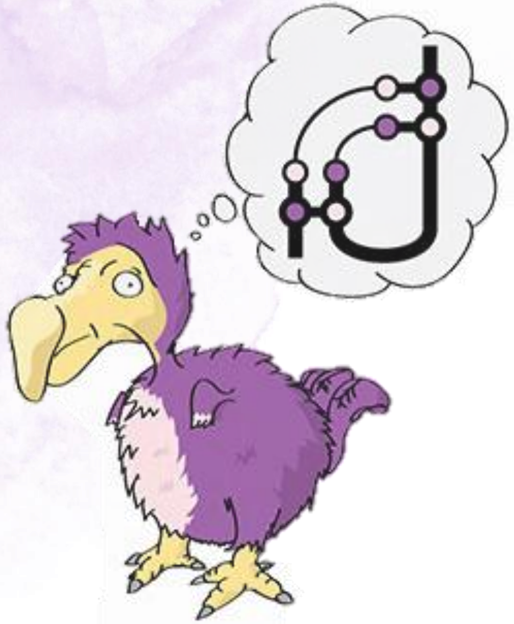
Focked-up ZX Calculus

Picturing continuous-variable quantum computation



Razin A. Shaikh, Lia Yeh and Stefano Gogioso
University of Oxford

arxiv: 2406.02905

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF OXFORD

Plan

1. Background and motivation

2. Focked-up ZX calculus
    - Generators, rules and common gates
    - Gaussian completeness

3. Applications
    - Quantum Error Correction with the GKP code
    - Gaussian Boson sampling

4. Conclusion and future work

Plan

1. Background and motivation

2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness

3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling

4. Conclusion and future work

Qubit quantum computing

Qubit quantum computing

Symmetric monoidal category of finite dimensional Hilbert spaces

Qubit quantum computing

Symmetric monoidal category of finite dimensional Hilbert spaces

States   Vectors in 2^n dimensional complex Hilbert space

Qubit quantum computing

Symmetric monoidal category of finite dimensional Hilbert spaces

States	Vectors in 2^n dimensional complex Hilbert space
Operations	Linear maps

Qubit quantum computing

Symmetric monoidal category of finite dimensional Hilbert spaces

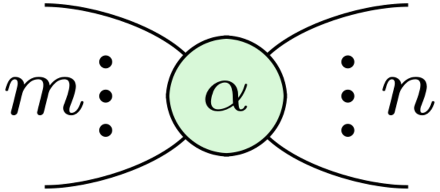
States	Vectors in 2^n dimensional complex Hilbert space
Operations	Linear maps
Sequential composition	Matrix multiplication

Qubit quantum computing

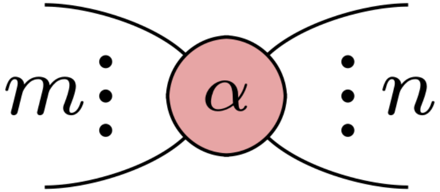
Symmetric monoidal category of finite dimensional Hilbert spaces

States	Vectors in 2^n dimensional complex Hilbert space
Operations	Linear maps
Sequential composition	Matrix multiplication
Parallel composition	Tensor product

Qubit ZX calculus



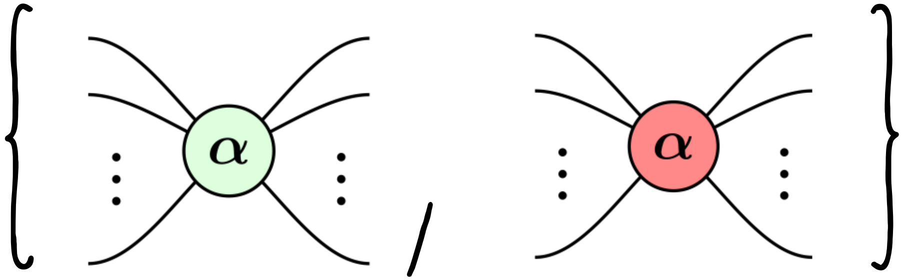
$$\xmapsto{[\![\cdot]\!]} \quad |0\rangle^{\otimes n} \langle 0|^{\otimes m} + e^{i\alpha} |1\rangle^{\otimes n} \langle 1|^{\otimes m}$$



$$\xmapsto{[\![\cdot]\!]} \quad |+\rangle^{\otimes n} \langle +|^{\otimes m} + e^{i\alpha} |-\rangle^{\otimes n} \langle -|^{\otimes m}$$

where $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad |\pm\rangle = |0\rangle \pm |1\rangle$

Qubit ZX calculus

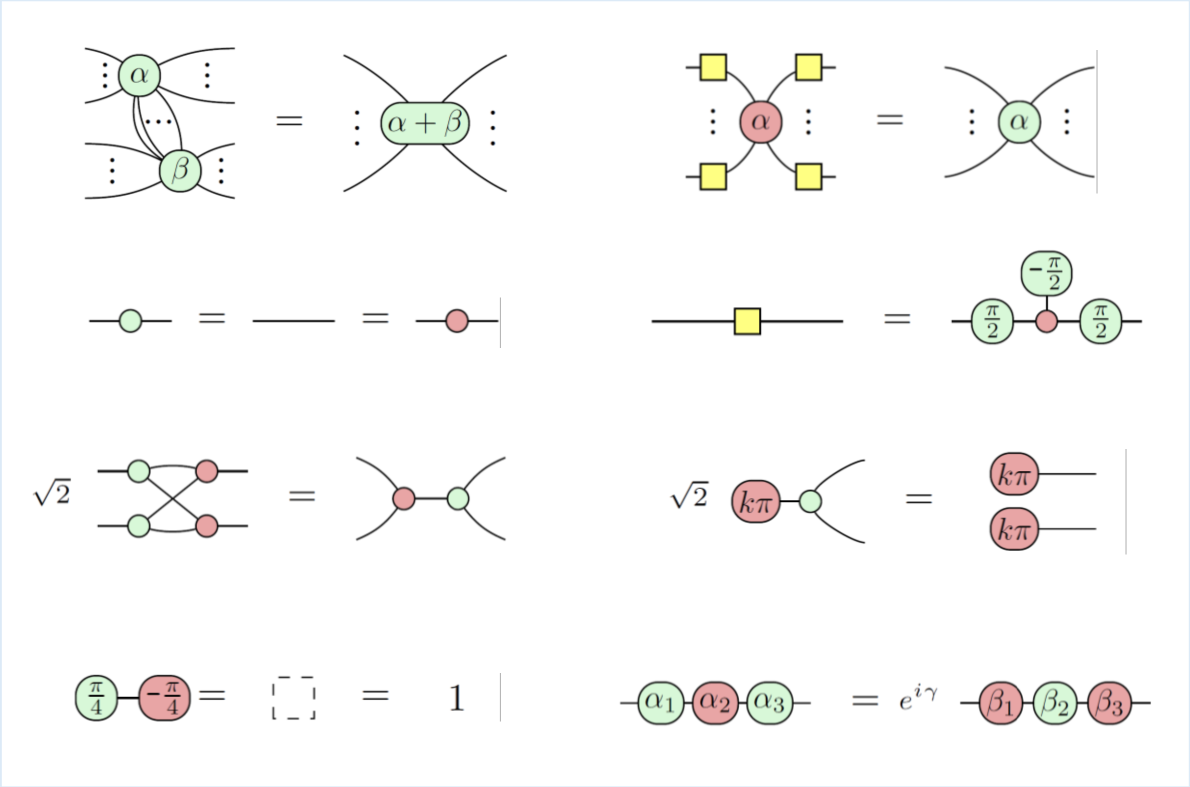


You can draw any quantum circuit using just these two types of spiders.

Universality:

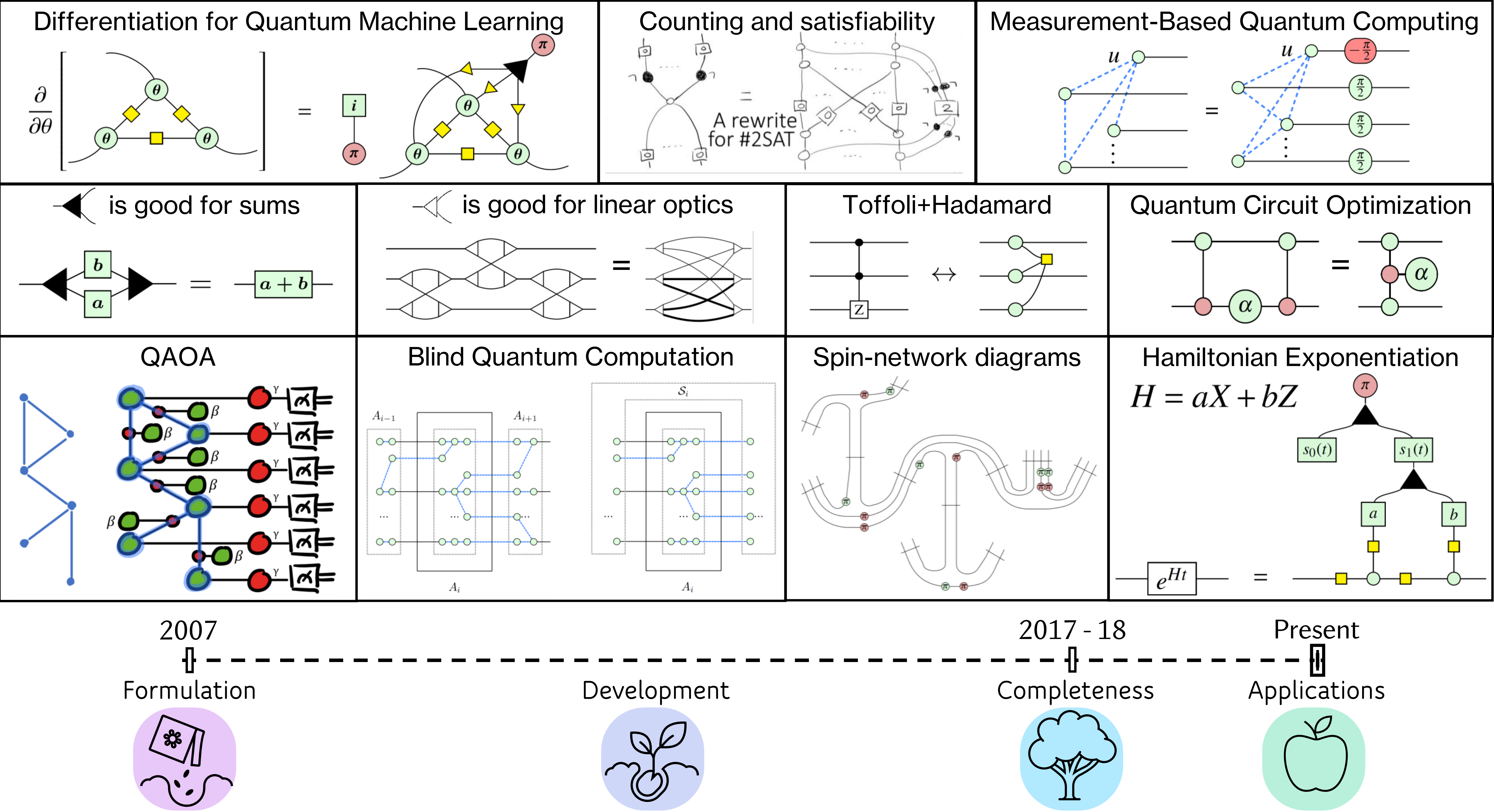
These suffice to represent any linear map on any number of qubits.

Qubit ZX calculus



Completeness:

You can prove any equalities of qubit linear maps using just these 8 rules.



Differentiation for Quantum Machine Learning
Counting and satisfiability
A rewrite for #2SAT
Measurement-Based Quantum Computing
is good for sums
is good for linear optics
Toffoli+Hadamard
Quantum Circuit Optimization
QAOA
Blind Quantum Computation
Spin-network diagrams
Hamiltonian Exponentiation
$H = aX + bZ$
2007
2017 - 18
Present
Formulation
Development
Completeness
Applications

ZX-calculus is Complete for Finite-Dimensional Hilbert Spaces

Boldizsár Poór[1] Razin A. Shaikh[1,2] Quanlong Wang[1]

[1]Quantinuum, 17 Beaumont Street, Oxford, OX1 2NA, United Kingdom
[2]University of Oxford, United Kingdom

The ZX-calculus is a graphical language for reasoning about quantum computing and quantum information theory. As a complete graphical language, it incorporates a set of axioms rich enough to derive any equation of the underlying formalism. While completeness of the ZX-calculus has been established for qubits and the Clifford fragment of prime-dimensional qudits, universal completeness beyond two-level systems has remained unproven until now. In this paper, we present a proof establishing the completeness of finite-dimensional ZX-calculus, incorporating only the mixed-dimensional Z-spider and the qudit X-spider as generators. Our approach builds on the completeness of another graphical language, the finite-dimensional ZW-calculus, with direct translations between these two calculi. By proving its completeness, we lay a solid foundation for the ZX-calculus as a versatile tool not only for quantum computation but also for various fields within finite-dimensional quantum theory.

Why continuous-variable quantum computation?

Why continuous-variable quantum computation?

- Native simulation of Bosonic systems / quantum field theories

Why continuous-variable quantum computation?

- Native simulation of Bosonic systems / quantum field theories

- Quantum error correction with Bosonic codes

Why continuous-variable quantum computation?

- Native simulation of Bosonic systems / quantum field theories

- Quantum error correction with Bosonic codes

- Scalable hardware implementation
  - On photonics, superconducting and trapped-ions platforms

CVQC – what is a qumode?

	Qubit	Qudit	Qumode
Hilbert space	$\mathbb{C}^2$		
State	$\alpha \|0\rangle + \beta \|1\rangle$		

CVQC – what is a qumode?

	Qubit	Qudit	Qumode
Hilbert space	$\mathbb{C}^2$	$\mathbb{C}^d$	
State	$\alpha \|0\rangle + \beta \|1\rangle$	$\sum_{n=0}^{d-1} a_n \|n\rangle$	

CVQC – what is a qumode?

	Qubit	Qudit	Qumode
Hilbert space	$\mathbb{C}^2$	$\mathbb{C}^d$	$L^2(\mathbb{R})$
State	$\alpha \|0\rangle + \beta \|1\rangle$	$\sum_{n=0}^{d-1} a_n \|n\rangle$	$\int_{\mathbb{R}} \psi(x) \|x\rangle \, dx$ or $\sum_{n=0}^{\infty} a_n \|n\rangle$

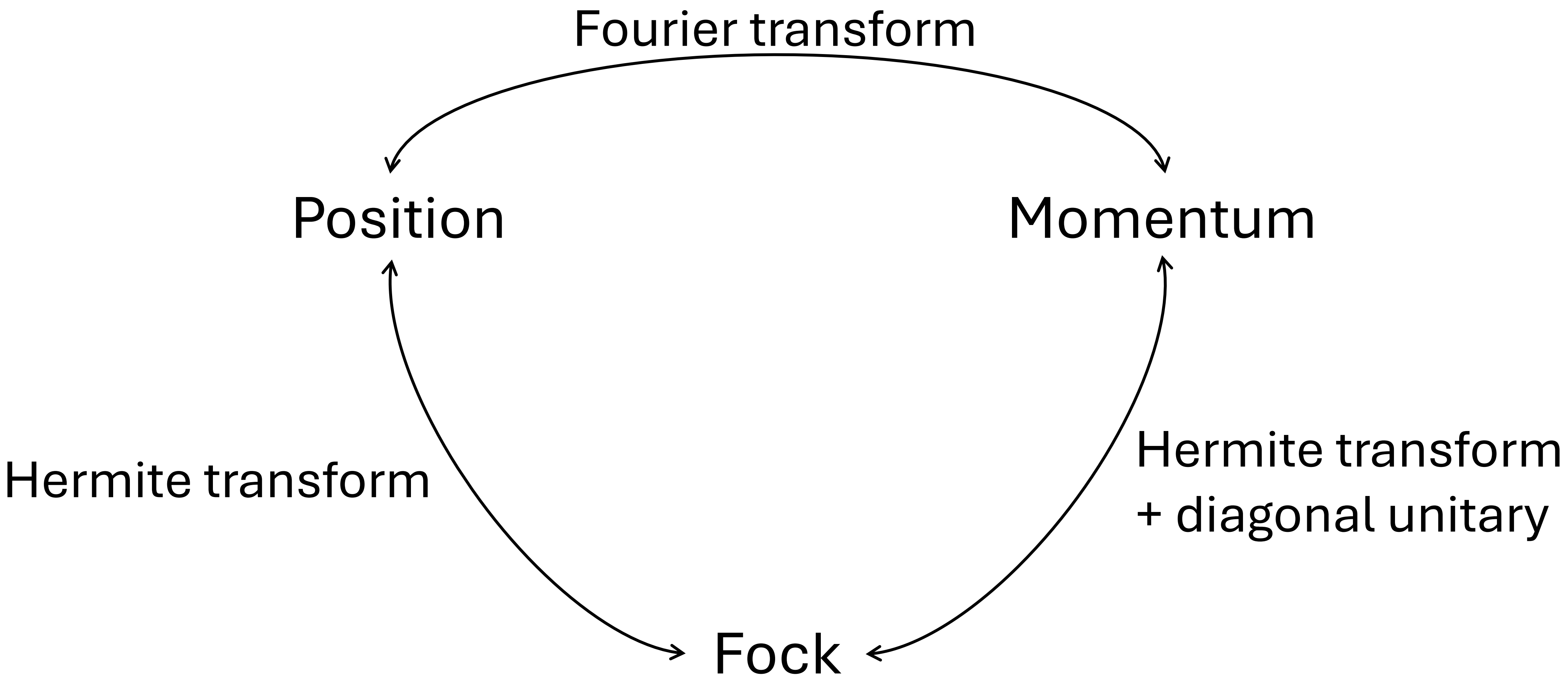
CVQC – Orthogonal bases

Position basis $|x\rangle_{x\in\mathbb{R}}$

Momentum basis $|p\rangle_{p\in\mathbb{R}}$

Fock basis $|n\rangle_{n\in\mathbb{N}}$

CVQC – Orthogonal bases



Example of Hermite transform

$$|n\rangle = \int \psi_n(x)\,|x\rangle\ dx$$

where $\psi_n(x)$ is the n-th Hermite function.

Example of Hermite transform

$$|n\rangle = \int \psi_n(x)\,|x\rangle\; dx$$

where $\psi_n(x)$ is the n-th Hermite function.

$$\begin{aligned}
\psi_0(x) &= \pi^{-\frac{1}{4}}\, e^{-\frac{1}{2}x^2}, \\
\psi_1(x) &= \sqrt{2}\,\pi^{-\frac{1}{4}}\, x\, e^{-\frac{1}{2}x^2}, \\
\psi_2(x) &= \left(\sqrt{2}\,\pi^{\frac{1}{4}}\right)^{-1}\, (2x^2 - 1)\; e^{-\frac{1}{2}x^2}, \\
\psi_3(x) &= \left(\sqrt{3}\,\pi^{\frac{1}{4}}\right)^{-1}\, (2x^3 - 3x)\; e^{-\frac{1}{2}x^2}, \\
\psi_4(x) &= \left(2\sqrt{6}\,\pi^{\frac{1}{4}}\right)^{-1}\, (4x^4 - 12x^2 + 3)\; e^{-\frac{1}{2}x^2}, \\
\psi_5(x) &= \left(2\sqrt{15}\,\pi^{\frac{1}{4}}\right)^{-1}\, (4x^5 - 20x^3 + 15x)\; e^{-\frac{1}{2}x^2}.
\end{aligned}$$

Plan

1. Background and motivation

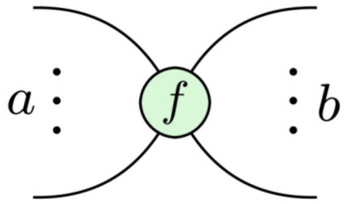
2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness

3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling

4. Conclusion and future work

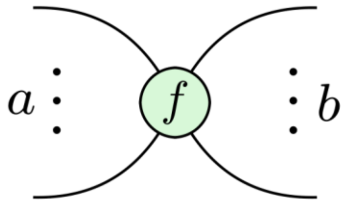
ZX calculus - Generators

ZX calculus - Generators

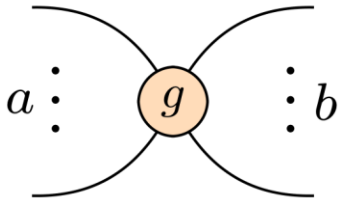


$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \int f(x)\ |x\rangle^{\otimes b}\ \langle x|^{\otimes a}\ dx \qquad \text{(Z-Spider)}$$

ZX calculus - Generators

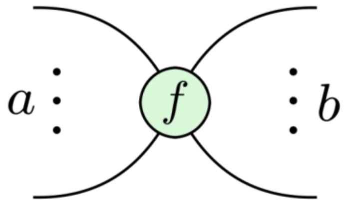


$$\xmapsto{[\![\cdot]\!]} \quad \int f(x)\ |x\rangle^{\otimes b} \langle x|^{\otimes a}\ dx \qquad (\textsc{Z-Spider})$$

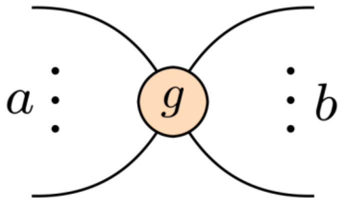


$$\xmapsto{[\![\cdot]\!]} \quad \sum_{n=0}^{\infty} g(n)\, |n\rangle^{\otimes b} \langle n|^{\otimes a} \qquad (\textsc{Fock-Spider})$$

ZX calculus - Generators



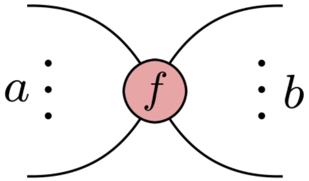
$$\xmapsto{[\![\cdot]\!]} \quad \int f(x)\ |x\rangle^{\otimes b}\ \langle x|^{\otimes a}\ dx \qquad \text{(Z-Spider)}$$



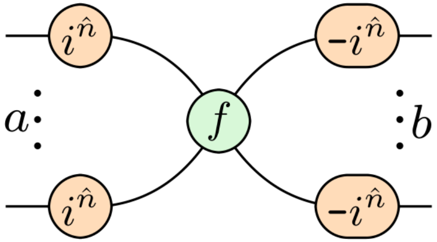
$$\xmapsto{[\![\cdot]\!]} \quad \sum_{n=0}^{\infty} g(n)\,|n\rangle^{\otimes b}\ \langle n|^{\otimes a} \qquad \text{(Fock-Spider)}$$

$$k \quad \xmapsto{[\![\cdot]\!]} \quad k \qquad \text{(Global-Scalar)}$$

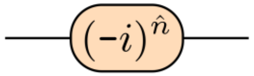
ZX calculus – Notations



$:=$

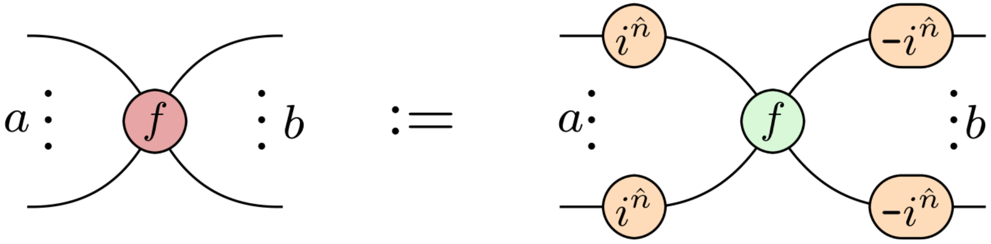


$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \int f(p)\; |p\rangle^{\otimes b} \langle p|^{\otimes a} \; dp \qquad \text{(X-Spider)}$$

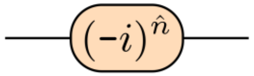
where 

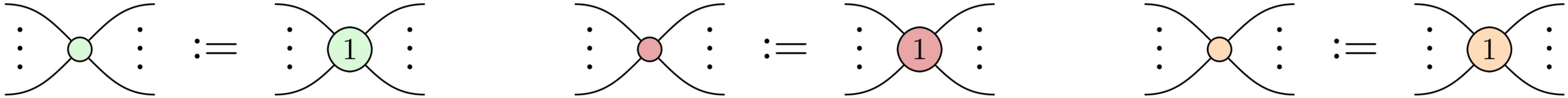
is the Fourier transform.

ZX calculus – Notations

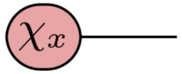


$$\overset{[\![\cdot]\!]}{\longmapsto} \quad \int f(p)\ |p\rangle^{\otimes b}\, \langle p|^{\otimes a}\ dp \qquad \text{(X-SPIDER)}$$

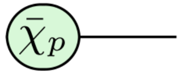
where  is the Fourier transform.



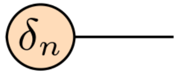
ZX calculus – Notations



$\xmapsto{[\![\cdot]\!]} |x\rangle$



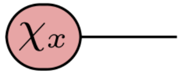
$\xmapsto{[\![\cdot]\!]} |p\rangle$



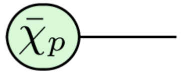
$\xmapsto{[\![\cdot]\!]} |n\rangle$

where $\chi_x(p) = e^{-i2\pi px}$, $\bar{\chi}_p(x) = e^{i2\pi px}$, and δ_n is the Kronecker delta at n.

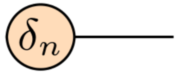
ZX calculus – Notations



$\xrightarrow{[\![\cdot]\!]} \quad |x\rangle$



$\xrightarrow{[\![\cdot]\!]} \quad |p\rangle$

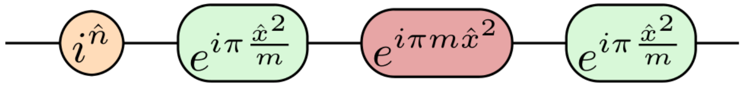


$\xrightarrow{[\![\cdot]\!]} \quad |n\rangle$

where $\chi_x(p) = e^{-i2\pi px}$, $\bar{\chi}_p(x) = e^{i2\pi px}$, and δ_n is the Kronecker delta at n.

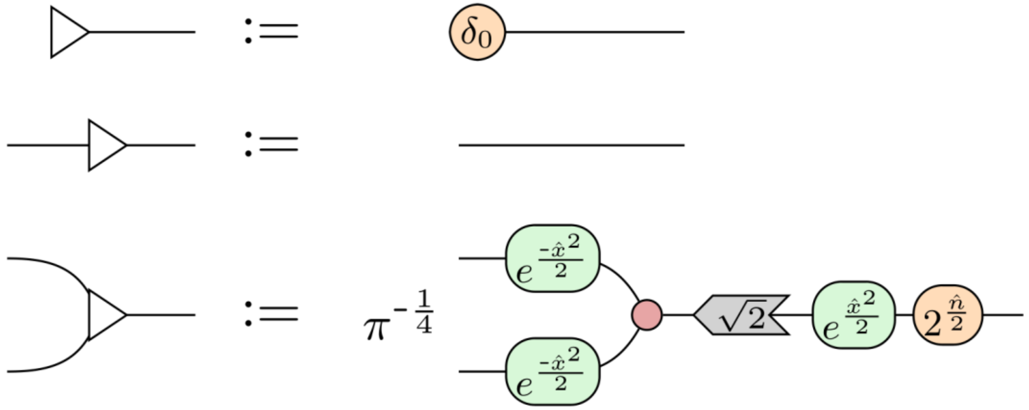
$m \in \mathbb{R}$

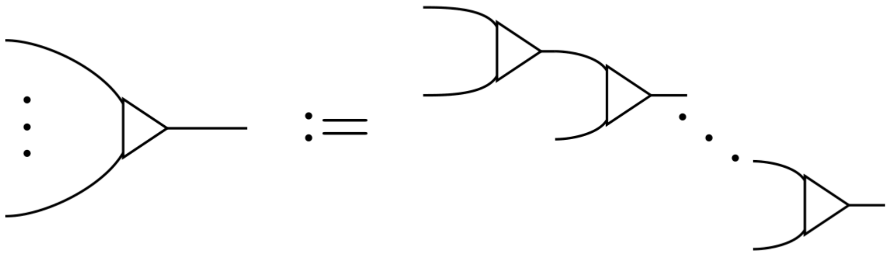
$:=$



$$\xrightarrow{[\![\cdot]\!]} \quad \int |mx\rangle_X \langle x|_X \, dx \qquad (\textsc{Multiplier})$$

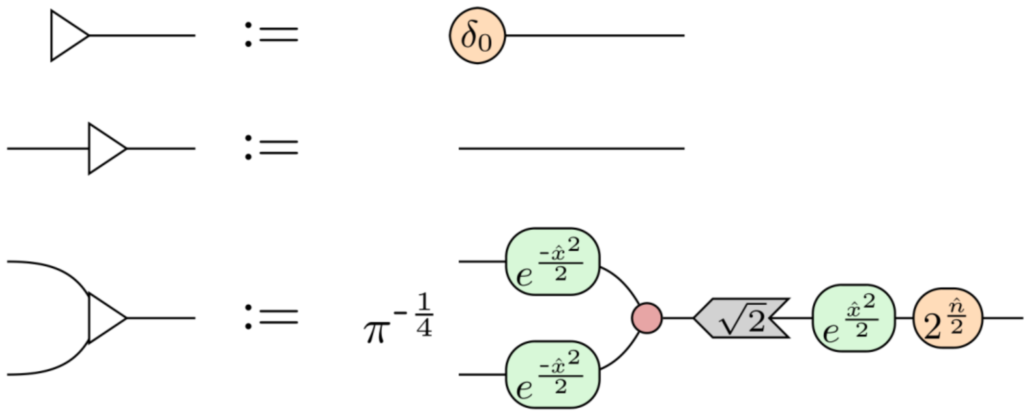
ZX calculus – Notations
W – node

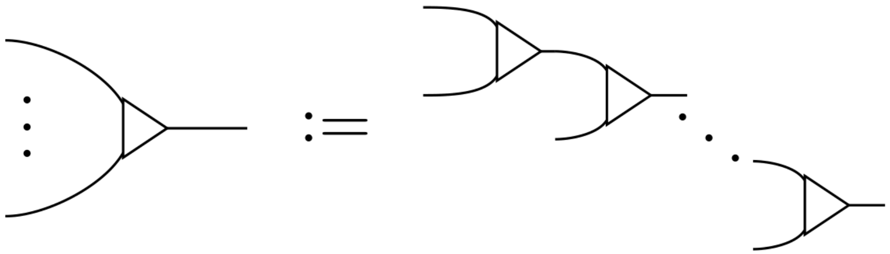


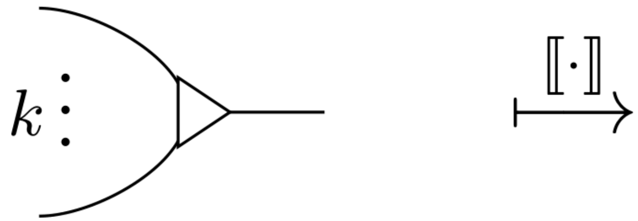


ZX calculus – Notations

W – node

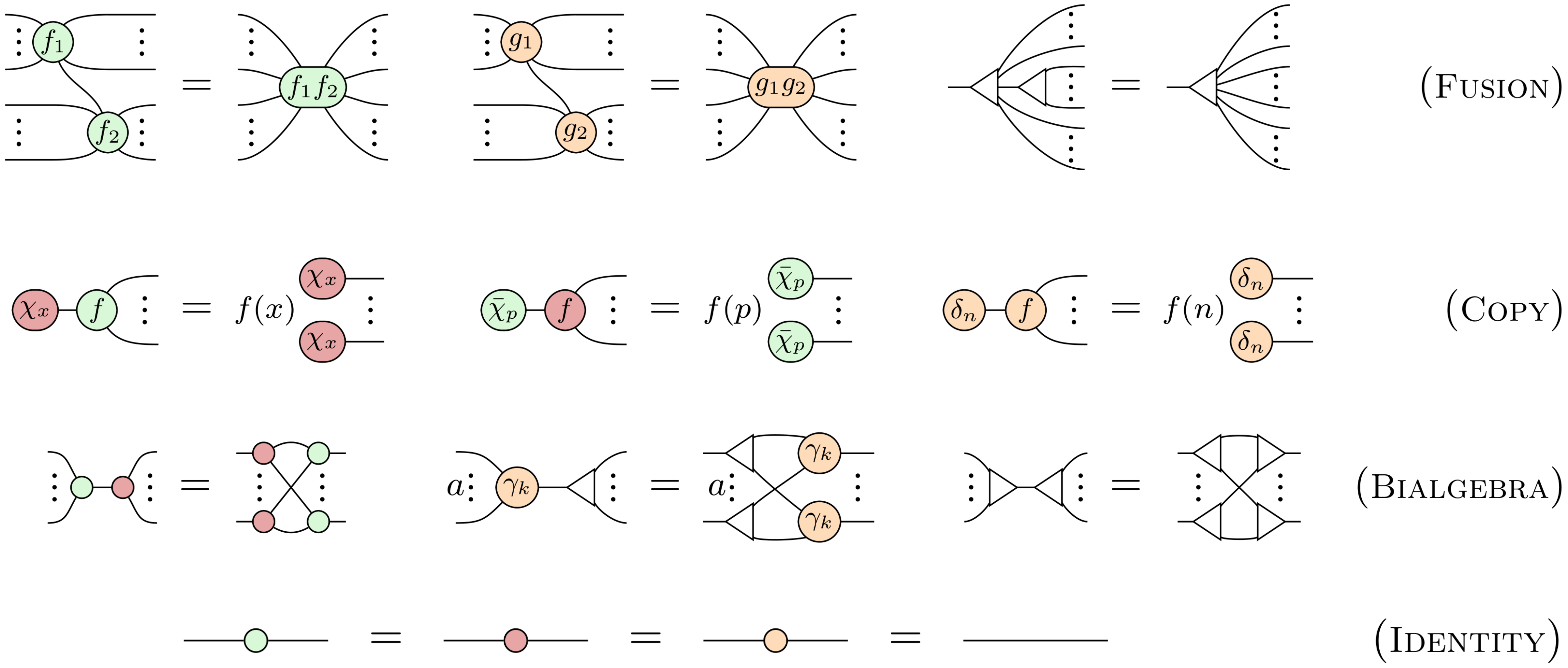




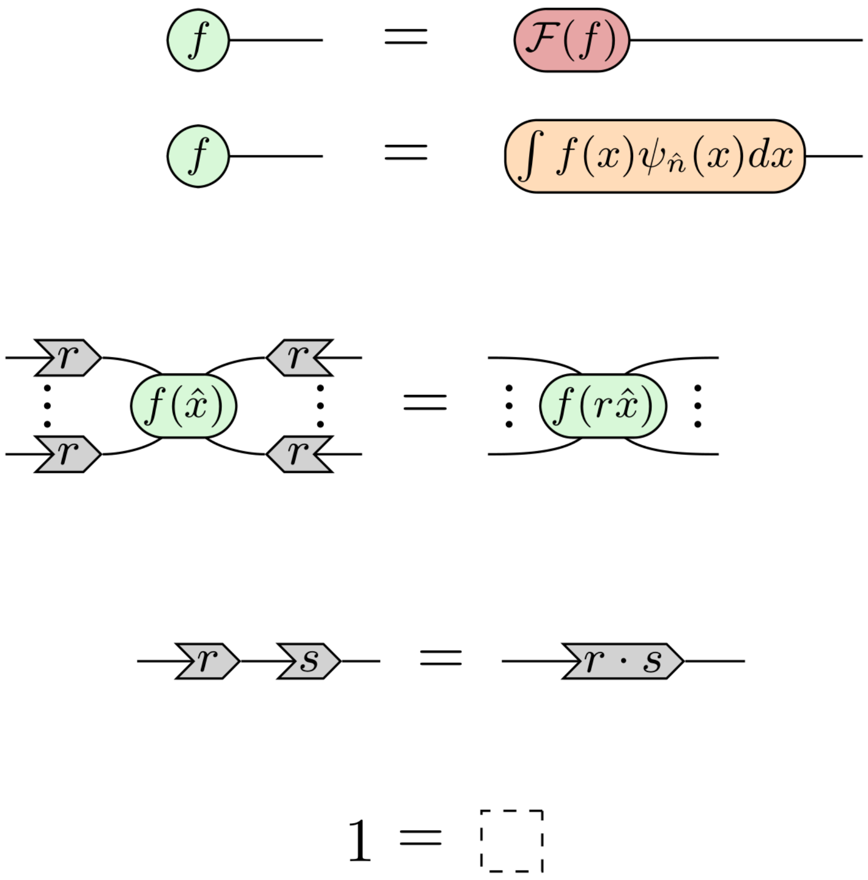


$$\sum_{n_1,\ldots,n_k\geqslant 0}\sqrt{\frac{(\sum_i n_i)!}{\prod_i n_i!}}\,\Big|\sum_i n_i\Big\rangle\Big\langle n_1,\ldots,n_k\Big|$$

ZX calculus – Rules 1



ZX calculus – Rules 2



(FOURIER)

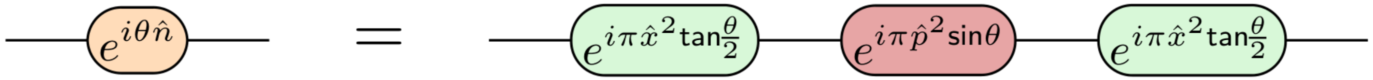
(HERMITE)

(MULT)

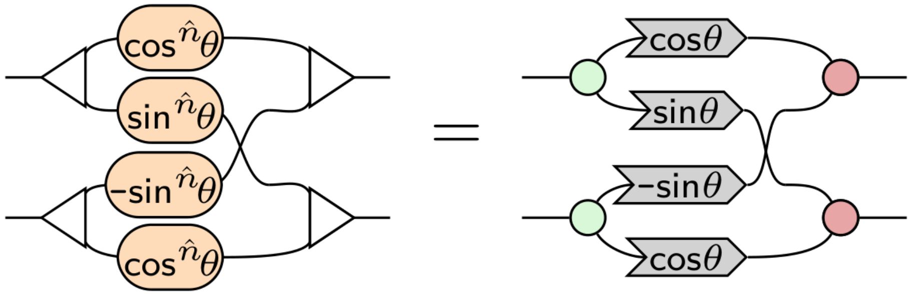
(TIMES)

(ONE)

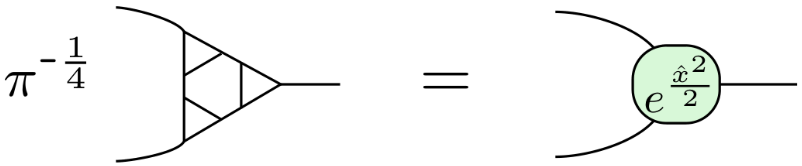
ZX calculus – Rules 3



(EULER)



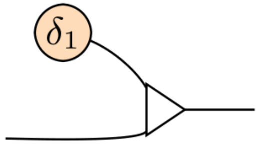
(BEAM-SPLITTER)

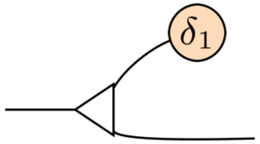


(TRIFORCE)

Operators

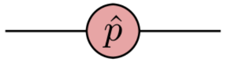
Creation and annihilation operators $a^\dagger|n\rangle = \sqrt{n+1}|n+1\rangle$ and $a|n\rangle = \sqrt{n}|n-1\rangle$



$\xmapsto{[\![\cdot]\!]}$ $a^\dagger$ and 

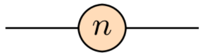
$\xmapsto{[\![\cdot]\!]}$ a

Quadrature operators $\hat{x}\,|x\rangle = x\,|x\rangle$ and $\hat{p}\,|p\rangle = p\,|p\rangle$

$\xmapsto{[\![\cdot]\!]}$ $\hat{x}$ and 

$\xmapsto{[\![\cdot]\!]}$ $\hat{p}$

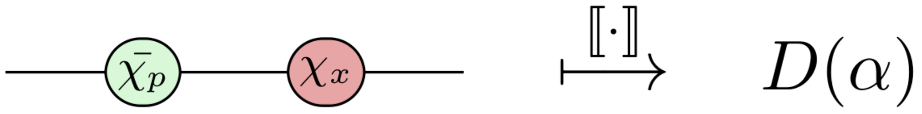
Number operator $\hat{n}\,|n\rangle = \; n\,|n\rangle$



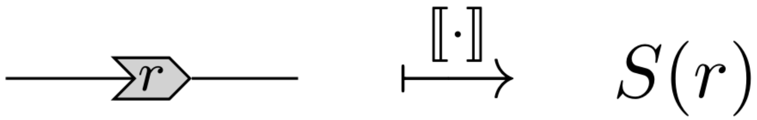
$\xmapsto{[\![\cdot]\!]}$ $\hat{n}$

Gaussian gates – single mode

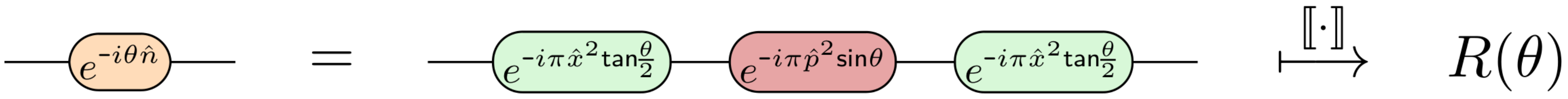
Displacement



Squeezing



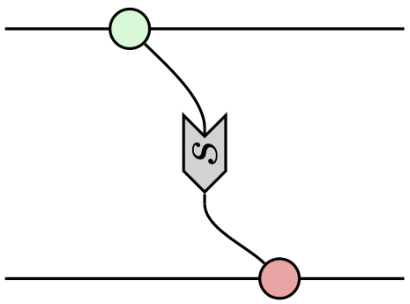
Rotation



Gaussian gates – two modes

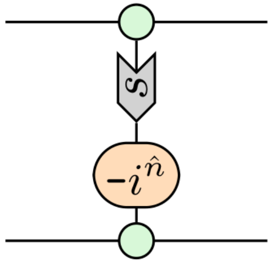
Controlled-X

$$CX(s)\,|x\rangle\,|y\rangle = |x\rangle\,|y+sx\rangle$$



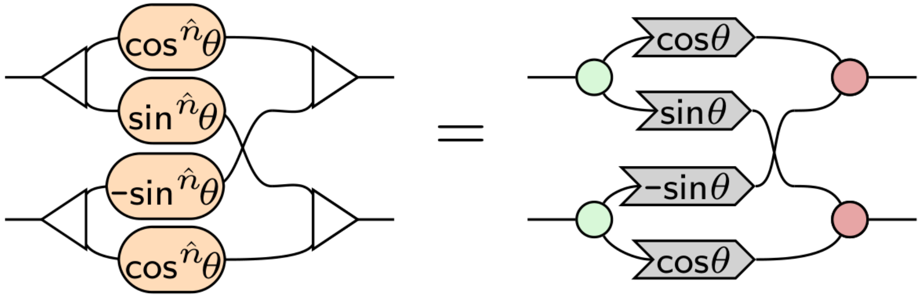
Controlled-phase

$$CZ(s)\,|x\rangle_X\,|y\rangle_X = e^{i2\pi sxy}\,|x\rangle_X\,|y\rangle_X$$



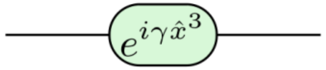
Beam splitter

$$B(\theta,\phi) \;=\; \exp\Big(\theta(e^{i\phi}a_1a_2^\dagger - e^{-i\phi}a_1^\dagger a_2)\Big)$$

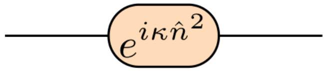


Non-Gaussian gates

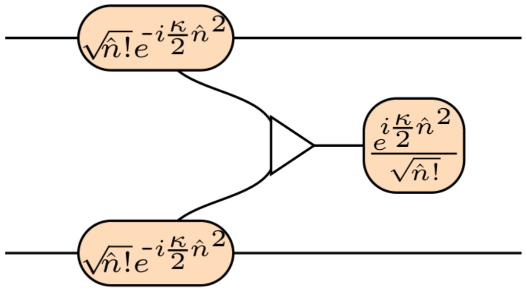
Cubic phase $\quad V(\gamma) = e^{i\gamma\hat{x}^3} =$



Kerr $\quad K(\kappa) = e^{i\kappa\hat{n}^2} =$



Cross-Kerr $\quad CK(\kappa) = e^{i\kappa\hat{n}_1\hat{n}_2} =$



Plan

1. Background and motivation

2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness

3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling

4. Conclusion and future work

Completeness for the Gaussian fragment

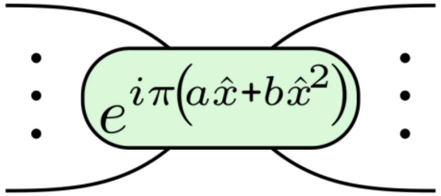
Completeness for the Gaussian fragment

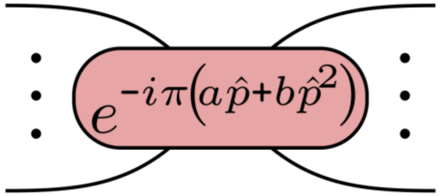
Universal CV gate set: Gaussian gates + 1 non-Gaussian gate

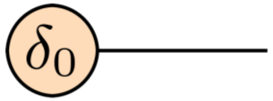
Completeness for the Gaussian fragment

Universal CV gate set: Gaussian gates + 1 non-Gaussian gate

ZX Gaussian fragment:



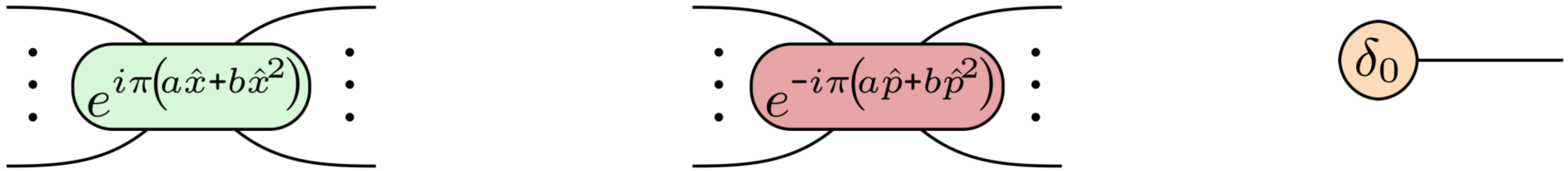




Completeness for the Gaussian fragment

Universal CV gate set: Gaussian gates + 1 non-Gaussian gate

ZX Gaussian fragment:



Theorem 4.6. $\mathrm{ZX_G}$ *is complete for the Gaussian fragment of CVQC: For any two diagrams* D_1 *and* D_2 *in* $\mathrm{ZX_G}$*, if* $[\![D_1]\!] = [\![D_2]\!]$*, then* $\mathrm{ZX_G} \vdash D_1 = D_2$.

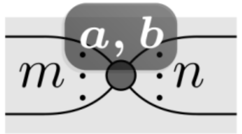
Completeness via translation

Graphical symplectic algebra – (Booth, Carette, Comfort 2024)

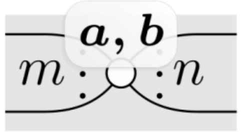
Completeness via translation

Graphical symplectic algebra – (Booth, Carette, Comfort 2024)

GSA generators:



$$\overset{[\![\cdot]\!]}{\longmapsto} \left\{ \left(\begin{bmatrix} \underline{z} \\ x \\ \vdots_{\,m} \\ x \end{bmatrix}, \begin{bmatrix} \underline{z'} \\ x \\ \vdots_{\,n} \\ x \end{bmatrix} \right) \middle| \begin{array}{l} \underline{z} \in \mathbb{R}^m, \underline{z'}; \in \mathbb{R}^n, x \in \mathbb{R} \quad \text{such that} \\ \sum_{j=0}^{m-1} z_j - \sum_{k=0}^{n-1} z'_k + bx = a \end{array} \right\}$$



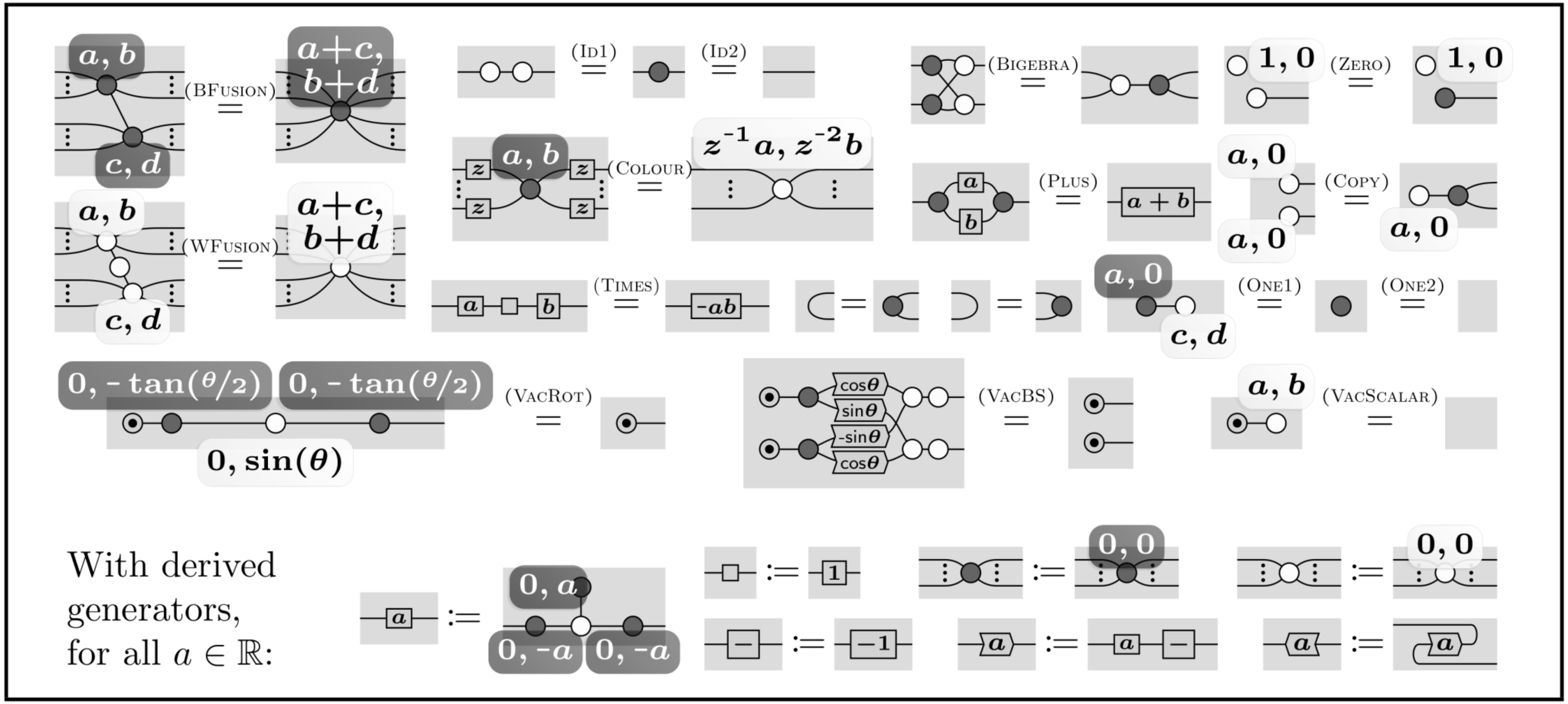
$$\overset{[\![\cdot]\!]}{\longmapsto} \left\{ \left(\begin{bmatrix} z \\ \vdots_{\,m} \\ z \\ \underline{x} \end{bmatrix}, \begin{bmatrix} -z \\ \vdots_{\,n} \\ -z \\ \underline{x'} \end{bmatrix} \right) \middle| \begin{array}{l} \underline{x} \in \mathbb{R}^m, \underline{x'} \in \mathbb{R}^n, z \in \mathbb{R} \quad \text{such that} \\ \sum_{j=0}^{m-1} x_j + \sum_{k=0}^{n-1} x'_k - bz = a \end{array} \right\}$$

$$\overset{[\![\cdot]\!]}{\longmapsto} \left\{ \left(\bullet, \begin{bmatrix} ix \\ x \end{bmatrix} \right) \middle| \, x \in \mathbb{R} \right\}$$

Completeness via translation

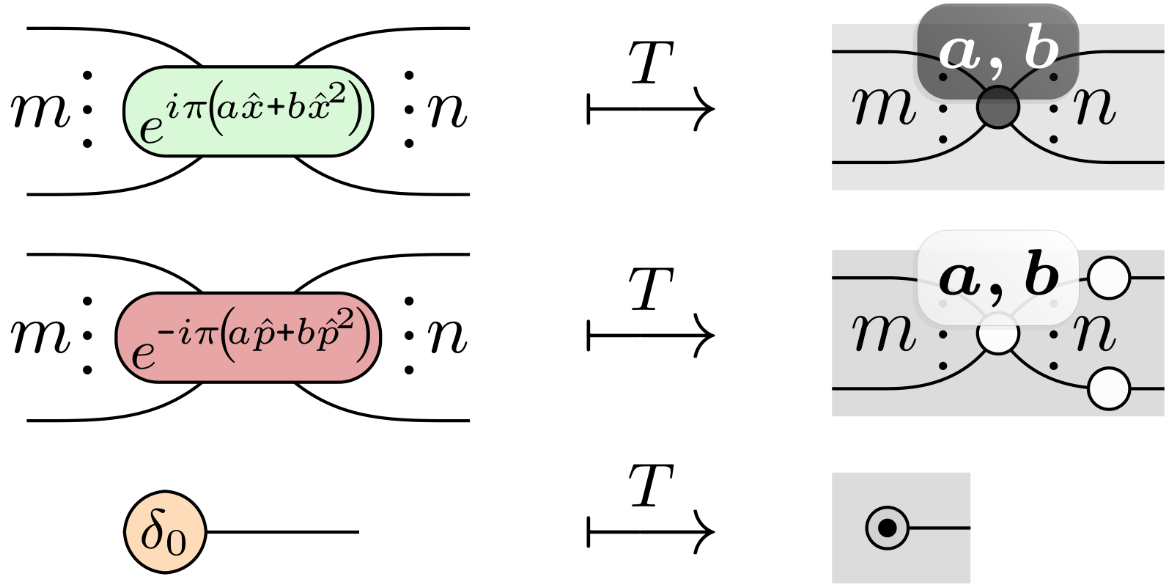
Graphical symplectic algebra – (Booth, Carette, Comfort 2024)

GSA rules:



Completeness via translation

Invertible translation functor $T : \mathrm{ZX_G} \to \mathrm{GSA}$



Completeness via translation

All GSA rules are derivable

Proposition 4.5. *For diagrams D_1 and D_2 in* GSA*, if* GSA $\vdash D_1 = D_2$*, then* $\mathrm{ZX_G} \vdash T^{-1}(D_1) = T^{-1}(D_2)$*.*

Proof. By the functoriality of T^{-1}, it is sufficient to show that all the axioms of GSA (Figure 3) are derivable in $\mathrm{ZX_G}$. The table below summarizes the proofs for each rule.

GSA rule	Follows from
(BFusion)	(Fusion)
(WFusion)	(Fusion)
(Id)	(Identity)
(Bialgebra)	(Bialgebra)
(Zero)	Lemma C.1 & 2.2
(Colour)	(X-Spider) & (Mult)
(Plus)	(Plus)
(Copy)	(Copy)
(Times)	Lemma C.2
(One)	Lemma C.3 & C.4
(VacRot)	Lemma C.5
(VacBS)	Lemma C.6
(VacScalar)	Lemma C.7

□

Completeness via translation

Theorem 4.6. $\mathrm{ZX_G}$ *is complete for the Gaussian fragment of CVQC: For any two diagrams* D_1 *and* D_2 *in* $\mathrm{ZX_G}$*, if* $[\![D_1]\!] = [\![D_2]\!]$*, then* $\mathrm{ZX_G} \vdash D_1 = D_2$.

Plan

1. Background and motivation

2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness

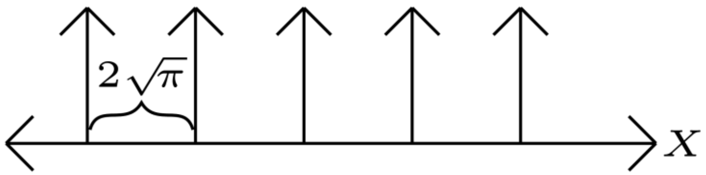
3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling

4. Conclusion and future work

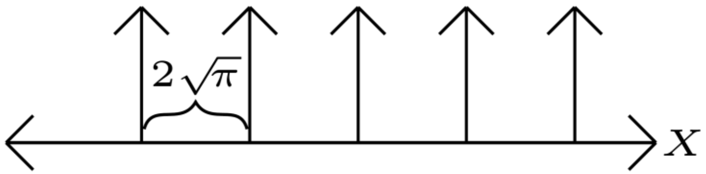
Quantum Error Correction with the GKP code

Encoding a qubit in an oscillator:

$$|0_L\rangle := \sum_{k\in\mathbb{Z}} |2k\sqrt{\pi}\rangle_X$$



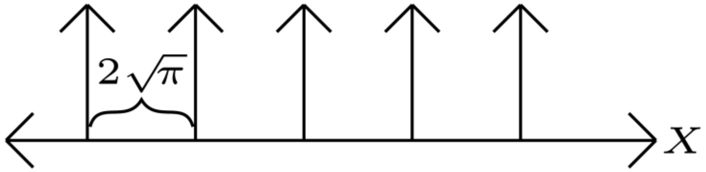
$$|1_L\rangle := \sum_{k\in\mathbb{Z}} |(2k+1)\sqrt{\pi}\rangle_X$$



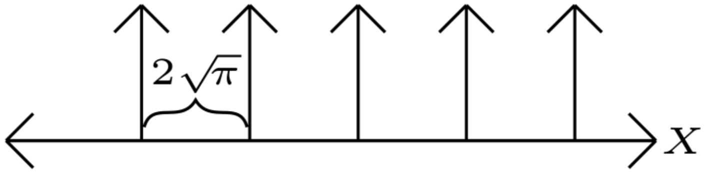
Quantum Error Correction with the GKP code

Encoding a qubit in an oscillator:

$$|0_L\rangle := \sum_{k\in\mathbb{Z}} |2k\sqrt{\pi}\rangle_X$$

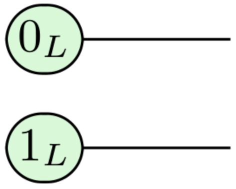


$$|1_L\rangle := \sum_{k\in\mathbb{Z}} |(2k+1)\sqrt{\pi}\rangle_X$$



$|0_L\rangle$ =

$|1_L\rangle$ =



where $0_L(x) = \sum_{k\in\mathbb{Z}} \delta(x - 2k\sqrt{\pi})$ and $1_L(x) = \sum_{k\in\mathbb{Z}} \delta(x - (2k+1)\sqrt{\pi})$.

Quantum Error Correction with the GKP code

Encoding a qubit in an oscillator – finite squeezing

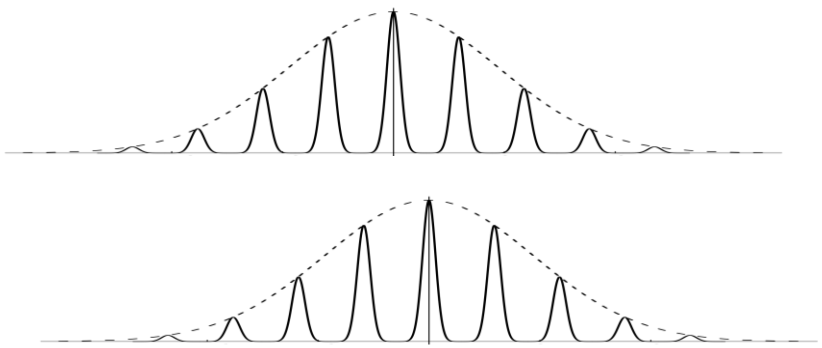
Quantum Error Correction with the GKP code

Encoding a qubit in an oscillator – finite squeezing

$$|\tilde{0}_L\rangle = \int \sum_{k\in\mathbb{Z}} e^{-2\pi\Delta^2 k^2} e^{-\frac{x^2}{2\Delta^2}} \, |x + 2k\sqrt{\pi}\rangle \, dx$$

$$|\tilde{1}_L\rangle = \int \sum_{k\in\mathbb{Z}} e^{-2\pi\Delta^2 k^2} e^{-\frac{x^2}{2\Delta^2}} \, |x + (2k+1)\sqrt{\pi}\rangle \, dx$$

where Δ is the width of the Gaussian

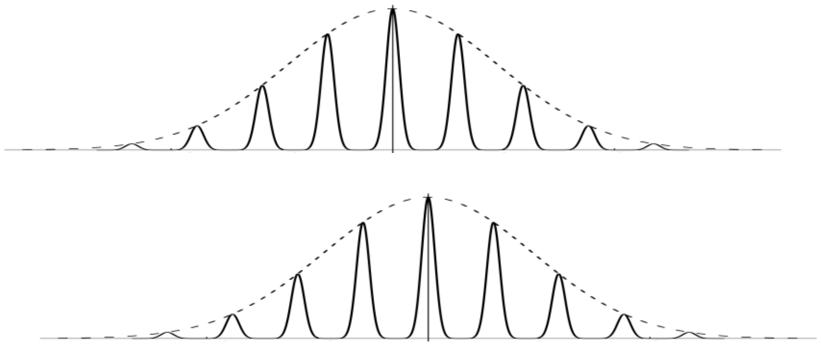


Quantum Error Correction with the GKP code

Encoding a qubit in an oscillator – finite squeezing

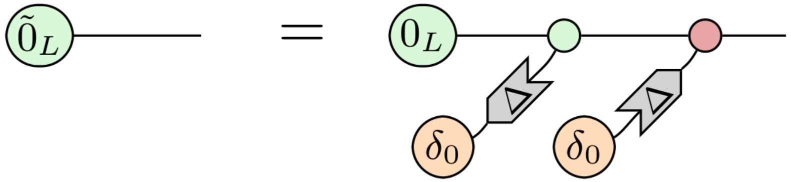
$$|\tilde{0}_L\rangle = \int \sum_{k\in\mathbb{Z}} e^{-2\pi\Delta^2 k^2} e^{-\frac{x^2}{2\Delta^2}} \left|x + 2k\sqrt{\pi}\right\rangle \, dx$$

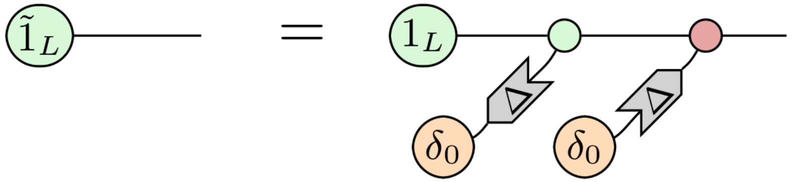
$$|\tilde{1}_L\rangle = \int \sum_{k\in\mathbb{Z}} e^{-2\pi\Delta^2 k^2} e^{-\frac{x^2}{2\Delta^2}} \left|x + (2k+1)\sqrt{\pi}\right\rangle \, dx$$



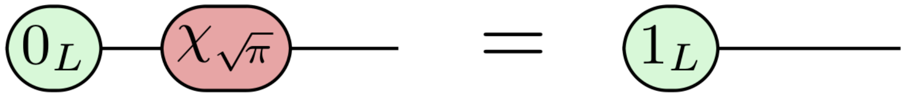
where Δ is the width of the Gaussian

Proposition 5.1.

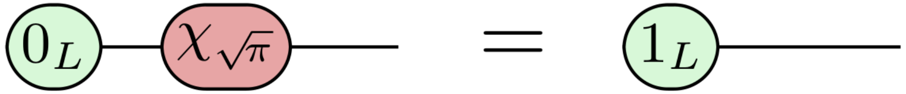


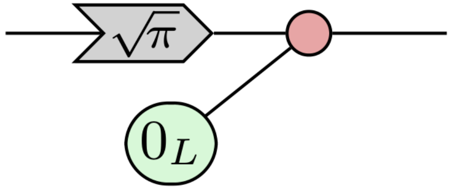


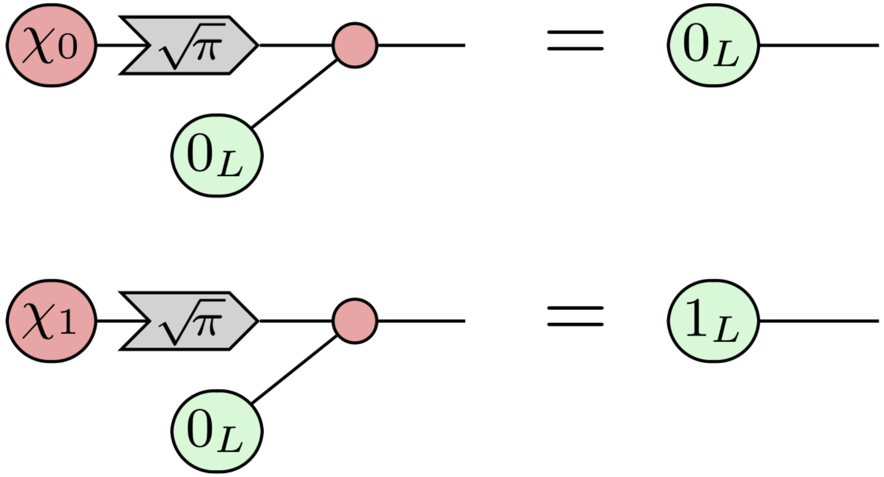
GKP Encoder



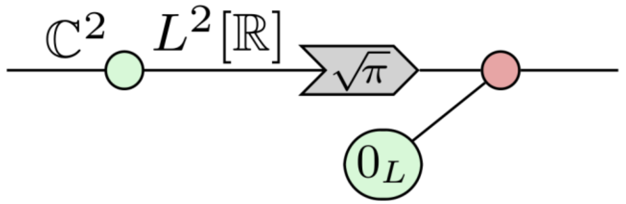
GKP Encoder

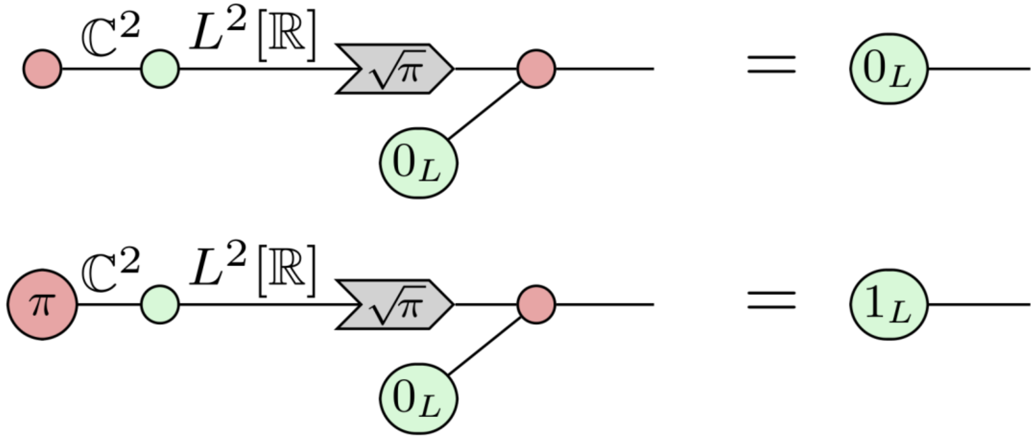






GKP Encoder

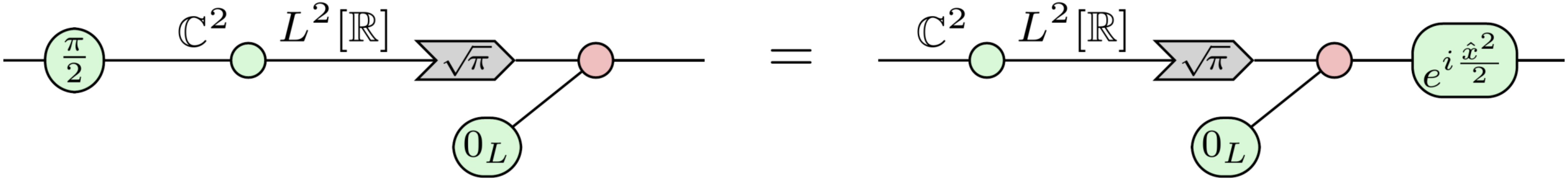




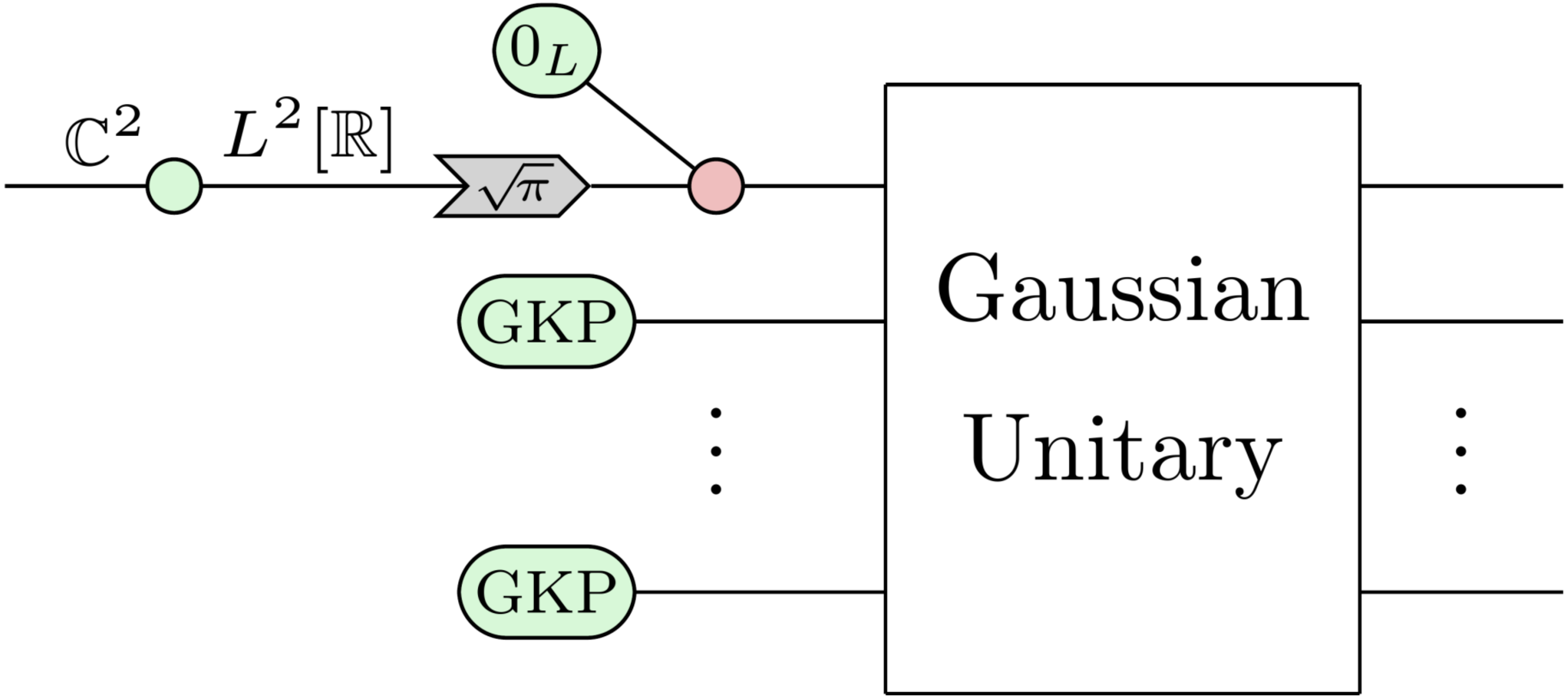
Mixed-dimensional Z spider sends $|0\rangle$ to $|0\rangle_X$ and $|1\rangle$ to $|1\rangle_X$

Logical operators by pushing-through-the-encoder

Example: $\frac{\pi}{2}$ Z rotation (or S gate)

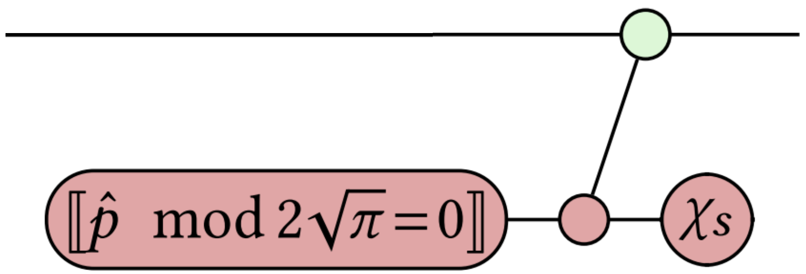


General GKP encoder

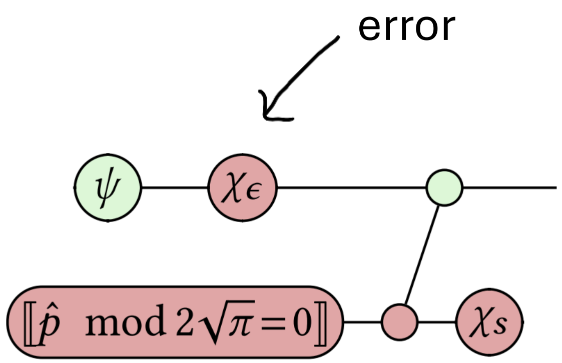


Error detection and correction

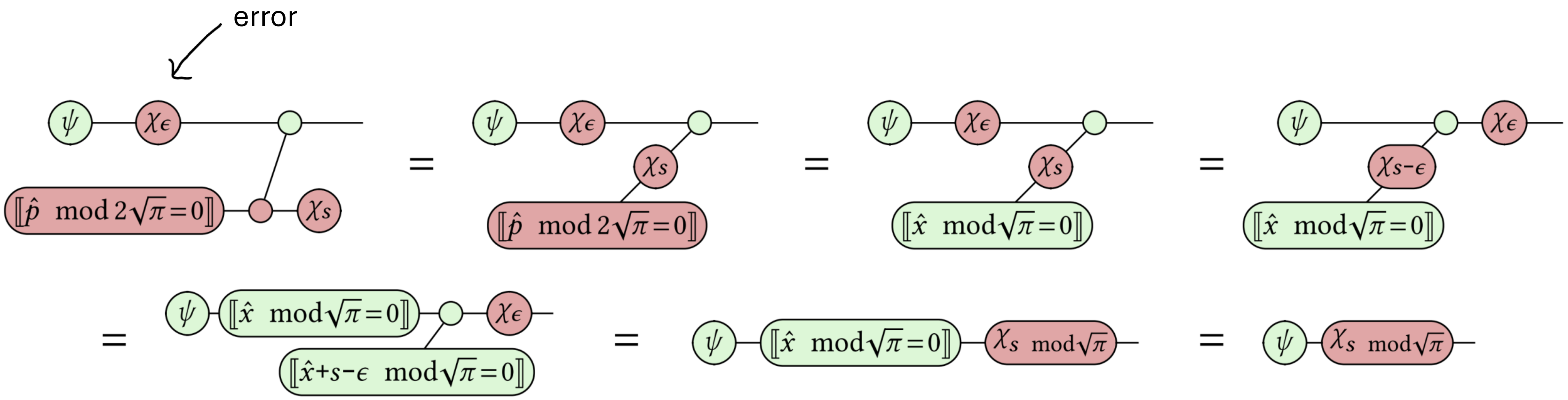
Syndrome measurement circuit:



Error detection and correction



Error detection and correction



Plan

1. Background and motivation

2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness

3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling

4. Conclusion and future work

Gaussian boson sampling

- Non-universal model of quantum computation
- Samples from #P-hard problem
- Experimental demonstrations of quantum advantage

We prove hardness of gaussian boson sampling diagrammatically

Gaussian boson sampling

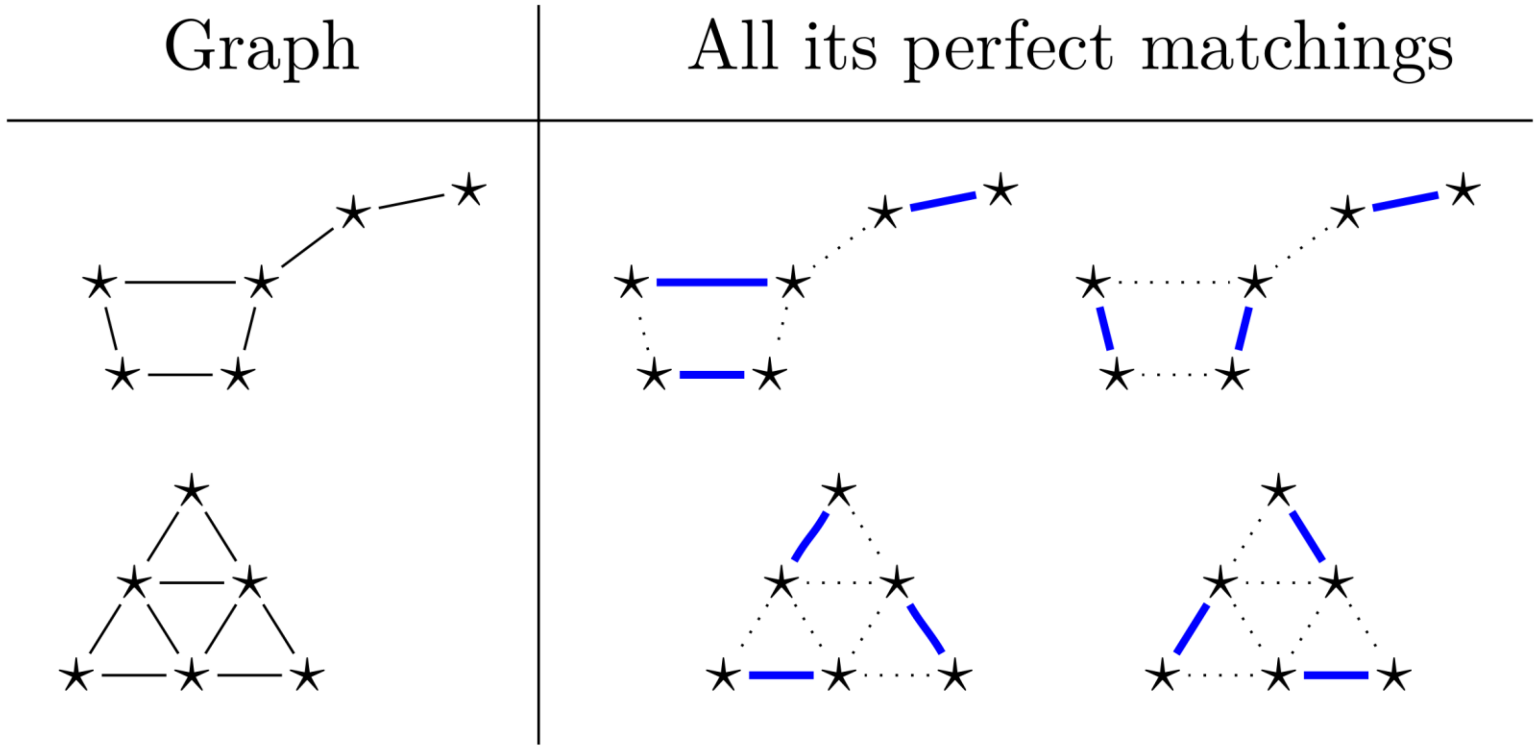
$$\langle n_1, \ldots, n_s | \psi \rangle \;\propto\; \mathrm{Haf}\left(U \bigoplus_{i=1}^{s} \tanh(r_i) U^T \right)_{\mathrm{sub}}$$

where Haf - hafnian function

U - interferometer matrix

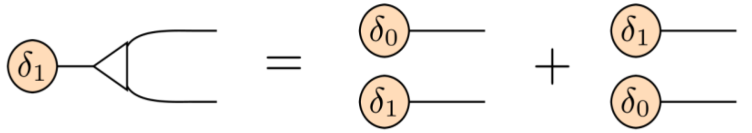
r_i - squeezing parameters

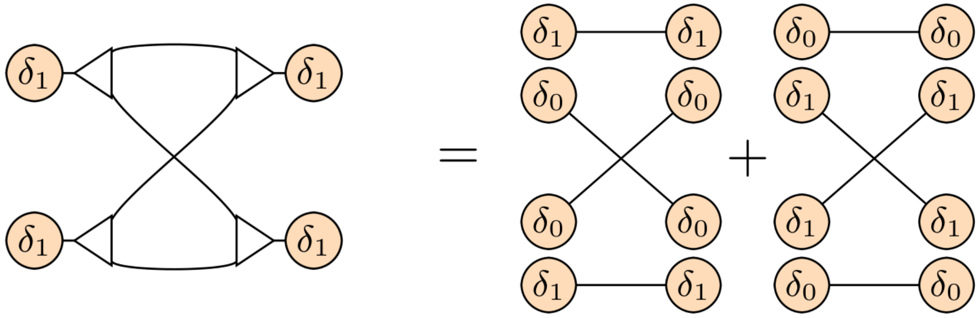
Perfect matchings of graphs



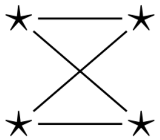
Hafnian is the (weighted) sum of perfect matchings of a graph

Perfect matchings using W-node

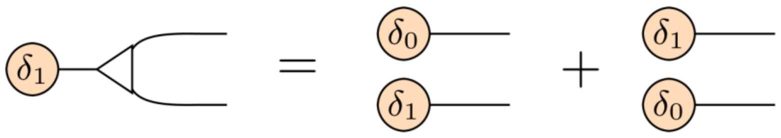


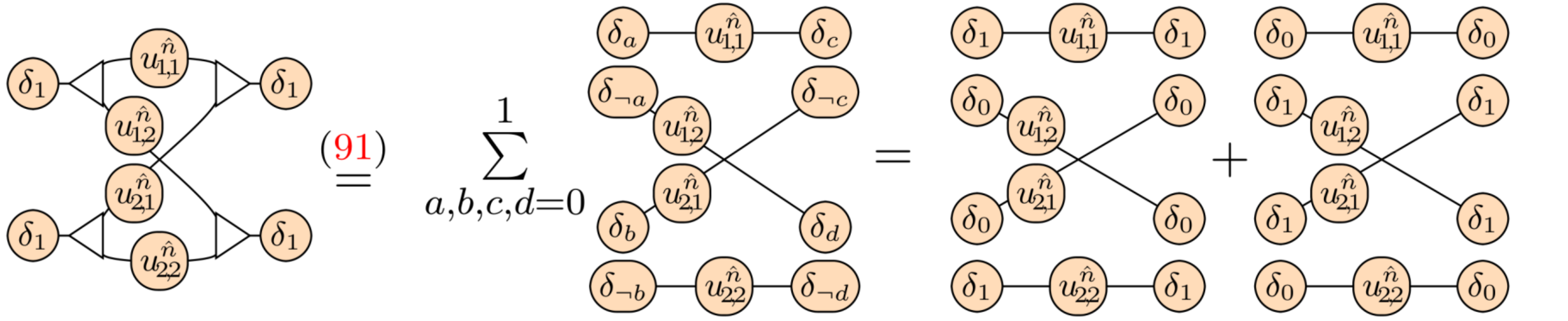


$= \quad 1 + 1 \quad = \quad 2$

Graph	*All its perfect matchings*
	

Perfect matchings using W-node

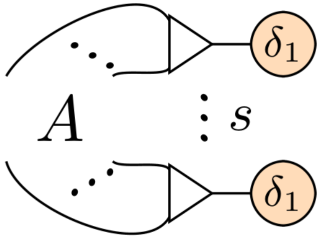




$$= u_{1,1}u_{2,2} + u_{1,2}u_{2,1}$$

Perfect matchings using W-node

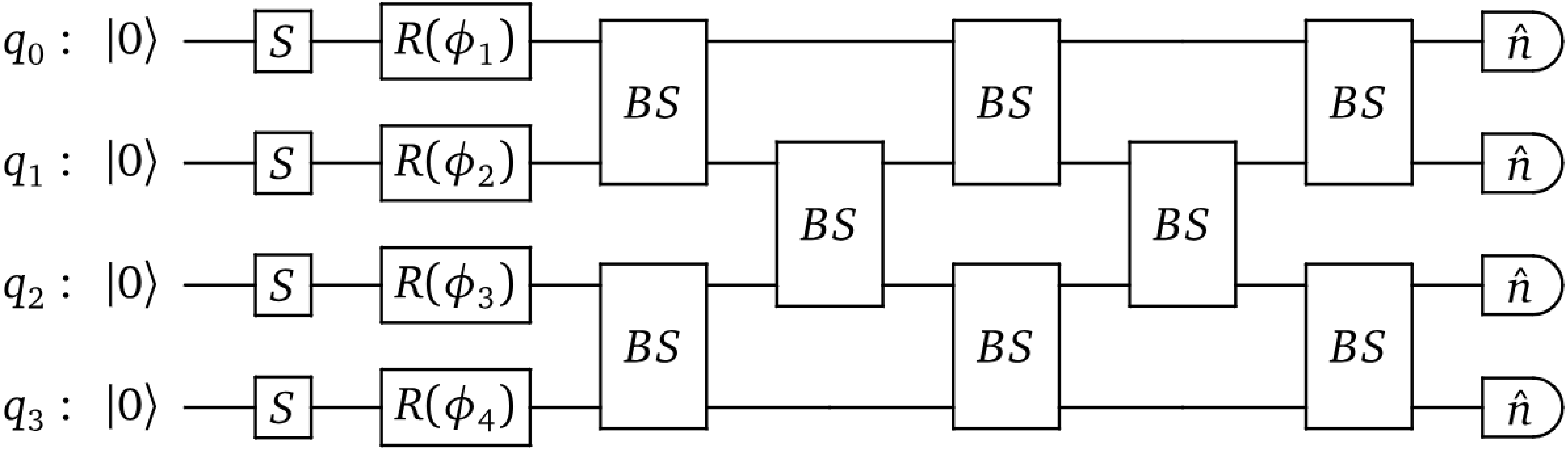
Proposition 6.2. *For a weighted adjacency matrix A of a graph with s vertices,*



$$= \quad \mathit{haf}(A)$$

Proof is by induction

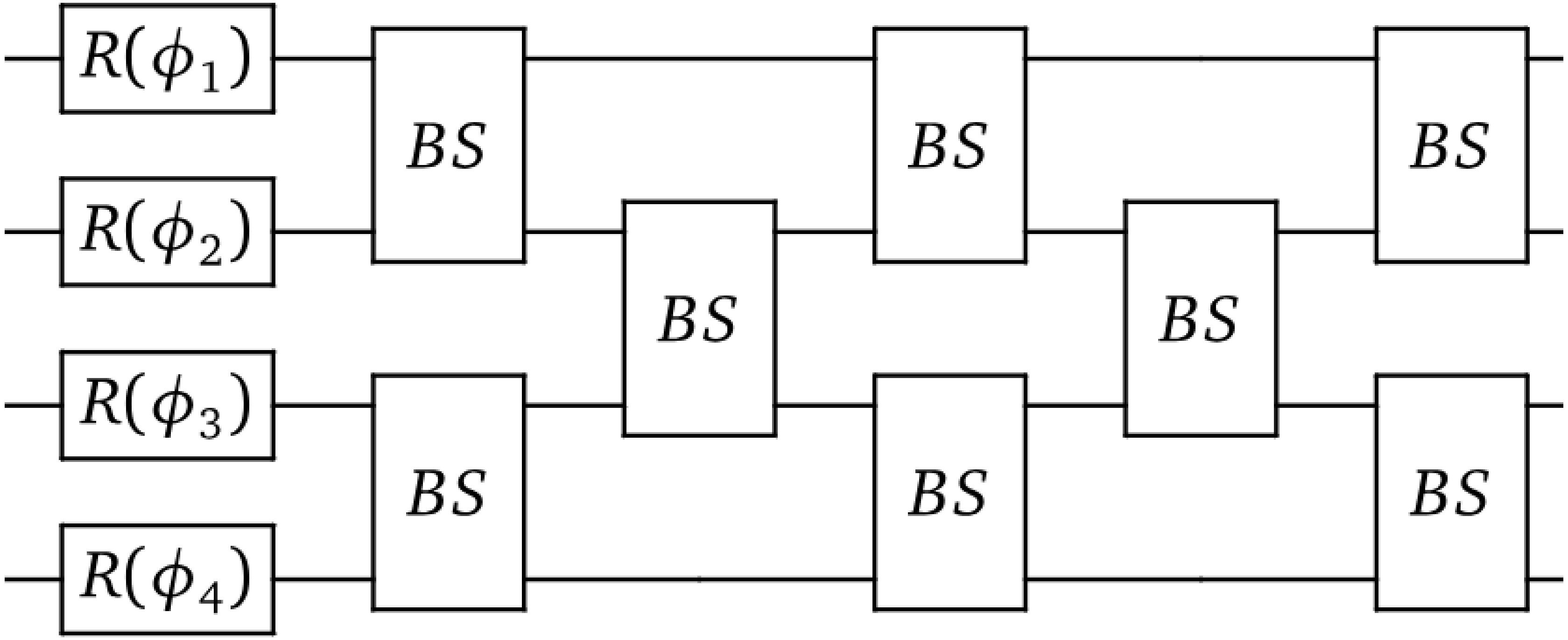
Gaussian boson sampling



An example 4-mode Gaussian boson sampling circuit

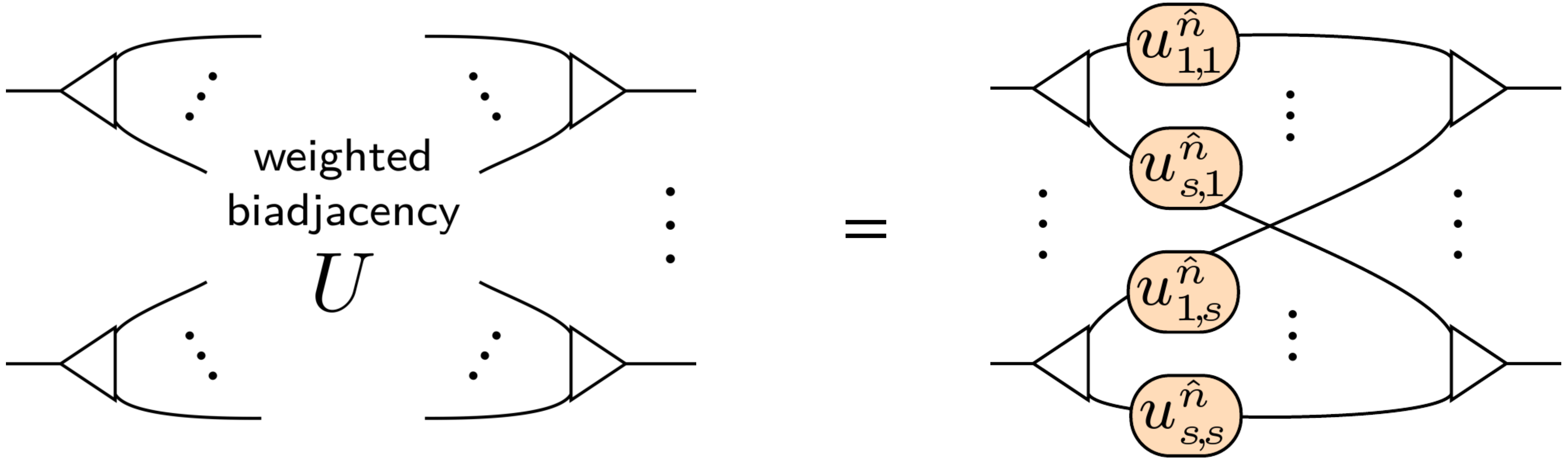
Figure taken from: https://strawberryfields.ai/photonics/demos/run_gaussian_boson_sampling.html

Gaussian boson sampling



An example 4-mode interferometer

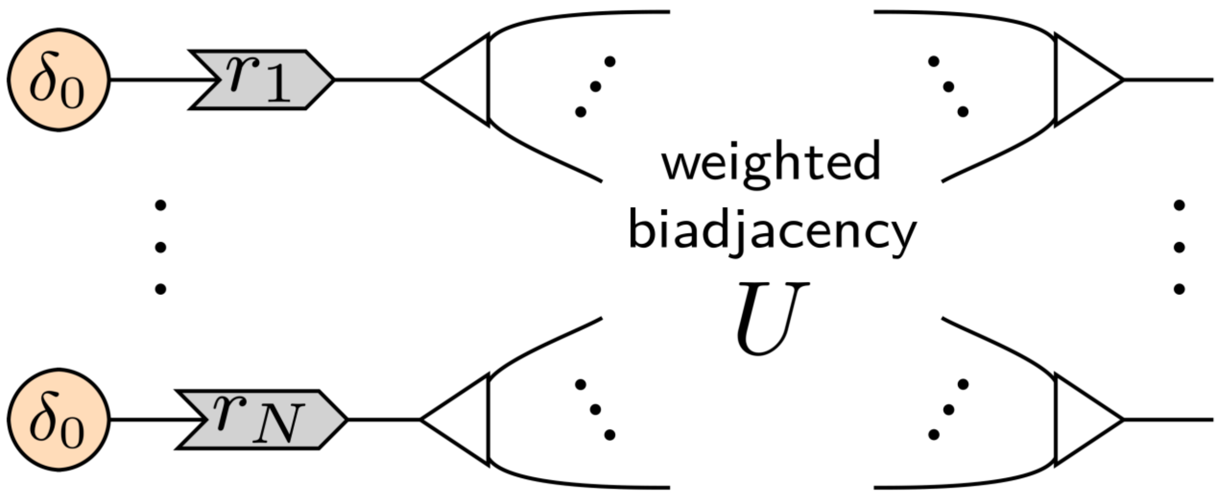
Gaussian boson sampling



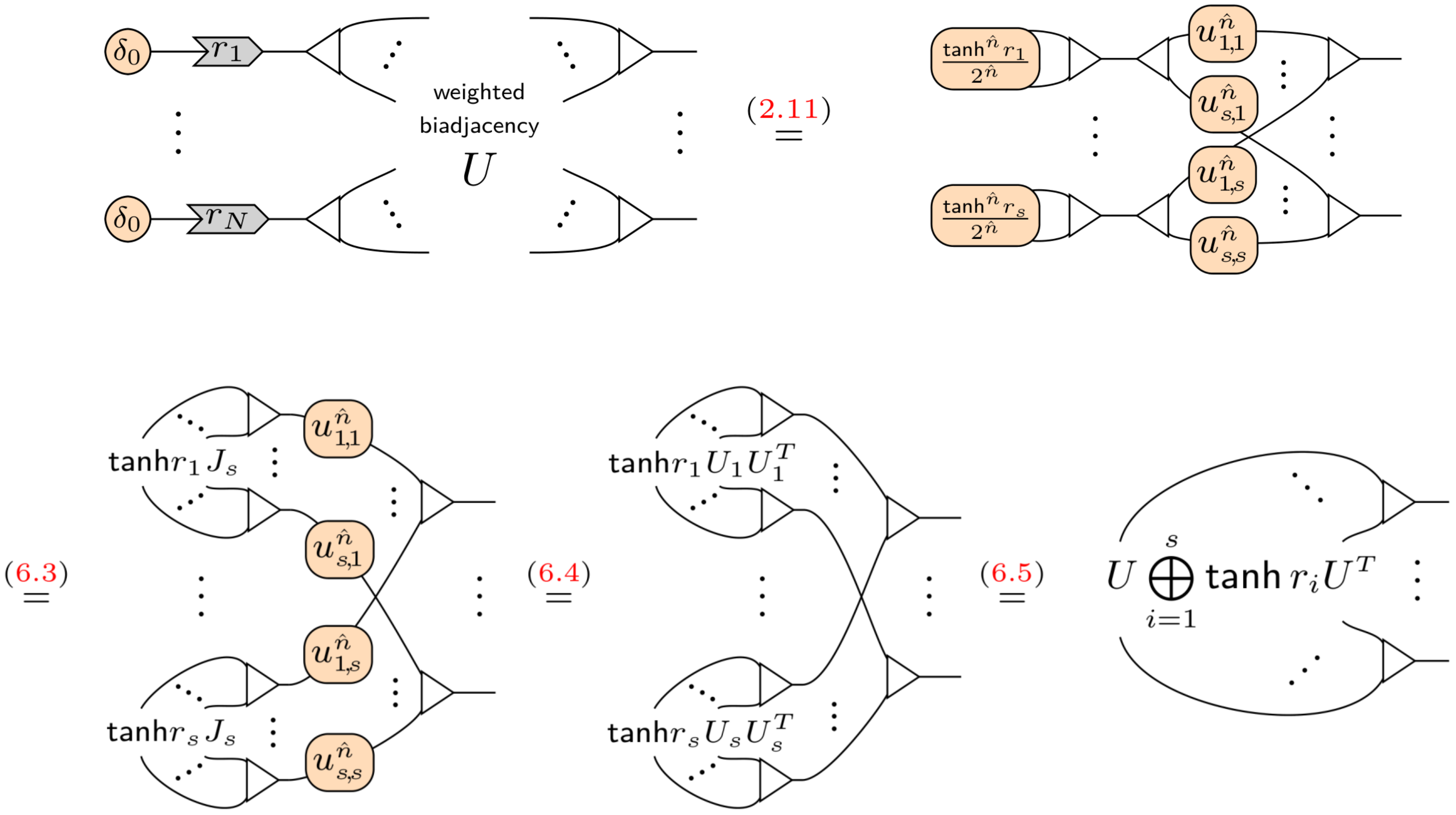
Interferometer normal form

(de Felice et. al 2022), (Bonchi et. al. 2014)

Normal form for Gaussian boson sampling

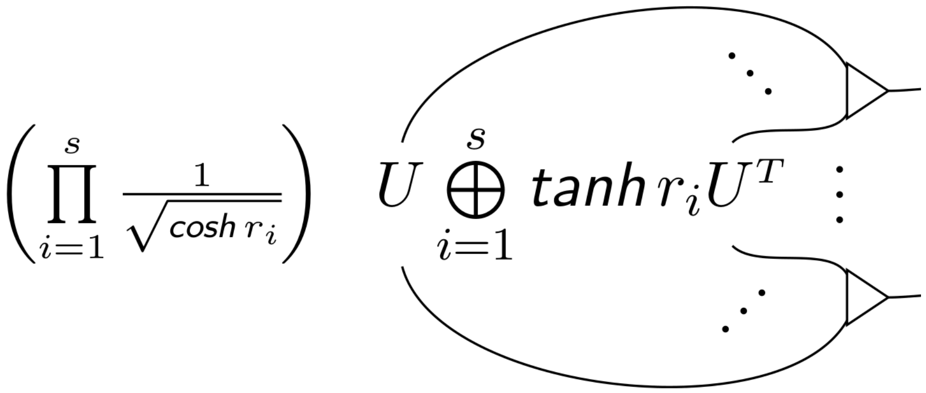


Normal form for Gaussian boson sampling



Normal form for Gaussian boson sampling

Theorem 6.6. *The circuit of Gaussian boson sampling can be reduced to the following normal form:*

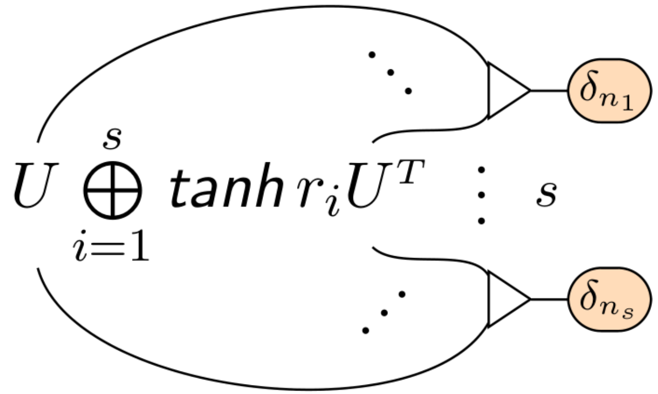


where U is the matrix of the interferometer, r_i represents the amount of squeezing

Sampling amplitude

Theorem 6.12. *The amplitude of observing* $n_1, \ldots, n_s \in \mathbb{N}$ *photons in the Gaussian boson sampling circuit is*

$$\left(\prod_{i=1}^{s} \frac{1}{\sqrt{\cosh r_i}}\right)$$



$$= \prod_{i=1}^{s} \frac{1}{\sqrt{n_i \cosh r_i}} \, haf\left[U \bigoplus_{i=1}^{s} \tanh r_i U^T\right]_{sub}$$

Plan

1. Background and motivation
2. Focked-up ZX calculus
   - Generators, rules and common gates
   - Gaussian completeness
3. Applications
   - Quantum Error Correction with the GKP code
   - Gaussian Boson sampling
4. Conclusion and future work

Conclusion

Conclusion

Generalized ZX calculus to infinite dimensions

- With continuous and discrete generators
- Complete for the Gaussian fragment

Conclusion

Generalized ZX calculus to infinite dimensions

- With continuous and discrete generators
- Complete for the Gaussian fragment

Graphical analysis of

- GKP quantum error correction
- Gaussian Boson sampling

Future work

Future work

ZX-based algorithms for

- Compiling and circuit optimization
- Classical simulation
- MBQC with hybrid-CV cluster states

Future work

ZX-based algorithms for

- Compiling and circuit optimization
- Classical simulation
- MBQC with hybrid-CV cluster states

Completeness

- Gaussian completeness for the Fock-W fragment
- Completeness for the approximately universal fragment

Future work

ZX-based algorithms for

- Compiling and circuit optimization
- Classical simulation
- MBQC with hybrid-CV cluster states

Completeness

- Gaussian completeness for the Fock-W fragment
- Completeness for the approximately universal fragment

Quantum error correction

- Pauli-webs and floquetification for GKP
- Multi-mode GKP code and concatenated GKP code
- Cat codes and binomial codes