How to sum and exponentiate Hamiltonians in ZXW calculus

Razin A. Shaikh Quanlong Wang Richie Yeung
Cambridge Quantum / Quantiniuum

This paper develops practical summation techniques in ZXW calculus to reason about quantum dynamics, such as unitary time evolution. First we give a direct representation of a wide class of sums of linear operators, including arbitrary qubit Hamiltonians, in ZXW calculus. As an application, we give a diagrammatic representation of the Hamiltonian in Greene-Diniz et al [13], which is the first paper that models carbon capture using quantum computing. We then use the Cayley-Hamilton theorem to exponentiate arbitrary qubit Hamiltonians in ZXW calculus. This sets up the framework for using ZXW calculus to improve Hamiltonian simulation. We demonstrate for 1-qubit Hamiltonians how to obtain the circuit representing its one-parameter unitary group. Finally, we use the ZXW calculus to demonstrate the linearity of the Schrödinger equation.

1 Introduction

ZX calculus [5] is a graphical representation for quantum circuits and linear algebra in general: a diagram with n inputs and m outputs represents a $2^n \times 2^m$ linear map. ZX calculus comes with a complete set of rewrite rules such that two diagrams that represent the same linear map can always be rewritten to one another. Using this philosophy of *computation via rewriting*, ZX calculus has been successfully applied to a wide range of problems in quantum computing, including circuit compilation [2, 8, 9, 19], circuit equality validation [20,22], circuit simulation [21], error correction [3,4], natural language processing [6, 23] and quantum machine learning [27, 36, 37].

To solve a problem using ZX calculus, or other diagrammatic calculii, we first need to synthesise the expressions involved into diagrams. This can be done in an ad-hoc way or using general methods such as elementary matrices [34]. For example: in Hamiltonian simulation problems [24] the Hamiltonian is typically given as a sum of Pauli operators, and without an efficient representation of such Hamlitonians in ZX calculus, there is simply no way to proceed with the problem.

As of this writing, there is no published work on how to efficiently combine sums of ZX diagrams. Both [17] and [32] first require an inefficient conversion step to an intermediate form before the diagrams can be summed together. Furthermore, the resulting diagram is large and does not resemble the original symbolic expression, so the advantage of using diagrammatic reasoning is diminished.

To obtain intuitive sums of diagrams, we introduce the framework of *ZXW calculus*. Coecke and Kissinger [7] proposed the idea of developing a graphical calculus based on the interaction between GHZ and W states. Following up on that, Hadzihasanovic [15, 16] developed the ZW calculus. In this paper, we combine the ZX and ZW calculii to create the *ZXW calculus*. The W-spider of the ZXW calculus plays an important role in producing efficient and compact sums of diagrams. We define the ZXW calculus through a slight modification of the algebraic ZX calculus [31].

In this work, we modify and extend the notion of controlled states [18] to give direct representations of controlled diagrams for a wide class of matrices and show how to sum them within the framework

Submitted to:
QPL 2022

of ZXW calculus. These representations, combined with the recently developed techniques of differentiating arbitrary ZX diagrams [17, 35], allow us to *practically reason about analytical problems that were previously inaccessible to ZX calculus*. Moreover, the addition, exponentiation and differentiation operations interact nicely, allowing us to work with ZX diagrams as naturally as the traditional matrix notation, while still keeping the compact representation and rewriting advantages of the ZX calculus.

Specifically, these techniques allow us to reason about quantum dynamics and quantum chemistry in the ZXW calculus. To approach the problems of quantum chemistry in ZXW, we first devise a diagrammatic form for arbitrary Hamiltonians. One of the main problems in quantum computational chemistry is that of Hamiltonian simulation: given a Hamiltonian, find an approximation to its unitary time evolution. The ideal evolution is given by e^{-iHt}, where H is the Hamiltonian and t is time. Using the diagrammatic form of Hamiltonian and the summation techniques developed in this paper, we show how to diagrammatically exponentiate Hamiltonians. This allows us to write the unitary time evolution graphically and apply the ZXW rewrite rules to extract a quantum circuit for Hamiltonian simulation. Finally, to show the applicability of ZXW calculus in reasoning about traditional quantum mechanics, we provide the graphical form of the Schrödinger equation and prove the linearity of its solutions using ZXW rewrite rules.

A related diagrammatic approach to quantum dynamics is the work of Gogioso [10, 11, 12], which formulates this in the framework of categorical quantum mechanics [1]. While interesting from a foundational perspective, the higher-level approach does not allow for explicit reasoning on concrete real-world examples. In this paper, we will demonstrate the applicability of our method by expressing the Hamiltonian used recently to model carbon capture [13].

Summary of results

1. **Representing arbitrary Hamiltonians**: As examples, we express the Hamiltonians used in Greene-Diniz et. al. [13] (Figure 1), and Turner et. al. [30] (Example 5.3) using Theorem 5.2.

2. **Time-evolution operator**: We convert between a Hamiltonian and its one-parameter unitary group, i.e. the time-evolution operator, using exponentiation in ZXW calculus. (Section 6)

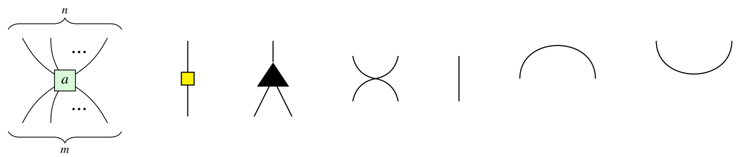
3. **Schrödinger's Equation**: We formulate the Schrödinger equation in ZXW calculus and show the linearity of its solutions. (Section 7)

2 ZXW Calculus

In this section, we give an introduction to ZXW calculus which is a slight modification of the algebraic ZX calculus [31], including its generators and rewriting rules. Note that algebraic ZX calculus is complete [31] for matrices of size $2^m \times 2^n$ and hence, so is the ZXW calculus. In this paper diagrams are either read from left to right or top to bottom.

2.1 Generators

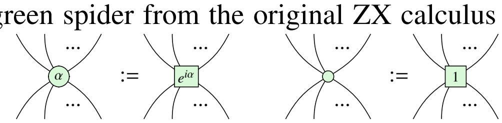
The diagrams in ZXW calculus are defined by freely combining the following generating objects. Note that a can be any complex number.



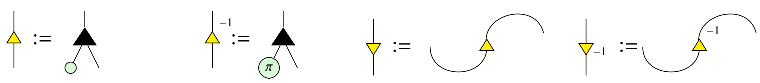
2.2 Additional notation

For simplicity, we introduce additional notation based on the given generators:

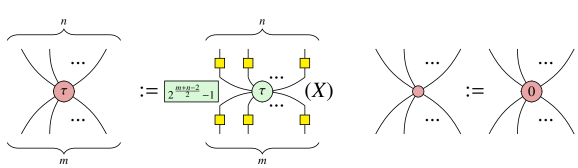
1. The green spider from the original ZX calculus can be defined using the green box spider in ZXW calculus.



2.The triangle and the inverse triangle can be expressed as follows. The transposes of the triangle and the inverse triangle can be drawn as inverted triangles.

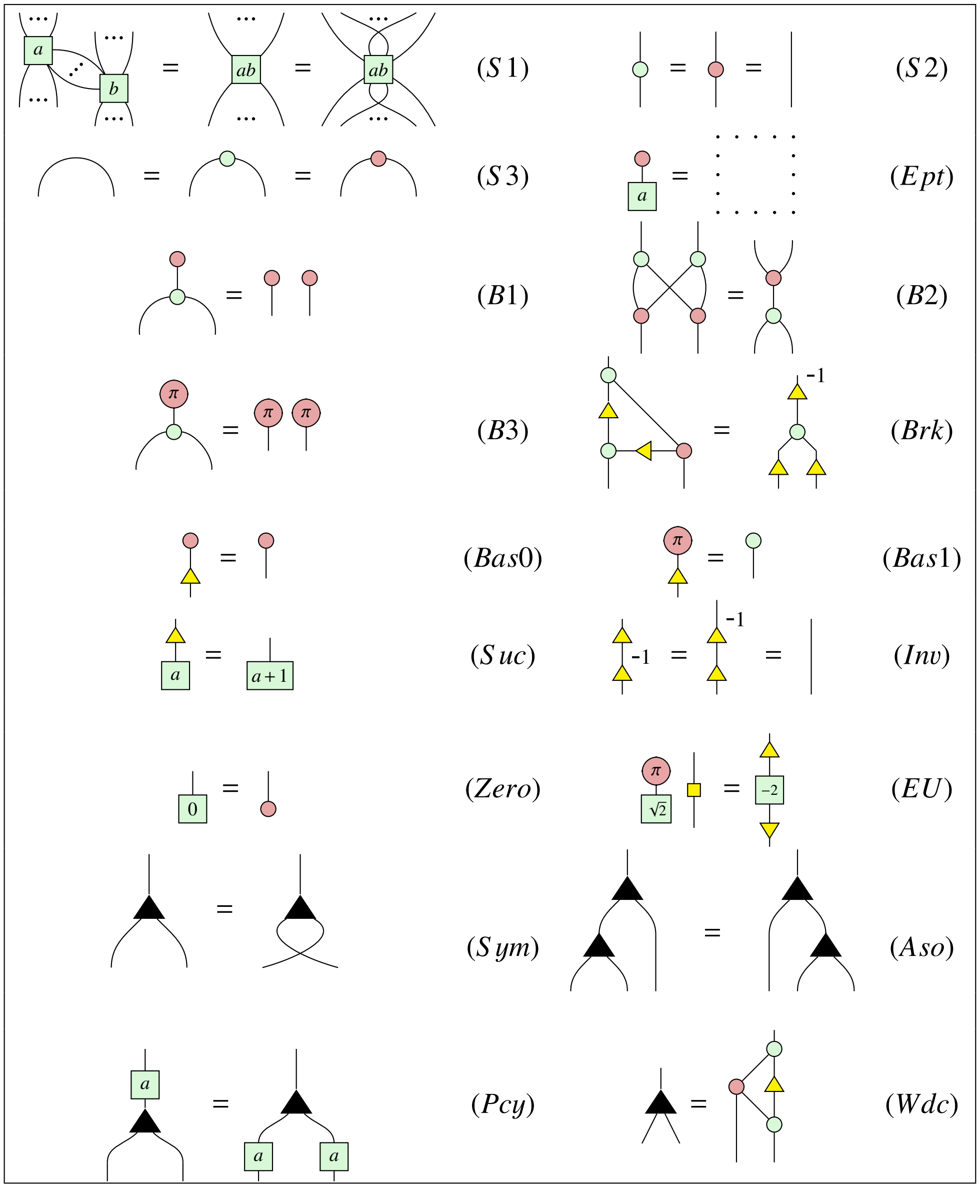


3. The red spider from the original ZX calculus can be defined by performing Hadamard conjugation on each leg of the green spider, and the pink spider is the algebraic equivalent of the red spider. It is only defined for $\tau \in \{0, \pi\}$, and is rescaled to have integer components in its matrix representation.



2.3 Rules

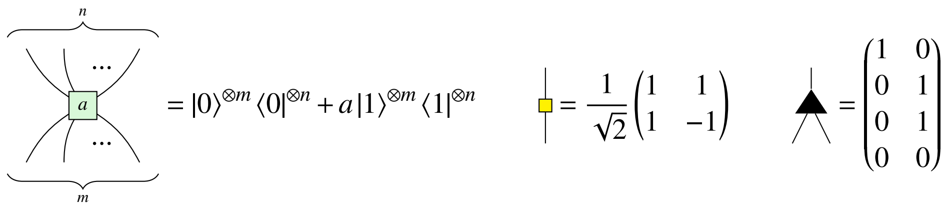
Now we give the rewriting rules of ZXW calculus.



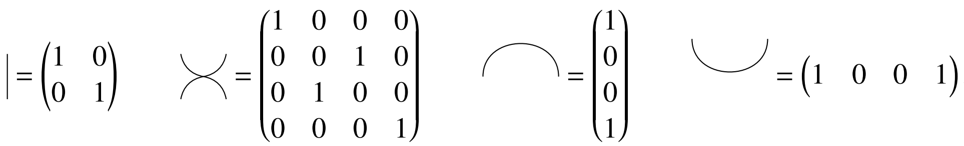
Where $a, b \in \mathbb{C}$. The vertically flipped versions of the rules are assumed to hold as well.

2.4 Interpretation

Although the generators in ZXW calculus are formal mathematical objects in their own right, in the context of this paper we interpret the generators as linear maps, so each ZXW diagram is equivalent to a vector or matrix.



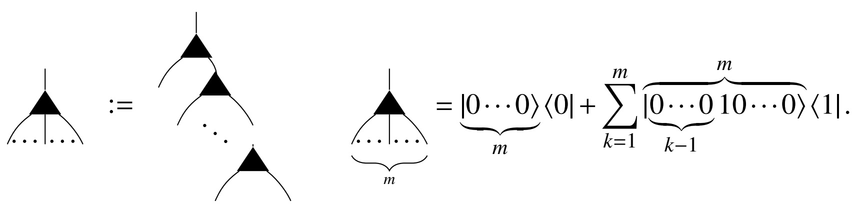
$$= |0\rangle^{\otimes m}\langle 0|^{\otimes n} + a|1\rangle^{\otimes m}\langle 1|^{\otimes n} \qquad = \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \qquad = \begin{pmatrix}1 & 0\\ 0 & 1\\ 0 & 1\\ 0 & 0\end{pmatrix}$$



$$= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix} \qquad = \begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix} \qquad = \begin{pmatrix}1\\0\\0\\1\end{pmatrix} \qquad = \begin{pmatrix}1&0&0&1\end{pmatrix},$$

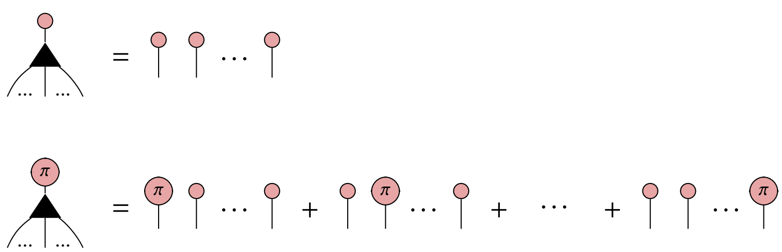
The interpretations of the additional generators can be found in the appendix.

Remark 2.1. *Due to the associative rule (Aso), we can define the W spider and give its interpretation as follows [33]:*



$$= |\underbrace{0\cdots 0}_{m}\rangle\langle 0| + \sum_{k=1}^{m} |\overbrace{\underbrace{0\cdots 0}_{k-1}10\cdots 0}^{m}\rangle\langle 1|.$$

As a consequence, we have

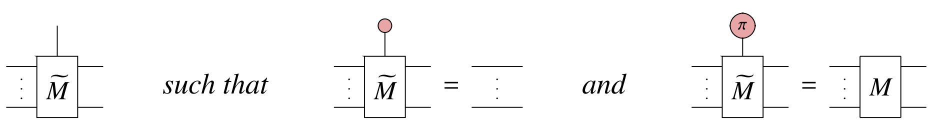


(1)

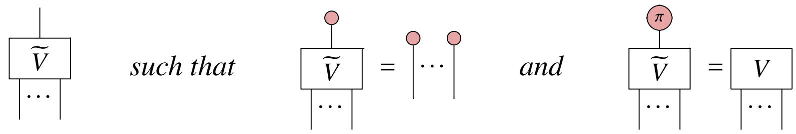
3 Controlled diagrams

We start by stating the definitions of controlled states and matrices, and how to perform operations on them. Note that our definition of controlled states are slightly different from that of Jeandel et al [17], as we send $|0\rangle$ to $|0\rangle^{\otimes n}$ instead of $|+\rangle^{\otimes n}$. Jeandel et al. also have a notion of controlled matrices by applying the map-state duality to the controlled state, which maps $|0\rangle$ to $|+\rangle^{\otimes n}\langle +|^{\otimes n}$ instead of the identity matrix as used in our definition, so their definition is not equivalent to ours.

Definition 3.1 (Controlled matrix)**.** *The controlled matrix $\widetilde{M}$ corresponding to the matrix M is a diagram*

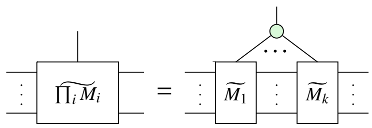


Definition 3.2 (Controlled state)**.** *The controlled state $\widetilde{V}$ corresponding to the state V is a diagram*



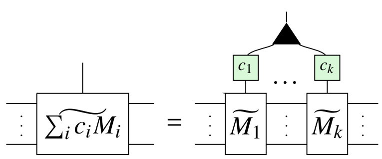
Now, we show how to construct control diagrams for sums and products of matrices, given the controlled diagrams for the original matrices.

Proposition 3.3 (Controlled product of matrices)**.** *Given controlled matrices $\widetilde{M_1},\ldots,\widetilde{M_k}$ corresponding to matrices $M_1,\ldots,M_k$, the controlled matrix for $\prod_i M_i$ is given by*



(2)

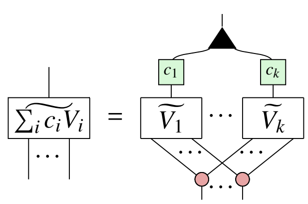
Proposition 3.4 (Controlled sum of matrices)**.** *Given controlled matrices $\widetilde{M_1},\ldots,\widetilde{M_k}$ corresponding to matrices $M_1,\ldots,M_k$ and complex numbers $c_1,\ldots,c_k$, the controlled matrix for $\sum_i c_i M_i$ is given by*



(3)

Both of these can be verified by simply plugging in the standard basis states. Similarly, we can construct the controlled diagram for sums of states.

Proposition 3.5 (Controlled sum of states)**.** *Given controlled states $\widetilde{V_1},\ldots,\widetilde{V_k}$ corresponding to states $V_1,\ldots,V_k$ and complex numbers $c_1,\ldots,c_k$, the controlled state for $\sum_i c_i V_i$ is given by*



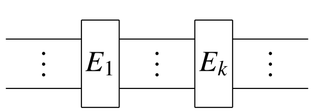
(4)

4 Realising controlled diagrams

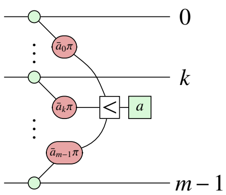
In this subsection, we show how to directly construct the controlled matrix $\tilde{M}$ given a square matrix M, and the controlled state $\tilde{\psi}$ given a state ψ.

4.1 Controlled matrix

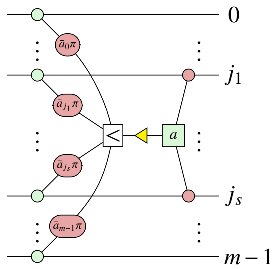
Consider a matrix M of size $2^m \times 2^m$. As given in [34], M can be represented in diagrams as follows:



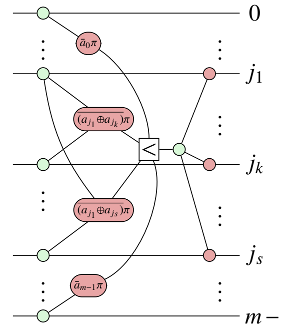
where $E_i, 1 \le i \le k$ is an elementary matrix. As a consequence of Proposition 3.3, if we know how to construct any controlled elementary matrix, then we are able to depict the controlled matrix M. It has been shown in [34] that the elementary matrix E_i must be of one of the following forms:



row multiplication



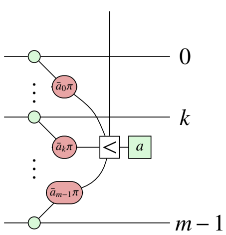
row addition



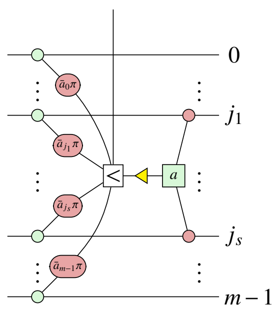
row switching

Then it can be verified by plugging standard basis that their corresponding controlled matrices can be obtained by simply adding a branch to the And-gate:

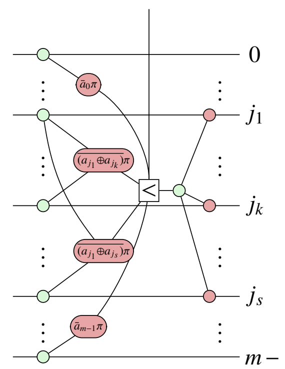
Proposition 4.1. *The controlled elementary matrices are given as:*



row multiplication



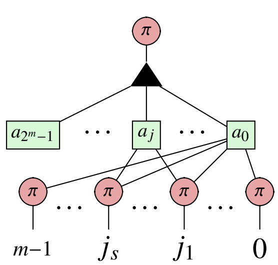
row addition



row switching

4.2 Controlled state

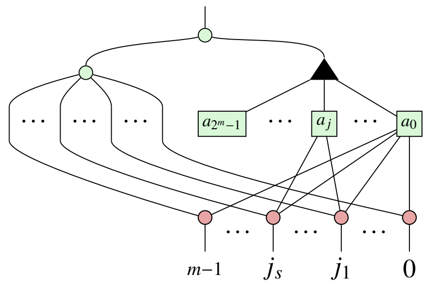
According to Wang [32], a state vector $(a_0, \ldots, a_{2^m-1})^T$ of dimension 2^m, $m \in \mathbb{N}$, can be represented in the following normal form:



(5)

Hence, we can realise any controlled state by constructing the controlled diagram of the above normal form. This is given in the following proposition.

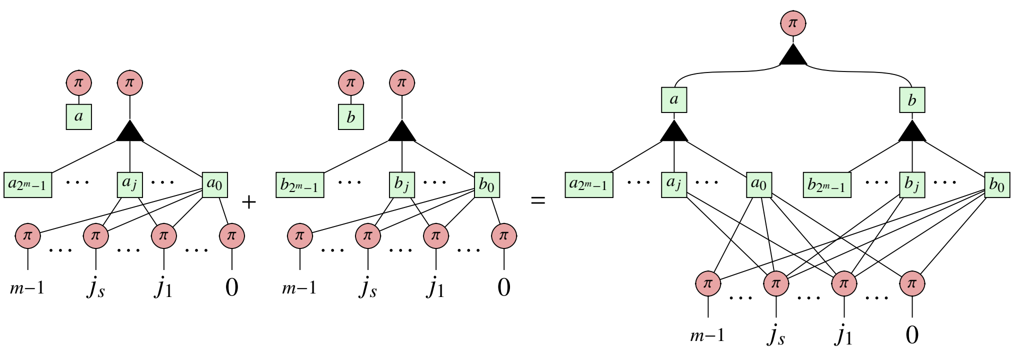
Proposition 4.2. *Controlled state for a vector* $(a_0, \ldots, a_{2^m-1})^T$ *is given by*



(6)

Using the above result and Proposition 3.5, we can derive the expression for linear combination of states, represented in the normal form.

Theorem 4.3.

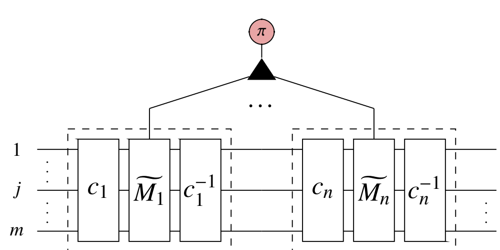


The proof of Theorem 4.3 is given in the appendix. This theorem can be generalised to sums of more than 2 terms which is given in Corollary B.10 in the appendix.

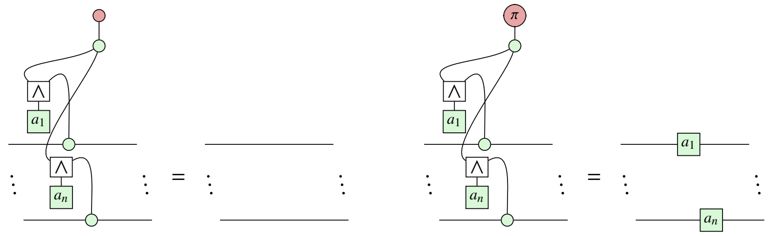
5 Representing Hamiltonians in ZXW

Here we give an efficient representation for a wide class of matrices using the ZXW calculus.

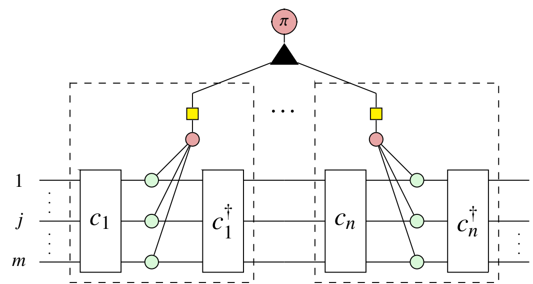
Lemma 5.1 (Sum of controlled-green-boxes)**.** *Any matrix of the form* $\sum_{i=1}^{n} c_i^{-1}\left(\bigotimes_{j=1}^{m} D(a_j)\right)c_i$*, where* $D(a) = |0\rangle\langle 0| + a|1\rangle\langle 1|$ *and* c_i *is the conjugation, can be expressed in ZXW calculus as*



where is $\tilde{M}_i$ *is the following controlled matrix:*



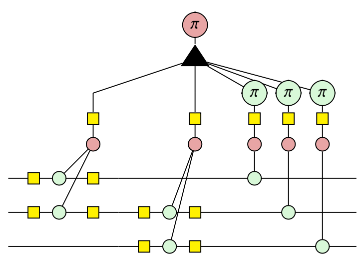
Theorem 5.2. *Any Hamiltonian $\sum_{i=1}^{n}\bigotimes_{j=1}^{m}P_{ij}$ can be expressed in ZXW calculus using controlled-Paulis.*



For each controlled-Pauli, there is a leg on the j-th qubit if $P_{ij} \neq I$, and c_i is the Clifford conjugation corresponding to the Pauli operator $P_{ij} = c_{ij}^{\dagger}Zc_{ij}$.

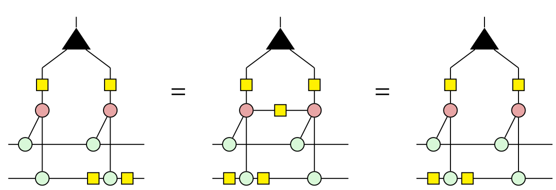
Proof. Follows from Lemma 5.1 where $a_i = e^{i0}$ or $e^{i\pi}$, corresponding to whether $P_{ij} = \pi$ or not. □

Example 5.3. *For the Hamiltonian $H = X_1X_2 + X_2X_3 - Z_1 - Z_2 - Z_3$ from [30], we have*



For an even larger example, the Hamiltonian used in [13] is shown in Figure 1.

Proposition 5.4. *The diagrammatic representation of sum of Hamiltonians in Theorem 5.2 respects commutativity of addition (i.e. $P_i + P_j = P_j + P_i$):*



The first equation in Proposition 5.4 shows that a Hadamard edge is added to the 'body' of the Pauli gadgets when their Hamiltonians anti-commute, the proof can be found in [36, Theorem 3]. While this proposition is obvious in non-diagrammatic calculations, but our goal here is to redevelop the foundations of linear algebra using diagrammatic techniques. In general, we want to fully characterise the commutative properties of controlled matrices.

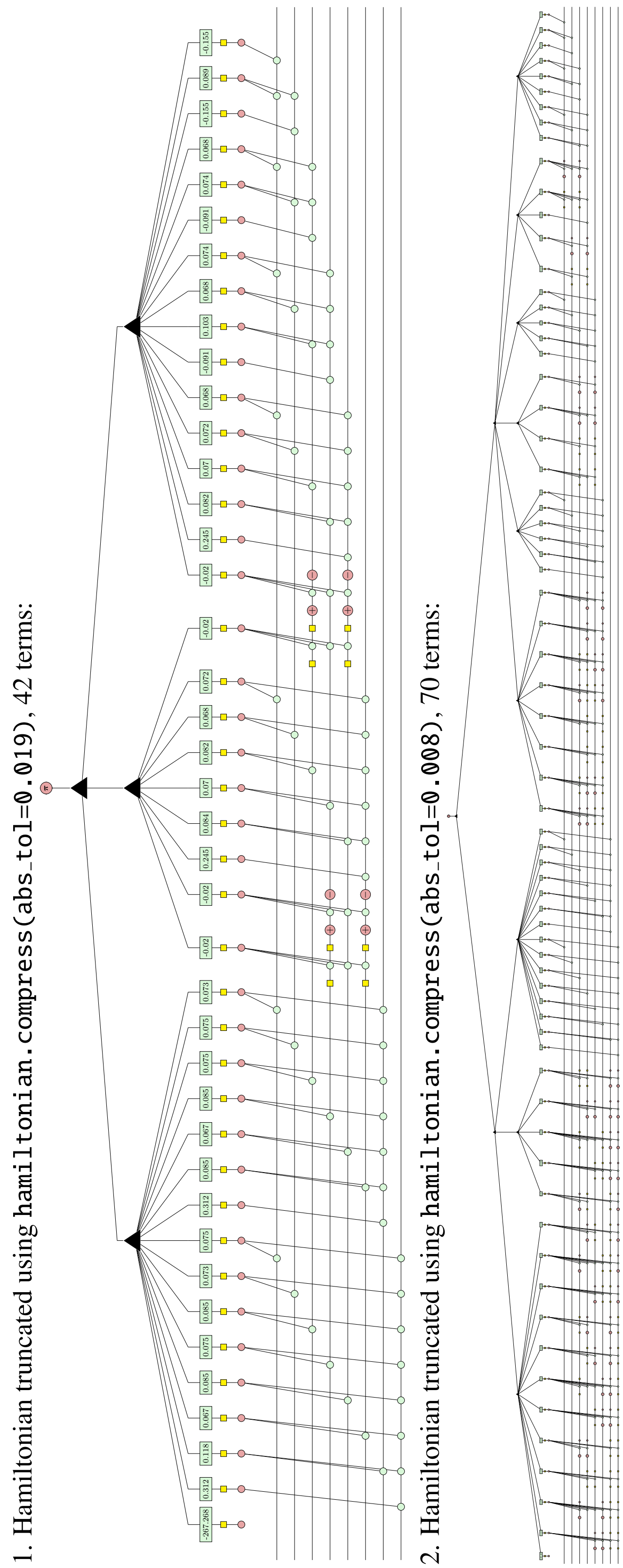
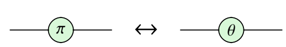
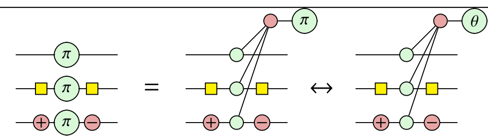
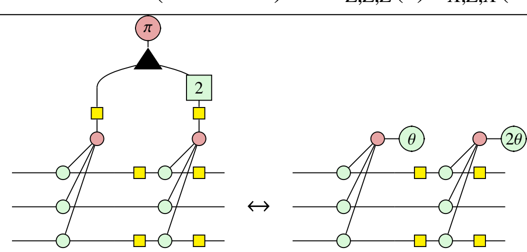


Figure 1: Hamiltonian in Greene-Diniz et al. [13]

6 Hamiltonian Exponentiation

According to Stone's theorem [28], the time-evolution operator of a Hamiltonian is the one-parameter unitary group generated by the Hamiltonian. Tasks such as Hamiltonian simulation require finding a unitary circuit that approximates the time evolution operator e^{-iHt} for a given Hamiltonian H. In general, finding such unitary circuit is a hard problem. Developing a method to obtain the one-parameter unitary group of Hamiltonian diagrams in ZXW will allow us tackle problems from quantum chemistry using diagrammatic tools. We could further use the rules of ZXW calculus to rewrite a Hamiltonian exponential diagram to a unitary time-evolution circuit.

For Hamiltonians comprised of only commuting terms, we have a simple correspondence between the diagrams of Hamiltonians and their exponentials. In the following table, we first show the correspondence between Pauli matrices and their exponentials; then we show the correspondence between sums of Pauli matrices and their exponentials. For more details on how to calculate this correspondence in ZX see [17, 29, 36].

$Z \leftrightarrow R_z(\theta)$	$Z \otimes X \otimes Y \leftrightarrow \Phi_{Z,X,Y}(\theta)$
	
$Z \otimes Z \otimes Z + 2 * (X \otimes Z \otimes X) \leftrightarrow \Phi_{Z,Z,Z}(\theta)\Phi_{X,Z,X}(2\theta)$	
	

In most of the interesting examples, the Hamiltonian contains some non-commuting terms. To exponentiate an arbitrary Hamiltonian, we need to use more sophisticated machinery. We use the Cayley-Hamilton theorem to obtain a closed form representation for a matrix exponential function. It allows us to express a matrix exponential function as a polynomial of order $n-1$ for a $n \times n$ matrix.

Here, we provide a brief summary of Cayley-Hamilton theorem to calculate matrix exponential functions. For detailed description, the reader may refer to [14, 25, 26].

Theorem 6.1 (Cayley-Hamilton theorem)**.** *The characteristic polynomial $p_A(\lambda)$ of a $n \times n$ matrix A is polynomial of degree n given by*

$$p_A(\lambda) = \det(\lambda I - A) \tag{7}$$

This polynomial expression, when evaluated on A, gives the zero matrix, i.e. $p_A(A) = 0$.

The Cayley-Hamilton theorem provides a relation between powers of a square matrix. If A is a $n \times n$ matrix, we can express A^m, for any $m \geq n$, as a linear combination of A^k, where $k = 0, \ldots, n-1$. Since the exponential function

$$e^{At} = \sum_{k=0}^{\infty} \frac{A^k t^k}{k!} \tag{8}$$

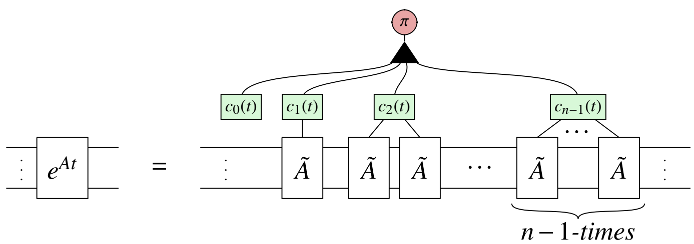
is defined as a polynomial of A, it can be expressed as

$$e^{At} = c_0(t)I + c_1(t)A + \cdots + c_{n-1}(t)A^{n-1} \tag{9}$$

for some $c_0, \ldots, c_{n-1}$. These coefficient functions can be easily calculated using Putzer's algorithm [26].

Now, we can present the general form of the matrix exponential function in ZXW.

Theorem 6.2. *For any matrix A of size $n \times n$, the exponential function e^{At} can be represented using the following ZXW diagram:*

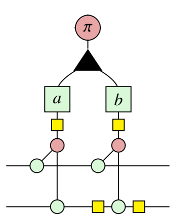


(10)

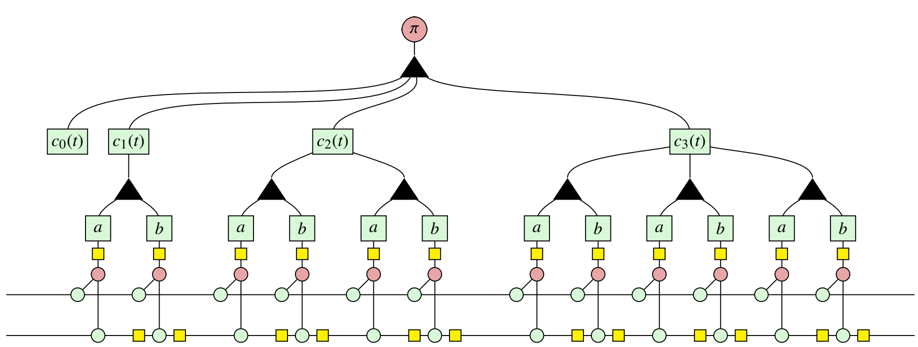
where $\tilde{A}$ is the controlled diagram of matrix A and $c_0, \ldots, c_{n-1}$ are some functions of t.

Proof. Follows from Theorem 6.1, Proposition 3.3, and Proposition 3.4. □

Example 6.3. *Consider the Hamiltonian $H = aZ_1Z_2 + bZ_1X_2$. The diagram for this Hamiltonian is*



The exponential of this Hamiltonian e^{Ht} is the following diagram



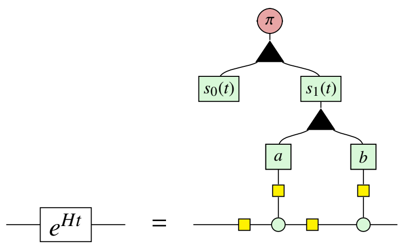
where $c_0, \ldots, c_3$ are the coefficient functions.

Now, we demonstrate the utility of the ZXW calculus by using its rules to rewrite a Hamiltonian exponential diagram to a quantum circuit.

Example 6.4. *Consider the following Hamiltonian:*

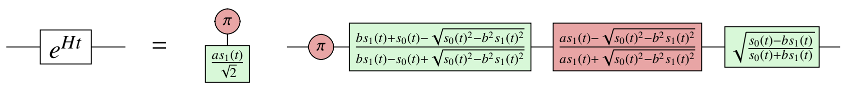
$$H = aX + bZ \tag{11}$$

for any complex a and b. There exists $s_0(t)$ and $s_1(t)$ such that



(12)

Using ZXW, we can rewrite it to the following circuit. The full simplification steps are shown in Appendix C.



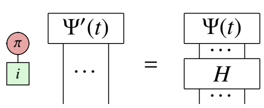
(13)

7 Schrödinger Equation

We recall the Schrödinger equation:

$$i\frac{\partial}{\partial t}|\Psi(t)\rangle = H|\Psi(t)\rangle \tag{14}$$

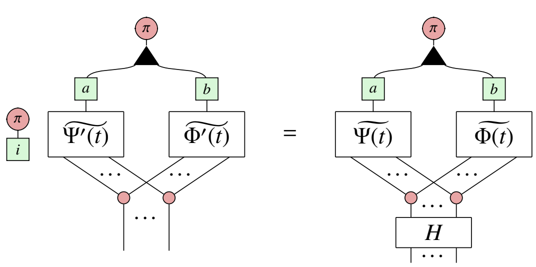
where t is time, $|\Psi(t)\rangle$ is the state vector of the quantum system in question, and H is a Hamiltonian operator. We can write the Schrödinger equation diagrammatically as:



(15)

where $|\Psi(t)\rangle$ can be expressed using the normal form and $|\Psi'(t)\rangle$ can be obtained by applying the differentiation gadget [35] to $|\Psi(t)\rangle$. Using this, we can show that any linear combination of solutions of Schrödinger equation is also a solution.

Theorem 7.1. *Assume that $\Psi(t)$ and $\Phi(t)$ satisfy the Schrödinger equation (14) and a, b are arbitrary complex numbers, then so does $a\Psi(t) + b\Phi(t)$, i.e.,*



(16)

The proof is given in the appendix.

8 Future work

In this paper, we give direct representations of controlled diagrams for a wide class of matrices and show how to sum them. As applications, we express the Hamiltonians used for carbon capture in [13], convert between a Hamiltonian and its one-parameter unitary group, and formulate Schrödinger equation in ZX and show the linearity of its solutions.

We would like to apply the summation techniques developed in this paper to practical problems, like quantum approximate optimisation, integration on arbitrary ZX diagrams, or quantum machine learning. Also we would like to develop tools to rewrite Hamiltonian exponentiation diagrams to quantum circuits.

Acknowledgements

We would like to thank Gabriel Greene-Diniz and David Zsolt Manrique for providing the Pauli operators form of the Hamiltonian from their paper [13]. We would also like to thank Bob Coecke, Stefano Gogioso, Konstantinos Meichanetzidis, Vincent Wang and Robin Lorenz for the feedback on this paper.

References

[1] Samson Abramsky & Bob Coecke (2004): *A Categorical Semantics of Quantum Protocols*. In: *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science*, LICS '04, IEEE Computer Society, Washington, DC, USA, pp. 415–425. Available at https://doi.org/10.1109/LICS.2004.1. doi:10.1109/LICS.2004.1.

[2] 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, pp. 11:1–11:23, doi:10.4230/LIPIcs.TQC.2020.11.

[3] Titouan Carette, Dominic Horsman & Simon Perdrix (2019): *SZX-Calculus: Scalable Graphical Quantum Reasoning*. In Peter Rossmanith, Pinar Heggernes & 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. Available at http://drops.dagstuhl.de/opus/volltexte/2019/10999.

[4] Nicholas Chancellor, Aleks Kissinger, Joschka Roffe, Stefan Zohren & Dominic C. Horsman (2016): *Graphical Structures for Design and Verification of Quantum Error Correction*. *arXiv: Quantum Physics*.

[5] Bob Coecke & Ross Duncan (2011): *Interacting quantum observables: categorical algebra and diagrammatics*. *New Journal of Physics* 13(4), p. 043016. Available at http://stacks.iop.org/1367-2630/13/i=4/a=043016. doi:10.1088/1367-2630/13/4/043016.

[6] Bob Coecke, Giovanni de Felice, Konstantinos Meichanetzidis & Alexis Toumi (2020): *Foundations for near-term quantum natural language processing*. *arXiv preprint arXiv:2012.03755*.

[7] Bob Coecke & Aleks Kissinger (2010): *The Compositional Structure of Multipartite Quantum Entanglement*. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide & Paul G. Spirakis, editors: *Automata, Languages and Programming*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 297–308. doi:10.1007/978-3-642-14162-1_25.

[8] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons & Seyon Sivarajah (2020): *Phase Gadget Synthesis for Shallow Circuits*. *Electronic Proceedings in Theoretical Computer Science* 318, p. 213?228, doi:10.4204/eptcs.318.13.

[9] 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.

[10] Stefano Gogioso (2015): *Categorical Semantics for Schrödinger's Equation*. arXiv:1501.06489.

[11] Stefano Gogioso (2017): *Categorical Quantum Dynamics*. Ph.D. thesis, University of Oxford. arXiv:1709.09772.

[12] Stefano Gogioso (2019): *A Diagrammatic Approach to Quantum Dynamics*. In: 8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019).

[13] Gabriel Greene-Diniz, David Zsolt Manrique, Wassil Sennane, Yann Magnin, Elvira Shishenina, Philippe Cordier, Philip Llewellyn, Michal Krompiec, Marko J Rančić & David Muñoz Ramo (2022): *Modelling Carbon Capture on Metal-Organic Frameworks with Quantum Computing*. arXiv preprint arXiv:2203.15546.

[14] Werner Greub (1975): *Linear algebra*, 4 edition. Graduate Texts in Mathematics, Springer.

[15] Amar Hadzihasanovic (2015): *A Diagrammatic Axiomatisation for Qubit Entanglement*. In: 2015 30th Annual ACM/IEEE Symposium on Logic in Computer Science, pp. 573–584. doi:10.1109/LICS.2015.59.

[16] Amar Hadzihasanovic (2017): *The algebra of entanglement and the geometry of composition*. Ph.D. thesis, University of Oxford. arXiv:1709.08086.

[17] Emmanuel Jeandel, Simon Perdrix & Margarita Veshchezerova (2022): *Addition and Differentiation of ZX-diagrams*. arXiv preprint arXiv:2202.11386.

[18] Emmanuel Jeandel, Simon Perdrix & Renaud Vilmart (2019): *A Generic Normal Form for ZX-Diagrams and Application to the Rational Angle Completeness*. In: 34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2019, Vancouver, BC, Canada, June 24-27, 2019, pp. 1–10, doi:10.1109/LICS.2019.8785754. Available at `https://doi.org/10.1109/LICS.2019.8785754`.

[19] Aleks Kissinger & John van de Wetering (2020): *PyZX: Large Scale Automated Diagrammatic Reasoning*. Electronic Proceedings in Theoretical Computer Science 318, pp. 229–241, doi:10.4204/eptcs.318.14. Available at `https://doi.org/10.4204%2Feptcs.318.14`.

[20] Aleks Kissinger & John van de Wetering (2020): *Reducing the number of non-Clifford gates in quantum circuits*. Phys. Rev. A 102, p. 022406, doi:10.1103/PhysRevA.102.022406. Available at `https://link.aps.org/doi/10.1103/PhysRevA.102.022406`.

[21] Aleks Kissinger, John van de Wetering & Renaud Vilmart (2022): *Classical simulation of quantum circuits with partial and graphical stabiliser decompositions*.

[22] Louis Lemonnier, John van de Wetering & Aleks Kissinger (2021): *Hypergraph Simplification: Linking the Path-sum Approach to the ZH-calculus*. Electronic Proceedings in Theoretical Computer Science.

[23] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis & Bob Coecke (2021): *Qnlp in practice: Running compositional models of meaning on a quantum computer*. arXiv preprint arXiv:2102.12846.

[24] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin & Xiao Yuan (2020): *Quantum computational chemistry*. Rev. Mod. Phys. 92, p. 015003, doi:10.1103/RevModPhys.92.015003. Available at `https://link.aps.org/doi/10.1103/RevModPhys.92.015003`.

[25] Héctor Manuel Moya-Cessa & Francisco Soto-Eguibar (2011): *Differential equations: an operational approach*. Rinton Press, Incorporated.

[26] Eugene J Putzer (1966): *Avoiding the Jordan canonical form in the discussion of linear systems with constant coefficients*. The American Mathematical Monthly 73(1), pp. 2–7.

[27] Tobias Stollenwerk & Stuart Hadfield (2022): *Diagrammatic Analysis for Parameterized Quantum Circuits*.

[28] Marshall H Stone (1932): *On one-parameter unitary groups in Hilbert space*. Annals of Mathematics, pp. 643–648.

[29] Alexis Toumi, Richie Yeung & Giovanni de Felice (2021): *Diagrammatic Differentiation for Quantum Machine Learning*. In Chris Heunen & Miriam Backens, editors: Proceedings 18th International Conference

on Quantum Physics and Logic, QPL 2021, Gdansk, Poland, and online, 7-11 June 2021, EPTCS 343, pp. 132–144, doi:10.4204/EPTCS.343.7. Available at `https://doi.org/10.4204/EPTCS.343.7`.

[30] Christopher J Turner, Konstantinos Meichanetzidis, Zlatko Papić & Jiannis K Pachos (2017): *Optimal free descriptions of many-body theories*. Nature communications 8(1), pp. 1–7.

[31] Quanlong Wang (2020): *An algebraic axiomatisation of ZX-calculus*. Proceedings of the 17th International Conference on Quantum Physics and Logic (QPL) 2020. arXiv:1911.06752.

[32] Quanlong Wang (2020): *Algebraic complete axiomatisation of ZX-calculus with a normal form via elementary matrix operations*. arXiv:2007.13739v3.

[33] Quanlong Wang (2021): *A non-anyonic qudit ZW-calculus*. arXiv:2109.11285.

[34] Quanlong Wang (2021): *Representing Matrices Using Algebraic ZX-calculus*. arXiv preprint arXiv:2110.06898.

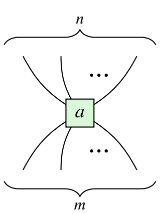
[35] Quanlong Wang & Richie Yeung (2022): *Differentiating and Integrating ZX Diagrams*. arXiv preprint arXiv:2201.13250.

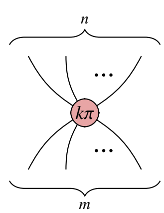
[36] Richie Yeung (2020): *Diagrammatic Design and Study of Ans\"{a} tze for Quantum Machine Learning*. arXiv preprint arXiv:2011.11073.

[37] 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. Available at `https://doi.org/10.22331/q-2021-06-04-466`.

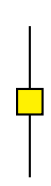
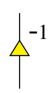
A Interpretation

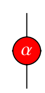
Although the generators in ZXW calculus are formal mathematical objects in their own right, in the context of this paper we interpret the generators as linear maps, so each ZXW diagram is equivalent to a vector or matrix.



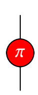
$= |0\rangle^{\otimes m}\langle 0|^{\otimes n} + a|1\rangle^{\otimes m}\langle 1|^{\otimes n}$,


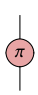
$= \sum_{\sum_i \bar{x}_i + \sum_i \bar{y}_i \equiv k (mod\ 2)} |x\rangle\langle y|, k \in \{0,1\}$,

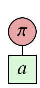
 $= \frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$,  $= \begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}$, 

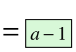
$= \begin{pmatrix}1 & -1\\ 0 & 1\end{pmatrix}$, 

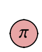
$= e^{i\frac{\alpha}{2}}\begin{pmatrix}\cos\frac{\alpha}{2} & -i\sin\frac{\alpha}{2}\\ -i\sin\frac{\alpha}{2} & \cos\frac{\alpha}{2}\end{pmatrix}$,



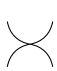
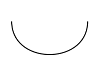
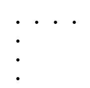
$=$ 

$= \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$, 

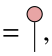
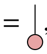
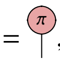
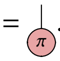
$=$ 

$= a$, 

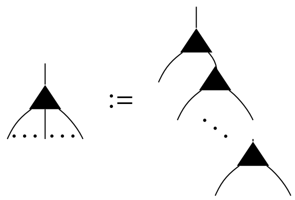
$= 0$, $= \begin{pmatrix}1 & 0\\ 0 & 1\end{pmatrix}$,  $= \begin{pmatrix}1 & 0\\ 0 & 1\\ 0 & 1\\ 0 & 0\end{pmatrix}$,

 $= \begin{pmatrix}1 & 0 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$,  $= \begin{pmatrix}1\\ 0\\ 0\\ 1\end{pmatrix}$,  $= \begin{pmatrix}1 & 0 & 0 & 1\end{pmatrix}$,  $= 1$,

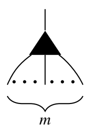
where

$$a \in \mathbb{C}, \quad \alpha \in \mathbb{R}, \quad |0\rangle = \begin{pmatrix}1\\0\end{pmatrix} =$$  $, \quad \langle 0| = \begin{pmatrix}1 & 0\end{pmatrix} =$  $, \quad |1\rangle = \begin{pmatrix}0\\1\end{pmatrix} =$  $, \quad \langle 1| = \begin{pmatrix}0 & 1\end{pmatrix} =$ .

Remark A.1. *Due to the associative rule (Aso), we can define the W spider*

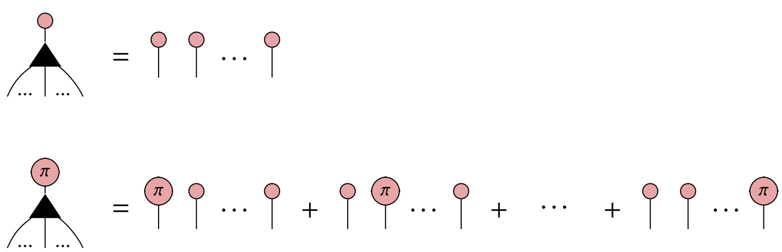


and give its interpretation as follows [33]:



$$= \underbrace{|0\cdots0\rangle}_{m}\langle 0| + \sum_{k=1}^{m} \overbrace{\underbrace{|0\cdots0}_{k-1}10\cdots0\rangle}^{m}\langle 1|.$$

As a consequence, we have



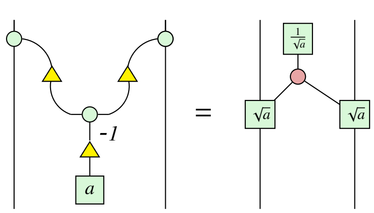
(17)

B Proofs and Lemmas

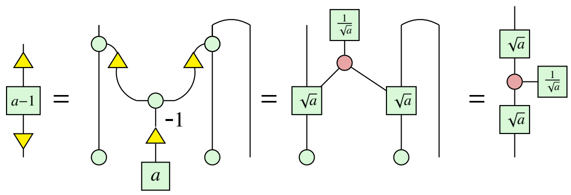
In this appendix, we include all the lemmas which have been essentially existed (up to scalars) in previous papers. The lemmas are given in the order which they appear in this paper.

B.1 Useful lemmas

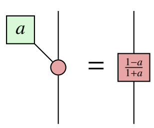
Lemma B.1. *Suppose $a \neq 0, a \in \mathbb{C}$. Then*



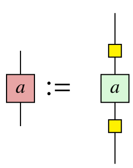
This equality can be verified by plugging in the standard basis on the inputs of the diagrams. As a consequence, we have



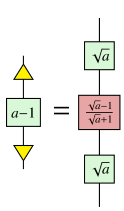
Also one can check that

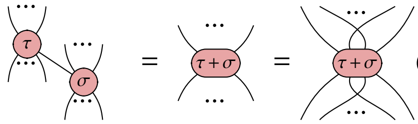
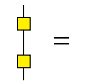
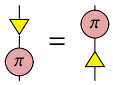
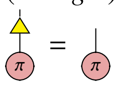
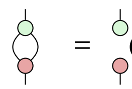
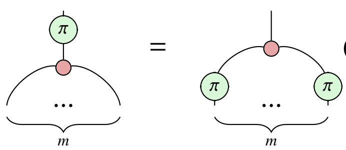
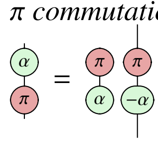


where

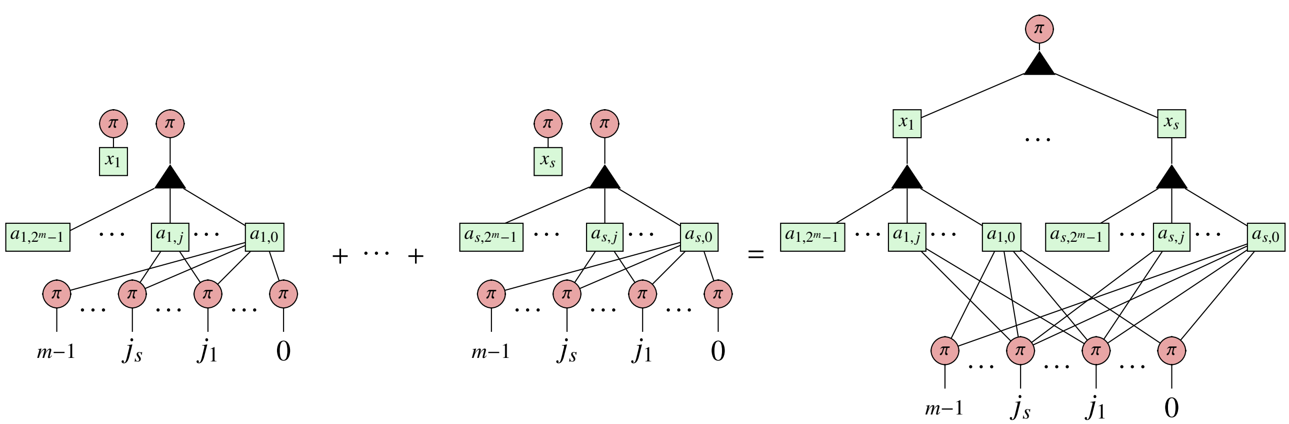


Therefore

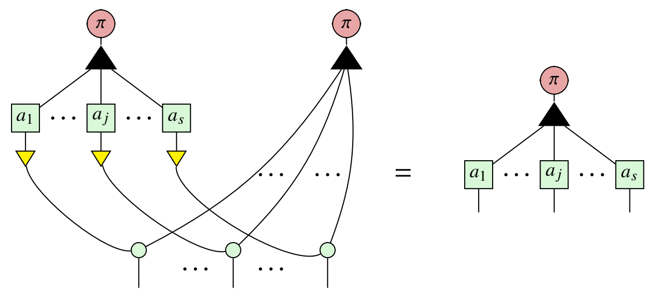


Lemma B.2. *[32]* *For* $\tau,\sigma \in \{0,\pi\}$*, pink spiders fuse.*  $(S1r)$	**Lemma B.3.** *[32]* *Hadamard is involutive.*  *(H2)*
Lemma B.4. *[32]* *Pink* π *transposes the triangle.* 	**Lemma B.5.** *[32]* *Green* π *inverts the triangle.* 
Lemma B.6. *[32]* $(triangle)^T$ *stabilises* $\|1\rangle$*.* 	**Lemma B.7.** *[32]* *Hopf rule.*  $(Hopf)$
Lemma B.8. *[32]* π *copy rule. For* $m \geq 0$*:*  (Pic)	**Lemma B.9.** *[32]* π *commutation rule.* 

Corollary B.10.

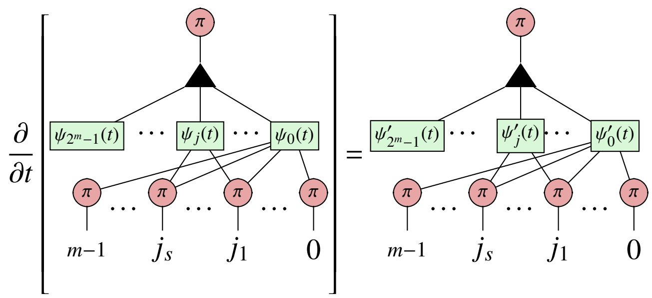


Lemma B.11.



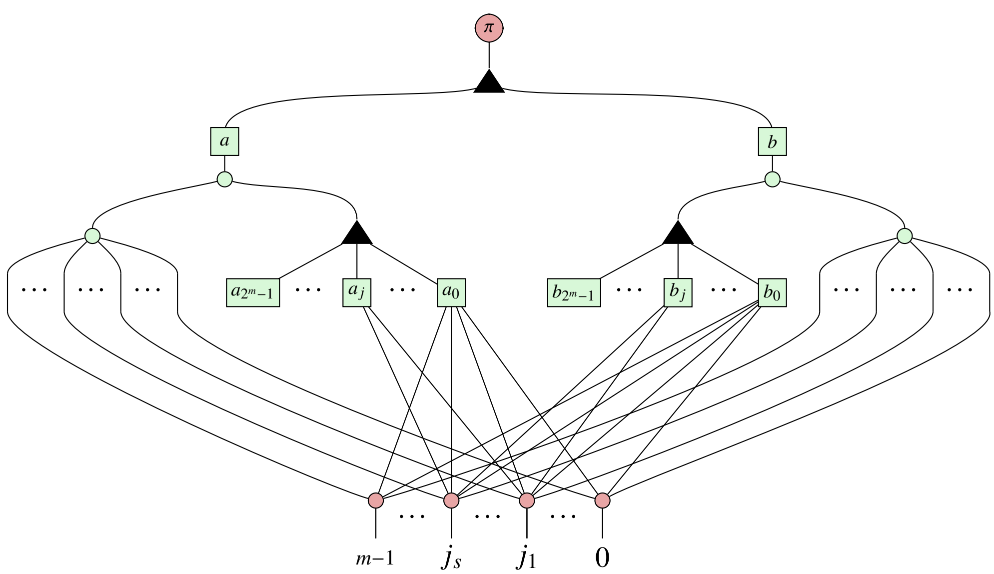
where $a_1, \cdots, a_s$ are arbitrary complex numbers. This equality can be verified by plugging standard basis on the outputs.

Lemma B.12.

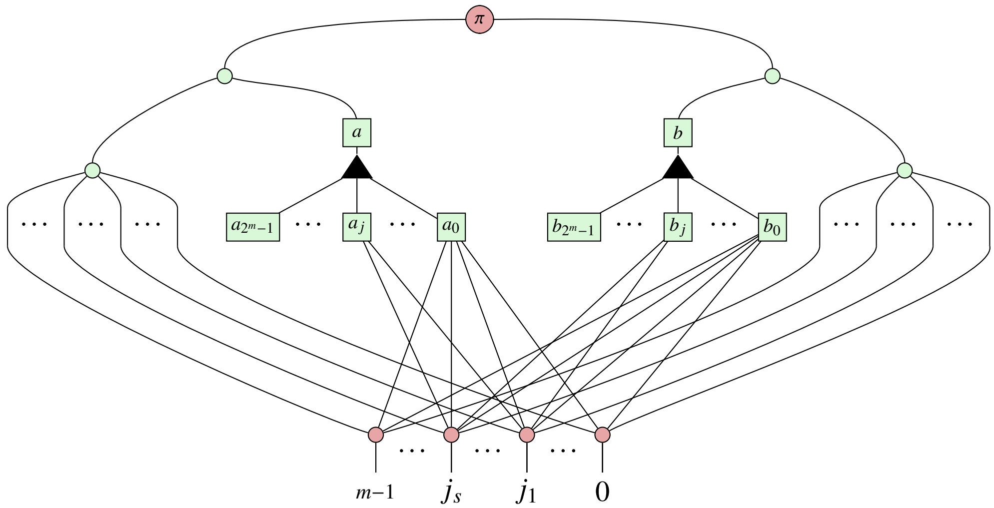


B.2 Proofs

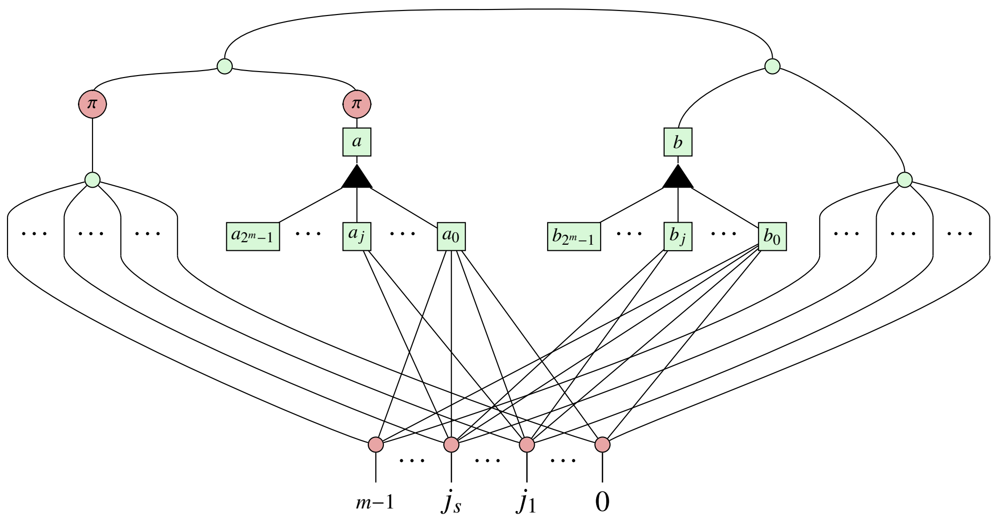
Proof of Theorem 4.3. Using Proposition 3.5, we can write the LHS as



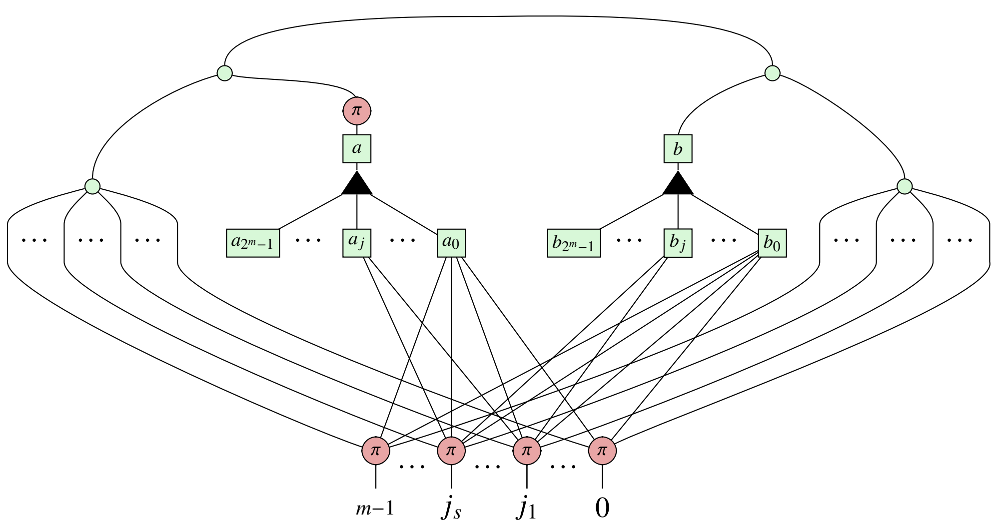
=


π
a
b
a2m−1
aj
a0
b2m−1
bj
b0
m−1
js
j1
0

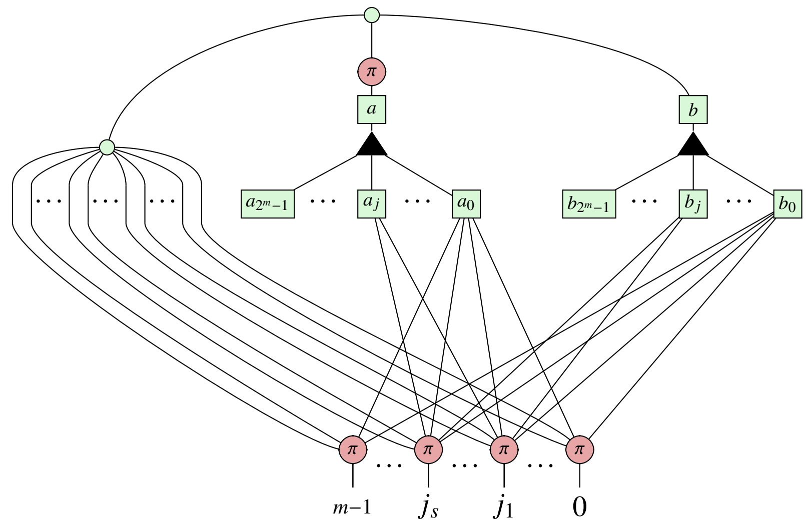
=


π
π
a
b
a2m−1
aj
a0
b2m−1
bj
b0
m−1
js
j1
0

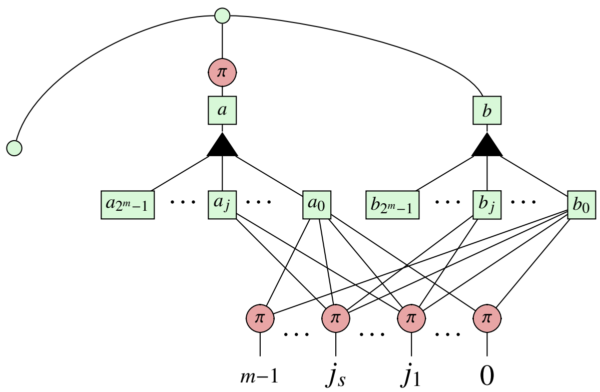
=


π
a
b
a2m−1
aj
a0
b2m−1
bj
b0
π
π
π
π
m−1
js
j1
0

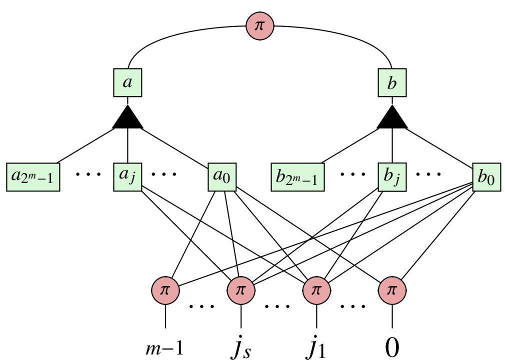
=

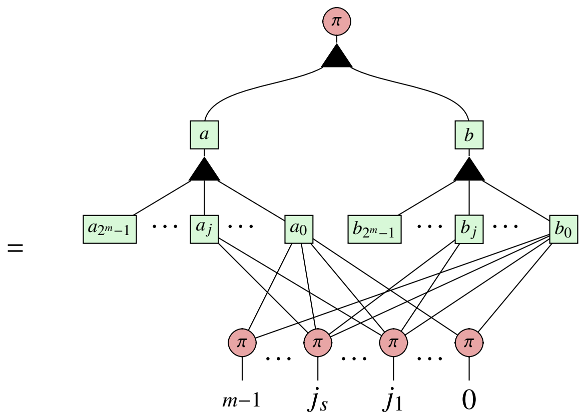

π
a
b
a_{2^m-1} ⋯ a_j ⋯ a_0
b_{2^m-1} ⋯ b_j ⋯ b_0
π ⋯ π ⋯ π ⋯ π
$m-1$ j_s j_1 0

=


π
a
b
a_{2^m-1} ⋯ a_j ⋯ a_0
b_{2^m-1} ⋯ b_j ⋯ b_0
π ⋯ π ⋯ π ⋯ π
$m-1$ j_s j_1 0

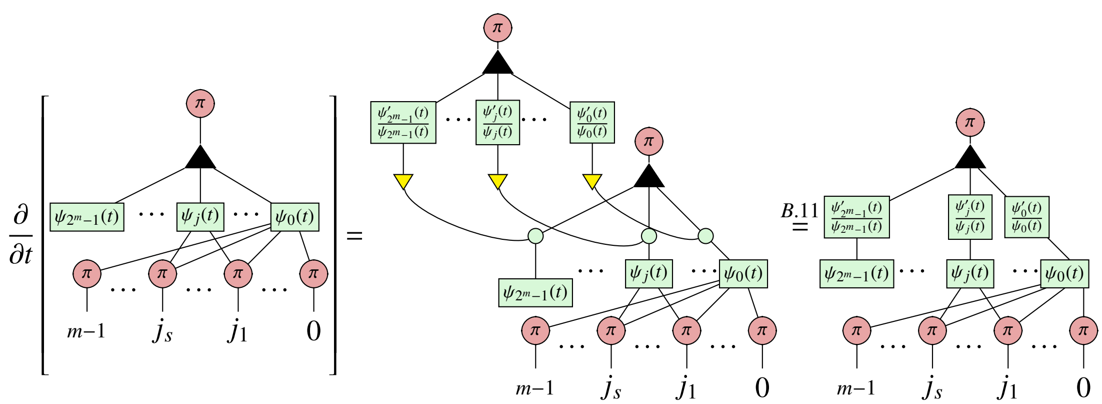
=

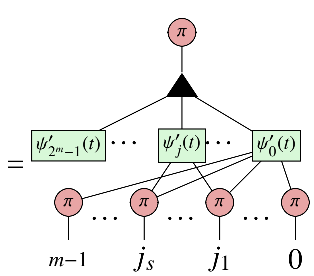

π
a
b
a_{2^m-1} ⋯ a_j ⋯ a_0
b_{2^m-1} ⋯ b_j ⋯ b_0
π ⋯ π ⋯ π ⋯ π
$m-1$ j_s j_1 0



□

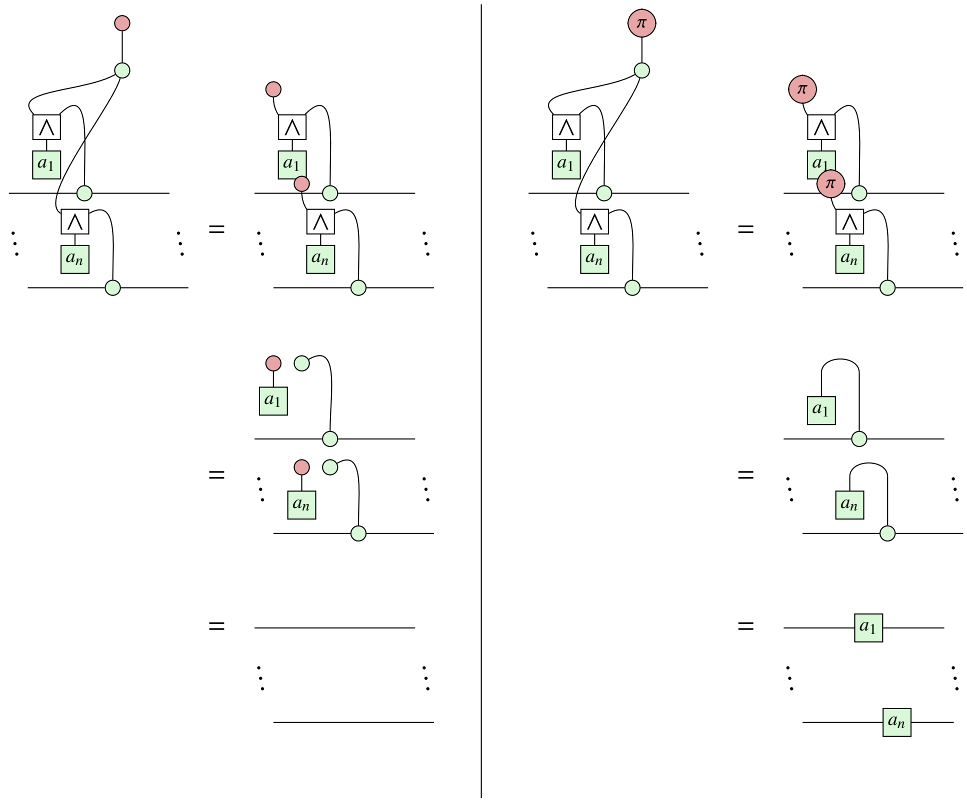
Proof of Lemma B.12.





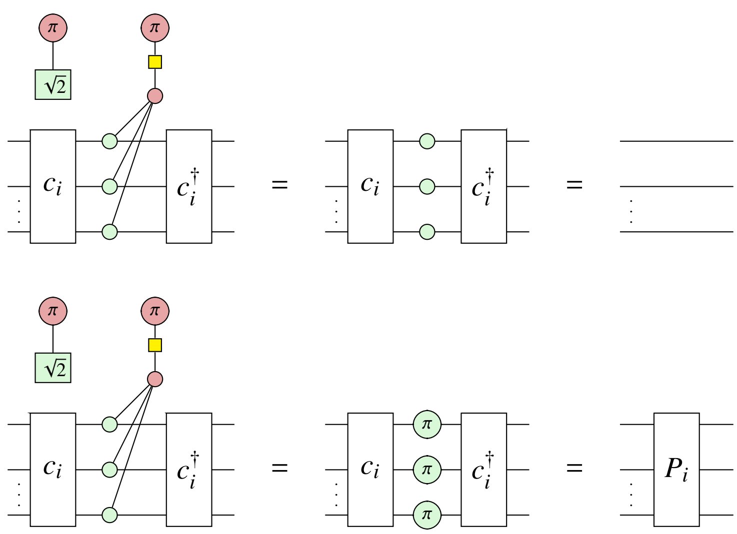
□

Proof of Lemma 5.1. Use basic properties of the AND gate.



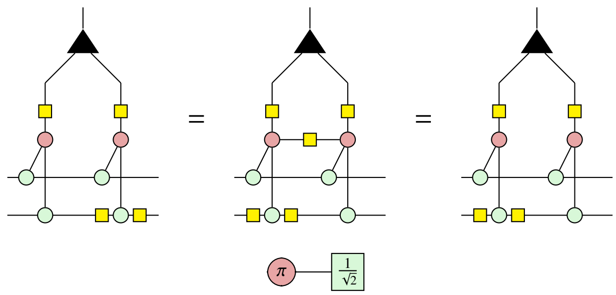
□

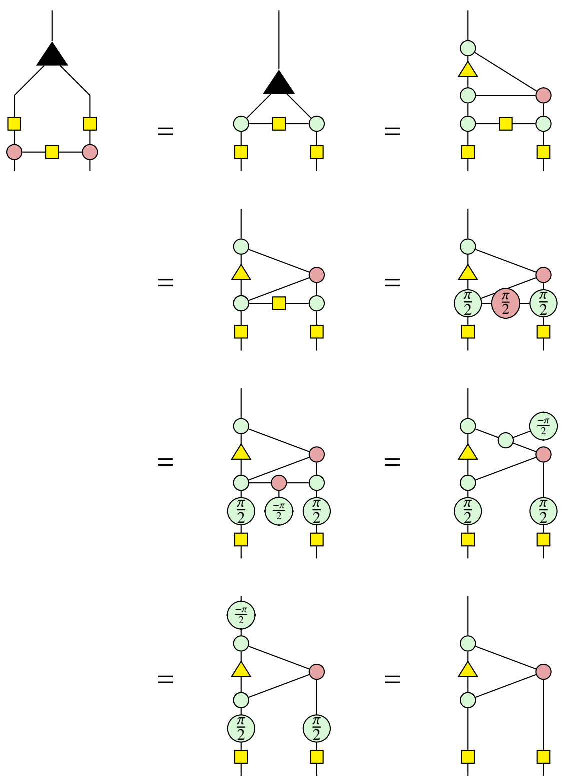
Proof of Theorem 5.2. We first verify that the subdiagrams connected to the W spider are indeed a controlled-Paulis:



Since $|W\rangle_n = |1\dots0\rangle + \dots + |0\dots1\rangle$, the overall diagram is equal to $\sum_i P_i$. □

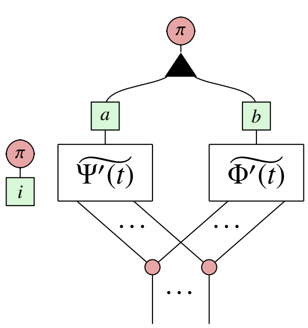
Proof of Proposition 5.4.

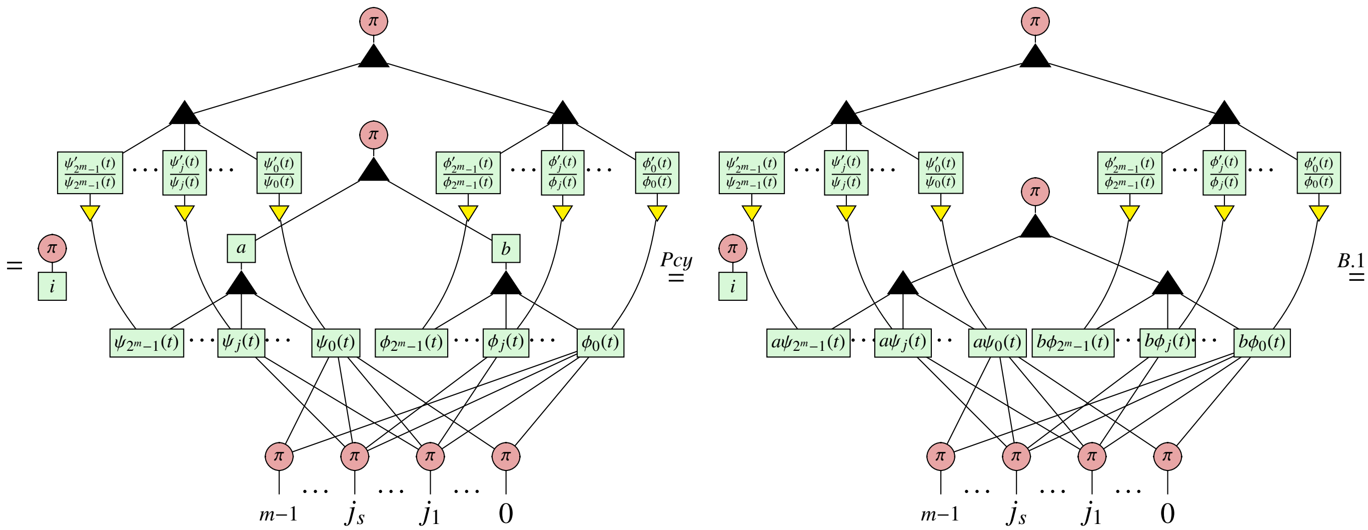


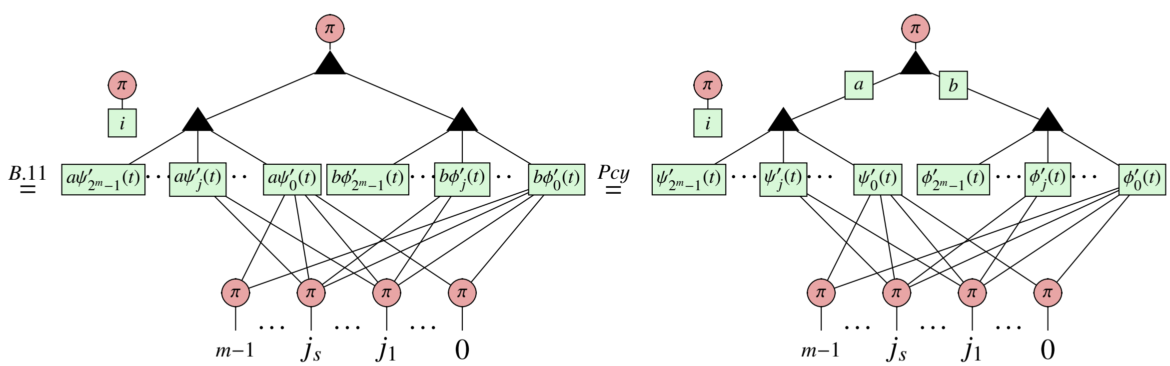


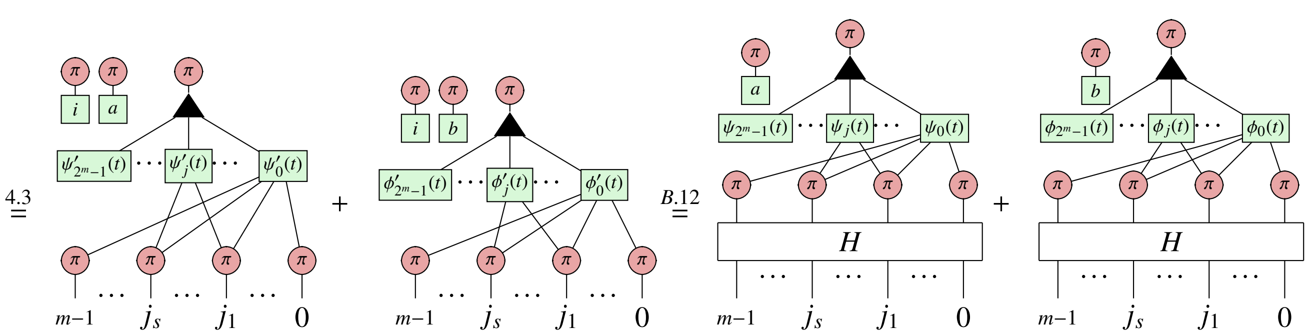
□

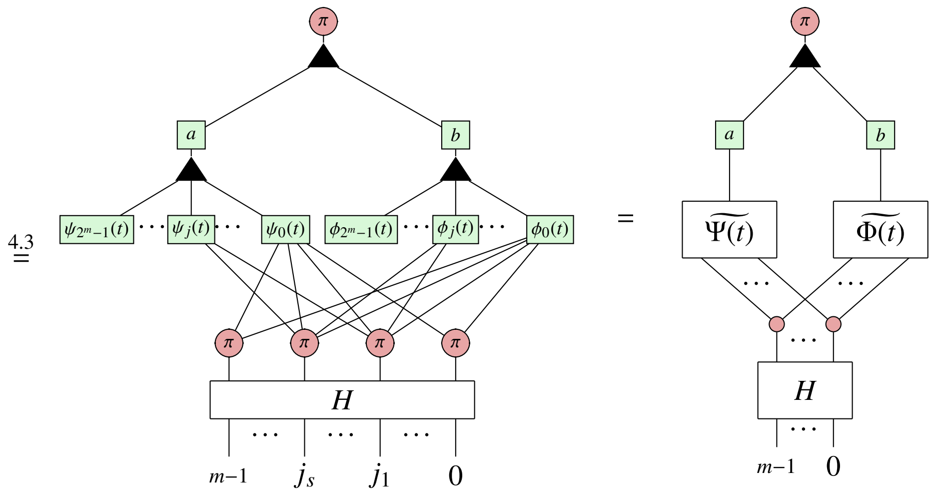
Proof of Theorem 7.1.




=
Pcy
=
B.11
=
m−1
j_s
j_1
0


B.11
=
Pcy
=
m−1
j_s
j_1
0


4.3
=
+
B.12
=
H
+
H
m−1
j_s
j_1
0

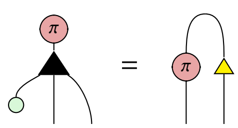


□

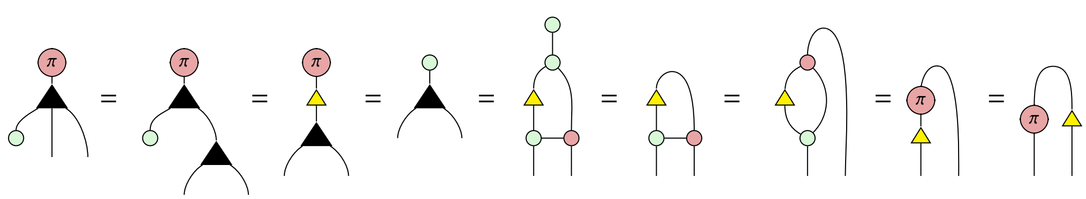
C Circuit extraction of the exponential from Equation 12

To simplify this diagram to a circuit, we will use the following two propositions.

Proposition C.1.

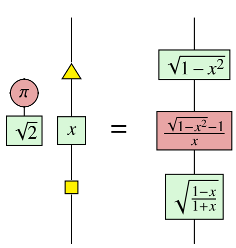


Proof of Proposition C.1.

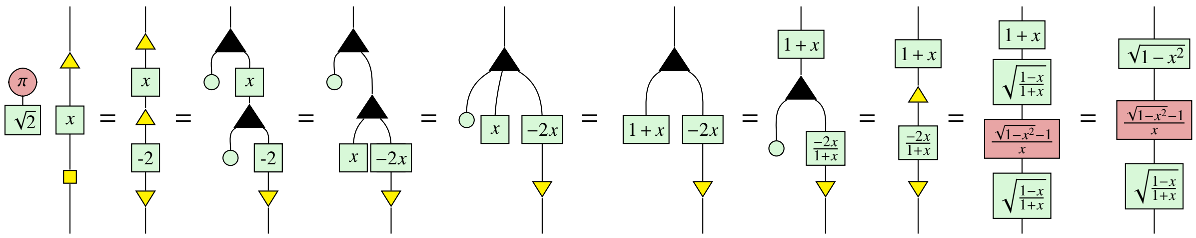


□

Proposition C.2.



Proof of Proposition C.2.



□

Now, we begin simplifying (12).