Relating Measurement Patterns to Circuits via Pauli Flow

Will Simmons

Cambridge Quantum Computing Ltd
9a Bridge Street, Cambridge, UK

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, UK

`will.simmons@cambridgequantum.com`

The one-way model of Measurement-Based Quantum Computing and the gate-based circuit model give two different presentations of how quantum computation can be performed. There are known methods for converting any gate-based quantum circuit into a one-way computation, whereas the reverse is only efficient given some constraints on the structure of the measurement pattern. Causal flow and generalised flow have already been shown as sufficient, with efficient algorithms for identifying these properties and performing the circuit extraction. Pauli flow is a weaker set of conditions that extends generalised flow to use the knowledge that some vertices are measured in a Pauli basis. In this paper, we show that Pauli flow can similarly be identified efficiently and that any measurement pattern whose underlying graph admits a Pauli flow can be efficiently transformed into a gate-based circuit without using ancilla qubits. We then use this relationship to derive simulation results for the effects of graph-theoretic rewrites in the ZX-calculus using a more circuit-like data structure we call the Pauli Dependency DAG.

1 Introduction

There are numerous approaches to quantum computation that differ in how problems are encoded and how the system evolves over time. The one-way model [31] is a variant of Measurement-Based Quantum Computing (MBQC) where the inputs are embedded into a general graph state, and the chosen computation is specified by a sequence of single-qubit measurements to apply, consuming the state and inducing a desired output state on the remaining qubits. In contrast, the gate-based model [29] applies unitary gates from some universal gateset to the input qubits directly. Gate-based circuits may use ancilla qubits and measurements, though they are only necessary for expanding to a larger state space for outputs and reading out final data.

There is a wide literature on rewriting both measurement patterns [16, 17, 32] and circuits [2, 11, 18, 21, 25, 34] to reduce the resource cost. Of particular interest here is the use of graph-theoretic rewrites of ZX-diagrams (in close correspondence with measurement patterns [16]) by Kissinger and van de Wetering [21], and of representations focussed on sequences of rotations about Pauli tensors such as by Zhang and Chen [34] herein referred to as Pauli Dependency DAGs (PDDAGs) as they capture the temporal dependencies between these rotations. Both works presented these as techniques for T gate reduction with identical performance. The extraction method in this paper maps a measurement pattern/ZX-diagram directly into the form of a PDDAG, allowing us to formally demonstrate that each of the graph-theoretic rewrites can be simulated by moving Clifford-angled rotations through the PDDAG. This complements existing work on deriving equivalences between the PathSum formalism and the ZH calculus [3,4,23] and between different graphical calculi [22, 28] which have shown to provide useful insight on the semantics of operations for the more abstract calculi.

Previous work in this area has given algorithms for extracting circuits from measurement patterns that satisfy flow properties which describe how the errors from unwanted measurement outcomes can be corrected using the stabilizers of the graph state [5, 7]. The weakest such property that allows for interesting (i.e. non-Pauli) measurements is generalised flow (gflow) for which the single-plane form exactly describes the set of patterns that are uniformly, strongly, and stepwise deterministic [9]. Pauli flow [9] weakens the uniformity condition by labelling some of the measurements as being in a Pauli basis rather than at some arbitrary angle in a plane of the Bloch sphere. This allows for more interesting corrections that make use of Pauli projections to absorb some terms of the graph state stabilizers. This paper directly extends the works of Mhalla and Perdrix [27] and Backens et al. [5] for gflow to handle this wider class of measurement patterns.

This paper starts by laying out the key definitions and concepts behind measurement patterns and existing results on causal flow and gflow in Section 2. Section 3 will give a short overview of the Pauli Dependency DAG structure and how it can be used to rewrite circuits. We will then introduce the notion of Pauli flow and decompose the circuit extraction problem to generate a procedure that works for any measurement pattern with Pauli flow in Section 4. Then Section 5 will cover a number of methods of rewriting measurement patterns in turn and investigate their effects on the PDDAG extracted to show how they can be simulated.

2 Measurement-Based Quantum Computing

The one-way model follows the typical structure of MBQC of building some resource state which is then consumed by single-qubit measurements. The particular resource considered is a graph state, constructed by matching qubits (inputs and ancillas prepared in the $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state) with vertices in a graph, and entangling them (with a *CZ* gate) according to the edges. Measurements are generally non-deterministic so, in order to have a deterministic effect overall, local gates can be applied to the remaining qubits to counteract the difference between the projectors for each outcome, giving the net effect of post-selecting the desired outcome. The introduction of such correction gates can be viewed as adapting the choice of basis for the future measurements.

Single qubit measurements are characterised by the bases they project into, which themselves can be described by a vector in the Bloch sphere. For MBQC, we conventionally restrict measurements into a plane of the Bloch sphere spanned by two of the Pauli bases. Such planar measurement bases are described by the following for $\alpha \in [0, 2\pi)$:

$$
\begin{aligned}
|+_{XY,\alpha}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle + e^{i\alpha}|1\rangle\right) & |-_{XY,\alpha}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle - e^{i\alpha}|1\rangle\right) \\
|+_{XZ,\alpha}\rangle &= \cos\left(\tfrac{\alpha}{2}\right)|0\rangle + \sin\left(\tfrac{\alpha}{2}\right)|1\rangle & |-_{XZ,\alpha}\rangle &= \sin\left(\tfrac{\alpha}{2}\right)|0\rangle - \cos\left(\tfrac{\alpha}{2}\right)|1\rangle \\
|+_{YZ,\alpha}\rangle &= \cos\left(\tfrac{\alpha}{2}\right)|0\rangle + i\sin\left(\tfrac{\alpha}{2}\right)|1\rangle & |-_{YZ,\alpha}\rangle &= \sin\left(\tfrac{\alpha}{2}\right)|0\rangle - i\cos\left(\tfrac{\alpha}{2}\right)|1\rangle
\end{aligned}
\tag{1}
$$

with the following special cases for Pauli measurements for $\alpha = a\pi$ ($a \in \{0, 1\}$):

$$
\begin{aligned}
|+_{X,a\pi}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle + (-1)^a|1\rangle\right) & |-_{X,a\pi}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle - (-1)^a|1\rangle\right) \\
|+_{Y,a\pi}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle + i(-1)^a|1\rangle\right) & |-_{Y,a\pi}\rangle &= \tfrac{1}{\sqrt{2}}\left(|0\rangle - i(-1)^a|1\rangle\right) \\
|+_{Z,a\pi}\rangle &= (1-a)|0\rangle + a|1\rangle & |-_{Z,a\pi}\rangle &= a|0\rangle + (1-a)|1\rangle
\end{aligned}
\tag{2}
$$

For any measurement basis, the negative outcome at angle α is equivalent to the positive outcome at angle $\alpha + \pi$. We usually treat the positive measurement outcome as the desired branch, i.e. the projector we want to apply to achieve our desired end state.

Formally, measurement patterns (i.e. MBQC programs) are defined as follows:

Definition 2.1 (Measurement pattern)**.** A *measurement pattern* consists of a collection V of qubits with distinguished subsets $I, O \subseteq V$ of inputs and outputs, and a sequence of commands from:

- Preparations N_u, initialising qubit $u \notin I$ to $|+\rangle$;

- Entangling operators E_{uv}, applying a *CZ* gate between distinct qubits $u, v \in V$;

- Destructive measurements $M_u^{\lambda,\alpha}$, projecting qubit $u \notin O$ onto either $|+_{\lambda,\alpha}\rangle$ with outcome 0 or $|-_{\lambda,\alpha}\rangle$ with outcome 1;

- Corrections $[X_u]^v$ or $[Z_u]^v$, conditionally applying an X gate or a Z gate to qubit $u \in V$ if the outcome of the measurement for qubit v was 1.

A measurement pattern is *runnable* if additionally:

- All non-input qubits are prepared exactly once;

- A non-input qubit is not acted on by any other command before its preparation;

- All non-output qubits are measured exactly once;

- A non-output qubit is not acted on by any other command after its measurement;

- No correction depends on an outcome not yet measured.

The intended branch of a measurement pattern (where all measurement outcomes are 0 and hence no corrections are required) can be summarised by a tuple (Γ, α) of a labelled open graph Γ describing the entanglement and measurement planes, and an assignment of measurement angles $\alpha : \overline{O} \to [0, 2\pi)$.

Definition 2.2 (Labelled open graph)**.** A *labelled open graph* is a tuple $\Gamma = (G, I, O, \lambda)$ where $G = (V, E)$ is an undirected graph, $I, O \subseteq V$ are (possibly overlapping) subsets of vertices for inputs and outputs respectively, and $\lambda : \overline{O} \to \{XY, XZ, YZ, X, Y, Z\}$ is a labelling function assigning a measurement plane or Pauli to each non-output vertex.

Remark 2.3. To fix notation, we will use $\overline{I} = V \setminus I$ for non-input (prepared) vertices and $\overline{O} = V \setminus O$ for non-output (measured) vertices. $u \sim v$ denotes vertices $u, v \in V$ being adjacent in G. Neighbour sets are $N_G(u) := \{w \in V \,|\, w \sim u\}$ and odd neighbourhoods are $\mathrm{Odd}_G(A) := \{w \in V \,|\, |N_G(w) \cap A| \text{ is odd}\}$, writing $\mathrm{Odd}(A)$ when the choice of graph is obvious. The symmetric difference of sets is written as $A \Delta B := (A \setminus B) \cup (B \setminus A)$. When drawing diagrams for measurement patterns, we will distinguish between measured and output vertices as filled and unfilled points respectively, and inputs are specified by boxes around the vertices. For linear maps, subscripts may be used to specify that a map acts on some given qubit(s) and is the identity on all others, such as $\langle +_{\lambda(v), \alpha(v)}|_v$ or P_v for some Pauli P.

Each branch (combination of measurement outcomes) gives rise to a linear map from the inputs to the outputs based on the commands run and the projections observed. The overall channel from considering all branches is a completely-positive trace-preserving map whose Kraus maps are exactly the branch maps. When we have a strongly deterministic pattern (all branches are equal up to global phase), we just associate it with the linear map of the intended branch. Since runnable patterns can be standardised to perform all initialisations first, then all entangling gates, and finally alternate measurements and corrections, this linear map also has a standard representation.

Definition 2.4 (Linear map of a pattern)**.** The linear map associated with a measurement pattern (Γ, α) is given by

$$M_{\Gamma,\alpha} := \left(\prod_{u \in \overline{O}} \langle +_{\lambda(u),\alpha(u)}|_u \right) E_G N_{\overline{I}} \tag{3}$$

where $E_G := \prod_{u \sim v} E_{uv}$ entangles pairs of adjacent qubits in the graph with *CZ* gates and $N_{\bar{I}} := \prod_{u \in \bar{I}} N_u$ initialises every non-input in the $|+\rangle$ state.

Interpreting the graph state as $E_G N_{\bar{I}}$ gives rise to a stabilizer per initialised vertex $u \in \bar{I}$:

$$E_G N_{\bar{I}} = E_G X_u N_{\bar{I}} = \left(\prod_{v \in N_G(u)} Z_v \right) X_u E_G N_{\bar{I}} \tag{4}$$

Since the linear map of a pattern is not just a simple graph state but a projected graph state, it is possible for some Pauli terms introduced by graph state stabilizers to be absorbed by the Pauli projections:

$$\begin{aligned}
\langle +_{X,a\pi} |_u &= (-1)^a \langle +_{X,a\pi} |_u X_u \\
\langle +_{Y,a\pi} |_u &= (-1)^a \langle +_{Y,a\pi} |_u Y_u \\
\langle +_{Z,a\pi} |_u &= (-1)^a \langle +_{Z,a\pi} |_u Z_u
\end{aligned} \tag{5}$$

These stabilizers and Pauli absorptions are practical for deriving possible means of correcting for unwanted measurement outcomes.

The intention behind different types of flow conditions is to capture the ability to propagate errors from unwanted measurement outcomes forward to the rest of the circuit in order to correct them, aiming for stepwise determinism (each measurement can be corrected independently). Causal flow is the simplest case where we suppose all vertices are measured in the *XY* basis and each error can be corrected by considering a single stabilizer of the graph state.

Definition 2.5 (Causal flow [13]). Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ such that $\forall u \in \bar{O}.\lambda(u) = XY$, a *causal flow* for Γ is a tuple $(f, \prec)$ of a map $f : \bar{O} \to \bar{I}$ and a strict partial order $\prec$ over V such that for all $v \in \bar{O}$:

- $v \sim f(v)$
- $v \prec f(v)$
- $\forall w \in N_G(f(v)).w = v \lor v \prec w$

The idea here is that Z_v from the graph stabilizer $\left(\prod_{w \in N_G(f(v))} Z_w \right) X_{f(v)}$ will eliminate the effect of the measurement error on qubit v, meaning we can correct the error by applying $\left(\prod_{w \in N_G(f(v)) \setminus \{v\}} Z_w \right) X_{f(v)}$ and implicitly invoking the stabilizer. The partial order $\prec$ indicates a required order of the measurements, ensuring that none of the vertices required for correcting v have been measured yet.

Generalised flow takes a similar approach, but allows us to take combinations of the basic stabilizers. If the stabilizer of a candidate $f(v)$ would require a *Z* correction on some $u \in N_G(f(v))$ that has already been measured, there may exist some other stabilizer we can apply that cancels out the *Z* for us. Now, instead of the stabilizer being determined by a single vertex $f(v) \in \bar{I}$, we have a set $g(v) \subseteq \bar{I}$ giving $\left(\prod_{w \in \text{Odd}(g(v))} Z_w \right) \left(\prod_{w \in g(v)} X_w \right)$ up to phase. By relaxing these restrictions on the stabilizers used, we can also generate *Y* or *X* on the measured vertex, allowing the correction of measurements in the *XZ* and *YZ* planes respectively.

Definition 2.6 (Generalised flow [9]). Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ such that $\forall u \in \bar{O}.\lambda(u) \in \{XY, XZ, YZ\}$, a *generalised flow* (or *gflow*) for Γ is a tuple $(g, \prec)$ of a map $g : \bar{O} \to \mathscr{P}[\bar{I}]$ and a strict partial order $\prec$ over V such that for all $v \in \bar{O}$:

- $\forall w \in g(v).v \neq w \Rightarrow v \prec w$
- $\forall w \in \text{Odd}(g(v)).v \neq w \Rightarrow v \prec w$

Gate	Equivalent Pauli exponentials
CX_{ct}	$e^{-i\frac{\pi}{4}Z_cX_t}e^{i\frac{\pi}{4}Z_c}e^{i\frac{\pi}{4}X_t}$
CZ_{ct}	$e^{-i\frac{\pi}{4}Z_cZ_t}e^{i\frac{\pi}{4}Z_c}e^{i\frac{\pi}{4}Z_t}$
$RZ(\alpha)$	$e^{-i\frac{\alpha}{2}Z}$
$RX(\alpha)$	$e^{-i\frac{\alpha}{2}X}$
H	$e^{-i\frac{\pi}{4}Z}e^{-i\frac{\pi}{4}X}e^{-i\frac{\pi}{4}Z}$
CCX_{abt}	$e^{-i\frac{\pi}{8}Z_aZ_bX_t}e^{i\frac{\pi}{8}Z_aZ_b}e^{i\frac{\pi}{8}Z_aX_t}e^{i\frac{\pi}{8}Z_bX_t}e^{-i\frac{\pi}{8}Z_a}e^{-i\frac{\pi}{8}Z_b}e^{-i\frac{\pi}{8}X_t}$

Table 1: Summary of conversions from common gates into Pauli exponentials (up to global phase). Subscripts are used to indicate specific qubits for multi-qubit gates. Other gates can similarly be represented by first decomposing into a universal gate set such as $\{CX, RZ, RX\}$ and optionally tidying up using the Reorder Rules.

- $\lambda(v) = XY \Rightarrow v \notin g(v) \wedge v \in \mathrm{Odd}(g(v))$

- $\lambda(v) = XZ \Rightarrow v \in g(v) \wedge v \in \mathrm{Odd}(g(v))$

- $\lambda(v) = YZ \Rightarrow v \in g(v) \wedge v \notin \mathrm{Odd}(g(v))$

There exist polynomial-time algorithms for identifying whether or not a labelled open graph admits causal flow or gflow [5, 14, 27], and for extracting an equivalent unitary circuit for the measurement pattern given either type of flow [5, 7].

3   Pauli Dependency DAGs

The Pauli Dependency DAG is a data structure for representing the action of a pure quantum circuit that abstracts away gate set, Clifford gates, and gate commutations. This was notably covered in the work of Zhang and Chen [34] where it was used to identify possible pairs of T gates which could be merged through some sequence of gate commutations and Clifford gate relations in order to reduce the number of T gates in the circuit. Similar structures are also covered by Litinski [24] for compiling circuits for lattice surgery and by Gosset et al. [19] as an intermediate for synthesis of Clifford+T circuits.

The Clifford group is the group of linear maps that can be formed from CX, Hadamard, and $RZ(\frac{\pi}{2})$ with qubit initialisation in the $|0\rangle$ state. These hold a special place in quantum theory as the maximal group of actions under which the Pauli group is closed. Their behaviour is conveniently captured by the stabilizer framework, allowing for canonical representations like the stabilizer tableau [1, 33]. They are often viewed as the "easy" gates in a circuit because of their simple algebra, efficient simulation [1, 20], and low resource cost in many error correction schemes [10].

Most common gate types can be expressed as (combinations of) Pauli exponentials ($e^{i\theta P}$ for some $P \in \{I, X, Y, Z\}^{\otimes n}$), which also benefit from elegant relations with stabilizers.

Lemma 3.1 (Product Rotation Lemma). *Let A and B be commuting operators such that $B\mathscr{C} = \mathscr{C}$ for some linear map $\mathscr{C}$. Then $e^{i\theta A}\mathscr{C} = e^{i\theta AB}\mathscr{C}$.*

Proof. For any analytic function $F(A)$, we can expand its Taylor series in $F(A)\mathscr{C}$, then introduce and commute B in each term to form the Taylor series for $F(AB)\mathscr{C}$. $\square$

Pauli exponentials where the coefficient is an integral multiple of $\frac{\pi}{4}$ correspond to Clifford gates. The action of Clifford gates on both Paulis and arbitrary Pauli exponentials can be summarised in a few equations.

Lemma 3.2 (Reorder Rules)**.** For any Pauli strings $A, B \in \{I, X, Y, Z\}^{\otimes n}$ and angles θ, ϕ, if A and B commute, then

$$e^{i\theta A} B = B e^{i\theta A} \tag{6}$$

$$e^{i\theta A} e^{i\phi B} = e^{i\phi B} e^{i\theta A} \tag{7}$$

and otherwise (i.e. they anticommute)

$$e^{i\frac{\pi}{4}A} B = (iAB) e^{i\frac{\pi}{4}A} \tag{8}$$

$$e^{i\frac{\pi}{4}A} e^{i\phi B} = e^{i\phi(iAB)} e^{i\frac{\pi}{4}A} \tag{9}$$

Proof. Any operator satisfying $A^2 = I$ and real θ permit the decomposition $e^{i\theta A} = \cos\theta I + i\sin\theta A$ by grouping terms in the Taylor expansion. Each equation follows from decomposing one of the exponentials in this way. Whilst the right-hand side of Equation 9 appears to contain a real exponent, remember that iAB is a real Pauli string when A and B anticommute. $\qquad\square$

We can represent any circuit as a set of qubit initialisations followed by a sequence of Pauli exponentials by decomposing each gate in turn. The above rules can then be used to move any Clifford-angled exponentials to the start of the circuit, resulting in the product form $\left(\prod_k e^{i\phi_k A_k}\right) \mathscr{C}$ where $\mathscr{C}$ is a stabilizer process and each ϕ_k is not an integral multiple of $\frac{\pi}{4}$. Previous presentations [24, 34] collected the Clifford gates at the end of the circuit rather than at the start. When considering unitary circuits, the choice of direction is arbitrary, but in the general case we may not always be able to transport qubit initialisations through the Pauli exponentials to the end.

$\mathscr{C}$ can be expressed canonically by its stabilizer tableau. Specifically, since we could have an arbitrary number of inputs we need a variant mixing the extremes of the unitary case [1] and the usual state case [33] to capture both how Paulis over the inputs are transformed to Pauli strings over the outputs and additional generators for free stabilizers over the outputs. In examples, we will describe this by the stabilizers of the Choi operator of $\mathscr{C}$ (which we shall refer to as an *isometry tableau* for clarity), from which it is simple to convert it to any binary matrix format of choice. Such tableaux can be identified and synthesised by embedding them into unitary tableaux by breaking down the isometry into an initialisation of $|O| - |I|$ qubits in the $|0\rangle$ state followed by a unitary circuit which maps Z on the initialised qubits to each of the free output stabilizers and X to any choice of operators that extend our generators to span the full Pauli group. Since the rows are just generators for the stabilizer group, we still have the notion of "free actions" given by reordering rows or multiplying two rows together.

For the rotation list, there is still some obvious redundancy in the product form resulting from commutations. Because anticommuting Pauli strings prevent commutation of their exponentials, any valid ordering of the rotations will preserve the relative order of rotations with anticommuting strings, inducing a temporal dependency between them. Taking the transitive closure of these dependencies gives a partial order between the exponentials which represents $\prod_k e^{i\phi_k A_k}$ up to any number of commutations.

Definition 3.3 (Pauli Dependency DAG)**.** A *Pauli Dependency DAG* (PDDAG) for a circuit is a pair of an isometry tableau for a Clifford map $\mathscr{C}$ and a directed acyclic graph for a partial order $\prec$ over
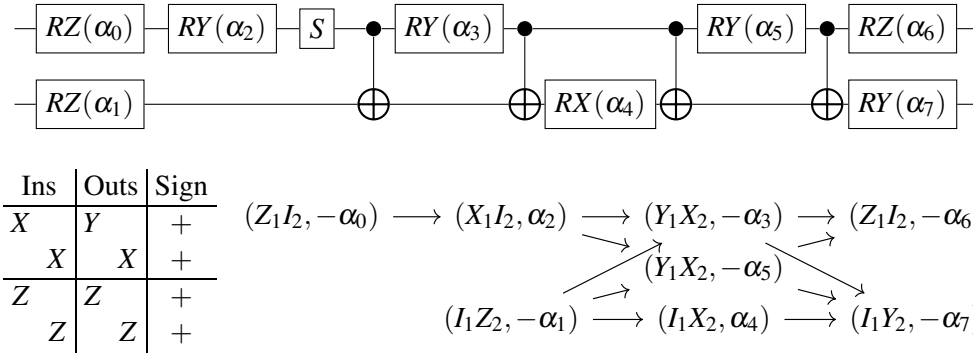
Figure 1: An example quantum circuit and the corresponding isometry tableau and rotation DAG which form the PDDAG. Since the nodes of angles α_3 and α_5 have no relative dependencies and share the same Pauli string, it is possible to commute them to be adjacent and merge them to a single rotation. The α_4 rotation can commute through the CX gates so is only blocked by α_1 in the past and α_7 in the future.

terms $\{(A_k, \alpha_k)\}_k$ ($A_k \in \{I, X, Y, Z\}^n$, $\alpha_k \in [0, 2\pi)$) such that the linear map of the circuit is given by $\left(\prod_k^\succ e^{i\frac{\alpha_k}{2}A_k} \right) \mathscr{C}^1$ up to global phase.

Since the full $\prec$ relation is rather large, we use the minimal representation/Hasse diagram which includes edges in the DAG for immediate successors (those pairs of exponentials whose strings anticommute and could appear adjacent to one another in some valid topological ordering).

Rewrite rules for PDDAGs change the structure but preserve the linear map. The most useful rewrite for circuit reduction is rotation merging: two nodes in the graph j, k can be merged to give a single node with $(A_j, \alpha_j + \alpha_k)$ if $A_j = A_k$ and both $j \not\prec k$ and $k \not\prec j$ (i.e. there is some valid topological ordering of the graph which puts these rotations together).

The PDDAG definition permits Clifford-angled rotations, allowing the movement of Clifford rotations through the circuit via the Reorder Rules to be an explicit rewrite. Combining this with merging/splitting of phases and eliminating rotations with angle 0 permits a more rigid canonical form where the range of angles for each rotation is reduced from $[0, 2\pi)$ to $\left(0, \frac{\pi}{2}\right)$.

When the initial Clifford process includes some qubit initialisations, it will have some free stabilizers over its outputs that can be used to change the Pauli strings of rotations in the DAG using the Product Rotation Lemma. We can use the Reorder Rules to propagate the stabilizers beyond the initial rotations to find more places to perform this rewrite. Changing the Pauli strings of rotations in this way can affect how they commute/anticommute with their neighbours, modifying the dependency relation $\prec$.

Extracting a circuit back out of an arbitrary PDDAG can be naively done by synthesising the stabilizer tableau [1, 26] and each of the Pauli exponentials in turn according to some topological ordering using some standard decompositions [6, 11] although this will typically add an extremely high amount of redundant Clifford gates. More efficient synthesis can be performed using techniques for synthesising pairs of rotations simultaneously [11] or by diagonalising sets of mutually commuting rotations [8, 12]. It is also possible that future architectures may find efficient ways to perform each Pauli rotation natively or employ lattice surgery where it is practical to just perform them directly [24].

---

[1]The convention of using $e^{i\frac{\alpha_k}{2}A_k}$ rather than $e^{i\alpha_k A_k}$ is to give a closer semblance to the definitions of usual rotation gates like RZ and their multi-qubit counterparts from literature on Pauli gadgets [11, 12] and to make the notation easier when we come to discuss extraction but is just a notational choice.

Alternatively, similar to the action of phase teleportation in ZX-diagram rewriting [21], one can also retain the structure of a circuit and just use the PDDAG structure to spot where non-adjacent gates can be merged via the rotation merging rewrites. This is typically good when the original circuit already has a relatively low density of Clifford gates when it is unlikely that resynthesis will give as efficient a circuit structure.

4   Circuit Extraction

The goal of circuit extraction is to identify a sequence of gates that implements the same linear map as a given measurement pattern. Simply calculating the linear map and using standard matrix decomposition techniques is not sufficient since it will scale exponentially with the size of the pattern, making it impractical in the long-run. The method presented here will make use of a Pauli flow to determine the effect that each measurement angle has on the outputs. Doing so will yield the PDDAG representation directly, giving a rotation per planar measurement and a stabilizer process.

To motivate our method, recall the possible measurement bases from Section 2. Each planar basis projection can be constructed as a basic rotation of some Pauli basis:

$$
\begin{aligned}
\langle \pm_{XY,\alpha}| &\approx \langle \pm_{X,0}| e^{i\frac{\alpha}{2}Z} \\
\langle \pm_{XZ,\alpha}| &\approx \langle \pm_{Z,0}| e^{i\frac{\alpha}{2}Y} \\
\langle \pm_{YZ,\alpha}| &\approx \langle \pm_{Z,0}| e^{-i\frac{\alpha}{2}X}
\end{aligned}
\tag{10}
$$

The key idea driving this method of circuit extraction is to apply the Product Rotation Lemma to alter these rotations and move them to onto the output qubits, but this requires identifying an appropriate stabilizer of the graph state to use. In a similar way, flow conditions already describe stabilizers for propagating errors onto other qubits. Pauli flow is one such example which can handle both planar measurements $\lambda(v) \in \{XY, XZ, YZ\}$ and special cases for Pauli measurements $\lambda(v) \in \{X, Y, Z\}$. Similar to gflow, it gives a stabilizer that applies a Pauli X to vertices in $p(v)$ and a Pauli Z to vertices in $\mathrm{Odd}(p(v))$. However, because Equation 5 means Pauli corrections have no observable effect if the qubit is measured in the same basis, it doesn't matter if, for example, a vertex $u \in p(v)$ with $\lambda(u) = X$ has already been measured before v.

$$
\langle +_{X,\alpha(u)}|_u E_G N_{\overline{I}} \approx \langle +_{X,\alpha(u)}|_u \left(\prod_{\substack{w \in p(v) \\ w \neq u}} X_w \right) \left(\prod_{w \in \mathrm{Odd}(p(v))} Z_w \right) E_G N_{\overline{I}}
\tag{11}
$$

Definition 4.1 (Pauli flow [9, 30])**.** Given a labelled open graph $\Gamma = (G, I, O, \lambda)$, a *Pauli flow* for Γ is a tuple $(p, \prec)$ of a map $p : \overline{O} \to \mathscr{P}[\overline{I}]$ and a strict partial order $\prec$ over V such that for all $u \in \overline{O}$:

$[\prec.X]$ $\forall v \in p(u).u \neq v \wedge \lambda(v) \notin \{X, Y\} \Rightarrow u \prec v$

$[\prec.Z]$ $\forall v \in \mathrm{Odd}(p(u)).u \neq v \wedge \lambda(v) \notin \{Y, Z\} \Rightarrow u \prec v$

$[\prec.Y]$ $\forall v \preceq u.u \neq v \wedge \lambda(v) = Y \Rightarrow (v \in p(u) \Leftrightarrow v \in \mathrm{Odd}(p(u)))$

$[\lambda.XY]$ $\lambda(u) = (X, Y) \Rightarrow u \notin p(u) \wedge u \in \mathrm{Odd}(p(u))$

$[\lambda.XZ]$ $\lambda(u) = (X, Z) \Rightarrow u \in p(u) \wedge u \in \mathrm{Odd}(p(u))$

$[\lambda.YZ]$ $\lambda(u) = (Y, Z) \Rightarrow u \in p(u) \wedge u \notin \mathrm{Odd}(p(u))$

[$\lambda.X$] $\lambda(u) = X \Rightarrow u \in \mathrm{Odd}(p(u))$

[$\lambda.Z$] $\lambda(u) = Z \Rightarrow u \in p(u)$

[$\lambda.Y$] $\lambda(u) = Y \Rightarrow (u \notin p(u) \wedge u \in \mathrm{Odd}(p(u))) \vee (u \in p(u) \wedge u \notin \mathrm{Odd}(p(u)))$

where $u \preceq v := \neg(v \prec u)$.

Definition 4.2 (Extraction string). Let (Γ, α) describe a measurement pattern with Pauli flow $(p, \prec)$ and some chosen measured vertex $v \in \overline{O}$. A Pauli string $\mathbf{P}$ over the outputs is a *P-extraction string* $(P \in \{X,Y,Z\})$ for v if $P_v\mathbf{P}$ is a stabilizer of the linear map $\mathscr{C}$:

$$\mathscr{C} := \left(\prod_{\substack{w \in \overline{O} \\ w \succ v \\ \lambda(w) \in \{XY,XZ,YZ\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right) E_G N_{\overline{I}} \tag{12}$$

A *primary extraction string* $\mathbf{P}^{\perp v}$ for v is a $P^{\perp v}$-extraction string where

$$P^{\perp v} := \begin{cases} X & \text{if } v \in p(v) \wedge v \notin \mathrm{Odd}(p(v)) \\ Y & \text{if } v \in p(v) \wedge v \in \mathrm{Odd}(p(v)) \\ Z & \text{if } v \notin p(v) \wedge v \in \mathrm{Odd}(p(v)) \end{cases} \tag{13}$$

Relating this definition to the goal of using the Product Rotation Lemma, the cases for $P^{\perp v}$ match the rotations in Equation 10. The fact that the rest of the stabilizer is over the outputs means we are successfully removing it from the scope of the pattern. For the purposes of extraction, we can assume all measurement angles of future vertices have already been extracted, making them Pauli projections. The notion of a focussed flow [5, 27] then guarantees that all Pauli terms are absorbed, leaving something in the form of an extraction string.

Definition 4.3 (Focussed). Given a labelled open graph Γ, a set $\hat{p} \subset \overline{I}$ is *focussed over* $S \subseteq \overline{O}$ if:

[*FX*] $\forall w \in S \cap \hat{p}. \lambda(w) \in \{XY,X,Y\}$

[*FZ*] $\forall w \in S \cap \mathrm{Odd}(\hat{p}).\lambda(w) \in \{XZ,YZ,Y,Z\}$

[*FY*] $\forall w \in S.\lambda(w) = Y \Rightarrow (w \in \hat{p} \Leftrightarrow w \in \mathrm{Odd}(\hat{p}))$

A *focussed set* $\hat{p}$ for Γ is focussed over $\overline{O}$. A Pauli flow $(p, \prec)$ is *focussed* if $p(v)$ is focussed over $\overline{O} \setminus \{v\}$ for all $v \in \overline{O}$.

Lemma 4.4. Let $\Gamma = (G,I,O,\lambda)$ be a labelled open graph with some measurement angles $\alpha : \overline{O} \to [0,2\pi)$ and a focussed Pauli flow $(p, \prec)$. Then for any vertex $v \in \overline{O}$, $p(v)$ determines a $P^{\perp v}$-extraction string.

Proof. Proof in Appendix (Lemma C.2). $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ □

To make this extraction technique practical, we need to be able to find a focussed Pauli flow when one exists. This follows similarly to the corresponding results for causal flow and gflow.

Theorem 4.5. (Generalisation of [27, Theorem 2], [5, Theorem 3.11]) There exists an algorithm that decides whether a given labelled open graph has a Pauli flow, and that outputs such a Pauli flow if it exists. Moreover, this output is maximally delayed, and the algorithm completes deterministically in time that grows polynomially with the number of vertices in the graph.

Proof. Proof in Appendix (Theorem A.6). □

Lemma 4.6. (Generalisation of [5, Proposition 3.14]) If a labelled open graph has a Pauli flow, then it has a maximally delayed, focussed Pauli flow.

Proof. Proof in Appendix (Lemma B.5). □

We are now in the position where we can take any Pauli flow and extract all of the planar measurement angles from a measurement pattern. This just leaves a Clifford process which can be characterised completely by stabilizer theory. The rows of the isometry tableau describing how inputs are mapped are given by extraction strings of the inputs (taking input extensions of the measurement pattern as appropriate), and the remaining generators over the outputs are obtained from extraction strings.

Theorem 4.7. (Generalisation of [5, Theorem 5.5]) Let (Γ, α) describe a measurement pattern where Γ has a Pauli flow. Then there is an algorithm that identifies an equivalent circuit requiring no ancillae which completes in time polynomial in the number of vertices in Γ.

Proof. Proof in Appendix (Theorem C.12). □

5 Relating Rewrites

The most common rewrites in PDDAGs are merging terms and moving Clifford rotations, most notably moving a Clifford phase between a node in the DAG and the initial Clifford process. The structure or redundancy being exploiting for optimisation is very clear and easy to understand. On the other hand, graph-theoretic rewrites used for measurement patterns or ZX-diagrams have less obvious interpretations in how they relate to optimisations in the gate-based model. To compare the two, we consider extracting a PDDAG from a measurement pattern before and after a rewrite to observe the changes in the order, Pauli strings, and phases of the rotations from each measurement or tableau row, and then find a sequence of simple PDDAG rewrites that produces the same effect. Detailed proofs and examples can be found in Appendix D.

Given the special treatment of Pauli measurements in our representation of measurement patterns, the simplest rewrite we can do is to relabel a vertex measured in a Pauli basis between a planar label and a Pauli label, without changing the graph or other vertices.

Theorem 5.1. Let (Γ, α) describe a measurement pattern with some vertex $u \in \overline{O}$ such that $\lambda(u) \in \{XY, XZ, YZ\}$ and $\alpha(u) \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Relabelling u to the equivalent Pauli label corresponds to pushing the rotation from u to the start of the PDDAG and absorbing it into the initial stabilizer process.

Proof. Proof in Appendix (Theorem D.4). □

Removing vertices from the measurement pattern typically involves reducing a vertex's measurement to the Z basis, at which point it no longer needs to be entangled with the other qubits. We can consider doing this both when the vertex is labelled as a Pauli Z measurement and as a planar measurement in the XZ or YZ planes.

Theorem 5.2. Let (Γ, α) describe a measurement pattern with some vertex $u \in \overline{O}$ such that $\lambda(u) \in \{XZ, YZ, Z\}$ and $\alpha(u) \in \{0, \pi\}$. Eliminating u from the graph corresponds to the following sequence of actions on the PDDAG:

1. If u has a planar (XZ or YZ) label, then its rotation is pulled from the rotation DAG into the stabilizer block;

2. For each neighbour n of u that is an output, a Z_n rotation of $\alpha(u)$ is pulled from the stabilizer block through the entire rotation DAG to the end of the circuit;

3. For each neighbour n of u with $\lambda(n) = XY$, a $\mathbf{P}^{\perp n}$ rotation of $\alpha(u)$ is pulled from the stabilizer block and merged with the existing rotation for n.

Proof. Proof in Appendix (Theorem D.9). $\square$

Here, we have a slightly different effect for planar labels compared to the Pauli Z label, though this can be thought of as a combination of the previous rewriting rule and the basic case for Pauli Z elimination. The additional rotations generated are exactly the Z gates introduced on neighbouring qubits to preserve the semantics (Lemma D.6).

One of the more interesting rewrites on measurement patterns is performing local complementation about some vertex u. This inverts the connectivity between each pair of vertices neighbouring u. To preserve the semantics, we also update the labels and measurement angles of u and its neighbours. Combining this with Z vertex elimination allows the elimination of vertices measured in the Y basis.

Theorem 5.3. Performing local complementation about a vertex u corresponds to the following sequence of actions on the PDDAG:

1. For each output w neighbouring u, a $(Z_w, \frac{\pi}{2})$ rotation is pulled from the initial stabilizer block all the way to the end of the rotation DAG;

2. If u is an output, a $(X_u, -\frac{\pi}{2})$ rotation is pulled from the initial stabilizer block all the way to the end of the rotation DAG;

3. For each vertex w neighbouring u with $\lambda(w) = XY$, and for u itself if $\lambda(u)$ is planar, we pull a $\left(\mathbf{P}'^{\perp w}, (-1)^{D_w}\frac{\pi}{2}\right)$ rotation from the initial stabilizer block to merge it into the existing rotation for w or u. We do this in $\succ$-order over such w vertices.

Proof. Proof in Appendix (Theorem D.17). $\square$

Another similar operation to local complementation is the act of pivoting a diagram about an edge, which can be used to prepare a Pauli X measurement for elimination. This action can be decomposed into a sequence of local complementations about each end of the edge, so this can also be simulated by some sequence of Clifford transformations in the PDDAG.

In each of the above, we assume that a particular transformation is made to the focussed Pauli flow and focussed sets for the measurement pattern based on the rewrite chosen, as detailed in Appendix D. In truth, focussed Pauli flows need not be unique for a measurement pattern. We can view the map between focussed Pauli flows as a rewrite on the measurement pattern that changes the corrections applied after each measurement, keeping the labelled open graph and measurement angles the same. We can compare the differences to the focussed sets for the pattern to obtain our final correspondence theorem:

Theorem 5.4. Let (Γ, α) describe a measurement pattern with some focussed Pauli flow $(p, \prec)$, a focussed set $\hat{p}$, and some vertex $u \in \overline{O}$ such that $\forall w \in \hat{p} \cup \text{Odd}(\hat{p}).\lambda(w) \in \{XY, XZ, YZ\} \Rightarrow w \neq u \wedge u \prec w$. Updating $p(u)$ to $p(u)\Delta\hat{p}$ corresponds to a free action on the isometry tableau if u is an input, and applying the Product Rotation Lemma to the rotation from u with the stabilizer of $\hat{p}$ if u has a planar label.

Therefore, any two focussed Pauli flows for the same labelled open graph yield PDDAGs that are related by a sequence of applications of the Product Rotation Lemma and free actions on the isometry tableau.

Proof. Proof in Appendix (Theorem D.23). □

An interesting consequence of this result is the uniqueness of focussed Pauli flow for unitary patterns (up to weakening of the partial order), since there are no free stabilizers with which to apply the Product Rotation Lemma.

6   Conclusion

We have investigated the significance of Pauli flow in the extraction of gate-based reversible circuits from measurement patterns, demonstrating that it is sufficient. This weakens the requirements for circuit extraction from universal measurement patterns compared to previous results. The principal contributions are a polynomial-time algorithm for identifying a Pauli flow for a measurement pattern (if one exists) and using this to construct an equivalent circuit without using ancillas or measurements.

The subsequent investigation using this map formally demonstrated that the effects of common graph-theoretic rewrites on measurement patterns/ZX-diagrams can be simulated via movement of Clifford rotations in a Pauli Dependency DAG. We know that the reverse simulation must be possible from the completeness of the ZX-calculus.

There are a number of possible avenues for future development regarding the topics of this paper:

- The simulation of rewrites demonstrates that ZX-calculus and PDDAGs can perform equivalent effects, but analysis of the computational complexities is needed to determine preferences for practical usage such as the actual data structure of choice for quantum circuit compilation. This will be heavily dependent on the choice of the graph structure used in each case and how this can capture the dependency relation in the PDDAG.

- Given a PDDAG (or generic gate-based circuit), the work here provides some practical constraints on Pauli flows that any equivalent measurement pattern should admit. This could give rise to a reverse construction to find a minimal measurement pattern for a given PDDAG, allowing for explicit simulations for any PDDAG rewrite in measurement patterns.

- The flexibility of the PDDAG structure could be extended by incorporating measurements, discards or decoherence, conditional gates, and more elaborate rewrites aiming for completeness for deciding circuit equivalence.

References

[1] Scott Aaronson & Daniel Gottesman (2004): *Improved Simulation of Stabilizer Circuits*. *Physical Review A* 70(5), p. 052328, doi:10.1103/PhysRevA.70.052328.

[2] Matthew Amy (2019): *Formal Methods in Quantum Circuit Design*. Ph.D. thesis. Available at http://hdl.handle.net/10012/14480.

[3] Matthew Amy (2019): *Towards Large-scale Functional Verification of Universal Quantum Circuits*. *Electronic Proceedings in Theoretical Computer Science* 287, pp. 1–21, doi:10.4204/EPTCS.287.1.

[4] Miriam Backens & Aleks Kissinger (2019): *ZH: A Complete Graphical Calculus for Quantum Computations Involving Classical Non-linearity*. *Electronic Proceedings in Theoretical Computer Science* 287, pp. 23–42, doi:10.4204/EPTCS.287.2.

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

[6] Panagiotis Kl Barkoutsos, Jerome F. Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J. Egger, Matthias Troyer, Antonio Mezzacapo, Stefan Filipp & Ivano Tavernelli (2018): *Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions*. *Physical Review A*, doi:10.1103/PhysRevA.98.022322.

[7] Niel De Beaudrap (2010): *Unitary-circuit semantics for measurement-based computations*. *International Journal of Quantum Information*, doi:10.1142/S0219749910006113.

[8] Ewout van den Berg & Kristan Temme (2020): *Circuit optimization of Hamiltonian simulation by simultaneous diagonalization of Pauli clusters*. *Quantum* 4, p. 322, doi:10.22331/q-2020-09-12-322.

[9] Daniel E Browne, Elham Kashefi, Mehdi Mhalla & Simon Perdrix (2007): *Generalized flow and determinism in measurement-based quantum computation*. *New Journal of Physics* 9(8), pp. 250–250, doi:10.1088/1367-2630/9/8/250.

[10] A. R. Calderbank, E. M. Rains, P. W. Shor & N. J.A. Sloane (1997): *Quantum error correction and orthogonal geometry*. *Physical Review Letters*, doi:10.1103/PhysRevLett.78.405.

[11] Alexander Cowtan, Silas Dilkes, Ross Duncan, Will Simmons & Seyon Sivarajah (2020): *Phase gadget synthesis for shallow circuits*. In: *Electronic Proceedings in Theoretical Computer Science, EPTCS*, doi:10.4204/EPTCS.318.13.

[12] Alexander Cowtan, Will Simmons & Ross Duncan (2020): *A Generic Compilation Strategy for the Unitary Coupled Cluster Ansatz*. Available at http://arxiv.org/abs/2007.10515.

[13] Vincent Danos & Elham Kashefi (2006): *Determinism in the one-way model*. *Physical Review A* 74(5), doi:10.1103/PhysRevA.74.052310.

[14] Niel De Beaudrap (2008): *Finding flows in the one-way measurement model*. *Physical Review A - Atomic, Molecular, and Optical Physics* 77(2), p. 022328, doi:10.1103/PhysRevA.77.022328.

[15] Ross Duncan, Aleks Kissinger, Simon Perdrix & John Van De Wetering (2020): *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus*. *Quantum* 4, p. 279, doi:10.22331/q-2020-06-04-279.

[16] Ross Duncan & Simon Perdrix (2010): *Rewriting Measurement-Based Quantum Computations with Generalised Flow*. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 285–296, doi:10.1007/978-3-642-14162-1_24.

[17] Maryam Eslamy, Mahboobeh Houshmand, Morteza Saheb Zamani & Mehdi Sedighi (2018): *Optimization of One-Way Quantum Computation Measurement Patterns*. *International Journal of Theoretical Physics*, doi:10.1007/s10773-018-3844-x.

[18] Andrew Fagan & Ross Duncan (2019): *Optimising Clifford Circuits with Quantomatic*. *Electronic Proceedings in Theoretical Computer Science* 287, pp. 85–105, doi:10.4204/EPTCS.287.5.

[19] David Gosset, Vadym Kliuchnikov, Michele Mosca & Vincent Russo (2014): *An algorithm for the T-count*. *Quantum Information and Computation* 14(15&16), pp. 1261–1276, doi:10.26421/QIC14.15-16-1.

[20] Daniel Gottesman (1998): *The Heisenberg Representation of Quantum Computers*. Available at `http://arxiv.org/abs/quant-ph/9807006`.

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

[22] Stach Kuijpers, John van de Wetering & Aleks Kissinger (2019): *Graphical Fourier Theory and the Cost of Quantum Addition*. Available at `http://arxiv.org/abs/1904.07551`.

[23] Louis Lemonnier, John van de Wetering & Aleks Kissinger (2020): *Hypergraph simplification: Linking the path-sum approach to the ZH-calculus*. Available at `http://arxiv.org/abs/2003.13564`.

[24] Daniel Litinski (2019): *A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery*. Technical Report, doi:`10.22331/q-2019-03-05-128`.

[25] Dmitri Maslov (2017): *Basic circuit compilation techniques for an ion-trap quantum machine*. Technical Report, doi:`10.1088/1367-2630/aa5e47`.

[26] Dmitri Maslov & Martin Roetteler: *Shorter stabilizer circuits via Bruhat decomposition and quantum circuit transformations*. Technical Report, doi:`10.1109/TIT.2018.2825602`.

[27] Mehdi Mhalla & Simon Perdrix (2008): *Finding Optimal Flows Efficiently*. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 5125 LNCS, Springer, Berlin, Heidelberg, pp. 857–868, doi:`10.1007/978-3-540-70575-8_70`.

[28] Kang Feng Ng & Quanlong Wang (2017): *A universal completion of the ZX-calculus*. Available at `http://arxiv.org/abs/1706.09877`.

[29] Michael A. Nielsen & Isaac L. Chuang (2010): *Quantum Computation and Quantum Information*. doi:`10.1017/cbo9780511976667`.

[30] Simon Perdrix & Luc Sanselme (2017): *Determinism and Computational Power of Real Measurement-Based Quantum Computation*. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10472 LNCS, Springer Verlag, pp. 395–408, doi:`10.1007/978-3-662-55751-8_31`.

[31] Robert Raussendorf & Hans J. Briegel (2001): *A one-way quantum computer*. Physical Review Letters, doi:`10.1103/PhysRevLett.86.5188`.

[32] Raphael Dias da Silva, Einar Pius & Elham Kashefi (2013): *Global Quantum Circuit Optimization*. Available at `http://arxiv.org/abs/1301.0351`.

[33] Maarten Van den Nest, Jeroen Dehaene & Bart De Moor (2004): *Graphical description of the action of local Clifford transformations on graph states*. Physical Review A - Atomic, Molecular, and Optical Physics, doi:`10.1103/PhysRevA.69.022316`.

[34] Fang Zhang & Jianxin Chen (2019): *Optimizing T gates in Clifford+T circuit as $\pi/4$ rotations around Paulis*. Available at `http://arxiv.org/abs/1903.12456`.

A   An Algorithm for Identifying Pauli Flow

Here we will build up to an algorithm for identifying a Pauli flow from a graph, which follows the key principle of previous algorithms for causal flow and gflow [5, 27] of delaying measurements as long as possible in order to have as many vertices available for corrections as possible at each step. This is achieved by working backwards from the outputs and calculating the sets of vertices of a given "correction depth".

Definition A.1. For a given labelled open graph (G, I, O, λ) and a given Pauli flow $(p, \prec)$, for each $k \geq -1$ we define the vertices at depth k under $\prec$ by

$$V_k^{\prec} := \begin{cases} \emptyset & \text{if } k = -1 \\ \max_{\prec}(V \setminus \bigcup_{i < k} V_i^{\prec}) & \text{if } k \geq 0 \end{cases} \tag{14}$$

the cumulative vertices up to depth k by

$$V_{\cup k}^{\prec} := \bigcup_{i \leq k} V_i^{\prec} \tag{15}$$

and the vertices in each plane correctable at depth $k+1$ by

$$V_{k+1}^{\prec, XY} := \left\{ u \in \overline{O} \,\middle|\, \begin{array}{l} \lambda(u) \in \{XY, X, Y\} \wedge \exists K \subseteq (V_{\cup k}^{\prec} \cup \Lambda_u^X \cup \Lambda_u^Y) \cap \overline{I}. \\ K \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} = \mathrm{Odd}(K) \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} \\ \wedge \mathrm{Odd}(K) \setminus (V_{\cup k}^{\prec} \cup \Lambda_u^Y \cup \Lambda_u^Z) = \{u\} \end{array} \right\} \tag{16}$$

$$V_{k+1}^{\prec, XZ} := \left\{ u \in \overline{O} \,\middle|\, \begin{array}{l} \lambda(u) \in \{XZ, X, Z\} \wedge \exists K \subseteq (V_{\cup k}^{\prec} \cup \Lambda_u^X \cup \Lambda_u^Y) \cap \overline{I}. \\ K \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} = \mathrm{Odd}(K \cup \{u\}) \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} \\ \wedge \mathrm{Odd}(K \cup \{u\}) \setminus (V_{\cup k}^{\prec} \cup \Lambda_u^Y \cup \Lambda_u^Z) = \{u\} \end{array} \right\} \tag{17}$$

$$V_{k+1}^{\prec, YZ} := \left\{ u \in \overline{O} \,\middle|\, \begin{array}{l} \lambda(u) \in \{YZ, Y, Z\} \wedge \exists K \subseteq (V_{\cup k}^{\prec} \cup \Lambda_u^X \cup \Lambda_u^Y) \cap \overline{I}. \\ K \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} = \mathrm{Odd}(K \cup \{u\}) \cap \Lambda_u^Y \setminus V_{\cup k}^{\prec} \\ \wedge \mathrm{Odd}(K \cup \{u\}) \setminus (V_{\cup k}^{\prec} \cup \Lambda_u^Y \cup \Lambda_u^Z) = \emptyset \end{array} \right\} \tag{18}$$

where $\Lambda_u^P := \{v \in \overline{O} \,|\, v \neq u \wedge \lambda(v) = P\}$.

The sets $V_k^{\prec, XY}, V_k^{\prec, XZ}, V_k^{\prec, YZ}$ capture the vertices v at correction depth k for which the v component of the correcting stabilizer is Z, Y, or X respectively. These definitions are intended to mirror the Pauli flow conditions, with the restrictions on $\lambda(v)$ matching [$\lambda.XY$]-[$\lambda.Y$] and the other conditions capturing [$\prec.X$]-[$\prec.Y$] using K for $p(v) \setminus \{v\}$ and $V_{\cup k}^{\prec}$ to represent the future vertices under $\prec$.

Definition A.2 (Delayed)**.** Given two Pauli flows $(p, \prec)$ and $(p', \prec')$ for the same labelled open graph, $(p, \prec)$ is *more delayed* than $(p', \prec')$ if for all k,

$$\left| V_{\cup k}^{\prec} \right| \geq \left| V_{\cup k}^{\prec'} \right| \tag{19}$$

and there exists a k for which this inequality is strict. A Pauli flow $(p, \prec)$ is *maximally delayed* if there exists no Pauli flow over the same labelled open graph that is more delayed.

Remark A.3. If a labelled open graph admits a Pauli flow, then there must be a maximally delayed Pauli flow since the delayed relation is a strict partial order and there are only finitely many possible Pauli flows (the graph is finite, so there are only finite possible choices of maps p and orders $\prec$). Any sequence of increasingly delayed flows must be finite, so we can always reach a maximal point. We will freely assume this without statement in subsequent proofs.
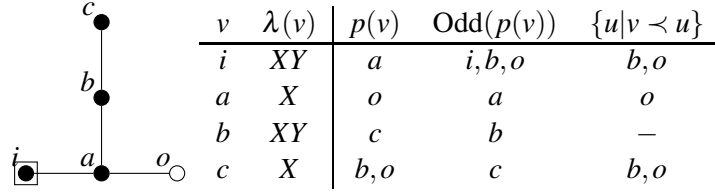
v	$\lambda(v)$	$p(v)$	$\mathrm{Odd}(p(v))$	$\{u\,|\,v \prec u\}$
i	XY	a	i,b,o	b,o
a	X	o	a	o
b	XY	c	b	$-$
c	X	b,o	c	b,o

Figure 2: A labelled open graph and a maximally delayed Pauli flow with a measured vertex (b) in $V_0^\prec$. If $(p,\prec)$ is focussed, then the primary extraction string for such a vertex is the identity, meaning the measurement angle/outcome has no effect on the overall process.

For causal flow and gflow, the set $V_0^\prec$ for a maximally delayed flow is exactly the set of outputs [5,27] since every measured vertex needs corrections. However, since Pauli flow allows some corrections to be effectively applied in the past of a vertex, we may now have some measured vertices in $V_0^\prec$. An example is given in Figure 2.

Lemma A.4. (Generalisation of [27, Lemma 1], [5, Lemma C.2]) If $(p,\prec)$ is a maximally delayed Pauli flow of (G,I,O,λ), then $V_0^\prec = O \cup V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$.

Proof. $V_0^\prec \subseteq O \cup V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$: Since the output case is trivial, suppose $u \in V_0^\prec$ is a non-output. We start by showing that $p(u) \subseteq (V_{-1}^\prec \cup \Lambda_u^X \cup \Lambda_u^Y \cup \{u\}) \cap \overline{I}$. $p(u) \subseteq \overline{I}$ from the definition of Pauli flow, since $p : \overline{O} \to \mathscr{P}[\overline{I}]$. By definition $V_{-1}^\prec = \emptyset$, so we just require $p(u) \subseteq \Lambda_u^X \cup \Lambda_u^Y \cup \{u\}$ which is given by condition $[\prec .X]$. It remains to show that $p(u) \setminus \{u\}$ is a suitable choice for K for at least one of the three $V_0^{\prec,PQ}$ sets.

Every vertex v satisfies $v \preceq u$ ($u \in V_0^\prec$), so condition $[\prec .Z]$ becomes $\mathrm{Odd}(p(u)) \setminus (\Lambda_u^Y \cup \Lambda_u^Z) \subseteq \{u\}$. Condition $[\prec .Y]$ gives $p(u) \cap \Lambda_u^Y = \mathrm{Odd}(p(u)) \cap \Lambda_u^Y$. We can rewrite these to the appropriate form from the definition of the $V_0^{\prec,PQ}$ sets based on whether or not u is in $p(u)$ and $\mathrm{Odd}(p(u))$. As the cases are mutually exclusive, we only place u into a single set with conditions $[\lambda.XY]$-$[\lambda.Y]$ restricting $\lambda(u)$ as needed.

$V_0^\prec \supseteq O \cup V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$: We will assume that there is some $w \in O \cup V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$ but $w \notin V_0^\prec$ and aim for a contradiction by generating a new Pauli flow that is more delayed than $(p,\prec)$. If $w \in O$, then we can simply construct the new Pauli flow $(p, \prec \setminus (\{w\} \times V))$. This is still a Pauli flow since the conditions only concern the measured vertices $\overline{O}$. This is also more delayed since $w \in V_0^{\prec \setminus (\{w\} \times V)}$ and no vertex is made deeper.

For the remaining case, we shall suppose that $w \in V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$ instead. Let $K \subseteq (\Lambda_w^X \cup \Lambda_w^y) \cap \overline{I}$ be a witness for $w \in V_0^{\prec,XY} \cup V_0^{\prec,XZ} \cup V_0^{\prec,YZ}$. We define $p' : \overline{O} \to \mathscr{P}[\overline{I}]$ and $\prec'$ be the map and partial order:

$$p'(u) := \begin{cases} K & \text{if } u = w \text{ and } w \in V_0^{\prec,XY} \\ K \cup \{w\} & \text{if } u = w \text{ and } w \notin V_0^{\prec,XY} \\ p(u) & \text{if } u \neq w \end{cases} \tag{20}$$

$$\prec' := \prec \setminus (\{w\} \times V) \tag{21}$$

For $u \neq w$, each of the Pauli flow conditions is trivially preserved, but we need to show they hold for $p'(w)$.

This satisfies condition $[\prec .X]$ from $K \subseteq (\Lambda_w^X \cup \Lambda_w^Y) \cap \overline{I}$. For each of the $V_0^{\prec,PQ}$ options, we get $\text{Odd}(p'(w)) \setminus (\Lambda_w^Y \cup \Lambda_w^Z) \subseteq \{w\}$, so we can show $[\prec .Z]$. Similarly, we get $p'(w) \cap \Lambda_w^Y = \text{Odd}(p'(w)) \cap \Lambda_w^Y$ in each of the $V_0^{\prec,PQ}$ options, so condition $[\prec .Y]$ also follows.

The membership of w in $p'(w)$ and $\text{Odd}(p'(w))$ is decided by which of the $V_0^{\prec,PQ}$ it belongs to, which also restricts $\lambda(w)$ to the compatible planes according to conditions $[\lambda.XY]$-$[\lambda.Y]$.

By assumption, $w \in V_k^{\prec}$ for some $k > 0$. By construction, the depth of every vertex under $\prec'$ is no larger than its depth under $\prec$ and $w \in V_0^{\prec'}$. This means $\forall k \geq 0. |V_{\cup k}^{\prec}| \leq |V_{\cup k}^{\prec'}|$ and $|V_0^{\prec}| < |V_0^{\prec'}|$, so $(p', \prec')$ is a more delayed Pauli flow than $(p, \prec)$. $\qquad\square$

Lemma A.5. (Generalisation of [27, Lemma 3], [5, Lemma C.4]) If $(p, \prec)$ is a maximally delayed Pauli flow of (G, I, O, λ), then $\forall k > 0. V_k^{\prec} = V_k^{\prec,XY} \cup V_k^{\prec,XZ} \cup V_k^{\prec,YZ}$.

Proof. For any given k, the proof of this follows the same strategy as Lemma A.4. When constructing the more delayed flow given a counterexample w, we need to adapt the definition of $\prec'$ to $\prec' := (\prec \setminus (\{w\} \times V)) \cup (\{w\} \times V_{\cup k-1}^{\prec})$. Vertices in $V_{\cup k-1}^{\prec}$ are not covered by the constraints in the definitions of $V_k^{\prec,PQ}$, so we add them all to the future of w to make sure we satisfy conditions $[\prec .X]$-$[\prec .Y]$. $\qquad\square$

These characterisations of the sets $V_k^{\prec}$ give us an iterative method of identifying them since we can simply search for possible witness sets K for each vertex.

Theorem A.6. (Restatement of Theorem 4.5) There exists an algorithm that decides whether a given labelled open graph has a Pauli flow, and that outputs such a Pauli flow if it exists. Moreover, this output is maximally delayed, and the algorithm completes deterministically in time that grows polynomially with the number of vertices in the graph.

Proof. The function `PauliFlow(`V, Γ, I, O, λ`)` in Algorithm 1 takes sets V and $I, O \subseteq V$ of vertices, an adjacency matrix Γ over V and a basis labelling function λ and returns "true" with a Pauli flow if one exists and "false" otherwise.

To see this, we consider the auxilliary method `PauliFlowAux(`$V, \Gamma, I, \lambda, A, B, k$`)` which aims to identify $V_k^{\prec}$, given sets $A = V_{\cup k-1}^{\prec}$ of possible correctors and $B \subseteq V_{\cup k}^{\prec}$ ($B \supseteq A$) of solved vertices, and then proceed recursively over the remainder of the graph for higher depths. For the case of $k = 0$, $O \subseteq V_0^{\prec}$ has already been handled by the setup in `PauliFlow(`V, Γ, I, O, λ`)` so it is sufficient to just find $V_k^{\prec,XY} \cup V_k^{\prec,XZ} \cup V_k^{\prec,YZ}$ due to Lemma A.4. For $k > 0$, $A = B$ so we are just finding $V_k^{\prec,XY} \cup V_k^{\prec,XZ} \cup V_k^{\prec,YZ}$ in accordance with Lemma A.5. In each recursive call, it examines each candidate vertex in turn and looks for a witness set K for membership into either $V_k^{\prec,XY}$, $V_k^{\prec,XZ}$, or $V_k^{\prec,YZ}$, from which we can identify a valid correction set. The global variables p and d map vertices to their correction sets and depths from the output (this defines the order $\prec$ over vertices with $v \prec w \Leftrightarrow d(v) > d(w)$).

This algorithm is guaranteed to terminate because V is finite and $B \subseteq V$ strictly grows with each call for $k \geq 1$, so C will eventually be an empty set.

The resulting Pauli flow must be maximally delayed because we are constructing the largest set possible at each value of k. If there were a more delayed Pauli flow $(p', \prec')$, there must be some minimal k for which $V_k^{\prec'} \setminus V_k^{\prec}$ is non-empty. However, if a vertex v belonged in this set then a suitable correction set for v must exist. Algorithm 1 would find this when testing v and place it in $V_k^{\prec}$.

For a given vertex u, finding the witness set K in each of the three possible search cases can be achieved by solving the linear equation system $M_{A,u} X_K = S_{\bar{\lambda}}$ in $\mathbb{F}_2$ defined by:

```
PauliFlow(V,Γ,I,O,λ) = begin
    L_X := ∅; L_Y := ∅; L_Z := ∅;
    forall v ∈ V do
        if v ∈ O then d(v) := 0;
        if λ(v) = X then L_X := L_X ∪ {v};
        if λ(v) = Y then L_Y := L_Y ∪ {v};
        if λ(v) = Z then L_Y := L_Z ∪ {v};
    end
    return PauliFlowAux(V,Γ,I,λ,∅,O,0);
end
PauliFlowAux(V,Γ,I,λ,A,B,k) = begin
    C := ∅;
    forall u ∈ B^C do
        if λ(u) ∈ {XY,X,Y} then
            K_XY := Solution for K ⊆ (A∪L_X∪L_Y)∩I^C \ {u} where
                K∩L_Y \ ({u}∪A) = Odd(K)∩L_Y \ ({u}∪A) and
                Odd(K)∩((A∪L_Y∪L_Z)^C∪{u}) = {u};
        end
        if λ(u) ∈ {XZ,X,Z} then
            K_XZ := {u}∪ (Solution for K ⊆ (A∪L_X∪L_Y)∩I^C \ {u} where
                K∩L_Y \ ({u}∪A) = Odd(K∪{u})∩L_Y \ ({u}∪A) and
                Odd(K∪{u})∩((A∪L_Y∪L_Z)^C∪{u}) = {u});
        end
        if λ(u) ∈ {YZ,Y,Z} then
            K_YZ := {u}∪ (Solution for K ⊆ (A∪L_X∪L_Y)∩I^C \ {u} where
                K∩L_Y \ ({u}∪A) = Odd(K∪{u})∩L_Y \ ({u}∪A) and
                Odd(K∪{u})∩((A∪L_Y∪L_Z)^C∪{u}) = ∅);
        end
        if a solution K_0 is found for any of K_XY,K_XZ,K_YZ then
            C := C∪{u};
            p(u) := K_0;
            d(u) := k;
        end
    end
    if C = ∅ and k > 0 then
        if B = V then return (true,p,d);
        else return (false,∅,∅);
    end
    else
        B' := B∪C;
        return PauliFlowAux(V,Γ,I,λ,B',B',k+1);
    end
end
```

Algorithm 1: An algorithm for identifying whether a labelled open graph has a Pauli flow.

$$M_{A,u} := \left[\frac{\Gamma \cap \mathbb{K}_{A,u} \times \mathbb{P}_{A,u}}{(\Gamma + \mathrm{Id}) \cap \mathbb{K}_{A,u} \times \mathbb{Y}_{A,u}} \right] \tag{22}$$

$$S_{\tilde{\lambda}} := \begin{cases} \left[\dfrac{\{u\}}{0} \right] & \text{if } \tilde{\lambda} = XY \\[2ex] \left[\dfrac{(N_\Gamma(u) \cap \mathbb{P}_{A,u}) \cup \{u\}}{N_\Gamma(u) \cap \mathbb{Y}_{A,u}} \right] & \text{if } \tilde{\lambda} = XZ \\[2ex] \left[\dfrac{N_\Gamma(u) \cap \mathbb{P}_{A,u}}{N_\Gamma(u) \cap \mathbb{Y}_{A,u}} \right] & \text{if } \tilde{\lambda} = YZ \end{cases} \tag{23}$$

where $\mathbb{K}_{A,u} := (A \cup \Lambda_u^X \cup \Lambda_u^Y) \cap I^C$ is the set of possible elements of the witness set K, $\mathbb{P}_{A,u} := (A \cup \Lambda_u^Y \cup \Lambda_u^Z)^C$ is the set of vertices in the past/present which should remain corrected afer measuring and correcting u, $\mathbb{Y}_{A,u} := \Lambda_u^Y \setminus A$ is the set of vertices we have to consider for condition $[\prec .Y]$, $\tilde{\lambda} \in \{XY, XZ, YZ\}$ denotes which of the three cases we are considering, and X_K is the column vector with 1 in the position of v if $v \in K$ and a 0 otherwise.

Taking $\tilde{\lambda} = XY$ as an example, the top block of equations encodes $\mathrm{Odd}(K) \cap \mathbb{P}_{A,u} = \{u\}$ and the lower block encodes $K \cap \Lambda_u^Y \setminus A = \mathrm{Odd}(K) \cap \Lambda_u^Y \setminus A$, so the solutions to these equations are exactly the possible witness sets K. Such solutions can be identified by Gaussian elimination and back substitution in $O(|V|^3)$ time.

This part of the algorithm is hit at most $|V|$ times per call to `PauliFlowAux`, which may be called at most $|V|$ times, hence the overall complexity of this algorithm is $O(|V|^5)$. $\qquad\square$

The complexity of this algorithm is higher than the $O(|V|^4)$ for the equivalent method for identifying gflow [5] [2]. This is because we can no longer do a single gaussian elimination per depth round because the matrix $M_{A,u}$ and the range of the witness set K are both dependent on the particular vertex u under consideration.

B   Focussed Sets and Pauli Flows

Even though the Pauli flow generated by Algorithm 1 is guaranteed to be maximally delayed, the correction sets may not be unique as the back substitution step may not have a single solution. In this section, we will investigate some transformations on Pauli flows resulting from this freedom to show that we can always reach a focussed Pauli flow.

Lemma B.1. Given a Pauli flow $(p, \prec)$ for a labelled open graph (G, I, O, λ) with two vertices $u, v \in \overline{O}$ such that $u \prec v$, then $(p', \prec)$ is a Pauli flow where $p'(u) := p(u) \Delta p(v)$ and $\forall w \in \overline{O} \setminus \{u\}.p'(w) := p(w)$. Moreover, if $(p, \prec)$ is maximally delayed, then so is $(p', \prec)$.

Proof. The Pauli flow conditions hold trivially for any vertex in $\overline{O} \setminus \{u\}$ since the correction sets have not changed, so it is sufficient to show they are preserved for u. We should first observe that $\mathrm{Odd}(p'(u)) = \mathrm{Odd}(p(u) \Delta p(v)) = \mathrm{Odd}(p(u)) \Delta \mathrm{Odd}(p(v))$.

$[\prec .X]$: For any $w \in p'(u)$ with $w \neq u$ and $\lambda(w) \notin \{X, Y\}$, we must have either $w \in p(u)$, $w = v$, or $w \in p(v) \wedge w \neq v$. In any of these cases, we have $u \prec w$ from $[\prec .X]$ for $(p, \prec)$ and $u \prec v$.

---

[2] It should be noted that Eslamy et al. provide an even more efficient algorithm for finding gflow in $O(|V|^3)$ time [17]. It may be possible to generalise this to Pauli flow in search of a more efficient routine.

$[\prec .Z]$: This follows similarly from $u \prec v$ and $[\prec .Z]$ on $\mathrm{Odd}(p(u))$ and $\mathrm{Odd}(p(v))$.

$[\prec .Y]$: For any $w \preceq u$ with $\lambda(w) = Y$, we also must have $w \preceq v$ since $u \prec v$. Hence by $[\prec .Y]$, $w \in p(u) \Leftrightarrow w \in \mathrm{Odd}(p(u))$ and the same for $p(v)$. Therefore, we find that $w \in p'(u) = p(u)\Delta p(v) \Leftrightarrow w \in \mathrm{Odd}(p(u))\Delta\mathrm{Odd}(p(v)) = \mathrm{Odd}(p'(u))$ as required.

$[\lambda.XY]$-$[\lambda.YZ]$: $u \notin p(v)$ and $u \notin \mathrm{Odd}(p(v))$ by $[\prec .X]$ and $[\prec .Z]$ since $u \prec v$, so the requirements are given by the corresponding conditions for $(p, \prec)$.

$[\lambda.X]$: $u \notin \mathrm{Odd}(p(v))$ by $[\prec .Z]$ and $u \in \mathrm{Odd}(p(u))$ by $[\lambda.X]$, so $u \in \mathrm{Odd}(p'(u))$.

$[\lambda.Z]$: $u \notin p(v)$ by $[\prec .X]$ and $u \in p(u)$ by $[\lambda.Z]$, so $u \in p'(u)$.

$[\lambda.Y]$: $u \in p(v) \Leftrightarrow u \in \mathrm{Odd}(p(v))$ by $[\prec .Y]$ and $u \in p(u) \Leftrightarrow u \notin \mathrm{Odd}(p(u))$ by $[\lambda.Y]$, then it is straightforward to show $u \in p'(u) \Leftrightarrow u \notin \mathrm{Odd}(p'(u))$ by cases.

The maximally delayed property of a Pauli flow only concerns the partial order between the vertices, so since $(p, \prec)$ and $(p', \prec)$ both use $\prec$ the property is trivially preserved. $\square$

This gives us a mechanism to generate new Pauli flows by adding correction sets together. We now show that this can help us to make progress towards satisfying the focussed property.

Lemma B.2. For any labelled open graph Γ, if sets $\hat{p}, \hat{q} \subseteq \overline{I}$ are focussed over $S \subseteq \overline{O}$, then so is $\hat{p}\Delta\hat{q}$.

Proof. $[FX]$: For any vertex $v \in (\hat{p}\Delta\hat{q}) \cap S$, we have either $v \in \hat{p}$ or $v \in \hat{q}$. Since $\hat{p}$ and $\hat{q}$ are focussed over S, the corresponding $[FX]$ condition gives $\lambda(v) \in \{XY, X, Y\}$.

$[FZ]$: Similarly, for any vertex $v \in \mathrm{Odd}(\hat{p}\Delta\hat{q}) \cap S = (\mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q})) \cap S$, $\lambda(v) \in \{XZ, YZ, Y, Z\}$ follows from either $v \in \mathrm{Odd}(\hat{p})$ or $v \in \mathrm{Odd}(\hat{q})$ and $[FZ]$.

$[FY]$: For any $v \in S$ with $\lambda(v) = Y$, we have $v \in \hat{p} \Leftrightarrow v \in \mathrm{Odd}(\hat{p})$ and $v \in \hat{q} \Leftrightarrow v \in \mathrm{Odd}(\hat{q})$ from $[FY]$ for $\hat{p}$ and $\hat{q}$. Hence $v \in \hat{p}\Delta\hat{q} \Leftrightarrow v \in \mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q})$. $\square$

Lemma B.3. For any labelled open graph Γ, if sets $\hat{p}, \hat{q} \subseteq \overline{I}$ are not focussed over $\{v\}$ ($v \in \overline{O}$), then $\hat{p}\Delta\hat{q}$ is focussed over $\{v\}$.

Proof. We consider each case for $\lambda(v)$ and how the focussed conditions could fail for $\hat{p}$ and $\hat{q}$:

$\lambda(v) \in \{XY, X\}$: $[FX]$ and $[FY]$ are trivially satisfied, so we must have $v \in \mathrm{Odd}(\hat{p})$ and $v \in \mathrm{Odd}(\hat{q})$ to fail $[FZ]$. This means $v \notin \mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q}) = \mathrm{Odd}(\hat{p}\Delta\hat{q})$, satisfying $[FZ]$ for $\hat{p}\Delta\hat{q}$.

$\lambda(v) \in \{XZ, YZ, Z\}$: Similarly, $[FZ]$ and $[FY]$ hold trivially, so we must have $v \in \hat{p}$ and $v \in \hat{q}$ to fail $[FX]$. We hence have $v \notin \hat{p}\Delta\hat{q}$, satisfying $[FZ]$ for $\hat{p}\Delta\hat{q}$.

$\lambda(v) = Y$: Now $[FX]$ and $[FZ]$ are trivial and we have $v \in \hat{p}\Delta\mathrm{Odd}(\hat{p})$ and $v \in \hat{q}\Delta\mathrm{Odd}(\hat{q})$ to fail $[FY]$. This now satisfies $[FY]$ since $v \notin (\hat{p}\Delta\mathrm{Odd}(\hat{p}))\Delta(\hat{q}\Delta\mathrm{Odd}(\hat{q})) = (\hat{p}\Delta\hat{q})\Delta\mathrm{Odd}(\hat{p}\Delta\hat{q})$. $\square$

Combining these two lemmas, we can find combinations of correction sets that fix unfocussed vertices whilst preserving those we have already focussed.

Lemma B.4. (Generalisation of [5, Lemma 3.13]) Let (G, I, O, λ) be a labelled open graph with a Pauli flow $(p, \prec)$ and some vertex $v \in \overline{O}$. Then there exists $p' : \overline{O} \to \mathscr{P}[\overline{I}]$ such that:

1. $\forall w \in \overline{O}. v = w \vee p'(w) = p(w)$;

2. $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$;

3. $(p', \prec)$ is a Pauli flow for (G, I, O, λ).

Proof. Let $J : \mathbb{Z}_{|\overline{O}|} \to \overline{O}$ be some indexing of the vertices that respects the order $\prec$ ($\forall i, j < |\overline{O}|.J(i) \prec J(j) \Rightarrow i < j$). We define a sequence of functions p_k as:

$$p_0(u) := p(u) \tag{24}$$

$$p_{k+1}(u) := \begin{cases} p_k(u)\Delta p_k(J(k)) & \text{if} \begin{pmatrix} u = v, \\ J(k) \neq v, \\ p_k(v) \text{ is not focussed over } \{J(k)\} \end{pmatrix} \\ p_k(u) & \text{otherwise} \end{cases} \tag{25}$$

1 is satisfied for all $(p_k, \prec)$ by construction, and 3 is also satisfied for all by Lemma B.1. To work towards 2, we proceed inductively with hypothesis $\Phi(k) :=$ "$p_k(v)$ is focussed over $\{J(i)\}_{i<k} \setminus \{v\}$". The $k = 0$ case holds vacuously.

Suppose we have $\Phi(k)$. If $J(k) = v$, then $p_{k+1}(v) = p_k(v)$ and $\Phi(k+1)$ is an immediate consequence of $\Phi(k)$. If $p_k(v)$ is focussed over $\{J(k)\}$, then $p_{k+1}(v) = p_k(v)$, so $\Phi(k+1)$ follows from this assumption and $\Phi(k)$. If, on the other hand, $p_k(v)$ is not focussed over $\{J(k)\}$, we have $p_{k+1}(v) = p_k(v)\Delta p_k(J(k))$. From conditions $[\lambda.XY]$-$[\lambda.Y]$, $p_k(J(k))$ is also not focussed over $\{J(k)\}$, so by Lemma B.3 we have $p_{k+1}(v)$ is focussed over $\{J(k)\}$. For any of the remaining $i < k$ (where $J(i) \neq v$), $\Phi(k)$ says that $p_k(v)$ is focussed over $\{J(i)\}$. Since J respects the order $\prec$, we also have $J(i) \preceq J(k)$, and hence conditions $[\prec.X]$-$[\prec.Y]$ imply that $p_k(J(k))$ is focussed over $\{J(i)\}$. We combine these with Lemma B.2 to deduce that $p_{k+1}(v)$ is also focussed over $\{J(i)\}$.

At the end of this chain, we have $(p_{|\overline{O}|}, \prec)$ where $p_{|\overline{O}|}$ is focussed over $\overline{O} \setminus \{v\}$. $\qquad\square$

Lemma B.5. (Restatement of Lemma 4.6) If a labelled open graph has a Pauli flow, then it has a maximally delayed, focussed Pauli flow.

Proof. Let $(p, \prec)$ be such a maximally delayed Pauli flow according to Remark A.3. Applying Lemma B.4 for each $v \in \overline{O}$ in turn, we reach some $(p', \prec)$ where, for every $v \in \overline{O}$, $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$, i.e. $(p', \prec)$ is a focussed Pauli flow. Since the partial order $\prec$ remains the same, this is also still maximally delayed. $\qquad\square$

For general measurement patterns, even focussed Pauli flows may not be unique. However, given multiple focussed Pauli flows, the differences between their correction sets are given by the focussed sets of the graph.

Lemma B.6. Let $\Gamma = (G, I, O, \lambda)$ be a labelled open graph with a focussed Pauli flow $(p, \prec)$ and a focussed set $\hat{p} \subseteq \overline{I}$. Let $v \in \overline{O}$ be a vertex such that $\forall w \in \hat{p} \cup \text{Odd}(\hat{p}).\lambda(w) \in \{XY, XZ, YZ\} \Rightarrow w \neq v \land v \preceq w$. Then $(p', \prec')$ is a focussed Pauli flow, where:

$$p'(w) := \begin{cases} p(w)\Delta\hat{p} & \text{if } w = v \\ p(w) & \text{if } w \neq v \end{cases} \tag{26}$$

and $\prec'$ is the transitive closure of $\prec \cup \{(v, w) | w \in \hat{p} \cup \text{Odd}(\hat{p}) \land \lambda(w) \in \{XY, XZ, YZ\}\}$.

Proof. Firstly, $\prec'$ is still a strict partial order. Transitivity is immediate from the definition as a transitive closure. For antisymmetry (and similarly for strictness), suppose on the contrary that we have some $a \prec' b$ and $b \prec' a$ (or directly $a \prec' a$). This gives a transitive loop $[a, \ldots, b, \ldots, a]$ where each step is

either in $\prec$ or $\{(v,w)|w \in \hat{p} \cup \mathrm{Odd}(\hat{p}) \wedge \lambda(w) \in \{XY,XZ,YZ\}\}$. Since $\prec$ is a strict partial order, we cannot have such a loop where every step is in $\prec$, so at least one step must be some such (v,w). We can freely eliminate inner loops around v, so we can assume wlog that v only appears once in the loop. This means the rest of the loop is only from $\prec$, so $w \prec v$ by transitivity. However, we assumed that $v \preceq w$ since $w \in \hat{p} \cup \mathrm{Odd}(\hat{p})$ and $\lambda(w) \in \{XY,XZ,YZ\}$, giving us the contradiction we need.

Conditions $[\prec .X]$-$[\prec .Y]$ are preserved from the extension to $\prec'$ covering planar labels and the focussed property covering Pauli labels.

Conditions $[\lambda.XY]$-$[\lambda.YZ]$ are preserved since $v \notin \hat{p} \cup \mathrm{Odd}(\hat{p})$ gives $v \in p'(v) \Leftrightarrow v \in p(v)$ and $v \in \mathrm{Odd}(p'(v)) \Leftrightarrow v \in \mathrm{Odd}(p(v))$.

For conditions $[\lambda.X]$-$[\lambda.Y]$, the correction amounts to saying that $p'(v)$ is not focussed over $\{v\}$. If this were not the case, then $p'(v)\Delta\hat{p} = p(v)$ would be focussed over $\{v\}$ by Lemma B.2, contradicting the corresponding Pauli flow condition for $p(v)$.

Finally, $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$ using Lemma B.2, so $(p', \prec')$ is focussed. $\square$

Lemma B.7. Let Γ be a labelled open graph with two focussed Pauli flows $(p, \prec)$ and $(p', \prec')$. Then for any vertex $v \in \overline{O}$, $p(v)\Delta p'(v)$ is a focussed set.

Proof. Each of $p(v)$ and $p'(v)$ are focussed over $\overline{O} \setminus \{v\}$, so their combination must also be by Lemma B.2. For each case of $\lambda(v)$, the corresponding condition from $[\lambda.XY]$-$[\lambda.Y]$ is then enough to show that $p(v)\Delta p'(v)$ is also focussed over $\{v\}$. $\square$

An important consequence of this result is that in order to fully identify the space of all focussed Pauli flows for a given labelled open graph, it is sufficient to find one using Algorithm 1 that we focus with Lemma 4.6, and find all of the focussed sets. The following proofs show that the focussed sets form a group, allowing us to only need some generating set, and giving an algorithm for finding such generators.

Lemma B.8. The focussed sets of a labelled open graph form a group under Δ.

Proof. Closure is a direct consequence of Lemma B.2 with $S = \overline{O}$. We then extend this to a group with identity $\emptyset$ and each focussed set is self-inverse. $\square$

Lemma B.9. Any labelled open graph $\Gamma = (G,I,O,\lambda)$ with a focussed Pauli flow has $2^{|O|-|I|}$ distinct focussed sets.

Proof. We can find the number of subsets of $\overline{I}$ that are focussed over some $S \subseteq \overline{O}$ inductively over the size of S using the following induction hypothesis:

$$\Phi(k) := \forall S \subseteq \overline{O}.|S| = k \Rightarrow \left|\left\{\hat{p} \subseteq \overline{I} \big| \hat{p} \text{ is focussed over } S\right\}\right| = 2^{|\overline{I}|-|S|} \tag{27}$$

For $k = 0$, we only have to consider $S = \emptyset$. Any subset of $|\overline{I}|$ satisfies the required properties, so there are $2^{|\overline{I}|}$ many such sets, so $\Phi(0)$ holds.

Suppose $\Phi(k)$ holds for some $k \geq 0$. Suppose $S \subseteq \overline{O}$ is of size $k+1$ and choose any vertex $v \in S$. Let $S_v := S \setminus v$, so $|S_v| = k$. $\Phi(k)$ implies there are $2^{|\overline{I}|-k}$ subsets of $\overline{I}$ that are focussed over S_v. The subsets focussed over S are those that are focussed over both S_v and $\{v\}$.

Let $(p, \prec)$ be some focussed Pauli flow for Γ, so $p(v)$ is focussed over $\overline{O} \setminus v$, and hence for S_v. Lemma B.2 implies the sets focussed over S_v are closed under symmetric difference with $p(v)$. However, Lemma B.3 implies $\hat{p}$ is focussed over $\{p\}$ if and only if $\hat{p}\Delta p(v)$ is not because $p(v)$ corrects the measurement at v. This means taking the symmetric difference with $p(v)$ defines a bijection between the sets that are

focussed for S and those that are focussed for S_v but not for $\{v\}$. Since these are disjoint and partition the $2^{|\bar{I}|-k}$ that are focussed over S_v, we must have $2^{|\bar{I}|-(k+1)}$ that are focussed over S, giving $\Phi(k+1)$.

Focussed sets are those that are focussed over the entirety of $\overline{O}$, so $\Phi(|\overline{O}|)$ tells us that there are $2^{|\bar{I}|-|\overline{O}|} = 2^{|O|-|I|}$ focussed sets. $\qquad\square$

Lemma B.10. Given a labelled open graph Γ with focussed Pauli flow, there exists an algorithm that identifies $|O|-|I|$ independent generators for the group of focussed sets of Γ. Furthermore, this algorithm completes in time polynomial in the number of vertices in Γ.

Proof. Similar to Theorem 4.5, we can encode the conditions for focussed sets into a linear equation system $MX = S$ in $\mathbb{F}_2$ which can be solved by Gaussian elimination and back substitution to obtain a single focussed set.

$$M := \left[\frac{\Gamma \cap \mathbb{P} \times \mathbb{O}^C}{(\Gamma + \mathrm{Id}) \cap \mathbb{P} \times \mathbb{Y}} \right] \tag{28}$$

$$S := \left[\frac{0}{0} \right] \tag{29}$$

$$\mathbb{P} := (O \cup \{w \in \overline{O} | \lambda(w) \in \{XY, X, Y\}\}) \cap \overline{I} \tag{30}$$

$$\mathbb{O} := O \cup \{w \in \overline{O} | \lambda(w) \in \{XZ, YZ, Y, Z\}\} \tag{31}$$

$$\mathbb{Y} := \{w \in \overline{O} | \lambda(w) = Y\} \tag{32}$$

In the above, we let Γ stand for the adjacency matrix of its graph. We define $\mathbb{P}$ to be the set of vertices that could be included in a focussed set and $\mathbb{O}$ the set of vertices that could be in the odd neighbourhood. Solutions satisfy $[FX]$ since X only ranges over $\mathbb{P}$. The top block of the system encodes $[FZ]$ since multiplying by Γ in $\mathbb{F}_2$ gives the odd neighbourhood, and similarly the bottom block encodes $[FY]$.

We need to add additional conventions to make sure that the focussed sets we obtain are non-empty and independent. Since this system is underconstrained (when there exist non-empty focussed sets), there will always be some freedom of choice during the back substitution step. Since every focussed set must be a solution, these free substitutions must generate the full set. Hence we can obtain independent, non-empty focussed sets by taking a single free substitution for each focussed set, iterating through each free substitution in turn.

Since both $|\mathbb{P}|$ and $|\mathbb{O}^C| + |\mathbb{Y}|$ are at most $|V|$, the Gaussian elimination takes $O(|V|^3)$ time. Each back substitution takes $O(|V|^2)$ time, which we must repeat $|O| - |I|$ times giving $O(|V|^3)$ again. $\qquad\square$

It should be noted that similar conditions can be incorporated into Algorithm 1 to directly obtain focussed Pauli flows by imposing further restrictions on the vertices that can be included in the correction sets and adding extra equations to restrict the odd neighbourhood.

A final comment we will make about the focussed conditions is that it removes the need for any ordering relation between Pauli vertices. When the measurement pattern uses only Pauli measurements, all of them can be measured simultaneously, giving a single round of corrections on the outputs. When there are additional planar measurements, we can do all Pauli measurements in a single layer before considering any of the planar measurements.

Lemma B.11. If a labelled open graph has a Pauli flow, then there exists a Pauli flow $(p, \prec)$ which satisfies $\forall v \in \overline{O}.\lambda(v) \in \{X, Y, Z\} \Rightarrow \forall u \in V.v \preceq u$.

Proof. Given a Pauli flow $(p, \prec)$, we can assume wlog that it is focussed from Lemma 4.6. Let $\prec' := \prec \setminus \{(u, v) \in V \times \overline{O} | \lambda(v) \in \{X, Y, Z\}\}$. By construction, this satisfies $\forall v \in \overline{O}. \lambda(v) \in \{X, Y, Z\} \Rightarrow \forall u \in V. v \preceq' u$. To show that $(p, \prec')$ is a valid Pauli flow, we just need to show that $[\prec .X]$-$[\prec .Y]$ are unaffected which all follow from the focussed conditions. □

C   Proofs for Circuit Extraction

This section will explore how to identify extraction strings and their properties, building up to a proof of Theorem 4.7 for circuit extraction. As dicussed in Section 4, the key principle is to extract planar measurement angles as rotations over the outputs, to leave behind a stabilizer process. We start by building up an explicit construction for extraction strings from a focussed Pauli flow.

Lemma C.1. Given a Pauli flow $(p, \prec)$ for a labelled open graph (G, I, O, λ), for any vertex $v \in \overline{O}$ the size of $p(v) \cap \text{Odd}(p(v))$ is even.

Proof. Let $G = (V, E)$ and define the subgraph $G' = (p(v), E \cap (p(v) \times p(v)))$. For any vertex $w \in V$, $w \in p(v) \cap \text{Odd}(p(v))$ if and only if w has odd degree in G'. Since the sum of the vertex degrees must equal 2 times the number of edges in G', there must be an even number of vertices with odd degree. □

Lemma C.2. (Restatement of Lemma 4.4) Let $\Gamma = (G, I, O, \lambda)$ be a labelled open graph with some measurement angles $\alpha : \overline{O} \to [0, 2\pi)$ and a focussed Pauli flow $(p, \prec)$. Then for any vertex $v \in \overline{O}$, $p(v)$ determines a $P^{\perp v}$-extraction string.

Proof. Consider an arbitrary measured vertex $v \in \overline{O}$.

Since each correction set $p(v)$ is a subset of the non-input vertices, we can combine the graph state stabilizers. We may reorder the Z and X terms with the possible introduction of a (-1).

$$E_G N_{\overline{I}} = (-1)^a \left(\prod_{u \in p(v)} X_u \right) \left(\prod_{u \in \text{Odd}(p(v))} Z_u \right) E_G N_{\overline{I}} \tag{33}$$

where $a = |E \cap (p(v) \times p(v))|$ is the number of edges in the subgraph of $p(v)$, since for each edge here we have to reorder the Z and X terms on exactly one of the two vertices.

There are an even number of vertices with both an X and a Z (see Lemma C.1), so we can apply $Y = iXZ$ on all such instances, again introducing a possible (-1) term.

$$E_G N_{\overline{I}} = (-1)^b \left(\prod_{u \in p(v) \setminus \text{Odd}(p(v))} X_u \right) \left(\prod_{u \in \text{Odd}(p(v)) \setminus p(v)} Z_u \right) \left(\prod_{u \in p(v) \cap \text{Odd}(p(v))} Y_u \right) E_G N_{\overline{I}} \tag{34}$$

where $b = a + |p(v) \cap \text{Odd}(p(v))|/2$.

Since $(p, \prec)$ is a Pauli flow, every vertex in $p(v)$ or $\text{Odd}(p(v))$ is either a Pauli measurement or greater than v in $\prec$ (from conditions $[\prec .X]$ and $[\prec .Z]$). To fit the form from Equation 12, we consider adding the corresponding projections. Since $(p, \prec)$ is focussed, each vertex (besides outputs and v) in $p(v) \setminus \text{Odd}(p(v))$ is projected into an X basis eigenvector, and similarly $\text{Odd}(p(v)) \setminus p(v)$ into Z and

$p(v) \cap \mathrm{Odd}(p(v))$ into Y. This means we can absorb these Pauli operators into the projections, again with the possible introduction of a (-1).

$$\left(\prod_{\substack{u \succ v \\ \lambda(u) \notin \{X,Y,Z\}}} \langle +_{\lambda(u),0}|_u \right) \left(\prod_{\substack{u \in \overline{O} \backslash \{v\} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u),\alpha(u)}|_u \right) E_G N_{\overline{I}}$$

$$= (-1)^c \left(\prod_{\substack{u \succ v \\ \lambda(u) \notin \{X,Y,Z\}}} \langle +_{\lambda(u),0}|_u \right) \left(\prod_{\substack{u \in \overline{O} \backslash \{v\} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u),\alpha(u)}|_u \right)$$

$$\left(\prod_{u \in (p(v) \backslash \mathrm{Odd}(p(v))) \cap (O \cup \{v\})} X_u \right) \left(\prod_{u \in (\mathrm{Odd}(p(v)) \backslash p(v)) \cap (O \cup \{v\})} Z_u \right) \tag{35}$$

$$\left(\prod_{u \in p(v) \cap \mathrm{Odd}(p(v)) \cap (O \cup \{v\})} Y_u \right) E_G N_{\overline{I}}$$

where $c = b + \left| (p(v) \cup \mathrm{Odd}(p(v))) \cap \{u \in \overline{O} | \lambda(u) \in \{X,Y,Z\} \wedge \alpha(u) = \pi\} \right|$.

This is now a Pauli operator with $P^{\perp v}$ on v and remainder over only the outputs, making it a valid primary extraction string. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\square$

Recall that the Product Rotation Lemma can use these extraction strings to remove the measurement angles. We can summarise Equation 10 by

$$\langle +_{\lambda(v),\alpha(v)}| \approx \langle +_{\lambda(v),0}| e^{(-1)^{D_v} i \frac{\alpha(v)}{2} P^{\perp v}} \tag{36}$$

$$D_v := \begin{cases} 1 & \text{if } \lambda(v) = YZ \\ 0 & \text{otherwise} \end{cases} \tag{37}$$

where D_v dictates the direction of rotation about the Bloch sphere. This means the rotation we extract over the outputs is $e^{(-1)^{D_v} i \frac{\alpha(v)}{2} \mathbf{P}^{\perp v}}$.

After extracting all planar measurement angles, we are left with the following stabilizer process:

$$\mathscr{C} = \left(\prod_{\substack{u \in \overline{O} \\ \lambda(u) \notin \{X,Y,Z\}}} \langle +_{\lambda(u),0}|_u \right) \left(\prod_{\substack{u \in \overline{O} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u),\alpha(u)}|_u \right) E_G N_{\overline{I}} \tag{38}$$

We can characterise this by finding its isometry tableau. Recall from Section 3 that the isometry tableau generalises variants of the stabilizer tableau representation from previous literature for both stabilizer states [33], where the rows represent generators for the stabilizer group, and unitaries [1], where the rows represent the actions of the unitary on Z_u and X_u for each input u - that is, the operator $\mathbf{P}$ such that $\mathbf{P}\mathscr{C}Z_u = \mathscr{C}$. It is enough for us to find such Z_u and X_u actions for each input along with $|O| - |I|$ generators for the free stabilizers over the outputs.

Since Z on an input will commute through the CZ gates, we can get the corresponding stablizer row from the primary extraction string assuming $P^{\perp u} = Z$ for this input. This is guaranteed since the correction set cannot contain any inputs, including itself.

We can use the same trick to obtain the X rows from primary extraction strings. In this case, we find an altered pattern (Γ', α') such that $\mathscr{C} \approx \mathscr{C}' H_u$. The Hadamard maps X to a Z, meaning we can use the extraction string with respect to (Γ', α'). One way to achieve this is by input extension.

Definition C.3 (Input Extension). Given an open graph (G, I, O), *input extension* about $u \in I$ adds a new vertex u' that is only connected to u and replaces u in the input set. Formally, $V' = V \cup \{u'\}$, $E' = E \cup \{u' \sim u\}$, $I' = (I \cup \{u'\}) \setminus \{u\}$.

Lemma C.4. Let (Γ, α) describe a measurement pattern with some chosen input $u \in I$. Taking an input extension about u gives an equivalent measurement pattern (Γ', α') assuming $\lambda' = \lambda \cup \{u' \mapsto XY\}$, $\alpha' = \alpha \cup \{u' \mapsto 0\}$, and a Hamadard gate is applied to input u' before the start of the pattern.

Proof. It is enough to verify the following equation:

$$H = \sqrt{2} \left(\langle +_{XY,0} | \otimes \mathbb{I} \right) CZ \left(\mathbb{I} \otimes | +_{X,0} \rangle \right) \tag{39}$$

$\square$

In order to actually use this, we need to show that Pauli flows are preserved by input extensions.

Lemma C.5. (Generalisation of [5, Lemma 3.8]) Let Γ and Γ' be labelled open graphs related by an input extension on vertex $u \in I$ (generating $u' \in I'$). If Γ has a Pauli flow, then so does Γ'.

Proof. Suppose Γ has a Pauli flow $(p, \prec)$. Let $p' = p \cup \{u' \mapsto \{u\}\}$ and $\prec'$ be the transitive closure of $\prec \cup \{(u', w) | w \in N_G(u) \cup \{u\}\}$.

For any $v \in V \setminus O$, $u' \notin p'(v) = p(v)$ and $u' \notin \mathrm{Odd}_{G'}(p'(v)) = \mathrm{Odd}_G(p(v))$ since its only neighbour u is an input in Γ which could not appear in any correction sets from p. This allows us to inherit all of the Pauli flow properties from $(p, \prec)$ for v as they have remained unchanged.

For u', the definition of $\prec'$ guarantees conditions $[\prec .X]$-$[\prec .Y]$ since $u' \prec' w$ for every $w \in N_G(u) \cup \{u\} = (\mathrm{Odd}_{G'}(p'(u')) \setminus \{u'\}) \cup p'(u')$. From the construction of $p'(u')$, we satisfy condition $[\lambda .XY]$, and the remainder hold trivially. $\square$

The remaining stabilizer generators can be obtained from the focussed sets of the measurement pattern.

Lemma C.6. Given a measurement pattern (Γ, α) where the labelled open graph $\Gamma = (G, I, O, \lambda)$ has a focussed set $\hat{p}$, then $\hat{p}$ determines a stabilizer $\hat{\mathbf{P}}$ of the linear map:

$$\left(\prod_{\substack{w \in \overline{O} \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0} |_w \right) \left(\prod_{\substack{w \in \overline{O} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)} |_w \right) E_G N_{\overline{I}} \tag{40}$$

Proof. This follows similarly to Lemma 4.4. After applying the measurement projections, the only Pauli terms that are not absorbed into the projections are over the output qubits. $\square$

Lemma B.9 implies that we can get $|O| - |I|$ generators for the group of focussed sets, and we are looking for $|O| - |I|$ generators for the free stabilizers. For this to work, we need the map from Lemma C.6 to be a bijection (i.e. the generators for the focussed sets don't degenerate when mapped into stabilizers). We go one step further by showing that it also preserves the group action as a result of Lemmas C.7 and C.8.

Lemma C.7. Let Γ be a labelled open graph. For any set of vertices $S \subseteq \overline{I}$, denote the output substring of the stabilizer as $\mathbf{P}^S = \left(\prod_{w \in O \cap S} X_w \right) \left(\prod_{w \in O \cap \mathrm{Odd}(S)} Z_w \right)$. Then for any $S, T \subseteq \overline{I}$ we have $\mathbf{P}^S \mathbf{P}^T \approx \mathbf{P}^{S \Delta T}$.

Proof. This follows from the anticommutativity of X and Z and $\mathrm{Odd}(S \Delta T) = \mathrm{Odd}(S) \Delta \mathrm{Odd}(T)$.

$$
\begin{aligned}
\mathbf{P}^S\mathbf{P}^T &= \left(\prod_{w\in O\cap S} X_w \right)\left(\prod_{w\in O\cap \mathrm{Odd}(S)} Z_w \right)\left(\prod_{w\in O\cap T} X_w \right)\left(\prod_{w\in O\cap \mathrm{Odd}(T)} Z_w \right) \\
&\approx \left(\prod_{w\in O\cap S} X_w \right)\left(\prod_{w\in O\cap T} X_w \right)\left(\prod_{w\in O\cap \mathrm{Odd}(S)} Z_w \right)\left(\prod_{w\in O\cap \mathrm{Odd}(T)} Z_w \right) \\
&= \left(\prod_{w\in O\cap (S\Delta T)} X_w \right)\left(\prod_{w\in O\cap \mathrm{Odd}(S\Delta T)} Z_w \right) \\
&= \mathbf{P}^{S\Delta T}
\end{aligned}
\tag{41}
$$

$\square$

Lemma C.8. For any non-zero linear map $\mathscr{C}$ with stabilizers A and B, if A and B share the same Pauli string then $A = B$.

Proof. If A and B share the same Pauli string, they can only differ by phase, so $B = e^{i\theta}A$. Since stabilizers form a group, $AB = A(e^{i\theta}A) = e^{i\theta}I$ is also a stabilizer, i.e. $\mathscr{C} = e^{i\theta}\mathscr{C}$. $e^{i\theta} \neq 1$ implies $\mathscr{C}$ is a zero map, so we must have $A = B$. $\square$

Because the groups of focussed sets and free stabilizers have the same finite cardinality, injectivity is enough to show that we have an isomorphism. We can prove this by showing that only the empty set gets mapped to the identity.

Lemma C.9. Let (Γ, α) describe a measurement pattern with a focussed Pauli flow $(p, \prec)$. Then for any vertices $u, v \in \overline{O}$ the primary extraction strings satisfy

$$
\mathbf{P}^{\perp u}\mathbf{P}^{\perp v} = (-1)^{F^p_{u\to v} + F^p_{v\to u}}\mathbf{P}^{\perp v}\mathbf{P}^{\perp u}
\tag{42}
$$

where $F^p_{x\to y} := |\{y\} \cap (p(x) \cup \mathrm{Odd}(p(x)))|$ indicates whether y is used in the correction of x. Similarly, given a focussed set $\hat{p}$, its stabilizer satisfies

$$
\mathbf{P}^{\perp u}\hat{\mathbf{P}} = (-1)^{G_{\hat{p}\to u}}\hat{\mathbf{P}}\mathbf{P}^{\perp u}
\tag{43}
$$

where $G_{\hat{p}\to u} = |\{u\} \cap (\hat{p} \cup \mathrm{Odd}(\hat{p}))|$ indicates whether u is used in the generation of the stabilizer.

Proof. Since $\mathbf{P}^{\perp u}$ and $\mathbf{P}^{\perp v}$ are tensor products of Pauli matrices, they must either commute or anticommute. Consider the linear map $\mathscr{C}$ given by:

$$
\mathscr{C} := \left(\prod_{\substack{w\in \overline{O}\setminus\{u,v\} \\ \lambda(w)\notin\{X,Y,Z\} \\ u\prec w \vee v\prec w}} \langle +_{\lambda(w),0}|_w \right)\left(\prod_{\substack{w\in \overline{O}\setminus\{u,v\} \\ \lambda(w)\in\{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right) E_G N_{\overline{I}}
\tag{44}
$$

This may not be in the exact form needed to introduce extraction strings for u and v if they use each other for corrections. Let $P^{x\to y}$ be the Pauli induced on y when correcting x.

$$P^{x\to y} := \begin{cases} I & \text{if } y \notin p(x) \cup \text{Odd}(p(x)) \\ X & \text{if } y \in p(x) \setminus \text{Odd}(p(x)) \\ Y & \text{if } y \in p(x) \cap \text{Odd}(p(x)) \\ Z & \text{if } y \in \text{Odd}(p(x)) \setminus p(x) \end{cases} \tag{45}$$

We can still follow the construction in the proof of Lemma 4.4 to deduce that $(-1)^{A^p_{u\to v}} P^{\perp u}_u P^{u\to v}_v \mathbf{P}^{\perp u}$ and $(-1)^{A^p_{v\to u}} P^{v\to u}_u P^{\perp v}_v \mathbf{P}^{\perp v}$ are stabilizers of $\mathscr{C}$ where

$$A^p_{x\to y} := \begin{cases} F^p_{x\to y} & \text{if } \lambda(y) \in \{X,Y,Z\} \wedge \alpha(y) = \pi \\ 0 & \text{otherwise} \end{cases} \tag{46}$$

Comparing conditions $[\lambda.XY]$-$[\lambda.Y]$ and the focussed conditions, we see that, for any vertex $x \in \overline{O}$, $P^{\perp x}$ is orthogonal to the measurement plane/Pauli of x and hence anticommutes with any $P^{y\to x} \neq I$. Equationally, $P^{\perp x} P^{y\to x} = (-1)^{F^p_{y\to x}} P^{y\to x} P^{\perp x}$.

The stabilizer group of $\mathscr{C}$ is abelian, so we have

$$\begin{aligned} & (-1)^{A^p_{u\to v}} P^{\perp u}_u P^{u\to v}_v \mathbf{P}^{\perp u} (-1)^{A^p_{v\to u}} P^{v\to u}_u P^{\perp v}_v \mathbf{P}^{\perp v} \\ =& (-1)^{A^p_{v\to u}} P^{v\to u}_u P^{\perp v}_v \mathbf{P}^{\perp v} (-1)^{A^p_{u\to v}} P^{\perp u}_u P^{u\to v}_v \mathbf{P}^{\perp u} \\ =& (-1)^{F^p_{u\to v}+F^p_{v\to u}} (-1)^{A^p_{u\to v}} P^{\perp u}_u P^{u\to v}_v \mathbf{P}^{\perp v} (-1)^{A^p_{v\to u}} P^{v\to u}_u P^{\perp v}_v \mathbf{P}^{\perp u} \end{aligned} \tag{47}$$

i.e. $\mathbf{P}^{\perp u} \mathbf{P}^{\perp v} = (-1)^{F^p_{u\to v}+F^p_{v\to u}} \mathbf{P}^{\perp v} \mathbf{P}^{\perp u}$.

The proof for focussed sets is virtually the same, though we only need care about anticommuting Paulis on u. $\square$

Lemma C.10. Given a labelled open graph $\Gamma = (G,I,O,\lambda)$ with a focussed Pauli flow and any focussed set $\hat{p}$, then $(\hat{p} \cup \text{Odd}(\hat{p})) \cap O = \emptyset \Leftrightarrow \hat{p} = \emptyset$.

Proof. $\hat{p} = \emptyset \Rightarrow (\hat{p} \cup \text{Odd}(\hat{p})) \cap O = \emptyset$ is trivial. For the other direction, suppose for a contradiction that we have a non-empty $\hat{p}$ for which $(\hat{p} \cup \text{Odd}(\hat{p})) \cap O = \emptyset$, so we have some $v \in \hat{p} \cap \overline{O}$ and the corresponding stabilizer is the identity. However, Lemma C.9 implies that the stabilizer must anticommute with the primary extraction string $\mathbf{P}^{\perp v}$, contradicting the identity assumption. $\square$

Theorem C.11. Let (Γ, α) describe a measurement pattern with a focussed Pauli flow. Then the construction from Lemma C.6 gives an isomorphism between the groups of focussed sets and the stabilizers of the linear map for (Γ, α) after removing all planar measurement angles.

Proof. The group action is preserved due to Lemmas C.7 and C.8. By Lemma C.10, the map is injective (if we had two distinct focussed sets with the same stabilizer, combining them with Δ would give a non-empty focussed set that maps to the identity stabilizer) and subsequently bijective by cardinality from Lemma B.9. $\square$

This isomorphism guarantees that the focussed sets identified by Lemma B.10 actually give a generating set for the stabilizer group, giving a valid isometry tableau. All that remains is to put everything together into a formal procedure.

Theorem C.12. (Restatement of Theorem 4.7) Let (Γ, α) describe a measurement pattern where Γ has a Pauli flow. Then there is an algorithm that identifies an equivalent circuit requiring no ancillae which completes in time polynomial in the number of vertices in Γ.

Proof. Using Theorem 4.5 and Lemma 4.6, we can identify a maximally delayed, focussed Pauli flow $(p, \prec)$. Let $J : \mathbb{Z}_{|\overline{O}|} \to \overline{O}$ be an indexing over the measured qubits such that $J(i) \prec J(j) \Rightarrow j < i$, i.e. J iterates from outputs to inputs respecting $\prec$. We will define a sequence of patterns (Γ, α_k) and circuits C_k such that $[\![C_k]\!] \cdot [\![\Gamma; \alpha_k]\!] \approx [\![\Gamma; \alpha]\!]$ (where $[\![-]\!]$ denotes the interpretation as a linear function). We start with $\alpha_0 = \alpha$ and C_0 is the empty (identity) circuit.

For any $k \geq 0$, the update from α_k to α_{k+1} is just setting the angle of $J(k)$ to 0 if it is planar.

$$\alpha_k(J(j)) := \begin{cases} 0 & \text{if } j < k \wedge \lambda(J(i)) \in \{XY, XZ, YZ\} \\ \alpha(J(j)) & \text{otherwise} \end{cases} \tag{48}$$

Defining the update from C_k to C_{k+1} requires us to find a Pauli string $\mathbf{P}$ over the outputs such that:

$$
\begin{aligned}
&\left(\prod_{j<k \wedge \lambda(J(j)) \notin \{X,Y,Z\}} \langle +_{\lambda(J(j)),0}|_{J(j)} \right) \\
&\left(\prod_{j \geq k \vee \lambda(J(j)) \in \{X,Y,Z\}} \langle +_{\lambda(J(j)),\alpha(J(j))}|_{J(j)} \right) E_G N_{\overline{I}} \\
\approx\, & e^{i\frac{\alpha(J(k))}{2}\mathbf{P}} \left(\prod_{j \leq k \wedge \lambda(J(j)) \notin \{X,Y,Z\}} \langle +_{\lambda(J(j)),0}|_{J(j)} \right) \\
&\left(\prod_{j > k \vee \lambda(J(j)) \in \{X,Y,Z\}} \langle +_{\lambda(J(j)),\alpha(J(j))}|_{J(j)} \right) E_G N_{\overline{I}}
\end{aligned}
\tag{49}
$$

When $\lambda(J(k)) \in \{X, Y, Z\}$ the projections on each side of Equation 49 are identical, meaning it suffices to let $\mathbf{P}$ be the identity and set $C_{k+1} = C_k$.

When $\lambda(J(k)) \in \{XY, XZ, YZ\}$, the projectors differ by a $e^{(-1)^{p_{J(k)}} i \frac{\alpha(J(k))}{2} P_{J(k)}^{\perp J(k)}}$ according to Equation 10. Equation 49 then holds by using the Product Rotation Lemma with a primary extraction string for $J(k)$, which we can obtain by Lemma 4.4. We decompose this rotation into basic gates (for example, by *CX* ladder constructions [6, 11]) and add them to the start of C_k so that $[\![C_{k+1}]\!] = [\![C_k]\!] \cdot e^{(-1)^{p_{J(k)}} i \frac{\alpha(J(k))}{2} \mathbf{P}^{\perp J(k)}}$.

At the end of this sequence, we have the pattern $(\Gamma, \alpha_{|\overline{O}|})$ where every measurement is in a Pauli basis and the associated circuit $C_{|\overline{O}|}$. This means the corresponding isometry for this pattern is a stabilizer process.
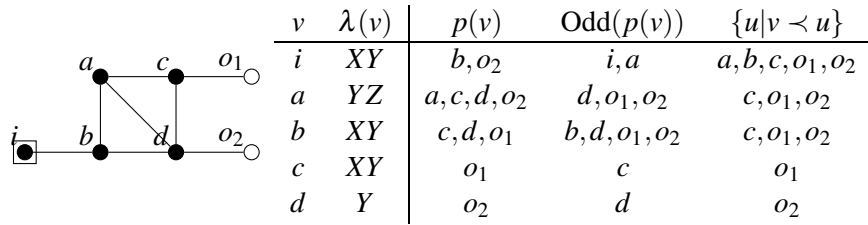
To characterise this process, it is sufficient to identify its isometry tableau. For each input, we can find the Z row from the primary extraction string of the input vertex, and by taking an input extension with Lemma C.5 we can similarly get the X row. Each of our $|O| - |I|$ generators for the free stabilizers is given by Lemma B.10.

We can efficiently synthesise the isometry tableau and qubit preparations into a circuit C_{stab}. The final circuit is the combination of C_{stab} followed by $C_{|\overline{O}|}$ which implements the same operator as (Γ, α).

To examine the complexity, we suppose an explicit Pauli flow is not given up front. Theorem 4.5 says that we can identify it in $O(|V|^5)$ time. We can extend the flow to consider all of the input extensions

simultaneously using Lemma C.5 in $O(|I|)$ time (ignoring the update to $\prec$ since it won't affect the outcome of the extraction procedure). The construction of Lemma 4.6 focusses this Pauli flow in $O(|V|^3)$ time (we apply at most $O(|V|^2)$ updates, each of which takes $O(|V|)$ time to identify and apply). Adding the extra focussed sets also takes $O(|V|^3)$ time by Lemma B.10. Synthesising this into a circuit using the naive CX ladder construction takes $O(|V| \cdot |O|)$ time for the rotations from non-Pauli measurements, and $O(|O|^3)$ to synthesise a stabilizer tableau using Aaronson and Gottesman's method [1] (mapping an isometry tableau into that form just requires identifying the additional generators to fill the Pauli group, which can be done by Gaussian elimination in $O(|O|^3)$ time). The overall complexity is therefore dominated by the $O(|V|^5)$ Pauli flow identification. $\square$

Example C.13. Suppose we start with the following measurement pattern and focussed Pauli flow.

	v	$\lambda(v)$	$p(v)$	$\text{Odd}(p(v))$	$\{u \mid v \prec u\}$
	i	XY	b, o_2	i, a	a, b, c, o_1, o_2
	a	YZ	a, c, d, o_2	d, o_1, o_2	c, o_1, o_2
	b	XY	c, d, o_1	b, d, o_1, o_2	c, o_1, o_2
	c	XY	o_1	c	o_1
	d	Y	o_2	d	o_2

Because we have two outputs and only one input, there is also an extra focussed set $\hat{p} = \{c, o_2\}$. We start by constructing the primary extraction strings for each vertex from Lemma 4.4. Let $a_d \in \{0, 1\}$ be such that $\alpha(d) = a_d \pi$. Recall that we get one phase flip per edge between adjacent vertices in the correction/focussed set ($|E \cap (p(v) \times p(v))|$), one phase flip for every two Ys that appear in the graph state stabilizer ($|p(v) \cap \text{Odd}(p(v))|/2$), and one for each term in the graph state stabilizer that is absorbed from a Pauli measurement with angle π ($|(p(v) \cup \text{Odd}(p(v))) \cap \{w \mid \lambda(w) \in \{X, Y, Z\} \wedge \alpha(w) = \pi\}|$).

S	$S \cap O$	$\text{Odd}(S) \cap O$	Edges	Ys	Paulis	$\mathbf{P}^S$
$p(i)$	o_2		0	0		$I_1 X_2$
$p(a)$	o_2	o_1, o_2	4	2	d	$(-1)^{a_d+1} Z_1 Y_2$
$p(b)$	o_1	o_1, o_2	2	2	d	$(-1)^{a_d+1} Y_1 Z_2$
$p(c)$	o_1		0	0		$X_1 I_2$
$p(d)$	o_2		0	0		$I_1 X_2$
$\hat{p}$	o_2	o_1	0	0		$Z_1 X_2$

We start by extracting the planar angles as rotations in $\succ$-order. We must start with c, extracing it as $e^{i\frac{\alpha(c)}{2}X_1 I_2}$ over the outputs.

Having extracted this, the projection on c becomes the positive X basis vector so it can absorb the corresponding Paulis on the graph state stabilizer from $p(a)$ and $p(b)$. Either a or b can be extracted next, giving $e^{(-1)^{a_d}\frac{\alpha(a)}{2}Z_1 Y_2}$ and $e^{(-1)^{a_d+1}\frac{\alpha(b)}{2}Y_1 Z_2}$ respectively. Note that the phase of the rotation from a got an extra -1 since $D_a = 1$ ($\lambda(a) = YZ$).

The final rotation to be extracted is $e^{i\frac{\alpha(i)}{2}I_1 X_2}$ from i. We don't extract any rotation from d since the measurement is already in a Pauli basis, and hence we can treat it as part of the leftover stabilizer process.

We can get the isometry tableau for this from the primary extraction strings. Having both the input and its only neighbour labelled with XY means we don't necessarily have to perform input extensions to obtain the rows of the tableau during extraction, since they will function identically to i and b here because Γ can already be viewed as an input extension of $\Gamma \setminus \{i\}$ with input b. This process maps Z on

the input to I_1X_2 ($\mathbf{P}^{\perp i}$) over the outputs and X on the input to $(-1)^{a_d+1}Y_1Z_2$ ($\mathbf{P}^{\perp b}$). We have an extra row from the free stabilizer Z_1X_2 obtained from $\hat{p}$.

The structure we now have exactly maps to the following PDDAG:

Ins	Outs	Sign
X	Y Z	$(-)^{a_d+1}$
Z	X	+
	Z X	+

$(I_1X_2, \alpha(i)) \longrightarrow ((-1)^{a_d}Z_1Y_2, \alpha(a)) \longrightarrow (X_1I_2, \alpha(c))$

$((-1)^{a_d+1}Y_1Z_2, \alpha(b))$

Note that producing this PDDAG just amounted to reading off the focussed Pauli flow, using the correction sets to produce the strings and the order $\prec$ for the dependency relation between the rotations.

For the rotations from a and b, we put the (-1) phase with the strings rather than the angle. Whilst nodes $(-\mathbf{P}, \theta)$ and $(\mathbf{P}, -\theta)$ represent the same rotations, it is useful to keep the convention of writing the extracted rotation as $((-1)^{D_v}\mathbf{P}^{\perp v}, \alpha(v))$ to simplify the simulations in Section 5.

We can then synthesise a circuit from the PDDAG component-wise. We start with the isometry tableau, for which an example circuit is given below.
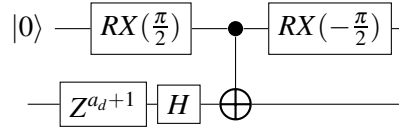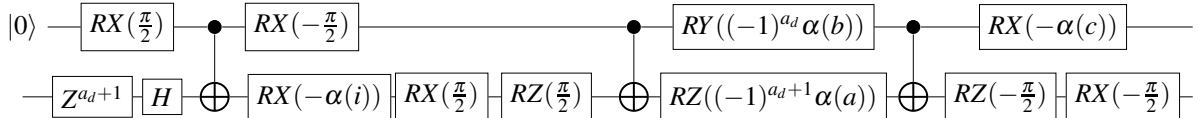

For the rotations, the $(I_1X_2, \alpha(i))$ and $(X_1I_2, \alpha(c))$ are simply RX gates. The other two rotations commute, and so can be efficiently diagonalised simultaneously.
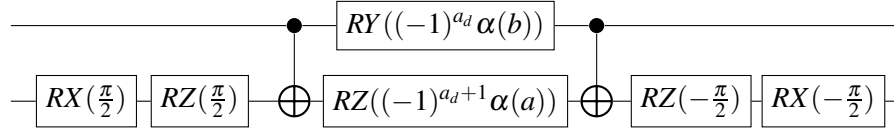
Note that the angles of rotations acquire an extra -1 phase flip, since we defined $(\mathbf{P}, \theta)$ to map to $e^{i\frac{\theta}{2}\mathbf{P}}$, but basic rotation gates are defined as $RP(\theta) = e^{-i\frac{\theta}{2}P}$.

Putting it all together, we obtain the following circuit:

D Proofs for Rewrite Correspondences

Throughout this section, we will follow a general pattern when deriving the correspondences between rewrites of measurement patterns and PDDAGs:

1. Prove that the existence of Pauli flows is preserved by giving explicit updates to the focussed Pauli flows (and focussed sets);

2. Give the explicit updates to the primary extraction strings (and free stabilizers) for each rewrite;

3. Justify that the same updates are performed by the PDDAG rewrite.

We will also add additional proofs that semantics are preserved by rewriting the measurement pattern as required. When deriving equalities in the following proofs, we will freely use $\approx$ to ignore global constants that are later removed symmetrically using the same substitution.

D.1 Relabelling Pauli Measurements

Lemma D.1. Let (Γ, α) describe a measurement pattern and $u \in \overline{O}$ be a vertex with $\lambda(u) \in \{XY, XZ, YZ\}$ and $\alpha(u) \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Then there exists an equivalent pattern (Γ', α') from relabelling u as a Pauli measurement, where the new labels and angles are given by

$$(\lambda'(u), \alpha'(u)) = \begin{cases} (X, \alpha(u)) & \text{if } \lambda(u) = XY \wedge \alpha(u) \in \{0, \pi\} \\ (Y, \alpha(u) - \frac{\pi}{2}) & \text{if } \lambda(u) = XY \wedge \alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\} \\ (Z, \alpha(u)) & \text{if } \lambda(u) = XZ \wedge \alpha(u) \in \{0, \pi\} \\ (X, \alpha(u) - \frac{\pi}{2}) & \text{if } \lambda(u) = XZ \wedge \alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\} \\ (Z, \alpha(u)) & \text{if } \lambda(u) = YZ \wedge \alpha(u) \in \{0, \pi\} \\ (Y, \alpha(u) - \frac{\pi}{2}) & \text{if } \lambda(u) = YZ \wedge \alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\} \end{cases} \tag{50}$$

and $(\lambda'(v), \alpha'(v)) = (\lambda(v), \alpha(v))$ for any $v \in \overline{O} \setminus \{u\}$.

Proof. The following equations describe how the Pauli measurements are embedded into the planes:

$$\begin{aligned} \langle +_{X,\alpha}| &\approx \langle +_{XY,\alpha}| \approx \langle +_{XZ,\alpha+\frac{\pi}{2}}| \\ \langle +_{Y,\alpha}| &\approx \langle +_{XY,\alpha+\frac{\pi}{2}}| \approx \langle +_{YZ,\alpha+\frac{\pi}{2}}| \\ \langle +_{Z,\alpha}| &\approx \langle +_{XZ,\alpha}| \approx \langle +_{YZ,\alpha}| \end{aligned} \tag{51}$$

The equivalent Pauli measurement label and angle are determined uniquely by these embeddings. $\quad\square$

Lemma D.2. Let (Γ, α) and (Γ', α') describe measurement patterns related by relabelling some planar $u \in \overline{O}$ as a Pauli according to Lemma D.1. If Γ has a focussed Pauli flow $(p, \prec)$, then $(p', \prec)$ is a focussed Pauli flow for Γ' where for any $v \in \overline{O}$

$$p'(v) := \begin{cases} p(v)\Delta p(u) & \text{if } v \neq u \wedge u \in p(v) \cup \text{Odd}(p(v)) \wedge \alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\} \\ p(v) & \text{otherwise} \end{cases} \tag{52}$$

Furthermore, focussed sets $\hat{p}$ can be updated as

$$\hat{p}' := \begin{cases} \hat{p}\Delta p(u) & \text{if } u \in \hat{p} \cup \text{Odd}(\hat{p}) \wedge \alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\} \\ \hat{p} & \text{otherwise} \end{cases} \tag{53}$$

Proof. Suppose that $(p, \prec)$ is a focussed Pauli flow for Γ. Since $\lambda(u) \in \{XY, XZ, YZ\}$, any $v \in \overline{O} \setminus \{u\}$ such that $u \in p(v) \cup \text{Odd}(p(v))$ must satisfy $v \prec u$ by conditions $[\prec .X]$ and $[\prec .Z]$. $(p', \prec)$ is then also a Pauli flow for Γ using Lemma B.1. Any Pauli flow for Γ is also a Pauli flow for Γ', since $[\prec .X]$-$[\prec .Y]$ are strictly weakened and $[\lambda .X]$-$[\lambda .Y]$ for Γ' follow from $[\lambda .XY]$-$[\lambda .YZ]$ for Γ.

To show that $(p', \prec)$ is focussed for Γ', consider and vertex $v \in \overline{O}$. As λ and λ' only differ on u, when $p'(v) = p(v)$ it remains focussed over $\overline{O} \setminus \{u, v\}$ in Γ'. This is enough when $v = u$. If $u \notin p(v) \cup \text{Odd}(p(v))$, then it is trivially focussed over $\{u\}$ in Γ'. Finally, in each of the cases of Equation 50 where $\alpha(u) \in \{0, \pi\}$, the compatible corrections in the focussed conditions do not change, so it remains focussed over $\{u\}$ in Γ'.

Suppose instead that $v \neq u$, $u \in p(v) \cup \mathrm{Odd}(p(v))$, and $\alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\}$, so $p'(v) = p(v)\Delta p(u)$. Both $p(v)$ and $p(u)$ remain focussed over $\overline{O} \setminus \{u, v\}$ in Γ', and Lemma B.2 gives the same for $p'(v)$. For each case of $\lambda(u) \in \{XY, XZ, YZ\}$, what was a compatible correction for $\lambda(u)$ is now incompatible with $\lambda'(u)$, so $p(v)$ is not focussed over $\{u\}$ in Γ'. Additionally, conditions $[\lambda.X]$ and $[\lambda.Y]$ mean that $p(u)$ is also not focussed over $\{u\}$. However, we can then use Lemma B.3 to conclude that their combination $(p'(v))$ is focussed.

The proof of updating a focussed set $\hat{p}$ to $\hat{p}'$ follows similarly. $\qquad\square$

Lemma D.3. Let (Γ, α) and (Γ', α') be the measurement patterns and respective focussed Pauli flows $(p, \prec)$ and $(p', \prec)$ from Lemma D.2 formed by changing the label of some $u \in \overline{O}$ from a planar measurement to a Pauli measurement. Then the new primary extraction string for each vertex $v \in \overline{O} \setminus \{u\}$ is given by

$$\mathbf{P}'^{\perp v} = P_v'^{\perp v} P_v^{\perp v} \mathbf{P}^{\perp v} \left(\cos \alpha(u) I - i \sin(-1)^{D_u + A_{u \to v}^p} \alpha(u) P_v^{u \to v} \mathbf{P}^{\perp u} \right)^{F_{v \to u}^p} \tag{54}$$

and $\mathbf{P}'^{\perp u} = \mathbf{P}^{\perp u}$. Furthermore, a stabilizer from a focussed set $\hat{p}$ is updated to

$$\hat{\mathbf{P}}' = \hat{\mathbf{P}} \left(\cos \alpha(u) I - i \sin(-1)^{D_u} \alpha(u) \mathbf{P}^{\perp u} \right)^{G_{\hat{p} \to u}} \tag{55}$$

Proof. The case for u is simple because $p'(u) = p(u)$ and the linear map $\mathscr{C}$ in the definition of extraction strings is unchanged. For the remainder, we consider some $v \in \overline{O} \setminus \{u\}$.

Firstly, we show that $P'^{\perp v} P^{\perp v} \approx I$ if $\alpha(u) \in \{0, \pi\}$ or $(P^{u \to v})^{F_{v \to u}^p}$ if $\alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\}$ to deduce that $\mathbf{P}'^{\perp v}$ only acts on the outputs. If either $u \notin p(v) \cup \mathrm{Odd}(p(v))$ (i.e. $F_{v \to u}^p = 0$) or $\alpha(u) \in \{0, \pi\}$, then $p'(v) = p(v)$ and hence $P'^{\perp v} = P^{\perp v}$. Otherwise, we have $p'(v) = p(v)\Delta p(u)$. By following a similar logic to the proof of Lemma C.7 considering $\{v\}$ instead of O, we end up with $P'^{\perp v} \approx P^{\perp v} P^{u \to v}$.

Now we show that $\mathbf{P}'^{\perp v}$ is a primary extraction string in (Γ', α'). Let $\mathscr{C}'$ be the linear map from Definition 4.2 for (Γ', α') and let $\mathscr{C}$ be the corresponding for (Γ, α) (plus an extra projection on u if $u \preceq v$).

$$\mathscr{C}' := \left(\prod_{\substack{w \succ v \\ \lambda'(w) \notin \{X,Y,Z\}}} \langle +_{\lambda'(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda'(w) \in \{X,Y,Z\}}} \langle +_{\lambda'(w),\alpha'(w)}|_w \right) E_G N_{\overline{I}}$$

$$\approx \langle +_{\lambda(u),\alpha(u)}|_u \left(\prod_{\substack{w \succ v \\ w \neq u \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right) E_G N_{\overline{I}}$$

$$\approx \left(\prod_{\substack{w \succ v \vee w = u \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right) e^{(-1)^{D_u} i \frac{\alpha(u)}{2} P_u^{\perp u}} E_G N_{\overline{I}}$$

$$= e^{(-1)^{D_u + A_{u \to v}^p} i \frac{\alpha(u)}{2} P_v^{u \to v} \mathbf{P}^{\perp u}} \mathscr{C} \tag{56}$$

$$= e^{(-1)^{D_u + A_{u \to v}^p} i \frac{\alpha(u)}{2} P_v^{u \to v} \mathbf{P}^{\perp u}} P_v^{\perp v} \mathbf{P}^{\perp v} \mathscr{C}$$

$$= P_v^{\perp v} e^{(-1)^{D_u + A_{u \to v}^P + F_{u \to v}^P} i \frac{\alpha(u)}{2} P_v^{u \to v} \mathbf{P}^{\perp u}} \mathbf{P}^{\perp v} \mathscr{C}$$

$$= P_v^{\perp v} \mathbf{P}^{\perp v} e^{(-1)^{D_u + A_{u \to v}^P + F_{v \to u}^P} i \frac{\alpha(u)}{2} P_v^{u \to v} \mathbf{P}^{\perp u}} \mathscr{C}$$

$$= P_v^{\perp v} \mathbf{P}^{\perp v} e^{-(-1)^{D_u + A_{u \to v}^P} F_{v \to u}^P i \alpha(u) P_v^{u \to v} \mathbf{P}^{\perp u}} e^{(-1)^{D_u + A_{u \to v}^P} i \frac{\alpha(u)}{2} P_v^{u \to v} \mathbf{P}^{\perp u}} \mathscr{C}$$

$$\approx P_v^{\perp v} \mathbf{P}^{\perp v} e^{-(-1)^{D_u + A_{u \to v}^P} F_{v \to u}^P i \alpha(u) P_v^{u \to v} \mathbf{P}^{\perp u}} \mathscr{C}'$$

$$= P_v^{\perp v} \mathbf{P}^{\perp v} \left(\cos \alpha(u) I - i \sin(-1)^{D_u + A_{u \to v}^P} \alpha(u) P_v^{u \to v} \mathbf{P}^{\perp u} \right)^{F_{v \to u}^P} \mathscr{C}'$$

$$= P_v'^{\perp v} \mathbf{P}'^{\perp v} \mathscr{C}'$$

In the above derivation, moving $P_v^{\perp v} \mathbf{P}^{\perp v}$ through the rotation requires Lemma C.9 and $P^{\perp v} P^{u \to v} = (-1)^{F_{u \to v}^P} P^{u \to v} P^{\perp v}$.

Finally, by doing a case distinction on $F_{v \to u}^P$ and Lemma C.7, the extraction string from $(p', \prec)$ matches $\mathbf{P}'^{\perp v}$ up to phase, with Lemma C.8 making them equal.

The proof for focussed sets and stabilizers follows similarly. $\qquad \square$

Theorem D.4. (Restatement of Theorem 5.1) Let (Γ, α) describe a measurement pattern with some vertex $u \in \overline{O}$ such that $\lambda(u) \in \{XY, XZ, YZ\}$ and $\alpha(u) \in \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. Relabelling u to the equivalent Pauli label corresponds to pushing the rotation from u to the start of the PDDAG and absorbing it into the initial stabilizer process.

Proof. The corresponding PDDAG initially has a rotation node $((-1)^{D_u} \mathbf{P}^{\perp u}, \alpha(u))$ (representing the operator $e^{(-1)^{D_u} i \frac{\alpha(u)}{2} \mathbf{P}^{\perp u}}$) since u originally has a planar label. Moving this to the front of the circuit updates the Pauli strings of other vertices according to the Reorder Rules, so we need to check that the same update is made to planar vertices and the isometry tableau is updated appropriately.

For any planar vertex $v \preceq u$, $v \notin p(u) \cup \mathrm{Odd}(p(u))$ and hence $P^{u \to v} = I$ and $A_{u \to v}^P = 0$. If $u \notin p(v) \cup \mathrm{Odd}(p(v))$ then $\mathbf{P}^{\perp v}$ and $\mathbf{P}^{\perp u}$ commute, so no update is needed which matches the formula for $\mathbf{P}'^{\perp v}$ with $F_{v \to u}^P = 0$. Otherwise, they anticommute. If $\alpha(u) = 0$, the rotation is an identity so $\mathbf{P}^{\perp v}$ is unchanged. If $\alpha(u) = \pi$, the rotation is a Pauli operator and hence flips the phase of v's rotation, matching the $\cos \alpha(u)$ update. For $\alpha(u) \in \{\frac{\pi}{2}, \frac{3\pi}{2}\}$, the extraction string is updated by multiplication with the correct phase.

For the tableau, we need to handle rows from both the inputs and free stabilizers. For an input i gained by input extension with $\lambda(i) = XY$, the same reductions can be made as for planar vertices and the updates for free stabilizers are essentially identical. In each case of $\alpha(u)$, these updates match the effect on the isometry tableau since the Reorder Rules show the same effect for Paulis and generic rotations. $\qquad \square$

Example D.5. Let's start with the same measurement pattern from Example C.13. Suppose $\alpha(c) = \frac{\pi}{2}$, so we can obtain a semantically equivalent pattern by relabelling $(\lambda'(c), \alpha'(c)) := (Y, 0)$. This would make $p(a)$, $p(b)$ and $\hat{p}$ no longer focussed. Refocussing gives:

v	$\lambda'(v)$	$p'(v)$	$\mathrm{Odd}(p'(v))$	$\{u \mid v \prec' u\}$
i	XY	b, o_2	i, a	a, b, o_1, o_2
a	YZ	a, c, d, o_1, o_2	c, d, o_1, o_2	o_1, o_2
b	XY	c, d	b, c, d, o_1, o_2	o_1, o_2
c	Y	o_1	c	o_1
d	Y	o_2	d	o_2

and $\hat{p}' = \{c, o_1, o_2\}$. The restriction of $\prec$ to $\prec'$ is not necessary, but will help to see the resemblance to the PDDAG later.

Now let's look at applying the corresponding rewrite on the PDDAG from C.13. That is, we use the Reorder Rules to move $(X_1 I_2, \frac{\pi}{2})$ (i.e. $e^{i\frac{\pi}{4}X_1 I_2}$) into the tableau. $X_1 I_2$ anticommutes with the rotations from a and b, the tableau row from b, and the free stabilizer from $\hat{p}$. Each of these strings will then be updated by multiplication by $X_1 I_2$. These are exactly the correction/focussed sets that were updated in the measurement pattern by combining with $p(c)$. The result of this rewrite on the PDDAG is:

Ins	Outs	Sign
X	Z Z	$(-)^{a_d}$
Z	X	$+$
	Y X	$+$

$(I_1 X_2, \alpha(i)) \longrightarrow \begin{array}{l} ((-1)^{a_d+1} Y_1 Y_2, \alpha(a)) \\ ((-1)^{a_d} Z_1 Z_2, \alpha(b)) \end{array}$

One can check that this is identical to the PDDAG extracted using $(p', \prec)$.

D.2 Z Measurement Elimination

Lemma D.6. Let (Γ, α) describe a measurement pattern with some vertex $u \in \overline{O}$ such that $\lambda(u) \in \{XZ, YZ, Z\}$ and $\alpha(u) = a\pi$ ($a \in \{0, 1\}$). Then eliminating vertex u gives an equivalent measurement pattern (Γ', α') where $\Gamma' = (G \setminus \{u\}, I, O, \lambda)$, for all $w \in \overline{O} \setminus \{u\}$

$$\alpha'(w) := \begin{cases} \alpha(w) + a\pi & \text{if } w \in N_G(u) \wedge \lambda(w) \in \{XY, X, Y\} \\ (-1)^a \alpha(w) & \text{if } w \in N_G(u) \wedge \lambda(w) \in \{XZ, YZ\} \\ \alpha(w) & \text{otherwise} \end{cases} \tag{57}$$

and each output neighbouring u is followed by a Z^a gate.

Proof. This is just an extension of the planar case given in the proof of [5, Lemma 4.7] to cover Pauli labels by considering the embeddings from Equation 51. The idea is that the Z-basis projections $\langle 0|$ and $\langle 1|$ reduce each of the CZ gates on that qubit to either an identity or a Z gate (modelled as a Z conditioned on the measurement angle), in the same way that they reduce when applied to a $|0\rangle$ or $|1\rangle$. This Z commutes with other CZ gates to the neighbouring outputs or to change the effective measurement on the neighbour. The qubit u can then be removed since the only operations on it are the preparation and measurement. ∎

Lemma D.7. (Generalisation of [5, Lemma 3.4]) Let (Γ, α) and (Γ', α') be measurement patterns related by elimination of a vertex u as in Lemma D.6. If $(p, \prec)$ is a Pauli flow for Γ, then $(p', \prec)$ is a Pauli flow for Γ' where for any $v \in \overline{O} \setminus \{u\}$

$$p'(v) := \begin{cases} p(v) \Delta p(u) & \text{if } u \in p(v) \\ p(v) & \text{if } u \notin p(v) \end{cases} \tag{58}$$

If $(p, \prec)$ is focussed, then so is $(p', \prec)$. Furthermore, if $\hat{p}$ is a focussed set for Γ, then it is also a focussed set for Γ'.

Proof. We first need to show that p' is a function from $\overline{O} \setminus \{u\}$ to $\mathscr{P}[\overline{I} \setminus \{u\}]$. This follows since $\lambda(u) \in \{XZ, YZ, Z\}$ implies $u \in p(u)$ (by conditions [$\lambda.XZ$], [$\lambda.YZ$], and [$\lambda.Z$]), so both cases of the definition of $p'(v)$ give $u \notin p'(v)$.

Because $\lambda(u) \in \{XZ, YZ, Z\}$, if $u \in p(v)$ then $v \prec u$ from $[\prec .X]$. This means $(p', \prec)$ is a valid Pauli flow for (G, I, O, λ) by repeated applications of Lemma B.1. In particular, this means it satisfies all of the Pauli flow conditions for any $v \in \overline{O} \setminus \{u\}$. These aren't invalidated when restricting to $G \setminus \{u\}$ since $\text{Odd}_{G \setminus \{u\}}(A) = \text{Odd}_G(A) \setminus \{u\}$ for any $A \subseteq V \setminus \{u\}$ and $\forall v \in \overline{O} \setminus \{u\}. p'(v) \subseteq V \setminus \{u\}$.

If $(p, \prec)$ is a focussed Pauli flow, then $u \notin p(v)$ for any $v \neq u$ since $\lambda(u) \in \{XZ, YZ, Z\}$. This means $p'(v) = p(v)$ for every v, so the focussed property is trivially preserved. Similarly, the same holds for focussed sets. $\qquad \square$

Lemma D.8. Let (Γ, α) and (Γ', α') be the measurement patterns and respective Pauli flows $(p, \prec)$ and $(p', \prec)$ from Lemma D.7 formed by the elimination of a vertex u. Then the new primary extraction string for each vertex $v \in \overline{O} \setminus \{u\}$ is given by

$$\mathbf{P}'^{\perp v} = (-1)^{a|p(v) \cap N_G(u) \cap (O \cup \{v\})|} \left(\prod_{\substack{n \in (N_G(u) \cup \{u\}) \cap \overline{O} \\ n \succ v \\ n = u \Rightarrow \lambda(n) \in \{XZ, YZ\} \\ n \neq u \Rightarrow \lambda(n) = XY}} (-1)^{F^P_{v \to n}} \right)^a \mathbf{P}^{\perp v} \tag{59}$$

where $\alpha(u) = a\pi$. Furthermore, the stabilizer from a focussed set $\hat{p}$ is updated similarly.

Proof. The focussed property implies $u \notin p(v)$ (resp. $u \notin \hat{p}$), so $p'(v) = p(v)$ (resp. $\hat{p}$ is still a focussed set). Then $p'(v) \cup \text{Odd}_{G \setminus u}(p'(v)) = (p(v) \cup \text{Odd}_G(p(v))) \setminus \{u\}$ (resp. $\hat{p} \cup \text{Odd}_{G \setminus u}(\hat{p}) = (\hat{p} \cup \text{Odd}_G(\hat{p})) \setminus \{u\}$). This is enough to show that the strings are preserved up to phase.

We find a primary extraction string for v in (Γ', α') by considering the linear map $\mathscr{C}'$ from Definition 4.2, using the respective linear map $\mathscr{C}$ for (Γ, α).

$$\mathscr{C}' = \left(\prod_{\substack{w \succ v \\ w \neq u \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{u\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha'(w)}|_w \right) E_{G \setminus u} N_{\overline{I} \setminus \{u\}} \tag{60}$$

We can reintroduce qubit u with measurement angle 0. We can then map the angles for the Pauli projections from α' to α using orthogonal Paulis, letting $z_u := 1$ if $\lambda(u) = Z$ and 0 if $\lambda(u) \in \{XZ, YZ\}$.

$$\mathscr{C}' \approx \sqrt{2} \left(\prod_{\substack{w \succ v \\ w \neq u \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{u\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha'(w)}|_w \right) \langle +_{\lambda(u),0}|_u E_G N_{\overline{I}}$$

$$\approx \sqrt{2} \left(\prod_{\substack{w \succ v \vee w = u \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right) \tag{61}$$

$$(P_u^{\perp u})^{z_u a} \left(\prod_{\substack{n \in N_G(u) \\ \lambda(n) \in \{X,Y\}}} P_n^{\perp n} \right)^a E_G N_{\overline{I}}$$

We can then apply the stabilizer of the graph state centred around u to move these Paulis to other qubits, some of which can be absorbed into projections again.

$$
\mathscr{C}' \approx \sqrt{2} \left(\prod_{\substack{w \succ v \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w),0}|_w \right) \left(\prod_{\substack{w \in \overline{O} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w \right)
$$

$$
(P_u^{\perp u})^{(1-z_u)a} \left(\prod_{\substack{n \in N_G(u) \cap (O \cup \{v\})}} Z_n \right)^a \left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \succ v \\ \lambda(n)=XY}} P_n^{\perp n} \right)^a \left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \preceq v \\ \lambda(n) \notin \{X,Y,Z\}}} Z_n \right)^a E_G N_{\overline{I}}
$$

(62)

Each of the Paulis trapped under projections can be replaced by the primary extraction strings of those qubits. We leave the extraction strings without any order since reordering only introduces phases at this point which will be later removed.

$$
\mathscr{C}' \approx \sqrt{2} \left(\prod_{\substack{w=u \\ w \succ v}} P_v^{u \to v} \mathbf{P}^{\perp u} \prod_{\substack{w=u \\ w \preceq v}} P_u^{\perp u} \right)^{(1-z_u)a} \left(\prod_{\substack{n \in N_G(u) \cap (O \cup \{v\})}} Z_n \right)^a
$$

$$
\left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \succ v \\ \lambda(n)=XY}} P_v^{n \to v} \mathbf{P}^{\perp n} \right)^a \left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \preceq v \\ \lambda(n) \notin \{X,Y,Z\}}} Z_n \right)^a \mathscr{C}
$$

(63)

We can now introduce $P_v^{\perp v} \mathbf{P}^{\perp v}$. Commuting this to the front induces phase changes according to Lemma C.9 and $P^{\perp v} P^{u \to v} = (-1)^{F_{u \to v}^P} P^{u \to v} P^{\perp v}$. We then undo the previous steps to get back to $\mathscr{C}'$.

$$
\mathscr{C}' \approx \sqrt{2} \left(\prod_{\substack{w=u \\ w \succ v}} P_v^{u \to v} \mathbf{P}^{\perp u} \prod_{\substack{w=u \\ w \preceq v}} P_u^{\perp u} \right)^{(1-z_u)a} \left(\prod_{\substack{n \in N_G(u) \cap (O \cup \{v\})}} Z_n \right)^a
$$

$$
\left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \succ v \\ \lambda(n)=XY}} P_v^{n \to v} \mathbf{P}^{\perp n} \right)^a \left(\prod_{\substack{n \in N_G(u) \cap \overline{O} \\ n \preceq v \\ \lambda(n) \notin \{X,Y,Z\}}} Z_n \right)^a P_v^{\perp v} \mathbf{P}^{\perp v} \mathscr{C}
$$

$$
= \sqrt{2} P_v^{\perp v} \left(\prod_{\substack{w=u \\ w \succ v}} (-1)^{F_{u \to v}^P} P_v^{u \to v} \mathbf{P}^{\perp u} \prod_{\substack{w=u \\ w \preceq v}} P_u^{\perp u} \right)^{(1-z_u)a}
$$

(64)

$$
(-1)^{a|p(v) \cap N_G(u) \cap \{v\}|} \left(\prod_{\substack{n \in N_G(u) \cap (O \cup \{v\})}} Z_n \right)^a
$$

$$\left(\prod_{\substack{n\in N_G(u)\cap\overline{O}\\ n\succ v\\ \lambda(n)=XY}} (-1)^{F^p_{n\to v}} P^{n\to v}_v \mathbf{P}^{\perp n} \right)^a \left(\prod_{\substack{n\in N_G(u)\cap\overline{O}\\ n\preceq v\\ \lambda(n)\notin\{X,Y,Z\}}} Z_n \right)^a \mathbf{P}^{\perp v}\mathscr{C}$$

$$=\sqrt{2} P^{\perp v}_v \mathbf{P}^{\perp v} \left(\prod_{\substack{w=u\\ w\succ v}} (-1)^{F^p_{v\to u}} P^{u\to v}_v \mathbf{P}^{\perp u} \prod_{\substack{w=u\\ w\preceq v}} P^{\perp u}_u \right)^{(1-z_u)a}$$

$$(-1)^{a|p(v)\cap N_G(u)\cap(O\cup\{v\})|} \left(\prod_{n\in N_G(u)\cap(O\cup\{v\})} Z_n \right)^a$$

$$\left(\prod_{\substack{n\in N_G(u)\cap\overline{O}\\ n\succ v\\ \lambda(n)=XY}} (-1)^{F^p_{v\to n}} P^{n\to v}_v \mathbf{P}^{\perp n} \right)^a \left(\prod_{\substack{n\in N_G(u)\cap\overline{O}\\ n\preceq v\\ \lambda(n)\notin\{X,Y,Z\}}} Z_n \right)^a \mathscr{C}$$

$$\approx P^{\perp v}_v \mathbf{P}^{\perp v}(-1)^{a|p(v)\cap N_G(u)\cap(O\cup\{v\})|} \left(\prod_{\substack{n\in(N_G(u)\cup\{u\})\cap\overline{O}\\ n\succ v\\ n=u\Rightarrow\lambda(n)\in\{XZ,YZ\}\\ n\neq u\Rightarrow\lambda(n)=XY}} (-1)^{F^p_{v\to n}} \right)^a \mathscr{C}'$$

The calculations for focussed sets follow similarly. We can finish by using Lemma C.8 to conclude that these primary extraction strings (resp. stabilizers) are the ones generated by the new Pauli flow (resp. focussed sets). $\square$

Theorem D.9. (Restatement of Theorem 5.2) Let (Γ,α) describe a measurement pattern with some vertex $u\in\overline{O}$ such that $\lambda(u)\in\{XZ,YZ,Z\}$ and $\alpha(u)\in\{0,\pi\}$. Eliminating u from the graph corresponds to the following sequence of actions on the PDDAG:

1. If u has a planar (XZ or YZ) label, then its rotation is pulled from the rotation DAG into the stabilizer block;

2. For each neighbour n of u that is an output, a Z_n rotation of $\alpha(u)$ is pulled from the stabilizer block through the entire rotation DAG to the end of the circuit;

3. For each neighbour n of u with $\lambda(n)=XY$, a $\mathbf{P}^{\perp n}$ rotation of $\alpha(u)$ is pulled from the stabilizer block and merged with the existing rotation for n.

Proof. Each action is moving Pauli operators through rotations. The Reorder Rules simplify to inducing phase flips. In the PDDAG, we can flip either the operator or angle of the rotation. The $\alpha(u)=0$ case is trivial, so we suppose that $\alpha(u)=\pi$ ($a=1$).

For both the rotations and tableau rows, we need only consider extraction strings of planar vertices. Each of the rotations moved through the PDDAG to/from location w (either $w=u$ or $w\in N_G(u)$) would induce a phase shift in the rotation or tableau row for v if the corresponding Pauli string $\mathbf{P}^{\perp w}$ anticommutes with $\mathbf{P}^{\perp v}$ and originated beyond v (either w is an output or $v\preceq w$), i.e. $w\in p(v)\cap O$ or $F^p_{v\to w}=1$.

Equation 59 also negates the angle on v if $v \in N_G(u)$ and $\lambda(v) \in \{XZ, YZ\}$ (i.e. $v \in p(v)$), as covered in Lemma D.6. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad\Box$

Example D.10. We will use the same example measurement pattern as Example C.13. a is the only vertex with a label ready for vertex elimination, so to see something actually interesting we suppose $\alpha(a) = \pi$. Eliminating a gives the following residual pattern.

v	$\lambda'(v)$	$\alpha'(v)$	$p'(v)$	$\mathrm{Odd}(p'(v))$	$\{u\|v \prec u\}$
i	XY	$\alpha(i)$	b, o_2	i	b, c, o_1, o_2
b	XY	$\alpha(b) + \pi$	c, d, o_1	b, d, o_1, o_2	c, o_1, o_2
c	XY	$\alpha(c) + \pi$	o_1	c	o_1
d	Y	$\alpha(d) + \pi$	o_2	d	o_2

The focussed set $\hat{p} = \{c, o_2\}$ remains valid. Following the rewrite sequence in the PDDAG, step 1 sees the rotation from a pulled into the tableau. The $Z_1 Y_2$ term anticommutes with the rotation and tableau row from i, as well as the free stabilizer from $\hat{p}$.

Ins	Outs	Sign
X $\;Y\;Z$		$(-)^{a_d + 1}$
Z	X	$-$
	$Z\;X$	$-$

$$(-I_1 X_2, \alpha(i)) \longrightarrow ((-1)^{a_d+1} Y_1 Z_2, \alpha(b)) \longrightarrow (X_1 I_2, \alpha(c))$$

For step 2, none of the neighbours of a are outputs, so there is no change. For step 3, only b and c are labelled XY, so we move $Y_1 Z_2$ and $X_1 I_2$ from the tableau and merge into their corresponding rotations.

Ins	Outs	Sign
X $\;Y\;Z$		$(-)^{a_d}$
Z	X	$+$
	$Z\;X$	$+$

$$(I_1 X_2, \alpha'(i)) \longrightarrow ((-1)^{a_d} Y_1 Z_2, \alpha'(b)) \longrightarrow (X_1 I_2, \alpha'(c))$$

This same PDDAG is obtained by extracting from the reduced measurement pattern above. Since extraction will give everything in terms of α', we note that the phase on the primary extraction string from b is now $(-1)^{1+\alpha'(d)/\pi} = (-1)^{a_d}$.

D.3   Local Complementation of Graphs

Definition D.11. Given a graph $G = (V, E)$ with some designated vertex $u \in V$, the local complementation of G about u is the operation resulting in the graph:

$$G \star u := (V, E\Delta\{(v, w) | (v, u), (w, u) \in E \wedge v \neq w\}) \tag{65}$$

Lemma D.12. Let (Γ, α) describe a measurement pattern with $\Gamma = (G, I, O, \lambda)$ and some vertex $u \in \bar{I}$. Then performing a local complementation about u gives an equivalent measurement pattern (Γ', α') where $\Gamma' = (G \star u, I, O, \lambda')$, for all $v \in \overline{O} \setminus \{u\}$

$$(\lambda'(u), \alpha'(u)) := \begin{cases} (XZ, \alpha(u) + \frac{\pi}{2}) & \text{if } \lambda(u) = XY \\ (XY, \frac{\pi}{2} - \alpha(u)) & \text{if } \lambda(u) = XZ \\ (YZ, \alpha(u) + \frac{\pi}{2}) & \text{if } \lambda(u) = YZ \\ (X, \alpha(u)) & \text{if } \lambda(u) = X \\ (Z, \alpha(u) + \pi) & \text{if } \lambda(u) = Y \\ (Y, \alpha(u)) & \text{if } \lambda(u) = Z \end{cases} \tag{66}$$

$$(\lambda'(v), \alpha'(v)) := \begin{cases} (XY, \alpha(v) + \frac{\pi}{2}) & \text{if } v \in N_G(u) \wedge \lambda(v) = XY \\ (YZ, \alpha(v)) & \text{if } v \in N_G(u) \wedge \lambda(v) = XZ \\ (XZ, -\alpha(v)) & \text{if } v \in N_G(u) \wedge \lambda(v) = YZ \\ (Y, \alpha(v)) & \text{if } v \in N_G(u) \wedge \lambda(v) = X \\ (X, \alpha(v) + \pi) & \text{if } v \in N_G(u) \wedge \lambda(v) = Y \\ (\lambda(v), \alpha(v)) & \text{otherwise} \end{cases} \tag{67}$$

plus each output neighbouring u is followed by an $RZ(-\frac{\pi}{2})$ gate and if u is an output then it is followed by an $RX(\frac{\pi}{2})$ gate.

Proof. The case for planar measurements is given in the proof of [5, Lemma 4.3]. We can extend this to the Pauli measurements using the embeddings in Equation 51. For completeness, the rough idea is as follows:

In our notation, Van den Nest's Theorem for local complementation [33] (specifically, the version allowing for inputs as in [5, Lemma 4.3]) becomes:

$$E_{G \star u} N_{\bar{I}} \approx e^{-i\frac{\pi}{4}X_u} \left(\prod_{w \in N_G(u)} e^{i\frac{\pi}{4}Z_w} \right) E_G N_{\bar{I}} \approx e^{i\frac{\pi}{4}X_u} \left(\prod_{w \in N_G(u)} e^{-i\frac{\pi}{4}Z_w} \right) E_G N_{\bar{I}} \tag{68}$$

Each of these rotations then modifies any subsequent measurement on that qubit. By considering each case for $\lambda(v)$, we find the updates for λ' and α' are suitable to capture this rotation. $\square$

We quote the following Lemma from Duncan et al. [15] as it is useful in showing the preservation of Pauli flow here.

Lemma D.13. ([15, Lemma B.4]) Given a graph $G = (V, E)$, a subset $A \subseteq V$ and a vertex $u \in V$, we have

$$\text{Odd}_{G \star u}(A) = \begin{cases} \text{Odd}_G(A) \Delta (N_G(u) \cap A) & \text{if } u \notin \text{Odd}_G(A) \\ \text{Odd}_G(A) \Delta (N_G(u) \setminus A) & \text{if } u \in \text{Odd}_G(A) \end{cases} \tag{69}$$

Lemma D.14. Let (Γ, α) and (Γ', α') be measurement patterns related by local complementation about a vertex $u \in \bar{I}$ as in Lemma D.12. If $(p, \prec)$ is a Pauli flow for Γ, then $(p', \prec)$ is a Pauli flow for Γ' where for any $v \in \bar{O}$

$$p'(v) := \begin{cases} p(v) \Delta \{u\} & \text{if } u \in \text{Odd}_G(p(v)) \\ p(v) & \text{if } u \notin \text{Odd}_G(p(v)) \end{cases} \tag{70}$$

Proof. For this proof, we will suppose $u \in \bar{O}$. The case where u is an output has strictly weaker requirements from the flow conditions and hence will be covered by the cases examined here.

Firstly, we prove a useful decomposition of the odd neighbourhoods of correction sets. If $u \in \text{Odd}_G(p(v))$, then using Lemma D.13 we have

$$\begin{aligned} \text{Odd}_{G \star u}(p'(v)) &= \text{Odd}_{G \star u}(p(v)) \Delta \text{Odd}_{G \star u}(\{u\}) \\ &= \text{Odd}_G(p(v)) \Delta (N_G(u) \setminus p(v)) \Delta N_G(u) \\ &= \text{Odd}_G(p(v)) \Delta (N_G(u) \cap p(v)) \end{aligned} \tag{71}$$

and similarly if $u \notin \mathrm{Odd}_G(p(v))$ we have

$$\begin{aligned}
\mathrm{Odd}_{G \star u}(p'(v)) &= \mathrm{Odd}_{G \star u}(p(v)) \\
&= \mathrm{Odd}_G(p(v)) \Delta \left(N_G(u) \cap p(v) \right)
\end{aligned} \tag{72}$$

In either case, we can now use the following equation for the remainder of the proof:

$$\mathrm{Odd}_{G \star u}(p'(v)) = \mathrm{Odd}_G(p(v)) \Delta \left(N_G(u) \cap p(v) \right) \tag{73}$$

We will first prove the Pauli flow conditions for $v = u$ and then separately show that they all hold for any $v \in \overline{O} \setminus \{u\}$. For u, $[\prec .X]$ is preserved since $p'(u)$ and $p(u)$ are either identical or differ by u, and $\forall w \in \overline{O} \setminus \{u\}. \lambda'(w) \in \{X,Y\} \Leftrightarrow \lambda(w) \in \{X,Y\}$.

For condition $[\prec .Z]$, consider any $w \in \mathrm{Odd}_{G \star u}(p'(u))$ such that $w \neq u$ and $w \preceq u$. By using Equation 73, we can consider the following cases:

- Suppose $w \in \mathrm{Odd}_G(p(u))$ and $w \notin N_G(u) \cap p(u)$. From condition $[\prec .Z]$ for $(p, \prec)$, we have that $\lambda(w) \in \{Y,Z\}$. If $\lambda(w) = Z$ then $\lambda'(w) = Z$ which is sufficient, so suppose that $\lambda(w) = Y$. Condition $[\prec .Y]$ for $(p, \prec)$ gives $w \in p(u)$, and hence $w \notin N_G(u)$ which implies $\lambda'(w) = Y$.

- Suppose $w \notin \mathrm{Odd}_G(p(u))$ and $w \in N_G(u) \cap p(u)$. It must be the case that $\lambda(w) \neq Y$ by condition $[\prec .Y]$ for $(p, \prec)$, and condition $[\prec .X]$ gives $\lambda(w) \in \{X,Y\}$. So $\lambda(w) = X$ and $\lambda'(w) = Y$ which is sufficient.

For condition $[\prec .Y]$, consider any $w \preceq u$ such that $u \neq w$ and $\lambda'(w) = Y$. $w \in p'(u)$ if and only if $w \in p(u)$ from the definition of p'.

- Suppose $w \in N_G(u)$. The definition of λ' gives that $\lambda(w) = X$, therefore $w \notin \mathrm{Odd}_G(p(u))$ by condition $[\prec .Z]$ for $(p, \prec)$. Equation 73 now reduces to give $w \in \mathrm{Odd}_{G \star u}(p'(u)) \Leftrightarrow w \in p(u)$ $(w \in p'(u))$.

- Suppose $w \notin N_G(u)$. We similarly find $\lambda'(w) = Y$, and then $w \in \mathrm{Odd}_G(p(u)) \Leftrightarrow w \in p(u)$ from condition $[\prec .Y]$ for $(p, \prec)$. Equation 73 reduces to give $w \in \mathrm{Odd}_{G \star u}(p'(u)) \Leftrightarrow w \in \mathrm{Odd}_G(p(u))$. Therefore, we have $w \in \mathrm{Odd}_{G \star u}(p'(u)) \Leftrightarrow w \in p'(u)$ as required.

Suppose $u \in \mathrm{Odd}_G(p(u))$. Equation 73 implies $u \in \mathrm{Odd}_{G \star u}(p'(u))$ because $u \notin N_G(u)$. We also have $u \in p'(u)$ if and only if $u \notin p(u)$ from the definition of p'. Each of the conditions $[\lambda.XY]$-$[\lambda.Y]$ then follow straightforwardly since the flipping of set membership matches the changes to labels between λ and λ'.

Suppose instead that $u \notin \mathrm{Odd}_G(p(u))$. Equation 73 now implies $u \notin \mathrm{Odd}_{G \star u}(p'(u))$. Since u must be corrected in some way, $u \in p(u)$ must hold, and hence $u \in p'(u)$. The only measurement bases that correspond to this correction give $\lambda(u) \in \{YZ,Y,Z\}$. This also gives $\lambda'(u) \in \{YZ,Y,Z\}$, so the conditions $[\lambda.XY]$-$[\lambda.Y]$ still hold.

Now we need to show that the same conditions hold for any $v \in \overline{O} \setminus \{u\}$. For $[\prec .X]$-$[\prec .Y]$, we will do this by case distinction over the cases of $p'(v)$.

Suppose that $u \in \mathrm{Odd}_G(p(v))$. Note that $u \preceq v$ implies $\lambda(u) = Z \Leftrightarrow u \notin p(v)$ and $\lambda(u) = Y \Leftrightarrow u \in p(v)$ from conditions $[\prec .X]$-$[\prec .Y]$ for $(p, \prec)$.

$[\prec .X]$: Consider some $w \in p'(v) = p(v) \Delta \{u\}$ with $w \neq v$ and $w \preceq v$.

- If $w = u$, then $w \notin p(v)$ and $u \preceq v$, so $\lambda(u) = Z$ and $\lambda'(u) = Y$.

- If $w \in p(v)$ and $w \neq u$, then $[\prec .X]$ gives $\lambda(w) \in \{X,Y\}$, so $\lambda'(w) \in \{X,Y\}$.

$[\prec .Z]$: Consider some $w \in \mathrm{Odd}_{G\star u}(p'(v)) = \mathrm{Odd}_G(p(v))\Delta(N_G(u) \cap p(v))$ with $w \neq v$ and $w \preceq v$.

- If $w = u$ then $u \preceq v$, so $\lambda(w) \in \{Y,Z\}$ and $\lambda'(w) \in \{Y,Z\}$.

- If $w \in \mathrm{Odd}_G(p(v))$, $w \neq u$, and $w \notin N_G(u) \cap p(v)$ then by $[\prec .Z]$ we have $\lambda(w) \in \{Y,Z\}$. If $\lambda(w) = Z$ then $\lambda'(w) = Z$. If $\lambda(w) = Y$, then $w \in p(v)$ by $[\prec .Y]$, so $w \notin N_G(u)$ by Equation 73 and therefore $\lambda'(w) = Y$.

- If $w \notin \mathrm{Odd}_G(p(v))$ and $w \in N_G(u) \cap p(v)$, then by $[\prec .X]$ and $[\prec .Y]$ we must have $\lambda(w) = X$, so $\lambda'(w) = Y$.

$[\prec .Y]$: Consider any $w \preceq v$ with $w \neq v$ and $\lambda'(w) = Y$.

- If $w \in N_G(u)$, then $\lambda(w) = X$ and $w \neq u$. Using $[\prec .Z]$ we get $w \notin \mathrm{Odd}_G(p(v))$, so Equation 73 reduces to $w \in \mathrm{Odd}_{G\star u}(p'(v))$ if and only if $w \in p(v)$. Since $w \neq u$, this happens if and only if $w \in p'(v) = p(v)\Delta\{u\}$.

- If $w = u$, then $\lambda(w) = Z$. By $[\prec .X]$ we have $u \notin p(v)$ and by assumption $u \in \mathrm{Odd}_G(p(v))$. These mean $u \in p'(v) = p(v)\Delta\{u\}$ and $u \in \mathrm{Odd}_{G\star u}(p'(v))$ by Equation 73.

- If $w \notin N_G(u)$ and $w \neq u$, then $\lambda(w) = Y$. Equation 73 gives $w \in \mathrm{Odd}_{G\star u}(p'(v))$ if and only if $w \in \mathrm{Odd}_G(p(v))$. Using $[\prec .Y]$, this happens if and only if $w \in p(v)$ and thus if and only if $w \in p'(v) = p(v)\Delta\{u\}$.

Otherwise, suppose $u \notin \mathrm{Odd}_G(p(v))$.

$[\prec .X]$: Consider some $w \in p'(v) = p(v)$ with $w \neq v$ and $w \preceq v$.

- If $w = u$, then by $[\prec .X]$ and $[\prec .Y]$ we have $\lambda(u) = X = \lambda'(w)$.

- If $w \neq u$, then by $[\prec .X]$ we have $\lambda(w) \in \{X,Y\}$, so $\lambda'(w) \in \{X,Y\}$.

$[\prec .Z]$: Consider some $w \in \mathrm{Odd}_{G\star u}(p'(v)) = \mathrm{Odd}_G(p(v))\Delta(N_G(u) \cap p(v))$ with $w \neq v$ and $w \preceq v$. Since $u \notin N_G(u)$ and $u \notin \mathrm{Odd}_G(p(v))$, $w \neq u$.

- If $w \in N_G(u)$, then Equation 73 gives $w \in \mathrm{Odd}_G(p(v))\Delta p(v)$. Conditions $[\prec .X]$-$[\prec .Y]$ imply that $\lambda(w) \in \{X,Z\}$, so $\lambda'(w) \in \{Y,Z\}$.

- If $w \notin N_G(u)$, $w \in \mathrm{Odd}_G(p(v))$ and $\lambda'(w) = \lambda(w)$, so the result follows straightforwardly from $[\prec .Z]$ for $(p, \prec)$.

$[\prec .Y]$: Consider any $w \preceq v$ with $w \neq v$ and $\lambda'(w) = Y$.

- If $w = u$, then $\lambda(u) = Z$. From $[\prec .X]$, $u \notin p(v) = p'(v)$, and $u \notin \mathrm{Odd}_{G\star u}(p'(v))$ since $u \notin \mathrm{Odd}_G(p(v))$ by assumption and $u \notin N_G(u)$.

- If $w \in N_G(u)$, then $\lambda(w) = X$. From $[\prec .Z]$, $w \notin \mathrm{Odd}_G(p(v))$. Equation 73 then reduces to $w \in \mathrm{Odd}_{G\star u}(p'(v))$ if and only if $w \in p(v) = p'(v)$.

- If $w \notin N_G(u)$ and $w \neq u$, then $\lambda(w) = Y$. Equation 73 gives $w \in \mathrm{Odd}_{G\star u}(p'(v))$ if and only if $w \in \mathrm{Odd}_G(p(v))$, which happens if and only if $w \in p(v) = p'(v)$ by $[\prec .Y]$ for $(p, \prec)$.

$[\lambda .XY]$-$[\lambda .YZ]$: Suppose $\lambda(v) \in \{XY,XZ,YZ\}$. Since $v \in \overline{O} \setminus \{u\}$, $v \in p'(v)$ if and only if $v \in p(v)$. Whether or not $v \in p(v)$ and $v \in \mathrm{Odd}(p(v))$ is given by $[\lambda .XY]$-$[\lambda .YZ]$ for $(p, \prec)$ and the value of $\lambda(v)$.

- If $v \in N_G(u)$, then Equation 73 gives $v \in \mathrm{Odd}_{G\star u}(p'(v))$ if and only if $v \in \mathrm{Odd}_G(p(v))\Delta p(v)$. This change in the odd neighbourhood coincides with the definition of λ' flipping the labels of XZ and YZ.

- If $v \notin N_G(u)$, then Equation 73 gives $v \in \mathrm{Odd}_{G \star u}(p'(v))$ if and only if $v \in \mathrm{Odd}_G(p(v))$, which is enough since $\lambda'(v) = \lambda(v)$.

$[\lambda.X]$-$[\lambda.Y]$: We will prove these conditions by considering the possible cases for $\lambda(v) \in \{X, Y, Z\}$ in turn. In each case, since $v \neq u$, $v \in p'(v)$ if and only if $v \in p(v)$.

If $\lambda(v) = Z$, then $\lambda'(v) = Z$ and we get $v \in p(v)$ ($v \in p'(v)$) from $[\lambda.Z]$ for $(p, \prec)$.

Suppose $\lambda(v) = X$, so $v \in \mathrm{Odd}_G(p(v))$ from $[\lambda.X]$ for $(p, \prec)$.

- If $v \in N_G(u)$, then $\lambda'(v) = Y$ and Equation 73 gives $v \in \mathrm{Odd}_{G \star u}(p'(v))$ if and only if $v \notin p(v)$ ($v \notin p'(v)$).

- If $v \notin N_G(u)$, then $\lambda'(v) = X$ and Equation 73 gives $v \in \mathrm{Odd}_{G \star u}(p'(v))$.

Suppose $\lambda(v) = Y$, so $v \in p(v) \Leftrightarrow v \notin \mathrm{Odd}_G(p(v))$ from $[\lambda.Y]$ for $(p, \prec)$.

- If $v \in N_G(u)$, then $\lambda'(v) = X$. Since $v \in \mathrm{Odd}_G(p(v)) \Delta p(v)$, Equation 73 gives $v \in \mathrm{Odd}_{G \star u}(p'(v))$.

- If $v \notin N_G(u)$, then $\lambda'(v) = Y$. Equation 73 gives $v \in \mathrm{Odd}_{G \star u}(p'(v))$ if and only if $v \in \mathrm{Odd}_G(p(v))$. From $[\lambda.Y]$ and $v \neq u$, this happens if and only if $v \notin p'(v)$.

$\square$

Lemma D.15. Let (Γ, α) and (Γ', α') be the measurement patterns from Lemma D.12 related by local complementation about a vertex $u \in \overline{I}$. If $(p, \prec)$ is a focussed Pauli flow for Γ, then $(p', \prec)$ is a focussed pauli flow for Γ', where p' is defined in $\succ$-order (from outputs backwards) as

$$p'(v) := \left\{ \begin{array}{ll} p(v) \Delta \{u\} & \text{if } u \in \mathrm{Odd}_G(p(v)) \\ p(v) & \text{if } u \notin \mathrm{Odd}_G(p(v)) \end{array} \right\} \Delta \underset{\substack{w \in \overline{O} \setminus \{v\} \\ w \in p(v) \cup \mathrm{Odd}_G(p(v)) \\ w \in N_G(u) \cup \{u\} \\ w \in N_G(u) \Rightarrow \lambda(w) = XY \\ w = u \Rightarrow \lambda(u) \notin \{X, Y, Z\}}}{\Delta} p'(w) \tag{74}$$

Furthermore, if $\hat{p}$ is a focussed set for Γ, then we obtain a corresponding focussed set for Γ' as

$$\hat{p}' := \left\{ \begin{array}{ll} \hat{p} \Delta \{u\} & \text{if } u \in \mathrm{Odd}_G(\hat{p}) \\ \hat{p} & \text{if } u \notin \mathrm{Odd}_G(\hat{p}) \end{array} \right\} \Delta \underset{\substack{w \in \overline{O} \\ w \in \hat{p} \cup \mathrm{Odd}_G(\hat{p}) \\ w \in N_G(u) \cup \{u\} \\ w \in N_G(u) \Rightarrow \lambda(w) = XY \\ w = u \Rightarrow \lambda(u) \notin \{X, Y, Z\}}}{\Delta} p'(w) \tag{75}$$

Proof. Lemma D.14 gives a Pauli flow for Γ', which we will rename as $(q, \prec)$. We can view $(q, \prec)$ as an intermediate in the construction of $(p', \prec)$. It is not necessarily focussed, but we can refocus it using Lemma 4.6. Recall that this adds correction sets together according to Lemma B.1 for any case where the focussed conditions do not hold for q. We just need to show that these cases match those in the definition of p'.

As usual, the proofs for Pauli flows and focussed sets are almost identical, so herein we could change out references of q for

$$\hat{q} := \begin{cases} \hat{p} \Delta \{u\} & \text{if } u \in \mathrm{Odd}_G(\hat{p}) \\ \hat{p} & \text{if } u \notin \mathrm{Odd}_G(\hat{p}) \end{cases} \tag{76}$$

as an intermediate in the construction of $\hat{p}'$.

We will find the counterexamples to the focussed conditions for $(q, \prec)$ by considering $q(v)$ for an arbitrary $v \in \overline{O}$, and all of the cases for $w \in \overline{O} \setminus \{v\}$, covering each case of $\lambda(w)$ for each of $w = u$, $w \in N_G(u)$, and $w \notin N_G(u) \cup \{u\}$.

Firstly, we have $w = u$:

- $\lambda(u) = XY$, $\lambda'(u) = XZ$: By $[FZ]$, $u \notin \mathrm{Odd}_G(p(v))$, so $q(v) = p(v)$. This hence breaks $[FX]$ for q if $u \in p(v)$, i.e. if $u \in p(v) \cup \mathrm{Odd}_G(p(v))$. ✗

- $\lambda(u) = XZ$, $\lambda'(u) = XY$: $[FX]$ gives $u \notin p(v)$. By Equation 73, we break $[FZ]$ for q if $u \in \mathrm{Odd}_G(p(v))$, i.e. if $u \in p(v) \cup \mathrm{Odd}_G(p(v))$. ✗

- $\lambda(u) = YZ = \lambda'(u)$: $[FX]$ gives $u \notin p(v)$. From the definition of $q(v)$, $u \in q(v)$ (breaking $[FX]$ for q) if $u \in \mathrm{Odd}_G(p(v))$, i.e. if $u \in p(v) \cup \mathrm{Odd}_G(p(v))$. ✗

- $\lambda(u) = X = \lambda'(u)$: By $[FZ]$, $u \notin \mathrm{Odd}_G(p(v))$. We hence have $u \notin \mathrm{Odd}_{G\star u}(q(v))$ by Equation 73. ✓

- $\lambda(u) = Y$, $\lambda'(u) = Z$: $[FY]$ gives $u \in p(v) \Leftrightarrow u \in \mathrm{Odd}_G(p(v))$. In both cases for the definition of $q(v)$, this gives $u \notin q(v)$. ✓

- $\lambda(u) = Z$, $\lambda'(u) = Y$: We have $u \notin p(v)$ from $[FX]$. The definition of $q(v)$ then gives $u \in q(v) \Leftrightarrow u \in \mathrm{Odd}_G(p(v))$, and Equation 73 gives $u \in \mathrm{Odd}_{G\star u}(q(v)) \Leftrightarrow u \in \mathrm{Odd}_G(p(v))$. Hence $u \in q(v) \Leftrightarrow u \in \mathrm{Odd}_{G\star u}(q(v))$. ✓

Next, suppose $w \in N_G(u)$. In each case, the definition of $q(v)$ gives $w \in q(v) \Leftrightarrow w \in p(v)$ and Equation 73 gives $w \in \mathrm{Odd}_{G\star u}(q(v)) \Leftrightarrow w \in p(v)\Delta\mathrm{Odd}_G(p(v))$.

- $\lambda(w) = XY$, $\lambda'(w) = XY$: By $[FZ]$, we have $w \notin \mathrm{Odd}_G(p(v))$. This means it breaks $[FZ]$ for q when $w \in p(v)$, i.e. when $w \in p(v) \cup \mathrm{Odd}_G(p(v))$. ✗

- $\lambda(w) \in \{XZ, YZ, Z\}$, $\lambda'(w) \in \{XZ, YZ, Z\}$: $w \notin p(v)$ by $[FX]$, so $w \notin q(v)$. ✓

- $\lambda(w) = X$, $\lambda'(w) = Y$: $w \notin \mathrm{Odd}_G(p(v))$ holds by $[FZ]$, so $w \in \mathrm{Odd}_{G\star u}(q(v)) \Leftrightarrow w \in p(v)$ and hence $w \in q(v) \Leftrightarrow w \in \mathrm{Odd}_{G\star u}(q(v))$. ✓

- $\lambda(w) = Y$, $\lambda'(w) = X$: By $[FY]$, we have $w \notin p(v)\Delta\mathrm{Odd}_G(p(v))$, and therefore $w \notin \mathrm{Odd}_{G\star u}(q(v))$. ✓

For the final case of $w \notin N_G(u) \cup \{u\}$, we have $\lambda'(w) = \lambda(w)$, $w \in q(v) \Leftrightarrow w \in p(v)$, and $w \in \mathrm{Odd}_{G\star u}(q(v)) \Leftrightarrow w \in \mathrm{Odd}_G(p(v))$, so all of the focussed conditions are preserved.

In summary, we find that applying Lemma 4.6 to $(q, \prec)$ will add focussed corrections to $q(v)$ for any $w \in p(v) \cup \mathrm{Odd}_G(p(v))$ such that $w = u$ and $\lambda(u) \in \{XY, XZ, YZ\}$ or $w \in N_G(u)$ and $\lambda(w) = XY$. This exactly results in $(p', \prec)$, meaning this is a focussed Pauli flow for Γ'. □

Lemma D.16. *Let (Γ, α) and (Γ', α') be the measurement patterns and corresponding focussed Pauli flows $(p, \prec)$ and $(p', \prec)$ from Lemma D.15 formed by local complementation around a vertex $u \in \overline{I}$. Let $\mathbf{P}^{\perp v}$ be the primary extraction string for v in (Γ, α) from $(p, \prec)$. Then $\mathbf{P}'^{\perp v} := P_v'^{\perp v}\mathscr{R}^\dagger P_v^{\perp v}\mathbf{P}^{\perp v}\mathscr{R}$ is the*

corresponding primary extraction string in (Γ', α') from $(p', \prec)$, where

$$
\mathscr{R} := \left(\prod_{\substack{w \in N_G(u) \\ w \in O \cup \{v\}}} e^{i\frac{\pi}{4}Z_w} \right) \left(\prod_{\substack{w=u \\ w \in O \cup \{v\}}} e^{-i\frac{\pi}{4}X_w} \right)
$$
$$
\left(\prod_{\substack{w \in \overline{O} \cap (N_G(u) \cup \{u\}) \setminus \{v\} \\ w \succ v \\ w \in N_G(u) \Rightarrow \lambda(u) = XY \\ w = u \Rightarrow \lambda(w) \notin \{X,Y,Z\}}}^{\succ} e^{(-1)^{D_w + A^{p'}_{w \to v}} i\frac{\pi}{4} P^{w \to v}_v \mathbf{P'}^{\perp w}} \right) \tag{77}
$$

Furthermore, if $\mathbf{P}$ is the stabilizer from a focussed set $\hat{p}$ in (Γ, α), then $\mathbf{P}' := \mathscr{R}^\dagger \mathbf{P} \mathscr{R}$ is the corresponding stabilizer for (Γ', α').

Proof. We proceed following a similar strategy to Lemma D.3 of simultaneously showing that $\mathbf{P}'^{\perp v}$ operates only on the outputs and that it matches the extraction string from $(p', \prec)$ up to phase, then showing that it is a valid extraction string.

The conjugation of $P^{\perp v}_v \mathbf{P}^{\perp v}$ by the $e^{-i\frac{\pi}{4}X_u}$ and $e^{i\frac{\pi}{4}Z_w}$ terms adjusts the Pauli terms of the operator in the same way as the intermediate Pauli flow from Lemma D.14. The $e^{-i\frac{\pi}{4}X_u}$ term adds an X when conjugating Z or Y, matching the addition of $\{u\}$ to $p(v)$ if $u \in \mathrm{Odd}(p(v))$. Similarly, the $e^{i\frac{\pi}{4}Z_w}$ term adds a Z when conjugating X or Y, matching the update of the odd neighbourhood according to Equation 73.

Each of the remaining terms in $\mathscr{R}$ cover the cases for the focussing in Lemma D.15. The movement of $e^{(-1)^{D_w + A^{p'}_{w \to v}} i\frac{\pi}{4} P^{w \to v}_v \mathbf{P'}^{\perp w}}$ adds $P^{w \to v}_v \mathbf{P'}^{\perp w}$ if it anticommutes with the operator. From Lemma C.9 and $P^{\perp v} P^{w \to v} = (-1)^{F^{p_k}_{w \to v}} P^{w \to v} P^{\perp v}$, the combined operators anticommute if $w \in p_k(v) \cup \mathrm{Odd}(p_k(v))$ (where p_k is the intermediate Pauli flow at this stage in the focussing). This matches the focussing step, which adds $p'(w)$ when $w \in p_k(v) \cup \mathrm{Odd}(p_k(v))$ doesn't satisfy the focussed conditions - by Lemma D.15, this corresponds to the conditions on w in the definition of $\mathscr{R}$.

Combining these together, we get that $\mathscr{R}$ maps $P^{\perp v}_v \mathbf{P}^{\perp v}$ to the equivalent for $(p', \prec)$ up to phase.

To complete this for the phase, it remains to show that $\mathbf{P}'^{\perp v}$ is a valid primary extraction string. Recall from Van den Nest's Theorem (Equation 68) that local complementation of the graph state is balanced by rotations on u and its neighbours. Let's look at how these rotations affect the measurements on those qubits in each case.

$\lambda(w)$	$w = u$	$w \in N_G(u)$
XY	$\langle+_{XZ,\frac{\pi}{2}}\rvert e^{i\frac{\pi}{4}X} = \langle+_{XY,0}\rvert$	$\langle+_{XY,\frac{\pi}{2}}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{XY,0}\rvert$
XZ	$\langle+_{XY,\frac{\pi}{2}}\rvert e^{i\frac{\pi}{4}X} \approx \langle+_{XZ,0}\rvert$	$\langle+_{YZ,0}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{XZ,0}\rvert$
YZ	$\langle+_{YZ,\frac{\pi}{2}}\rvert e^{i\frac{\pi}{4}X} = \langle+_{YZ,0}\rvert$	$\langle+_{XZ,0}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{YZ,0}\rvert$
X	$\langle+_{X,\alpha(w)}\rvert e^{i\frac{\pi}{4}X} \approx \langle+_{X,\alpha(w)}\rvert$	$\langle+_{Y,\alpha(w)}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{X,\alpha(w)}\rvert$
Y	$\langle+_{Z,\alpha(w)+\pi}\rvert e^{i\frac{\pi}{4}X} \approx \langle+_{Y,\alpha(w)}\rvert$	$\langle+_{X,\alpha(w)+\pi}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{Y,\alpha(w)}\rvert$
Z	$\langle+_{Y,\alpha(w)}\rvert e^{i\frac{\pi}{4}X} \approx \langle+_{Z,\alpha(w)}\rvert$	$\langle+_{Z,\alpha(w)}\rvert e^{-i\frac{\pi}{4}Z} \approx \langle+_{Z,\alpha(w)}\rvert$

We can now consider the primary extraction string of some $v \in \overline{O}$. As usual, we define $\mathscr{C}'$ to be the linear map of interest from Definition 4.2 for (Γ', α') and let $\mathscr{C}$ refer to the equivalent for (Γ, α).

$$\mathscr{C}' := \left(\prod_{\substack{w \succ v \\ \lambda'(w) \notin \{X,Y,Z\}}} \langle+_{\lambda'(w),0}\rvert_w \right) \left(\prod_{\substack{w \in \overline{O}\setminus\{v\} \\ \lambda'(w) \in \{X,Y,Z\}}} \langle+_{\lambda'(w),\alpha'(w)}\rvert_w \right) E_{G\star u} N_{\overline{I}} \tag{78}$$

We start by applying the reverse of the extraction algorithm to modify the measurement angles in the diagram to match those in the table above. Since this changes the angles (which must be 0 to extract any rotation before them), we must perform this in $\prec$-order.

$$\mathscr{C}' \approx \left(\prod_{\substack{w \succ v \\ w \in N_G(u)\cup\{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w)=XY \\ w=u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle+_{\lambda'(w),\frac{\pi}{2}}\rvert_w \; e^{(-1)^{1+D_w}i\frac{\pi}{4}P_w^{\perp w}} \right)$$

$$\left(\prod_{\substack{w \succ v \\ w \neq u \\ w \in N_G(u) \Rightarrow \lambda'(w) \in \{XZ,YZ\} \\ w \notin N_G(u) \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle+_{\lambda'(w),0}\rvert_w \right) \left(\prod_{\substack{w \in \overline{O}\setminus\{v\} \\ \lambda'(w) \in \{X,Y,Z\}}} \langle+_{\lambda'(w),\alpha'(w)}\rvert_w \right) E_{G\star u} N_{\overline{I}}$$

$$= \left(\prod_{\substack{w \succ v \\ w \in N_G(u)\cup\{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w)=XY \\ w=u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}}^{\prec} e^{(-1)^{1+D_w+A_{w\to v}^{p'}}i\frac{\pi}{4}P_v^{w\to v}\mathbf{P}'^{\perp w}} \right) \left(\prod_{\substack{w \succ v \\ w \in N_G(u)\cup\{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w)=XY \\ w=u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle+_{\lambda'(w),\frac{\pi}{2}}\rvert_w \right)$$

$$\left(\prod_{\substack{w \succ v \\ w \neq u \\ w \in N_G(u) \Rightarrow \lambda'(w) \in \{XZ,YZ\} \\ w \notin N_G(u) \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle+_{\lambda'(w),0}\rvert_w \right) \left(\prod_{\substack{w \in \overline{O}\setminus\{v\} \\ \lambda'(w) \in \{X,Y,Z\}}} \langle+_{\lambda'(w),\alpha'(w)}\rvert_w \right) E_{G\star u} N_{\overline{I}} \tag{79}$$

Next, we apply Van den Nest's Theorem to yield $\mathscr{C}$.

$$
\mathscr{C}' \approx \left(\prod_{\substack{w \succ v \\ w \in N_G(u) \cup \{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w) = XY \\ w = u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}}^{\prec} e^{(-1)^{1+D_w+A_{w \to v}^{p'}} i \frac{\pi}{4} P_v^{w \to v} \mathbf{P'}^{\perp w}} \right) \left(\prod_{\substack{w \succ v \\ w \in N_G(u) \cup \{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w) = XY \\ w = u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle +_{\lambda'(w), \frac{\pi}{2}} |_w \right)
$$

$$
\left(\prod_{\substack{w \succ v \\ w \neq u \\ w \in N_G(u) \Rightarrow \lambda'(w) \in \{XZ,YZ\} \\ w \notin N_G(u) \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}} \langle +_{\lambda'(w), 0} |_w \right) \left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda'(w) \in \{X,Y,Z\}}} \langle +_{\lambda'(w), \alpha'(w)} |_w \right)
$$

$$
e^{i \frac{\pi}{4} X_u} \left(\prod_{w \in N_G(u)} e^{-i \frac{\pi}{4} Z_w} \right) E_G N_{\bar{I}}
$$

$$
\approx \left(\prod_{\substack{w \succ v \\ w \in N_G(u) \cup \{u\} \\ w \in N_G(u) \Rightarrow \lambda'(w) = XY \\ w = u \Rightarrow \lambda'(w) \notin \{X,Y,Z\}}}^{\prec} e^{(-1)^{1+D_w+A_{w \to v}^{p'}} i \frac{\pi}{4} P_v^{w \to v} \mathbf{P'}^{\perp w}} \right) \left(\prod_{\substack{w = u \\ w \in O \lor w \preceq v \\ w \in \overline{O} \setminus \{v\} \Rightarrow \lambda(w) \notin \{X,Y,Z\}}} e^{i \frac{\pi}{4} X_w} \right) \tag{80}
$$

$$
\left(\prod_{\substack{w \in N_G(u) \\ w \in O \lor w \preceq v \\ w \in \overline{O} \setminus \{v\} \Rightarrow \lambda(w) \notin \{X,Y,Z\}}} e^{-i \frac{\pi}{4} Z_w} \right) \left(\prod_{\substack{w \succ v \\ \lambda(w) \notin \{X,Y,Z\}}} \langle +_{\lambda(w), 0} |_w \right)
$$

$$
\left(\prod_{\substack{w \in \overline{O} \setminus \{v\} \\ \lambda(w) \in \{X,Y,Z\}}} \langle +_{\lambda(w), \alpha(w)} |_w \right) E_G N_{\bar{I}}
$$

$$
= \mathscr{R}^\dagger \left(\prod_{\substack{w = u \neq v \\ w \preceq v \\ \lambda(w) \notin \{X,Y,Z\}}} e^{i \frac{\pi}{4} X_w} \right) \left(\prod_{\substack{w \in \overline{O} \cap N_G(u) \setminus \{v\} \\ w \preceq v \\ \lambda(w) \notin \{X,Y,Z\}}} e^{-i \frac{\pi}{4} Z_w} \right) \mathscr{C}
$$

The final step is then to introduce $P_v^{\perp v}\mathbf{P}^{\perp v}$, move it through $\mathscr{R}$ and reverse the construction to return to $\mathscr{C}'$.

$$
\mathscr{C}' \approx \mathscr{R}^{\dagger} \left(\prod_{\substack{w=u\neq v \\ w\preceq v \\ \lambda(w)\notin\{X,Y,Z\}}} e^{i\frac{\pi}{4}X_w} \right) \left(\prod_{\substack{w\in\overline{O}\cap N_G(u)\setminus\{v\} \\ w\preceq v \\ \lambda(w)\notin\{X,Y,Z\}}} e^{-i\frac{\pi}{4}Z_w} \right) \mathbf{P}^{\perp v}P_v^{\perp v}\mathscr{C}
$$

$$
= \mathbf{P'}^{\perp v}P_v'^{\perp v}\mathscr{R}^{\dagger} \left(\prod_{\substack{w=u\neq v \\ w\preceq v \\ \lambda(w)\notin\{X,Y,Z\}}} e^{i\frac{\pi}{4}X_w} \right) \left(\prod_{\substack{w\in\overline{O}\cap N_G(u)\setminus\{v\} \\ w\preceq v \\ \lambda(w)\notin\{X,Y,Z\}}} e^{-i\frac{\pi}{4}Z_w} \right) \mathscr{C} \tag{81}
$$

$$
\approx \mathbf{P'}^{\perp v}P_v'^{\perp v}\mathscr{C}'
$$

Hence $\mathbf{P'}^{\perp v}$ is a valid primary extraction string for v, and by Lemma C.8 it is exactly the one generated by $(p',\prec)$. As usual, the proof is virtually identical for stabilizers from focussed sets. □

Theorem D.17. (Restatement of Theorem 5.3) Performing local complementation about a vertex u corresponds to the following sequence of actions on the PDDAG:

1. For each output w neighbouring u, a $(Z_w, \frac{\pi}{2})$ rotation is pulled from the initial stabilizer block all the way to the end of the rotation DAG;

2. If u is an output, a $(X_u, -\frac{\pi}{2})$ rotation is pulled from the initial stabilizer block all the way to the end of the rotation DAG;

3. For each vertex w neighbouring u with $\lambda(w) = XY$, and for u itself if $\lambda(u)$ is planar, we pull a $\left(\mathbf{P'}^{\perp w}, (-1)^{D_w}\frac{\pi}{2}\right)$ rotation from the initial stabilizer block to merge it into the existing rotation for w or u. We do this in $\succ$-order over such w vertices.

Proof. Lemma D.12 shows that the additions of $\frac{\pi}{2}$ to measurement angles and outputs matches the set of rotations that are moved in the above PDDAG rewrite sequence.

The updates for the Pauli strings for each rotation and the stabilizers/tableau are covered by Lemma D.16. In particular, in each case we consider planar vertices and so all of the $P_v^{w\to v}$ terms in $\mathscr{R}$ are I, meaning the update to the extraction string corresponds to the effect of the Reorder Rules. However, the phase may differ in the update from $P^{\perp v}$ to $P'^{\perp v}$.

$P^{\perp v}$	$e^{i\frac{\pi}{4}X_v}P^{\perp v}e^{-i\frac{\pi}{4}X_v}$	$e^{-i\frac{\pi}{4}Z_v}P^{\perp v}e^{i\frac{\pi}{4}Z_v}$
X	X	Y
Y	$-Z$	$-X$
Z	Y	Z

This shows that we experience an extra phase flip in the extraction string for u when $\lambda(v) = XZ$, balanced by the negating of $\alpha(u)$ in the statement of Lemma D.12. For $v \in N_G(u)$, we get an extra phase flip for $\lambda(v) \in \{XZ, YZ\}$ from Equation 10, which cancels with the above for $\lambda(v) = XZ$ and is balanced by the negating of $\alpha(v)$ when $\lambda(v) = YZ$. □

Example D.18. Again, we start from the same measurement pattern from Example C.13 and apply local complementation about vertex d. This complements the connectivity within $\{a,b,c,o_2\}$.

v	$\lambda'(v)$	$\alpha'(v)$	$p'(v)$	$\mathrm{Odd}(p'(v))$	$\{u \mid v \prec u\}$
i	XY	$\alpha(i)$	b,c,o_2	i,a,d,o_1	a,b,c,o_1,o_2
a	XZ	$-\alpha(a)$	a,c,o_1,o_2	a,d,o_1	c,o_1,o_2
b	XY	$\alpha(b)+\frac{\pi}{2}$	c	b,d,o_1,o_2	c,o_1,o_2
c	XY	$\alpha(c)+\frac{\pi}{2}$	o_1	c	o_1
d	Z	$\alpha(d)+\pi$	d,o_2	d,o_2	o_2

We update the focussed set $\hat{p}$ to $\hat{p}' = \{c,o_1,o_2\}$, and to preserve semantics we also have an $RZ(-\frac{\pi}{2})$ applied to output o_2.

Now we turn to the sequence of rewrites on the PDDAG. For step 1, o_2 neighbours d so we pull a $(I_1Z_2,\frac{\pi}{2})$ rotation through to the end. The Reorder Rules update the rotations with anticommuting Pauli strings (the rotation from a, the tableau row and rotation from i, and the free stabilizer from $\hat{p}$) by multiplication.

Ins	Outs	Sign
X	$Y\ Z$	$(-)^{a_d+1}$
Z	Y	$+$
	$Z\ Y$	$+$

$$(I_1Y_2,\alpha(i)) \longrightarrow ((-1)^{a_d+1}Z_1X_2,\alpha(a)) \longrightarrow (I_1Z_2,\tfrac{\pi}{2})$$
$$((-1)^{a_d+1}Y_1Z_2,\alpha(b)) \longrightarrow (X_1I_2,\alpha(c))$$

Step 2 does not apply since d is not an output. Finally, in step 3, d is not planar but it is adjacent to b and c that are labelled as XY. Since $b \prec c$, we first pull out the $(X_1I_2,\frac{\pi}{2})$ rotation.

Ins	Outs	Sign
X	$Z\ Z$	$(-)^{a_d+1}$
Z	Y	$+$
	$Y\ Y$	$-$

$$(I_1Y_2,\alpha(i)) \longrightarrow ((-1)^{a_d}Y_1X_2,\alpha(a)) \longrightarrow (I_1Z_2,\tfrac{\pi}{2})$$
$$((-1)^{a_d+1}Z_1Z_2,\alpha(b)) \longrightarrow (X_1I_2,\alpha(c)+\tfrac{\pi}{2})$$

Then we pull out the $((-1)^{a_d+1}Z_1Z_2,\frac{\pi}{2})$ rotation for b.

Ins	Outs	Sign
X	$Z\ Z$	$(-)^{a_d+1}$
Z	$Z\ X$	$(-)^{a_d}$
	$Y\ Y$	$-$

$$((-1)^{a_d}Z_1X_2,\alpha(i)) \longrightarrow ((-1)^{a_d}Y_1X_2,\alpha(a)) \longrightarrow (I_1Z_2,\tfrac{\pi}{2})$$
$$((-1)^{a_d+1}Z_1Z_2,\alpha(b)+\tfrac{\pi}{2}) \longrightarrow (X_1I_2,\alpha(c)+\tfrac{\pi}{2})$$

We now find that this is the same PDDAG that would be extracted after applying the local complementation on the measurement pattern (take care when determining the phases during extraction as the phase from including d in $p'(v) \cup \mathrm{Odd}(p'(v))$ is now $(-1)^{a_d+1}$ because $\alpha'(d) = \alpha(d)+\pi$). In the final step, it is important to have maintained the convention of extracting as $((-1)^{D_v}\mathbf{P}^{\perp v},\alpha(v))$ and keeping the -1 terms with the string rather than the angle. If we had instead moved $((-1)^{a_d}Z_1Z_2,-\frac{\pi}{2})$, we would end up with a rotation of $\alpha(b)-\frac{\pi}{2}$ and different phases on the rotation and tableau row for i.

Remark D.19. One should note that the action of local complementation is not quite self-inverse due to some measurement angles being updated by $\frac{\pi}{2}$. However, performing the inverse update is also valid and just relies on using the other case of Equation 68 to flip the angles that are added. This can similarly be simulated by just flipping the angles of the rotations being moved through the PDDAG.

Now that we have the ability to simulate the effects of local complementation, extending this to pivoting is trivial since it simply decomposes into a sequence of local complementations.

Definition D.20. Given a graph $G = (V, E)$ with some designated edge $u \sim v \in E$, the pivot of G about the edge $u \sim v$ is the operation resulting in the graph:

$$G \wedge uv := G \star u \star v \star u (= G \star v \star u \star v) \tag{82}$$

Lemma D.21. Let (Γ, α) describe a measurement pattern with $\Gamma = (G, I, O, \lambda)$ and some chosen vertices $u, v \in \overline{I}$ such that $u \sim v$. Then performing a pivot around the edge $u \sim v$ gives an equivalent measurement pattern (Γ', α') where $\Gamma' = (G \wedge uv, I, O, \hat{\lambda})$, for all $a \in \{u, v\}$ and $w \in \overline{O} \setminus \{u, v\}$

$$(\lambda'(a), \alpha'(a)) := \begin{cases} (YZ, -\alpha(a)) & \text{if } \lambda(a) = XY \\ (XZ, \frac{\pi}{2} - \alpha(a)) & \text{if } \lambda(a) = XZ \\ (XY, -\alpha(a)) & \text{if } \lambda(a) = YZ \\ (Z, \alpha(a)) & \text{if } \lambda(a) = X \\ (Y, \alpha(a) + \pi) & \text{if } \lambda(a) = Y \\ (X, \alpha(a)) & \text{if } \lambda(a) = Z \end{cases} \tag{83}$$

$$(\lambda'(w), \alpha'(w)) := \begin{cases} (\lambda(w), \alpha(w) + \pi) & \text{if } w \in N_G(u) \cap N_G(v) \wedge \lambda(w) \in \{XY, X, Y\} \\ (\lambda(w), -\alpha(w)) & \text{if } w \in N_G(u) \cap N_G(v) \wedge \lambda(w) \in \{XZ, YZ\} \\ (\lambda(w), \alpha(w)) & \text{otherwise} \end{cases} \tag{84}$$

plus each output neighbouring both u and v is followed by a Z gate and if each of u or v is an output then it is followed by a Hadamard gate.

Proof. We can similarly extend the case for planar measurements given in the proof of [5, Lemma 4.5] to Pauli measurements by considering how they embed into the planes as in Equation 51. Similarly, this follows by using Van den Nest's Theorem [33] (Equation 68) multiple times, alternating the phases of the angles each time, and rotating the measurement projections accordingly. $\qquad \square$

Since this proof requires alternating between the different phase variants of Equation 68, simulating the effect in the PDDAG requires alternating between the different variants of local complementation as mentioned in Remark D.19 but is otherwise just a trivial application of Theorem 5.3.

D.4 Alternative Focussed Pauli Flows

Lemma D.22. Let (Γ, α) describe a measurement pattern with a focussed set $\hat{p}$, vertex $v \in \overline{O}$, and two focussed Pauli flows $(p, \prec)$ and $(p', \prec)$ related by the addition of $\hat{p}$ to $p(v)$ as in Lemma B.6. Let $\hat{\mathbf{P}}$ be the stabilizer corresponding to $\hat{p}$, and $\mathbf{P}^{\perp v}$ and $\mathbf{P}'^{\perp v}$ be the primary extraction strings for v according to $(p, \prec)$ and $(p', \prec')$ respectively. Then $\mathbf{P}'^{\perp v} = \mathbf{P}^{\perp v} \hat{\mathbf{P}}$.

Proof. Lemma C.7 immediately gives this result up to phase. To invoke Lemma C.8, we consider the linear map $\mathscr{C}$ for extraction strings of v. From the conditions of Lemma B.6 and the definition of $\prec'$, any $w \in \hat{p} \cup \text{Odd}(\hat{p})$ must satisfy $w \neq v$ and $\lambda(w) \notin \{X, Y, Z\} \Rightarrow v \prec w$. So both $P_v^{\perp v} \mathbf{P}^{\perp v}$ and $\hat{\mathbf{P}}$ are stabilizers of $\mathscr{C}$, meaning they combine to give the stabilizer $P_v^{\perp v} \mathbf{P}^{\perp v} \hat{\mathbf{P}}$ which defines this as a valid primary extraction string. $\qquad \square$

Theorem D.23. (Restatement of Theorem 5.4) Let (Γ, α) describe a measurement pattern with some focussed Pauli flow $(p, \prec)$, a focussed set $\hat{p}$, and some vertex $u \in \overline{O}$ such that $\forall w \in \hat{p} \cup \mathrm{Odd}(\hat{p}).\lambda(w) \in \{XY, XZ, YZ\} \Rightarrow w \neq u \wedge u \prec w$. Updating $p(u)$ to $p(u)\Delta\hat{p}$ corresponds to a free action on the isometry tableau if u is an input, and applying the Product Rotation Lemma to the rotation from u with the stabilizer of $\hat{p}$ if u has a planar label.

Therefore, any two focussed Pauli flows for the same labelled open graph yield PDDAGs that are related by a sequence of applications of the Product Rotation Lemma and free actions on the isometry tableau.

Proof. This update on the Pauli flow is the same as from Lemma B.6, meaning Lemma D.22 gives the update to the primary extraction strings. For both uses of u in extraction, we assume u has a planar label. If u is an input considered for the tableau of the PDDAG, multiplication of a row of the tableau by a free stabilizer is a free action as it just gives another equivalent set of generators for the stabilizer group. On the other hand, if u is some planar vertex giving a rotation in the PDDAG, we note that $\hat{\mathbf{P}}$ remains a stabilizer after applying each of the rotations for vertices $w \preceq u$ (they commute according to Lemma C.9), allowing us to apply the Product Rotation Lemma to get the same effect.

To show that any two focussed Pauli flows $(p, \prec)$ and $(p', \prec')$ can be related using this, we start with $(p, \prec)$ and derive the sequence of transformations needed to reach $(p', \prec')$.

We consider weakening of the partial order in a Pauli flow to be a free action, as well as the reverse so long as the Pauli flow conditions are preserved, since neither affect the semantics or extraction. So suppose wlog that $\prec$ and $\prec'$ are minimal, i.e. $\prec$ is the transitive closure of $\{(u, v)| v \in p(u) \cup \mathrm{Odd}(p(u)) \wedge u \neq v \wedge \lambda(v) \in \{XY, XZ, YZ\}\}$ (we ignore Pauli measurements by Lemma B.11).

We proceed inductively over $\prec$. Under some indexing of the vertices respecting $\prec$, let v be the first for which $p(v) \neq p'(v)$, i.e. $\forall w \prec v.p(v) = p'(v)$. Our assumptions on minimality then imply that $\forall w \in \overline{O}.w \prec v \Leftrightarrow w \prec' v$.

By Lemma B.7, $\hat{p}^v := p(v)\Delta p'(v)$ is a focussed set for any $v \in \overline{O}$. Conditions $[\lambda.XY]\text{-}[\lambda.YZ]$ imply that $\lambda(v) \in \{XY, XZ, YZ\} \Rightarrow v \notin \hat{p}^v \cup \mathrm{Odd}(\hat{p}^v)$. Using conditions $[\prec .X]$ and $[\prec .Z]$, we also have $\forall w \in \hat{p}^v \cup \mathrm{Odd}(\hat{p}^v).w \neq v \wedge \lambda(w) \in \{XY, XZ, YZ\} \Rightarrow v \prec w \vee v \prec' w$. Since $v \prec' w \Rightarrow v \preceq' w$ and $v \preceq' w \Leftrightarrow v \preceq w$ by the minimality assumption, we now have $\forall w \in \hat{p}^v \cup \mathrm{Odd}(\hat{p}^v).\lambda(w) \in \{XY, XZ, YZ\} \Rightarrow w \neq v \wedge v \preceq w$. This means that combining $(p, \prec)$ with $\hat{v}$ according to Lemma B.6 (and reducing to the minimal order) gives a focussed Pauli flow equal to $(p', \prec')$ up to v in $\prec$.

Repeating this inductively over the remaining vertices will eventually yield $(p', \prec')$. $\qquad\square$

Example D.24. We revisit the measurement pattern from Example C.13. Recall that we have a focussed set $\hat{p} = \{c, o_2\}$ with $\mathrm{Odd}(\hat{p}) = \{a, o_1\}$, so we can freely add this to the correction sets for any of i, b, or d. Suppose we do so for b and d to give the following focussed Pauli flow:

v	$\lambda(v)$	$p'(v)$	$\mathrm{Odd}(p'(v))$	$\{u\|v \prec u\}$
i	XY	b, o_2	i, a	a, b, c, o_1, o_2
a	YZ	a, c, d, o_2	d, o_1, o_2	c, o_1, o_2
b	XY	d, o_1, o_2	a, b, d, o_2	a, c, o_1, o_2
c	XY	o_1	c	o_1
d	Y	c	a, d, o_1	a, c, o_1, o_2

The difference between $p(d)$ and $p'(d)$ has no effect on the PDDAG since d is not involved in the extraction procedure. So the only rewrites needed are the multiplication of the top row of the isometry tableau by the bottom, and applying the Product Rotation Lemma to the b rotation mapping $Y_1Z_2 \mapsto (Y_1Z_2)(Z_1X_2) = -X_1Y_2$.

Ins	Outs	Sign
X	X Y	$(-)^{a_d}$
Z	X	$+$
	Z X	$+$

$$(I_1 X_2, \alpha(i)) \longrightarrow ((-1)^{a_d} Z_1 Y_2, \alpha(a)) \longrightarrow (X_1 I_2, \tfrac{\pi}{2})$$

$$((-1)^{a_d} X_1 Y_2, \alpha(b))$$