

Working with measurement-based computations on qudits

Piotr Mitosek 

Institute of Theoretical Physics
Leibniz University Hannover
Germany

piotr.mitosek(at)itp.uni-hannover.de

Miriam Backens 

Université de Lorraine, CNRS, Inria, LORIA
F-54000 Nancy
France

Measurement-based quantum computing is a universal model of quantum computation in which successive product measurements of an entangled resource state drive the computation. The non-deterministic nature of measurements necessitates adaptivity to ensure an overall deterministic computation. Flow structures characterise cases in which such an adaptive correction procedure is possible. Recently, flow has been defined in a setting where the resource states are prime-dimensional qudit graph states rather than the usual qubit graph states. Yet, this qudit flow definition is more burdensome to work with than analogous definitions for qubits.

Here, we give a simpler definition of qudit flow and consider various useful properties of this flow, drawing on results for the qubit case. In particular, we show how to focus qudit flow and argue that focused flow is canonical. We improve the previous algebraic formulation to capture focused flow and use it to obtain an $O(n^3)$ flow-finding algorithm (where n is the number of qudits), matching the best known complexity for qubit flows and improving on the previous $O(n^4)$ result for qudits. Furthermore, we explore multiple flow-preserving transformations, thus opening a pathway to using flow for optimisation. These transformations include pivoting, removal and insertion of certain types of vertices, and reversibility of flow. Lastly, we propose an algorithmic approach to generating large qudit computations with flow, for testing or machine learning.

1 Introduction

In computer science, it is standard to encode information in bits, i.e. using binary values 0 or 1. This has carried across to quantum computing, where most research focuses on qubits, aka ‘quantum bits’, that have two classical states (and superpositions thereof). True qubit systems exist in nature, nevertheless, many of the most promising candidates for physical qubits naturally have more than two basic states, and there is increasing interest in quantum computation using d -dimensional systems, called qudits.

Much of the theoretical research about qudit quantum computation focuses on the quantum circuit model [20, 23, 43]. This means a computation is driven entirely by unitary gates; state preparation and measurements can be restricted to the computational basis and the latter are needed only for reading out information at the end. The qudit Clifford fragment for prime dimensions d is core to qudit quantum computing analogous to its role with respect to qubits [25, 20]. There are definitions of qudit stabiliser states and also qudit graph states [6], which in dimension d correspond to $\mathbb{F}_d$ -weighted graphs: where in the binary case, edges are either present or not, in an $\mathbb{F}_d$ -weighted graph, each pair of vertices is associated with a weight $w \in \mathbb{F}_d$ such that $w = 0$ corresponds to there being no edge between the two vertices.

Qudit graph states then permit the generalisation of the one-way model of measurement-based quantum computation (MBQC) [36] to prime-dimensional systems [33, 10, 8, 37]. In the one-way model, an entangled resource graph state is prepared at the beginning; this resource can be independent of the computation (as long as it is large enough). The computation proceeds by successive adaptive single qudit measurements in different *measurement spaces* (called measurement planes in the qubit case). The

individual measurements change the underlying state and can thereby drive a computation, yet they are not generally deterministic. This is why the ‘successive adaptive’ part matters: If the combination of underlying graph state and measurement spaces for each qudit (formalised as a ‘labelled open $\mathbb{F}_d$ -graph’) is chosen well, then it is possible to modify later measurements depending on the outcomes of earlier measurements so as to make the overall computation deterministic. Whether overall determinism is possible is characterised by a property called $\mathbb{F}_d$ -flow¹ [8], the qudit equivalent of ‘extended gflow’ on qubits [11]. Yet the current definition of $\mathbb{F}_d$ -flow is somewhat unwieldy in practice as one validity condition requires considering any totalisation of the partial order that is part of the $\mathbb{F}_d$ -flow specification.

In this work, we develop a different perspective on $\mathbb{F}_d$ -flow on qudits, which adapts the ‘algebraic Pauli flow’ formulation [32] to qudits and is more straightforward to work with. This then allows several results from qubit gflow to be reproduced for qudits. In particular, we adapt the qubit gflow concept of ‘focusing’ [3] to $\mathbb{F}_d$ -flow and show that the existence of $\mathbb{F}_d$ -flow is equivalent to the existence of focused $\mathbb{F}_d$ -flow. Next, we directly generalise previous algebraic formulations of qubit flow [29, 31, 32]: given a labelled open graph, we show simple linear algebra conditions allowing verification of the existence of $\mathbb{F}_d$ -flow. We also show how to lift the definition of and core results about qubit ‘focused sets’ [39, 32] to qudit *focused vectors* – these structures classify ‘do nothing’ stabilizers, i.e. stabilizers that do not impact the correction procedure. We use the new algebraic interpretation to devise new fast algorithms for finding $\mathbb{F}_d$ -flow, improving previous $O(n^4)$ routine [8] to $O(n^3)$, i.e. the complexity matching our qubit result [32]. While $\mathbb{F}_d$ -flow only works for prime d , our algorithms solve the underlying linear algebra problem over any finite field $\mathbb{F}_d$, not necessarily a prime field. Lastly, we adapt some flow-preserving rules previously known for qubit flows to also work for qudit flow. In addition to the local complementation and local scaling rules that were already known for the qudit case, this includes pivoting [17, 3, 39], reversibility [32], and the removal or insertion of a family of measurements [17, 3, 27, 5]. We also show how these flow-preserving rules allow the generation of arbitrary labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow, generalising a recent result of one of the authors in the qubit case [2].

The generalisation from qubits to qudits comes with several intricacies as the field $\mathbb{F}_2$ (having characteristic 2) behaves quite differently than $\mathbb{F}_d$ for any other prime d . In general, working over qudits means there are many more possible measurement spaces, correction operators, and graph symmetries to be considered and new choices had to be made to define e.g. the concept of focusing or a useful pivot operation in the qudit context.

In Section 2, we provide the necessary background. Next, in Section 3, we adapt the notion of ‘focusing’ to the qudit case and show that focused $\mathbb{F}_d$ -flow exists whenever any $\mathbb{F}_d$ -flow exists. In Section 4, we give an algebraic formulation of focused $\mathbb{F}_d$ -flow. We use this formulation in Section 5 to derive a faster flow-finding algorithm. In Section 6, we consider $\mathbb{F}_d$ -flow-preserving rewriting. Lastly, in Section 7, we give conclusions and discuss potential further work. Various technical parts of the paper are included in the appendices.

2 Background

We work with finite fields $\mathbb{F}_d$ of order $d = p^k$ for prime p and a positive integer k . Unless specified otherwise, we assume throughout that $k = 1$ and thus $d = p$ is prime. We use $\mathbb{F}_d^*$ for $\mathbb{F}_d \setminus \{0\}$. We also write $*$ for an arbitrary element of $\mathbb{F}_d^*$, i.e. an arbitrary non-zero element of $\mathbb{F}_d$. All matrices are always over the field $\mathbb{F}_d$. Let K be a matrix with row labels $\mathcal{R}$ and column labels $\mathcal{C}$. We refer to it as $\mathcal{R} \times \mathcal{C}$

¹ This property was previously called $\mathbb{Z}_d$ -flow, yet it actually relies on properties of a field, not just a ring. We change the notation and terminology to emphasise this.

matrix. We write $K_{v,*}$ where $v \in \mathcal{R}$ for the v -labelled row of K and, similarly, $K_{*,u}$ with $u \in \mathcal{C}$ for the u -labelled column of K . Given $\mathcal{R}' \subseteq \mathcal{R}$ and $\mathcal{C}' \subseteq \mathcal{C}$, we write $K_{\mathcal{R}'}^{\mathcal{C}'}$ for the $\mathcal{R}' \times \mathcal{C}'$ submatrix of K .

A comparison of these concepts and other notions throughout this paper to what is known for qubits (i.e. the case $d = 2$) can be found in Appendix A.

Definition 2.1. An $\mathbb{F}_d$ -graph $G = (V, E)$ is a simple loop-free $\mathbb{F}_d^*$ -weighted graph. We write $n := |V|$ for the number of vertices of G . We associate the graph with its *adjacency matrix*: the $V \times V$ matrix G , where $G_{u,v}$ stands for the weight of edge (u, v) , or 0 if there is no such edge.

Definition 2.2. An open $\mathbb{F}_d$ -graph (G, I, O) is a triple where G is an $\mathbb{F}_d$ -graph, and $I, O \subseteq V$ are sets of input and output vertices. We write $\bar{I} := V \setminus I$ and $\bar{O} := V \setminus O$ for the sets of non-input and non-output vertices respectively. Finally, we write $B := V \setminus (I \cup O)$ for the set of internal vertices (B for in-Between).

Definition 2.3. A labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) is a quadruple where (G, I, O) is an open $\mathbb{F}_d$ -graph and $\lambda: \bar{O} \rightarrow \mathbb{F}_d^2 \setminus \{(0, 0)\}$ is a *measurement labelling* such that for all $i \in I \setminus O$: $\lambda(i) = (0, *)$.

2.1 Graph states and the one-way model of measurement-based quantum computation

Like graphs in the qubit case, $\mathbb{F}_d$ -graphs can be used to define a class of quantum states on qudits called *graph states* [6, Section C]. These states are stabiliser states, meaning they are eigenstates with eigenvalue 1 of certain tensor products of the qudit Pauli matrices.

We first introduce some of the formalism of qudit stabiliser quantum mechanics. Let $\{|k\rangle\}_{k \in \mathbb{F}_d}$ be the computational basis of a qudit, then the Pauli-Z and X matrices are defined as $Z|k\rangle = \omega^k|k\rangle$ and $X|k\rangle = |k+1\rangle$, where $\omega = e^{\frac{2i\pi}{d}}$ and the addition inside the ket is modulo d . The qudit Hadamard operator performs the discrete Fourier transform of dimension d , it is defined as $H|k\rangle = \sum_{j \in \mathbb{F}_d} \omega^{jk}|j\rangle$. Unlike for qubits, the qudit Hadamard is not self inverse; instead it satisfies $H^4 = I$. It is straightforward to show that $Z^d = I = X^d$ and the gates satisfy $ZX = \omega XZ$, $HZH^{-1} = X^{-1}$, as well as $HXH^{-1} = Z$. The *Pauli group* on one qudit is generated by Z and X , its elements are of the form $\omega^a X^b Z^c$ for some $a, b, c \in \mathbb{F}_d$. A tensor product of single-qudit Pauli group elements is called a *Pauli product*. In the following, for any single-qudit unitary U whose eigenvalues are $\{\omega^k\}_{k \in \mathbb{F}_d}$, we will use $|k : U\rangle$ to denote the eigenstate of U with eigenvalue ω^k . In particular, $|k : Z\rangle = |k\rangle$ and $|k : X\rangle = H^{-1}|k\rangle$ for all $k \in \mathbb{F}_d$. We will also need an entangling two-qudit gate, the controlled-Z gate defined as $CZ|jk\rangle = \omega^{jk}|jk\rangle$. As for qubits, CZ gates applied to any (even overlapping) pairs of qudits commute.

In the following, we use subscripts to denote the qudit (or qudits) to which a gate is applied, e.g. $CZ_{u,v}$ acts on qudits u and v .

Definition 2.4. Suppose $G = (V, E)$ is an $\mathbb{F}_d$ -graph, then the associated *graph state* is given by $|G\rangle := \prod_{u,w \in V} CZ_{u,w}^{G_{u,w}} \bigotimes_{v \in V} |0 : X\rangle$.

Lemma 2.5 ([6]). Suppose $G = (V, E)$ is an $\mathbb{F}_d$ -graph, then the stabiliser group for the graph state $|G\rangle$ is generated by the Pauli products of the form $X_v \prod_{w \in V} Z_w^{G_{v,w}}$ for each $v \in V$.

In the one-way model of MBQC, a computation is performed via successive adaptive single-qudit measurements on a resource ($\mathbb{F}_d$)-graph state. Each measurement has a distinguished *desired outcome*, which is the one driving the computation in the desired direction. While individual measurements are non-deterministic, by restricting measurements to specific well-behaved measurement spaces, it is possible to compensate for a specific measurement having resulted in an undesired outcome through changing later measurements. In this way one can use measurements to perform a computation that is overall deterministic. The following presentation of the one-way model on qudits follows [8].

A projective measurement on a qudit consisting of measurement projectors $\{\pi_k\}_{k \in \mathbb{F}_d}$ can be identified with a unitary matrix $U = \sum_{k \in \mathbb{F}_d} \mu_k \pi_k$ for some set of eigenvalues μ_k . We will assume that the desired outcome is associated with eigenvalue 1, i.e. we consider only unitary representations of measurements that satisfy $U|\psi\rangle = |\psi\rangle$ for some $|\psi\rangle \in \mathbb{C}^d$ called the *fixpoint*.

Each single-qudit Pauli $X^a Z^b$ for some $a, b \in \mathbb{F}_d$ such that a, b are not both 0 determines a *measurement space*, which is denoted $\mathcal{M}(a, b)$ and which groups together measurements with shared characteristics. The measurement labelling λ of a labelled open $\mathbb{F}_d$ -graph (Definition 2.3) refers exactly to these measurement spaces. Then a measurement unitary U is in $\mathcal{M}(a, b)$ if and only if it satisfies $X^a Z^b U = \omega U X^a Z^b$, implying the following useful properties.

Proposition 2.6 ([8, Proposition 2]). If $U \in \mathcal{M}(a, b)$, then the spectrum of U is $\{\omega^m \mid m \in \mathbb{F}_d\}$, each eigenvalue has multiplicity 1, and U is special unitary. Denoting $|0 : U\rangle$ the fixpoint of U , then $|m : U\rangle = (X^a Z^b)^{-m} |0 : U\rangle$ is an eigenvector of U associated with eigenvalue ω^m .

If the measurement U yields the desired outcome, then the projection applied to the resource state is $|0 : U\rangle\langle 0 : U|$. If instead the measurement yields an undesired outcome, the projection is $|m : U\rangle\langle m : U|$ for some $m \in \mathbb{F}_d^*$. By the above proposition, the undesired projectors are linked to the desired projector via powers of $X^a Z^b$: $|0 : U\rangle\langle 0 : U| = (X^a Z^b)^m |m : U\rangle\langle m : U| (X^a Z^b)^{-m}$.

This is the basis for the adaptive procedure that can correct undesired measurement outcomes: Suppose the measurement on qudit v has resulted in the undesired outcome $m \in \mathbb{F}_d^*$. If there is a stabiliser of the underlying graph that acts on v as $(X^a Z^b)^m$ and otherwise acts non-trivially only on qudits that have not yet been measured, then this stabiliser can be used to correct the measurement on v . The Pauli operators of the correcting stabiliser that act on qudits other than v itself are called *by-products*. The assumption that non-trivial by-products arise only on unmeasured qudits implies that they can later be incorporated into the measurement on that qudit. Generalising from a single measurement to an entire computation, we need a partial order on the measurements that indicates which measurements happen before or after others, as well as a correcting stabiliser for each measured qubit; the partial order and the stabilisers need to be compatible to ensure that by-products affect only qudits that come later in the partial order. Some care also needs to be taken around treating qudits that act as inputs for the entire computation, meaning they are not prepared in the state $|0 : X\rangle$ but arrive in some arbitrary state and are entangled into the graph; as well as qudits that are not measured but remain as outputs at the end. These conditions were formalised in [8], we will introduce them now.

2.2 $\mathbb{F}_d$ -flow

We use [10, Lemma 4.3] as a definition of $\mathbb{F}_d$ -flow instead of the original version of the definition in [8, Definition 9], changing the name to ‘ $\mathbb{F}_d$ -flow’ with $\mathbb{F}_d$ instead of $\mathbb{Z}_d$ to emphasise that the underlying algebraic structure needs to be a finite field.

Definition 2.7 ($\mathbb{F}_d$ -flow [10, Lemma 4.3]). A labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) has $\mathbb{F}_d$ -flow if there exists a tuple $(C, \prec)$ where $C \in \mathbb{F}_d^{V \times V}$ is called the *correction matrix* and $\prec$ is a strict partial order on V , such that:

$$(CF) \quad \forall u \in \bar{O}. \lambda(u) = (C_{u,u}, (GC)_{u,u}),$$

$$(ZF) \quad C_{u,v} = 0 \text{ whenever } u \in I \text{ or } v \in O, \text{ and}$$

$$(OF) \quad \text{when the columns and rows of } G \text{ and } C \text{ are ordered according to any totalisation of } \prec, \text{ then } C \text{ and } GC \text{ are lower triangular.}$$

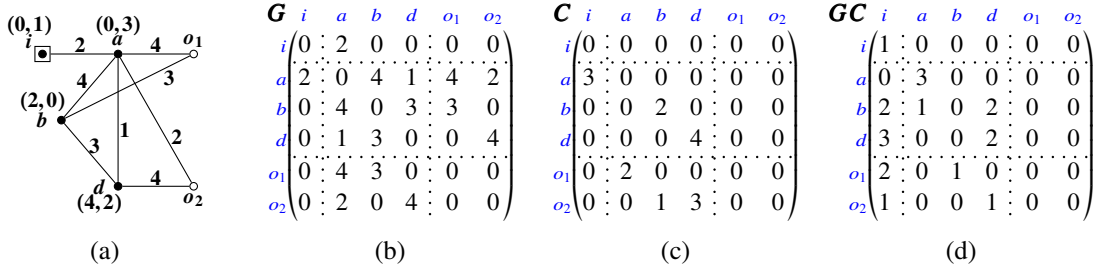


Figure 1: Running example of a labelled open $\mathbb{F}_d$ -graph with $d = 5$. **1a**: a labelled open $\mathbb{F}_d$ -graph with one input i and two outputs o_1 and o_2 . The measurement labels of non-outputs are given near the names of the vertices. Edge weights are denoted by numbers near the middles of the edges. **1b**: the corresponding adjacency matrix. The dotted lines subdivide input, internal, and output vertices for improved readability but otherwise play no role. **1c**: the corresponding correction matrix. This matrix is lower triangular. **1d**: the product of adjacency and correction matrices from **1b** and **1c** respectively. It is not lower triangular, however it becomes lower triangular when the vertices are ordered $i \prec a \prec d \prec b \prec o_1 \prec o_2$. Under this ordering, the correction matrix from **1c** remains lower triangular, hence the labelled open $\mathbb{F}_d$ -graph has $\mathbb{F}_d$ -flow.

The $\mathbb{F}_d$ -flow definition implicitly specifies the Pauli product $\bigotimes_{u \in V} X^{C_{u,v}} Z^{(GC)_{u,v}}$ for correcting the measurement on qudit v . It is straightforward to check that this is indeed a stabiliser of the underlying graph state. Indeed, the flow correction condition (CF) ensures that the Pauli by-products and the desired correction together form a stabiliser for each measured qudit. The zeroes condition (ZF) states that inputs cannot correct other vertices and that outputs do not need a correction. Finally, the order condition (OF) states that it is possible to correct measurements in a causal way. Out of the three conditions, the third one is clearly the most complicated. Not only is it the only condition in which the strict partial order $\prec$ is involved, the condition also considers any totalisation of this strict partial order.

A worked example presenting $\mathbb{F}_d$ -flow is given in Figure 1. The unweighted graph underlying this example is a modification of the graph in [3, Example 2.43]. Verifying that the relevant matrices are lower triangular with respect to some totalisation is a mildly tedious step that requires reordering rows and columns.

3 Focusing $\mathbb{F}_d$ -flow

If a labelled open $\mathbb{F}_d$ -graph has $\mathbb{F}_d$ -flow, this $\mathbb{F}_d$ -flow is not generally unique. For qubits, the notion of *focused gflow* (or its generalisation, focused Pauli flow) has been very useful: for example, focused flow is unique if there are equal numbers of inputs and outputs [29, 3, 39], it can be easier to work with for flow-preserving rewriting [17, 3, 5], and it enables a linear-algebraic formulation of the flow conditions, leading to more efficient flow-finding algorithms [29, 32]. In this context, ‘focusing’ means restricting the correction by-products that a qubit can receive, depending on its measurement label. We now generalise this idea to qudit $\mathbb{F}_d$ -flow. We use notation $(\lambda_X(v), \lambda_Z(v)) := \lambda(v)$.

Definition 3.1. An $\mathbb{F}_d$ -flow $(C, \prec)$ on a labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) is *focused* when:

(FX) for all $v \in \bar{O}$ such that $\lambda_X(v) = 0$, we have $(GC)_{v,u} = 0$ for any $u \in \bar{O} \setminus \{v\}$ and

(FZ) for all $v \in \bar{O}$ such that $\lambda_X(v) \neq 0$, we have $C_{v,u} = 0$ for any $u \in \bar{O} \setminus \{v\}$.

For example, the $\mathbb{F}_d$ -flow from Figure 1 is focused.

The first condition (FX) (F for focused) ensures that $(0, *)$ -measured vertices receive only by-products of the form X^a (with trivial Z component). All other vertices receive only by-products of the form Z^b (with trivial X -component), as captured by condition (FZ). This corresponds to focusing conditions for the qubit case: XY -measured vertices (now generalised to $(0, *)$) can only get an X by-product, while YZ and XZ (now generalised to $(*, 0)$ and $(*, *)$ respectively) can only get a Z by-product.

We show, that existence of $\mathbb{F}_d$ -flow is equivalent to existence of focused $\mathbb{F}_d$ -flow. The proof may be found in Appendix B.

Theorem 3.2. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph which has $\mathbb{F}_d$ -flow. Then there exists a focused $\mathbb{F}_d$ -flow on (G, I, O, λ) .

Later on, in Corollary 4.6, we will explain that focused $\mathbb{F}_d$ -flow can be viewed as canonical, in the sense that when $|I| = |O|$, then there exists a unique focused $\mathbb{F}_d$ -flow, up to extensions of the partial order.

4 Algebraic formulation of focused $\mathbb{F}_d$ -flow

The definition of $\mathbb{F}_d$ -flow gives linear algebraic conditions that a correction procedure must satisfy to enable deterministic MBQC. Yet, condition (O) is awkward as it asks for a specific property to hold for any totalisation of the partial order. We now refine the algebraic formulation, tailoring it specifically towards focused $\mathbb{F}_d$ -flow. This loses some generality – the new definition can no longer express non-focused $\mathbb{F}_d$ -flow – but we already proved that any labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow also has focused $\mathbb{F}_d$ -flow. The small restriction is therefore outweighed in many cases by the associated gains in simplicity and efficiency of avoiding the arbitrary totalisations of the partial order.

First, we reduce some redundancy. In Definition 2.7, the correction matrix C is a $V \times V$ matrix, but the condition (ZF) forces I -labelled rows and O -labelled columns of C to be identically zero. Thus, instead of considering the full $V \times V$ matrix C , it suffices to instead consider its $\bar{I} \times \bar{O}$ submatrix. Correspondingly, it suffices to consider a submatrix of the adjacency matrix; because we want to multiply the two submatrices, the shape of this second submatrix will be $\bar{O} \times \bar{I}$:

Definition 4.1. Given an open $\mathbb{F}_d$ graph (G, I, O) , the matrix $A_{(G, I, O)} := G_{\bar{O}}^{\bar{I}}$ is its *reduced adjacency matrix*.

The reduced adjacency matrix allows the following restatement of the $\mathbb{F}_d$ -flow definition, with a proof of equivalence of two definitions given in Appendix C.1.

Definition 4.2 ($\mathbb{F}_d$ -flow). A labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) has an $\mathbb{F}_d$ -flow $(C, \prec)$, where $C \in \mathbb{F}_d^{\bar{I} \times \bar{O}}$ is called a *reduced correction matrix* and $\prec$ is a strict partial order on $\bar{O}$ such that:

- (C) $\forall u \in \bar{O} \setminus I. \lambda(u) = (C_{u,u}, (AC)_{u,u})$ and $\forall u \in I \cap \bar{O}. \lambda_Z(u) = (AC)_{u,u}$
- (O) when the columns and rows of A and C are ordered according to any totalisation of $\prec$, then C and AC are lower triangular.

The reduced adjacency matrix and reduced correction matrix of the running example from Figure 1 are given in Figures 2a and 2b, respectively.

For the new formulation, instead of working directly with the (reduced) adjacency matrix of the labelled open $\mathbb{F}_d$ -graph, we will define two matrices (both of shape $\bar{O} \times \bar{I}$) that encode the measurement labels as well as relevant properties of the adjacency matrix. First, we introduce some relevant notation. We write $\delta_{a,b}$ for the Kronecker delta, i.e. $\delta_{a,b} = 1$ when $a = b$ and 0 otherwise. We write $\text{Id}_{\mathcal{R}}^C$ for a $\mathcal{R} \times C$ matrix where $(\text{Id}_{\mathcal{R}}^C)_{u,v} = \delta_{u,v}$ for all $u \in \mathcal{R}$ and $v \in C$. When $\mathcal{R} = C$, $\text{Id}_{\mathcal{R}}^C$ is the identity matrix.

When $|\mathcal{R}| = 1$ or $|\mathcal{C}| = 1$, then $\text{Id}_{\mathcal{R}}^{\mathcal{C}}$ is a row or column vector, respectively, which has zeroes everywhere except for the at most one entry in $\mathcal{R} \cap \mathcal{C}$. We use $\text{supp}(\vec{a})$ to denote the support of the vector $\vec{a}$, i.e. if $\mathcal{R}$ is the set of labels of the elements of $\vec{a}$, then $\text{supp}(\vec{a}) = \{r \in \mathcal{R} \mid \vec{a}_r \neq 0\}$.

Definition 4.3 (Flow-demand matrix). Given a labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) , the corresponding *flow-demand matrix* M is a $\bar{O} \times \bar{I}$ matrix where for $v \in \bar{O}, u \in \bar{I}$:

$$M_{v,u} = \begin{cases} \lambda_Z(v)^{-1} A_{v,u} & \text{if } \lambda_X(v) = 0 \\ \lambda_X(v)^{-1} \delta_{v,u} & \text{if } \lambda_X(v) \neq 0 \end{cases}$$

The expressions in the definition above are valid, in the sense that the inverse of $\lambda_Z(v)$ is used only when $\lambda_X(v) = 0$, in which case $\lambda_Z(v) \neq 0$ as $(\lambda_X(v), \lambda_Z(v)) = \lambda(v) \neq (0, 0)$ by Definition 2.3. Moreover, the required inverses exists due to $\mathbb{F}_d$ being a field. Similarly, the inverse of $\lambda_X(v)$ is only used when $\lambda_X(v) \neq 0$. Furthermore $A_{v,u}$ exists in the reduced adjacency matrix as $v \in \bar{O}$ and $u \in \bar{I}$. See Figure 2c for an example of the flow-demand matrix and Appendix E for pseudocode that constructs this matrix from a specification of the labelled open $\mathbb{F}_d$ -graph.

Definition 4.4 (Order-demand matrix). Given a labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) , the corresponding *order-demand matrix* N is an $\bar{O} \times \bar{I}$ matrix where for $v \in \bar{O}, u \in \bar{I}$: $N_{v,u} = \lambda_X(v) A_{v,u} - \lambda_Z(v) \delta_{v,u}$.

See Figure 2d for the order-demand matrix of the running example, and Appendix E for pseudocode that generates the order-demand matrix from the specification of a labelled open $\mathbb{F}_d$ -graph.

We can now state the main result of this section:

Theorem 4.5. Suppose $\Gamma = (G, I, O, \lambda)$ is a labelled open $\mathbb{F}_d$ -graph; let M be its flow-demand matrix and N its order-demand matrix. Then Γ has $\mathbb{F}_d$ -flow if and only if there exists a matrix C (of shape $\bar{I} \times \bar{O}$) such that $MC = \text{Id}_{\bar{O}}^{\bar{O}}$ and NC forms a (weighted) directed acyclic graph.

Here, the focusing conditions (FZ) and (FX) are fully captured by the requirement that $MC = \text{Id}_{\bar{O}}^{\bar{O}}$. The condition (C) is mostly, but not fully, captured by the requirement $MC = \text{Id}_{\bar{O}}^{\bar{O}}$. Part of (C) is instead captured by the property that the main diagonal of the product NC is identically 0, which follows from NC being a DAG. The condition (O) is fully captured by NC being a DAG. The proof of Theorem 4.5 is given in Appendix C. See Figure 2 for the application of this result to the running example from Figure 1.

When $|I| = |O|$, the flow-demand matrix is square and has at most one right inverse: the two-sided inverse. Therefore, there exists at most one focused $\mathbb{F}_d$ -flow (up to refinements of the partial order). Thus the focused flow uniqueness result known for qubits [29, 3, 39] generalises:

Corollary 4.6. Suppose $\Gamma = (G, I, O, \lambda)$ is a labelled open $\mathbb{F}_d$ -graph where $|I| = |O|$, then all focused $\mathbb{F}_d$ -flows on Γ share the same reduced correction matrix.

4.1 Focused vectors

In the qubit case, the kernel of the flow-demand matrix encodes a structure called *focused sets* [32, Section 3.4]. These sets, originally defined by Simmons [39, Definition 4.3], correspond to ‘do nothing’ stabilisers, i.e. stabilisers that do not help with correcting measurement by-products in any way. We extend the notion of focused sets to higher dimensions and recover the relationship to the kernel of the flow-demand matrix. As a corollary, since the kernel of a matrix forms a vector space and thus an Abelian group, we also recover the Simmons qubit result [39, Lemma B.8]. Relevant proofs are included in Appendix C.3.

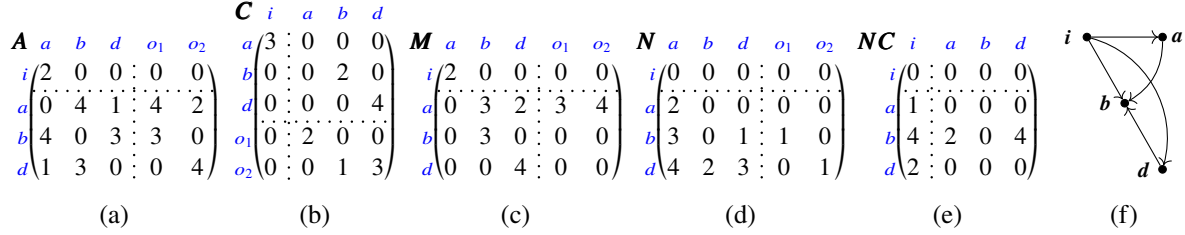


Figure 2: The algebraic formulation of the labelled open $\mathbb{F}_d$ -graph from Figure 1. 2a: the reduced adjacency matrix which is obtained from the matrix in 1b by removing output rows and input columns; the reduced correction matrix in 2b is obtained from the matrix in 1c by a transpose of this process. 2c: the flow-demand matrix, which has the reduced correction matrix in 2b as its right inverse. 2d: the order-demand matrix. 2e: the product of the order-demand matrix with the reduced correction matrix. This matrix corresponds to the relation $\prec_C$ in 2f, which is a DAG. Hence, the reduced correction matrix results in focused flow.

Definition 4.7 (Focused vector). Given a labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) , a column vector $\vec{c} \in \mathbb{F}_d^{\bar{I}}$ is *focused over* $S \subseteq \bar{O}$ if:

(Fs1) $\forall w \in S \cap \text{supp}(\vec{c}). \lambda_X(w) = 0$ and

(Fs2) $\forall w \in S \cap \text{supp}(A\vec{c}). \lambda_X(w) \neq 0$.

A vector focused over $\bar{O}$ is simply called *focused*. The set of all focused vectors for Γ is denoted $\mathfrak{F}_\Gamma$.

Focused vectors could also be treated as multisets with support contained in $\bar{I}$ and multiplicities given by elements of $\mathbb{F}_d$. We usually treat them as vectors, as that is most convenient for the proofs, but all interpretations are equivalent.

Theorem 4.8. Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let M be its flow-demand matrix. Then $\mathfrak{F}_\Gamma \cong \ker M_\Gamma$.

For example, the kernel of the flow-demand matrix M from Figure 2c is $\ker M = \text{span}((0, 0, 0, 1, 3)^T)$ and thus the only focused vectors of the labelled open $\mathbb{F}_d$ -graph in Figure 1a are those of form $k(0, 0, 0, 1, 3)^T$ for some $k \in \mathbb{F}_d$. Under the multiset interpretation, these consist of vertex o_1 with multiplicity k and vertex o_2 with multiplicity $3k$.

5 Flow-finding

With the new formulation of focused flow, we now generalise the flow-finding algorithm from [32] to qudits. In our previous work [32, Corollary 4.1], we presented algorithms for qubit flow which effectively solve $\mathbb{F}_d$ -flow-finding when $d = 2$, in the process solving the following linear algebra problem over the field $\mathbb{F}_2$ only²:

Inverse + DAG

Input: Two matrices M and N of matching shape $\mathcal{A} \times \mathcal{B}$ over $\mathbb{F}_d$.

Output: A $\mathcal{B} \times \mathcal{A}$ matrix C over $\mathbb{F}_d$ such that $MC = \text{Id}_{\mathcal{A}}$ and NC forms a directed acyclic $\mathbb{F}_d$ -weighted graph, or the message that no such C exists.

² Our previous work in [32] addresses the more general setting of Pauli flow, in which measurements in the eigenbasis of a Pauli operator can be accorded special treatment. However when Pauli measurements are not treated separately, Pauli flow collapses to gflow, the latter being exactly the $\mathbb{F}_d$ -flow where $d = 2$.

The structure of the qubit algorithms can be understood as follows: given a labelled open graph, construct the flow-demand and order-demand matrices. Next, solve **Inverse + DAG**. The qubit flow-finding algorithms do not assume anything about the structure of M and N after their initial construction. These algorithms can be adapted to solve **Inverse + DAG**, and thus $\mathbb{F}_d$ -flow-finding, for any d such that $\mathbb{F}_d$ forms a field. The parameter d does not have to be prime, but only a power of a prime. We obtain:

Theorem 5.1. There exists a $O(n^3)$ algorithm that, given a labelled open $\mathbb{F}_d$ -graph, outputs the corresponding $\mathbb{F}_d$ -flow, or gives a message that no $\mathbb{F}_d$ -flow exists.

For explanation of how to incorporate the computational complexity of finite field arithmetic into the complexity analysis, see Appendix D.

In the qubit case, we split flow-finding into two cases depending on whether $|I| = |O|$, i.e. on whether the input to **Inverse + DAG** is given by square matrices. When $|I| = |O|$, the algorithm for the qubit case applies to qudits without any changes. However, in the general case, the qubit algorithm must be adjusted. We include the pseudocode for $\mathbb{F}_d$ -flow-finding in the general case in Appendix E together with an explanation of where it differs from the qubit case. Rather than writing out what would be near identical proofs to those in [32], we provide a working example of finding flow for the labelled open $\mathbb{F}_d$ -graph from Figure 1, see Appendix F.

Speed-ups to matrix multiplication and inversion can be deployed when $|I| = |O|$, reducing the complexity from $O(n^3)$ to a possibly smaller exponent, see [32, Section 4.3.1] for details. We also obtain a lower bound by linking $\mathbb{F}_d$ -flow-finding to the problems of matrix multiplication and matrix inversion over a finite field, see Appendix G.

6 $\mathbb{F}_d$ -flow-preserving rewriting

So far, we have worked with a fixed labelled open $\mathbb{F}_d$ -graph in defining and finding $\mathbb{F}_d$ -flow, and relied only on the data contained in this labelled open $\mathbb{F}_d$ -graph. To implement a one-way computation, more information is needed, in particular it is necessary to specify a measurement operator (with its desired outcome) for each measured qudit. Given such an expression for an entire computation in the one-way model, one immediate question is about alternative ways of implementing the same linear operation, which may be more efficient, better suited to a specific hardware connectivity, or have other desired properties. Thus, in the qubit case, there is a thriving system of ‘flow-preserving rewrite rules’ [17, 3, 39, 27, 28, 5] which map some given one-way computation to a different computation implementing the same linear operation. These rules can be used to optimise quantum computational resources in the circuit or one-way models according to different metrics [40, 24], or even to obfuscate quantum computations so they can be delegated into the cloud while preserving security properties [12].

We will therefore now consider flow-preserving rewriting for $\mathbb{F}_d$ -flow.

6.1 Local scaling, local complementation, and pivot

Some relevant rewrite rules were already defined and proved flow-preserving by Booth [10, Section 5.1]. These rules are based on the two different types of operations on an $\mathbb{F}_d$ -weighted graph which translate to local Clifford equivalences of the corresponding qudit graph states [6, Section D]. The first of these operations is *local scaling*, which multiplies the weights of all edges incident on a chosen vertex w by some non-zero constant $\gamma \in \mathbb{F}_d^*$ and is usually denoted $G \overset{\gamma}{\circ} w$. The second operation is *local complementation*, which is also parametrised by a constant $\gamma \in \mathbb{F}_d$ and changes the weights of all edges among the neighbours of the chosen vertex w in a more complicated way, the resulting graph is denoted $G \overset{\gamma}{\star} w$. In both cases, the

graph state $|G\rangle$ corresponding to the initial graph can be transformed into the graph state corresponding to the final graph ($|G \circ^\gamma w\rangle$ or $|G \star^\gamma w\rangle$, respectively) by applying a product of single-qudit Clifford operations. As a result, both operations preserve the existence of $\mathbb{F}_d$ -flow, given certain updates to the labelled open $\mathbb{F}_d$ -graph [10, Propositions 5.5 and 5.8]; formal statements of this are given in Appendix H.1.

A *pivot*, a sequence of three local complementations about the endpoints of an edge, gives rise to a flow-preserving transformation on qubits and it interacts more smoothly with focused gflow and focused Pauli flow than individual local complementations [5, Lemma 5.3]. The pivot has been generalised to a family of pivot operations on qutrits [42, Theorem 3.5] and a qudit pivot-and-delete operation (where the endpoints of the edge are deleted after the pivot) [34, Lemma 10]. We work out the effects of a pivot operation (without deletion) for general prime-dimensional qudits in Appendix H.1 and summarise them here.

Proposition 6.1 (informal statement). Let G be an $\mathbb{F}_d$ -weighted graph, suppose $w, w' \in V$ satisfy $\epsilon := G_{w,w'} \neq 0$, and define $G \wedge ww'$ to be the graph after a pivot on the edge ww' . Let $F^{(\gamma)} := \sum_{j,k \in \mathbb{F}_d} \omega^{\gamma jk} |j\rangle\langle k|$ for any $\gamma \in \mathbb{F}_d^*$, then $|G \wedge ww'\rangle = F_w^{(-\epsilon)} F_{w'}^{(-\epsilon)} \prod_{v \in V} R_v^{2\epsilon^{-1} G_{v,w} G_{v,w'}} |G\rangle$.

We also argue that the above pivot operation can be implemented using three successive local complementations and at least one local scaling (the latter is needed to get a pivot operation that is symmetric under interchange of w and w'). This decomposition immediately implies that a pivot preserves the existence of $\mathbb{F}_d$ -flow.

6.2 Flow reversal

In the qubit case, for a labelled open graph with $|I| = |O|$, it is possible to reverse the direction of flow [29, 32] by swapping inputs with outputs (and assigning XY measurement labels to the former outputs, i.e. the new inputs). Moreover, there is a close relationship between the flow in one direction and the flow in the other direction: for the fragment consisting only of XY measurements, inputs and outputs, the correction matrix in the reverse direction is just the transpose of the original correction matrix, although this relationship is more complicated for the other measurement planes. We lift this result to qudits.

Definition 6.2. Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph which satisfies $|I| = |O|$. Then its *reverse labelled open $\mathbb{F}_d$ -graph* is $\Gamma' = (G, O, I, \lambda')$, where $\lambda'(v) = (0, 1)$ if $v \in O$ and $\lambda'(v) = \lambda(v)$ if $v \in B$.

Theorem 6.3 ($\mathbb{F}_d$ -flow reversibility). Let Γ be a labelled open $\mathbb{F}_d$ -graph which satisfies $|I| = |O|$. Let Γ' be the reverse of Γ . Then Γ has $\mathbb{F}_d$ -flow if and only if Γ' has $\mathbb{F}_d$ -flow.

The proof is in Appendix H.3; it is quite close to the qubit case, except that working over an arbitrary prime field requires extra care with subtraction (which over $\mathbb{F}_2$ is equivalent to addition).

6.3 Z-like removal and insertion

In a focused flow, certain kinds of measured vertices receive only by-products that are powers of Z , these vertices are called ‘Z-like’. Z-like measured qubits can be removed from one-way computations [17, 3, 39] and, under certain conditions, inserted into one-way computations [5] while preserving the existence of flow. We lift these flow-preserving qubit removal and insertion results to qudits; the corresponding proofs are presented in Appendix H.4.

For any $\mathbb{F}_d$ -graph G and any vertex v in G , we denote by $G - v$ the graph that results from removing v (and all its incident edges).

Definition 6.4. Let (G, I, O) be an open $\mathbb{F}_d$ -graph and let $S \in \mathbb{F}_d^V$ be any vector. Define the *open $\mathbb{F}_d$ -graph extended by v* as the open $\mathbb{F}_d$ -graph (G', I, O) obtained by inserting a new vertex v into (G, I, O) with its connections determined by S , i.e. $G' - v = G$ and $G'_{v,u} = S_u$ for all $u \in V$. Given (G, I, O, λ) and $\lambda'(v)$, we define the *labelled open $\mathbb{F}_d$ -graph extended by v with measurement label $\lambda'(v)$* as the labelled open $\mathbb{F}_d$ -graph (G', I, O, λ') , where (G', I, O) is the open $\mathbb{F}_d$ -graph extended by v , and $\lambda'(u) = \lambda(u)$ for all $u \in \bar{O}$.

From now on, we add a prime symbol for all notation referring to the (labelled) open $\mathbb{F}_d$ -graph extended by v ; this includes $G', \lambda', M', N', \bar{O}'$, and $\bar{I}'$.

Theorem 6.5 (Z-like deletion). Let $\Gamma' := (G', I, O, \lambda')$ be the labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) extended by v , where $\lambda'_X(v) \neq 0$, and suppose Γ' has $\mathbb{F}_d$ -flow. Then (G, I, O, λ) also has $\mathbb{F}_d$ -flow.

For Z-like insertion, we restrict to the case $|I| = |O|$ as it is exactly the uniqueness of focused $\mathbb{F}_d$ -flow in the case $|I| = |O|$ (cf. Corollary 4.6) that makes the insertion of Z-like vertices and the corresponding updates to the $\mathbb{F}_d$ -flow possible while performing fewer³ operations than would be required by the brute-force approach of inserting the vertex and running the full flow-finding procedure [5, page 109].

Theorem 6.6 (Z-like insertion). Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph with focused $\mathbb{F}_d$ -flow, where $|I| = |O|$. Further suppose that C, NC are known, where C is the reduced correction matrix guaranteed to exist by Theorem 4.5 and to be unique by Corollary 4.6; and N is the order-demand matrix of Γ . Let $\Gamma' = (G', I, O, \lambda')$ be the open $\mathbb{F}_d$ -graph extended by v via a known vector $S \in \mathbb{F}_d^V$. Then, in $O(n^2)$ time complexity, it is possible to determine the set of all $\lambda'(v)$ such that $\lambda'_X(v) \neq 0$ and (G', I, O, λ') has $\mathbb{F}_d$ -flow.

6.4 Generation of labelled open graphs with $\mathbb{F}_d$ -flow

Benchmarking flow-based algorithms on qubits, such as finding and circuit extraction, already requires generating large labelled open graphs with flow, and such labelled open graphs have also found application in the generation of ansätze for optimisation and machine learning [18]. For these applications, the flows should be *non-trivial*, i.e. the correction mechanism should not be obvious from the structure of the graphs alone. Since most labelled open graphs do not have flow⁴, it can be difficult to generate instances that have flow in a non-trivial way. For example, one way to generate large instances with flow on qubits is to translate quantum circuits into measurement patterns. However, this method results in diagrams with causal flow, where the correction mechanism is particularly simple: each row of the correction-matrix contains only a single non-zero value. Another approach for generating labelled open graphs with flow is to compose smaller instances with flow. However, this leads to a layered structure with limited connectivity. Based on a recent result of one of the authors for qubits [2], we now give an algorithmic method for generating instances with $\mathbb{F}_d$ -flow that can overcome these shortcomings: the algorithm has non-zero probability of returning *any* labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow, not only labelled open $\mathbb{F}_d$ -graphs of the restricted types outlined above.

Theorem 6.7. There exists an algorithm (presented in Appendix H.5) that takes a pair of integers n and k as input and it returns a random labelled open $\mathbb{F}_d$ -graph with $|V| = n$ and $|I| = |O| = k$ that has $\mathbb{F}_d$ -flow. Furthermore, for any labelled open $\mathbb{F}_d$ -graph Γ with $|V| = n$ and $|I| = |O| = k$ with $\mathbb{F}_d$ -flow, the algorithm has non-zero probability of returning Γ .

³ In fact: strictly fewer, unless the lower bound from Theorem G.2 is also an upper bound.

⁴ Informally, most flow-demand matrices will not have maximal row-rank and thus have no right inverse. In particular, a certain submatrix of the adjacency matrix needs to be invertible, but most graphs do not have this property [38].

The proof is in Appendix H.5. The method we use may not be very efficient, as it is based on attempts of Z-like insertions, which can show that insertion of a vertex with a fixed neighbourhood is simply impossible in an $\mathbb{F}_d$ -flow-preserving manner. However, the method is guaranteed to return cases with $\mathbb{F}_d$ -flow. Thus, a one-time generation could be run for an extended period to produce instances that can then be used for benchmarking flow algorithm implementations. This method is more elegant than the trivial approach of generating large labelled open graphs and running the flow-finding algorithm. We also expect it to be more practical for generating cases with $\mathbb{F}_d$ -flow, though we leave the last fact without a formal proof.

7 Conclusions and future directions

Building on the work of [8] about determinism in qudit MBQC, we developed a definition of focused $\mathbb{F}_d$ -flow. We showed that the existence of $\mathbb{F}_d$ -flow is equivalent to the existence of focused $\mathbb{F}_d$ -flow; the uniqueness of focused $\mathbb{F}_d$ -flow (at least when the numbers of inputs and outputs are equal) thus means focused $\mathbb{F}_d$ -flow can be considered canonical. We refined the algebraic formulation to allow a restatement of the $\mathbb{F}_d$ -flow definition using matrix multiplication and a check whether a certain directed graph is acyclic, thus avoiding the previous complicated verification of conditions related to the order of measurements. Utilising this new algebraic formulation, we obtained an $O(n^3)$ $\mathbb{F}_d$ -flow-finding algorithm, the same complexity as for qubits. Furthermore, we laid the groundwork for applications of $\mathbb{F}_d$ -flow in optimisation by deriving several new $\mathbb{F}_d$ -flow-preserving rules.

The most natural direction for future work is to apply our results in optimisation. All properties of flow and flow-preserving transformations used in [17] have now been generalised to higher dimensions suggesting that circuit optimisation procedure over qubits has a great potential to lift to qudits. Furthermore, our results also generalise flow results essential in refinements of this optimisation method, for instance in [26], albeit some results that do not concern flow remain to be generalised.

Another direction for the future work would be to develop a ‘causal $\mathbb{F}_d$ -flow’ analogous to [14]. While causal flow is not necessary for determinism in the qubit case, it corresponds more closely to the circuit model than gflow, which can be advantageous for certain applications [24]. Moreover causal flow can be found in sub-cubic time [15, 16]. Additionally, an extended version of causal flow plays a core role in the completeness result for flow-preserving rewriting on qubits [4], which could potentially be extended to qudits in the future.

In the qubit case, separate treatment of Pauli measurement leads to a different flow [11] with its own applications [39] and algebraic presentation [31, 32]; all while offering a broader picture of determinism than gflow [11, 30]. It would be interesting to investigate the existence of a Pauli flow for qudits.

Lastly, our techniques exploit the field structure of $\mathbb{F}_d$ for an improved algebraic presentation. In the present work on qudits and its predecessor [8], these fields are necessarily finite. There also exists a definition of MBQC and an associated flow on quantum systems described by continuous variables [9], which effectively works over the infinite field $\mathbb{R}$. We expect our algebraic approach could be adjusted and applied to this continuous variable flow, however the infinite field might lead to difficulties on the algorithmic side of this work. Care would be needed to choose a suitable representation of real numbers that permits efficient computation of relevant field operations.

Acknowledgements

We thank Robert Booth for helpful discussions about $\mathbb{F}_d$ -flow. PM is supported by Niedersächsisches Ministerium für Wissenschaft und Kultur. MB is supported by the Plan France 2030 through the PEPR integrated project EPiQ ANR-22-PETQ-0007 and the HQI platform ANR-22-PNCQ-0002.

References

- [1] Miriam Backens (2014): *The ZX-calculus Is Complete for Stabilizer Quantum Mechanics*. *New Journal of Physics* 16(9), p. 093021, doi:[10.1088/1367-2630/16/9/093021](https://doi.org/10.1088/1367-2630/16/9/093021). arXiv:[1307.7025](https://arxiv.org/abs/1307.7025).
- [2] Miriam Backens (2026): *Generating one-way computations with flow: flow-preserving rewriting that ignores the interpretation*. To appear.
- [3] 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](https://doi.org/10.22331/q-2021-03-25-421). arXiv:[2003.01664](https://arxiv.org/abs/2003.01664).
- [4] Miriam Backens & Simon Perdrix (2026): *Completeness for flow-preserving rewrite rules*. To appear.
- [5] Miriam Backens & Thomas Perez (2025): *Inserting Planar-Measured Qubits into MBQC Patterns while Preserving Flow*. *Electronic Proceedings in Theoretical Computer Science* 426, p. 100–126, doi:[10.4204/eptcs.426.4](https://doi.org/10.4204/eptcs.426.4).
- [6] Mohsen Bahrangiri & Salman Beigi (2006): *Graph States Under the Action of Local Clifford Group in Non-Binary Case*. arXiv:[quant-ph/0610267](https://arxiv.org/abs/quant-ph/0610267). arXiv:[quant-ph/0610267](https://arxiv.org/abs/quant-ph/0610267).
- [7] Robert I. Booth & Titouan Carette (2022): *Complete ZX-Calculi for the Stabiliser Fragment in Odd Prime Dimensions*. In Stefan Szeider, Robert Ganian & Alexandra Silva, editors: *47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 241, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 24:1–24:15, doi:[10.4230/LIPIcs.MFCS.2022.24](https://doi.org/10.4230/LIPIcs.MFCS.2022.24). Extended version at arXiv:[2204.12531](https://arxiv.org/abs/2204.12531).
- [8] Robert I. Booth, Aleks Kissinger, Damian Markham, Clément Meignant & Simon Perdrix (2023): *Outcome Determinism in Measurement-Based Quantum Computation with Qudits*. *Journal of Physics A: Mathematical and Theoretical* 56(11), p. 115303, doi:[10.1088/1751-8121/acbace](https://doi.org/10.1088/1751-8121/acbace). arXiv:[2109.13810](https://arxiv.org/abs/2109.13810).
- [9] Robert I. Booth & Damian Markham (2023): *Flow conditions for continuous variable measurement-based quantum computing*. *Quantum* 7, p. 1146, doi:[10.22331/q-2023-10-19-1146](https://doi.org/10.22331/q-2023-10-19-1146).
- [10] Robert Ivan Booth (2022): *Measurement-Based Quantum Computation beyond Qubits*. Ph.D. thesis, Sorbonne Université.
- [11] 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), p. 250, doi:[10.1088/1367-2630/9/8/250](https://doi.org/10.1088/1367-2630/9/8/250).
- [12] Shuxiang Cao (2023): *Multi-Agent Blind Quantum Computation without Universal Cluster States*. *New Journal of Physics* 25(10), p. 103028, doi:[10.1088/1367-2630/acfab6](https://doi.org/10.1088/1367-2630/acfab6).
- [13] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest & Clifford Stein (2022): *Introduction to Algorithms, Fourth Edition*. MIT Press.
- [14] Vincent Danos & Elham Kashefi (2006): *Determinism in the One-Way Model*. *Physical Review A* 74(5), p. 052310, doi:[10.1103/PhysRevA.74.052310](https://doi.org/10.1103/PhysRevA.74.052310).
- [15] N. de Beaudrap (2007): *A Complete Algorithm to Find Flows in the One-Way Measurement Model*, doi:[10.48550/arXiv.quant-ph/0603072](https://doi.org/10.48550/arXiv.quant-ph/0603072). arXiv:[quant-ph/0603072](https://arxiv.org/abs/quant-ph/0603072).
- [16] N. de Beaudrap (2008): *Finding Flows in the One-Way Measurement Model*. *Physical Review A* 77(2), p. 022328, doi:[10.1103/PhysRevA.77.022328](https://doi.org/10.1103/PhysRevA.77.022328).

- [17] 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](https://doi.org/10.22331/q-2020-06-04-279). arXiv:[1902.03178](https://arxiv.org/abs/1902.03178).
- [18] Tom Ewen, Ivica Turkalj, Patrick Holzer & Mark-Oliver Wolf (2025): *Application of ZX-calculus to quantum architecture search*. *Quantum Machine Intelligence* 7(1), p. 34, doi:[10.1007/s42484-025-00264-6](https://doi.org/10.1007/s42484-025-00264-6).
- [19] Sergey B. Gashkov & Igor S. Sergeev (2013): *Complexity of Computation in Finite Fields*. *Journal of Mathematical Sciences* 191(5), pp. 661–685, doi:[10.1007/s10958-013-1350-5](https://doi.org/10.1007/s10958-013-1350-5).
- [20] Vlad Gheorghiu (2014): *Standard form of qudit stabilizer groups*. *Physics Letters A* 378(5), pp. 505–509, doi:[10.1016/j.physleta.2013.12.009](https://doi.org/10.1016/j.physleta.2013.12.009).
- [21] David Harvey & Joris van der Hoeven (2021): *Integer Multiplication in Time $O(N \log n)$* . *Annals of Mathematics* 193(2), pp. 563–617, doi:[10.4007/annals.2021.193.2.4](https://doi.org/10.4007/annals.2021.193.2.4).
- [22] David Harvey & Joris van der Hoeven (2022): *Polynomial Multiplication over Finite Fields in Time $O(n \log n)$* . *J. ACM* 69(2), doi:[10.1145/3505584](https://doi.org/10.1145/3505584).
- [23] Luke E. Heyfron & Earl Campbell (2019): *A quantum compiler for qudits of prime dimension greater than 3*. arXiv:[1902.05634](https://arxiv.org/abs/1902.05634).
- [24] Calum Holker (2024): *Causal flow preserving optimisation of quantum circuits in the ZX-calculus*. arXiv:[2312.02793](https://arxiv.org/abs/2312.02793).
- [25] Erik Hostens, Jeroen Dehaene & Bart De Moor (2005): *Stabilizer states and Clifford operations for systems of arbitrary dimensions and modular arithmetic*. *Physical Review A* 71(4), p. 042315, doi:[10.1103/PhysRevA.71.042315](https://doi.org/10.1103/PhysRevA.71.042315).
- [26] Aleks Kissinger & John van de Wetering (2020): *Reducing T-count with the ZX-calculus*. *Physical Review A* 102(2), p. 022406, doi:[10.1103/PhysRevA.102.022406](https://doi.org/10.1103/PhysRevA.102.022406). arXiv:[1903.10477](https://arxiv.org/abs/1903.10477).
- [27] Tommy McElvanney & Miriam Backens (2023): *Complete Flow-Preserving Rewrite Rules for MBQC Patterns with Pauli Measurements*. *Electronic Proceedings in Theoretical Computer Science* 394, pp. 66–82, doi:[10.4204/EPTCS.394.5](https://doi.org/10.4204/EPTCS.394.5).
- [28] Tommy McElvanney & Miriam Backens (2023): *Flow-Preserving ZX-calculus Rewrite Rules for Optimisation and Obfuscation*. *Electronic Proceedings in Theoretical Computer Science* 384, pp. 203–219, doi:[10.4204/EPTCS.384.12](https://doi.org/10.4204/EPTCS.384.12). arXiv:[2304.08166](https://arxiv.org/abs/2304.08166).
- [29] Mehdi Mhalla, Mio Murao, Simon Perdrix, Masato Someya & Peter S. Turner (2014): *Which Graph States Are Useful for Quantum Information Processing?* In Dave Bacon, Miguel Martin-Delgado & Martin Roetteler, editors: *Theory of Quantum Computation, Communication, and Cryptography*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 174–187, doi:[10.1007/978-3-642-54429-3_12](https://doi.org/10.1007/978-3-642-54429-3_12).
- [30] Mehdi Mhalla, Simon Perdrix & Luc Sanselme (2025): *Shadow Pauli Flow: Characterising Determinism in MBQCs involving Pauli Measurements*, doi:[10.48550/arXiv.2207.09368](https://doi.org/10.48550/arXiv.2207.09368). ArXiv:[2207.09368](https://arxiv.org/abs/2207.09368) [quant-ph].
- [31] Piotr Mitosek (2024): *Pauli Flow on Open Graphs with Unknown Measurement Labels*. *Electronic Proceedings in Theoretical Computer Science* 406, pp. 117–136, doi:[10.4204/EPTCS.406.6](https://doi.org/10.4204/EPTCS.406.6). arXiv:[2408.06059](https://arxiv.org/abs/2408.06059).
- [32] Piotr Mitosek & Miriam Backens (2026): *An algebraic formulation of Pauli flow, leading to faster flow-finding algorithms*. *Journal of Physics A: Mathematical and Theoretical* 59(3), p. 035301, doi:[10.1088/1751-8121/ae2999](https://doi.org/10.1088/1751-8121/ae2999).
- [33] Stefano Paesani, Jacob F. F. Bulmer, Alex E. Jones, Raffaele Santagati & Anthony Laing (2021): *Scheme for Universal High-Dimensional Quantum Computation with Linear Optics*. *Physical Review Letters* 126(23), p. 230504, doi:[10.1103/PhysRevLett.126.230504](https://doi.org/10.1103/PhysRevLett.126.230504).
- [34] Boldizsár Poór, Robert I. Booth, Titouan Carette, John Van De Wetering & Lia Yeh (2023): *The Qupit Stabiliser ZX-travaganza: Simplified Axioms, Normal Forms and Graph-Theoretic Simplification*. *Electronic Proceedings in Theoretical Computer Science* 384, pp. 220–264, doi:[10.4204/EPTCS.384.13](https://doi.org/10.4204/EPTCS.384.13).
- [35] Dana Randall (1993): *Efficient Generation of Random Nonsingular Matrices*. *Random Structures & Algorithms* 4(1), pp. 111–118, doi:[10.1002/rsa.3240040108](https://doi.org/10.1002/rsa.3240040108).

- [36] Robert Raussendorf & Hans J. Briegel (2001): *A One-Way Quantum Computer*. *Physical Review Letters* 86(22), pp. 5188–5191, doi:[10.1103/PhysRevLett.86.5188](https://doi.org/10.1103/PhysRevLett.86.5188).
- [37] Alena Romanova & Wolfgang Dür (2026): *Measurement-based quantum computing with qudit stabilizer states*. *Quantum Science and Technology* 11(1), p. 015054, doi:[10.1088/2058-9565/ae3b6f](https://doi.org/10.1088/2058-9565/ae3b6f).
- [38] Irene Sciriha (2007): *A characterization of singular graphs*. *ELA. The Electronic Journal of Linear Algebra [electronic only]* 16, pp. 451–462. Available at <http://eudml.org/doc/129125>.
- [39] Will Simmons (2021): *Relating Measurement Patterns to Circuits via Pauli Flow*. *Electronic Proceedings in Theoretical Computer Science* 343, pp. 50–101, doi:[10.4204/EPTCS.343.4](https://doi.org/10.4204/EPTCS.343.4). arXiv:[2109.05654](https://arxiv.org/abs/2109.05654).
- [40] Korbinian Staudacher, Tobias Guggemos, Sophia Grundner-Culemann & Wolfgang Gehrke (2023): *Reducing 2-QuBit Gate Count for ZX-Calculus Based Quantum Circuit Optimization*. *Electronic Proceedings in Theoretical Computer Science* 394, pp. 29–45, doi:[10.4204/EPTCS.394.3](https://doi.org/10.4204/EPTCS.394.3).
- [41] Andrew Sutherland (2025): *Lecture Notes: Mathematics 18.783 – Elliptic Curves, Lecture #3*. Available at <https://math.mit.edu/classes/18.783/2025/LectureNotes3.pdf>. Accessed: 28th February 2025.
- [42] Alex Townsend-Teague & Konstantinos Meichanetzidis (2022): *Simplification Strategies for the Qutrit ZX-Calculus*, doi:[10.48550/arXiv.2103.06914](https://doi.org/10.48550/arXiv.2103.06914). arXiv:[2103.06914](https://arxiv.org/abs/2103.06914).
- [43] Shuai Yang, Lihao Xu, Guojing Tian & Xiaoming Sun (2025): *Quantum circuit synthesis with qudit phase gadget method*, doi:[10.48550/arXiv.2504.12710](https://doi.org/10.48550/arXiv.2504.12710).

A Comparison to the qubit case

Qubit MBQC and its flow properties are more widely studied and thus many readers might be familiar with qubit flow, but not qudit flow. For such readers, in this appendix, we explain how various definitions and results correspond to the qubit-case literature, i.e. to the case of $d = 2$. In particular, when $d = 2$, all matrices become binary matrices. From now on, we only focus on differences regarding notation, terminology, or provision of literature reference when $d = 2$. We skip over statements where $d = 2$ implies all changes on its own.

Background

- The $\mathbb{F}_d$ -graph from Definition 2.1 becomes a simple graph. In particular, it is no longer weighted.
- The codomain of the measurement labelling associated with a labelled open $\mathbb{F}_d$ -graph (Definition 2.3) codomain would become $\{(0, 1), (1, 0), (1, 1)\}$. However, all three measurement spaces are known in the literature under different names:
 - The $(0, 1)$ space becomes the XY -plane, i.e. the plane spanned by the X and Y eigenstates; this is the plane of the Bloch sphere perpendicular to the Z axis,
 - the $(1, 0)$ space becomes the YZ -plane defined analogously, and
 - the $(1, 1)$ space becomes the XZ -plane, again defined analogously.

The condition that all measured inputs must be measured in the $(0, *)$ space translates to a similar condition involving the XY -plane.

- The single qubit Paulis become X , Y , and Z only (no nontrivial powers needed), where $Y \propto XZ$.
- The $\mathbb{F}_d$ -flow from Definition 2.7 becomes the gflow of [11]; a detailed translation is explained in [8]. The translation requires mapping the correction matrix C to a correction function c by associating columns of C with set indicator functions, i.e. $c(v) = \{u \mid C_{u,v} = 1\}$.

Focusing $\mathbb{F}_d$ -flow

- The notion of focused $\mathbb{F}_d$ -flow collapses to that of focused gflow from [3].
- The existence of focused $\mathbb{F}_d$ -flow generalises the existence of focused gflow from [29, Section 3.1] and [3].

Algebraic interpretation of focused $\mathbb{F}_d$ -flow

- The reduced adjacency matrix from Definition 4.1 has been considered over qubits in [29, Definition 6].
- The definitions of the flow-demand matrix (Definition 4.3) and the order-demand matrix (Definition 4.4) become the similarly-named matrices over qubits from [32, Definition 3.4 and Definition 3.5].
- The new algebraic formulation from Theorem 4.5 becomes the main result from our previous work [32, Theorem 3.1]. In particular, NC is no longer a weighted graph.
- The uniqueness of focused $\mathbb{F}_d$ -flow when $|I| = |O|$ (Corollary 4.6), becomes the uniqueness of gflow under the same restriction from [3].
- The focused vectors of Definition 4.7 become the focused sets from [39, Definition 4.3]. Originally, focused sets were defined in the setting of Pauli flow, yet they apply just as well to gflow.
- Theorem 4.8 becomes [32, Theorem 3.23].

Flow-finding The correspondence between qudit $\mathbb{F}_d$ -flow-finding and qubit flow-finding has been carefully explained in Section 5 and its corresponding appendices.

$\mathbb{F}_d$ -flow-preserving rewriting

- Regarding the qubit equivalents of the operations in Section 6.1, local scaling is trivial and local complementations are not parametrised. Unlike for qudits, there are actually two variants of local complementation on a qubit, which differ by a non-trivial graph stabiliser; conversely, one variant corresponds to three successive applications of the other about the same qubit. The definition of the qubit pivot uses both variants. Flow preservation of local complementation and pivoting for gflow was proved in [3, Lemma 3.1 and Corollary 3.3].
- Flow reversal in Theorem 6.3 translates to [32, Theorem 3.19].
- The notion of ‘Z-like measurements’ from Section 6.3 captures YZ- and XZ-measured vertices only, since we are not treating Pauli vertices separately and hence there are no Z measurements.
- Z-like removal from Theorem 6.5 corresponds to [3, Lemma 3.4].
- Z-like insertion from Theorem 6.6 corresponds to results from [5, Section 4 and Section 5].
- Generation of instances with $\mathbb{F}_d$ -flow from Theorem 6.7 translates to a result about generating MBQC patterns with different types of flow from the upcoming paper [2].

B Proving focusing

To prove Theorem 3.2, we first prove that a labelled open $\mathbb{F}_d$ -graph has focused $\mathbb{F}_d$ -flow if and only if it has $\mathbb{F}_d$ -flow, by giving a procedure that allows any $\mathbb{F}_d$ -flow to be modified in a stepwise fashion until it is focused, with each step being a valid $\mathbb{F}_d$ -flow. This procedure generalises [3, Section 3.3] to the qudit case. First, we specify a definition of focusing that applies to a single correction set and a single vertex.

Definition B.1. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow $(C, \prec)$, and let $u, v \in V$ be two distinct vertices, i.e. $u \neq v$. We say the correction $C_{*,u}$ is focused with respect to v if the following two statements hold:

(FXI) $v \in O \vee (C_{v,v} = 0 \implies (GC)_{v,u} = 0)$, and

(FZI) $C_{v,v} \neq 0 \implies C_{v,u} = 0$.

Observation B.2. An $\mathbb{F}_d$ -flow is focused according to Definition 3.1 if and only if (FXI) and (FZI) hold for all distinct pairs $u, v \in V$. Indeed, if Definition B.1 holds for all distinct pairs of vertices u, v , then Definition 3.1 is immediate.

For the converse, note that if $v \in O$, then $C_{v,v} = 0$ by (ZF), so both (FXI) and (FZI) are satisfied for any $C_{*,u}$. Similarly, if $(C_{v,u}, (GC)_{v,u}) = (0, 0)$, then (FXI) and (FZI) are satisfied for $C_{*,u}$ and v . In particular, if $u \in O$, then $C_{w,u} = 0$ for all $w \in V$ by (ZF); this then also implies $(GC)_{v,u} = \sum_{w \in V} G_{v,w} C_{w,u} = 0$.

Recall that if $v \in \bar{O}$, then $\lambda_X(v) = C_{v,v}$ by (CF). Thus, if an $\mathbb{F}_d$ -flow $(C, \prec)$ is focused according to Definition 3.1, then (FXI) and (FZI) hold for all distinct pairs $u, v \in \bar{O}$. Moreover, the two conditions hold whenever at least one of u, v is in O by the argument above. Thus, the two conditions hold for all pairs of distinct vertices.

We now show how to focus a flow in a step-wise fashion. The first argument is to show the construction that focuses a new correction set with respect to some chosen vertex, without breaking focusing conditions that already hold for other vertices.

Lemma B.3. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow $(C, \prec)$, and let $u, v \in V$ be two distinct vertices, i.e. $u \neq v$. Then we can construct an $\mathbb{F}_d$ -flow $(C', \prec)$ on (G, I, O, λ) such that $C'_{*,u}$ is focused with respect to v . Moreover, the construction ensures that if there exists a vertex $v' \neq v$ such that $C_{*,v}$ is focused with respect to v' , then, for any $w \in V$, $C'_{*,w}$ is focused with respect to v' if and only if $C_{*,w}$ is focused with respect to v' .

Proof. By Observation B.2, (FXI) and (FZI) are satisfied if at least one of u, v is an output, or if $(C_{v,u}, (GC)_{v,u}) = (0, 0)$; in those cases we are done by taking $C' = C$ and the second part of the lemma is trivial. Hence it only remains to consider the case where $u, v \in \bar{O}$ and $(C_{v,u}, (GC)_{v,u}) \neq (0, 0)$; by (OF) this implies $u \prec v$. We distinguish cases according to whether or not $C_{v,v}$ is 0.

- Suppose $C_{v,v} = \lambda_X(v) = 0$, then (FZI) is trivially satisfied. If $(GC)_{v,u} = 0$, (FXI) is also satisfied and we are done by taking $C' = C$.

Otherwise, note that by (CF) and Definition 2.3 we have $\lambda_Z(v) = (GC)_{v,v} \neq 0$. Define a new correction matrix C' as

$$C'_{w,w'} = \begin{cases} C_{w,w'} & \text{if } w' \neq u \\ C_{w,u} + \frac{d - (GC)_{v,u}}{\lambda_Z(v)} C_{w,v} & \text{if } w' = u. \end{cases}$$

Then by linearity:

$$(GC')_{w,w'} = \begin{cases} (GC)_{w,w'} & \text{if } w' \neq u \\ (GC)_{w,u} + \frac{d - (GC)_{v,u}}{\lambda_Z(v)} (GC)_{w,v} & \text{if } w' = u. \end{cases}$$

In particular, $(GC')_{v,u} = (GC)_{v,u} + \frac{d-(GC)_{v,u}}{\lambda_Z(v)} \lambda_Z(v) = 0$ over $\mathbb{F}_d$, meaning (FXI) is satisfied. Furthermore, $(C', \prec)$ satisfies Definition 2.7:

- The only column of C that has changed is $C_{*,u}$. We have $C_{u,v} = 0 = (GC)_{u,v}$ by (OF) since $(GC)_{v,u} \neq 0$; thus $(C'_{u,u}, (GC')_{u,u}) = (C_{u,u}, (GC)_{u,u})$ and (CF) is satisfied.
- If $w' \in O$, then $w' \neq u$ and thus $C'_{w,w'} = C_{w,w'}$. Now $C_{w,w'} = 0$ by (ZF) applied to C , so the desired property holds for C' as well. Similarly, if $w \in I$, then $C_{w,u} = C_{w,v} = 0$ by (ZF) applied to C and thus $C'_{w,u} = 0$. Hence C' satisfies (ZF).
- Suppose $C'_{w,u} \neq 0$ for some w , then $C_{w,u} \neq 0$ or $C_{w,v} \neq 0$. In the former case, $u \prec w$ and in the latter case $u \prec v \prec w$. The argument is analogous for GC' , thus (OF) is satisfied.

Hence $(C', \prec)$ is a valid $\mathbb{F}_d$ -flow.

- Now suppose instead $C_{v,v} = \lambda_X(v) \neq 0$, then (FXI) is trivially satisfied. If $C_{v,u} = 0$, then (FZI) is also satisfied and we are done. Otherwise, define a new correction matrix C' as

$$C'_{w,w'} = \begin{cases} C_{w,w'} & \text{if } w' \neq u \\ C_{w,u} + \frac{d-C_{v,u}}{\lambda_X(v)} C_{w,v} & \text{if } w' = u. \end{cases}$$

Then $C_{v,u} = C_{v,u} + \frac{d-C_{v,u}}{\lambda_X(v)} \lambda_X(v) = 0$ over $\mathbb{F}_d$, meaning (FZI) is satisfied. The argument that $(C', \prec)$ is a valid $\mathbb{F}_d$ -flow is analogous to the one in the previous case.

We have shown how to satisfy (FXI) and (FZI) for the pair u, v . Now suppose there exists $v' \neq v$ such that $C_{*,v}$ is focused with respect to v' . Assume without loss of generality that $v' \in \bar{O}$, otherwise (FXI) and (FZI) are trivial by Observation B.2. From the argument that $(C', \prec)$ is a valid $\mathbb{F}_d$ -flow at the end of the first case, we know that $C'_{v',v'} = C_{v',v'}$. Now, for any $w \in V$, we have $C'_{v',w} = C_{v',w} + \ell C_{v',v}$, where

$$\ell = \begin{cases} 0 & \text{if } w \neq u \\ \frac{d-(GC)_{v,u}}{\lambda_Z(v)} & \text{if } w = u \text{ and } \lambda_X(v) = C_{v,v} = 0 \\ \frac{d-C_{v,u}}{\lambda_X(v)} & \text{if } w = u \text{ and } \lambda_X(v) = C_{v,v} \neq 0. \end{cases}$$

As u, v are fixed, the value of ℓ depends only on w . Furthermore, by linearity, $(GC')_{v',w} = (GC')_{v',w} + \ell(GC')_{v',v}$. Thus:

- If $\lambda_X(v') = C_{v',v'} = 0$, then (FZI) is trivial. Moreover, $C_{*,v}$ being focused with respect to v' implies $C_{v',v} = 0$. Thus, for all $w \in V$, we have $C'_{v',w} = C_{v',w} + \ell C_{v',v} = C_{v',w}$, so $C'_{*,w'}$ is focused with respect to v' if and only if $C_{*,w}$ is.
- If $C_{v',v'} \neq 0$, then (FXI) is trivial. Moreover, $C_{*,v}$ being focused with respect to v' implies $(GC)_{v',v} = 0$. Thus, for all $w \in V$, we have $(GC')_{v',w} = (GC)_{v',w} + \ell(GC)_{v',v} = (GC)_{v',w}$, so again $C'_{*,w}$ is focused with respect to v' if and only if $C_{*,w}$ is.

This completes the proof. $\square$

We can now prove Theorem 3.2:

Proof of Theorem 3.2. By Observation B.2, instead of working with Definition 3.1, we may consider Definition B.1 for all pairs of distinct $u, v \in V$. A focused $\mathbb{F}_d$ -flow can thus be focused using the following procedure.

Let $<$ be any totalisation of $\prec$; for simplicity we will from now on identify V with the set $[n] := \{1, 2, \dots, n\}$, where $n = |V|$. We will prove the following property by structural induction on k ranging from 1 to n :

There exists an $\mathbb{F}_d$ -flow $(C', \prec)$ such that for all $v \in [n] \setminus [n-k]$ and all $u \neq v$, $C'_{*,u}$ is focused with respect to v .

The base case is $k = 1$, and the property holds for $C' := C$ since $C_{u,n} = 0 = (GC)_{u,n}$ for any $u < n$ by (OF). Now suppose the property holds for some k such that $1 \leq k < n$ and denote the corresponding $\mathbb{F}_d$ -flow by $(C, \prec)$. Consider what happens for $k+1$. Let $a := n - (k+1)$ and define the following sequence of correction matrices: $C^{(0)} := C$ and for $j \in [a-1]$,

$$C_{u,v}^{(j)} := \begin{cases} C_{u,v}^{(j-1)} & \text{if } v \neq j \\ C_{u,j}^{(j-1)} + \frac{d - (GC^{(j-1)})_{a,j}}{(GC^{(j-1)})_{a,a}} C_{u,a}^{(j-1)} & \text{if } v = j \text{ and } C_{a,a}^{(j-1)} = 0 \\ C_{u,j}^{(j-1)} + \frac{d - C_{a,j}^{(j-1)}}{C_{a,a}^{(j-1)}} C_{u,a}^{(j-1)} & \text{if } v = j \text{ and } C_{a,a}^{(j-1)} \neq 0. \end{cases}$$

Note that if $C_{*,j}^{(j-1)}$ is already focused with respect to a , then $C^{(j)} = C^{(j-1)}$. Then by Lemma B.3, $C_{*,j}^{(j)}$ is focused with respect to a . Moreover, by the same lemma, $C_{*,b}^{(j)}$ is focused with respect to a for all $b \in [j]$, and together with the induction hypothesis, for any $j \in [a-1]$, any $v > a$, and any $u \neq v$, we have that $C_{*,u}^{(j)}$ is focused with respect to v . Finally, $C_{*,u}^{(a-1)}$ is focused with respect to a for all $u > a$ since (OF) implies that $C_{a,u}^{(a-1)} = (GC^{(a-1)})_{a,u} = 0$. Thus the property holds for $k+1$ with correction matrix $C' = C^{(a-1)}$.

This means the property holds for $k = n$. The final $\mathbb{F}_d$ -flow $(C', \prec)$ has the property that for all $v \in V$ and all $u \in V \setminus \{v\}$, the correction $C'_{*,u}$ is focused with respect to v . By Observation B.2, this means $(C', \prec)$ satisfies Definition 3.1, i.e. it is focused. $\square$

C Proofs regarding algebraic formulation

The main result shown in this appendix is Theorem 4.5, but we also prove some related results here. In Subsection C.1, we lay ground for the main proof. Next, in Subsection C.2, we provide detailed proof of Theorem 4.5. Lastly, in Subsection C.3, we prove some properties of the focused vectors that are associated with the new algebraic formulation.

C.1 Preliminary simplifications

Before moving to the main proof, we perform some initial simplification by showing that the $\mathbb{F}_d$ -flow definition can be restricted to operate only on the reduced adjacency matrix rather than the full matrix. Moreover, we define a special relation induced by the correction matrix, which is sufficient to determine whether that correction matrix is compatible with any $\mathbb{F}_d$ -flow, thus eliminating the need to consider arbitrary partial orders.

Note that Condition (C) from Definition 4.2 can also be simplified to $\forall u \in \bar{O}. \lambda(u) = (C_{u,u}, (AC)_{u,u})$ under the convention that non-existent entries of a matrix are treated as 0s. Such a non-existent entry appears e.g. for $C_{u,u}$ with $u \in I$. We will now show that the simplified $\mathbb{F}_d$ -flow definition is equivalent to the original one.

Lemma C.1. There exists $\mathbb{F}_d$ -flow according to Definition 2.7 if and only if there exists $\mathbb{F}_d$ -flow according to Definition 4.2.

Proof. $(\Rightarrow)$: suppose there exists an $\mathbb{F}_d$ -flow $(C, \prec)$ on the labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) according to Definition 2.7. Let C' be the reduced correction matrix obtained from C and let $\prec'$ be the restriction

of $\prec$ to an order on $\bar{O}$ only. This restriction is immediately a strict partial order. Now, from (ZF) we know $C_{i,v} = 0$ for all $i \in I$ and $C_{v,o} = 0$ for all $o \in O$ (with $v \in V$ arbitrary), which implies C' contains all non-zero entries of C . As all the non-zero entries are preserved, we furthermore have $(AC)_{u,v} = (GC)_{u,v}$ for all $u \in \bar{O}$ and $v \in \bar{I}$. Therefore, condition (OF) implies (O), and (CF) implies (C). As a consequence, (G, I, O, λ) has $\mathbb{F}_d$ -flow $(C', \prec')$ according to Definition 4.2.

($\Leftarrow$): suppose there exists an $\mathbb{F}_d$ -flow $(C', \prec')$ on the labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) according to Definition 4.2. Let C be the extension of C' to a matrix of shape $V \times V$ by filling all missing entries with 0: this immediately makes C satisfy (ZF). Let $\prec$ be the extension of $\prec'$ to an order on V by adding relations $u \prec o$ for all $u \in \bar{O}$ and $o \in O$. This process makes $\prec$ a strict partial order. Similar to the proof above, $(AC)_{u,v} = (GC)_{u,v}$ for all $u \in \bar{O}$ and $v \in \bar{I}$ as once again C' contains all non-zero entries of C . Therefore condition (C) implies (CF). The matrices C' and AC' are lower-triangular according to any totalisation of $\prec'$, hence C and GC are lower triangular according to any totalisation of $\prec$, because $\prec' \subset \prec$ and rows and columns of C not featured in C' are identically 0. In other words, condition (O) implies (OF). Thus (G, I, O, λ) has $\mathbb{F}_d$ -flow $(C, \prec)$ according to Definition 2.7. $\square$

The definition of focused $\mathbb{F}_d$ -flow (cf. Definition 3.1) is compatible with both $\mathbb{F}_d$ -flow definitions. Thus, also, the existence of focused $\mathbb{F}_d$ -flow with respect to one $\mathbb{F}_d$ -flow definition is equivalent to the existence of focused $\mathbb{F}_d$ -flow with respect to the other $\mathbb{F}_d$ -flow definition:

Corollary C.2. There exists focused $\mathbb{F}_d$ -flow according to Definition 2.7 if and only if there exists focused $\mathbb{F}_d$ -flow according to Definition 4.2.

The usual flow definitions simultaneously describe the correction function and the strict partial order. Yet the conditions (FX) and (FZ) do not depend on the strict partial order, but only on the correction matrix. Thus, it makes sense to introduce the notion of a ‘focused correction matrix’, dropping the partial order from the definition:

Definition C.3 (Focused correction matrix). Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and C a reduced correction matrix. We say that C is *focused* if it satisfies (FX) and (FZ).

Definition C.4 (Extensive correction matrix). Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let C be a reduced correction matrix on Γ . We call C *extensive* when there exists a strict partial order $\prec$ such that $(C, \prec)$ is an $\mathbb{F}_d$ -flow on Γ . We call C *focused extensive* when there exists $\prec$ such that $(C, \prec)$ is a focused $\mathbb{F}_d$ -flow on Γ .

When determining whether a correction matrix is extensive, one must determine whether there exists a strict partial order that satisfies condition (O) from Definition 4.2. It turns out that, for any correction matrix, there exists a unique relation that determines extensivity.

Definition C.5 (Induced relation). Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix on Γ . The *induced relation* $\prec_C$ is the coarsest relation on $\bar{O}$ such that for any pair of distinct $u, v \in \bar{O}$ we have $u \prec_C v$ if and only if at least one of the following hold:

- (O1) $(AC)_{v,u} \neq 0$, or
- (O2) $C_{v,u} \neq 0$.

When $\prec_C$ is an acyclic relation, we denote the transitive closure of $\prec_C$ as $\prec_C$.

The entries of the product AC do not depend on any ordering of the vertices, and thus the above definition is sound. The relation $\prec_C$ captures all information regarding which off-diagonal entries of AC are non-zero, motivating the following lemma:

Lemma C.6 (Induced order containment). Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let $(C, \prec)$ be an $\mathbb{F}_d$ -flow. Then $\prec_C \subseteq \prec$.

Proof. Let $\prec'$ be any totalisation of $\prec$. By Definition 4.2, both AC and C must be lower triangular with respect to the ordering $\prec'$. Suppose that $u \prec_C v$. By Definition C.5, $(AC)_{v,u} \neq 0$ or $C_{v,u} \neq 0$. Now if $v \prec' u$, then either AC or C would not be lower triangular, a contradiction. Since $\prec'$ is a total order and $\neg(v \prec' u)$, we get that necessarily $u \prec' v$. Repeating for all elements of $\prec_C$, we find that $\prec_C \subseteq \prec'$. This inclusion must hold for all totalisations $\prec'$ of $\prec$ and thus $\prec_C$ forms a subset of the intersection of all such totalisations. This intersection is precisely $\prec$, ending the proof. $\square$

We now have all the parts needed to prove that an extensive correction matrix determines a valid $\mathbb{F}_d$ -flow.

Theorem C.7 (Extending a correction matrix to an $\mathbb{F}_d$ -flow). Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix on Γ . Then C is (focused) extensive if and only if $(C, \prec_C)$ is a (focused) $\mathbb{F}_d$ -flow.

Proof. $(\Rightarrow)$: Suppose that C is extensive. Then there exists a strict partial order $\prec$ on $\bar{O}$ such that $(C, \prec)$ is an $\mathbb{F}_d$ -flow. By Lemma C.6, $\prec_C \subseteq \prec$. Thus, since $\prec$ is a strict partial order, $\prec_C$ is also a strict partial order. Since C is extensive, it must satisfy condition (C) (which does not depend on the partial order). Similarly, if C is focused extensive, it must necessarily satisfy conditions (FX) and (FZ), which are also independent of the partial order. It remains to show that $(C, \prec_C)$ satisfies condition (O). Suppose for a contradiction that (O) fails for $\prec_C$. Then, for some totalisation $\prec'_C$ of $\prec_C$, we must have $v \prec'_C u$ and either $C_{v,u} \neq 0$ or $(AC)_{v,u} \neq 0$. However, either of $C_{v,u}$ or $(AC)_{v,u}$ being non-zero implies $u \prec_C v$ and thus $u \prec'_C v$. This contradicts the assumption $v \prec'_C u$. Thus $(C, \prec_C)$ satisfies (O) and hence forms an $\mathbb{F}_d$ -flow, which moreover is focused if and only if C is focused extensive.

$(\Leftarrow)$: Immediate from the definitions. $\square$

Theorem C.7 states that in order to determine whether C is (focused) extensive, it is sufficient to check whether $(C, \prec_C)$ forms a (focused) $\mathbb{F}_d$ -flow. In fact, based on the proof, a slightly stronger property holds. If C satisfies condition (C) (and conditions (FX) and (FZ)) and $\prec_C$ is a strict partial order, then $(C, \prec_C)$ is a (focused) $\mathbb{F}_d$ -flow, effectively skipping verification of condition (O) apart from ensuring that $\prec_C$ is a strict partial order. This allows us to avoid considering arbitrary totalisations of $\prec_C$ when verifying the $\mathbb{F}_d$ -flow conditions. Furthermore, since $\prec_C$ is a transitive closure of $\prec_C$, $\prec_C$ is a strict partial order if and only if $\prec_C$ forms a directed acyclic graph. Together, this amounts to the following simpler algebraic formulation of $\mathbb{F}_d$ -flow:

Theorem C.8. Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph and let C be any correction matrix on Γ . Then C is extensive if and only if C satisfies condition (C) and $\prec_C$ is a strict partial order. Assuming the above holds, C is focused extensive if and only if it satisfies conditions (FX) and (FZ).

C.2 Main proof

We now move on to proving the main result, Theorem 4.5. The first step is to split condition (C) into several smaller parts that will be encoded in different parts of the algebraic $\mathbb{F}_d$ -flow definition:

Observation C.9. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix of shape $\bar{I} \times \bar{O}$. Define the following conditions for all $v \in \bar{O}$:

$$(C1) \quad \lambda_X(v) = 0 \rightarrow (AC)_{v,v} = \lambda_Z(v)$$

$$(C2) \quad \lambda_X(v) \neq 0 \rightarrow C_{v,v} = \lambda_X(v)$$

$$(C3) \quad \lambda_X(v) = 0 \rightarrow C_{v,v} = 0$$

$$(C4) \quad \lambda_X(v) \neq 0 \rightarrow (AC)_{v,v} = \lambda_Z(v)$$

Then, (C) is equivalent to the four above conditions, as $\lambda_X(v) = C_{v,v}$ is captured by (C2) and (C3), while $\lambda_Z(v) = (AC)_{v,v}$ is captured by (C1) and (C4).

The goal is to show that $MC = \text{Id}_{\bar{O}}$ corresponds to (C1), (C2), (FX), and (FZ); while NC being a DAG corresponds to (C3), (C4), and the condition that $\prec_C$ is a strict partial order. We split the proof of this into two lemmata:

Lemma C.10. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix of shape $\bar{I} \times \bar{O}$. Then $MC = \text{Id}_{\bar{O}}$ if and only if C satisfies (C1), (C2), (FX), and (FZ).

Proof. First, consider some relevant properties of the entries of M in relation to the entries of A and C . Let $v \in \bar{O}$ and $u \in \bar{I}$, then by Definition 4.3:

$$M_{v,u} = \begin{cases} \lambda_Z(v)^{-1} A_{v,u} & \text{if } \lambda_X(v) = 0 \\ \lambda_X(v)^{-1} \delta_{v,u} & \text{if } \lambda_X(v) \neq 0. \end{cases}$$

Let $w \in \bar{O}$ and consider two cases depending on $\lambda_X(v)$:

1. Suppose $\lambda_X(v) = 0$ and thus $M_{v,u} = \lambda_Z(v)^{-1} A_{v,u}$. Multiplying by $C_{u,w}$, we get:

$$M_{v,u} C_{u,w} = \lambda_Z(v)^{-1} A_{v,u} C_{u,w}$$

and hence by linearity:

$$(MC)_{v,w} = \lambda_Z(v)^{-1} (AC)_{v,w} \quad (1)$$

2. Suppose $\lambda_X(v) \neq 0$ and thus $M_{v,u} = \lambda_X(v)^{-1} \delta_{v,u}$. When $\lambda_X(v) \neq 0$, then necessarily $v \notin I$ and thus M has a v -column. Multiplying $M_{v,u}$ by $C_{u,w}$, we get:

$$M_{v,u} C_{u,w} = \lambda_X(v)^{-1} \delta_{v,u} C_{u,w}$$

which equals 0 unless $v = u$ when it equals $\lambda_X(v)^{-1} C_{v,w}$. Using this fact, we find that:

$$(MC)_{v,w} = \sum_{u \in \bar{I}} M_{v,u} C_{u,w} = \lambda_X(v)^{-1} C_{v,w} \quad (2)$$

Now we can split the proof into two directions.

($\Rightarrow$) : Suppose $MC = \text{Id}_{\bar{O}}$; this implies $(MC)_{v,v} = 1$ and $(MC)_{v,w} = 0$ for all $w \in \bar{O} \setminus \{v\}$. The following hold:

1. For all v such that $\lambda_X(v) = 0$, by Equation (1), we have $1 = (MC)_{v,v} = \lambda_Z(v)^{-1} (AC)_{v,v}$. Thus $(AC)_{v,v} = \lambda_Z(v)$, i.e. (C1) holds.
2. Furthermore, for all v such that $\lambda_X(v) = 0$ and for all w such that $v \neq w$, by Equation (1) we have $0 = (MC)_{v,w} = \lambda_Z(v)^{-1} (AC)_{v,w}$. Since $\lambda_Z(v) \neq 0$ (as $\lambda(v) \neq (0,0)$), this implies $(AC)_{v,w} = 0$ and thus (FX) holds.
3. For all v such that $\lambda_X(v) \neq 0$, by Equation (2), we have $1 = (MC)_{v,v} = \lambda_X(v)^{-1} C_{v,v}$. Thus $C_{v,v} = \lambda_X(v)$, i.e. (C2) holds.

4. Finally, for all v such that $\lambda_X(v) \neq 0$ and for all w such that $v \neq w$, by Equation 2 we have $0 = (MC)_{v,w} = \lambda_X(v)^{-1}C_{v,w}$. This implies $C_{v,w} = 0$ and thus (FZ) holds.

Hence the desired properties are all satisfied.

($\Leftarrow$) : Suppose the four properties hold. We must show that $(MC)_{v,v} = 1$ and $(MC)_{v,w} = 0$ for all $v \neq w \in \bar{O}$. Observe that all four points in the ($\Rightarrow$) part of the proof are reversible. Therefore $(MC)_{v,v} = 1$ follows from reversing the two first points of the ($\Rightarrow$) part of the proof and $(MC)_{v,w} = 0$ follows from reversing the latter two points of the ($\Rightarrow$) part of the proof. Then $MC = \text{Id}_{\bar{O}}$ as desired. $\square$

The proof for the product NC being a DAG follows the same structure. We split it up into three parts for simplicity. Before proving those lemmata, first note that for any $v, w \in \bar{O}$, by Definition 4.4:

$$N_{v,u} = \lambda_X(v)A_{v,u} - \lambda_Z(v)\delta_{v,u}$$

for any $u \in \bar{I}$. Since $\delta_{v,u} = 0$ unless $u = v$, we get the following:

$$(NC)_{v,w} = \sum_{u \in \bar{I}} N_{v,u}C_{u,w} = \left(\sum_{u \in \bar{I}} \lambda_X(v)A_{v,u}C_{u,w} \right) - \lambda_Z(v)C_{v,w} = \lambda_X(v)(AC)_{v,w} - \lambda_Z(v)C_{v,w} \quad (3)$$

where we use the convention $C_{v,w} = 0$ for $v \in I$ to make sense of the final expression despite the rows of NC being labelled by $\bar{O}$ and the rows of C being labelled by $\bar{I}$.

Lemma C.11. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix of shape $\bar{I} \times \bar{O}$ satisfying (C1), (C2), (FX), and (FZ). Then the main diagonal of NC is identically zero if and only if (C3) and (C4) hold.

Proof. Fix $v = w \in \bar{O}$ in (3) and consider two cases depending on whether $\lambda_X(v)$ is zero:

1. Suppose $\lambda_X(v) = 0$. Then $(NC)_{v,v} = -\lambda_Z(v)C_{v,v}$ which equals 0 if and only if $C_{v,v} = 0$. Considering all v with $\lambda_X(v) = 0$, we obtain equivalence of $(NC)_{v,v} = 0$ to (C3).
2. Suppose $\lambda_X(v) \neq 0$. By the assumption that C satisfies (C2), we have $C_{v,v} = \lambda_X(v)$. Then:

$$(NC)_{v,v} = \lambda_X(v)(AC)_{v,v} - \lambda_Z(v)C_{v,v} = \lambda_X(v)(AC)_{v,v} - \lambda_Z(v)\lambda_X(v) = \lambda_X(v)((AC)_{v,v} - \lambda_Z(v))$$

which equals 0 if and only if $(AC)_{v,v} = \lambda_Z(v)$. Considering all v with $\lambda_X \neq 0$, we obtain equivalence of $(NC)_{v,v} = 0$ to (C4).

Thus, the above two cases imply that the main diagonal of NC is identically zero if and only if (C3) and (C4) hold. $\square$

Lemma C.12. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix of shape $\bar{I} \times \bar{O}$ satisfying (C1), (C2), (FX), and (FZ). Then the off-diagonal part of NC corresponds to $\prec_C$ in the sense that, for any pair of distinct $v, w \in \bar{O}$, we have $w \prec_C v$ if and only if $(NC)_{v,w} \neq 0$.

Proof. Let $v, w \in \bar{O}$ be distinct. We show that $w \prec_C v$ if and only if $(NC)_{v,w} \neq 0$.

($\Rightarrow$): Suppose that $w \prec_C v$. As before, we consider two cases depending on whether $\lambda_X(v)$ is zero:

1. Suppose $\lambda_X(v) = 0$. By the assumption that C satisfies (FX), we have $(AC)_{v,w} = 0$. Thus the condition of (O1) does not hold. Ergo the condition of (O2) necessarily applies, leading to $C_{v,w} \neq 0$. Therefore (3) becomes:

$$(NC)_{v,w} = \lambda_X(v)(AC)_{v,w} - \lambda_Z(v)C_{v,w} = -\lambda_Z(v)C_{v,w} \neq 0$$

2. Suppose $\lambda_X(v) \neq 0$. By the assumption that C satisfies (FZ), we have $C_{v,w} = 0$. Thus the condition of (O2) does not apply. Ergo the condition of (O1) necessarily applies, leading to $(AC)_{v,w} \neq 0$. Therefore (3) becomes:

$$(NC)_{v,w} = \lambda_X(v)(AC)_{v,w} - \lambda_Z(v)C_{v,w} = \lambda_X(v)(AC)_{v,w} \neq 0$$

In both cases, $(NC)_{v,w} \neq 0$ as desired.

($\Leftarrow$): Suppose that $(NC)_{v,w} \neq 0$, which means:

$$\lambda_X(v)(AC)_{v,w} - \lambda_Z(v)C_{v,w} \neq 0$$

If both $(AC)_{v,w} = 0$ and $C_{v,w} = 0$, the above cannot hold. Thus at least one of conditions (O1) or (O2) applies, and we get $w \prec_C v$.

Therefore the off-diagonal part of NC corresponds to $\prec_C$. $\square$

Lemma C.13. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let C be a correction matrix of shape $\bar{I} \times \bar{O}$ satisfying (C1), (C2), (FX), and (FZ). Then NC forms a (weighted) directed acyclic graph if and only if C satisfies (C3), (C4), and $\prec_C$ is a strict partial order.

Proof. Let $v, w \in \bar{O}$. We combine the previous two lemmata to get the full proof.

($\Rightarrow$): Under the assumption that NC forms a (weighted) DAG, by Lemma C.12, we get that $\prec_C$ is a DAG and thus $\prec_C$ is a strict partial order. Furthermore, NC being a DAG implies $(NC)_{v,v} = 0$ for all $v \in \bar{O}$ and thus, by Lemma C.11, conditions (C3) and (C4) both hold.

($\Leftarrow$): (C3) and (C4) imply that $(NC)_{v,v} = 0$ by Lemma C.11 and $\prec_C$ being a strict partial order implies that $\prec_C$ is a DAG. The latter implies that the off-diagonal part of NC is a DAG by Lemma C.12; by combining the two parts the entire matrix NC forms a DAG. $\square$

We can now prove the main result.

Proof of Theorem 4.5. ($\Rightarrow$): suppose that Γ has $\mathbb{F}_d$ -flow. Then by Theorem 3.2 it has a focused $\mathbb{F}_d$ -flow $(C, \prec)$ and by Theorem C.7 $(C, \prec_C)$ is also a focused flow. Hence, by Theorem C.8, C satisfies (C), (FX), (FZ), and $\prec_C$ is a strict partial order. By Observation C.9, C satisfies (C1), (C2), (C3), and (C4). Therefore, by Lemma C.10 we get $MC = \text{Id}_{\bar{O}}$, and by Lemma C.13 we get that NC forms a (weighted) DAG.

($\Leftarrow$): all steps in the ($\Rightarrow$) part of the proof follow from if and only if statements and thus the above argument can be reversed. $\square$

C.3 Proofs regarding focused vectors

Before proving Theorem 4.8, we first show that given $\vec{a} \in \mathbb{F}_d^{\bar{I}}$, one can find a maximal set over which $\vec{a}$ is focused (cf. Definition 4.7) by considering the product $M\vec{a}$, where M is the flow-demand matrix. Note that the definition of focused vectors involves the reduced adjacency matrix A , whereas we now use the flow-demand matrix; this change is justified in the following lemma.

Lemma C.14. Let (G, I, O, λ) be a labelled open $\mathbb{F}_d$ -graph and let $\vec{a} \in \mathbb{F}_d^{\bar{I}}$ be a vector. Then, $\vec{a}$ is focused over $\bar{O} \setminus \text{supp}(M\vec{a})$ and $\vec{a}$ is not focused over any larger subset of the non-outputs.

Proof. Let $F_{\vec{a}} = \bar{O} \setminus \text{supp}(M\vec{a})$. By inspecting conditions (Fs1) and (Fs2), for any sets $S, S' \subseteq \bar{O}$, the vector $\vec{a}$ is simultaneously focused over S and focused over S' if and only if it is focused over $S \cup S'$. Therefore, it is sufficient to show that for all $v \in \bar{O}$: $v \in F_{\vec{a}}$ if and only if $\vec{a}$ is focused over $\{v\}$.

($\Rightarrow$): Let $v \in F_{\vec{a}}$. Then $v \notin \text{supp}(M\vec{a})$, so $(M\vec{a})_v = 0$. We consider two cases depending on $\lambda_X(v)$, following a similar proof structure as in Lemma C.10:

1. Suppose $\lambda_X(v) = 0$, then:

$$0 = (M\vec{a})_v = \sum_{u \in \bar{I}} M_{v,u} \vec{a}_u = \lambda_Z(v)^{-1} \sum_{u \in \bar{I}} A_{v,u} \vec{a}_u = \lambda_Z(v)^{-1} (A\vec{a})_v$$

Since $\lambda_X(v) = 0$, necessarily $\lambda_Z(v) \neq 0$. This implies $(A\vec{a})_v = 0$, i.e. $v \notin \text{supp}(A\vec{a})$. By contraposition: if $v \in \text{supp}(A\vec{a})$, then $\lambda_X(v) \neq 0$, i.e. (Fs2) holds.

2. Suppose $\lambda_X(v) \neq 0$. Since $M_{v,u} = 0$ for $v \neq u$, we have:

$$0 = (M\vec{a})_v = \sum_{u \in \bar{I}} M_{v,u} \vec{a}_u = \lambda_X(v)^{-1} \vec{a}_v$$

Then as $\lambda_X(v) \neq 0$, we have $\vec{a}_v = 0$, i.e. $v \notin \text{supp}(\vec{a})$. Again by contraposition: if $v \in \text{supp}(\vec{a})$, then $\lambda_X(v) = 0$, meaning (Fs1) holds.

Hence $\vec{a}$ satisfies conditions (Fs1) and (Fs2) over the set $\{v\}$ as required.

($\Leftarrow$): Let $\vec{a}$ be focused over $\{v\}$. Once again, we consider two cases on $\lambda_X(v)$:

1. Suppose $\lambda_X(v) = 0$, then by (Fs2): $v \notin \text{supp}(A\vec{a})$, i.e. $(A\vec{a})_v = 0$. Following the same sequence of equalities as in the first case of the ($\Rightarrow$) part of the proof, we find $(M\vec{a})_v = 0$.
2. Suppose $\lambda_X(v) \neq 0$, then by (Fs1): $v \notin \text{supp}(\vec{a})$, i.e. $(\vec{a})_v = 0$. Following the same sequence of equalities as in the second case of the ($\Rightarrow$) part of the proof, we again find $(M\vec{a})_v = 0$.

Therefore, $(M\vec{a})_v = 0$, meaning $v \notin \text{supp}(M\vec{a})$, ergo $v \in F_{\vec{a}}$, which ends the proof. $\square$

Using the above lemma, we find that the kernel of the flow-demand matrix forms is exactly the set of focused vectors.

Proof of Theorem 4.8. By Lemma C.14, a vector $\vec{a} \in \mathbb{F}_d^{\bar{I}}$ is focused over $\bar{O} \setminus \text{supp}(M\vec{a})$. Now, suppose $\vec{a} \in \mathfrak{F}_\Gamma$. Then $\vec{a}$ is focused over $\bar{O}$. Hence $\bar{O} = \bar{O} \setminus \text{supp}(M\vec{a})$. But $M\vec{a}$ is a vector in $\mathbb{F}_d^{\bar{O}}$, thus $\text{supp}(M\vec{a}) = \emptyset$, i.e. $\vec{a} \in \ker M$. All these steps are reversible, which leads to $\mathfrak{F}_\Gamma \cong \ker M_\Gamma$. $\square$

D Finite field arithmetic

In finite fields, the complexity of arithmetic operations (by which we mean addition, negation, multiplication, and inversion) depends on the order d of the field [19]. In main part of the paper, we consider d to be fixed (or at least bounded above), i.e. we treat the cost of arithmetic operations in $\mathbb{F}_d$ as $O(1)$. In general, the cost of such operations can be expressed as $\Theta(\text{poly}(\log d))$, i.e. it is polynomial in $\log d$. The representation of an element of a finite field is possible in space $\Theta(\log d)$. Addition and negation take $\Theta(\log d)$ time, but the cost for multiplication and inversion is slightly higher. It is known that $O(\log d \log \log d)$ time suffices for multiplication and $O(\log d \log \log^2 d)$ for inversion [19, 21, 22]. The result from [22] is critical here and is conditional on a widely believed hypothesis regarding prime numbers; this requirement may be omitted for fields with $d = p^\ell$ where ℓ is small (and p is prime) [41,

Theorem 3.35 and Theorem 3.41]). In particular, the stated complexity holds unconditionally for prime fields. However, the methods with the best known complexity bounds are impractical for small d . Yet again, for bounded d , we can treat these costs as $O(1)$.

The exact complexity results, i.e. without fixing d , can be derived from the ones stated in the main part of the paper by including the cost of arithmetic operations. For the time complexity, that means increasing lower bounds by $\log d$ factors, and increasing upper bounds by $O(\log d \log \log^2 d)$. The memory complexity increases by $\Theta(\log d)$, e.g. our $\mathbb{F}_d$ -flow-finding method requires $O(n^2 \log d)$ memory.

E Pseudocode

Algorithm 1 Construction of flow and order-demand matrices

Returns the flow-demand matrix of a given labelled open $\mathbb{F}_d$ -graph

```

1: procedure FLOW-DEMANDMATRIX( $G, I, O, \lambda$ )
2:   initialise  $\bar{O} \times \bar{I}$  matrix  $M$ 
3:   for  $v \in \bar{O}$ 
4:     for  $u \in \bar{I}$   $\triangleright$  Loop over all entries of  $M$ 
5:       if  $\lambda_X(v) = 0$ 
6:          $M_{v,u} = \lambda(Z)^{-1} G_{v,u}$ 
7:       else
8:          $M_{v,u} = \lambda(X)^{-1} \delta_{v,u}$ 
9:   return  $M$ 

```

Returns the order-demand matrix of a given labelled open $\mathbb{F}_d$ -graph

```

10: procedure ORDER-DEMANDMATRIX( $G, I, O, \lambda$ )
11:   initialise  $\bar{O} \times \bar{I}$  matrix  $N$ 
12:   for  $v \in \bar{O}$ 
13:     for  $u \in \bar{I}$   $\triangleright$  Loop over all entries of  $N$ 
14:        $N_{v,u} = \lambda_X(v) G_{v,u} - \lambda_Z(v) \delta_{v,u}$ 
15:   return  $N$ 

```

The pseudocode for the general $\mathbb{F}_d$ -flow-finding algorithm (which works even if $|I| \neq |O|$) then differs from the corresponding qubit algorithm in [32, Appendix B] on only a few lines:

- On line 25, we must look for non-zero values rather than values exactly equal to 1. (In the $d = 2$ case, the only possible non-zero value is 1.)
- On lines 28, 29, and 35, instead of simply adding a row to another row, we must add a row multiplied by some value such that the addition will cancel an undesired dependency in the second row. (In the $d = 2$ case, the multiplicity is always 1.)
- On a new line 36, we must multiply the row by a suitable value so that its pivot becomes 1. (In the $d = 2$ case, a pivot always necessarily equals 1.)

Furthermore, while line 22 matches the previous (qubit) algorithm, we point out that the linear system comprises equations of the following form, where val corresponds to the value from the second block:

$$a_1 x_1 + \cdots + a_{n_O - n_I} x_{n_O - n_I} + val = 0$$

and not:

$$a_1x_1 + \dots + a_{n_O-n_I}x_{n_O-n_I} = val$$

It means, that the values encoded in the matrix $\mathcal{P}$ (initialised on line 15) are effectively negations of the values one would obtain if interpreting the system in the usual way. (When $d = 2$, negation leaves elements invariant.)

Algorithm 2 Finding flow in the general case

Checks if a labelled open $\mathbb{F}_d$ -graph $\Gamma = (G, I, O, \lambda)$ has $\mathbb{F}_d$ -flow and, if it exists, returns the focused flow.

```

1: procedure FINDFLOWGENERAL( $G, I, O, \lambda$ )
2:    $M = \text{FLOW-DEMANDMATRIX}(G, I, O, \lambda)$ 
3:    $N = \text{ORDER-DEMANDMATRIX}(G, I, O, \lambda)$ 
4:   if  $\text{rank } M \neq n - n_O$ 
5:     return 'NO FLOW EXISTS'
6:    $C_0 = \text{any right inverse of } M$ 
7:    $F = \text{matrix with columns forming basis of } \ker M$ 
8:    $C' = [C_0 \mid F]$ 
9:    $N_{\mathcal{B}} = NC'$ 
10:   $N_L = \text{first } n - n_O \text{ columns of } N_{\mathcal{B}}$ 
11:   $N_R = \text{last } n_O - n_I \text{ columns of } N_{\mathcal{B}}$ 
12:   $K_{ILS}, K_{LS} = [N_R \mid N_L \mid Id_{\bar{O}}]$   $\triangleright$  Two independent copies of the same linear system
 $\triangleright$  From now, we refer to the three parts of  $K_{ILS}$  and  $K_{LS}$  as the first block, the second block, and the third block
13:  bring first block of  $K_{LS}$  into row echelon form
14:   $S = \emptyset$ 
15:  initialize  $(n_O - n_I) \times (n - n_O)$  matrix  $\mathcal{P}$   $\triangleright$  Columns of  $\mathcal{P}$  correspond to non-outputs  $\bar{O}$ 
16:  while  $S \neq \bar{O}$  :
17:    find the first row  $r_z$  in  $K_{LS}$  whose first  $n_O - n_I$  entries equal 0
18:     $L = \{v \in \bar{O} \setminus S \mid \text{column } v \text{ in the first } K_{LS} \text{ block has all entries from row } r_z \text{ on equal to } 0\}$ 
19:    if  $L = \emptyset$ 
20:      return 'NO FLOW EXISTS'
21:    for  $v \in L$  :
22:       $\mathcal{P}_{*,v} = \text{SOLVELINEARSYSTEM}(\text{first block of } K_{LS}, v \text{ column in second block of } K_{LS})$ 
23:    for  $v \in L$  :
24:       $S = S \cup \{v\}$ 
25:       $R = [r \mid r \text{ is a row whose intersection with the } v \text{ column in the third block of } K_{LS} \text{ is } \neq 0]$ 
26:       $r_{last} = \text{last element of } R$ 
27:      for  $r \in R[-1]$   $\triangleright$  Iterate over all but last element
28:         $\text{add row } r_{last} \text{ of } K_{LS} \text{ with correct multiplicity to row } r \text{ in } K_{LS} \text{ to remove dependency}$ 
29:         $\text{add row } v \text{ of } K_{ILS} \text{ with correct multiplicity to row } r_{last} \text{ in } K_{LS} \text{ to remove dependency}$ 
30:        for  $r \in \text{rows of } K_{LS} \text{ except } r_{last}$  :  $\triangleright$  Iterate from top
31:          if row  $r$  of  $K_{LS}$  has first  $n_O - n_I$  entries 0
32:            break
33:           $y = \text{column of the leading 1 in row } r \text{ of } K_{LS}$ 
34:          if intersection of  $r_{last}$  and column  $y$  in  $K_{LS}$  is  $\neq 0$ 
35:             $\text{add row } r \text{ with correct multiplicity to } r_{last} \text{ in } K_{LS} \text{ to remove dependency}$ 

```

```

36: |      | multiply  $r_{last}$  by a correct value that makes the first non-zero entry of  $r_{last}$  in the first
    |      | block equal 1
37: |      | swap  $r_{last}$  with other rows to bring first block of  $K_{LS}$  back into row echelon form
38: |  $C^{\mathcal{B}} = \left[ \frac{Id_{\hat{O}}}{\hat{\varphi}} \right]$ 
39: | return  $(C'C^{\mathcal{B}}, NC'C^{\mathcal{B}})$ 

```

F Run of flow-finding algorithm

We now provide a detailed overview of how the pseudocode from Appendix E runs on the labelled open $\mathbb{F}_d$ -graph from Figure 1.

Construction of the linear system Firstly, in lines 2 and 3, the flow-demand and order-demand matrices are created, these are the same matrices as presented in Figures 2c and 2d. Since $\text{rank } M = 4$, the check in line 4 passes. Next, in lines 6, 7, and 8 three further matrices are created: any right inverse C' of M , a basis F of $\ker M$, and the concatenation C_0 of these two. For example:

$$C_0 = [C' \mid F] = \begin{array}{c} \begin{array}{c} i \\ a \\ b \\ d \\ o_1 \\ o_2 \end{array} \begin{array}{c} a \\ b \\ d \\ 2 \\ 1 \end{array} \begin{array}{c} a \\ b \\ d \\ 2 \\ 0 \end{array} \begin{array}{c} b \\ d \\ 4 \\ 1 \\ 3 \end{array} \begin{array}{c} d \\ 0 \\ 0 \\ 1 \\ 1 \end{array} \begin{array}{c} F_1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 3 \end{array} \end{array}$$

where we use F_1 as the label of the single column in F . Next, in line 9, we compute the product of N and C' , denoted $N_{\mathcal{B}}$:

$$N_{\mathcal{B}} = \begin{array}{c} \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} a \\ b \\ d \\ 3 \end{array} \begin{array}{c} a \\ b \\ d \\ 0 \end{array} \begin{array}{c} b \\ d \\ 2 \\ 0 \end{array} \begin{array}{c} d \\ 0 \\ 0 \\ 3 \end{array} \begin{array}{c} F_1 \\ 0 \\ 0 \\ 3 \end{array} \end{array}$$

In lines 10 and 11, we break $N_{\mathcal{B}}$ into two submatrices N_L and N_R , and use them to obtain two (initially identical) copies K_{LS} and K_{ILS} of the following linear system. We give primed labels to columns from the third block, to distinguish them from the corresponding columns in the second block:

$$K_{ILS} = K_{LS} = \begin{array}{c} \begin{array}{c} F_1 \\ i \\ a \\ b \\ d \end{array} \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} a \\ b \\ d \\ 3 \end{array} \begin{array}{c} a \\ b \\ d \\ 0 \end{array} \begin{array}{c} b \\ d \\ 2 \\ 0 \end{array} \begin{array}{c} d \\ 0 \\ 0 \\ 3 \end{array} \begin{array}{c} i' \\ a' \\ b' \\ d' \end{array} \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \end{array} \end{array} \quad (4)$$

Next, in line 13, we perform Gaussian elimination on the first block of K_{LS} , i.e. on the first column. The independent copy K_{ILS} remains unchanged. The Gaussian elimination involves adding the third row to the fourth row with multiplicity 2 so that the original pivot of the fourth block becomes 0. After that, we swap the first and third rows. We drop the previous row labels, as they no longer make sense after performing row operations. (Nevertheless, the relationship of each modified row to the original vertex-labelled rows continues to be encoded in the third block.) We obtain the following updated K_{LS}

linear system:

$$K_{LS} = \begin{array}{c} F_1 \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \left(\begin{array}{c|cccc|cccc} i & a & b & d & i' & a' & b' & d' \\ \hline 1 & 1 & 2 & 4 & 0 & 0 & 0 & 1 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 4 & 0 & 3 & 0 & 2 & 1 \end{array} \right) \quad (5)$$

We initialise S and $\mathcal{P}$ in lines 14 and 15 to:

$$\mathcal{P} = F_1 \left(\begin{array}{c} i & a & b & d \\ * & * & * & * \end{array} \right) \quad S = \emptyset$$

First run of the main loop We go into the main loop in line 16. During the first run, we identify that vertex b can be solved (line 18). This is because the entries of column b in the second block of the matrix in (5) are all 0 from the second row onwards. Therefore $L = \{b\}$. In line 22, we identify the solution of the following linear system with a single equation (arising from the first block of K_{LS} together with column b), where we name the unique variable f_1 :

$$1f_1 + 4 = 0$$

As the running example is over $\mathbb{F}_5$, $f_1 = 1$ is the only solution:

$$\mathcal{P} = F_1 \left(\begin{array}{c} i & a & b & d \\ * & * & 1 & * \end{array} \right)$$

Now, we must transform the system of linear equations to the form it would have had if b had never been included (this is the loop starting in line 23). We add b to the set S of solved vertices in line 24. Next, we identify the set R of rows in the updated matrix that depend on the original row b of (4) by looking for non-zero values in column b' of (5). This yields $R = \{1, 4\}$ and $r_{last} = 4$.

We enter the next for loop in line 27: here, this loop runs only once, for $r = 1$, i.e. the first row. In line 29, we add row r_{last} , i.e. the fourth row, to the first row to cancel dependency on original b row from the latter. From column b' , the first row depends on 1 times the original row b while the fourth row depends on 2 times the original row b . Thus, we must add 2 times the fourth row to the first row to cancel the dependency, obtaining:

$$K_{LS} = \begin{array}{c} F_1 \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \left(\begin{array}{c|cccc|cccc} i & a & b & d & i' & a' & b' & d' \\ \hline 1 & 0 & 4 & 1 & 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 0 & 0 & 4 & 0 & 3 & 0 & 2 & 1 \end{array} \right) \quad (6)$$

We exit the loop that started on line 27. At this point, only the fourth row has any dependency on row b from initial system K_{ILS} . More specifically, the dependency is on 2 times row b . We remove this dependency in line 29 by subtracting 2 times the b row of K_{ILS} from the fourth row of the current K_{LS} given in (6). This is equivalent to adding 3 times row b from K_{ILS} to row 4 of K_{LS} , since $-2 = 3$ in $\mathbb{F}_5$; it yields:

$$K_{LS} = \begin{array}{c} F_1 \\ \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \end{array} \end{array} \left(\begin{array}{c|cccc|cccc} i & a & b & d & i' & a' & b' & d' \\ \hline 1 & 0 & 4 & 1 & 0 & 0 & 0 & 0 & 2 \\ 2 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 3 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 4 & 3 & 3 & 0 & 2 & 3 & 0 & 0 & 1 \end{array} \right)$$

In the subsequent for loop from line 30, we iterate over those rows of K_{LS} containing pivots, to cancel as much of fourth row as possible. Here, this leads to the following system, obtained by adding 2 times the first row to the fourth row:

$$K_{LS} = \begin{array}{c} \begin{array}{c} F_1 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} i' \\ a' \\ b' \\ d' \end{array} \end{array} \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \end{array} \begin{array}{c} 4 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array} \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \end{array} \end{array} \quad (7)$$

In the second run of the loop from line 30, we immediately break because the second row of K_{LS} has no pivot in the first block. Finally, in lines 36 and 37 nothing happens, as the fourth row is already identically zero in the first block (or in fact in all blocks) and is correctly placed. This completes the first run of the main loop from line 16.

Second run of the main loop In the second run of the main loop, we observe that vertices a and d can be solved, but not the vertex i . This is because the columns a and d in the second block are identically zero from the second row onwards, but the same is not true for column i . Hence, $L = \{a, d\}$. We find and store the solutions for these vertices. The linear system for a is:

$$1f_1 + 0 = 0$$

hence $f_1 = 0$ is the only solution. The linear system for d is:

$$1f_1 + 1 = 0$$

hence $f_1 = -1 = 4$ is the only solution. Thus, at the start of line 23, we have:

$$\mathcal{P} = F_1(* \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} 0 \\ 1 \\ 4 \end{array})$$

Now, we have to remove the dependencies on the original rows a and d . We start with a : here the situation is simple, as only the second row of (7) depends on a (as indicated by column a' in the third block). Thus, after the entire run of the loop starting on line 23, with $v = a$, we obtain the updated system:

$$K_{LS} = \begin{array}{c} \begin{array}{c} F_1 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} i' \\ a' \\ b' \\ d' \end{array} \end{array} \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 4 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 1 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 2 \\ 0 \\ 0 \\ 0 \end{array} \end{array}$$

In the next run of the loop starting on line 23, we have $v = d$. Here, again only one row, the first row, depends on the original row d and the dependency has multiplicity 2. After this run of the loop, we get the system:

$$K_{LS} = \begin{array}{c} \begin{array}{c} F_1 \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} i \\ a \\ b \\ d \end{array} \begin{array}{c} i' \\ a' \\ b' \\ d' \end{array} \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 1 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \end{array}$$

In the process, S is updated to contain b , a , and d . This ends the second run of the main loop from line 16.

Third run of the main loop In the third run of the loop starting on line 16, i can now be solved. The corresponding linear system is:

$$0f_1 + 0 = 0$$

which is singular and consistent. It is satisfied by any value of f_1 . So, for example, we can choose the solution $f_1 = 0$, leading to:

$$\mathcal{P} = \begin{matrix} & i & a & b & d \\ F_1 & (0 & 0 & 1 & 4) \end{matrix}$$

Vertex i is added to S , i.e. at this point $S = \{b, a, d, i\} = \bar{O}$. Afterwards, the system is still updated to the form it would have had if i was never included, which here will result in a system that has 0s everywhere.

Final steps Since $S = \bar{O}$, no more runs of the loop from line 16 occur. Instead, on line 38, we construct $C^{\mathcal{B}}$:

$$C^{\mathcal{B}} = \begin{matrix} & i & a & b & d \\ i & (1 & 0 & 0 & 0) \\ a & (0 & 1 & 0 & 0) \\ b & (0 & 0 & 1 & 0) \\ d & (0 & 0 & 0 & 1) \\ F_1 & (0 & 0 & 1 & 4) \end{matrix}$$

and finally output $C'C^{\mathcal{B}}$, which is the reduced correction matrix, and $NC'C^{\mathcal{B}}$, which encodes the partial order:

$$C'C^{\mathcal{B}} = \begin{matrix} & i & a & b & d \\ a & (3 & 0 & 0 & 0) \\ b & (0 & 0 & 2 & 0) \\ d & (0 & 0 & 0 & 4) \\ o_1 & (2 & 2 & 0 & 0) \\ o_2 & (1 & 0 & 1 & 3) \end{matrix} \quad NC'C^{\mathcal{B}} = \begin{matrix} & i & a & b & d \\ i & (0 & 0 & 0 & 0) \\ a & (1 & 0 & 0 & 0) \\ b & (1 & 2 & 0 & 4) \\ d & (3 & 0 & 0 & 0) \end{matrix}$$

Note that the $\mathbb{F}_d$ -flow found here differs from that presented in Figures 1 and 2, which is fine as the example has more outputs than inputs and so it can have multiple focused $\mathbb{F}_d$ -flows. If, during the third run of the main loop, we had chosen the solution for i to be 3 instead of 0, then we would instead recover the solution in Figure 2.

Lastly, we note that the solutions found each run of the main loop from line 16 form the ‘layers’ of the partial order. For the worked example, these layers are thus $[b]$, $[a, d]$, and $[i]$. The algorithm for $\mathbb{F}_d$ -flow-finding from [8] would produce the same layers. However, our method also returns the induced partial order $\prec_C$ corresponding to the correction matrix C , as encoded in the product $NC = NC'C^{\mathcal{B}}$. This order is the minimal one (cf. Lemma C.6). In general, the partial order returned by Algorithm 2 could be less restrictive than an order found via layers⁵.

G Lower bound for the complexity of $\mathbb{F}_d$ -flow-finding

The complexity bound of our flow-finding algorithm is $O(n^3)$ only when $|I| \neq |O|$. For $|I| = |O|$, we invoke subroutines for the problem of matrix inversion and multiplication, which, at least in principle, can be solved with sub-cubic complexity. Here, we show that, in fact, reducing the other way is also possible,

⁵ Nevertheless, for this particular example, the minimal order is precisely the same as the order implied by ordering via found layers, i.e. $i \prec a \prec b$ and $i \prec d \prec b$.

thus tightly linking the complexity of flow-finding with the aforementioned fundamental problems of linear algebra.

Firstly, the Theorem 4.5 implies the following corollary, which is more suitable for the discussion of lower bounds:

Corollary G.1. Let $\Gamma = (G, I, O, \lambda)$ be a labelled open $\mathbb{F}_d$ -graph, let M be its flow-demand matrix, and let N be its order-demand matrix. Then Γ has $\mathbb{F}_d$ -flow if and only if there exists a reduced correction matrix C over $\mathbb{F}_d$ such that:

- C has shape $\bar{I} \times \bar{O}$,
- $MC = \text{Id}_{\bar{O}}$, and
- NC is the adjacency matrix of a directed acyclic $\mathbb{F}_d$ -weighted graph.

The $\mathbb{F}_d$ -flow on Γ implied by the existence of this matrix C is $(C, \prec_c)$, where furthermore the following holds for any $v, w \in \bar{O}$:

- $v \prec_c w \Leftrightarrow (NC)_{w,v} \neq 0$ and
- $\prec_c$ is the transitive closure of $\prec_c$.

Proof. By inspection, this follows from Theorem 4.5 and its supporting Lemmata C.14 and C.13. $\square$

We adapt the lower bound argument for Pauli flow on qubits [32, Theorem 4.3] to $\mathbb{F}_d$ -flow:

Theorem G.2 ($\mathbb{F}_d$ -flow-finding lower bound). The problem of finding $\mathbb{F}_d$ -flow is at least as computationally expensive as the problem of finding the inverse of a matrix over $\mathbb{F}_d$.

Proof. Let M be an arbitrary $n \times n$ matrix over $\mathbb{F}_d$. Define $\Gamma := (G, I, O, \lambda)$ as follows:

$$\begin{aligned} I &= (i_1, \dots, i_n) \\ O &= (o_1, \dots, o_n) \\ V &= (I \cup O) \\ G_{i_k, o_\ell} &= M_{k, \ell} \quad (\forall k, \ell \in \{1, \dots, n\}) \\ \lambda(i_k) &= (0, 1) \quad (\forall k \in \{1, \dots, n\}) \end{aligned}$$

Then the flow-demand matrix M_Γ of (G, I, O, λ) is precisely equal to M and the order-demand matrix N_Γ is identically zero. Thus, by Corollary G.1, Γ has $\mathbb{F}_d$ -flow if and only if M is invertible. If $\mathbb{F}_d$ -flow exists, the correction matrix is given by the inverse of M . $\square$

The above theorem leads to an $\Omega(n^2)$ lower bound for $\mathbb{F}_d$ -flow-finding. While this bound is already implied by the size of the input, it is generally expected that matrix inversion and multiplication require $\omega(n^2)$ field operations; currently a formal proof of this is known only for matrices over $\mathbb{R}$ or $\mathbb{C}$ [13, Section 28.2].

H Details about $\mathbb{F}_d$ -flow-preserving rewriting

This section goes into more detail about the $\mathbb{F}_d$ -flow-preserving rewrite rules and related procedures summarised in Section 6. In Subsection H.1, we first explain known results about which graph operations can be implemented using local Clifford unitaries. We use these in Subsection H.2 to formally define pivoting and show that it is $\mathbb{F}_d$ -flow-preserving. In Subsection H.3, we prove that $\mathbb{F}_d$ -flow reversal is

also $\mathbb{F}_d$ -flow-preserving. In Subsection H.4, we show how to remove and insert Z-like vertices in an $\mathbb{F}_d$ -flow-preserving way. Finally, in Subsection H.5, we combine Z-like insertion with other Clifford-based transformations to sketch an algorithm that generates arbitrary instances of labelled open $\mathbb{F}_d$ -graphs with $\mathbb{F}_d$ -flow.

H.1 Graph operations that can be implemented via local Clifford unitaries

Several relevant rewrite rules were already defined and proved flow-preserving by Booth [10, Section 5.1]. These rules are based on the two different types of operations on an $\mathbb{F}_d$ -weighted graph which translate to local Clifford equivalences of the corresponding qudit graph states [6, Section D]. The first of these operations is *local scaling*, which multiplies the weights of all edges incident on a chosen vertex w by some non-zero constant $\gamma \in \mathbb{F}_d^*$. Denoting by $_{\circ}^{\gamma} w$ the local scaling operation just described, we have:

$$(G_{\circ}^{\gamma} w)_{u,v} := \begin{cases} \gamma G_{u,v} & \text{if } u = w \text{ or } v = w \\ G_{u,v} & \text{otherwise} \end{cases} \quad \text{and} \quad \left| G_{\circ}^{\gamma} w \right\rangle = \mathcal{R}_w^{(\gamma^{-1})} |G\rangle, \quad (8)$$

where $\mathcal{R}^{(\alpha)} |k\rangle = |\alpha k\rangle$ for any $\alpha, k \in \mathbb{F}_d$ and the subscript w indicates that this single-qudit Clifford unitary is applied to qudit w [10, Section 5.1.1].

The second operation is *local complementation*, which is also parametrised by a constant⁶ $\gamma \in \mathbb{F}_d$ and changes the weights of all edges among the neighbours of some chosen vertex w in a more complicated way. Denoting by $_{\star}^{\gamma} w$ the local complementation operation of weight γ , we have:

$$(G_{\star}^{\gamma} w)_{u,v} = \begin{cases} G_{u,v} + \gamma G_{u,w} G_{w,v} & \text{if } u \neq v \\ G_{u,v} & \text{otherwise.} \end{cases} \quad (9)$$

Note that $G_{u,w} G_{w,v} = 0$ unless both u and v are neighbours of w , so it is only edges between two neighbours of w that change.

Let τ be the d -th root of unity that satisfies $\tau^2 = \omega$ and $\tau^d = 1$. Denote by R a qudit phase gate⁷ satisfying $R|k\rangle = \tau^{k^2}|k\rangle$ and by $\tilde{R} := H R H^{-1}$ an equivalent gate in the X -eigenbasis. Then the corresponding relationship between graph states is⁸

$$\left| G_{\star}^{\gamma} w \right\rangle = \tilde{R}_w^{\gamma} \prod_{v \in V} R_v^{-\gamma G_{w,v}^2} |G\rangle \quad (10)$$

cf. [7, Proposition 12 of arXiv version], see Observation H.8 in Appendix H.2 for the translation from the ZX-calculus. Both of the above operations have been shown to preserve $\mathbb{F}_d$ -flow according to Definition 2.7, i.e. working with full (not reduced) correction matrices. By Theorem 3.2, this implies that focused $\mathbb{F}_d$ -flow is also preserved. Yet the changes necessary to preserve focusing might be more complicated⁹ and will not be worked out here.

⁶ Unlike for local scaling, we may allow $\gamma = 0$ in this case. This yields a trivial transformation that leaves the graph invariant.

⁷ Note that this is not the usual qudit phase gate, which is generally defined as $S|k\rangle = \tau^{k(k-1)}|k\rangle = R Z^{-2^{-1}}$, where 2^{-1} is the multiplicative inverse of 2 in $\mathbb{F}_d$.

⁸ This relationship is not entirely consistent with that given in [10, Section 5.1.2], yet the differences consist only of Pauli operations which can be pushed past the Paulis used to define measurement spaces and corrections up to some scalar factor. Hence we can nevertheless use results from the latter source about the changes of measurement planes and the preservation of $\mathbb{F}_d$ -flow during local complementations.

⁹ C.f. [39, Lemma D.15] for the explicit expression of the focused flow after a local complementation in the qubit case.

Proposition H.1 ([10, Proposition 5.5]). Suppose the labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) has an $\mathbb{F}_d$ -flow given by $(C, \prec)$. Then, for any $w \in V$ and any non-zero $\gamma \in \mathbb{Z}_d^*$, the labelled open $\mathbb{F}_d$ -graph $(G \overset{\gamma}{\circ} w, I, O, \lambda')$ with new measurement labels

$$\lambda'(u) := \begin{cases} (\gamma^{-1}C_{w,w}, \gamma(GC)_{w,w}) & \text{if } u = w \\ (C_{u,u}, (GC)_{u,u}) & \text{otherwise} \end{cases}$$

has an $\mathbb{F}_d$ -flow $(C', \prec)$ defined as

$$C'_{u,v} := \begin{cases} \gamma^{-1}C_{w,v} & \text{if } u = w \\ C_{u,v} & \text{otherwise.} \end{cases}$$

Proposition H.2 ([10, Proposition 5.8]). Suppose the labelled open $\mathbb{F}_d$ -graph (G, I, O, λ) has an $\mathbb{F}_d$ -flow given by $(C, \prec)$. Then, for any $w \in \bar{I}$ and any $\gamma \in \mathbb{Z}_d$, the labelled open $\mathbb{F}_d$ -graph $(G \star w, I, O, \lambda')$ with new measurement labels¹⁰

$$\lambda'(u) := \begin{cases} (C_{w,w} - \gamma(GC)_{w,w}, (GC)_{w,w}) & \text{if } u = w \\ (C_{u,u}, (GC)_{u,u} - \gamma G_{u,w}^2 C_{u,u}) & \text{otherwise} \end{cases}$$

has an $\mathbb{F}_d$ -flow $(C', \prec)$ defined as

$$C'_{u,v} := \begin{cases} C_{w,v} - \gamma(GC)_{w,v} & \text{if } u = w \\ C_{u,v} & \text{otherwise.} \end{cases}$$

Remark H.3. The above transformation of the flow is much simpler than that presented in the qubit local complementation proof for (extended) gflow [3, Lemma 3.1]. It can in fact be adapted to give a simpler transformation for gflow, though again this transformation does not preserve focusing.

In the binary case, given an edge ww' in a graph, the sequence of three local complementations alternating about the two endpoints of the edge (in either order), $G \wedge ww' = ((G \star w) \star w') \star w = ((G \star w') \star w) \star w'$, is known as a *pivot* or an *edge-local complementation*. Pivots also give rise to flow-preserving transformations and they interact better with focused gflow and Pauli flow than individual local complementations [5, Lemma 5.3]. Again in the qubit case, the associated graph states satisfy $|(G \star w) \star w' \star w\rangle = H_w H_{w'} \prod_{v \in N_G(w) \cap N_G(w')} Z_v |G\rangle$, i.e. a pivot corresponds to applying Hadamard gates to the two endpoints of the edge and Z-gates to all joint neighbours of both endpoints.

Generalising the qubit case, a family of pivot operations on qutrits was defined in [42, Theorem 3.5]. Another work presents a qudit generalisation of the pivot-and-delete operation, where the vertices w and w' are deleted from the graph after the pivot [34, Lemma 10].

Motivated by the elegance of the qubit pivot operation, and the usefulness of having a graph operation that corresponds to Hadamard gates on certain qudits, we generalise the pivot operation to qudits.

Proposition H.4 (Formal version of Proposition 6.1). Let G be an $\mathbb{F}_d$ -weighted graph with vertices V , suppose $w, w' \in V$ satisfy $\epsilon := G_{w,w'} \neq 0$, and define $G \wedge ww'$ to be the graph on V with the following

¹⁰ Note there is a typo in the definition of the updated labels in [10, Eq. (5.61)], this has been corrected here.

adjacency matrix:

$$(G \wedge ww')_{u,v} := \begin{cases} 0 & \text{if } u = v \\ -\epsilon & \text{if } \{u, v\} = \{w, w'\} \\ G_{w',v} & \text{if } u = w \text{ and } v \notin \{w, w'\} \\ G_{u,w'} & \text{if } v = w \text{ and } u \notin \{w, w'\} \\ G_{w,v} & \text{if } u = w' \text{ and } v \notin \{w, w'\} \\ G_{u,w} & \text{if } v = w' \text{ and } u \notin \{w, w'\} \\ G_{u,v} - \epsilon^{-1}(G_{u,w'}G_{w,v} + G_{u,w}G_{w',v}) & \text{otherwise,} \end{cases}$$

Then the associated qudit graph state satisfies:

$$|G \wedge ww'\rangle = F_w^{(-\epsilon)} F_{w'}^{(-\epsilon)} \prod_{v \in V} R_v^{2\epsilon^{-1}G_{v,w}G_{v,w'}} |G\rangle$$

where $F^{(\gamma)} := \sum_{j,k \in \mathbb{F}_d} \omega^{\gamma jk} |j\rangle\langle k|$ for any $\gamma \in \mathbb{F}_d^*$, and $R = \sum_{k \in \mathbb{F}_d} \tau^{k^2} |k\rangle\langle k|$.

As proving this result using standard matrix notation showed itself to be onerous and error-prone, we prove the above proposition in Appendix H.2 using the qudit stabiliser ZX-calculus of [34], which is also introduced in that appendix.

Analogous to the qubit case, a pivot operation can be built up from three successive local complementations. Yet for qudits, at least one local scaling operation is also needed to get a pivot operation that is symmetric under interchange of w and w' .

Observation H.5. Let G be an $\mathbb{F}_d$ -weighted graph with vertices V , suppose $w, w' \in V$ satisfy $G_{w,w'} \neq 0$, and define $\gamma := G_{w,w'}^{-1}$. Then:

$$G \wedge ww' = (((G \overset{\gamma}{\star} w) \overset{-\gamma}{\star} w') \overset{\gamma}{\star} w) \overset{-1}{\circ} w = (((G \overset{\gamma}{\circ} w') \overset{1}{\star} w) \overset{-1}{\star} w') \overset{1}{\star} w) \overset{-1/\gamma}{\circ} w$$

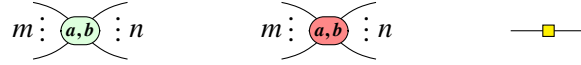
and similarly with the roles of w, w' swapped.

The proof is straightforward but sufficiently onerous that we leave it out. Nevertheless, as a pivot corresponds to a sequence of local complementations and local scalings, it is immediate that it preserves the existence of $\mathbb{F}_d$ -flow.

H.2 Proving the pivoting property using the qudit stabiliser ZX-calculus

The ZX-calculus is a graphical language for reasoning about quantum computations. While the qubit version is the most developed, there is also a complete ZX-calculus for the stabiliser fragment of d -dimensional qudits for every prime d . The case $d = 2$ was shown in [1] and odd primes were treated in [7, 34]. Since generalising the pivot rule from the qubit case to qudits using algebraic techniques requires long, detailed, but not very insightful manipulations, and there is already a ZX-calculus proof of the related pivot-and-delete rule (also called “pivot simplification”) [34, Lemma 10], we prove a (non-deletion version) of the pivot rule using the qudit ZX-calculus. We will introduce here only those aspects of the rules and interpretation that are immediately necessary for the proof. A reader interested in more detail may find a full introduction in [34].

The qudit stabiliser ZX-calculus is generated by three kinds of components: green Z-spiders and red X-spiders, which have a phase label in $\mathbb{Z}_d^2$ and may have arbitrary number of inputs and output wires, as well as Hadamard gates, which always have one input and one output:



A phase 0,0 on a spider is usually left out. ZX-diagrams may be composed sequentially by connecting the input wires of one diagram with the output wires of another, and they may be composed in parallel via vertical juxtaposition. Recall that τ denotes the d -th root of unity satisfying $\tau^2 = \omega$, and that $|k : U\rangle$ for some $k \in \mathbb{F}_d$ denotes the eigenstate of the unitary operator U with eigenvalue ω^k . Then the components are interpreted as follows [34]:

$$\begin{aligned} \left[\begin{array}{c} m \vdots \text{green spider}(a,b) \vdots n \end{array} \right] &= d^{\frac{n+m-2}{4}} \sum_{k \in \mathbb{Z}_d} \tau^{ak+bk^2} |k : Z\rangle^{\otimes n} \langle k : Z|^{\otimes m} \\ \left[\begin{array}{c} m \vdots \text{red spider}(a,b) \vdots n \end{array} \right] &= d^{\frac{n+m-2}{4}} \sum_{k \in \mathbb{Z}_d} \tau^{ak+bk^2} |-k : X\rangle^{\otimes n} \langle k : X|^{\otimes m} \\ \left[\text{Hadamard gate} \right] &= \sum_{k \in \mathbb{Z}_d} |k : Z\rangle \langle k : X| \\ \left[\text{identity wire} \right] &= \sum_{k \in \mathbb{Z}_d} |k : Z\rangle \langle k : Z| \\ \left[\text{crossing} \right] &= \sum_{k, \ell \in \mathbb{Z}_d} |k, \ell : Z\rangle \langle \ell, k : Z| \end{aligned}$$

Every ZX-calculus diagram can be treated as a labelled graph in the sense that, as long as the input and output wires of the entire diagram remain in the same order, only the connectivity of the components matters; this is made rigorous via category theory.

The following two syntactic sugars will be useful. Firstly, denote by $\text{H}^\square$ the inverse of the Hadamard operation so that

$$\text{H}^\square \text{H}^\square = \text{identity} = \text{H}^\square \text{H}^\square$$

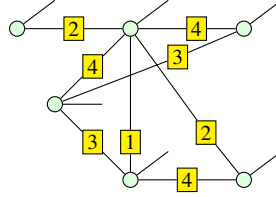
Secondly, introduce the *H-box*: a Hadamard with a label $a \in \mathbb{F}_d$ called its *weight*. This represents a parallel Hadamard edges between a pair of Z-spiders:

$$\text{H}^\square(a) := \text{green spider} \text{---} \text{H}^\square(a) \text{---} \text{green spider}$$

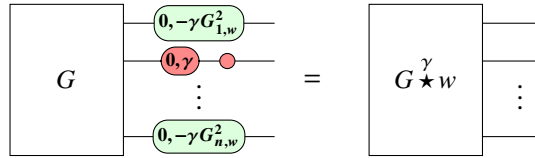
Observation H.6. An $\mathbb{F}_d$ -weighted graph state $|G\rangle$ (cf. Definition 2.4) is represented in the ZX-calculus as follows:

- For each graph vertex, there is a green spider of phase (0,0) which is connected to one output wire.
- For each edge of weight $k \in \mathbb{Z}_d$, there is an H-box of weight k connecting the spiders corresponding to the two endpoints of the edge.

For example, the $\mathbb{F}_d$ -weighted graph state underlying the running example of Figure 1a corresponds to the following ZX-diagram:



Proposition H.7 ([7, Proposition 12]). For any $\mathbb{F}_d$ -weighted graph G identified with its adjacency matrix $G \in \mathbb{F}_d^{V \times V}$, for any $\gamma \in \mathbb{F}_d$, and for any $w \in V = \{1, \dots, n\}$:



where the labelled white boxes denote the graph state diagrams described in Observation H.6 and, on the left-hand side, the X -spiders are connected to vertex w .

Observation H.8. The ZX-diagram equality of Proposition H.7 implies:

$$|G \star^\gamma w\rangle = \tilde{R}_w^\gamma \prod_{v \in V} R_v^{-\gamma G_{w,v}^2} |G\rangle$$

since the matrix G is symmetric and the two-legged spiders satisfy:

$$\begin{aligned} \llbracket \text{red circle } 0, \gamma \rrbracket &= \left(\sum_{k \in \mathbb{F}_d} \tau^0 |-k : X\rangle \langle k : X| \right) \left(\sum_{k \in \mathbb{F}_d} \tau^{\gamma k^2} |-k : X\rangle \langle k : X| \right) = \sum_{k \in \mathbb{F}_d} \tau^{\gamma k^2} |k : X\rangle \langle k : X| = \tilde{R}^\gamma \\ \llbracket \text{green oval } 0, -\gamma G_{v,w}^2 \rrbracket &= \sum_{k \in \mathbb{F}_d} \tau^{-\gamma G_{v,w}^2 k^2} |k\rangle \langle k| = R^{-\gamma G_{v,w}^2} \end{aligned}$$

To prove an analogous result for pivots, we will need the following rewrite rules of the qudit stabiliser ZX-calculus.

Proposition H.9 ([34, Proposition 6]). H-boxes of weight 0 and 1 satisfy $\text{H-box } 0 = \text{green circle}$ and $\text{H-box } 1 = \text{yellow square}$. Additionally, H-boxes compose as follows:

$$\begin{aligned} \text{H-box } x \text{ followed by H-box } y &= \text{H-box } xy^{-1} \text{ followed by H-box } y \\ \text{H-box } x \text{ followed by H-box } -x &= \text{H-box } xy \\ \text{green circle } x \text{ and green circle } y &= \text{H-box } x+y \end{aligned}$$

where in the first equality y has to be non-zero, but otherwise $x, y \in \mathbb{Z}_d$ are arbitrary.

Proposition H.10 ([7, Proposition 8 of arXiv version]). The inverse Hadamard is equal to the H-box of weight -1 , for any $x \in \mathbb{Z}_d^*$ the H-boxes of weight x and weight $-x$ are inverse to each other, and repeating a non-zero-weighted H-box three times yields its inverse:¹¹

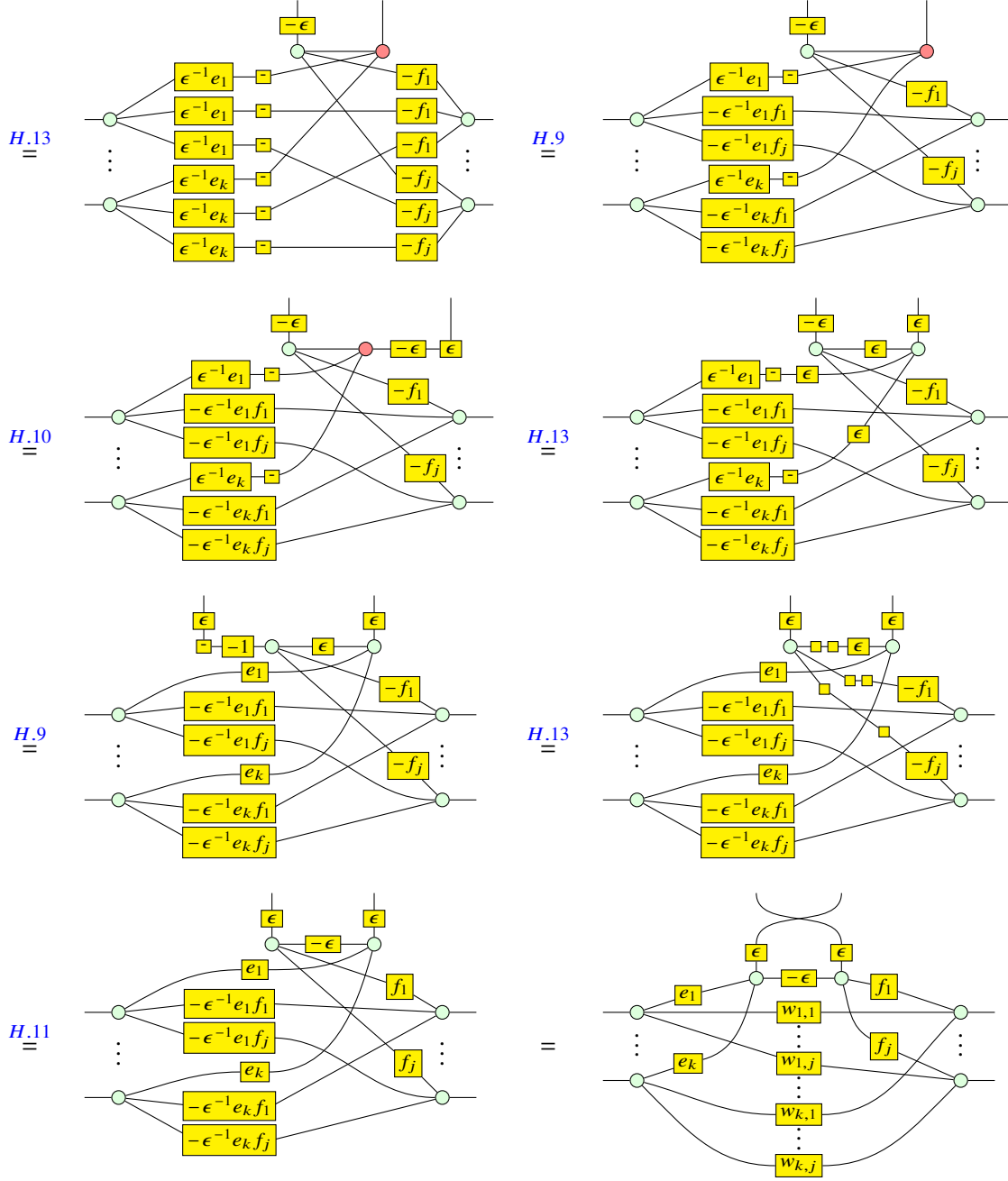
$$\begin{aligned} \text{H-box } x &= \text{H-box } -1 \\ \text{H-box } x \text{ followed by H-box } -x &= \text{line} \\ \text{H-box } x \text{ followed by H-box } x \text{ followed by H-box } x &= \text{H-box } -x \end{aligned}$$

Corollary H.11. Following from the two propositions above, for any $x \in \mathbb{Z}_d$, we have:

$$\text{H-box } x \text{ followed by H-box } x \text{ followed by H-box } x = \text{H-box } -1 \text{ followed by H-box } x = \text{H-box } -x$$

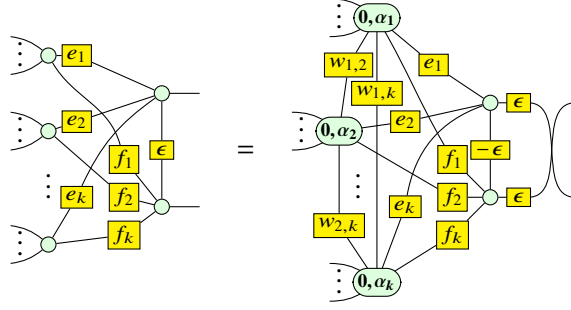
Lemma H.12 ([34, Lemma 31]). The bialgebra law holds for any $m, n \in \mathbb{N}$ such that $m, n \geq 2$:

¹¹ The definition of the ZX-calculus in [7] differs slightly from the ‘well-tempered’ normalisation used in [34] and given above, but those differences are not relevant to the results used here.



□

Lemma H.15. The following qudit pivot rule holds:



where $\alpha_m = -2\epsilon^{-1}e_m f_m$ and $w_{m,n} = -\epsilon^{-1}(e_m f_n + e_n f_m)$ for all $m, n \in [k]$.

Proof. This is analogous to the proof of [34, Lemma 10], using our Lemma H.14 instead of their Lemma 9. $\square$

We can now show which local unitary connects two graph states related by a pivot operation.

Proof of Proposition H.4. Express the graph states in the ZX-calculus as in Observation H.6 and note that we may treat non-existent edges as H-boxes of weight 0 (cf. Proposition H.9). Apply Lemma H.15 (bending the inputs into outputs) and use the standard interpretation to find

$$|G\rangle = \left(\llbracket -\epsilon \rrbracket_w \otimes \llbracket -\epsilon \rrbracket_{w'} \otimes \bigotimes_{v \in V \setminus \{w, w'\}} \llbracket -0, \alpha_v \rrbracket_v \right) |G \wedge ww'\rangle$$

where $\alpha_v = -2\epsilon^{-1}G_{v,w}G_{v,w'}$. The interpretation of the graph underlying the right-hand side of Lemma H.15 as $|G \wedge ww'\rangle$ is correct by inspection.

Now, it is straightforward to verify that

$$\begin{aligned} \llbracket -\epsilon \rrbracket &= \sum_{j, k \in \mathbb{F}_d} \omega^{\epsilon jk} |j\rangle\langle k| = F^{(\epsilon)} \\ \llbracket -0, \alpha_v \rrbracket &= \sum_{k \in \mathbb{F}_d} \tau^{\alpha_v k^2} |k\rangle\langle k| = R^{\alpha_v} \end{aligned}$$

Note that $(F^{(\gamma)})^{-1} = F^{(-\gamma)}$ for any $\gamma \in \mathbb{F}_d^*$. Hence the desired result follows by moving the single-qudit operators to the other side of the equation, noting that we may allow R operators on w and w' as $\alpha_w = \alpha_{w'} = 0$, and that the single-qudit operators all commute, meaning $\bigotimes$ can be changed to $\prod$. $\square$

H.3 Proving flow reversibility

Before proving Theorem 6.3, we provide a build-up with various supporting observations, like we have done in our proof for the qubit case [32, Subsection 3.3].

We start by naming relevant types of measurements:

Definition H.16. We distinguish two types of measurements denoted ‘X-like measurements’ $\mathcal{X}$ and ‘Z-like measurements’ $\mathcal{Z}$ as follows (as a reminder, B is the set of vertices internal vertices, cf. Definition 2.2):

$$\begin{aligned} \mathcal{X} &:= \{v \in B \mid \lambda_X(v) = 0\} \\ \mathcal{Z} &:= \{v \in B \mid \lambda_X(v) \neq 0\} \end{aligned}$$

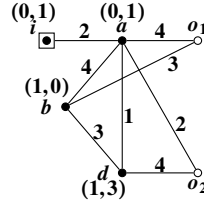


Figure 3: The labelled open $\mathbb{F}_d$ -graph from Figure 1 after normalisation of measurement labels.

We exclude each vertex that is an input and an output simultaneously:

Observation H.17. Vertices that are both inputs and outputs do not impact the existence of $\mathbb{F}_d$ -flow at all, neither in Γ nor in its reverse Γ' . Thus, from now on, we assume $I \cap O = \emptyset$ and $V = I \cup X \cup Z \cup O$.

Given a labelled open $\mathbb{F}_d$ -graph and some vertex $v \in \bar{O}$, $\mathbb{F}_d$ -flow is preserved if $\lambda(v)$ is replaced by its scalar multiple for any non-zero scalar in $\mathbb{F}_d^*$ [8, Section 3]. For example, in the labelled open $\mathbb{F}_d$ -graph from Figure 1, vertex d could instead be given the measurement label $3 \cdot (4, 2) = (2, 1)$ and this would preserve the existence of $\mathbb{F}_d$ -flow.

Now, for any measurement label (a, b) , there exists $c \in \mathbb{F}_d$ such that (ac, bc) is of the form $(0, 1)$ or $(1, k)$ for some $k \in \mathbb{F}_d$: if $a = 0$, then necessarily $b \neq 0$, so taking $c = b^{-1}$ works; and if $a \neq 0$, taking $c = a^{-1}$ works. Therefore, without loss of generality, one may work with normalised labelled open $\mathbb{F}_d$ -graphs in the following sense:

Definition H.18. A labelled open $\mathbb{F}_d$ -graph is called *normalised* if all measurements labels are of the form $(0, 1)$ or $(1, k)$ for some $k \in \mathbb{F}_d$.

For example, normalisation of the labelled open $\mathbb{F}_d$ -graph from Figure 1 results in the labelled open $\mathbb{F}_d$ -graph in Figure 3.

After the normalisation of measurement labels, we will now introduce some notation related to the graph part of a labelled open $\mathbb{F}_d$ -graph. Since we identify the graph with its adjacency matrix, let us name relevant submatrices of G as follows, where $*$ stands for an unnamed submatrix:

$$G =: \begin{pmatrix} * & G_{00} & G_{01} \\ * & * & * \\ * & G_{10} & G_{11} \end{pmatrix} \begin{matrix} I \cup X \\ O \\ Z \end{matrix}$$

Now, we can rewrite the flow-demand matrix in block form:

Observation H.19. Let $\Gamma = (G, I, O, \lambda)$ be a normalised labelled open $\mathbb{F}_d$ -graph which satisfies $|I| = |O|$ and $I \cap O = \emptyset$; and let Γ' be its reverse (cf. Definition 6.2). Then the flow-demand matrix M of Γ and the flow-demand matrix M' of Γ' satisfy:

$$M = \begin{pmatrix} G_{00} & G_{01} \\ 0 & \text{Id} \end{pmatrix} \begin{matrix} X \cup O \\ Z \end{matrix} \quad \text{and} \quad M' = \begin{pmatrix} (G_{00})^T & (G_{01})^T \\ 0 & \text{Id} \end{pmatrix} \begin{matrix} I \cup X \\ Z \end{matrix} = \begin{pmatrix} G_{00} & 0 \\ G_{10} & \text{Id} \end{pmatrix}^T$$

Similarly, we can present the order-demand matrix in block form:

Observation H.20. Let $\Gamma = (G, I, O, \lambda)$ be a normalised labelled open $\mathbb{F}_d$ -graph which satisfies $|I| = |O|$ and $I \cap O = \emptyset$; and let Γ' be its reverse. In the order-demand matrix N of Γ , rows corresponding to vertices in I and in X are defined analogously; yet this results in the rows corresponding to vertices in I

being identically zero. We can write the order-demand matrix as a product $N = PS$, where P is a diagonal $\bar{O} \times \bar{O}$ matrix identifying the internal vertices, i.e.:

$$P := \text{Diag}(1 \text{ for } v \in B \text{ and } 0 \text{ for } v \in I) \quad \text{and} \quad S := \begin{pmatrix} \overset{X \cup O}{J} & \overset{Z}{0} \\ G_{10} & G_{11} - D \end{pmatrix} \begin{matrix} I \cup X \\ Z \end{matrix}$$

with $J_{vw} = \delta_{vw}$ and D being a diagonal matrix expressing $\lambda_Z(v)$, i.e. $D = \text{Diag}(\lambda_Z(v) \text{ for } v \in Z)$. From now on, we write $K_{11} := G_{11} - D$. Note that K_{11} is symmetric because G_{11} is a diagonal block of the (symmetric) adjacency matrix and D is diagonal. Moreover, while J is square, it is not generally the identity matrix because the sets of row and column labels need not match.

Similarly to N , the order-demand matrix N' of Γ' can be written as a product $N' = P'S'$ where P' is a diagonal $\bar{I} \times \bar{I}$ matrix:

$$P' := \text{Diag}(1 \text{ for } v \in B \text{ and } 0 \text{ for } v \in O) \quad \text{and} \quad S' := \begin{pmatrix} \overset{I \cup X}{J^T} & \overset{Z}{0} \\ (G_{10})^T & K_{11} \end{pmatrix} \begin{matrix} X \cup O \\ Z \end{matrix} = \begin{pmatrix} J & G_{01} \\ 0 & K_{11} \end{pmatrix}^T$$

where the last equality follows from the symmetry of K_{11} .

We are now ready to prove Theorem 6.3:

Proof of Theorem 6.3. Without loss of generality, we assume that Γ is normalised and that $I \cap O = \emptyset$.

($\Rightarrow$): assume Γ has $\mathbb{F}_d$ -flow. By Theorem 4.5, there exists C such that $MC = \text{Id}$ and NC forms a DAG, where M and N are the flow-demand and order-demand matrices of Γ , respectively. By Observation H.19:

$$M = \begin{pmatrix} \overset{X \cup O}{G_{00}} & \overset{Z}{G_{01}} \\ 0 & \text{Id} \end{pmatrix} \begin{matrix} I \cup X \\ Z \end{matrix}$$

Since $|I| = |O|$, M is square and thus, by Corollary 4.6, C is unique and $C = M^{-1}$. Break C into blocks corresponding to the block structure of M :

$$C = \begin{pmatrix} \overset{I \cup X}{C_{00}} & \overset{Z}{C_{01}} \\ C_{10} & C_{11} \end{pmatrix} \begin{matrix} X \cup O \\ Z \end{matrix}$$

Then:

$$MC = \begin{pmatrix} \overset{I \cup X}{G_{00}C_{00} + G_{01}C_{10}} & \overset{Z}{G_{00}C_{01} + G_{01}C_{11}} \\ C_{10} & C_{11} \end{pmatrix} \begin{matrix} I \cup X \\ Z \end{matrix}$$

Therefore $C_{10} = 0$ and $C_{11} = \text{Id}$ by inspection. Using these, the top left block implies $\text{Id} = G_{00}C_{00}$, i.e. $C_{00} = (G_{00})^{-1}$ (note that G_{00} is square). Similarly, the top right block becomes $0 = G_{00}C_{01} + G_{01}$. Hence $G_{00}C_{01} = -G_{01}$ and finally $C_{01} = -(G_{00})^{-1}G_{01} = -C_{00}G_{01}$. To summarize:

$$C = \begin{pmatrix} \overset{I \cup X}{C_{00}} & \overset{Z}{-C_{00}G_{01}} \\ 0 & \text{Id} \end{pmatrix} \begin{matrix} X \cup O \\ Z \end{matrix}$$

Again by Observation H.19, the flow-demand matrix of Γ' satisfies:

$$M' = \begin{pmatrix} G_{00} & 0 \\ G_{10} & \text{Id} \end{pmatrix}^T$$

As G_{00} is invertible (with inverse C_{00}) and because of its block structure, M' is also invertible. By an argument analogous to that for M , we find that:

$$C' := (M')^{-1} = \begin{pmatrix} C_{00} & 0 \\ -G_{10}C_{00} & \text{Id} \end{pmatrix}^T$$

To show that Γ' has $\mathbb{F}_d$ -flow, by Theorem 4.5, it remains to show that $N'C'$ forms a DAG, where N' is the order-demand matrix of Γ' . First, by Observation H.20:

$$NC = PSC = P \begin{pmatrix} J & 0 \\ G_{10} & K_{11} \end{pmatrix} \begin{pmatrix} C_{00} & -C_{00}G_{01} \\ 0 & \text{Id} \end{pmatrix} = P \begin{pmatrix} JC_{00} & -JC_{00}G_{01} \\ G_{10}C_{00} & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}$$

and similarly:

$$\begin{aligned} N'C' &= P'S'C' = P' \begin{pmatrix} J & G_{01} \\ 0 & K_{11} \end{pmatrix}^T \begin{pmatrix} C_{00} & 0 \\ -G_{10}C_{00} & \text{Id} \end{pmatrix}^T = P' \left(\begin{pmatrix} C_{00} & 0 \\ -G_{10}C_{00} & \text{Id} \end{pmatrix} \begin{pmatrix} J & G_{01} \\ 0 & K_{11} \end{pmatrix} \right)^T \\ &= P' \begin{pmatrix} C_{00}J & C_{00}G_{01} \\ -G_{10}C_{00}J & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}^T \end{aligned}$$

Note that NC is a $\bar{O} \times \bar{O}$ matrix satisfying $(NC)_{vw} = 0$ whenever $v \in I$ and thus rows labelled by I are irrelevant to determining whether NC is a DAG. This means we can also exclude columns labelled by I , as all inputs will be initial in the partial order and thus may have arbitrary successors among the non-inputs. Hence it suffices to consider only the submatrix of NC whose rows and columns are both labelled by B . An analogous argument holds for $N'C'$, where it is rows and columns labelled by O that are excluded to leave only the submatrix whose rows and columns are both labelled by B .

Now, the three matrices P , P' , and J each reduce to an identity matrix when restricted to rows and columns labelled by B . In the following, we denote a submatrix obtained by restricting rows and columns to vertices in B by adding a subscript and superscript next to the matrix. Then the $B \times B$ submatrices satisfy:

$$(N'C')_B^B = \left(\begin{pmatrix} C_{00}J & C_{00}G_{01} \\ -G_{10}C_{00}J & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}^T \right)_B^B = \left(\begin{pmatrix} C_{00} & C_{00}G_{01} \\ -G_{10}C_{00} & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}_B^B \right)^T$$

since taking submatrix commutes with taking the transpose; and:

$$(NC)_B^B = \begin{pmatrix} JC_{00} & -JC_{00}G_{01} \\ G_{10}C_{00} & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}_B^B = \begin{pmatrix} C_{00} & -C_{00}G_{01} \\ G_{10}C_{00} & -G_{10}C_{00}G_{01} + K_{11} \end{pmatrix}_B^B$$

Finally, observe that the support of the transpose of the first matrix and the support of the second matrix are identical, as the block expressions only differ by signs, which do not impact the support. By the assumption of C specifying an $\mathbb{F}_d$ -flow on Γ , NC is a DAG. Since only the support is relevant when considering whether a matrix encodes a DAG, $N'C'$ is also a DAG. Thus, by Theorem 4.5, Γ' has $\mathbb{F}_d$ -flow, ending this part of the proof.

($\Leftarrow$): observe that, due to the normalisation of measurement labels, the reversal operation of Definition 6.2 is an involution, i.e. $(\Gamma')' = \Gamma$. Hence the desired result follows from the ($\Rightarrow$) part of the proof. $\square$

H.4 Proving that Z-like removal and insertion are $\mathbb{F}_d$ -flow-preserving

Proof of Theorem 6.5. Let M and N be the flow-demand and order-demand matrices for (G, I, O, λ) respectively and let M' and N' be flow-demand and order-demand matrices for (G', I, O, λ') respectively. Furthermore, let $\bar{O}'$ and $\bar{I}'$ be the set of non-output vertices and the set of non-input vertices of (G', I, O, λ') .

Let C' be the correction matrix for (G', I, O, λ') , whose existence follows by Theorem 4.5 from the assumption that Γ' has $\mathbb{F}_d$ -flow. Thus $M'C' = \text{Id}_{\bar{O}'}$ and $N'C'$ forms a DAG. Since $\lambda'_X(v) \neq 0$, necessarily $v \notin I$ and thus both a v -labelled row and a v -labelled column exist in each of M' , N' , C' , and $N'C'$.

By construction (cf. Definitions 4.3 and 4.4), M and N are submatrices of M' and N' . Let C be the restriction of C' to shape $\bar{O} \times \bar{I}$, the shape compatible with M and N . We now show that $MC = \text{Id}_{\bar{O}}$ and that NC forms a DAG.

The $\mathbb{F}_d$ -flow guaranteed by Theorem 4.5 is focused, so we can apply condition (FZ) to C' to find $C'_{v,w} = 0$ for any $w \in \bar{O}' \setminus \{v\}$. Therefore, for $u, w \in \bar{O}$:

$$(M'C')_{u,w} = \sum_{t \in \bar{I}'} M'_{u,t} C'_{t,w} = \left(\sum_{t \in \bar{I}} M'_{u,t} C'_{t,w} \right) + M'_{u,v} C'_{v,w} = \left(\sum_{t \in \bar{I}} M_{u,t} C_{t,w} \right) + 0 = (MC)_{u,w}$$

Thus MC is the $\bar{O} \times \bar{O}$ submatrix of $M'C'$, which implies $MC = \text{Id}_{\bar{O}}$. By replacing M' with N' in the above equation, NC is the $\bar{O} \times \bar{O}$ submatrix of $N'C'$; and since $N'C'$ forms a DAG, so must NC . Hence, by Theorem 4.5, (G, I, O, λ) has $\mathbb{F}_d$ -flow. $\square$

Proof of Theorem 6.6. Let M' and N' be the flow-demand and order-demand matrices of Γ' , and analogously define M and N for Γ . Suppose that (G', I, O, λ') has $\mathbb{F}_d$ -flow. Our goal is to find sufficient and necessary conditions that λ' must satisfy and which can be checked in $O(n^2)$.

Let $\bar{O}', \bar{I}'$ be the non-outputs and non-inputs of Γ' , respectively, i.e. $\bar{O}' = \bar{O} \cup \{v\}$ and analogously for $\bar{I}'$. By the constructions of flow-demand and order-demand matrices (i.e. Definitions 4.3 and 4.4), we have:

$$M' = \begin{matrix} \bar{I} & v \\ \bar{O} \begin{pmatrix} M & \mathbf{x} \\ \mathbf{0} & \lambda'_X(v)^{-1} \end{pmatrix} \end{matrix} \quad \text{and} \quad N' = \begin{matrix} \bar{I} & v \\ \bar{O} \begin{pmatrix} N & \mathbf{z} \\ \mathbf{n} & -\lambda'_Z(v) \end{pmatrix} \end{matrix}$$

where $\mathbf{0}$ is identically zero. The remaining three submatrices $\mathbf{x}, \mathbf{z} \in (\mathbb{F}_d^{\bar{O}})$ and $\mathbf{n} \in (\mathbb{F}_d^{\bar{I}})^T$ are vectors defined as follows for any $u \in \bar{O}$:

$$\begin{aligned} \mathbf{x}_u &= \begin{cases} \lambda_Z(u)^{-1} S_u & \text{when } \lambda_X(u) = 0 \\ 0 & \text{otherwise} \end{cases} \\ \mathbf{z}_u &= \lambda_X(u) S_u \\ \mathbf{n} &= \lambda'_X(v) (S|_{\bar{I}})^T \end{aligned} \tag{11}$$

In particular, $\mathbf{x}$ and $\mathbf{z}$ do not depend on $\lambda'(v)$ at all, while $\mathbf{n}$ is scaled by $\lambda'_X(v)$ but otherwise is also independent of $\lambda'(v)$.

According to Theorem 4.5, there exists C' such that $M'C' = \text{Id}_{\bar{O}'}$ and $N'C'$ forms a DAG. Since $|I| = |O|$, by Corollary 4.6, this C' is unique. Similarly, C is also unique, and by utilising the proof of Theorem 6.5 which describes the construction of a valid C as a submatrix of C' , necessarily C forms

a submatrix of C' . Furthermore, since Theorem 4.5 guarantees focused $\mathbb{F}_d$ -flow, by (FZ) we obtain $C'_{v,u} = 0$ for $u \neq v$ and by Definition 4.2 we obtain $C'_{v,v} = \lambda'_X(v)$. Thus:

$$C' = \begin{matrix} \bar{I} & \bar{O} & \bar{v} \\ \begin{matrix} C & \mathbf{k} \\ 0 & \lambda'_X(v) \end{matrix} \end{matrix} \quad (12)$$

for some $\mathbf{k} \in \mathbb{F}_d^{\bar{I}}$.

Now, we have:

$$M'C' = \begin{matrix} \bar{I} & \bar{v} \\ \bar{O} \begin{pmatrix} M & \mathbf{x} \\ \mathbf{0} & \lambda'_X(v)^{-1} \end{pmatrix} \end{matrix} \begin{matrix} \bar{O} & \bar{v} \\ \begin{pmatrix} C & \mathbf{k} \\ \mathbf{0} & \lambda'_X(v) \end{pmatrix} \bar{I} \end{matrix} = \begin{matrix} \bar{O} & \bar{v} \\ \bar{O} \begin{pmatrix} MC & M\mathbf{k} + \mathbf{x}\lambda'_X(v) \\ \mathbf{0} & 1 \end{pmatrix} \end{matrix} \quad (13)$$

and:

$$N'C' = \begin{matrix} \bar{I} & \bar{v} \\ \bar{O} \begin{pmatrix} N & \mathbf{z} \\ \mathbf{n} & -\lambda'_Z(v) \end{pmatrix} \end{matrix} \begin{matrix} \bar{O} & \bar{v} \\ \begin{pmatrix} C & \mathbf{k} \\ \mathbf{0} & \lambda'_X(v) \end{pmatrix} \bar{I} \end{matrix} = \begin{matrix} \bar{O} & \bar{v} \\ \bar{O} \begin{pmatrix} NC & N\mathbf{k} + \mathbf{z}\lambda'_X(v) \\ \mathbf{n}C & \mathbf{n}\mathbf{k} - \lambda'_X(v)\lambda'_Z(v) \end{pmatrix} \end{matrix} \quad (14)$$

We want $M'C' = \text{Id}_{\bar{O}'}^{\bar{O}'}$ and $N'C'$ to be a DAG, so we must necessarily have:

$$M\mathbf{k} + \mathbf{x}\lambda'_X(v) = \mathbf{0} \quad (15a)$$

$$\mathbf{n}\mathbf{k} - \lambda'_X(v)\lambda'_Z(v) = 0 \quad (15b)$$

Since M is square and C is its right inverse, C is the unique inverse of M , hence also $CM = \text{Id}_{\bar{I}'}^{\bar{I}'}$ (note that $|\bar{O}'| = |\bar{I}'|$). Multiplying both sides of (15a) by C on the left, we find:

$$CM\mathbf{k} + C\mathbf{x}\lambda'_X(v) = \mathbf{0}$$

which simplifies to:

$$\mathbf{k} + C\mathbf{x}\lambda'_X(v) = \mathbf{0}$$

Therefore:

$$\mathbf{k} = -\lambda'_X(v)C\mathbf{x} \quad (16)$$

Plugging the above into (15b), we find:

$$-\lambda'_X(v)\mathbf{n}C\mathbf{x} - \lambda'_X(v)\lambda'_Z(v) = 0$$

and since $\lambda'_X(v) \neq 0$ and by (11) we find:

$$\lambda'_Z(v) = -\mathbf{n}C\mathbf{x} = -\lambda'_X(v)(S|_{\bar{I}})^T C\mathbf{x} \quad (17)$$

Next, we focus on not-yet considered parts of the product of $N'C'$ from (14). We have by (11):

$$\mathbf{n}C = \lambda'_X(v)(S|_{\bar{I}})^T C$$

and by substituting via (16):

$$N\mathbf{k} + \mathbf{z}\lambda'_X(v) = -\lambda'_X(v)NC\mathbf{x} + \lambda'_X(v)\mathbf{z} = \lambda'_X(v)(\mathbf{z} - NC\mathbf{x})$$

i.e. these expressions have supports that do not depend on $\lambda'_X(v)$. Thus, if $N'C'$ is a DAG for some $\lambda'_X(v)$, it is a DAG for all $\lambda'_X(v)$.

Note that the requirement $M'C' = \text{Id}$ also does not fully fix $\lambda'(v)$, instead (17) gives a formula for $\lambda'_Z(v)$ as a function of $\lambda'_X(v)$. This is to be expected since treating a measurement label as a vector and scaling it by a non-zero number does not change whether there is an $\mathbb{F}_d$ -flow, cf. the discussion before Definition H.18 in Section H.3. Now we may treat $\lambda'_X(v)$ as a variable; let $a := \lambda'_X(v)$. We find that if (G', I, O, λ') has $\mathbb{F}_d$ -flow, then necessarily, for any $a \in \mathbb{F}_d^*$:

$$\begin{aligned}\lambda'_X(v) &= a \\ \lambda'_Z(v) &= -a(S|_{\bar{I}})^T C \mathbf{x}\end{aligned}$$

with correction matrix (from substituting into (12)):

$$C' = \begin{matrix} \bar{O} & v \\ I & C & -aC\mathbf{x} \\ v & 0 & a \end{matrix}$$

Moreover, the following matrix (from substituting into (14)) must be a DAG:

$$\begin{matrix} \bar{O} & v \\ \bar{O} & NC & a(\mathbf{z} - NC\mathbf{x}) \\ v & a(S|_{\bar{I}})^T C & 0 \end{matrix}$$

It is sufficient to check whether the final matrix is a DAG for just one value of $a \in \mathbb{F}_d^*$, for instance $a = 1$. All relevant vectors and the above matrix can be computed in $O(n^2)$ (note that NC is known so it does not need to be computed from scratch). Checking whether a matrix corresponds to a DAG can also be done in $O(n^2)$ [13, Section 20.4]. Thus, in time $O(n^2)$ we verify that either there are no $\lambda'(v)$ resulting in $\mathbb{F}_d$ -flow, or that all pairs $(a, -a(S|_{\bar{I}})^T C \mathbf{x})$ for any $a \in \mathbb{F}_d^*$ (and no other pairs) result in $\mathbb{F}_d$ -flow. $\square$

Remark H.21. The proof above also implies that, for any $\lambda'(v)$ with $\lambda'_X(v) \neq 0$ such that (G', I, O, λ') has $\mathbb{F}_d$ -flow, it is possible to explicitly return the corresponding pair of the correction matrix C' and the DAG $N'C'$ in $O(n^2)$ time, i.e. we can not only find the possible labels for the added vertex, but also update the $\mathbb{F}_d$ -flow accordingly.

H.5 Proofs relating to the generation of instances with $\mathbb{F}_d$ -flow

Usually, the term ‘flow-preserving rewriting’ refers to transformations that preserve both the linear operation implemented by a computation and the existence of flow. Yet the existence of flow (or $\mathbb{F}_d$ -flow) depends only on the labelled open $\mathbb{F}_d$ -graph, which does not contain enough information to determine a specific linear operator. For example, it was pointed out in [3] that the deletion of a YZ -measurement is always flow-preserving, even though it preserves the interpretation only if the measurement angle is 0. Thus, as argued in [2], it can also be useful to analyse rewrite rules that preserve the existence of flow but not necessarily the interpretation.

We expect that the completeness proof from [2] can be adapted to the qudit setting, but leave that to future work, focusing solely on the generation of labelled open $\mathbb{F}_d$ -graphs with $\mathbb{F}_d$ -flow for now. First, we prove a supporting lemma which shows that $\mathbb{F}_d$ -flow-preserving rewriting can be used to transform any labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow into a labelled open $\mathbb{F}_d$ -graph without internal vertices.

Lemma H.22 (Deconstruction of a labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow). Given a labelled open $\mathbb{F}_d$ -graph Γ with $\mathbb{F}_d$ -flow, a sequence of local-complementations and Z -like removals can bring Γ to a labelled open $\mathbb{F}_d$ -graph with no internal vertices.

Proof. Suppose $v \in B$. If $\lambda_X(v) \neq 0$, then, by Theorem 6.5, v can be removed while preserving $\mathbb{F}_d$ -flow. If instead $\lambda_X(v) = 0$, then a local complementation about v (with any non-zero parameter γ) is flow-preserving and transforms $\lambda_X(v)$ to some $\lambda'_X(v)$ with $\lambda'_X(v) \neq 0$, which again can be removed (cf. Proposition H.2). Repeating this process for all $v \in B$ while ignoring any Clifford effects on outputs eventually results in a labelled open $\mathbb{F}_d$ -graph with $B = \emptyset$. $\square$

Proof of Theorem 6.7. Start by generating a random invertible matrix M over $\mathbb{F}_d$, for example by following [35]. Construct the labelled open $\mathbb{F}_d$ -graph Ξ with flow-demand matrix M via the construction from Theorem G.2. By Theorem 4.5, Ξ has $\mathbb{F}_d$ -flow and Ξ is normalised (cf. Definition H.18). Undo normalisation: for each input i , multiply $\lambda(i)$ by some non-zero scalar from $\mathbb{F}_d^*$, transforming Ξ into $\Gamma^{(0)}$. Now repeat the following k times:

- Choose a random vector $S \in \mathbb{F}_d^{\bar{V}}$ and using Theorem 6.6, attempt inserting a new Z -like internal vertex with connections determined by S , trying all measurement labels of the form $(a, b) \in \mathbb{F}_d$ where $a \neq 0$. If the insertion is not $\mathbb{F}_d$ -flow-preserving for any measurement label, choose a different S and try again, otherwise pick any suitable label and move on to the second step.
- Perform a local complementation around the newly inserted vertex with a randomly chosen (and not necessarily non-zero) parameter $\gamma \in \mathbb{F}_d$, ignoring any resulting local Clifford operators on outputs.

Finally, return the resulting labelled open graph $\Gamma^{(k)}$.

The above procedure always produces a labelled open $\mathbb{F}_d$ -graph with $\mathbb{F}_d$ -flow. In particular, the procedure has a non-zero probability of reversing all steps from Lemma H.22 when used on Γ . Thus, with non-zero probability, the above procedure returns Γ . $\square$

As of now, our method is not necessarily efficient: Choosing a random vector $S \in \mathbb{F}_d^{\bar{V}}$ for connectivity of the new vertex may fail many times before finding a vector that does permit vertex insertion. One can observe that an identically zero S always permits insertion, thus, with probability 1, our algorithm eventually performs a vertex insertion (albeit potentially a trivial one). However, bounding the efficiency of the insertion in a more meaningful way requires deeper analysis of the structure of all vectors S permitting insertions. We leave this to future work.