A dataflow programming framework for linear optical distributed quantum computing

Giovanni de Felice[1], Boldizsár Poór[1,2], Cole Comfort[3], Lia Yeh[4], Mateusz Kupper[1], William Cashman[2], and Bob Coecke[5,6]

[1]Quantinuum, 17 Beaumont Street, Oxford, OX1 2NA, United Kingdom
[2]University of Oxford, Department of Computer Science, Oxford, OX1 3QD, United Kingdom
[3]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Inria, Laboratoire Méthodes Formelles, 91190, Gif-sur-Yvette, France
[4]University of Cambridge, Department of Computer Science and Technology, Cambridge, CB3 0FD, United Kingdom
[5]Wolfson college, Linton Road, Oxford, OX2 6UD, United Kingdom
[6]Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada

Photonic systems offer a promising platform for interconnecting quantum processors and enabling scalable, networked architectures. Designing and verifying such architectures requires a unified formalism that integrates linear algebraic reasoning with probabilistic and control-flow structures. In this work, we introduce a graphical framework for distributed quantum computing that brings together linear optics, the ZX-calculus, and dataflow programming. Our language supports the formal analysis and optimization of distributed protocols involving both qubits and photonic modes, with explicit interfaces for classical control and feedforward, all expressed within a synchronous dataflow model with discrete-time dynamics. Within this setting, we classify entangling photonic fusion measurements, show how their induced Pauli errors can be corrected via a novel flow structure for fusion networks, and establish correctness proofs for new repeat-until-success protocols enabling arbitrary fusions. Layer by layer, we construct qubit architectures incorporating practical optical components such as beam splitters, switches, and photon sources, with graphical proofs that they are deterministic and support universal quantum computation. Together, these results establish a foundation for verifiable compilation and automated optimization in networked quantum computing.

Contents

Giovanni de Felice: giodefelice@protonmail.com

Cole Comfort: this work has been partially funded by the European Union through the MSCA SE project QCOMICAL and within the framework of "Plan France 2030", under the research projects EPIQ ANR-22-PETQ-0007 and HQI-R&D ANR-22-PNCQ-0002.

Lia Yeh: this work was done while at Quantinuum and University of Oxford.

1 Introduction

Photonic systems are a leading platform for interconnecting modular quantum hardware into distributed networks. Recent proposals for scalable quantum computing exploit interfaces between matter qubits and discrete-variable photonic modes, including distributed hybrid architectures based on ions [53, 74], neutral atoms [26, 100], or spins [24, 42], or drive the entire computation by measurements on photonic qubits, as in fusion-based quantum computing [8]. These models point toward the possibility of quantum networks capable of executing large-scale distributed quantum algorithms.

Current quantum programming languages lack many necessary capabilities in order to natively capture non-qubit systems such as photons. The majority are designed solely for circuit-based models, where computation derives from a fixed set of unitary gates. Distributed quantum computation, however, demands a new paradigm that hybridizes circuit-based and measurement-based approaches while accommodating heterogeneous quantum systems. Software for such networks must therefore represent diverse physical systems and their interactions while supporting the computational abstractions needed for verification, optimization, and compilation. Meeting these requirements calls for a formalism that links physical notions of scalability with computer science concepts such as semantics, abstraction, control flow, concurrency, and rewriting.

The ZX calculus [21] has already proved effective across circuit-based [22, 56], measurement-based [5, 35, 71], and fault-tolerant [28, 49, 55, 92, 93, 103] models of quantum computation. In the context of photonic quantum computing, the ZX calculus is starting to be applied in a top-down direction, e.g. to design novel error correcting codes [10, 86]. However, the bottom-up direction, from the hardware described in the language of linear optics [1] to the logic of ZX calculi, has not been established: current graphical languages for optics [20, 30, 31, 47] are restricted to *passive* optical setups and lack formal notions of measurement and feedforward which are crucial to performing entangling gates in photonic architectures.

A second critical gap is the absence of formalisms for the *dynamic* nature of optical protocols. Current approaches rely on static circuit diagrams, which obscure a central advantage of photonics: the ability to exploit both spatial and temporal degrees of freedom [67, 75]. Optical circuits are inherently time-dependent, acting on streams of photons with components such as routers, delays, switches, and time-controlled emitters. A suitable formalism should therefore have the expressivity to specify what each component does at each time step. Extending recent graphical formalisations of discrete-time dynamics [15, 33] to real-world hardware components would enable recursive reasoning and rewriting across a wide variety of experimental setups.

In this work, we develop the mathematical foundations to enable: (i) verification and comparison of proposals for networked quantum computing, (ii) translation of techniques from circuit-based models into the photonic, measurement-driven setting, and (iii) compilation of distributed algorithms across heterogeneous hardware platforms.

A number of compilation frameworks and algorithms have recently been developed for photonic one-way computation [61, 109, 110] and for distributed architectures more broadly [3, 40]. These approaches are typically algorithmic or architecture-specific, focusing on resource optimization and circuit translation. Our approach is complementary: by grounding compilation in formal languages and rewriting, we provide a unifying framework in which such strategies can be expressed, analyzed, and verified. In doing so, we connect hardware-level models such as linear optics with high-level programming concepts such as control flow and recursion, enabling systematic reasoning and the development of new design principles for networked quantum architectures.

1.1 Graphical framework

Our main contribution is a graphical language `Stream`(`Channel`(**ZXLO**)) for reasoning about quantum networks with discrete-time dynamics. Its construction proceeds in three stages:

1. **Base layer.** The language **ZXLO** (introduced in Section 2) combines the ZX calculus with linear optics, incorporating a node for dual-rail encoding.

2. **Channel layer.** The `Channel` construction (Section 3) generalizes classical–quantum maps [22] to discrete-variable linear optics. It introduces a syntax based on Kraus maps, formalizing previously ad hoc variables in diagrams [5, 34] and extending them with classical annotations to model feedforward and coarse-graining of measurement outcomes.

3. **Stream layer.** The `Stream` construction (Section 4) builds on intensional monoidal streams [33] and the notion of observational equivalence from [15], enriching our syntax with time-indexed diagrams and feedback loops. It further extends previous work by introducing novel rewriting rules and a recursive `unroll` operation, which enables inductive reasoning about protocol executions.

This language yields formal definitions of determinism and universality for hybrid qubit-photonic channels, and supports the representation of practical optical modules such as routers, delays, and emitters. Its structure reflects three core features of networked architectures: (i) a qubit-photon interface that allows encoding units of quantum information in photonic modes; (ii) measurement and classical feedforward in the style of measurement-based quantum computing (MBQC); (iii) protocols with discrete-time dynamics involving both quantum and classical memory. The graphical framework developed in this paper forms the basis of the Python package Optyx [59], implemented within the DisCoPy ecosystem [39, 102].

1.2 Applications

Characterization of fusion measurements As a first application of our framework, in Section 5, we derive formal representations of fusion measurements in the ZX calculus, and we characterize all locally equivalent measurements whose Pauli byproducts can be corrected. We classify as 'green failure' all such measurements which preserve entanglement of any graph state under the fusion failure outcome. We then narrow down which of these induce correctable Pauli errors in both success and failure outcomes; see Theorem 5.9. This class includes phase gadgets, an important family of non-Clifford entangling gates [56]. By additionally restricting to stabilizer measurements, the two most commonly used fusion measurements — the Type II fusion of [8, 14] and the CZ fusion of [42, 64] — naturally arise from our characterization as X and Y fusions, respectively.

Proofs of determinism In MBQC, computation proceeds by measuring a graph state resource, with undesired outcomes corresponding to Pauli errors. Determinism is guaranteed when these errors can be corrected by classical control according to *flow structure* [13] on the underlying graph. Current formal models, however, do not handle two-qubit measurements such as fusions.

In Section 6, we introduce a notion of *partially static flow* for fusion networks which enables correction of Pauli errors while avoiding active classical control on fusion nodes. We prove that these fusion networks can be implemented by a deterministic pattern where fusion operations precede single-qubit measurements (Theorem 6.11), and that any decomposition of an open graph as a fusion network of X and Y fusions admits partially static flow provided the original graph has Pauli flow (Theorem 6.14). This yields the first formal analysis of photonic fusion errors, establishing flow conditions for determinism in fusion-based MBQC.

Proofs of universality Universality —- the ability of a quantum protocol to simulate any other —- is particularly challenging in linear optics, where entanglement can only be generated probabilistically. Our framework expresses entire photonic architectures as single diagrams, enabling their properties and expressivity to be investigated by inductive, graphical reasoning. We give a powerful example of this form of reasoning in Theorem 4.15, showing that the honeycomb lattice with measurements restricted to the YZ-plane is universal for quantum computing.

In Section 7.1, we prove the correctness of new repeat-until-success (RUS) protocols that boost the success probability of fusion measurements with green failure, given access to entangled resource states. As special cases, this recovers the RUS optical CZ gates of [64] and the boosted Bell measurement of [62]. The proof uses the unrolling technique of Section 4 and, to our knowledge, constitutes the first entirely graphical proof by induction. We end Section 7 by comparing two approaches to universal photonic quantum computing: lattice-based and reprogrammable emitter-based architectures. We provide graphical proofs of determinism and universality for minimal examples of each, allowing us to appreciate the scaling of these approaches. Our constructive proof for emitter-based architectures demonstrates how any given MBQC pattern can be compiled into a sequence of instructions for a concrete optical setup, without resorting to universal graph states. This indicates an alternative path to near-term photonic quantum computing where programmable setups are enhanced and optimized by compilation techniques.

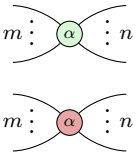
Part I
Graphical framework for quantum protocols

2 Base language

We introduce the graphical notations used in this paper, allowing us to relate linear optical circuits and ZX-diagrams via the dual-rail encoding. This fixes our base language of linear maps **ZXLO** generated by ZX diagrams, linear optical circuits and a triangle node introduced at the end of the section. The graphical framework developed in subsequent sections is however independent of this choice of linear maps, and thus the reader may modify the base language as long as the new generators have an interpretation as bounded linear maps.

2.1 ZX calculus

The ZX calculus is a graphical language used to represent and reason about qubit quantum computation. Its elementary building blocks are the green *Z-spider* and the red *X-spider* (therefore ZX).



$$\llbracket\cdot\rrbracket \quad |0\rangle^{\otimes n}\langle 0|^{\otimes m} + e^{i\alpha}\,|1\rangle^{\otimes n}\langle 1|^{\otimes m} \tag{1}$$

$$\llbracket\cdot\rrbracket \quad |+\rangle^{\otimes n}\langle +|^{\otimes m} + e^{i\alpha}\,|-\rangle^{\otimes n}\langle -|^{\otimes m} \tag{2}$$

These spiders can have any number of input and output legs, corresponding to qubit ports, and they have a phase parameter α. Notice that since the $e^{i\alpha}$ function in the interpretation is 2π periodic, we can take the parameter of spiders modulo 2π. The last generator of the ZX calculus is the yellow *Hadamard box* —□— that corresponds to the Hadamard gate, $|0\rangle\langle +| + |1\rangle\langle -|$. We define the star symbol as the diagram corresponding to $\frac{1}{\sqrt{2}}$, that is, $\star :=$  . Using these building blocks, we are able to intuitively represent common quantum gates as well as any unitary.

Further to its universal representational power for unitary maps, the ZX calculus also comes equipped with a set of graphical rewrite rules, presented in Figure 3. These rules provide a powerful mechanism for transforming diagrams, which can be used to establish equivalences between quantum circuits and to formally verify quantum protocols. Crucially, this rule set is known to be complete for qubit stabilizer quantum mechanics, meaning any equality that holds between qubit maps can be derived diagrammatically [52, 79].

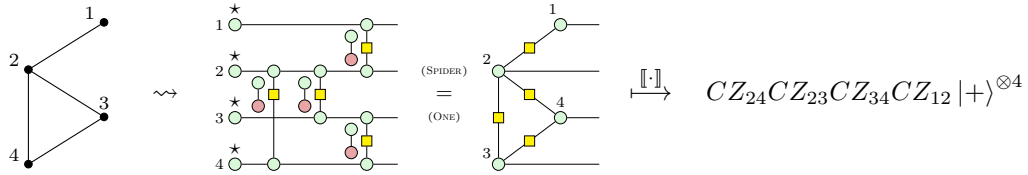
Remark 2.1. There are many different axiomatizations of the ZX calculus. The version presented in Figure 3 maintains exact equality for the stabilizer fragment. Versions of the ZX calculus can be found in [51] for the Clifford+T fragment, and in [104] for the full language.

A ZX-diagram with no inputs or outputs corresponds to some scalar, which we sometimes choose to omit from calculations. We use $=$ when a rewrite preserves equality on the nose, and $\approx$ denotes equality *up to some non-zero scalar*.

2.2 MBQC graphs

The ZX calculus is closely related to the measurement-based model of quantum computing. In MBQC, computation is performed in two stages: (i) a *graph state* is prepared and (ii) it is processed by a sequence of *single-qubit measurements*. A graph state associated with the graph $G = (V, E)$ is an entangled quantum state constructed by preparing a qubit for each vertex in the $|+\rangle$ state and applying CZ gates for each edge. We may depict graph states equivalently as ZX diagrams or qubit circuits.

Example 2.2.



$$\xmapsto{[\![\cdot]\!]} \quad CZ_{24}CZ_{23}CZ_{34}CZ_{12}\,|+\rangle^{\otimes 4}$$

Qubits can be inputs or outputs, which we depict by connecting wires to the left or right boundaries of the diagram.

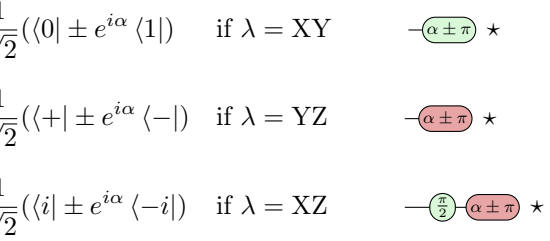
Definition 2.3 (Open graph)**.** An open graph is a tuple (G, I, O), where $G = (V, E)$ is an undirected graph, and $I, O \subseteq V$ are (possibly overlapping) subsets representing inputs and outputs. We use the notations $\overline{O} \coloneqq V \backslash O$ for the non-output and $\overline{I} \coloneqq V \backslash I$ for the non-input vertices.

During computation, every non-output vertex of the graph is measured in a certain basis specified by a measurement plane (λ) and angle (α).

Definition 2.4 (Labelled open graph)**.** A labelled open graph is a tuple $\mathcal{M} = (G, I, O, \lambda, \alpha)$, where (G, I, O) is an open graph, $\lambda : \overline{O} \to \{\,\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}\,\}$ is an assignment of measurement planes, and $\alpha : \overline{O} \to [0, 2\pi)$ assigns measurement angles to each non-output qubit.

We use the following notation to denote pure single qubit effects (the corresponding states are defined analogously).

$$\langle \pm_{\lambda,\alpha}| = \begin{cases} \frac{1}{\sqrt{2}}(\langle 0| \pm e^{i\alpha} \langle 1|) & \text{if } \lambda = \mathrm{XY} \\ \frac{1}{\sqrt{2}}(\langle +| \pm e^{i\alpha} \langle -|) & \text{if } \lambda = \mathrm{YZ} \\ \frac{1}{\sqrt{2}}(\langle i| \pm e^{i\alpha} \langle -i|) & \text{if } \lambda = \mathrm{XZ} \end{cases}$$



Any labelled open graph defines a target linear map which corresponds to the quantum computation that we want to execute. To ensure such a map is well defined, we provide a measurement pattern which contains a concrete sequence of instructions to generate the graph.

Definition 2.5. Suppose $\mathcal{M} = (G, I, O, \lambda, \alpha)$ is a labelled open graph. The *target linear map of* $\mathcal{M}$ is given by

$$T(\mathcal{M}) \coloneqq \left(\prod_{i \in \overline{O}} \langle +_{\lambda(i),\alpha(i)}|_i \right) E_G N_{\overline{I}},$$

where $E_G \coloneqq \prod_{i \sim j} CZ_{ij}$ and $N_{\overline{I}} \coloneqq \prod_{i \in \overline{I}} |+\rangle_i$.

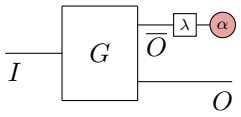
As shown in [5], a labelled open graph is completely characterized by a ZX diagram in MBQC form.

Definition 2.6. [5] A graph state diagram is a ZX diagram where all vertices are green, all the connections between vertices are Hadamard edges and a single output wire is incident on each vertex in the diagram. A ZX diagram is in MBQC form if it consists of a graph state diagram in which each vertex may additionally be connected to:

- an input (in addition to its output), and
- a measurement effect in one of the measurement planes $\{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}\}$ (instead of the output).

Given a ZX diagram D in MBQC form, its *underlying graph* $G(D)$ is the labelled open graph $(G, I, O, \lambda, \alpha)$ where G is the graph whose vertices are the green spiders of D and whose edges are the Hadamard edges of D, the inputs (outputs) $I, O \subseteq G$ are the spiders connected to the inputs (outputs) of D, and the pair λ, α are the measurement effects on each spider.

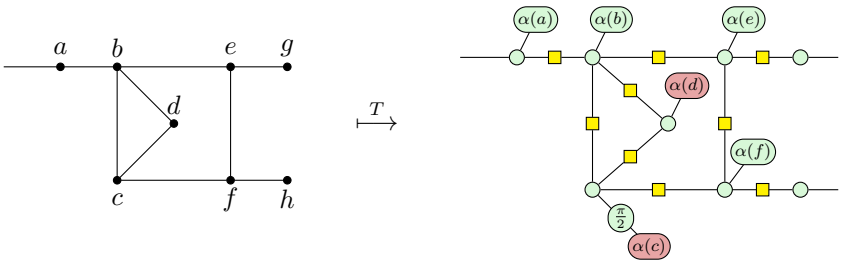
We introduce a scalable notation for MBQC form ZX diagrams, which are completely characterized by an open graph (G, I, O) and a choice of measurement planes and angles λ, α on non-output qubits:



Proposition 2.7. *[5] For any labelled open graph $\mathcal{M} = (G, I, O, \lambda, \alpha)$ there is a ZX diagram D in MBQC form with $G(D) = \mathcal{M}$ and whose interpretation is the target linear map of $\mathcal{M}$, i.e. $T(\mathcal{M}) = [\![D]\!]$.*

We can thus represent $T(\mathcal{M})$ in the ZX calculus by attaching the appropriate effects to the dangling qubits in the graph state.

Example 2.8.



where the input set is $I = \{\, a \,\}$, the set of outputs is $O = \{\, g, h \,\}$, and the measurement planes are $\lambda(v) = \mathrm{XY}$ for all $v \in \{\, a, b, e, f \,\}$, $\lambda(d) = \mathrm{YZ}$, and $\lambda(c) = \mathrm{XZ}$.

The above description only discussed MBQC with post-selected measurement outcomes, i.e. assuming determinism of measurements. However, quantum measurements are fundamentally probabilistic processes: they may or may not induce Pauli errors upon observation. In Section 3, we formalise measurements as completely positive maps or quantum channels.

2.3 Linear optical circuits

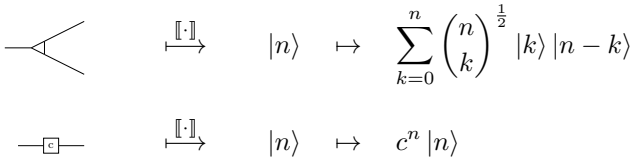
Linear optical circuits are the basic ingredients of photonic computing. They are generated by two physical gates: beam splitters and phase shifts. When equipped with n-photon state preparations and number-resolving photon detectors, they give rise to quantum statistics [1]. We depict the above-mentioned components, respectively, as follows:

We call the graphical language generated by the above components **LO**. Graphical axiomatisations of these components can be found in [20, 30, 47]. We use a special notation ○— := 0— for the 'empty' state. Diagrams in **LO** have an interpretation as linear maps acting on tensor products

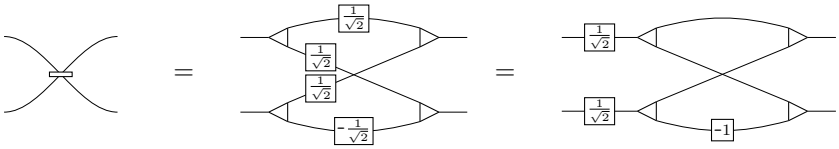
of $\mathcal{F}(\mathbb{C})$ — the bosonic Fock space with a single degree of freedom, also called *bosonic mode*. Formally, the bosonic Fock space over a vector space $\mathcal{H}$ is given by $\mathcal{F}(\mathcal{H}) = \bigoplus_n \mathcal{H}^{\tilde{\otimes} n}$ where $\tilde{\otimes}$ denotes the quotient of the tensor product by $x \otimes y \sim y \otimes x$ and $\bigoplus$ denotes the infinite direct sum. $\mathcal{F}(\mathcal{H})$ is usually completed to a Hilbert space but we here focus on states containing finitely many photons and thus use the above definition. For a finite number of modes m, the Fock space is isomorphic to the tensor product of m copies of the single bosonic mode $\mathcal{F}(\mathbb{C}^m) \simeq \mathcal{F}(\mathbb{C})^{\otimes m}$ [30]. Given an **LO** circuit on m modes, the amplitudes of different input-output pairs can be computed by taking permanents of an underlying $m \times m$ unitary matrix. We fix the interpretation of the phase shift of angle θ to be the matrix $\left(e^{i\theta}\right)$, and of the beam splitter to be the Hadamard matrix, $\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$. The underlying matrix of arbitrary circuits is obtained by a simple block-diagonal matrix multiplication.

Remark 2.9. Throughout the paper, we use two different types of classical variables. Underlined variables are *outcomes* of a quantum measurement: they can take different values probabilistically. Other variables are *controlled*: they have a fixed value, set before the computation is executed. For example, in the picture above, the number of prepared photons n is controlled while the number of detected photons $\underline{n}$ is a probabilistic outcome. We formalise this notation in Section 3.1.

Circuits in **LO** can be further decomposed using the generators of the QPath calculus [30], given by the following maps:



for $c \in \mathbb{C}$. This enables a rewriting system given in Section A, and a more fine-grained diagram for the beam splitter,



which we use in Section B. Note that the QPath maps are bounded on the Fock space as defined above, yet unbounded on its Hilbert space completion. We must therefore be careful to use them only if there are finitely many photons. This assumption is valid throughout the paper and further discussed in Section 3.2.

2.4 Dual-rail encoding

In photonic quantum computing, qubits are usually encoded by a photon in a pair of bosonic modes, a method known as *dual-rail encoding* [57]. These could be two possible positions of the photon (spatial modes), or any other binary degree of freedom of the photon such as polarisation. We use a 'double wire' to denote dual-rail modes. These wires are interpreted as $\mathcal{F}(\mathbb{C}^2)$, the bosonic Fock space over a qubit, meaning that there can be any number of qubits in the same dual-rail mode. Linear optical operations on these modes are defined from **LO** by using the following maps:

These two maps are inverses of each other, corresponding to the natural isomorphism $\mathcal{F}(\mathbb{C}^2) \simeq \mathcal{F}(\mathbb{C}) \otimes \mathcal{F}(\mathbb{C})$. We wish to represent processes acting on dual-rail qubits using the ZX calculus [21]. However, ZX diagrams act on qubit spaces of the form $(\mathbb{C}^2)^{\otimes m}$ and there is no standard way of extending this action to $\mathcal{F}(\mathbb{C}^2)^{\otimes m}$. There is, however, an isometry $\mathbb{C}^2 \to \mathcal{F}(\mathbb{C}^2)$, encoding a qubit state into its dual-rail representation. We call it 'triangle' and represent it as follows:

 (3)

Note that the adjoint of the triangle is a projector onto the qubit subspace, and we never use it in this paper. We can now translate between dual-rail circuits and ZX diagrams using this graphical component. For example, the qubit computational basis states are given by the dual-rail states.

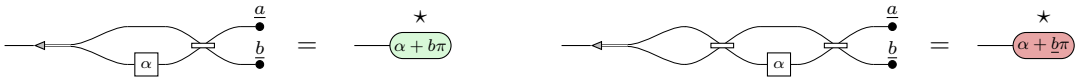
Rewrite 1.

Going further, the following equations imply that any single-qubit unitary can be realised on dual-rail qubits using only linear optical instruments.

Rewrite 2.

Similarly, we may perform any single-qubit measurement on dual-rail qubits using photon detectors.

Rewrite 3.



By pushing triangles from left to right in a dual-rail diagram, we compute the action of this diagram assuming that a single photon is input in each dual-rail mode.

Remark 2.10. Our results in this paper focus on dual-rail encoded qubits. However, the only properties we use of the dual-rail encoding are the equations given in this subsection. For a straightforward generalisation to qubit encodings in n (rather than 2) optical modes, one may replace the beam splitter and tensored phase shift with $n \times n$ unitaries satisfying the same equations. For a more involved generalisation to *qudit* encodings, similar equations could be used to relate linear optics and the qudit ZX calculus [89].

3 Channel construction

The graphical language **ZXLO** described so far has a standard interpretation in *pure* quantum mechanics, consisting of an assignment of a linear map $[\![D]\!]$ to any diagram $D \in \mathbf{ZXLO}$. In order to interpret these diagrams as *quantum channels*, we need to define notions of environment and classical interface on the diagram. In categorical quantum mechanics, the standard approach to modeling quantum channels uses Frobenius algebras [2, 23] and employs a diagrammatic doubling procedure based on Stinespring dilation to capture mixed-state quantum mechanics. However, these approaches require the burden of either having twice as large diagrams, or keeping track of both quantum and classical wire types. Here, working with Kraus decompositions fixed by measurements, we utilize variables to represent classical data for cleaner syntax. This approach can be found in the literature [34], but we here extend the notation to also model feedforward of classical information as variable annotations on the diagram. This gives us a premonoidal syntax [88] for describing quantum channels where the order of boxes in the diagram defines the allowed classical annotations. We give a formal interpretation for this notation in terms of completely positive maps between vector spaces, focusing on their action on discrete-variable photonic states. We then show that the determinism results of [13, 27] can be naturally expressed in the language of $\texttt{Channel}(\mathbf{ZX})$ diagrams, giving us tools for proving determinism of quantum channels which we will extend in subsequent sections.

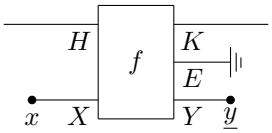
3.1 Kraus notation for abstract channels

Let $\mathbf{C}$ be any base graphical language of linear maps. We extend $\mathbf{C}$ with classical annotations to construct a language of classical-quantum maps $\texttt{Channel}(\mathbf{C})$ Fix a global set of variables $x, y \in \chi$ and consider the abstract notion of Kraus map over a base graphical language.

Definition 3.1 (Kraus map)**.** A Kraus map $f : Q \to Q'$ over $\mathbf{C}$ is given by the following data:

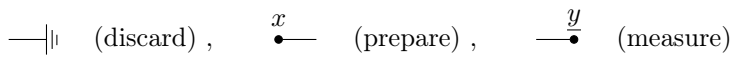
1. a type E in $\mathbf{C}$, called the environment,
2. a pair of input and output typed variables $x : X$ and $y : Y$ where X, Y are objects of $\mathbf{C}$, called the classical interface,
3. a morphism of type $f : H \otimes X \to K \otimes E \otimes Y$ in $\mathbf{C}$.

and denoted as follows:



A Kraus map is called *pure* if $X = Y = E = I$ are the unit of the tensor.

We consider arbitrary compositions of Kraus maps over $\mathbf{C}$, in sequence and in parallel. This is equivalent to considering arbitrary diagrams built from $\mathbf{C}$ and the following additional generators on every wire:



(4)

We however distinguish between the two types of composition below:

Intuitively, measurements have classical outcomes modeled as side-effects. In the first diagram the classical output variable y can influence the input variable x, while in the second diagram it cannot. Thus, any diagram comes with a total order on the measurement and preparation commands, which induces a strict ordering of the classical variables appearing in the diagram. For a diagram D we denote by $X_D \subset \chi$ the set of preparation variables in D, by $Y_D \subset \chi$ the set of measurement variables in D, and by $<_D$ the total order on $X_D \cup Y_D$.

Definition 3.2 (Causal annotations)**.** Fix a set of function symbols f, g and constant symbols a, b. Given a diagram D built from $\mathbf{C}$ and the generators (4), a classical annotation $\mathcal{E}$ for D is a set of typed *coarse-graining variables* $Z_D \subset \chi \backslash (X_D \cup Y_D)$, and a set of functional relations including:

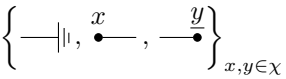
1. *coarse-graining annotations* for every $z \in Z_D$ and some $\nu_i \in Y_D \cup Z_D \backslash \{ z \}$, of the form
$$\underline{z} = g(\nu_1, \ldots, \nu_n) \,, \tag{5}$$
2. *feedforward annotations* for some $x \in X_D$ and $\nu_i \in Y_D \cup Z_D$, of the form
$$x = f(\nu_1, \ldots \nu_n) \,, \tag{6}$$
3. *post-selection annotations* for some constant symbol a and $y \in Y_D$, of the form
$$\underline{y} = a \,. \tag{7}$$

This defines a relation $\leq_{D,\mathcal{E}}$ on the set of variables $X_D \cup Y_D \cup Z_D$ where $\nu \leq_{D,\mathcal{E}} \mu$ whenever either $\nu = \mu$, or $\nu <_D \mu$, or there is an annotation in $\mathcal{E}$ such that ν appears on the right-hand side and μ appears on the left-hand side. We say that a classical annotation $\mathcal{E}$ is *causally compatible* with D if $\leq_{D,\mathcal{E}}$ is a partial order, i.e. reflexive, antisymmetric and transitive.

Annotations are interpreted as functions acting on classical data, allowing to specify the classical input variables of boxes with respect to previous output variables in the diagram.

Definition 3.3 (Channel)**.** A channel diagram $D : H \to K$ in $\texttt{Channel}(\mathbf{C})$ consists of the following data:

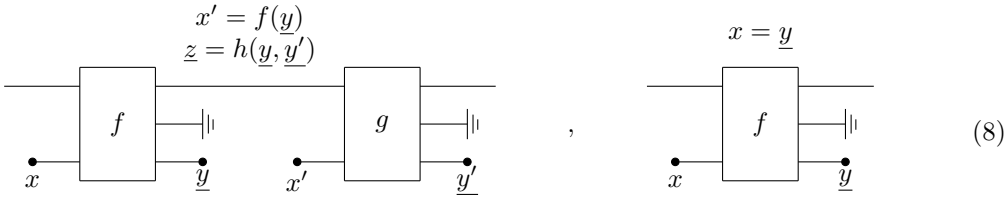
1. a diagram D built from $\mathbf{C}$ and the following additional generators



2. a classical annotation $\mathcal{E}_D$ causally compatible with D.

The set of variables appearing in a channel diagram D is $X_D \cup Y_D \cup Z_D$. The classical input variables I_D are the variables in X_D that do not appear in any feedforward annotation. The output variables O_D are the variables in $Y_D \cup Z_D$ that do not appear on the right-hand side of any coarse-graining or feedforward annotation, or on the left-hand side of a post-selection annotation.

For example, the channel diagram below (left) has causally compatible coarse-graining and feedforward annotations with $X_D = \{ x, x' \}$, $Y_D = \{ y, y' \}$, $Z_D = \{ z \}$ and $I_D = \{ x \}$, $O_D = \{ z \}$. Instead, the annotation on the right is not causally compatible with the diagram as x appears before y, and thus gives an invalid channel diagram.



(8)

Parallel and sequential compositions of diagrams in $\texttt{Channel}(\mathbf{C})$ are given by composing the underlying diagrams and appending the sets of equations, after appropriate relabeling with fresh variables in χ. This ensures that the causal compatibility condition of Definition 3.2 is respected.

Remark 3.4. Formally, $\texttt{Channel}(\mathbf{C})$ is a premonoidal category [88] with monoidal center $\mathbf{C}+\{$—|ı|$\}$, that is, the interchange law holds for any pair of morphisms in $\texttt{Channel}(\mathbf{C})$ that does not involve a controlled preparation or a measurement. The coarse-graining and feedforward annotations may be viewed as operations acting on an implicit "runtime object" for $\texttt{Channel}(\mathbf{C})$ [95].

3.2 Interpretation as completely positive maps

Diagrams in **ZXLO** correspond to linear maps acting on tensor products of qubits and bosonic modes. We now give an interpretation of $\texttt{Channel}(\mathbf{ZXLO})$ diagrams as completely positive maps acting on mixed states with finitely many particles. For any object H of **ZXLO** consisting of m bosonic modes and k qubits, define the N-particle projector $P_{\leq N} : H \to H$ as the operator that acts as the identity on qubits and projects out all states with more than N photons over the m bosonic modes. A *finite-particle state* on H, also known as a regular state [17], can be defined as a positive operator $\rho : H \to H$ satisfying $\rho = P_{\leq N} \rho P_{\leq N}$ for some natural number N. We denote by $B_*(H)$ the space of all finite-particle states. In finite dimensions this coincides with the full space of bounded operators. For bosonic modes, it contains all discrete-variable states [17, 57, 91] and truncated optical states [73, 83, 87], widely used in the literature. Finite-particle states are finite-rank and trace-class so that $B_*(H)$ forms an inner-product space with the usual Hilbert-Schmidt inner product. We do not impose any normalisation condition on states, as this will allow us to extract physical quantities from our interpretation.

Fix a basis $\mathcal{B}_H$ of the vector space associated to any object in **ZXLO**. For the qubit space $\mathcal{B} = 2$ is the set with two elements, while for single-mode bosonic space we have $\mathcal{B} = \mathbb{N}$. For a channel diagram D, we denote by $\mathcal{B}_{X_D}, \mathcal{B}_{Y_D}, \mathcal{B}_{Z_D}$ the basis elements for the preparation, measurement and coarse-graining variables, respectively. Similarly, the unbound input and output classical variables range over the basis elements $\mathcal{B}_{I_D}$ and $\mathcal{B}_{O_D}$. We moreover fix an interpretation of each function (or constant) symbol as a function on the basis elements associated to its input and output variable types. We group the environments E_i of individual Kraus maps within a diagram D into a single environment $E_D = \bigotimes_i E_i$.

Definition 3.5 (Branches)**.** The branches of a diagram $D : H \to K$ in $\texttt{Channel}(\mathbf{ZXLO})$ with preparation commands in X_D and measurement commands in Y_D, are the collection of diagrams

$H \to K \otimes E_D$ in **ZXLO** obtained by setting different values for the variables $x \in \mathcal{B}_{X_D}$ and $y \in \mathcal{B}_{Y_D}$. We sometimes view a branch as a diagram $D_{x,y} : H \to K$ in $\texttt{Channel}(\textbf{ZXLO})$ by attaching the discard map to the environment E_D.

We can now give a direct interpretation of a channel D as a completely positive map $B_*(H) \otimes X' \to B_*(K) \otimes Y'$ where $B_*(H)$ is the space of finite-particle states $\rho : H \to H$, and X', Y' are the spaces of the input and output classical variables in D.

Definition 3.6 (CQ interpretation)**.** Given a diagram D in $\texttt{Channel}(\textbf{ZXLO})$, the classical-quantum map (CQ) interpretation is given by the following operator on finite-particle states:

$$[\![D]\!]_{CQ}(\rho) = \sum_{x,y,z \in \mathcal{B}_D \text{ satisfying } \mathcal{E}_D} |y'\rangle\, tr_E([\![D_{x,y}]\!]^\dagger \rho\, [\![D_{x,y}]\!])\, \langle x'|$$

where $x, y, z \in \mathcal{B}_D = \mathcal{B}_{X_D} \times \mathcal{B}_{Y_D} \times \mathcal{B}_{Z_D}$, $E = E_D$ is the total environment of the diagram, and $x', y' \in \mathcal{B}_{I_D} \times \mathcal{B}_{O_D}$ are the unbound input and output variables of the channel diagram.

To show that the above interpretation is sound and does not result in infinite quantities we prove the following proposition.

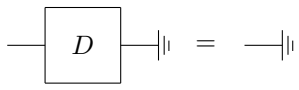
Proposition 3.7 (Soundness)**.** *For any diagram $D \in$ **ZXLO** and positive finite-particle state ρ, the CQ interpretation $[\![D]\!]_{CQ}(\rho)$ is a positive finite-particle state.*

Proof. Let $\rho : H \to H$ be a positive finite-particle state. Every generator G introduced in the previous section, except for the n-photon state preparation, commutes with the N-particle projector, $P_{\leq N} G = G P_{\leq N}$. The n-photon state preparation creates n additional photons and thus satisfies $(|n\rangle \otimes \texttt{id}_H) P_{\leq N} = P_{\leq N+n}(|n\rangle \otimes \texttt{id}_H)$ for any reference space H. Therefore for any diagram D in **ZXLO** we have $[\![D]\!]\, P_{\leq N} = P_{\leq N+a}\, [\![D]\!]$ where a is the total number of photon creations in D. Therefore for any x, y,

$$[\![D_{x,y}]\!]^\dagger \rho\, [\![D_{x,y}]\!] = [\![D_{x,y}]\!]^\dagger P_{\leq N} \rho P_{\leq N}\, [\![D_{x,y}]\!] = P_{\leq N+a}\, [\![D_{x,y}]\!]^\dagger \rho\, [\![D_{x,y}]\!]\, P_{\leq N+a}$$

gives a finite-particle state which is moreover positive by positivity of ρ. Any finite-particle state is trace-class, and the partial trace preserves positivity, so that $tr_E([\![D_{x,y}]\!]^\dagger \rho\, [\![D_{x,y}]\!])$ is also positive and bounded. It also has finitely many particles since the trace induces a sum over finitely many finite-particle states. Moreover, finite-particle states can only result in finitely many measurement outcomes, and the causal compatibility condition ensures that all the feedforward and coarse graining variables are deterministic functions of previous measurement variables and thus can only take finitely many values. For example, the invalid diagram on the right of Equation (8) could result in infinitely many consistent assignments of values to x and y, but this circularity is avoided in valid channel diagrams. Therefore the sum $\sum_{x,y,z \in \mathcal{B}_D \text{ satisfying } \mathcal{E}_D} tr_E([\![D_{x,y}]\!]^\dagger \rho\, [\![D_{x,y}]\!])$ has only finitely many terms, and thus gives a positive finite-particle state as required. □

Since all states considered are finite-rank and trace-class, the *trace operator* $\rho \mapsto \sum_{x \in \mathcal{B}_H} \langle x|\, \rho\, |x\rangle$ is well-defined and corresponds to the CQ interpretation of the discarding map —|||. We often restrict our attention to *trace-preserving* channels, satisfying:



(9)

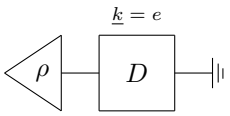
When D is a pure map the equation above corresponds to $[\![D]\!]^\dagger [\![D]\!] = I$, i.e. $[\![D]\!]$ is an isometry.

We interpret formal sums of diagrams in $\texttt{Channel}(\textbf{ZXLO})$ as the sum of their CQ interpretation. With the scalars present in **ZXLO** we will only be able to form positive real linear combinations of diagrams, since $[\![s]\!]_{CQ} = |\, [\![s]\!]\, |^2$ for any scalar diagram s in **ZXLO**. Note that this sum is akin to *mixing*, rather than superposition, which instead corresponds to summing the Kraus maps in **ZXLO** before constructing the channel. In other words:

$$[\![D]\!]_{CQ} + [\![D']\!]_{CQ} \text{ (mixing)} \neq [\![D + D']\!]_{CQ} \text{ (superposition)}$$

Causal maps are closed under taking *probability distributions*: for a discrete probability distribution $p_i \in [0,1]$ with $i \in X$, X finite, and causal maps $\{ f_i \}_{i \in X}$, the map defined by $\sum_{i \in X} p_i f_i$ is also causal.

We can represent a quantum channel with classical output as a diagram D labelled by an output variable $\underline{k}$. Then, the probability of an outcome e, given an input state ρ, is obtained by setting $\underline{k} = e$ in D and tracing out the remaining outputs:

$P_D(\underline{k} = e \,|\, \rho) =$ 

We can use these post-selection annotations to define a notion of *implementation* between channels.

Definition 3.8 (Probabilistic implementation)**.** We say that a channel D implements a channel C with probability p if there exists a function $\underline{s} = f(\underline{y_1}, \ldots, \underline{y_m})$ of the classical outcome variables in D such that $[\![D_{\underline{s}=1}]\!]_{CQ} = p\, [\![C]\!]_{CQ}$. We say that D implements C deterministically if $[\![D]\!]_{CQ} = [\![C]\!]_{CQ}$.

Measurements in quantum mechanics induce probabilistic branching and mixing. To result in a *deterministic* computation, all the individual branches of the channel must contribute to the same process.

Definition 3.9 (Determinism)**.** A channel D is deterministic if all the branches are proportional to each other in the CQ interpretation. It is strongly deterministic if all branches are equal up to a global phase, i.e. if all the branches are equal to each other in the CQ interpretation.

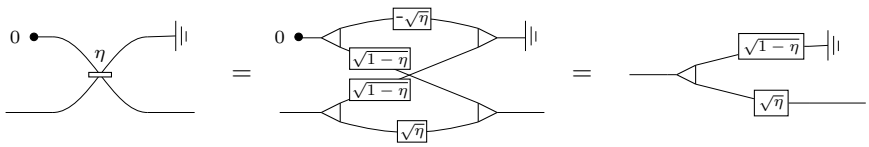
Thus, a channel D is deterministic if for any pattern of measurement outcomes, it results in the same pure computation. It is strongly deterministic if these outcome patterns have equal probability.

3.3 Qubit and optical channels

We now apply the `Channel` construction to our base language **ZXLO** of ZX diagrams and linear optical circuits, and give examples of how it can be used to compute properties of quantum channels.

Discarding and noise We start by considering the class of processes generated by only pure maps and discarding. The discarding maps for each space together with the relations between them are as follows:

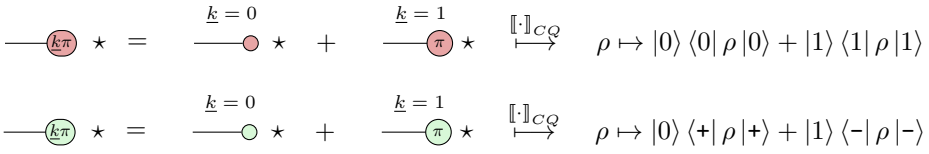
Using these maps we can represent common error channels from their Kraus decomposition. For example, the photon loss channel with transmittivity η is modeled as a beam splitter with an output discarded [82]:



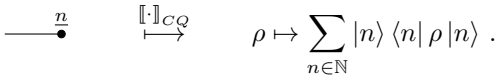
Similarly, the single-qubit bitflip error channel with probability p can be written as:

using the single-qubit state $[\![|\sqrt{p}\rangle]\!] = \sqrt{p}\,|1\rangle + \sqrt{1-p}\,|0\rangle$. Our focus in this paper is however on *noiseless channels* where the environment is fully observed by the experimenter. These are simply defined as channels D with a trivial environment $E_D = I$, i.e. such that the discarding maps do not appear in the diagram.

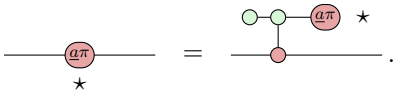
Measurements and coarse-graining Even in a noiseless environment, destructive measurements can lead to branching and non-determinism in the CQ interpretation. The X and Z single-qubit measurements can be written as:



The photon-number resolving measurement on an optical mode has the following interpretation:

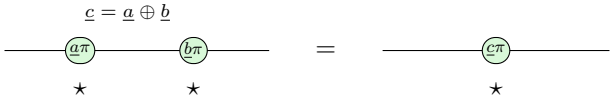


We can treat the diagrams built from these measurement commands as parametrised diagrams in **ZXLO**. For example, we can define spiders with a classical output variable, corresponding to entangling a qubit to an ancilla and measuring it.



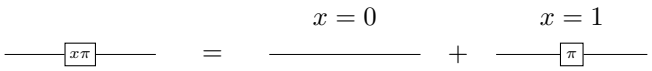
While these spiders satisfy the usual spider fusion law, note that coarse-graining equations between output variables have an action on the scalars in the diagram. Diagrams with coarse graining annotations satisfy the following rule.

Rewrite 4.

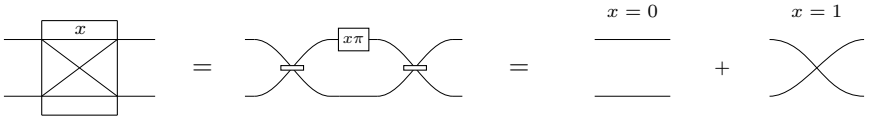


This rewrite removes a $\star$, a pair of variables, and an annotation from the diagram to correctly account for the coarse-grained probabilities: there are two possible outcomes for c, each happening with probability $\llbracket\star\rrbracket_{CQ} = \frac{1}{2}$. Spiders labeled by a classical outcome allow to construct general projective-valued measures (POVMs), such as ancilla quantum measurements. The outcomes correspond to Pauli byproducts and can be propagated through the diagram via ZX rewrites.

Feedforward and correction Feedforward is crucial to perform universal quantum computation with linear optics. In this work, we only use bit-controlled operations generated by the optical controlled phase flip gate, given by:



This process acts as the identity if the control parameter $x = 0$ and as the phase flip if $x = 1$. Formally, the Kraus map for this controlled process should be considered an additional generator of our base language, but the simplified syntax above directly captures the properties of its interpretation. Switches can be built using bit-controlled phases:



(10)

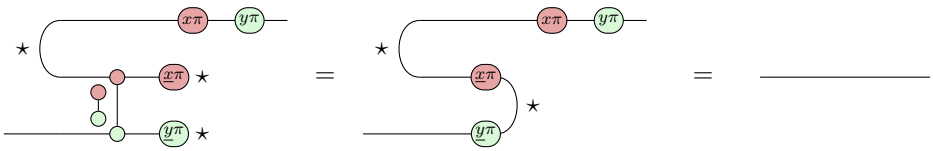
and, vice versa, a composition of two switches can be used to build a controlled phase. Probabilistic controlled phase gates can be obtained with passive linear optics [63, 90] but higher fidelities can be achieved with active optical switches [108].

On qubits, we use the classically controlled X and Z gates:

We moreover encode feedforward operations by reusing variables in the diagram. Diagrams with feedforward annotations can be simplified using the following rule.

Rewrite 5.

As an example, we can use this to prove the teleportation protocol with a perfect Bell measurement:



where the last step uses the feedforward rewrite twice. This rewrite allows to remove measurement and correction commands from the diagram, while increasing the overall success probability (removing stars). We can see the rewrite sequence above as showing *determinism* for the teleportation protocol. This correction argument can in fact be generalized to arbitrary ZX diagrams with flow, as we summarise in the remainder of this section.

3.4 Pauli flow and determinism in qubitq patterns

The notion of determinism of Definition 3.9 has been studied in detail for *qubit channels* [13, 27]. We now recast these results in our framework. *Measurement patterns* [27] are a declarative language for MBQC, that describe how qubits are prepared, entangled, corrected, and measured. They are usually defined as follows:

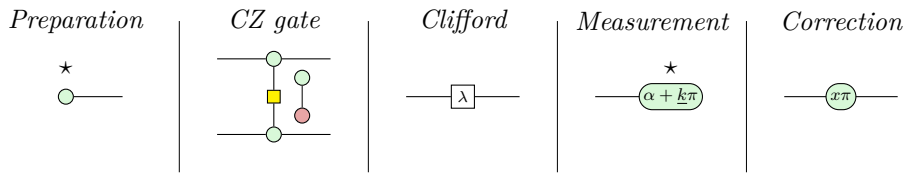
Definition 3.10 (Measurement pattern [27])**.** A *measurement pattern* consists of an n-qubit register V with distinguished sets $I, O \subseteq V$ of input and output qubits and a sequence of commands consisting of the following operations:

- Preparations N_i, which initialise a qubit $i \in \overline{I}$ in the state $|+\rangle$.
- Entangling operators E_{ij}, which apply a CZ-gate to two distinct qubits i and j.
- Destructive measurements $M_i^{\lambda,\alpha,\underline{s}}$, which project a qubit $i \in \overline{O}$ onto the orthonormal basis $\{|+_{\lambda,\alpha}\rangle, |-_{\lambda,\alpha}\rangle\}$, where $\lambda \in \{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}\}$ is the measurement plane, α is the non-corrected measurement angle . The projector $|+_{\lambda,\alpha}\rangle\langle+_{\lambda,\alpha}|$ corresponds to outcome $\underline{s} = 0$ and $|-_{\lambda,\alpha}\rangle\langle-_{\lambda,\alpha}|$ corresponds to outcome $\underline{s} = 1$.
- Clifford operations C_i, which act on qubit i by applying any Clifford unitary generated by the S and H gates.
- Corrections $[X_i]^t$, which depend on a measurement outcome (or a linear combination of measurement outcomes) $t \in \{0, 1\}$ and act as the Pauli-X operator on qubit i if t is 1 and as the identity otherwise,
- Corrections $[Z_j]^s$, which depend on a measurement outcome (or a linear combination of measurement outcomes) $s \in \{0, 1\}$ and act as the Pauli-Z operator on qubit j if s is 1 and as the identity otherwise.

A measurement pattern is *runnable* if no command acts on a qubit already measured or not yet prepared (except preparation commands) and no correction depends on a qubit not yet measured.

Runnable measurement patterns can be viewed as noiseless circuits in $\mathtt{Channel}(\mathbf{ZX})$.

Proposition 3.11. *Any runnable measurement pattern uniquely corresponds to a diagram in* `Channel`(**ZX**) *generated by the following operations:*

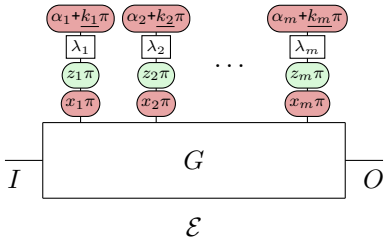


where $\lambda \in \{ I, S, H \}$ *is a Clifford map, and* $\alpha, \omega \in [0, 2\pi)$ *is an arbitrary angle.*

Proof. The defining operations of measurement patterns can all be built from the above generators, thus any runnable list of commands from Definition 3.10, defines a unique premonoidal diagram. □

We can rewrite any measurement pattern to a `Channel`(**ZX**) diagram in a special form where all corrections appear before single-qubit measurements.

Definition 3.12 (Channel MBQC form)**.** A diagram D in `Channel`(**ZX**) is in MBQC form if there is a labelled open graph $(G, I, O, \lambda, \alpha)$ such that D can be written as:



where x_i, z_i are control variables, $\underline{k_i}$ are output variables, $\mathcal{E}$ is a set of feedforward annotations (exclusively) and G denotes a graph state ZX diagram, called the *underlying topology* of D.

Proposition 3.13. *Any MBQC-form diagram in* `Channel`(**ZX**) *defines a measurement pattern with the same CQ interpretation.*

Proof. To go from an MBQC form diagram to a measurement pattern, we simply (1) decompose the graph state diagram with CZ gates and (2) extract the control variables as X and Z correction gates using spider un-fusion. □

In the context of measurement patterns, we require an even stronger form of determinism.

Definition 3.14 (Uniform and stepwise determinism)**.** A measurement pattern is (strongly) deterministic if it is (strongly) deterministic as a diagram in `Channel`(**ZX**). A measurement pattern is uniformly deterministic if it is deterministic for all choices of measurement angles α_i. It is stepwise deterministic if all the patterns $\{ P_i \}_{i=1}^{m}$ obtained by truncating P after the ith measurement command, and adding back every correction command that depends on the first i measurement variables, are deterministic.

Flow structure gives sufficient (and sometimes necessary) conditions for a labelled open graph to be implementable by a deterministic measurement pattern. It incorporates a time-ordering of the measurements and a function that indicates where to correct undesired measurement outcomes. Gflow (or generalized flow) is a specific type of flow structure that ensures that the target linear map is an isometry for all choices of measurement angles.

Definition 3.15. For a graph $G = (V, E)$ and a subset of its vertices $K \subseteq G$, let $\texttt{Odd}(K) := \{ u \in V : |N(u) \cap K| \equiv 1 \mod 2 \}$ be the *odd neighbourhood* of K in G, where $N(u)$ is the set of neighbours of u.

Definition 3.16 (Gflow [13])**.** An open graph (G, I, O) labelled with measurement planes $\lambda : \overline{O} \to \{ \mathrm{XY}, \mathrm{XZ}, \mathrm{YZ} \}$ has generalized flow (or gflow) if there exists a map $g : \overline{O} \to \mathcal{P}(\overline{I})$, where $\mathcal{P}$ is the power set function, and a strict partial order $<$ over V such that for all $v \in \overline{O}$:

1. for all $w \in g(v)$ if $v \neq w$ then $v < w$
2. for all $w \in \mathtt{Odd}(g(v))$ if $v \neq w$ then $v < w$
3. $\lambda(v) = \mathrm{XY} \implies v \notin g(v) \wedge v \in \mathtt{Odd}(g(v))$
4. $\lambda(v) = \mathrm{XZ} \implies v \in g(v) \wedge v \in \mathtt{Odd}(g(v))$
5. $\lambda(v) = \mathrm{YZ} \implies v \in g(v) \wedge v \notin \mathtt{Odd}(g(v))$

The set $g(v)$ is called the *correction set of* v.

Extending the notion of gflow, *Pauli flow* allows for the flow structure to take into account which vertices are measured in a Pauli basis.. In this setting, the function λ defining measurement planes is of type $\lambda : \overline{O} \to \{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$, while the function α is only defined for nodes $v \in G$ when $\lambda(v) \in \{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}\}$. In other words, the pattern specifies vertices that are measured in the X, Y, or Z basis. For these specific measurements, the correction set is less restricted, and we obtain the conditions below.

Definition 3.17 (Pauli flow [13, 98])**.** An open graph (G, I, O) labelled with measurement planes $\lambda : \overline{O} \to \{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}, \mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$ has Pauli flow if there exists a map $p : \overline{O} \to \mathcal{P}(\overline{I})$ and a strict partial order $<$ over V such that:

1. for all $w \in p(v)$ if $\lambda(w) \notin \{\mathrm{X}, \mathrm{Y}\} \wedge v \neq w$ then $v < w$
2. for all $w \in \mathtt{Odd}(p(v))$ if $\lambda(w) \notin \{\mathrm{Y}, \mathrm{Z}\} \wedge v \neq w$ then $v < w$
3. for all $w \leq v$ if $\lambda(w) = \mathrm{Y} \wedge v \neq w$ then $(w \in p(v) \iff w \in \mathtt{Odd}(p(v)))$
4. $\lambda(v) = \mathrm{XY} \implies v \notin p(v) \wedge v \in \mathtt{Odd}(p(v))$
5. $\lambda(v) = \mathrm{XZ} \implies v \in p(v) \wedge v \in \mathtt{Odd}(p(v))$
6. $\lambda(v) = \mathrm{YZ} \implies v \in p(v) \wedge v \notin \mathtt{Odd}(p(v))$
7. $\lambda(v) = \mathrm{X} \implies v \in \mathtt{Odd}(p(v))$
8. $\lambda(v) = \mathrm{Z} \implies v \in p(v)$
9. $\lambda(v) = \mathrm{Y} \implies (v \notin p(v) \wedge v \in \mathtt{Odd}(p(v))) \vee (v \in p(v) \wedge v \notin \mathtt{Odd}(p(v)))$

To understand the definition above, first note that for measurements in the planes $\{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}\}$, the conditions are the same as for gflow. The above conditions $7-9$ are obtained by taking the pairwise disjunctions '$\vee$' of conditions $4-6$, using the fact that each Pauli measurement belongs to a pair of planes. To obtain condition 1, note that a Pauli X error on a qubit measured in the X basis only induces a global phase on the state. Therefore, we must not correct X errors on X measurements. Condition 2 is the equivalent condition for Z errors and condition 3 ensures that Y measurements need only carry $Y = XZ$ corrections. A consequence is that Y measurements in a graph with Pauli flow need not carry corrections, justifying conditions $1-2$.

We can now state the main result of [13] which ensures that labelled open graphs with flow are implementable by deterministic patterns.

Theorem 3.18. *[13] If a labelled open graph $\mathcal{M}$ has generalized flow, then the pattern defined by:*

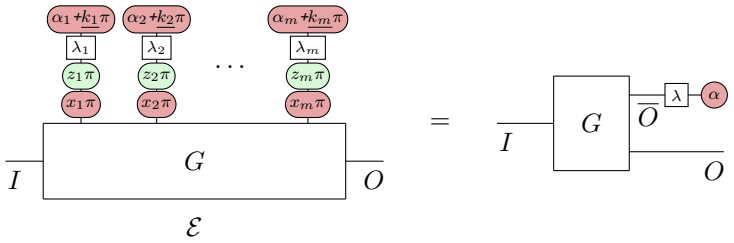
$$\prod_i^{<} (X^{s_i}_{g(i) \cap \{j | i<j\}} Z^{s_i}_{\mathtt{Odd}(g(i)) \cap \{j | i<j\}} M_i^{\lambda_i, \alpha_i, \underline{s}_i}) E_G N_{\overline{I}}$$

where $\prod^{<}$ denotes concatenation in the order $<$, is runnable, uniformly, strongly and stepwise deterministic and realises the target linear map $T(\mathcal{M})$, which is guaranteed to be an isometry.

The same result holds for Pauli flow where the measurement planes λ_i can take the values $\{\mathrm{X}, \mathrm{Y}, \mathrm{Z}\}$. However, a converse version of this theorem only holds for gflow, see [13]. The theorem indicates that X corrections will be performed in $g(v) \setminus \{v\}$ and Z corrections in $\mathtt{Odd}(g(v)) \setminus \{v\}$, for any qubit $v \in \overline{O}$. Moreover, any qubit circuit can be turned into a labelled open graph satisfying the gflow conditions, which ensures that MBQC can perform universal quantum computation [5].

The above result can be expressed succinctly as an equation in **Channel**(**ZX**).

Corollary 3.19. *If a labelled open graph (G, I, O, λ) has flow then, for any choice of measurement angles α, the following equality holds in the CQ interpretation:*



where $\mathcal{E}$ is the set of feedforward equations $x_i = \oplus_{j<i,\, i \in g(j)} \underline{k_j}$ and $z_i = \oplus_{j<i,\, i \in \mathtt{Odd}(g(j))} \underline{k_j}$.

4 Stream construction

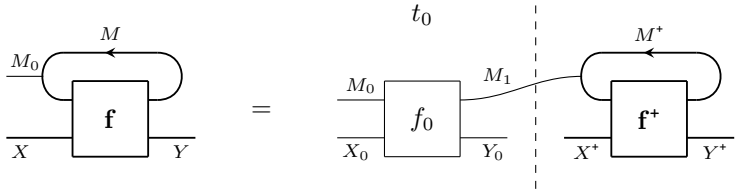
Currently available optical setups are built using photon sources, optical routers and delay lines. In order to represent these additional components, we need to add a time dimension to our diagrams, and allow classical and quantum feedback loops connecting different time-steps.

The stream processes introduced in this section can be seen as a generalisation of the notion of *quantum comb* [18, 19] to a recursive setting with infinitely many inputs and outputs. They enable formal descriptions of lattices, quantum optical setups and circuits studied in the context of quantum convolutional codes [81, 106]. Our definition is based on the concept of intensional monoidal stream of [33, 60]. The notion of observational equality for quantum protocols has already been studied in [15], but we here extend it with rewriting rules allowing us to reason by *induction*. This is done while avoiding the regularity condition of [15, Section E] where processes are assumed to become constant after a finite number of time-steps. Bridging theory and practice, we give multiple examples throughout the section of how this language can be used to reason about real-world experimental setups.

4.1 Recursive definition of stream processes

We define a stream process recursively by what it does at time step zero, together with a stream describing what it does at future time steps. We use letters X, Y to denote infinite sequences $(X_0, X_1, X_2, \dots)$ of objects in a base graphical language $\mathbf{C}$, and define $X \otimes Y$ as the sequence $(X_0 \otimes Y_0, X_1 \otimes Y_1, X_2 \otimes Y_2, \dots)$. We use X^+ to denote the sequence obtained from X by removing the head and by ∂X the sequence $(I, X_0, X_1, \dots)$ obtained by adding the monoidal unit I to X as the head. For M_0 an object of $\mathbf{C}$, we denote by $M_0 \cdot X$ the sequence $(M_0 \otimes X_0, X_1, X_2, \dots)$.

Definition 4.1 (Stream)**.** An intensional stream $\mathbf{f} : X \to Y$ in $\mathtt{Stream}(\mathbf{C})$ with 'initial memory' M_0 is a process $f_0 : M_0 \otimes X_0 \to M_1 \otimes Y_0$ in $\mathbf{C}$ (called 'now') and a stream $\mathbf{f}^+ : X^+ \to Y^+$ with initial memory M_1 (called 'later').



The wire labelled M_0 carries the initial state of the memory, X_0 and Y_0 are the input and output at time-step 0, and M_1 is the memory created at time-step 0 which serves as the initial memory for the stream $\mathbf{f}^+$.

The recursive definition above defines the set of intensional streams with an initial memory as the final fixpoint [58] of the following equation:

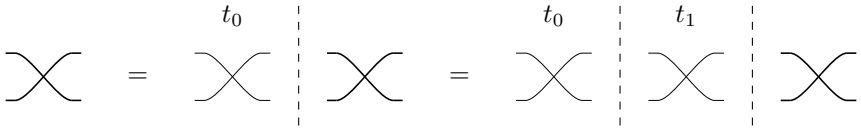
$$\texttt{Stream}(\mathbf{C})(M_0 \cdot X, Y) = \sum_{M_1 \in \mathbf{C}} \mathbf{C}(M_0 \otimes X_0, M_1 \otimes Y_0) \times \texttt{Stream}(\mathbf{C})(M_1 \cdot X^+, Y^+)$$

where $\sum$ denotes the disjoint union and $\times$ the cartesian product of sets. As shown in [33], intensional streams are fully specified by their action at every time step and we have:

$$\mathbf{Stream}(\mathbf{C})(X, Y) \simeq \sum_{M \in \mathbf{C}^{\mathbb{N}}} \prod_{i \in \mathbb{N}} \mathbf{C}(M_i \otimes X_i, M_{i+1} \otimes Y_i)$$

We say that two streams are *intensionally equal* — denoted by the equal sign $=$ — if they have the same action at every time-step and are thus equal in the above set.

The simplest class of streams are *constant streams* with no memory: given any morphism $f : x \to y$ in $\mathbf{C}$ we obtain a stream $\mathbf{f} : X \to Y$ between constant objects $X = (x, x, \dots)$ and $Y = (y, y, \dots)$, with empty memory $M = I$, with $f_0 = f$ and $\mathbf{f}^+ = \mathbf{f}$. We denote the constant stream induced by a diagram f simply by thickening its wires. For example, the following constant stream defines the swap between any two constant objects:

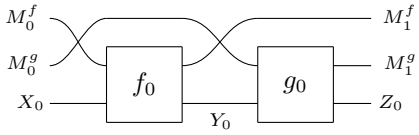


The equation above is read as a recursive definition: the swap stream is the 'swap' now and itself later. More generally we may consider the class of *memoryless streams* where the memory type is the unit of the tensor $(I, I, \dots)$, corresponding to sequences $\{ f_t : x_t \to y_t \}_{t \in \mathbb{N}}$ in the base category. For example, we can now define the swap operation between any two objects $X = (X_0, X_1, \dots)$ and $Y = (Y_0, Y_1, \dots)$ as the sequence $\{ \texttt{swap}_t : X_t \otimes Y_t \to Y_t \otimes X_t \}_{t \in \mathbb{N}}$.

Streams can be composed in sequence or in parallel, forming a symmetric premonoidal category.

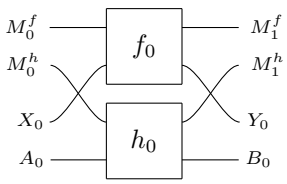
Proposition 4.2. *Intensional streams over a base symmetric (pre)monoidal category* $\mathbf{C}$ *form a symmetric (pre)monoidal category* $\texttt{Stream}(\mathbf{C})$ *where, for streams* $\mathbf{f} : X \to Y$, $\mathbf{g} : Y \to Z$ *and* $\mathbf{h} : A \to B$ *with initial memory types* M_0^f, M_0^g *and* M_0^h*:*

- *the* sequential composition $\mathbf{g} \circ \mathbf{f} : X \to Z$ *is the stream with initial memory* $M_0^f \otimes M_0^g$*, acting* now *as:*

  and later *as* $(\mathbf{g} \circ \mathbf{f})^+ \coloneqq \mathbf{g}^+ \circ \mathbf{f}^+$ *with initial memory* $M_1^f \otimes M_1^g$.

- *the* parallel composition $\mathbf{f} \otimes \mathbf{h} : X \otimes A \to Y \otimes B$ *is the stream with initial memory* $M_0^f \otimes M_0^h$*, acting* now *as:*

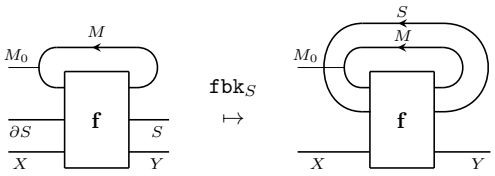
  

  and later *as* $(\mathbf{f} \otimes \mathbf{h})^+ \coloneqq \mathbf{f}^+ \otimes \mathbf{h}^+$ *with initial memory* $M_1^f \otimes M_1^h$.

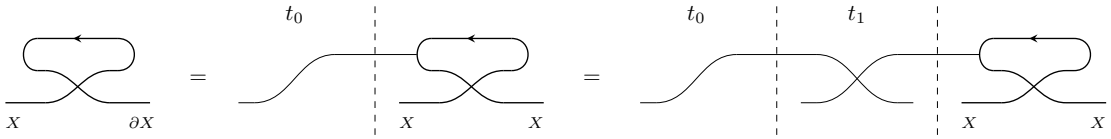
Proof. The identity stream $X \to X$ is simply the memoryless stream $\{ \texttt{id}_{X_t} : X_t \to X_t \}_{t \in \mathbb{N}}$, and the symmetry is given by the swap stream defined above. It was shown in [60, Section 7.1] that the above composition is associative and unital with respect to the identity stream, and that the

above parallel composition is functorial in the monoidal case, i.e. it satisfies the interchange law $(f \circ g) \otimes (h \circ k) = (f \otimes h) \circ (g \otimes k)$. These proofs are stated for a quotient of intensional streams, but the quotient condition is never used in the proofs. As shown in [11, Theorem IV.9], the same results hold for premonoidal categories where the interchanger law is replaced by functoriality of 'whiskering', i.e. $(\mathtt{id} \otimes f) \circ (\mathtt{id} \otimes g) = \mathtt{id} \otimes (f \circ g)$ and similarly for tensor product with $\mathtt{id}$ on the right. □

In order to link different time steps and model feedback of information, we need streams with a memory. We can obtain these by taking the *feedback* $\mathtt{fbk}_S(\mathbf{f}) : X \to Y$ of a memoryless stream $\mathbf{f} : \partial S \otimes X \to S \otimes Y$. This corresponds to adding S to the memory of the stream by feeding back its output values to the inputs, as shown below.



We can use this to model *delay*. For example, the delay of length 1 is defined as $\mathtt{delay}_X = \mathtt{fbk}_X(\mathtt{swap}) : X \to \partial X$, the feedback of $\mathtt{swap} : \partial X \otimes X \to X \otimes \partial X$:



Streams represent infinite processes, but it is often useful to consider their execution for a finite number of time steps. This is done by *unrolling* the stream, giving us a family of functions $\mathtt{unroll}_n : \mathtt{Stream}(\mathbf{C}) \to \mathbf{C}$ parametrised by a natural number.

Definition 4.3 (Unrolling)**.** Given a stream $\mathbf{f} : X \to Y$ with memory M, the unrolling for n time-steps of $\mathbf{f}$ is a process in $\mathbf{C}$ of the form:

$$\mathtt{unroll}_n(\mathbf{f}) \colon M_0 \otimes X_0 \otimes \ldots X_n \to Y_0 \otimes \ldots Y_n \otimes M_{n+1}$$

defined by induction as follows:

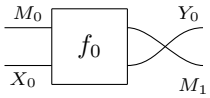
$$\mathtt{unroll}_1(\mathbf{f}) = \mathtt{swap}_{M_1,Y_0} \circ f_0$$

$$\mathtt{unroll}_n(\mathbf{f}) = (\mathtt{id}_{Y_0} \otimes \mathtt{unroll}_{n-1}(\mathbf{f}^{\dagger})) \circ (\mathtt{unroll}_1(\mathbf{f}) \otimes \mathtt{id}_Z)$$

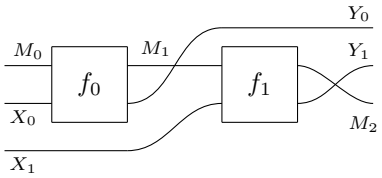
where $Z = X_1 \otimes X_2 \otimes \cdots \otimes X_{n-1}$.

For example, below are the three first unrollings of a generic stream.

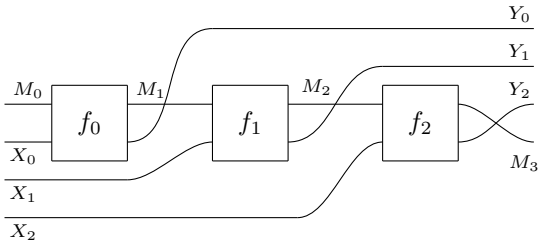
$\mathtt{unroll}_1(\mathbf{f})$ =



$\mathtt{unroll}_2(\mathbf{f})$ =

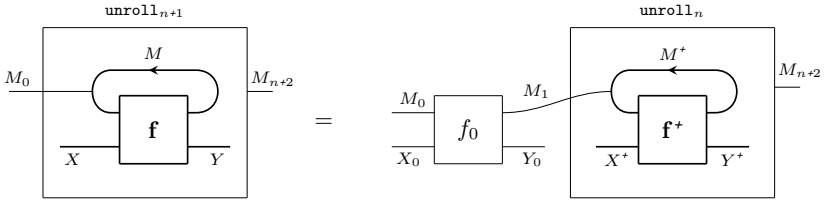


$\texttt{unroll}_3(\mathbf{f})$ =

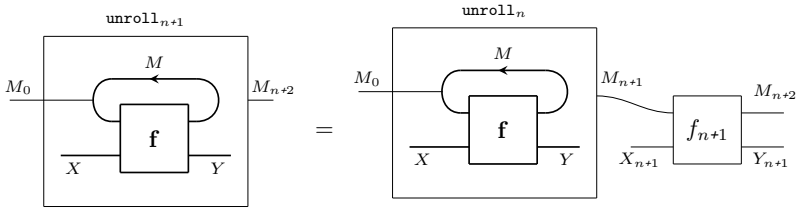


We represent the unrolling graphically with a box surrounding the stream. The following rules are the graphical equivalent of Definition 4.3, and allow us to reason inductively about finite protocol executions.

Rewrite 6.



Rewrite 7.



Note that these rules avoid the bureaucracy of ordering input and output wires which could be made more formal using the concept of 'open diagrams' [94].

In streams we have access to a 'followed by' operation, common in dataflow programming languages [44]. Given a process $r : M_0' \to M_0$ in $\mathbf{C}$ and a stream $\mathbf{f} : X \to Y$ with initial memory M_0, the stream $\texttt{fby}(r, \mathbf{f})$ is given by $f_0 \circ (r \otimes \texttt{id}_{X_0})$ 'now' and $\mathbf{f}^+$ 'later'. It satisfies the following rule.

Rewrite 8. *For any $r : M_0' \to M_0$ and $\mathbf{f} : X \to Y$ with initial memory M_0,*

$$\texttt{unroll}_n(\texttt{fby}(r, \mathbf{f})) = \texttt{unroll}_n(\texttt{fby}_{M_1}(\texttt{id}_{M_1}, \mathbf{f})) \circ (r \otimes \texttt{id}_{X_0})$$

This allows us to slide processes acting on the input memory outside of the unroll box, as in the following example.

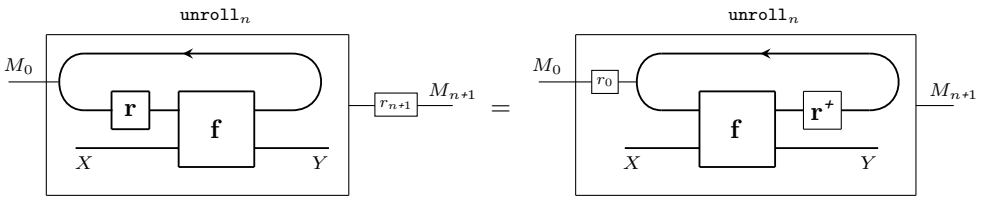
Example 4.4. Let $P : M \to M \in \mathbf{C}$ be a process such that $P^2 = \texttt{id}_M$, then the streams $\mathbf{A} = \texttt{fbk}_M(\mathbf{P})$ and $\mathbf{B} = \texttt{fby}(P, \texttt{fbk}_M(\mathbf{I}))$ are intensionally different but $\texttt{unroll}_n(\mathbf{A}) = \texttt{unroll}_n(\mathbf{B})$ for all n. Indeed at time-step 0 both streams act as P, then by induction we have:

$$\begin{aligned}\texttt{unroll}_{n+1}(\mathbf{P}) &= \texttt{unroll}_n(\mathbf{P}) \circ P = \texttt{unroll}_n(\mathbf{Q}) \circ P \\ &= \texttt{unroll}_n(\texttt{fby}(P, \texttt{fbk}_M(\mathbf{id}))) \circ P = \texttt{unroll}_n(\texttt{fbk}_M(\mathbf{id})) \circ P^2 \\ &= \texttt{unroll}_n(\mathbf{I}) \circ P = \texttt{unroll}_{n+1}(\mathbf{Q})\end{aligned}$$

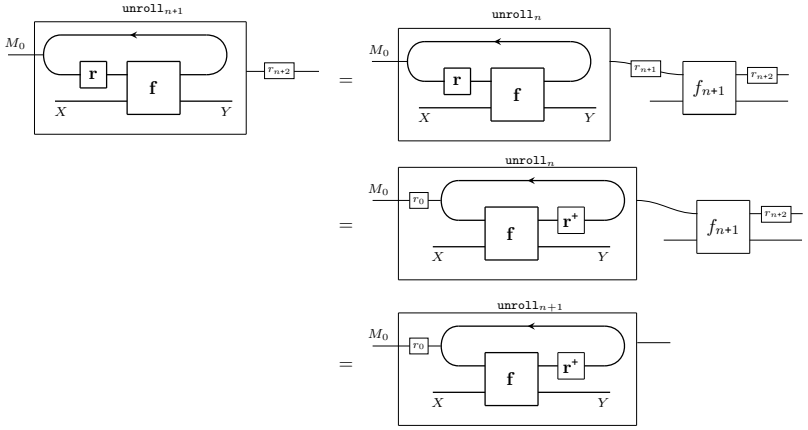
using Rewrite 6 and Rewrite 8.

Finally, we show the following using induction with Rewrite 7.

Lemma 4.5. *For any pair of sequences $r_t : M_t \to M_{t+1}$ and $f_t : M_{t+1} \otimes X_t \to M_{t+1} \otimes Y_t$, the following holds in* $\mathbf{C}$*:*



Proof. The statement for $n = 0$ is easy to show, then we proceed by induction as follows:



where the first step is Rewrite 7, the second step is induction and the last step uses both Rewrite 8 (twice) and Rewrite 7. □

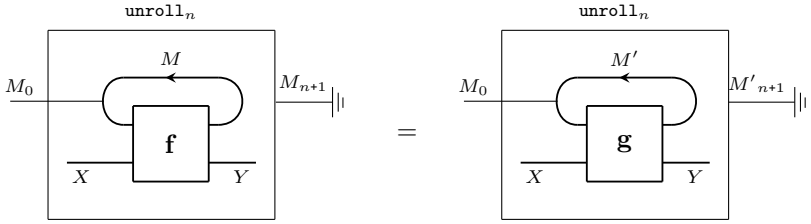
4.2 Interpretation as quantum protocols

Constructing streams over our base language of quantum channels, we have access to the discarding map on every wire, as well as preparation and measurement commands taking values $x_t, \underline{y_t}$ at each time step t. The feedforward of classical information is again left implicit: the output variables at time step t are input (and output) variables for time-step $t+1$. This allows us to build annotations relating variables across different time-steps, while interpreting the unrolling direction as time.

Definition 4.6 (Quantum protocol)**.** A quantum protocol $\mathbf{f} : X \to Y \in \texttt{Stream}(\texttt{Channel}(\mathbf{C}))$ is stream of Kraus maps with a global set $\mathcal{E}$ of equations of the form of Definition 3.2 on the set of input and output variables $x_t, \underline{y_t}$, causally compatible with the order induced by $\underline{y_t} < \underline{y_{t+1}}$.

The discarding maps together with the unrolling operation allow us to define the notion of *observational* equality from [15] for general quantum protocols.

Definition 4.7 (Observational equality)**.** We say that two quantum protocols $\mathbf{f}$ and $\mathbf{g}$ with the same initial memory M_0 are observationally equivalent if for any $n \in \mathbb{N}$:

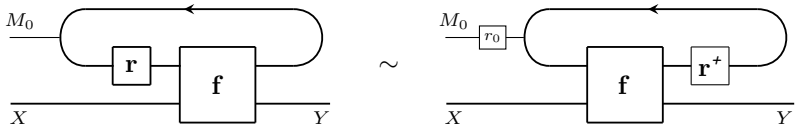


where M_n and M'_n denote the memory types of $\mathbf{f}$ and $\mathbf{g}$, respectively, at time step n.

We use the symbol $\sim$ between streams for observational equivalence, and reserve the symbol $=$ for intensional equality. If two intensional streams are equal, then they are observationally

equivalent; however, the converse isn't true in general. The following *sliding rule*, for example, equates streams with different memories. A proof is obtained by Lemma 4.5 and using the fact that isometries commute with discards.

Rewrite 9. *For any sequence of isometries* $r_t : M_t \to M_{t+1}$ *and channels* $f_t : M_{t+1} \otimes X_t \to M_{t+1} \otimes Y_t$*, the following streams are observationally equal:*



Remark 4.8. Similar sliding equations restricted to *causal* maps r_t appear in [11, 15]. An important consequence of the r_t being isometries is that the only scalar that can slide between different time steps is 1, which ensures that every finite execution of the protocol has a well defined probability.

We can now define causality and determinism for quantum protocols through their observational interpretation.

Definition 4.9 (Causality)**.** We say that a quantum protocol $\mathbf{f}$ is *causal* if $\texttt{unroll}_n(\mathbf{f})$ is causal for any $n \in \mathbb{N}$.

Definition 4.10 (Determinism)**.** We say that a quantum protocol $\mathbf{f}$ is *deterministic* if every unrolling of $\mathbf{f}$ is deterministic in $\texttt{Channel}(\mathbf{C})$. We say that it is uniformly deterministic if it is deterministic for every choice of sequence of angle parameters α_t.

The mixed sum + of channels induces an infinite branching behaviour at the level of streams. We can however prove properties of this behaviour by induction over the unrolling, using the following rule.

Rewrite 10. *If* $\mathbf{f} = \texttt{fby}_M(A + B, \mathbf{g})$ *then* $\texttt{unroll}_n(\mathbf{f}) = A\,\texttt{unroll}_n(\mathbf{g}) + B\,\texttt{unroll}_n(\mathbf{g})$

When branching occurs, we are interested in the probability of implementing a target channel D in some time n, this motivates the following definition.

Definition 4.11 (Probabilistic simulation)**.** We say that a quantum protocol $\mathbf{f}$ implements a channel $D \in \texttt{Channel}(\mathbf{C})$ in time n with probability p, if $\texttt{unroll}_n(\mathbf{f})$ implements D with probability p (see Definition 3.8). We say that $\mathbf{f}$ simulates a quantum protocol $\mathbf{g}$ with time-dependent probability $p(n)$ if $\texttt{unroll}_n(\mathbf{f})$ implements $\texttt{unroll}_n(\mathbf{g})$ with probability $p(n)$ for all $n \in \mathbb{N}$.

Universality then refers to the ability of a stream to implement any circuit in some set within a fixed probability. We give two alternative definitions of universality. The strong notion requires that any circuit in the set can be implemented with probability arbitrarily close to 1, while the weak notion allows for a constant probability of successful implementation, which may be more suited for intermediate-scale architectures.

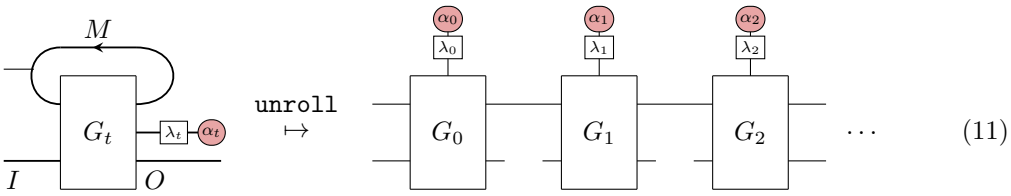
Definition 4.12 (Strong universality)**.** We say that a parametrised quantum protocol $\mathbf{f}(\alpha)$ is strongly universal for a family of qubit circuits $\{ C_i \}$ if for any circuit C_i and tolerance $\epsilon \in (0, 1)$ there exists an integer n and parameter values α_i such that $\mathbf{f}(\alpha_i)$ implements C_i in time n with probability $p > 1 - \epsilon$.

Definition 4.13 (Weak universality)**.** We say that a parametrised quantum protocol $\mathbf{f}(\alpha)$ is weakly universal for a family of qubit circuits $\{ C_i \}$ if there exists an ϵ such that for any circuit C_i there exists an integer n and parameter values α_i such that $\mathbf{f}(\alpha_i)$ implements C_i in time n with probability $p > 1 - \epsilon$.

We give examples of these two types of universality in Theorem 4.15 and in Section 7.

4.3 Reasoning with streams of ZX diagrams

Since any ZX diagram can be put in MBQC form, a general process in $\texttt{Stream}(\mathbf{ZX})$ has the following form:



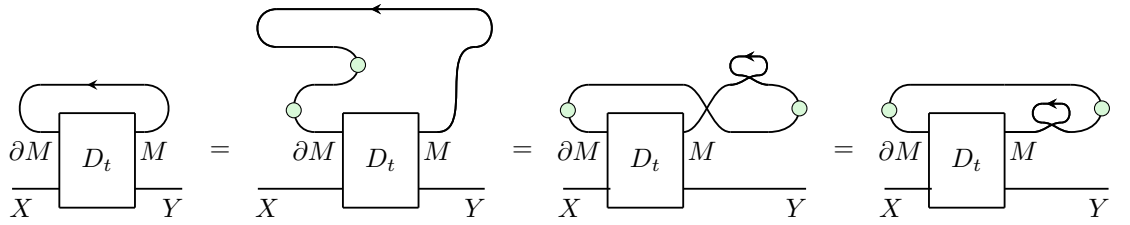
(11)

for some sequence of open graphs $(G_t, I_t + M_t, O_t + M_{t+1})$ with measurement planes λ_t and angles α_t. Streams of ZX diagrams can thus be seen as *foliations* of an infinite (open) graph state with specified measurement planes and angles on internal nodes.

A key difference between $\texttt{Stream}(\mathbf{LO})$ and $\texttt{Stream}(\mathbf{ZX})$ is that the latter is *compact closed*, that is, it has Bell states and Bell effects on every object X. These are induced from **ZX** by building the memoryless streams $\texttt{cup}_X = \{\, \texttt{cup}_{X_t} : X_t \otimes X_t \to I \,\}_{t\in\mathbb{N}}$ and $\texttt{cap}_X = \{\, \texttt{cap}_{X_t} : I \to X_t \otimes X_t \,\}_{t\in\mathbb{N}}$. The snake equation lifts to $\texttt{Stream}(\mathbf{ZX})$ as it holds at every time-step. It allows us to relate feedback loops with delays.

Proposition 4.14. *[32] Any stream in* $\texttt{Stream}(\mathbf{ZX})$ *can be written as a composition of memoryless streams and the delay.*

Proof. This follows from the snake equation and the interchange law which holds by Proposition 4.2 since **ZX** is a symmetric monoidal category:

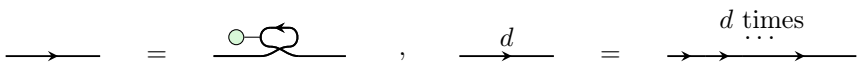


□

As a consequence, the slogan of the ZX calculus remains true after applying $\texttt{Stream}$:

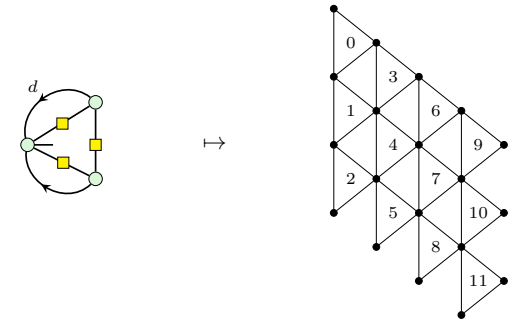
‘Only connectivity matters.’

To see this, consider the sequence in Equation (11) with a constant ‘generating graph’ $G_t = G$. The resulting infinite graph has a periodic lattice structure, as illustrated in Figure 1. In the figure, we use the following notation for delays on a single qubit initialised with $|+\rangle$ states:

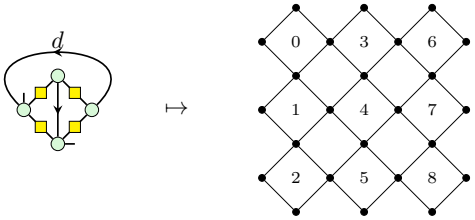


and one can verify that only the connectivity of these delays to spiders in the diagram determines the generated infinite structure.

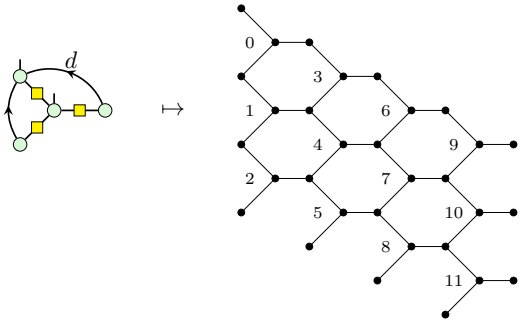
Streams of ZX diagrams enable finite representations of infinite graph states and allow us to prove equivalences between these infinite objects by graphical rewrite rules. To illustrate the power of our approach, we now prove that the honeycomb lattice with only YZ measurements is universal for qubit circuits. Indeed, the triangular lattice is known to be universal for qubit circuits using only XZ measurements [72], and we show that the honeycomb lattice with periodic Y measurements simulates the triangular lattice. While this was already observed in [78], our one-line formal proof requires only local rewrites on a finite diagram (instead of an argument on infinite graphs with ‘. . . ’). It moreover allows us to compute the planar measurements required to simulate the XZ measurements on the triangular lattice, and we find that YZ measurements suffice.



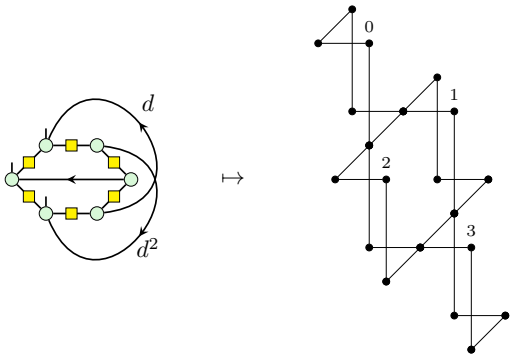
Foliation of a triangular lattice on a cylinder, obtained by $\texttt{unroll}_{11}$ with $d = 3$.



Foliation of a rectangular lattice on a cylinder, obtained by $\texttt{unroll}_{8}$ with $d = 3$.



Foliation of a honeycomb lattice on a cylinder, obtained by $\texttt{unroll}_{11}$ with $d = 3$.

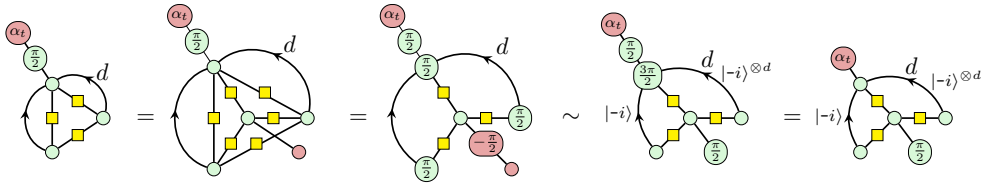


Foliation of a Raussendorf lattice on a duocylinder, obtained by $\texttt{unroll}_{3}$ with $d = 2$.

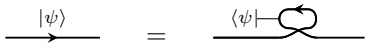
Figure 1: Lattice foliations as streams of ZX diagrams. The labels indicate the time step at which the recurring part of the graph is generated. The unrolled graph states are depicted on the plane but they are actually embedded in cylinders, since the presence of a delay of 1 implies that time step t is always connected to time step $t + 1$.

Theorem 4.15. *The hexagonal lattice with only* YZ *measurements is strongly universal for qubit circuits. More precisely, the honeycomb lattice with* YZ *measurements can simulate a triangular lattice with* XZ *measurements.*

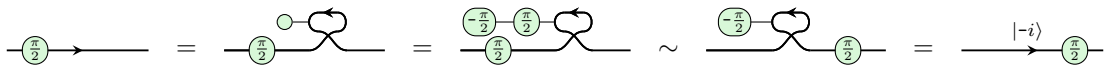
Proof. Figure 1 gives the ZX representation of triangular and hexagonal lattices. We only use two flow-preserving ZX rules: Z insertion and local complementation. These hold for constant streams which means we can apply them in $\texttt{Stream}(\mathbf{ZX})$ as long as they don't act on delayed lines. We thus obtain the following proof:



where we used the notation for delays initialised differently with a single qubit state $|\psi\rangle$:



and the observational equality $\sim$ follows from Rewrite 9:

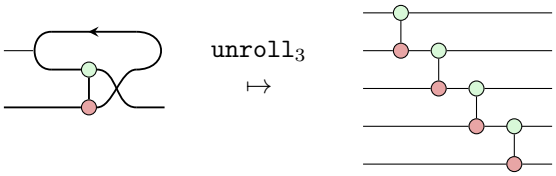


These local Clifford operations result only in measurements in the YZ plane. Note that both lattices can be flattened to the plane by applying Z measurements (which are both in the XZ and

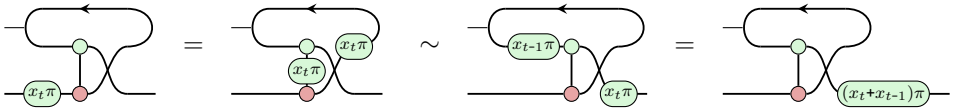
YZ measurement planes). It was shown in [72, Theorem 17], that a faultless triangular lattice of size $4n \cdot 4k$ is sufficient to implement any qubit circuit of width n and depth k, generated by CZ, Hadamard and Z-phase gates. For fixed d, we thus obtain strong universality in the sense of Definition 4.12 for circuits of fixed width or depth. Since the rewrite above holds for any d, we conclude that the (infinite, faultless) hexagonal lattice is universal for arbitrary qubit circuits. □

As a second application of our calculus, we can use the inductive rules and sliding to compute the commutation relations between ZX streams and infinite Pauli sequences. To illustrate this, we study a classical example from the literature on convolutional codes [81, 106].

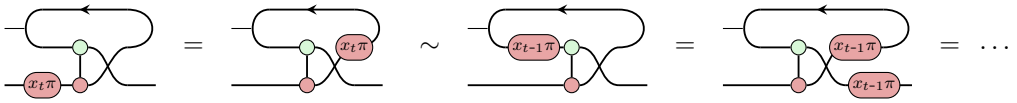
Example 4.16 (Infinite-depth CNOT gate)**.** Consider the ladder of CNOT gates defined by the following stream:



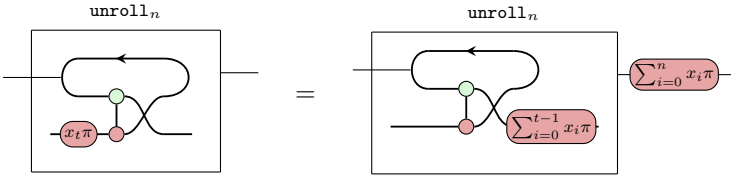
Given any sequence of Pauli Z gates, we have the following commutation relation, using Rewrite 9:



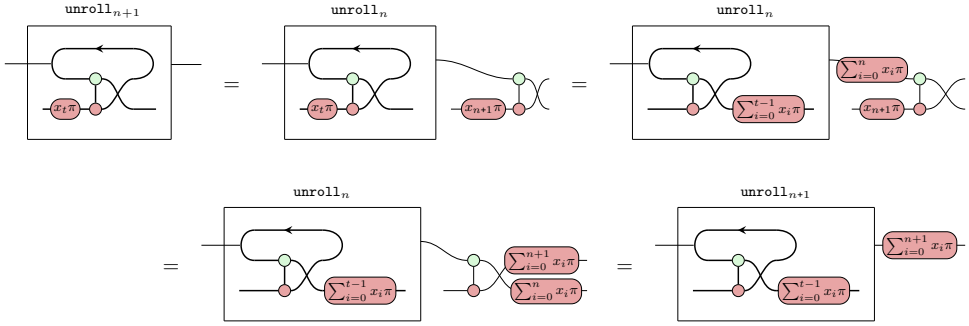
However, if we try to apply the same strategy to commute a sequence of Pauli X gates, we get stuck in a recursive loop:



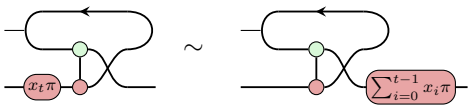
In fact, the Pauli X gates accumulate on the output memory. We can show the following equation about any finite unrolling of the stream:



This is done by induction as follows:



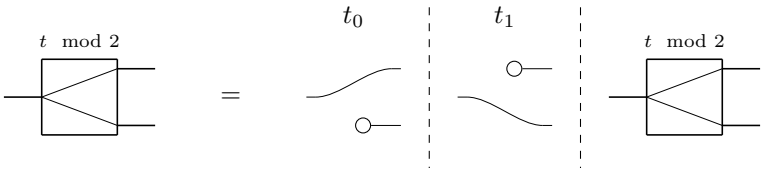
After post-composing both sides with the discard map, we obtain the following observational equality:



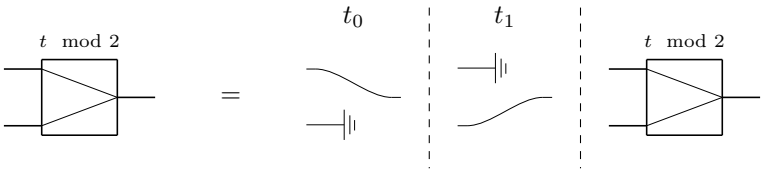
4.4 Routing and measurement modules

Applying the `Stream` construction to linear optical circuit in `Channel`(**LO**) we can now construct linear optical protocols acting on photon time-bins, containing delays and feedback loops. These components are used throughout optical computing, see e.g. [6, 16, 76].

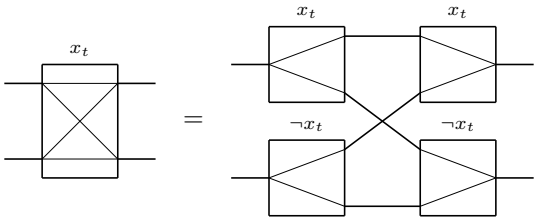
The ability to prepare an empty optical mode gives rise to a useful class of memoryless processes in `Stream`(**LO**) called *routers*. For example, the binary oscillating router is defined by:



And a similar 2 to 1 router is obtained using the discard map:



We assume that we have access to routers that can be controlled by a stream of classical variables x_t with a predefined value at every time-step. We may construct arbitrary routers from the binary router. For example, the following setup implements an identity if $x_t = 0$ and a swap if $x_t = 1$.

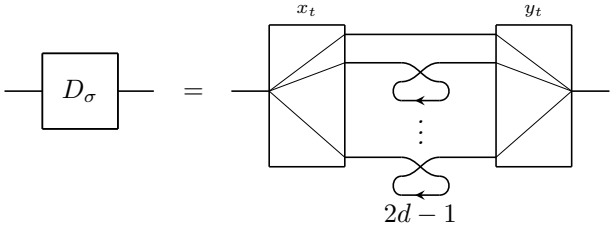


Remark 4.17. Routers with more outputs than inputs can only be defined on *optical modes* in `Stream`(**ZXLO**). This is because the qubit space $\mathbb{C}^2$ does not allow for the empty state.

Remark 4.18. We distinguish between routers and switches, defined in Equation (10). The control parameters of a router every time step are set before executing the program. With switches, the routing can be actively controlled by a measurement outcome or a classical variable computed at run-time. Such actively controlled switches are denoted as routers but with the control parameter drawn inside the box, as in Equation (13).

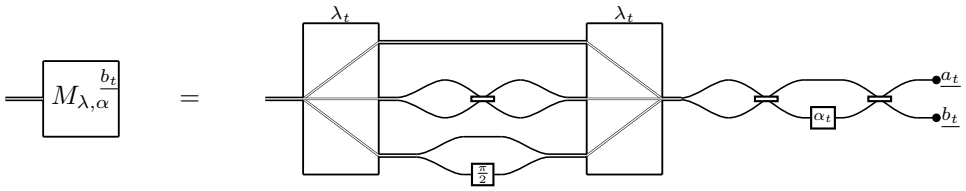
By composing delays and routers, we can route arbitrary permutations in time encoding.

Lemma 4.19. *The following setup implements any permutation σ of length d in time encoding:*



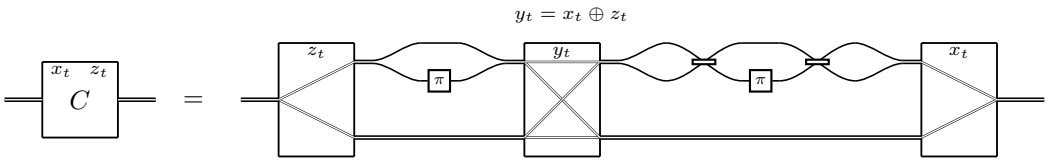
Proof. Given a permutation σ, we set $x_t = t + \sigma(t) \mod 2d$ and $y_t = d + t \mod 2d$. A more efficient protocol for routing arbitrary permutation of size d with $\log(d)$ binary routers is given in [6]. □

Working in $\texttt{Stream}(\texttt{Channel}(\mathbf{LO}))$, we are now ready to introduce the basic optical components used to perform universal MBQC: a measurement module and an active correction module. For the measurement module, we need to be able to measure in the XY, XZ, or YZ planes at different time-steps, typically fixed before running the experiment. Given a choice of measurement plane λ_t and angle α_t for each time-step t, the measurement module $M_{\lambda,\alpha}$ is given by the following setup:



(12)

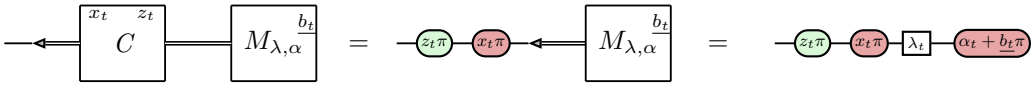
Finally, any sequence of Pauli corrections can be implemented using switches with classical control parameters $x_t, z_t \in \{0, 1\}$, as follows:



(13)

Combining streams of linear optics and ZX diagrams, we can reason about optical protocols used in photonic quantum computing. For example, the measurement and correction modules defined above with linear optical components, give rise to the expected streams of ZX diagrams.

Lemma 4.20.



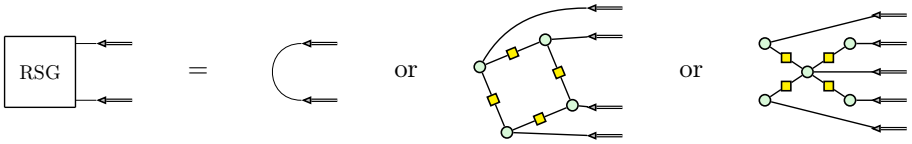
Proof. Since both the correction and measurement modules are memoryless streams, it is sufficient to prove that the equation above holds for any given time step t. In each case, the routers and switches define a specific path and the result follows from the equations of Section 2.4. □

4.5 Resource state generators

We now discuss different methods for the generation of photonic resource states (RSG) and how they can be modeled as streams of $\texttt{Channel}(\mathbf{ZX})$ diagrams. These can be broadly assigned to two classes — (i) photonic and (ii) matter-based methods — and we give two examples of the latter. These different procedures can in principle be used in conjunction.

Photonic RSG Linear optical methods begin with single photons, which are typically generated by spontaneous parametric down-conversion [45]. These photons are then entangled using linear optical Bell measurements or other heralded linear optical circuits [7]. The advantage of this approach is that the resource states can in principle have arbitrary connectivity as photons are not spatially or temporally restricted. Moreover, photons of different resource states can be prepared with low distinguishability by active alignment of sources [16]. Examples of photonic graph states used in the literature include the star graph [41, 61], rings [8] and complete-like graphs [4]. The main drawback of these approaches is that, because of the fundamental limits of linear optics [99], entanglement can only be generated probabilistically. This drawback can be mitigated by using ancillary photons [7] and 'switch networks' [6] to boost probabilities of success [38].

Assuming deterministic generation, we model photonic resource state generators of a graph state $|G\rangle$ as the constant stream over the underlying graph G expressed as a ZX diagram:

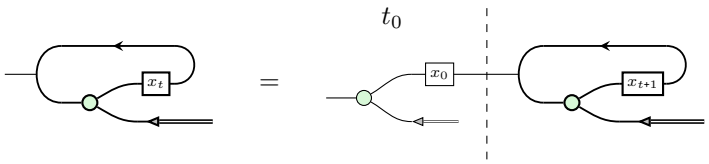


(14)

where we omitted the normalisation scalar at every time step given by a constant stream $\mathbf{s}$ satisfying $\texttt{unroll}_n(\mathbf{s}) = s^n$.

Remark 4.21. Streams are *synchronous* processes with a predefined global clock rate. It is sometimes useful to define streams acting at different clock rates, such as integer multiples of the global clock rate. A possible application would be to formalise the switch networks of [6] where multiple low-fidelity heralded resource states are actively routed to produce higher fidelity resource states in each time-bin.

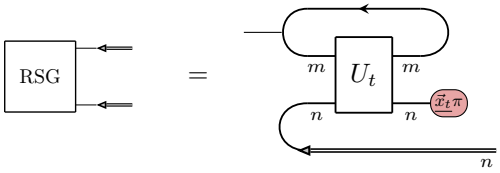
Emitter-based RSG Matter-based approaches for resource state generation rely on the emission of photons by excitation of a trapped ion [9] or an artificial atom [37, 54, 77, 96]. Spin-based emitters, such as quantum dots, are able to keep the atom in a coherent superposition during the emission process. They produce a stream of photons entangled with the atom and with each other [101]. As such, they may be considered 'photonic machine guns' [65], and represented accordingly:



(15)

The atom is the memory of the stream, entangled via a Z spider to the dual-rail states of the emitted photons. At each time-step t we may perform a single qubit unitary x_t on the atom. Special cases of interest are GHZ state generators when the white box is the identity, linear clusters when it is a Hadamard gate and variable GHZ-linear clusters, or caterpillar states [50], when it can be programmed arbitrarily. This technology has proved particularly effective for the generation of entangled photonic graph states [24, 25, 97], and it has the advantage that resource states can be generated deterministically [24, 97]. Nevertheless, as photons are emitted one at a time, the resulting entanglement is restricted to *linear* structure, and photons need to be demultiplexed making them more susceptible to loss. Photons emitted by non-identical atoms also suffer from distinguishability, although methods for mitigating this are being developed [107]. A great advantage of emitters is that they can be used in repeat-until-success protocols, as shown in Section 7.1, that enables the near-deterministic implementation of entangling gates by fusion measurements [42, 64].

Ion-based RSG In the case of generation of photons from an ion trap, the emission process can be stimulated by a classically controlled laser pulse [68, 80]. One obtains an ion-photon Bell pair, but this destructively measures the previous state of the ion which has to be prepared in a fixed state at each entanglement stage. Ions with this property are more generally called *network qubits*. If we are further able to perform gates between network qubits and other more stable *memory qubits*, we obtain a matter-based resource state generator of the form:



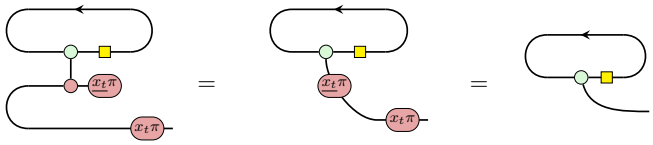
(16)

where U_t is a unitary on $m+n$ qubits, and a k-labeled wire is interpreted as the tensor product of k single wires. A general quantum processing unit (QPU), with both matter and photonic degrees of freedom can be represented as a variable stream with memory given by a tensor product of qubits and with photonic outputs at regular time intervals.

Simulations between RSGs The stream language which we have developed can be used to prove (bi)simulations between these different physical setups.

Proposition 4.22. *The quantum emitter can be simulated deterministically by a matter-based QPU and classical feedforward.*

Proof. We use a QPU with a single memory qubit entangled with a CNOT gate to the network qubit:

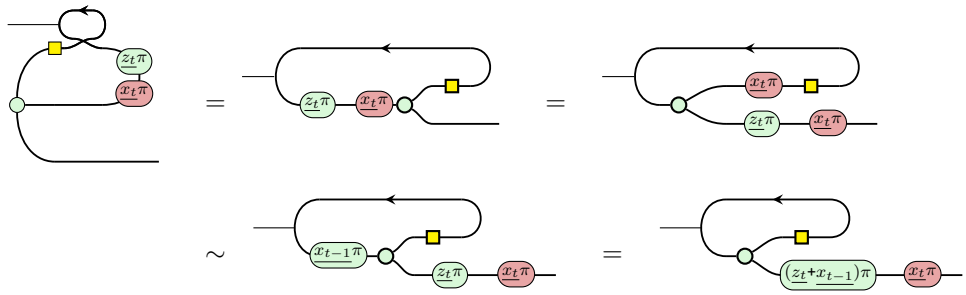


□

In order to simulate these matter-based memories with constant RSGs, we moreover need entangling measurements.

Proposition 4.23. *The quantum emitter can be simulated deterministically by a constant GHZ state, the delay, perfect Bell measurements and classical feedforward.*

Proof. This is shown by the derivation below which computes the feedforward instructions to correct the measurement-induced Pauli byproducts.



The first step follows by Proposition 4.14 and the propagation of Pauli byproducts follows from Rewrite 9. □

Crucially, the above proposition relies on a perfect Bell measurement which is not available in linear optics. In the remainder of this paper we study the kind of entanglement afforded by linear optics and how it can be leveraged for photonic and distributed quantum computing.

Part II
Application to distributed architectures

5 Characterization of correctable fusion measurements

Fusion measurements are linear optical entangling measurements on dual-rail qubits. They were originally introduced by Browne and Rudolf [14], although several variations of this idea are present in the literature [62, 64]. They are at the heart of recent *fusion-based* proposals for performing quantum computation with photons [8, 86], but are also used in *distributed* quantum architectures to create entanglement between remote processors [68]. Beyond 'Type I' and 'Type II fusion' [14], linear optical circuits allow to construct different types of entangling gates on dual-rail qubits [41, 42, 64].

In this section, we exploit the graphical language developed in Part I to obtain ZX representations of fusion measurements. We then generalise this concept by classifying all locally equivalent measurements whose Pauli byproducts *can be corrected*. Our notion of correctability requires that the non-determinism of measurement outcomes can be propagated as heralded Pauli errors in the input qubits. The notion of planar fusion measurements introduced here (in the XY, XZ and YZ planes) allows to perform non-Clifford entangling gates in photonic circuits while reducing the total number of photons (compared to separate Type II fusions and non-Clifford single qubit measurements), see Remark 7.7 for an example. We also use it in Section 7.3 to switch between X and Y measurements with a single parameter shift. The characterization results of this section are previewed in Section 5.

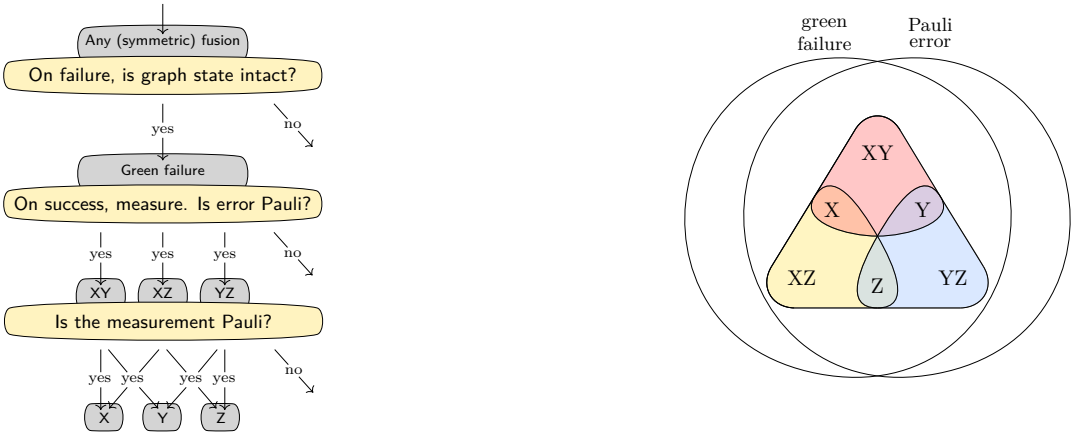


Figure 2: Overview of the results of this section. (a) Flowchart of the principles used to characterize correctability in fusion measurements (b) The zoo of correctable fusion measurements. Type II fusion is an instance of X fusion, CZ fusion is an instance of Y fusion, and two-qubit Z phase gadgets are a special case of YZ fusions.

Remark 5.1. Ref. [66] also analyses the action of generalisations of fusion measurements on graph states. However, the generalisations of fusion they consider are local Clifford equivalent to Bell measurements, and they consider only the success case. Here instead, we consider all measurements local unitarily equivalent to Type I fusion followed by arbitrary single-qubit measurement. In particular, we obtain planar fusions and the CZ fusion of [64] that cannot be expressed in their setting.

5.1 Fusion measurements in ZX

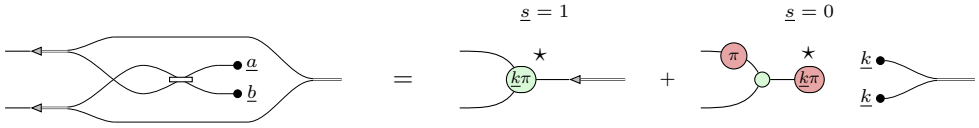
In Ref [14], fusions were expressed with quarter-wave plates and polarizing beam splitters, respectively, as follows:

Note that Type I fusion is a partial measurement having two inputs and one output dual-rail mode, while Type II is destructive and measures both qubits. We can translate from polarization primitives to **LO** circuits using the following equalities:

Expressing the Type I and Type II fusions using this translation, we obtain the following diagrams, respectively:

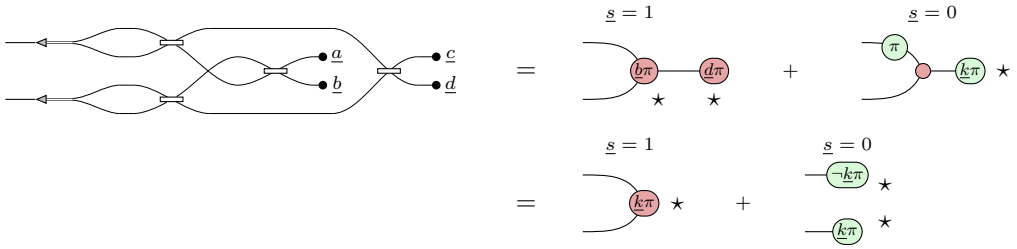
This representation enables us to diagrammatically calculate the action of these circuits by representing them as a mixture of ZX diagrams. Starting with Type I fusion, we get the following Kraus decomposition, proved in Section B.

Proposition 5.2. *The following equation holds in the CP interpretation:*



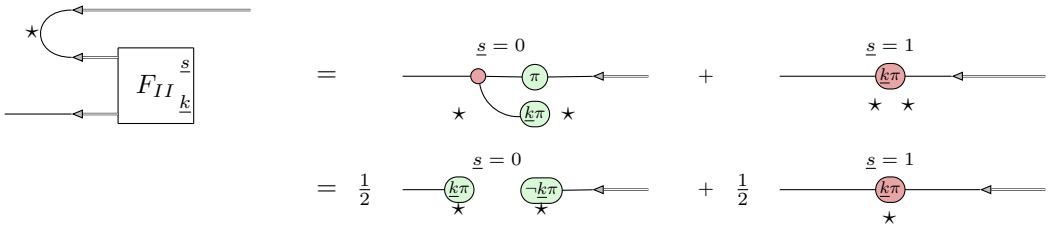
after coarse-graining of the measurement operator by the equations $\underline{s} = \underline{a} \oplus \underline{b}$ *and* $\underline{k} = \underline{s}\underline{b} + \neg\underline{s}(1 - \frac{\underline{a}+\underline{b}}{2})$. *Here,* s *is the Boolean value of success and* k *is the Pauli measurement error.*

This means that the error is $\underline{b}$ in case of success and $1 - \frac{\underline{a}+\underline{b}}{2}$ in the failure case. Note that, in case of failure, the pair of output modes is no longer in the qubit subspace defined in Rewrite 1. Now, considering the Type II fusion, we see that this is just a Type I fusion preceded by beam splitters and followed by a single qubit measurement in the Z-basis. We can thus compute its action on the qubit subspace:



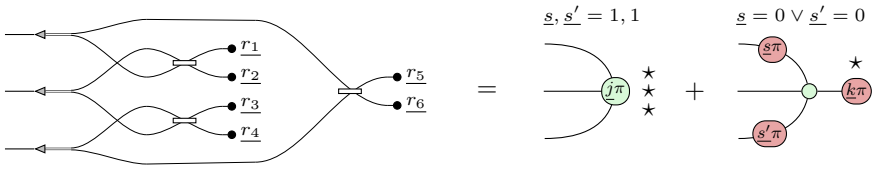
where $\underline{s} = \underline{a} \oplus \underline{b}$ is the Boolean value of success and $\underline{k} = \underline{s}(\underline{b} + \underline{d}) + \neg\underline{s}(1 - \frac{\underline{a}+\underline{b}}{2})$ is the error. This means that the error is $\underline{b} + \underline{d}$ in case of success and $1 - \frac{\underline{a}+\underline{b}}{2}$ in case of failure. The scalars in the diagrams above are crucial for computing probabilities, but they can be disregarded in many cases. We use them in Section D, to compute the probability of success of fusion measurements for different input states. However, in many other cases we can disregard them as in Example 6.4.

As a first application, we consider a teleportation protocol where two parties share a dual-rail encoded photon Bell pair, and one of the parties performs a fusion measurement with a dual-rail encoded input qubit. By Proposition 5.2, the protocol can be rewritten as a distribution over two causal maps:



If we apply a Pauli X correction conditioned on $\underline{k}$ on the output qubit, this results in the identity channel with probability $\frac{1}{2}$, otherwise the input qubit is measured and the opposite state is prepared in the output.

Remark 5.3. Both Type I and Type II fusions can be generalized to arbitrary number of inputs. A natural generalisation of Type II fusion gives the n-GHZ state analyser studied in [12, 85, 86] in the success case. For the 3-GHZ state analyser we obtain:

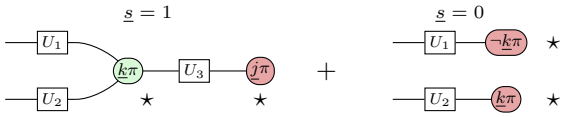


where $\underline{j} = r_2 + r_4 + r_6 \mod 2$, $\underline{s} = r_1 + r_2 \mod 2$, $\underline{s'} = r_3 + r_4 \mod 2$, $\underline{k} = 1 - \frac{r_3+r_4}{2}$ if $\underline{s} = 1$ and $\underline{k} = 1 - \frac{r_1+r_2}{2}$ otherwise.

5.2 General fusion measurements

In Section 5.1, we saw that Type II fusion differs from Type I fusion only by single-qubit unitaries applied before and after the fusion, as well as an additional measurement. Similarly, we can describe

the success and failure outcomes of all possible entangling measurements unitarily equivalent to Type II fusion as follows:



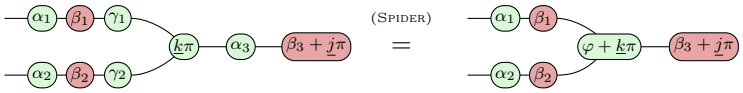
(17)

Here, U_1, U_2, and U_3 are arbitrary single-qubit unitaries which, up to some global phase, can be expressed by the Euler decomposition as three alternating rotations around the Z and X axes:

(18)

Using this decomposition, we get a total of 9 parameters to characterize a general fusion; however, we are able to reduce this number using different observations that we support with calculations in ZX calculus. First, one parameter can be eliminated from U_3 as it is followed by a single-qubit measurement that only contributes an irrelevant global phase:

Second, we observe that $Z(\gamma_1)$, $Z(\gamma_2)$, $Z(\alpha_3)$, and the fusion error $\underline{k}\pi$ itself are simultaneously diagonalizable in the Z basis. In other words, we can apply the spider fusion rule of the ZX calculus as follows:

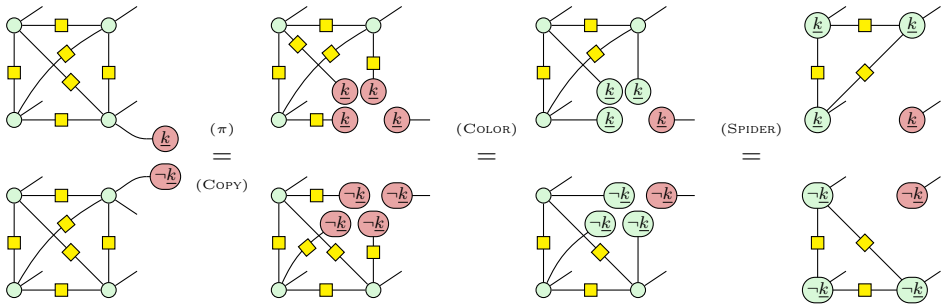


where $\varphi = \gamma_1 + \gamma_2 + \alpha_3$.

With this, we have reduced the number of parameters to describe an entangling measurement to 6. However, by only considering fusion with certain desirable properties, we can reduce this number even further.

5.3 Green failure

Consider the fusion given in Equation (17) the fusion with U_1, U_2, and U_3 all being the identity, corresponding to Type I fusion composed with a Z measurement. An unsuccessful fusion in this case acts as a projector in the Z-basis. This means that in addition to failing to fuse the two nodes, it also *disconnects* them from their neighbours. For example:

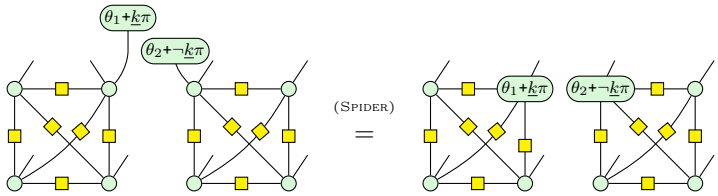


In order to preserve the entanglement of the graph state, we want failures to be 'green'.

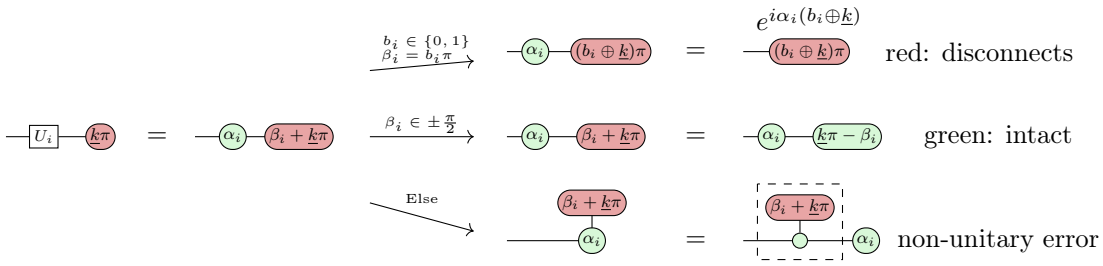
Definition 5.4 (Green failure)**.** We say that a fusion measurement has green failure if its failure outcome satisfies:

for some θ_1, θ_2.

This means that, upon failure, the underlying resource graph preserves its connectivity. For our running example:



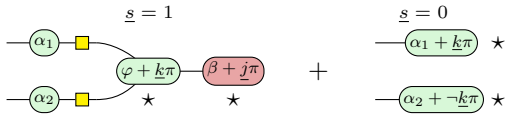
We can characterize all types of fusions up to Z rotations on the nodes by the choices of β_i in the Euler decomposition shown in Equation (18). We have the following measurement outcomes for different choices of β_i, with $i \in \{0, 1\}$,



In other words, the failure is either green and keeps the connection of the graph, red and disconnects the graph, or it induces a non-unitary (and thus not correctable) error on the graph. Asking for green failure reduces U_1 and U_2 to be of the following shape:

This reduces the 6 parameters of a general fusion to 4 if it has green failure:

Proposition 5.5. *Any fusion measurement with green failure has the following form:*



(19)

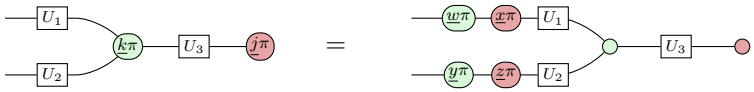
for some choice of angles $\alpha_1, \alpha_2, \beta, \varphi \in [0, 2\pi)$ *and measurement outcomes* $\underline{j}, \underline{k} \in \{0, 1\}$.

Note that a similar characterization can be obtained for red fusion by reversing colours. However if the fusion is neither "red" nor "green" then we have shown that it induces a non-unitary and thus non-correctable error on the target qubits in the failure case. We now analyse the success case and further refine our notion of correctability.

5.4 Fusion with correctable Pauli error

Since measurements in quantum computing are probabilistic processes, a fusion induces random errors in both its success and failure cases. In MBQC, such errors can be propagated when the measurement pattern has flow. To similarly propagate fusion measurement errors in our framework, they must be equivalent to local Pauli gates on the input qubits. Diagrammatically, this means that the measurement errors $\underline{k}\pi$ and $\underline{j}\pi$ can be pulled out to the input wires while keeping them in X, Y or, Z basis.

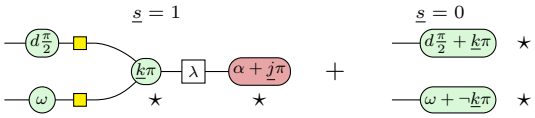
Definition 5.6 (Pauli error)**.** A fusion measurement has Pauli error when the success outcome satisfies:



for some bits $\underline{w}, \underline{x}, \underline{y}, \underline{z} \in \{0, 1\}$.

From the equation above, we deduce that either U_1 or U_2 must be Clifford, and that U_3 must be a gate locally equivalent to H, S, or Id so that the measurement is in the YZ, XZ, or XY plane, respectively. Further requiring that failure is green gives us the following characterization.

Proposition 5.7. *Any fusion measurement with green failure and Pauli error has the following form:*



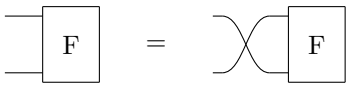
(20)

for a measurement plane $\lambda \in \{\mathrm{YZ}, \mathrm{XZ}, \mathrm{XY}\}$, angles $\alpha, \omega \in [0, 2\pi)$, and a choice of Clifford parameter $d \in \{0, 1, 2, 3\}$.

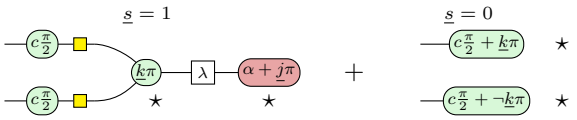
The detailed proof is given in Section C.

In practical applications, it is desirable that the action of a fusion measurement on its target qubits is symmetric, so that errors can be propagated on either qubit at will.

Definition 5.8 (Symmetric fusion)**.** We say that a fusion measurement is symmetric if it is invariant under swap in the success case, that is,



Theorem 5.9. *Any symmetric fusion measurement with green failure and Pauli error has the following form:*

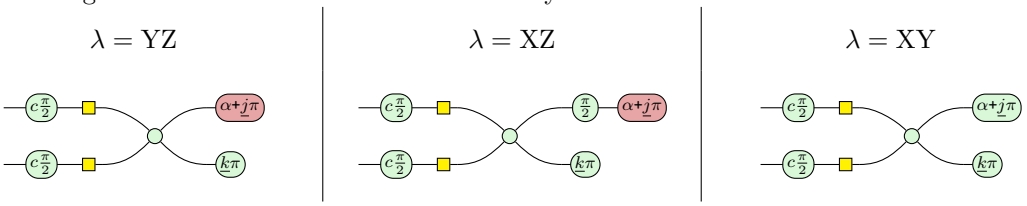


(21)

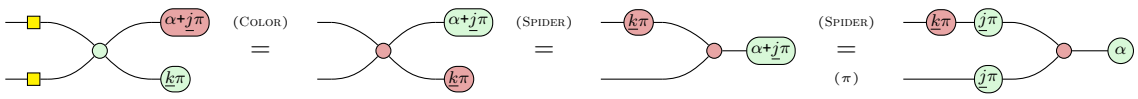
where $c \in \{0, 1\}$, $\lambda \in \{\mathrm{YZ}, \mathrm{XZ}, \mathrm{XY}\}$, and $\alpha \in [0, 2\pi)$.

Proof. This directly follows from Proposition 5.7 proved in Section C. □

Definition 5.10 (Planar fusion)**.** We call YZ, XZ, and XY fusion the three classes of symmetric fusion with green failure and Pauli error obtained by the choice of λ:



Example 5.11 (Phase gadgets)**.** A *phase gadget* is an entangling non-Clifford gate that plays an important role in quantum circuit optimization [29, 56, 105] and quantum machine learning where they allow tuning the amount of entanglement between their inputs. A phase gadget is an instance of a YZ-fusion with $c = 0$ and $\alpha \in [0, 2\pi)$:

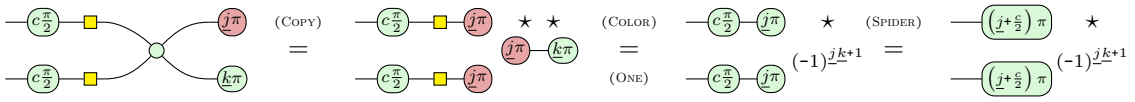


5.5 Stabilizer X and Y fusions

The characterization that we obtained for fusions with green failure and Pauli error is three-fold, corresponding to the three planes on the Bloch sphere. We now consider measurements with the additional property of being Pauli measurements, in the X, Y or Z basis.

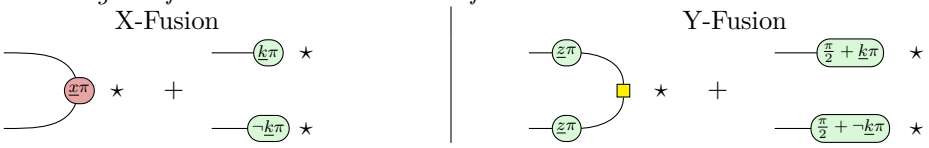
Definition 5.12 (Stabilizer fusion)**.** A fusion measurement is a stabilizer measurement if all its branches are stabilizer ZX diagrams, i.e. only involve phases with multiples of $\frac{\pi}{2}$.

Among symmetric green fusions, the characterization Theorem 5.9 together with the stabilizer assumption gives us only three possibilities for whether the measurement in plane λ is a Pauli X, Y or Z measurement. Note first that the Z-fusion is trivial: it is a separable two-qubit measurement and leaves the connectivity of the graph state unchanged.



X and Y fusions instead are entangling measurements that qualitatively change the connectivity of the graph: they either fuse two nodes into one (X-fusion) or add a Hadamard edge between them (Y-fusion).

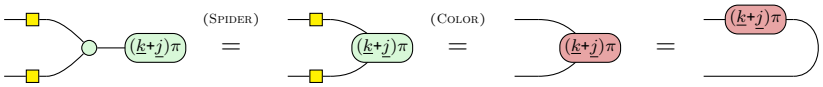
Theorem 5.13 (Stabilizer green fusions)**.** *Up to local Clifford rotation on the target qubits, entangling stabilizer green fusions are either X or Y fusions.*



Proof. Suppose we write $\alpha = a\frac{\pi}{2}$ where $a \in \{0, 1, 2, 3\}$. Then from YZ-fusion we obtain Z-fusion when a is even and Y-fusion when odd, from XZ-fusion we obtain X-fusion when a is odd and Z-fusion when even, and from XY-fusion we obtain X-fusion when a is even and Y-fusion when odd. □

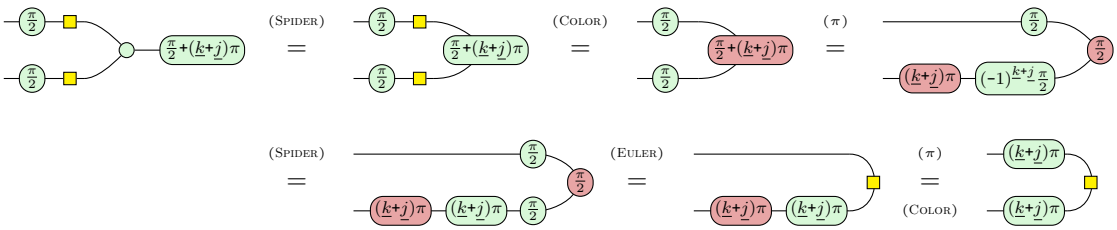
The notion of stabilizer green fusion recovers the two main examples of fusion measurements used in the literature.

Example 5.14 (Type II as X-fusion)**.** The Type II fusion [14] is an instance of X-fusion with $c = 0$:

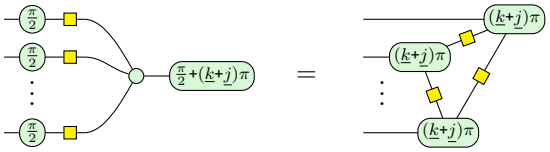


Note that setting $c = 1$ is undesirable in this case as it only changes the errors from the X to the Y basis.

Example 5.15 (CZ with Y-fusion)**.** The fusion measurement for performing CZ gates with linear optics, studied in [42, 64], is an instance of Y-fusion. Indeed, up to Pauli errors, Y-fusion with $c = 1$ adds a Hadamard edge in the success case:



Remark 5.16. Recall from Remark 5.3 that Type II fusions can be generalized to an arbitrary number of input legs. Similarly, Y-fusion can also be generalized to any number of inputs. Its action corresponds to applying a CZ gate between each pair of qubits:



where the connections of the spiders form a complete graph on the right-hand side. This rewrite rule corresponds to toggling the CZ edges between all of the nodes being fused in the fusion of the underlying graph state; a formal proof can be derived from [36, Lemma 5.2.].

To conclude this section, we have carried out a thorough characterization of all fusion measurements that preserve graph states' connectivity on failure (*green failure*) and induce correctable Pauli errors on success (*Pauli error*). Fusion measurements satisfying both these properties as well as being symmetric are XY, XZ, or YZ fusions.

6 Flow structure for fusion networks

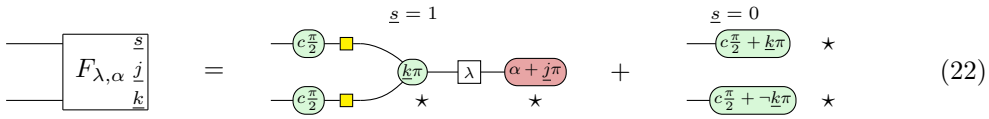
In the previous section we characterized fusion measurements that induce Pauli errors on their input qubits. The aim of this section is to describe the flow structure that enables correction of these Pauli errors. To do this, we extend the treatment of errors in standard MBQC to the fusion-based setting. We study a notion of "determinism on success" for measurement patterns with additional destructive fusion measurements. We define a notion of partially static flow for fusion networks which ensures errors can be propagated without applying corrections on fusion nodes. We prove that any fusion network with partially static flow defines a fusion pattern that is deterministic on success. Focusing then on fusion networks consisting of stabilizer X and Y fusions, we show that any decomposition of an open graph as an XY-fusion network has partially static flow, provided that the original graph has Pauli flow.

Remark 6.1. The notion studied in this section only ensures determinism after post-selecting on fusion successes, it is focused at correcting errors while reducing the feedforward required on fusion circuits. This assumption is justified in view of the repeat-until-success and boosting protocols studied in the following sections.

6.1 Fusion patterns

In fusion-based quantum computing, photons may be prepared, entangled and measured by destructive fusions and single-qubit measurements. We thus begin extending the definition of measurement patterns with the ability to perform *correctable* fusion measurements.

Definition 6.2 (Fusion pattern)**.** A fusion pattern is a diagram in `Channel`(**ZXLO**) generated by the same commands as measurement patterns (Definition 3.10) and additional two-qubit fusion commands defined according to Proposition 5.7 by:



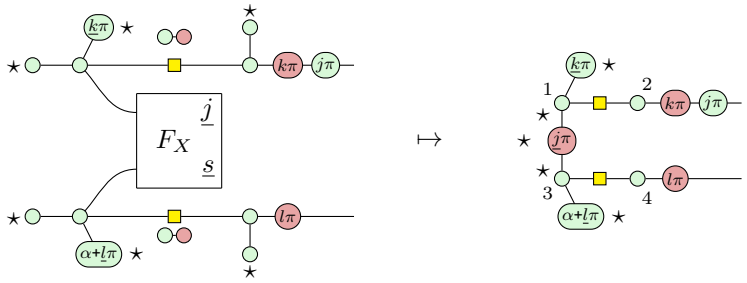
(22)

for some plane $\lambda \in \{\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}, X, Y, Z\}$, angle $\alpha \in [0, 1)$ and Clifford parameter $c \in \{0, 1\}$.

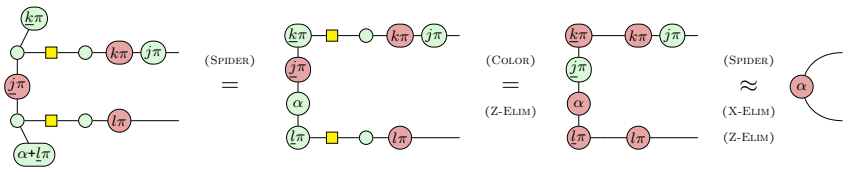
For a fusion pattern P with m measurement commands and f fusion commands, the *fusion branches* are the 2^f measurement patterns $\{P_{\vec{s}}\}_{\vec{s}\in\mathbb{B}^f}$ obtained by post selecting each success outcome according to $\vec{s}$, i.e. $\underline{s}_i = \vec{s}_i$. This corresponds to replacing the $F_{\lambda,\alpha}$ command with the success or failure outcome, according to Proposition 5.7. The branch $P_{\vec{1}}$ where all fusions are successful is called the *success branch*.

Definition 6.3 (Determinism on success)**.** We say that a fusion pattern P is *deterministic on success* if $P_{\vec{1}}$ is deterministic. Similarly, P is uniformly, strongly or stepwise deterministic on success if $P_{\vec{1}}$ is uniformly, strongly or stepwise deterministic.

Example 6.4. As an example, consider the fusion pattern defined by the following diagram P and its corresponding success pattern $P_{\vec{1}}$:



where two stars have been cancelled by the scalars from the two entangling gates. The success pattern has 8 branches obtained by setting the different values of $k, l, j \in \{0, 1\}$. By rewriting the ZX diagram above, we can show that these 8 branches are proportional to each other:



where the last equation holds Therefore, this specific pattern is deterministic on success. Moreover, each of the branches carries the same scalar, making the pattern strongly deterministic. Since the rewrite above holds for any angle α, the pattern is also uniformly deterministic. By considering the pattern truncated at single-qubit measurement commands, a similar rewrite shows that it is also stepwise deterministic on success.

6.2 Fusion networks

Fusion networks represent configurations of single-qubit measurements and fusion measurements on a resource graph state. Just as open graphs define the underlying topology of a measurement pattern, fusion networks capture the underlying topology of fusion patterns.

Definition 6.5 (Fusion network)**.** A fusion network, denoted by $\mathcal{F} = (G, I, O, F, \lambda, \alpha, c)$, is given by the following:

1. an open graph (G, I, O) (called 'resource graph'),
2. a set of fusions $F \subseteq \mathcal{M}(\overline{O} \times \overline{O})$,
3. an assignment of measurement planes $\lambda_G : \overline{O} \to \{\,\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}, X, Y, Z, *\,\}$ and $\lambda_F : F \to \{\,\mathrm{XY}, \mathrm{XZ}, \mathrm{YZ}, X, Y\,\}$ (which we denote simply by λ when the domain is unambiguous),
4. an assignment of measurement angles $\alpha : \overline{O} + F \to [0, 2\pi)$ ($\alpha(v)$ is set to zero if $\lambda(v) \in \{\,X, Y, Z\,\}$), and
5. a Clifford parameter for each measured qubit $c : \overline{O} \to \{\,0, 1, 2, 3\,\}$.

where $\mathcal{M}$ denotes the multi-set construction, $+$ denotes the disjoint union, and $-$ denotes the set difference. An XY-fusion network is a fusion network where all fusions are either X or Y fusions.

Remark 6.6. The assignment of measurement planes to nodes in the resource graph λ_G may return $*$ which means that the node does not have a single-qubit measurement. We call these nodes "unmeasured". This makes the definition more flexible, as nodes may be part of a fusion and not of a single-qubit measurement. Moreover, the definition of fusion network given here references a single resource graph. In practice, the graph G may be the disjoint union of multiple copies of the same basic resource state.

Following Theorem 5.9, a successful fusion has the effect of introducing an additional node in the graph, measured in an arbitrary plane and angle. Thus, any fusion network $\mathcal{F}$ defines a *target open graph*, denoted $\mathcal{M}_\mathcal{F}$, capturing the computation performed when we post select on fusion successes.

Definition 6.7 (Target open graph)**.** Given a fusion network $\mathcal{F} = (G, I, O, F, \lambda, \alpha, c)$ with $G = (V, E)$. Any fusion $f \in F$ contributes an extra vertex to the graph, labelled v_f. The *target open graph* of $\mathcal{F}$ is $\mathcal{M}_\mathcal{F} \coloneqq (G_\mathcal{F}, I, O, \lambda_\mathcal{F}, \alpha_\mathcal{F})$, where $G_\mathcal{F} = (V_\mathcal{F}, E_\mathcal{F})$,

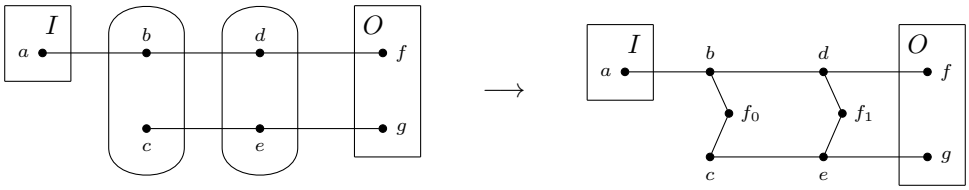
$$V_\mathcal{F} = V \cup \{\, v_f \mid f \in F \,\} \qquad E_\mathcal{F} = E \cup \{\, (v_f, w) \mid w \text{ belongs to } f \in F \,\}$$

$$\lambda_{\mathcal{F}}(u) = \begin{cases} \lambda(f), & \text{if } u = v_f \text{ for some } f \in F \\ X & \text{if } \lambda(u) = * \wedge c(u) \bmod 2 \equiv 0 \\ Y, & \text{if } \lambda(u) = * \wedge c(u) \bmod 2 \equiv 1 \\ \mathrm{YZ}, & \lambda(u) = \mathrm{XZ} \wedge c(u) \bmod 2 \equiv 1 \\ \mathrm{XZ}, & \lambda(u) = \mathrm{YZ} \wedge c(u) \bmod 2 \equiv 1 \\ \lambda(u), & \text{otherwise} \end{cases} \qquad \alpha_{\mathcal{F}}(u) = \begin{cases} \alpha(f), & \text{if } u = v_f \text{ for some } f \in F \\ \alpha(u) + \dfrac{c(u)\pi}{2}, & \lambda(u) = \mathrm{XY} \\ (-1)^{\left\lceil \frac{c(u)}{2} \right\rceil} \alpha(u), & \lambda(u) = XZ \\ (-1)^{\left\lfloor \frac{c(u)}{2} \right\rfloor} \alpha(u), & \lambda(u) = YZ \end{cases}$$

The *target linear map* of the fusion network $T(\mathcal{F})$ is the target linear map of $\mathcal{M}_{\mathcal{F}}$.

In other words, nodes and edges of G are extended with those coming from the set of fusions F. The fusion measurement planes and angles are part of the new single-qubit measurement parameters. We model "unmeasured" nodes in the resource graph (such that $\lambda(u) = *$) as measured in the X basis. Furthermore, some of the original single-qubit measurements are modified if their Clifford parameters are non-zero. Clifford parameters on measured qubits correspond to changes in measurement planes. We can capture these by the following equations:

Example 6.8. Consider a fusion network with a pair of lines as the resource graph and two fusions. The target measurement graph is obtained by adding a new node in the graph for each fusion.



6.3 Partially static flow

We now define a notion of flow for fusion networks that makes them deterministically implementable by a fusion pattern. We assume that corrections can only be applied before a single-qubit measurement. This ensures that fusion measurements are implemented by a *static* setup. Similarly, unmeasured nodes in the resource graph are modeled as X measurements, but it is not possible to apply a correction on these nodes. According to the definition of Pauli flow from Definition 3.17, X corrections will be performed in $g(v) - \{ v \}$ and Z corrections in $\mathtt{Odd}(g(v)) - \{ v \}$, for any non-output node v. We thus obtain the following definition.

Definition 6.9 (Partially static flow)**.** A static flow for a fusion network $\mathcal{F} = (G, I, O, F, \lambda, \alpha)$ is a Pauli flow $(p, \leq)$ on the target open graph $\mathcal{M}_{\mathcal{F}}$, such that no corrections need to be applied on fusion nodes. Concretely, for any node in the resource graph $u \in \overline{O}$ we must have:

- if $\lambda(u) = *$ then for any $v \in \mathcal{M}_{\mathcal{F}}$, $f \notin \mathtt{Odd}(p(v))$.

and for any fusion node $f \in F$ we must have:

- if $\lambda(f) = X$ then for any $v \in \mathcal{M}_{\mathcal{F}}$, $f \notin \mathtt{Odd}(p(v))$,
- if $\lambda(f) \in \{ \mathrm{XY}, \mathrm{XZ}, \mathrm{YZ} \}$ then for any $v \in \mathcal{M}_{\mathcal{F}} - \{ f \}$, $f \notin \mathtt{Odd}(p(v)) \cup p(v)$.

Remark 6.10. Note that the condition above is precisely what is necessary to define a flow on the target open graph which does not require corrections on fused and unmeasured nodes. For Y-measured nodes this is already the case by conditions $1-3$ in Definition 3.17, so we do not need to impose additional conditions.

Following [13], we prove that our notion of flow is both necessary and sufficient for an XY-fusion pattern to be uniformly, strongly and stepwise deterministic on success. Moreover, every such pattern can be factorized such that all fusions appear before single-qubit measurements.

Theorem 6.11. *Given a fusion network $\mathcal{F}$ with static flow $(p, <)$ the fusion pattern defined by:*

$$\left(\prod_i^{<} X^{k_i}_{g(i)} Z^{k_i}_{\mathtt{Odd}(g(i))} M_i^{\lambda_i,\alpha_i,\underline{k_i}}\right)\left(\prod_{f=(i,j)\in F} X^{k_f}_{g(f)} Z^{k_f}_{\mathtt{Odd}(g(f))} F_{ij}^{\lambda(f),\underline{s_f},\underline{k_f}}\right) E_G N_{\overline{I}}$$

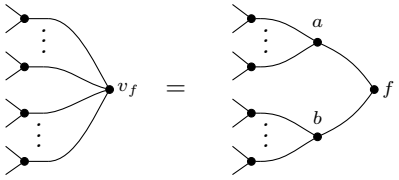
is uniformly, strongly, and stepwise deterministic on success and implements the target linear map $T(\mathcal{F})$ when all fusions are successful. Here, $g(i) = p(i) \cap \{j \mid i < j\}$, $\prod^{<}$ denotes concatenation in the order $<$ and $\prod$ denotes concatenation in any order.

Proof. Since $\mathcal{M}_{\mathcal{F}}$ has Pauli flow, we have a correction function $p : \overline{O} + F \to \mathcal{P}(\overline{O} + F)$ satisfying the Pauli flow conditions. Since the errors in the success branches of $\mathcal{F}$ correspond exactly to the errors in $\mathcal{M}_{\mathcal{F}}$, by [13, Theorem 4], $\mathcal{F}$ is uniformly, strongly and stepwise deterministic on success and implements the target linear map $T(\mathcal{F}) = T(\mathcal{M}_{\mathcal{F}})$. Moreover, since no corrections need to be applied on fusion nodes, the fusion nodes can be made initial in the order $<$. This gives us the factorisation required, where every fusion appears before single qubit measurements. □

6.4 Decomposing open graphs as XY-fusion networks

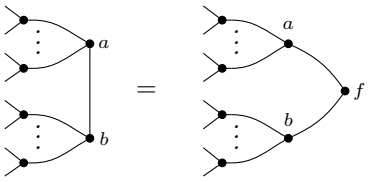
Suppose we wish to implement a given labelled open graph $\mathcal{G}$ as an XY-fusion network. Several different fusion networks may exist that have $\mathcal{G}$ as their simplified target graph. We now show that any such decomposition of $\mathcal{G}$ as a fusion network $\mathcal{F}$ is guaranteed to have static flow, provided that $\mathcal{G}$ has Pauli flow. To prove this, we use rewrites that preserve the existence of Pauli flow [70, 71, 98].

Proposition 6.12 (X-fusion)**.** *The following open graph rewrite preserves the existence of Pauli flow:*



where $\lambda(f) = \lambda(b) = X$, $\lambda(a) = \lambda(v_f)$, and $\alpha(a) = \alpha(v_f)$.

Proposition 6.13 (Y-fusion)**.** *The following open graph rewrite preserves the existence of Pauli flow:*



where $\lambda(f) = Y$, $c(a) = c(b) = 0$ on the left and $c(a) = c(b) = 1$ on the right-hand side.

The proofs are in Appendix E.

We can now show that Pauli flow on the target open graph $\mathcal{G}_{\mathcal{F}}$ is both necessary and sufficient for $\mathcal{F}$ to have static flow.

Theorem 6.14. *An XY-fusion network $\mathcal{F}$ has static flow if and only if the simplified target graph $\mathcal{G}_{\mathcal{F}}$ has Pauli flow.*

Proof. This follows from the two propositions above. For X-fusion we moreover need to show that when rewriting from $\mathcal{G}_{\mathcal{F}}$ to $\mathcal{M}_{\mathcal{F}}$ the newly introduced fusion node is not in the odd neighbourhood of some correction set. Using the notation of Proposition 6.12, suppose that $v \in \mathtt{Odd}(g(u))$ for some node u in $\mathcal{G}_{\mathcal{F}}$ where g is the Pauli flow on $\mathcal{G}_{\mathcal{F}}$, then if u is a neighbour of v, it is a neighbour of either a or b in $\mathcal{M}_{\mathcal{F}}$ (and not a neighbour of both). Therefore, we can set $p(u) = g(u) - \{v\} + \{a, b\}$ as the correction function in $\mathcal{M}_{\mathcal{F}}$ without changing the connectivity of u to its correction set, and thus without violating the Pauli flow conditions for u. Then we have $f \notin \mathtt{Odd}(p(u))$, as required. □

7 Universality in fusion-based architectures

Different architectures have recently been proposed to perform photonic and distributed quantum computing. These include all-photonic approaches such as the FBQC proposal [8], and matter-based architectures such as spin-optical [42] and ion-based [68, 74] architectures. The graphical framework developed in this paper allows us to represent, reason about and compare these architectures on an equal footing.

In this section, we study universality in distributed architectures, focusing on two examples:

1. lattice-based photonic architectures based on a constant-size resource state generator and fusion measurements [8].
2. reprogrammable architectures based on quantum emitters [42, 48]

We work in an idealised setting satisfying the following assumptions:

- resource state generation is deterministic,
- photons are indistinguishable,
- all components are noiseless and have perfect efficiency.

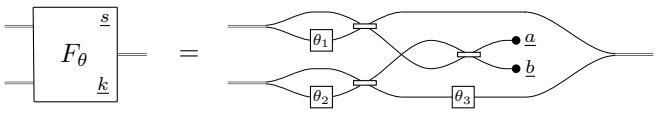
While these assumptions are strong, a proof of universality in this setting needs to address both the probabilistic nature of fusion measurements and the correction of Pauli byproducts induced by undesired measurement outcomes. Relaxing any of the above assumptions gives an error model for these architectures which may be further analysed with the components of Section 3.3.

We start this section by proving the correctness of protocols that boost the probability of success of fusion measurements. Exploiting these results we then provide two minimal examples of architectures — lattice-based and reprogrammable — giving graphical proofs of determinism and (weak) universality for each. Compared to previous approaches [8, 42] based on the generation of a universal lattice, our results show the potential for intermediate scale photonic quantum computing in programmable optical setups enhanced by compilation techniques.

7.1 Boosting fusion with entangled resource states

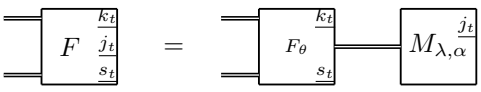
Boosting the success probability of fusion measurements is an essential requirement for scaling FBQC. Indeed the naive fusion measurement has a probability of success of only $\frac{1}{2}$ (see Section D). Common approaches to boosting include the use of ancillary single photons [38, 43], which however require exponentially many ancillas to arbitrarily approach probability 1. Here, we focus on boosting protocols that use entangled states as resource and reach probabilities arbitrarily close to 1 with a linear number of attempts [62, 64]. Known methods apply only to X-fusion [62] and Y-fusion [64]. We give a formal graphical proof of their correctness and we generalise the protocol of [64] to arbitrary fusions with green failure.

Let us consider the circuit of a non-destructive fusion measurement with green failure, parametrised by three phases θ_1, θ_2 and θ_3.



(23)

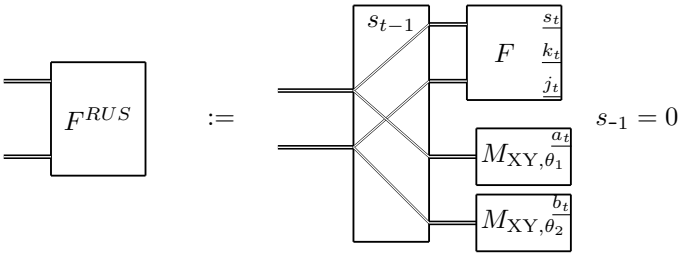
We define the *passive fusion module*, implementing arbitrary correctable fusions with green failure, as the following stream:



(24)

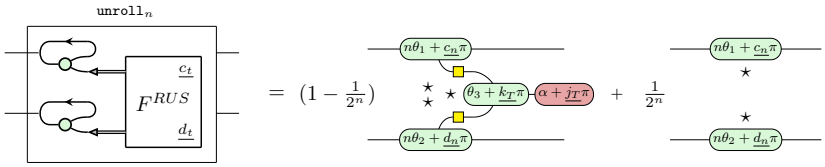
This module produces two streams of classical outputs: the success values $\underline{s_t}$ and the Pauli byproducts $\underline{k_t}, \underline{j_t}$. We are now ready to state our results about repeat-until-success protocols. The proofs are given in Section F.

Definition 7.1 (RUS protocol)**.** The repeat-until-success fusion protocol is defined by the following setup:



Note that the control parameter of the switch takes the value s_{t-1} of the previous success outcome.

Theorem 7.2. *Any fusion with green failure can be boosted with a repeat-until-success protocol. More precisely, the following holds for $n \geq 1$:*



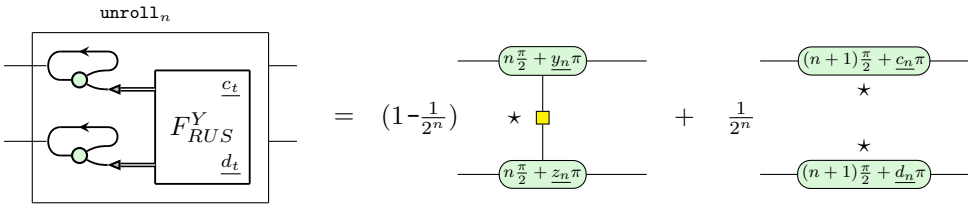
where T is the time of the first successful fusion (if it exists) and:

$$\underline{\Sigma_t} \coloneqq \sum_{i=0}^{t} \underline{s_t} \qquad c_t = c_{t-1} \oplus (\neg \underline{\Sigma_t}) k_t \oplus \underline{\Sigma_{t-1}} a_t \qquad d_t = d_{t-1} \oplus (\neg \underline{\Sigma_t})(\neg k_t) \oplus \underline{\Sigma_{t-1}} b_t$$

with $s_0 = 0$, $c_0 = d_0 = 1$.

As a consequence, the probability of success of a repeat-until-success fusion protocol after n time-steps is $1 - \frac{1}{2^n}$. We recover the RUS protocol of [64] as the special case for Y-fusion.

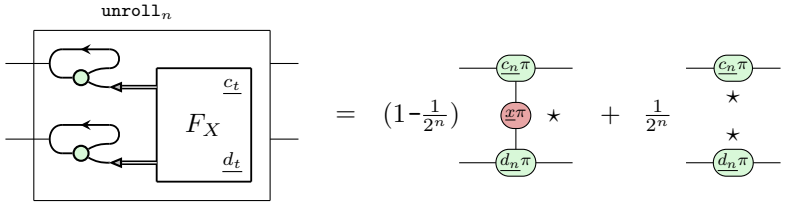
Corollary 7.3 (Y fusion RUS)**.** *For $n \geq 1$ we have:*



where $z_t = (k_T \oplus j_T) \oplus c_t$ and $y_t = (k_T \oplus j_T) \oplus d_t$ if $T < t$ and $y_t = (k_T \oplus j_T) \oplus \neg c_t$ if $T = t$.

It was shown in [62] that X-fusion, corresponding to a Bell measurement in the success case, can be boosted with entangled resource states. As the X-fusion is an idempotent projection in the success case, and it has a correctable failure, we can boost it simply by repeating it on a GHZ resource state.

Proposition 7.4 (Boosted X Fusion)**.** *For $n \geq 1$ we have:*

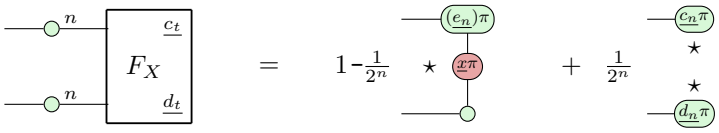


where $c_{-1} = d_{-1} = 0$ and

$$\underline{c_{n+1}} = \begin{cases} \underline{c_n} & \text{if } s_{n+1} = 1 \\ \underline{c_n} + \underline{k_{n+1}} & \text{if } s_{n+1} = 0 \end{cases} \qquad \underline{d_{n+1}} = \begin{cases} \underline{d_n} & \text{if } s_{n+1} = 1 \\ \underline{d_n} + \neg \underline{k_{n+1}} & \text{if } s_{n+1} = 0 \end{cases}$$

This has practical advantages compared to the RUS protocol in Definition 7.1, as the above stream requires no feed forward.

Corollary 7.5. *For $n \geq 1$ we have:*



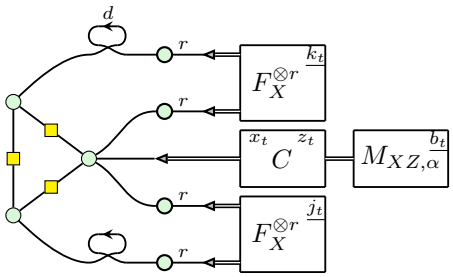
where $e_n = c_n + d_n$.

7.2 Lattice-based architecture

Lattice-based architectures for universal quantum computing [8, 41] are structured as follows:

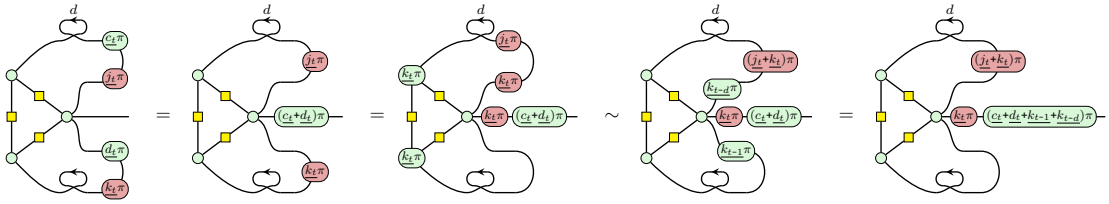
1. a resource state generator (see Section 4.5) is used to construct a small constant-size entangled state at every time-step,
2. the photons undergo a bounded-depth circuit built from linear optics and delays,
3. photons from different time-bins are either fused or directly measured.

To illustrate these architectures we focus on a minimal example involving a *triangle resource state* and XZ measurements, inspired by [72]. Other examples of universal lattice-based architectures can be obtained by replicating the argument below for the streams in Figure 1. We define our triangle architecture as the following protocol:

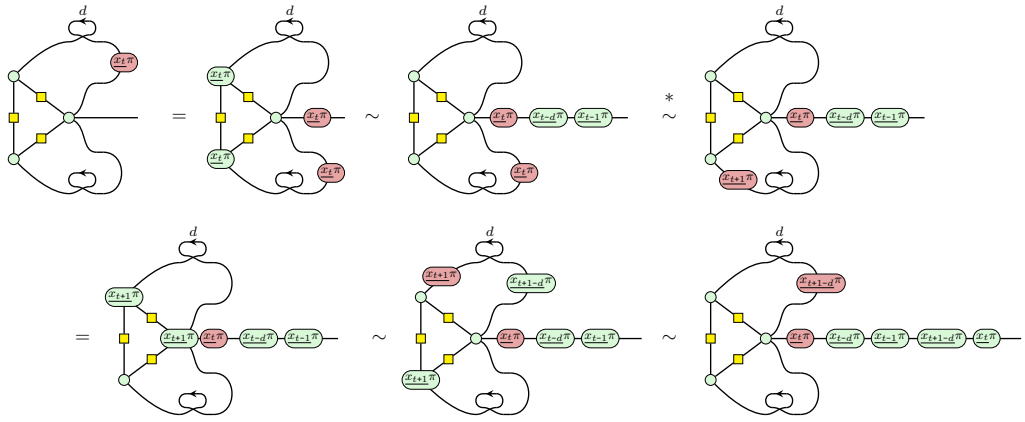


(25)

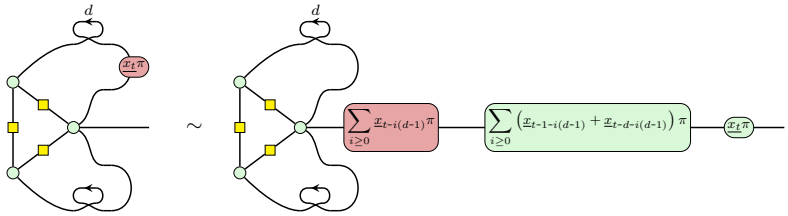
where the delays are initialised with $|+\rangle$ states and the delay labelled d is the d-fold composition of such delays (outputting $|+\rangle$ for the first d time steps). Using Corollary 7.5 and post-selecting on fusion successes, we obtain a diagram in $\texttt{Stream}(\texttt{Channel}(\mathbf{ZX}))$ which we can simplify as follows:



Then, we show that:



where $(*)$ uses Rewrite 9 and the fact that the delay is initialised with a $|+\rangle$ state satisfying $X|+\rangle = |+\rangle$. Applying the above equality recursively, similarly to Example 4.16, we obtain the observational equality:



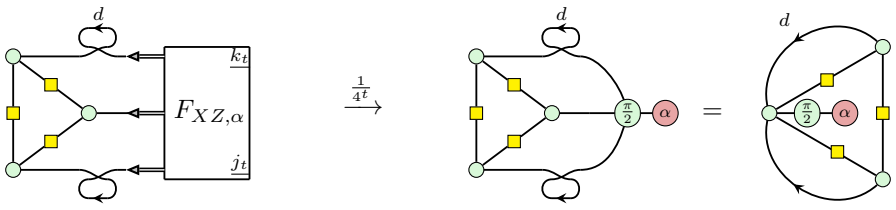
where $x_t := 0$ for $t < 0$ so that the sums are over finitely many previous measurement outcomes. Putting this all together, we need to apply X correction and Z correction summing the above. Moreover, applying Definition 4.3 to the resulting stream builds a triangular lattice where each node is measured in the XZ plane, see Figure 1 and Theorem 4.15. As shown in [72], the triangular lattice with XZ measurements is universal for qubit circuits. Therefore, we see that the protocol in Equation (25) simulates a universal architecture with decreasing probability $(1 - \frac{1}{2^r})^{2t}$, in the sense of Definition 4.11. This is however not sufficient to prove strong universality of the protocol, in the sense of Definition 4.12, since the parameter r is constant in the setup. Instead, we obtain a weak universality result, in the sense of Definition 4.13, as follows.

Theorem 7.6. *The architecture in Equation (25) is weakly universal for qubit circuits generated by* $\{CZ, H, Z(\alpha)\}$ *of width* $\frac{d-1}{4}$ *and depth bounded by a constant* k.

Proof. Note first that, according to [72], a triangular lattice of size $4n \cdot 4k$ is required to compile a circuit of width n and depth k. Thus, given a circuit as above, we need at most $4dk$ time-steps to build a lattice of the correct size, using one time step every d to flatten the lattice to the plane by applying Z measurements. Moreover, by the rewrite above, all the Pauli byproducts can be corrected by classical feedforward and the execution of the protocol is deterministic on success. The lattice is obtained when all fusions succeed, with probability $(1 - \frac{1}{2^r})^{8dk}$. Therefore we can set $\epsilon \geq 1 - (1 - \frac{1}{2^r})^{8dk}$ to obtain a simulation of all the circuits in the set within probability $1 - \epsilon$. □

Stronger universality results may be obtained using percolation methods to avoid post-selecting on fusion successes. In fact, the percolation threshold of the triangular lattice is $\frac{1}{2}$ [69], which is reached when $r = 2$. While this ensures the existence of a path from the input to the output boundary of the lattice, it does not give us information about the size of this connected component and thus, given a circuit C, we cannot ensure that the connected component will be large enough to accommodate C. Renormalisation ideas [46] may offer a way to show this rigorously, but we leave their formalisation in the ZX calculus to future work.

Remark 7.7. The architecture of Equation (25) can be simplified by exploiting the planar fusion measurements introduced in Section 5.4. Instead of performing two X-fusion measurements and an XZ single-qubit measurement, we could perform a single ternary XZ-fusion measurement:



where the arrow denotes simulation according to Definition 4.11 and we omitted the Pauli byproducts for simplicity. A RUS protocol for this ternary fusion can be obtained by a simple generalisation of Definition 7.1 although it requires more active switching compared to the static boosted X-fusion.

Remark 7.8. The architecture studied in this section relies on the deterministic generation of the triangle graph state. For an all-photonic approach this could be achieved using, for example, switch networks [6], although a convincing experimental demonstration has not been achieved

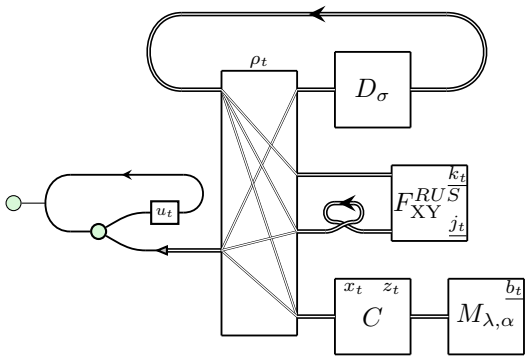
to date. The other main source of noise in an all-photonic setup is photon loss, which can be heralded but will further impact the probability of successfully generating the lattice, and thus the tolerance ϵ used in Theorem 7.6. The triangle resource state could also be constructed using other resource-state generation techniques such as quantum emitters.

7.3 Reprogrammable architecture

Quantum computing architectures based on quantum emitters [42, 84] can leverage variable-length resource states allowing constructive forms of universality where a given quantum computation is directly generated, rather than carved out of a lattice. They are generally organised as follows:

1. quantum emitters generate variable-length linear cluster states,
2. photons from different quantum emitters and photon time-bins are routed into either fusions or single-qubit measurements.

We now study a simple version of these architectures, based on a single quantum emitter, and show that it is universal for *arbitrary* qubit circuits. The protocol is built from a quantum emitter, the delay, photonic measurement and correction modules defined in Section 4.4, and the repeat-until-success XY-fusion module of Definition 7.1. We define it as the following diagram:

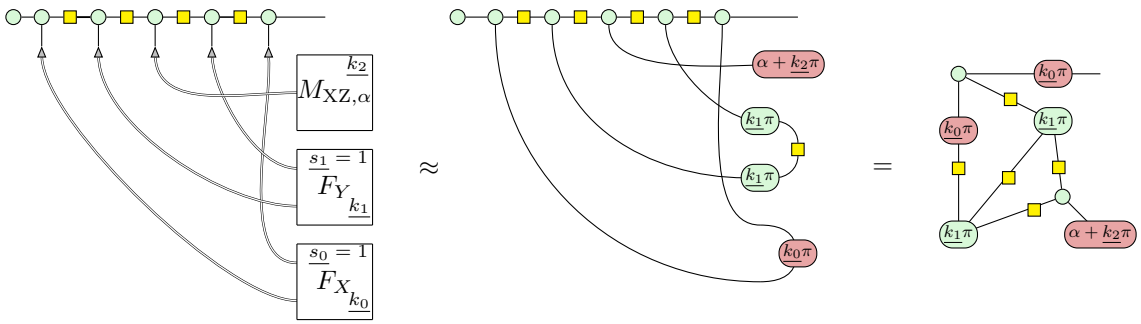


(26)

By recursively applying Definition 4.3, we produce a diagram in **ZXLO** which is structured as follows:

- the top part of the diagram consists of a variable GHZ-linear cluster produced by the emitter,
- the middle part is obtained by unrolling the delay module and equates to a permutation of the qubits,
- the bottom part is a sequence of fusions and single qubit measurements of arbitrary types.

As an example, the following is the success term of a possible unrolling of the architecture:



To prove universality of the architecture, we show that it can be used to implement any XY-fusion network where the resource graph is a line.

Definition 7.9. A linear XY-fusion pattern is an XY-fusion pattern where the n-qubit register is totally ordered $V = \{1, \ldots, n\}$ and the entangling commands are restricted to be of the form $E_{i,i+1}$. A linear XY-fusion network is an XY-fusion network where the resource graph is a disjoint union of lines.

Proposition 7.10. *For any labeled open graph $\mathcal{G}$ with flow, there is a linear XY-fusion network $\mathcal{F}$ with flow with the same target linear map $T(\mathcal{G}) = T(\mathcal{F})$.*

Proof. We use exclusively the Y-fusion measurement which adds a hadamard edge between nodes. Given any labeled open graph $\mathcal{G} = (G, I, O, \lambda, \alpha)$ with flow, we may extend it to an equivalent labeled open graph $(G', I, O, \lambda, \alpha)$ such that G' has a Hamiltonian path by finding a Hamiltonian completion of G. The additional edges in the completion are constructed by introducing nodes measured in the Z-basis, an operation which preserves the existence of Pauli flow [70]. Then we may construct a linear XY fusion network (L, F) where $L \subseteq G'$ is the Hamiltonian path and F is the set of all remaining edges. It is easy to check that this fusion network has the same target linear map as G. □

Proposition 7.11. *The protocol in (26) has settings $\lambda, \alpha, \sigma, \rho, u$ that implement any runnable linear XY-fusion pattern, with probability arbitrarily close to* 1.

Proof. Fix a linear XY-fusion pattern. Let f_i be the number of fusions applied to qubit i. The total number of fusion operations is $f = \frac{1}{2}\sum_{i=1}^{n} f_i$. For each qubit i, we emit $kf_i + 1$ photons entangled as a GHZ state with the atom, where r is a positive integer. Between rounds, we either apply a hadamard gate on the atom if the command $E_{i,i+1}$ is present, or else we emit an additional photon to be measured in the X basis. By setting the parameters ρ and σ we may route these photons arbitrarily in either a RUS fusion or a single-qubit measurement, following Lemma 4.19. We use r photons for each node in a RUS fusion operation, giving us a probability of success of $(1 - \frac{1}{2^r})$ for each of the f fusions. If the RUS fusion fails after k rounds we restart the whole computation. Finally, we can apply any sequence of single qubit measurements and corrections on the remaining n photons, following Lemma 4.20. In order to achieve a total success probability ϵ close to 1 we just have to set r an odd integer such that $(1 - \frac{1}{2^r})^f > 1 - \epsilon$. □

Note that the RUS X and Y fusions defined above may induce additional Z errors on the target qubits. Even for Y-fusion, it is sufficient to set n to be even (i.e. repeat an odd number of times) to ensure that the error is Pauli. Thus, in order to correct these errors in an XY-fusion pattern, we must add Z corrections on the target qubit. This is always possible with the factorisation given in Theorem 6.11, since fusion nodes precede their target qubits in the partial order.

Theorem 7.12. *The protocol in Equation (26) is strongly universal for any qubit unitary.*

Proof. Given any qubit unitary, we may represent it as a labeled open graph $\mathcal{G}$ with flow. By Proposition 7.10, there is a linear XY fusion pattern $\mathcal{F}$ with flow and the same target linear map. This gives rise to a runnable linear XY fusion pattern and the result follows by Proposition 7.11. □

The proof of strong universality above crucially relies on a linear cluster state of length arbitrary. In practice however, the length of the resource state is limited by the coherence time of the atom. If R is the rate of emission and T is the maximum coherence time, we can have at most $N = RT$ photons in each resource state. With this restriction, we cannot show strong universality, as this relied on the ability to boost the fusion success probability arbitrarily. We thus obtain the following weak universality result, which can be directly compared with Theorem 7.6.

Theorem 7.13. *The protocol in Equation (26), with maximum resource state length N, is weakly universal for MBQC patterns with number of edges bounded by a constant E.*

Proof. Given a set of MBQC patterns with number of edges bounded by a constant E. We follow the same procedure as in Proposition 7.11 but we now can attempt each fusion for only a bounded number of times. In the worst case scenario, each node in the graph state is implemented by a single round of N photon emissions and we can thus repeat each fusion at most $\frac{N}{m}$ times, where m is the maximum vertex order of the MBQC patterns. The total number of photons for this worst-case scenario is NE, which is proportional to the total time required to implement any pattern in

the set. Then we can set $\epsilon \geq 1 - (1 - \frac{1}{2^{\frac{N}{m}}})^E$ to get the desired approximation over all patterns in the set. □

The scaling obtained applies to a worst-case scenario and may be improved for optimized MBQC patterns by using the linear cluster states to cover a larger part of the graph, thus reducing the number of fusions which are the main limiting factor. Using multiple quantum emitters would allow to reduce delays and the total execution time. The programmable nature of the setup in Equation (26) allows for these types of optimization which are not available in the setup of Equation (25) that relies on a fixed compilation process onto the lattice. Moreover, note that implementing an MBQC graph G with the gate set of Equation (25) may require a circuit with up to $O(E^3)$ gates [72]. This suggests that programmable setups with variable delays may offer a more feasible route to short-term photonic computing, although the additional number of components required could have an impact on their long-term scalability.

Remark 7.14. The architecture studied in this section can be implemented using optical components currently available in advanced photonic labs. In particular, as discussed in Section 4.5, the quantum emitter gives a high level of determinism to the resource state generation part. The other sources of error will be photon loss and distinguishability. The first can be heralded but will impact the tolerance ϵ of Theorem 7.13. The latter cannot be heralded and thus will have an impact on the fidelity of the resulting graph state. Using a *single* quantum emitter can however mitigate this latter source of error, at the cost of longer delay lines and thus photon loss.

8 Conclusion

We have introduced a graphical framework for networked quantum computing that integrates linear optics, the ZX calculus, and dataflow programming. This framework connects photonic hardware models with formal methods from programming languages, supporting verification, optimization, and compilation in distributed quantum architectures.

At the **base layer**, we combined ZX diagrams with linear optical circuits to analyze qubit-photon interactions (Section 2). We characterized all fusion measurements with failure outcomes in the XY plane and whose success outcomes induce correctable Pauli errors, and identified the class of correctable measurements that give rise to useful entangling operations (Section 5).

At the **channel layer**, we extended the calculus to capture measurement, feedforward, and classical–quantum interaction (Section 3). Within this setting, we formally distinguished fusion patterns from fusion networks, and introduced the notion of partially static flow, showing that decompositions of open graphs with Pauli flow yield fusion networks with guaranteed correctability (Section 6). These results support recent work on fusion networks [61, 109, 110] by providing the first systematic account of determinism in measurement-driven photonic protocols.

At the **stream layer**, we enriched the formalism with discrete-time dynamics to describe protocols acting on photon streams with routers, delays, switches, and time-controlled emitters (Section 4). This extension enables inductive proofs about dynamic quantum processes, including the correctness of new repeat-until-success protocols that boost the probability of success of planar fusion measurements (Section 7.1). Using these tools, we established universality results for photonic architectures (Section 7), showing how MBQC patterns can be compiled into concrete optical instructions without recourse to universal graph states.

Several avenues for future work naturally emerge from this work. Among them are the characterization of higher-arity fusion measurements, the development of Pauli-flow techniques for stream processes, and the use of diagrammatic methods to capture renormalization ideas [46] in universality proofs. On the practical side, optimization strategies that reduce fusion counts, time delays, and optical component depth will be crucial to bridging theory and experiment. More broadly, the framework establishes a foundation for systematic methods in distributed quantum computing, with applications ranging from distributed error correction to advanced quantum communication protocols.

Acknowledgements

GdF would like to thank Mario Román for helping elucidate multiple aspects of the stream formalism. We would like to thank Will Simmons for reviewing an earlier version of this manuscript. We benefited from discussions with Ross Duncan, Pierre-Emmanuel Emeriau, Paul Hilaire, Dan Mills, Lucía Tormo Bañuelos, Razin A. Shaikh, and Richie Yeung.

References

[1] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 333–342, New York, NY, USA, June 2011. Association for Computing Machinery. ISBN 978-1-4503-0691-1. DOI: 10.1145/1993636.1993682.

[2] Samson Abramsky and Chris Heunen. H*-algebras and nonunital Frobenius algebras: First steps in infinite-dimensional categorical quantum mechanics. In Samson Abramsky and Michael Mislove, editors, *Mathematical Foundations of Information Flow*, volume 71 of *Proceedings of Symposia in Applied Mathematics*, pages 1–24. American Mathematical Society, 2012. ISBN 978-0-8218-4923-1. DOI: 10.1090/psapm/071.

[3] Pablo Andres-Martinez, Tim Forrer, Daniel Mills, Jun-Yi Wu, Luciana Henaut, Kentaro Yamamoto, Mio Murao, and Ross Duncan. Distributing circuits over heterogeneous, modular quantum computing network architectures. *Quantum Science and Technology*, 9(4):045021, October 2024. ISSN 2058-9565. DOI: 10.1088/2058-9565/ad6734.

[4] Koji Azuma, Kiyoshi Tamaki, and Hoi-Kwong Lo. All-photonic quantum repeaters. *Nature Communications*, 6(1):6787, April 2015. ISSN 2041-1723. DOI: 10.1038/ncomms7787.

[5] Miriam Backens, Hector Miller-Bakewell, Giovanni de Felice, Leo Lobski, and John van de Wetering. There and back again: A circuit extraction tale. *Quantum*, 5:421, March 2021. DOI: 10.22331/q-2021-03-25-421.

[6] Sara Bartolucci, Patrick Birchall, Damien Bonneau, Hugo Cable, Mercedes Gimeno-Segovia, Konrad Kieling, Naomi Nickerson, Terry Rudolph, and Chris Sparrow. Switch networks for photonic fusion-based quantum computing, September 2021. URL `https://doi.org/10.48550/arXiv.2109.13760`.

[7] Sara Bartolucci, Patrick M. Birchall, Mercedes Gimeno-Segovia, Eric Johnston, Konrad Kieling, Mihir Pant, Terry Rudolph, Jake Smith, Chris Sparrow, and Mihai D. Vidrighin. Creation of Entangled Photonic States Using Linear Optics, June 2021. URL `https://doi.org/10.48550/arXiv.2106.13825`.

[8] Sara Bartolucci, Patrick Birchall, Hector Bombín, Hugo Cable, Chris Dawson, Mercedes Gimeno-Segovia, Eric Johnston, Konrad Kieling, Naomi Nickerson, Mihir Pant, Fernando Pastawski, Terry Rudolph, and Chris Sparrow. Fusion-based quantum computation. *Nature Communications*, 14(1):912, February 2023. ISSN 2041-1723. DOI: 10.1038/s41467-023-36493-1.

[9] B. B. Blinov, D. L. Moehring, L.-M. Duan, and C. Monroe. Observation of entanglement between a single trapped atom and a single photon. *Nature*, 428(6979):153–157, March 2004. ISSN 1476-4687. DOI: 10.1038/nature02377.

[10] Hector Bombin, Daniel Litinski, Naomi Nickerson, Fernando Pastawski, and Sam Roberts. Unifying flavors of fault tolerance with the ZX calculus. 8:1379. DOI: 10.22331/q-2024-06-18-1379.

[11] Filippo Bonchi, Elena Di Lavore, and Mario Román. Effectful mealy machines: Coalgebraic and causal traces (invited talk). In *11th Conference on Algebra and Coalgebra in Computer Science (CALCO 2025)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik. DOI: 10.4230/LIPIcs.CALCO.2025.1.

[12] S. Bose, V. Vedral, and P. L. Knight. Multiparticle generalization of entanglement swapping. *Physical Review A*, 57(2):822–829, February 1998. DOI: 10.1103/PhysRevA.57.822.

[13] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. Generalized flow and determinism in measurement-based quantum computation. 9(8):250–250. ISSN 1367-2630. DOI: 10.1088/1367-2630/9/8/250.

[14] Daniel E. Browne and Terry Rudolph. Resource-Efficient Linear Optical Quantum Computation. *Physical Review Letters*, 95(1):010501, June 2005. DOI: 10.1103/PhysRevLett.95.010501.

[15] Titouan Carette, Marc de Visme, and Simon Perdrix. Graphical Language with Delayed Trace: Picturing Quantum Computing with Finite Memory. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–13. IEEE Computer Society, June 2021. ISBN 978-1-66544-895-6. DOI: 10.1109/LICS52264.2021.9470553.

[16] Jacques Carolan, Uttara Chakraborty, Nicholas C. Harris, Mihir Pant, Tom Baehr-Jones, Michael Hochberg, and Dirk Englund. Scalable feedback control of single photon sources for photonic quantum technologies. *Optica*, 6(3):335–340, March 2019. ISSN 2334-2536. DOI: 10.1364/OPTICA.6.000335.

[17] Jan M. Chaiken. Finite-particle representations and states of the canonical commutation relations. *Annals of Physics*, 42(1):23–80, March 1967. ISSN 0003-4916. DOI: 10.1016/0003-4916(67)90186-8.

[18] Giulio Chiribella, Giacomo Mauro D'Ariano, and Paolo Perinotti. Quantum Circuits Architecture. *Physical Review Letters*, 101(6):060401, August 2008. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.101.060401.

[19] Giulio Chiribella, Giacomo M. D'Ariano, and Paolo Perinotti. Theoretical framework for quantum networks. *Physical Review A*, 80(2):022339, August 2009. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.80.022339.

[20] Alexandre Clément, Nicolas Heurtel, Shane Mansfield, Simon Perdrix, and Benoît Valiron. $\mathrm{Lo_v}$-calculus: A graphical language for linear optical quantum circuits. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th international symposium on mathematical foundations of computer science (MFCS 2022)*, volume 241 of *Leibniz international proceedings in informatics (LIPIcs)*, page 35:1–35:16, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-256-3. DOI: 10.4230/LIPIcs.MFCS.2022.35.

[21] Bob Coecke and Ross Duncan. Interacting Quantum Observables. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 298–310, Berlin, Heidelberg, 2008. Springer. ISBN 978-3-540-70583-3. DOI: 10.1007/978-3-540-70583-3_25.

[22] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes*. Cambridge University Press, March 2017. ISBN 978-1-107-10422-8. DOI: 10.1017/9781316219317.

[23] Bob Coecke, Chris Heunen, and Aleks Kissinger. Categories of quantum and classical channels. *Quantum Information Processing*, 15(12):5179–5209, December 2016. ISSN 1573-1332. DOI: 10.1007/s11128-014-0837-4.

[24] Dan Cogan, Zu-En Su, Oded Kenneth, and David Gershoni. Deterministic generation of indistinguishable photons in a cluster state. *Nature Photonics*, 17(4):324–329, April 2023. ISSN 1749-4893. DOI: 10.1038/s41566-022-01152-2.

[25] N. Coste, D. A. Fioretto, N. Belabas, S. C. Wein, P. Hilaire, R. Frantzeskakis, M. Gundin, B. Goes, N. Somaschi, M. Morassi, A. Lemaître, I. Sagnes, A. Harouri, S. E. Economou, A. Auffeves, O. Krebs, L. Lanco, and P. Senellart. High-rate entanglement between a semiconductor spin and indistinguishable photons. *Nature Photonics*, 17(7):582–587, July 2023. ISSN 1749-4893. DOI: 10.1038/s41566-023-01186-0.

[26] Jacob P. Covey, Harald Weinfurter, and Hannes Bernien. Quantum networks with neutral atom processing nodes. *npj Quantum Information*, 9(1):1–12, September 2023. ISSN 2056-6387. DOI: 10.1038/s41534-023-00759-9.

[27] Vincent Danos, Elham Kashefi, and Prakash Panangaden. The measurement calculus. *Journal of the ACM*, 54(2):8–es, April 2007. ISSN 0004-5411. DOI: 10.1145/1219092.1219096.

[28] Niel de Beaudrap and Dominic Horsman. The ZX calculus is a language for surface code lattice surgery. *Quantum*, 4:218, January 2020. DOI: 10.22331/q-2020-01-09-218.

[29] Niel de Beaudrap, Xiaoning Bian, and Quanlong Wang. Fast and Effective Techniques for T-Count Reduction via Spider Nest Identities. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2020)*, volume 158 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:23, Dagstuhl, Germany,

2020. Schloss-Dagstuhl - Leibniz Zentrum für Informatik. ISBN 978-3-95977-146-7. DOI: 10.4230/LIPIcs.TQC.2020.11.

[30] Giovanni de Felice and Bob Coecke. Quantum Linear Optics via String Diagrams. In Stefano Gogioso and Matty Hoban, editors, *Proceedings 19th International Conference on Quantum Physics and Logic*, volume 394 of *Electronic Proceedings in Theoretical Computer Science*, pages 83–100, Wolfson College, Oxford, UK, 2023. Open Publishing Association. DOI: 10.4204/EPTCS.394.6.

[31] Giovanni de Felice, Razin A. Shaikh, Boldizsár Poór, Lia Yeh, Quanlong Wang, and Bob Coecke. Light-Matter Interaction in the ZXW Calculus. In Shane Mansfield, Benoit Valîron, and Vladimir Zamdzhiev, editors, *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, volume 384 of *Electronic Proceedings in Theoretical Computer Science*, pages 20–46. Open Publishing Association, July 2023. DOI: 10.4204/EPTCS.384.2.

[32] Elena Di Lavore, Alessandro Gianola, Mario Román, Nicoletta Sabadini, and Paweł Sobociński. A canonical algebra of open transition systems. In *Formal Aspects of Component Software: 17th International Conference, FACS 2021, Virtual Event, October 28–29, 2021, Proceedings*, pages 63–81. Springer-Verlag. ISBN 978-3-030-90635-1. DOI: 10.1007/978-3-030-90636-8_4.

[33] Elena Di Lavore, Giovanni de Felice, and Mario Román. Monoidal Streams for Dataflow Programming. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '22, pages 1–14, New York, NY, USA, August 2022. Association for Computing Machinery. ISBN 978-1-4503-9351-5. DOI: 10.1145/3531130.3533365.

[34] Ross Duncan. A graphical approach to measurement-based quantum computing. In Chris Heunen, Mehrnoosh Sadrzadeh, and Edward Grefenstette, editors, *Quantum Physics and Linguistics: A Compositional, Diagrammatic Discourse*, pages 50–89. Oxford University Press, February 2013. ISBN 978-0-19-964629-6. DOI: 10.1093/acprof:oso/9780199646296.001.0001.

[35] Ross Duncan and Simon Perdrix. Rewriting Measurement-Based Quantum Computations with Generalised Flow. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *Automata, Languages and Programming*, Lecture Notes in Computer Science, pages 285–296, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-14162-1. DOI: 10.1007/978-3-642-14162-1_24.

[36] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, June 2020. DOI: 10.22331/q-2020-06-04-279.

[37] A. I. Ekimov and A. A. Onushchenko. Quantum Size Effect in Three-Dimensional Microscopic Semiconductor Crystals. *JETP Letters*, 118(1):S15–S17, August 1981. ISSN 1090-6487. DOI: 10.1134/S0021364023130040.

[38] Fabian Ewert and Peter van Loock. 3/4-efficient bell measurement with passive linear optics and unentangled ancillae. *Physical Review Letters*, 113(14):140403, September 2014. DOI: 10.1103/PhysRevLett.113.140403.

[39] Giovanni de Felice, Alexis Toumi, and Bob Coecke. DisCoPy: Monoidal categories in python. 333:183–197. ISSN 2075-2180. DOI: 10.4204/EPTCS.333.13.

[40] Davide Ferrari, Angela Sara Cacciapuoti, Michele Amoretti, and Marcello Caleffi. Compiler Design for Distributed Quantum Computing. *IEEE Transactions on Quantum Engineering*, 2:1–20, 2021. ISSN 2689-1808. DOI: 10.1109/TQE.2021.3053921.

[41] Mercedes Gimeno-Segovia, Pete Shadbolt, Dan E. Browne, and Terry Rudolph. From Three-Photon Greenberger-Horne-Zeilinger States to Ballistic Universal Quantum Computation. *Physical Review Letters*, 115(2):020502, July 2015. DOI: 10.1103/PhysRevLett.115.020502.

[42] Grégoire de Gliniasty, Paul Hilaire, Pierre-Emmanuel Emeriau, Stephen C. Wein, Alexia Salavrakos, and Shane Mansfield. A spin-optical quantum computing architecture. 8:1423. DOI: 10.22331/q-2024-07-24-1423.

[43] W. P. Grice. Arbitrarily complete Bell-state measurement using only linear optical elements. *Physical Review A*, 84(4):042331, October 2011. DOI: 10.1103/PhysRevA.84.042331.

[44] Nicolas Halbwachs, Fabienne Lagnier, and Christophe Ratel. Programming and Verifying Real-Time Systems by Means of the Synchronous Data-Flow Language LUSTRE.

IEEE Trans. Softw. Eng., 18(9):785–793, September 1992. ISSN 0098-5589. DOI: 10.1109/32.159839.

[45] S. E. Harris, M. K. Oshman, and R. L. Byer. Observation of Tunable Optical Parametric Fluorescence. *Physical Review Letters*, 18(18):732–734, May 1967. DOI: 10.1103/PhysRevLett.18.732.

[46] Daniel Herr, Alexandru Paler, Simon J. Devitt, and Franco Nori. A local and scalable lattice renormalization method for ballistic quantum computation. *npj Quantum Information*, 4(1): 27, June 2018. ISSN 2056-6387. DOI: 10.1038/s41534-018-0076-0.

[47] Nicolas Heurtel. A complete graphical language for linear optical circuits with finite-photon-number sources and detectors. In Jörg Endrullis and Sylvain Schmitz, editors, *33rd EACSL Annual Conference on Computer Science Logic (CSL 2025)*, volume 326 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 38:1–38:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-362-1. DOI: 10.4230/LIPIcs.CSL.2025.38.

[48] Paul Hilaire, Théo Dessertaine, Boris Bourdoncle, Aurélie Denys, Grégoire de Gliniasty, Gerard Valentí-Rojas, and Shane Mansfield. Enhanced Fault-tolerance in Photonic Quantum Computing: Floquet Code Outperforms Surface Code in Tailored Architecture, October 2024. URL https://doi.org/10.48550/arXiv.2410.07065.

[49] Jiaxin Huang, Sarah Meng Li, Lia Yeh, Aleks Kissinger, Michele Mosca, and Michael Vasmer. Graphical CSS code transformation using ZX calculus. In Shane Mansfield, Benoit Valîron, and Vladimir Zamdzhiev, editors, *Proceedings of the Twentieth International Conference on Quantum Physics and Logic*, volume 384 of *Electronic Proceedings in Theoretical Computer Science*, pages 1–19. Open Publishing Association, 2023. DOI: 10.4204/EPTCS.384.1.

[50] H. Huet, P. R. Ramesh, S. C. Wein, N. Coste, P. Hilaire, N. Somaschi, M. Morassi, A. Lemaître, I. Sagnes, M. F. Doty, O. Krebs, L. Lanco, D. A. Fioretto, and P. Senellart. Deterministic and reconfigurable graph state generation with a single solid-state quantum emitter. 16(1):4337. ISSN 2041-1723. DOI: 10.1038/s41467-025-59693-3.

[51] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. A Complete Axiomatisation of the ZX-Calculus for Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 559–568, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209131.

[52] Emmanuel Jeandel, Simon Perdrix, and Renaud Vilmart. Diagrammatic Reasoning beyond Clifford+T Quantum Mechanics. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '18, pages 569–578, New York, NY, USA, July 2018. Association for Computing Machinery. ISBN 978-1-4503-5583-4. DOI: 10.1145/3209108.3209139.

[53] Liang Jiang, Jacob M. Taylor, Anders S. Sørensen, and Mikhail D. Lukin. Distributed quantum computation based on small quantum registers. *Physical Review A*, 76(6):062323, December 2007. DOI: 10.1103/PhysRevA.76.062323.

[54] Marc A. Kastner. Artificial Atoms. *Physics Today*, 46(1):24–31, 1993. ISSN 0031-9228. DOI: 10.1063/1.881393.

[55] Aleks Kissinger. Phase-free zx diagrams are css codes (...or how to graphically grok the surface code), April 2022. URL https://doi.org/10.48550/arXiv.2204.14038.

[56] Aleks Kissinger and John van de Wetering. Reducing the number of non-Clifford gates in quantum circuits. *Physical Review A*, 102(2):022406, August 2020. DOI: 10.1103/PhysRevA.102.022406.

[57] E. Knill, R. Laflamme, and G. J. Milburn. A scheme for efficient quantum computation with linear optics. *Nature*, 409(6816):46–52, January 2001. ISSN 1476-4687. DOI: 10.1038/35051009.

[58] Dexter Kozen and Alexandra Silva. Practical coinduction. *Mathematical Structures in Computer Science*, 27(7):1132–1152, October 2017. ISSN 0960-1295, 1469-8072. DOI: 10.1017/S0960129515000493.

[59] Mateusz Kupper, Richie Yeung, Boldizsár Poór, Alexis Toumi, William Cashman, and Giovanni de Felice. Optyx: A ZX-based python library for networked quantum architectures. URL https://doi.org/10.48550/arXiv.2512.09648.

[60] Elena Di Lavore, Giovanni de Felice, and Mario Román. Coinductive Streams in Monoidal Categories. *Logical Methods in Computer Science*, Volume 21, Issue 3:10759, August 2025. ISSN 1860-5974. DOI: 10.46298/lmcs-21(3:18)2025.

[61] Seok-Hyung Lee and Hyunseok Jeong. Graph-theoretical optimization of fusion-based graph state generation. *Quantum*, 7:1212, December 2023. DOI: 10.22331/q-2023-12-20-1212.

[62] Seung-Woo Lee, Kimin Park, Timothy C. Ralph, and Hyunseok Jeong. Nearly deterministic Bell measurement with multiphoton entanglement for efficient quantum-information processing. *Physical Review A*, 92(5):052324, November 2015. DOI: 10.1103/PhysRevA.92.052324.

[63] Karel Lemr, Antonin Cernoch, Jan Soubusta, Konrad Kieling, Jens Eisert, and Miloslav Dusek. Experimental implementation of the optimal linear-optical controlled phase gate. *Physical Review Letters*, 106(1):013602, January 2011. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.106.013602.

[64] Yuan Liang Lim, Almut Beige, and Leong Chuan Kwek. Repeat-Until-Success Linear Optics Distributed Quantum Computing. *Physical Review Letters*, 95(3):030505, July 2005. DOI: 10.1103/PhysRevLett.95.030505.

[65] Netanel H. Lindner and Terry Rudolph. Proposal for Pulsed On-Demand Sources of Photonic Cluster State Strings. *Physical Review Letters*, 103(11):113602, September 2009. DOI: 10.1103/PhysRevLett.103.113602.

[66] Matthias C. Löbl, Love A. Pettersson, Stefano Paesani, and Anders S. Sørensen. Transforming graph states via Bell state measurements, May 2024.

[67] Lars S. Madsen, Fabian Laudenbach, Mohsen Falamarzi Askarani, Fabien Rortais, Trevor Vincent, Jacob F. F. Bulmer, Filippo M. Miatto, Leonhard Neuhaus, Lukas G. Helt, Matthew J. Collins, Adriana E. Lita, Thomas Gerrits, Sae Woo Nam, Varun D. Vaidya, Matteo Menotti, Ish Dhand, Zachary Vernon, Nicolás Quesada, and Jonathan Lavoie. Quantum computational advantage with a programmable photonic processor. *Nature*, 606(7912):75–81, June 2022. ISSN 1476-4687. DOI: 10.1038/s41586-022-04725-x.

[68] D. Main, P. Drmota, D. P. Nadlinger, E. M. Ainley, A. Agrawal, B. C. Nichol, R. Srinivas, G. Araneda, and D. M. Lucas. Distributed quantum computing across an optical network link. *Nature*, 638(8050):383–388, February 2025. ISSN 1476-4687. DOI: 10.1038/s41586-024-08404-x.

[69] Krzysztof Malarz. Site percolation thresholds on triangular lattice with complex neighborhoods. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 30(12):123123, December 2020. ISSN 1054-1500. DOI: 10.1063/5.0022336.

[70] Tommy McElvanney and Miriam Backens. Complete flow-preserving rewrite rules for MBQC patterns with pauli measurements. In Stefano Gogioso and Matty Hoban, editors, *Proceedings 19th International Conference on Quantum Physics and Logic, Wolfson College, Oxford, UK, 27 June - 1 July 2022*, volume 394 of *Electronic Proceedings in Theoretical Computer Science*, pages 66–82. Open Publishing Association, 2023. DOI: 10.4204/EPTCS.394.5.

[71] Tommy McElvanney and Miriam Backens. Flow-preserving ZX-calculus rewrite rules for optimisation and obfuscation. In Shane Mansfield, Benoit Valîron, and Vladimir Zamdzhiev, editors, *Proceedings of the Twentieth International Conference on Quantum Physics and Logic, Paris, France, 17-21st July 2023*, volume 384 of *Electronic Proceedings in Theoretical Computer Science*, pages 203–219. Open Publishing Association, 2023. DOI: 10.4204/EPTCS.384.12.

[72] Mehdi Mhalla and Simon Perdrix. Graph States, Pivot Minor, and Universality of (X,Z)-measurements, February 2012. URL `https://doi.org/10.48550/arXiv.1202.6551`.

[73] Adam Miranowicz, Wiesław Leoński, and Nobuyuki Imoto. Quantum-optical states in finite-dimensional hilbert space. i. general formalism. In *Modern Nonlinear Optics*, pages 155–193. John Wiley & Sons, Ltd. ISBN 978-0-471-23147-9. DOI: 10.1002/0471231479.ch3.

[74] C. Monroe, R. Raussendorf, A. Ruthven, K. R. Brown, P. Maunz, L.-M. Duan, and J. Kim. Large Scale Modular Quantum Computer Architecture with Atomic Memory and Photonic Interconnects. *Physical Review A*, 89(2):022317, February 2014. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.89.022317.

[75] Keith R. Motes, Alexei Gilchrist, Jonathan P. Dowling, and Peter P. Rohde. Scalable Bo-

son Sampling with Time-Bin Encoding Using a Loop-Based Architecture. *Physical Review Letters*, 113(12):120501, September 2014. DOI: 10.1103/PhysRevLett.113.120501.

[76] Keith R. Motes, Alexei Gilchrist, Jonathan P. Dowling, and Peter P. Rohde. Scalable Boson Sampling with Time-Bin Encoding Using a Loop-Based Architecture. *Physical Review Letters*, 113(12):120501, September 2014. DOI: 10.1103/PhysRevLett.113.120501.

[77] C. B. Murray, D. J. Norris, and M. G. Bawendi. Synthesis and characterization of nearly monodisperse CdE (E = sulfur, selenium, tellurium) semiconductor nanocrystallites. *Journal of the American Chemical Society*, 115(19):8706–8715, September 1993. ISSN 0002-7863. DOI: 10.1021/ja00072a025.

[78] Maarten Van den Nest, Akimasa Miyake, Wolfgang Dür, and Hans J. Briegel. Universal resources for measurement-based quantum computation. *Physical Review Letters*, 97(15): 150504, October 2006. ISSN 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.97.150504.

[79] Kang Feng Ng and Quanlong Wang. A universal completion of the ZX-calculus, June 2017.

[80] Ramil Nigmatullin, Christopher J Ballance, Niel de Beaudrap, and Simon C Benjamin. Minimally complex ion traps as modules for quantum communication and computing. *New Journal of Physics*, 18(10):103028, October 2016. ISSN 1367-2630. DOI: 10.1088/1367-2630/18/10/103028.

[81] Harold Ollivier and Jean-Pierre Tillich. Description of a Quantum Convolutional Code. *Physical Review Letters*, 91(17):177902, October 2003. DOI: 10.1103/PhysRevLett.91.177902.

[82] Michał Oszmaniec and Daniel J. Brod. Classical simulation of photonic linear optics with lost particles. *New Journal of Physics*, 20(9):092002, September 2018. ISSN 1367-2630. DOI: 10.1088/1367-2630/aadfa8.

[83] Sahin Kaya Ozdemir, Adam Miranowicz, Masato Koashi, and Nobuyuki Imoto. Quantum-scissors device for optical state truncation: A proposal for practical realization. *Physical Review A*, 64(6):063818, November 2001. ISSN 1050-2947, 1094-1622. DOI: 10.1103/PhysRevA.64.063818.

[84] Stefano Paesani and Benjamin J. Brown. High-Threshold Quantum Computing by Fusing One-Dimensional Cluster States. *Physical Review Letters*, 131(12):120603, September 2023. DOI: 10.1103/PhysRevLett.131.120603.

[85] Jian-wei Pan and Anton Zeilinger. Greenberger-Horne-Zeilinger-state analyzer. *Physical Review A*, 57(3):2208–2211, March 1998. DOI: 10.1103/PhysRevA.57.2208.

[86] Brendan Pankovich, Alex Neville, Angus Kan, Srikrishna Omkar, Kwok Ho Wan, and Kamil Brádler. Flexible entangled state generation in linear optics, October 2023. URL `https://doi.org/10.48550/arXiv.2310.06832`.

[87] David T. Pegg, Lee S. Phillips, and Stephen M. Barnett. Optical State Truncation by Projection Synthesis. *Physical Review Letters*, 81(8):1604–1606, August 1998. ISSN 0031-9007. DOI: 10.1103/PhysRevLett.81.1604.

[88] John Power and Edmund Robinson. Premonoidal categories and notions of computation. *Mathematical. Structures in Comp. Sci.*, 7(5):453–468, October 1997. ISSN 0960-1295. DOI: 10.1017/S0960129597002375.

[89] Boldizsár Poór, Razin A. Shaikh, and Quanlong Wang. ZX-calculus is complete for finite-dimensional hilbert spaces. 426:127–158. ISSN 2075-2180. DOI: 10.4204/EPTCS.426.5.

[90] Robert Prevedel, Philip Walther, Felix Tiefenbacher, Pascal Böhi, Rainer Kaltenbaek, Thomas Jennewein, and Anton Zeilinger. High-speed linear optics quantum computing using active feed-forward. *Nature*, 445(7123):65–69, January 2007. ISSN 0028-0836, 1476-4687. DOI: 10.1038/nature05346.

[91] Michael Reck, Anton Zeilinger, Herbert J. Bernstein, and Philip Bertani. Experimental realization of any discrete unitary operator. *Physical Review Letters*, 73(1):58–61, July 1994. DOI: 10.1103/PhysRevLett.73.58.

[92] Benjamin Rodatz, Boldizsár Poór, and Aleks Kissinger. Floquetifying stabiliser codes with distance-preserving rewrites. URL `https://doi.org/10.48550/arXiv.2410.17240`.

[93] Benjamin Rodatz, Boldizsár Poór, and Aleks Kissinger. Fault Tolerance by Construction, June 2025. URL `https://doi.org/10.48550/arXiv.2506.17181`.

[94] Mario Román. Open Diagrams via Coend Calculus. In David I. Spivak and Jamie Vicary, editors, *Proceedings of the 3rd Annual International Applied Category Theory Conference 2020, ACT 2020, Cambridge, USA, 6-10th July 2020*, volume 333 of *Electronic Proceedings*

in Theoretical Computer Science, pages 65–78. Open Publishing Association, 2020. DOI: 10.4204/EPTCS.333.5.

[95] Mario Román and Paweł Sobociński. String diagrams for premonoidal categories. Volume 21, Issue 2. ISSN 1860-5974. DOI: 10.46298/lmcs-21(2:9)2025.

[96] R. Rossetti, S. Nakahara, and L. E. Brus. Quantum size effects in the redox potentials, resonance Raman spectra, and electronic spectra of CdS crystallites in aqueous solution. *The Journal of Chemical Physics*, 79(2):1086–1088, July 1983. ISSN 0021-9606. DOI: 10.1063/1.445834.

[97] I. Schwartz, D. Cogan, E. R. Schmidgall, Y. Don, L. Gantz, O. Kenneth, N. H. Lindner, and D. Gershoni. Deterministic generation of a cluster state of entangled photons. *Science*, 354 (6311):434–437, October 2016. DOI: 10.1126/science.aah4758.

[98] Will Simmons. Relating measurement patterns to circuits via pauli flow. In Chris Heunen and Miriam Backens, editors, *Proceedings 18th International Conference on Quantum Physics and Logic*, volume 343 of *Electronic Proceedings in Theoretical Computer Science*, pages 50–101, Gdansk, Poland, 2021. Open Publishing Association. DOI: 10.4204/EPTCS.343.4.

[99] Stasja Stanisic, Noah Linden, Ashley Montanaro, and Peter S. Turner. Generating entanglement with linear optics. *Physical Review A*, 96(4):043861, October 2017. DOI: 10.1103/PhysRevA.96.043861.

[100] Shinichi Sunami, Shiro Tamiya, Ryotaro Inoue, Hayata Yamasaki, and Akihisa Goban. Scalable Networking of Neutral-Atom Qubits: Nanofiber-Based Approach for Multiprocessor Fault-Tolerant Quantum Computers. *PRX Quantum*, 6(1):010101, February 2025. DOI: 10.1103/PRXQuantum.6.010101.

[101] Philip Thomas, Leonardo Ruscio, Olivier Morin, and Gerhard Rempe. Efficient generation of entangled multiphoton graph states from a single atom. *Nature*, 608(7924):677–681, August 2022. ISSN 1476-4687. DOI: 10.1038/s41586-022-04987-5.

[102] Alexis Toumi, Giovanni de Felice, and Richie Yeung. DisCoPy for the quantum computer scientist. URL `https://doi.org/10.48550/arXiv.2205.05190`.

[103] Alex Townsend-Teague, Julio Magdalena de la Fuente, and Markus Kesselring. Floquetifying the colour code. In Shane Mansfield, Benoit Valîron, and Vladimir Zamdzhiev, editors, *Proceedings of the Twentieth International Conference on Quantum Physics and Logic*, volume 384 of *Electronic Proceedings in Theoretical Computer Science*, pages 265–303. Open Publishing Association, 2023. DOI: 10.4204/EPTCS.384.14.

[104] Renaud Vilmart. 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)*, pages 1–10, June 2019. DOI: 10.1109/LICS.2019.8785765.

[105] John van de Wetering, Richie Yeung, Tuomas Laakkonen, and Aleks Kissinger. Optimal compilation of parametrised quantum circuits. 9:1828. DOI: 10.22331/q-2025-08-27-1828.

[106] Mark M. Wilde. Quantum-shift-register circuits. *Physical Review A*, 79(6):062325, June 2009. DOI: 10.1103/PhysRevA.79.062325.

[107] Patrick Yard, Alex E. Jones, Stefano Paesani, Alexandre Maïnos, Jacob F. Bulmer, and Anthony Laing. On-chip quantum information processing with distinguishable photons. 132 (15):150602. DOI: 10.1103/PhysRevLett.132.150602.

[108] Guilherme Luiz Zanin, Maxime J. Jacquet, Michele Spagnolo, Peter Schiansky, Irati Alonso Calafell, Lee A. Rozema, and Philip Walther. Fiber-compatible photonic feed-forward with 99% fidelity. *Optics Express*, 29(3):3425–3437, February 2021. ISSN 1094-4087. DOI: 10.1364/OE.409867.

[109] Hezi Zhang, Anbang Wu, Yuke Wang, Gushu Li, Hassan Shapourian, Alireza Shabani, and Yufei Ding. OneQ: A Compilation Framework for Photonic One-Way Quantum Computation. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–14, Orlando FL USA, June 2023. ACM. ISBN 9798400700958. DOI: 10.1145/3579371.3589047.

[110] Felix Zilk, Korbinian Staudacher, Tobias Guggemos, Karl Fürlinger, Dieter Kranzlmüller, and Philip Walther. A compiler for universal photonic quantum computers. In *2022 IEEE/ACM Third International Workshop on Quantum Computing Software (QCS)*, pages 57–67, November 2022. DOI: 10.1109/QCS56647.2022.00012.

A Rewrite rules used in the paper

This section contains a selection of rewrite rules used in the paper.

ZX calculus The axioms of the ZX calculus used in the paper are shown below.

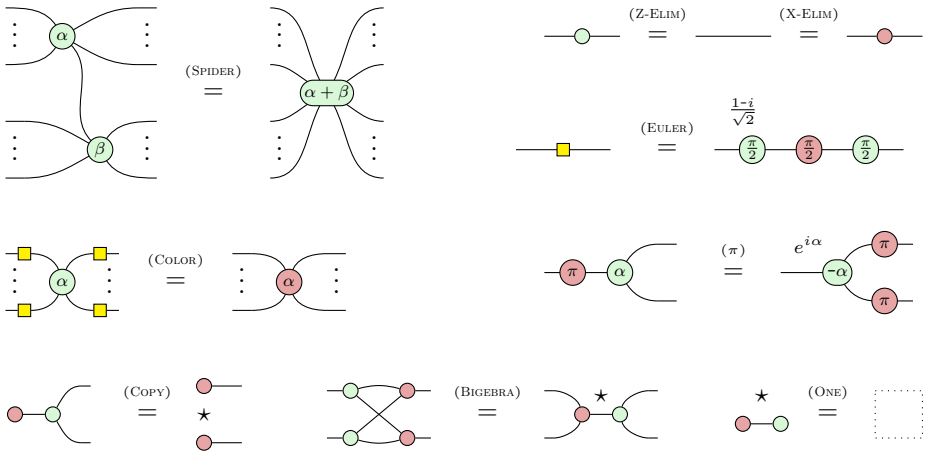
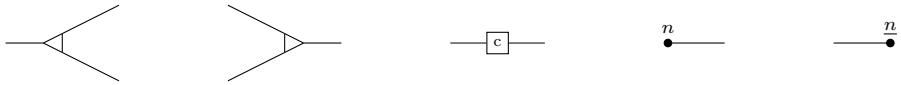


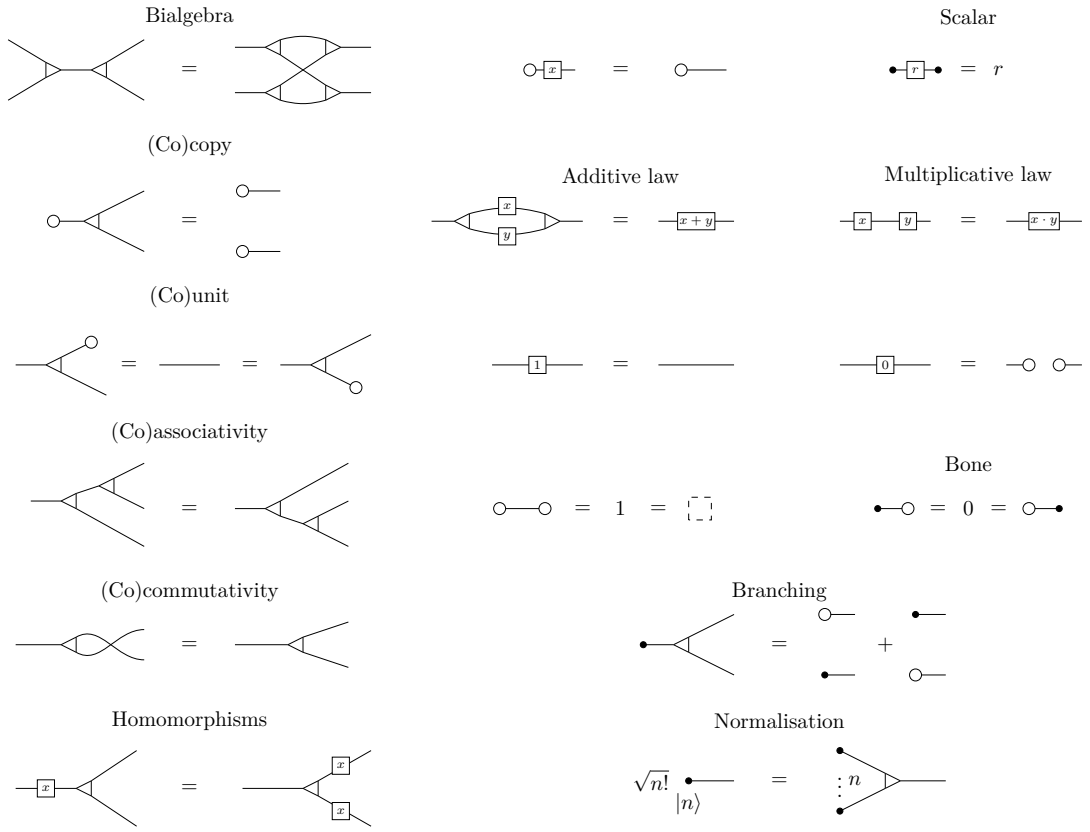
Figure 3: Axioms of the ZX calculus.

QPath We review the axioms of the QPath calculus [30], which are used in Section B. Diagrams in **QPath** are generated by:



for all $n \in \mathbb{N}$ and $c \in \mathbb{C}$. The **QPath** calculus admits the following graphical rewrite rules. Additionally, all rules hold under transposition of the linear maps, which is represented by horizontal

reflection of the diagrams.



Dual-rail The following rewrite rules relate to dual-rail encoding.

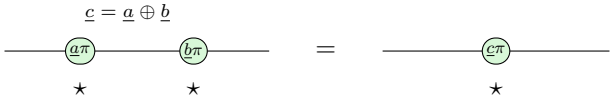
Rewrite 1.

Rewrite 2.

Rewrite 3.

Channel The following rewrite rules relate to the `Channel` construction.

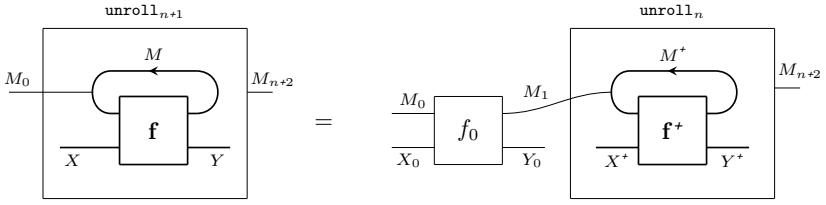
Rewrite 4.



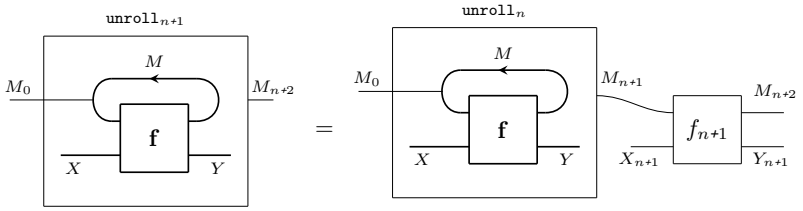
Rewrite 5.

Stream The following rewrite rules relate to the `Stream` construction.

Rewrite 6.



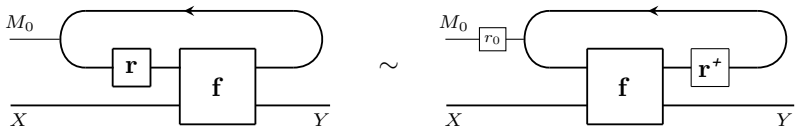
Rewrite 7.



Rewrite 8. *For any $r : M_0' \to M_0$ and $\mathbf{f} : X \to Y$ with initial memory M_0,*

$$\texttt{unroll}_n(\texttt{fby}(r, \mathbf{f})) = \texttt{unroll}_n(\texttt{fby}_{M_1}(\texttt{id}_{M_1}, \mathbf{f})) \circ (r \otimes \texttt{id}_{X_0})$$

Rewrite 9. *For any sequence of isometries $r_t : M_t \to M_{t+1}$ and channels $f_t : M_{t+1} \otimes X_t \to M_{t+1} \otimes Y_t$, the following streams are observationally equal:*

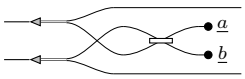


Rewrite 10. *If $\mathbf{f} = \texttt{fby}_M(A + B, \mathbf{g})$ then $\texttt{unroll}_n(\mathbf{f}) = A\,\texttt{unroll}_n(\mathbf{g}) + B\,\texttt{unroll}_n(\mathbf{g})$*

B Kraus decomposition of Type-I fusion

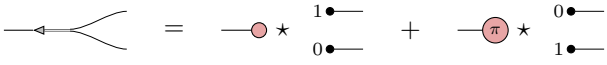
To prove Proposition 5.2, we derive diagrams in **QPath** for each measurement outcome of the linear optical circuit implementing Type I fusion:

$D^{\underline{a},\underline{b}}$:=



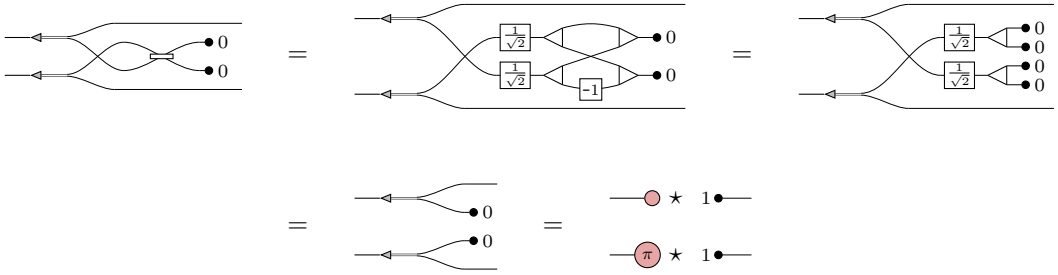
Because the input to $D^{\underline{a},\underline{b}}$ is two dual rail qubits and hence at most two photons, and no photons are created in this process, we can restrict our attention to only measurement outcomes observing at most two photons.

From the definitions of the dual-rail encoding Rewrite 1, we can define the following decomposition:



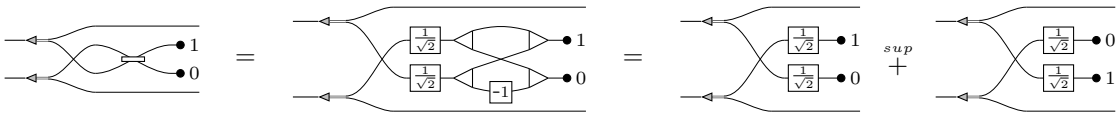
(27)

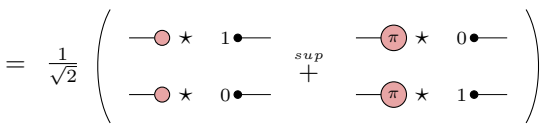
Using this, we can now prove a set of lemmas for the different cases of $D^{\underline{a},\underline{b}}$. First, $D^{\underline{a}=0,\underline{b}=0}$ evaluates to:

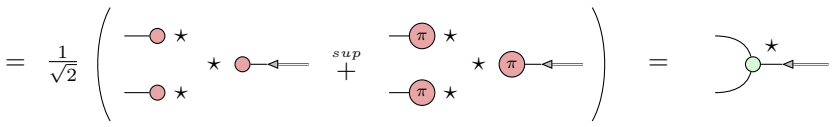


(28)

Continuing on, we compute $D^{\underline{a}=1,\underline{b}=0}$:

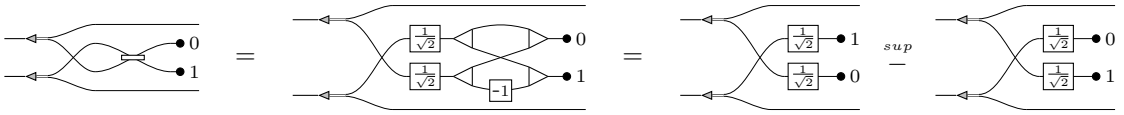




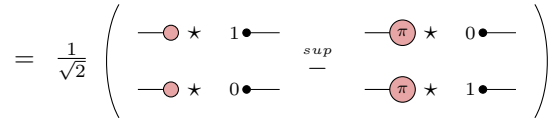


(29)

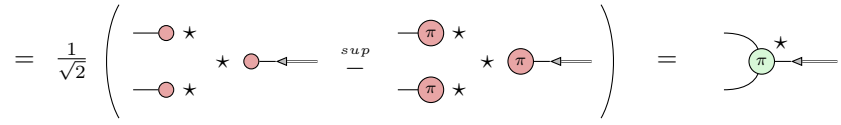
Similarly, $D^{\underline{a}=0,\underline{b}=1}$ computes to:



(30)

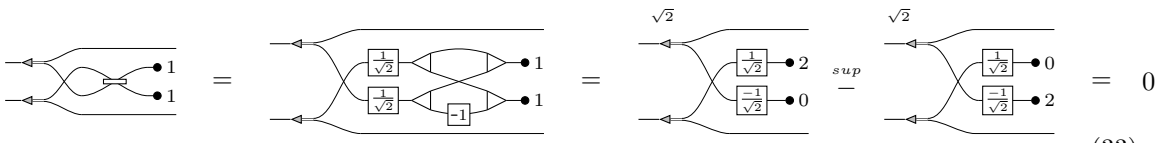


(31)



(32)

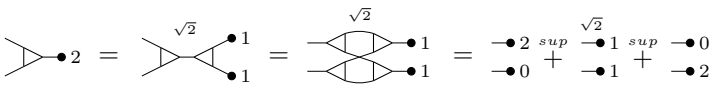
We can show that $D^{\underline{a}=1,\underline{b}=1}$ evaluates to:



(33)

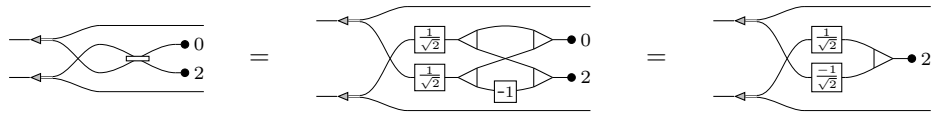
This means that the probability of observing $D^{\underline{a}=1,\underline{b}=1}$ is zero which is due to the Hong-Ou-Mandel effect.

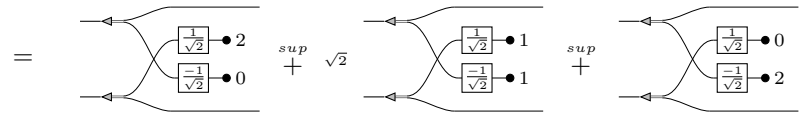
For the last two cases of $D^{\underline{a},\underline{b}}$, we use the following equality:



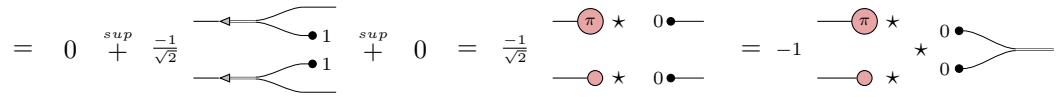
(34)

Proceeding with the calculations, we compute $D^{\underline{a}=0,\underline{b}=2}$ as follows:

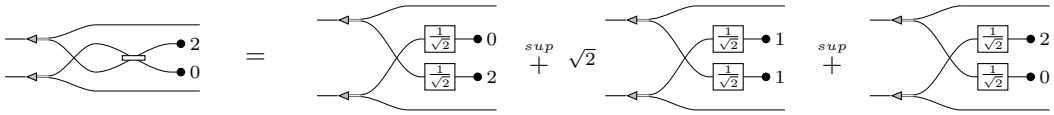




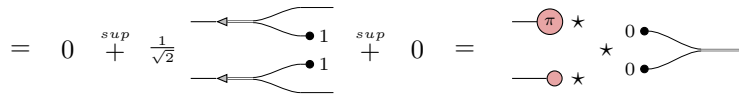
(35)



and finally, $D^{\underline{a}=2,\underline{b}=0}$ evaluates to:



(36)



To realize $D^{\underline{s},\underline{k}}$, we determine the action of a classical function from measurement outcomes to the Boolean variable indicators of success ($\underline{s}$) and correction ($\underline{k}$):

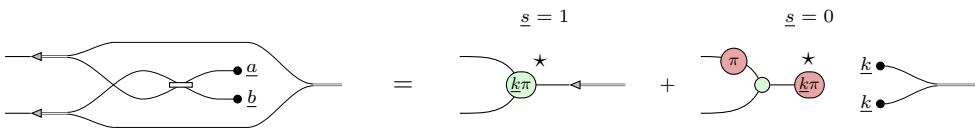
$\underline{a}$	$\underline{b}$	$\underline{s}$	$\underline{k}$
0	0	0	1
1	0	1	0
0	1	1	1
2	0	0	0
0	2	0	0

For three of the five possible values of $(\underline{s},\underline{k})$, the original measurement outcomes $(\underline{a},\underline{b})$ can be identified. For the case $(\underline{s}=0,\underline{k}=1)$, it happens that $D^{\underline{a}=2,\underline{b}=0} = D^{\underline{a}=0,\underline{b}=2}$ up to a global phase which is thereafter eliminated upon invoking the CPM construction. Because of this, in mixed-state quantum mechanics we have

$$D^{\underline{s}=0,\underline{k}=1} = D^{\underline{a}=2,\underline{b}=0} + D^{\underline{a}=0,\underline{b}=2} = 2\ D^{\underline{a}=2,\underline{b}=0} \tag{37}$$

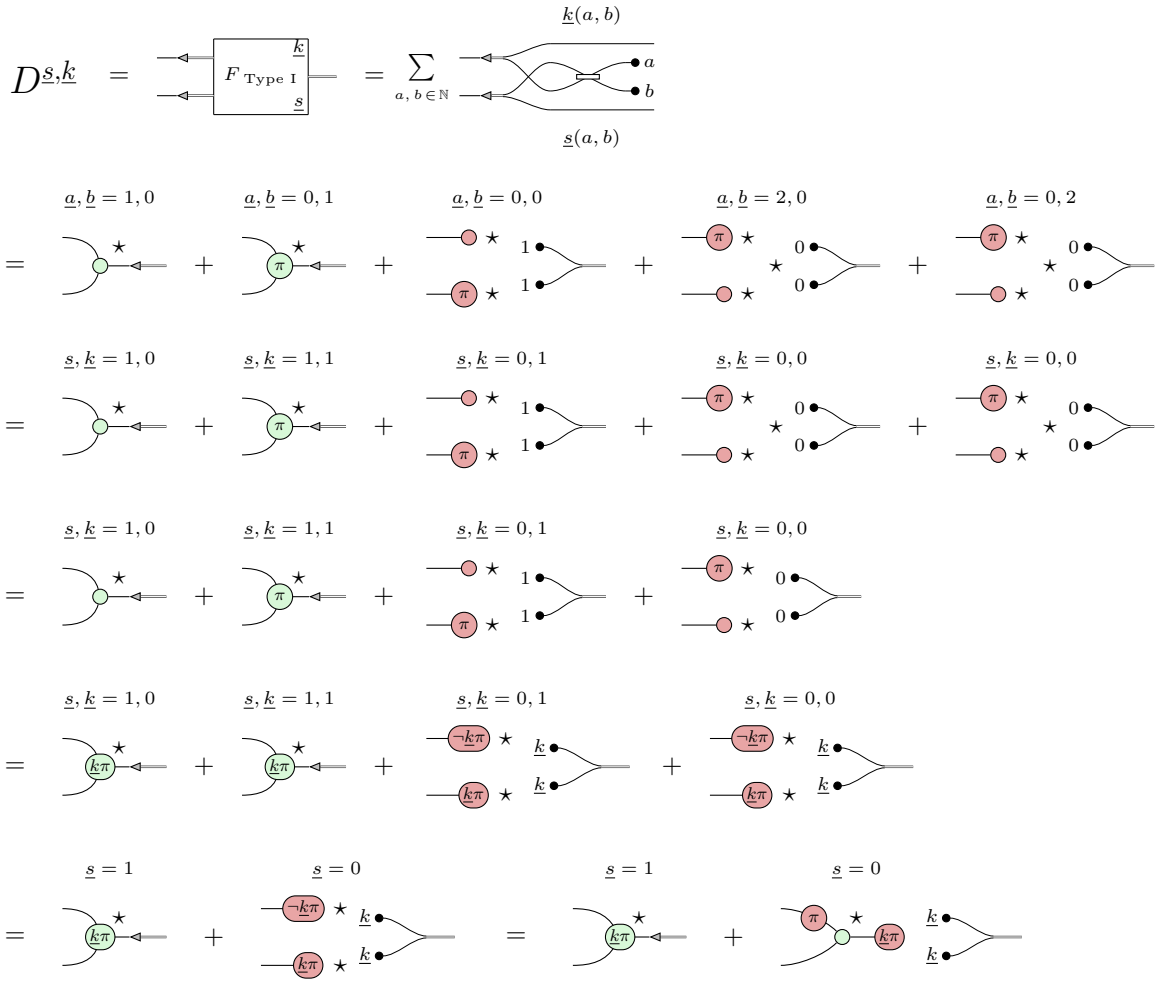
Therefore, $D^{\underline{s},\underline{k}}$ is a non-destructive measurement that is a sum of four terms, one for each possible value of $(\underline{s},\underline{k})$. By performing a mixed sum over all possible measurement outcomes of $D^{\underline{a},\underline{b}}$, and quotienting by $\underline{s}$ and $\underline{k}$, we obtain the proposition.

Proposition 5.2. *The following equation holds in the CP interpretation:*



after coarse-graining of the measurement operator by the equations $\underline{s} = \underline{a} \oplus \underline{b}$ *and* $\underline{k} = \underline{s}\underline{b} + \neg\underline{s}(1 - \frac{\underline{a}+\underline{b}}{2})$. *Here,* s *is the Boolean value of success and* k *is the Pauli measurement error.*

Proof.

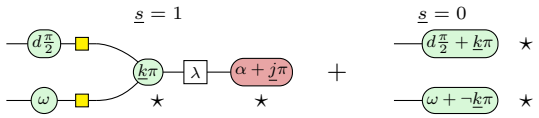


□

C Proof of characterization

In section, we prove Proposition 5.7, characterizing all fusion measurements with green failure and Pauli error.

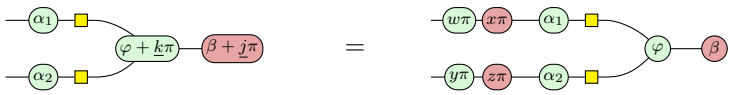
Proposition 5.7. *Any fusion measurement with green failure and Pauli error has the following form:*



(20)

for a measurement plane $\lambda \in \{\mathrm{YZ}, \mathrm{XZ}, \mathrm{XY}\}$*, angles* $\alpha, \omega \in [0, 2\pi)$*, and a choice of Clifford parameter* $d \in \{0, 1, 2, 3\}$*.*

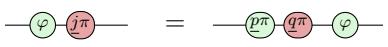
We are interested in fusions with green failure satisfying the following equation:



(38)

for some $w, x, y, z \in \{0, 1\}$.

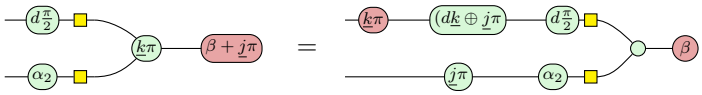
First note that in order for the $\underline{j}\pi$ error to commute to the inputs as a Pauli error, φ must satisfy the commutation relation



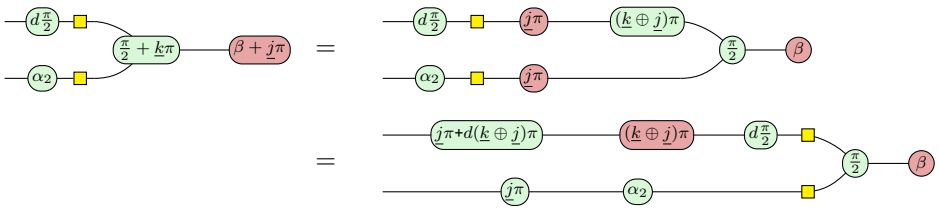
(39)

for some $\underline{p}, \underline{q} \in \{0, 1\}$. That is, φ must be a Clifford phase. Therefore, necessarily $\varphi = v\frac{\pi}{2}$ for some $v \in \{0, 1, 2, 3\}$. We thus consider two cases: (i) v is odd and (ii) v is even.

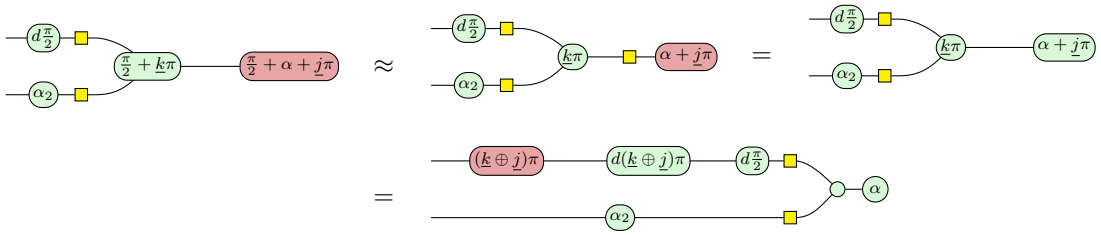
If v is even, we can take $\varphi = 0$; the $\varphi = \pi$ case is redundant because of the random error $\underline{k}\pi$. In this case, the $\underline{j}\pi$ error induces Z errors on both input qubits. In order to correct $\underline{k}\pi$ error, one of α_1 or α_2 must be Clifford. Without loss of generality, we assume α_1 is a Clifford phase, i.e. $\alpha_1 = d\frac{\pi}{2}$, and the result follows with $\lambda = \mathrm{XZ}$.



If v is odd, we can take $\varphi = \frac{\pi}{2}$; the $\varphi = -\frac{\pi}{2}$ case is redundant because of the random error $\underline{k}$. Then, the $\underline{j}\pi$ error induces a Y error when commuting with φ. Since $Y = XZ$, this induces an Z error on both input qubits and an X error which merges with the $\underline{k}\pi$ error. In order to correct the resulting $(\underline{k} \oplus \underline{j})\pi$ error, either α_1 or α_2 must be a an integer multiple of $\frac{\pi}{2}$. Without loss of generality we can set $\alpha_1 = d\frac{\pi}{2}$, and the result follows with $\lambda = \mathrm{YZ}$.

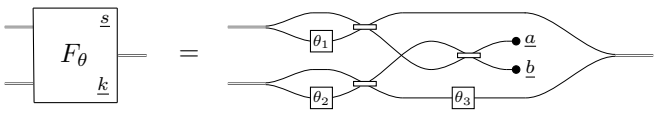


The third case, where $\lambda = \mathrm{XY}$ is obtained when $\beta = \frac{\pi}{2} + \alpha$ for some angle α and $\varphi = \frac{\pi}{2}$. This is technically an instance of the case where v is odd, but we distinguish it from the other cases because the errors are different, as they are induced by a single-qubit measurement in the XY plane.



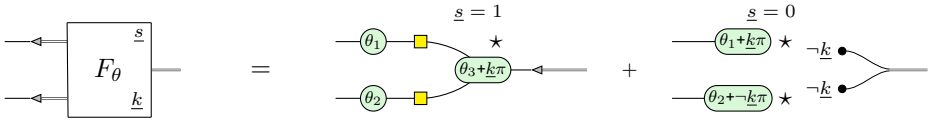
D Probability of success of fusions with green failure

Recall the circuit for fusion measurements with green failure, parametrised by three phases:



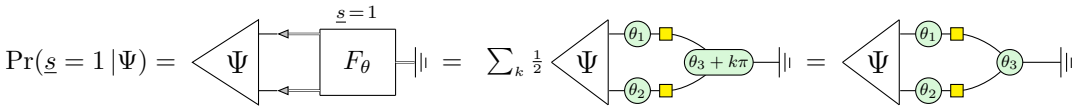
(40)

Recall from Proposition 5.2, that the behaviour of Type I fusion measurements on dual-rail encoded qubits can be described by a sum of ZX diagrams. A similar equation holds for the circuit defined above.

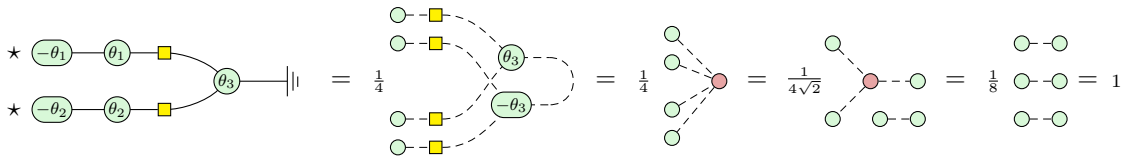


(41)

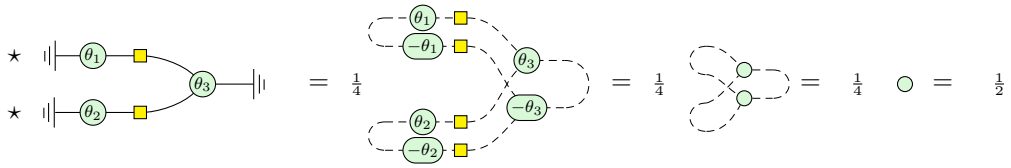
where $\underline{s} = \underline{a} \oplus \underline{b}$ and $\underline{k} = \underline{s}\underline{b} + \neg\underline{s}(1 - \frac{\underline{a}+\underline{b}}{2})$. The two diagrams above represent the action of fusion in case of success and failure, respectively. The *probability of success* for an input state Ψ is obtained by taking the *trace* of the success diagram, and discarding the classical output $\underline{k}$, which corresponds to summing over its possible values.



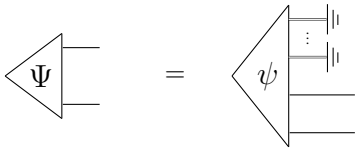
Note that the probability will usually depend on the input state Ψ. For example, we may engineer input states for which the fusion 'always succeeds'.



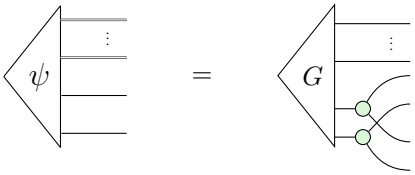
The calculation above uses dotted wires to represent ZX diagrams in the standard (rather than the CP) interpretation. Replacing the first input with a green $\theta_1 + \pi$ spider we obtain an example where the fusion 'always fails'. If the input state is the completely mixed state on two qubits, the probability is found to be $\frac{1}{2}$ by the following derivation.



By the purification theorem, a general mixed state Ψ can be expressed in terms of a pure state ψ on a larger space:



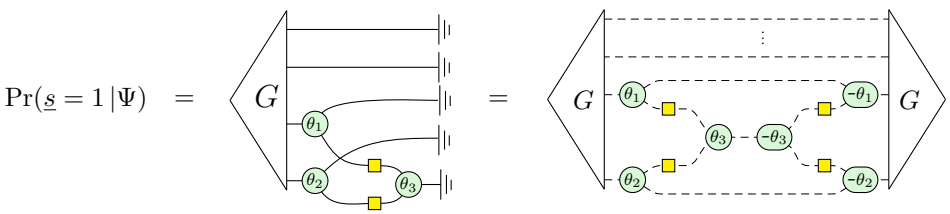
The dependence of the probability of success on the input can be avoided if we assume that the inputs are unmeasured qubits in a graph state. Then the state ψ has the following form:

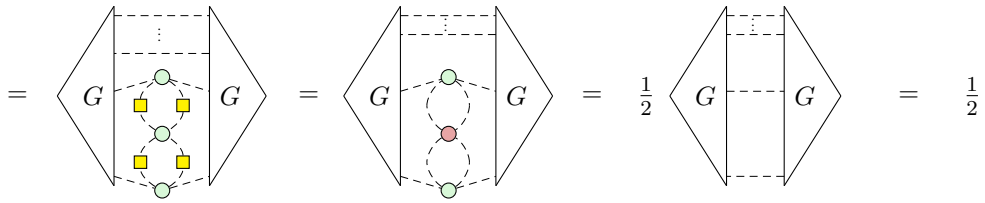


where $|G\rangle$ is a graph state.

Proposition D.1. *When the inputs are unmeasured qubits in a graph state $|G\rangle$, the success probability of any fusion with green failure is $\frac{1}{2}$.*

Proof.



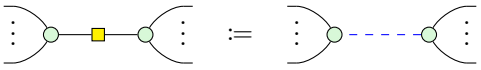


□

The action of fusion on unmeasured qubits in a graph state may be seen as a non-demolition measurement, obtained by composing the fusion operation with Z spiders as above. It is useful to write this operation as a classical probability distribution over causal maps, as follows.

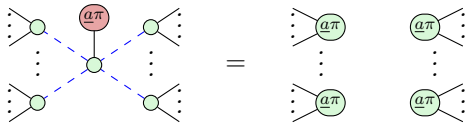
E Proof of flow preserving rewrites

In this appendix, we prove the results about preservation of Pauli flow of Section 6. We use an alternative notation to simplify the diagrams, and replace a Hadamard between two spiders by a blue dashed edge, as illustrated below.

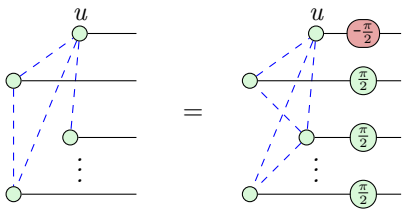


Both the blue edge notation and the Hadamard box can always be translated back into spiders when necessary. We refer to the blue edge as a Hadamard edge.

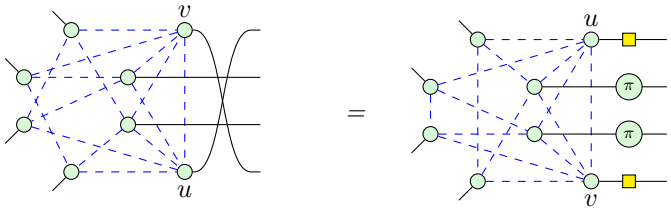
Lemma E.1 (Copy)**.** *Copying preserves the existence of Pauli flow [98, Lemma D.6]. Graphically, this corresponds to the copy rule of the ZX calculus [71, Lemmas 2.7 and 2.8]:*



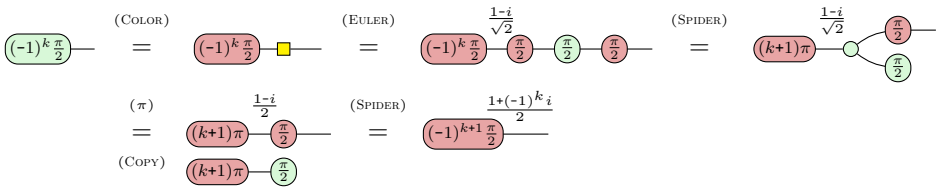
Lemma E.2 (Local Complementation)**.** *Local complementation about a vertex u preserves the existence of Pauli flow [98, Lemma D.12]. In the ZX calculus, this rule is also called local complementation, and is given as follows [71, Lemma 2.10]:*



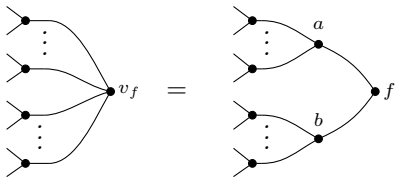
Lemma E.3 (Pivot)**.** *Pivoting about an edge (u, v) preserves the existence of Pauli flow [98, Lemma D.21]. In the ZX calculus, this rule is also called pivoting, and is given as the following rewrite rule [71, Lemma 2.11]:*



Lemma E.4 (State Change)**.**

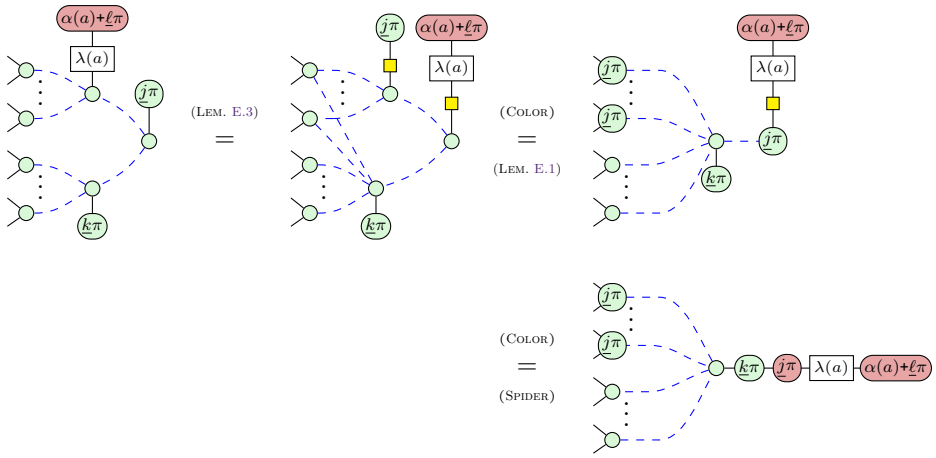


Proposition 6.12 (X-fusion)**.** *The following open graph rewrite preserves the existence of Pauli flow:*



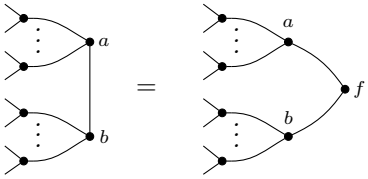
where $\lambda(f) = \lambda(b) = X$*,* $\lambda(a) = \lambda(v_f)$*, and* $\alpha(a) = \alpha(v_f)$*.*

Proof.



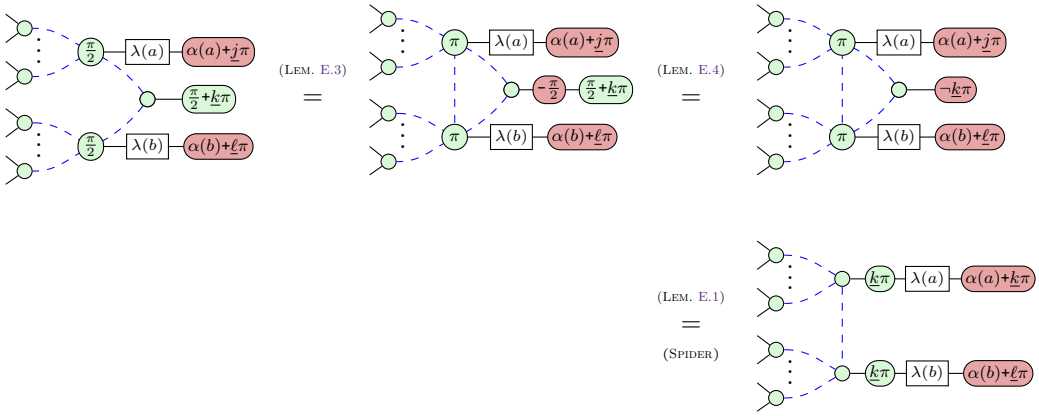
Since each of the rewrites preserves the existence of Pauli flow, the additional errors appearing above can always be corrected, but we have given them here for completeness. □

Proposition 6.13 (Y-fusion)**.** *The following open graph rewrite preserves the existence of Pauli flow:*



where $\lambda(f) = Y$*,* $c(a) = c(b) = 0$ *on the left and* $c(a) = c(b) = 1$ *on the right-hand side.*

Proof.

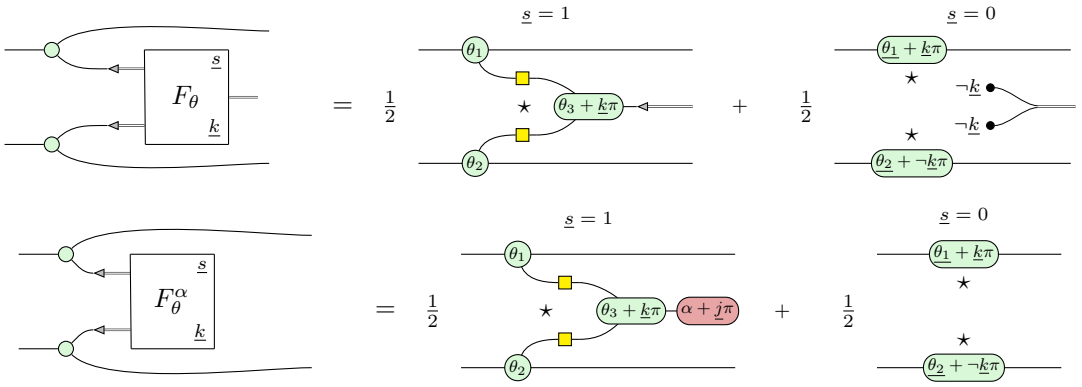


□

F Proofs of repeat-until-success

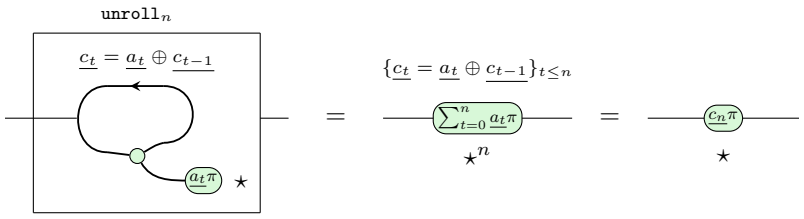
We prove the results of Section 7.1. To this end, we will need the following lemmas.

Lemma F.1. *Let F_θ be the optical circuit defined above and F_θ^α the same optical circuit followed by a measurement in the* XZ *plane of angle α. Then we have:*

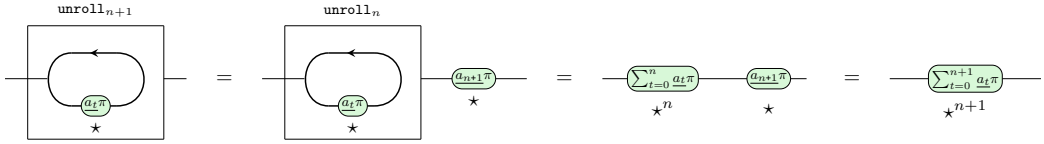


Proof. This follows from Proposition 5.5 and $[\![\star]\!]_{CP} = \frac{1}{2}$. □

Lemma F.2.

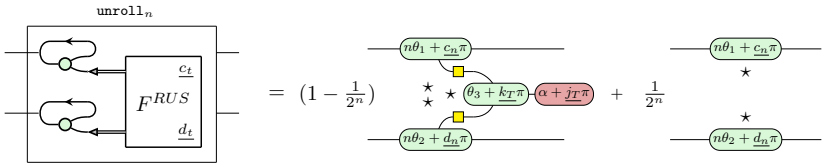


Proof. We prove this by induction. The statement for $n = 0$ is easy to show. Then using Definition 4.3, we have



Finally, the last equality holds after coarse-graining the measurement outcomes a_t. Indeed there are 2^n possible measurement outcomes for the list a_t, but they together induce one uniformly random bitflip $c_n\pi$ after n time-steps. The statement then follows from Rewrite 4. □

Theorem 7.2. *Any fusion with green failure can be boosted with a repeat-until-success protocol. More precisely, the following holds for $n \geq 1$:*

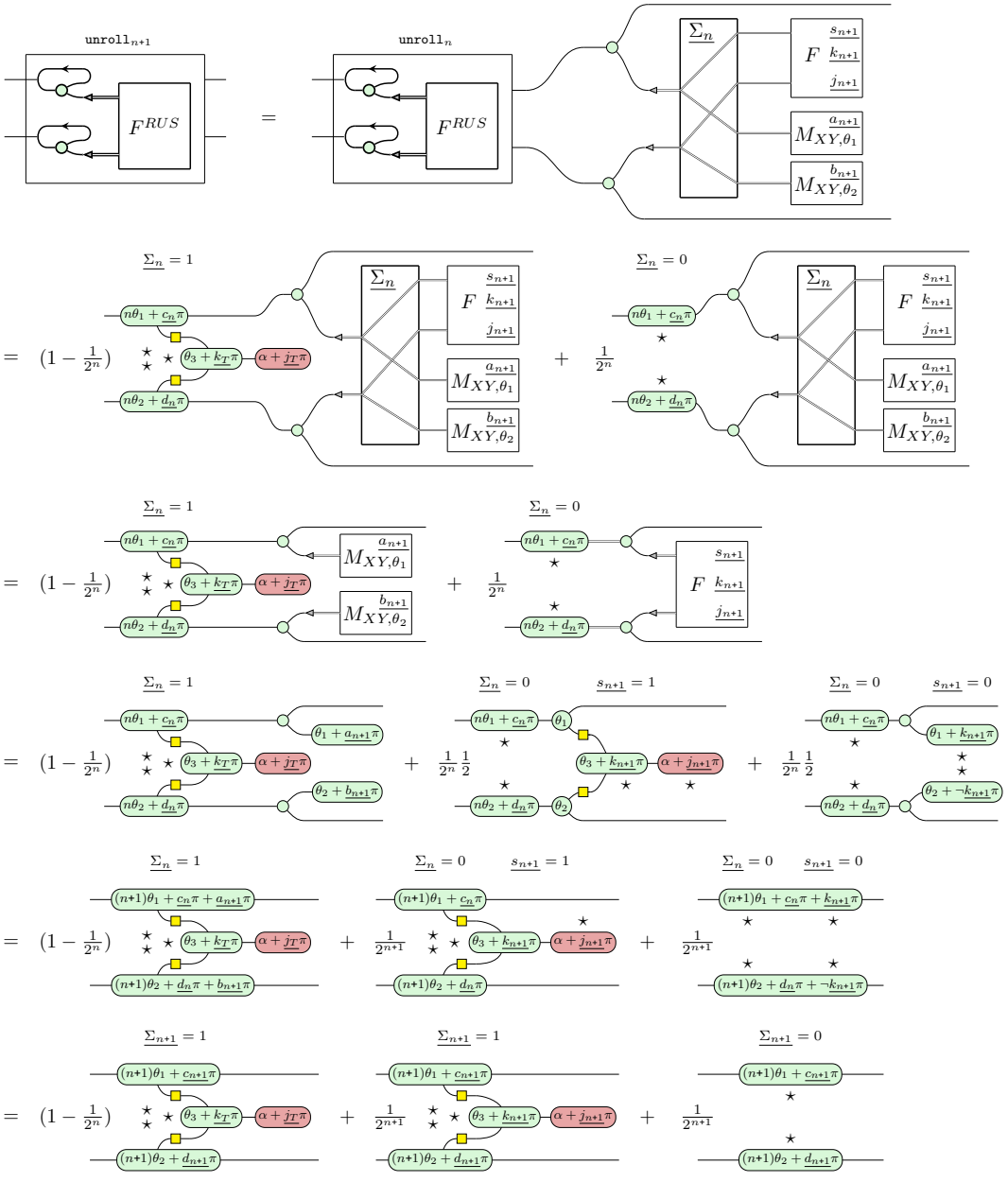


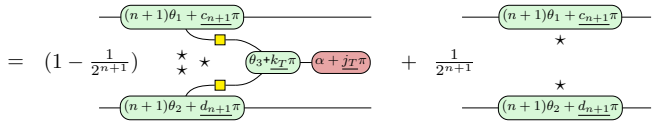
where T is the time of the first successful fusion (if it exists) and:

$$\Sigma_t \coloneqq \sum_{i=0}^{t} s_t \qquad c_t = c_{t-1} \oplus (\neg\Sigma_t)k_t \oplus \Sigma_{t-1}a_t \qquad d_t = d_{t-1} \oplus (\neg\Sigma_t)(\neg k_t) \oplus \Sigma_{t-1}b_t$$

with $s_0 = 0$, $c_0 = d_0 = 1$.

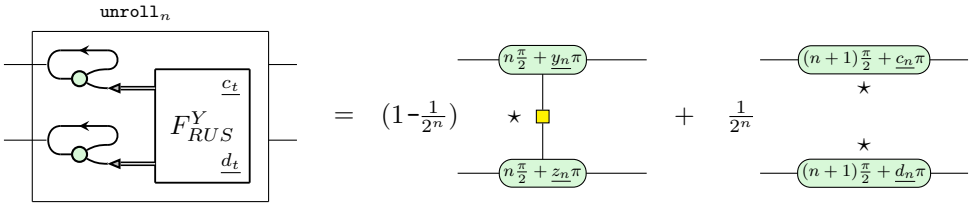
Proof. We prove this inductively. The $n = 1$ case follows from Lemma F.1. Then, given the statement for n, we show it for $n + 1$:





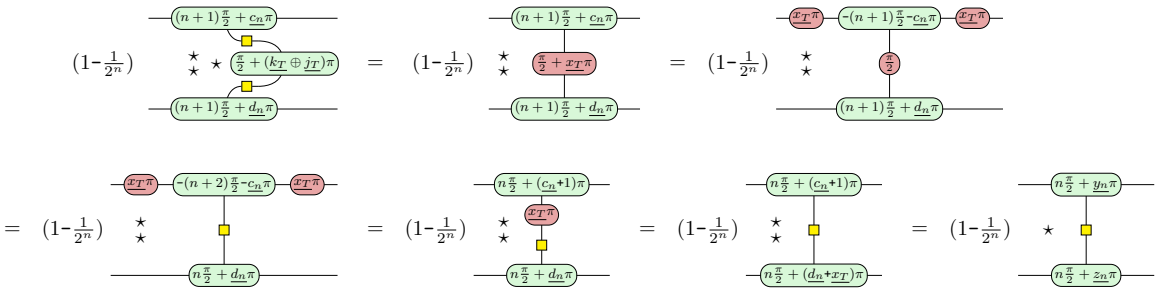
□

Corollary 7.3 (Y fusion RUS)**.** *For $n \geq 1$ we have:*



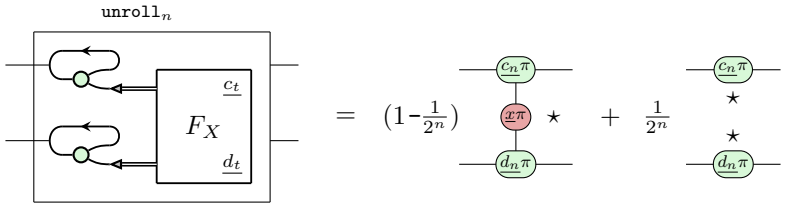
where $z_t = (k_T \oplus j_T) \oplus c_t$ and $y_t = (k_T \oplus j_T) \oplus d_t$ if $T < t$ and $y_t = (k_T \oplus j_T) \oplus \neg c_t$ if $T = t$.

Proof. Follows from Theorem 7.2 and:



where $x_T = k_T \oplus j_T$. □

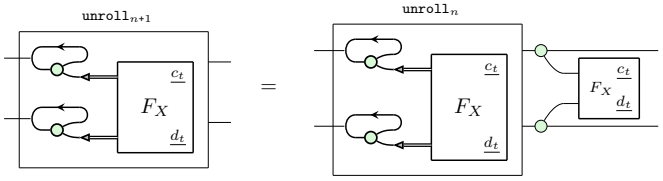
Proposition 7.4 (Boosted X Fusion)**.** *For $n \geq 1$ we have:*

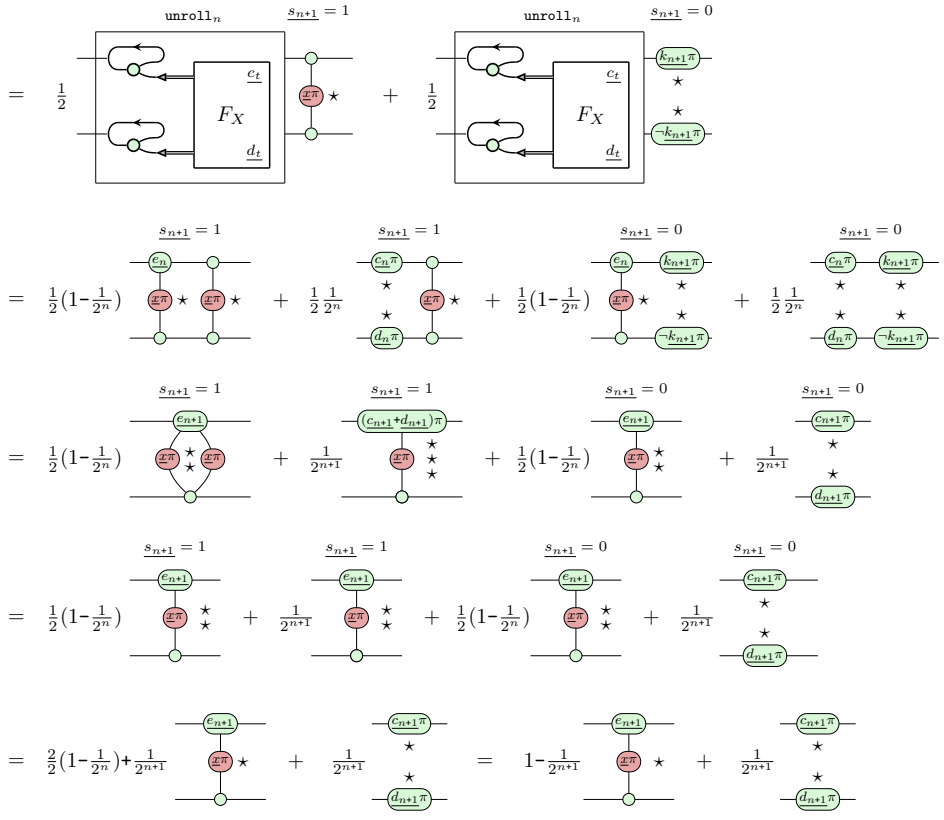


where $c_{-1} = d_{-1} = 0$ and

$$\underline{c_{n+1}} = \begin{cases} \underline{c_n} & \text{if } s_{n+1} = 1 \\ \underline{c_n} + \underline{k_{n+1}} & \text{if } s_{n+1} = 0 \end{cases} \qquad \underline{d_{n+1}} = \begin{cases} \underline{d_n} & \text{if } s_{n+1} = 1 \\ \underline{d_n} + \neg\underline{k_{n+1}} & \text{if } s_{n+1} = 0 \end{cases}$$

Proof.




unroll$_n$
$\underline{s_{n+1}} = 1$
$\underline{c_t}$
F_X
$\underline{d_t}$
$\underline{x\pi}$
$\star$
$+$
$\frac{1}{2}$
unroll$_n$
$\underline{s_{n+1}} = 0$
$\underline{c_t}$
F_X
$\underline{d_t}$
$\underline{k_{n+1}\pi}$
$\star$
$\underline{\neg k_{n+1}\pi}$
$=$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 1$
$\underline{c_n}$
$\underline{x\pi}$
$\underline{x\pi}$
$\star$
$+$
$\frac{1}{2}\frac{1}{2^n}$
$\underline{s_{n+1}} = 1$
$\underline{c_n\pi}$
$\star$
$\underline{d_n\pi}$
$\underline{x\pi}$
$\star$
$+$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 0$
$\underline{c_n}$
$\underline{k_{n+1}\pi}$
$\underline{x\pi}$
$\star$
$\underline{\neg k_{n+1}\pi}$
$+$
$\frac{1}{2}\frac{1}{2^n}$
$\underline{s_{n+1}} = 0$
$\underline{c_n\pi}$
$\underline{k_{n+1}\pi}$
$\star$
$\underline{d_n\pi}$
$\underline{\neg k_{n+1}\pi}$
$=$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 1$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\underline{x\pi}$
$\star$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{s_{n+1}} = 1$
$(\underline{c_{n+1}+d_{n+1}})\pi$
$\underline{x\pi}$
$\star$
$\star$
$\star$
$+$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 0$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{s_{n+1}} = 0$
$\underline{c_{n+1}\pi}$
$\star$
$\star$
$\underline{d_{n+1}\pi}$
$=$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 1$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{s_{n+1}} = 1$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$\star$
$+$
$\frac{1}{2}(1-\frac{1}{2^n})$
$\underline{s_{n+1}} = 0$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{s_{n+1}} = 0$
$\underline{c_{n+1}\pi}$
$\star$
$\star$
$\underline{d_{n+1}\pi}$
$=$
$\frac{2}{2}(1-\frac{1}{2^n})+\frac{1}{2^{n+1}}$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{c_{n+1}\pi}$
$\star$
$\star$
$\underline{d_{n+1}\pi}$
$=$
$1-\frac{1}{2^{n+1}}$
$\underline{e_{n+1}}$
$\underline{x\pi}$
$\star$
$+$
$\frac{1}{2^{n+1}}$
$\underline{c_{n+1}\pi}$
$\star$
$\star$
$\underline{d_{n+1}\pi}$

□