} |j\rangle \langle k|,$$

where $\omega = e^{i\frac{2\pi}{d}}$ is the d -th root of unity. Note that we could also use the other generators to define the Hadamard box (see [45, Rule (HD)]), however, for the sake of simplicity, it is given as a generator here.

- The *identity*,

$$\begin{array}{c} | \\ \text{---} \end{array} \xrightarrow{[\cdot]} I_d = \sum_{j=0}^{d-1} |j\rangle \langle j|.$$

- *Swap*,

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \xrightarrow{[\cdot]} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |j, i\rangle \langle i, j|.$$

Notations

For convenience, we introduce the following notation which will be used throughout the paper:

- We omit the label of the spider when it corresponds to the all-ones vector.

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green circle) } := \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green box with } \vec{1} \text{) } \quad \text{where } \vec{1} = (1, \dots, 1)$$

- We make two short notations with the Z spider whose first/last $d-2$ components of the parameter vector are all zeros as follows:

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green box with } a \text{) } := \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green box with } (0, \dots, 0, a) \text{) } \quad \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green box with } \bar{a} \text{) } := \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green box with } (a, 0, \dots, 0) \text{) } \quad \text{where } a \in \mathbb{C}.$$

- Throughout the paper, we use the following notation:

$$\vec{1}_k = \overbrace{(1, \dots, 1, 0, \dots, 0)}^{d, k-1} \quad \text{for some } d.$$

- We define the cup and cap as follows:

$$\text{---} \text{ (cup) } := \text{---} \text{ (cup with green circle) } \quad \text{---} \text{ (cap) } := \text{---} \text{ (cap with green circle) } \quad (\text{S3})$$

- The phase-free X spider is defined as

$$\begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (red circle) } := \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (red box with } K_0 \text{) }$$

- We use a yellow D box to denote the *dualiser* as defined in [12]:

$$\begin{array}{c} | \\ \text{---} \end{array} \text{ (yellow box with } D \text{) } := \begin{array}{c} \text{---} \\ \text{---} \end{array} \text{ (green circle) } \xrightarrow{[\cdot]} \sum_{i=0}^{d-1} |d-i\rangle \langle i| \quad (\text{Du})$$

- A multiplier [6, 9, 7] labelled by m indicates the number of connections between green and pink nodes.

$$\begin{array}{c} \text{[m]} \end{array} := \begin{array}{c} \text{green node} \\ \vdots \\ \text{pink node} \end{array} \quad \begin{array}{c} \text{[m]} \end{array} := \begin{array}{c} \text{[d-m]} \end{array} \quad \begin{array}{c} \text{[m]} \end{array} := \begin{array}{c} \text{loop with [m]} \end{array} = \begin{array}{c} \text{green node} \\ \vdots \\ \text{pink node} \end{array} \quad (\text{Mu})$$

2.3 ZX rules prerequisite for Spin-ZX calculus

ZX ultimately uses the same qubit dimension for all its wires. Though practical for qubit engineering purposes for broader theoretical physics it is often the case that different dimensional quantum systems interact. For this reason, a calculus designed specifically for the representation of spin must be able to succinctly describe the interaction of any finite-dimensional quantum systems. We present a set of ZX rules for the ZX calculus for finite-dimensional quantum theory here:

$$\begin{array}{c} m_1 \dots m_i \\ \vdots \\ m_{i+1} \dots m_j \end{array} \begin{array}{c} \text{green node } \vec{a} \\ \vdots \\ \text{pink node } \vec{b} \end{array} \begin{array}{c} n_1 \dots n_k \\ \vdots \\ n_{k+1} \dots n_\ell \end{array} = \begin{array}{c} m_1 \dots m_i \\ \vdots \\ m_{i+1} \dots m_j \end{array} \begin{array}{c} \text{green node } \vec{ab'} \\ \vdots \\ \text{pink node } \vec{ab'} \end{array} \begin{array}{c} n_1 \dots n_k \\ \vdots \\ n_{k+1} \dots n_\ell \end{array} = \begin{array}{c} m_1 \dots m_i \\ \vdots \\ m_{i+1} \dots m_j \end{array} \begin{array}{c} \text{green node } \vec{ab'} \\ \vdots \\ \text{pink node } \vec{ab'} \end{array} \begin{array}{c} n_1 \dots n_k \\ \vdots \\ n_{k+1} \dots n_\ell \end{array} \quad (\text{S1})$$

where $m = \min_{t=1}^j m_t$, $n = \min_{t=1}^\ell n_t$, $r = \min_{t=1}^s r_t$, $M = \min\{m, n, r\}$, $\vec{a} = (a_1, \dots, a_{m-1})$,

$$\vec{b} = (b_1, \dots, b_{n-1}), \quad \text{and} \quad \vec{ab'} = (a_1 b_1, \dots, a_{M-1} b_{M-1}, 0, \dots, 0).$$

$$\begin{array}{c} n \\ \vdots \\ m \end{array} \begin{array}{c} \text{green node } K_i \\ \vdots \\ \text{pink node } K_j \end{array} \begin{array}{c} s \\ \vdots \\ t \end{array} = \begin{array}{c} n \\ \vdots \\ m \end{array} \begin{array}{c} \text{green node } K_i \\ \vdots \\ \text{pink node } K_j \end{array} \begin{array}{c} s \\ \vdots \\ t \end{array} = \begin{array}{c} n+s \\ \vdots \\ m+t \end{array} \begin{array}{c} \text{green node } K_{i+j} \\ \vdots \\ \text{pink node } K_{i+j} \end{array} \begin{array}{c} n+s \\ \vdots \\ m+t \end{array} = \begin{array}{c} n+s \\ \vdots \\ m+t \end{array} \begin{array}{c} \text{green node } K_{i+j} \\ \vdots \\ \text{pink node } K_{i+j} \end{array} \begin{array}{c} n+s \\ \vdots \\ m+t \end{array} \quad (\text{S4})$$

$$\begin{array}{c} \text{green node} \end{array} = \begin{array}{c} \text{pink node} \end{array} = \begin{array}{c} \text{vertical wire} \end{array} \quad (\text{S2})$$

$$\begin{array}{c} \text{pink node} \\ \text{[a]} \end{array} = \begin{array}{c} \text{square grid of dots} \end{array} \quad (\text{Ept})$$

$$\begin{array}{c} \text{green node} \\ \vdots \\ \text{pink node} \end{array} = \begin{array}{c} \text{green node} \\ \vdots \\ \text{pink node} \end{array} \quad (\text{B2})$$

$$\begin{array}{c} a \\ \vdots \\ a \end{array} \begin{array}{c} \text{green node } K_j \\ \vdots \\ \text{pink node } K_j \end{array} \begin{array}{c} a \\ \vdots \\ a \end{array} = \begin{array}{c} a \\ \vdots \\ a \end{array} \begin{array}{c} \text{green node } K_{j-1} \\ \vdots \\ \text{pink node } K_j \end{array} \begin{array}{c} a \\ \vdots \\ a \end{array} \quad (\text{HZ})$$

$$\boxed{0} = \text{red dot} \quad (\text{Zer})$$

$$\text{red dot } K_{-j} \text{ with } m \text{ inputs } n_1, n_2 = \text{red dot } K_{-j} \text{ with } m \text{ inputs } n_1, n_2 \text{ and } \boxed{1_N} \quad (\text{K0})$$

where $0 \leq j \leq m-1$, $N = \min\{m, n_1, n_2\}$

$$\text{red dot } K_j \text{ with } \boxed{\vec{a}} = \text{red dot } K_j \text{ with } \boxed{k_j(\vec{a})} \text{ and } \text{red dot } K_j \text{ with } \boxed{\vec{a}} \quad (\text{K2})$$

where $k_j(\vec{a}) = \left(\frac{a_{1-j}}{a_{d-j}}, \dots, \frac{a_{d-1-j}}{a_{d-j}} \right)$.

$$\boxed{D} = \text{red dot with } \boxed{1} \quad (\text{P1})$$

$$\boxed{\vec{a}} \text{ and } \boxed{D} = \boxed{D} \text{ and } \boxed{\overleftarrow{a}} \quad (\text{D1})$$

where $\overleftarrow{a} = (a_{d-1}, \dots, a_1)$

$$\boxed{\vec{a}} \text{ and } \boxed{\vec{b}} \text{ with red dot } K_1 \text{ and } \boxed{\vec{c}} \text{ and } \boxed{r_0} \quad (\text{PA})$$

where $r_0 = \sum_{i=0}^{d-1} a_i b_{d-i} \pmod{d}$, the k^{th} element

of $\vec{c}$ is $c_k = \frac{1}{r_0} \sum_{i=0}^{d-1} a_i b_{k-i} \pmod{d}$

$$\text{red dot with } a, b \text{ inputs and } a+b-1 \text{ output} = \text{red dot with } a, b \text{ inputs and } a+b+k \text{ output} \quad (\text{DA})$$

$$\text{red dot with } \boxed{p}, \boxed{q} \text{ inputs and } d \text{ output} = \text{red dot with } \boxed{q} \text{ input and } d \text{ output} \quad (\text{PC})$$

if $p_i p_j = q_{i+j} \pmod{d}$ for all $1 \leq i < m$ and $1 \leq j < n$.

$$\text{red dot with } n \text{ inputs and } n \text{ outputs} = \text{red dot with } n \text{ inputs and } n \text{ outputs} \quad (\text{DZX})$$

$$\text{red dot with } 2 \text{ inputs and } m \text{ outputs} = \text{red dot with } 2 \text{ inputs and } m \text{ outputs} \quad (\text{XB})$$

where $m > n$.

$$\text{red dot } K_{-1} \text{ with } \boxed{D} \text{ and } m \text{ inputs} = \text{red dot } K_{-(n-1)} \text{ with } \boxed{D} \text{ and } m \text{ inputs} \quad (\text{PK})$$

where $2 \leq n \leq m$.

3 Spin-ZX calculus

In this section, we introduce the Spin-ZX calculus itself, which we elevate to the level of a formal diagrammatic language by embedding it into the ZX calculus as the fragment that describes the recoupling theory of $SU(2)$. This fragment encompasses Yutsis diagrams, and their compositions, which are often called *spin networks* [36]. A brief introduction to Yutsis diagrams can be found in Appendix B.

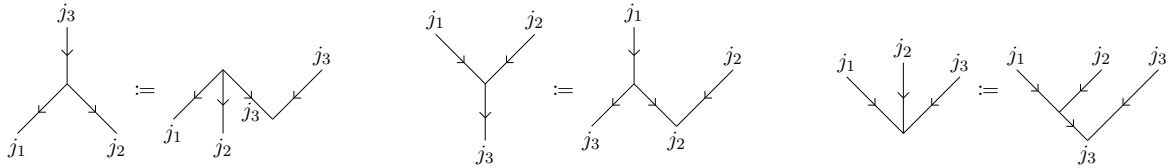
The generators of the Spin-ZX calculus are given in Table 1, also given in the left half of

Figure 1. Using these generators, we can define other orientations of the 3-valent vertex as

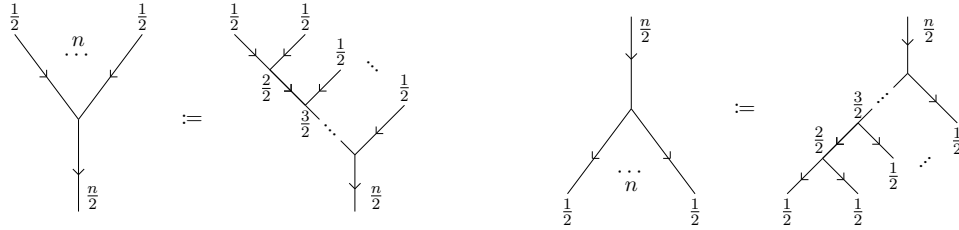
Yutsis diagram	ZX diagram	Interpretation
		$\sum_m j; m\rangle \langle j; m $
		$\sum_{m_1, m_2, m_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} j_1; m_1, j_2; m_2, j_3; m_3\rangle$
		$\sqrt{2j+1} \sum_m (-1)^{j+m+1} \langle j; m, j; -m $

Table 1: Generators of the Spin-ZX calculus. Here, $j, j_1, j_2, j_3 \in \mathbb{N}/2$, $x_{lm} = j_l + j_m - j_n$, and $\vec{1}^- = ((-1)^1, \dots, (-1)^{2j})$. j_1, j_2, j_3 satisfy the Clebsch-Gordan conditions (44). The normalisation factor $N(j_1, j_2, j_3)$ is given by Equation (95).

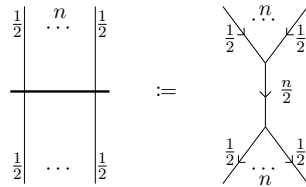
shown below:



We can define the vertex that embeds spin $\frac{n}{2}$ into n copies of the fundamental irrep by stacking the 3-valent vertices as shown below:



Composing the embedding vertex with its adjoint, we can define the symmetriser:



If $j_1 = j_3 = \frac{1}{2}, j_2 = 0$, then we have

$$\begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \quad \searrow \\ \frac{1}{2} \quad \frac{1}{2} \end{array} \mapsto \frac{1}{\sqrt{2}} \begin{array}{c} \pi \quad \pi \end{array}$$

or in the conjugate case

$$\begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \quad \searrow \\ \frac{1}{2} \quad \frac{1}{2} \end{array} \mapsto \frac{1}{\sqrt{2}} \begin{array}{c} \pi \quad \pi \end{array}$$

Next, we provide the rules of the Spin-ZX calculus, which can be derived from the rules of the ZX calculus.

$$\begin{array}{l}
 \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (\text{even-permute}) \\
 (-1)^{j_1+j_2+j_3} \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = (-1)^{j_1+j_2+j_3} \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (\text{odd-permute}) \\
 \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \frac{1}{N(j_1, j_2, j_3)} \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (3j\text{-decomposition}) \\
 \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (\text{isometry}) \\
 \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (\text{projector}) \\
 \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} = \begin{array}{c} \swarrow \downarrow \searrow \\ j_1 \quad j_2 \quad j_3 \end{array} \quad (\text{stacking})
 \end{array}$$

$$\text{cap}_{2k} = 0 = \text{cup}_{2k}$$

$$\text{line}_{2j} \text{ through cup} = \text{line}_{2j} \text{ through cap}$$

$$\text{line}_{2j} \text{ with loop} = \frac{2j+1}{4j} \text{line}_{2j-1}$$

(capping)

(sliding)

(looping)

The rest of the section will motivate the generators and derive the rules of the Spin-ZX calculus.

3.1 $SU(2)$ Representation theory

Let us now understand the representation theory of $SU(2)$ better in order to understand the fragment more clearly and along the way outline how it embeds into ZX. The Lie group $SU(2)$ has importance in many areas in physics. It is most well-known for the theory of quantum angular momentum, but it is also central to the study of isospins, electroweak theory, and qubit quantum computation. $SU(2)$ is the group of 2×2 unitary matrices with determinant 1. A *unitary representation* of $SU(2)$ is a group homomorphism $SU(2) \rightarrow U(\mathcal{H})$ where $U(\mathcal{H})$ denotes the set of unitaries on a (finite-dimensional) Hilbert space $\mathcal{H}$. We can decompose these as a direct sum of *irreducible representations* (irreps). Each such representation corresponds to a dimension, and for every dimension there is an irrep. We label these irreps by half-integers $j \in \mathbb{N}/2$ which in physics are better known as *spins*. The spin- j irrep has dimension $2j + 1$ and is unique up to isomorphism.

The irreps of $SU(2)$. The *fundamental* irrep (the spin-1/2 representation) $SU(2) \rightarrow U(\mathbb{C}^2)$ is simply the 2×2 matrix multiplication of $SU(2)$ itself over $\mathbb{C}^2$. We can then define the higher-spin irreps by using ‘symmetrised copies’ of this representation. First, define the *symmetrisation projection* where for $n = 2j$ we write³ $\mathcal{S}_n$ as the endomorphism over $(\mathbb{C}^2)^{\otimes n}$ such that

$$\mathcal{S}_n(v_1 \otimes \cdots \otimes v_n) = \frac{1}{(n)!} \sum_{\sigma \in \mathfrak{S}_n} U_\sigma(v_1 \otimes \cdots \otimes v_n). \quad (31)$$

Here $\mathfrak{S}_n$ is the $2j$ -element permutation group and the U_σ is the *permutation unitary*⁴. The action of $\mathcal{S}$ is to send every vector to a symmetric vector that is invariant under the action of the permutation unitaries. It is straightforward to check that $\mathcal{S}$ is indeed a projector (i.e. self-adjoint and idempotent). It then defines a subspace of $(\mathbb{C}^2)^{\otimes n}$ that we denote by $\mathcal{H}_j$. This Hilbert-space consists of the symmetric vectors and has dimension $2j + 1$ (i.e. $n + 1$). Since $\mathcal{H}_j$

³Here we encounter the mathematicians labelling of the irreps by integers versus the physicists half integers, the difference is a result of the actual measurement outcomes observed in spin experiments which includes a factor of $\frac{1}{2}$.

⁴Which permutes the labels $U_\sigma(v_1 \otimes \cdots \otimes v_n) \stackrel{\text{def}}{=} v_{\sigma(1)} \otimes \cdots \otimes v_{\sigma(n)}$.

is a subspace of $(\mathbb{C}^2)^{\otimes n}$, there exists an isometry $V_j : \mathcal{H}_j \rightarrow (\mathbb{C}^2)^{\otimes n}$ which embeds the canonical basis of $\mathcal{H}_j$ into $(\mathbb{C}^2)^{\otimes n}$.

Proposition 1. *This isometry is given by the following diagram:*

(32)

where $\vec{\sqrt{a_n}} = \left(\frac{1}{\sqrt{\binom{n}{1}}}, \dots, \frac{1}{\sqrt{\binom{n}{k}}}, \dots, \frac{1}{\sqrt{\binom{n}{n}}} \right)$.

We can verify that this diagram is indeed an isometry:

Proposition 2. $V_j^\dagger V_j = I$. That is,

(33)

Proof. It follows from (S1) and Lemma 6. □

With the above defined, we can represent the symmetriser operator $\mathcal{S}_n$ diagrammatically with the following ZX diagram. This composition of the isometry and its adjoint $V_j V_j^\dagger = \mathcal{S}_n$ is precisely the projection onto the subspace $\mathcal{H}_j$.

Proposition 3.

where $n = 2j$, $\vec{a_n} = \left(\frac{1}{\binom{n}{1}}, \dots, \frac{1}{\binom{n}{k}}, \dots, \frac{1}{\binom{n}{n}} \right)$.

(34)

Proof. It follows from Proposition 1 and spider fusion (S1). □

We can write the canonical orthonormal basis of $\mathcal{H}_j$ as

$$|j; m\rangle := \sqrt{\binom{2j}{j-m}} \mathcal{S}_n \underbrace{(|0\rangle \otimes \dots \otimes |1\rangle)}_{j-m \text{ times } 1}, \quad \text{with } m \in \{-j, -j+1, \dots, j-1, j\} \quad (35)$$

Proposition 4 (Projector). $\mathcal{S}_n^2 = \mathcal{S}_n$, that is,

Proposition 7 (Capping / Cupping).

Proposition 5 (Invariance). For an arbitrary 2×2 unitary represented by $\boxed{U}$, the following equality holds:

Proposition 8 (Symmetriser Sliding).

Proposition 6 (Stacking). Suppose $k \leq j$. Then

Proposition 9 (Looping Relation).

Figure 2: Symmetriser properties

Diagrammatic representation of the Wigner Matrix With the isometry (32) and its dagger, we can represent the Wigner matrix D_u^j as follows:

(38)

where $u \in SU(2)$.

To prove the correctness of the diagram (38), we just need to show the following:

Proposition 10.

$$D_{mn}^j(u) = \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } K_{j,n+1} \text{ with a line labeled } 2j+1 \text{ connecting it to a green square labeled } \sqrt{a_{2j}}. Below this is a red circle labeled } 2j+1 \text{ connected to two green circles labeled } 2j \text{ by a horizontal line. These } 2j \text{ nodes are connected to two more green circles labeled } 2j \text{ by a horizontal line. These } 2j \text{ nodes are connected to a red circle labeled } 2j+1 \text{ by a horizontal line. This } 2j+1 \text{ node is connected to a green square labeled } \sqrt{a_{2j}} \text{ with a line labeled } 2j+1. At the bottom is a red oval labeled } K_{j-m} \text{ with a line labeled } 2j+1 \text{ connecting it to the square.} \\ \left[\begin{array}{c} \text{Left side: A vertical chain of nodes. At the top is a red oval labeled } K_{j,n+1} \text{ with a line labeled } 2j+1 \text{ connecting it to a green square labeled } \sqrt{a_{2j}}. Below this is a red circle labeled } 2j+1 \text{ connected to two green circles labeled } 2j \text{ by a horizontal line. These } 2j \text{ nodes are connected to two more green circles labeled } 2j \text{ by a horizontal line. These } 2j \text{ nodes are connected to a red circle labeled } 2j+1 \text{ by a horizontal line. This } 2j+1 \text{ node is connected to a green square labeled } \sqrt{a_{2j}} \text{ with a line labeled } 2j+1. At the bottom is a red oval labeled } K_{j-m} \text{ with a line labeled } 2j+1 \text{ connecting it to the square.} \end{array} \right] = E, \end{array}$$

where

$$u = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{and} \quad E = \sum_k \frac{\sqrt{(j-m)!(j+m)!(j-n)!(j+n)!}}{k!(j-m-k)!(j+n-k)!(m-n+k)!} a^{j+n-k} b^{m-n+k} c^k d^{j-m-k}$$

where the sum runs over all the values of k for which the argument of every factorial is non-negative [37].

3.2 Spin coupling

Due to linearity, we can define an action of a group element $U \in SU(2)$ on $\mathcal{H}_j$ as follows:

$$U \cdot (v_1 \otimes \cdots \otimes v_{2j}) = (Uv_1) \otimes \cdots \otimes (Uv_{2j}). \quad (39)$$

where Uv_j is the usual matrix multiplication of the 2×2 matrix U with the $\mathbb{C}^2$ vector v_j . The corresponding action of a Lie algebra element, $a \in \mathfrak{su}(2)$, is then given as

$$a \cdot (v_1 \otimes \cdots \otimes v_{2j}) = \sum_{k=1}^{2j} v_1 \otimes \cdots \otimes (av_k) \otimes \cdots \otimes v_{2j}. \quad (40)$$

These are the fundamental building blocks from which other representations are built. Any finite-dimensional representation of $SU(2)$ is completely reducible, meaning it can be written as a direct sum of irreps. Specifically, a tensor product of irreps can be decomposed into a direct sum of irreps via a bijective intertwiner (or equivariant map). Recoupling theory aims to study the space of such intertwiners.

Clebsch-Gordan coefficients. Given $\mathcal{H}_{j_1}$ and $\mathcal{H}_{j_2}$, their tensor product $\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2}$ can be decomposed into a direct sum of irreps with the following equivalence of representations

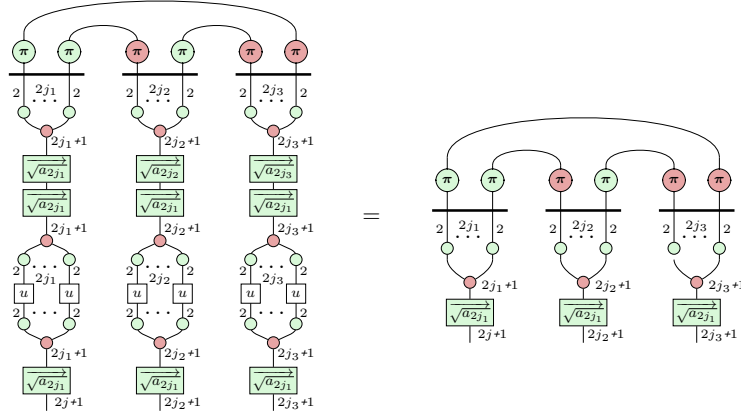
$$C : \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \longrightarrow \bigoplus_{j=|j_1-j_2|}^{j_1+j_2} \mathcal{H}_j, \quad (41)$$

Specifically for any state $|j; m\rangle \in \mathcal{H}_j$ we can write

$$|j; m\rangle = \sum_{m_1=-j_1}^{j_1} \sum_{m_2=-j_2}^{j_2} |j_1; m_1, j_2; m_2\rangle \langle j_1; m_1, j_2; m_2 | j; m\rangle = \sum_{m_1=-j_1}^{j_1} \sum_{m_2=-j_2}^{j_2} C_{j_1 m_1 j_2 m_2}^{j m} |j_1; m_1, j_2; m_2\rangle \quad (42)$$

depicts a vector that belongs to the one dimensional space $\text{Inv}_{SU(2)}(\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3})$, which can be justified by the fact that the vector remains unchanged after plugging in the diagrams of Wigner matrices (38) with the same $u \in SU(2)$ as shown in Proposition 11. Here the normalisation factor $N(j_1, j_2, j_3)$ is given by Equation (95). This should be interpreted as a kind of ‘railroad switch’, where the fundamental wires within the three symmetrised bundles redistribute between themselves. Because we are dealing with symmetrised spaces, we only care about the number of wires going from each bundle to the other bundle. It turns out that there is only one way to connect the wires when it is not impossible. The Clebsch-Gordan conditions (44) precisely state when such a recoupling is possible.

Proposition 11.



Now we derive a simplified ZX diagram which represents the state $|j_1, j_2, j_3\rangle$. This 3j state corresponds to the basic generator of Yutsis diagrams, which is a 3-valent vertex.

Theorem 1.

$$|j_1, j_2, j_3\rangle = \frac{1}{N(j_1, j_2, j_3)} \text{ (Diagram) } \longleftrightarrow \begin{array}{c} \swarrow \\ j_1 \\ \downarrow \\ j_2 \\ \searrow \\ j_3 \end{array} \quad (48)$$

The diagram in (48) is a complex ZX diagram representing the 3j state. It features a central vertex with three outgoing wires labeled $x_{12}+1$, $x_{13}+1$, and $x_{23}+1$. The wires are decorated with various labels including $\frac{1}{a}$, $\frac{1}{a^-}$, and square roots of a_{2j_1} , a_{2j_2} , and a_{2j_3} . The diagram is equated to a simplified tree-like structure with three branches labeled j_1 , j_2 , and j_3 .

where $x_{12} = j_1 + j_2 - j_3$, $x_{13} = j_1 + j_3 - j_2$, $x_{23} = j_2 + j_3 - j_1$; $\overrightarrow{\frac{1}{a}} = \left((-1)^1 \binom{x_{k\ell}}{1}, \dots, (-1)^{x_{k\ell}} \binom{x_{k\ell}}{x_{k\ell}} \right)$.

The arrows pointing out of the vertex indicate it is an output. A general 3-valent vertex can also have inputs, which are represented by wires entering the vertex. Changing the direction of the wires corresponds to negating the magnetic basis, along with some phase. We can change the wire directions by composing with the ZX diagram shown below, which is an analogue of the singlet effect.

$$\begin{array}{c} j \\ \swarrow \quad \searrow \\ \vdots \end{array} \mapsto \sqrt{2j+1} \begin{array}{c} \boxed{\frac{1}{a}} \\ \downarrow \\ \text{K}_3 \end{array} \quad (49)$$

The diagram in (49) shows a transformation. On the left, a 3-valent vertex with two incoming wires labeled j and one outgoing wire (represented by a vertical dotted line). An arrow points to the right, where the same vertex is shown with a different configuration: a square box labeled $\frac{1}{a}$ is connected to a red circle labeled K_3 , and the entire structure is multiplied by $\sqrt{2j+1}$.

where $\overrightarrow{1} = \left((-1)^1, \dots, (-1)^{2j} \right)$. Composing this diagram with the 3j-state gives us the injection map given in Equation (42).

Proposition 12. *The injection $\mathcal{H}_{j_3} \longrightarrow \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2}$ as given in Equation (42) can be represented by the following diagram:*

(50)

In general, the composition of 3j-states naturally gives a network of interacting spin spaces, which is often termed a *spin network*. For example, the four-valent intertwiner can be obtained by composing two 3-valent vertices as follows:

In the rest of the paper, we will omit the arrows on the edges and use the orientation of the diagrams to indicate input and output wires.

3.4 Wigner's $3jm$ -symbol.

We can decompose the 3j vector in the magnetic basis as:

$$|j_1, j_2, j_3\rangle = \sum_{m_1, m_2, m_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} |j_1; m_1, j_2; m_2, j_3; m_3\rangle, \quad (51)$$

where the coefficients are called the *Wigner's $3jm$ -symbol*. We can derive their explicit form by plugging in $|j_1; m_1, j_2; m_2, j_3; m_3\rangle$ in Theorem 1.

Theorem 2. *Suppose $m_1 + m_2 + m_3 = 0$, and j_1, j_2, j_3 satisfy the Clebsch-Gordan conditions. Then, the $3jm$ -symbol is given by the following*

where

$$D = (-1)^{j_1-j_2-m_3} \sqrt{\frac{(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}{(j_1+j_2+j_3+1)!}} \\ \times \sqrt{(j_1-m_1)!(j_1+m_1)!(j_2-m_2)!(j_2+m_2)!(j_3-m_3)!(j_3+m_3)!} \\ \times \sum_{k=K}^N \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!(j_2+m_2-k)!}$$

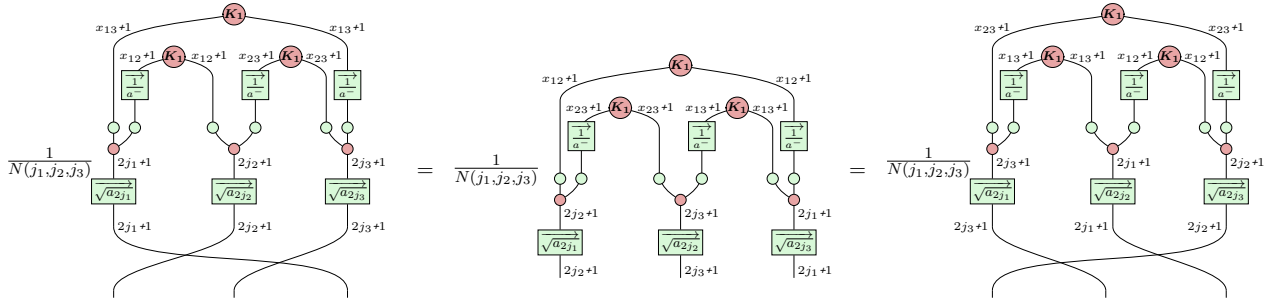
which is given in [52].

As one can see from the diagrams (48) and (50), these $3jm$ -symbol have more symmetries than the Clebsch-Gordan coefficients because they treat the three Hilbert spaces on the same level. We can now diagrammatically prove the symmetry properties of the $3jm$ -symbol.

Proposition 13. *A $3j$ -symbol is invariant under an even permutation of its columns:*

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \begin{pmatrix} j_2 & j_3 & j_1 \\ m_2 & m_3 & m_1 \end{pmatrix} = \begin{pmatrix} j_3 & j_1 & j_2 \\ m_3 & m_1 & m_2 \end{pmatrix}$$

Diagrammatically, this is stated as follows:



where permutation of the columns of a $3j$ -symbol is realised by dragging the corresponding parts of the diagram to the right place.

The following property can be proved similarly, by invoking the last equality of Equation (113).

Proposition 14. *An odd permutation of the columns gives a phase factor:*

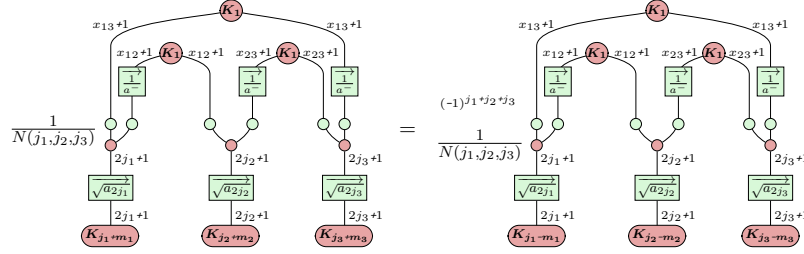
$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_2 & j_1 & j_3 \\ m_2 & m_1 & m_3 \end{pmatrix} \\ = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_3 & j_2 \\ m_1 & m_3 & m_2 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_3 & j_2 & j_1 \\ m_3 & m_2 & m_1 \end{pmatrix}$$

Furthermore, we have

Proposition 15. *Changing the sign of the m quantum numbers gives a phase:*

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

Diagrammatically, this is stated as follows:



4 The Lie algebra $\mathfrak{su}(2)$

One immediate advantage of the embedding is that it expands the scope of what the Spin-ZX calculus as diagrams can formally interact with. In particular, we can straightforwardly describe via diagrams properties of the related Lie algebra $\mathfrak{su}(2)$. As all the elements are embedded into ZX it is simpler to formally interact elements of the algebra and group while maintaining confidence in the formal meaning of what is depicted and calculated.

We first introduce the angular momentum operators. The angular momentum operators are the generators of $SU(2)$ in the sense that they are a basis for the group's Lie algebra $\mathfrak{su}(2)$. Any $g \in SU(2)$ can be represented as the exponent of some $\mathfrak{g} \in \mathfrak{su}(2)$, specifically $g = e^{i\mathfrak{g}}$. The elements of $SU(2)$ and $\mathfrak{su}(2)$ are also in local correspondence, as the derivate of the group at the identity gives the Lie algebra. In this way, the algebra elements also act as infinitesimal transformations (i.e. generators of $SU(2)$).

They can be defined in terms of the magnetic basis⁶ as follows:

$$\begin{aligned} J_1 |j; m\rangle &= \frac{1}{2} \sqrt{(j-m)(j+m+1)} |j; m+1\rangle + \frac{1}{2} \sqrt{(j+m)(j-m+1)} |j; m-1\rangle \\ J_2 |j; m\rangle &= \frac{1}{2i} \sqrt{(j-m)(j+m+1)} |j; m+1\rangle - \frac{1}{2i} \sqrt{(j+m)(j-m+1)} |j; m-1\rangle \\ J_3 |j; m\rangle &= m |j; m\rangle \end{aligned}$$

The matrices D_g^j representing the operators $g = e^{-i\alpha \vec{n} \cdot \vec{J}}$ on the Hilbert space H_j in the $|j; m\rangle$ basis define a $(2j+1)$ -dimensional irreducible representation of $SU(2)$, where $\vec{J} = (J_1, J_2, J_3)$ [37]. These matrices D_g^j , called *Wigner matrices*, are square matrices of size $2j+1$, whose elements are given by [37, 38]:

$$D^{jmn}(g) = \langle j; m | e^{-i\alpha \vec{n} \cdot \vec{J}} | j; n \rangle$$

The angular momentum operators J_i generate a non-abelian rotation group. They are characterised by the *fundamental commutation relations of angular momentum*:

$$[J_j, J_k] = i \epsilon_{jkl} J_l \quad (52)$$

where the Levi-Civita symbol ϵ_{jkl} is 1 if (j, k, l) is an even permutation of $(1, 2, 3)$, -1 if it is an odd permutation, and 0 otherwise. Alongside these generators, the other key operators

⁶The m can be seen as the alignment of the angular momentum in the z-direction (an arbitrary choice but one used by convention), along which one could discriminate these states physically by applying a magnetic field. The variation in the energy levels due to doing this is the famous Zeeman effect.

of interest are the ladder operators which allow us to transition between the states within a representation

$$J_+ := J_1 + iJ_2 \quad \text{and} \quad J_- := J_1 - iJ_2$$

whose action on the magnetic basis is

$$J_+ |j; m\rangle = \sqrt{(j-m)(j+m+1)} |j; m+1\rangle$$

$$J_- |j; m\rangle = \sqrt{(j+m)(j-m+1)} |j; m-1\rangle.$$

Using the ladder operators, we can write the commutation relations equivalent to (52) as:

$$[J_+, J_-] = 2J_3 \quad \text{and} \quad [J_3, J_\pm] = \pm J_\pm. \quad (53)$$

It is easy to check that the ladder operators can be expressed as the following diagrams.

$$J_+ \mapsto \sqrt{2j} \begin{array}{c} \text{K}_1 \\ \text{L}_j \end{array} \quad \text{and} \quad J_- \mapsto \sqrt{2j} \begin{array}{c} \text{L}_j \\ \text{K}_{-1} \end{array}$$

where $L_j = \frac{1}{\sqrt{2j}}(\sqrt{2(2j-1)}, \dots, \sqrt{(k+1)(2j-k)}, \dots, \sqrt{(2j+1)(2j-2j)})$. We can add/subtract the above diagrams to construct the angular momentum operators J_1 and J_2 .

$$\frac{J_+ \pm J_-}{\sqrt{2j}} = \begin{array}{c} \text{K}_1 \\ \text{L}_j \end{array} \pm \begin{array}{c} \text{L}_j \\ \text{K}_{-1} \end{array} = \begin{array}{c} \text{K}_1 \\ \text{D} \\ \text{L}_j \\ \text{K}_{-1} \end{array} \pm \begin{array}{c} \text{L}_j \\ \text{D} \\ \text{L}_j \\ \text{K}_{-1} \end{array} = \pm \begin{array}{c} \pm 1 \\ \text{D} \\ \text{L}_j \\ \text{K}_{-1} \end{array} = \pm \begin{array}{c} \text{D} \\ \pm 1 \\ \text{L}_j \\ \text{K}_{-1} \end{array}$$

This allows us to define

$$J_1 = \frac{J_+ + J_-}{2} = \frac{\sqrt{2j}}{2} \begin{array}{c} \text{D} \\ 1 \\ \text{L}_j \\ \text{K}_{-1} \end{array} \quad \text{and} \quad J_2 = \frac{J_+ - J_-}{2i} = \frac{-\sqrt{2j}}{2i} \begin{array}{c} \text{D} \\ -1 \\ \text{L}_j \\ \text{K}_{-1} \end{array}$$

Finally, to obtain the diagram for J_3 , we observe that it is a diagonal operator given by

$$J_3 = j \begin{array}{c} \text{A}_j \end{array} \quad \text{where} \quad A_j = \frac{1}{j}(j-1, \dots, j-k, \dots, j-2j).$$

Next, we derive the commutation relations given in (53).

Proposition 16. $[J_+, J_-] = 2J_3$, or diagrammatically,

$$\begin{array}{c} \sqrt{2j} \begin{array}{c} \text{L}_j \\ \text{K}_{-1} \end{array} \\ \sqrt{2j} \begin{array}{c} \text{K}_1 \\ \text{L}_j \end{array} \end{array} - \begin{array}{c} \sqrt{2j} \begin{array}{c} \text{K}_1 \\ \text{L}_j \end{array} \\ \sqrt{2j} \begin{array}{c} \text{L}_j \\ \text{K}_{-1} \end{array} \end{array} = 2j \begin{array}{c} \text{A}_j \end{array} \quad (54)$$

Proposition 17. $[J_3, J_+] = J_+$, or diagrammatically,

$$\begin{array}{c} \sqrt{2j} \text{ } \boxed{K_1} \\ | \\ \sqrt{2j} \text{ } \boxed{L_j} \\ | \\ j \text{ } \boxed{A_j} \end{array} - \begin{array}{c} j \text{ } \boxed{A_j} \\ | \\ \sqrt{2j} \text{ } \boxed{K_1} \\ | \\ \sqrt{2j} \text{ } \boxed{L_j} \end{array} = \begin{array}{c} \sqrt{2j} \text{ } \boxed{K_1} \\ | \\ \sqrt{2j} \text{ } \boxed{L_j} \end{array} \quad (55)$$

Proposition 18. $[J_3, J_-] = -J_-$, or diagrammatically,

$$\begin{array}{c} \sqrt{2j} \text{ } \boxed{L_j} \\ | \\ \sqrt{2j} \text{ } \boxed{K_{-1}} \\ | \\ j \text{ } \boxed{A_j} \end{array} - \begin{array}{c} j \text{ } \boxed{A_j} \\ | \\ \sqrt{2j} \text{ } \boxed{L_j} \\ | \\ \sqrt{2j} \text{ } \boxed{K_{-1}} \end{array} = -\sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ \boxed{K_{-1}} \end{array} \quad (56)$$

4.1 Expressing Hamiltonians

The ZXW calculus has previously been used to sum and exponentiate qubit Hamiltonians [50]. Similarly, it is possible to extend our generators, enabling the Spin-ZX calculus to express Hamiltonians in terms of angular momentum operators provided we also make use of the ZXW W node

$$\begin{array}{c} \blacktriangle \end{array} \xrightarrow{[\![\cdot]\!]} |00\rangle \langle 0| + \sum_{i=1}^{d-1} (|0i\rangle + |i0\rangle) \langle i|.$$

and the triangle node

$$\begin{array}{c} \text{yellow triangle} \end{array} := \begin{array}{c} \text{green circle} \\ | \\ \blacktriangle \end{array} \xrightarrow{[\![\cdot]\!]} I_d + \sum_{i=1}^{d-1} |i\rangle \langle 0|.$$

We can then express the controlled J_1 , J_2 , and J_3 operators in our calculus respectively as follows:

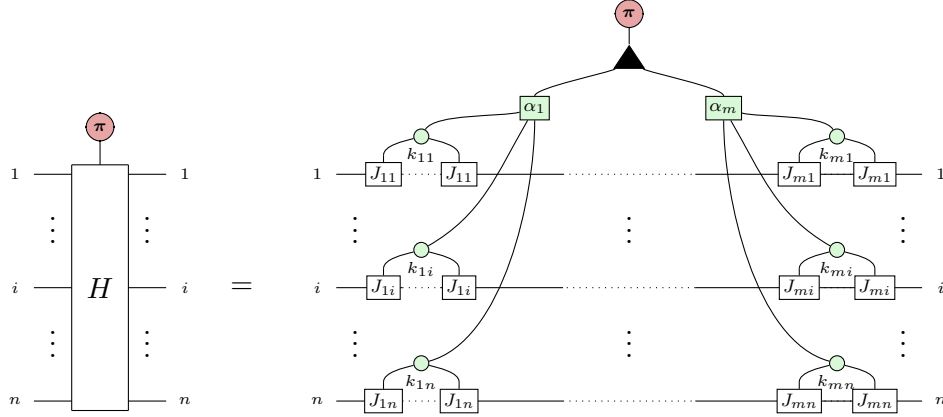
$$\begin{array}{c} \boxed{J_1} \end{array} := \begin{array}{c} \text{diagram for } J_1 \end{array} \quad \begin{array}{c} \boxed{J_2} \end{array} := \begin{array}{c} \text{diagram for } J_2 \end{array} \quad \begin{array}{c} \boxed{J_3} \end{array} := \begin{array}{c} \text{diagram for } J_3 \end{array}$$

where $d = 2j + 1$ and $\vec{0}, \vec{1} = (\overbrace{0, \dots, 0}^{d-1}, \overbrace{1, \dots, 1}^{d-1})$.

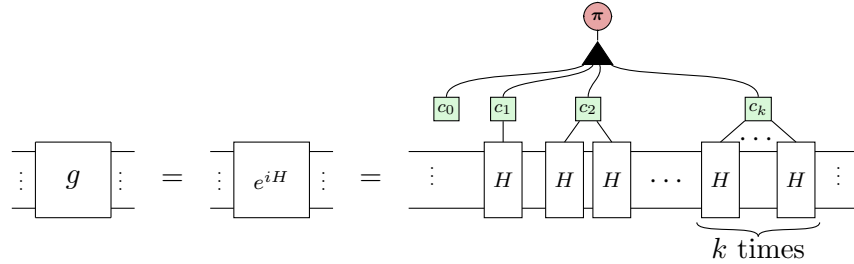
Then a Hamiltonian composed of angular momentum operators can be written as a self-adjoint operator of the form:

$$H = \sum_{r=1}^m \alpha_r \bigotimes_{i=1}^n J_{ri}^{k_{ri}}$$

where each α_r is a real number, $J_{ri} \in \{J_1, J_2, J_3\}$. It can be diagrammatically represented as follows:



We can now exponentiate the Hamiltonian as done in [50] as follows:



for some coefficients $c_0, \dots, c_k$ where k is the dimension of the Hilbert space.

5 Applications

5.1 Permutational quantum computing

An early model of spin-network computation was introduced by Marzuoli and Rasetti in [39]. This, in turn, was specialised into a formal computational model called permutational quantum computing (PQC) by Jordan in [30] by restricting computations to the expectation values of permutations between states defined by binary-tree-shaped spin networks that can be more simply understood as the basis of total angular momentum outcomes from a collection of qubits. Consider three qubits in the standard computational basis $\{b_i b_j b_k\}$ where each b is a binary value detailing if the qubit is spin up or spin down. It can be shown that an alternative *spin basis* is given by pairwise coupling via total angular momentum measurements. For instance, we could measure the angular momenta of the first two qubits, getting either spin-1 or spin-0 and then getting the total angular momentum measurement for the spin of this outcome with the final qubit leading to total spin- $\frac{1}{2}$ or spin- $\frac{3}{2}$. If we then also fix the state of this final angular momentum's m -value, i.e. its z-direction angular momentum, this also gives us a basis (and indeed, any other choice of a fixed coupling pattern is another such basis see Figure 4 for an example).

The PQC model is the class of computations defined as expectation values of qubit swap operators on states prepared in (one of the) spin bases. These computations have previously

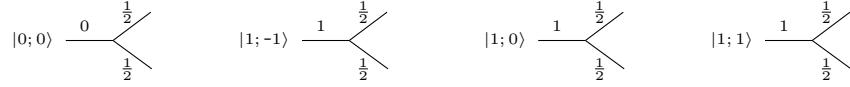


Figure 3: A graphical depiction of how the basis constructed by combining angular momentum of two spin- $\frac{1}{2}$ systems, and the possible outcomes of total and z -directed angular momenta.

been done by hand by making use of the Racah identities, and the interested reader is directed to [30, 39] for further explanation and details. Here, however, by using the work in [23] it is possible to simply represent these computations as ZX-based tensor networks and with the symmetry-based rewrite language described here reduce these down to scalar values.

Consider the following example where the final leaves of the tree are all spin- $\frac{1}{2}$ i.e. qubits:

$$\left\langle \begin{array}{c} \frac{1}{2} \\ \diagup \quad \diagdown \\ 1 \end{array} \right| \begin{array}{c} \diagup \quad \diagdown \\ \diagdown \quad \diagup \\ \hline \end{array} \right| \begin{array}{c} \diagdown \quad \diagup \\ \diagup \quad \diagdown \\ 0 \end{array} \right\rangle = \frac{\sqrt{3}}{2} \quad (57)$$

which was calculated in [30] via the Racah transformation rules in the manner first seen in [39]. To alternatively represent this in a directly graphical manner, we first find the normalisations of the PQC spin basis states as diagrams. For the basis element where the first coupling forms spin-1, we find

$$\begin{aligned} & \left\langle \begin{array}{c} \frac{1}{2} \\ \diagup \quad \diagdown \\ 1 \end{array} \right| \begin{array}{c} \diagup \quad \diagdown \\ \diagdown \quad \diagup \\ \hline \end{array} \right\rangle \\ &= \text{Diagram with nodes } \pi, \text{ and } \frac{1}{2} \text{ and edges labeled } 2, 3, \text{ and } \frac{3}{2} \\ &\stackrel{(4)}{=} \text{Diagram with nodes } \pi, \text{ and } \frac{1}{2} \text{ and edges labeled } 2, 3, \text{ and } \frac{3}{2} \stackrel{(9)}{=} \frac{3}{2} \cdot \frac{3}{2} = \frac{3}{2} \end{aligned}$$

For the element where they form a spin-0, we find that

$$\left\langle \begin{array}{c} \frac{1}{2} \\ \diagup \quad \diagdown \\ 0 \end{array} \right| \begin{array}{c} \diagup \quad \diagdown \\ \diagdown \quad \diagup \\ \hline \end{array} \right\rangle = \text{Diagram with nodes } \pi, \text{ and } \frac{1}{2} \text{ and edges labeled } 2, 3, \text{ and } \frac{3}{2} = 2 \quad (58)$$

We can then represent the whole computation as a closed diagram composed of the representations of the states with a permutation between them. The solution is given by applying the rewrite rules:

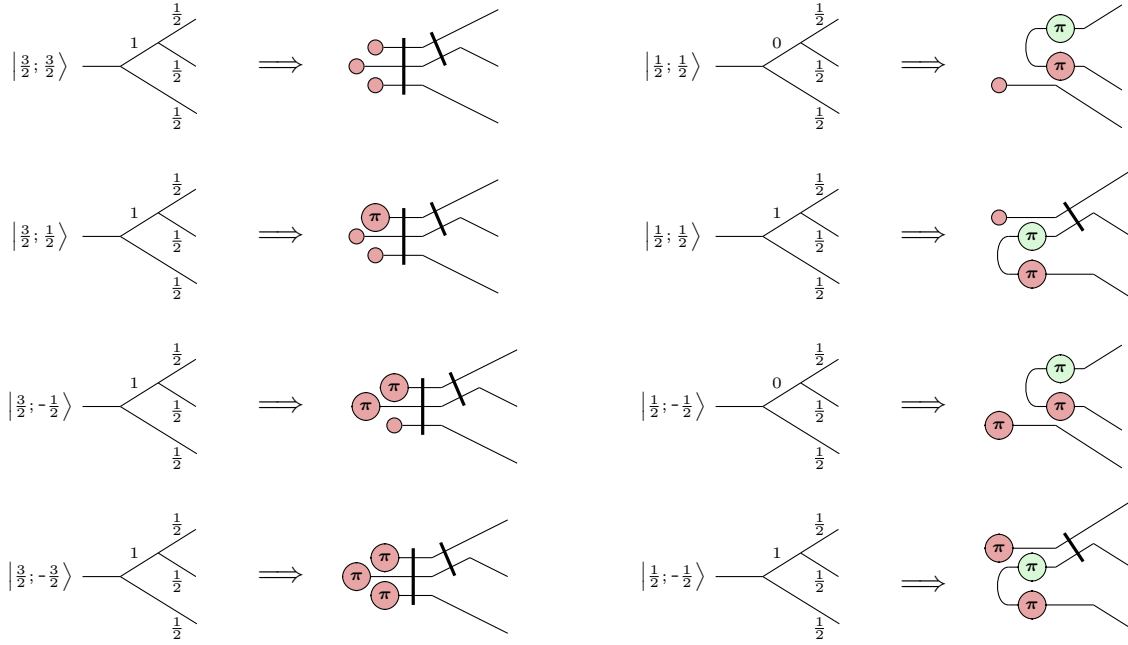


Figure 4: Graphical depiction of a coupling basis of three qubits, where the pairwise coupling of the spaces proceeds from the left (other possibilities give alternative bases). The top row details the ways one can obtain total angular momentum of $\frac{1}{2}$, the second of $\frac{3}{2}$. The elements in the rows correspond to the specific different states, giving a final m value on the spaces at the bottom of the trees. Note that in the absence of specifying the m values, the set of diagrams belongs to one of a class of three different spin networks determined by the recoupling outcomes in the tree.

$$\begin{aligned}
 & \frac{1}{\sqrt{3}} \left\langle \begin{array}{c} \frac{1}{2} \\ \frac{1}{2} \end{array} \middle| \begin{array}{c} \frac{1}{2} \\ 1 \end{array} \right\rangle \left| \begin{array}{c} \frac{1}{2} \\ 0 \end{array} \right\rangle \begin{array}{c} \frac{1}{2} \\ \frac{1}{2} \end{array} \right\rangle \quad (59) \\
 &= \frac{1}{\sqrt{3}} \begin{array}{c} \text{Diagram 1: A complex spin network with multiple nodes and connections, including a central box labeled } \frac{1}{2} \end{array} = \frac{1}{\sqrt{3}} \begin{array}{c} \text{Diagram 2: A similar spin network with a different configuration of nodes and connections} \end{array} \\
 & \stackrel{(9)}{=} \frac{1}{\sqrt{3}} \cdot \frac{3}{2} \begin{array}{c} \text{Diagram 3: A simplified spin network with a central box labeled } \frac{1}{2} \end{array} = \frac{\sqrt{3}}{2} \begin{array}{c} \text{Diagram 4: A further simplified spin network} \end{array} = \frac{\sqrt{3}}{2} \begin{array}{c} \text{Diagram 5: A single node} \end{array} = \frac{\sqrt{3}}{2}
 \end{aligned}$$

Where the initial scalar of $\frac{1}{\sqrt{3}}$ is the overall normalisation given the inner product of the two initial states.

5.2 Analysis of SPT phases

In Ref. [24] it was shown that a modification of the ZX calculus could find use beyond quantum computing and in broader theoretical physics. In particular, it was used to analyse the AKLT (Affleck-Kennedy-Lieb-Tasaki) state. The form presented in [24] contains a large degree of redundancy due to the true focus of the calculus used being qubits quantum computing and not spin, in this section we will show some of the same arguments but in a simpler fashion with simpler derivations. The AKLT state [3, 4], named after Affleck, Lieb, Kennedy, and Tasaki, is a quantum state that arises as the ground state of the one-dimensional AKLT Hamiltonian. This Hamiltonian is defined for a chain of spin-1 degrees of freedom, where each ‘site’ (element in the chain) interacts with its neighbour in the following manner:

$$H = \sum_i \left(\vec{S}_i \cdot \vec{S}_{i+1} + \frac{1}{3} (\vec{S}_i \cdot \vec{S}_{i+1})^2 \right)$$

Here, $\vec{S}_i$ represents the spin-1 operator at site i , and the sum runs over all pairs of neighbouring sites. The AKLT state is formed of spin-1 sites partitioned into two spin-1/2 sites. These spin-1/2 sites then form singlet states with the neighbouring sites, creating a state with a maximum total spin of 1 between neighbouring spin-1’s. This ensures the state is annihilated by the spin-2 projectors included in the Hamiltonian to give us the ground state.

For our interests, the AKLT state has two notable properties:

1. When the chain has open boundary conditions, the termination of the chain leaves unpaired spin-1/2 particles at each end, leading to a four-fold degeneracy due to the two possible states (spin up or down) of each edge spin.
2. The AKLT state exhibits a non-local hidden order known as string order.
3. This order is characterised by being an equal-weight superposition of antiferromagnetic states where, ignoring the zero spin states, the remaining spins alternate in orientation (anti-ferromagnetic order).
4. This means that a configuration like $|1, 0, 0, 0, -1, 0, 0, 0, 1\rangle$ is allowed, while $|1, 0, 0, 1, 0, 0, 0, 1\rangle$ is not.

Proceeding in the manner of Ref. [24], we connect the AKLT state’s MPS (matrix product state) representation to a diagrammatic one. Recalling that any quantum state can be written as a product of matrices, i.e. as a matrix product state, via

$$|\psi\rangle = \sum_{j_1, \dots, j_N} \sum_{\alpha_2, \dots, \alpha_{N+1}} M_{\alpha_1, \alpha_2}^{[1]j_1} \cdots M_{\alpha_N, \alpha_{N+1}}^{[N]j_N} |j_1, j_2, \dots, j_N\rangle. \quad (60)$$

The indices j_i are called physical indices because they span the local Hilbert space at a given site n for example, $j_i = 0, \pm 1$ for spin-1. For a given j_i and n , the $M_{\alpha_i, \alpha_{i+1}}^{[n]j_n}$ are matrices in the indices α_i , known as bond indices. Though this does offer an exact representation of any finite system, the maximum dimension of the bond indices, or the bond dimension χ , typically grows exponentially with system size.

The AKLT state is notable for its simple representation as an MPS with bond dimension $\chi = 2$ meaning each site is defined by a 2×2 matrix (with a 1×2 matrix at the boundaries).

where $\vec{b} = (1, \frac{1}{3}, \frac{1}{3})$.

$$(68)$$

$$(69)$$

The result is that the lattice in Equation (66) reduces to a graph state, though as compared with the original paper the demonstration of this is more diagrammatically elegant (C.F. appendix 2 in Ref. [24]).

In the original work, further properties were demonstrated diagrammatically which can also be converted to the Spin-ZX calculus; however, it is not our intention to replicate all the results of that paper. The ambition is to highlight to interested readers from the domain of condensed matter physics that the Spin-ZX calculus is a versatile tool for $SU(2)$ -based system in this domain.

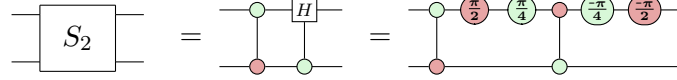
5.3 Analysis of QML ansätze

The ZX calculus can be used to represent derivatives and integrals of maps and states represented as diagrams themselves [59, 57]. One application of this is quantum machine learning.

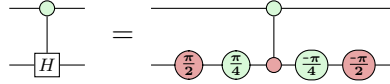
Here we make use of the spin-network ansatz seen in Ref. [22] which uses parametrised $SU(2)$ equivariant unitaries to learn maps and states which possess this symmetry. Let us focus on the two-qubit unitary vertex gate of Ref. [22]. It functions by taking the fact that $\frac{1}{2} \otimes \frac{1}{2} \simeq 0 \oplus 1$ and applying a phase to the spin-0 component. Essentially constructing the map $V(\theta) = |J=1\rangle\langle J=1| + e^{i\theta} |J=0\rangle\langle J=0|$ which we can write as a sum of projectors with a phase as:

$$(70)$$

To reach a single diagram form we make use of the fact that the above is a direct partition of the qubit space into the block diagonal spin basis. The quantum gate that achieves this is the 2-qubit Schur transform:



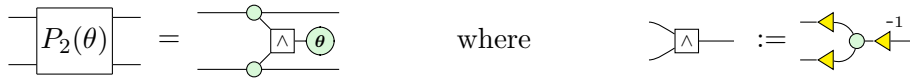
Here, we used a shorthand for the controlled-Hadamard gate, defined as follows:



Following [22], we can now express $V(\theta)$ using these definitions as a single diagram:

$$\begin{aligned}
 V(\theta) &= S_2 P_2(\theta) S_2^\dagger \\
 &= \text{[Diagram of } S_2 P_2(\theta) S_2^\dagger \text{]} \\
 &= \text{[Simplified diagram of } V(\theta) \text{]}
 \end{aligned} \tag{71}$$

where $P_2(\theta)$ is the controlled Z -rotation, defined as follows:

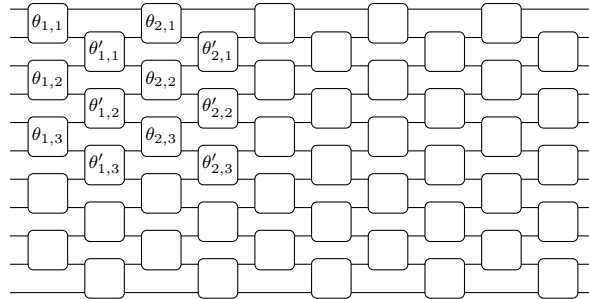


This applies the phase to the spin-0 sector once it is mapped to a single qubit computational basis element by the Schur transform.

As an application of this ansatz and calculus on it consider the permutational quantum computing model described above (Section 5.1). Here the expectation of the Hamiltonian is written as:

$$\langle H \rangle = \langle \psi | W(\vec{\theta}) S_n W^\dagger(\vec{\theta}) | \psi \rangle$$

where $W(\vec{\theta})$ is a brick-wall ansatz made up of a tiling of the vertex gates $V(\theta)$:



and the PQC Hamiltonian itself S_n is an arbitrary permutation of n qubits, with the initial states $|\psi\rangle$ set as the product of singlets, i.e. the spin-0 state:

$$|\psi\rangle = \left(\begin{array}{c} \pi \\ \pi \end{array} \right)^{\otimes k} \quad \text{that can be physically implemented as} \quad \begin{array}{c} \pi \\ \pi \end{array} = \text{[Diagram of singlet state implementation]}$$

Given that each parameter appears exactly once in the ansatz, we can represent the expectation values as follows [35]:

$$\langle H \rangle = \begin{array}{c} \theta_1 \\ \vdots \\ \theta_m \end{array} \begin{array}{c} \vdots \\ E_H \\ \vdots \end{array} \begin{array}{c} -\theta_1 \\ \vdots \\ -\theta_m \end{array} := \begin{array}{c} \theta_1 \quad \theta_m \\ \vdots \quad \vdots \end{array} \begin{array}{c} \vdots \\ U \\ \vdots \end{array} \begin{array}{c} \vdots \\ H \\ \vdots \end{array} \begin{array}{c} -\theta_m \quad -\theta_1 \\ \vdots \quad \vdots \end{array} \begin{array}{c} \vdots \\ U^\dagger \\ \vdots \end{array} = \langle 0 | U^\dagger(\vec{\theta}) H U(\vec{\theta}) | 0 \rangle$$

Since H can be represented as a diagram, we can calculate the gradient variance as follows [57]:

$$\text{Var} \left(\frac{\partial \langle H \rangle}{\partial \theta_j} \right) = \begin{array}{c} \text{Diagram showing the variance calculation. It features a central box labeled } E_H \text{ with multiple inputs and outputs. The inputs are labeled } \theta_1, \dots, \theta_m \text{ and the outputs are labeled } -\theta_1, \dots, -\theta_m. \text{ The diagram is surrounded by a network of lines and nodes, including red nodes labeled } \pi \text{ and green nodes. The entire structure is enclosed in a large box with a double line border. The equation is labeled (72) on the right.} \end{array} \quad (72)$$

Such a diagram can then be simulated using standard ZX-based techniques, such as the one given in Ref. [35].

5.4 Quantised volume calculations in loop quantum gravity

Though a full introduction to loop quantum gravity (LQG) is beyond the scope of this paper, the key elements for our purposes are that it proposes that space itself is quantised and that properties like area and volume can be obtained from the appropriate area and volume operators. In LQG the Hilbert space of space itself is given by the sum over all possible 4-valent spin networks. Each particular spin network can be viewed as a geometric object by looking at its dual, which gives us an image of tessellating tetrahedra, each edge corresponding to a face, each intertwiner corresponding to a volume, with intertwiners sharing edges indicating a shared face of two tetrahedra. These can all be represented in ZX form as was first discussed in Ref. [21]. Given this perspective, LQG tells us that the area of a face is given simply by the operator:

$$\hat{A}_l = 8\pi\gamma \frac{\hbar G}{c^3} \sqrt{\vec{J}_l^2} \quad (73)$$

which has the following eigenstates:

$$\hat{A}_l |\Gamma\rangle = 8\pi\gamma \frac{\hbar G}{c^3} \sqrt{j_l(j_l + 1)} |\Gamma\rangle \quad (74)$$

We can see that, up to some fundamental constants, the area operator for a face of a quantised tetrahedron (given by a spin-network's edge) is directly proportional to the magnitude of the total angular momentum operator.

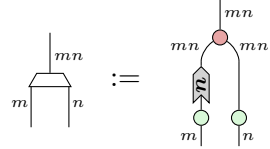
The case of volume is more challenging as the operator is not diagonal in this basis. The volume operator is given by:

$$\hat{V} = \frac{\sqrt{2}}{V} (8\pi G \hbar \gamma)^{3/2} \sqrt{|\vec{J}_1 \cdot (\vec{J}_2 \times \vec{J}_3)|} \quad (75)$$

We can see the core calculational element here can be isolated as:

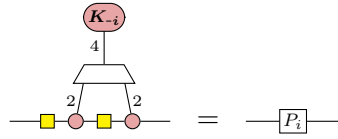
$$V^2 \propto \tilde{V}^2 = \epsilon_{ijk} J_x^i J_y^j J_z^k = \frac{1}{8} \epsilon_{ijk} \sigma_x^i \sigma_y^j \sigma_z^k \quad (76)$$

We can construct this operation in the Spin-ZX calculus by making use of dimension splitters:



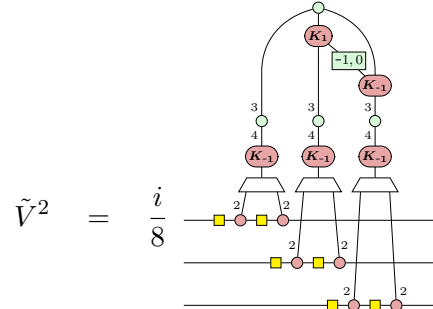
$$:= \quad (77)$$

and writing the controlled versions of the Pauli matrices as follows:



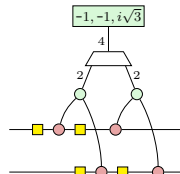
$$= \quad (78)$$

where P_i for $i = 0, 1, 2, 3$ is I , σ_x , σ_y and σ_z respectively. With this, we construct $\tilde{V}^2$:



$$= \quad (79)$$

where the yellow box is the usual Hadamard gate. An equivalent of this diagram was used in ZXH to demonstrate that the state corresponded to the minimum spin- $\frac{1}{2}$ volume. It can be shown that the eigenstate of this volume is given by the sum $\iota_0 + i\sqrt{3}\iota_1$, where ι_0 and ι_1 are the respective recoupling possibilities for two spin- $\frac{1}{2}$, namely spin-0 and spin-1 (see [38] for more details). Diagrammatically, we can represent this as follows:

$$\left| \iota_0 + i\sqrt{3}\iota_1 \right\rangle = \frac{i}{\sqrt{3}} \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & -e^{i\frac{\pi}{3}} & e^{i\frac{2\pi}{3}} & 0 \\ 0 & e^{i\frac{2\pi}{3}} & -e^{i\frac{\pi}{3}} & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix} = \frac{1 - \frac{i}{\sqrt{3}}}{4} \quad (80)$$


where we use a slight abuse of notation by denoting this map as a state. The question then becomes about witnessing the action of this operator on the volume to see if it is an eigenvector and retrieve the eigenvalue. We can see this for the minimal 4 spin-1/2 volume as follows

Proposition 21. *The state $|\iota_0 + i\sqrt{3}\iota_1\rangle$ is an eigenvector of the operator $I \otimes \tilde{V}^2$ with eigenvalue $\frac{-\sqrt{3}}{4}$.*

$$= \frac{-\sqrt{3}}{4} \quad (81)$$

Comparing with the literature [49] we see we indeed obtain the correct minimal volume (recalling from equation (75) that we are only interested in the modulus so the volume is not imaginary). This calculation is of broader interest being one of the first examples of a diagrammatic eigenvector calculation. The determination of the eigenvectors of diagrammatic operators is a broad question and one that, though unaddressed further here, merits further study.

6 Conclusion and discussion

In this paper, we have introduced the Spin-ZX calculus as a formal diagrammatic language tailored for the representation theory of $SU(2)$, elevating Penrose's original diagrams into a full language for quantum information and spin physics. Our work demonstrates the versatility of the Spin-ZX calculus across multiple domains. In permutational quantum computing, we have demonstrated how transition amplitudes can be derived entirely through diagrammatic reasoning, simplifying calculations that traditionally required a knowledge of Racah transforms. In condensed matter physics, the calculus offers a more natural and streamlined approach to analysing AKLT states, extending beyond previous qubit-based diagrammatic techniques. In quantum machine learning, we have illustrated how parametrised quantum circuits for spin systems can be analysed, providing insights into the variance of $SU(2)$ equivariant ansätze. In quantum gravity, our techniques simplify the representation of the volume operator in loop quantum gravity, enabling the calculation of minimal quantised volumes through purely diagrammatic methods.

The Spin-ZX calculus not only simplifies the representation and manipulation of $SU(2)$ systems but also offers new ways to look at the foundational relationships within representation theory. By providing a formal diagrammatic language, we pave the way for the development of new algorithms and methodologies in theoretical physics. This work represents a significant step towards integrating quantum information science with the fundamental symmetries of nature, and we anticipate that it will inspire further research and applications in both quantum computing and broader theoretical physics. Future work will continue the effort to diagrammatise physics by the creation of calculi for representations of $SU(N)$, in particular $SU(3)$, which will allow for the natural treatment of systems of interest to particle physicists and allow for the diagrammatic description of the standard model. It will also look to apply the Spin-ZX calculus to more $SU(2)$ systems such as more 2D lattices, spin glasses, $SU(2)$ resistant decoherent free subspaces, and error correction. In addition it will seek to create new methods for devising algorithms of interest to physicists and computer scientists alike by leveraging this new perspective

on quantum information in spin systems.

In general we note that quantum information is something of a curiosity in terms of foundational physics. In other domains of physics pertaining to the basic structure of nature, symmetries reign supreme – indeed the definitions of what things in themselves *are* tend to be expressed in terms of symmetries. Particles from photons and electrons to quarks and anyons are all tied to various (representations of) symmetry groups. In the case of quantum information however the dominant theoretical framework is registers of qubits, which are ultimately an engineering principle. We often want quantum algorithms to analyse natural physical systems in which symmetry groups are the characteristic feature. As such, we should devise tools where the presentation of these symmetries is not just possible but natural.

Acknowledgements

We would like to thank Nathan Fitzpatrick, Tuomas Laakkonen, and Richie Yeung for their feedback on the paper and suggestions for improvement. RS is supported by the Clarendon Fund Scholarship. RDPE acknowledges financial support from a Blaumann Foundation grant and useful conversations with A. Grushin. LY acknowledges funding from the Google PhD Fellowship and from the Dowling Postdoctoral Fellowship. BP is supported by the Engineering and Physical Sciences Research Council grant number EP/Z002230/1, ‘(De)constructing quantum software (DeQS)’.

References

- [1] Samson Abramsky & Bob Coecke (2004): *A Categorical Semantics of Quantum Protocols*. In: *19th IEEE Symposium on Logic in Computer Science (LICS 2004)*, IEEE Computer Society, turku, finland, pp. 415–425, doi:10.1109/LICS.2004.1319636.
- [2] Samson Abramsky & Bob Coecke (2008): *Categorical Quantum Mechanics*. In: *Handbook of Quantum Logic and Quantum Structures*, 2, Elsevier, pp. 261–325. arXiv:0808.1023.
- [3] Ian Affleck, Tom Kennedy, Elliott H. Lieb & Hal Tasaki (1987): *Rigorous Results on Valence-Bond Ground States in Antiferromagnets*. *Physical Review Letters* 59(7), pp. 799–802, doi:10.1103/PhysRevLett.59.799.
- [4] Ian Affleck, Tom Kennedy, Elliott H. Lieb & Hal Tasaki (1988): *Valence Bond Ground States in Isotropic Quantum Antiferromagnets*. *Communications in Mathematical Physics* 115(3), pp. 477–528, doi:10.1007/BF01218021.
- [5] Niel de Beaudrap, Xiaoning Bian & Quanlong Wang (2020): *Fast and effective techniques for t-count reduction via spider nest identities*. In Steven T. Flammia, editor: *15th conference on the theory of quantum computation, communication and cryptography (TQC 2020)*, *Leibniz international proceedings in informatics (lipics)* 158, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, p. 11:1–11:23, doi:10.4230/LIPIcs.TQC.2020.11.
- [6] Filippo Bonchi, Paweł Sobociński & Fabio Zanasi (2017): *Interacting Hopf Algebras*. *Journal of Pure and Applied Algebra* 221(1), pp. 144–184, doi:10.1016/j.jpaa.2016.06.002. arXiv:1403.7048.
- [7] Robert I. Booth & Titouan Carette (2022): *Complete ZX-calculi for the Stabiliser Fragment in Odd Prime Dimensions*. In Stefan Szeider, Robert Ganian & Alexandra Silva, editors: *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, *Leibniz International Proceedings in Informatics (Lipics)* 241, Schloss Dagstuhl – Leibniz-Zentrum für In-

- formatik, Dagstuhl, Germany, pp. 24:1–24:15, doi:10.4230/LIPIcs.MFCS.2022.24. Available at <https://drops.dagstuhl.de/opus/volltexte/2022/16822>.
- [8] Robert I. Booth, Titouan Carette & Cole Comfort (2024): *Complete Equational Theories for Classical and Quantum Gaussian Relations*. arXiv:2403.10479.
 - [9] Titouan Carette, Dominic Horsman & Simon Perdrix (2019): *SZX-Calculus: Scalable Graphical Quantum Reasoning*. In Peter Rossmanith, Pinar Heggenes & Joost-Pieter Katoen, editors: *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 138, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 55:1–55:15, doi:10.4230/LIPIcs.MFCS.2019.55.
 - [10] Nicholas Chancellor, Aleks Kissinger, Stefan Zohren, Joschka Roffe & Dominic Horsman (2023): *Graphical Structures for Design and Verification of Quantum Error Correction*. *Quantum Science and Technology* 8(4), p. 045028, doi:10.1088/2058-9565/acf157.
 - [11] Bob Coecke & Ross Duncan (2008): *Interacting Quantum Observables*. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfssdóttir & Igor Walukiewicz, editors: *Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 298–310, doi:10.1007/978-3-540-70583-3_25. Available at <http://personal.strath.ac.uk/ross.duncan/papers/iquo-icalp.pdf>.
 - [12] Bob Coecke & Ross Duncan (2011): *Interacting Quantum Observables: Categorical Algebra and Diagrammatics*. *New Journal of Physics* 13(4), p. 043016, doi:10.1088/1367-2630/13/4/043016.
 - [13] Bob Coecke, Ross Duncan, Aleks Kissinger & Quanlong Wang (2016): *Generalised Compositional Theories and Diagrammatic Reasoning*. In Giulio Chiribella & Robert W. Spekkens, editors: *Quantum Theory: Informational Foundations and Foils*, Fundamental Theories of Physics, Springer Netherlands, Dordrecht, pp. 309–366, doi:10.1007/978-94-017-7303-4_10. arXiv:1506.03632.
 - [14] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons & Seyon Sivarajah (2020): *Phase Gadget Synthesis for Shallow Circuits*. In Bob Coecke & Matthew Leifer, editors: *Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019, Electronic Proceedings in Theoretical Computer Science* 318, Open Publishing Association, pp. 213–228, doi:10.4204/EPTCS.318.13.
 - [15] Niel de Beaudrap, Ross Duncan, Dominic Horsman & Simon Perdrix (2020): *Pauli Fusion: A Computational Model to Realise Quantum Transformations from ZX Terms*. In Bob Coecke & Matthew Leifer, editors: *Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019, Electronic Proceedings in Theoretical Computer Science* 318, Open Publishing Association, pp. 85–105, doi:10.4204/EPTCS.318.6.
 - [16] Niel de Beaudrap & Dominic Horsman (2020): *The ZX Calculus Is a Language for Surface Code Lattice Surgery*. *Quantum* 4, p. 218, doi:10.22331/q-2020-01-09-218.
 - [17] Niel de Beaudrap, Aleks Kissinger & John van de Wetering (2022): *Circuit Extraction for ZX-Diagrams Can Be #P-Hard*. In Mikołaj Bojańczyk, Emanuela Merelli & David P. Woodruff, editors: *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 229, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 119:1–119:19, doi:10.4230/LIPIcs.ICALP.2022.119.
 - [18] Giovanni de Felice, Razin A. Shaikh, Boldizsár Poór, Lia Yeh, Quanlong Wang & Bob Coecke (2023): *Light-Matter Interaction in the ZXW Calculus*. In Shane Mansfield, Benoit Valiron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023, Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 20–46, doi:10.4204/EPTCS.384.2.
 - [19] Ross Duncan, Aleks Kissinger, Simon Perdrix & John van de Wetering (2020): *Graph-Theoretic Simplification of Quantum Circuits with the ZX-calculus*. *Quantum* 4, p. 279, doi:10.22331/q-2020-06-04-279.

- [20] Ross Duncan & Simon Perdrix (2010): *Rewriting Measurement-Based Quantum Computations with Generalised Flow*. In Samson Abramsky, Cyril Gavioille, Claude Kirchner, Friedhelm Meyer auf der Heide & Paul G. Spirakis, editors: *Automata, Languages and Programming*, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, pp. 285–296, doi:10.1007/978-3-642-14162-1_24. Available at <http://personal.strath.ac.uk/ross.duncan/papers/gflow.pdf>.
- [21] Richard East (2022): *Physique formelle de spin diagrammatique (Don't fret it's in English)*. Ph.D. thesis, Université Grenoble Alpes [2020-..]. Available at <https://theses.hal.science/tel-03719945>.
- [22] Richard D. P. East, Guillermo Alonso-Linaje & Chae-Yeun Park (2023): *All You Need Is Spin: $SU(2)$ Equivariant Variational Quantum Circuits Based on Spin Networks*. arXiv:2309.07250.
- [23] Richard D. P. East, Pierre Martin-Dussaud & John Van de Wetering (2022): *Spin-Networks in the ZX-calculus*. arXiv:2111.03114.
- [24] Richard D.P. East, John van de Wetering, Nicholas Chancellor & Adolfo G. Grushin (2022): *AKLT-States as ZX-Diagrams: Diagrammatic Reasoning for Quantum States*. PRX Quantum 3(1), p. 010302, doi:10.1103/PRXQuantum.3.010302.
- [25] Craig Gidney & Austin G. Fowler (2019): *Efficient magic state factories with a catalyzed $|CCZ\rangle$ to $2|T\rangle$ transformation*. Quantum 3, p. 135, doi:10.22331/q-2019-04-30-135.
- [26] Jean-Yves Girard (1987): *Linear Logic*. Theoretical Computer Science 50(1), pp. 1–101, doi:10.1016/0304-3975(87)90045-4.
- [27] Michael Hanks, Marta P. Estarellas, William J. Munro & Kae Nemoto (2020): *Effective Compression of Quantum Braided Circuits Aided by ZX-Calculus*. Physical Review X 10(4), p. 041030, doi:10.1103/PhysRevX.10.041030.
- [28] Jiaxin Huang, Sarah Meng Li, Lia Yeh, Aleks Kissinger, Michele Mosca & Michael Vasmer (2023): *Graphical CSS Code Transformation Using ZX Calculus*. In Shane Mansfield, Benoit Valiron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 1–19, doi:10.4204/EPTCS.384.1.
- [29] Gérard Huet (1980): *Confluent Reductions: Abstract Properties and Applications to Term Rewriting Systems: Abstract Properties and Applications to Term Rewriting Systems*. J. ACM 27(4), pp. 797–821, doi:10.1145/322217.322230.
- [30] Stephen P. Jordan (2010): *Permutational Quantum Computing*. Quantum Info. Comput. 10(5), pp. 470–497. arXiv:0906.2508.
- [31] André Joyal & Ross Street (1991): *The Geometry of Tensor Calculus, I*. Advances in Mathematics 88(1), pp. 55–112, doi:10.1016/0001-8708(91)90003-P.
- [32] Aleks Kissinger (2022): *Phase-free ZX diagrams are CSS codes (...or how to graphically grok the surface code)*. arXiv:2204.14038.
- [33] Aleks Kissinger & John van de Wetering (2020): *Reducing T-count with the ZX-calculus*. Physical Review A 102(2), p. 022406, doi:10.1103/PhysRevA.102.022406. arXiv:1903.10477.
- [34] Aleks Kissinger & John van de Wetering (2019): *Universal MBQC with Generalised Parity-Phase Interactions and Pauli Measurements*. Quantum 3, p. 134, doi:10.22331/q-2019-04-26-134.
- [35] Mark Koch, Richie Yeung & Quanlong Wang (2024): *Contraction of ZX Diagrams with Triangles via Stabiliser Decompositions*. Physica Scripta 99(10), p. 105122, doi:10.1088/1402-4896/ad6fd8. arXiv:2307.01803.
- [36] Seth A. Major (1999): *A Spin Network Primer*. American Journal of Physics 67(11), pp. 972–980, doi:10.1119/1.19175. arXiv:gr-qc/9905020.
- [37] Ilkka Mäkinen (2019): *Introduction to $SU(2)$ Recoupling Theory and Graphical Methods for Loop Quantum Gravity*. arXiv:1910.06821.

- [38] Pierre Martin-Dussaud (2019): *A Primer of Group Theory for Loop Quantum Gravity and Spin-Foams*. *General Relativity and Gravitation* 51(9), p. 110, doi:10.1007/s10714-019-2583-5. arXiv:1902.08439.
- [39] Annalisa Marzuoli & Mario Rasetti (2005): *Computing Spin Networks*. *Annals of Physics* 318(2), pp. 345–407, doi:10.1016/j.aop.2005.01.005. arXiv:quant-ph/0410105.
- [40] Hironari Nagayoshi, Warit Asavanant, Ryuhoh Ide, Kosuke Fukui, Atsushi Sakaguchi, Jun-ichi Yoshikawa, Nicolas C. Menicucci & Akira Furusawa (2024): *ZX Graphical Calculus for Continuous-Variable Quantum Processes*. arXiv:2405.07246.
- [41] Kang Feng Ng & Quanlong Wang (2017): *A Universal Completion of the ZX-calculus*. arXiv:1706.09877.
- [42] Roger Penrose (1971): *Applications of Negative Dimensional Tensors*. *Combinatorial mathematics and its applications*, pp. 221–244.
- [43] Elisha Peterson (2006): *Trace Diagrams, Representations, and Low-Dimensional Topology*. Ph.D. thesis, University of Maryland. Available at <http://hdl.handle.net/1903/3500>.
- [44] Boldizsár Poór, Robert I. Booth, Titouan Carette, John van de Wetering & Lia Yeh (2023): *The Qupit Stabiliser ZX-travaganza: Simplified Axioms, Normal Forms and Graph-Theoretic Simplification*. In Shane Mansfield, Benoit Valiron & Vladimir Zamdzhiev, editors: *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, *Electronic Proceedings in Theoretical Computer Science* 384, Open Publishing Association, pp. 220–264, doi:10.4204/EPTCS.384.13.
- [45] Boldizsár Poór, Razin A. Shaikh & Quanlong Wang (2024): *ZX-calculus Is Complete for Finite-Dimensional Hilbert Spaces*. arXiv:2405.10896.
- [46] Boldizsár Poór, Quanlong Wang, Razin A. Shaikh, Lia Yeh, Richie Yeung & Bob Coecke (2023): *Completeness for Arbitrary Finite Dimensions of ZXW-calculus, a Unifying Calculus*. In: *2023 38th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, Boston, MA, USA, pp. 1–14, doi:10.1109/LICS56636.2023.10175672. arXiv:2302.12135.
- [47] Benjamin Rodatz, Boldizsár Poór & Aleks Kissinger (2024): *Floquetifying Stabiliser Codes with Distance-Preserving Rewrites*. arXiv:2410.17240.
- [48] Benjamin Rodatz, Boldizsár Poór & Aleks Kissinger (2025): *Fault Tolerance by Construction*. arXiv:2506.17181.
- [49] Carlo Rovelli & Francesca Vidotto (2014): *Covariant Loop Quantum Gravity: An Elementary Introduction to Quantum Gravity and Spinfoam Theory*. Cambridge University Press, Cambridge, doi:10.1017/CB09781107706910.
- [50] Razin A. Shaikh, Quanlong Wang & Richie Yeung (2023): *How to Sum and Exponentiate Hamiltonians in ZXW Calculus*. In Stefano Gogioso & Matty Hoban, editors: *Proceedings 19th International Conference on Quantum Physics and Logic, Wolfson College, Oxford, UK, 27 June - 1 July 2022*, *Electronic Proceedings in Theoretical Computer Science* 394, Open Publishing Association, pp. 236–261, doi:10.4204/EPTCS.394.14.
- [51] Razin A. Shaikh, Lia Yeh & Stefano Gogioso (2024): *The Focked-up ZX Calculus: Picturing Continuous-Variable Quantum Computation*. arXiv:2406.02905.
- [52] Bruce W. Shore & Donald Howard Menzel (1967): *Principles of Atomic Spectra*. Wiley, New York.
- [53] Daniel Bochen Tan, Murphy Yuezhen Niu & Craig Gidney (2024): *A SAT Scalpel for Lattice Surgery: Representation and Synthesis of Subroutines for Surface-Code Fault-Tolerant Quantum Computing*. In: *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture (ISCA)*, pp. 325–339, doi:10.1109/ISCA59077.2024.00032. arXiv:2404.18369.
- [54] Renaud Vilmart (2019): *A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics*. In: *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–10, doi:10.1109/LICS.2019.8785765. arXiv:1812.09114.

- [55] Quanlong Wang (2022): *Qufinite ZX-calculus: A Unified Framework of Qudit ZX-calculi*. arXiv:2104.06429.
- [56] Quanlong Wang, Boldizsár Poór & Razin A. Shaikh (2024): *Completeness of Qufinite ZXW Calculus, a Graphical Language for Finite-Dimensional Quantum Theory*. arXiv:2309.13014.
- [57] Quanlong Wang, Richie Yeung & Mark Koch (2024): *Differentiating and Integrating ZX Diagrams with Applications to Quantum Machine Learning*. Quantum 8, p. 1491, doi:10.22331/q-2024-10-04-1491.
- [58] Adolfo P. Yutsis, I. B. Levinson & V. V. Vanagas (1962): *Mathematical Apparatus of the Theory of Angular Momentum*. Israel Program for Scientific Translations, Jerusalem. Available at http://archive.org/details/nasa_techdoc_19630001624.
- [59] Chen Zhao & Xiao-Shan Gao (2021): *Analyzing the Barren Plateau Phenomenon in Training Quantum Neural Networks with the ZX-calculus*. Quantum 5, p. 466, doi:10.22331/q-2021-06-04-466.

A Traditional Penrose diagrams and ZX

Penrose diagrams (as opposed to the Spin-ZX calculus we introduce here) are a diagrammatic notation introduced by Roger Penrose to reason about spins. A form of these diagrams decomposed into strands of spin- $\frac{1}{2}$ i.e. qubits is of particular interest to us - often termed the binor calculus [38]. It is this that we alter and raise to the level of a diagrammatic language. Before we can do this, we must look at its original form and compare with the ZX rules discussed above. To do this, we take the route presented in [23] and direct the reader seeking deeper explanation to [36].

In the binor calculus, the identity over $\mathbb{C}^2$ is a single strand

$$\Big| \cong |0\rangle\langle 0| + |1\rangle\langle 1|. \quad (82)$$

The free legs carry implicit labels of copies of $\mathbb{C}^2$. There is also a duality between up and down. These are given by the ‘cups’ and ‘caps’. The cup stands for

$$\cup \cong i \langle 01| - i \langle 10|. \quad (83)$$

The cap is

$$\cap \cong i |01\rangle - i |10\rangle. \quad (84)$$

With these definitions, the diagrams are then well-behaved under deformation:

$$\sim = \Big| \quad (85)$$

Finally, the crossing is the regular swap, but with a global minus sign:

$$\times = -|00\rangle\langle 00| - |10\rangle\langle 01| - |01\rangle\langle 10| - |11\rangle\langle 11|. \quad (86)$$

These rules guarantee planar isotopy, i.e. diagrams can be continuously deformed while preserving their interpretation as linear maps.

The cap and cup as Penrose diagrams can be translated into ZX diagrams as follows:



where for convenience, we ignore the imaginary number i during the translation. Similarly, though more importantly, the swap Penrose diagram is translated as the normal ZX swap without introducing the minus sign - below this has implication for how one structures the projectors to higher irreps.

Fundamental equations Binor calculus has two core equations that can be deduced from the definitions above.

$$\bigcirc = -2 \quad (87)$$

$$\bigg| \bigg| + \begin{array}{c} \diagup \diagdown \\ \diagdown \diagup \end{array} + \begin{array}{c} \diagdown \diagup \\ \diagup \diagdown \end{array} = 0 \quad (88)$$

The value of the loop in the first equation secretly gives the ‘dimension’ of the tensor calculus. Here it is -2 : the ‘2’ explains the name ‘binor’, and the minus sign explains the title of the original article, *Negative Dimensional Tensors* by Penrose [42]. The second equation is known under the name of ‘skein relation’ or ‘binor identity’. A glance at the ZX generators shows a difference between Equation (87) and our Spin-ZX calculus is the minus sign. The ‘skein relation’, however, does not exist in the same form (though of course still holds) as it relies on the negative sign incurred by the swap. This difference rears its head more seriously when we construct the symmetric projector below.

(Anti-)symmetriser A key difference between the ZX formalism and Penrose’s original diagrams is in how spaces are symmetrised. In ZX we have seen the operator presented above in Equation (34) however the same mapping in Penrose’s original diagrams to get the symmetrised space of multiple $\text{spin-}\frac{1}{2}$ ’s to give $\text{spin-}j$ irrep is given by anti-symmetrising the strands in binor calculus:

$$\left| \begin{array}{c} \cdot \cdot \cdot \\ \cdot \cdot \cdot \\ \cdot \cdot \cdot \end{array} \right| \stackrel{\text{def}}{=} \frac{1}{(2j)!} \sum_{\sigma \in \mathfrak{S}_{2j}} (-1)^{|\sigma|} \left| \begin{array}{c} \cdot \cdot \cdot \\ \sigma \end{array} \right| \quad (89)$$

where $|\sigma|$ is the parity of σ and the σ -labelled box represents the corresponding permutation of the $2j$ strands from one side to the other. Although this looks like an anti-symmetrisation (since we have the $(-1)^{|\sigma|}$ term), the operator is actually a projector from $\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2$ to $\mathcal{S}_{2j}(\mathbb{C}^2 \otimes \cdots \otimes \mathbb{C}^2)$, because of the minus sign in Equation (86). So we have

$$\begin{array}{c} \overbrace{0 \dots 1}^{j+m \text{ times } 0} \\ \left| \begin{array}{c} 2j \\ \cdot \cdot \cdot \end{array} \right| \\ \hline \end{array} = \mathcal{S}_{2j}(\underbrace{|0\rangle \dots |1\rangle}_{j+m \text{ times } 0}) = \frac{1}{\sqrt{\binom{2j}{j-m}}} |j; m\rangle. \quad (90)$$

Taking the trace of the symmetrised space (i.e. making a loop) gives the value

$$j - \bigcirc = (-1)^{2j} (2j + 1), \quad (91)$$

which is the dimension of the space up to a phase.

Invariant tensors. Now, let us draw in binor calculus, the analogue of the 3-valent vertex, i.e. the $3jm$ -symbol. The essential property to be noticed is the invariance of the cup under the action of $SU(2)$, i.e. for any $u \in SU(2)$:

$$\begin{array}{c} \text{cup with two boxes labeled } u \end{array} \cong i u \cdot (|01\rangle - |10\rangle) = i (|01\rangle - |10\rangle) \cong \begin{array}{c} \text{cup} \end{array} \quad (92)$$

Then the diagram

$$\begin{array}{c} \text{Diagram with three cups labeled } 2j_1, 2j_2, 2j_3 \end{array} \quad (93)$$

depicts a vector that belongs to $\text{Inv}_{SU(2)}(\mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2} \otimes \mathcal{H}_{j_3})$, so it is proportional to $|j_1, j_2, j_3\rangle$. This should be interpreted as a kind of ‘railroad switch’, where the fundamental wires within the three symmetrised bundles redistribute between themselves. Because we are dealing with symmetrised spaces, we only care about how many wires go from each bundle to the other bundle. It turns out that there is then actually only one way in which to connect the wires, when it is not impossible. The Clebsch-Gordan conditions (44) precisely state when such a recoupling is possible.

Plugging the vectors of Equation (90), one gets

$$\begin{array}{c} \text{Diagram with three cups labeled } 2j_1, 2j_2, 2j_3 \end{array} \quad (94)$$

$$= N(j_1, j_2, j_3) i^{j_1+j_2+j_3} (-1)^{j_1-j_2+j_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \frac{1}{\sqrt{\binom{2j_1}{j_1-m_1} \binom{2j_2}{j_2-m_2} \binom{2j_3}{j_3-m_3}}}$$

An important case is when one of the strands has spin 0. The spin-0 strand is then represented with a dashed line (no arrow needed):

$$\begin{array}{c} j_1 m_1 \quad j_3 m_3 \\ \diagdown \quad \vdots \quad \diagup \\ \text{---} \end{array} = \frac{(-1)^{j_1+m_1}}{\sqrt{2j_1+1}} \delta_{m_1,-m_3} \delta_{j_1,j_3}. \quad (99)$$

We can then graphically define the two basic operations of multiplication and summation. Multiplication is implemented simply by juxtaposition of diagrams:

$$\begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j_3 m_3 \\ \diagdown \quad \diagup \quad \diagup \\ \text{---} \end{array} \quad \begin{array}{c} j_4 m_4 \quad j_5 m_5 \quad j_6 m_6 \\ \diagdown \quad \diagup \quad \diagup \\ \text{---} \end{array} = \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \begin{pmatrix} j_4 & j_5 & j_6 \\ m_4 & m_5 & m_6 \end{pmatrix} \quad (100)$$

Then, the gluing of two external wires with the same label j , but opposite directions, defines the sum over m (from $-j$ to j), with the additional factor $(-1)^{j-m}$ in the summand, like:

$$\begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j_3 m_3 \quad j_4 m_4 \\ \diagdown \quad \diagup \quad \diagdown \quad \diagup \\ \text{---} \end{array} = \sum_{m=-j}^j (-1)^{j-m} \begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j m \\ \diagdown \quad \diagup \quad \diagup \\ \text{---} \end{array} \begin{array}{c} j m \quad j_3 m_3 \quad j_4 m_4 \\ \diagdown \quad \diagup \quad \diagup \\ \text{---} \end{array} \quad (101)$$

On the RHS, we recognise the definition of the $4jm$ -symbol:

$$\begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j_3 m_3 \quad j_4 m_4 \\ \diagdown \quad \diagup \quad \diagdown \quad \diagup \\ \text{---} \end{array} = \begin{pmatrix} j_1 & j_2 & j_3 & j_4 \\ m_1 & m_2 & m_3 & m_4 \end{pmatrix}^{(j)} \quad (102)$$

The wire between the two nodes, whose magnetic index is summed over, is called an *internal* wire. Reversing the arrow of an internal wire gives an overall phase:

$$\begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j_3 m_3 \quad j_4 m_4 \\ \diagdown \quad \diagup \quad \diagdown \quad \diagup \\ \text{---} \end{array} = (-1)^{2j} \begin{array}{c} j_1 m_1 \quad j_2 m_2 \quad j_3 m_3 \quad j_4 m_4 \\ \diagdown \quad \diagup \quad \diagup \quad \diagdown \\ \text{---} \end{array}.$$

For convenience it is also helpful to consider a single wire with an arrow on it as an element of the graphical calculus:

$$\begin{array}{c} j_1 m_1 \\ \uparrow \\ j_2 m_2 \end{array} = (-1)^{j_2-m_2} \delta_{j_1 j_2} \delta_{m_1 m_2} \quad \text{or} \quad \begin{array}{c} m \\ \uparrow \\ j \\ \uparrow \\ n \end{array} = (-1)^{j-n} \delta_{mn}. \quad (103)$$

C Some General Lemmas

Lemma 1. [55]

$$\begin{array}{c} \text{green circle} \\ \vdots \\ \text{red circle} \end{array} = \begin{array}{c} \text{green circle} \\ \text{red circle} \end{array} \quad \begin{array}{c} \text{red circle} \\ \vdots \\ \text{green circle} \end{array} = \begin{array}{c} \text{red circle} \\ \text{green circle} \end{array} \quad (\text{Hopf})$$

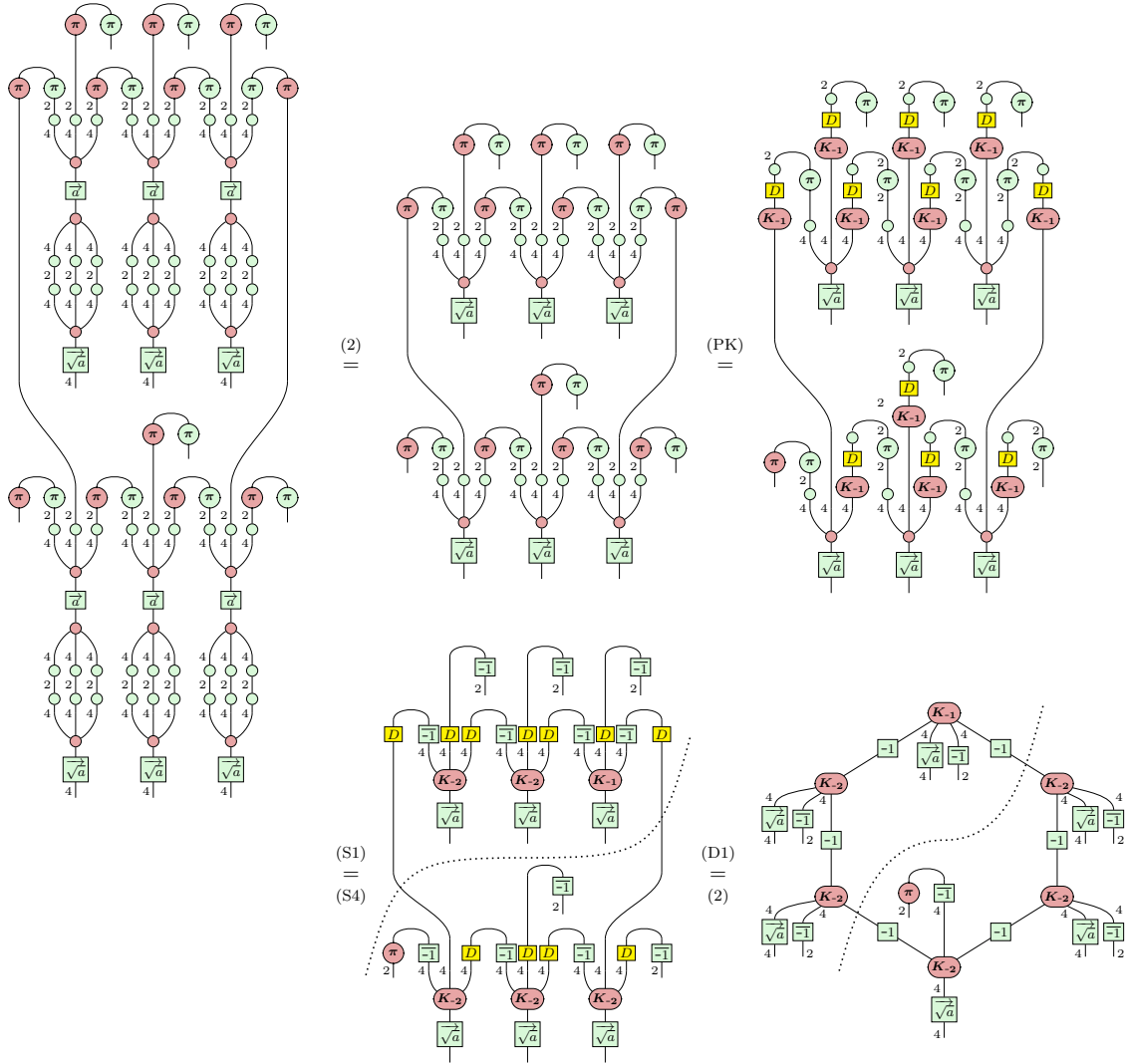
Lemma 2.

$$\begin{array}{ccc} \text{diagram 1} = \text{diagram 2} = \text{diagram 3} & \text{diagram 4} = \text{diagram 5} \\ \text{diagram 6} = \text{diagram 7} = \text{diagram 8} & \text{diagram 9} = \text{diagram 10} \end{array}$$

Proposition 20. *We can transform from the Penrose-style diagram to the Yutsis-style diagram by composing the former with isometries (32) and simplifying it to get the Honeycomb shape.*

$$\begin{array}{ccc} \text{Penrose-style diagram} & \rightsquigarrow & \text{Yutsis-style diagram} \end{array} \quad (66)$$

Proof.



On the last line, the dotted line indicates how the nodes are being rearranged in the last step. $\square$

D Diagrammatic Proofs of Symmetriser Properties

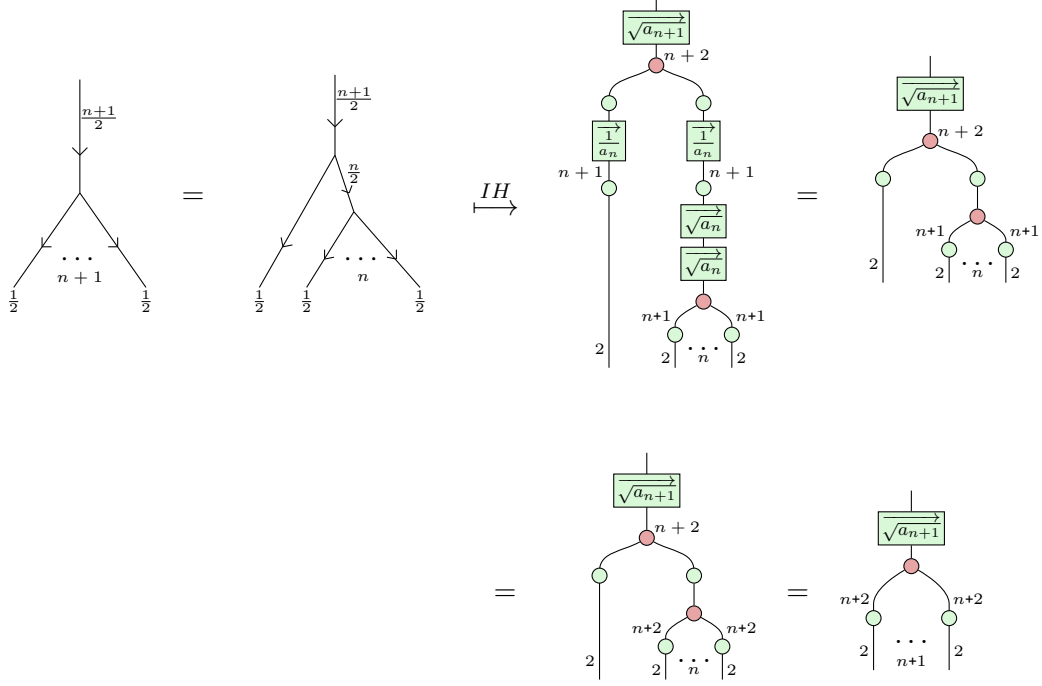
This section provides proofs of different properties of the symmetriser. Note that some proofs are stated as equalities between Yutis diagrams, while the proofs are given as a derivation between their ZX representations. In such cases, we assume an implicit mapping from the given Yutis diagrams to their ZX representations.

Proposition 1. *This isometry is given by the following diagram:*

$$\begin{array}{c} \frac{n}{2} \\ \downarrow \\ \begin{array}{c} \diagup \quad \dots \quad \diagdown \\ \frac{1}{2} \quad \quad \quad \frac{1}{2} \end{array} \end{array} \quad \mapsto \quad \begin{array}{c} \boxed{\sqrt{a_n}} \\ \downarrow \\ \begin{array}{c} \diagup \quad \diagdown \\ n+1 \quad n+1 \\ \vdots \\ 2 \quad \quad \quad 2 \end{array} \end{array} \quad (32)$$

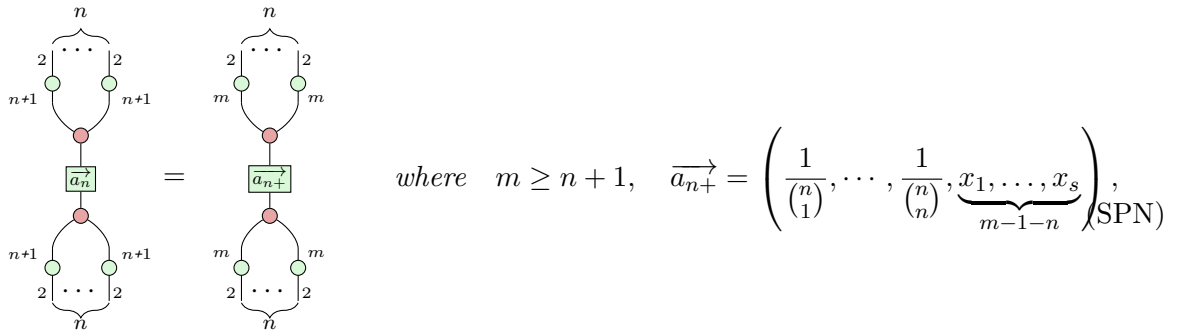
where $\overrightarrow{\sqrt{a_n}} = \left(\frac{1}{\sqrt{\binom{n}{1}}}, \dots, \frac{1}{\sqrt{\binom{n}{k}}}, \dots, \frac{1}{\sqrt{\binom{n}{n}}} \right)$.

Proof. We prove this by induction.



□

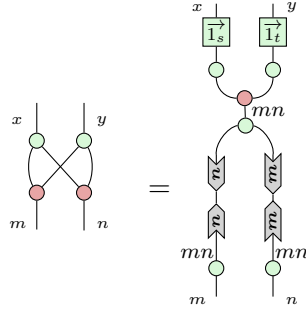
Lemma 3.



and x_i are arbitrary complex numbers.

Proof. It follows from the Rules (DA) and (S1). □

Lemma 4.

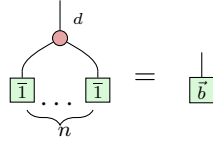


where $s = \min\{x, m, n\}$, $t = \min\{y, m, n\}$.

Proof. It follows from the Rules (S1), (DZX) and (B2). □

To show that $\mathcal{S}_n$ is a projector, we need the following lemmas:

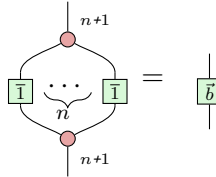
Lemma 5.



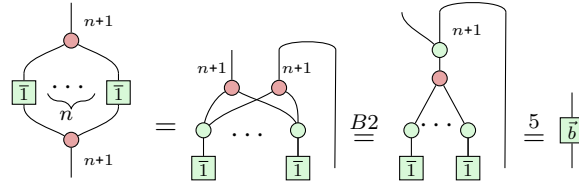
where $d \geq 2, n < d, \vec{b} = ((\binom{n}{1}), \dots, (\binom{n}{k}), \dots, (\binom{n}{n}), \underbrace{0, \dots, 0}_{d-1-n})$.

Proof. It follows from the rule (PA). □

Lemma 6.

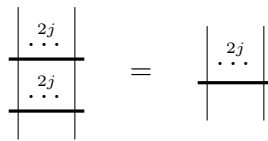


Proof.



Now, we can show that the symmetriser is a projector:.

Proposition 4 (Projector). $\mathcal{S}_n^2 = \mathcal{S}_n$, that is,



Proof.

$$\begin{array}{c}
 \mathcal{S}_n^2 = \text{Diagram 1} \stackrel{(S1)}{=} \text{Diagram 2} \stackrel{(6)}{=} \text{Diagram 3} \stackrel{(S1)}{=} \text{Diagram 4} = \mathcal{S}_n
 \end{array}$$

The diagrams represent tensor network transformations. Diagram 1 shows a vertical stack of three nodes, each with two inputs and two outputs, connected by red dots. Diagram 2 shows a similar stack but with a different internal structure. Diagram 3 shows a stack of three nodes with a different internal structure. Diagram 4 shows a stack of three nodes with a different internal structure. The transformations are labeled (S1), (6), and (S1).

□

The next property we show is *invariance*, that is, that any 2×2 matrix commutes with the symmetriser. To do this, we first derive some lemmas:

Lemma 7.

$$\begin{array}{c}
 \text{Diagram 1} = \text{Diagram 2}
 \end{array}$$

Diagram 1 shows a node with two inputs and two outputs, connected by red dots. Diagram 2 shows a similar node but with a different internal structure. The transformation is labeled (S1).

where $\vec{a}^n = (a, \dots, a^k, \dots, a^n, \underbrace{0, \dots, 0}_{d-1-n})$, $d \geq 2, n < d$.

Proof. It follows from the rule (PC).

□

$$\begin{array}{c}
 \text{Diagram 1} = \text{Diagram 2}
 \end{array}$$

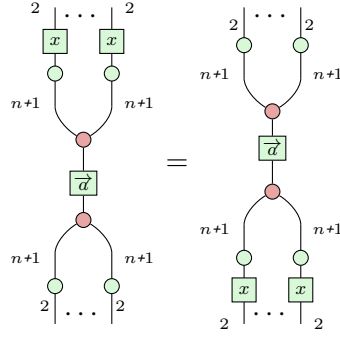
Diagram 1 shows a node with two inputs and two outputs, connected by red dots. Diagram 2 shows a similar node but with a different internal structure. The transformation is labeled (S1).

Lemma 8.

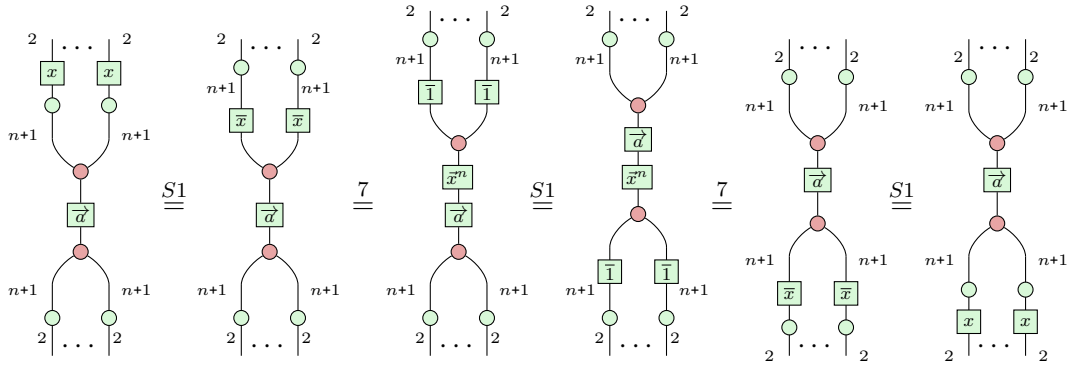
Proof. It follows from the rule (PC).

□

Lemma 9. *The qubit Z spiders could commute with the symmetriser:*

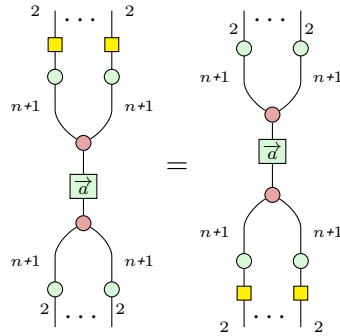


Proof.



□

Lemma 10. *The qubit Hadamard gates could commute with the symmetriser:*



Proof. Since the qubit Hadamard gate is self-adjoint, it is equivalent to prove the following

equality:

The last equality holds because

$$\sum_{j=0}^n \sum_{\substack{s_1+\dots+s_n=j, \\ s_1, \dots, s_n \in \{0,1\}}} \begin{array}{cc} s_1 \pi & s_n \pi \\ k_1 \pi & k_n \pi \end{array} = \sum_{m=0}^n \sum_{\substack{t_1+\dots+t_n=m, \\ t_1, \dots, t_n \in \{0,1\}}} \begin{array}{cc} l_1 \pi & l_n \pi \\ t_1 \pi & t_n \pi \end{array}$$

which is a power of -1 . □

Proposition 5 (Invariance). *For an arbitrary 2×2 unitary represented by $\begin{array}{|c|} \hline U \\ \hline \end{array}$, the following equality holds:*

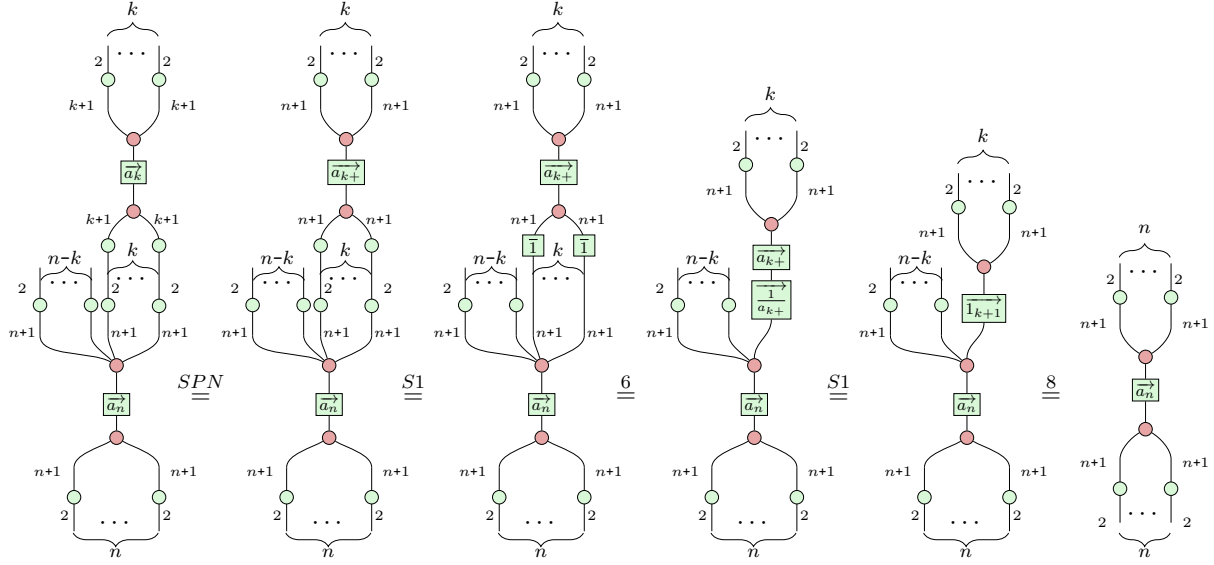
Proof. Note that any 2×2 unitary can be decomposed as a product of Z spiders and Hadamard gates. So it follows directly from Lemmas 9 and 10. □

Proposition 6 (Stacking). *Suppose $k \leq j$. Then*

Proof. Let

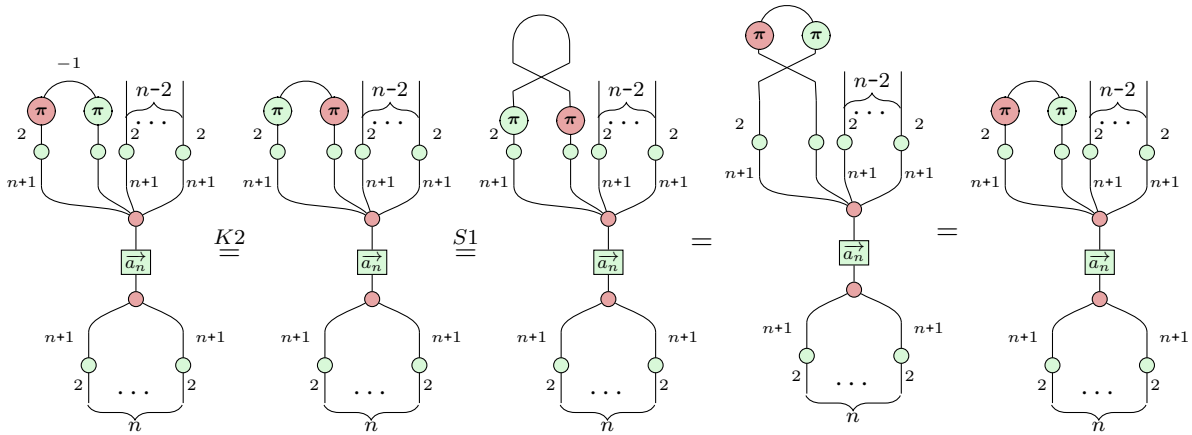
$$\overrightarrow{a_{k+}} = \left(\frac{1}{\binom{k}{1}}, \dots, \frac{1}{\binom{k}{k}}, \underbrace{0, \dots, 0}_{n-k} \right), \quad \overleftarrow{1}_{a_{k+}} = \left(\binom{k}{1}, \dots, \binom{k}{k}, \underbrace{0, \dots, 0}_{n-k} \right).$$

Then



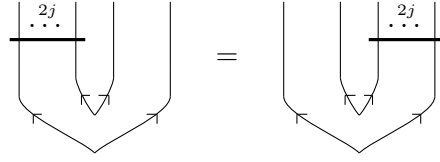
Proposition 7 (Capping / Cupping).

Proof. The first equality holds due to the following:

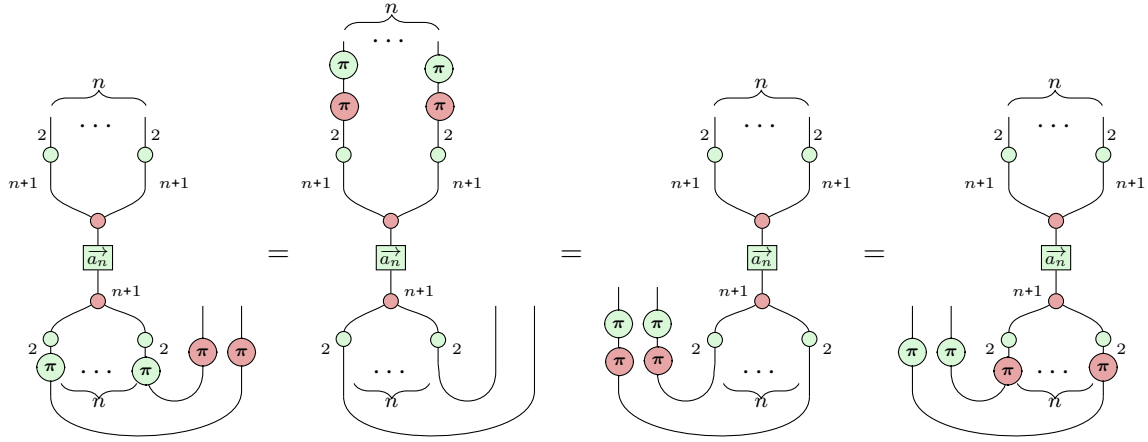


The second equality can be proved similarly. □

Proposition 8 (Symmetriser Sliding).

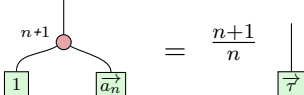


Proof.



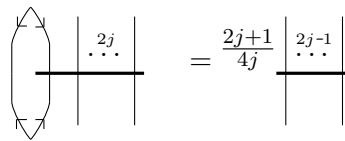
where for the second equality we used the self-transpose property of the symmetriser. $\square$

To prove the looping relation property, we need the following lemma:

Lemma 11.  where $\vec{\tau} = \left(\frac{1}{\binom{n-1}{1}}, \dots, \frac{1}{\binom{n-1}{n-1}}, \frac{2n}{n+1} \right)$.

Proof. It follows from the rule (PA). $\square$

Proposition 9 (Looping Relation).



1

Proposition 16. $[J_+, J_-] = 2J_3$, or diagrammatically,

$$\begin{array}{c} \sqrt{2j} \\ \text{---} \\ \text{---} L_j \\ \text{---} K_{-1} \\ \text{---} \\ \sqrt{2j} \\ \text{---} K_1 \\ \text{---} \\ \sqrt{2j} \\ \text{---} L_j \end{array} - \begin{array}{c} \sqrt{2j} \\ \text{---} \\ \text{---} K_1 \\ \text{---} L_j \\ \text{---} \\ \sqrt{2j} \\ \text{---} L_j \\ \text{---} K_{-1} \end{array} = 2j \begin{array}{c} \text{---} \\ \text{---} A_j \\ \text{---} \end{array} \quad (54)$$

$$[J_+, J_-] = J_+ J_- - J_- J_+ = 2j \begin{array}{c} \square_{L_j} \\ \circ_{K_{-1}} \\ \circ_{K_1} \\ \square_{L_j} \end{array} - 2j \begin{array}{c} \circ_{K_1} \\ \square_{L_j} \\ \square_{L_j} \\ \circ_{K_{-1}} \end{array} = 2j \begin{array}{c} \square_{L_j^2} \end{array} - 2j \begin{array}{c} \circ_{K_1} \\ \square_{L_j^2} \\ \circ_{K_{-1}} \end{array}$$
$$[J_+, J_-] = 2j \boxed{A_j} = 2J_3$$

where we used

$$\begin{array}{c} \bullet \\ | \end{array} - \begin{array}{c} \textcolor{red}{K}_{-1} \\ | \end{array} = \begin{array}{c} \boxed{-1} \\ | \end{array} \quad \text{and} \quad \begin{array}{c} \boxed{L_j^2} \quad \boxed{-1} \\ \diagdown \quad \diagup \\ \bullet \\ | \end{array} = \begin{array}{c} \boxed{A_j} \\ | \end{array}.$$

□

Proposition 17. $[J_3, J_+] = J_+$, or diagrammatically,

$$\begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ j \begin{array}{c} \boxed{A_j} \end{array} \end{array} \end{array} - \begin{array}{c} j \begin{array}{c} \boxed{A_j} \end{array} \\ | \\ \sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \boxed{L_j} \end{array} \end{array} = \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \sqrt{2j} \begin{array}{c} \boxed{L_j} \end{array} \end{array} \quad (55)$$

Proof.

$$\begin{aligned} [J_3, J_+] &= J_3 J_+ - J_+ J_3 = j\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \boxed{L_j} \\ | \\ \boxed{A_j} \end{array} - j\sqrt{2j} \begin{array}{c} \boxed{A_j} \\ | \\ \textcolor{red}{K}_1 \\ | \\ \boxed{L_j} \end{array} = j\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \boxed{L_j} \\ | \\ \boxed{A_j} \end{array} - j\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_{-1} \\ | \\ \boxed{A_j'} \\ | \\ \boxed{L_j} \end{array} \\ &= j\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \bullet \\ | \\ \boxed{L_j} \quad \boxed{A_j} \end{array} - j\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_{-1} \\ | \\ \bullet \\ | \\ \boxed{A_j} \quad \boxed{A_j'} \end{array} = \sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \bullet \\ | \\ \boxed{L_j} \quad \boxed{A_j''} \end{array} = \sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \boxed{A_j''} \\ | \\ \boxed{L_j} \end{array} = \sqrt{2j} \begin{array}{c} \textcolor{red}{K}_1 \\ | \\ \boxed{L_j} \end{array} = J_+ \end{aligned}$$

where

$$A_j' = \left(\frac{j-2}{j-1}, \frac{j-3}{j-1}, \dots, \frac{j-2j}{j-1}, \frac{j}{j-1} \right), \quad A_j'' = \frac{1}{j} (1, 1, \dots, 1, -2j).$$

□

Proposition 18. $[J_3, J_-] = -J_-$, or diagrammatically,

$$\begin{array}{c} \boxed{L_j} \\ | \\ \sqrt{2j} \begin{array}{c} \textcolor{red}{K}_{-1} \\ | \\ j \begin{array}{c} \boxed{A_j} \end{array} \end{array} \end{array} - \begin{array}{c} j \begin{array}{c} \boxed{A_j} \end{array} \\ | \\ \sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \end{array} \end{array} = \begin{array}{c} \boxed{L_j} \\ | \\ -\sqrt{2j} \begin{array}{c} \textcolor{red}{K}_{-1} \end{array} \end{array} \quad (56)$$

Proof.

$$\begin{aligned} [J_3, J_-] &= J_3 J_- - J_- J_3 = j\sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \\ | \\ \boxed{A_j} \end{array} - j\sqrt{2j} \begin{array}{c} \boxed{A_j} \\ | \\ \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \end{array} = j\sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \\ | \\ \boxed{A_j'} \end{array} - j\sqrt{2j} \begin{array}{c} \boxed{A_j} \\ | \\ \textcolor{red}{K}_{-1} \\ | \\ \boxed{L_j} \end{array} \\ &= j\sqrt{2j} \begin{array}{c} \bullet \\ | \\ \boxed{L_j} \quad \boxed{A_j'} \\ | \\ \textcolor{red}{K}_{-1} \quad \textcolor{red}{K}_{-1} \end{array} - j\sqrt{2j} \begin{array}{c} \bullet \\ | \\ \boxed{L_j} \quad \boxed{A_j} \\ | \\ \textcolor{red}{K}_{-1} \end{array} = -\sqrt{2j} \begin{array}{c} \bullet \\ | \\ \boxed{L_j} \quad \boxed{A_j''} \\ | \\ \textcolor{red}{K}_{-1} \end{array} = -\sqrt{2j} \begin{array}{c} \boxed{A_j''} \\ | \\ \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \end{array} = -\sqrt{2j} \begin{array}{c} \boxed{L_j} \\ | \\ \textcolor{red}{K}_{-1} \end{array} = -J_- \end{aligned}$$

□

F Diagrammatic Proofs of the Wigner Matrix

Proposition 10.

$$D_{mn}^j(u) = \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } K_{j,n+1}. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } K_{j-m} \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} = E,$$

where

$$u = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{and} \quad E = \sum_k \frac{\sqrt{(j-m)!(j+m)!(j-n)!(j+n)!}}{k!(j-m-k)!(j+n-k)!(m-n+k)!} a^{j+n-k} b^{m-n+k} c^k d^{j-m-k}$$

where the sum runs over all the values of k for which the argument of every factorial is non-negative [37].

Proof.

$$\begin{aligned} & \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } K_{j,n+1}. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } K_{j-m} \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} \\ &= \frac{1}{\sqrt{\binom{2j}{j-n}}} \begin{array}{c} \text{Diagram: Similar to the first diagram, but the top red oval is labeled } K_{j,n+1} \text{ and the bottom red oval is labeled } K_{j-m}. \end{array} \\ &= \frac{1}{\sqrt{\binom{2j}{j-n}}} \frac{1}{\sqrt{\binom{2j}{j-m}}} \sum_{\substack{i_1+\dots+i_{2j}=j-n \\ i_1+\dots+i_{2j} \in \{0,1\}}} \frac{1}{\sqrt{\binom{2j}{j-n}}} \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } i_1\pi. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } l_1\pi \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} \\ &= \frac{1}{\sqrt{\binom{2j}{j-n}}} \frac{1}{\sqrt{\binom{2j}{j-m}}} \sum_k \frac{(2j)!}{k!(j-m-k)!(j+n-k)!(m-n+k)!} a^{j+n-k} b^{m-n+k} c^k d^{j-m-k} \\ &= \sum_k \frac{\sqrt{(j-m)!(j+m)!(j-n)!(j+n)!}}{k!(j-m-k)!(j+n-k)!(m-n+k)!} a^{j+n-k} b^{m-n+k} c^k d^{j-m-k} \\ & \text{where } \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } i_s\pi. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } l_s\pi \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} \in \{a, b, c, d\}. \text{ If the number of the term of } \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red circle labeled } \pi. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } \pi \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} = c \text{ in } \begin{array}{c} \text{Diagram: A vertical chain of nodes. At the top is a red oval labeled } i_1\pi. Below it is a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow. This is connected to a red circle labeled } 2j+1. Below this is a chain of green circles connected by arcs, with a central green circle labeled } 2j. Below the chain is another red circle labeled } 2j+1, followed by a green box labeled } \sqrt{a_{2j}} \text{ with a double arrow, and finally a red oval labeled } l_1\pi \text{ at the bottom. The entire chain is flanked by two vertical paths of green circles, each with a central node labeled } 2j. \end{array} \text{ is } k, \text{ then the number} \end{aligned}$$

of the term of $\begin{array}{c} \pi \\ | \\ \boxed{u} \\ | \\ \pi \end{array} = d$ is $j - m - k$, the number of the term of $\begin{array}{c} \pi \\ | \\ \boxed{u} \\ | \\ \bullet \end{array} = b$ is $j - n - (j - m - k) = m - n + k$, and the number of the term of $\begin{array}{c} \bullet \\ | \\ \boxed{u} \\ | \\ \bullet \end{array} = a$ is $j + n - k$. Therefore, by the formula of full permutations with repeated elements, the number of the term of $a^{j+n-k}b^{m-n+k}c^k d^{j-m-k}$ is $\frac{(2j)!}{k!(j-m-k)!(j+n-k)!(m-n+k)!}$.

□

G Diagrammatic Proofs of the 3jm-Symbol Properties.

Irreducibles are the fundamental building blocks from which other representations are built. Any finite-dimensional representation of $SU(2)$ is completely reducible, meaning it can be written as a direct sum of irreps. Specifically, a tensor product of irreps can be decomposed into a direct sum of irreps via a bijective intertwiner (or equivariant map). Recoupling theory aims to study the space of such intertwiners.

Intertwiners. Given two representations V and W of a group G , a linear map $\iota : V \longrightarrow W$ is an *intertwiner* when it commutes with the group action, meaning that

$$\iota(g \cdot v) = g \cdot \iota(v) \quad (107)$$

for all $g \in G$ and $v \in V$. It's easy to see that the set of intertwiners forms a vector space. We denote this vector space by $\text{Hom}_G(V, W)$. This space $\text{Hom}_G(V, W)$ is isomorphic to $\text{Inv}_G(V \otimes W^*)$, where W^* is the dual vector space of W with the dual representation $(\rho^*(g) = \rho(g^{-1})^T)$, and we define the *invariant subspace* of any representation on $\mathcal{H}$ as

$$\text{Inv}_G(\mathcal{H}) \stackrel{\text{def}}{=} \{ \psi \in \mathcal{H} \mid \forall g \in G, g \cdot \psi = \psi \}. \quad (108)$$

Equivalently, we can characterise it by

$$\text{Inv}_G(\mathcal{H}) = \{ \psi \in \mathcal{H} \mid \forall a \in \mathfrak{g}, a \cdot \psi = 0 \}. \quad (109)$$

Physicists use the term “intertwiner” for any vector in $\text{Inv}_G(\mathcal{H})$.

Spin networks. Given a collection of intertwiners, there is a way to weave these together into networks of spins for which the connective topology is all that matters.

Definition 1. A spin network is a triple $(\Gamma, L_\Gamma, \iota)$ that consists of

- a directed⁷ labelled graph Γ ,
- a labelling L_Γ that assigns an irrep (i.e. spin) j of $SU(2)$ to each edge e of Γ , and

⁷Though in the literature the directedness is often suppressed.

- a set of intertwiners ι that assigns to each vertex v a mapping:

$$|\iota_v\rangle \in \text{Inv}(v, j) \stackrel{\text{def}}{=} \text{Inv}_{\text{SU}(2)} \left(\bigotimes_{e \in \mathcal{S}(v)} H_{j_e} \otimes \bigotimes_{e \in \mathcal{T}(v)} H_{j_e}^\dagger \right). \quad (110)$$

This is an $SU(2)$ equivariant⁸ map from the irrep Hilbert spaces H_{j_e} inbound source edges $\mathcal{S}(v)$ to the outbound target edge $\mathcal{T}(v)$ Hilbert spaces $H_{j_e}^\dagger$.

G.1 3j state

Lemma 12.

where $\vec{u} = \left((-1)^1 \binom{d}{1}, \dots, (-1)^d \binom{d}{d} \right)$.

Proof.

where $\vec{v} = (-1)^{-d} \left((-1)^0 \binom{d}{0}, (-1)^1 \binom{d}{1}, \dots, (-1)^{d-1} \binom{d}{d-1} \right)$.

$$\begin{aligned} \vec{v} &= (-1)^{-d} \left((-1)^{d-1} \binom{d}{d-1}, (-1)^{d-2} \binom{d}{d-2}, \dots, (-1)^0 \binom{d}{0} \right) \\ &= \left((-1)^{-1} \binom{d}{1}, (-1)^{-2} \binom{d}{2}, \dots, (-1)^{-d} \binom{d}{d} \right) \\ &= \left((-1)^1 \binom{d}{1}, \dots, (-1)^d \binom{d}{d} \right) = \vec{u}. \end{aligned}$$

□

Now we can derive the diagram which represent the state $|j_1, j_2, j_3\rangle$ as defined in Equation (51). For this, we will need to use the following lemma.

⁸That is, it lives in the space of $SU(2)$ invariant objects.

Lemma 13. For all pairs $(k, l) \in \{(1, 2), (1, 3), (2, 3)\}$, we have:

$$\begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ 2j_k+1 \end{array} = \begin{array}{c} \overrightarrow{\frac{1}{a}} \\ \vdots \\ \pi \\ \vdots \\ D \\ n_k \end{array} = \begin{array}{c} \overrightarrow{\frac{1}{a}} \\ \vdots \\ \pi \\ \vdots \\ D \\ n_k \end{array} = (-1)^{x_{kl}} \begin{array}{c} \overrightarrow{\frac{1}{a}} \\ \vdots \\ \pi \\ \vdots \\ D \\ n_k \end{array}$$

where $x_{12} = j_1 + j_2 - j_3$, $x_{13} = j_1 + j_3 - j_2$, $x_{23} = j_2 + j_3 - j_1$.

Proof.

$$\begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ 2j_k+1 \end{array} \stackrel{PK}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{S4}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{S1}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \quad (111)$$

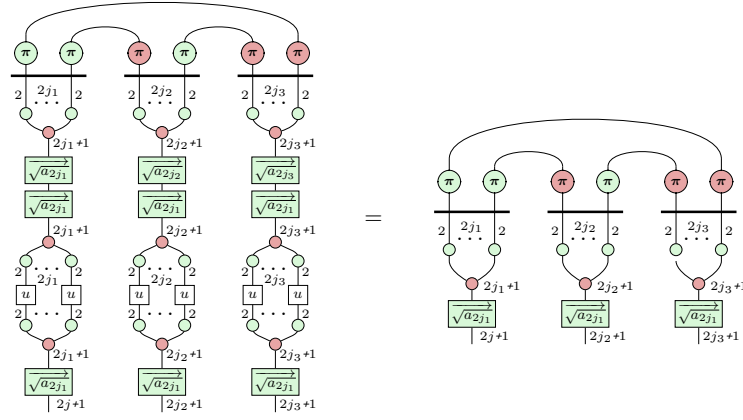
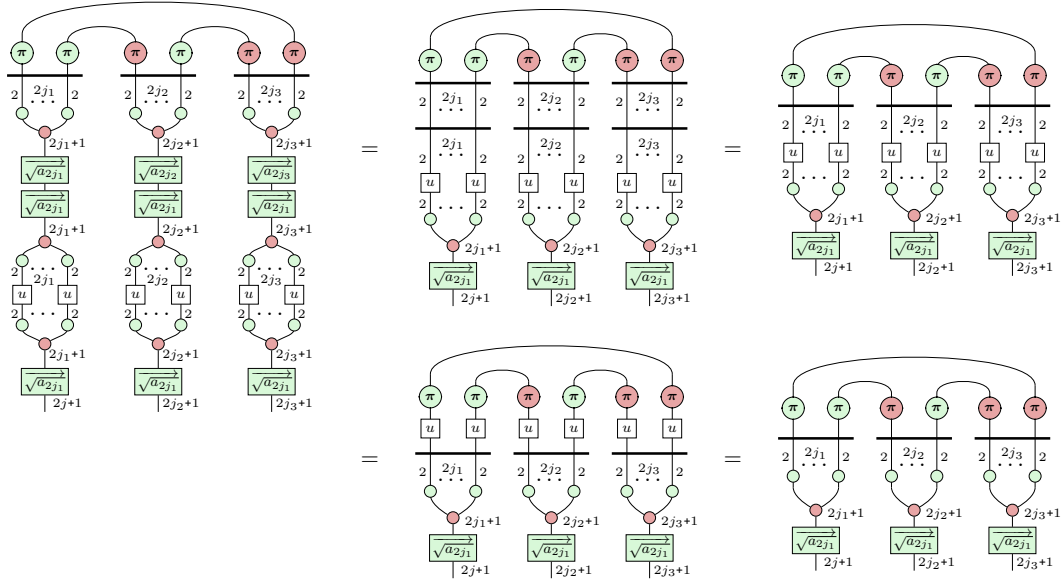
$$\begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{4}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{PA}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{44}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{DZX}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{S1}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \quad (112)$$

$$\begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{44}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{PK}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \stackrel{12}{=} \begin{array}{c} \pi \quad \pi \\ \vdots \\ \pi \quad \pi \\ \vdots \\ n_k \end{array} \quad (113)$$

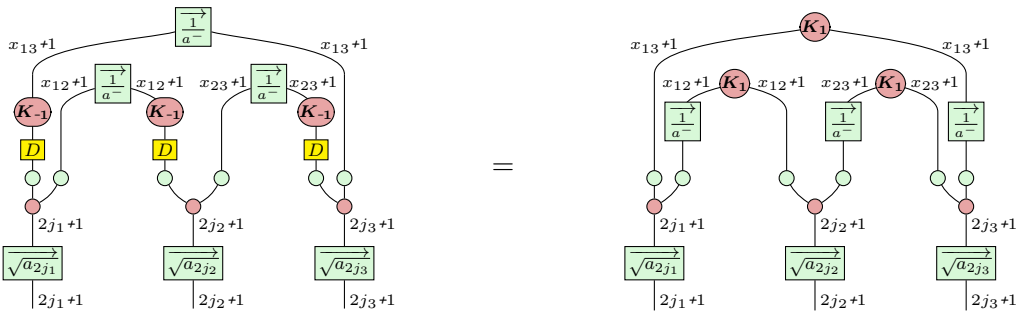
where $n_k = 2j_k + 1$,

$$\overrightarrow{\frac{1}{a_{x_{kl}}}} = \left((-1)^1 \binom{x_{kl}}{1}, \dots, (-1)^{x_{kl}} \binom{x_{kl}}{x_{kl}}, \underbrace{0, \dots, 0}_{n_k n_l - 1 - x_{kl}} \right), \quad \text{and} \quad \overrightarrow{\frac{1}{a^-}} = \left((-1)^1 \binom{x_{kl}}{1}, \dots, (-1)^{x_{kl}} \binom{x_{kl}}{x_{kl}} \right).$$

□

Proposition 11.*Proof.*

□

Lemma 14.

Proof.

$$(114)$$

$$(115)$$

□

Then we can derive the diagram for the 3j state as follows:

Theorem 1.

$$|j_1, j_2, j_3\rangle = \frac{1}{N(j_1, j_2, j_3)} \quad (48)$$

where $x_{12} = j_1 + j_2 - j_3$, $x_{13} = j_1 + j_3 - j_2$, $x_{23} = j_2 + j_3 - j_1$; $\overrightarrow{\frac{1}{a}} = \left((-1)^1 \binom{x_{k\ell}}{1}, \dots, (-1)^{x_{k\ell}} \binom{x_{k\ell}}{x_{k\ell}} \right)$.

$$|j_1, j_2, j_3\rangle \stackrel{47}{=} \frac{(-1)^{j_1-j_2+j_3}}{N(j_1, j_2, j_3)} \text{ (Diagrammatic Expression) } \quad (116)$$

$$\stackrel{13}{=} \frac{1}{N(j_1, j_2, j_3)} \quad (118)$$

$$\stackrel{14}{=} \frac{1}{N(j_1, j_2, j_3)} \quad (119)$$

☐

Theorem 2. *Suppose $m_1 + m_2 + m_3 = 0$, and j_1, j_2, j_3 satisfy the Clebsch-Gordan conditions.*

Then, the $3jm$ -symbol is given by the following

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} = \frac{1}{N(j_1, j_2, j_3)} \text{ (Diagram) } = D,$$

where

$$D = (-1)^{j_1-j_2-m_3} \sqrt{\frac{(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}{(j_1+j_2+j_3+1)!}} \\ \times \sqrt{(j_1-m_1)!(j_1+m_1)!(j_2-m_2)!(j_2+m_2)!(j_3-m_3)!(j_3+m_3)!} \\ \times \sum_{k=K}^N \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!(j_2+m_2-k)!}$$

which is given in [52].

Proof. Below, we calculate the explicit value for the $3jm$ -symbol.

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \quad (120)$$

$$= \frac{1}{N(j_1, j_2, j_3)} \text{ (Diagram) } \quad (121)$$

$$\stackrel{14}{=} \frac{1}{N(j_1, j_2, j_3)} \text{ (Diagram) } = C \text{ (Diagram) } \quad (122)$$

$$\begin{aligned}
&= C \sum_{k=0}^{x_{12}} (-1)^k \binom{x_{12}}{k} \text{ (Diagram 1) } = C \sum_{k=0}^{x_{12}} (-1)^k \binom{x_{12}}{k} \text{ (Diagram 2) } \\
&\quad \text{(123)}
\end{aligned}$$

Diagram 1: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-k} and D . K_{-k} has children $K_{j_1-m_1}$ and $K_{j_2-m_2}$. D has children $K_{j_2-m_2}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{12}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

Diagram 2: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_1-m_1-k}$ and $K_{j_2-m_2-x_{12}+k}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

$$\begin{aligned}
&= C \sum_{k=0}^{x_{12}} (-1)^{k+x_{13}} \binom{x_{12}}{k} \text{ (Diagram 3) } = C \sum_{k=0}^{x_{12}} (-1)^{k+x_{13}} \binom{x_{12}}{k} \text{ (Diagram 4) } \\
&\quad \text{(124)}
\end{aligned}$$

Diagram 3: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_1-m_1-k}$ and $K_{j_2-m_2-x_{12}+k}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

Diagram 4: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_1-m_1-k}$ and $K_{j_2-m_2-x_{12}+k}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

$$\begin{aligned}
&= C \sum_{k=0}^{x_{12}} (-1)^{k+x_{13}} \binom{x_{12}}{k} \text{ (Diagram 5) } = C \sum_{k=0}^{x_{12}} (-1)^{k+x_{13}} \binom{x_{12}}{k} \text{ (Diagram 6) } \\
&\quad \text{(125)}
\end{aligned}$$

Diagram 5: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_1-m_1-k}$ and $K_{j_2-m_2-x_{12}+k}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

Diagram 6: A tree structure with root $\frac{1}{a}$. Left child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_1-m_1-k}$ and $K_{j_2-m_2-x_{12}+k}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Right child $\frac{1}{a}$ has children K_{-1} and D . K_{-1} has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. D has children $K_{j_2-m_2-x_{12}+k}$ and $K_{j_3-m_3}$. Labels: $x_{13}+1$, $x_{23}+1$, $2j_1+1$, $2j_2+1$, $2j_3+1$.

$$\text{where } C = \frac{1}{\sqrt{\binom{2j_1}{j_1-m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2-m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3-m_3}}} \frac{1}{N(j_1, j_2, j_3)}.$$

Now we need to know the range of the values $j_1 - m_1 - k$ and $j_2 - m_2 - x_{12} + k$.

1) Bounds for $j_1 - m_1 - k$.

i) First, we give an upper bound for $j_1 - m_1 - k$:

$$\begin{aligned}
-j_1 &\leq m_1 \leq j_1 \Rightarrow -j_1 - m_1 \leq 0 \\
k &\geq 0 \Rightarrow -j_1 - m_1 \leq k \\
-j_1 - m_1 &\leq k \Leftrightarrow j_1 - m_1 - k \leq 2j_1
\end{aligned}$$

Therefore, $j_1 - m_1 - k \leq 2j_1$.

ii) Then we give a lower bound for $j_1 - m_1 - k$:

$$\left. \begin{aligned} 0 &\leq k \leq x_{12} = j_1 + j_2 - j_3 \\ j_1 + j_2 - j_3 &\leq 2j_1 \Leftrightarrow j_2 - j_3 \leq j_1 \text{ (Clebsch-Gordan condition)} \end{aligned} \right\} \Rightarrow k \leq 2j_1$$

Since $m_1 \leq j_1$, we have $k \leq 2j_1 + j_1 - m_1$, which is equivalent to $j_1 - m_1 - k \geq -2j_1$. But this lower bound is not tight enough. To get a tighter one, we assume that $j_1 - m_1 - k < 0$. Then we have

$$\left. \begin{aligned} j_1 - m_1 - k &\geq -2j_1 \\ j_1 - m_1 - k &< 0 \end{aligned} \right\} \Rightarrow 1 \leq 2j_1 + 1 + j_1 - m_1 - k < 2j_1 + 1$$

Furthermore,

$$\left. \begin{array}{l} k \leq j_1 + j_2 - j_3 \\ j_1 - m_1 \geq 0 \end{array} \right\} \Rightarrow$$

$$2j_1 + 1 + j_1 - m_1 - k - x_{13} = 2j_1 + 1 + j_1 - m_1 - k - (j_1 + j_3 - j_2) = j_1 + j_2 - j_3 - k + j_1 - m_1 + 1 > 0,$$

i.e.,

$$2j_1 + 1 + j_1 - m_1 - k > x_{13}.$$

Hence

$$\begin{array}{c} x_{13}+1 \\ | \\ \textcircled{K_{j_1-m_1-k}} \end{array} \begin{array}{c} 2j_1+1 \\ | \\ \textcircled{K_{2j_1+1+j_1-m_1-k}} \end{array} = \begin{array}{c} x_{13}+1 \\ | \\ \textcircled{K_{2j_1+1+j_1-m_1-k}} \end{array} \begin{array}{c} 2j_1+1 \\ | \\ \textcircled{K_{2j_1+1+j_1-m_1-k}} \end{array} = 0 \begin{array}{c} x_{13}+1 \\ | \\ \textcircled{K_{2j_1+1+j_1-m_1-k}} \end{array}$$

which means the value of the 3jm-symbol is 0. To avoid this trivial case under the assumption $j_1 - m_1 - k < 0$, we can naturally set the lower bound as $j_1 - m_1 - k \geq 0$.

2) Bounds for $j_2 - m_2 - x_{12} + k$. Substitute x_{12} with its explicit value, we get

$$j_2 - m_2 - x_{12} + k = j_2 - m_2 - (j_1 + j_2 - j_3) + k = j_3 - j_1 - m_2 + k.$$

i) First we give a upper bound for $j_3 - j_1 - m_2 + k$:

$$\left. \begin{array}{l} k \leq j_1 + j_2 - j_3 \\ j_2 + m_1 \geq 0 \end{array} \right\} \Rightarrow k \leq j_1 + j_2 - j_3 + j_2 + m_2$$

which is equivalent to the following upper bound:

$$j_3 - j_1 - m_2 + k \leq 2j_2.$$

ii) Second, we give a lower bound for $j_3 - j_1 - m_2 + k$:

$$\left. \begin{array}{l} k \geq 0 \\ j_2 + j_3 - j_1 \geq 0 \\ j_2 - m_2 \geq 0 \end{array} \right\} \Rightarrow k \geq -(j_2 + j_3 - j_1 + j_2 - m_2)$$

which is equivalent to the following upper bound:

$$j_3 - j_1 - m_2 + k \geq -2j_2.$$

This lower bound is not tight enough. To get a tighter one, we assume that $j_3 - j_1 - m_2 + k < 0$. Then we have

$$\left. \begin{array}{l} j_3 - j_1 - m_2 + k \geq -2j_2 \\ j_3 - j_1 - m_2 + k < 0 \end{array} \right\} \Rightarrow 1 \leq 2j_2 + 1 + j_3 - j_1 - m_2 + k < 2j_2 + 1$$

Moreover,

$$\left. \begin{array}{l} k \geq 0 \\ j_2 - m_2 \geq 0 \end{array} \right\} \Rightarrow$$

$$2j_2 + 1 + j_3 - j_1 - m_2 + k - x_{23} = 2j_2 + 1 + j_3 - j_1 - m_2 + k - (j_2 + j_3 - j_1) = k + j_2 - m_2 + 1 > 0,$$

i.e.,

$$2j_2 + 1 + j_3 - j_1 - m_2 + k > x_{23}.$$

Therefore,

$$\begin{array}{c} x_{23}+1 \\ | \\ \text{green circle} \\ | \\ 2j_2+1 \\ | \\ \text{red oval } K_{j_2-m_2-x_{12}+k} \end{array} = \begin{array}{c} x_{23}+1 \\ | \\ \text{green circle} \\ | \\ 2j_2+1 \\ | \\ \text{red oval } K_{2j_2+1+j_3-j_1-m_2+k} \end{array} = \begin{array}{c} x_{23}+1 \\ | \\ 0 \\ | \\ \text{red circle} \end{array}$$

which means the value of the 3jm-symbol is 0. To avoid this trivial case under the assumption $j_3 - j_1 - m_2 + k < 0$, we can naturally set the lower bound as $j_3 - j_1 - m_2 + k \geq 0$.

Combining the bounds we just obtained for $j_1 - m_1 - k$ and $j_2 - m_2 - x_{12} + k$, we have

$$\begin{cases} 0 \leq k \leq x_{12} \\ 0 \leq j_1 - m_1 - k \leq 2j_1 \\ 0 \leq j_2 - m_2 - x_{12} + k \leq 2j_2 \end{cases}$$

To avoid the trivial case that the following diagram is 0,

$$\begin{array}{c} x_{13}+1 \\ | \\ \text{green circle} \\ | \\ 2j_1+1 \\ | \\ \text{red oval } K_{j_1-m_1-k} \end{array} \quad \begin{array}{c} x_{23}+1 \\ | \\ \text{green circle} \\ | \\ 2j_2+1 \\ | \\ \text{red oval } K_{j_2-m_2-x_{12}+k} \end{array}$$

we also need

$$\begin{cases} j_1 - m_1 - k \leq x_{13} \\ j_2 - m_2 - x_{12} + k \leq x_{23} \end{cases}$$

Since $x_{13} \leq 2j_1, x_{23} \leq 2j_2$, we then have

$$\begin{cases} 0 \leq k \leq x_{12} \\ 0 \leq j_1 - m_1 - k \leq x_{13} \\ 0 \leq j_2 - m_2 - x_{12} + k \leq x_{23} \end{cases}$$

which is equivalent to

$$\begin{cases} k \geq 0 \\ k \geq j_2 - j_3 - m_1 \\ k \geq j_1 - j_3 + m_2 \\ k \leq j_1 - m_1 \\ k \leq j_1 + j_2 - j_3 \\ k \leq j_2 + m_2 \end{cases}$$

Under these conditions, we have

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

$$= C \sum_{k=0}^{x_{12}} (-1)^{k+x_{13}} \binom{x_{12}}{k} \begin{array}{c} x_{13}+1 \\ | \\ \text{green box } \frac{1}{a} \\ | \\ x_{23}+1 \quad x_{23}+1 \\ | \quad | \\ \text{green circle} \quad \text{green circle} \\ | \quad | \\ 2j_1+1 \quad 2j_2+1 \\ | \quad | \\ \text{red oval } K_{j_1-m_1-k} \quad \text{red oval } K_{j_2-m_2-x_{12}+k} \end{array} \quad \begin{array}{c} x_{13}+1 \\ | \\ \text{green box } \frac{1}{a} \\ | \\ x_{23}+1 \quad x_{23}+1 \\ | \quad | \\ \text{green circle} \quad \text{green circle} \\ | \quad | \\ 2j_2+1 \quad 2j_3+1 \\ | \quad | \\ \text{red oval } K_{j_2-m_2-x_{12}+k} \quad \text{red oval } K_{j_3+m_3} \end{array} = \begin{array}{c} \text{red oval } K_{-(j_1-m_1-k)} \\ | \\ \text{red oval } K_{-(j_2-m_2-x_{12}+k)} \\ | \\ \text{red oval } K_{j_3+m_3} \end{array} \quad \begin{array}{c} D \\ | \\ \text{red oval } K_s \\ | \\ 2j_3+1 \end{array} = D$$

$$\begin{aligned}
&= C \sum_{k=K}^N (-1)^{j_1-m_1+j_2-m_2-x_{12}+k+x_{13}} \binom{x_{12}}{k} \binom{x_{13}}{j_1-m_1-k} \binom{x_{23}}{j_2-m_2-x_{12}+k} \\
&= \frac{1}{\sqrt{\binom{2j_1}{j_1-m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2-m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3-m_3}}} (-1)^{j_1-j_2+j_3} \\
&\times N^{-1}(j_1, j_2, j_3) \sum_{k=K}^N (-1)^{j_1-m_1+j_2-m_2-x_{12}+k} \binom{x_{12}}{k} \binom{x_{13}}{j_1-m_1-k} \binom{x_{23}}{j_2-m_2-x_{12}+k} \\
&= (-1)^{j_1-m_1+j_1-j_2+j_3+j_2-m_2-x_{12}} \frac{1}{\sqrt{\binom{2j_1}{j_1-m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2-m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3-m_3}}} \\
&\times N^{-1}(j_1, j_2, j_3) \sum_{k=K}^N (-1)^k \binom{x_{12}}{k} \binom{x_{13}}{j_1-m_1-k} \binom{x_{23}}{j_2-m_2-x_{12}+k}
\end{aligned}$$

where

$$\begin{aligned}
D &= C \sum_{k=K}^N (-1)^{j_1-m_1+j_2-m_2-x_{12}+k+x_{13}} \binom{x_{12}}{k} \binom{x_{13}}{j_1-m_1-k} \binom{x_{23}}{j_2-m_2-x_{12}+k}, \\
K &= \max\{0, j_2-j_3-m_1, j_1-j_3+m_2\}, \\
N &= \min\{j_1-m_1, j_2+m_2, j_1+j_2-j_3\}, \\
s &= -(j_2-m_2-x_{12}+k) - (j_1-m_1-k) + j_3+m_3 \\
&= -(j_2-m_2-(j_1+j_2-j_3)+k+j_1-m_1-k) + j_3+m_3 \\
&= -(j_3-m_2-m_1) + j_3+m_3 = m_2+m_1+m_3 = 0.
\end{aligned}$$

We can further simplify by

$$\begin{aligned}
(-1)^{j_1-m_1+j_1-j_2+j_3+j_2-m_2-x_{12}} &= (-1)^{2j_1+j_3-m_1-m_2-(j_1+j_2-j_3)} \\
&= (-1)^{j_1-j_2+2j_3+m_3} = (-1)^{j_1+j_3-j_2} (-1)^{j_3+m_3} \\
&= (-1)^{j_1+j_3-j_2} (-1)^{-j_3-m_3} = (-1)^{j_1-j_2-m_3}.
\end{aligned}$$

Second,

$$\begin{aligned}
\frac{1}{\sqrt{\binom{2j_1}{j_1-m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2-m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3-m_3}}} &= \sqrt{\frac{(j_1-m_1)!(j_1+m_1)!(j_2-m_2)!(j_2+m_2)!(j_3-m_3)!(j_3+m_3)!}{(2j_1)!(2j_2)!(2j_3)!}}, \\
N(j_1, j_2, j_3)^{-1} &= \sqrt{\frac{(2j_1)!(2j_2)!(2j_3)!}{(j_1+j_2+j_3+1)!(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}}, \\
\binom{x_{12}}{k} \binom{x_{13}}{j_1-m_1-k} \binom{x_{23}}{j_2-m_2-x_{12}+k} &= \frac{x_{12}!}{k!(x_{12}-k)!} \frac{x_{13}!}{(j_1-m_1-k)!(x_{13}-(j_1-m_1-k))!} \frac{x_{23}!}{(j_2-m_2-x_{12}+k)!(x_{23}-(j_2-m_2-x_{12}+k))!} \\
&= \frac{(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!(j_2+m_2-k)!}
\end{aligned}$$

With these simplifications and binomial coefficients expansions, we have

$$\begin{aligned} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} &= (-1)^{j_1-j_2-m_3} \sqrt{\frac{(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}{(j_1+j_2+j_3+1)!}} \\ &\times \sqrt{(j_1-m_1)!(j_1+m_1)!(j_2-m_2)!(j_2+m_2)!(j_3-m_3)!(j_3+m_3)!} \\ &\times \sum_{k=K}^N \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!(j_2+m_2-k)!} \end{aligned}$$

If we let the value of the 3j-symbol to be 0 for the case when $m_1 + m_2 + m_3 \neq 0$, then we get

$$\begin{aligned} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} &= \delta(m_1 + m_2 + m_3, 0) (-1)^{j_1-j_2-m_3} \sqrt{\frac{(j_2+j_3-j_1)!(j_1+j_3-j_2)!(j_1+j_2-j_3)!}{(j_1+j_2+j_3+1)!}} \\ &\times \sqrt{(j_1-m_1)!(j_1+m_1)!(j_2-m_2)!(j_2+m_2)!(j_3-m_3)!(j_3+m_3)!} \\ &\times \sum_{k=K}^N \frac{(-1)^k}{k!(j_1+j_2-j_3-k)!(j_1-m_1-k)!(j_3-j_2+m_1+k)!(j_3-j_1-m_2+k)!(j_2+m_2-k)!} \end{aligned}$$

where $\delta(i, j)$ is the Kronecker delta, which is exactly the same formula as given in [52]. □

G.2 Injection map

Proposition 12. *The injection $\mathcal{H}_{j_3} \longrightarrow \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2}$ as given in Equation (42) can be represented by the following diagram:*

Proof. We first prove that the injection $\mathcal{H}_{j_3} \longrightarrow \mathcal{H}_{j_1} \otimes \mathcal{H}_{j_2}$ as given in (42) can be represented by the following diagram:

We just need to prove that if we plug $|j_3 - m_3\rangle$ on top and $\langle j_1 - m_1| \otimes \langle j_2 - m_2|$ on bottom of the diagram, which is $C_{j_1 m_1 j_2 m_2}^{j_3 m_3}$ by definition, then we will get [38]

$$C_{j_1 m_1 j_2 m_2}^{j_3 m_3} = (-1)^{-j_1+j_2-m_3} \sqrt{2j_3+1} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & -m_3 \end{pmatrix}.$$

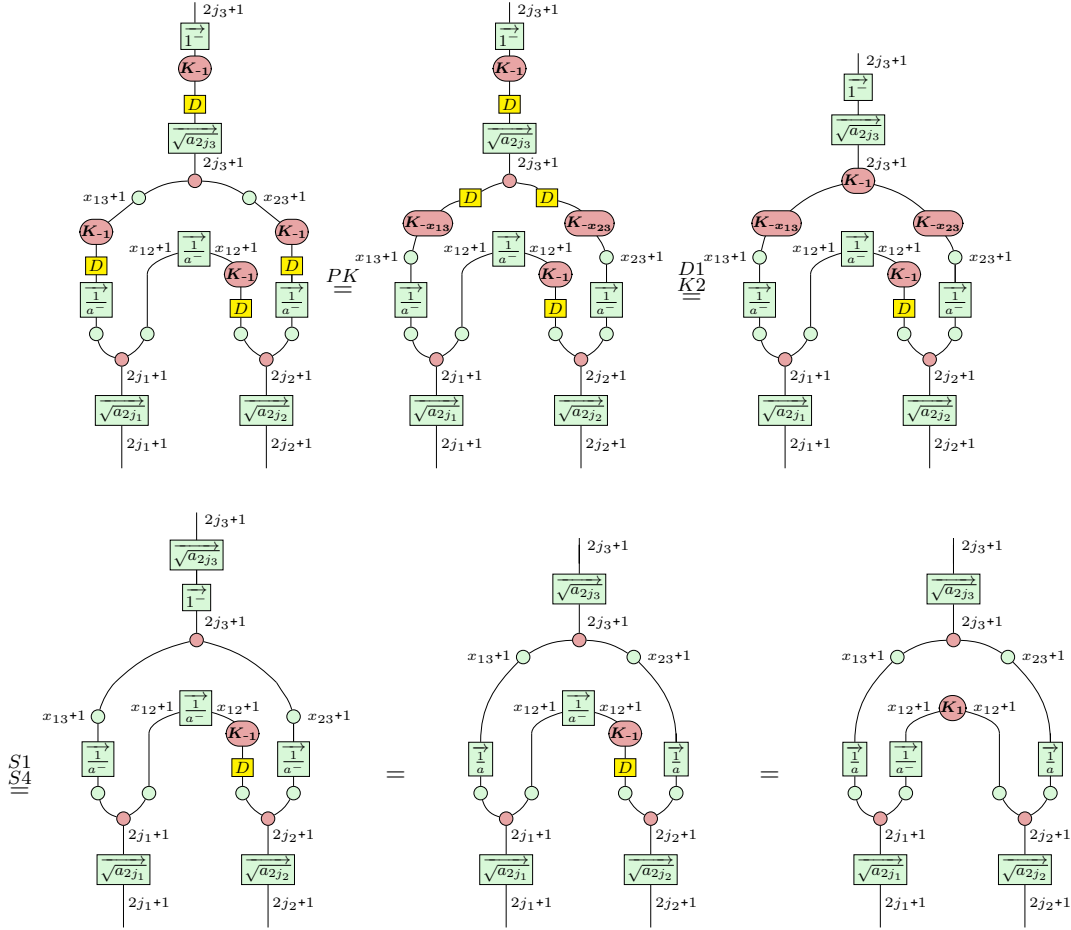
In fact, we have

$$\begin{aligned}
& \frac{\sqrt{2j_3+1}}{N(j_1, j_2, j_3)} \text{Diagram 1} = \frac{\sqrt{2j_3+1}}{N(j_1, j_2, j_3)} (-1)^{j_3-m_3} \text{Diagram 2} = \frac{(-1)^{-j_1+j_2-m_3} \sqrt{2j_3+1}}{N(j_1, j_2, j_3)} \text{Diagram 3} \\
& = \frac{(-1)^{-j_1+j_2-m_3} \sqrt{2j_3+1}}{N(j_1, j_2, j_3)} \text{Diagram 4} = \frac{(-1)^{-j_1+j_2-m_3} \sqrt{2j_3+1}}{N(j_1, j_2, j_3)} \text{Diagram 5} \\
& = (-1)^{-j_1+j_2-m_3} \sqrt{2j_3+1} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & -m_3 \end{pmatrix}.
\end{aligned}$$

where

$$(-1)^{j_3-m_3} (-1)^{x_{13}} = (-1)^{m_3-j_3} (-1)^{j_1+j_3-j_2} = (-1)^{j_1-j_2+m_3} = (-1)^{-j_1+j_2-m_3}.$$

Secondly, we have



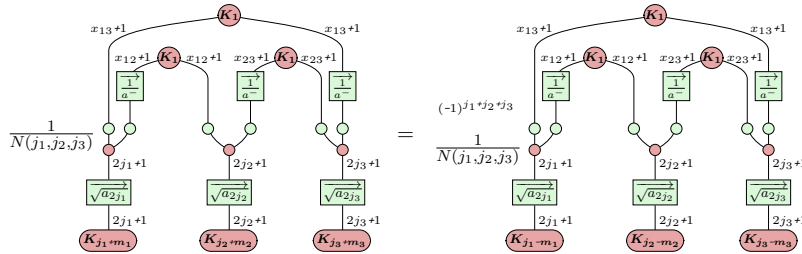
□

G.3 Symmetry properties of 3j-symbols

Proposition 15. *Changing the sign of the m quantum numbers gives a phase:*

$$\begin{pmatrix} j_1 & j_2 & j_3 \\ -m_1 & -m_2 & -m_3 \end{pmatrix} = (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix}$$

Diagrammatically, this is stated as follows:



Proof.

$$\left(\begin{array}{ccc} j_1 & j_2 & j_3 \\ -m_1 & -m_2 & -m_3 \end{array} \right) \stackrel{14}{=} \frac{1}{N(j_1, j_2, j_3)}$$
(126)

$$= C$$

$$= C$$
(127)

$$= C$$

$$= C$$
(128)

$$= C$$

$$= \frac{1}{N(j_1, j_2, j_3)}$$
(129)

$$= (-1)^{j_1+j_2+j_3} \begin{pmatrix} j_1 & j_2 & j_3 \\ m_1 & m_2 & m_3 \end{pmatrix} \quad (130)$$

where

$$C = \frac{1}{\sqrt{\binom{2j_1}{j_1+m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2+m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3+m_3}}} N(j_1, j_2, j_3) = \frac{1}{\sqrt{\binom{2j_1}{j_1-m_1}}} \frac{1}{\sqrt{\binom{2j_2}{j_2-m_2}}} \frac{1}{\sqrt{\binom{2j_3}{j_3-m_3}}} N(j_1, j_2, j_3)$$

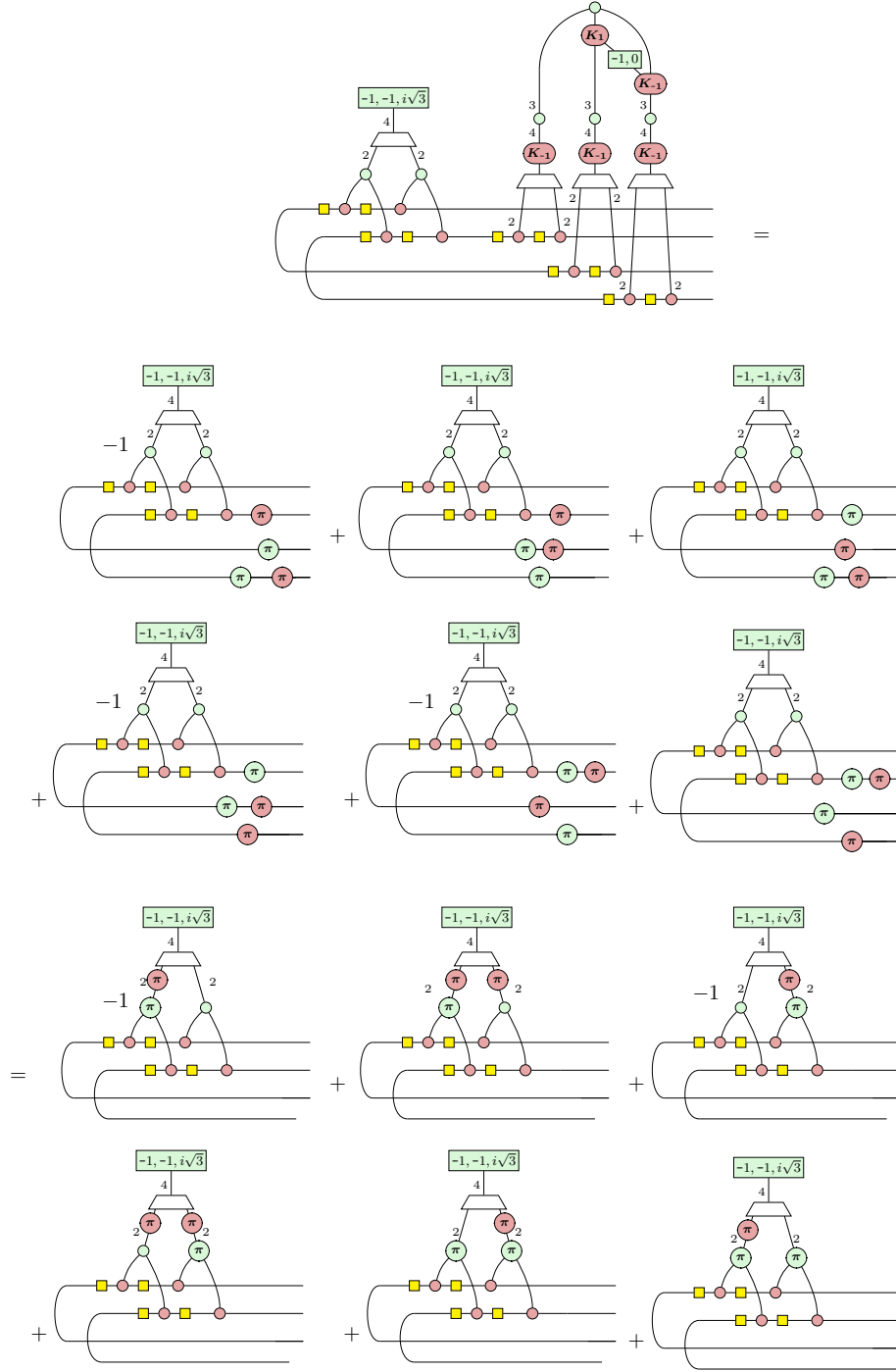
□

H Diagrammatic Calculations of Quantised Volume

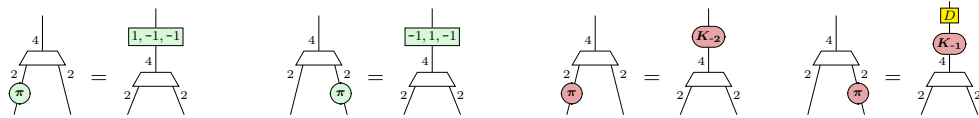
Proposition 21. *The state $|\iota_0 + i\sqrt{3}\iota_1\rangle$ is an eigenvector of the operator $I \otimes \tilde{V}^2$ with eigenvalue $\frac{-\sqrt{3}}{4}$.*

$$= \frac{-\sqrt{3}}{4} \quad (81)$$

Proof.



Using the following rules



we get

$$\begin{aligned}
 &= \text{Diagram 1} + \text{Diagram 2} + \text{Diagram 3} \\
 &+ \text{Diagram 4} + \text{Diagram 5} + \text{Diagram 6} \\
 &= \text{Diagram 7}
 \end{aligned}$$

The diagrams are as follows:

- Diagram 1:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The bottom crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The crossings are labeled 2, 2, 2, 2.
- Diagram 2:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The bottom crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The crossings are labeled 2, 2, 2, 2.
- Diagram 3:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The bottom crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The crossings are labeled 2, 2, 2, 2.
- Diagram 4:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The bottom crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The crossings are labeled 2, 2, 2, 2.
- Diagram 5:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The bottom crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The crossings are labeled 2, 2, 2, 2.
- Diagram 6:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The bottom crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The crossings are labeled 2, 2, 2, 2.
- Diagram 7:** A braid with 4 strands. The top two strands are crossed. The bottom two strands are crossed. The top crossing is labeled K_{-2} with a box containing $-1, 1, -i\sqrt{3}$. The bottom crossing is labeled K_{-1} with a box containing $-i\sqrt{3}, 1, -1$. The crossings are labeled 2, 2, 2, 2.

□