Completeness for Fault Equivalence of Clifford ZX Diagrams

Maximilian Rüsch, Benjamin Rodatz, Aleks Kissinger

University of Oxford, Oxford, UK

Two circuits are considered to be equivalent under noise if the effect of faults on one circuit is no worse than the effect of faults on the other circuit. We call this relationship *fault equivalence*. Fault equivalence offers a way to transform circuits while provably preserving their fault-tolerant properties, enabling a framework for fault-tolerant circuit synthesis and optimisation that is correct by construction. The ZX calculus, a set of graphical rewrite rules for quantum computations, provides a useful tool for manipulating circuits while preserving fault equivalence. For this, the usual set of ZX rewrites has to be restricted to not only preserve the underlying linear map represented by the diagram but also fault equivalence.

In this work, we provide a set of ZX rewrites that are sound and complete for fault equivalence of Clifford ZX diagrams. This means that any equivalence that can be derived using the proposed rules is certain to be correct, and any correct equivalence can be derived using only these rules. For this, we utilise diagrammatic constructions called fault gadgets to reason about arbitrary, possibly correlated Pauli faults in ZX diagrams. Fault gadgets allow us to separate the diagram into a fault-free part, which captures the noise-free behaviour of a diagram, and a noisy part that enumerates the effects of all possible faults. Using this, we provide a unique normal form for ZX diagrams under noise and show that any diagram can be brought into this normal form using our proposed rule set.

Contents

1 Introduction

Scaling quantum computation involves, among many other components, optimising quantum programs and suppressing noise. Independently, both fields are making progress: entire frameworks, libraries and calculi are built to both systematically and heuristically simplify quantum programs while preserving their semantics [Nam+18; Jav+24; KvdW20; vdWet20], while fault-tolerant quantum computing research offers a versatile range of tools to reason about, detect, and correct faults resulting from noise [Got09; Got97; Bac+17; DP23]. More recently, [RPK24; RPK25] have suggested a way to combine both fields, providing a framework to rewrite quantum circuits while preserving their behaviour under noise and therefore their fault tolerance properties. This allows the provably fault-tolerant manipulation of quantum circuits, which can, for example, be used for circuit synthesis and optimisation. This paper advances this line of research by providing a complete set of such rewrite rules, meaning that, using these rules, any two circuits can be transformed into each other if and only if they are equivalent under noise.

The core notion that underlies this framework is *fault equivalence*. Fault equivalence is based on the observation that two quantum circuits that implement the same linear map may have very different behaviour under noise, e.g. a single fault on one circuit may be much more detrimental than a single fault on the other. To formalise this, fault equivalence requires that for every undetectable fault that can happen on one circuit, there exists an equivalent fault on the other circuit. Additionally, this equivalent fault must have at most the same weight as the original fault, meaning it is at least as likely as the original one. This relationship must hold in both directions. This guarantees that in a larger protocol, fault-equivalent circuits can be interchanged without affecting the protocol's behaviour under noise. Therefore, we can perform fault-tolerant circuit synthesis and optimisation as long as we preserve fault equivalence.

To efficiently perform fault-tolerant circuit manipulation, [RPK25] proposes using the ZX calculus. The ZX calculus formalises quantum programs as undirected diagrams obtained from a few simple generators [CD11]. It has proven successful as a framework for circuit synthesis and optimisation [Dun+20; Cow+20; Sta+24]. However, rewrites on the vanilla ZX calculus only preserve semantic equivalence, i.e. whether the corresponding diagrams represent the same linear map. They do not guarantee the preservation of the stronger notion of fault equivalence. Therefore, [RPK25] restricts the set of allowed rewrites to *fault-equivalent rewrites*. These are ZX rewrites that not only preserve semantic equivalence but also fault equivalence.

A major question concerning any formal rewrite system is whether it is sound and complete. That is, the calculus should not allow deriving equivalences between expressions that do not actually hold, and it should allow deriving every equivalence that does hold. While previous work in this domain has proven the soundness of the proposed rewrite rules for fault equivalence, it remained an open question of whether all valid fault equivalences could be derived from these rules. In this work, we propose a set of fault-equivalent rewrites and show that it is sound and complete.

To facilitate the proof, we generalise the previous notion of a *noise model* from [RPK25] that described noise through a set of independent faults which may occur during computation, called *atomic faults*. In our revised notion, atomic faults may be assigned integer weights describing their individual likelihood, inducing a weight for all faults generated under their multiplication. By generalising fault equivalence to these noise models, we increase the expressiveness of the framework, enabling circuit synthesis for operations with varying noise levels.

Further, we utilise *fault gadgets* [RPK25] to represent and manipulate atomic faults in circuits. This allows us to separate a diagram into a fault-free part with non-trivial semantics and a faulty part with trivial semantics. Therefore, to rewrite one diagram into another, we have to show that each part can be rewritten individually. For the fault-free part of the diagram, we leverage existing completeness results [Bac14a; BPW17]. For the faulty part of the diagram, we provide a novel procedure to rewrite any diagram into another fault-equivalent diagram, through a unique normal form that any diagram can be rewritten into.

As part of the rewriting procedure, we construct a framework to manipulate faults in spacetime. We recover *inconsequential faults* [Mag+25] that leave the diagram unchanged and form a group under multiplication. This yields an equivalence of faults in spacetime: Two faults change the diagram in the same manner if and only if multiplying one by some inconsequential fault yields the other. We further show that the group of inconsequential faults is exactly generated by local stabilisers inside the diagram, allowing us to reproduce the spacetime equivalence with fault gadgets. Finally, we augment the framework with *flip operators*, a class of faults such that every detectable or undetectable atomic fault may be expressed as a combination of flip operators and actions on the boundary. These tools for manipulating faults open avenues to explore the behaviour of circuits under noise beyond fault equivalence checking.

Our proof of completeness is constructive, so an algorithmic procedure for checking fault equivalence can be directly derived. An implementation of this algorithm to automatically check fault equivalence of ZX diagrams is provided in an open source Python package [Rüs25b] for specific kinds of noise models[1].

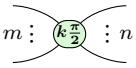
2 Background

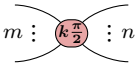
2.1 ZX Calculus

The ZX calculus is a graphical language for representing and reasoning about linear maps between qubits [CD11]. Throughout this work, we focus on the Clifford ZX calculus, a fragment for Clifford circuits. For a more thorough introduction to the entire calculus, we refer to [vdWet20].

The basic building blocks of the ZX calculus are the *spiders* that represent linear maps from $(\mathbb{C}^2)^{\otimes m}$ to $(\mathbb{C}^2)^{\otimes n}$:

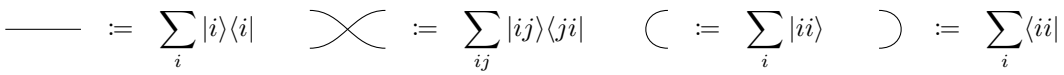
Definition 2.1 (Z spider, X spider)**.**

Z-spider:  $:= |0\rangle^{\otimes n}\langle 0|^{\otimes m} + e^{ik\frac{\pi}{2}}|1\rangle^{\otimes n}\langle 1|^{\otimes m}$

X-spider:  $:= |+\rangle^{\otimes n}\langle +|^{\otimes m} + e^{ik\frac{\pi}{2}}|-\rangle^{\otimes n}\langle -|^{\otimes m}$

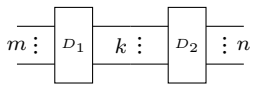
By convention, when the phase of a spider is 0, we omit the phase parameter.

We also include *identities*, *swaps*, *cups*, and *caps* in ZX diagrams:



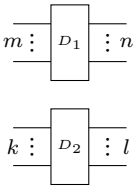
We can compose these elementary building blocks to form larger ZX diagrams:

Definition 2.2 (Composition of ZX diagrams)**.** The *sequential composition* of two diagrams $D_1 : (\mathbb{C}^2)^{\otimes m} \to (\mathbb{C}^2)^{\otimes k}$ and $D_2 : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes n}$ is the diagram $D_2 \circ D_1$, which corresponds graphically to:

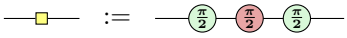
$D_2 \circ D_1 \quad \rightsquigarrow$  $: (\mathbb{C}^2)^{\otimes m} \to (\mathbb{C}^2)^{\otimes n}$

[1] Available at https://github.com/maximilianruesch/fault-gadgets

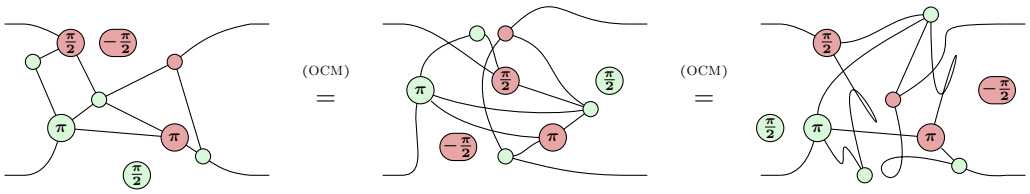
The *parallel composition* of two diagrams $D_1 : (\mathbb{C}^2)^{\otimes m} \to (\mathbb{C}^2)^{\otimes n}$ and $D_2 : (\mathbb{C}^2)^{\otimes k} \to (\mathbb{C}^2)^{\otimes l}$ is the diagram $D_1 \otimes D_2$, which corresponds graphically to:

$D_1 \otimes D_2 \quad \rightsquigarrow$  $: (\mathbb{C}^2)^{\otimes(m+k)} \to (\mathbb{C}^2)^{\otimes(n+l)}$

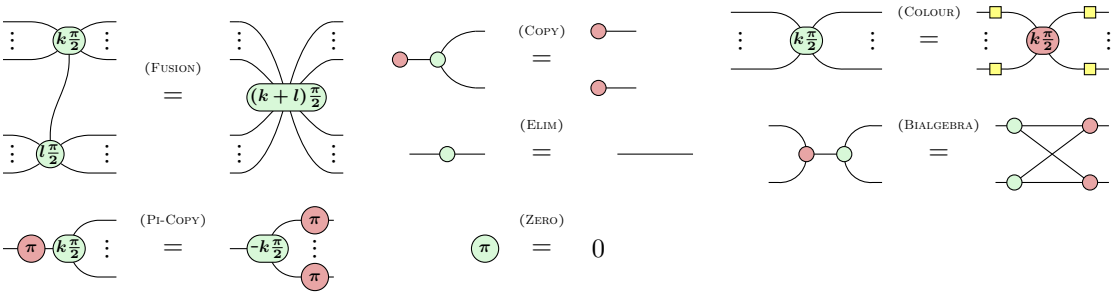
As a shorthand, we further introduce a special notation for the Hadamard gate H, one of the basic Clifford unitaries:



Along with diagrams, the ZX calculus also features a set of axioms called 'rewrite rules', that enable showing equivalences between diagrams. An essential rule that we most often assume implicitly is 'Only Connectivity Matters', commonly abbreviated 'OCM'. This rule states that as long as we keep the connectivity of the spiders internally and the order of inputs and outputs the same, we may move spiders and bend legs arbitrarily without changing the underlying linear map:



Beyond OCM, we present seven additional rewrite rules:



The axioms represent the most basic rewrites that transform one diagram into another semantically equivalent diagram. All of these rules also hold in the colour inverse, but not all of them are scalar accurate as presented, since we mostly disregard non-zero global phases in diagrams. We note that these rewrites are inherently bidirectional, i.e. the transformation they apply has no particular direction in which it does not hold.

A key characteristic of the ZX calculus is that it is sound and complete with respect to semantic equivalence:

Theorem 2.3 (Completeness of the Clifford ZX calculus)**.** We can rewrite two Clifford ZX diagrams D_1, D_2 into each other using the axioms above if and only if they represent the same linear map.

Proof. See [Bac14a; BPW17]. □

Finally, we note that through the Choi-Jamiołkowski isomorphism [NC10] we can uniquely identify a linear map with a pure quantum state and vice versa. In the ZX calculus, this isomorphism corresponds to bending around the input wires of a diagram into output wires, being careful

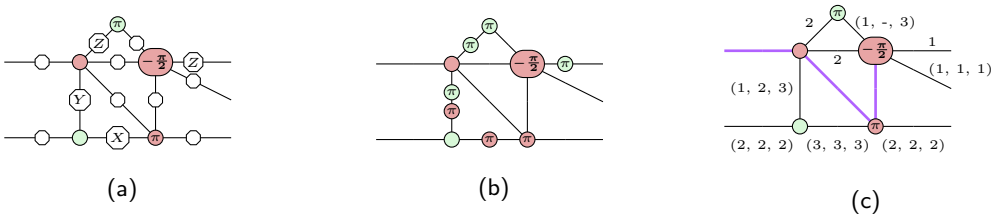


Figure 1: An example ZX diagram from two qubits to three qubits. We have (a) A fault on the diagram as indicated by the Paulis in the octagons. (b) The faulty diagram obtained from applying the fault from (a). (c) An example annotation of an edge-flip noise model under which the fault from (a) has weight 8.

to keep their relative order, or bending them back into input wires. For the Clifford fragment, ZX diagrams uniquely correspond to stabiliser states following the stabiliser formalism. Throughout this work, we will thus not distinguish between a Clifford ZX diagram representing a linear map and its corresponding stabiliser state.

2.2 Noise on ZX Diagrams

Following [Bac+17; Got22; DP23; RPK25], we consider Pauli faults in spacetime. We define:

Definition 2.4 (Pauli group)**.** The *Pauli group* $\mathcal{P}^1$ is defined as:

$$\mathcal{P}^1 = \{\alpha P \mid \alpha \in \{1, -1, i, -i\}, P \in \{I, X, Y, Z\}\}$$

where

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The n-qubit Pauli group $\mathcal{P}^n$ is defined as the n-fold tensor product of $\mathcal{P}^1$. The Pauli group up to scalars, denoted $\overline{\mathcal{P}^n}$, is defined as the quotient of the Pauli group by its centre $\mathcal{P}^n/\{\pm I, \pm iI\}$.

We can represent faults in spacetime as elements of the Pauli group up to scalars:

Definition 2.5 (Faults on ZX diagrams)**.** Let D be a ZX diagram with edges E. A *fault* F on D is an element of $\overline{\mathcal{P}^{|E|}}$. The *faulty diagram* D^F is the diagram D with the corresponding Pauli rotation applied to the respective edges as indicated by F.

To indicate a specific fault F, we draw the effect of the fault on each edge in octagons on the edges, to clearly distinguish faults from Paulis that are part of the original diagram (see Fig. 1 (a)). Applying F to D then gives rise to the new faulty diagram D^F (see Fig. 1 (b)).

We treat the ZX diagram D as postselected on an expected measurement outcome. From [KvdW24] we know that if a diagram is zero, we will never see that particular measurement outcome. Thus, if $D^F = 0$, we never see the expected measurement outcome. In other words, we always see an unexpected outcome and therefore know that something went wrong — the error is detected. We have:

Definition 2.6 (Detectability of faults)**.** Let D be a diagram and $F \in \overline{\mathcal{P}^{|E|}}$ be a fault on D. We say F is detectable if $D^F = 0$.

Having defined faults in spacetime, we now provide a way to specify what faults can happen if something goes wrong:

Definition 2.7 (Weighted noise model)**.** Let D be a ZX diagram with edges E. Let $\mathcal{F} \subseteq \overline{\mathcal{P}^{|E|}}$ be a set of faults that we treat as atomic faults. Then a *weighted noise model* $\mathcal{N}$ is a map $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$ that assigns each atomic fault a weight, which we call *atomic weight*. The set of all possible faults under the noise model $\mathcal{N}$ is the group $\langle \mathcal{F} \rangle$ generated by the atomic faults $\mathcal{F}$.

In this definition, atomic faults make up the elementary faults that can occur. A fault is a combination of atomic faults. The atomic weight of an atomic fault indicates how likely it is — faults with a lower weight are more likely to occur than faults with a higher weight. A weight can also be provided for combinations of atomic faults, through finding the combination with the lowest atomic weight sum:

Definition 2.8. (Induced weight function) A noise model $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$ induces a *weight* function $\mathrm{wt}_{\mathcal{N}} : \langle \mathcal{F} \rangle \to \mathbb{N}$, defined for a fault $F \in \langle \mathcal{F} \rangle$ as the minimum weight of any product of atomic faults that yields F:

$$\mathrm{wt}_{\mathcal{N}}(F) = \min_{\widetilde{\mathcal{F}} \subseteq \mathcal{F}} \sum_{F_i \in \widetilde{\mathcal{F}}} \mathcal{N}(F_i) \qquad \text{where} \qquad \prod_{F_i \in \widetilde{\mathcal{F}}} F_i = F.$$

When the noise model $\mathcal{N}$ inducing the weight function $\mathrm{wt}_{\mathcal{N}}$ is clear from context, we simply write wt.

With these two definitions, we generalise the noise model from [RPK25], which is a special case where all atomic faults have weight 1.

Throughout this work, we will primarily focus on one class of noise models:

Definition 2.9. A noise model $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$ is an *edge-flip noise model* if all atomic faults $\mathcal{F}$ act non-trivially on at most one edge.

Edge-flip noise is a helpful restriction as it substantially simplifies reasoning about diagrams under noise. Simultaneously, in Proposition 4.3, we will argue that edge-flip noise is a sufficiently general noise model, as for any diagram under any noise model, we can find an equivalent diagram under edge-flip noise.

Furthermore, edge-flip noise allows us to depict the ZX diagram and the noise model in a single picture by annotating each edge with a triplet (w_X, w_Y, w_Z) indicating the respective weight of the X, Y, and Z edge-flip affecting that edge. If a fault is not part of the atomic noise model, we indicate its weight as $-$. As we will later primarily reason about Z edge flips, if on a specific edge neither X nor Y edge flips are part of the noise model, we will annotate the edge with a single number w as a shorthand for $(-, -, w)$. If an edge allows no faults, we will omit the annotation entirely and draw the edge in purple to show that it is idealised as fault-free. Fig. 1 (c) shows an example noise model. In practice, noise models will often be more uniform.

2.3 Fault Equivalence

In a noise-free setting, we say that two diagrams are equivalent if they represent the same linear map. However, in the presence of faults, we need a stronger notion of equivalence that also takes the behaviour of the circuits under noise into account. For this purpose, the notion of *fault equivalence* was first introduced by [RPK24] and later refined in [RPK25]. For an in-depth motivation of fault equivalence and an overview of its applications, we refer to [RPK25; RPK24].

In this paper, we relax the notion of fault equivalence to an asymmetric relation called fault boundedness. If two diagrams are mutually fault-bounded by each other, they are fault-equivalent.

Definition 2.10 (w-fault boundedness, fault boundedness)**.** Let D_1, D_2 be ZX diagrams with respective noise models $\mathcal{N}_1 : \mathcal{F}_1 \to \mathbb{N}^+, \mathcal{N}_2 : \mathcal{F}_2 \to \mathbb{N}^+$. We say that D_1 under $\mathcal{N}_1$ is *w-fault-bounded* by D_2 under $\mathcal{N}_2$ for some $w \in \mathbb{N}^+$ if for any fault F_1 on D_1 with $\mathrm{wt}_{\mathcal{N}_1(F_1)} < w$, we have either:

- F_1 is detectable, **or**
- there exists a fault F_2 on D_2 with $\mathrm{wt}_{\mathcal{N}_2}(F_2) \leq \mathrm{wt}_{\mathcal{N}_1}(F_1)$ such that $D_1^{F_1} = D_2^{F_2}$.

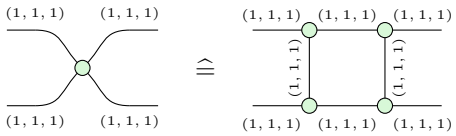
We write $D_1 \underset{w}{\widehat{\leq}} D_2$, leaving the noise models implicit when they are clear from context. D_1 under $\mathcal{N}_1$ is *fault-bounded* by D_2 under $\mathcal{N}_2$, written $D_1 \widehat{\leq} D_2$, if $D_1 \underset{w}{\widehat{\leq}} D_2$ for all $w \in \mathbb{N}^+$.

Using this, we can now define:

Definition 2.11 (w-fault equivalence, fault equivalence)**.** Let D_1, D_2 be ZX diagrams with respective noise models $\mathcal{N}_1, \mathcal{N}_2$. We say that D_1 under $\mathcal{N}_1$ is *w-fault-equivalent* to D_2 under $\mathcal{N}_2$ for some $w \in \mathbb{N}^+$, written $D_1 \underset{w}{\widehat{=}} D_2$, if $D_1 \underset{w}{\widehat{\leq}} D_2$ and $D_2 \underset{w}{\widehat{\leq}} D_1$. D_1 under $\mathcal{N}_1$ is *fault-equivalent* to D_2 under $\mathcal{N}_2$, written $D_1 \widehat{=} D_2$, if $D_1 \underset{w}{\widehat{=}} D_2$ for all $w \in \mathbb{N}^+$.

Example 2.12: Four-legged spider

One example of two fault-equivalent diagrams is the four-legged spider unfusion under edge-flip noise [RPK24]:



The unfused diagram introduces one detecting region (see Section 2.4). Therefore, odd many X and Y edge-flips on the internal edges are detectable. To check fault equivalence, we can iterate over all undetectable faults and check that they have corresponding faults on the other side. For more details, see [RPK24]. In this work, we will use this fault equivalence as a running example to show how these two diagrams can be deformed into one another using our proposed rule set.

This paper is mainly devoted to showing the completeness of a set of rewrite rules for fault equivalence. For this, an essential property of w-fault boundedness, and therefore also of fault boundedness and fault equivalence, is that it is transitive:

Proposition 2.13. w-fault boundedness is transitive, i.e. for ZX diagrams D_1, D_2, D_3 under respective noise models $\mathcal{N}_1 : \mathcal{F}_1 \to \mathbb{N}^+, \mathcal{N}_2 : \mathcal{F}_2 \to \mathbb{N}^+, \mathcal{N}_3 : \mathcal{F}_3 \to \mathbb{N}^+$ it holds that

$$D_1 \underset{w_1}{\widehat{\leq}} D_2 \quad \text{and} \quad D_2 \underset{w_2}{\widehat{\leq}} D_3 \qquad \Longrightarrow \qquad D_1 \underset{\min(w_1,w_2)}{\widehat{\leq}} D_3 \,.$$

Proof. We start with an undetectable fault $F_1 \in \mathcal{F}_1$ with $\mathrm{wt}(F) < \min(w_1, w_2)$. As we assumed $D_1 \underset{w_1}{\widehat{\leq}} D_2$, there is some $F_2 \in \mathcal{F}_2$ with $D_1^{F_1} = D_2^{F_2}$ and $\mathrm{wt}(F_2) \leq \mathrm{wt}(F_1)$. As $D_2 \underset{w_2}{\widehat{\leq}} D_3$, we can similarly obtain some $F_3 \in \mathcal{F}_3$ with $D_2^{F_2} = D_3^{F_3}$ and $\mathrm{wt}(F_3) \leq \mathrm{wt}(F_2)$.

But then via transitivity, $D_1^{F_1} = D_2^{F_2} = D_3^{F_3}$ and $\mathrm{wt}(F_3) \leq \mathrm{wt}(F_2) \leq \mathrm{wt}(F_1)$, completing the claim. □

Additionally, we can observe that w-fault boundedness is preserved under sequential and parallel composition:

Proposition 2.14. Fault boundedness is compositional, i.e. for ZX diagrams D_1, D_1', D_2, D_2' with respective noise models $\mathcal{N}_1, \mathcal{N}_1', \mathcal{N}_2, \mathcal{N}_2'$ it holds that

$$D_1 \underset{w_1}{\widehat{\leq}} D_2 \quad \text{and} \quad D_1' \underset{w_2}{\widehat{\leq}} D_2' \qquad \Longrightarrow \qquad D_1' \circ D_1 \underset{\min(w_1,w_2)}{\widehat{\leq}} D_2' \circ D_2 \,,$$

$$D_1 \underset{w_1}{\widehat{\leq}} D_2 \quad \text{and} \quad D_1' \underset{w_2}{\widehat{\leq}} D_2' \qquad \Longrightarrow \qquad D_1 \otimes D_2 \underset{\min(w_1,w_2)}{\widehat{\leq}} D_1' \otimes D_2' \,,$$

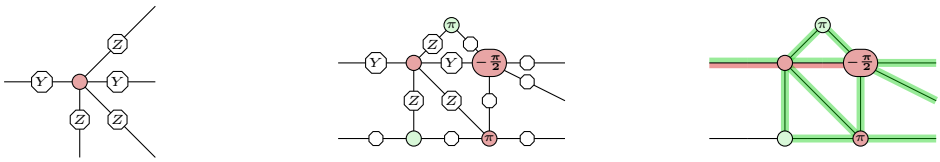
where the composed diagrams have noise models uniquely composed of $\mathcal{F}_1, \mathcal{F}_1', \mathcal{F}_2, \mathcal{F}_2'$ under atomic fault set union (while padding faults with I for edges in the other diagrams) and addition of atomic weights.

Proof. We prove the claim for sequential composition. The proof for parallel composition is analogous. Now let $\mathcal{N}_1^\circ, \mathcal{N}_2^\circ$ be the noise models obtained for $D_1' \circ D_1$ and $D_2' \circ D_2$ respectively. These noise models are well defined, because the original noise models they are respectively sourced from are operating on disjoint sets of edges, and thus in disjoint spaces of $\overline{\mathcal{P}^{|\cdot|}}$.

We start with considering an undetectable fault $F_1^\circ \in \mathcal{F}_1^\circ$ of weight less than $\min(w_1, w_2)$. F_1° is composed of F_1, F_1' drawn from $\mathcal{F}_1, \mathcal{F}_1'$ respectively, and since $\mathcal{F}_1^\circ$ was formed under the addition of atomic weights, it must be that

$$\mathrm{wt}(F_1^\circ) = \mathrm{wt}(F_1) + \mathrm{wt}(F_1') < \min(w_1, w_2) \,.$$

Then, via the precondition of w_1- and w_2-fault boundedness, we can find semantically equivalent F_2, F_2' drawn from $\mathcal{F}_2, \mathcal{F}_2'$ that have weights of at most those of F_1, F_1'. Their composite $F_2^\circ = F_2 F_2'$



(a) (b) (c)

Figure 2: (a) A spider stabiliser of a single spider. (b) The same spider stabiliser in a larger diagram. (c) A Pauli web on a diagram.

is then a fault on $D_2' \circ D_2$ that can be drawn from $\mathcal{F}_2^\circ$ and it holds that

$$\begin{array}{ccccccc} (D_2' \circ D_2)^{F_2^\circ} & = & {D_2'}^{F_2'} \circ D_2^{F_2} & = & {D_1'}^{F_1'} \circ D_1^{F_1} & = & (D_1' \circ D_1)^{F_1^\circ}, \\ \mathrm{wt}(F_2^\circ) & = & \mathrm{wt}(F_2) + \mathrm{wt}(F_2') & \leq & \mathrm{wt}(F_1) + \mathrm{wt}(F_1') & = & \mathrm{wt}(F_1^\circ). \end{array}$$

Thus, there is an equivalent F_2° for F_1° with lower or equal weight. □

We close this discussion on fault equivalence by considering the case that a noise model $\mathcal{N}$ contains no faults for a diagram D. In this case, we call D *fully idealised*. Intuitively, D being fault-bounded by a second diagram D' is equivalent to stating that the noise model $\mathcal{N}'$ of D' describes 'worse' behaviour under noise for D' than $\mathcal{N}$ describes for D. Then, the following result is unsurprising:

Proposition 2.15. Let D be a diagram and $\mathcal{N}$ a noise model such that D is fully idealised. Then for any diagram D' with $D = D'$ and any accompanying noise model $\mathcal{N}'$, it holds that $D \mathrel{\widehat{\leq}} D'$.

Furthermore, if $\mathcal{N}'$ is such that D' is fully idealised as well, the symmetric $D \mathrel{\widehat{=}} D'$ holds.

Proof. As $\mathcal{N}$ has no atomic faults, trivially, there exists a corresponding fault in $\mathcal{N}'$ for all faults in $\mathcal{N}$.

The second statement follows from applying the first statement in both directions, giving fault equivalence of the two diagrams. □

2.4 Pauli Webs

Pauli webs are decorations for ZX diagrams that highlight some of the edges of the diagram in red and/or green. They are useful tools for tracking stabilisers and detectors, and as we will formalise in Section 4.2, they can be used to characterise and classify faults.

For this, we first observe that spiders are stabilised by Paulis, as they represent Clifford maps. We can define:

Definition 2.16 (Spider stabilisers)**.** Let a ZX diagram D consist of one spider s with k legs. Then a Pauli $P \in \overline{\mathcal{P}^k}$ is a *spider stabiliser* of s if $D^P = D$. In a larger diagram, a Pauli P is a spider stabiliser of a spider s if it is a spider stabiliser of the subdiagram formed by s alone and acts trivially on all other edges.

For example, Fig. 2 (a) shows a spider stabiliser of a single spider, while Fig. 2 (b) shows the same spider stabiliser in a larger diagram.

Based on this, we define:

Definition 2.17 (Pauli web)**.** Let D be a ZX diagram with edges E. An element $P \in \overline{\mathcal{P}^{|E|}}$ is a *Pauli web* if for each spider s of D, P restricted to the neighbourhood of s is a spider stabiliser of s.

We can view Pauli webs as highlighting the edge $i \in 1, \ldots, |E|$ of a diagram as red if $P_i = X$, green if $P_i = Z$, and red and green if $P_i = Y$.

It is important to note that not every spider stabiliser is a valid Pauli web in the context of the entire diagram. Instead, we require that locally, restricted to the neighbourhood of each spider, the Pauli web acts like a spider stabiliser. For example, Fig. 2 (c) shows a Pauli web that includes the

spider stabiliser of Fig. 2 (b) and additionally has a non-trivial action on other edges. In contrast, Fig. 2 (b) would not be a valid Pauli web as it does not act like a spider stabiliser on some of the spiders, such as the green π-phased spider at the top.

Using this, we can recover the usual definition of Pauli webs as in [Bom+24; RPK24]:

Proposition 2.18 (Pauli Web)**.** Let D be a ZX diagram with edges E. $P \in \overline{\mathcal{P}^{|E|}}$ is a Pauli web if and only if:

- a spider with phase in $\{0, \pi\}$ satisfies:
  - P highlighting an *even* number of its legs in its own colour *and all or none* in the opposite colour
- a spider with phase in $\{-\frac{\pi}{2}, \frac{\pi}{2}\}$ satisfies:
  - P highlighting an *even* number of its legs in its own colour *and none* in the opposite colour, **or**
  - P highlighting an *odd* number of its legs in its own colour *and all* in the opposite colour.

Proof. The constraints on the neighbourhood of spiders guarantee exactly that the corresponding restrictions of P to the legs of each spider are spider stabilisers. Therefore, P is a Pauli web. Similarly, all Pauli webs must satisfy these constraints. □

As Pauli webs, locally restricted to a spider, correspond to spider stabilisers, we can 'fire' spiders according to a Pauli web:

Definition 2.19 (Firing a diagram according to a Pauli web)**.** Let D be a ZX diagram and P be a Pauli web on D. Then we say we *fire all spiders in D according to P* when we:

1. Introduce Paulis on all the edges according to P, **and**
2. On all the internal edges, merge the newly fired Paulis.

As all the internal edges are connected to two spiders, any highlighted internal edge will have two Paulis of the same type introduced on it, which are removed in the second step.

Firing a spider according to the Pauli web does not change the underlying linear map (up to a global phase). We can see this by considering that a Pauli web locally corresponds to stabilisers of the spiders, so the first step does not change the diagram. Similarly, merging Paulis does not change the semantics of the diagram, as all Paulis are self-adjoint and thus square to the identity.

After firing a diagram according to a Pauli web, having cancelled out all the Paulis on the internal edges, all we are left with are Paulis on the boundary edges. We just reasoned that firing the web does not change the diagram's semantics, so we can freely introduce or remove these Paulis on the boundaries — they stabilise the diagram:

Definition 2.20 (Stabilising Pauli webs)**.** A Pauli web on some diagram is called a stabilising Pauli web if it acts non-trivially on at least one boundary edge.

Stabilising Pauli webs of a diagram D are in direct correspondence with the stabilisers of the underlying linear map of D [Bom+24; RPK24]; the Paulis remaining on the boundary edges after firing a diagram according to a Pauli web exactly correspond to its stabilisers.

However, this idea of firing spiders according to a Pauli web gives us a second, powerful tool. First, we observe that firing a diagram according to a Pauli web might change the global phase of a diagram by a factor of -1. In the first step of firing a diagram according to a Pauli web, we introduce Paulis that locally correspond to stabilisers of the spiders. Therefore, this preserves the underlying semantics of the diagram up to a global phase of $+1$ or -1, depending on whether the spider is a $+1$ or -1 eigenstate of the corresponding Pauli. Merging Paulis in the second step does not change the phase, again because all Paulis are self-adjoint.

Using these observations, we define:

Definition 2.21 (Sign of a Pauli web)**.** Let D be a ZX diagram and P be a Pauli web on D. Then the sign of P is the change in global phase we get when firing all the spiders of D according to P.

For stabilising Pauli webs, the sign of the Pauli web indicates whether the whole diagram is a $+1$ or -1 eigenstate of the corresponding stabiliser.

However, there is a second type of Pauli webs for which this sign is even more essential:

Definition 2.22 (Detecting region)**.** A non-trivial Pauli web on some diagram is called a detecting region if it acts trivially on all boundary edges.

For detecting regions, we can show:

Proposition 2.23. Let D be a ZX diagram with some detecting region R that has a negative sign. Then D is equivalent to zero.

Proof. We can fire all the spiders of D according to R. This changes the underlying linear map by a global phase of -1. R is a detecting region, all highlighted edges are internal and therefore now have two Paulis of the same type. But these cancel each other out, meaning we are back to the original diagram. But then $D = -D$, which can only be true when D is equivalent to zero. □

But then, we can make a second observation:

Proposition 2.24. Let D be a ZX diagram with edges E, $F \in \overline{\mathcal{P}^{|E|}}$ be a fault on D. Then for all Pauli webs on $P \in \overline{\mathcal{P}^{|E|}}$ there exists a corresponding Pauli web P_F on D^F that has the same action on the boundary. The sign of P is the same as the sign of P_F if and only if F commutes with P.

Proof. Let P be a Pauli web on D. Then we can construct a corresponding Pauli web on D^F. We can observe that D^F has the same edges as D, just that some of the edges of D are split into two by the faults. Therefore, we can construct a Pauli web P_F that acts the same as P on all edges, even the split ones. On all the original spiders of D, P_F clearly satisfies the Pauli web constraints of locally corresponding to a spider stabiliser. For the new spiders introduced by the fault, it also satisfies these constraints as the faults are two-legged π-phased spiders and P_F has the same action on both sides of that spider. Therefore, P_F is a corresponding Pauli web to P and clearly has the same action on the boundary as P.

The sign of P_F is obtained by firing all the spiders of D^F according to P_F. For all the spiders that are already present in D, this firing has the same effect as firing the spiders of D according to P. However, additionally, we have to fire the spiders newly introduced by F. For each edge, if F commutes with P on that edge, firing that spider does not change the global phase. However, if F anticommutes with P, firing the fault on that edge according to P_F flips the global phase. Therefore, if F commutes with P, the global phase will be changed evenly many times, and therefore the signs of P and P_F are the same. However, if they anticommute, the sign of P_F will be the opposite of the sign of P. □

For stabilising Pauli webs, anticommuting faults flip the sign of the corresponding stabilisers, thereby changing whether D^F lives in the same eigenspace of the stabiliser as D. For detecting regions, the commutative relationship between the fault and the regions tells us whether the fault is detectable. If we assume D to be non-zero, by Proposition 2.23, all detecting regions in D must have a positive sign. But then, any fault that anti-commutes with a detecting region will make the sign of that region in D^F negative. Therefore, by Proposition 2.23, D^F must be zero, meaning F is detected. Thus, detecting regions are a crucial tool for reasoning about detectable faults.

3 The Rules and Strategy

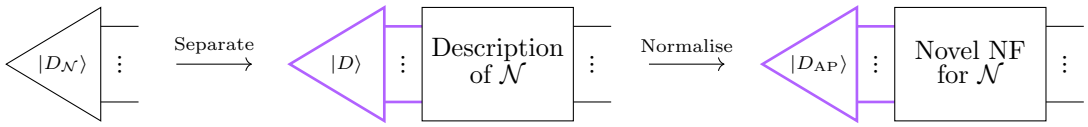
The main focus of this paper is to provide a diagrammatic rewrite system for the ZX calculus that is sound and complete for fault equivalence, of which the latter is often significantly harder to prove. Historically, showing completeness of a ZX rewrite system usually follows one of two general approaches: Either one reduces one calculus to another which is already known to be complete, e.g. [BPW17; NW17; Vil19], or one provides a unique normal form for diagrams from the calculus reachable through its rewrite system, e.g. [Bac14b; KvdW24]. As fault equivalence is a novel idea introduced only recently, there are no rewrite systems known to be complete. Therefore, we will show the completeness of the proposed rule set via a unique normal form.

3.1 The Normal Form and Proof Sketch

Completeness proofs via a unique normal form consist of showing that any diagram can be rewritten into a normal form that is unique for each equivalence class of diagrams, following some notion of equivalence. Then, any two equivalent diagrams must have the same normal form, reachable through two potentially different sequences of rewrites. All ZX rewrites are undirected, so we can invert one of those sequences to obtain a sequence that deforms one diagram into the other diagram, effectively showing the equivalence diagrammatically.

We now begin constructing our unique normal form. For two diagrams to be fault-equivalent, they have to satisfy two conditions: They have to implement the same linear map, and the sets of potentially generated fault effects along with their minimum weight must be equivalent. The first condition is exactly the *semantic equivalence* between ZX diagrams, which is a core feature of the calculus, and many (normal) forms have surfaced for it [KvdW24]. If we are able to find a diagrammatic representation to address both fault equivalence conditions individually, we are able to reuse these forms for the semantic part of the diagram.

For this, we make use of *fault gadgets*, a tool that can find a diagrammatic representation for arbitrary faults on ZX diagrams [RPK25]. Thus, for a diagram D and an associated noise model $\mathcal{N}$, we can find another diagram $D_{\mathcal{N}}$ describing both. However, going beyond the work of [RPK25] in Section 4, we show how fault gadgets can be fault-equivalently manipulated and moved. This allows us to separate the fault gadgets from the remainder of the diagram, such that we obtain a fully idealised diagram with non-trivial semantics followed by a noisy diagram consisting of fault gadgets with trivial semantics:



Our composite normal form consists of the unique *reduced AP form* for the idealised part of the diagram from [KvdW24], followed by a novel normal form on the noisy part of the diagram as specified in Section 5.

It then remains to show that the normal form for the diagrammatic description of noise is reachable through our proposed set of rewrites, and indeed unique for each fault equivalence class of diagrams. We continue in this section with introducing our rewrites, and show reachability of the normal form in Section 5. Then, if both parts of the composite normal form are unique, they yield an overall unique normal form and thus completeness of the rewrite system.

3.2 The Rules

We present our set of rules, proposed to form a complete calculus, in Fig. 3. These rules can be divided into the following three classes:

1. **Fault-Free Clifford**: All rules in this class are fully idealised variants of the basic ZX axiomatisation that we use. Among other things, they will be used to obtain the normal form for the semantic part of the separated diagram.

2. **Moving Fault Gadgets**: We require additional rules to move fault gadgets throughout a diagram. Along with (ElimFe), which enables applications of fully idealised rules next to noisy settings, we introduce (XPhaseFe) and (CommuteFe), which allow us to ignore global phases introduced when moving and commuting fault gadgets, respectively.

3. **Changing Atomic Faults**: To obtain our novel normal form, we will have to fault-equivalently change the atomic fault set and the weights associated with its members. Using only descriptions of Z flips, rules from this class describe intuitive changes to the atomic fault set.

   The first rule (ScalarFe) allows ignoring faults that have no observable effect. The second rule (MergeFe) removes equivalent faults, keeping the one with lower weight. The third rule (CombFe) introduces combinations of faults, again in a way that does not affect the induced weight function. Finally, the last rule (DetectFe) is essential to removing detectable atomic faults that are irrelevant for fault equivalence.

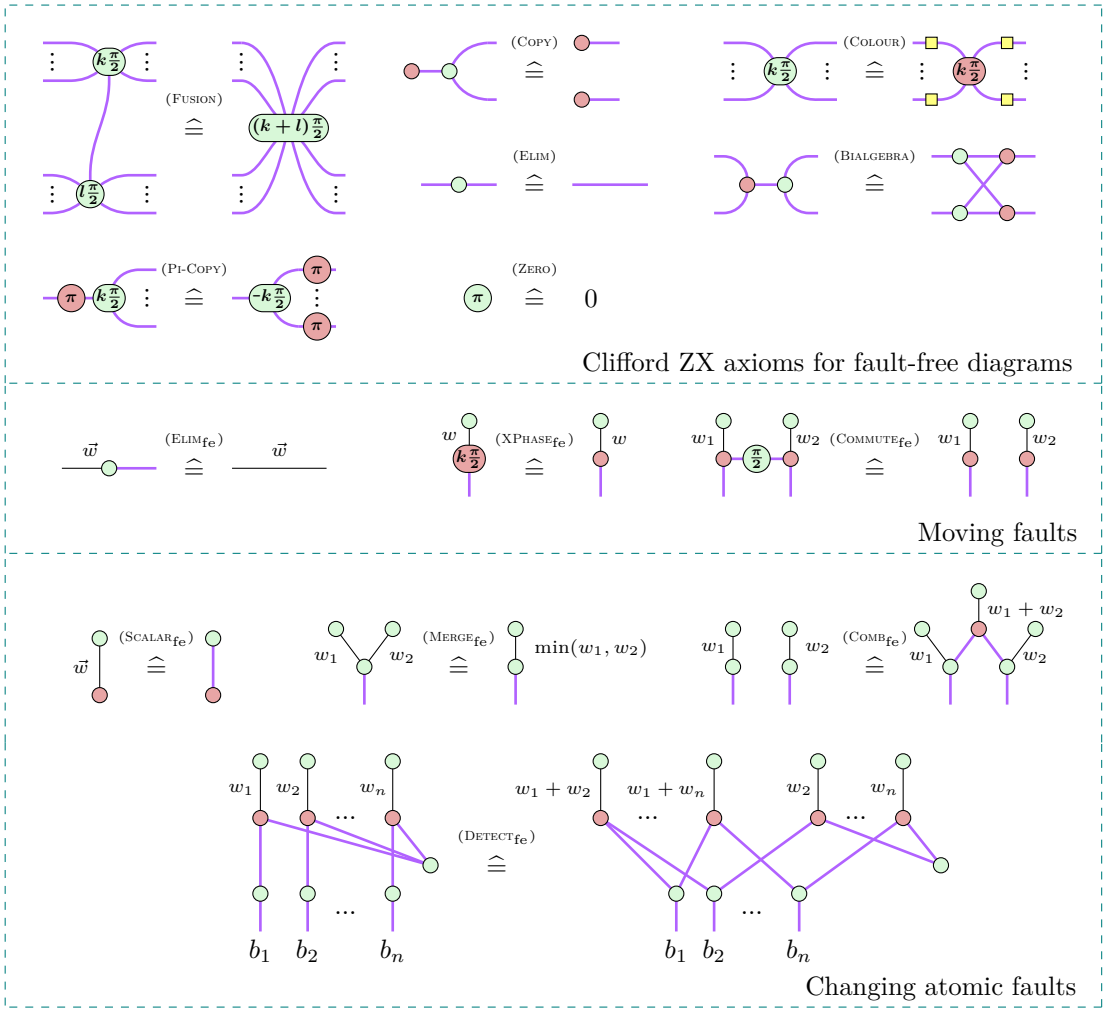


Figure 3: The list of rules which we propose to be complete for fault equivalence.

It is essential that these rules are sound, i.e. that they preserve fault equivalence, as otherwise we could accidentally equate diagrams that are not fault-equivalent. Thus, we show:

Proposition 3.1. The rules displayed in Fig. 3 are sound with respect to fault equivalence.

Proof. See Appendix A. □

Since fault equivalence is compositional and transitive, we can freely apply these rules in a larger context, and any sequence of sound rewrites is still sound.

Finally, we are now able to state our main result:

Theorem. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N}, \mathcal{N}'$ such that D under $\mathcal{N}$ is fault-equivalent to D' under $\mathcal{N}'$. Further, let $D_{\mathcal{N}}$ be the diagram describing both D and $\mathcal{N}$ using fault gadgets, and similar for $D'_{\mathcal{N}'}$.

Then, only using the rules in Fig. 3, $D_{\mathcal{N}}$ can be deformed into $D'_{\mathcal{N}'}$ and vice versa.

We revisit this theorem in Section 6 with Theorem 6.2, where we will prove it as well. Afterward, building on the completeness result, we will propose additional rules for obtaining a sound and complete calculus for w-fault equivalence, fault boundedness, and w-fault boundedness.

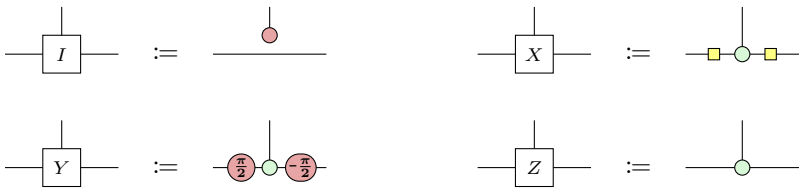
4 Separating Semantics and Noise

To bring any diagram into its unique normal form, we follow two steps: First, push all faults to the boundary, then normalise and reduce them as much as possible. This section focuses on the first step, pushing the faults outside the diagram. We use fault gadgets [RPK25] to represent faults diagrammatically. This lets us express any diagram under any noise model as fault-equivalent to one under edge-flip noise. Extending the work of [RPK25], we show that fault gadgets can furthermore be used to diagrammatically transform faults into equivalent ones.

4.1 Fault Gadgets

To reason about faults in spacetime, we use *fault gadgets* [RPK25]. First, we define:

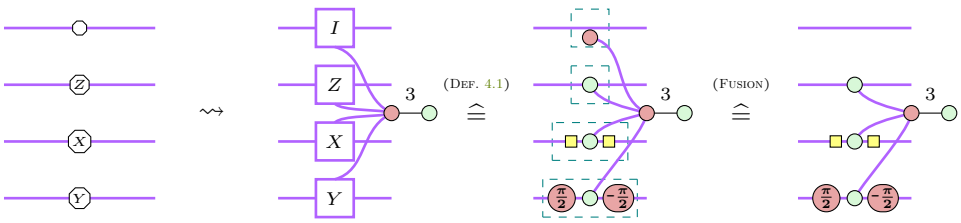
Definition 4.1 (Pauli boxes)**.** The following four diagrams are the *Pauli boxes*, associated with the annotated Pauli rotation:



Using these, we define:

Definition 4.2. Let F be a fault of weight w for a ZX diagram D with edges E. A *fault gadget* for F is constructed by including fault-free Pauli boxes, called *targets*, of the types and on the edges indicated by F. These boxes are connected via idealised edges to a red spider. Further, the red spider is connected via a regular edge to a green spider, called the *spawning edge* of the fault gadget. The spawning edge is annotated with w.

For example, the following diagram consists of four standalone wires, i.e. it implements a four-qubit identity. Suppose we now consider the fault $I \otimes Z \otimes X \otimes Y$ that has weight 3. The fault gadget corresponding to this fault is:



The identity Pauli box is disconnected, so its one-legged spider can be merged into the red spider. Therefore, for the remainder of this work, we choose not to draw the identity Pauli box at all.

We can now justify our focus on edge-flip noise:

Proposition 4.3. Let D be a ZX diagram and $\mathcal{N}$ be any weighted noise model for D. Then there exists a ZX diagram $D_{\mathcal{N}}$ along with a weighted *edge-flip* noise model $\mathcal{N}'$ such that $D = D_{\mathcal{N}}$ and D under $\mathcal{N}$ is fault-equivalent to $D_{\mathcal{N}}$ under $\mathcal{N}'$.

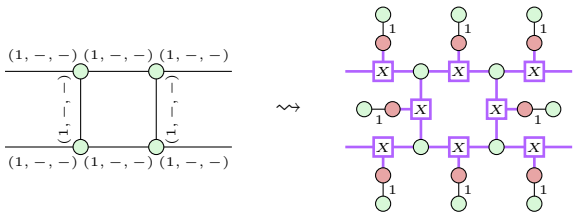
Proof. Let D be a ZX diagram and $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$ be a weighted noise model for D. To obtain $D_{\mathcal{N}}$ and $\mathcal{N}'$, first idealise all edges within D. Then add a fault gadget for each atomic fault $F \in \mathcal{F}$ with a weight annotation equal to $\mathcal{N}(F)$ on the spawning edge.

Fault gadgets do not influence the linear map of the underlying diagram [RPK25], so we directly have $D = D_{\mathcal{N}}$. Additionally, for every fault F on D, there exists a corresponding fault in $D_{\mathcal{N}}$ on its spawning edge of the corresponding fault gadget. Thus, for all F in $\mathcal{F}$, there exists a fault F' on $D_{\mathcal{N}}$ with $wt_{\mathcal{N}'}(F') = wt_{\mathcal{N}}(F)$ such that $D^F = D^{F'}_{\mathcal{N}}$. Similarly, by construction, the only allowed faults in $D_{\mathcal{N}}$ are on spawning edges of fault gadgets and naturally correspond to a fault in D. Therefore, the two diagrams are fault-equivalent. □

For our completeness result, we will only reason about diagrams in this form, meaning that prior to the diagrammatic reasoning, some straightforward pre-processing is necessary.

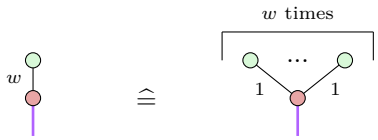
Example 4.4: Four-legged spider

Returning to our example of the four-legged spider, we can show what this construction looks like. To keep an interesting example and reduce clutter at the same time, we focus on the expanded diagram under just X edge-flips.



D $D_{\mathcal{N}}$

Remark 4.5. Going one step further, we can observe that, in fact, unweighted edge-flip noise, i.e. assigning each atomic edge-flip a weight of one, is already sufficiently expressive. More formally, for any fault gadget with weight w, we can find an equivalent fault gadget under unweighted edge-flip noise:



For more details, in particular how this is proven using just the rules presented in Section 3, see [Rüs25a, Lemma 4.5].

Weighted edge-flip noise can be seen as syntactic sugar for unweighted edge-flip noise. However, as weighted edge-flip noise is easier to work with, we choose to focus on that.

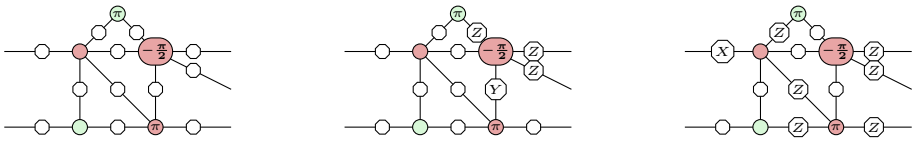
4.2 Inconsequential Faults

We can identify a special group of faults called inconsequential faults [Mag+25], also referred to as trivial faults [BH25]. These are the faults in spacetime that do not change a diagram's underlying linear map, i.e. the faults that have no consequence.

We define:

Definition 4.6 (Inconsequential faults)**.** Let D be a non-zero ZX diagram with edges E. Then a Pauli $S \in \overline{\mathcal{P}^{|E|}}$ is an *inconsequential fault* of D if $D^S = D$.

Faults that correspond to stabilisers of D are inconsequential. However, there are more inconsequential faults that are spread through spacetime. For example, we can observe that the following faults all leave the diagram unchanged:

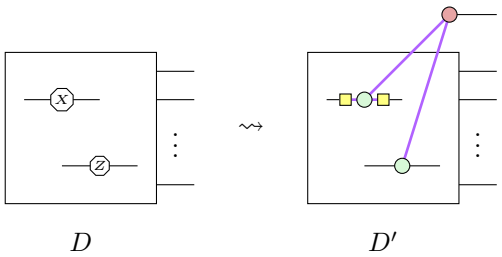


The first fault is the trivial fault, which is clearly inconsequential. The second fault corresponds to a spider stabiliser of the red $-\frac{\pi}{2}$-phased spider. The third fault is a product of multiple spider stabilisers. This leads us to a more general proposition:

Proposition 4.7. The group of inconsequential faults is equivalent to the group generated by all spider stabilisers.

Proof. Clearly, any spider stabiliser leaves the diagram unchanged and therefore is an inconsequential fault. Furthermore, any product of inconsequential faults is again inconsequential. Therefore, it remains to be shown that any inconsequential fault can be generated by stabilisers of spiders.

Let D be a non-zero ZX diagram and let F be an inconsequential fault of D. Then we can construct a diagram D' which helps us find a generating set of spider stabilisers that generate F. Let D' be D where we additionally add a fault gadget for F, however, leaving the spawning edge open, making it a boundary edge. For example, when F has two non-trivial targets, we get:

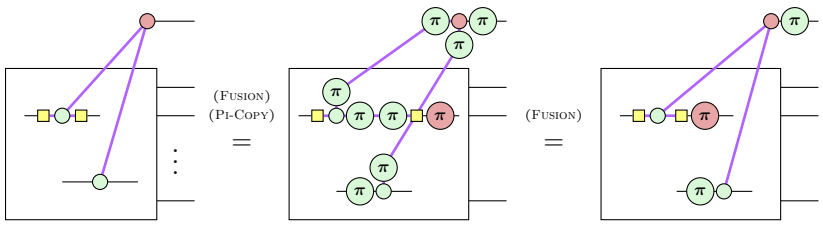


D $\qquad$ D'

Then, we can observe that placing a single Z flip on the newly created boundary edge and pushing it inwards has the same effect as applying F to D and must therefore be inconsequential. In other words, the Pauli that acts as Z on the new boundary edge and trivially everywhere else is a stabiliser of D'.

But then, by [Bor19], there must exist a generating set of spider stabilisers of D' that generate this Pauli. We can study this set of spider stabilisers more closely to find a generating set of spider stabilisers of D that generate F.

First, we observe that to create the Z flip on the spawning edge, we must use the stabiliser of the red spider of the fault gadget, introducing Z flips on all the edges connected to the targets. To remove these flips, within the Pauli boxes, we must use the stabiliser of the green spider, introducing another flip on the target edge, either to the left or to the right of this spider. To remove these flips, the stabiliser of the Clifford conjugation must be used, introducing a flip on the target edge of the colour of the Pauli box.



But then, having multiplied these spider stabilisers, we now have a fault that is equivalent to F on D. As we know that the overall multiplication is inconsequential on D', the remaining stabilisers must cancel out F. But then, as all the remaining spiders have a direct correspondence in D, the corresponding stabilisers in D must similarly multiply to F. Therefore, we have shown that there must exist a generating set of spider stabilisers in D that generate F.

We remark that the algorithm from [Bor19] only works for diagrams in red-green form, i.e. diagrams where red spiders are only connected to green spiders and vice versa. However, we can always bring a diagram into red-green form by introducing an identity spider of the opposite colour on all edges that connect two spiders of the same colour. The obtained spider stabilisers of this diagram can be directly translated back to spider stabilisers of the original diagram by ignoring all introduced identity spiders. □

Using these inconsequential faults, we can define an equivalence relation on faults:

Definition 4.8 (Spacetime equivalence)**.** Two faults F_1, F_2 on a ZX diagram D are *spacetime-equivalent* if there exists an inconsequential fault S such that $F_1 S = F_2$.

Spacetime equivalence of faults guarantees that two faults are the same up to some inconsequential fault, which does not change the diagram, i.e. that the two faults change the diagram in the same way. Therefore, when considering how to identify and, in particular, correct faults, it is sufficient to identify faults up to spacetime equivalence. We can formalise this:

Proposition 4.9. Let D be a ZX diagram and F_1, F_2 be faults on D. F_1 and F_2 are spacetime-equivalent if and only if they anticommute with the same Pauli webs.

Proof. If F_1 and F_2 are spacetime-equivalent, there exists an inconsequential fault S such that $F_1 S = F_2$. As $F_1 S$ and F_2 are the same, we know that they anticommute with the same Pauli webs. We will now argue that S commutes with all Pauli webs and therefore, F_1 and F_2 must anticommute with the same Pauli webs. By Proposition 2.24, we know that faults flip the sign of Pauli webs when the fault anticommutes with the Pauli web. When they flip the sign of a detecting region, they make the diagram zero and are therefore detected. As inconsequential faults do not change the diagram, they do not make it go to zero. Therefore, they have to commute with all detection regions. For undetectable faults, when they flip the sign of a stabilising Pauli web, they change the eigenspace that D^F lives in with respect to the corresponding stabiliser. But as inconsequential faults do not change the diagram, they can also not change the eigenspace that D^F lives in. Therefore, they have to commute with all stabilising Pauli webs as well. But then, as $F_1 S$ and F_2 anticommute with the same Pauli webs and S commutes with all Pauli webs, F_1 and F_2 must anticommute with the same Pauli webs.

Conversely, let F_1 and F_2 be faults that anticommute with exactly the same Pauli webs. But then $F_1 F_2$ must commute with all Pauli webs, as F_1 and F_2 either both commute or both anticommute with any given web. Thus, $D^{F_1 F_2} = D$ as $F_1 F_2$ does not flip the sign of any detecting region or stabiliser. Thus, $S = F_1 F_2$ must be an inconsequential fault. But then $F_1 S = F_1 F_1 F_2 = F_2$, so F_1 and F_2 are spacetime-equivalent. □

But then we have:

Proposition 4.10. Two undetectable faults F_1, F_2 on a non-zero diagram D are spacetime-equivalent if and only if $D^{F_1} = D^{F_2}$.

Proof. By Proposition 4.9, F_1 and F_2 are spacetime-equivalent if and only if they anticommute with the same Pauli webs. Therefore, they must anticommute with the same stabilising Pauli webs. But then D^{F_1} and D^{F_2} must live in exactly the same eigenspaces of all the stabilisers of D and therefore, by stabiliser theory, they must be the same. □

For detectable faults, we recall that a fault is detectable if it causes the diagram to go to zero. Therefore, we cannot strengthen the statement from Proposition 4.10 to include detectable faults, since two detectable faults may not be spacetime-equivalent, yet both make the diagram go to zero, i.e. satisfy $D^{F_1} = D^{F_2}$. However, as spacetime-equivalent faults must anticommute with the same Pauli webs, including all the detecting regions, each spacetime equivalence class of faults is either detectable or undetectable. This leaves us with the following adaptation of Proposition 4.10:

Corollary 4.11. If F_1, F_2 are two spacetime-equivalent faults on a diagram D, it must be that $D^{F_1} = D^{F_2}$.

Proof. If F_1 and F_2 are undetectable, this follows from Proposition 4.10. If they are detectable, they both make the diagram go to zero. Therefore, the statement holds in all cases. □

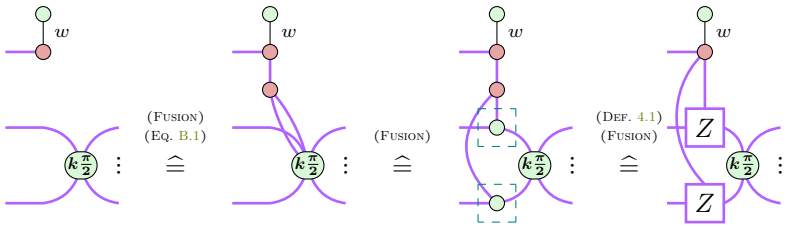
Fault equivalence, as defined, is only concerned with the faulty diagrams D^{F_1}, D^{F_2}, so through Corollary 4.11 it is safe to handle faults up to spacetime equivalence.

4.3 Manipulating Fault Gadgets

Beyond being capable of representing faults, fault gadgets can also be used to reason about faults. Since we just argued that for fault equivalence we only care about faults up to spacetime equivalence, we need to show that we can transform spacetime-equivalent fault gadgets into one another. First, we show:

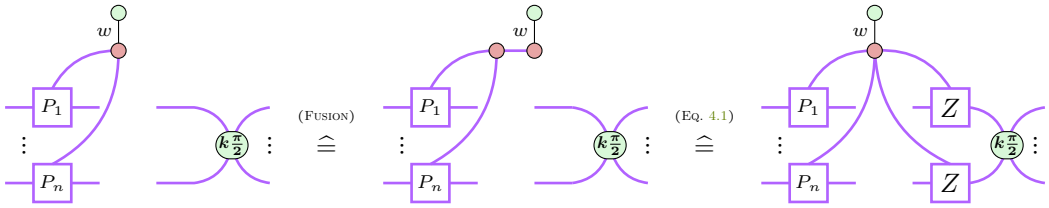
Proposition 4.12 (Adding spider stabilisers to fault gadgets)**.** Let D be a fault-free ZX diagram with a single fault gadget for a fault F. Then, for any spider stabiliser S of a spider s in D, we can use our fault-equivalent rules to add the elements of S as targets to the fault gadget.

Proof. Let s be a green spider in D with stabiliser S introducing Z flips on two of its legs and acting trivially on all other edges. Then, we can first observe:



(4.1)

Using this, we have:



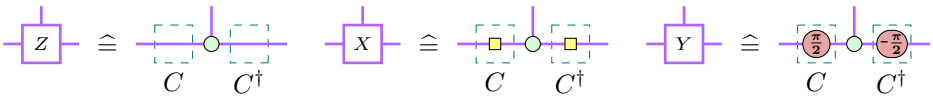
All other stabilisers can be added similarly. See Appendix A. □

But then, using this, we can go one step further:

Proposition 4.13. Let D be a fault-free ZX diagram with spacetime-equivalent faults F_1, F_2. Then, using our fault-equivalent rules, we can rewrite the fault gadget for F_1 into a fault gadget for F_2.

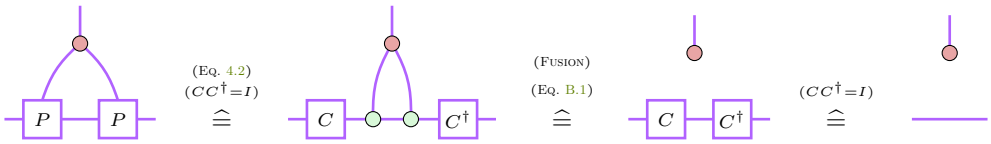
Proof. As F_1 and F_2 are spacetime-equivalent, by definition, there exists some inconsequential fault S such that $F_1 S = F_2$. But then, by Proposition 4.7, we know that there exists a set of stabilisers $S_1, \ldots, S_m$ of spiders in D such that $S_1 \ldots S_m = S$. Then, we can iteratively apply the previous proposition to add the targets of $S_1, \ldots, S_m$ to the fault gadget for F_1. We know that $F_1 S_1 \ldots S_m$ multiplies to F_2. Thus, all we have to show is that if multiple targets are on the same edge, we can merge them into a single target obeying the multiplication rules of Pauli matrices.

First, we show that if we have two targets of the same type on the same edge, they cancel each other out. To start, we see that every non-trivial Pauli box consists of a Clifford diagram C, a green spider, and the adjoint $C^\dagger$ of C:

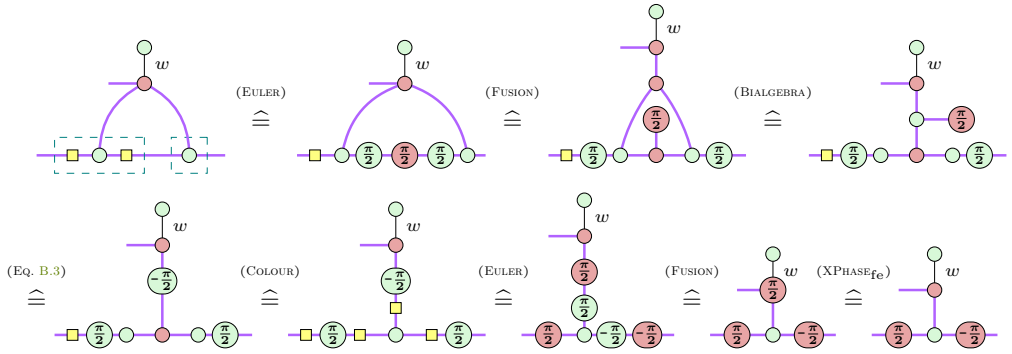


(4.2)

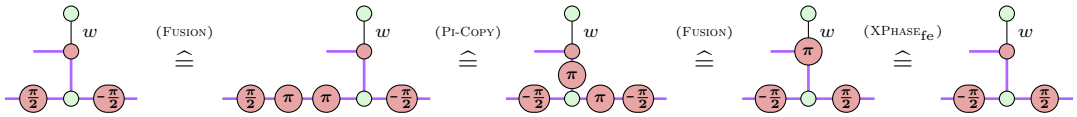
Using this, we can observe:



Next, we show that a Y target can be rewritten into an X and a Z target:



Finally, we show that we can transpose Y targets:

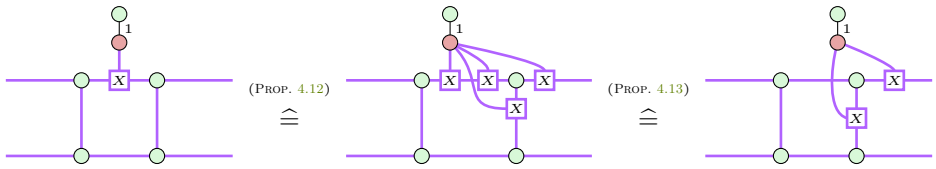


Using this, we can commute X and Z by merging them into a Y target, transposing the Y target, and then decomposing it again in the other direction.

Using these three rules, we can merge any number of targets on the same edge into a single target obeying the multiplication rules of Pauli matrices. If any of the targets are Y targets, we decompose them into their X and Z components. Then, we commute all X targets to the one side of the edge and all Z targets to the other side. Finally, we cancel out pairs of X and Z targets. We now have at most one X and one Z target on the edge. If we have both, we can replace them with a Y target. Thus, we have shown that we can merge all targets on the same edge into a single target. □

Example 4.14: Four-legged spider

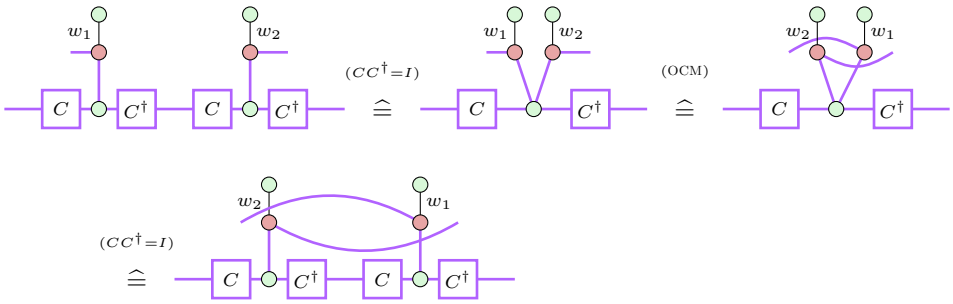
For example, we can use a spider stabiliser of the top, right spider to move the following fault gadget around the diagram:



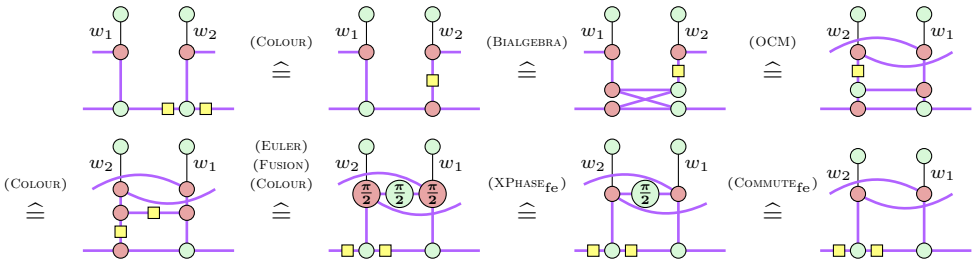
Thus, for diagrams with a single fault gadget, we can manipulate the fault gadget to represent any spacetime-equivalent fault. However, we can go one step further. For this, we first observe:

Proposition 4.15. Let D be a fault-free ZX diagram where the only allowed faults are on the spawning edges of fault gadgets. Then, we can freely commute the targets of different fault gadgets on the same edge past each other.

Proof. Intuitively, we can commute fault gadgets past each other, as we only care about diagrams up to a global phase. For completeness, we have to show that we can do this with the proposed rule set as well. For targets of the same type, we have:



For an X and a Z target, we have:



We can now commute arbitrary targets on the same edge by first decomposing all targets into their X and Z components and then commuting the components using the two rules above. □

But then, we have:

Proposition 4.16. Let D be a fault-free ZX diagram where the only allowed faults are on the spawning edges of fault gadgets, and let F_1, F_2 be spacetime-equivalent faults on D. Then, using our fault-equivalent rules, we can rewrite the fault gadget for F_1 into a fault gadget for F_2.

Proof. Similar to Proposition 4.13, we can add the targets of a generating set of stabilisers to the fault gadget for F_1. To merge all the targets, we can apply Proposition 4.15 to make sure that all targets on the same edge are next to each other. Finally, we can merge all targets on the same edge as in the previous proposition. □

4.4 Pushing Faults Out

Our goal is to push all the faults to the boundary. In this subsection, we will show that all faults have a spacetime-equivalent fault that only acts on the boundary and some very specific edge-flips.

For this, we first define:

Definition 4.17 (Flip operator collection)**.** Given a ZX diagram D with edges E, a flip operator collection is a set

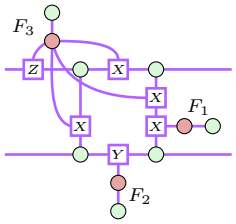
$$\mathcal{F}_{flipop} \subseteq \overline{\mathcal{P}^{|E|}}$$

such that for some basis of detecting regions $R_1, \ldots, R_d$, for each R_i, there exists a unique fault $F_{R_i} \in \mathcal{F}_{flipop}$ such that F_{R_i} anticommutes with R_i and commutes with all other R_j.

A flip operator collection can be understood as providing a basis for all operators that flip detecting regions. We note that while a particular collection contains single operators for elements of a specific basis, we can obtain flip operators for elements of any basis: Every new basis element is uniquely expressed in terms of the old basis, and a flip operator is obtained by reconstructing this decomposition with flip operators of the old basis.

Example 4.18: Four-legged spider

Since our running example only contains one detecting region, our choice of a flip operator for that region is not constrained by the commutation requirement with other regions. We could thus apply a range of different flip operators to form a valid flip operator collection. Recall that flip operators are faults, so we can visualize them using fault gadgets. Here we could choose any one of the following three example flip operators:



Note that only odd-numbered products of flip operators form another flip operator, since an even-numbered product of operators does not anticommute with the detecting region anymore. For simplicity, we will continue our example using the operator F_1 on the right, vertical wire.

We can show that such a collection always exists. Even stronger, we can find a collection that only consists of edge flips:

Proposition 4.19. Let D be a ZX diagram with d independent detecting regions, then there exists a flip operator collection consisting of d edge flips.

Proof. Let $\mathcal{R}$ be the detecting region group of the diagram D and let $R_1, \ldots, R_d$ be a minimal generating set for $\mathcal{R}$. We will construct a flip operator collection $\mathcal{F}_{flipop}$ along with a new basis $R'_1, \ldots, R'_d$ such that the i-th flip operator anticommutes with exactly R'_i.

We start with $\mathcal{F}_{flipop} = \emptyset$ and $R'_i = R_i$ for all $i = 1, \ldots, d$. Then, for $i = 1, \ldots, d$, let e_i be an edge that is highlighted by R'_i. We then create an edge flip p_i that acts non-trivially only on e_i. If e_i is highlighted by R'_i in red, we set $p_i = Z$; otherwise, we set $p_i = X$. This Pauli operator p_i anticommutes with the highlighting R'_i defined for e_i and thus placing p_i on e_i would flip *at least* R'_i.

To ensure it flips *only* R'_i, we obtain new generators for all $j \neq i$ that p_i would flip, and we update $R'_j \coloneqq R'_j R'_i$. This ensures that p_i now commutes with R'_j while preserving the fact that $R'_1, \ldots, R'_d$ generate $\mathcal{R}$. This results in p_i commuting with every generating detecting region but R'_i. Furthermore, as we only multiplied by R'_i and we have previously ensured that for all $j < i$, R'_i commutes with p_j, we have not changed whether p_j commutes or anticommutes with any of the new detecting regions.

We can iterate this procedure until we find d unique edge flips, each of which exactly flips one of the d independent detecting regions. Therefore, the constructed set of edge flips is a flip operator collection. □

But then, we can state:

Proposition 4.20. Let D be a non-zero ZX diagram with a flip operator collection $\mathcal{F}_{flipop}$ constructed w.r.t. a detecting region basis $R_1, \ldots, R_d$. Then, for any fault F on D, there exists a spacetime-equivalent fault F' that consists solely of flip operations in $\mathcal{F}_{flipop}$ and actions on the boundary.

Proof. Let $S_1, \ldots, S_m$ be stabilising Pauli webs of D such that $R_1, \ldots, R_d, S_1, \ldots, S_m$ form a basis of all Pauli webs of D. Furthermore, let F be a fault on D. Then we will construct a fault F' that satisfies the conditions above.

By the definition of a flip operator collection, for each R_i there exists a unique $F_i \in \langle \mathcal{F}_{flipop} \rangle$ that anticommutes with R_i and commutes with all other R_j. Therefore, we can construct a fault F_R that consists solely of flip operations in $\mathcal{F}_{flipop}$ and that anticommutes with exactly the same detecting regions as F. Then, F and F_R may only differ in which stabilising webs they anticommute with. Let $S_{i_1}, \ldots, S_{i_k}$ be the stabilising Pauli webs that F and F_R differ on. By stabiliser theory, we know that for each stabiliser corresponding to some Pauli web S_i there exists a destabiliser A_i that anticommutes exactly with that stabiliser. Using this, we can construct a fault F_{S_i} that consists solely of actions on the boundary and anticommutes with exactly S_i and commutes with all other stabilisers and detecting regions. Then, by construction, the fault $F' = F_R F_{S_{i_1}} \ldots F_{S_{i_k}}$ anticommutes with the same detecting regions and stabilising Pauli webs as F. Therefore, by Proposition 4.9, F' and F are spacetime-equivalent. Furthermore, by construction, F' only consists of flip operations in $\mathcal{F}_{flipop}$ and actions on the boundary. □

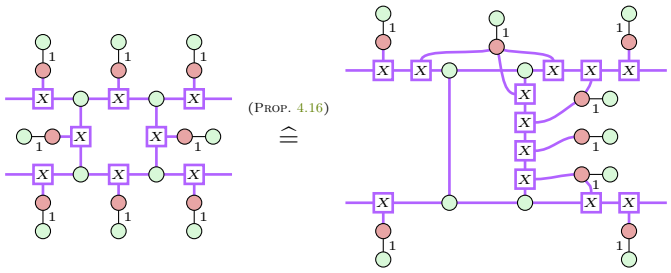
Therefore, we have:

Proposition 4.21. Let D be a non-zero ZX diagram with some noise model $\mathcal{N}$ and flip operator collection $\mathcal{F}_{flipop}$. Then, we can fault-equivalently rewrite $D_{\mathcal{N}}$ into a diagram D' such that all faults on D' act on fault gadgets whose targets consist only of $\mathcal{F}_{flipop}$ and actions on the boundary.

Proof. Recall that $D_{\mathcal{N}}$ is the diagram obtained by Proposition 4.3 to create a diagram under edge-flip noise that is fault-equivalent to D under $\mathcal{N}$. Then, by Proposition 4.20, for each fault gadget in $D_{\mathcal{N}}$, there exists a spacetime-equivalent fault gadget whose targets only act on $\mathcal{F}_{flipop}$ and the boundary. But then, by Proposition 4.16, we can rewrite each fault gadget into the corresponding fault gadget on the boundary. □

Example 4.22: Four-legged spider

Using an X flip on the right, vertical wire as our flip operator, we can now push all the fault gadgets of our diagram to only act on the boundary and flip operators:

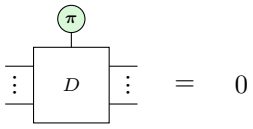


4.5 Removing Flip Operators

We have now taken major steps towards fully separating the fault gadgets from the bulk of the diagram. However, fault gadgets may still have targets in the bulk of the diagram, namely the flip operators. Therefore, as a last step to fully isolate the fault gadgets, we now have to remove these flip operators from the diagram.

First, we observe:

Proposition 4.23. Let D be a Clifford ZX diagram with boundary edges B, and let $b \in B$ be a boundary edge such that capping b with a green π-phase makes the composite diagram zero, i.e.:

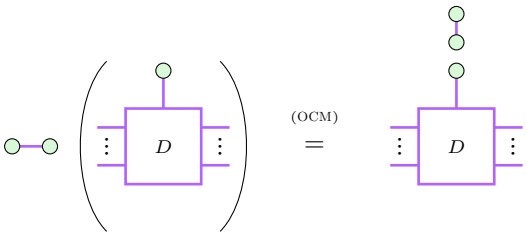


Then we can rewrite D fault-equivalently as:

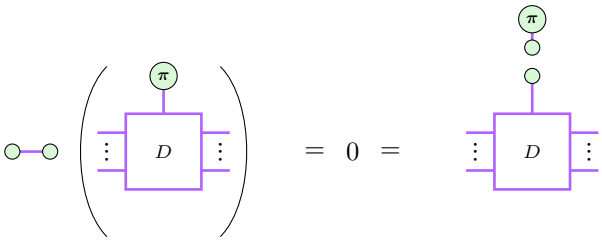
(4.3)

Proof. As the two diagrams are both idealised as fault-free, by Proposition 2.15 we only need to show that the two diagrams are semantically equivalent. Then, as our rewrite rules include the complete set of Clifford ZX rewrites for fault-free diagrams, we know that we can rewrite one into the other.

To show semantic equivalence, we employ the fact that two diagrams are semantically equivalent if and only if they have the same action on all elements of a basis [CK18]. For our purposes, we choose the X-basis, represented through green $\{0, \pi\}$-phase spiders. We have:



and



So the two diagrams have the same action on the X-basis elements.

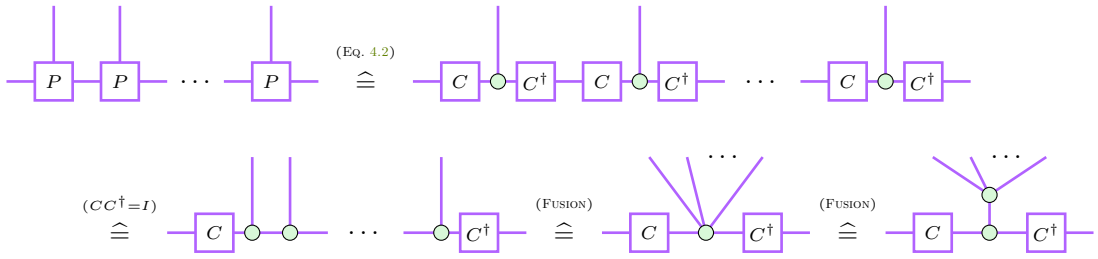
We remark that for this equality to hold, it is crucial that the actions on the basis elements are scalar accurate. Therefore, we have rescaled the LHS diagram accordingly. As the only equalities we used were OCM and the spider fusion rule, which are both scalar accurate, the equalities are also scalar accurate. But now, as the rescaled diagram is exactly equivalent to the RHS, we know that the original two diagrams are semantically equivalent up to a global scalar. □

But then we can finally state:

Proposition 4.24. Let D be a non-zero ZX diagram with some noise model $\mathcal{N}$. Then, we can fault-equivalently rewrite $D_{\mathcal{N}}$ into a diagram $D_{\mathcal{N}'}$ such that all faults drawn from $\mathcal{N}'$ are represented by fault gadgets whose targets only act on the boundary or free-floating spiders.

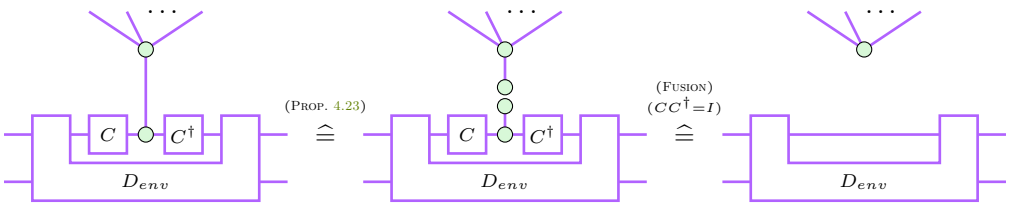
Proof. First, we use Proposition 4.19 to find a flip operator collection for D. Then, we use Proposition 4.21 to bring the diagram into a form where all faults only act on these flip operators and the boundary. Finally, we use Proposition 4.23 to remove all flip operators.

We recall that each flip operator obtained by Proposition 4.19 only acts on a single edge. Then, in preparation for the last step, we see that we can merge all targets belonging to the flip operator on such an edge:



So we effectively recover the single edge flip operator as a Pauli box that is controlled by the fault gadgets.

But then, as flip operators were constructed specifically for anticommuting with detecting regions, we know that this edge with the new Pauli box on it is in the context of a detecting region, and thus a π phase as a control on the box would make the overall diagram zero. Thus, by Proposition 4.23, where we symbolize the surrounding diagram as D_{env}, we can disconnect a free-floating spider from the remainder of the diagram:



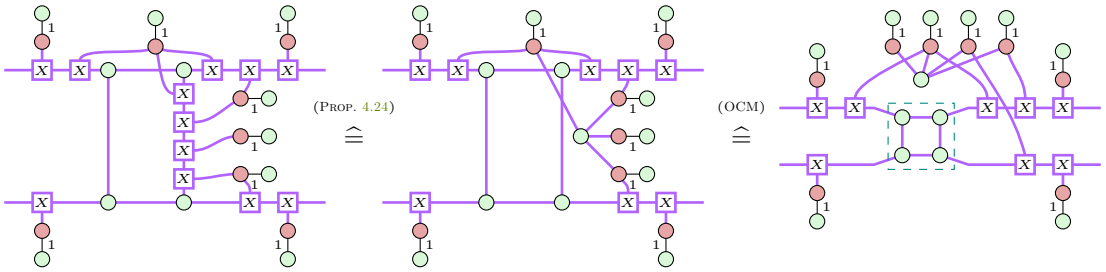
After we iteratively apply this to remove all flip operators, we have fault-equivalently rewritten the diagram into a form where all faults only act on the boundary or free-floating spiders. □

We can observe that if an odd number of faults occur that are connected to a free-floating spider, the overall diagram becomes zero. As such, the free-floating spiders are taking the role of the detecting regions, which are being detached from the original diagram. Faults, now represented

as fault gadgets, are connected exactly to those free-floating spiders that represent the detecting regions they violate. We can, therefore, read off the detector-error model [Gid21] directly from the diagram; each fault gadget represents an atomic fault, each free-floating spider a detecting region, and the connections between them represent which faults violate which detecting regions.

Example 4.25: Four-legged spider

In our example, we only have one detecting region, corresponding to one flip operator. Disconnecting this operator, we get:



We can observe that the only part of the diagram with a non-trivial semantics is the fault-free subdiagram highlighted by the dotted box. Everything else consists of fault gadgets and detecting regions which have a trivial semantics, however, non-trivial behaviour under noise.

We have now managed to completely separate the semantic part of the diagram, which is idealised as fault-free, from the faults, which are only represented through fault gadgets connected to the free-floating detecting regions and the boundary edges using only fault-equivalent rewrites. As we have previously shown that these rewrites are sound, i.e. that they preserve fault equivalence, we are guaranteed that the resulting diagram must be fault-equivalent to the diagram we started with.

5 Normal form

So far, we have rewritten our diagram such that all fault gadgets only act on the boundary or on detecting region spiders. We thus achieved a separation between the part of the diagram describing the underlying semantics and the part of the diagram describing the noise model through fault gadgets. By rewriting both parts into unique normal forms, respectively, we can achieve an overall unique normal form for the entire diagram.

The first part, describing the underlying semantics, is a fully idealised Clifford ZX diagram. Every Clifford ZX diagram has a unique normal form, namely the reduced AP form [KvdW24]. We allow all rules from [BPW17] in their fully idealised form, and as they provide a complete calculus, the reduced AP form is reachable through our axiomatisation. Thus, we can reuse this form in the fully idealised setting for the semantic part of the diagram.

It remains to show that the fault gadget part of the diagram has a unique normal form. For this, we consider how the pushed-out fault gadgets of two fault-equivalent diagrams may differ, which reduces to four challenges:

1. The fault-equivalent diagrams may have different detectable faults. Fault equivalence only requires their undetectable faults to be the same.

2. Two fault-equivalent diagrams may have different atomic faults, i.e. generating sets for all possible faults. Fault equivalence only cares about the group of all possible faults.

3. In a single diagram, multiple faults may have the same effect. For fault equivalence, we only care about the lowest-weight fault in each spacetime equivalence class.

4. Fault gadgets inherently have some ordering on the boundary edges. For fault equivalence, we only care about the set of all faults, i.e. an unordered collection.

We will tackle these challenges individually and fault-equivalently until reaching a unique normal form for fault gadget descriptions of noise models.

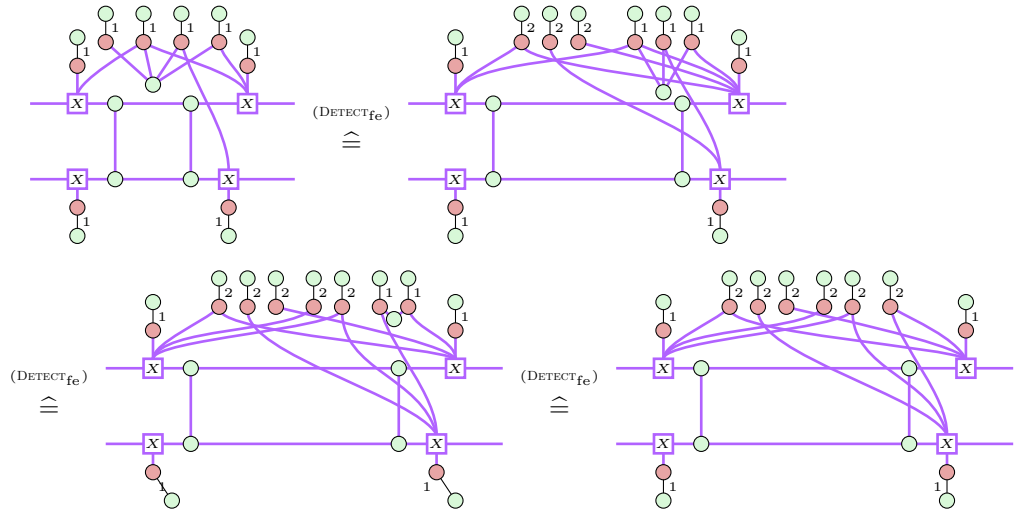
We start with the first challenge: Two fault-equivalent diagrams might have different undetectable faults. For example, in our running example, the unfused spider has a detecting region and therefore has detectable faults. However, the fused spider has no detectable faults. We have:

Proposition 5.1. Let D be a non-zero ZX diagram with a noise model $\mathcal{N}$. Then, using the fault-equivalent axioms, we can fault-equivalently rewrite $D_{\mathcal{N}}$ into D' where all fault gadgets represent undetectable faults, i.e. no fault gadget is connected to a detecting region spider.

Proof. As shown in Section 4, we can fault-equivalently rewrite $D_{\mathcal{N}}$ such that all fault gadgets only act on the boundary of the diagram and detecting region spiders. For a single detecting region spider s, $(\text{DETECT}_{\text{fe}})$ can be applied to reduce the number of fault gadgets connected to s by one, at the expense of introducing pairwise combinations with all gadgets still connected to s. This rewrite may itself be iterated until no fault gadget connects to s, leaving it as a free-floating scalar diagram equal to the scalar 2. But as we handle diagrams up to a non-zero scalar, we can simply remove s. We apply this rewrite until no such spiders remain, leaving fault gadgets that can only act on the boundary. These fault gadgets represent faults that are now undetectable. □

Example 5.2: Four-legged spider

Applying this procedure to our running example, we get:



We make use of the fact that Pauli boxes can be merged, as for example shown in the proof of Proposition 4.24, to write Pauli boxes with multiple output wires, instead of multiple Pauli boxes.

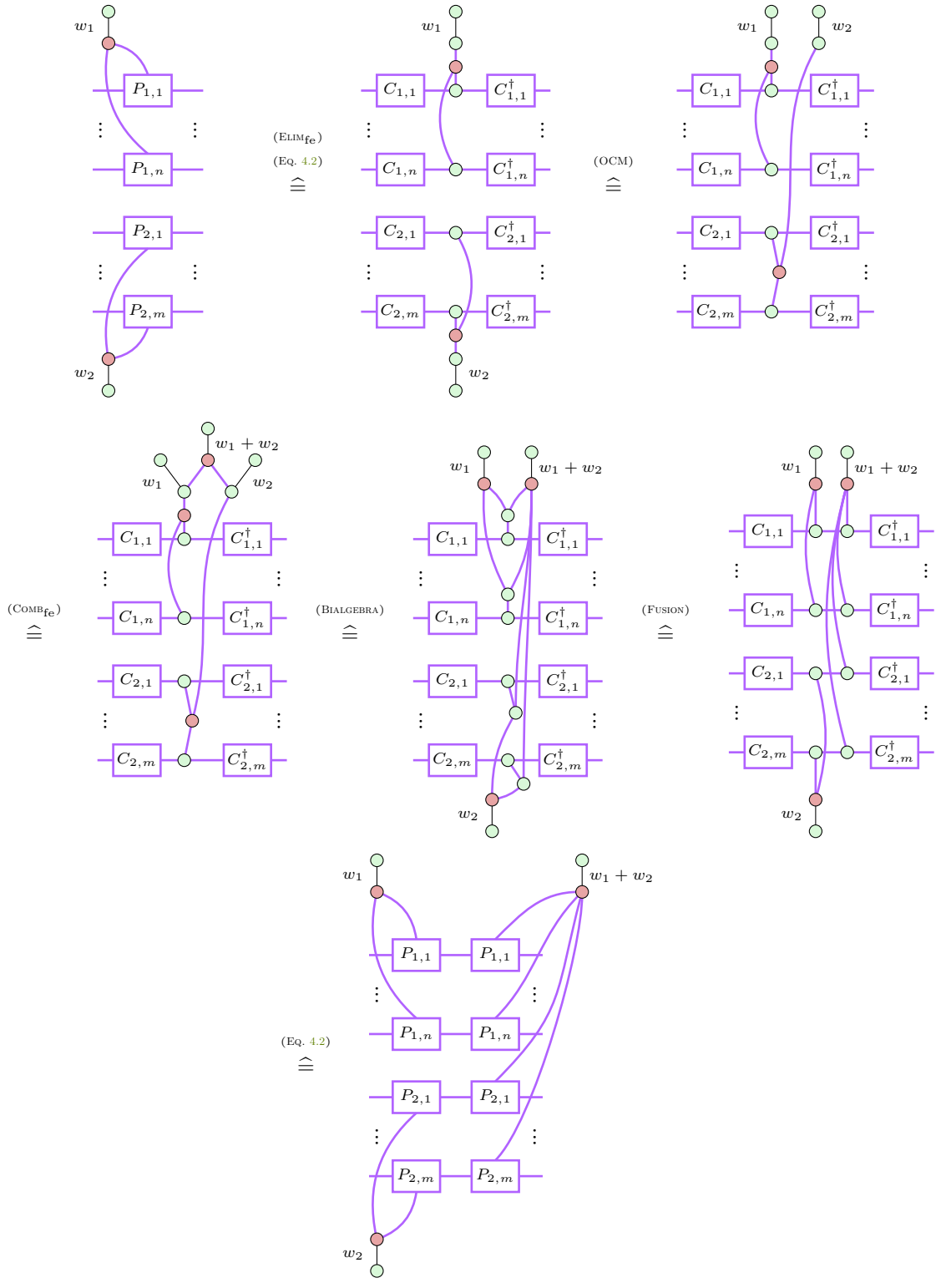
Note that when there are n faults connected to the same detecting region spider s, the number of faults generated from the above process is $\binom{n}{2}$, thus growing exponentially with n. This is expected, as checking fault equivalence is NP-hard [RPK25]. Therefore, any unique normal form is expected to require exponentially many steps to be obtained.

We are not yet guaranteed to have described *all* possible undetectable faults, i.e. we have not resolved the second challenge. In particular, fault-equivalent diagrams may have different, equivalent generating sets of atomic faults. To overcome this, in our normal form, we simply include all possible faults. Therefore, if two sets of atomic faults generate the same faults, we must end up with the same fault gadgets.

We start locally to produce combinations of two fault gadgets:

Proposition 5.3. Given two fault gadgets, we can fault-equivalently introduce their combination into the diagram using the fault-equivalent axioms.

Proof. We have:



□

To ensure that there is at least one fault gadget describing each undetectable fault, we apply Proposition 5.3 iteratively. Fault gadgets commute by Proposition 4.15, thus if we want to combine two arbitrary gadgets, we can easily make them direct neighbours.

We have:

Proposition 5.4. A diagram $D_{\mathcal{N}}$ describing its noise model using fault gadgets can be fault-equivalently rewritten into a diagram $D_{\mathcal{N}'}$ describing a noise model $\mathcal{N}' : \mathcal{F}' \to \mathbb{N}^+$, such that every undetectable fault $F \in \langle \mathcal{F}' \rangle$ is described by at least one fault gadget annotated with $\mathrm{wt}(F)$.

Proof. All undetectable faults in $\langle \mathcal{F}' \rangle$ can, by definition, be algebraically formed by multiplying elements in $\mathcal{F}'$. We can replicate any particular combination diagrammatically using fault gadgets by combining gadgets two at a time using Proposition 5.3 and applying commutation as needed with Proposition 4.15.

In particular, for a given undetectable fault F, we can include the combination that determines the weight $\mathrm{wt}(F)$, i.e. the combination that has the lowest sum of atomic weights. Replicating such a combination with fault gadgets leads to this sum as the annotation, yielding the claim. □

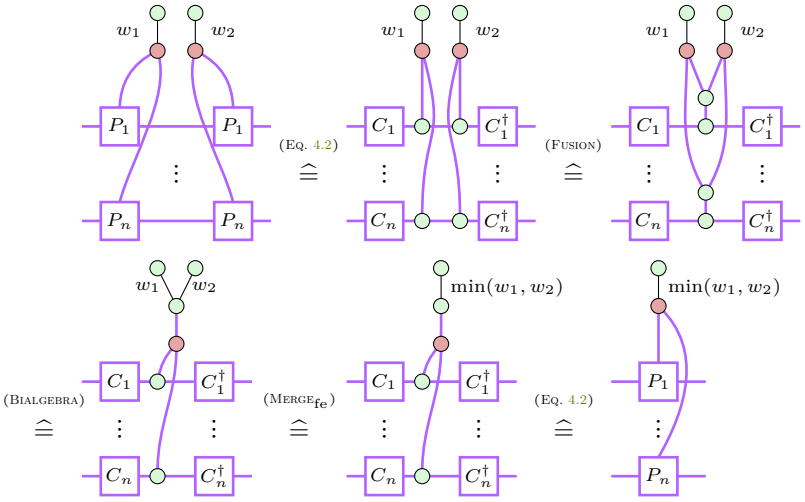
The resulting diagram enumerates all undetectable faults of $\mathcal{N}'$, and since the rewrite is fault-equivalent, it also enumerates all undetectable faults of the original noise model $\mathcal{N}$. We have thus, in addition to the first, resolved the second challenge.

Once again, this step substantially increases the number of fault gadgets. Given the size of the resulting diagram, there is little value in showing it for our running example. We will return to the example after the next step, which reduces the number of fault gadgets.

For the third challenge, we realise that two fault gadgets that have the same targets describe faults that obviously have the same effect. When checking fault equivalence, one of these faults is redundant; fault equivalence only requires that for any undetectable fault on one side, there exists an equivalent fault of at most the same weight on the other side. Therefore, if one diagram has two equivalent fault gadgets, we can fault-equivalently remove the one with higher weight. We have:

Proposition 5.5. If two fault gadgets have the same targets, we can fault-equivalently remove the one with higher weight.

Proof.



□

However, two arbitrary faults may have the same effect without being exactly the same, i.e. their fault gadgets do not have the same targets. We thus require a normalisation method for faults.

We observed in Proposition 4.10 that two undetectable faults F_1, F_2 have $D^{F_1} = D^{F_2}$ if and only if we can convert one into the other via multiplication with some inconsequential fault. That is, the effect equivalence classes of undetectable faults are exactly the spacetime equivalence classes of the noise model. Thus, once we choose some representative $F^\star$ of the spacetime equivalence class of F_1, F_2, we can utilise Proposition 4.13 to convert F_1, F_2 into $F^\star$ fault-equivalently. The representative choice is arbitrary, but for simplicity, we will choose the lexicographically smallest fault in each class.

We then obtain:

Proposition 5.6. Let $D_{\mathcal{N}}$ be a diagram describing a noise model $\mathcal{N}$ using fault gadgets. Then $D_{\mathcal{N}}$ can be fault-equivalently rewritten into a diagram $D_{\mathcal{N}'}$ describing a new noise model $\mathcal{N}'$, where each undetectable spacetime equivalence class of faults generated by $\mathcal{N}'$ has exactly one fault gadget in $D_{\mathcal{N}'}$ that is describing a class member and is annotated with the lowest weight attained in the class.

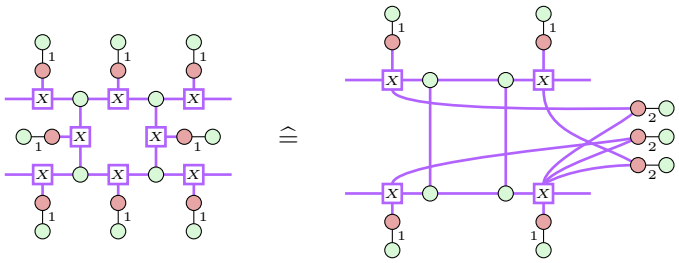
Proof. Through Proposition 5.4 we obtain an intermediate diagram $D_{\mathcal{N}''}$ where every undetectable fault has at least one describing fault gadget annotated with $\mathrm{wt}(F)$.

Now for a spacetime equivalence class of faults generated by $\mathcal{N}''$, choose some element $F^\star$ as the representative and apply Proposition 4.13 to convert every other class member fault-equivalently to $F^\star$ while keeping their annotations. This yields potentially many instances of the same fault gadget with different annotations, including one with the lowest weight attained in the class. Finally, apply Proposition 5.5 to reduce all resulting instances to a single instance, where the lowest weight naturally prevails as the annotation. □

This resolves the third challenge by reducing the number of fault gadgets to the lowest number obtainable in a fault-equivalent manner.

Example 5.7: Four-legged spider

For our running example, we have seven different equivalence classes of X-type faults:



The fourth and final challenge is about the order of fault gadgets on the boundary edges. For our normal form to be unique, we have to impose some canonical ordering on the fault gadgets. To this end, the lexicographical ordering of faults again suffices, and we can rearrange the fault gadgets via fault-equivalent commutation by Proposition 4.15 to conform to this requirement. Alternatively, one can use the merging of targets, as shown in the examples, to remove any ordering between the fault gadgets.

We now define:

Definition 5.8 (Fault normal form)**.** A diagram $D_{\mathcal{N}}$ describing its noise model $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$ using fault gadgets is in *fault normal form* if:

1. $D_{\mathcal{N}}$ has a separation of the semantics of the diagram D and fault gadgets following Section 4, **and**
2. the semantic part D inside $D_{\mathcal{N}}$ is in its unique normal form following [KvdW24], **and**
3. every spacetime equivalence class of faults in $\langle\mathcal{F}\rangle$ has exactly its lexicographically smallest member in $\mathcal{F}$, **and**
4. every fault gadget is annotated with the lowest weight attainable in the spacetime equivalence class of the fault it describes, **and**
5. the fault gadgets for all faults in $\mathcal{F}$ are ordered lexicographically.

Intuitively, the fault normal form is the diagram that is obtained by repeated application of Proposition 5.1, Proposition 5.4, Proposition 5.6, and finally ordering all fault gadgets lexicographically. The fault normal form features none of the issues outlined at the beginning of this section by construction. Furthermore, we can infer:

Proposition 5.9. Every diagram $D_{\mathcal{N}}$ describing some noise model using fault gadgets has a fault normal form that is obtained using only the rules from Fig. 3.

Proof. This follows directly from the construction outlined in this section. □

6 Completeness

We have shown that any diagram can be brought into a form that consists of two parts: a completely fault-free Clifford ZX diagram followed by an ordered layer of undetectable faults that have a unique representative for each possible fault. We call this form the *fault normal form* following Definition 5.8. The final step toward completeness is to show that any two fault-equivalent diagrams have the same fault normal form, and can thus be rewritten into each other.

6.1 Completeness for Fault Equivalence

We first state:

Proposition 6.1. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N}, \mathcal{N}'$ such that D under $\mathcal{N}$ is fault-equivalent to D' under $\mathcal{N}'$. Then $D_{\mathcal{N}}$ and $D'_{\mathcal{N}'}$ have the same fault normal form.

Proof. As fault equivalence implies semantic equivalence [RPK25], the normalised fault-free part of the two diagrams must implement the same linear map and therefore, by [Bac14a] must be the same.

For the fault gadgets, we first argue that the two diagrams in fault normal form must have the same gadgets, labelled by the same weights. As the two diagrams are fault-equivalent, they allow for the same group of undetectable faults. Therefore, they must have exactly the same equivalence classes of faults and therefore exactly the same fault gadgets. The fault gadgets are labelled by the minimum weight of all faults in an equivalence class. Fault equivalence requires those minimum representatives between two diagrams to have the same weight, as otherwise, there would exist an undetectable fault on one diagram that does not have a corresponding fault on the other diagram of at most the same weight, violating the assumption that the two diagrams are fault-equivalent. But then, the fault gadgets of the two diagrams must not only be the same but also labelled with the same weight. Finally, as all fault gadgets must be different — we have previously eliminated duplicates — the lexicographical ordering is unique and must therefore be the same for both diagrams, and thus the diagrams must be the same. □

Finally, it remains to show our main result:

Theorem 6.2. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N}, \mathcal{N}'$ such that D under $\mathcal{N}$ is fault-equivalent to D' under $\mathcal{N}'$. Then, only using the rules in Fig. 3, $D_{\mathcal{N}}$ can be rewritten into $D'_{\mathcal{N}'}$.

Proof. Via Proposition 5.9, we can rewrite both $D_{\mathcal{N}}$ and $D'_{\mathcal{N}'}$ into their fault normal form. By Proposition 6.1, the normal form must be the same. We can thus first rewrite $D_{\mathcal{N}}$ into normal form, and then rewrite the normal form into $D'_{\mathcal{N}'}$ by reversing the sequence of rewrites used to bring $D'_{\mathcal{N}'}$ into normal form. Therefore, we have diagrammatically shown their fault equivalence. □

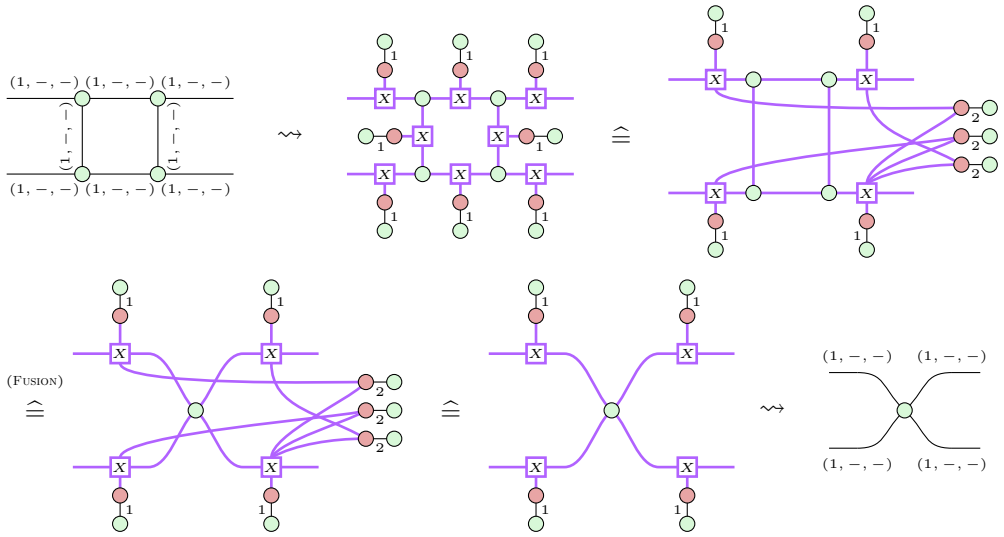
As we have previously shown that our rules are sound, we have now shown that the rule set from Fig. 3 is sound and complete.

(a) (b)

Figure 4: Axioms that, in addition to Fig. 3, are required for a calculus that is complete for (a) w-fault equivalence and (b) fault boundedness.

Example 6.3: Four-legged spider

For our running example, we now rewrite the two diagrams into one another:



6.2 Completeness for w-Fault Equivalence and Fault Boundedness

Adding some additional rules, we can easily extend the completeness result to other relationships of noisy ZX diagrams. We first show how adding one more rule gives us a complete rule set for w-fault boundedness. Then we show that a different rule gives completeness for fault boundedness. Finally, we show that together, these two rules extend the completeness to w-fault boundedness.

The two rules that are necessary to extend the completeness results are shown in Fig. 4. The first rule is necessary to have completeness for w-fault equivalence, and the second rule is necessary for fault boundedness. For the second rule, we note that by $w = -$ we mean that no faults are allowed on the corresponding edge, i.e. that it is idealised as fault-free. An idealised edge is always fault-bounded by a non-idealised one, in that, as there are no allowed faults, any fault on the idealised edge trivially has a correspondence on the non-idealised one. Therefore, in that case, the rewrite can introduce a fault with any weight.

Similar to the rules complete for fault equivalence, we must show that these additional rewrite rules are sound:

Proposition 6.4. The rule Fig. 4 (a) is sound for w-fault equivalence, and the rule Fig. 4 (b) is sound for fault boundedness. Both rules are sound for w-fault boundedness.

Proof. See Appendix A. □

We note that every fault-equivalent rule by definition also implies weaker relationships like w-fault equivalence. So in particular the previously introduced rewrite rules from Fig. 3 are sound w.r.t. such weaker relationships.

Again, since w-fault boundedness is compositional and transitive, and so are each of its derived properties, we receive a usable rewrite system when including either of or both of the rules. So we can now turn to completeness.

First, we show:

Theorem 6.5. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$, $\mathcal{N}' : \mathcal{F}' \to \mathbb{N}^+$ such that D under $\mathcal{N}$ is w-fault-equivalent to D' under $\mathcal{N}'$. Then, only using the rules in Fig. 3 and Fig. 4 (a), $D_{\mathcal{N}}$ can be rewritten into $D'_{\mathcal{N}'}$.

Proof. Using the rules from Fig. 3, we can fault-equivalently bring both diagrams into their unique fault normal form. As D and D' are only w-fault-equivalent, there might be fault equivalence classes that are present on one diagram but not the other, or where the fault gadgets have different values; however, only for classes where the minimum attained weight is at least w. For all equivalence classes with minimum attained weight less than w, the two diagrams must be the same. We can now apply the rule from Fig. 4 (a) together with fault-free Clifford rules to remove all other fault gadgets annotated with weight w or more. The only remaining fault gadgets must be the same on both sides, and therefore, the two modified normal forms must be the same. But then, only using our diagrammatic rewrite rules, we can rewrite one diagram into the other. □

Similarly, we can show:

Theorem 6.6. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$, $\mathcal{N}' : \mathcal{F}' \to \mathbb{N}^+$ such that D under $\mathcal{N}$ is fault-bounded by D' under $\mathcal{N}'$. Then, only using the rules in Fig. 3 and Fig. 4 (b), $D_{\mathcal{N}}$ can be rewritten into $D'_{\mathcal{N}'}$.

Proof. Similar to above, we can bring both diagrams into their fault normal form using only our fault-equivalent rewrites. As D is fault-bounded by D', any fault gadget on D must have at most the same weight as the corresponding fault gadget on D'. But then, we can use the rewrite from Fig. 4 (b) to increase the weight of all the fault gadgets in $D_{\mathcal{N}}$ that have a lower weight than their corresponding gadget in $D'_{\mathcal{N}'}$. If there are faults that are possible on $D'_{\mathcal{N}'}$ but not on $D_{\mathcal{N}}$, we can use the fault-free rules to introduce a fault-free fault gadget that corresponds to that fault and then use Fig. 4 (b) to unidealise the spawning edge. But then, we have a directed rewrite from $D_{\mathcal{N}}$ to $D'_{\mathcal{N}'}$ using only the allowed rules. □

Finally, using the two rules, we can state:

Theorem 6.7. Let D, D' be Clifford ZX diagrams with respective noise models $\mathcal{N} : \mathcal{F} \to \mathbb{N}^+$, $\mathcal{N}' : \mathcal{F}' \to \mathbb{N}^+$ such that D under $\mathcal{N}$ is w-fault-bounded by D' under $\mathcal{N}'$. Then, only using the rules in Fig. 3 and Fig. 4, $D_{\mathcal{N}}$ can be rewritten into $D'_{\mathcal{N}'}$.

Proof. As D is w-fault-bounded by D', we are guaranteed that all faults of weight less than w on D' have a corresponding fault of at least the same weight on D'. Therefore, just as for the previous two proofs, we can first remove all faults of weight more than w from $D_{\mathcal{N}}$ using Fig. 4 (a) and then increase the weight and introduce the missing ones to rewrite the normal form of $D_{\mathcal{N}}$ into the normal form of $D'_{\mathcal{N}'}$. □

7 Conclusion

In this work, we provided an axiomatisation of a ZX calculus that is sound and complete for showing fault equivalence between Clifford diagrams. At the centre of the diagrammatic implementation we utilised fault gadgets to represent and manipulate faults. Using the fault-equivalent rewrites, we provided a method to push out all the faults, separating the semantic part of the diagram from the noisy part. This allowed us to compare two different ZX diagrams by independently looking at each of the two parts. Leveraging existing completeness results for fault-free Clifford ZX diagrams [Bac14a; BPW17; KvdW24], our work primarily focused on comparing the noisy part of the diagram. We showed that, using fault-equivalent rewrites, the fault gadget on the diagrams can be rewritten to have one representative fault gadget per spacetime equivalence class of faults.

Beyond the theoretical implications of the completeness results, their constructive proofs offer insight into an algorithm for automatic fault equivalence checking. This work is accompanied by an implementation [Rüs25b] that uses fault gadgets both for internal representations and visualization of faults on diagrams, providing e.g. an automated variant of checking fault equivalence and an

interface for extracting a Tanner graph. The implementation does not exactly follow the rewrite rules from Fig. 3, but instead it contains many optimisations and simplifications that are possible when allowing some transformations outside the ZX calculus. The approach to checking fault equivalence derived from the constructive proof has an exponential complexity advantage over the naive approach [Rüs25a]. Instead of scaling exponentially in the number of faults, this algorithm scales in the number of physical qubits and detecting regions.

This work provides several avenues for future work, of both a practical and theoretical nature. One way to view the procedure presented in this work is that when we manipulate the fault gadgets, we fault-equivalently change the noise model the diagram is exposed to. For example, after pushing out the fault gadgets, we can view the result as the original diagram under a different, fault-equivalent noise model, where now all faults only act on the boundary and the flip operators. As pointed out at the end of Section 4, this allows us, for example, to extract the detector-error model [Gid21; Der+24] of the circuit under the given noise model. However, we can consider that after eliminating all the detecting regions, the remaining faults are the undetectable ones. This allows us to infer the distribution of undetectable faults induced by the noise model, from which we can derive other properties such as the circuit distance. In future research, it would be interesting to focus more on the insights that can be gained via such procedures and explore whether they can be leveraged into efficient circuit analysis tools.

Furthermore, the ZX calculus is usually considered in the case of qubits, i.e. diagrams evaluate to linear maps moving between products of two-dimensional Hilbert spaces. However, the case of higher-dimensional systems has gathered some recent attention [Ran14; Wan22], especially for prime-dimensional ZX calculi which are proven to have a complete Clifford fragment [BC22; Poó+23]. Similarly, continuous-variable variations of the ZX calculi have been developed and applied to the context of quantum error correction [SYG24; BCC24; Nag+25]. A natural direction for future research is to lift the completeness results for fault-equivalent rewrites to other calculi. Among other things, this would require generalising fault gadgets and spacetime equivalence.

Additionally, fault equivalence, as defined in this work, considers weighted Pauli noise. While this captures a common noise model assumed in the literature, an interesting avenue for future work would be to generalise fault equivalence beyond weighted Pauli noise to other noise models, such as stochastic Pauli noise. For this, the complete set of rewrites presented in this work provides a promising starting point. Once other noise models are proposed, completeness results for fault equivalence under these noise models would probably be derived from this work.

Finally, the implementation accompanying this work is still in early stages, and usability as well as performance can be considerably improved. More features could be added, such as integration with other QEC libraries like the highly-parallel sampling library `stim` [Gid21]. Additionally, as research progresses, support for alternative noise models or circuits beyond qubits could be considered. Considering the exponential cost of the proposed algorithm adaptations for highly structured, larger circuits could be added. For example, substantial improvements could be achieved for circuits that are composed of smaller gadgets.

Overall, this work is an important step in understanding diagrammatic calculi for fault equivalence while providing exciting new avenues for analysing and understanding the behaviour of circuits under noise that ranges beyond completeness results.

Acknowledgements

We would like to thank Șerban Cercelescu for the useful discussions that led to the inspiration for the pushing out procedure. We are grateful to Boldizsár Poór, Julio Magdalena De La Fuente, and Maximilian Schweikart for the discussions, inspiration, and feedback on various drafts of this work. BR thanks Simon Harrison for his generous support for the Wolfson Harrison UK Research Council Quantum Foundation Scholarship. AK is supported by the Engineering and Physical Sciences Research Council grant number EP/Z002230/1, "(De)constructing quantum software (DeQS)". BR is employed part-time by Quantinuum. The wording in some sections of this paper has been refined using LLMs.

References

[Bac+17] Dave Bacon et al. "Sparse Quantum Codes From Quantum Circuits". In: *IEEE Transactions on Information Theory* 63.4 (2017), pp. 2464–2479. DOI: 10.1109/TIT.2017.2663199.

[Bac14a] Miriam Backens. "The ZX-calculus Is Complete for Stabilizer Quantum Mechanics". In: *New Journal of Physics* 16.9 (Sept. 17, 2014), p. 093021. ISSN: 1367-2630. DOI: 10.1088/1367-2630/16/9/093021. arXiv: 1307.7025 [quant-ph].

[Bac14b] Miriam Backens. "The ZX-calculus is complete for the single-qubit Clifford+T group". In: *Electronic Proceedings in Theoretical Computer Science* 172 (Dec. 2014), pp. 293–303. ISSN: 2075-2180. DOI: 10.4204/eptcs.172.21.

[BC22] Robert I. Booth and Titouan Carette. "Complete ZX-Calculi for the Stabiliser Fragment in Odd Prime Dimensions". In: *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*. Ed. by Stefan Szeider, Robert Ganian, and Alexandra Silva. Vol. 241. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 24:1–24:15. ISBN: 978-3-95977-256-3. DOI: 10.4230/LIPIcs.MFCS.2022.24.

[BCC24] Robert I. Booth, Titouan Carette, and Cole Comfort. *Complete equational theories for classical and quantum Gaussian relations*. 2024. arXiv: 2403.10479 [cs.LO].

[BH25] Keller Blackwell and Jeongwan Haah. *The code distance of Floquet codes*. 2025. arXiv: 2510.05549 [quant-ph].

[Bom+24] Hector Bombin et al. "Unifying Flavors of Fault Tolerance with the ZX Calculus". In: *Quantum* 8 (June 18, 2024), p. 1379. ISSN: 2521-327X. DOI: 10.22331/q-2024-06-18-1379. arXiv: 2303.08829 [quant-ph].

[Bor19] Coen Borghans. "ZX-calculus and Quantum Stabilizer Theory". MA thesis. Radboud University, 2019. URL: https://www.cs.ox.ac.uk/people/aleks.kissinger/papers/borghans-thesis.pdf (visited on 08/06/2025).

[BPW17] Miriam Backens, Simon Perdrix, and Quanlong Wang. "A Simplified Stabilizer ZX-calculus". In: *Electronic Proceedings in Theoretical Computer Science* 236 (Jan. 2017), pp. 1–20. ISSN: 2075-2180. DOI: 10.4204/eptcs.236.1.

[CD11] Bob Coecke and Ross Duncan. "Interacting Quantum Observables: Categorical Algebra and Diagrammatics". In: *New Journal of Physics* 13.4 (Apr. 14, 2011), p. 043016. ISSN: 1367-2630. DOI: 10.1088/1367-2630/13/4/043016. arXiv: 0906.4725 [quant-ph].

[CK18] Bob Coecke and Aleks Kissinger. "Picturing quantum processes: A first course on quantum theory and diagrammatic reasoning". In: *International conference on theory and application of diagrams*. Springer. 2018. URL: https://www.cs.ox.ac.uk/people/aleks.kissinger/PQP.pdf.

[Cow+20] Alexander Cowtan et al. "Phase Gadget Synthesis for Shallow Circuits". In: *Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019*. Ed. by Bob Coecke and Matthew Leifer. Vol. 318. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2020, pp. 213–228. DOI: 10.4204/EPTCS.318.13.

[Der+24] Peter-Jan H. S. Derks et al. *Designing Fault-Tolerant Circuits Using Detector Error Models*. Dec. 25, 2024. DOI: 10.48550/arXiv.2407.13826. Pre-published.

[DP23] Nicolas Delfosse and Adam Paetznick. *Spacetime codes of Clifford circuits*. 2023. arXiv: 2304.05943 [quant-ph].

[Dun+20] Ross Duncan et al. "Graph-Theoretic Simplification of Quantum Circuits with the ZX-calculus". In: *Quantum* 4 (June 2020), p. 279. ISSN: 2521-327X. DOI: 10.22331/q-2020-06-04-279.

[Gid21] Craig Gidney. "Stim: a fast stabilizer circuit simulator". In: *Quantum* 5 (July 2021), p. 497. ISSN: 2521-327X. DOI: 10.22331/q-2021-07-06-497.

[Got09] Daniel Gottesman. *An Introduction to Quantum Error Correction and Fault-Tolerant Quantum Computation*. Apr. 16, 2009. arXiv: 0904.2557 [quant-ph]. Pre-published.

[Got22] Daniel Gottesman. *Opportunities and Challenges in Fault-Tolerant Quantum Computation*. 2022. arXiv: 2210.15844 [quant-ph].

[Got97] Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. 1997. arXiv: quant-ph/9705052 [quant-ph].

[Jav+24] Ali Javadi-Abhari et al. *Quantum computing with Qiskit*. 2024. arXiv: 2405.08810 [quant-ph].

[KvdW20] Aleks Kissinger and John van de Wetering. "PyZX: Large Scale Automated Diagrammatic Reasoning". In: *Proceedings 16th International Conference on Quantum Physics and Logic, Chapman University, Orange, CA, USA., 10-14 June 2019*. Ed. by Bob Coecke and Matthew Leifer. Vol. 318. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2020, pp. 229–241. DOI: 10.4204/EPTCS.318.14.

[KvdW24] Aleks Kissinger and John van de Wetering. *Picturing Quantum Software: An Introduction to the ZX-calculus and Quantum Compilation*. Preprint, 2024. URL: https://github.com/zxcalc/book.

[Mag+25] Julio C. Magdalena de la Fuente et al. "Ruby Code: Making a Case for a Three-Colored Graphical Calculus for Quantum Error Correction in Spacetime". In: *PRX Quantum* 6.1 (Mar. 2025). ISSN: 2691-3399. DOI: 10.1103/prxquantum.6.010360.

[Nag+25] Hironari Nagayoshi et al. "ZX graphical calculus for continuous-variable quantum processes". In: *Physical Review Research* 7.3 (Aug. 2025). ISSN: 2643-1564. DOI: 10.1103/physrevresearch.7.033141.

[Nam+18] Yunseong Nam et al. "Automated optimization of large quantum circuits with continuous parameters". In: *npj Quantum Information* 4.1 (2018), p. 23. DOI: https://doi.org/10.1038/s41534-018-0072-4.

[NC10] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge: Cambridge University Press, 2010.

[NW17] Kang Ng and Quanlong Wang. "A universal completion of the ZX-calculus". In: (June 2017). DOI: 10.48550/arXiv.1706.09877.

[Poó+23] Boldizsár Poór et al. "The Qupit Stabiliser ZX-travaganza: Simplified Axioms, Normal Forms and Graph-Theoretic Simplification". In: (2023). arXiv: 2306.05204 [quant-ph].

[Ran14] André Ranchin. "Depicting qudit quantum mechanics and mutually unbiased qudit theories". In: *Electronic Proceedings in Theoretical Computer Science* 172 (Dec. 2014), pp. 68–91. ISSN: 2075-2180. DOI: 10.4204/eptcs.172.6.

[RPK24] Benjamin Rodatz, Boldizsár Poór, and Aleks Kissinger. *Floquetifying Stabiliser Codes with Distance-Preserving Rewrites*. Dec. 16, 2024. arXiv: 2410.17240 [quant-ph]. Pre-published.

[RPK25] Benjamin Rodatz, Boldizsár Poór, and Aleks Kissinger. *Fault Tolerance by Construction*. 2025. arXiv: 2506.17181 [quant-ph].

[Rüs25a] Maximilian Rüsch. "Completeness for Fault Equivalence of Clifford ZX Diagrams". MA thesis. University of Oxford, 2025. URL: https://maximilianruesch.de/scientific/msc-thesis/msc-thesis.pdf (visited on 10/10/2025).

[Rüs25b] Maximilian Rüsch. *Project code for this work*. 2025. URL: https://github.com/maximilianruesch/fault-gadgets.

[Sta+24] Korbinian Staudacher et al. "Multi-Controlled Phase Gate Synthesis with ZX-calculus Applied to Neutral Atom Hardware". In: *Proceedings of the 21st International Conference on Quantum Physics and Logic, Buenos Aires, Argentina, July 15-19, 2024*. Ed. by Alejandro Díaz-Caro and Vladimir Zamdzhiev. Vol. 406. Electronic Proceedings in Theoretical Computer Science. Open Publishing Association, 2024, pp. 96–116. DOI: 10.4204/EPTCS.406.5.

[SYG24] Razin A. Shaikh, Lia Yeh, and Stefano Gogioso. *The Focked-up ZX Calculus: Picturing Continuous-Variable Quantum Computation*. 2024. arXiv: 2406.02905 [quant-ph].

[vdWet20] John van de Wetering. *ZX-calculus for the working quantum computer scientist*. 2020. arXiv: 2012.13966 [quant-ph].

[Vil19] 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)*. 2019, pp. 1–10. DOI: 10.1109/LICS.2019.8785765.

[Wan22] Quanlong Wang. *Qufinite ZX-calculus: a unified framework of qudit ZX-calculi*. 2022. arXiv: 2104.06429 [quant-ph].

A Missing proofs

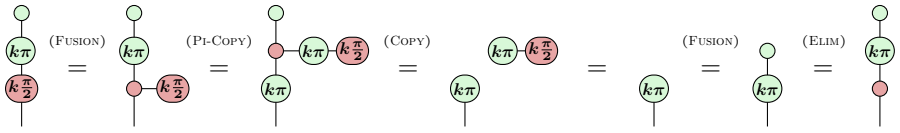
Proposition 3.1. The rules displayed in Fig. 3 are sound with respect to fault equivalence.

Proof. In this proof, we refer to the sides of the rules in their displayed form as LHS and RHS. Furthermore, whenever we find an atomic Z flip with weight w_i on diagram D, we will refer to it as $Z_{D,i}$, and to the faulty diagram as $D^{Z_{D,i}}$.

Clifford Rules For each such rule, both the LHS and RHS are fully idealised. Thus, Proposition 2.15 is applicable to yield the claim directly.

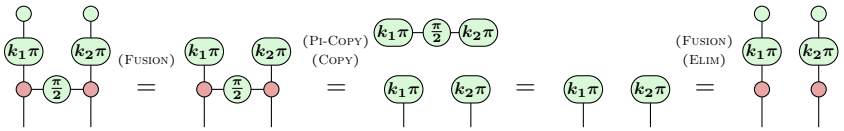
(ElimFE) The atomic faults on the LHS have a one-to-one correspondence to the RHS, such that for each atomic fault the respective faulty diagram is the same. This implies that the group of possible faults generated by both diagrams must be isomorphic, and since the weight annotation does not change, the induced weight function is also the same. Thus, there cannot exist any atomic fault on one side that does not have a correspondence or even a different atomic weight on the other, and so it can also not be the case with non-atomic faults.

(XPhaseFE) Both sides only allow either the trivial fault or a single Z flip to occur. Then, we can observe that for both faults, the faulty diagrams are directly the same:



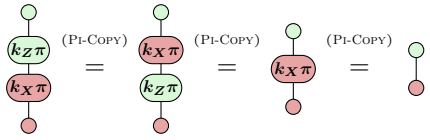
Furthermore, the annotated weight of Z_{LHS} and Z_{RHS} is the same, so the induced weight function must also be the same, yielding fault equivalence.

(CommuteFE) The proof is analogous to the previous case, in that we can again reason about the faulty diagrams:



We are able to remove the scalar in the intermediate step as there is no k_1, k_2 where the scalar is zero, and we do not distinguish faulty diagrams up to a scalar.

(ScalarFE) The RHS is fully idealised and thus trivially fault-bound by the LHS. Furthermore, we can observe that all faults on the LHS have faulty diagrams equal to the trivial fault I:



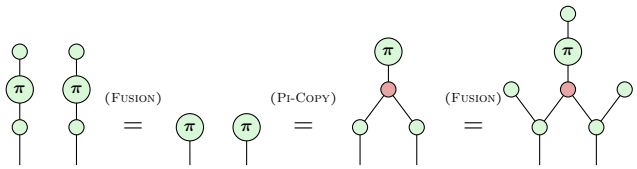
Since $\mathrm{wt}(I) = 0 \leq \mathrm{wt}(F)$ for any fault F on the LHS, every fault has a correspondence with less or equal weight, providing fault equivalence by definition.

(MERGE$_{\text{fe}}$) Via repeated application of (FUSION), both $Z_{\mathsf{LHS},1}, Z_{\mathsf{LHS},2}$ have a faulty diagram equivalent to that of as Z_{RHS}, while their product and the trivial fault for the LHS have an equivalent faulty diagram to the trivial fault on the RHS. So every fault that can be generated on either side has at least one corresponding fault on the other, thus checking their induced weight remains. But everything corresponding to a trivial fault forms a straightforward case, and by definition of the minimum, it holds that

$$\forall i \in 1,2 : \min(w_1, w_2) \leq w_i \quad \text{and} \quad \exists i \in 1,2 : w_i \leq \min(w_1, w_2)\,,$$

providing the other cases.

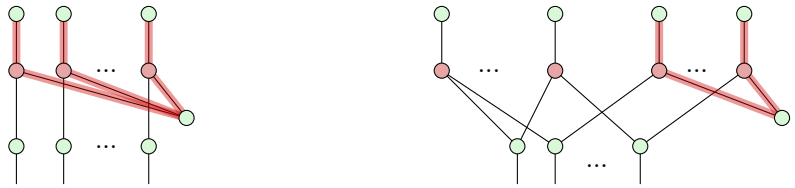
(COMB$_{\text{fe}}$) Via application of (COPY) and (FUSION), we see that both $Z_{\mathsf{LHS},1}, Z_{\mathsf{LHS},2}$ have one-to-one correspond to $Z_{\mathsf{RHS},1}, Z_{\mathsf{RHS},2}$, and so their product also has a correspondence to the product on the other side. We then consider $Z_{\mathsf{RHS},12}$, and see that it forms an additional possibility to correspond to the product on the LHS:



Thus, any combination on the RHS is reproducible on the LHS, so arguing about their induced weight remains.

On both sides, the atomic faults weighted with w_1, w_2 are already annotated with their induced weight. Additionally, the atomic fault weighted with $w_1 + w_2$ on the RHS corresponding to the product on the LHS is exactly weighted with the weight sum of said product. Thus, every atomic fault on both sides has a correspondence of exactly equal weight. Any non-atomic fault F has at least one atomic fault with the same faulty diagram, and it is straightforward to check that the weight of F is at least that of the atomic fault. Thus, every fault on either side has a correspondence of at most the same weight.

(DETECT$_{\text{fe}}$) First, we see that the faults annotated with $w_2, \ldots, w_n$ on the LHS have a direct correspondence to the faults on the RHS, and vice versa. For $Z_{\mathsf{LHS},1}$, observe that the free-floating spider on either side forms a detecting region with the annotated edges:



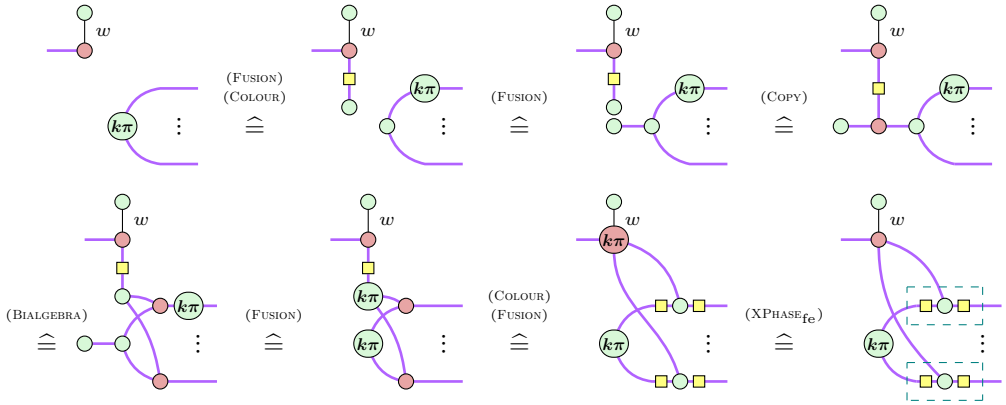
Thus, $Z_{\mathsf{LHS},1}$ in particular is detectable. Further, observe that the faults annotated with sums on the RHS generate the faulty diagrams of all undetectable combinations of $Z_{\mathsf{LHS},1}$ with other atomic faults. This implies that removing/introducing $Z_{\mathsf{LHS},1}$ connected to the free-floating spider does not remove/introduce any new class of possible faulty diagrams. The reasoning about weights is directly analogous to the case of (DETECT$_{\text{fe}}$), and so fault equivalence follows.

□

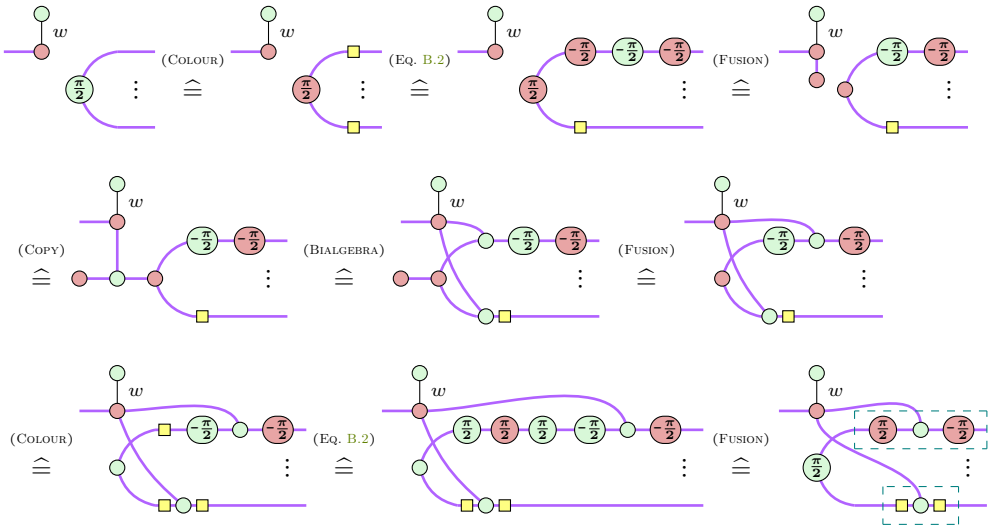
Proposition 4.12 (Adding spider stabilisers to fault gadgets)**.** Let D be a fault-free ZX diagram with a single fault gadget for a fault F. Then, for any spider stabiliser S of a spider s in D, we can use our fault-equivalent rules to add the elements of S as targets to the fault gadget.

Proof. In the main body, we have already shown how to add a ZZ stabiliser to a green spider. The other cases follow similarly: For a green spider with a phase of $k \in \{0, \pi\}$, we can add an X

stabiliser as follows:

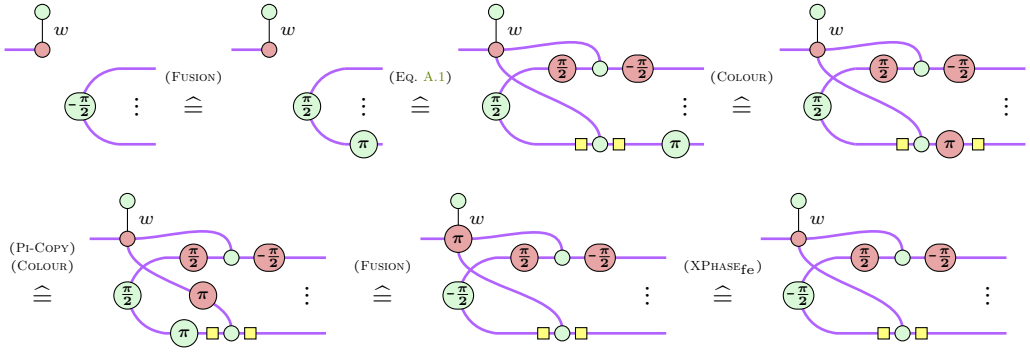


For a green spider with a phase of $\frac{\pi}{2}$, we can add an all X and a single Y stabiliser as follows:



(A.1)

For a green spider with a phase of $-\frac{\pi}{2}$, we can add an all X and a single Y stabiliser as follows:



As these generate all spider stabilisers, we can now add any generating set of spider stabilisers to the fault gadget. Finally, to complete the proof, we have to merge all the targets as shown in Proposition 4.13. □

Proposition 6.4. The rule Fig. 4 (a) is sound for w-fault equivalence, and the rule Fig. 4 (b) is sound for fault boundedness. Both rules are sound for w-fault boundedness.

Proof. For any two diagrams D_1, D_2 under any two noise models, both $D_1 \mathrel{\underset{w}{\widehat{=}}} D_2$ and $D_1 \mathrel{\underset{w}{\widehat{\leq}}} D_2$ imply $D_1 \mathrel{\underset{w}{\widehat{\leq}}} D_2$. So if both rules are sound w.r.t. their respective properties, both are also sound w.r.t. w-fault boundedness individually. We show their individual soundness:

(Limit$_{\text{fe}}$) The RHS is fully idealised and thus trivially fault-bound by the LHS. As the single non-trivial atomic fault on the LHS is by the side condition at or above weight w, checking w-fault equivalence reduces to finding a correspondence for the trivial fault I_{LHS}, which is naturally I_{RHS} with a straightforward weight check.

(Bound$_{\text{fb}}$) We distinguish the cases $w = -$ and $w \geq w'$. In the first case, $w = -$ is just a more explicit version of an idealised edge, yielding a fully idealised LHS and thus leading to the claim trivially.

In the second case, the only non-trivial faults on both sides are Z_{LHS} and Z_{RHS}, and they clearly have the same faulty diagram. Via the assumed side condition, it holds that $\mathrm{wt}(Z_{\mathsf{LHS}}) = w \geq w' = \mathrm{wt}(Z_{\mathsf{RHS}})$. Thus, for all $\widetilde{(w)} \in \mathbb{N}^+$, all conditions for $\widetilde{w}$-fault boundedness are fulfilled, yielding the claim.

□

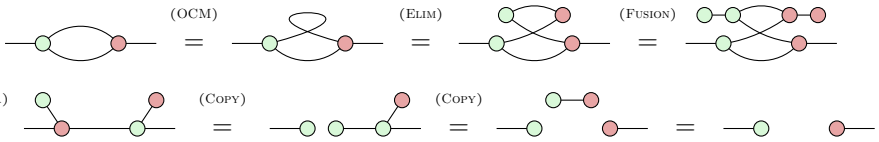
B Additional Derived ZX Calculus Rules

We introduce three additional ZX calculus rules, derivable from our axiomatisation of the ZX calculus:

Proposition B.1 (Hopf Rule)**.**

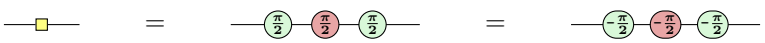
(B.1)

Proof.



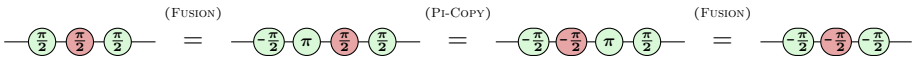
□

Proposition B.2 (Alternative Hadamard Decomposition)**.**



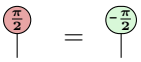
(B.2)

Proof. The first equality holds by definition of the Hadamard gate. The second equality is easily shown in the ZX calculus:



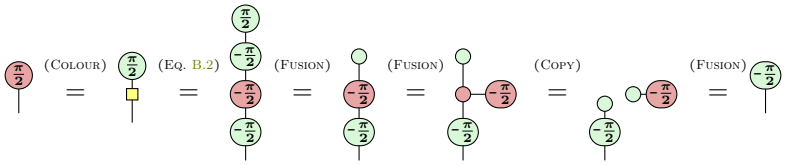
□

Proposition B.3 (Transformation of one-legged spiders from [BPW17])**.**



(B.3)

Proof.



□